Relax, it's just a joke!
A Twitter sentiment analysis study.

Isabel Rosário

June 29, 2022

Contents

Summary

Following up on my previous project for this course, in which I, along with some colleagues, made an analysis of Twitter as a technology and as a service, I propose now to talk about a closely related tool: sentiment analysis.

In the previous report, we explored the inner workings of Twitter from a deeply technical point of view, and one of the things we strove to focus on was the phenomenon of Twitter trends: what they are, how they work and what their lifecycle looks like. Even though this report was of much interest to me personally, and I wanted to continue exploring the theme, I knew our research had been extensive and our report covered a lot of the main points already. I needed to approach the idea from a different angle.

This was when I thought I might use my curiosity with Artificial Intelligence research and somehow combine the two topics in this new assignment. The result came quite naturally after some investigation, and the topic of sentiment analysis quickly became my new focus point.

As such, I intend to begin by giving a careful explanation of all the parties involved, so that all potential readers may be familiarized with all the concepts necessary for what follows. I then go into technical detail on sentiment analysis: how it works, its different types, some possible algorithms and challenges in its development.

The following section explains how this tool is relevant for Twitter trends and for the social network overall. I shall then present a tool I built that facilitates sentiment analysis data visualization, while also giving some insight into Twitter trends themselves.

I will then conclude with an overall appreciation of this technology and some thoughts on my experience writing this assignment.

Chapter 1

Framework

Before I delve into the technical overview of how sentiment analysis works, it is important to define it, as well as the remaining technologies that contextualize it within this assignment. On account of this, I shall start by talking about what Twitter is, following it up with a look at Twitter trends specifically. Next, it is important to give some introduction to Artificial Intelligence and Machine Learning, which are the scientific fields in computer science into which sentiment analysis fits. This section ends with a definition of sentiment analysis itself, which shall prepare the reader for the technical dive that follows.

1.1 What is Twitter?

To quote the previous assignment I worked on, Twitter is a social network that has been around since 2006, becoming popular around 2009. It has, since then, been ubiquitous in the online social landscape, being the main stage for popular trends, social commentary and public opinion.

To briefly explain its usage, each user can tweet, given that a tweet can consist of simple text (with a character limit), one or multiple images, videos, links, among others, as well as a mix of all of them. Everyone will be able to see such tweets, unless the user chooses to narrow down their audience. The user can also restrict who can retweet (share) or reply to their tweets. The social network is equipped with a private messaging system that allows users to exchange private messages or create group chats. As well as following other users, one can also follow topics, which represent areas of interest (such as "Art", "Soccer" or "Programming").

1.2 What are Twitter trends?

Twitter trends, also sometimes called "trending topics", are words, phrases or topics that get sudden popularity among Twitter's users. This popularity is translated in terms of number of tweets that mention the word or expression at hand, or that are about the given topic, as well as interactions with those same tweets (in the form of likes, replies or shares).

These topics gain popularity if the users themselves make an effort to tweet about them in mass, or if some external event triggers conversation concerning them.

1.3 What are artificial intelligence and machine learning?

Artificial Intelligence is a field of computer science that focuses on developing machine intelligence through artificial agents, which are systems capable of perceiving their environment and making choices based on information extracted from said environment, with the objective of fulfilling a given goal as optimally as possible.

Machine Learning is a subsection of Artificial Intelligence and it refers to the research field devoted to developing systems that "learn". This means that they use the data available to them with the purpose of improving their own performance in a set of tasks. The workflow in this field tends to start with the building of a model based on sample data, which must then be capable of making predictions that have not been explicitly woven into its code.

1.4 What is sentiment analysis?

Sentiment analysis is a field of Natural Language Processing (the development of computer programs that understand and process human language) that focuses on the identification and extraction of subjective opinions or feelings from written speech in natural language. This can be done with the focus of identifying bias about a given topic, or just the general feeling of a certain text. To sum up, the goal is to find whatever there is in the text that points to attitude, emotion, or bias.

Chapter 2

Technical Analysis

In this section, I shall give a detailed explanation of the technical details behind sentiment analysis. I will start with a general delineation of how it works and follow that by listing the several types of sentiment analysis that have been developed. Next, I will talk a bit about some algorithms or methods used in this research field, as well as the major challenges it faces.

2.1 How sentiment analysis works

As has been explained before, sentiment analysis aims at extracting opinions or emotions out of a text in natural language. For this purpose, it combines natural language processing (NLP) and machine learning (ML) algorithms. It would then be useful to start by defining what we mean by "opinion" in the scope of sentiment analysis work. An opinion is a **subjective assessment of a given topic**, often based on empirical experience. It is influenced by both facts and emotions.

Knowing that, we can now list the main use cases of sentiment analysis:

- Identify and extract biased or opinionated information from a text;
- Attribute a polarity to it (meaning if it is positively or negatively opinionated);
- Identify the topic or subject matter of the text;
- Identify the opinion holder.

Sentiment analysis can be applied on document, sentence or sub-sentence levels. Since natural language is such a complex and nuanced affair, it is useful to categorize opinions into different labels:

- A **direct opinion** makes a statement about a topic in a direct manner (for example, saying "X is very pretty.");
- A **comparative opinion** happens when X is compared to Y relative to some attribute they both share (this is like saying "X is prettier than Y.");
- An **explicit opinion** is one where the bias or emotion is clearly expressed (say, for example, "I didn't enjoy the Web app because it had image loading problems.");

- An **implicit opinion** expresses sentiment without directly formulating it with words (for example, "The Web app had image loading problems from the start.").

Further along, I shall talk about concrete sentiment analysis algorithms.

2.2 Types of sentiment analysis

Let us now take a look at some different types of sentiment analysis:

- **Fine-grained** analysis is directed at extracting polarity from the text. As has been stated before, polarity refers to the positive or negative nature of an opinion, and it can be separated into the categories of "very positive", "positive", "neutral", "negative", or "very negative", which is equivalent to a 5-star scale. This type of sentiment analysis is immensely useful when processing review-type texts;

- **Aspect-based** sentiment analysis makes for more in-depth processing than the type presented before, as it allows for the identification of specific topics that are being discussed in the text. Even in a review of a product, let's say, it not only allows for polarity extraction, but it also identifies the specific aspects of the product or service that the author has opinions on. If a user expressed "The ferris wheel at the county fair felt quite unsafe!", aspect-based analysis would identify that the opinion was not only negative, but also directed at the ferris wheel in particular;

- **Emotion detection** helps detect specific emotions, as the name suggests. These emotions include happiness, anger, fear, etc. Systems that go about doing emotion detection often utilize lexicons, which are sets of words representative of certain emotions, while more advanced algorithms include machine learning strategies. Using ML is quite more effective than lexicons, as different people will often express the same emotions in varied ways, and lexicons fail to take that variety into account;

- **Intent analysis** tries to accurately determine the author's intent in writing the text. For example, in terms of product or service consumers, this type of analysis can help identify if the user plans to make a purchase or if they are just browsing. This information is then used for ad targeting.

2.3 Sentiment analysis algorithms

I will now go into further detail on a couple of different concrete approaches to sentiment analysis. These constitute the two main methods that are currently used in this field of work.

2.3.1 Rule-based approach

The rule-based approach to sentiment analysis does not make use of any machine learning models in order to do its job. As the name suggests, it is simply based on a **set of**

rules that are applied to a text in order to classify it as positive, negative or neutral. Additionally, it can also identify subjectivity and the topic of the text or opinion.

There are a set of defined ordered steps that must be followed in order to correctly perform rule-based sentiment analysis, and those are:

1. **Stemming**: this means cleaning the text, i.e., removing any special characters, like numbers, from it;

2. **Tokenization**: this means breaking up the target text into smaller pieces (tokens), which can be either sentences or just words (sentence vs. word tokenization);

3. **Enrichment**: also known as Part of Speech (POS) tagging, this is converting each token into a tuple in the form of (word, tag). This helps to preserve the context of the token within the text;

4. **Parsing**: this can also be referred to as "stopwords removal", and it relates to the removal of words that carry very little to no useful information (stopwords). There is a list of stopwords for every language. In English, this includes words like "me", "those", "haven't", "why", etc.;

5. **Lexicon analysis**: this step is responsible for obtaining the stem words. A stem is a part of a word that gives away its lexical meaning. There are two popular techniques for doing this: **stemming** and **lemmatization**. Stemming often results in meaningless root words, since it only removes some characters at the end of the target words (so, for example, the stem word for "computer" could be "comput"). Lemmatization results in meaningful root words (returning "gift" for "gifting", for example), and it makes use of POS tagging.

This approach works by using two different sets of words: one containing only the positively connotated ones, and the other containing only the negative words. The algorithm parses the text and finds words that match either list, calculing which kind is more often found in the text. The most prevalent type decides the text's polarity.

As might be obvious, this method lacks flexibility, as well as precision. It does not take the context into account at all. It really is too simple to be of any use in complex situations, but it can be applied to simpler scenarios like determining the tone of messages in the context of customer support, or as groundwork to then implement a machine learning solution.

2.3.2 Automatic approach

Unlike the previously discussed approach, automatic sentiment analysis uses machine learning algorithms in order to truly dig into and analyse a text. Due to this, its precision, flexibility and accuracy greatly surpass those of the rule-based approach, and it becomes possible to evaluate a text according to a set of diverse criteria without adding too much complexity to the code itself.

This method uses a specific type of machine learning algorithms called **classification algorithms** which, as the name indicates, go about placing the target data into a set of different categories. This implies recognizing, understanding, and grouping ideas.

These methods are generally supervised, meaning they use labeled data sets in order to train themselves. Additionally, unsupervised ML algorithms are often used to explore new data.

In terms of specific classification algorithms that are used for sentiment analysis, we have the following:

- **Linear regression**: this machine learning model is represented by a linear equation which feeds on a set of input values and results in a set of output values. It estimates the values of the coefficients for this equation;
- **Naive Bayes**: this is actually a collection of classification algorithms based on the Bayes' Theorem, which attempts to determine the probability of an event based on previous conditions related to it;
- **Support vector machines**: these are supervised learning models based on statistical learning frameworks that, given some training data belonging to one category or another, will attempt to assign new data to its respective category;
- **RNN derivatives LSTM and GRU**: recurrent neural networks (RNN) work based on connections between nodes in a graph-like manner, exhibiting temporal dynamic behavior. They use their internal memory to process variable length input.

2.4 Sentiment analysis challenges

As one would expect, there are various challenges that are posed to this kind of research, since one is trying to develop a machine algorithm that will understand the infinitely complex and nuanced world of human emotions. These are just some of them:

Tone Tone can be difficult to interpret in a textual form. This can be a challenge even for humans themselves: have you ever had trouble understanding the tone of a text message? When it comes to sentiment analysis use cases, this problem can be aggravated by the sheer volume of text to be analysed, which can contain both subjective and objective phrasing.

Polarity Words of extreme sentiment such as "love" or "hate" are often easy to classify in terms of polarity, but there is a myriad of lukewarm words that become much harder to analyse, such as "not so bad".

Sarcasm The contradictory nature of sarcasm is quite difficult for machines to detect, since it is virtually indistinguishable from genuine statements in textual form.

Emojis Sentiment analysis is language-specific, and emojis can completely change the meaning and intention of a sentence when used.

Idioms It is difficult for a machine learning algorithm to identify or be familiar with figures of speech, especially as these vary from language to language, or even according to the author's geographical origins.

Multilingual sentiment analysis When the text author mixes different languages within the same text, often even in the same sentence, this makes the sentiment analysis job much more difficult, often compounding on the previous issues as well.

Audio-visual data Videos are quite different from textual data, and the task of transcribing them into such is often a challenge in and of itself, since it is necessary to keep as much of the context of the original media as possible in order to then accurately identify tone and intent.

Chapter 3

How it relates to Twitter trends

In this chapter, I intend to correlate everything that has been explained before about sentiment analysis to the Twitter platform, explaining how people are using sentiment analysis on tweets and what purposes that serves. I shall also present a small tool I developed for sentiment-related data visualization, as well as use it to explore some social, economic and political aspects of Twitter trends.

3.1 How sentiment analysis of Twitter trends is used by entities external to Twitter

Sentiment analysis of tweets and Twitter trends specifically has a number of different applications. Twitter has been quite stable in its position as one of the most relevant social networks and micro-blogging platforms of the last few years, especially given the very fast-paced rate of relocation to newly emerging social networks in the Web space.

Namely, we can point to these use cases:

- **Business**: companies use sentiment analysis applied to the Twitter space for business strategies. This particular use case shall be further detailed in the next section, but basically it allows these companies to assess the public's general sentiment towards their products or specific product;

- **Politics**: seeing as Twitter is a heavily politically charged network, tweet data can and is used to keep track of the public's political views. This can mean both theoretical ideologies, as well as how they feel about their national or local governments. This kind of analysis is also utilized to try to predict election results, as well as process concrete election outcomes;

- **Public actions**: interestingly enough, Twitter sentiment analysis is employed in identifying and understanding social phenomena, determining the general mood of the Web space and attempting to predict potentially dangerous or threatening situations (for example, banning a user that consistently threatens others).

3.2 Business and marketing

As the reader may have already gathered, sentiment analysis is immensely valuable for marketing and economics. It allows for an automatized way of understanding the public or the target audience's perception of a product or service.

There are multiple sources of opinionated text on a product that can be used as input or training data for sentiment analysis, such as product reviews, social media posts, Internet forums and customer support correspondence. The sum of all this information across the entirety of the Internet results in a massive amount of data that we could only ever hope to process with the aid of machines. The point of such processing is usually to achieve two goals: understand what users think about the product/service, and understand the state of the market for that same product/service.

Sentiment analysis can be decisive in choosing where the marketing efforts should be directed, allowing producers to understand their product's weakest and strongest attributes, as well as receive some performance indicators. It is an invaluable tool for market research.

3.3 My tool

The tool I developed is a small Web application called **APSEI: Twitter Trends**, and it is composed of two main pages. The first one, Count Tweets, allows the user to search for a keyword (whether that is a single word or an expression) and select a preferred language (optional) in order to obtain information about the number of tweets containing said expression in the last 7 days. This is, of course, useful for the detection and analysis of trend lifecycles. Altough this more closely related to the previous report I worked on, it will also be useful in this case, as will become clear further along.

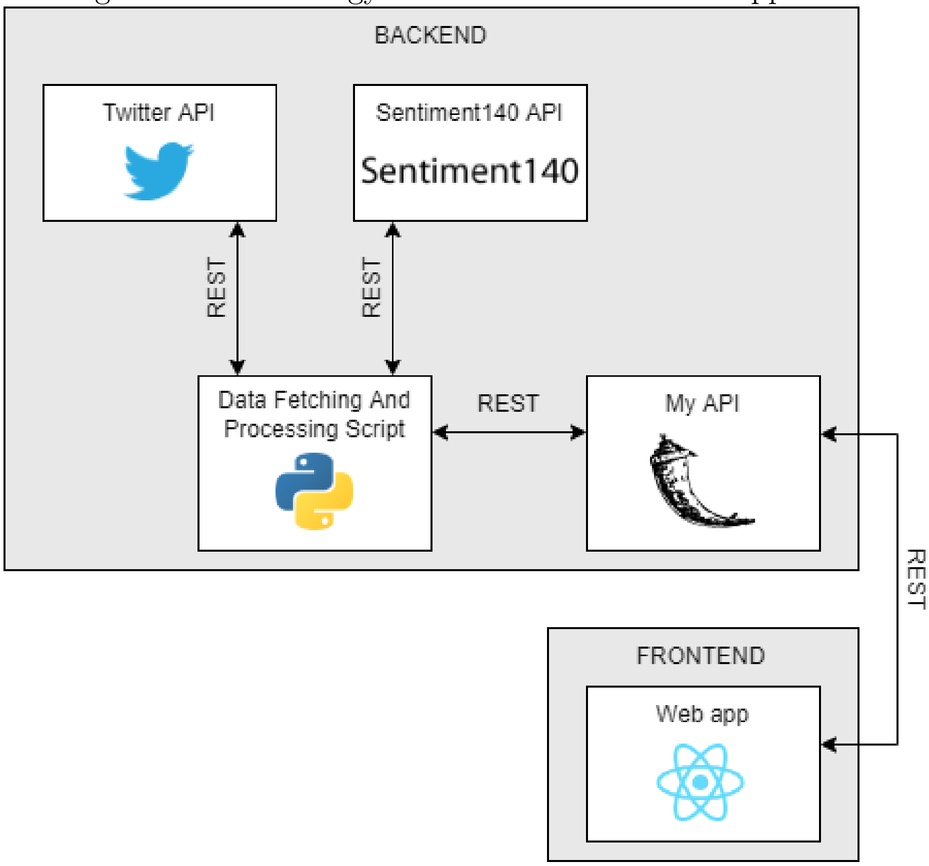
The second one is called Sentiment Analysis and, again, given a keyword and a preferred language, it displays information about a set of 100 random tweets from the last 7 days relating to their sentiment. This means the tweets are analyzed for emotional intent and rated as Negative, Neutral or Positive.

The GitHub repository for the project is publicly available and, since the application has not been deployed, a video demo was done as well. The reasons for not deploying this tool will be explained further ahead. In the `README.md` file of the repository, you will find a user manual if you wish to run the application in your computer. However, the video demo already explores all the app's functionalities, so I encourage you to take a look at that first.

3.3.1 Technology architecture

Concerning the technology architecture of the application, which is represented in Figure 3.1, the first thing one might notice is that there are both backend and frontend layers, but no database layer. This was a decision that I made for the sake of simplicity, as it is quite a small app, and destined for a demo that only makes sense within the scope of this report. More than that, it does not deal with a huge amount of data, as shall be explained momentarily, and, thus, saving the data it does require in local files generated by the backend scripts is quite enough.

Figure 3.1: Technology architecture for the Web application.



Frontend

In terms of frontend, the Web application's interface was developed using ReactJS and, as has been stated before, is constituted by two pages which encompass all the functionality, plus a very simple homepage for usability's sake. The communication between the frontend and my own API was done in a RESTful manner.

Backend

Twitter API

The backend is composed of three distinct APIs, all of which I will explain, and a Python script for data fetching and processing. The first API I used was the Twitter API. There was a not-so-simple authentication process that I went through in order to be able to use. Firstly, I had to own a Twitter account (which I already did) and, with that, apply for a Developer account, which was quite a straightforward process - I simply had to fill out some extra information about myself. This gave me access to my own dashboard in Twitter's Developer Portal. There, I created a project for this assignment and generated my own Bearer Token, along with some other secret credentials. Whenever I make calls to the Twitter API, I have to use this token for the sake of authentication.

For security reasons, the Twitter API does not allow calls from a browser and, as such, I could not call it directly from the frontend. Instead, I call it in my Python script, which then processes the data and writes it to local files, so that the frontend may read them. For obvious security concerns, I did not put my Bearer Token in cleartext in this Python script for the API calls. Instead, I saved it as an environment variable called `TWITTERTOKEN` in my operating system, and the Python script fetches that variable. It is because of all these authentication and security hurdles that I chose not to deploy the application - if I did, users would either have to use my personal authentication token in order for requests to work, and could quickly exhaust it, or they would have to request their own, in which case the deployment's easy access aspect would be damaged anyway.

This API was used for both pages of the application. The `count` endpoints of the API were useful for the Count Tweets page, while the `search` endpoints allowed me to fetch 100 random tweets with the input keyword on the Sentiment Analysis page.

Sentiment140 API

The second API used for the application was the Sentiment140 API. This API was built specifically for sentiment analysis of Twitter text posts, although you may provide other texts. In your requests, you provide the text to be analysed and some other optional parameters, such as language and topic. Providing a topic is very useful for the sentiment analysis process: if you specify that the topic of the tweet is "anger", the algorithm won't take the mentions of the word "anger" in the tweet as input for the sentiment analysis, since that is the tweet's subject matter and not an expression of that emotion. For each tweet, it returns a numeric value that can be either 0 for negative polarity, 2 for neutral or 4 for positive, making it fine-grained analysis.

This API also allows for batch processing of a set of texts/tweets, which is the option I used. If you provide an ID for each text, the response will use those IDs to attribute each polarity result to its corresponding text.

This API was used for the Sentiment Analysis page, in order to extract the sentiment of the 100 random tweets selected containing the user's input keyword.

Python script

I then developed a Python script for fetching and processing data from the two APIs detailed above. Again, the communication between these entities was done in a RESTful way. After receiving the necessary API data for whatever request was being handled at the time, this script saved the data locally by writing it to a file that the frontend could then read. I shall not explain the code line by line here, but I will give a small overview of what purpose each method in this script serves:

- `get_day(days_ago)`: takes `days_ago` as input, that being the number of days ago desired (0 days ago means today) and returns a string of the corresponding date in the correct format for requests to the Twitter API;

- `auth()`: fetches the value of the `TWITTERTOKEN` variable from the environment variables;

- `create_headers(bearer_token)`: creates the correct authentication headers for requests to the Twitter API (using the environment variable `bearer_token`);
- `general_count(keyword)`: prepares URL and parameters of the request to the Twitter API for the number of tweets containing `keyword` per day over the last 7 days;
- `day_count(keyword, start_date, end_date)`: prepares URL and parameters of the request to the Twitter API for the number of tweets containing `keyword` per hour within a specific timespan that goes from `start_date` to `end_date`. I used this to see the number of tweets per hour over the last 3 days (counting from when the user makes the request on the frontend);
- `general_tweets(keyword)`: prepares URL and parameters of the request to the Twitter API for 100 random tweets containing `keyword` from the last 7 days;
- `connect_to_endpoint(url, headers, params, next_token = None)`: makes an HTTP GET request based on the URL, headers and request parameters. The `next_token` input variable has to do with request response pagination, but I had no use for it;
- `keyword_7days(keyword, language)`: processed user input from the interface in order to personalize the request built in `general_count(...)`;
- `keyword_days(keyword, language, days_ago)`: processed user input from the interface in order to personalize the request built in `day_count(...)`;
- `batch_sentiment(keyword, language)`: processed user input from the interface in order to personalize the request built in `general_tweets(...)`;
- `sentiment(keyword, language)`: calls the Sentiment140 API in order to perform sentiment analysis on the data processed in `batch_sentiment(...)`, storing the result in a file.

My API

This is a simple API I developed using Flask that allowed me to establish communication between my backend and frontend modules. Its endpoints are as follows:

- `/keyword_7days`: a GET endpoint that returns information on the tweet counts per day that include the last searched keyword, over the 7 days prior to that search;
- `/keyword_daysago`: a GET endpoint that returns information on the tweet counts per hour that include the last searched keyword, over the 3 days prior to that search;
- `/sentiment`: a GET endpoint that returns information on the sentiment analysis of 100 random tweets that include the last searched keyword, over the 7 days prior to that search;

- `/new_search`: a POST endpoint for new keyword searches in the Count Tweets page;
- `/new_sentiment_search`: a POST endpoint for new keyword searches in the Sentiment Analysis page.

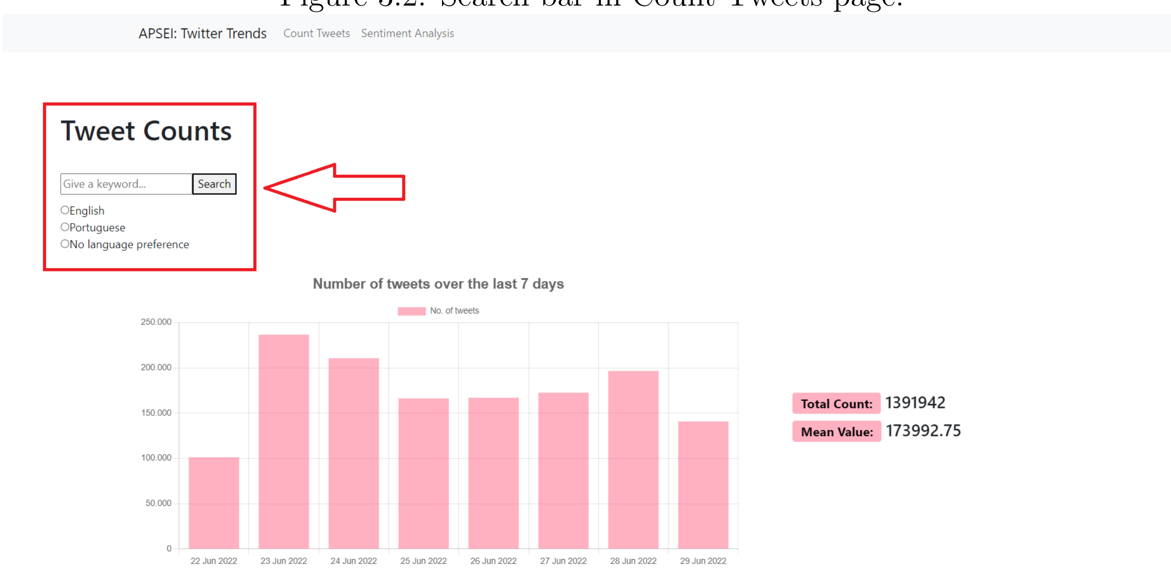
3.3.2 Usage and direct analysis of Twitter trends

This section of the report is mostly a written down version of the Web application's video demo. If you would like to watch that first, click here. I strongly recommend you do so, as the demo contains a full explanation of how to work with the tool, which is not provided here, and because the visual nature of it is inherently better for the good understanding of what the tool provides.

Count Tweets page

On the top left of this page, we have a search bar with some language options (see Figure 3.2). This allows us to give a word or an expression as input, as well as select the language in which we would like our query to focus on. If you fill that out and click on the search button, this will trigger a query that will display the new retrieved information on the graphs.

Figure 3.2: Search bar in Count Tweets page.

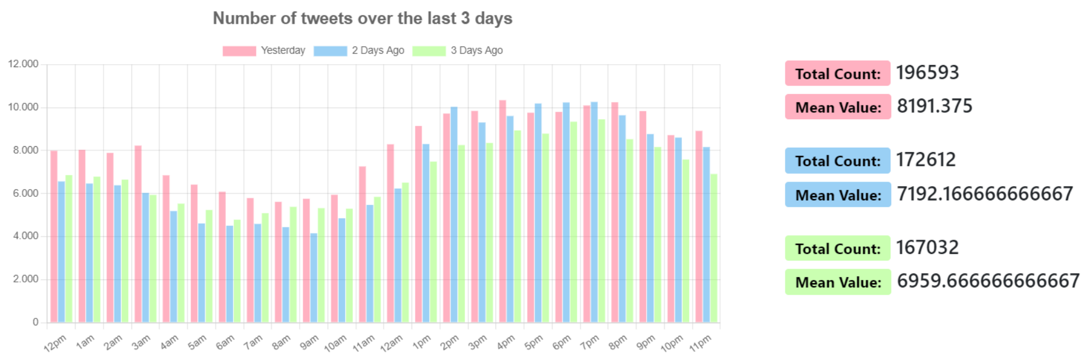


The first graph provided in this page, which is the one visible on Figure 3.2, gives information about the number of tweets in the selected language and containing the input word or expression for each of the last seven days, as well as today.

If you scroll down to the second graph (see Figure 3.3), you will see that it displays similar information. This graph tells us how many tweets in the selected language and containing the input word or expression were posted in each hour of the last three days.

As the viewer might have seen on the news, there has been quite a lot of debate online recently concerning abortion laws and women's health in the United States,

Figure 3.3: Second graph in Count Tweets page.



following the overturning of Roe vs. Wade by the Supreme Court. This is a technical report on sentiment analysis and not a personal political piece and, as such, it will of course remain neutral on the subject matter. I, the author, simply chose this subject as it was the most relevant Twitter trend happening at the moment and, as such, one that would bear interesting fruits in terms of trend analysis and sentiment analysis.

That being said, let us now search for "abortion", selecting the English language. As we can see in the Y axis of the graph in Figure 3.4, the scale grew considerably due to the discussion and debate online being quite more relevant and on a worldwide level. It is also very clear that the overturning of Roe vs. Wade happened on the 24^{th}, as that is when the clear spike in conversation happens. Additionally, you will notice that the trend's lifecycle becomes crystal clear with the graph's help. The first couple of days since the law's overturning (the 24^{th} and 25^{th}) are made up of intense discussion and, from that point on, the numbers slowly descend and will eventually level down to what they looked like before the controversy. Even though the current day's data is not complete, as we are making this query while the day is not yet over, we can safely assume that its count will still be lower then yesterday's by the day's end. This data reflects the Twitter trend lifecycle that was discussed on the previous report I was a part of.

Scrolling down to the second graph (see Figure 3.5), we can see that although different days had different tweet counts, we still have a distinct pattern that repeats itself every day, with some hours being more popular than the rest. The dip that we see every day from around 5AM to 11AM could be due to most of the American population being asleep at that time, as the graph hours are relative to our own timezone, and not any of the timezones in American territory.

Sentiment Analysis page

With this worldwide discussion scale, we see that the sentiment analysis results become a bit more divisive than if we were to search, for example, for the University of Aveiro (see Figure 3.6). For starters, there is a lot more negatively charged content, although neutral tweets are still the most prevalent. Furthermore, we can see that the number of positive tweets is approximately the same as the number of negative ones. This might

Figure 3.4: Searching for the keyword "abortion".

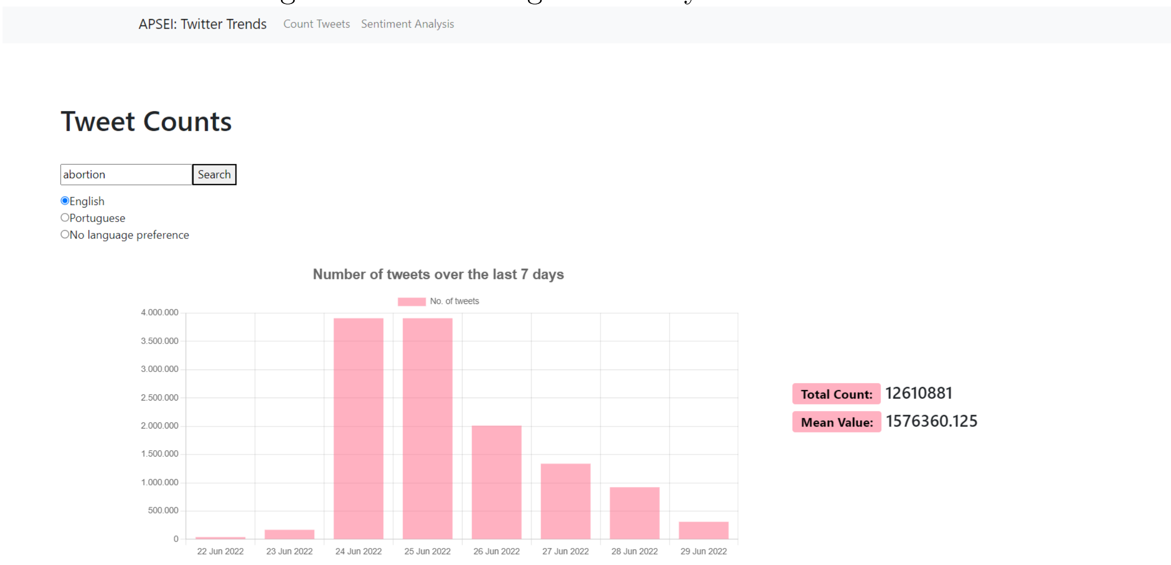
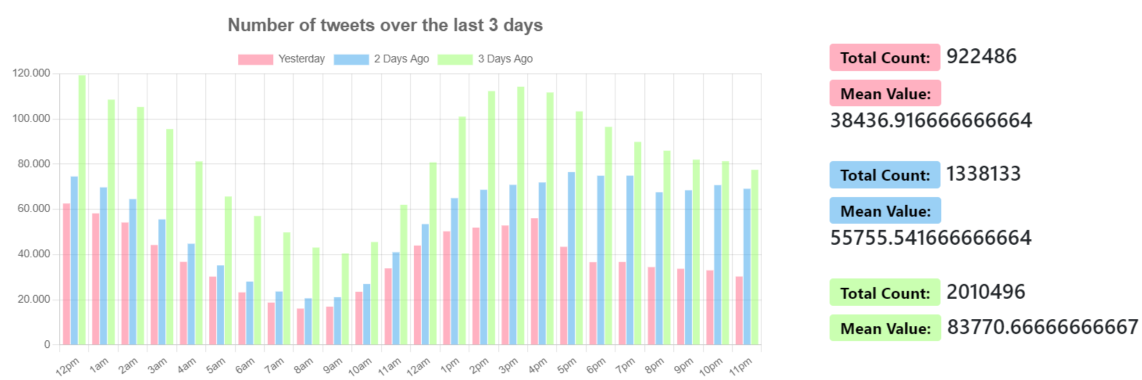


Figure 3.5: Searching for the keyword "abortion" (part 2).

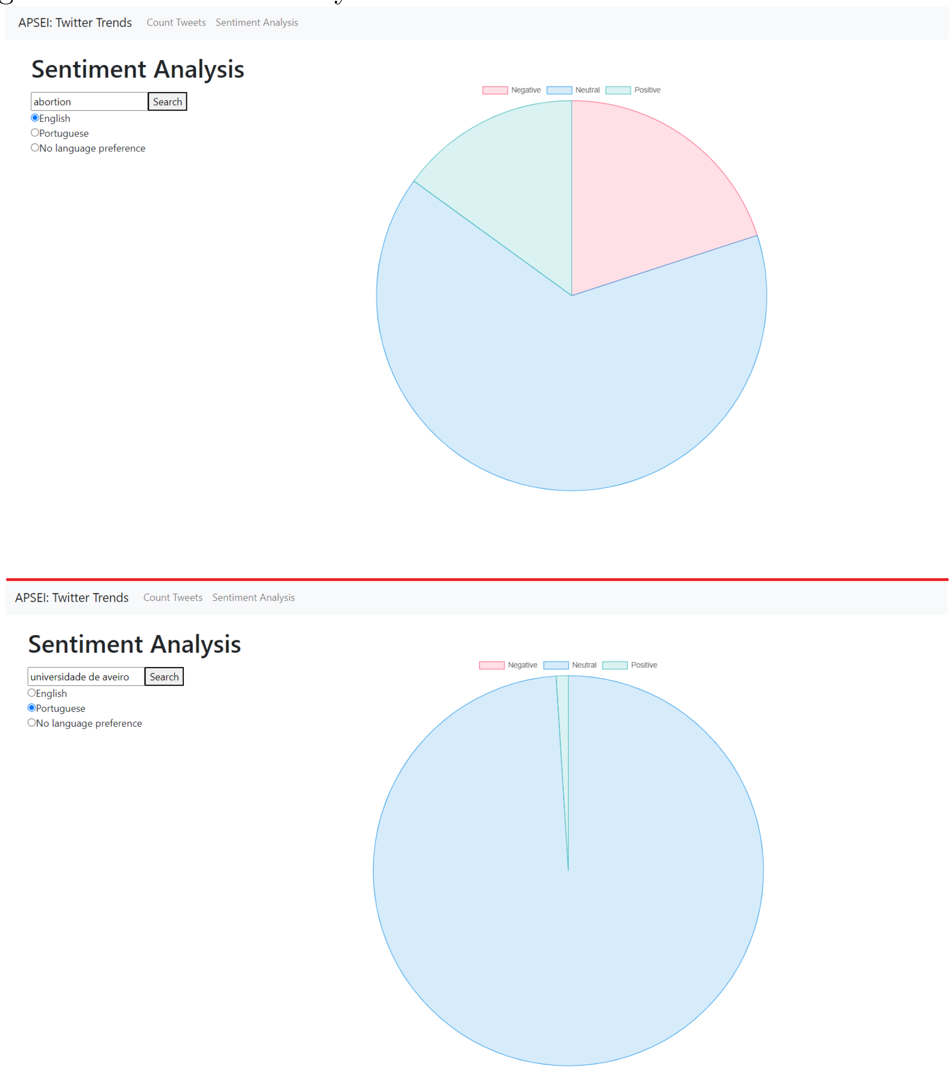


be indicative of an evenly split set of opinions in the general population, meaning that there might be roughly as many people in favour of the Supreme Court's decision as there are people against it.

If we click multiple times on the search button for the same keyword and language, we can check how the results vary with different sets of 100 random tweets (see Figure 3.7). As we can see, the ratio of positive to neutral to negative tweets does not undergo any drastic changes, although small changes are, in fact, visible.

Let us note that there are many tweets on this topic that cannot even possibly be chosen for this analysis, as they do not contain the word "abortion" itself, but discuss this situation regardless. This is because of the way I built this tool, such that it will always look for tweets containing the input keyword. This was done mostly because I was using the resources available to me, such that I was limited to what the Twitter API's developer version offered me in terms of endpoints. However, I do believe that searching for the trend's most prevalent keywords will yield satisfactory results that

Figure 3.6: Sentiment analysis of "abortion" vs. "Universidade de Aveiro".



are, in fact, representative of reality.

As an additional effort in the sentiment analysis of this trend, let us search for "Roe vs. Wade" in the search box and see what happens. The results are similar (see Figure 3.8), which is what one would wish in this situation. It means that the keyword search might be satisfactory enough for the intended purposes.

Let us finish by searching for a stereotypically positive word, and checking if the sentiment analysis reflects that. Then, let's do the same for a word typically associated with negative feelings and repeat the experiment (see results in Figure 3.9). For the first scenario, I chose the word "sunshine". As we can see, the graph turned to an overall positive sentiment, as was to be expected. Now for the second case, let's search for the word "sad". Again, and even though the word "sad" itself was not taken into account when calculating the polarity of the tweets, it was mostly prevalent in tweets with negative sentiment.

Figure 3.7: Same search variations.

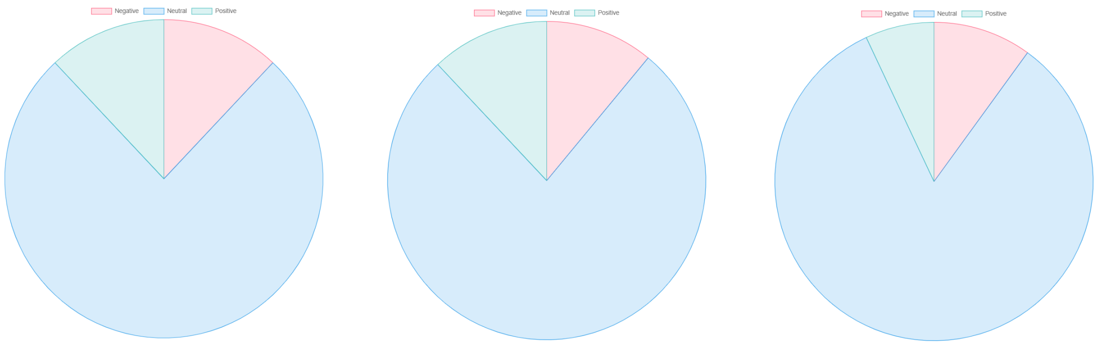


Figure 3.8: Sentiment analysis search for ”Roe vs. Wade”.

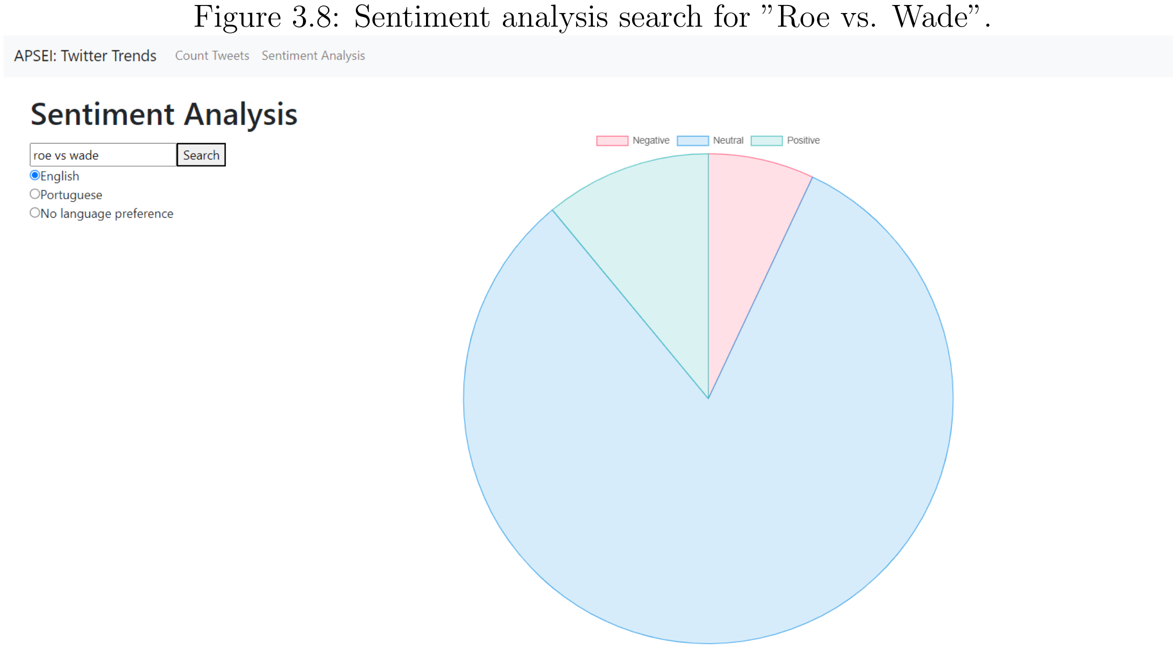
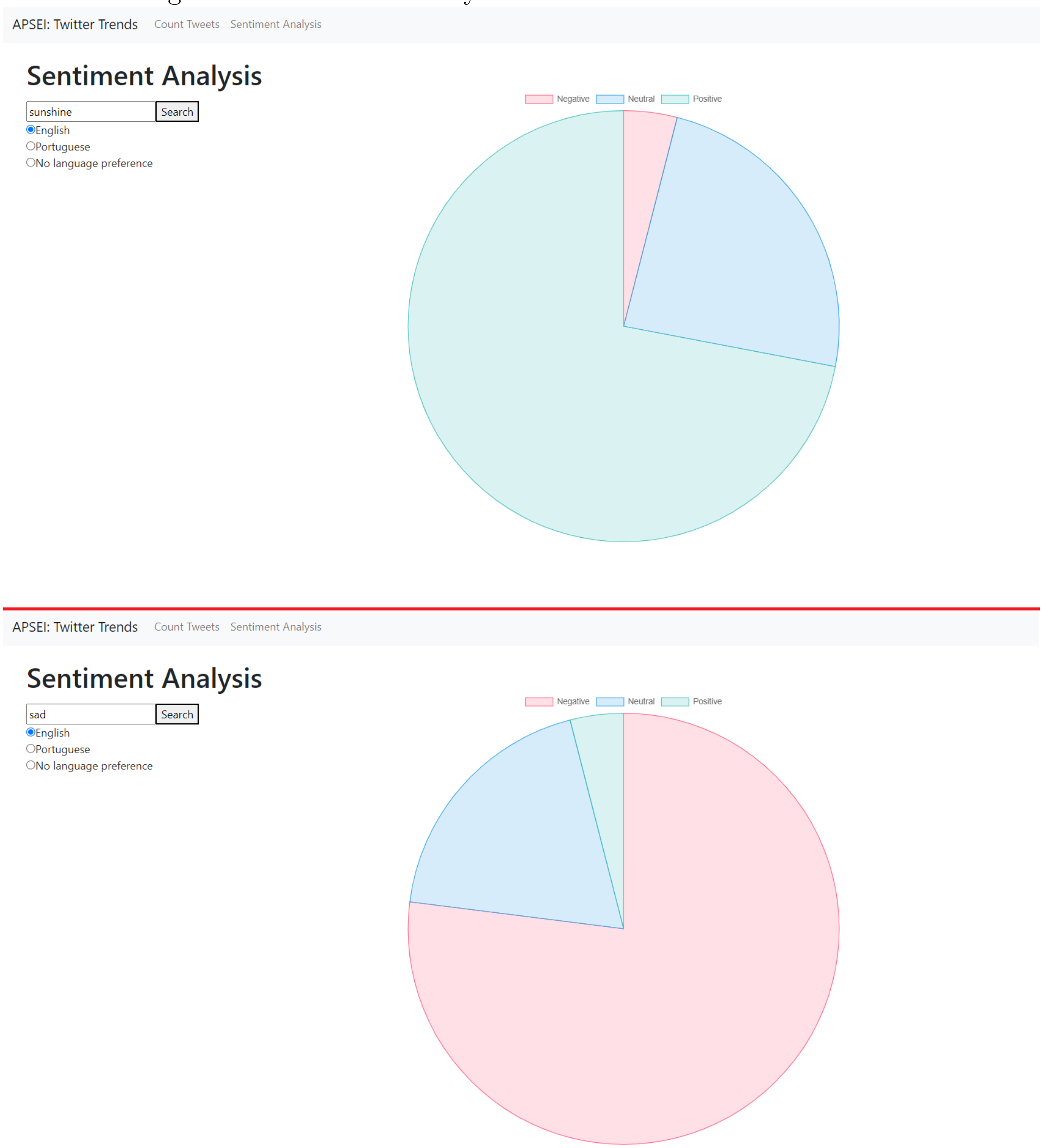


Figure 3.9: Sentiment analysis search for ”sunshine” vs. ”sad”.



Chapter 4

Conclusion

To sum up, sentiment analysis is not only quite an interesting area of research, but also uniquely useful in the modern socioeconomic landscape. With the ever growing amount of data generated and circulating online, it has become more than helpful - in fact, it is now necessary - for a good functional assessment of market needs and target audience moods.

On a more personal note, this assignment was immensely exciting for me as a student. First and foremost, it truly spiked my curiosity and desire to learn: much more than the economic, practical and useful side of sentiment analysis, which would perhaps be more enticing for a student of Economics, it was the existential nature of it that attracted me so much. Much of my previous interest in the area of Artificial Intelligence came from its daring character. AI research strives to push the limits of what computers can do, but it is clear that, along with the many advantages of technological evolution, humans are also using it to try and know more about themselves, much as they do with Art.

Neural network models in Machine Learning are an attempt to model a computer system after the human brain, but certainly they are also an attempt to know more about the brain itself. If we can build an artificial brain that works like ours, maybe we will have ours figured out.

Maybe if we build a system that identifies and understands human emotion from a natural language text in a perfectly accurate manner, we will be on the right track to understand our emotions in a way we did not manage before. The human endeavour is painful and lonely, but it is also beautiful, and, despite common belief, people express themselves through Science as much as they do through Art.

I have always believed there is lyricism in logic.

Bibliography

[1] Twitter Sentiment Analysis – Introduction and Techniques
Available online (29/06/2022):
https://www.digitalvidya.com/blog/twitter-sentiment-analysis-introduction-and-techniques/

[2] What Is Sentiment Analysis: Definition, Key Types And Algorithms
Available online (29/06/2022):
https://theappsolutions.com/blog/development/sentiment-analysis/

[3] Types Of Sentiment Analysis And How Brands Perform Them
Available online (29/06/2022):
https://www.analyticsinsight.net/types-of-sentiment-analysis-and-how-brands-perform-them/

[4] Sentiment Analysis Challenges And How To Overcome Them
Available online (29/06/2022):
https://www.repustate.com/blog/sentiment-analysis-challenges-with-solutions/

[5] Rule-Based Sentiment Analysis in Python
Available online (29/06/2022):
https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/

[6] 5 Types of Classification Algorithms in Machine Learning
Available online (29/06/2022):
https://monkeylearn.com/blog/classification-algorithms/

[7] Supervised vs. Unsupervised Learning: What's the Difference?
Available online (29/06/2022):
https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

[8] Linear Regression for Machine Learning
Available online (29/06/2022):
https://machinelearningmastery.com/linear-regression-for-machine-learning/

[9] Naive Bayes Classifiers
Available online (29/06/2022):
https://www.geeksforgeeks.org/naive-bayes-classifiers/

[10] Support-Vector Machine
Available online (29/06/2022):
https://en.wikipedia.org/wiki/Support-vector$_{m}achine$

[11] Recurrent Neural Network
Available online (29/06/2022):
https://en.wikipedia.org/wiki/Recurrent$_{n}eural_{n}etwork$