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

Task 1 Managing Environmental Data with DVC

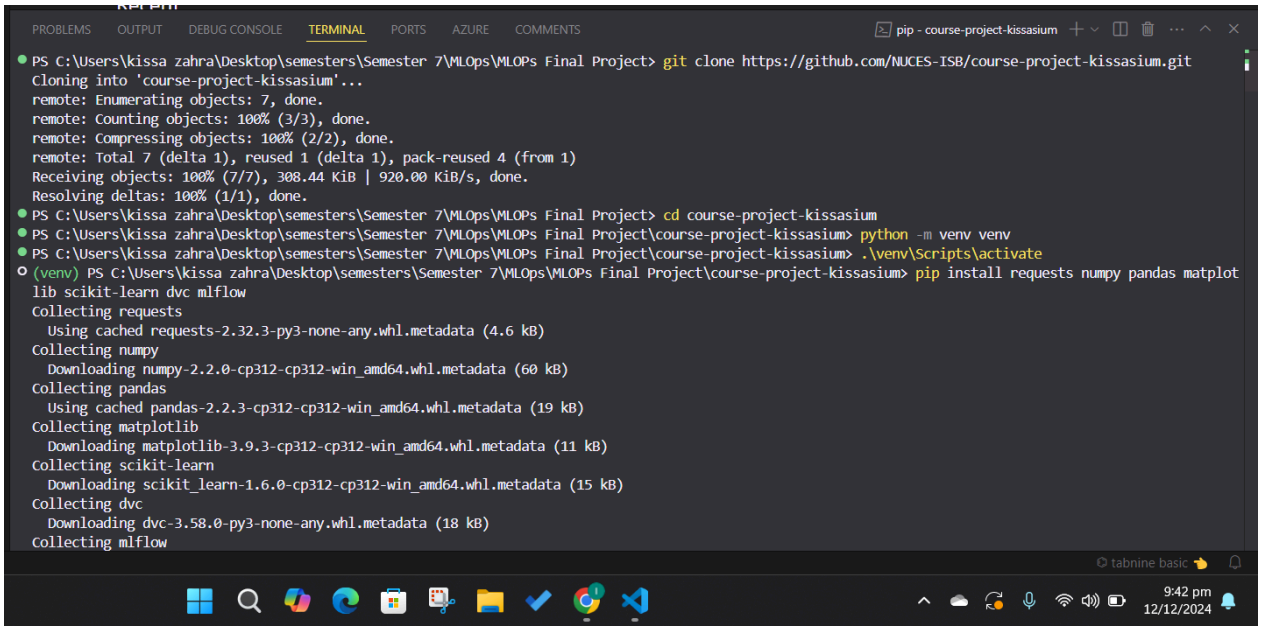
Objective: Use DVC to manage real-time environmental data streams collected from APIs.

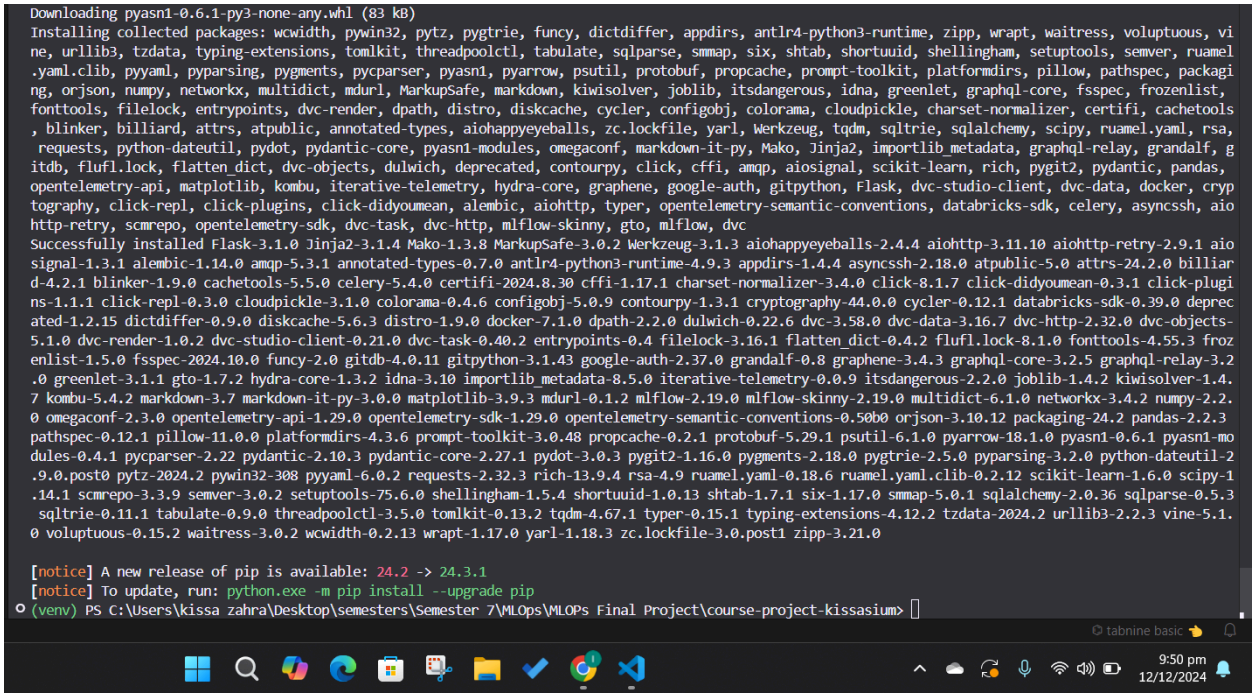
Clone the repo and run these commands on the terminal to create virtual environment.

` python -m venv venv `

` .\venv\Scripts\activate `

` pip install requests numpy pandas matplotlib scikit-learn dvc mlflow `



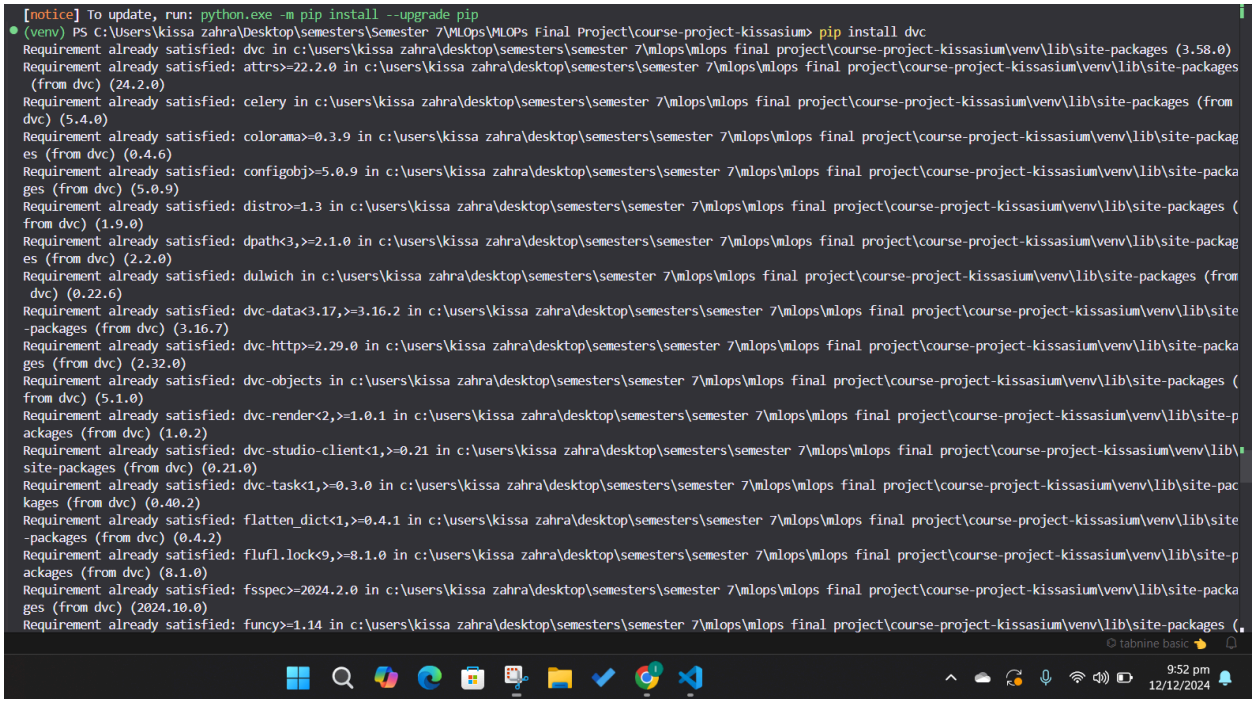


1. Research Live Data Streams

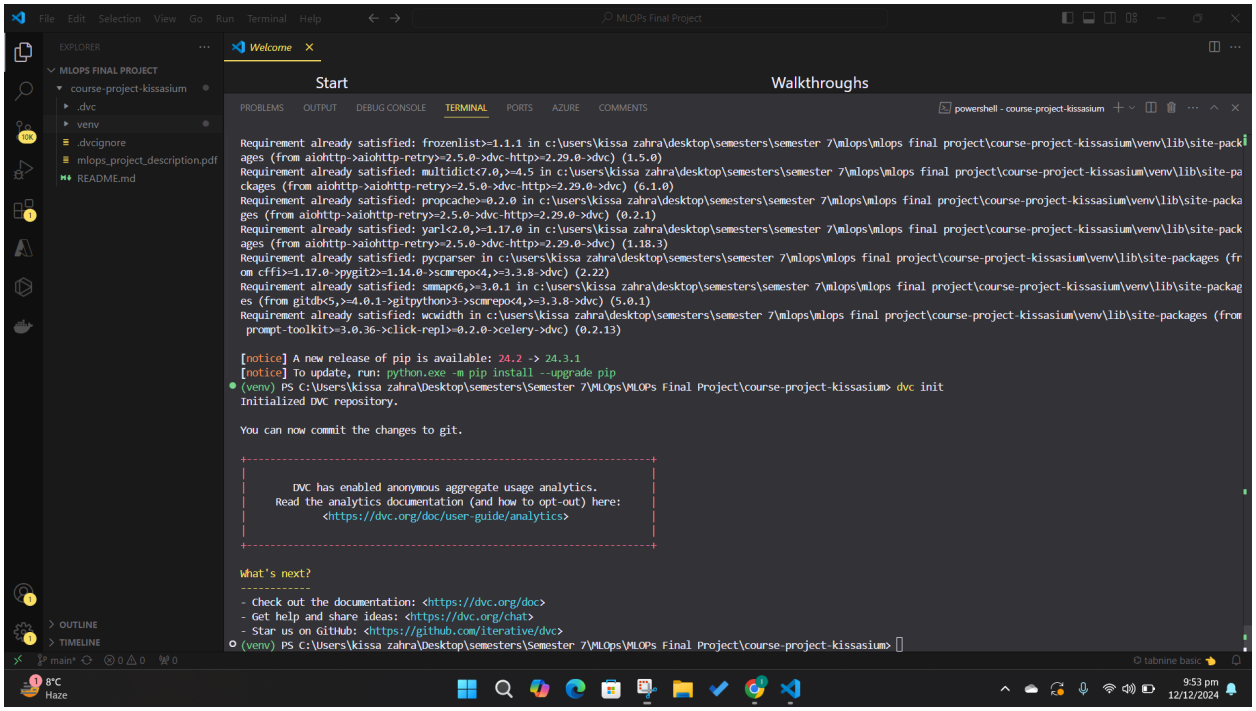
OpenWeatherMap API: Provides weather data, including temperature, humidity, and air pollution data (https://openweathermap.org/api).

2. Set Up DVC Repository

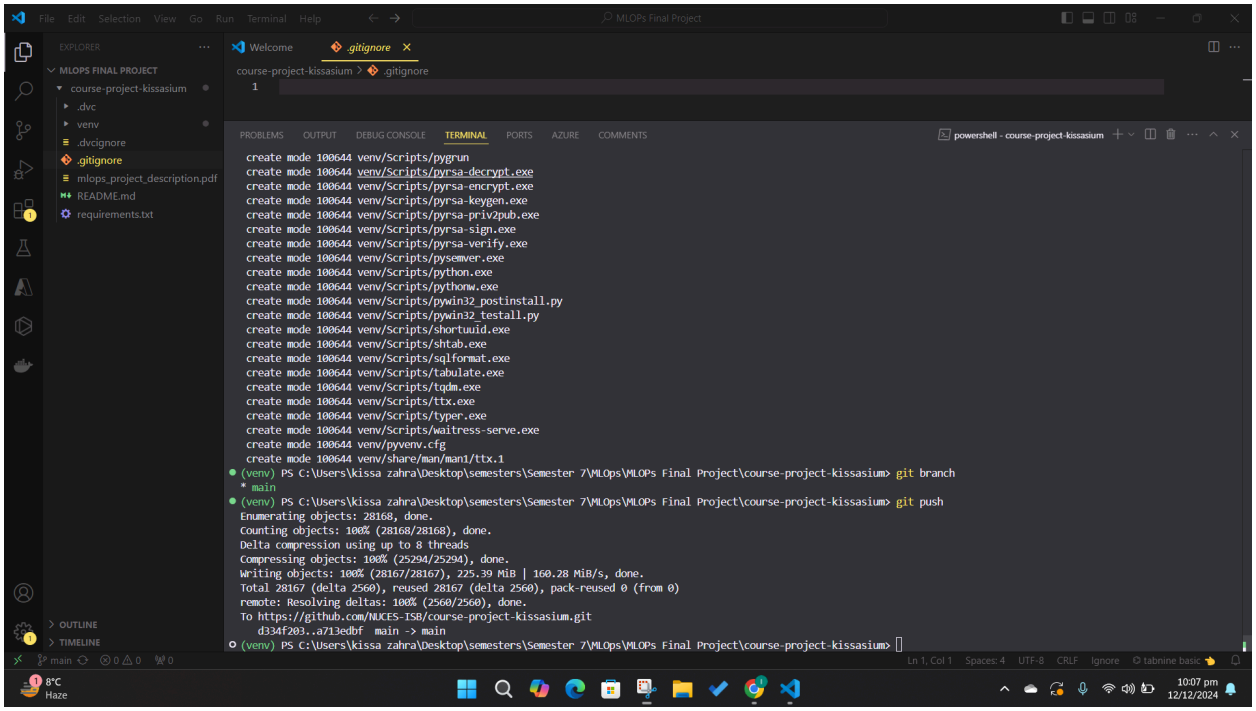
install dvc



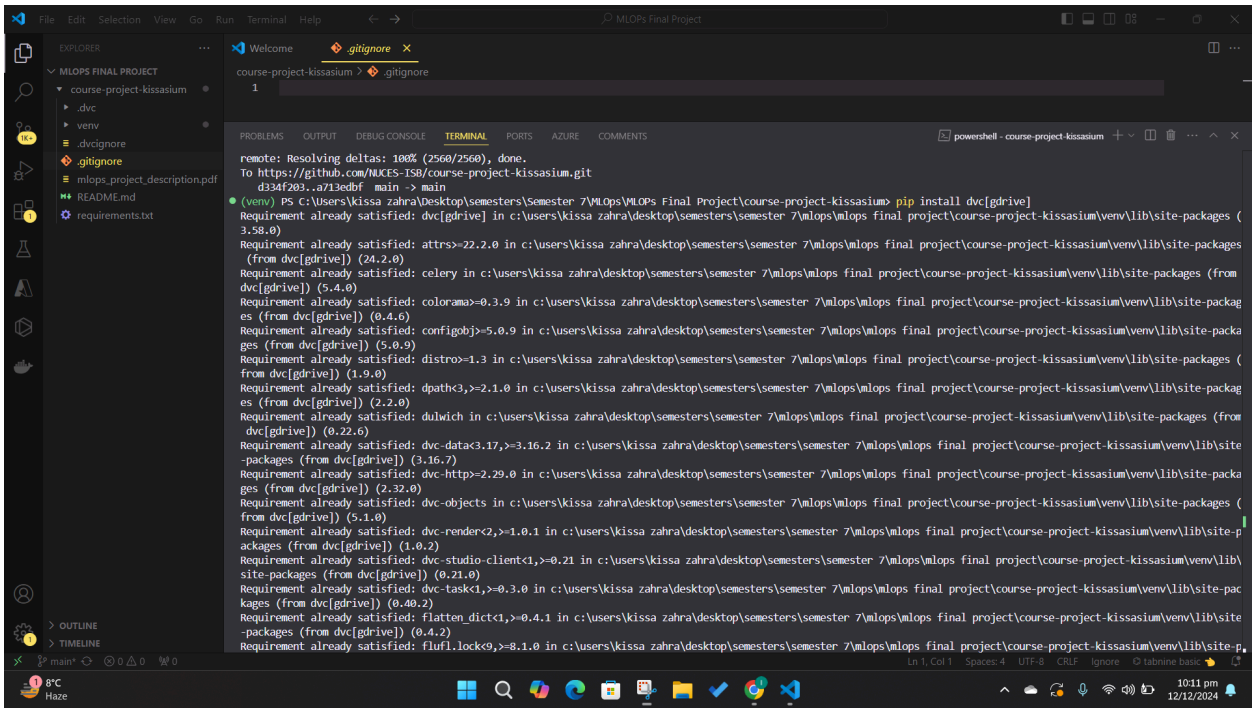
dvc init

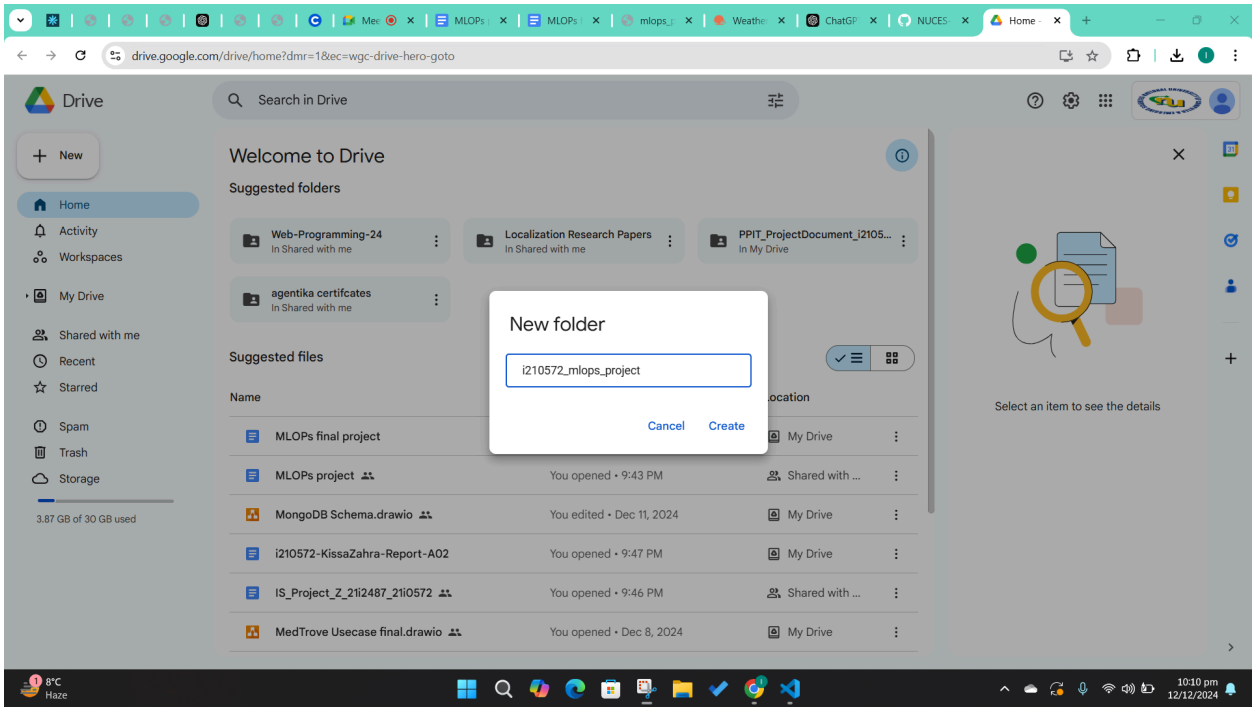


After initializing the dvc repo, I pushed it on github



3. Remote Storage Configuration

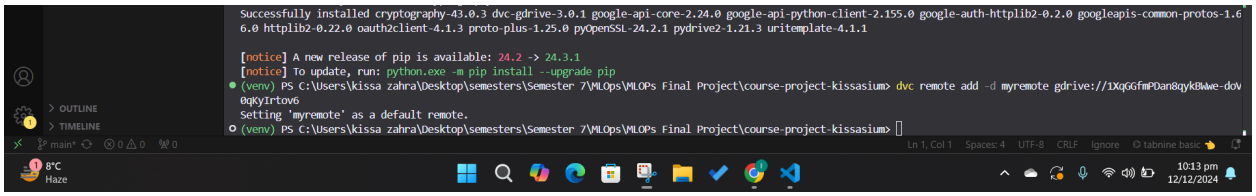




Full url of my folder:
https://drive.google.com/drive/folders/1XqGGfmPDan8qykBWwe-doV0qKyIrtov6?dmr=1&ec=wgc-drive-hero-goto

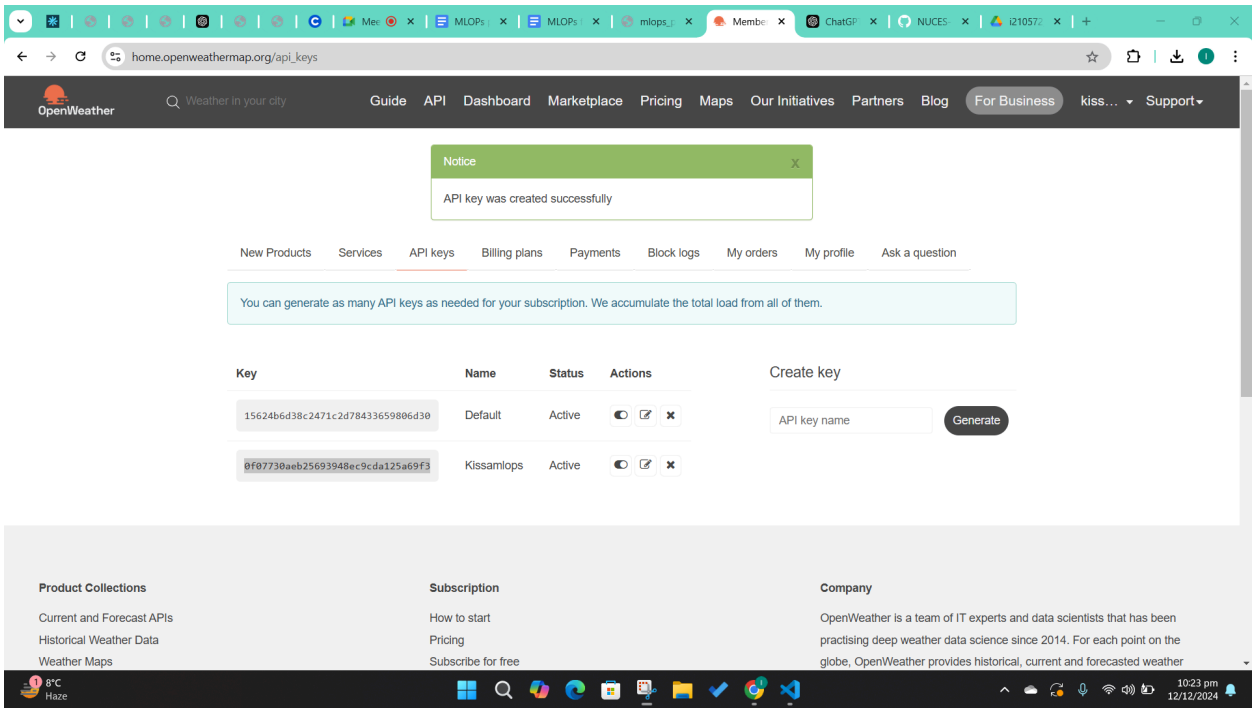
ID of the folder: 1XqGGfmPDan8qykBWwe-doV0qKyIrtov6

My remote drive: dvc remote add -d myremote gdrive://1XqGGfmPDan8qykBWwe-doV0qKyIrtov6



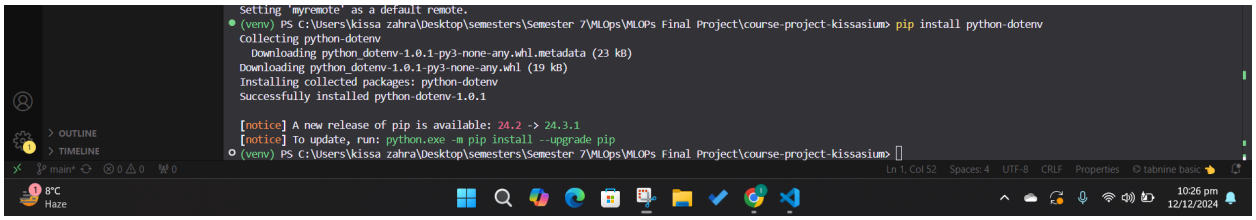
4. Data Collection Script

Creating keys



I put my key in the .env file

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

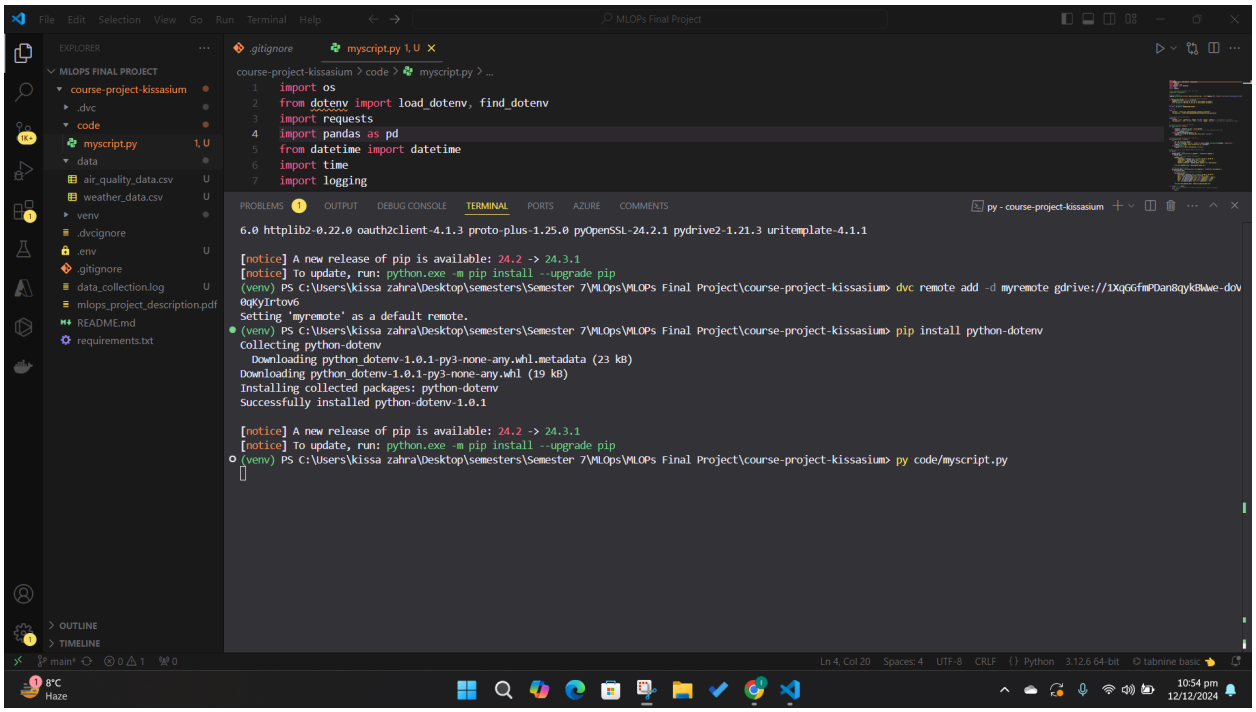


I made 2 folders; code and data.

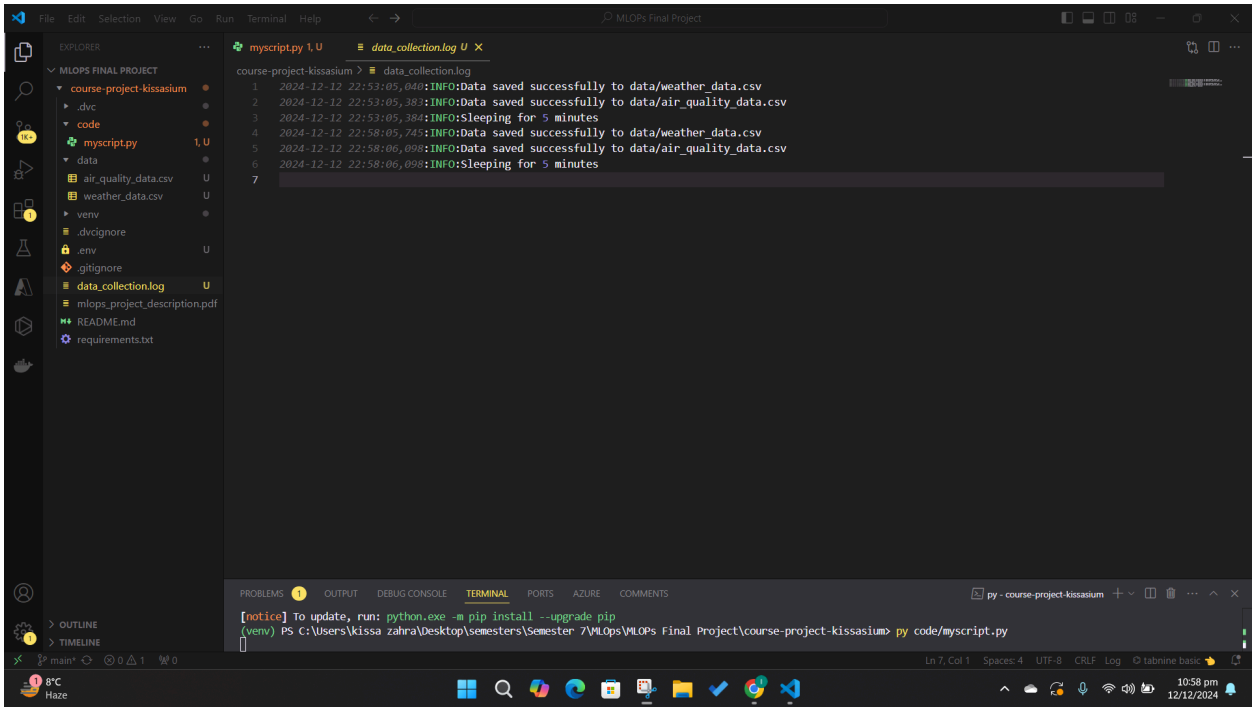
In the code folder I put my code in code/data_collection.py, it automatically collects weather and air quality data from the OpenWeatherMap API. It fetches the weather data including temperature, humidity, and weather conditions ,and the air quality data, including AQI and pollutants like PM2.5 and NO2. 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.**

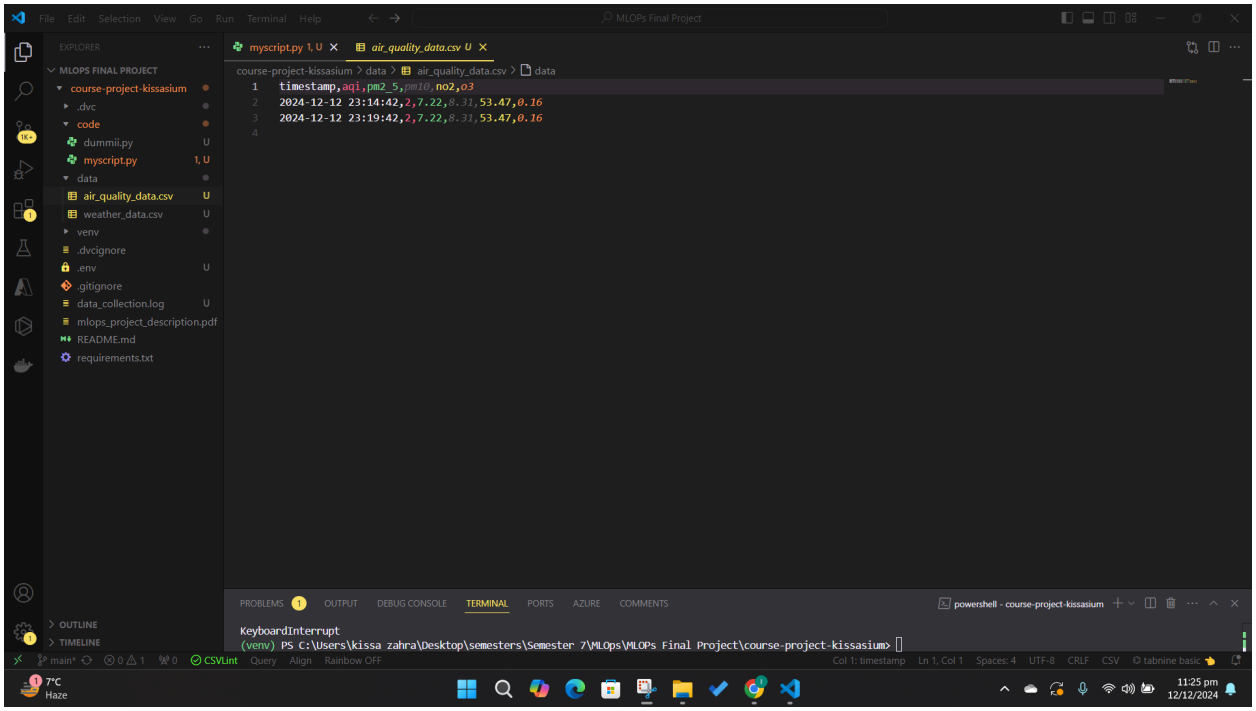
New files have been created inside the data folder; **air_quality_data.csv and weather_data.csv**



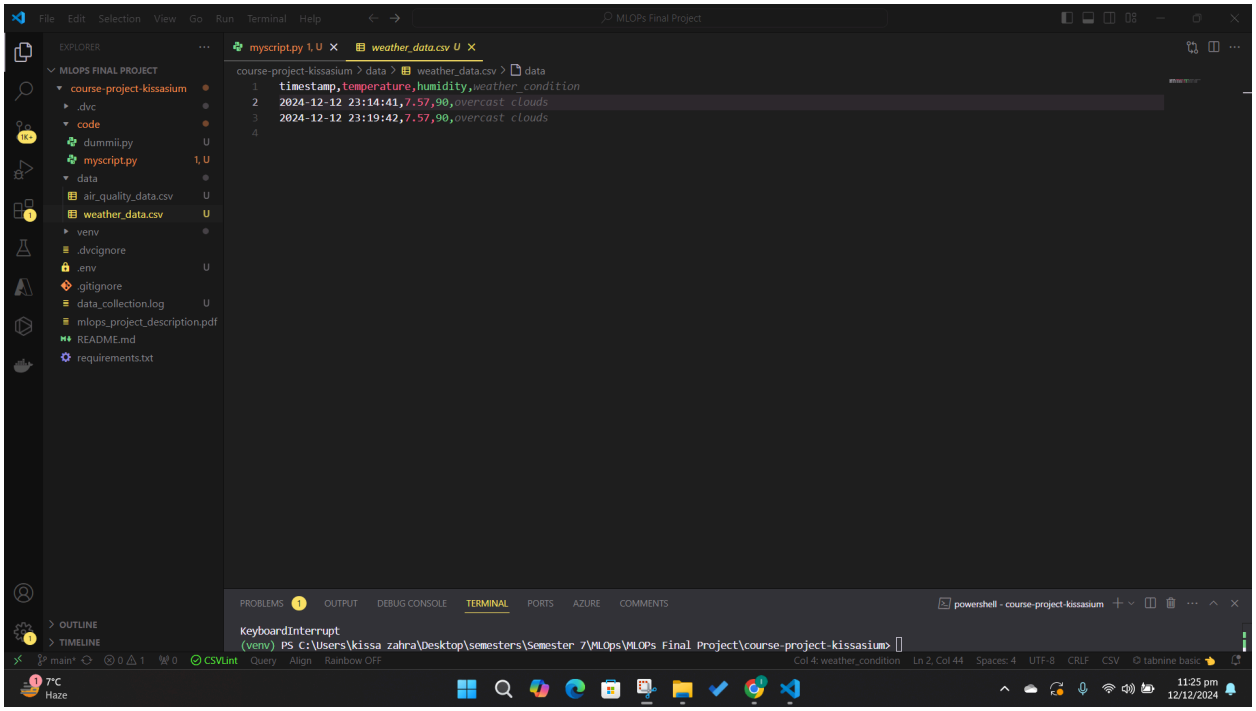
data_collection.log



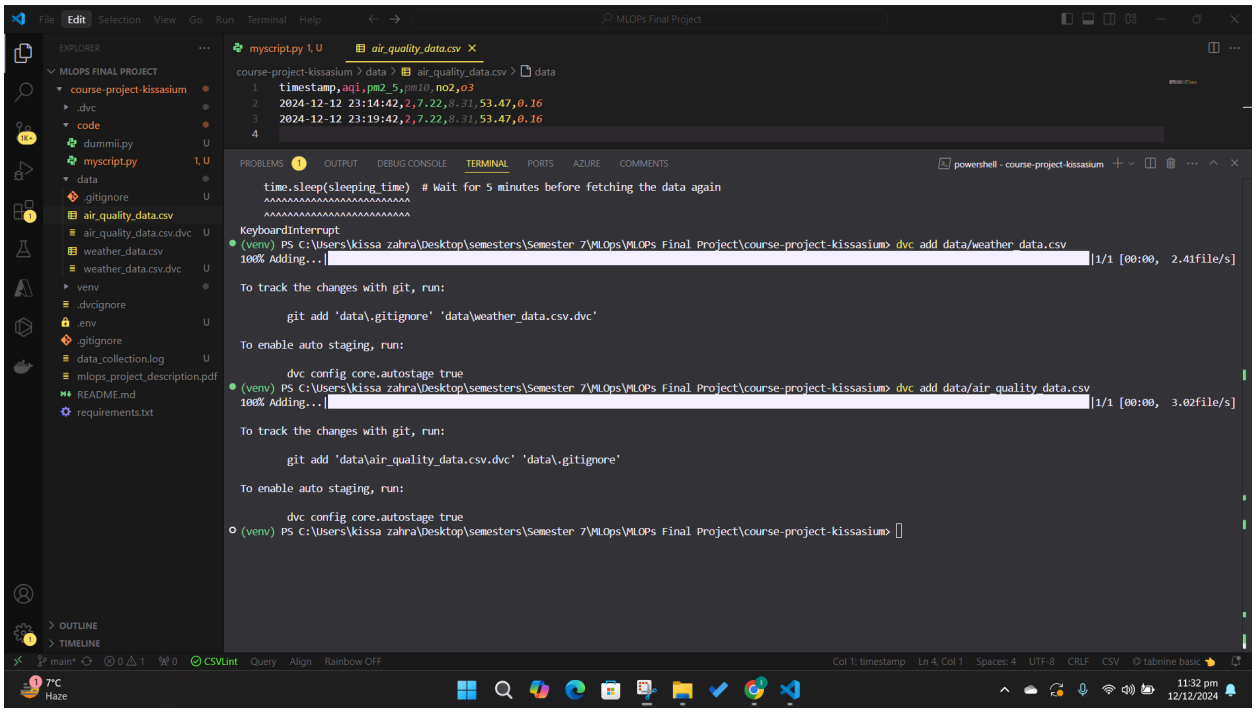
air_quality_data.csv

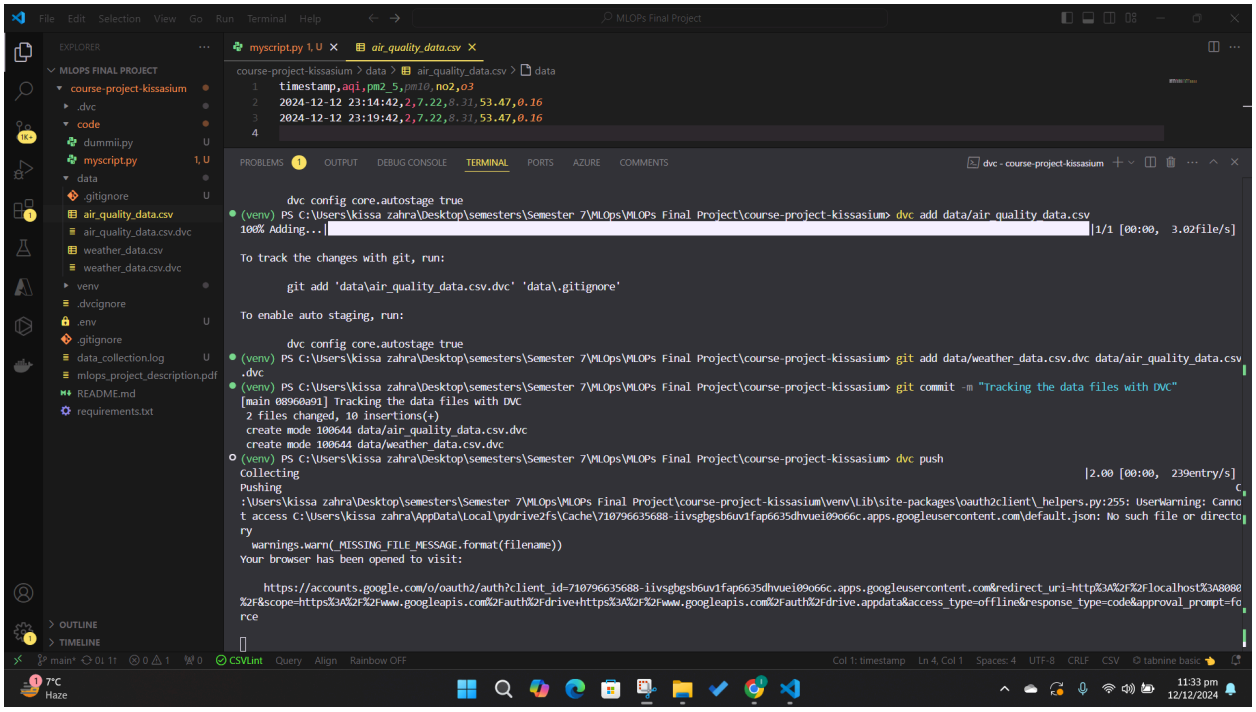


weather_data.csv



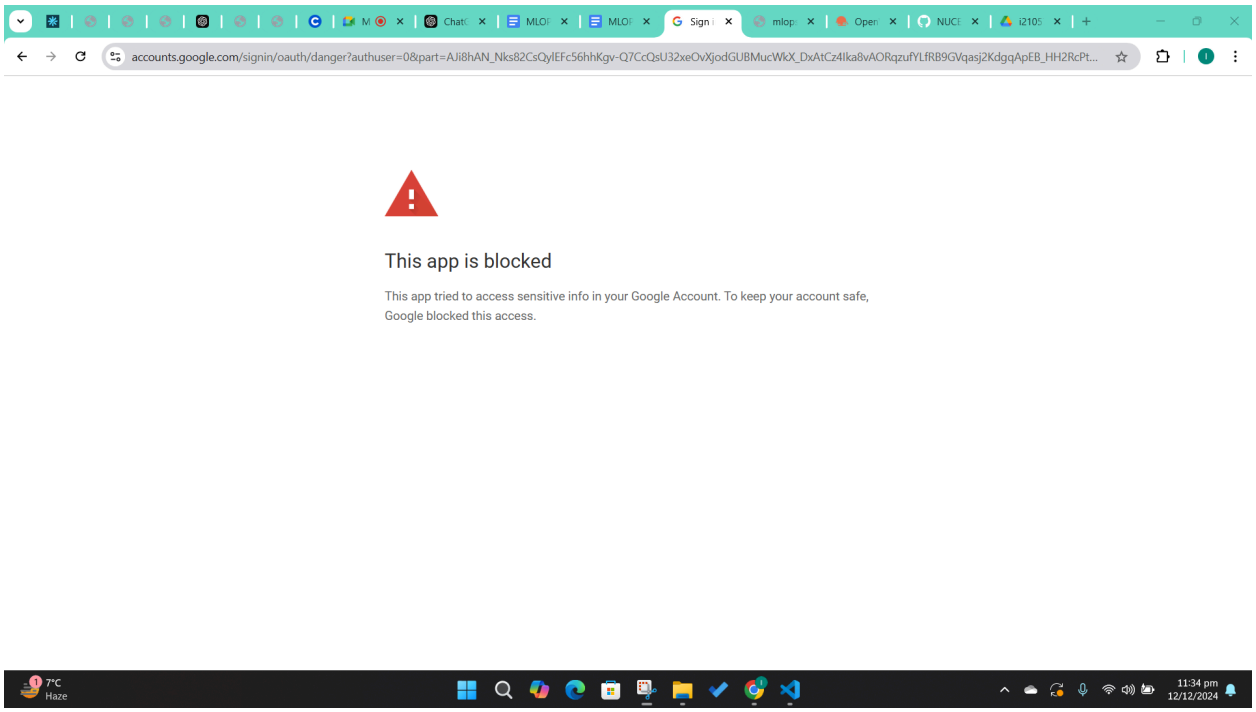
5. Version Control with DVC





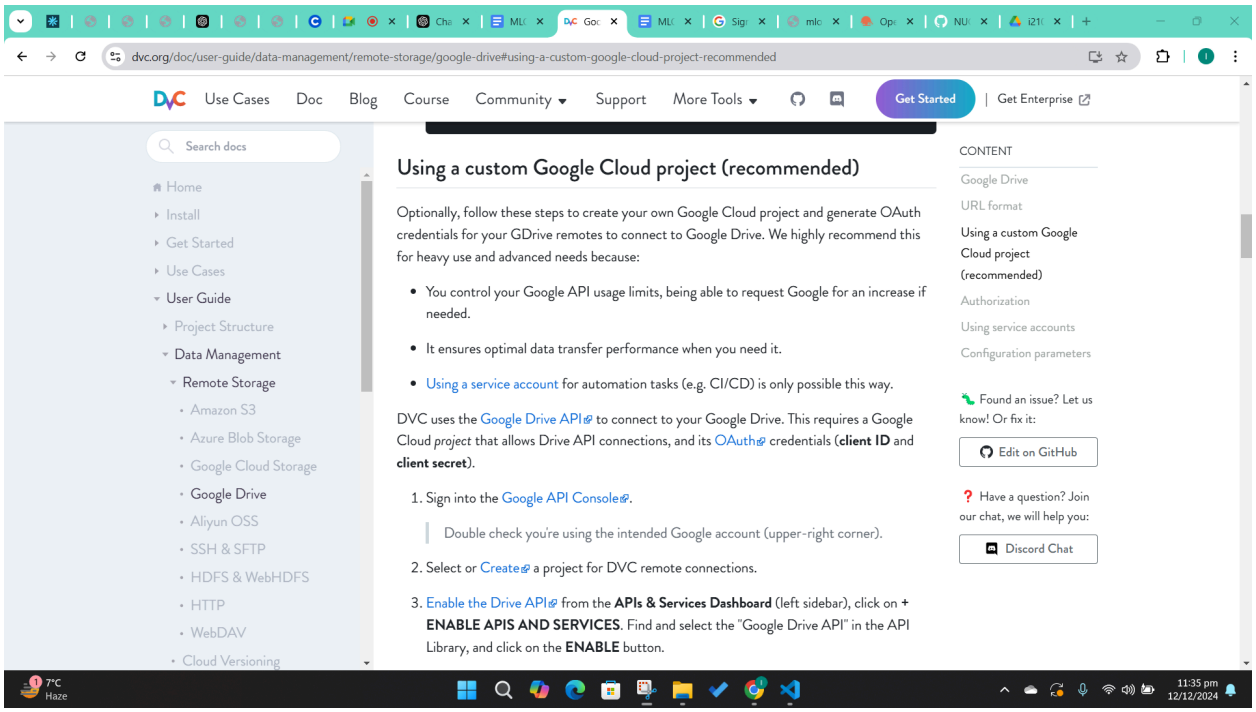
(venv) PS C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPs Final Project\course-project-kissasium> dvc push
Collecting
|2.00 [00:00, 239entry/s]
Pushing
C:\Users\kissa zahra\Desktop\semesters\Semester 7\MLOps\MLOPs Final Project\course-project-kissasium\venv\Lib\site-packages\oauth2client_helpers.py:255: UserWarning: Cannot access C:\Users\kissa zahra\AppData\Local\pydrive2fs\Cache\710796635688-iivsgbgsb6uv1fap6635dhvuei09o66c.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-iivsgbgsb6uv1fap6635dhvuei09o66c.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

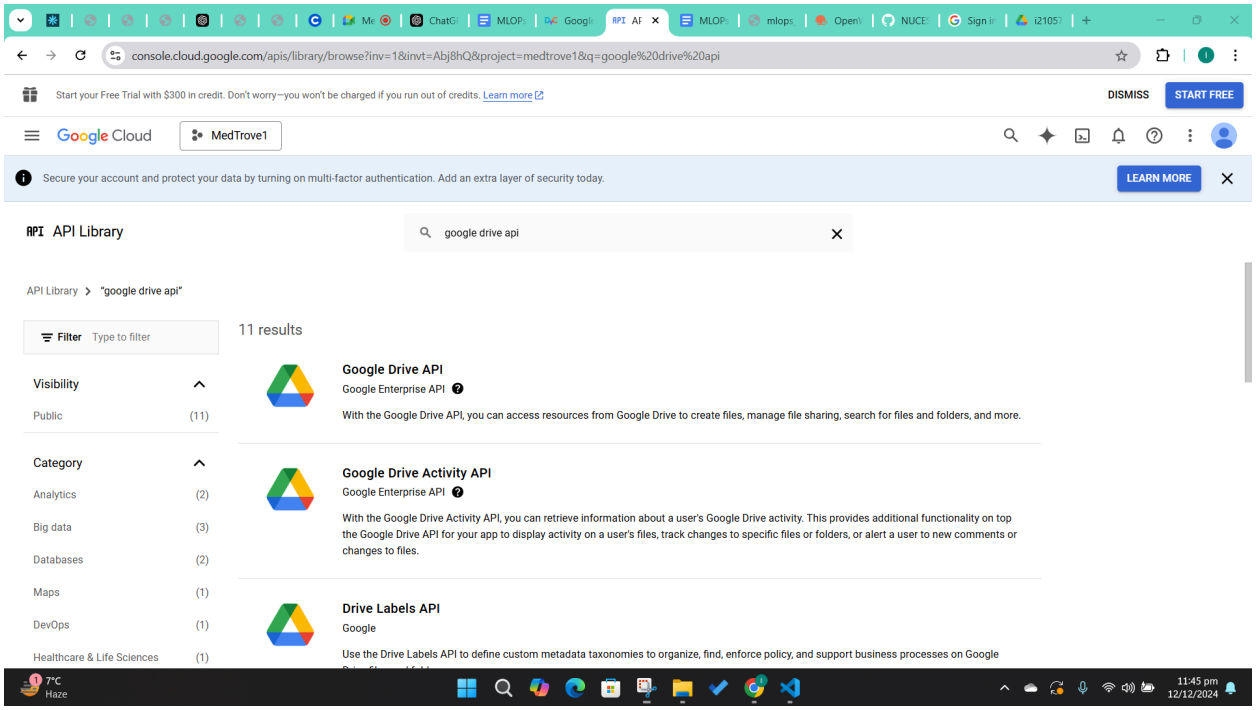


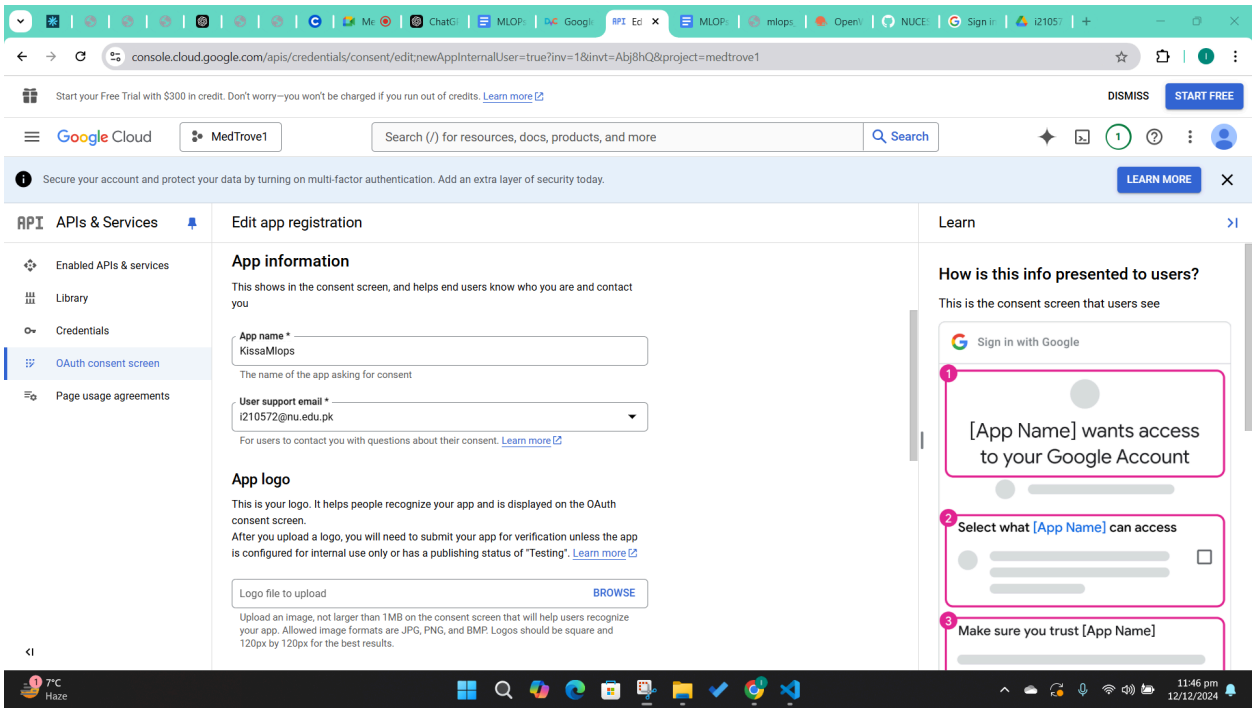
DVC IS BLOCKED BY GOOGLE!!

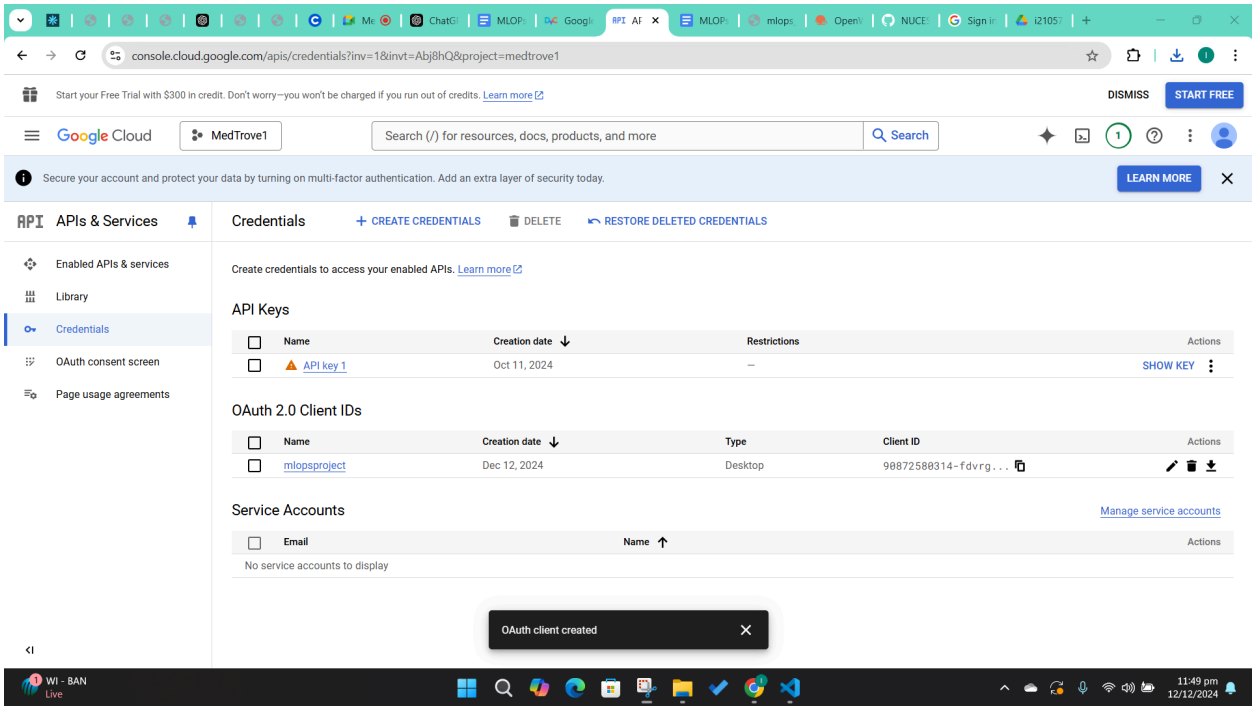
So now i am gonna use a custom Google Cloud project, link: https://dvc.org/doc/user-guide/data-management/remote-storage/google-drive#using-a-custom-google-cloud-project-recommended



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.

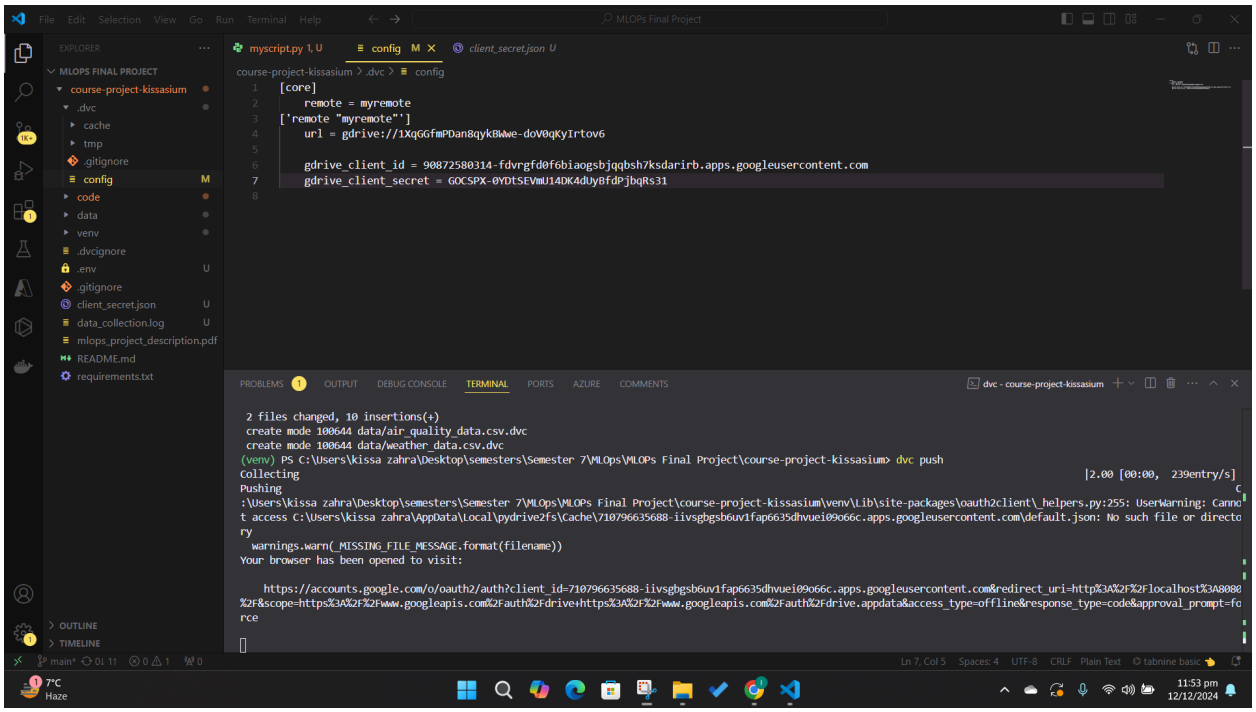

console.cloud.google.com/apis/library/browse?inv=1&invt=Abj8hQ&project=medtrove1&q=google%20drive%20api
Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more
DISMISS
START FREE
Google Cloud
MedTrove1
Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.
LEARN MORE
API Library
google drive api
API Library > "google drive api"
Filter Type to filter
Visibility
Public (11)
Category
Analytics (2)
Big data (3)
Databases (2)
Maps (1)
DevOps (1)
Healthcare & Life Sciences (1)
11 results
Google Drive API
Google Enterprise API
With the Google Drive API, you can access resources from Google Drive to create files, manage file sharing, search for files and folders, and more.
Google Drive Activity API
Google Enterprise API
With the Google Drive Activity API, you can retrieve information about a user's Google Drive activity. This provides additional functionality on top the Google Drive API for your app to display activity on a user's files, track changes to specific files or folders, or alert a user to new comments or changes to files.
Drive Labels API
Google
Use the Drive Labels API to define custom metadata taxonomies to organize, find, enforce policy, and support business processes on Google
7°C Haze
11:45 pm 12/12/2024


console.cloud.google.com/apis/credentials/consent/edit;newAppInternalUser=true?inv=1&invt=Abj8hQ&project=medtrove1
Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more
DISMISS
START FREE
Google Cloud
MedTrove1
Search (/) for resources, docs, products, and more
Search
Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.
LEARN MORE
APIs & Services
Enabled APIs & services
Library
Credentials
OAuth consent screen
Page usage agreements
Edit app registration
App information
This shows in the consent screen, and helps end users know who you are and contact you
App name *
KissaMlops
The name of the app asking for consent
User support email *
i210572@nu.edu.pk
For users to contact you with questions about their consent. Learn more
App logo
This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.
After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". Learn more
Logo file to upload
BROWSE
Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.
Learn
How is this info presented to users?
This is the consent screen that users see
Sign in with Google
[App Name] wants access to your Google Account
Select what [App Name] can access
Make sure you trust [App Name]
7°C Haze
11:46 pm 12/12/2024



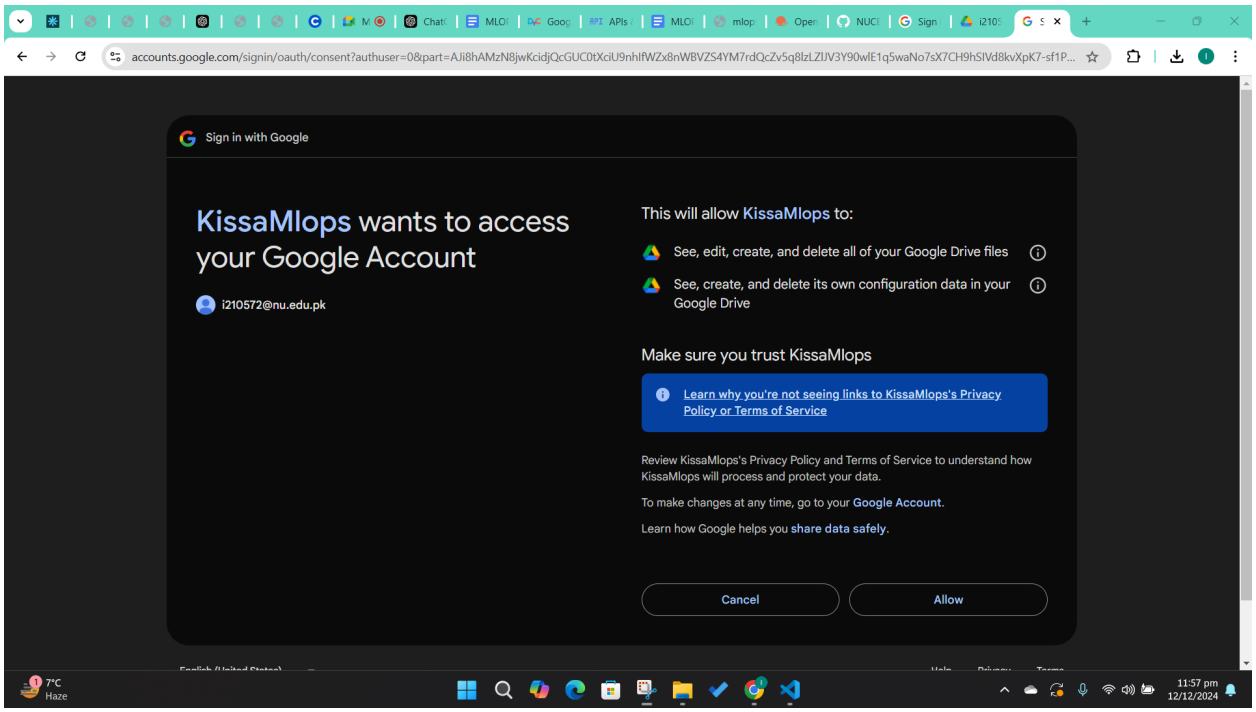
Download the json file.

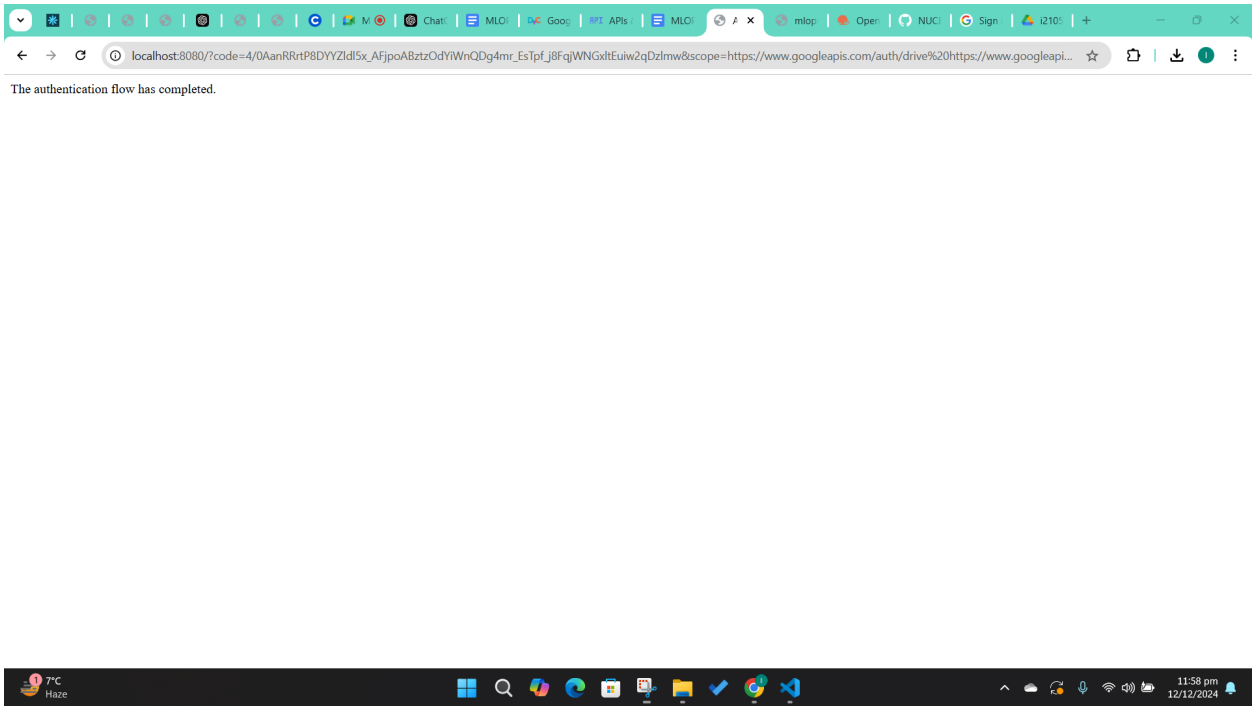
Use this file client id and client secrets to access the g-drive



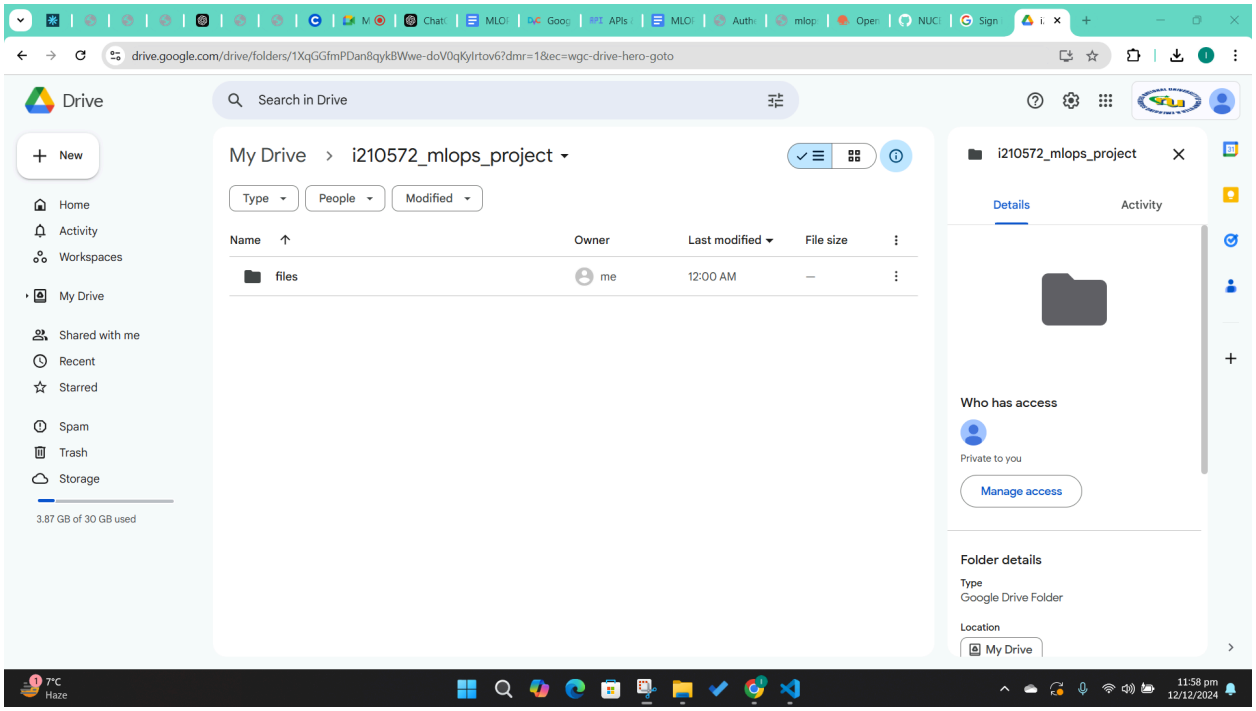
Run these two commands:
`dvc remote modify myremote gdrive_client_id 90872580314-fdvrgfd0f6biaogsbjqqbsh7ksdarirb.apps.googleusercontent.com`
`dvc remote modify myremote gdrive_client_secret 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`
`dvc push`

You can access the google drive now!!!

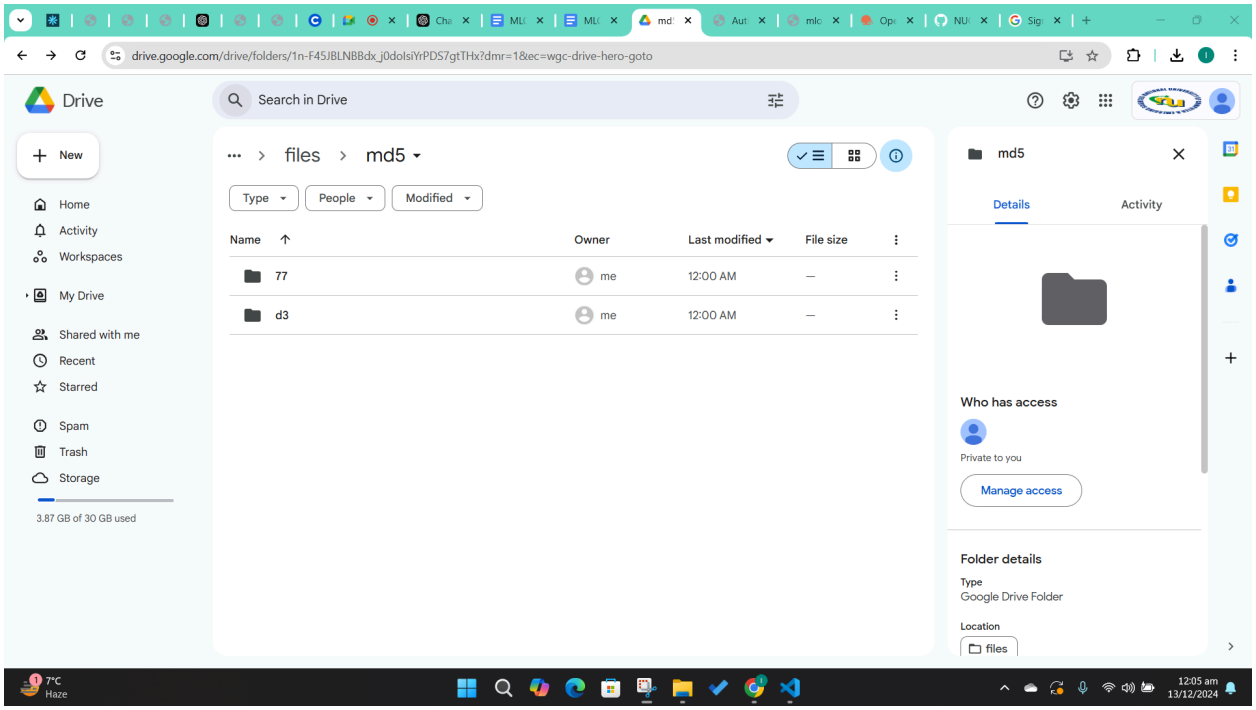


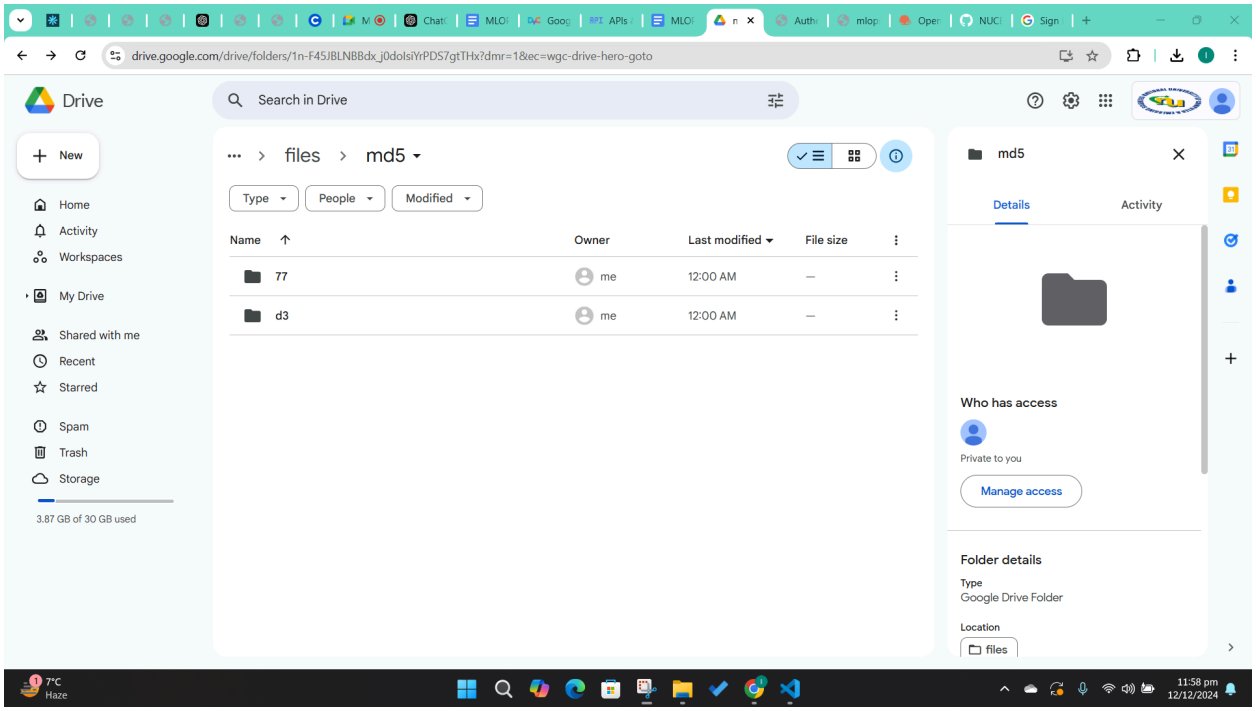


One folder has been created!!



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-F45JBLNBBdx_j0dolsiYrPDS7gtTHx?dmr=1&ec=wgc-drive-hero-goto
Drive
Search in Drive
New
Home
Activity
Workspaces
My Drive
Shared with me
Recent
Starred
Spam
Trash
Storage
3.87 GB of 30 GB used
files
md5
Type
People
Modified
Name
Owner
Last modified
File size
77
me
12:00 AM
d3
me
12:00 AM
md5
Details
Activity
Who has access
Private to you
Manage access
Folder details
Type
Google Drive Folder
Location
files
7°C
Haze
12:05 am
13/12/2024


drive.google.com/drive/folders/1n-F45JBLNBBdx_j0dolsiYrPDS7gtTHx?dmr=1&ec=wgc-drive-hero-goto
Drive
Search in Drive
New
Home
Activity
Workspaces
My Drive
Shared with me
Recent
Starred
Spam
Trash
Storage
3.87 GB of 30 GB used
files
md5
Type
People
Modified
Name
Owner
Last modified
File size
77
me
12:00 AM
d3
me
12:00 AM
md5
Details
Activity
Who has access
Private to you
Manage access
Folder details
Type
Google Drive Folder
Location
files
7°C
Haze
11:58 pm
12/12/2024

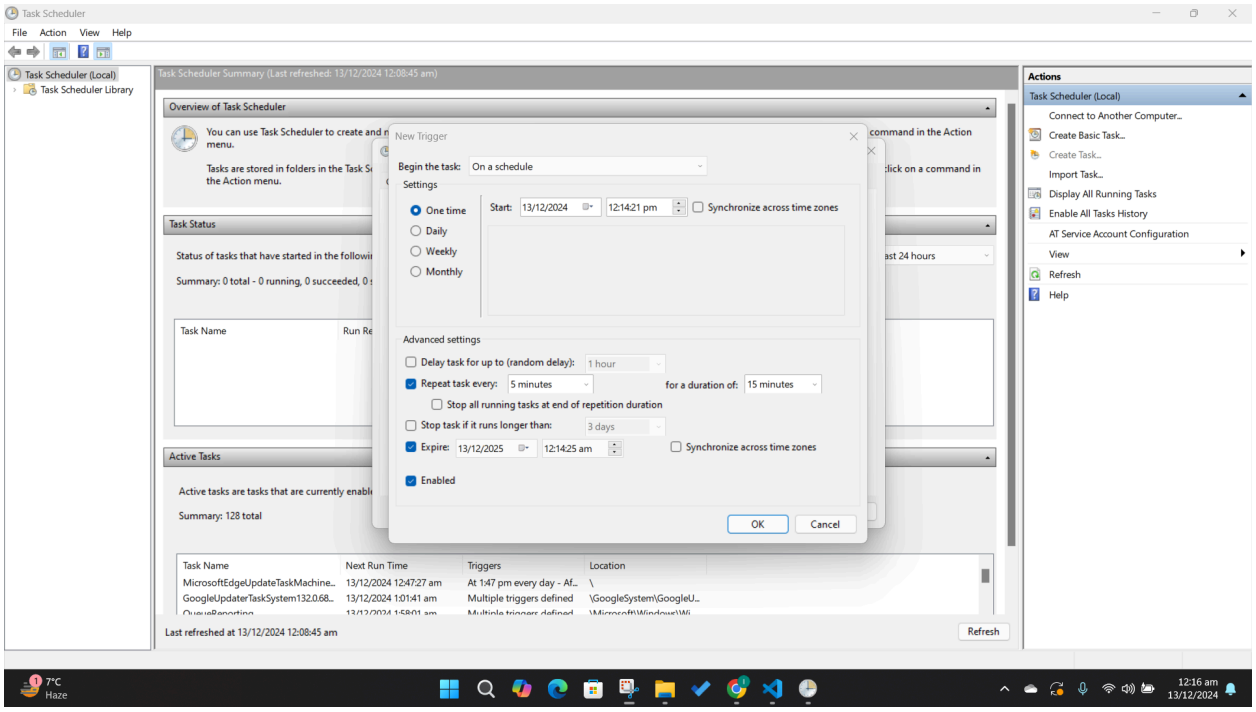
6. Automate Data Collection

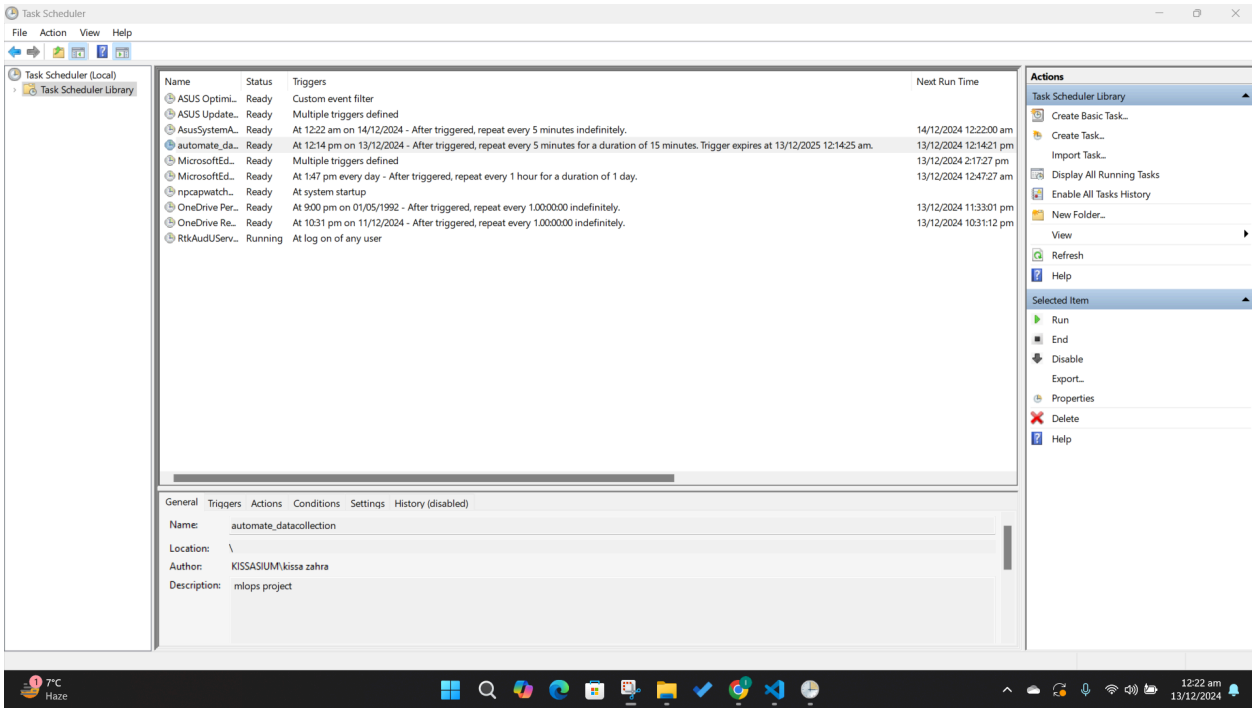
Schedule regular data fetching using cron jobs or task schedulers. I used ‘task scheduler’ since I’m using Windows.

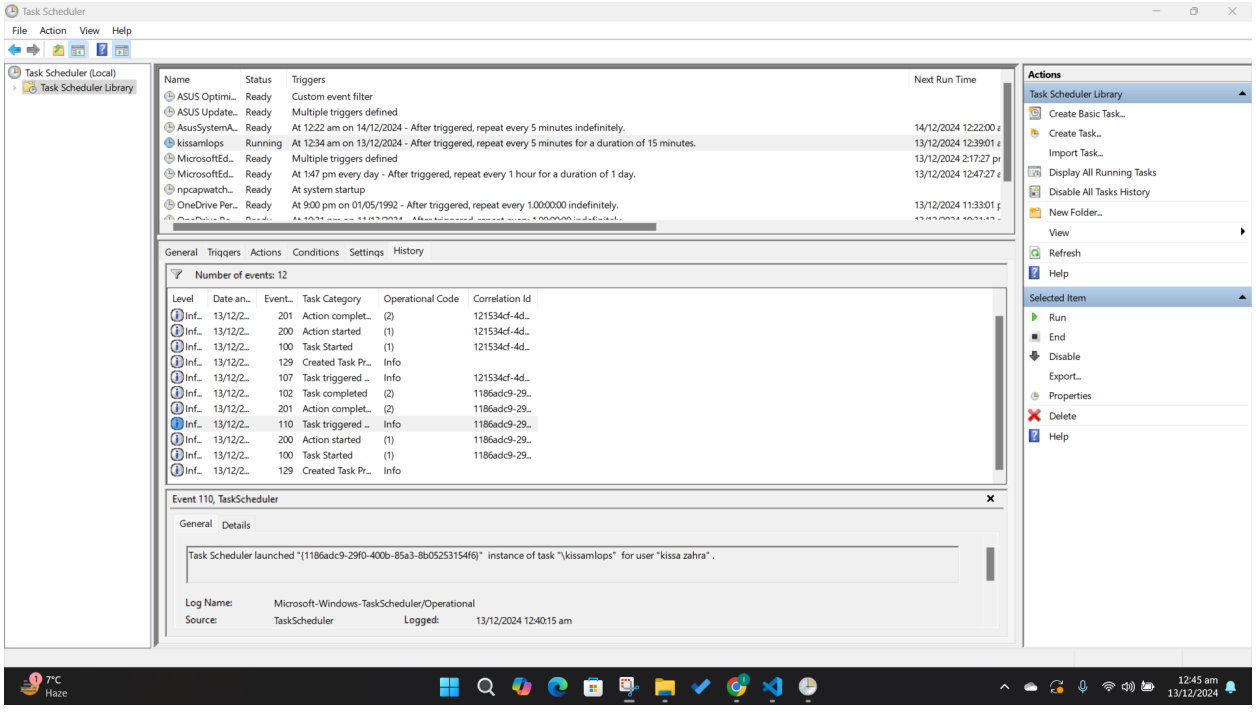
Open Task Scheduler and click on Create Task.

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

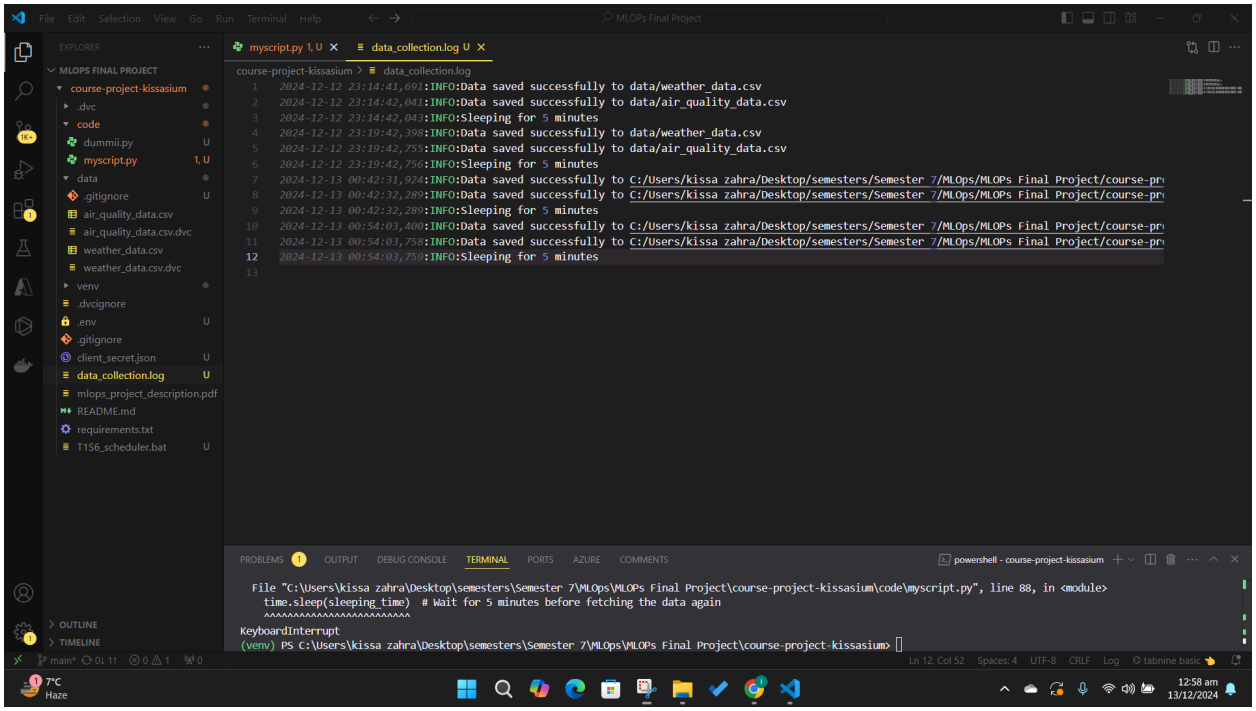
Go to Actions and select the virtual environment Python executable as the Program/Script. Add the absolute path enclosed in double quotes, to the .py code file as the argument in the task. Ensure that you are using an absolute path when saving your files within the code too.



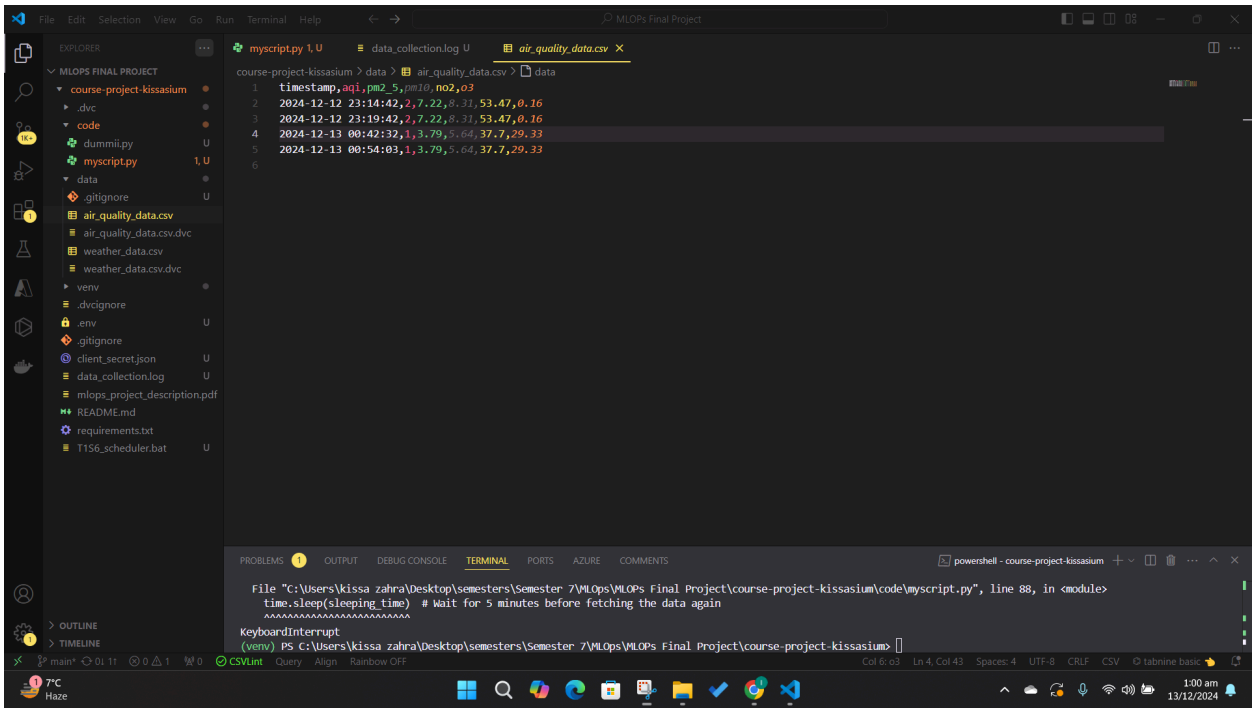

Task Scheduler
File Action View Help
Task Scheduler (Local)
Task Scheduler Library
Name Status Triggers Next Run Time
ASUS Optimi... Ready Custom event filter
ASUS Update... Ready Multiple triggers defined
AsusSystemA... Ready At 12:22 am on 14/12/2024 - After triggered, repeat every 5 minutes indefinitely. 14/12/2024 12:22:00 am
automate_da... Ready At 12:14 pm on 13/12/2024 - After triggered, repeat every 5 minutes for a duration of 15 minutes. Trigger expires at 13/12/2025 12:14:25 am. 13/12/2024 12:14:21 pm
MicrosoftEd... Ready Multiple triggers defined 13/12/2024 2:17:27 pm
MicrosoftEd... Ready At 1:47 pm every day - After triggered, repeat every 1 hour for a duration of 1 day. 13/12/2024 12:47:27 am
npcapwatch... Ready At system startup
OneDrive Per... Ready At 9:00 pm on 01/05/1992 - After triggered, repeat every 1.00:00:00 indefinitely. 13/12/2024 11:33:01 pm
OneDrive Re... Ready At 10:31 pm on 11/12/2024 - After triggered, repeat every 1.00:00:00 indefinitely. 13/12/2024 10:31:12 pm
RtkAudUServ... Running At log on of any user
General Triggers Actions Conditions Settings History (disabled)
Name: automate_datacollection
Location: \
Author: KISSASIUM\kissa zahra
Description: mlops project
Actions
Task Scheduler Library
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Enable All Tasks History
New Folder...
View
Refresh
Help
Selected Item
Run
End
Disable
Export...
Properties
Delete
Help
7°C Haze
12:22 am 13/12/2024


Task Scheduler
File Action View Help
Task Scheduler (Local)
Task Scheduler Library
Name Status Triggers Next Run Time
ASUS Optimi... Ready Custom event filter
ASUS Update... Ready Multiple triggers defined
AsusSystemA... Ready At 12:22 am on 14/12/2024 - After triggered, repeat every 5 minutes indefinitely. 14/12/2024 12:22:00 a
kissamlops Running At 12:34 am on 13/12/2024 - After triggered, repeat every 5 minutes for a duration of 15 minutes. 13/12/2024 12:39:01 a
MicrosoftEd... Ready Multiple triggers defined 13/12/2024 2:17:27 pr
MicrosoftEd... Ready At 1:47 pm every day - After triggered, repeat every 1 hour for a duration of 1 day. 13/12/2024 12:47:27 a
npcapwatch... Ready At system startup
OneDrive Per... Ready At 9:00 pm on 01/05/1992 - After triggered, repeat every 1.00:00:00 indefinitely. 13/12/2024 11:33:01 p
General Triggers Actions Conditions Settings History
Number of events: 12
Level Date an... Event... Task Category Operational Code Correlation Id
Inf... 13/12/2... 201 Action complet... (2) 121534cf-4d...
Inf... 13/12/2... 200 Action started (1) 121534cf-4d...
Inf... 13/12/2... 100 Task Started (1) 121534cf-4d...
Inf... 13/12/2... 129 Created Task Pr... Info
Inf... 13/12/2... 107 Task triggered ... Info 121534cf-4d...
Inf... 13/12/2... 102 Task completed (2) 1186adc9-29...
Inf... 13/12/2... 201 Action complet... (2) 1186adc9-29...
Inf... 13/12/2... 110 Task triggered ... Info 1186adc9-29...
Inf... 13/12/2... 200 Action started (1) 1186adc9-29...
Inf... 13/12/2... 100 Task Started (1) 1186adc9-29...
Inf... 13/12/2... 129 Created Task Pr... Info
Event 110, TaskScheduler
General Details
Task Scheduler launched "{1186adc9-29f0-400b-85a3-8b0525315 4f6}" instance of task "\kissamlops" for user "kissa zahra" .
Log Name: Microsoft-Windows-TaskScheduler/Operational
Source: TaskScheduler Logged: 13/12/2024 12:40:15 am
Actions
Task Scheduler Library
Create Basic Task...
Create Task...
Import Task...
Display All Running Tasks
Disable All Tasks History
New Folder...
View
Refresh
Help
Selected Item
Run
End
Disable
Export...
Properties
Delete
Help
7°C Haze
12:45 am 13/12/2024

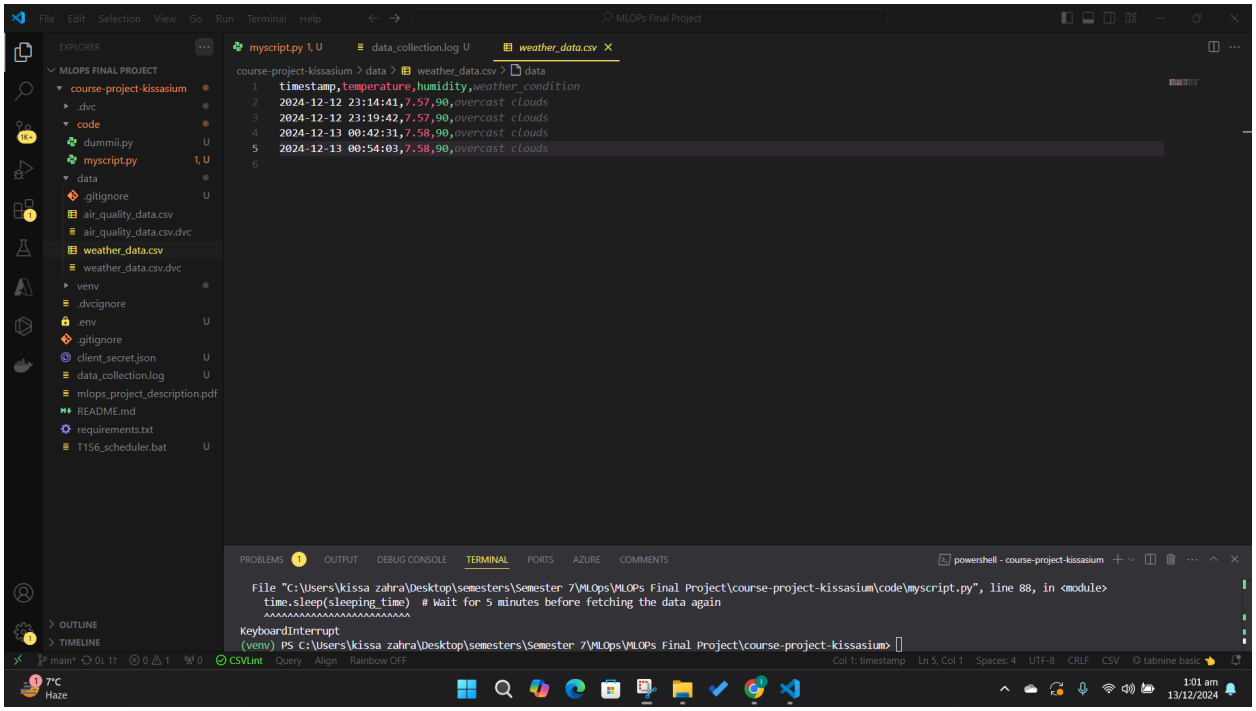
Updating the data_collection.log file with task scheduling method for fetching the data



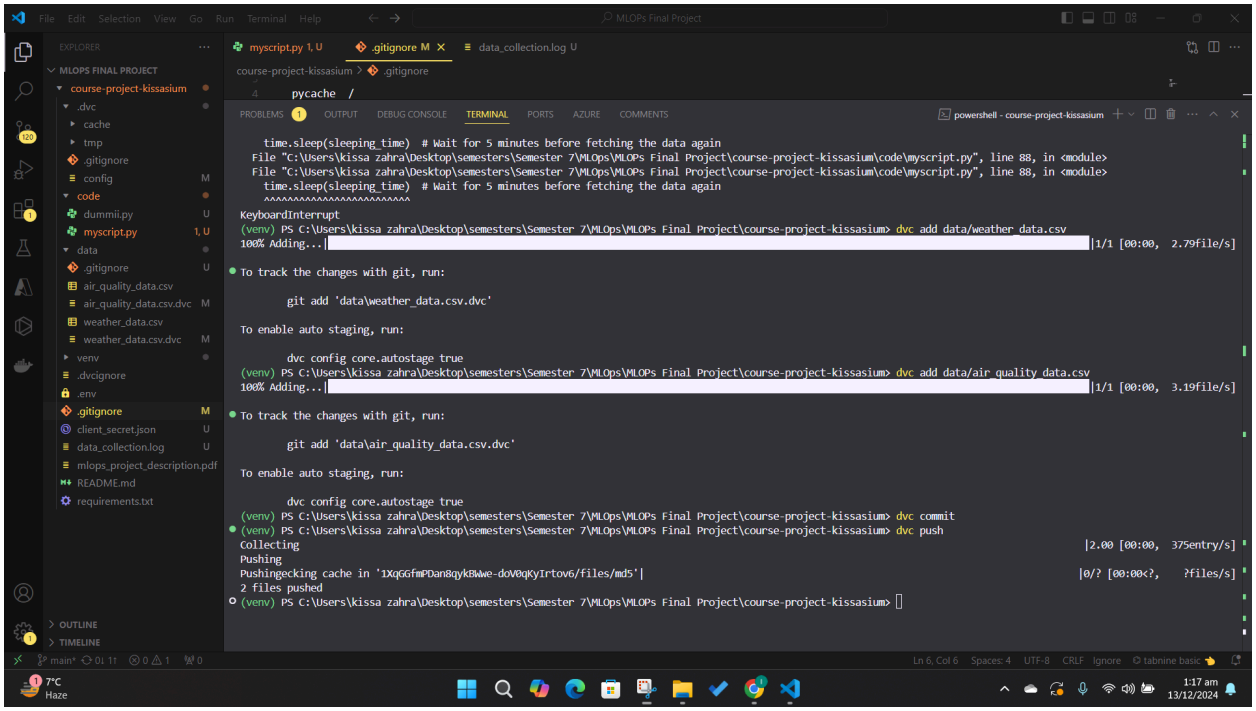
Updating the air_quality_data.csv file with task scheduling



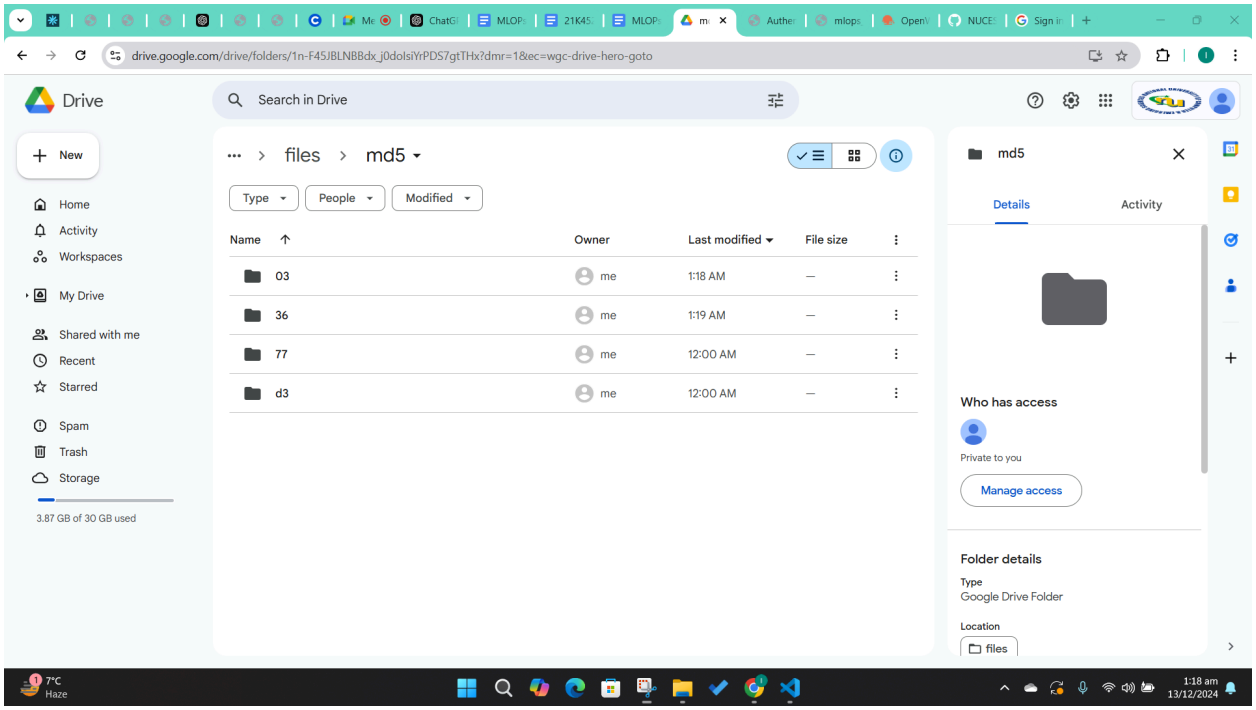
Updating the weather_data.csv file with task scheduling



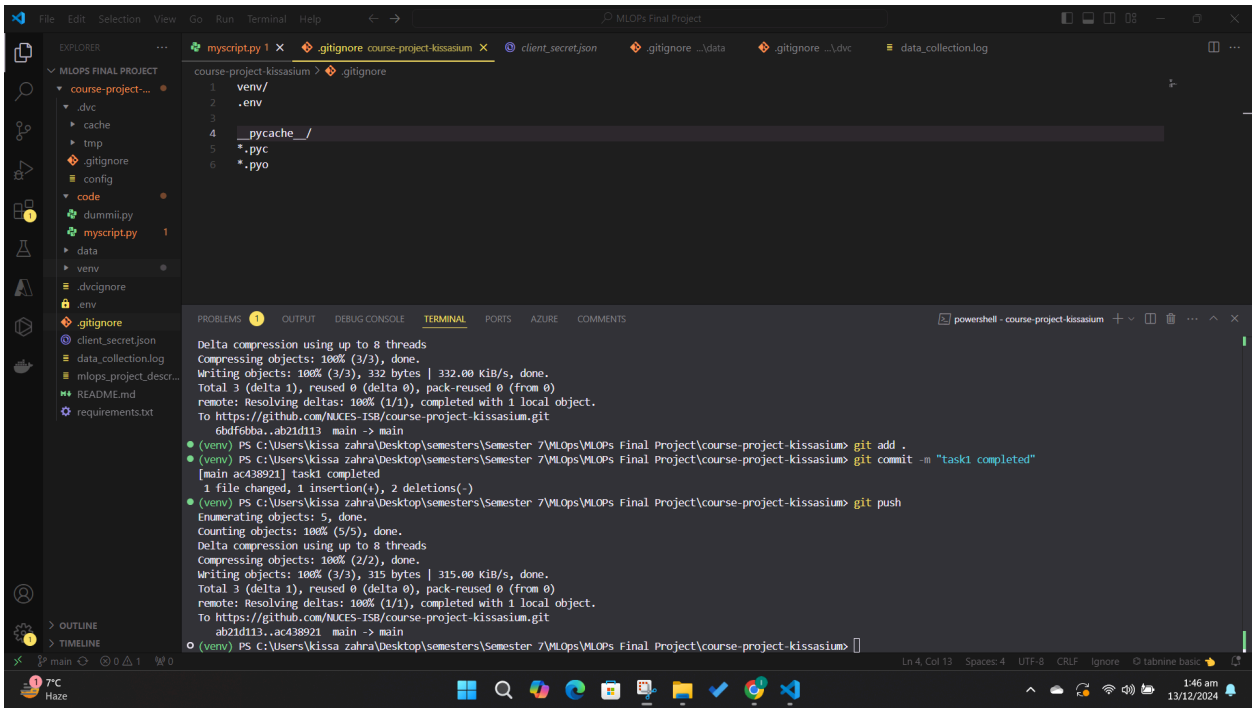
7. Update Data with DVC



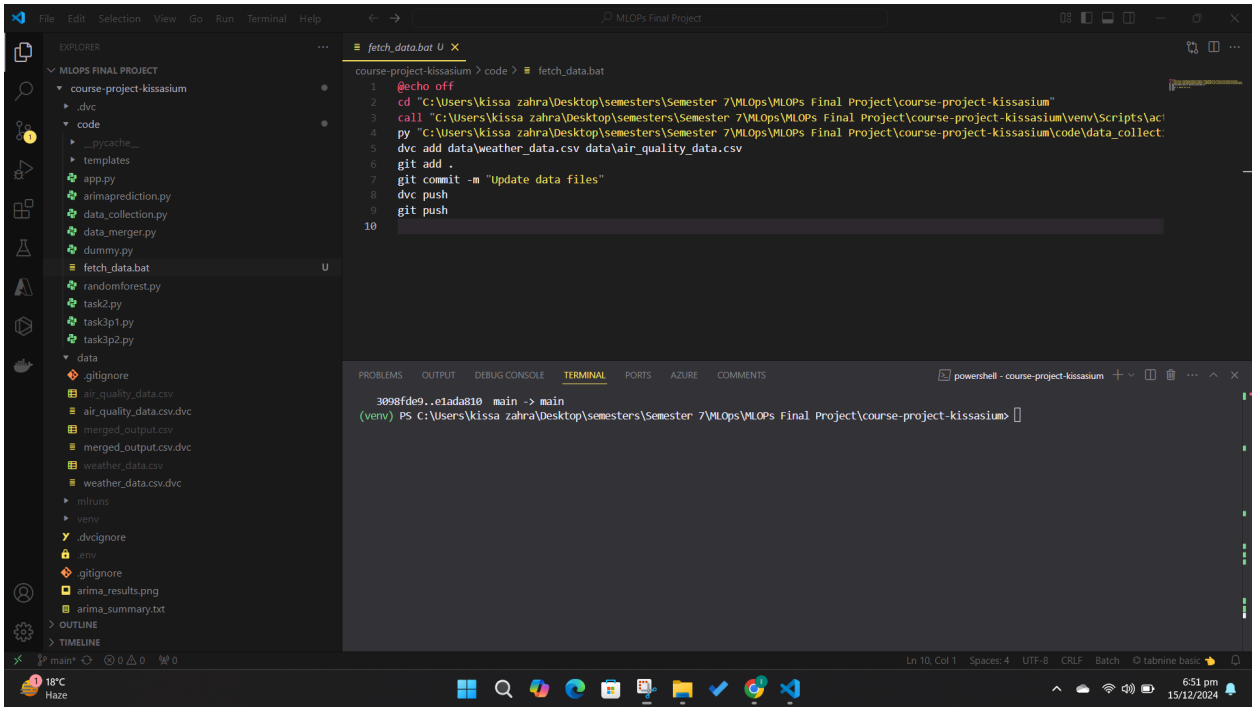
Updated on the drive



Now pushing it on github



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.



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.

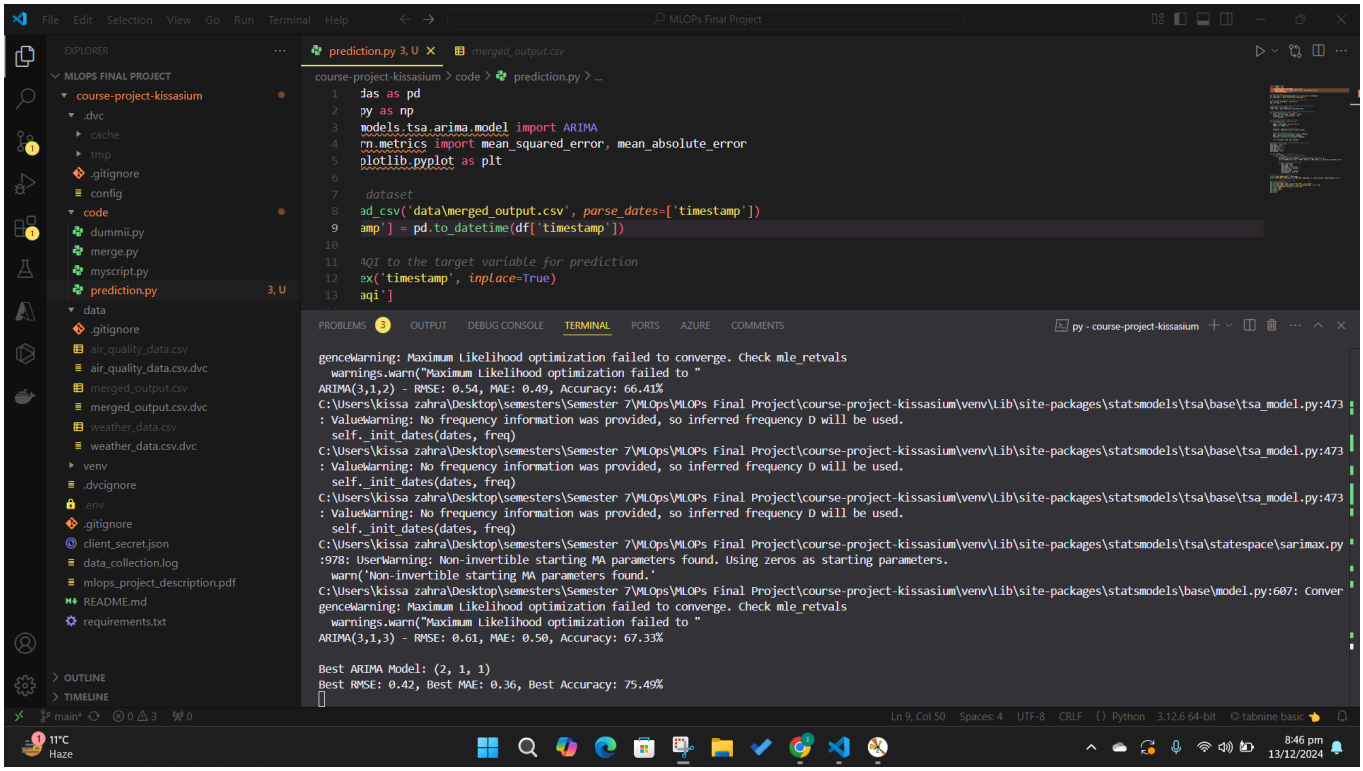
3. Evaluation

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.

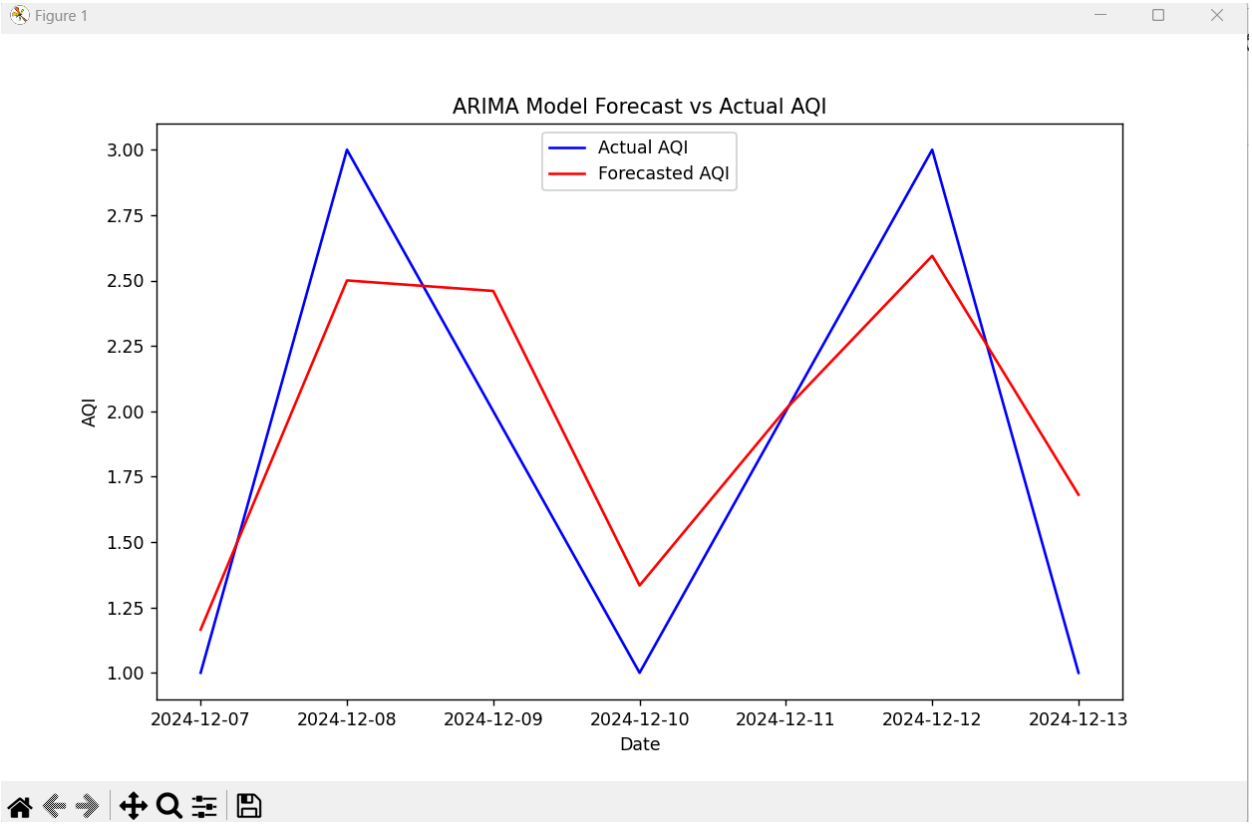
Output with ARIMA

After running the code



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:

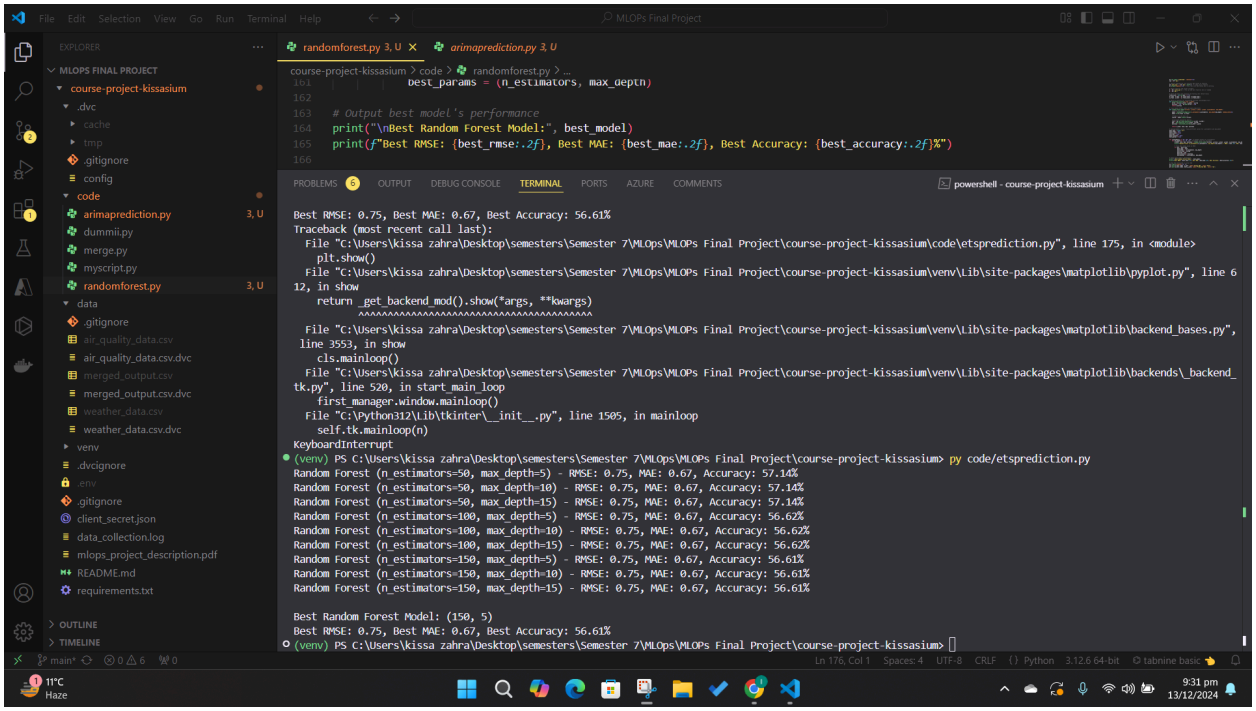
- **n_estimators**: Number of trees in the forest.
- **max_depth**: Limits the depth of each tree to prevent overfitting.
- **min_samples_split**: Minimum samples required to split a node.

3. Evaluation

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

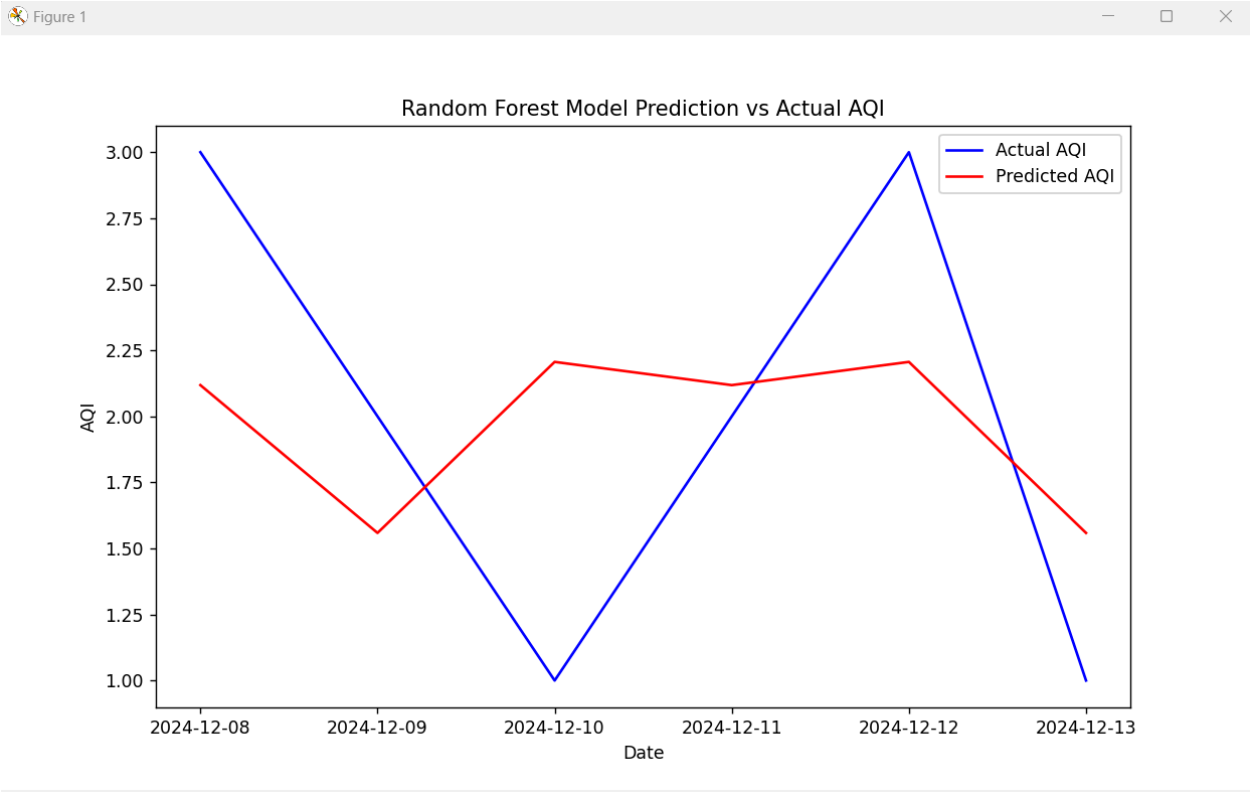
Lower RMSE and MAE values and higher accuracy indicate better performance.

Output with Random Forest Model



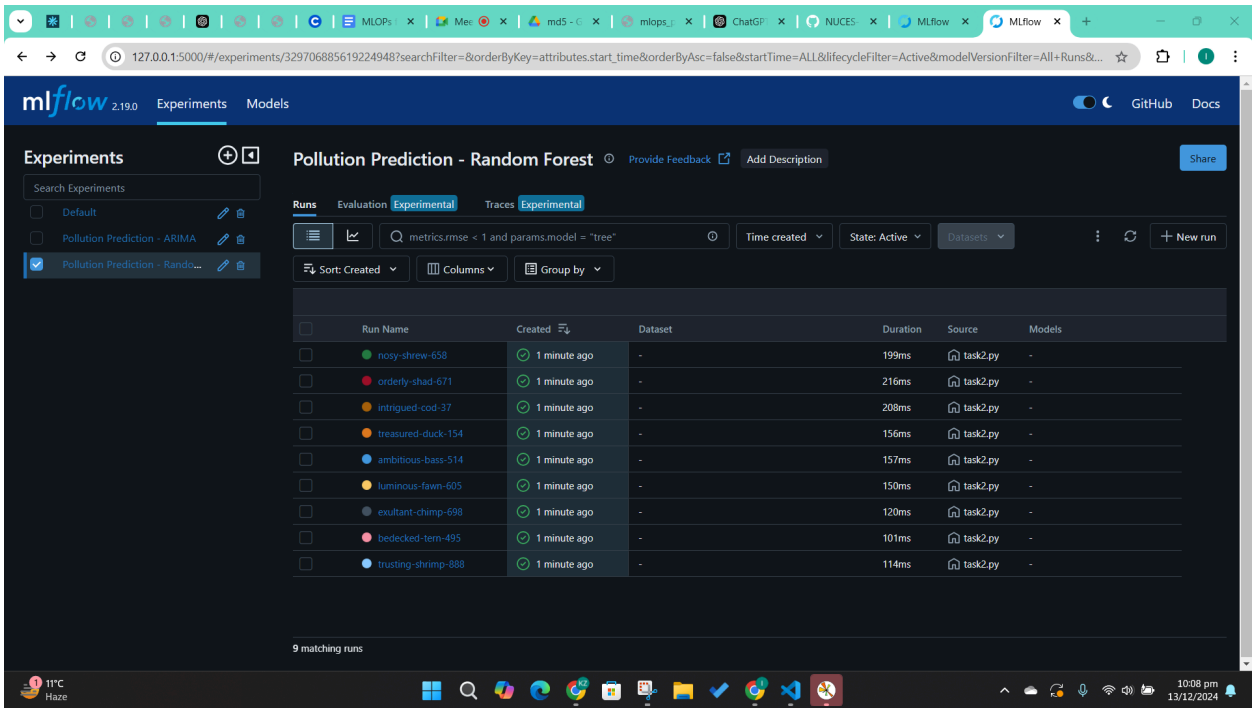
Best Random Forest Model: (150, 5)
Best RMSE: 0.75, Best MAE: 0.67, Best Accuracy: 56.61%

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

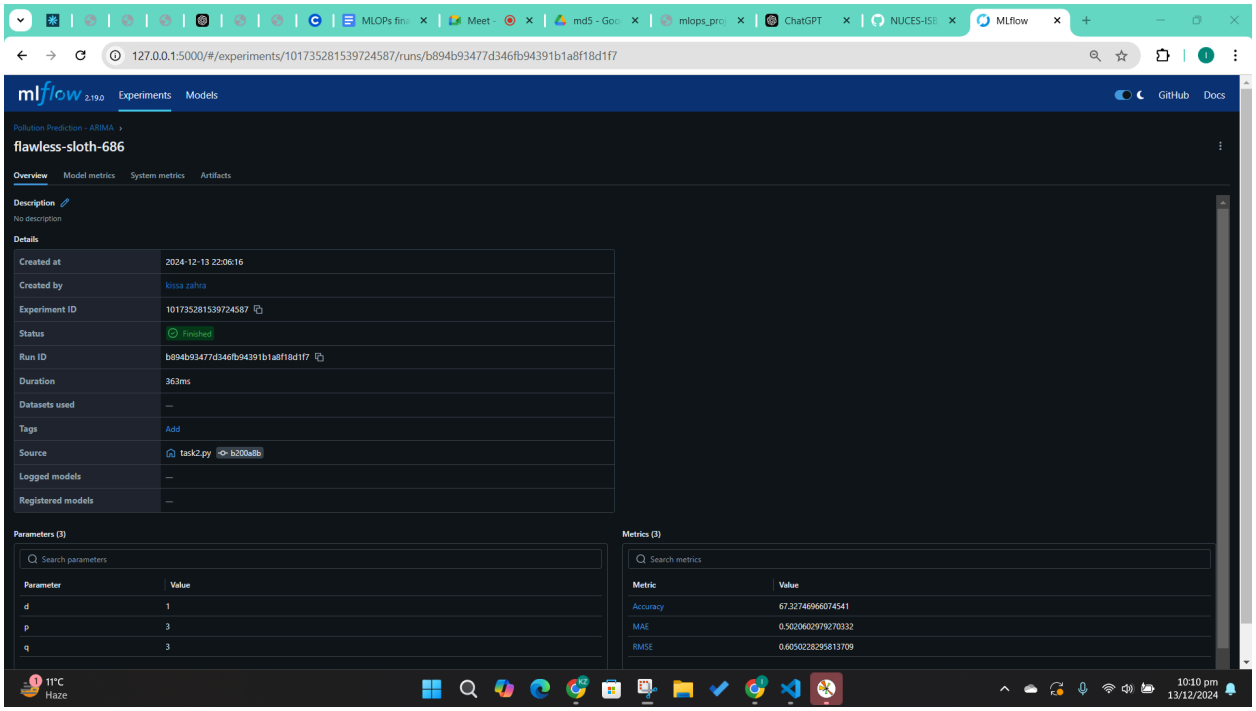


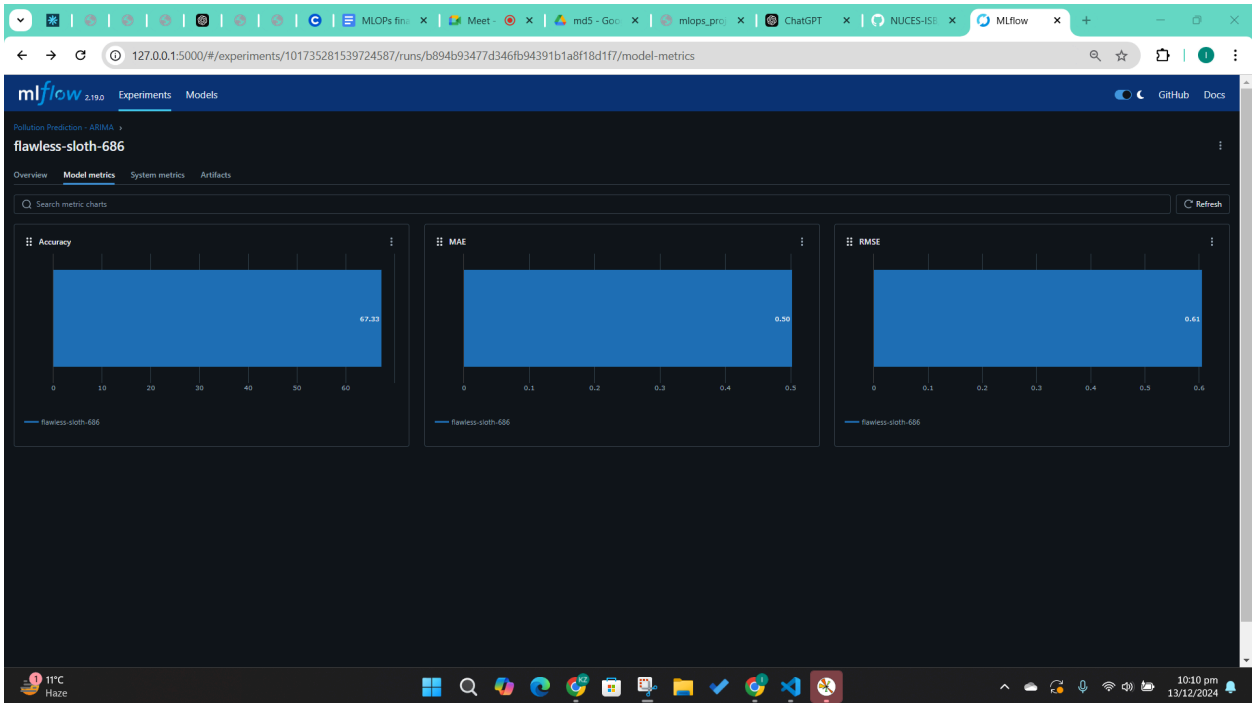
3. Train Models with MLflow

`pip install mlflow`

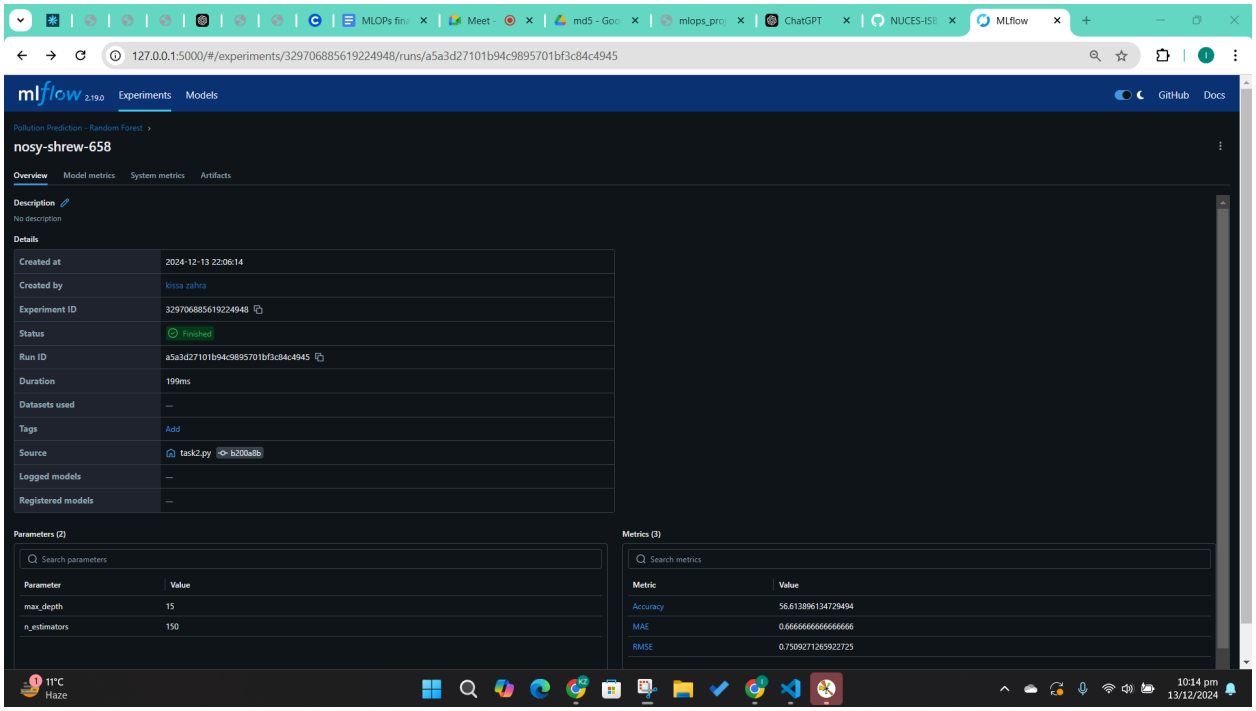


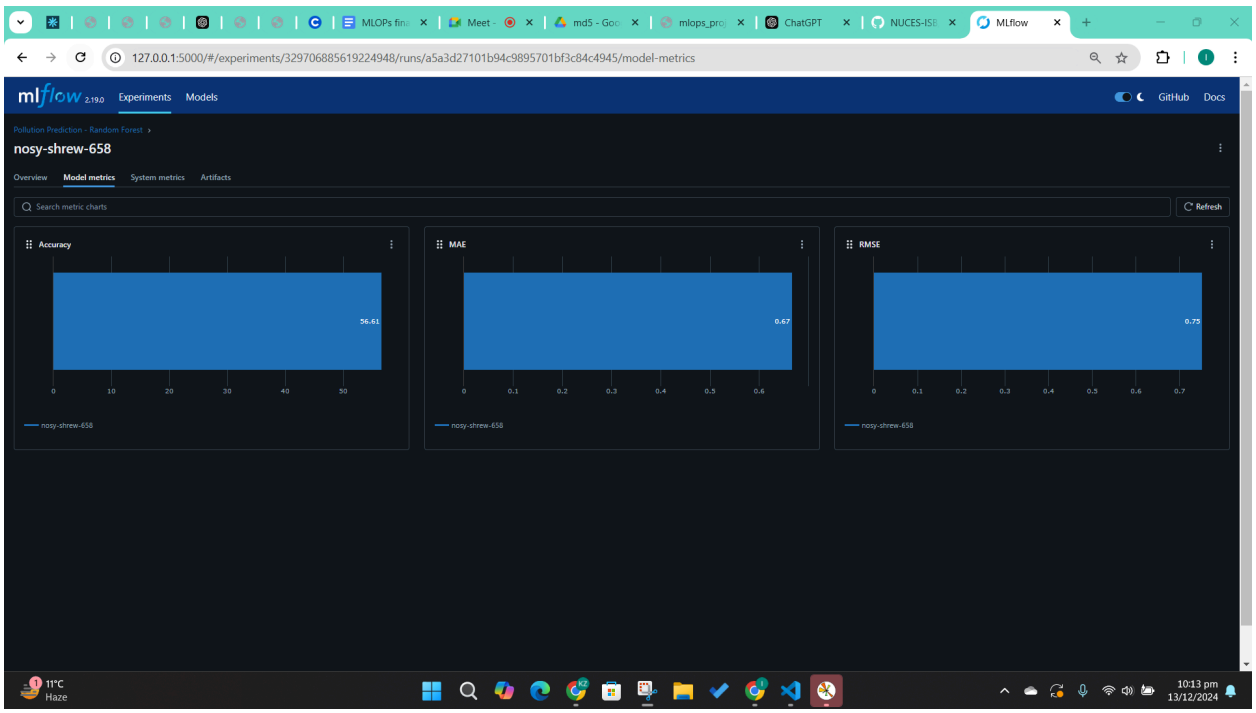
ARIMA


MLflow
127.0.0.1:5000/#/experiments/101735281539724587/runs/b894b93477d346fb94391b1a8f18d1f7
mlflow 2.19.0
Experiments
Models
GitHub
Docs
Pollution Prediction - ARIMA
flawless-sloth-686
Overview
Model metrics
System metrics
Artifacts
Description
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 task2.py b200a8b
Logged models –
Registered models –
Parameters (3)
Search parameters
Parameter Value
d 1
p 3
q 3
Metrics (3)
Search metrics
Metric Value
Accuracy 67.32746966074541
MAE 0.5020602979270332
RMSE 0.6050228295813709
11°C Haze
10:10 pm 13/12/2024


MLflow
127.0.0.1:5000/#/experiments/101735281539724587/runs/b894b93477d346fb94391b1a8f18d1f7/model-metrics
mlflow 2.19.0
Experiments
Models
GitHub
Docs
Pollution Prediction - ARIMA
flawless-sloth-686
Overview
Model metrics
System metrics
Artifacts
Search metric charts
Refresh
Accuracy
67.33
MAE
0.50
RMSE
0.61
flawless-sloth-686
11°C Haze
10:10 pm 13/12/2024

Random Forest Model





4. Hyperparameter Tuning

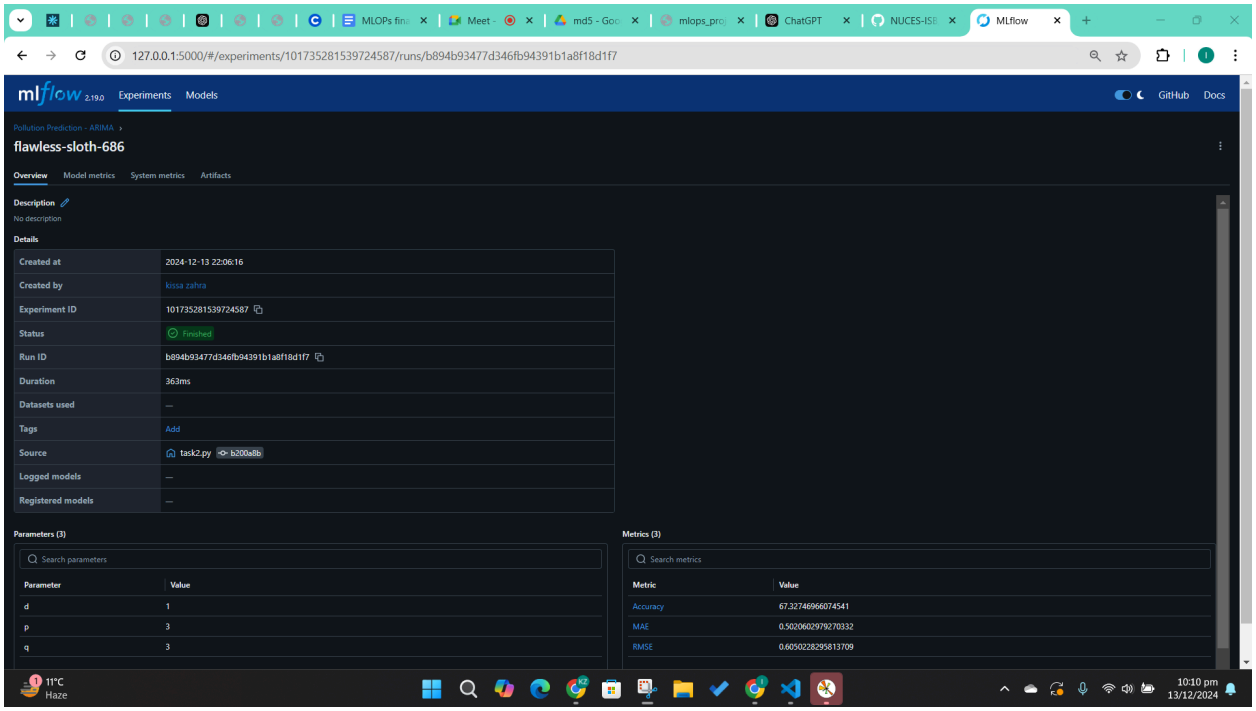
I have used a grid search approach. It involves exhaustively trying all possible combinations of the specified hyperparameters over a defined grid.

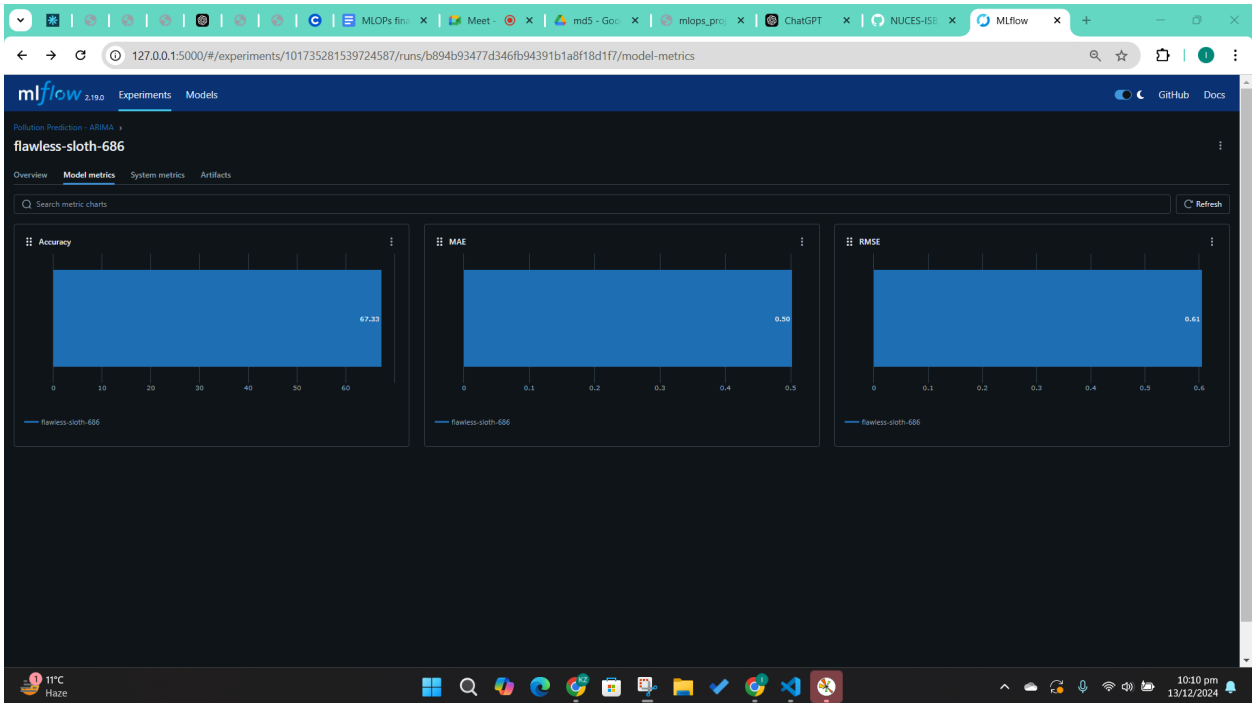
ARIMA

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
```


MLflow
127.0.0.1:5000/#/experiments/101735281539724587/runs/b894b93477d346fb94391b1a8f18d1f7
mlflow 2.19.0
Experiments
Models
GitHub
Docs
Pollution Prediction - ARIMA
flawless-sloth-686
Overview
Model metrics
System metrics
Artifacts
Description
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
task2.py
b200a8b
Logged models
Registered models
Parameters (3)
Search parameters
Parameter
Value
d
1
p
3
q
3
Metrics (3)
Search metrics
Metric
Value
Accuracy
67.32746966074541
MAE
0.5020602979270332
RMSE
0.6050228295813709
11°C
Haze
10:10 pm
13/12/2024

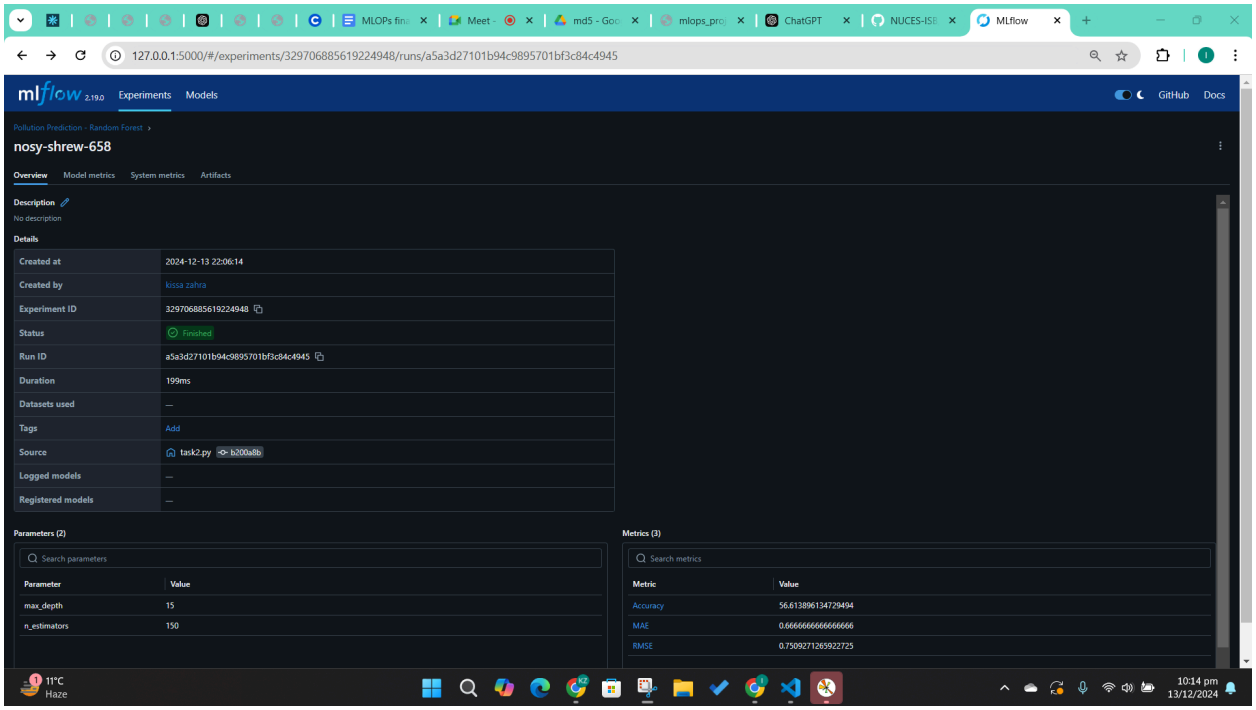

MLflow
127.0.0.1:5000/#/experiments/101735281539724587/runs/b894b93477d346fb94391b1a8f18d1f7/model-metrics
mlflow 2.19.0
Experiments
Models
GitHub
Docs
Pollution Prediction - ARIMA
flawless-sloth-686
Overview
Model metrics
System metrics
Artifacts
Search metric charts
Refresh
Accuracy
67.33
MAE
0.50
RMSE
0.61
flawless-sloth-686
11°C
Haze
10:10 pm
13/12/2024

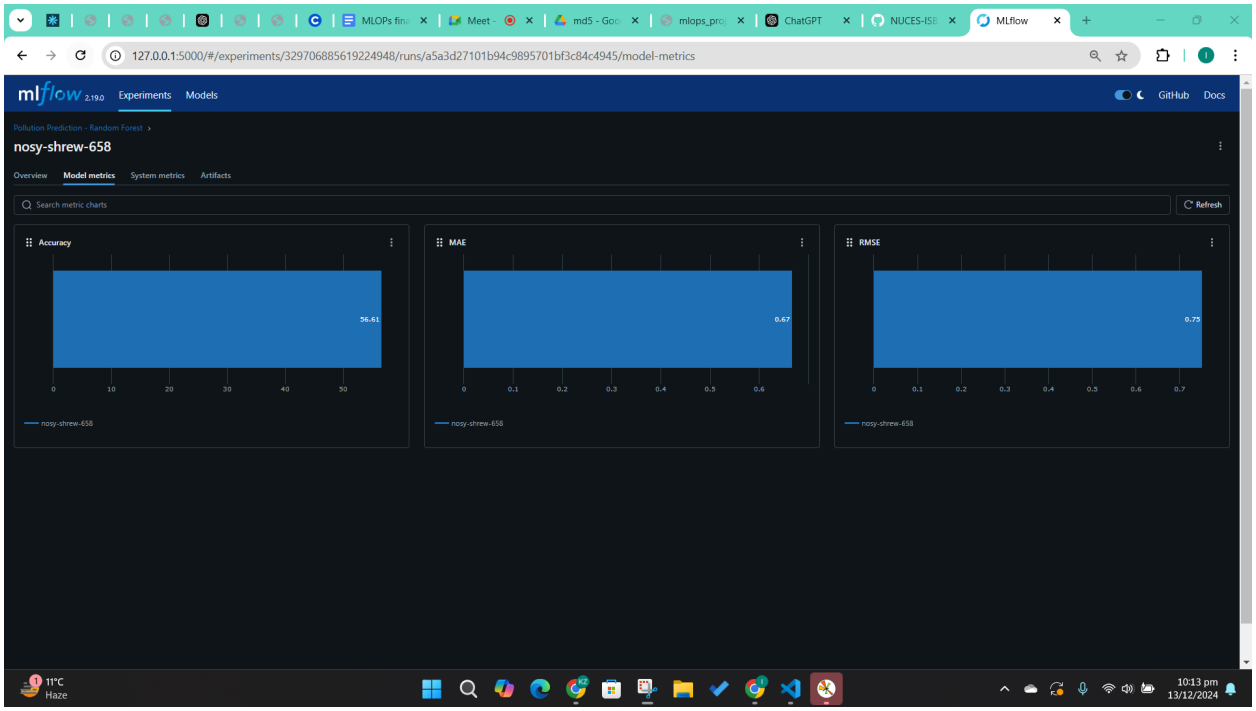
Random Forest Model

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
```





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

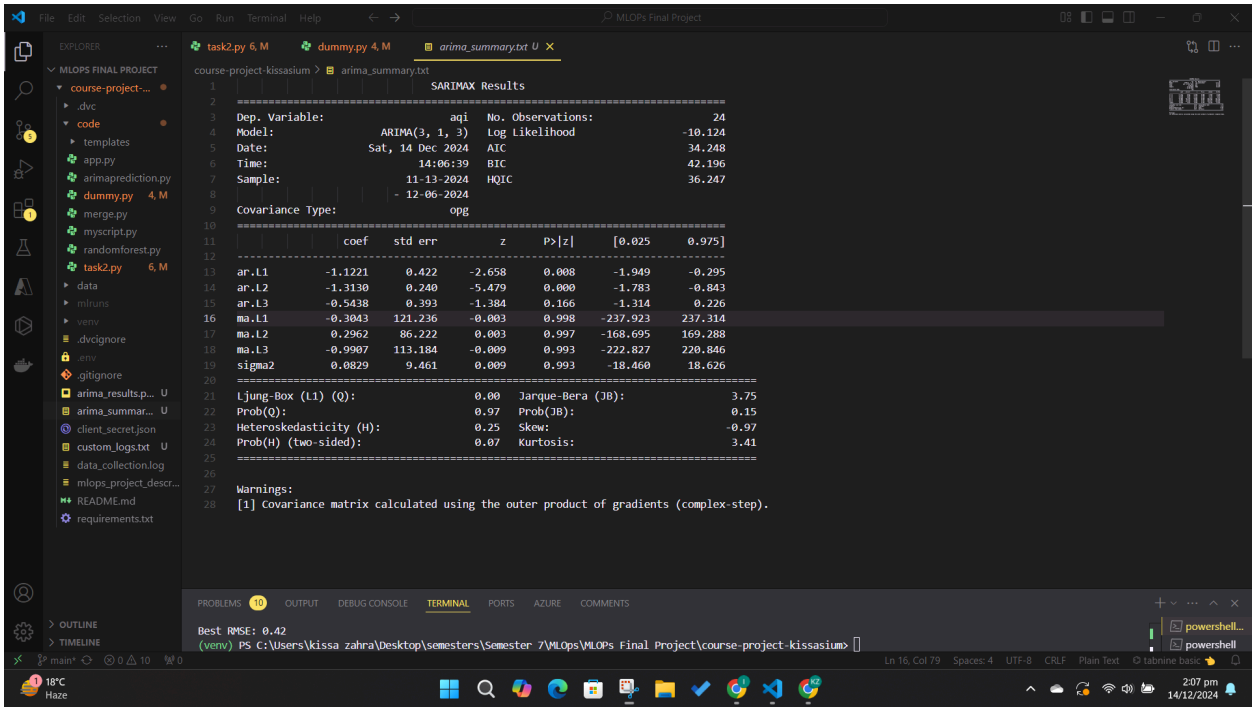
5. Model Evaluation

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%.

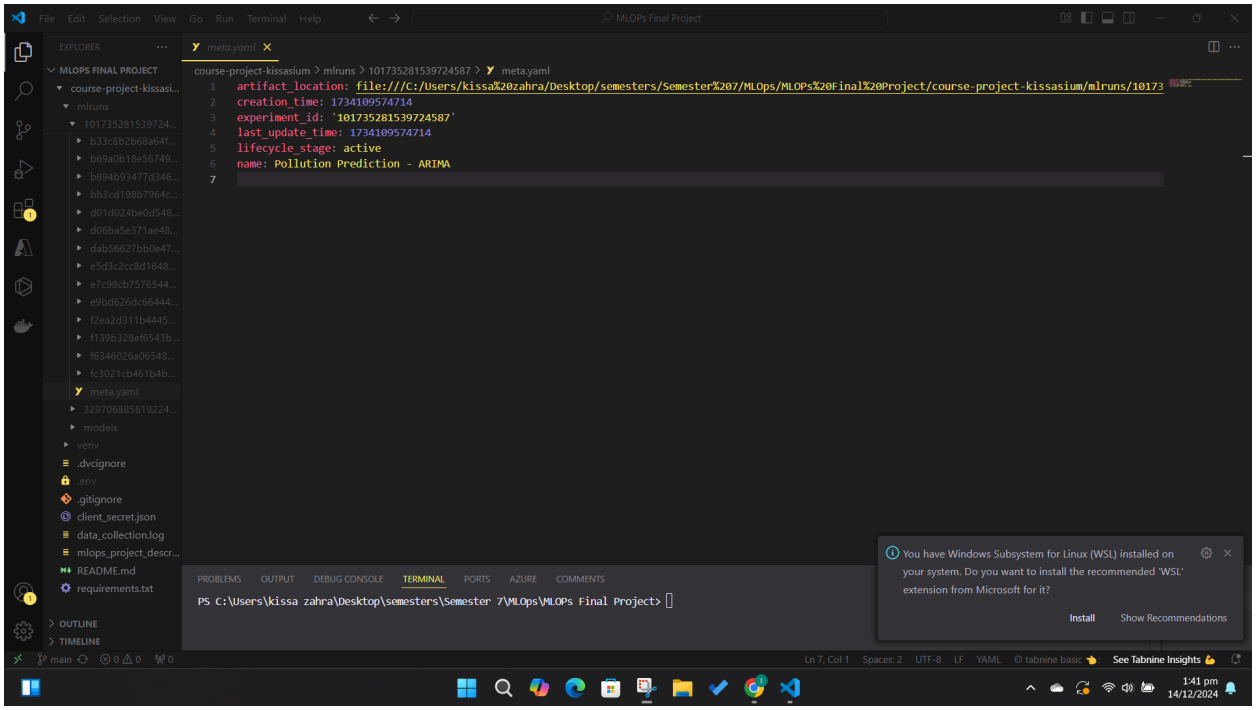
Model matrix



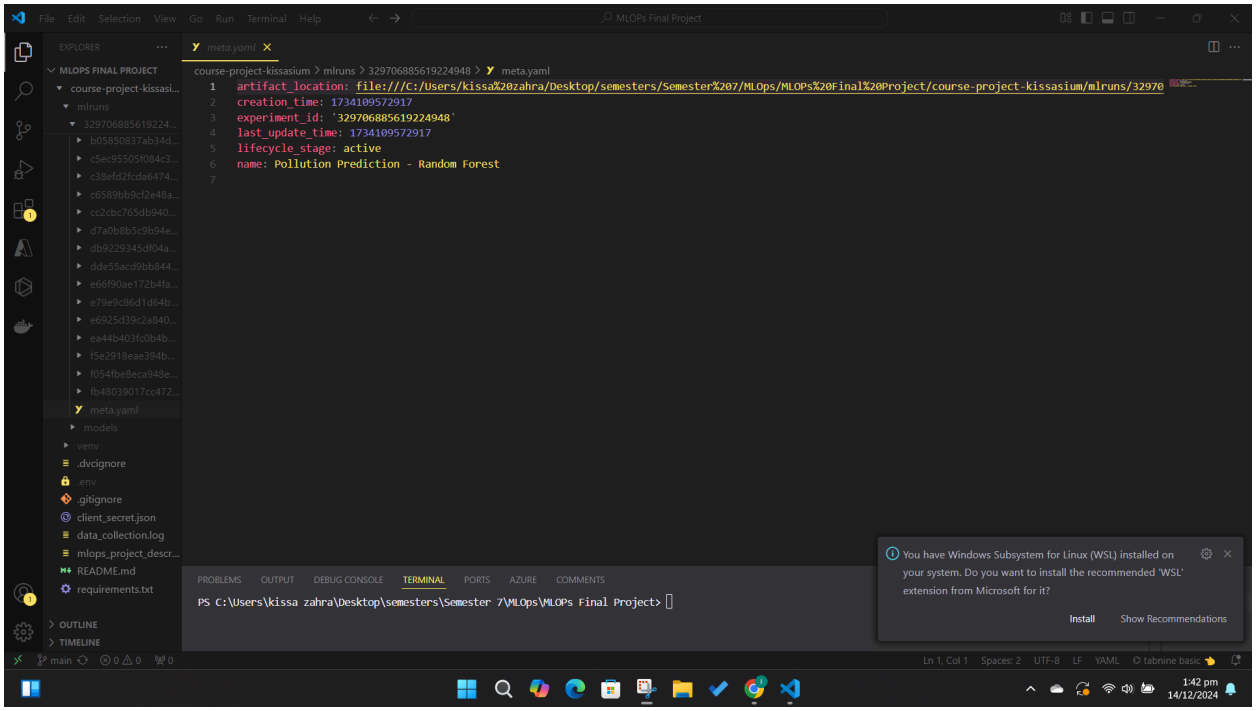
Information about my model

In meta.yaml file

Arima

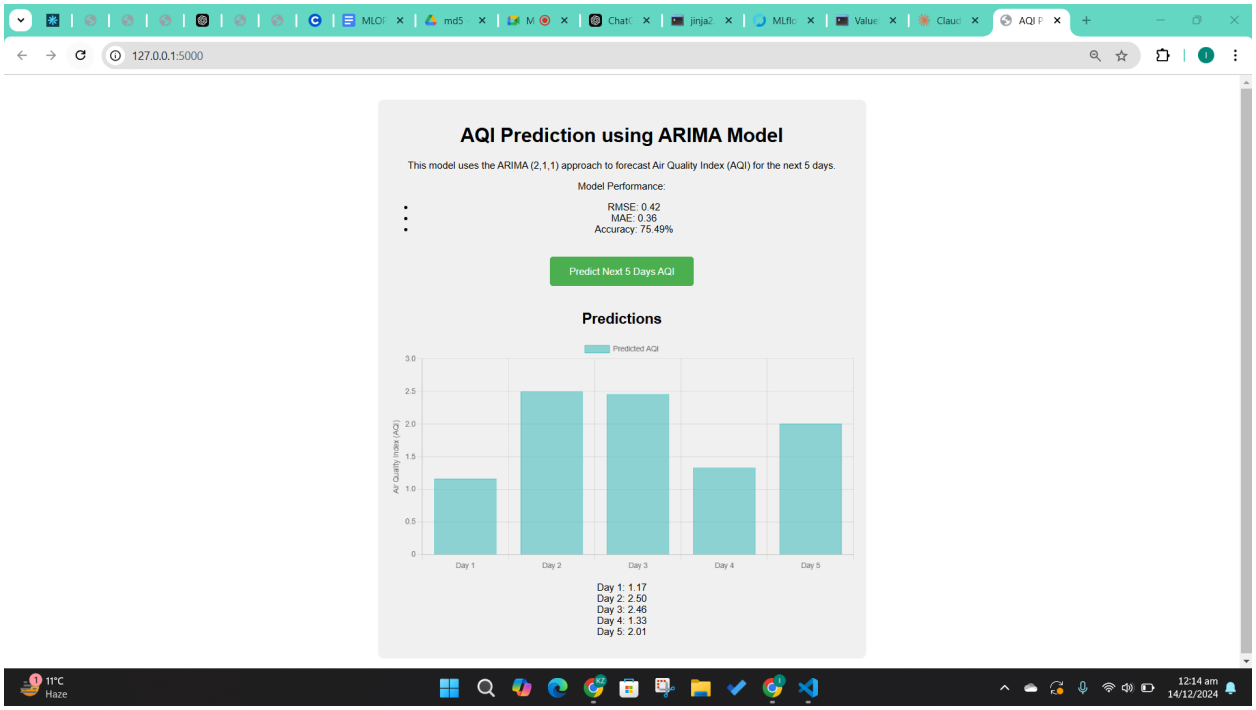


Random Forest



6. Deployment

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

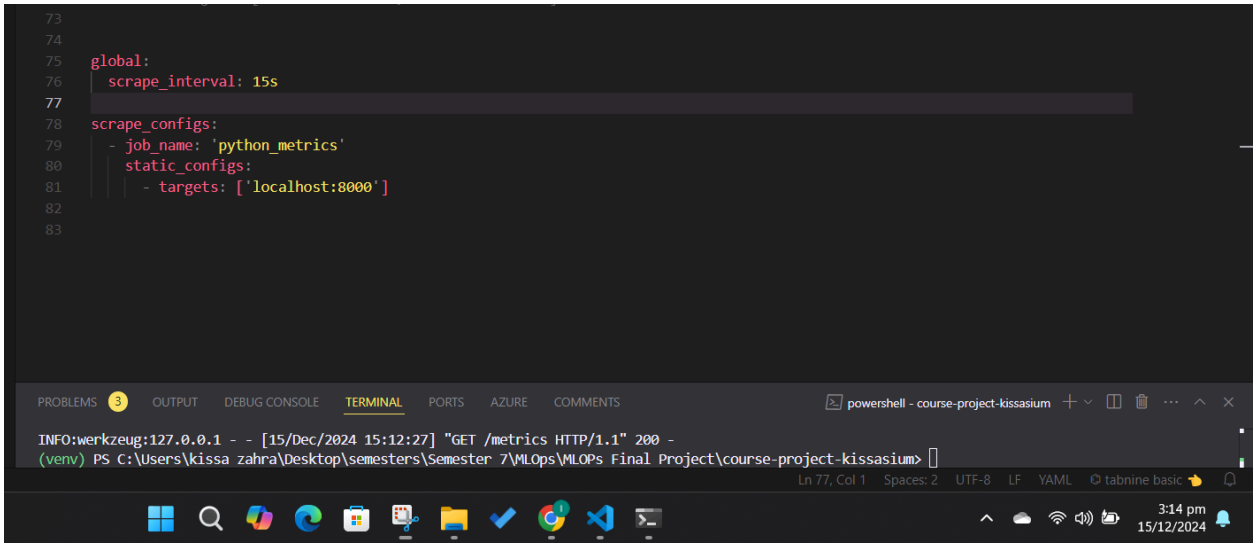


Task 3: Monitoring and Live Testing

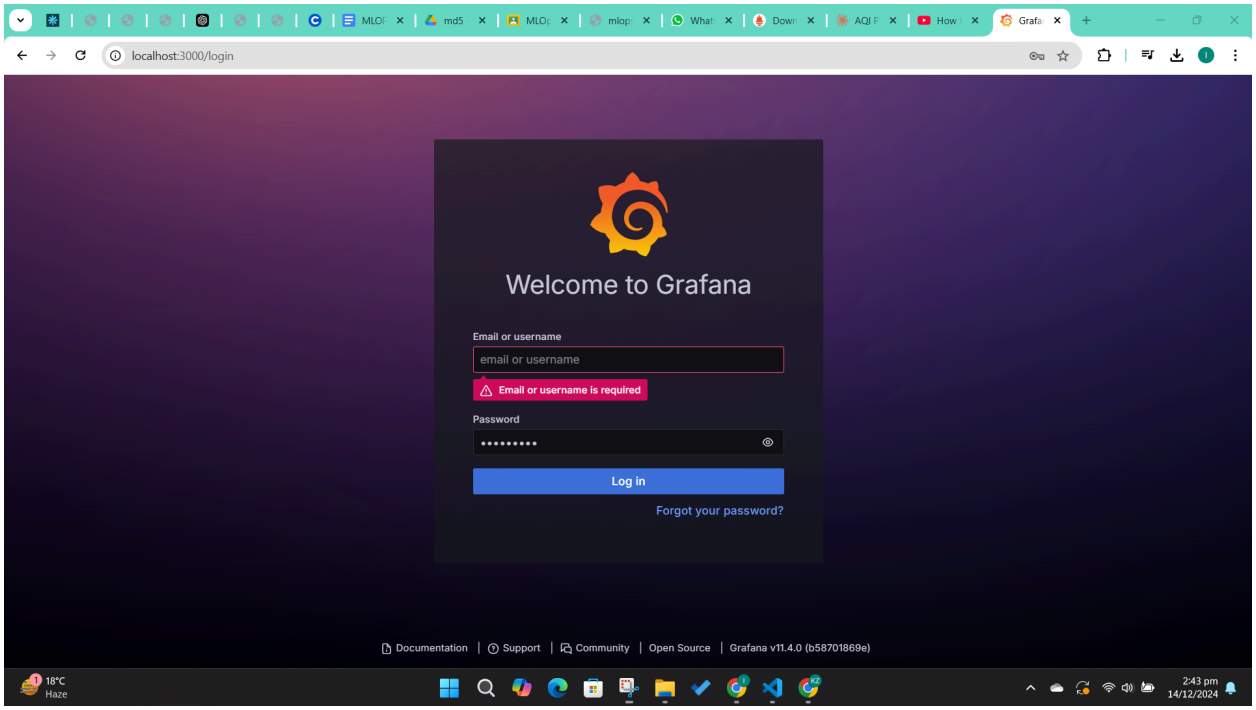
Objective: Test the pipeline with live data and monitor the deployed system.

1. Set Up Monitoring

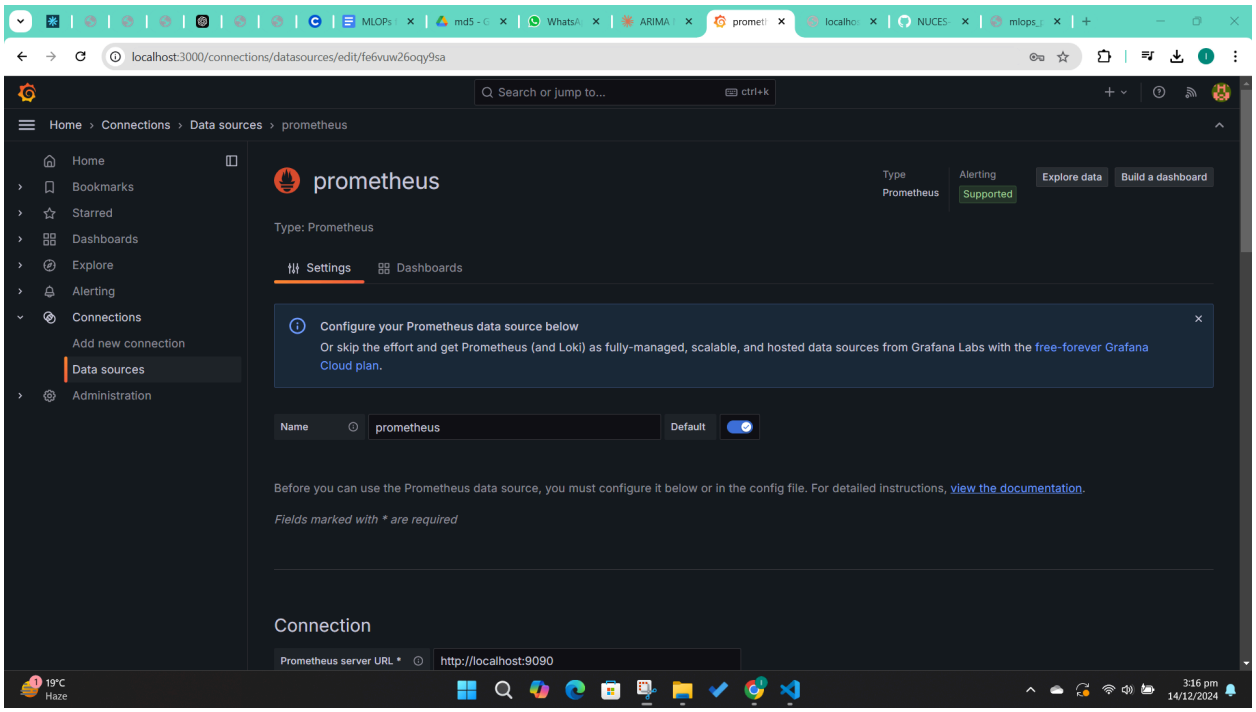
Updating prometheus file to:

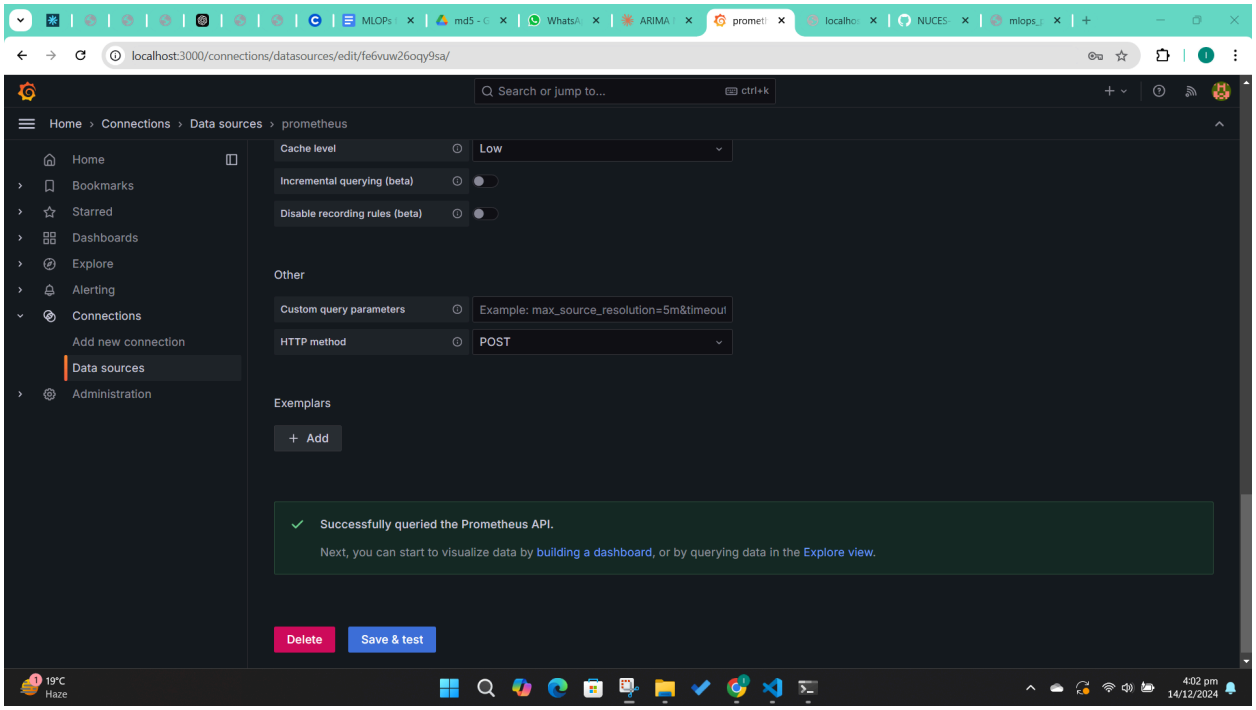


rnd ran it.

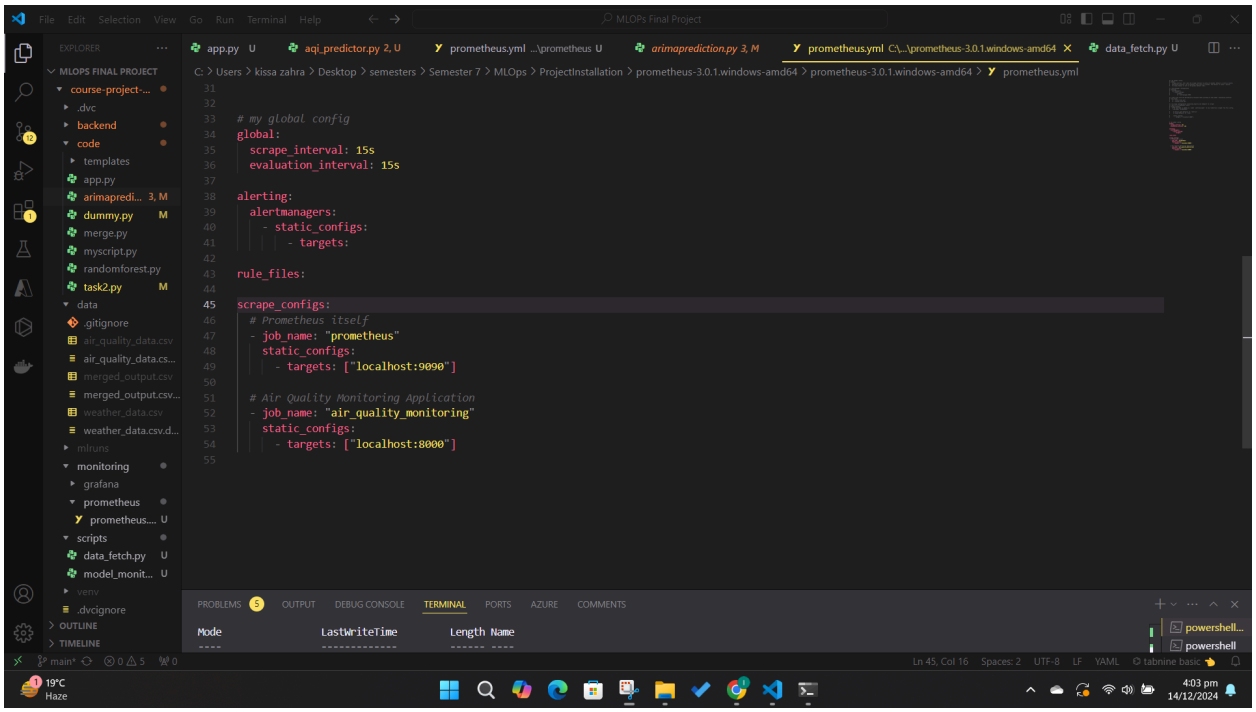


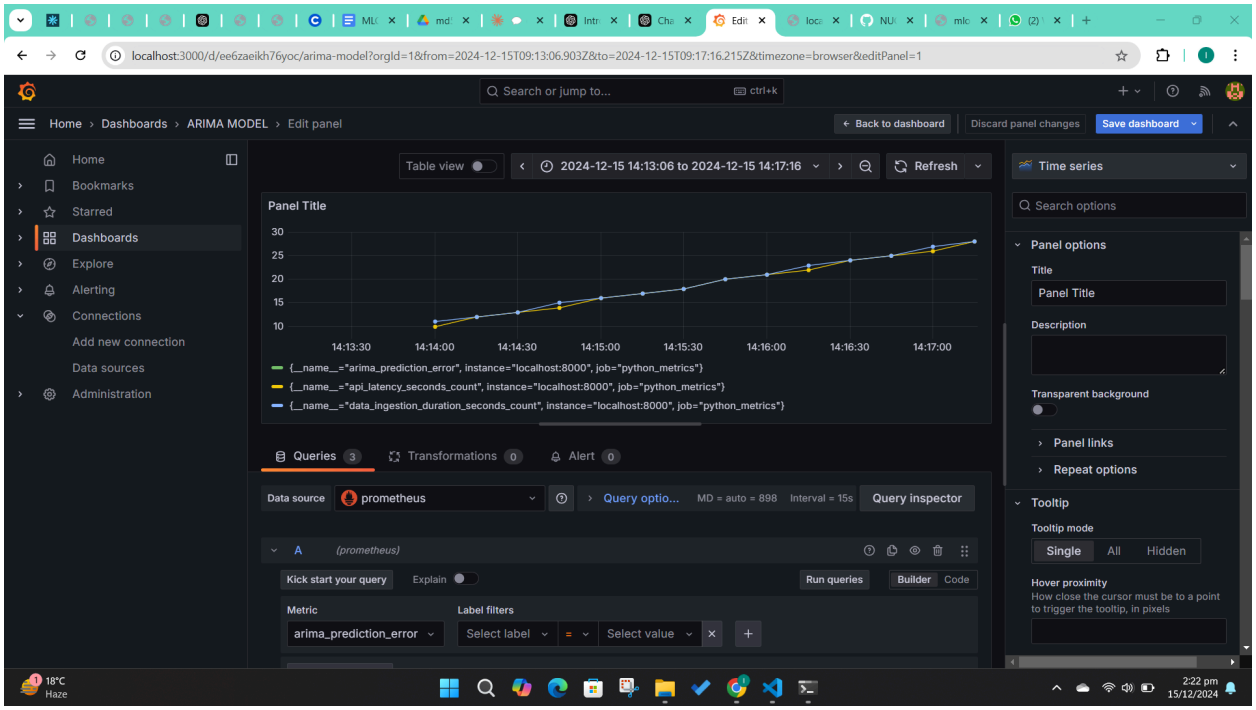
pass: admin or admin1



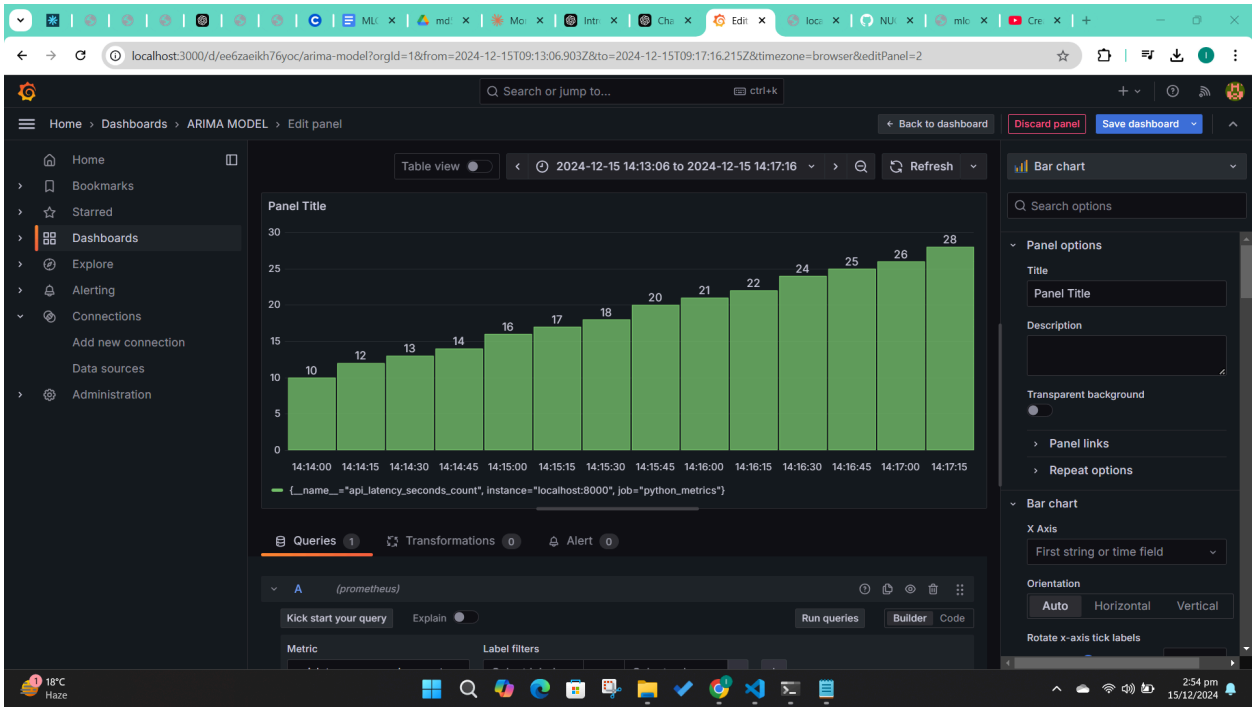


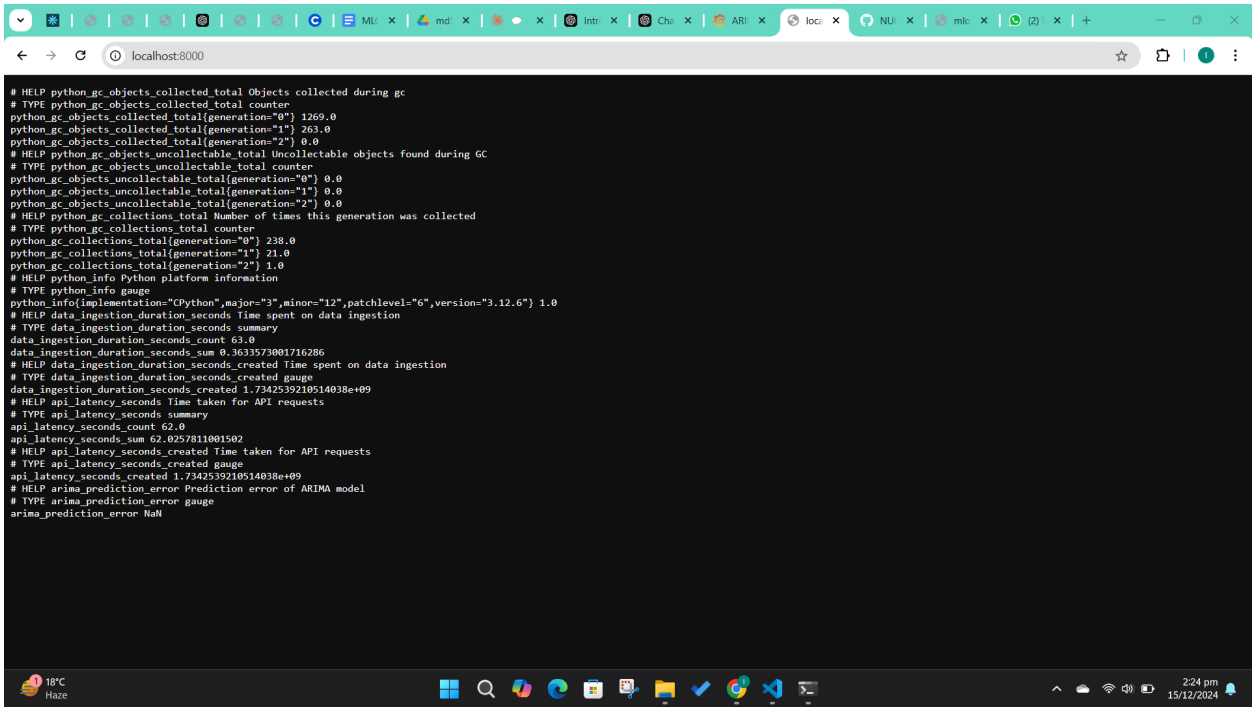
Updated the prometheus.yml file with the code:



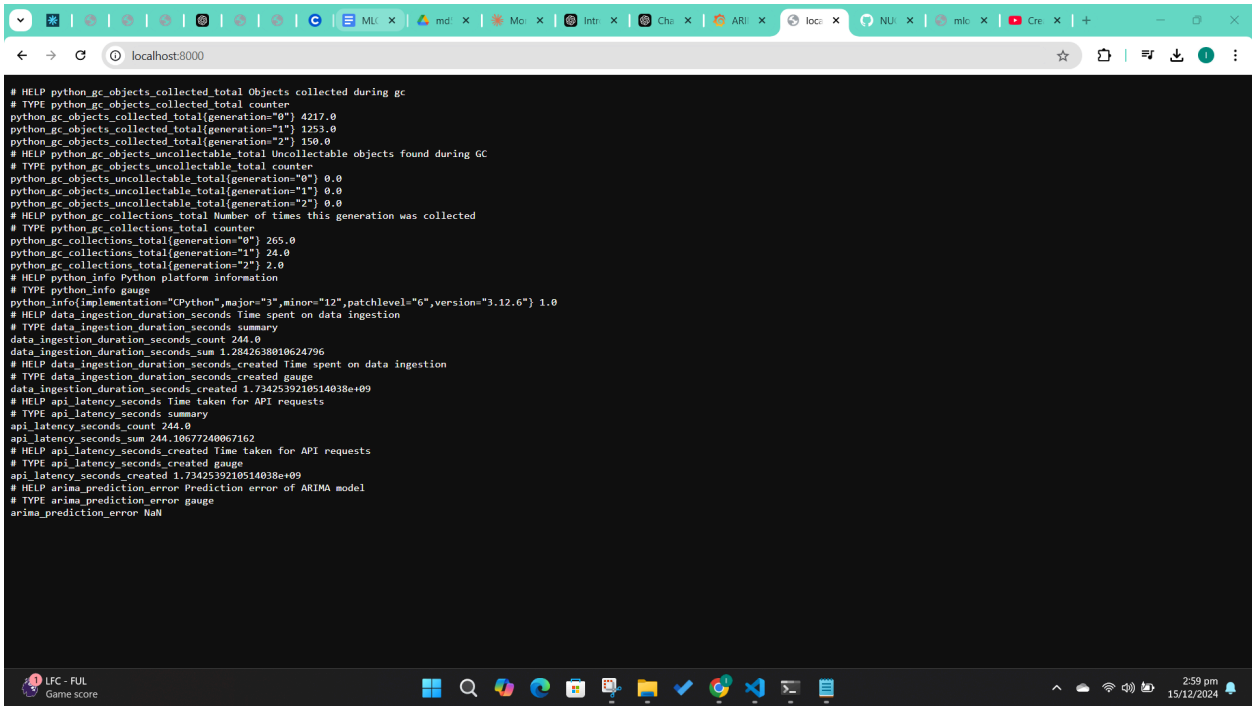


API_LATENCY_SECOND_COUNT



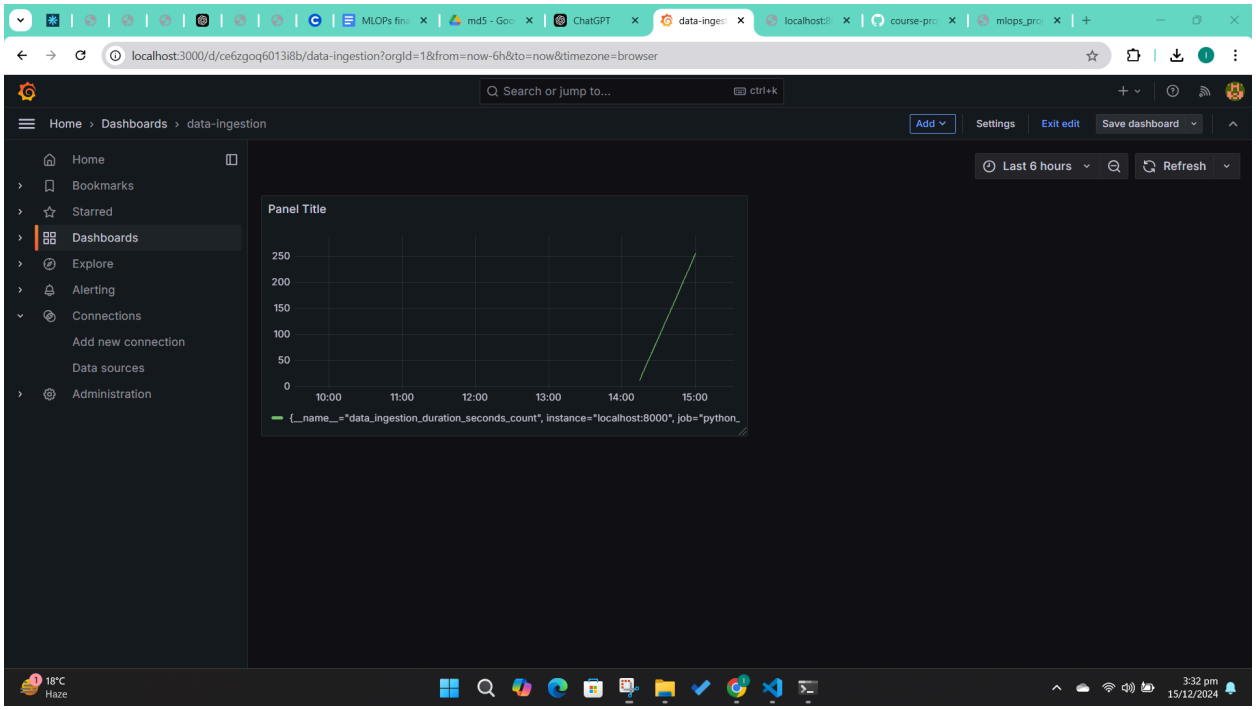


Information present in localhost:8000



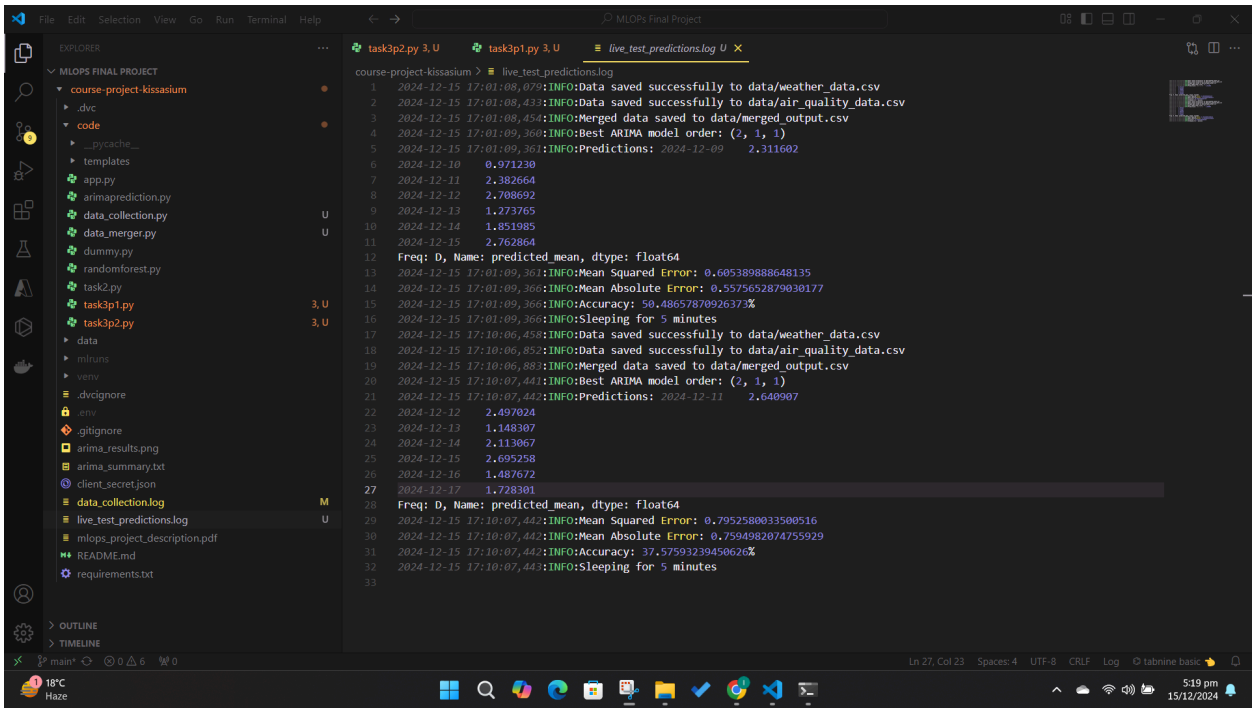
Updated after half an hour

.



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**



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 cam 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.