3.6:

df\_thongtin = spark.read.option("header", True).option("sep", "\t").csv(path\_thongtin)

data fame của thông tin bằng đọc dữ liệu với spark với dòng tiêu đề ở đầu file ,dữ liệu được phân cách bằng ‘\t’ đọc dữ liệu từ file phim.binhchon.csv dữ liệu lưu trong path\_thong tin

df\_binhchon = df\_binhchon.withColumn("numVotes", col("numVotes").cast("int"))

df\_binhchon đọc file dữ liệu chứa đánh giá phim thay thế hoặc tạo côt numvote với dữ liệu mới

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  | | --- | | .withColumn("numVotes", ...) |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Tạo hoặc thay thế cột "numVotes" với dữ liệu mới. |  |  | | --- | |  | |
| |  | | --- | | col("numVotes") |  |  | | --- | |  | | |  | | --- | | Truy xuất cột "numVotes" hiện tại. |  |  | | --- | |  | |
| |  | | --- | | .cast("int") |  |  | | --- | |  | | Chuyển kiểu dữ liệu của cột này sang **IntegerType** (số nguyên). |

|  |
| --- |
|  |

|  |
| --- |
|  |

df\_binhchon = df\_binhchon.withColumn("averageRating", col("averageRating").cast("float"))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | .withColumn("averageRating", ...) |  |  | | --- | |  | | |  | | --- | | Tạo hoặc ghi đè cột "averageRating" với dữ liệu mới. |  |  | | --- | |  | |
| |  | | --- | | col("averageRating") |  |  | | --- | |  | | |  | | --- | | Truy xuất dữ liệu từ cột "averageRating". |  |  | | --- | |  | |
| |  | | --- | | .cast("float") |  |  | | --- | |  | | Chuyển kiểu dữ liệu sang **FloatType** (số thực dấu chấm động). |

df\_merged = df\_thongtin.join(df\_binhchon, on="tconst")

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | .join(...) |  |  | | --- | |  | | |  | | --- | | Hàm dùng để **gộp hai bảng** dựa trên cột chung |  |  | | --- | |  | |
| |  | | --- | | on="tconst" |  |  | | --- | |  | | Thực hiện gộp hai bảng dựa trên cột **chung tên là tconst** |

filtered = df\_merged.filter(

    (col("titleType") == "tvSeries") & (col("numVotes") > 150000)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | df\_merged |  |  | | --- | |  | | |  | | --- | | DataFrame đã được gộp từ df\_thongtin và df\_binhchon. |  |  | | --- | |  | |
| |  | | --- | | .filter(...) |  |  | | --- | |  | | |  | | --- | | Hàm dùng để **lọc các dòng** thỏa mãn điều kiện nhất định. |  |  | | --- | |  | |
| |  | | --- | | col("titleType") == "tvSeries" |  |  | | --- | |  | | |  | | --- | | Chỉ chọn các dòng mà loại tiêu đề là **phim truyền hình dài tập** (tvSeries). |  |  | | --- | |  | |
| |  | | --- | | col("numVotes") > 150000 |  |  | | --- | |  | | Chỉ lấy các phim có **hơn 150.000 lượt bình chọn**. |

top7 = filtered.orderBy(col("averageRating").desc()).limit(7)

|  |  |  |  |
| --- | --- | --- | --- |
| filtered | |  | | --- | | DataFrame sau khi đã lọc, chỉ còn các phim truyền hình (tvSeries) có hơn 150.000 lượt bình chọn. |  |  | | --- | |  | |
| |  | | --- | | .orderBy(...) |  |  | | --- | |  | | |  | | --- | | Dùng để **sắp xếp** các dòng trong DataFrame. |  |  | | --- | |  | |
| |  | | --- | | col("averageRating").desc() |  |  | | --- | |  | | |  | | --- | | Sắp xếp theo cột averageRating theo **thứ tự giảm dần** (điểm cao → thấp). |  |  | | --- | |  | |
| |  | | --- | | .limit(7) |  |  | | --- | |  | | **Chỉ lấy 7 dòng đầu tiên**, tức là 7 bộ phim có điểm trung bình cao nhất. |

3.7

Col():tham chiếu đến môt cột trong datafame

count\_per\_year = filtered.groupBy("startYear").agg(count("\*").alias("so\_luong\_phim"))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | .groupBy("startYear") |  |  | | --- | |  | | |  | | --- | | Nhóm các dòng lại theo **năm phát sóng** |  |  | | --- | |  | |
| |  | | --- | | .agg(...) |  |  | | --- | |  | | |  | | --- | | Thực hiện các **hàm tổng hợp (aggregate functions)** trên từng nhóm |  |  | | --- | |  | |
| |  | | --- | | count("\*") |  |  | | --- | |  | | |  | | --- | | Đếm số dòng (tức là số phim) trong mỗi nhóm năm |  |  | | --- | |  | |
| |  | | --- | | .alias("so\_luong\_phim") |  |  | | --- | |  | | Đặt tên cho cột kết quả là **"so\_luong\_phim"** (tức là số lượng phim trong năm đó) |

df = df.withColumn("genres", split(col("genres"), ",")) df\_exploded = df.withColumn("genre", explode(col("genres")))

split(...) – Tách thể loại thành mảng

explode(...) – Bùng nổ mỗi phần tử thành dòng riêng

window\_spec = Window.partitionBy("genre").orderBy(col("averageRating").desc())

|  |  |  |  |
| --- | --- | --- | --- |
| Window | |  | | --- | | Là một module trong PySpark để tạo **cửa sổ tính toán** |  |  | | --- | |  | |
| |  | | --- | | partitionBy("genre") |  |  | | --- | |  | | |  | | --- | | Chia dữ liệu thành từng **nhóm theo thể loại phim** (ví dụ: "Drama", "Comedy",...) |  |  | | --- | |  | |
| |  | | --- | | orderBy(col("averageRating").desc()) |  |  | | --- | |  |   12 | Trong mỗi nhóm, **sắp xếp phim giảm dần theo điểm đánh giá** |