**1. Environment and Tools**

* **Platform**: Google Colab
* **Language**: Python
* **Libraries Used**:
  + pandas for data loading and manipulation
  + transformers from Hugging Face for inference with pre-trained NLP models
  + nltk for text preprocessing (stopword removal and stemming)
  + re for regex-based text cleaning

**2. Objective**

To perform emotion classification on user messages using a fine-tuned BERT model, and annotate each message with both a predicted emotion label and an associated emoji. This task includes:

* Classifying single text inputs
* Running inference on a full dataset
* Exporting annotated results to a new CSV

**3. Model Used**

* **Model Name**: boltuix/bert-emotion
* **Type**: Fine-tuned BERT model for multi-class emotion classification
* **Hugging Face Pipeline Task**: text-classification

**4. Emotion-to-Emoji Mapping**

A dictionary was defined to map emotion labels to corresponding emojis. This aids in visual representation of emotion, though it is not essential for model inference.

Sample mappings:

* Sadness → 😢
* Anger → 😠
* Love → ❤️
* Happiness → 😄
* Neutral → 😐
* Sarcasm → 😏

**5. Text Classification Example (Single Input)**

* Input Text: "i am proud of you!"
* The model predicted the emotion label with associated confidence score.
* Result was printed with emotion label and mapped emoji.

**6. Dataset-Based Emotion Detection**

* **Input Dataset**: emo\_data.csv
* **Initial Checks**: Loaded and reviewed dataset using .head(), .tail(), and .columns.

**6.1 Text Cleaning and Preprocessing**

* Removed all non-letter characters using regex.
* Converted text to lowercase.
* Tokenized words and removed common stopwords (except not).
* Applied stemming using PorterStemmer.
* Reconstructed cleaned messages into a new list corpus.

**6.2 Inference on Full Dataset**

* Ran emotion classification pipeline on the preprocessed corpus.
* Extracted predicted emotion labels for each message.
* Mapped labels to emojis using the previously defined dictionary.

**6.3 Output Columns Added**

* predicted\_emotion: Predicted label (capitalized)

**7. Export Results**

* Final annotated DataFrame was saved to:
  + emo\_data\_predictions\_with\_emoji.csv
* File includes original messages along with predicted emotion and emoji.

**8. Conclusion**

This project demonstrates the use of a fine-tuned BERT model for emotion classification in text. It supports both:

* Real-time emotion prediction for individual inputs
* Batch emotion analysis on a dataset

The structured output (CSV) can be used for:

* Emotion-based content analysis
* Chatbot response tuning
* Customer sentiment monitoring