

Aprendizaje automático

Automatización de procesos de AA
AutoML y MLOps



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

AutoML



FACULTAD DE
INGENIERÍA



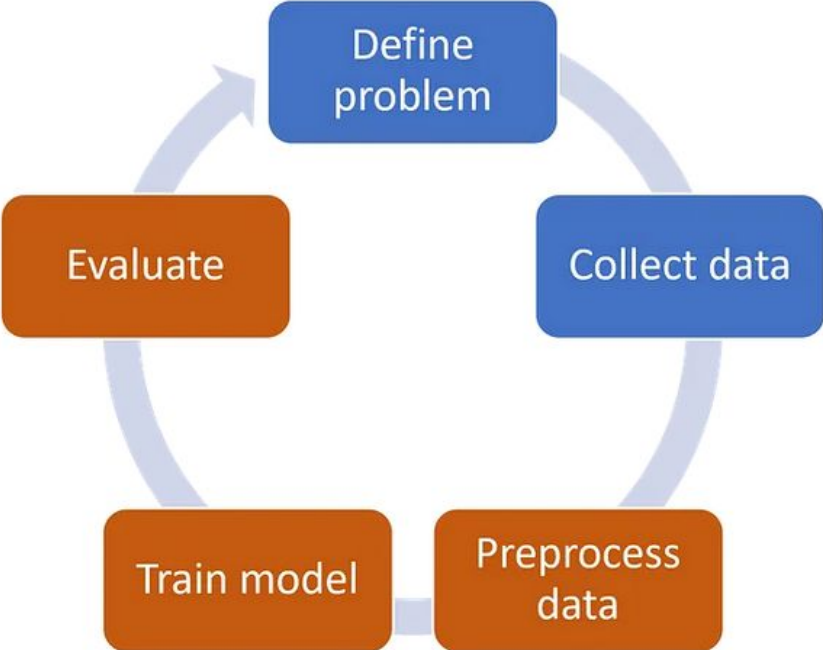
UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Automatización del Aprendizaje Automático (AutoML)

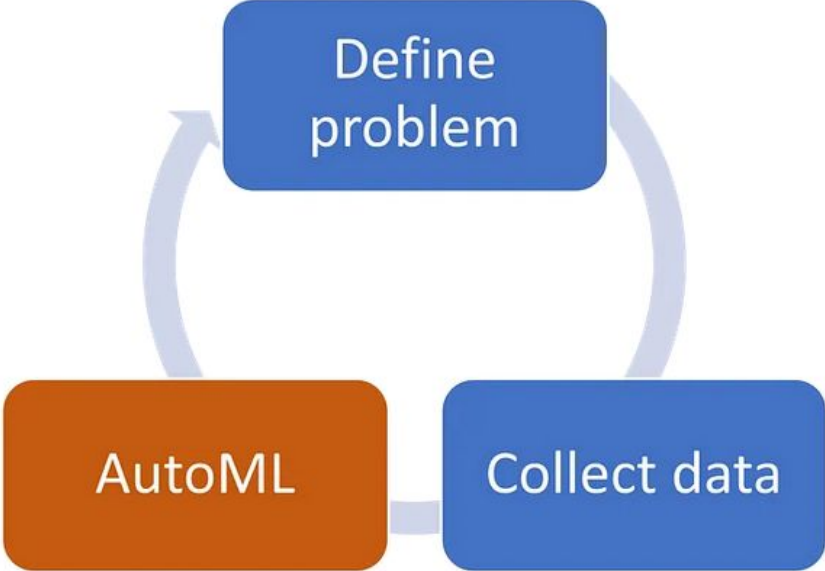
- Idea:
Automatizar partes del proceso de construcción de modelos de aprendizaje automático para reducir la complejidad y el tiempo necesario.
- Simplifica el proceso general de AA
- Evita la repetición de tareas comunes y tediosas de AA
- Permite evaluar métodos, hiperparámetros y otras alternativas
 - con mayor velocidad de implementación
 - con menor necesidad de expertise

AutoML vs Manual ML

Traditional ML training workflow



AutoML workflow



AutoML - Tareas que puede simplificar

- **Selección de Modelo:** probar varios algoritmos y configuraciones para determinar cuál produce los mejores resultados.
- **Preprocesamiento de Datos:** automatizar imputación de valores faltantes, la normalización de características y la codificación de variables categóricas.
- **Ajuste de Hiperparámetros:** búsqueda de la combinación óptima
- **Optimización del Conjunto de Modelos:** construir y combinar múltiples modelos en un ensamble para mejorar el rendimiento general del sistema.
- **Interpretación de Modelos:** capacidades de interpretación de modelos que ayudan a entender su funcionamiento
- **Despliegue de Modelos:** ayuda al despliegue en producción para que otros sistemas puedan utilizarlo para hacer predicciones en tiempo real.

AutoML - Algunas plataformas en la nube

1. Google Cloud AutoML

- Enlace: [Google Cloud AutoML](#)
- Características principales:
 - Permite a los usuarios construir y entrenar modelos de aprendizaje automático personalizados sin necesidad de experiencia en programación.
 - Admite diversos tipos de modelos, como clasificación, detección de objetos, segmentación de imágenes y modelado de series temporales.
 - Ofrece una interfaz de usuario intuitiva y herramientas de evaluación del modelo.

2. Amazon SageMaker Autopilot

- Enlace: [Amazon SageMaker Autopilot](#)
- Características principales:
 - Automatiza el proceso de construcción, entrenamiento y ajuste de modelos de aprendizaje automático.
 - Utiliza técnicas avanzadas para explorar automáticamente diferentes combinaciones de algoritmos y hiperparámetros.
 - Proporciona visualizaciones y métricas para evaluar y comparar modelos.

3. Microsoft Azure Automated Machine Learning

- Enlace: [Azure Automated Machine Learning](#)
- Características principales:
 - Simplifica el proceso de construcción de modelos de aprendizaje automático mediante la automatización de tareas como selección de algoritmos y ajuste de hiperparámetros.
 - Admite una variedad de tipos de modelos, incluyendo clasificación, regresión y series temporales.
 - Proporciona integración con otras herramientas de Azure, como Azure Machine Learning Studio.

4. DataRobot

- Enlace: [DataRobot AutoML](#)
- Características principales:
 - Automatiza el ciclo completo de vida del aprendizaje automático, desde la preparación de datos hasta la implementación de modelos en producción.
 - Utiliza inteligencia artificial para explorar miles de posibles pipelines de aprendizaje automático y seleccionar las mejores opciones.
 - Proporciona una interfaz gráfica de usuario y API para una fácil integración con sistemas existentes.

AutoML - Algunas soluciones para usar también localmente

1. Auto-sklearn

- Enlace: [Auto-sklearn](#)
- Características principales:
 - Implementa un enfoque basado en la biblioteca de aprendizaje automático scikit-learn de Python.
 - Utiliza búsqueda bayesiana y selección de modelos para encontrar automáticamente la mejor configuración.
 - Puede ejecutarse en máquinas locales y en la nube.

2. MLBox

- Enlace: [MLBox](#)
- Características principales:
 - Ofrece una interfaz de usuario sencilla y un conjunto de herramientas para la automatización del aprendizaje automático.
 - Incluye módulos para la limpieza de datos, la selección de características y el ajuste de hiperparámetros.
 - Compatible con Python y se puede ejecutar localmente o en la nube.

3. TPOT (Tree-based Pipeline Optimization Tool)

- Enlace: [TPOT](#)
- Características principales:
 - Utiliza algoritmos genéticos para buscar automáticamente el mejor pipeline de aprendizaje automático.
 - Genera código Python para los pipelines descubiertos, lo que permite su inspección y modificación.
 - Se puede ejecutar en entornos locales y en la nube.

4. AutoKeras

- Enlace: [AutoKeras](#)
- Características principales:
 - Implementa técnicas de búsqueda de arquitectura automatizada para el aprendizaje profundo.
 - Permite a los usuarios definir y entrenar modelos de aprendizaje profundo con un mínimo de código.
 - Es compatible con Python y puede ejecutarse en sistemas locales y en la nube.

AutoML - Algunas soluciones para usar también localmente

5. AutoGluon

- Enlace: [AutoGluon](#)
- Características principales:
 - Desarrollado por AWS y la Universidad de Washington, AutoGluon ofrece una interfaz de alto nivel para el aprendizaje automático.
 - Permite a los usuarios entrenar modelos de aprendizaje profundo y de aprendizaje automático de manera fácil y rápida.
 - Utiliza técnicas de búsqueda automática para ajustar automáticamente los hiperparámetros y seleccionar modelos.

6. FLAML (Fast, Lightweight AutoML)

- Enlace: [FLAML](#)
- Características principales:
 - FLAML es una biblioteca ligera y rápida para la búsqueda automatizada de hiperparámetros y la selección de modelos.
 - Utiliza técnicas como la optimización de búsqueda aleatoria y el aprendizaje de árboles de decisión para encontrar la mejor configuración.
 - Compatible con una variedad de tipos de modelos, incluyendo árboles de decisión, bosques aleatorios y refuerzo de gradientes.

7. H2O AutoML

- Enlace: [H2O AutoML](#)
- Características principales:
 - Ofrece una amplia gama de algoritmos de aprendizaje automático optimizados automáticamente para cada conjunto de datos.
 - Permite a los usuarios personalizar el proceso de AutoML y controlar los recursos computacionales utilizados.
 - Proporciona una interfaz de usuario basada en web y API para facilitar la integración con otras aplicaciones.

Ejemplo: AutoGLUON sobre series temporales de demanda

Tutorial: <https://auto.gluon.ai/stable/tutorials/timeseries/index.html>

Zoo: <https://auto.gluon.ai/stable/tutorials/timeseries/forecasting-model-zoo.html>

Ejemplo: AutoGLUON sobre series temporales de demanda

```
train_data = TimeSeriesDataFrame.from_data_frame(  
    df_train,  
    id_column="objeto",  
    timestamp_column="fecha_hora"  
)  
train_data.head()
```

✓ 0.6s

		potencia	
	item_id	timestamp	
	APA5DTR1EAE	2019-01-01	0.0000
	APA5DTR1EAS	2019-01-01	0.3276
	BACODTR1EAE	2019-01-01	0.0000
	BACODTR1EAS	2019-01-01	0.5725
	BACOROSEAE	2019-01-01	0.0000

		potencia	
	item_id	timestamp	
	CMINRRA1EAS	2019-05-31 23:45:00	2.5700
	CMINRRA2EAS	2019-05-31 23:45:00	2.5890
	DRIADG01EAS	2019-05-31 23:45:00	0.0000
	OCRLBTR1EAE	2019-05-31 23:45:00	0.0000
	OCRLBTR1EAS	2019-05-31 23:45:00	0.0675

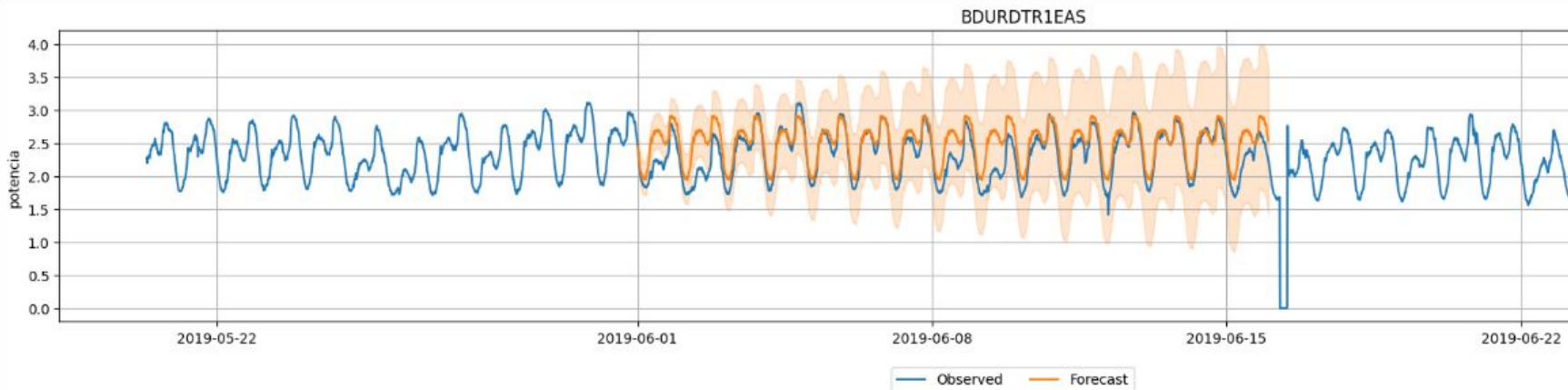
```
predictor = TimeSeriesPredictor(  
    prediction_length=1440, # 15 días (1440 intervalos = 4 intervalos de 15min * 24h * 15 días)  
    path="autogluon-demanda-15min", # donde se va a guardar los logs y modelos evaluados  
    target="potencia",  
    eval_metric="MASE",  
)  
  
predictor.fit(  
    train_data,  
    presets="medium_quality", #determina los modelos que se van a evaluar  
    time_limit=600, # limita el tiempo de entrenamiento y prueba de modelos  
)
```

Ejemplo: AutoGLUON sobre series temporales de demanda

```
leaderboard = predictor.leaderboard(test_data)
leaderboard
```

✓ 0.0s

	model	score_test	score_val	pred_time_test	pred_time_val	fit_time_marginal	fit_order
0	WeightedEnsemble	-5.851251	-1.297389	2.203647	2.851679	47.444633	7
1	SeasonalNaive	-5.856748	-1.315411	0.908199	0.893642	3.526597	2
2	Naive	-6.318516	-1.866896	1.196462	1.958036	3.456799	1
3	Theta	-7.297091	-3.089474	29.931154	28.617978	3.253967	6
4	ETS	-44.377992	-54.414212	22.711642	25.418150	3.307052	5
5	DirectTabular	-655.900663	-546.522424	18.544826	17.630248	83.676277	4
6	RecursiveTabular	-1434.693439	-1221.651536	172.406426	178.585447	170.444562	3



MLOps (Machine Learning Operations)



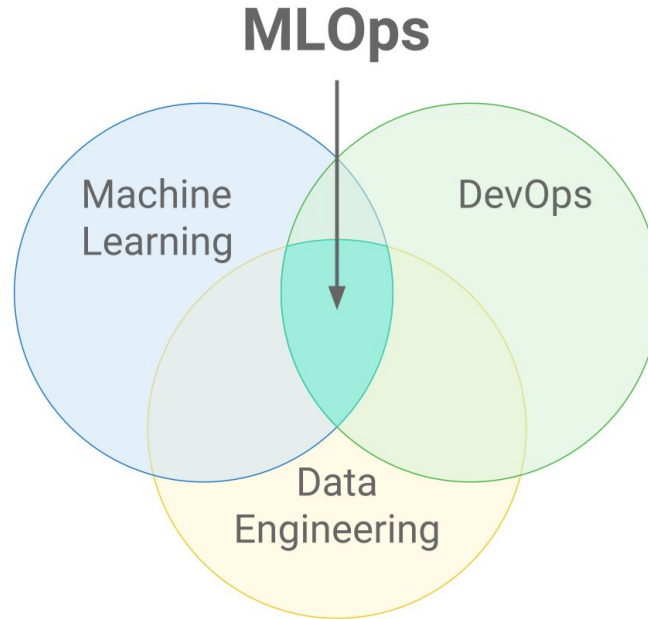
FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

MLOps (Machine Learning Operations)

- Algunos lo llaman ModelOps (model operations or model operationalization)



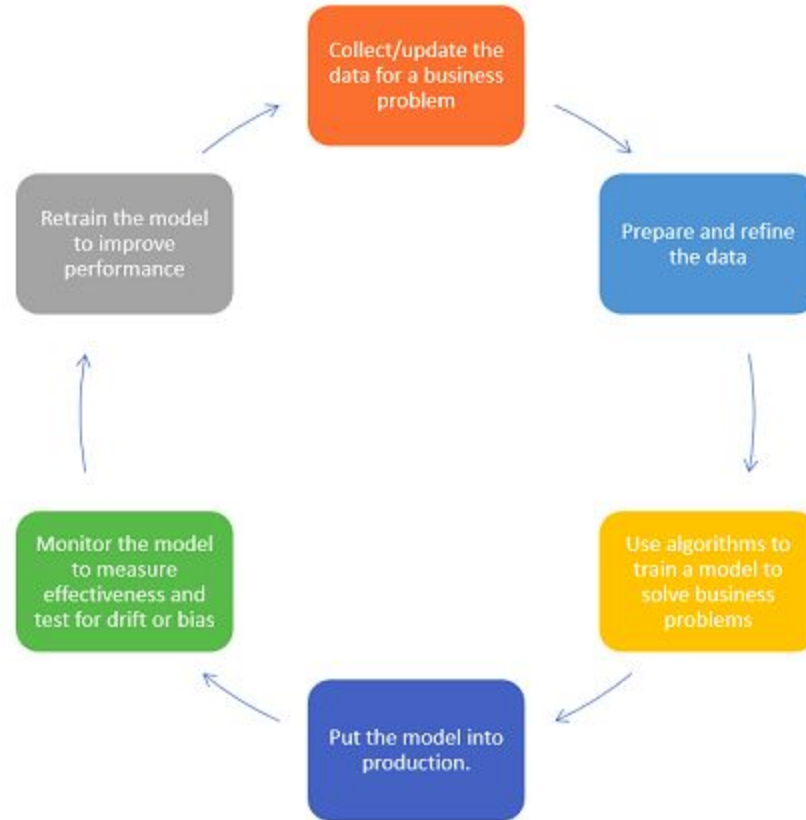
MLOps (Machine Learning Operations)

Integración de la ciencia de datos y el desarrollo de modelos de machine learning en los procesos operativos y de desarrollo de software

Algunos componentes:

- Automatización del flujo de trabajo
 - ciclo de vida completo desde la adquisición y preparación de datos hasta el despliegue y monitorización en producción.

Ciclo completo de vida de proceso de ML



MLOps (Machine Learning Operations)

Integración de la ciencia de datos y el desarrollo de modelos de machine learning en los procesos operativos y de desarrollo de software

Algunos componentes:

- **Automatización del flujo de trabajo**
 - ciclo de vida completo desde la adquisición y preparación de datos hasta el despliegue y monitorización en producción.
- **Gestión de versiones y control de código**
 - garantizar la reproducibilidad
- **Despliegue de modelos en entornos de producción**
 - gestión de dependencias, escalabilidad, monitoreo continuo del rendimiento
- **Infraestructura escalable y reproducible**
 - servidores, contenedores y orquestadores de contenedores
- **Monitorización y mantenimiento**
 - rendimiento de los modelos en producción, detección de problemas, ajustes o actualizaciones a los modelos
- **Gobierno y cumplimiento normativo**
 - gobierno de datos y aseguramiento de la calidad, garantizar la transparencia, la ética y el cumplimiento normativo

Algunas herramientas para implementar MLOps

- MLFlow <https://mlflow.org/>
- TFX - Tensorflow Extended <https://www.tensorflow.org/tfx>

Características generales de las herramientas de MLOps

- Ofrecen módulos que permiten implementar los pasos de proceso
- Pasos se organizan en un grafo dirigido sin ciclos (DAG)
- Capacidad de guardar la información completa de los modelos
 - Datos con los que fue entrenado
 - Partición de datos usada
 - Ambiente de software en el que fue creado
 - Hiperparámetros del modelo
 - Scores de entrenamiento, validación y test
 - Código de software usado
- Histórico de modelos, comparación.
- Facilidad de organizar el despliegue en producción en distintas plataformas

TFX



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

TFX CONFIG

AIRFLOW RUNTIME

KUBEFLOW RUNTIME

OTHER

TensorFlow Extended



TRAINING & EVAL DATA

ExampleGen

StatisticsGen

SchemaGen

Tuner

Transform

Trainer

Evaluator

Example Validator

Infra Validator

Pusher

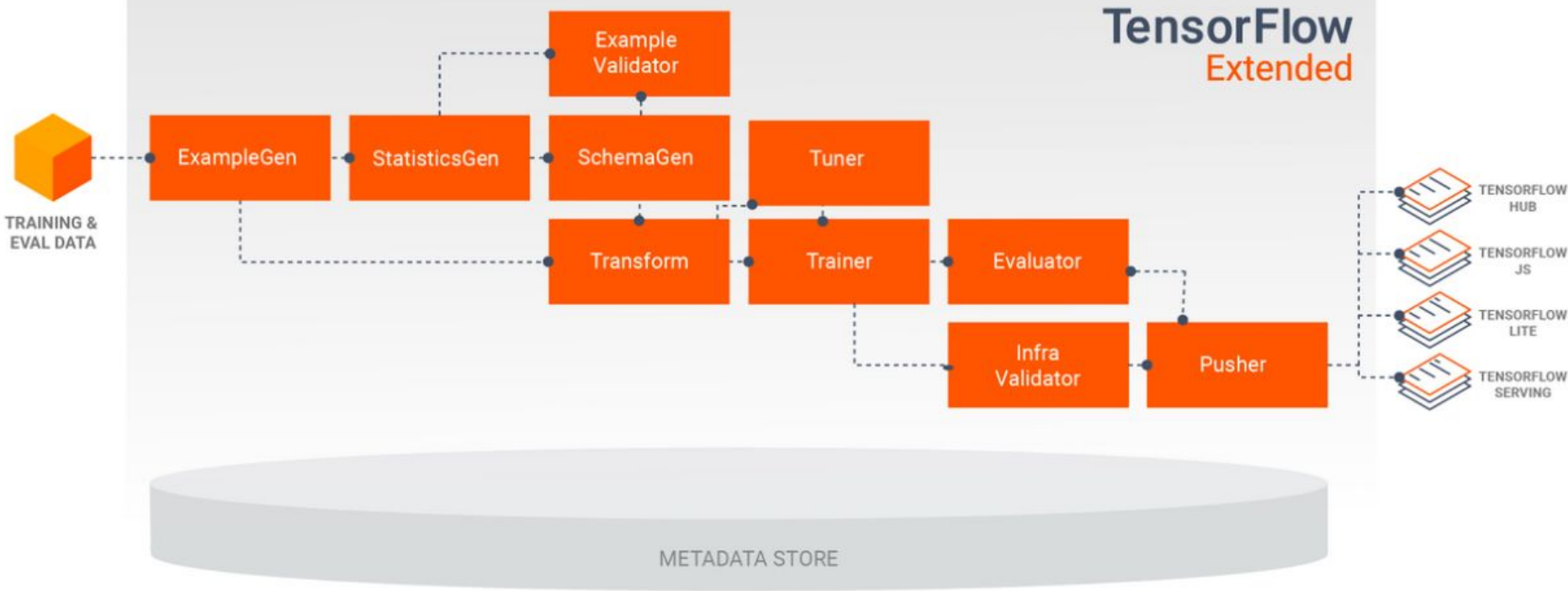
TENSORFLOW HUB

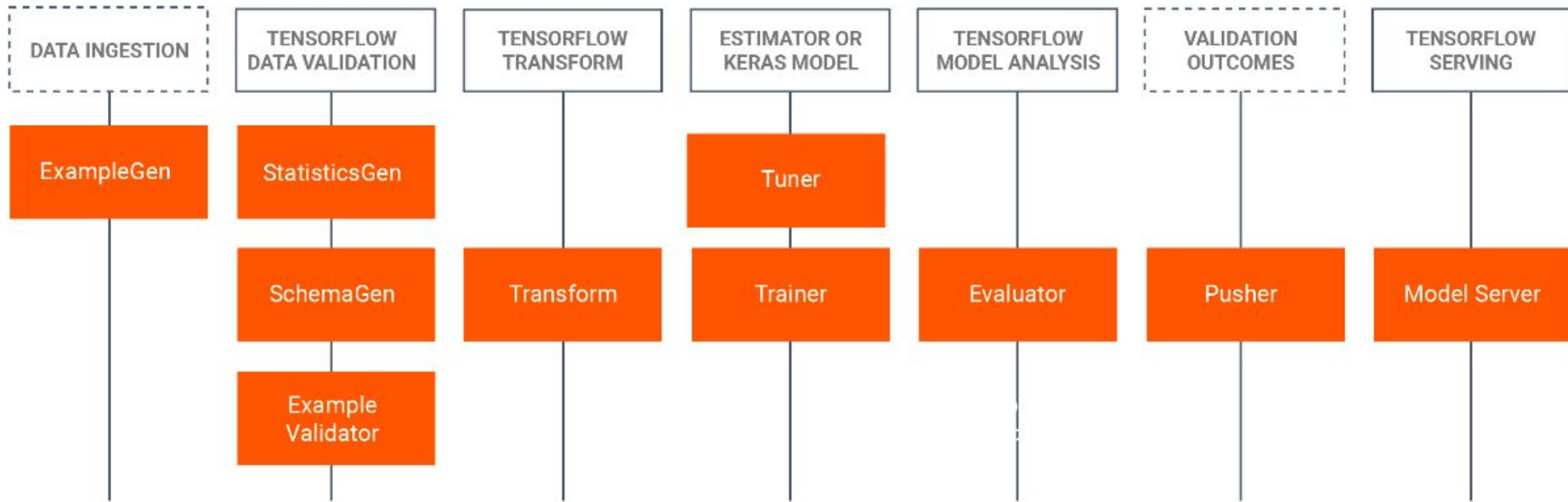
TENSORFLOW JS

TENSORFLOW LITE

TENSORFLOW SERVING

METADATA STORE





MLFlow



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Componentes principales

- **MLflow Tracking:** Permite el seguimiento de experimentos, parámetros, métricas y resultados de modelos de machine learning. Esto facilita la comparación y la reproducción de modelos, así como la colaboración entre equipos.
- **MLflow Projects:** Proporciona un formato estándar para organizar el código, los datos y los entornos de ejecución de los modelos de machine learning. Esto simplifica la reproducción y la implementación de modelos en diferentes entornos.
- **MLflow Models:** Proporciona un formato estándar para empaquetar modelos de machine learning en contenedores o en otros formatos compatibles con diferentes herramientas de implementación. Esto facilita la integración y la implementación de modelos en sistemas de producción.
- **MLflow Registry:** Proporciona un repositorio centralizado para el almacenamiento, la gestión y el seguimiento de modelos de machine learning. Esto facilita la colaboración, el control de versiones y la implementación de modelos en entornos de producción.

MLFlow Recipes

- Marco experimental de MLFlow que permite desarrollar rápidamente modelos de alta calidad e implementarlos en producción.
- Idea:
 - Los problemas típicos de ML son similares -> Se pueden definir de forma similar a través de una receta
 - Recetas predefinidas para clasificación y regresión
- Receta de clasificación:
ingest -> split -> transform -> train -> evaluate -> register

<https://github.com/mlflow/recipes-classification-template>

<https://github.com/mlflow/recipes-examples>

Ejemplo de clasificación

<https://github.com/mlflow/recipes-examples/tree/main/classification>

MLflow Classification Recipe Notebook

This notebook runs the MLflow Classification Recipe on Databricks and inspects its results. For more information about the MLflow Classification Recipe, including usage examples, see the [Classification Recipe overview documentation](#) the [Classification Recipe API documentation](#).

```
from mlflow.recipes import Recipe  
  
r = Recipe(profile="local")
```

✓ 0.7s

P

Experiments

Search Experiments

Default

sklearn_classification_experiment

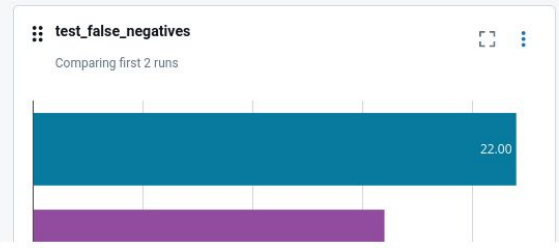
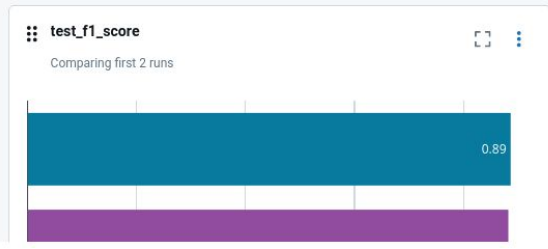
sklearn_classification_experiment [Provide Feedback](#) [Add Description](#)

Share

Q metrics.rmse < 1 and params.model = "tree" ⌚ Time created ▾ State: Active ▾ Datasets ▾ Sort: Created ▾ Group by ▾ ⋮ ↻ + New run

Table **Chart** Evaluation **Experimental**

<input type="checkbox"/>	<input type="checkbox"/>	Run Name
<input type="checkbox"/>	<input type="checkbox"/>	orderly-snail-959
<input type="checkbox"/>	<input type="checkbox"/>	caring-carp-872



14 matching runs



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY