1. Create a folder to store data
2. Run DB_data_creation.ipynb to generate the data

Loading dataset...

```
input_matrix_path = "C:/CSE3800/Normal_6_raw/CN4/"
```

Input matrix path can be changed to generate data for different samples

```
directory = "C:/CSE3800/TestStuff/"
```

Make sure to change the destination directory to the correct folder

Name	Date modified	Type
CN4	12/20/2024 12:55 AM	File folder
OA6	12/20/2024 12:38 AM	File folder

Each time the notebook is run, move all generated csv files into a new subfolder for the sample that was clustered

Setting up a virtual environment
https://www.geeksforgeeks.org/create-virtual-environment-using-venv-python/

If a permission error occurs when trying to activate the virtual environment run the following command in Powershell (command temporarily gives permission)

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

This keeps any libraries installed limited to this particular virtual environment

Create Database

```
pip install mysql-connector-python
```

May need to install mysql connector

A database has to be created before sql.py is run. The code can be altered to do this with in the same file by commenting and uncommenting certain lines.

sql.py

```python
import mysql.connector
import pandas as pd
import numpy as np
import os


mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="INSERT PASSWORD", # change to correct pw
)

mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE [INSERT DB NAME]") # set db name to whatever you want to
```

This should be all the code that is needed for this part

Set the correct information to connect to MySQL

Also set DB name

Creating Tables

Don't try to create another database when generating tables (recomment out line)

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="INSERT PASSWORD", # 
    database="INSERT DB NAME" # cha
)
```

When connecting to MySQL this time, make sure to specify the DB

Change directory to where data is stored

```
directory = 'C:/CSE3800/ReportData/' # change path as needed
# mycursor.execute("CREATE TABLE Samples (Sample_ID INT PRIMARY KEY AUTO_INCREMENT, Sample VARCHAR(255) NOT NULL);")
```

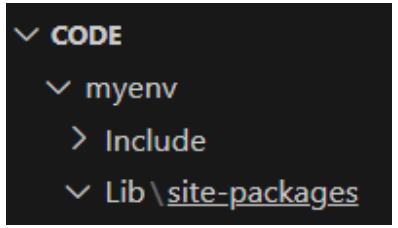
Uncomment this line to create the sample table. This stores the names of the different samples (ex. OA6, CN4, etc)

Make sure to pip install SQLAlchemy if not already installed

Setting up environment

https://docs.llamaindex.ai/en/stable/getting_started/installation/

```
pip install llama-index
```



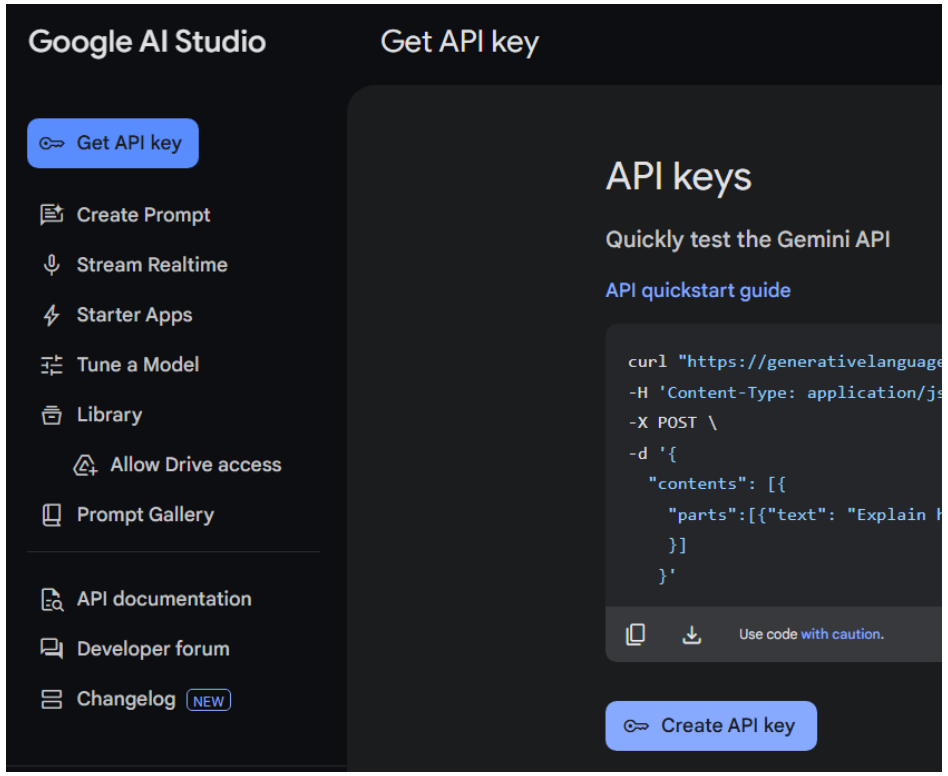
Navigate to Lib/sitepackages in your virtual env folder to see what files were installed

My install was missing some stuff from llama_index/core so I manually copy and pasted the correct files from the GitHub: https://github.com/run-llama/llama_index/tree/main/llama-index-core/llama_index/core

```
from llama_index.core.utilities.sql_wrapper import SQLDatabase
from llama_index.indices.struct_store.sql_query import SQLTableRetrieverQueryEngine
from llama_index.indices.vector_store.base import VectorStoreIndex
from llama_index.core.objects import (
    SQLTableNodeMapping,
    ObjectIndex,
    SQLTableSchema,
)
from llama_index.core import Settings
from llama_index.llms.gemini import Gemini
from llama_index.embeddings.gemini import GeminiEmbedding
```

The file locations for the modules in sitepackages can be seen from the import statements

Using Gemini https://docs.llamaindex.ai/en/stable/examples/llm/gemini/




```
pip install llama-index-llms-gemini llama-index
```

Navigate to Google AI Studio ->
Get API Key -> Create API Key

Copy and paste key into
proper place in nlp.py

```
Settings.embed_model = GeminiEmbedding(model='models/embedding-001')
Settings.llm = Gemini(model="models/gemini-1.5-flash")
```

Can change specific model here

```
query = "How many clusters do each sample have?"
```

Try different queries

```
(myenv) PS C:\CSE3800\Code> python nlp.py
```

Run program to see results