## Project Title: Churn Prediction API with MLOps Integration

### 🌟 Project Overview

This mini-project simulates an end-to-end MLOps pipeline for a practical business case: predicting customer churn. “Churn” refers to when a customer stops using a company’s service, which is a critical business metric in sectors such as telecommunications, SaaS, banking, and insurance.

The project will involve:

1. Building and training a machine learning model to predict churn.
2. Serving the model via a FastAPI-based web API.
3. Logging and monitoring all inference requests and predictions.
4. Automating model retraining using updated data logs.
5. Containerising the application using Docker for portability.

### 🧐 Model Type: Machine Learning vs. Deep Learning

For this project, we will use **machine learning** instead of deep learning. The justification is:

* Churn prediction is typically performed on **structured tabular data**.
* ML models (like **Logistic Regression**, **Random Forest**, or **XGBoost**) offer faster training, easier interpretability, and are more resource-efficient than deep learning.
* DL is generally preferred for unstructured data (e.g., images, text, audio).

**Final choice**: Machine Learning (using scikit-learn or xgboost)

### 📊 Data Description

We will use the **Telco Customer Churn dataset**, a publicly available dataset containing telecom customer usage data and whether they churned.

**Dataset type**: Structured tabular data  
**Target variable**: churn (binary classification)

#### Example Columns:

| Column | Type | Description |
| --- | --- | --- |
| gender | Categorical | Male / Female |
| senior\_citizen | Binary | 0 or 1 |
| tenure | Numeric | Months as a customer |
| monthly\_charges | Numeric | Amount billed monthly |
| contract | Categorical | Type of customer contract |
| churn (target) | Binary | Whether the customer churned (Yes/No) |

### ⚖️ Tools & Technologies

| Task | Tool |
| --- | --- |
| Model Training | Scikit-learn / XGBoost |
| API Service | FastAPI |
| Data Handling | pandas |
| Logging | CSV or SQLite (basic) |
| Automation | Python script / cron scheduler |
| Containerisation | Docker |
| Optional Monitoring | Prometheus + Grafana |

### ✅ Learning Objectives

* Understand the steps in deploying a machine learning model.
* Build and test a RESTful API for model inference.
* Log inference results for monitoring and retraining.
* Implement automated retraining and versioning of models.
* Containerise an ML application for portability and cloud readiness.

### 📂 Suggested Folder Structure

mlops-churn-api/  
|  
├── data/  
│ ├── telco.csv  
│ └── prediction\_logs.csv  
│  
├── model/  
│ └── churn\_model\_v1.pkl  
│  
├── app/  
│ ├── main.py # FastAPI app  
│ ├── train.py # Initial training script  
│ └── retrain.py # Auto-retraining script  
│  
├── Dockerfile  
├── requirements.txt  
└── README.md

Let me know when you’re ready to begin the coding phase or if you’d like help sourcing the dataset or writing the training script.