Approach and Methodology

Objective

The goal of this project was to develop a text classification model that identifies emotions such as sadness, joy, anxiety, and stress based on input text. This model is integrated with a chatbot for generating context-aware responses.

Data Preparation Techniques

Dataset Cleaning: Removed NaN values, punctuation, and stopwords using the NLTK library. Text Tokenization: Tokenized text into words while converting them to lowercase. Data Split: The dataset was split into training (80%) and validation (20%) sets, ensuring balanced representation for stratified labels. Label Encoding: Emotion labels were mapped to numeric values for model compatibility.

Model Choices

Used the BERT-base-uncased pre-trained transformer model for sequence classification. Leveraged Hugging Face’s Transformers library for fine-tuning the BERT model on emotion classification tasks. Implemented the Trainer API to simplify training, evaluation, and saving of the model.

Pipeline and Implementation

Preprocessed text was tokenized using BERT’s tokenizer, and encodings were fed into the BERT model. Fine-tuned the model on labeled emotion data for one epoch with a learning rate of 2e-5. Used the Trainer class for training and evaluation. Developed a sentiment-analysis pipeline for inference and response generation based on predicted emotions.

Challenges Faced

Data Preprocessing:

Handling noisy text required careful preprocessing to preserve meaningful content while removing unnecessary details like punctuation and stopwords.

Class Imbalance:

Some emotions had fewer samples, making it harder for the model to generalize well for underrepresented classes.

Model Complexity and Computational Resources:

With a dataset of over 5000 rows, training the BERT model was computationally demanding. Fine-tuning required a balance between batch size and epochs to optimize performance without overloading available resources.

Solution:

The batch size was adjusted to fit the available hardware, and training was limited to a single epoch to mitigate resource constraints. Future work could explore using lighter models such as DistilBERT for faster processing.

Overfitting:

Despite the large dataset, there was still the risk of overfitting, especially with such a powerful model as BERT. If the model overfits, it may perform well on the training data but fail to generalize to unseen text.

Solution:

Regularization techniques, such as weight decay and dropout, were incorporated during training. Cross-validation could be implemented in the future to further monitor and prevent overfitting.

Cultural Nuances in Emotion Recognition:

The model’s ability to understand cultural sensitivities in emotional expressions was limited by the dataset’s scope. Emotions are expressed differently across cultures, and the dataset may not have fully captured these variations, potentially affecting the model’s accuracy in diverse contexts.

Solution:

To address this, the dataset could be expanded to include text from a variety of linguistic and cultural backgrounds. Multilingual models or fine-tuning on cross-cultural text data could also improve the model’s robustness to cultural differences.

Results

The model achieved strong performance on the validation set, with high classification accuracy and recall scores across emotion categories. The classification report generated during evaluation provided insights into the model's effectiveness.

Example output from the classification report:

Classification report

```
                  precision    recall  f1-score   support

          Anxiety       0.84      0.79      0.82       768
          Bipolar       0.82      0.77      0.79       556
       Depression       0.76      0.74      0.75      3081
           Normal       0.93      0.94      0.93      3269
Personality disorder    0.72      0.60      0.65       215
           Stress       0.65      0.65      0.65       517
         Suicidal       0.69      0.74      0.71      2131

         accuracy                           0.80     10537
        macro avg       0.77      0.75      0.76     10537
     weighted avg       0.80      0.80      0.80     10537
```

Integration with the chatbot provided emotion-based responses such as:

Input: "I feel so overwhelmed with everything happening." Chatbot: I'm here to help with anything you're feeling.

Reflection and Future Improvements

Addressing Model Complexity and Computational Resources:

While the large dataset of 5000+ rows provided a good foundation for training, BERT's complexity can still pose challenges in terms of resource usage. Future work could explore alternative models like DistilBERT or ALBERT for faster training with less computational overhead. Additionally, mixed-precision training and gradient accumulation techniques could be applied to optimize resource usage during fine-tuning.

Preventing Overfitting:

Despite having a large dataset, overfitting remains a concern. To mitigate this, data augmentation techniques, such as paraphrasing or adding noise, could be applied to further enhance the dataset. Cross-validation and model ensemble methods would also help ensure that the model generalizes well across different data splits.

Improving Understanding of Cultural Nuances:

To enhance the model’s understanding of cultural differences in emotional expression, future iterations could include text from more diverse linguistic and cultural sources. Fine-tuning BERT on multilingual datasets or incorporating region-specific emotion lexicons would improve the model's accuracy in recognizing emotion across cultures. Additionally, sentiment-aware word embeddings or multilingual models could help capture emotional expression more accurately across different languages.

Ethical Considerations and Bias Mitigation:

The model’s ability to handle sensitive emotional data responsibly is essential. Future versions should include mechanisms to detect and mitigate biases related to gender, race, or other social factors. Ensuring the chatbot's responses are empathetic and free from harmful stereotypes should be a priority in the next phase of developmen

Model Deployment and User Interaction

To deploy the trained model, we used Streamlit, a framework that allows for quick deployment of machine learning models through interactive web interfaces. Streamlit enables users to input text and receive responses based on the emotion detected in their input.

Streamlit Integration:

The model was integrated into a Streamlit app, where users can chat with the bot and receive emotion-aware responses. The app takes user input, predicts the emotion using the trained model, and provides a corresponding response from a predefined set of emotion-related responses. Model Loading: The trained model and tokenizer were loaded from the directory where the model was saved (in this case, r"D:\CODES\Projects\Github-Projects\sentiment\trained_model").

```
python

model = AutoModelForSequenceClassification.from_pretrained(MODEL_PATH)
tokenizer = AutoTokenizer.from_pretrained(MODEL_PATH)
```

Emotion Prediction: The predict_emotion function processes the user's input, passes it through the model, and maps the output to an emotion label.

```
python

def predict_emotion(text):
    encodings = tokenizer([text], truncation=True, padding=True, max_length=128,
return_tensors="pt")
    logits = model(**encodings).logits
    predicted_label = torch.argmax(logits, axis=1).item()
    return predicted_label
```

Emotion-Response Mapping: Based on the predicted emotion, the chatbot responds with empathetic messages tailored to the detected emotional state:

```

emotion_responses = {
    0: "I'm here for you. Want to talk about it?",  # Anxiety
    1: "Take a deep breath. What's on your mind?",  # Normal
    2: "I'm sorry to hear you're feeling this way. It's okay to seek help. I'm here to
listen.",  # Depression
    3: "I'm really concerned. Please talk to someone close to you or seek professional
help.",  # Suicidal
    4: "It sounds like you're under a lot of stress. Take a moment to breathe. I'm
here for you.",  # Stress
    5: "It can be overwhelming dealing with bipolar mood swings. You're not alone in
this.",  # Bipolar
    6: "Personality disorders can be really challenging. Consider talking to a mental
health professional.",  # Personality disorder
}
```

Streamlit Interface: The interactive Streamlit interface allows users to input text, triggering the chatbot's response based on the detected emotion:

```

user_input = st.text_input("Type... ", "")
if st.button("Send"):
    if user_input.strip():
        emotion_label = predict_emotion(user_input)
        emotion_name = label_map[emotion_label]  # Map the label to its name
        response = emotion_responses.get(emotion_label, "How can I assist you?")
        st.write(f"**Detected Emotion:** {emotion_name}")
        st.write(f"**Bot:** {response}")
    else:
        st.write("**Bot:** Please say something!")
```

Example:

```
Emotional Chatbot
Chat with me, and I'll respond to your emotions.

Type...

i am worried about anything

Detected Emotion: Anxiety

Bot: I'm here for you. Want to talk about it?
```