Indian Institute of Technology, Dharwad

॥ सा विद्या या विमुक्तये ॥
ಭಾ.ತಂ.ಸಂ. ಧಾರವಾಡ
भा. प्रौ. सं. धारवाड
I.I.T. DHARWAD

CS209 : Artificial Intelligence
And
CS214 : Artificial Intelligence Laboratory
Project Report :**Analysis of Public Sentiment in the 2024 U.S. Presidential Election**

Course Instructor:
Dr. Dileep A.D.

Mentor Name:
Suraj

Submitted by: Team 11

1. Tapash Hiren Darji
2. Dileepan S K
3. Gaddam Viswavijeth
4. Reddimalla Ganesh Reddy

Contents

List of Figures

List of Tables

1 Introduction

Social media is now a powerful lens through which we can witness the expression of public opinion and the engagement in political action, especially concerning significant occurrences like the 2024 U.S. Presidential Election. Social Media platforms such as X (formerly known as twitter) has become an avenue for dialogue and discussion, there are incredibly rich sources of information documenting how people feel, how they think, or what they do in connection to political events as they unfold. The focus of this project is to examine the digital conversations more closely in order ascertain what they tell us about voter mood and campaign impact.To do this, we have leveraged a dataset of tweets occurring during December 2024 and January 2025. We were interested in the relationship between engagement measures and measures of sentiment (e.g., likes and retweets) to see if we could also model and/or predict public mood from data-driven methodologies.We used a combination of statistical and machine learning methods such as, linear regression, neural network models (classification and regression), polynomial curve fitting, and K-Nearest Neighbor (KNN) methods. Natural Language Processing (NLP) methods assisted us in the preprocessing of the text-based data, allowing us to convert raw social media texts into useful and usable numerical features to be used in our models.

2 Methodology

In order to investigate how individuals were feeling and interacting online during the 2024 U.S. Presidential Election, we took a step-by-step approach that involved data cleaning, text processing, and an assortment of machine learning methods. Here is how we went about the analysis.

2.1 Dataset Description

The dataset contains the following key attributes:

- **Tweet Text** – The textual content of each tweet.
- **Timestamp** – The date and time when the tweet was posted.
- **User Handle** – The username of the account posting the tweet.
- **Sentiment Labels** – Categorical sentiment classification (Positive, Neutral, Negative).
- **Engagement Metrics** – Number of retweets and likes per tweet.
- **Party Affiliation** – The political party associated with the tweet's subject.

The dataset spans from January to February 2025, covering key moments in the final phase of the 2024 U.S. Presidential Election.

2.2 Data Preprocessing

We began by using a dataset of tweets gathered from X (formerly Twitter). Every tweet contained the user handle, tweet content, timestamp, the number of likes and retweets it had, its sentiment tag (positive, neutral, or negative), and the political party and candidate's name.

We had to clean things up before we could analyze anything. That involved dealing with any missing data, eliminating extraneous or noisy records, and standardizing everything to be in the same format. We also translated the sentiment labels into numeric form—using -1 for negative, 0 for neutral, and 1 for positive—to make them simpler to manipulate in our models. For ensuring fairness between features, we scaled the engagement data so that likes and retweets were on comparable scales.

2.3 Natural Language Processing

Because tweets consist of free-form text, we employed natural language processing (NLP) to convert that text into something our models could process. We first tokenized the tweets into separate words, then stripped out frequent stopwords and special characters that contribute little meaning.

Lastly, we transformed the cleaned text into numerical vectors through techniques such as TF-IDF or BERT embeddings, depending on what was best for the particular model. This allowed us to capture the essence of the sentiment and themes of the tweets in a format readable by machines.

2.4 Machine Learning Models

After preparing the data, we tried out a number of different machine learning models to examine and predict sentiment. Each model provided us with a different insight:

2.4.1 Linear Regression

- **Linear Regression** helped us analyze how sentiment, engagement metrics (likes and retweets), and the vectorized content of tweets are related.
- It works by finding the best-fitting straight line that models the relationship between input features and the target variable (sentiment).
- The algorithm assigns weights to each feature to minimize the prediction error between the actual and predicted sentiment values.
- During training, the model learns these weights using a portion of the dataset (training set).
- It is then evaluated on unseen data (test set) to check how well it generalizes and predicts sentiment based on new tweet data.
- This approach allows us to mathematically capture the influence of features like likes and retweets on sentiment.
- The general equation of the hyperplane used to fit the training data can be expressed as a linear combination of the input features.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Where:

- y is the predicted sentiment score,
- $x_1, x_2, \ldots, x_n$ are the input features (e.g., tweet engagement metrics),
- $\beta_0, \beta_1, \ldots, \beta_n$ are the model coefficients,
- ϵ is the error term.

2.4.2 Neural Network Regression

- **Neural Network Regression** offered a more flexible, non-linear approach to explore the same relationships, particularly when the data did not follow simple or linear patterns.
- Unlike linear regression, neural networks are capable of capturing complex interactions among features through multiple layers of computation.
- The predicted output $\hat{y}$ for a given input x is computed by passing the input through one or more hidden layers using activation functions:

$$\hat{y} = f(W_n \cdot f(W_{n-1} \cdot \ldots f(W_1 \cdot x + b_1) \cdots + b_{n-1}) + b_n)$$

where W_i and b_i are the weights and biases of each layer, and f is a non-linear activation function (e.g., ReLU, sigmoid).

2.4.3 Neural Network Classification

- **Neural Network Classification** enabled us to predict whether a tweet's sentiment was positive, negative, or neutral based on both engagement metrics and the textual content.
- This model uses multiple layers to capture non-linear patterns in the data and outputs a probability distribution over possible sentiment classes.
- The final layer applies the **softmax function** to convert raw output scores into class probabilities:

$$P(y = k \mid \mathbf{x}) = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

where:

- $P(y = k \mid \mathbf{x})$ is the probability that the input $\mathbf{x}$ belongs to class k,
- z_k is the output (logit) corresponding to class k,
- The denominator sums the exponentiated logits over all classes to normalize the output.

2.4.4 KNN Classifier

- **K-Nearest Neighbors (KNN)** classified sentiment by measuring similarity between tweets based on their features, making it a useful baseline for evaluating the performance of more complex models.
- The algorithm predicts the class of a tweet by looking at the most common sentiment among its k nearest neighbors in the feature space.
- Similarity is typically calculated using the **Euclidean distance** between feature vectors:

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2}$$

where:

- $\mathbf{x}$ and $\mathbf{x}'$ represent two tweets in the feature space,
- x_i and x'_i are the values of the i^{th} feature for each tweet.

2.4.5 Polynomial Curve Fitting

- **Polynomial Curve Fitting** was helpful in modeling how sentiment evolved over time, especially in cases where linear trends were too simplistic to capture the nuances in the data.
- This method fits a polynomial equation to the data, allowing us to capture non-linear relationships between sentiment and variables such as time or engagement.
- The general form of a polynomial curve of degree n is:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon$$

where:

- y is the predicted sentiment value,
- x is the input feature (e.g., time or engagement metric),
- $\beta_0, \beta_1, \ldots, \beta_n$ are the learned polynomial coefficients,
- ϵ represents the random error or noise in the model.

2.5 Model Evaluation

In order to gauge how well our models were performing, we employed standard evaluation metrics that are commonly used for both regression and classification tasks. These metrics helped us quantify the accuracy and reliability of the predictions made by our models.

2.5.1 Evaluation for Regression Models

For regression models, we used two main metrics:

R-squared (R^2): R-squared is a measure of how well the model explains the variance in the dependent variable. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where: - y_i is the actual value, - $\hat{y}_i$ is the predicted value, - $\bar{y}$ is the mean of the actual values.

An R^2 value close to 1 indicates a good fit, while a value close to 0 means the model does not explain the variance well.

Mean Squared Error (MSE): MSE measures the average squared difference between the actual and predicted values. It provides an overall indication of the magnitude of error in the predictions. The formula for MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where: - y_i is the actual value, - $\hat{y}_i$ is the predicted value, - n is the number of data points.

Lower values of MSE indicate better performance, as it means the predictions are closer to the actual values.

2.5.2 Evaluation for Classification Models

For classification models, we used a set of metrics to evaluate their performance, especially when dealing with multiple sentiment classes (positive, negative, neutral).

Accuracy: Accuracy measures the proportion of correct predictions out of all predictions made. It is given by the formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Where: - TP = True Positive (correctly predicted positive sentiment), - TN = True Negative (correctly predicted negative or neutral sentiment), - FP = False Positive (incorrectly predicted positive sentiment), - FN = False Negative (incorrectly predicted negative or neutral sentiment).

Accuracy is a simple metric but may not be ideal when dealing with imbalanced classes.

Precision: Precision calculates the proportion of positive predictions that were actually correct. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Where: - TP = True Positive (correctly predicted positive sentiment), - FP = False Positive (incorrectly predicted positive sentiment).

High precision means that most positive predictions made by the model are correct.

Recall (Sensitivity): Recall measures the proportion of actual positives that were correctly identified by the model. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Where: - TP = True Positive (correctly predicted positive sentiment), - FN = False Negative (incorrectly predicted negative or neutral sentiment).

High recall indicates that the model is good at identifying positive cases, but it may also incorrectly classify some negatives as positives.

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's accuracy when dealing with imbalanced classes. It is given by:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score balances the trade-off between precision and recall. A higher F1-score means a better balance between these two metrics.

3 Result

We now present the results of our analysis. This section begins with the outcomes of the preprocessing and exploratory steps, followed by the performance evaluation of each individual model, and concludes with a comparative analysis of all models.

3.1 Preprocessing and Exploratory Analysis

Before training our models, we carried out preprocessing and exploratory analysis to better understand the dataset and prepare it for machine learning tasks. The following steps were performed:

- **Normalization of Engagement Metrics:** The number of likes and retweets were normalized to ensure consistency across features.
- **Conversion of Sentiment Labels:** Sentiment categories were converted into numerical values for model compatibility — negative as -1, neutral as 0, and positive as 1.
- **Trend Analysis:** We conducted time-series analysis using visual plots to observe how public sentiment evolved over the data collection period. To gain deeper insights, we also visualized sentiment trends for individual candidates, allowing us to compare public perception across the political spectrum.

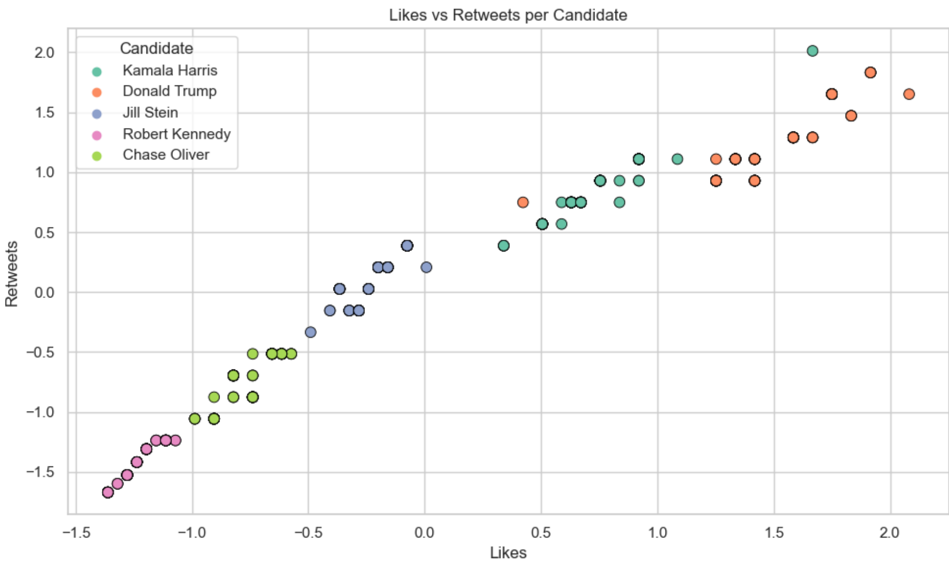

Figure 3.1: Likes vs Retweets for different candidates

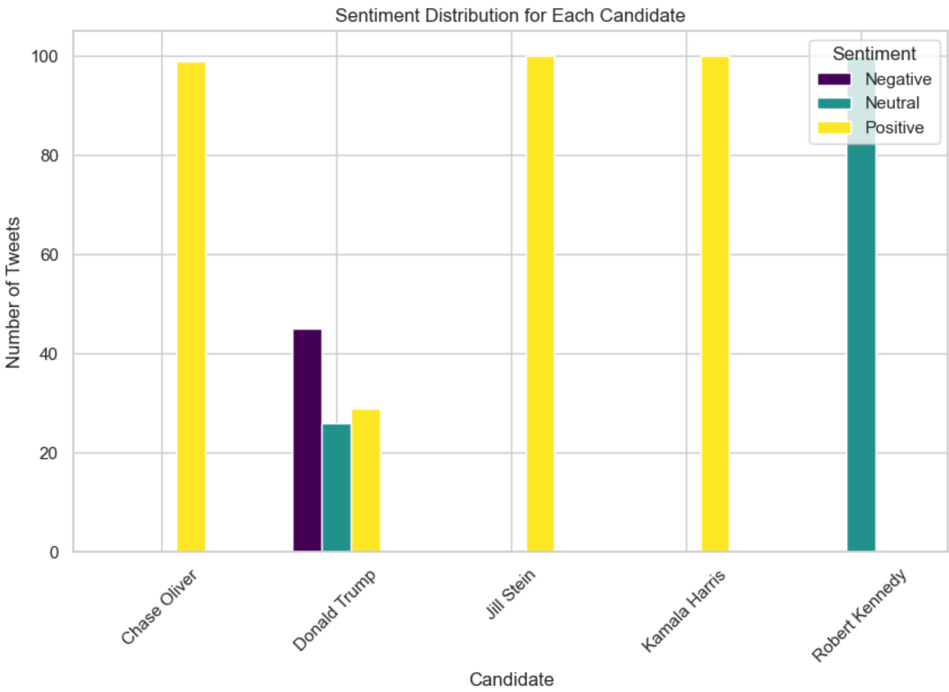

Figure 3.2: Sentiment Distribution for different candidates

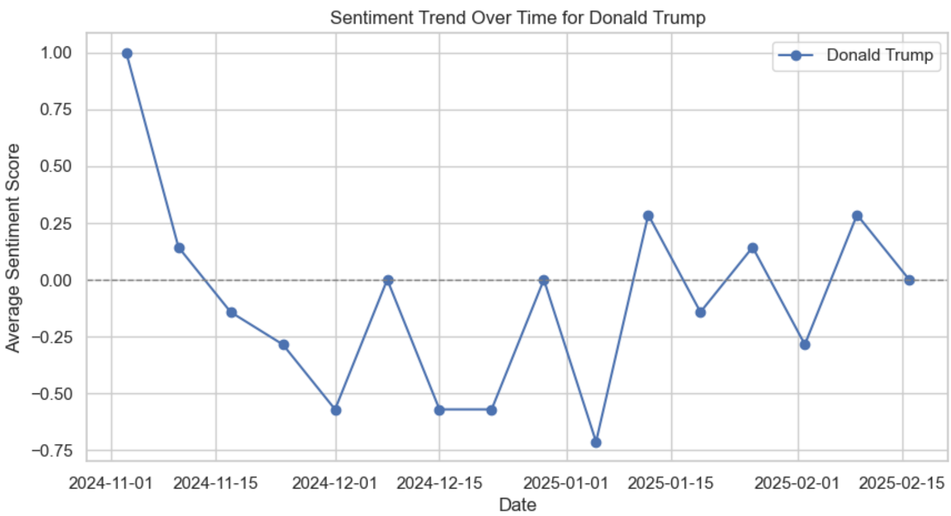

Figure 3.3: Average Sentiment Trend for Donald Trump over time

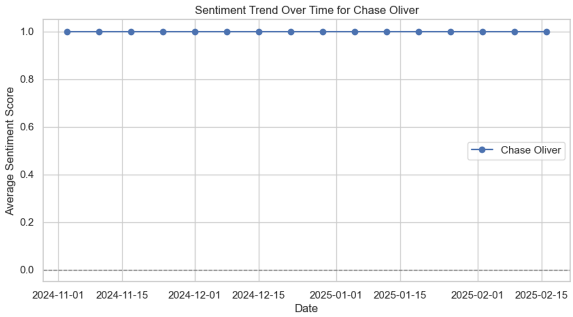

Figure 3.4: Average Sentiment Trend for Chase Oliver over time

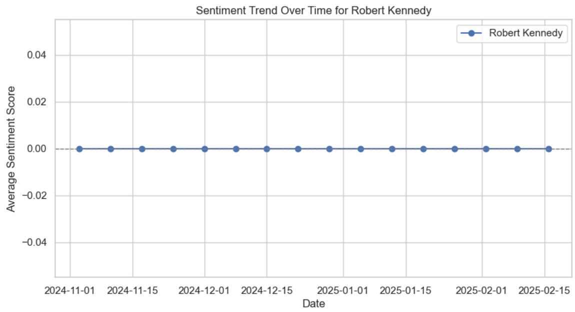

Figure 3.5: Average Sentiment Trend for Robert Kennedy over time

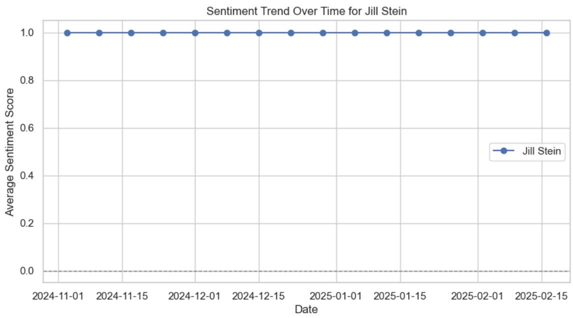

Figure 3.6: Average Sentiment Trend for Jill Stein over time

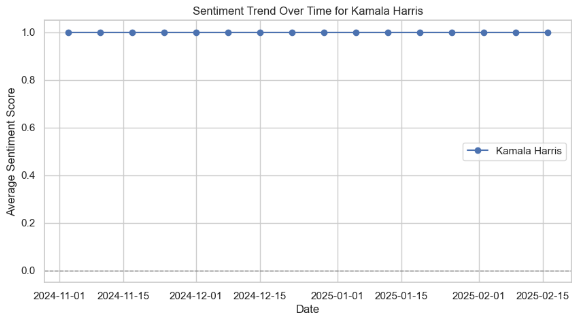

Figure 3.7: Average Sentiment Trend for Kamala Harris over time

Visual Analysis and Observations

- Sentiment Distribution:
  - Chase Oliver, Kamala Harris, and Jill Stein each have tweets that are overwhelmingly positive, with almost no neutral or negative sentiments recorded.
  - Donald Trump has the most diverse sentiment spread, with a significant proportion of negative, neutral, and positive tweets—indicating polarized public opinion.
  - Robert Kennedy's tweets are predominantly neutral, with no notable positive or negative sentiment in the distribution.
- Sentiment Trends Over Time:
  - Chase Oliver, Kamala Harris, and Jill Stein maintain a consistently high sentiment score of 1.0 across the observed time period, suggesting sustained positive engagement.
  - Robert Kennedy's sentiment score remains flatlined at 0, indicating tweets are consistently neutral with no emotional or opinionated tone.
  - Donald Trump exhibits high variability in sentiment over time-Early peaks near +1 are followed by sharp drops, with several weeks of negative sentiment.This fluctuation points to shifting public reactions or controversial discussions over time.

- Likes vs. Retweets Distribution:
  - Donald Trump and Kamala Harris are associated with tweets that receive higher engagement (likes and retweets), indicating greater user interaction and possibly wider reach.
  - Jill Stein occupies a middle ground, with moderate levels of engagement.
  - Robert Kennedy and Chase Oliver show lower engagement, clustering in the bottom-left of the plot—fewer likes and retweets suggest less viral or impactful content on Twitter.

3.2 Linear Regression Model

We developed two linear regression models to analyze the relationship between sentiment scores and various features. The first model explored the relationship between sentiment scores, tweet text (represented as numerical vectors), and engagement metrics (likes and retweets). The second model focused solely on the relationship between sentiment scores and engagement metrics.

The performance of both models was evaluated using standard regression metrics: R-squared (R^2) and Mean Squared Error (MSE).

Model	**NLP Valid**	**NLP Test**	**NO NLP Valid**	**NO NLP Test**
Unprocessed MSE	1.36×10^{-26}	1.36×10^{-26}	0.24	0.24
Processed MSE	2.97×10^{-5}	2.97×10^{-5}	0.24	0.24
Unprocessed R Square	1.0	1.0	0.34	0.34
Processed R Square	0.99	0.99	0.34	0.34

Table 3.1: Comparison of Linear Regression model performance with and without NLP features

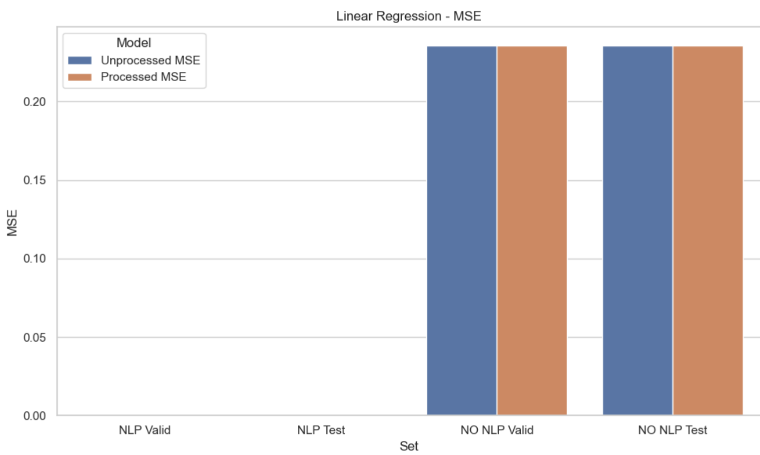

Figure 3.8: Comparison of MSE in Linear Regression Models with and without NLP Processing

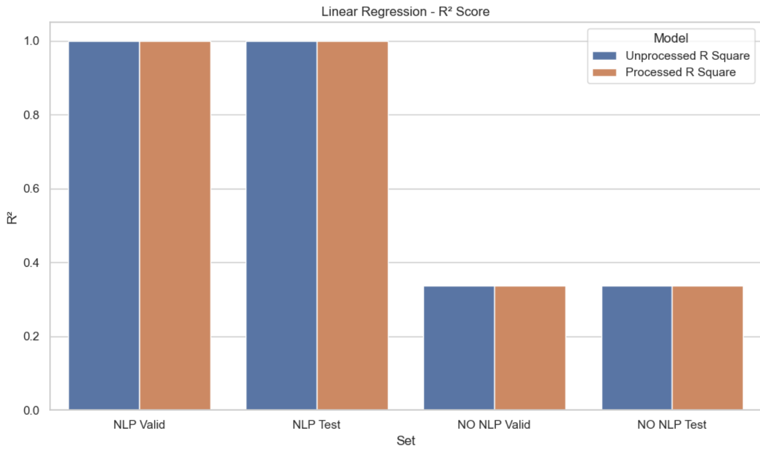

Figure 3.9: Comparison of R^2 Scores for Linear Regression with and without NLP Processing

The performance of the linear regression models under different configurations is summarized in the plots above. We evaluated four experimental settings based on two axes: whether the tweet text was processed or unprocessed, and whether NLP features were included or excluded.

From the MSE plot, we observe that:

- When NLP features were included, both processed and unprocessed text yielded extremely low MSE values, with the unprocessed variant achieving a near-zero error. This suggests a near-perfect fit, although it may also indicate potential overfitting.
- In contrast, when NLP features were not used, MSE values were significantly higher (0.235), indicating a much weaker model performance. Notably, the processed version performed marginally better than the unprocessed version in this setting.

From the R-squared plot, we note that:

- With NLP, both models achieved near-perfect R^2 values, again highlighting a strong correlation between the input features (including vectorized tweet text) and sentiment scores.
- Without NLP, the R^2 values dropped to approximately 0.337, confirming that engagement metrics alone (likes and retweets) were not strong predictors of sentiment in comparison to text-based features.
- The slightly higher R^2 for the processed text (0.3378) compared to the unprocessed version (0.3374) suggests that basic preprocessing adds some value even when text features are not directly used.

Overall, these results emphasize the importance of incorporating textual information through NLP techniques for effective sentiment prediction using linear regression. They also indicate that simple engagement metrics alone are insufficient to capture the complexity of public sentiment.

3.3 KNN Classifier Model

To evaluate the effectiveness of the K-Nearest Neighbors (KNN) classifier in predicting tweet sentiment, we tested the model across four configurations—using both preprocessed and unprocessed text, with and without NLP features. The plots and table below summarize the performance in terms of accuracy, precision, recall, and F1-score.

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Unprocessed + NLP	100	100	100	100
Processed + NLP	100	100	100	100
Unprocessed - NLP	98	98.14	98	98.02
Processed - NLP	98	98.05	98	98

Table 3.2: Comparison of KNN model performance with and without NLP features

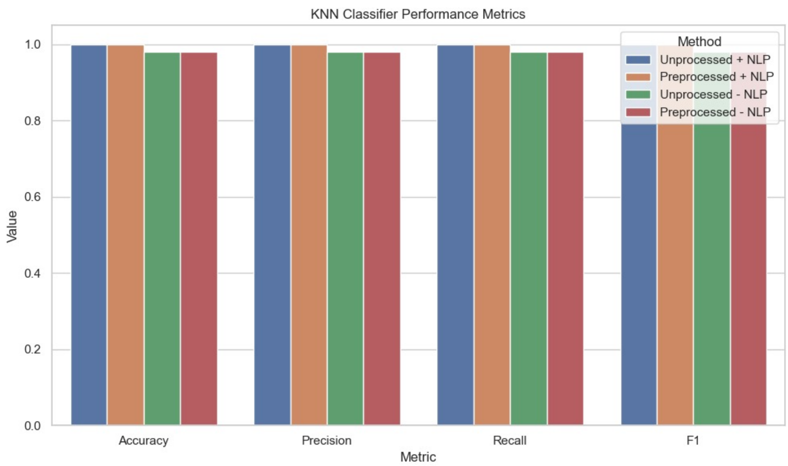

Figure 3.10: Comparison of Accuracy, Precision, Recall, F1-Score in KNN Classifier Model with and without NLP Processing

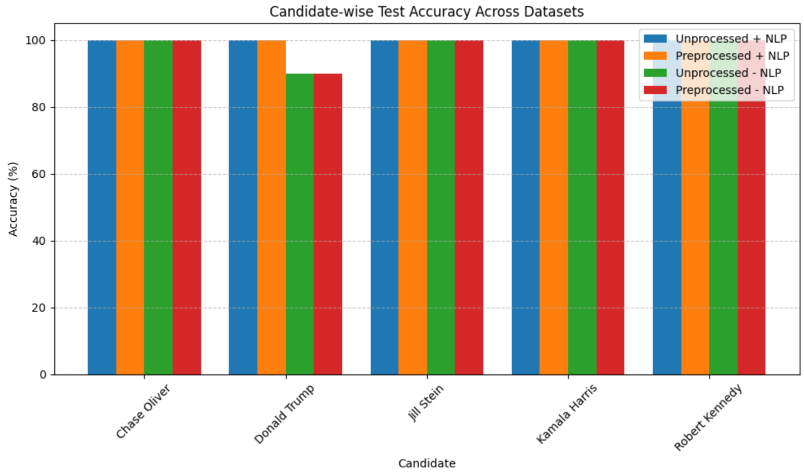

Figure 3.11: Comparison of Accuracy for each individual candidate in KNN Classifier Model with and without NLP Processing

The KNN classifier performed exceptionally well across all configurations. When NLP features were included, both unprocessed and preprocessed text variants achieved perfect scores—accuracy, precision, recall, and F1-score all reached 1.0, indicating flawless sentiment classification.

In contrast, when NLP features were excluded, performance dropped slightly. All metrics were around 0.98, showing that while engagement metrics alone can provide reasonable predictions, they are less effective than text-based features.

These results highlight the significant impact of NLP in sentiment classification and show that KNN can perform reliably even on unprocessed text.

3.4 Neural Networks Regression Model

To assess the performance of the Neural Network Regression model, we evaluated it under four experimental conditions—using preprocessed and unprocessed text, both with and without NLP features. The model's effectiveness was measured using Mean Squared Error (MSE) and R-squared (R^2) scores for both validation and test sets. The results are summarized below.

Model	NLP Valid	NLP Test	NO NLP Valid	NO NLP Test
Unprocessed MSE	0.20×10^{-3}	0.20×10^{-3}	0.15	0.15
Processed MSE	0.14×10^{-3}	0.14×10^{-3}	0.15	0.15
Unprocessed R Square	0.99	0.99	0.58	0.58
Processed R Square	0.99	0.99	0.57	0.57

Table 3.3: Comparison of Neural Network Regression model performance with and without NLP features

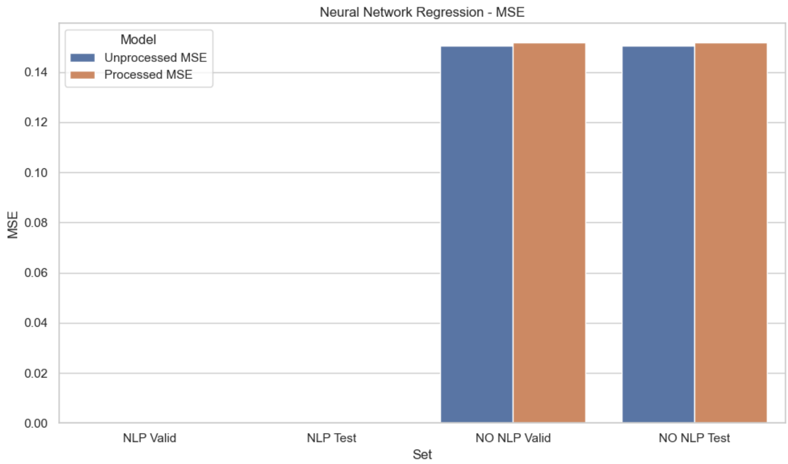

Figure 3.12: Comparison of MSE in Neural Network Regression Models with and without NLP Processing

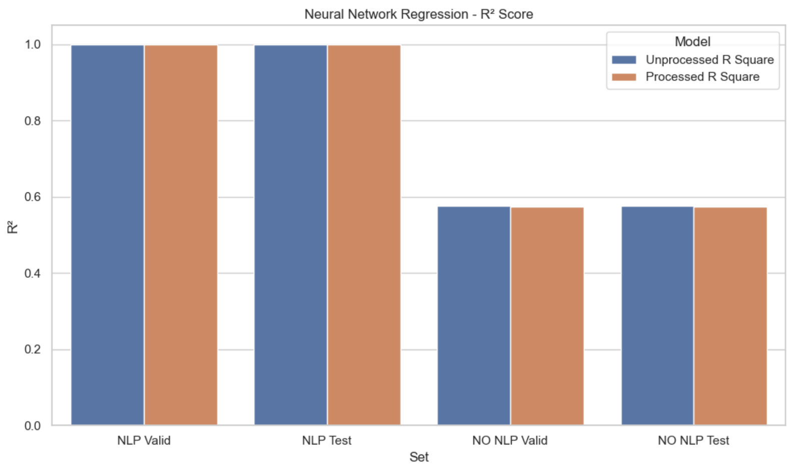

Figure 3.13: Comparison of R^2 Scores for Neural Network Regression with and without NLP Processing

The neural network regression model demonstrated excellent performance when NLP features were included. Both unprocessed and preprocessed versions achieved near-perfect results, with R^2 scores above 0.999 and extremely low MSE values, indicating a strong fit and highly accurate predictions.

In contrast, models without NLP features showed noticeably lower performance. R^2 scores dropped to around 0.57, and MSE values increased significantly, highlighting the limitations of using only engagement metrics to predict sentiment.

These results confirm that incorporating textual features is crucial for accurate sentiment regression, and that neural networks can effectively capture the complex relationships in text-based data.

3.5 Neural Network Classifier Model

We developed a neural network classifier to predict the political affiliation of tweets based on a combination of engagement metrics and textual data. To evaluate the impact of natural language processing (NLP) features, we trained three variants of the model: one using only numerical engagement metrics (likes and retweets), another incorporating raw tweet text alongside the numerical features, and a third using preprocessed text. The models were assessed using standard classification metrics: accuracy, precision, recall, and F1-score, along with candidate-wise performance analysis.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Unprocessed + NLP	100	100	100	100
Processed + NLP	100	100	100	100
Unprocessed - NLP	86	93.38	86	88.67
Processed - NLP	84	90.73	84	86.73

Table 3.4: Comparison of Neural Network Classifier model performance on Test Data with and without NLP features

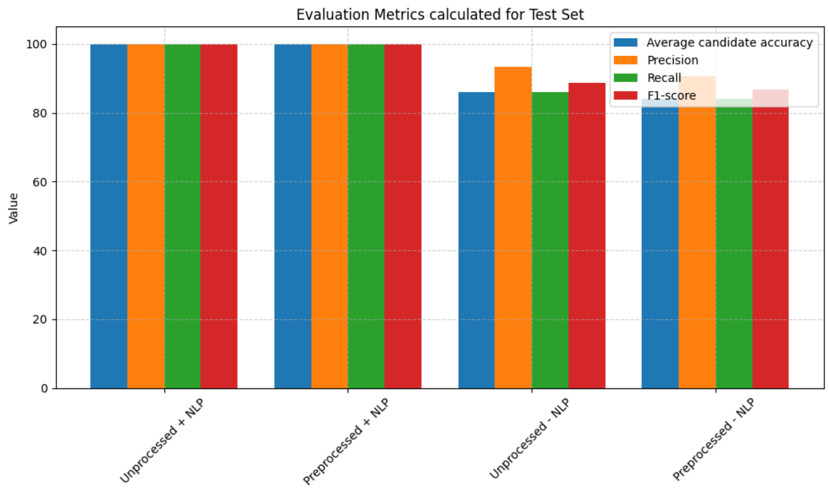

Figure 3.14: Comparison of Accuracy, Precision, Recall, F1-Score in Neural Network Classifier Model with and without NLP Processing

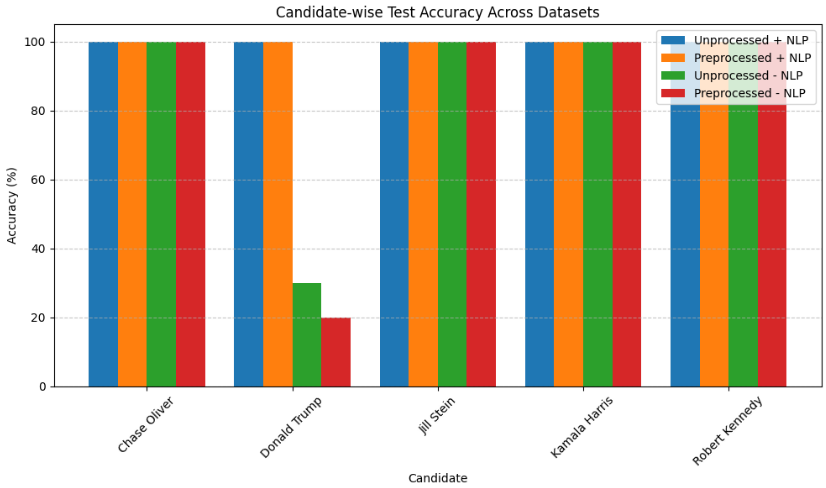

Figure 3.15: Comparison of Accuracy for each individual candidate in Neural Network Classifier Model with and without NLP Processing

Incorporating NLP features into the neural network classifier significantly enhances its performance across all key evaluation metrics. Models utilizing NLP, whether with raw or preprocessed text, consistently achieve perfect scores in accuracy, precision, recall, and F1-score, indicating highly effective and reliable classification. In contrast, models without NLP features exhibit a noticeable drop in performance, particularly in precision and recall, highlighting the importance of textual information in capturing underlying patterns. Furthermore, the candidate-wise accuracy comparison reveals that the inclusion of NLP features leads to more consistent and robust predictions across all classes, whereas the absence of NLP results in greater variability and reduced effectiveness for certain candidates. Overall, these findings reinforce the critical role of NLP processing in improving both the accuracy and generalizability of the neural network classifier.

3.6 Polynomial Regression Model

We developed polynomial regression models to predict sentiment scores using a combination of engagement metrics and tweet text. One set of models incorporated textual data, both in its raw and preprocessed forms, alongside numerical features such as likes and retweets. Another model was trained using only the numerical features, excluding all textual information. The performance of these models was evaluated using standard regression metrics: R-squared (R^2) and Mean Squared Error (MSE).

Model	NLP Valid	NLP Test	NO NLP Valid	NO NLP Test
Unprocessed MSE	0.03	0.03	0.03	0.03
Processed MSE	0.029	0.034	0.12	0.12
Unprocessed R Square	0.97	0.94	0.91	0.91
Processed R Square	0.98	0.91	0.66	0.66

Table 3.5: Comparison of Polynomial Regression model performance with and without NLP features

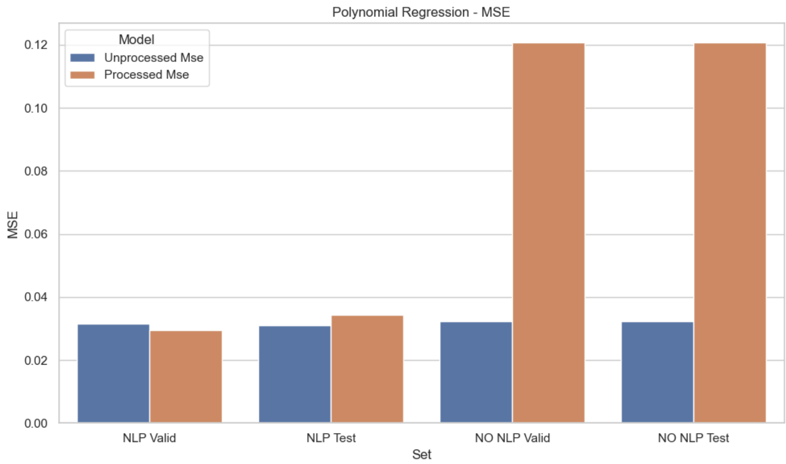

Figure 3.16: Comparison of MSE in Polynomial Regression Models with and without NLP Processing

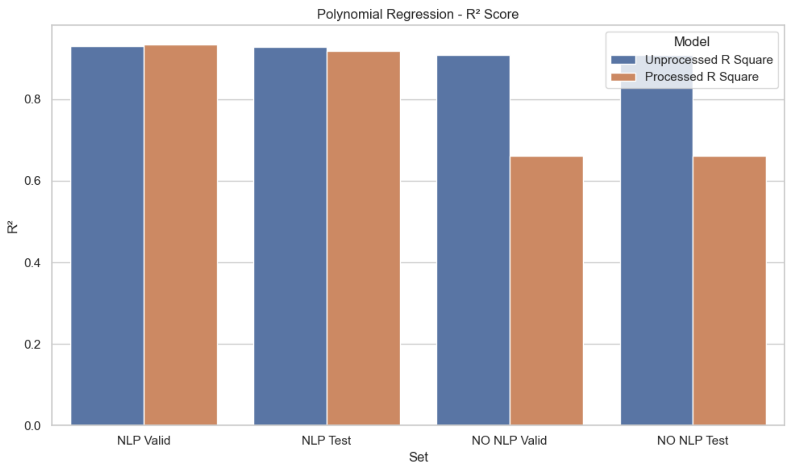

Figure 3.17: Comparison of R^2 Scores for Polynomial Regression with and without NLP Processing

The incorporation of NLP features in polynomial regression models leads to notable improvements in performance, as evidenced by both lower Mean Squared Error (MSE) and higher R^2 scores. Models that include either raw or preprocessed textual data consistently outperform those without NLP features, indicating that the presence of text contributes significantly to the model's ability to capture underlying patterns. The similarity in performance between processed and unprocessed NLP inputs suggests that even minimal handling of text data can yield substantial benefits. Additionally, the close alignment of results across validation and test sets reinforces the reliability and generalizability of models enhanced with NLP, while models without text inputs demonstrate higher errors and reduced explanatory power, underscoring the value of incorporating textual features into regression tasks.

3.7 Comparison of Classification and regression models

In this section, we present a comparative analysis of different machine learning models applied to both regression and classification tasks. The performance of regression models is evaluated using Mean Squared Error (MSE) and R-Squared (R^2) metrics, while classification models are assessed based on their prediction accuracy.

For regression, Linear Regression, Neural Network, and Polynomial Regression models were trained and tested on two datasets: one with NLP features and one without. Similarly, classification performance was compared between K-Nearest Neighbors (KNN) and Neural Network models under four preprocessing strategies (with/without NLP and preprocessed/unprocessed).

The plots below illustrate the effectiveness of each model and preprocessing technique across these tasks.

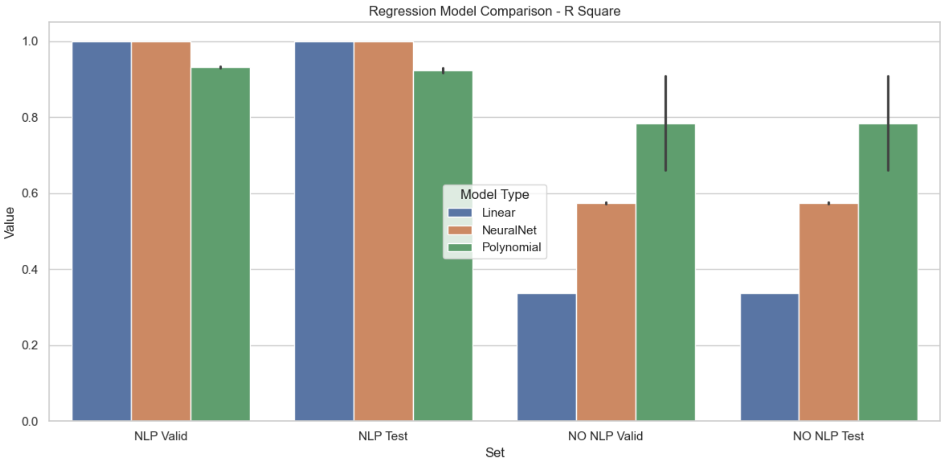

Figure 3.18: Comparison of R^2 scores of all the models with and without NLP Processing

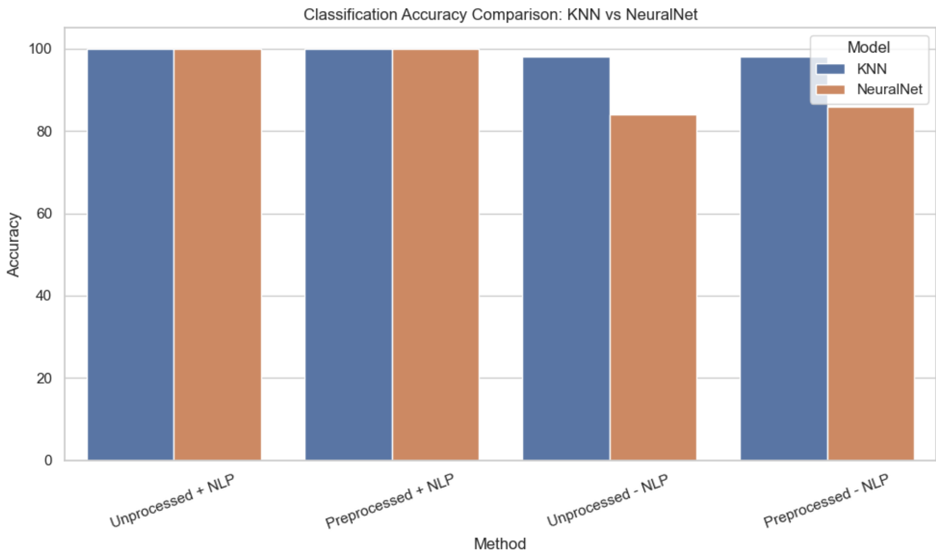

Figure 3.19: Comparison of accuracy scores for all the models with and without NLP Processing

Analysis and Observations

From the regression model comparison:

- **Linear Regression** and **Neural Networks** both achieve high R^2 values (close to 1.0) when NLP features are used, indicating excellent predictive power in those scenarios.
- **Polynomial Regression** performs slightly below the other two models when NLP is included, with R^2 values around 0.93–0.94, but shows strong robustness in non-NLP settings, maintaining higher R^2 (0.78) than both NeuralNet and Linear models.
- Without NLP features, **Linear Regression** performs the worst (R^2 0.33), while **Neural Networks** perform moderately (R^2 0.57), and **Polynomial Regression** maintains reasonable predictive power with some variance.

In the classification comparison:

- **KNN and Neural Networks both achieve near-perfect accuracy** (close to 100%) in cases where NLP features are used—regardless of preprocessing—indicating that the presence of NLP features dominates model performance.
- In the absence of NLP features, **KNN still performs very well** (accuracy 98%), while **Neural Networks show a notable drop in accuracy** (down to 84–86%), suggesting KNN's robustness in low-feature settings.
- Contrary to common assumptions, **KNN slightly outperforms Neural Networks in NLP-absent scenarios**, showcasing its strength in high-bias, low-variance configurations.
- The effect of preprocessing is minimal across all configurations, emphasizing that NLP features are the primary drivers of classification accuracy.

These results highlight that while deep models like Neural Networks perform exceptionally well when supplied with informative features (like NLP-derived inputs), simpler models such as KNN and Polynomial Regression can outperform or match them in more constrained or lower-dimensional environments. Therefore, model selection should align with feature richness and task complexity.

4 Conclusion

This project aimed to analyze the sentiment of tweets related to various political candidates using different machine learning models. Here are the key takeaways and insights:

4.1 Mixed Opinions on Donald Trump

Out of all the candidates, Donald Trump stood out for having mixed sentiment tweets . This is likely due to the timing of the data collection, which probably occurred shortly after the elections. During such periods, public opinion tends to be divided, especially when analyzing the sitting president.

4.2 Models used

We experimented with and evaluated five different models in this project:

- Linear Regression
- Polynomial Regression
- Neural Network (Classification)
- Neural Network (Regression)
- K-Nearest Neighbors (KNN) Classifier

4.3 Top Performing Models

- Across all models, incorporating NLP-based vectorization significantly improved performance. The added semantic information helped models better understand the context and sentiment embedded in the tweets.
- When NLP vectorization was not used, simpler models such as **K-Nearest Neighbors (KNN)** and **Linear Regression** often outperformed more complex ones. These models relied more effectively on basic numerical features like engagement metrics (likes, retweets).
- This comparative analysis highlights the importance of both feature engineering and model selection in achieving optimal performance. It also emphasizes how simpler models can sometimes be more effective depending on the nature of the input data.

4.4 Trend Observation

- For most candidates, tweets generally leaned towards positive or neutral sentiment, suggesting that public opinion was more aligned or apathetic towards them.
- Negative sentiment was notably rarer, except in Trump's case.
- There was a noticeable correlation between high tweet engagement (retweets, likes) and positive sentiment, although this wasn't always the case—especially for more controversial figures.

In conclusion, this project demonstrated the potential of basic machine learning models like KNN and Linear Regression in analyzing social media sentiment. The findings provide valuable insights into how public sentiment can be gauged through social media interactions, offering a clearer picture of political discourse, especially during times of heightened political tension.