Problema 2

Dado un dataset que contiene información sobre los videos de Youtube (https://netsg.cs.sfu.ca/youtubedata/), crear un programa llamado CategoriaDeVideosMenosVista que obtenga cuál es la categoría de videos menos vista de la plataforma Youtube y el número total de visualizaciones que hay en esa categoría. El programa debe recibir dos parámetros de entrada: la carpeta en la que está el dataset y la carpeta en la que se guardará el resultado. En la carpeta donde está el dataset se tienen que descomprimir UNO de los archivos 0222.zip, 0301.zip, etc., que se encuentran en el enlace anterior. Importante: si la persona que hace la actividad dispone de pocos recursos computaciones, entonces se recomienda que únicamente descomprima algún .zip pequeño para que pueda desarrollar el programa. La carpeta de datos de entrada debería quedar como se ve en la Figura 1. Los datos de entrada están en los archivos 0.txt, 1.txt, etc y cada fila contiene la información de un video tabulada con el siguiente formato: id del video de youtube, usuario que subió el video, número de días desde que se subió el video y la fecha en la que obtuvieron los datos, categoría del video, longitud del video, número de visitas del video, puntuación del video, número de puntuaciones del video, número de comentarios del video, y una lista de ids de videos relacionados. Se valorará positivamente la eficiencia del programa, por ejemplo no usar transformaciones innecesarias.

Ejemplo:

Entrada	Salida
... Gadgets & Games ... 30 ... Gadgets & Games ... 10 ... Music ... 90 ... Sports ... 20 ... Music ... 50 ... Gadgets & Games ... 95	Sports;20

Notar que la categoría “Sports” es la que menos visitas tiene: 20 en un único vídeo. “Music” es la que más visitas tiene: 90 en un video + 50 en otro video, es decir, en total 140 visitas, y la categoría “Gadgets & Games” tiene en total 135 visitas obtenidas de 30 + 10 + 95. El programa debe funcionar independientemente del número de categorías, para cualquier cantidad de filas que se pueda llegar a tener, para cualquier cantidad de ficheros e ignorar el log.txt.

2.1 Solución

- Se inicializa programa generando contexto spark
- Se genera programa de lectura de datos a partir de una ruta data, se descromprime archivo zipeado

- Se emplea método wholeTextFiles para lectura de los archivos .txt
- Se ignora el archivo 'log.txt'
- Se procede a realizar un flapMap para separar los datos en líneas por cada uno de los archivos leídos.

```
from pyspark import SparkContext, SparkConf
from zipfile import ZipFile
import os
import sys
import shutil

# Crear un contexto de Spark
conf = SparkConf().setAppName("problema2")
sc = SparkContext(conf=conf)


def read_rdd_from_dataset(dataset_path:str):
    """Lectura de un rdd desde un archivo zip"""

    folder_path = os.path.dirname(dataset_path)
    filename = os.path.basename(dataset_path).split('.')[0]

    # 1. Eliminamos la carpeta data si existe
    shutil.rmtree(os.path.join(folder_path, 'data', filename))

    # descomprimiendo el archivo
    with ZipFile(dataset_path, 'r') as zip_ref:
        zip_ref.extractall(os.path.join(folder_path, 'data'))


    # lectura de los archivos, eliminando el archivo log.txt
    rdd = sc.wholeTextFiles(os.path.join(folder_path,'data', filename))
    rdd = rdd.filter(lambda x: "log.txt" not in x[0])

    # separando por lineas
    rdd_lines = rdd.flatMap(lambda x: x[1].splitlines())

    return rdd_lines
```

- Se separa la información por el separador del archivo 'tabulador'
- Se eligen las columnas de interés, según posición: categoría y nro de visitas

```
    # 4. Lectura manteniendo columnas de interes: categoria y nro de
    visitas
        select_rdd = (
            input_rdd
```

```
            .map(lambda line: line.split("\t"))
            .filter(lambda fields: len(fields) > 5) # quito nulos
            .map(lambda fields: (fields[3], int(fields[5])))
        )
```

- Agrupo los datos para sumarizar el nro de visitas
- Finalmente ordeno y elijo el elemento con menor nro de visitas

```
# 5. Agrupo por categoria y sumo las visitas
    grouped_rdd = select_rdd.reduceByKey(lambda a, b: a + b)

    # 6. Encontrar la categoria con menos visitas -> retorna una lista
con la categoria y el nro de visitas
    min_category = grouped_rdd.takeOrdered(1, key=lambda x: x[1])
```

- Escribimos data en ruta especificada

```
 # 7. Escribir el resultado en un archivo de texto
    writeRddAsText(sc.parallelize(min_category) ,output_path)
```

2.2 Ejecución Programa

Se ejempla comando de ejecución desde el repositorio

```
spark-submit problema2/categoriaDeVideosMenosVista.py
./problema2/dataset/0301.zip ./problema2/output
```

2.3 Evidencia Ejecución