Analysis of customer satisfaction using sentiment analysis

Contents

1. Introduction

Sentiment analysis is a natural language processing (NLP) task that involves determining the sentiment or emotion expressed in a piece of text. This documentation outlines the process of sentiment analysis, from data collection to model evaluation, using a combination of manual and API-based data collection, data exploration, preprocessing techniques, and model development and training.

2. Objectives

The primary objectives of this sentiment analysis project are to classify comments into sentiment categories (positive, neutral, negative) and compare the performance of traditional machine learning (SVM) and deep learning (BERT) models.

3. Data Collection

3.1 YOUTUBE API

The data for sentiment analysis was collected using a combination of theYouTube API and manual copy/paste. The YouTube API was utilized for automated data retrieval. There was also another solution using the extension - Export YouTube Comments. Export YouTube Comments is a free tool that allows you to export and download YouTube comments and related user data from YouTube videos to Excel or Numbers in CSV format. As a dataset, we used the Lord of the Rings trailer comments.

3.2 Manual Copy/Paste

Manual copy/paste was employed for additional data collection to ensure diversity and coverage.

4. Data Exploration

4.1 Class Distribution

An analysis of the distribution of sentiment classes in the dataset was performed to understand the balance or imbalance of positive, negative, and neutral sentiments.

After the distribution into classes, it was discovered that there were significantly fewer negative comments than positive and neutral ones, new negative comments were added to the dataset manually.

```
Class Distribution:
negative    610
neutral     604
positive    598
Name: Sentiment, dtype: int64
```

4.2 Null Check

Null values were checked and handled appropriately to ensure data integrity and completeness.

5. Data Pre-processing

5.1 Stop-word Removal

Common stop-words were removed to eliminate noise and focus on meaningful words in the text.

5.2 Lowercasing

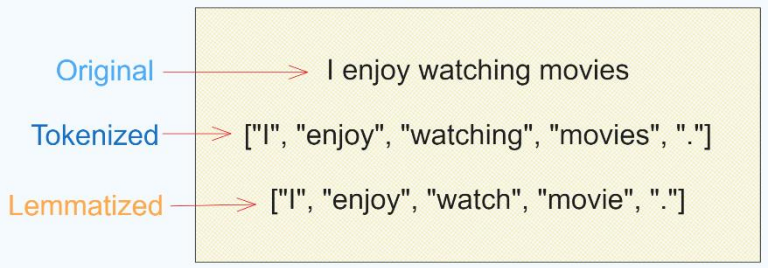
All characters in the text were converted to lowercase to maintain uniformity and reduce the dimensionality of the data.

5.3 Stemming and Lemmatization

Stemming and lemmatization were applied to reduce words to their base or root form, aiding in feature reduction and improving model efficiency.

5.4 Tokenization

The text data was tokenized, breaking it into individual words or tokens, to facilitate further analysis.



5.5 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF was applied to convert the tokenized text into numerical features, giving more weight to words that are important in a document but less frequent across the dataset.

6. Model Development

The next step involved developing and training sentiment analysis models using various approaches:

6.1 Support Vector Machine (SVM)

A Support Vector Machine was employed as a traditional machine learning model for sentiment classification.

Accuracy using this method: **0.68.**

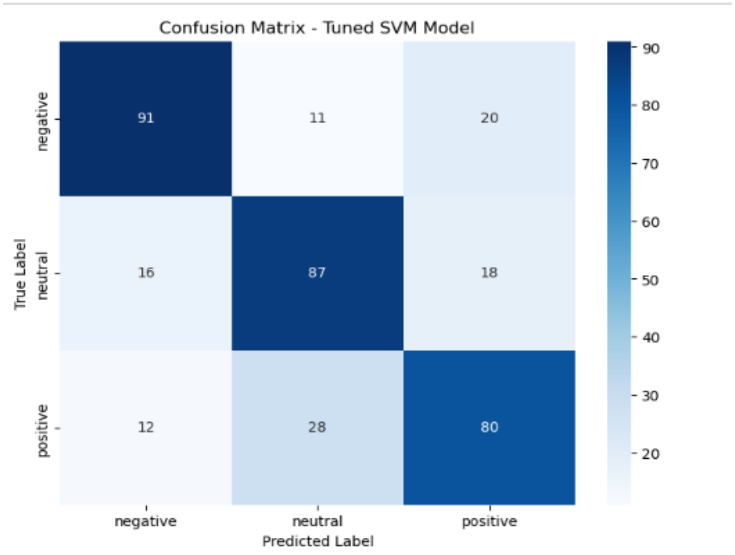
6.1.1 Hyperparameter Tuning- SVM

Grid search is employed to find optimal hyperparameters for the SVM model, including regularization parameter (C), kernel type, and gamma to improve overall performance of the Model.

SVM Model Accuracy with Hyperparameter tuning: **0.71**

6.1.2 Classification Report and Confusion Matrix:

	precision	recall	f1-score	support
0	0.76	0.75	0.76	122
1	0.69	0.72	0.70	121
2	0.68	0.67	0.67	120
accuracy			0.71	363
macro avg	0.71	0.71	0.71	363
weighted avg	0.71	0.71	0.71	363



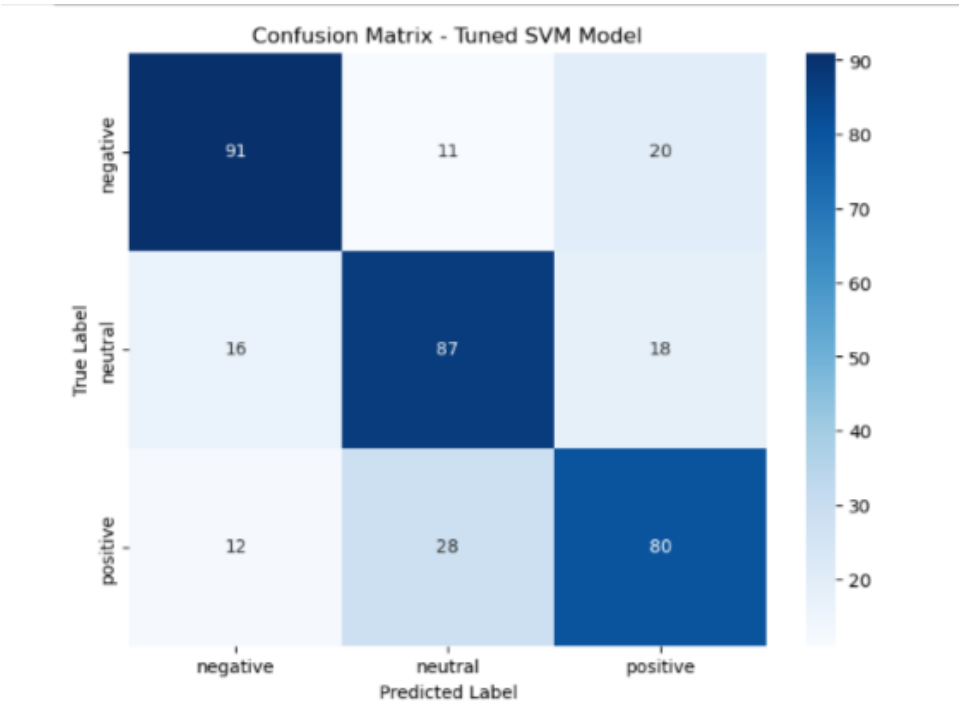
6.2 Ensemble Methods

Ensemble methods were used to combine multiple models to enhance predictive performance. It trains a voting classifier in Scikit-Learn consisting of three different classifiers: SVM, Logistic regression and Naive Bayes.

Voting Classifier without Hyperparameter Tuning:
Accuracy**: 0.67**

Voting Classifier with Hyperparameter Tuning:
Accuracy: **0.69**

Confusion Matrix for Voting Classifier:



6.3 BERT-based Solutions

State-of-the-art BERT models were utilized for sentiment analysis, taking advantage of pre-trained contextual embeddings. This method was chosen because it has the good advantage: bidirectionally trained (this is also its key technical innovation). This means it is possible to have a deeper sense of language context and flow compared to the single-direction language models.

BERT relies on a Transformer (the attention mechanism that learns contextual relationships between words in a text). A basic Transformer consists of an encoder to read the text input and a decoder to produce a prediction for the task. Since BERT's goal is to generate a language representation model, it only needs the encoder part. The input to the

encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network.

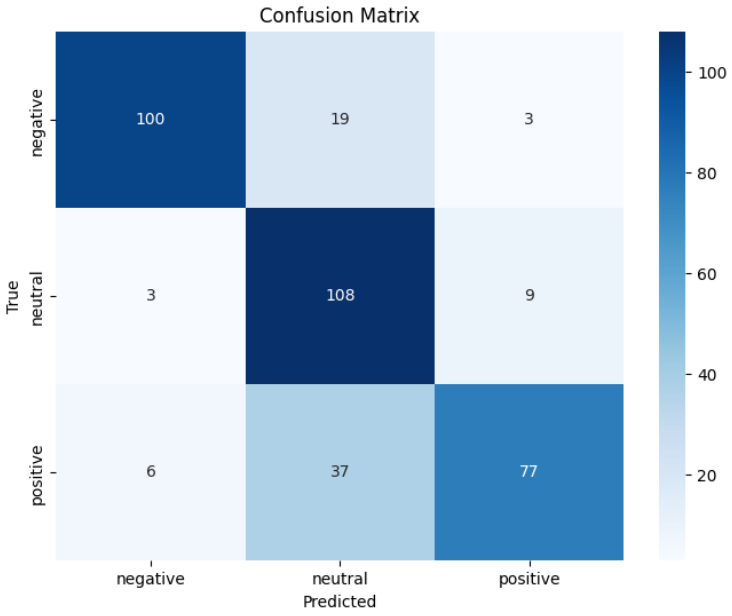
Training process:

- The data was divided into training, validation and testing sets.
- Sometimes during training, the accuracy was equal to 0.9638, but validation accuracy was equal to 0.7828.
- Overall result is:
  Average Performance Across 5 Folds => Final Average Train Accuracy: 0.7722, Final Average Validation Accuracy: 0.7920.

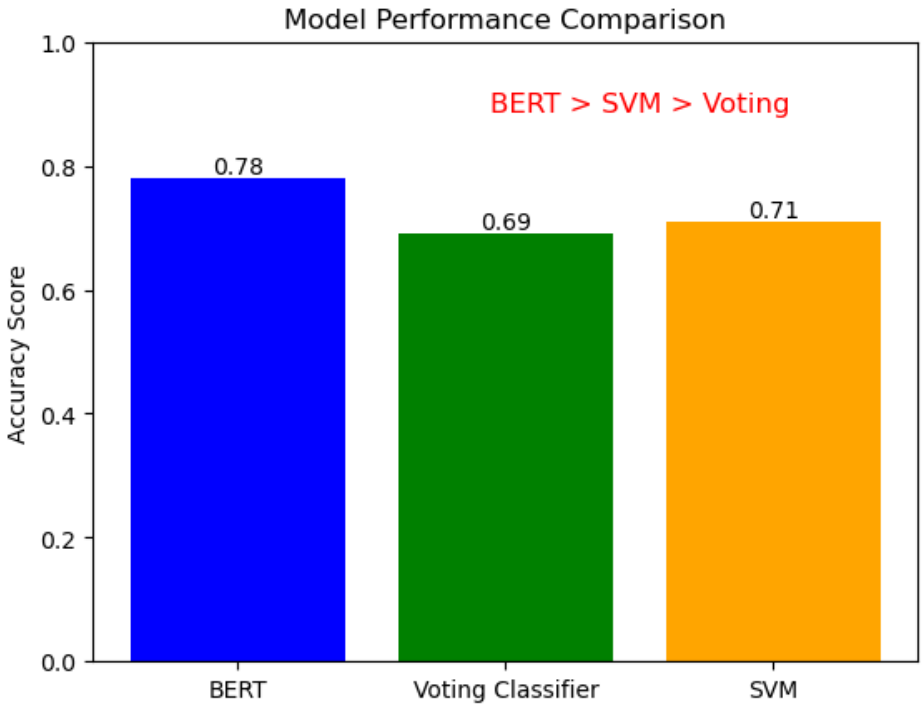
Evaluation process:

- Test Accuracy**: 0.78**

Confusion Matrix:



7. Comparative Analysis



- BERT achieved the highest accuracy score among the classifiers, with a value of 0.78.
- SVM follows closely with an accuracy score of 0.71.
- The Voting Classifier lags behind with an accuracy score of 0.69.

Based on confusion Matrix:

BERT Performance:

- BERT exhibits better performance in accurately predicting all three classes (Negative, Neutral, Positive) compared to SVM.
- BERT shows higher counts in the diagonal elements, indicating strong predictive capability across all classes.

SVM Performance:

- SVM demonstrates reasonable performance, but it appears to struggle more in distinguishing between classes, as evidenced by higher off-diagonal counts in the confusion matrix.

Recommendation:

- Based on the confusion matrices, BERT seems to be the more reliable model, achieving better accuracy in classifying instances across all sentiment classes.

8. Conclusion

This documentation provides a comprehensive overview of the sentiment analysis project, covering data collection, exploration, preprocessing, model development, and evaluation.

In conclusion, our comprehensive sentiment analysis project revealed that the BERT method outperformed other approaches. BERT can achieve higher accuracy and better adapt to the complexities of human language.

Future work should focus on making sentiment analysis tools more adaptable, robust, and valuable in real-world applications.

9. References

I. https://www.youtube.com/watch?v=x8UAUAuKNcU&t=4s&ab_channel=PrimeVideo

II. https://scikit-learn.org/stable/modules/svm.html

III. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

IV. https://pytorch.org/hub/huggingface_pytorch-transformers/

V. https://huggingface.co/docs/transformers/model_doc/bert

VI. https://docs.chatlayer.ai/understanding-users/natural-language-processing-nlp

VII. https://blog.hubspot.com/website/how-to-get-youtube-api-key

VIII. https://chat.openai.com/

Submitted By:

Raghu Bhetwal (7216790)

Alexandra Goraichuk(7213889)