**Applied LLM Engineer Assignment: Table Generation from Natural Language Input**

## Objective:

This assignment aims to implement a language model (LLM) that can take natural language input and generate corresponding tabular data. The model will be expected to interpret natural language descriptions or instructions and produce well-structured tables.

## Dataset:

https://drive.google.com/file/d/1w3SoPPU8ItB8bdy0xBNmq1AOs2-J8ZYG/view?usp=sharing

You are provided with 10 tabular datasets. All_datasets_metadata.xlsx file contains the general description of the 10 datasets.

- Data source
- Data usage

within each dataset's folder, there is a file called column_info.csv, which contains the following information

- Description for each column
- Data type for each column

Folder "50K_Songs_Dataset_-_Generated_by_AI" misses the column_info.csv file.

## Instructions:

1. **Pretrained LLM Selection:**
    - You may choose a pretrained LLM you are most familiar with (e.g., GPT, Llama, deepseek, etc.).
    - Fine-tune the LLM to generate tabular data based on natural language input.
2. **Task:**
    - Your model should generate tabular data corresponding to the input natural language query. This includes generating column names and data based on the descriptions provided in the datasets.
3. **Evaluation of Synthetic Data Quality:**
    - Propose a method to evaluate the quality of the synthetic tabular data generated by the model. This could include aspects such as:
        - Consistency with the original data distribution
        - Fidelity to the data types
        - Validity and completeness of the generated tables
        - Statistical measures (e.g., similarity metrics, data consistency checks)
4. **Optimization:**
    - Consider strategies to optimize the model's fine-tuning and inference processes. This can include:

        - Techniques for efficient fine-tuning of pretrained models
        - Reducing inference time for generating tabular data
        - Any other method to improve the overall system performance.

5. **Reporting:**
    - In your final report, you need to include:
        - **Model Fine-Tuning Time:** Report the time it took to fine-tune the model on the provided datasets.
        - **Inference Time:** Measure and report the time it takes for the model to generate a table based on a given query.
        - **Synthetic Data Quality:** Discuss your chosen evaluation metrics and how the synthetic data compares to the original datasets.

## Deliverables:

- **Code Implementation:** Include all scripts and code used for fine-tuning and generating the synthetic tables.
- **Evaluation Results:** Provide the evaluation results for the generated synthetic data.
- **Final Report:** A comprehensive report that includes your approach, optimization strategies, time measurements, and a discussion of the results.
    - Brownie points for sharing something with us that we wouldn’t know!

## Submission Guidelines:

- Submit all code and the final report in a well-organized format.
- Provide clear instructions on how to run your code.

## Notes:

The user will only prompt the model, for instance, here is an example prompt:

"You are tasked with generating a synthetic dataset based on the following description. The dataset should include the following columns:

1. **Name** (String): A person's name. The name should reflect common nam=es in Singapore, including both English and Chinese names.
2. **Age** (Integer): The age of the individual. It should be a random number between 18 and 60.
3. **Gender** (String): The gender of the individual. The possible values are 'Male' or 'Female'.
4. **Location** (String): The geographical region within Singapore where the individual resides. The possible values are 'Central', 'East', 'West', 'North', 'South'.
5. **Income** (Float): The monthly income of the individual. The income should range from $3000 to $8000, rounded to two decimal places.
6. **Occupation** (String): The occupation of the individual. The possible values are 'Engineer', 'Teacher', 'Doctor', 'Artist', 'Entrepreneur', or 'Nurse'.

"

So for your example, of course you may not use the same column name and type, but this information should be given as a prompt.

The number of rows may also need to be given within the prompt if your fine-tuning can take this parameter into account. But if you fix it, say 100 rows, it is also ok. And what we expect is to output a csv file or the csv file format data so that we can export it into a csv file. If you can wrap it into a table view to visualize, it can be even better.

And yes, we expect you to fine-tune one model that can synthesize all datasets (in the Google Drive link above).

Regarding GPUs,use Google colab to do your assignment. We do not need you to work on a super big model. A small and quantized LLM is enough. What really matters is your thinking process and the optimization tricks that you considered.  If your resource is limited, fine-tune on less datasets is also ok.