Text Embedding and Similarity Analysis - Documentation

# Introduction

This project demonstrates how to retrieve text embeddings from a model (such as Llama 2) and perform similarity analysis on predefined text titles using FAISS for fast vector search. It also calculates cosine similarity, Euclidean similarity, and Pearson correlation to compare a user prompt with the stored titles.

# Installation

1. Install the required libraries:  
```bash  
pip install faiss-gpu numpy requests scikit-learn scipy  
```  
2. Ensure that the embedding server is running on `http://localhost:11434/api/embeddings`.

# Code Explanation

## 1. Importing Libraries

The necessary libraries such as FAISS, Requests, Numpy, and Scikit-learn are imported. FAISS is used for efficient similarity search, while Numpy is used to handle vector operations. The `requests` library is used for API calls to retrieve embeddings, and Scikit-learn and Scipy are used for calculating similarity metrics.

## 2. Embedding Function

The `get\_embedding` function sends a POST request to the local API server to retrieve embeddings for a given text. If the API request is successful, it returns the embedding as a Numpy array; otherwise, it handles the error gracefully.

## 3. FAISS Index Creation

An FAISS index is initialized using `faiss.IndexFlatL2(d)` where `d` is the dimensionality of the embeddings (4096). The embeddings of predefined titles are stored in a matrix `X` and added to the FAISS index.

## 4. Similarity Calculation

For a new prompt, the following similarity measures are calculated against the stored embeddings:  
- \*\*Cosine Similarity\*\*: Measures the cosine of the angle between two vectors.  
- \*\*Euclidean Similarity\*\*: Inverted Euclidean distance between vectors.  
- \*\*Pearson Correlation\*\*: Measures the linear correlation between vectors.

# Usage

1. Ensure that the embedding API is running locally.  
2. Run the script to retrieve embeddings for titles and compare a new prompt using FAISS and various similarity metrics.  
3. The script outputs the cosine similarity, Euclidean similarity, and Pearson correlation for each predefined title.