Consigna técnica – ML Engineer

Entrega

Puede ser en un repositorio o un comprimido .zip

Contexto

Con el archivo CSV adjunto con datos de ventas, país y fecha (la fecha está en formato argentino DD/MM/YYYY), tu tarea es resolver los siguientes pasos:

A) Función de agregación

Implementar una función en Python que:
- Pueda calcular el promedio, la mediana o el máximo de las ventas.
- Sea posible parametrizar para que opere sobre uno o varios países (ejemplo: solo AR, solo CL, o AR+CL+UY).
- Sea posible parametrizar para que opere sobre un año específico tomado de la columna de fecha.
- Permita devolver el resultado de forma global o separado por país.
- En caso de no haber datos luego del filtrado, debe dar un error legible.

Ejemplo de uso (orientativo):
“Imaginá que necesitamos una función que pueda calcular tanto el promedio como el valor máximo de ventas y que sea posible adaptarlo rápidamente al territorio AR, UY, CL o todos. También debe adaptarse a distintos años.”

B) API

Crear una API sencilla que:
- Cargue el CSV una sola vez al iniciar.
- Exponga un único endpoint donde se pueda enviar la operación a realizar, el o los países, el año y si se quiere resultado global o por país.
- Devuelva la respuesta en formato JSON.
- En caso de error (filtros sin datos, parámetros inválidos), devuelva un mensaje claro.

C) Docker y documentación

- Incluir un Dockerfile que permita construir y ejecutar la API en un contenedor.
- Incluir un archivo de dependencias con lo necesario para correr el proyecto.
- Incluir un README con instrucciones claras y concisas sobre:
  - Cómo ejecutar el proyecto en forma local.

- Cómo construir y correr la imagen en Docker.
- Un ejemplo de request y respuesta a la API.

D) Tests unitarios

- Proponer los casos de prueba que cubrirías (lista breve en el README).
- Implementar al menos dos tests unitarios que validen:
  - Un caso de funcionamiento correcto (por ejemplo, calcular el promedio de un país en un año válido).
  - Un caso de error (por ejemplo, no hay datos para el país/año elegido).

E) Versionado de Código y buenas prácticas

- Estructurar las carpetas y los archivos optimizando el uso de los repositorios.
- Buenas prácticas de versionado de código

F) Preguntas conceptuales

- Experiencia previa en nuestro stack tecnológico (Snowflake, Airflow, DBT, AWS, Databricks, CI/CD en gitlab) o similares. Breve resumen de algún proyecto en el que hayas aplicado estas herramientas.
- ¿Cuál te parece la mejor estrategia para versionar y guardar datos y modelos en Databricks?
- ¿Qué estrategias usarías para optimizar costos en clústeres?
- ¿Cómo implementarías seguridad (secret scopes, roles)?
- ¿Cómo harías troubleshooting si un job tarda 3 horas en vez de 30 min?
- ¿Para qué sirve un Dockerfile y qué produce cuando se construye?
- ¿Qué diferencias hay entre un request GET y uno POST trabajando con una API?
- ¿Qué buenas prácticas seguirías para trabajar en equipo usando herramientas de versionado?