Reazul Hoque

Mitchell Wade

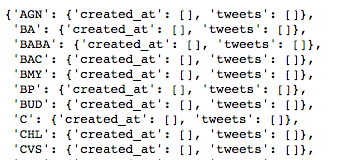
Curtis Wilkerson

Miller Moore

December 9, 2015

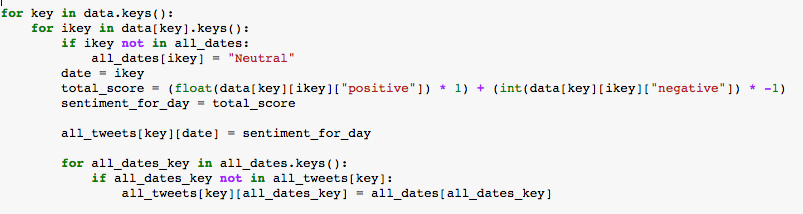
**Effect of Tweet Sentiment on Future Stock Price**

1. Project Goal: The goal of this project was to determine whether tweet information from the largest 50 stocks traded on the New York Stock Exchange (NYSE) is associated with future price movements in these stocks.
2. Project Plan: The team planned to complete most work using github and docker containers for writing code. The team pursued development of a web scraper to obtain free tweet data from Twitter since API access is prohibitively expensive. The team planned to use available natural language processing libraries to convert raw tweet data to sentiment scores. Sentiment scores would then be used to create various historical sentiment features as inputs to a data mining algorithm. The modeling algorithm would be trained against a vector of future price movements. Once trained, the model would be evaluated against a future time period not available to the algorithm during training.
3. Project Execution: A web scraper was developed in Python to grab tweets from Twitter. In the interest of time, the team requested a year’s worth of tweets from StockTwits, which was provided in a structured JSON file. The JSON file was processed into a dictionary object where primary keys were stock symbols, secondary keys were dates, and secondary values were tweets. Subsequently, the tweets were scored using the nltk.sentiment.util library, which converts a text string to a single number based by summing the scores of each word inside the string (where 1 is positive, -1 is negative, and 0 is neutral). Scores per tweet were summed to create scores per day for each of the 50 stocks. This information was then written to a csv file to be imported by Rfor preparing sentiment variables and training a model. Example code for the above steps is shown below:

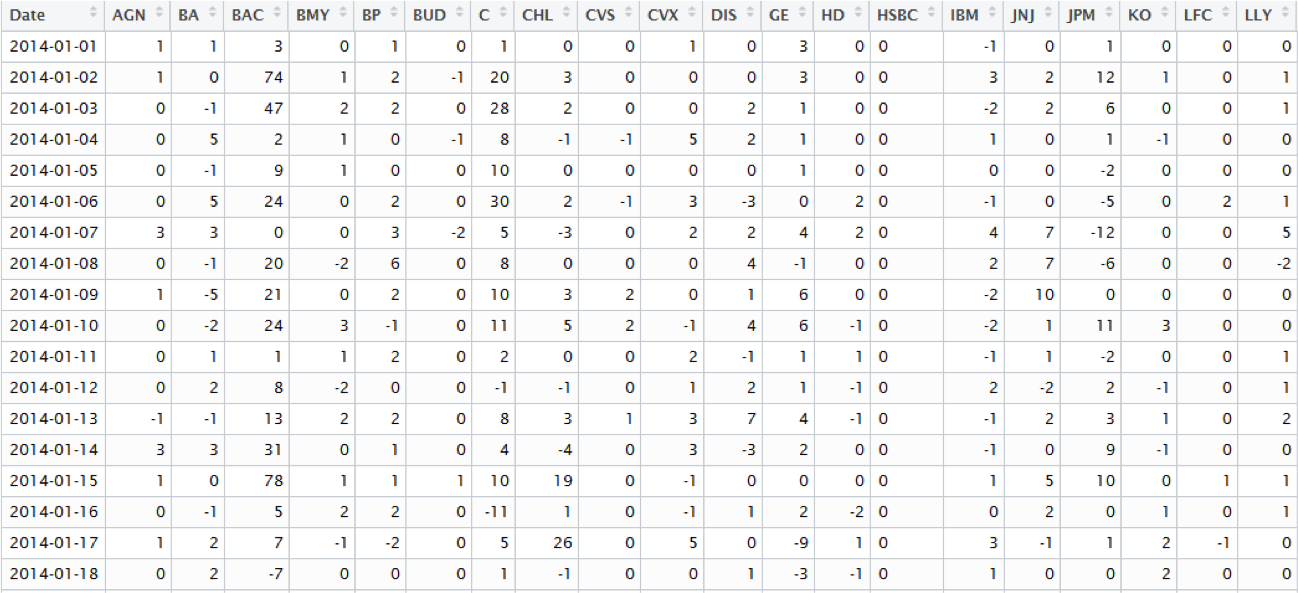
Sample of dictionary used to store extracted data:

Code snippet showing the sentiment scoring:

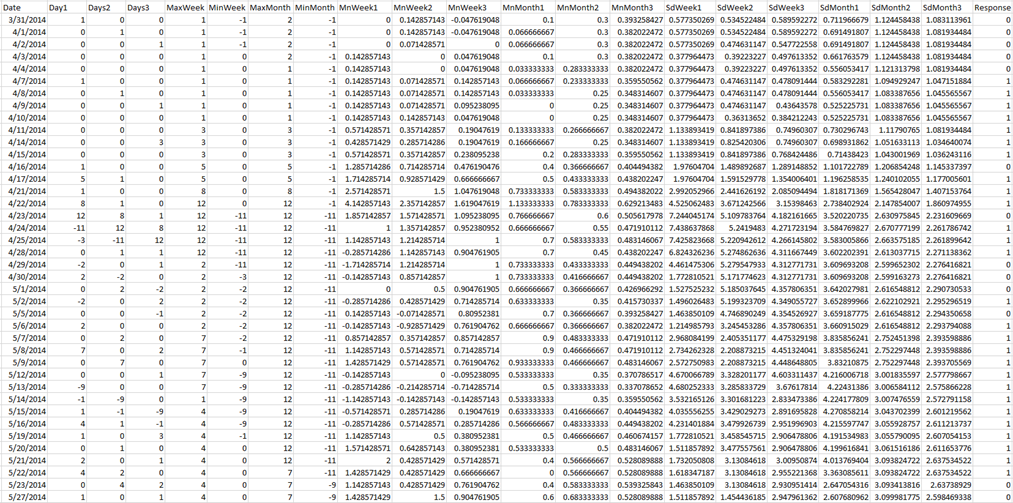
 Code snippet showing aggregation of all scores per day:



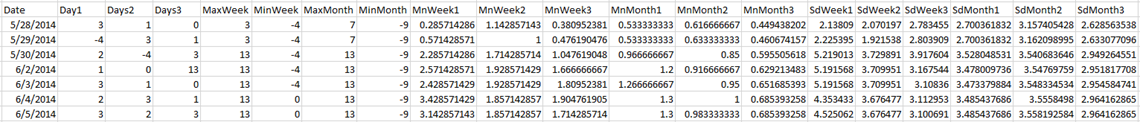
The data was processed to the following structure after aggregation:



1. Preparation of Data for Modeling: Once imported into R, an initial training data set was created where unique rows represent a date. Information at each date were assumed to represent independent observations for modeling purposes. In this analysis, all stocks were grouped together and individual stock effects were ignored and assumed to be independent from one another. Variables for each observation were created by aggregating daily sentiment scores over different historical periods (where *historical* means any data available prior to the observation’s (row) date). Specifically, the following variables were created representing aggregations/transformations of the daily sentiment scores: sentiment 1, 2, and 3 days ago (Day1, Day2, Day3), the maximum and minimum sentiment of the prior week and prior month (MaxWeek, MinWeek, MaxMonth, MinMonth), the mean sentiment of the prior 1,2, and 3 weeks and prior 1, 2, and 3 months (MnWeek1, MnWeek2, MnWeek3, MnMonth1, MnMonth2, and MnMonth3), and the standard deviation of sentiment over the prior 1, 2, and 3 weeks and 1, 2, and 3 months (SdWeek1, SdWeek2, SdWeek3, SdMonth1, SdMonth2, SdMonth3). The response variable at date *t* for stock *i* was created using price data for each stock, where the response is “1” if Price\_*i* at *t*+*h* is greater than Price\_*i* at *t* and “0” otherwise. The *h* represents the forecast period and was chosen to be 7 trading days in this analysis. An example of the initial training matrix is shown below:



1. Modeling: Based on documented good “out-of-the-box” performance, a random forest algorithm in R was chosen as the modeling technique for this analysis. At each training period, the random forest model was trained on a matrix similar to the one shown above using a rolling time period type methodology. For each time period, a matrix was created per stock. When each stock’s matrix was complete, all matrices were bonded together into one larger matrix to maximize the number of observations available for training. Each test period consisted of 7 trading days per stock, which progressed forward in time, following the rolling training methodology. The testing table was also generated per stock and bonded together at completion of all stocks in a similar fashion to the training matrix. An example of the testing matrix for one stock is shown below, which does not include the response variable since the response variable should not be a variable when making predictions:



At each test period, the predictions were compared to the actual up or down price movement for subsequent statistical analyses, described in the next section.

1. Analysis Results: After obtaining predictions at each test period, the Receiver Operating Characteristic (ROC) curve was generated along with the area under the curve (AUC) as a metric for performance. Further, a confusion matrix was developed between predictions and actual results and a chi-squared test of independence was performed on the confusion matrix of predicted labels versus actual labels. AUC and chi-square test p-values were stored into a vector at completion of each rolling test period. The following plots show the AUC values (left plot) and chi-square test p-values (right plot) over the full testing period:

|  |  |
| --- | --- |
|  |  |

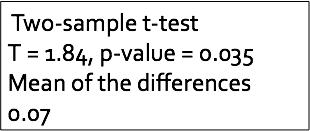
An AUC of 0.5 is the expected performance of a binary classification model by random chance. Except for one 7-day period, all AUC’s were above 0.5, indicating the model performs better than random during almost all time periods between 2015-05-28 and 2015-12-22. The chi-square p-values are less convincing with only 16/27 weeks indicating a significant association between predicted price movements and actual price movements. However, AUC values are judged to be more indicative of trading performance since the ROC curve accounts for all thresholds, allowing the trader to select optimal thresholds or the use of ranked scores to better select trading positions. Chi-square tests are performed on the table of predicted labels based on a threshold of 0.5, where the model predicts “Up” price movement if the model’s score is greater than 0.5 and “Dn” price movement otherwise. Further, a p-value of 0.05 may be too restrictive to use as a measure of significance in a trading scenario since only a slight edge is needed to outperform the market (i.e., the consequences of not recognizing an association between predictions and actual response may be just as great as the consequences of deciding there is an association when there is none).

1. Alternate Analysis:

From the testing above, it is clear that tweet sentiment can be used to predict stock price fluctuations. However, to determine tweet sentiment data’s usefulness, it needs to be shown that the information in tweets can improve a model predictive power beyond what is capable with traditional features such as price lag features.

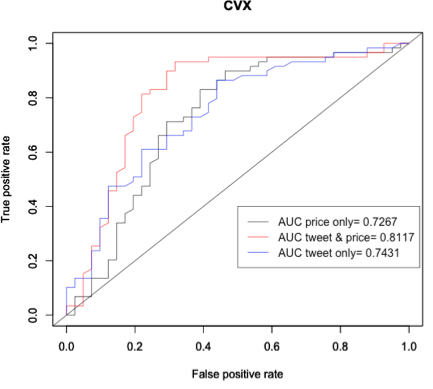
In order to analyze the marginal improvement tweet sentiment data can provide over price lag features, an alternate test was conducted. For each stock, two random forest models were built, holding all hyper parameters constant. One model included only price lag features and the other included both tweet sentiment feature and price lag features. In order to make the fairest comparison, the features included for the price lags and sentiment data were analogous. AUC over a testing period of 90 days was chosen to measure performance differences between the two models. This test was conducted on the top 50 stock of the S&P 500, excluding BABA and HSBC due to missing data.

To analyze the improvement tweet sentiment data can offer, both AUC’s for each model, for each stock (96 total AUCs) a pair two-sample t-test was conducted. The results of the test are shown below.

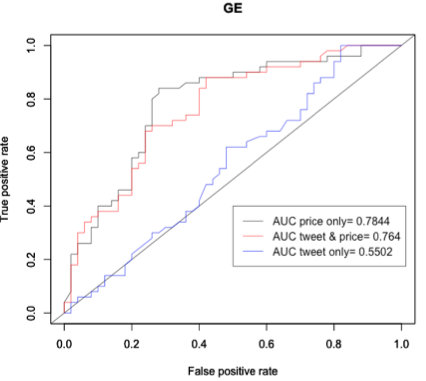


At a 5% confidence level, it can be shown that there is an improvement in the models with tweet data on average. Overall, including tweet data improves the AUC by about 1% on average.

Below is a ROC plots showing the stock that had the largest improvement in AUC when adding tweet data. In fact, for this stock, for the given testing period, a model constructed using only tweet data outperformed the model with only price lag data. This is the only stock where this was the case. By including both tweet and price lag data, a significant jump in performance can be seen.



Below is an ROC plot of the company that performed the worse when including tweet data. It is worth noting that for GE’s tweet only model, the performance is barely better than random. Additionally, we see a decrease in performance by including both price lag and tweet data over only price lag data. This is probably due to throwing in more noise in the model and overfitting on the noise.



1. Conclusions: Based on the analysis results shown and discussed above, sentiment information obtained from tweet data seems to be significantly associated with future stock price movement. Therefore, models used to make trading decisions should not exclude this information if maximum predictive performance is desired.