A picture containing diagram

Description automatically generated

MSc in Data Analytics

**CA2**

Author: Mara Carcione

e-mail: sba22243@student.cct.ie

Student ID: sba22243

GitHub repository: <https://github.com/sba22243/ca2-sem2>

Words count: 2770

# Abstract

In this paper, we examine and forecast tweet sentiment about the covid vaccine over a year, from June 2020 to May 2021. We present some data on tweeters and estimate the sentiment of people using Twitter to discuss this topic.

# Introduction

Sentiment Analysis is a sub-fields of Natural Language Processing. This methodology has attracted the attention of many researchers as it allows them to investigate various problems on social media. The devastating spread of Covid-19 has created particular concern on the part of the world population. There have been conflicting views on social media about it. During the first months of the pandemic, medical science was researching, was dedicated, and continues to do so, to study the effects of the virus on people and how to counter and prevent its spread. In the second half of 2020, the first trials of various vaccines began, and even in this case, social media played an important role in guiding people's opinions. This project consists in carrying out a Sentiment Analysis in the period from June 2020 to May 2021, exactly between the period in which no vaccine was available and when the first vaccination started. The reference social media is Twitter, and the topic analysed is Vaccine.

# Data Analysis process

The Cross Industry Standard Process for Data Mining (CRISP-DM) represents the most common basic methodology used to standardise data mining processes in all sectors (Hotz, 2018). It includes six steps:

1. Business/Research Understanding Phase
2. Data Understanding Phase
3. Data preparation Phase
4. Modelling Phase
5. Evaluation Phase
6. Deployment Phase



Figure - CRISP-DM

1. Business / Research Understanding Phase is the essential phase that focuses on the objectives of the project and, therefore, on the determination of business objectives with a deep understanding of the customer’s needs. At this stage, it is important to determine the availability of resources by making a cost-and-benefit analysis. Finally, it is also very important to define the technical aspect of data mining, producing a project plan that selects the technologies to be used. (Hotz, 2018)
2. Data Understanding Phase is the phase of understanding the data in which the initial data are collected; the data are described by examining their properties, the data are explored by identifying their relationships and finally the quality of the data is verified to examine how dirty or clean this data is. (Hotz, 2018)
3. Data Preparation Phase prepares the final datasets for modelling. In this phase, the data to be used is determined. Then we move on to cleaning the data by correcting or removing incorrect values. If necessary, variables are transformed, and data are reformatted as needed. (Hotz, 2018)
4. Modelling Phase is the shortest phase of the project. The various models are built and evaluated. (Hotz, 2018) It consists of four tasks:
   1. Select modelling techniques - determines which algorithms to try
   2. Generate test design - split data into training, test and validation sets
   3. Build model
   4. Assess model - the data scientist interprets the results of the model applied based on knowledge of the domain
5. Evaluation Phase examines which model best suits the company through:
   1. Evaluate results

The next sections show the phases implemented to create the current report.

# Stage One - Determine Business Objectives and Assess the Situation

The year 2020 was an important year for the whole world. The pandemic caused by the Covid-19 virus forced the world to enter a global lockdown. The effects of the virus were unknown. The current research analysed the tweets from June 2020 until May 2021 to understand the user’s sentiment around an important topic such as vaccines. The period was chosen in order to analyse two specific periods where in the first part, several vaccines were under study and development, and in the second one, the first vaccine was available to the population.

# Stage Two - Data Understanding

## Database benchmarking

For this project, we used open-source software, Yahoo! Cloud Serving Benchmark(YCSB), which offers us the possibility to test the performance of different databases performing read and write operations. The YCBS system monitors various metrics and creates a report.

In the current project, databases were tested, MySQL and MongoDB.

MySQL is a rational database that uses SQL, while MongoDB uses documents in JSON format. MongoDB is a database that does not support SQL and has a different logic from relational databases. YCSB has six types of tests called workload. In the project, it was used the workload that performed 50% of insert and 50% of updates.

The test was done on a virtual machine with Ubuntu 22.04 provided to us by the lecturer. MongoDB and MySQL were then installed on that virtual machine.

The test was performed by executing the several times the following commands for both databases:

* **MongoDB**
  + ./bin/ycsb load mongodb -s -P workloads/workloada -p recordcount=1000
  + ./bin/ycsb run mongodb -s -P workloads/workloadb -p recordcount=1000
* **MySQL**
  + ./bin/ycsb load jdbc -P ./jdbc-binding/conf/db.properties -P workloads/workloada -p recordcount=1000
  + ./bin/ycsb run jdbc -P ./jdbc-binding/conf/db.properties -P workloads/workloadb -p recordcount=1000

For the purpose of the research, the recordcount parameter was 100, 1000, 5000 and 10000. Each output was copied into an Excel file, and then the relevant values were extracted. The result of interest was the Throughput measured in ops/sec.

The diagram below shows the results

A picture containing screenshot, text, line, plot

Description automatically generated

According to the graphic above, MongoDB performs better than MySQL when there are more than 5000 operation insert/update having a double throughput.

## Collect Data and Software Architecture

The dataset was downloaded from archive.org using the links provided in the assignment: The dataset was downloaded from the link  
 <https://archive.org/details/twitterstream?sort=-publicdate>

The structure of the dataset consist in several nested directory where the data were organised by year, month, day and hours. The tweets were stored into compressed json files. The compressor was bz2. The size of the whole dataset was more than 800GB.

The download of the files forming the datasets was made on a phased basis. The datasets were downloaded by month and uncompressed into the hd. The directory with the files was made available to the VirtualMachine with Ubuntu used for part of the research. Each month had a size between 30GB and 80GB.

For the phase of reading the json files and filtering the tweets according to the keyword of the chosen topic, i.e. “vaccines”, a Python file was created (read-filter-dataset.py). This file uses Apache Spark to read the files, unpack them, filter them and save them in a new json file, much smaller and with only the tweets of interest.

In the Python program, these steps are performed:

* Creation of the list with bz2 files to be analysed
* Reading of the file present in the list
* Filtering of the file considering only the English language and the presence of the keyword vaccine within the tweet
* Creation of the dataframe with the following columns: created\_at, retweeted, text, timestamo\_ms
* Writing the file to the filesystem

As was written above, this activity was made for each month.

We chose to use Spark for this phase because it provides a lot of dataframe management functions and has an internal job optimisation engine. It also supports parallel processing.

The json file filtered using the above program were then stored in MongoDB using Apache Spark. The import into MongoDB and the next activities were performed using Jupyter Notebook and will be described in the next paragraphs.

## Exploratory Data Analysis (EDA)

In this step, exploratory data analysis was performed. A Jupyter Notebook was created for this purpose and is named as “sba22243\_Integrated\_CA2\_sentiment\_step\_1.ipynb”.

In the EDA phase, the files json created previously were loaded into a Spark Dataframe

The dataframe is composed of 4 columns, names ad created\_at, retweeted, text and timestamp\_ms.

Created\_at is a string, retweeted is a Boolean, text is a string and timestamp\_ms is a string. In the dataframe there are no null values and the number of elements is 242125.

The retweets were filtered because they may not reflect the original sentiment of the author of tweet 1, in fact a user could retweet a negative tweet with a positive or neutron sentiment or vice versa. As a result, the inclusion of retweets in the sentiment analysis may cause inconsistency in the dataset (Sailunaz and Alhajj, 2019). The tweets that begin with “RT” as they are retweets as well have been removed.

From the tweets, the web address and non-ASCII characters were removed. The column retweeted was also removed because non needed anymore.

Furthermore, using the timestamp\_ms column, new columns have been created relating to the day, month and year in which the tweet was written.

A count of tweets is then made for each single day, obtaining a table as shown below

A picture containing text, screenshot, font, number

Description automatically generated

The following diagram shows the tweets per day:

**A picture containing text, line, font, plot

Description automatically generated**

At the end of this stage, the dataframe, now ready, was stored in MongoDB for further analysis.

# Stage Three - Data Preparation

The method used to classify the text by identifying the various subjects expressed there is called Sentiment Analysis. A text is classified as neutral, positive or negative, or with a score called polarity, which indicates the strength of the sentiment (Mathworks, 2022).

In this step, it was performed the preparation of the dataset for the Sentiment Analysis. Sentiment Analysis is the analysis of a text (word, sentence or document), which can be positive, negative or neutral.

The method used for the analysis was VADER (Valence Aware Dictionary and sEntiment Reasoner) because, being pre-trained it allows to obtain results much more quickly than other analysers. Furthermore, VADER is much better suited for the language used in social media, made up of short sentences or abbreviations. VADER has pros and cons. Among the advantages are the possibility to label the text data without labeling, the flexibility to change the threshold value and label the data and finally the advantage of reducing manual effort. Among the disadvantages, the accuracy of the analyses is sometimes not good because VADER cannot analyse the text that contains sarcastic tones.

For the sentiment analysis phase, the tweets were preprocessed before. A function was created for that. The tweet\_preprocessor function is very useful for preparing test tweets for sentiment analysis because it reduces word variety and eliminates noise.

At this stage, the tweets have all been placed in lowercase. A Lemmatizer has been used in order to reduce the dimensionality of the features and helps to keep the semantic meaning of the tweet. Stop words were eliminated as well.

Then with the sentiment analyser, the polarity was extracted for each tweet and stored into the sentiment\_compound column.

The researcher was able to classify every tweet and prepare the dataset for the modelling.

* Polarity>0.05: Positive
* -0.05<Polarity<0.05: Neutral
* Polarity <-0.05: Negative

The distribution of the sentimend is depicted in the below diagram

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

The distribution per month and year is shown in the pictures below

A picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing text, diagram, line, plot

Description automatically generated

# Stage Four - Modeling Phase

In this stage, the Time-Series analysis was performed in order to predict the sentiment for the requested period of time, such as one week, one month and three months.

The dataset was prepared for this analysis and brought into the accounts only the columns that are needed for the analysis. A resampling of the dataset was performed in steps at 1 day. The sentiment\_compound was calculated using the mean function.

The diagram below show the distribution of the polarity for the full dataset

A picture containing screenshot, text, plot, line

Description automatically generated

After the resampling, the polarity diagram for each day is shown in the picture below

A picture containing text, line, font, plot

Description automatically generated

With the resampling, now the diagram is much more clear, and it is possible to understand visually if there are missing values. In our case, there are a few intervals with missing parameters. The backwards-filling method was used for the purpose. To proceed further with the analysis, the +1 was added to the sentiment\_compound values in order to have only positive values.

Autocorrelation and Partial Autocorrelation analysis was performed. The diagram with the results is depicted below:

A picture containing text, line, screenshot, number

Description automatically generated

Multiplicative and Additive decomposition were calculated in order to understand the seasonality and patterns in the time series.

A picture containing text, font, line, screenshot

Description automatically generated

A picture containing text, font, plot, line

Description automatically generated

The test of stationarity was performed using the ADF and KPSS tests. ADF is a statistical test named Augmented Dickey-Fuller test and is used to test if a series is stationary. The following result was obtained

* ADF Statistic: -4.298732172575729
  + p-value: 0.0004464633342597592
* Critial Values:
  + 1%, -3.448748905151901
* Critial Values:
  + 5%, -2.8696473721448728
* Critial Values:
  + 10%, -2.5710891239349585

Based on the values obtained, we can see that the ADF statistic is more negative than the Critical values. The p-value is also very small. With this result, we can reject the null hypothesis of a unit root at any level of significance, and we can certainly conclude that the time series is stationary.

The KPSS is a statistical test named Kwiatkowski–Phillips–Schmidt–Shin and is used to test if a time series is stationary around a linear trend. The results obtained are:

* KPSS Statistic: 0.138466
  + p-value: 0.100000
* Critial Values:
  + 10%, 0.347
* Critial Values:
  + 5%, 0.463
* Critial Values:
  + 2.5%, 0.574
* Critial Values:
  + 1%, 0.739

In this scenario, we had KPSS statistics as 0.138466. This value is smaller than any critical value calculated. The p-value is 0.1 and is large. In this case, the null hypothesis cannot be rejected at any level of significance, and we can conclude that the time series is stationary around a linear trend.

Another important test regarding the forecastability of the time series was performed. A Sample Entropy was performed. The method is a modification of Approximate Entropy. This method has the advantage of being data length independent and an easy implementation. This test measures the complexity of a time series, and it is based on the probability that two parts of the series are similar. Lower is the probability, and mode complex or unpredictable is the time series. For this research, it was obtained a value of 0.7358. This means that the time series has a moderate level of irregularity or complexity.

At this stage, it was performed the prediction as requested in the assignment.

**Prediction for 7 Days**

The dataset was divided into train and test datasets. The window, as requested, was seven days. The diagram below shows the actual values that are considered for the prediction

A picture containing text, screenshot, plot, line

Description automatically generated

For the prediction phase, the ForecasterAutoreg was used. As a regressor, the RandomForestRegressor was used with six legs. This preliminary prediction returned the below result

A picture containing text, screenshot, plot, line

Description automatically generated

The result shows that the prediction is not so accurate, and further tuning is needed. For the hyperparameters tuning it was used the grid search forecaster. This function, perform several tests and find the best values for the prediction. For our 7 days analysis, the best parameters were identified. It was suggested that the lags was 10, the max depth was 3 and the number of estimators was 100.

The new RandomForestRegressor was now configured with the above values and the following prediction was made

A picture containing text, screenshot, plot, line

Description automatically generated

As we can observe, the prediction is much more better than before.

**Prediction for 30 Days**

The dataset was divided into train and test datasets. The window, as requested, was 30 days. The diagram below shows the actual values that are considered for the prediction

A picture containing text, screenshot, plot, line

Description automatically generated

For the prediction phase, the ForecasterAutoreg was used. As a regressor, the RandomForestRegressor was used with six legs. This preliminary prediction returned the below result

A picture containing text, screenshot, plot, diagram

Description automatically generated

The result shows that the prediction is not so accurate, and further tuning is needed. For the hyperparameters tuning it was used the grid search forecaster. This function perform several tests and finds the best values for the prediction. For our 30 days analysis, the best parameters were identified. It was suggested that the lags was 10, the max depth was 10 and the number of estimators was 100.

The new RandomForestRegressor was now configured with the above values and the following prediction was made

A picture containing text, screenshot, plot, diagram

Description automatically generatedA

**Prediction for 90 Days**

The dataset was divided into train and test datasets. The window, as requested, was 90 days. The diagram below shows the actual values that are considered for the prediction

A picture containing text, screenshot, plot, line

Description automatically generated

For the prediction phase, the ForecasterAutoreg was used. As a regressor, the RandomForestRegressor was used with six legs. This preliminary prediction returned the below result

A picture containing text, screenshot, plot, line

Description automatically generated

The result shows that the prediction is not so accurate, and further tuning is needed. For the hyperparameters tuning it was used the grid search forecaster. This function perform several tests and finds the best values for the prediction. For our 90 days analysis, the best parameters were identified. It was suggested that the lags was 10, the max depth was 5 and the number of estimators was 100.

The new RandomForestRegressor was now configured with the above values and the following prediction was made

A picture containing text, screenshot, plot, line

Description automatically generated

# References

1. Al-Garaady, J. and Mahyoob, M. (2022) ‘Public Sentiment Analysis in Social Media on the SARS-CoV-2 Vaccination Using VADER Lexicon Polarity’. Rochester, NY. Available at: <https://papers.ssrn.com/abstract=3647564> (Accessed: 22 May 2023).
2. Hotz, N. (2018) ‘What is CRISP DM?’, *Data Science Process Alliance*, 10 September. Available at: <https://www.datascience-pm.com/crisp-dm-2/> (Accessed: 25 May 2023).
3. Mathworks (2022) *Sentiment Analysis - MATLAB & Simulink*. Available at: <https://uk.mathworks.com/discovery/sentiment-analysis.html> (Accessed: 25 May 2023).
4. Sailunaz, K. and Alhajj, R. (2019) ‘Emotion and sentiment analysis from Twitter text’, *Journal of Computational Science*, 36, p. 101003. Available at: <https://doi.org/10.1016/j.jocs.2019.05.009>.