

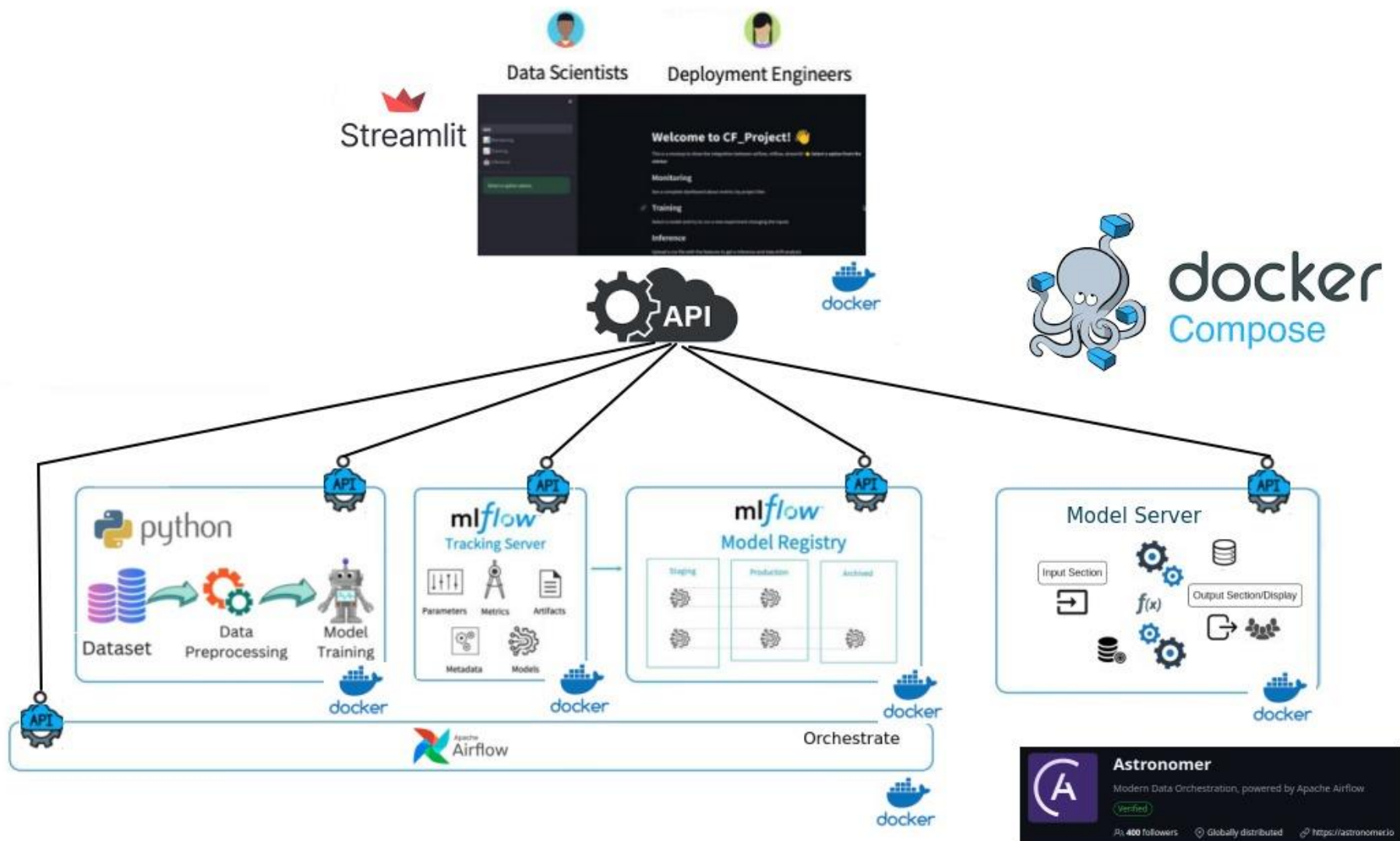
MLOps Proyecto Final

Jose Manuel Mejia Perales

Eduardo Fazzio Cervantes Gutierrez

Curso: MLOps Codigo Facilito

10 De Octubre de 2024



Desarrollado en AstroDev

- Dags: Instrucciones y tareas para MLFlow
- Include: contiene los datos y estructuras locales, así como los modelos que son necesarios para la función del programa.
- Requirements: incluye todas las librerías que son necesarias para poder realizar los flujos.

AstroDev:

Astro Dev es una plataforma de desarrollo que se centra en facilitar la creación de aplicaciones web modernas. Algunas de sus características clave incluyen:

- Framework para aplicaciones: Proporciona un marco que permite desarrollar aplicaciones de una manera eficiente y rápida, optimizando el rendimiento y la carga de contenido.
- Componentes reutilizables: Facilita el uso de componentes que se pueden reutilizar en diferentes partes de una aplicación, lo que ahorra tiempo y esfuerzo en el desarrollo.
- Optimización para SEO: Incluye características que ayudan a mejorar la visibilidad en motores de búsqueda, lo que es crucial para muchas aplicaciones web.
- Enfoque en la experiencia del desarrollador: Proporciona herramientas y documentación que hacen que el proceso de desarrollo sea más fluido y accesible.

Para el uso de astro dev basta con simplemente correr el comando

```
astro dev start
```

Para detener los contenedores en caso de que sea necesarios basta con correr

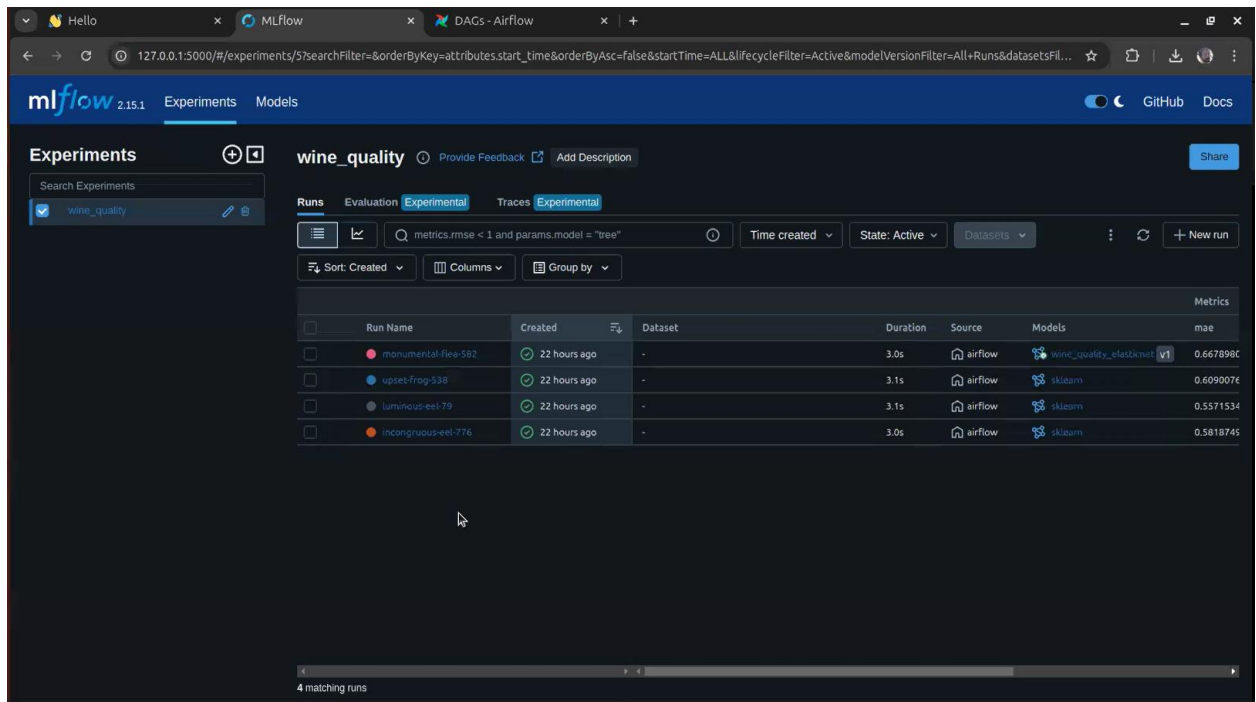
```
astro dev stop
```

MLflow:

MLflow es una plataforma de código abierto diseñada para gestionar el ciclo de vida del aprendizaje automático. Proporciona herramientas para:

- Seguimiento de experimentos: Permite registrar y comparar diferentes experimentos de modelado, incluyendo parámetros, métricas y artefactos.
- Proyectos: Facilita la organización y la ejecución de proyectos de aprendizaje automático de manera reproducible.
- Modelos: Permite empaquetar y gestionar modelos de manera que se puedan desplegar en diferentes entornos.
- Registro de modelos: Ofrece un sistema para almacenar y versionar modelos, lo que facilita la colaboración y el despliegue.

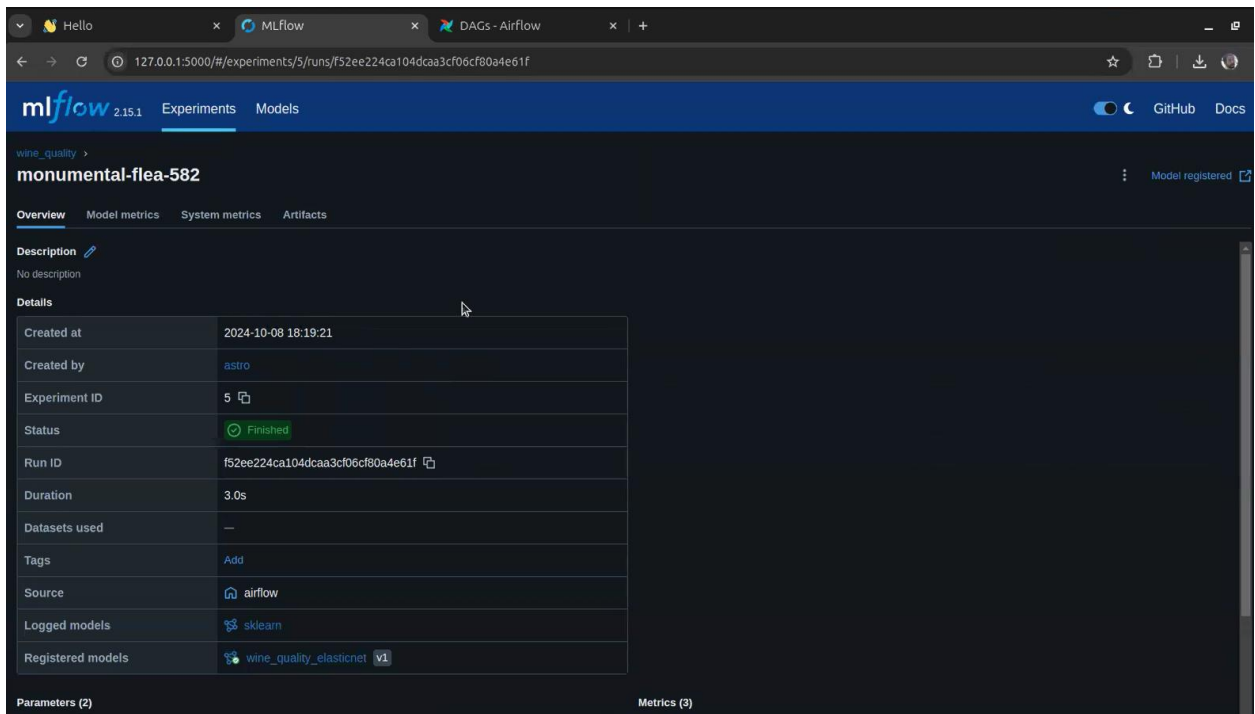
Vista general de los distintos experimentos:



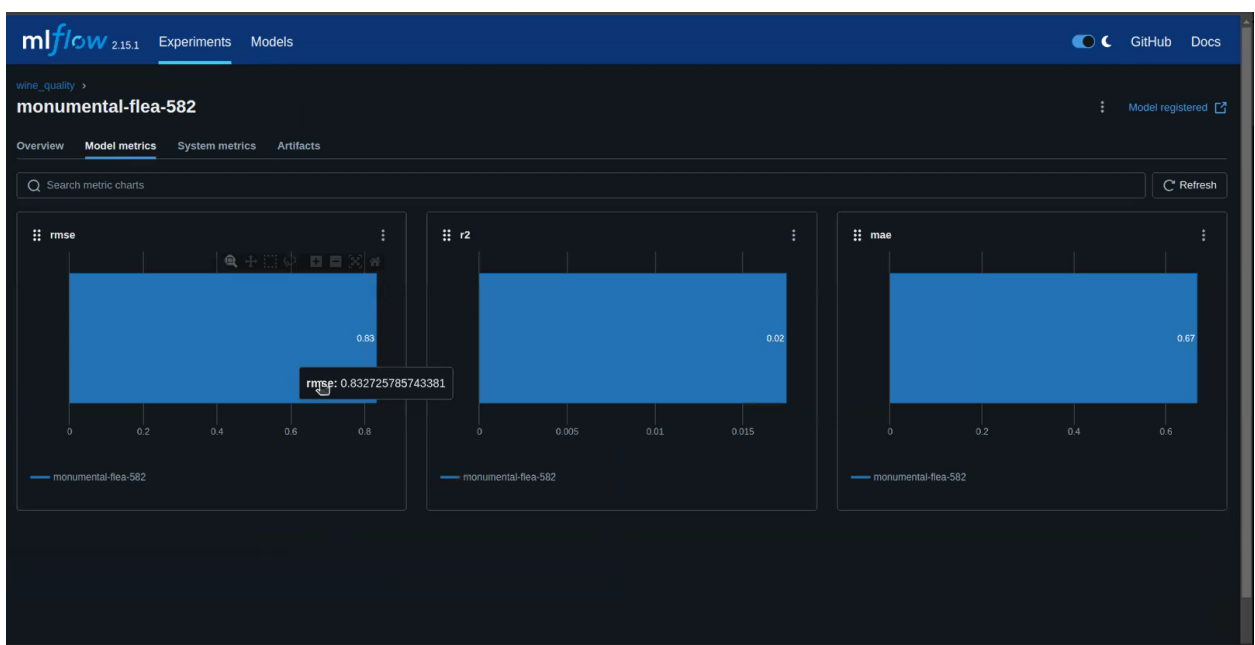
The screenshot displays the MLflow web interface for the 'wine_quality' experiment. The interface includes a search bar, filters, and a table of runs. The table lists four runs, each with a unique name, creation time, duration, source, and associated models. The 'mae' metric is also shown for each run.

Run Name	Created	Dataset	Duration	Source	Models	Metrics
monumental-flee-582	22 hours ago	-	3.0s	airflow	wine_quality_elasticnet [v1]	0.667898
upset-frog-538	22 hours ago	-	3.1s	airflow	sklearn	0.609007
luminous-eel-79	22 hours ago	-	3.1s	airflow	sklearn	0.557153
incongruous-eel-776	22 hours ago	-	3.0s	airflow	sklearn	0.581874

Descripción de la pruebas:



Metricas:



Artifacts:

The screenshot shows the MLflow web interface. At the top, there's a navigation bar with 'mlflow 2.15.1', 'Experiments', and 'Models'. Below this, the breadcrumb 'wine_quality >' leads to the experiment 'monumental-flea-582'. The 'Artifacts' tab is selected, showing a file tree on the left with files like 'MLmodel', 'conda.yaml', 'model.pkl', 'python_env.yaml', 'requirements.txt', 'data_drift_suite.html', and 'winequalityred.csv'. The main content area displays details for the 'wine_regression' artifact, including its path and a link to the model registry. Below this, there are two sections: 'Model schema' and 'Validate the model before deployment'. The 'Model schema' section has a table with columns 'Name' and 'Type', but it's empty with a note that no schema is present. The 'Validate the model before deployment' section contains a code snippet for validating the model.

monumental-flea-582

Overview Model metrics System metrics **Artifacts**

wine_regression

Path: s3://mlflowdataredwine/f52ee224ca104dcaa3cf06cf80a4e61f/artifacts/wine_regression

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the [model registry](#).

Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
No schema. See MLflow docs for how to include input and output schema with your model.	

Validate the model before deployment

Run the following code to validate model inference works on the example payload, prior to deploying it to a serving endpoint

```
from mlflow.models import validate_serving_input

model_uri = 'runs:/f52ee224ca104dcaa3cf06cf80a4e61f/wine_regression'

# The logged model does not contain an input example.
# Manually generate a serving payload to verify your model prior to deployment.
from mlflow.models import convert_input_example_to_serving_input

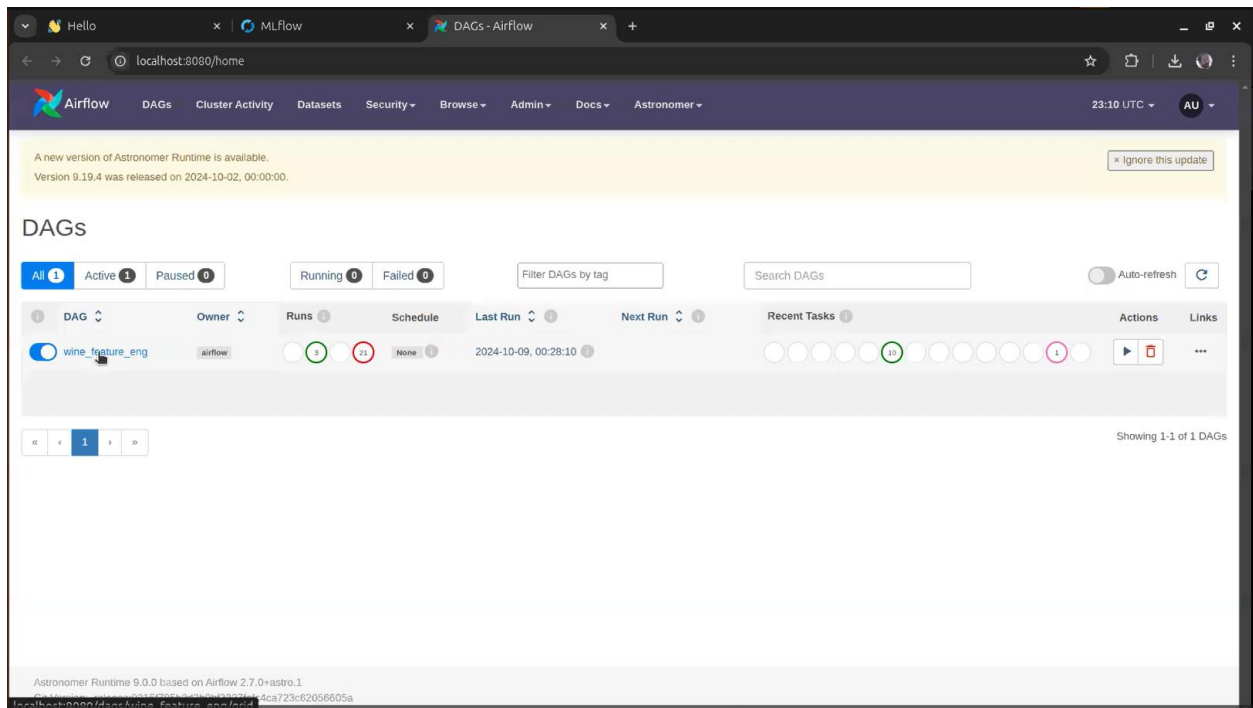
# Define INPUT_EXAMPLE via assignment with your own input example to the model
# A valid input example is a data instance suitable for pyfunc prediction
serving_payload = convert_input_example_to_serving_input(INPUT_EXAMPLE)
```

En esta encontraremos el modelo para poder ser exportado a distintas instancias.

Airflow

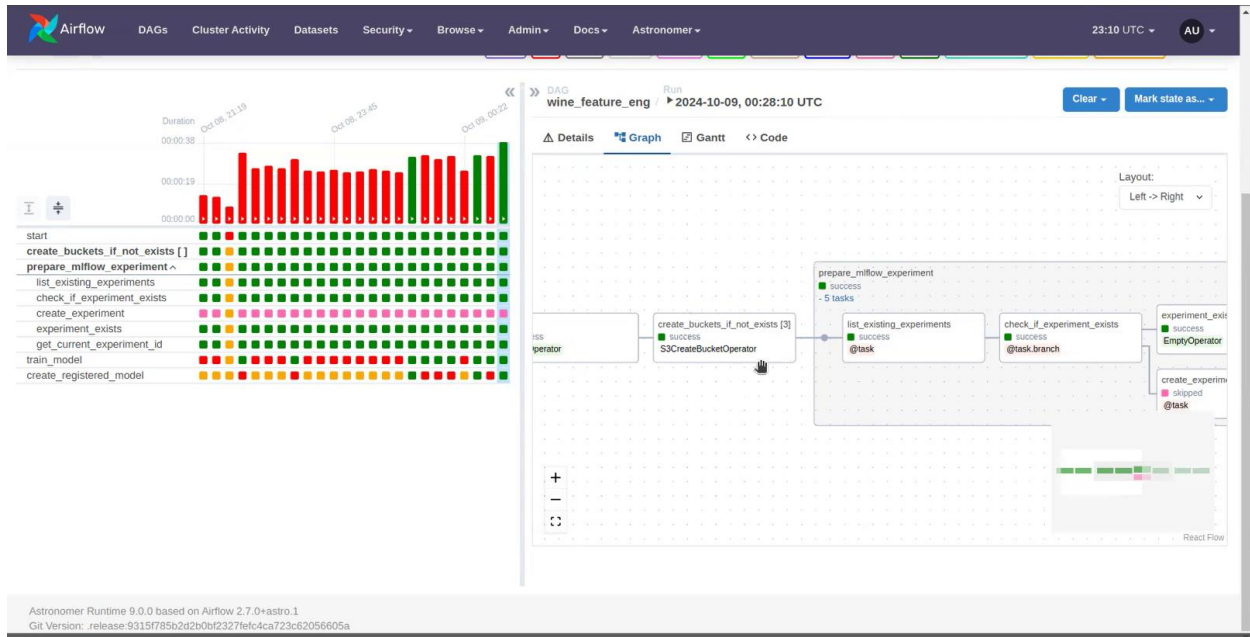
Apache Airflow es una plataforma de código abierto diseñada para programar y monitorizar flujos de trabajo de datos. Permite a los usuarios definir tareas y sus dependencias, lo que facilita la orquestación de procesos complejos. Algunas de sus características principales incluyen:

- Definición de flujos de trabajo: Los flujos de trabajo se definen como "DAGs" (Directed Acyclic Graphs), lo que permite visualizar y organizar las tareas de manera clara.
- Programación de tareas: Airflow permite programar tareas para que se ejecuten en momentos específicos o en respuesta a eventos, facilitando la automatización de procesos.
- Interfaz de usuario: Ofrece una interfaz web intuitiva para monitorizar y gestionar los flujos de trabajo, permitiendo a los usuarios ver el estado de las tareas y realizar seguimientos.
- Extensibilidad: Se puede integrar con diversas herramientas y servicios, como bases de datos, sistemas de almacenamiento y plataformas en la nube, gracias a su arquitectura modular.
- Manejo de errores: Incluye mecanismos para manejar errores y reintentos, lo que ayuda a garantizar la estabilidad de los flujos de trabajo.



Los Dag son series de tareas establecidas para poder correr el modelo que tenemos planeado realizar.

Experimentos a realizar



Intefaz principal



Monitoreo:

The screenshot displays the Evidently AI Monitoring interface. On the left sidebar, the 'Monitoring' tab is selected. The main panel is titled 'Reporting' and shows a project named 'smiling-bee-730'. A list of drift tests is shown, all with a green checkmark indicating they passed. The tests include 'Share of Drifted Columns', 'Drift per Column' for 'sulphates', and 'Drift per Column' for 'quality'. Each test entry includes a 'Details' link.

app

- Monitoring
- Training
- Inference

EVIDENTLY AI

[Source code](#) [Evidently docs](#)

Select model from run

smiling-bee-730

Reporting

Project: smiling-bee-730

40e6be643bbc4700b64de03bf75ac265

All tests

- ☒ Share of Drifted Columns Details
The drift is detected for 0% features (0 out of 12). The test threshold is $lt=0.3$
- ☒ Drift per Column Details
The drift score for the feature **sulphates** is 0.021. The drift detection method is Wasserstein distance (normed). The drift detection threshold is 0.1.
- ☒ Drift per Column Details
The drift score for the feature **quality** is 0.023. The drift detection method is Wasserstein distance (normed). The drift detection threshold is 0.1.
- ☒ Drift per Column Details

Interfaz:

The screenshot displays the Evidently AI Inference interface. On the left sidebar, the 'Inference' tab is selected. The main panel shows a list of input features with corresponding numerical values (all 0.00) and a 'Predict' button. The features listed are 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', and 'alcohol'. Below the input fields, there is a note: 'If you have a DataFrame from a batch please upload'.

app

- Monitoring
- Training
- Inference

free sulfur dioxide

total sulfur dioxide

density

pH

sulphates

alcohol

Predict

If you have a DataFrame from a batch please upload