Project Report: Sentiment Classification

1. Introduction

This project, titled Sentiment Classification, focuses on analyzing and classifying movie reviews into positive or negative sentiment labels using Natural Language Processing (NLP) and Machine Learning techniques. The objective is to build a reliable classification model capable of generalizing well on unseen data.

2. Highlights

- Preprocessed 50,000 movie reviews using:
  - Regex-based HTML tag removal
  - Lowercasing and punctuation filtering
  - NLTK-based stopword removal and tokenization

- Converted cleaned text into sparse TF-IDF (Term Frequency–Inverse Document Frequency) features, capturing term importance while reducing noise from frequently occurring words.

- Trained a Logistic Regression classifier using scikit-learn on the TF-IDF features.
- Evaluated on an unseen test split and achieved an accuracy of 89.57% using standard scikit-learn evaluation metrics (accuracy, precision, recall, F1-score).

3. Dataset

Dataset: review_data.csv
Contains >50,000 labeled movie reviews with binary sentiment (positive, negative).

4. Preprocessing Workflow

1. HTML Cleaning: Removed any embedded HTML tags using regular expressions.
2. Text Normalization: Converted text to lowercase and removed non-alphabetic characters.
3. Tokenization: Tokenized using NLTK's word_tokenize.
4. Stopword Removal: Filtered out English stopwords using NLTK.

Processed reviews were stored in a new column called cleaned_review.

5. Feature Extraction

Applied TF-IDF Vectorization using TfidfVectorizer from scikit-learn.
Transformed cleaned text into high-dimensional sparse feature vectors.
This helped quantify term relevance across documents while reducing the influence of frequently used but less meaningful words.

6. Model Development

Algorithm Used: Logistic Regression
Data Split: 80% for training, 20% for testing using train_test_split.
Training: Fitted the model on the TF-IDF features of training data.
Evaluation:
- Accuracy achieved: 89.57% on unseen test data.
- Evaluation metrics used: Accuracy Score, Classification Report (Precision, Recall, F1).

7. Conclusion

This project successfully builds a baseline sentiment classifier that achieves strong performance using classic machine learning techniques. It shows that with well-designed preprocessing and feature engineering, even simple models like Logistic Regression can perform competitively on text classification tasks.