Simple Model-Based Stock Market Prediction; Predicting the Dow Jones Industrial Average Index Value Based on News Headlines

Khaled Tamimy and Berend Jansen

December 2016

1 Introduction

Within the field of financial investing, the stock market plays a vital role (Hamed et al. 2012), and due to economic globalization and technology advances, this particular field of finance has opened up in such a way that nearly anybody can own, trade, or manage stocks (Seidy 2016). The enhanced accessibility of the stock markets has led to the emergence and demand of analysis and prediction of stock market values.

This analysis and prediction builds on the dynamical management of financial information portfolios, which are ensembles of economical information of which stock market trends can be deduced. Stock market actors, then, base their financial strategies on these respective portfolios, as well as adopt and adapt to the societal and market circumstances. Aside from the enormously large volume of information, stock market data also tends to be highly dependent on time and, unfortunately, nonlinear of nature (Seidy 2016). Consequently, effective analysis and prediction of stock markets by humans has become nearly impossible, and extremely hard at best, thus leading to the evolution of computer-algorithm-based analysis and prediction.

The use of computer algorithms in the analysis and prediction of market information has been an emerging field within both the fields of applied machine learning as well as finance for the past decades (Li et al. 2016), and writing programs that enable one to make predictions of and on the financial market has been one of the most challenging goals of applied machine learning (Hamed et al. 2012). While many of these computer programs focus on the quantitative information available, only few focus on qualitative information. One specific example of qualitative information and its influence on the stock market, is the strong correlation between (financial) news reports and stock market values, as studied by Tetlock (2007) and Schumaker et al. (2009).

In this essay, an attempt is made to write a simple machine learning program that uses daily news headlines to predict the behaviour of stock markets. A news report headlines data set is retrieved from Kaggle, an online datascience platform for (aspiring) machine learning scientist or appliers (see references). This data set is tuned to be able to make predictions on whether the Dow Jones daily index value increases or decreases according the the news report headlines of that respective day.

First, a brief description of the data set is provided. Following, this essay describes the methodology used, by describing the data set and the approach of data processing, as well as the justification for choices made within the context of the research. Subsequently, the results are presented and evaluated. Finally, a discussion is provided and an evaluation of the approach and solution is given.

1.1 The Data Set

The data set is retrieved from Kaggle, as explained in the introduction, and its location, and that of Kaggle, can be found in the references. The data consists of two subsets, one news data set and one stock data set. As for the former, it is collected from the Reddit Worldnews Channel. Public users and followers of this channel can 'upvote' certain news reports. Based on the ranking obtained from this 'upvoting', the top 25 headlines of the day were collected, for the period range: 2008-06-08 to 2016-07-01. These 25 top headlines per day are saved in the file 'RedditNew.csv', in which it is saved in a tabulated way; the first column representing the dates, and the second containing lists with the 25 top headlines of that date. As for the stock data set, it contains the Dow Jones Industrial Average index values (DJIA) on opening, peak, through, and closing for every date in the period range of the news data set (2008-06-08 to 2016-07-01). It is collected in the file 'DJIAtable.csv', and it can be retrieved directly from Yahoo Finance. However, stock market data is besides its high dependence on time and nonlinearity, very inconsistent, thus making it hard for regression solutions. Besides, as the problem in this research is to predict increase or decrease in the DJIA, it is more naturally to consider classification solutions. To be able to do so, the data set was altered as follows:

$$\begin{gathered}\text{For day } i\text{:}\\ \text{If } (DJIA)_i^{closed} - (DJIA)_{i-1}^{closed} \geq 0, \text{ set } Label = 1\\ \text{If } (DJIA)_i^{closed} - (DIJA)_{i-1}^{closed} < 0, \text{ set } Label = 0\end{gathered} \tag{1}$$

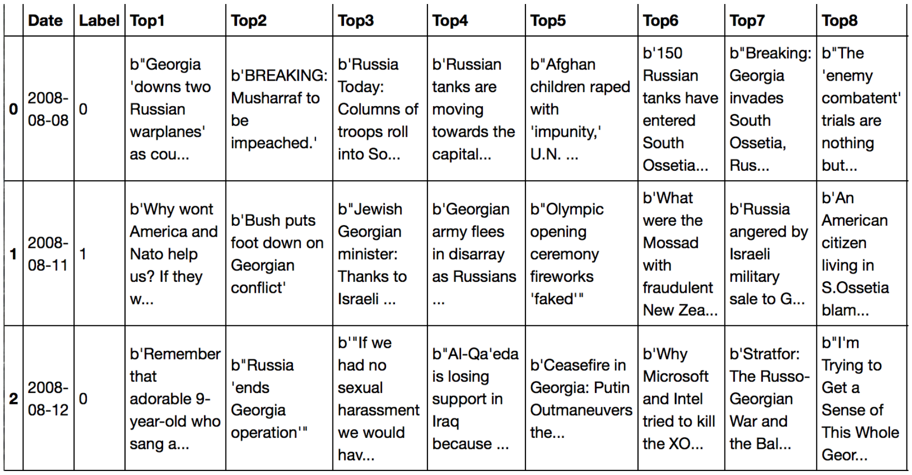
The dates with their corresponding labels, and the 25 top headlines were collected in 27 columns, where the first column contains the dates, the second the label and the following 25 containing the top 25 headlines of that day (see figure 1). It is saved in the file 'CombinedNewDJIA.csv'.

2 Methodology

The methodology of this research consists of two main parts: the pre-processing of the data and the application of the classifiers.

2.1 Pre-Processing the Data Set

To be able to process the data using scikit-learn (SK) classification algorithms (see references), the data had to be pre-processed. As the data consists of natural language, the main part of the pre-processing is the processing of the natural language. To do so, a SK function, CountVectorizer, was used. This function processes natural language into document-term matrices. To use CountVectorizer, an array was made with strings as entries. Every string inside the array represents all the news headlines from one day. Since the array contains all the headlines, it also contains all the words in the data set, and therefore one could see it as a dictionary. This array can be converted into a document-term matrix by the fittransform method of the CountVectorizer function. In this matrix, all the words in the dictionary are on the horizontal axis and the vertical axis consists of all the days in the training set. The entries in the matrix tell how many times a word appears on a particular day. For example, entry 100x56 shows how many times the 56th word from the dictionary appears on day 100. The same was one with the test set, but keeping the dictionary from the training set. Also, using the pandas library it was possible to visualize the data, see figure 1 below. It is a limited visualization since the data set is very large, but it helps understanding the set.

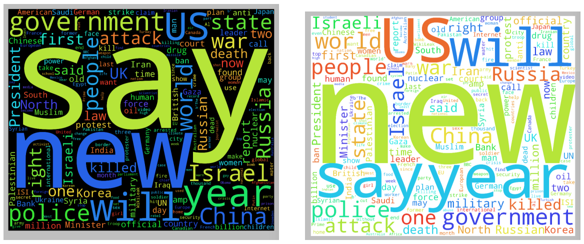


	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8
0	2008-08-08	0	b"Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b"Afghan children raped with 'impunity,' U.N. ...	b'150 Russian tanks have entered South Ossetia...	b"Breaking: Georgia invades South Ossetia, Rus...	b"The 'enemy combatent' trials are nothing but...
1	2008-08-11	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b"Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b"Olympic opening ceremony fireworks 'faked'"	b'What were the Mossad with fraudulent New Zea...	b'Russia angered by Israeli military sale to G...	b'An American citizen living in S.Ossetia blam...
2	2008-08-12	0	b'Remember that adorable 9-year-old who sang a...	b"Russia 'ends Georgia operation'"	b'"If we had no sexual harassment we would hav...	b"Al-Qa'eda is losing support in Iraq because ...	b'Ceasefire in Georgia: Putin Outmaneuvers the...	b'Why Microsoft and Intel tried to kill the XO...	b'Stratfor: The Russo-Georgian War and the Bal...	b"I'm Trying to Get a Sense of This Whole Geor...

Figure 1: Visualization of the data set

Wordclouds are very useful to get an idea of what the most appearing words are. Using the wordcloud function developed by MIT, one was made for the most appearing words in the headlines on a non-decreasing day, and one for the most appearing words in the headlines on a decreasing day (see figure 2 below).

As language is powerful within its context, words alone would not mean very much, neither for humans nor for the computer. Therefore, an important



(a) Words appearing on non-decreasing day

(b) Words appearing on decreasing day

Figure 2: Wordclouds

parameter inside the CountVectorizer function is n-gram. In this case, an n-gram is a sequence of words extracted by the CountVecotrizer function. How this actually works can be explained by an example. Consider the sentence ”The cat runs outside”. CountVectorizer with n-gram range (1,1) would make a document-term matrix with on the horizontal axis, the words: ”the”, ”cat”, ”runs”, ”outside”. With the n-gram range on (2,2), it would make a different matrix with the following entries on the horizontal axis: ”the cat”, ”cat runs”, ”runs outside”.

Also removing words without a real meaning is important when processing natural language. A parameter in CountVectorizer called ’stopwords’ does this. Using the standard ’English’ list, around 400 words were removed.

2.2 Application of Classifiers

The classifiers used in this research were all obtained from SK. As the dependent variable in this research is a binary variable, decreased or nondecreased, the use of classification algorithms seemed natural. Additionally, as the input data, after preprocessing, was also of a binary kind -a word or set of words eithers appears in the headline or not-, a network algorithm seemed natural as well.

The two algorithms used in this research are Logistic Regression and K-Nearest Neighbor; both will be analyzed briefly below.

2.2.1 Logistic Regression

Stock markets are highly time dependent and nonlinear of nature. These characteristics, among others, make stock markets nearly impossible to predict accurately. Therefore, within the field and context of stock market predictions, one wants a probabilistic model, as a prediction will never be completely certain; stock market predictions add to information portfolio within the concept of risk management. Additionally, logistic regression assigns coefficients/weights to all words or combinations of words -thus features-, allowing for a variety of ways to regularize the model; unlike with Naive Bayes, one does not have to worry about the features being correlated and the algorithm weighs the importance of every feature. This, together with the large size of the data set were justifications for the use of logistic regression.

2.2.2 K-Nearest Neighbor

When the data set is small, high bias/low variance classifiers, such as Naive Bayes, seem to be a better choice than low bias/high variance classifiers, such as K-Nearest Neigbord, as the latter are prone to overfitting. However, when the data set is large, as is the case in this research, low bias/high variance classifiers have an advantage over the high bias/low variance classifiers, since the latter will often not be 'strong' enough to predict accurately. Additionally, K-Nearest Neighbor decision boundaries are not restricted in form since this classification algorithm does not make assumptions about the distribution of the data. Hence, a K-Nearest Neigbor algorithm was used.

3 Results

3.1 Logistic Regression

While varying the parameters, a n-gram range of (2,2) gave the highest accuracy for logistic regression. Also, after running several tests, it appeared that removing the digits from the dictionary containing the features enhances the predictions as well.

The highest-achieved accuracy for logistic regression was 56.7%. The ROC curve for this algorithm is plotted and shown below (figure 3). Additionally, the weights of combinations of words is shown in figure 4.

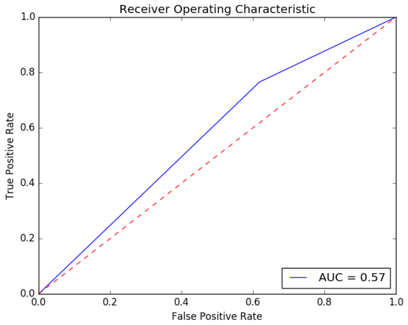
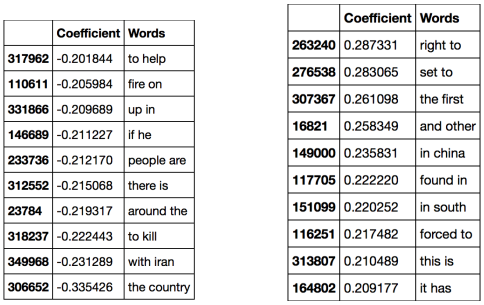


Figure 3: ROC curve logistic regression



	Coefficient	Words
317962	-0.201844	to help
110611	-0.205984	fire on
331866	-0.209689	up in
146689	-0.211227	if he
233736	-0.212170	people are
312552	-0.215068	there is
23784	-0.219317	around the
318237	-0.222443	to kill
349968	-0.231289	with iran
306652	-0.335426	the country

(a) 10 words with lowest coefficients

	Coefficient	Words
263240	0.287331	right to
276538	0.283065	set to
307367	0.261098	the first
16821	0.258349	and other
149000	0.235831	in china
117705	0.222220	found in
151099	0.220252	in south
116251	0.217482	forced to
313807	0.210489	this is
164802	0.209177	it has

(b) 10 words with highest coefficients

Figure 4: Logistic Regression parameter values (weights)

3.2 K-Nearest Neighbors

With the K-Nearest Neighbors algorithm, several good accuracies were found. For a n-gram range of (2,3), an accuracy of 55,5% was found; for a n-gram range of (1,1), an accuracy of 54,5% was found; and for a n-gram range of (1,3), an accuracy of 54,49% was found. All three accuracies were found at different k-values, respectively: 97, 87 and 7. However, the highest accuracy was found at a n-gram range of (2,2), just as with the logistic regression. By running a loop, the optimal value for k was found to be at 215 (see figure 5 below), with an accuracy of 56.9% (see figure 6 below).

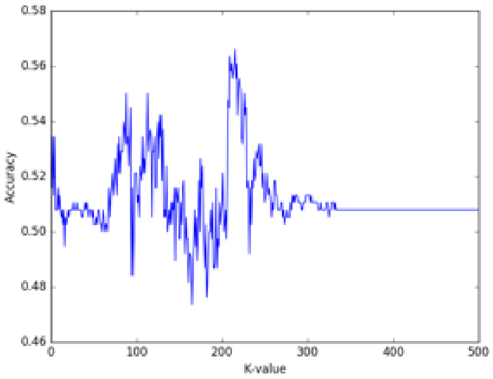


Figure 5: Accuracy vs. K-value plot; finding optimal K-value

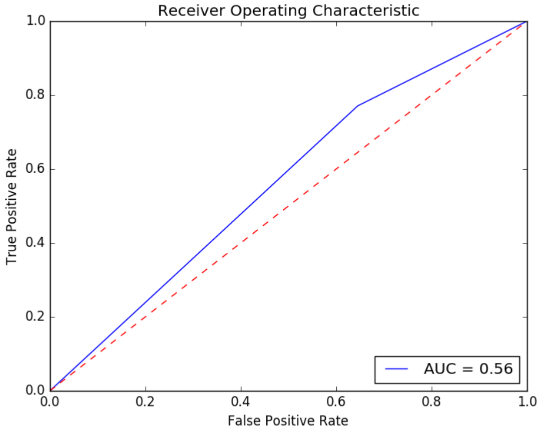


Figure 6: ROC curve k-nearest neighbors

4 Evaluation of Approach and Solution

In this paper an attempt is made to make a simple stock market prediction program. The problem considered is a prediction on whether the DJIA index value decreases or non-decreases at a certain day, according to news headlines of that day. The problem is treated as a classification problem, and suitable classification algorithms were chosen. The accuracy to which this prediction holds is limited due to the complexity of stock markets; reaching accuracies higher than 58% seems nearly impossible. The classification models used in this research were Logistic Regression and K-Nearest Neighbors. These classifiers achieved accuracies of 56.7% and 56.9% respectively. As noticeable from the results, the K-Nearest Neighbor algorithm as classifier seemed more suitable than Logistic Regression, although the difference was small.

Additionally, both classifiers are prone to a variety of variables, of which the many were not taken into consideration. Some limitations of this research were, for example, the fact that the words based on the same nouns were considered as different words. Using a stemmer algorithm, one could solve this problem by shortening words to their noun-roots, which would result in a slighly smaller, but more accurate data set. An advantage of this would be that there would be no difference between plural and singular words, nor between words that have the same semantic meaning but are written slighly different. Besides, predicting stock markets solely on qualitative data -news articles in this case-, can never be very accurate. This, of course, is also a strong limitation of this research. Also, as the training data consists only of news headlines from 2008 to 2014, words or phenomena that arose later than 2014 will not be considered in the algorithm. Another major limitation is the fact that the headlines were separated into words, thus depriving them of their semantic meaning or significance. For example, the words 'Russia' may appear in both decreasing and non-decreasing days; it depends, however, on the context of the headline whether the DJIA index value would actually increase or decrease within the scope of this word. Finally, a more general limitation is the fact that stock markets are too complex to be modeled by such a simple algorithm. In future research, one could improve prediction of stock markets by applying more complex classification algorithms. Using more sophisticated natural language processing algorithms, thus involving information extraction techniques, would also be a great improvement. Additionally, as the input data for the classifiers consists of either words or short combinations of words, one might involve a correlation analysis, to be able to rule out certain classifiers and try to 'catch' relational meaning between words and word-combinations. All with all, this simple model-based research provides a simplistic overview of how one could apply machine learning within the field of financial economics. Although this research is limited by a variety of factors, it is still indicative of the approach one could take while trying to analyse or predict stock market values.

5 Bibliography

[1] Hamed et al. 2012. An Intelligent Model for Stock Market Prediction. International Journal of Computational Intelligence Systems. 5(4): 639-652

[2] Li et al. 2016. Empirical Analysis: Stock Market Prediction via Extreme Learning Machine. Neural Computing and Applications. 27(1): 67-78

[3] Schumaker and Chen. 2009. Textual Analysis of Stock Market Predicting using Breaking Financial News. Computer. 43(1):51–56

[4] Seidy. 2016. A New Particle Swarm Optimization Based Stock Market Prediction Technique. International Journal of Advanced Computer Science and Applications. 7(4): 322-327

[5] Tetlock. 2007. Giving Content to Investor Sentiment: the Role of Media in the Stock Market. J. Finance. 62(3): 1139-1168