“What’s the weather like today?” Looking out of the Twitter window.

Alisa Hussain

Monday 24th May, 2021

1 Introduction

In this report we investigate the extent to which we can infer the weather from the sentimentality of relevant posts on Twitter. Twitter is a social media platform where users post ‘tweets’ with text of no more than 280 characters. We use an API to extract tweets regarding London weather from each day over an 8 day period. We then compare two sentiment identification methods: Stanford University’s CoreNLP library and training a logistic regression model on a labelled tweet corpus using the TF-IDF metric. We determine the average sentiment across each day - ‘positive’, referring to a favourable disposition towards the weather or ‘negative’, an unfavourable disposition. We finally investigate if different sentiments reflect the weather forecast. In the following sections we describe the methodologies undertaken to reach a conclusion.

2 Data

Twitter We analyse posts regarding the weather in London on the social media platform, Twitter. London was chosen for two main reasons: it is small enough (35km radius) for weather conditions be similar across it; and we were confident there would be sufficient relevant tweets with it being the capital of the United Kingdom. We configured credentials for a Twitter API and used Tweepy [1], a python library, to connect to this API through Python.

CoreNLP Stanford University’s natural language processing library was one of the two means of identifying sentiment. Sentences are scored with a value between 1, the most negative sentiment, and 4, the most positive.

Logistic Regression Before training a LR model, data must first be preprocessed. Nltk was the python library used to remove stopwords, split sentences into tokens and reduce individual words to their stem [3]. The sci-kit learn library was used to convert text data into a TF-IDF vector [4].

Evaluating model performance To train the LR model and evaluate the performance of our two approaches we acquired a Twitter sentiment corpus comprising 1,578,627 tweets posted between 2007 and 2011 from two sources: the University of Michigan Sentiment Analysis competition on Kaggle and the Twitter Sentiment Corpus by Niek Sanders [5]. Tweets have been hand-labelled with a 1 representing positive and 0 representing negative sentiment.

Result Visualisation To visualise the performance of the models we used the data visualisation libraries matplotlib [6] and seaborn [7]. To interpret the overall sentimentality of tweets from each day, we used the pandas [8] package for the use of its dataframes and plotting capabilities.

Historical weather data To determine if sentiment in tweets reflect weather data, we consult each day’s temperature and weather conditions from timeanddate.com [9].

3 Model and Implementation

We compared the performance of the CoreNLP library with a LR model when detecting sentiment in tweets and determined the proportion of positive to negative tweets regarding London weather each day. We found these two approaches performed very differently.

3.1 Gathering tweets

3.2 CoreNLP

In order to use CoreNLP, the 504MB package had to be downloaded. A background Java CoreNLP server process was started and a client instance created to pass text to the server process and accept the returned annotated results. In the initialisation of the client and server we specify: the amount of memory to be allocated to the Java process; the endpoint for communication between the server and client; and the requested annotations, which in our case is only sentiment scores.

The server returns the text, having labelled each sentence with a number between 1 and 4 - 1 representing the most negative sentiment, and 4 the most positive. We average these ratings across the entire tweet, and class the tweet as 'positive' if the average rating is greater than or equal to 2.5, and 'negative' otherwise.

3.3 Logistic Regression

Central to our LR model was the labelled corpus of tweets. The aim of the machine learning model is to identify a pattern between the structure of tweets and its assigned label. We decided between naive bayes (NB) and logistic regression because of their suitability for handling discrete data - in this case, binary labels - and their prevalence in research. For example, the former has been used historically because of its training speeds and ability to perform well with a small amount of training data. An example was its use in sentiment analysis of restaurant reviews in 2012 by Kang et al [10] The latter has been used more recently, for example by Jain et al in 2018 for predicting cryptocurrency prices using twitter sentiment analysis [11]. We trained and tested both these models and found LR predicted with higher accuracy. Scores are detailed in section *4.1*.

We loaded the dataset and retained only relevant columns - sentiment label and text. We then preprocessed the tweets to simplify the content of them as follows:

- **Convert to lower case** The same words whether capitalised or not should be treated the same.
- **Remove URLs, user tags and hashtags** We want to focus on the main body of text. Additionally there are different variations of URLs (e.g a shortened versions, whether or not "http" is included) which may confound the model.
- **Remove punctuation** Punctuation does not generally add to sentiment information in text.
- **Remove stop words** These words do not add meaning so are removed to save computation.
- **Stem words** Affixes of words are removed so they are reduced to their root and different versions of words are treated the same. For example, 'loving' and 'loved' would both be reduced to 'lov'.
- **Lemmatisation** Reduces words to their basic form. For example, 'worst' would be reduced to 'bad'.
- **Tokenisation** Converts text into an array of words or 'tokens'.

Post pre-processing, tweets were converted vector form to be handled by the machine learning algorithm. We achieved this using TF-IDF: analysing the occurrence of words in the corpus and assigning creater importance to less common words. The sci-kit learn library was use to form a vector for both training and testing. An alternate option to TF-IDF was bag of words, although this was avoided since vectors from this method contain no information on the importance of words and this therefore often results in worse machine learning model performance. The dataset was split with a ratio of 80:20 for training and testing sets respectively. The sci-kit learn library was used to train both a NB and LR model. Upon comparing accuracy scores, as will be detailed in the following section, LR was deemed the most appropriate model.

3.4 Applying the models

After establishing the models, we applied them to twitter data for analysis. Twitter API credentials were configured to produce API keys. Unfortunately, only tweets from the 8 days prior to the date of API request were accessible, so trends could only be established over a week-long period. For access to the Twitter API through Python, we used the Tweepy library [1].

Our initial approach was to query the word "weather" and request tweets located in the London area, by inputting the longitude, latitude and radius. However, this required tweets to be geo-tagged - of which only 0.85% are [12]. Significantly limiting data available and affecting validity of results, this approach was written off. Instead, we queried "London weather", and collected 500 tweets a day at a time. In addition, since the corpus was in English, we specified this language in the query. For example, the query for tweets from the previous day was formatted as follows:

```
tweets = tweepy.Cursor(api.search,
q = "london weather",
tweet_mode = "extended",
until = datetime.date.today() - datetime.timedelta(1),
lang = "en").items(500)
```

After retrieval, we predicted sentiment using CoreNLP and the LR model. Pre-processing of tweets before using CoreNLP was limited to removal of URLs, hashtags and user tags. These were passed from the client to the server using the `annotate` function. Numerical labels between 1 and 4 were returned for each sentence in the tweet. We averaged these annotations across each tweet. For example, take the following tweet. A number enclosed in square brackets represents a sentiment value.

I'm so disappointed in british weather. [1] I can't believe its been raining all week! [2]

Average sentiment: $\frac{1+2}{2} = 1.5$

To make comparisons with LR outputs easier, we converted each average sentiment value across a tweet, l to a binary label:

Table 1: Converting CoreNLP labels to binary labels

Original label value, l	New label	Sentiment
$1 \leq l < 2.5$	0	Negative
$2.5 \leq l \leq 4$	1	Positive

The above example would now be labelled *0*.

For LR predictions, we applied the pre-processing steps specified in section *3.3* and use the TF-IDF to create the text feature fed into the predict method, which we recall outputs binary labels. After gathering a prediction for each tweet, the overall proportion of negative to positive sentiment for each day is outputted graphically for interpretation.

4 Evaluation

4.1 Performance of models

We analysed the performance of our machine learning models - NB and LR - and CoreNLP to understand tthe extent to which results are valid. The metrics used were accuracy, precision, recall and F1-score. We use the twitter sentiment corpus to derive these scores.

Accuracy

Accuracy indicates the proportion correctly classified samples. It calculated as

$$Accuracy = \frac{\text{total correctly classified samples}}{\text{total samples}}$$

Table 2 shows the difference in accuracy between models. Since LR proved to be a more accurate model than NB we used this technique for our machine learning model. Surprisingly, it also classified 9.5% more sample correctly than the pre-trained CoreNLP library.

Precision, Recall and F1-score

Precision is the ratio between the true positives and all predicted positives. On the other hand, recall is the measure of our model correctly identifying true positives. Each measure comes with a trade-off from the other so we refer to the F_1-score, a value between 0 and 1, which is the harmonic mean of precision and recall. This is advantageous

Table 2: Accuracy of different models

Naive Bayes	Logistic Regression	CoreNLP
0.770	0.793	0.675

over a normal mean because it punishes extreme values. A higher score indicates less misclassified samples. These metrics are calculated as

$$Precision = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Positive (FP)}}$$

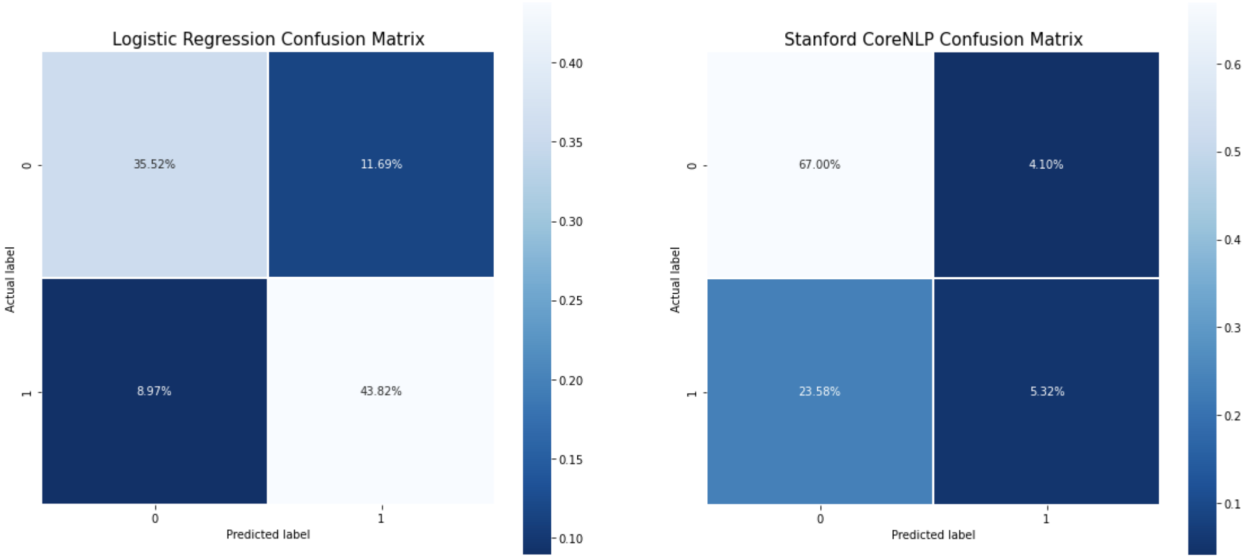
$$Recall = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Negative (FN)}}$$

$$F_1 = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

Table 3 compares these scores between the LR model and CoreNLP. These results are supported by the confusion matrices in Figure 1. The LR precision and recall scores, resulting in a high F$_1$ score of 0.809. On the other hand, CoreNLP's 40% discrepency between precision and recall results in a very low F$_1$ score of 0.217. When texts were classified as positive by CoreNLP, they were only actually positive just over half of the time as opposed to 78% of the time using LR. Additionally, CoreNLP only correctly identified positive sentiment 13.6% of the time - LR is a significantly better approach with a recall of 83%.

Table 3: Precision, Recall and F$_1$ scores

Metric	Logistic Regression	CoreNLP
Precision	0.789	0.536
Recall	0.830	0.136
F$_1$ Score	0.809	0.217



(a) Logistic Regression

(b) CoreNLP library

Figure 1: Confusion Matrices for tests with Logistic Regression and CoreNLP models

4.2 Sentiment Results

Figure 2 and 3 show the proportion of positive to negative sentiment across the top 500 tweets regarding ”london weather” over a week as determined by LR and CoreNLP respectively. At first glance, the predictions vary significantly between the models each day - on 22/05/2021 we have 51% of tweets classified as negative by our LR model, but 100% by the library, a 49% difference. Additionally, using the LR model, there appears to be a greater change in sentiment across the week - there is a 56% difference between the most positive and negative days, as opposed to a 21% difference when using the library.

CoreNLP results suggest an overwhelming negative opinion towards London weather across the week with a maximum proportion of positive sentiment of 21% on 18/05/2021. At a glance of the graph, one may question the validity of this data. For this reason, we broke down these predictions into the original scores (Figure 4). There were no tweets valued with a sentiment of 4, representing very positive, and there is an overwhelming majority of sentiment 2 being predicted, representing negative. This result is supported by the confusion matrix indicating an overwhelming prediction of negative sentiment - shown by the lighter shades of blue in Figure 1b. This also supports the low recall score - few tweets of a positive sentiment are identified because the significant majority of predictions are negative. Possible reasons for this will be discussed in the next section. The LR predictions are likely to be more accurate, because of the high values across all performance metrics in the previous section.

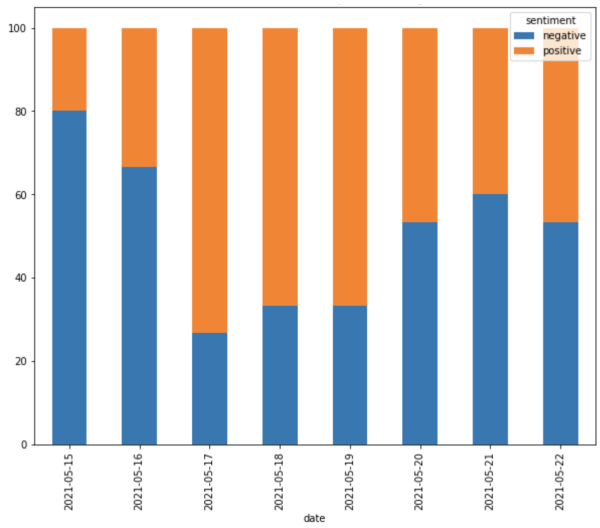


Figure 2: Logistic regression-identified sentiment over a week

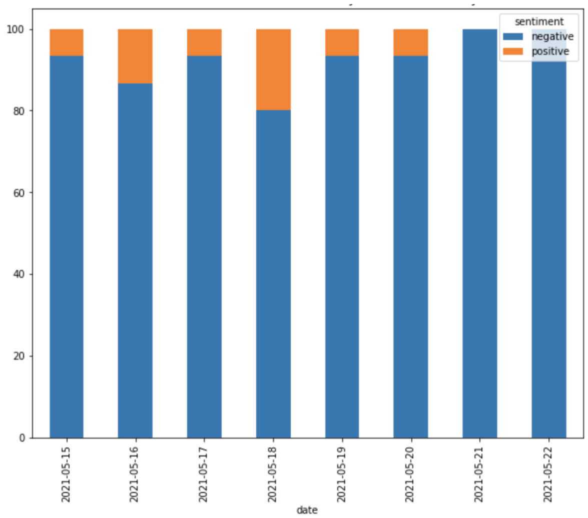


Figure 3: CoreNLP-identified sentiment over a week

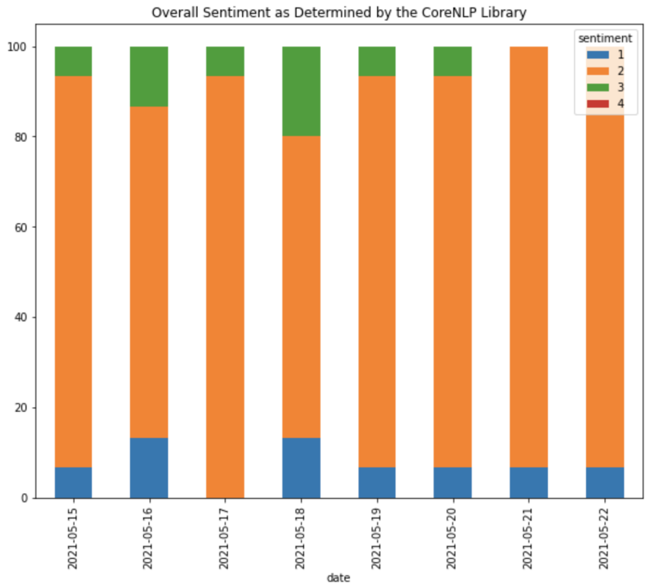


Figure 4: Breakdown of CoreNLP-identified sentiment into original labels

We consult historic average weather data in London from the 8 day period we are analysing [9] (Figure 5). See Figure 6 for weather icon labels. At first glance, we notice that the two days with rain also received the most negative tweets, according to the LR model. The day with the most positive attitudes was 17/05/2021. This positivity could be attributed to the fact this was the first day of no rain, which also exhibited the highest temperature across the week. Additionally, the sunniest day, 19/05/2021, which shared the highest temperature received the second most positive tweets.

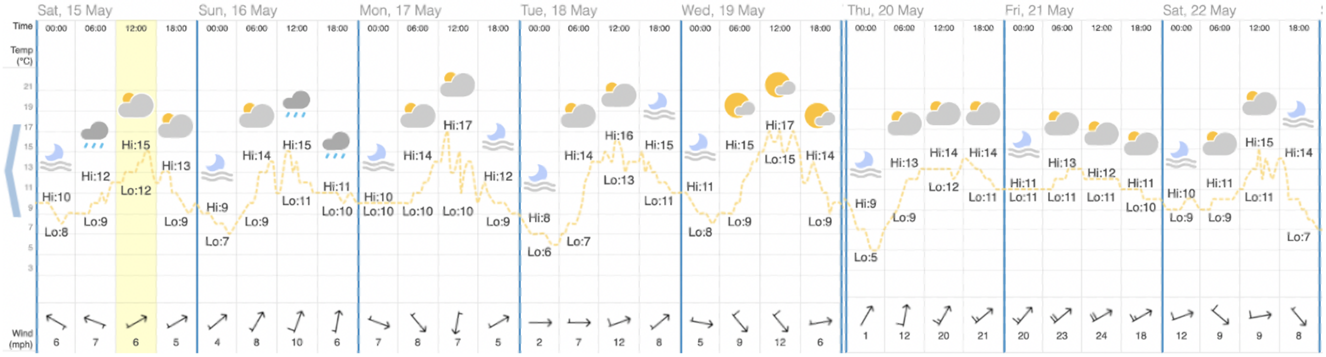
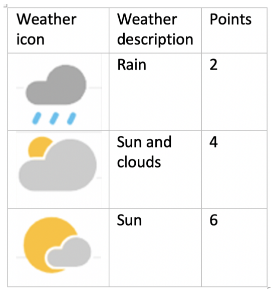


Figure 5: Historic weather data over an 8 day period



Weather icon	Weather description	Points
🌧	Rain	2
⛅	Sun and clouds	4
🌤	Sun	6

Figure 6: Weather point allocation

To supplement the visual trends, we convert the weather data into a numerical scale to reach valid scientific conclusions. We create a score for each day, using the weather condition icons and temperatures. A lower scores would represent colder or rainy conditions, and higher scores would represent warm or sunny conditions. Points allocated for weather conditions is shown in Figure 6. The average temperature every 6 hours is scored: the lowest average temperature across the week, 7°C is allocated one point and each degree above that is worth one point more.

In Figure 7, each plot represents a day. This shows each day's overall weather score against proportion of tweets that were positive that day.There appears to be a positive correlation. The pearson's correlation coefficient, **r**, calculated by

$$\mathbf{r} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

is 0.7972, indicating a strong positive correlation. For this particular week, according to our LR model, sentiment in tweets towards the local weather in London is more positive on sunnier, warmer days, and more negative on cooler, cloudier and rainy days.

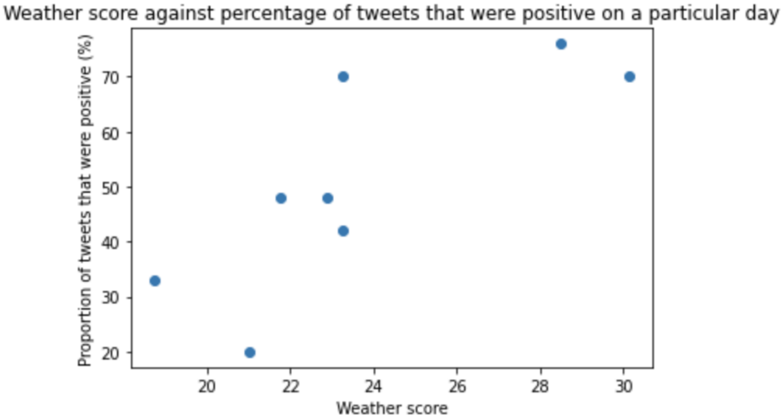


Figure 7

4.3 Further evaluation

Although our logistic regression model appears strong, there are assumptions our conclusions depend on. Additionally, we consider reasons why CoreNLP performed poorly in sentiment classification of tweets.

We analysed tweets of those based in London in the second last week of May 2021. We have assumed there is significant representation in the Twitter corpus of those with a particular demographic: Londoners of a similar age to those who typically use Twitter in 2021. Recall the corpus is composed of Tweets over 10 years ago - it would be understandable if the average structure and content of a tweet has evolved since then. For example, the maximum character count has doubled since date of corpus creation. Emoji use may have also changed, especially since the emoji library has increased since then. Additionally, we assume hashtags, user tags and URLs have no significance on twitter sentiment information for the sake of simplicity, and because of the minimal information on these in the corpus. We also only train on and analyse tweets in English. According to the 2011 census, 22.1% of Londoners speak another main language - the highest proportion across the UK. Our results may not be generalisable to those with an alternate main language. Furthermore, when pre-processing tweets for our LR model, although TF-IDF had clear advantages, word2vec may have been a better alternative because of its ability to recognise synonymous words.

The biggest issue related to the use of CoreNLP is that the sentiment annotation component has been trained on complete sentences. The structure of tweets may include more partial sentences and are often broken up by tags or emojis, which the library would struggle to recognise.

Although we determined a strong correlation between sentiment and weather conditions, this conclusion must be taken lightly. Not only is this trend valid for one city, but it is only valid for one week in May. This trend is therefore not generalisable, and it is important that further research be undertaken before applying these results to other contexts.

5 Conclusion

To conclude, we demonstrated the performance CoreNLP and logistic regression for predicting sentiment in tweets regarding the weather in London over an 8 day period. In particular, we used performance metrics to determine that out of Stanford's CoreNLP library, a naive bayes and logistic regression model, logistic regression performs best. CoreNLP performed poorly across all metrics, particularly recall - indicating the inability to recognise positive sentiment. This was in turn reflected in the analysis of weather-related tweets, where 89% of tweets were classed as negative. However, we cannot rule out the effectiveness of CoreNLP, since it could potentially be more effective at analysing other social media outputs such as LinkedIn, where there may be more full sentences with fewer emojis and hashtags.

Using the LR model we established that over 8 days in London, on sunnier and warmer days, the overall sentiment in tweets regarding weather were more positive than days colder and with rain. However, due to reasons discussed in section *4.3*, such as the small number of data points, this conclusion cannot be generalised to other dates or locations until further research has been undertaken.

There is a lot of room for further research to extend our work. For example, it would be valuable to analyse data over a longer period since trends could vary over different times of the year. Similarly, research could be extended to different geographical regions to improve representation. Furthermore, to boost accuracy of predictions, a more recent corpus could be analysed, or one could investigate the impact of hashtags and user tags on sentiment. Data sources could also be extended to additional social media platforms such as Facebook to increase and diversify the data analysed, since the user characteristics across different platforms could differ. Finally, with regards to CoreNLP specifically, more research should be undertaken regarding its effectiveness in alternate settings, particularly alternate social media platforms e.g. Facebook.

Despite the shortcomings in our approach a trend was established between sentiment in tweets regarding London weather and actual weather data using a logistic regression model. This is significant since it motivates further research into data that can be inferred solely through sentiment analysis of social media posts, such as weather conditions.

References

[1] J. Roesslein, "Tweepy: Twitter for python!," *URL: https://github.com/tweepy/tweepy*, 2020.

[2] "Search result faqs." https://help.twitter.com/en/using-twitter/top-search-results-faqs. (Accessed on 05/21/2021).

[3] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[5] "Twitter sentiment analysis training corpus (dataset) — thinknook." http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/. (Accessed on 05/21/2021).

[6] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[7] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[8] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.

[9] "Past weather in london, england, united kingdom — yesterday or further back." https://www.timeanddate.com/weather/uk/london/historic. (Accessed on 05/21/2021).

[10] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews," *Expert Systems with Applications*, vol. 39, no. 5, pp. 6000–6010, 2012.

[11] A. Jain, S. Tripathi, H. D. Dwivedi, and P. Saxena, "Forecasting price of cryptocurrencies using tweets sentiment analysis," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–7, IEEE, 2018.

[12] L. Sloan, J. Morgan, W. Housley, M. Williams, A. Edwards, P. Burnap, and O. Rana, "Knowing the tweeters: Deriving sociologically relevant demographics from twitter," *Sociological research online*, vol. 18, no. 3, pp. 74–84, 2013.