Course 7

IMDb Movie Review Sentiment Analysis

NLP Project Part A

Satyajit Kumar
DATA SCIENCE

Introduction

Sentiment analysis is a natural language processing (NLP) task that involves determining whether a given text expresses a positive or negative sentiment. In this project, I have analyzed movie reviews from the IMDb dataset and predict the sentiment (positive or negative) based on the text of the reviews. By leveraging various text preprocessing techniques, feature extraction methods, and classification algorithms, In this project I have developed a machine learning model capable of accurately predicting the sentiment of movie reviews. The insights derived from this analysis will be useful for movie producers, critics, and platforms like IMDb to understand public opinion and tailor marketing or content strategies accordingly.

Problem Statement

The primary objective of this project is to build a machine learning classification model that can predict the sentiment of IMDb movie reviews. The dataset contains a collection of movie reviews, and each review is labeled as either positive or negative.
Using text preprocessing, feature extraction techniques (such as TF-IDF), and various classification algorithms, the project will aim to develop a model that can effectively classify the sentiment of movie reviews. The model's performance will be evaluated using standard classification metrics, such as accuracy, precision, recall, and F1-score.

Data Exploration and Preprocessing

The dataset, named imdb_data.csv, comprises a collection of 50,000 movie reviews, each systematically labeled with its corresponding sentiment polarity.

- **Dataset Dimensions:** The dataset contains 50000 rows and two columns namely review (Textual Content) and sentiment (Categorical Value)
- **Missing Values:** There were no missing values found.
- **Sentiment Distribution:** The distribution of sentiment classes shows that there is equal distribution of positive and negative sentiments. Which means it is a balanced data.
- **Review Length Analysis:** The mean review length was approximately 1309 characters, with values ranging from a minimum of 7 to a maximum of 13704 characters.

Text Preprocessing

In this phase, I conducted the following preprocessing techniques.

1. **Lowercasing:** All textual content was converted to lowercase to ensure uniformity and prevent variations in capitalization from being treated as distinct linguistic entities.
2. **HTML Tag Removal:** Removed HTML tags using regular expressions.
3. **Punctuation Removal:** Using re.sub() function from regular expression module to remove noise

4. **Special Character and Number Removal:** Used re.sub() function to remove non-alphabets.
5. **Tokenization:** Reviews were tokenized
6. **Stop Word Removal:** Common English stop words (e.g., "the", "a", "is") were removed.
7. **Lemmatization:** Using WordNetLematizer class, lemmatized all the tokenized words to its dictionary root form.

Feature Engineering

converted cleaned text into numerical features and add other textual features. I also combined intrinsic textual features with TF-IDF vectorization.

Textual Features

- **Word Count:** The number of words present within each processed review.
- **Character Count:** The aggregate number of characters, excluding whitespace, contained within each processed review.
- **Average Word Length:** The arithmetic mean of word lengths within each processed review.

TF-IDF Vectorization

TF-IDF (Term Frequency-Inverse Document Frequency) It assigns a weighted significance to each term, reflecting its prevalence within a specific document relative to its rarity across the entire corpus.

- A TfidfVectorizer instance was initialized with specific parameters: max_features=5000 (constraining the feature set to the top 5000 most frequent terms), min_df=5 (excluding terms appearing in fewer than 5 documents), and max_df=0.8 (disregarding terms present in over 80% of documents, as these are often ubiquitous and less discriminative).
- The vectorizer was subsequently fitted and applied to the cleaned_review column, yielding a sparse TF-IDF matrix with dimensions of (50000, 5000).
- Ultimately, the TF-IDF features were conjoined horizontally with the word_count, char_count, and avg_word_length features. This concatenation resulted in a comprehensive feature matrix, X_features, with overall dimensions of (50000, 5003).
- The sentiment variable was numerically encoded, with positive reviews assigned a value of 1 and negative reviews assigned a value of 0, serving as the designated target variable, y.

Model Development and Evaluation

Data Splitting

Dataset was split into training and test set

- **Training Set (80%):** Contains 40,000 samples.

- **Testing Set (20%):** Consisting of 10,000 samples.

Model Training and Evaluation:

Four machine learning models were trained: Logistic Regression, Multinomial Naive Bayes, Linear Support Vector Machine (SVM), and Random Forest. For each model, a standardized set of performance metrics was computed and reported:

- **Accuracy:** proportion of correctly classified instances relative to the total number of instances.
- **Precision:** Measures the proportion of instances predicted as positive that were indeed true positives.
- **Recall:** Assesses the proportion of actual positive instances that were correctly identified by the model.
- **F1-Score:** Represents the harmonic mean of precision and recall, serving as a balanced indicator of a model's performance, particularly relevant in classification tasks with imbalanced classes
- **Classification Report:** Breakdown of precision, recall, and F1-score for each individual class ('negative' and 'positive').
- **Confusion Matrix:** Visual representation of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Performance Comparison

A comparative summary of the performance metrics across all models is presented in the table below:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8860	0.8776	0.8992	0.8883
Multinomial Naive Bayes	0.8210	0.8246	0.8190	0.8218
Linear SVM	0.8823	0.8750	0.8942	0.8845
Random Forest	0.8491	0.8564	0.8416	0.8490

Observations:

- **Logistic Regression** consistently gave superior performance, achieving the highest F1-Score (0.8883) and Accuracy (0.8860). Its precision and recall values for both positive and negative classes were observed to be well-balanced.
- **Linear SVM** demonstrated a F1-Score of 0.8845 which is close to what Logistic Regression predicted
- **Random Forest** also performed well. It lagged behind the other models by a slight margin.

- **Multinomial Naive Bayes** registered the comparatively lowest performance among the evaluated models, with an F1-Score of 0.8218.

Conclusion

Among all the models tested, Logistic Regression performed the best, giving the most accurate and balanced results. Linear SVM also gave strong results.