Using the ArangoDB Vector Index

Blog Post:
https://arangodb.com/2024/11/vector-search-in-arangodb-practical-insights-and-hands-on-examples/

Step 1: Provision the ArangoDB Instance using Docker

```
None
>>> docker run --name arango -p 8529:8529 -e
ARANGO_ROOT_PASSWORD=test arangodb/arangodb:3.12.4
--experimental-vector-index true
```

Confirm instance is running by navigating to http://localhost:8529/

Step 2: Load sample data using python

The following packages must be first installed with **pip**:

- **python-arango**
- **arango-datasets**

```python
Python

# pip install python-arango
# pip install arango-datasets

from arango import ArangoClient
from arango_datasets import Datasets
import math

################
# Step 1: Connect to the DB & load a sample dataset
################

db = ArangoClient().db(password="test")
```

```
# IMDB_PLATFORM is a dataset containg USERs that VIEW a selection of MOVIEs
Datasets(db).load("IMDB_PLATFORM")

###############
# Step 2: Create the ArangoDB Vector Indices (using Cosine metric):
###############

# Vector Index for USER collection
db.collection("USER").add_index(
    {
        "name": "vector-index",
        "type": "vector",
        "fields": ["features"],
        "params": {
            "metric": "cosine",
            "dimension": len(db.collection("USER").random()["features"]),
            "nLists": round(math.sqrt(db.collection("USER").count())),
        },
    }
)

# Vetor Index for MOVIE collection
db.collection("MOVIE").add_index(
    {
        "name": "vector-index",
        "type": "vector",
        "fields": ["features"],
        "params": {
            "metric": "cosine",
            "dimension": len(db.collection("MOVIE").random()["features"]),
            "nLists": round(math.sqrt(db.collection("MOVIE").count())),
        },
    }
)
```

Step 3: Execute AQL queries that use Approximate Nearest Neighbor functionality:

```
None
```

```
example_query_1 =  """
LET user_1 = DOCUMENT("USER/1")

FOR u IN USER
       // Fetch top 5 similar users
       LET score = APPROX_NEAR_COSINE(u.features, user_1.features)
       SORT score DESC
       LIMIT 5
       RETURN {id: u._id, score}
"""

example_query_2 = """
LET user_1 = DOCUMENT("USER/1")

LET user_1_movie_watched = (
  FOR movie IN 1..1 OUTBOUND user_1 VIEWS
    RETURN movie._id
)

FOR u IN USER

  // Fetch top 5 similar users
  LET score = APPROX_NEAR_COSINE(u.features, user_1.features)
  SORT score DESC
  LIMIT 5

  // Iterate over movies to recommend to USER/1
  // based on Movies watched by similar Users
  FOR movie, edge IN 1..1 OUTBOUND u VIEWS
    FILTER movie._id NOT IN user_1_movie_watched

    COLLECT title = movie.title WITH COUNT INTO number
    SORT number DESC
    LIMIT 5
    RETURN {title, number}


"""
```