**Web and Social Media Analytics**

***7 BUIS025W.3***

**Hasan Enes Guray**

**Student ID: 19489124**

**Word Count:**

**SOCIAL MEDIA COURSEWORK  
ELON MUSK**

|  |
| --- |
| *Hasan E. Guray* |
|  |
| MSc Data Science and Analytics student |
| University of Westminster |
| w1948912@my.westminster.ac.uk |

ABSTRACT

This research paper presents a case study focused on analysing public perceptions of Elon Musk after his latest decision to change Twitter to X. Leveraging a dataset comprising 2000 Reddit comments, the study delves into five primary discussion topics and employs three distinct sentiment analysis methods to gain comprehensive insights. The findings shed light on the sentiment surrounding the businessman’s portrayal and offer valuable implications for understanding general reception in society.

**Keywords:** Elon Musk, Twitter, Reddit comments, Sentiment analysis, Topic modelling.

# PERSON SELECTED

Elon Musk, a prominent magnate, holds key roles like SpaceX's founder-CEO, Tesla's CEO, Twitter's owner-CTO, Boring Co.'s founder, Neuralink and OpenAI's co-founder, and Musk Foundation's president (<https://www.britannica.com/biography/Elon-Musk> accessed on 3 August 2023). Worth $248.8B by July 2023, he is the world's wealthiest (<https://nypost.com/2023/07/06/musk-zuckerberg-worlds-richest-852b-wealthier-in-2023/> accessed on 3 August 2023). Musk, controversial yet admired, surprises with impactful ventures. A recent move involved Twitter's rebranding to "X," announced uniquely on the platform. With 151.3M followers and 396 followings on X (as of August 4, 2023), Musk leads in Twitter followers (<https://www.brandwatch.com/blog/most-twitter-followers/> accessed on 2 August 2023).

# Data COLLECTION

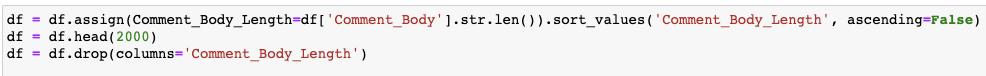
In this tech-driven era, information spreads swiftly through social media, aiding exchanges. To analyse Elon Musk's image and recent feats, Reddit was chosen, leveraging its API and traffic. As the 10th most visited site as of June 2023 (<https://www.semrush.com/website/top/> accessed on 29 July 2023), Reddit's global appeal is noteworthy. Data were collected from July 24th to 30th, 2023, capturing Twitter's CEO-initiated changes and their societal impact.

Employing Python's pandas, praw, and datetime libraries, an extensive dataset comprising 5974 comments (Figure 1) was meticulously compiled by leveraging Reddit's API support. To enhance the meaningfulness of the results, a filtering process was executed, focusing on the 2000 comments boasting the longest character count (Figure 2).

A screenshot of a computer

Description automatically generated

**Figure 1** *This is a screenshot of Python code to scrape Reddit comments about Elon Musk.*



**Figure 2** *This is a screenshot of Python code to filter the longest 2000 Reddit comments about Elon Musk.*

Using this dataset, researchers can explore Elon Musk's recent work perception on Reddit, assessing sentiment and popularity within the specified timeframe. Reddit's user-generated content provides insights and data-driven analysis chances. Top-level comments were chosen to reduce ambiguity from user interactions. Exclusions like system-generated and user-specific comments upheld data integrity. This approach yielded a comprehensive dataset with attributes like "Comment Body," "Author," "Upvotes," "Awards," and "Date" for analysis (Figure 3).

A screenshot of a computer

Description automatically generated

**Figure 3** *This is a screenshot of Pandas dataframe, which consists of Reddit comments about Elon Musk.*

Before analysing, raw user-generated comments need thorough data cleaning. The initial step was converting the "Date" column to datetime format (Figure 4) to make time series data suitable for analysis.

A screenshot of a computer code

Description automatically generated

**Figure 4** *This is a screenshot of Pandas dataframe’s column details, consists Reddit comments about Elon Musk.*

To enhance topic and sentiment analysis, raw comments were preprocessed using Python's contractions library, which standardizes text by resolving contractions, ensuring a coherent representation (Figure 5).

A screenshot of a computer

Description automatically generated

**Figure 5** *This is a screenshot of a Python formula to fix contractions.*

Python's "re" library standardized textual attributes (price, size, volume, weight) to ensure consistent analysis (Figure 6), mitigating disparities and preserving integrity.

A screenshot of a computer program

Description automatically generated

**Figure 6** *This is a screenshot of a Python formula to substitute patterns.*

Hyperlinks are prevalent in comments, guiding users to external resources for various purposes. To enhance analysis efficacy, Python's "re" library was used to consolidate and standardize these links, promoting coherence (Figure 7). This process ensured more consistent and interpretable investigative outcomes.

A screenshot of a computer

Description automatically generated

**Figure 7** *This is a screenshot of the Python formula to replace http links.*

In this phase of the investigation, comments comprising only a singular hyperlink were identified and subsequently omitted from the dataset, as such comments were deemed inconsequential to the analytical process (Figure 8).

A close-up of a white background

Description automatically generated

**Figure 8** *This is a screenshot of the Python formula to filter out the comments comprising only a singular hyperlink.*

Though not readily visible on Reddit, some dataset comments contain "\n" for line breaks. Python's "replace" method was utilized (Fig. 9) to remove these, enhancing dataset coherence, precision, and clarity for subsequent analyses.

A screenshot of a computer error

Description automatically generated

**Figure 9** *This is a screenshot of the Python formula to remove line breaks.*

Typographical errors in comments stem from rushed writing, social media abbreviations, and more, risking analysis accuracy. Python's TextBlob library addressed this by correcting misspellings and standardizing words. Yet, caution was needed when processing full comments to avoid unintentional errors (Figure 10).

A screenshot of a computer

Description automatically generated

**Figure 10** *This is a screenshot of the Python formula to correct the words.*

A precise approach was used to identify mistranslated words before individually processing them with TextBlob (Figure 11). This improved accuracy by preventing false corrections and enhancing analysis results.

A screenshot of a computer program

Description automatically generated

**Figure 11** *This is a screenshot of the Python formula to correct the words.*

For effective topic model analysis, crucial preprocessing involved converting comments to lowercase using Python's "lower()" function (Figure 12), ensuring uniformity and consistency. As the interplay between uppercase and lowercase letters holds emotional significance in sentiment analysis, applying a lowercase transformation to sentiment analysis was avoided.

A screenshot of a computer

Description automatically generated

**Figure 12** *This is a screenshot of the Python formula to convert to lowercase all of the letters.*

Tokenization is pivotal in enhancing topic model analysis and structuring unprocessed text for pattern discernment. Python's "nltk" library enables tokenization (Figure 13), yielding robust topic modelling by aligning with preprocessing needs, enhanced accuracy, and interpretability.

A screenshot of a computer

Description automatically generated

**Figure 13** *This is a screenshot of the Python formula to convert to tokenize the words.*

Excluding stopwords, devoid of substantive meaning, is vital in topic model analysis. Eliminating these common non-informative words reduces noise, improves pattern identification, and enhances topic coherence. Python's "nltk" library aided this task (Figure 14). While "not" is a stopword, its transformative role retains it. This resourceful language processing tool systematically removed stopwords, purifying data for effective topic scrutiny. Thus, it extracted more refined, cohesive topics, heightening latent structure visibility within the textual corpus.

A screenshot of a computer

Description automatically generated

**Figure 14** *This is a screenshot of a Python formula to remove stopwords.*

Emojis and punctuation hold paramount importance in comments. For effective topic model analysis, removing them is crucial to reduce noise, enhance coherence, and improve interpretability. Expunging emojis and punctuation yields a standardized, refined textual representation, focusing on content and enhancing efficiency. The "re" and "string" libraries in Python achieved this (Figure 15), aligning with data refinement for rigorous topic analysis. This prep task aided in identifying meaningful lexical elements and establishes a strong foundation for coherent topic extraction from the corpus.

A screenshot of a computer program

Description automatically generated

**Figure 15** *This is a screenshot of a Python formula to remove punctuations and emojis.*

Lemmatization is vital in topic model analysis, normalizing words to base forms and enhancing interpretability. This process reduces vocabulary dimensionality, producing more precise topic outcomes that capture core themes. Implemented through Python's "nltk" library (Figure 16), lemmatization refined data significantly, utilizing linguistic insights for nuanced semantic understanding. This approach strengthened topic analysis, yielding robust and coherent modelling outcomes aligned with the textual corpus.

A screenshot of a computer

Description automatically generated

**Figure 16** *This is a screenshot of the Python formula for lemmatisation.*

To preserve analytical integrity and clarity by eliminating redundant data, especially repeated comments from the same users, the use of Python's "drop\_duplicates()" function was essential (Figure 17). This step efficiently identified and removed duplicates, ensuring a streamlined and trustworthy dataset. The exclusion of duplicates enhanced data quality and bolsters the accuracy and credibility of subsequent data science insights.



**Figure 17** *This is a screenshot of the Python formula to remove duplicates.*

After comprehensive data cleaning, "na" values replaced empty comments. Python's "dropna()" function (Figure 18) ensured dataset integrity by removing such entries, and refining the dataset. Eliminating uninformative or incomplete data bolstered reliability, enhancing accuracy and insight in subsequent analyses.

A close up of a sign

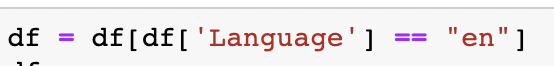
Description automatically generated

**Figure 18** *This is a screenshot of the Python formula to remove na records.*

Comments' linguistic makeup influences analysis, tailored to a language context. As English predominates, purging non-English comments via Python's "langdetect" library (Fig. 19) was vital, retaining 1932 rows. Google Translate validated classifications due to langdetect's potential misinterpretation.

A screenshot of a computer program

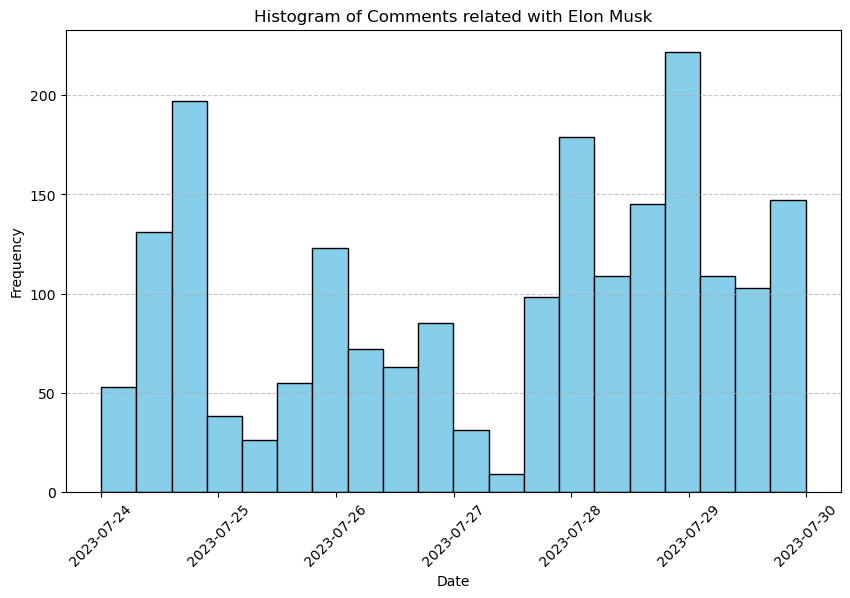
Description automatically generated



**Figure 19** *This is a screenshot of the Python formula to keep just English comments.*

# EXPLORATORY ANALYSIS

The dataset covers a week, starting from Twitter's transition to X on July 24-30, 2023, during which a noticeable augmentation has been observed (Figure 20). It included the 2000 longest comments within this period, representing users affected by mobile app changes. On July 24, the day the decision was announced, an increase was observed in the number of long comments (<https://www.cnbc.com/2023/07/24/read-ceo-linda-yaccarinos-message-to-twitter-staff-about-the-rebrand-to-x.html> accessed on 4 Aug 2023).



**Figure 20** *This is a histogram of the comments count related to Elon Musk on Reddit.*

Diverse contributors were essential for robust conclusions. Comment distribution analysis revealed a heterogeneous dataset, with the most prolific author contributing only 6 comments (Figure 21).

A graph of a number of authors

Description automatically generated

**Figure 21** *This is a bar graph of the comments count of the authors related to Elon Musk on Reddit.*

Using upvotes and awards, a top-comment analysis revealed the comment with the phrase "I will never understand how someone posts a picture of a video and then is satisfied the message was conveyed" amassed 7553 upvotes and 13 awards. In terms of language distribution, 1932 comments were in English, accounting for 96.8% of the total dataset. French and Italian were the subsequent prominent languages (Figure 22).

A green and white rectangular object

Description automatically generated

**Figure 22** *This is a bar graph of the comments count of the languages related to Elon Musk on Reddit.*

Analysing the average comment length, Italian comments emerged as the most extensive, with an average of 182.15 characters, while French ranked second with 177.47 characters and English ranked third with 161.18 characters (Figure 23).

A graph of blue rectangular bars

Description automatically generated

**Figure 23** *This is a bar graph of the average comment length of the languages related to Elon Musk on Reddit.*

Examining term frequency aids theme discovery, insight extraction, and anomaly detection. Notable terms revealed included "twitter", "elon", "people", "like", "size", and "musk”, while "size" related numerically. Additionally, within the encompassing corpus, the lexemes "x", "company", and "tesla" held positions among the upper echelon of the 20 most recurrently reiterated terms. Common noun phrases were "elon musk", "free speech", "social medium", "yes men", and "dark mode". The “social media” phrase was changed to “social medium” in lemmatisation process. "yes men" became a nickname for Elon Musk among users. These phrases and words were associated with Elon Musk.

A close up of words

Description automatically generated

**Figure 24** *This is a wordcloud of the most common words in the comments related to Elon Musk on Reddit.*

# TOPIC MODEL ANALYSIS

Topic modeling is key for identifying themes in Reddit comments. It aids exploration and analysis but has limitations. Combining it with sentiment analysis and varied approaches improves insights. Latent Dirichlet Allocation (LDA) is a popular technique, excelling in coherent, scalable topic discovery. LDA's adaptability and integration in Natural Language Processing enhance its appeal. Employing LDA with Python's nltk and gensim libraries, five distinct topics were found (Figure 25).

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Figure 25** *This is a Python code to build LDA topic modelling.*

Topic 0 was related to "Speculation and Curiosity" in that Reddit users speculated on Twitter's transformation into X by Elon Musk, discussing its size, impact, and implications. Topic 1 was about "Elon Musk's Role and Company" that users analysed Musk's influence in the transformation, its impact on his company, and strategic advantages. Topic 2 was relevant to "User Engagement and Money" in that discussions explored engagement features, financial aspects, and Musk's vision alignment. Topic 3 was "Comparisons and Musk's Image" where users compared the transformation with Musk's ventures, discussing its reflection on his image. Topic 4 was linked to "Musk's Vision and Company Direction" which focused on Musk's platform vision and its alignment with company goals.

According to the available dataset, the comments took place the most in the "User Engagement and Money" topic with 23.7% (Figure 26). It was essential to subject these topics and their corresponding keyword weights to rigorous testing with a more extensive dataset to ensure the establishment of more robust and effective thematic representations.

A pie chart with numbers and a number of people

Description automatically generated with medium confidence

**Figure 26** *This is a pie chart to show the topic distribution of the Reddit Comments about Elon Musk.*

# SENTIMENT ANALYSIS

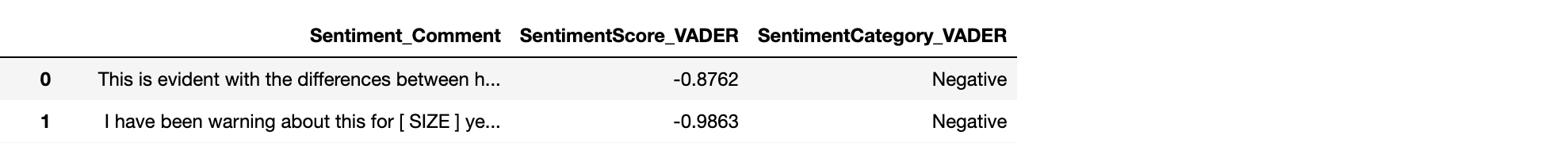
Sentiment analysis plays a pivotal role in social media modelling, extracting and quantifying sentiments from user-generated content like Reddit comments. Scrutinizing sentiments provides insights into public perceptions, emotions, and responses, such as reactions to Elon Musk's decisions, benefiting businesses, businessmen, and researchers in understanding reception, trends, and data-driven choices.

Using diverse sentiment analysis methods across disciplines requires a systematic approach to choosing the best fit. This involves trying techniques, evaluating performance, and integrating with machine learning models. VADER (Valence Aware Dictionary and Sentiment Reasoner) is a prominent model for such analysis, tailored for social media. It categorizes text as positive, negative, or neutral sentiment based on emotional tone.

VADER with Python's nltk library analysed the dataset (Figure 27), establishing sentiment categories using scores. Comments with scores ≥ 0.05 were positive, while ≤ -0.05 were negative. Much of the dataset fell from 0.0 to 0.1 (Figure 28).

A screenshot of a computer program

Description automatically generated



**Figure 27** *This is a Python code to build VADER sentiment analysis.*

A graph of blue bars

Description automatically generated

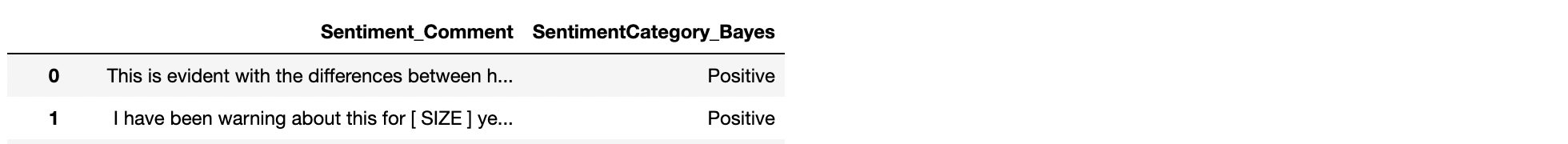
**Figure 28** *This is a histogram to show the VADER sentiment analysis score breakdown.*

A closer examination of sentiment extremes showed VADER's success in detecting strong positive and negative emotions. Yet, the dataset's prevalence at sentiment borders requires scrutiny. For instance, the highest-scored comment had impactful praise for Musk, boosting its score. In contrast, the lowest-scored comment's negative tone stemmed from a key sentence about Tesla's finances. VADER excels at extreme sentiments but demands careful handling near boundaries.

TextBlob's sentiment classifier, a pre-trained model on movie reviews, adeptly categorizes text sentiment as positive, negative, or neutral through specific words. It was employed with Python's textblob and nltk libraries (Figure 29).

A screenshot of a computer program

Description automatically generated



**Figure 29** *This is a Python code to build TextBlob’s sentiment classifier.*

In contrast to VADER, TextBlob provides distinct positive and negative sentiment scores, signifying specific emotions at a 0.6 threshold. TextBlob's multi-score approach differs from VADER's single-score output in that 1070 comments differed emotionally from VADER, demanding thoughtful use for insightful sentiment analysis. Stanford University's SocialSent method analyses 1900-2000 word corpus with predetermined scores (<https://nlp.stanford.edu/projects/socialsent/> accessed on 31 July 2023). The sentiment score is derived by summing the preassigned coefficients and then dividing them by the total number of words in the text (Figure 30). Upon conducting this analysis, it was observed that a significant proportion of the dataset resulted in sentiment scores proximate to 0 (Figure 31).

A screen shot of a computer code

Description automatically generated

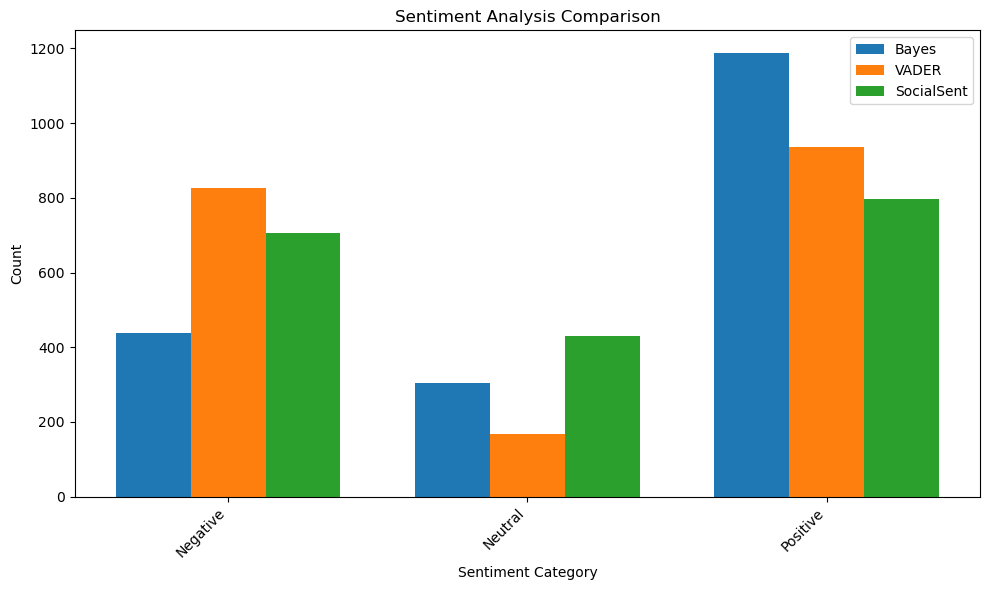
**Figure 30** *This is a Python code to build SocialSent sentiment analysis.*

A graph of a person with blue bars

Description automatically generated with medium confidence

**Figure 31** *This is a histogram to show the SocialSent sentiment analysis score breakdown.*

In SocialSent analysis, the top sentiment score came from a comment praising Elon's action on Twitter, while the lowest score stemmed from a negative exclamation. SocialSent's compact vocabulary emphasizes words' emotional weight, making certain words impactful. This reveals the divergence between VADER and SocialSent, influencing sentiment analysis's effectiveness and interpretation. Such nuances highlight the importance of understanding each technique's specifics for accurate insights from user-generated content. Evaluating all three methods, sentiment categorization varied (Figure 32). Although positive comments dominate for all three analyses, the difference between positive comments and negative comments in Bayes is quite large, while in SocialSent it is quite close to each other. The number of comments classified as neutral is quite low compared to other categories. In general, the conclusion is that words with positive emotions are used more than negative ones ironically or normally in comments.



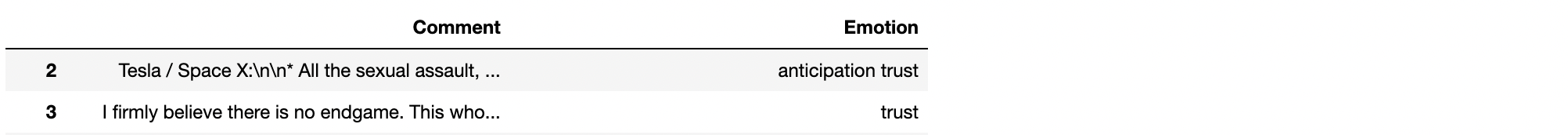
**Figure 32** *This is a bar chart to compare the sentiment category of VADER, TextBlob, and SocialSent analysis.*

In comparison to a NaiveBayesClassifier-based sentiment analysis model, TextBlob using NaiveBayes demonstrated the highest efficacy, achieving a 56% accuracy rate. VADER followed with a commendable 51%, and SocialSent with 50%. These results highlight NaiveBayes' superiority for the dataset, albeit the need for larger datasets for greater success, as a score near 50% implies almost half the predictions are inaccurate. The substantial gap in accuracy between NaiveBayes and other methods indicates the importance of selecting a suitable sentiment analysis model to extract optimal insights from user-generated content.

Sentiment analysis surpasses the binary classification of comments into positive or negative; it involves nuanced categorization that discerns emotional elements within. The LexMo library (<https://betterprogramming.pub/unlocking-emotions-in-text-using-python-6d062b48d71f> accessed on 3 August 2023) aids in identifying emotional attributes in English text. Some comments in the dataset encompassed multiple emotional facets concurrently (Figure 33). Trust and anticipation predominantly imbued the comments (Figure 34), signifying Reddit users' reliance on and expectations from Elon Musk due to the unexpected change.

A computer screen shot of a computer code

Description automatically generated



**Figure 33** *This is a Python formula to calculate the emotion of the comments.*

A graph of emotions

Description automatically generated

**Figure 34** *This is a bar chart to show the distribution of emotions in Reddit comments about Elon Musk.*

# CONCLUSION

In this case study, Reddit comments related to Elon Musk and his recent decision about transforming “Twitter” into “X” were analysed. A dataset of 2000 comments was collected and underwent rigorous data cleaning and pre-processing. The exploratory analysis revealed a consistent daily influx of comments, and the dataset exhibited heterogeneity with diverse contributions from various authors.

The topic model analysis using Latent Dirichlet Allocation (LDA) identified five distinct topics, encompassing discussions about speculation and curiosity, Elon Musk’s role and company, user engagement and money, comparisons and Musk’s image, Musk’s vision and company direction.

Three sentiment analysis methodologies, VADER, TextBlob, and SocialSent, were employed to assess the sentiments expressed in the comments. While all methods showed a higher proportion of positive sentiments, TextBlob emerged as the most accurate model for this dataset, achieving a notable accuracy rate of 56%. Additionally, emotion analysis was applied with the LexMo library, trust and anticipation became the most popular ones.

However, improvements and further developments are recommended, such as increasing the dataset size for better topic modelling and sentiment analysis. Overall, the case study provided valuable insights into public perceptions and sentiments surrounding Elon Musk, but ongoing enhancements in data collection and analysis methodologies would lead to more robust and accurate conclusions.

references