Data/ML Engineer Assignment

Context

The purpose of this assignment is to showcase your abilities in software, data, prompt and ML engineering, as well as your coding proficiency. You will have the opportunity to demonstrate your problem-solving skills, creativity, and experience with various tools that you choose to implement. This will help us to gain deeper insight into your technical capabilities, which will be invaluable in determining your suitability for various projects and roles within young startup.

It is important to convince your future teammates with your competences in:

- Python3
- Data structures and algorithms
- Software development principles
- ML engineering skillsets
- Prompt engineering best practices
- Containerization, CI/CD automation and maintaining a production-grade service
- An insightful, readable, well-structured *README.md*; (We don't expect extensive/fancy documentation or diagrams)

> 💡 It is completely up to you to choose which tools, frameworks, data formats, schedulers, etc. you use. However, please provide an explanation of **why** you chose a specific tool for this specific task and in the context of this assignment.

Assignment

Develop a vector database API that utilizes search through a large number of PDF documents. Additionally, containerize the entire solution using Docker-compose or Minikube or similars for continuous integration and deployment. Finally, expose the API using GRPC endpoints for efficient data uploading and retrieval.

Requirements

1. Vector Database:
    - Design and implement a vector database for efficient storage and retrieval of high-dimensional vectors.
    - Text extraction: Utilize techniques or document parsing tools to extract the textual content from the PDF documents. This process will convert the documents into text data, which can be further processed and transformed into vectors.
    - Text preprocessing: Apply text preprocessing techniques such as tokenization, stop word removal, stemming, and/or lemmatization to clean the extracted text data. This step helps in reducing noise and preparing the text for vector representation.
    - Text summarization: Implement text summarization techniques to generate short summaries of the documents, which can be used for quick document scanning and retrieval.
    - Vectorization: Utilize vectorization techniques such as word embeddings or similar, to convert the preprocessed text data into vector representations. These vectors will capture the semantic meaning and context of the textual content.
    - Vector indexing: Implement vector indexing techniques such as similarity search, cosine similarity, or other relevant algorithms to enable efficient retrieval of PDF documents based on similarity to a query vector.

- Ensure the vector search process is efficient and scalable.

2. GRPC API Development:
    - Implement a GRPC API to interact with the vector database.
    - Define appropriate API endpoints for uploading PDF documents, searching through documents, etc.
    - Define appropriate API endpoints for “Text summarization”.
    - Ensure proper error handling and input validation within the API.
3. Containerization and CI/CD Implementation:
    - Set up a CI/CD pipeline for the entire solution that automatically builds containers.
    - Configure automated testing to validate the functionality and quality of the API.

Remember to follow software engineering principles throughout the assignment, including proper code structure, adherence to SOLID principles, testing strategies, error handling, and version control practices.

Final words

We wish you the best of luck in completing this assignment and hope that you enjoy the challenge! Remember to take advantage of this opportunity to showcase your skills and creativity.

We look forward to seeing your solution and gaining deeper insight into your technical capabilities. Good luck!