Exploring 2020 US election tweet sentiment: analysis based on big data and machine learning

Cheng Yan
Department of Computer Science
University of Nottingham
Nottingham, UK
psxcy8@nottingham.ac.uk

Xuanhong Luo
School of Computer Science
University of Nottingham
Nottingham, UK
psxxl19@nottingham.ac.uk

Wenchao Xia
School of Computer Science
University of Nottingham
Nottingham, UK
psxwx2@nottingham.ac.uk

Shangkai Jiang
School of Computer Science
University of Nottingham
Nottingham, UK
psxsj13@nottingham.ac.uk

Chengkai Wang
School of Computer Science
University of Nottingham
Nottingham, UK
psxcw11@nottingham.ac.uk

Wentong Du
School of Computer Science
University of Nottingham
Nottingham, UK
psxwd3@nottingham.ac.uk

***Abstract*—This paper uses Apache Spark, natural language processing and machine learning related technologies to analyze public sentiment on Twitter during the 2020 US presidential election. Considering the huge influence of social media in modern political communication, this article takes Twitter as the research object and selects the tweets of Biden and Trump related to the election as a data set. This paper uses sentiment analysis algorithm TextBlob to classify tweets into different types of emotions, and then uses TF-IDF technology to extract text features. Next, the effects of different machine learning algorithms on sentiment classification prediction of text data, such as Logistic Regression, One vs Rest, Naive Bayes, and Random Forest, are compared and evaluated through multiple indicators. The results show that logistic regression has the best overall performance. At the same time, pineline technology is used in this paper, including the pipeline work of the above steps. In addition, the paper uses the Tableau data visualization tool to examine and explore the sentiment data of the cleaned tweet to get a better understanding of the relation between a winning nominee and tweet sentiments. Through these methods to develop a machine learning model that accurately predicts tweet sentiment.**

***Index Terms*—Sentiment Analysis, Natural Language Processing, Machine Learning, 2020 US Presidential Election, Pineline, Data Visualization, Apache Spark, Pipeline, Tableau**

I. Introduction and background

The US presidential election has been a global issue of concern, as the entrance of a new president may bring drastic effects to the political and economic sectors of the United States and the world. The data shows that approximately, 22% of American adults are users of Twitter (Fujiwara, M ü ller, &Schwarz, 2021)[5]. During the US presidential election, a lot of election-based tweets will flood the social media platform, majorly on Twitter. The candidates can use the platform to describe their sentiments and ideas in their policies by sending out tweets. Similarly, their fans can describe their support and conductiveness to the election by either sharing tweets or use the platform to write their personal ones to reflect their positions (Buccoliero et al., 2018)[2].

Twitter availed an abundance of real-time tweet data during elections associated with each candidate. It, therefore, provided valuable data sources for scholars to explore a deeper understanding of the level of the public support of different candidates by user unit and the voter's attitudes and emotions. Reasonable analysis and emotional prediction of such can assist in predicting understanding public opinion before an election is held and determining the final election results to some extent.

The main aim of this study is to realize how to undertake sentiment analysis using twitter in predicting candidate-related data classification, getting to train tweet data related to candidates so that we understand which candidate has more supporter data or not and so to get a reference perspective.

During the project implementation process, we first cleaned and preprocessed Twitter data, including handling missing values, removing noise, labeling emotions, and extracting features. Then, we constructed multiple models using machine learning algorithms to classify Twitter emotions. Finally, we evaluated and compared the accuracy of the model, recall, and F1 score and explored and analyzed the cleaned Twitter sentiment data through data visualization tools.

Research questions

This project's main research questions are as follows:

1) How can Twitter data be effectively used for sentiment analysis to know the public candidate support level and the voters' attitudes and emotion collections?

2) Which model is the most effective in classifying sentiment in tweets?

3) How to evaluate and fine tune model performance?

The case of our project can be summarized as: by applying Apache Spark and machine learning algorithm due to emotion analysis, the emotion classification and predictions of Twitter will be managed. It provides a fresh way of predicting the supporter and candidates' results. It not only provides a fresh perspective for political analysis and decision-making but also gives concerns and examples based on verification and justification. We pray that this study's implementation can more accurately establish further predictable analytical tools for method and research on social sciences and political conclusions.

II. Methodology

Overall, this study discusses a systematic method used to analyze Twitter data on the 2020 US presidential election. It integrates methods of data preprocessing, natural language processing, machine learning, and data visualization to enhance the understanding of voters' emotions on Twitter and how they are likely to influence the election outcome. The following is an overview of the key steps:

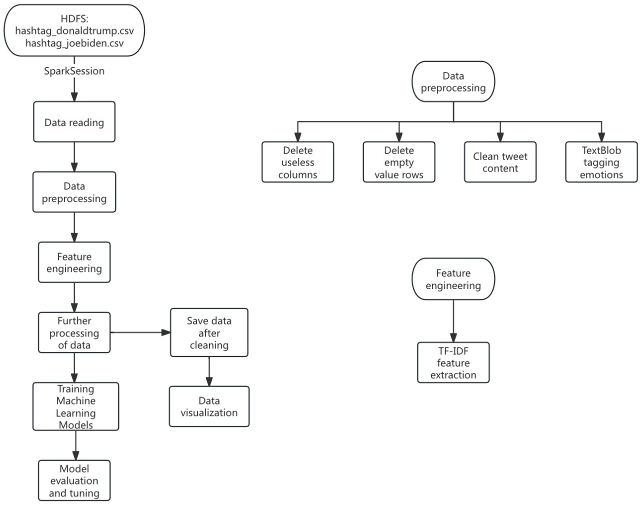


Fig. 1. The complete flowchart of this project.

Data Reading

Firstly, we use Apache Spark to create a SparkSession and load the original Twitter data file using this SparkSession to read it into the Spark cluster memory, which can improve data processing efficiency.

Data Preprocessing

Secondly, delete some columns in the original data that are not related to the project goal, such as user ID, username, likes, etc. Cleaning up these useless columns is beneficial for improving the predictive performance of the model during the training phase. In addition, due to the large volume of the original data with over 1.4 million entries, we have adopted the direct deletion of rows with missing values. After deletion, there are still about 580000 entries in the data, which will not affect the accuracy of model training and prediction.

Then we adopt a local design approach to clean up the text content in the original tweet, removing irrelevant text content for sentiment analysis, such as URLs, stop words, punctuation, numbers, and spaces, in order to reduce data noise and improve model prediction accuracy.

Use TextBlob to analyze personal tweets and add emotional polarity as a new column to the data for training and predicting machine learning models. Diyasa, I.G.S.M. et al. (2021)[4] mentioned that Textblob is a Python library used to classify data. In order to classify data, Textblob calculates polarity values. In addition, we also compared the impact of setting different polarity thresholds on the accuracy of model predictions when using Textblob.

Feature Engineering

Using TF-IDF for Feature Extraction of Text.TF-IDF plays a crucial role in the sentiment analysis of Twitter text, which can help us to extract keywords, filter noise, and distinguish word importance, so as to improve the accuracy and effectiveness of sentiment analysis(Mee, A et al., 2021)[8].

The TF-IDF value is calculated as follows:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{1}$$

where:

- $TF(t,d)$ is the term frequency of term t in document d.
- $IDF(t)$ is the inverse document frequency of term t and is given by:

$$IDF(t) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \tag{2}$$

In the above equation:

- N is the total number of documents in the corpus.
- $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears (i.e., $documentfrequency$)(Liu et al., 2018)[7].

Further Processing of Data

Next, merge these two data tables and add candidate names as new columns to distinguish different data for comparison and display in the data visualization process. Data visualization can provide a deeper understanding of complex data information from different dimensions and aspects.

Training Machine Learning Models

Then, multiple machine learning models (such as Logistic Regression, One vs Rest, Naive Bayes, and Random Forests) are used to train and predict the dataset. Then, obtain the best performance configuration model through grid search and cross validation. Finally, by comparing the training runtime and accuracy of different models, select the model with the best performance and most suitable for the current dataset.

Model Evaluation and Tuning

In the model evaluation phase, we use accuracy, recall, and F1 score to evaluate the performance of the model. In addition, We also compared the runtime of different CPU cores and dataset sizes in Spark clusters to understand model scalability.

Data Visualization

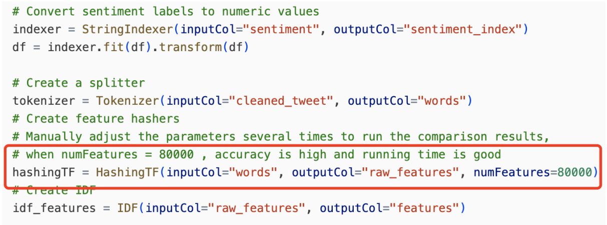
Finally, use Tableau to visualize the cleaned data and explore the relationship between Twitter sentiment and election results. Tableau is a data visualization software widely used in the field of data analysis, which provides powerful features to create interactive dashboards and charts using multiple large datasets. Users can use these tools to deeply analyse data in order to identify patterns, trends and correlations in the data (Batt, S et al., 2020)[1].

III. Experimental set-up

This project is based on two original datasets, involving Twitter data from Trump and Biden. The data information includes creation time, tweet content, creator ID, tweet latitude and longitude, country and region. In the data reading phase, use toDF() to convert dataset rdd to Dataframe to improve the processing performance of projects with large datasets.

Then we deleted a large number of data columns that were not related to the purpose of this experiment, and only retained key data such as tweet content, country, state, etc. Then, using Textblob natural language processing to predict and classify the sentiment of the tweet text content, when using Textblob to process text data, by comparing the impact of setting different polarity thresholds on the model's prediction accuracy, setting a threshold of 0.5, that is, if it is greater than 0.5, it is positive, if it is less than -0.5, it is negative, and if it is between -0.5 and 0.5, it is neutral, to classify the sentiment of the tweet content. After cleaning and preprocessing the original data (including text content and label (sentiment) information), they are merged into a complete dataset for subsequent model training and prediction.

In the feature engineering stage, we use TF-IDF technology to extract features from raw text data. Through manual testing and parameter adjustment, as shown in the pseudocode in Figure 2, we found that setting ”numFeatures=80000” can improve the accuracy and good performance of the model.

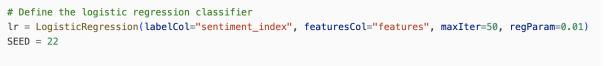


```
# Convert sentiment labels to numeric values
indexer = StringIndexer(inputCol="sentiment", outputCol="sentiment_index")
df = indexer.fit(df).transform(df)

# Create a splitter
tokenizer = Tokenizer(inputCol="cleaned_tweet", outputCol="words")
# Create feature hashers
# Manually adjust the parameters several times to run the comparison results,
# when numFeatures = 80000 , accuracy is high and running time is good
hashingTF = HashingTF(inputCol="words", outputCol="raw_features", numFeatures=80000)
# Create IDF
idf_features = IDF(inputCol="raw_features", outputCol="features")
```

Fig. 2. numFeatures parameter setting.

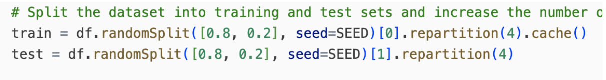
In the current dataset training task, considering the accuracy and runtime of machine learning model comparisons, we compared four models: Logistic Regression, Naive Bayes, One vs Rest, and Random Forest, and the best performance was the Logistic Regression model. When considering the parameter settings of the logistic regression model, with grid search and cross validation, it is found that the performance of the model's configuration parameters is optimal for “maxIter=50” and “regParam=0.01”, as shown in Figure 3 Pseudocode.



```
# Define the logistic regression classifier
lr = LogisticRegression(labelCol="sentiment_index", featuresCol="features", maxIter=50, regParam=0.01)
SEED = 22
```

Fig. 3. logistic regression model parameter setting.

The pipeline technology is used to organize the dataset preprocessing steps feature engineering steps and model training steps into a complete process. For the model training and prediction steps in Figure 4, the first step is to divide the dataset into the training set and the test set in the ratio 0.8:0.2, and at the same time, use the.repartition(4) to divide the larger dataset to facilitate the model training efficiency. Re partition the training data set to 4 partitions and cache them. In Spark, re-partitioning helps optimize the performance of subsequent processing, especially when large datasets are processed in parallel. The process of caching is designed to keep training data in memory so that multilevel accesses do not require a recalculation from the disk.



```
# Split the dataset into training and test sets and increase the number o
train = df.randomSplit([0.8, 0.2], seed=SEED)[0].repartition(4).cache()
test = df.randomSplit([0.8, 0.2], seed=SEED)[1].repartition(4)
```

Fig. 4. dataset split and partitions.

In the final step, we calculated the performance metrics of each model and assessed its performance, we estimated the performance of the model which provides a deeper understanding of its generalization ability, it also enables us to know how the model will perform on new, unseen data, which will be used for real-world productivity. Because the final purpose of machine learning is to apply this technology to predict unseen data instances more accurately and continuously(Raschka, S., 2018)[10].

IV. Results and discussion

Through multiple experiments and analyses, several findings have been discovered that can prove the effectiveness of methods and parameter settings used for the training of emotion classification prediction models.

Impact of data preprocessing on model performance

First, we managed to demonstrate the importance of data preprocessing. The quality of the data preprocessing phase impacts the rest of the model training and models the end results. When performing the sentiment analysis of complex textual content, a reduction of noise in the text and the elimination of irrelevant content can improve the accuracy of the sentiment classification and the performance during feature extraction. The results of this experiment on data preprocessing demonstrate the importance of this stage in real-world applications and provide a solid foundation for the development of efficient and accurate sentiment analysis models.

Reduce data movement in the Spark cluster

On the question of the work with the large-volume data that is going to be reused, we focused on how it is possible to reduce the data movement in building the Spark cluster and make the entire project work more efficiently. In particular, it was determined that the adequate use of the caching approach can reduce data movement costs overhead and can be specifically beneficial for the large-volume data that is going to be reused in different steps. This measure has proven to be efficient in practice is capable of improving performance in all large-scale data processing. We found that the rational use of caching can effectively reduce the cost overhead of data movement, especially for large-volume data that needs to be reused. This strategy proves to be effective in practice and can significantly improve performance in large-scale data processing tasks.

Model performance comparison

For model performance comparison, we systematically evaluated four commonly used machine learning models: Logistic Regression, Naive Bayes, One-vs-Rest and Random Forest, as shown in Table I. Our results show that the logistic regression model performs well in terms of evaluation metrics such as prediction accuracy. In terms of running time, although logistic regression is not as fast as the naive Bayes algorithm, it is still within the acceptable range. In comparison, although the naive Bayes algorithm has the fastest running time, its accuracy is far inferior to the logistic regression and One-vs-Rest models. In summary, we believe that logistic regression is most suitable for sentiment analysis tasks on Twitter data related to political figures. Through grid search and cross-validation techniques, we identified optimal model parameter settings that further improved the performance and stability of the model. Our comparison of key performance indicators such as model accuracy and runtime is shown in Figure 5

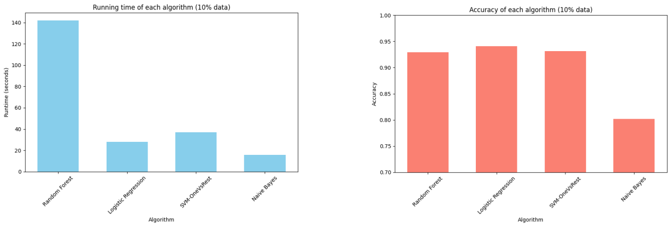


Fig. 5. Comparison of running time and accuracy of each model.

TABLE I
PERFORMANCE METRICS OF EACH MODEL

Model	Accuracy	Runtime (s)	Recall	F1 Score
Logistic Regression	0.9405	27.97	0.9405	0.9303
SVM(OneVsRest)	0.9312	36.99	0.9312	0.9148
Naive Bayes	0.8015	15.69	0.8015	0.8434
Random Forest	0.9292	141.98	0.9292	0.8951

Impact of parameter tuning on model performance

As shown in Figure 6, after doing what we called model performance optimal mechanism about maxIter equals 50 and regParam equals 0.01 significance, we discovered that the Logistic regression model would show the best performance using the standard mechanism; hence good parameters are significant when thinking about a model.

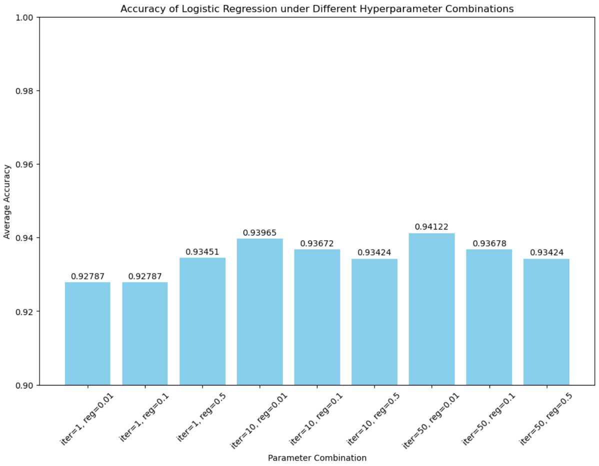


Fig. 6. Evaluation results of logistic regression adjustment parameters.

Scalability of the solution

As for the evaluation of the scalability of logistic regression models, we can say that the current model possesses good scalability. In particular, we noticed that the model's performance is weakly affected by the size of the dataset. In other words, we checked datasets from 10% to 100% using Notebook in Databricks and noticed that when using Spark clusters with the same number of CPU cores, the model calculated its running time within a stable range; it is represented in Figure 7. Therefore, it is possible to argue that the results obtained show that the created model still has good performance in processing big data. Moreover, it is worth mentioning that if looking at the range, there were some fluctuations; however, they should be associated with the cluster load status and cluster health. Hence, it is possible to say that the current logistic regression model has excellent scalability, which means that we developed a good machine learning model on the current dataset. Moreover, the results of the current study make a great contribution to the use of big data and machine learning methods in predicting the emotional state of social media text.

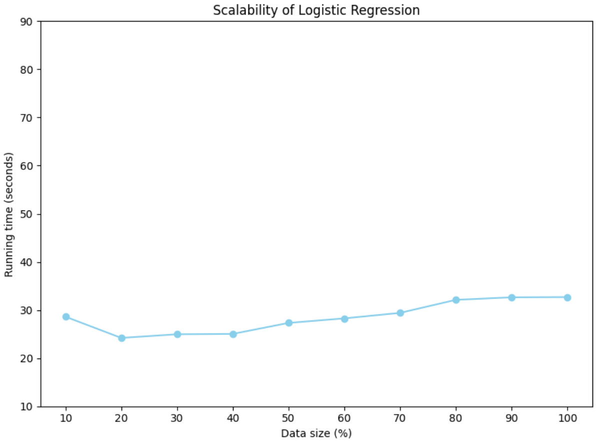


Fig. 7. Scalability of Logistic Regression.

The superior performance of simple models like logic regression models in tasks such as sentiment analysis suggests that traditional and simple models are still more effective when working with social media data. Furthermore, the approach of manual parameter tuning was also revealed to be an effective method of improving the model. Lastly, this study provides an initial assessment of the model's scaling ability, which can be further tested in the future for working with larger volumes of data, and how to optimize the model even more to handle even more significant amounts of data.

V. Data visualization

As the preprocessing and cleaning of Twitter data were performed, we decided to use Tableau to visualize and present the results so that we could further investigate the speculation of the correlation between the sentiment of the tweet and the win of the candidate.

First, we counted the winners in each country and the state and determined the winner as of that time by analyzing the amount of support sentiment for the current candidate in the country apart. From figure 8, it can be seen that Trump's Twitter support sentiment was often higher than Biden's.

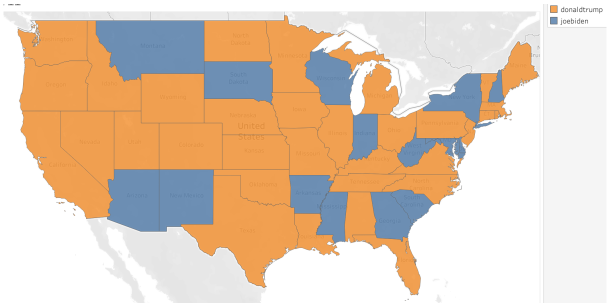


Fig. 8. Based on the sentiment of the tweet, calculate the winner in each state.

Grouped the states by state and performed a bar chart comparison, which displayed the contrast in support sentiment for Trump and the Biden elections for the state. Same as in figure 9, we can see that in most comparisons, Trump's support sentiment was outnumbering Biden's.

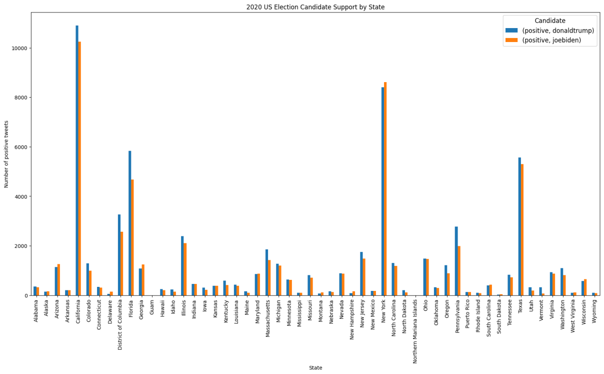


Fig. 9. Detailed data disaggregating state-by-state sentiment in support of both candidates.

In addition, we grouped the states by state, same as the previous point, and created the bar chart shown in figure 10, which presented fine-grained data on support, neutrality, and opposition for the two candidates divided by the states.

Finally, using pie and bubble charts, which can be seen from figure 11 and figure 12; we presented another dimension of the data. As can be seen, there were much more tweets in California, New York, Florida, and Texas, while, as a result,

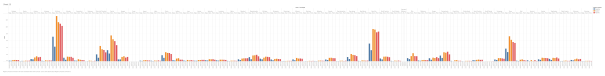


Fig. 10. Detailed data on sentiment categorisation in tweet data for both candidates.

it can be clearly concluded that the number of adults living in those states and using Twitter far outnumbered those of other states, also, they were perhaps more interested and motivated by political topics. Thus, if the candidate gets support in those countries, in the final ballot, they will get a lot of wins.

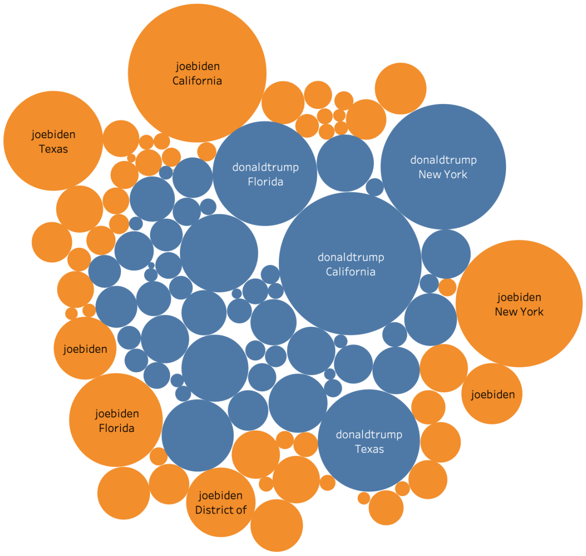


Fig. 11. Bubble chart comparing Biden and Trump's data on support tweets in different locations.

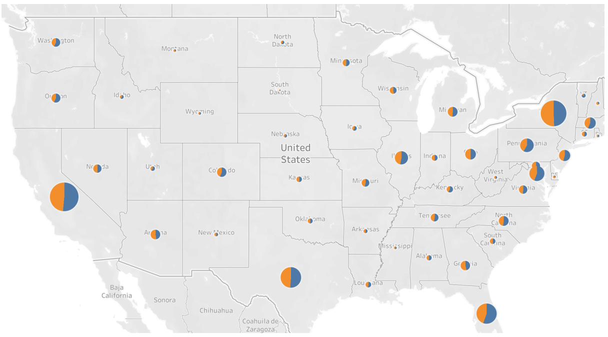


Fig. 12. Geographic distribution of the proportion of Biden- and Trump-supporting tweet data among U.S. states.

VI. Conclusion

After using big data technology and machine learning models to analyze emotions in tweets about Donald Trump and Joe Biden, and after multiple experiments and analyses, we finally get a prediction accuracy of up to 94%, which shows that our model can accurately predict.

And emotional classification in the tweet, and in the aspects this paper has also drawn a lot of conclusions, which also has a considerable shock. Firstly, before analyzing complex text data, it is necessary to preprocess the data reasonably to eliminate the added value added by noise in the raw text data. This is a very important step to ensure the accuracy of the model and can effectively reduce the interference of noise on the results of emotional analysis.

Secondly, when analyzing and researching datasets, it is necessary to compare the transaction efficiency between several machine learning models in detail to select the most suitable machine learning model for the current dataset to study. Our comparison of machine learning models shows that the Logistic Regression model has higher accuracy than Navie Bayesian, OneVsRest and Random Forest models, but its efficiency is slightly reduced compared to ordinary Bayesian models. This suggests that the logistic regression model itself has a greater advantage in dealing with emotional classification of social software text content.

In addition, the feasibility of how much effect the parameter adjustment has on the model efficiency expresses that the model developed in the detail process needs to be carefully optimized. By adjusting parameters such as maxIter and regParam. The performance of the logistic regression model can well achieve perfect results, which can demonstrate the realization effect of the development model by iteration and experiment optimization well.

Meanwhile, it is worth acknowledging that the assessment of the scalability of the logistic regression model in question also shows that the model works well with large datasets. Thus, logistic regression models to predict sentiment in social media text big data could be considered viable and valuable methods, which can serve as a reference for further studies in the related area.

Overall, the conclusions of this study are vital to understand how to apply and learn a machine learning technique in social software emotional feature analysis. In the future, we also plan to explore more advanced models and technologies to extend the accuracy and efficiency of emotional analysis and the scalability of the model.

References

[1] Batt, S., Grealis, T., Harmon, O. R., & Tomolonis, P. (2020b). Learning Tableau: A data visualization tool. *the Journal of Economic Education, 51*(3–4), 317–328. https://doi.org/10.1080/00220485.2020.1804503

[2] Buccoliero, L., Bellio, E., Crestini, G., & Arkoudas, A. (2018b). Twitter and politics: Evidence from the US presidential elections 2016. *Journal of Marketing Communications, 26*(1), 88–114. https://doi.org/10.1080/13527266.2018.1504228

[3] Chandra, R., & Saini, R. S. (2021). Biden vs Trump: Modeling US General Elections Using BERT Language Model. IEEE Access, 9, 128494–128505. https://doi.org/10.1109/access.2021.3111035

[4] Diyasa, I. G. S. M., Mandenni, N. M. I. M., Fachrurrozi, M. I., Pradika, S. I., Manab, K. R. N., & Sasmita, N. R. (2021b). Twitter Sentiment Analysis as an evaluation and service base on Python Textblob. *IOP Conference Series. Materials Science and Engineering, 1125*(1), 012034. https://doi.org/10.1088/1757-899x/1125/1/012034

[5] Fujiwara, T., Müller, K., & Schwarz, C. (2020). The Effect of Social Media on Elections: Evidence from the United States. *Social Science Research Network.* https://doi.org/10.2139/ssrn.3719998

[6] López-Chau, A., Valle-Cruz, D., & Sandoval-Almazán, R. (2020). Sentiment analysis of Twitter data through machine learning techniques. *In Computer communications and networks* (pp. 185–209). https://doi.org/10.1007/978-3-030-33624-0_8

[7] Liu, C. Z., Sheng, Y. X., Wei, Z. Q., & Yang, Y. Q. (2018, August). Research of text classification based on improved TF-IDF algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)* (pp. 218-222). IEEE. https://doi.org/10.1109/IRCE.2018.8492945

[8] Mee, A., Homapour, E., Chiclana, F., & Engel, O. (2021b). Sentiment analysis using TF–IDF weighting of UK MPs' tweets on Brexit. *Knowledge-based Systems*, 228, 107238. https://doi.org/10.1016/j.knosys.2021.107238

[9] Mary, G. P. A., Hema, M. S., Maheshprabhu, R., & Guptha, M. N. (2021, December). Sentimental Analysis of Twitter Data using Machine Learning Algorithms. *In 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)* (Vol. 1, pp. 1-5). IEEE. https://doi.org/10.1109/FABS52071.2021.9702681

[10] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808.*. https://doi.org/10.48550/arXiv.1811.12808

[11] Xia, E., Yue, H., & Liu, H. (2021, April). Tweet sentiment analysis of the 2020 US presidential election. *In Companion proceedings of the web conference 2021* (pp. 367-371). https://doi.org/10.1145/3442442.3452322