Fake News Detection

Shreeja Pal

1. Introduction

1.1 Background

News is an essential part of everyone's life. To the analysis of a newspaper delivering correct news is essential, especially in a country like India. The freedom of media is being exploited by fake news and sharing content that's against the law. All news is not real or are based on someone's judgment and is not factual. We can classify news on the basis of their authenticity using python. We can find the performance of a certain newspaper studying their ratio of real news to fake news. A type of yellow journalism, fake news encapsulates pieces of news that may be hoaxes and is generally spread through social media and other online media. This is often done to further or impose certain ideas and is often achieved with political agendas. Such news items may contain false and/or exaggerated claims and may end up being virtualized by algorithms, and users may end up in a filter bubble.

1.2 Problem

There have to be some parameters that can decide whether a news is fake or not. There can be a polling system to decide so. This could decentralize the system but could lead to a lot of controversies

2. Data acquisition and cleaning

2.1 Data sources

The dataset used for this capstone project- is called news.csv. This dataset has a shape of 7796×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE. The dataset takes up 29.2MB of space.

2.2 Data cleaning

The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

This advanced python project of detecting fake news deals with fake and real news. Using sklearn, we build a TfidfVectorizer on our dataset. Then, we initialize a PassiveAggressive Classifier and fit the model. In the end, the accuracy score and the confusion matrix tell us how well our model fares.

2.3 Feature selection

The authenticity of Information has become a longstanding issue affecting businesses and society, both for printed and digital media. On social networks, the reach and effects of information spread occur at such a fast pace and so amplified that distorted, inaccurate, or false information acquires a tremendous potential to cause real-world impacts, within minutes, for millions of users. Recently, several public concerns about this problem and some approaches to mitigate the problem were expressed. In this paper, we discuss the problem by presenting the proposals into categories: content-based, source-based, and diffusion-based. We describe two opposite approaches and propose an algorithmic solution that synthesizes the main concerns. We conclude the paper by raising awareness about concerns and opportunities for businesses that are currently on the quest to help automatically detecting fake news by providing web services, but who will most certainly, in the long term, profit from their massive usage.

The dataset used for this capstone project- is called news.csv. This dataset has a shape of 7796×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE.

This section describes the preprocessing of the collected data for data analysis. As TW's text information contained a variety of information, information extraction through text analysis was required. Therefore, URL information, special characters (hashtag, mention, question, emphasis), emphasized words, emotion information (number of positive/negative words, emotional score) were extracted from the TW text information, and the extracted information was saved in 'text_info'. The NLTK package was used for the words and sentences expressing emotions in texts. For the emotional score, a method used by Castillo et al. [42] was used, that is, 1 point was assigned whenever a strong positive word was present in the text information and 0.5 points in the case of a weak positive word. Likewise, −1 point was assigned for a strongly negative word and −0.5 point for a weak negative word.

A process was required to merge the information scattered across different tables for convenient data analysis. Since the schema of 'ST_info' was consistent with that of 'QRT_info', the two tables were merged. That table was then merged with 'text_info' and 'user_info' using TW ID as a key, and the result was saved in the 'TW_info' table. The schema of 'TW_info' is defined in Table 1, and each record contains the status information of the TW related to certain news, information extracted from the text of TW, the information about the user who wrote the TW, and the status of real/fake news. 'TW_info' contained more information than the information collected using only conventional ST. Furthermore, because the status information of TW included the ID information of the parent TW of TW, it was easy to express the propagation pattern of TW and the depth of the propagation tree by tracing the parent TW ID.

3. Exploratory Data Analysis

3.1 Calculation of target variable

For the fake news and real news to be used in the analysis, data provided by Kaggle were used [57,58]. Kaggle provides global open data for various areas in the CSV or JSON format and provides data for already-confirmed fake news and real news. The collected information consisted of the news article's headline, writer, date of the Tweet, and real/fake news status, and was stored in the 'news_info' table in the database. For data analysis, the news released after 2015 was used; this was the time point when the Quote Retweet function was added.

Tweepy [59] and Selenium, a Web-scraping tool, were used [60] to collect the ST that mentioned each news, QRT that quoted it, and information of the user who wrote the TW. ST and QRT that had mentioned certain news from January 2015 to April 2019 were collected using Selenium.

Function 1 expressed pseudocodes for collecting ST that mentioned news for a certain period (startDate, endDate). By using the Selenium driver, ST that mentioned certain news was searched (lines 1–2). Moreover, to fetch all ST information on a Web page, the page was scrolled through until the end of the Web page was reached (lines 3–7). This was done because only partial content was shown when the output content of ST was large. When all ST information was displayed on one page, the respective Tweet information was parsed using the tagged keywords from HTML codes of Webpage and then read into a list (lines 8–23). Respective information of the list was merged into one data frame and saved in the 'ST_info' table (lines 24–25).

Function 2 showed the pseudocodes for receiving the ID of the collected ST and collecting QRT that quoted it. QRT had depth information (lines 1–2), and the ID of TW was used to search QRT that quoted the TW (lines 3–4). Afterward, the process proceeded following the same method as that of Function 1 (lines 7–26), and the added QRT information was saved in 'QRT_info' (line 27). If QRT that quoted a QRT existed, the QRT information could also be collected using a recursive call (line 28).

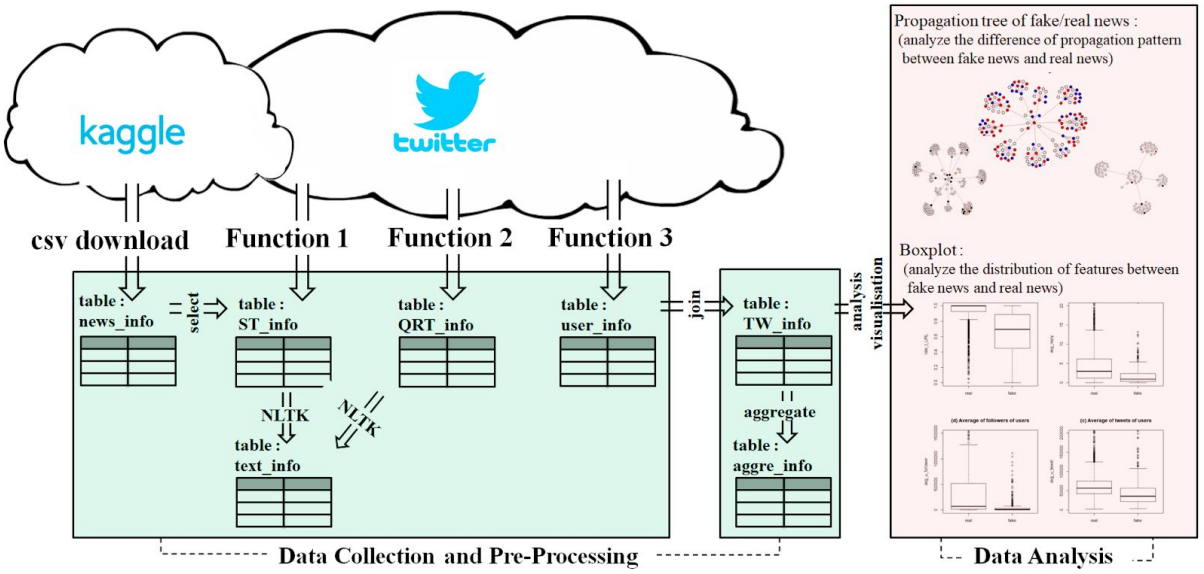
Function 3 showed the pseudocodes for collecting user information. For the user information, the user's followers, following, number of Tweets, user's URL, user-account creation date, and bio were collected using Tweepy and the ID of the user who wrote the TW. The collected user information was saved in the 'user_info' table.

Based on this method, information was collected for 16,453 cases of TW related to 1387 fake news and 56,651 cases of TW related to 2085 real news, and 65,405 cases of users who wrote the TWs.

3.2 Data Collection and Preprocessing

This section describes the data collection method and preprocessing. Figure displays an overview of fake news analysis modelling using Twitter data. It shows a method of collecting Twitter data, including news reports, for which the veracity has been confirmed, Tweets, Quote Retweets, and user information. In addition, Figure shows the preprocessing, visualisation, and statistical analysis for data analysis. A Tweet that mentioned a news directly was called ST (Seed Tweet) and a Quote Retweet was called QRT (Quote ReTweet). Moreover, Tweets, including both ST and QRT, were called TW (Tweets). Information of ST that had mentioned respective collected news was saved in the 'ST_info'

table, QRT information in the ‘QRT_info’ table, and user information in the ‘user_info’ table. The ST collection method was described in Function 1, the QRT collection method in Function 2, and the user information collection method in Function 3.



Furthermore, to extract a variety of additional information from ‘ST_info’ and ‘QRT_info’, information such as URL, special characters, emphasised words, and emotional score, were extracted by using the Natural Language Toolkit (NLTK) package [56] and stored in the ‘text_info’ table. ‘ST_info’, ‘QRT_info’, ‘user_info’, and ‘text_info’ were merged as one data and saved in the ‘TW_info’ table. ‘TW_info’ was aggregated by news and saved in the ‘aggre_info’ table. The data of ‘TW_info’ and ‘aggre_info’ were used as statistical and visualisation data for data analysis

3.3. Preprocessing

This section describes the preprocessing of the collected data for data analysis. As TW’s text information contained a variety of information, information extraction through text analysis was required. Therefore, URL information, special characters (hashtag, mention, question, emphasis), emphasised words, emotion information (number of positive/negative words, emotional score) were extracted from the TW text information, and the extracted information was saved in ‘text_info’. The NLTK package was used for the words and sentences expressing emotions in texts. For the emotional score, a method used by Castillo et al. [42] was used, that is, 1 point was assigned whenever a strong positive word was present in the text information and 0.5 point in the case of weak positive word. Likewise, −1 point was assigned for a strong negative word and −0.5 point for a weak negative word.
A process was required to merge the information scattered across different tables for convenient data analysis. Since the schema of ‘ST_info’ was consistent with that of ‘QRT_info’, the two tables were merged. That table was then merged with ‘text_info’ and ‘user_info’ using TW ID as a key, and the result was saved in the ‘TW_info’ table. The schema of ‘TW_info’ is defined in Table 1, and each record contains the status information of the TW related to certain news, information extracted from the text of TW, the information

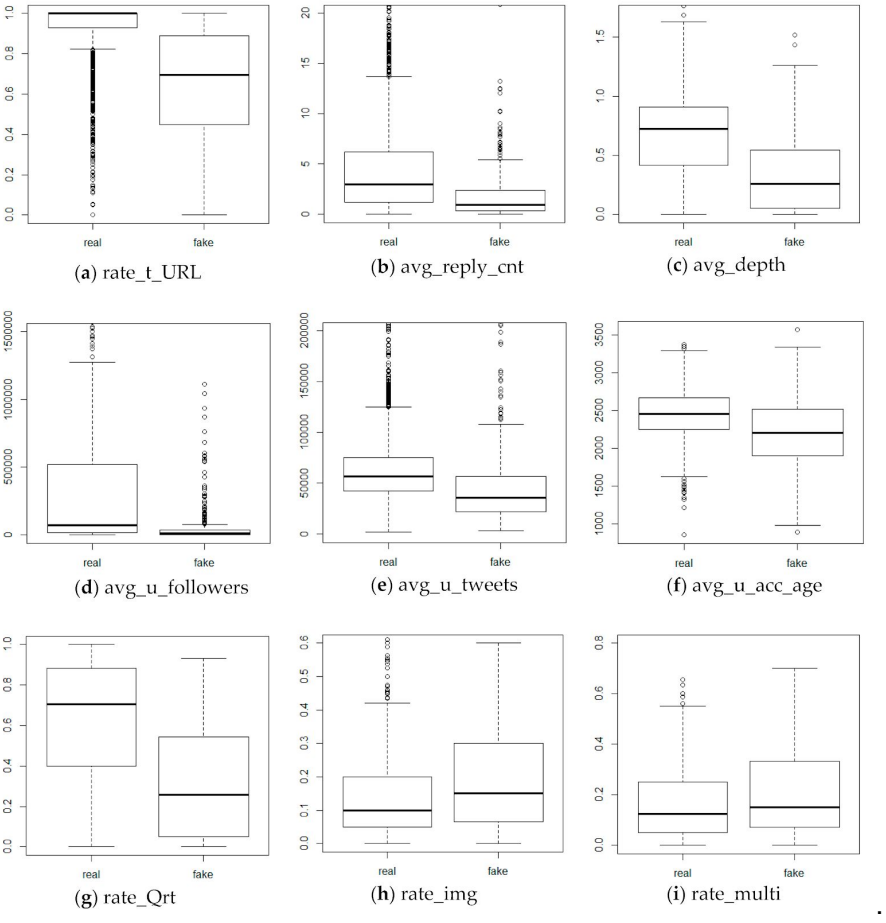
about the user who wrote the TW, and the status of real/fake news. ‘TW_info’ contained more information than the information collected using only conventional ST. Furthermore, because the status information of TW included the ID information of the parent TW of TW, it was easy to express the propagation pattern of TW and the depth of the propagation tree by tracing the parent TW ID.

The information of ‘TW_info’ was aggregated by news in order to show TW information for each news, and this information was saved in the ‘aggre_info’ table. Here, 405 cases of fake news and 2085 cases of real news, for which ten or more STs were registered, were used. Table 2 defines the schema of ‘aggre_info’ table, and each record of ‘aggre_info’ shows the average value of TW information and the status of real/fake news for each news.

4. Predictive Modeling

4.1. Statistics and Visualisation

By using the information collected and preprocessed in the previous chapter, statistical analysis was performed to find the best features of fake news, and the results were visualised in boxplots, as shown in Figure. Figure shows the features representing the differences visually between the fake news and the real news by using the boxplots. In Figure, the x-axis of each boxplot shows the status of real/fake news (0: real news, 1: fake news), and the y-axis indicates the value range of each feature.



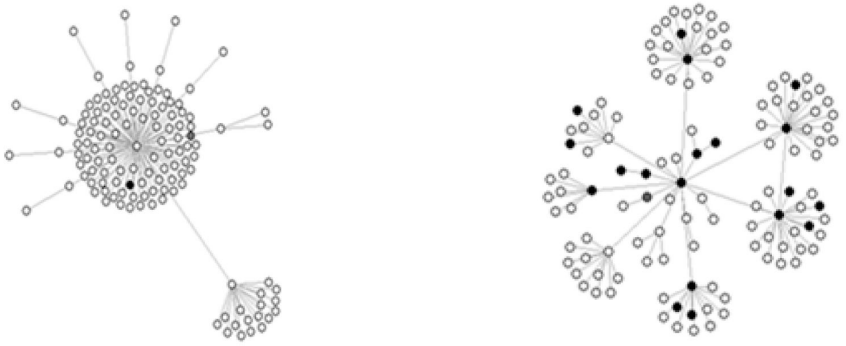
(a) rate_t_URL (b) avg_reply_cnt (c) avg_depth

(d) avg_u_followers (e) avg_u_tweets (f) avg_u_acc_age

(g) rate_Qrt (h) rate_img (i) rate_multi

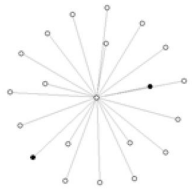
The attributes that showed significant differences between fake news TW and real news TW were the average number of replies, average depth of propagation tree, proportion of including URL, user's influence/activeness/active period, QRT's proportion, proportion of including multimedia, etc. Figure compares the proportion distribution of URLs in TW, and it was confirmed that for real news, majority of TWs mentioned the URL. Figure compares the average number of replies for TW written. It confirms that the average number of replies is lower for the TW of fake news than that of the TW of real news. Figure shows the average depth of TW propagation tree, and confirms that real news is propagated slightly more deeply. This means that there are more users who come to know about the news indirectly via propagation compared to users who encounter the news directly, and it was confirmed that fake news had relatively lower spreading power. Figure compares the average number of followers, that is, influencing power of TW users, and confirms that the distribution of users having a relatively larger influence is high for the real news TW. Figure shows the TW users' average number of TWs, and confirms that the average number of writing TWs is slightly higher for the real news TW users. Figure shows the user account age (days) of TW writers, and confirms that the account age of fake news TW writers is slightly lower. Figure compares the rate of using QRT between real news and fake news, and confirms that the QRT is used more frequently for real news. Figure shows the proportion of pictures included in the TW, and Figure shows the proportion of multimedia contents (picture or video) included in the TW. It was confirmed that the proportion of including pictures and videos was relatively higher for fake news.

Figure expresses the propagation patterns of real news and fake news based on the information of Table. In the propagation tree, the centre Root node indicates a news report, and child nodes consist of TWs that mentioned the news report (the nodes with the tree level of 1 are ST, and the nodes with higher number than 1 are QRT). The nodes expressed in black indicate the highly influential users.

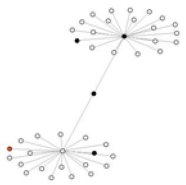


(**a**) TW propagation tree of fake news; (**b**) TW propagation tree of real news

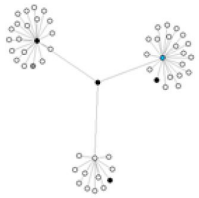
In this study, having 200,000 followers was assigned as the threshold of high influential power. As shown in Figure, the real news propagation tree has a higher proportion of highly influential followers compared to the fake news propagation tree. Moreover, it was confirmed that more QRTs were used for real news, and more TWs were propagated from influential users. It means that the difference between fake news and real news is statistically significant for every feature excluding the average number of replies in Figure.



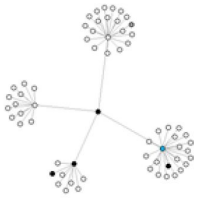
Dec. 13, 2017



Mar. 07, 2018

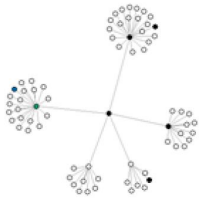


May. 09, 2018

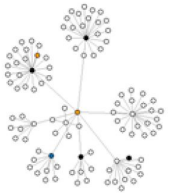


May. 23, 2018

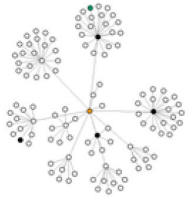
Jun. 06, 2018



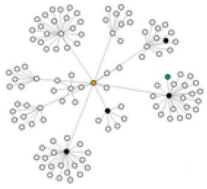
Jun. 13, 2018



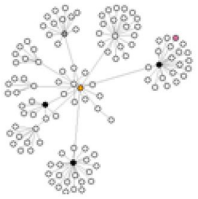
Aug. 29, 2018



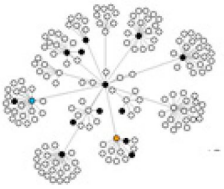
Sep. 19, 2018



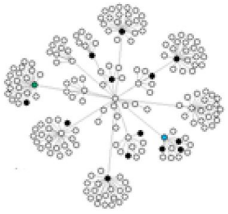
Sep. 26, 2018



Oct. 10, 2018



Nov. 07, 2018



Nov. 09, 2018

Mar. 22, 2015

Mar. 29, 2015

Apr. 05, 2015

Apr. 12, 2015

May. 24, 2015

Jun. 14, 2015

Sep. 06, 2015

Sep. 13, 2015

Sep. 20, 2015

Oct. 11, 2015

Oct. 25, 2015

Nov. 01, 2015

Nov. 22, 2015

May. 08, 2016

May. 15, 2016

Jun. 26, 2016

Jul. 24, 2016

Aug. 14, 2016

Sep. 25, 2016

Oct. 02, 2016

Oct. 09, 2016

Oct. 16, 2016

Dec. 04, 2016

Dec. 11, 2016

Apr. 23, 2017

Apr. 30, 2017

May. 28, 2017

Jul. 16, 2017

Aug. 13, 2017

Nov. 05, 2017

Jan. 07, 2018

Mar. 18, 2018

Apr. 15, 2018

Aug. 05, 2018

Sep. 02, 2018

Nov. 11, 2018

Nov. 20, 2018

Mar. 03, 2019

Mar. 17, 2019

Mar. 23, 2019

Each propagation tree was additionally drawn for only the registered date of TW. For the fake news, it was confirmed that the propagation period of TW was long although the propagation depth was low.

4.2. Neural Network-Based Fake News Classifier

This section discusses the comparative experiment performed using Castillo's method [42] to verify the performance of the fake news analysis modelling method proposed in this study using neural networks, which is one of the machine learning techniques. Furthermore, classification models were compared by using all features as learning data to verify the best variables confirmed earlier. For this, the classification model was defined according to the range and combination of data used for three types of learning, as shown in Table 6. Classification Model 1 used the best features proposed by Castillo et al. [42] as learning data. The best features confirmed by Castillo et al. [42] were topic-based features (rate of url, avg of senti, rate of exclam), user-based features (avg of Tweets, avg of friend), and propagation-based features (avg of rtcnt), and only the data of ST (depth is 0) were used. Classification Model 2 used all features of TW, including both ST and QRT, for the learning data. Classification Model 3 used only the best features of TW, including both ST and QRT, for the learning data. As the experimental tools of the performance evaluation, the neural network model of R nnet package [62,63] provided as open source was used, and training and validation was performed by classifying it into 70% training data and 30% valid data [64].

5. Conclusions

In the present era, numerous amounts of information are generated every day amid advancement of electronic devices and social media, and among such information, false information, called fake news, exists as well. Fake news creates various problems in our society, and endeavours are required to solve them. Accordingly, many researchers have conducted studies to detect rumours and fake news by using Twitter data, one of the popular social media outlets. However, Twitter has added new features over time, and consequently, additional studies are required to consider them.

In 2015, Quote Retweet was added as a feature in Twitter. It contains more information than the conventional Retweet, and has an advantage that the propagation path of information can be easily identified since the parent Tweet can be easily found. Furthermore, the users are switching from conventional Retweets to Quote Retweet. Therefore, this study proposed a fake news analysis modelling method to acquire more data by collecting Quote Retweets and identify the best features that would have positive impact on fake news detection. The proposed fake news analysis modelling method provided a method to conveniently collect Tweets, Quote Retweets, and user information from Twitter and to preprocess the collected data into a format that could be easily used in data analysis. Furthermore, the best features having influence on fake news were identified through effective visualisation and results of statistical analysis obtained from the preprocessed data.

The data containing news and veracity information of news were collected from Kaggle, an open data analysis platform. Furthermore, Selenium was used as a tool for collecting the information of Tweets and Quote Retweets from Twitter, and Tweepy was used to collect the information of users who had written the Tweets. In addition, the NLTK package was used to extract the emotion information, emphasized words, special characters, and URLs from the texts of collected Tweets and Quote Retweets.

The results of visualisation and statistical analysis to investigate the best features from the collected data indicated significant differences between fake news and real news in terms of existence/absence of information source, replies for Tweets, influencing power of Tweets, rate of using Quote Retweet, depth of Tweet propagation tree, and rate of quoting picture/video. Furthermore, the results of propagation period confirmed that fake news was propagated for a longer period gradually but constantly compared to real news.
Performance evaluation was performed using the neural network-based fake news classifier to investigate whether the best features identified through the proposed method really had a positive impact on the fake news classification accuracy. In the results, the classification model that added Quote Retweet information showed 4.57%, 10.51%, and 9.79% higher classification accuracy for fake news, real news, and total, respectively, compared to the classification model using the conventional Tweet information only. Furthermore, in the performance comparison between the classification model using all features and the classification model using the best features only, the classification model that learned the best features only showed 8.48% higher classification accuracy for fake news, thereby confirming that the best features had a positive impact on fake news classification.
There is still room for improving the quality of text information by applying more detailed text analysis on Tweets. If the user reaction information and emotion information of a much higher quality can be used through this process, it is expected that the fake news classification performance can be further increased. This will be left for future work.