CCT College Dublin
MSc Data Analytics : Tweets

Milo Moran - sbs23081

2023

1 INTRODUCTION

1.1 Project Framework

1.2 Background

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.

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. Other time steps, like hourly, or three-day intervals, 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 extremely variable. Internetlivestats puts the average daily tweets for the time period between 2.5 and 35M(**?**), this data set contains a fraction of that. This means that this data has been selected from a greater number of tweets based on some criteria. The filtering based on this criteria has not been

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 completely randomly (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 is extremely unlikely but would make analysis of average sentiment useless. Also, non-temporal methods could not be adjusted for through analysis.

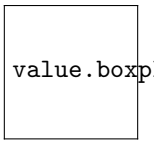

Figure 1: Boxplot of the values in the VALUE column

3 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.

3.1 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.

3.2 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 will need to be stored in its own table, separate from the rest of the data. The best way to do this is to import the whole database into a table, and then separate the column with the tweets into its own table. Upon importing the tweets, it was seen that only 1.3M were loaded by Cassandra, 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 limited on the number of rows was experimented with to see if timeout configurations in the

Code	Mann-Kendall	Trend	Dickey-Fuller	stationary	Shapiro-Wilk	normal
IRE	2.6e-05	y	0.007	y	0.547	y
DNK	0.57	n	0.13	n	0.431	y
CAN	6.2e-09	y	0.0	y	0.014	n

Table 1: Comparing p values of three tests for trend, normality, and stationarity

4 NLP and sentiment analysis

There are several 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 was chosen to use for this project. While text pre-processing is a common task prior to sentiment analysis, VADER actually has the 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 run that demonstrated the differences in compound score with and without pre-processing, and the raw scores were used going forward. Note the results of the test could not be evaluated for unlabelled data, and the decision to use the raw scores is based on knowledge of VADER itslef.

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.

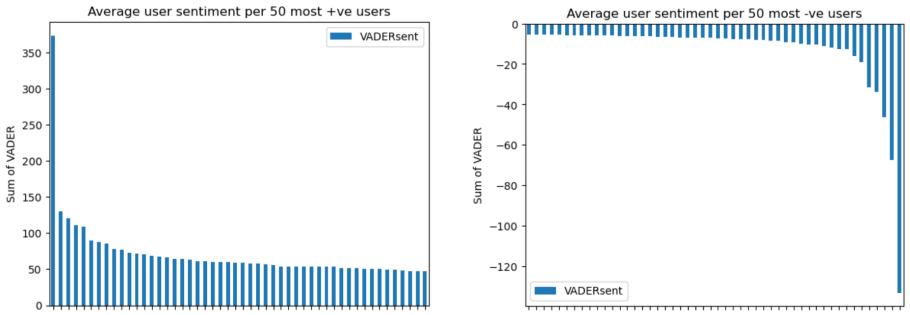

Figure 2: Distributions of sentiment for users with the most +ve and -ve sentiment

4.2 Sentiment by hour

From the EDA the hourly sentiment was determined to be worthy of investigation. The two following graphs produced show some interesting information about the data. 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 at 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 if there is a form of

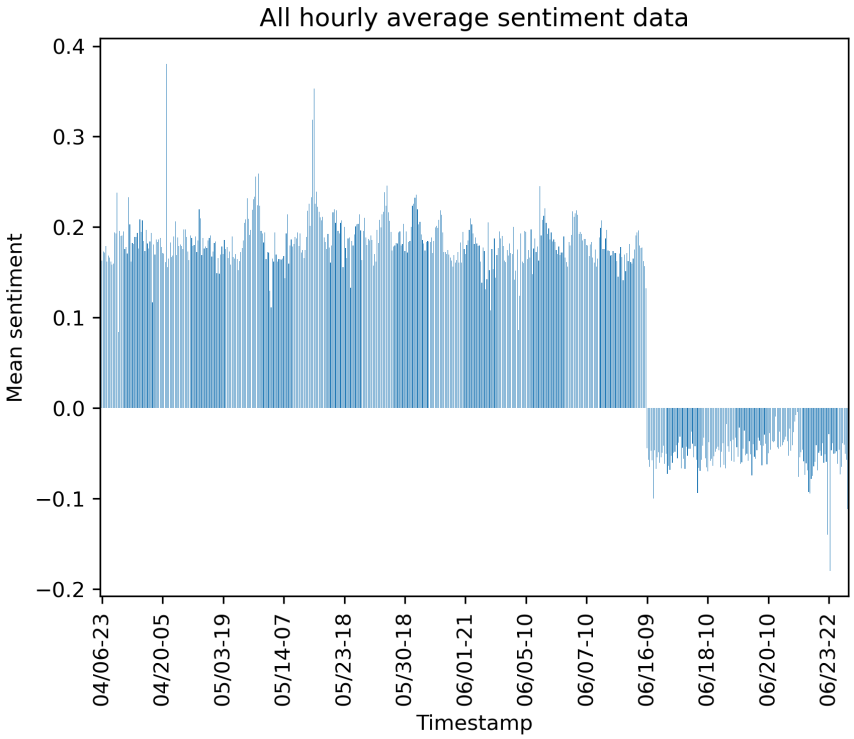

Figure 3: Sentiment of the entire time range by hour. Note that time is not linear

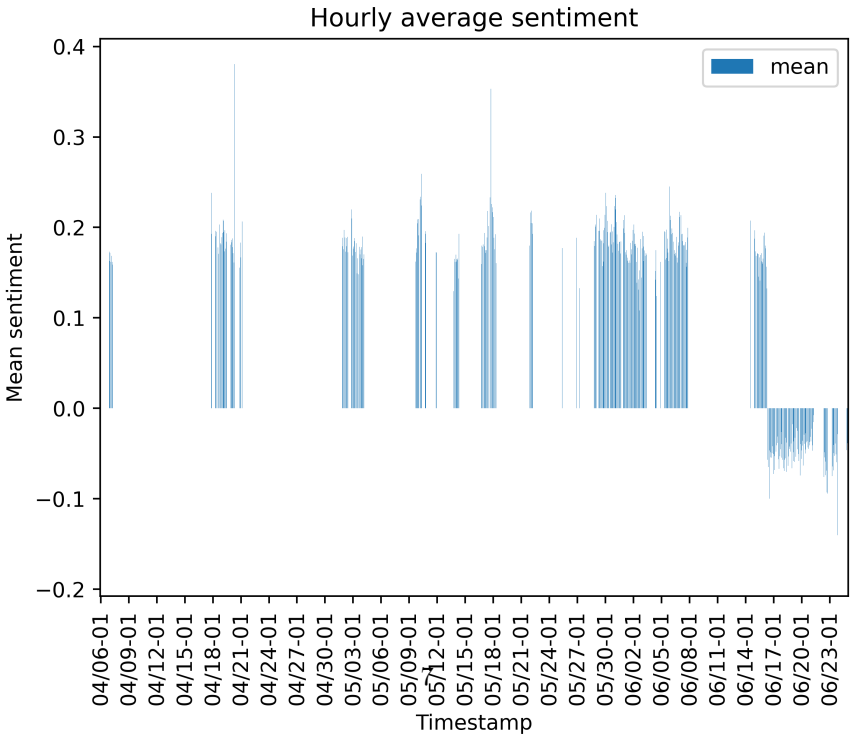

Figure 4: Hourly sentiment of whole time range

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

5 DASHBOARD

References

A Appendix A