Documentation

**Introduction**

In this assignment, we going to pick two types of datasets:

* [Structured data](https://www.kaggle.com/datasets/kapturovalexander/football-mini/data) - Football dataset from Kaggle
* [Unstructured data](https://ncert.nic.in/textbook/pdf/lekl101.pdf) - Short story from a textbook

The assignment will be a fast api interface that ingest pdf (unstructured data) or csv data (structured data) and provides relevant response from our rag system. We have used both open source and closed source model to infer our outputs.

**Code Structure**

* **disney/main.py:** This is the fast api server and running this will start the server.
* **disney/requirements.txt:** Has list of all required libraries to be used
* **disney/dataset:** This a folder where all the dataset will be stored
* **disney/create\_dataset.py:** A simple script that create the dataset and stores in under 'dataset' directory
* **disney/modules/clean\_data.py:** Checks for the null, datatypes, duplicates and fix them. Removal of unwanted characters.
* **disney/modules/create\_llm.py:** Create local mistral llm or use chat-gpt api
* **disney/modules/embed\_data.py:** Create data embedding and returns vectorstore
* **disney/modules/ingest\_data.py:** Reads data from csv and pdf and returns dataframe and extracted data respectively
* **disney/modules/ingest\_sql.py:** Dumps data in dataframe to sql
* **disney/modules/structured\_rag.py:** Create a sql command and use them to get expected data for RAG and generated answers
* **disney/modules/unstructured\_rag.py:** Uses the context from the nearest match in vectorstore and generates the answer.

**Design choice**

* Fastapi - Fast api is robust, lightweight and easy to use and deploy framework for python
* Python: Due to its simplicity, readability, extensive library ecosystem including powerful tools which allows developers to quickly prototype and implement complex algorithms with ease
* Datasets: The football data set and the PDF data set is presentation of most used cases that can be handled for Rag in LLM
* LLM: We chose mistral 7B instruct model as it is one of the easiest model to host and also have a good quality output for this you would require a GPU to run the model as you can just call the open AI ChatGPT API.
* Embedding data: For this, we will be using BGE small embedding model as this is one of the few models that rank highest when it comes to benchmark in hugging face.
* Database: Here we are using SQLite as it's easy to use, easy to be deployed and is robust
* Recursion splitters: Even though this is a very basic splitter, and there are many complicated splitting techniques due to the simple nature of the PDF data, we are just sticking to the recursive character splitting.

**How to use**

* The project ran under python version 3.11.4, make sure you have the required python version. Most python3 versions should work here as it is a simple project.
* First let us create a virtual env, this can be done in many ways. For now we will use this:
  + cd disney
  + python -m venv venv
  + for windows run venv\Scripts\activate (Any error check for Set-ExecutionPolicy or run as administrator)
  + for linux or mac run source venv/bin/activate
* lets install all libraries simply by running pip install -r requirements.txt (Any error check of the python version, pip version or install directly from source)
* Now lets create a dataset i.e csv and pdf dataset by running create\_dataset.py
* create\_dataset.py will create a folder call dataset and drop pdf and csv files into it.
* Before we run we need api keys for hugging face token (Can be generated by logging into hugging face and then going to settings) use this if we are running the model locally else we can add open ai api key
* Now we are read to run main.py file run by using 'python main.py' if it is successful then open another terminal and run

curl -G "http://localhost:8000/" \

--data-urlencode "path=dataset" \

--data-urlencode "question=who was paid 260000usd?" \

--data-urlencode "type=structured" \

--data-urlencode "llm\_type=openai"

Sample results:

The individuals who were paid 260,000 USD are De Gea and I Rakiti

curl -G "http://localhost:8000/" \

--data-urlencode "path=dataset" \

--data-urlencode "question=What happened on the day of Pablo Neruda?" \

--data-urlencode "type=unstructured" \

--data-urlencode "llm\_type=openai"

' Sample results:

{"message":"On the day of Pablo Neruda's visit, he spent the morning hunting for big game at second-hand bookstores and purchased an old, dried-out volume at Porter for a high price."}