Tarea #1 - Big Data

Objetivo

Introducir a los estudiantes al uso de operaciones de Apache Spark para cargar e integrar datos a través del uso de pytest.

Resultados esperados

Para esta asignación se espera que los estudiantes concluyan dos entregables relacionados:

- Un programa principal que dada la información sobre atletas que retorne el top 5 por país, tanto en total de metros como en promedio de metros por día, independiente del deporte.
- Una serie de pruebas unitarias que permitan corroborar la correctitud de las diferentes funciones internas al programa.
- Ambos deben ser entregados como un directorio comprimido donde también se encuentra la configuración de Docker para crear el contenedor y ejecutar las pruebas y el programa principal exitosamente. Los/as estudiantes serán responsables de documentar en su entregable cualquier instrucción necesaria para la correcta ejecución.

Entrega: Archivo comprimido con código y PDF en TEC Digital a más tardar el 20 de junio de 2024 a las 11:00 PM

Datos de entrada

Asumiremos que existen 3 entidades cada una con los siguientes atributos:

1. Atleta (Cada fila es un atleta)
    a. Correo electrónico (string; no es necesario validar formato)
    b. Nombre (string)
    c. País (string)
2. Ejercicio-Nadar (Cada fila es una sesión de nado)
    a. Correo electrónico (string; no es necesario validar formato)
    b. Ritmo cardiaco (numerico 0-200)
    c. Distancia total en metros (numérico)
    d. Total de brazadas (numérico)
    e. Total de minutos de actividad (numérico)
    f. Fecha (Formato YYYY-MM-DD)
3. Ejercicio-Correr (Cada fila es una sesión de corrida)

a. Correo electrónico (string; no es necesario validar formato)
b. Ritmo cardiaco (numerico 0-200)
c. Distancia total en metros (numérico)
d. Ganancia de altura en metros (numérico)
e. Total de minutos de actividad (numérico)
f. Fecha (Formato YYYY-MM-DD)

Para la ejecución del programa principal, los estudiantes deberán proveer 3 archivos en formato CSV (separados por comas) con suficientes datos para ejemplificar la correcta funcionalidad. Si los/as estudiantes tienen dudas sobre la cantidad de datos se les insta a compartir versiones preliminares por correo.

Los archivos no deben llevar fila de encabezado y las columnas deben llevar el mismo orden mencionado en cada uno de los 3 apartados anteriores.

Programa principal (20 puntos)

Se espera que los estudiantes entreguen un manual en PDF con las instrucciones para ejecutar el programa principal. Idealmente esto debería realizarse con una simple llamada a "spark-submit programaestudiante.py atleta.csv nadar.csv correr.csv"

Cualquier detalle necesario para la ejecución debe agregarse en este documento. La imposibilidad de ejecución del programa impedirá la obtención de los puntos.

Pruebas esperadas

Para realizar las pruebas unitarias se espera que los estudiantes piensen en las diferentes partes necesarias para conseguir el objetivo final. Éstas podrán arrancar de datos que se **encuentren en memoria**, asumiendo que el código deberá ser suficientemente modular para que el programa principal simplemente llame a funciones reutilizables que son validadas en diferentes pruebas unitarias.

Los estudiantes deberán diseñar sus propias pruebas unitarias, utilizando la discusión en clase como base para guiar su diseño. Para efectos de evaluación se espera que haya suficientes pruebas para probar las diferentes áreas funcionales:

- **Unión de los datos.** El primer paso debe ser unir los 3 conjuntos de datos diferentes. Deberá existir funciones que solamente se encarguen de esta parte. Nótese que la unión de los datos no necesariamente es trivial. Por ejemplo, es posible que se tenga el correo electrónico de un atleta pero todavía no haya hecho ninguna actividad. De manera similar, podría suceder que el mismo atleta corra o nade múltiples veces en un día. **(25 puntos)**

- **Agregaciones parciales.** Con el objetivo final de encontrar los/las atletas con mayor cantidad de logros se pide que se realice código que permita crear dataframes intermedios donde se tiene el total de **metros recorridos por persona**, **por actividad, por país y por día**. Se espera que haya pruebas que arranquen de dataframes intermedios ya construidos (i.e., no empezarán desde tener que unirlos) y revisen la correcta agregación de los datos. **(35 puntos)**

- **Resultados finales.** Los estudiantes deberán crear pruebas que arranquen de las agregaciones ya construidas y pueden retornar el top N de atletas por país, tanto en total de metros como en promedio de metros por día. **(20 puntos)**

Las pruebas deben cubrir casos excepcionales. Tanto el profesor como el asistente se reservan el derecho de agregar pruebas unitarias adicionales en cada apartado para asegurar el correcto funcionamiento.

La nota será completamente derivada de las pruebas unitarias. Deberá ser posible ejecutar las pruebas simplemente al correr el comando pytest en la carpeta que se entrega con el código.

Los/as estudiantes deben tener cuidado de sólo utilizar código Spark para operar sobre los dataframes. **En particular no deben usar código en Pandas.**