Homework 3

Traccia

Traccia III HOMEWORK (consegna giorno dell'esame) - Dato uno o più dataset a libera scelta, implementare l'import e memorizzazione dei dati ed una serie di analytics relative a pattern di "filtering" e "summarization" con un sistema nosql a scelta dello studente.

Framework

Il framework usato per la gestione del compito con un sistema NoSQL, è stato **MongoDB**, opportunamente collegato al nostro ambiente streamlit.

Nella documentazione si allegano i relativi file:

- LoadData.py → script python per il caricamento dei dataset su cluster MongoDB
- Homepage.py → script python per l'esecuzione della dashboard streamlit

N.B. *Si consiglia di seguire quanto scritto nel* ***README.md*** *per poter eseguire tutto*

Dataset

Il dataset usato è quello del primo e del secondo homework. Per il quale si sono seguiti i medesimi passi di analisi delle precedenti consegne; concentrandoci sull'esplorazione e l'analisi dei dati, la loro gestione, e realizzazione delle operazioni descritte di seguito.

Considerazioni

Dal momento che i database NoSQL hanno una struttura libera da schemi fissi e sono consigliati per gestire grandi moli di dati di natura diversa, abbiamo deciso di inserire tutti i film dei nostri due dataset in un'unica collection. L'obiettivo è stato quello di evitare le varie join realizzate nei precedenti homework, dato che quest'operazione si adatta meglio a database di tipo relazionale. Quindi abbiamo inglobato tutti i nostri dati in un database eterogeneo (che abbiamo caricato su MongoDB come '`Data_NoSQL`').

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:

è stata realizzata una pipeline comprensiva di una **`facet`** che permette di dividerla in due flussi:

a. Raggruppamento per titolo e anno di tutti i film del primo dataset
b. Medesimo flusso ma con il secondo dataset

Elaborato di Big Data
Daiana Cipollaro M63001490
Francesco Di Serio M63001442

I due flussi sono stati mergiati per poi operare un conteggio del numero di film (raggruppati per titolo e anno). Infine, sono stati presi solo i film con conteggio >1, indice del fatto che apparivano in entrambi i dataset.

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:

i film sono stati raggruppati per titolo e anno in modo da eliminare tutte le 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:

è stata fatta la media del rating, raggruppando i film per titolo e anno. È seguito un ordinamento per rating e una limit per estrarre i migliori 10.

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 dal flusso descritto nella query precedente, è stato realizzato un **ordinamento** crescente del rating.

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:

È stato effettuato un **raggruppamento** per titolo e anno, prendendo il **`max()`** della colonna 'votes', rappresentativa del numero di voti, e poi è stato fatto un **ordinamento** decrescente e un **limit** per estrarre il film più votato.

6. Film più recensito

Trovare il film recensito da più utenti:

È stato fatto un **raggruppamento** per titolo e anno, prendendo l'**`avg()`** della colonna '`num_reviews`' , sulla quale è stato fatto un **ordinamento** decrescente e un **limit** per estrarre il film più recensito.

7. Film per Genere

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

Usando l'**`unwind`** sono stati isolati i generi per ogni film (in quanto un film poteva appartenere anche a più categorie). È seguito un raggruppamento per titolo e anno e genere in modo da eliminare i film ripetuti, e infine è stato fatto un conteggio dei film per ogni genere con una **`group`**.

8. Media voti per Genere

Calcolare il rating medio per ogni genere:

il flusso eseguito per la pipeline è il medesimo della query precedente, ma anziché fare il conteggio è stata calcolata la **media** dei rating per ogni genere.

9. Durata Media di un Film

Calcolare la durata media di un film:

Tramite l'uso della funzione **`avg()`** è stata calcolata la media della colonna '`run_length`'.

10. Durata media per Genere

Calcolare la media della durata dei film per genere:

Usando l'**`unwind`** è stata isolata la colonna relativa al genere. È seguito un raggruppamento per titolo, anno e genere in modo da eliminare i film ripetuti, e infine è stata calcolata la **media** della durata dei film per ogni genere con una **`group`**.

11. 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 stato fatto raggruppamento per titolo e anno, in modo da calcolare la **media** della colonna '`rating`' e il **`max()`** della colonna '`votes`'. Dopodiché è stato calcolato il rating ponderato con la seguente equazione:

$$weighted\_rating = rating * \log_{10} votes$$

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

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):

per calcolare la media del rating è stato fatto un raggruppamento per titolo e anno. Dopodiché è stata aggiunta una colonna '`decade`' relativa al decennio di appartenenza di ogni film e abbiamo fatto un **raggruppamento** per questa colonna in modo da ottenere i 5 film con rating maggiore per ogni decade.

13. Top e Mean Decennio

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

a partire dal flusso della query precedente, è stato fatto un **`group`** per decennio e calcolato il **`max()`** e l'**`avg()`** del rating rispettivamente per le due richieste.

14. Attori in più film

Trovare gli attori presenti in più film:

è stata fatta un'**`unwind`** della colonna '`stars`' per isolare gli attori per ogni film, seguita da un **`group`** per la colonna esplosa per poter contare il numero di film per ogni attore. Infine, 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:

è stata realizzata una pipeline comprensiva di una **`facet`** i cui flussi:

a. è stata fatta un'**`unwind`** della colonna '`stars`' per isolare gli attori per ogni film, seguita da un **`group`** per la colonna esplosa per poter contare il numero di film per ogni attore.
b. È stata presa la top dieci dei film (*vedere query 3)*, dopodiché è stata fatta l'**`unwind`** per isolare gli attori

È stato applicato un filtro per poter simulare l'operazione di join tra i due flussi appena descritti (rispetto agli attori). È seguito un raggruppamento per attori e un ordinamento decrescente per numero di 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'**`unwind`** per attore, e successivamente un **`match`** per eliminare le coppie in cui regista e attore coincidevano. È, infine, stato fatto un **`group`** in modo da **`contare`** il numero di film per coppia.

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:

per ottimizzare le operazioni si è fatto uso di una collect apposita ('`collect_reviews`' creata opportunamente nel file *LoadData.py*). Dalla quale sono stati estratti i 10 film migliori (*query 3*).

Dopodiché, per isolare ogni parola è stata fatta un'**`unwind`**, seguita da un **filtraggio** per eliminare le stop_words, poi un raggruppamento per titolo, anno, rating e parola per **contare** il numero di parole per ogni film.

Infine, con un **raggruppamento** finale (titolo, anno, rating), è stato realizzato un vettore contenente le 5 parole più presenti.

18. Parole più ricorrenti nei commenti della Flop 10

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

il flusso applicato è il medesimo della query precedente, ma nel **filtraggio** iniziale sono stati presi i film peggiori.