Tarea #2 - Big Data

Objetivo

Introducir a los estudiantes al procesamiento de tipos de datos complejos, agregaciones y métricas con Apache Spark.

Resultados esperados

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

- Un programa principal que recibirá como entrada un patrón de archivos tipo JSON que contienen los detalles de viajes realizados por los conductores de **Diber**, una plataforma digital ficticia de transporte de personas. El programa principal generará varios archivos de salida con información procesada de los viajes.
- Una serie de pruebas unitarias que permitan corroborar la correctitud de las diferentes funciones internas al programa.

Entrega: Archivo comprimido con el código y la guía de ejecución en formato PDF en el TEC Digital a más tardar el **lunes 04 de diciembre de 2023 a las 12:00 MD**.

Datos de entrada

Cada archivo de entrada será un objeto JSON que describe todos los viajes realizados por cada conductor(a) de Diber. La expectativa es que cada corrida del programa cargará un número posiblemente grande de estos archivos.

El formato de cada archivo es:

- Un atributo **identificador** numérico que sirve como valor único para identificar a la persona.
- Un nodo **viajes** que contiene la lista de cada uno de los viajes realizados.
- **codigo_postal_origen** y **codigo_postal_destino** representan los distritos de Costa Rica que determinaron el viaje.
- **kilometros**, que representa la cantidad de kilómetros recorridos en un viaje. Puede ser un número decimal.
- **precio_kilometro** que corresponde al precio del viaje específico. Puede ser un número decimal.

El siguiente es un ejemplo del contenido que tendría un archivo de tipo JSON:

```
{
	"identificador":	"78625",
	"viajes": [
		{
			"codigo_postal_origen": "11504",
			"codigo_postal_destino": "11501",
			"kilometros": "2.8",
			"precio_kilometro": "550",
		},
		{
			"codigo_postal_origen": "10101",
			"codigo_postal_destino": "60101",
			"kilometros": "96.3",
			"precio_kilometro": "300"
		}
	]
}
```

Para la ejecución del programa principal, los estudiantes deberán proveer 5 archivos de prueba con al menos 10 viajes diferentes cada uno. Los archivos deben contener JSON válido.

Programa principal (25 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 persona*.json"

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

El producto de la ejecución del programa principal será una serie de archivos de texto:

- total_viajes.csv: contiene 3 columnas que representan 1) el código postal, 2) si es **origen** o **destino** y 3) la cantidad total de viajes para ese código postal como destino u origen.
- total_ingresos.csv: contiene 3 columnas que representan 1) el código postal, 2) si es **origen** o **destino** y 3) la cantidad de dinero generado en ingresos para ese código postal como destino u origen.
- metricas.csv: contiene 2 columnas que representan el tipo de métrica y su valor. En particular deberá generarse las siguientes métricas
    - persona_con_mas_kilometros: identificador de la persona con más kilómetros recorridos
    - persona_con_mas_ingresos: identificador de la persona con más dinero generado

    - percentil_25: si se ordenan todas las personas de menor ingresos a mayor, cuál valor monetario representa el percentil 25
    - percentil_50: si se ordenan todas las personas de menor ingresos a mayor, cuál valor monetario representa el percentil 50
    - percentil_75: si se ordenan todas las personas de menor ingresos a mayor, cuál valor monetario representa el percentil 75
    - codigo_postal_origen_con_mas_ingresos: el número que identifica el código postal de origen que generó la mayor cantidad de ingresos a las personas
    - codigo_postal_destino_con_mas_ingresos: el número que identifica el código postal de destino que generó la mayor cantidad de ingresos a las personas

Pruebas esperadas

Para realizar las pruebas unitarias se espera que los estudiantes piensen en las diferentes partes necesarias para conseguir el objetivo final. Estas deberán arrancar de datos que se encuentren en memoria. En este caso, pueden arrancar de dataframes en los que cada fila es el string del JSON para una caja.

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

- **Total de viajes: 20 puntos**
- **Total de ingresos: 20 puntos**
- **Métricas: 5 puntos cada una**

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.

Se recuerda a los estudiantes que 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.

Resumen de la evaluación:

- 25 pts programa principal
- 20 pts total viajes
- 20 pts total ingresos
- 35 pts métricas (5pts c/u)