National University of Ireland Galway

CT413 Final Year Project BSc (CSIT)

Analysis of Political Tweets with respect to Topic, Sentiment, and Stance

[redacted]

[redacted]

19 April 2021

Project Demo Video: https://tinyurl.com/aideenmc-fyp-demo
Github : https://github.com/a-mcloughlin/final-year-project
Project URL : http://web1.cs.nuigalway.ie:8081/

Supervised by

[redacted]
School of Computer Science
College of Science and Engineering

Acknowledgements

I would like to thank my Final Year Project advisor, Dr Josephine Griffith for all her valuable support and guidance throughout the development of this project. The resources, advice and encouragement which she provided to me over the 8 months of this project were integral to the process of planning, researching, implementing and testing this application. Our regular online meetings helped me to stay on schedule and maintain a consistent plan despite the challenges of working on a final year project remotely during COVID-19 lockdowns.

I would also like to acknowledge the lecturers and support staff of the school of Computer Science and Information Technology of the National University of Ireland, Galway. The skills which I developed in my first three years as a computer science student, through the modules I took and assignments I completed, gave me the technical ability and project management experience that was essential in completing this project.

I would like to thank my colleagues from Workday Inc. Dublin, with whom I completed my Software Engineering Internship, for teaching me many of the CICD skills and high coding standards which I was able to apply to this project. The knowledge of code legibility, logical comments, clear commit messages and the importance of testing enabled me to execute this project to an industry standard, allowing me to further practice these skills and prevent any major bugs or losses of code.

Finally, I would like to thank my family and friends for supporting me throughout this development process. The support, advice and patience they provided over these 8 months were imperative to the development process, as completing a project while in lockdown was challenging in many more ways than I had expected. Thank you for celebrating the successes and commiserating the challenges of this project with me.

Declaration

I hereby declare that this final year project, which I now submit for correction, is entirely my own work and has not been influenced by the work of others except where cited.

[redacted]

19 April 2021

Abstract

The aim of this project was to develop an application which can analyse sets of tweets with respect to sentiment, emotions expressed, political leaning and other interesting linguistic features.

This report describes the processes through which this application was researched, implemented, developed and evaluated.

The significant background research which was conducted before the project development commenced is shown, along with the evolution of the various project components over the project development cycle.

A thorough exploration is provided of the linguistic complications associated with analysing and classifying Twitter data, specifically political rhetoric, on Twitter. The focus of this project was on NLP (Natural Language Processing) techniques, so this project explores those topics in depth.

The developed application combines many different language processing tools and techniques to provide interesting insights into Twitter use, with a specific focus on its use in politics. A user interface was developed for the application to convey this data in a clear and visually interesting way.

The developed application received positive user evaluations,and succeeded in analysing, to a significant level of accuracy, the sentiment, emotions expressed and political leaning of a set of tweets.

Contents

Figures

1 Introduction

1.1 Project Overview

The initial project scope was outlined by my Final Year Project supervisor Dr. Josephine Griffith.
The project definition was as follows:
"*Analysis of Political Tweets with respect to topic, sentiment, and stance. Political tweets offer a rich source of data for observers to mine where tweets can be analysed to detect topics and the drift, or change, in these topics over time; to detect the political stance of tweets (or accounts), and to detect the sentiment of tweets and/or accounts. Many past studies in this area have attempted to predict election outcomes based on features and sentiment of political tweets. The presence and influence of fake accounts offers additional challenges when working in this domain. This project will focus on applying suitable NLP (Natural Language Processing) techniques to a dataset of political tweets to analyse characteristics such as topic, sentiment, and stance, and to determine how these characteristics vary across human and bot accounts.*"

This scope expanded and evolved over the course of the project development. The decision was made to develop a tool, along with a user interface, which could analyse the sentiment, emotions and political leaning expressed within a set of tweets, along with other interesting aspects of the language used in these tweets. The 'topic' of the tweets was not expanded as the 'topic' is given by the user of the application when they enter the hashtag or user account to analyse. The core aspects of this project focused on exploring and developing tools for natural language processing to analyse Twitter data.

1.2 Project Objectives

This application will be developed in Python and will have a simple Python Flask front-end. It will allow users to analyse the tweets within a specific hashtag, or by a specific user. The main data that will be analysed from the set of tweets will be their sentiment, the most common words and phrases used and their implied political leaning, left to right. This web app will also allow users to compare the activity of two separate accounts or hashtags to see similarities and differences.

The application can be broken down into a set of objectives. These objectives are classified either as core objectives, which are required as a central aspect of the project, and extra objectives, which would be nice, but not necessary additions to the project.

1.2.1 Core Objectives

Analyse Twitter Data within a hashtag
The goal is to build a system which takes a hashtag as input, and fetches the most recent set of tweets using that hashtag. This set of tweets will then be analysed in a number of ways.
Some data which can be extracted from a set of tweets would be:

- The number of unique words
- The most frequently used words
- The most common emojis used

Analyse Twitter data for a user account
The system which handles hashtags can be extended to analyse data from a specific Twitter account. The tweets from the user can be analysed to extract the same data as stated above. Additional account data can also be analysed to show information about the Twitter account including:

- Number of account followers
- Number of accounts followed by the account in question
- Account location (If available)
- 'Pinned' tweet (If available)

Estimate sentiment of set of tweets
Once a set of tweets has been fetched, whether through a hashtag or a user account, these tweets can be analysed to determine the overall sentiment on a scale of positive to neutral to negative. The ability to see how positive, or negative, different discussions, and accounts, are on Twitter can show how differently certain groups see issues, and why people may be drawn to groups that express flawed but highly optimistic views.

Estimate emotions present in a set of tweets
Analysing the emotions expressed in a set of tweets can allow for very interesting conclusions to be drawn from the data. Fear is a common tool used by politicians to stoke support for policies that purport to provide a solution to the issue being 'feared'. Anger is another emotion which is often used as a political tool, and is arguably one of the most prevalent emotions expressed on the internet. The ability to analyse and compare the use of different emotions across different Twitter accounts and conversations can be very interesting and therefore will be implemented in this project.

Estimate Political Leaning of a set of tweets
A set of tweets can be analysed to infer their political leaning.
Analysing political leaning can provide significant insights into the differences between the ways different political viewpoints are expressed on Twitter, and can highlight the use of political language in a seemingly non-political discussion on Twitter. It is worth noting that any classifications or 'political predictions' made by an algorithm within this project are not to be taken as fact, as any political classification algorithm can be, and will be, inaccurate in certain scenarios. Scenarios that may lead to inaccurate results could occur due to a very limited set of tweets found for a query (i.e. fewer than 20 tweets), or the use of sarcasm about, commentary on, or mockery of, the opposing political viewpoints.

Develop a simple and effective user interface for the web application
A simple and effective user interface is integral to allow users to easily use the system, and to easily understand the results that the system provides.
The data should be displayed in a way that avoids overwhelming the user while allowing them to access more complex data if they wish.
A significant importance will be placed on clear and concise data visualization within the user interface. There will be a lot of data returned from any single query, and it is important to display this data in a clear and understandable way.

1.2.2 Extra Objectives

The extra objectives of this project are not necessary core components. These features are nice additional features which were implemented after the core objectives as tasks which were not in the initial project plan. These features are based on new ideas discovered over the course of the project development, or suggestions from the user feedback process.

Estimate Probability of Twitter account being fake

Fake Twitter accounts are a prevalent problem within the political sphere of many countries. Online radical communities have been known to create fake accounts masquerading as members of minority groups, or people of different nationalities, to manipulate political discussions and political sentiments for specific issues. Since there is no way to know for sure whether any Twitter account is a fake or not, this probability, generated from an established system or my own system, will only be an estimation, however it can still provide some useful information.

Allow web-app results to be easily shared to Twitter or other platforms

A nice feature to implement into the web-application would be an easily integrated way to share the analysis returned by the webapp on Twitter.
This feature is widely supported by Twitter and there is significant online support for implementing a 'share on Twitter' button.

1.2.3 Objectives not Implemented

Some initially planned objectives could not be implemented in the final version of the project due to the constraints of the tools and technology used.

Estimate Political Polarisation within a Hashtag

Many discussions and hashtags on Twitter are very political in nature. However lots of these discussions happen within a political group who share the same ideas. This can lead to people overestimating the support for their ideas, and underestimating the strength of the opposition to those views.

Once the political leaning of a set of tweets can be found, the plan was to extend the system to analyse the range and strength of political leaning within a hashtag, and thus the polarization of the hashtag.
This feature was not implemented due to the time constraints of political leaning detection and the Twitter API rate limit . Detecting the political leaning of separate accounts within a popular hashtag would be a very time intensive procedure and would require separate Twitter API calls for each user account. This would potentially exhaust the Twitter API rate limit of 500,000 queries a month before the month's end, making further project development in that month much more difficult.

Estimate Ratio of fake accounts in a hashtag

This idea of this feature was to analyse the prevalence of inauthentic accounts in various Twitter discussions. However it could not be implemented due to the Bot detection rate limits. The implemented bot detection tool, Botometer, allows 500 API calls per user, per day. If all accounts in a hashtag were analysed with Botometer then this rate limit could be exhausted after just 2 queries. This made this feature impractical to implement in the project.

1.3 Report Overview

This report provides a thorough explanation of the development process of this application, which was developed as a Final Year Project over the 8 month period from October 2020 to May 2021.

The Introduction provides an overview of the objectives, timelines, risks and constraints for this project.

The Background Research section explores the research which was conducted in preparation for, and over the course of, the project development. Previous work in the field was explored, and thorough research was conducted into Sentiment Analysis, Emotion Analysis, Political Leaning classification and Twitter 'bot' detection.

The Implementation section describes how the planned work was developed, and shows how the project components evolved over the course of the development. This section shows, in detail, the process of developing suitable sentiment analysis and emotion detection systems. The process of developing the political leaning analysis system is also explored in depth, with explanations of the different approaches taken, and difficulties encountered in handling this complex language classification task. The development and evolution of the graphical user interface is also shown in this section.

The Development Processes section details the CICD (Continuous Integration, Continuous Development) processes which were implemented and rules which were followed as the project was developed to ensure consistently high code quality. These processes consisted of a reliable version control system, complete with thorough and well maintained unit tests for all code written.

The Application Evaluation section details how the project was evaluated once all objectives had been completed. Testing was performed on all major project components to ensure that they were performing as expected, with a more in depth testing performed on the Political Leaning analysis system due to the complexity of this component. The user evaluation process is also described in this section,with the results of this evaluation and the conclusions drawn from it started clearly.

The Future Work and Improvements section explores ways upon which the scope of this project could be expanded in further work, and the Conclusion contains some final insights into the development of this project.

1.4 Project Deadlines

1.4.1 Project Tasks

This task planning table was put together in November 2020 in the initial planning stage of the project, to divide the project work into individual, time estimated tasks. The planned completion dates were based on 4 weekly hours of project work in semester 1 and 16 weekly hours of project work in semester 2. This table was used and updated constantly throughout the project to monitor the project progress, with the "actual completion dates" column being updated once a task had been completed.

Figure 1: Project Task Planning Table

Task ID	Tasks	Estimated Time	Dependencies	Planned Completion Date	Actual Completion Date
t0	Receive FYP allocation	-	-	-	13/10/2020
t1	Research Twitter Developer API	4h	-	20/10/2020	20/10/2020
t2	Investigate previous research in this area	4h	-	27/10/2020	21/10/2020
t3	Working command line system - analyse hashtag and get most used words	8h	t1	10/11/2020	22/10/2020
t4	Working system - analyse user or hashtag, get sentiment (Azure)	8h	t3	24/11/2020	29/10/2020
t5	Create Flask application to run system	4h	t4	1/12/2020	05/11/2020
t6	Setup Flask application to work on college linux server	4h	t5	8/12/2020	12/11/2020
t7	Apply weighting to sentiment analysis	4h	t4	15/12/2020	12/11/2020
t8	Create unit tests and setup github repo to properly use them	8h	t3	05/01/2021	19/11/2020
Break from FYP work from 1/12/2020 to 25/1/2021 (Semester 1 exams)					
t9	Estimate political leaning of tweets	32h	t4	05/01/2021	08/02/2020
t10	Design and Implement GUI for project	16h	t6	19/01/2021	15/02/2020
t11	Research, rescope and re-estimate	16h	-	02/02/2021	22/02/2020
t12	Analyse user account data - location	8h	t11	16/02/2021	25/02/2020
t13	implement account authenticity estimator	16h	t11	02/03/2021	11/03/2020
t14	implement political polarisation estimator	16h	t11	16/03/2021	Not implemented
t15	allow web-app data to be tweeted	4h	t11	23/03/2021	15/03/2021
t16	Conduct User Evaluation	8h	all	26/03/2021	29/03/2021
t17	Updates in response to user Evaluation	16h	t16	02/04/2021	05/04/2021
t18	Write Final Year Project Report	32h	all	20/04/2021	19/04/2021
t19	Compose FYP demo video, Prepare for viva voce	32h	all	27/04/2021	30/04/2021
Finished FYP work early as scheduled to study for Semester 2 exams					
	Initial Report Due Date		06/05/2021	Updated	07/06/2021
	Initial Project Demonstration and Viva Voce Dates		10/05/2021	Updated	09/06/2021

1.4.2 Risk Assessment

A risk assessment was conducted in the planning stage of the project to determine the biggest risks to the project completion. Most of the risks investigated could be mitigated, or avoided, by staying ahead of schedule and maintaining a consistent, functional version of the project in a Github repository.

Figure 2: Risk Assessment Table

Risk	Consequence	Impact	Risk response strategy
Hardware Issues with laptop	Unable to work on project, unable to meet deadlines	High	Stay ahead of schedule
Loss of already completed work	Work needs to be redone from scratch	High	Maintain versioning System (Github)
Unable to work on Final Year Project due to demands of other modules	Unable to meet deadlines	Medium	Start work early, and stay on top of deadlines for all modules
Previously working feature is now broken	Work needs to be fixed or redone	Medium	Implement consistent unit tests on github repository
Limitations of API make it too difficult to gather large dataset	Results of data analysis are less interesting and less useful	Medium	Gather and store datasets to better train and understand data
Rate limit of 500,000 tweets per month from Twitter developer API used up	Unable to analyse any new or current data	High	Store dataset of 10,000 tweets from various hashtags/accounts to allow for offline development
Insufficient time to complete goals	Lower quality project	High	Avoid by staying on schedule, if it happens, handle by re-planning and re-estimating

1.4.3 Constraints

The most significant constraints encountered in the development of this project were as follows:

- Tool limitations, such as Twitter and Bot Detection API limits.
- Language classification complexity. The significant variations in the way language is used by different groups.
- Time constraints, due to demands of other modules.

1.4.4 Deliverables

- Project Definition Document - 28th November 2020
- Final Project Report - 6th May 2021, updated to 7th June 2021
- Project Demonstration and Viva Voce- 10th-14th May 2021, updated to 9th-11th June 2021
  Viva Voce completed early on 10th May 2021 to meet initial schedule

2 Background Research

2.1 Academic Papers

Before beginning this Project, I wanted to take a look at what research had been done in this field in the past.

The most relevant research to my project was done by S. Stieglitz and L. Dang-Xuan of the University of Duisburg-Essen, who published a research paper entitled “Political Communication and Influence through Microblogging - An Empirical Analysis of Sentiment in Twitter Messages and Retweet Behavior”. [1]
In this research paper, written for the 45th Hawaii International Conference on System Sciences in 2012, Stieglitz and Dang-Xuan analysed a dataset political tweets with respect to how often tweets were ‘retweeted’, and the correlation between tweet sentiment and retweet rate. This study examined a total of 64,431 tweets from the week of March 21 to 27, 2011 prior to state parliamentary elections in Germany, and found a strong correlation between emotive language use and retweet rate, with strongly expressed sentiments, whether positive or negative, making a tweet much more likely to be spread widely on Twitter. This paper was very interesting to read and provided much insight into how certain tweets and ideas can spread quickly across the platform of Twitter.

Other interesting articles I read in researching the area of Twitter analysis were “Finding interesting posts on Twitter based on retweet graph analysis”, and “Style Matters!: Investigating Linguistic Style in Online Communities”.
“Finding interesting posts on Twitter based on retweet graph analysis” by M.-C. Yang, J.-T. Lee, S.-W. Lee, and H.-C. Rim, published in Proceedings of SIGIR in 2012 [2] focused on the process of identifying social media posts which are likely to be of interest to the user, through using 64,107,169 tweets by 2,824,365 Twitter users to analyse the likelihood of a tweet being interesting based on the degree to which the tweet is spread beyond the normal community of that Twitter user. This paper found that ‘interesting’ tweets cannot be found purely through popularity measure by retweets and that taking the online community in which the Twitter user normally inreacts into account can provide much deeper insights into the likely interest level of a tweet.
“Style Matters!: Investigating Linguistic Style in Online Communities”, written by Padmini Srinivasan and Osama Khalid and published in ICWSM in 2020 [3], investigates how linguistic style can vary across different online communities using different online platforms. This report analysed data from communities discussing politics, travel and television from the three social networks of Reddit, Voat and 4chan. The study found that each of these communities developed their own unique linguistic styles, with these styles varying across both social networks and discussion groups. This highlighted some challenges with interpreting all communities on Twitter on the same scales and provided valuable insight into the vast linguistic variation of online dialect.

2.2 Sentiment Analysis

One core aspect of Twitter discourse to analyse was the sentiment of tweets, ie, how positive or negative the tweets were. This can be approached in a number of ways:

Microsoft Azure Text Analytics Service [4]

- This service can easily analyse a set of tweets and return a sentiment score indicating how positive or negative the tweets are.
- The limitations of this system are in the rate limits of the system. With a student account, only 5000 requests per month are provided free.

The NRC Emotion Lexicon [5]

- This dataset requires more complex implementation, however it is open source and provides emotion mapping for over 14,000 english words. Words are mapped as expressing, or not expressing, a range of different emotions, along with positive and negative sentiments.
- This system would allow for easy integration with the emotion analysis, however since the dataset is primarily built for emotion classification and not sentiment classification, it was not the most accurate system when implemented in this project.

VADER SentimentIntensityAnalyzer with NLTK [6]

- VADER, the Valence Aware Dictionary and Sentiment Reasoner, is a pre-trained sentiment lexicon which is tuned to detecting sentiments expressed online using social media. This makes it an ideal candidate for analysing Twitter text data. It is available in the Python NLTK (Natural Language Toolkit) package for easy integration into Python projects. This tool returns a positive, neutral and negative score for a given sentence, indicating how strongly the sentiment applies to the sentence. Thus, each tweet can be classified as either positive, neutral or negative based on the scores it is allocated. This can provide a clear breakdown of the frequency with which each sentiment classification applies to the tweets in a given hashtag or by a given user.

Machine Learning Approaches

- Another option for analysing the sentiment of a set of tweets would be through Machine Learning, by training a machine learning algorithm with a data set of tweets matched to sentiments. That algorithm could then be used to estimate the sentiment of a set of tweets with unknown sentiment.
- The limitations of this scenario are that a suitable dataset with relevant sentiment labels must be found to correctly analyse the sentiment of Twitter data.

After analysing the performance and limitations of the various sentiment analysis tools, the decision was made to use VADER SentimentIntensityAnalyzer with NLTK as the sentiment detection tool in the final version of this project as it proved to be the most straightforward and accurate way to determine the sentiment of a set of tweets.

2.3 Emotion Analysis

Emotion analysis of Twitter data can provide some very interesting data about how different topics are discussed, and how different public figures communicate on Twitter. Different options for how to draw these emotion conclusions from Twitter data were explored.

The NRC Emotion Lexicon [5]

- The NRC Emotion Lexicon was the obvious choice for determining emotions from sets of text data. This dataset provides emotion mapping for over 14,000 english words, stating whether or not each word expresses each of the configured emotions. These detectable emotions are anger, anticipation, disgust, fear, joy, sadness, surprise, trust, negative and positive. The negative and positive sentiments were excluded from this analysis as the intent was to analyse individual emotions.
- This system could be easily developed to determine the emotions of tweets, however it offers a simplistic approach, and would not be able to understand nuance, sarcasm or other linguistic techniques used to express emotion in more complex ways.

Machine Learning Approaches

- As in sentiment analysis, machine learning could be implemented to determine the emotions expressed in a set of tweets.
- There are significant challenges associated with using this approach however, as a very specific dataset of sentences mapped to emotions would be required to train the machine learning algorithm. A lot of data would likely be required to build an accurate algorithm and thus a suitable dataset is not likely to be available on the internet, and would be too complex and time intensive to build specifically for this project.

Due to the challenges of the Machine Learning approach, and the ease of implementation and relative accuracy of the NRC Emotion Lexicon, the decision was made to use the NRC Emotion Lexicon for emotion classification for this project.

2.4 Political Leaning Classification

Classifying political leaning from Twitter text data was one of the most complex aspects of this project, as political leaning is subjective and varies across countries and even communities. Strongly left and strongly right political speech often have more in common than either would have with centrist political speech. Political leaning classification will never be 100% accurate, however a simple version of a political leaning classification tool can still be developed to provide a general guess at the political leaning of a set of tweets. One thing that was important to keep in mind when examining political leaning was that political leaning is considered private information under GDPR. This should not present concerns in the development of this project, however, as the political leaning detection will not be developed as a 100% accurate tool to definitively state the political leaning of a person, only to predict political leaning of a set of public tweets.

Two possible ways to implement this political leaning detection are as follows:

Assigning Political values to individual words

- Many words can be assigned an inherent political leaning when used in political speech. The Political Sentiment Lexicon [7] was constructed in 2017, mapping roughly 1000 words to integer values from 4 for strongly liberal and -4 for strongly conservative.
- This dataset is easy and quick to use as a base classification tool, however it has significant drawbacks.
    - The dataset is very strongly US based, and thus the words it interprets are interpreted from a purely US standpoint. The American political spectrum of Democrats and Republicans cannot be used to accurately classify Irish, UK or other international political speech.
    - The dataset only maps 1000 words to political leaning. This is a small dataset to analyse and could lead to the use of very few words within a set of Twitter data having a disproportionate effect on the political leaning classification of that data. Political leaning being determined solely based on the use of 1000 words is likely to be inaccurate and unable to interpret the use of commentary or sarcasm.

Machine Learning trained with politically classified text

- Since political leaning is such a complex classification problem, it is a very good candidate for the application of machine learning. Machine Learning can analyse text data in a more complex and intuitive way, and identify patterns that could not easily be identified by human analysis alone. The most complex problem to tackle if using machine learning is the training dataset selection. This problem is explored in more depth in Section 3.4: Political Leaning Detection on Tweet Set.
- Another aspect of machine learning application to consider is what machine learning package to use. The most popular Python machine learning packages available online are Tensorflow [8] and Scikit Learn [9].
    - Tensorflow is a very well supported Python machine learning tool which is targeted at use in Deep Learning. Deep Learning can be described as machine learning with extremely large datasets using Neural network models. Tensorflow requires slightly more configuration to work on a basic machine learning problem than Scikit Learn.
    - Scikit Learn is a very user friendly and highly customizable general purpose machine learning Python library. Scikit Learn has lots of online support and is a less CPU intensive software to run.
- Since Scikit Learn is more user friendly and less intensive to run and there is a limited size range of available datasets, Scikit Learn was chosen as the most suitable machine learning package to implement for political classification in this project.

2.5 Probability of Twitter account being fake - Bot Detection

There are a few different strategies to implement bot detection on Twitter accounts for this project.
There are some tools already available which can return a probability of a Twitter account being authentic, such as Botometer or BotSight. A custom tool could also be built to determine account authenticity probability based on the public data available about the account

Botometer [10]

- The Botometer tool can be used to return a probability of a Twitter account being authentic. Botometer has a Python API and returns the probability of a Twitter account falling into a number of different bot categories, such as:
    - Astroturf: manually labeled political bots and accounts
    - Fake follower: bots purchased to increase follower counts
    - Financial: bots that post using payment tags for the 'Cash App' app.
    - Self declared: bots from botwiki.org, a website providing information on a range of interesting and creative online bots
    - Spammer: accounts flagged as 'spambots' from different public datasets
    - Other: miscellaneous other bots obtained from manual annotation, user feedback, etc.
- The Botometer analysis is somewhat limited as a student account is restricted to 500 queries per day, however this rate limit should be sufficient for use in analysing the authenticity of individual accounts.

Botsight [11]

- Botsight is a very popular tool for Twitter bot detection which works as a browser extension. Botsight provides a single authenticity percentage for any Twitter account, and does not have rate limits so it can be used for large scale applications.
- The main drawback of Botsight is that, since it does not have a Python API, there would be quite a bit of in-depth implementation required to use Botsight in this project.

Custom Built System

- If none of the bot detectors available prove to be suitable for my specific use case, a simple custom Bot detection system could be built.
- This system would work by analysing the user data that can be fetched with the Twitter API such as account creation date, account location and follower numbers, along with other key determinants of a fake account, to give a probability of an account's authenticity.

After examining all of these approaches to Twitter bot detection, Botometer seems to be the most intuitive and suitable tool to use for bot detection in this project.

3 Implementation

3.1 Project Technologies

Python [12]
The majority of the code for this project was written in the Python language. I chose to use Python as it is very widely used and well supported, it works with the Twitter developer APIs and it allows for easy deployment as a web application.

Python Flask [13]
Python Flask is a Python framework for building web applications with python. It requires some html to render web pages, but makes the process of deploying a Python web application relatively easy.

Python unittest and pytest [14], [15]
Python unittest and pytest are two Python testing tools which support easy building and deployment of Python code tests. Each section of Python has its own set of unittest tests, which can be run together as one pytest test. If any one unit test fails, then the test set will fail, flagging the error. This makes it easier to keep code functional, and keep all the tests consistently up to date.

Twitter Developer APIs [16]
The Twitter developer APIs are written for Python, Ruby and Node.js. They provide an easy and well-maintained API system through which to fetch Twitter data. This is a very efficient way to analyse tweets, however the API has some limitations. When searching for tweets in a hashtag or user account, only tweets from the last 7 days can be accessed, which makes analysing past activity very difficult.

NRC Emotion Lexicon [5]
The NRC emotion Lexicon is a list of English words, mapped along with their association with a range of basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). This dataset was collected through crowdsourcing by the National Research Council of Canada. The data is open source, and can be downloaded in csv format to allow for easy Python processing. This emotion lexicon can provide an easy way to estimate emotions from a dataset of words used in tweets.

Botometer [10]
Botometer, a project created by the ‘Observatory of Social Media’ at Indiana University, is a machine learning based bot detection algorithm. Botometer analyses the public data available about a public Twitter account and its public activity to estimate the probability of the account in question falling into a number of distinct bot categories. Botometer offers a Python API through the rapidAPI framework which supports free accounts to use the Botometer API with a rate limit of 500 API requests per day. Premium Botometer access with bulk and unlimited API calls are available however these tools were not explored as this project was constructed using only freely accessible services.

Kaggle [17]
One significant aspect of building the political leaning detection system was in finding a suitable machine learning training dataset for the model. Kaggle was an integral tool in this process, and allowed for the testing of many different public Twitter datasets. Kaggle is an online system for cooperation, collaboration and

sharing of datasets and models for use in data science and machine learning. Since the Twitter API rate limits restricted the data which could be gathered in bulk for this project, Kaggle offered a solution to this problem in the form of many varied datasets which could be used to train and test the machine learning models.

Git, Github [18]
I used Github to manage my Project throughout its development. This allows for easy data backup and progress logging. Github workflows provide an easy way to automate unit testing on a github project. In my github repository, changes must pass these automated Python pytest and unittest tests before they can be merged into the main branch. This helps ensure that all elements of the code continue to work following each change.

3.2 Sentiment Detection on Tweet Set

The sentiment of a set of tweets, i.e how positive or negative that set of tweets is, is a useful datapoint to examine when analysing or comparing sets of tweets. The implementation of sentiment detection in this project changed and evolved over the course of the project development as the limitations of and flaws in the previously implemented system were detected.

Microsoft Azure Text Analytics Service [4]
The first sentiment detection tool implemented in this project was the Microsoft Azure Text Analytics Service. This is an AI service offered by Microsoft Azure which returns a numerical value in the range of zero to one representing the sentiment of unstructured text data passed to it, zero being very negative and one being very positive.
This Service is very efficient and incorporates well with the Twitter Developer APIs. The Text Analytics Sentiment Analysis scores could be determined directly from a set of tweets, without the need for any data parsing or manipulation. The biggest drawback to the use of the Microsoft Azure Text Analytics Service within this project was the API rate limit of 5000 requests per month. With this system implemented, that 5000 request rate limit was exceeded within two days of development work, demonstrating early on in the development process that the system would not be a sensible choice for analysing sentiment in this project.

NRC Emotion Lexicon [5]
The next sentiment detection tool was implemented along with emotion detection, using the NRC emotion Lexicon.
This system was implemented in the project in a very straightforward way. Integer variables were created to store the sum of positive and negative word matches found. Each word used in the set of tweets was checked to see if it matched a word within the NRC emotion lexicon. If it did, then the binary values for positive and negative sentiments were added to the positive and negative sum counts. Once all words had been checked, the positive and negative sum counts were then compared to see which emotion was represented more often in the set of tweets, and a summary phrase was printed to describe the ratio between the positive and negative word representations.
The outcome of this system was that almost any set of tweets would be described as ‘more positive than negative’, including #covid, #disgrace and even #death. This bizarre behavior can be attributed to the fact that only 14,000 words were classified within the NRC emotion Lexicon, and the tool was built primarily for emotion analysis and not sentiment analysis. Some attempts were made to improve the system performance,

such as scaling the positive and negative sum counts in proportion to their representation within the 14,000 word database however none of these efforts improved the accuracy of the system.

VADER SentimentIntensityAnalyzer [6]

Once it became clear that an sentiment detection alternative must be found to the NRC Emotion Lexicon, the VADER Sentiment Intensity Analyser quickly emerged as a possible alternative for easily analysing the sentiment of text data.

This system was implemented by treating each tweet as separate strings, and determining the most likely sentiment for the string from the sentiment intensity ratios it returned. This provided an easy way to see the representations of the various emotions across sets of tweets.

There are still some drawbacks to this system, most notably the fact that the system cannot understand sarcasm, cynicism or satire. This can lead to discussions criticising and mocking a certain standpoint being understood as positive. These issues in detecting the nuance and complexity of language online were present in all systems investigated and solving the difficult problem of teaching sarcasm and cynicism to an algorithm goes beyond the scope of this project.

The Sentiment Intensity Analyser produced by VADER is a simple and accurate way to analyse sentiment from text data without imposing rate limits or oversimplifying the data.

A flow diagram of the implemented Sentiment Analysis System is shown below in Figure 3.
A full page version can be found in the appendices (Section 8.2.1)

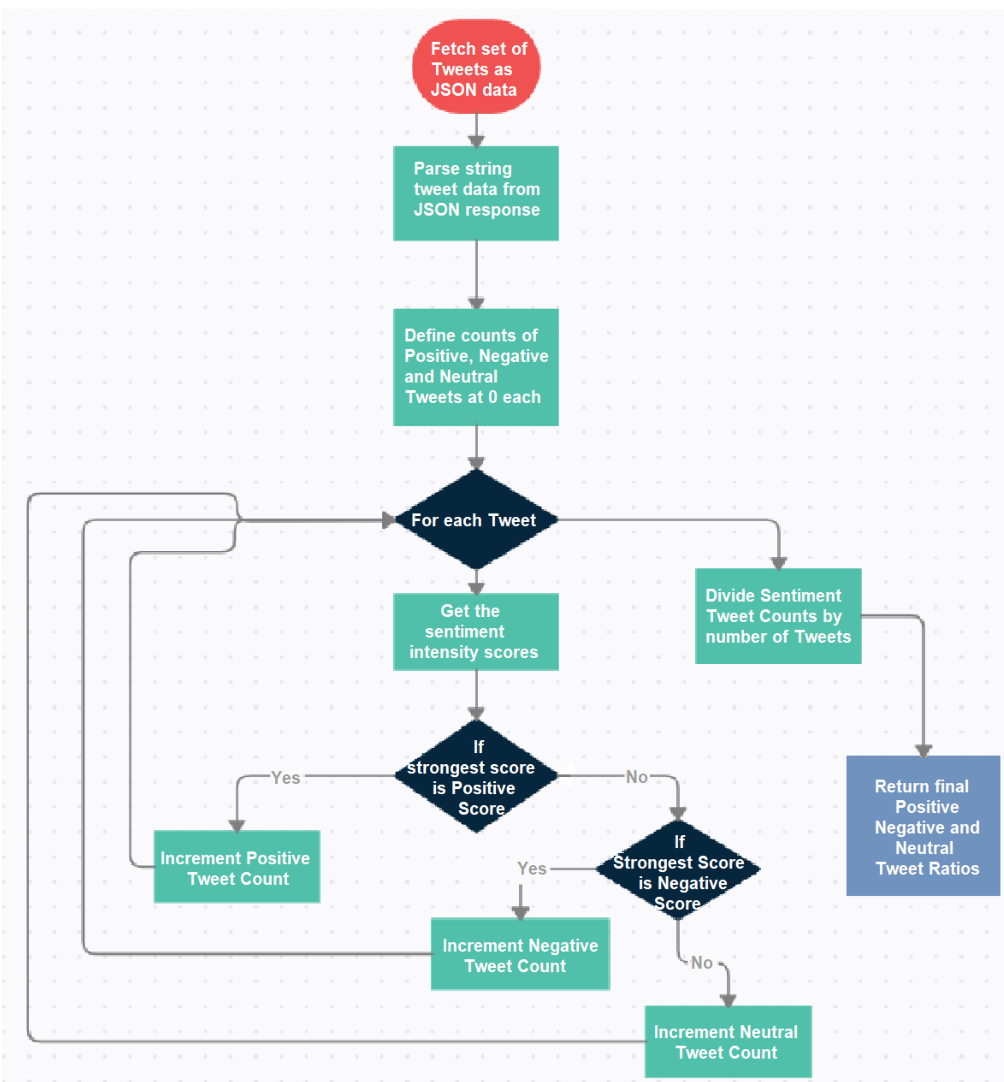


Figure 3: Sentiment Analysis System Flow Diagram

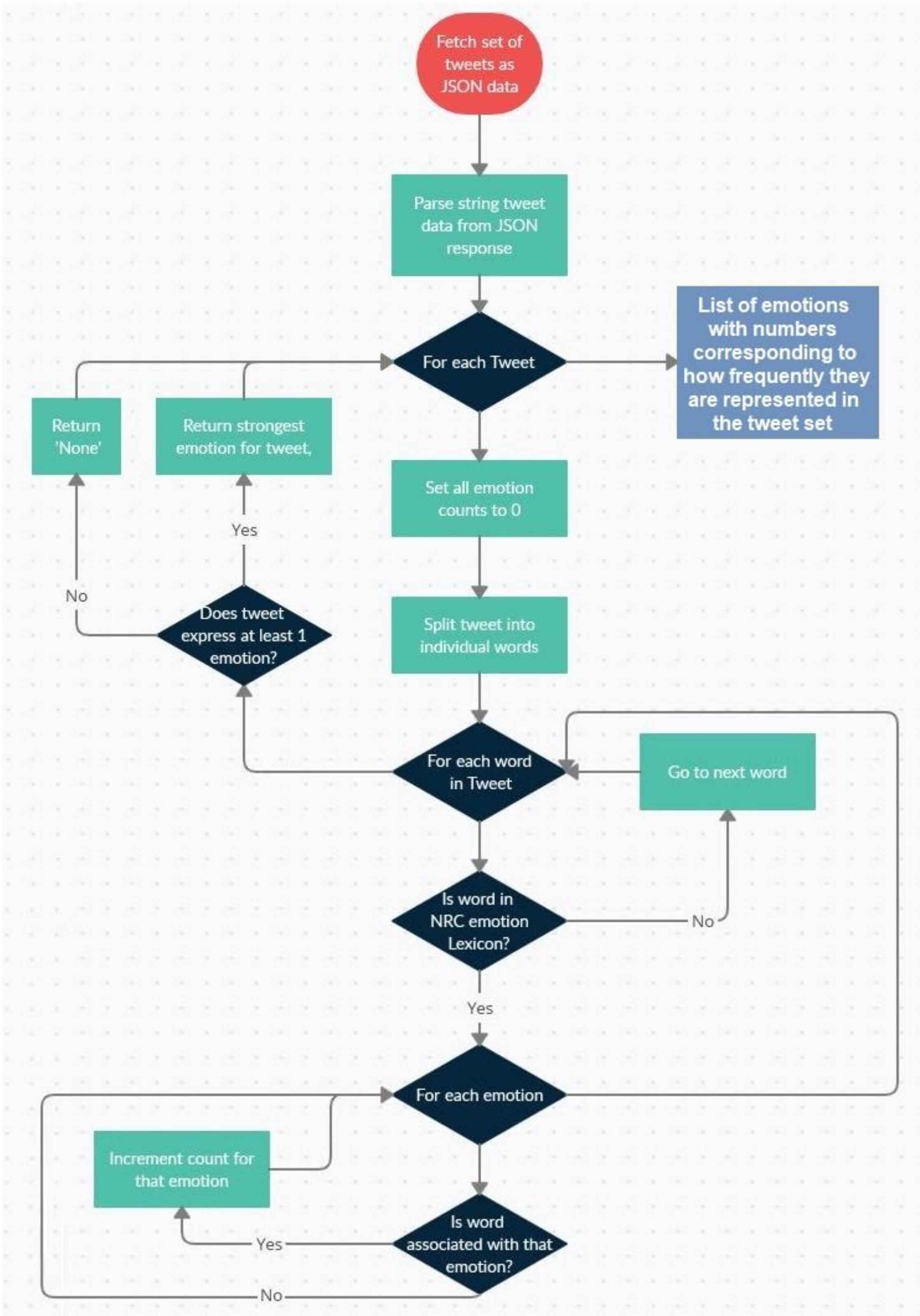
3.3 Emotion detection on Tweet Set

Emotion detection and analysis was implemented in this project using the NRC Emotion Lexicon.
The initial system implemented with the NRC emotion lexicon analysis was a very basic version, where all words used within a set of tweets were checked, and for any word which had an entry in the NRC emotion Lexicon, the counts for the emotions expressed by that word were incremented. The positive and negative sentiments were disregarded for this evaluation as the focus was on pure emotions, and not sentiments.
This system was mostly functional, however it had some drawbacks.
Once the full set of words used within english language tweets was collected for almost any search parameter, this emotion detection system would conclude that the strongest emotions represented were trust, joy and anticipation. The order of which was most prevalent would vary, however the emotions themselves were almost always the same, regardless of the search parameter used to collect tweets. This issue seemed to be a result of language use across Twitter on average representing these emotions more strongly than any others, when words used in tweets are treated as one single dataset to evaluate.

This issue was resolved by treating each tweet as an individual data string on which to analyse emotion, as this allowed for more infrequent and nuanced tweets to be represented in the dataset. The words used in each individual tweet were compared against the NRC emotion lexicon to find word matches, with emotion counts being used to find the most prevalent emotion within the tweet. Thus, each tweet was allocated a single strongest emotion to account for the varying of emotions conveyed across topics discussed and days of data. The emotions of Anticipation, Trust and Joy are still represented strongly for many search queries, however if less frequent emotions such as Anger or Fear are detected as the strongest emotions for a number of tweets, these data points are represented and shown within the ratio of detected emotions. Data visualisation within the user interface was then used to show the emotions detected and the proportion of the tweets for which these are the strongest emotions.

A flow diagram of the implemented Emotion Analysis System is shown on the next page, in Figure 4
A full page version can be found in the appendices (Section 8.2.2)

Figure 4: Emotion Analysis System Flow Diagram



3.4 Political Leaning Detection on Tweet Set

The political leaning detection was the most complex aspect of this project to implement. This was because of the complexities of language classification and the peculiarities of political speech.

The goal of this component of the project was to find a way to build a simple, accurate tool for classifying tweets on a spectrum from left leaning or liberal, to right leaning or conservative. The biggest difficulty that arose in achieving this goal was in the different speech patterns and policies of political groups across different countries. The speech of Irish conservative politicians tends to have more in common with American liberal politicians than American conservative politicians. The Twitter usage and speech of politicians from all political groups seemed to be based much more in populist performance than in conveying information, making it difficult to differentiate political groups based on Twitter use alone.

The final implemented political leaning system consisted of a machine learning algorithm with options for different political region datasets training this model. These datasets were constructed as part of this project. Details on the performances of, and flaws in, all existing datasets which could be found online are explored in section 3.4.2, with the process through which the custom datasets were collected explored in section 3.4.3. The implementation of multiple political region datasets allows the model to give a more accurate and nuanced analysis of the political leaning of a set of tweets than if all regions were placed on the same political spectrum.
A flow diagram of this implemented system is shown below in Figure 5
A full page version can be found in the appendices (Section 8.2.3)

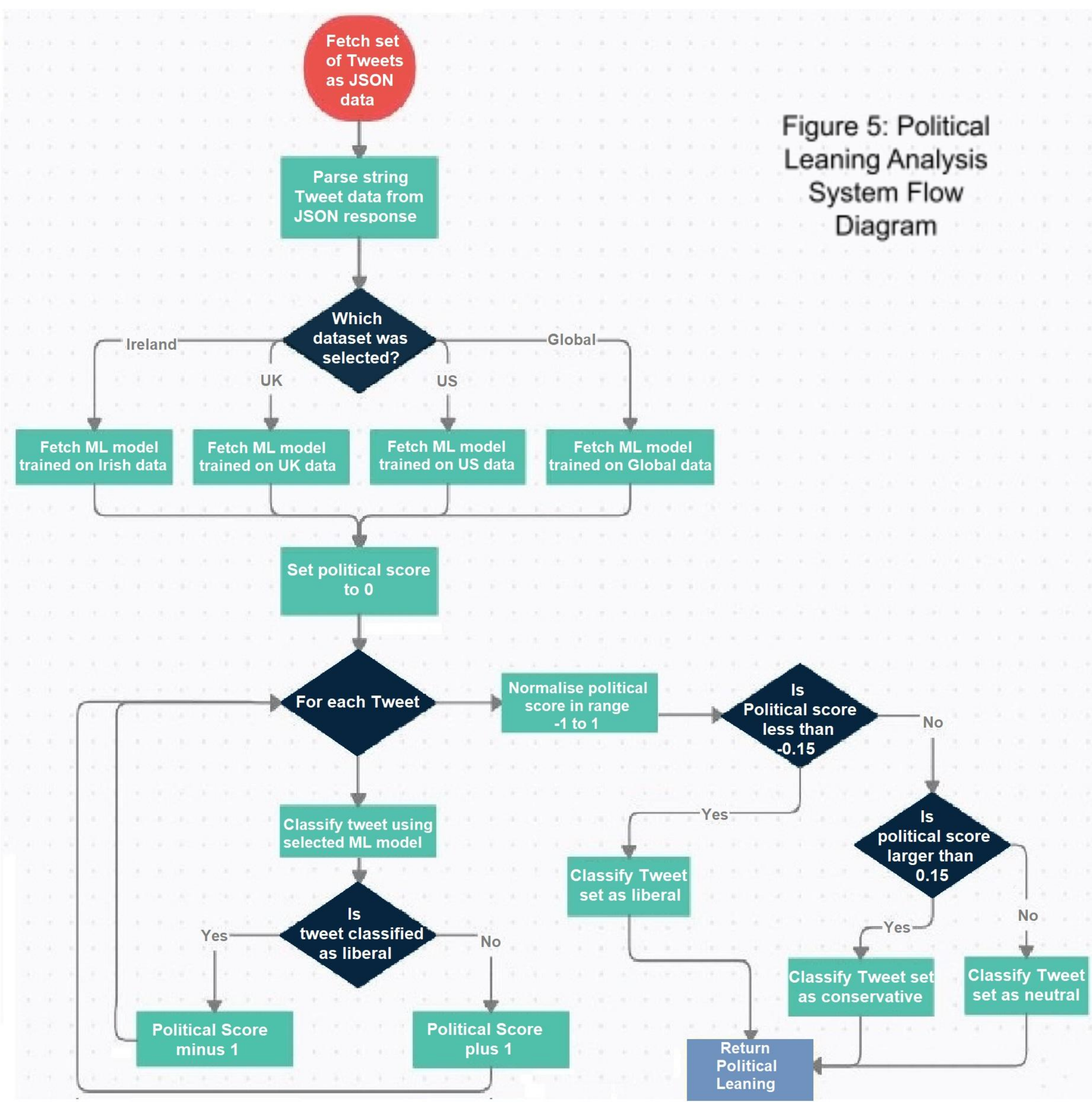


Figure 5: Political Leaning Analysis System Flow Diagram

The implementation of this political leaning analysis went through many stages over the development of this project, in the effort to develop the most accurate system possible in light of the linguistic challenges faced. This process is detailed in depth in the following sections.

3.4.1 Political Sentiment Lexicon

The first system was implemented using a simple word mapping system, where a set of words were mapped to political leanings. The dataset used for this word mapping was the Political Sentiment Lexicon [7]. This dataset was constructed by an MBA student in 2017, and includes an in-depth report into the dataset development. This level of insight into the dataset creation is very valuable, however there were some significant shortcomings of this dataset in classifying political speech in this project.

The dataset is very clearly a dataset based on the Politics of the United States of America. Examples of words included are americorps (2), charlottesville (-3), guantanamo (4) and ivanka (-3). These words are undeniably American words, which could not be used to accurately classify politicians from other countries on the political spectrum. Notably within the dataset, locations within the USA and the names of American public figures are more common among the speech of conservative politicians, and thus are associated with negative values. The natural progression of this would be that this dataset would have difficulty identifying conservative politicians from other countries.

This dataset was constructed in 2017, and is thus based on data which is at least 3 years old and political speech evolves quickly, especially when much of the political discourse happening in 2020 and 2021 is related to the COVID-19 pandemic using words with which most members of the public would not have been familiar in 2017. It is reasonable to assume that the use of words such as positive (-2), restrictions (-3), healthcare (-4) and capacity (2) would have a different meaning in 2021 to their meaning in 2017. Thus, this dataset would be unable to interpret any political leaning from conversations about restrictions, lockdown or facemasks, and may make incorrect classifications based on the usage of these words in the pre-pandemic political lexicon.

Despite these flaws, this dataset was still quite accurate in classifying American politicians as liberal or conservative, correctly identifying key Democrats such as Alexandria Ocasio-Cortez as liberal and key Republicans such as Donald Trump as conservative. It was highly inaccurate, however, at classifying political speech from any other country and failed to recognise politically neutral or centrist viewpoints in their Twitter data. This makes sense given the simplicity of the dataset and the naivety of the implementation. It was clear that a different way of analysing political leaning from Twitter data was required.

3.4.2 Machine Learning

The next option explored as a method to assign a political leaning to a set of tweets was machine learning. The Scikit Learn [9] Python machine learning package allowed for easy implementation of a text classification system which can classify unseen text data based on a model built from a training dataset. The machine learning code was relatively straightforward to implement, however finding a suitable dataset with which to train the model was a significant challenge. The process of exploring potential training datasets provided the opportunity for a more in depth linguistic exploration of political speech and the complexities of language used online, specifically the language used in political posts on Twitter. The political leaning datasets explored over the course of this project were as follows:

- *Ideological Books Corpus* [19]
  In researching potential datasets to use for this political classification, the first dataset which stood out as a well researched and accurate dataset on which to base my machine learning algorithm was the Ideological Books Corpus. This dataset, which was constructed in 2013, consists of 4062 sentences from various books and articles with each sentence classified as either liberal, conservative or neutral.

  This dataset, constructed by the University of Maryland, is a very well researched and developed dataset and so it was a surprise when it was highly inaccurate at classifying politicians' tweets when implemented in the machine learning system of this project. The tweets of all left leaning politicians tested using this dataset were classed as right leaning. Many of the openly and notably left leaning politicians from around the world were classified as more strongly right leaning than many of their strongly right leaning counterparts.

  After checking that there had been no errors made in the implementation of the dataset, it was important to identify why this dataset was performing so badly as a training dataset. To do that, it was important to look at exactly how this training data differed from the data it was attempting to classify. The main differences between the training data (sentences from political books and speeches in 2013) and the data to classify (tweets posted by public figures and politicians in 2021), was the year and form of communication. The difference in time, between 2013 and 2021, represents a significant shift in global political ideology as over the time span between 2013 and 2021 there was a significant global movement towards nationalist populism movements led by authoritarian leaders. It could be said that global conservatism became much more entwined with fearmongering and xenophobia than it had been in prior decades or at the very least, many conservative movements became more open about their xenophobic views. Combined with that significant shift in political discussion over the time span of 2013 to 2021, there is also a significant difference between how politics is discussed in official forums and formal settings, such as speeches and manifestos, and how politicians post online and communicate through more informal means. Politicians who are using Twitter to spread a message are incentivised to simplify the language used and the messages conveyed to reach as wide an audience as possible, and so not to exclude those without the education level to understand the highly formal language used in laws and political articles. When conveying those same points to other politicians, academics or experts, these same messages may be conveyed in a complex and verbose manner so as to convey a well researched and intelligent argument.

  When considering these two significant differences between the training dataset and the data being classified, it becomes clear why the Ideological Books Corpus failed to accurately determine the political leaning of tweets. Twitter feeds in 2021 are just far too different from political articles in 2013 to use one in classifying the other. It was important to keep these lessons in mind in exploring and examining other potential datasets to use for this political classification of tweets.

- *Covote Dataset* [19]
  One notable dataset for classifying political speech which was identified early on in the development of this project was The Convote dataset. This dataset was constructed in 2006 from transcripts of the speeches made in the United States Congress by a variety of politicians. In total 7,816 sentences were used to construct this dataset and thus it stood out as a definite and thorough dataset to use.

Upon identifying the reasons that the Ideological Books Corpus was an unsuitable dataset for classifying tweets from 2021, it became clear that this dataset would result in those same inaccuracies. It was constructed in 2006, before the release of the iPhone and the explosion of social media. Political speech in 2006 was certainly not going to be accurate at classifying tweets from 2021, especially given the extreme formality of speeches made in the US Congress. There were far too many differences between this training dataset and the data which needed classification for this dataset to be accurate in determining the political leaning of sets of tweets.

- *Political Twitter Corpus* [20]
  After eliminating the Ideological Books Corpus and the Covote Dataset as suitable to use in political tweet classification, another potential strategy emerged to increase the accuracy of this classification. The Political Twitter Corpus was collected in 2012 and consists of 4000 total public tweets classified as either political or non-political. This dataset would allow for another level of analysis of the data, and could potentially improve the accuracy of political tweets classified with a separate dataset by determining whether or not a set of tweets was political before attempting to assign them a political leaning. This dataset has the significant advantage of being created from Twitter data, and so will be much more accurate at classifying the specific forms of speech used on Twitter.

  Upon implementing and testing this dataset however, every set of tweets tested were classified as political, even when the tweets were notably unrelated to politics in any way. This may be due to how the use of language on Twitter has changed since this dataset was collected in 2012, or may just be due to inaccuracies within the dataset itself. Upon seeing that this dataset significantly underperformed at determining whether or not tweets were political, the dataset was abandoned as it was not a core requirement of the project and was not contributing at all to the quality of the political classification of sets of tweets.

After learning the weaknesses of, and flaws within, the previously investigated datasets, further datasets were eliminated if they did not meet a few key criteria. The dataset must be based on social media posts, preferably Twitter data but other informal social media forms would be acceptable. The dataset must also be from no earlier than 2015, which was the year in which Donald Trump first became a political figure and his form of agressive, xenophobic politics began to gain significant popularity across the world. With these new criteria in mind, the number of potential datasets was greatly reduced.

- *Harvard Dataverse Twitter Datasets* [21]
  Public datasets related to academic research were some of the most promising contenders to use in political Twitter classification, with datasets available in the Harvard Dataverse 'George Washington University Dataverse' Libraries being very promising contenders for implementing in the machine learning algorithm of this project. One interesting dataset which looked extremely promising was the '2018 U.S. Congressional Election Tweet IDs' dataset. This dataset contained tweet IDs related to tweets by republican and democrat candidates for the US Senate and US House of Representatives elections in 2018, between the dates of January 22nd 2018 and January 3rd 2019. This is a total of 171,248,476 tweets, which could provide a very thorough dataset for classifying political Twitter speech based on how these candidates used Twitter over the recent period of time of 2018 to 2019. The major problem with this dataset, however, was that it contained tweet IDs and not the tweet text itself. This would have meant that for every tweet id, a specific request would have had to be made to the Twitter API to fetch the content of that tweet. While this process could be automated and left to

work without much difficulty, the Twitter APIs have a rate limit of 500,000 requests per user per month, and gaining access to the Twitter APIs for a single Twitter account was difficult enough to make establishing multiple Twitter accounts beyond the realm of reasonable possibility. Upon further investigation, it became clear that all tweets within the Harvard Dataverse Twitter Datasets had this same issue , as this is required by Twitter's developer policies which state that "*tweet IDs may be publicly shared for academic purposes; tweets may not*". This made it clear that any Twitter dataset available on the internet would thus either present this challenge by using only Twitter IDs, or have been constructed in an informal manner by a group which either didn't know or didn't care that they were breaking Twitter's developer policies by sharing a dataset of public tweets.

- *Kaggle Political Tweet Classification Dataset* [22]
  One large forum for sharing datasets is Kaggle [17]. It is an online system for sharing and collaborating on datasets, and so there was the potential to find a naively researched but well developed dataset of tweets on Kaggle which would provide the text of tweets in breach of the Twitter Developer Policies. The largest and most well implemented dataset which could be found on Kaggle for political Twitter classification was the 'Democrat Vs. Republican Tweets' dataset collected in 2018 and consisting of 95274 tweets labeled with the politician who made the tweets, and the US political party to which they belong (Democrat or Republican). This dataset was promising in ticking a number of key dataset criteria. It was classifying Twitter data and it was recent, however it was only looking at politics from a US perspective and the dataset was in breach of the Twitter API developer guidelines by being publicly available and so it could not be assumed that the data was trustworthy. With those pros and cons in mind, it was relatively simple to implement and test the dataset to discern its accuracy, and this was the most accurate dataset implemented to date. When classifying US politicians, the machine learning model trained with this dataset was reasonably accurate and could correctly classify politicians on a range with respect to one another, however it was flawed in that it was strongly biased towards classifying politicians as right leaning or centrist and would very rarely classify politicians as left leaning. This dataset was extremely unpredictable and inaccurate at classifying politicians from countries other than the USA, which identified a potential fundamental flaw in classifying left leaning US politicians with left leaning UK politicians, or right leaning Irish politicians with right leaning US politicians. These political wings could be loosely categorised together however they really did not align enough to expect an algorithm to identify which politicians are globally right and left leaning based on their speech.

- *Custom Seperate National Tweet Datasets*
  From the thorough investigation of the different available political datasets, it became clear that the best solution may be to develop a custom dataset, or set of datasets, specifically for this problem. These datasets would be limited by the rate limits of the Twitter developer APIs. These APIs do not allow for the bulk collection of tweets which are more than a week old, and limit requests to 500,000 a month for student access. This strategy would also raise the issue of how to determine what tweets should be the training data for liberal and what tweets should be the training data for conservatives. One significant advantage of this approach would be that it would allow different datasets to be made for different political regions, to avoid the complexities and inaccuracies of lumping all global political environments into one single dataset.

  To implement this strategy on a trial basis, I attempted to construct a database for use in classifying tweets related to irish politics on a spectrum from left to right leaning. This dataset should be able to

correctly assign the correct political leaning to irish politicians. Simply collecting tweets from politicians and assigning them a political leaning could potentially work, however it would then be difficult to test the system on politicians since the machine learning may then be classifying tweets which it has already experienced in the training dataset.
With this in mind, the strategy taken for creating this dataset was to collect the tweets from the official Twitter accounts for the main political parties, annotated with the political leanings of those political parties. Tweets were collected from the primary conservative Irish parties of Fianna Fail, Fine Gael and Renua, and the primary liberal Irish parties of Labour, Green Party, Social Democrats, People before Profit and Sinn Fein. That dataset of tweets was then scaled at random such that each political leaning was equally represented in this training dataset, as there were more tweets found for the liberal parties than there were for the conservative parties. Only tweets from the previous 7 days could be collected in these datasets. Despite this limited training data, this dataset was the most accurate dataset examined so far at classifying Irish politicians based on their Twitter activity, with this dataset being capable of classifying, within a good range of accuracy, where a politician fell on the Irish political spectrum between left and right leaning.

Based on these results, the decision was made to move forward with this strategy of machine learning using custom datasets for different political spheres for the political leaning detection system within this project.

3.4.3 Dataset Collection

A Python script was created which could fetch and sort tweets with a known political leaning into csv files for easy processing within the machine learning system. This could then be run weekly over the continued project development to expand and improve the datasets. A decision had to be made at this point on how many national political spheres to consider, and which national political spheres on which to base these datasets. Since this project was focusing only on english language data, this eliminated any countries in which english is not the primary language used in political discussion. This also eliminated Canada as a country to consider, as one of Canada's 4 main political parties, Bloc Québécois, tweets only in french.
Both Australia and New Zealand were eliminated as political spheres as there did not seem to be much Twitter activity by politicians within these countries. This left 3 main political spheres to examine, those being Ireland, the UK and the USA.

As in the dataset collected to test this system, the Irish dataset consists of tweets from the primary conservative Irish parties of Fianna Fail (@fiannafailparty), Fine Gael (@finegael) and Renua (@renuaireland), and the primary liberal Irish parties of Labour (@labour), Green Party (@greenparty_ie), Social Democrats (@socdems), People before Profit (@pb4p) and Sinn Fein (@sinnfeinireland)

The UK dataset consists of tweets from the primary conservative UK parties of the Conservatives (@conservatives), UKIP (@ukip) and the DUP (@duponline), and the primary liberal UK parties of UK Labour (@uklabour), the SNP (@thesnp) and the Liberal Democrats (@libdems).

The USA dataset was slightly more complex, due to the 2 party system. The strategy implemented in collecting this dataset was to fetch the tweets from all the official Twitter accounts associated with the republican party and the democrat party. The official Twitter accounts associated with the republic party were @gop, @senategop and @nrsc. The official Twitter accounts associated with the democrat party were @senatedems, @housedemocrats and @thedemocrats. The Kaggle dataset was maintained as backup for classifying political tweets from the USA, as it was quite accurate and contained a larger set of tweets than could be reasonably fetched within the custom dataset over the course of this project.

The collection of these datasets continued over the course of this project's development with a new option added to the GUI to select which dataset to use in analysing the political data. These options were Ireland, UK, USA or global, which was the collection of the other 3 datasets.

3.5 User Interface Design

For the first few months of this project's development, the project work was done using the command line and a very simple html webpage as a basic user interface placeholder. The project scope and plans were still evolving and it made sense to postpone the development of any complex user interface design until the project scope was more defined.
The user interface was not a large aspect of this project as the main focus of the project was on the natural language processing and machine learning functions of the application. The addition of the user interface was a way to show the complex data in a clear way, without the focus of this project becoming web development or graphical design. The user interface design and development was allocated roughly 10% (16 hours) of the available development time.

3.5.1 Initial Mockup Designs

A mockup user interface design was developed as part of the project planning phase, using Microsoft Powerpoint to create graphics, and using a minimal muted colour palette to avoid creating an overly noisy design. The logo for the project, which was created as part of this user interface mockup design, was created from a simple graphic of a magnifying glass, with the Twitter logo placed inside the magnifying glass. This logo was also created within a powerpoint file, and exported and saved as a 'png' image, which is an image format that supports transparency.

Figure 6: Mockup Analyse Tweets Page

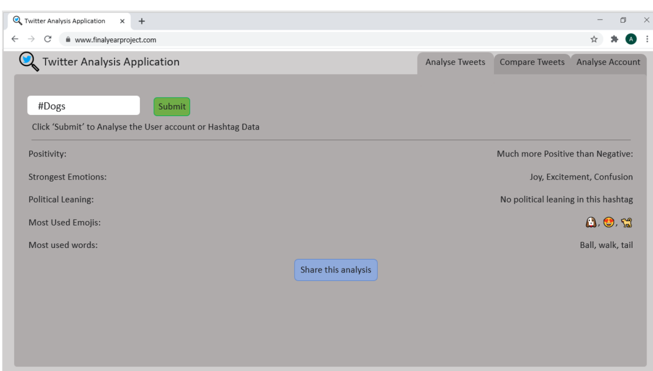


Figure 7: Mockup Compare Tweets Page

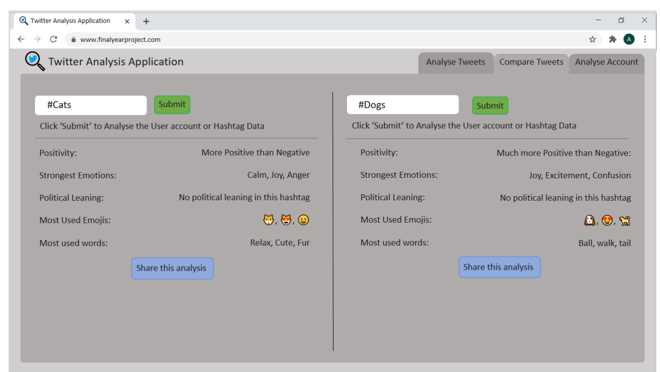


Figure 8: Mockup Analyse Account Page

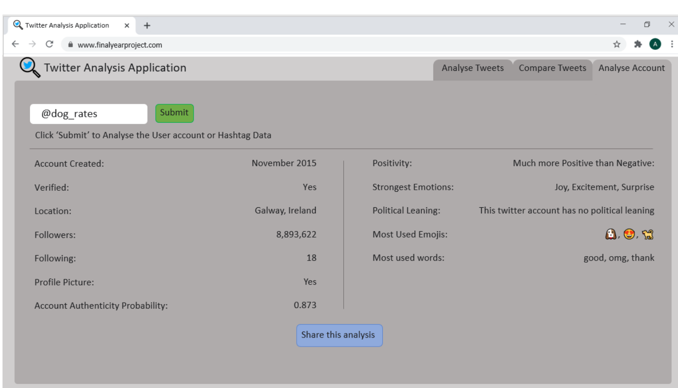


Figure 9: Project Logo

3.5.2 Implementation of Initial User Interface

Task 10 of the project development plan was implementing the first draft of the user interface for the application. This was completed using Python Flask, html, css and Python jinja templating.

The user interface development went through a few phases:

1. Basic Initial Implementation

The first version of the user interface was based entirely on the mockup, and represented the data very simply. The application had two functional pages, the Analyse Tweets and the Compare Tweets pages, as more backend project work was needed to complete the planned Analyse Account / Analyse Content Functionality.

Figure 10: V1 of Analyse Tweets Page

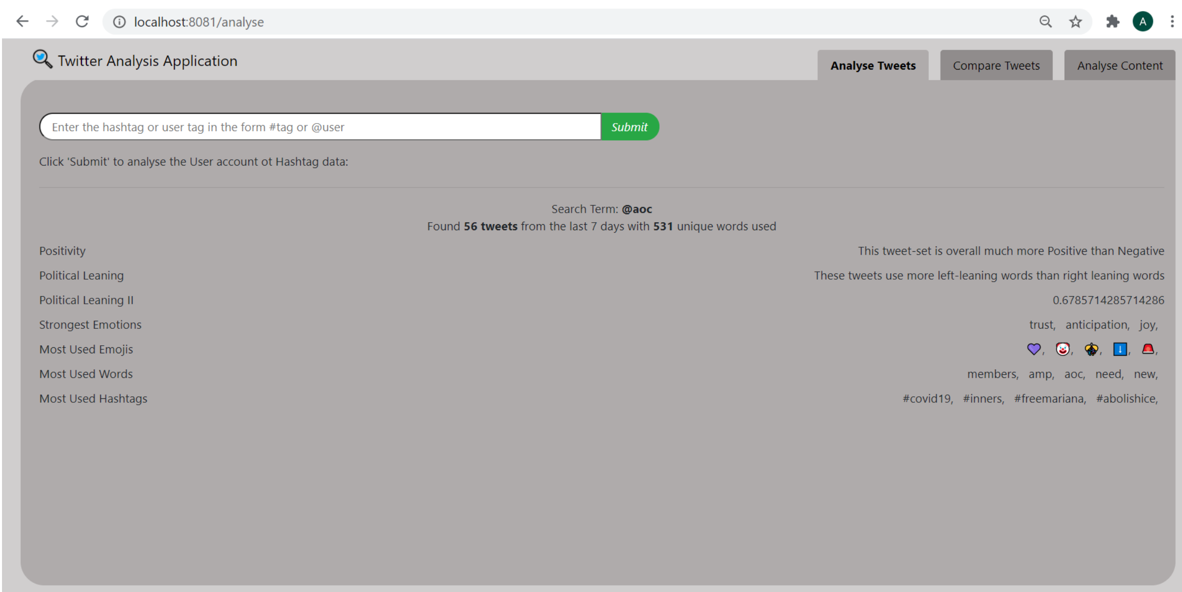
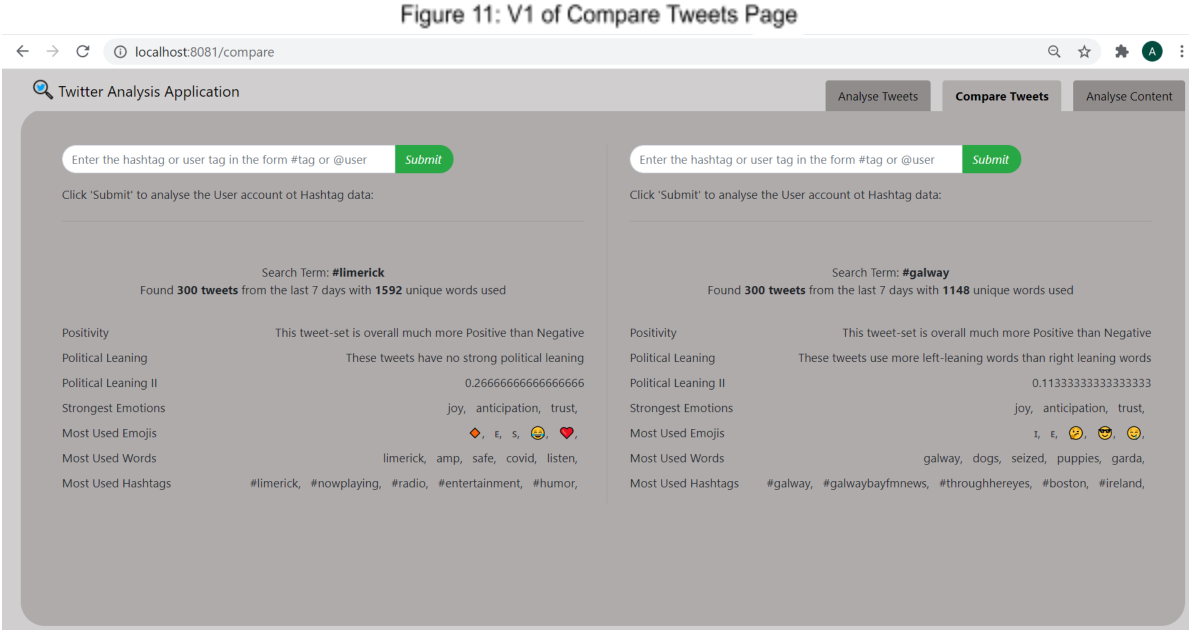


Figure 11: V1 of Compare Tweets Page



2. Bootstrap Responsive for Mobile and Tablet

The application was updated to be easily scaled to different devices using the Bootstrap [23] framework. This enabled the same application to work seamlessly across a variety of browser sizes and devices.

Figure 12: Bootstrap Responsive Compare Tweets Page (Desktop)

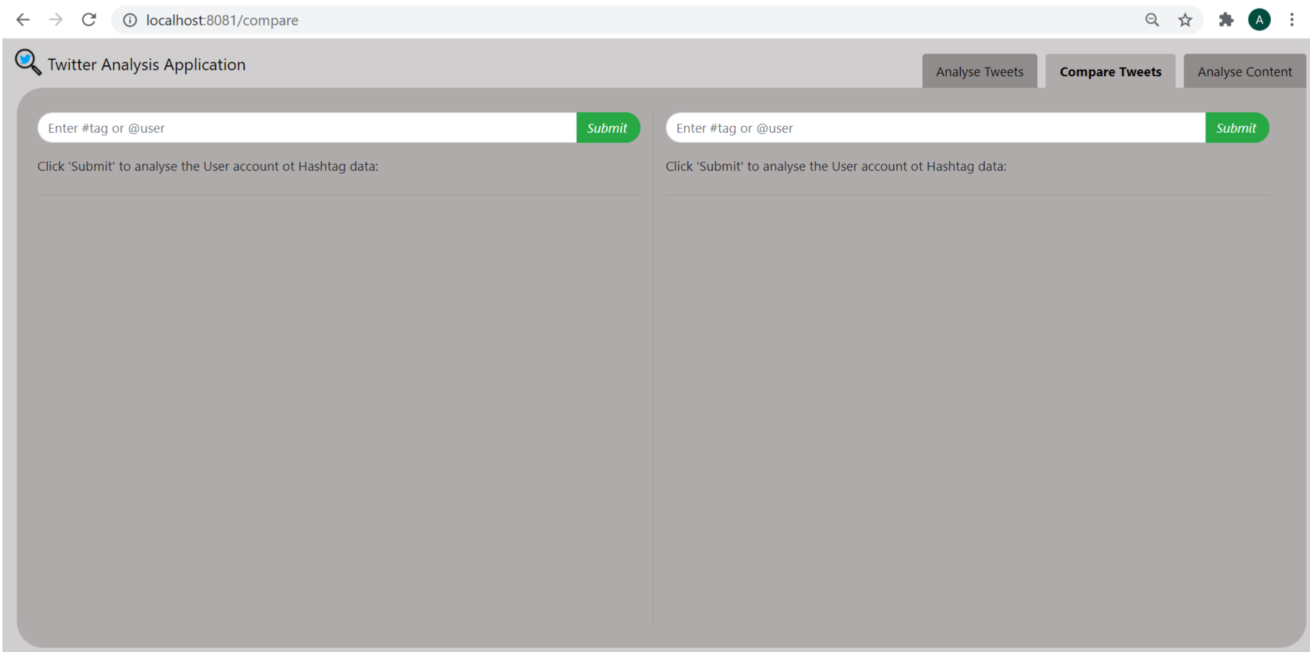


Figure 13: Bootstrap Responsive Compare Tweets Page (Tablet)

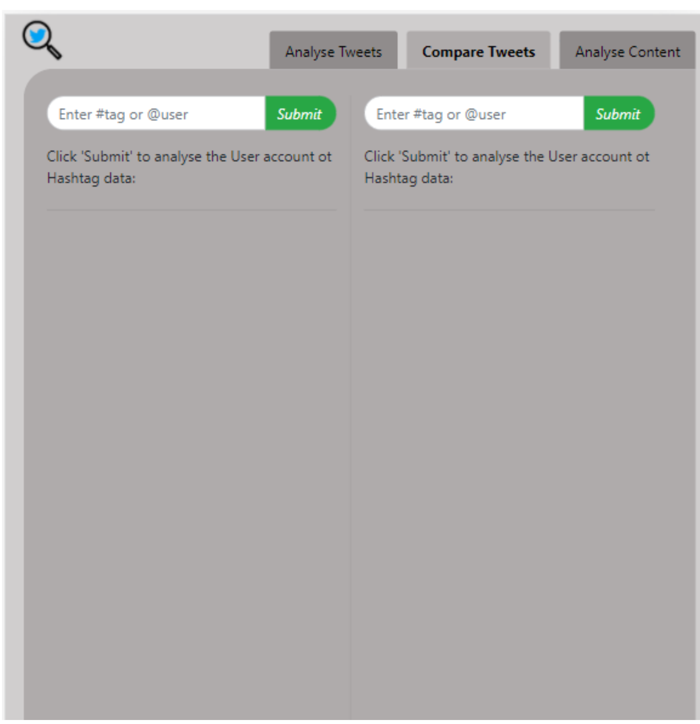
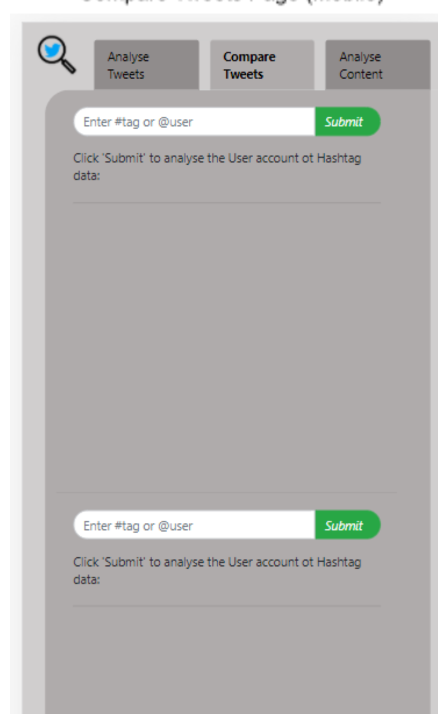


Figure 14: Bootstrap Responsive Compare Tweets Page (Mobile)



3. Political Region Toggle Implementation

Once the implementation of the political leaning analysis had been completed using multiple datasets from different regions, this functionality had to be incorporated into the user interface through the use of a selection toggle, allowing the users of the application to select what region on which they wished to focus their political leaning analysis.

Figure 15: Political Region Toggle Implemented (Desktop)

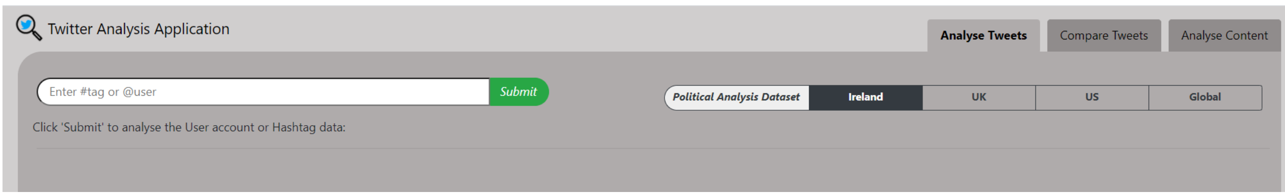
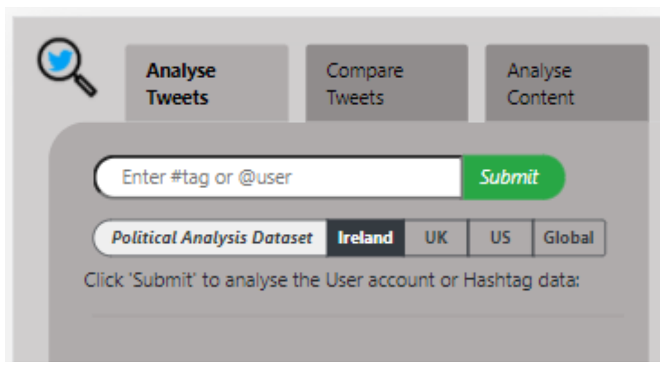


Figure 16: Political Region Toggle Implemented (Mobile)



4. Error Messaging

A system was needed to inform the users of any mistakes/errors which could occur.
The error situations handled with these messages were

- Error: User submits form with no Twitter search query typed
- Error: Invalid query typed (i.e not in form @query or #query)
- Error: No tweets found for an otherwise valid query
- Info: Fewer than 20 tweets found for a query (This will lead to less accurate and trustworthy analysis, as it is too small a dataset). Analysis will be performed but the user must be informed than conclusions are less trustworthy.

Figure 17: No Search Query Error (Desktop)

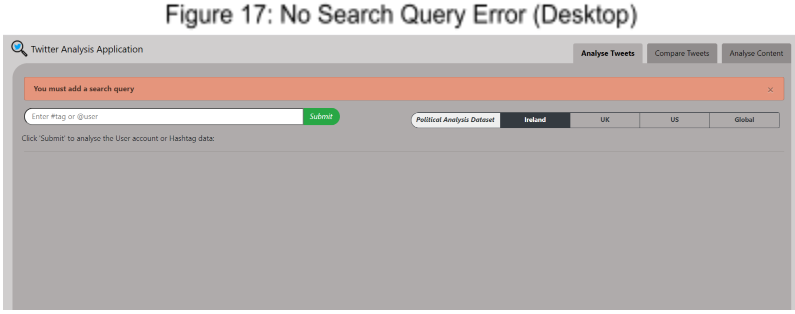
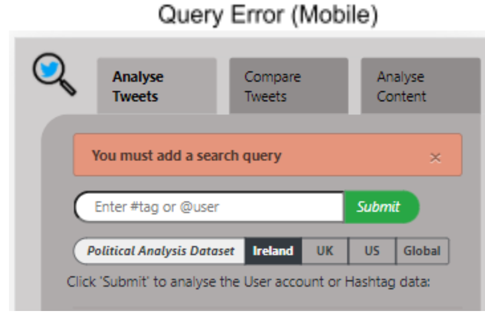


Figure 18: No Search Query Error (Mobile)



5. Loading Animation

Some queries were taking quite a while to return results, and it was not obvious to users that the application was working. To make this process easier to understand, a loading animation was added while Twitter data was being analysed so that users would know that the application was working and not frozen.

Figure 19: Loading Animation Screenshot (Desktop)

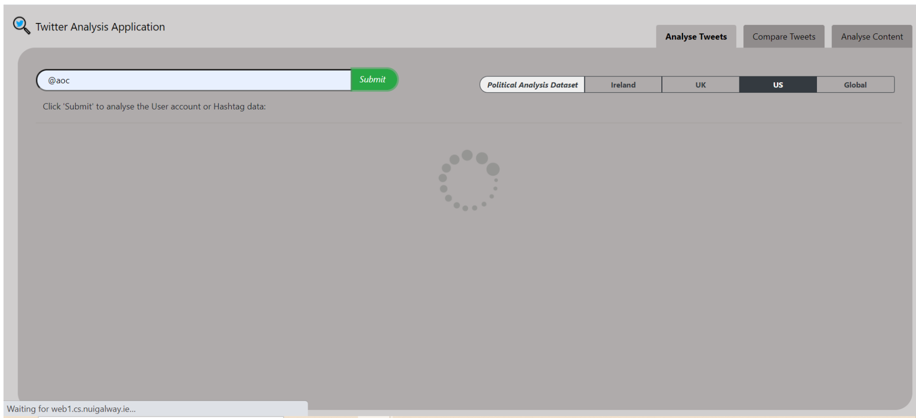
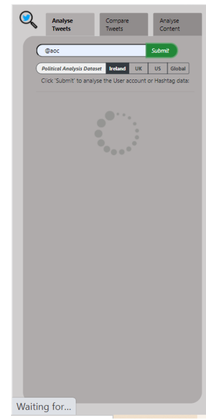


Figure 20: Loading Animation Screenshot (Mobile)



6. Analyse Account Page Added

Once the Backend functionality of the Twitter Account Analysis had been completed, the user interface for the analyse account page was developed. This included some basic Twitter account stats such as follower count and following count, along with the Twitter profile photo, account bio, verification status, pinned tweet, number of tweets posted and botometer account authenticity data. The analyse account page also includes all the regular analysis that can be performed on a Twitter query.

Figure 21: Analyse Account Page V1 (Desktop)

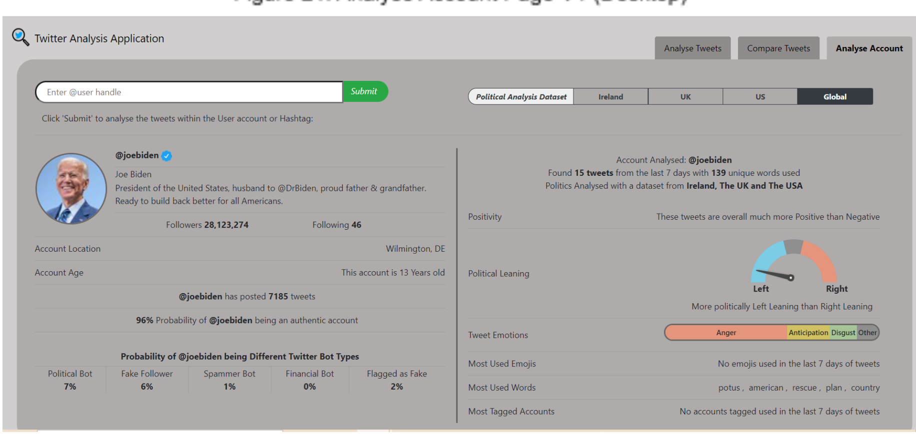
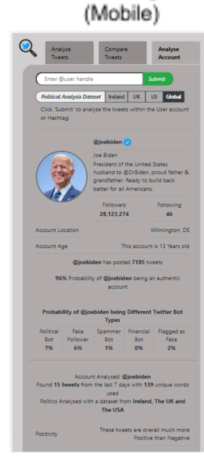


Figure 22: Analyse Account Page V1 (Mobile)



7. Home Page Added

In conducting user evaluations, some users were unclear about much of the application functionality. To ease this confusion, a home page was added to the application which gives information on

- The Analyse Tweets Page
- The Compare Tweets Page
- The Analyse Account Page
- The Political Analysis Dataset System
- Explanations of the various result fields.

Figure 23: First Screen of Home Page V1 (Desktop)

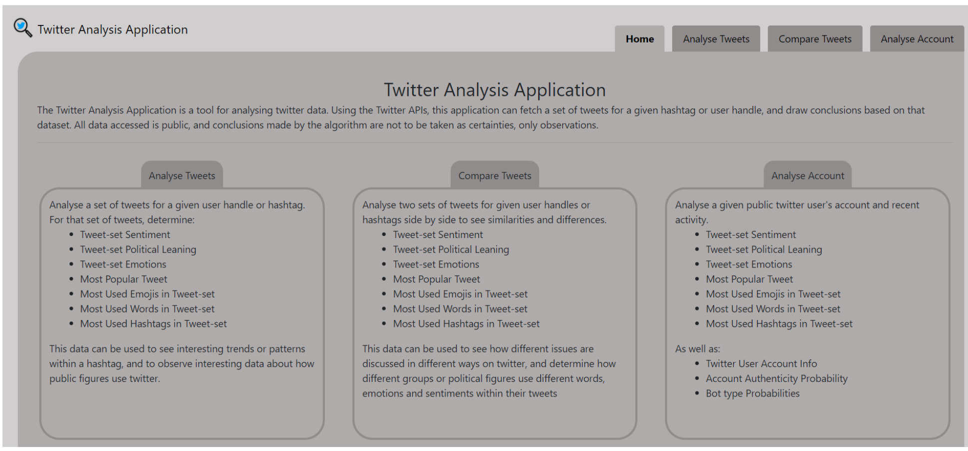
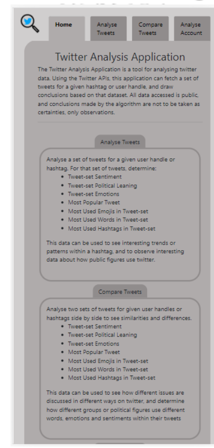


Figure 24: First Screen of Home Page



8. Re-Format of Compare Tweets Page

The feedback from user evaluations showed that some users were confused by the Compare Tweets' page, and would prefer a more direct comparison to be shown than simply displaying the data side by side. This page was updated to include a common search field and single submit button, along with a direct comparison section between the sets of tweets.

Figure 25: Compare Page V2 (Desktop)

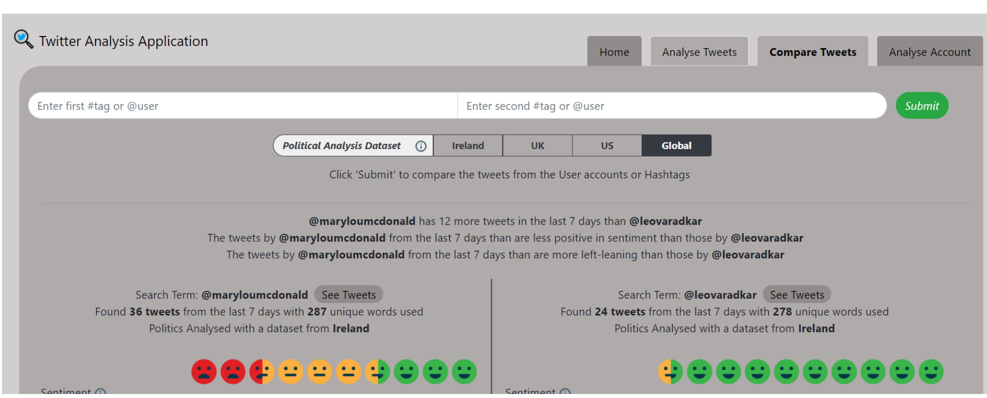
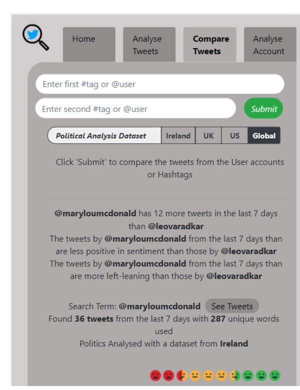


Figure 26: Compare Page V2 (Mobile)



Other Updates

Other minor user interface updates over the course of the project development included:

- Informational Bubbles beside complex project sections
- 'Share on Twitter' Button
- Text size increase (in response to user input)
- Show most popular tweet of tweet set (in response to user input)
- Add 'see tweets' button to see the set of tweets that is being analysed on Twitter (in response to user input)
- Show Informational pop-up messages in a different colour than error messages. Error messages are shown as red, info messages are shown as yellow

3.5.3 Data Visualisation

One important factor in creating a clear and intuitive graphical user interface was applying suitable data visualisation techniques to convey the data gathered from the tweets. For some of the data returned, such as the most-used emojis, words, hashtags and user tags within the tweet set, visualisation was easy. These most-used parameters could be displayed as simple lists.
Clearly showing the sentiment, political leaning and tweet emotions graphically were more difficult features to implement.

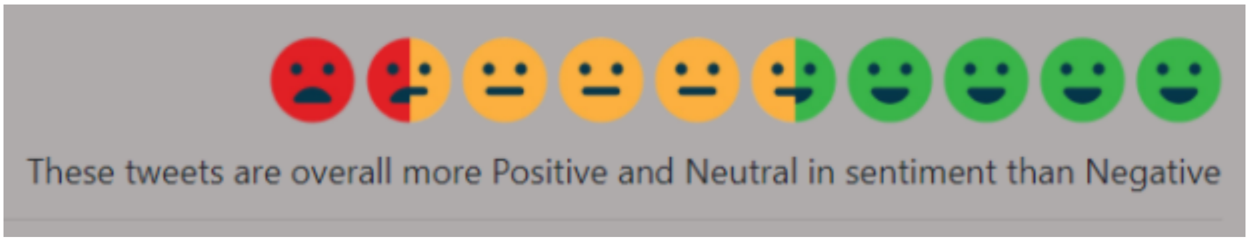
Sentiment Visualisation

The sentiment of a set of tweets is returned as 3 decimal values, the proportion of the tweets categorised as positive, the proportion of the tweets categorised as negative and the proportion of the tweets categorised as neutral in sentiment. These values could be analysed to return a text description of the tweet set sentiment, however a visual representation would add to the visual interest of the application and make the data easier to interpret.
To represent these 3 decimal values, which would always add up to a value of 1, the visualisation approach taken was to represent the sentiments with a total of 10 emojis. The numbers of the relevant sentiment emojis would represent the proportion of the tweet set categorised as that respective emotion.
This set of emojis was initially represented as shown below, with the text sentiment description below for added clarity:

Figure 27: V1 of sentiment Visualisation

This implementation had some issues, however. Representing tweet sentiment with 10 single emojis sometimes oversimplified the data and didn't make it clear enough. To add more detail and nuance to this visualization, partial emojis were added at the boundary between negative/neutral and neutral/positive. The messaging displayed under these sentiment emojis was also updated to be more descriptive.

Figure 28: V2 of sentiment Visualisation (Final Version) - different set of tweets



Political Leaning Visualisation

Political Leaning was an interesting piece of data to visually represent, as it was important to be able to visually differentiate the degrees to which different sets of tweets aligned with a political classification. Since political leaning can be easily understood as a spectrum from left leaning to right leaning, the decision was made to represent the political leaning of a set of tweets as an arrow on a gauge from left leaning to right leaning, in combination with a text description of the political leaning for the sake of clarity and consistency.

Figure 29: Political Leaning Visualisation (Left Leaning)

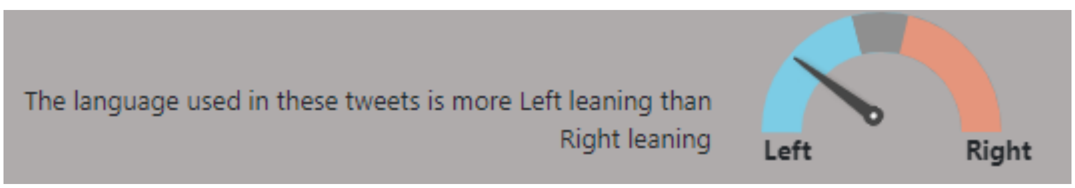


Figure 30: Political Leaning Visualisation (Centrist)

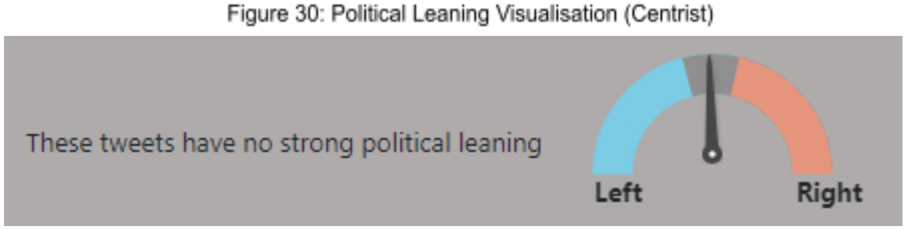
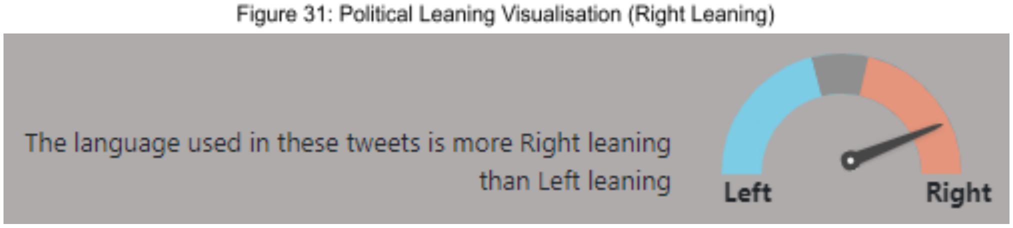


Figure 31: Political Leaning Visualisation (Right Leaning)



Tweet Emotions Visualisation

The emotions detectable within a set of tweets are anger, anticipation, disgust, fear, joy, sadness, surprise and trust. This was too large and complex a set to use emoji representation, however simply stating the most prevalent emotions within the text set oversimplified the data.

The strategy applied to display emotions represented within a set of tweets was to show the proportions of detected emotions on a single bar, with data labels to make the emotion proportions clear.

These labels are represented in a legend below the bar on mobile as the labels were too constricted on the bar when shown on a smaller screen.

Figure 32: Emotion Visualisation Example (Desktop)

Figure 33: Emotion Visualisation Example (Mobile)

3.5.4 Final Version of Graphical User Interface

After much development and many adjustments, these screenshots show the Graphical User Interface of the final version of this project application.

Home Page

http://web1.cs.nuigalway.ie:8081/

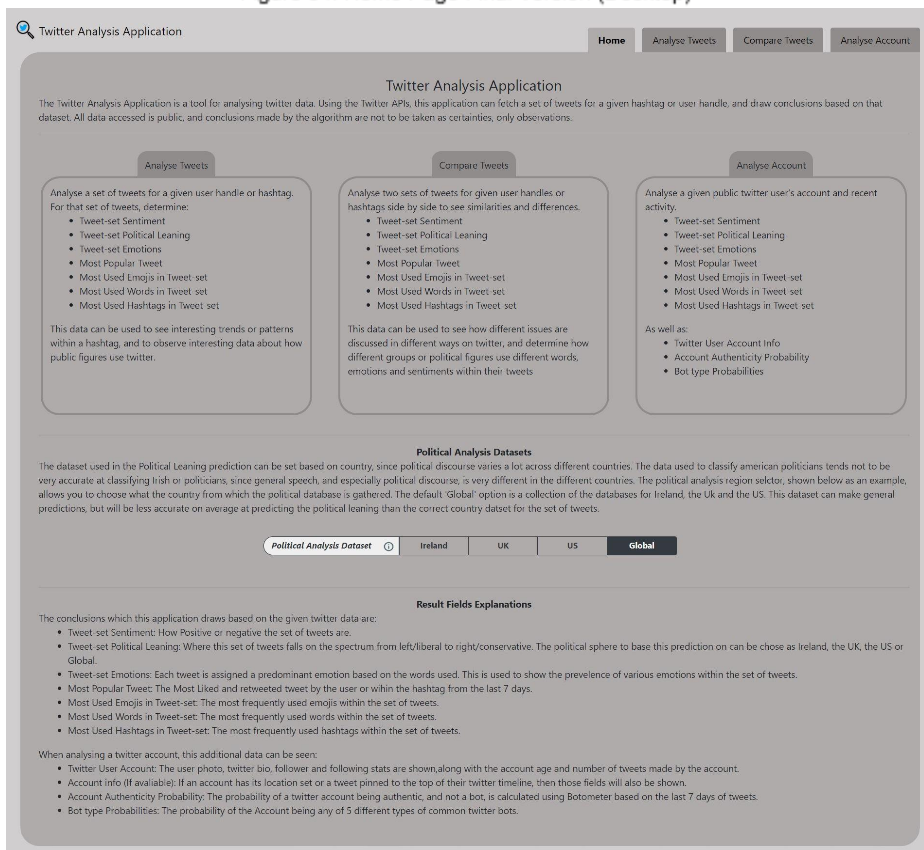


Figure 34: Home Page Final Version (Desktop)

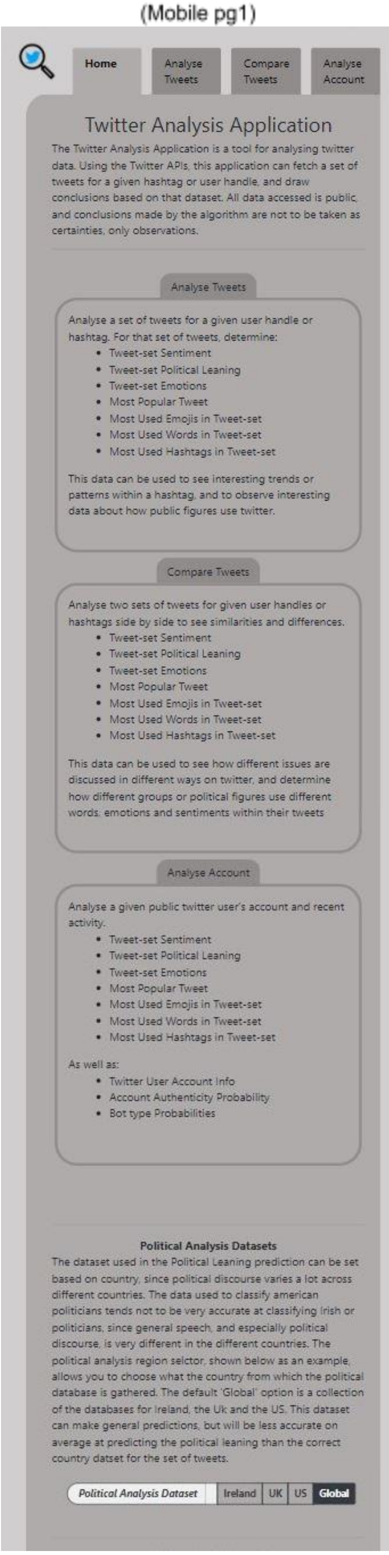


Figure 35: Home Page Final Version (Mobile pg1)

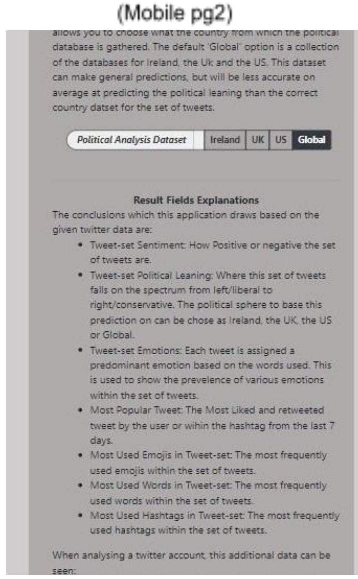


Figure 36: Home Page Final Version (Mobile pg2)

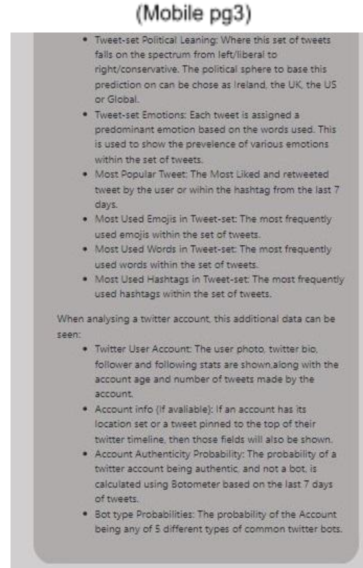


Figure 37: Home Page Final Version (Mobile pg3)

Analyse Tweets Page

http://web1.cs.nuigalway.ie:8081/analyse_tweets

Figure 38: Analyse Tweets Page Final Version (Desktop)

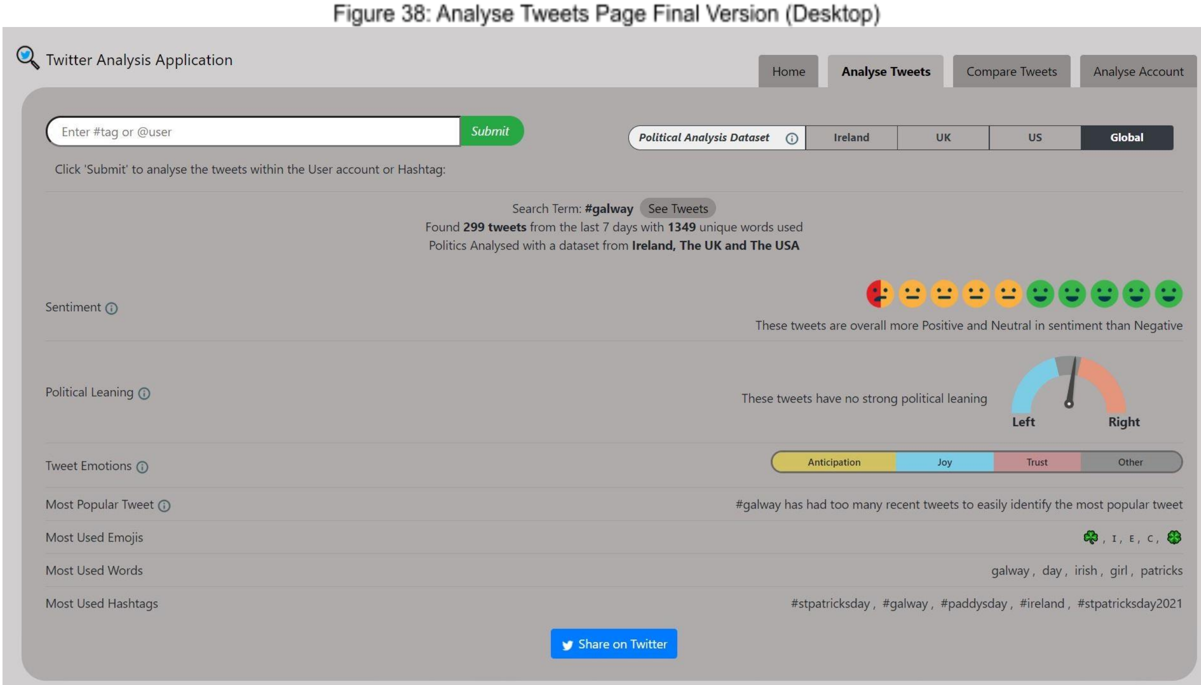


Figure 39: Analyse Tweets Page Final Version (Mobile pg1)

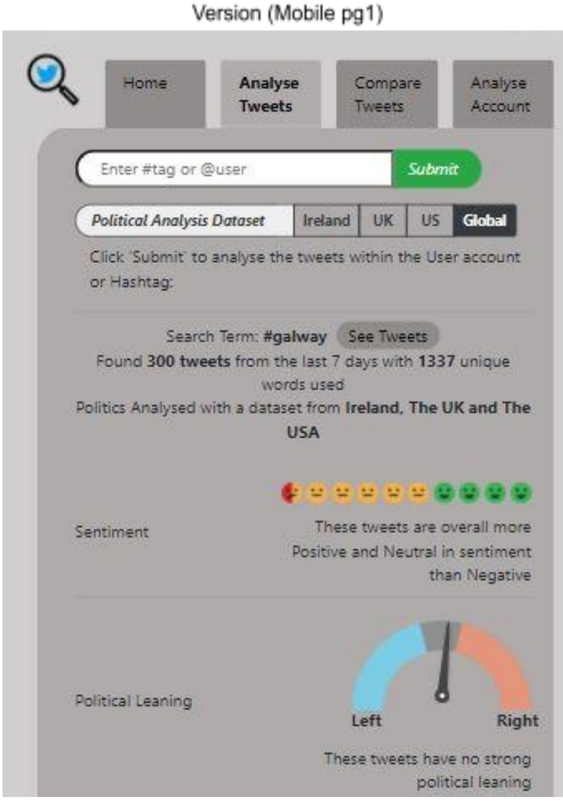
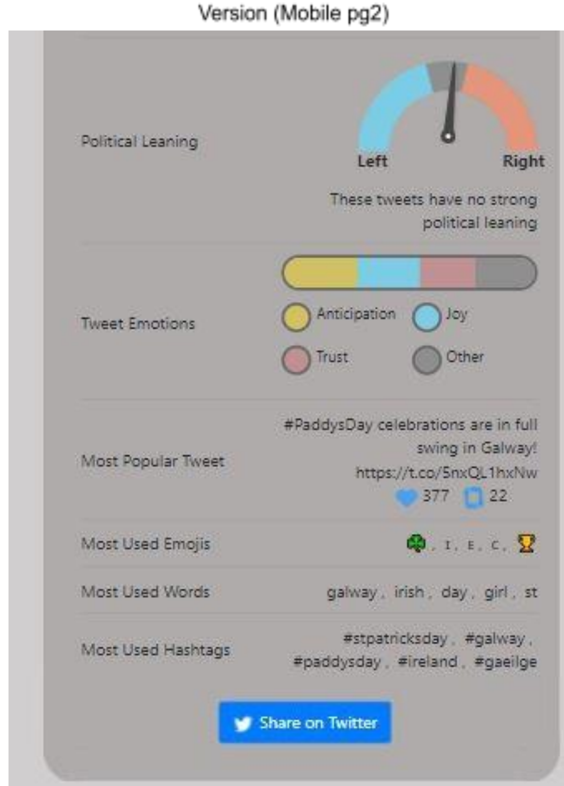


Figure 40: Analyse Tweets Page Final Version (Mobile pg2)



Compare Tweets Page

http://web1.cs.nuigalway.ie:8081/compare_tweets

Figure 41: Compare Tweets Page Final Version (Desktop)

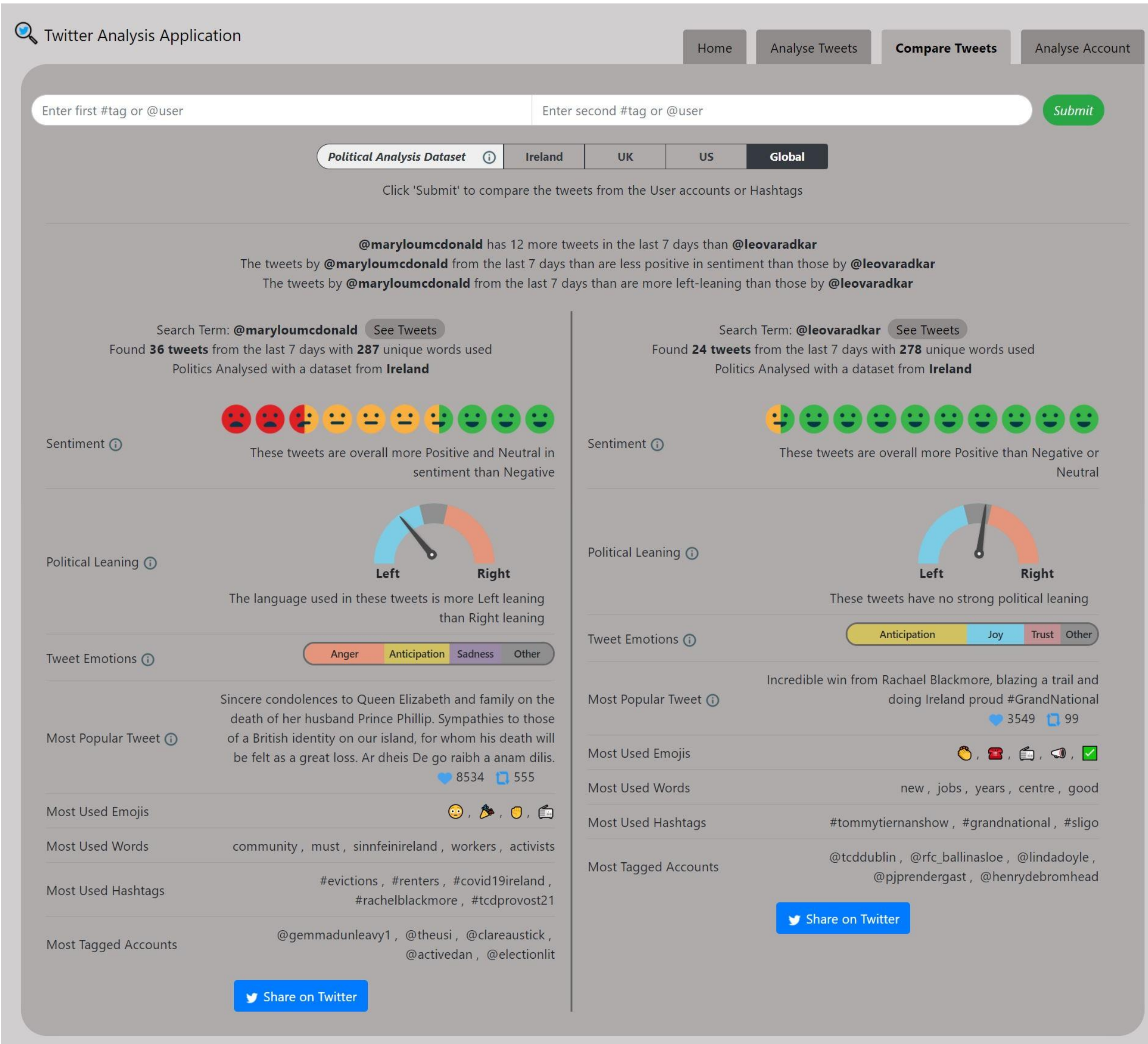


Figure 42: Compare Tweets Page
Final Version (Mobile pg1)

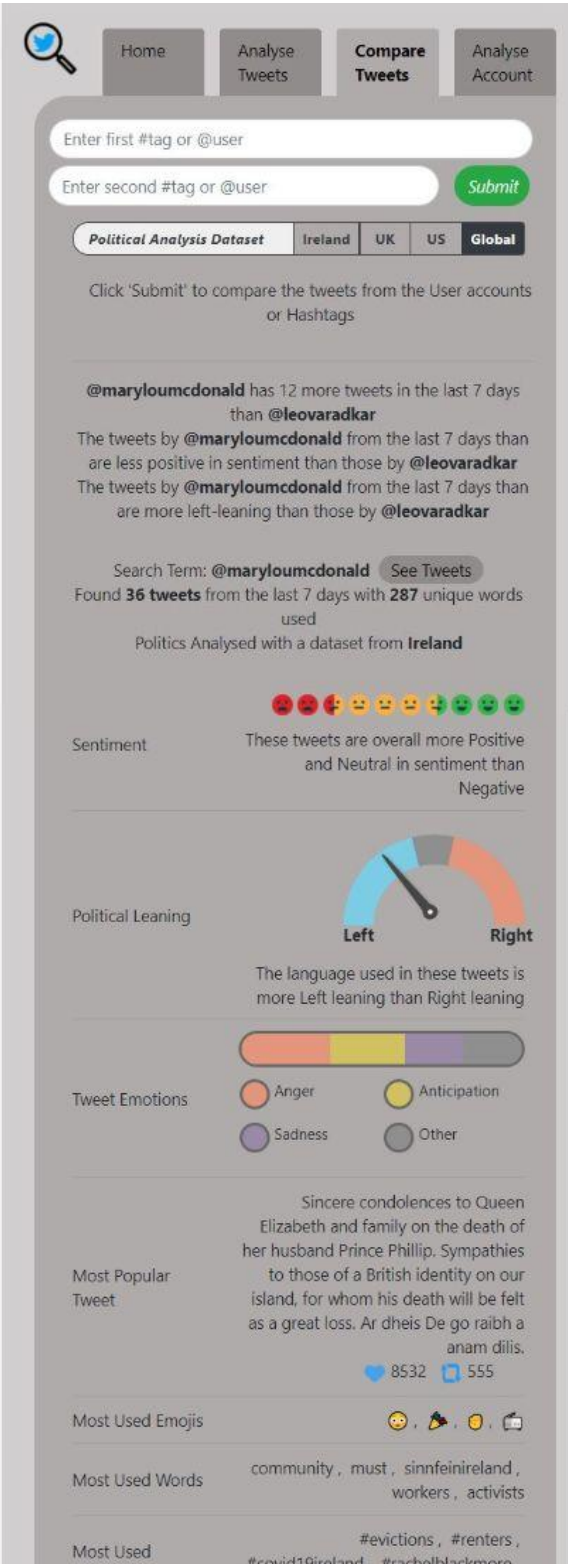
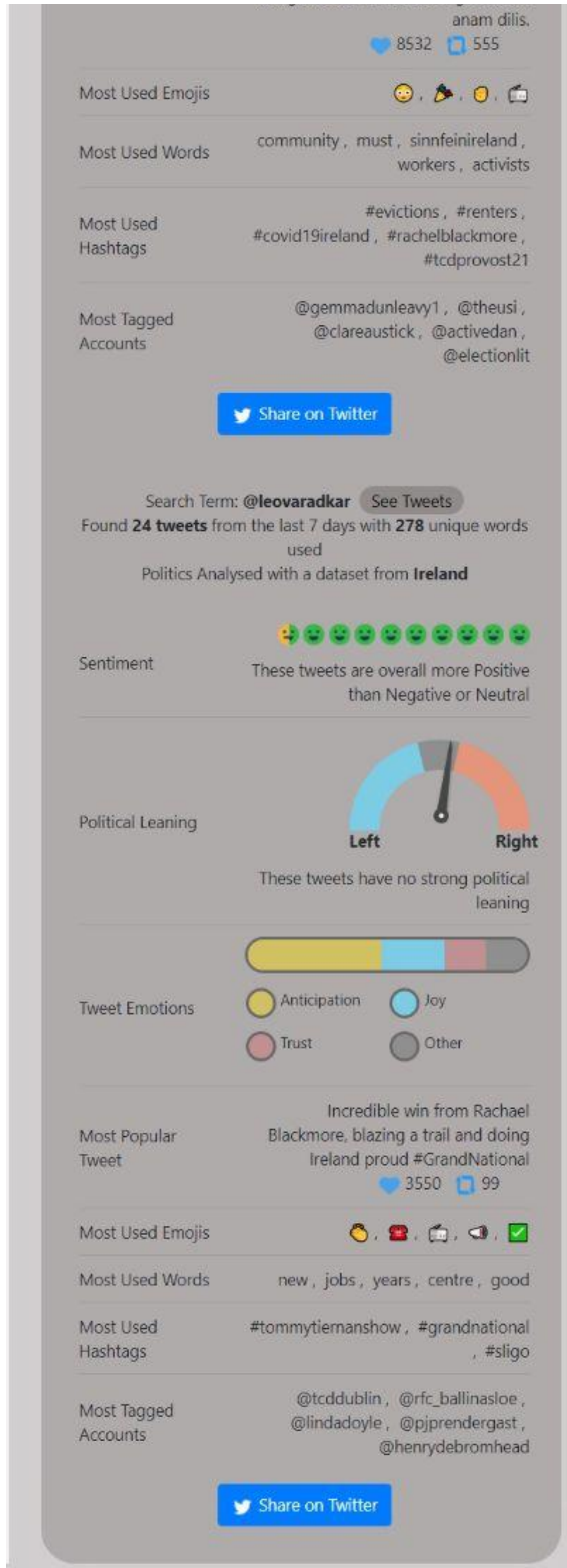


Figure 43: Compare Tweets Page
Final Version (Mobile pg2)



Analyse Account Page

http://web1.cs.nuigalway.ie:8081/analyse_account

Figure 44: Analyse Account Page Final Version (Desktop)

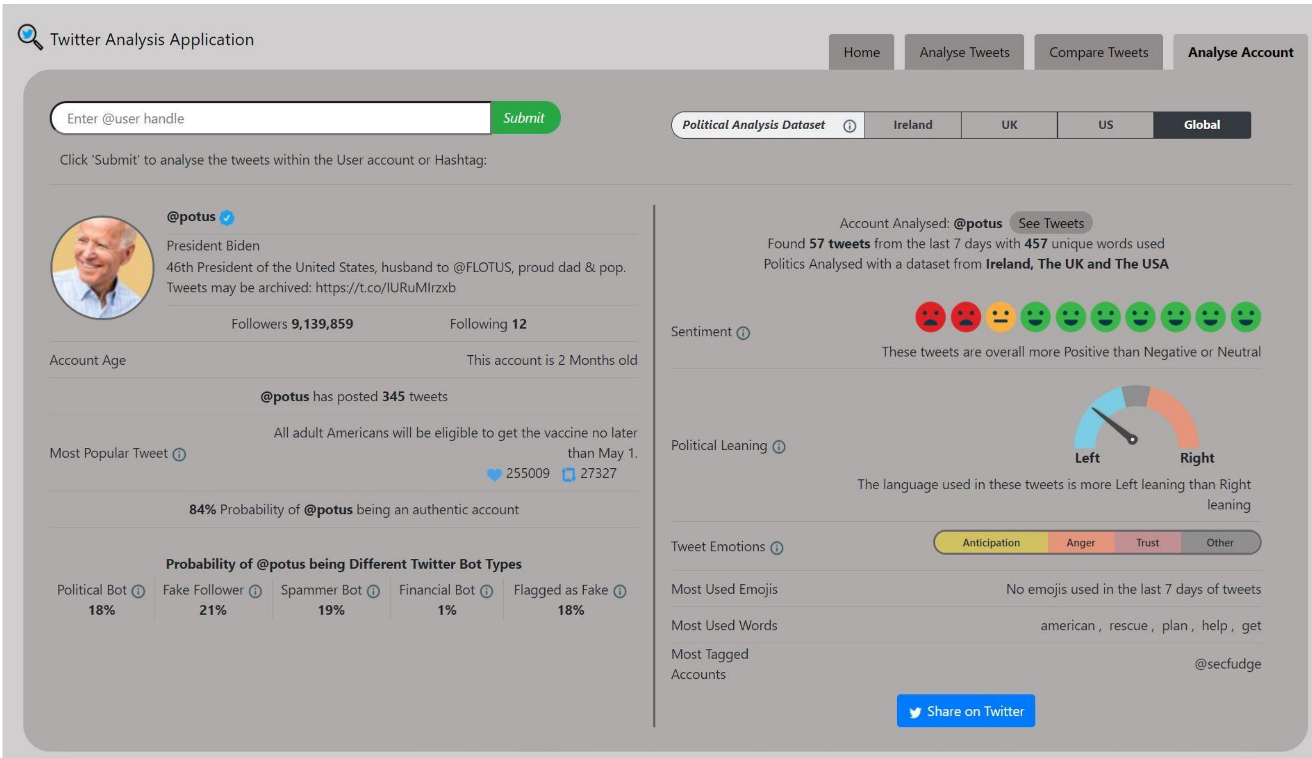


Figure 45: Compare Tweets Page Final Version (Mobile pg1)

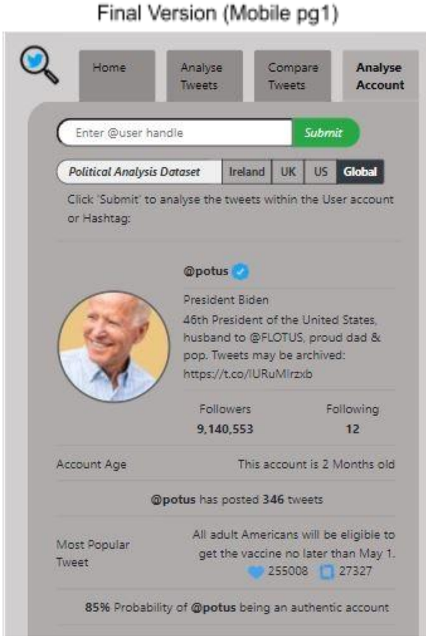


Figure 46: Compare Tweets Page Final Version (Mobile pg2)

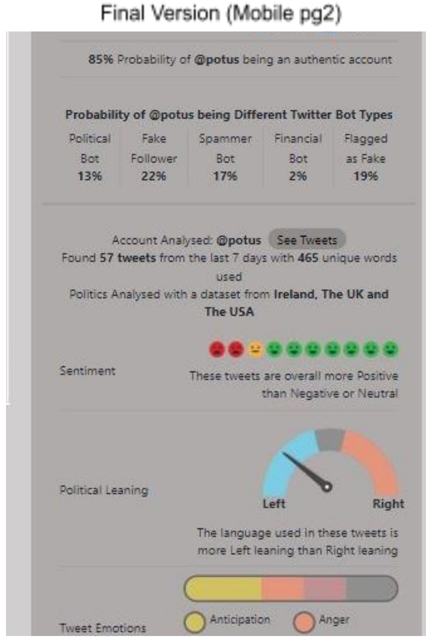
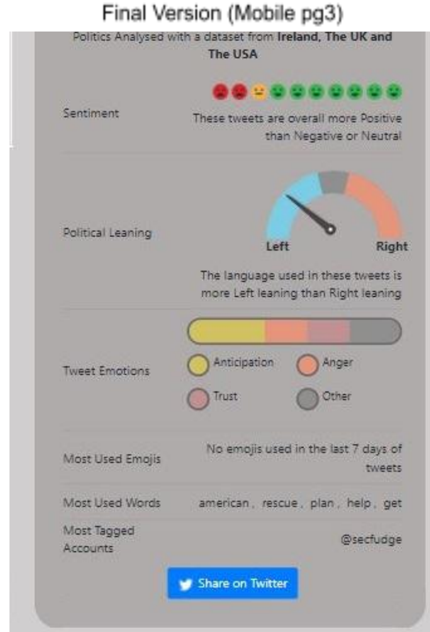


Figure 47: Compare Tweets Page Final Version (Mobile pg3)



4 Development Processes

This project was completed over the course of 8 months, from October 2020 to May 2021. To ensure that the code remained fully functional, and no work would be lost, the project was undertaken using CICD (Continuous Integration, Continuous Development) processes.
This involved setting up a version control system, using github, on which a working version of the project was stored at all times. This repository was then protected using unit tests, so that no new changes could be implemented into the stable project version without first passing a set of defined tests. This system was integral to ensuring that all aspects of the code remained functional throughout the development process.

4.1 Version Control

The project was maintained using Github [18] as a version control system.
The code is accessible at the url https://github.com/a-mcloughlin/final-year-project.
Consistent code quality and stability was ensured through following a few self-imposed rules when making changes to the code in this repository. These rules are:

1. No changes can be merged directly to the main project branch. All changes must be merged as ‘Pull Requests’.
2. A Pull Request must explain exactly what the code change does, and include screenshots of any changes made to the user interface.
3. A Pull Request should only include a single change. Multiple changes in the same Pull Request leads to more confusing and convoluted logs.
4. Pull Requests cannot be merged unless they have passed all unit tests.
5. Any new function added to the code should not be added without unit tests to test the new functionality.

By following these steps, the Github Version Control software allowed this project to progress without any losses of code or significant bugs over the course of the 8 month development cycle

The structure of the github repository can be seen in Figure 48.

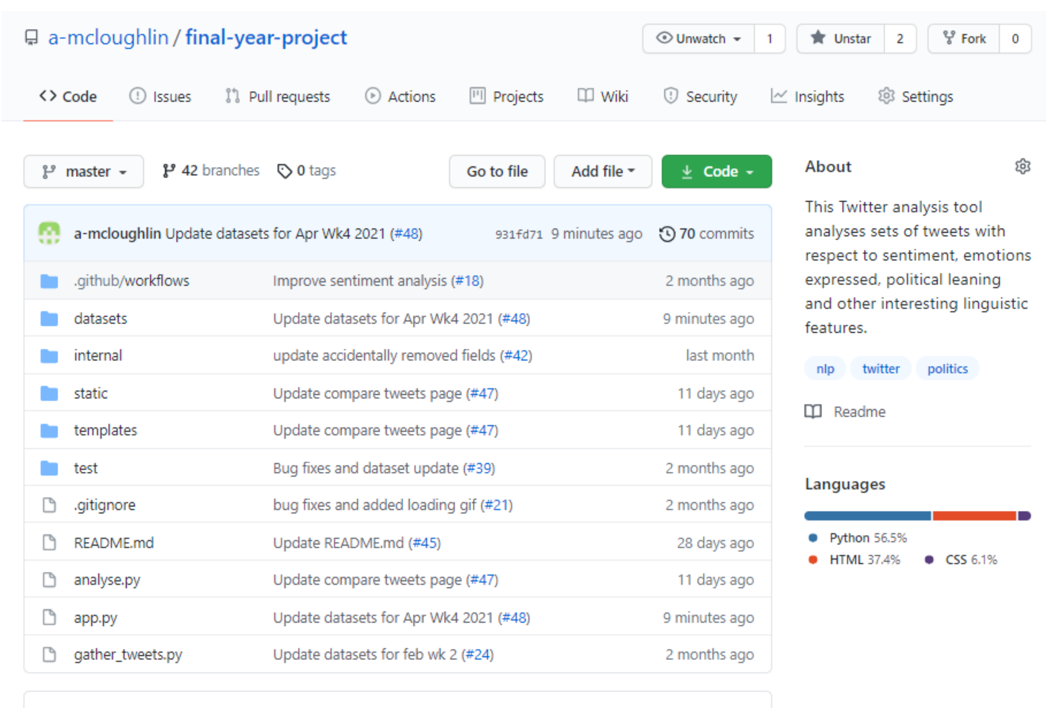


Figure 48: Screenshot of Github Repository Structure

File Structure

The file structure of the project can be seen in the tree graph below. This graph was generated using the MS-DOS ‘tree’ command. It is identical to the file structure of the github repository, with the addition of the auth.yml file used to store API token, which was not pushed to the github repository for security reasons.

```
├── .github
│   └── workflows
│       └── python–test.yml
│
├── datasets
│   ├── NRC-Emotion-Lexicon-Wordlevel-v0.92.txt
│   │
│   ├── ml_models
│   │   ├── global.joblib
│   │   ├── ie.joblib
│   │   ├── uk.joblib
│   │   └── us.joblib
│   │
│   └── ml_training
│       ├── ie_parties_full_set.csv
│       ├── ie_uk_us_full_set.csv
│       ├── uk_parties_full_set.csv
│       └── us_parties_full_set.csv
│
├── internal
│   ├── data_analysis
│   │   ├── analyse_sentiment_emotions.py
│   │   ├── detect_bot_account.py
│   │   ├── detect_emotions.py
│   │   └── detect_sentiment.py
│   │
│   ├── political_leaning
│   │   ├── analyse_political_leaning.py
│   │   └── political_leaning_ml.py
│   │
│   ├── twitter
│   │   ├── auth.py
│   │   └── requests.py
│   │
│   └── word_processing
│       ├── handle_wordlist.py
│       ├── interpret_data.py
│       └── process_json_tweets.py
│
├── static
│   ├── images
│   │   ├── fyplogo.png
│   │   ├── happy.png
│   │   ├── info_icon.png
│   │   ├── loading-gif.gif
│   │   ├── meter-gauge-needle.svg
│   │   ├── neutral.png
│   │   ├── neut_happy.png
│   │   ├── retweet.png
│   │   ├── sad.png
│   │   ├── sad_neut.png
│   │   ├── sharebtn.png
│   │   ├── twitter_like.png
│   │   ├── twitter_logo.png
│   │   └── twitter_verified_tick.png
│   │
│   └── styles
│       └── styles.css
│
├── templates
│   ├── base.html
│   │
│   └── tabs
│       ├── analyse_account.html
│       ├── analyse_tweets.html
│       ├── compare_tweets.html
│       └── home.html
│
├── test
│   ├── install_nltk.py
│   ├── json_tweetset.json
│   ├── json_userdata.json
│   ├── mocked_auth.yaml
│   ├── mocked_data.py
│   ├── __init__.py
│   │
│   ├── data_analysis
│   │   ├── test_data_analysis.py
│   │   └── __init__.py
│   │
│   ├── twitter
│   │   ├── test_twitter.py
│   │   └── __init__.py
│   │
│   └── word_processing
│       ├── test_interpret_data.py
│       ├── test_word_processing.py
│       └── __init__.py
│
├── .gitignore
├── analyse.py
├── app.py
├── auth.yaml
├── gather_tweets.py
└── README.md
```

4.2 Testing

Automated unit testing was integral to catching any development bugs early on and to maintaining a consistent and fully functional project.
Unit testing in this project was performed using Python Unittest, Pytest and using Github's Actions tool.

Each section of Python code has a corresponding Unittest test file with its own set of custom unit tests. These tests are run to individually test each method within the code, to ensure that each method performs as expected. The set of all unittest tests can then be run as a single pytest test. If any one unittest test fails, then the test set will fail, flagging the error. This makes it easier to keep code functional, and keep all the tests consistently up to date.

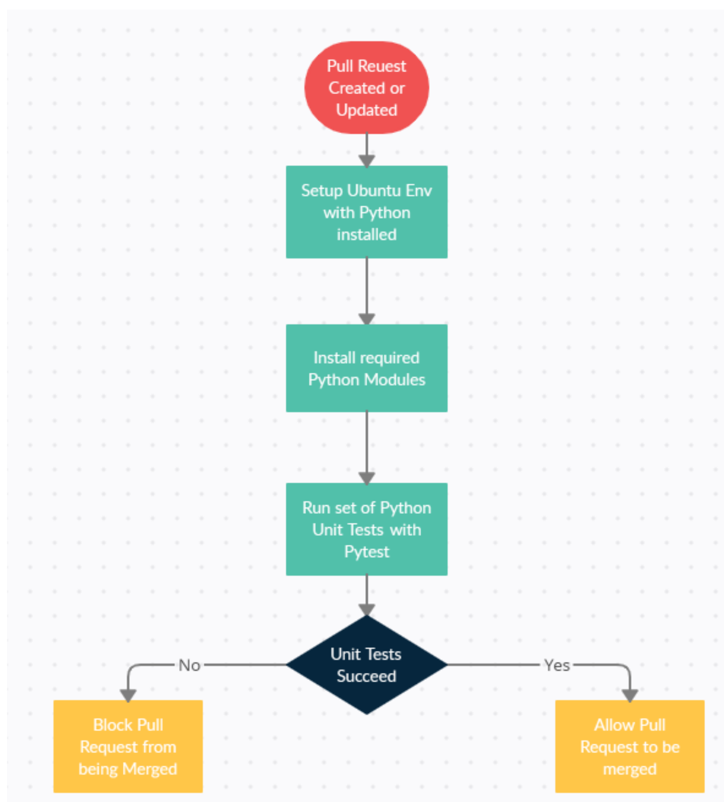
This process was then automated on github using the Github 'Actions' tool.
Github Actions allow for processes to be defined and automated so that the processes are performed in response to certain actions taken by a user on github, such as pushing a commit to a branch. Github Actions are defined using yaml files, which define the action to trigger the process, the environment to perform the process in and the process to perform.

This project required tests to be triggered by pull requests to the main branch being opened or updated. These tests being triggered must run on a virtual machine with the Python language and all the required Python modules installed. This required a two step process with both a build and a test stage. In the build stage, the environment was set up to enable all tests to run, and in the test stage, the Python Unit Tests were run using Pytest, with any failures clearly logged. To ensure consistency across the development machine and the automated testing virtual machine, specific versions were defined for all installed components and Python modules.

The Github Action setup was defined such that a Pull Request cannot be merged if the Pytest test failed, forcing the underlying issue to be identified and resolved before the change can be merged into the stable project version.

The testing process is illustrated in Figure 49.

Figure 49: Automated Testing Process Flow Diagram



5 Application Evaluation

5.1 Algorithm Performances

Once the planned application development had been completed, it was important to evaluate just how effectively the application analysed various aspects of Twitter data. There were known limits and constraints in the implementation of the Sentiment, Emotion and Political Leaning analysis, so it was important to ensure that there were no significant flaws in the system by performing a set of 'sanity checks' on the system, and interpreting the results of these checks.

5.1.1 Sentiment Analysis

A simple sanity check could be performed on the sentiment analysis component of the application using hashtags.

Positive
One would expect the hashtag #happy to be much more positive than negative, and that was the same conclusion reached by the algorithm as shown in below.

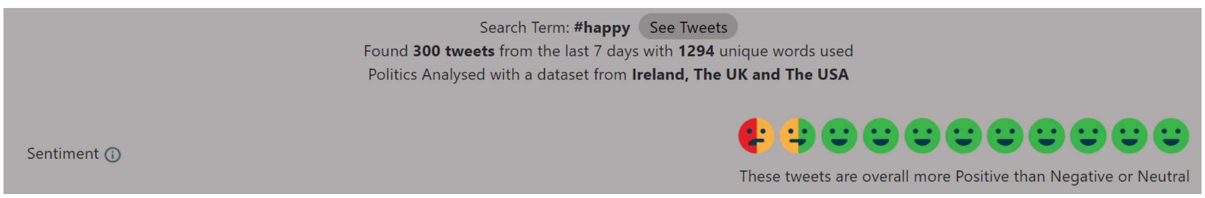


Figure 50: #happy sentiment analysis screenshot

Negative
One would expect the hashtag #sad to be much more negative than positive, which was the same conclusion reached by the algorithm as shown below.

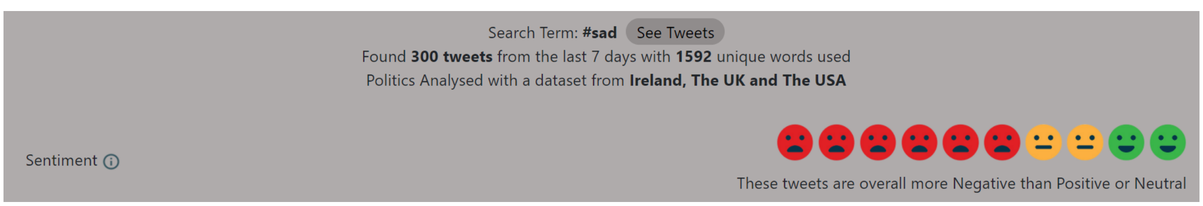


Figure 51: #sad sentiment analysis screenshot

Neutral
Deciding on a word to use to test neutral sentiments was more difficult. The word #neutral is used too frequently in passionate discussion about issues such as international relations, so a word must be chosen which would be used in unemotional, unpassionate conversations only. The hashtag #banana can be expected to be filled with neutral, unemotional posts about fruit and the algorithm supported that expectation by finding #banana to be more neutral than positive or negative, as shown below:

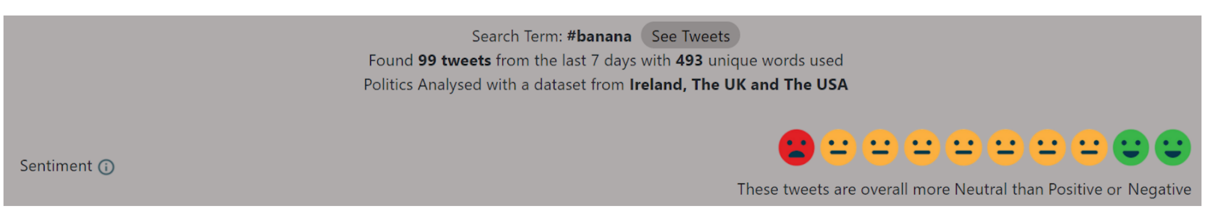


Figure 52: #banana sentiment analysis screenshot

5.1.2 Emotion Analysis

A sanity check of the emotion analysis component of the application could be performed using hashtags, as detailed below.

Anger

Detection of the emotion of 'anger' was checked using the hashtag #angry. As expected, the most predominant emotion within the tweet set fetched for #anger was anger.
A screenshot of this test is shown below

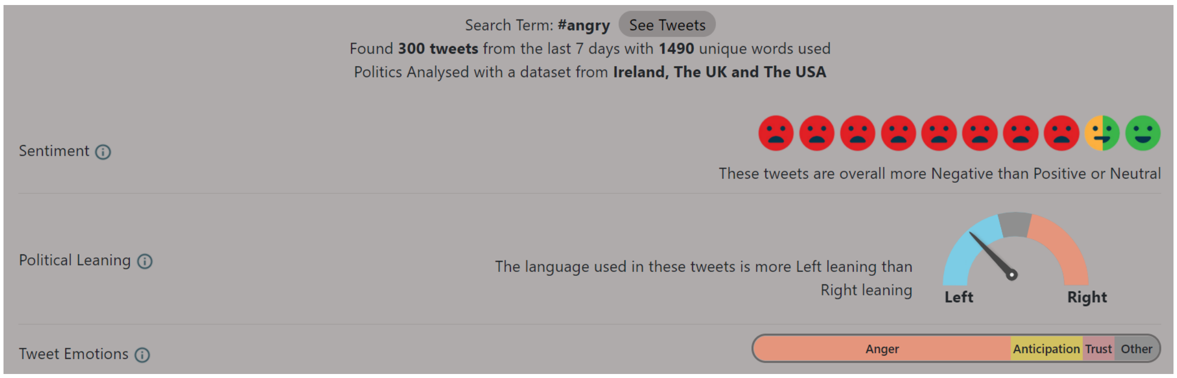


Figure 53: #anger emotion analysis screenshot

Anticipation

Detection of the emotion of 'anticipation' was checked using the hashtag #anticipation. As expected, the most predominant emotion within the tweet set fetched for #anticipation was anticipation.
A screenshot of this test is shown below

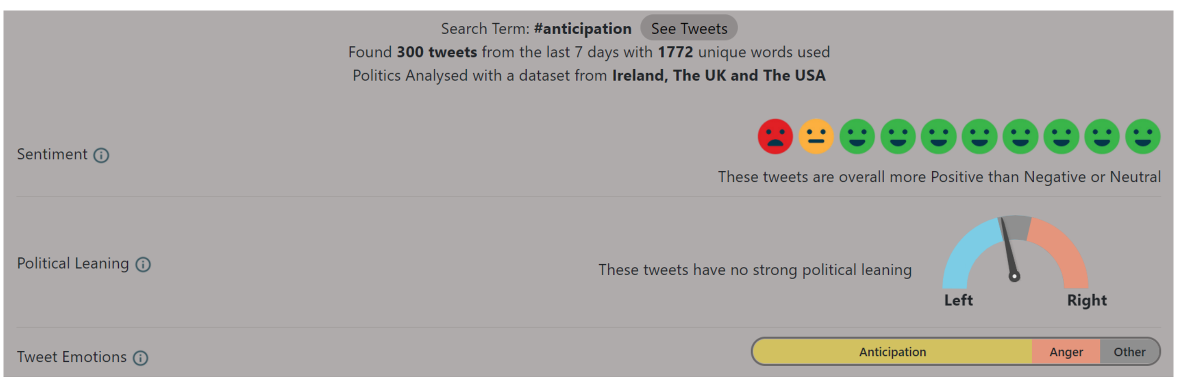


Figure 54: #anticipation emotion analysis screenshot

Disgust
Detection of the emotion of 'disgust' was initially checked using the hashtag #disgust.
Surprisingly, the most predominant emotion detected in #disgust was anger, and not disgust. This is likely due to the use of the words disgusted and disgusting in online discourse to express anger at an action taken or a position held by others.
A screenshot of this test is shown below

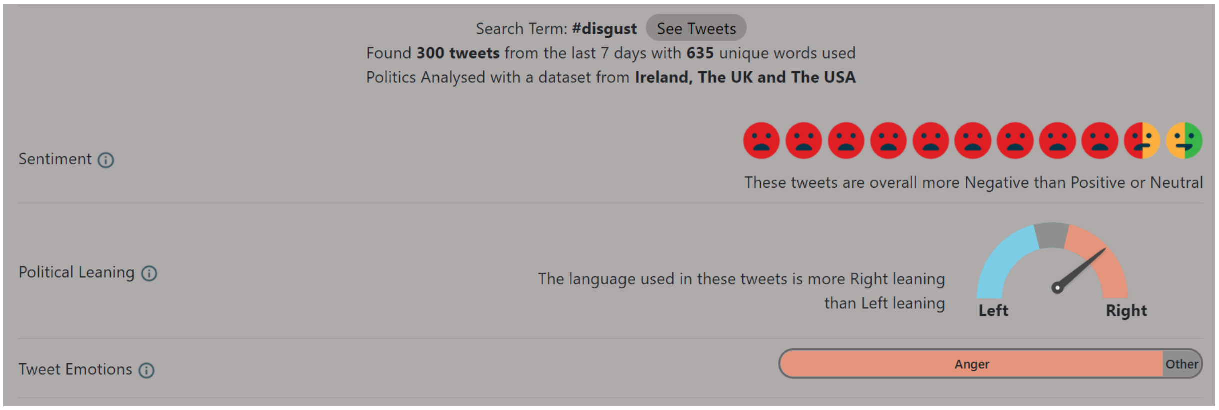


Figure 55: #disgust emotion analysis screenshot

To check that the emotion detection for the actual emotion of disgust was working, a hashtag had to be found that would be used online to express the emotion of disgust. After some simple research, the hashtag #sick emerged as a hashtag used to express disgust online. When this test was run, the results were as expected. with the most predominant emotion within the tweet set fetched for #sick being disgust.

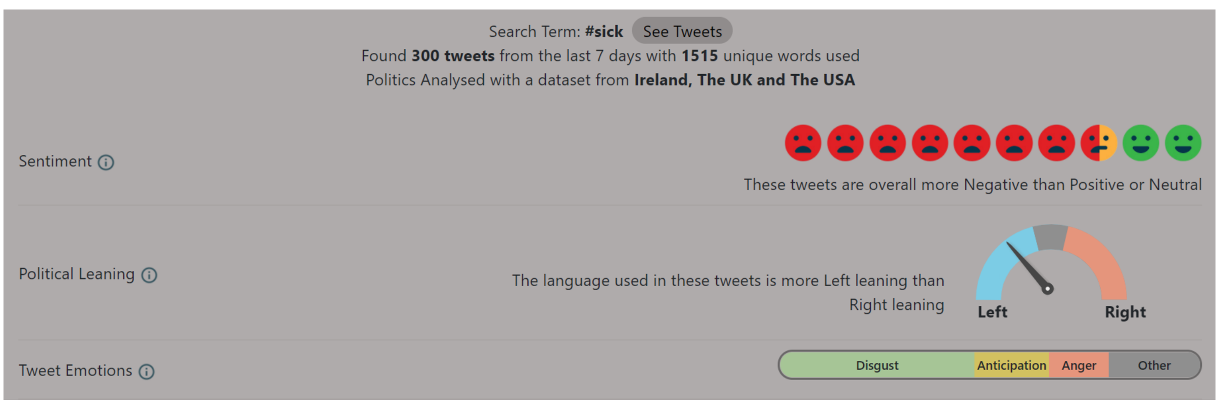


Figure 56: #sick emotion analysis screenshot

Fear

Detection of the emotion of ‘fear’ was checked using the hashtag #fearful. As expected, the most predominant emotion within the tweet set fetched for #fearful was fear.

A screenshot of this test is shown below.

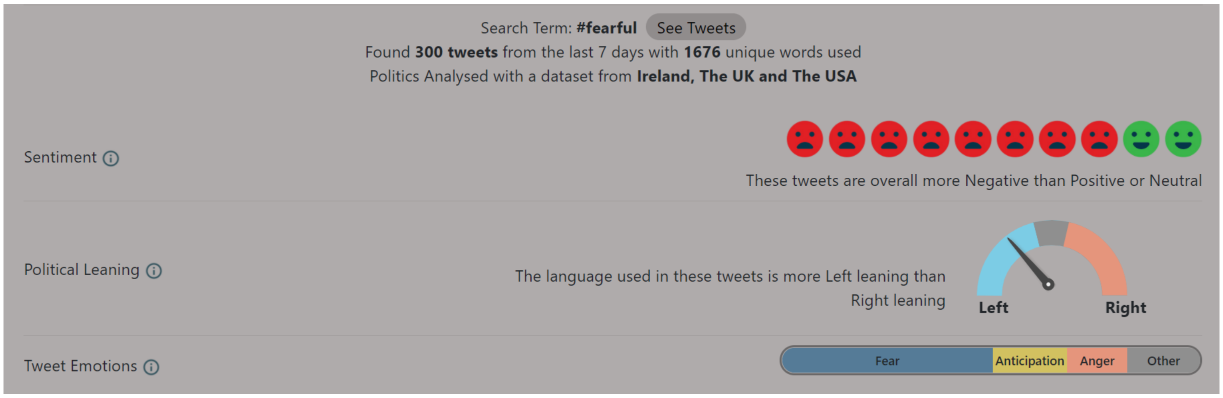


Figure 57: #fearful emotion analysis screenshot

Joy

Detection of the emotion of ‘joy’ was checked using the hashtag #joyful. As expected, the most predominant emotion within the tweet set fetched for #joyful was joy.

A screenshot of this test is shown below.

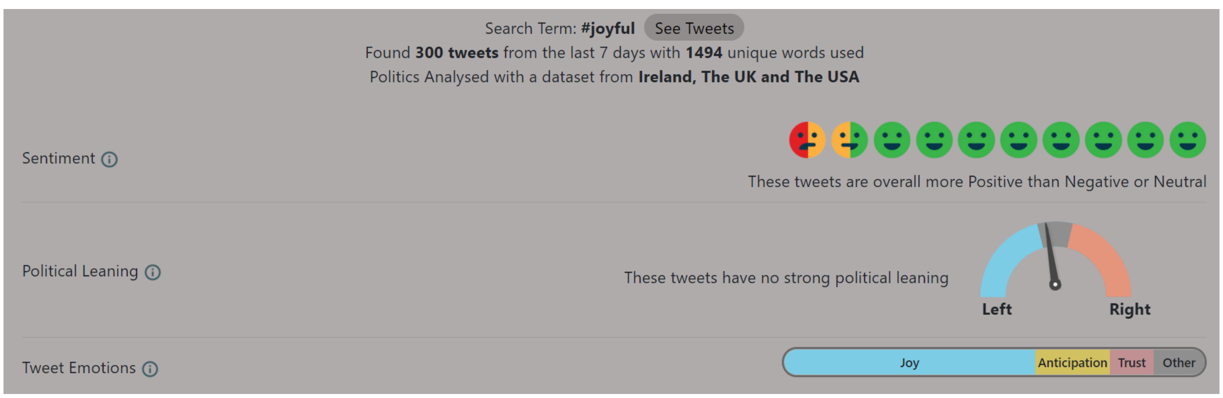


Figure 58: #joyful emotion analysis screenshot

Sadness

Detection of the emotion of ‘sadness’ was checked using the hashtag #sadness. As expected, the most predominant emotion within the tweet set fetched for #sadness was sadness.

A screenshot of this test is shown below.

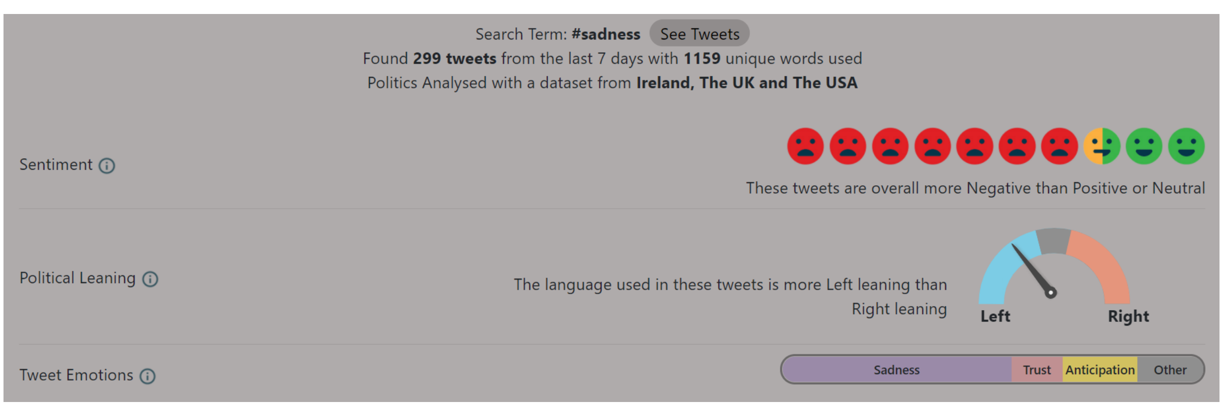


Figure 59: #sadness emotion analysis screenshot

Surprise

Detection of the emotion of ‘Surprise’ was initially checked using the hashtag #surprise. Surprisingly, the most predominant emotion detected in #surprise was fear, and not surprise. A brief investigation into the relevant tweets, it seems that the hashtag #surprise was being used widely as a sarcastic way to criticise decisions that seemed to have been taken hastily, where the consequences of these decisions may not have been sufficiently examined. An example of a tweet from the set of tweets which used #surprise to express a level of fear and anger is shown below in Figure 61. A screenshot of this test is shown below.

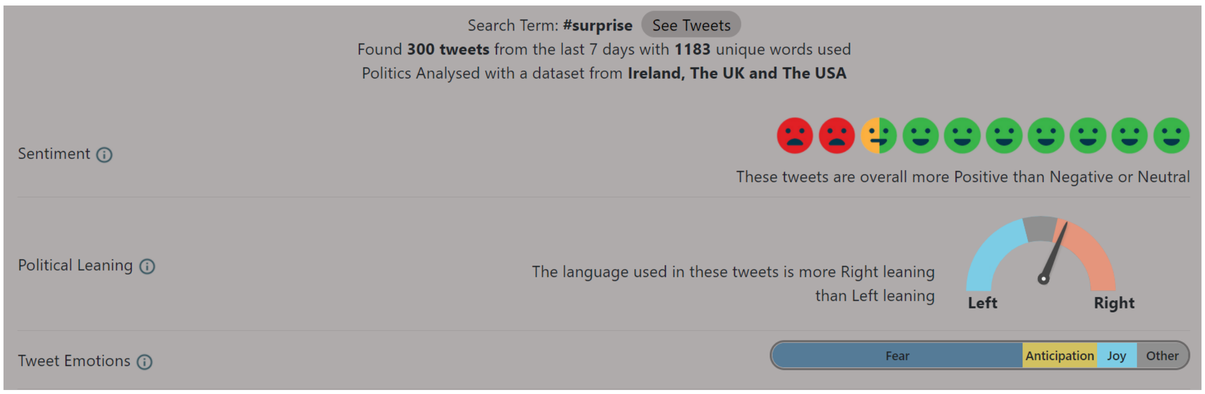


Figure 60: #surprise emotion analysis screenshot

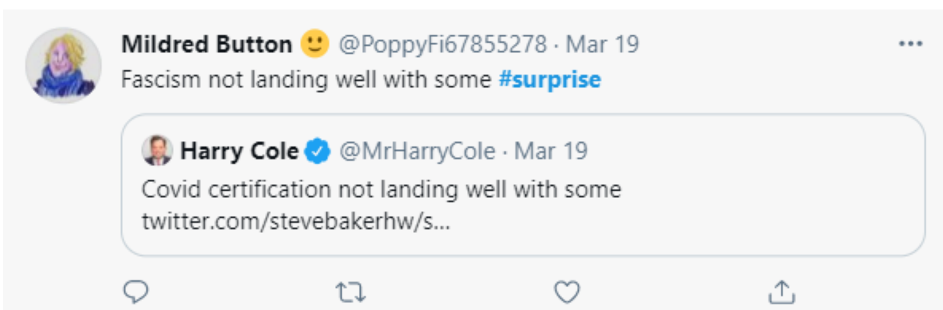


Figure 61: screenshot of sarcastic tweet using #surprise

To check that the emotion detection for the actual emotion of surprise was working, a hashtag had to be found that would be used online to express the emotion of surprise. After some research, the hashtag #amaze emerged as a hashtag used to express surprise online. When this test was run, the results were as expected. with the most predominant emotion within the tweet set fetched for #amaze being surprise.

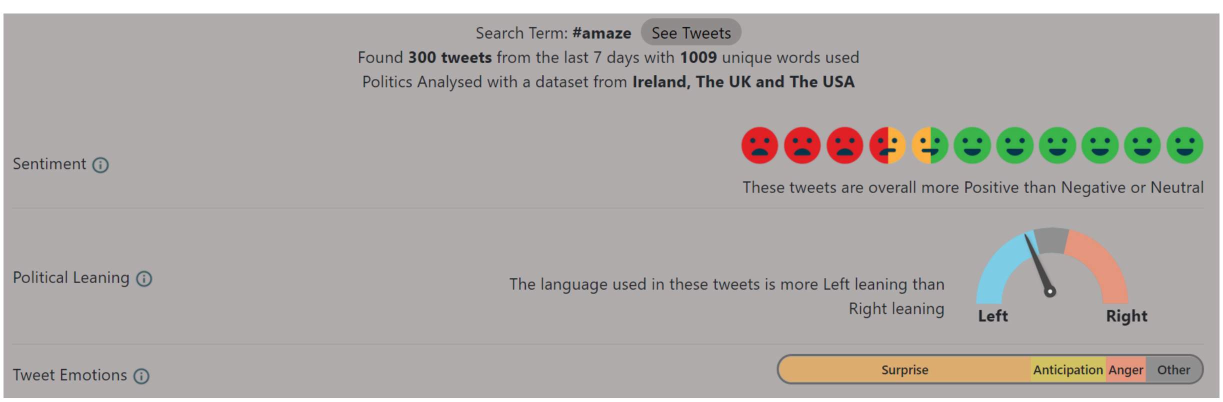


Figure 62: #amaze emotion analysis screenshot

Trust
Detection of the emotion of 'trust' was checked using the hashtag #trust. As expected, the most predominant emotion within the tweet set fetched for #trust was trust.
A screenshot of this test is shown below.

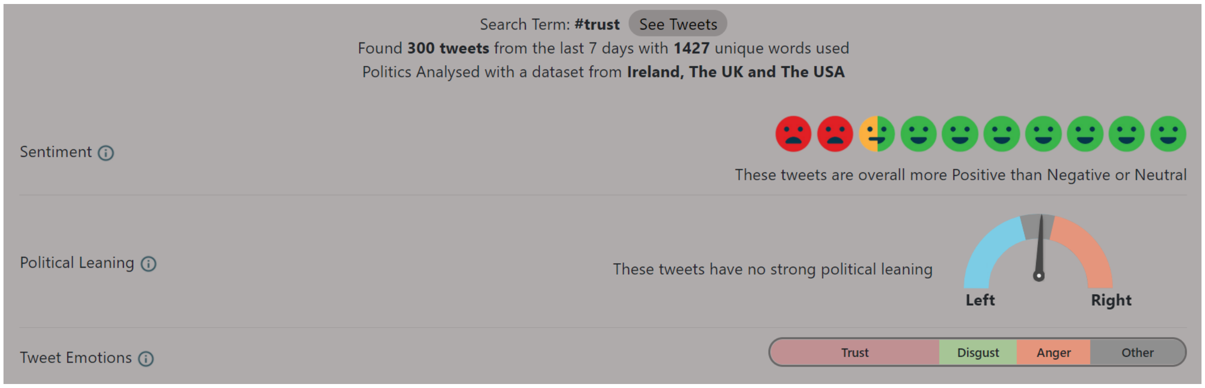


Figure 63: #trust emotion analysis screenshot

5.1.3 Political Leaning Analysis

Evaluation of the performance of the political leaning analysis involves analysing the performance of each of the four datasets on politicians relevant to those datasets.

Irish Dataset
It was important to pick politicians who are representative of their respective political spheres in order to accurately evaluate the success of the political leaning analysis.
Within the Irish political sphere, the obvious politician to represent current left leaning politics is Mary Lou McDonald, the leader of the Sinn Fein party, while the obvious politician to represent current right leaning politics is Micheal Martin, the Taoiseach and leader of Fianna Fail. When performing this evaluation, however, Micheal Martin had tweeted fewer than 20 times in the last week, and thus could not be used as a thorough and representative candidate for evaluation of this system. For that reason, the politician chosen to represent current right leaning Irish politics is Leo Varadkar, the Tanaiste and head of Fine Gael.

Screenshots of these tests are shown in Figure 64 and Figure 65 below.

Figure 64: @maryloumcdonald political leaning analysis screenshot (Irish Dataset)

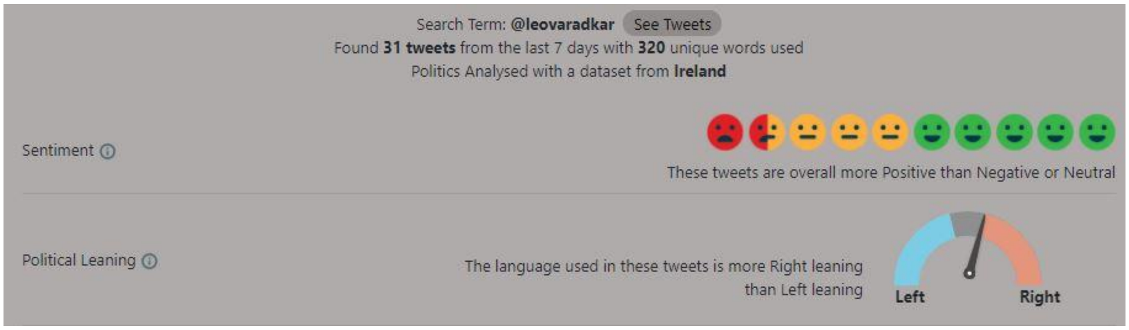


Figure 65: @leovaradkar political leaning analysis screenshot (Irish Dataset)

The tweets made by both MaryLou McDonald (@maryloumcdonald) and Leo Varadkar (@leovaradkar) were correctly classified by this political leaning machine learning model trained with the Irish political dataset.

UK dataset

The obvious politician to use as a representative of current UK left leaning politics is Keir Starmer, the leader of the UK Labour party and the leader of the opposition within the parliament. The obvious politician to use as a representative of current UK right leaning politics is Boris Johnson, the leader of the UK Conservatives party and the Uk prime minister.

Screenshots of these tests are shown in Figure 66 and Figure 67 below.

Figure 66: @keir_starmer political leaning analysis screenshot (UK Dataset)

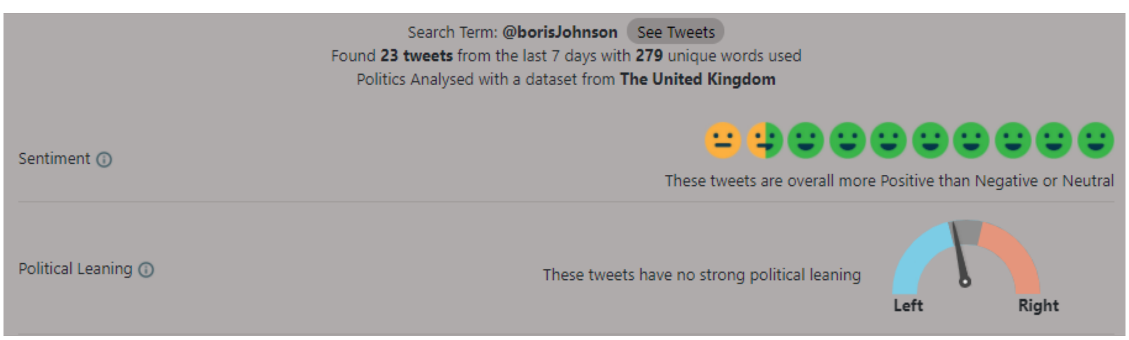


Figure 67: @borisjohnson political leaning analysis screenshot (UK Dataset)

The tweets made by Keir Starmer (@keir_starmer) were correctly classified by this political leaning machine learning model trained with the UK political dataset, however the tweets made by Boris Johnson (@borisjohnson) were incorrectly classified as having no strong political leaning. This inaccuracy can likely be attributed to the fact that Boris Johnson had posted fewer tweets within the previous 7 days than Keir Starmer, only 23 tweets compared to 60 by Keir Starmer. Another anomaly that could have led to this unexpected result could be that it may have been an unusual week for Boris Johnson, which may be unrepresentative of his typical Twitter use. Overall, this imperfect result is indicative of the fact that no classification algorithm can be 100% correct.

US Dataset

The obvious politician to use as a representative of current US left leaning politics is Joe Biden, the leader of the Democratic party and the president of the USA. The obvious politician to use as a representative of current US right leaning politics would be Donald Trump, the leader of the Republican party and the former president, however this is not possible as Donald Trump has been permanently banned from Twitter. The next most obvious conservative US politician who uses Twitter is Ted Cruz, an ally of Donald Trump and the US senator for the conservative stronghold state of Texas.

Screenshots of these tests are shown in Figure 68 and Figure 69 below.

Figure 68: @potus political leaning analysis screenshot (USA Dataset)

Figure 69: @tedcruz political leaning analysis screenshot (USA Dataset)

The tweets made by both Joe Biden (@potus) and Ted Cruz (@tedcruz) were correctly classified by this political leaning machine learning model trained with the US political dataset.

Global Dataset
In order to accurately evaluate the performance of the global dataset across political regions, the global dataset was used to train a model to classify all 6 politicians who were previously classified with their relevant national datasets.

The liberal politicians are Mary Lou McDonald, Keir Starmer and Joe Biden.
Screenshots of these tests are shown in Figure 70, Figure 71 and Figure 72 below.

Figure 70: @maryloumcdonald political leaning analysis screenshot (Global Dataset)

Figure 71: @keir_starmer political leaning analysis screenshot (Global Dataset)

Figure 72: @potus political leaning analysis screenshot (Global Dataset)

The tweets made by all three global left leaning politicians @maryloumcdonald, @keir_starmer and @potus were correctly classified as left leaning by this political leaning machine learning model trained with the Global political dataset.
It could be argued that this dataset is incorrect in placing these politicians at a similar place on the political spectrum however, as Joe Biden is quite centrist even within the conservative country of the USA, and Mary Lou McDonald is widely considered to be strongly left-leaning in the more liberal sphere of Irish politics. While the generalised classifications are all correct, it oversimplifies global politics to place these three politicians into the same category, or to try to quantify them on the same spectrum. The global dataset is quite accurate at rough classifications, but there is much more in depth analysis and nuance to be seen when using the national dataset specific to the political sphere being analysed.

The conservative politicians are Leo Varadkar, Boris Johnson and Ted Cruz.
Screenshots of these tests are shown in Figure 73, Figure 74 and Figure 75 below.

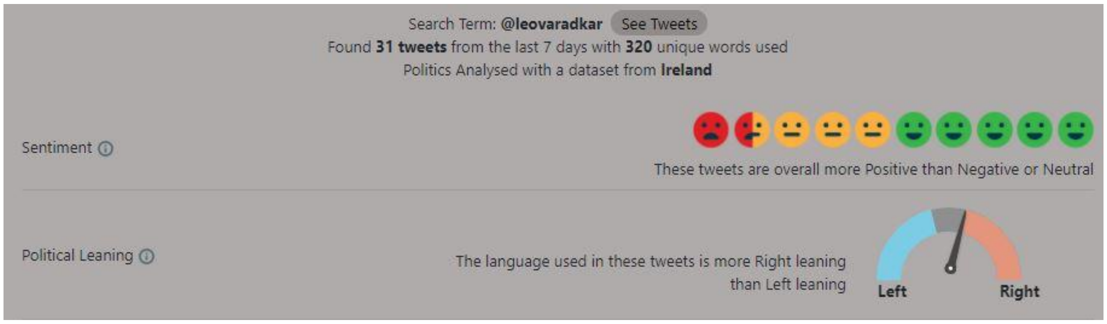


Figure 73: @leovaradkar political leaning analysis screenshot (Global Dataset)

Figure 74: @borisjohnson political leaning analysis screenshot (Global Dataset)

Figure 75: @tedcruz political leaning analysis screenshot (Global Dataset)

The tweets made by all three global right leaning politicians @leoVaradker, @borisjohnson and @tedcrux were correctly classified as right leaning by this political leaning machine learning model trained with the Global political dataset.

Complete Test Results

Evaluation the performance of the political leaning analysis was much more complicated than evaluating the sentiment or emotion detection systems. For this reason, a more thorough evaluation was performed on this component. The political leaning of 18 politicians was fetched using each of the 4 datasets. These 18 politicians consisted of 3 liberal and 3 conservative politicians from each of the relevant countries, those being Ireland, the UK and the USA. Only politicians who had tweeted at least 20 times in the preceding 7 days were analysed to avoid inaccuracies in the classifications of these small sets of tweets.

These results are shown in figure 76 below.
For each politician, the table shows the actual political leaning of the politician as either liberal (‘lib') or

conservative (‘con’), the country where this politician is active, the political Party that the politician belongs to and their Twitter user handle. The raw political leaning value is then shown as calculated by each of the four dataset options. This political leaning value will be calculated in the range -1 to 1, with -1 being strongly liberal and 1 being strongly conservative. The strength of these political leanings are shown using colour intensity, with a concentrated blue colour showing a strongly left result and a concentrated red showing a strongly right result.
For example, Joe Biden is included in the table as a liberal politician from the United States, a member of the democrat political party with the Twitter handle @potus. He was categorised as Conservative by the irish dataset and liberal by each of the UK , US and global datasets.

Figure 76: Political Analysis Evaluation Results

Leaning	Country	Politician	Party	Twitter Handle	Ireland	UK	US	Global
lib	IE	Mary Lou McDonald	Sinn Fein)	@maryloumcdonald	**-0.29**	-0.41	0.29	-0.37
lib	IE	Jen Whitmore	Social Democrats	@whitmorejen	**-0.20**	-0.56	0.07	-0.47
lib	IE	Alan Kelly	Labour	@alankellylabour	**-0.15**	-0.24	0.14	-0.33
lib	UK	Keir Starmer	Labour	@keir_starmer	0.13	**-0.70**	-0.43	-0.43
lib	UK	Angela Rayner	Labour	@angelarayner	0.21	**-0.74**	-0.28	-0.43
lib	UK	Ed Miliband	Labour	@ed_miliband	-0.44	**-0.60**	-0.12	-0.36
lib	US	Joe Biden	Democrat	@potus	0.25	-0.57	**-0.69**	-0.37
lib	US	Bernie Sanders	Democrat	@berniesanders	0.35	-0.95	**-0.53**	-0.16
lib	US	Hillary Clinton	Democrat	@hillaryclinton	0.16	-0.47	**-0.26**	0.16
con	IE	Lucinda Creighton	Renua	@lcreighton	**0.18**	0.06	0.29	0.06
con	IE	Peader Toibin	Aontu	@toibin1	**-0.16**	-0.08	0.34	-0.24
con	IE	Leo Varadkar	Fine Gael	@leovaradkar	**0.16**	0.16	0.16	0.29
con	UK	Neil Hamilton	UKIP	@neilukip	0.47	**0.16**	0.47	0.37
con	UK	Nigel Farage	Reform UK	@nigel_farage	0.16	**-0.47**	0.16	0.26
con	UK	Boris Johnson	Conservatives	@borisJohnson	0.38	**-0.13**	-0.25	0.00
con	US	Kevin McCarthy	Republican	@gopleader	-0.12	-0.43	**0.64**	0.54
con	US	Ted Cruz	Republican	@tedcruz	-0.18	-0.62	**0.81**	0.59
con	US	Donald Trump Jr	Republican	@donaldjtrumpjr	-0.13	-0.65	**0.52**	0.39
				Total Accuracy	0.50	0.67	0.83	0.67
				Accuracy with regional Datatset	0.83	0.67	1.00	0.67
				Accuracy of Global dataset on region	0.67	0.83	0.83	0.67

The results of this evaluation highlight a few interesting flaws and areas for future development within the implemented political leaning algorithm.

The Irish Politicians were correctly classified 67% of the time when using the global dataset, and 83% of the time when using the Irish dataset, while the Irish dataset gave an accuracy of 50% on the complete set of international politicians. The global dataset model misclassified Lucinda Creighton as neutral while both the Irish dataset model and the global dataset model misclassified Peader Toibin as liberal. Although the Irish dataset model is not perfect at classifying Irish politicians, it is able to discern the nuances of Irish conservatism more accurately than the global dataset model. Both dataset models correctly classified all liberal Irish politicians.

The UK Politicians were correctly classified 83% of the time with the global dataset model, and 67% of the time with the UK dataset model, while the UK dataset model had an accuracy of 61% on the complete set of international politicians. The global dataset model misclassified Boris Johnson as neutral, while the UK dataset model also misclassified Boris Johnson as neutral and classified Nigel Farage as liberal. The underperformance of the UK machine Learning dataset model can likely be attributed to the widely differing political agendas of the parties classed under the same label. While DUP and the Conservatives are both conservative within UK politics, they likely have very little in common when looking at speech patterns and talking points. Similarly, the SNP and Labour are both liberal, but have very different primary concerns within politics. Further development into this project in the future could raise the possibility of further splitting the UK into more specific political regions to alleviate the effect of this.

The US Politicians were correctly classified 83% of the time with the global dataset model, and 100% of the time with the US dataset model, while the US dataset model had an accuracy of 83% on the complete set of international politicians. The global dataset model misclassified Hillary Clinton as conservative. All other US politicians were correctly classified by both the US and the global dataset models.

Conclusions

From the study of all politicians classified by all the relevant datasets, we can see a few trends within the data.

The global dataset is very successful when used to identify liberal politicians globally, however the model struggles to identify conservative politicians. One reason for this may be that conservative politicians tend to take a traditional, nationalistic view of politics, which will appear very different across different nations with different traditions.

When the specific national datasets are used, the resulting models are better at identifying and correctly classifying the specifics of conservatism within their specific political regions. It could be argued that liberal politicians worldwide tend to support globalism and international cooperation more than conservative politicians. This may explain why the global data model was much more accurate at classifying liberal politicians, as they likely speak with less regional language and about less regional issues.

When looking at the topics discussed by the far reaches of either political ideology, we can see some similarities. The more centrist politicians on both the left and the right tend to comment on noncontroversial issues such as attracting jobs and improving infrastructure. Politicians to either side of the political spectrum would be more likely to focus on large scale international issues such as global warming or refugee and migrant rights, whether they are speaking in support of, or opposition to, prioritising these issues.

Another factor which can greatly impact how politicians communicate on Twitter is whether they are in the government, or in the opposition. Those in government express much more positive sentiments than those in opposition, as can be seen in Figures 64 through 69. Along with this more positive sentiment, politicians who are holding government positions naturally express more centrist views than they may have while campaigning or while they were in governmental opposition. This is an understandable and predictable phenomenon, however it makes the task of classifying political tweets even more complicated.

Overall the global dataset results in a reasonably accurate model for all regions studied, while the national datasets provide models that are more nuanced when analysing data relevant to a specific nation. Based on the complications encountered in this classification task, it seems highly unlikely that any one dataset could be created to train a model that would correctly classify international politicians, as political rhetoric is a many layered and ever changing metric to measure.

5.2 User Feedback

Once the planned project work had been completed, the next step of the project was to conduct user evaluation, and apply any relevant updates or fixes that present themselves during this user evaluation process.

5.2.1 Survey Layout

A comprehensive survey of the various application components was created and distributed along with a link to the live website. This survey was distributed to a wide group of people with differing technical abilities, using different devices to evaluate the website.

The survey was distributed in an email, which included basic instructions and a list of possible search queries. The text of this email can be found in the appendices (Section 8.2).

The survey consisted of 5 sections:

- **Instructions**.
  This section gave an overview of the rest of the survey so that users were clear on what was required and how to complete the survey. This section was added after some evaluators failed to recognise that the different sections began with different instructions.
  The Instructions given were as follows:
  This Evaluation will consist of 4 sections.
  Section 1: Explore the Analyse Tweets section
  Section 2: Explore the Compare Tweets section
  Section 3: Explore the Analyse Account section
  Section 4: Final Evaluation

  Please follow the instructions given at the start of each section
  Thank you for helping me out by filling out this evaluation

- **Analyse Tweets for a #tag or @user**
  Instructions:
  1. Go to the 'Analyse Tweets' page of the Twitter analysis application
  2. Search for a Twitter hashtag or user handle
  3. Read through the results output

  Questions:
  - *What was the #tag or @user that you searched for? (Short text answer)*
  - *Did you have any difficulties performing the search? (Yes/No/Other…)*
  - *If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)*
  - *Is the search response data understandable? (Yes/No/Other…)*

- ❖ *If you answered No to the above question, what was not understandable about the search response data? (Short text answer)*
- ❖ *Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)*
- ❖ *If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)*
- ❖ *Does the data make sense to you given the hashtag or user handle searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)*
- ❖ *If you answered No to the above question, what area or areas of the response did you disagree with? (Short text answer)*

- **Compare Tweets for #tags or @users**

  Instructions:

  1. Go to the 'Compare Tweets' page of the Twitter analysis application
  2. Search for a Twitter hashtag or user handle in the first section
  3. Search for a related/contrasting Twitter hashtag or user handle in the second section
  4. Read through the results output

  Questions:

  - ❖ *What were the #tags or @users that you searched for? (Short text answer)*
  - ❖ *Did you have any difficulties performing the search? (Yes/No/Other…)*
  - ❖ *If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)*
  - ❖ *Is the search response data understandable? (Yes/No/Other…)*
  - ❖ *If you answered No to the above question, what was not understandable about the search response data? (Short text answer)*
  - ❖ *Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)*
  - ❖ *If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)*
  - ❖ *Does the data make sense to you given the hashtags or user handles searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)*
  - ❖ *If you answered No to the above question, what area or areas of the responses did you disagree with? (Short text answer)*

- **Analyse Account for a Twitter @user**
  Instructions:
  1. Go to the 'Analyse Account' page of the Twitter analysis application
  2. Search for a Twitter user handle
  3. Read through the results output

  Questions:
  - *What was the Twitter @user handle that you searched for? (Short text answer)*
  - *Did you have any difficulties performing the search? (Yes/No/Other…)*
  - *If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)*
  - *Is the search response data understandable? (Yes/No/Other…)*
  - *If you answered No to the above question, what was not understandable about the search response data? (Short text answer)*
  - *Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)*
  - *If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)*
  - *Does the data make sense to you given the user handle searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)*
  - *If you answered No to the above question, what area or areas of the response did you disagree with? (Short text answer)*

- **Final Feedback**
  No instructions given as this survey section was self explanatory and retrospective based on the experience in completing section 2 through 5.

  Questions:
  - *What device did you use to perform this evaluation? (Computer/Tablet/Mobile Phone/Other...)*
  - *Overall, I am satisfied with the ease of completing the tasks in Sections 1, 2 and 3 of this evaluation. (Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)*
  - *Overall, I am satisfied with the amount of time it took to complete the tasks in Sections 1, 2 and 3 of this evaluation. (Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)*

- *Overall, I am satisfied with the support information (online-line help, messages, documentation) available to me in completing the tasks in Sections 1, 2 and 3 of this evaluation.* *(Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)*

- *How would you rate the graphic design of the application on a scale from 1 to 10?* *(Rate on an unlabeled scale from 1 to 10)*

- *How would you rate clarity of the terminology used in the application on a scale from 1 to 10?* *(Rate on a scale from 1: Unclear to 10: Clear)*

- *What was the weakest aspect of the application in your opinion?*
  (choose 1 of :
  The graphic design
  The Navigability of the site
  The application was slow
  The information was not interesting
  The information seemed inaccurate
  There was too much information presented
  There was not enough information presented
  Errors when trying to use the application
  Other..)

- *What was the strongest aspect of the application in your opinion?*
  (choose 1 of :
  The graphic design
  The Navigability of the site
  The application was fast
  The information was interesting
  The information was well presented
  Other..)

- *Any other comments or feedback?* *(Long text answer)*

5.2.2 Survey Results

I received 20 responses to my survey. The specific user feedback highlighting bugs which were fixed or features which were implemented are flagged in this section with the labels 'Bug since resolved' and 'Feature since implemented'.

The responses to the survey were as follows:

Analyse Tweets for a #tag or @user

- What was the #tag or @user that you searched for?
    - *@jacobinmag,@RuthCoppingerSP,#scamdemic,@piersmorgan,@HeinzBeans,@timjdillon, @POTUS,@OANN,#FreeCian,@markgoldbridge,@ripoffnuig,@rtenews,#trump, @MaryLouMcDonald,#firstdates,#covid,@roisinshortall,@davidmcw,@careersportal,@cnn*

- Did you have any difficulties performing the search? (Yes/No/Other…)
    - *No (90%)*
    - *Yes (10%)*

- If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)
    - *Tried a good few Twitter handles before I got one that worked*
    - *ValueError: not enough values to unpack (expected 9, got 8)* (Bug since resolved)

- Is the search response data understandable? (Yes/No/Other…)
    - *Yes (85%)*
    - *No (5%)*
    - *Other (10%)*

- If you answered No to the above question, what was not understandable about the search response data? (Short text answer)
    - *Just an error* (Bug since resolved)
    - *The political DB buttons. From a software dev perspective I understood why I had to pick an option before searching and you can't just change dataset once the information is gathered. I'd suggest having some kind of visual locking on this setting once you change it and results are already displayed. Or perform another search using this new political dataset preference if its changed (if traffic isnt an issue)*
    - *Sentiment, most popular, most used emojis, most used words, hashtags and tagged accounts - all good. Political Leaning interesting that it is different if I choose Ireland or Global filter. In the tweet emotions I would like to understand what "other" includes*
    - *most popular tweet in german*

- Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)
    - *Yes (25%)*
    - *No (65%)*
    - *Other (10%)*

- If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)
    - *I wonder about displaying the tweet with the most likes maybe?* (Feature since implemented)

- *Id like to have been able to go in and see the tweets* (Feature since implemented)
- *For both political leaning and sentiment, give examples of users tweets that reflect the results the most*
- *Possibly the countries that the data came from?*
- *handle(s) that engage most with content*
- *A link to the Twitter account would be a nice to have but not critical at all. Maybe also some sort of popularity i.e usually gets 10k likes per tweet*

- Does the data make sense to you given the hashtag or user handle searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)
  - *Yes (80%)*
  - *No (10%)*
  - *Other (10%)*

- If you answered No to the above question, what area or areas of the response did you disagree with? (Short text answer)
  - *Generally yes, however it does seem to struggle sometimes, for example Arlene Foster's account was labelled as being more left wing than Michelle O'Neills*
  - *Different sentiments when using different datasets (Global/US etc)* (Bug since resolved)
  - *See above*
  - *It says that the tweets are overall more Positive and Neutral in sentiment than Negative, but considering the hashtag, isn't the sentiment mostly negative around the whole FreeCian scandal?*

Compare Tweets for #tags or @users

- What was the #tag or @user that you searched for?
  - *@nytimes and @jacobinmag,@RuthCoppingerSP,@newsmax @cnn,@piersmorgan @chrissyteigen,@DunnesStores @Tesco,@timjdillon and @potus,@POTUS @rtenews,#maga,#ElonMusk,@chelsearory,@markgoldbridge,#berniesanders #trump,@newsmax,#trump #biden,@LeoVaradkar @RBoydBarrett,@maryloumcdonald @michealmartintd,@leovaradkar, @maryloumcdonald,@michealmartintd, @leovaradkar,@davidmcw @rtenews,@careersportal, @Education_Ire, @NCGEGuidance;,@cnn, @newsmax*

- Did you have any difficulties performing the search? (Yes/No/Other…)
  - *Yes (10%)*
  - *No (90%)*

- If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)
  - *I entered both tags, but the second disappeared when i submit* (Feature since implemented)
  - *Hitting enter only loads results for one side of the comparison and removes the Twitter handle from the other* (Feature since implemented)

- Is the search response data understandable? (Yes/No/Other…)
  - *Yes (95%)*
  - *No (0%)*
  - *Other (5%)*

- If you answered No to the above question, what was not understandable about the search response data? (Short text answer)
    - *When comparing, it would be easier if the corresponding fields were underneath each other*

- Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)
    - *Yes (25%)*
    - *No (70%)*
    - *Other (5%)*

- If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)
    - *Id like a button to go in and read the tweets* (Feature since implemented)
    - *rather than just displaying the individual stats maybe something along the lines of the second user's tweets are more political by a factor of 1.2* (Feature since implemented)
    - *both left leaning and similar - would it be possible to see the differences in more detail?*
    - *Would like to see number of tweets with no engagement*
    - *Possibly some of the top recent tweets in this category, twitter does this when showing a tag to give you a quick insight* (Feature since implemented)
    - *Maybe identify any previous interactions between the accounts? Or tweets where both of them are tagged*

- Does the data make sense to you given the hashtag or user handle searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)
    - *Yes (88.2%)*
    - *No (11.8%)*
    - *Other (0%)*

- If you answered No to the above question, what area or areas of the response did you disagree with? (Short text answer)
    - *I thought Leo Varadkar's tweets would be leaning to one side, rather than in the middle*
    - *Would like to understand what words indicate "anger" or "trust"*
    - *Anger everywhere???* (Bug since resolved)

Analyse Account for a Twitter @user

- What was the #tag or @user that you searched for?
    - *@rtenews,@tuirseachgodeo,@irishtimes,@johnlegend,@marknorm,@leovaradkar, @SimonHarrisTD,@aoc,@tedcruz,@BorisJohnson,@markgoldbridge,@piersmorgan, @presidentirl,@officialmcafee,#covid,@pontifex,@careersportal,@nigella_lawson*

- Did you have any difficulties performing the search? (Yes/No/Other…)
    - *No (95%)*
    - *Yes (5%)*

- If you answered Yes to the above question, what were the difficulties that you encountered? (Short text answer)
    - *It was slow enough, I though the webpage had frozen but it loaded then* (Feature since implemented)

- Is the search response data understandable? (Yes/No/Other…)
    - *Yes (90%)*
    - *No (0%)*
    - *Other (10%)*

- If you answered No to the above question, what was not understandable about the search response data? (Short text answer)
    - *Somewhat. If I was a twitter user it might be more obvious*
    - *"amp" (&) as one of the most used words* (Bug since resolved)

- Is there any data which you would have liked to see in the response data, which is not there? (Yes/No/Other…)
    - *Yes (5%)*
    - *No (95%)*
    - *Other (0%)*

- If you answered Yes to the above question, what additional response field do you think would be nice to add to the application? (Short text answer)
    - No responses

- Does the data make sense to you given the hashtag or user handle searched? i.e, Do you agree with the statements made by the algorithm? (Yes/No/Other…)
    - *Yes (65%)*
    - *No (25%)*
    - *Other (10%)*

- If you answered No to the above question, what area or areas of the response did you disagree with? (Short text answer)
    - *More anger?* (Bug since resolved)
    - *Only thing slightly surprising is that FG is right leaning but Leo's tweets have no strong political leaning*
    - *Emotion overwhelmingly anger even though sentiment very positive* (Bug since resolved)
    - *John legend isn't right wing*
    - *Tweets were marked as strongly Left.*
    - *Question Political Bot, Fake followers and flagged as fake*
    - *I thought RTE News would have a more Neutral Sentiment than Negative*

Final Feedback

- What device did you use to perform this evaluation? (Computer/Tablet/Mobile Phone/Other...)
    - *Computer (80%)*
    - *Tablet (10%)*
    - *Mobile Phone (10%)*

- Overall, I am satisfied with the ease of completing the tasks in Sections 1, 2 and 3 of this evaluation. (Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)
    - *10 (50%)*
    - *9 (20%)*
    - *8 (20%)*
    - *7 (5%)*
    - *6 (5%)*

- Overall, I am satisfied with the amount of time it took to complete the tasks in Sections 1, 2 and 3 of this evaluation. (Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)
    - *10 (35%)*
    - *9 (35%)*
    - *8 (20%)*
    - *7 (10%)*

- Overall, I am satisfied with the support information (online-line help, messages, documentation) available to me in completing the tasks in Sections 1, 2 and 3 of this evaluation. (Rate on a scale from 1: Strongly Disagree to 10: Strongly Agree)
    - *10 (40%)*
    - *9 (20%)*
    - *8 (20%)*
    - *7 (5%)*
    - *5 (15%)*

- How would you rate the graphic design of the application on a scale from 1 to 10? (Rate on an unlabeled scale from 1 to 10)
    - *10 (10%)*
    - *9 (15%)*
    - *8 (25%)*
    - *7 (25%)*
    - *6 (20%)*
    - *5 (5%)*

- How would you rate clarity of the terminology used in the application on a scale from 1 to 10? (Rate on a scale from 1: Unclear to 10: Clear)
    - *10 (40%)*
    - *9 (25%)*
    - *8 (25%)*
    - *7 (10%)*

- What was the weakest aspect of the application in your opinion?
    - *The graphic design (40%)*
    - *The application was slow (15%)*
    - *Errors when trying to use the application (10%)*
    - *The information seemed inaccurate (10%)*
    - *There is a lot of writing, and the colour of the writing doesn't make it stand out (5%)*
    - *nothing stood out as being weak (5%)*

    - *Everything was excellent! (5%)*
    - *I couldn't remember some twitter handles it would be handy if there was a link which I could have searched (5%)*
    - *all seemed fine to me (5%)*

- What was the strongest aspect of the application in your opinion?
    - *The information was interesting (60%)*
    - *The information was well presented (30%)*
    - *The Navigability of the site (5%)*
    - *The application was fast (5%)*

- Any other comments or feedback? (Long text answer)
    - *I wonder could some of the text/graphics be re-sized to "use up" more of the space - so you have less grey space?* (Feature since implemented)
    - *First 3 pages all the same?*
    - *I only noticed the dataset selection when I was almost finished.*
    - *This is an interesting project! Certain people would find this application extremely useful for political analysis*
    - *Very well done, very impressed by it*
    - *Very impressive website, very thorough*
    - *I really enjoyed it and would spend loads of time searching different handles - would like to understand some of the information behind the algorithm as not sure why the @careersportal got some of the probability results - would also like to see what % of tweets per handle get little or no engagement*
    - *Looks great, just one small thing. When I clicked the 'Twitter Analysis Application' in the top-left corner, I expected it to bring me to Home. Instead, I got URL Not Found* (Bug since resolved)

5.2.3 Survey Conclusion

The responses to the survey showed that most survey respondents were interested in the data returned by the application and pleased with the application features and performance. The User evaluation did highlight some minor bugs which were present in the application. Any identified bugs were addressed quickly and so affected very few survey respondents.

The specific bugs which were identified and resolved due to this User evaluation survey were as follows:

- System error if searching for nonexistent Twitter handle
- Tweets classed as angry if no emotion identified
- Data error leading to different sentiments when using different datasets
- "amp" treated as a word within the most used words

Some features suggested by responders were implemented, such as the ‘see tweets’ button which enables a user to see the stream of tweets which was analysed. The font size was also increased in response to user feedback. Informational messages and a home page were added in response to feedback from users who were confused about some aspects of the application functionality, such as the political leaning datasets selection.

Users were confused by the design of the compare tweets page, which had separate data entry and submit areas for each of the two tweet-sets to compare. In response to this feedback, The entry fields were combined with a single submit button,and a direct comparison selection was added to show a clear and simple direct comparison between the users.

Some user feedback showed that the political leaning results for certain public figures were not as expected. These unexpected political leaning results were likely due to the political datasets being tuned with overtly political speech, and thus struggling in classifying tweets which have no political intentions.

One user identified that a hashtag which was predominantly angry was classified as positive in sentiment. Upon further investigation into the hashtag in question, it seems that there was a lot of sarcasm and cynicism being used in this hashtag, along with a lot of internet specific language and memes. The sentiment analysis tool which was implemented in this project struggles to detect these linguistic techniques, and thus classified a negative hashtag as more positive than negative.

There seemed to be significant consensus that the strongest aspect of the application was the fact that the information it showed was interesting. This is a promising result which highlights the sheer volume of work that went into the backend development of the application, with the goal of making something that was both interesting and useful.
There was also a reasonable consensus that the weakest aspect of the application was the user interface. This is an understandable response by the evaluators as the user interface was the only aspect of the application with which they could easily see and interact, despite it only comprising a small fraction of the project work. If this application were to be released as a public service or product, then it would definitely be valuable to consult with graphical experts and front end engineers to optimise the user experience.

Overall, the user evaluation showed that the project goals had been met and the developed application is an interesting tool for users to use to analyse the Twitter activity of public figures and within hashtags. Users who were active on Twitter were very interested in how their own profiles were analysed, and some informed me that they were pleased to see that their actual political leaning was reflected in their Twitter activity.

6 Future Work and Improvements

This was a very interesting project to work on. It provided many challenges to overcome or work around.
The constraints encountered in the project development, (Tool/API limits, Language classification complexity, time constraints), were significant challenges to overcome in the development of the application.
There are many interesting ways in which to expand upon this project in the future which were not explored in this project due to the stated project constraints.

Some of these potential future expansions of the project would be

Expanding tweet collection beyond 7 days
Twitter offers a paid tier in its API which supports full archive queries, removing the difficulties and constraints of analysing only tweets from within the last 7 days. The ability to analyse the full set of tweets by a given user, or within a given hashtag would greatly expand the scope and accuracy of this project.

Further improving political datasets
If the Twitter rate limits were not a constraint, then the datasets used in training the machine learning algorithm could be massively increased in scale, thus improving the accuracy of the political leaning detection. Comprehensive large scale political tweet datasets would be fascinating to examine, and would not be difficult to gather using the paid tier of the Twitter developer API.

Support for other languages
While English language Twitter activity was the only Twitter data analysed in this project, there is certainly enough traffic to analyse the Twitter activity within countries that use other languages. This could allow for very interesting analysis into how internet discourse varies across languages, and could highlight similarities and differences between Twitter activity in different countries.

Support for more political regions
Within the scope of this project, only the political regions of Ireland, the UK and the USA were considered, with the 'Global' dataset option simply consisting of the bulk dataset from those 3 regions. If datasets could be gathered from all of history, rather than being limited to the last 7 days, then datasets could potentially be gathered to expand the political regions to include other english speaking countries. In conjunction with the support for other languages, this could allow for further expansion into analysing political data for countries such as Spain, Germany, France or countries such as Canada in which multiple languages are used in politics.

Political Polarisation
The political polarisation detection, which was explored as part of the initial project scope, would be a very interesting component to implement in a future version of the application. This feature was not feasible to implement over the course of this project due to the constraints of the free tier of the Twitter developer APIs. With the paid tier of the Twitter developer APIs, and some improved server hardware to handle more complex and frequent queries, it would be possible to quantify the political polarisation within a topic, to see how much discussion there is between people of opposing viewpoints can be seen in a topic.

Expansion into linguistics of political speech and online speech
This project, being a single person Final Year Project, was conducted as an individual project by a computer science student. This raised some challenges when the linguistic complexities of analysing and categorising political language became apparent. The project could be expanded further into the field of linguistics, in conjunction with experts within that field, to learn more about the intricacies of political speech along with how and why online political speech differs from more formal or live political speech. Further, more in-depth linguistic analysis of political rhetoric on Twitter could provide the data and knowledge needed to improve upon the political leaning analysis system implemented within this project.

Improved User Interface Design
40% of survey respondents found the user interface of the application to be the weakest aspect of the application. The focus of this project was on the backend algorithms and natural language processing techniques, so it is understandable for the user interface to be one of the weaker aspects of the application. If this project were to be developed further, it would be valuable to improve the user interface to make it a bit cleaner and more modern. The current user interface achieves the basic goal of being fully functional, while a more complex implementation could make the application more intuitive and overall more easy to use.

More reliable and secure web server and web security
The project was hosted on the web1.cs.nuigalway server which was freely available to NUI Galway students of computer science. This was an easy way to support a publicly accessible website, however it is not the fastest and most reliable system. If the project scope were to be expanded, the application could be hosted on a paid server with a more relevant url and faster processing. Protection against injection attacks and server manipulation could also be implemented to make the website more secure.

7 Conclusion

Over the course of this project's development, I have had the opportunity to learn many new skills, and build upon my existing skills in the areas of software engineering and project management.

The main lessons I draw from the process of planning, developing and testing this project are:

- Classification of text data is complex and the algorithms used to classify text data must be able to evolve along with the evolution of language. This is particularly evident when looking at online language use, since language use online evolves more rapidly than language use in more formal forms of media such as books and articles.
- Sensible project planning and robust development practices must be at the core of any successful project. The thorough planning for this project, along with the CICD development processes of consistent, commented, high quality and always functional code allowed this ambitious project to be developed without any major issues, and completed ahead of the scheduled Final Year Project deadlines in May 2020.
- Completing a project of this scope as an individual developer was an interesting challenge in exploring a wide range of skills, from front end html/css web design, to back end machine learning algorithm implementation and project management and planning. If this project were to be expanded by a team of people with different expertise and skill sets, there would be the potential to push it further, most notably by improving the user interface through work by a front end design developer and through deeper linguistic analysis of sentiment, emotion and political leaning detection through work by those with a speciality in linguistics.

This project was a very valuable part of my college experience, and gave me many opportunities to improve upon and demonstrate my skills. I consider this final year project to be a success, as I achieved the project objectives while learning a lot and continually adjusting and expanding upon the project scope.

8 Appendix

8.1 References

[1] - Political communication and influence through microblogging - Accessed 11/03/2021
S. Stieglitz and L. Dang-Xuan.
an empirical analysis of sentiment in Twitter messages and retweet behavior.
In 2012 45th Hawaii International Conference on System Sciences.

[2] - Finding interesting posts in Twitter based on retweet graph analysis - Accessed 11/03/2021
M.-C. Yang, J.-T. Lee, S.-W. Lee, and H.-C. Rim
In Proceedings of SIGIR, 2012.

[3] - Style Matters!: Investigating Linguistic Style in Online Communities - Accessed 11/03/2021
Padmini Srinivasan, Osama Khalid
ICWSM, 2020

[4] - Microsoft Azure Text Analytics Service Documentation - Accessed 11/03/2021
https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/

[5] - NRC Emotion Lexicon Documentation - Accessed 11/03/2021
https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[6] - VADER SentimentIntensityAnalyzer NLTK Documentation - Accessed 11/03/2021
https://www.nltk.org/api/nltk.sentiment.html

[7] - Political Sentiment Lexicon - Accessed 11/03/2021
https://rstudio-pubs-static.s3.amazonaws.com/338458_3478e1d95ccf49bf90b30abdb4e3bd40.html

[8] - Tensorflow Documentation - Accessed 11/03/2021
https://www.tensorflow.org/learn

[9] - Scikit Learn Documentation - Accessed 11/03/2021
https://scikit-learn.org/stable/getting_started.html

[10] - Botometer - Accessed 11/03/2021
https://botometer.osome.iu.edu/

[11] - BotSight - Accessed 11/03/2021
https://download.botsight.nlok-research.me/

[12] - Python 3.8 documentation - Accessed 11/03/2021
https://developer.twitter.com/en

[13] - Python Flask Documentation - Accessed 11/03/2021
https://flask.palletsprojects.com/en/1.1.x/

[14] - Python unittest Documentation - Accessed 11/03/2021 https://docs.python.org/3/library/unittest.html

[15] - Python Pytest Documentation - Accessed 11/03/2021
https://docs.pytest.org/en/stable/contents.html

[16] - Twitter Developer API Documentation - Accessed 11/03/2021 https://developer.twitter.com/en

[17] - Kaggle Dataset Search Page - Accessed 11/03/2021
https://www.kaggle.com/datasets

[18] - Github Documentation - Accessed 11/03/2021
https://docs.github.com/en

[19] - Ideological Books Corpus - Accessed 22/03/2021
https://people.cs.umass.edu/~miyyer/ibc/index.html

[19] - Covote Dataset - Accessed 22/03/2021
http://www.cs.cornell.edu/home/llee/data/convote.html

[20] - Political Twitter Corpus - Accessed 22/03/2021
https://www.usna.edu/Users/cs/nchamber/data/twitter/

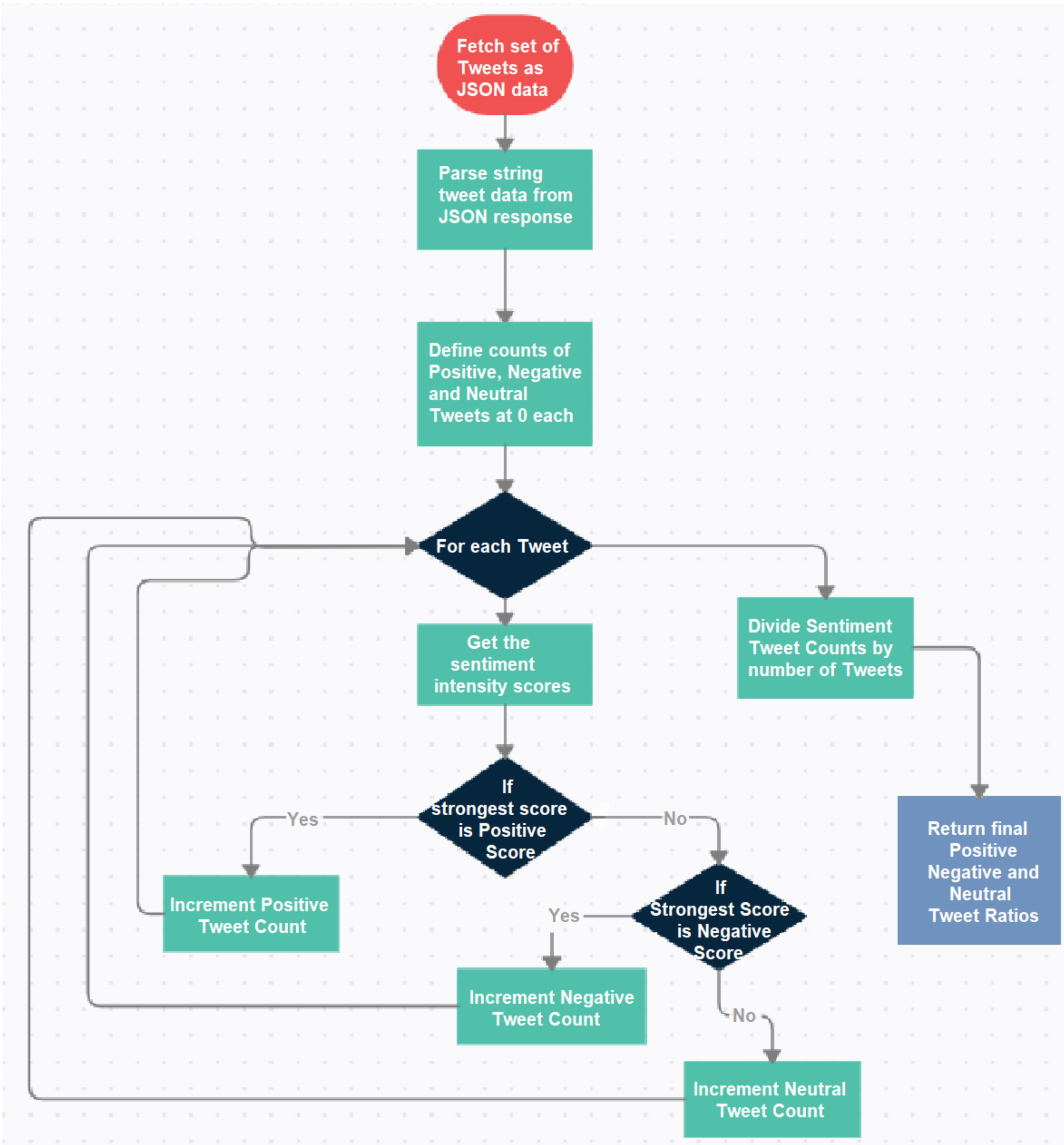
[21] - Harvard Dataverse Twitter Datasets - Accessed 22/03/2021
https://dataverse.harvard.edu/dataverse/gwu-libraries

[22] - Kaggle Political Tweet Classification Dataset - Accessed 22/03/2021
https://www.kaggle.com/kapastor/democratvsrepublicantweets

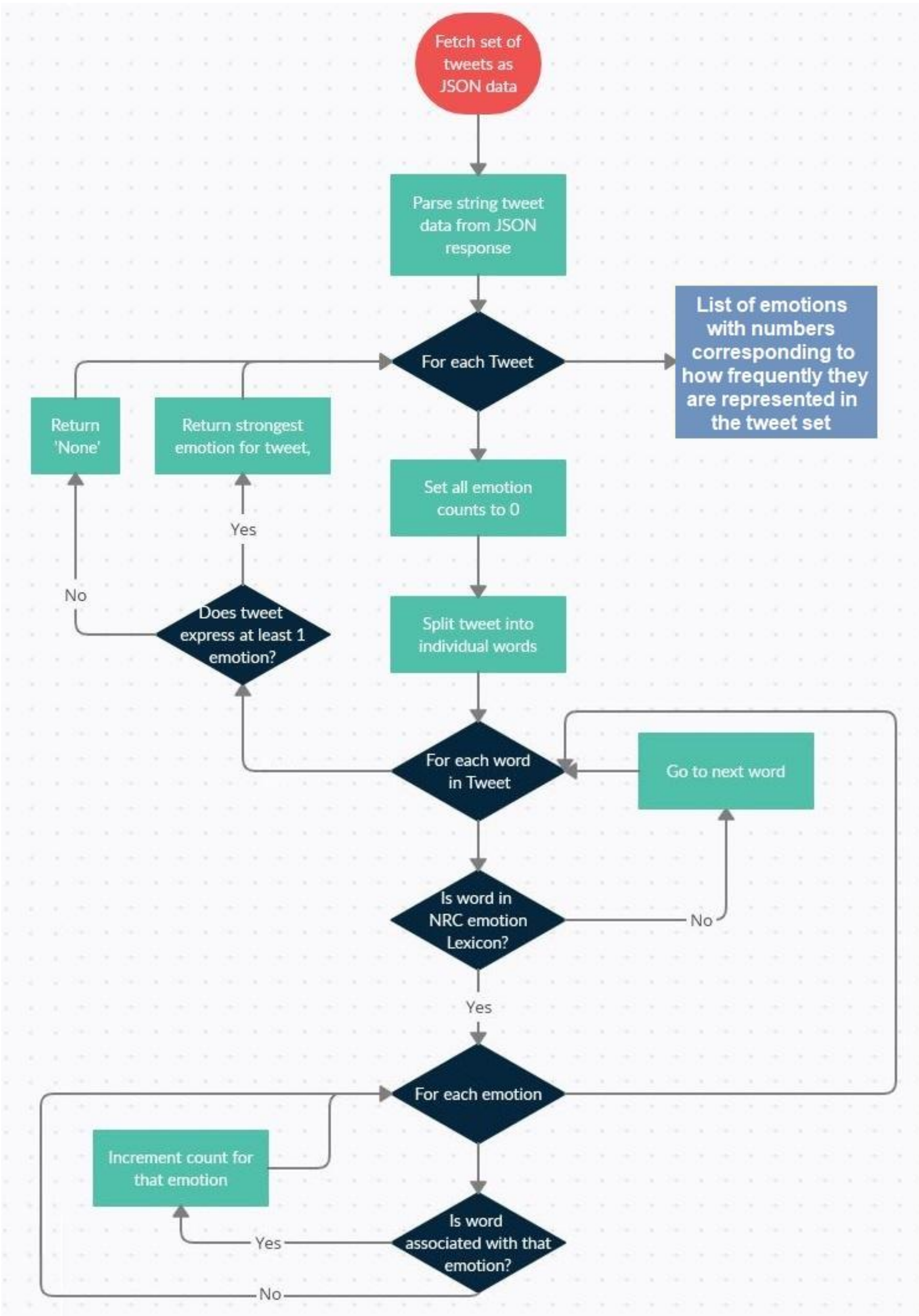
[23] - Bootstrap Documentation - Accessed 22/03/2021
https://getbootstrap.com/docs/4.1/getting-started/introduction/

8.2 Process Flow Diagrams

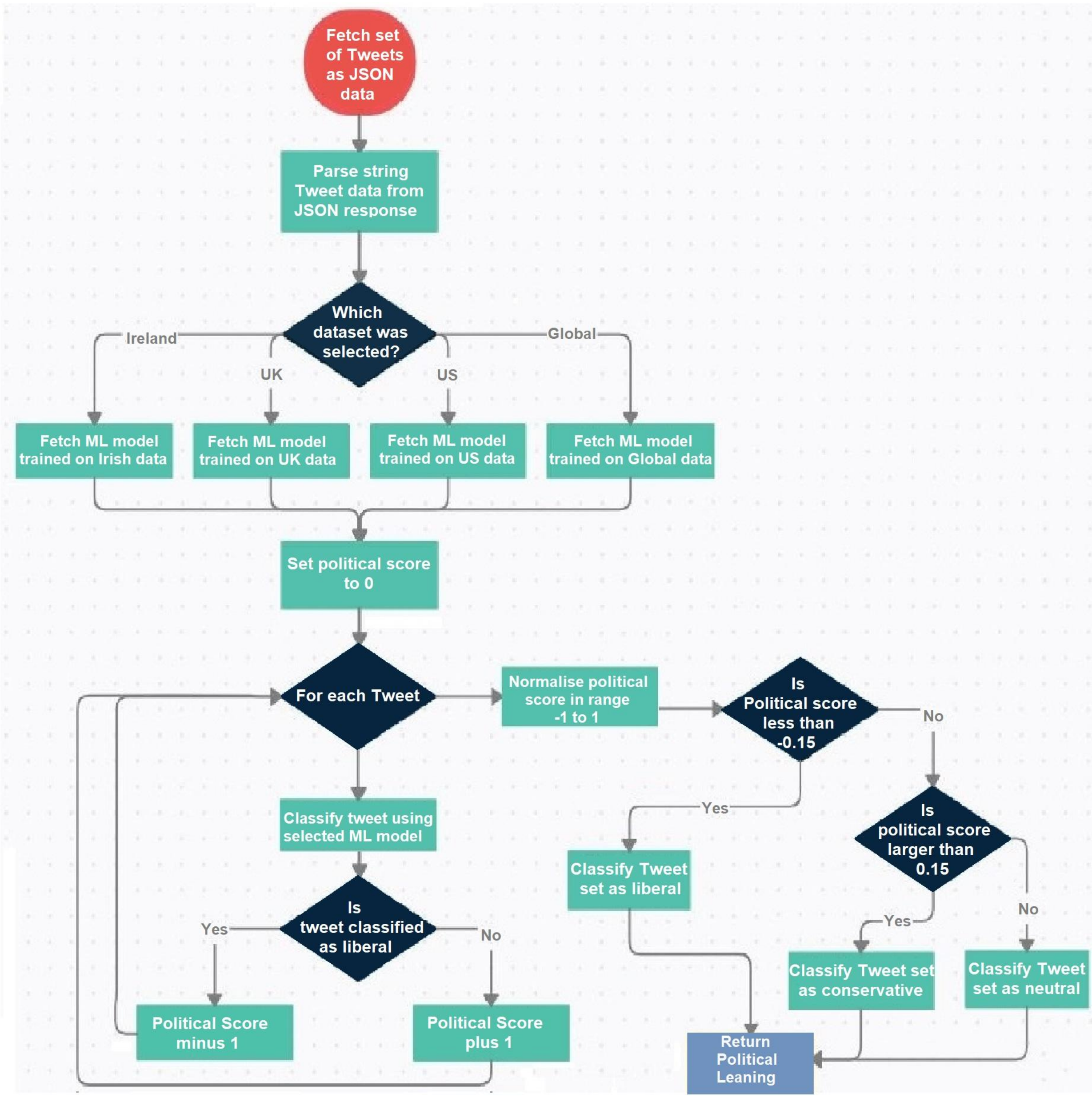
8.2.1 Sentiment Analysis Flow Diagram



8.2.2 Emotion Analysis Flow Diagram



8.2.3 Political Leaning Analysis Flow Diagram



8.3 User Evaluation Email

Hi,
I am attaching a feedback form and a link to my website for evaluation.
If you have a chance to complete the evaluation that would be much appreciated!

FYP application website: http://web1.cs.nuigalway.ie:8081/
Feedback survey: https://forms.gle/tPApq7bZPY3aN48o7

The best way to fill out the survey is to have the survey open alongside the website, following the survey instructions, and filling the survey out as you go..

Possible #tags and @users to search:

1. @rtenews – should be politically neutral
2. @newsmax – strongly right leaning US news network
3. @cnn – left leaning US news network
4. @aoc – liberal Us politician
5. @tedcruz – conservative Us politician
6. @maryloumcdonald – liberal Irish politician
7. @leovaradkar – centre-conservative Irish politician
8. @keir_starmer – liberal UK politician
9. @jacob_rees_mogg – conservative UK politician
10. #ireland
11. #covid
12. #disgrace
13. #trump
14. #biden
15. #galway
16. #dogs
17. #happy

Thank you!
Aideen