Sentiment Classification of Reviews: A Comparative Study of Machine Learning Techniques with Cross-Validation and Error Analysis

Group 24 : Eunice Lim Jing Ru, Kathleen Pang Qi Yu, Seah Jun Hui, Shervon Teo Jie Ling

Introduction

Our project applies sentiment analysis to classify movie reviews as positive or negative, a key task in natural language processing. With digital content rapidly expanding, businesses increasingly rely on automated sentiment analysis to understand public opinion, track brand reputation, and make data-driven decisions to improve customer experience.

In this study, we will compare several machine learning techniques, including Logistic Regression, Multinomial Naive Bayes, Convolutional Neural Networks, and Support Vector Machines. This study aims to reveal the most effective approach for sentiment classification and suggest areas for improvement.

Investigating Recent Works

Several recent works have examined sentiment analysis methods for movie review classification, highlighting both their effectiveness and limitations. (Birjali et al. 2021) noted that traditional methods like Logistic Regression and Naive Bayes are still widely used due to their simplicity, but they face challenges with complex language constructs like sarcasm and negation. This suggests that while these models are computationally efficient, they may not capture deeper semantic relationships in text. (Dang et al. 2020) compared Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for sentiment classification, concluding that CNNs excel in detecting local features but struggle with long-term dependencies, an area where RNNs perform better, though at the expense of higher computational costs. This indicates that selecting the right model involves trade-offs between feature extraction and computational resources. Lastly, (Teja et al. 2018) proposed a hybrid method that integrates machine learning with lexicon-based techniques, achieving a balance between interpretability and accuracy, yet it underperforms compared to deep learning models due to limitations in predefined lexicons. These insights will guide our approach in evaluating different AI/ML techniques, helping us understand the trade-offs between accuracy, complexity, and interpretability.

Dataset

The dataset used in this project include: 25000 positive movie reviews, stored in data/pos.zip, and 25000 negative movie reviews, stored in data/neg.zip. Positive and negative reviews were extracted from zip files into separate directories, and then loaded into lists. These reviews were combined into a single list, X, along with a corresponding labels list, y, indicating sentiment.

Pre-processing the Data

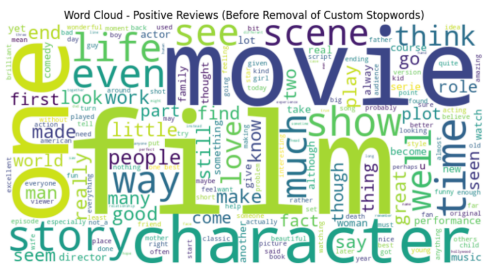
Pre-processing steps are crucial to improve dataset quality and consistency, enabling the sentiment classification model to better capture and interpret nuanced sentiments in reviews. We removed HTML tags, expanded contractions, removed special characters, converted all text to lowercase, performed tokenisation, handled negations, removed stopwords, performed lemmatisation, removed rare words, and removed any missing data.

Data Visualisation

We used a Count Plot to visualize the distribution of positive and negative classes, helping us detect any class imbalance. Identifying class imbalance is crucial, as it can bias the model towards predicting the more frequent class, limiting its accuracy across both positive and negative reviews.

For the movie dataset, the class distribution is balanced. However, if imbalance were present, methods like re-sampling or weighted classes could be used to ensure equal treatment of both classes, thus preventing biased predictions.

Next, we generated a word cloud that provided a quick insight into the most distinctive words across the dataset.



This visualisation highlighted certain common terms like “movie”, “film”, and “character”. While these words appeared frequently, they did not contribute meaningful sentiment-specific information. More refinement is needed to gain deeper insights into the emotional tone of the reviews.

Additional Dataset

After establishing a solid understanding of the model's performance on movie reviews, we conducted separate testing using a dataset of 40,000 Spotify User reviews, sampled from a larger set of 51,473 reviews from Kaggle. By testing on this separate dataset, we aimed to assess the model's versatility and effectiveness across different types of reviews, providing insight into its potential for generalization to varied domains.

Methods

Custom Stopwords Removal

Gaining insights from data visualisation, we designated those common terms, such as "movie," "film," and "character," as custom stop words and removed them from the dataset. This step allowed us to focus on words with greater relevance to sentiment, such as "good," "great," "well," and "disappointing." By applying these custom stop words, we generated a second word cloud:

.

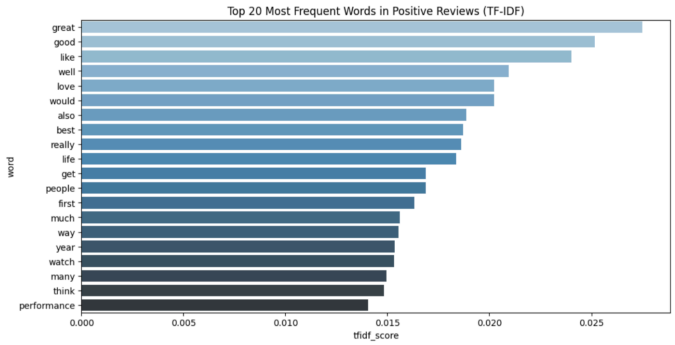
This updated visualisation displayed terms more meaningful to our analysis, improving the focus and relevance of our data.

Bag-of-Words Vectoriser

The Bag of Words (BoW) model is a text representation technique that transforms documents into numerical feature vectors based on the frequency of each word. It treats each unique word as a feature, disregarding word order and context. This results in a matrix where rows represent documents and columns represent individual words, with values indicating word counts.

TF-IDF (Term Frequency-Inverse Document Frequency)

Term Frequency-Inverse Document Frequency (TF-IDF) is a method for text representation that quantifies the importance of a word in a document relative to a collection of documents. It combines term frequency, which counts how often a word appears in a document, with inverse document frequency, which measures how common or rare a word is across the corpus.



Comparing the 2 feature extraction models, while Bag of Words can capture common patterns and raw word counts, it lacks the nuance to differentiate between general language and sentiment-specific terms. Therefore, for sentiment analysis, where the goal is to identify words that are both frequent and contextually significant, **TF-IDF is the preferred method.**

Machine Learning Models

Next, we build several machine learning models, evaluating their performance. A standardized approach was used for training and testing to ensure fairness: each model was trained on 80% of the data and tested on 20%. This method allows for direct comparison of performance metrics. Cross-validation further enhances model robustness and mitigates overfitting.

Logistic Regression

Logistic Regression is chosen as it is a statistical method commonly used for binary classification tasks in data analysis, aiming to predict the probability of a specific class or event based on one or more predictor variables derived from the dataset. Unlike linear regression, which predicts continuous outcomes, logistic regression employs the logistic function to transform predicted values into probabilities ranging from 0 to 1. This model outputs a probability score that can be thresholded (typically at 0.5) to determine binary outcomes, such as classifying sentiments as positive or negative in textual data.

Multinomial Naive Bayes (MNB)

The Multinomial Naive Bayes (MNB) classifier is chosen for its effectiveness in handling high-dimensional sparse data typical of text classification tasks. Grounded in Bayes' theorem, MNB operates under the assumption of conditional independence of features given the class label, which simplifies the computation of likelihoods and enhances efficiency. This model excels with categorical

data and is robust to class imbalances, providing interpretable probability estimates for class membership. Its computational efficiency and effectiveness make MNB a compelling choice for sentiment classification in this study.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are chosen for their ability to automatically learn spatial hierarchies of features, making them highly effective for text classification tasks. By applying convolutional layers with filters, CNNs capture local patterns in text data, allowing them to identify relevant features like n-grams. The combination of convolutional and pooling layers enhances dimensionality reduction and translation invariance. This architecture enables CNNs to learn both low-level and high-level abstractions simultaneously, making them a robust choice for sentiment analysis in this study.

Support Vector Machines (SVM)

Support Vector Machines (SVM) are chosen for their effectiveness in high-dimensional spaces and ability to classify linear and nonlinear data. By maximizing the margin between classes, SVMs construct optimal decision boundaries. They utilize kernel functions to transform data into higher dimensions, enabling the separation of complex patterns. SVMs are robust against overfitting, particularly in text classification tasks, making them well-suited for text-based sentiment analysis.

Results and Discussion

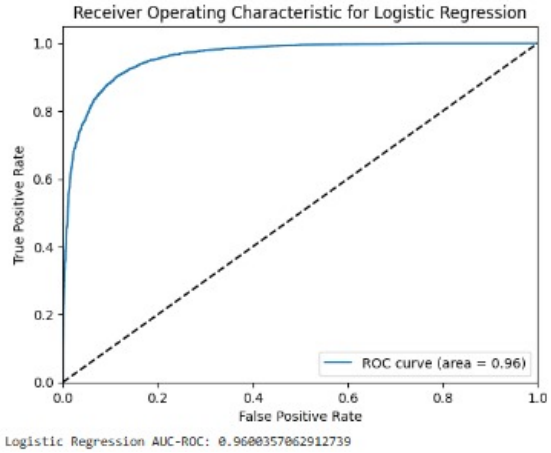
To evaluate our models, we utilised classfication_report from scikit-learn, which generated a comprehensive summary of key metrics, including Accuracy, Precision, Recall and F1 score for each sentiment class (positive and negative). Accuracy provides an overall measure of correct predictions, while precision and recall focus specifically on the relevance and completeness of predictions, respectively. The F1 score combines precision and recall to give a balanced view of model performance. To evaluate the model's ability to distinguish sentiments, we calculated the ROC AUC score, which measures its performance across different thresholds. A higher ROC AUC indicates better separation between positive and negative reviews, reflecting fewer misclassifications and stronger sentiment recognition.

Although the differences are slight, we observe that precision is consistently higher and recall lower for negative predictions compared to positive ones, suggesting a cautious approach by the model toward classifying reviews as negative. This subtle imbalance highlights potential for fine-tuning to boost recall for negative reviews while maintaining high precision, leading to more balanced performance across both sentiment classes.

Precision, Recall, and F1 scores are reported separately for negative (top) and positive (bottom) reviews.

M	Acc	Prec	Rec	F1	AUC
LR	0.892	0.90 0.88	0.88 0.91	0.89 0.90	0.960
NB	0.867	0.87 0.86	0.86 0.88	0.86 0.87	0.936
CNN	0.878	0.88 0.87	0.87 0.89	0.88 0.88	0.949
SVM	0.893	0.90 0.89	0.88 0.90	0.89 0.89	0.956

From our results, the SVM model showed the highest accuracy and the Logistic Regression model showed the highest ROC AUC. ROC AUC is a better metric than accuracy because it evaluates model performance across all classification thresholds and accounts for the trade-off between true positive and false positive rates, which is crucial in sentiment analysis where accurately capturing positive sentiments is essential. Therefore, we choose Logistic Regression for further exploration.



Logistic Regression AUC-ROC: 0.9600357062912739

Hyperparameter Tuning

In order to fine-tune the Logistic Regression model's performance, we used hyperparameter tuning with RandomizedSearchCV. This approach allowed us to explore a range of parameter values for regularization (C), penalty types (l1 and l2), and solver options (liblinear and saga). By randomly sampling 50 different parameter combinations, we aimed to identify an optimal

configuration over a 5-fold cross-validation, using accuracy as the scoring metric.

```
Best Parameters: {'C': 2.8103450968738075, 'penalty': 'l2', 'solver': 'liblinear'}
Best Cross-Validation Score: 0.89035
Optimized Model Accuracy: 0.8924
Classification Report:
              precision    recall  f1-score   support

    negative       0.90      0.88      0.89      4978
    positive       0.89      0.90      0.89      5022

    accuracy                           0.89     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.89      0.89      0.89     10000

Optimized Logistic Regression AUC-ROC: 0.9598533827614902
```

Despite this tuning, we observed minimal change in the model's performance metrics, including accuracy and ROC AUC. This limited impact is likely due to the nature of Logistic Regression, which has fewer parameters and is relatively robust to minor adjustments, especially when the initial settings are already close to optimal.

Error Analysis

To understand the limitations of our Logistic Regression model, we conducted an error analysis by examining specific misclassifications using a confusion matrix to identify false positives (FPs) and false negatives (FNs).

For instance, in one misclassified positive review, the reviewer expressed, "rather appalled to see the low rating movie received." The shock at the low rating indicates that the reviewer perceives the movie as good, as they believe it deserves a higher score. The model misclassified this review as negative by focusing on the opening expression, demonstrating its difficulty in interpreting nuanced sentiments.

Misclassifications in sentiment analysis often arise from the model's challenges in detecting **sarcasm** and **nuanced expressions**. For example, a sarcastic remark may be interpreted as positive, leading to incorrect predictions. To enhance model accuracy, we can improve its ability to recognize these subtleties by incorporating advanced natural language processing techniques, such as sentiment lexicons and context-aware embeddings, which can help capture the complexities of language more effectively.

Evaluation on New Dataset

When testing the model on new data, it achieved a low accuracy of 0.500 alongside a high ROC AUC of 0.876 indicates that while the model can effectively distinguish between positive and negative classes, it struggles with overall classification performance.

The model may have overfitted to the training data, capturing noise instead of the underlying sentiment patterns. As a result, it struggles to generalize to new data, leading to suboptimal classification performance. Future iterations of the model could benefit from additional, diverse training data to improve generalization and reduce overfitting.

Conclusion

In conclusion, our results affirm that Logistic Regression remains the best-performing model for this sentiment analysis task, with its high ROC AUC reflecting strong discrimination between positive and negative sentiments across thresholds. While we observed a decline in accuracy on the new dataset, likely due to dataset-specific nuances, this does not detract from the model's core effectiveness. Future work could consider incorporating transformer-based models such as BERT, RoBERTa, or DistilBERT, which excel at capturing complex linguistic features like sarcasm and nuanced sentiment shifts. Additionally, integrating a dedicated sarcasm model could address one of the primary challenges in sentiment analysis, further enhancing the adaptability and robustness of our approach across diverse, real-world contexts.

Comparison to Human Evaluation

The model lacks the nuanced understanding humans have, especially with context and sarcasm. However, it can complement human efforts by processing large volumes of data, offering insights that support decision-making rather than replacing human judgment.

Societal Impacts

Our model's implementation raises privacy concerns, as it processes substantial user data, demanding strict safeguards. Fairness and transparency are essential to avoid biases and ensure interpretability. Additionally, while automation improves efficiency, it could impact roles held by human analysts, requiring responsible integration into the workforce.

References

Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowledge-Based Systems, 226, 107134. https://doi.org/10.1016/j.knosys.2021.107134

Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment Analysis Based on Deep Learning: A Comparative Study. Electronics, 9(3), Article 3. https://doi.org/10.3390/electronics9030483

kim, A. (2024, October 3). Spotify user reviews. Kaggle. https://www.kaggle.com/datasets/alexandrakim2201/spotify-dataset/data

Teja, J. S., Sai, G., Kumar, D., & Manikandan, R. (2018). Sentiment Analysis of Movie Reviews Using Machine Learning Algorithms-A Survey. https://www.semanticscholar.org/paper/Sentiment-Analysis-of-Movie-Reviews-Using-Machine-Teja-Sai/5ba73bdb42324f213d67c979ca0b739f3a65fe25