Analisis basico de un archivo .csv con pyspark

Introduccion

He decido escoger el dataset de kaggle de ventas de videojugos (https://www.kaggle.com/datasets/hosammhmdali/video-game-sales-2024?resource=download) donde se muestras las ventas e informacion de videojugos.

Primeros pasos

```
hdoop@namenode:~$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/02/24 21:07:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
Spark context Web UI available at http://namenode:4040
Spark context available as 'sc' (master = local[*], app id = local-1740427631563).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.4
      /_/

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 1.8.0_442)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val df = spark.read.option("header", "true").option("inferSchema", "true").csv("archivos/videogames.csv")
df: org.apache.spark.sql.DataFrame = [img: string, title: string ... 12 more fields]

scala> df.show(5)
+--------------------+--------------------+-------+-------+--------------+--------------+------------+-----------+--------+---
-----+---------+-----------+------------+------------+-----------+
|                 img|               title|console|  genre|     publisher|     developer|critic_score|total_sales|na_sales|jp_
sales|pal_sales|other_sales|release_date|last_update|
+--------------------+--------------------+-------+-------+--------------+--------------+------------+-----------+--------+---
-----+---------+-----------+------------+------------+-----------+
|/games/boxart/ful...|  Grand Theft Auto V|    PS3| Action|Rockstar Games|Rockstar North|         9.4|      20.32|    6.37|
 0.99|     9.85|       3.12|  2013-09-17|        NULL|
|/games/boxart/ful...|  Grand Theft Auto V|    PS4| Action|Rockstar Games|Rockstar North|         9.7|      19.39|    6.06|
  0.6|     9.71|       3.02|  2014-11-18| 2018-01-03|
|/games/boxart/827...|Grand Theft Auto:...|    PS2| Action|Rockstar Games|Rockstar North|         9.6|      16.15|    8.41|
 0.47|     5.49|       1.78|  2002-10-28|        NULL|
|/games/boxart/ful...|  Grand Theft Auto V|   X360| Action|Rockstar Games|Rockstar North|        NULL|      15.86|    9.06|
 0.06|     5.33|       1.42|  2013-09-17|        NULL|
|/games/boxart/ful...|Call of Duty: Bla...|    PS4|Shooter|    Activision|      Treyarch|         8.1|      15.09|    6.18|
 0.41|     6.05|       2.44|  2015-11-06| 2018-01-14|
+--------------------+--------------------+-------+-------+--------------+--------------+------------+-----------+--------+---
-----+---------+-----------+------------+------------+-----------+
only showing top 5 rows
```


Juegos con mas ventas (sumando todas las consolas)

```
scala> val topGames = df.groupBy("title").agg(sum("total_sales").alias("Total_Sales")).orderBy(desc("Total_Sales"))
topGames: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [title: string, Total_Sales: double]

scala> topGames.show(10)
+--------------------+------------------+
|               title|       Total_Sales|
+--------------------+------------------+
|  Grand Theft Auto V|             64.29|
|Call of Duty: Bla...|30.990000000000002|
|Call of Duty: Mod...|             30.71|
|Call of Duty: Bla...|29.590000000000003|
|Call of Duty: Ghosts|28.800000000000004|
|Call of Duty: Bla...|             26.72|
|Call of Duty: Mod...|             25.02|
|           Minecraft|24.009999999999998|
| Grand Theft Auto IV|             22.53|
|Call of Duty: Adv...|             21.78|
+--------------------+------------------+
only showing top 10 rows
```

Consolas con más juegos vendidos (suma de ventas por consola)

```
scala> val topConsoles = df.groupBy("console").agg(sum("total_sales").alias("Total_Sales")).orderBy(desc("Total_Sales"))
topConsoles: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [console: string, Total_Sales: double]

scala>

scala> topConsoles.show(10)
+-------+------------------+
|console|       Total_Sales|
+-------+------------------+
|    PS2|1027.7599999999927|
|   X360| 859.7899999999987|
|    PS3| 839.6999999999981|
|     PS| 546.2499999999987|
|    PS4| 539.9199999999955|
|    Wii|459.43999999999835|
|     DS|458.16999999999416|
|   XOne| 268.9599999999991|
|    PSP|245.29000000000013|
|     XB|232.05000000000058|
+-------+------------------+
only showing top 10 rows
```

Géneros más populares de 2018 (el de más ventas en ese año)

```
scala> import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._

scala> val dfClean = df.withColumn("Year", year(to_date(col("release_date"), "yyyy-MM-dd")))
dfClean: org.apache.spark.sql.DataFrame = [img: string, title: string ... 13 more fields]

scala> val sales2018 = dfClean.filter(col("Year") === 2018).groupBy("genre").agg(sum("total_sales").alias("Total_Sales")).orde
rBy(desc("Total_Sales"))
sales2018: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [genre: string, Total_Sales: double]

scala>

scala> sales2018.show()
+---------------+------------------+
|          genre|       Total_Sales|
+---------------+------------------+
|Action-Adventure| 35.10000000000001|
|         Sports|             27.16|
|        Shooter| 21.17000000000001|
|         Action|17.030000000000022|
|   Role-Playing| 12.14999999999998|
|       Platform|7.6599999999999975|
|         Racing| 6.709999999999998|
|     Simulation| 4.809999999999996|
|       Fighting| 4.409999999999998|
|      Adventure| 2.789999999999997|
|          Music|1.9000000000000001|
|        Sandbox|              1.89|
|           Misc|1.7700000000000005|
|         Puzzle|1.1900000000000002|
|       Strategy|1.1300000000000006|
|          Party|              0.51|
|   Visual Novel|0.3800000000000002|
|            MMO|              0.26|
|     Board Game|              0.01|
+---------------+------------------+
```