SI 618 Project 1 Report

Gurpreet Singh Bhatia
gsbhatia@umich.edu

Motivation:

Stock market is volatile and get affected very easily but rumours and news, very quickly. If a company is not doing well the volume of its shares traded can increase significantly. There is also quite evident change in the stock value. Stock market is based on beliefs. Traders bet on a company doing well or not doing well. To use the more financial terms of Bullish or Bearish. If people feel the company is going to do well, they buy its stock. If they don't, they sell. Whether it actually does well or goes bankrupt is part of the game, no one can really predict future. When a lot of people feel the company will do well, they buy its stock. What happens when a lot of people are buying the stocks of a company? It stock values increases. What happens when a bad news related to the company comes out it the morning right before the start of trading day? A lot of people who own the stock will see it, sending its value down.

In my project I try to map this relations where I what happens to stock market value of company when it was in news. As an added question, I also look at a significant major event alters the way in which newspaper talk about.

Data Sets:

I. NSE Listed 1000+ Companies Historical Dataset: This dataset contains the stock values from over 1000 publicaly traded companies in India on the NSE stock market. India has two stock markets, namely BSE and NSE. The dataset I have used relates to the National Stock Exchange (NSE) market. It contains, (1) list of all companies and their trading symbols and (2) a csv file for every company containing the following rows:
   A. Date: record date
   B. Open: opening value of stock for the date
   C. High: highest stock value went during the day
   D. Low: lowest stock value went during the day
   E. Close: Closing value
   F. Volume: Volume of shares trades

   https://www.kaggle.com/abhishekyana/nse-listed-1384-companies-data

Date ranges for most companies is between 2000-01-03 and 2019-05-15. I have chosen to use "Close" to reflect value for stock as it concretely reflects mood of the market with respect to the company. Volume on the other hand, can be quite misleading as a company may have a lot more shares in market. Hence, closing stock value is most concrete indicator. One file per company, and each such file contains close to 4800 rows.

II. News Headlines of India: This dataset contains 2.9 million news headlines in India as recorded by top journalist in the Times of India newspapers. Its from the year 2001 till 2018. It should contain, almost every big event happened during the time period covered. Times of India is a reputed news agency, with multiple publications across the country, hence achieves a lateral spread. Moreover, the news-headlines come from several different categories like entertainment, sports, and regional news particular to the city. The dataset contained one csv file with the following rows:

A. Publish date: YYYYmmDD format
B. Headline category: category of the headline
C. Headline Text: textual piece of headline

There are in total 1952 different categories of headlines presented in the dataset. A category is assigned to the dataset based on the sitemap. For my purpose I have ignored the categories.

Link :https://www.kaggle.com/therohk/india-headlines-news-dataset

Data Manipulation:

Both of my datasets are available from Kaggle from where I have downloaded them. They were present in CSV file format. One of the things I first did was to load and filter the data so as to reduce the computation time for different iterations. I did using pyspark, where I created a basic set of commands that converged exactly the required subset of data for me.

The script can be found in the file headline_mentions.py.

Headline dataset

It starts by removing the header from csv files and creating an RDD from it. The headline text data is in unicode format, that is converted back to normal python string for easy manipulation.

After loading it as an RDD, i converted it to a dataframe to apply column manipulation.

I converted the date, present again as unicode, in “yyyyMMdd” to datetime format “yyyy-mm-dd” format. This will be essential for filtering the dataset later on.

Spark-SQL query to filter for dates between 2008-01-01 and 2008-12-31. I have chosen to use data from the year 2008 for my purpose. This is essential for two reasons. (1) a lot of crucial events like global recession and 26-11 terrorist attack happened during this time. This helps to inherently reduce noise from the dataset. (2) Even after filtering the dataset contained >120,000 rows, giving me meaningful insights.

Stock Dataset:

Reading the dataset for company name, in csv format, as an RDD.

Removing non-essential index.

Dropping non-essenitial rows apart from the dataset.

Removing names of companies that match another commonly occuring words so as to not confuse it in the headline.

```
company_names.remove("d b reality")
company_names.remove("k s oils")
company_names.remove('w s industries (i)')
company_names.remove('premier')
company_names.remove('gati')
```

```
ignore_list = ['moil','d b', 'the great', 'c &','atul','arvind','new delhi','tamil nadu','the state']
```

Extract company code (code used on the trading platform) and company name for the dataset.

First join
Headline Text -   Company Name
Scan through all the headlines to see which companies name comes it in.
For company name, I had to remove “_” symbol present in the dataset.

Using company name, I get company symbol

Second Join
Company Name - Company Symbol
Use company symbol to access its stock values. The csv files for stock values are named using company symbol as opposed to the name, hence the requirement to join these two files in the same dataset.

Analysis and Visualization:

Research Question 1: Find out the list and number of times a company features in the headlines.
After my first join, where I have names of all companies in the dataset, I scan through each headline, using pyspark, and check whether the headline contains the company name or not. I had to shorten the company name used for comparison, or just take top 2 words as full name might not often be used in the headline.

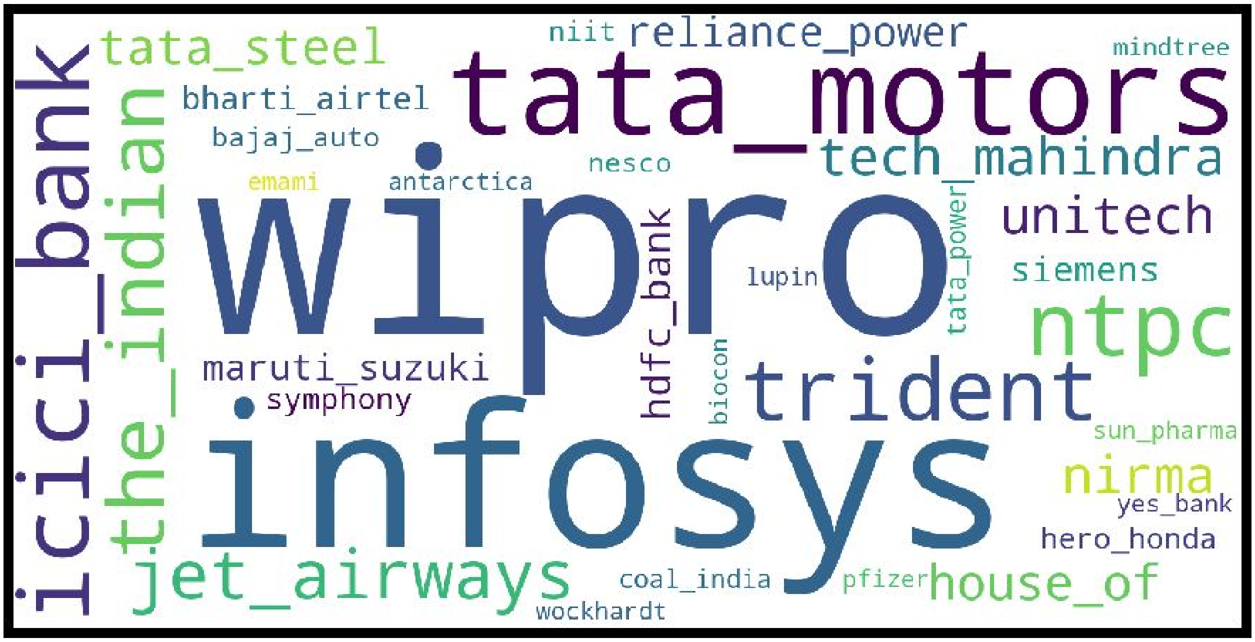
Code for this is mentioned in headline_mentioned.py and after joining, I run every line through the find_name_in_headline function. It returns (1,name_of_company) for every company there is match. For my simple case I assume just one company per headline.

Reducing Noise: had to perform some manual scrapping to remove some company names that gave me false positives.

Then I count the number of times each company was mentioned using reduceByKey function.

Results:

```
[('wipro', 126), ('the karnataka', 2), ('crisil', 4), ('emco', 3), ('jsw energy', 1), ('jsw steel', 1), ('apollo
hospitals', 1), ('transport corporation', 1), ('aurobindo pharma', 1), ('reliance industries', 1),
('dpsc', 1), ('the indian', 15), ('trent', 2), ('titan industries', 1), ('orient green', 1), ('apollo tyres', 2),
('tata communications', 3), ('reliance power', 15), ('spanco', 3), ('niit', 8), ('heritage foods', 1),
('hero honda', 9), ('aditya birla', 2), ('biocon', 6), ('software technology', 1), ('mphasis', 5),
('kingfisher airlines', 2), ('prime focus', 1), ('amara raja', 1), ('hsil', 5), ('hindustan unilever', 1),
('coal india', 7), ('wyeth', 1), ('texmaco', 2), ('geometric', 2), ('bosch', 2), ('antarctica', 6), ('oil
india', 3), ('the jammu', 1), ('jai corp', 1), ('beml', 1), ('utv software', 1), ('infosys', 72), ('balaji
telefilms', 4), ('lic housing', 1), ('cesc', 4), ('jet airways', 21), ('cipla', 4), ('core projects', 1),
('nmdc', 3), ('idea cellular', 1), ('pnb gilts', 1), ('emami', 6), ('atlanta', 2), ('bharti airtel', 11), ('natco
pharma', 1), ('fedders lloyd', 1), ('rajshree sugars', 1), ('nirma', 18), ('kpit cummins', 1), ('escorts',
4), ('ultratech cement', 1), ('archies', 1), ('dabur india', 1), ('pfizer', 6), ('hdfc bank', 13), ('hikal', 4),
('maruti suzuki', 14), ('godrej consumer', 1), ('hindustan motors', 1), ('wockhardt', 6), ('jk tyre', 1),
('max india', 1), ('marksans pharma', 1), ('mindtree', 6), ('irb infrastructure', 1), ('ashok leyland',
3), ('oil &', 3), ('bajaj auto', 8), ('house of', 16), ('essar oil', 3), ('tata teleservices', 3), ('jindal steel',
1), ('geodesic', 1), ('jindal saw', 2), ('ballarpur industries', 1), ('trident', 13), ('adani power', 1),
('power grid', 3), ('cox &', 2), ('tide water', 2), ('neyveli lignite', 1), ('sun pharma', 6), ('nesco', 8),
('blue chip', 5), ('nhpc', 4), ('tata power', 7), ('r systems', 1), ('pantaloon retail', 1), ('lupin', 6),
('mastek', 4), ('ifci', 4), ('bombay dyeing', 1), ('symphony', 8), ('tata steel', 18), ('ntpc', 29),
('bhushan steel', 1), ('tata elxsi', 2), ('west coast', 1), ('tvs motor', 2), ('bombay rayon', 1), ('yes
bank', 6), ('omaxe', 4), ('thomas cook', 2), ('indo tech', 2), ('reliance communications', 3),
('reliance capital', 1), ('jai balaji', 1), ('icra', 5), ('ashima', 1), ('icici bank', 32), ('siemens', 11),
('berger paints', 1), ('india infoline', 1), ('bharat forge', 1), ('idbi bank', 1), ('tata motors', 56), ('the
south', 5), ('the india', 2), ('patni computer', 1), ('unitech', 20), ('3i infotech', 4), ('marico', 2),
('indian oil', 4), ('kinetic motor', 2), ('alchemist', 1), ('punj lloyd', 2), ('tata chemicals', 1), ('cairn
india', 2), ('satyam computer', 1), ('merck', 4), ('jk paper', 1), ('eid parry', 1), ('ambuja cements',
1), ('tech mahindra', 19), ('tata coffee', 1), ('ceat', 1)]
```

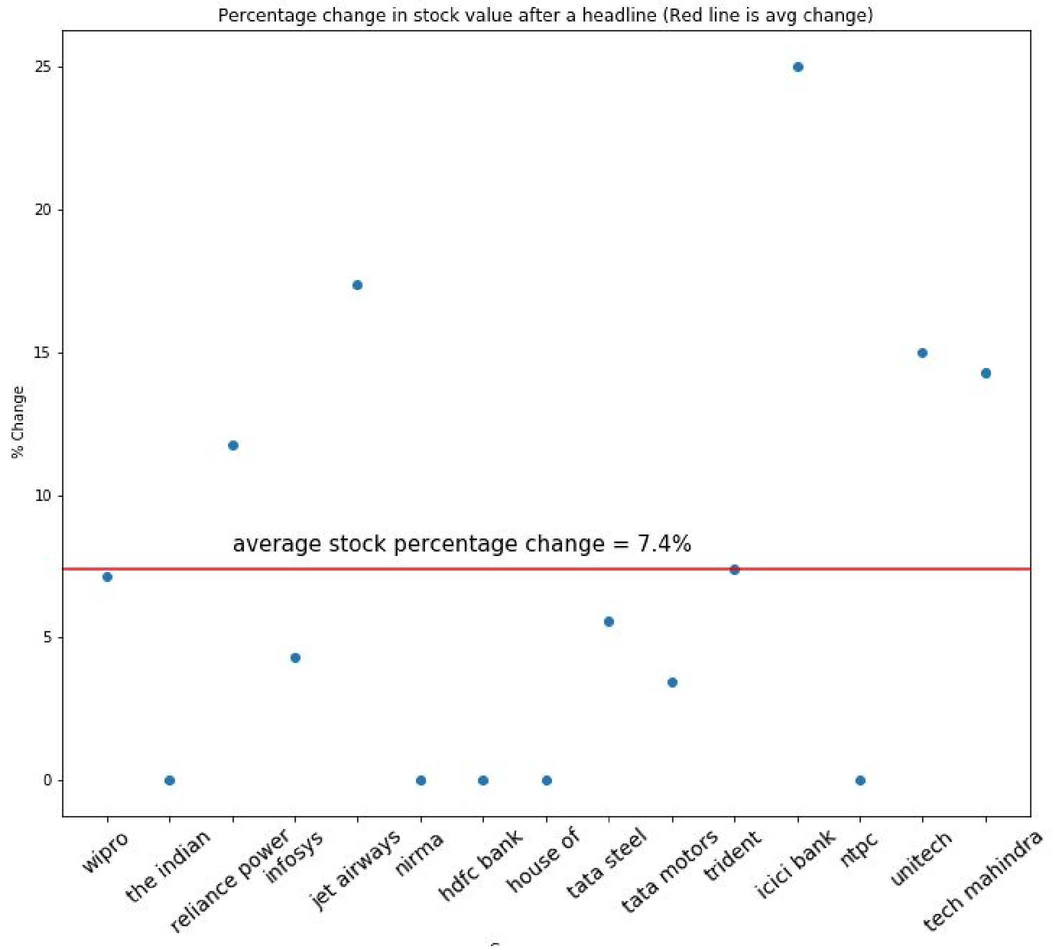


Research Question 2: Find out if the day a company was featured in a headline, its stock value changed in comparison to previous day

Code file: stock_change_with_headline.py and check_Stock_change.py
Using results from the previous RQ, I check for every date a company was mentioned in the news whether or not its stock value changed with respect to the previous day.

For this purpose, I pass every row (one for every company with all the headline dates) and open up the stock csv file using the company symbol. I perform similar data manipulation with company.csv as will company_list.csv. This included changing date format and removing non-essential rows. This initial filtering is done using pyspark and Spark-sql.

After this, i iterate for every date, and fetch the stock values for the headline day and previous day. Then I check if there was a change of more than 5% or not. Changes for all the companies are recorded and as either a zero or one, with one indicating significant change and zero indicating non-significant change.

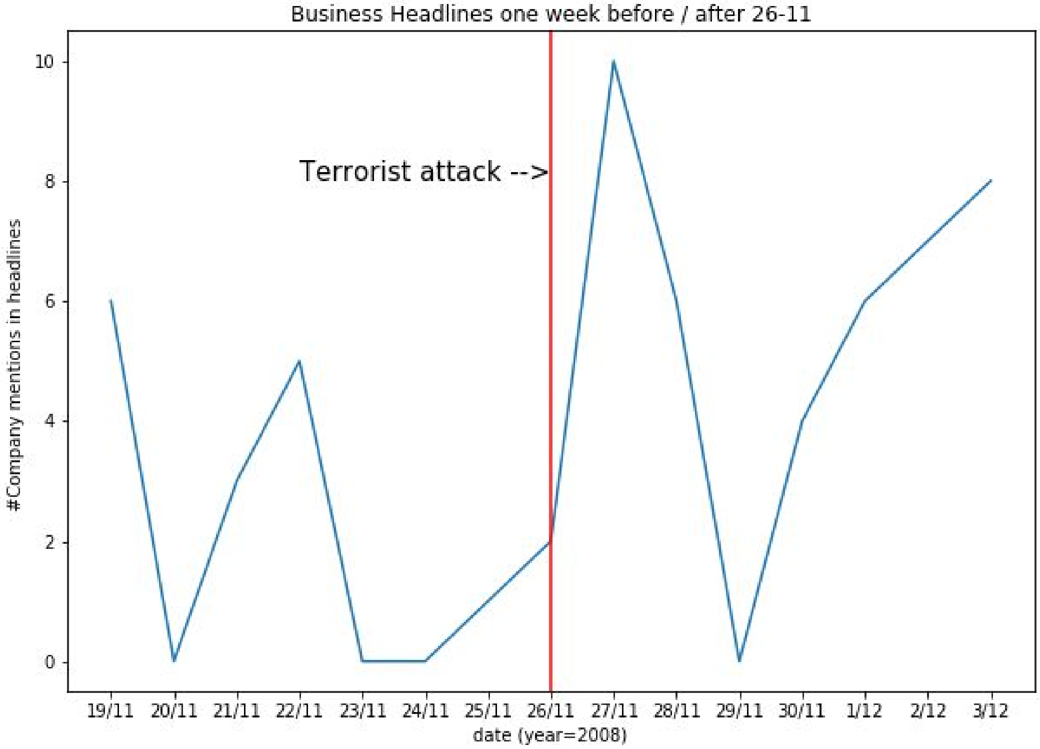


Please note, average stock percentage change records here whether or not percentage of change happening, given a headline for the company. Not the amount of stock change.

Research Question 3: Find out if a significant event shifts focus away from finance and business.

Do big events alter the way in which headlines change, specifically, do they shift focus away from other things. For this research question, I compared the number of times any company was mentioned week before and after the terrorist attack. One would hope that the focus shifts from finance and business to the main event happening.But in my data I observed an obnormality.

26-11 was India’s biggest terrorist attack but it did not shift focus.



Conclusions:

Challenges: Manual removing of noise in the company dataset when joined with headlines.