University of South Florida

Data Science Programming

FINAL PROJECT ON

«NEWS SENTIMENT ANALYSIS»

Team members

Motahareh Pourbehzadi (Code-Video-Paper)

Devesh Tomar (Code-Video-Paper)

Sandeep Kumar (Code-Video-Paper)

Sivaramakrishna Allam (Code-Video-Paper)

Andrei Pomorov (Code-Video-Paper)

Tampa, Fl

2021

Contents

INTRODUCTION

Sentiment analysis is a branch of natural language processing (NLP) which makes our ML model capable of extracting attitudes out of sentences. In this project, we have worked on finding a correlation between the stock price of the Facebook Company following the consequences of the Facebook outage on Oct 4th, 2021, and the news sentiments. In order to realize this objective, we first identified a suitable web API (NEWS API) and collected top 100 headlines that corresponded to predefined keywords that reflected upon this incident from several resources. Afterward, the data went through cleaning and pre-processing to increase productivity. Next, we extracted the sentiments of the data using the natural language toolkit (NLTK). The sentiment analysis consists of word frequency distribution and Zipf's law, word cloud, and bigram analysis. Finally, we extracted the stock prices through the world trading data API and checked the correlation of the news sentiments with the stock prices.

TASK 1. IDENTIFICATION OF A SUITABLE WEB API

The main objective of our project was to find out how sentiments appeared in articles on October 4th reflected the consequences of the Facebook outage in terms of their market share. That being said, a good practice in machine learning is to first solve a more general problem of sentiment analysis and explore how do articles from different resources (i.e. journals, blogs and etc.) reflect the nature of stock market. Therefore, we shall pound a question of how news sentiments affect stock prices and we will try to analyze it. Then, we will specifically consider how those sentiments affected Facebook stock prices. Obviously, people started selling their shares and we anticipate to observe the decrease of stock prices what indicates that the company is going down. For this purpose, we've decided to aggregate news articles from NewsAPI (this is API that allows us to collect articles all over the WEB in real time). To access these articles, we need an API key which can be retrieved within the same website however with a limitation of articles being collected one month prior to the current date at most. Once we've collected the data, we start processing it by transforming the raw data into the pandas data frame which will further allow us to manipulate it in a straightforward manner. Function get_articles return a list which we will use to create a data frame.

TASK 2. DATA COLLECTION

Apart from the limitation on timeframe, the API is only capable of returning 100 news articles in API call. To aggregate more data, we have retrieved 100 articles per each publisher and divided the whole process into two stages. As we've made sure that the function works fine (which we did by collecting 100 articles about Facebook through the json() method of the response interface and transforming this into a pandas data frame), we went over to creating a list of different resources (domains list according to the code) where we were going to fetch the data from. Again, here we use json() method to make an API call to fetch the latest data from those sources.

TASK 3. CLEANING AND PRE-PROCESSING THE DATA

IMPORTANCE

In order to increase your productivity, you need to clean your data because it improves the overall quality of your data. All outdated or incorrect information will be removed when you clean your data, leaving only the highest quality information. Your team will not be forced to dig through outdated documents, allowing them to utilize their work time more efficiently. The text can be noisy with emojis, punctuations, or different case types. There is no benefit to machines from all of these noises, so cleaning is needed. We might be unable to effectively analyze raw text data if it contains unwanted or unimportant text, making our results difficult to understand and analyze.

We all know that we can apply math and statistics to numerical data in order to gain insights. However, when it comes to the tedious form of textual data, we lack it in many places. Language is a structured medium which we as humans being use to communicate with each other. The medium can be either written or spoken. We can understand words such as “Hello”, “Goodbye”, “Sorry” and etc., but the question is: “Can computers understand them?”. Unfortunately, the answer is, obviously, negative. In fact, machines cannot understand any text data at all, be it the word “yes” or the word “machine”. They only understand numbers. Over the decades, scientists have studied how machines can understand our language. Hence, we pre-process our data before feeding it to algorithms. Text pre-processing is a method to clean the text in order to make it ready to feed to models. The results also vary depending on how the data is pre-processed. Pre-processing is therefore the most important task in NLP. It helps us remove all the unimportant things from our data and make our data ready for further processing.

The python library which we will use for text pre-processing is Natural Language Toolkit (NLTK). It is a powerful tool full of different Python modules and libraries to carry out simple to complex natural language processing (NLP). NLP libraries translate between machines (like Alexa, Siri, or Google Assistant) and humans in such a way that the machines have the appropriate response. NLTK has a large, structured text known as a corpus that contains machine-readable text files in a directory produced for NLP tasks.

CLEANING PROCESS

In this project we used various steps for cleaning the data:

- We used “source_getter” function to extract the name of the source of the news article and exclude other details.
- Then, we converted the publication date to date time format for future analysis.
- After that we looked for missing data, and we observed that there are quite a few missing values present in the dataset and since the dataset is purely textual in nature and given the types of fields having the missing value, it is not possible to fill the missing values in this case. Hence, we decided to drop all the rows with null values.

- At last, we combined the “title” and “content” columns because they provided significant detail regarding the story and contained some keywords which were essential for further analysis.

PRE-PROCESSING

In the following sections, some of the text pre-processing steps were applied to the data. The steps included:

- Tokenization: It is like splitting a whole sentence into words. A simple separator can be considered for this purpose. Challenges increase when more languages are included. Further, it can be used to convert text into numeric data that can be absorbed by machine learning models.
- Removing the non-ASCII characters from the text.
- Stop words removal: Social media uses English as one of the most common languages. For instance, "a", "our", "for", "in" and etc. are in the set of most commonly used words. By eliminating these words, the model can focus on key features. These words also don't carry much information.
- Removing punctuations, apostrophe, special characters etc.
- Lemmatize the text: A lemmatized word retains only its base, or dictionary form, and removes all inflectional endings, which is known as the lemma.

TASK 4. SENTIMENT ANALYSIS OF NEWS ARTICLES FOR ANSWERING THE ABOVE QUESTIONS

We still have to figure out how we will do the sentiment analysis after we have the dataset ready. For this we used the following process:

- Calculating the polarity of the news articles. We import “SentimentIntensityAnalyzer” from “nltk.sentiment.vader” library for calculating the polarity of the news.
- Creating a new data frame of only the polarity score, the headline and the source of the news.
- Categorize news as positive or negative based on the obtained compound score. We have considered the news as positive if the compound score is greater than 0.2 and if the compound score is below 0.2 then it is considered negative, hence the label as 1 and -1 respectively.
- We will also count the number of words in news headline.
- Grouping the news articles by their source and calculating their mean polarity.

VISUALIZATION

In this section the results of implementing sentiment analysis on the Facebook-related news is demonstrated (Fig. 1). Here, the first step is to analyze the overall distribution of negative, neutral and positive sentiments. The results indicate that the negative, neutral and positive sentiments fractions are 40%, 27% and 33%, which indicates that at least the news resources that were chosen for this study were slightly leaning

towards negative news. The same result is seen in the univariate polarity distribution diagram, as there is a considerable fraction of the news that has neutral polarity, but the negative polarity seems to have more weight. We might often think that only negative news make it to the headline, whereas that is certainly not the case. It is our brain that is susceptible to this negativity bias, wherein we forget the positive side of the story and focus only on the negative side of it.

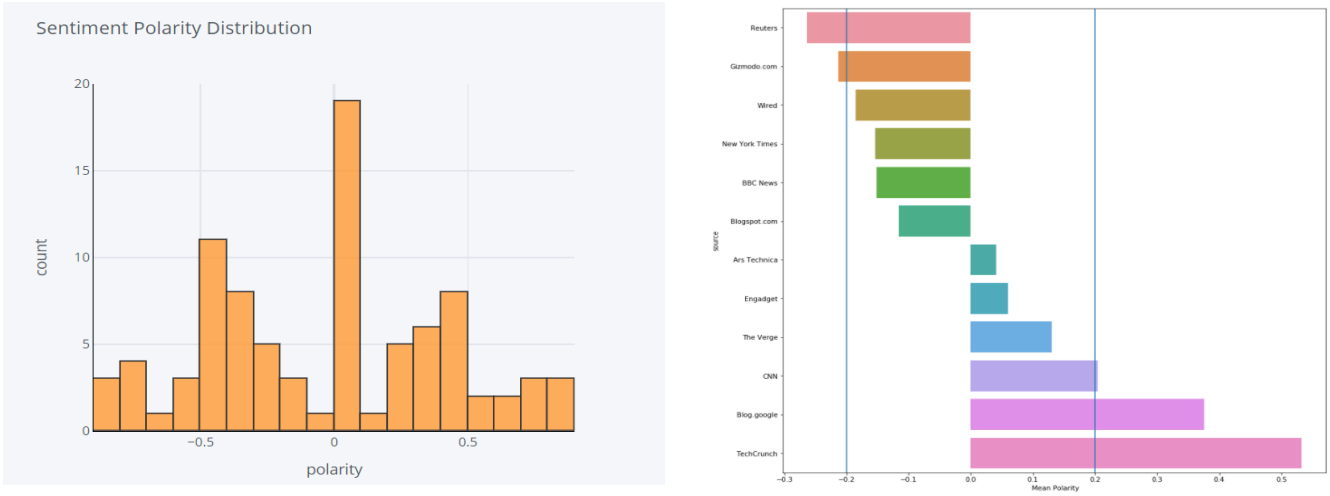

Fig. 1. Sentiment polarity distribution (on the left) and univariate polarity distribution diagram (on the right)

The mean polarity of a selected group of news publishers from around the world is shown in Fig. 2. The two vertical lines act a baseline to categorize whether the overall sentiment of the news is positive or negative. We can see that Reuters, Gizmodo and Wired have negative polarity, which means these publishers generally tend to cover stories that are negative in their sentiments. On the other side of the spectrum, we have publishers such as TechCrunch that often focus on the technology news and has considerably higher positive sentiments.

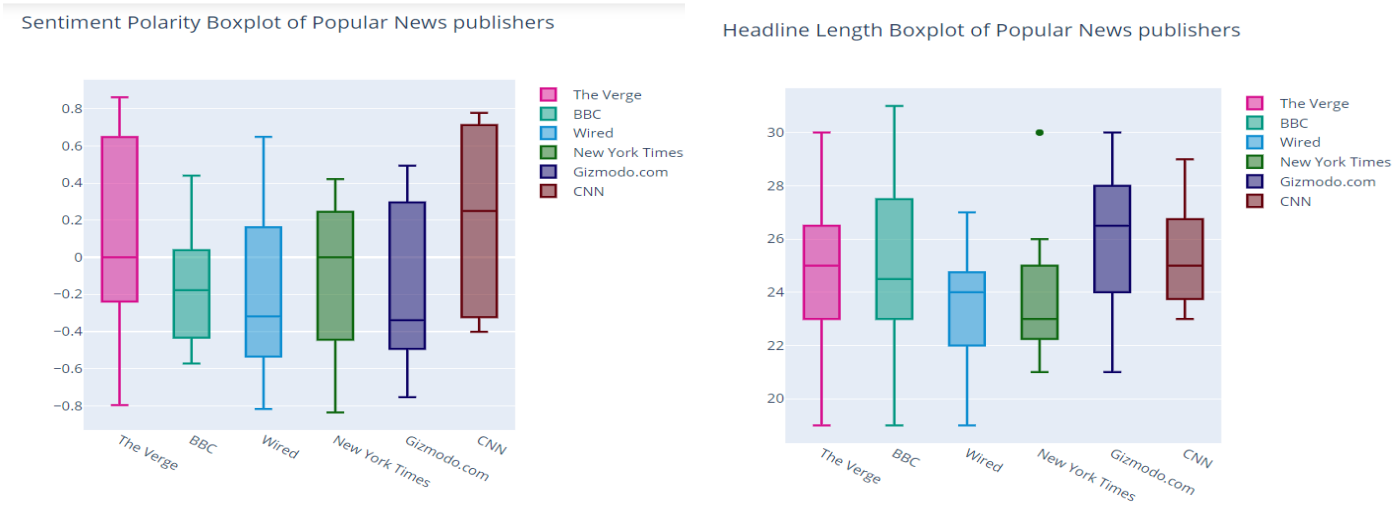

Fig. 2. The mean polarity

Going another level deeper into the sentiment analysis of these news sources, the boxplot of every news publisher represents interesting information. The five number summary, which includes the minimum, Q1 (the first quartile, or the 25% mark), the median, Q3 (the third quartile, or the 75% mark), and the maximum gives us the detailed idea about the exact nature of news these sources publish. Sometimes mean can be misleading as it is prone to extreme values or outliers and in such cases, median gives us a better idea. Looking at the graph we can see that all the publishers report news on both sides of the spectrum (i.e. positive and negative). For example, the medians of Wired and Gizmodo are both significantly low which confirms without any doubt that these publishers tend to have negative outlook to their stories. The word count in the headline is often neglected, but it is a critical aspect when it comes to engaging with the viewers. According to research, it was discovered that the sweet spot for headlines is 18–30 words. Anything above and below that saw reduced click-through rates. We see that most of the news sources seem to follow the similar trend.

WORD FREQUENCY DISTRIBUTION AND ZIPF'S LAW

Zipf's law states that given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table. Therefore, word number ‘n’ has a frequency proportional to ‘1/n’. Thus, the most frequent word will occur about twice as often as the second most frequent word, three times as often as the third most frequent word, etc. So essentially, what the law actually states is:

1. The most appearing word in a corpus suppose has frequency f
2. The second most appearing word would have frequency roughly f/2
3. Then the third most appearing word would have frequency roughly f/3 and so on.

What is astonishing is that this law holds true for almost all large natural language corpuses. For e.g., books, ancient scripts, even temperature trends over past years etc. In order to validate this law on our data we have plotted the distribution for positive and negative sentiment words separately (Fig. 3).

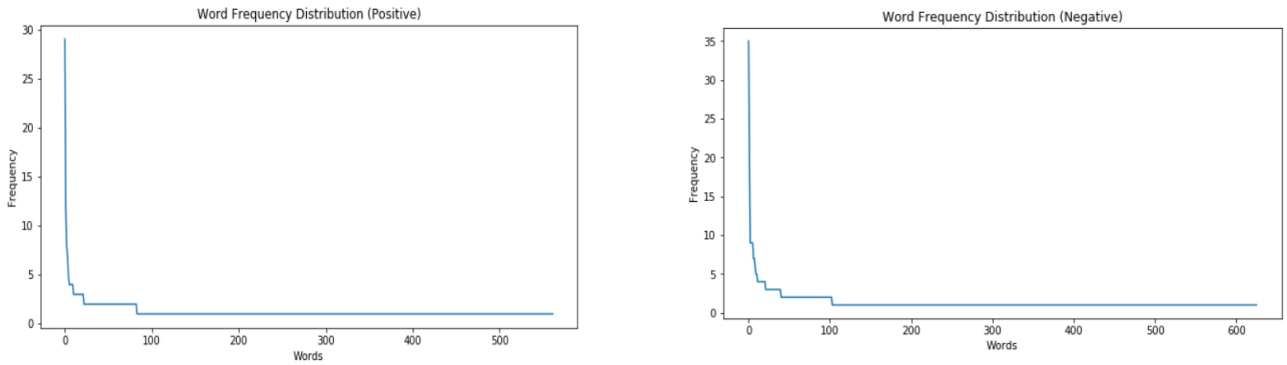

Fig. 3. Word frequency distribution for positive (on the left) and negative (on the right) sentiments

It can be seen from the above two plots that the both the positive sentiment words as well as the negative sentiment words follow the Zipf’s law. But how do we know whether it follows the Zipf’s law? By observing the plot, we see words which close to zero have a very high word frequency and then it drops to an 'L' shaped curve, which is typical characteristic of a Zipfian distribution. Furthermore, it can also be inferred that Zipf’s

law follows the Pareto distribution where only first 20% of words results into 80% of frequency of entire corpus. It is important to mention that in all the results, there are some instances where the word frequency not always falls to 1/2 to its previous, which is fine because overall pattern approximates the nature.

WORD CLOUD: TO SEE WHAT IS DOMINATING THE NEWS ARTICLES

The word cloud is an analysis technique wherein more commonly the term appears within the text being analysed, the larger the word appears in the image generated. The Word cloud image below (Fig. 4) shows which words dominated the content of the news during the period that the data was collected. It is no surprise that the word Facebook is the largest in the word cloud, given the fact that we have collected news about the Facebook outage. Some of the other words that appear more frequently in the news are outage, Instagram, WhatsApp, user, service and app that are all related to this incident.

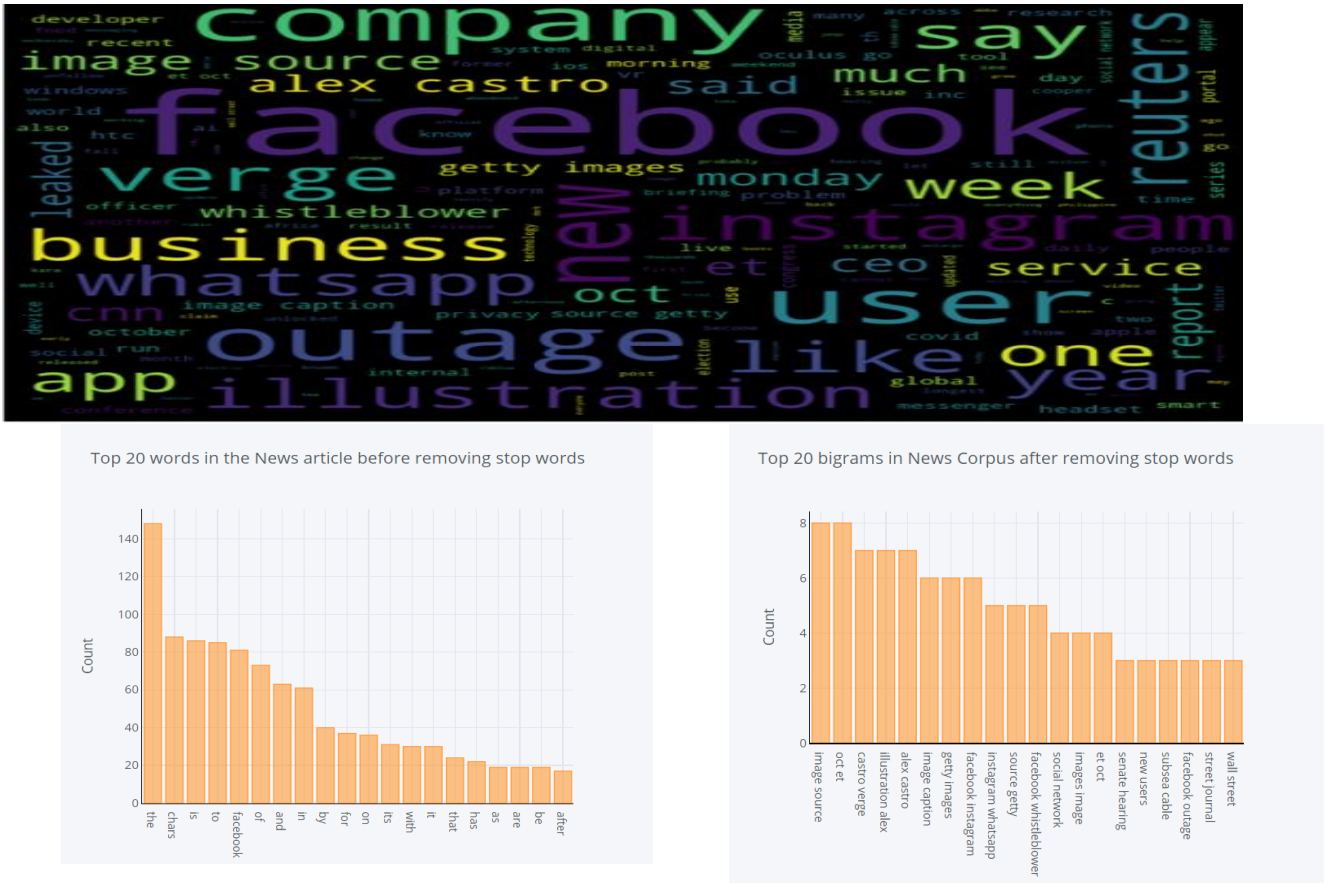

Fig. 4. The word cloud analysis

ANALYZING THE DISTRIBUTION OF WORDS AND BIGRAM ANALYSIS

Some English words occur together more frequently. For example - blue sky, do or die, best performance, heavy rain etc. So, in a text document. In this section, we have analyzed which two words appeared most frequently in the news article corpus. Before we get to that, the stop words must be removed. We can see that before removing the stop word, the top 20 most frequently occurring words in the corpus are the, to, of, in, etc. which does not make much sense these are the words that do not convey anything about the topic. Other thing to notice here is that again even here we do see the distribution following the Zipf's law. After removing the stop words, we see that most of the words make a lot more sense, even though there are still some stop

words present like chars, which do not make much sense but overall we see that the stop word removal does a good job to give us an indication as to what dominated the news. We can see that by looking at bigrams instead of just single words we are able to get a better sense of what is trending in the news. Take Facebook whistleblower as an example.

SECTION 2. ANALYZING THE SHARE MARKET

Once we are done with the analysis of sentiments and got visualization for our predictions ready to go, we can go over to the last stage of our project and explore how those results affect share market and describe what has happened to Facebook market share on October 4th.

Here we are going to fetch the data from marketstack.com API for one month from September 19th to October 19th.

Collection process of share market data for a company consists of the following:

- Use the above link and sign up to get the API key
- Get the required stock market data from the API URI
- Pre-process the obtained share market data and keep only the required fields
- Filter the data such that to get data for only one company and for certain date range.

Finally, we merged datasets so that we could see both news articles and stock market together and then calculated the sentiment scores for the stock news articles exactly as it was described earlier. For the sake of plotting we used MinMaxScaler() method from sklearn pre-processing module to normalize close price of the stock market so that it would be between 0 and 1 just like the sentiment score.

Below (Fig. 5) we showed how the negative sentiment of the news and the stock's end of the day close price correlate with one another. The Pearson's and Spearman's correlation coefficients approved that there is no correlation between those two. These coefficients are related to each other and tell us how the sentiments affected the stock market and vice versa. As Pearson's correlation is not suitable when the distribution of data is normal, we needed to take into account Spearman's coefficient as well. However, despite all of this and going through various news sources, none of them showed any significant correlation between two curves.

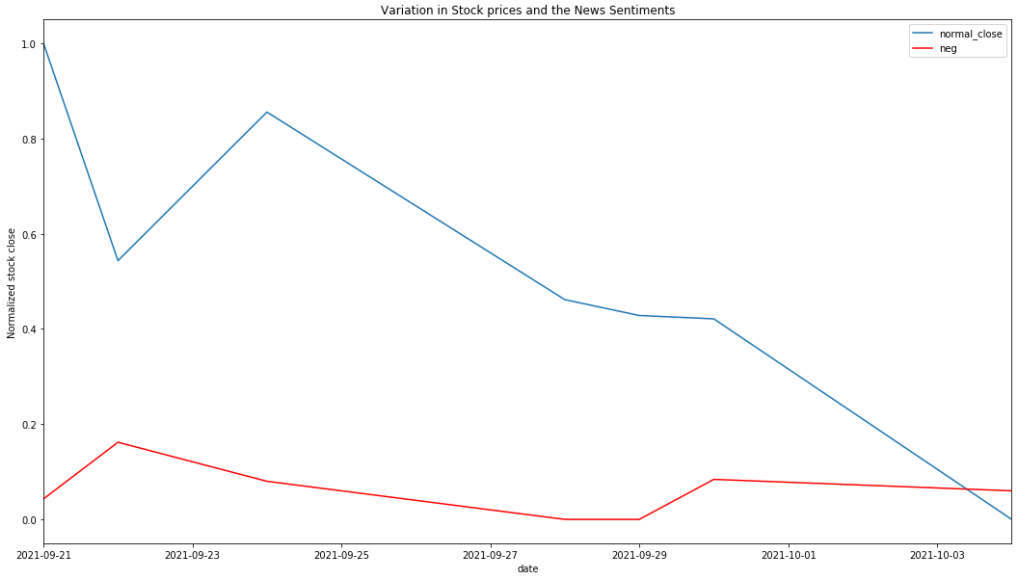

Fig. 5. Variation in stock prices and the news sentiments

CONCLUSION

In this paper, we have gone through sentiment analysis of news articles and illustrated different tools to visualize and apply this analysis for understanding how it can be related to an estimate of big companies' stock shares (e.g. Facebook) in times of their outage or any occasion causing those companies' share go down.

We have concluded that both positive and negative sentiments associated with over 4000 news articles indicate approximately equal presence in the news articles. Furthermore, we also found that there are some news publishers like Reuters, which have a high median negative sentiment polarity, and, on the contrary, some publishers like TechCrunch tend to publish primarily positive news, which requires further explanation. We were also able to figure out the ideal length of the headlines from the trends observed what told us that 18-30 words is the most common headline's length. All the subsequent analysis was aimed to find out the major trends in the news articles. In the end, we showed how Facebook's shares were affected by the accident in terms of their price and performed the sentiment analysis of the financial market news to determine if there exists any correlation between the end of the day close price and the negative news sentiment. It was found that there appears to be very small degree of negative correlation between the two.

We believe that this project can potentially be extended to solve globally more significant problems to employ sentiment analysis of the news articles in order to see its impact on the world around us and identify fake news using the proposed solution methodology by recognizing biased and unreliable publishers.

REFERENCES

1. https://www.analyticsvidhya.com/blog/2021/08/why-must-text-data-be-pre-processed/
2. https://www.pluralsight.com/guides/importance-of-text-pre-processing