MLOPs Project
Environmental Monitoring and Pollution Prediction System

Muhammad Ali
21I-0887
A

Task 1: Managing Environmental Data with DVC

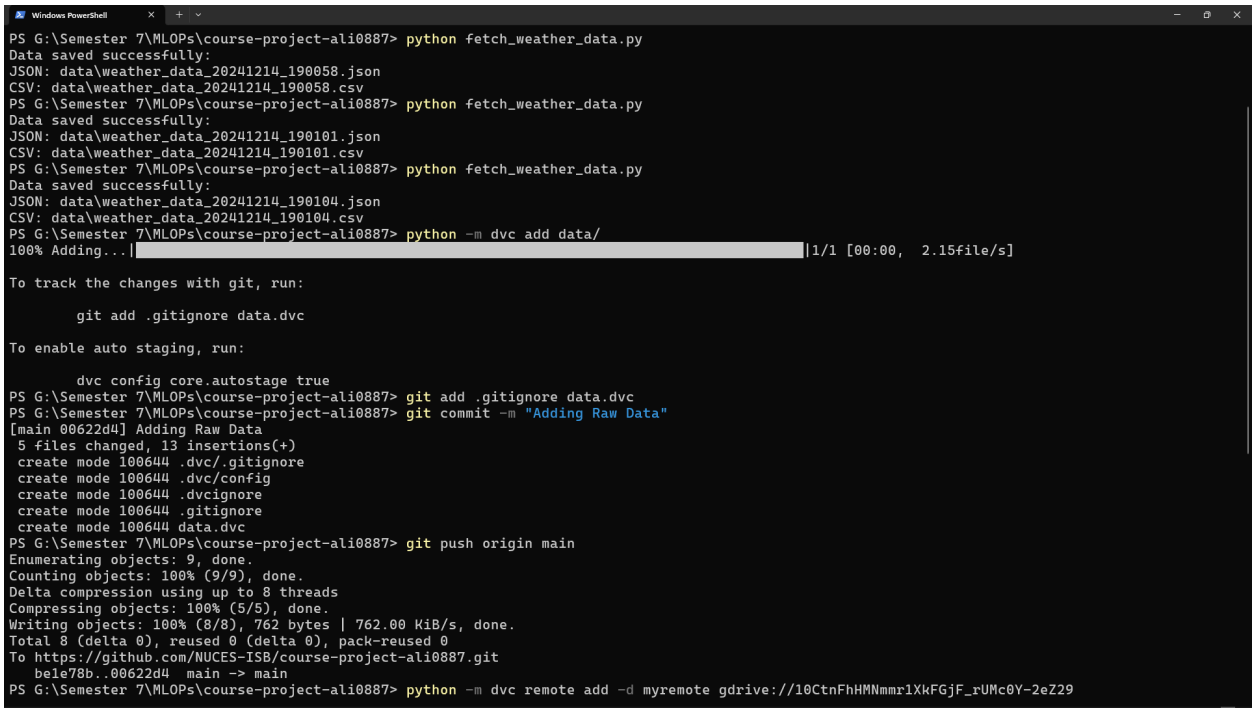
The task involved setting up an automated DVC pipeline that will automatically get data using IQAir API and then automatically push the data changes to Google Drive which we are using as our remote source.

The automation job is scheduled to run every 10 minute to get data from IQ AIR and then pushes relevant changes to Github and Google Drive.

For Google Drive, a service account was configured that provided us with a JSON file to add to our application for data pushing. All environment variables were kept safely in an environment file.

The Python Script was written that automatically fetches important metrics such as time, city, country, temperature, humidity, wind speed, air quality us, and air quality cn and saves them as CSV and JSON files accordingly. The Python Script is then run by a BAT script every 10 minute to log all data.

The commands used to automate the process are:



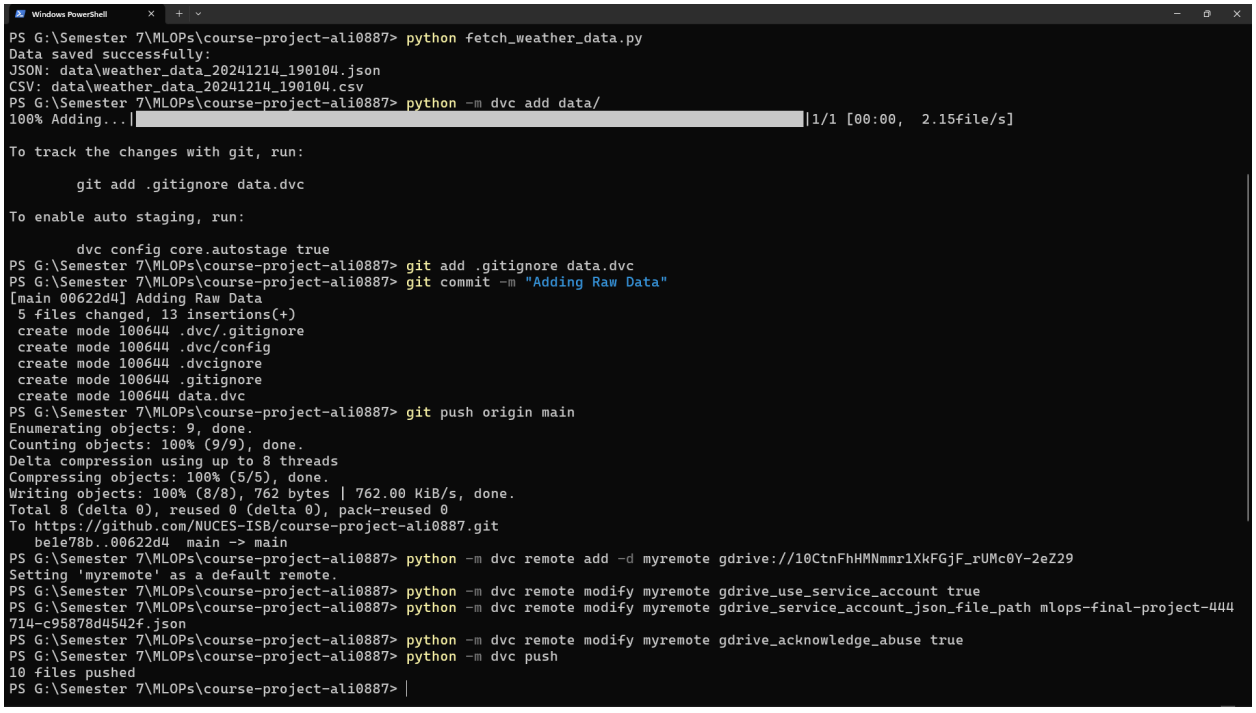
```
Windows PowerShell
PS G:\Semester 7\MLOPs\course-project-ali0887> python fetch_weather_data.py
Data saved successfully:
JSON: data\weather_data_20241214_190058.json
CSV: data\weather_data_20241214_190058.csv
PS G:\Semester 7\MLOPs\course-project-ali0887> python fetch_weather_data.py
Data saved successfully:
JSON: data\weather_data_20241214_190101.json
CSV: data\weather_data_20241214_190101.csv
PS G:\Semester 7\MLOPs\course-project-ali0887> python fetch_weather_data.py
Data saved successfully:
JSON: data\weather_data_20241214_190104.json
CSV: data\weather_data_20241214_190104.csv
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc add data/
100% Adding...|                                                                                |1/1 [00:00,  2.15file/s]

To track the changes with git, run:

        git add .gitignore data.dvc

To enable auto staging, run:

        dvc config core.autostage true
PS G:\Semester 7\MLOPs\course-project-ali0887> git add .gitignore data.dvc
PS G:\Semester 7\MLOPs\course-project-ali0887> git commit -m "Adding Raw Data"
[main 00622d4] Adding Raw Data
 5 files changed, 13 insertions(+)
 create mode 100644 .dvc/.gitignore
 create mode 100644 .dvc/config
 create mode 100644 .dvcignore
 create mode 100644 .gitignore
 create mode 100644 data.dvc
PS G:\Semester 7\MLOPs\course-project-ali0887> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 762 bytes | 762.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NUCES-ISB/course-project-ali0887.git
   be1e78b..00622d4  main -> main
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc remote add -d myremote gdrive://10CtnFhHMNmmr1XkFGjF_rUMc0Y-2eZ29
```



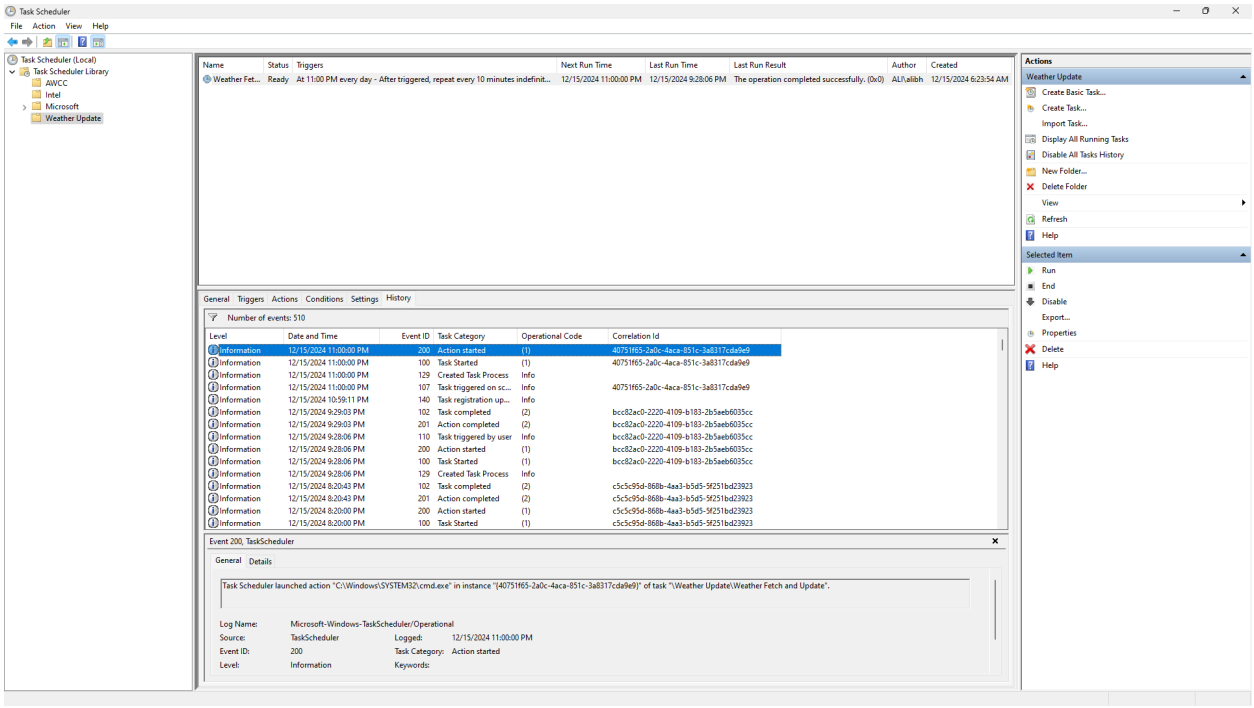
```
Windows PowerShell
PS G:\Semester 7\MLOPs\course-project-ali0887> python fetch_weather_data.py
Data saved successfully:
JSON: data\weather_data_20241214_190104.json
CSV: data\weather_data_20241214_190104.csv
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc add data/
100% Adding...|                                                                                |1/1 [00:00,  2.15file/s]

To track the changes with git, run:

        git add .gitignore data.dvc

To enable auto staging, run:

        dvc config core.autostage true
PS G:\Semester 7\MLOPs\course-project-ali0887> git add .gitignore data.dvc
PS G:\Semester 7\MLOPs\course-project-ali0887> git commit -m "Adding Raw Data"
[main 00622d4] Adding Raw Data
 5 files changed, 13 insertions(+)
 create mode 100644 .dvc/.gitignore
 create mode 100644 .dvc/config
 create mode 100644 .dvcignore
 create mode 100644 .gitignore
 create mode 100644 data.dvc
PS G:\Semester 7\MLOPs\course-project-ali0887> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 762 bytes | 762.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NUCES-ISB/course-project-ali0887.git
   be1e78b..00622d4  main -> main
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc remote add -d myremote gdrive://10CtnFhHMNmmr1XkFGjF_rUMc0Y-2eZ29
Setting 'myremote' as a default remote.
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc remote modify myremote gdrive_use_service_account true
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc remote modify myremote gdrive_service_account_json_file_path mlops-final-project-444
714-c95878d4542f.json
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc remote modify myremote gdrive_acknowledge_abuse true
PS G:\Semester 7\MLOPs\course-project-ali0887> python -m dvc push
10 files pushed
PS G:\Semester 7\MLOPs\course-project-ali0887>
```



Task 2: Pollution Trend Prediction with MLFlow

For the task of incorporating MLFlow with our DVC pipeline and then filtering out the best model, an MLFlow experiment was setup that would test and log a combination of models with the following parameters:

Sequence Length : 2, 6, 12, 24
Hidden LSTM Units: 16, 32, 64, 128
Dropout: 0.1, 0.2, 0.3

Learning Rate: 0.001
Epochs: 30, 50
Batch Size: 4, 8, 16

The training is done in Tensorflow using 2 LSTM layers with a dropout layer after each and a Dense layer for predicting.

The total number of combinations came out to 192 and all the results were logged in MLFlow. Chosen Metrics are as follow:

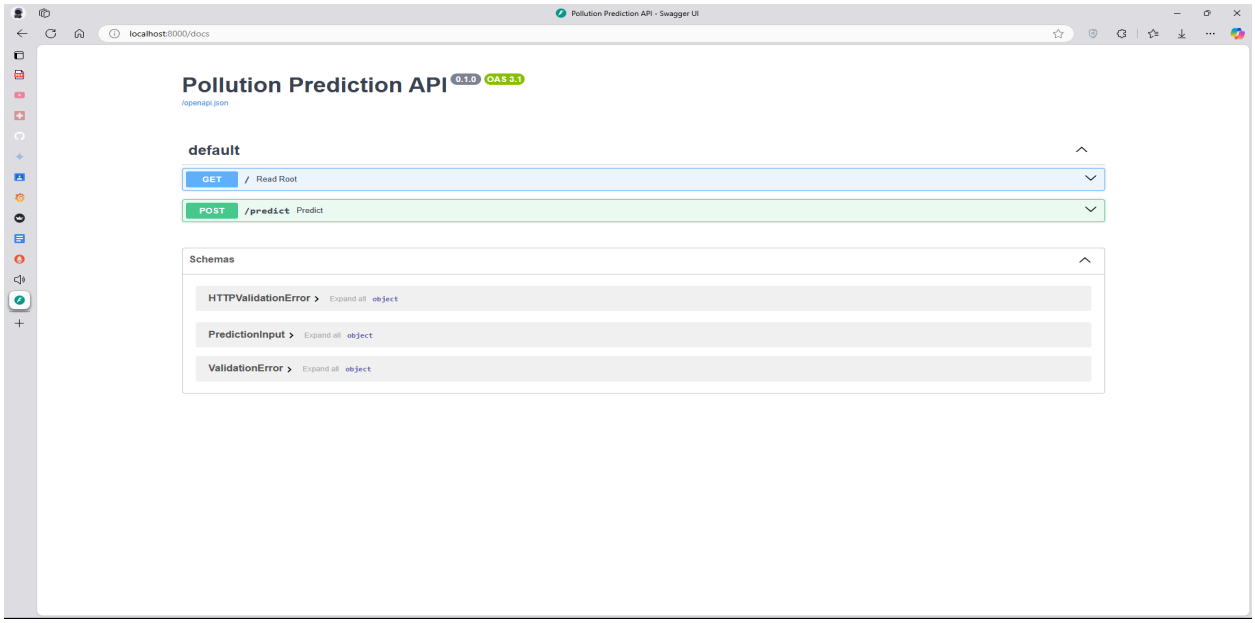
1. Mean Accuracy Error (MAE)
2. Mean Squared Error (MSE)
3. Root Mean Squared Error (RMSE)

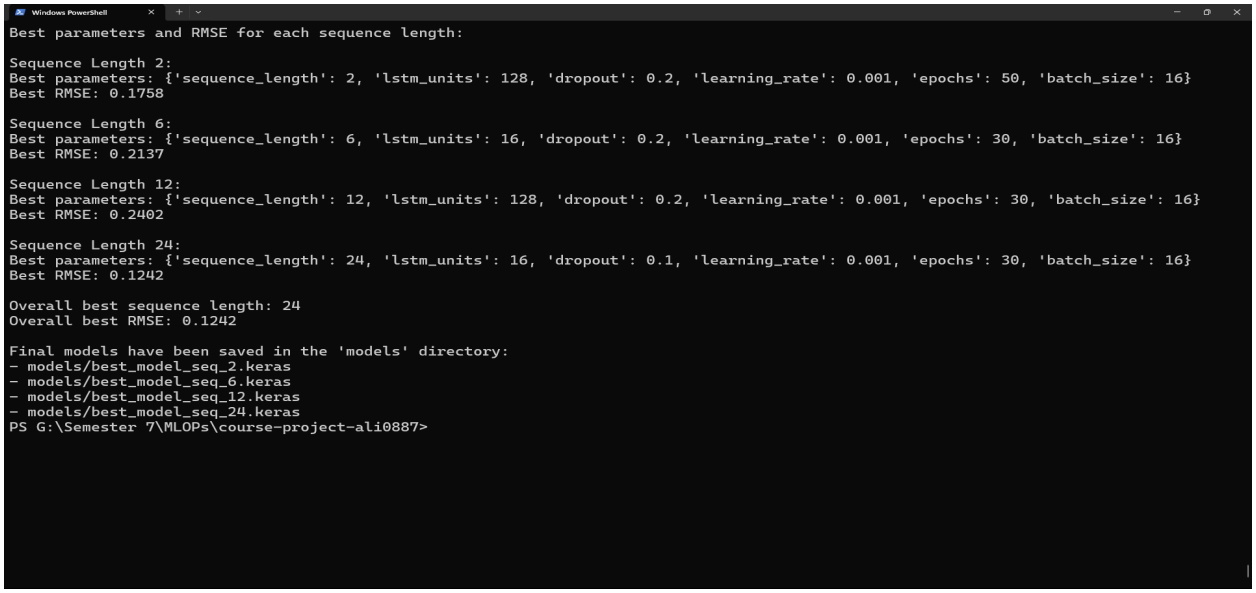
As per the results, the best results came for the model with sequence length 2, with 128 hidden lstm units which makes sense since in a timeframe of 20 minutes results will not change much and 128 hidden lstm units allow for a better inference due to more lstm cell retaining more knowledge.

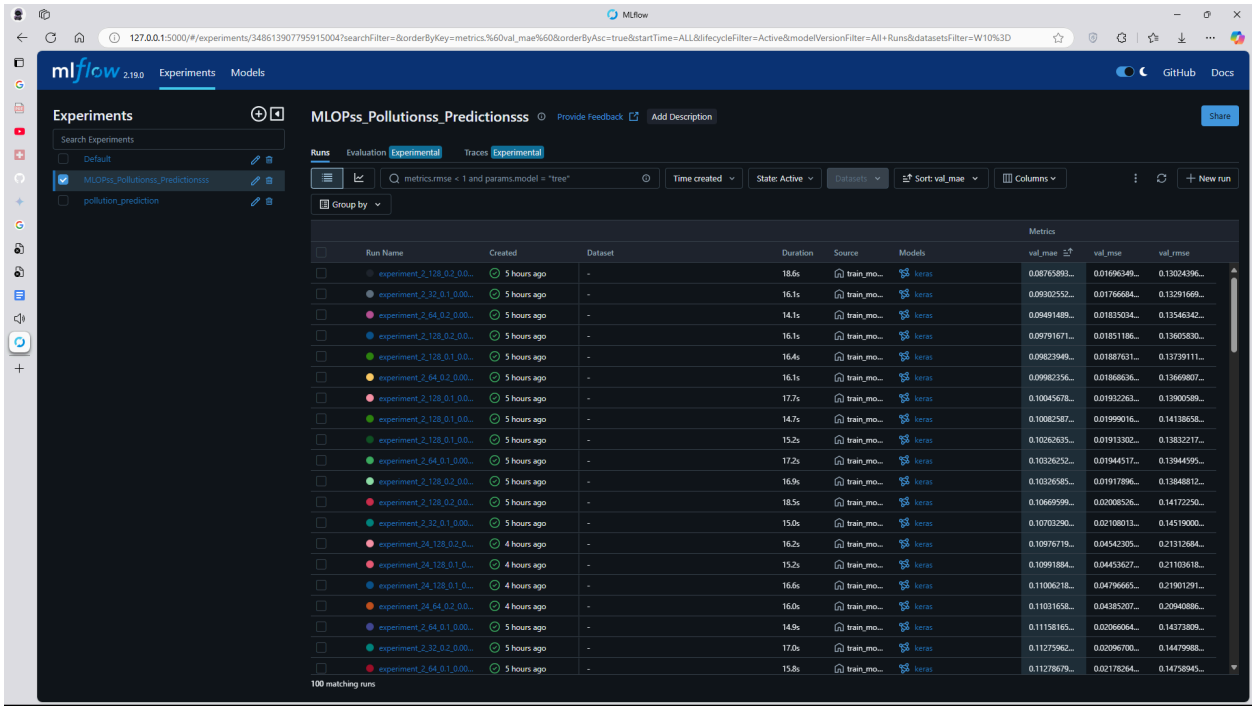
However, during inference the best results came with the model with sequence length = 24 or 4 hours history.

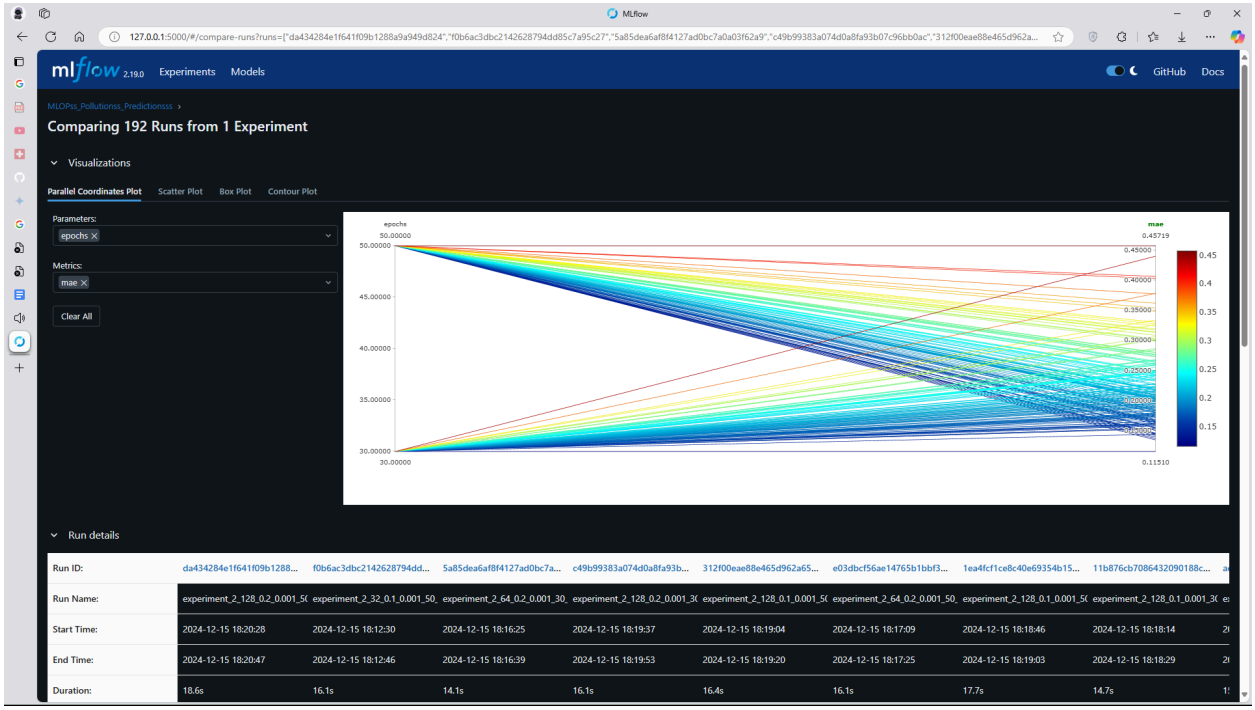
This could be partially because more data equals better predictions. Once again, with 128 hidden lstm units the metrics were the best.

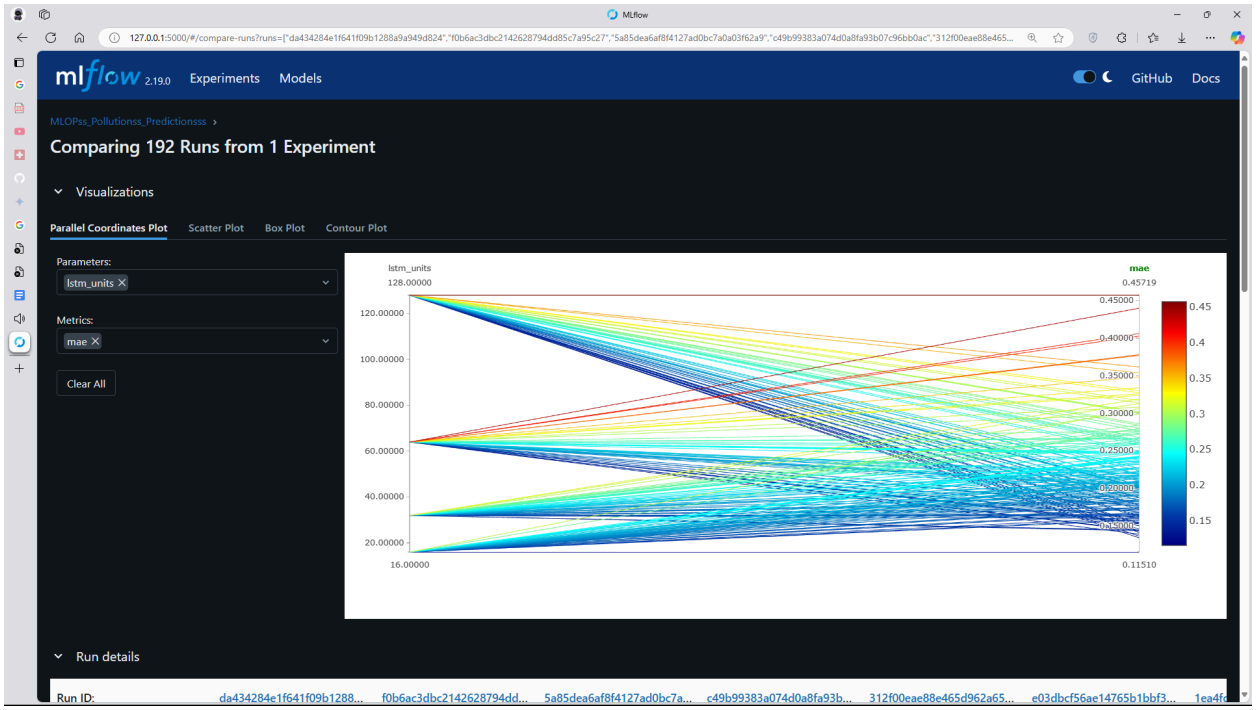
Each Model was deployed using a FAST API on the local machine.

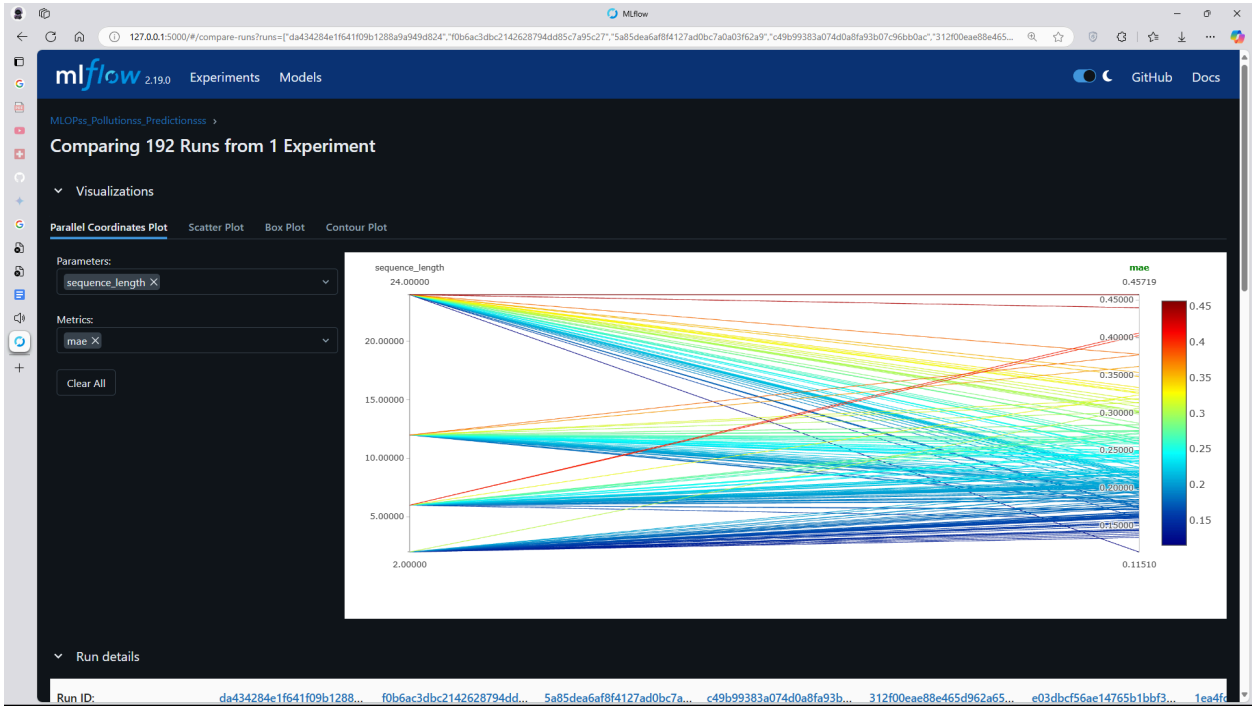


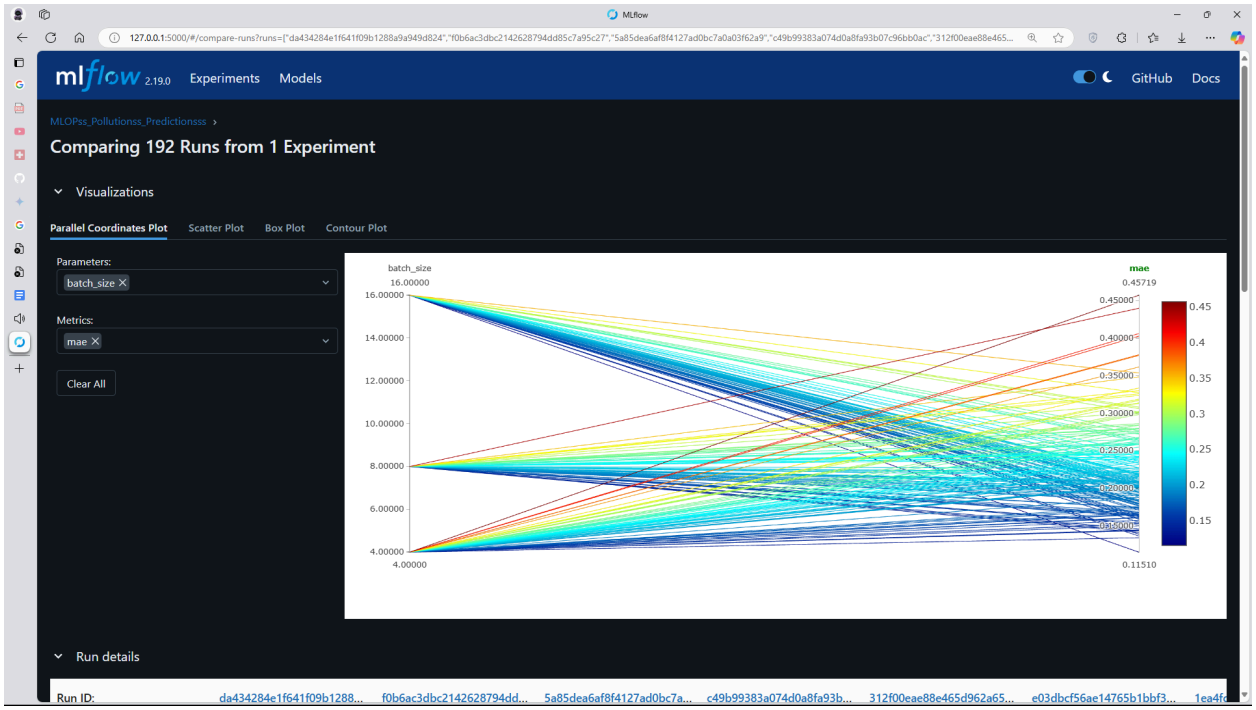



MLflow
127.0.0.1:5000/#/experiments/348613907795915004?searchFilter=&orderByKey=metrics.%60val_mae%60&orderByAsc=true&startTime=ALL&lifecycleFilter=Active&modelVersionFilter=All+Runs&datasetsFilter=W10%3D
mlflow 2.19.0
Experiments
Models
GitHub
Docs
Experiments
Search Experiments
Default
MLOPss_Pollutionss_Predictionsss
pollution_prediction
MLOPss_Pollutionss_Predictionsss
Provide Feedback
Add Description
Share
Runs
Evaluation Experimental
Traces Experimental
metrics.rmse < 1 and params.model = "tree"
Time created
State: Active
Datasets
Sort: val_mae
Columns
New run
Group by
Metrics
Run Name
Created
Dataset
Duration
Source
Models
val_mae
val_mse
val_rmse
100 matching runs


MLflow
127.0.0.1:5000/#/compare-runs?runs=["da434284e1f641f09b1288a9a949d824","f0b6ac3dbc2142628794dd85c7a95c27","5a85dea6af8f4127ad0bc7a0a03f62a9","c49b99383a074d0a8fa93b07c96bb0ac","312f00eae88e465d962a...
mlflow 2.19.0
Experiments
Models
GitHub
Docs
MLOPss_Pollutionss_Predictionsss
Comparing 192 Runs from 1 Experiment
Visualizations
Parallel Coordinates Plot
Scatter Plot
Box Plot
Contour Plot
Parameters:
epochs
Metrics:
mae
Clear All
epochs
mae
50.00000
45.00000
40.00000
35.00000
30.00000
0.45719
0.11510
Run details
Run ID:
Run Name:
Start Time:
End Time:
Duration:
2024-12-15 18:20:28
2024-12-15 18:12:30
2024-12-15 18:16:25
2024-12-15 18:19:37
2024-12-15 18:19:04
2024-12-15 18:17:09
2024-12-15 18:18:46
2024-12-15 18:18:14
2024-12-15 18:20:47
2024-12-15 18:12:46
2024-12-15 18:16:39
2024-12-15 18:19:53
2024-12-15 18:19:20
2024-12-15 18:17:25
2024-12-15 18:19:03
2024-12-15 18:18:29
18.6s
16.1s
14.1s
16.1s
16.4s
16.1s
17.7s
14.7s

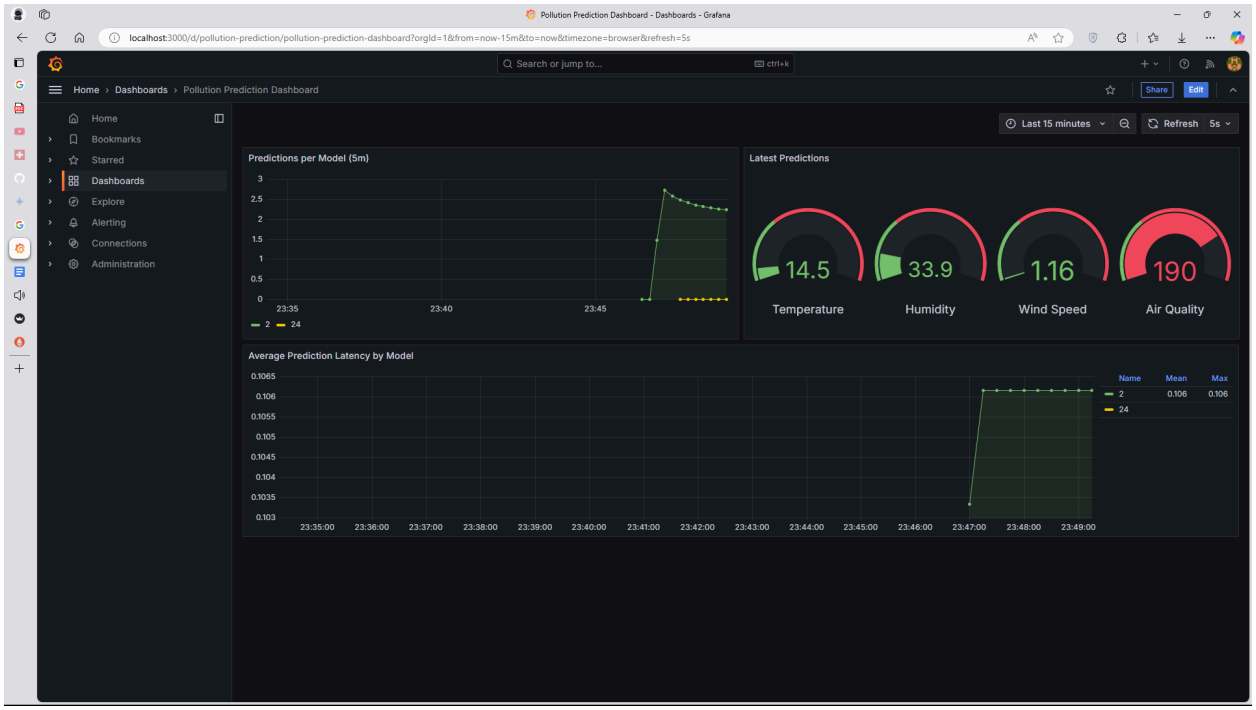

MLflow
mlflow 2.19.0
Experiments
Models
GitHub
Docs
MLOPss_Pollutionss_Predictionsss
Comparing 192 Runs from 1 Experiment
Visualizations
Parallel Coordinates Plot
Scatter Plot
Box Plot
Contour Plot
Parameters:
lstm_units
Metrics:
mae
Clear All
lstm_units
128.00000
120.00000
100.00000
80.00000
60.00000
40.00000
20.00000
16.00000
mae
0.45719
0.45000
0.40000
0.35000
0.30000
0.25000
0.20000
0.15000
0.11510
0.45
0.4
0.35
0.3
0.25
0.2
0.15
Run details
Run ID:
da434284e1f641f09b1288...
f0b6ac3dbc2142628794dd...
5a85dea6af8f4127ad0bc7a...
c49b99383a074d0a8fa93b...
312f00eae88e465d962a65...
e03dbcf56ae14765b1bbf3...

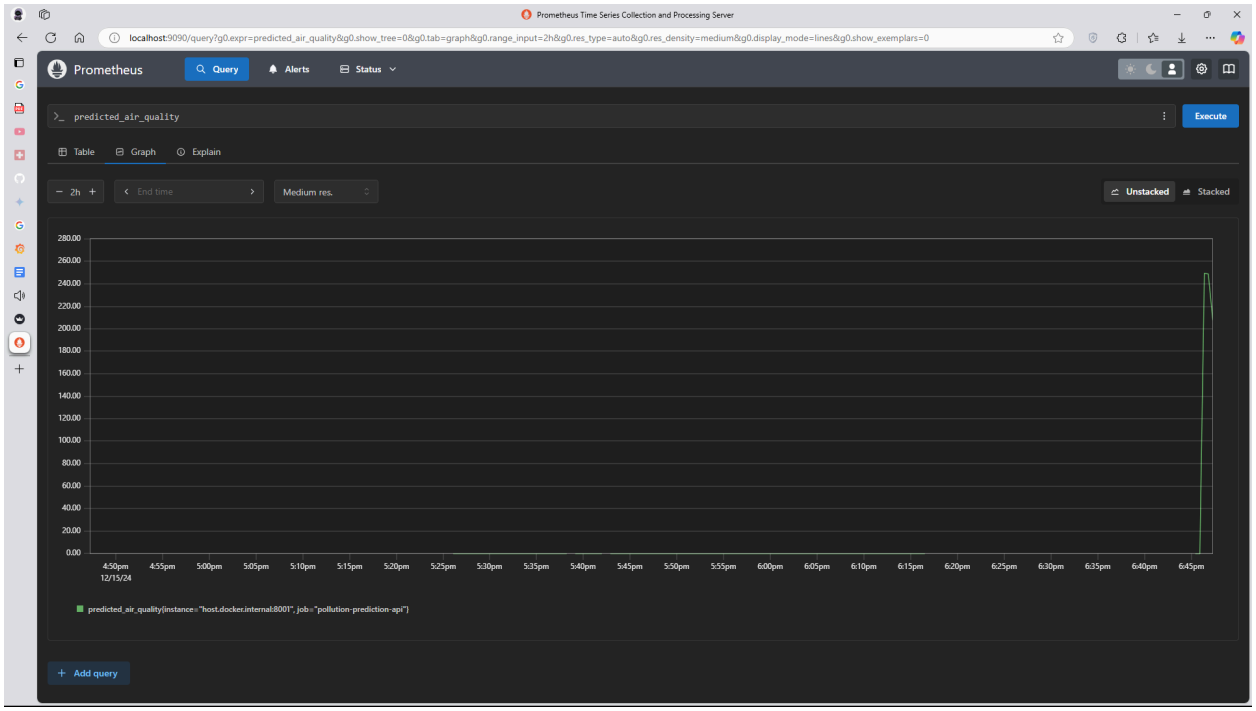

MLflow
mlflow 2.19.0
Experiments
Models
GitHub
Docs
MLOPss_Pollutionss_Predictionsss
Comparing 192 Runs from 1 Experiment
Visualizations
Parallel Coordinates Plot
Scatter Plot
Box Plot
Contour Plot
Parameters:
sequence_length
Metrics:
mae
Clear All
sequence_length
24.00000
20.00000
15.00000
10.00000
5.00000
2.00000
mae
0.45719
0.45000
0.40000
0.35000
0.30000
0.25000
0.20000
0.15000
0.11510
0.45
0.4
0.35
0.3
0.25
0.2
0.15
Run details
Run ID:
da434284e1f641f09b1288...
f0b6ac3dbc2142628794dd...
5a85dea6af8f4127ad0bc7a...
c49b99383a074d0a8fa93b...
312f00eae88e465d962a65...
e03dbcf56ae14765b1bbf3...



Task 3: Monitoring and Live Testing

Setting up Grafana and Prometheus on Docker involved writing up a Docker Compose Script that automatically fetches relevant data from when the API is called and displays these graphs in Prometheus and Grafana.


Pollution Prediction Dashboard - Dashboards - Grafana
localhost:3000/d/pollution-prediction/pollution-prediction-dashboard?orgId=1&from=now-15m&to=now&timezone=browser&refresh=5s
Search or jump to...
Home › Dashboards › Pollution Prediction Dashboard
Share
Edit
Home
Bookmarks
Starred
Dashboards
Explore
Alerting
Connections
Administration
Last 15 minutes
Refresh 5s
Predictions per Model (5m)
3
2.5
2
1.5
1
0.5
0
23:35
23:40
23:45
2
24
Latest Predictions
14.5
Temperature
33.9
Humidity
1.16
Wind Speed
190
Air Quality
Average Prediction Latency by Model
0.1065
0.106
0.1055
0.105
0.1045
0.104
0.1035
0.103
23:35:00
23:36:00
23:37:00
23:38:00
23:39:00
23:40:00
23:41:00
23:42:00
23:43:00
23:44:00
23:45:00
23:46:00
23:47:00
23:48:00
23:49:00
Name
Mean
Max
2
0.106
0.106
24


Prometheus Time Series Collection and Processing Server
localhost:9090/query?g0.expr=predicted_air_quality&g0.show_tree=0&g0.tab=graph&g0.range_input=2h&g0.res_type=auto&g0.res_density=medium&g0.display_mode=lines&g0.show_exemplars=0
Prometheus
Query
Alerts
Status
predicted_air_quality
Execute
Table
Graph
Explain
2h
End time
Medium res.
Unstacked
Stacked
280.00
260.00
240.00
220.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
4:50pm
12/15/24
4:55pm
5:00pm
5:05pm
5:10pm
5:15pm
5:20pm
5:25pm
5:30pm
5:35pm
5:40pm
5:45pm
5:50pm
5:55pm
6:00pm
6:05pm
6:10pm
6:15pm
6:20pm
6:25pm
6:30pm
6:35pm
6:40pm
6:45pm
predicted_air_quality{instance="host.docker.internal:8001", job="pollution-prediction-api"}
Add query

Improvements / Optimizations:

As far as Improvements and Optimizations are considered, while not implemented, the identified problems are there not being a model trained on time of day as during the night the temperature usually decreases alongside the overall pollution.

Additionally, incorporating situational awareness into the model like what kind of weather helps to predict the pollution levels better e.g. when it is raining, air pollution is very low.

Moreover, the data collection for this task was low so the model may not converge effectively for different scenarios. Training on more data will help the model identify pollution better.

Alongside there, one may also consider the model architecture as 2 LSTM Layers may not be enough to model a complex task such as weather pollution prediction.