Decision tree and KNN based imputation methods are appropriate for non-parametric data such as this [1].

The data is assumed to be MCAR, as there is no known reason why there would be no tweets at all on the missing dates, and there is no common factor amongst the missing data [2].

[1] Bertsimas, D., Pawlowski, C., and Zhuo, Y. D. (2018). From Predictive Methods to Missing Data Imputation: An Optimization Approach. *Journal of Machine Learning Research*, 18, pp.1-39.

Yang, L. and Chiang, J.A. (2020) ‘Use case and performance analyses for missing data imputation methods in big data analytics’, *Proceedings of 2020 6th International Conference on Computing and Data Engineering* [Preprint]. doi:10.1145/3379247.3379270.

To use ycsb with Cassandra a change was needed to conf/cassandra.yaml, after [<https://github.com/brianfrankcooper/YCSB/pull/98/files>]

To run the Cassandra load test in ycsb a ycsb keyspace had to be created, as shown in appendix, and similarly a usertable had to be created in hbase.

‌

1. Details of the data storage and processing activities carried out, including preparation of the data and processing the data in a MapReduce/ Spark environment;[0-20]

2. A discussion of the rationale and justification for the choices you have made in terms of data processing and storage, programming language choice, machine learning [models](https://moodle.cct.ie/mod/resource/view.php?id=146081) and algorithms that you have implemented.[0-40]

3.Comparative analysis for at least two databases using any benchmarking tool. (For example, ycsb)[0-10]

4. Your analysis of  any change sentiment that occurs over the time period that you have selected.[0-10]

5. Your forecast of the sentiment at 1 week, 1 month and 3 months going forward[0-10]

6. Presentation of results by making appropriate use of figures along with caption, tables, etc and your dashboard for your forecast.[0-10]

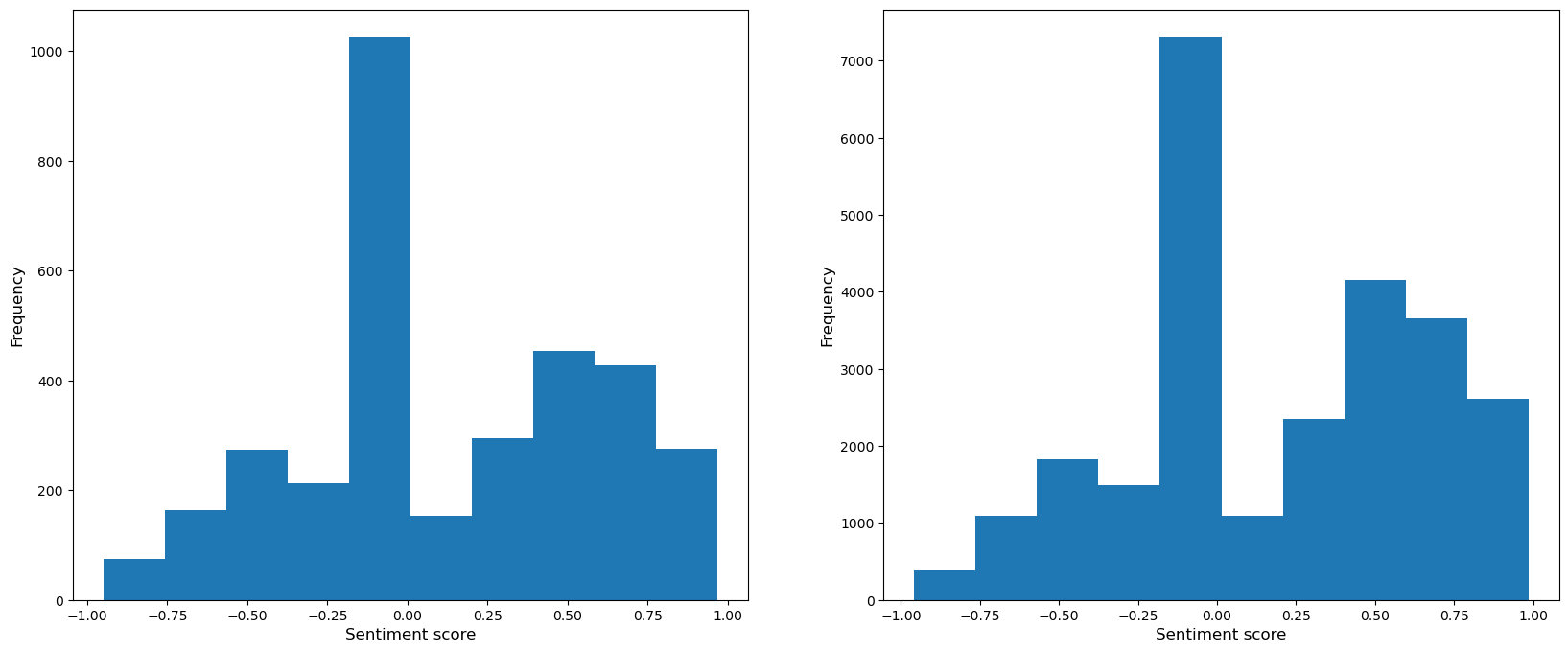
**Introduction**

**Method**

The first step was to do some initial EDA to start to understand the content and shape of the data. With 6 millions tweets, and 6 variables, there was no missing data, though there were also no queries associated with any of the tweets. Also, many of the dates at which the tweets were recorded were not unique, as the field only recorded to the nearest second, while milliseconds or even smaller denominations may have be required to distinguish tweet submission time. Finally, the username field was not thoroughly examine, because it was not going to be used in this analysis of tweet sentiment over time, but it was found that there were a number of users who regular tweeted. It was found that the tweet ID was, unexpectedly, not unique. This was because some of the tweet entries were duplicates. The mapper was used to remove these duplicated values by selecting only unique tweet IDs, leaving 1,598,315 unique tweets. This course of action was taken because it was assumed that the tweet duplication was a mistake. If these tweets had been posted twice deliberately - maybe for emphasis - they arguably should have been retained, but in that case the tweet ID would have been unique, so they were not.

The MapReduce program used took the raw data and returned only the date and punctuationless tweet. A complication was that the tweets themselves often contained commas, and the mapper – programmed to split the data on a comma - was truncating those tweets with commas. To resolve this, a for loop in the mapper acted on any line longer than expected (i.e. any line that had been split into more parts due to the presence of a comma in the tweet) and concatenated the tweet segments together to form a single output tweet. The mapper returned the tweet number, the date, and the un-edited (other than removed commas) tweet. The reducer took this output and stripped all punctuation from it (i.e. removed all non-alpha characters), then returned the tweet number, the date, and the tweet, and output them to a comma separated file. The reason that the tweet letters were not all changed to lowercase was because the VADER model used to determine the sentiment distinguishes between capital and lowercase (Beri, 2020), and the case can impact the sentiment. VADER also assigns meaning to punctuation, so it may be advisable to leave the punctuation, or only remove aspects of it, such as ellipses or tildes, and leave more potentially meaningful ones such as exclamation marks, or question marks. However, punctuation was removed here to demonstrate a reducer programme. Stopwords, however, were not, to avoid removing context, especially as VADER can understand that adverbs, such as ‘not’, can change the sentiment of a word (e.g. ‘happy’, vs ‘not happy’) (Beri, 2020). The reducer output was used for sentiment analysis (after renaming to add an extension, .txt, to allow it to be read by pandas in Python).

The VADER model scores the sentiment of each word in a piece of text, and sums them to determine the sentiment f the text as a whole (Beri, 2020). Capitalisation and punctuation are considered by the model, for example, the word ‘Happy’ gives a score of 0.5719, while ‘Happy!’ gives 0.6114. It is part of the NLTK package, one of the most commonly used Python natural languauge processors, and is simple and fast (DiBattista, 2021). As a commonly used, reliable, and easy-to-use package, it was selected for use here. A score was determined for each tweet, then the daily mean sentiment score was calculated. A daily mean was selected as the unit of time, as it was considered that an hourly mean may invite inherent bias (for example, people tweeting as soon as they wake up, or in the evenings, in different parts of the world may be reasonably expected to be more negative in areas and regions affected by war, for example). Also, a daily mean would be less likely to be impacted by missing values, and would usually be expected to contain a relatively large dataset, with a spread of sentiment scores approximating a statistical distribution that could be modelled. For example, the 4th of June 2009 saw 3354 tweets, and the 10th of May 2009 saw 25,984 tweets. Both sentiment score distributions appear bimodal (non-parametric) as shown in figure 1.

Figure 1: Two daily sentiment score distributions showing non-parametric nature of the data.

**Discussion**

**Results**

**Conclusion**

Beri, A. (2020). *SENTIMENTAL ANALYSIS USING VADER*. [online] Medium. Available at: <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>.

DiBattista, J. (2021). *The Best Python Sentiment Analysis Package (+1 Huge Common Mistake)*. [online] Medium. Available at: https://towardsdatascience.com/the-best-python-sentiment-analysis-package-1-huge-common-mistake-d6da9ad6cdeb.

‌

‌