Classification Report on Sentiment Analysis of IMDB Movie Reviews

Aaran Poon 36228203
Department of Electrical and Computer Engineering
University of British Columbia
aaranp1919@gmail.com

Abstract

This report presents a comprehensive analysis of sentiment classification on the IMDB movie review dataset using machine learning techniques. The goal is to develop a robust model capable of accurately predicting the sentiment (positive or negative) associated with movie reviews based on the textual content. The report covers the problem definition, dataset description, methodology, experiments, results analysis, and a discussion on potential future improvements.

1 Problem Definition

The primary objective of this project is to build a reliable binary classification model that can effectively distinguish between positive and negative sentiments expressed in movie reviews.

2 Dataset

The dataset used in this study is the IMDB Dataset from Kaggle and the link is `https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`, which contains a collection of movie reviews along with their corresponding sentiment labels (positive or negative). The dataset is preprocessed and cleaned to remove irrelevant information, such as HTML tags, URLs, and special characters. Additionally, techniques like tokenization, stop word removal, and lemmatization are applied to enhance the feature representation.

2.1 Data Preprocessing

The raw text data is cleaned and preprocessed using the following steps:

1. Convert all text to lowercase.
2. Remove hashtags and Twitter handles.
3. Remove HTML tags, including
 and its variations.
4. Remove line breaks and replace with spaces.
5. Remove special characters and punctuation.
6. Remove URLs.
7. Tokenize the text into words.
8. Remove stop words (excluding the word 'over').
9. Lemmatize the remaining tokens.
10. Join the processed tokens back into a single string.

Doing so helps with optimizing the data and help with comprehending human words.

2.2 Feature Engineering

In addition to the preprocessed text data, one-hot encoding for words is used. This helps to capture the presence of positive and negative words in the review text. Common words that have positive and negative connotation in movie reviews are predefined like so

- Postive words: excellent, amazing, outstanding, fantastic, brilliant, good, superb, terrific, masterpiece, incredible, phenomenal, great
- Negative words: terrible, awful, horrible, disappointing, boring, mediocre, trash, disgusting, unbearable, dreadful, bad

A custom feature called 'One Hot Encoded' is created. This feature is a binary indicator that captures the presence of positive or negative words in the review text. The lists of positive and negative words are predefined, and the feature is set to 1 if a positive word is present and no negative words are present, and 0 otherwise.

	review	sentiment	One Hot Encoded
0	one reviewers mentioned watching 1 oz episode youll hooked right exactly hap...	1	0
1	wonderful little production filming technique unassuming oldtimebbc fashion ...	1	1
2	thought wonderful way spend time hot summer weekend sitting air conditioned ...	1	1
3	basically theres family little boy jake thinks theres zombie closet parents ...	0	0
4	petter matteis love time money visually stunning film watch mr mattei offers...	1	0
...	...	...	...
49995	thought movie right good job wasnt creative original first expecting whole l...	1	0
49996	bad plot bad dialogue bad acting idiotic directing annoying porn groove soun...	0	0
49997	catholic taught parochial elementary schools nuns taught jesuit priests high...	0	0
49998	im going disagree previous comment side maltin one second rate excessively v...	0	0
49999	one expects star trek movies high art fans expect movie good best episodes u...	0	0

50000 rows × 3 columns

Figure 1: One hot encoded data with sentiment

Figure 1 shows how the data looks after the one hot encoding feature was added based on the positive and negative words.

3 Method and Steps

The methodology employed in this project involves the following steps:

1. Data Preprocessing: As described in Section 2.1, the raw text data is cleaned and preprocessed using techniques like tokenization, stop word removal, and lemmatization.
2. Feature Extraction: The preprocessed text data is transformed into a numerical representation using the TF-IDF vectorizer, allowing for efficient feature extraction.
3. Feature Engineering: The 'One Hot Encoded' feature is created based on the presence of positive and negative words in the review text, as described in Section 2.2.
4. Data Splitting: The dataset is split into training and test sets using an 80:20 ratio.
5. Model Training: Two different machine learning models, Logistic Regression and Random Forest Classifier, are trained on the preprocessed data using stratified k-fold cross-validation (k=5).
6. Hyperparameter Tuning: Grid search is employed to find the optimal hyperparameters for both models, maximizing their performance on the validation set. For Logistic Regression, the regularization parameter (C) is tuned, while for the Random Forest Classifier, the number of estimators, maximum depth, and maximum leaf nodes are tuned.
7. Model Evaluation: The trained models are evaluated on a held-out test set, and various performance metrics, such as accuracy, precision, recall, and F1-score, are calculated.

8. Result Analysis: The results obtained from the trained models are analyzed, and their strengths and weaknesses are discussed.

4 Experiment, Techniques, and Results Analysis

The experiments involved varying the hyperparameters, such as the regularization parameter (C) for Logistic Regression and the number of estimators, maximum depth, and maximum leaf nodes for the Random Forest Classifier.

4.1 Logistic Regression

The Logistic Regression model was trained using different values of the regularization parameter (C). The range of the hyperparameter (C) was varied linearly spaced from 3.5 to 5. Grid search was employed to find the optimal combination of these hyperparameters. Figure 2 shows the plots of accuracy, precision, recall, and F1-score against the C values.

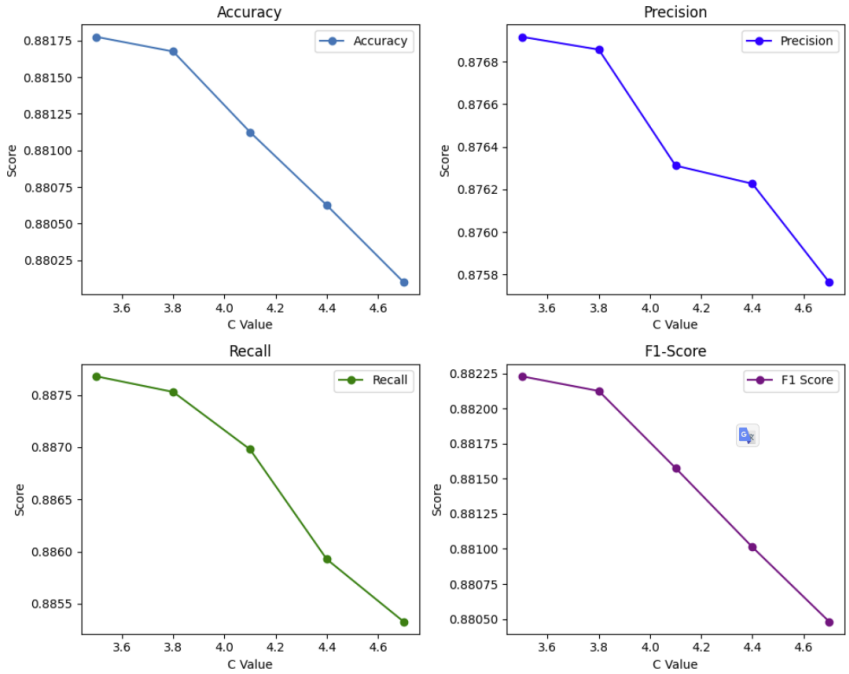

Figure 2: Evaluation metrics for Logistic Regression model with varying C values.

From the plots, it can be observed that the model's performance varies with different C values, and there is a trade-off between the different metrics. The optimal value of C needs to be chosen based on the specific requirements of the problem. It can be concluded that a C value of 3.5 is the optimal value since it produces the highest value across the metrics relative to other parameter values.

For C = 3.5 We get an accuracy: 0.8816, precision: 0.8745, recall: 0.8932, and F1-score: 0.8838 for our tuned logistic regression model.

4.2 Random Forest Classifier

The Random Forest Classifier model was trained using different combinations of hyperparameters, including the number of estimators, maximum depth, and maximum leaf nodes.

We see that despite using the optimal parameters for the Random Forest Classifier model, it still has a lower score across the different metrics in comparison to the tuned Logistic Regression model.

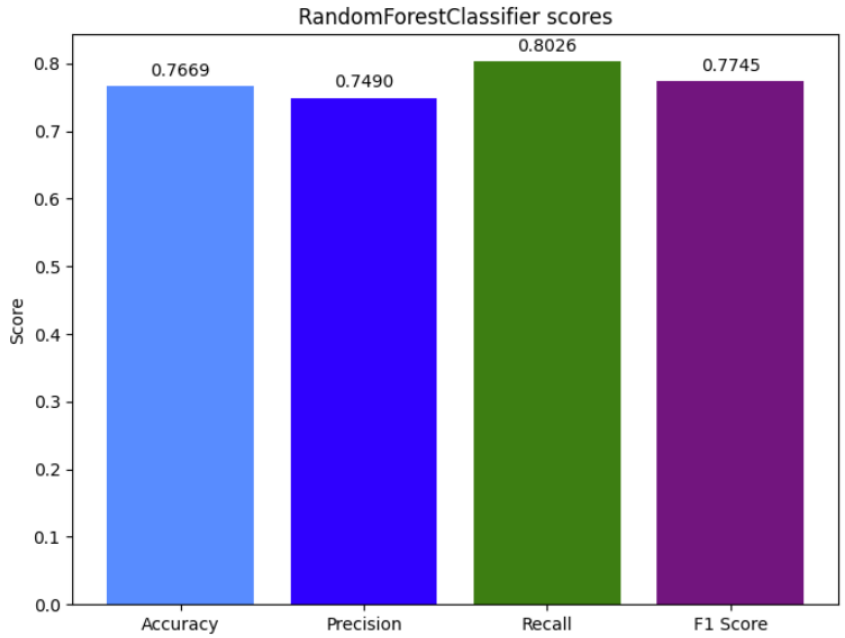

Figure 3: Evaluation metrics for the Random Forest Classifier model.

4.3 Comparison of Models

Figure 4 presents a comparison of the evaluation metrics for the Logistic Regression and Random Forest Classifier models on the test set.

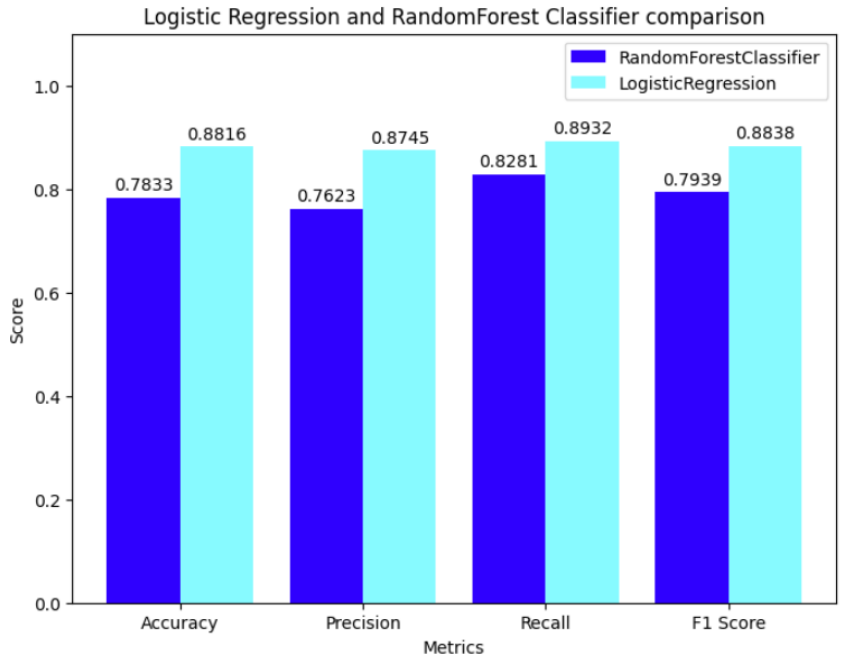

Figure 4: Comparison of evaluation metrics for Logistic Regression and Random Forest Classifier models.

The results demonstrate that while both models are comparable in sentiment classification, the tuned Logistic Regression model outperforms the Random Forest Classifier across all evaluation metrics.

The regression model exhibited better generalization capabilities and was able to capture the complex patterns in the textual data more effectively.

5 Discussion

In this project,machine learning models were developed for sentiment classification on the IMDB movie review dataset. The Logistic Regression and Random Forest Classifier models were trained and evaluated using various techniques, including data preprocessing, feature extraction, feature engineering, and hyperparameter tuning.

While the achieved performance is promising, there is still room for improvement. One potential avenue for further research is to explore more advanced natural language processing (NLP) techniques, such as word embeddings (e.g., Word2Vec or GloVe) or transformer-based models (e.g., BERT or GPT). These techniques can capture more contextual and semantic information from the text, potentially leading to better feature representations and improved classification performance.

Additionally, ensemble methods that combine multiple models could be investigated. By leveraging the strengths of different models, ensemble techniques may yield more robust and accurate predictions.

Another area for future work is to expand the scope of the sentiment analysis task to include multiclass classification or sentiment intensity prediction for movie recommendation. Instead of binary classification (positive or negative), models could be developed to categorize reviews into multiple sentiment classes (e.g., very positive, positive, neutral, negative, very negative) or predict a continuous sentiment score representing the intensity of the sentiment expressed.

References

[1] Movie review data set: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[2] CPEN355 class notes.