Comparison Between Machine Learning & Deep Learning in Sentiment Analysis using NLP

1. Introduction

The goal of this project is to create a strong sentiment analysis system that can categorize text data (such as product reviews or social media posts) into positive, negative, or neutral emotions. Understanding public opinion, customer feedback, and social media trends is essential for completing this task. We aim to utilize and evaluate conventional machine learning and deep learning techniques for sentiment analysis, prioritizing accuracy and efficiency.

2. Data Description

- **Source**: Kaggle - [Social Media Sentiments Analysis Dataset](https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset)

- **Total Instances:** 732 rows

- **Features:** Text, Sentiment, Timestamp, User, Platform, Hashtags, Likes, Retweets, Country, Year, Month, Day, Hour

- **Division between training and testing:** The data is split into training (70%), validation (15%), and testing (15%) sets.

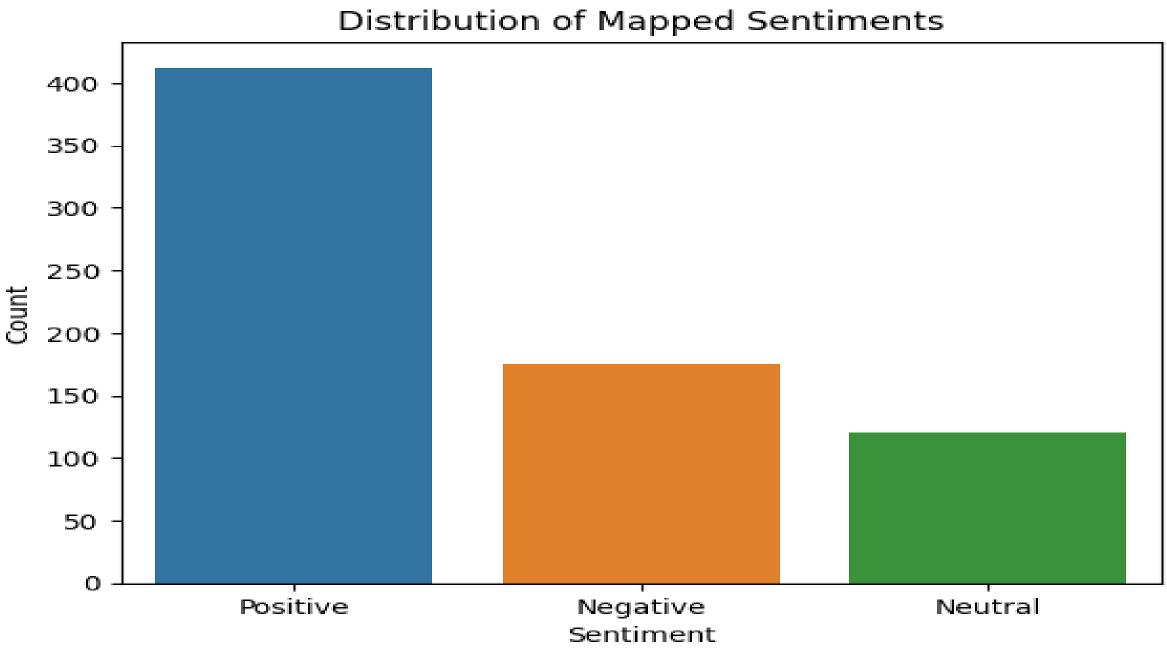
3. Baseline experiments

- **Goal:** Create baseline performance using traditional machine learning model with Naïve Bayes.

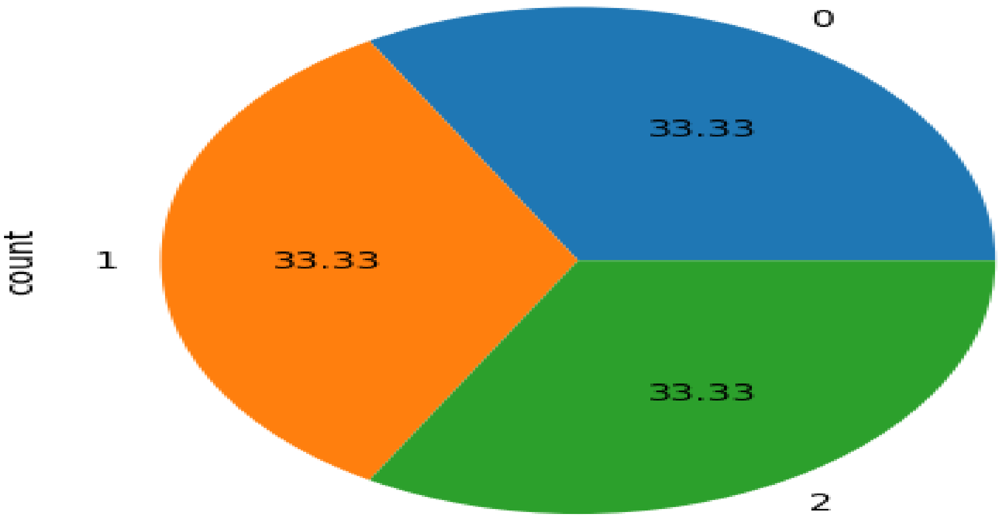
- **Present initial experiments**

- **Exploration and Preprocessing:** Apply **`RandomOverSampler'** to balance text data which was cleaned by removing non-alphabetic characters, tokenizing, removing stopwords, and applying stemming and lemmatization using **NLTK** library.

Imbalanced labels:



Balance labels:



- **Feature Extraction**: Used Term Frequency-Inverse Document Frequency (**TF-IDF**) to convert text data into numerical features.

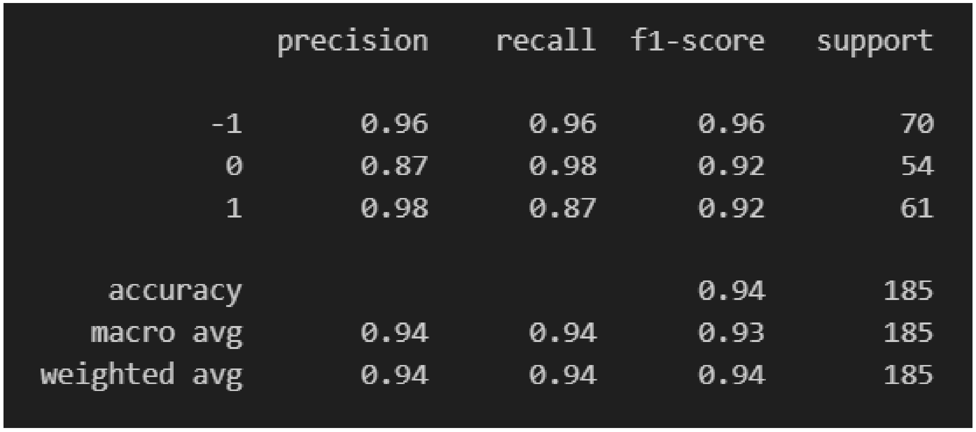
- **Model Training:** A Naive Bayes classifier was trained on the TF-IDF features.

- **Hyperparameter Tuning:** Performed **randomized search** to optimize hyperparameters for the **TF-IDF vectorizer** and **Naive Bayes classifier**.

- **Conclusions:**

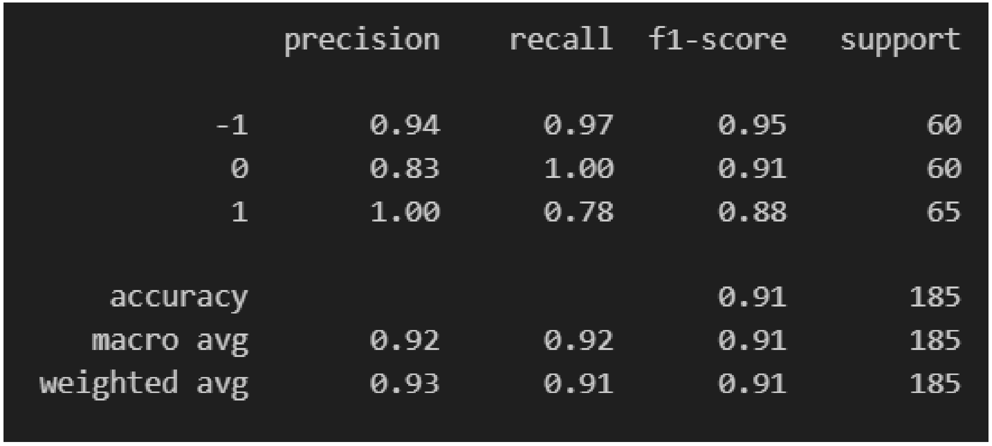
- The best hyperparameters were identified through **randomized search**

- Classification Report for validation data:



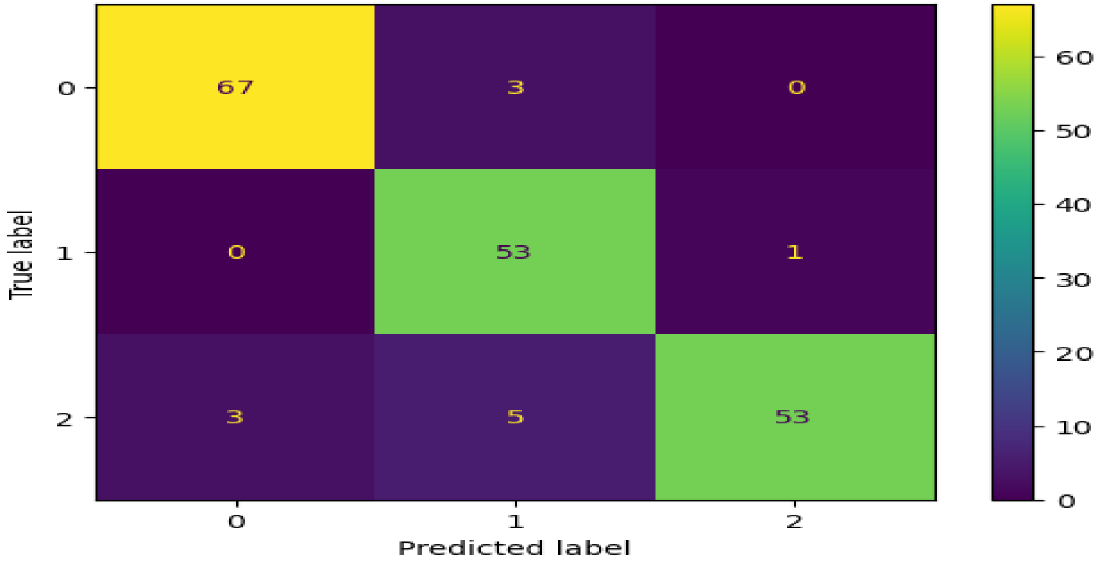
```
              precision    recall  f1-score   support

          -1       0.96      0.96      0.96        70
           0       0.87      0.98      0.92        54
           1       0.98      0.87      0.92        61

    accuracy                           0.94       185
   macro avg       0.94      0.94      0.93       185
weighted avg       0.94      0.94      0.94       185
```

- Classification Report for testing data:

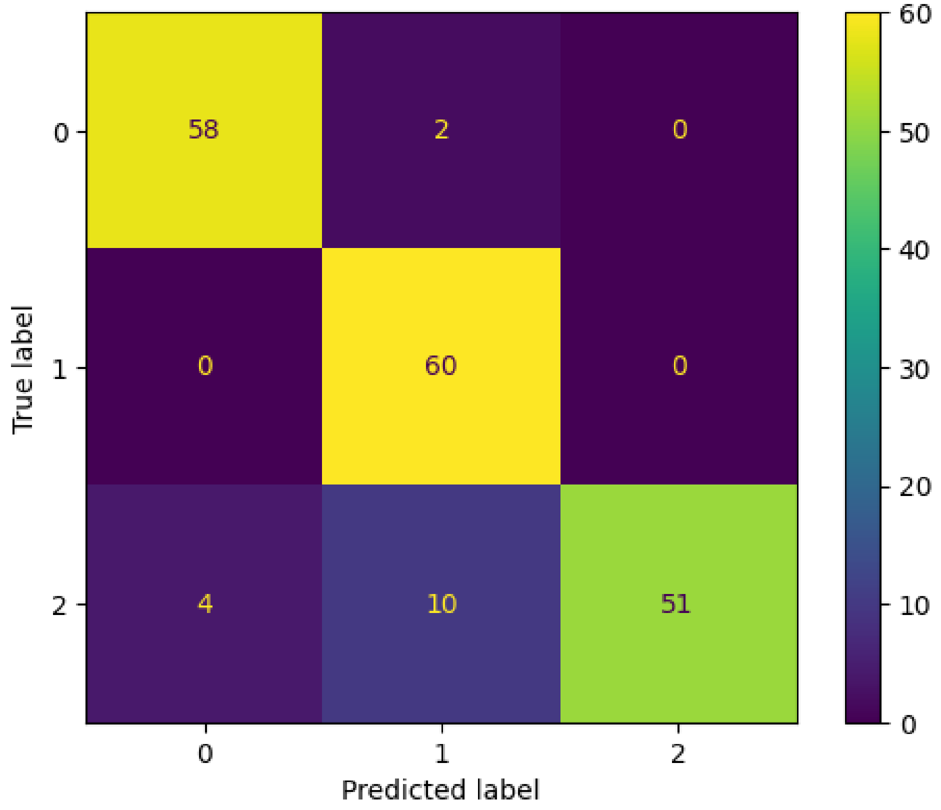


```
              precision    recall  f1-score   support

          -1       0.94      0.97      0.95        60
           0       0.83      1.00      0.91        60
           1       1.00      0.78      0.88        65

    accuracy                           0.91       185
   macro avg       0.92      0.92      0.91       185
weighted avg       0.93      0.91      0.91       185
```

- Confusion Matrix for validation data



- **Confusion Matrix** for testing



data

-

4. Advanced Experiments

- **Goal:** To improve sentiment classification performance by exploring deep learning approaches using **LSTM** networks.

- **Tokenization**: Converted texts to tokens using a tokenizer from **TensorFlow**.

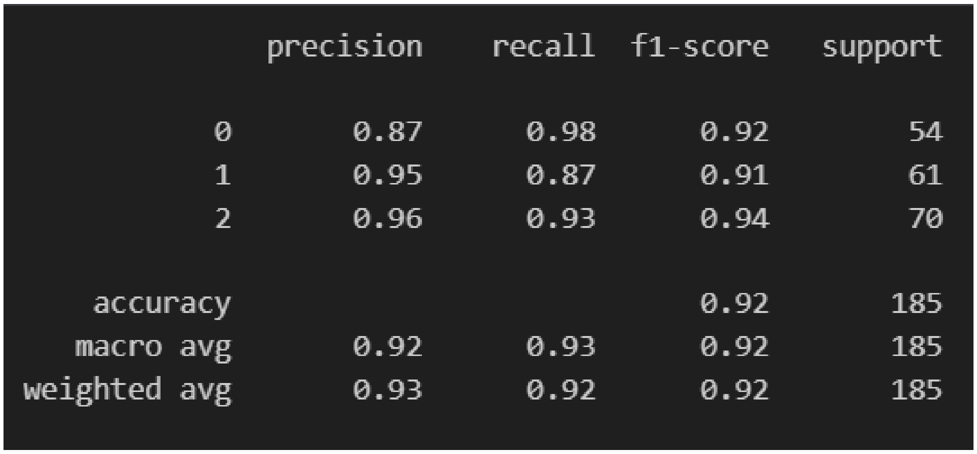
- **Padding**: Ensured all sequences have the same length by padding the sequences from **TensorFlow**.

- **Model Architecture**: Implemented an LSTM model with an Embedding layer, Bidirectional layer, Dropout layer, three Dense layers, Flatten layer, and Final Dense layer.

- **Training and Evaluation:** Trained the LSTM model on the training data with **20 epochs** with Adam optimizer, categorical crossentropy loss and evaluated its performance on the validation data.

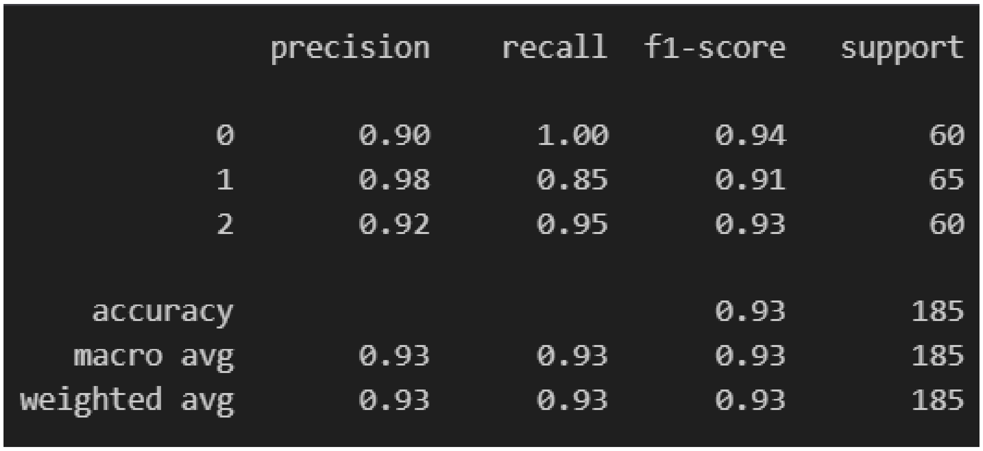
- **Conclusions**

-**Classification Report** for validation data:



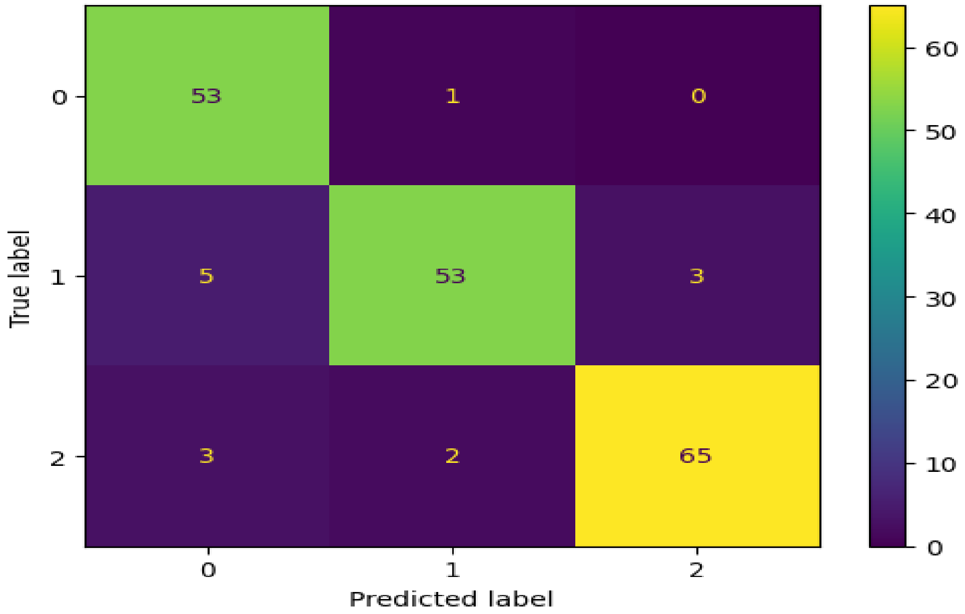
```
              precision    recall  f1-score   support

           0       0.87      0.98      0.92        54
           1       0.95      0.87      0.91        61
           2       0.96      0.93      0.94        70

    accuracy                           0.92       185
   macro avg       0.92      0.93      0.92       185
weighted avg       0.93      0.92      0.92       185
```

- **Classification Report** for testing data:



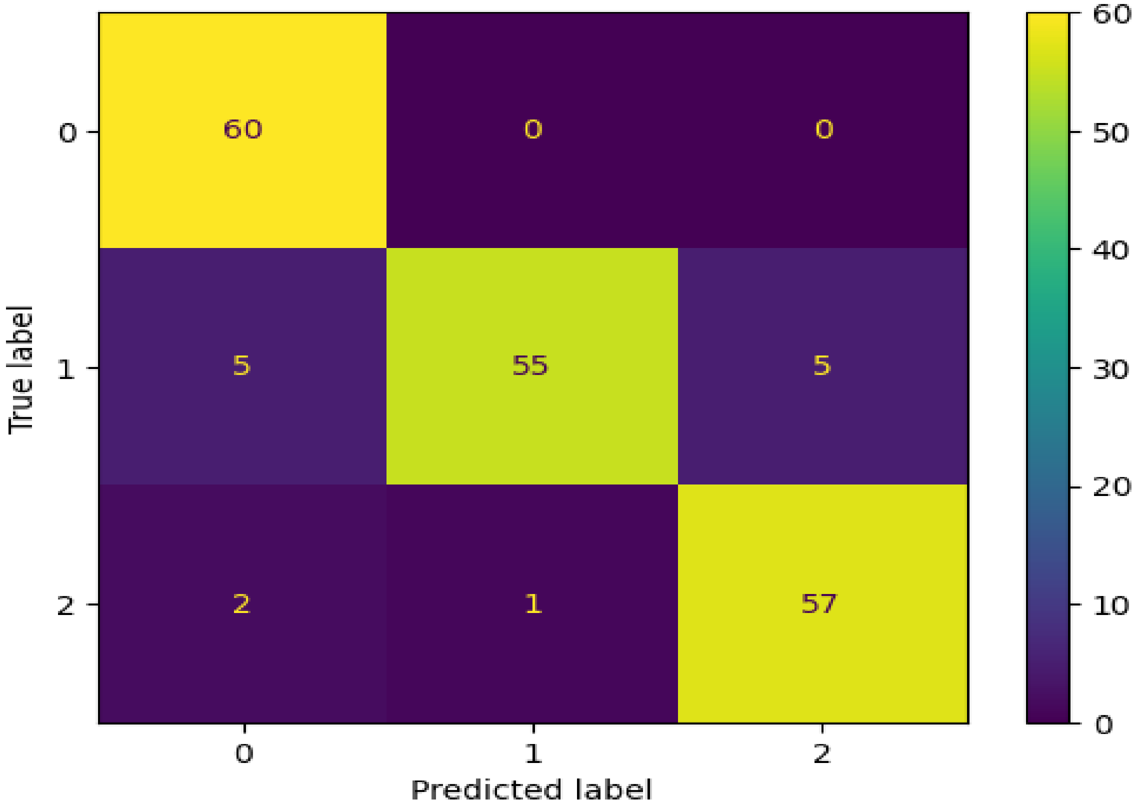
```
              precision    recall  f1-score   support

           0       0.90      1.00      0.94        60
           1       0.98      0.85      0.91        65
           2       0.92      0.95      0.93        60

    accuracy                           0.93       185
   macro avg       0.93      0.93      0.93       185
weighted avg       0.93      0.93      0.93       185
```

-**Confusion Matrix** for validation data;



-

-Confusion Matrix for testing data;



5. Overall Conclusion

This sentiment analysis project has pipeline machine learning and deep learning model for categorizing social media sentiments.

Our baseline model, which used TF-IDF vectorization and a Multinomial Naive Bayes classifier, showed good results with an accuracy of 94% on the validation data and 91%.

The deep learning model utilizing LSTM displayed a slight enhancement, reaching 92% accuracy on the validation dataset and 93% on the testing dataset.

6. Tools and Libraries

- Python
- Pandas, NumPy

- Matplotlib, Seaborn
- Scikit-learn
- NLTK
- TensorFlow/Keras
- Imbalanced-learn
- Joblib
- Re

7. External Resources: Kaggle

8. What was the biggest challenge you faced in implementing sentiment analysis?

Mapping diverse emotional labels to three broad categories (positive, negative, neutral)

9. What insights did you gain about NLP and sentiment analysis through this project?

Comparison between Traditional Machine Learning and Deep Learning Model to understand their strengths and weaknesses.