Prueba Técnica Ingeniero de Machine Learning

**Punto 1. Despliegue de modelo**

**Introducción:** En esta prueba, demostrarás las habilidades técnicas al desplegar un modelo de machine learning como un servicio web. El objetivo es construir una API que interactúe con el modelo y ofrecer un despliegue eficiente y seguro, siguiendo buenas prácticas de desarrollo y documentación. Se le proporcionará un modelo entrenado de clasificación de imágenes, y su tarea es desplegarlo como un servicio web.

**Tareas:**

1. **Crear una API RESTful** usando Flask, FastAPI o Django (o de su preferencia), para exponer el modelo y permitir que reciba solicitudes de predicción. La API debe recibir los datos necesarios y devolver los resultados de la predicción de manera clara y precisa.
2. **Contenerizar la aplicación** usando Docker. Esto debe facilitar el despliegue, haciendo uso de una imagen Docker que contenga todas las dependencias y configuraciones necesarias.
3. **Escribir un Dockerfile eficiente y seguro**, siguiendo las mejores prácticas. Esto incluye optimizar el tamaño de la imagen, mantener un entorno de ejecución seguro y garantizar un proceso de construcción claro y reproducible.
4. **Proporcionar documentación básica** sobre cómo utilizar el servicio. Esta documentación debe detallar los endpoints disponibles, ejemplos de solicitudes (requests), los parámetros requeridos, el formato de las respuestas (responses), y una estimación de la latencia de la API.

**Opcional (para puntaje adicional):**

1. **Implementar pruebas unitarias** para asegurar el correcto funcionamiento de la API y su manejo de casos de prueba comunes.
2. **Configurar el despliegue en la nube** usando un servicio como Google Cloud Platform (GCP), Azure, o Snowflake, con instrucciones claras sobre cómo acceder al servicio desplegado.
3. **Implementar autenticación básica** en la API para protegerla, garantizando que solo usuarios autorizados puedan acceder a los endpoints.

Utilice la siguiente ayuda para leer el código y realizar la predicción:

import io

import numpy as np

import pickle

import base64

from PIL import Image

with open("clf.pickle", "rb") as f:

    clf = pickle.load(f)

with open("digit2.jpg", "rb") as image\_file:

    encoded\_string = base64.b64encode(image\_file.read())

img = Image.open(io.BytesIO(base64.decodebytes(encoded\_string)))

number = np.round((np.array(img)/255)\*16)

clf.predict(number.reshape(1,-1))

La salida es: array ([8])

**La solicitud de request hacía la API será:**

{

"request\_id": "uuid",

"modelo": "clf.pickle",

"image": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/wAALCAAIAAgBAREA/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/9oACAEBAAA/AOYitTJ8BgJYkn8tZJ4maMLFEpnC5EhXJnBVh5YYZR884Ar/2Q==",

"auth": "definir\_authentication"

}

Donde:

* request\_id: Numero de ID del request
* modelo: Nombre del modelo, en este caso ya está dado.
* Image: Imagen en formato base64, ya dada en el código.
* Auth: Autenticación (opcional)

**Punto 2. Manejo de Drift y Re-entrenamiento**

**Descripción:** Imagina un escenario en el cual un modelo de machine learning en producción empieza a perder precisión debido a cambios en los datos de entrada (data drift). La meta es diseñar una estrategia para detectar y abordar este problema.

**Tareas:**

1. **Detectar el Drift:** Propón un enfoque para detectar de manera proactiva el drift en los datos de entrada y/o en las predicciones del modelo. Describe qué métricas y métodos utilizarías para identificar estos cambios en los datos.
2. **Estrategia de Re-entrenamiento:** Diseña una estrategia para re-entrenar el modelo de forma automática una vez que se detecte el drift. Explica qué factores considerarías al decidir cuándo y cómo re-entrenar el modelo.
3. **Pipeline Automatizado:** Presenta una propuesta para un pipeline automatizado que permita:
   * Recolectar nuevos datos relevantes para el re-entrenamiento.
   * Re-entrenar el modelo de manera eficiente.
   * Desplegar la nueva versión del modelo sin interrumpir el servicio actual.

Al desarrollar tu propuesta, considera los desafíos técnicos y operativos que podrían surgir y cómo los abordarías.

Nota: Para este punto 2 no es necesario implementar la solución, se solicita una propuesta que aborde los puntos anteriormente presentados. Es opcional si quiere implementar código, se dará bonificación extra pero no es obligatorio en la prueba.