Project Report: Mental Health Sentiment Analysis

1. Introduction

In recent years, mental health has gained increased attention due to its significance in overall well-being. Social media platforms, particularly Twitter, serve as a rich source of data for understanding public sentiment regarding mental health issues. This project aims to develop a sentiment analysis model that classifies tweets related to mental health into four categories: positive, negative, neutral, and irrelevant.

2. Dataset

The dataset used for this project is sourced from Kaggle and contains tweets labeled with sentiments related to mental health. The dataset comprises the following classes:

Positive: Tweets expressing positive sentiments about mental health.

Negative: Tweets expressing negative sentiments.

Neutral: Tweets that do not convey strong feelings or opinions.

Irrelevant: Tweets that do not relate to mental health.

3. Methodology

3.1 Data Preprocessing

The data preprocessing steps include:

Data Loading: Loading the training and testing datasets.

Text Cleaning: Removing URLs, user mentions, hashtags, and special characters from the tweets. This is achieved using regular expressions.

Encoding Labels: Converting sentiment labels into numerical format using LabelEncoder.

Class Weight Calculation: Computing class weights to handle class imbalance.

3.2 Tokenization and Padding

Tokenization: A Tokenizer is used to convert the cleaned tweets into sequences of integers.

Padding: The sequences are padded to ensure uniform length for input into the LSTM model.

3.3 Model Architecture

A Bidirectional LSTM model is utilized for sentiment analysis due to its ability to capture context from both directions of the text. The architecture includes:

Embedding Layer: Transforms input integers into dense vectors of fixed size.

Bidirectional LSTM Layer: Learns dependencies in both forward and backward directions.

Dropout Layers: Prevent overfitting by randomly setting a fraction of input units to 0 during training.

Dense Layer: A fully connected layer to output the sentiment classes.

3.4 Model Training

The model is trained using:

Loss Function: Sparse categorical crossentropy.

Optimizer: Adam optimizer with a learning rate of 0.001.

Early Stopping: To prevent overfitting, training is stopped if the validation loss does not improve for 5 consecutive epochs.

4. Results

4.1 Model Performance

The performance metrics for the Bidirectional LSTM model are as follows:

Test Accuracy: 96.20%

Test Loss: [Include test loss value here]

4.2 Comparison with Other Models

Two models were trained during this project:

1. Random Forest

Accuracy: 94.6%

2. Bidirectional LSTM

Accuracy: 96.20%

The Bidirectional LSTM outperformed the Random Forest model, indicating better capability in understanding the nuances of the sentiment in the tweets.

5. Challenges Faced

5.1 Class Imbalance

One of the major challenges encountered during the project was dealing with class imbalance in the dataset. Certain sentiment classes, particularly the irrelevant and positive categories, had significantly fewer samples compared to the neutral and negative classes. This imbalance could lead to the model being biased towards the majority classes, reducing its ability to accurately predict minority classes.

5.2 Class Weight Adjustment

To mitigate the class imbalance, class weights were calculated using sklearn’s class\_weight module. These weights assigned higher importance to the minority classes to ensure the model paid more attention to them during training. However, adjusting the class weights introduced additional challenges:

Overfitting: The model occasionally overfitted the minority classes during early epochs, which resulted in reduced overall accuracy during validation. To overcome this, early stopping was employed to halt training when validation loss stopped improving.

Instability in Loss Function: The loss function was more volatile when incorporating class weights, particularly in the initial stages of training. This required a fine-tuning of hyperparameters such as the learning rate and batch size to stabilize the model.

Difficulty in Optimizing: Balancing the effect of class weights without disproportionately affecting the accuracy of majority classes was tricky. It required iterative experimentation to find the optimal configuration.

Despite these challenges, incorporating class weights led to a more balanced performance across all sentiment classes, improving the model’s overall ability to detect minority class sentiments like positive and irrelevant.

6. Web Application

A web application was developed using Streamlit to allow users to input text and receive sentiment predictions. The application consists of:

User Interface: Simple interface prompting users to enter their message.

Prediction Functionality: Takes the input message, processes it, and predicts the sentiment using the trained model.

6.1 User Interaction

Users can type a message, click submit, and receive feedback on their mood based on the sentiment detected in their input.

7. Conclusion

This project successfully demonstrates the potential of machine learning techniques in understanding public sentiment towards mental health on social media. The Bidirectional LSTM model achieved a high accuracy of 96.20%, making it suitable for deployment in real-world applications to aid in mental health monitoring and support.

8. Future Work

Enhanced Dataset: Incorporating more diverse datasets to improve model robustness.

Real-time Monitoring: Developing a system to analyze sentiments in real time on social media platforms.

User Feedback Mechanism: Allowing users to provide feedback on predictions to continuously improve the model.