Pando2 CO2 Prediction Project: Workflow and Script Guide

This document provides a comprehensive overview of the files in this project and a step-by-step guide on how to run the end-to-end MLOps workflow.

1. Project Overview

This project implements a complete MLOps pipeline on AWS to train, evaluate, deploy, and monitor a CO2 prediction model using Amazon SageMaker. The workflow is automated, reproducible, and includes advanced features like hyperparameter tuning, bias detection, and automated retraining.

The core workflow is:

Prepare Data -> Tune Model -> Train Best Model -> Check for Bias -> Evaluate -> Deploy -> Monitor

2. Core Pipeline Scripts

These are the Python scripts that perform the main ML tasks. They are executed by SageMaker jobs within the pipeline and are not meant to be run directly by the user.

`preprocess.py`

- **Purpose**: Performs data preparation and feature engineering.
- **Functionality**:
    - Loads the raw `co2_data.csv`.
    - Creates time-series features like lags and rolling averages.
    - Splits the data into training, validation, and test sets.
    - Saves the processed datasets as Parquet files in S3.
- **Executed By**: The `PrepareCO2Data` `ProcessingStep` in `launch_pipeline.py`.

3. Pipeline Orchestration & Execution

These scripts are used to define, launch, and manage the SageMaker Pipelines.

`launch_pipeline.py`

- **Purpose**: The main entry point for the entire MLOps workflow.
- **Functionality**:
  - Defines every step of the pipeline: data prep, quality checks, tuning, training, bias checks, evaluation, and conditional deployment.
  - Connects the inputs and outputs of each step.
  - Upserts (creates or updates) the pipeline definition in your AWS account.
  - Starts a new execution of the pipeline.
- **How to Use**: Run this script from a SageMaker environment (like Studio) to start the entire process.

```
python launch_pipeline.py
```

4. AWS Lambda Function Scripts

These scripts contain the code for AWS Lambda functions that perform specific serverless tasks. You need to create Lambda functions in the AWS console and paste this code into them.

`deploy.py`

- **Purpose**: Deploys a registered SageMaker model to a real-time endpoint.
- **Functionality**:
  - Receives a model package ARN from the pipeline.
  - Creates a SageMaker Model, Endpoint Configuration (with data capture enabled), and Endpoint.
  - Handles updates if the endpoint already exists.
- **Used By**: The `DeployCO2Model` `LambdaStep` in `launch_pipeline.py`.

`cleanup.py`

5. Monitoring and Automation Scripts

These are one-time setup scripts to configure monitoring, alerting, and automated retraining for your deployed endpoint.

`setup_model_monitor.py`

- **Purpose**: Sets up a schedule to monitor the live endpoint for **data drift**.
- **Functionality**: Creates a `DefaultModelMonitor` schedule that runs hourly, compares live traffic against the baseline created by the pipeline, and generates drift reports.
- **How to Use**: Run this script once after the endpoint is successfully deployed.

```
python setup_model_monitor.py
```

`setup_bias_monitor.py`

6. Utility and Testing Scripts

`test_endpoint.py`

- **Purpose**: A simple script to test if the deployed endpoint is working correctly.
- **Functionality**: Sends a sample payload (a CSV string of feature values) to the endpoint and prints the predicted CO2 value.
- **How to Use**: Run this after the pipeline has successfully deployed the endpoint to verify it's live.

```
python test_endpoint.py
```

`hpo_strategies_example.py`

- **Purpose**: An experimental script to compare different hyperparameter tuning strategies (Bayesian, Random, Hyperband).
- **Functionality**: Defines and launches standalone SageMaker Tuning jobs. This is useful for research to determine the

7. Recommended Workflow

1. **Initial Setup**:
    - Upload your raw data (`co2_data.csv`) to the S3 bucket specified in `launch_pipeline.py` .
    - Create the Lambda functions (`deploy.py` , `cleanup.py`) in the AWS Console and update the ARNs in `launch_pipeline.py` .
2. **Launch the Main Pipeline**:
    - Run `python launch_pipeline.py` .
    - Monitor the execution in the SageMaker Studio UI.
3. **Verify Deployment**:
    - Once the pipeline succeeds, run `python test_endpoint.py` to confirm the endpoint is live and returning predictions.
4. **Set Up Monitoring**: