You have the option to choose any one of the assignments (Porject1/Project2) and Project 3

Project 1: Q&A Chatbot with Vector Store

Project Overview:
In this project, you will be tasked with creating a Q&A chatbot (**Basic RAG**) that utilizes a vector store and Language Model (LLM) to provide accurate responses to user queries. You will be provided with the necessary resources to get started, including bootstrap code, input files, and a demo video showcasing the working of the chatbot. Your main objectives will be to create a vector store from provided Ubuntu documentation in markdown format and integrate it with the chatbot system.

Main Tasks(100 marks):
Part 1: Create Vector Store (Any local Vector store would work)
Your first task is to create a vector store using the provided Ubuntu documentation in markdown format. The vector store will serve as the knowledge base for the chatbot. **(Preferably Use Faiss or ChromaDB), Use any OS Embeddings models from HF.**

Part 2: Create Chatbot using Vector Store and LLM
In the second part of the project, you will build a chatbot that utilizes the vector store created in Part 1 and a Language Model (LLM) to perform Q&A with users. The chatbot should be able to understand user queries, search for relevant answers from the vector store, and provide appropriate responses.
Use any free open-ai endpoint to do the querying of the LLM Chain

Bonus Tasks:

- Create a small **FastApi (app.py)** endpoint to locally deploy the application and record hitting the solution endpoint via **Swagger.(50 marks)**

- Deploy the solution using docker and test it via **Swagger** and record the outcome.**(50 marks)**

Requirements:

- Utilize the provided input files, and demo video to understand the workings of the chatbot.
- Implement a chatbot that can interpret user queries.
- Utilize the vector store to search for relevant answers.
- Generate appropriate responses based on the user's query.
- Implement proper error and exception handling
- Define your chucking strategy and further improvement can be done for better retrieval
- What are the best and fast ways to parse the documents

General Requirements:
- Utilize best practices in coding and documentation.
- Follow a modular and organized approach in your code implementation(scripts).
- Ensure your code is well-structured, readable, and maintainable.
- Utilize logging for different levels.

- Share your working demo video in the final deliverable
- Please ensure your repo is public in github and push in ordered manner like commit wise grouping
- Please create an .md file to post the screenshots for the outcomes

Chatbot bootstrap code:
https://drive.google.com/drive/folders/17v4J1DNukG2eyD5D24nDes4Q8W-eDrtI?usp=sharing
Chatbot working video: demo

Project 2: Text Classification Model for Ubuntu Customer Centre Inquiries

Project Overview:
The aim of this project is to develop a text classification model that can accurately categorize E Commerce Categories
This is the classification-based E-commerce text dataset for 4 categories - "Electronics", "Household", "Books" and "Clothing & Accessories", which almost cover 80% of any E-commerce website.
The dataset is in ".csv" format with two columns - the first column is the class name and the second one is the datapoint of that class.
Number of Instances: 50425
Number of classes: 4

Main Deliverables(100 marks):
At the completion of this project, you should provide the following deliverables:
Codebase: Share your well-structured and well-documented code for data preprocessing, feature engineering, model training, and evaluation. Use appropriate Python libraries and frameworks for efficient implementation.
One Entry(main.py) point script to test the model with live data shared by the user from the cli

Bonus Tasks:

- Create a small **FastApi** endpoint**(app.py)** to locally deploy the application and record hitting the solution endpoint via **Swagger.(50 marks)**

- Deploy the solution using docker and test it via **Swagger** and record the outcome.**(50 marks)**

Note:
- You are allowed to use any relevant open-source libraries or frameworks.(Please stick to BERT/ML Models for Text classification)
- Consider utilizing techniques to address challenges such as class imbalance, overfitting, or handling long or noisy texts.
- Feel free to explore and experiment with different approaches or improvements beyond the initial project requirements.

- Don’t use any any LLM to do the classification and try using **Stop Words** for quicker turn around
- if the dataset is too big, take a sample dataset but with proper representation of the actual class distribution

General Requirements:
- Utilize best practices in coding and documentation.
- Follow a modular and organized approach in your code implementation(scripts).
- Ensure your code is well-structured, readable, and maintainable.
- Utilize logging for different levels.
- Share your working demo video in the final deliverable
- **Please ensure your repo is public in github and push in ordered manner like commit wise grouping**
- **Please create an .md file to post the screenshots for the outcomes**

Ecommerce Prod Classification dataset:
https://drive.google.com/drive/folders/17v4J1DNukG2eyD5D24nDes4Q8W-eDrtI?usp=sharing

Project 3: Workflow Execution

Objective:The objective is to read the below mentioned research paper and build a workflow that has been discussed in the paper. Looking at the framework one can understand what is being discussed in the paper

Link:

[2402.10200] Chain-of-Thought Reasoning Without Prompting

General Requirements:

- You can use any application of your choice (draw.io/figma/Powerpoint)
- Markdown files would even work if you feel to make it in a report format