

Examen Parcial MLOPS

Nombre: Luis Arturo Garro Hurtado

Repositorio: https://github.com/luisgarro36-dotcom/mlops_project

Código: 202412011

Curso: MLOPS

Pasos y evidencias:

1. GitHub guarda el código y las versiones del proyecto.
2. DVC guarda y versiona los datos.
3. Docker garantiza que todo funcione igual donde sea que se ejecute.
4. MLflow guarda los resultados del modelo y del monitoreo (usa indicador psi)

1. Configuración Inicial y Estructura del Proyecto

Comando	Descripción
mkdir mlops_project	Crea la carpeta principal del proyecto.
cd mlops_project	Entra al directorio.
mkdir data\raw data\processed src models	Crea subcarpetas para datos, código (src) y modelos.
ni requirements.txt -Force	Crea el archivo de dependencias vacío (PowerShell).
ni Dockerfile -Force	Crea el archivo de configuración de Docker vacío.
ni dvc.yaml -Force	Crea el archivo de definición de <i>pipeline</i> de DVC.
ni src\preprocessing.py -Force	Crea el archivo de código de preprocesamiento.
ni src\monitor_data_drift.py -Force	Crea el archivo de código de monitoreo.

2. Entorno Virtual e Instalación de Dependencias

Comando	Descripción
python -m venv .venv	Crea el entorno virtual llamado .venv.
.\.venv\Scripts\Activate	Activa el entorno virtual.
pip install --upgrade pip	Actualiza pip dentro del entorno.
pip install pandas numpy scikit-learn dvc mlflow python-dotenv	Instala las librerías principales del proyecto.

3. Inicialización de Versiones (Git y DVC)

Comando	Descripción
git init	Inicializa el repositorio de Git.
dvc init	Inicializa el repositorio de DVC.

dvc add data/raw/dataset.csv	Versiona el archivo de datos pesados (dataset.csv) con DVC.
git add data/raw/dataset.csv.dvc .gitignore	Añade a Git el puntero de DVC y el archivo de ignorados.
git commit -m "feat: inicializacion de dvc y versionamiento de datos crudos"	Guarda la versión inicial en Git.

4. Ejecución de la Pipeline y Seguimiento

Comando	Descripción
dvc repro	Ejecuta la <i>pipeline</i> definida en dvc.yaml (preprocesamiento, entrenamiento, monitoreo).
git add dvc.yaml dvc.lock src/ requirements.txt	Añade todos los archivos de configuración y código nuevos a Git.
git commit -m "feat: primera ejecucion de la pipeline y codigo fuente"	Guarda la versión de la <i>pipeline</i> en Git.

5. Contenerización y Prueba Local (Docker)

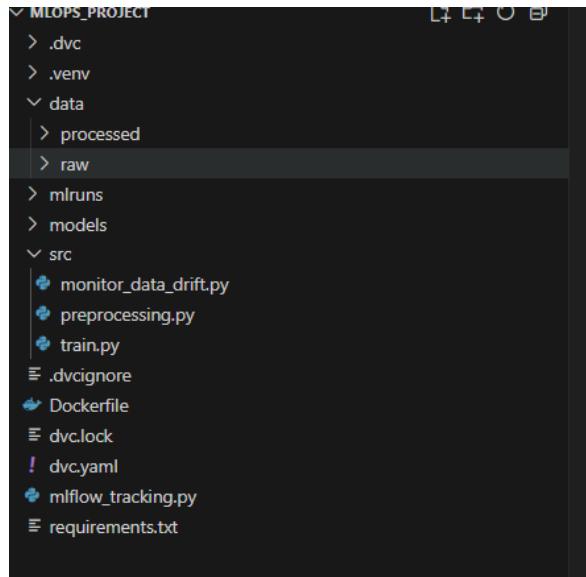
Comando	Descripción
deactivate	Desactiva el entorno virtual para usar Docker.
docker build -t mlops_project .	Construye la imagen Docker a partir del Dockerfile.
docker run -p 5000:5000 mlops_project	Ejecuta el contenedor y mapea el puerto 5000 (para ver MLflow en http://localhost:5000).

6. Configuración de DVC Remoto y Subida Final a GitHub

Comando	Descripción
deactivate	Desactiva el entorno (si estaba activo) para comandos de Git.
git remote set-url origin https://github.com/luisgarro36-dotcom/mlops_project.git	Corrige y establece la URL remota de GitHub.
.venv\Scripts\Activate	Activa el entorno para DVC.
dvc remote add -d local_remote C:/Users/Luisinho/Documents/dvc_remote	Define la carpeta local como el almacenamiento remoto de DVC.
dvc push	Sube los datos y modelos pesados a la "Bodega" (dvc_remote).
deactivate	Desactiva el entorno para Git.
git add .	Añade todos los archivos restantes, incluyendo la configuración de DVC Remote.
git commit -m "feat: subida final con configuracion de dvc remote y punteros"	Confirma la versión final de la "Receta".
git push -u origin main	Sube la <i>pipeline</i> y el código final a GitHub.

Anexos:

Carpeta de proyecto:

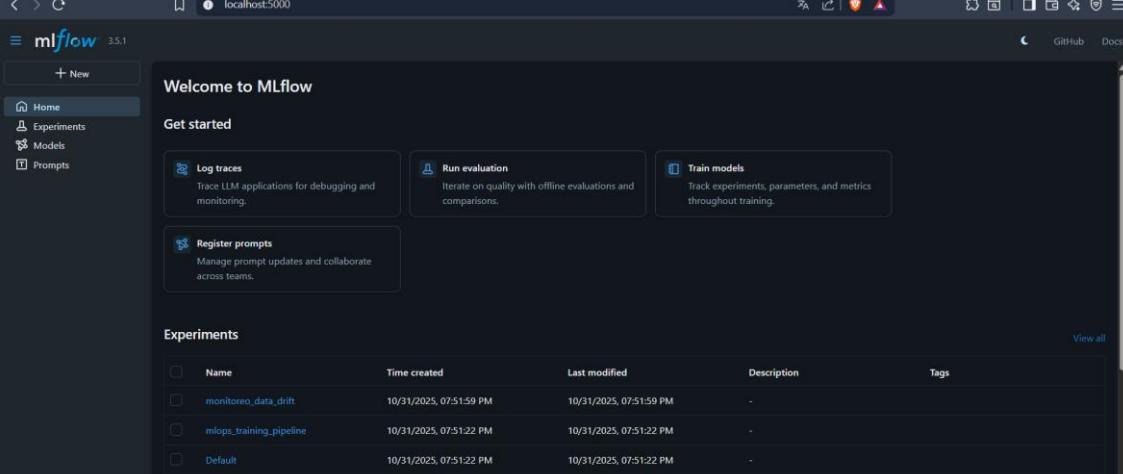


Ejecución:

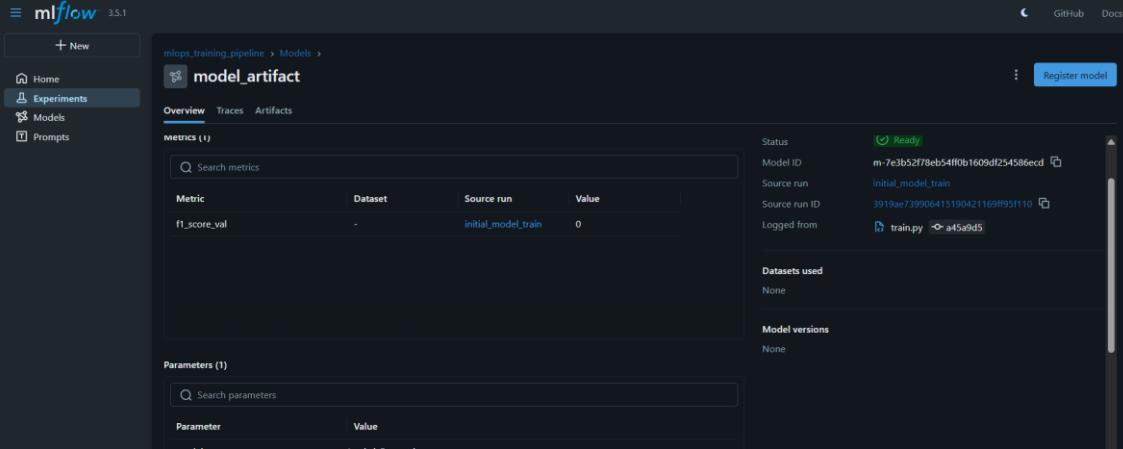
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS docker + ×
```

```
PS C:\Users\Luisinho\Documents\mlops_project> Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned
>>
PS C:\Users\Luisinho\Documents\mlops_project> .\venv\Scripts\Activate
(.venv) PS C:\Users\Luisinho\Documents\mlops_project> dvc repro
'data\raw\dataset.csv.dvc' didn't change, skipping
Stage 'preprocess' didn't change, skipping
Stage 'train' didn't change, skipping
Stage 'monitor_drift' didn't change, skipping
Data and pipelines are up to date.
(.venv) PS C:\Users\Luisinho\Documents\mlops_project> deactivate
PS C:\Users\Luisinho\Documents\mlops_project> docker run -p 5000:5000 mllops_project
[MLflow] Security middleware enabled with default settings (localhost-only). To allow connections from other hosts, use --host 0.0.0.0 and configure --allowed-hosts and --cors-allowed-origins.
INFO:     Uvicorn running on http://0.0.0.0:5000 (Press CTRL+C to quit)
INFO:     Started parent process [24]
INFO:     Started server process [26]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Started server process [27]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Started server process [29]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Started server process [28]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO: 172.17.0.1:43718 - "GET / HTTP/1.1" 304 Not Modified
INFO: 172.17.0.1:43718 - "GET /ajax-api/2.0/mlflow/experiments/search?max_results=5&order_by=last_update_time+DESC HTTP/1.1" 200 OK
```

Visualización en MLflow:



The screenshot shows the MLflow 3.5.1 home page at localhost:5000. The top navigation bar includes links for GitHub and Docs. The main header says "Welcome to MLflow". Below it, a "Get started" section features three cards: "Log traces" (Trace LLM applications for debugging and monitoring.), "Run evaluation" (Iterate on quality with offline evaluations and comparisons.), and "Train models" (Track experiments, parameters, and metrics throughout training.). A "Experiments" section lists three entries: "monitoreo_data_drift" (Time created: 10/31/2025, 07:51:59 PM), "mlops_training_pipeline" (Time created: 10/31/2025, 07:51:22 PM), and "Default" (Time created: 10/31/2025, 07:51:22 PM). A "View all" link is located at the top right of this table.



The screenshot shows the MLflow model artifact page for the "model_artifact" entry under the "mlops_training_pipeline" experiment. The top navigation bar includes links for GitHub and Docs. The left sidebar shows the experiment path "mlops_training_pipeline > Models > model_artifact". The main content area has tabs for "Overview", "Traces", and "Artifacts". The "Overview" tab displays metrics and parameters. The "metrics (1)" section shows one metric: f1_score_val with a value of 0. The "Parameters (1)" section shows one parameter: model_type with a value of LogisticRegression. To the right, detailed information is provided: Status (Ready), Model ID (m-7e3b52f78eb54ff0b1609df254586ecd), Source run (initial_model_train), Source run ID (3919ae739906415190421169ff95f110), and Logged from (train.py -> a45a9d5). Sections for "Datasets used" and "Model versions" are also present.