Sentiment Analysis of IMDb Reviews using Machine Learning Based Classification

Mam Amna Zafar

Introduction

This project leverages natural language processing (NLP) and machine learning techniques to perform sentiment analysis on IMDb reviews. Sentiment analysis is conducted at three levels: sentence, document, and aspect. Key preprocessing steps include removing stop words, normalizing text, and transforming reviews into word matrices, which serve as input features for classification. Multiple machine learning algorithms, such as logistic regression, SVM, Naive Bayes, random forest, boosting, and deep neural networks, are employed to train and test the data.
The model aims to identify the most effective algorithm for accurate sentiment classification.

- Data

Data Collection:

- The dataset is retrieved using the method described in [5, 6]. This dataset consists of 50,000 movie reviews taken from IMDb. Half of the data is used for training.
- Moreover, both the training and testing dataset have 50 percent of positive reviews and 50 percent of negative reviews.

Preprocessing Steps:

a. Data Cleaning:

- Remove punctuation, line breaks, numbers, and stop words (e.g., "a," "the," "of") to eliminate noise.

b. Vectorization Techniques

- Binary Vectorization

  Represent text as a binary matrix where each element indicates the presence (1) or absence (0) of a vocabulary.

- n-Grams Vectorization
- Word Count Vectorization

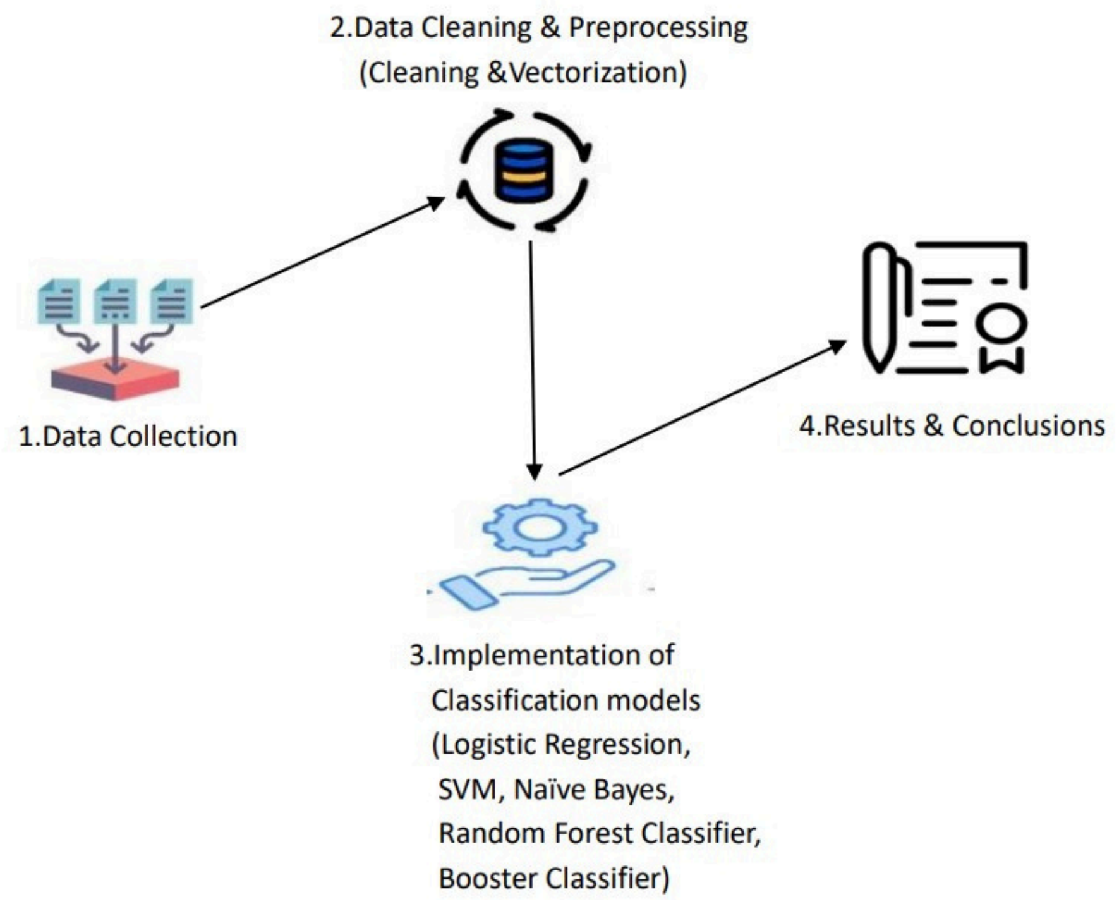
Dataset Split:

- Training Set (70%): Used for model training.

  Validation Set (15%): Optimizes model & prevents overfitting.

- Test Set (15%): Evaluates model performance on unseen data

Methodology



- Baseline Model: Analysis model was built using traditional machine learning algorithms
- Fine-Tuning: Advanced models, including random forest, boosting, and deep neural networks, were fine-tuned using labeled data and cross-entropy loss.
- Custom Feature Engineering: Vectorization techniques like binary, word-count, n-grams, and TF-IDF were used to transform text into feature matrices.

Results

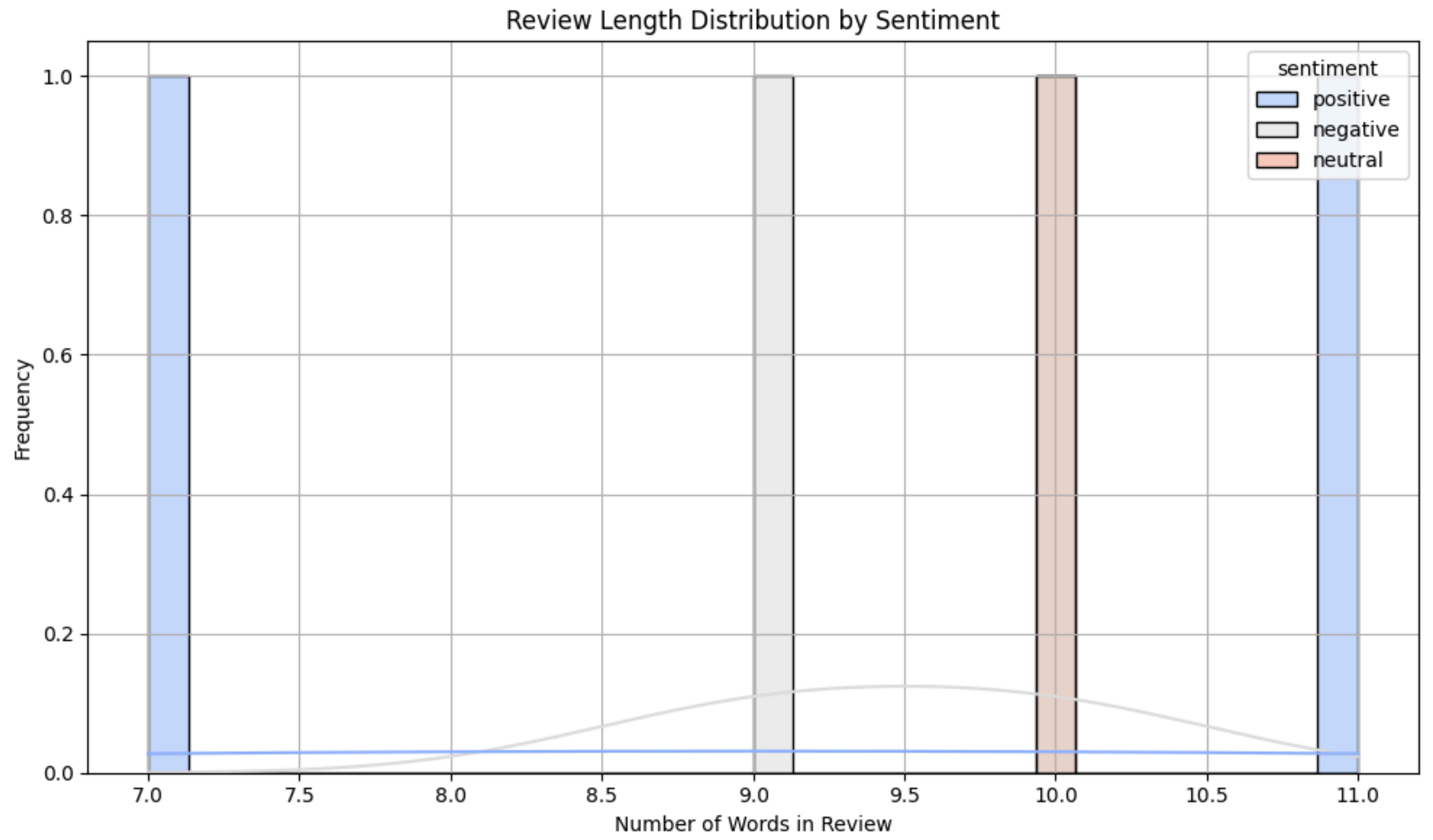
Confusion Matrix

	True Labels	
	Positive	Negative
Predicted Labels	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

Comparison of Different Algorithms

Algorithm	Vectorization	Regularization	Positive precision	Negative precision	Accuracy
Logistic	binary, 3 grams	1	0.908	0.893	0.900
	word count, 3 grams	1	0.899	0.894	0.897
	tf-idf, 3 grams	1	0.881	0.872	0.877
SVM	binary, 3 grams	100	0.908	0.894	0.901
	word count, 3 grams	20	0.900	0.895	0.898
	tf-idf, 3 grams	1	0.904	0.896	0.900
Naïve Bayes classifier	binary, 3 grams	-	0.839	0.923	0.881
	word count, 3 grams	-	0.836	0.912	0.874
	tf-idf, 3 grams	-	0.819	0.868	0.879
Random Forest classifier	binary, 3 grams	-	0.859	0.845	0.852
	word count, 3 grams	-	0.860	0.839	0.849
	tf-idf, 3 grams	-	0.864	0.786	0.844
Boosting classifier	binary, 3 grams	-	0.863	0.798	0.831
	word count, 3 grams	-	0.869	0.800	0.834
	tf-idf, 3 grams	-	0.865	0.789	0.825
Deep neural network	binary, 3 grams	0.0001	0.911	0.901	0.906
	word count, 3 grams	0.0001	0.896	0.900	0.898
	tf-idf, 3 grams	0.0001	0.881	0.921	0.901

Review’s Distrbution



Conclusion

This study analyzed IMDb reviews using machine learning models, achieving 90% accuracy in sentiment classification with methods like DNN, logistic regression, and SVM. Binary and 3-gram vectorization proved most effective across models, with Naïve Bayes and DNN excelling in negative sentiment prediction, while other models performed better.
Future Work: Explore additional vectorization techniques, such as subject removal from sentences, and implement more sophisticated models like recurrent neural networks (RNNs) to capture deeper relationships and improve performance further.

Related Work

Tripathy et al. proposed a text classification approach using Naïve Bayes (NB) and support vector machine (SVM), demonstrating that these algorithms achieve higher accuracy compared to existing methods.

Sharma et al. explored sentiment classification of short sentences using convolutional neural networks (CNN) with Word2Vec vectorization. Their methodology involved cleaning data with Word2Vec and applying CNN to address noise inconsistencies in text. Results showed that CNN effectively extracted features for categorizing short sentences.

Junaid Zia(2022-CS-02) Abdullah Fasih(2022-CS-06) Kumail Haider(2022-CS-23)