CCT College Dublin
MSc Data Analytics : Time Series Analysis of average Tweet sentiment

Milo Moran - sbs23081

10/11/2023

1 Introduction

For this project, the sentiment of a dataset of tweets will be analysed, and the average sentiment across timesteps will be used to create forecasts of sentiment into the future. Big data storage and processing will be incorporated into the analysis, and advanced data analytics techniques will be used Python has been chosen as the language for this project, due to its flexibility, ease of use, and all the useful libraries that can be used with it. Jupyter notebooks are used due to their ability to run subsections of code independently, allowing for more transparency of what is happening.

2 Big Data Storage and Processing strategy

The important criteria for environment and database selection are dependent on the nature of the task and dataset at hand. In this case, the dataset is not especially large and is not dynamic, the queries for analysis do not need to be performed in real time, and the importance of execution time is a matter of minor convenience. This makes this task different from many other projects that make use of distributed computing technologies in that scalability, latency, and throughput are not hugely important metrics for consideration. Eliminating these leaves other metrics such as cost, ease-of-use, and the degree of integration with other tools, all of which still apply to this project. In terms of minimizing cost, only free to use technologies will be considered.

2.1 Environment selection

The decision for environment among open source free technologies with high degrees of integration is between Apache Spark and Apache Hadoop MapReduce. Spark is built on top of Hadoop and is designed for performance. (databricks No Date) Many of the advantages of each technology over the other are listed in the following article (IBM 2021), but none are hugely relevant to the problem at hand. Spark may be faster but as outlined this is a minor concern. Hadoop MapReduce methods are simple and easily managed so they will be used.

2.2 Database selection

One consideration for the DBMS used is the type of storage. Based on the nature of the data, column based storage would make specifying the important text column in queries more efficient, whereas many relational DBMS read data row by row and would involve a significant amount of unnecessary operations. (MongoDB No Date)

There are in fact 2 types of "column stores" with some being non relational pseudo column stores that store and access "column families" separately(group A, including Cassandra and HBase) as opposed to relational systems that access columns separately(group B). The differences between these two groups and

Figure 1: Cassandra table populated with the tweets data

their significance are outlined in the following blog post (Abadi 2010). However, the post mentions that despite this difference, it is still possible to select only the relevant subset of a row corresponding to a given column using group A systems, which is the desired ability for this project. Cassandra is appropriate to take advantage of this design aspect

In Cassandra, each table is equivalent to an independent column family. So to perform MapReduce tasks on the text column alone without making wasteful reads to the other columns, it would need to be stored in its own new table, separate from the rest of the data. Unfortunately, this is not actually a possible task in Cassandra. There is no capacity to SELECT a subset of columns to import into a column. To make use of this feature, the data would need to be pre-stripped.

2.3 Cassandra implemenation

A table was successfully created and populated in Cassandra. From here it can be queried using the CQL language, and benefits from all the advantages of Cassandra such as no single point of failure storage. As mentioned before it could be beneficial to upload the columns of the data to Cassandra separately. There were issues copying the data into Cassandra, likely related to issues with commas. There were still very many tweets in the dataset, and for the purposes of investigating word frequency, it was deemed sufficient for the task, as the goal is exploration rather than being results orientated. However when creating the new CSV using Cassandra, with new delimiters to reduce errors with commas, 1599598 rows were added to the file. This suggests that by some mechanism most of the missing rows were reintroduced, or never went missing.

2.4 Hadoop Streaming and MapReduce

MapReduce methods were used to investigate the word content of the tweet data. The output file from Cassandra was loaded into the Hadoop system. A

mapper file was written, and the was run with the data file in a hadoop streaming job. The mapper file separated all words in all tweets, removed punctuation and stopwords, and produced a list of word frequencies. Demonstration of several of the steps involved can be found in the Appendix. A file was also written that did not remove the punctuation, and was much bigger in size, with a lot of 'words' made up of random strings of punctuation. It was revised to avoid these, reducing the output from 19M to 12M words. This involved processing a very large number of words, and was quite computationally intensive. Here, the distributed system of Hadoop streaming split the computation into blocks and significantly lowered the time for completion, demonstrating the benefits of using such MapReduce technologies. It would have been beneficial to perform other similar tasks here, such as only selecting words that occurred more than 3 times to reduce the size of the output, or listing the top 100 most frequent words for inspection.

2.5 Database comparison

Two databases had to be selected, tested, and compared. While metrics specific to the project at hand should be prioritised, using a given database, other more traditional metrics for database performance will also be tested for and considered. Two types of database management systems(DBMS) stand out as warranting comparison, relational databases, like MySQL, and non relational NoSQL databases. YCSB is a software explicitly created to compare the performances of different databases and quantitatively describe the trade offs between each.YCSB contains 5 workloads that correspond to tasks one might perform using the database being tested. For comparisons related to this dataset, methods using the Latest, or Uniform records would be best, as tasks are likely to be based on selecting all tweets, or the latest tweets from a dataset like this. There is no strong reason to update existing tweets, so it's possible that custom workloads with 100% read operations and the Latest and Uniform distributions would be best for comparison. In terms of statistics to compare, an important one for this project would be resource utilisation, as it is being run on a user hosted VM. Standard measures like Latency and thoroughput would also be important to show how databases would scale with more data

3 Exploratory data analysis

Knowing that the data would be processed for sentiment, and contained usernames of accounts, average sentiment per user was selected for investigation. About 60% of the tweets are from accounts with more than one tweet in the data set. Further investigation will attempt to answer the question, "Are a small proportion of accounts responsible for a disproportionate amount of negative sentiment?".

In selecting a time step for the time series analysis, the daily value is an obvious choice based on the ease of separating the data along that line. Hourly

values may also prove quite useful. A significant discovery is that while the data contains tweets over a period, many of the dates during said period do not contain any tweets at all. To create a time series forecast with predictive validity, an appropriate imputation method will need to be used. Also, the amount of tweets for each day is found to be extremely variable. Internetlivestats puts the average daily tweets for the time period in the millions(internetlivestats.com No Date), this data set contains a fraction of that meaning that this data has been selected somehow from a greater number of tweets.

In setting the data to various timescales, it is found that at a daily scale there are 81 days with 34 missing(40%). On an hourly scale there are 1921 hrs total, with data for only 593, so 69% are missing. These are both extremely significant amounts of missing data, and choice of imputation methods will be vital to the accuracy of the forecasts.

An important question arises: "Has the method of selection introduced bias into the data?" Investigation as to the method of tweet removal will aid in answering this question. Assuming that sentiment of tweets is normally distributed, random reduction will be unlikely to affect the average timestep sentiment. Assuming that there is a temporal dependence of average sentiment, removal of tweets tweeted at certain hours will fundamentally change the nature of the average sentiment at the daily scale. For most other potential methods of tweet removal, there is no reason to assume it would influence the average sentiment, aside from explicitly filtering tweets for sentiment and corrupting data based on those results. This would make analysis of average sentiment very specific to the dataset itself and not representative of any greater whole. Also, non-temporal methods could not be adjusted for through analysis.

4 Sentiment analysis

There are many different libraries for sentiment analysis, all with different advantages. VADER is a package within the NLTK library, and is built especially for sentiment analysis with modern informal language and emoticons. This makes it ideal for dealing with the sentiment of the tweet data and is why it has been chosen to use for this project. While text pre-processing is a common task prior to sentiment analysis, VADER actually has not only the capacity to deal with things like capitalization and punctuation, it effectively makes use of them to contribute to the scores it returns (Geetha 2023). A short test was done that demonstrated the differences in compound score with and without pre-processing showing drastically different results in some cases, and the raw scores were used going forward (see appendix A). Note the results of the test could not be evaluated for unlabelled data, and the decision to use the raw scores is based on the capabilities and specialities of VADER itself.

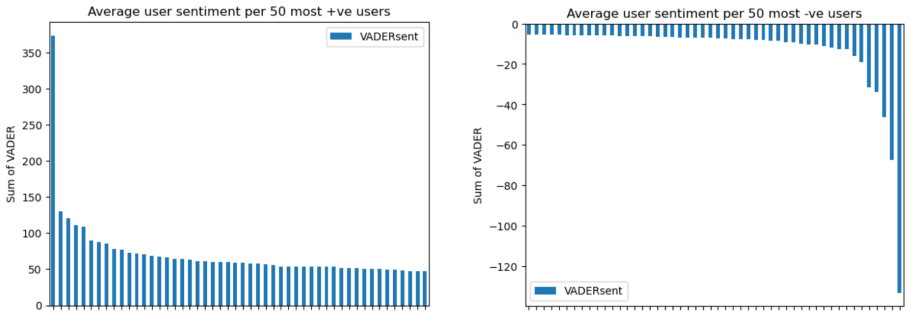

Figure 2: Distributions of sentiment for users with the most +ve and -ve sentiment

4.1 Sentiment by user

A small investigation was done on the dataset to determine if any information of interest could be gleaned from individual user contributions to the sentiment of the dataset. Certain users added very skewed sentiment to the dataset, both -ve and +ve. It was found that the accounts which had some of the most skew were accounts that posted the same tweet with a skewed sentiment again and again, bots as opposed to individuals who were singularly cheerful or dismayed. The ends of the distribution of sentiment between users was selected for display, with the centre being an extremely long bridge between the two. The patterns here shown alongside all of the other values would not be visible otherwise.

4.2 Sentiment by hour and imputation justification

From the EDA the hourly sentiment was determined to be worthy of investigation. The two following graphs produced show the average sentiment of hours included in the data. Both graphs are included to show the data at very high detail and resolution, both with and without empty hours, as displaying these differences are critical to understanding how much data is missing, and ensure data integrity. The date labels are minimised and the legends are removed to reduce non-data ink.

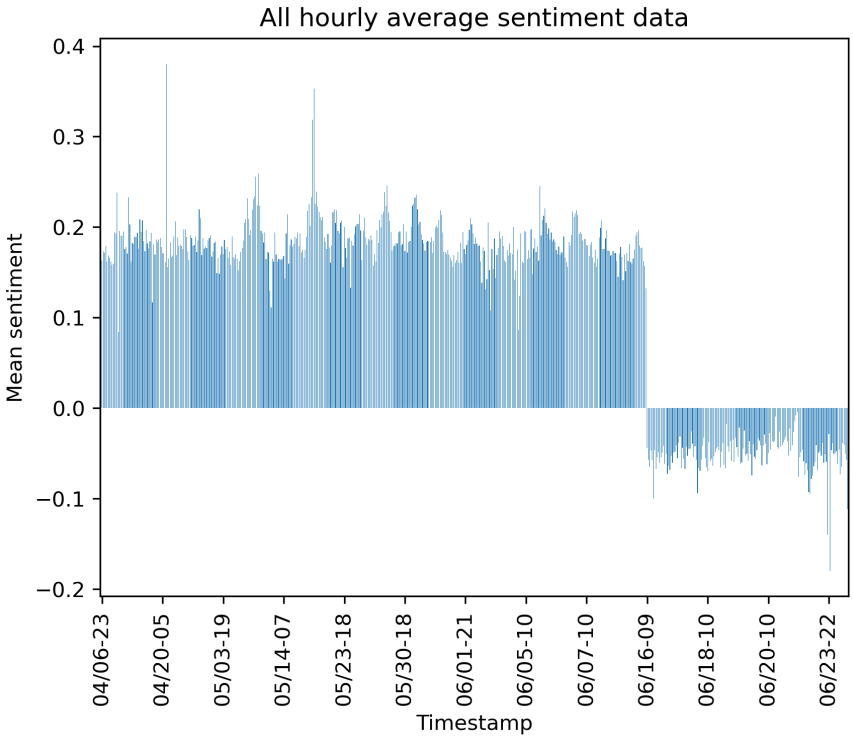

Figure 3: Sentiment of the entire time range by hour. Note that time intervals are not linear

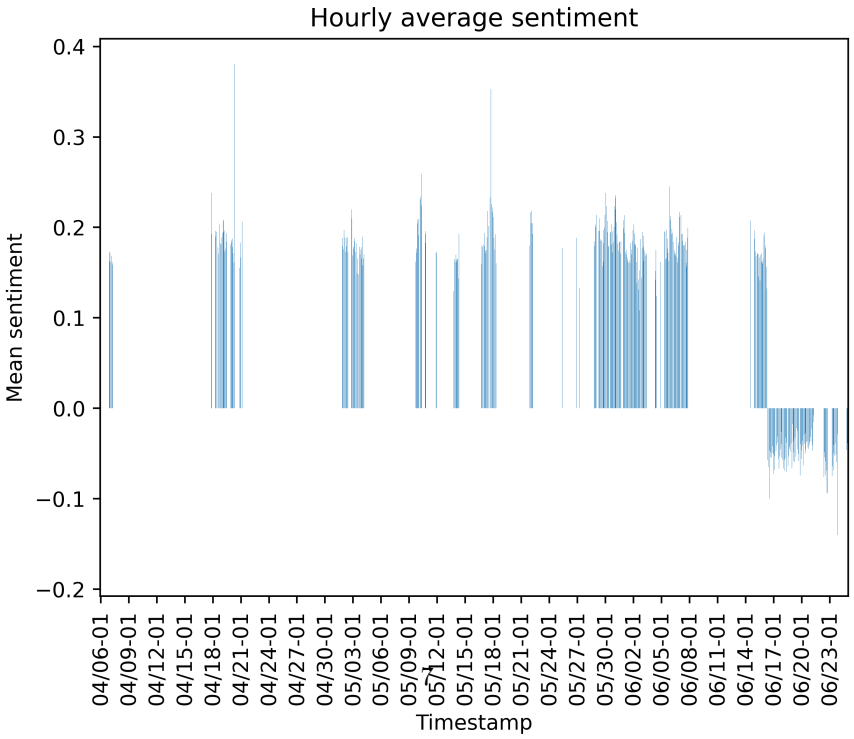

Figure 4: Hourly sentiment of whole time range at 3-day intervals

Figure 5: Hourly average sentiment for entire date range with and without blank hours added

At varying levels of resolution they both show a complete inversion of mean hourly sentiment at $\tilde{8}$AM on the 16th of June. While the sheer volume of missing data means that the decision of method for imputing missing data was always going to be important, the complete inversion of the sentiment warrants separate treatment for the two halves of the span. It also calls into question the accuracy of any predictions made with a time series that completely inverts in sentiment towards the end. While the daily fluctuations will not be massively important for the predictions 1 week, 1 month, and three weeks going forward, there is a lot of value to be gained from knowing what hours are missing, and if there is a related pattern. Upon inspection of the positive average sentiment domain, the hourly spread is notably not even, with fewer hours between 11am and 7pm PST being included. Also, an hourly fluctuation in sentiment has been found of about 16%. Assuming these represent the actual patterns of the daily tweet sentiment that will be imputed, and that the missing hours are missing at random, these findings show a bias in the data towards containing hours that have slightly higher average sentiment. This will skew the daily average values higher than they would be were they not missing any hours. As a result, imputation methods for this domain should account for this. There was no evidence of the same skew in the negative sentiment domain. These findings will heavily influence choice of imputation method.

5 Imputation

Based on the information gained, there is no missing data in the -ve domain, and there are two potential options for imputation in the +ve domain. One is to attempt to impute at the hourly level, using some kind of method that accounts for the hourly pattern for each day. However this method is hindered greatly by several mechanisms related to the amount of missing data relative to the daily scale. Larger amounts of missing data can lead to further increased bias, loss of information, and inflation of accuracy all due to misrepresentation of the real trends and seasonality of the data. The daily timescale is still affected by these mechanisms but to a much smaller degree, and so was decided upon. The advantages gained by imputing at the hourly level would be in representing shorter term fluctuation, not weekly or monthly patterns as required for forecasting, so the loss of this information is not hugely significant. What's more, the bias for the selection in favour of more positive tweets over the day can be accounted for, by altering an imputation method to slightly reduce the expected value. An attempt was made to find the amount of this difference from the difference between actual and a theoretical expected even sample, but the assumption that the relative frequency was independent of the hour count could not be proven. As a result, an arbitrary 5% will be subtracted from imputed values in this domain to attempt to account for some the bias. Assuming that with almost 50% missing values, attempts to account for seasonality in imputation such as STL would be more likely to introduce false patterns to the dataset due to an insufficient number of complete cycles (Abulkhair 2023). After extensive testing

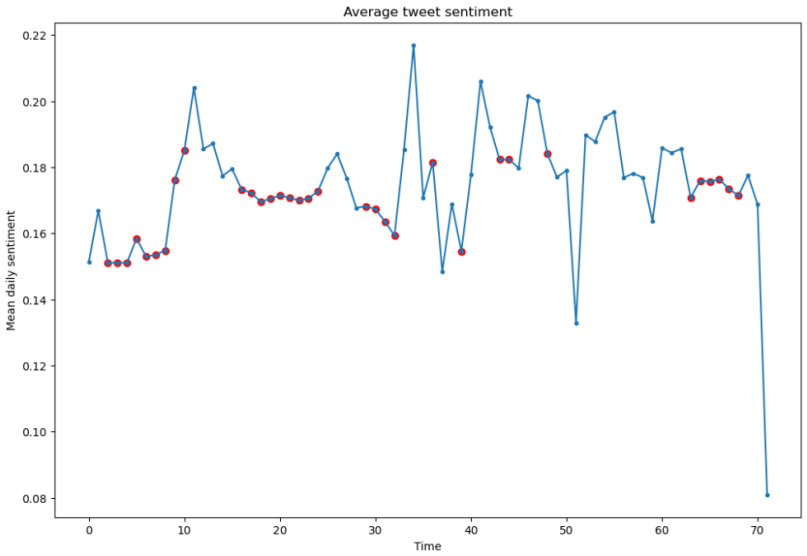

Figure 6: Imputed(red) and original data for the positive domain

of methods and hyperparameter tuning. A desired shape was found that would not add false seasonality and only dull existing seasonality without removing it. A double rolling mean replacement method was used with the 5% subtraction and some amount of backfilling from the negative shift values. The result of imputation at this level can be seen in fig 5, with colour used to demonstrate different kinds of data clearly, and a high data:ink ratio.

6 Time series analysis

6.1 Intuition of evaluation and forecasts

The most important aspect of this forecast is accounting for the inversion at the end. Any attempts to determine the 'accuracy' of a model by using the end values as a testing set will not be useful under the assumption that the mean shift is temporary. Therefore, quantitative accuracy will not be used as a metric for comparison between these forecasts, and instead, qualitative observations of plots will be used to justify them. To do this, an intuition of the data, and an expectation for the forecasts will be essential. Looking at the hourly average sentiment is important as it removes the impression of a downward trend that is present in the daily scale. The dataset on the hourly scale has two distinct stationary components about which are slight fluctuations due to seasonality or noise. While the pattern of daily sentiment is likely fabricated and has no basis in any real life underlying trends, this cannot be assumed for the purposes of this analysis. Therefore, the inversion must be assumed to be either

temporary or permanent. Both could be possible, and both lead to very different expectations going forward. In the case that it's permanent, a continuation of the mean of the second domain would be expected, but with the seasonality and residuals of both domains. In the case that it is temporary, a return to the first domain mean would be expected at some point, with more periodic or random inversions thereafter, depending on the length of the forecast. The fact that only a very small proportion of the dataset has the inversion points towards the likelihood that it is temporary, but is completely inconclusive. In the case that it is permanent the expected forecast can be found by subtracting the domain mean average daily sentiment from each domain to better find any seasonality, and then accounting for the negative domain mean for forecasts. This was explored briefly using a minimally tuned SARIMA model to accurately capture the patterns already within the dataset. Instead, methods will be used with the assumption that the reversal is temporary, and that trend mostly towards the positive domain.

6.2 Methods and implementation

There is a lot of evidence in forecasting that says that simple, methods consistently outperform more complex ones, mainly(Green & Armstrong 2015). However, this dataset contains a very significant inversion that may be too complicated for simple models to account well for, and machine learning models have increased significantly in complexity since 2015.

First, for a simple forecasting method, Holt winters Exponential smoothing was used. It allows for the capture of seasonality, and requires specification of seasonality period for the dataset. It also has three hyperparameters related to level, trend, and seasonality, with lower values giving more weight to earlier data and less responsive to recent changes, and higher values the opposite. The seasonality period was set to 30 days, as when set to weekly it was too heavily influenced by the false trend detected by the algorithm. Through trial and error, the parameters were all set, with very low alpha being the most important to capture the level earlier on in the dataset. Beta was minimally influential for low alpha, and gamma was selected to 0.37(= 81 days/30(one month)). As the prediction experienced drastic changes in level on a monthly scale, and it was deemed appropriate that if the forecast would contain more level shifts, they should be proportionally smaller in size vs the original data. The one month forecast can be seen in figure 6, the others can be accessed via the dashboard.

Secondly, and in less detail because this is the less likely possibility based on the ehort length of the negative domain, a SARIMA model was trained using a dataset created by assuming a permanent 'inversion' of the mean. Unlike the other assumption, this model would have benefitted from being trained using training and test set along with a hyperparameter optimiser attempting to increase accuracy, but this is outside the scope of this project. Instead minimal hyperparameter tuning was performed based on expected observations. Chosen parameters include a seasonal period m of 30 representing one month, and has p,d,q =2, 1, 2, and P,D,Q =3, 2, 0. A plot can be seen in figure 7. The significant

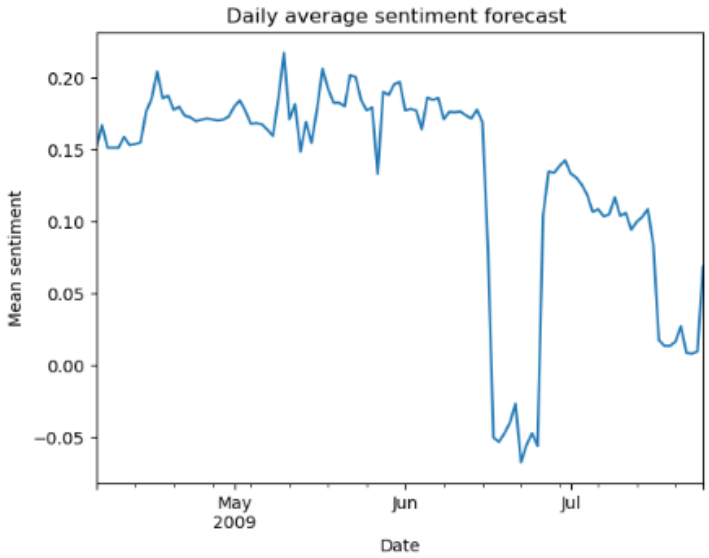

Figure 7: Forecast of the Holt Winters Exponential Smoothing model at one month assuming temporary inversion

negative spike shown is due to the method not accounting for the day there was both positive and negative hourly mean sentiments.

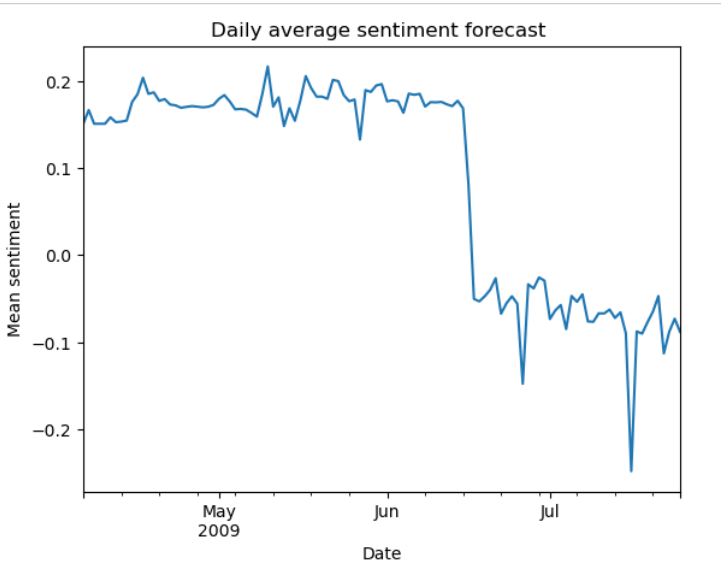

Figure 8: Forecast of the SARIMA model at one month assuming permanent inversion

7 Dashboard and forecast evaluation

To ensure the dashboard was able to integrate seamlessly with the forecasting, it was created inside a Jupyter notebook, with a widget to select various forecast depths(Appendix B). Once the dataset of sentiment is stored in the same folder as the file, it prompts the user to select the Run All option from the Cell menu, and the dashboard is created. The forecasts are available for a range of different time steps. The dashboard is minimalist, clear, and easy to operate and understand, there is no unnecessary information provided to the operator. However, the forecasted points are not distinct form the original data. This could be done using colour to delineate between them.

The forecasts are based upon manipulation of the hyperparameters to create something that would be expected given that the level shift is a rare occurrence. At a forecast of 90 days, 3 such inversions can be found, but with proportionally smaller amplitude as discussed before. There is still a significant trend to the forecast, likely from the downward trend in peak values that can be seen in the second half of the positive domain, and the spike in the negative domain is repeated for all the 'inverted domains going forwards'. As discussed before, evaluating this forecast has to be done based on assumptions about expected behaviour and the limited context and knowledge of the dataset. The downward trend stands out as not in line with those expectations, and is likely due to the model not being able to give equal weight to values at the start of the time series as the middle, thus overemphasising a the degree of trend. Weighting

distribution across the data frequently causes issues for small datasets such as this, and attempts were made to account for it through hyperparameter tuning, but they were not successful. Therefore the chosen model is limited in it's ability to deal with this particular data.

References

Abadi, D. (2010), 'Distinguishing two major types of column-stores', https://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of_29.html. Accessed Nov 10, 2023.

Abulkhair, A. (2023), 'Data imputation demystified — time series', https://medium.com/@aaabulkhair/data-imputation-demystified-time-series-data-69bc9c798cb7. Accessed Nov 10, 2023.

databricks (No Date), 'Apache spark', https://www.databricks.com/spark/about. Accessed Nov 10, 2023.

Geetha, L. (2023), 'Vader: A comprehensive guide to sentiment analysis in python', https://medium.com/@rslavanyageetha/vader-a-comprehensive-guide-to-sentiment-analysis-in-python-c4f1868b0d2e. Accessed Nov 10, 2023.

Green, K. C. & Armstrong, J. S. (2015), 'Simple versus complex forecasting: The evidence', *Journal of Business Research* **68**, 1678–1685.

IBM (2021), 'Hadoop vs. spark: What's the difference?', https://www.ibm.com/blog/hadoop-vs-spark/. Accessed Nov 10, 2023.

internetlivestats.com (No Date), 'Twitter usage statistics', https://www.internetlivestats.com/twitter-statistics/. Accessed Nov 10, 2023.

MongoDB (No Date), 'Types of database', https://www.mongodb.com/databases/types. Accessed Nov 10, 2023.

```
In [30]: dataf["raw"] = dataf["text"].apply(predict_sentiment)
         dataf["processed"] = dataf["clean"].apply(predict_sentiment)

         dataf
```

Out[30]:

	no	ids	date	flag	user	text	clean	raw	processed
0	1	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...	upset updat facebook text might cri result sch...	-0.7500	-0.4588
1	2	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...	kenichan dive mani time ball manag save rest g...	0.4939	0.4939
2	3	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire	whole bodi feel itchi like fire	-0.2500	0.0258
3	4	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....	nationwideclass behav mad see	-0.4939	-0.4939
4	5	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew	kwesidei whole crew	0.0000	0.0000

Figure 9: Test demonstrating Vader scores on raw vs preprocessed data

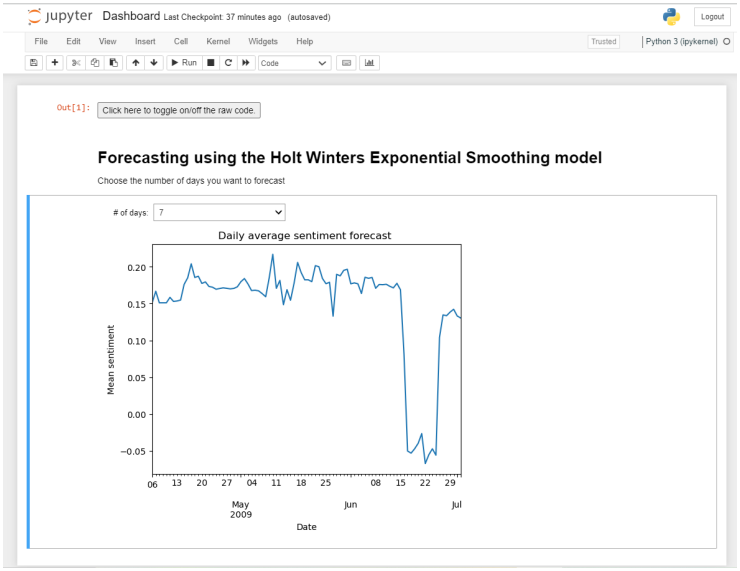

Figure 10: Dashboard after running all cells and adjusting widget

A Appendix

Figure 11: The mapper file used

Figure 12: A successful use of hadoop streaming

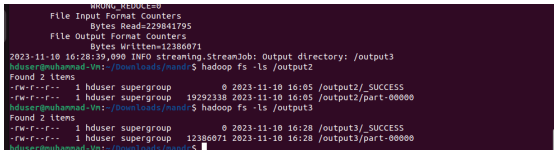

Figure 13: Comparison in size between the output files of two mappers

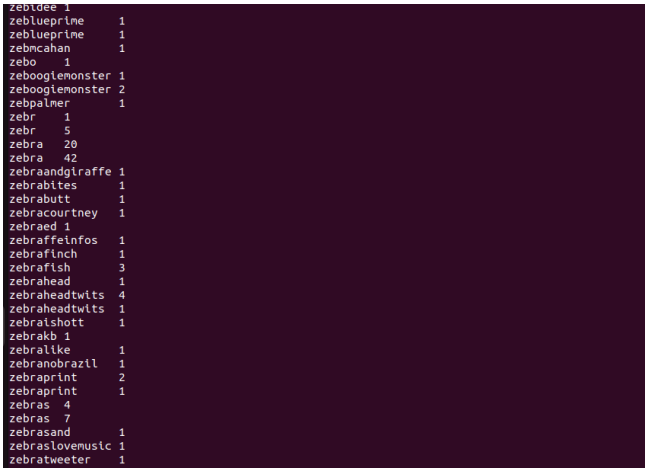

Figure 14: A subsection of the output showing Z words and their frequencies