Senior MLOps Engineer – Take-Home Assessment

Dataset

Telco Customer Churn

Objective

You've just joined a mid-sized telecom company where multiple data scientists are building ML and GenAI models - but without consistent infrastructure. Your job is to start shaping a **minimal MLOps foundation** to help bring order and scalability.

In this assignment, you'll **focus on one core MLOps component** (e.g. a Model Registry or Feature Store) and **mock or describe the others**.

Scenario

The company needs a way to:

- Register and version models
- Reuse features
- Monitor model health in production
- Retrain and redeploy models automatically

You'll build a **minimal but functional version** of **one** of these components and sketch out how the others would work.

Your Task

In the attached Jupyter notebook **mlops_take_home_assignment.ipynb** you will find a basic churn prediction model for the given dataset. Your task is to professionalize this setup using MLOps best practices. Pick **ONE** of the following components (**Model Registry or Feature Store**) to implement in code (your choice) and **mock other components,** just include brief placeholder implementations or markdown descriptions to show how you'd approach them in a full setup:

1. **Model Registry**
   Use MLflow or your own versioning mechanism
   Log: model version, training metrics, parameters, timestamp
   Register at least one trained model (any scikit-learn model is fine)

2. **Feature Store**
   Define and persist a few reusable features
   Use lightweight storage (e.g. DuckDB/SQLite) or open-source tools (e.g. Feast)
   Provide CLI/script to generate features for both training and inference
3. **Monitoring**
   Simulate basic monitoring using dummy or replayed inference data
   Track: inference volume, latency, simple performance metrics or drift
4. **Orchestration and Deployment**
   Build a minimal retraining pipeline using Airflow or Python scripts
   Retrain model on updated data, register a new version
5. **Deployment**
   Implement a CI/CD pipeline for containerized inference of the model, using e.g. docker, docker-compose, kubernetes, github actions, gitlab ci/cd...

Deliverables

Submit a GitHub repo or ZIP file with:

- Your code (for the one component you implemented)
- A README.md with:
  - Setup instructions
  - Which component you implemented and why
  - How to run it
  - Short descriptions or pseudocode for the other components

Time Estimate

~2 hours. Focus on writing **clean, maintainable code** for one component. No need for perfect coverage or polishing—this is about demonstrating your architectural thinking and hands-on skills.