

□□□

□Now

□Dream Small

□Life as an Album

□Thoughts from Work

□Links

Thoughts from Work / Vector Search in BigQuery - Test Drive

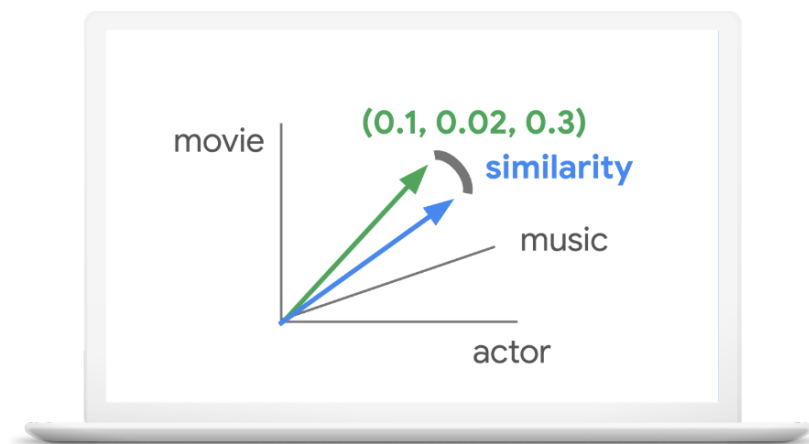
#BigQuery #VectorDB #VectorSearch #GCP

Conducting Clustering Analysis on users, consumers groups is common techniques for us to uncover hidden patterns, hidden user behaviors etc, from classic K Means to now commonly used DBSCAN, we are able compute **Vectors for each customers from the features we selected depends on particular scenarios.**

e.g.:

A customer's age, income, purchase_amount: **[30, 60000, 100]**, thus a customer is now visualized in a vector.

whereas vector search uses **vector similarity**
to find relevant content



It is ideal for “semantic” search based on similarity. Imagining we store vectorized survey data in DB, we are now able to compute similar feedbacks from users without initiated a python notebook again and again.

Given now, BigQuery native support **VECTOR_SEARCH^[1]**, now we basically can conduct ****Approximate Nearest Neighbor**** search directly in BigQuery console, without using python, below is a quick demonstration.

Create two tables with `Array` data format storing `embedding` values`:

1. Create a base table - imagining this table is called Zoo, each vector are different animals

```
-- random is the dataset name in my GCP project
create or replace table random.embedding_example_base (
  id INT64,
  my_embedding array<float64>,
);

insert random.embedding_example_base (id, my_embedding)
VALUES(1, [1.0, 2.0]),
(2, [2.0, 4.0]),
(3, [1.5, 7.0]),
(4, [1.0, 3.2]),
(5, [5.0, 5.4]),
(6, [3.7, 1.8]),
(7, [4.4, 2.9]);
```

1. Create a query table - imagining data points in this table is animals you try to

1. Create a query table - imagine data points in this table is animals you try to find their "similar" species, in this case, you have dog and cat

```
create or replace table random.embedding_example_query (
  query_id STRING,
  my_embedding array<float64>,
);
```

```
insert random.embedding_example_query (query_id, my_embedding)
VALUES('dog', [1.0, 2.0]),
('cat', [3.0, 5.2]);
```

embedding_e...

<

Schema

Details

Preview

>

Row	id	my_embedding
1	5	5.0
		5.4
2	2	2.0
		4.0
3	1	1.0
		2.0
4	3	1.5
		7.0
5	7	4.4
		2.9
6	4	1.0
		3.2
7	6	3.7
		1.8

embedding_example_query

<

Schema

Details

Preview

>

Row	query_id	my_embedding
1	dog	1.0
		2.0
2	cat	3.0
		5.2

Now, let's do `VECTOR_SEARCH`

```
select
  *
from vector_search(
  table random.embedding_example_base -- base table
  , 'my_embedding' -- column to search
  , (
    select query_id
    , my_embedding as query_embedding
    from random.embedding_example_query
  )
  , 'query_embedding'
  , top_k => 2
  , distance_type => 'COSINE' -- EUCLIDEAN, DOT_PRODUCT
)
```

- `=>` this is assign symbol

And as for Distance Types:

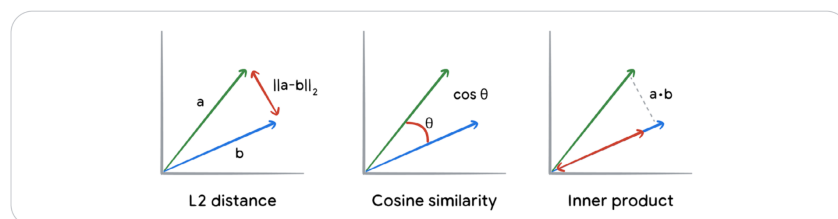


Figure 11. Visualization of how different metrics compute vector similarity

Then it return following values:

- Top 2 most similar **vectors** based on distance values

query.query_id	query_embedding	base.id	my_embedding	distance
dog	1.0	1	1.0	0.0

dog	1.0	1	1.0	0.0
	2.0		2.0	
cat	3.0	1	1.0	0.0017773842088002478
	5.2		2.0	
dog	1.0	2	2.0	0.0
	2.0		4.0	
cat	3.0	2	2.0	0.0017773842088002478
	5.2		4.0	

Final Thoughts

Besides, use BQ as a vector store Db for applications. Now, if a company has ML pipeline (or just BigQuery's own ML sql)^[2] constantly output vectors represent customers behaviors, then now, Data Scientists could directly query those tables and conduct quick similarity analysis without using Python.

Vector Search is the new select *

[BigQuery vector search and embedding generation - YouTube](#)

[BigQuery as a Vector Database — how cool is that_ _ by Shuvro @ Nimesa _ Medium.pdf](#)

1. [Search functions | BigQuery | Google Cloud](#)

2. [BigQuery as a Vector Database — how cool is that? | by Shuvro @ Nimesa | Medium](#)

[Previous](#)

[Next](#)

2018-04-01

Tuning the Supply and Demand in R...

2025-02-10

AI-Driven Layoff ~ Quick thoughts o...

Klyn | © 2022-2025

[Tags](#) [Archive](#) [RSS feed](#) [Twitter](#) [Instagram](#) [GitHub](#) [Youtube](#) [Email](#) [QR Code](#)

Made with [Montaigne](#) and [bigmission](#) 🇺🇦