Sentiment Analysis Using NLP and Machine Learning

This report details a mini-project focused on building an end-to-end sentiment classification system. We processed textual data, extracted features using TF-IDF, and evaluated multiple machine learning models to identify the best-performing algorithm for classifying sentiment as Positive, Negative, or Neutral.

Project Overview and Objectives

Sentiment Analysis is a crucial task in Natural Language Processing (NLP), aimed at discerning the emotional tone within text. Its applications span social media monitoring, customer feedback analysis, and brand reputation management. This project developed a comprehensive sentiment classification system, leveraging feature extraction via TF-IDF and evaluating several machine learning models to determine optimal performance.

Preprocessing & Cleaning

Prepare raw text data for machine learning algorithms.

Feature Vectorization

Convert text into numerical features using TF-IDF.

Model Training

Train and compare various ML models for sentiment classification.

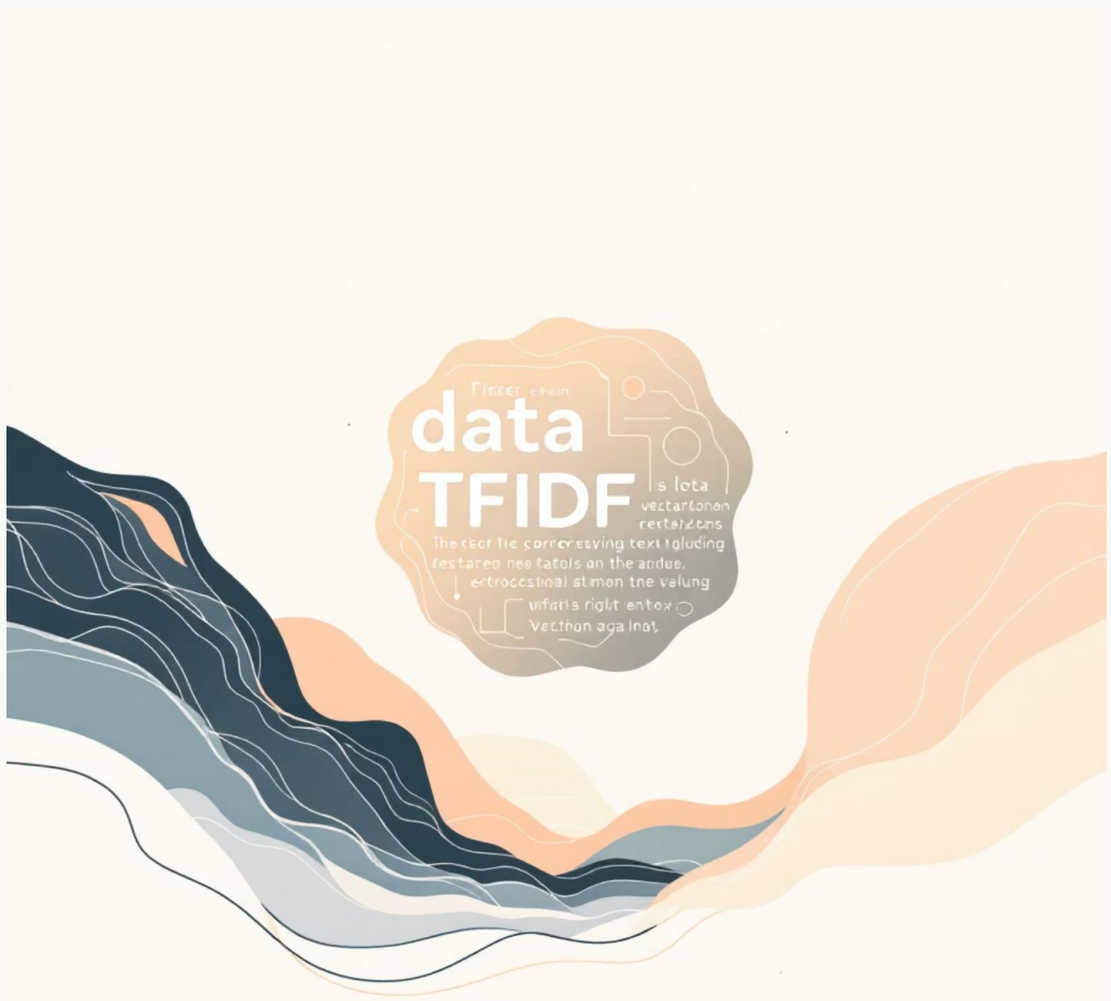
Performance Evaluation

Assess model performance using accuracy, reports, and confusion matrices.

The primary goal was to preprocess and clean raw text, convert it into numerical features using TF-IDF, and then train and compare multiple machine learning models for sentiment classification. We aimed to evaluate model performance comprehensively using metrics like accuracy, classification reports, and confusion matrices. Ultimately, the project sought to identify the best-performing model among Logistic Regression, Linear SVC, Naive Bayes, and Random Forest for this specific dataset.

Dataset Preparation and Preprocessing

The initial dataset comprised four columns: id, entity, sentiment, and tweet. For this project, columns were streamlined for consistency, retaining only the 'text' and 'sentiment' fields. A critical step involved mapping the original sentiment categories into three main classes: Positive, Negative, and Neutral. The 'Irrelevant' sentiment was merged into 'Neutral' to simplify the classification task and enhance model performance. Additionally, any rows with missing or empty text were meticulously removed to ensure the integrity and cleanliness of the training data.



Data Transformation Steps:

- **Column Standardization:** Renamed columns and selected 'text' and 'sentiment'.
- **Sentiment Mapping:** Consolidated sentiments into Positive, Negative, and Neutral (Irrelevant → Neutral).
- **Missing Value Handling:** Eliminated rows with empty or NaN text entries.

Feature Extraction with TF-IDF:

To prepare the textual data for machine learning algorithms, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was employed. This technique converts text into numerical representations, highlighting the importance of words within the context of the entire dataset.

- **Stop Words:** English stop words were removed.
- **Max Features:** Limited to 10,000 to manage dimensionality.

Model Training and Performance Evaluation

Four distinct machine learning models were selected for comparative analysis in sentiment classification: Logistic Regression, Linear Support Vector Classifier (Linear SVC), Multinomial Naive Bayes, and Random Forest Classifier. Each model underwent a rigorous training and evaluation process.

1

Logistic Regression

A fundamental model providing a solid baseline for comparison.

2

Linear Support Vector Classifier (Linear SVC)

Effective for high-dimensional data, focusing on clear margin separation.

3

Naive Bayes

A probabilistic classifier, often a strong contender for text data.

4

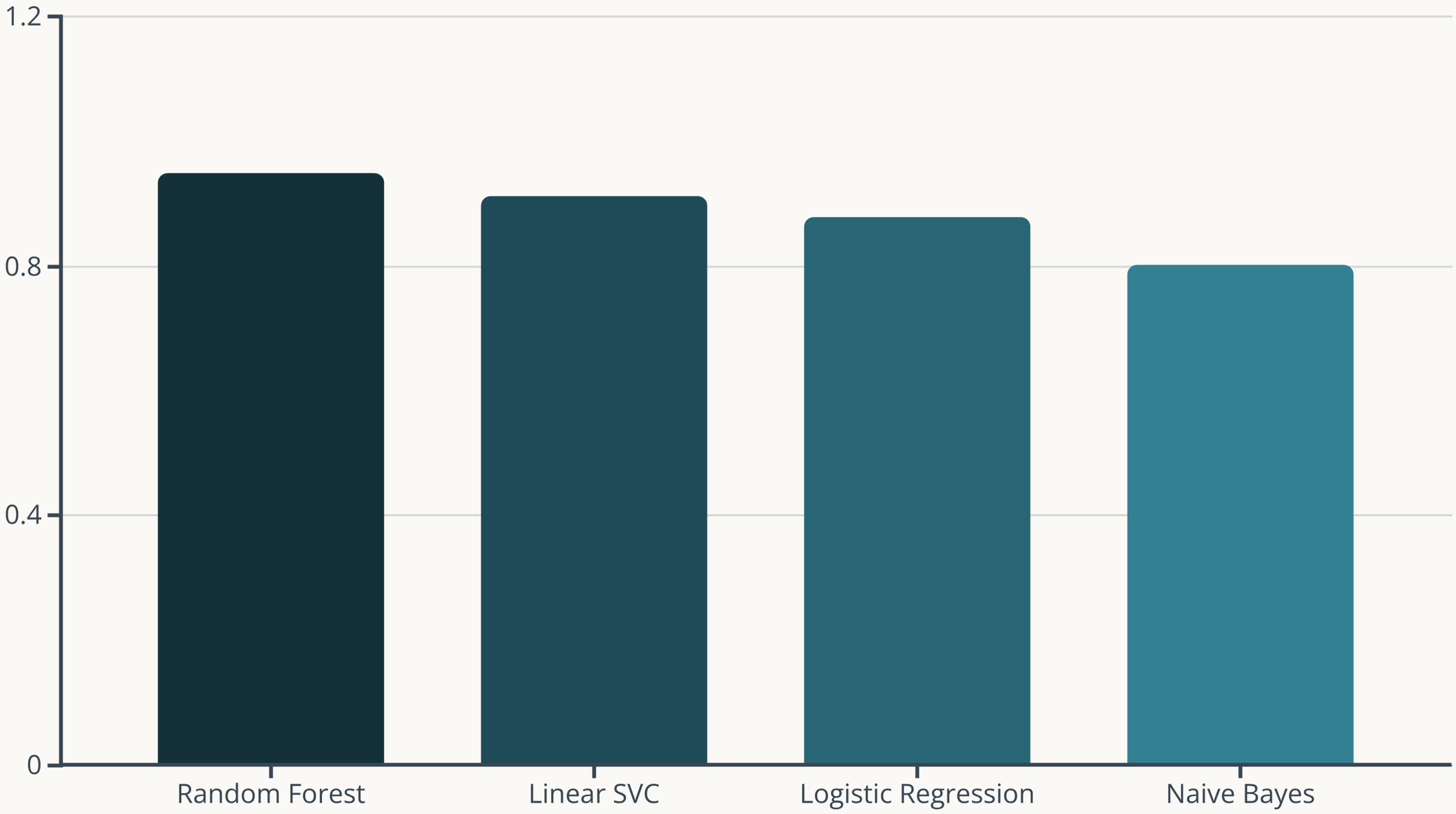
Random Forest Classifier

An ensemble method known for its robustness and ability to handle complex patterns.

The TF-IDF vectorizer was fitted on the training data's text column, and both training and validation sets were transformed into numerical matrices. Each model was then trained independently and evaluated using a suite of metrics. Accuracy served as the primary performance indicator, supplemented by precision, recall, and F1-score from detailed classification reports. Confusion matrices were also generated to provide visual insights into misclassification patterns.

Results and Conclusion

The evaluation results clearly indicated Random Forest as the top-performing model, showcasing its strong capability to learn non-linear patterns within the dataset. Linear SVC also delivered excellent performance, validating its suitability for high-dimensional sparse text data. Logistic Regression achieved moderate results, serving as a reliable baseline. Naive Bayes, however, performed the lowest, likely due to its inherent independence assumptions not fully holding true for real-world tweet data.



This project successfully demonstrated a complete NLP pipeline for sentiment analysis, from meticulous preprocessing to robust model comparison. The key takeaways include the superior performance of Random Forest, the reliable alternative offered by Linear SVC, the effectiveness of TF-IDF vectorization in feature conversion, and the significant impact of proper preprocessing on model performance. This mini-project effectively highlights the synergy between NLP and machine learning in addressing real-world text classification challenges.

Future Improvements

- **Deep Learning Integration**
  Explore models like LSTMs or BERT for enhanced contextual understanding.
- **Advanced Text Cleaning**
  Implement sophisticated handling for hashtags, emojis, and lemmatization.
- **Hyperparameter Tuning**
  Optimize model parameters for further gains in performance.
- **Expanded Dataset**
  Train on larger or more diverse datasets for improved generalization.