



generative-ai-jupyterlab

File Edit View Run Kernel Git Tabs Settings Help

e2-standard-2 ▾

!

Untitled.ipynb X + Code ⏴ git ⏳ Execute

Python 3 (ipykernel) (Local)

Prepare VPC Network and Install Packages

Install Stuff

```
[35]: PROJECT_ID = "qwiklabs-gcp-00-349101c967a5"
NETWORK_NAME = "default"
PEERING_RANGE_NAME = "cym-range"

# Reserve IP range
! gcloud compute addresses create {PEERING_RANGE_NAME} --global --prefix-length=16 --network={NETWORK_NAME} --purpose=VPC_PEERING --project={PROJECT_ID} --description="peering range

# Set up peering with service networking
! gcloud services vpc-peerings connect --service=servicenetworking.googleapis.com --network={NETWORK_NAME} --ranges={PEERING_RANGE_NAME} --project={PROJECT_ID}

ERROR: (gcloud.compute.addresses.create) Could not fetch resource:
- The resource 'projects/qwiklabs-gcp-00-349101c967a5/global/addresses/cym-range' already exists

Operation "operations/pssn.p24-272048638912-82e63a28-8cf9-4ccb-88ff-1f93565c27c2" finished successfully.
```

Download and install the latest (preview) version of the Vertex AI SDK for Python.

```
[2]: ! pip install -U git+https://github.com/googleapis/python-aiplatform.git@main --user

Collecting git+https://github.com/googleapis/python-aiplatform.git@main
  Cloning https://github.com/googleapis/python-aiplatform.git (to revision main) to /var/tmp/pip-req-build-h8agz4ck
    Running command git clone --filter=blob:none --quiet https://github.com/googleapis/python-aiplatform.git /var/tmp/pip-req-build-h8agz4ck
      Resolved https://github.com/googleapis/python-aiplatform.git to commit 2b2dc4fbc5d5b16159868b112affafa8c716ae
      Preparing metadata (setup.py) ... done
Requirement already satisfied: google-api-core<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (2.24.2)
Requirement already satisfied: google-auth<3.0.0,>>=2.14.1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (2.38.0)
Requirement already satisfied: proto-plus<2.0.0,>>=1.22.3 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.26.1)
Requirement already satisfied: protobuf<=4.21.0,!>=4.21.1,!>=4.21.2,!>=4.21.3,!>=4.21.5,<7.0.0,>>=3.20.2 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (3.20.3)
Requirement already satisfied: packaging<=14.3 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (2.4.2)
Requirement already satisfied: google-cloud-storage<3.0.0,>>=1.32.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.32.0->google-cloud-aiplatform==1.94.0) (2.19.0)
Requirement already satisfied: google-cloud-bigquery<!=3.20.0,>4.0.0,>>=1.15.0 in ./local/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.15.0->google-cloud-aiplatform==1.94.0) (3.33.0)
Requirement already satisfied: google-cloud-resource-manager<3.0.0,>>=1.3.3 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.3.3->google-cloud-aiplatform==1.94.0) (2.0.7)
Requirement already satisfied: shapeley<3.0.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (2.0.7)
Requirement already satisfied: google-genai<2.0.0,>>=1.0.0 in ./local/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.0.0->google-cloud-aiplatform==1.94.0) (1.17.0)
Requirement already satisfied: pydantic<3 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (2.11.0)
Requirement already satisfied: typing_extensions< in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.43.0->google-cloud-aiplatform==1.94.0) (4.13.0)
Requirement already satisfied: docstring_parser<1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.43.0->google-cloud-aiplatform==1.94.0) (0.16)
Requirement already satisfied: googleapis-common-protos<2.0.0,>>=1.56.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.69.2)
Requirement already satisfied: requests<3.0.0,>>=2.18.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=2.18.0->google-cloud-aiplatform==1.94.0) (2.32.3)
Requirement already satisfied: grpcio<2.0.0dev,>>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.71.0)
Requirement already satisfied: grpcio-status<2.0.dev.0,>>=1.33.2 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.49.0rc1)
Requirement already satisfied: cachetools<6.0,>>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=2.0.0->google-cloud-aiplatform==1.94.0) (5.5.2)
Requirement already satisfied: pyasn1-modules<=0.2.1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=0.2.1->google-cloud-aiplatform==1.94.0) (0.4.2)
Requirement already satisfied: rsa<5,>>=3.1.4 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=3.1.4->google-cloud-aiplatform==1.94.0) (4.9)
Requirement already satisfied: google-cloud-bigquery<!=3.20.0,>4.0.0->=1.15.0->google-cloud-aiplatform==1.94.0 (2.4.3)
Requirement already satisfied: google-resumable-media<3.0.0,>>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.71.0)
Requirement already satisfied: python-dateutil<3.0.0,>>=2.8.2 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=2.8.2->google-cloud-aiplatform==1.94.0) (5.5.2)
Requirement already satisfied: pyasn1<=0.2.1 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=0.2.1->google-cloud-aiplatform==1.94.0) (0.4.2)
Requirement already satisfied: google-cloud-resource-manager<3.0.0,>>=1.3.3 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.3.3->google-cloud-aiplatform==1.94.0) (2.4.3)
Requirement already satisfied: google-crc32c<2.0.dev.0,>>=1.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.0->google-cloud-aiplatform==1.94.0) (1.7.1)
Requirement already satisfied: aiohttp<5.0.0,>>=4.8.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=4.8.0->google-cloud-aiplatform==1.94.0) (4.9.0)
Requirement already satisfied: httpx<1.0.0,>>=0.28.1 in ./local/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=0.28.1->google-cloud-aiplatform==1.94.0) (0.28.1)
Requirement already satisfied: websockets<15.1.0,>>=13.0.0 in /opt/conda/lib/python3.10/site-packages (from google-aiplatform[grpc]!<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=13.0.0->google-cloud-aiplatform==1.94.0) (15.0.1)
Requirement already satisfied: annotated-types<=0.6.0 in /opt/conda/lib/python3.10/site-packages (from pydantic<3>,>=1.34.1->google-cloud-aiplatform==1.94.0) (0.7.0)
Requirement already satisfied: pydantic-core<=2.33.0 in /opt/conda/lib/python3.10/site-packages (from pydantic<3>,>=1.34.1->google-cloud-aiplatform==1.94.0) (2.33.0)
Requirement already satisfied: typing-inspect<=0.4.0 in /opt/conda/lib/python3.10/site-packages (from pydantic<3>,>=1.34.1->google-cloud-aiplatform==1.94.0) (0.4.0)
Requirement already satisfied: numpy<3,>>=1.14 in /opt/conda/lib/python3.10/site-packages (from shapely<3.0.0,>>1.34.1->google-cloud-aiplatform==1.94.0) (2.1.3)
Requirement already satisfied: exceptiongroup<=1.0.2 in /opt/conda/lib/python3.10/site-packages (from anyio<5.0.0,>>4.8.0->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (1.2.2)
Requirement already satisfied: idna<=2.8 in /opt/conda/lib/python3.10/site-packages (from anyio<5.0.0,>>4.8.0->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (3.10)
Requirement already satisfied: sniffio<=1.1 in /opt/conda/lib/python3.10/site-packages (from anyio<5.0.0,>>4.8.0->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (1.3.1)
Requirement already satisfied: certifi in /opt/conda/lib/python3.10/site-packages (from httpx<1.0.0,>>0.28.1->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (2025.1.3)
Requirement already satisfied: httpcore<=1.* in ./local/lib/python3.10/site-packages (from httpx<1.0.0,>>0.28.1->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (1.0.9)
Requirement already satisfied: h11<=0.16 in ./local/lib/python3.10/site-packages (from httpcore<=1.*->httpx<1.0.0,>>0.28.1->google-genai<2.0.0,>>1.0.0->google-cloud-aiplatform==1.94.0) (0.16.0)
Requirement already satisfied: pyasn1<0.7.0,>>=0.6.1 in /opt/conda/lib/python3.10/site-packages (from pyasn1-modules<=0.2.1->google-auth<3.0.0,>>2.14.1->google-cloud-aiplatform==1.94.0) (0.6.1)
Requirement already satisfied: six<=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil<3.0.0,>>2.8.2->google-cloud-bigquery<!=3.20.0,>4.0.0->google-cloud-aiplatform==1.94.0) (1.17.0)
Requirement already satisfied: charset_normalizer<4,>>=2 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>>2.18.0->google-api-core<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=2.1.1->google-cloud-aiplatform==1.94.0) (3.4.1)
Requirement already satisfied: urllib3<3,>>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests<3.0.0,>>2.18.0->google-api-core<=2.0.*,!>=2.1.*,!>=2.2.*,!>=2.3.*,!>=2.4.*,!>=2.5.*,!>=2.6.*,!>=2.7.*,<3.0.0,>>=1.34.1->google-cloud-aiplatform==1.94.0) (1.2.6.20)
```

Install the h5py to prepare sample dataset, and the grpcio-tools for querying against the index.

```
[3]: ! pip install -U grpcio-tools --user
! pip install -U h5py --user
! pip install proto-plus==1.24.0.dev1

Collecting grpcio-tools
  Downloading grpcio_tools-1.71.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.3 kB)
Collecting protobuf<6.0dev,>=5.26.1 (from grpcio-tools)
  Downloading protobuf-5.29.5-cp38-abi3-manylinux2014_x86_64.whl.metadata (592 bytes)
Requirement already satisfied: grpcio==1.71.0 in /opt/conda/lib/python3.10/site-packages (from grpcio-tools) (1.71.0)
Requirement already satisfied: setuptools<5.0 in /opt/conda/lib/python3.10/site-packages (from grpcio-tools) (75.8.2)
Downloaded grpcio_tools-1.71.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.5 MB)
```

```

Downloading protobuf-5.29.5-cp38abi3manylinux2014_x86_64.whl (319 kB)
Installing collected packages: protobuf, grpcio-tools
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
google-cloud-datastore 1.15.5 requires protobuf<4.0.0dev, but you have protobuf 5.29.5 which is incompatible.
kfp 2.5.0 requires protobuf<4,>=3.13.0, but you have protobuf 5.29.5 which is incompatible.
kfp-pipeline-spec 0.2.2 requires protobuf<4,>=3.13.0, but you have protobuf 5.29.5 which is incompatible.
opentelemetryproto 1.15.0 requires protobuf<5.0,>=3.19, but you have protobuf 5.29.5 which is incompatible.
Successfully installed grpcio-tools-1.71.0 protobuf-5.29.5
Collecting h5py
  Downloading h5py-3.13.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.5 kB)
Requirement already satisfied: numpy>=1.19.3 in /opt/conda/lib/python3.10/site-packages (from h5py) (2.1.3)
  Downloading h5py-3.13.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.5 MB)
                                             4.5/4.5 MB 61.5 MB/s eta 0:00:00
Installing collected packages: h5py
Successfully installed h5py-3.13.0
Collecting proto-plus==1.24.0.dev1
  Downloading proto_plus-1.24.0.dev1-py3-none-any.whl.metadata (2.2 kB)
Collecting protobuf<5.0.0dev,>=3.19.0 (from proto-plus==1.24.0.dev1)
  Downloading protobuf-4.25.8-cp37abi3manylinux2014_x86_64.whl.metadata (541 bytes)
Downloaded proto_plus-1.24.0.dev1-py3-none-any.whl (49 kB)
Downloaded protobuf-4.25.8-cp37abi3manylinux2014_x86_64.whl (294 kB)
Installing collected packages: protobuf, proto-plus
Attempting uninstall: protobuf
  Found existing installation: protobuf 5.29.5
  Uninstalling protobuf-5.29.5:
    Successfully uninstalled protobuf-5.29.5
Attempting uninstall: proto-plus
  Found existing installation: proto-plus 1.26.1
  Uninstalling proto-plus-1.26.1:
    Successfully uninstalled proto-plus-1.26.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
grpcio-tools 1.71.0 requires protobuf<6.0dev,>=5.26.1, but you have protobuf 4.25.8 which is incompatible.
google-cloud-datastore 1.15.5 requires protobuf<4.0.0dev, but you have protobuf 4.25.8 which is incompatible.
kfp 2.5.0 requires protobuf<4,>=3.13.0, but you have protobuf 4.25.8 which is incompatible.
kfp-pipeline-spec 0.2.2 requires protobuf<4,>=3.13.0, but you have protobuf 4.25.8 which is incompatible.
Successfully installed proto-plus-1.24.0.dev1 protobuf-3.20.3

```

Restart Kernel

After you install the additional packages, you need to restart the notebook kernel so it can find the packages.

```
[4]: # Automatically restart kernel after installs
import os

if not os.getenv("IS_TESTING"):
    # Automatically restart kernel after installs
    import IPython

    app = IPython.Application.instance()
    app.kernel.do_shutdown(True)
```

Create a Cloud Storage Bucket

Create a Cloud Storage Bucket in [Lab GCP Region] region

Set the Variables

```
[1]: BUCKET_NAME = "gs://qwiklabs-gcp-00-349101c967a5-aip"
REGION = "us-central1"
PROJECT_ID = "qwiklabs-gcp-00-349101c967a5"
NETWORK_NAME = "default"
```

Make Bucket

```
[2]: ! gsutil mb -l $REGION -p $PROJECT_ID $BUCKET_NAME
Creating gs://qwiklabs-gcp-00-349101c967a5-aip/...
```

Validate access to your Cloud Storage bucket by examining its contents

```
[3]: ! gsutil ls -al $BUCKET_NAME
```

Import Libraries

Import the Vertex AI (unified) client library into your Python environment.

Import Vertex AI client library

```
[4]: # Upgrade protobuf to the latest version
!pip install --upgrade protobuf

import time
import grpc
import h5py
from google.cloud import aiplatform_v1
from google.protobuf import struct_pb2

Requirement already satisfied: protobuf in /opt/conda/lib/python3.10/site-packages (3.20.3)
Collecting protobuf
  Downloading protobuf-6.31.1-cp39abi3manylinux2014_x86_64.whl.metadata (593 bytes)
Downloaded protobuf-6.31.1-cp39abi3manylinux2014_x86_64.whl (321 kB)
Installing collected packages: protobuf
Attempting uninstall: protobuf
  Found existing installation: protobuf 3.20.3
  Uninstalling protobuf-3.20.3:
    Successfully uninstalled protobuf-3.20.3
WARNING: Failed to remove contents in a temporary directory '/opt/conda/lib/python3.10/site-packages/google/protobuf/_pycache_'.
You can safely remove it manually.
WARNING: Failed to remove contents in a temporary directory '/opt/conda/lib/python3.10/site-packages/google/protobuf/internal/_pycache_'.
You can safely remove it manually.
WARNING: Failed to remove contents in a temporary directory '/opt/conda/lib/python3.10/site-packages/google/protobuf/_yext'.
You can safely remove it manually.
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
grpcio-tools 1.71.0 requires protobuf<6.0dev,>=5.26.1, but you have protobuf 6.31.1 which is incompatible.
google-cloud-datastore 1.15.5 requires protobuf<4.0.0dev, but you have protobuf 6.31.1 which is incompatible.
kfp 2.5.0 requires protobuf<4,>=3.13.0, but you have protobuf 6.31.1 which is incompatible.
kfp-pipeline-spec 0.2.2 requires protobuf<4,>=3.13.0, but you have protobuf 6.31.1 which is incompatible.
opentelemetryproto 1.15.0 requires protobuf<5.0,>=3.19, but you have protobuf 6.31.1 which is incompatible.
proto-plus 1.24.0.dev1 requires protobuf<5.0.0dev,>=3.19.0, but you have protobuf 6.31.1 which is incompatible.
Successfully installed protobuf-4.25.8
```

```
[5]: REGION = "us-central1"
ENDPOINT = "{}-aiplatform.googleapis.com".format(REGION)
NETWORK_NAME = "default"

AUTH_TOKEN = !gcloud auth print-access-token
PROJECT_NUMBER = !gcloud projects list --filter="PROJECT_ID:'{}'".format(PROJECT_ID) --format='value(PROJECT_NUMBER)'
PROJECT_NUMBER = PROJECT_NUMBER[0]

PARENT = "projects/{}/locations/{}".format(PROJECT_ID, REGION)

print("ENDPOINT: {}".format(ENDPOINT))
print("PROJECT_ID: {}".format(PROJECT_ID))
print("REGION: {}".format(REGION))

!gcloud config set project {} --quiet
!gcloud config set ai_platform/region {} --quiet

ENDPOINT: us-central1-aiplatform.googleapis.com
PROJECT_ID: quicklabs-gcp-00-349101c967a5
REGION: us-central1
Updated property [core/project].
Updated property [ai_platform/region].
```

Prepare the Data

The GloVe dataset consists of a set of pre-trained embeddings.

The embeddings are split into a "train" split, and a "test" split.

We will create a vector search index from the "train" split, and use the embedding vectors in the "test" split as query vectors to test the vector search index.

Note: While the data split uses the term "train", these are pre-trained embeddings and thus are ready to be indexed for search.

The terms "train" and "test" split are used just to be consistent with usual machine learning terminology.

Download the GloVe dataset

```
[6]: ! gsutil cp gs://cloud-samples-data/vertex-ai/matching_engine/glove-100-angular.hdf5 .

Copying gs://cloud-samples-data/vertex-ai/matching_engine/glove-100-angular.hdf5...
\ [1 files][462.9 MiB/462.9 MiB]
Operation completed over 1 objects/462.9 MiB.
```

Read the data into memory

```
[7]: # The number of nearest neighbors to be retrieved from database for each query.
k = 10
```

```
h5 = h5py.File("glove-100-angular.hdf5", "r")
train = h5["train"]
test = h5["test"]
```

```
[8]: train[0]
```

```
[8]: array([-0.11333 ,  0.48402 ,  0.090771 , -0.22439 ,  0.034206 ,
-0.55831 ,  0.041849 , -0.53573 ,  0.18809 , -0.58722 ,
0.015313 , -0.014555 ,  0.80842 , -0.038519 ,  0.75348 ,
0.70502 , -0.17863 ,  0.3222 ,  0.67575 ,  0.67198 ,
0.26044 ,  0.4187 , -0.34122 ,  0.2286 , -0.53529 ,
1.2582 , -0.091543 ,  0.19716 , -0.037454 , -0.3336 ,
0.31399 ,  0.36488 ,  0.71263 ,  0.1307 , -0.24654 ,
-0.52445 , -0.036091 ,  0.55068 ,  0.18017 ,  0.48095 ,
0.71104 , -0.053462 ,  0.22325 ,  0.30917 , -0.39926 ,
0.036634 , -0.35431 , -0.42795 ,  0.46444 ,  0.25586 ,
0.68257 , -0.20821 ,  0.38433 ,  0.055773 , -0.2539 ,
-0.20804 ,  0.52522 , -0.11399 , -0.3253 , -0.44104 ,
0.17528 ,  0.62255 ,  0.50237 , -0.7607 , -0.071786 ,
0.0080131, -0.13286 ,  0.50097 ,  0.18824 , -0.54722 ,
-0.42664 ,  0.4292 ,  0.14877 , -0.0072514 , -0.16484 ,
-0.059798 ,  0.9895 , -0.61738 ,  0.054169 ,  0.48424 ,
-0.35084 , -0.27053 ,  0.37829 ,  0.11503 , -0.39613 ,
0.24266 ,  0.39147 , -0.075256 ,  0.65093 , -0.20822 ,
-0.17456 ,  0.53571 , -0.16537 ,  0.13582 , -0.56016 ,
0.016964 ,  0.1277 ,  0.94071 , -0.22608 , -0.021106 ],
dtype=float32)
```

Save the train split in JSONL format

```
[9]: # Add restricts to each data point, in this demo, we only add one namespace and the allowlist is set to be the same as the id.
# Later on, we will demo how to return only the allowlisted data points.
# Split datapoints into two groups 'a' and 'b'. The datapoint whose ids are even are in group 'a', otherwise are in group 'b'
# We will demo how to configure the query to return up to k data points for each group.
with open('glove100.json', "w") as f:
    for i in range(len(train)):
        f.write('{"id":"' + str(i) + '",')
        f.write('"embedding":"' + ",".join(str(x) for x in train[i]) + '",')
        f.write('"restricts": [{"namespace": "class", "allow": ["' + str(i) + '"]}],')
        f.write('"crowding_tag": "' + ("a" if i % 2 == 0 else "b") + '"')
        f.write("\n")
    if i >= 100:
        break
```

Upload the training data to Google Cloud Storage Bucket created earlier

```
[10]: # NOTE: Everything in this GCS DIR will be DELETED before uploading the data.

! gsutil rm -rf {BUCKET_NAME}/*
```

CommandException: 1 files/objects could not be removed.

```
[11]: ! gsutil cp glove100.json {BUCKET_NAME}/glove100.json

Copying file://glove100.json [Content-Type=application/json]...
/ [1 files][ 93.1 KiB/ 93.1 KiB]
Operation completed over 1 objects/93.1 KiB.
```

List the newly added contents in the bucket

```
[12]: ! gsutil ls {BUCKET_NAME}
gs://quickstarts-gcp-00-349101c967a5-aip/glove100.json
```

Create Stream | Update Index

Create Stream Update Index

Run the following code snippet in the next cells to create a instance of the `IndexServiceClient` from the AI Platform (Unified) Python client library. This is used for interacting with AI Platform services related to indexes, such as creating and managing Approximate Nearest Neighbor (ANN)

```
[13]: index_client = aiplatform_v1.IndexServiceClient(  
        client_options=dict(api_endpoint=ENDPOINT)  
)  
  
[14]: DIMENSIONS = 100  
DISPLAY_NAME = "glove_100_1"
```

Let's define the configuration for creating an Approximate Nearest Neighbor (ANN) index.

```
[15]: treeAhConfig = struct_pb2.Struct(  
    fields={  
        "leafNodeEmbeddingCount": struct_pb2.Value(number_value=500),  
        "leafNodesToSearchPercent": struct_pb2.Value(number_value=7),  
    })  
  
algorithmConfig = struct_pb2.Struct(  
    fields={"treeAhConfig": struct_pb2.Value(struct_value=treeAhConfig)})  
  
config = struct_pb2.Struct(  
    fields={  
        "dimensions": struct_pb2.Value(number_value=DIMENSIONS),  
        "approximateNeighborsCount": struct_pb2.Value(number_value=150),  
        "distanceMeasureType": struct_pb2.Value(string_value="DOT_PRODUCT_DISTANCE"),  
        "algorithmConfig": struct_pb2.Value(struct_value=algorithmConfig),  
    })  
  
metadata = struct_pb2.Struct(  
    fields={  
        "config": struct_pb2.Value(struct_value=config),  
        "contentsDeltaUri": struct_pb2.Value(string_value=BUCKET_NAME),  
    })  
  
ann_index = {  
    "display_name": DISPLAY_NAME,  
    "description": "Glove 100 ANN index",  
    "metadata": struct_pb2.Value(struct_value=metadata),  
    "index_update_method": aiplatform_v1.Index.IndexUpdateMethod.STREAM_UPDATE,  
}
```

Create an Approximate Nearest Neighbor (ANN) index using the specified configuration

```
[16]: ann_index = index_client.create_index(parent=PARENT, index=ann_index)
```

Retrieve the result of previously executed `ann_index`.

Note It can take up to 15 minutes to be available.

```
[17]: ann_index.result()  
  
[17]: name: "projects/272048638912/locations/us-central1/indexes/5062153733303959552"  
display_name: "glove_100_1"  
description: "Glove 100 ANN index"  
metadata_schema_uri: "gs://google-cloud-aiplatform/schema/matchingengine/metadata/nearest_neighbor_search_1.0.0.yaml"  
metadata {  
    struct_value {  
        fields {  
            key: "config"  
            value {  
                struct_value {  
                    fields {  
                        key: "algorithmConfig"  
                        value {  
                            struct_value {  
                                fields {  
                                    key: "treeAhConfig"  
                                    value {  
                                        struct_value {  
                                            fields {  
                                                key: "leafNodeEmbeddingCount"  
                                                value {  
                                                    string_value: "500"  
                                                }  
                                            }  
                                            fields {  
                                                key: "leafNodesToSearchPercent"  
                                                value {  
                                                    number_value: 7.0  
                                                }  
                                            }  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
    fields {  
        key: "approximateNeighborsCount"  
        value {  
            number_value: 150.0  
        }  
    }  
    fields {  
        key: "dimensions"  
        value {  
            number_value: 100.0  
        }  
    }  
    fields {  
        key: "distanceMeasureType"  
        value {  
            string_value: "DOT_PRODUCT_DISTANCE"  
        }  
    }  
    fields {  
        key: "shardSize"  
        value {  
    }
```

Finally, retrieve the name of the created index (ann_index) and store it in the variable INDEX_RESOURCE_NAME

```
[18]: INDEX_RESOURCE_NAME = ann_index.result().name  
INDEX_RESOURCE_NAME
```

```
[18]: 'projects/272048638912/locations/us-central1/indexes/506215373303959552'
```

Create an IndexEndpoint with VPC Network

Run the following code snippet to initialize an AI Platform Index Endpoint Service client.

```
[19]: index_endpoint_client = aiplatform_v1.IndexEndpointServiceClient(
        client_options=dict(api_endpoint=ENDPOINT)
    )
```

Define a configuration for an AI Platform Index Endpoint.

```
[20]: VPC_NETWORK_NAME = "projects/{}/global/networks/{}".format(PROJECT_NUMBER, NETWORK_NAME)
VPC_NETWORK_NAME
```

```
[20]: 'projects/272048638912/global/networks/default'
```

```
[1]: index_endpoint = {
    "display_name": "index_endpoint_for_demo",
    "network": VPC_NETWORK_NAME,
}
```

Create an AI Platform Index Endpoint

```
[22]: r = index_endpoint_client.create_index_endpoint(  
        parent=PARENT, index_endpoint=index_endpoint  
)
```

Retrieve the result of previous operation.

```
[23]: r.result()

[23]: name: "projects/272048638912/locations/us-central1/indexEndpoints/6122294048712556544"
      display_name: "index_endpoint_for_demo"
      network: "projects/272048638912/global/networks/default"
      encryption_spec {
      }
```

Finally, retrieve the name of the created index endpoint and store it in the variable INDEX_ENDPOINT_NAME

```
[24]: INDEX_ENDPOINT_NAME = r.result().name  
INDEX ENDPOINT NAME
```

```
[24]: 'projects/272048638912/locations/us-central1/indexEndpoints/6122294048712556544
```

Deploy Stream Update Index

Run the following code snippet in the next cells to deploy the index to an index endpoint.

```
[25]: DEPLOYED_INDEX_ID = "stream_update_glove_deployed"
```

```
[26]: deploy_ann_index = {
        "id": DEPLOYED_INDEX_ID,
        "display_name": DEPLOYED_INDEX_ID,
        "index": INDEX_RESOURCE_NAME,
    }
```

```
[36]: r = index_endpoint_client.deploy_index(
    index_endpoint=INDEX_ENDPOINT_NAME, deployed_index=deploy_ann_index
)
```

Wait for the operation to complete.

Note: This operation might take upto 15 minutes to complete.

```
[37]: # Roll the operation until it's done successfully
```

```
while True:  
    if r.done():  
        break  
    print("Poll the operation to deploy index...")  
    time.sleep(60)
```

```
Poll the operation to deploy index...
```

Retrieve the result of the operation.

```
[38]: r.result()
[38]: deployed_index {
    id: "stream_update_glove_deployed"
}
```

Create Online Queries

After you built your indexes, you may query against the deployed index through the online querying gRPC API (Match service) within the virtual machine instances from the same region.

Run the following code snippet to create write `match_service.proto` locally. This is a Protocol Buffer (protobuf) definition for a service called `MatchService`.

```
[39]: %%writefile match_service.proto
syntax = "proto3";

package google.cloud.aiplatform.container.v1;

// MatchService is a Google managed service for efficient vector similarity
// search at scale.
service MatchService {
    // Returns the nearest neighbors for the query. If it is a sharded
    // deployment, calls the other shards and aggregates the responses.
    rpc Match(MatchRequest) returns (MatchResponse) {}

    // Parameters for a match query.
    message MatchRequest {
        // The ID of the DeployedIndex that will serve the request.
        // This MatchRequest is sent to a specific IndexEndpoint of the Control API,
        // as per the IndexEndpoint.network. That IndexEndpoint also has
        // IndexEndpoint.deployed_indexes, and each such index has an
        // DeployedIndex.id field.
        // The value of the field below must equal one of the DeployedIndex.id
        // fields of the IndexEndpoint that is being called for this request.
        string deployed_index_id = 1;

        // The embedding values.
        repeated float float_val = 2;

        // The number of nearest neighbors to be retrieved from database for
        // each query. If not set, will use the default from
        // the service configuration.
        int32 num_neighbors = 3;

        // The list of restricts.
        repeated Namespace restricts = 4;

        // Crowding is a constraint on a neighbor list produced by nearest neighbor
        // search requiring that no more than some value k' of the k neighbors
        // returned have the same value of crowding_attribute.
        // It's used for improving result diversity.
        // This field is the maximum number of matches with the same crowding tag.
        int32 per_crowding_attribute_num_neighbors = 5;

        // The number of neighbors to find via approximate search before
        // exact reordering is performed. If not set, the default value from scam
        // config is used; if set, this value must be > 0.
        int32 approx_num_neighbors = 6;

        // The fraction of the number of leaves to search, set at query time allows
        // user to tune search performance. This value increase result in both search
        // accuracy and latency increase. The value should be between 0.0 and 1.0. If
        // not set or set to 0.0, query uses the default value specified in
        // NearestNeighborSearchConfig.TreeAHConfig.leaf_nodes_to_search_percent.
        int32 leaf_nodes_to_search_percent_override = 7;
    }

    // Response of a match query.
    message MatchResponse {
        message Neighbor {
            // The ids of the matches.
            string id = 1;

            // The distances of the matches.
            double distance = 2;
        }
        // All its neighbors.
        repeated Neighbor neighbor = 1;
    }
}

// Namespace specifies the rules for determining the datapoints that are
// eligible for each matching query, overall query is an AND across namespaces.
message Namespace {
    // The string name of the namespace that this proto is specifying,
    // such as "color", "shape", "geo", or "tags".
    string name = 1;

    // The allowed tokens in the namespace.
    repeated string allow_tokens = 2;

    // The denied tokens in the namespace.
    // The denied tokens have exactly the same format as the token fields, but
    // represents a negation. When a token is denied, then matches will be
    // excluded whenever the other datapoint has that token.
    //

    // For example, if a query specifies {color: red, blue, !purple}, then that
    // query will match datapoints that are red or blue, but if those points are
    // also purple, then they will be excluded even if they are red/blue.
    repeated string deny_tokens = 3;
}
```

```
    }
```

Overwriting match_service.proto

Clone the repository that contains the dependencies of match_service.proto.

```
[40]: ! git clone https://github.com/googleapis/googleapis.git
```

fatal: destination path 'googleapis' already exists and is not an empty directory.

Compile the protocol buffer, that generates the following files: match_service_pb2.py and match_service_pb2_grpc.py.

```
[41]: ! python -m grpc_tools.protoc -I=. --proto_path=./googleapis --python_out=. --grpc_python_out=. match_service.proto
```

Obtain the private Endpoint.

```
[42]: DEPLOYED_INDEX_SERVER_IP = (
    list(index_endpoint_client.list_index_endpoints(parent=PARENT))[0]
    .deployed_indexes[0]
    .private_endpoints.match_grpc_address
)
```

```
[42]: '10.51.0.14'
```

Test your query.

```
[43]: import match_service_pb2
import match_service_pb2_grpc

channel = grpc.insecure_channel("{}:10000".format(DEPLOYED_INDEX_SERVER_IP))
stub = match_service_pb2_grpc.MatchServiceStub(channel)

/opt/coda/lib/python3.10/site-packages/google/protobuf/runtime_version.py:98: UserWarning: Protobuf gencode version 5.29.0 is exactly one major version older than the runtime version 6.31.1 at match_service.proto. Please update the gencode to avoid compatibility violations in the next runtime release.
  warnings.warn(
```

```
[44]: # Test query
```

```
query = [
```

```
    -0.11333,
```

```
    0.48402,
```

```
    0.09071,
```

```
    -0.22439,
```

```
    0.034206,
```

```
    -0.55831,
```

```
    0.041849,
```

```
    -0.53573,
```

```
    0.18809,
```

```
    -0.58722,
```

```
    0.015313,
```

```
    -0.014555,
```

```
    0.80842,
```

```
    -0.038519,
```

```
    0.75348,
```

```
    0.70502,
```

```
    -0.17863,
```

```
    0.3222,
```

```
    0.67575,
```

```
    0.67198,
```

```
    0.26044,
```

```
    0.4187,
```

```
    -0.34122,
```

```
    0.2286,
```

```
    -0.53529,
```

```
    1.2582,
```

```
    -0.091543,
```

```
    0.19716,
```

```
    -0.037454,
```

```
    -0.3336,
```

```
    0.31399,
```

```
    0.36488,
```

```
    0.71263,
```

```
    0.1307,
```

```
    -0.24654,
```

```
    -0.52445,
```

```
    -0.036091,
```

```
    0.55068,
```

```
    0.10017,
```

```
    0.48095,
```

```
    0.71104,
```

```
    -0.053462,
```

```
    0.22325,
```

```
    0.30917,
```

```
    -0.39926,
```

```
    0.036634,
```

```
    -0.35431,
```

```
    -0.42795,
```

```
    0.46444,
```

```
    0.25586,
```

```
    0.68257,
```

```
    -0.20821,
```

```
    0.38433,
```

```
    0.055773,
```

```
    -0.2539,
```

```
    -0.20804,
```

```
    0.52522,
```

```
    -0.11399,
```

```
    -0.3253,
```

```
    -0.44104,
```

```
    0.17528,
```

```
    0.62255,
```

```
    0.50237,
```

```
    -0.7607,
```

```
    -0.071786,
```

```
    0.0080131,
```

```
    -0.13286,
```

```
    0.50097,
```

```
    0.18824,
```

```
    -0.54722,
```

```
    -0.42664,
```

```
    0.4292,
```

```
    0.14877,
```

```
    -0.0072514,
```

```
    -0.16484,
```

```
    -0.059798,
```

```
    0.9895,
```

```
    -0.61738,
```

```
    0.054169,
    0.48424,
    -0.35084,
    -0.27053,
    0.37829,
    0.11503,
    -0.39613,
    0.24266,
    0.39147,
    -0.075256,
    0.65093,
    -0.20822,
    -0.17456,
    0.53571,
    -0.16537,
    0.13582,
    -0.56016,
    0.016964,
    0.1277,
    0.94071,
    -0.22608,
    -0.021106,
]
```

```
[45]: request = match_service_pb2.MatchRequest()
request.deployed_index_id = DEPLOYED_INDEX_ID
for val in query:
    request.float_val.append(val)

# The output before stream update
response = stub.Match(request)
response
```

```
[45]: neighbor {
        id: "0"
        distance: 17.592369079589844
    }
neighbor {
        id: "31"
        distance: 14.614908218383789
    }
neighbor {
        id: "50"
        distance: 11.242000579833984
    }
neighbor {
        id: "42"
        distance: 10.925321578979492
    }
neighbor {
        id: "46"
        distance: 10.185911178588867
    }
neighbor {
        id: "100"
        distance: 10.031323432922363
    }
neighbor {
        id: "71"
        distance: 9.460129737854004
    }
neighbor {
        id: "64"
        distance: 9.329634666442871
    }
neighbor {
        id: "54"
        distance: 9.25944709777832
    }
neighbor {
        id: "98"
        distance: 8.94312858581543
    }
```

Insert datapoints.

```
[46]: insert_datapoints_payload = aiplatform_v1.IndexDatapoint(
    datapoint_id="101",
    feature_vector=query,
    restricts=[{"namespace": "class", "allow_list": ["101"]}],
    crowding_tag=aiplatform_v1.IndexDatapoint.CrowdingTag(crowding_attribute="b"),
)
```

```
upsert_request = aiplatform_v1.UpsertDatapointsRequest(
    index=INDEX_RESOURCE_NAME, datapoints=[insert_datapoints_payload]
)
```

```
index_client.upsert_datapoints(request=upsert_request)
```

```
request = match_service_pb2.MatchRequest()
request.deployed_index_id = DEPLOYED_INDEX_ID
for val in query:
    request.float_val.append(val)

# The new inserted datapoint with id 101 will show up in the output
response = stub.Match(request)
response
```

```
[46]: neighbor {
        id: "0"
        distance: 17.592369079589844
    }
neighbor {
        id: "31"
        distance: 14.614908218383789
    }
neighbor {
        id: "50"
        distance: 11.242000579833984
    }
neighbor {
        id: "42"
        distance: 10.925321578979492
    }
neighbor {
        id: "46"
        distance: 10.185911178588867
    }
```

```

neighbor {
    id: "100"
    distance: 10.031323432922363
}
neighbor {
    id: "71"
    distance: 9.460129737854004
}
neighbor {
    id: "64"
    distance: 9.329634666442871
}
neighbor {
    id: "54"
    distance: 9.25944709777832
}
neighbor {
    id: "98"
    distance: 8.9431285851543
}

```

Add filtering.

```
[47]: request = match_service_pb2.MatchRequest()
request.deployed_index_id = DEPLOYED_INDEX_ID
for val in query:
    request.float_val.append(val)

# Only the datapoints whose id is 1 and 101 will show up in the output
restrict = match_service_pb2.Namespace()
restrict.name = "class"
restrict.allow_tokens.append("1")
restrict.allow_tokens.append("101")

request.restricts.append(restrict)

response = stub.Match(request)
response
```

```
[47]: neighbor {
    id: "101"
    distance: 17.592369079589844
}
neighbor {
    id: "1"
    distance: 2.4347081184387207
}
```

Update datapoint filtering:

```
[48]: update_datapoints_payload = aiplatform_v1.IndexDatapoint(
    datapoint_id="101",
    feature_vector=query,
    restricts=[{"namespace": "class", "allow_list": ["102"]}],
    crowding_tag=aiplatform_v1.IndexDatapoint.CrowdingTag(crowding_attribute="b"),
)

upsert_request = aiplatform_v1.UpsertDatapointsRequest(
    index=INDEX_RESOURCE_NAME, datapoints=[update_datapoints_payload]
)

index_client.upsert_datapoints(request=upsert_request)

response = stub.Match(request)
response
```

```
[48]: neighbor {
    id: "1"
    distance: 2.4347081184387207
}
```

Add crowding.

```
[49]: request = match_service_pb2.MatchRequest()
request.deployed_index_id = DEPLOYED_INDEX_ID
for val in query:
    request.float_val.append(val)

# Set the limit of the number of neighbors in each crowding to 1
# So no more than one neighbor of each crowding group will appear in the output
request.per_crowding_attribute_num_neighbors = 1

response = stub.Match(request)
response
```

```
[49]: neighbor {
    id: "0"
    distance: 17.592369079589844
}
neighbor {
    id: "101"
    distance: 17.592369079589844
}
```

Update datapoint crowding.

```
[50]: # Change the crowding_attribute from 'b' to 'a' for the datapoint with id '101' by using stream update
update_datapoints_payload = aiplatform_v1.IndexDatapoint(
    datapoint_id="101",
    feature_vector=query,
    restricts=[{"namespace": "class", "allow_list": ["101"]}],
    crowding_tag=aiplatform_v1.IndexDatapoint.CrowdingTag(crowding_attribute="a"),
)

upsert_request = aiplatform_v1.UpsertDatapointsRequest(
    index=INDEX_RESOURCE_NAME, datapoints=[update_datapoints_payload]
)

index_client.upsert_datapoints(request=upsert_request)

response = stub.Match(request)
response
```

```
[50]: neighbor {
    id: "0"
    distance: 17.592369079589844
}
...
```

```
neighbor {
    id: "31"
    distance: 14.614908218383789
}
```

Remove datapoints.

```
[51]: # Remove the datapoint with id '101' from the index
remove_request = aiplatform_v1.RemoveDatapointsRequest(
    index=INDEX_RESOURCE_NAME, datapoint_ids=["101"]
)

index_client.remove_datapoints(request=remove_request)
```

```
request = match_service_pb2.MatchRequest()
request.deployed_index_id = DEPLOYED_INDEX_ID
for val in query:
    request.float_val.append(val)

response = stub.Match(request)
response
```

```
neighbor {
    id: "0"
    distance: 17.592369079589844
}
neighbor {
    id: "101"
    distance: 17.592369079589844
}
neighbor {
    id: "31"
    distance: 14.614908218383789
}
neighbor {
    id: "50"
    distance: 11.242000579833984
}
neighbor {
    id: "42"
    distance: 10.925321578979492
}
neighbor {
    id: "46"
    distance: 10.185911178588867
}
neighbor {
    id: "100"
    distance: 10.031323432922363
}
neighbor {
    id: "71"
    distance: 9.460129737854004
}
neighbor {
    id: "64"
    distance: 9.329634666442871
}
neighbor {
    id: "54"
    distance: 9.25944709777832
}
```

[1]:

▶ ⏪ ⏹ ⏺ ⏻ ⏸ ⏹ ⏷