**MLOPS**

**Project Report**

**Name Roll-Num Section**

Muhammad Osama Fahim 21i-0439 G

A blue circle with white text and a letter

Description automatically generated

**Department of Computer Science**

**National University of Computer and Emerging Sciences Islamabad, Pakistan**

**Dated: 15 December 2024**

Integration Process:

* OpenWeatherMap API: First we signed in their website and registered an API for environmental change.
* Initialized DVC repository for versioning collected data and integrated it with Google Drive Folder so that new data can be versioned.
* Data collection script: Data\_collection.py uses the obtained API key to fetch data of the given city (latitudes and longitudes are given).
* Version control with DVC: The Data\_collection.py contains the logic for adding the newly obtained data file to DVC G drive.
* Regular Data Fetching: A screen shot of a computer

  Description automatically generated

The \* 0 \* \* \* indicates that the provided script will run after every hour

* Update Data with DVC: Data has been updated with DVC in the run\_DVC() function which is called as soon as new file is fetched using the API which automatically add, commit and push to version and push changes to remote storage.

Model Training:

* ARIMA Model: ARIMA model has been used to forecast “components.pm2\_5”.
* Hyperparameters used:

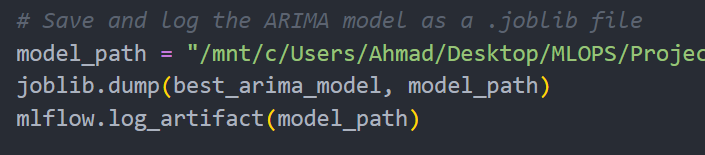
Performed GridSearch on ARMIA (using TimeSeriesSplit for the time series cross-validation)

A black background with yellow and blue text

Description automatically generated

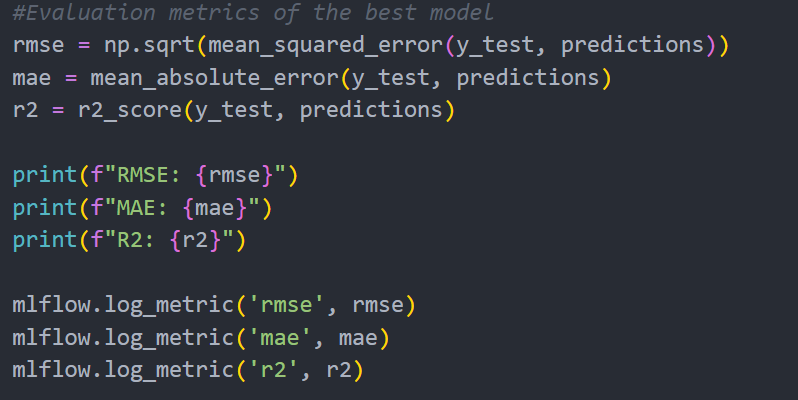
tscv = TimeSeriesSplit(n\_splits=5), Time series cross validation which is important to keep time order.

* Logging Model in MLFlow: Model has been logged as an artifact in MLFlow.

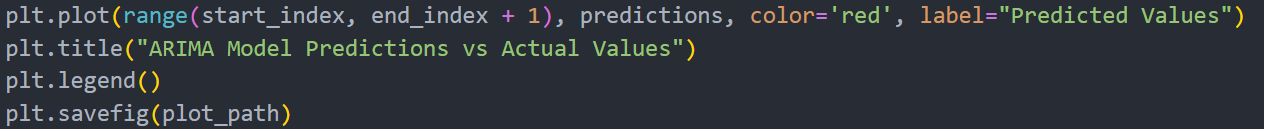


* Evaluation Metrics:

Rmse, mae, and r2 have been tracked of the evaluation metrics.



Evaluation metrics have been logged in MLFlow as well



Model Predictions vs Actual Values Graph has also been saved locally and logged in MLFlow.

Model Deployment:

* FASTAPI has been used to deploy the model
* The latest model dynamically loads from the MLFlow
* The predicted model values along with metrics are shown in the Swagger UI.
* The “unicorn app:app –reload” command launches the deployed API.

Summary report on the system’s live performance:

* MLFLOW showing model metrics which was uploaded to it.

A screenshot of a computer

Description automatically generated

* Model’s MSE and RME values

A screenshot of a computer

Description automatically generated

* App requests, Model’s r2 value and app requests crearted

A screenshot of a computer

Description automatically generated