✅ Project 1: End-to-End MLOps Pipeline (Local Batch Training Project)

📅 Project Week Plan: Tuesday to Friday

🧠 Purpose: Simulate a real-world MLOps workflow using only your local system and available tools. No cloud infra or Kubernetes server needed — we’re focusing on logic, architecture, reproducibility, and tracking.

# 🎯 Goal of the Project

By the end of this week, you should be able to:  
- Build a reproducible batch ML training pipeline  
- Explore data and track preprocessing steps  
- Train and version a model using Kubeflow locally  
- Track metrics and artifacts using MLflow  
- Store models on MinIO  
- Setup Prometheus + Grafana to monitor metrics locally  
- Create a logic for retraining trigger when new data is added  
- Document and present everything as if this was a real industry project

# 🧰 Tools & Tech Stack (All Local)

|  |  |
| --- | --- |
| Area | Tool |
| Data | Azure SQL DB (local connector) |
| Pipeline Orchestration | Kubeflow Pipelines (locally via k3s/minikube/docker-desktop or kfp sdk) |
| Version Control | Git |
| Data Versioning | DVC |
| Experiment Tracking | MLflow |
| Artifact Store | MinIO (local) |
| Monitoring | Prometheus & Grafana (local setup) |
| Command Line | kubectl (if needed for Kubeflow pipelines) |

# 📆 Project Plan (Tuesday to Friday)

## 🟩 Tuesday – Dataset, EDA & Preprocessing Setup

* Tasks:
* Connect to Azure SQL DB (or simulated local CSV dump)
* Perform EDA and document insights
* Preprocess the data for model input
* Set up DVC to version the raw and preprocessed data
* Initialize a GitHub repository and push your progress
* Expected by EOD:
* data/ and preprocessed/ folders versioned with DVC
* Preprocessing script or notebook
* Initial commit on GitHub repo
* Notes on Azure DB schema or data source

## 🟦 Wednesday – Batch Training Pipeline + MLflow

* Tasks:
* Use the given model code
* Create a Kubeflow batch pipeline to train the model
* Integrate MLflow to track metrics and params
* Save the model artifacts to MinIO
* Expected by EOD:
* pipeline.py or pipeline YAML file
* MLflow tracking UI showing metrics
* Screenshot of model file stored in MinIO
* A diagram or explanation of your pipeline flow

## 🟨 Thursday – Monitoring and Retraining Logic

* Tasks:
* Simulate metric scraping via Prometheus
* Visualize model accuracy or loss on Grafana
* Design logic for retraining when new data is pushed (can be simulated)
* Update pipeline to handle full retraining upon trigger
* Expected by EOD:
* Prometheus scraping config + Grafana dashboard screenshot
* Notebook/script that checks for data change and triggers pipeline
* Updated pipeline that saves new metrics to MLflow

## 🟥 Friday – Wrap-up & Documentation

* Tasks:
* Finalize your GitHub repo (add README, organize folders)
* Write a technical documentation (Markdown or Google Doc)
* Prepare a 5-6 slide presentation
* Write a short LinkedIn post about the project
* Expected by EOD:
* Public GitHub link or add instructor as a collaborator
* Technical doc covering architecture, tools used, and challenges
* PPT explaining project overview and workflow
* LinkedIn post draft (even if you don’t post it)

# 📘 Submission Format (By Sunday Night)

* ✅ GitHub Repository  
  - Include folders: data/, scripts/, pipeline/, monitoring/, docs/  
  - Add instructor as a contributor if not public
* 📄 Technical Documentation (Markdown or Google Doc)  
  - Tools used  
  - Data pipeline and preprocessing  
  - Kubeflow batch training  
  - MLflow + MinIO integration  
  - Monitoring setup  
  - Retraining logic  
  - Screenshots of each step
* 📊 Presentation (5-6 slides)  
  - Project goal & setup  
  - Architecture overview  
  - Tools used  
  - Pipeline overview with visuals  
  - Monitoring and retraining logic  
  - Learnings
* 💬 LinkedIn Post (optional but recommended)  
  - Share your experience  
  - Include screenshots, learnings, and hashtags  
  - Tag instructor and batchmates

# 🤝 Notes from Instructor

- This is not an evaluation – it's an opportunity to explore what MLOps looks like in real-world projects  
- Creative workflows are encouraged — feel free to extend the architecture  
- If you face any blockers, reach out early  
- Focus on clarity, modularity, and reproducibility