

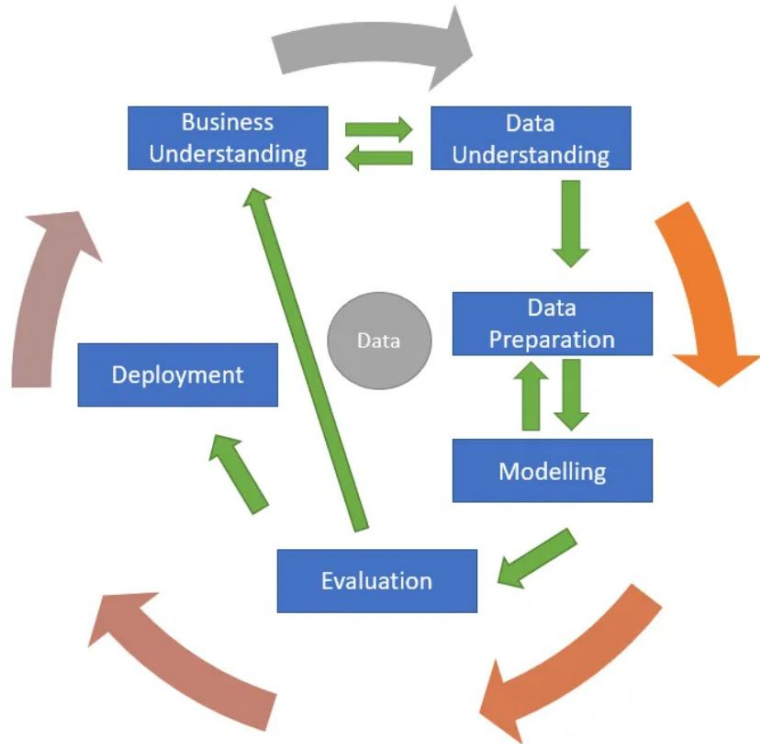
MLOps= ML + DEV + OPS



Machine Learning Operations

Diego Mosquera Uzcátegui
Marzo 2025

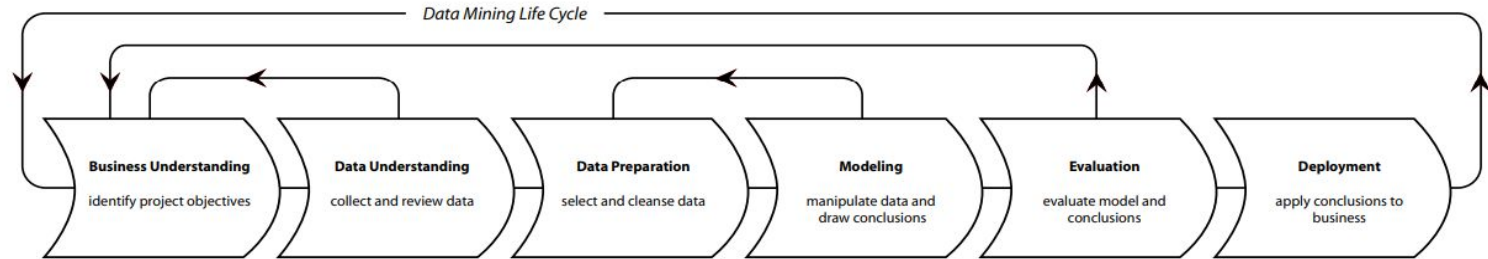
Cross-Industry Standard Process for Data Mining



**Metodología iterativa para
gestionar el ciclo de vida de
proyectos de minería de datos
orientados a data driven**

Diagrama de procesos CRIPS-DM

Tareas de CRIPS-DM



Determine Business Objectives

*Background
Business Objectives
Business Success Criteria
(Log and Report Process)*

Assess Situation

*Inventory of Resources,
Requirements, Assumptions,
and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
(Log and Report Process)*

Determine Data Mining Goals

*Data Mining Goals
Data Mining Success Criteria
(Log and Report Process)*

Produce Project Plan

*Project Plan
Initial Assessment of Tools and
Techniques
(Log and Report Process)*

Collect Initial Data

*Initial Data Collection Report
(Log and Report Process)*

Describe Data

*Data Description Report
(Log and Report Process)*

Explore Data

*Data Exploration Report
(Log and Report Process)*

Verify Data Quality

*Data Quality Report
(Log and Report Process)*

Data Set

*Data Set Description
(Log and Report Process)*

Select Data

*Rationale for Inclusion/
Exclusion
(Log and Report Process)*

Clean Data

*Data Cleaning Report
(Log and Report Process)*

Construct Data

*Derived Attributes
Generated Records
(Log and Report Process)*

Integrate Data

*Merged Data
(Log and Report Process)*

Format Data

*Reformatted Data
(Log and Report Process)*

Select Modeling Technique

*Modeling Technique
Modeling Assumptions
(Log and Report Process)*

Generate Test Design

*Test Design
(Log and Report Process)*

Build Model Parameter Settings

*Models
Model Description
(Log and Report Process)*

Assess Model

*Model Assessment
Revised Parameter
(Log and Report Process)*

Evaluate Results

*Align Assessment of Data
Mining Results with
Business Success Criteria
(Log and Report Process)*

Approved Models

*Review Process
Review of Process
(Log and Report Process)*

Determine Next Steps

*List of Possible Actions
Decision
(Log and Report Process)*

Plan Deployment

*Deployment Plan
(Log and Report Process)*

Plan Monitoring and Maintenance

*Monitoring and
Maintenance Plan
(Log and Report Process)*

Produce Final Report

*Final Report
Final Presentation
(Log and Report Process)*

Review Project

*Experience
Documentation
(Log and Report Process)*

¿Cuál es el problema?

El 85% de los modelos de ML no llegan a producción.

¿Por qué?











Despliegue vs Producción

Concepto	Despliegue de un Modelo	Llevar un Modelo a Producción (MLOps)
Definición	El acto de exponer un modelo entrenado para su consumo en un entorno de ejecución (local, nube, edge, API, etc.).	Todo el proceso necesario para que el modelo esté en un entorno productivo con monitoreo, escalabilidad y mantenibilidad.
Alcance	Se enfoca en la fase de exposición del modelo (ej., API REST, Batch Processing, Edge AI).	Incluye el despliegue, pero también aspectos como versionamiento, CI/CD, monitoreo, actualización, seguridad y gobernanza.
Ejemplo	Exponer un modelo mediante FastAPI o Flask en un servidor.	Hacer que ese modelo sea parte de un sistema empresarial con infraestructura robusta, escalabilidad y monitoreo.
Herramientas comunes	FastAPI, Flask, TensorFlow Serving, TorchServe, Docker.	MLflow, Kubeflow, Airflow, SageMaker, Vertex AI, Azure ML, Prometheus, Grafana.

¿Qué haremos en este curso?

- Nos enfocaremos en el proceso de **despliegue** de modelos de ML.
 - Flask, FastAPI, etc.
- Aprenderemos sobre los principios de MLOps.
 - MLFlow
- Analizaremos las herramientas disponibles para integrar un modelo expuesto a los sistemas productivos de la empresa.

Alcance MLOps

Aspecto	 Sin MLOps (Modelo Solo Desplegado)	 Con MLOps (Modelo en Producción)
Entrenamiento del modelo	Se entrena manualmente y se expone a través de una API.	Se despliega automáticamente cada nueva versión del modelo usando CI/CD .
Pruebas automatizadas	 No hay pruebas automatizadas, pueden aparecer errores en producción.	 Se realizan pruebas automáticas en cada nueva versión antes de su despliegue.
Monitoreo del modelo	 No se monitorea la performance del modelo, lo que puede llevar a predicciones erróneas.	 Se monitorea el rendimiento en tiempo real y se detecta drift en los datos .
Manejo del cambio en los datos	 Si los datos cambian, no hay un mecanismo automático para actualizar el modelo.	 Se automatiza el reentrenamiento cuando el modelo pierde precisión.
Rollback y control de versiones	 No se puede volver fácilmente a versiones anteriores si hay errores.	 Se pueden hacer rollback a versiones anteriores si algo falla, gracias al versionamiento de modelos.

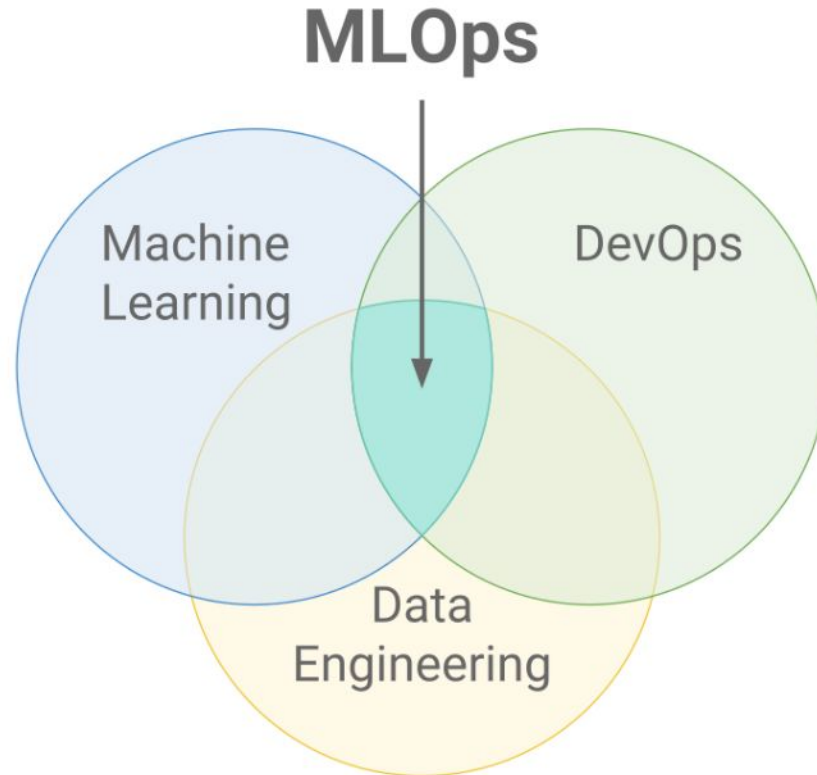
Ejemplo: detección de fraude bancario

Aspecto	Sin MLOps (Modelo Solo Desplegado)	Con MLOps (Modelo en Producción)
Entrenamiento del modelo	Se entrena manualmente en un Jupyter Notebook.	Se entrena automáticamente con un pipeline en MLflow o Kubeflow .
Versionamiento de código y datos	No hay control de versiones para código ni datos.	Se usa Git para código y DVC/MLflow para datos y modelos .
Despliegue	Se ejecuta en un servidor local o API manualmente.	Se implementa con CI/CD en un clúster Kubernetes .
Consumo del modelo	Se expone mediante una API simple con Flask.	API escalable en la nube con FastAPI + Kubernetes .

Ejemplo: detección de fraude bancario

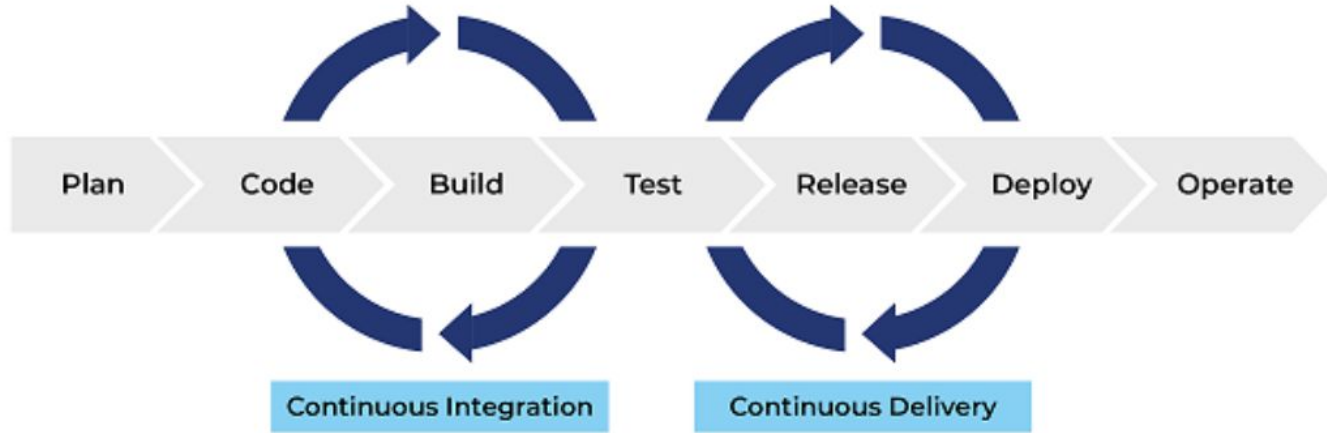
Monitoreo de métricas	No hay monitoreo. Si el modelo falla, hay que revisar logs manualmente.	Se usa Prometheus y Grafana para monitorear latencia, drift y métricas de desempeño.
Manejo de Data Drift	No se detecta si los datos han cambiado.	Se usa Evidently AI para detectar data drift y alertar al equipo.
Reentrenamiento del modelo	Manual: un analista tiene que darse cuenta de que el modelo ha perdido precisión.	Automático: Airflow/Kubeflow reentrena el modelo si la precisión baja de un umbral.
Rollback (volver a versiones anteriores)	Si el modelo nuevo falla, es difícil volver a la versión anterior.	Se usa MLflow Model Registry para restaurar un modelo anterior fácilmente.
Seguridad y compliance	No hay control de acceso a los modelos.	Se usa MLflow + auditoría de modelos para control y trazabilidad.
Escalabilidad	Si hay más usuarios, la API se satura y colapsa.	El modelo se escala automáticamente en la nube con Kubernetes + autoscaling .

¿Qué es MLOps (Machine Learning Operations)?



¿Qué es MLOps (Machine Learning Operations)?

CI/CD



DevOps

Diferencias entre ML Tradicional y MLOps

✓ Machine Learning Tradicional:

- Modelos entrenados en notebooks de forma manual.
- No hay un pipeline estructurado para entrenamiento y despliegue.
- Los modelos no se versionan correctamente.

✓ Machine Learning con MLOps:

- Pipelines automatizados para entrenamiento y despliegue.
- Versionamiento de datos y modelos con **MLflow, DVC, Git**.
- Integración con **CI/CD y monitoreo** para detectar fallos.

Beneficios y Desafíos de MLOps

✓ Beneficios:

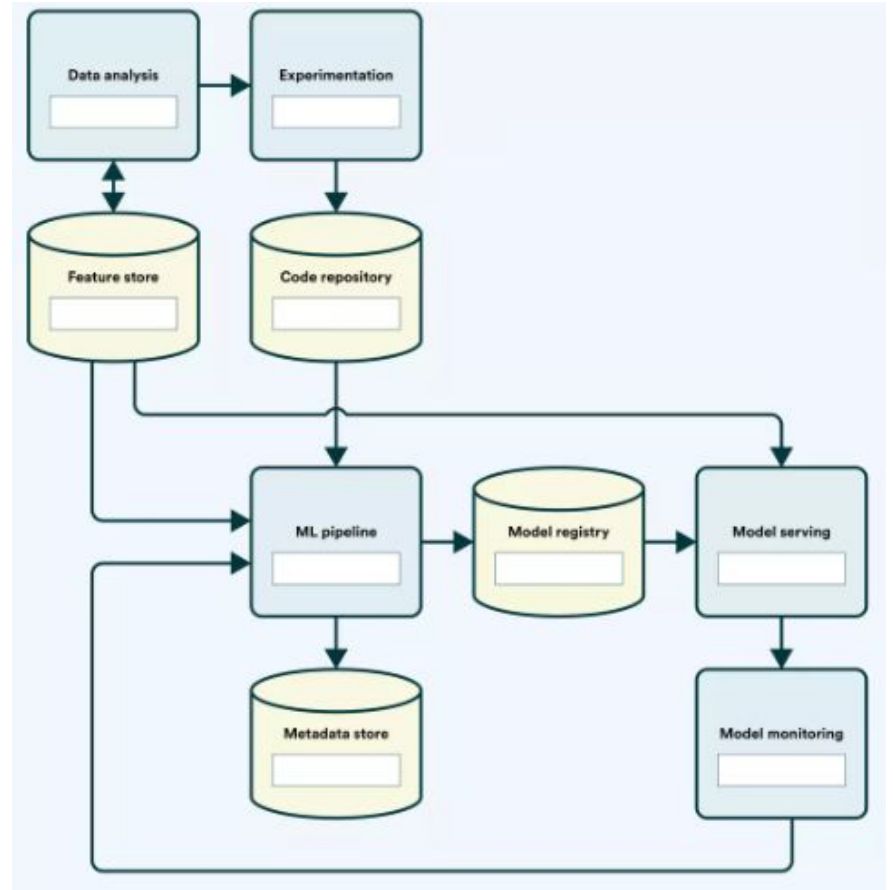
- Despliegues más rápidos y confiables.
- Menos intervención manual y menos errores.
- Facilita la colaboración entre equipos de Data Science, Ingeniería y DevOps.
- Mayor estabilidad y monitoreo en producción.

✓ Desafíos:

- Requiere conocimientos de **DevOps y Data Engineering**.
- Mayor complejidad en comparación con el desarrollo tradicional de ML.
- Necesidad de herramientas especializadas (**MLflow, Kubeflow, Airflow**).

MLOps Stack

Herramientas necesarias para implementar un flujo de trabajo efectivo de MLOps.



Discusión interactiva

Pregunta a los estudiantes:

- ¿Han trabajado en proyectos de ML? ¿Han tenido problemas con la implementación?
- ¿Cómo creen que MLOps puede ayudar en su experiencia?
- ♦ Breve lluvia de ideas y participación.

Actividad 1: Creación de un Entorno de Trabajo para ML

📌 **Duración:** 20 min.

📌 **Conceptos:** Configuración de entornos virtuales, estructuración de proyectos ML, control de versiones con Git.

📌 **Qué harán los estudiantes:**

- ✅ Crear un entorno virtual con `venv` o `conda`.
- ✅ Inicializar un repositorio en Git y hacer su primer commit.
- ✅ Crear la estructura base de un proyecto ML.

```
# Estructura básica de un proyecto ML
mkdir proyecto_ml
cd proyecto_ml
mkdir data src models tests
```


Actividad 2: Implementar una Prueba Unitaria

- **Duración:** 20 min.
- **Conceptos:** Creación de pruebas unitarias con `unittest` para validar código de ML.
- **Qué harán los estudiantes:**
 - ✓ Implementar una función de preprocesamiento de datos.
 - ✓ Crear una prueba unitaria para verificar su funcionamiento.