Capstone Project: End-to-End MLOps Pipeline for Customer Churn Prediction

Project Overview

Goal: Build a fully automated MLOps pipeline to predict telecom customer churn using real-world enterprise workflows including preprocessing, model training, MLflow model registration, Dockerized API deployment on EKS, CI/CD via GitHub Actions, and drift monitoring with Slack alerting.

Dataset Access

Dataset: Telco Customer Churn Dataset – Kaggle
(https://www.kaggle.com/datasets/blastchar/telco-customer-churn)

1. Download the CSV (WA_Fn-UseC_-Telco-Customer-Churn.csv)
2. Place it inside: mlops-churn-capstone/data/raw/

Setup Instructions

```
1. Clone Repository:
    git clone https://github.com/manifoldailearning/mlops-churn-
capstone.git
    cd mlops-churn-capstone

2. Create and Activate Virtual Environment:
    conda create --prefix ./envs python==3.10
    conda activate ./envs
```

3. Install Dependencies:
   pip install -r requirements.txt

Pipeline Breakdown

Preprocessing (src/preprocessing/preprocess.py)
- Clean raw dataset, encode categorical, normalize numerical, split into train/test/val.
- Save inside data/processed/

Training (src/training/train.py)
- Train model, log metrics, and register model to MLflow.

Example MLflow Code:

```
  mlflow.set_experiment("churn-prediction")
  with mlflow.start_run():
    mlflow.log_params(params)
    mlflow.log_metrics(metrics)
    mlflow.sklearn.log_model(model, artifact_path="model",
registered_model_name="ChurnModel")
```

Inference API (src/inference/app.py)
- FastAPI app that loads model from MLflow and serves /predict endpoint.

Dockerization:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY src/inference/ .
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

Push to DockerHub:

```
  docker build -t <your-dockerhub-username>/churn-inference .
  docker push <your-dockerhub-username>/churn-inference
```

Deployment to Kubernetes (EKS)

1. Create EKS Cluster using eksctl
2. Apply manifests: deployment.yaml, service.yaml, hpa.yaml

GitHub Actions for Deployment (.github/workflows/deploy.yml):
name: Deploy to EKS

```
on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          region: us-east-1
      - name: Deploy to EKS
```

```
        run: |
          kubectl apply -f k8s/deployment.yaml
          kubectl apply -f k8s/service.yaml
          kubectl apply -f k8s/hpa.yaml
```

Monitoring & Drift Detection

Use Evidently or custom script to detect drift.

Send Slack Alerts:

```
import requests
def send_slack_alert(message):
    webhook_url = "https://hooks.slack.com/services/..."
    requests.post(webhook_url, json={"text": message})

if drift_detected:
    send_slack_alert("⚠️ Drift Detected in Churn Model! Please
investigate.")
```

Submission Checklist

Deliverables:

- Preprocessing scripts
- Training & Evaluation
- MLflow model registration
- Inference FastAPI app
- Docker image + DockerHub push
- Kubernetes YAMLs (with HPA)
- GitHub Actions for deployment
- Drift detection + Slack alert
- Screenshots to show the Completed Project

    - Github Repo

Submission Form: https://tally.so/r/w4RP2b