When Trump Tweets, Wall Street Trades

Group 4- A and his friends

Project Overview

The project, by processing and analyzing twitter data, attempted to understand how the market reacts to President Trump's tweets. The project is interesting considering the following aspects:

1. We focused on Trump's tweets specifically and applied machine learning algorithms to predict the stock price change based on Trump's tweets.
2. We further output how each word in Trump's tweets influence the stock price.
3. The stock price was recorded by every 1 minute, ensuring that macroeconomic factors were controlled in modeling.

Part 1. Data Scraping

1.1 S&P 500 Index Price: We downloaded the S&P 500 Index Price from Bloomberg Terminal. Assuming that Trump's tweets could only influence the stock price in a short period, we acquired the intraday historical S&P 500 index close price by 1-minute (60807 records), 3-minute(32421 records) and 5-minute(19453 records) interval respectively.

1.2 Twitter: Using Twitter API and 'tweepy' package, we did web scraping on Twitter. We mainly used 'user_timeline' function to get the content of Trump's tweets from Dec 2017 to Dec 2018. The total number of records is 3214.

Part 2. Data Processing

2.1 S&P 500 Index Price

Next, we processed to match the time of stock price and Trump's tweets. At first, we imported 'time' package and used strptime() function and mktime() function to convert all the dates stored as string to integers representing seconds since the epoch for further calculation. When Trump posted his tweets at time t_0, we would compare the stock price at t_0 and that after 1/3/5/10/15/30 minutes. If the stock price went up, this record was marked as tag '1'; if the stock price tumbled, the tag should be '-1'; otherwise, '0'. The project did not count those tweets posted out of trading time (from 9:30 to 16:00 on weekdays).

2.2 Twitter

We accessed 3214 rows of tweets through API totally and firstly converted the dates of tweets into Datetime Objects. Then by using pandas.merge, we merged tweets data and stock price data into one frame. Next, we got rid of tweets that have missing stock price change, remaining 1094 tweets. Then we defined a tokenize function by using str.maketrans, through which we removed all the ASCII punctuations and numbers, tokenized the sentence and built a word list. To count word frequency, we

applied FreqDist() to tokenized text for each tweet, and then added up among all tweets and got the frequency for each word appearing in tweets. After drawing the histogram for frequency, we found most words' frequency is between 2 and 100, hence we generated a selected list of word, containing these words, and the rest that is meaningless was put into stop-word.

Part 3. Modeling and Results

1. In text mining we used Naïve Bayes, Linear SVC and SGD classifier, all of which are powerful supervised learning model.
2. In order to generate word vectors more efficiently, the CountVectorizer and TfidfVectorizer functions which are based on the sparse matrix in scipy were imported.
3. To generate the significant words, we calculated the probability of words conditional on each class. This can be realized by the coef_ function in sklearn.
4. We built a pipeline model for training and testing. The greatest advantage is that when there is a new sample, we can quickly generate the prediction.
5. Multinomial Naïve Bayes beat the other two models. Particularly, the Naïve Bayes model with Tf-idf vectors input performed better than Count vectors. When the price was recorded by every 1 minute and 30 minute, the precision of Naïve Bayes model reached highest score, 0.56 and 0.57 respectively. Therefore, we chose 1-minute interval Naïve Bayes model with Tf-idf vectors input for prediction.

Part 4. Data Visualization

4.1 Twitter

We analyzed Trump's Tweets from several perspectives. Firstly, we figured out it seemed that Trump only sleeps five hours every day(4:00 am-9:00 am) according to the number of Tweets Trump posted in each hour. Secondly, Trump becomes more and more interested in posting his ideas and opinions through Twitter due to the increasing number of Tweets he posted every month from Dec 2017. Additionally, through Tweets length analysis, the average length of Tweets amounts around 20-30 words, more than 90% of the selected tweets are within length 30. Last but not least, we analyzed the content of Trump's Tweets through WordCloud and Word2Vec to find out high-frequency words and the words that are similar to each other. The high-frequency words include 'border', 'job', 'trade', 'crime'. The graph representation of embedding of words after processing by PCA intuitively shows that words having vectors close to each other are similar, like 'China', 'dollar' and 'tax'.

4.2 GUI

In order to make a more readable interface for prediction, we utilized 'tkinter' package to generate a GUI. By inputting trump's tweet and pressing predict button, we could get the predicted result directly.