

# National University of Computer & Emerging Sciences



## MLOps

Semester Project: Environmental Monitoring and Pollution Prediction System

*Instructor: Dr. Hammad Majeed*

**21i-0572 Kissa Zahra**

**Submission Date:** 15/12/2024

# Table of Content

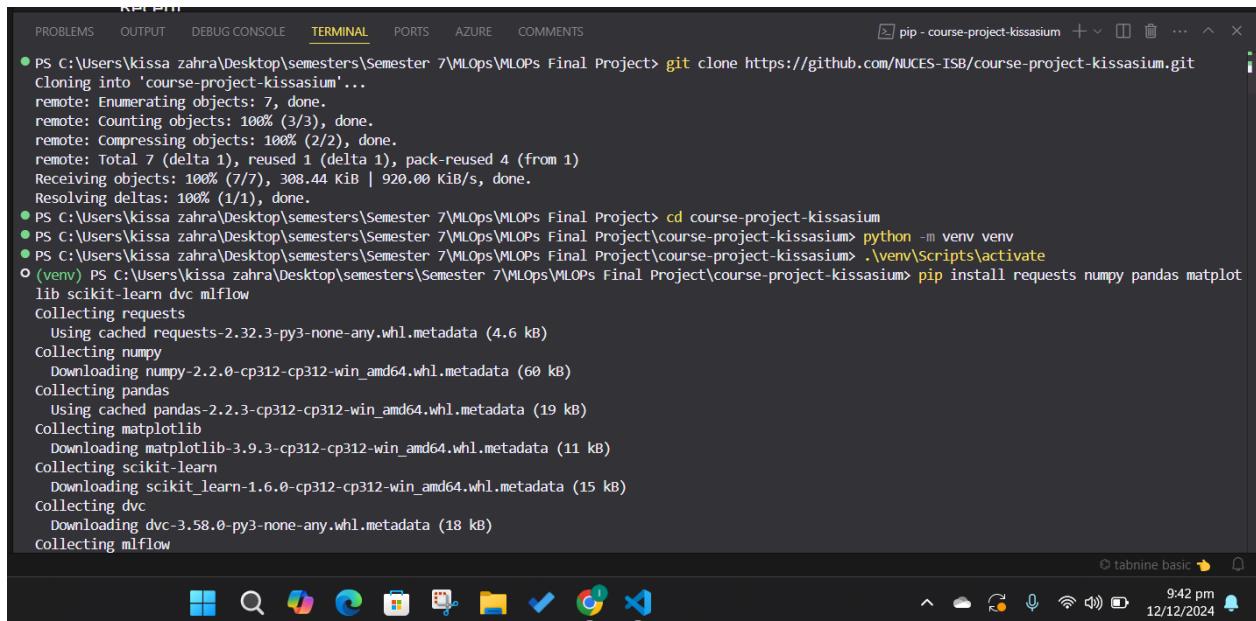
<b>Task 1 Managing Environmental Data with DVC.....</b>	<b>2</b>
1. Research Live Data Streams.....	3
2. Set Up DVC Repository.....	3
3. Remote Storage Configuration.....	5
4. Data Collection Script.....	6
5. Version Control with DVC.....	13
6. Automate Data Collection.....	21
7. Update Data with DVC.....	24
<b>Task 2 Pollution Trend Prediction with MLflow.....</b>	<b>27</b>
1. Data Preparation.....	27
ARIMA (AutoRegressive Integrated Moving Average) for AQI Forecasting.....	27
1. Splitting the Data.....	27
2. Model Training and Forecasting.....	27
3. Evaluation.....	28
Output with ARIMA.....	28
Random Forest Model.....	29
1. Splitting the Data.....	29
2. Model Training and Forecasting.....	29
3. Evaluation.....	30
Output with Random Forest Model.....	30
3. Train Models with MLflow.....	31
ARIMA.....	33
Random Forest Model.....	34
4. Hyperparameter Tuning.....	35
ARIMA.....	35
Random Forest Model.....	37
5. Model Evaluation.....	39
Information about my model.....	40
6. Deployment.....	41
<b>Task 3: Monitoring and Live Testing.....</b>	<b>42</b>
1. Set Up Monitoring.....	42
2. Test Predictions with Live Data.....	48
3. Analyze and Optimize.....	49
<b>Summary report on the system's live performance.....</b>	<b>49</b>

# Task 1 Managing Environmental Data with DVC

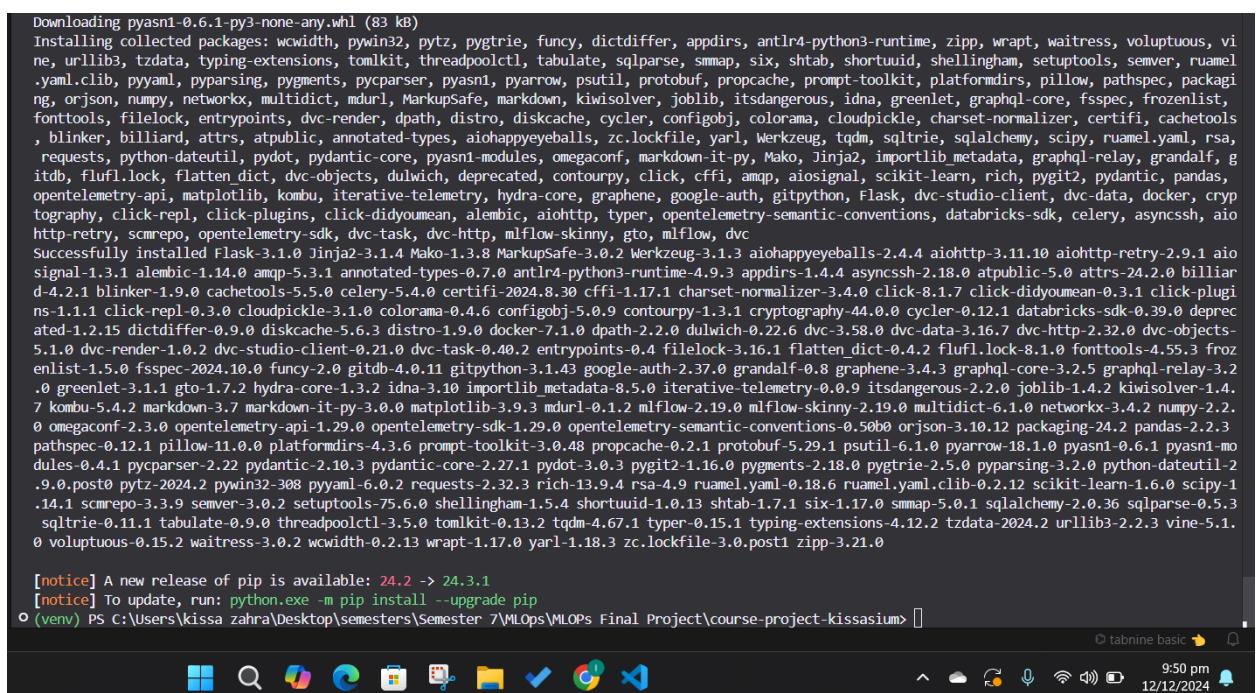
Objective: Use DVC to manage real-time environmental data streams collected from APIs.

**Clone the repo and run these commands on the terminal to create virtual environment.**

```
` python -m venv venv`  
` .\venv\Scripts\activate`  
` pip install requests numpy pandas matplotlib scikit-learn dvc mlflow`
```



```
PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project> git clone https://github.com/NUCES-ISB/course-project-kissassium.git  
Cloning into 'course-project-kissassium'...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 7 (delta 1), reused 1 (delta 1), pack-reused 4 (from 1)  
Receiving objects: 100% (7/7), 308.44 KiB | 920.00 KiB/s, done.  
Resolving deltas: 100% (1/1), done.  
PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project> cd course-project-kissassium  
PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissassium> python -m venv venv  
PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissassium> .\venv\Scripts\activate  
○ (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissassium> pip install requests numpy pandas matplotlib  
lib scikit-learn dvc mlflow  
Collecting requests  
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)  
Collecting numpy  
  Downloading numpy-2.2.0-cp312-cp312-win_amd64.whl.metadata (60 kB)  
Collecting pandas  
  Using cached pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)  
Collecting matplotlib  
  Downloading matplotlib-3.9.3-cp312-cp312-win_amd64.whl.metadata (11 kB)  
Collecting scikit-learn  
  Downloading scikit_learn-1.6.0-cp312-cp312-win_amd64.whl.metadata (15 kB)  
Collecting dvc  
  Downloading dvc-3.58.0-py3-none-any.whl.metadata (18 kB)  
Collecting mlflow
```



```
Downloading pyasn1-0.6.1-py3-none-any.whl (83 kB)  
Installing collected packages: wcwidth, pywin32, pytz, pygtrie, funcy, dictdiffer, appdirs, antlr4-python3-runtime, zipp, wrapt, waitress, voluptuous, vi  
ne, urllib3, tzdata, typing-extensions, tomllib, threadpoolctl, tabulate, sqlparse, smmap, six, shtab, shortuuid, shellingham, setuptools, semver, ruamel  
.yaml.lib, pyyaml, pyparsing, pygments, pycparser, pyasn1, pyarrow, psutil, protobuf, propcache, prompt-toolkit, platformdirs, pillow, pathspec, packagi  
ng, orjson, numpy, networkx, multidict, mdurl, MarkupSafe, markdown, kiwisolver, joblib, itsdangerous, idna, greenlet, graphql-core, fsspec, frozenlist,  
fonttools, filelock, entrypoints, dvc-render, dpath, distro, diskcache, cycler, configobj, colorama, cloudpickle, charset-normalizer, certifi, cachetools  
, blinker, billiard, attrs, atpublic, annotated-types, aiohappyeyeballs, zc.lockfile, yarl, Werkzeug, tqdm, sqltrie, sqlalchemy, scipy, ruamel.yaml, rsa,  
requests, python-dateutil, pydot, pydantic-core, pyasn1-modules, omegaconf, markdown-it-py, Mako, Jinja2, importlib_metadata, graphql-relay, grandalf, g  
itdb, flufl.lock, flatten dict, dvc-objects, dulwich, deprecated, contourpy, click, cfpi, amqp, aiosignal, scikit-learn, rich, pygit2, pydantic, pandas,  
opentelemetry-api, matplotlib, kombu, iterative-telemetry, hydra-core, graphene, google-auth, gitpython, Flask, dvc-studio-client, dvc-data, docker, cryp  
tography, click-repl, click-plugins, click-didyoumean, alembic, aiohttp, typer, opentelemetry-semantic-conventions, databricks-sdk, celery, asyncssh, aio  
http-retry, scmrepo, opentelemetry-sdk, dvc-task, dvc-http, mlflow-skinny, gto, mlflow, dvc  
Successfully installed Flask-3.1.0 Jinja2-3.1.4 Mako-1.3.8 MarkupSafe-3.0.2 Werkzeug-3.1.3 aiohappyeyeballs-2.4.4 aiohttp-3.11.10 aiohttp-retry-2.9.1 aio  
signal-1.3.1 alembic-1.14.0 amqp-5.3.1 annotated-types-0.7.0 antlr4-python3-runtime-4.9.3 appdirs-1.4.4 asyncssh-2.18.0 atpublic-5.0 attrs-24.2.0 billiar  
d-4.2.1 blinker-1.9.0 cachetools-5.5.0 celery-5.4.0 certifi-2024.8.30 cffi-1.17.1 charset-normalizer-3.4.0 click-8.1.7 click-didyoumean-0.3.1 click-plug  
ins-1.1.1 click-repl-0.3.0 cloudpickle-3.1.0 colorama-0.4.6 configobj-5.0.9 contourpy-1.3.1 cryptography-44.0.0 cypher-0.12.1 databricks-sdk-0.39.0 deprec  
ated-1.2.15 dictdiffer-0.9.0 diskcache-5.6.3 distro-1.9.0 docker-7.1.0 dpath-2.2.0 dulwich-0.22.6 dvc-3.58.0 dvc-data-3.16.7 dvc-http-2.32.0 dvc-objects-  
5.1.0 dvc-render-1.0.2 dvc-studio-client-0.21.0 dvc-task-0.40.2 entrypoints-0.4 filelock-3.16.1 flatten_dict-0.4.2 flufl.lock-8.1.0 fonttools-4.55.3 froz  
enlist-1.5.0 fsspec-2024.10.0 funcy-2.0 gitdb-4.0.11 gitpython-3.1.43 google-auth-2.37.0 grandalf-0.8 graphene-3.4.3 graphql-core-3.2.5 graphql-relay-3.2  
.0 greenlet-3.1.1 gto-1.7.2 hydra-core-1.3.2 idna-3.10 importlib_metadata-8.5.0 iterative-telemetry-0.9.0 itsdangerous-2.2.0 joblib-1.4.2 kiwisolver-1.4,  
7 kombu-5.4.2 markdown-3.7 markdown-it-py-3.0.0 matplotlib-3.9.3 mdurl-0.1.2 mlflow-2.19.0 mlflow-skinny-2.19.0 multidict-6.1.0 networkx-3.4.2 numpy-2.2.  
0 omegaconf-2.3.0 opentelemetry-api-1.29.0 opentelemetry-sdk-1.29.0 opentelemetry-semantic-conventions-0.50b0 orjson-3.10.12 packaging-24.2 pandas-2.2.3  
pathspec-0.12.1 pillow-11.0.0 platformdirs-4.3.6 prompt-toolkit-3.0.48 propcache-0.2.1 protobuf-5.29.1 psutil-6.1.0 pyarrow-18.1.0 pyasn1-0.6.1 pyasn1-mo  
dules-0.4.1 pycparser-2.22 pydantic-2.10.3 pydantic-core-2.27.1 pydot-3.0.3 pygit2-1.16.0 pygments-2.18.0 pygtrie-2.5.0 pyparsing-3.2.0 python-dateutil-2  
.9.0.post0 pytz-2024.2 pywin32-308 pyyaml-6.0.2 requests-2.32.3 rich-13.9.4 rsa-4.9 ruamel.yaml-0.18.6 ruamel.yaml.lib-0.2.12 scikit-learn-1.6.0 scipy-1  
.14.1 scmrepo-3.3.9 semver-3.0.2 setuptools-75.6.0 shellingham-1.5.4 shortuuid-1.0.13 shtab-1.7.1 six-1.17.0 smmap-5.0.1 sqlalchemy-2.0.36 sqlparse-0.5.3  
sqltrie-0.11.1 tabulate-0.9.0 threadpoolctl-3.5.0 tomllib-0.13.2 tqdm-4.67.1 typer-0.15.1 typing-extensions-4.12.2 tzdata-2024.2 urllib3-2.2.3 vine-5.1.  
0 voluptuous-0.15.2 waitress-3.0.2 wctwidth-0.2.13 wrapt-1.17.0 yarl-1.18.3 zc.lockfile-3.0.post1 zipp-3.21.0  
  
[notice] A new release of pip is available: 24.2 -> 24.3.1  
[notice] To update, run: python.exe -m pip install --upgrade pip  
○ (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissassium>
```

# 1. Research Live Data Streams

**OpenWeatherMap API:** Provides weather data, including temperature, humidity, and air pollution data (<https://openweathermap.org/api>).

# 2. Set Up DVC Repository

## install dvc

```
[notice] To update, run: python.exe -m pip install --upgrade pip
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps Final Project\course-project-kissarium> pip install dvc
Requirement already satisfied: dvc in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (3.58.0)
Requirement already satisfied: attrs>=22.2.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (24.2.0)
Requirement already satisfied: celery in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (5.4.0)
Requirement already satisfied: colorama>=0.3.9 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (0.4.6)
Requirement already satisfied: configobj>=5.0.9 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (5.0.9)
Requirement already satisfied: distro>=1.3 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (1.9.0)
Requirement already satisfied: dpath<3,>=2.1.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (2.2.0)
Requirement already satisfied: dulwich in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (0.22.6)
Requirement already satisfied: dvc-data<3.17,>=3.16.2 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (3.16.7)
Requirement already satisfied: dvc-http>=2.29.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (2.32.0)
Requirement already satisfied: dvc-objects in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (5.1.0)
Requirement already satisfied: dvc-render<2,>=1.0.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (1.0.2)
Requirement already satisfied: dvc-studio-client<1,>=0.21 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (0.21.0)
Requirement already satisfied: dvc-task<1,>=0.3.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (0.40.2)
Requirement already satisfied: flatten_dict<1,>=0.4.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (0.4.2)
Requirement already satisfied: fluff.lock<9,>=8.1.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (8.1.0)
Requirement already satisfied: fsspec>=2024.2.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (2024.10.0)
Requirement already satisfied: funcy>=1.14 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissarium\venv\lib\site-packages (from dvc) (1.14.0)
```

## dvc init

The screenshot shows the VS Code interface with the terminal tab selected. The command `dvc init` is being run in a PowerShell window. The output shows several dependency requirements and a notice about a new pip release. A callout box highlights a message about DVC enabling anonymous aggregate usage analytics.

```
Requirement already satisfied: frozenlist<=1.1.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from aiohttp>aiohttp-retry>2.5.0->dvc>http>2.29.0->dvc) (1.5.0)
Requirement already satisfied: multidict<2.0,>>4.5 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from aiohttp>aiohttp-retry>2.5.0->dvc>http>2.29.0->dvc) (6.1.0)
Requirement already satisfied: propcache<0.2.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from aiohttp>aiohttp-retry>2.5.0->dvc>http>2.29.0->dvc) (0.2.1)
Requirement already satisfied: yarl<2.0,>>1.17.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from aiohttp>aiohttp-retry>2.5.0->dvc>http>2.29.0->dvc) (1.18.3)
Requirement already satisfied: pyparser<1.14.0,>>3.0.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from aiohttp>aiohttp-retry>2.5.0->dvc>http>2.29.0->dvc) (2.22)
Requirement already satisfied: smmap<6,>>3.0.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from gitobcs,>>0.1.>gipython>>3>smmap>4,>>3.3.8->dvc) (5.0.1)
Requirement already satisfied: prompt_toolkit<>3.0.36->click-repl>0.2.0->xelery->dvc) (0.2.13)

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\semester 7\mlops\mlops Final Project\course-project-kissasium> dvc init
Initialized DVC repository.

You can now commit the changes to git.

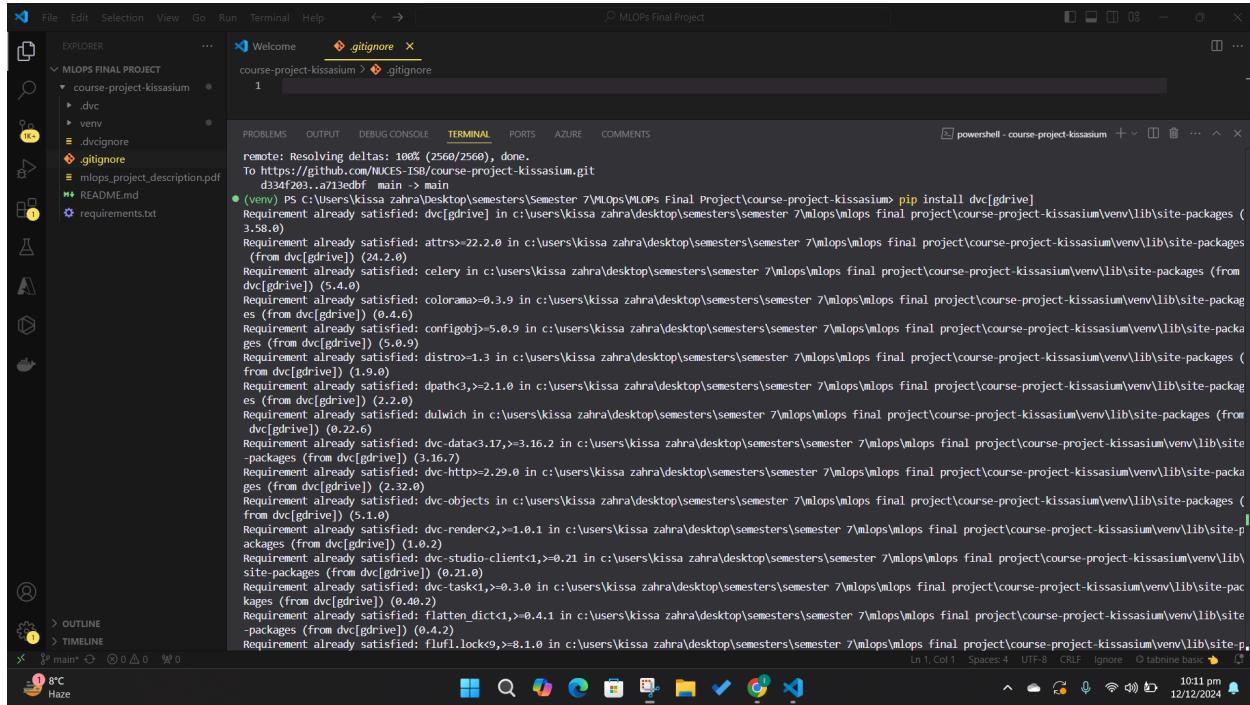
DVC has enabled anonymous aggregate usage analytics.
Read the analytics documentation (and how to opt-out) here:
<https://dvc.org/doc/user-guide/analytics>
```

## After initializing the dvc repo, I pushed it on github

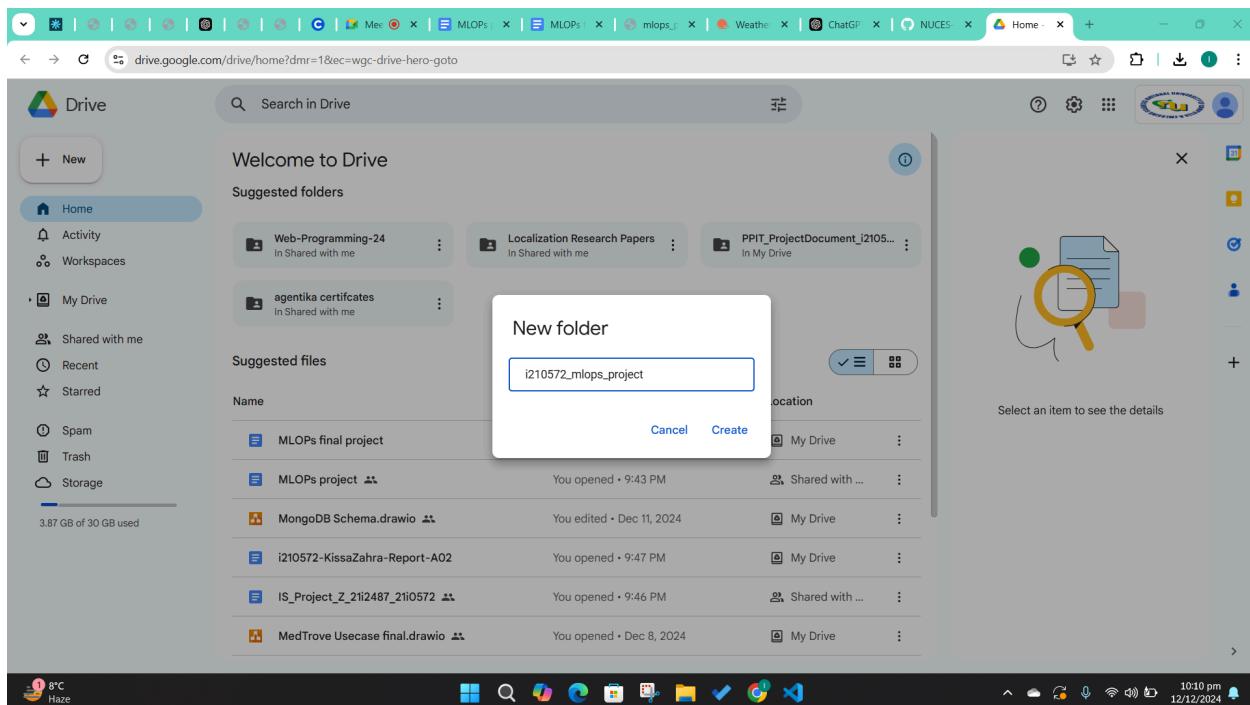
The screenshot shows the VS Code interface with the terminal tab selected. The command `git push` is being run in a PowerShell window. The output shows the cloning of the repository, enumeration of objects, compression, writing of objects, and resolution of deltas. A callout box highlights the URL of the GitHub repository.

```
create mode 100644 venv\scripts\pygrun
create mode 100644 venv\scripts\pvrsa-decrypt.exe
create mode 100644 venv\Scripts\pvrsa-encrypt.exe
create mode 100644 venv\scripts\pvrsa-keygen.exe
create mode 100644 venv\Scripts\pvrsa-prv2pub.exe
create mode 100644 venv\scripts\pvrsa-sign.exe
create mode 100644 venv\Scripts\pvrsa-verify.exe
create mode 100644 venv\scripts\pysemer.exe
create mode 100644 venv\scripts\python.exe
create mode 100644 venv\scripts\pythone.exe
create mode 100644 venv\scripts\pywin2_postinstall.py
create mode 100644 venv\scripts\pywin2_testall.py
create mode 100644 venv\Scripts\shortwid.exe
create mode 100644 venv\Scripts\shstab.exe
create mode 100644 venv\Scripts\sqlformat.exe
create mode 100644 venv\Scripts\tabulate.exe
create mode 100644 venv\Scripts\tygh.exe
create mode 100644 venv\Scripts\tx.exe
create mode 100644 venv\Scripts\typer.exe
create mode 100644 venv\Scripts\witness-server.exe
create mode 100644 venv\pyenv.cfg
create mode 100644 venv\share\man\man1.txt
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\semester 7\mlops\mlops Final Project\course-project-kissasium> git branch
* main
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\semester 7\mlops\mlops Final Project\course-project-kissasium> git push
Enumerating objects: 28168, done.
Counting objects: 100% (28168/28168), done.
Delta compression using up to 8 threads
Compressing objects: 100% (25294/25294), done.
Writing objects: 100% (28167/28167), 225.39 MiB | 160.28 MiB/s, done.
Total 28167 (delta 2560), reused 28167 (delta 2560), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2560/2560), done.
To https://github.com/NCEES-ISB/course-project-kissasium.git
    d334f203..a713edb main -> main
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\semester 7\mlops\mlops Final Project\course-project-kissasium>
```

### 3. Remote Storage Configuration



```
remote: Resolving deltas: 100% (2560/2560), done.
To https://github.com/NUICES-ISR/course-project-kissasium.git
Requirement already satisfied: dvc[gdrive] in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (3.58.0)
Requirement already satisfied: attrs>=22.2.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (24.2.0)
Requirement already satisfied: celery in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (5.4.0)
Requirement already satisfied: colorama>=0.3.9 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (0.4.6)
Requirement already satisfied: configobj>=5.0.9 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (5.0.9)
Requirement already satisfied: distro>=1.3 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (1.9.0)
Requirement already satisfied: dpath<3,>=2.1.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (2.2.0)
Requirement already satisfied: dulwich in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (0.22.6)
Requirement already satisfied: dvc-data<3.17,>=3.16.2 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (3.16.7)
Requirement already satisfied: dvc-http>=2.29.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (2.32.0)
Requirement already satisfied: dvc-objects in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (5.1.0)
Requirement already satisfied: dvc-render<2,>=1.0.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (1.0.2)
Requirement already satisfied: dvc-studio-client<1,>=0.21 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (0.21.0)
Requirement already satisfied: dvc-task<1,>=0.3.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (0.40.2)
Requirement already satisfied: flatten_dict<1,>=0.4.1 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (0.4.2)
Requirement already satisfied: fluff.lock9,>=8.1.0 in c:\users\kissa zahra\desktop\semesters\semester 7\mlops\mlops final project\course-project-kissasium\venv\lib\site-packages (from dvc[gdrive]) (8.1.0)
```

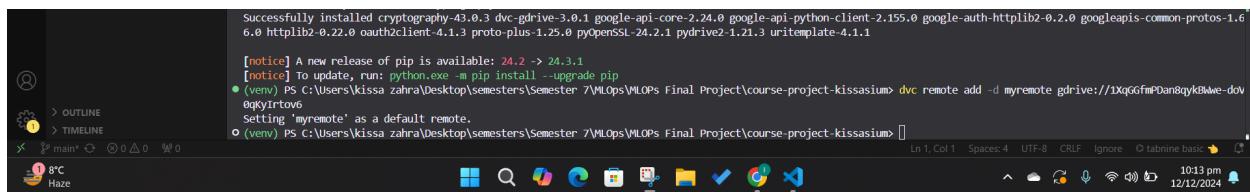


## Full url of my folder:

<https://drive.google.com/drive/folders/1XqGGfmPDan8qykBWwe-doV0qKylrtov6?dmr=1&ec=wgc-drive-hero-goto>

ID of the folder: 1XqGGfmPDan8qykBWwe-doV0qKylrtov6

My remote drive: dvc remote add -d myremote  
gdrive://1XqGGfmPDan8qykBWwe-doV0qKylrtov6

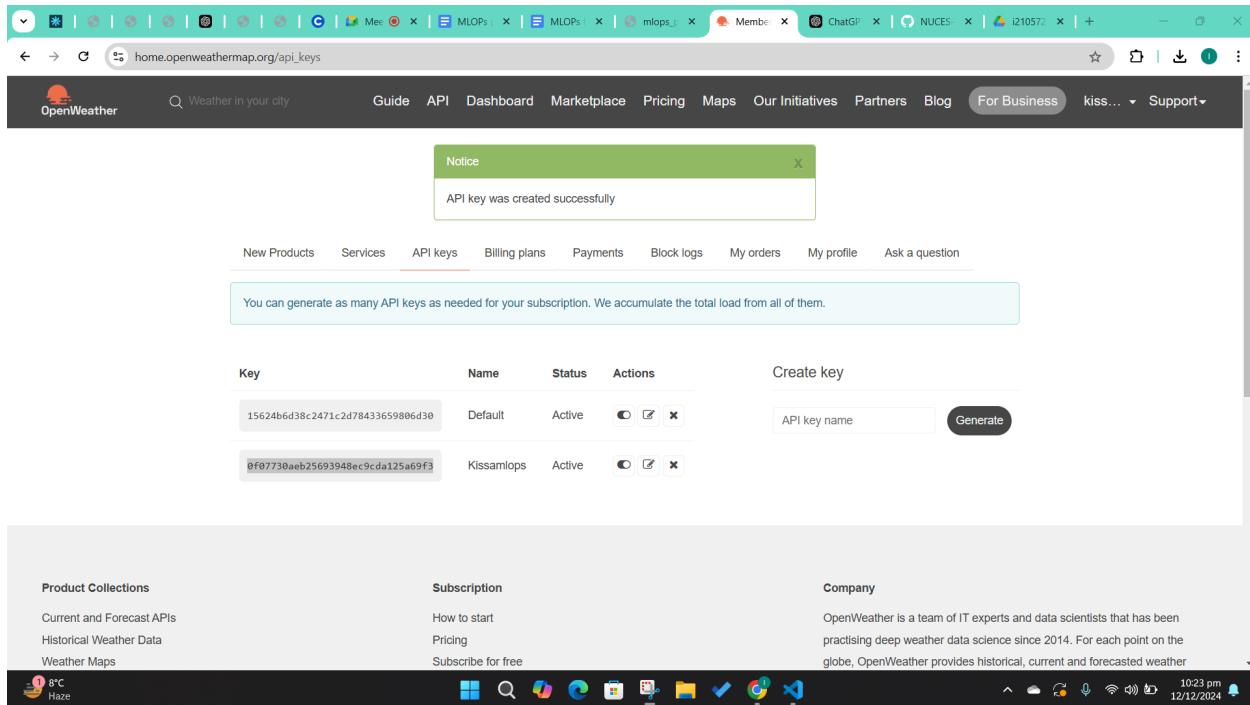


```
Successfully installed cryptography-43.0.3 dvc-gdrive-3.0.1 google-api-core-2.24.0 google-api-python-client-2.155.0 google-auth-httplib2-0.2.0 googleapis-common-protos-1.6.0 httplib2-0.22.0 oauth2client-4.1.3 proto-plus-1.25.0 pyOpenSSL-24.2.1 pydrive2-1.21.3 uritemplate-4.1.1

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPS Final Project\course-project-kissassium> dvc remote add -d myremote gdrive://1XqGGfmPDan8qykBWwe-doV0qKylrtov6
Setting "myremote" as a default remote.
○ (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPS Final Project\course-project-kissassium>
```

## 4. Data Collection Script

### Creating keys



Notice

API key was created successfully

New Products Services API keys Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions	Create key
15624b6d38c2471c2d78433659806d30	Default	Active		<input type="text" value="API key name"/>
0f07730aeb25693948ec9cda125a69f3	Kissamlops	Active		

Product Collections

- Current and Forecast APIs
- Historical Weather Data
- Weather Maps

Subscription

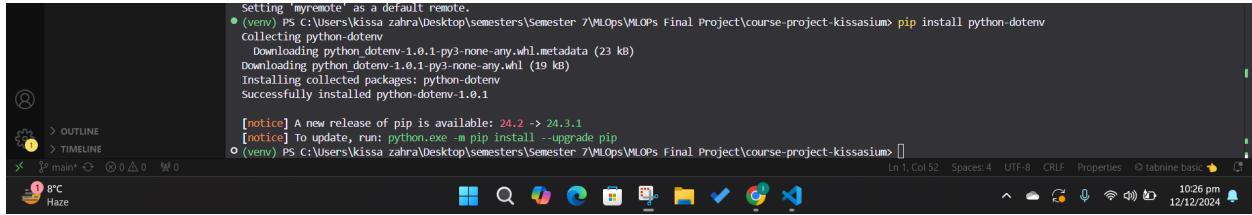
- How to start
- Pricing
- Subscribe for free

Company

OpenWeather is a team of IT experts and data scientists that has been practising deep weather data science since 2014. For each point on the globe, OpenWeather provides historical, current and forecasted weather

I put my key in the .env file

To run .env file: `pip install python-dotenv`



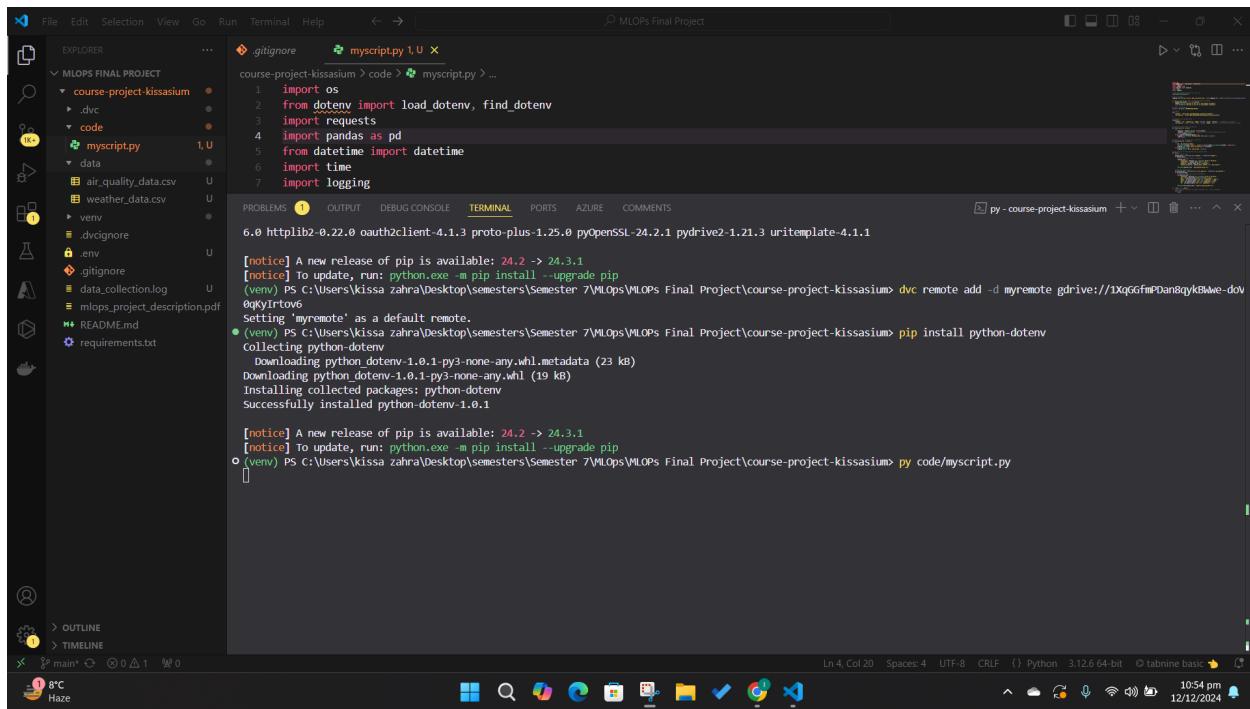
```
Setting 'myremote' as a default remote.
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissassium> pip install python-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.1
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissassium>
```

I made 2 folders; code and data.

In the code folder I put my code in code/data\_collection.py, it automatically collects weather and air quality data from the OpenWeatherMap API. It fetches the weather data including temperature, humidity, and weather conditions ,and the air quality data, including AQI and pollutants like PM2.5 and NO2. It fetches this data every 5 minutes for London, UK (I can change the location) and saves it in two separate CSV files: **weather\_data.csv** and **air\_quality\_data.csv**. The data is appended with each fetch and logs are maintained in log file (**data\_collection.log**).

The code runs continuously in the background collecting and saving data at regular intervals and logs any issues or successes in a **data\_collection.log** file.

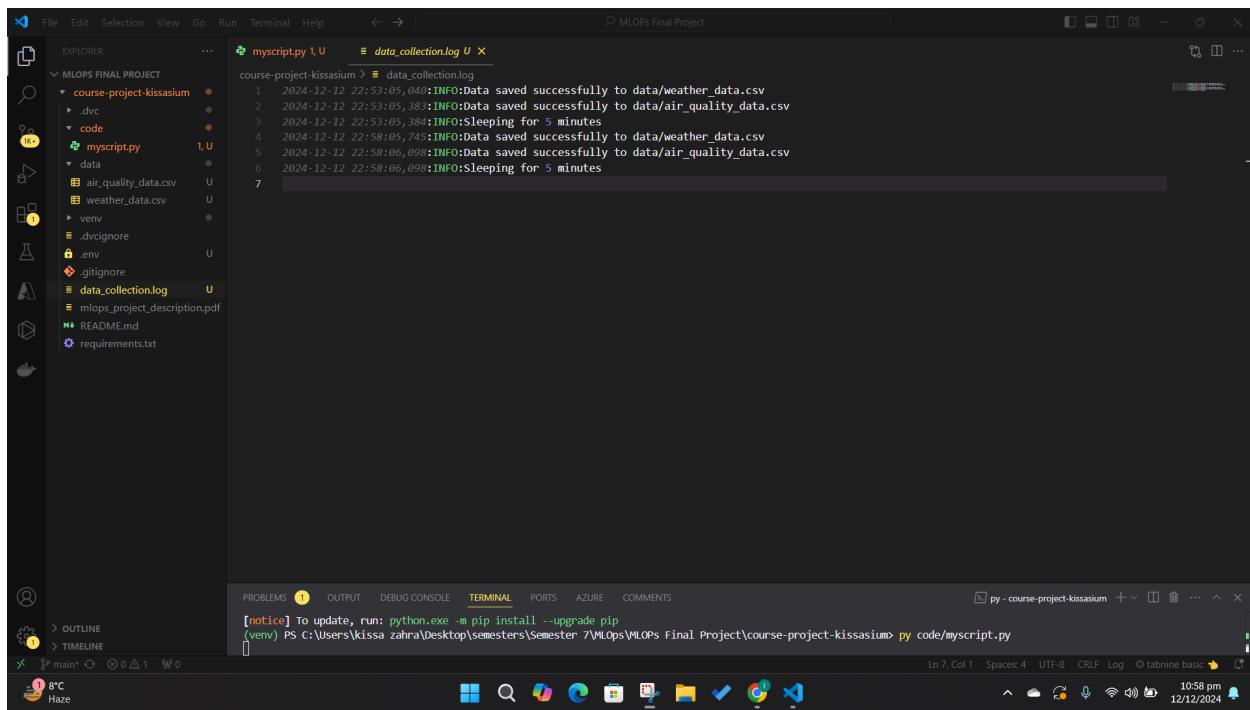
New files have been created inside the data folder; **air\_quality\_data.csv** and **weather\_data.csv**



The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the execution of the script and its output:

```
6.0 httplib2-0.22.0 oauth2client-4.1.3 proto-plus-1.25.0 pyOpenSSL-24.2.1 pydrive2-1.21.3 uritemplate-4.1.1
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissasium> dvc remote add -d myremote gdrive://1XqGfPdn8gyk8We-dov
Setting 'myremote' as a default remote.
0gkyYrtow6
Collecting python-dotenv
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
  Installing collected packages: python-dotenv
  Successfully installed python-dotenv-1.0.1
[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissasium> py code/myscript.py
```

### data\_collection.log



The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the execution of the script and its output, including log messages from the data collection process:

```
1 2024-12-12 22:53:05,048:INFO:Data saved successfully to data/weather_data.csv
2 2024-12-12 22:53:05,383:INFO:Data saved successfully to data/air_quality_data.csv
3 2024-12-12 22:53:05,384:INFO:Sleeping for 5 minutes
4 2024-12-12 22:58:05,745:INFO:Data saved successfully to data/weather_data.csv
5 2024-12-12 22:58:06,098:INFO:Data saved successfully to data/air_quality_data.csv
6 2024-12-12 22:58:06,098:INFO:Sleeping for 5 minutes
7
```

## air\_quality\_data.csv

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the contents of the 'air\_quality\_data.csv' file. The data consists of three rows of comma-separated values:

```
1 timestamp,aqI,pm2_5,pm10,no2,o3
2 2024-12-12 23:14:42,2,7.22,8.31,53.47,0.16
3 2024-12-12 23:19:42,2,7.22,8.31,53.47,0.16
```

The terminal also shows a 'KeyboardInterrupt' message and the command being run: '(venv) PS C:\Users\kissa\_zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissasium>'. The status bar at the bottom right indicates the time as 11:25 pm and the date as 12/12/2024.

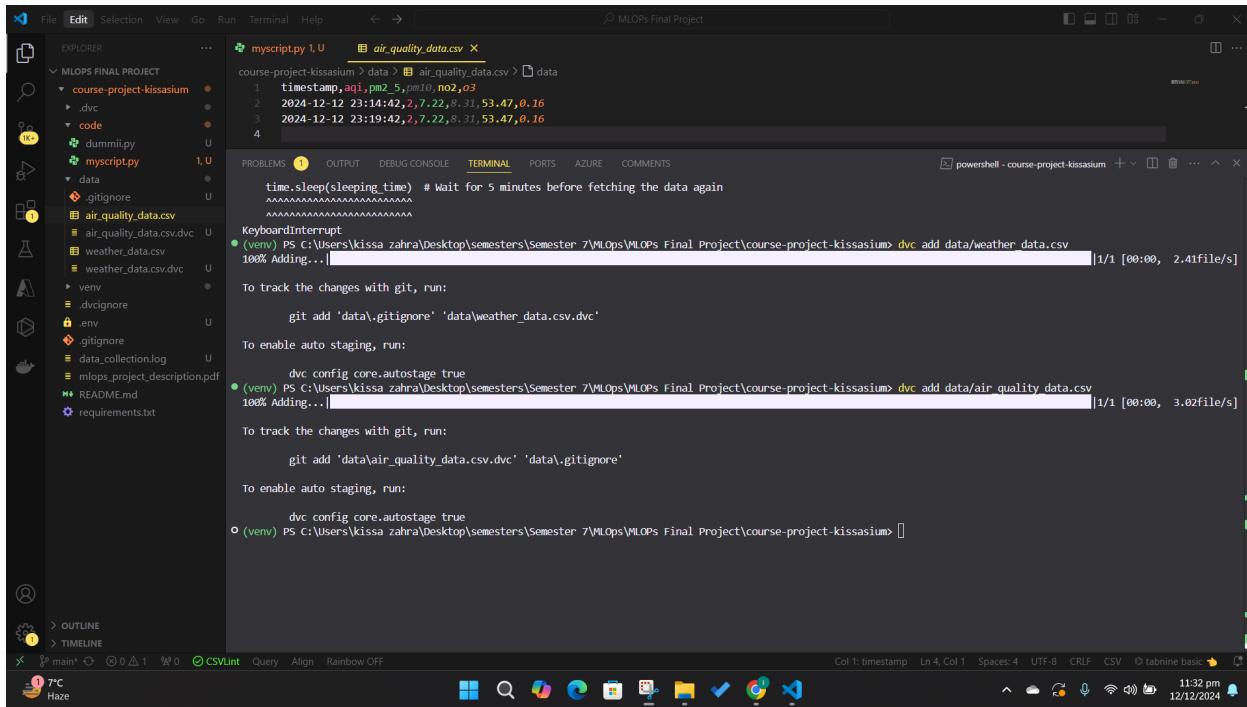
## weather\_data.csv

The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the contents of the 'weather\_data.csv' file. The data consists of three rows of comma-separated values:

```
1 timestamp,temperature,humidity,weather_condition
2 2024-12-12 23:14:41,7.57,90,overcast clouds
3 2024-12-12 23:19:42,7.57,90,overcast clouds
```

The terminal also shows a 'KeyboardInterrupt' message and the command being run: '(venv) PS C:\Users\kissa\_zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissasium>'. The status bar at the bottom right indicates the time as 11:25 pm and the date as 12/12/2024.

## 5. Version Control with DVC



The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following command and its execution:

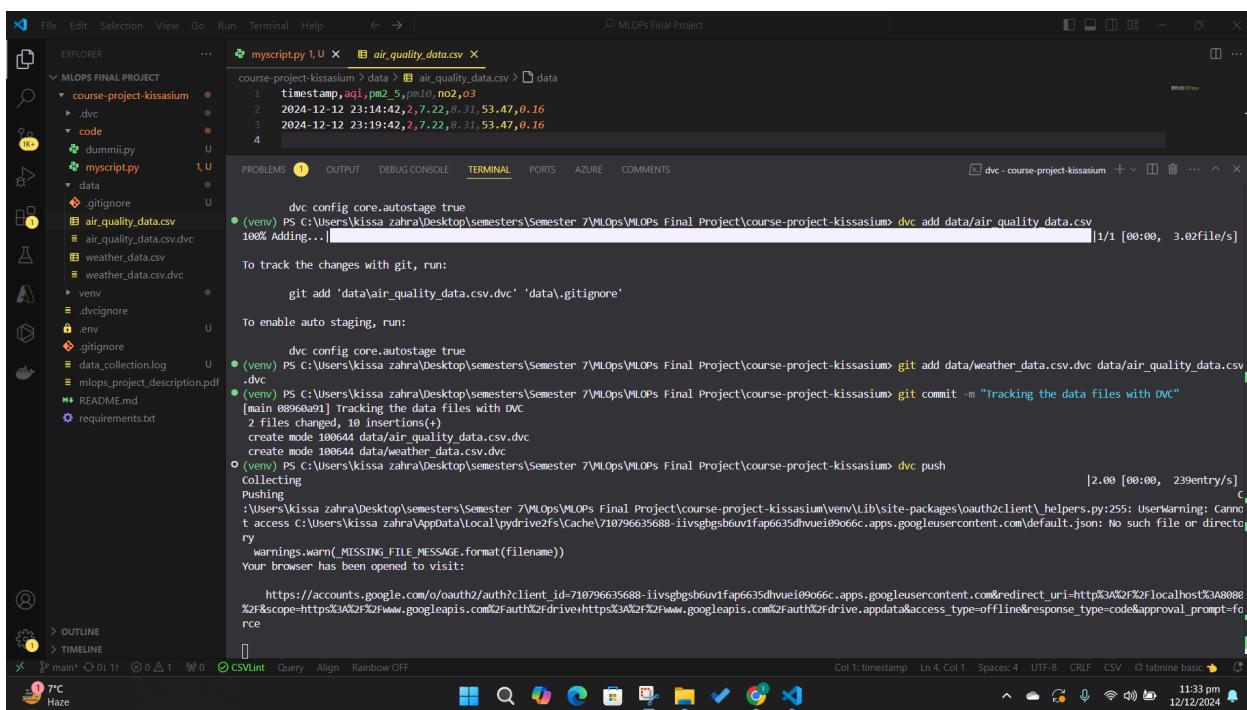
```
dvc add data/weather_data.csv
```

Output:

```
100% Adding... | 1/1 [00:00, 2.41file/s]
```

Below the terminal, several explanatory comments are provided:

- To track the changes with git, run:  
git add 'data\weather\_data.csv.dvc'
- To enable auto staging, run:  
dvc config core.autostage true
- To track the changes with git, run:  
git add 'data\air\_quality\_data.csv.dvc' 'data\gitignore'
- To enable auto staging, run:  
dvc config core.autostage true



The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following commands and their execution:

```
git add data/weather_data.csv data/air_quality_data.csv
```

Output:

```
100% Adding... | 1/1 [00:00, 3.02file/s]
```

Following this, the user runs:

```
git commit -m "Tracking the data files with DVC"
```

Output:

```
[main 0896e95] Tracking the data files with DVC
 2 files changed, 10 insertions(+)
 create mode 100644 data/air_quality_data.csv
 create mode 100644 data/weather_data.csv
```

Then, the user runs:

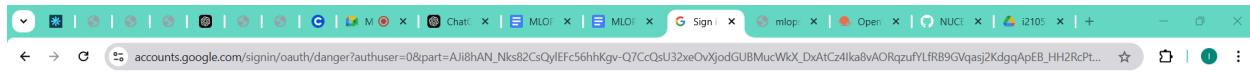
```
dvc push
```

Output:

```
2.00 [00:00, 239entry/s]
Collecting
Pushing
:Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\Lib\site-packages\oauth2client\_helpers.py:255: UserWarning: Cannot access C:\Users\kissa zahra\AppData\Local\pydrive2fs\cache\710796635688-ii1vsgbgsb6uv1fap6635dhvui09066c.apps.googleusercontent.com\default.json: No such file or directory
  warnings.warn(_MISSING_FILE_MESSAGE.format(filename))
Your browser has been opened to visit:
https://accounts.google.com/o/oauth2/auth?client_id=710796635688-ii1vsgbgsb6uv1fap6635dhvui09066c.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.appdata&access_type=offline&response_type=code&approval_prompt=force
```

```
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPs Final  
Project\course-project-kissassium> dvc push  
Collecting  
|2.00 [00:00, 239entry/s]  
Pushing  
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPs Final  
Project\course-project-kissassium\venv\Lib\site-packages\oauth2client\_helpers.py:255:  
UserWarning: Cannot access C:\Users\kissa  
zahra\AppData\Local\pydrive2fs\Cache\710796635688-iivgsgbsb6uv1fap6635dhvui09o66c.ap  
ps.googleusercontent.com\default.json: No such file or directory  
    warnings.warn(_MISSING_FILE_MESSAGE.format(filename))  
Your browser has been opened to visit:
```

[https://accounts.google.com/o/oauth2/auth?client\\_id=710796635688-iivgsgbsb6uv1fap6635dhvui09o66c.apps.googleusercontent.com&redirect\\_uri=http%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.appdata&access\\_type=offline&response\\_type=code&approval\\_prompt=force](https://accounts.google.com/o/oauth2/auth?client_id=710796635688-iivgsgbsb6uv1fap6635dhvui09o66c.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.appdata&access_type=offline&response_type=code&approval_prompt=force)



This app is blocked

This app tried to access sensitive info in your Google Account. To keep your account safe,  
Google blocked this access.



So now i am gonna use a custom Google Cloud project, link:

<https://dvc.org/doc/user-guide/data-management/remote-storage/google-drive#using-a-custom-google-cloud-project-recommended>

The screenshot shows a web browser window with the URL <https://dvc.org/doc/user-guide/data-management/remote-storage/google-drive#using-a-custom-google-cloud-project-recommended>. The page title is "Using a custom Google Cloud project (recommended)". The left sidebar contains a navigation menu with links like Home, Install, Get Started, Use Cases, User Guide, Project Structure, Data Management, Remote Storage (with sub-links for Amazon S3, Azure Blob Storage, Google Cloud Storage), Google Drive, Aliyun OSS, SSH & SFTP, HDFS & WebHDFS, HTTP, WebDAV, and Cloud Versioning. The main content area starts with a note about creating a Google Cloud project for OAuth credentials. It lists three bullet points: "You control your Google API usage limits, being able to request Google for an increase if needed.", "It ensures optimal data transfer performance when you need it.", and "Using a service account for automation tasks (e.g. CI/CD) is only possible this way." Below this, there are three numbered steps: 1. Sign into the [Google API Console](#). (Note: Double check you're using the intended Google account (upper-right corner).), 2. Select or [Create](#) a project for DVC remote connections., and 3. [Enable the Drive API](#) from the [APIs & Services Dashboard](#) (left sidebar), click on **ENABLE APIs AND SERVICES**. Find and select the "Google Drive API" in the API Library, and click on the **ENABLE** button. The right sidebar contains links to "CONTENT" (Google Drive, URL format, Using a custom Google Cloud project (recommended), Authorization, Using service accounts, Configuration parameters), a "Found an issue? Let us know! Or fix it:" section with a "Edit on GitHub" button, and a "Have a question? Join our chat, we will help you:" section with a "Discord Chat" button. The browser status bar at the bottom shows the time as 11:35 pm and the date as 12/12/2024.

1. Sign into the [Google API Console](#) and create a new API key for the google drive.
2. From the left sidebar, select Credentials, and click the Create credentials dropdown to select OAuth client ID. Choose Desktop app and click Create to proceed with a default client name.
3. Download the json file.

The screenshot shows the Google Cloud API Library interface. A search bar at the top right contains the query "google drive api". Below the search bar, a message encourages users to secure their account with multi-factor authentication. A "LEARN MORE" button is available for this message. The main content area displays a list of 11 results under the heading "11 results". The results are categorized by visibility (Public) and category (Analytics, Big data, Databases, Maps, DevOps, Healthcare & Life Sciences). Each result entry includes a thumbnail icon, the API name, its provider, and a brief description. The "Google Drive API" is listed first, followed by "Google Drive Activity API" and "Drive Labels API". The status bar at the bottom right shows the time as 11:45 pm and the date as 12/12/2024.

The screenshot shows the "Edit app registration" page for the "OAuth consent screen". The left sidebar lists sections: "Enabled APIs & services", "Library", "Credentials", "OAuth consent screen" (which is selected), and "Page usage agreements". The main content area has tabs for "App information" and "App logo". Under "App information", there is a section for "App name" (set to "KissaMlops"), "User support email" (set to "i21057@nu.edu.pk"), and a note about contacting users with questions about consent. Under "App logo", there is a section for uploading a logo, with a note about file size and format requirements. To the right, a "Learn" section titled "How is this info presented to users?" shows a simulated consent screen from "Sign in with Google". The simulated screen is divided into three numbered sections: 1. A box stating "[App Name] wants access to your Google Account". 2. A box asking "Select what [App Name] can access" with a checkbox. 3. A box stating "Make sure you trust [App Name]". The status bar at the bottom right shows the time as 11:46 pm and the date as 12/12/2024.

The screenshot shows the Google Cloud API & Services Credentials page. On the left sidebar, 'Credentials' is selected. Under 'API Keys', there is one entry: 'API key 1' (Oct 11, 2024). Under 'OAuth 2.0 Client IDs', there is one entry: 'mllopsproject' (Dec 12, 2024, Desktop, Client ID: 90872580314-fdvrg...). A modal window at the bottom center says 'OAuth client created'. The status bar at the bottom right shows '11:49 pm 12/12/2024'.

**Download the json file.**

**Use this file client id and client secrets to access the g-drive**

The screenshot shows a terminal window in VS Code. The code in 'myscript.py' is:

```

course-project-kissassium > dvc > config
1 [core]
2   remote = myremote
3   ['remote "myremote"']
4   url = gdrive://1XqjGfmPDanBykBwe-doV0qKyIrtovw
5
6   gdive_client_id = 90872580314-fdvrgfd0f6biaoggbjqqbsh7ksdaribr.apps.googleusercontent.com
7   gdive_client_secret = GOCSPX-0YDTSEWU14OK4dUy8fdpJbqRs31
8

```

The terminal output shows:

```

2 files changed, 10 insertions(+)
create mode 100644 data/air_quality_data.csv.dvc
create mode 100644 data/weather_data.csv.dvc
(venv) PS C:\Users\kissa.zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissassium> dvc push
Collecting
Pushing
;C:\Users\kissa.zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissassium\venv\Lib\site-packages\oauth2client\helpers.py:255: UserWarning: Can't access C:\Users\kissa.zahra\AppData\Local\pydrive2fs\Cache\710796635688-ivsgsb6uv1fap6635dhvue109066c.apps.googleusercontent.com/default.json: No such file or directory
  warnings.warn(MISSING_FILE_MESSAGE.format(filename))
Your browser has been opened to visit:

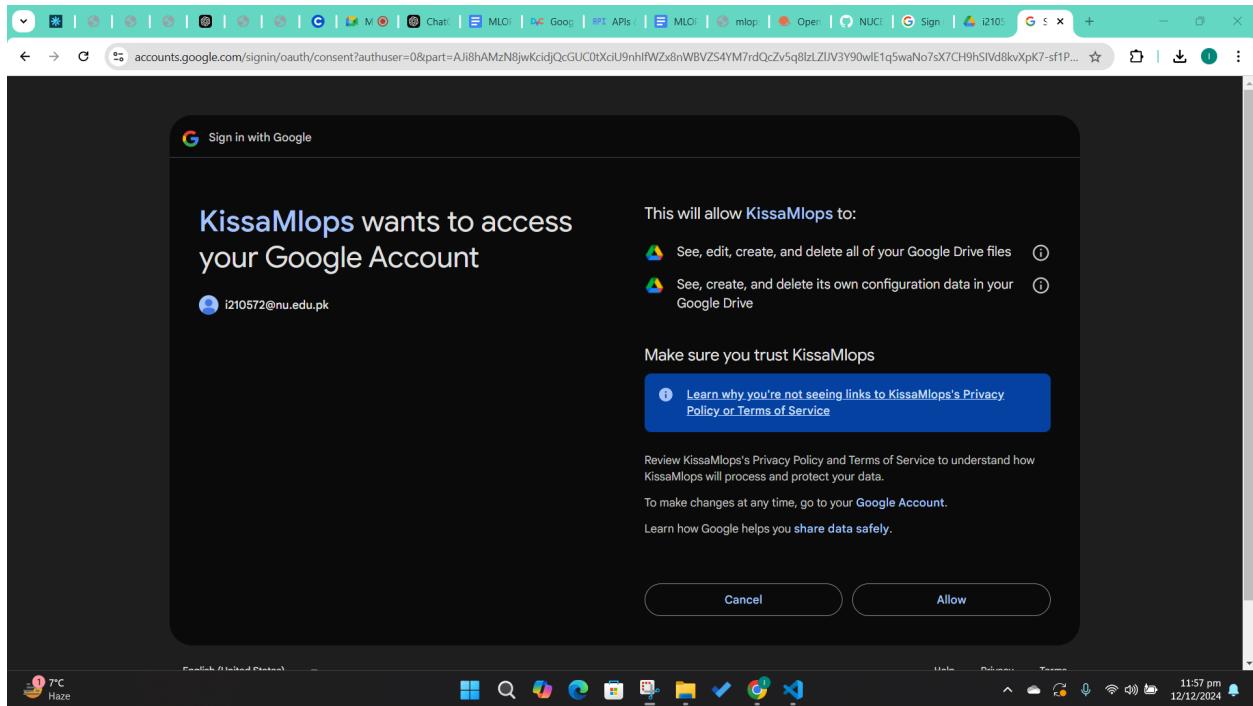
https://accounts.google.com/o/oauth2/client_id=710796635688-ivsgsb6uv1fap6635dhvue109066c.apps.googleusercontent.com&redirect_uri=http%3A%2F%2localhost%3A48888%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.appdata.access_type=offline&response_type=code&approval_prompt=force

```

**Run these two commands:**

```
`dvc remote modify myremote gdrive_client_id  
90872580314-fdvrgfd0f6biaogsbjqqbsh7ksdarirb.apps.googleusercontent.com`  
`dvc remote modify myremote gdrive_client_secret  
GOCSNX-0YDtSEVmU14DK4dUyBfdPjbqRs31`  
`dvc push`
```

You can access the google drive now!!!





One folder has been created!!

A screenshot of a Google Drive interface. The left sidebar shows navigation options like Home, Activity, Workspaces, My Drive, Shared with me, Recent, Starred, Spam, Trash, and Storage. The main area shows a folder named 'i210572\_mlops\_project' under 'My Drive'. Inside this folder, there is a single item named 'files'. On the right side, there is a detailed view of the folder, showing 'Who has access' (Private to you) and 'Folder details' (Type: Google Drive Folder, Location: My Drive). The status bar at the bottom shows the date and time as '11:58 pm 12/12/2024'.

Within that folder there exist two folder for the datasets (weather\_data.csv and air\_quality\_data.csv) that has been pushed using `dvc push`

drive.google.com/drive/folders/1n-F45JBLNB8dx\_j0dolsiYrPDS7gtTHx?dmr=1&ec=wgc-drive-hero-goto

Drive

Search in Drive

... > files > md5

Type People Modified

Name	Owner	Last modified	File size
77	me	12:00 AM	—
d3	me	12:00 AM	—

Details Activity

Who has access

Private to you

Manage access

Folder details

Type Google Drive Folder

Location files

12:05 am 13/12/2024

drive.google.com/drive/folders/1n-F45JBLNB8dx\_j0dolsiYrPDS7gtTHx?dmr=1&ec=wgc-drive-hero-goto

Drive

Search in Drive

... > files > md5

Type People Modified

Name	Owner	Last modified	File size
77	me	12:00 AM	—
d3	me	12:00 AM	—

Details Activity

Who has access

Private to you

Manage access

Folder details

Type Google Drive Folder

Location files

11:58 pm 12/12/2024

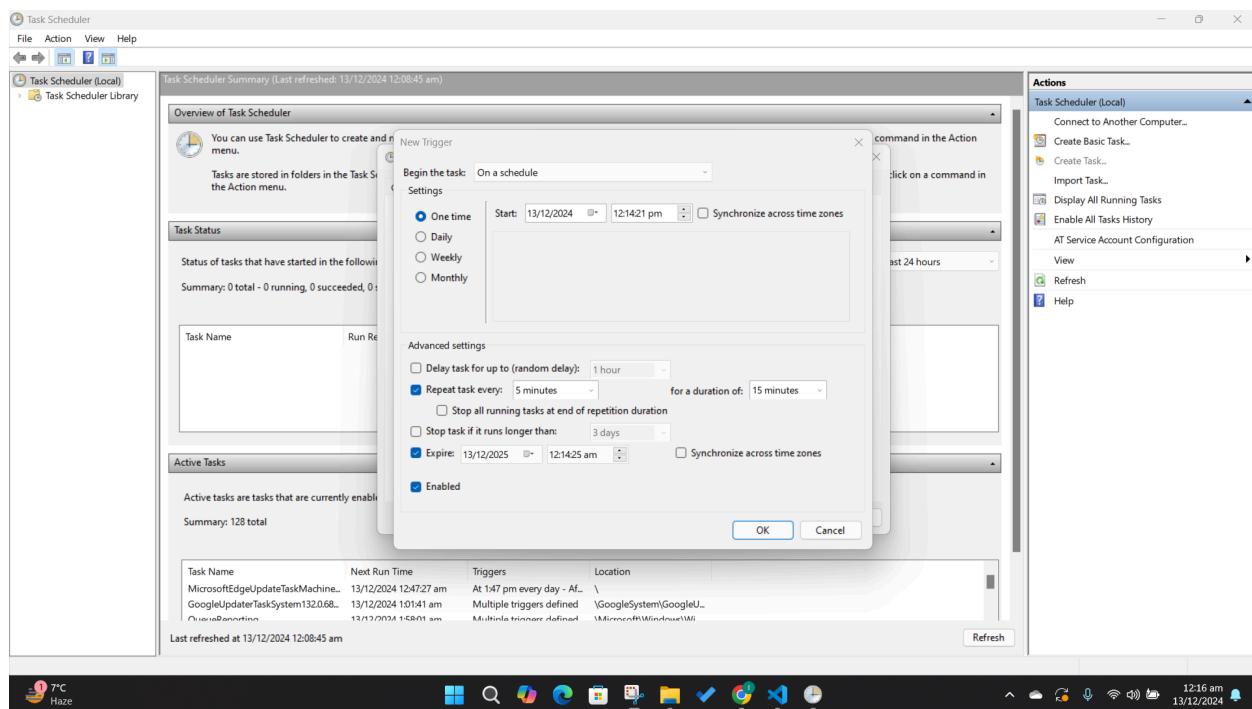
## 6. Automate Data Collection

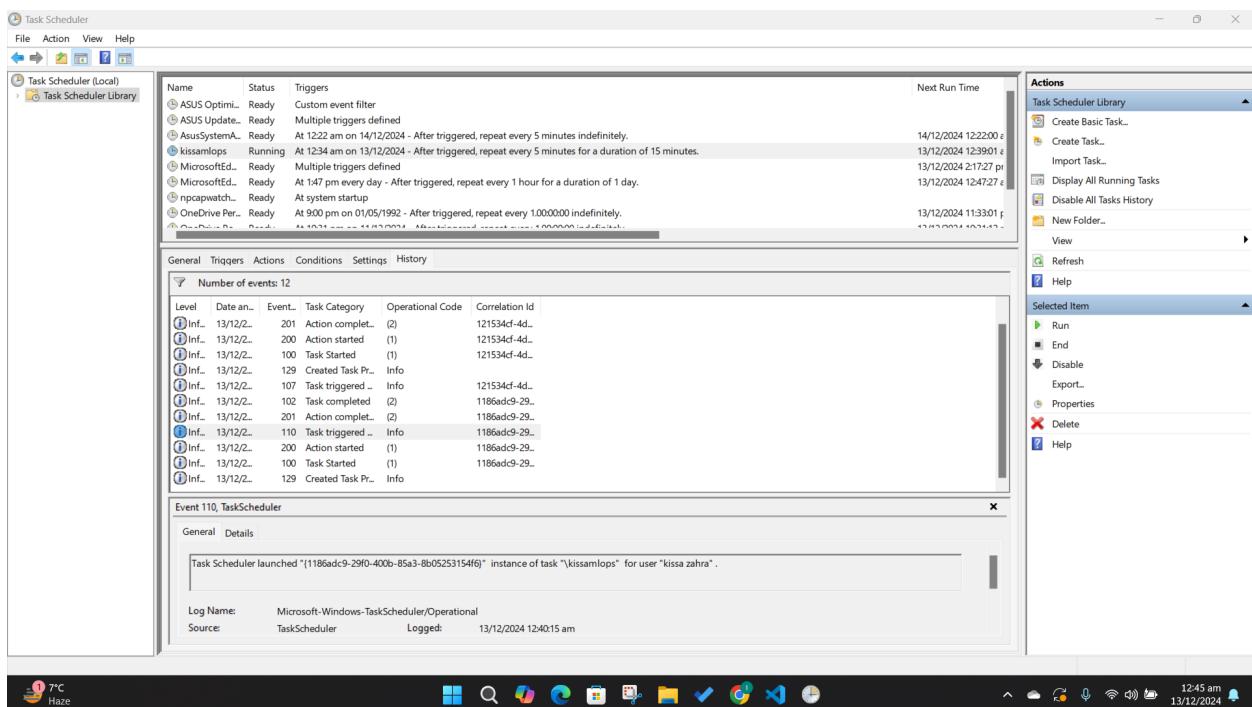
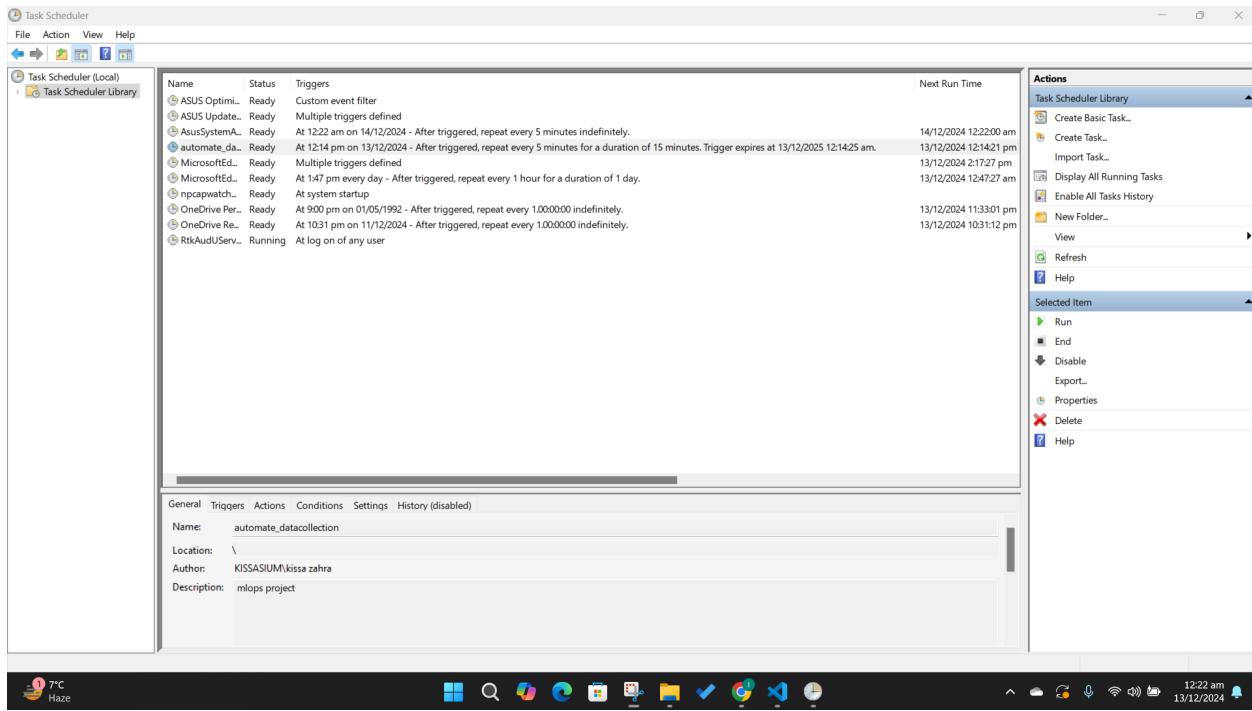
Schedule regular data fetching using cron jobs or task schedulers. I used ‘task scheduler’ since I’m using Windows.

Open Task Scheduler and click on Create Task.

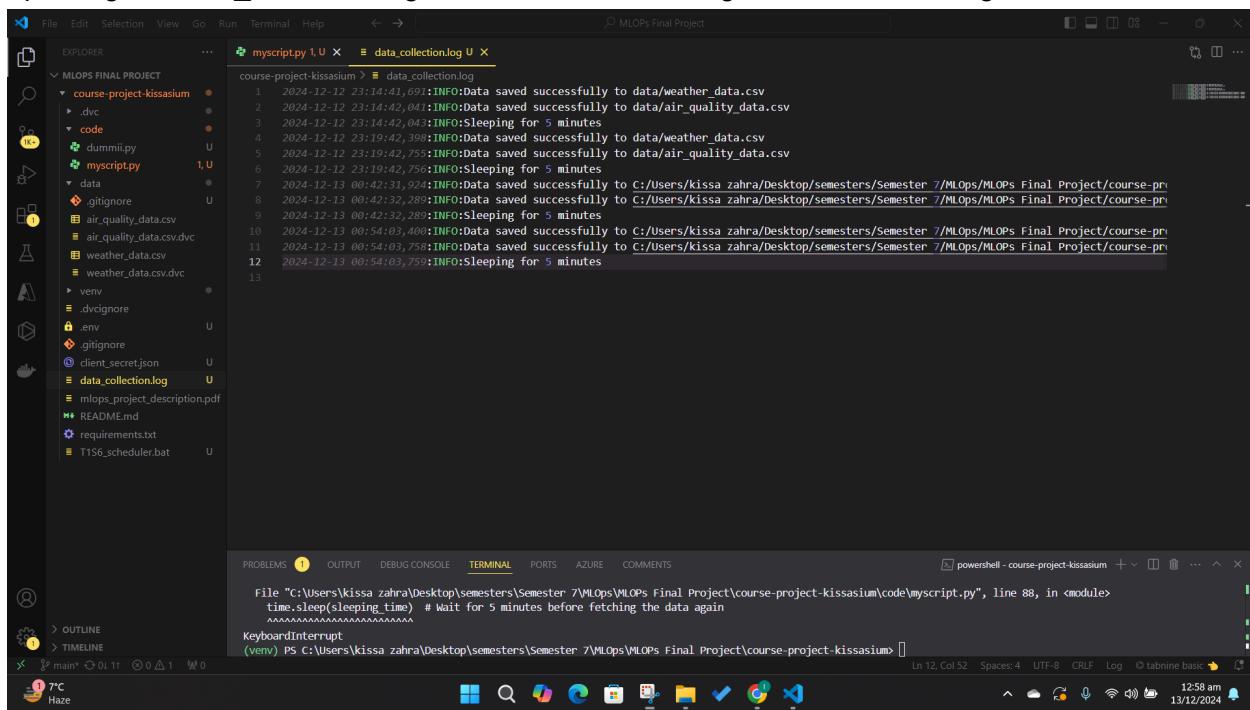
Go to the Triggers tab to set the schedule for the task.

Go to Actions and select the virtual environment Python executable as the Program/Script. Add the absolute path enclosed in double quotes, to the .py code file as the argument in the task. Ensure that you are using an absolute path when saving your files within the code too.





## Updating the data\_collection.log file with task scheduling method for fetching the data



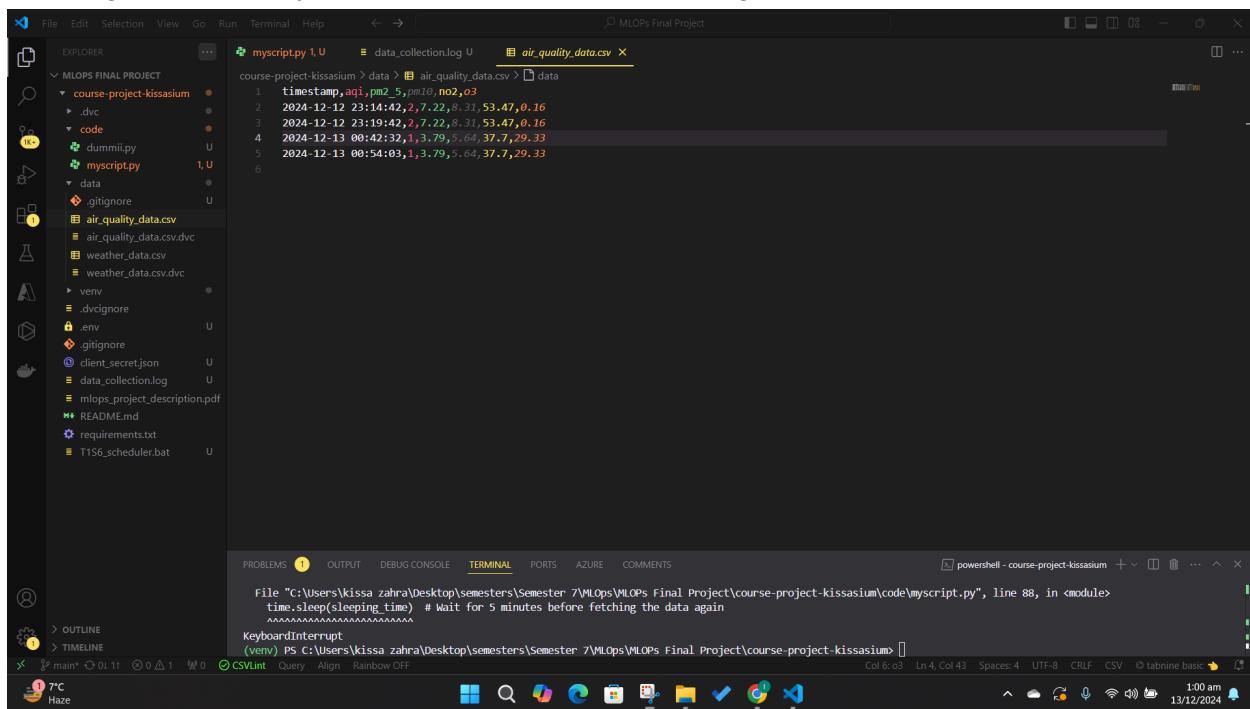
The screenshot shows the VS Code interface with the 'data\_collection.log' file open in the center editor pane. The log file contains the following content:

```
course-project-kissassium > data_collection.log
1 2024-12-12 23:14:41,691:INFO:Data saved successfully to data/weather_data.csv
2 2024-12-12 23:14:42,041:INFO:Data saved successfully to data/air_quality_data.csv
3 2024-12-12 23:14:42,043:INFO:Sleeping for 5 minutes
4 2024-12-12 23:19:42,398:INFO:Data saved successfully to data/weather_data.csv
5 2024-12-12 23:19:42,755:INFO:Data saved successfully to data/air_quality_data.csv
6 2024-12-12 23:19:42,756:INFO:Sleeping for 5 minutes
7 2024-12-13 00:42:31,924:INFO:Data saved successfully to C:/Users/kissa zahra/Desktop/semesters/Semester 7/MLOps/MLOPS Final Project/course-project-kissassium/data/air_quality_data.csv
8 2024-12-13 00:42:32,756:INFO:Data saved successfully to C:/Users/kissa zahra/Desktop/semesters/Semester 7/MLOps/MLOPS Final Project/course-project-kissassium/data/weather_data.csv
9 2024-12-13 00:42:32,289:INFO:Sleeping for 5 minutes
10 2024-12-13 00:54:03,409:INFO:Data saved successfully to C:/Users/kissa zahra/Desktop/semesters/Semester 7/MLOps/MLOPS Final Project/course-project-kissassium/data/weather_data.csv
11 2024-12-13 00:54:03,758:INFO:Data saved successfully to C:/Users/kissa zahra/Desktop/semesters/Semester 7/MLOps/MLOPS Final Project/course-project-kissassium/data/air_quality_data.csv
12 2024-12-13 00:54:03,759:INFO:Sleeping for 5 minutes
13
```

The terminal below shows the command run:

```
File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPS Final Project\course-project-kissassium\code\myscript.py", line 88, in <module>
    time.sleep(sleeping_time) # Wait for 5 minutes before fetching the data again
~~~~~
KeyboardInterrupt
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPS Final Project\course-project-kissassium\code\myscript.py
```

## Updating the air\_quality\_data.csv file with task scheduling



The screenshot shows the VS Code interface with the 'air\_quality\_data.csv' file open in the center editor pane. The CSV file contains the following data:

timestamp	aqi	pm2_5	pm10	no2	o3	
2024-12-12 23:14:42	2	7	22	8.31	53.47	0.16
2024-12-12 23:19:42	2	7	22	8.31	53.47	0.16
2024-12-13 00:42:32	1	3	79	5.64	37.7	29.33
2024-12-13 00:54:03	1	3	79	5.64	37.7	29.33

The terminal below shows the command run:

```
File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPS Final Project\course-project-kissassium\code\myscript.py", line 88, in <module>
    time.sleep(sleeping_time) # Wait for 5 minutes before fetching the data again
~~~~~
KeyboardInterrupt
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPS Final Project\course-project-kissassium\code\myscript.py
```

## Updating the weather\_data.csv file with task scheduling

The screenshot shows a code editor interface with several tabs open. The main tab displays a Python script named `mymyscript.py` with the following content:

```
1 timestamp,temperature,humidity,weather_condition
2 2024-12-12 23:14:41,7.57,90,overcast clouds
3 2024-12-12 23:19:42,7.57,90,overcast clouds
4 2024-12-13 00:42:31,7.58,90,overcast clouds
5 2024-12-13 00:54:03,7.58,90,overcast clouds
6
```

The sidebar on the left shows the project structure:

- MLOPS FINAL PROJECT
  - course-project-kissassium
    - .dvc
  - code
    - dummii.py
    - mymyscript.py
  - data
    - .gitignore
    - air\_quality\_data.csv
    - air\_quality\_data.csv.dvc
    - weather\_data.csv
    - weather\_data.csv.dvc
    - venv
  - .dvcignore
  - env
    - .gitignore
  - client\_secret.json
  - data\_collection.log
  - mlops\_project\_description.pdf
- README.md
- requirements.txt
- T1S6\_scheduler.bat

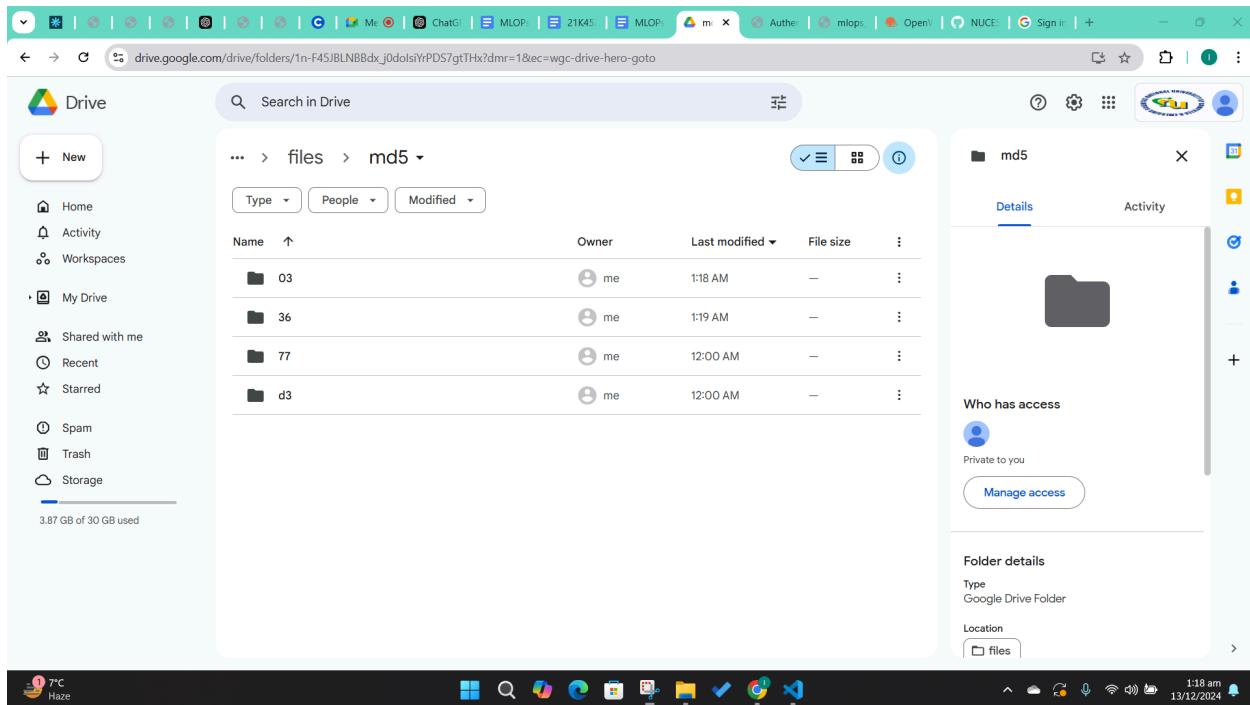
At the bottom, the terminal window shows the command run and its output:

```
File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPS Final Project\course-project-kissassium\code\mymyscript.py", line 88, in <module>
    time.sleep(sleeping_time) # Wait for 5 minutes before fetching the data again
    ~~~~~~
```

The status bar at the bottom right indicates the current time as 101 am on 13/12/2024.

## 7. Update Data with DVC

Updated on the drive



## Now pushing it on github

```

course-project-kissassium> git add .
course-project-kissassium> git commit -m "task1 completed"
[main ac438921] task1 completed
 1 file changed, 1 insertion(+), 2 deletions(-)
course-project-kissassium> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/NUCES-1SB/course-project-kissassium.git
 ab21d113..ab21d113 main -> main
course-project-kissassium>

```

The screenshot shows a terminal window with a dark theme. It displays the command-line process of committing and pushing changes to a GitHub repository. The terminal output shows the user navigating to the 'course-project-kissassium' directory, adding files with 'git add .', committing with a message 'task1 completed', and then pushing the changes to the remote repository at 'https://github.com/NUCES-1SB/course-project-kissassium.git'. The commit hash 'ab21d113' is shown as the main branch.

I have made the .bat file for the automation as well! It automates the process of updating and versioning data in your project. It starts by navigating to the project directory and activating the Python virtual environment. Then it runs the **data\_collections.py** script to fetch new data, tracks the updated data files (**weather\_data.csv** and **air\_quality\_data.csv**) using DVC and stages changes with Git. The script commits the updates with a message and pushes the data to the DVC remote storage and finally pushes the changes to the Git repository. This ensures both the code and data are updated and versioned efficiently.

```

course-project-kissasium > code > fetch_data.bat
1  echo off
2  cd "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium"
3  call "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\Scripts\activate"
4  py "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\code\data_collections.py"
5  dvc add data\weather_data.csv data\air_quality_data.csv
6  git add .
7  git commit -m "Update data files"
8  dvc push
9  git push
10

```

## Task 2 Pollution Trend Prediction with MLflow

Objective: Develop and deploy models to predict pollution trends and alert high-risk days.

First of all i ran the code so that i can have some good number of values on which i can train the model. I will merge the two csv files to make a 1 csv file that contains all the values!

### 1. Data Preparation

I loaded my merged.csv file on colab and ran a python script on which i removed the outliers and removed the missing values using the built -in function of pandas library. By chance I didn't have any missing values.

### 2. Model Development

#### ARIMA (AutoRegressive Integrated Moving Average) for AQI Forecasting

##### 1. Splitting the Data

The dataset is split into training (80%) and testing (20%) sets. The training set is used to train the model, while the testing set is used to evaluate the model's performance.

##### 2. Model Training and Forecasting

#### ARIMA (AutoRegressive Integrated Moving Average) for AQI Forecasting

I used the ARIMA model to predict AQI trends due to its efficiency with time series data, especially without GPU access.

##### ARIMA Components:

1. **AR (AutoRegressive)**: Uses past AQI values to predict future ones.
2. **I (Integrated)**: Makes the data stationary by removing trends or seasonality.
3. **MA (Moving Average)**: Models the relationship between observations and forecast errors.

##### Why ARIMA?

ARIMA is ideal for non-seasonal data with trends, like AQI and doesn't require heavy computational resources.

## Key Hyperparameters:

1. **p**: Lag observations used for prediction.
2. **d**: Times the data is different to make it stationary.
3. **q**: Size of the moving average window.

This approach works well on local machines for AQI forecasting.

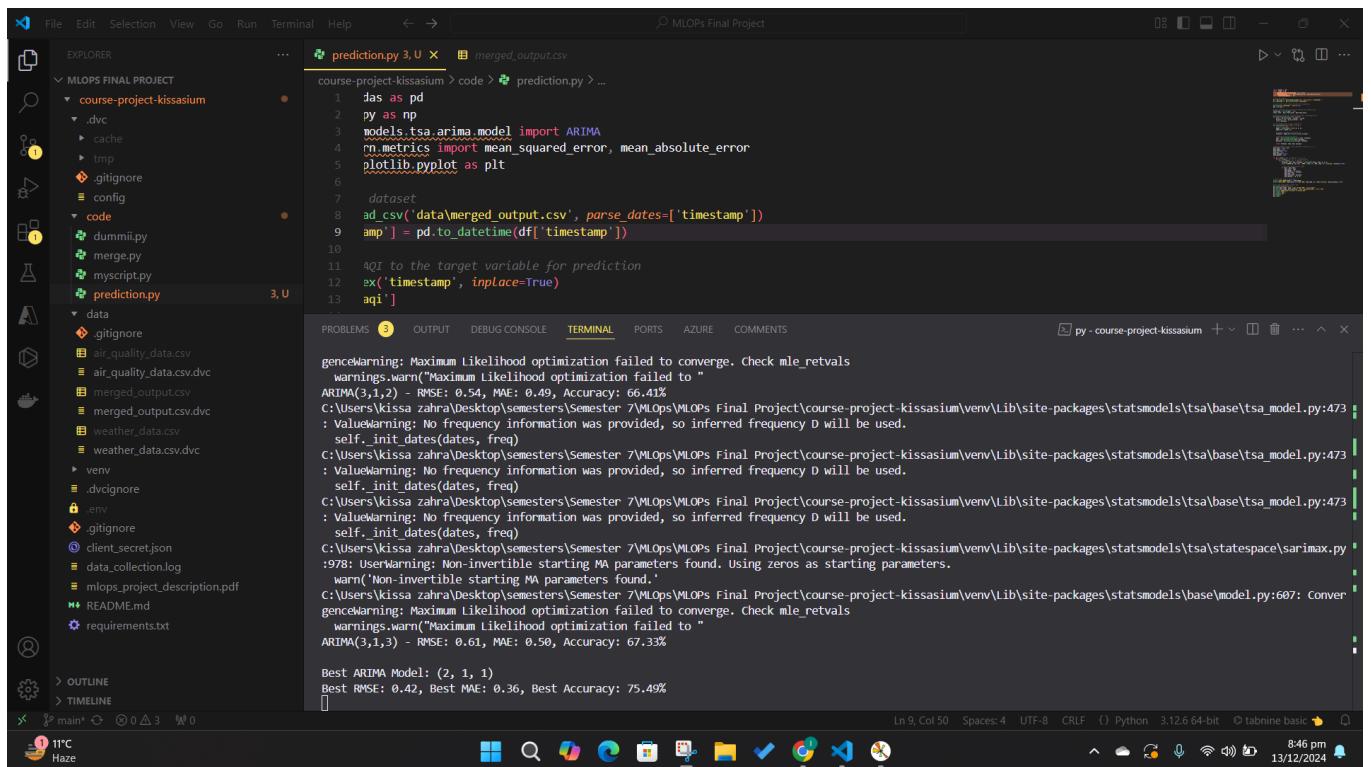
## 3. Evaluation

The model's accuracy is evaluated using metrics like RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and accuracy (100 - mean error percentage).

Lower RMSE and MAE indicate better performance.

## Output with ARIMA

### After running the code



A screenshot of a code editor (VS Code) showing the results of running a Python script named `prediction.py`. The code uses the ARIMA model from the statsmodels library to predict AQI based on merged data. The terminal output shows the ARIMA parameters (2, 1, 1), RMSE (0.42), MAE (0.36), and Accuracy (75.49%).

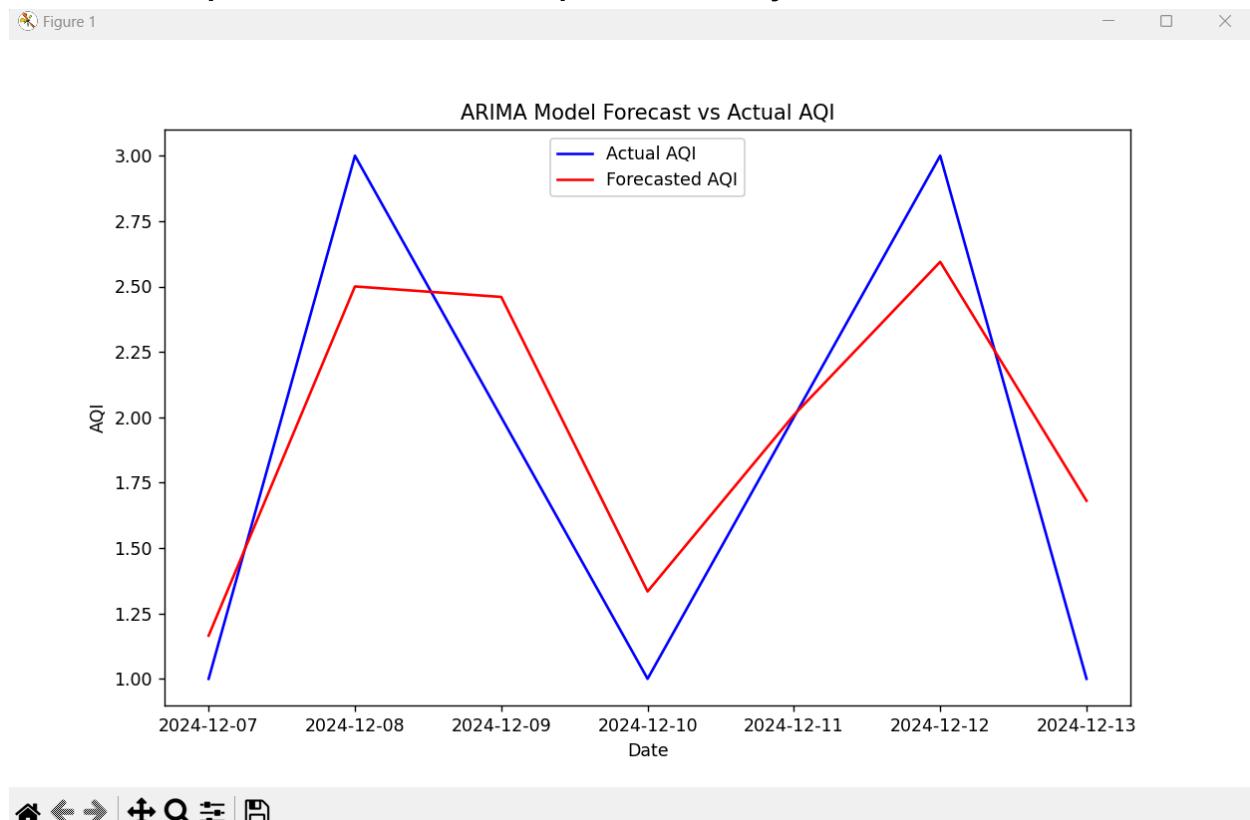
```
prediction.py 3. U merged_output.csv
course-project-kissasium > code > prediction.py > ...
1 das as pd
2 py as np
3 models.tsa.arima.model import ARIMA
4 from metrics import mean_squared_error, mean_absolute_error
5 import matplotlib.pyplot as plt
6
7 dataset
8 ad_.csv('data\merged_output.csv', parse_dates=['timestamp'])
9 df['aqi'] = pd.to_datetime(df['timestamp'])
10
11 #AQI to the target variable for prediction
12 ex('timestamp', inplace=True)
13 aqi[]

gencWarning: Maximum Likelihood optimization failed to converge. Check mle_restarts
warnings.warn("Maximum Likelihood optimization failed to "
ARIMA(2,1,1) - RMSE: 0.42, MAE: 0.36, Accuracy: 75.49%
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
self._init_dates(dates, freq)
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
self._init_dates(dates, freq)
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\lib\site-packages\statsmodels\tsa\base\tsa_model.py:473
: ValueWarning: No frequency information was provided, so inferred frequency D will be used.
self._init_dates(dates, freq)
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\lib\site-packages\statsmodels\tsa\statespace\sarimax.py
:978: UserWarning: Non-invertible starting MA parameters found. Using zeros as starting parameters.
warn('Non-invertible starting MA parameters found.')
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project\course-project-kissasium\venv\lib\site-packages\statsmodels\base\model.py:607: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_restarts
warnings.warn("Maximum Likelihood optimization failed to "
Best ARIMA Model: (2, 1, 1)
Best RMSE: 0.42, Best MAE: 0.36, Best Accuracy: 75.49%
```

**Best ARIMA Model: (2, 1, 1)**

**Best RMSE: 0.42, Best MAE: 0.36, Best Accuracy: 75.49%**

The actual vs. predicted AQI values are plotted for 5 days from 7 dec to 13 dec.



## Random Forest Model

### 1. Splitting the Data

The dataset is divided into 80% for training and 20% for testing. The model is trained on the training set and evaluated on the testing set.

### 2. Model Training and Forecasting

#### Random Forest for AQI Prediction

I used the Random Forest model for predicting AQI because it can handle complex relationships in the data. It works by building multiple decision trees using random subsets of the data and features, then averaging their predictions. This helps reduce overfitting and improves accuracy.

#### Why Random Forest?

It is effective for regression tasks with non-linear relationships and is computationally efficient making it suitable for AQI prediction.

## Key Hyperparameters:

- **n\_estimators**: Number of trees in the forest.
- **max\_depth**: Limits the depth of each tree to prevent overfitting.
- **min\_samples\_split**: Minimum samples required to split a node.

## 3. Evaluation

The model's performance is measured using RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and accuracy (mean percentage error).

Lower RMSE and MAE values and higher accuracy indicate better performance.

## Output with Random Forest Model

A screenshot of a code editor (VS Code) showing a terminal window output. The terminal shows the execution of a Python script named `randomforest.py` which prints out the best model's performance. The output includes Best RMSE, Best MAE, and Best Accuracy. The terminal also shows a stack trace for a KeyboardInterrupt exception. The code editor interface is visible on the left, showing a file structure for a project named "MLOPS FINAL PROJECT".

```
randomforest.py 3.U arimaprediction.py 3.U
course-project-kissasium\code> randomforest.py ...
161     best_params = (n_estimators, max_depth)
162
163     # Output best model's performance
164     print("\nBest Random Forest Model:", best_model)
165     print(f"\nBest RMSE: {best_rmse:.2f}, Best MAE: {best_mae:.2f}, Best Accuracy: {best_accuracy:.2f}%")
166

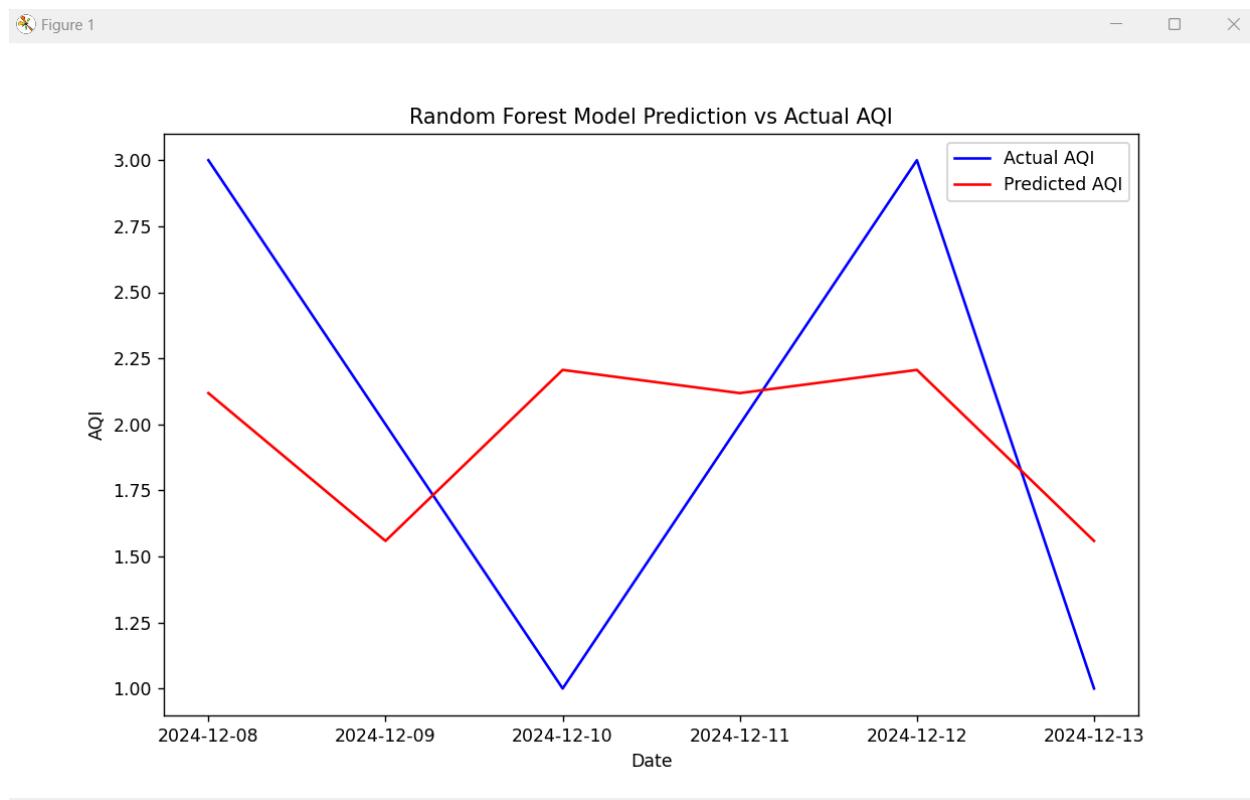
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE COMMENTS
powershell - course-project-kissasium + ... x

Best RMSE: 0.75, Best MAE: 0.67, Best Accuracy: 56.61%
Traceback (most recent call last):
  File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPs\MLOPs Final Project\course-project-kissasium\code\etsprediction.py", line 175, in <module>
    plt.show()
  File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPs\MLOPs Final Project\course-project-kissasium\venv\lib\site-packages\matplotlib\pyplot.py", line 6
12, in show
    return _get_backend_mod().show(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPs\MLOPs Final Project\course-project-kissasium\venv\lib\site-packages\matplotlib\backend_bases.py", line 3553, in show
    cls.mainloop()
  File "C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPs\MLOPs Final Project\course-project-kissasium\venv\lib\site-packages\matplotlib\backends\backend_tk.py", line 520, in start_main_loop
    first_manager.window.mainloop()
  File "C:\Python312\lib\tkinter\_init_.py", line 1505, in mainloop
    self.tk.mainloop(n)
KeyboardInterrupt
● (venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOPs\MLOPs Final Project\course-project-kissasium> py code\etsprediction.py
Random Forest (n_estimators=50, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%
Random Forest (n_estimators=50, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%
Random Forest (n_estimators=50, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%
Random Forest (n_estimators=100, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%
Random Forest (n_estimators=100, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%
Random Forest (n_estimators=100, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%
Random Forest (n_estimators=150, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%
Random Forest (n_estimators=150, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%
Random Forest (n_estimators=150, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%
```

**Best Random Forest Model: (150, 5)**

**Best RMSE: 0.75, Best MAE: 0.67, Best Accuracy: 56.61%**

The actual vs. predicted AQI values are plotted for 5 days from 7 dec to 13 dec.



### 3. Train Models with MLflow

`pip install mlflow`

MLflow 2.19.0 Experiments Models

Pollution Prediction - Random Forest

Run Name	Created	Dataset	Duration	Source	Models
nosy-shrew-658	1 minute ago	-	199ms	task2.py	-
orderly-shad-671	1 minute ago	-	216ms	task2.py	-
intrigued-cod-37	1 minute ago	-	208ms	task2.py	-
treasured-duck-154	1 minute ago	-	156ms	task2.py	-
ambitious-bass-514	1 minute ago	-	157ms	task2.py	-
luminous-fawn-605	1 minute ago	-	150ms	task2.py	-
exultant-chimp-698	1 minute ago	-	120ms	task2.py	-
bedecked-tern-495	1 minute ago	-	101ms	task2.py	-
trusting-shrimp-888	1 minute ago	-	114ms	task2.py	-

9 matching runs

## ARIMA

The screenshot shows the MLflow UI for an experiment named "flawless-sloth-686". The experiment was created at 2024-12-13 22:06:16 by "kissa zahra". It has an Experiment ID of 101735281539724587 and a Status of "Finished". The Run ID is b894b93477d346fb94391b1a8f18d1f7, with a Duration of 363ms. No datasets were used, and there are no registered models. The parameters listed are d=1, p=3, and q=3. The metrics shown are Accuracy (67.33), MAE (0.30), and RMSE (0.61). The system status bar indicates it's 11°C and shows the date and time as 13/12/2024 10:10 pm.

The screenshot shows the MLflow UI for the same experiment, focusing on the "Model metrics" tab. It displays three horizontal bar charts for Accuracy, MAE, and RMSE. The Accuracy chart shows a value of 67.33, the MAE chart shows 0.30, and the RMSE chart shows 0.61. All three charts are labeled "flawless-sloth-686". The system status bar indicates it's 11°C and shows the date and time as 13/12/2024 10:10 pm.

## Random Forest Model

The screenshot shows the MLflow UI for a Random Forest model named 'nosy-shrew-658'. The 'Overview' tab is selected. Key details include:

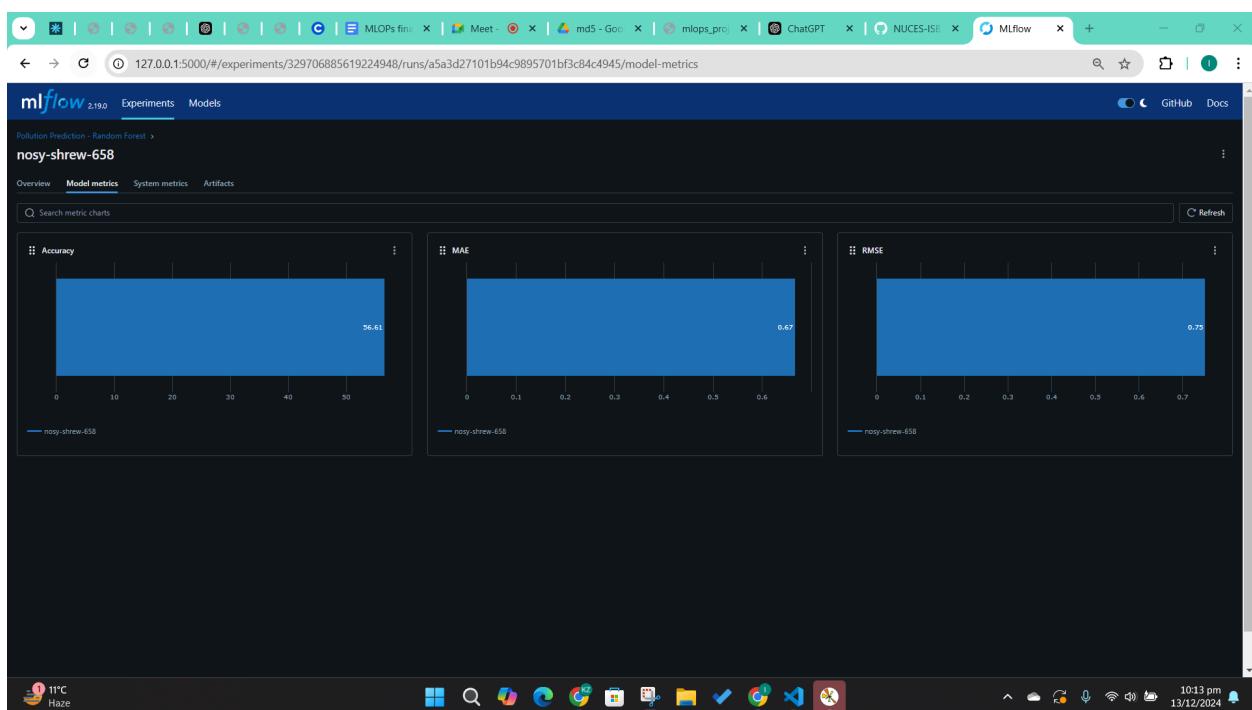
- Created at:** 2024-12-13 22:06:14
- Created by:** kosa zafra
- Experiment ID:** 329706885619224948
- Status:** Finished
- Run ID:** a5a3d27101b94c9895701bf3c84c4945
- Duration:** 199ms
- Datasets used:** —
- Tags:** Add
- Source:** task2.py (b200a85)
- Logged models:** —
- Registered models:** —

**Parameters (2):**

Parameter	Value
max_depth	15
n_estimators	150

**Metrics (3):**

Metric	Value
Accuracy	56.613896134729494
MAE	0.6666666666666666
RMSE	0.7509271265922725



## 4. Hyperparameter Tuning

I have used a grid search approach. It involves exhaustively trying all possible combinations of the specified hyperparameters over a defined grid.

### ARIMA

**Hyperparameter tuning is done for the parameters p, d, and q (the ARIMA model's order).**

p: The order of the autoregressive part (1 to 3). **(3 values)**

d: The degree of differencing (only 1 is tested). **(1 values)**

q: The order of the moving average part (1 to 3). **(3 values)**

The code iterates over all possible combinations of p, d, and q, and logs the performance metrics (RMSE, MAE, Accuracy) for each combination.

The best model (with the lowest RMSE) is selected based on the performance during the iterations.

```
ARIMA(3,1,3) - RMSE: 0.61, MAE: 0.50, Accuracy: 67.33%
```

```
Best ARIMA Model: (2, 1, 1)
```

```
Best RMSE: 0.42
```

**Pollution Prediction - ARIMA > flawless-sloth-686**

**Overview** Model metrics System metrics Artifacts

**Description** [Edit](#)  
No description

**Details**

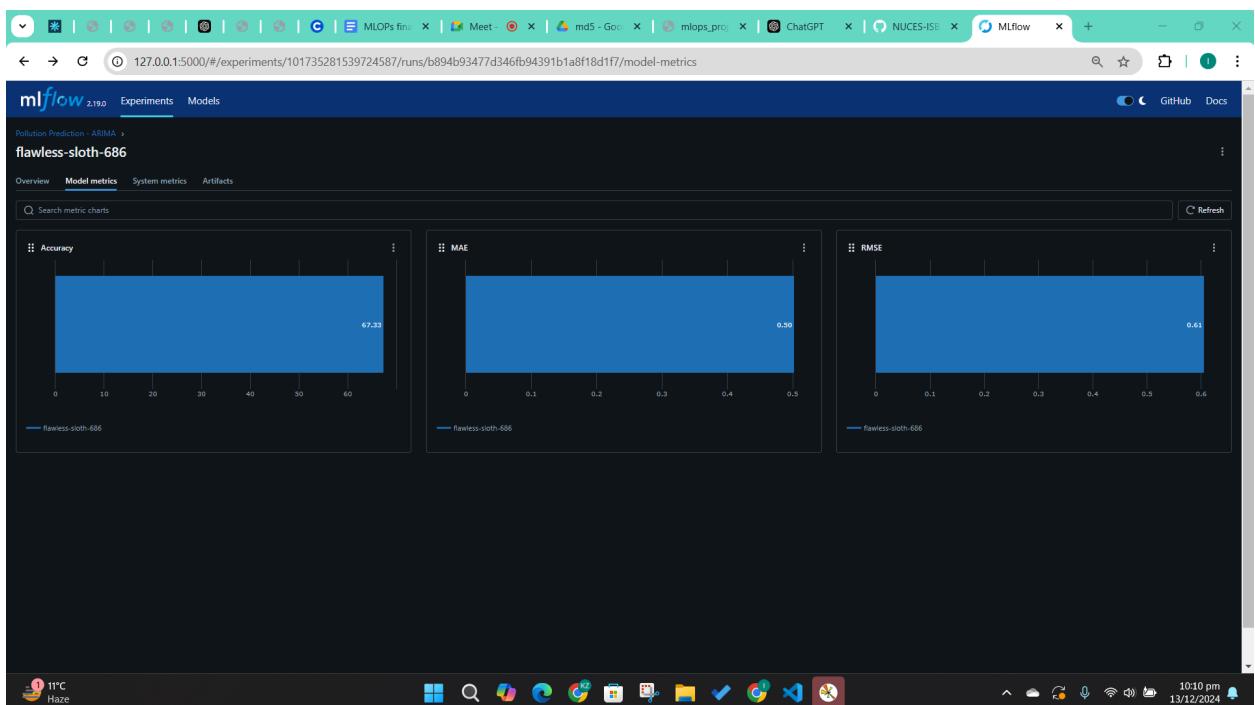
Created at	2024-12-13 22:06:16
Created by	kissa zahra
Experiment ID	101735281539724587
Status	Finished
Run ID	b894b93477d346fb94391b1a8f18d1f7
Duration	363ms
Datasets used	—
Tags	Add
Source	<a href="#">task2.py</a> ↗ b200a8b
Logged models	—
Registered models	—

**Parameters (3)**

Search parameters	
Parameter	Value
d	1
p	3
q	3

**Metrics (3)**

Search metrics	
Metric	Value
Accuracy	67.32
MAE	0.50
RMSE	0.61



## Random Forest Model

**Hyperparameter tuning is done for two parameters:** n\_estimators (the number of trees) and max\_depth (the maximum depth of each tree).

The code iterates over different values for these parameters (n\_estimators = 50, 100, 150; max\_depth = 5, 10, 15). **(3 values each)**

And logs the performance metrics (RMSE, MAE, Accuracy) for each combination.

The best model (with the lowest RMSE) is selected based on the performance during the iterations.

```
2024/12/13 22:06:12 INFO mlflow.tracking.fluent: Experiment with name 'Pollution Prediction - Ra  
ment.  
Random Forest (n_estimators=50, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%  
Random Forest (n_estimators=50, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%  
Random Forest (n_estimators=50, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 57.14%  
Random Forest (n_estimators=100, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%  
Random Forest (n_estimators=100, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%  
Random Forest (n_estimators=100, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.62%  
Random Forest (n_estimators=150, max_depth=5) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%  
Random Forest (n_estimators=150, max_depth=10) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%  
Random Forest (n_estimators=150, max_depth=15) - RMSE: 0.75, MAE: 0.67, Accuracy: 56.61%  
  
Best Random Forest Model: (150, 5)  
Best RMSE: 0.75
```

MLflow 2.19.0 Experiments Models

Pollution Prediction - Random Forest > nosy-shrew-658

Overview Model metrics System metrics Artifacts

Description

No description

Details

Created at	2024-12-13 22:06:14
Created by	kissa zahra
Experiment ID	329706885619224948
Status	Finished
Run ID	a5a3d27101b94c9895701bf3c84c4945
Duration	199ms
Datasets used	—
Tags	<a href="#">Add</a>
Source	<a href="#">task2.py</a> <a href="#">bz00a80</a>
Logged models	—
Registered models	—

Parameters (2)

Parameter	Value
max_depth	15
n_estimators	150

Metrics (3)

Metric	Value
Accuracy	56.613896134729494
MAE	0.6666666666666666
RMSE	0.7509271265922725

MLflow 2.19.0 Experiments Models

Pollution Prediction - Random Forest > nosy-shrew-658

Overview Model metrics System metrics Artifacts

Search metric charts

Accuracy

nosy-shrew-658	56.61
nosy-shrew-658	56.61

MAE

nosy-shrew-658	0.67
nosy-shrew-658	0.67

RMSE

nosy-shrew-658	0.75
nosy-shrew-658	0.75

**Total iterations = (3x3x1) + (3x3) = 18 iterations**

## 5. Model Evaluation

### Random Forest:

The Random Forest models with different combinations of n\_estimators and max\_depth (50, 100, 150) showed the following performance:

**RMSE:** 0.75

**MAE:** 0.67

**Accuracy:** Around 56-57%

The best Random Forest model is (150, 5) with an RMSE of 0.75.

### ARIMA:

The ARIMA models with different combinations of (p,d,q) parameters were trained, and their performance was as follows:

**ARIMA(1,1,1): RMSE:** 0.85, **MAE:** 0.72, **Accuracy:** 47.16%

**ARIMA(1,1,2): RMSE:** 0.76, **MAE:** 0.66, **Accuracy:** 52.14%

**ARIMA(1,1,3): RMSE:** 0.74, **MAE:** 0.61, **Accuracy:** 53.88%

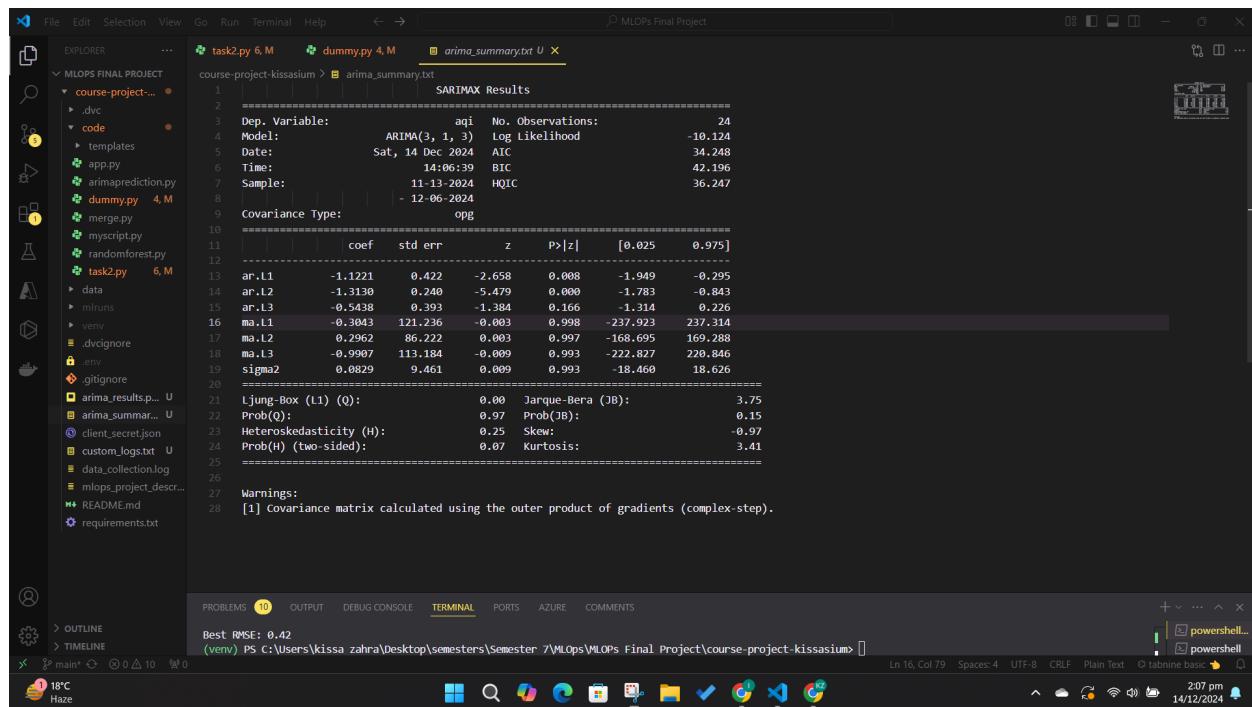
**ARIMA(2,1,1): RMSE:** 0.42, **MAE:** 0.36, **Accuracy:** 75.49% (Best model so far)

**ARIMA(2,1,2): RMSE:** 0.50, **MAE:** 0.45, **Accuracy:** 68.81%

**ARIMA(2,1,3): RMSE:** 0.55, **MAE:** 0.43, **Accuracy:** 72.19%

ARIMA model is giving better performance as compare to random forest model with an RMSE of 0.42, MAE of 0.36, and Accuracy of 75.49%.

### Model matrix



SARIMAX Results

	Dep. Variable:	aqi	No. Observations:	24		
Model:	ARIMA(3, 1, 3)	Log Likelihood	-10.124			
Date:	Sat, 14 Dec 2024	AIC	34.248			
Time:	14:06:39	BIC	42.196			
Sample:	11-13-2024	HQIC	36.247			
Covariance Type:	<td></td> <td></td>					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-1.1221	0.422	-2.658	0.008	-1.949	-0.295
ar.L2	-1.3130	0.240	-5.479	0.000	-1.783	-0.843
ar.L3	-0.5438	0.393	-1.384	0.166	-1.314	0.226
ma.L1	-0.3043	121.236	-0.003	0.998	-237.923	237.314
ma.L2	0.2962	86.222	0.003	0.997	-168.695	169.288
ma.L3	-0.9907	113.184	-0.009	0.993	-222.827	220.846
sigma2	0.0829	9.461	0.009	0.993	-18.468	18.626

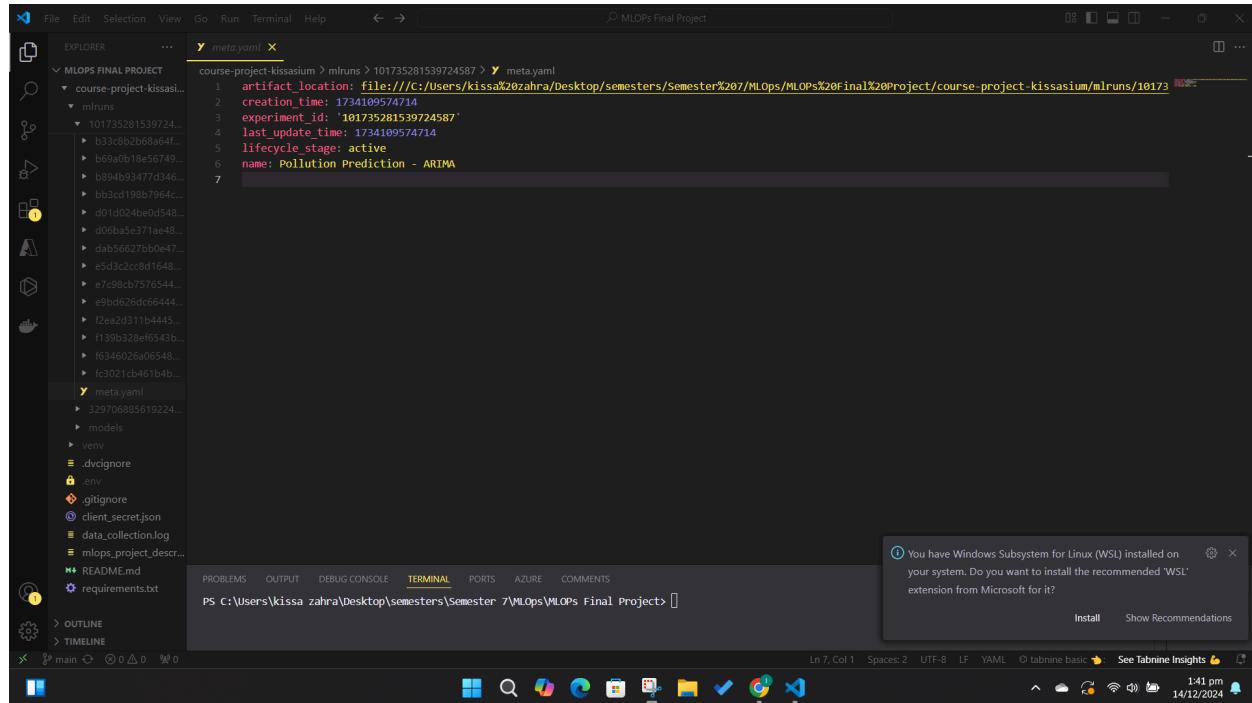
Ljung-Box (L1) (Q): 0.00 Jarque-Bera (JB): 3.75  
Prob(Q): 0.97 Prob(JB): 0.15  
Heteroskedasticity (H): 0.25 Skew: -0.97  
Prob(H) (two-sided): 0.07 Kurtosis: 3.41

Warnings:  
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

# Information about my model

## In meta.yaml file

### Arima

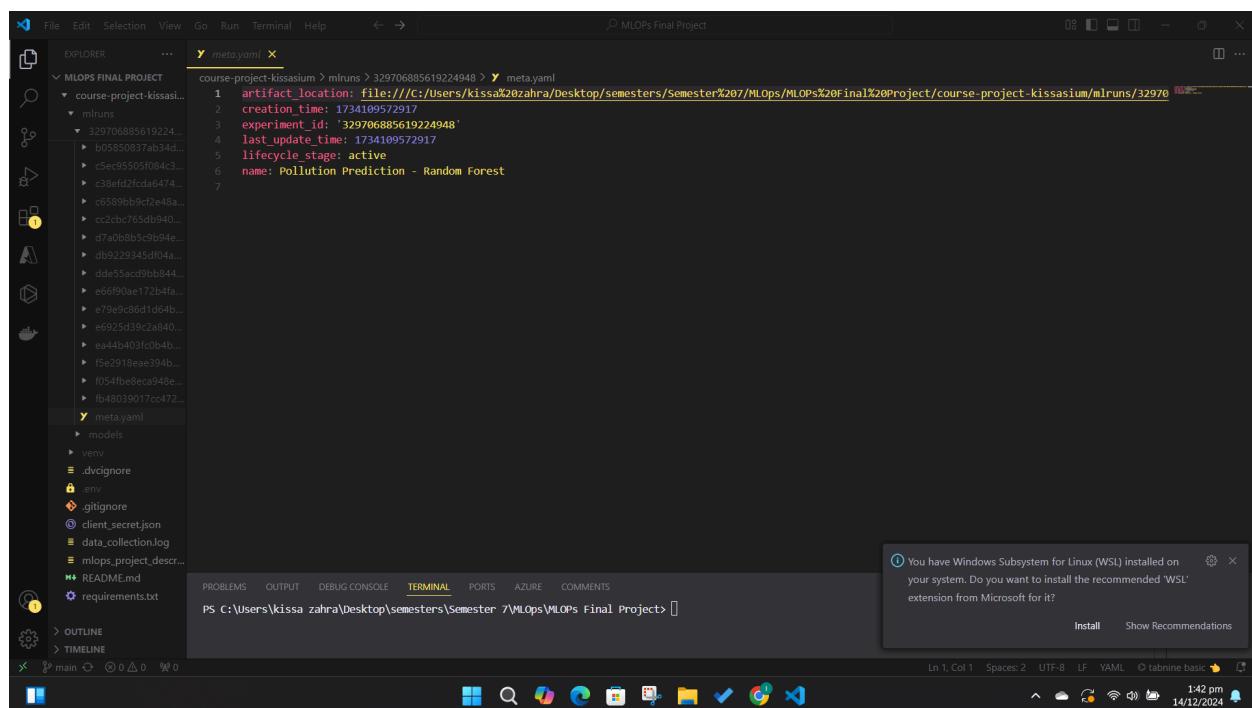


The screenshot shows the VS Code interface with the 'MLOps Final Project' workspace open. The Explorer sidebar shows a tree structure of project files, including 'course-project-kissasium', 'mlruns', and 'meta.yaml'. The 'meta.yaml' file is selected and displayed in the main editor area:

```
course-project-kissasium > mlruns > 101735281539724587 > meta.yaml
1 artifact_location: file:///C:/Users/kissa%20zahra/Desktop/semesters/Semester%207/MLOps/MLOps%20Final%20Project/course-project-kissasium/mlruns/101735281539724587
2 creation_time: 1734109574714
3 experiment_id: '101735281539724587'
4 last_update_time: 1734109574714
5 lifecycle_stage: active
6 name: Pollution Prediction - ARIMA
7
```

The terminal tab at the bottom shows a PowerShell prompt: PS C:\Users\kizza zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project>. A tooltip in the center right of the interface suggests installing the 'WSL' extension.

## Random Forest



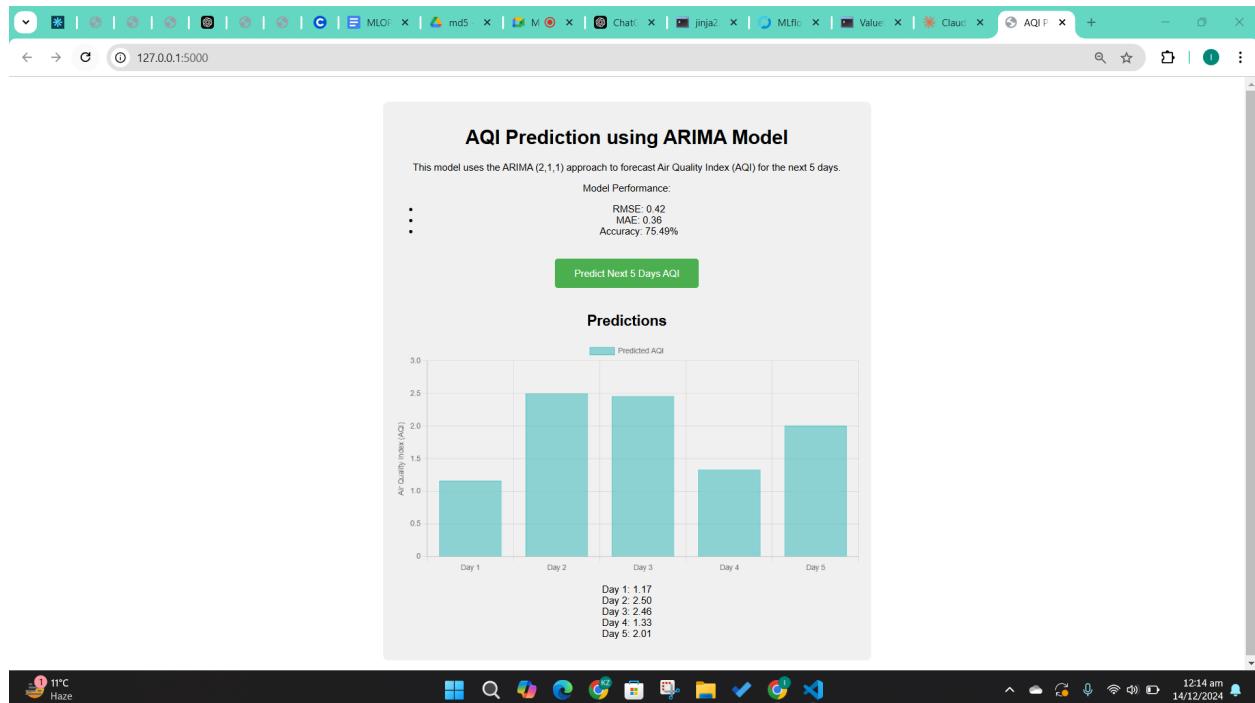
The screenshot shows the VS Code interface with the 'MLOps Final Project' workspace open. The Explorer sidebar shows a tree structure of project files, including 'course-project-kissasium', 'mlruns', and 'meta.yaml'. The 'meta.yaml' file is selected and displayed in the main editor area:

```
course-project-kissasium > mlruns > 329706885619224948 > meta.yaml
1 artifact_location: file:///C:/Users/kizza%20zahra/Desktop/semesters/Semester%207/MLOps/MLOps%20Final%20Project/course-project-kissasium/mlruns/329706885619224948
2 creation_time: 1734109572917
3 experiment_id: '329706885619224948'
4 last_update_time: 1734109572917
5 lifecycle_stage: active
6 name: Pollution Prediction - Random Forest
7
```

The terminal tab at the bottom shows a PowerShell prompt: PS C:\Users\kizza zahra\Desktop\semesters\Semester 7\MLOps\MLOps Final Project>. A tooltip in the center right of the interface suggests installing the 'WSL' extension.

## 6. Deployment

This model will predict the AQI value for next 5 days, based on the fetched data.

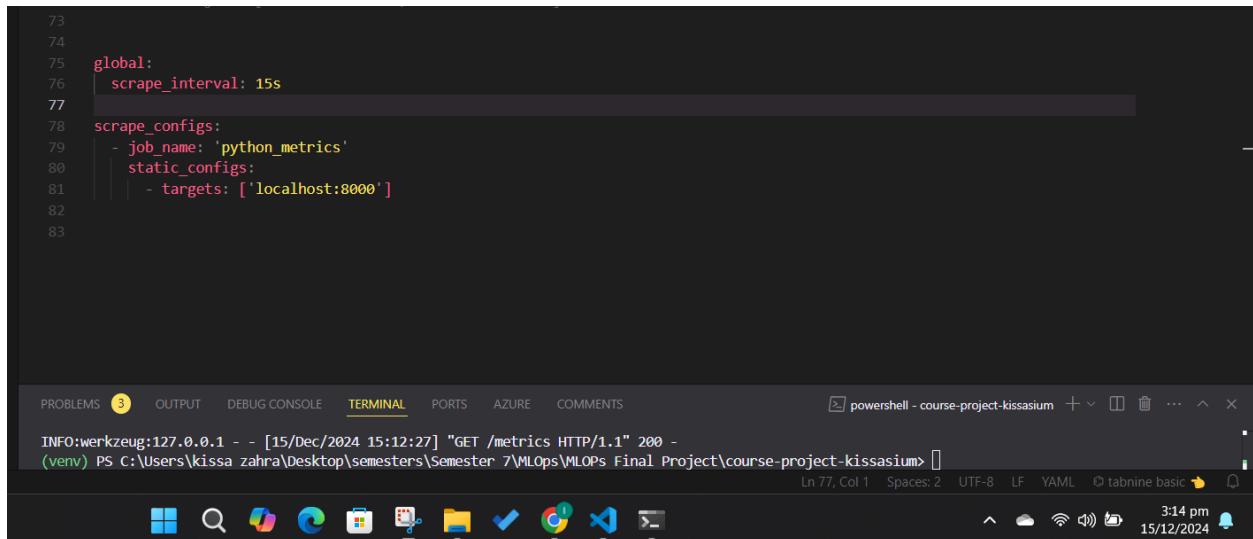


## Task 3: Monitoring and Live Testing

Objective: Test the pipeline with live data and monitor the deployed system.

### 1. Set Up Monitoring

Updating prometheus file to:



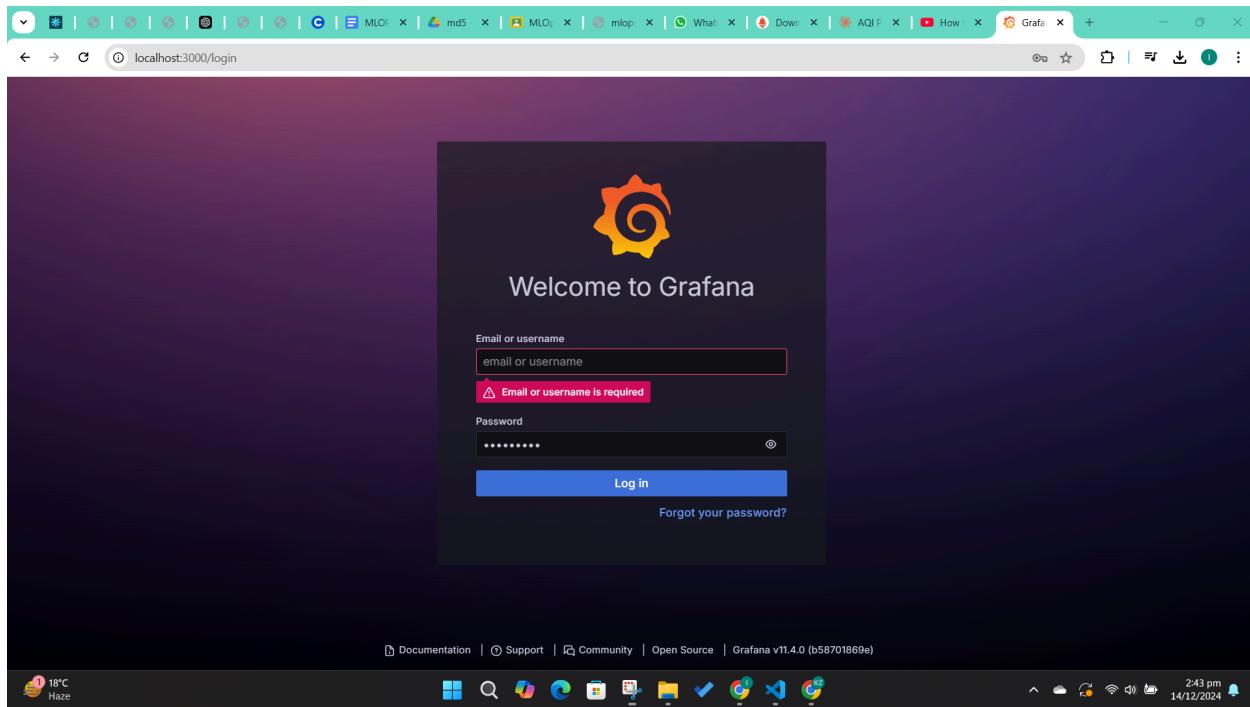
```
73
74
75 global:
76   scrape_interval: 15s
77
78 scrape_configs:
79   - job_name: 'python_metrics'
80     static_configs:
81       - targets: ['localhost:8000']
82
83
```

The screenshot shows a terminal window in VS Code with the following content:

```
INFO:werkzeug:127.0.0.1 - - [15/Dec/2024 15:12:27] "GET /metrics HTTP/1.1" 200 -
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLops\MLops Final Project\course-project-kissarium> []
```

The terminal tab is selected at the top. The status bar at the bottom right shows the date and time: 3:14 pm, 15/12/2024.

rnd ran it.



pass: admin or admin1

Search or jump to... ctrl+k

prometheus

Type: Prometheus

Alerting Supported

Explore data Build a dashboard

Configure your Prometheus data source below  
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-forever Grafana Cloud plan](#).

Name: prometheus Default:

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

Fields marked with \* are required

Connection

Prometheus server URL \* http://localhost:9090

Cache level: Low

Incremental querying (beta): Off

Disable recording rules (beta): Off

Custom query parameters: Example: max\_source\_resolution=5m&timeout

HTTP method: POST

**Exemplars**

+ Add

✓ Successfully queried the Prometheus API.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

Delete Save & test

## Updated the prometheus.yml file with the code:

```

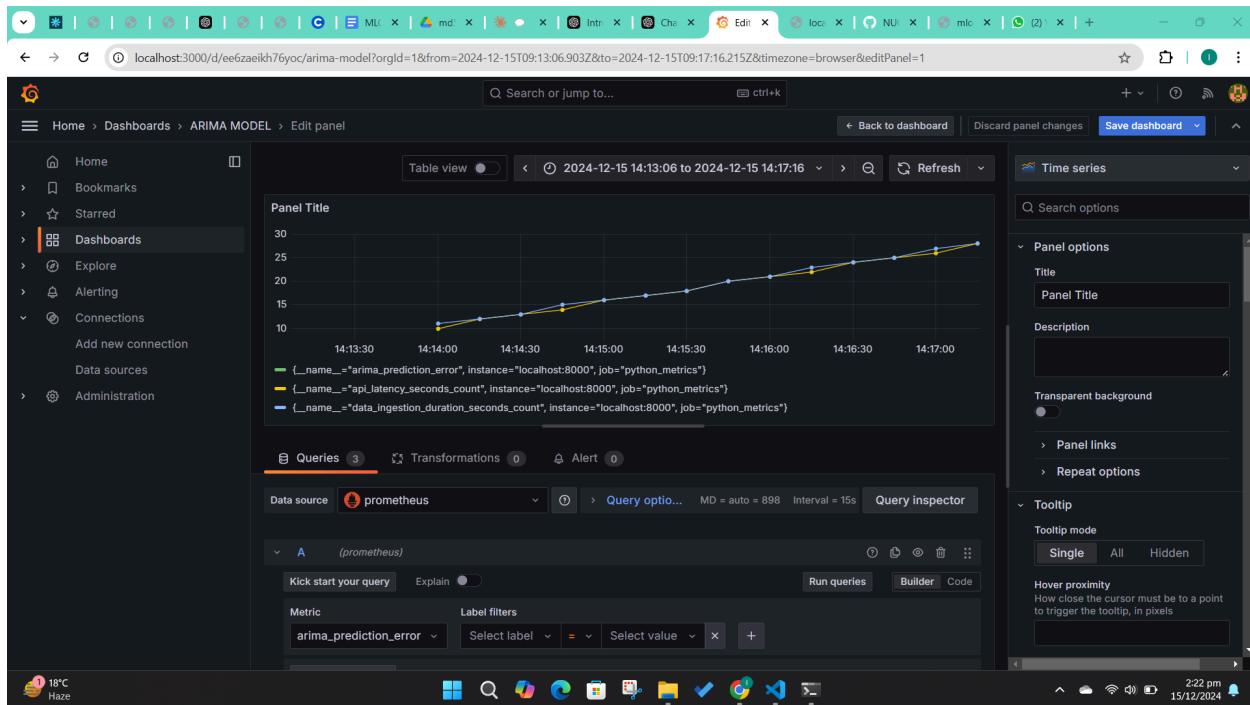
global:
  scrape_interval: 15s
  evaluation_interval: 15s

alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - localhost:9090

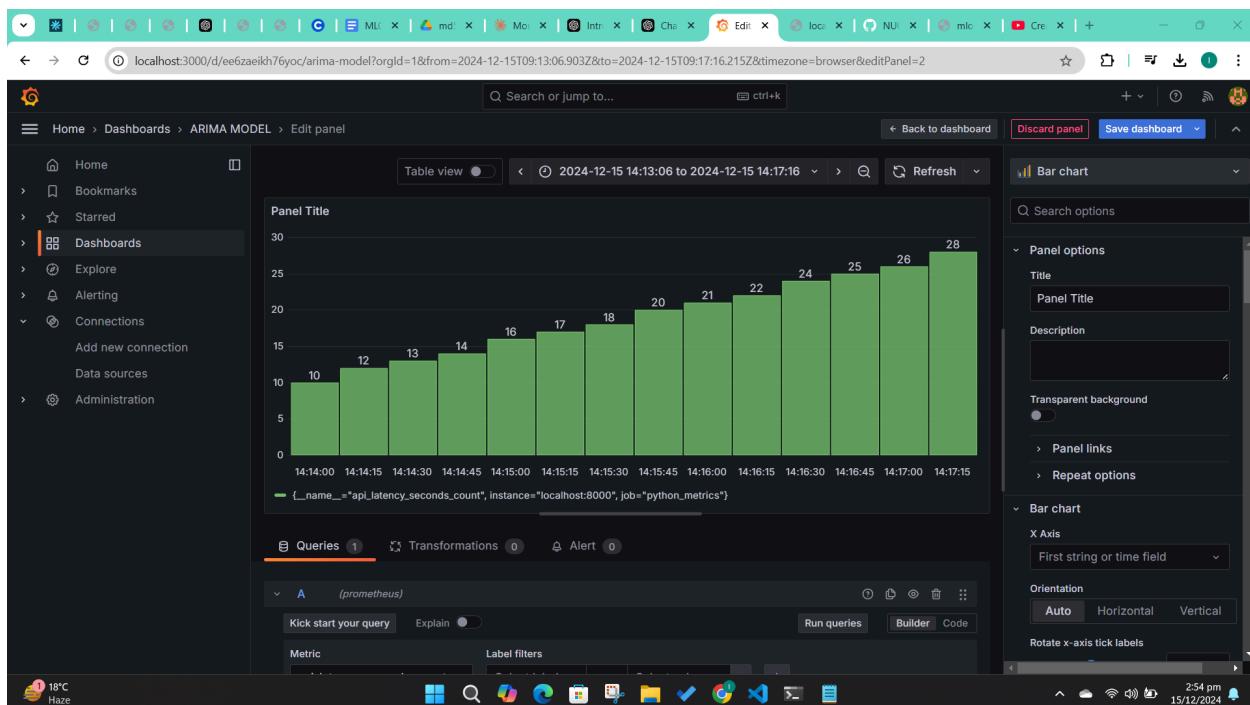
rule_files:
  - scrape_configs:
      # Prometheus itself
      - job_name: "prometheus"
        static_configs:
          - targets: ["localhost:9090"]

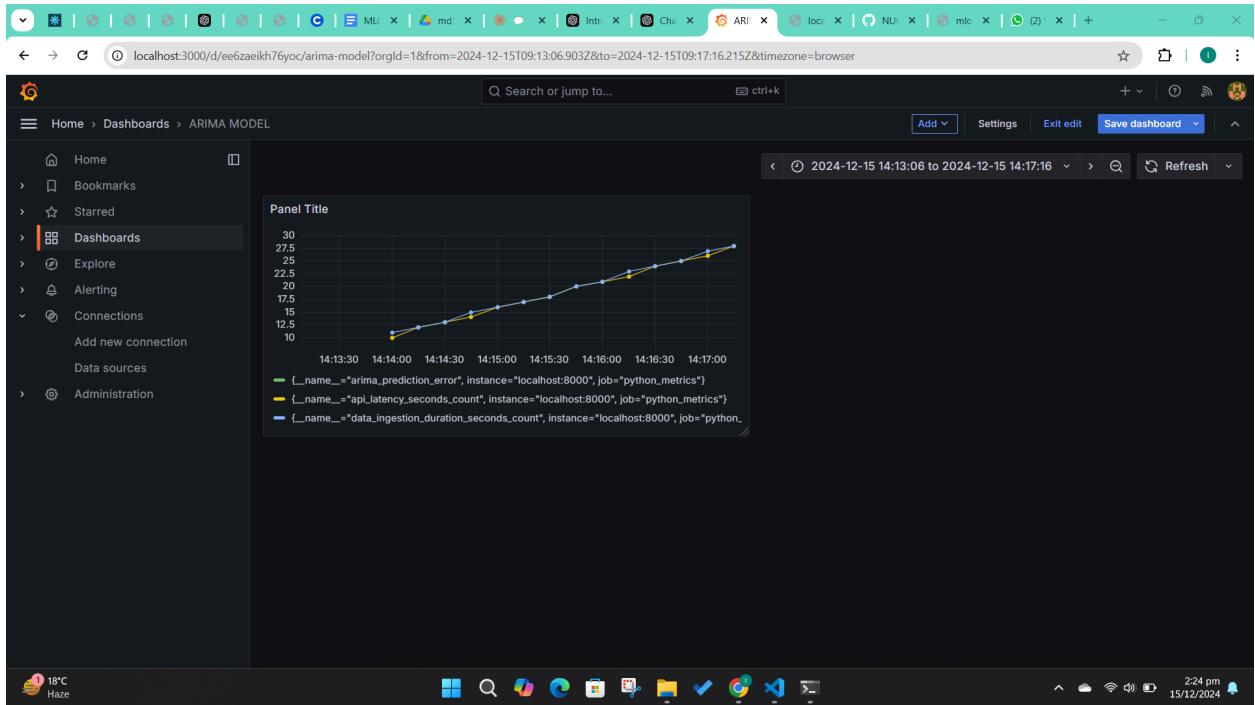
      # Air Quality Monitoring Application
      - job_name: "air_quality_monitoring"
        static_configs:
          - targets: ["localhost:8000"]

```



## API\_LATENCY\_SECOND\_COUNT





```
# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 1269.0
python_gc_objects_collected_total{generation="1"} 263.0
python_gc_objects_collected_total{generation="2"} 0.0
# HELP python_gc_objects_uncollectable_total Uncollectable objects found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 238.0
python_gc_collections_total{generation="1"} 21.0
python_gc_collections_total{generation="2"} 1.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="12",patchlevel="6",version="3.12.6"} 1.0
# HELP data_ingestion_duration_seconds Time spent on data ingestion
# TYPE data_ingestion_duration_seconds summary
data_ingestion_duration_seconds_count 631
data_ingestion_duration_seconds_sum 0.033673001715286
# HELP data_ingestion_duration_seconds_created Time spent on data ingestion
# TYPE data_ingestion_duration_seconds_created gauge
data_ingestion_duration_seconds_created 1.7342539210514038e+09
# HELP api_latency_seconds Time taken for API requests
# TYPE api_latency_seconds summary
api_latency_seconds_count 62.0
api_latency_seconds_sum 62.025781001592
# HELP api_latency_seconds_created Time taken for API requests
# TYPE api_latency_seconds_created gauge
api_latency_seconds_created 1.7342539210514038e+09
# HELP arima_prediction_error Prediction error of ARIMA model
# TYPE arima_prediction_error gauge
arima_prediction_error NaN
```

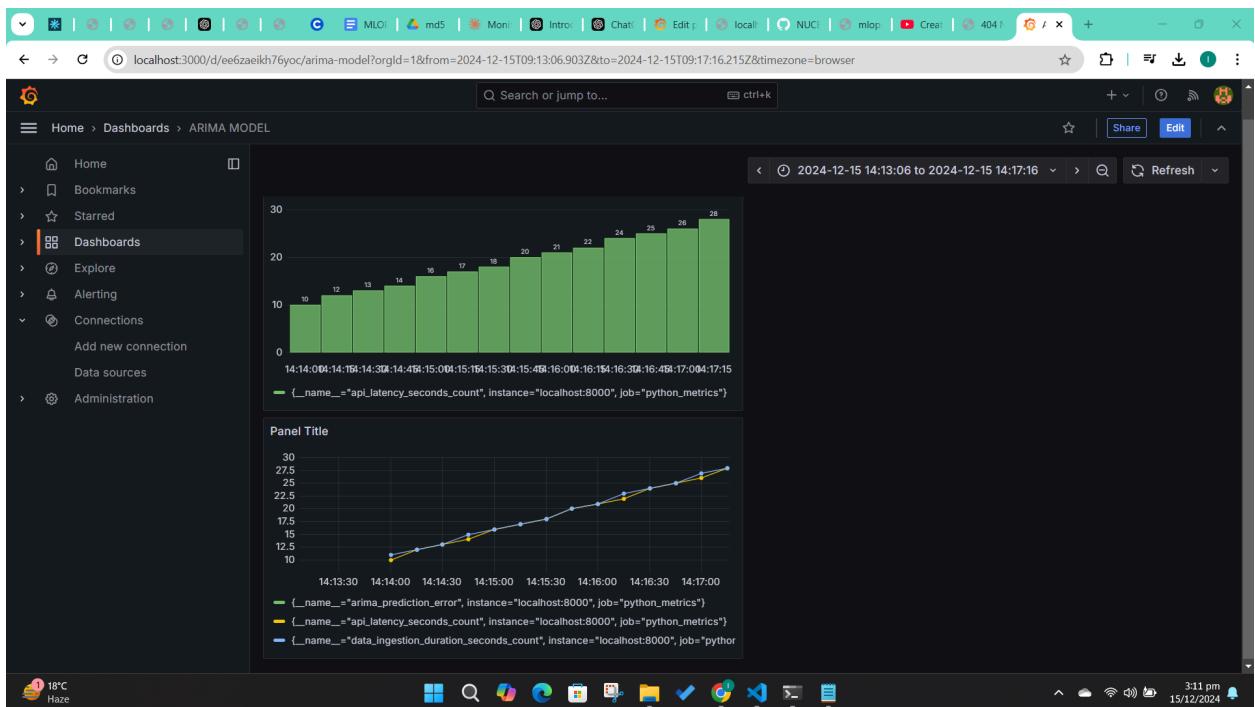
**Information present in localhost:8000**

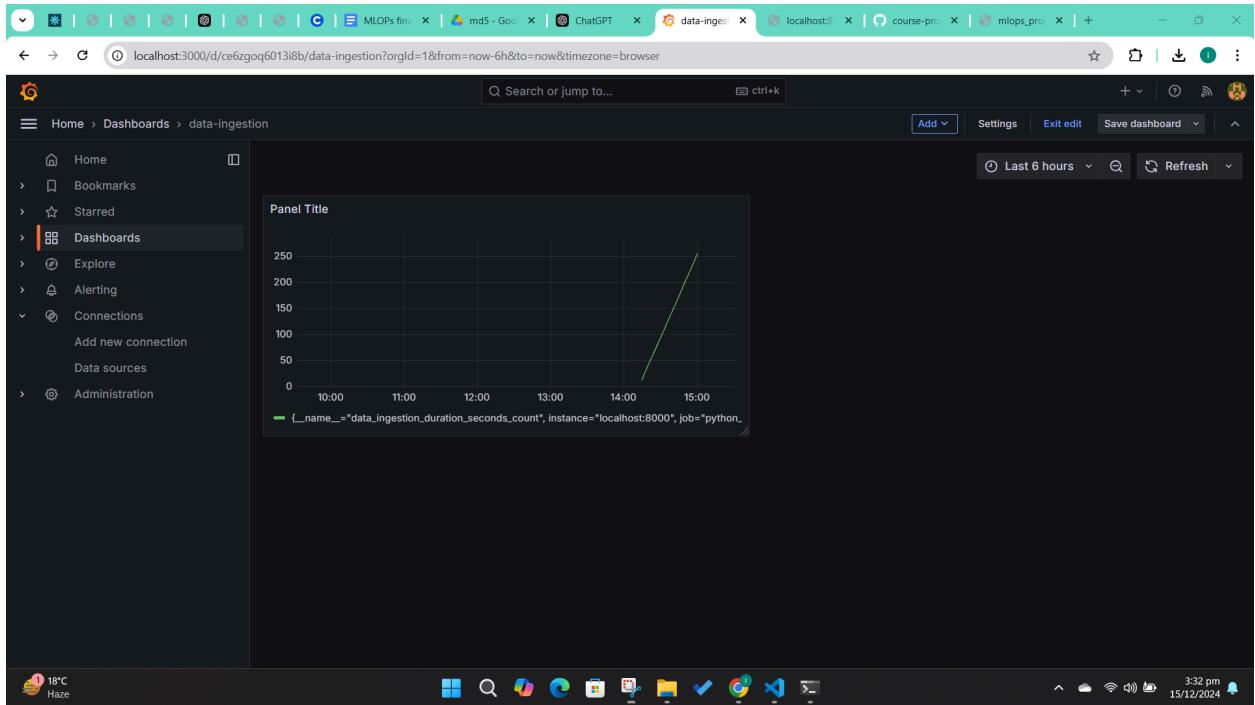
```

# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 4217.0
python_gc_objects_collected_total{generation="1"} 1253.0
python_gc_objects_collected_total{generation="2"} 150.0
# HELP python_gc_objects_uncollectable_total Uncollectable objects found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 265.0
python_gc_collections_total{generation="1"} 24.0
python_gc_collections_total{generation="2"} 2.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython", major="3", minor="12", patchlevel="6", version="3.12.6"} 1.0
# HELP data_ingestion_duration_seconds Time spent on data ingestion
# TYPE data_ingestion_duration_seconds summary
data_ingestion_duration_seconds_count 244.0
data_ingestion_duration_seconds_sum 1.2842638010624796
# HELP data_ingestion_duration_seconds_created Time spent on data ingestion
# TYPE data_ingestion_duration_seconds_created gauge
data_ingestion_duration_seconds_created 1.7342539210514038e+09
# HELP api_latency_seconds_time API time taken for API requests
# TYPE api_latency_seconds_summary
api_latency_seconds_count 244.0
api_latency_seconds_sum 244.10677240067162
# HELP api_latency_seconds_created Time taken for API requests
# TYPE api_latency_seconds_created gauge
api_latency_seconds_created 1.7342539210514038e+09
# HELP arima_prediction_error Prediction error of ARIMA model
# TYPE arima_prediction_error gauge
arima_prediction_error NaN

```

**Updated after half an hour**





## 2. Test Predictions with Live Data

For this task i wrote a python script that fetches the live data from with API and ran the arima model on it, all the stats will be stored in the `live_test_predictions.log`

```

File Edit Selection View Go Run Terminal Help ← → MLOPS Final Project
EXPLORER ... task3p2.py 3 U task3p1.py 3 U live_test_predictions.log x
course-project-kissasium > live_test_predictions.log
1 2024-12-15 17:01:08,079 :INFO:Data saved successfully to data/weather_data.csv
2 2024-12-15 17:01:08,432 :INFO:Data saved successfully to data/air_quality_data.csv
3 2024-12-15 17:01:08,454 :INFO:Merged data saved to data/merged_output.csv
4 2024-12-15 17:01:09,360 :INFO:Best ARIMA model order: (2, 1, 1)
5 2024-12-15 17:01:09,361 :INFO:Predictions: 2024-12-09 2.311602
6 2024-12-10 0.971230
7 2024-12-11 2.382664
8 2024-12-12 2.708692
9 2024-12-13 1.273765
10 2024-12-14 1.851985
11 2024-12-15 2.762864
12 Freq: D, Name: predicted_mean, dtype: float64
13 2024-12-15 17:01:09,361 :INFO:Mean Squared Error: 0.605389888648135
14 2024-12-15 17:01:09,360 :INFO:Mean Absolute Error: 0.5575652879030177
15 2024-12-15 17:01:09,360 :INFO:Accuracy: 50.48657870926373%
16 2024-12-15 17:01:09,360 :INFO:Sleeping for 5 minutes
17 2024-12-15 17:10:06,458 :INFO:Data saved successfully to data/weather_data.csv
18 2024-12-15 17:10:06,852 :INFO:Data saved successfully to data/air_quality_data.csv
19 2024-12-15 17:10:06,883 :INFO:Merged data saved to data/merged_output.csv
20 2024-12-15 17:10:07,441 :INFO:Best ARIMA model order: (2, 1, 1)
21 2024-12-15 17:10:07,442 :INFO:Predictions: 2024-12-11 2.640907
22 2024-12-12 2.497024
23 2024-12-13 1.148307
24 2024-12-14 2.113067
25 2024-12-15 2.695258
26 2024-12-16 1.487672
27 2024-12-17 1.728391
28 Freq: D, Name: predicted_mean, dtype: float64
29 2024-12-15 17:10:07,442 :INFO:Mean Squared Error: 0.7952580033500516
30 2024-12-15 17:10:07,442 :INFO:Mean Absolute Error: 0.7594982074755929
31 2024-12-15 17:10:07,442 :INFO:Accuracy: 37.575932394549626%
32 2024-12-15 17:10:07,442 :INFO:Sleeping for 5 minutes
33
Ln 27, Col 23 Spaces: 4 UTF-8 CRLF Log tabnine basic ↻

```

### **Data Fetching:**

It fetches live data from both the OpenWeather API (for weather data) and the Air Quality API (for AQI and related metrics) every 5 minutes.

This data is saved to **data/weather\_data.csv** and **data/air\_quality\_data.csv**.

### **Data Merging:**

The data is merged on the timestamp column to create a single dataset (**data/merged\_output.csv**).

### **ARIMA Model:**

The merged data is used to train an ARIMA model. It tries different ARIMA hyperparameters (p, d, q) to find the best model.

The models performance is evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE) and Accuracy (calculated based on the MSE).

The best hyperparameters and the predictions are **logged**.

### **Continuous Operation:**

The main function runs in a loop; fetching new data, merging it and testing the model every 5 minutes.

## **3. Analyze and Optimize: improvements identified or implemented.**

### **Reduced redundancy in data fetching:**

The data is fetched at regular intervals causing repeated input and these inputs can cause model overfitting and decreased accuracy. To optimize implement data validation checks to avoid saving duplicate entries.

### **Efficient ARIMA model tuning:**

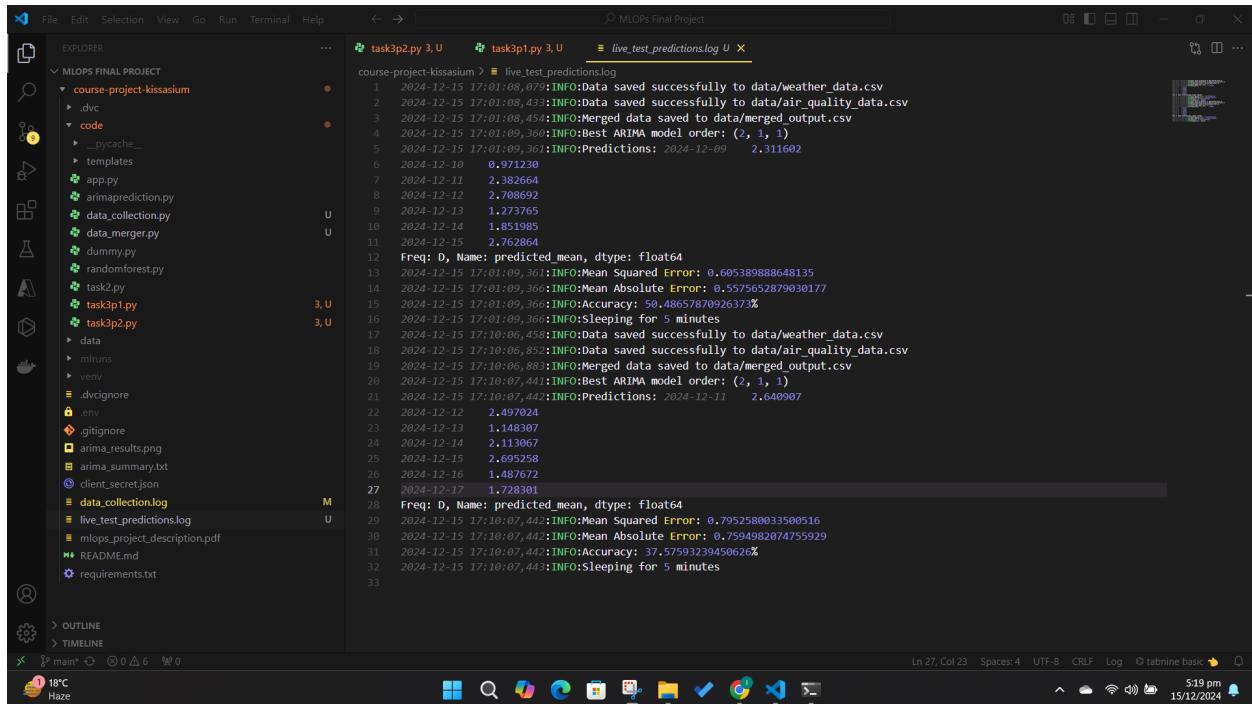
Fine tune the ARIMA model more and train it on a bigger dataset for more accuracy.

### **Parallel processing:**

Data fetching, merging and model training are performed sequentially. Parallelize these processes where possible to improve efficiency.

# Summary report on the system's live performance

From the log file we can see that the accuracy of the ARIMA model decreases with each cycle of data fetching and model training. This is because of **the lack of variation in the input data**. Since the newly fetched data is similar to the previous data the model is effectively being trained on nearly **identical inputs**. This results in the model struggling to generalize or identify new patterns causing a gradual reduction in prediction accuracy.



```
task3p2.py 3.U task3p1.py 3.U live_test_predictions.log U
course-project-kissasium > live_test_predictions.log
1 2024-12-15 17:01:08,075 :INFO:Data saved successfully to data/weather_data.csv
2 2024-12-15 17:01:08,433 :INFO:Data saved successfully to data/air_quality_data.csv
3 2024-12-15 17:01:08,454 :INFO:Merged data saved to data/merged_output.csv
4 2024-12-15 17:01:09,366 :INFO:Best ARIMA model order: (2, 1, 1)
5 2024-12-15 17:01:09,366 :INFO:Predictions: 2024-12-09 2.311602
6 2024-12-10 0.971238
7 2024-12-11 2.382664
8 2024-12-12 2.708692
9 2024-12-13 1.273765
10 2024-12-14 1.851985
11 2024-12-15 2.762864
12 Freq: D, Name: predicted_mean, dtype: float64
13 2024-12-15 17:01:09,366 :INFO:Mean Squared Error: 0.60538988648135
14 2024-12-15 17:01:09,366 :INFO:Mean Absolute Error: 0.5575652879030177
15 2024-12-15 17:01:09,366 :INFO:Accuracy: 56.48657876926373%
16 2024-12-15 17:01:09,366 :INFO:Sleeping for 5 minutes
17 2024-12-15 17:10:06,458 :INFO:Data saved successfully to data/weather_data.csv
18 2024-12-15 17:10:06,852 :INFO:Data saved successfully to data/air_quality_data.csv
19 2024-12-15 17:10:06,883 :INFO:Merged data saved to data/merged_output.csv
20 2024-12-15 17:10:07,441 :INFO:Best ARIMA model order: (2, 1, 1)
21 2024-12-15 17:10:07,442 :INFO:Predictions: 2024-12-11 2.640907
22 2024-12-12 2.497024
23 2024-12-13 1.148307
24 2024-12-14 2.113067
25 2024-12-15 2.695258
26 2024-12-16 1.487672
27 2024-12-17 1.728301
28 Freq: D, Name: predicted_mean, dtype: float64
29 2024-12-15 17:10:07,442 :INFO:Mean Squared Error: 0.7952580033500516
30 2024-12-15 17:10:07,442 :INFO:Mean Absolute Error: 0.7594982074755929
31 2024-12-15 17:10:07,442 :INFO:Accuracy: 37.57593239450626%
32 2024-12-15 17:10:07,443 :INFO:Sleeping for 5 minutes
Ln 27, Col 23 Spaces: 4 UTRF-8 CRLF Log tabnine basic 5:19 pm 15/12/2024
```