Week 2 Report

Text Classification on Toxic Content Dataset

1. Class Imbalance Handling and Experiment Results

To address the issue of class imbalance in the toxic content dataset, a **Convolutional Neural Network (CNN)** model was trained using **class weighting**. This technique dynamically assigns higher weights to underrepresented classes, ensuring that the model pays more attention to rare but important categories during training.

As a result of applying class weights:

- The model became less **biased** toward the majority classes.
- **Sensitivity** and **precision** improved for rare toxicity types.

This approach significantly enhanced the model's generalization across all classes, particularly its ability to accurately detect less frequent but critical categories of toxic content.

2. Integration of LLaMA Guard API and BLIP

LLaMA Guard Integration:

Due to technical limitations in downloading and deploying the official **LLaMA Guard** model locally, an alternative approach was adopted using **LLaMA 3 via the Groq API**. To simulate the moderation functionality of LLaMA Guard, the LLaMA 3 model was provided with a structured system prompt that defines its role and expected output. The prompt used is as follows:

The LLaMA 3 model is prompted to:

"You are LLaMA Guard, a content moderation model."
"Classify the following text as either 'safe' or 'unsafe'. "
"Only respond with one word: 'safe' or 'unsafe'."

BLIP Integration

To support image moderation, the **BLIP (Bootstrapped Language-Image Pretraining)** model was integrated using the Salesforce/blip-image-captioning-base checkpoint from Hugging Face. Upon uploading an image, it is processed by the BLIP model to generate a **natural language caption** describing the image content. This caption is then evaluated using the LLaMA-based moderation pipeline described above. This two-stage pipeline ensures consistent moderation for both text and image inputs, leveraging vision-language capabilities to extract meaning from visual content and apply safety filtering accordingly.

3. **Dual-Stage Moderation Logic in Streamlit**
   The Streamlit app applies a **two-stage moderation pipeline** to ensure safety for both text and image inputs:

   Stage 1: Safety Classification via LLaMA Guard (Groq API)

   - All inputs (text or image captions) are first passed to LLaMA 3, prompted to behave as LLaMA Guard.
   - The model returns a strict one-word classification: "safe" or "unsafe".
   - If the content is marked unsafe, processing stops, and a warning is displayed.

   Stage 2: Toxic Content Categorization (CNN Model)

   - If the input passes the safety check, it proceeds to a CNN classifier trained to detect specific toxic content categories (e.g., violent crime, child exploitation).
   - This provides more granular insight into the nature of the content if it is deemed safe.

   For images, the **BLIP model** generates a descriptive caption, which is then passed through the same moderation logic, treating visual content as text for unified safety handling.