Text Classification Task

Sentiment Analysis on IMDB Reviews

Submitted By:
Pankaj-kumar
Chander Parkash

1. Introduction:

In this report, we present an analysis of sentiment classification on a subset of the IMDb movie review dataset. The IMDb dataset consists of movie reviews labeled with sentiments, either positive or negative, reflecting the overall opinion expressed in the review. The original dataset comprises 50,000 movie reviews, but for the purpose of this analysis, we have utilized a subset of 5,000 reviews.

The objective of this analysis is to build and evaluate machine learning models capable of accurately classifying the sentiment of movie reviews. We employ two different approaches for feature extraction and two distinct classification algorithms to compare their performance in sentiment classification.

2. Feature Extraction

Feature engineering enhances machine learning model performance by transforming raw data into informative features. Two techniques were used for IMDb movie reviews dataset analysis.

1. **Count Vectorization (CV):**
   Count Vectorization is a scikit-learn library technique that converts text documents into a matrix representation, with 5000 features. This method captures word frequency, creating a feature matrix suitable for machine learning algorithms. The matrix's shape reflects the data's dimensions, providing insight into feature space structure.

2. **TF-IDF Vectorization:**
   Term Frequency-Inverse Document Frequency (TF-IDF) Vectorization was used to calculate the importance of each word in a document relative to the entire corpus. With 5000 features, it generated a feature matrix with weighted scores, providing valuable information about the transformed data and its distribution.

3. Classification Algorithms

In our analysis of sentiment classification on the IMDb movie review dataset, we have explored a diverse set of classification algorithms to ascertain their efficacy in sentiment prediction. These algorithms include:

1. **Logistic Regression:** Logistic Regression is a popular linear classification technique for binary classification tasks. It uses the logistic function to model the connection between the binary target variable and the feature variables. Even though it is straightforward,

Logistic Regression can yield useful outcomes, particularly in cases where there is a linear relationship between the target and the feature

2. **Support Vector Machine (SVM):** Strong supervised learning algorithms like SVM are employed in classification applications. It functions by locating the ideal hyperplane in the feature space that maximizes the margin between various classes. SVM works well with high-dimensional data and is especially helpful in situations when the data cannot be separated linearly.

3. **Random Forest:** Random Forest is a flexible ensemble learning technique that builds several decision trees during training and produces the class mode for classification tasks. To provide diversity and robustness in the ensemble, each tree in the forest is trained using random subsets of features and training data.

4. **Gradient Boosting Classifier:** The Gradient Boosting Classifier builds a strong predictive model by gradually combining several weak learners through the use of an ensemble learning technique. The total prediction ability is to be improved by this algorithm's iterative correction of prior misclassifications.

4. Preprocessing of IMDb Movie Reviews Dataset

In our analysis, we undertook a series of preprocessing steps to prepare the IMDb movie reviews dataset for sentiment classification. Below is an outline of the preprocessing methodology:

1. **Loading the Dataset:** Initially, we loaded the IMDb dataset, containing reviews and corresponding sentiment labels, using the Pandas library.

2. **Data Cleaning:**
    - Removing HTML Tags: We eliminated HTML tags present in the text using regular expressions.
    - Removing Non-Alphanumeric Characters: Non-alphanumeric characters were removed from the text while converting all characters to lowercase. This step helps in standardizing the text data.

3. **Tokenization and Stopword Removal:**
    - Tokenization: The text was tokenized, splitting it into individual words.

- Stopword Removal: We removed common stopwords from the text data to reduce noise and improve the efficiency of subsequent analysis. Stopwords are words that do not contribute significant meaning to the text.

4. **Lemmatization:**
   - Word Lemmatization: Lemmatization was performed on the tokenized words to reduce them to their base or dictionary form. This step helps in standardizing words and improving the quality of the text data.

5. **Subset Selection:**
   - Random Subset: To manage computational resources effectively, we randomly selected a subset of 5000 rows from the preprocessed dataset while ensuring the representation of the original dataset.

6. **Saving Preprocessed Data:**
   - Saving to CSV: The preprocessed dataset, containing the processed text and sentiment labels, was saved to a separate CSV file for future use in model training and evaluation.

5. Results

Count Vectorization

Models	**Imbalanced dataset**		**Balanced dataset**	
	CV=5	**Test & Train**	**CV=5**	**Test & Train**
Logistic regression	F1-Score: 0.8330 Accuracy: 0.8330	Test Set Accuracy: 0.8420 Train Set Accuracy: 0.9995	F1-Score: 0.8332 Accuracy: 0.8333	Test Set Accuracy: 0.8422 Train Set Accuracy: 0.9988
SVM	F1-Score: 0.8351 Accuracy: 0.8266	Test Set Accuracy: 0.8320 Train Set Accuracy: 0.9673	F1-Score: 0.8231 Accuracy: 0.8233	Test Set Accuracy: 0.8249 Train Set Accuracy: 0.9665
Random Forest	F1-Score: 0.8340 Accuracy: 0.8308	Test Set Accuracy: 0.8510 Train Set Accuracy: 1.0000	F1-Score: 0.8292 Accuracy: 0.8268	Test Set Accuracy: 0.8363 Train Set Accuracy: 1.0000
Gradient Boosting	F1-Score: 0.8198 Accuracy: 0.8088	Test Set Accuracy: 0.8374 Train Set Accuracy: 0.8665	F1-Score: 0.8048 Accuracy: 0.8052	Test Set Accuracy: 0.8204 Train Set Accuracy: 0.8695

TF-IDF

Models	Imbalanced dataset		Balanced dataset	
	CV=5	**Test/Train**	**CV=5**	**Test/Train**
Logistic regression	F1-Score: 0.8560 Accuracy: 0.8518	Test Set Accuracy: 0.8660 Train Set Accuracy: 0.9390	F1-Score: 0.8562 Accuracy: 0.8563	Test Set Accuracy: 0.8550 Train Set Accuracy: 0.9347
SVM	F1-Score: 0.8573 Accuracy: 0.8528	Test Set Accuracy: 0.8640 Train Set Accuracy: 0.9965	F1-Score: 0.8582 Accuracy: 0.8583	Test Set Accuracy: 0.8519 Train Set Accuracy: 0.9963
Random Forest	F1-Score: 0.8308 Accuracy: 0.8276	Test Set Accuracy: 0.8480 Train Set Accuracy: 1.0000	F1-Score: 0.8240 Accuracy: 0.8280	Test Set Accuracy: 0.8343 Train Set Accuracy: 1.0000
Gradient Boosting	F1-Score: 0.8161 Accuracy: 0.8036	Test Set Accuracy: 0.8265 Train Set Accuracy: 0.8958	F1-Score: 0.8010 Accuracy: 0.8010	Test Set Accuracy: 0.8106 Train Set Accuracy: 0.8939

A. After applying CV , TF-IDF and Normal train/test approach:

1. **Balanced Dataset:**

   SVM achieved the highest F1-Score of 0.8582 with CV, while LR achieved the highest F1-Score of 0.8550 on the test set.

2. **Imbalanced Dataset:**

   SVM achieved the highest F1-Score of 0.8573 with CV, while LR achieved the highest F1-Score of 0.8660 on the test set.

B. Which one is better Count Vector/ TF-IDF:

- After comparing the results, TF-IDF appears to perform slightly better than Count Vectorization, especially on the balanced dataset.
- TF-IDF considers the importance of words by weighing them based on their frequency in the document and across the corpus, which might contribute to its better performance.

C. Which model performs better:

- The model performance varies depending on the dataset imbalance and the feature extraction method used.
- For the imbalanced dataset, SVM and LR with TF-IDF achieved the highest F1-Scores.
- For the balanced dataset, SVM with TF-IDF performed the best.

D. Best model classification report and confusion matrix:

Test Set Classification Report:

```
                precision    recall  f1-score   support

    negative       0.87      0.82      0.84       491
    positive       0.84      0.88      0.86       517

    accuracy                           0.85      1008
   macro avg       0.85      0.85      0.85      1008
weighted avg       0.85      0.85      0.85      1008


Test Set Confusion Matrix:
[[402  89]
 [ 60 457]]


Train Set Classification Report:
              precision    recall  f1-score   support

    negative       1.00      1.00      1.00      2028
    positive       1.00      1.00      1.00      2002

    accuracy                           1.00      4030
   macro avg       1.00      1.00      1.00      4030
weighted avg       1.00      1.00      1.00      4030



Train Set Confusion Matrix:
[[2019    9]
 [   6 1996]]
```

Dataset and Code

1. CV & TF-IDF IMbalanced & Balanced.ipynb
2. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/data