Procesamiento de Datos en JupyterHub con PySpark

1. Introducción

Este laboratorio se centró en el uso de JupyterHub para ejecutar tareas de procesamiento de datos con PySpark. Las actividades incluyeron:

1. Exploración y transformación de datos almacenados en un Bucket de S3.
2. Implementación de un programa de *Word Count* utilizando un clúster EMR y visualización de resultados en Hue.

---

2. Exploración de Datos con PySpark en JupyterHub

Descripción General

Se utilizó JupyterHub para interactuar con PySpark y realizar operaciones de análisis en un archivo CSV almacenado en AWS S3. El flujo de trabajo abarcó desde la lectura de datos hasta su almacenamiento tras el procesamiento.

Pasos Realizados

1. **Configuración del Entorno**:
    - Se configuró PySpark en JupyterHub, habilitando la conexión con el Bucket de S3.
    - Se autenticaron las credenciales de AWS para acceder a los datos.
2. **Carga del Dataset**:
    - Se utilizó PySpark para leer un archivo CSV directamente desde el Bucket.
    - Este archivo contenía los datos necesarios para la exploración inicial.
3. **Exploración de Datos**:
    - Se analizaron las primeras filas del dataset para entender su estructura.
    - Se calcularon estadísticas básicas y se identificaron valores faltantes o inconsistencias.
4. **Procesamiento de Datos**:
    - Transformaciones realizadas:
        - Filtrado de registros irrelevantes.
        - Modificación de valores en columnas específicas según los requerimientos.
    - Los datos procesados se guardaron nuevamente en el Bucket S3.

---

3. Implementación de Word Count en un Clúster EMR

Descripción General

Se creó un Notebook en JupyterHub conectado a un clúster EMR. En este, se implementó un programa de *Word Count* que procesó el dataset *gutenberg-small*.

Pasos Realizados

1. **Configuración del Clúster EMR**:
    - El clúster se configuró con JupyterHub como aplicación principal.
    - Se habilitó PySpark como entorno de procesamiento distribuido.
2. **Implementación del Programa**:
    - Se leyó el dataset *gutenberg-small*.
    - Se aplicó un algoritmo para contar la frecuencia de palabras en el texto.
    - Los resultados se guardaron para ser consultados en la herramienta Hue.
3. **Visualización de Resultados**:
    - Se accedió a Hue para visualizar la salida generada por el programa.
    - Los resultados confirmaron que el programa funcionaba correctamente al mostrar el conteo de palabras.

```
('thoroughly', 15)
('themselves', 192)
('them.', 371)
('letter', 312)
('A.', 1456)
('ORIGINALS', 1)
('THEY', 1)
('sum', 59)
('singular', 18)
('let', 414)
('particularly', 46)
('Johnston:--', 1)
('but', 2485)
('_idler_.', 1)
('good', 543)
('work', 154)
('wasting', 4)
('habit.', 3)
('out', 701)
('charge', 153)
('And', 578)
('other', 1267)
("months'", 7)
('back,', 27)
('unkind', 6)
('contrary,', 65)
('eight', 64)
('1864.', 219)
('Not', 75)
```

---

4. Observaciones y Resultados

1. **Exploración de Datos**:
    - La integración entre JupyterHub y PySpark facilitó la manipulación de grandes volúmenes de datos.
    - Se destacó la utilidad de los clústeres EMR para operaciones de procesamiento distribuido.
2. **Word Count**:
    - El análisis del dataset *gutenberg-small* proporcionó información detallada sobre la frecuencia de palabras.
    - Hue fue una herramienta efectiva para inspeccionar y validar los resultados.