Comparative Analysis of Logistic Regression, Support Vector Machines and DistilBERT for Sentiment Analysis Using TF-IDF Features

Rony Macwan
College of Information Science
University of Arizona
ronymacwan@arizona.edu

Abstract

This project presents a comparative analysis of three different methods for sentiment analysis: Logistic Regression, Support Vector Machines (SVM), and DistilBERT, by utilizing TF-IDF features extracted from text data. We implemented Logistic Regression with hyperparameter tuning via GridSearchCV to achieve a comprehensive evaluation of model performance through accuracy metrics and classification reports. Similarly, we employed SVM and preprocessing the text by removing punctuation and stop words, to generate TF-IDF features for training. Finally, we utilized the DistilBERT model from Hugging Face to capture contextual information in the text. The results from each method were analyzed and compared to get insights into their effectiveness in sentiment classification tasks. This project will compare the evaluation metrics of traditional machine learning approaches with the deep learning technique of DistilBERT.

1 Introduction

Sentiment analysis is an essential aspect of natural language processing (NLP) that aims to identify the emotional tone behind written text. With the rise of social media, online reviews, and user-generated content, the importance of understanding public sentiment has grown significantly. Businesses, policymakers, and organizations can benefit from insights into how their audiences feel, which is important for decision making. In this project, we explore different approaches to sentiment analysis, comparing traditional machine learning methods like Logistic Regression and Support Vector Machines (SVM), with the deep learning technique of DistilBERT. By examining these methods, we aim to highlight their effectiveness in analyzing sentiments. Logistic Regression, a foundational statistical method, is often used for binary classification tasks and provides interpretable results. SVM, on the other hand, is effective in high-dimensional spaces and excels at finding optimal decision boundaries. In contrast, DistilBERT uses deep learning for NLP. As a distilled version of the BERT model, it retains much of the original's performance while being more efficient in terms of computation and speed. DistilBERT utilizes contextual embeddings to capture nuanced meanings in text that traditional methods might overlook. By analyzing accuracy, precision, recall, and F1-score, we seek to provide insights into the strengths and weaknesses of each approach.

2 Methods

We began by cleaning the text data to enhance the relevance and focus of our analysis. This involved several key steps: stripping HTML tags using BeautifulSoup to extract visible content, removing URLs to prevent irrelevant information from influencing our results, and filtering out common stop words such as "the" and "is." The denoise_text function combined these processes. Finally, we applied this cleaning function to the 'review' column of our dataset and saved the cleaned data to a new CSV file..

For the Logistic Regression model, we began by preparing the cleaned text data, which served as our feature set (X) while the sentiment labels constituted our target variable (y). The dataset was split into training and testing sets using an 80-20 ratio. We utilized the TfidfVectorizer to convert the text data into TF-IDF features to capture the significance of words in relation to their frequency across documents. This representation emphasizes informative words while diminishing the weight of common terms. To optimize model performance, we implemented GridSearchCV, which allowed us to systematically explore various hyperparameter configurations by tuning the regularization parameter (C) to control overfitting, and selecting between the 'lbfgs' and 'liblinear' solvers. By fitting the logistic regression model with a maximum of 200 iterations to the training data, we aimed for convergence while maintaining computational efficiency. The best-performing model from the grid search was then used to predict sentiment on the test set.

The next model is a Support Vector Machine (SVM) classifier which was chosen for its effectiveness in handling high-dimensional data such as text. We began by loading a subset of the dataset and specifically took the first 10,000 cleaned reviews to ensure a diverse representation and also keeping the analysis manageable. The textual data underwent a preprocessing phase using NLTK, which included tokenization to convert the text into lowercase tokens, as well as the removal of punctuation and common stopwords. This step was important for enhancing the model's focus on meaningful words that contribute significantly to sentiment classification. Following preprocessing, we transformed the cleaned text data into a numerical format using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. The TfidfVectorizer generated a sparse matrix of TF-IDF features which allowed the model to capture the significance of words relative to their frequency across the dataset. This representation facilitated the SVM's ability to distinguish between different sentiments based on the relevant terms in the reviews. Similar to the Logistic Regression model, the SVM model was also trained and evaluated using an 80-20 split of the dataset.

The final model utilized in our analysis is DistilBERT, a transformer-based architecture that enhances the performance of sentiment classification tasks while maintaining computational efficiency. To begin, we loaded a dataset of cleaned reviews, and focused on the first 2,000 entries to expedite the training process. The dataset was divided into training and testing sets with an 80-20 ratio. We employed the DistilBERT tokenizer to preprocess the text data, which involved tokenizing the cleaned reviews, truncating and padding them to maintain uniform input lengths. This transformation is essential for preparing the data to be fed into the DistilBERT model. To facilitate model training and evaluation, we created a custom dataset class, SentimentDataset, which handled the organization of the encodings and corresponding labels. This allowed for smooth integration with PyTorch's data loading utilities. For training the model, we defined a set of hyperparameters through the TrainingArguments class by specifying parameters such as the number of epochs, batch size, and weight decay. The training process was conducted using the Trainer class from the Transformers library, which simplified the training loop and incorporated features such as logging and evaluation.

3 Results

The performance of the models was evaluated using key metrics such as precision, recall, F1-score, overall accuracy, and error rate (Figure 1). The Logistic Regression model achieved a negative precision of 0.88 and a recall of 0.84, resulting in an F1-score of 0.86 for the negative class. For the positive class, its precision and recall were 0.84 and 0.88, respectively, also yielding an F1-score of 0.86. The overall accuracy of the Logistic Regression model was 85.95%, with an error rate of 14.05%. Similarly, the SVM model showed comparable performance, with a negative precision of 0.87 and recall of 0.83, resulting in an F1-score of 0.85 for the negative class. The positive class metrics were slightly lower, with a precision of 0.84 and a recall of 0.88, and has an overall accuracy of 85.55% and an error rate of 14.45%. DistilBERT achieved a negative precision of 0.90 and a recall of 0.80, resulting in an F1-score of 0.85 for the negative class. For the positive class, it had a precision of 0.82 and a recall of 0.91, leading to an overall accuracy of 85.50% and an error rate of 14.50%. Overall, all models demonstrated strong performance with similar accuracy rates. However, the Logistic Regression model had slightly better precision and recall for negative sentiment, while SVM and DistilBERT performed well in positive sentiment metrics.

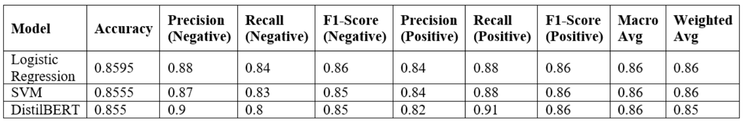
Confusion Matrix (Figure 2): For Logistic Regression, out of 996 actual negative samples, the model correctly identified 832 as true negatives while misclassifying 164 as false positives. In the positive class, it accurately predicted 887 true positives out of 1004 actual positives, with 117 false negatives. The SVM model exhibited similar performance, correctly classifying 827 true negatives and misclassifying 169 as false positives, along with 884 true positives and 120 false negatives from the positive class. In contrast, DistilBERT had a smaller dataset, with 199 actual negatives where it correctly identified 167 as true negatives and misclassified 32 as false positives. It also identified 177 true positives from 201 actual positives, resulting in 24 false negatives.

The ROC curve analysis for the Logistic Regression model (Figure 3) shows a strong performance with an AUC (Area Under the Curve) of 0.95 which indicates a higher level of predictive accuracy. This improvement is visually represented by a curve that remains well above the diagonal dashed line, which symbolizes random guessing (AUC = 0.5). The initial steep rise of the curve suggests that the model effectively achieves a high true positive rate while maintaining a low false positive rate at early thresholds. In comparison, the Support Vector Machine (SVM) model (Figure 4) demonstrates performance of an AUC of 0.94, that means a 94% probability of correctly ranking a positive instance higher than a negative one. The SVM's ROC curve shares a similar steep ascent. DistilBERT (Figure 5) also exhibits a strong AUC score of 0.93, which, while still impressive, now reflects a slightly lower performance compared to the updated Logistic Regression model. Its ROC curve is characterized by a noticeable stepped pattern, which is likely due to the model being trained on a limited dataset of only the first 2000 rows. This limitation may have affected the distribution of confidence scores, resulting in distinct jumps in the curve. However, DistilBERT still demonstrates a steep initial ascent.

4 Conclusion

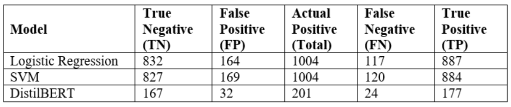
Each model demonstrated strong performance across key metrics such as precision, recall, F1-score, and overall accuracy. Despite their similar accuracy rates of 86%, subtle differences emerged in their handling of sentiment classifications. Logistic Regression excelled in identifying negative sentiment by achieving the highest precision and recall for this class, while SVM and DistilBERT showed stronger performance in positive sentiment detection. These findings indicate that while all models are effective, the choice of model may depend on the specific requirements of the application, such as the importance of accurately predicting one sentiment over another. Overall, the results suggest that utilizing a combination of traditional machine learning approaches and modern deep learning techniques can showcase valuable insights into sentiment analysis. Future work may explore further tuning of hyperparameters, the integration of additional features, or the use of ensemble methods to enhance predictive performance.

5 Tables and Figures



Model	Accuracy	Precision (Negative)	Recall (Negative)	F1-Score (Negative)	Precision (Positive)	Recall (Positive)	F1-Score (Positive)	Macro Avg	Weighted Avg
Logistic Regression	0.8595	0.88	0.84	0.86	0.84	0.88	0.86	0.86	0.86
SVM	0.8555	0.87	0.83	0.85	0.84	0.88	0.86	0.86	0.86
DistilBERT	0.855	0.9	0.8	0.85	0.82	0.91	0.86	0.86	0.85

Figure 1: Model Performance



Model	True Negative (TN)	False Positive (FP)	Actual Positive (Total)	False Negative (FN)	True Positive (TP)
Logistic Regression	832	164	1004	117	887
SVM	827	169	1004	120	884
DistilBERT	167	32	201	24	177

Figure 2: Confusion Matrix

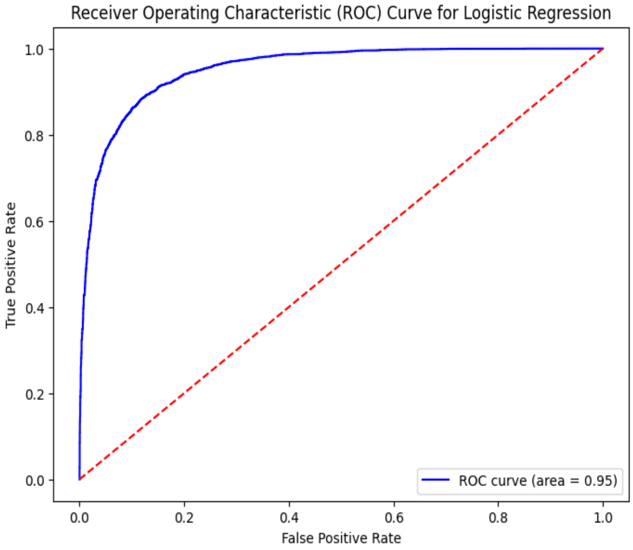


Figure 3: ROC Curve for Logistic Regression

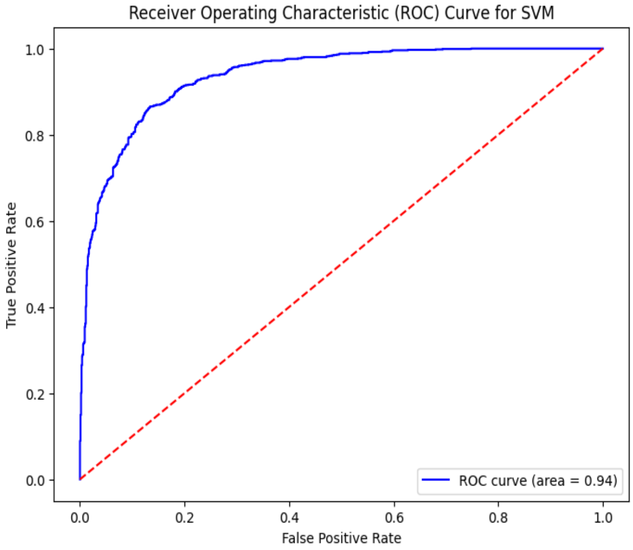


Figure 4: ROC Curve for Support Vector Machine

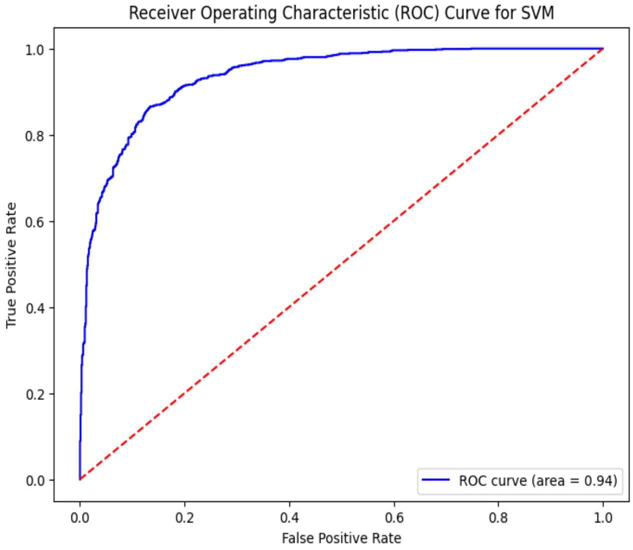


Figure 5: ROC Curve for DistilBERT