Sentiment Analysis Report

Debanjan Saha, Northeastern University, Boston

Introduction

Sentiment analysis is a natural language processing (NLP) task aimed at determining the sentiment or opinion expressed in a piece of text. In this report, we evaluate the performance of three popular classification algorithms, namely Naive Bayes, Logistic Regression, and Multilayer Perceptron (MLP), for sentiment analysis using two different feature representations: Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF). The goal is to analyze the effectiveness of each algorithm and the impact of feature representation on classification performance.

Methodology

Data Description

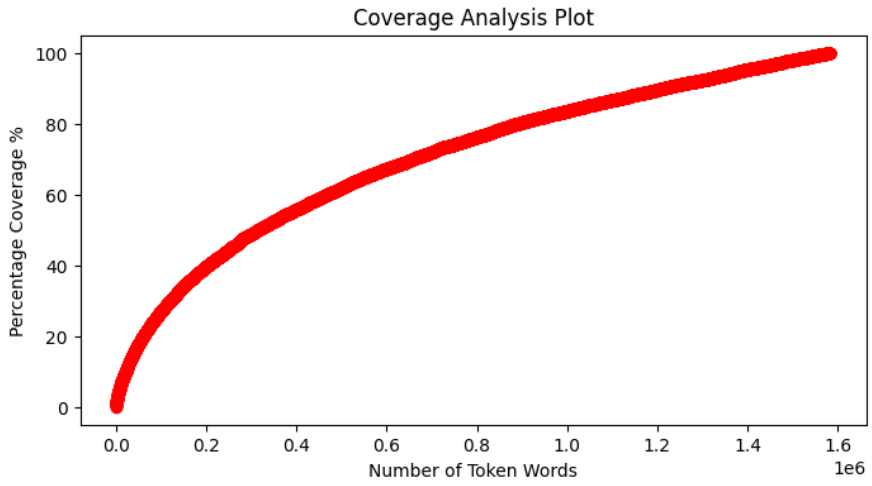
The dataset used for sentiment analysis consists of movie reviews labeled as positive or negative sentiments. Each review is preprocessed and represented as a feature vector using either TF or TF-IDF encoding.

Experimental Setup

1. **Coverage Analysis**
2. **Algorithms**:
    - Naive Bayes
    - Logistic Regression
    - Multilayer Perceptron (MLP)
3. **Feature Representations**:
    - Term Frequency (TF)
    - Term Frequency-Inverse Document Frequency (TF-IDF)
4. **Performance Metrics**:
    - Accuracy
    - True Positive Rate (TPR)
    - False Positive Rate (FPR)

Coverage Analysis

Following is a plot showing the coverage analysis to identify the percentage of unique words covered by the preprocessing of the document:



Discussions

Coverage analysis

The coverage analysis provides valuable insights into how the preprocessing steps affect the coverage of unique words in the dataset. Here are the observations based on the analysis:

- **How does the coverage change with the number of tokens considered?**

  Initially, as the number of tokens considered increases, the coverage percentage also increases rapidly. This is because adding more tokens introduces new unique words, thereby increasing coverage. However, as the number of tokens continues to increase, the rate of increase in coverage slows down.

- **At what point does the coverage seem to stabilize?**

  The coverage seems to stabilize at a certain point, indicating that adding more tokens beyond this point has diminishing returns in terms of covering unique words. After reaching this stabilization point, further increasing the number of tokens has minimal impact on increasing coverage.

- **Are there diminishing returns in terms of coverage as the number of tokens increases?**

  There are indeed diminishing returns in terms of coverage as the number of tokens increases. Initially, adding more tokens leads to a significant increase in coverage. However, as the vocabulary size grows larger, the incremental increase in coverage becomes smaller, indicating diminishing returns.

Rationalization for Vocabulary Choice

The choice of vocabulary size for modeling involves several considerations to strike a balance between informativeness and computational efficiency:

- **The trade-off between a larger vocabulary (more words) and computational efficiency.**

  A larger vocabulary (more words) increases the complexity of the model and computational requirements. On the other hand, a smaller vocabulary reduces the computational burden but may result in loss of important information. Therefore, choosing an optimal vocabulary size involves balancing computational efficiency with the need for informative features.

- **The impact of rare or very common words on the model's generalization.**

  Rare words may introduce noise into the model, while very common words may not provide much discriminatory power. Therefore, it's essential to consider the impact of rare and very common words on the model's generalization. Choosing an appropriate cutoff for including or excluding such words in the vocabulary can help improve model performance.

- **The need to balance informativeness and model complexity.**

  The vocabulary size directly impacts the model's complexity and its ability to generalize to unseen data. A vocabulary that is too small may result in underfitting, while a vocabulary that is too large may lead to overfitting. Therefore, it's crucial to strike a balance between informativeness and model complexity by choosing a vocabulary size that captures the essential information without introducing unnecessary complexity.

- **Any specific considerations for the chosen algorithms (Naive Bayes, Logistic Regression, MLP) in terms of vocabulary size.**

  Different algorithms may have different sensitivities to vocabulary size. For example, Naive Bayes classifiers rely on word frequencies and may benefit from a larger vocabulary to capture more nuanced patterns in the data. In contrast, logistic regression and MLP models may be more flexible in handling larger vocabularies but may also require more computational resources. Therefore, the choice of algorithms may influence the decision regarding the vocabulary size.

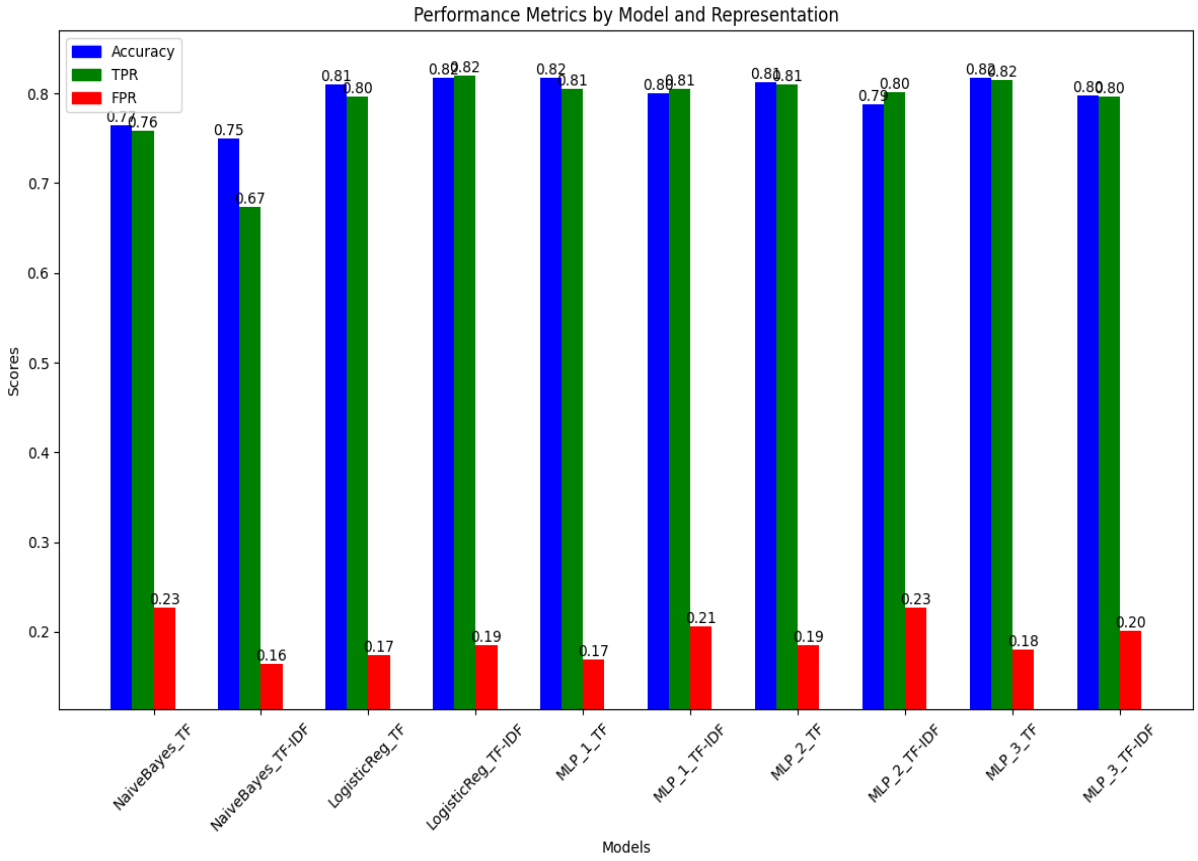
Results

Performance Metrics Overview

The table below summarizes the performance metrics for each algorithm with TF and TF-IDF representations:

Model	Accuracy	TPR	FPR
NaiveBayes_TF	0.7650	0.758	0.228
NaiveBayes_TF-IDF	0.7500	0.673	0.164
LogisticReg_TF	0.8100	0.796	0.175
LogisticReg_TF-IDF	0.8175	0.820	0.185
MLP_1_TF	0.8175	0.806	0.169
MLP_1_TF-IDF	0.8000	0.806	0.206
MLP_2_TF	0.8125	0.810	0.185
MLP_2_TF-IDF	0.7875	0.801	0.228
MLP_3_TF	0.8175	0.815	0.180
MLP_3_TF-IDF	0.7975	0.796	0.201

Comparative Analysis



Observed Trends and Differences in Performance:

Accuracy Trends:

- Logistic Regression consistently performs well, with both TF and TF-IDF representations yielding high accuracies.
- Naive Bayes also performs decently, but with slightly lower accuracies compared to Logistic Regression, especially with TF-IDF representation.
- MLP shows varying performance across different configurations, but generally achieves competitive accuracies.

True Positive Rate (TPR):

- Logistic Regression consistently exhibits high TPR values, indicating its effectiveness in correctly identifying positive sentiment.
- MLP also shows competitive TPR values, especially for certain configurations.
- Naive Bayes tends to have lower TPR values compared to the other algorithms, especially noticeable in the TF-IDF representation.

False Positive Rate (FPR):

- Naive Bayes exhibits higher FPR values across both TF and TF-IDF representations, indicating a tendency to incorrectly classify negative instances as positive.
- Logistic Regression and MLP generally have lower FPR values, suggesting better precision in distinguishing negative instances.

Comparison and Analysis of Algorithm Results:

Naive Bayes:

- Performs reasonably well but tends to have lower TPR and higher FPR compared to Logistic Regression and MLP.
- Simple and efficient, making it suitable for large datasets.
- However, it assumes independence among features, which may not hold true for natural language, impacting performance.

Logistic Regression:

- Shows consistently high performance across both TF and TF-IDF representations.
- Effective in modeling linear relationships between features and target variable.
- May suffer from overfitting if the feature space is large or if there are too many irrelevant features.

Multilayer Perceptron (MLP):

- Exhibits competitive performance, especially with certain configurations.
- Can capture complex non-linear relationships in the data, potentially improving performance.
- Requires careful tuning of hyperparameters and may be computationally expensive, especially with large datasets.

Impact of TF vs. TF-IDF on Classification Performance:

TF Representation:

- Generally yields lower accuracies compared to TF-IDF, especially for Naive Bayes.
- May be susceptible to the influence of frequent but less informative words (e.g., stopwords), affecting model performance.

TF-IDF Representation:

- Tends to improve classification performance, particularly for algorithms like Naive Bayes.
- Accounts for the importance of words in the corpus by weighing down frequent but less discriminative terms.
- Offers better generalization by reducing the impact of common words that may not contribute significantly to sentiment classification.

Strengths and Limitations of Each Algorithm

Naive Bayes:

- **Strengths:** Simple, fast, and easy to implement. Handles large datasets well.
- **Limitations:** Relies on the assumption of feature independence, which may not hold true for text data. Limited ability to capture complex relationships.

Logistic Regression:

- **Strengths:** Robust, interpretable, and performs well with linearly separable data.
- **Limitations:** Assumes linear relationship between features and target variable, may not capture non-linear patterns effectively without feature engineering.

Multilayer Perceptron (MLP):

- **Strengths:** Can capture complex non-linear relationships, making it suitable for tasks with intricate patterns.
- **Limitations:** Requires careful tuning of hyperparameters, susceptible to overfitting, and computationally expensive, especially with large datasets.

Conclusion

- Logistic Regression demonstrates superior performance in terms of accuracy and TPR, making it a strong candidate for sentiment analysis tasks.
- Naive Bayes, while simple and efficient, may suffer from lower TPR and higher FPR compared to other algorithms.
- MLP shows competitive performance but requires careful hyperparameter tuning and may be computationally expensive.

- TF-IDF representation generally enhances classification performance by accounting for word importance, particularly beneficial for algorithms like Naive Bayes.

Recommendations

- Based on the analysis, Logistic Regression with TF-IDF representation emerges as the top-performing algorithm for sentiment analysis tasks.
- Further experimentation with hyperparameter tuning and feature engineering may improve the performance of MLP.
- Consideration of additional algorithms or ensemble methods could be explored to further enhance classification performance.