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

Milo Moran - sbs23081

10/11/2023

1 Big Data Storage and Processing strategy

The important criterion 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.

1.1 Environment selection

The decision for environment among open source free technologies with high degrees of inegration is between Apache Spark and Apache Hadoop MapReduce. Spark is built on top of Hadoop and is designed for performance. (**?**) Many of the advantages of each technology over the other are listed in the following article (**?**), but none are hugely relevant to the problem at hand. Spark may be faster but as already outlined this is only a minor concern. MapREduce methods are simple, and so they will be used for

1.2 Database selection

Two databases needed to be selected, tested, and compared. While metrics specific to the project at hand will be used to justify 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 here, relational databases, like MySQL, and non relational databases such as NoSQL.

One of each will be tested.

One consideration for the DBMS used is the type of storage. Based on the nature of the columns in the dataset, it seems like column based storage would be quite beneficial to use. It would allow for the obsolete column of flag to be ignored, and would allow for separate treatment for the date, time, id, and text columns, whereas many relational DBMS read data row by row and would involve a significant amount of unnecessary operations. (**?**)

There are in fact 2 types of ”column stores” with some being non relational pseudo column stores that in fact 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 and significance of the differences between these two groups are outlined in the following blog post (**?**). 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.

HBase and Cassandra would both be appropriate to use 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

1.3 Implementation

Upon importing the tweets into Cassandra, it was seen that only 1.3M were loaded, and a a total of about 73 were skipped entirely due to errors in row length, likely related to issues between commas and and quotation marks. The limit on the number of rows was experimented with to see if timeout configurations would fix the issue to no avail. There is still many many tweets in the dataset, and for the purposes of investigating word frequency, it was deemed sufficient, as the goal is exploration rather than being results orientated However when creating the new CSV with cassandra, 1599598 rows were added to the file. This suggests that by some mechanism most of the missing 300k rows were reintroduced.

An attempt was made to upload the tweets data to a google cloud environment, which has the benefit of freeing up computing power for other task, but this approach was hindered by a lack of python interpreter

1.4 MapReduce

MapReduce methods will be used to investigate the word content of the tweet data. First, a mapper file will read in every line from the CSV, and output every word in the entire dataset along with a 1. Next, the reducer will sum all the occurences for each word. In this instance, the NLTK library will then be used to remove stopwords from the resulting word frequency pairs, leaving more interesting and noteworthy words behind. This will involve a very large number of words, (1.6M *mean tweet length), and will be quite computationally intensive. This is where the distributed nature of hadoop will split the computation and lower the time for completion, generally

2 Exploratory data analysis

Knowing that the data would be processed for sentiment, and contained usernames of accounts tweeting, the average sentiment for various users was isolated as a potential area of 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?”. This will be revisited as a point of interest.

In selecting a time step for the time series analysis, the daily value seems like a solid 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 some level of predictive validity, an appropriate imputation method will need to be used for the missing time units. 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(**?**), this data set contains a fraction of that. This means that this data has been selected or arranged from a greater number of tweets based on some criteria.

In setting the data to various timescales, it is found that at a daily scale there are 81 days with 34 or 40% missing. On an hourly scale there are 1921 hrs total, with only 593 present, meaning 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 Time series analysis

An important question arises: ”Has the method of selection introduced bias into the data? Has it influenced the average sentiment of given timesteps worth of tweets?” Investigation as to the method of tweet removal will aid in answering this question. Assuming that sentiment of tweets is normally distributed, any method of reduction will be unlikely to affect the average timestep sentiment. Assuming that there is a temporal dependence on average sentiment, removal of the tweets tweeted at sub timesteps(e.g. specific hours) will fundamentally change the nature of the average sentiment at the full timesteps(e.g. the days). 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.

3 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.

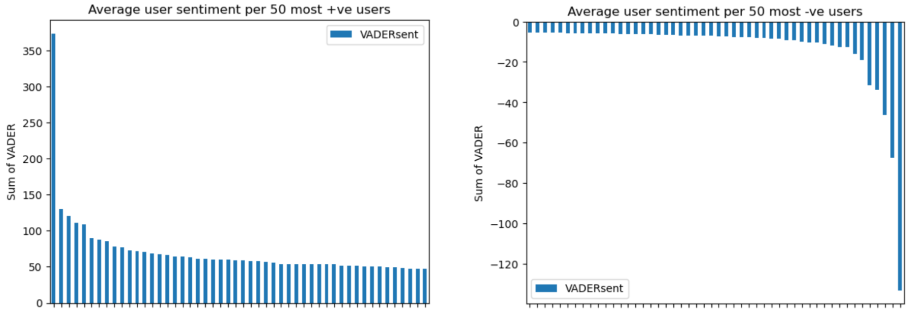


Figure 1: Distributions of sentiment for users with the most +ve and -ve sentiment

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 (**?**). 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.

3.1 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.

3.2 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.

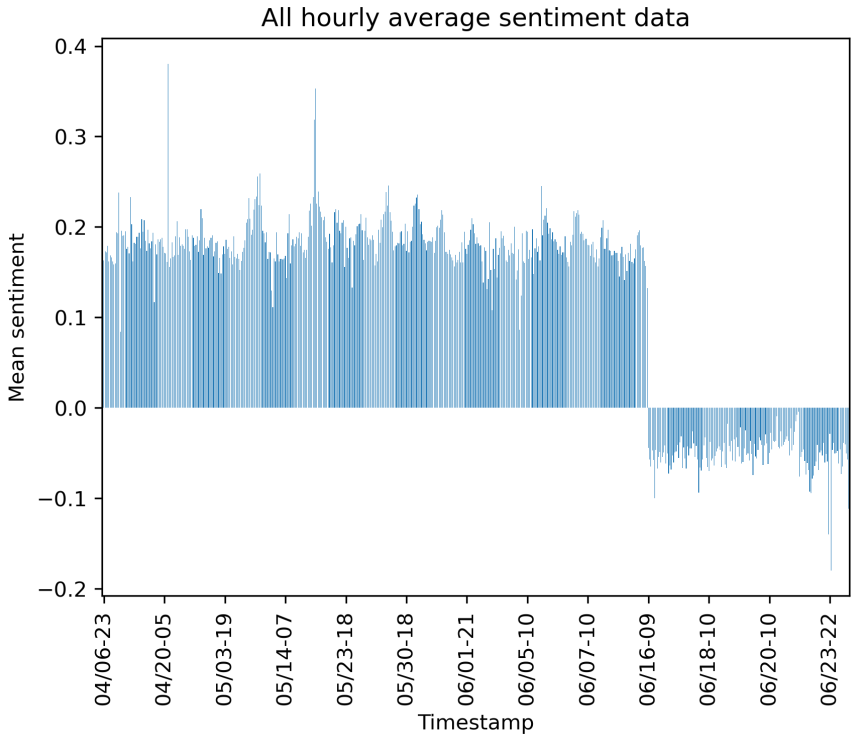


Figure 2: Sentiment of the entire time range by hour. Note that time intervals are not linear

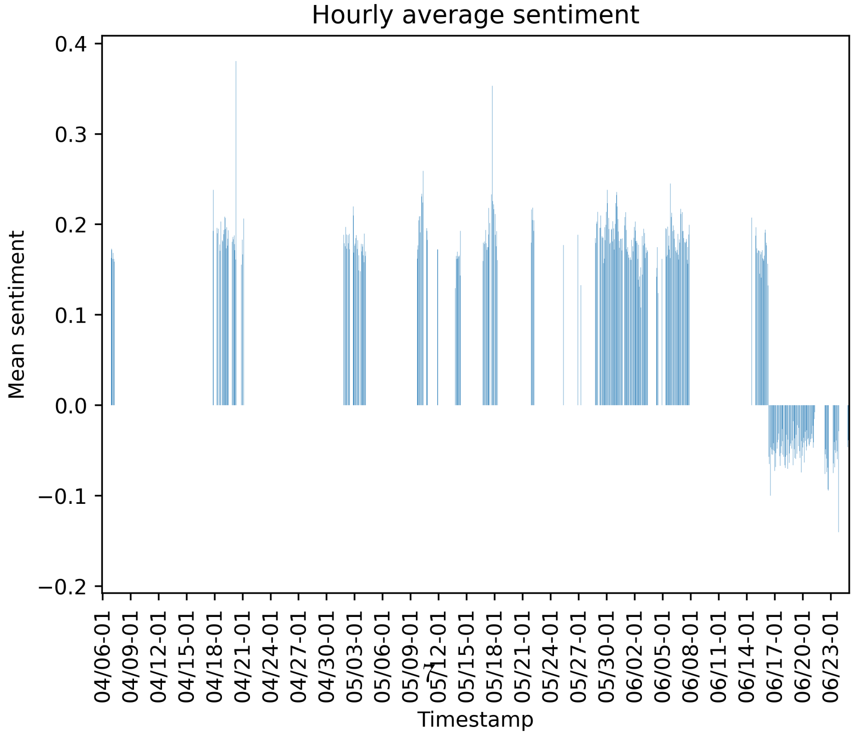


Figure 3: Hourly sentiment of whole time range at 3-day intervals

Figure 4: 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 timeseries 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.

4 Imputation

Based on the information gained up to this point, 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

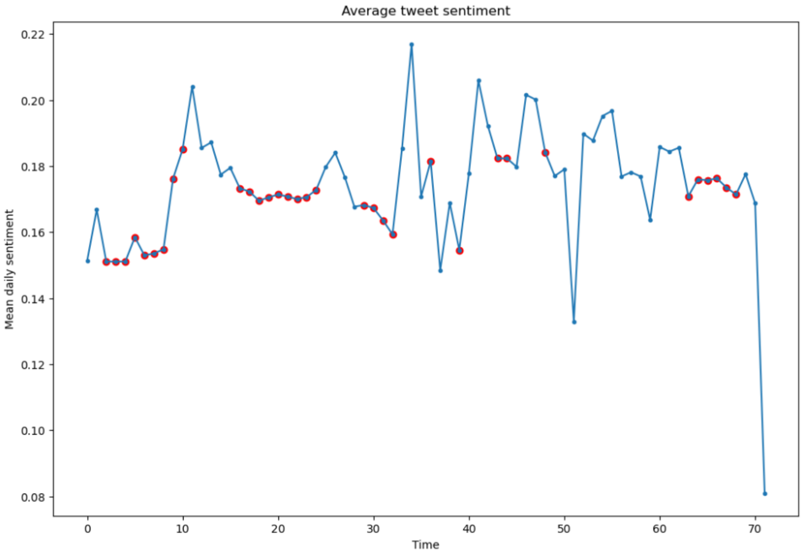


Figure 5: Imputed(red) and original data for the positive domain

(**?**). After extensive testing of methods and hyperparameter tuning based on a desired shape 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 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.

5 Time series analysis

5.1 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 inevitably not really demonstrate anything useful, whether the test set contains all of the inverted values or not. Therefore, quantitative accuracy will not be used as a metric for comparison between 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 could be found by subtracting the domain mean average daily sentiment from each domain to better find any seasonality, and then using the seasonality combined with the mean of the negative domain for forecasts. This could be interesting but will not be further investigated for this project. Instead, methods will be used with the assumption that the reversal is temporary, and that trend mostly towards the positive domain.

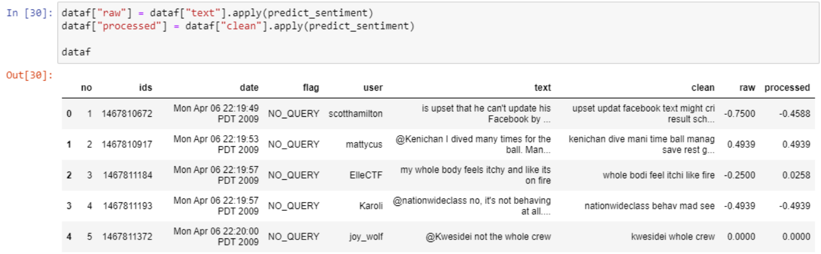
5.2 Methods and implementation

There is a lot of evidence in forecasting that says that simple, methods consistently outperform more complex ones, mainly(**?**). 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 forecastig method, Holt winters Exponential smoothing will be used. It will allow for the capture of seasonality, and requires specification of seasonality period for the dataset, as well as 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. through ttrial and error, the parameters were all set quite low

6 Dashboard and forcasts

References



```
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 6: Test demonstrating Vader scores on raw vs preprocessed data

A Appendix A