Learning Session Series

Topic: MLOps for AI Engineer and Data Scientist

Sub-topic: Leveraging Cloud Computing for MLOps

Speaker: Joseph Itopa Abubakar

November 29st, 2021

Omdena

OPEN EDUCATION BY LOCAL CHAPTERS

Learning section objective:

- To gain a fundamental understanding of operations monitoring: infrastructure and model monitoring
- To get introduced to data, and model versioning

Agenda

- ✓ Monitoring and Automation
- ➢ Overview of Monitoring
- ➢ Infrastructure monitoring
- ➢ Monitor and evaluate model performance
- ➢ Maintenance guide for model updating
- ➢ Data pipeline monitoring

Overview of ML Monitoring

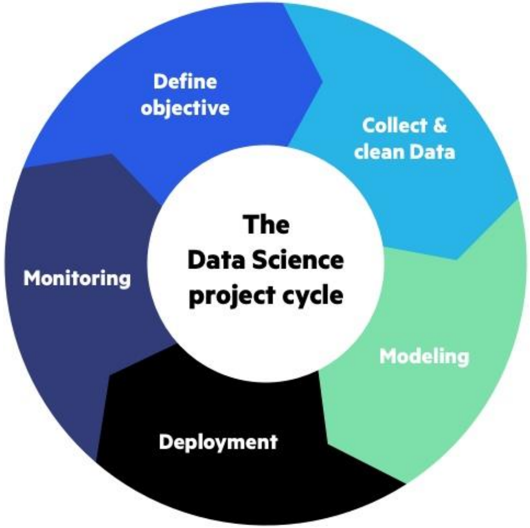


Figure 1a: A data science project life cycle

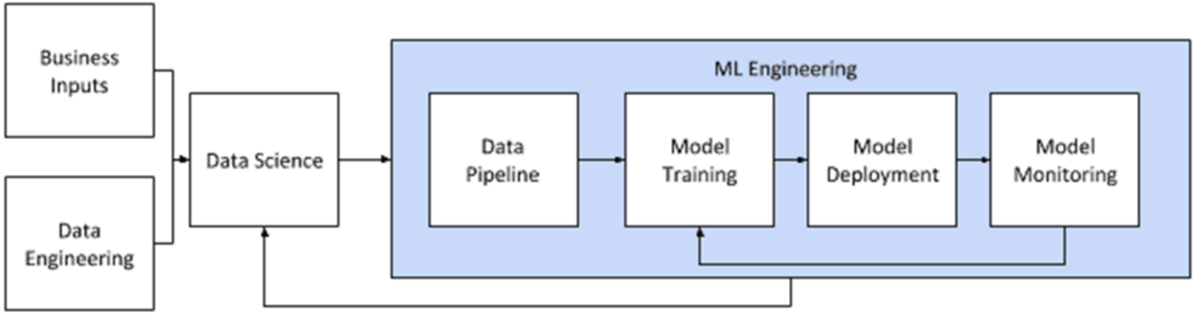


Figure 1b: Breakdown of data science project life cycle

Reference: https://christophergs.com/machine%20learning/2020/03/14/how-to-monitor machine-learning-models/https://www.imperva.com/blog/deployment-isnt-the-final-step monitoring-machine-learning-models-in-production/

Overview of ML Monitoring

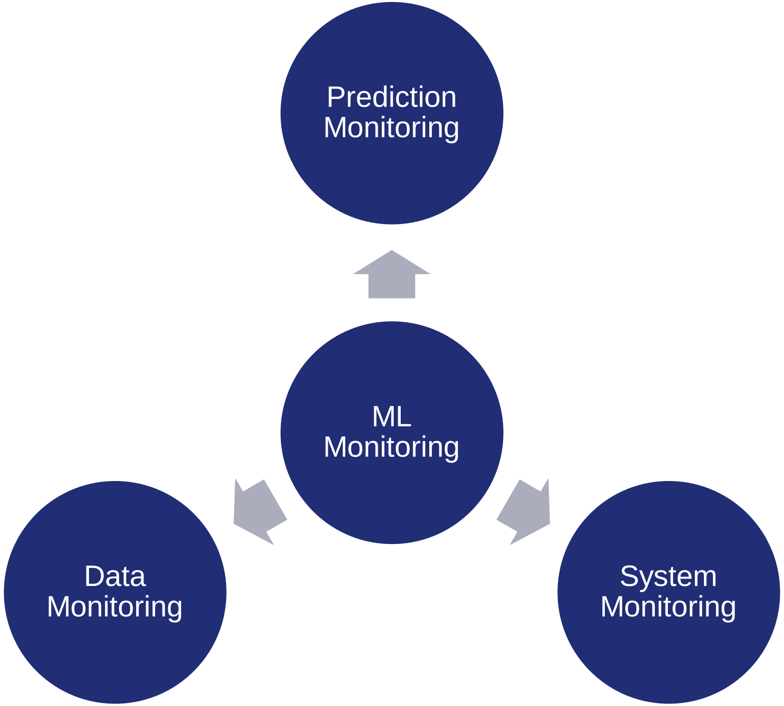


Figure 2: Categories of ML monitoring

- Functional level monitoring: [input(s), output(s)] – Data monitoring, model performance
- Operational level monitoring: System/Resource level monitoring

Reference: https://christophergs.com/machine%20learning/2020/03/14/how-to-monitor-machine-learning-models/

Monitor and evaluate model performance

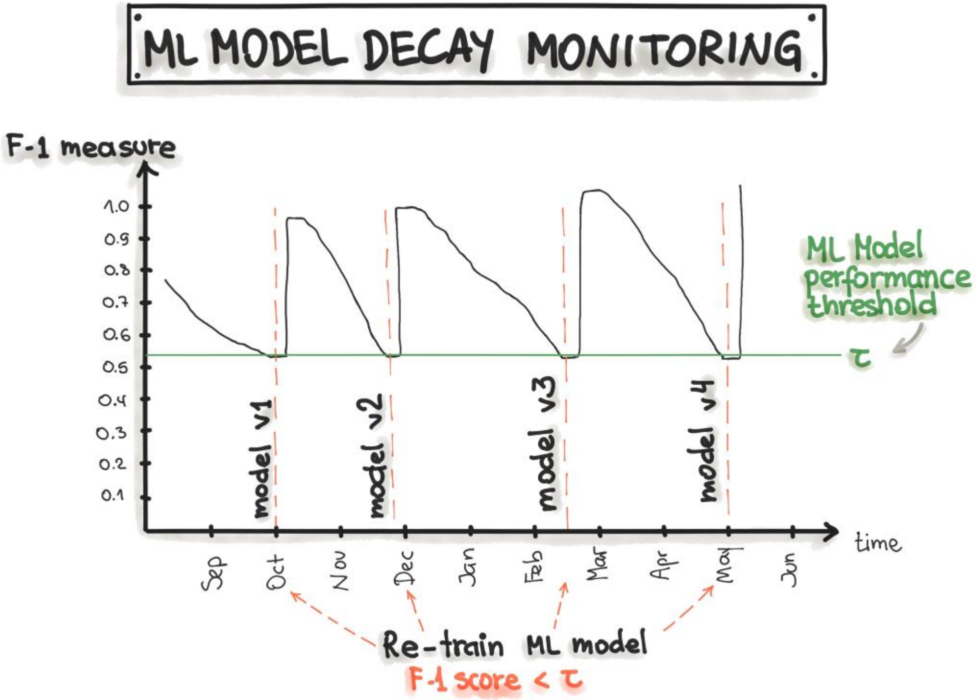


Figure 3: Monitoring model decay

Why do we monitor predictions?

Reference: https://ml-ops.org/content/mlops-principles

Monitor and evaluate model performance

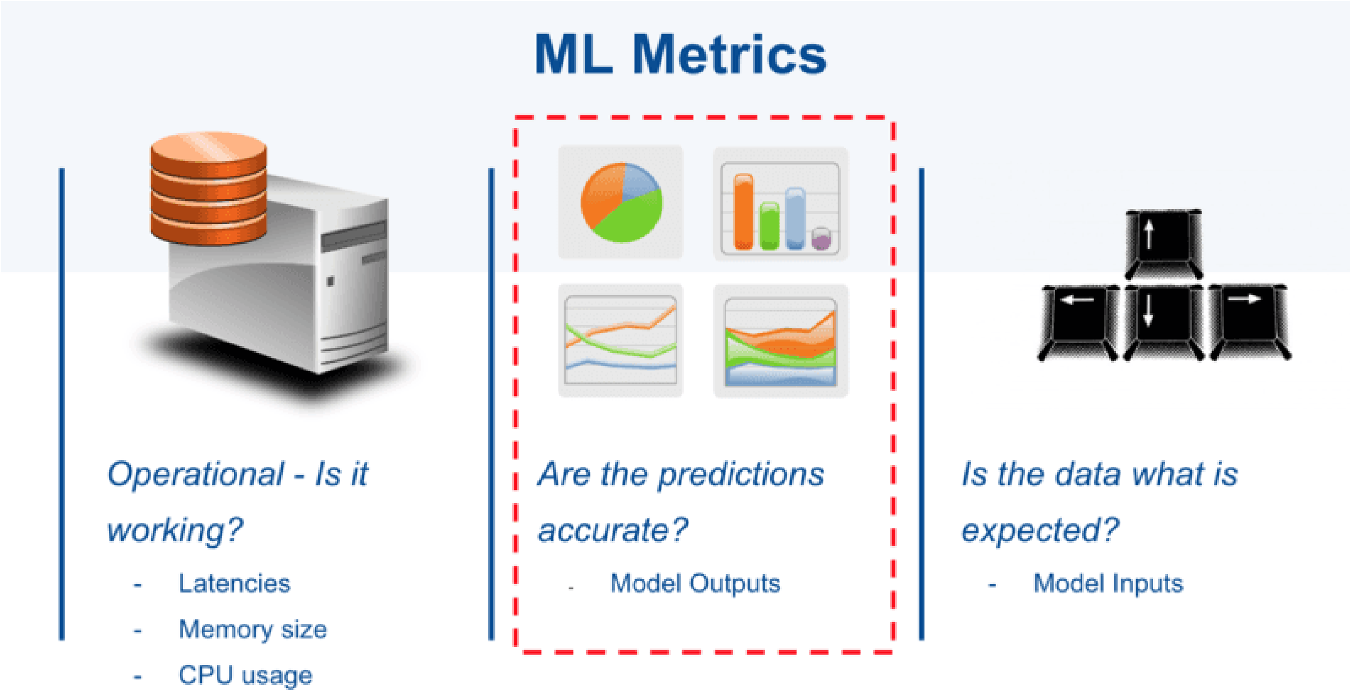


Figure 1: The life cycle of a typical deep learning project

Reference: https://ml-ops.org/content/mlops-principles

Tea Break | Quiz

Why do we monitor model predictions?

a) To observe the model performance over time
b) To avoid data leakage
c) To ignore model metrics
d) Non of the above

What are the three categories of monitoring?

a) System, Model, Operations
b) System, Model, Data
c) Data, Model, f1-score
d) Non of the above

Model Infrastructure monitoring

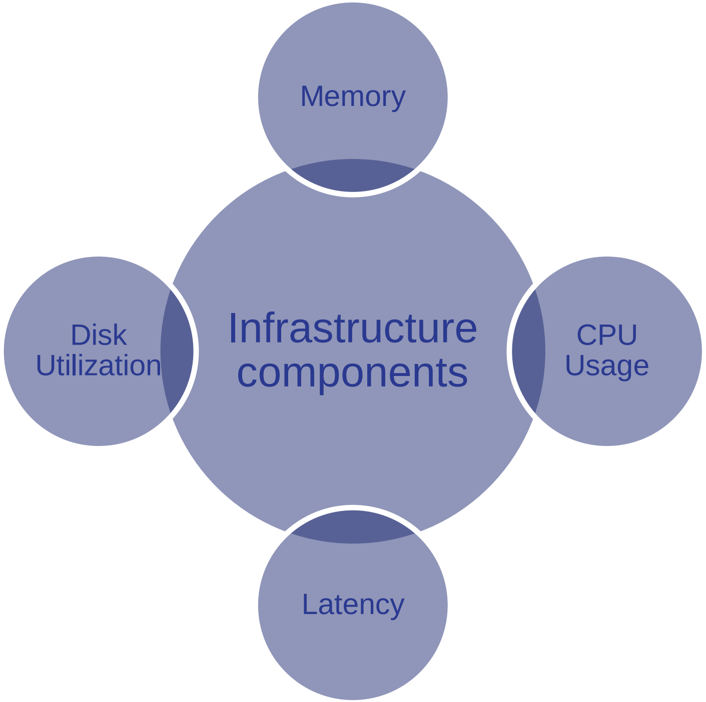
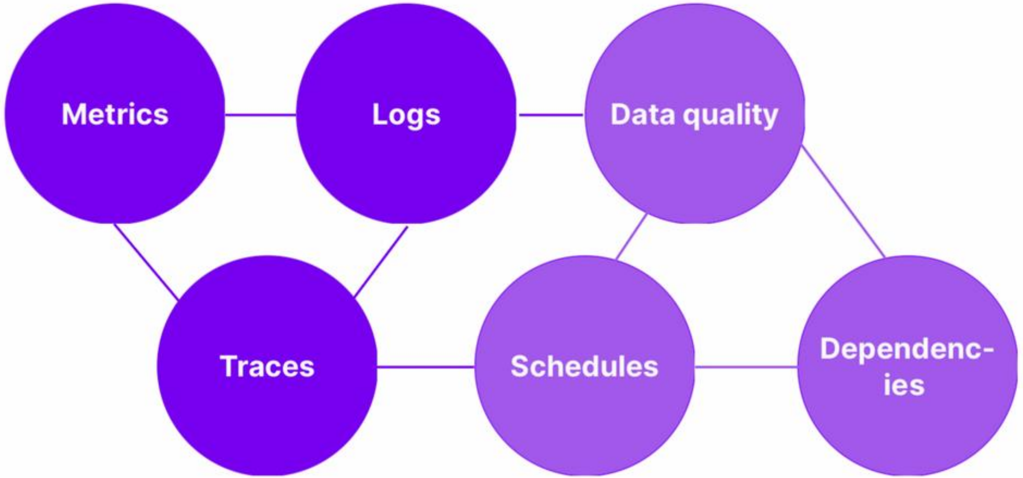


Figure 2: Components of infrastructure

Reference: https://morioh.com/p/1329107ceb96

Data pipeline monitoring



Reference: https://www.castordoc.com/blog/data-monitoring-and-observability

Challenges with model monitoring

1. Who **owns** the model after deployment? Software engineer, data scientist, machine learning engineer.

2. Performance Measurement: Architecting **ground truth** data. Bias, subjectivity (experts opinion), measurement errors are threats.

3. When do we **retrain** the model? Time-based or continuous training.

Ref: Gartner

Reference:
https://appen.com/blog/ai-model-maintenance-guide-to-managing-model/

Maintenance guide for models in production

1. ML model retraining to avoid **model drift.**
2. **Log** both training data, and serving data against contamination effects.
3. Using model performance metrics, **observe models misbehavior** during retraining
4. Identify best practices to minimize **bias** and adopt **fairness** in ML solution
5. Ensure the team layout encourages **innovation** and iteration of process.
6. After the first deployment, design and implement a training **pipeline**.
7. Feedback from model users.

Reference:
https://eugeneyan.com/writing/practical-guide-to-maintaining-machine-learning/

Ref: Gartner

RECAP

- ❑ Overview of monitoring
- ❑ Categories of monitoring
- ❑ Challenges with model monitoring
- ❑ Maintenance guide for model updating

PROJECT COMPLETION

1) Install all needed libraries

=> pip install mlfow

=> pip install neptune-notebooks

=> jupyter nbextension enable --py neptune-notebooks

=> pip install neptune-client[sklearn]

If 'neptune-client[sklearn] installs successfully, please ignore the next line below.

=> pip install neptune-sklearn

2) Go to neptune, and create an account using your email address. After that, create a project and then copy your api and project directory from neptune.

3) Replace the old api and project directory with the new one from Neptune, on the class notebook.

4) Run the cells but when you get to mlfow, run the command below; before you run any cell under the mlflow:

mlflow server \ --backend-store-uri sqlite:///mlflow.db \ --default-artifact-root ./artifacts \ --host 0.0.0.0

PROJECT COMPLETION

5) After that, go to your browser and enter the following link: http://127.0.0.1:5000/

This will bring up the mlflow dashboard.

6) When you are through with experiment logging, take screenshots of your metadata on neptune and mlflow.

7) Then export your models.

8) Select any of the models and create either flaskapi or fastapi or both and import the ml model.

Note: ensure you install sklearn on the virtual environment[pip install scikit-learn == <version on your notebook>

Install joblib on the virtual environment: pip install joblib.

9) Test app locally on postman and then, deploy to heroku, and test via postman again. Take a screenshot and share your result via github link.

10) If you want to explore further, you can use streamlit or html to create a frontend that will interact with the api but its **optional.**

REFERENCE

- https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide


Q&A