Matt Fannin

Prof Quarles

SI 330

4/22/2020

SI 330 Final Project: Financial Markets in the News

Motivation:

a.) In my final project, I am trying to determine whether or not the frequency of a company in the news has an impact on that company's stock price, or volume of trades of that stock. I decided to choose this topic because I have always liked economics and finance. I enjoy researching stocks and deciding whether I should purchase a piece of a company. I decided to compare financial markets to news headlines because I feel as if there are always articles about certain companies such as: Google, Facebook, and Amazon. This made me curious if the number of articles written about a company has an effect on trader's attitudes towards the company, resulting in a decrease or increase in demand for a company's stock.

b.) Do companies in the news more often have higher stock prices?
Do companies in the news stocks get traded more often?
Is there common language used in articles written about different companies?
Perform an analysis on one companies' stock of my choosing: Amazon

Data Sources:

The financial data was retrieved by making a call to the Alpha Vantage API. Using the API, I decided to gather stock prices for ten companies: Thomson Reuters, Facebook, Apple, Amazon, Netflix, General Motors, Starbucks, Nike, Boeing, and McDonald's. For each company I grabbed its: open price, high price, low price, closing price, and volume traded for days that are spaced one month apart. The first entry is 4/2/2020 and dates all the way back to 4/28/2000, if the company has been publicly traded that long. Most companies that are technology related have not been traded that long, so the data begins at the company's initial public offering (IPO) date. The data was formatted in a dictionary nested inside of a tuple. I used the first key to grab the data I wanted, then saved the keys (dates) as the index in a data frame and the values (stock data). So, each company has 241 rows of data.

The news article data was retrieved by making a call to the News API. In the call to the API I was able to specify the: company name, the sources, and the dates to gather articles from. I decided to gather news data from: Bloomberg, the Wall Street Journal, and Reuters. The article data is gathered from 3/30/2020 back to 7/31/2002. The data that was returned from the API was formatted in a dictionary. The values that I decided to keep the following: ‘totalResults’, ‘title’, ‘description’, ‘publishedAt’, and ‘content’. The API only allowed me to get all this data for 20 articles, so the data frame created as 20 rows.

Data Processing:

First, I had to clean the financial data after making the ten calls to the API. To begin, I created a function that would grab the values I wanted to keep and create a data frame from the call. Then, to make labels for each company I created a column with the company’s name, with blank rows, to serve as a label. I did this for each of the ten data frames. Next, I combined all ten data frames together using concatenation to make one. After this, I used fillna to replace all NaN values with a 0. I changed the name of the column ‘index’ to ‘date’ then reset the index. Lastly, the data frame created was saved as a CSV.

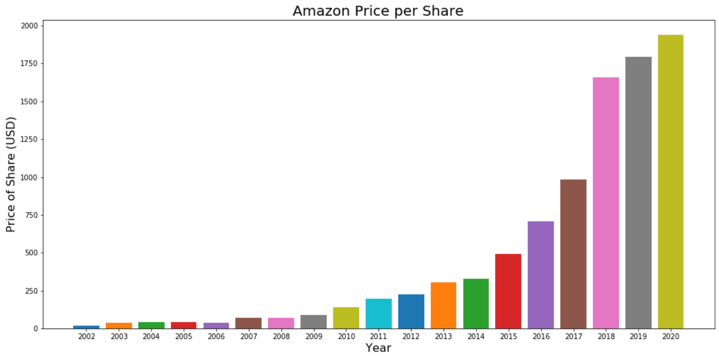
When beginning the process of cleaning the news data, I made the call to the API ten times. A function was created that would append the values of certain keys to lists. The function looped through the ‘articles’ key in order to append the: ‘title’, ‘description’, ‘publishedAt’, and ‘content’ to lists, the function then returned these lists. I then had to clean the dates, I had to keep the first ten characters and remove the rest. I decided to make a function to accomplish this. I then created the data frames and did a similar trick by adding the company name to serve as a label, and make the rows blank. Then, I used concatenation to combine the ten data frames into a single data frame. Lastly, the data frame was saved a an excel file (xlsx).

I imported the two data frames into a new notebook. I had to use fillna in order to get rid of the NaN values that were in the company name columns, I replaced them with blank spaces. Before merging the data frames together, in the financial data I had to only get the data dating back to the beginning of 2017 to correspond with the news data. To do this, I utilized iloc to select data I wanted to use in the merged table. In addition, I used iloc to select the all the financial data for each company and save them to a variable. I repeated the same process, using iloc, to select each companies' news data and save them to variables as well. Now the data was ready to be merged. Each company was merged individually using the merge function. I merged left and right on the date columns using an outer merge. I did this for each of the ten companies, then using concatenation made one complete data frame. This final data frame was saved to the variable named 'full_df', and it contained 234 rows and 109 columns. The data was then ready to be analyzed.

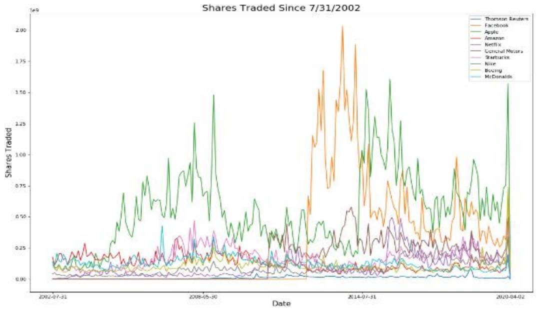
- When importing both data frames in a new notebook, the data frames had two indexes the date and an index that started at 0.
  - When saving the data frames inside the method I had to use index = false to fix the issue
- All of the dates of new articles being published were different than the dates used in the financial data frame
  - I used an outer join so that I could keep all the data in the newly created data frame

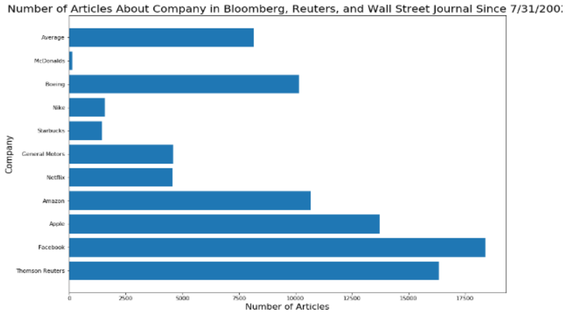
Analysis and Visualization:

First, I wanted to take a look at Amazon's stock price and how it has changed since its IPO. To begin I made a new data frame that contained just the dates and Amazon's stock data. I averaged its closing price and opening price to find that on the dates in the data, its stock grows on average by $8.23 a day. I then created a function that is passed in the data frame and a year as a string. Using the year passed in, it searches the data frame for that year and averages the stock price for that year and returns that value. I then create a bar plot that shows the growth of the company. It appears that the demand for Amazon's stock has increased exponentially. I do not see their price falling in the future, Amazon is one of the few companies who will grow stronger due to covid-19, and continue to grow.

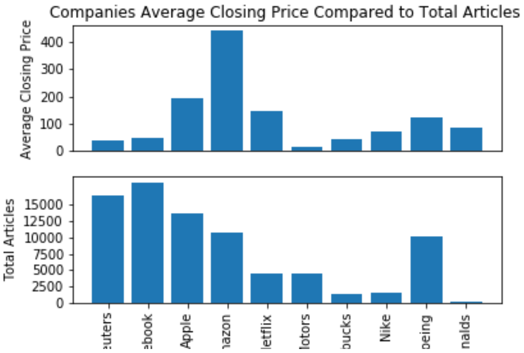


Next, I wanted to see if there was any relationship between the number of articles written about a company and how much their stock was traded. To do this I used iloc to get all the total entries columns, then found the average, lastly saving them to a variable. Also, I made a list of the company's names to serve as labels on my plot. Next, I gathered the volume and dates and saved those values. To do this I had to use the sort_index method. In the beginning I struggled because my data was plotting backwards and couldn't find out why. From this it seemed that Facebook was mentioned the most in the news and also was traded the most. Apple was traded the second most and had the third most articles. Although this was interesting there was no obvious relationship that found. Initially, I wanted to compare volume traded and articles written by year, but the news API only gave me the total number of entries, so I had to change my approach.

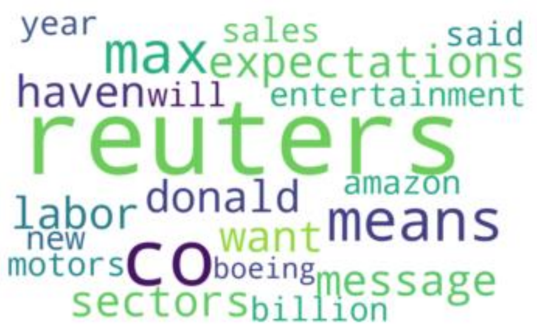




Next, I wanted to see if companies mentioned in the news more often had higher stock prices. In order to answer this, I used iloc to the dates closing prices for each company, and then made a new variable to store the averages. I created bar plots to compare the two figures. Unfortunately, no clear relationships emerged. Along with the bar plot, I created separate data frames that show numerically first sorted from highest stock price to lowest. Then, the next data frame was sorted from most articles written to least articles written. When creating these data frames, I had to create new indexes and rename columns in order make it look presentable.



The last question I wanted to answer was if there was common language in the articles written about the different companies. I first saved all the articles into a variable, did a dropna, then converted it to one big string. I used regular expressions to get just the words, I removed unnecessary labels at the beginning, and the news source along with the (### chars +) at the end of each article. Another issue I ran into was the API didn’t give me the whole articles. It left of some of the article and told me how many characters the API doesn’t display, so my analysis doesn’t use the whole article. I then used Word2Vec to make model with a minimum count of one. From this I used the most similar method for each of the company names. If the name was more than one word, I split it up (Ex: ‘general’ ‘motors’). I then added the most similar words together and saved them to a variable. Lastly, I generated a word cloud from the most similar words. I did this by passing my data, and appended the first value of the most similar and created a string, that could then be visualized by a word cloud. Common words included Reuters, which is a news source and Amazon appeared as well. In addition, expectations and sales appeared often which makes sense. What I found interesting is that Donald appeared too, and I assume it refers to Donald Trump which means he is mentioned in the news often.



Attributions:

Sources are sited through out my code in the notebook. Also, in class notebooks and homework’s were referenced as well. The data is only being used for this class and this project specifically.

Source sites include: datacamp.com, geeksforgeeks.org, newapi.org, alphavantage.co, weirdgeek.com, stackoverflow.com

API documentation used:

https://newsapi.org/docs

https://www.alphavantage.co/documentation/