**Sahil Wani**

SUID: 466848553

Jan 15th report

**Introduction**

The objective of this week’s task was to create a vector database for storing and retrieving course information efficiently. By utilizing Pinecone for vector similarity search and sentence-transformers for generating embeddings, I successfully developed a system capable of handling complex queries. Below, I describe the step-by-step process implemented to achieve this.

**Step 1: Data Preparation**

The initial phase involved preparing a dataset of course information. The data was manually scrapped and structured as a list of dictionaries from iSchool’s course catalog website.

Link: <https://courses.syracuse.edu/content.php?catoid=39&navoid=4900&p1170=1#ent_courses1170>

Each entry included:

* course\_name: The name of the course.
* description: A detailed description containing credits, topics, and learning outcomes.

**Data :**

**A screenshot of a computer screen

Description automatically generated**

**Data Loading**: The course data was stored as dictionaries with course\_name and description fields.

**Step 2: Selecting and Setting Up the Vector Database**

Pinecone was chosen as the vector database for its scalability and integration capabilities. The following steps were taken to set up the environment:

1. **Account Creation**: An account was created on Pinecone.io, and the API key and environment details were retrieved.
2. **API Initialization**: The Pinecone client was initialized in Python using the provided API key and region.

**Step 3: Environment Setup**

The development environment was set up by installing the necessary dependencies:

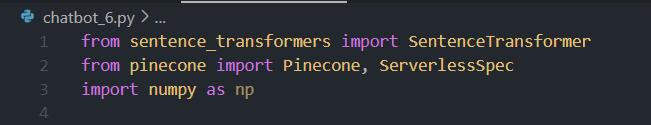
* sentence-transformers: To generate embeddings for the course descriptions.
* pinecone-client: To interact with the Pinecone database.
* numpy: For handling numerical operations.

**Installation Command**



**Step 4: Code Implementation**

**1. Initializing Pinecone**

****

Using the Pinecone API, the index was created to store course embeddings.

A screen shot of a computer program

Description automatically generated

**2. Generating Embeddings**

The sentence-transformers library was used to convert course descriptions into vector embeddings.

A screen shot of a computer code

Description automatically generated

**Embedding Generation**: The all-MiniLM-L6-v2 model converted text descriptions into 384-dimensional embeddings.

**Data Upsert**: Embeddings were added to the Pinecone index along with metadata for each course.

**3. Querying the Database**

A query function was defined to search the database for courses similar to a given query.

A screen shot of a computer code

Description automatically generated

**Query Execution**: The query text was embedded and matched against stored embeddings using cosine similarity.

**Results**

**A screen shot of a computer program

Description automatically generated**

**Conclusion**

This task of vector database embedding successfully demonstrated the use of Pinecone and sentence-transformers to create a vector database for course information. The system provides efficient semantic search capabilities, making it a scalable solution for similar use cases.