

# **Práctica Calificada**

## **Objetivo del entregable:**

Crear un pipeline de entrenamiento para una BD.

BD: <https://www.kaggle.com/competitions/playground-series-s4e6>

Tipo de modelo: Clasificación

Entrega grupal: Se puede entregar en grupos de 3 personas como máximo.

Inscribirse en la competición de Kaggle con el siguiente prefijo: dmc\_

Ejemplo de nombres: "dmc\_grupo1", "dmc\_jose", etc.

## **1. Crea la clase MLSystem para entrenar un modelo con PyCaret [7 pts]**

- Debe generar como mínimo el archivo: automl.py

## **2. Crear pruebas de testeo para la clase creada (Elija uno o más de los tipos de testeo enseñados en clase) [6 pts]**

- Debe generar como mínimo el archivo: test.py

## **3. Crea los componentes necesarios para un workflow en Apache Airflow [7 pts]**

### **Componentes:**

- **GetDataKaggle**
- **AutoML\_PyCaret**
- **SubmitKaggle**
  - Adjuntar foto del pipeline en ApacheAirflow con los nombres de los componentes anteriores.
  - Adjuntar foto del nombre de grupo creado en Kaggle.
  - Debe generar los archivos: dmc\_pipeline.py, requirements.txt, Dockerfile

### Requisitos de entrega:

El trabajo debe ser entregado como un repositorio en GitHub que contenga los archivos solicitados. (Los archivos mencionados anteriormente son lo mínimo necesario, puede agregar más archivos en caso sea necesario).

Debe adjuntar las dos imágenes solicitadas en el punto 3.

### Evaluación:

La evaluación se centrará en la correcta lógica de los archivos y correcto funcionamiento del pipeline en Apache Airflow.

### Entrega:

Subir imágenes y link de repositorio en la plataforma DMC.

Fecha límite de entrega: **04/06/2024**. No se aceptarán entregas después de esta fecha. Entregas posteriores se penalizará con una deducción de 1 punto por cada día de retraso.

### Puntos extra:

- **Opción de canje:** +7 puntos adicionales en participaciones o +2 puntos en la práctica.
- **Requisitos para Puntos Extra:** De los grupos que subieron su solución en la competencia de Kaggle (el nombre de grupo debe tener el prefijo 'dmc\_'). De todos los grupos, los 3 que estén en mejor puntuación dentro del "leaderboard" de Kaggle podrán acceder a los puntos extra.

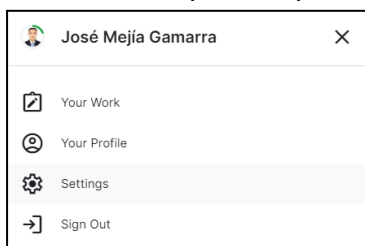
### Anexos:

**Paso 0:** Instalar librería para API Kaggle

```
pip install kaggle
```

**Paso 1:** Unirse a la competición de Kaggle

**Paso 2:** En la parte superior derecha selecciona la opción 'Settings'



### Paso 3: Dar clic en el botón 'Create New Token'


Settings

AccountNotifications

ThemeBETA

Check out our new beta dark mode by choosing your Kaggle UI theme below. [Please provide any feedback here](#)

THEME

 Light theme

API

Using Kaggle's beta API, you can interact with Competitions and Datasets to download data, make submissions, and more via the command line. [Read the docs](#)

Create New Token

Expire Token

#### Paso 4: Abrir el archivo kaggle.json descargado y apuntar los datos de 'username' y 'key'

### Paso 5: Definir variables de entorno y líneas de código para obtener la BD

```
import os
os.environ['KAGGLE_USERNAME'] = 'josmejagamarra'
os.environ['KAGGLE_KEY'] = '945a4c9a6kjda kjdsakjdsabc9442c646163a17'

from kaggle.api.kaggle_api_extended import KaggleApi
import zipfile

api = KaggleApi()
api.authenticate()

# Download the competition files
competition_name = 'playground-series-s4e4'
download_path = 'data/'
api.competition_download_files(competition_name, path=download_path)

# Unzip the downloaded files
for item in os.listdir(download_path):
    if item.endswith('.zip'):
        zip_ref = zipfile.ZipFile(os.path.join(download_path, item), 'r')
        zip_ref.extractall(download_path)
        zip_ref.close()
        print(f"Unzipped {item}")
```

## Paso 6: Entrenar con AutoML

### Paso 7: Subir archivo respuesta a Kaggle.

[illegible]