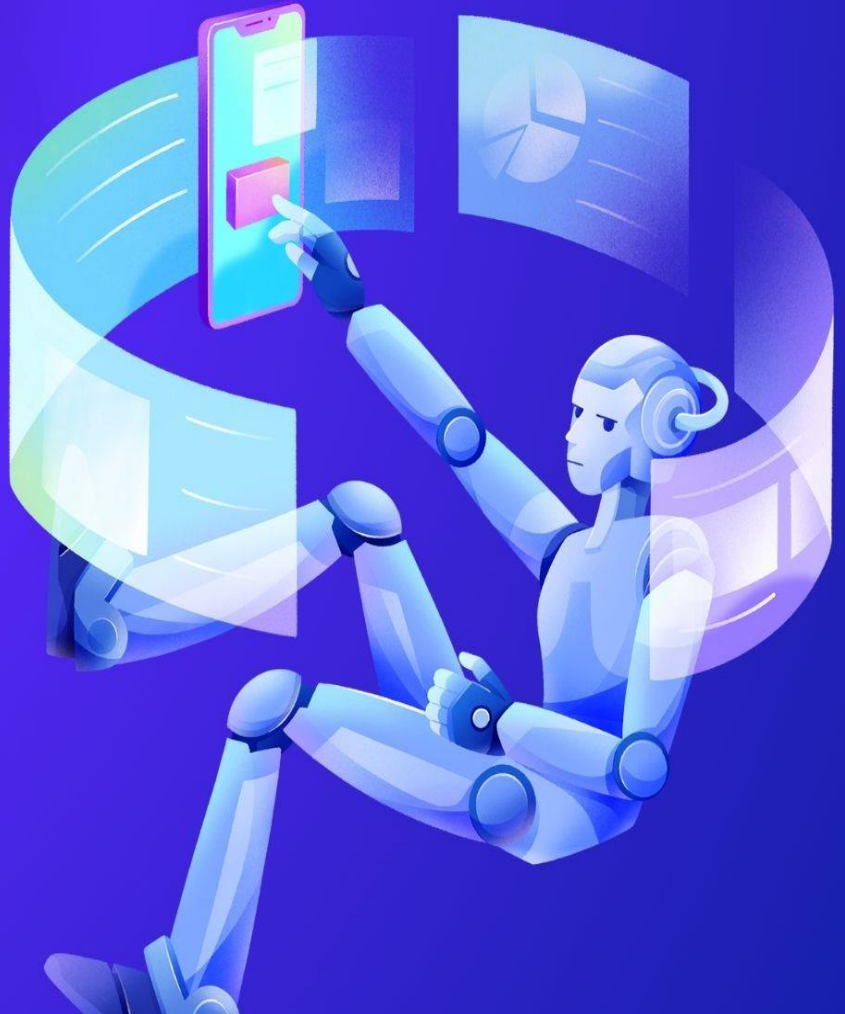
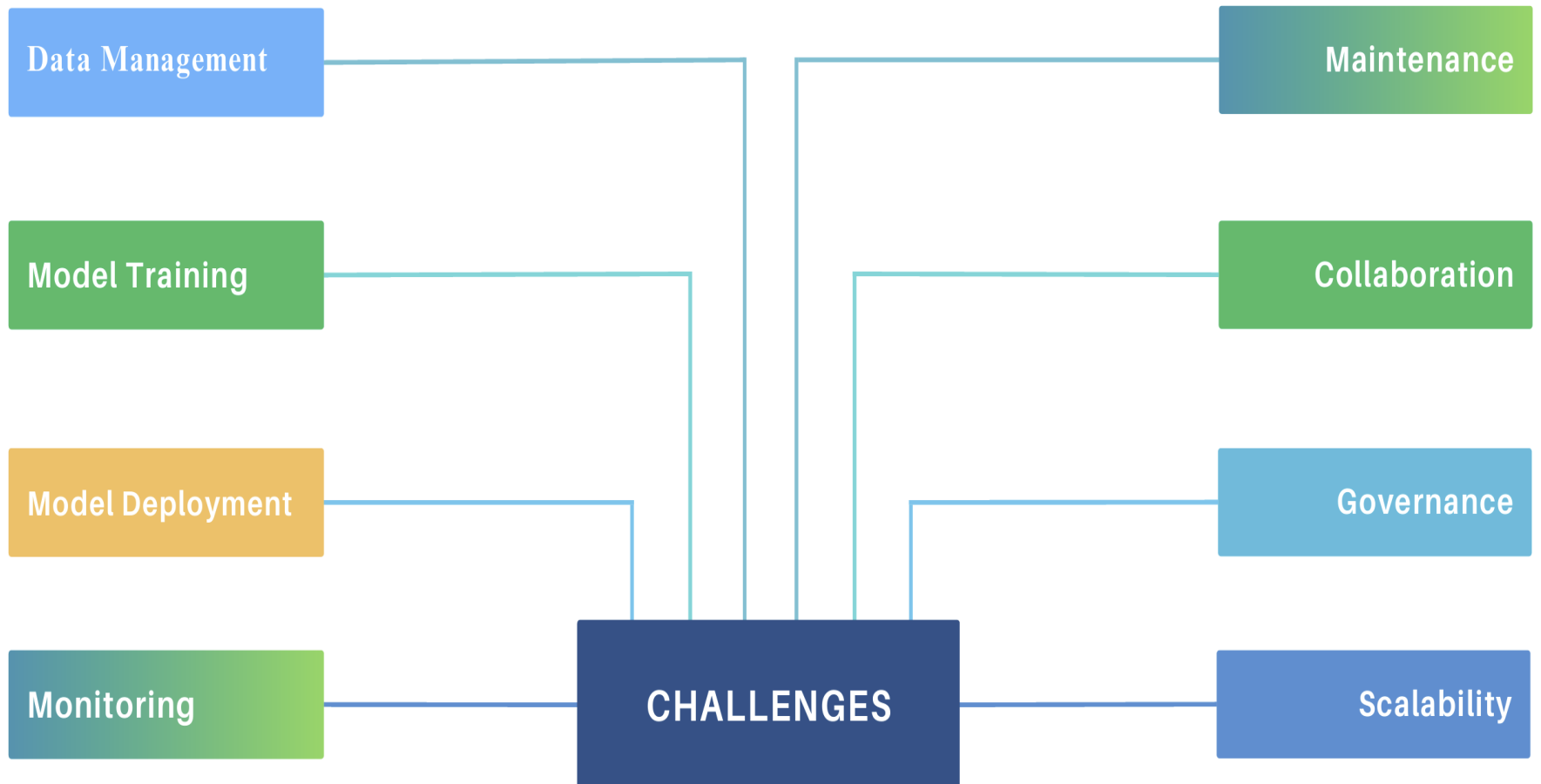


MLOPS



Challenges in ML Lifecycle



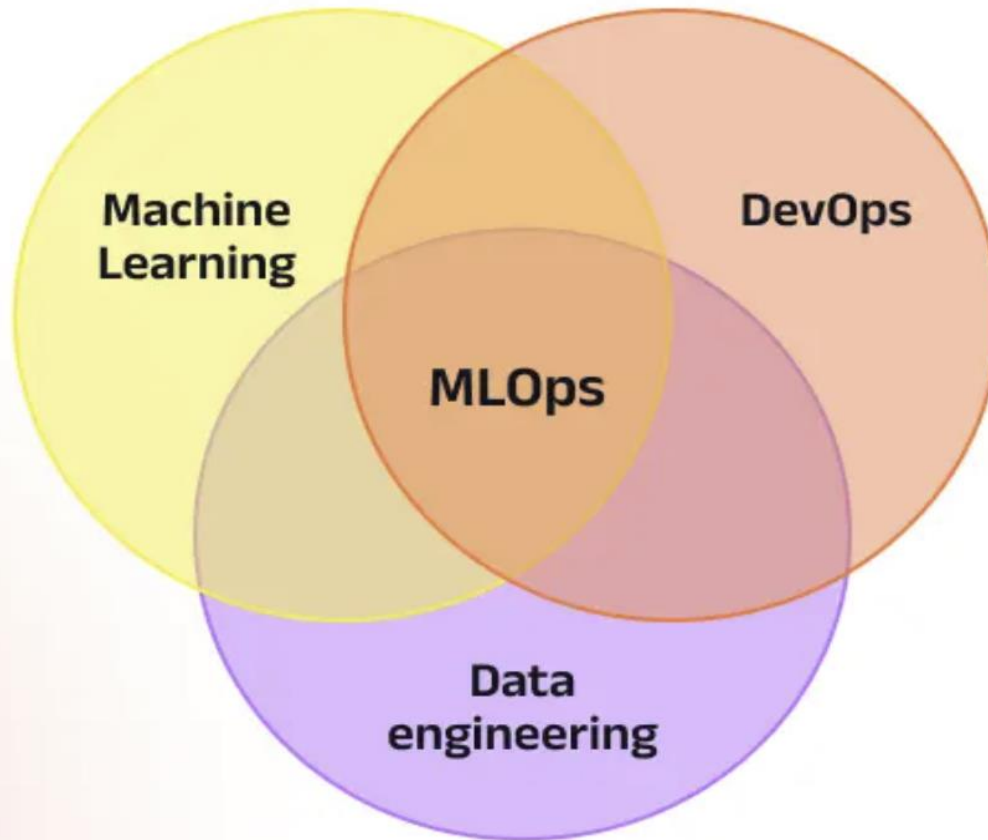
Tools and Techniques for MLOps

Tool / Platform	Purpose
Git	Version control for code.
DVC (Data Version Control)	Version control for datasets.
MLflow	Model versioning and managing the ML lifecycle.
Docker	Containerization of ML models and environments.
Kubernetes	Orchestration of containerized applications.
Jenkins / GitLab CI	Automating the testing and deployment of ML models.
Prometheus / Grafana	Monitoring model performance and tracking data drift.
Apache Airflow	Orchestrating and scheduling complex ML workflows.
Kubeflow	Managing and deploying ML workflows on Kubernetes.
Weights & Biases	Experiment tracking and management.
AWS / Google Cloud / Azure	Cloud-based services for ML operations and hosting.

Agenda

- MLOps Vs DevOps
- ML Engineer Hierarchy
- Mlflow
- Components of Mlflow
- Environment Setup
- Lab 1-Using MLflow to track CPU utilization and memory usage(System Monitoring)
- Lab 2:Track scikit-learn model training with MLflow
- Lab 3:Track Keras model training with MLflow

MLOps Vs DevOps



DevOps

- Focuses on the Software development lifecycle (SDLC).
- Involves continuous integration and continuous deployment (CI/CD) of software applications



Jenkins

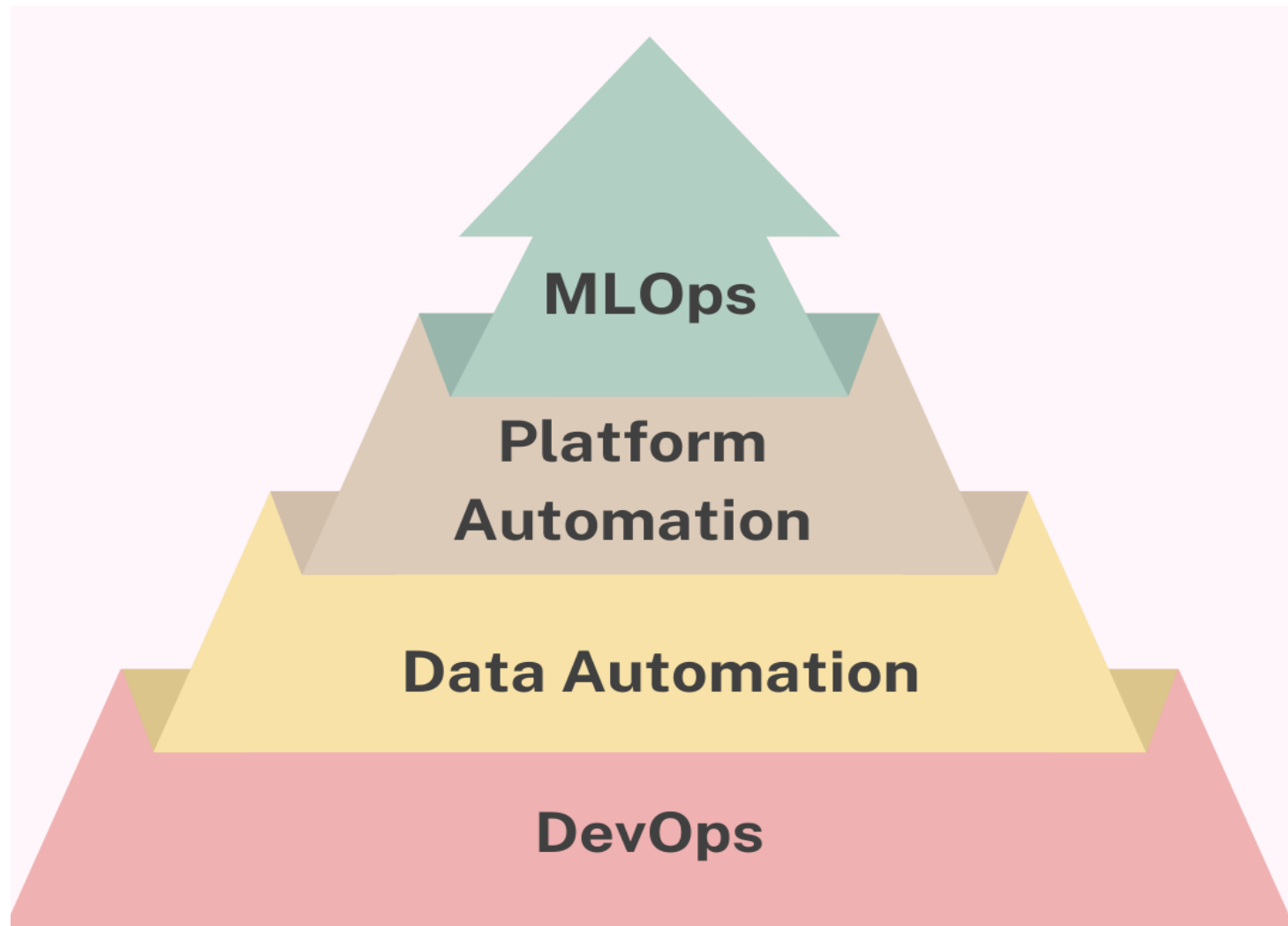


MLOps

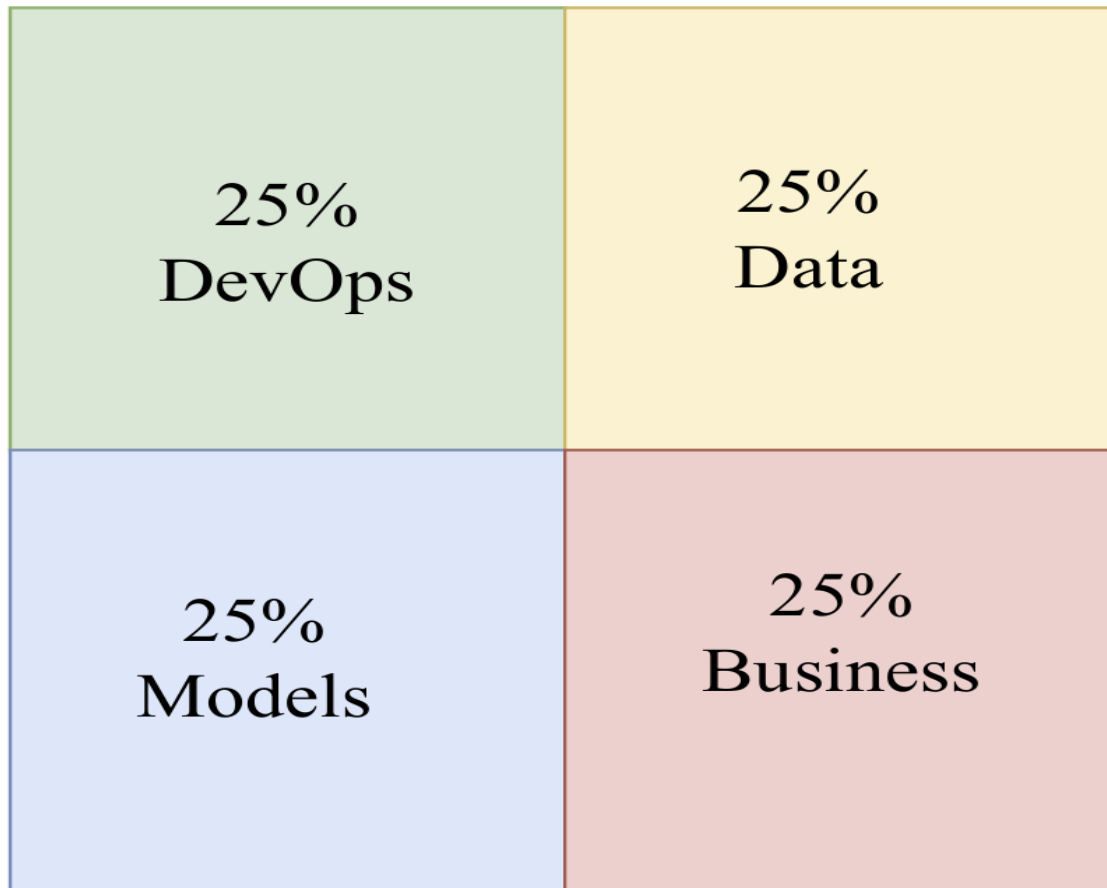
- Focuses on the Machine learning lifecycle, including data collection, model training, validation, deployment, and monitoring



ML Engineer Hierarchy



Rule of 25%



The logo features the text "mlflow" in a sans-serif font, with "ml" in black and "flow" in blue. A small "TM" trademark symbol is positioned to the upper right of the "w". The text is centered within a white circle, which is surrounded by a thick, glowing ring with a pink-to-orange gradient.

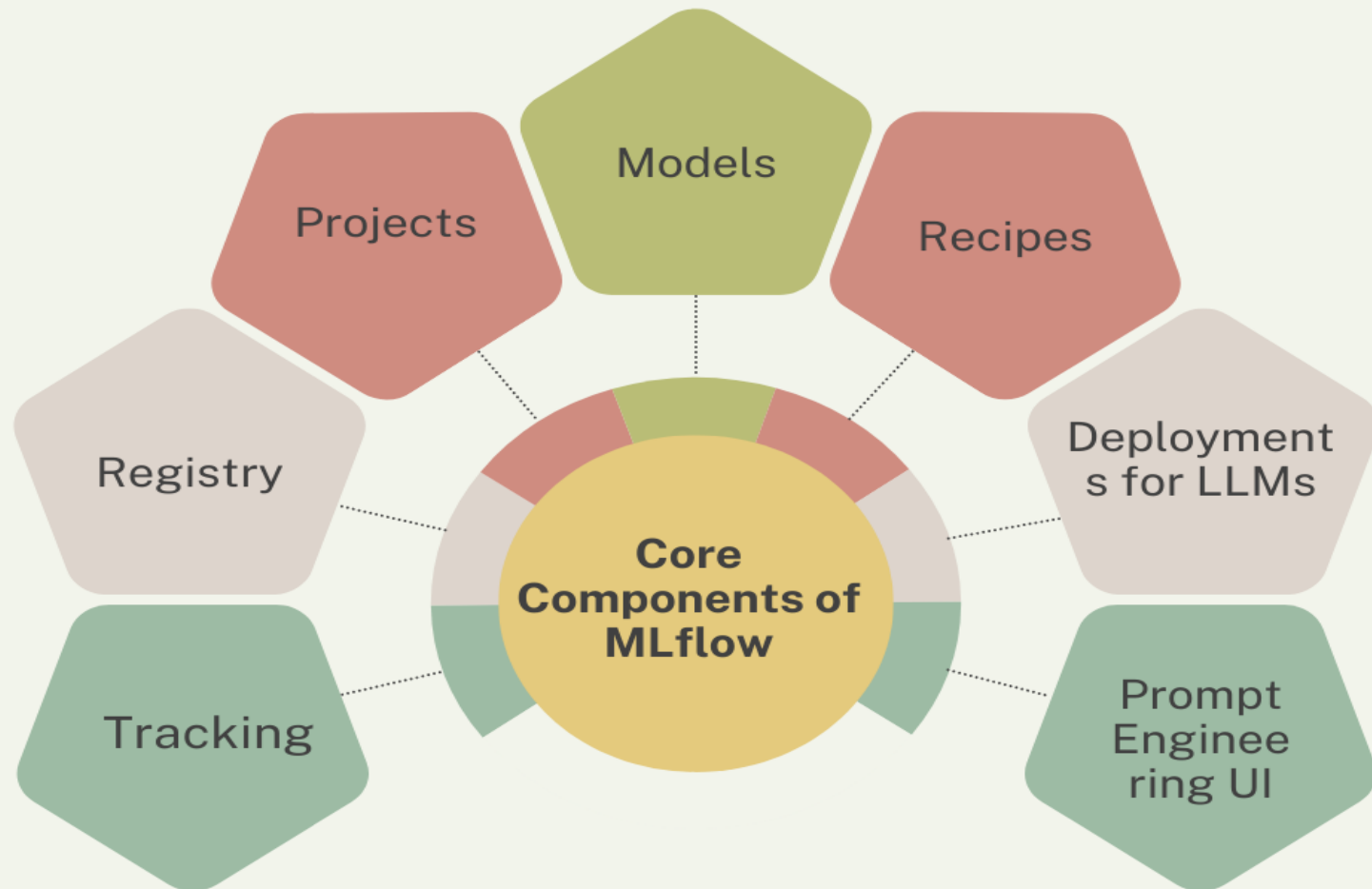
mlflow™

MLflow

MLflow is an open-source platform designed to manage machine learning workflows, widely utilized by MLOps teams and data scientists.

It provides tools for experiment tracking, model management, and deployment

Core Components of MLflow



Key Components of MLflow



Tracking

logs and
queries
experiments



Projects

standardize
packaging
and running
code



Models

standardize
packaging and
deploying
models



Registry

manages the
lifecycle of
models

MLflow Tracking

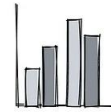
Data



Models



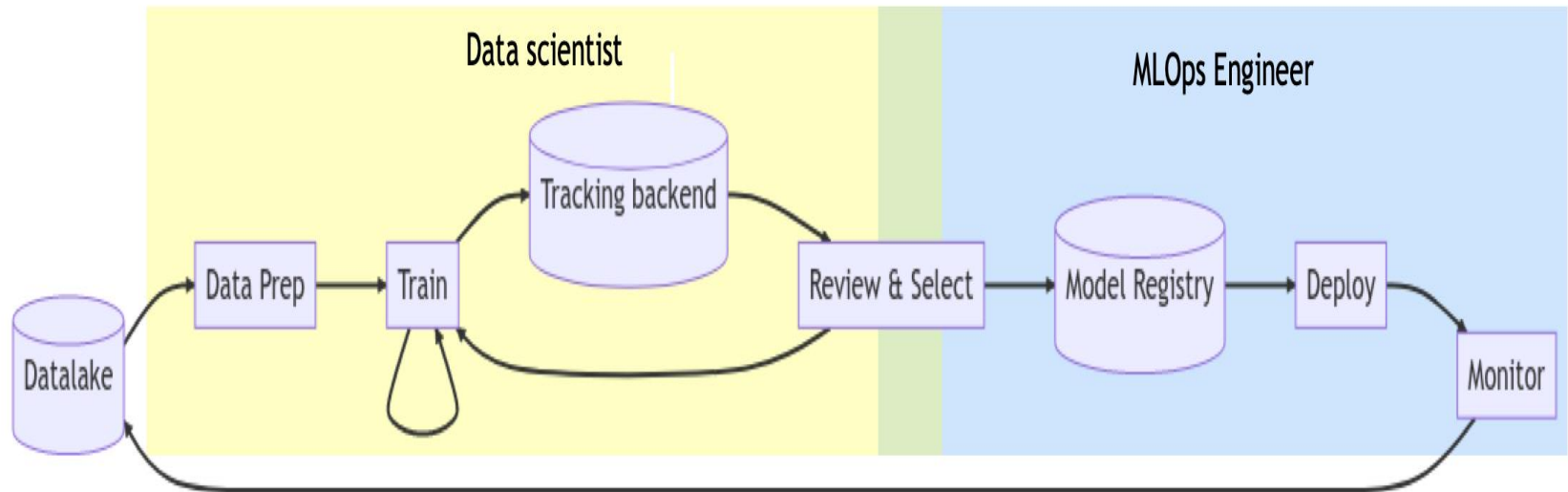
Metrics



model.pkl.dvc

- MLflow Tracking allows data scientists to work with experiments.
- MLflow Tracking is built around runs, that is, executions of code for a data science task.

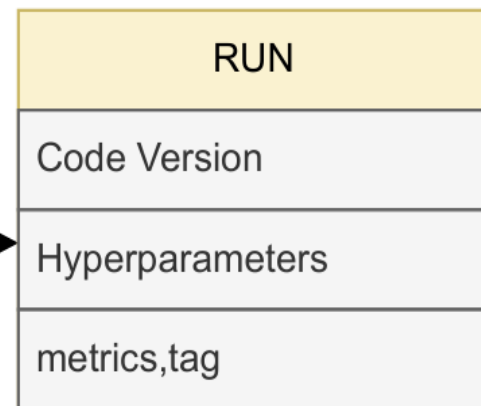
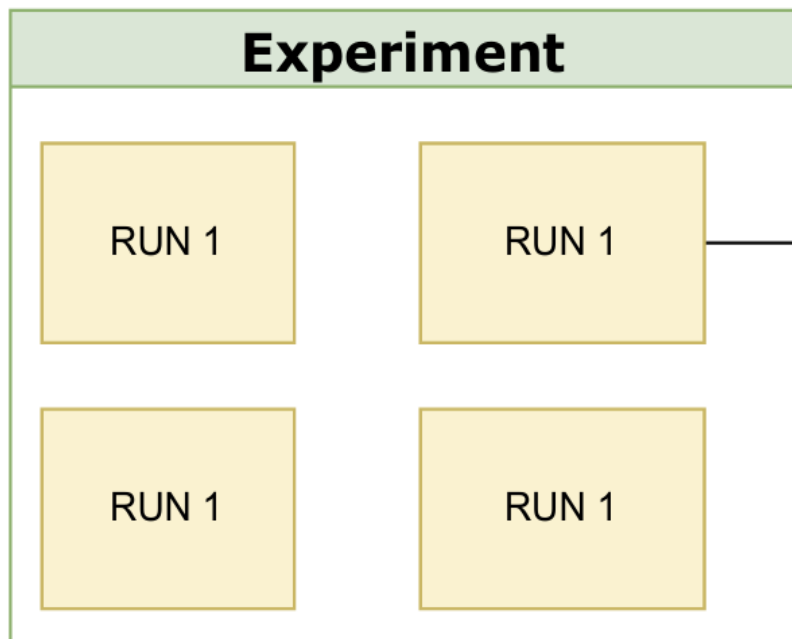
MLflow Tracking



Experiment Tracking Capabilities

- Logging and Querying Experiments
- UI for Visualization
- Centralized Server
- APIs and Libraries Support





Environment Setup

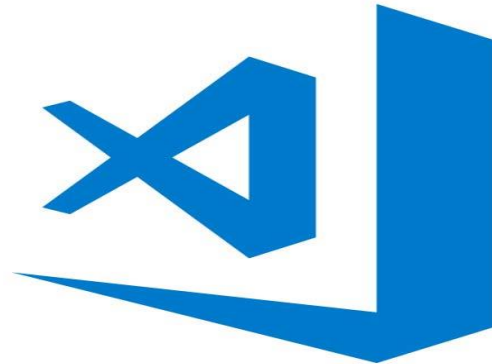


Download & Install

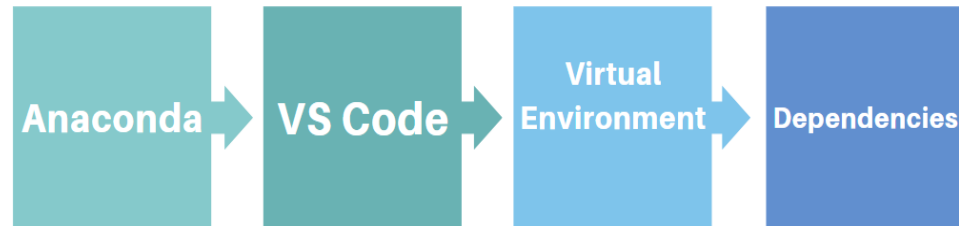


Anaconda

[Installing Anaconda on Windows \(youtube.com\)](#)



Visual Studio Code



Virtual Environment

```
conda create --name mlops python=3.11
```

Dependencies

```
pip install -r requirements.txt
```

mlflow ui

mlflow 2.13.0

Experiments

Models



GitHub

Docs

Experiments



Search Experiments



Default



Default



[Provide Feedback](#)



[Add Description](#)

Share

Q metrics.rmse < 1 and params.model = "tree"



Time created ▾



+ New run

State: Active ▾

Datasets ▾

Sort: Created ▾

Columns ▾

Group by ▾

Table

Chart

Evaluation

Experimental



Run Name

Created



Duration

Source

Models



No runs logged

No runs have been logged yet. [Learn more](#) about how to create ML model training runs in this experiment.

Logging functions

Function Name	Purpose
<code>mlflow.set_experiment</code>	Sets the name of the experiment under which to log runs (optional).
<code>mlflow.start_run</code>	Starts a new MLflow run.
<code>mlflow.log_param</code>	Logs a parameter used by the model (e.g., hyperparameters).
<code>mlflow.log_metric</code>	Logs a metric (e.g., RMSE, accuracy) to evaluate the model's performance.
<code>mlflow.log_artifact</code>	Logs an artifact (e.g., a file, plot, or other outputs).
<code>mlflow.sklearn.log_model</code>	Logs a scikit-learn model.

System Monitoring

System Monitoring >

agreeable-moose-413



Overview Model metrics System metrics Artifacts

Registered models

—

Parameters (1)

Search parameters

Parameter	Value
tracking_interval_seconds	10

Metrics (11)

Search metrics

Metric	Value
available_memory	19936251904
bytes_received	1293811388
bytes_sent	173422071
cpu_usage	6.5
disk_usage_percent	34.6
free_disk_space	317869191168
memory_usage_percent	41.4
total_disk_space	486386167808
total_memory	34038341632
used disk space	168516976640