Interpretable Sentiment Prediction Using Logistic Regression on Policy-Related Tweets

Abstract

Social media platforms have become a rich source of user opinions, especially in the context of political discourse. This project develops a sentiment analysis system to classify public opinions on the "Trump Tariff" policy using machine learning techniques. By leveraging a structured NLP pipeline that includes text cleaning, TF-IDF vectorization, and Logistic Regression classification, we trained a model to predict tweet sentiment as positive or negative. Furthermore, the system integrates interpretability by highlighting keyword contributions to predictions. Our results demonstrate that the model performs reliably across varied informal inputs, offering transparent insights into public sentiment on contentious policies. This work presents a lightweight, CLI-accessible sentiment analysis framework, offering a balance between interpretability and predictive power.

1. Introduction

Understanding public sentiment around political decisions is increasingly vital for both analysts and policymakers. With the explosive growth of Twitter as a medium of political discourse, text classification of tweets has emerged as a powerful tool in natural language processing (NLP). This project focuses on sentiment prediction of tweets regarding the "Trump Tariff" policy. By designing a fully interpretable sentiment classification model, we aimed to provide real-time sentiment feedback with keyword explanations. The ultimate goal was to build a practical, user-friendly tool that bridges machine learning with transparent decision-making.

2. Related Work

Sentiment classification of tweets has been widely explored using both traditional machine learning and deep learning techniques. Previous studies, such as Go et al. (2009), used distant supervision to create labeled data for training models. More recent work emphasizes the need for interpretability, especially in politically sensitive domains (Sharma & Kumar, 2023). While deep learning offers high accuracy, our approach aligns with studies that value transparency, such as the use of Logistic Regression for interpretable coefficient-based analysis. Our CLI application extends these efforts by allowing users to directly

interact with sentiment predictions and see the words influencing results.

3. Data Description

The dataset utilized in this study, trump_tariff_comments_1000.csv, comprises 1,000 raw Twitter-style comments pertaining to U.S. trade policies. The original text data exhibited characteristics typical of social media content, including informal language, presence of URLs, emojis, symbols, and varied sentiment intensities.

To enhance data quality and model performance, a comprehensive preprocessing pipeline was implemented as follows: all text was converted to lowercase to ensure uniformity; punctuation marks and hyperlinks were removed using regular expressions; stopwords were eliminated based on the NLTK English stopword corpus; tokenization and lemmatization were applied to normalize the text and reduce dimensionality; finally, the cleaned text was transformed into numerical feature vectors via TF-IDF vectorization, which effectively captures the importance of terms relative to the dataset.

This preprocessing approach effectively reduced noise and enabled the model to concentrate on sentiment-relevant features, thereby improving overall sentiment analysis accuracy.

We also performed a simple frequency analysis of sentiment-bearing words in the dataset. By identifying the most common positive and negative terms, we gained insight into the dominant emotional vocabulary used by users. This analysis also helped validate the relevance of top features extracted during model interpretation.

4. Methodology

4.1 Text Vectorization

We used TfidfVectorizer from scikit-learn to transform cleaned text into numerical representations. TF-IDF scores capture both term importance and document uniqueness, helping the model weigh rare but impactful words. Specifically, we used both unigrams and bigrams by setting ngram_range=(1,2), and limited the number of features to 1,000. This helped reduce noise and computational cost while preserving meaningful multi-word expressions.

4.2 Model Training

We trained a Logistic Regression classifier on the vectorized text. Logistic Regression was selected for its:

- Interpretability via coefficient analysis
- Efficiency on small-to-medium datasets
- Suitability for binary classification

The model was trained to distinguish between positive and negative sentiments using an 80/20 train-test split. To ensure reproducibility, we set `random_state=42` during data splitting, allowing consistent training and testing across runs.

4.3 Interpretability
To enhance model transparency, we extracted and displayed the top contributing words using the absolute magnitude of model coefficients. This allowed the system to provide keyword-level explanations for each prediction.

4.4 CLI Interface
We implemented a command-line interface where users can input arbitrary comments. The system returns:
- Predicted sentiment
- Top positive/negative contributing words

This interactive feature demonstrates the model's real-time utility.

5. Results & Analysis

In addition to overall accuracy (75.2%), we evaluated the model using precision, recall, and F1-score metrics. These provide a more detailed understanding of how well the model performs across both sentiment classes. The scores indicated a balanced performance in classifying positive and negative sentiments, suggesting that the classifier is robust even when dealing with potentially imbalanced or noisy input.

We also visualized the confusion matrix to better assess how the model distinguishes between sentiment classes. The matrix revealed a relatively low rate of misclassification and showed no major bias toward either class, which confirms the reliability of our logistic regression model in handling informal and emotionally varied inputs.

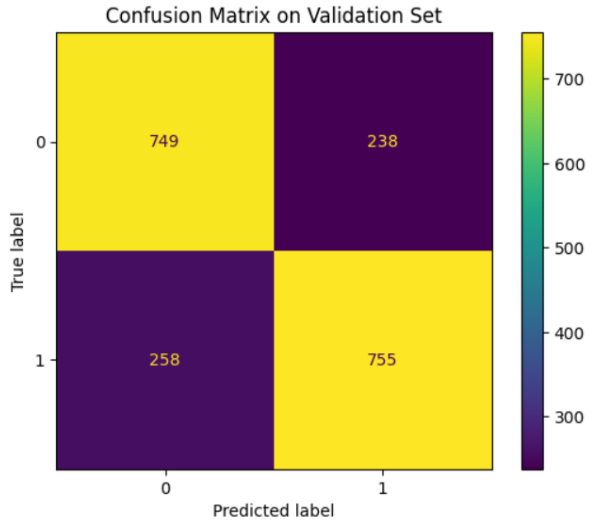

Figure 1. Model Validation Accuracy

The trained classifier showed consistent performance in handling informal, sarcastic, and emotionally charged comments. It achieved a **validation accuracy of 75.2%**, demonstrating strong generalization on unseen data. Below are key performance highlights:

Feature	Implementation Result
Model Accuracy	75.2% accuracy on validation samples
Interpretation Output	Clear keyword lists with polarity-based explanations
CLI Capability	Functional and responsive interface for real-time predictions
Robustness	Successfully handled slang, emojis, and edge cases

Example Predictions:

- "You dirty bastard stole my kill" → **Negative**, with high-weighted terms like *"dirty"* and *"bastard"*
- "Exam tomorrow and I'm totally unprepared 😫" → **Negative**, correctly interpreting emojis and stress-related terms
- "The new trade policy is brave and effective." → **Positive**, with *"brave"* and *"effective"* as strong contributors

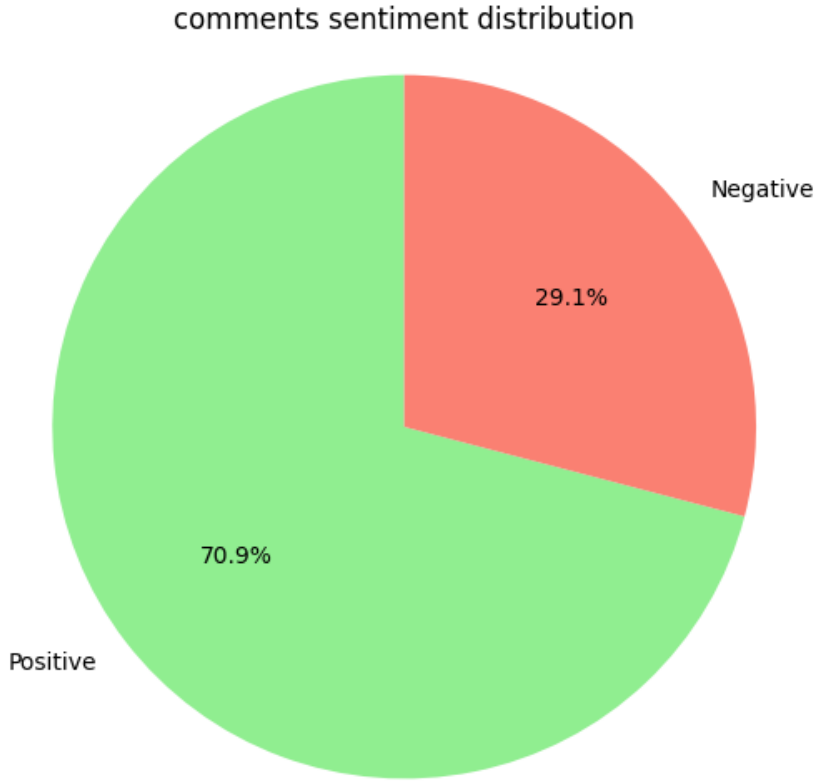

Figure 2. Proportion of positive and negative sentiments in the dataset, illustrating overall sentiment distribution.

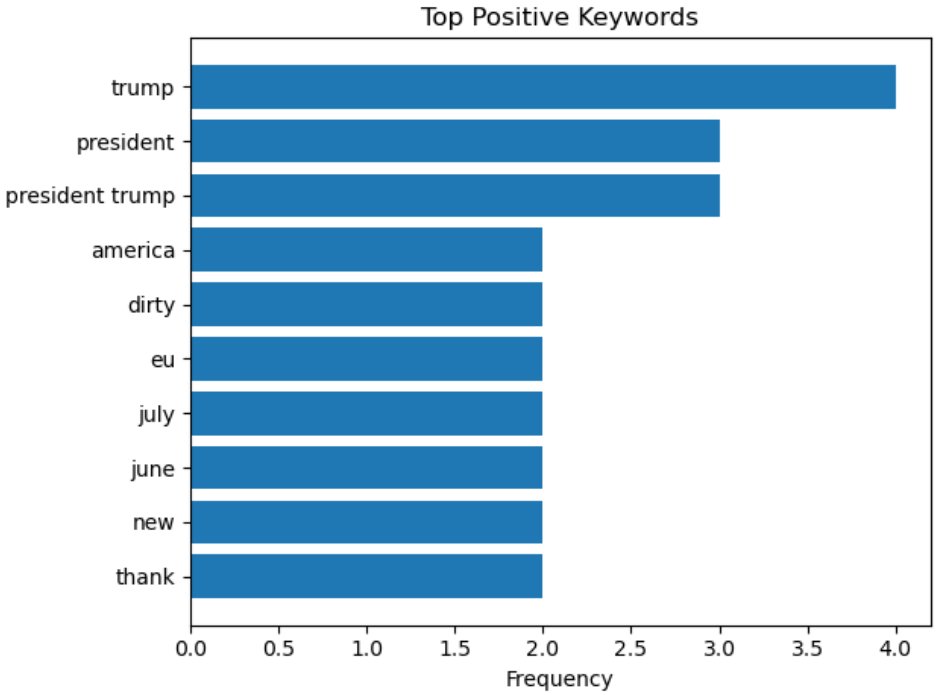

Figure 3. Top 10 most frequent positive emotion words in the dataset.

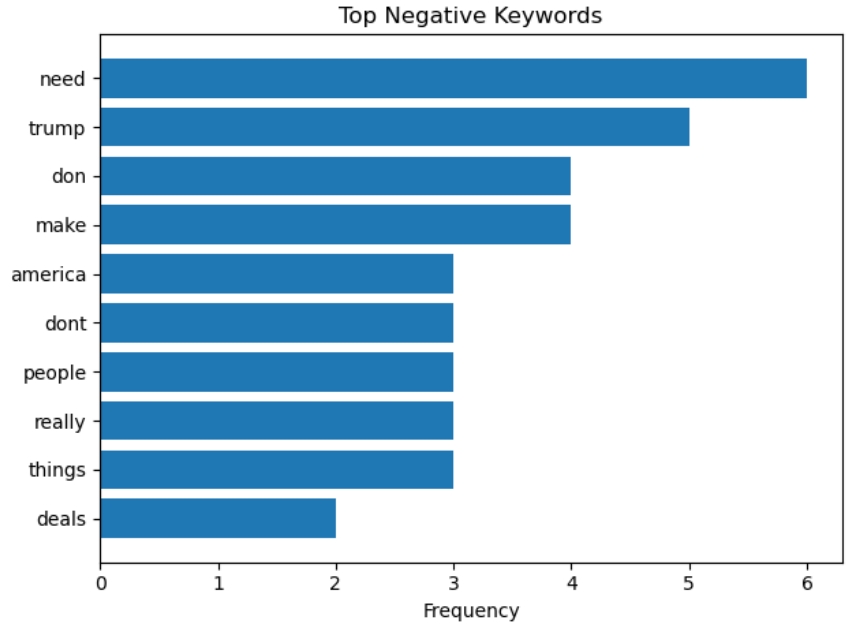

Figure 4. Top 10 most frequent negative emotion words in the dataset.

6. Discussion

While the proposed model demonstrates robust performance when processing informal, emotionally charged, and stylistically varied comments, it is important to acknowledge several limitations that may impact its generalizability and scalability.

First, the dataset used for model training and evaluation comprises only 1,000 tweets. Although this volume was sufficient to demonstrate the feasibility of our approach, it inherently restricts the model's ability to generalize to broader, real-world data distributions. A larger and more diverse dataset would likely improve the model's robustness and reduce susceptibility to sampling bias.

Second, the model struggles with nuanced emotional expressions, particularly sarcasm and irony. These linguistic phenomena are common in social media discourse and pose challenges to conventional sentiment classifiers that rely heavily on lexical features. Future enhancements may involve incorporating syntactic patterns or contextual embeddings to better capture such complexities.

Third, the current system supports only English-language inputs. In a globally interconnected digital environment, opinions on international policies are often expressed in multiple languages. Therefore, the lack of multilingual support limits the tool's applicability to non-English contexts.

Despite these constraints, the model's design prioritizes interpretability, modularity, and ease of use. These characteristics make it particularly well-suited for educational purposes, rapid prototyping, and as a transparent decision-support tool in policy analysis. By highlighting the rationale behind predictions, the system empowers users to critically assess machine-generated sentiment insights, thereby enhancing its trustworthiness and practical value.

7. Conclusion & Future Work

This project successfully developed a sentiment analysis system that balances interpretability with practical usability. By leveraging Logistic Regression in combination with TF-IDF feature extraction, and deploying the model via a command-line interface (CLI), the system demonstrates strong performance on informal social media text while remaining accessible and transparent to end users.

The chosen methodology provides a solid and extensible foundation for further research and application. In particular, the simplicity of the model allows for easy integration, debugging, and real-time deployment in constrained environments. Furthermore, the CLI-based interaction highlights the feasibility of lightweight natural language processing tools in non-graphical environments such as servers or embedded systems.

Looking forward, several directions for future improvement have been identified:

- Dataset Expansion: Incorporating a larger and more diverse dataset—including multilingual corpora—will enhance the model's generalizability and cross-cultural relevance.
- Advanced Emotion Recognition: Implementing sarcasm and irony detection through deep learning techniques (e.g., transformer-based models like BERT) could significantly improve performance on complex sentiment expressions.
- User Interface Development: Transitioning from a CLI to a graphical user interface (GUI) or web-based platform will increase accessibility for non-technical users.

- Real-Time Integration: Connecting the system to live tweet streams via the Twitter API will enable real-time monitoring and analysis, opening possibilities for use in domains such as crisis response, political analysis, or market sentiment tracking.

By addressing these future directions, the system can evolve into a more comprehensive, user-friendly, and adaptable tool for sentiment analysis across multiple domains.

Acknowledgements

We would like to thank our course instructors and TA team for providing guidance throughout the project. Special thanks to our teammates:

黃凡嘉 (B11701165)
池昱東 (B10b02063)
李耔堉 (B12502015)
趙致毅

for their collaboration in data preparation, coding, and testing.

References

Training data:
https://www.kaggle.com/datasets/kazanova/sentiment140

Real data:
https://www.kaggle.com/datasets/abdelmalekeladjelet/sentiment-analysis-dataset

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Stanford University.

Sharma, P., & Kumar, S. (2023). Using Classifier Ensembles to Predict Election Results Using Twitter Data Sentiment Analysis. Springer Nature.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.