MLOps Engineering
Machine Learning Operations V2.0.0
Sessions 8 - 9

MsC in Business Analytics and Data Science

Madrid, May 2025

Agenda

- Q&A Project phase
- 15’ Quiz
- Towards ML pipeline automation
- WandB and Mlflow 101

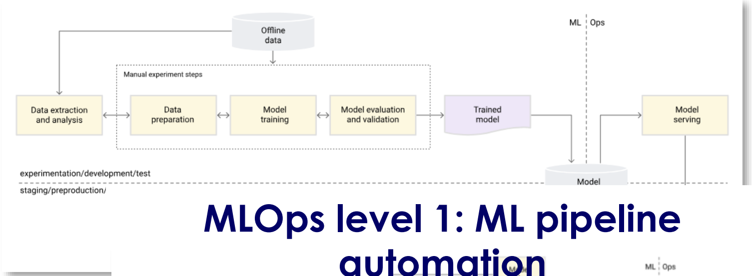
Evaluation methodology v2.0.0

Class Participation: **0.0**
Final exam fail: **3.5/10.0**

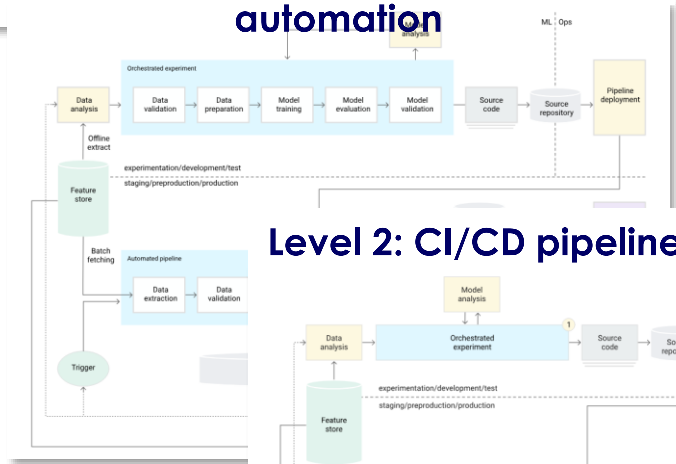
Activity	Weight	Description	(Tentative) Deadline
Ind. Class participation	20%	0 Relevant contributions in class and/ or via Teams	ON GOING
1st Group ~~Presentation~~	15%	1 1st group deliverable (Business case, VC'ed, production-ready-code)	SESSION 8 (29st May)
Intermediate test	10%	2 Initial core concepts & fundamental best practices	SESSION 9 (29th May)
2nd Group Work Presentation	25%	3 Final group project - Presentation (End-to-end CI/ CD)	SESSION 14 (25th Jun)
Ind. Final Exam	30%	4 Final closed-book exam	SESSIONS 15 (25th Jun)
Total	100%		

MLOps is not a destination, but a journey. Involving people, processes, tools, data and governance

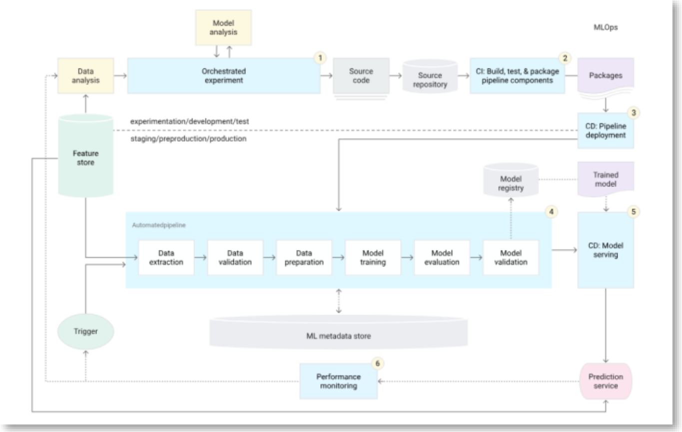
Level 0: Manual process

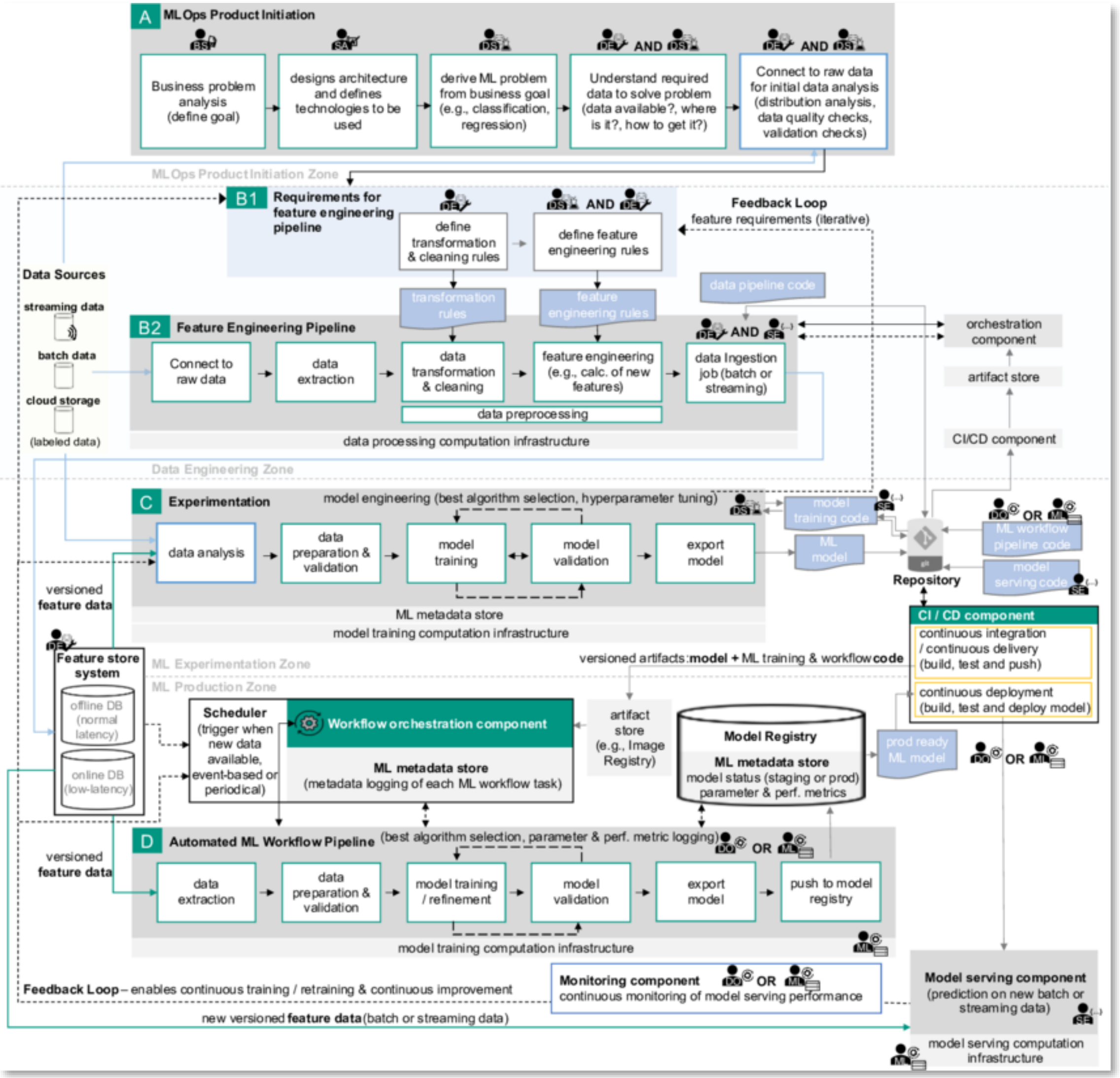


MLOps level 1: ML pipeline automation



Level 2: CI/CD pipeline automation



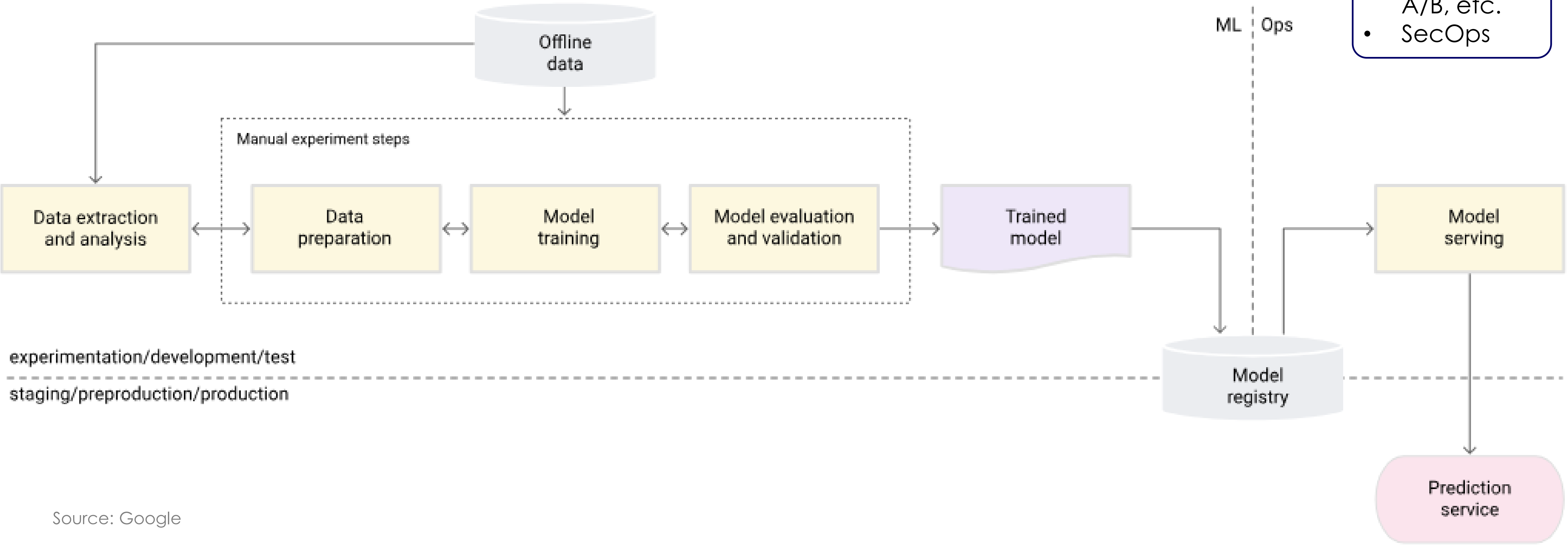


Source: Google

Level 0: Manual ML workflows hinder scalability and reliability, highlighting the need for MLOps automation to streamline operations and reduce errors

Business Problem/ Opportunity definition

- RestAPIs
- Testing suit, Canary, A/B, etc.
- SecOps



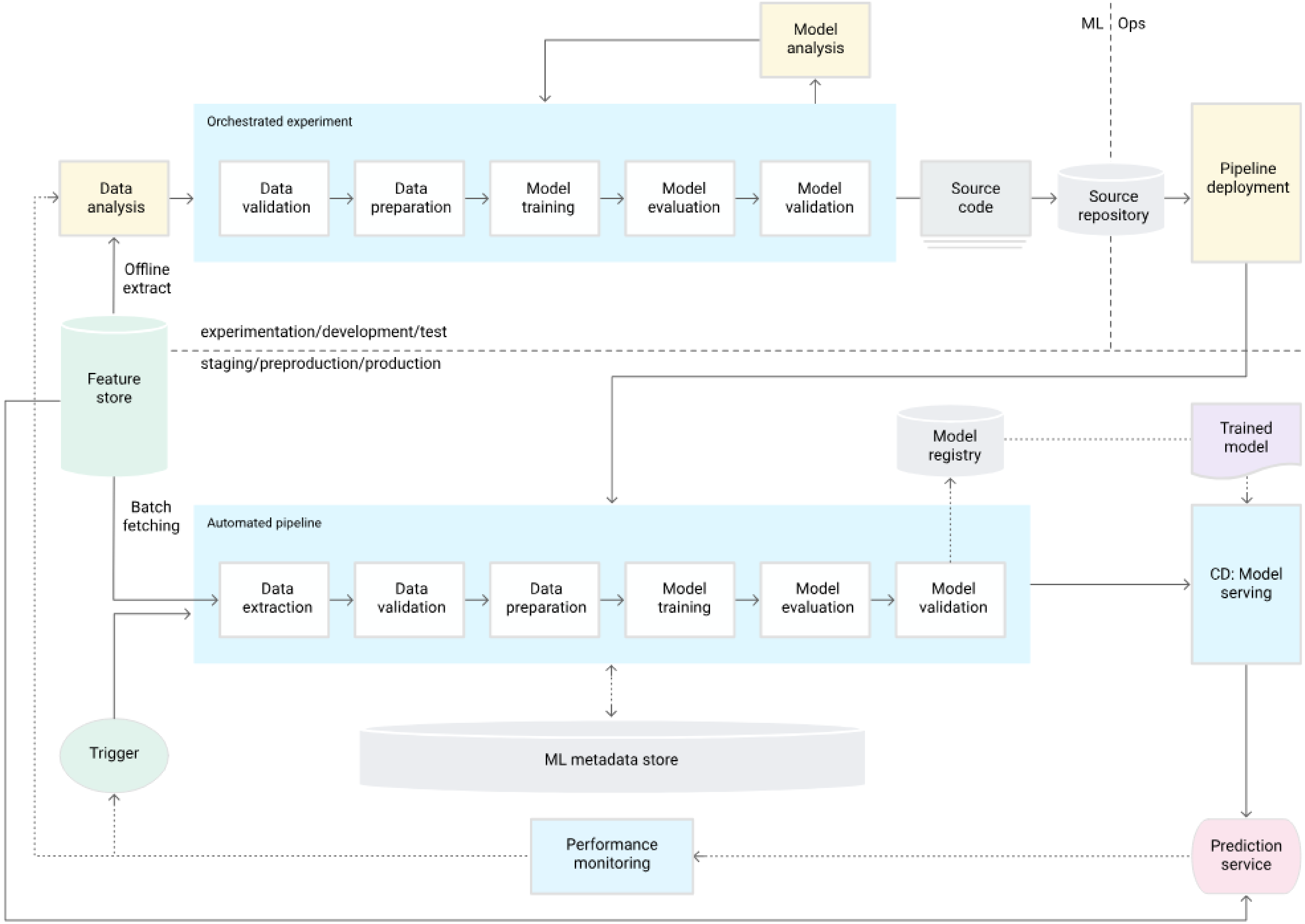
Source: Google

Automating ML pipelines enable fast experiments, continuous model updates, and reliable deployments

- **Rapid Experimentation**
  - **Automate pipeline** steps for quick iterations
  - Seamlessly transition **experiments to production**
- **Continuous Training**
  - **Automatically retrain** models with fresh data
  - **Triggered** by schedules, data availability, or performance
- **Experimental-Operational Symmetry**
  - Use **identical pipeline** in development and production
  - Ensures consistency and **simplifies management**
- **Modular Components**
  - **Decouple** code execution (EDA can still live in notebooks)
  - Reusable, isolated components **enhance reproducibility**
- **Continuous Model Delivery**
  - **Automatically deploy** updated models
  - **Regularly serve** improved prediction services

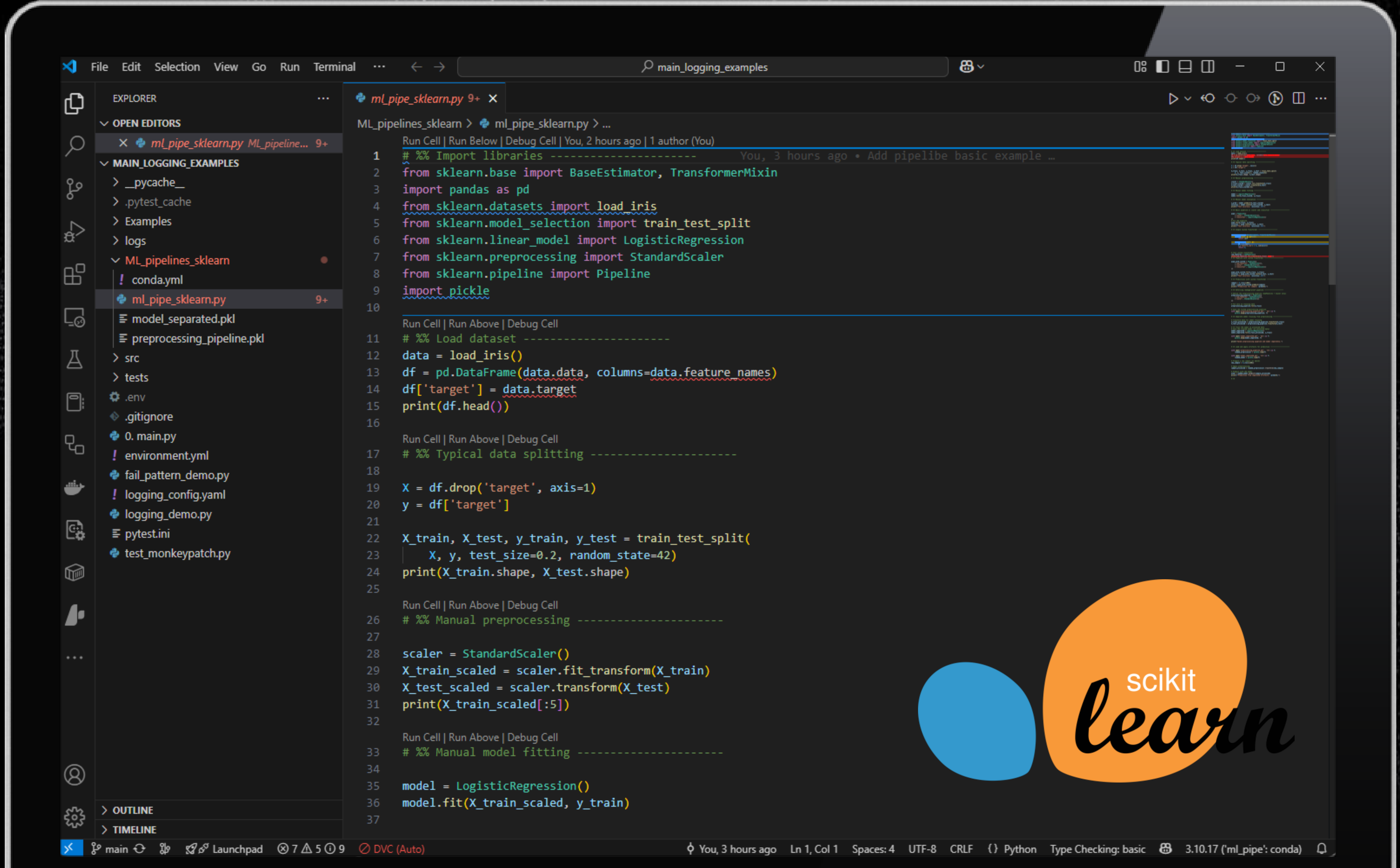
Level 1

Enabling continuous training of the model by automating the ML pipeline



Source: Google

ie UNIVERSITY



https://github.com/2025-IE-MLOps-course/main_logging_examples

Harnessing the combined power of MLflow & WandB for Experimentation and Lifecycle Management

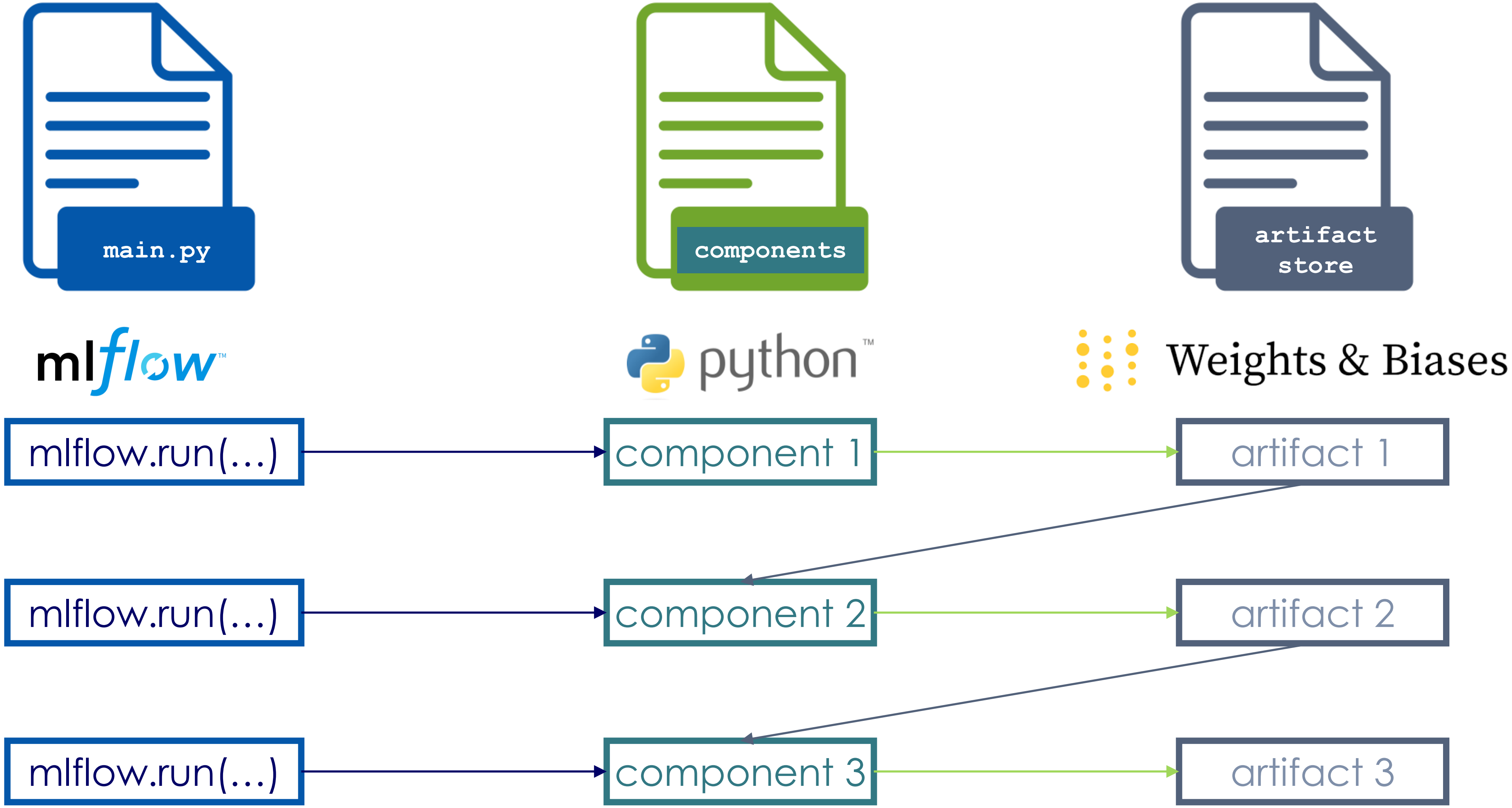
	mlflow	Weights & Biases
Overview	Open-source platform managing the end-to-end ML lifecycle	Experiment tracking, dataset versioning, and model evaluation
Importance	Facilitates reproducibility, collaboration, and model deployment	Real-time insights, collaboration, and reproducibility in ML projects
Components	Tracking, Projects, Models, and Registry	Experiments, Reports, Artifacts, Tables, Sweeps, Launch, Models…
Business model	Open-sourced	Managed Cloud (Free student)

Source: mlFlow, Weight & Biases

Integrating MLflow and WandB ensures your work is trackable, reproducible, and scalable

Dimension	Current (main.py only)	With MLflow	With WandB
Experiment Tracking	Manual, limited, error-prone	Automated, centralized, queryable	Best-in-class, collaborative, real-time
Pipeline Automation	Custom scripting, no standardization	Standardized, modular, reproducible	Still manual, focus on tracking, not flow
Monitoring	Basic logs, hard to compare runs	UI for metrics, basic monitoring	Advanced live metrics, alerts, dashboards
Visualization	Print/logs, no central dashboard	Simple UI, basic charts	Rich dashboards, interactive comparisons
Reproducibility	Depends on discipline, not enforced	Enforced via MLproject, Conda/Docker	Good with artifacts and configs
Collaboration	Manual sharing, hard to track changes	Easier, but limited UI	Team-focused, cloud-based collaboration
Model Registry	Manual versioning, error-prone	Built-in, production-ready registry	Artifacts system, suitable for most projects

MLflow and WandB integration



Source: based on Udacity

MLflow and WandB integration

mlflow python

HYDRA

Argparse()

main.py

- MLproject
- conda.yml
- config.yalm
- main.py

Component 1

- MLproject
- conda.yml
- run.py

Component 2

- MLproject
- conda.yml
- run.py

Remote WandB Repository

Weights & Biases

Canonical Mlflow directory

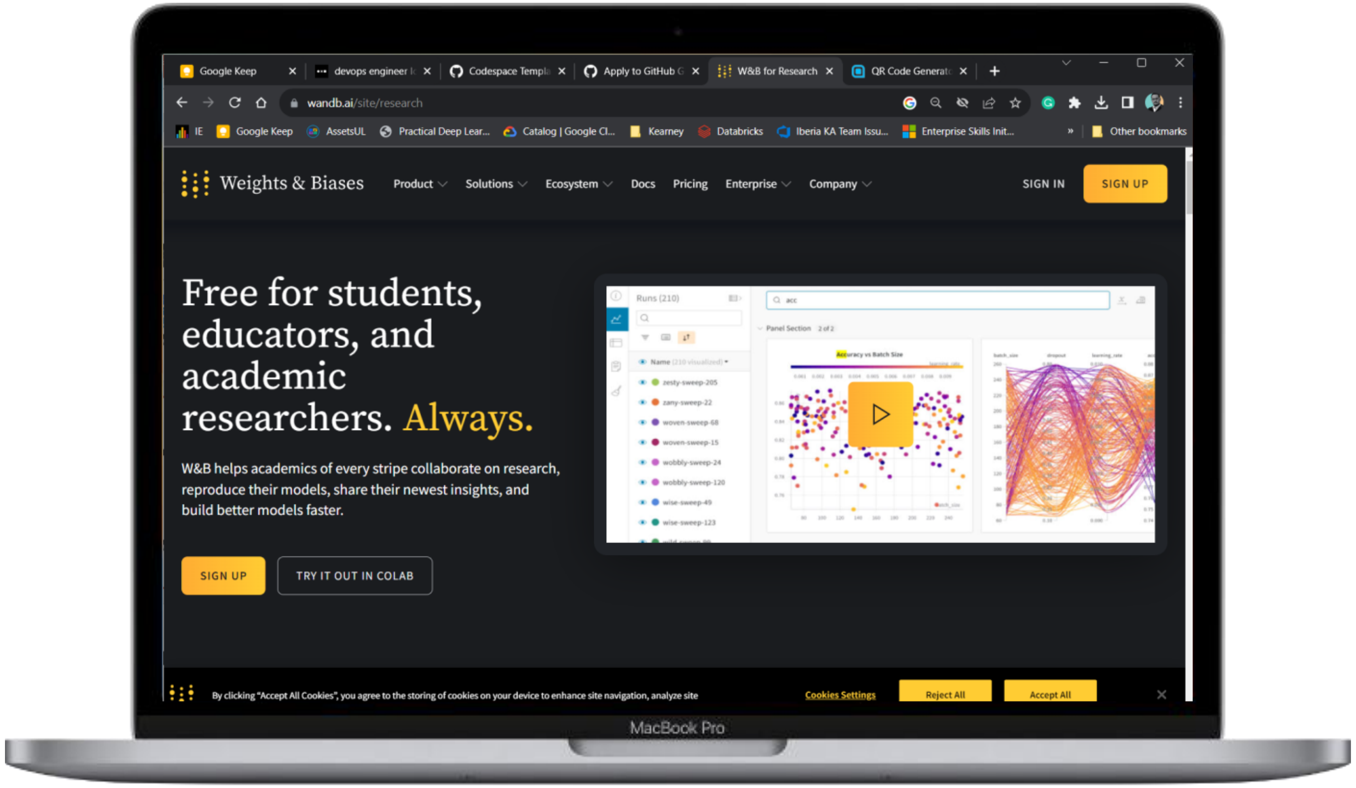
```
.
├── MLproject
├── main.py
├── environemnt.yml
├── conda.yml
└── src
    ├── basic_cleaning
    │   ├── MLproject
    │   ├── conda.yml
    │   └── run.py
    ├── data_check
    │   ├── MLproject
    │   ├── conda.yml
    │   └── run.py
    ├── eda
    │   ├── EDA.ipynb
    │   ├── MLproject
    │   └── conda.yml
    └── train_random_forest
        ├── MLproject
        ├── conda.yml
        └── run.py
```

Next project steps (25/06)

1. Refactor pipeline to use **MLflow and W&B**
2. Add/configure **Hydra** for flexible experimentation
3. Create and push tests; see CI results on **GitHub Actions**
4. Deploy API to <free server>; share the endpoint
5. Wrap model in **FastAPI**; test endpoint locally

Weights and Biases

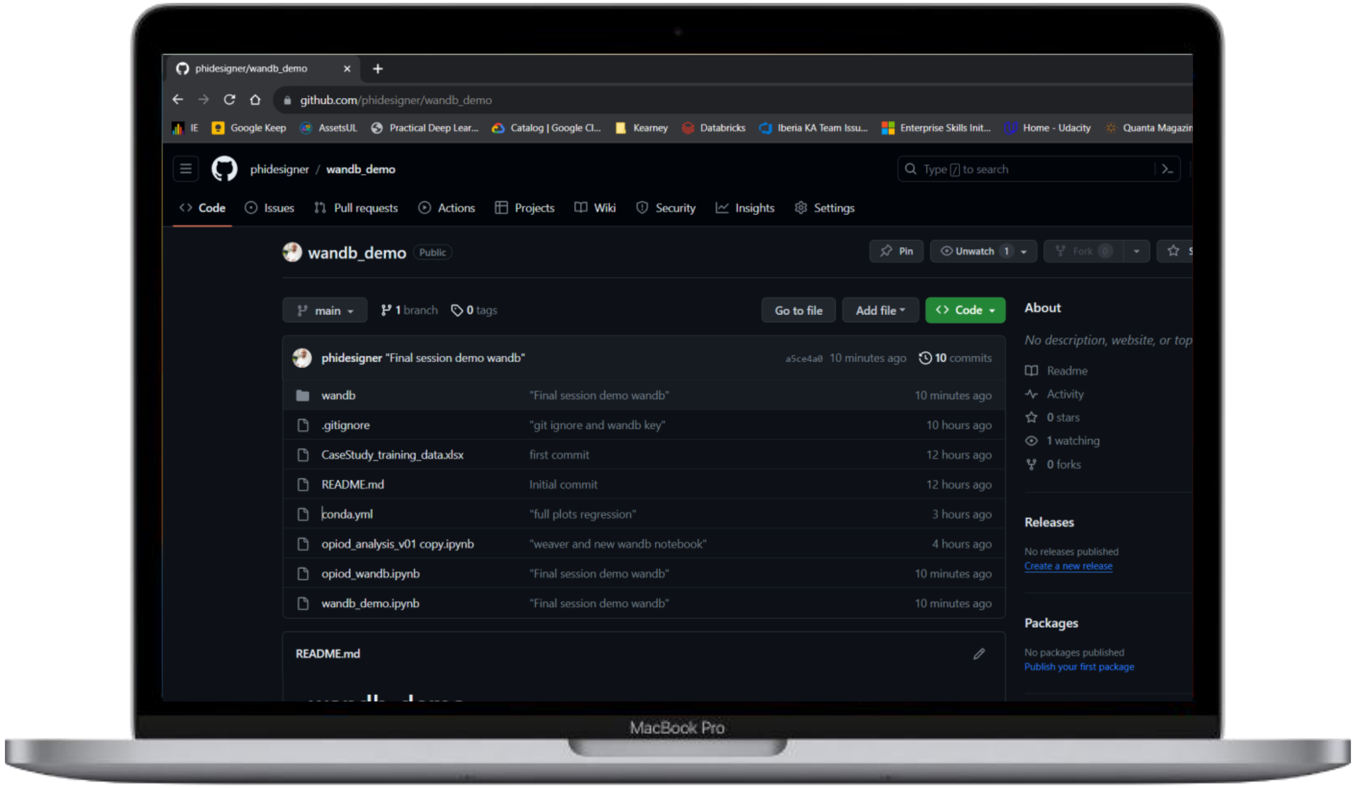
SCAN ME



WandB

GitHub repo – WandB demo

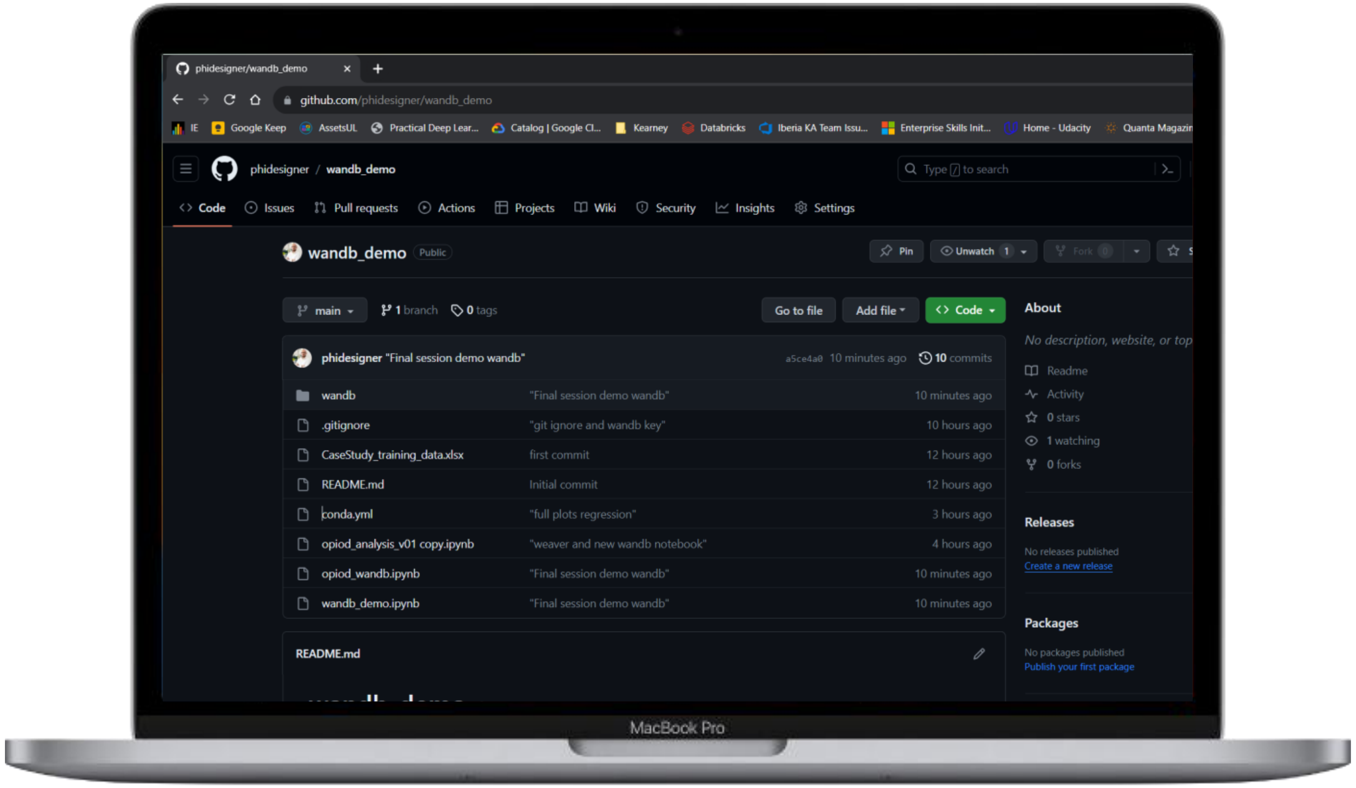
Weights & Biases



[GitHub repo]

GitHub repo – MLflow demo

mlflow™



[GitHub repo]

Mlflow relevant commands

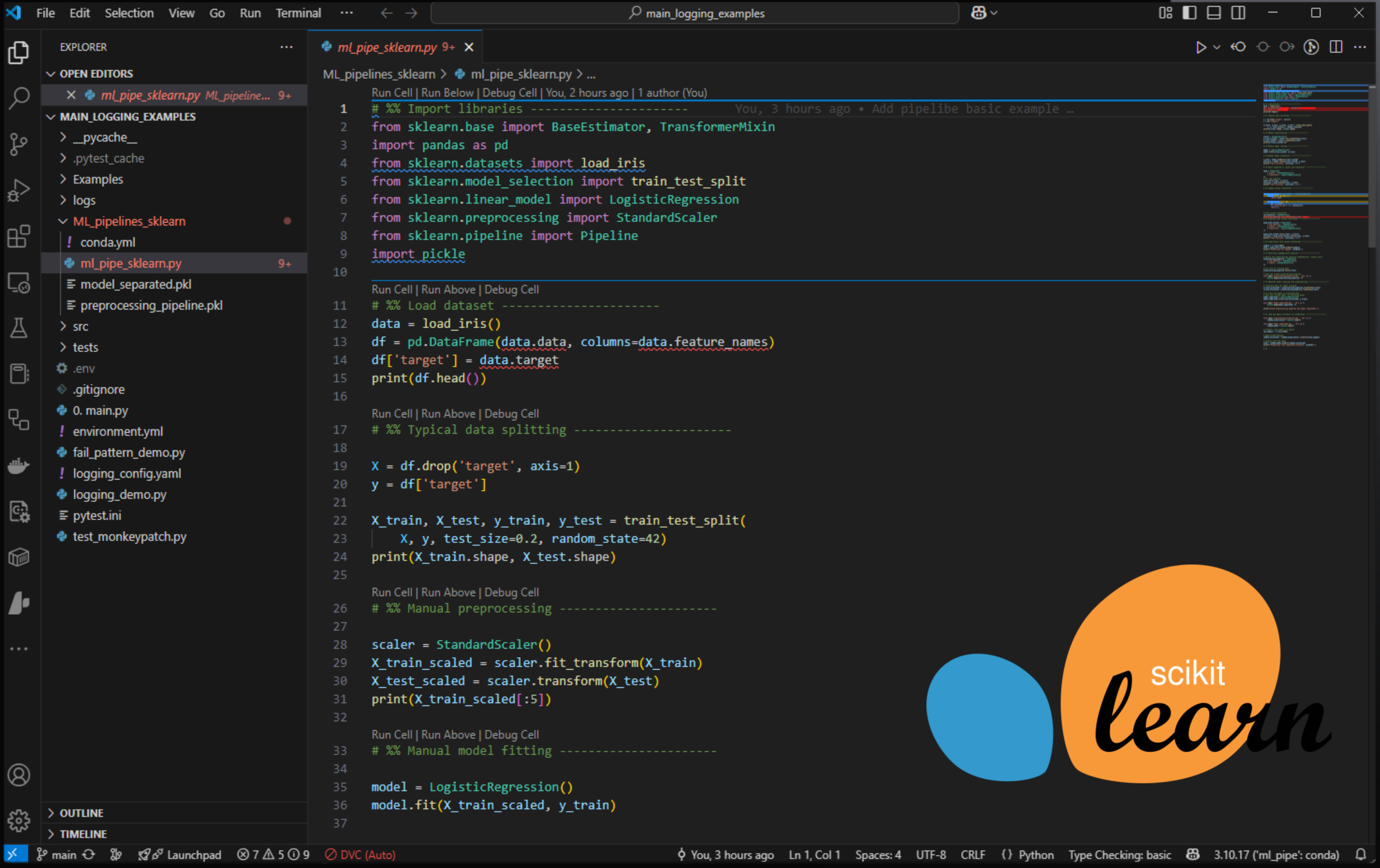
```
- conda clean -all
- conda env remove -n mlflow-<hash>
- mlflow run .
- mlflow run . -P <arg name>=<“script”>
- mlflow run src/<module>
- mlflow ui
```

Mlflow commands in WSL

 # Install Ubuntu

> wsl --install

- # Install conda
- wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
- bash Miniconda3-latest-Linux-x86_64.sh
- # Install MAMBA
- conda install -n base -c conda-forge mamba
- conda config --set solver libmamba
- # Set MAMBA as solver
- echo 'export MLFLOW_CONDA_CREATE_ENV_CMD=mamba' >> ~/.bashrc
- source ~/.bashrc



https://github.com/2025-IE-MLOps-course/main_logging_examples