

Homework 2

Traccia

Dato uno o più dataset a libera scelta, implementare una serie di analytics relative a pattern di "filtering" e "summarization" con HIVE, utilizzando una piattaforma di cloud data management (es. Databricks, etc.).

Framework

Per affrontare il compito assegnato, è stata configurata una sessione **Hive SQL** mediante **Databricks** che ci ha permesso di operare efficacemente sui dataset selezionati (che ammontano ad un totale di 9184 righe per il primo dataset e 1338 per il secondo).

Nella documentazione si allegano i relativi file:

- #2 – Homework → notebook Databricks (link: <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1803943783530797/1922731113263695/7027118924330605/latest.html>)

Dataset

Per questa prova è stato rivisitato l'elaborato consegnato per il primo Homework ma implementato su Hive.

Query

Le analisi effettuate sul dataset comprendono le seguenti operazioni:

1. Film Comuni tra i Dataset

Trovare tutti i film presenti in entrambi i dataset:

tramite l'uso dell'operatore **join** sono state selezionate tutte le righe che presentavano lo stesso titolo e lo stesso anno di produzione (dal momento che nei dataset sono presenti anche omonimi o remake).

2. Tutti i Film nei Dataset

Trovare tutti i film presenti nei dataset, ovvero ottenere una tabella che abbia come righe tutti i film presenti in almeno uno dei due dataset:

tramite l'uso dell'operatore di unione **union** sono stati selezionati tutti i film comuni e non dei due dataset, senza ripetizioni.

3. Top 10

Trovare, all'interno della tabella contenente tutti i film presenti nei due dataset, i 10 che hanno ricevuto il punteggio (*rating*) più alto:

Tramite l'uso di vari **join** abbiamo aggiunto una nuova colonna alla tabella creata al punto precedente, prendendo l'informazione relativa al rating da ognuna delle tabelle (qualora il film fosse presente in entrambe, abbiamo calcolato la media dei rating). Dopodiché, è stato fatto un **ordinamento** per prendere quelli con rating più alto.

4. Flop 10

Trovare, all'interno della tabella contenente tutti i film presenti nei due dataset, i 10 che hanno ottenuto il punteggio (*rating*) più basso:

A partire dalla tabella ottenuta al punto precedente, è stato realizzato un **ordinamento** inverso.

5. Film più votato

Trovare, all'interno della tabella contenente tutti i film presenti nei due dataset, il film che è stato votato da più utenti:

Tramite l'uso di vari **join** abbiamo aggiunto una nuova colonna alla tabella, prendendo l'informazione relativa al numero di voters da ognuna delle tabelle (qualora un film fosse presente in entrambi i dataset, abbiamo preso il valore massimo del numero di votanti). Dopodiché, è stato effettuato un **ordinamento** per prendere il film con maggior numero di voti.

6. Film più recensito

Trovare il film recensito da più utenti:

Data la presenza duplicata di alcuni film, ma con un numero di recensioni differenti, ne abbiamo calcolato la **media**. In seguito, è stato fatto un **ordinamento** per prendere quello con numero di recensioni più alto.

7. Rating Ponderato

Trovare, all'interno della tabella contenente tutti i film presenti nei due dataset, i film che abbiano, a parità di numero di votanti, il rating più alto (in modo da non privilegiare nelle classifiche tutti quei film che avevano avuto una votazione di 9, ma con molti voti in meno rispetto a quelli con votazione di 8.5):

Innanzitutto, è stata operata una **join** per ottenere le informazioni relative al numero di voti e il rating di ogni film. In seguito, è stata creata una nuova colonna 'rating ponderato' (calcolato con una formula definita dai noi):

$$weighted_rating = rating * \log_{10} voter$$

Infine, è stato fatto un **ordinamento** per prendere i film con rating ponderato maggiore e, a parità di tale parametro, con rating maggiore.

8. Film per Genere

Trovare quanti film per ogni genere sono presenti in entrambi i dataset:

Usando l'**explode** abbiamo isolato i generi per ogni film (in quanto un film poteva appartenere anche a più categorie). Successivamente abbiamo determinato il **conteggio** del numero di righe per ogni genere da entrambi i dataset e, infine, ne abbiamo calcolato la **somma**.

9. Media voti per Genere

Calcolare il rating medio per ogni genere:

A partire dal dataset comprensivo di tutti i film è stata calcolata la media dei voti per ogni genere, usando un **groupby** per genere seguito dalla **media** per rating.

10. Durata Media di un Film

Calcolare la durata media di un film:

Si è effettuata la conversione della durata dei film in minuti. Dopodiché, tramite l'uso della funzione **avg()** è stata calcolata la media della colonna ('run_length').

11. Durata media per Genere

Calcolare la media della durata dei film per genere:

È stata isolata la colonna relativa al genere con l'**explode**, applicando in seguito un **groupby** per genere. Dopodiché, è stata calcolata la **media** della colonna 'run_length', e **ordinato** il dataframe risultante per valore decrescente di 'avg(run_length)'.

12. Top Film per cadenza decennale

Stilare delle classifiche dei film per cadenza decennale (questa query nasce dalla considerazione che il voto di alcuni film sarebbe potuto essere stato contestualizzato alle tecnologie fruibili del periodo storico considerato):

innanzitutto è stata aggiunta una colonna 'decennio' utilizzando la seguente formula:

$$\text{floor}\left(\frac{\text{year}}{10}\right) * 10$$

Successivamente abbiamo **partizionato** la tabella per 'decennio' ordinando per 'rating'. Infine, abbiamo **filtrato** le prime 5 righe di ciascuna partizione.

13. Top e Mean Decennio

Determinare i valori max e medi di rating per ogni decennio:

a partire dalla vista con la colonna aggiuntiva 'decennio', è stato fatto un **groupby** per decennio e calcolato il **max()** e l'**avg()** del rating per ottenere una vista risultante con tre colonne: 'decennio', 'mean_rating' e 'max_rating'.

14. Attori in più film

Trovare gli attori presenti in più film:

è stata fatta un'**explode** della colonna 'stars' seguita da un **groupby** per la colonna esplosa. Per ogni gruppo creato, relativo ad un singolo attore, è stato conteggiato (**count()**) il numero di righe. Infine, è seguito un **ordinamento** per prendere la top 10.

15. Attori più presenti nei film più apprezzati

Trovare quegli attori che hanno partecipato a più film, tra quelli che hanno avuto le votazioni migliori:

una volta trovati i 10 film migliori **ordinati** per rating, è stata fatta una prima **join** per unire questi dati con la vista delle star esplose, e una seconda per prendere il numero di film fatti. In questo modo sono stati presi tutti gli attori che hanno partecipato ai 10 film più apprezzati; infine, sono stati **ordinati** per valore decrescente di partecipazioni a film.

16. Attori e registi con più collaborazioni

Trovare le coppie attori/registi presenti in più film insieme:

è stata isolata l'informazione relativa al regista di ogni film, fatto l'**explode** per attore, e successivamente un **groupby** per regista e attore in modo da **contare** le righe per ciascun gruppo, escludendo con un **where** le coppie in cui `attore == regista`. Infine, è stato realizzato un **ordinamento** per valore decrescente di numero di film.

17. Parole più ricorrenti nei commenti della Top 10

Trovare le parole più ricorrenti nei film che hanno avuto punteggio maggiore e quindi più positivi:

è stata innanzitutto realizzata una vista contenente gli url delle recensioni dei top 10 film. Per poter ricavare i tweet, è stato necessario accedere all'url tramite codice **python**. Il contenuto di questi ultimi è stato integrato nella vista. Dopodiché, è stata fatta un'**explode** per parole e un **groupby** per parola **contando** le occorrenze di ognuna di esse. Successivamente è stato realizzato un **ordinamento** per valore decrescente di conteggio. Con una **collect** finale è stata realizzata una vista in cui le parole fossero seguite dal proprio conteggio.

18. Parole più ricorrenti nei commenti della Flop 10

Trovare le parole più ricorrenti nei film che hanno avuto votazioni peggiori:

il procedimento è il duale della query precedente, tranne che per un'inversione dell'**ordinamento** per individuare la flop 10 dei film.