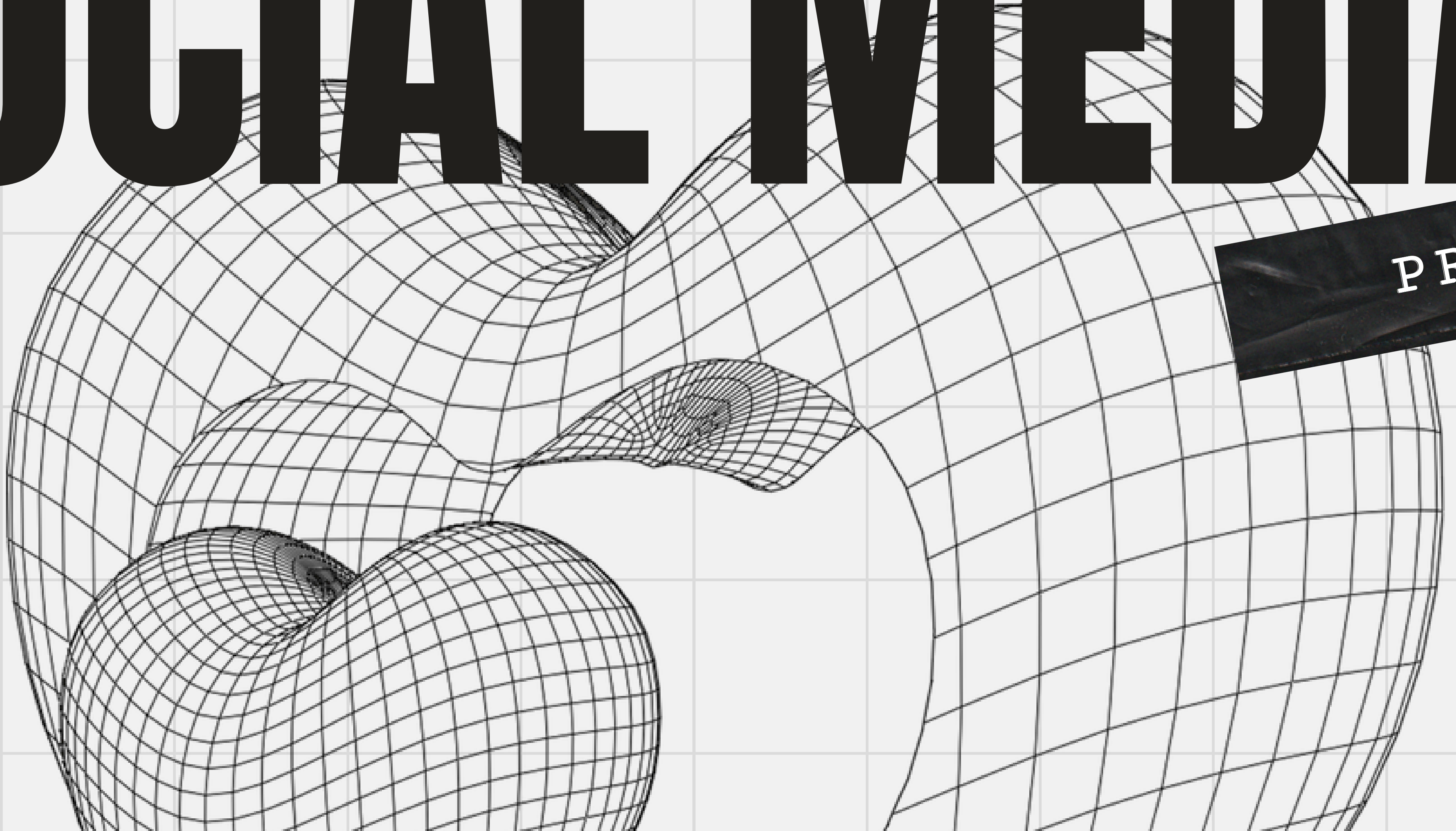
GROUP 12

JKLU

MACHINE LEARNING

SENTIMNETS ANALYSIS ON SOCIAL MEDIA

PROJECT



TEAM MEMBERS:

Aastha Gupta (2022btech002)
Jatin Choudhary (2022btech044)
Riya Singh (2022btech088)
Siddhi Nyati (2022btech101)

GROUP 12

TABLE OF CONTENT

2 PROBLEM STATEMENT

3 METHODOLOGIES USED

4 DATASET USED

5 EXPERIMENTS

6 RESULTS

7 INDIVIDUAL COBTRIBUTION

8 REFERENCE

9 Q&A

PROBLEM STATEMENT

- **Objective:** Explore and compare sentiment analysis methodologies on Instagram, Facebook, and Twitter.
- **Approach:** Evaluate traditional NLP techniques and advanced methods like LSTM, SVM, and Random Forest.
- **Goal:** Determine comparative performance, accuracy, and efficiency of these methods.
- **Impact:** Provide insights into public sentiment on social media and inform stakeholder decision-making.

GROUP 12

METHODOLOGIES USED

This section describes the methodologies used to conduct sentiment analysis, namely Natural Language Processing and Long Short-Term Memory. However, a combination of Support Vector Machine and Random Forest was also implemented.

NLP

NLP is a crucial component in understanding and processing human language data. It forms the foundation for sentiment analysis on social media platforms.

LSTM

LSTM networks excel at encoding temporal information and long-term dependencies in text data, making them ideal for capturing contextual nuances in sentiment analysis tasks, especially on social media.

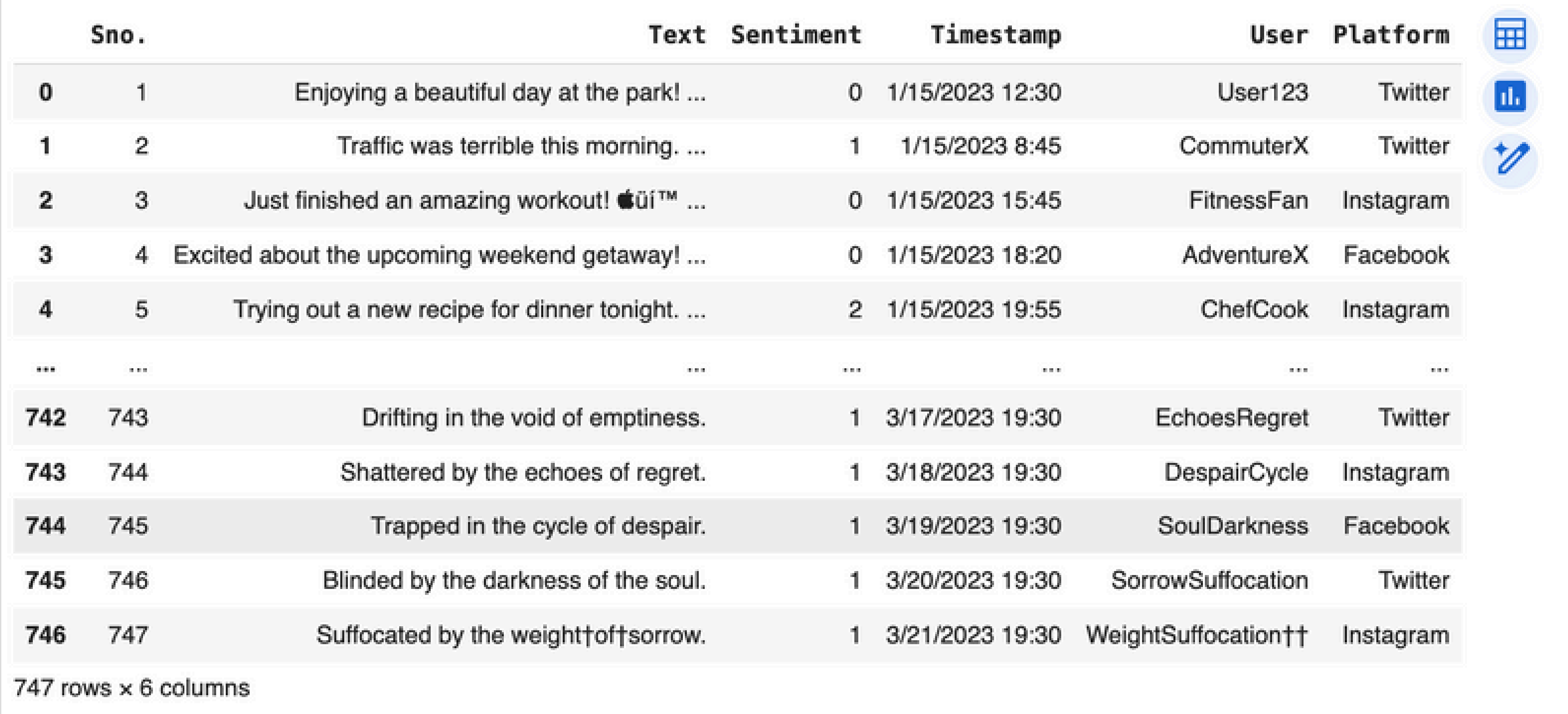
SVM & RANDOM FOREST

SVM is chosen for its ability to handle high-dimensional feature spaces and generalize well to unseen data. Random Forest, an ensemble learning method, aggregates the predictions of multiple decision trees, providing robust predictions even with noisy data.

GROUP 12

DATASET USED

- The dataset comprises 747 records collected from Kaggle for sentiment analysis.
- Each record contains text samples and corresponding sentiment labels.
- Sentiments: Positive, Negative, and Neutral
- The dataset offers diverse textual data from social media, providing insights into sentiment patterns.
- It facilitated training and evaluating sentiment analysis models to interpret sentiments expressed online.



	Sno.	Text	Sentiment	Timestamp	User	Platform
0	1	Enjoying a beautiful day at the park! ...	0	1/15/2023 12:30	User123	Twitter
1	2	Traffic was terrible this morning. ...	1	1/15/2023 8:45	CommuterX	Twitter
2	3	Just finished an amazing workout! üí™ ...	0	1/15/2023 15:45	FitnessFan	Instagram
3	4	Excited about the upcoming weekend getaway! ...	0	1/15/2023 18:20	AdventureX	Facebook
4	5	Trying out a new recipe for dinner tonight. ...	2	1/15/2023 19:55	ChefCook	Instagram
...	...	...	...	...	...	...
742	743	Drifting in the void of emptiness.	1	3/17/2023 19:30	EchoesRegret	Twitter
743	744	Shattered by the echoes of regret.	1	3/18/2023 19:30	DespairCycle	Instagram
744	745	Trapped in the cycle of despair.	1	3/19/2023 19:30	SoulDarkness	Facebook
745	746	Blinded by the darkness of the soul.	1	3/20/2023 19:30	SorrowSuffocation	Twitter
746	747	Suffocated by the weight†of†sorrow.	1	3/21/2023 19:30	WeightSuffocation††	Instagram

747 rows × 6 columns

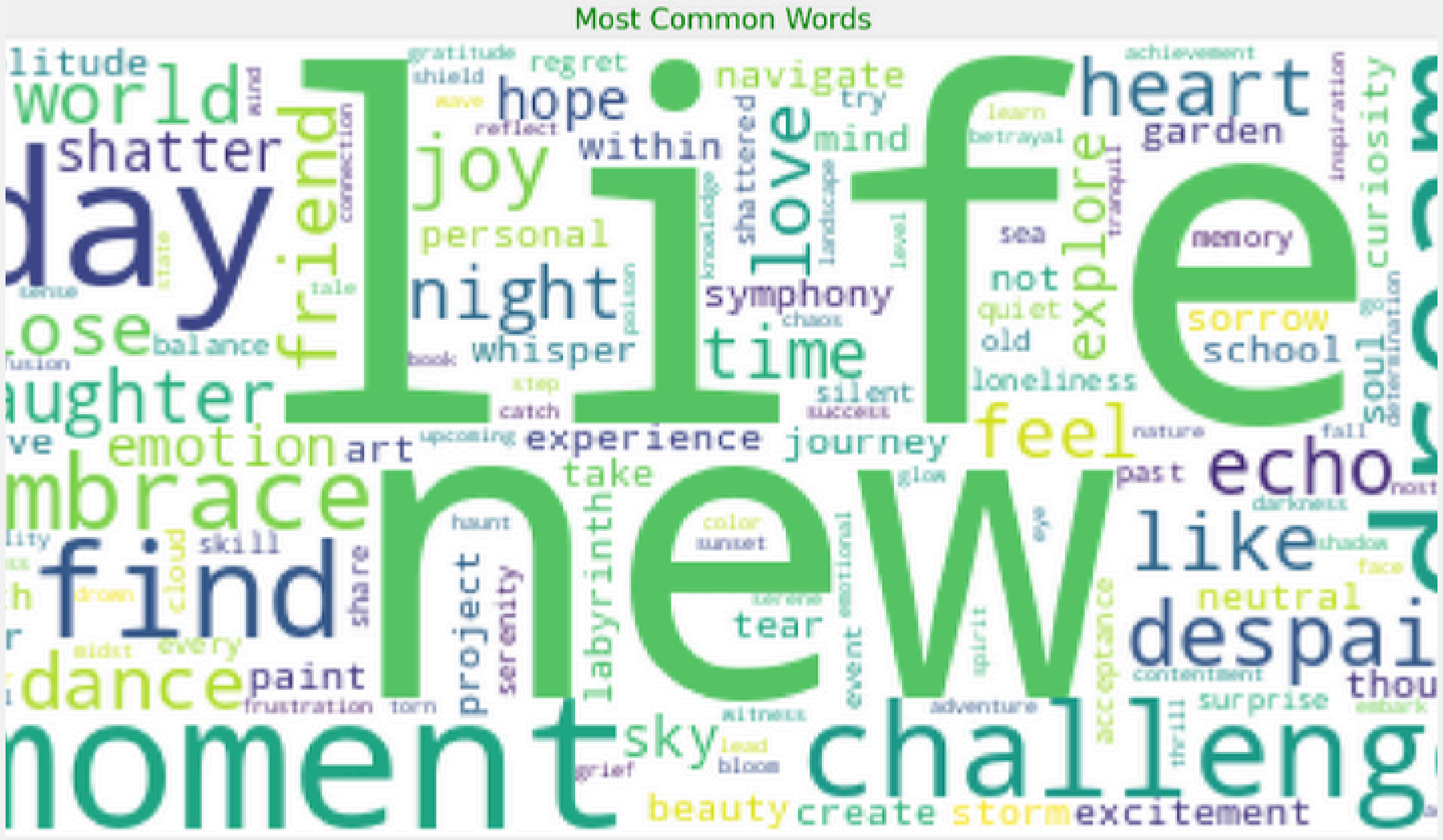
EXPERIMNETS

PROJECT

GORUP 12

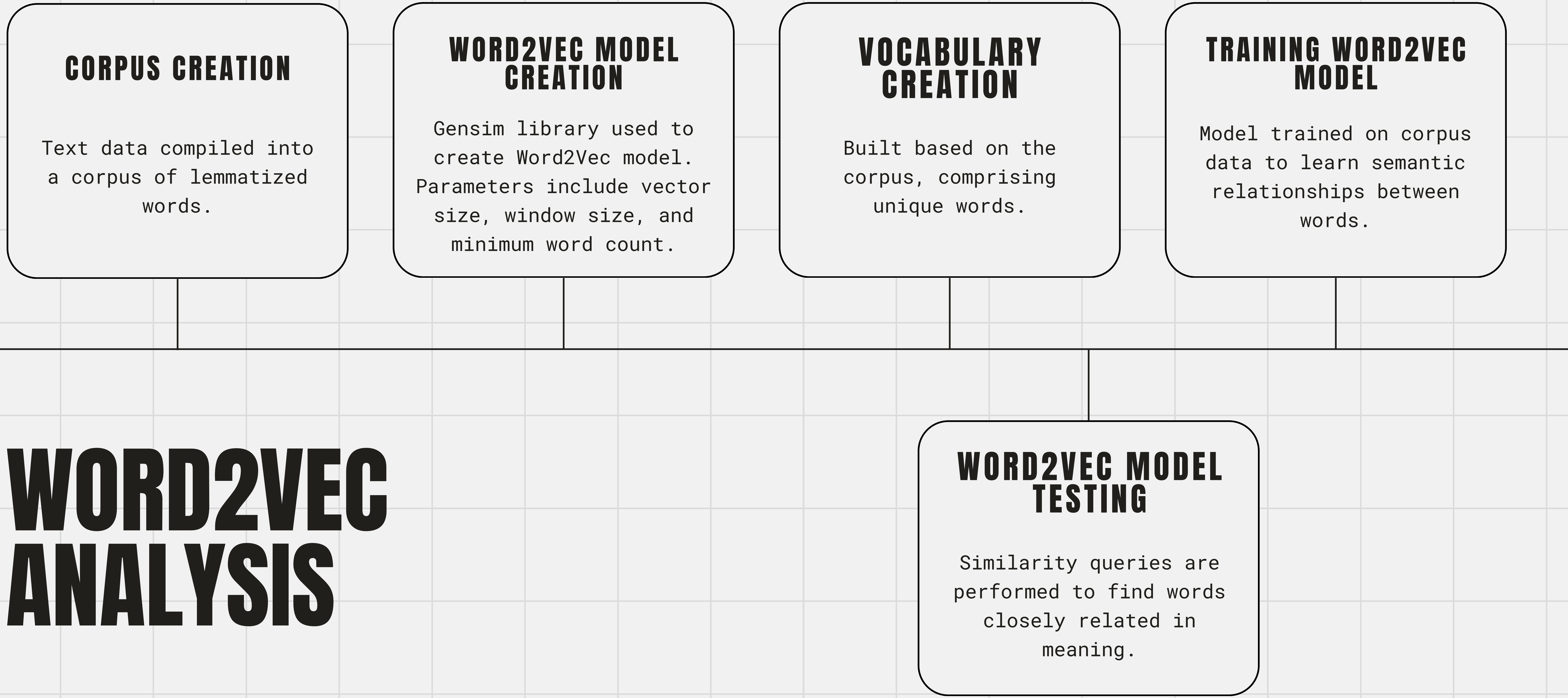
WORD CLOUD ANALYSIS

- Visualization generated based on lemmatized text data.
- Word clouds represent word frequency, with larger words indicating higher occurrence.
- Offers a quick and insightful view of prevalent themes and sentiments in the dataset.
- Enables identification of key topics and sentiments expressed in the text data.


Most Common Words

GROUP 12

WORD2VEC ANALYSIS



CORPUS CREATION

Text data compiled into a corpus of lemmatized words.

WORD2VEC MODEL CREATION

Gensim library used to create Word2Vec model. Parameters include vector size, window size, and minimum word count.

VOCABULARY CREATION

Built based on the corpus, comprising unique words.

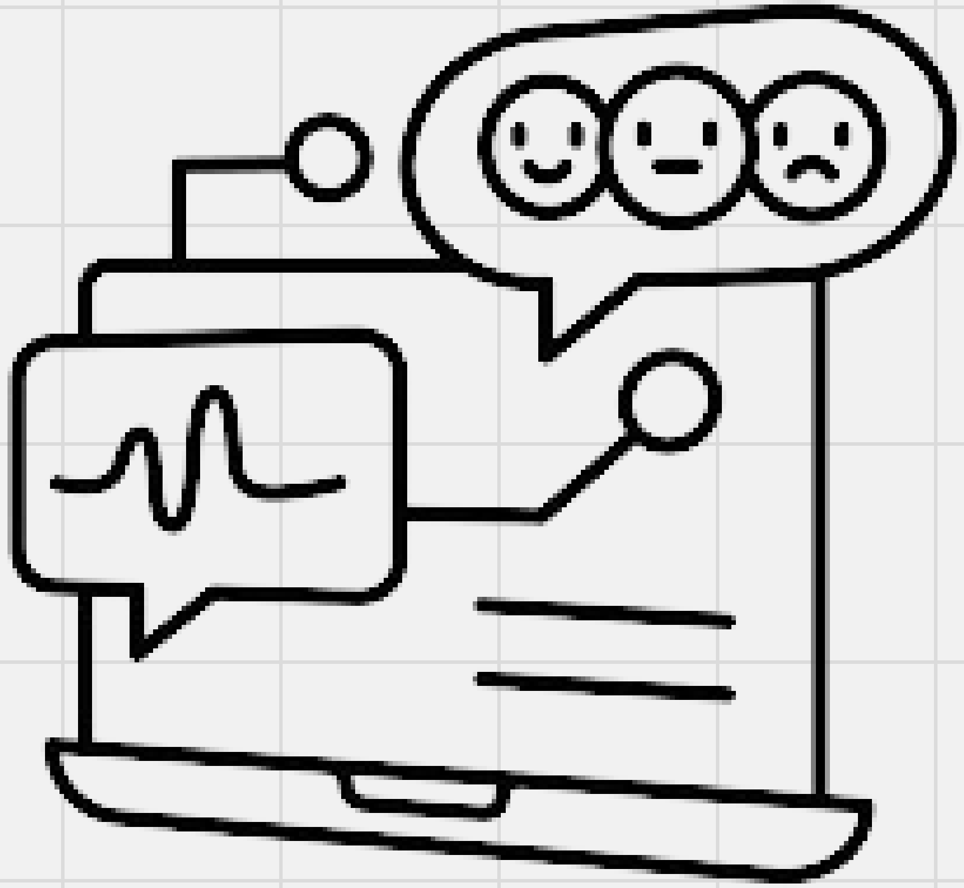
TRAINING WORD2VEC MODEL

Model trained on corpus data to learn semantic relationships between words.

WORD2VEC MODEL TESTING

Similarity queries are performed to find words closely related in meaning.

RESULT



NATURAL LANGUAGE PROCESSING(NLP)

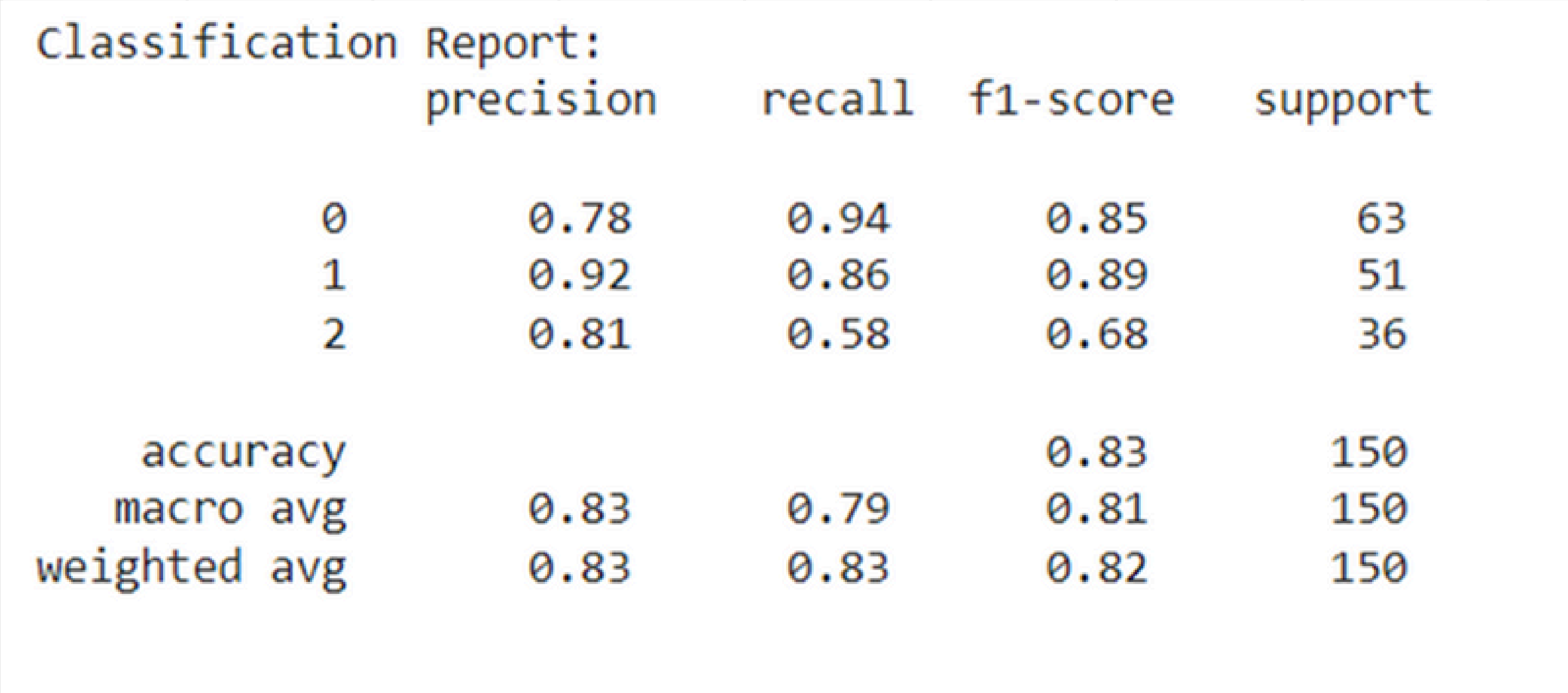
1.Dataset Pre-processing: Dataset preprocessing in NLP involves cleaning and standardizing raw text data by lowercasing, tokenizing, removing punctuation and stopwords, stemming or lemmatizing words, handling numerical values and special characters, and splitting the data for analysis and modelling.

2.Model Implementation: "I oversaw the creation of an NLP sentiment analysis model utilizing Python, TensorFlow, and scikit-learn." In my position, I specialized in constructing and educating deep learning structures using TensorFlow, while also making use of scikit-learn for processing text, extracting features, and evaluating models. I made sure to smoothly integrate these libraries in order to enhance the NLP model's efficiency.

3.Assessment and Examination: Carried out thorough assessment and analysis of the trained model's performance by employing common metrics like accuracy, precision, recall, and F1-score. Evaluated the model's advantages and disadvantages, pinpointed areas that need enhancement, and proposed suggestions for upcoming versions.

AASTHA GUPTA

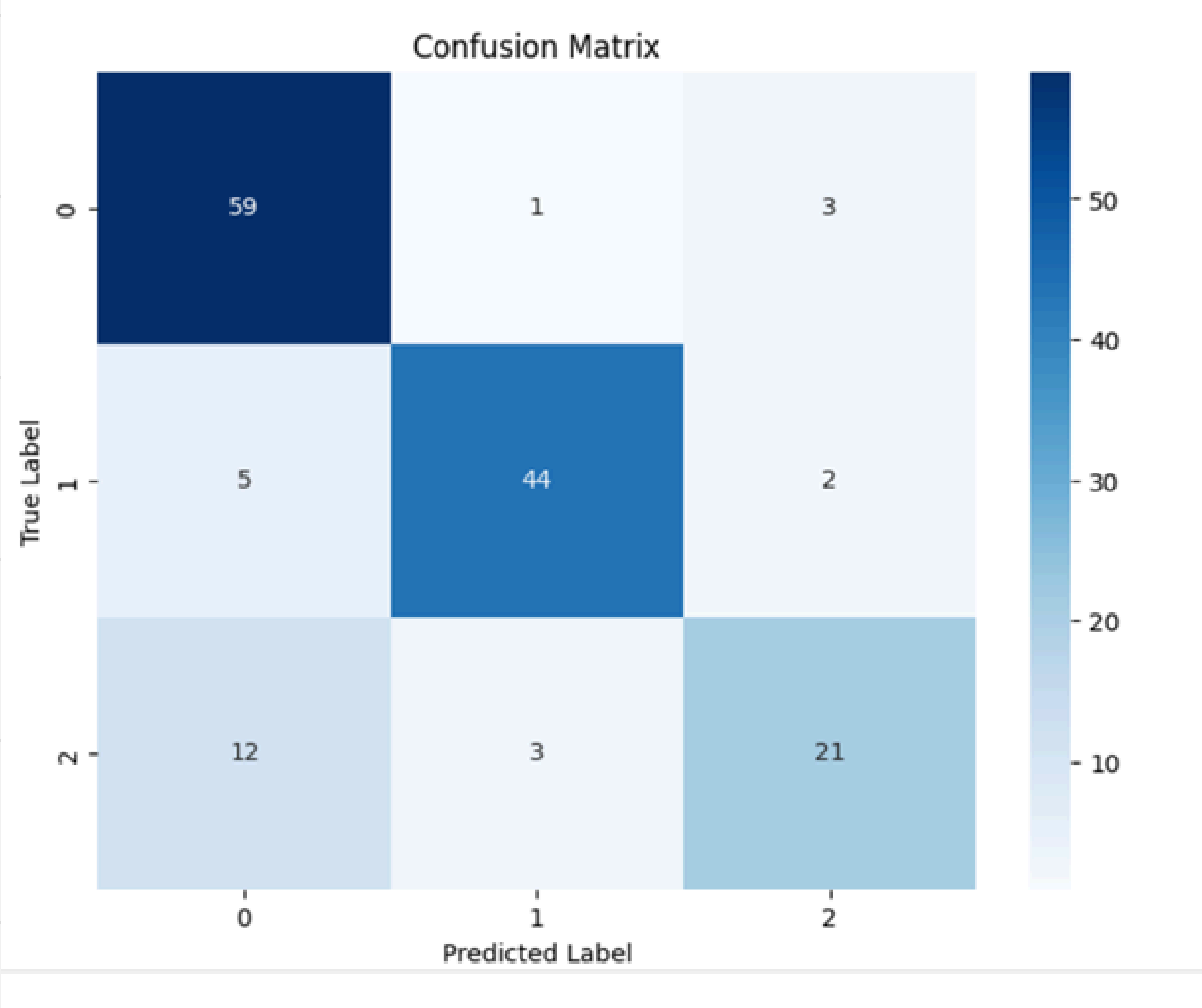
RESULT:

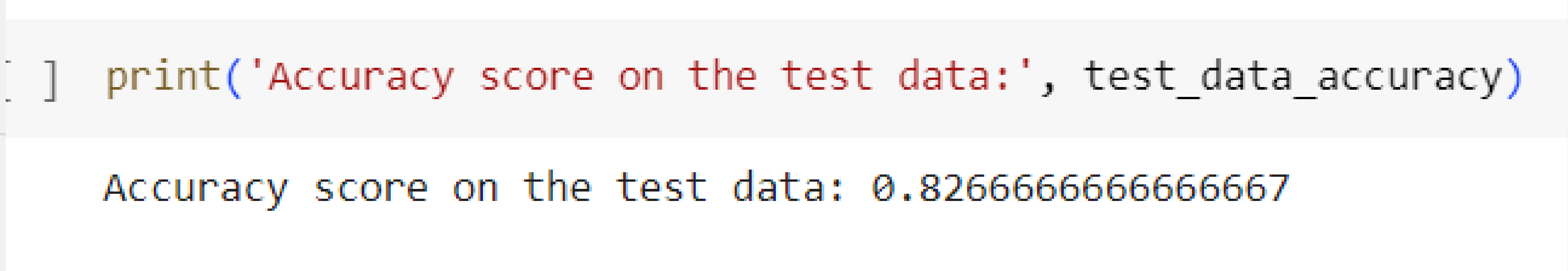


```
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.94      0.85        63
           1       0.92      0.86      0.89        51
           2       0.81      0.58      0.68        36

    accuracy                           0.83       150
   macro avg       0.83      0.79      0.81       150
weighted avg       0.83      0.83      0.82       150
```





```
[ ] print('Accuracy score on the test data:', test_data_accuracy)

    Accuracy score on the test data: 0.8266666666666667
```

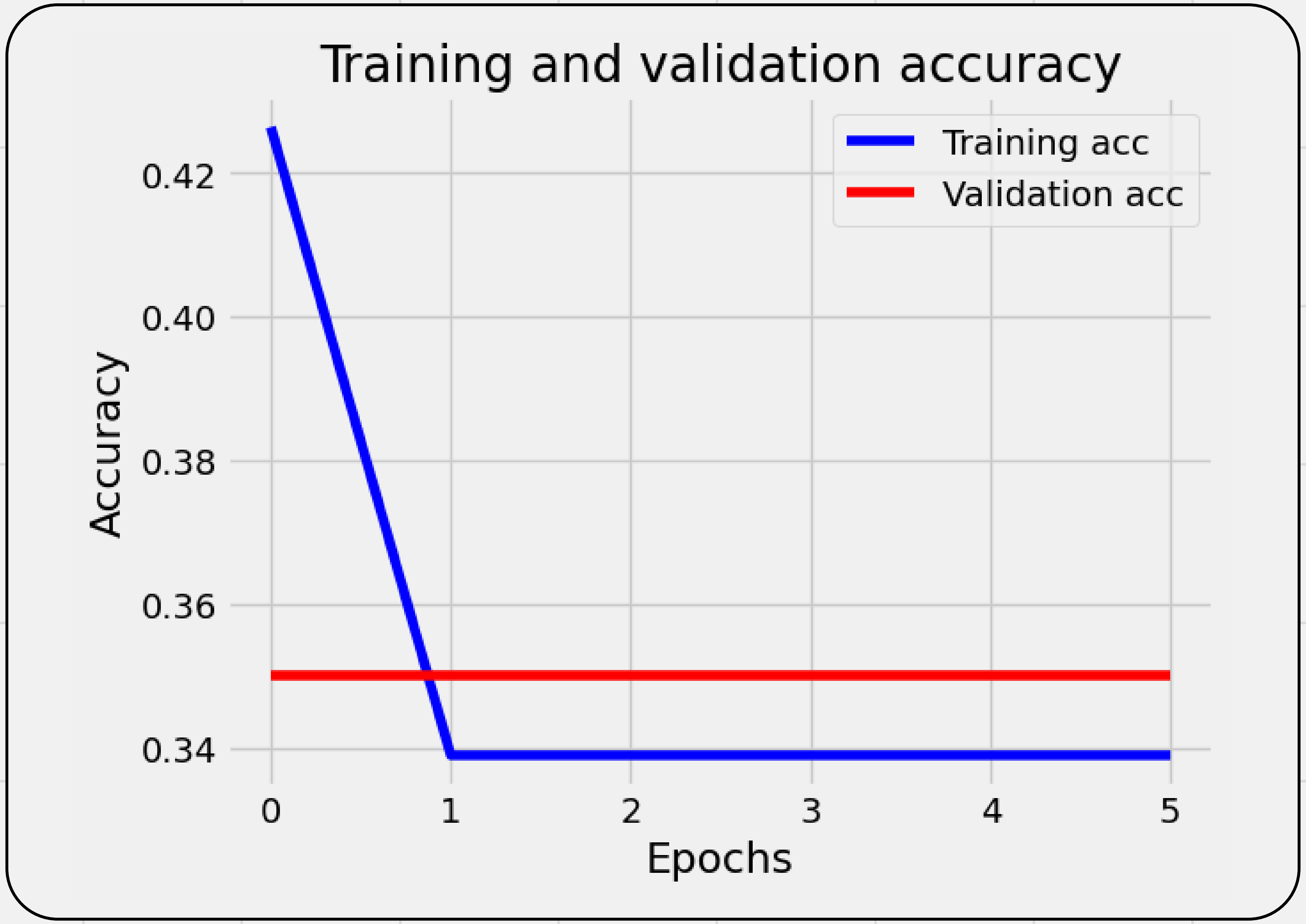
AASTHA GUPTA

GROUP 12

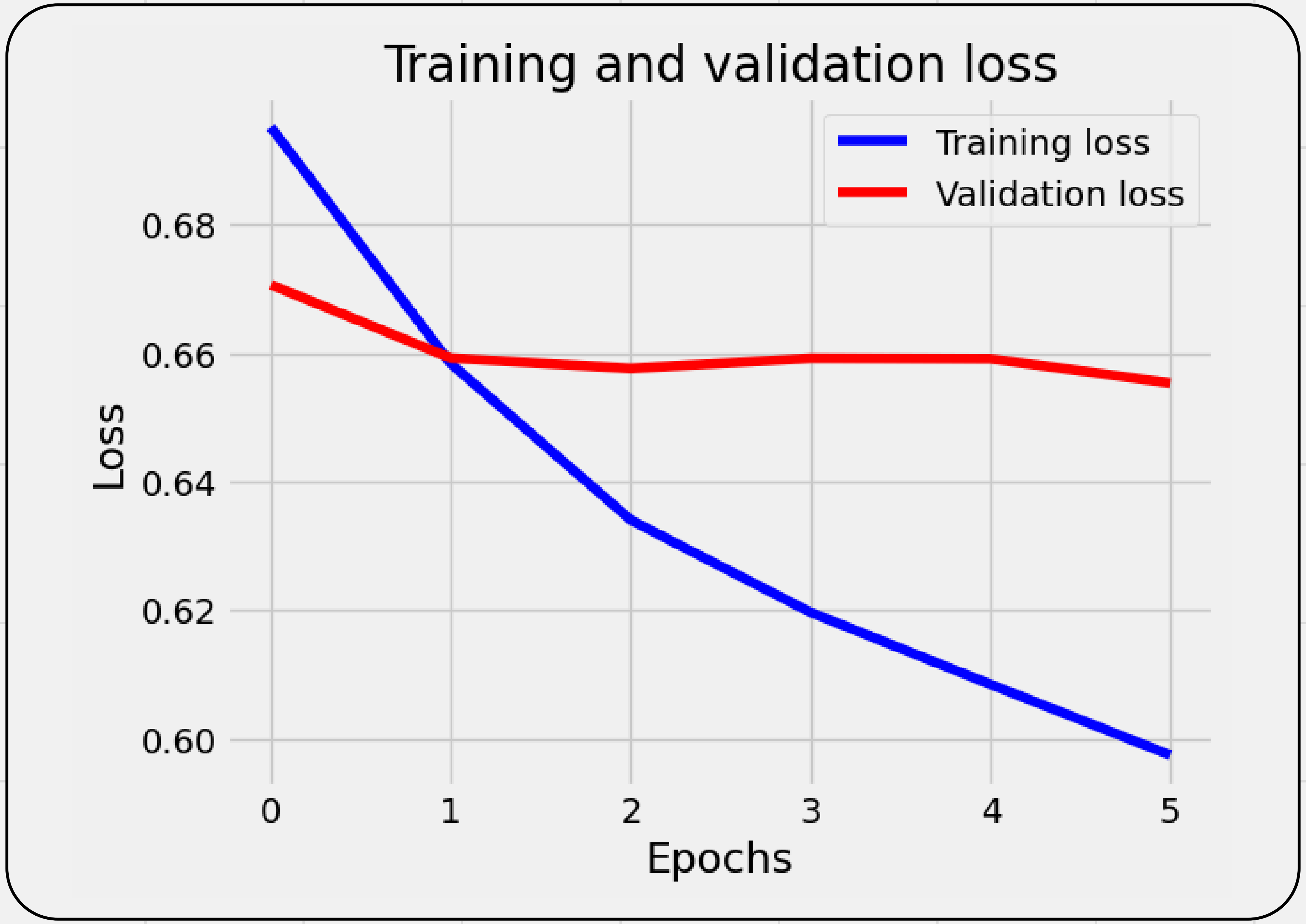
LSTM MODEL RESULT

The LSTM model achieved an accuracy of approximately **34%** on the test dataset.

Visualizing the training and validation accuracy and loss over epochs shows a typical pattern of deep learning training, with fluctuating performance indicative of a model that may require further tuning



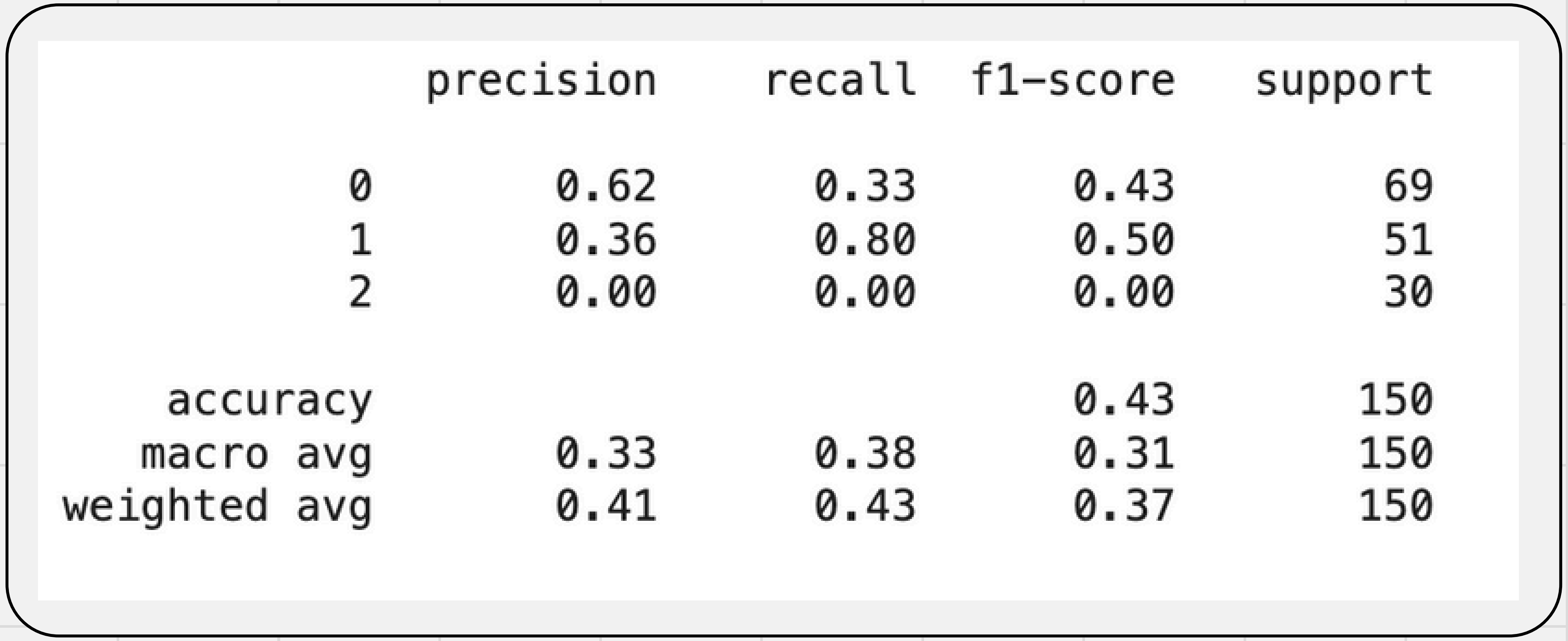
Training and validation accuracy



Training and validation loss

CLASSIFICATION REPORT

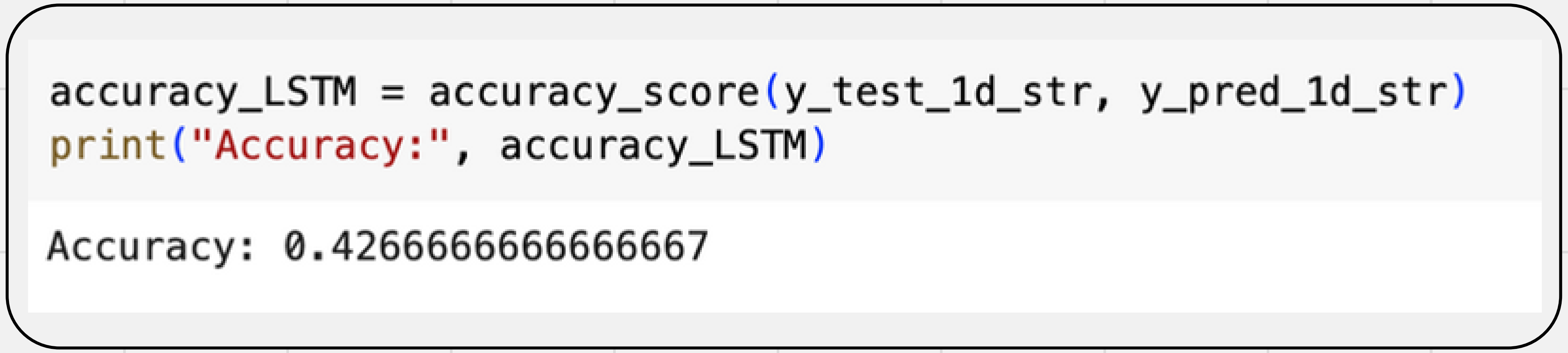
The classification report reveals precision, recall, and F1-score for each sentiment class. Notably, the precision and recall for the neutral class are lower compared to the other classes, indicating challenges in classifying neutral sentiments accurately.



```
              precision    recall  f1-score   support

           0       0.62      0.33      0.43        69
           1       0.36      0.80      0.50        51
           2       0.00      0.00      0.00        30

    accuracy                           0.43       150
   macro avg       0.33      0.38      0.31       150
weighted avg       0.41      0.43      0.37       150
```



```
accuracy_LSTM = accuracy_score(y_test_1d_str, y_pred_1d_str)
print("Accuracy:", accuracy_LSTM)
```

```
Accuracy: 0.4266666666666667
```

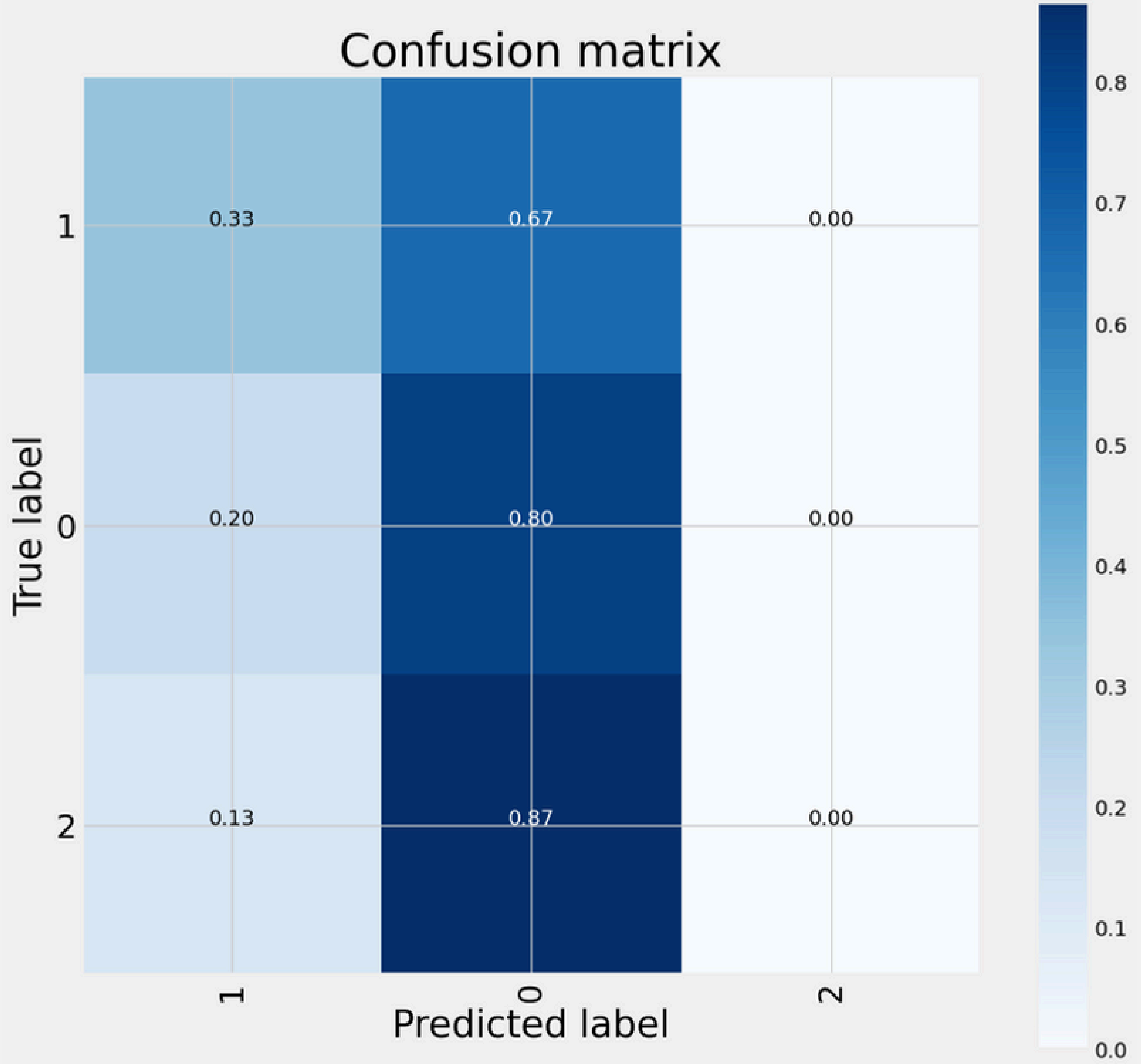
ACCURACY SCORE

The overall accuracy score is calculated, which is 42.67%.

GROUP 12

CONFUSION MATRIX

The confusion matrix provides a visual representation of the model's performance across different sentiment classes.



SUPPORT VECTOR MACHINE(SVM)

- Token and Vocab Creation

In SVM, instead of using neural network-specific tokenization techniques like in deep learning models, we often use methods like TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical vectors. The TfidfVectorizer from scikit-learn is utilized to perform this transformation. It tokenizes the text and computes the TF-IDF scores for each word, resulting in a vocabulary size of 1650 unique words.

- Label Encoding

The sentiment labels are encoded directly as numerical values using label encoding. This step prepares the target variable for training the SVM classifier.

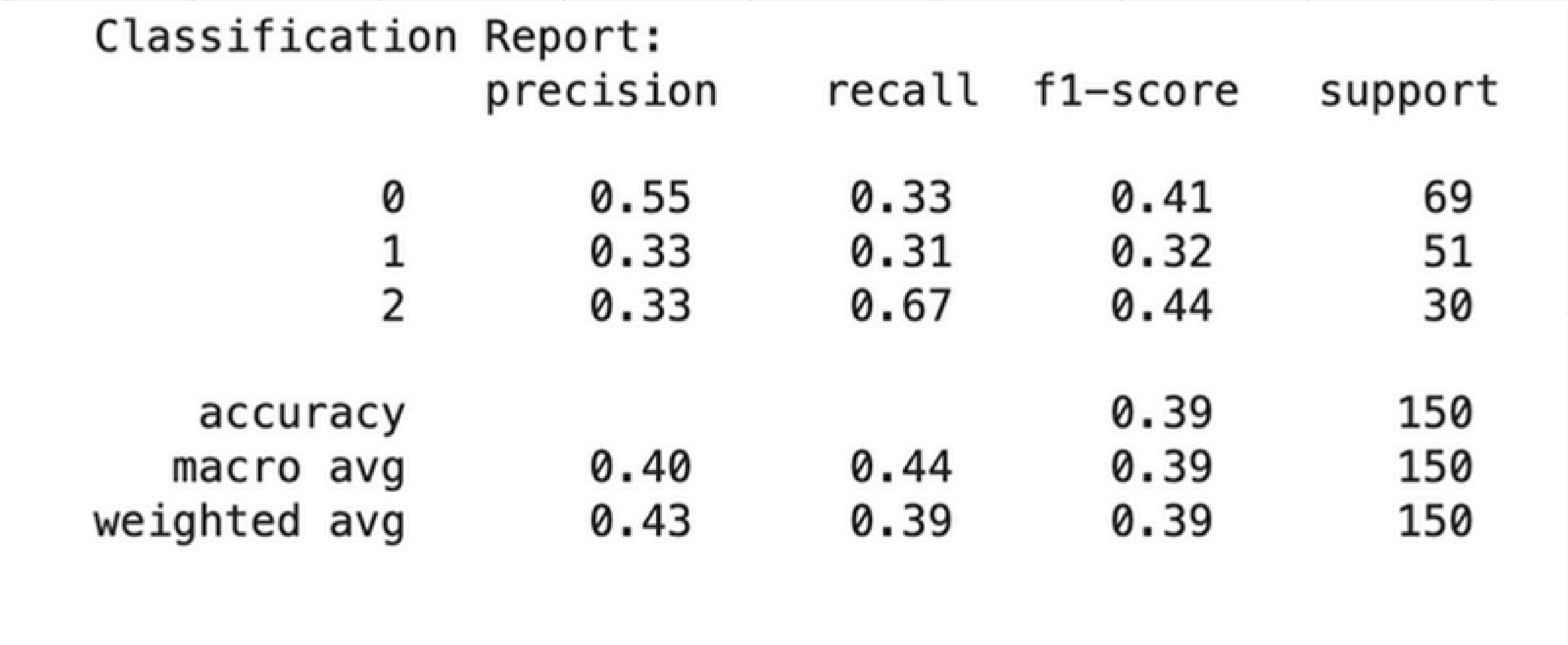
- Model Creation - SVM

The SVM classifier is instantiated with a linear kernel. The choice of kernel function can have a significant impact on the performance of the SVM model.

- Model Training

The SVM classifier is trained on the training data (x_train) along with the corresponding sentiment labels (y_train). During training, the SVM algorithm learns to find the optimal hyperplane that separates the different sentiment classes in the feature space.

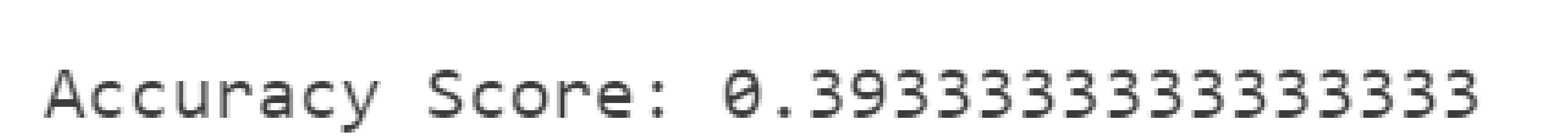
RESULT:



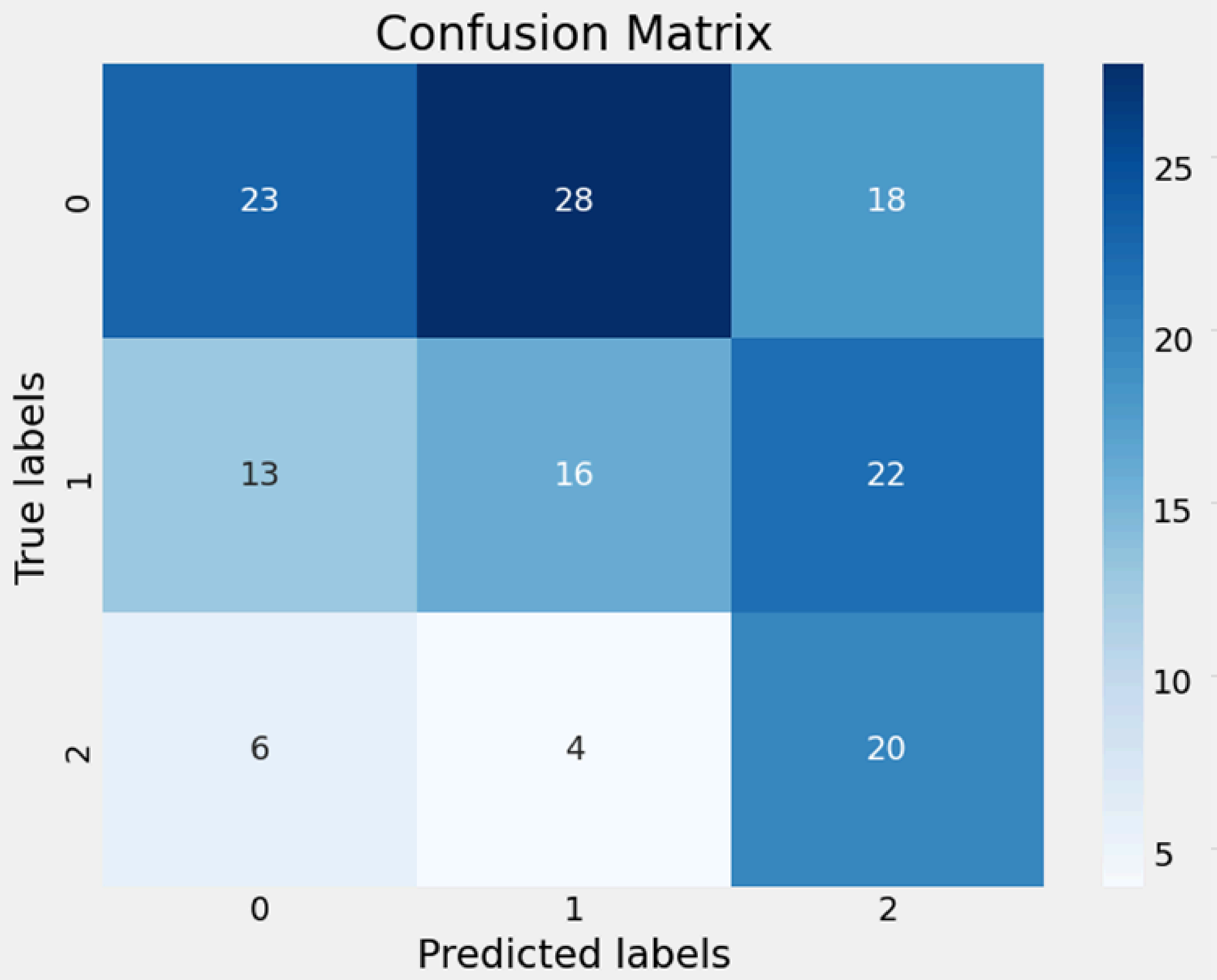
```
   Classification Report:
                 precision    recall  f1-score   support

              0       0.55      0.33      0.41        69
              1       0.33      0.31      0.32        51
              2       0.33      0.67      0.44        30

       accuracy                           0.39       150
      macro avg       0.40      0.44      0.39       150
   weighted avg       0.43      0.39      0.39       150
```



```
Accuracy Score: 0.39333333333333331
```



RANDOM FORREST

- **Token and Vocab Creation**

Similar to the SVM approach, TF-IDF vectorization is used to convert text into numerical vectors. The TfidfVectorizer from scikit-learn tokenizes the text and computes TF-IDF scores for each word, resulting in a vocabulary size of 1650 unique words.

- **Label Encoding**

The sentiment labels are directly encoded as numerical values using label encoding, preparing the target variable for training the Random Forest classifier.

- **Model Creation**

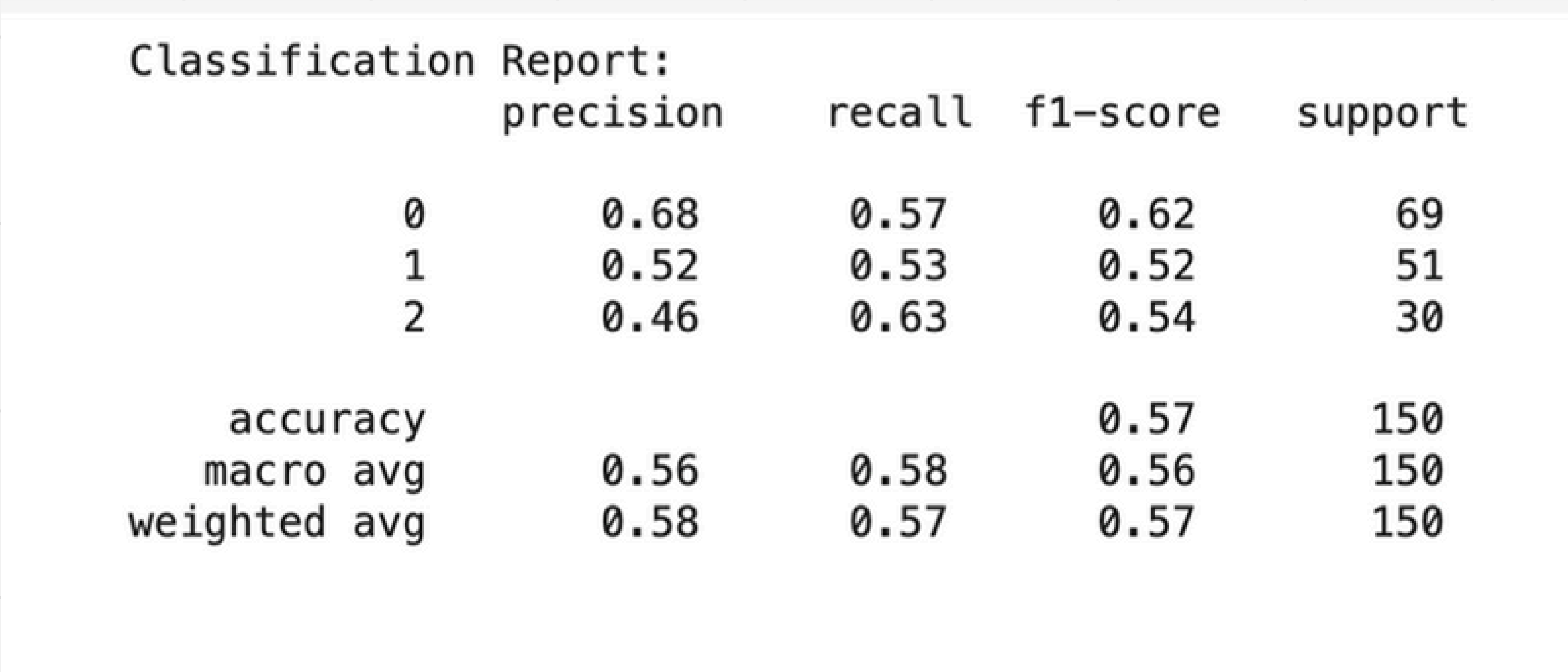
A Random Forest classifier is initialized with 100 decision trees using the RandomForestClassifier from scikit-learn. This step sets up the ensemble of decision trees for sentiment classification.

- **Model Training**

The Random Forest classifier is trained on the training data (x_train) along with the corresponding sentiment labels (y_train_RF). During training, the Random Forest algorithm builds multiple decision trees based on bootstrapped samples from the training data and averages predictions across the trees to make the final classification.

RIYA SINGH

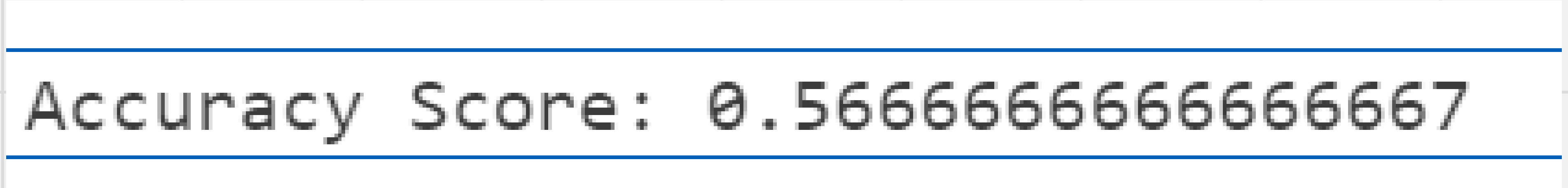
RESULT:



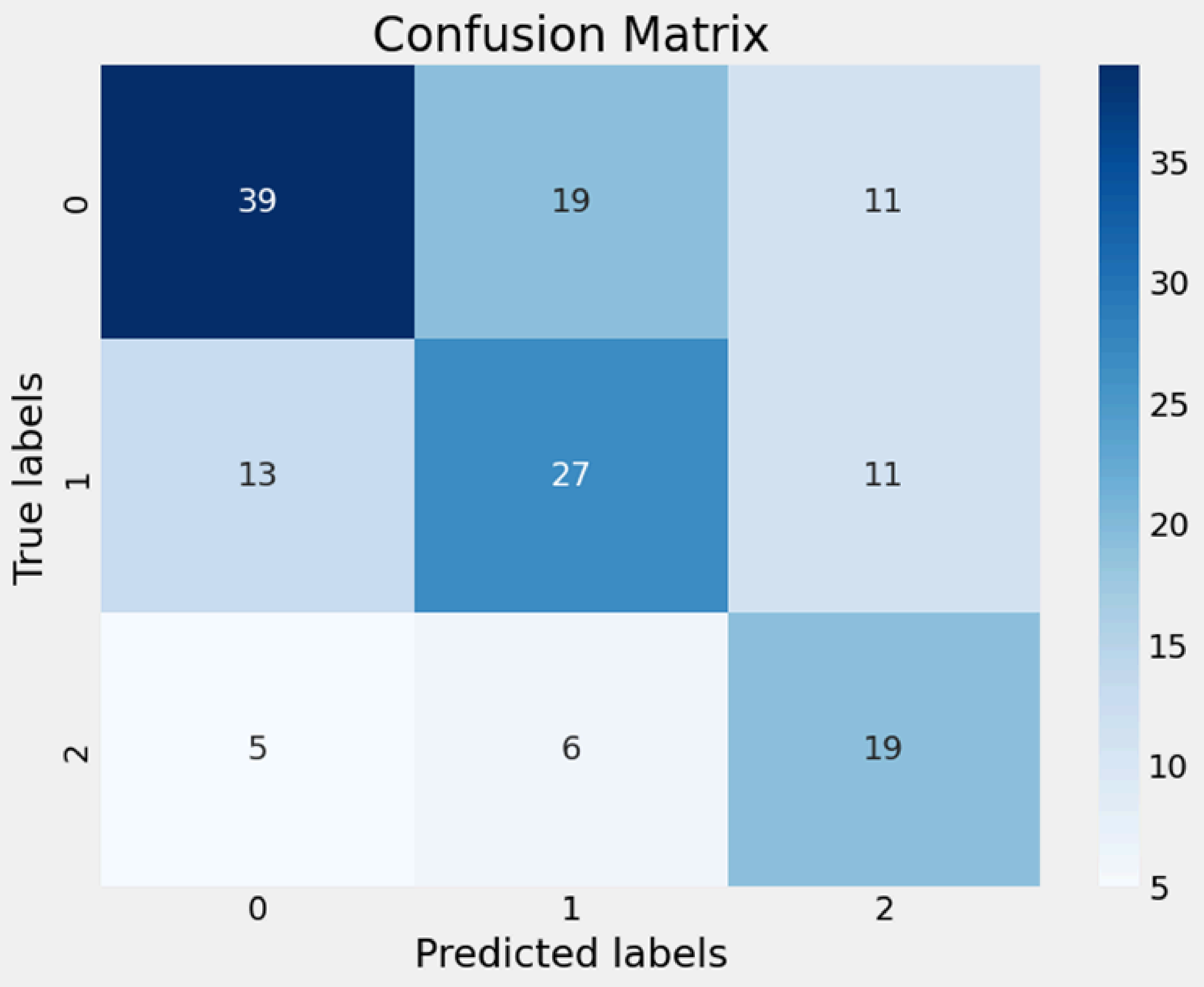
```
Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.57      0.62        69
           1       0.52      0.53      0.52        51
           2       0.46      0.63      0.54        30

    accuracy                           0.57       150
   macro avg       0.56      0.58      0.56       150
weighted avg       0.58      0.57      0.57       150
```



```
Accuracy Score: 0.5666666666666667
```



RIYA SINGH

COMPARISON OF DIFFERENT METHOD USED:

NLP

The NLP outperformed all the method applied on this data, achieving an accuracy of approximately 82.67% on the test data.

Random Forest

The Random Forest model, achieved an accuracy of approximately 56.7% on the test data.

LSTM

The LSTM model achieved an accuracy of approximately 34% on the test dataset.

SVM

The SVM classifier attained an accuracy of around 39.33% on the test set.

RESEARCH PAPER 1:

- The research paper focused solely on sentiment analysis using Support Vector Machines (SVM) and achieved an accuracy of around 80%. In contrast, our study employed four models: Long Short-Term Memory (LSTM), SVM, Random Forest, and Natural Language Processing (NLP). However, our accuracies varied: LSTM at approximately 42%, SVM around 40%, Random Forest at about 55%, and NLP with the highest accuracy at 83%.

- While the research paper outperformed our study in overall accuracy, our approach offered a broader perspective by exploring multiple models. This allowed for a comparative analysis of different methodologies, revealing insights into their strengths and weaknesses in sentiment analysis. Although the SVM model in the research paper excelled, our study's inclusion of diverse techniques, notably NLP, showcased promising results, suggesting the potential for advanced language processing techniques to enhance sentiment analysis tasks.

RESEARCH PAPER 2:

- In the research paper, sentiment analysis was conducted using K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Naive Bayes classifiers, achieving high accuracies: KNN at 98.2%, SVM at 90.5%, and Naive Bayes at 99%. Conversely, our study explored a broader array of models including Long Short-Term Memory (LSTM), SVM, Random Forest, and Natural Language Processing (NLP), with accuracies varying: LSTM at approximately 42%, SVM at around 40%, Random Forest at about 55%, and NLP with the highest accuracy at 83%.

- While the research paper achieved remarkable accuracies across all three models, our study exhibited lower overall accuracy rates. However, the inclusion of diverse methodologies in our research allowed for a more comprehensive understanding of sentiment analysis approaches. Although our individual model accuracies were lower compared to the specialized classifiers in the research paper, the exploration of NLP in our study demonstrated promising results, hinting at the potential for advanced language processing techniques to enhance sentiment analysis tasks.

THANK YOU