**Project 03: Fine-Tuning Transformer Architectures for Real-Time NLP Applications**

**(Group Assignment: 2 Persons)**

* **Objective:**

This project focuses on applying three distinct Transformer architectures (**Encoder-only, Decoder-only, and Encoder–Decoder**) to real-time NLP tasks. You need to fine-tune pre-trained models from the Hugging Face library for classification, code generation, and summarization, showcasing their comparative strengths.

A **Streamlit/Gradio** interface will be developed for real-time interaction, and results will be summarized in a **medium blog** post.

* **Submission Instructions**
* You can take help from **AI tools**, but **you are not allowed to use any AI tool for entire code generation**. *(If anyone is found submitting completely AI-generated code, the result will be a* ***direct 0****.)*
* This is a **15 absolutes assignment**, just like the previous one.
* **Group of Maximum 2 students** are allowed.
* **No Extension in Deadline** will be granted under any circumstances.
* **Deadline:**
* **Tasks**
* **Task 1: Encoder-Only (BERT) — Customer Feedback Classification**

**Problem:** Perform sentiment classification on customer feedback to determine whether the sentiment is positive, negative, or neutral.

**Dataset:** <https://www.kaggle.com/datasets/vishweshsalodkar/customer-feedback-dataset?select=sentiment-analysis.csv>

**Objective:** Fine-tune a BERT-based model to classify textual feedback according to customer sentiment.

**Deliverables:**

* Preprocessing and tokenization script
* Training and validation pipeline.
* Evaluation metrics (accuracy, F1-score, confusion matrix).
* Example predictions.
* **Task 2: Decoder-Only (GPT-2 or LLaMA) —Pseudo-code to Code Generation**

**Problem:** Generate executable Python code from structured pseudo-code instructions.

**Dataset:** <https://github.com/sumith1896/spoc>.

**Research Paper:** [**https://arxiv.org/pdf/1906.04908**](https://arxiv.org/pdf/1906.04908)

**Objective:** Fine-tune decoder-only models to translate pseudo-code into working code. The system should generate syntactically and semantically valid Python code.

**Deliverables:**

* Preprocessing of pseudo-code to code pairs
* Tokenization and formatting of training data
* Fine-tuning on GPT-2 (causal LM objective)
* Evaluation metrics: **BLEU**, **CodeBLEU**, and **Human Evaluation**
* Streamlit/Gradio interface for real-time pseudo-code to code generation
* **Task 3: Encoder–Decoder (T5 / BART) — Text Summarization**

**Problem:** Generate concise summaries of long documents or articles using an Encoder–Decoder architecture.

**Dataset:** <https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail>

**Objective:** Fine-tune an Encoder–Decoder model to perform **abstractive text summarization** of input passages.

**Deliverables:**

* Dataset preprocessing (article to summary pairs)
* Model fine-tuning code using Hugging Face Transformers
* Evaluation metrics: **ROUGE-1**, **ROUGE-2**, **ROUGE-L**, and qualitative comparison
* Example outputs with original text and generated summary
* **Overall Deliverables:**

1. Complete **GitHub repository** with full code and README.
2. **Deployed demo link** (Streamlit or Gradio).
3. **Evaluation report** including metrics and qualitative results.
4. **Medium blog post** describing dataset, model, metrics, and results.