WEB ANALYTICS

FINAL-TERM REPORT, TEAM-12, BIA-660C

TEXT MINING, SENTIMENT ANALYSIS, AND DATA VISUALIZATION

"Analyzing Impact of News and Tweets on Share Prices"

SHALEEN MAMGAIN, SHREY KONNUR, SOURABH RAJPUT, SANJAY KUMAR PATTANAYAK

APRIL 27, 2018

“Analyzing Impact of Tweets on Share Prices”

OBJECTIVE:

Proposal:

Twitter has a lot of data which is unscraped and is useful if we separate the irrelevant data. There is a lot of discussion about companies and people on twitter, discussions that showcase the views, opinion and the value of the company. These discussions form a trend and impact the share prices of the companies. Therefore, Twitter trends have an impact on the stock market. Bloomberg is a global provider of financial news. Do we have any relationship between Bloomberg financial news and Twitter tweets? Does the Bloomberg news and Twitter tweet impact the share prices? Let’s try to find out the answers to the above questions in our project.

The main objective is to study the impact on the stock market by analyzing the Bloomberg news feed and Twitter tweets for a specific timeline. The focus is to scrape and extract value information from the tweets and news feed which will help provide recommendations to users.

Mid Term:

We started in the direction of the proposal and found that scrapping data from Bloomberg was getting tougher as historical stock data was not available to us. Instead of Bloomberg we started to look for alternatives and after much search, deliberation, and effort we came across New York Times as a possible alternative to scrape data for both News and Stock Prices. Twitter data was available.

Final-Term:

We performed 1. Classification using Naïve Bayes, and Support Vector Machine, 2. Clustering using K-Means Euclidean distance and K-Means Centroid, 3. Topic Modelling, and 4. Correlation on the news articles and individual company news to further analyze our work.

DATA SOURCE:

Proposal:

Data will be gathered by scraping the Bloomberg news feed as well as the tweets for a specific timeline using Python.

Mid-Term:

Both the News Data and Stock Data were scrapped from New York times.

Separate Python code were written for both these tasks. We used dynamic web scrapping to get the data.

We scrapped data for eight companies:

1. Google
2. Apple
3. Microsoft
4. Facebook
5. Amazon
6. Bank of America
7. Boeing
8. JP Morgan
9. Netflix
10. Yahoo

The Stock data of these companies were stored in 10 different csv files named after each stock.

Similarly, the news of the companies was stored in different csv files, again, named after each company.

The Twitter data was also collected separately.

Final-Term (Project Flow):

The data source remained the same for the final term. Basically, we worked on the already available scrapped and processed data from mid-term.

Now that we have the collected data and have done some analysis. We moved ahead and started with the following flow:

- Text Mining:
    - Web Scraping for News.
    - Web Scraping for Tweets.
    - Web Scraping for Stocks.
- Data Cleaning and Preprocessing
- Sentiment Analysis
- Tweets Analysis
- Trends/ Data Visualization
- Classification:
    - Naïve Bayes
    - Support Vector Machine (SVM)
- Clustering:
    - K means
- Topic Modeling
- Correlation

METHODOLOGY:

Proposal:

Our methodology will consist of finding correlation between tweets, news stories and variation in stock prices using Python, Text Mining, Sentiment Analysis and Data Visualization.

1. **Text Mining:** Here we focus on collecting data from twitter and Bloomberg news feed. We will use python packages to get the texts for further analysis. For example, Tweepy will be used to getting the text from Twitter.
2. **Sentiment Analysis:** In this section we use the text collected from Step 1 to further analyze the sentiments represented by the texts. These sentiments lead to decision making and can give a clear idea behind the variation in the stock price.

   We can follow the steps given below for the sentiment analysis:

   - We can start with identifying a set of companies that are most talked about on twitter in a specific time frame.
   - Look at the variation in the price of the stocks in the above stated time frame.
   - For the stocks with maximum variation look for the tweets and news feeds.
   - Divide the complete timeline according to quarters to see the variance in quarters and check words in the text (tweets and news feed)
   - Finally conclude what words and the people who have maximum effect on stock prices

3. **Data Visualization:** Lastly, we use data visualization to convey our work i.e we provide graphical presentation of the sentiment analysis to better represent our work. Visualization helps us to represent the data to relevant authority so that they can read through the data and understand the results, thereby enabling better for better decision making.

Mid-Term:

Upon collection of the data we proceeded as below:

1. Python code was written for getting the news. The news data contained date of news, and the news article.
2. Python code was written for getting the stock prices. The Opening, High, Low, Closing, and Volume of stocks traded were found on different dates.
3. Then Sentiment analysis on the news was done. Python code was written to get the sentiment analysis of each article. This helped us understand the sentiment of an article on the day of the news publication. Did the sentiment help or affect the price of the stock on that particular day? There were many questions that were crossing our minds, we did find answer for few, but many are still to be answered and hopefully as we proceed through our course we will gradually learn ways to answer our questions.
4. Upon completion of the above three coding, we proceeded towards combing all the stock data (News and Prices) with the Sentiment of the news.
5. The purpose of the above step was to analyze the impact of the news on the stock prices.
6. A lot of reference material were used in this project, so that we could proceed correctly.
7. Finally, we tried some visualization techniques to help explore the data available with us.

Tweets Data report:

Tweepy package was used to access tweets stream. OAuthHandler, Stream, StreamListener packages used for twitter authentication and Datetime package is used for handling day time. Next step is to set user or customer keys to access tweets. Now establish connection from customer keys.

Once the connection established then we need to keep track of time span of the customer by providing start time. Further we set time limit to get data from website.

Calculated running time and then checked if running time is over time limit, if time limit is exceeded then loop will break, we set time limit of 10 mins for retrieving data for Google, Bank of America, Netflix, Facebook, apple, Microsoft, Boeing, Amazon, JP Morgan and Yahoo tags from twitter.

For getting tweets of Google generated API authentication and gave max tweets limits to 1000. then collected all tweets sorted by time in descending order. Repeated this step for other data and stored data into csv file.

Web scrapping News:

We have used data from NY times, as they have articles available from 1850 onwards.

We first scrapped links from the website, from all the pages for our specified search of company, date and Business News.

From each page we got 10 links. These were added to a list and then each link was opened to get the actual news articles.

Stocks data was taken using package DataReader.

Google is the website being referenced to download the Stock quotes. Stock quotes have been taken from January onwards.

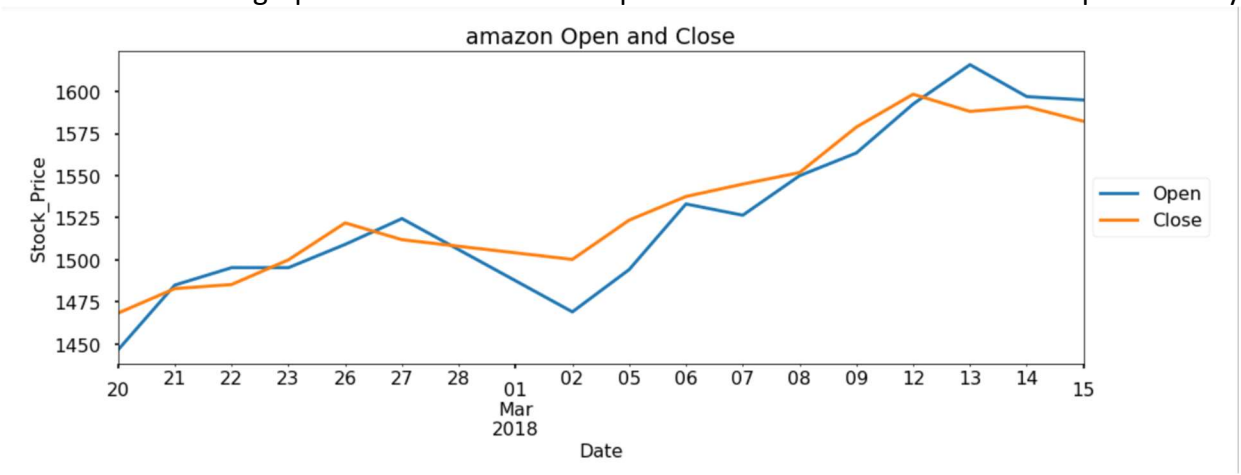
Data:

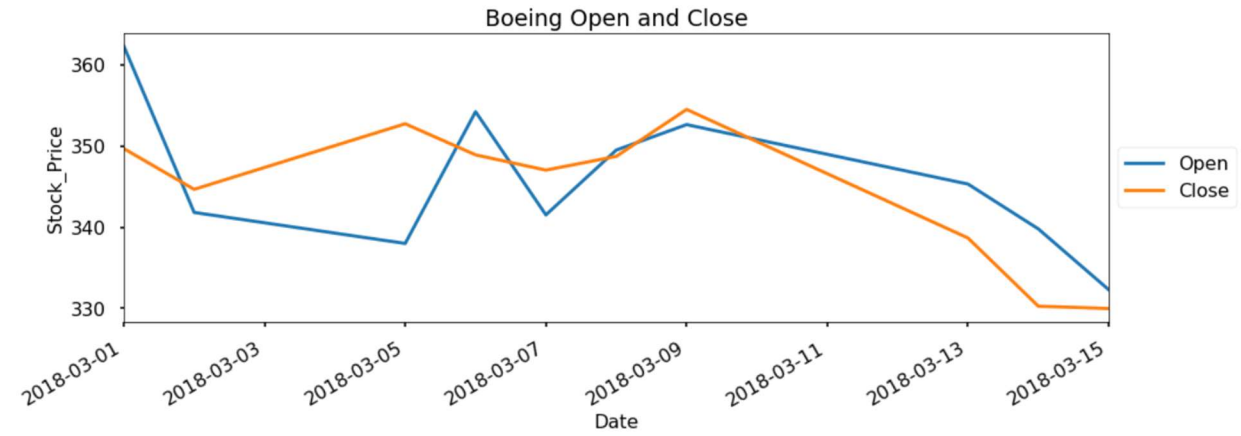
The news data is collated in the python and exported as data frame using Pandas. News data for all the companies is merged together. The same is followed for Stocks.

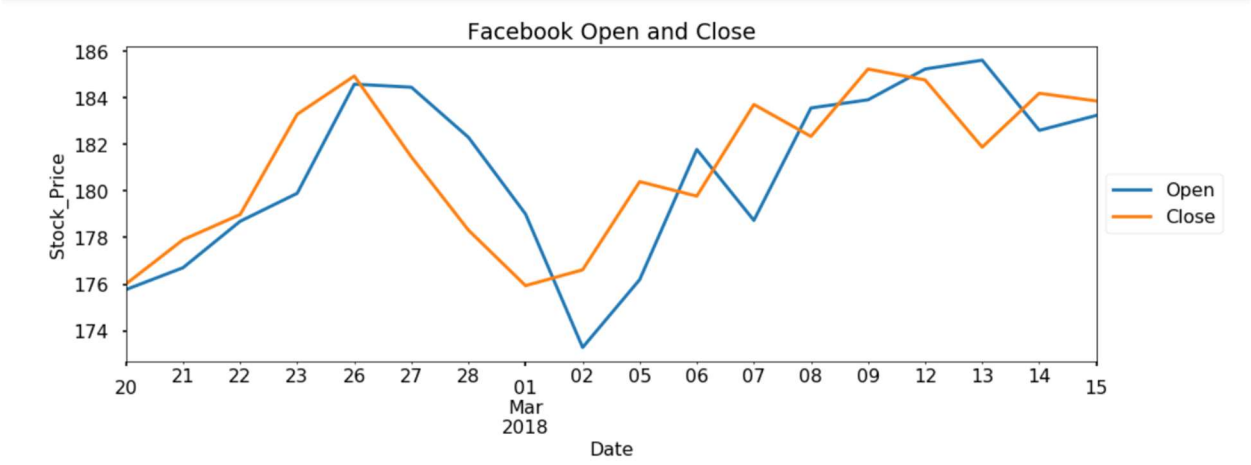
Using Inner join, we have merged the Stock and News Data. The columns used to merge this are Company Name and Date. This gave us a consolidated data set with news and stock data with data as reference.

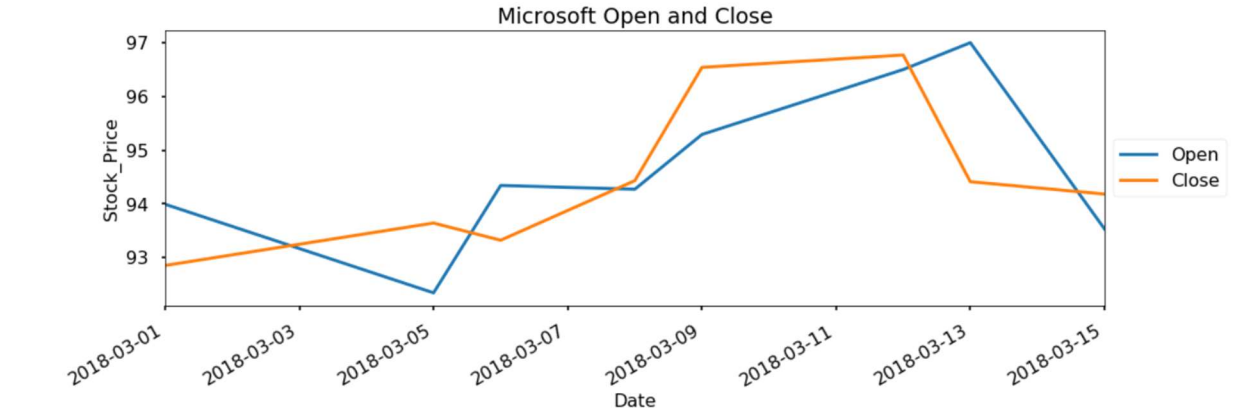
Sentiment Analysis of the news articles was added to this file. Plotting the graph using this dataset provides insights regarding the effect of news on the price of the stock. This is a basic approach and we will dive deeper in to this by using other different methods.

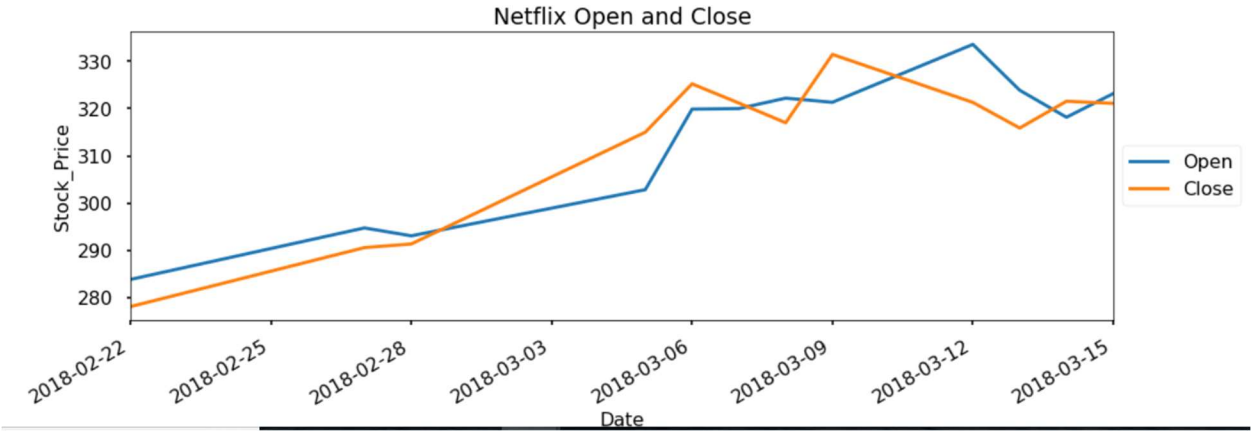
Below are the graphs that show the price of the stock on the specific day:



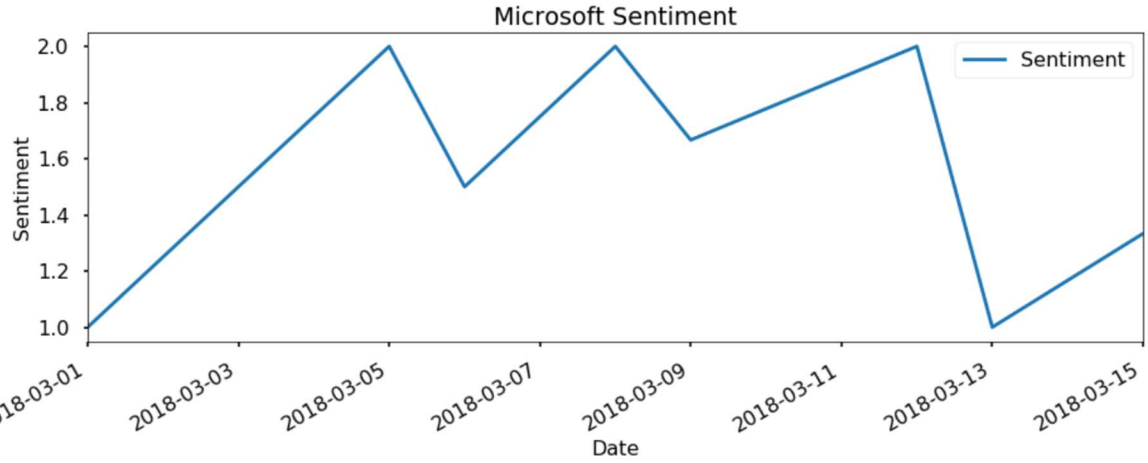


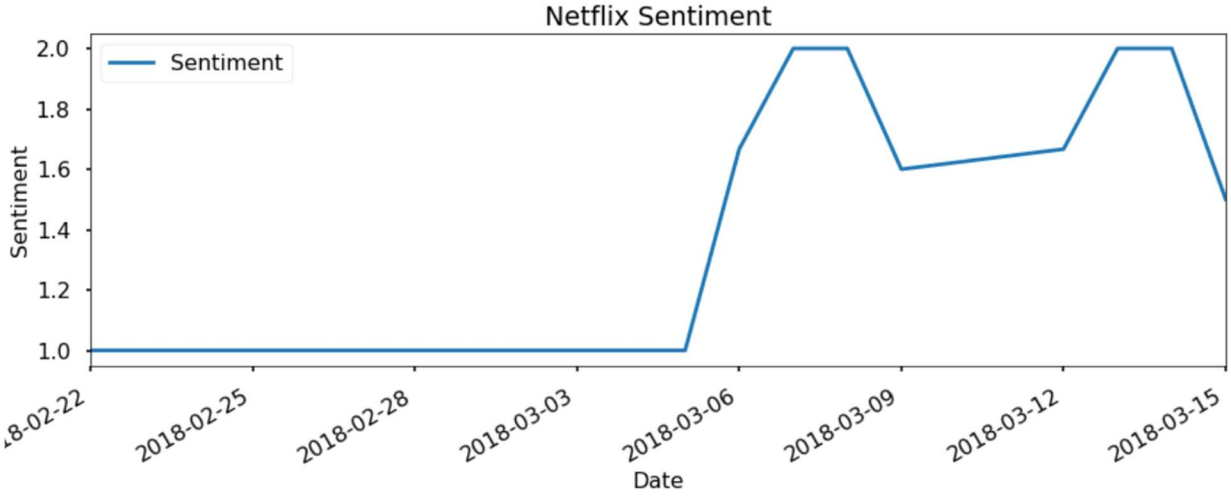


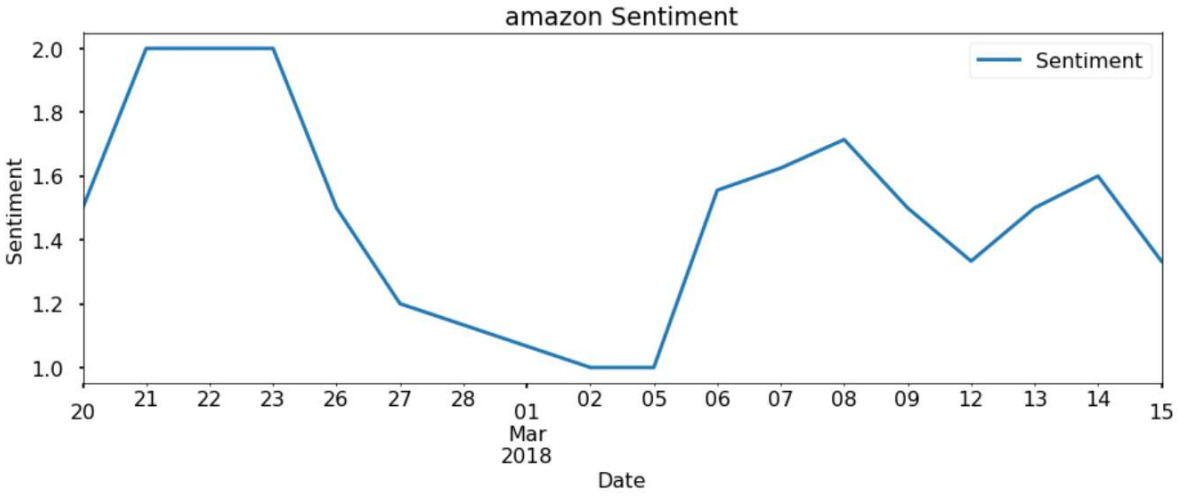


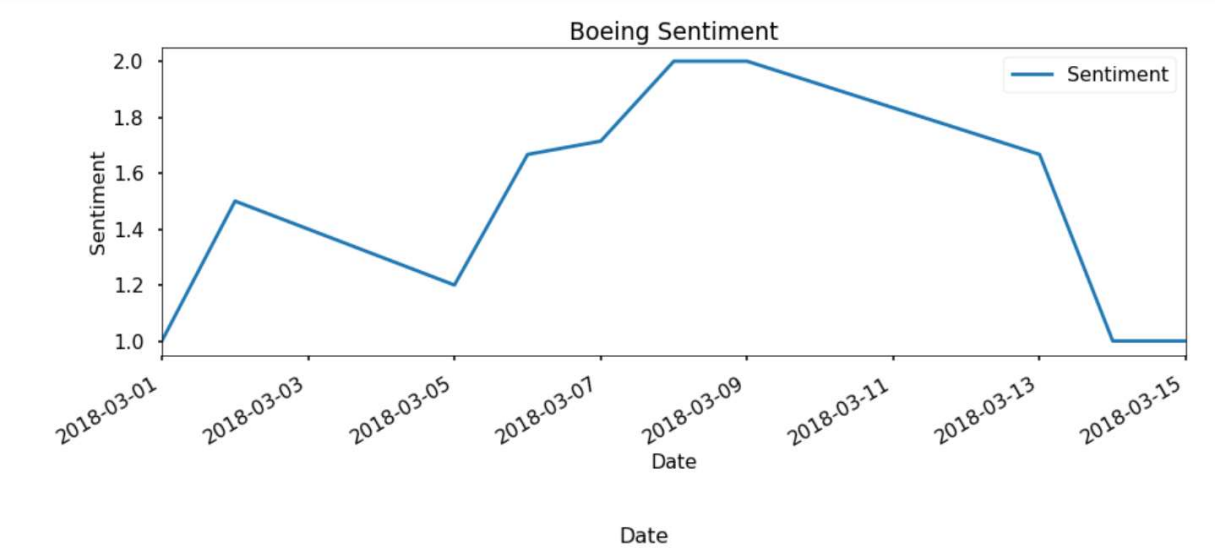


The date wise Sentiments of the companies according to our analysis is as follows:

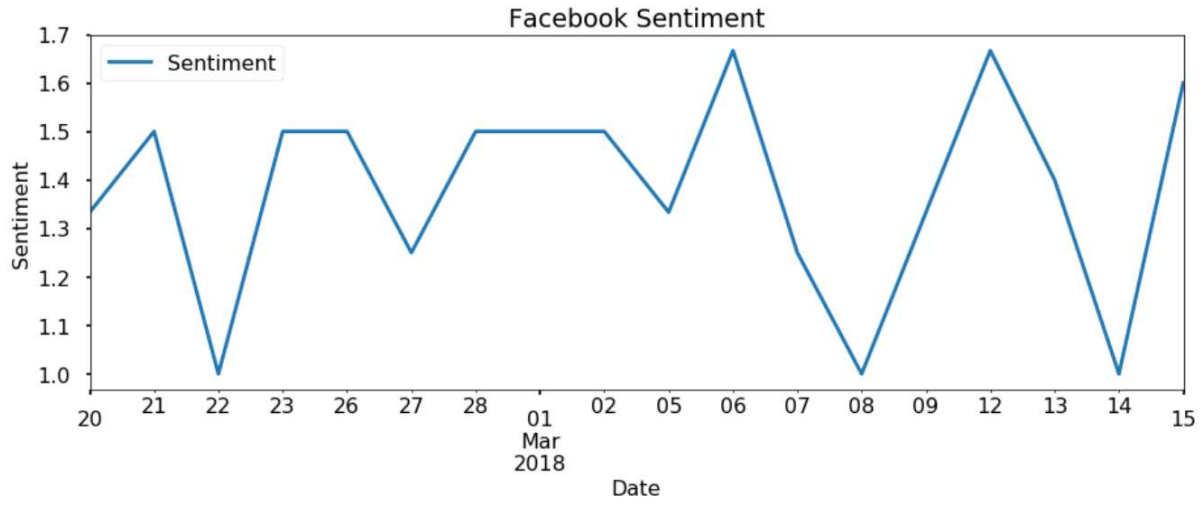

Microsoft Sentiment
Sentiment
2.0
1.8
1.6
1.4
1.2
1.0
Sentiment
2018-03-01
2018-03-03
2018-03-05
2018-03-07
2018-03-09
2018-03-11
2018-03-13
2018-03-15
Date


Netflix Sentiment
Sentiment
2.0
1.8
1.6
1.4
1.2
1.0
Sentiment
2018-02-25
2018-02-28
2018-03-03
2018-03-06
2018-03-09
2018-03-12
2018-03-15
Date


amazon Sentiment
Sentiment
2.0
1.8
1.6
1.4
1.2
1.0
Sentiment
20
21
22
23
26
27
28
01
Mar
2018
02
05
06
07
08
09
12
13
14
15
Date



Date



We can take example of Facebook, where we can see that there is a correlation between the sentiment and the stock prices. At some points it is very evident while at some it is faint.

Sentiment Analysis:

Sentiment Analysis is the process of determining the feeling that the text is conveying. We have analyzed the tweets and analyzed people's attitude based on the tweets. We have worked on the sentiments and categorized in positive and negative. Variations of the sentiments analyzed are considered useful to know about the customers and their choices.

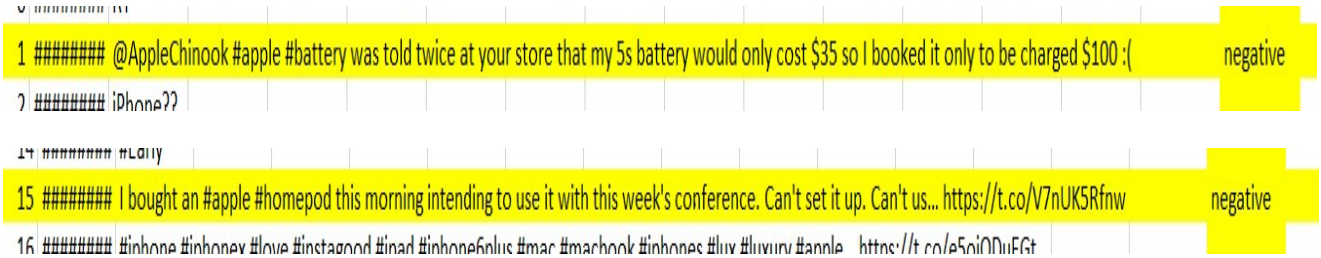
This kind of classification can help the companies to know about the areas of improvement.

We have done sentiment analysis manually on the tweets based on 10 companies: Amazon, Apple, Bank of America, Boeing, Facebook, Google, Microsoft, JPMorgan, Netflix, Yahoo.

Based on the classification, Please find the screenshots below :

1. Amazon:

2. Apple:



24 ######## Continuing on the current trend, Replacing your iPhone 6 battery! #Apple #iPhone https://t.co/pfhC5M84Da positive

39 ######## RT @lisabriercliffe: #Apple takes a very un-tech approach to solving fake news: human editors https://t.co/VGflaKv4rR #technology negative

80 ######## Sarahs Choice #book 1 in the #series get it at #APPLE #asmsg #spub #ian1 #iartg #scifi https://t.co/9/zsZf7Zak https://t.co/mVZ3Lb6816H

81 ######## I must say it sounds like an absolutely horrible working environment. I'd never want to work for a boss like that.... https://t.co/FPzWHseHiT negative

3. Bank of America:

17 ######## One of the reasons why I think @BankofAmerica is one of the best places to work is because we invest in our people!... https://t.co/Hz9GirYzym Positive

18 ######## #bankofamerica Your customer service is a joke. Negative

19 ######## when #bankofamerica cancels the wrong bank card and then won't fix the problem even after 30 min on the phone with... https://t.co/dqpTtxy6hH Negative

20 ######## #boa #bankofamerica did the right thing and dissolved my parents debt thank you Positive

23 ######## @BofA_Help If you want to lose your money, bank with BOA, they'll be glad to steal it! Oh they lie and get made whe... https://t.co/sN1zr7DJOb Positive

4. Boeing:

20 ######## A very strange feeling to sit in the pilot seat of one of the deadliest #airplane of the world: The #USAF #Boing... https://t.co/BLos6rXLom positive

53 ######## @realDonaldTrump Tell that to #Boing witch is loosing in the stock market big time today thanks to your reckless ac... https://t.co/ZnvhGtMdAt negative

5. Facebook:

10 ######## This is fundamentally the problem with #FB - bad actors can do essentially anything and FB will not police them. It... https://t.co/0GO6FT7ZDW negative

211 ######## RT @ArturTallada: Punt de recollida d'activitats per a la celebració del #pidayCAT #DiadePi Fes la teva aportació a https://t.co/5YIZHYIvt9...

212 ######## Please watch this: Cambridge Analytica whistleblower: 'We spent $1m harvesting millions of Facebook profiles' – vid... https://t.co/38HIJMMLN8 negative

213 ######## RT @AMCPMX: Anexo 15 Tarifa ISAN, claves vehiculares 2018. https://t.co/1sSdCsw1gM #FB #AMCPMX https://t.co/b7bBHtPSGi

220 ######## Yeah FB login just isn't worth the risk #fb https://t.co/U1j1dzTXH6 negative

258 ######## #Facebook's (aka #Fakebook) new algorithm for 'accurate' news feeds is intentionally killing off #Conservative site... https://t.co/TpUZE537Cy negative

905 ######## RT @KrauserOfIowa: #ChristopherWylie on #CambridgeAnalytica: "We exploited Facebook to harvest millions of people's profiles. And built mod...

906 ######## Facebook bans Cambridge Analytica #Yahoo: https://t.co/gFb5b9gX7W #FB.US #NASDAQ 100 Components #Stocks https://t.co/nC37LDWRWp positive

6. Google:

0 ######## These #Google employees used their '20 percent' time to improve Maps for people in wheelchairs Via https://t.co/0ai0jktjfl positive

1 ######## @sundarpichai @marioqueiroz @Google @GoogleIndia #Google please some one help me out with why I my #pixel is unusa... https://t.co/QzcfRyX3aX negative

7 ######## #Google "JNL": Kate McKinnon Destroys Betsy DeVos On Weekend Update "Watch" https://t.co/mkdd5XNWVL

8 ######## RT @Daily_Express: Google Chrome WARNING: Shock security risk could let hackers take control of YOUR computer https://t.co/zsGcHhH3kh #Goog... negative

379 ######## #Google Report: Abortion is safe but barriers reduce quality of care - WJLA https://t.co/0c9DKKA7bD negative

933 ######## Google can predict your risk of a heart attack by scanning your retina https://t.co/XUV3LUMW2x #google positive

924 ######## RT @abmysoftech: Happy Navratri ? #Navratri #Dadua #Ujjain #India #Seo #smo #digitalmarketing #internetmarketing #website #android #ios #app #

7. JPMorgan:

METHODOLOGY (Final – Term):

SENTIMENT MINING:

We used 2 methods for sentiment analysis.

1. Using Positive and Negative words list – Polarity Based
2. Using the NLTK – Vader Package – Valence Based

Positive and Negative words list: This is a polarity bases analysis, where the words are divided into positive and negative sentiments and the resulting sentiment is shown as the output. We have used this method and the resulting sentiment is called “Sentiment_Words” in our output. The graphs of the same is shown below.

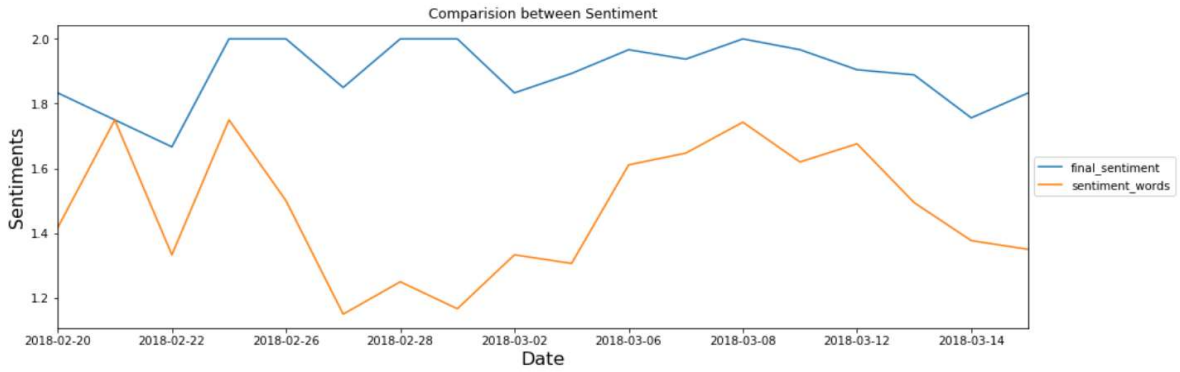
NLTK Vader: Vader stands for Valence Aware Dictionary and Sentiment Reasoner. This is a valence-based method, which means that it will give the Valence of words instead of the sentiment only.

Example: In the Polarity based method, good and excellent are classified as positive and have the same weightage. But in a Valence based method Excellent is given more weightage and is the score for excellent is more than that of good.

An example of the Vader scoring system is as given below:

Word	Sentiment rating
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

Below are some of the graphs we got in output:



The above graph is a comparison of Polarity based method and the Valence based method viz positive/negative words and VADER method plotted vs date

VADER = final_sentiment

Pos/Neg words = sentiment_words

We can see that the VADER method is giving higher positive sentiment than words.

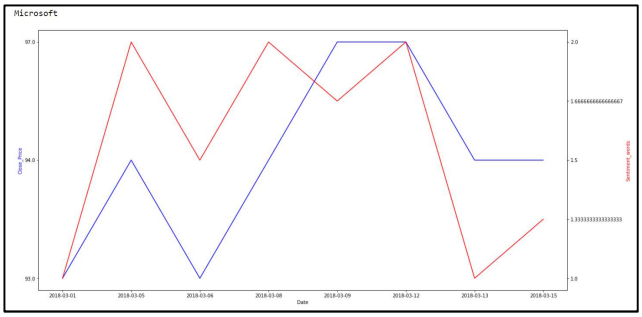
***Sentiment for both the methods is the average of the sentiment for that day for the above graphs. For individual companies, sentiment is the average sentiment for that day for specific company.**

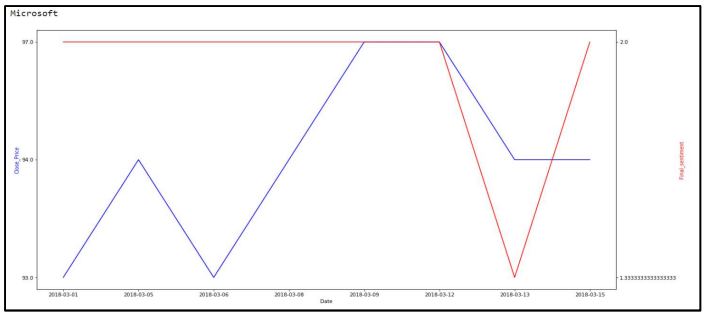
Below are the individual graphs for sentiments vs the close price.

Words – Polarity Based Sentiment

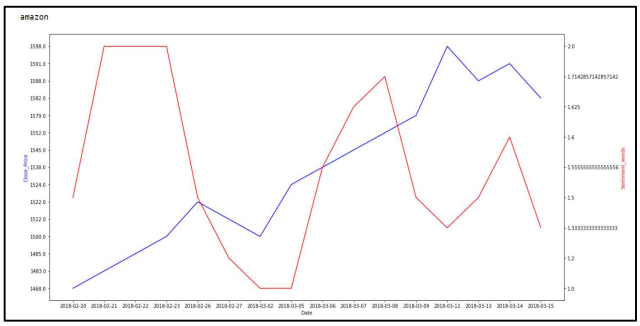
VADER – Valence Based Sentiment

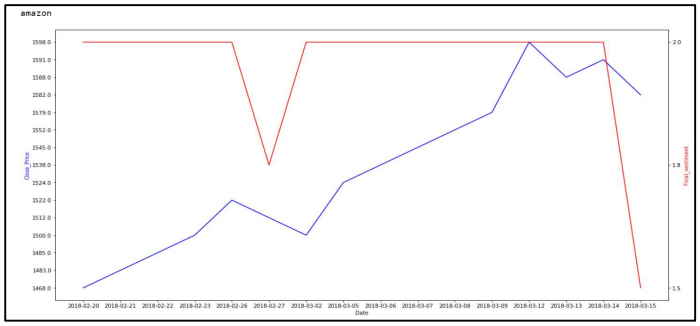
Microsoft



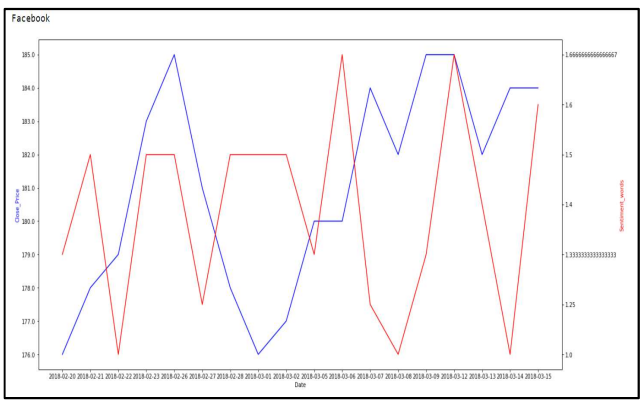


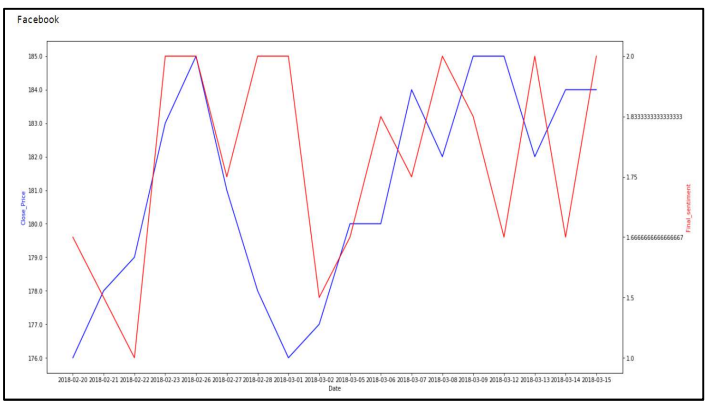
Amazon





Facebook





For Microsoft, the Pos/Neg graph was relating more to the close price compared to the sentiment plotted by NLTK package. But even the NLTK shows correlation in the second half of the graph.

For Amazon, the NLTK package does a better job in terms of correlation.

For Facebook, we can see that both have a good relation, but NLTK package is more accurate. However, we feel that this result is also because of the increased coverage Facebook has received in the recent time. Hence the graph has more variations with multiple sentiment shifts.

CLUSTERING:

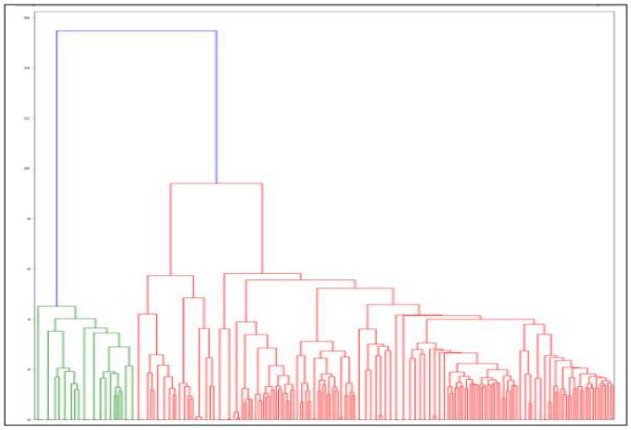
actual_class	reviews are negative	reviews are positive
cluster		
0	27	55
1	95	59

	precision	recall	f1-score	support
reviews are negative	0.62	0.78	0.69	122
reviews are positive	0.67	0.48	0.56	114
avg / total	0.64	0.64	0.63	236

```
tfidf_vect = TfidfVectorizer(stop_words="english",\
                             min_df=5)

# generate tfidf matrix
dtm= tfidf_vect.fit_transform(text)
print (dtm.shape)

num_clusters=2
clusterer = KMeansClusterer(num_clusters, \
                            cosine_distance, repeats=10)
clusters = clusterer.cluster(dtm.toarray(), \
                             assign_clusters=True)
```



actual_class	reviews are negative	reviews are positive
cluster		
0	27	55
1	95	59

	precision	recall	f1-score	support
reviews are negative	0.62	0.78	0.69	122
reviews are positive	0.67	0.48	0.56	114
avg / total	0.64	0.64	0.63	236

Precision is 64% with K-Means Clustering (Centroid) whereas in case of K means (Euclidean Distance) is 38%.

Hence, K means Clustering is better with Centroids than with Euclidean distance when all the news articles are considered.

Then we tried clustering with individual news for the companies. Here, we found that the results were different for individual company news. In some cases, both clustering methods gave similar results while in other cases difference in results were observed.

Clustering- Amazon

	precision	recall	f1-score	support
reviews are negative	0.58	0.71	0.64	35
reviews are positive	0.67	0.53	0.59	38
avg / total	0.63	0.62	0.61	73

K-Means with Euclidean Distance

	precision	recall	f1-score	support
reviews are negative	0.71	0.43	0.54	35
reviews are positive	0.62	0.84	0.71	38
avg / total	0.66	0.64	0.63	73

Clustering- Boeing

	precision	recall	f1-score	support
reviews are negative	0.84	0.80	0.82	20
reviews are positive	0.78	0.82	0.80	17
avg / total	0.81	0.81	0.81	37

K-Means with Euclidean Distance

	precision	recall	f1-score	support
reviews are negative	0.79	0.75	0.77	20
reviews are positive	0.72	0.76	0.74	17
avg / total	0.76	0.76	0.76	37

Clustering- Google

	precision	recall	f1-score	support
reviews are negative	0.64	0.50	0.56	14
reviews are positive	0.50	0.64	0.56	11
avg / total	0.58	0.56	0.56	25

K-Means with Euclidean Distance

	precision	recall	f1-score	support
reviews are negative	0.88	0.50	0.64	14
reviews are positive	0.59	0.91	0.71	11
avg / total	0.75	0.68	0.67	25

Clustering- Facebook

	precision	recall	f1-score	support
reviews are negative	0.68	0.76	0.71	33
reviews are positive	0.58	0.48	0.52	23
avg / total	0.64	0.64	0.64	56

K-Means with Euclidean Distance

	precision	recall	f1-score	support
reviews are negative	0.83	0.45	0.59	33
reviews are positive	0.53	0.87	0.66	23
avg / total	0.71	0.62	0.62	56

CLASSIFICATION:

We proceeded with classification of news articles using couple of algorithms:

1. Naïve Bayes
2. Support Vector Machine

Naïve Bayes:

	precision	recall	f1-score	support
1	0.86	0.84	0.85	43
2	0.77	0.79	0.78	29
avg / total	0.82	0.82	0.82	72

```
X_train, X_test, y_train, y_test = train_test_split(\
                 dtm, sent, test_size=0.3, random_state=0)
# train a multinomial naive Bayes model using the testing data
clf = MultinomialNB().fit(X_train, y_train)

predicted=clf.predict(X_test)

labels=sorted(list(set(sent)))

precision, recall, fscore, support=\
     precision_recall_fscore_support(\
     y_test, predicted, labels=labels)
```

SVM (Support Vector Machine):

	precision	recall	f1-score	support
1	0.87	0.79	0.83	43
2	0.73	0.83	0.77	29
avg / total	0.81	0.81	0.81	72

```
Test data set average precision:
[ 0.57166667  0.77097902  0.74456522  0.82358871  0.87884615]

Test data set average recall:
[ 0.58434783  0.76956522  0.74456522  0.79076087  0.87310606]

Test data set average fscore:
[ 0.57738095  0.76993464  0.74456522  0.78240741  0.86931818]
```

Comparing these two Multinomial Naïve Bayes and SVM algorithms we found that both give almost equal results that is precision of about 82%.

We also extended these algorithms to individual company news articles and observed that some results are similar while others have contrasting results. These can be attributed to the number of news articles and coverage received by a company.

SVM/ Naïve Bayes – Amazon

	precision	recall	f1-score	support
reviews are negative	0.67	0.73	0.70	11
reviews are positive	0.70	0.64	0.67	11
avg / total	0.68	0.68	0.68	22

SVM Result

	precision	recall	f1-score	support
reviews are negative	0.67	0.73	0.70	11
reviews are positive	0.70	0.64	0.67	11
avg / total	0.68	0.68	0.68	22

SVM/ Naïve Bayes: Facebook

	precision	recall	f1-score	support
reviews are negative	0.80	0.92	0.86	13
reviews are positive	0.50	0.25	0.33	4
avg / total	0.73	0.76	0.73	17

SVM Result

	precision	recall	f1-score	support
reviews are negative	0.89	0.62	0.73	13
reviews are positive	0.38	0.75	0.50	4
avg / total	0.77	0.65	0.67	17

SVM/ Naïve Bayes: Boeing

	precision	recall	f1-score	support
reviews are negative	0.67	1.00	0.80	6
reviews are positive	1.00	0.50	0.67	6
avg / total	0.83	0.75	0.73	12

SVM Result

	precision	recall	f1-score	support
reviews are negative	0.83	0.83	0.83	6
reviews are positive	0.83	0.83	0.83	6
avg / total	0.83	0.83	0.83	12

TOPIC MODELING

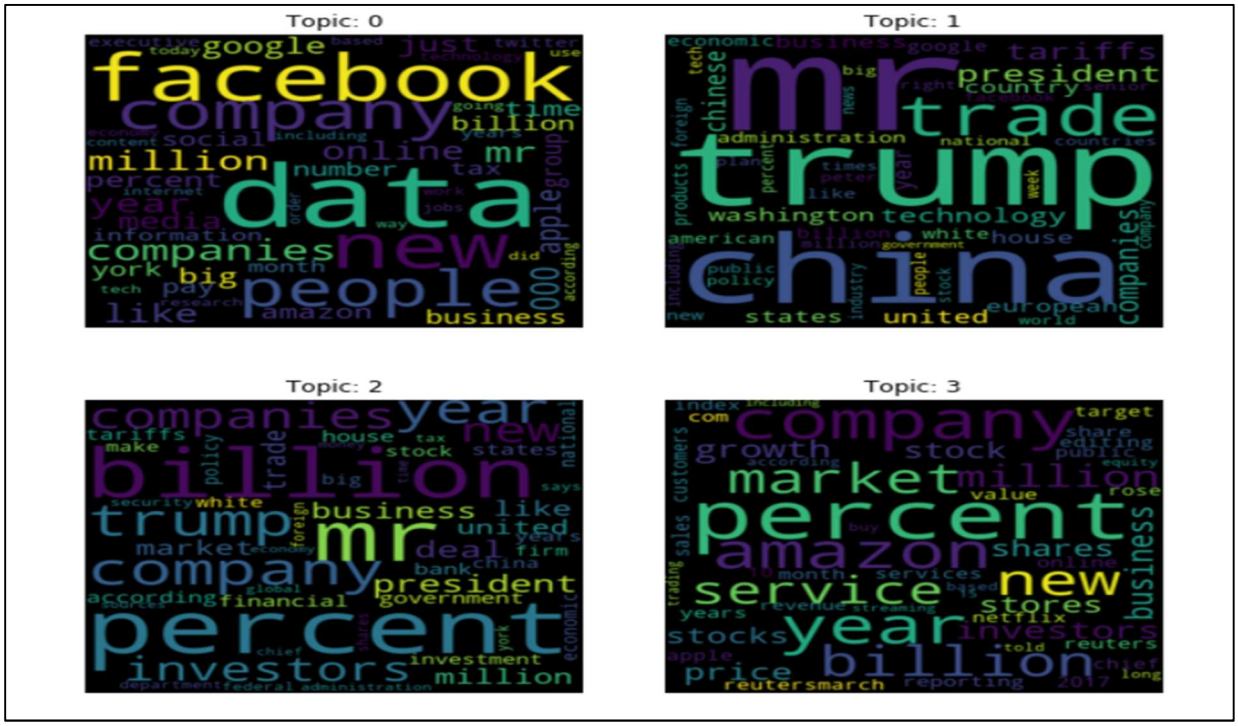
We proceeded with topic modelling to see what news article headlines can contribute to sentiments. We started with all the news articles and tried to cluster into top four clusters and get their top twenty words that can combine to give a topic. We observed all the topics had almost similar words implying news articles of companies had similar contents.

```
Topic 0:
[('data', 210.50787831832832), ('facebook', 201.42412237016003), ('new', 186.87733311128261), ('company', 177.18955189205985),
('people', 168.23111178641912), ('companies', 148.41969355005526), ('mr', 141.24296837324744), ('year', 120.38302173331216),
('million', 118.77572358710887), ('000', 109.99559416984143), ('like', 107.32890374034005), ('online', 104.35152328020283), ('g
oogle', 81.091030504644095), ('media', 80.649561148938119), ('apple', 80.336044786405509), ('just', 77.134094349250603), ('perc
ent', 74.556242277016821), ('big', 72.794335790078335), ('time', 71.741202946671137), ('amazon', 63.934936952503008)]


Topic 1:
[('mr', 224.89872297908997), ('trump', 213.39490800185126), ('china', 175.90479596513231), ('trade', 139.50517936764453), ('pre
sident', 104.02234208590991), ('tariffs', 98.4207425150435 85), ('companies', 85.386627605006595), ('chinese', 71.45363030276394
1), ('technology', 70.733248749648979), ('united', 62.08598467541416), ('washington', 57.669283074896313), ('house', 55.5089982
37653664), ('states', 55.105820647178817), ('business', 53.692490454866864), ('year', 51.804793244280852), ('country', 51.14548
6885517037), ('european', 50.563237945300408), ('administration', 49.218668392479444), ('american', 44.881699377859327), ('goog
le', 44.450622062023122)]


Topic 2:
[('percent', 373.29409527065314), ('mr', 338.36195100227707), ('billion', 329.68645911043836), ('company', 327.130733846644 09),
('trump', 314.47194147720575), ('year', 278.11630213598363), ('companies', 273.60314121810592), ('new', 250.88158459859935),
('investors', 234.01131255616664), ('president', 226.34725951147601), ('deal', 219.28522441800305), ('business', 183.2011679344
034), ('million', 166.86727178853502), ('like', 165.41587742462451), ('trade', 159.85717240061962), ('market', 159.307987377818
48), ('united', 153.5805871917685), ('government', 152.69311943854581), ('financial', 150.82467650938185), ('states', 150.79841
61887143)]


Topic 3:
[('percent', 364.11037418363941), ('company', 217.15963495889082), ('year', 202.25612365907372), ('amazon', 190.8955193483313
1), ('new', 147.56279594505327), ('billion', 137.40347325154565), ('market', 131.39683730795949), ('service', 112.5266275927599
9), ('million', 112.13956654859581), ('investors', 110.84875937078793), ('growth', 110.65815017420535), ('business', 110.022435
16158985), ('price', 97.553303174793882), ('stores', 91.541165554421966), ('shares', 90.772897034908027), ('stock', 90.01687975
4290571), ('stocks', 82.406723281110303), ('reporting', 79.504412238063864), ('years', 79.203975628838663), ('com', 75.95215060
9109765)]
```



To analyze further we took negative and positive sentiment news articles and tried to find the topics. Here, we could figure difference in the topics for some companies which can be segregated efficiently can define news headlines for positive and negative sentiments.

Microsoft Negative

Microsoft Positive

Netflix Negative

Netflix Positive

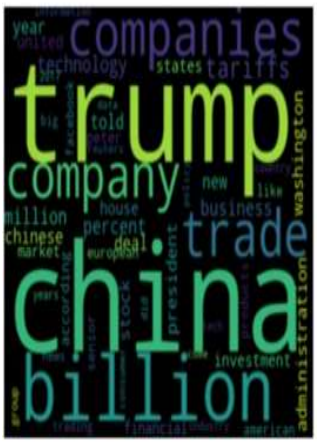
Amazon Negative

Amazon Positive

Boeing Negative

Boeing Positive

Google Negative



Google Positive

Facebook Negative

Facebook Positive

CORRELATION:

Correlation shows a mutual relation between one or more variables. Here we have used the Pearson method for correlation. Pearson's coefficient has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation.

Below are the outputs for the correlation functions that we have defined.

Corellation for Boeing

	open_price	close	high	low	sentiment_words	positive	Neg	final_sentiment
open_price	1.000000	0.482029	0.764558	0.759419	0.206669	0.235318	-0.220465	0.375581
close	0.482029	1.000000	0.850233	0.875109	0.356096	0.139961	-0.527525	0.503553
high	0.764558	0.850233	1.000000	0.837437	0.170904	0.241061	-0.367561	0.435389
low	0.759419	0.875109	0.837437	1.000000	0.476676	0.175882	-0.585598	0.587740
sentiment_words	0.206669	0.356096	0.170904	0.476676	1.000000	0.002616	-0.588887	0.364434
positive	0.235318	0.139961	0.241061	0.175882	0.002616	1.000000	0.008607	0.533727
Neg	-0.220465	-0.527525	-0.367561	-0.585598	-0.588887	0.008607	1.000000	-0.586999
final_sentiment	0.375581	0.503553	0.435389	0.587740	0.364434	0.533727	-0.586999	1.000000

Corellation for Facebook

	open_price	close	high	low	sentiment_words	positive	Neg	final_sentiment
open_price	1.000000	0.653272	0.895799	0.892316	0.019551	0.023352	-0.238966	0.251410
close	0.653272	1.000000	0.845890	0.872902	-0.036623	0.052980	-0.067298	0.132772
high	0.895799	0.845890	1.000000	0.906610	-0.028072	0.011617	-0.140369	0.216398
low	0.892316	0.872902	0.906610	1.000000	0.002555	0.050968	-0.145860	0.178249
sentiment_words	0.019551	-0.036623	-0.028072	0.002555	1.000000	0.179664	-0.558750	0.321398
positive	0.023352	0.052980	0.011617	0.050968	0.179664	1.000000	-0.158105	0.272358
Neg	-0.238966	-0.067298	-0.140369	-0.145860	-0.558750	-0.158105	1.000000	-0.720486
final_sentiment	0.251410	0.132772	0.216398	0.178249	0.321398	0.272358	-0.720486	1.000000

From the above correlation details, we see that there is a correlation between open, close price and final sentiment

For Boeing the correlation between final_sentiment and open is .37 and final_sentiment and close is .50, this is a good result since the data extracted by us is business news and there are many other factors affecting the price of stock.

For example: Boeing stock apart from being affected by business news, will also be affected by fluctuations in oil prices. Foreign trade policies of a country and environmental will also effect the stock price of Boeing, which are covered under International and Environmental news respectively.

FUTURE SCOPE AND CHALLENGES:

We believe that the availability of data is a major challenge in any data science project. We would prefer more access to news and tweets to analyze the results properly and find proper classification, clustering, and correlation among the new/tweets and share prices.
The availability of data will enable us to try to predict share price fluctuation, which we wanted to do in this project. With the available data we could see that the algorithms give better results when the volume of data is high and in those cases the results could be compared in a better way.

CONCLUSION:

We would here by conclude that there is a definite correlation among the news articles and tweets on the share prices. Though there are other contributing factors which can not be ignored, but, we can affirmatively say that if we properly analyze the news articles and tweets we can for sure predict with good precision the trend of the share prices in a day. Statistical algorithms along with web-scrapping technologies have enabled unforeseen abilities.

We thank Prof. Rong Liu and TA Yutian Zhou for all their help and good work through out this semester and wish them all the best for their future endeavors. It was a wonderful experience and association. Thank you for everything you did for us.

Reference:

http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

**********THANK YOU**********