Movie Review Sentiment Analysis using Custom K-Nearest Neighbors (KNN)

Sphoorthy Pallempati

1. Introduction

This project performs sentiment analysis on movie reviews using a custom-built K-Nearest Neighbors (KNN) classifier to classify reviews as positive or negative. The work focuses on manual algorithm implementation, NLP preprocessing, feature engineering, and performance optimization for large-scale text data.

2. Objectives

- Convert raw movie review text into numerical features.
- Implement a custom KNN classifier without predefined KNN libraries.
- Evaluate cosine similarity and Euclidean distance metrics.
- Improve efficiency using dimensionality reduction.
- Analyze performance using cross-validation.

3. Dataset Description

The dataset contains 25,000 labeled movie reviews classified as positive (+1) or negative (-1). The data is split into training and testing sets. Dataset files are excluded due to size and academic restrictions.

4. Data Preprocessing

Text Cleaning

- Lowercasing, punctuation and special character removal.
- Placeholder replacement for usernames and hashtags.

Tokenization and Normalization

- Tokenization and stopword removal using NLTK.
- Stemming with Porter Stemmer.

HTML Removal

- Extracted plain text using BeautifulSoup.

5. Feature Extraction

TF-IDF vectorization was applied with n-grams (1–3), minimum document frequency of 5, and maximum document frequency of 0.5 to convert text into numerical vectors.

6. Dimensionality Reduction

Truncated Singular Value Decomposition (SVD) reduced TF-IDF features to 50 components, improving runtime and reducing overfitting.

7. Custom KNN Classifier

A custom KNN classifier was implemented supporting cosine similarity and Euclidean distance. Predictions are based on majority voting among the k nearest neighbors.

8. Model Evaluation

Stratified K-Fold Cross-Validation was used to evaluate multiple k values and compare distance metrics.

k	**Euclidean**	**Cosine**
3	0.76	0.78
5	0.78	0.79
7	0.79	0.80
10	0.80	0.81
15	0.81	0.82
20	0.81	0.82

Best accuracy achieved was 82% using cosine similarity.

9. Performance Optimization

Runtime was improved using TF-IDF vectorization, SVD dimensionality reduction, and optimized NLP preprocessing. Preprocessing took 3–4 minutes and KNN training with cross-validation took 7–10 minutes.

10. Output Generation

Final predictions were generated on the test dataset using optimal parameters and saved to an output file.

11. Technologies Used

Python, NumPy, scikit-learn, NLTK, BeautifulSoup, TF-IDF, SVD, Custom KNN, Cross-Validation.

12. Conclusion

This project demonstrates a complete sentiment analysis pipeline using a custom KNN classifier, highlighting practical skills in NLP, machine learning, and algorithm optimization.