Product Recommender System: OpenAI Text Embedding

Key Takeaways

Task 1

Set Up the Project Environment

- Storing secret keys in an .env file is a vital security practice, reinforcing data protection and mitigating risks.
- *PCA from sklearn.decomposition*: This module is used for reducing the dimensionality of data while preserving as much information as possible.
- *Cosine Similarity from sklearn.metrics.pairwise*: This module is used for measuring the cosine of the angle between two vectors, providing a metric for determining similarity in high-dimensional spaces based on direction rather than magnitude.

Task 2

Prepare the dataset

- In Python, *isin()* method is used to check whether each element in a Series is contained in a provided list or another Series, returning a boolean Series indicating whether each element is in the passed list.
- Combining the title and description of products before extracting embedding vectors is recommended to include more comprehensive information about each product.

Task 3

Text embedding and visualization

- The longer the text input, the higher the dimension of the vector embedding should be, ensuring it encompasses enough information to capture the semantic meaning of the text input.
- While extracting the text embedding vectors, the **dimension** parameter is used to specify the number of dimensions the resulting output embeddings should have.
- A higher **dimension** implies that the vectors include more information about the text, but it also increases the cost.
- A text embedding vector is a vector representation of a given text input that includes *contextual semantic meaning*. This vector can be utilized for various purposes such as machine learning, classification, recommendation systems, and semantic search.

Task 4

Find similar products

- To compare two high-dimensional vectors, we can use various similarity measures such as cosine similarity to assess the similarity between these vectors.

- When developing a recommender system, it's advisable to compare vector directions rather than spatial distances, making cosine similarity the recommended choice for comparing the vectors.

Task 5

Recommend products based on the searched products

- To construct a text-based recommender system based on recently searched documents:

  - **Step 1:** Generate text embedding vectors for all documents.
  - **Step 2:** Generate text embedding vectors for searched documents, ensuring their dimensionality matches that of the document embedding vectors.
  - **Step 3:** Utilize a similarity measure to assess the similarity between each searched document vector and all other documents.
  - **Step 4:** Documents with vectors more similar to those of the searched documents are recommended, indicating shared semantic meaning.