**Build a Knowledge Based System with Vertex AI Vector Search, LangChain and Gemini**

**Overview**

[Gemini](https://deepmind.google/technologies/gemini/#introduction) is a family of generative AI models developed by Google DeepMind that is designed for multimodal use cases. The Gemini API gives you access to the Gemini Pro Vision and Gemini Pro models. In this lab, you use Vertex AI Vector Search to index documents and create a knowledge base. The knowledge base is utilized to retrieve relevant search results to supply with a query submitted to a large language model (LLM), in this case, Gemini, as context (a technique known as Retrieval Augmentation Generation).

This will in effect produce more specific results from the LLM and avoid re-training or fine-tuning an LLM with additional data.

[Vertex AI Vector Search](https://cloud.google.com/vertex-ai/docs/vector-search/overview) provides a high-scale, low latency vector database. Vector databases are commonly referred to as vector similarity-matching or an approximate nearest neighbor (ANN) service.

Vertex AI Vector Search can search from billions of semantically similar or semantically related items when queried. A vector similarity-matching service has many use cases such as implementing recommendation engines, search engines, chatbots, and text classification.

Semantic matching can be simplified into a few steps. First, you must generate embedding representations of many items (done outside of Vector Search). Next, you upload your embeddings to Google Cloud, and then link your data to Vector Search. After your embeddings are added to Vector Search, you can create an index to run queries to get recommendations or results.

Once the Vector Search index has been created it can be queried to return results which can be supplied to a Large Language Model (LLM) as context to generate better content using the results to without the need for fine-tuning the model which can be expensive or

**Vector Search**

There are a few key challenges when deploying Generative AI applications that leverage large language models (LLMs):

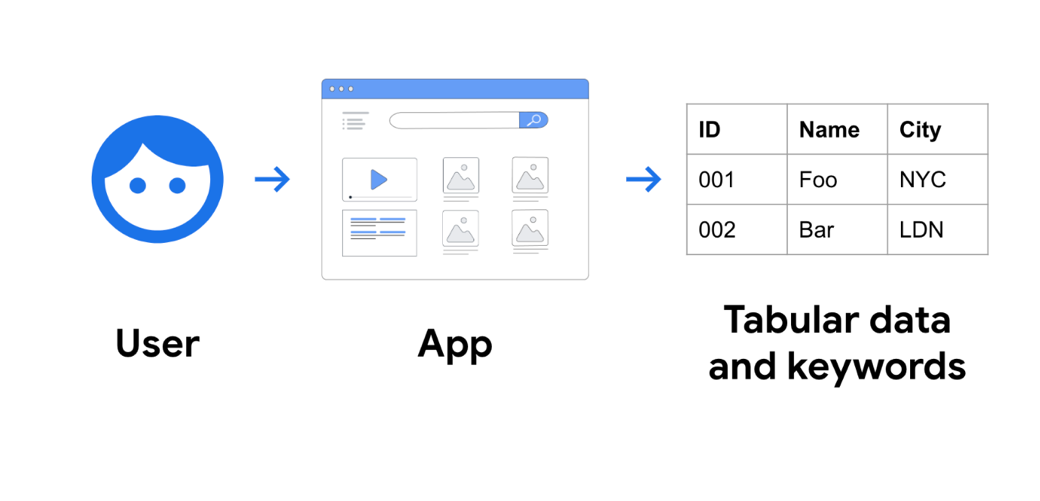
* Integration with existing systems
* Scalable product knowledge
* Mitigating hallucinations (inaccurate responses)

These challenges can be mitigated to an extent through the use of grounding techniques via knowledge-based systems using embeddings and vector databases.

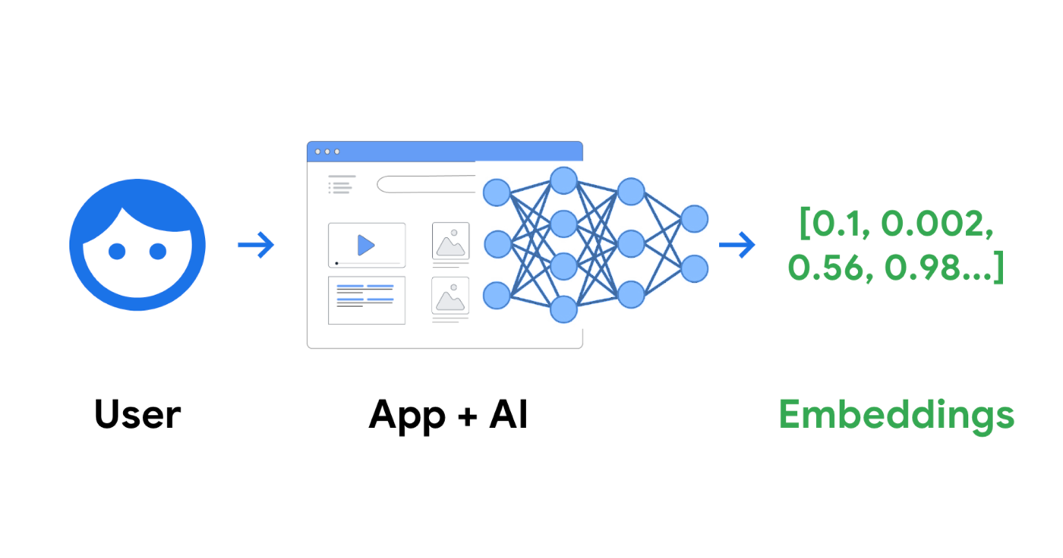
What are embeddings?

In traditional IT systems, most data is organized as structured or tabular data, using simple keywords, labels, and categories in databases and search engines.

This may look similar to the following figure.

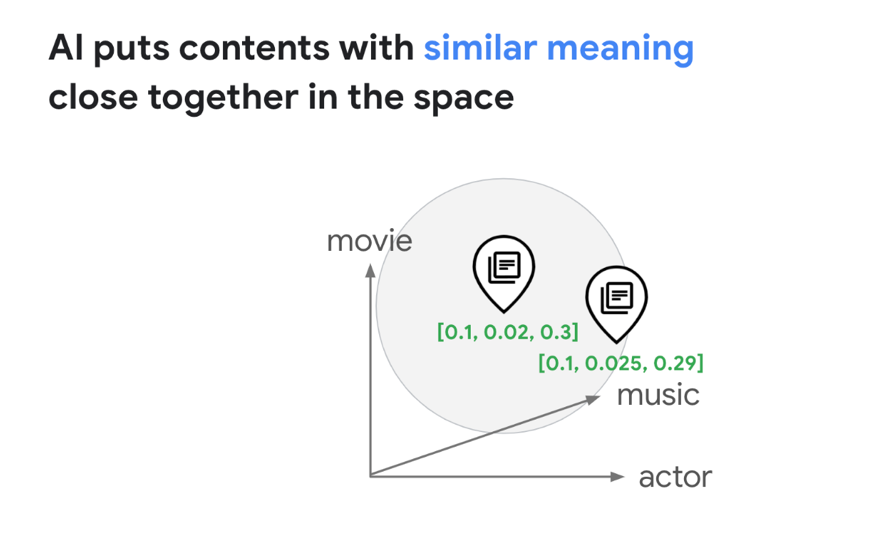


In contrast, AI-powered services arrange data into a simple data structure known as "embeddings". Embeddings are a numerical, floating point representation of data in a vector format.



Once trained with specific content like text, images, or any content, AI creates a space called an "embedding space", which is essentially a map of the content's meaning.

AI can identify the location of each content on the map, that's what embedding is. Let's take an example where a text paragraph discusses movies, music, and actors, with a distribution of 10%, 2%, and 30%, respectively. In this case, the AI can create an embedding with three values: 0.1, 0.02, and 0.3, in 3-dimensional space.

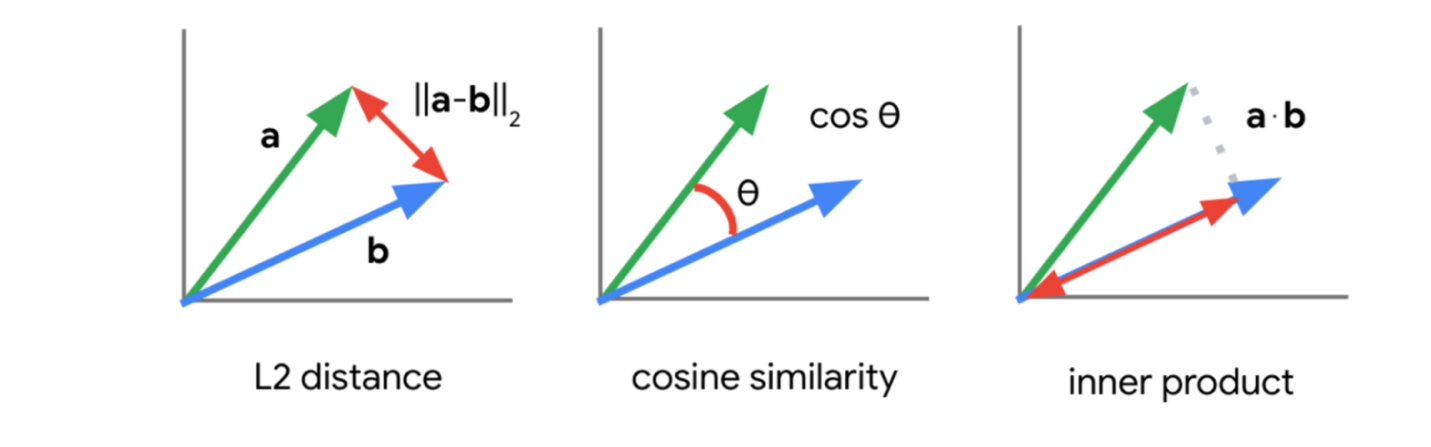


AI organizes data into embeddings, which represent what the user is looking for, the meaning of contents, or many other things you have in your business. This creates a new level of user experience that is becoming the new standard.

**Finding embedding similarities**

Once embeddings are generated by a LLM, they can be stored in a vector database for retrieval at a later time. Vector stores facilitate querying embeddings for similarities to questions posed by end users.

There are a few methodologies that can be used to determine vector similarities:



In this lab, you will use the inner product (also known as the dot product distance) to find similarities between vector embeddings.

Using this methodology, a query submitted to a LLM is embedded as a vector and used in its embedding representation to find similar documents indexed by the vector store. The results returned are used as **context** to prevent the LLM from hallucinating and providing more factual results. This technique is known as **Retrieval Augmentation Generation**.

**What is Vertex AI Vector Search?**

Google Cloud developers can take the full advantage of Google's vector search technology with [Vertex AI Vector Search](https://cloud.google.com/vertex-ai/docs/vector-search/overview) (previously called Matching Engine).

With Vertex AI Vector Search, developers can add the embeddings to an index and issue a search query for blazing fast vector search. Vertex AI Vector Search is capable of searching billions of embeddings with sub millisecond retrieval times.