Transformers

كلية الحاسبات والذكاء الاصطناعي
Faculty of computers & Artificial Intelligence

Team members:

1- Abdallah Ahmed Abdallah
2- Abdallah Hesham Gomaa
3- Abdelrhman Mohamed Talat
4- Abdelrahman Karam Mahmoud
5- Abdelfatah Ahmed Abdelfatah
6- Omar Ramadan Nabeh

Under Supervision of:

Eng/ Mariam Mahmoud

Mental Health Prediction

Problem Statement

Mental health disorders are a global concern affecting all age groups. Despite growing awareness, they remain underprioritized in healthcare, education, and workplaces. Conditions like depression, anxiety, and stress harm emotional well-being, cognitive function, and quality of life, leading to lower productivity, strained relationships, and higher healthcare costs. Stigma, limited access to services, and poor public education worsen the problem, leaving many untreated. Addressing mental health is both a humanitarian and socioeconomic necessity.

Dataset and Preprocessing

About this Dataset

This comprehensive dataset is a meticulously curated collection of mental health statuses tagged from various statements. The dataset amalgamates raw data from multiple sources, cleaned and compiled to create a robust resource for developing chatbots and performing sentiment analysis.

Data Overview

The dataset consists of statements tagged with one of the following seven mental health statuses:

- **Normal**
- **Depression**
- **Suicidal**
- **Anxiety**
- **Stress**
- **Bi-Polar**
- **Personality Disorder**

Data Preprocessing

To ensure clean and consistent text input for downstream natural language processing (NLP) tasks, a series of preprocessing steps are applied to the raw data. These functions are responsible for transforming noisy or unstructured text into a form that is more suitable for machine learning models. Below is a detailed overview of the preprocessing pipeline:

1. Tag Detection and Removal

- `has_tags(text)`: Checks if the input text contains any HTML tags using a regular expression.
- `remove_tags(text):` Strips away HTML tags from the text, leaving only the textual content.

2. Tokenization and Reassembly

- `tokenize(text):` Splits the input string into individual word tokens using NLTK's word_tokenize.
- `again(text):` Reverses tokenization by joining tokens back into a single string.

3. Text Cleaning

- `remove_punc(text):` Removes standard punctuation characters from a list of tokens.
- `remove_ex(text):` Eliminates non-ASCII characters (e.g., emojis, special symbols) from the text.
- `remove_links(text):` Removes URLs using a regex pattern to filter out web links.
- `remove_stop(text):` Filters out common English stopwords (e.g., "the", "and", "is") using NLTK's stopwords list.

4. Normalization

- `stemmer(text):` Applies stemming using NLTK's PorterStemmer to reduce words to their base form (e.g., "running" → "run").
- `lemmatizer(text):` Uses NLTK's WordNetLemmatizer to convert words to their root form based on context (e.g., "better" → "good").

5. Feature Extraction

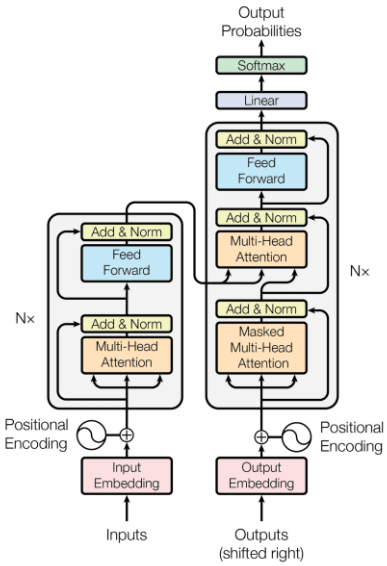
- *count_punc(text):* Calculates the percentage of punctuation characters in the text (excluding spaces), which can be useful for stylistic or sentiment features.

6. Tokenizer Integration for Model Input

- *tokenize_function(examples):* Prepares the text for input into a transformer model by applying a tokenizer (e.g., BERT tokenizer) with padding and truncation, ensuring each input is of fixed length (max 128 tokens).

Model Selection

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art language representation model developed by Google in 2018. It is designed to understand the context of a word based on all of its surrounding words, thanks to its **bidirectional** architecture. Unlike traditional models that read text left-to-right or right-to-left, BERT reads in both directions simultaneously, which allows it to capture deep contextual meaning.

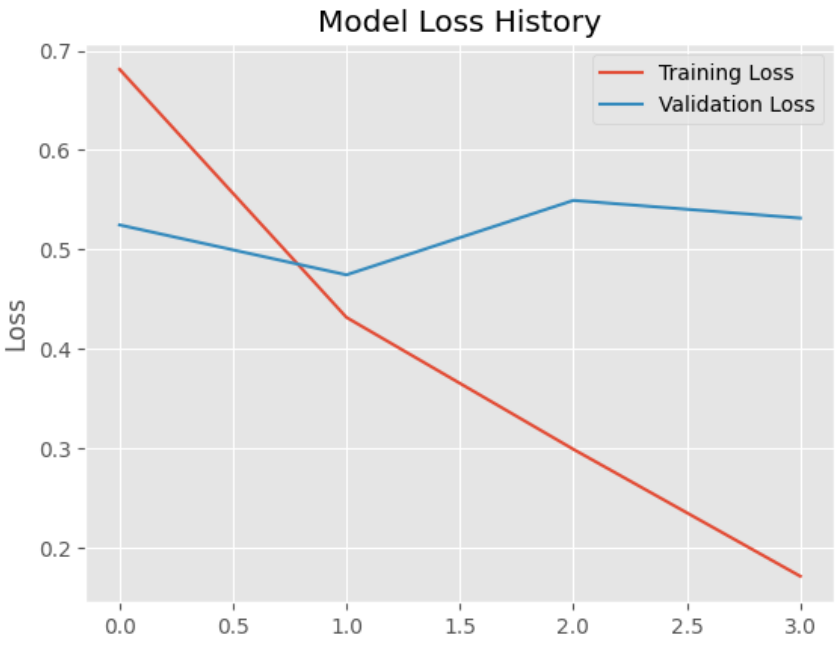


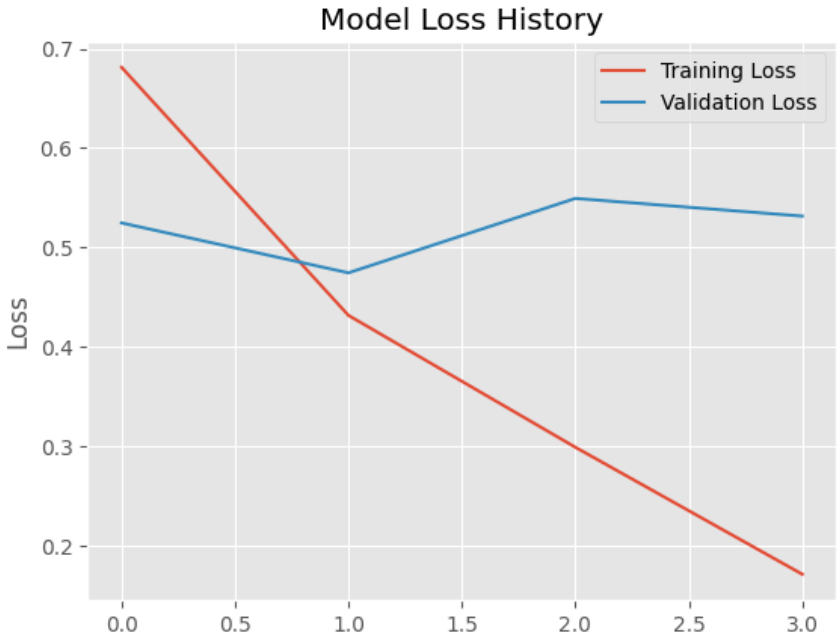
Implementation details

We implemented a text classification model using **TensorFlow** and the **Hugging Face Transformers** library. The input data was split into training and validation sets, and the labels were encoded numerically. We used the **BertTokenizer** from the **bert-base-uncased** model to tokenize the text statements with a maximum sequence length of 128. The tokenized inputs were transformed into TensorFlow **tf.data.Dataset** objects with appropriate batching and shuffling. The classification model was built using **TFBertForSequenceClassification** from the Transformers

library, configured for 6 output classes. The model was compiled in **TensorFlow** with the **Adam** optimizer, **sparse categorical cross-entropy loss** (from logits), and accuracy as the evaluation metric. Training was performed for up to 3 epochs with early stopping enabled to monitor validation loss and prevent overfitting.

Results and analysis





Transformers

Class	Precision	Recall	F1-Score	Support
Anxiety	0.92	0.85	0.88	1458
Bipolar	0.88	0.87	0.87	1046
Depression	0.75	0.7	0.73	3012
Normal	0.94	0.93	0.94	3145
Stress	0.78	0.77	0.78	950
Suicidal	0.66	0.77	0.71	2121
accuracy			0.82	11724
macro avg	0.82	0.82	0.82	11724
weighted avg	0.82	0.82	0.82	11724