Text Sentiment Analysis using Machine Learning

Project Overview:

This project focused on performing sentiment analysis on a text dataset containing various sentiments, including "positive," "negative," and "neutral." The goal was to train and evaluate multiple machine learning models to classify text sentiments accurately.

Total Data: The dataset consisted of 27,480 samples.

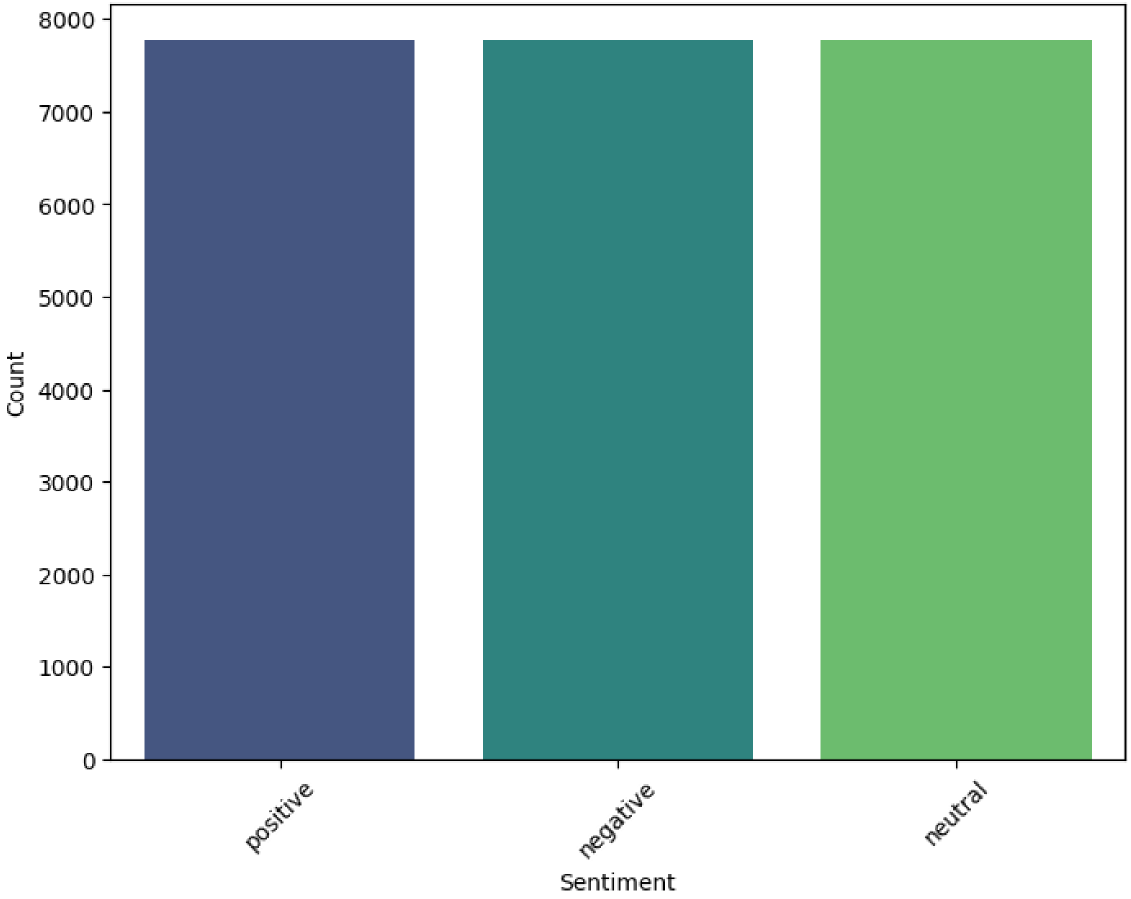
Data Cleaning: The following preprocessing steps were applied to the data:

- Removal of special characters.
- Conversion of text to lowercase.
- Elimination of stopwords.
- Removal of duplicate entries.
- Removal of empty data points.
- Cleaning of HTML tags.

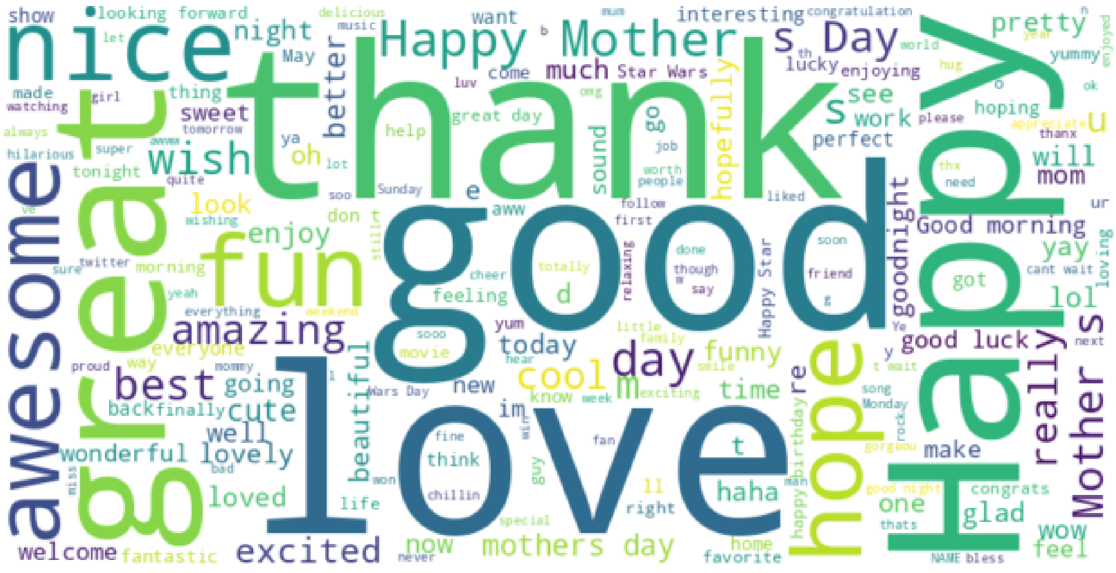
Data Splitting: The dataset was divided into a training set and a testing set to facilitate model training and evaluation.

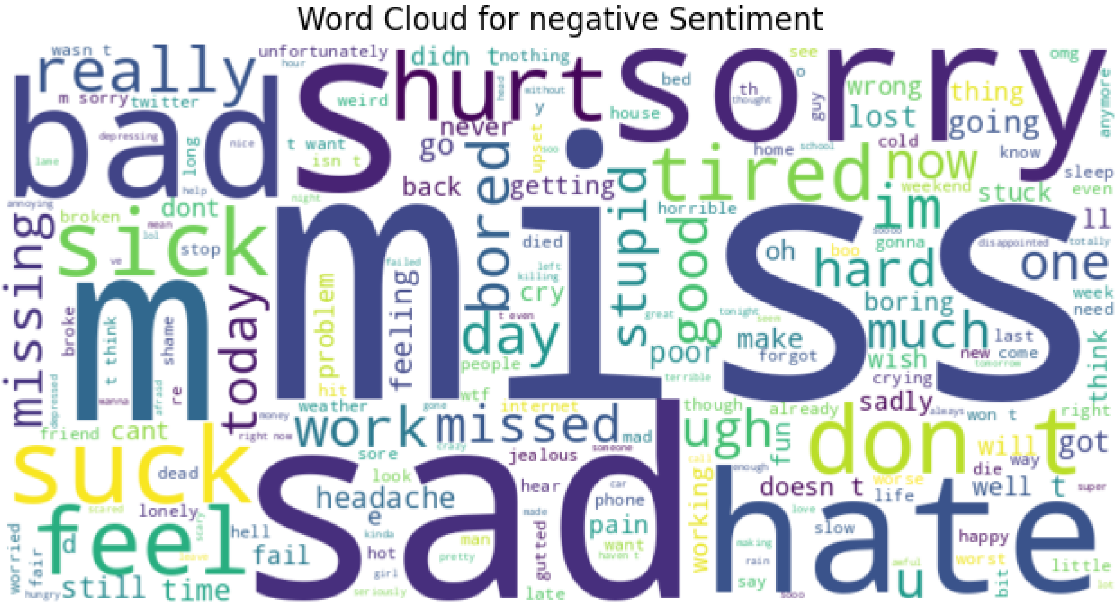
Feature Extraction: Count Vectorization was employed to convert text data into numerical format, making it suitable for machine learning.

Sentiment Distribution

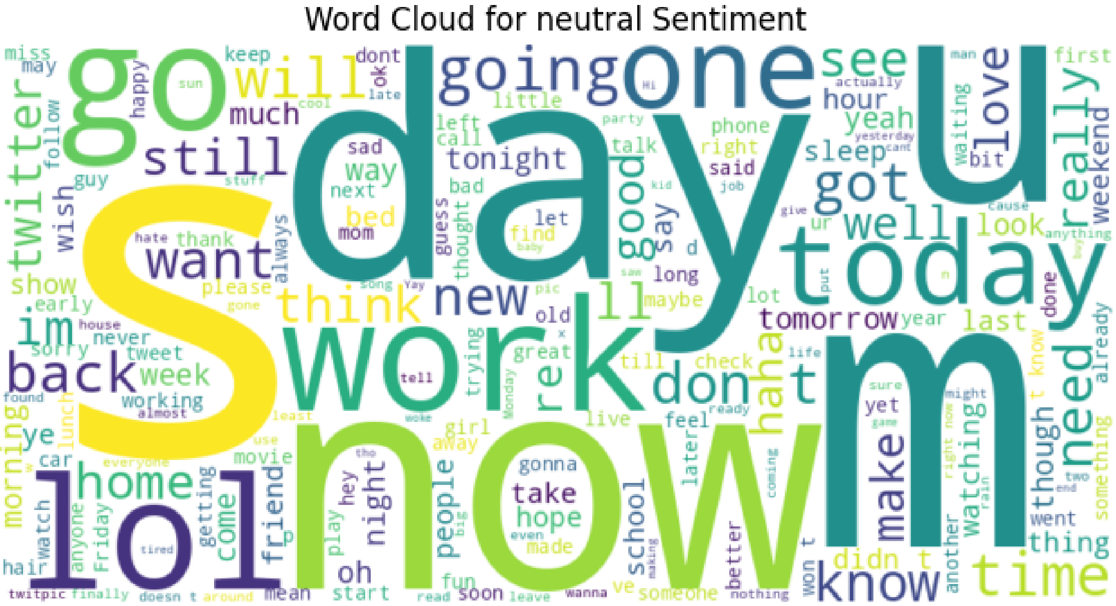


Word Cloud for positive Sentiment

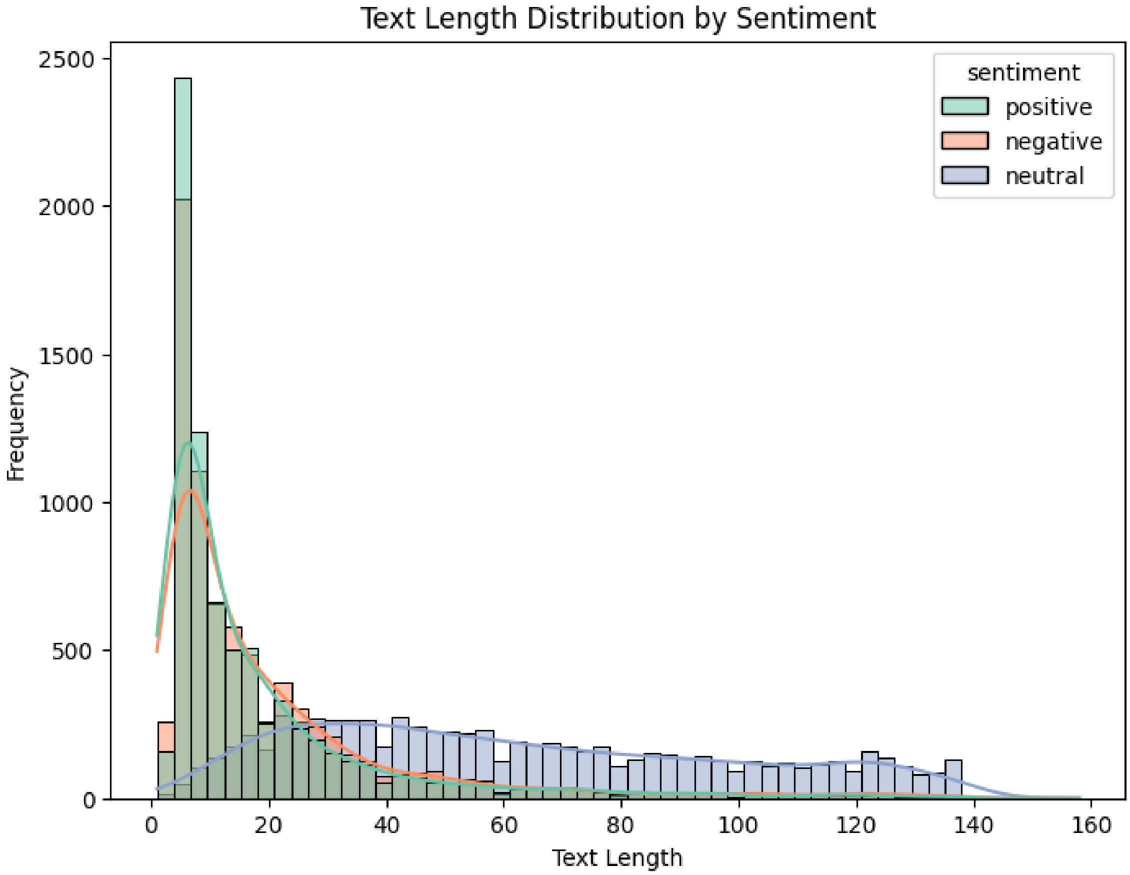




Word Cloud for negative Sentiment



Word Cloud for neutral Sentiment



Model Selection and Experimentation:

To identify the most suitable model for sentiment analysis, several machine learning algorithms were trained and evaluated on the preprocessed dataset.

Model Selection:

After thorough experiments and evaluations, the Random Forest Classifier was selected as the best-performing model for sentiment analysis.

Model Evaluation and Results:

The Random Forest Classifier achieved an accuracy of **70%** on the test data, indicating its effectiveness in classifying text sentiments. Additional performance metrics, such as precision, recall, and F1 score, can be calculated to provide a more comprehensive evaluation.

Precison:[0.73501763, 0.79440154, 0.80706344]

Challenges and Limitations:

Data Volume: The size of the original dataset (27,480 samples) posed challenges in terms of time and memory requirements. To mitigate this, a subset of 23,340 samples was selected for analysis. However, this data reduction could potentially impact the model's ability to generalize and achieve higher accuracy.

Computational Resources: Text classification tasks often require substantial computational resources, especially when dealing with large datasets. Model training and evaluation can be time-consuming and memory-intensive, necessitating access to suitable hardware.

Hyperparameter Tuning: The selected Random Forest Classifier may benefit from hyperparameter tuning to further enhance its performance. Fine-tuning parameters and conducting a more exhaustive search may yield even better results.

Conclusion:

In conclusion, this text sentiment analysis project demonstrated the capabilities and limitations of using machine learning for sentiment classification. The selected Random Forest Classifier achieved a respectable accuracy of 70%, which is promising for sentiment analysis tasks. However, challenges related to data volume, computational resources, and the need for more advanced feature engineering techniques were encountered.

Future work in this domain could involve exploring more advanced models, addressing class imbalances, and optimizing the preprocessing steps to achieve higher accuracy. Additionally, efforts to secure additional computational resources could be beneficial in handling larger datasets and conducting more extensive model experiments.