Movie Review Classifier: Sentiment Analysis using NLP and Random Forest

Executive Summary:

The "Movie Review Classifier" project is a comprehensive exploration into sentiment analysis for movie reviews, employing Natural Language Processing (NLP) techniques and the Random Forest algorithm. The objective is to automatically classify movie reviews as either positive or negative, providing valuable insights into user sentiments.

Introduction:

Movie reviews are a rich source of user opinions that can greatly influence audience choices. Sentiment analysis on such reviews can assist in understanding public perception and contribute to decision-making processes. This project aims to build a robust sentiment classifier using a combination of NLP methods and machine learning.

Data Source:

The dataset used for this project is sourced from *Kaggle* and is provided by *Lakshmipathi N*. The IMDB dataset consists of 50,000 movie reviews, each labeled with sentiments (positive or negative). The dataset structure comprises two columns: "review" and "sentiment."

Dataset Information:

- **review:** User statements after watching the movie.
- **sentiment:** Indicates whether the given review is positive or negative.

Dataset Link: *IMDB Dataset on Kaggle*

Methodology:

1. Data Preprocessing:

The raw textual data is preprocessed using the Count Vectorizer, a popular technique in NLP. This step involves converting the text data into a numerical format by representing the frequency of words in the reviews. Pandas and NumPy libraries are employed for efficient data manipulation and handling.

2. Random Forest Classification:

The scikit-learn library is utilized to implement the Random Forest classifier, chosen for its ability to handle complex datasets and provide accurate results. The classifier is configured with 50 estimators and uses the entropy criterion for decision tree construction. This configuration aims to maximize information gain during the classification process.

3. Model Training and Evaluation:

The model is trained on a labeled training set, and its performance is evaluated on a separate testing set. Metrics such as accuracy, precision, recall, and F1 score are employed to assess the effectiveness of the sentiment classifier.

Results and Findings:

The Movie Review Classifier demonstrates promising results in accurately categorizing movie reviews based on sentiment.

Positive Sentiments:

- Precision: Over 60%
- Recall: Over 50%
- F1-Score: Over 55%

Negative Sentiments:

- Precision: Over 100%
- Recall: Over 100%
- F1-Score: Over 100%

This Performance level is considered acceptable, indicating the model effectiveness in correctly identifying both positive and negative sentiments within the movie reviews.

Conclusion:

The "Movie Review Classifier" project successfully achieves its goal of sentiment analysis on movie reviews. The integration of NLP techniques, the use of the Random Forest classifier, and the quality of the IMDB dataset contribute to the overall success of the project.

Recommendations and Future Work:

- Explore additional NLP techniques for text preprocessing.
- Experiment with different machine learning algorithms to compare performance.
- Enhance the model with more extensive labeled datasets for improved generalization.

Acknowledgments and Credits:

This report is submitted by **Rounik Mondal**, a B.Tech Computer Science and Engineering Student at Lovely Professional University, under the guidance and support of **Ayan Kumar Ghosh**. This Project was completed as a part of the coursework for the **"**Artificial Intelligence**"** course offered by InternsElite.

For a detailed walkthrough and implementation of the Movie Review Classifier, please refer to the associated Google Colab notebook: https://colab.research.google.com/drive/1sd7kohiifMrIY6Yk9128Z4asRJi5QaFV

References:

- https://pandas.pydata.org/docs/
- https://numpy.org/doc/stable/
- https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
- https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html