Machine Learning based classification for Sentimental analysis of IMDb reviews

Project Team:

Uday, Tejas S

1. Problem Statement

The objective is to perform

document-level sentiment analysis on a dataset of 50,000 Internet Movie Database (IMDb) reviews. The task is a binary classification problem: to classify each movie review as either positive (label 1) or negative (label 0).

2. Methodology and Implementation Overview

A. Data Preprocessing and Cleaning: Raw text reviews were cleaned by converting to lowercase, removing HTML tags, punctuation, numbers, and common **stop words** (like 'a', 'the').

Lemmatization was applied to normalize words to their true root form (e.g., 'played' to 'play') to reduce vocabulary noise. The cleaned data was saved as *train_clean.csv and test_clean.csv.*

B. Feature Engineering (Vectorization): The cleaned text was converted into numerical feature matrices using two vectorization methods:

- **Binary Count Vectorization** (for Naive Bayes, CountVectorizer with binary=True).
- **TF-IDF Vectorization** (Term Frequency-Inverse Document Frequency).

N-grams (sequences of up to three consecutive words) were incorporated to capture word context.

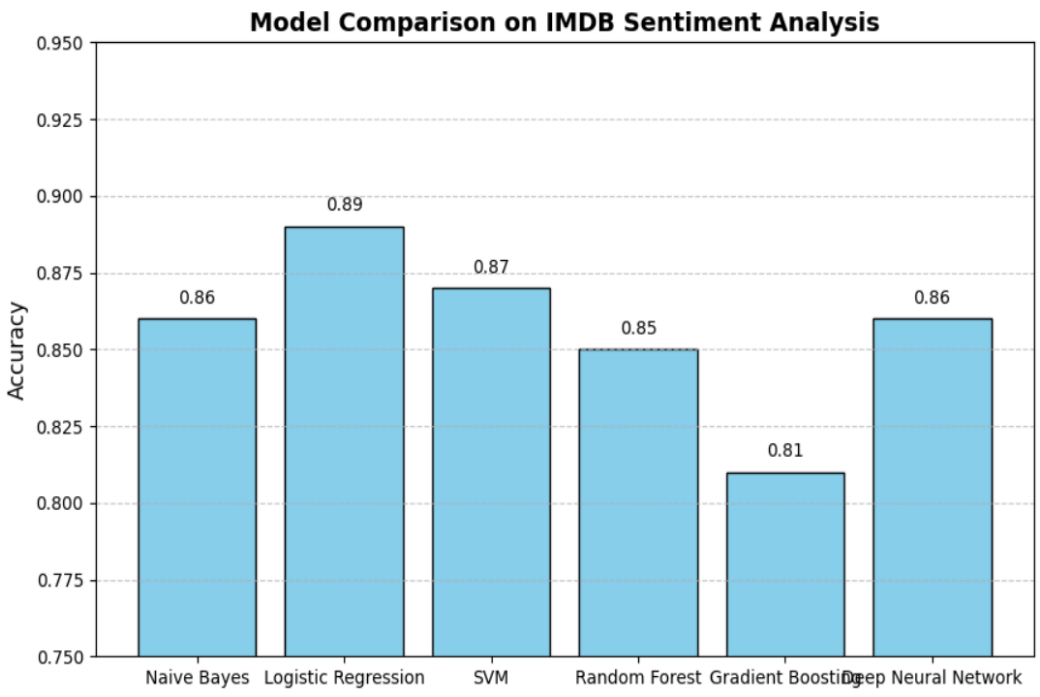
C. Classification Models: Six distinct machine learning models were implemented and evaluated using

scikit-learn pipelines, all trained on the processed feature matrices:

1. **Multinomial Naive Bayes (NB)**
2. **Logistic Regression (LR)**
3. **Support Vector Machine (SVM)** (using LinearSVC)
4. **Random Forest Classifier**
5. **Gradient Boosting Classifier**
6. **Deep Neural Network (DNN)** (using Keras/TensorFlow, with a 3-layer architecture trained on dense TF-IDF features).

3. Key Results and Conclusion

All models were evaluated based on their **accuracy** on the 25,000-review test dataset. The results are summarized below:



Model	Best Accuracy	Key Vectorization
Logistic Regression	**0.89**	TF-IDF (unigram + bigram)
SVM	0.87	TF-IDF (unigram + bigram)
Deep Neural Network	0.86	TF-IDF (unigram + trigram)
Naive Bayes	0.86	Binary Count (unigram + trigram)
Random Forest	0.85	TF-IDF (unigram + trigram)
Gradient Boosting	0.81	TF-IDF (unigram + bigram)

Conclusion: The **Logistic Regression** model, when combined with TF-IDF features (unigrams and bigrams), achieved the highest predictive performance with an accuracy of **89%**. The simple Naive Bayes model also performed well (86%), showing that linear models are highly effective for this large-scale text classification task.

Challenge/Future Work: A notable challenge was tuning tree-based models (Random Forest, Boosting) on large, sparse text data, often leading to slower training. Future work could explore more advanced vectorization techniques like Word2Vec embeddings or sequence models like Recurrent Neural Networks (RNNs) to potentially improve results.