CoVacSenti – a Sentiment Analysis System for the COVID-19 Vaccine Based on Twitter

Anna Christina Weir

MSc Computing

Supervised by Dr Daniela Tsaneva

School of Computer Science and Informatics

Cardiff University

05/11/2021

ABSTRACT

The priority in the British COVID-19 pandemic management strategy is the administration of vaccines to the general public. While the majority of the population has received two doses, uptake was slower than expected and there is still a sizeable number who have not received any dose. Understanding how the vaccine is perceived by the public and what influences their opinion is essential in improving booster jab schemes.

This project created *CoVacSenti* – a sentiment analysis program to determine the attitudes of the British population towards COVID vaccination using Twitter data. Tweets across 19 months of the pandemic were processed using a Multinomial Naïve Bayes algorithm with 79.4% accuracy. The impacts of pandemic milestones on daily sentiments were analysed using a Fisher's Exact test, and a dashboard was created to display results.

The results showed that 55.6% of all tweets were negative, but the proportion of positive and negative tweets varied over time. Sentiments towards the AstraZeneca vaccine were lower than all other types of vaccine offered in the UK. The start and end of lockdowns and the vaccine rollout to the 40+ age band were pivotal events that caused significant changes in daily sentiment.

These findings are applicable to modern pandemic management and reflect the need for sentiment analysis algorithms to be specific to their contexts due to the polarity of key phrases and inaccurate conclusions that may be drawn from summarising findings over time.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr Daniela Tsaneva for her guidance and support, without whom I would not have completed such an enriching project. I would also like to thank Brooklyn and Leslie for their constant encouragement throughout this process.

CONTENTS

1 Introduction 1
2 Aims and Objectives 3
3 Background 4
3.1 Sentiment Analysis 4
3.2 Twitter 5
3.3 Vaccination Hesitancy and Refusal 5
4 Problem 7
5 Approach 9
5.1 Data Collection 9
5.2 MySQL Database 12
5.3 Text Pre-Processing 13
5.4 Sentiment Analysis Algorithm 18
5.5 Key Events 20
5.6 Dashboard Prototype 21
6 Implementation 24
6.1 Overview 24
6.2 Data Collection 25
6.3 Tweet Pre-Processing 28
6.4 Naïve Bayes Algorithm 31
6.5 Algorithm Accuracy and Improvements 38
6.6 Key Events Implementation 45
6.7 Dashboard Prototype 47
7 Products 50
7.1 Sentiment analysis algorithm 50
7.2 Dashboard 51
8 Findings 54
8.1 Sentiments Regarding COVID-19 Vaccination 54

8.2 Key Events Findings 56
8.3 Types of Vaccine 57
9 Analysis 60
9.1 Measures of Success 60
9.2 Sentiments Over Time 61
9.3 Key Events Analysis 62
9.4 Sentiments Towards Types of Vaccine 64
10 Conclusion 65
10.1 Achievements 65
10.2 Drawbacks and improvements 66
10.3 Future Work 69
11 Reflection 71
11.1 The Research Process 71
11.2 The Field 72
11.3 Personal Growth 72
12 References 75
13 Appendices 83
Appendix 1 : Details of key events 83
Appendix 2 : Dashboard test case 84
Appendix 3 : Results of Fisher's Exact Test for all key dates 86
Appendix 4 : Gantt chart of dissertation progress 87

1 INTRODUCTION

In the past two years, an acute respiratory syndrome caused by the novel coronavirus SARS-CoV-2 has been at the forefront of our global consciousness (Gorbalenya et al. 2020). Since its initial outbreak in Wuhan, China, in late 2019 (World Health Organization 2020a), COVID-19 has become a global pandemic with 239 million cases and 4 million deaths as of the 15th of October 2021 (John Hopkins University 2021). In the United Kingdom there have been over 8 million positive cases recorded since the start of the pandemic with over 100,000 deaths (UK Health Security Agency 2021).

A priority in modern pandemic management is the development and administration of an effective vaccine to grant populations immunity. After a global race in development, the UK was the first country to administer a vaccine to the general population outside of human trials in December 2020 (Aspinall 2020). The UK's National Health Service (NHS) then began a rollout strategy, vaccinating vulnerable groups and care staff before progressing through age bands of the population (NHS 2021). Since then, over 78% of the British population aged 12+ have received two doses of a COVID-19 vaccine, and vaccination efforts for teenagers and children are being planned (NHS 2021). There are currently four COVID-19 vaccines approved for use in the UK: the Moderna vaccine, the Oxford/AstraZeneca vaccine, the Pfizer/BioNTech vaccine, and the Johnson and Johnson/Janssen vaccine (NHS 2021).

However, there is a portion of the British population that remains unvaccinated. Vaccine hesitancy has been present since the first vaccine (Dubé et al. 2014) but, on the scale of a global pandemic, the impacts of unvaccinated populations could be catastrophic.

Understanding public perception is key to an effective vaccination strategy, yet typical methods of assessing attitudes (such as nationwide surveys) are time and resource consuming. Social media therefore presents as a viable alternative, with billions of users posting their opinions on a wide variety of topics. These qualitative posts differ from the scaled numerical data typically collected in surveys, and with very large datasets, manual processing is infeasible.

These conditions have led to the use of artificial intelligence, particularly sentiment analysis; the use of computational methods to extract and quantify subjective tone (Feldman 2013). In the context of social media analysis, sentiment analysis identifies opinions in written text and assigns numeric scores based on polarity.

This report will address the creation of *CoVacSenti*, a sentiment analysis system used to determine how the British population feels towards the COVID-19 vaccine, with the following structure:

Chapter 2: The aims and objectives of the project.

Chapter 3: A background of sentiment analysis, the use of Twitter in sentiment analysis, and a history of vaccine hesitancy.

Chapter 4: A description of the problem CoVacSenti addresses; how the background context relates to the project aims and objectives.

Chapter 5: The chosen approach to creating CoVacSenti, including motivations for choosing specific data collection methods, stages of analysis, and the creation of the final product.

Chapter 6: Details on how the approach was implemented, using code snippets and examples of processing outputs.

Chapter 7: A summary of the products of the project: the sentiment analysis algorithm and the dashboard

Chapter 8: The main findings of CoVacSenti in regard to sentiments about the COVID vaccine.

Chapter 9: An analysis of the results and how these fit into the current research in the fields of sentiment analysis and vaccine hesitancy, and the context of the COVID-19 pandemic.

Chapter 10: Concluding remarks; returning to the aims and objectives of the project, improvements that could have been made, and potential future work.

Chapter 11: A reflection on the dissertation process, including lessons learnt about research, the field of sentiment analysis and vaccine hesitancy, and personal skills.

2 AIMS AND OBJECTIVES

This project has the following key aim:

> To create CoVacSenti, a sentiment analysis system that can determine how the UK population feels towards COVID-19 vaccination through an analysis of their tweets.

This overarching aim will be achieved through the following specific objectives:

- Collect tweets regarding COVID-19 vaccination from the British public
- Create a sentiment analysis algorithm that is independent from existing sentiment analysis package solutions
- Investigate how attitudes towards COVID-19 vaccination have changed over time
- Examine how sentiments differ towards the specific vaccines - Pfizer, Moderna, AstraZeneca, and Janssen
- Evaluate whether key events in the COVID-19 pandemic influenced the public's sentiments towards COVID-19 vaccination
- Develop a dashboard prototype to display the findings of CoVacSenti

3 BACKGROUND

3.1 SENTIMENT ANALYSIS

Sentiment analysis is the use of computational methods to analyse a text and deduce the author's *sentiment* or *polarity*, the measure of subjective opinion (Feldman 2013). Sentiment analysis is a classification task in Natural Language Processing, as it involves representing polarity categorically as either positive or negative, or on a continuous scale to measure strength of polarity (Medhat et al. 2014). This can be conducted on a range of scopes; a document-level analysis can determine how an author broadly feels about a topic based on a long text form such as an article, while a sentence-level analysis extracts sentiments expressed in a single sentence.

There are in two main types of sentiment analysis, both with their own efficacy based on the context of their use. The first are *lexicon-based* methods, which compare text documents against databases of contextually relevant emotive language data using either *corpus* or *dictionary-based* methods (Medhat et al. 2014). Corpus-based approaches compare the syntactic and probabilistic patterns of a document with those in a database, while dictionary-based approaches make comparisons using the words in the document (Darwich et al. 2019).

The other main approach in the field is *machine learning* sentiment analysis; methods that extract the linguistic features of a document and use them to deduce sentiment with a variety of algorithms (Zhou et al. 2017). Machine learning can infer patterns from large datasets using either *supervised* or *unsupervised* methods. An algorithm is considered supervised when it is taught a pattern, either by being provided with a desired result or through reinforcement techniques. Meanwhile an algorithm is unsupervised when no guidance is given and it instead learns the pattern heuristically (Sathya and Abraham 2013). The choice of machine learning approach typically depends on the task itself; in this case – the classification of tweets as positive or negative.

3.2 TWITTER

Twitter is a social media platform where approximately 6000 microblog posts – or *tweets* – are posted every second (Sayce 2010). Tweets are a commonly used resource in sentiment analysis due to their abundance and availability, as they are publicly available as per Twitter's Terms of Service (Twitter 2020). Tweets are limited to a maximum of 280 characters each, thus allowing users to exchange short posts that are suited for sentence-level sentiment analysis.

Twitter is also an invaluable tool for sentiment analysis as the most popular tweets with the furthest reach on the website are those using emotive language (Stieglitz and Dang-Xuan 2012). This is regardless of the polarity of the emotion, reflecting that strong positive and negative sentiments are both expected and celebrated on the platform. Sentiments expressed on Twitter are considered accurate representations of the user's beliefs and intentions, as sentiment analyses of political tweets have been found to be strong predictors of election results (Ceron et al. 2013). Sentiment analysis therefore has applicability in a wide range of research topics, as findings can be used to collect, predict, and track the general population's attitudes towards polarising topics.

3.3 VACCINATION HESITANCY AND REFUSAL

Vaccination hesitancy is the reluctance to receive vaccinations for oneself or to administer advised routine vaccinations to one's children. This hesitancy has existed since the creation of the first vaccine and is much more widespread than vaccination refusal (colloquially known as *anti-vax*), despite not receiving the same media coverage (Dubé et al. 2014). Hesitancy represents the continuum of opinions on the topic rather than two binary categories of pro- and anti-vax. Attitudes towards vaccination are based on one's personal experiences, health issues, beliefs (religious, political, and moral), and exposure to conspiratorial communities and misinformation (Kata 2010).

Vaccination refusal began rising in the 1980's in response to the diphtheria, pertussis, and tetanus vaccine in childhood, which lead to a sharp spike in all 3 diseases across the United States after decades of decline (Neighmond 2009). Since the 1990's, the anti-vax movement has been emboldened by false claims that the measles, mumps, and rubella (MMR) vaccine

causes autism. Even though the paper positing this was retracted, and its author was barred from practicing medicine in the UK for the fabrication of data, academic dishonesty, and abuse of infant research subjects, the link between vaccines and autism was inseparable (Dubé et al. 2014; DeStefano and Shimabukuro 2019). This misinformation initially spread through newspaper headlines, yet most reach has been achieved online through parenting forums, conspiracy websites, and social media networks (Stein 2017).

Both vaccine hesitancy and refusal are exemplary for sentiment analysis as arguments are typically expressed emotively, rather than factually (Federman 2014). Few public health strategies have effectively counteracted the strong emotive connections people have with their vaccination beliefs, and gaining an understanding of them could mitigate the rippling impacts anti-vax communities have on society (Dubé et al. 2014). With COVID-19 vaccination strategies underway in all corners of the globe, it is vital to monitor the changing dynamics of opinions towards vaccination to assure the effectiveness of life-saving national vaccine plans (Burki 2020).

4 PROBLEM

Despite the completion of the initial NHS vaccination rollout strategy and announcements for the administration of booster jabs, 14.3% of the UK population have only partially completed the plan of two vaccines administered several weeks apart, and 21.3% of the population have not had any COVID vaccine (UK Health Security Agency 2021). This is an expected statistic as vaccine hesitant citizens are thought to make up between 10% and 30% of the general population in the UK and US (Burki 2020). However, this poses a public health worry as vaccination is an essential step to heard immunity, and unvaccinated portions of the population may result in the mutation and spread of more virulent strains of COVID (Frederiksen et al. 2020).

Understanding how the British population feels about the vaccine is of key importance in informing future public health strategies and to encourage remaining members of the population to get vaccinated. Arguments against COVID-19 vaccination have been predominantly spread through social media, with anti-vax accounts on Twitter, Instagram, and YouTube gaining at least 7.8 million followers since the outbreak in 2019 (Burki 2020). As tweets are accurate reflections of true intentions (Ceron et al. 2013), opinions expressed about the COVID vaccines on Twitter can be used as representation of the beliefs held by the general population towards the vaccine.

Twitter data has been used in the past to inform researchers of the public's opinion about COVID vaccines. A collation of tweets from six months of the pandemic found that sentiments expressed regarding the UK vaccination scheme were mostly positive (Hussain et al. 2020). In contrast, a collation of Indonesian tweets across two days of the pandemic found 56% expressed negative opinions about vaccination (Pristiyono et al. 2021). These studies reflect that average sentiment scores across long or short time periods are reductive as public opinion is dynamic; for instance, vaccine approval ratings in the US dropped by 22% between May and September 2020 (Tibbett 2020). If we assume that those with negative sentiments regarding flu vaccines hold similar beliefs about COVID ones, then the proportion of positive and negative tweets may also have cyclical patters on a weekly,

monthly, and annual basis (Piedrahita-Valdés et al. 2021). This suggests that a longitudinal approach, rather than summary statistics for fixed time periods, could provide beneficial data in determining how vaccine opinions have changed throughout the pandemic.

Individuals with fringe beliefs are underrepresented on Twitter (Ceron et al. 2013), therefore members of the population with strong political motivations behind anti-vax beliefs may not be present in samples of tweets. However, fear of COVID vaccination is not just limited to those with pre-existing anti-vaccination beliefs (Lee and Chen 2021), so it can be assumed that there may be higher levels of negative sentiment than would be predicted from similar studies into general vaccination refusal.

Analysis in the US and Brazil found that Twitter sentiment trends about COVID vaccines matched those expressed by major news outlets (Garcia and Berton 2021). Similar patterns may be seen in the UK as the British population reacts to stories about vaccine production, rollout, and setbacks, especially considering that global news headlines regarding the pandemic have been highly emotive (Aslam et al. 2020). As the Twitter algorithm leads to the formation of pro- and anti-vax social network clusters (Himelboim et al. 2019), polar news headlines could be amplified, resulting the spread of similar sentiments across related communities. COVID vaccination intentions are also correlated with social norms, as willingness to receive a COVID vaccine is positively correlated with the perceived willingness of peers (Graupensperger et al. 2021). This further supports the notion that negative or positive sentiments would be amplified when shared across Twitter network clusters.

Tweets regarding generic vaccines are more positive than those about specific brands of vaccines (Masey et al. 2016), therefore attitudes towards certain brands of a vaccine are not representative of attitudes towards the concept of vaccination. Global studies found that sentiments towards Pfizer and Moderna remained stable between December 2020 and March 2021, whereas those towards AstraZeneca decreased over time (Marcec and Likic 2021). However, no studies have been conducted regarding how the British population's opinions have changed regarding these vaccines, as well as opinions towards the soon to be administered Janssen vaccine.

5 APPROACH

The approach taken in meeting the objectives of the project followed the structure outlined in figure 5.1.

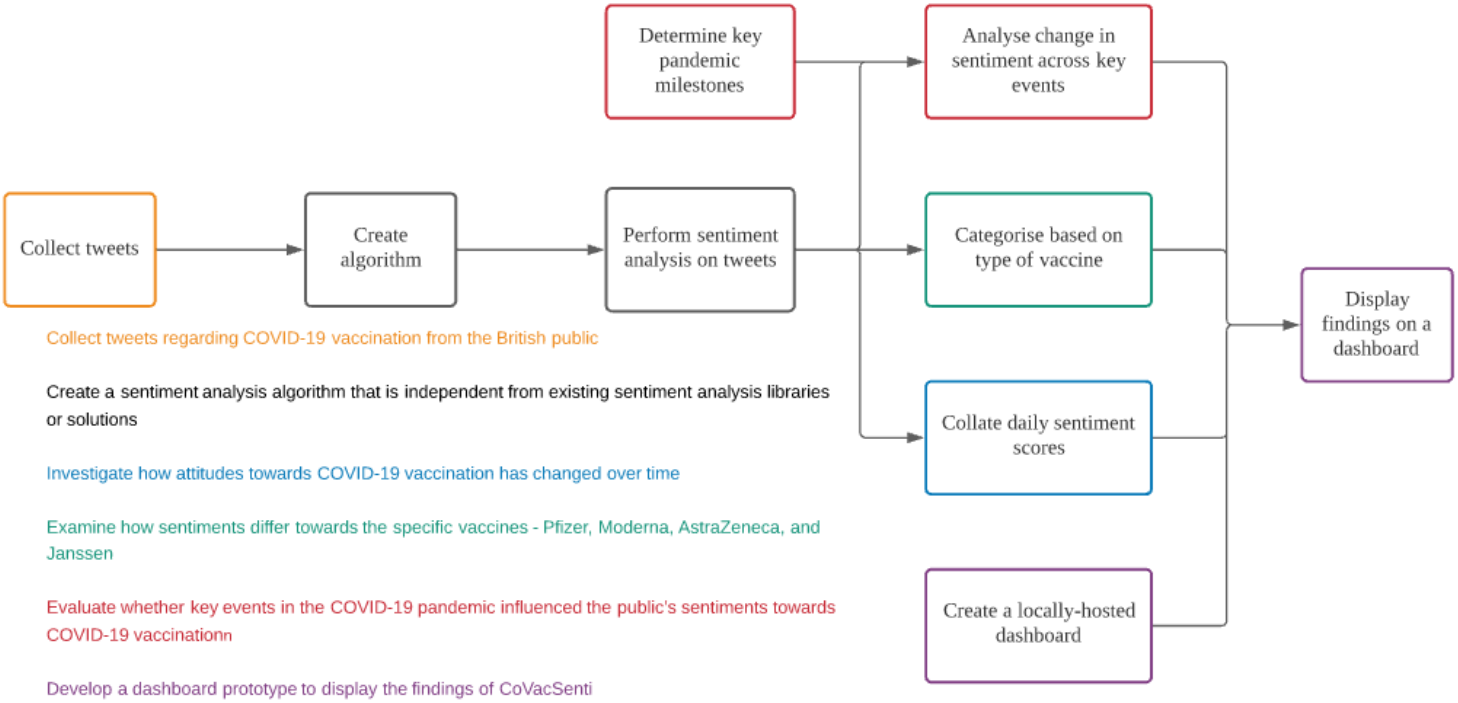


Figure 5.1 : Flowchart of overall project approach with colours representing project objectives met at each stage

5.1 DATA COLLECTION

To retrieve tweets from across the pandemic there needed to be a definitive start and end dates for the data collection period. I decided to collect tweets starting from 1st January 2020, as it would include responses to the first headlines of the outbreak and the transition of COVID-19 from an epidemic to pandemic disease (World Health Organization 2020a). I decided the end date to be the 31st of July 2021 as it marked the beginning of the dissertation period and allowed for tweets to be collected across 19 whole months.

To analyse tweets in response to key events in the British pandemic timeline, only tweets posted by users within the UK needed to be collected.

Three tools were analysed for their suitability in meeting these data collection requirements: Twint, Twitter's Standard Developer Account, and Twitter's Academic Research Product Track. These will now be explored in the order they were chronologically assessed in the planning stages of the project.

5.1.1 Twint

Twint is an open-source Python package that allows access to the Twitter API (Zacharias 2018). Twint allows for the retrieval of all publicly available tweets with no rate limitations, and can filter based on time, location, and search terms. Location searches use a tweet's *place* attribute to search by name (i.e., London, England) or *geo* attribute to search a given radius around a coordinate (i.e., 51.509865, -0.118092, 10 km).

I planned to retrieve tweets using their place attribute yet, after many failed attempts, found users detailing similar errors in unsolved GitHub issues and considered the attribute to be defunct.

I then planned to retrieve tweets using their geo attribute using a circle originating in the coordinates 54.093409, -2.8947 – the geographical centre of the UK in Morecambe Bay, Lancashire (Ordnance Survey Press Office 2014). The circle needed a radius of 760 km to cover the UK and all outlying islands. However, this resulted in the entirety of Ireland and the Netherlands falling into the search area, along with parts of France, Belgium, Luxembourg, Germany, Denmark, and Norway (figure 5.2). Despite attempts at moving the centre point northwest, it was not possible to create a search area that included the Shetland and Channel Islands yet did not include parts of Ireland. I considered iterating across multiple search circles using an algorithm based on Schlosser et al. (2021), yet collecting centre points, radii and population densities ultimately became too complex. Therefore, it was decided that Twint was not a suitable data collection method.

Figure 5.2 : Search area with a centre point at the geographical centre of the UK and radius large enough to include all outlying islands

5.1.2 Twitter's Standard Developer Account

Registered account holders can make calls to the Twitter API through the Developer Portal (Twitter 2021a). Standard Developer accounts are free, and Twitter has extensive documentation on how to query and process tweets. Tweets are retrieved with their full Twitter attributes, including hashtags, user mentions, and geotagged location. However, Twitter imposes limits on tweet retrieval for standard developers; a maximum of 180 requests may be made each 15 minutes, and search functionality is limited to tweets posted within the last week.

Research into this option was immediately abandoned upon learning of the limited scope of tweets posted within the previous week, which would not have allowed the retrieval of tweets from the start of the pandemic.

5.1.3 Twitter's Academic Research Product Track

Twitter's Academic Research Product Track was released in early 2021 to allow verified academic researchers to access the updated API version 2 for use in non-commercial projects (Twitter 2021b).

Academic researchers can perform full archive searches – the retrieval of all publicly available tweets on the platform from the past 15 years. Archive searches have a maximum of one request per second, and a monthly retrieval limit of 10 million tweets. Tweets retrieved through this track contain the same full attributes as those retrieved with a Standard Developer Account, allowing for the easy access of attributes such as location and

hashtags. However, users must apply and undergo a screening process to verify their academic status, research proposal, and intentions in terms of data use and storage.

I chose to use Twitter's Academic track to collect data due to the archive search functionality. This option was the most viable of the three as it allowed me to search for all tweets since the start of the pandemic that were geotagged to the UK.

5.2 MySQL Database

MySQL is a relational database system, which can be accessed through the web interface phpMyAdmin (Oracle Corporation 2019).

The database was designed for each entry to have an ID key that remained unchanged throughout the project to track the pre-processing and sentiment scores of tweets as modifications were made to the algorithm (figure 5.3). For each tweet, I planned to collect the date posted, the location, the text of the tweet, and any information stored in the tweet's hashtag attribute. The tweet ID linked the data table to the processed data table, which contained additional fields generated in later processing stages.

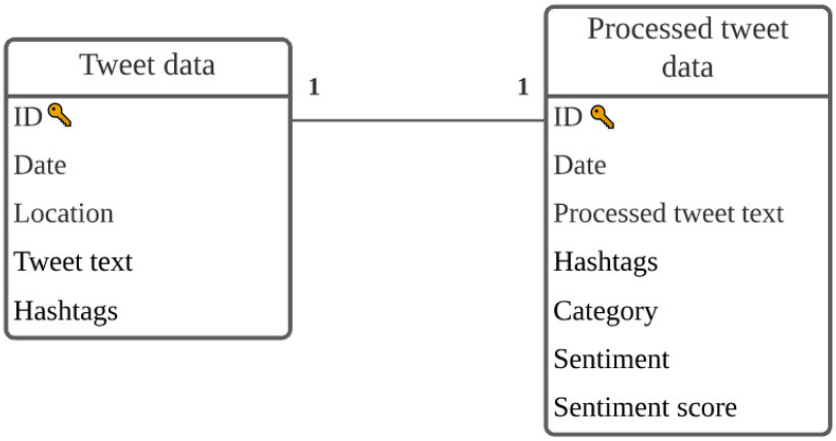


Figure 5.3 : Basic entity relationship diagram of the two related tables in the database

A disadvantage of phpMyAdmin was the required connection to the University VPN, which had experienced several outages over the previous months. With this, any issues in University servers would prevent access to the data.

5.3 TEXT PRE-PROCESSING

Pre-processing is essential in sentiment analysis to assure data is correctly interpreted and can be used for predictions. It aims to standardise data to reduce noise – text that carries little weight in determining polarity. Some sentiment analysis datasets are comprised of 40% irrelevant noise, which can greatly impact the accuracy of algorithms (Fayyad et al. 2003). One of the most common packages for text pre-processing is the Natural Language Toolkit (NLTK) (Bird et al. 2009), but I decided to create my own functions in its place to allow for more customization in my approach.

While there are many pre-processing techniques, sentiment analysis algorithms have highest accuracy when a small, tailored subset of tasks are carried out, rather than conducting as many as possible (Uysal and Gunal 2014).

Taking this into account, the pre-processing methods in this project were selected based on their specificity to the data (short 280 character-long fragments of text), avoiding all unnecessary processing where possible. The following sections will be an overview of common approaches in text processing, and whether they were implemented based on the context of the research question.

5.3.1 Hashtags

Hashtags are metadata tags that enable users to link their posts to wider discussions in the social network using the hash symbol followed by a keyword or short phrase. Hashtags provide useful insight in social analysis (Bruns and Burgess 2015), so I decided to retain them for this project. However, since the Twitter API provides access to a tweet's hashtag attribute, it was not necessary to retain hashtags within the tweet text itself and so they were removed.

5.3.2 User mentions

Twitter users can reply to tweets, or call another user's attention to a particular tweet, by referencing their username preceded with the at symbol. This is standard convention on Twitter, so I assumed all words beginning with @ were usernames, and therefore had no sentiment attached to them and should be removed. User mentions, along with hashtags, are a near-universal first step in text processing as these elements are present in most

tweets and would cause erroneous outputs if removed at later stages (Symeonidis et al. 2021).

5.3.3 Lowercasing

The conversion of all text to lowercase is one of the most common practices in sentiment analysis and, while capitalisation may indicate emphasis on emotive words or phrases (Symeonidis et al. 2018), lowercasing was chosen to be part of pre-processing to simplify the algorithm.

5.3.4 Punctuation

Punctuation may be used to indicate sentiment, such as the repetition of many exclamation or question marks to signal intense excitement or confusion, respectively. However, it is standard practice to remove punctuation in sentiment analysis (Lin and He 2009). I decided to keep with common practice and remove all punctuation because I believed most users would omit punctuation it in favour of additional words when expressing emotions within a strict character limit.

5.3.5 Emojis

The inclusion of emojis improves accuracy in paragraph-level analysis, and they often convey such strong sentiments that they surpass those conveyed by the text itself (Hogenboom et al. 2013). However, only 4% of tweets contain emojis, and the majority of popular emojis are associated with significantly positive sentiment scores (Kralj Novak et al. 2015). Due to this, I decided to forego emojis in this sentiment analysis to standardise the features across all tweets.

5.3.6 Links

Links to external sites were removed as they would have no weight in sentiments. However, in retrospect I realised it may have been insightful to analyse common sites linked to in positive and negative tweets.

5.3.7 HTML format errors

Ampersands are commonly escaped in HTML rendering, and so the removal of punctuation from the string would have caused erroneous sentiment levels caused by “&amp”.

5.3.8 Numbers

There are numerous cases in which numbers could be included in tweets. For instance, the following potential tweets all contain different uses of numbers:

The infection rate is 85 per 100,000

My vaccination experience was 10/10

I want 2 get vaccinated 4 my health

Due to variations in how numbers may be used, I decided it would be best to remove numbers during pre-processing. Some unsupervised machine learning models have improved accuracy when including numbers (Lin and He 2009), yet specific research into more relevant supervised classifiers have found that this has little effect when compared with other pre-processing options (Symeonidis et al. 2018).

5.3.9 Stop word removal

Stop words are common, topic-independent words, that are filtered out in natural language processing due to their high frequency and low specificity. While there is no universal corpus of stop words, they commonly include articles, pronouns, possessives, prepositions, conjunctions, demonstratives, and modal verbs (Ganesan 2014). While stop word filtering negatively impacts sentiment analysis accuracy in long documents such as news articles (Song, Liu, and Yang 2005), it is considered standard practice.

Words less than two characters in length are also considered to have low weight in sentiment analysis, and so are typically filtered out in pre-processing stages (Symeonidis et al. 2018). Other stopwords are typically collated in lists, such as the Natural Language Toolkit (NLTK) corpus, a list of 179 words typically filtered out in sentiment analysis (Bird et al. 2009):

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll",
"you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her',
'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what',
'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be',
'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but',
'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',
'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why',
'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not',
'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should',
"should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't",
'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn',
"shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Figure 5.4 : The Natural Language Toolkit stop word corpus

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll",
"you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her',
'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what',
'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be',
'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but',
'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between',
'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why',
'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not',
'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should',
"should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't",
'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn',
"shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Figure 5.5 : Modified Natural Language Toolkit corpus with removed terms in red

I was wary of relying on the NLTK corpus as pre-compiled non-specific stop word lists often negatively impact algorithm performance (Saif et al. 2014). Due to this I made a corpus (figure 5.5) specifically tailored to the data to compare with the standard NLTK corpus. This reflected the removal of the following terms:

- *both*

  Since all available COVID-19 vaccinations in the UK are administered in two doses, I believed that the term “both” could hold sentiment regarding the required double vaccination.

- Determiners

  The determiners *no* and *not* change the meaning of words, therefore I believed it would be worthwhile to retain them to investigate the polarity of

negations. Table 5.1 displays how tweets with opposite meanings would be processed with and without the inclusion of these terms as stop words.

- Modal verbs

  Modal verbs express necessity, obligation, and instruction. As anti-vaccine arguments are emotional while pro-vaccine arguments are factual (Federman 2014), I believed these terms would have valuable sentiment attached to them. This is particularly meaningful as Twitter conversations between users urging each other to get vaccinated or reject vaccination would look the same without the removal of modal verbs from the stop word corpus (table 5.1).

Table 5.1: Expected pre-processing outputs with standard and modified NLTK corpora

Original tweet	Standard NLTK corpus	Modified NLTK corpus
I don't want both vaccines	want vaccines	don't want both vaccines
I am vaccinated	vaccinated	vaccinated
I am not vaccinated	vaccinated	not vaccinated
You should get vaccinated	get vaccinated	should get vaccinated
You shouldn't get vaccinated	get vaccinated	shouldn't get vaccinated

5.3.10 Lemmatisation and Stemming

Lemmatisation and stemming are two methods of simplifying a word to its grammatical root to prevent variations with every plurality or tense. They are mutually exclusive methods, each with their own advantages and limitations.

Stemming removes trailing characters to standardise words, often leaving misspellings or additional characters. Meanwhile, lemmatisation finds the exact root of a word through reference to a dictionary to determine whether it contains an inflected ending (table 5.2). This is done with the addition of a speech tag to identify the use of the word. Despite the specification needed in lemmatisation, it does not significantly improve accuracy compared to stemming (Balakrishnan and Lloyd-Yemoh 2015).

Table 5.2: Comparison of stemming and lemmatisation

	Input	Output
Stemming	Caring	Car
Lemmatisation	Caring, verb	Care

When analysing long documents and large datasets, the reduction of words to their roots by either method can oversimplify and obfuscate meaning (Matsumoto et al. 2005). Both methods decrease algorithm accuracy when used on short fragments of text (Bao et al. 2014), and stop word removal is preferable for noise reduction in short text fragments (Tomana et al. 2007).

Taking previous research into consideration, I decided not to include lemmatisation or stemming stages in this project to maintain the meaning of the short 280-character tweets.

5.3.11 Tokenisation

Tokenisation is the splitting of a text entry into its component parts. This segmentation can be done at a paragraph, sentence, or phrase level. The average length of tweets is 61 characters and the length of tokens retrieved from tweets is negatively correlated with sentiment analysis reliability (Mayo 2014). Therefore, I decided to have the shortest possible token length by tokenising each word in the tweet.

5.4 Sentiment Analysis Algorithm

As mentioned in section 3.1, there are two distinct approaches to sentiment classification: either with a machine learning algorithm or with a lexicon-based solution. The machine learning approach often outperforms lexicon-based analyses when using short form data (Paltoglou and Thelwall 2012). As CoVacSenti involves the analysis of tweets, a machine learning approach was chosen to perform sentiment analysis.

The approach as to which method of machine learning to use depended on background research into the efficacy of the methods in similar contexts. Unsupervised methods are significantly more laborious for the programmer (Love 2002) and are not commonly used in classification tasks involving natural language (Gautam and Yadav 2014) as they significantly underperform compared to their supervised machine learning counterparts when

performing sentiment analysis (Schouten et al. 2018). Therefore, I decided that the machine learning algorithm used in CoVacSenti must be a supervised algorithm. One of the most used supervised algorithms in sentiment analysis is the Naïve Bayes classifier.

5.4.1 Naïve Bayes Classifiers

Naïve Bayes algorithms are based on Bayes' theorem of the probability of events occurring based on prior probabilities, with the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 5.6 : Bayes' theorem, where P(A|B) = the probability of A given B, P(B|A) = the probability of B given A, and P(A) and P(B) are the probabilities of A and B, respectively

The term "naïve" is applied as the algorithm assumes that the probability of a particular feature is independent of the presence of other features. In terms of sentiment analysis, this assumes that in the sentence:

I hate the idea of a COVID vaccine because I am afraid of doctors and needles

the probability of the word *vaccine* appearing is independent of the presence of the words *doctors* and *needles*.

I decided that a Naïve Bayes classifier would be best to perform sentiment analysis in CoVacSenti as they are significantly faster to train than the more complex algorithms of K-Nearest Neighbour classifiers or Support Vector Machines (Goel et al. 2016). Naïve Bayes also outperforms K-Nearest Neighbour classifiers in short and long form text (Islam et al. 2007) and outperforms Support Vector Machines when classifying tweet polarity (Kristiyanti et al. 2018) and personality types from emotive tweets (Pratama and Sarno 2015).

There are different types of Naïve Bayes classifiers depending on the distribution of data, with the two most relevant being *Multinomial* and *Bernoulli*. Multinomial classifiers are used with data comprised of discrete features, for instance the frequency each word in positive and negative tweets. Bernoulli classifiers are similar but interpret each term as a Boolean,

indicating the presence or absence of positive or negative words. As Multinomial algorithms tend to have higher accuracy than Bernoulli (Abbas et al. 2019; Singh et al. 2019), a Multinomial Naïve Bayes classifier was chosen as the sentiment analysis algorithm for CoVacSenti.

5.4.2 Measures of success

Naïve Bayes classifiers created in academic research have a wide range of accuracies. A meta-analysis of 216 sentiment analysis studies found the median accuracy of Naïve Bayes classifiers was 79.1% (Siebert et al. 2019). This median score was used as the baseline to measure the success of CoVacSenti as it represents the standard in the field.

The results of CoVacSenti were also compared against those generated by the NLTK to determine whether the created algorithm is a viable alternative to existing solutions.

5.5 Key Events

The approach to the key events analysis was based on hypothesis testing with the following alternate and null hypotheses:

> **H_1** There is a relationship between key events in the pandemic and the population's sentiments towards COVID-19 vaccination
>
> **H_0** There is no relationship between key events in the pandemic and the population's sentiments towards COVID-19 vaccination; any changes in sentiment are due to random chance.

A comparison of sentiments recorded the day before and day of each event was used to address the hypotheses. Even though events may have had long lasting impacts or caused discourse for a few days, this day-by-day analysis was chosen due to the "infodemic" of emotive, past-paced, and ever-changing news stories broken during the pandemic (World Health Organization 2020b). Therefore, the key events analysis was conducted under the assumption that most tweets regarding an event would be posted on the day of the event itself.

To normalise against varying numbers of tweets, the analysis of sentiment change was conducted twice for each event. First using the number of positive and negative tweets on each day, then again with the percentage of positive and negative tweets each day.

A Fisher's Exact test was chosen as the statistical significance test as this analysis measured the change in two categorical variables (the number/percent of positive and negative tweets) across two conditions (the day before and day of each event). This was chosen in lieu of a typical Chi-Squared test as exact accuracy was needed to account for days with less than 5 tweets, the data size threshold for the Fisher's test (Kim 2017). Additionally, the dimensions of the created contingency tables were fixed at 2x2, so the approximations offered in Chi-Squared analyses were unnecessary.

The level of significance was $p = 0.05$; results were considered statistically significant if there was less than 5% probability that changes in sentiment were due to random chance.

5.6 Dashboard Prototype

I decided to create a simple dashboard prototype to host the results of this study using Flask (Ronacher 2021). Flask was chosen due to ease of use and integration with the MySQL database. To measure success and have a definition of done I planned to begin the creation stage by designing a wireframe diagram to guide construction, and a test case to use post-production as criteria to determine whether the dashboard was implemented successfully.

User experience was prioritised in the prototype approach, so the site was planned to be as intuitive as possible. While a minimum of three graphs was necessary to address the results of the data-driven objectives (change in sentiment over time, attitudes towards specific vaccines, and key events analysis) I planned to use six visualisation to guide the user through the page with aesthetically pleasing symmetry, in alignment with the principles of dashboard design (Janes et al. 2013).

Many pallets in data visualisation packages are inaccessible for viewers with colour blindness (Cravit 2019). Therefore, I planned for categorical colours on the CoVacSenti dashboard to be based on distinct shades, rather than hues, so that results could be accessible for all users.

5.6.1 Visualising sentiment over time

As the date and the number of tweets each day are both continuous variables, a time series line graph was best suited to show changes in daily sentiment over time. I planned to overlay two lines on the same plot: one for positive tweets per day and one for negative, so that trends could be easily compared.

The proportion of sentiments each day was also an effective measure of change over time. As this used a different scale to represent the same information as the change in number of sentiments over time, I decided to create an integrated interactive visualisation with the ability to toggle between the two. To avoid confusion and create cohesion, I planned for them both to follow the same typical colour convention of green to represent positivity and red to represent negativity (Murray 2019).

As the daily percentage of positive and negative tweets sum to 100, I decided that a 100% stacked area chart would be an interesting way to showcase this relationship. Stacked area charts show how parts of a total number have changed over time, with the area above and below a line representing the percentage of each category.

5.6.2 Visualising key events

The initial approach for visualising events was a linear timeline yet initial drafts revealed a significant amount of white space, creating an unfavourable data:ink ratio that made the visualisation unintuitive for users (Sigdel 2020).

The alternative approach was to therefore not use this data as a stand-alone visualisation, but to include it as extra contextual information in the interactive plot (section 5.6.1) using a toggled tooltip over each relevant date. However, a pitfall in this approach is that essential information often remains hidden behind unclicked toggle options, as not all users engage with all interactable elements (Stabe 2016). To combat this, I chose to have only one toggle option to toggle all key event tooltips, and for the details of each event to be as brief as possible.

5.6.3 Visualising sentiments towards vaccine types

A scatter plot was chosen as the visualisation approach to reduce monotony on the website by avoiding another variation of a line graph. To exemplify the functionality of sentiment

analysis algorithms, I believed this plot should use the sentiment scores themselves, rather than the binary categories of positive or negative so that users could clearly see the polarity of sentiment towards the different brands. Overcrowded plots prevent users from understanding the meaning behind data, which can be addressed by reducing the number of data points on a graph (Sigdel 2020). As I feared this visualisation may be overwhelming with thousands of sentiment scores, a toggle feature was planned so that users could control what vaccine types were displayed on the plot.

5.6.4 Qualitative plots

A common approach in sentiment analysis research is the inclusion of qualitative findings (Pristiyono et al. 2021), mainly word clouds of the most common terms found in positive and negative entries. Therefore, the final visualisations on the CoVacSenti dashboard fit into the existing body of work by including these qualitative results. To extend this concept with a unique approach, I also planned to create *hashtag clouds* to represent the common hashtags used in positive and negative tweets as they offer insight into widespread trends when conducting social research (Bruns and Burgess 2015).

6 IMPLEMENTATION

6.1 OVERVIEW

The approach was implemented using Python 3.8.5 (van Rossum and Drake 2009), while the dashboard was implemented using HTML5, JavaScript 1.7, and CSS3.

The overall project structure was implemented using the file system outlined in figure 6.1.

```
CoVacSenti/
├─ program/
│  ├─ __init__.py
│  ├─ data/
│  │  ├─ processed/
│  │  └── sentiment140.csv
│  ├─ figures/
│  ├─ analysis.py
│  ├─ collect.py
│  ├─ index.html
│  ├─ naïve_bayes.py
│  ├─ process.py
│  ├─ style.css
│  └── train_test_run.py
├─ requirements.txt
├─ wsgi.py
└──.gitignore
```

Figure 6.1 : Directory tree with program outputs (data and image files) omitted for clarity

The script in *collect.py* contained all functions used to connect to the Twitter API and collect data. *Process* was a class that contained all methods for text pre-processing. Similarly, *Naïve Bayes* was a class that contained all methods and variables used to perform the Bayesian sentiment analysis.

Train Test Run was the general calling script of the program and was where Process and Naïve Bayes were instantiated. The script trained and tested the algorithm and then ran it on the collected Twitter data.

Analysis contained methods for performing statistical analysis as part of the key events portion of the project. This script also collated the findings of CoVacSenti into files suitable for plotting, and created the word cloud visualisations.

The files *index.html* and *style.css* were used to create and design the dashboard. The index file also contained a JavaScript script that governed the data processing, visualisation, and interactions on the dashboard.

The three folders – *figures*, *data*, and *processed* – stored figures, data, and processed data, respectively. All data files besides *sentiment140.csv* were omitted from figure 6.1 for simplicity as these were generated during implementation, whereas the Seniment140 data was sourced externally (section 6.4.1), and therefore took precedence.

Lastly, *__init__.py* and *wsgi.py* were used to establish the Flask framework, *requirements.txt* maintained a list of packages installed in the virtual environment, and *.gitignore* prevented cached and package files from being saved to the GitHub repository.

6.2 Data Collection

Data collection was conducted in collect.py and was managed by a calling function to iterate through each day in the pandemic timeline and call the following functions.

6.2.1 Create URL

```
def create_url(start_date, end_date, max_results = 500):

    search_url = "https://api.twitter.com/2/tweets/search/all"

    parameters = {'query': "vaccine (covid OR coronavirus OR moderna OR astrazeneca OR
                   pfizer OR janssen) lang:en place_country:GB",
                    'start_time': start_date,
                    'end_time': end_date,
                    'max_results': max_results,
                    'expansions': 'geo.place_id',
                    'tweet.fields': 'text,created_at,entities',
                    'next_token': {}}
    return (search_url, parameters)
```

Figure 6.2 : Create URL function

The full archive search was conducted by adding query parameters to a search URL (figure 6.2). The *query* field referred to the search term itself: "vaccine" along with either "covid", "coronavirus", or the names of the specific types of vaccine. The manufacturer name - Janssen - was used for the search term for the Johnson & Johnson vaccine to avoid collecting tweets regarding Boris Johnson. The query field was case insensitive, so variations of search terms were not supplied.

Other parameters included specifications for language and location, which were English and the UK, respectively. Daily start dates were provided by the calling function in collect.py.

Queries must include a maximum number of results as archive searches could potentially return all tweets since the launch Twitter. The upper limit for this value, as set by Twitter, is 500. I decided to use this value as it would return a maximum of 288,500 tweets over the 577 days and believed that, with the time-sensitive nature of the vaccination rollout, it was best to retrieve as much data as possible.

Both the *expansions* and *tweet field* attributes referred to metadata collected from each tweet – a geocode referring to the nearest town or large city, the tweet text itself, the date and time of the post, and the *entities* (any hashtags used in the tweet).

The *next token* was a meta data item included in each response from the API, which signalled that the full response was separated by pagination and could only be returned in batches. This variable therefore controlled page navigation and was supplied to each archive search automatically through the Connect Endpoint function (figure 6.3) if returned in a previous API response.

6.2.2 Connection to the Twitter API

The first stage in connecting to the API endpoint was the creation of a project on the Twitter developer portal. This allowed for the generation of a Bearer Token – a unique identifier to authenticate the connection between the local Python program and Twitter developer account. An authorized connection was formed when the Bearer Token was called as part of an authorization header (figure 6.3), which allowed connection to the archive search endpoint through GET requests using the Python Requests library (Reitz 2021).

```
def connect_endpoint(url, params, header, next_token = None):
#if applicable, add the next token to the search parameters for pagination
    params['next_token'] = next_token
    response = requests.request("GET", url, headers = header, params = params)
    print("Endpoint Response Code: " + str(response.status_code))
    if response.status_code != 200:
        raise Exception(response.status_code, response.text)
    return response.json()
```

Figure 6.3 : Connect Endpoint function

The Connect Endpoint function sent a GET request to the URL with specific search parameters, as established in the Create URL function, along with any saved next tokens for pagination. A response code of 200 signalled that a successful connection was made, and the response from the API was saved in JSON format.

6.2.3 Connection to MySQL

```
def push_to_mysql(json_response, result_count):

    for tweet in json_response['data']:
        result_count += 1
        id_num = result_count
        created_at = dateutil.parser.parse(tweet['created_at'])

        if ('geo' in tweet):
            place_id = tweet['geo']["place_id"]
        else:
            place_id = ""

        text = tweet['text']

        if "entities" in tweet:
            if "hashtags" in tweet["entities"]:
                hash_list = [data["tag"] for data in tweet['entities']["hashtags"]]
                hashtags = ",".join(hash_list)
            else:
                hashtags = ""
        else:
            hashtags = ""

        new_entry = Data(id=id_num,date_time=created_at, place_id=place_id,
                    tweet=text,hashtags=hashtags)
        db.session.add(new_entry)
        db.session.commit()

    print("SUCCESSFULLY ADDED TWEETS TO MYSQL")
```

Figure 6.4 : Push to MySQL function

Tweets were added to the MySQL database through SQLAlchemy (Bayer 2021). This was done by iterating through each tweet in each JSON response, assigning it a unique integer ID, and linking the attributes to the corresponding variables in the MySQL database (figure 6.4). Tweets were then pushed to the database chronologically, and their IDs and dates were printed to the terminal as a status update.

6.2.4 Database implementation

There were several errors importing data to MySQL due to atypical characters - hyphens, emoticons, trailing spaces, and accents. Tweets containing these could not be collated using the standard phpMyAdmin setting of Latin 1. However, changing the collation setting to 4-Byte UTF-8 Binary Unicode for all columns with text inputs resolved these issues.

The raw data was stored in Data (figure 6.5). Each column had a data required clause besides the hashtag column, as not all tweets contained them.

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**id**	int(11)			No	*None*
☐	2	**date_time**	datetime			No	*None*
☐	3	**place_id**	text	utf8mb4_bin		No	*None*
☐	4	**tweet**	text	utf8mb4_bin		No	*None*
☐	5	**hashtags**	text	utf8mb4_bin		Yes	*NULL*

Figure 6.5 : Layout of Data, the MySQL table that stored raw data

The accompanying Processed Data table (figure 6.6) was linked by the ID as the primary key and was set up with the same collation settings to avoid errors. Similarly, all columns in this table were required besides hashtags and category - the name brand of vaccine.

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**id**	int(11)			No	*None*
☐	2	**date_time**	datetime			No	*None*
☐	3	**tokenized_tweet**	text	utf8mb4_bin		No	*None*
☐	4	**hashtags**	text	utf8mb4_bin		Yes	*NULL*
☐	5	**category**	text	utf8mb4_bin		Yes	*NULL*
☐	6	**sentiment_score**	float			No	*None*
☐	7	**p**	float			No	*None*

Figure 6.6 : Layout of Processed Data, the MySQL table that stored data after text processing and sentiment analysis

6.3 Tweet Pre-Processing

Tweet processing methods were carried out as part of a class, *Process* (figure 6.7), with instances of the class called for each tweet. This modular organisation kept the processing methods separate from the calls to the database and allowed for improvements and

additional methods to be added at later stages of development (section 6.5) without impacting the existing functionality. Process involved four methods: a constructor, two methods for text pre-processing, and a calling function to link them.

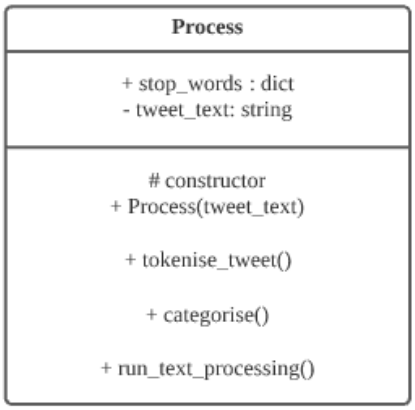


Figure 6.7 : Overview of the Process class

6.3.1 Tokenise Tweet

```
def tokenise_tweet(self):

        print("____________________________")
        print(f"ORIGINAL TWEET: [{self.tweet}]")

        text_list = []
        cleaned_text = ""

        for word in self.tweet.split():
            #remove usernames, links, hashtags
            if not word.startswith(("@","&amp","http","www.","#")):
                #remove punctuation and non-alpha chars, change to lower case
                cleaned_text += re.sub('[^a-zA-Z]',' ',word).lower() + " "

        #remove stopwords and words less than 2 chars
        tokens = [word for word in cleaned_text.split() if word not in self.stopwords and len(word)>2]

        print(f"TOKENIZED TWEET: [{tokens}]")
        return tokens
```

Figure 6.8 : Tokenise Tweet function

Tokenise Tweet (figure 6.8) conducted the main text pre-processing in the following order:

1) Split the tweet into individual words
2) Remove all user mentions, HTML format errors, links, and hashtags

3) Use a regular expression to remove all punctuation, numbers, and emojis
4) Lowercase the word and add it to an intermediary, semi-processed string
5) Remove stopwords and words shorter than two characters in length
6) Tokenise each word (convert remaining words into elements of a list for output)

6.3.2 Categorise

```
def categorise(self,tweet):

        categories = set(["moderna","astrazeneca","pfizer","janssen"])
        #remove all words except those in the categories set
        found_category = re.findall('|'.join(categories),tweet.lower())
        no_repeats = list(set(found_category))

        return " ".join(no_repeats)
```

Figure 6.9 : Categorise function

Categorise (figure 6.9) was created to determine which keywords were present in each tweet. As some tweets mention multiple types of vaccine, or the same type multiple times, the function accounted for repeats of key terms in the following steps:

1) Create a set of vaccine names
2) Remove all words from the tweet besides those in the union of the set and tweet text
3) Convert to set and list to remove repeat occurrences

Categorise outputted a string with each vaccine split by a space, or an empty string if no vaccine brands were mentioned.

6.3.3 Run Text Processing

Run Text Processing served as a calling function so that only one function needed to be called after the class was initialised. The running of Tokenise Tweet and Categorise completed all text pre-processing for CoVacSenti, and the results were outputted as elements in a list. An example of the output of the full text pre-processing can be seen in table 6.1.

Table 6.1 : Sample outputs from each text pre-processing function

Original tweet	Tokenise Tweet output	Categorise output	Run Text Processing output
Excited 2 get Pfizer vaccine @NHSuk	[“excited”, “get”, “pfizer”, “vaccine”]	“pfizer”	[[“excited”, “get”, “pfizer”, “vaccine”], “pfizer”]
COVID vaccine today #NHS #scared	[“covid”, “vaccine”, “today”]	“”	[[“covid”, “vaccine”, “today”],””]
Janssen is dangerous - boycott janssen, trust pfizer!	[“janssen”, “vaccine”, “dangerous”, “boycott”, “janssen”, “trust”, “pfizer”]	“janssen pfizer”	[[“janssen”, “vaccine”, “dangerous”, “boycott”, “janssen”], “janssen pfizer”]

6.4 NAÏVE BAYES ALGORITHM

The formation of the algorithm was split into three main categories: the collection and text pre-processing of the *teacher* data, the creation of positive and negative corpora, and the algorithm itself, which was modularly separated in its own class (figure 6.12) and called within Test Train Run.

6.4.1 Sentiment140

Supervised machine learning algorithms require a teacher dataset whose sentiments have been manually labelled. Most of this data is used to train the model, while a subset is used to test it by comparing the output against the existing labels to determine the model’s accuracy.

Sentiment140, a sentiment analysis dataset of 1.6 million tweets (Go et al. 2019), was used to train the CoVacSenti model. I chose not to create my own labelled database as this would be unnecessarily laborious. The Sentiment140 data was not manually marked, so there is the possibility that some tweets may be erroneously labelled, yet the distant supervision algorithm used to create the dataset had an accuracy above 80% so it can be assumed most tweets are labelled correctly (Go et al. 2009). These tweets are labelled as either positive or negative and were collected over a three-month period in 2009.

The Sentiment140 data was downloaded in one large comma-separated value file, which was difficult to use due to its size. For ease of use, this data was also imported into the MySQL database in two tables – one for positive and one for negative tweets – to have clear

separation between the sentiments without need for filtering commands. This created two teacher datasets with 800,000 tweets in each.

The reusability of modular methods through the Process class (figure 6.7) allowed for text pre-processing to be very efficient. As minimal information (the tweet text itself and the sentiment label) was needed to train the algorithm, only the Tokenise Tweet function was needed at this stage.

Sentiment140 used the label *0* to a signify negative sentiment and *4* to signify a positive sentiment. In pre-processing the label for positive sentiment was changed to 1 to be better suited to the binary classifications used in Multinomial Naïve Bayes.

As is convention in sentiment analysis, each of the labelled sentiment datasets were split so that 80% was used for training and 20% was used for testing (figure 6.10).

```
all_positive = PositiveSentiment140.query.all()
pos_training = [row.tweet for row in all_positive[:640000]]
pos_testing = [row.tweet for row in all_positive[640000:]]

all_negative = NegativeSentiment140.query.all()
neg_training = [row.tweet for row in all_negative[:640000]]
neg_testing = [row.tweet for row in all_negative[640000:]]
```

Figure 6.10 : Code to split testing and training data

This created positive and negative training datasets, each 640,000 tweets in length, and positive and negative testing datasets each 160,000 tweets in length. However, variations on this split were conducted later in the project timeline, as discussed in section 6.5.

6.4.2 Corpus creation

Multinomial Naïve Bayes algorithms determine sentiment through referencing corpora of positive and negative terms, as created from the training dataset. These were created using the Create Corpus function (6.11) in Train Test Run. This was external to the Naïve Bayes class as it was unnecessary to create new corpora upon each instantiation of the class.

```
def create_corpus(tweet_list):
    tweet_dict = {}
    for tweet in tweet_list:
        for word in tweet.split():
            if word not in more_stop:
                if word not in tweet_dict.keys():
                    tweet_dict[word] = 1
                else:
                    tweet_dict[word] += 1
    return tweet_dict
```

Figure 6.11 : Create Corpus function

This function took a list of tweets (either the positive training or the negative training list) and created a dictionary of the frequency counts of each word. The reusability of this code allowed for separate positive and negative corpora to be created, which were then stored locally in text files. This local storage was done to circumnavigate the outages that were occurring at the time with the University VPN, which prevented pulls from the MySQL databases.

6.4.3 The Naïve Bayes class

The Naïve Bayes algorithm was separated into its own class (figure 6.12). This modular layout allowed variables to be stored and re-assigned within the class without overcrowding Test Train Run, which was helpful when refining the algorithm under different sets of conditions (section 6.5). The purpose of this class was to compute the most probable sentiment for a tweet given the prior probability and the likelihood of a tweet having a certain sentiment given the words used in it. This was achieved through four key methods in the class: Corpus Search, Train, Predict Sentiment, and Test.

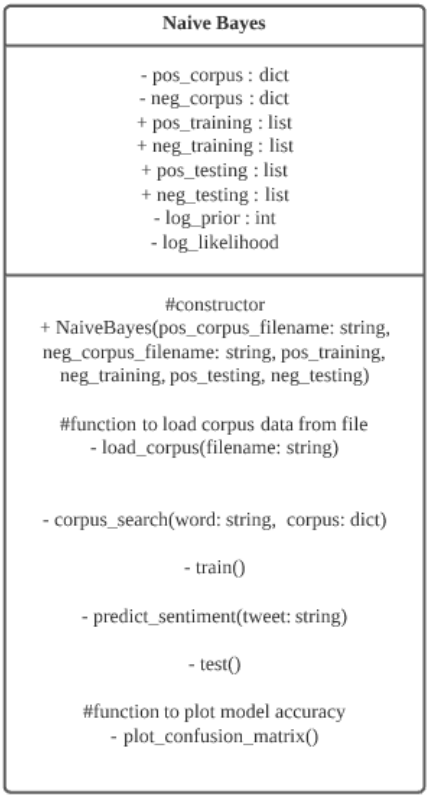


Figure 6.12 : The Naive Bayes class

6.4.4 Corpus Search

Corpus Search was a lookup function that linked the positive and negative corpora with the Naïve Bayes algorithm.

```
#need to specify positive or negative corpus
 def corpus_search(self, word, corpus):
     if word in corpus:
         return corpus[word]
     else:
         print(f"{word} could not be found in corpus")
         return 0
```

Figure 6.13 : Corpus Search function

The function took a word as the first input, and a corpus (either the positive or negative one) as the second input. If the word was present in the corpus, then it would output the corresponding frequency count of the word. If not, it would return 0.

6.4.5 Train

The Train function used the training datasets to determine the two main variables in Bayesian sentiment analysis – the prior probability and the likelihood.

```python
def train(self):

        loglikelihood = {}
        logprior = 0

        # number of unique words across both corpora
        all_words=list(self.pos_corpus.keys()) + list(self.neg_corpus.keys())

        vocab = list(dict.fromkeys(all_words))
        V = len(vocab)

        # number of unique positive words
        V_pos = len(self.pos_corpus)
        # number of unique negative words
        V_neg = len(self.neg_corpus)
        # total number of positive words
        N_pos = sum(self.pos_corpus.values())
        #total number of negative words
        N_neg = sum(self.neg_corpus.values())

        #total number of training tweets
        D = len(self.pos_training + self.neg_training)
        #number of positive training tweets
        D_pos = len(self.pos_training)
        #number of negative training tweets
        D_neg = len(self.neg_training)

        #calculate the logprior
        logprior = np.log(D_pos) - np.log(D_neg)

        #for each unique word used in all training tweets
        for word in vocab:
            #frequency of word in +ve tweets
            pos_freq = self.corpus_search(word, self.pos_corpus)

            #frequency word in -ve tweets
            neg_freq = self.corpus_search(word, self.neg_corpus)

            #calculate probability of word being +ve or -ve
            #apply laplace smoothing
            pos_word_prob = (pos_freq + 1) / (N_pos + V)
            neg_word_prob = (neg_freq + 1) / (N_neg + V)

            #calculate log likelihood of the word and add to the dictionary
            loglikelihood[word] = np.log(pos_word_prob) - np.log(neg_word_prob)

        return logprior, loglikelihood
```

Figure 6.14 : Train function

First, the function assigned variables corresponding to the total number of words, unique words, and corpora lengths. The *log prior* was then calculated. This is a form of prior probability - the existing probability distribution of sentiment prior to analysing tweets. This

was calculated using the log difference between the number of positive and negative tweets.

The next step in training the algorithm was the creation of the *log likelihood* dictionary; the likelihood of each word being positive and negative. The function iterated over each word in all the positive and negative training tweets and then used the Corpus Search function to determine the number of times the word occurred in both corpora. This was then divided by the total number of words in that corpus. Laplace smoothing was then applied (the addition of 1 in the numerator and the variable *V* in the denominator in figure 6.14) to avoid probabilities of zero.

The results of the Train function were the log prior and the log likelihood. A logarithmic scale was used as probabilities tend to become very small in sentiment analysis, therefore negligible differences must be accentuated.

6.4.6 Predict Sentiment

Based on the prior probability and the likelihood associated with each word, the sentiment of tweets was then assessed using the Predict Sentiment method. As per Multinomial Naïve Bayes conditions, if a word was not present in the log likelihood dictionary, it was not included in calculations.

```
def predict_sentiment(self, tweet):

        p = 0
        p += self.log_prior

        for word in tweet.split():
            if word in self.log_likelihood:
                p += self.log_likelihood[word]
        return p
```

Figure 6.15 : Predict Sentiment function

Predict Sentiment summed the log prior and log likelihood of each word in the tweet, returning a polarity score based around zero, with tweets below indicating a negative sentiment and tweets above indicating a positive sentiment. Sentiment strength increased with distance from zero.

6.4.7 Test

The Test method used the testing datasets to determine the model's accuracy. It iterated through the test tweets, calling the Predict Sentiment function on each. It then simplified the continuous polarity scores into a binary positive or negative classification before adding them to an ordered array. This was then compared against the true sentiment categorisations as provided by Sentiment140 by using the mean difference between the calculated and actual scores. The function finally returns the algorithm's accuracy and plots a confusion matrix.

```
def test(self):
    accuracy = 0
    predictions = []
    #combine all testing data together for iteration
    testing_tweets = self.pos_testing + self.neg_testing

    for tweet in testing_tweets:

        if self.predict_sentiment(tweet) > 0:

            predictions.append(1)
            print(f"[{tweet}] :: Positive sentiment -> {self.predict_sentiment(tweet)}")


        else:
            predictions.append(0)
            print(f"[{tweet}] :: Negative sentiment -> {self.predict_sentiment(tweet)}")

    #create array of actual sentiment scores (1 = +ve, 0 = -ve)
    actual_sentiments = [1]*(len(self.pos_testing)) + [0]*(len(self.neg_testing))
    error = np.mean(np.absolute(np.array(predictions) - np.array(actual_sentiments)))
    accuracy = 1-error
    print(f"THIS NAIVE BAYES ALGORITM HAS AN ACCURACY OF {accuracy*100}%")

    #plot confusion matrix
    self.plot_confusion_matrix(actual_sentiments,predictions)

    return accuracy
```

Figure 6.16 : Test function

6.5 Algorithm Accuracy and Improvements

6.5.1 Preliminary algorithm

The first attempt at creating the Naïve Bayes algorithm resulted in an accuracy of 50.3%, which was unexpected until finding a mistake where the Predict Sentiment function iterated over each letter in the test data rather than each word. Correcting this issue led to an algorithm with an accuracy of 76.0%. However, this did not meet the 79.1% threshold for success as defined in section 5.4.2.

The accuracy of a sentiment analysis algorithm can be affected by two factors: the quality of the algorithm itself, and the quality of the processed teacher data. To determine which area required improvements, I performed a Naïve Bayes classification using the NLTK package on the same data, which resulted in a model accuracy of 75.5%. This allowed me to assume that, since the models performed within 0.5% of each other, the teacher data was responsible for the model's sub-par performance.

6.5.2 Improvements to Sentiment140 processing

I created an additional function in the Process class named Tokenise Tweet Sentiment140 (figure 6.17), which was a modification of the Tokenise Tweet function (figure 6.8) that was tailored specifically to improve the integrity of the teacher data.

```python
def tokenise_tweet_sentiment140(self):
    print("_____________________________")
    print(f"ORIGINAL TWEET: [{self.tweet}]")
    text_list = []
    cleaned_text = ""

    for word in self.tweet.split():
        #remove usernames, links, hashtags
        if not word.startswith(("@","&amp;","http","www.","#")):
            cleaned_text += re.sub('[^a-zA-Z]',' ',word).lower() + " "

    #remove stopwords and words less than 2 letters
    tokens = [word for word in cleaned_text.split() if word not in self.stopwords and len(word)>2]

    # language filtering
    joined_str = " ".join(tokens)
    if len(tokens)!= 0 and detect(joined_str) != "en":
        print(f" --- INCOMPATIBLE LANG : tweet is detected as [{detect(joined_str)}]")
        return []

    print(f"TOKENIZED TWEET: [{tokens}]")
    return tokens
```

Figure 6.17 : Tokenise Tweet Sentiment140 function

Command line updates revealed that many words in the testing dataset were unused in calculating sentiment as they could not be found in neither corpora (figure 6.18). These were mainly words that were written in languages besides English or words that were misspelt.

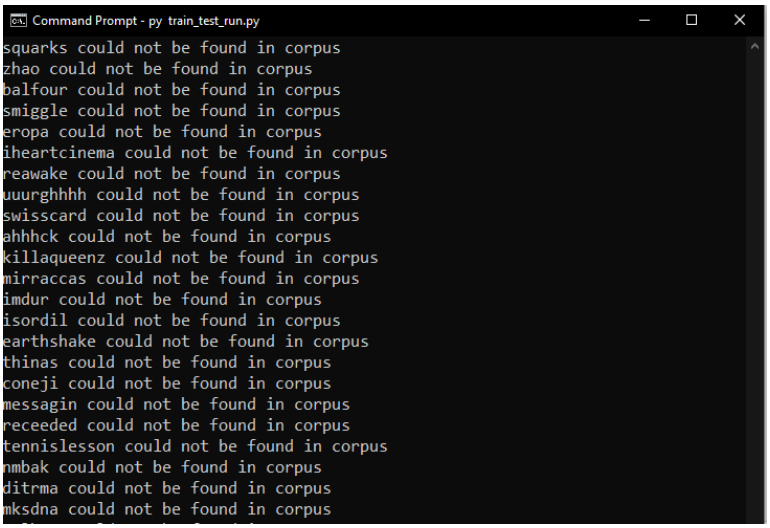


Figure 6.18 : Command line outputs of the Corpus Search function

Language issues were resolved by using the detect function from the package langdetect (Danilk 2016) within the function Tokenise Tweet Sentiment140. This was successful in removing non-English tweets. However, the package also removed English tweets that fell into the following categories:

- Tweets containing foreign names, either for places or people, such as:
    - *Jakarta here I come!*
    - *Emmanuel Macron said that?*
- Tweets using heavy text-speak
    - *OMG I h8 that… How r u 2day tho?*
- Tweets using many repeating characters
    - *I am soooooooooooooo exciiiiiiiiited*
- Tweets containing words cognates common across different languages
    - *Unique t-shirt – cool!*

While this filter was excessive, the elimination of text speech and repeating characters was beneficial as these did not have corpus representations therefore produced the same result as non-English tweets (figure 6.18).

Searching the MySQL database also revealed many tokenised tweets contained empty strings. This was a result of every character being removed during the pre-processing stage due to the tweet only containing hashtags, user mentions, words less than two characters, or stopwords. For example:

#WhosAFanOf @NHSuk? I am! :-)

When provided with an empty string, Predict Sentiment would output a sentiment score equal to the log prior probability (zero) which was therefore classified as negative by the Test function. Empty token filtration was added into the Clean Teacher Data (figure 6.19) function in the Train Test Run script.

```python
def clean_teacher_data():
    with open("sentiment140.csv",'r',encoding='utf-8') as data_csv:
        data_reader = csv.reader(data_csv, delimiter = ",",quotechar = '"')
        tweet_id = 1
        for line in data_reader:

            sentiment = line[0]
            training_id = tweet_id
            #create instance of Process with tweet text as input and run pre-processing
            p = process.ProcessTweets(line[5])
            tokens = p.tokenise_tweet_sentiment140()

            #filter out empty entries
            tweet_str = " ".join(tokens)
            if len(tweet_str.strip()) == 0:
                continue

            #add to positive or negative db
            if sentiment == "0":
                new_entry = NegativeSentiment140(id = training_id, tweet = tweet_str, sentiment_score = "0")
                print("ADDING NEGATIVE ENTRY")
            else:
                new_entry = PositiveSentiment140(id = training_id, tweet = tweet_str, sentiment_score = "1")
                print("ADDING POSITIVE ENTRY")

            db.session.add(new_entry)
            db.session.commit()
            print("---------- TWEET ADDED TO MYSQL ----------")
            tweet_id+=1
```

Figure 6.19 : Clean Teacher Data function

6.5.3 Variations on splitting the teaching data

After these modifications, the size of the positive tweet teacher dataset decreased from 800,000 to 594,543 and the negative dataset decreased to 753,483. This brought into question as to how the split of the overall teacher data into 80% for training the algorithm and 20% for testing could be conducted, as an unequal number of examples for each category impact the model's accuracy. Multiple variations of the split were conducted to determine which type of split would yield the highest accuracy.

The first condition conducted the standard split with 80% to train the model and 20% to test. This resulted in both the testing and training data containing significantly more negative tweets than positive. The second condition conducted the 80% split based on the size of the smaller dataset and allocated any remaining data to the testing. This meant that there were an equal number of positive and negative tweets in the training dataset, and an

unequal number in the testing data. The third condition removed 158,940 negative tweets so that both datasets had equal length, and then conducted the standard percentage split.

The results, as shown in table 6.2, reflected that reducing the teacher dataset size to maintain an equal number of positive and negative tweets resulted in the highest accuracy. CoVacSenti also outperformed the NLTK using the same teacher dataset split.

Table 6.2 : Summary table of algorithm accuracy under different conditions

Algorithm	**Condition**	**Accuracy**
Attempts prior to teacher data improvements		
CoVacSenti	Preliminary algorithm	76.0%
NLTK	Preliminary algorithm	75.5%
Attempts following teacher data improvements		
CoVacSenti	Equal percentage of positive and negative tweets in training and testing data	78.6%
CoVacSenti	Equal number of tweets in training data, unequal in testing data	79.2%
CoVacSenti	Teacher dataset reduction: equal number of tweets in training and testing data	**79.4%**
NLTK	Teacher dataset reduction	75.8%
Attempts following teacher dataset reduction		
CoVacSenti	Modified stop word dictionary	78.5%
CoVacSenti	Pandemic terms removed from corpora	**79.4%**

6.5.4 Impacts of alternative stop words

As mentioned in section 5.3.9, stop word removal would be approached by comparing two main stop word dictionaries, the standard NLTK dictionary and a modified dictionary, to determine whether the removal of the term “both”, determiners, and modal verbs from the stop word list would impact the algorithm.

All preliminary attempts at creating the algorithm involved the standard NLTK stop word dictionary. After the ideal teacher data split was established, the algorithm was trained again using these split conditions with the modified stop word dictionary. The algorithm using the modified stop word dictionary had an accuracy 0.97% lower than the one trained with the standard NLTK stop words (table 6.2) and was therefore not used in the final CoVacSenti algorithm.

6.5.5 Improvements after preliminary analysis of COVID-19 tweets

Final improvements were made to the algorithm after a preliminary analysis of the COVID-19 tweet data. I found that 23.5% of tweets were classified as positive and 76.5% were classified as negative. Initial scatter plots reflect this strong negative polarity, as the majority of the scatter plot points were below the x-axis (figure 6.20).

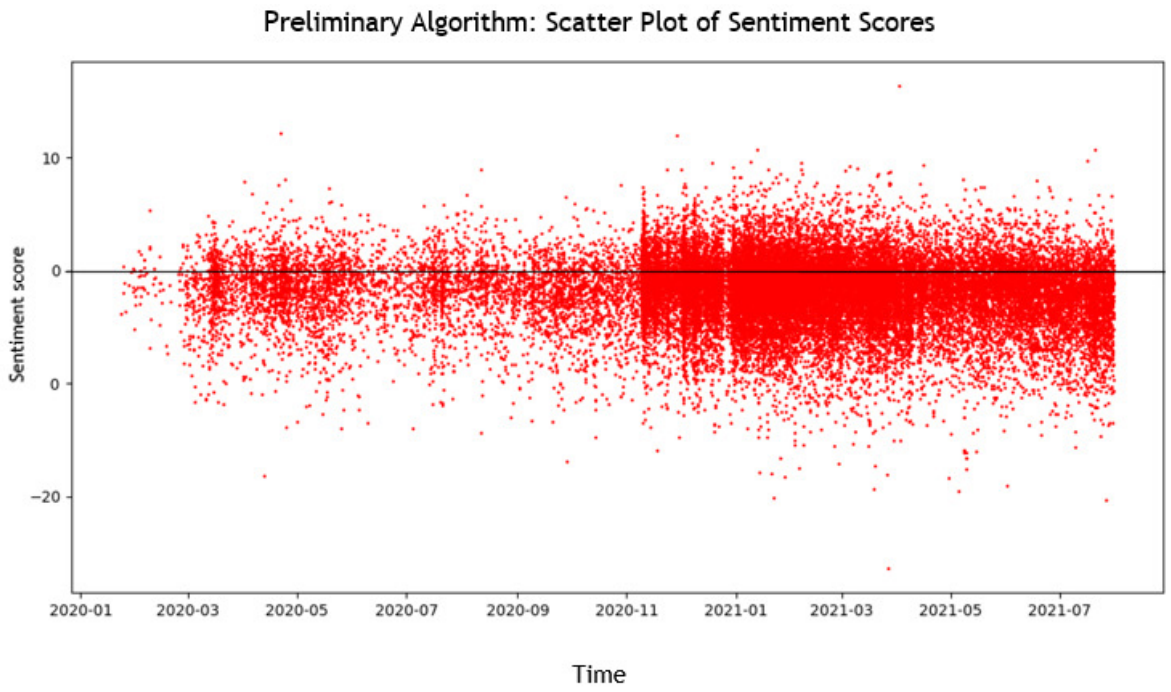


Figure 6.20 : Scatter plot of preliminary CoVacSenti sentiment scores

This was unexpected, and it caused me to compare the raw data with the processed data tables to determine whether these strong sentiments were an accurate reflection of the tweets.

The negative skew was due to a strong negative sentiment associated with key terms relevant to the pandemic, primarily ***vaccine***, ***vaccines***, ***vaccination***, and ***virus***. These terms were represented more in the negative corpus than in the positive corpus; for instance, virus appeared 352 times in the negative corpus and only 93 times in the positive one. This caused tweets with actual positive sentiments towards vaccination to be misinterpreted due to them referencing the pandemic. The following are examples of the types of positive sentiments that were misinterpreted:

Stories about trial breakthroughs make me think we can beat the virus! Who else is ready to get vaccinated? #pfizer

Cambridge laboratory methods isolate the virus; a major step in creating a vaccine

The terms ***vaccine***, ***vaccines***, ***vaccination***, and ***virus*** were removed from both corpora to assure that they were neutral terms, as they are essential to the context of the pandemic and therefore are not inherently negative in their use. The removal of these terms did not result in a change the algorithm accuracy.

The COVID-19 tweets underwent pre-processing and run through this final sentiment analysis algorithm, and the polarity score and classification was recorded for each tweet (figure 6.21).

```
def analyse_covid_tweets():

    for entry in Data.query.all():
        existing_entry = ProcessedData.query.filter(ProcessedData.id==entry.id).all()
        if len(existing_entry) != 0:
            print(f"-- TWEET ID {entry.id} HAS ALREADY BEEN ADDED --")
            continue
        p = process.ProcessTweets(entry.tweet)
        data_id = entry.id
        hashtags = entry.hashtags
        date = entry.date_time
        processing_output = p.run_text_processing()
        tweet = " ".join(processing_output[0])
        category = processing_output[1]
        prediction = nb.predict_sentiment(tweet)
        sentiment_score = 0

        if prediction > 0:
            sentiment_score = 1
            print(f" ----- POSITIVE SENTIMENT  :: {sentiment_score} ------ ")
        else:
            print(f" ----- NEGATIVE SENTIMENT  :: {sentiment_score} ----- ")

        new_entry = ProcessedData(id = data_id, date_time = date, tokenized_tweet = tweet, hasht
ags = hashtags, category = category, sentiment_score = sentiment_score, p = prediction)

        db.session.add(new_entry)
        db.session.commit()
        print("---------- TWEET ADDED TO MYSQL ----------")
```

Figure 6.21 : Analyse Covid Tweets function

6.6 KEY EVENTS IMPLEMENTATION

6.6.1 Sourcing key events data

Key events dates were sourced from governmental and independent resources, including the British Foreign Policy Group (Aspinall 2020) and the House of Commons Library (Priddy 2021). Key events were chosen based on their alignment with four main categories to examine any general trends across event types:

1) **National lockdowns**, including:
    a. Starts and ends of lockdowns
    b. Announcements of changes to lockdown measures
2) **NHS Strategy** regarding COVID-19 vaccination rollout, including:
    a. Announcements and updates
    b. Rollouts for age-banded categories
3) **Specific vaccine** news, including:
    a. Approval of Pfizer, Moderna, AstraZeneca, and Janssen vaccines
4) **Health concerns**, including:
    a. Headlines regarding blood clots after AstraZeneca vaccination
    b. Change in vaccination strategies in response to health concerns

Twenty-three key events were recorded across 19 days of the pandemic, an average of one event per each of the 19 months in the CoVacSenti timeline. There were more events than days with events as there were four days where two events occurred. Similarly, the categorisation into the four main categories was not binary as the events were multifaceted, so there were several events that fell into multiple categories. A full list of key events and their details can be found in appendix 1.

The collation of daily totals of sentiment numbers the day before and day of each event were conducted in the Analysis script. The first step was the creation of a dictionary with the key as the date and the value as another dictionary of the number and percent of each sentiment the day before and of each event.

Next, the daily totals were used as input into a Fisher's Exact Test, using the SciPy Stats module (Virtanen et al. 2020). This was done by iterating across each key event and placing the data into a contingency table array with the layout seen in table 6.3.

Table 6.3 : Mock contingency table for Fisher's Exact Test

	Day before event	**Day of event**
Number or Percent Positive	X_0	X_1
Number or Percent Negative	Y_0	Y_1

The Fisher's test was then conducted using the Perform Fisher function (figure 6.22); first comparing the number of each sentiment before and after the event, and then the percentage of sentiment.

```
#function to perform fisher analysis
def perform_fisher(stats_data):

    fisher_list = []
    for date in stats_data:

        date_dict = {"date":date}

        #create contingency table for each date

        print("----------")
        print(f"ANALYSIS OF NUM TWEETS FOR {date}")

        top_l = stats_data[date]["day_before"]["numPosTweets"]
        top_r = stats_data[date]["day_of"]["numPosTweets"]
        bot_l = stats_data[date]["day_before"]["numNegTweets"]
        bot_r = stats_data[date]["day_of"]["numNegTweets"]

        table = ([top_l,top_r],[bot_l,bot_r])
        date_dict["p_num"] = fisher_exact(table)[1]

        print("----------")
        print(f"ANALYSIS OF PERCENT TWEETS FOR {date}")

        top_l = stats_data[date]["day_before"]["percentPosTweets"]
        top_r = stats_data[date]["day_of"]["percentPosTweets"]
        bot_l = stats_data[date]["day_before"]["percentNegTweets"]
        bot_r = stats_data[date]["day_of"]["percentNegTweets"]

        table = ([top_l,top_r],[bot_l,bot_r])
        date_dict["p_per"] = fisher_exact(table)[1]

        fisher_list.append(date_dict)

    df = pd.DataFrame(fisher_list)
    df = df.melt(["date"])
    df.to_csv("data/processed/fisher_p_vals.csv",index=False)
```

Figure 6.22 : Perform Fisher function

6.7 Dashboard Prototype

6.7.1 Pre-production

The dashboard implementation began with the creation of a wireframe diagram and test case so that the creation process could be guided by an idea of how the final product would look. This also assured that all the base functionalities of the dashboard were met, to avoid spending excessive time adding features irrelevant to the project goals.

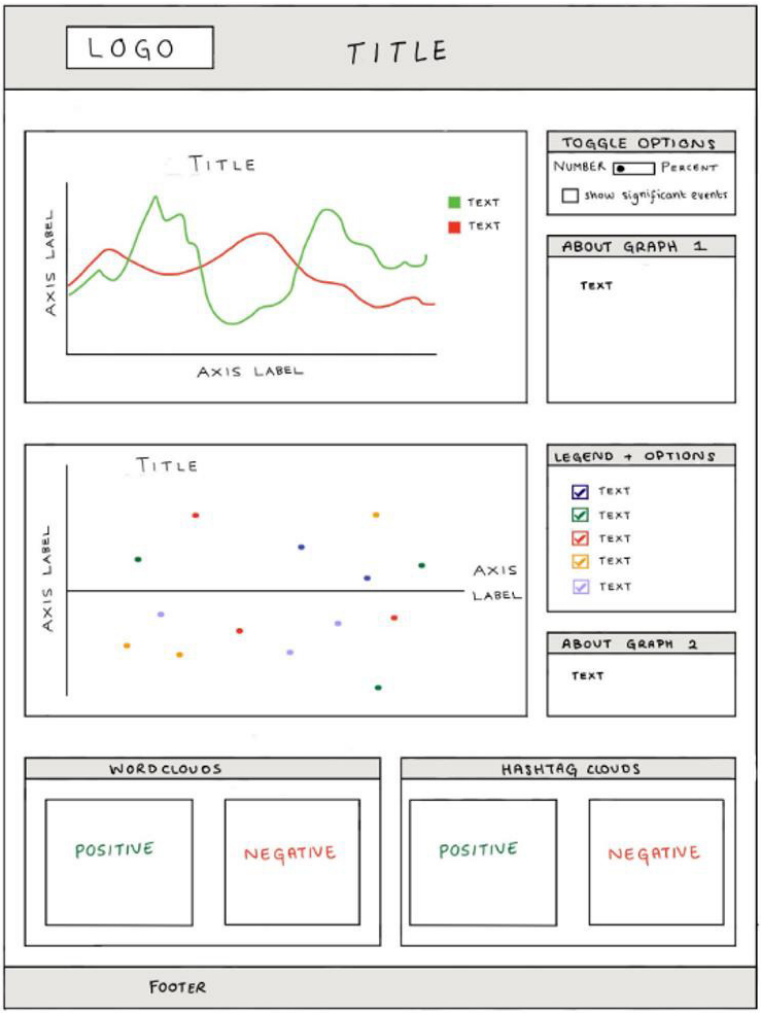


Figure 6.23 : Wireframe diagram

The wireframe diagram (figure 6.23) was based on the principles of data visualisation as discussed in the approach so the visual communication was as effective as possible. The wireframe diagram was also used within the test case to exemplify how the dashboard should look during use. The test case defined all the features necessary for the dashboard to be considered complete, as well as all default settings for interactive elements. The first two

visualisations were the primary focus of the test case as these had interactive features, therefore it was essential that they met all user experience goals. The completed test case can be found in appendix 2.

6.7.2 Dashboard creation

The dashboard was created using HTML, CSS, and JavaScript. The dashboard was hosted locally, rather than on a third-party server as the priority of the project was to obtain the results regarding Twitter sentiments, rather than to create of a functional webpage. Adobe Spark logo marker (Adobe 2021) was used to create a logo for CoVacSenti (figure 2.24) with emoticons relevant to the pandemic.

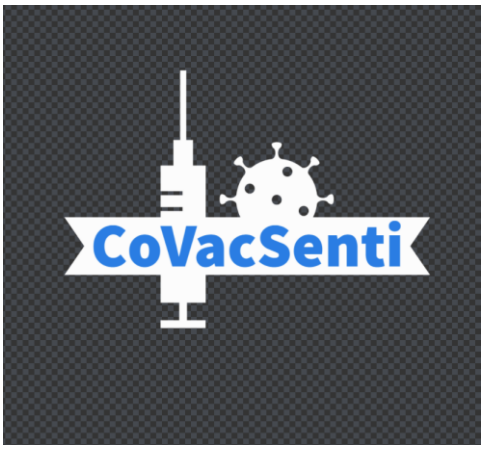


Figure 6.24 : CoVacSenti logo

The line and scatter plots were created using the JavaScript library d3 v.6 (Bostock 2013), and word clouds were created using the Python package Wordcloud (Mueller 2020). The interactive elements were controlled with JavaScript functions, event listeners, and the d3 Append, Update, and Exit pattern in d3 (Bostock 2013). For instance, the ability for users to toggle between numeric and proportional scales on the first visualisation was handled by an event listener, which called a plotting function that updated the graph with the appropriate column in the dataset (figure 6.25).

```
function update(selected){
    //load data
    d3.csv('data/processed/all_daily_totals.csv').then((data) => {
    //console.log(data)

    if (selected == "num"){
        //filter data by only including number of positive and neg tweets each day
        dataSelection = d3.group(data.filter(function(d) { return d.variable == "numPosTweets" || d.variable ==
                        "numNegTweets" }).sort((a,b)=>d3.ascending(a.date,b.date)), d => d.variable);
    }else{
        //filter data by only including % of positive tweets (neg. tweets implied)
        dataSelection = d3.group(data.filter(function(d) { return d.variable == "percentNegTweets" }).sort((a,b)=>
                         d3.ascending(a.date,b.date)), d => d.variable);
    };
```

Figure 6.25 : Data filtering lines from the update function of the first visualisation

The ability for users to select which vaccine types to display on the second visualisation was more complex, as it involved filtering rows in the dataset rather than switching between columns. This was implemented using an event listener that called a recursive function to filter the dataset based on checkbox selection before plotting (figure 6.26).

```
 function filterRec(dataSet, cIndex){
        //recursively filter the data
        if (cIndex < choices.length){
            filteredData = dataSet.filter(function(d) { return d.variable != choices[cIndex] });
            return filterRec(filteredData,cIndex+1);
        }else{
           return dataset;
        };};


    if (choices.length != 0){
        //recursively filter the data
        var filtered = filterRec(data,0);
        scatter(filtered);
    }else{
        //plot the whole dataset
        scatter(data);
    };
```

Figure 6.26 : Filter Rec function, as part of the Update function of the second visualisation

7 PRODUCTS

7.1 SENTIMENT ANALYSIS ALGORITHM

The final sentiment analysis algorithm had an accuracy of 79.4%, and the following normalised confusion matrix.

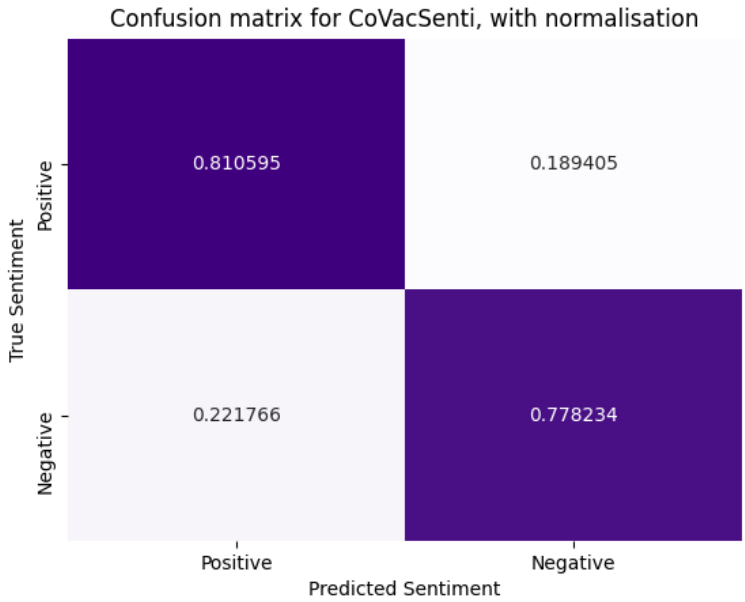


Figure 7.1 : Confusion matrix

Confusion matrices reflect an algorithm's ability identify positive and negative sentiments correctly. Normalisation was used in figure 7.1 to reflect the prediction potential of the algorithm based on the true sentiments of the testing dataset. The accuracy of an algorithm is the mean of the proportion of true positives and negatives; therefore, these values were of equal distance to 79.4%

The algorithm had a slight positive skew; both the true and false positive rates were 3.24% higher than their negative counterparts. However, the strong difference (both qualitatively through the tint of the quadrants, and quantitatively based on proportions) between true and false results reflected that this skew did not greatly impact sentiment predictions.

7.2 DASHBOARD

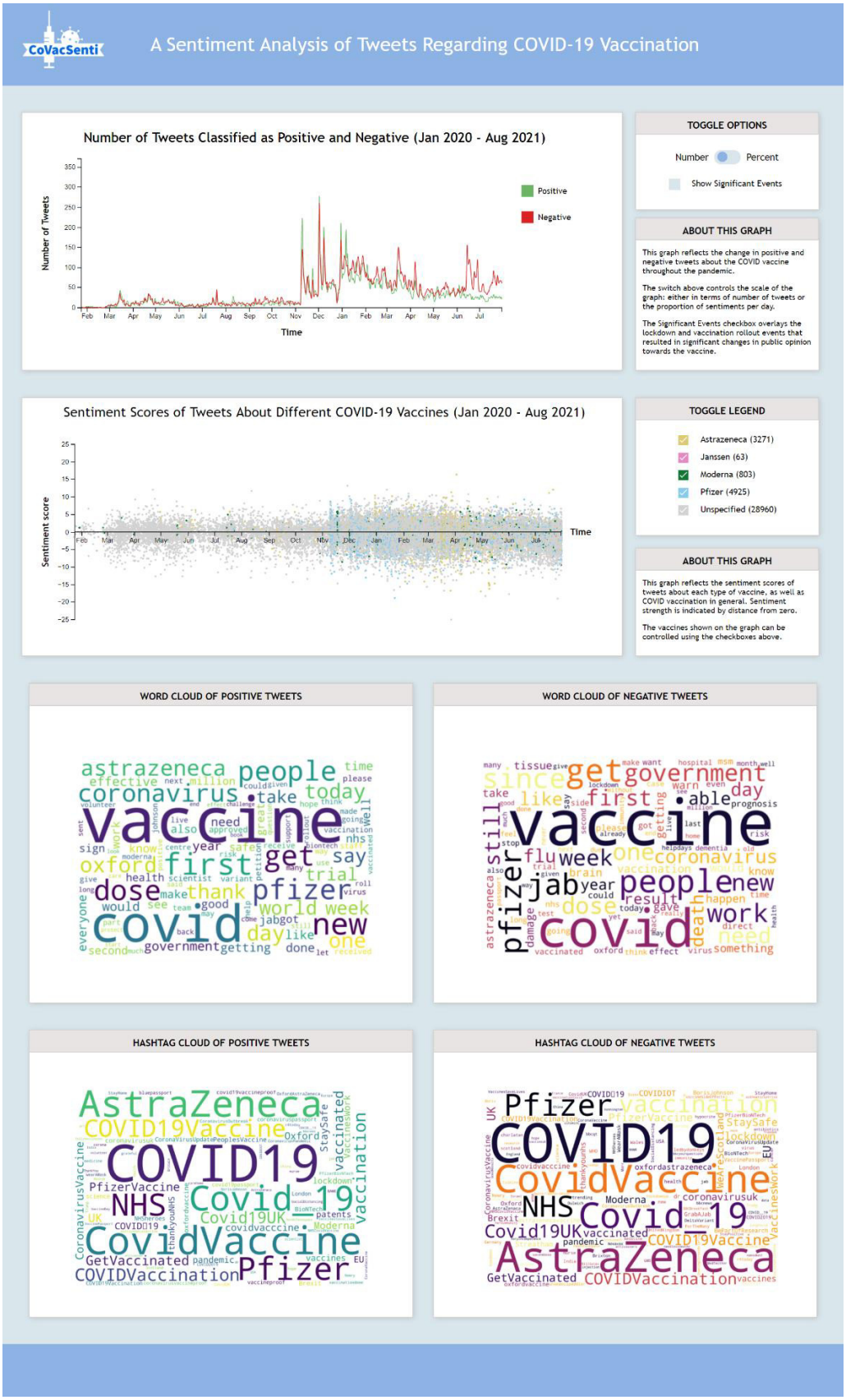


Figure 7.2 : Screenshot of the CoVacSenti dashboard

The dashboard was implemented successfully, as per the functionality requirements of the test case (appendix 2). The full dashboard (figure 7.2) included all interactive features specified in the test case, such as the ability to toggle between representing daily sentiment numerically or proportionately, the ability to show key event markers and hover on them for details, and the option to select results pertaining to only specific brands of vaccine (figures 7.3 and 7.4). The wireframe diagram (figure 6.23) was followed for the majority of the dashboard, however the word clouds were ultimately ordered horizontally, rather than vertically, to improve their clarity.

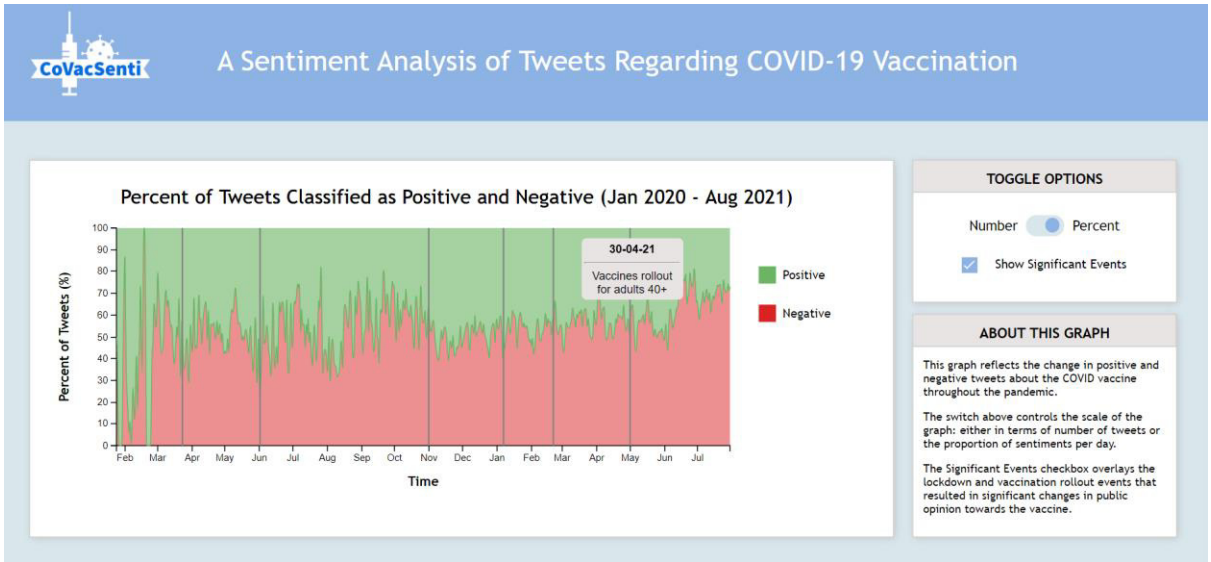


Figure 7.3 : Dashboard interactive features – daily sentiment plot toggle, key events selection, hover tooltips

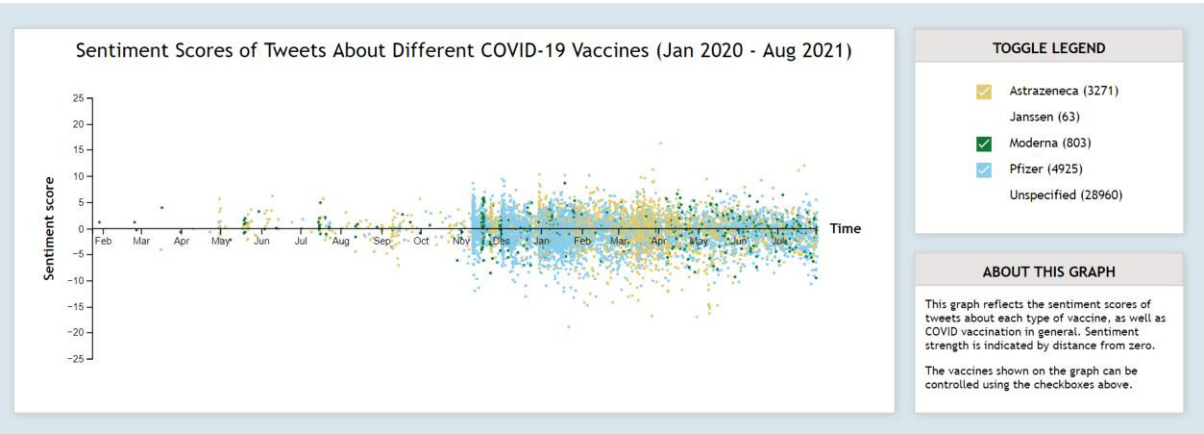


Figure 7.4 : Dashboard interactive features – vaccine type selectors

The other functional requirement, accessibility for user with colour blindness, was tested with an online colour simulator tool (Colblindor 2016), which showed that the categorical colour choices used across all the graphs remained distinct across all simulations. Results for the two most common types of colour blindness (National Eye Institute 2019) are shown in figure 7.5.

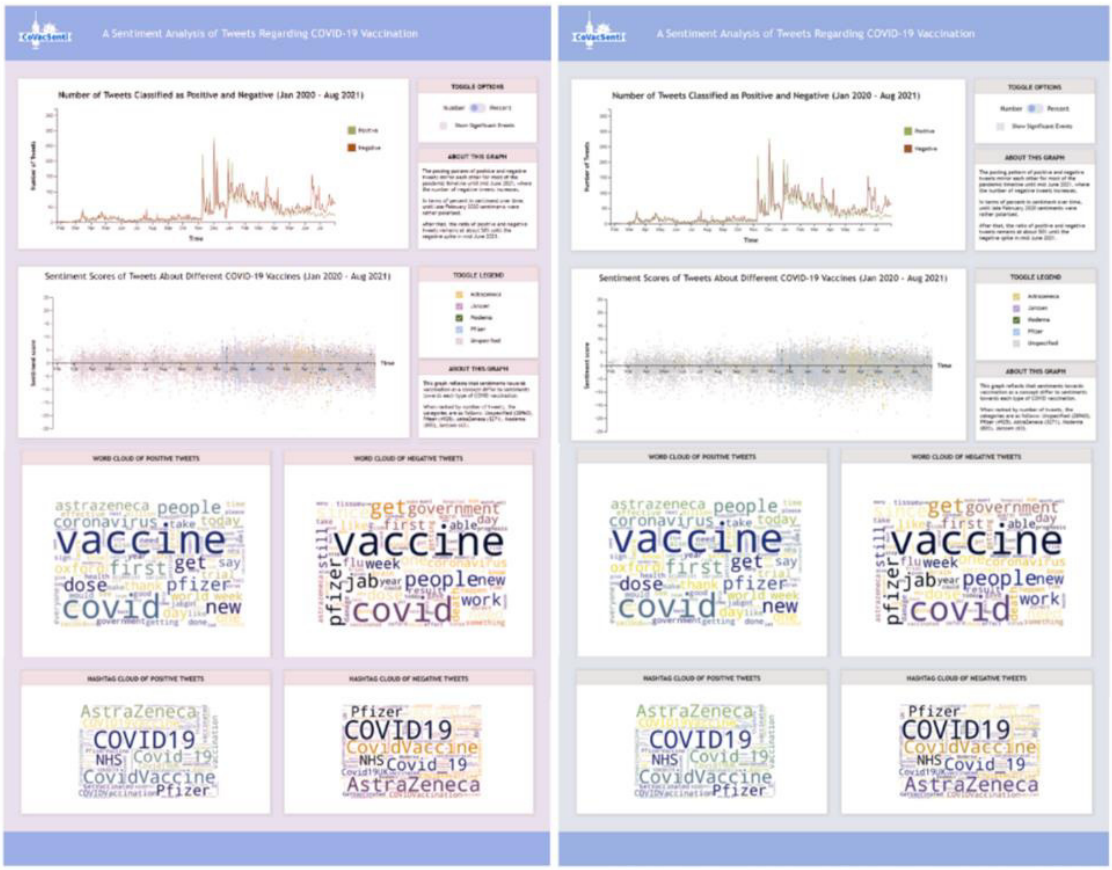


Figure 7.5: Results of Deuteranomaly and Protanomaly vision simulations

8 FINDINGS

8.1 Sentiments Regarding COVID-19 Vaccination

28,960 tweets were retrieved that contained the keywords vaccine along with either COVID or coronavirus. With the addition of the branded vaccine categories, the size of the dataset increased to 38,022 tweets. Despite data collection starting from the 1st of January 2020, tweets mentioning the vaccine did not begin until the 23rd of the month.

Out of the tweets, 16,576 (44.3%) were classified by CoVacSenti as positive, and 20,817 (55.6%) were classified as negative.

This reflects that most tweets regarding vaccination were negative, yet was not such a sharp negative skew as was previously found before removing the pandemic specific terms from the corpora (section 6.5.5).

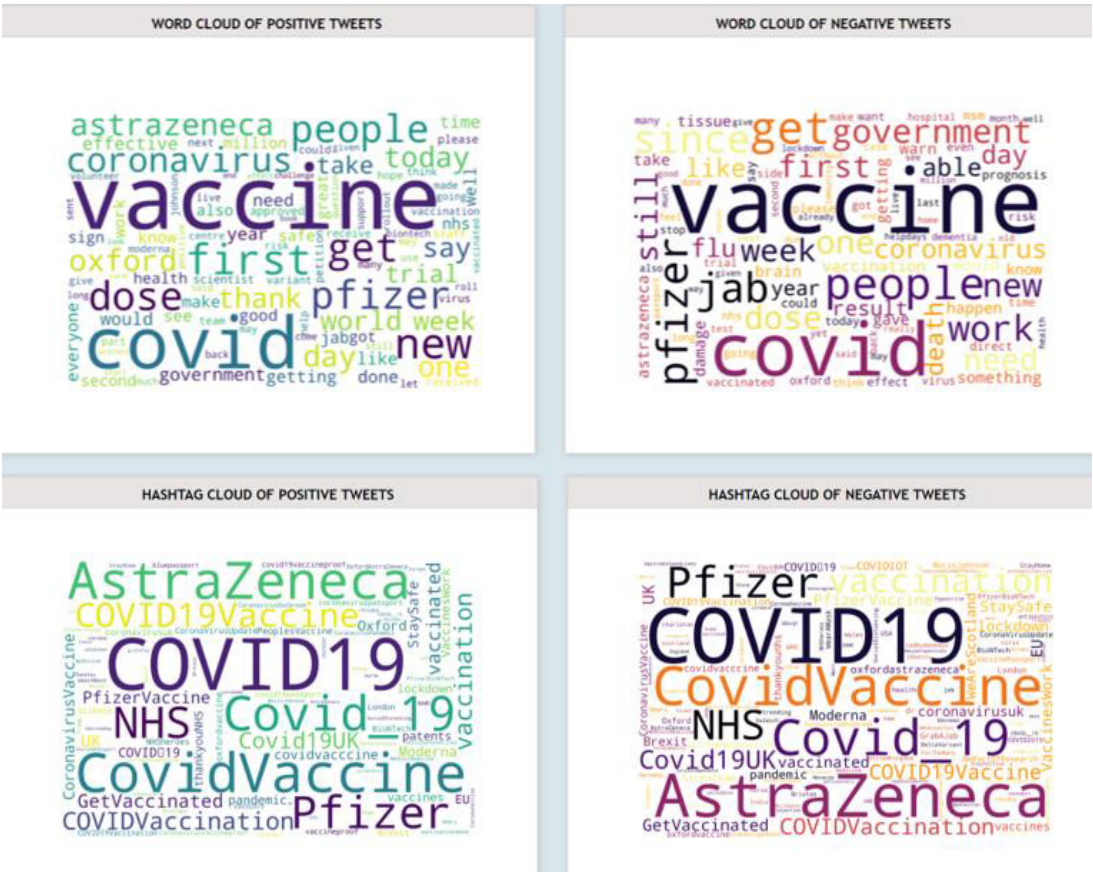


Figure 8.1 : Word and hashtag clouds of commonly used terms in positive and negative tweets, as taken from the dashboard

A word cloud analysis showed an overlap in the common terms used in both positive and negative tweets, including *vaccine*, *COVID*, *dose*, and the names of the types of vaccines. However, the positive word cloud has more occurrences (as can be seen from their size) of terms like *thank*, *effective*, and *trial*, whereas the negative cloud has more occurrences of *flu*, *government*, *tissue*, and *jab*.

8.1.1 Daily Totals Over Time

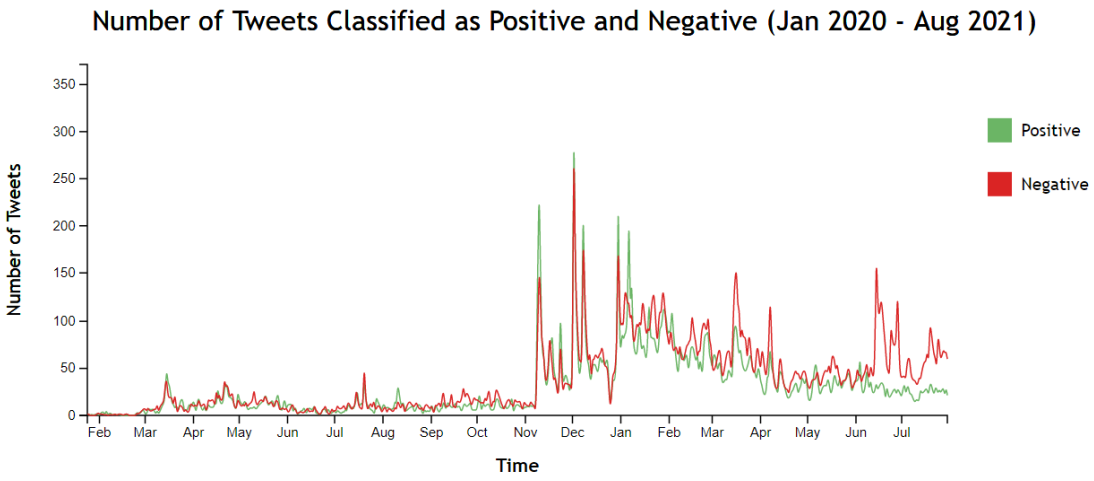


Figure 8.2 : Number of positive and negative tweets over time

The plot of number of tweets over time shows that the posting pattern of positive and negative tweets match each other for most of the pandemic timeline. There were a few times where one sentiment dominated (for instance mid-November 2020 and mid-March 2021), yet these were only in momentary peaks. The number of positive and negative tweets remained similar until midway through June 2021, where the number of negative tweets experienced several large peaks and the two lines diverged.

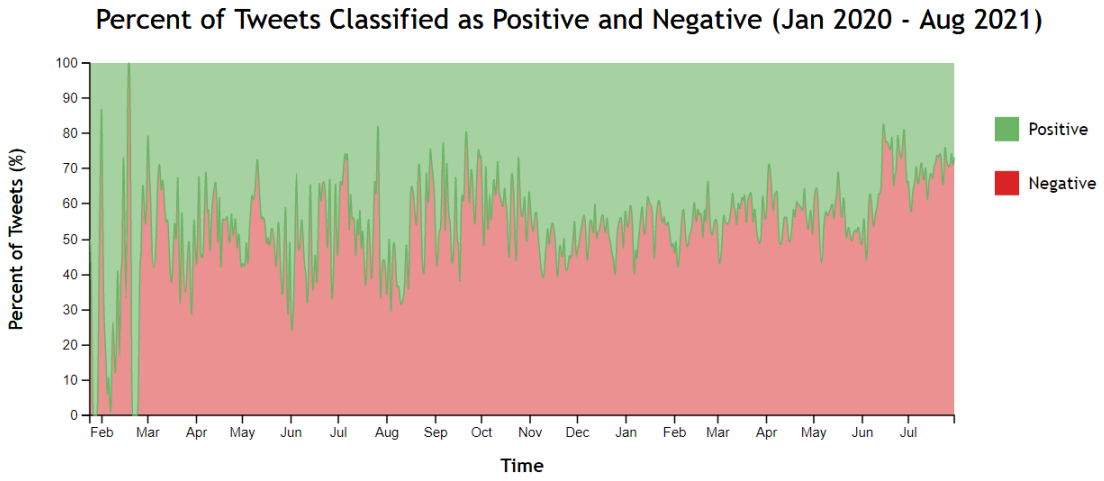


Figure 8.3 : Percent of positive and negative tweets over time

In terms of the change in percent in sentiment over time, until late February 2020 sentiments were rather polarised, though this was due to the small number of tweets posted during that time (figure 8.2). After that, the ratio of positive and negative tweets remained constant, despite daily oscillations, until the divergence in mid-June 2021.

8.2 Key Events Findings

Out of the 24 key events, four resulted in significant changes in the number of positive and negative tweets and six resulted in significant changes in the percentage of sentiments expressed (table 8.1).

Table 8.1: Significant results of Fisher's Exact Test

Date	Event	Category	P-value (number of tweets)	P-value (percent of tweets)
23/03/2020	Lockdown 1 begins	Lockdown	**0.0188**	**2.14×10^{-13}**
01/06/2020	Schools reopen after lockdown 1	Lockdown	0.0667	**2.04×10^{-7}**
31/10/2020	Lockdown 2 announced	Lockdown	0.451	**0.0214**
06/01/2021	Lockdown 3 begins	Lockdown	**8.65×10^{-8}**	**0.00160**
20/02/2021	Updates given on NHS vaccination strategy / Announcement of plans to leave lockdown 3	NHS strategy / Lockdown	**0.00638**	**0.0150**
30/04/2021	Vaccination rollout for all aged 40+	NHS strategy	**0.0421**	**0.0320**

Three of the six events were in the lockdown category, one was in NHS strategy, and one was in both categories. None of the events in the health scare or specific vaccine categories resulted in any change in the number of positive or negative tweets or the percentage of positive or negative tweets posted that day. A full documentation of p-values for all key events can be found in appendix 3.

8.3 Types of Vaccine

The visualisation of sentiment score for every type of vaccine are shown in figures 8.5 to 8.9, screenshots from the interactive visualisation on the dashboard that have had their legends and titles modified for clarity. There is an even spread of positive and negative sentiments, with a gradual increase in tweet frequency occurring over time. When ranked by number of tweets, the categories are as follows: Unspecified (28960), Pfizer (4925), AstraZeneca (3271), Moderna (803), Janssen (63).

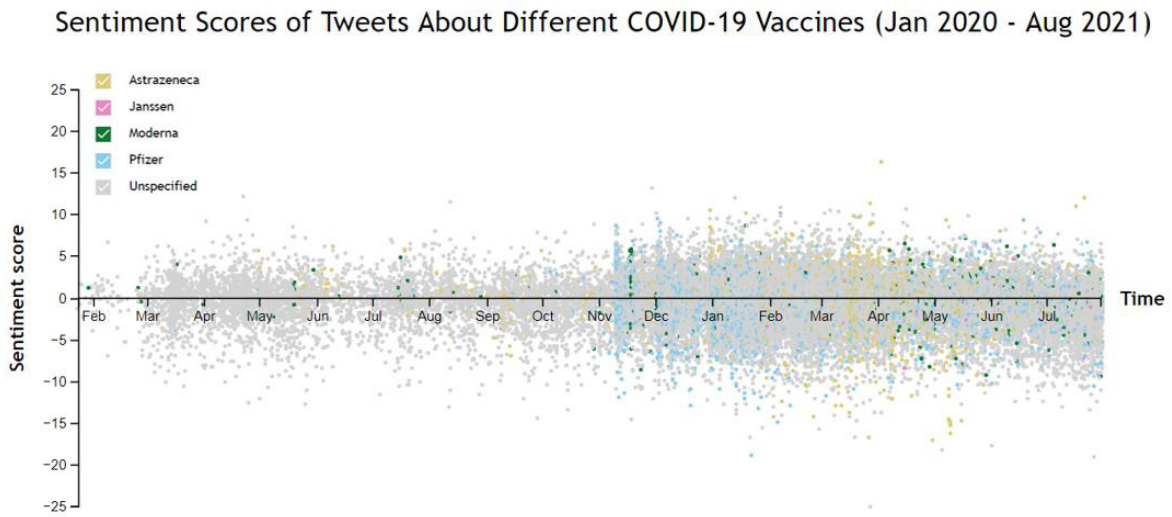


Figure 8.4 : Sentiment scores about all types of COVID vaccines and nonspecific, general COVID vaccination

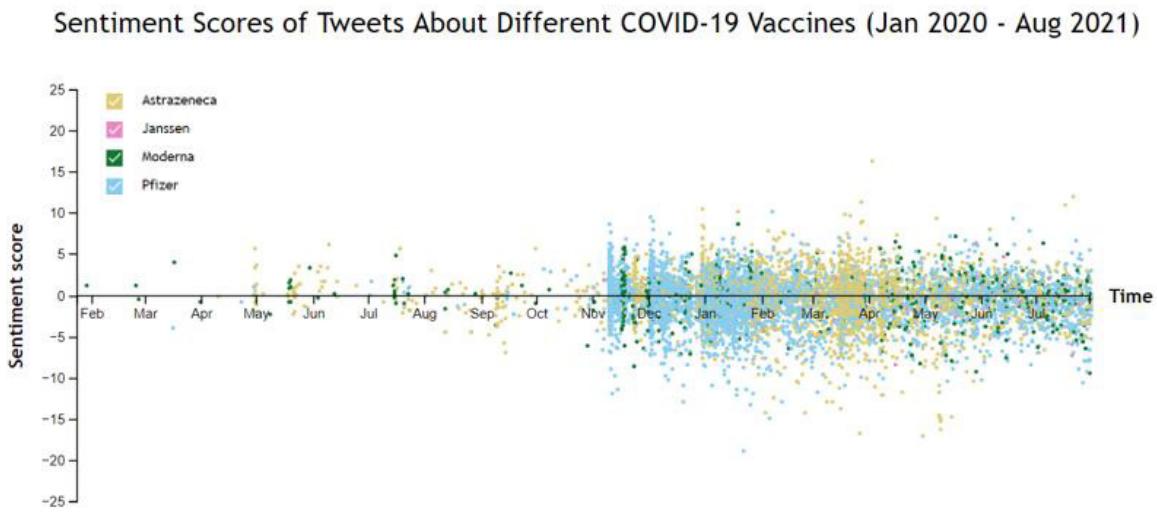


Figure 8.5 : Sentiment scores about all types of COVID vaccine

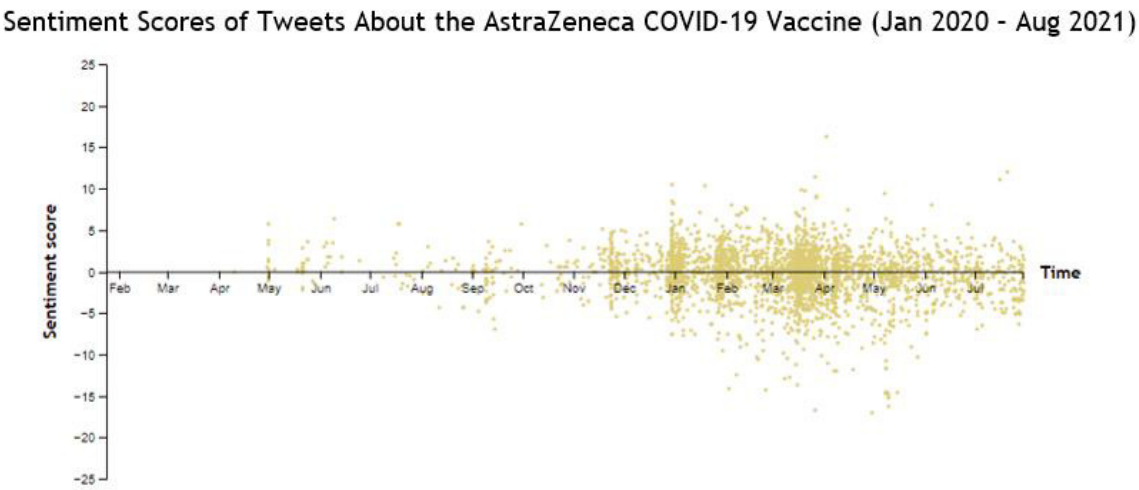


Figure 8.6 : Sentiment scores regarding the AstraZeneca vaccine

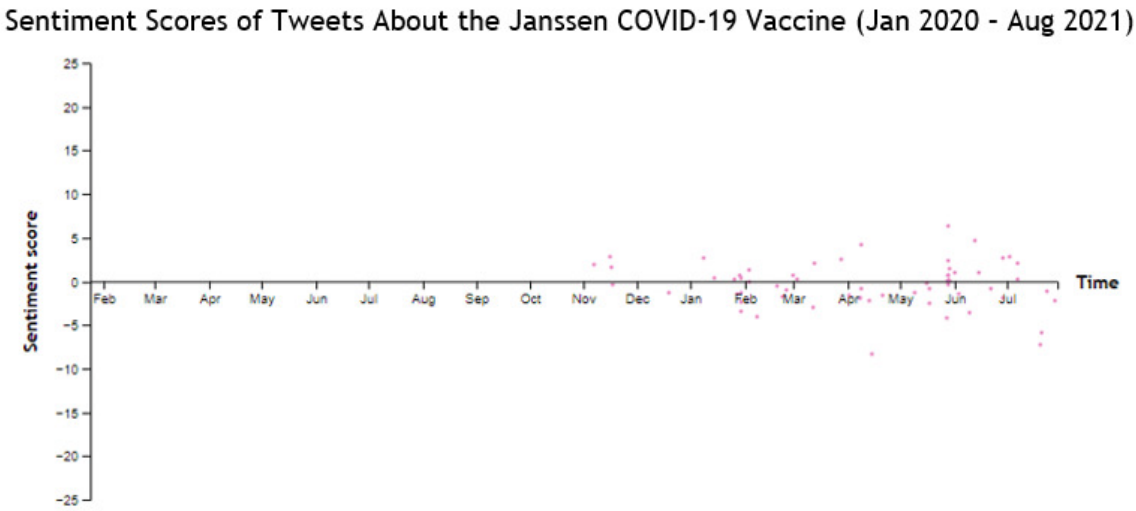


Figure 8.7 : Sentiment scores regarding the Janssen vaccine

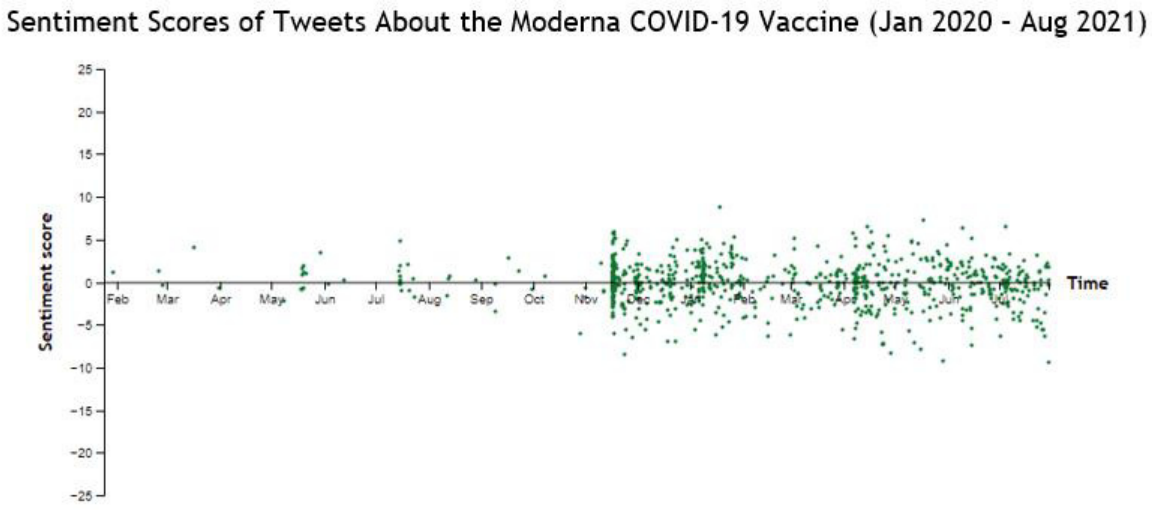


Figure 8.8 : Sentiment scores regarding the Moderna vaccine

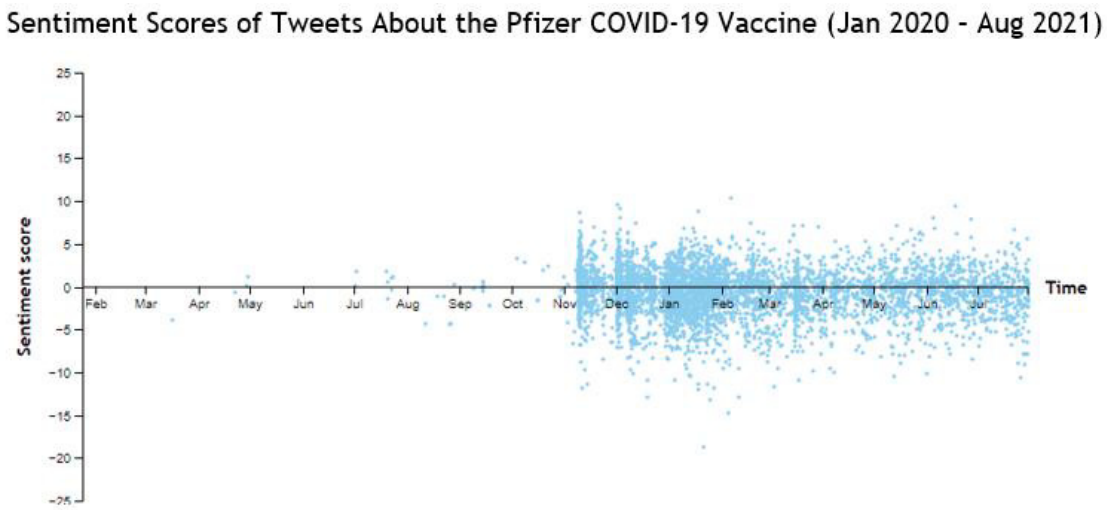


Figure 8.9 : Sentiment scores for tweets regarding the Pfizer vaccine

The first brand of vaccine to receive tweets was Moderna, receiving a positive tweet in late January 2020. Tweets about the general concept of vaccination were more widespread at the start of the pandemic, and then there was a spike in tweets regarding all specific types of vaccine (besides Janssen) in mid-November 2020, with a visible even spread sentiments. Pfizer gained a substantial number of tweets at this time and became the dominant brand of vaccine in discussion for the remainder of the period. Conversely, the number of tweets regarding AstraZeneca declined out over time, with numerous strongly negative tweets occurring between February and June 2021.

9 ANALYSIS

9.1 MEASURES OF SUCCESS

The final iteration of CoVacSenti had an accuracy of 79.4%, slightly above the measure of success set based on the median accuracy of Naïve Bayes classifiers in the field, 79.1% (Siebert et al. 2019). This reflects that the CoVacSenti algorithm performed at the expected level for research-grade sentiment analysis. Additionally, the algorithm outperformed the most popular package alternative, the NLTK, which had an accuracy of 75.8% when trained with the same data. This 3.6% outperformance by CoVacSenti marks a success, reflecting that the implementation of the Bayes' theorem may be slightly different within the NLTK functions.

While data was successfully collected from Twitter users in the UK regarding the COVID vaccine, it was surprising to see the small size of the dataset. With the collection of 500 tweets per day across the 577-day timeframe, it could be predicted that the dataset would have between 144,250 to 288,500 tweets. Similar studies had collected over 700,000 tweets regarding COVID vaccination across four months of the pandemic (Marcec and Likic 2021), so I believed these expectations to be appropriate. However, only 38,022 tweets were collected across the 19 months. This was because CoVacSenti was focused on the sentiments of the British public, therefore all tweets needed to originate from within the UK. The use of geotags is an optional, rather than a default, setting on tweets, and as a result are not widely used. In 2014 less than 0.7% of users actively add geotags to their tweets (Graham et al. 2014). This suggests that the size of the dataset is an accurate reflection of tweets tagged to the UK, as the data pool is limited by the number of tweets using this unpopular feature. The retrieval of tweets regardless of location would allow for a larger dataset but it would render the data inapplicable to the research questions regarding the British public's response to the UK pandemic timeline, such as feelings towards the different brands of vaccine offered by the NHS.

9.2 Sentiments Over Time

Considering the successes with the algorithm's accuracy, CoVacSenti provides a robust solution to determining the sentiment of tweets regarding the COVID-19 vaccine. The plots of number and percentage of sentiment category against time reflect the dynamic nature of the opinions, supporting the prediction that attitudes would change over time akin to flu vaccines (Piedrahita-Valdés et al. 2021). Most tweets regarding COVID-19 vaccination were found to be negative, reflecting a similar proportion of sentiments regarding the Indonesian vaccination scheme (Pristiyono et al. 2021). Viewing the results based on the total percentage seems to contradict contextually relevant studies, such as an analysis in the UK and US between March and November 2020 that found most tweets about COVID vaccination to be positive (Hussain et al. 2020). While this may be at odds with the overall sum of positive tweets detected by CoVacSenti, the time series graph (figure 8.3) shows that CoVacSenti detected a larger proportion of positive tweets across these four months, particularly in late March, early June, and early August 2020. This reflects the importance of long-term analysis, as the majority of negative tweets were posted after August 2020 as the NHS vaccination strategy was announced. I believe that, based on the alignment with existing research, CoVacSenti was able to determine the sentiments of tweets containing the COVID-19 vaccination keywords accurately.

However, an accurate analysis of polarity does not reflect that tweets necessarily directed that polarity towards the vaccine. Even though vaccination headlines can be accurately classified in sentiment analysis (with pro-vaccination headlines as positive and anti-vaccination as negative (Xu and Guo 2017)), the majority of headlines regarding COVID vaccination have been negative due to the polar skew associated of common pandemic terms (Aslam et al. 2020). This was noted in the creation of CoVacSenti, as the presence of terms like *vaccine* and *virus* in the negative corpus resulted in the detection of strong negative sentiment in seemingly positive tweets (section 6.5.5). Despite addressing this, the word cloud of common terms used in negative tweets (figure 8.1) reflects the skew that medical terms have in sentiment analysis, with terms like *flu*, *tissue*, and *prognosis* appearing fairly frequently, while the positive word cloud contained fewer medical terms. This raised the issue as to whether CoVacSenti could appropriately address the problem of how the British population feels about the vaccine.

9.2.1 Suitability of solution in addressing the problem

Studies into anti-vax Facebook groups have found that posts by users with pro-vaccination beliefs were detected as being significantly more negative by sentiment analysis algorithms than actual anti-vax posts (Klimiuk et al. 2021). This suggests that the strong negative sentiments detected by CoVacSenti could be people expressing anger towards those not wishing to get vaccinated, rather than expressing negativity towards the vaccine itself. The terms *please*, *people*, *damage*, and *death* appear in the negative word cloud (figure 8.1), yet a tweet containing these terms could easily be used to express negative sentiment towards those who are unvaccinated. This theory is supported by the negative hashtag cloud (figure 8.1) as *#COVIDIOTS*, an insult for those who go against public health advice, occurs fairly frequently along with other pro-vaccination terms such as *#GetVaccinated*, *#VaccinesWork*, and *#ThankYouNHS*. This suggests that a large proportion of negative tweets are expressing negativity towards the anti-vax population, rather than against vaccination itself.

Other commonly used terms in the negative hashtag cloud, such as *#EU*, *#Brexit*, *#WeAreScotland*, and *#Lockdown* suggest that not all tweets containing the COVID keywords necessarily were focused on the concept of vaccination itself. Anger towards lockdowns or vaccination supplies may have reignited polarising political debates, which would have been detected by CoVacSenti as negative despite not necessarily expressing negativity towards vaccination. This could also explain why there was a dramatic increase in negative sentiments in mid-June 2021 (figures 8.2 and 8.3) as this was leading up to *Freedom Day* on the 19th of July – the end of all of the third lockdown's restrictions across the UK – a polarising topic at the time given the rising case numbers.

9.3 Key Events Analysis

Of the 24 key events, four resulted in significant changes in the number of polar tweets, and six resulted in significant changes in their percentage (table 8.1). However, it can be seen from the time series plots (figures 8.2 and 8.3) that there were significant peaks and valleys in the proportion of sentiments that were not detected by the Fisher analysis. I believe this was due to the scale of the impact that these events had; the method of analysing significant change based on the day before and day of key events may have been too short of a time frame to determine reactions to events. The time of each key event was not taken

into consideration, so a headline announced at the end of the day would not have received the same number of on-the-day tweets as a headline released in the morning.

The significant key events were primarily lockdown changes, including the start of each of the three lockdowns, schools reopening after the first lockdown, and the announcement of the plans to leave the third lockdown (table 8.1). The change in the number of positive and negative tweets seen on these lockdown days was likely due to more people being on their devices while isolating at home and wishing to join in the nationwide conversations about the event. Lockdown events are wider reaching than other key event types, so it is understandable why this was the category that resulted in many sentiment changes.

The only significant key event not in the Lockdown category was the rollout of vaccines for all adults aged 40+. It was surprising that this age bracket resulted in a significant change in sentiments whereas the 60+ and 18+ brackets did not. When studying the demographics of anti-MMR vaccine Twitter users, Tomeny et al. found that the bracket between 40-44 years old was the strongest age-based predictor of anti-vax sentiment (2017). Using the notion that those with pre-existing anti-vaccine beliefs would be against the COVID vaccine, this would explain why significant sentiment change was recorded for this rollout category and not the others. This suggests that those in the 40+ age range should be focused on in further research into COVID vaccine hesitancy.

Events in the Health Concerns category did not mark significant changes in the proportion or number of positive and negative tweets. This is surprising as concerns over vaccine safety and personal health are key motivators in the anti-vaccination movement (Kata 2010). This suggests that vaccine wariness was already established prior to headlines about blood clots, therefore the days before the release of clot-related headlines already had a large proportion of negative tweets and therefore resulted in no significant change.

Lastly, events in the Specific Vaccines category also did not cause any change in sentiments, suggesting that the public does not have a strong polar view of the release or administration of types of COVID vaccines in the UK. This however is refuted when examining the polarity scores of the tweets regarding different COVID vaccinations (figure 8.5).

9.4 Sentiments Towards Types of Vaccine

The analysis of sentiments towards different types of vaccine resulted in an interactive visualisation (figures 8.4 to 8.9). I believe these plot clearly shows the overall trends regarding each vaccine and users can clearly determine the polarity of the tweets along with their relative density compared to others. There were numerous spikes on all the graphs, reflecting certain days of the pandemic where vaccines were discussed.

The trends of Pfizer, Moderna, and AstraZeneca match those found in previous research (Marcec and Likic 2021), suggesting that the trends observed regarding the brands of vaccine have continued in the same patterns. As expected by its early release, the AstraZeneca vaccine was discussed first and later caused many negative sentiments when news regarding health concerns was released (Marcec and Likic 2021). Similarly, Janssen has yet to be administered in the UK, so the lack of tweets is expected.

10 CONCLUSION

10.1 ACHIEVEMENTS

In conclusion, this study was successful in achieving its aim of creating a sentiment analysis system that determined how the UK population felt towards COVID-19 vaccination. This was done by successfully meeting the following objectives of the study.

10.1.1 Collect tweets regarding COVID-19 vaccination from the UK public

The data collection methods successfully collected tweets originating from within the UK. Despite the fact that this was a smaller dataset than initially expected, the size was reflective of the small proportion of Twitter users who geotag their posts.

10.1.2 Create a sentiment analysis algorithm that is independent from existing sentiment analysis package solutions

Using training data from Sentiment140, I created a Naïve Bayes classifier that could determine the sentiment of tweets. This performed with a higher accuracy than the commonly used package, the NLTK, and had higher accuracy than the threshold of success set by the median accuracy of Naïve Bayes algorithms used in academic research.

10.1.3 Investigate how attitudes towards COVID-19 vaccination have changed over time

CoVacSenti was able to show how the number of positive and negative tweets followed similar patterns and only deviated from each other on a few occasions. The difference between the number of polar tweets and the proportions of those polarities reflected that both measures should be used in conjunction in public health research, as they can lead to incomplete conclusions when used alone. Visualising these attitudes over time allowed trends throughout the pandemic to be clearly seen.

10.1.4 Examine how sentiments differ towards the specific vaccines - Pfizer, Moderna, AstraZeneca, and Janssen

The analysis of attitudes towards specific vaccinations found that the number of tweets for each brand of vaccine varied widely. Tweets about the AstraZeneca were more negative than those about the other types of vaccines, which had fairly even spread in tweet polarity.

10.1.5 Evaluate whether key events in the COVID-19 pandemic influenced the public's sentiments towards COVID-19 vaccination

This study found several key events that resulted in statistically significant change in sentiments. Lockdown news is the most influential type of event in the public eye, and future NHS strategies to vaccinate the remaining population should target those in the 40+ age category. Headlines about health worries did not have a significant impact on the daily number of tweets, suggesting anti-vax sentiments were present and strong before news about vaccine safety was broken. Additionally, the release of new vaccines in the UK did not result in a change in sentiments, indicating that the NHS should not expect changes in public opinion as the new Janssen vaccine is administered.

10.1.6 Develop a dashboard prototype to display the findings of CoVacSenti

The dashboard that was developed was successful at displaying the results of CoVacSenti and all functionality goals, as per the use case, were achieved.

10.2 DRAWBACKS AND IMPROVEMENTS

Despite the successes of this project, there are several limitations in the approach that, if addressed, would have greatly improved the strength of the findings.

10.2.1 Data

A common filtration technique in the field is to test for subjectivity before performing sentiment analysis. This would determine whether tweets are objective facts (e.g., "*100 new coronavirus cases today in London, according to GOV.UK*") or subjective opinions (e.g., *"Happy to see the cases in London finally declined to 100 today"*). As news headlines are shared on Twitter, the strong negative sentiment associated with objective news regarding the pandemic (Aslam et al. 2020) would have caused the number of negative tweets each

day to increase. As this study aimed to investigate the general public's feelings towards the vaccine, the presence of negative objective tweets limits the conclusions that can be drawn from these findings.

An additional way to improve the integrity of the dataset would be the removal of *retweets* – a tweet from one user that has been reposted by others. When a user retweets a tweet, they create a duplicate of it to share with their followers. Like news headlines, the inclusion of retweets in the dataset may have caused many duplicated tweets to be analysed, influencing the proportion of individual sentiments recorded each day. This could have been avoided by filtering out retweets during the Twitter archive search, or by checking the dataset for presence of duplicates during text pre-processing. However, while the presence of multiple retweets of headlines reflects the sharing of news updates, retweets of subjective sentiments could be interpreted as the echoing of shared sentiments. Therefore, the retweet of subjective posts may not be detrimental to dataset integrity as it potentially represents many people sharing the same opinion.

10.2.2 Naïve Bayes Algorithm

The accuracy of the algorithm was compared against that of the NLTK package to assess whether this approach outperformed existing alternatives. However, if time allowed it would have been beneficial to compare the accuracy with different types of supervised machine learning algorithms using NLTK or SciKit-Learn (Pedregosa et al. 2015), which would have strengthened my argument for choosing to base CoVacSenti on a Bayesian algorithm.

I also believe that the corpora created as part of the Naïve Bayes classifier could have been improved in two ways. Firstly, the negation of words was handled poorly by my created algorithm. For a tweet "*I don't trust doctors*", the stopword "*don't*" would be filtered out and "*trust*" would be added to the negative corpus. However, this removes the negation of the word, a key indicator of the sentiment of the tweet. This could have been resolved by indicating that the word is a preceded by a negation before adding it to the corpus. Similarly, my modified stop word corpus that accounted for the determiners *no* and *not* did not result in higher algorithm accuracy because the context of the negation was not preserved. For instance, a more complex pre-processing algorithm would have the potential results shown in table 10.1.

Table 10.1 : Pre-processing outputs with negation handling

Condition	Raw tweet input	Pre-processing output
Existing algorithm	I don't trust doctors	["trust", "doctors"]
Improved negation detection	I don't trust doctors	["NEG_trust", "doctors]
Improved negation detection	I have no trust in doctors	["NEG_trust", "doctors]

Accounting for the negation of terms would result in larger corpora, but this method has proven successful in assuring the context of negated verbs is preserved during sentiment analysis (Symeonidis et al. 2018).

Secondly, the corpora could have been improved by accounting for misspelled words. The language filtration process removed quite a lot of misspellings in the dataset, but correcting misspellings rather than deleting them would have improved the reliability and accuracy of the word frequency counts even further. Similarly, there were many words with repeated words in the corpora – for instance "*hellooo*", "*helloooo*", and "*hellooooo*" were all variations that were not included in the frequency count of "*hello*". I could have included a clause to check for the successive repetition of three or more of the same characters in a row, an uncommon occurrence in English, and removed all repetition and marked the misspelling as shown in table 10.2.

Table 10.2 : Pre-processing outputs with spelling correction

Original word	Modification
Helloooo	MOD_hello
Hhhhellooo	MOD_hello
helllllllo	MOD_helo

However, it can be seen from the last example in the table that repetitions of letters that appear as a double in the correct spelling of the word would cause the modification to be misspelt too. Misspellings and characters repetitions cause difficulty in sentiment analysis as there are many unique styles of typing, and the integration of spellcheck packages and text modification could have allowed my algorithm to account for these complexities.

10.2.3 Key Events

There were several limitations in my approach to testing key event significance. As mentioned previously, a maximum of 500 tweets were collected per day. Despite the maximum not being collected on many days, this limit still may have impacted significance tests for events where there were many successive days where the total number of tweets was close to or above this limit. For instance, if there were three days in a row with 500, 1000, and 2000 tweets, the data collection process would have stopped after collecting the first 500 for each of them. Therefore, there would be no difference in the total number of tweets on each of those days, erroneously suggesting that the events were not statistically significant. This could have been counteracted by increasing the number of tweets collected per day. Having smaller time steps, such as hourly (rather than daily steps) would have also prevented false negatives that could have occurred from news headlines that were released in the evenings in the key events analysis.

Since there are many peaks in the number and proportion of positive and negative tweets yet only six significant events, another issue can also be raised with how the key events analysis was conducted. My approach involved creating a list of key events and performing Fisher's Test on each one to determine if it was significant. However, it can be seen from the plots over time that there are many sudden changes in sentiment proportions that are not associated with key events (figure 7.3). I believe it would have been beneficial to develop the key events analysis further after completing the trend plots. This could have been done by working backwards from the plots, by identifying obvious peaks and valleys in the time series, creating a list of events that could have caused these changes, and performing a statistical analysis of each. This would have created a deeper understanding of what causes changes in vaccine sentiments, allowing the results of CoVacSenti to be of more use in public health.

10.3 Future Work

The findings of CoVacSenti could be expanded in numerous way through further research. I believe it would be of merit to explore whether the sentiments recorded by CoVacSenti are a true reflection of the public's vaccine intentions. This could be approached by retrieving the NHS daily vaccination statistics and then performing a regression analysis against the

CoVacSenti time series results. This would allow public health strategists to determine how much emphasis should be placed on social media observations as the pandemic continues.

Furthermore, it would be interesting to identify what emotions are associated with positive and negative tweets. Anti-vax headlines typically express sadness, fear, anger, and trust (Xu and Guo 2017), but no research has been done into the emotions expressed towards COVID vaccination in the UK. Determining the emotions associated with negative polarity would be helpful in solving whether the high proportion of negative tweets recorded in this study was due to anger towards anti-vaxxers or a genuine fear of the vaccine.

Currently the CoVacSenti dashboard is a simple prototype, but further work could develop it into a functional website for public use. This would involve hosting the website on a server, automating daily calls to the Twitter API to retrieve new tweets, and dynamically updating visualisations. The website could also have text entry fields for visitors to write their comments about the COVID vaccine and see the polarity scores given to them by CoVacSenti. Making this data available to the general public would help spread awareness about how pro- and anti-vaccination sentiments are expressed on Twitter, which may help visitors identify these sentiments in their own social networks.

Ultimately, CoVacSenti provides a resource to learn more about how the wider community feels about vaccination. As with any public health management strategy, understanding how the population thinks is essential in allowing the UK, and the world, to combat and recover from the COVID-19 pandemic.

11 REFLECTION

11.1 THE RESEARCH PROCESS

I learnt several things in this project that will invaluably help me in the future as I aim to continue working in research.

Firstly, I learnt the value of project management techniques and time allocation. I was initially intimidated by the scale of this project but was comforted once I created a Gantt chart (appendix 4). I had never used a Gantt chart before, and the organisation of breaking up the different sections (i.e., pre-reading, data collection, text pre-processing, algorithm creation, etc.) into week-long segments provided me with security when I felt anxious. I also created a plan for each day, which I found to be very helpful when I encountered setbacks or portions of the project that were more difficult than I anticipated. Before applying this technique, I would feel demotivated and catastrophize when I fell behind schedule. These daily plans helped me avoid stress by allowing me to break daunting tasks into daily sprints and readjust these sprint goals when necessary.

I also developed confidence in my ability to perform literature searches of academic papers, a skill I have much experience in but have never used at this scale. I believe I utilised a wide range of journal and conference papers as sources in this project and strengthened my skim-reading and library search skills while doing so. I believe that the papers I found were of good authority and felt secure in using them as reliable and genuine measures of success for the algorithm.

Completing CoVacSenti also taught me the importance of data visualisation in research, as well as the strengths of the d3 JavaScript library. A few months ago, I took *Data Visualisation* as an optional module, and have since felt some regret for not choosing a more industry applicable, technical module like *Cloud Computing* or *Databases*. However, I understood the importance of the course once I began creating the dashboard. Machine learning is a daunting topic, and simple interpretable visualisations can help make the results more understandable. The ability to communicate findings is a vital skill in science

and understanding which features of a graph can engage or deter the viewer can help bridge the gap between researchers and the wider public.

11.2 The Field

One of the key lessons I learnt about the field of sentiment analysis was that it is not as complicated as I initially believed. I was pleasantly surprised to find that it is based on statistical inference and enjoyed making connections with other disciplines. I had previously used Bayes' theorem in phylogenetic analyses, so it was comforting to find a familiar method in this intimidating field. I also did not know that sentiment analysis can classify text down to the nuanced emotions expressed (such as anger, fear, sadness). I had very little experience in machine learning prior to this, and so the literature review helped me understand that the algorithms, at their core, work by simply find the most probable solution to a complex-sounding problem.

I also gained a lot of knowledge on the vaccine hesitancy and anti-vax movement. I was disgusted to learn that the anti-vax movement in the 21st Century was spearheaded by a doctor who fabricated his results so that his patented MMR vaccine could be sold instead of the existing generic vaccine. In the process he abused autistic infants with invasive, life-threatening medical procedures, and then fabricated his data when he could not find a connection between their autism and the vaccine. Reading about this case study made me appreciate the importance of research in this field, as this disgraced former doctor remains an influential and respected member of the anti-vax community to this day. With COVID cases still incredibly high in the UK, understanding the motivations behind anti-vaxxers is of pivotal importance, now more than ever. This project gave me the sobering realisation that anti-vax beliefs are often based on a person's deep-rooted morality, and, despite numerous research studies, there have not been many methods of effectively counteracting these beliefs. However, I hope that sentiment analysis can help foster more understanding about vaccine refusal, and hopefully mitigate some of its societal impacts.

11.3 Personal Growth

I learnt in this project that one of my strengths is the desire to uncover as much as possible in my research. I had been very nervous about the dissertation since the start of the course

as I did not do one in my undergraduate degree. I did not think I would be able to motivate or guide myself with such a distant deadline and without being instructed what to do each step of the way. However, I was surprised to see that I became very invested in the project and wanted to keep working on it beyond my planned hours. I found myself asking more questions and adding more topics (such as they key events analysis) as I progressed, powered by the adrenaline generated from uncovering something new. However, I realized I could not fulfil all my research desires in the project timescale, and so wrote my ideal plans in the Future Work section. This dissertation helped me uncover that, once I find an in interest in something, I become fixated and want to research it to its fullest extent.

This adrenaline from problem solving also allowed me to see that I have issues making decisions and fall victim to the sunk cost fallacy when deciding between approaches. During data collection I decided to use Twint without realising that the location search functions were not working. Yet when I found this problem, I spent almost 2 weeks trying to use a coordinate search with Twint, spending hours trying to create a centre point and radius of a circle that could catch all tweets in the UK. This was very challenging and irrelevant to the actual goals of the project, yet it was very hard for me to accept defeat and move onto other options. I was unwilling to accept that all the time and labour I had put into this non-functional package had been for nothing, so continued working on it until the wasted time was undeniable.

Similarly, I learnt that while my perfectionism is a strength in some contexts, it became a weakness when creating a program. I spent too much time focusing on small details in my scripts instead of focusing on the general goals. I found it difficult to move on from a function without adding comments, perfecting indentation, and researching solutions to make each line as efficient as possible. This, counterintuitively, was not efficient at all and meant a lot of time was wasted. I had to teach myself to be satisfied with scrappy drafts and clunky code, and trust that I would return later to polish things up prior to submission.

Finally, one of the most valuable lessons learnt in terms of personal growth was that I need to have confidence in my abilities. At the beginning I was so intimidated by the technical jargon and complexity of academic papers in machine learning that I became overwhelmed to the point of shutting down (which, of course, made the papers even harder to read). However, after completing my algorithm I returned to the same articles that demotivated

me months before and found they were not as difficult as I had once thought. My lack of confidence prevented me from accepting that it was possible to build an understanding of a new topic from the ground up. This project seemed impossibly daunting in the first few weeks and in retrospect, I am incredibly proud of what I have accomplished. Ultimately, this dissertation period taught me that my mindset is the key determining factor in my success, and that approaching a challenge with positivity will create an enriching and surprisingly enjoyable overall experience.

12 REFERENCES

Abbas, M., Ali, K., Memon, S., Jamali, A., Memon, S. and Ahmed, A. 2019. Multinomial Naive Bayes Classification Model for Sentiment Analysis. Available at: https://www.researchgate.net/publication/334451164_Multinomial_Naive_Bayes_Classification_Model_for_Sentiment_Analysis [Accessed: 1 November 2021].

Adobe 2021. Make Social Graphics, Short Videos, and Web Pages To Stand Out—In Minutes | Adobe Spark. Available at: https://www.adobe.com/express/ [Accessed: 29 September 2021].

Aslam, F., Awan, T.M., Syed, J.H., Kashif, A. and Parveen, M. 2020. Sentiments and emotions evoked by news headlines of coronavirus disease (COVID-19) outbreak. *Humanities and Social Sciences Communications* 7(1). doi: 10.1057/s41599-020-0523-3.

Aspinall, E. 2020. COVID-19 Timeline - British Foreign Policy Group. Available at: https://bfpg.co.uk/2020/04/covid-19-timeline/ [Accessed: 10 September 2021].

Balakrishnan, V. and Lloyd-Yemoh, E. 2015. Stemming and lemmatization: A comparison of retrieval performances. In: *Proceedings of SCEI Seoul Conferences*.

Bao, Y., Quan, C., Wang, L. and Ren, F. 2014. The Role of Pre-processing in Twitter Sentiment Analysis. In: *Intelligent Computing Methodologies*. Springer.

Bayer, M. 2012. *The Architecture of Open Source Applications: Structure, Scale, and a Few More Fearless Hacks*. aosabook.org.

Bird, S., Klein, E. and Loper, E. 2009. *Natural language processing with Python*. O'Reilly.

Bostock, M. 2013. D3.js - Data-Driven Documents. Available at: https://d3js.org/. [Accessed: 1 September 2021].

Bruns, A. and Burgess, J. 2015. Twitter hashtags from ad hoc to calculated publics. In: *Hashtag publics : the power and politics of discursive networks*. New York: Peter Lang, Impr

Burki, T. 2020. The online anti-vaccine movement in the age of COVID-19. *The Lancet Digital Health* 2(10), pp. e504–e505. Available at: https://www.thelancet.com/journals/landig/article/PIIS2589-7500(20)30227-2/fulltext. [Accessed: 15 October 2021].

Colblindor 2016. Coblis — Color Blindness Simulator – Colblindor. Available at: https://www.color-blindness.com/coblis-color-blindness-simulator/ [Accessed: 1 November 2021].

Cravit, R. 2019. How to Use Color Blind Friendly Palettes to Make Your Charts Accessible - Venngage. Available at: https://venngage.com/blog/color-blind-friendly-palette/ [Accessed: 11 October 2021].

Danilk, M. 2016. langdetect. Available at: https://pypi.org/project/langdetect/ [Accessed: 8 October 2021].

Darwich, M., Mohd Noah, S.A., Omar, N. and Osman, N.A. 2019. Corpus-Based Techniques for Sentiment Lexicon Generation: A Review. *Journal of Digital Information Management* 17(5), p. 296. doi: 10.6025/jdim/2019/17/5/296-305.

DeStefano, F. and Shimabukuro, T.T. 2019. The MMR Vaccine and Autism. *Annual Review of Virology* 6(1). doi: 10.1146/annurev-virology-092818-015515.

Fayyad, U.M., Piatetsky-Shapiro, G. and Uthurusamy, R. 2003. Summary from the KDD-03 panel. *ACM SIGKDD Explorations Newsletter* 5(2), p. 191. doi: 10.1145/980972.981004.

Frederiksen, L.S.F., Zhang, Y., Foged, C. and Thakur, A. 2020. The Long Road Toward COVID-19 Herd Immunity: Vaccine Platform Technologies and Mass Immunization Strategies. *Frontiers in Immunology* 11(1817). doi: 10.3389/fimmu.2020.01817.

Ganesan, K. 2014. What are Stop Words? Available at: https://kavita-ganesan.com/what-are-stop-words/#.YX-p8hrLdnI [Accessed: 1 November 2021].

Garcia, K. and Berton, L. 2021. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing* 101, p. 107057. doi: 10.1016/j.asoc.2020.107057.

Gautam, G. and Yadav, D. 2014. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. *2014 Seventh International Conference on Contemporary Computing (IC3)* . doi: 10.1109/ic3.2014.6897213.

Go, A., Bhayani, R. and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford* 1(12)

Go, A., Bhayani, R. and Huang, L. 2019. Sentiment140 - A Twitter Sentiment Analysis Tool. Available at: http://help.sentiment140.com/. [Accessed: 14 August 2021].

Goel, A., Gautam, J. and Kumar, S. 2016. Real time sentiment analysis of tweets using Naive Bayes. Available at: https://ieeexplore.ieee.org/abstract/document/7877424. [Accessed: 23 August 2021].

Gorbalenya, A.E. et al. 2020. The species Severe acute respiratory syndrome-related coronavirus : classifying 2019-nCoV and naming it SARS-CoV-2. *Nature Microbiology* 5(4), pp. 1–9. doi: 10.1038/s41564-020-0695-z.

Graham, M., Hale, S.A. and Gaffney, D. 2014. Where in the World Are You? Geolocation and Language Identification in Twitter. *The Professional Geographer* 66(4), pp. 568–578. doi: 10.1080/00330124.2014.907699.

Granik, M. and Mesyura, V. 2017. Fake news detection using naive Bayes classifier. *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)* . Available at: https://ieeexplore.ieee.org/abstract/document/8100379/. [Accessed: 9 October 2021].

Graupensperger, S., Abdallah, D.A. and Lee, C.M. 2021. Social norms and vaccine uptake: College students' COVID vaccination intentions, attitudes, and estimated peer norms and comparisons with influenza vaccine. *Vaccine* 39(15). doi: 10.1016/j.vaccine.2021.03.018.

Himelboim, I., Xiao, X., Lee, D.K.L., Wang, M.Y. and Borah, P. 2019. A Social Networks Approach to Understanding Vaccine Conversations on Twitter: Network Clusters, Sentiment, and Certainty in HPV Social Networks. *Health Communication* 35(5), pp. 607–615. doi: 10.1080/10410236.2019.1573446.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F. and Kaymak, U. 2013. Exploiting emoticons in sentiment analysis. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*. doi: 10.1145/2480362.2480498.

Huang, J., Lu, J. and Ling, C.X. 2003. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. *Third IEEE International Conference on Data Mining* . doi: 10.1109/icdm.2003.1250975.

Hussain, A. et al. 2020. Artificial intelligence-enabled analysis of UK and US public attitudes on Facebook and Twitter towards COVID-19 vaccinations. *Journal of Medical Internet Research* 23(1). doi: 10.1101/2020.12.08.20246231.

Islam, M.J., Wu, Q.M.J., Ahmadi, M. and Sid-Ahmed, M.A. 2007. Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers. Available at: https://ieeexplore.ieee.org/document/4420473 [Accessed: 30 June 2020].

Janes, A., Sillitti, A. and Succi, G. 2013. Effective Dashboard Design. *Cutter IT Journal* 26(1), pp. 17–24.

John Hopkins University 2021. COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. Available at: https://gisanddata.maps.arcgis.com/apps/dashboards/bda7594740fd40299423467b48e9ecf6 [Accessed: 15 October 2021].

Kim, H.-Y. 2017. Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test. *Restorative Dentistry & Endodontics* 42(2), p. 152. Available at: https://synapse.koreamed.org/DOIx.php?id=10.5395/rde.2017.42.2.152 [Accessed: 10 October 2021].

Klimiuk, K., Czoska, A., Biernacka, K. and Balwicki, Ł. 2021. Vaccine misinformation on social media – topic-based content and sentiment analysis of Polish vaccine-deniers' comments on Facebook. *Human Vaccines & Immunotherapeutics* 17(7), pp. 1–10. doi: 10.1080/21645515.2020.1850072.

Kralj Novak, P., Smailović, J., Sluban, B. and Mozetič, I. 2015. Sentiment of Emojis. Perc, M. ed. *PLOS ONE* 10(12), p. e0144296. doi: 10.1371/journal.pone.0144296.

Kristiyanti, D.A., Umam, A.H., Wahyudi, M., Amin, R. and Marlinda, L. 2018. Comparison of SVM Naïve Bayes Algorithm for Sentiment Analysis Toward West Java Governor Candidate Period 2018-2023 Based on Public Opinion on Twitter. Available at: https://ieeexplore.ieee.org/document/8674352.

Kwok, S.W.H., Vadde, S.K. and Wang, G. 2021. Twitter Speaks: An Analysis of Australian Twitter Users' Topics and Sentiments About COVID-19 Vaccination Using Machine Learning (Preprint). *Journal of Medical Internet Research* 23(5). doi: 10.2196/26953.

Lee, T.H. and Chen, A.H. 2021. Last-Mile Logistics of Covid Vaccination — The Role of Health Care Organizations. *New England Journal of Medicine* 384(8), pp. 685–687. doi: 10.1056/nejmp2100574.

Lin, C. and He, Y. 2009. Joint sentiment/topic model for sentiment analysis. In: *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. Available at: https://dl.acm.org/ft_gateway.cfm?id=1646003 [Accessed: 8 October 2021].

Love, B.C. 2002. Comparing supervised and unsupervised category learning. *Psychonomic Bulletin & Review* 9(4), pp. 829–835. doi: 10.3758/bf03196342.

Marcec, R. and Likic, R. 2021. Using Twitter for sentiment analysis towards AstraZeneca/Oxford, Pfizer/BioNTech and Moderna COVID-19 vaccines. *Postgraduate Medical Journal* , p. postgradmedj-2021-140685. doi: 10.1136/postgradmedj-2021-140685.

Matsumoto, S., Takamura, H. and Okumura, M. 2005. Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In: *Lecture Notes in Computer Science*. Springer

Mayo, M. 2014. A Clustering Analysis of Tweet Length and its Relation to Sentiment. *ArXiv* 1406(3287), pp. 1–6. Available at: https://arxiv.org/abs/1406.3287 [Accessed: 8 October 2021].

Medhat, W., Hassan, A. and Korashy, H. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5(4), pp. 1093–1113. Available at: https://www.sciencedirect.com/science/article/pii/S2090447914000550.

Mueller, A. 2020. WordCloud for Python documentation — wordcloud 1.8.1 documentation. Available at: https://amueller.github.io/word_cloud/ [Accessed: 1 November 2021].

Murray, E. 2019. The Importance Of Color In Data Visualizations. Available at: https://www.forbes.com/sites/evamurray/2019/03/22/the-importance-of-color-in-data-visualizations/?sh=71d88cd757ec [Accessed: 11 October 2021].

National Eye Institute 2019. Color Blindness | National Eye Institute. Available at: https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness [Accessed: 1 November 2021].

Neighmond, P. 2009. Is Vaccine Refusal Worth The Risk? *NPR* 26 May. Available at: https://www.npr.org/templates/story/story.php?storyId=104523437&t=1634302177754 [Accessed: 15 October 2021].

NHS 2021. Coronavirus (COVID-19) vaccination. Available at: https://www.nhs.uk/conditions/coronavirus-covid-19/coronavirus-vaccination/ [Accessed: 15 October 2021].

Oracle Corporation 2019. MySQL :: MySQL 8.0 Reference Manual. Available at: https://dev.mysql.com/doc/refman/8.0/en/ [Accessed: 8 October 2021].

Ordnance Survey Press Office 2014. Where Is The Centre Of Great Britain? | Blog. Available at: https://www.ordnancesurvey.co.uk/newsroom/blog/where-is-the-centre-of-great-britain-2 [Accessed: 9 August 2021].

Paltoglou, G. and Thelwall, M. 2012. Twitter, MySpace, Digg. *ACM Transactions on Intelligent Systems and Technology* 3(4), pp. 1–19. doi: 10.1145/2337542.2337551.

Pandolfi, F. et al. 2018. The Importance of Complying with Vaccination Protocols in Developed Countries: “Anti-Vax” Hysteria and the Spread of Severe Preventable Diseases. *Current medicinal chemistry* 25(42), pp. 6070–6081. Available at: https://www.ncbi.nlm.nih.gov/pubmed/29773050.

Pedregosa, F., Buitinck, L., Louppe, G., Grisel, O., Varoquaux, G. and Mueller, A. 2015. Scikit-learn. *GetMobile: Mobile Computing and Communications* 19(1), pp. 29–33. doi: 10.1145/2786984.2786995.

Piedrahita-Valdés, H. et al. 2021. Vaccine Hesitancy on Social Media: Sentiment Analysis from June 2011 to April 2019. *Vaccines* 9(1), p. 28. doi: 10.3390/vaccines9010028.

Pratama, B.Y. and Sarno, R. 2015. Personality classification based on Twitter text using Naive Bayes, KNN and SVM. *2015 International Conference on Data and Software Engineering (ICoDSE)* . doi: 10.1109/icodse.2015.7436992.

Priddy, S. 2021. The coronavirus timeline: Measures taken by the House of Commons. Available at: https://commonslibrary.parliament.uk/house-of-commons-coronavirus-timeline/ [Accessed: 10 September 2021].

Pristiyono, Ritonga, M., Ihsan, M.A.A., Anjar, A. and Rambe, F.H. 2021. Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes Algorithm. *IOP Conference Series: Materials Science and Engineering* 1088(1), p. 012045. doi: 10.1088/1757-899x/1088/1/012045.

Reitz, K. 2021. requests: Python HTTP for Humans. Available at: https://pypi.org/project/requests/. [Accessed: 29 July 2021].

Ronacher, A. 2021. Welcome to Flask — Flask Documentation (2.0.x). Available at: https://flask.palletsprojects.com/en/2.0.x/ [Accessed: 2 September 2021].

Saif, H., Fernández, M., He, Y. and Alani, H. 2014. On stopwords, filtering and data sparsity for sentiment analysis of Twitter. Available at: http://oro.open.ac.uk/40666/ [Accessed: 8 October 2021].

Sathya, R. and Abraham, A. 2013. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence* 2(2). doi: 10.14569/ijarai.2013.020206.

Sayce, D. 2010. Number of tweets per day? | David Sayce. Available at: https://www.dsayce.com/social-media/tweets-day/ [Accessed: 14 October 2021].

Schlosser, S., Toninelli, D. and Cameletti, M. 2021. Comparing Methods to Collect and Geolocate Tweets in Great Britain. *Journal of Open Innovation: Technology, Market, and Complexity* 7(1), p. 44. doi: 10.3390/joitmc7010044.

Schouten, K., van der Weijde, O., Frasincar, F. and Dekker, R. 2018. Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data. *IEEE Transactions on Cybernetics* 48(4), pp. 1263–1275. doi: 10.1109/tcyb.2017.2688801.

Siebert, C., Hartmann, J., Heitmann, M. and Schamp, C. 2019. Accuracy of Automated Sentiment Analysis. *SSRN Electronic Journal* . doi: 10.2139/ssrn.3489963.

Sigdel, R. 2020. Improve Your Visualization Skills Using Tufte's Principles of Graphical Design. Available at: https://medium.com/nightingale/improve-your-visualization-skills-using-tuftes-principles-of-graphical-design-3a0f40a53a2c [Accessed: 22 March 2021].

Singh, G., Kumar, B., Gaur, L. and Tyagi, A. 2019. Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. Available at: https://ieeexplore.ieee.org/document/8776800/ [Accessed: 10 August 2020].

Stabe, M. 2016. Why the FT creates so few clickable graphics. Available at: https://www.ft.com/content/c62b21c6-7feb-11e6-8e50-8ec15fb462f4 [Accessed: 16 March 2021].

Stein, R.A. 2017. The golden age of anti-vaccine conspiracies. *Germs* 7(4), pp. 168–170. doi: 10.18683/germs.2017.1122.

Stieglitz, S. and Dang-Xuan, L. 2012. Political Communication and Influence through Microblogging--An Empirical Analysis of Sentiment in Twitter Messages and Retweet Behavior. *2012 45th Hawaii International Conference on System Sciences* . doi: 10.1109/hicss.2012.476.

Symeonidis, S., Effrosynidis, D. and Arampatzis, A. 2018. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications* 110, pp. 298–310. doi: 10.1016/j.eswa.2018.06.022.

Tibbetts, J.H. 2020. Will a COVID Vaccine Be Accepted? *BioScience* 71(1). doi: 10.1093/biosci/biaa133.

Tomana, M., Tesara, R. and Jezeka, K. 2007. Influence of Word Normalization on Text Classification. Available at: https://www.semanticscholar.org/paper/Influence-of-Word-Normalization-on-Text-Tomana-Tesara/49280008b5a46a6b45f01b58afa8777d91e7ebe2#citing-papers [Accessed: 8 October 2021].

Tomeny, T.S., Vargo, C.J. and El-Toukhy, S. 2017. Geographic and demographic correlates of autism-related anti-vaccine beliefs on Twitter, 2009-15. *Social Science & Medicine* 191, pp. 168–175. Available at: https://www.sciencedirect.com/science/article/pii/S0277953617305221. [Accessed: 22 October 2021].

Twitter 2020. Twitter Terms of Service. Available at: https://twitter.com/en/tos [Accessed: 1 November 2021].

Twitter 2021a. Twitter API Documentation. Available at: https://developer.twitter.com/en/docs/twitter-api [Accessed: 7 August 2021].

Twitter 2021b. Twitter API for Academic Research | Products. Available at: https://developer.twitter.com/en/products/twitter-api/academic-research [Accessed: 10 August 2021].

UK Health Security Agency. 2021. Coronavirus (COVID-19) cases in the UK. Available at: https://coronavirus.data.gov.uk/ [Accessed: 15 October 2021].

Uysal, A.K. and Gunal, S. 2014. The impact of preprocessing on text classification. *Information Processing & Management* 50(1), pp. 104–112. Available at: https://www.sciencedirect.com/science/article/pii/S0306457313000964 [Accessed: 6 July 2019].

van Rossum, G. and Drake, F.L. 2009. *Python 3 : reference manual*. United States: Sohobooks.

Virtanen, P. et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17(3), pp. 261–272. doi: 10.1038/s41592-019-0686-2.

W3Schools 2019. CSS Tutorial. Available at: https://www.w3schools.com/css/default.asp. [Accessed 20 September 2021].

World Health Organization 2020a. Pneumonia of unknown cause – China. Available at: https://www.who.int/emergencies/disease-outbreak-news/item/2020-DON229. [Accessed 10 October 2021].

World Health Organization 2020b. Managing the COVID-19 infodemic: Promoting healthy behaviours and mitigating the harm from misinformation and disinformation. Available at: https://www.who.int/news/item/23-09-2020-managing-the-covid-19-infodemic-promoting-healthy-behaviours-and-mitigating-the-harm-from-misinformation-and-disinformation [Accessed: 10 October 2021].

Xu, Z. and Guo, H. 2017. Using Text Mining to Compare Online Pro- and Anti-Vaccine Headlines: Word Usage, Sentiments, and Online Popularity. *Communication Studies* 69(1), pp. 103–122. doi: 10.1080/10510974.2017.1414068.

Zacharias, C. 2018. twintproject/twint. Available at: https://github.com/twintproject/twint [Accessed: 20 July 2021].

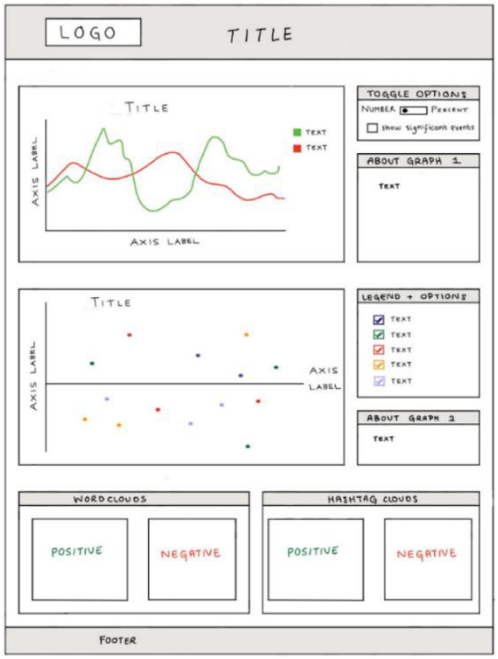
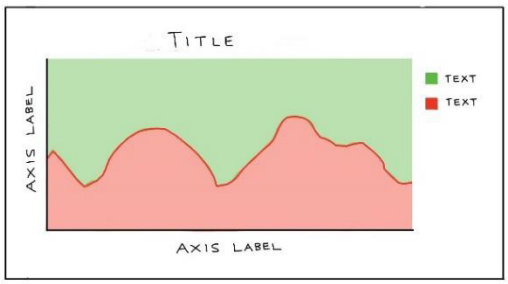
Zhou, L., Pan, S., Wang, J. and Vasilakos, A.V. 2017. Machine learning on big data: Opportunities and challenges. *Neurocomputing* 237, pp. 350–361. doi: 10.1016/j.neucom.2017.01.026.

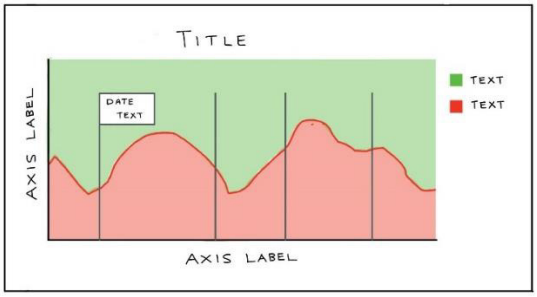
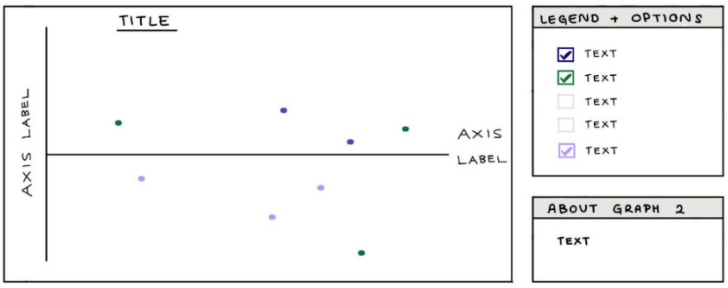
13 APPENDICES

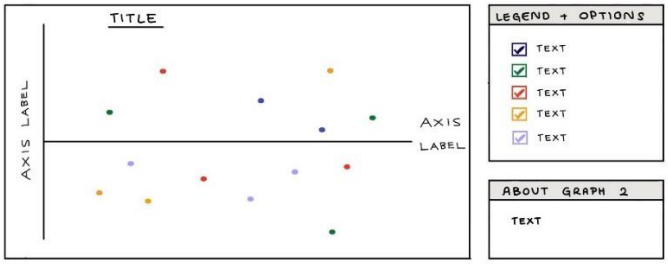
APPENDIX 1 : DETAILS OF KEY EVENTS

Date	Category	Details
23/03/2020	Lockdown	Lockdown 1 begins
01/06/2020	Lockdown	Schools reopen after lockdown 1
31/10/2020	Lockdown	Lockdown 2 announced
23/11/2020	Lockdown Lockdown	Lockdown 2 ends in favour of tier system Christmas bubble plans announced
02/12/2020	Specific vaccines NHS strategy	UK approves Pfizer vaccine NHS national vaccination plan announced
08/12/2020	NHS strategy	Vaccine rollout for care home residents and key workers
19/12/2020	Lockdown	Christmas U-turn: announcement of stricter lockdown over holidays
04/01/2021	Specific vaccines	UK begins use of AstraZeneca vaccine
06/01/2021	Lockdown	Lockdown 3 begins
20/02/2021	NHS strategy Lockdown	Updates given on NHS vaccination strategy Announcement of plans to leave lockdown 3
01/03/2021	NHS strategy	Vaccination rollout for all aged 60+
07/04/2021	Specific vaccines Health concerns	UK begins use of Moderna vaccine First reports of AstraZeneca blood clots released
14/04/2021	Health concerns	European nations pause use of AstraZeneca vaccine
30/04/2021	NHS strategy	Vaccination rollout for all aged 40+
07/05/2021	Health concerns Specific vaccines NHS strategy	UK announces AstraZeneca will not be administered to anyone ages 40+ due to health concerns
28/05/2021	Specific vaccines	UK approves use of Johnson & Johnson vaccine
04/06/2021	Specific vaccines	UK approves use of Pfizer vaccine for children aged 12-15
18/06/2021	NHS strategy	Vaccination rollout for all aged 18+
19/07/2021	Lockdown	Lockdown 3 ends (Freedom Day)

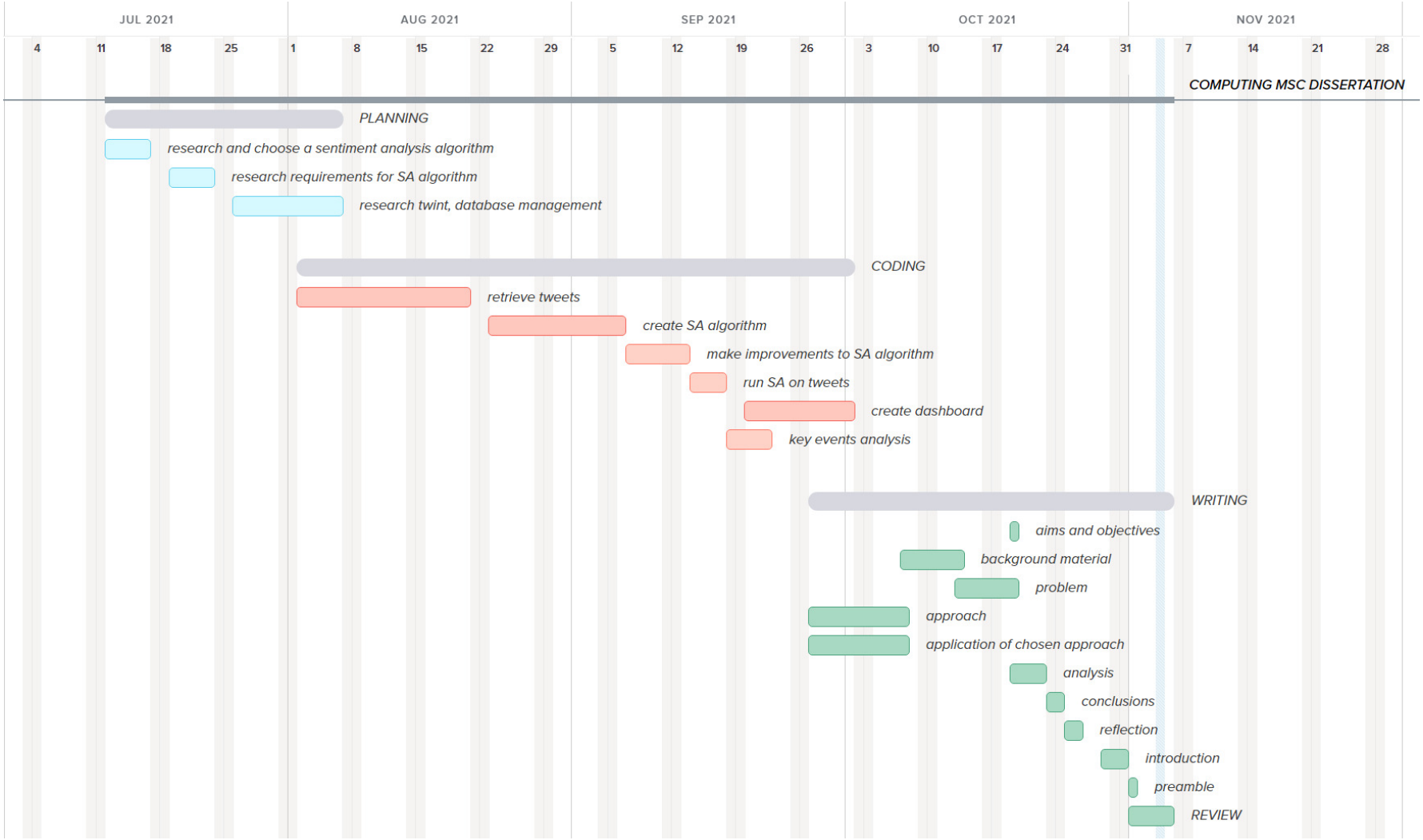
APPENDIX 2 : DASHBOARD TEST CASE

Test Case Id: 1	**Test Purpose:** To test the interactive visualisations on the CoVacSenti dashboard		
Environment: A local host server, using Google Chrome (v. 90.0.4430.93) on Windows 10 v.1909; OS build 18363.1500			
Preconditions: The user in on the CoVacSenti dashboard			
Test Case Steps:			
Step No	**Procedure**	**Expected Response**	**Pass/Fail**
1	Scroll through the whole page	All visualisations are fully loaded with the following layout:  The correct default values are present: - The **switch** in the **Toggle Options** container is set to **Number** - The **Significant Events** checkbox is unselected - Every checkbox in the **Legend Options** is selected	Pass (see notes)
2	Click the **switch** in **Toggle Options**	The switch slider moves to **Percent** (on the right) and does not move back. The visualisation changes to a stacked area chart with the following layout :  	Pass

		The y-axis label changes from **Number** to **Percent** and the y-axis scale changes so the maximum value is 100. The x-axis and legend remain unchanged.	
3	Click the **Significant Events** checkbox	The checkbox is filled in, signalling it is selected, and does not change when the cursor leaves the checkbox. Vertical markers ascend from the x axis on the first visualisation. No other changes occur on the dashboard.	Pass
4	Hover on one of the markers	A tooltip appears containing a date corresponding to the x-axis position of the marker, and a description of the event with the following layout:  The tooltip disappears when the cursor leaves the tooltip area.	Pass
5	Click the **Significant Events** checkbox again	The checkbox empties and the vertical markers are removed.	Pass
6	Click the **switch** in **Toggle Options**	The toggle slider returns to the **Number** setting. The plot and y-axis return to their original states from step 1.	Pass
7	Click any two of the **Legend Options** checkboxes	The checkboxes lose their colour when clicked, signalling they have been unchecked, and remain unchecked after the cursor leaves. All points that had the colours of the clicked checkboxes are removed from the scatter plot:  	Pass
			Pass

8	Click the same checkboxes as in step 7	The checkboxes regain their original colours, and the points that were removed in step 7 return to the scatter plot. The points retain their same position and colouring as in step 1. The visualisation has the following layout: 	
Comments: The dashboard has a slightly different layout; word clouds are ordered horizontally instead of vertically. All the same visualisations are present.			
Related Tests: None			
Author: c21014029	**Checker:** c21014029, tested 20/10/2021		

APPENDIX 3 : RESULTS OF FISHER'S EXACT TEST FOR ALL KEY DATES

Date	P-value (number of tweets)	P-value (percent of tweets)
2020-03-23	**0.0188**	**2.14×10^{-13}**
2020-06-01	0.0667	**2.04×10^{-7}**
2020-10-31	0.451	**0.0214**
2020-11-23	0.0853	0.0877
2020-12-02	1.00	1.00
2020-12-08	0.598	0.667
2020-12-19	0.803	0.887
2021-01-04	0.123	0.314
2021-01-06	**8.65×10^{-8}**	**0.00160**
2021-02-20	**0.006378**	**0.0150**
2021-03-01	0.597	0.669
2021-04-07	0.709	0.883
2021-04-14	0.891	0.886
2021-04-30	**0.0421**	**0.0320**
2021-05-07	0.0784	0.0633
2021-05-28	1.00	1.00
2021-06-04	0.141	0.154
2021-07-19	0.518	0.632

APPENDIX 4 : GANTT CHART OF DISSERTATION PROGRESS