BigData

Rapport de TP2

Spark

MOS 3.1

Étudiants :
Morchoisne Dewone
Maïche Ines

Enseignant :
Derrode Stéphane

20 Mars 2022

Sommaire

1 Exercice2

L'objectif de cet exercice est de créer une requête **Spark** sur un jeu de données de films. On dispose de 4 tables csv (ratings, movies, tags, links) contenant des informations relatives à des films. Pour réaliser notre requête nous allons utiliser les 2 tables **ratings** et **movies**. Le schéma de ces tables est le suivant :

— **Ratings** : userId,movieId,rating,timestamp

— **Movies** : movieId,title,genres (la colonne title doit être nettoyé)

L'idée de la requête va être de déterminer pour chaque genre de la base de donnée **Movies** sa note moyenne, sa popularité (nombre de notes) et le meilleur film du genre selon un critère défini. Ce critère pourra être la popularité du film ou sa note moyenne. Pour ce faire, il va donc falloir réaliser une jointure entre les 2 tables et réaliser plusieurs agrégations par films dans un premier temps puis par genre pour récupérer les statistiques associées.

Le lancement de la requête se fait avec :

```
spark-submit --master local[2] movie_request.py --sort=rating --genre=action
```

2 paramètres peuvent donc être indiqués :

— sort : le critère de définition du meilleur film par genre (popularity ou rating).

— L'ensemble des sous genre sélectionnés (action,aventure,...)

La requête retournera un résultat présentant tous les genres contenant le sous genre sélectionné. En effet un film présente en général plusieurs genres (action,aventure / thriller,horreur, ...).

Les détails du fonctionnement du code sont présentés au sein du code lui même (movie_request.py).

Les premières lignes des résultats pour les requêtes suivantes sont :

```
spark-submit --master local[2] movie_request.py --sort=rating --genre=action
```

```
('Action|Horror|Mystery|Sci-Fi', {'genre_rating': 5.0, 'genre_popularity': 1, 'best_genre_movie': 'Galaxy of Terror (Quest) (1981)', 'best_movie_rating': 5.0, 'best_movie_popularity': 1})
('Action|Crime|Drama|Sci-Fi', {'genre_rating': 5.0, 'genre_popularity': 1, 'best_genre_movie': 'Tokyo Tribe (2014)', 'best_movie_rating': 5.0, 'best_movie_popularity': 1})
('Action|Comedy|Drama|Romance', {'genre_rating': 5.0, 'genre_popularity': 1, 'best_genre_movie': 'Love Exposure (Ai No Mukidashi) (2008)', 'best_movie_rating': 5.0, 'best_movie_popularity': 1})
('Action|Adventure|Animation|Crime|Fantasy', {'genre_rating': 4.6, 'genre_popularity': 4, 'best_genre_movie': 'Tekkonkinkreet (Tekkon kinkurito) (2006)', 'best_movie_rating': 4.6, 'best_movie_popularity': 4})
('Action|Adventure|Comedy|Drama|Romance|Thriller', {'genre_rating': 4.5, 'genre_popularity': 3, 'best_genre_movie': 'Stunt Man; The (1980)', 'best_movie_rating': 4.5, 'best_movie_popularity': 3})
('Action|Comedy|Crime|Fantasy|Thriller', {'genre_rating': 4.5, 'genre_popularity': 1, 'best_genre_movie': 'Monday (2000)', 'best_movie_rating': 4.5, 'best_movie_popularity': 1})
('Action|Adventure|Comedy|Drama|Fantasy|Thriller', {'genre_rating': 4.5, 'genre_popularity': 1, 'best_genre_movie': 'Dragonheart 2: A New Beginning (2000)', 'best_movie_rating': 4.5, 'best_movie_popularity': 1})
('Action|Adventure|Drama|Fantasy|Romance|Sci-Fi|Thriller', {'genre_rating': 4.5, 'genre_popularity': 1, 'best_genre_movie': 'Aelita: The Queen of Mars (Aelita) (1924)', 'best_movie_rating': 4.5, 'best_movie_popularity': 1})
('Action|Adventure|Comedy|Fantasy|Sci-Fi|Thriller', {'genre_rating': 4.5, 'genre_popularity': 1, 'best_genre_movie': 'Maximum Ride (2016)', 'best_movie_rating': 4.5, 'best_movie_popularity': 1})
```

FIGURE 1 – Résultats première requête

```
spark-submit --master local[2] movie_request.py --sort=popularity --genre=adventure
```

```
('Action|Adventure|Sci-Fi', {'genre_rating': 3.0, 'genre_popularity': 2361, 'best_genre_movie': 'Star Wars: Episode IV - A New Hope (1977)', 'best_movie_rating': 4.2, 'best_movie_popularity': 251})
('Action|Adventure|Thriller', {'genre_rating': 3.3, 'genre_popularity': 1455, 'best_genre_movie': 'GoldenEye (1995)', 'best_movie_rating': 3.5, 'best_movie_popularity': 132})
('Action|Adventure|Sci-Fi|Thriller', {'genre_rating': 3.1, 'genre_popularity': 1446, 'best_genre_movie': 'Jurassic Park (1993)', 'best_movie_rating': 3.8, 'best_movie_popularity': 238})
('Action|Adventure|Sci-Fi|IMAX', {'genre_rating': 3.2, 'genre_popularity': 650, 'best_genre_movie': 'Avatar (2009)', 'best_movie_rating': 3.6, 'best_movie_popularity': 97})
('Action|Adventure|Fantasy', {'genre_rating': 2.9, 'genre_popularity': 615, 'best_genre_movie': 'Indiana Jones and the Temple of Doom (1984)', 'best_movie_rating': 3.6, 'best_movie_popularity': 108})
('Adventure|Fantasy', {'genre_rating': 3.1, 'genre_popularity': 584, 'best_genre_movie': 'Lord of the Rings: The Fellowship of the Ring; The (2001)', 'best_movie_rating': 4.1, 'best_movie_popularity': 198})
('Adventure|Animation|Children|Comedy|Fantasy', {'genre_rating': 3.3, 'genre_popularity': 574, 'best_genre_movie': 'Toy Story (1995)', 'best_movie_rating': 3.9, 'best_movie_popularity': 215})
```

FIGURE 2 – Résultats seconde requête

Les résultats de la première requête mettent en évidence le fait que les genres qui ont les meilleurs notes n'ont souvent qu'une seule excellente note. Il serait donc bienvenu de rajouter un paramètre à la requête permettant de sélectionner les genres ayant un nombre minimum de notes.