# Research on the Correlation between Twitter Sentiment Analysis and Presidential Campaign Prediction Applications

Yinghao Xu
University of Nottingham
Computer Science
Email: psxyx16@nottingham.ac.uk

Nikhil Menon
University of Nottingham
Computer Science
Email: psynm9@nottingham.ac.uk

Zichao An
University of Nottingham
Computer Science
Email: psxza4@nottingham.ac.uk

Minlong Chen
University of Nottingham
Physics and Astronomy
Email: ppxcm4@nottingham.ac.uk

Junjie Xia
University of Nottingham
Physics and Astronomy
Email: alyjx26@nottingham.ac.uk

Ruoyu Wen
University of Nottingham
Physics and Astronomy
Email: ppxrw2@nottingham.ac.uk

***Abstract*—With the development of digital information, the number of users on social media platforms has sharply increased. A large amount of data flows on social media every day, including content from different fields such as education, healthcare, technology, and politics. These data contents reflect the emotional attitudes of the public towards individuals, organisations, and careers in information exchange. Twitter, as one of the most popular social media platforms today, has at least 80.9 million active users in the United States, making it the country with the highest number of users currently using the platform. The political emotions in tweets have also become a focus of research, as they can reveal public insights into political parties and policies as well as predict election results. This study used TF-IDF, tokenisation, and sentiment analysis on tweets during the 2020 US presidential election, and then applied a sentiment intensity analyzer from the Natural Language Toolkit (NLTK) to predict the sentiments of tweets, and established a classification model which is XGBoost to predict presidents to make the prediction of election trends. The work done by our team not only demonstrates the potential application of big data and machine learning in political analysis but also provides new tools and methods for politicians, decision-makers, and researchers.**

**Keywords: Social Media, Big Data, Sentiment Analysis, Twitter, Machine Learning, Random Forest, XGBoost, Election Prediction, Classification Models**

## I. Introduction

In the era of rapid information development, big data and machine learning technologies have become important tools for research, analysis, and prediction. This article will focus on exploring how to use sentiment analysis to predict presidential elections. The introduction of big data technology brings convenience to processing large-scale and complex data sets[1]. Data can come from multiple data sources, such as social media, news reports, tweets on Twitter, online forums, etc., providing rich data information for the sentiment analysis research in this article. Emotional analysis is often presented in the form of text, using natural language processing, text analysis, and computational linguistics to identify and extract subjective information from users. Especially in the political field, sentiment analysis can be used to study the public's emotional reactions to political issues or candidates, thereby predicting the election results of candidates. When processing large amounts of tweet data, researchers usually use distributed computing and cluster technology to process large-scale data and improve information processing efficiency. When processing this data, we often encounter difficulties in storage, management, and analysis due to the large amount of data. Because when analysing the content of text, the text contains dialects, slang, and unstructured elements (such as likes and shares), which increases the complexity of data analysis, we cannot guarantee that the analysed data does not contain incorrect, duplicate, or irrelevant information, thereby affecting the accuracy of the results. This article adopts text vectorisation techniques such as tokenisation. It separate sentences into individual words, remove stop words from the tweets, and break down the tweets into individual words[2]. It converting Twitter text into fixed-length vectors in order to make computer gets better understanding of the target words[3], enabling sentiment classification or sentiment analysis using models. The TF-IDF method is used to represent the importance of hashtags in a tweet. The higher the weight, the more important the word is in the tweet[4], helping to distinguish important words from common ones, this helps improve the text processing efficiency. The class SentimentIntensityAnalyzer provides a method calculating the polarity score to determine the sentiments of tweets[5], which makes it convenient to train the sentiment model in this project. Another technology this article used is k-means clustering. It clusters the sentiment distribution under different presidential labels, which will help improves the accuracy of model predictions. To deal with the big data challenges, a generally efficient and suitable model is typically required. Choosing machine learning algorithms which support distributed training would be a good choice since they are able to efficiently train on large-scale datasets. Random Forest and XGBoost Algorithm are deployed for sentiment prediction and president prediction model respectively. In summary, this article encountered many challenges in utilising big data analysis to process data and using machine learning models for sentiment analysis to predict presidential approval rates. However, through continuous technical optimisation, these issues were ultimately resolved. With technological innovation, the accuracy and reliability of predictive models will continue to improve, providing a more favourable and scientific basis for political decision-making.

## II. Literature Review

### A. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a numerical statistic which vectorises text by giving numerical values to each word. Danyal et al., 2024 uses Count Vectoriser and TF-IDF to vectorise movie reviews [6]. Sheridan et al., 2024 investigates the connection between TF-IDF and the negative logarithm of the hypergeometric test P-value [7].

### B. Natural Language Toolkit (NLTK)

NLTK is a python package which is used for text-processing, tokenisation, stemming, tagging, parsing, and semantic reasoning. Vencer et al., 2023 uses spell check from NLTK to correct typos and errors in the tweet and calculate word count and sentence length for each tweet. Then Sentiment Intensity Analyser was used to give sentiment scores to each word [8].

### C. Word2Vec

Word2Vec is an unsupervised learning technique consisting of a two-layer network model that embeds the words [10]. Hitesh et al., 2019 performs sentiment analysis on real-time 2018 election twitter data using Word2Vec for feature selection and Random Forest for sentiment classification.

### D. K-Means Clustering

K-Means Clustering is an unsupervised learning technique that groups data into clusters. Zul et al., 2018 compares the accuracy of using a combination of K-Means and Naïve Beyes and just Naïve Bayes for sentiment analysis. The study found just using Naïve Bayes yielded a higher accuracy than using the combination of K-Means and Naïve Bayes [13].

### E. XGBoost

Tree boosting is a highly effective and widely used machine learning method.A new scalable endto-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges.[15]

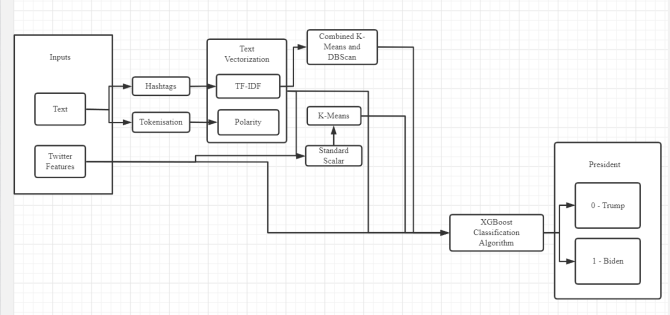

Fig. 1. Data Process

## III. Methodology

Based on our research background, we have decided to use Apache Spark as our computational support. Apache Spark is an open-source big data computing framework, initially developed by AMPLab at the University of California, Berkeley, and open-sourced in 2010. Designed for fast computing of large datasets, it outperforms earlier big data programs such as Hadoop MapReduce. Its core design principle involves loading data into memory for processing, reducing disk reads and writes. Spark provides multiple data handling options, including RDDs (Resilient Distributed Datasets) and higher-level APIs like DataFrames and Datasets, making it suitable for workflows that involve data cleaning, transformation, statistical analysis, data mining, and training and predicting machine learning models. Consequently, PySpark was chosen as our technical support.

We plan to explore several aspects, primarily analyzing the sentiment extracted from tweets, examining who garners more support in the election on Twitter based on various metrics, and analyzing which president a tweet is more closely related to.

### A. Data Collection and Pre-processing

*1) Dataset Selection:* The dataset chosen is “US Election 2020 Tweets” from Kaggle, divided into two files based on data scraped using Snsscrape and the Twitter API with specific hashtags for #Trump and #Biden.

*2) Data Cleaning:* Non-essential columns are removed to streamline the dataset for analysis, enhancing processing efficiency and focusing on relevant variables such as tweets, likes, retweets, and follower counts.

### B. Feature Engineering and Data Visualization

*1) Data Visualization:* Develop plots showing the comparative number of tweets for Trump and Biden, visualizing engagement levels and supporting judgments with additional visualizations.

*2) Feature Engineering:* Introduced a feature named ‘President’, indicating whether a tweet is more related to Biden (0) or Trump (1). Tags are extracted from tweets to create a ‘hashtag’ feature column, grouped and summed to generate ‘hashtag_count’, and the top 20 tags are sorted and categorized into Trump, Biden, and Neutral groups.

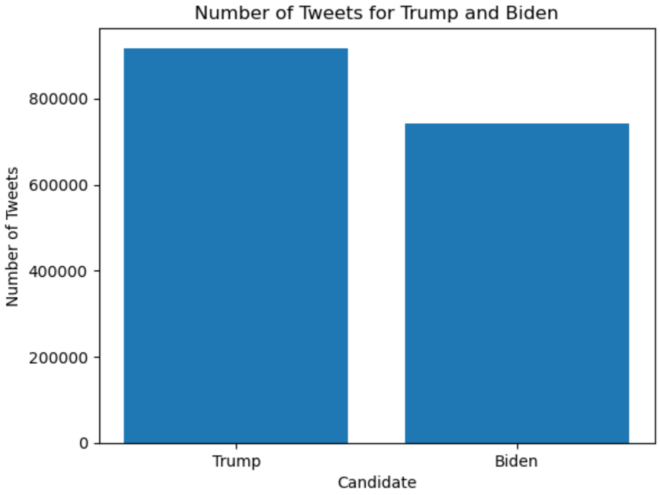

Fig. 2. Number of Tweets for different President

### C. Text Analysis and Sentiment Measurement

*1) TF-IDF Calculation:* As proposed by Kaggle, the hashtags were extracted from the tweets and vectorised them using TF-IDF. [8] Let $t_i$ be the hashtag in the tweet, $d_j$ be the total number of unique hashtags in the tweet, $k_{ij}$ be the Term Frequency (TF) which is 1 divided by $d_j$, N be the total number of tweets in the dataset, and $K_i$ be the number of documents containing $t_i$. The following equations where used to vectorise each hashtag.

$$TF = k_{ij} \tag{1}$$

$$IDF = \log(\frac{N}{K_i}) \tag{2}$$

$$TFIDF = TF * IDF \tag{3}$$

[6]

$$TFIDF = k_{ij} * \log(\frac{N}{K_i}) \tag{4}$$

[7]

*2) Sentiment Analysis:* The tweets underwent tokenisation where stop words, URLs, single letter character and special characters were removed. Then the remaining words were converted to lowercase and stored in an array for further sentiment analysis. Sentiment Intensity Analyzer (SIA) from the Natural Language Processing Toolkit (NLTK) was used to calculate sentiment scores from tweet texts, integrating these into a 'polarity' column to gauge public sentiment towards each candidate. NLTK's VADER sentiment analysis tool assigns the polarity score from range -1 (Negative Sentiment) to +1 (Positive Sentiment) by using a lexicon of words and evaluating the context of each word. [8]

### D. Data Scaling and Clustering

*1) Scaling:* A PySpark Standard Scaler was used to prepare features for clustering, ensuring equal contribution from all features.

*2) K-Means Clustering Algorithms:* K-Means clustering was used to understand data patterns in tweets and features by analyzing cluster centers. 'TF-IDF', the feature most correlated with the target variable 'President' was used for further clustering with a combined K-Means and DBSCAN algorithm.[10] This approach leverages the efficiency of K-Means to find centroids in large datasets and DBSCAN's approach to define cluster boundaries and remove noise points. Points exceeding the Minimum Euclidean Distance from the nearest cluster are categorized as noise points and subsequently removed to improve data quality.

*3) Scatter Plot Analysis:* From the analysis of the generated images, it can be seen that the yellow parts with TF_IDF¡0 are closer to positive values on the polarity axis, indicating a more positive sentiment towards Biden-related tweets. In the purple part, a larger proportion is negative, suggesting a higher satisfaction with Biden. In terms of quantity, the purple occupies a larger part, suggesting that tweets related to Trump are more frequent and his discussion is higher, but the sentiment is mixed, with more concerns.

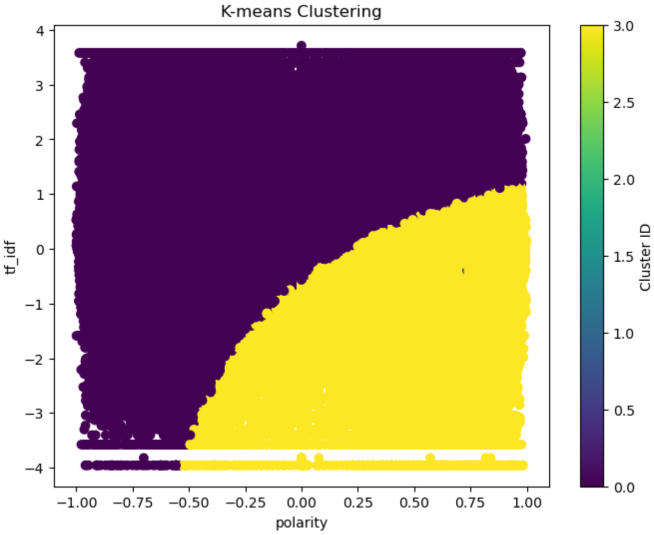

Fig. 3. K-means Clustering

### E. Predictive Modeling

*1) XGBoost Classifier:* An XGBoost classification model was used to classify tweets based on derived features, aimed at predicting user sentiment towards a candidate.

*2) Model Training and Evaluation:* 70% of the data was used for training and 30% was used for testing.

*3) Hyperparameter Tuning with Cross-Validation:* The following hyper parameters were used used for cross-validation:

| Max Depth | 3 | 5 | 7 |
|---|---|---|---|
| Learning Rate | 0.15 | 0.01 | 0.001 |

TABLE I
XGBoost Classification Cross Validation Parameters

## IV. Experimental set-up

### A. Performance Metrics

*1) Accuracy:* Accuracy was the main metric for evaluating the performance of the predictive models. Xgboost achieved an accuracy of 0.85661567% on the test dataset.

*2) Feature Importance:* Assesses which features most significantly influence the predictive accuracy, providing insights into the factors driving public opinion. Because the prediction involved presidential labels, it was observed that TF-IDF contributed the most significantly.It basically matched the expectations.Emotions and the number of followers a user has also influence the prediction of presidential labels.

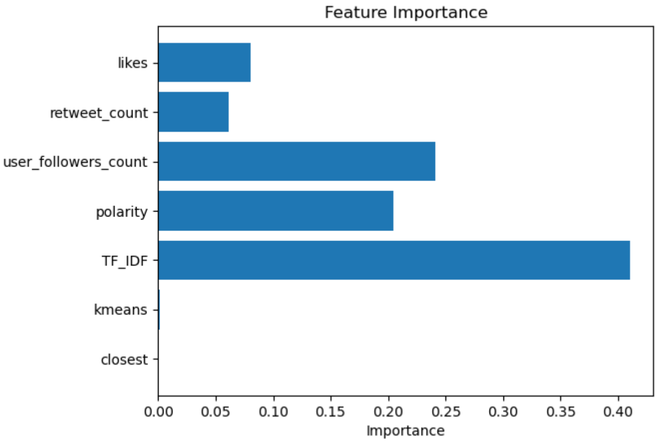

Fig. 4. Featrue Importance in Xgboost Model

### B. Datasets and Validation

*1) Dataset Description:* US Election 2020 Tweets US Election 2020. from Kaggle, Tweets collected, using the Twitter API statuses_lookup and Snsscrape for keywords, with the original intention to try to update this dataset daily so that the time frame will eventually cover 15.10.2020 and 04.11.2020.

*2) Validation Procedure:* Implements a training-test split of 70%-30% to validate model predictions against unseen data, ensuring robustness in the findings,and use random split set the seed = 654321

### C. Clustering and Model Optimization

*1) Elbow Method:* The elbow method is a technique commonly used to determine the optimal number of clusters, especially when performing K-means clustering analysis. [11] The core idea of this method is to identify the best number of clusters by evaluating the impact of the number of clusters on model performance. The Best K is 4.

*2) Hyperparameter Tuning:* Cross-Validation and Grid Search [12] 1. Define a parameter grid: First, define a grid of one or more model parameters.In this project,define max_depth [3,5,7] and learning_rate [0.15, 0.01, 0.001] 2. For each group of parameters in the grid, use the cross-validation method to train and validate the model multiple times. 3. Perform grid search: For each group of parameters in the grid, use the cross-validation method to train and validate the model multiple times.Considering performance issues, a 3-fold cross-validation was used in the experiment. 4. Evaluate the results:Result is depth = 3 and learning_rate = 0.15

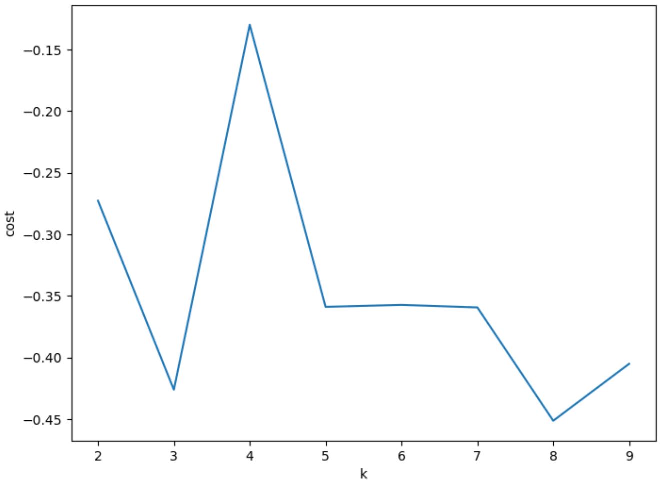

Fig. 5. elbow method find the best K in k-means

## V. Results and discussion

After doing data process including data clean and feature engineer with Spark SQL frame, we employ 3 types ML tasks on the comment datasets about US selection with Spark Mlib frame. The data refers to tweet comment and user information towards presidents. The experiments have been run in databricks cloud computing platform with 1 driver node and 6 workers. Each computational node has 28 core CPUs and 28 GB of RAM.In terms of software, platform used the Spark 3.5.0.

### A. Data Process

First, Reading two CSV datasets to HDFS via Spark, clean and tokenise tweet comments. Then, Use the tokenised data to generate TF-IDF and Document to Vector features. Finally, Splitting the Spark data frame into train and test sets, and use cache method to persist the data on RAM and SSD for next iteration ML calculation.

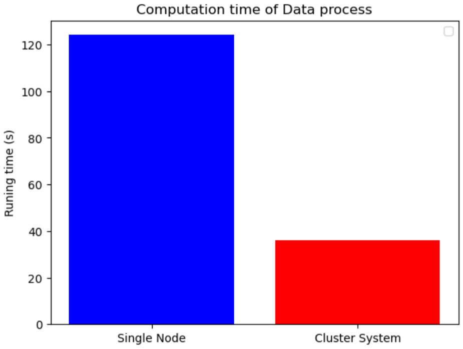

Fig. 6. Computation time of data process

To deal with huge amount of data, perform this process on the distributed system. As shown in Fig.1, when run on Single Node,the computation time is 124.12 (s), while that in cluster system is only 36.15 (s), which means distributed big data computation can largely reduce the data process time.

### *B. Sentiment Analysis*

To predict one tweet comment's sentiment properties, Using the 'Doc to Vec' feature and 'sent' label to build the random forest model. An accuracy of 61% was achieved on this task accuracy on this task.

| Model | RandomForest |
|---|---|
| Accuracy | 61% |

TABLE II
PERFORMANCE OF SENTIMENT ANALYSIS

We also compare the computation time of sentiment analysis task. In spark distributed mode, The running time nearly reduce by half.

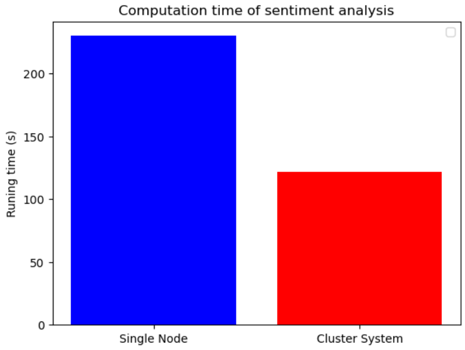

Fig. 7. Computation time of sentiment analysis

### *C. Cluster Analysis*

To explore the sentiment distribution of Twitter comments under different presidential themes, we use K-Means and modified clustering models to perform cluster analysis on TF-IDF, polarity, and MixF (the cross-feature of TF-IDF and polarity).

According to Fig.8, one president theme's comment prefer to be negative, because it shows high TF-IDF with low polaty compared to the another one.

According to Fig.9, We make cluster on TF-IDF.

$$MixF = TFIDF * polarity \tag{5}$$

Compared to the previous polarity feature,the MixF can not only reflect sentiment but also the intensity of the sentiment,which means it can handle problem in a generate method. We witness it shows low MixF with low polarity in one theme.

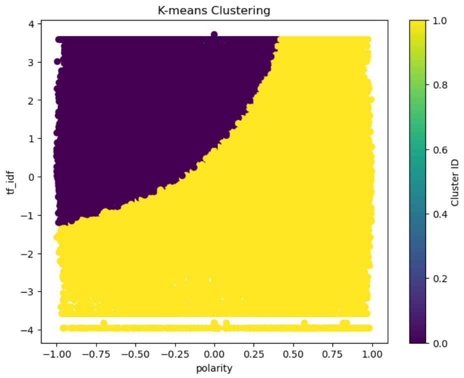

Fig. 8. Cluster anlysis on TF-IDF and Polarity

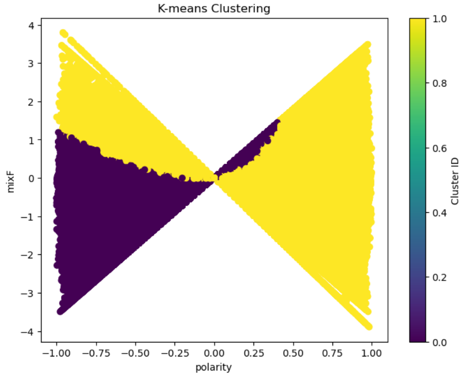

Fig. 9. Cluster anlysis on TF-IDF and MixF

### *D. President topic*

For the prediction of results, first we use a custom function to calculate which features are more important. This step allows us to more clearly understand which features can play a decisive factor. Then our team uses an XGBoost classification The machine is used to predict the president's support for tweets based on the content of the tweets. First, we performed feature engineering on the required data. In this step, we selected the number of likes of the Twitter article, the number of retweets, the user's fans, as well as the emotional extreme values, TF-IDF and K value clustering obtained from the previous model. A series of features such as the desired features. Then we used XGBoost in spark to build a classifier. We then performed parameter tuning by using cross-validation.

| Model | XGBoots |
|---|---|
| Accuracy | 85.7% |

TABLE III
PERFORMANCE OF SENTIMENT ANALYSIS

As for the computation time,the cluster system's run time is 100 (s) while the run time on single node is 280 (s) .

## VI. CONCLUSION

In this article, our team uses big data methods to analyze Twitter data related to the two U.S. presidential candidates.

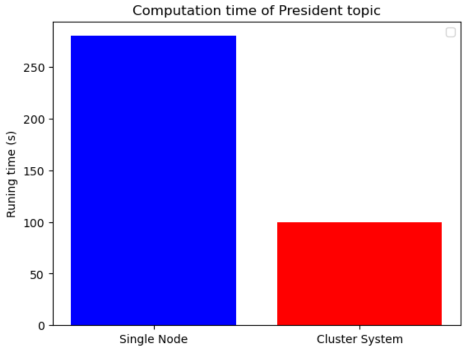

Fig. 10. Computation time of president topic

We achieve this goal by performing sentiment analysis, cluster analysis and HashTag selection on Twitter texts. Finally, a machine learning model was used to reveal public perceptions of the two candidates. The main findings of this study: First, through sentiment analysis of Twitter texts, it was found that American citizens have different emotional tendencies towards different candidates. For example: some Twitter posts are labeled Trump, but the sentiment expressed in the Twitter post expresses disgust for Trump. Our team then used cluster analysis to analyze the relationship between sentiment and hashtags in the two presidents' tweets. Among the tweets about Trump, it is clear that most of the tweets about Trump are negative. There are objections. Finally, a machine learning model is used to predict Twitter posts for classification. Research in this direction could help other politicians in the future better analyze sentiment toward each candidate based on large numbers of Twitter posts. This approach helps some government department personnel better provide theoretical support for government activities. Additionally, changes in public opinion toward each presidential candidate can be captured. Points that can be improved in the future include: for example, by increasing the amount of data, the information hidden in Twitter text data can be more effectively analyzed. In this way, we can better analyze the emotional changes of the masses towards a certain event or a certain person.

## References

[1] Ansari M Z, Aziz M B, Siddiqui M O, et al. *Analysis of political sentiment orientations on twitter[J].* Procedia computer science, 2020, 167: 1821-1828.

[2] Shetty S D. *Sentiment analysis, tweet analysis and visualization on big data using Apache Spark and Hadoop[C]*//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1099(1): 012002.

[3] Dogru H B, Tilki S, Jamil A, et al. Deep learning-based classification of news texts using doc2vec model[C]//2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA). IEEE, 2021: 91-96.

[4] Xiang L. Application of an improved TF-IDF method in literary text classification[J]. Advances in Multimedia, 2022, 2022.

[5] Elbagir S, Yang J. Twitter sentiment analysis using natural language toolkit and VADER sentiment[C]//Proceedings of the international multiconference of engineers and computer scientists. sn, 2019, 122(16).

[6] Danyal, M.M., Khan, S.S., Khan, M. et al. Sentiment analysis of movie reviews based on NB approaches using TF–IDF and count vectorizer. Soc. Netw. Anal. Min. **14**, 87 (2024). https://doi-org.nottingham.idm.oclc.org/10.1007/s13278-024-01250-9

[7] Sheridan, P., Onsjö, M. The hypergeometric test performs comparably to TF-IDF on standard text analysis tasks. Multimed Tools Appl **83**, 28875–28890 (2024). https://doi-org.nottingham.idm.oclc.org/10.1007/s11042-023-16615-z

[8] L. V. T. Vencer, H. Bansa and A. R. Caballero, "Data and Sentiment Analysis of Monkeypox Tweets using Natural Language Toolkit (NLTK)," 2023 8th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 2023, pp. 392-396, doi: 10.1109/ICBIR57571.2023.10147684.

[9] Coletta L F S, da Silva N F F, Hruschka E R, et al. Combining classification and clustering for tweet sentiment analysis[C]//2014 Brazilian conference on intelligent systems. IEEE, 2014: 210-215.

[10] Hitesh M S R, Vaibhav V, Kalki Y J A, et al. Real-time sentiment analysis of 2019 election tweets using word2vec and random forest model[C]//2019 2nd international conference on intelligent communication and computational techniques (ICCT). IEEE, 2019: 146-151.

[11] Qaiser S, Ali R. Text mining: use of TF-IDF to examine the relevance of words to documents[J]. International Journal of Computer Applications, 2018, 181(1): 25-29.

[12] Hardeniya N, Perkins J, Chopra D, et al. Natural language processing: python and NLTK[M]. Packt Publishing Ltd, 2016.

[13] Zul M I, Yulia F, Nurmalasari D. Social media sentiment analysis using K-means and naïve bayes algorithm[C]//2018 2nd International conference on electrical engineering and informatics (ICon EEI). IEEE, 2018: 24-29.

[14] Syakur M A, Khotimah B K, Rochman E M S, et al. Integration k-means clustering method and elbow method for identification of the best customer profile cluster[C]//IOP conference series: materials science and engineering. IOP Publishing, 2018, 336: 012017.

[15] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.