**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | Advanced Data Analytics |
| **Assessment Title:** | [CA2 Integrated with Big Data, Individual Assignment 60%](https://moodle.cct.ie/mod/assign/view.php?id=128153) |
| **Lecturer Name:** | David McQuaid & Muhammad Iqbal |
| **Student Full Name:** | Conor Dillon |
| **Student Number:** | sba22446 |
| **Assessment Due Date:** | 26/05/2023 |
| **Date of Submission:** | 26/05/2023 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

CA2 – ADA & BD



Group ID - MSc in Data Analytics

Author: Conor Dillon

e-mail: [sba22446@student.cct.ie](mailto:sba22446@student.cct.ie)

Student ID: sba22446

Date: 26/05/2023

# Abstract

This project explored the utilisation of NoSQL through MongoDB Compass and Apache Spark via PySpark for handling, processing, and analysing the sentiment of a large Twitter dataset comprising over 61 million entries. The data, converted into a Spark dataframe, underwent sentiment scoring with VADER, and daily average scores were computed and graphed. Correlation of sentiment trends with real-world events and third-party polling results was observed. ARIMA (AutoRegressive Integrated Moving Average) was used for time-series forecasting to predict sentiment trends 1 week, 1 month, and 3 months ahead. These forecasts also indicated a partial alignment with the polling data. It was determined that the processed dataset was likely too short in time to produce a reliable time-series analysis. An interactive dashboard was designed to exhibit forecast outcomes, allowing users to switch between various forecast durations.

# Introduction

The topic of Big Data Analytics has experienced enormous transformation in the recent past. The sheer volume and variety of data generated through the usage of different technologies and platforms offer a unique insight into public opinion and trends in sentiment.

In this project, a large Twitter dataset to track the sentiment on the topic of Donald Trump. The former president is an influential and divisive figure and is known for his activity on Twitter up to his removal from the platform in 2021. The hope for this project is to acquire all Tweets mentioning Donald Trump in the year surrounding the vote in 2016 and his eventual inauguration in January 2017. The time series of this sentiment will be presented, and a forecasting will be performed to predict sentiment, 1 week, 1 month, and 3 months post the end of the dataset.

A hybrid data storage model such as SQL or NoSQL will be utilised to store the presumably very large dataset. The format of the data, the size of the dataset, and the expected query complexity will be the decision maker between which of these to use.

Apache Spark, a powerful analytics engine for large scale data processing is the choice for reading in the large dataset from either SQL or NoSQL. Spark’s in-memory computing abilities make it a good choice for processing such a large dataset.

Sentiment analysis will be conducted using VADER (Valence Aware Dictionary for sEntiment Reasoning). This tool will assign both the polarity and the polarity intensity to each tweet and will represent both as a single value ranging between -1 and 1. A time-series forecasting will be performed on the time-series of sentiment and will be presented in the form of a dashboard. A 1 week, 1 month, and 3 month forecast will be produced and will be presented on an interactive dashboard.

This project incorporates elements of Big Data, Machine Learning, and Social Media Analysis. It offers insights into the brutality of public opinion towards a public figure that garners a huge amount of both positive and negative attention. This project intends on examining this polarisation during the lead up and around the U.S. presidential election in 2016.

The analysis is performed on a Linux Ubuntu 22.04 VM through VirtualBox. The instance has an allowance of 4 cores, 4GB of RAM and 150GB of storage.

Code is backed up to Github (<https://github.com/ConorD-CCT/CA2_ADA_BD>).

Data will be included in zip file with submission.

# Data Storage, Processing and Machine Learning Justification

## Data Acquisition

At the beginning of the project, it was found that the free tier Twitter API had been shut down. As a result, a dataset would need to be found online concerning this topic. Such a dataset was found to be publicly available on Kaggle. The dataset *2016\_US\_election\_tweets.csv* and contains over 61 million entries. The format of this dataset would form the basis of the decision as to choose SQL or NoSQL as the data storage technology.

Exploring the data (Storm, 2020), one can see that the dataset consists of many columns associated with Twitter metadata. Looking at ‘candidate\_id’, we can see that there are four candidates (1 = Hilary Clinton, 2 = Donald Trump, 3 = Barack Obama, 4 = Bernie Sanders). Note that Barack Obama and Bernie Sanders were not final candidates and are likely included in this dataset supplementarily.

A picture containing screenshot, text, diagram, line

Description automatically generated

Figure Kaggle Dataset on USA Presidential Election 2016, (Storm, 2020).

As Donald Trump is the only candidate of interest for this project the Spark dataframe is filtered using the ‘candidate\_id’. This leaves over 45 million tweets remaining.

As the dataset already contains a means of filtering by a candidate label, it was determined that NoSQL was the appropriate database to use. This is since advanced querying was now not necessary, and the choice would go to the database that is superior at hosting a large amount of data. The unstructured nature of tweet content also bodes well for NoSQL.

## Storage & Processing

A NoSQL was implemented by using MongoDB Compass. This is a GUI (Graphical User Interface) for MongoDB and was selected due to its ease of use. One can simply upload a CSV to a MongoDB collection through the interface. This feature was particularly useful in this project as it allowed for the seamless uploading of the CSV file to a MongoDB collection. This process is straightforward and intuitive, requiring only a few clicks, which significantly reduces the time and effort required for data preparation. Then, the database was connected to Jupyter Notebook using the PySpark shell command line, including the MongoDB PySpark connector when launching PySpark. This automatically initiates a Spark session without having to configure one within Jupyter Notebook.

The combination of MongoDB Compass, PySpark, and the MongoDB PySpark connector provides a robust and efficient solution for storing, processing, and analysing the large Twitter dataset used in this project. This setup streamlined the data handling process but also allowed for more complex operations and analyses to be performed on the data, ultimately leading to more accurate and insightful results.

The data is read into a Spark dataframe using the spark.read() function. Here, the address of the Mongo database is included. At this point, the relevant columns are extracted using .select() – ‘candidate\_id’, ‘created\_at’, and ‘tweet\_text. After this, nulls are dropped and the dataframe is further filtered to contain only Tweets referring to Donald Trump, using to .filter() function to filter for a candidate\_id of 2.

It was found that there were tweets in the database that contained invalid dates. As such, .filter() is used to keep years between 1 and 9999. The final step of spark data processing is to group the dataframe by dates using the .groupBy() function. The new sentiment column is resolved as the average of the sentiment per day. The new dataframe is then ordered by date. The workflow of uploading and storing to MongoDB and processing within the Spark framework is now complete. The result is a time-series describing daily average sentiment in tweets that mention Donald Trump.

## Sentiment Analysis

Next, the VADER SentimentIntensityAnalyzer is instantiated. VADER was chosen as the sentiment analyser due to its robustness when working with social media content. *‘VADER was the most accurate and highest performing lexicon when put to the test with social media text from Twitter, and just as good (if not better) than its counterparts across the remaining domains.’* (Ray & Anreddy, 2021). It is efficient in handling vocabularies, abbreviations, capitalisations, repeated punctions and emoticons that are usually used on social media platforms to express sentiment. This makes it a good fit for social media sentiment text analysis (Zoumana, 2022). The re.sub() function is used to remove URL’s, ‘@’, and ‘RT’. Stop words are not removed as it was considered that VADER could deal with these. Two functions were written – one for the processing of the text and the other for returning the polarity of each tweet. The .withColumn() function is used to apply these functions to each tweet in the database.

# Comparative Analysis of Databases

The Yahoo! Cloud Serving Benchmark (YCSB) is an open-source framework developed by Yahoo! for evaluating and benchmarking the performance of numerous modern NoSQL and SQL database management systems with simple database operations on synthetically generated data (BenchAnt, 2021). These systems, which are designed to handle large-scale data across distributed environments, can have significantly different architectures and performance characteristics, making them difficult to compare directly. Given the importance of big data and the rapid growth of NoSQL databases, YCSB plays a critical role in helping organisations make informed decisions about which data serving systems best meet their specific needs in terms of performance, scalability, and reliability.

A comparative analysis of NoSQL (MongoDB Compass) and SQL (mySQL) has been performed using the YCSB. The databases were charged with two workloads – the standard ‘workloada’ as provided by YCSB and a modified version of ‘workloada’ called ‘workloada\_10000’. ‘workloada\_10000’ was modified to perform 10000 records and operations as opposed to 1000.

Figure 2 and Figure 3 present the results from a benchmarking comparison in terms of RunTime between SQL (mySQL) and NoSQL (MongoDB Compass). Results show that NoSQL is significantly faster when dealing with these workloads. These results also suggest that MongoDB is superior when scaling. A tenfold increase in record numbers in mySQL resulted in an almost eightfold increase in runtime, where it was only a threefold increase in MongoDB. It is this property of MongoDB, as well as the low requirement for complex querying, that led to the selection of MongoDB for this project.

Figure Database Benchmarking – RunTime - mySQL vs MongoBD

Figure Database Benchmarking – Insert Average Latency - mySQL vs MongoBD

# Change in Sentiment

Figure 4 presents the results of the sentiment analysis described in the previous section. The graph represents the daily averaged sentiment scores for tweets mentioning Donald Trump’s Twitter (@realDonaldTrump) on the lead up to and beyond the U.S. presidential election in 2016. In total, there are over 45 million tweets considered here.

It is clear that there is a heightened level of volatility in the timeseries from exactly the point that the results of the election became clear (November 9th). A 20-day and a 50-day moving average are used to clarify the trend of the data.

A picture containing text, screenshot, diagram, plot

Description automatically generated

Figure Daily Averaged Sentiment Scores for Tweets Mentioning Donald Trump

There are four distinct positive spikes in sentiment in the data. These are 11/9 (election day), 24/9 (Thanksgiving Day), 25/12 (Christmas Day), 01/01 (New Year’s Day). These spikes in positive sentiment align with high level public engagement by then President Trump with the American people, often associated with a video message. It is clear that these public engagements result in a net positive approval by the Twitter audience.

From around the election, there is a clear increase in positive sentiment trend, with the 20-day crossing over the 50-day. As time progresses, there is a clear negative shift in sentiment towards the end of the dataset, with the 20-day average passing below the 50-day average.

Comparing the sentiment trend determined in this analysis to one by Gallup (2021), there appears to be agreement that there was a declining trend in sentiment leading into 2017. An effort to find a Twitter dataset to further validate this was unsuccessful.

A picture containing text, line, font, screenshot

Description automatically generated

Figure Donald Trump's Job Approval Ratings Trend (Gallup, 2021).

# Sentiment Forecast

Due to the fact that the time-series introduced in Figure 4 is essentially a white noise time-series (each value is independent of the previous value and may have a mean of zero), it has been decided for the purposes of this exercise to use the cumulative sentiment value, Figure 6. It represents the build-up of sentiment over time where the slope of the graph is a truer representation of trend, not the absolute value. This aligns with Figure 4, where there is a clear change in sentiment around election day (11/9) and a descending sentiment score heading into March 2017. The forecasting will be done on this time-series.

A picture containing text, diagram, line, plot

Description automatically generated

Figure Cumulative Sentiment

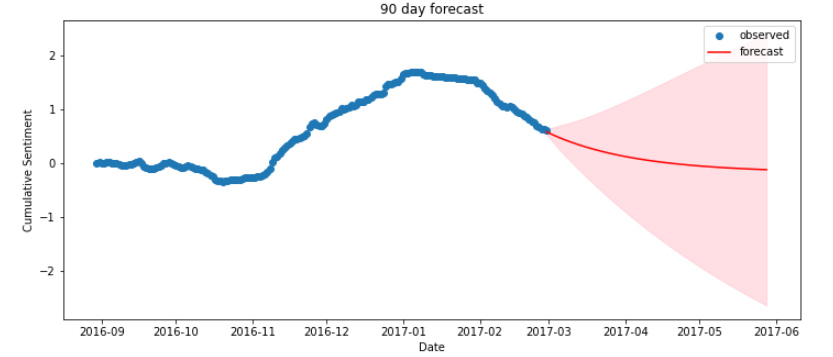
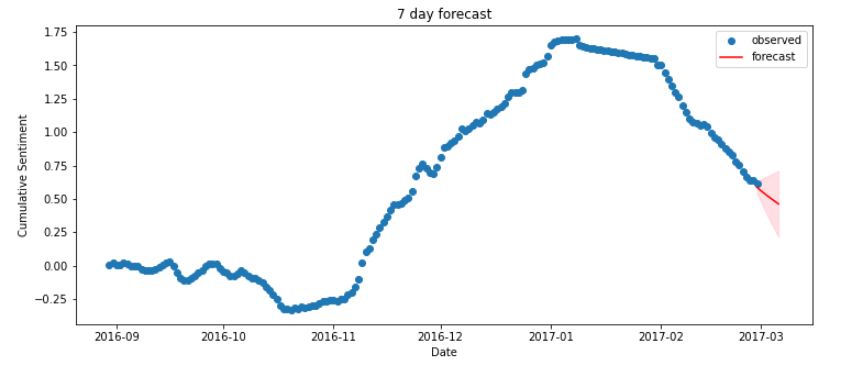
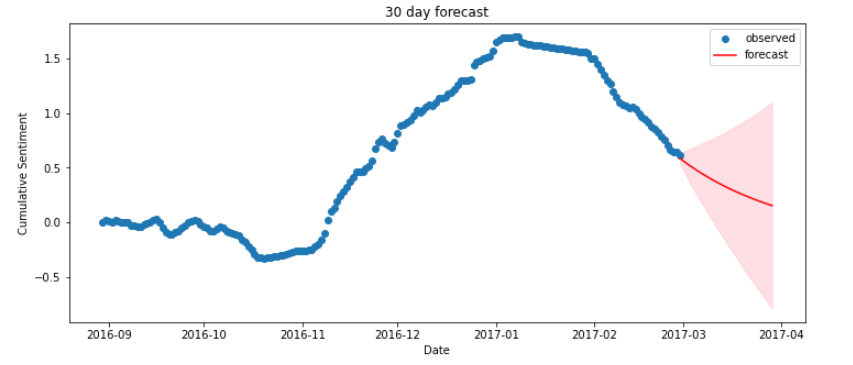
## ARIMA

ARIMA (Auto-Regressive Integrated Moving Average) has been used to predict 1 week, 1 month, and 3 months of cumulative sentiment data. ARIMA was chosen due to it’s flexibility and ability to work with no-stationary data. There is also no apparent trend in the cumulative sentiment time-series. However, this model is sensitive to the time-series consistency. Due to this, missing points in the time series needs to be accounted for. Interpolation was used to fill the missing points before fitting the model.

The ARIMA model has three parameters – p, d, and q. A GridSearchCV cross validation method was implemented in order to find the best combination. It was found to be (p q d) = ( 2 1 1). The figures below show the results of the time series forecasting.

Figure 7 show the results from the ARIMA modelling. The forecast is predicting a decline in sentiment until mid-2017 where it begins to level off. This is somewhat aligned with the poll data shown in Figure 5. It must be noted that there is low confidence here, with the interval showing a very large range for the longer-range forecasts. This is likely due to the low amount of datapoints in the timeseries and the complete lack of seasonality in the data. In reality, the outcome of the election and the actions of the president thereafter were and are wholly unpredictable. A model such as this should not be relied upon for to provide accurate results for this data.

Figure ARIMA 7-day, 30-day, and 90-day forecast



It is important to note that this model is univariate, in that it only considers the past values to try and predict. ARIMA can also work with multivariate data. Perhaps data on who made the tweet (male/female/democrat/republic/etc) could be used to improve the quality of the prediction.

# Dashboard

For the dashboard, it was not appropriate to command a run of the machine learning model every time the user wants to change forecast length. Instead, the Forecast section saves the results in multiple dictionaries that are then accessed by the dashboard itself. The dashboard is run from within Jupyter Notebook.

A picture containing text, screenshot, diagram, plot

Description automatically generated

Figure Interactive Dashboard for forecasting cumulative sentiment from tweets mentioning Donald Trump

# Conclusion

The project concludes after successfully demonstrating how to store, process, and analyse the sentiment of a twitter dataset originally containing over 61 million rows. NoSQl was employed through MongoDB Compass to hold the vast amount of information and Apache Spark through PySpark was used to process the information in Jupyter Notebook.

While stored as a Spark dataframe, the tweets were analysed in terms of sentiment scoring using VADER. The scores were grouped by day, and the average score was calculated. These scores were plotted over time and shifts and sentiment were identified and attributed to events that occurred in reality. The sentiment trends were shown to somewhat align with the polling results returned by a third party.

ARIMA was then used to perform a time-series forecasting (FB Prophet also used – available in the notebook). The forecast was performed and determined predicted values for 1 week, 1 month, and 3 months into the future. These also appear to somewhat align with the polling data. A conclusion drawn from the forecasting effort is that the dataset is too short in terms of time. A much longer dataset in time would increase the reliability of the prediction. Furthermore, the time-series itself is inherently non-stationary and is considered not to have any seasonality. The trend of the sentiment is heavily influenced by the actions of the individual who himself has proven volatile in action and in words.

Finally, an interactive dashboard was created to display the results of the forecast, with the user able to switch between forecast lengths.

# References

Gallup, 2021. *Presidential Approval Ratings -- Donald Trump.* [Online]   
Available at: https://news.gallup.com/poll/203198/presidential-approval-ratings-donald-trump.aspx  
[Accessed 26 May 2023].

Ray, T. & Anreddy, S., 2021. *Using vaderSentiment to Intuitively Predict the Sentiment of Social Media Posts.* [Online]   
Available at: https://ds3.ssrc.msstate.edu/2021/01/14/using-vadersentiment-to-intuitively-predict-the-sentiment-of-social-media-posts/  
[Accessed 26 May 2023].

Storm, D., 2020. *2016 USA Presidential Election Tweets-61M Rows.* [Online]   
Available at: https://www.kaggle.com/datasets/paulrohan2020/2016-usa-presidential-election-tweets61m-rows  
[Accessed 20 May 2023].

Zoumana, K., 2022. *Social Media Sentiment Analysis In Python With VADER — No Training Required!.* [Online]   
Available at: https://towardsdatascience.com/social-media-sentiment-analysis-in-python-with-vader-no-training-required-4bc6a21e87b8#:~:text=It%20can%20efficiently%20handle%20vocabularies,social%20media%20sentiment%20text%20analysis.  
[Accessed 26 May 2023].