Sistemi Informativi Evoluti e Big Data – A.A. 2025-2026

Homework #1 – MongoDB

Introduzione

L'esercizio richiede di caricare in MongoDB un dataset di recensioni e prodotti, effettuare una valutazione di base dell'apprezzamento espresso dai clienti nei confronti dei prodotti e, sulla base anche di quest'ultimo, effettuare delle analisi avanzate, includendo un semplice sistema di raccomandazione. Il dataset da utilizzare è il seguente (Amazon Fine Food Reviews): https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

Descrizione del compito

1) Caricamento dei dati. Importare in MongoDB il dataset, organizzando i documenti secondo la seguente struttura:

```
{
  "product_id": "...",
  "user_id": "...",
  "profile_name": "...",
  "score": 4,
  "summary": "...",
  "text": "...",
  "time": "...",
  "sentiment": "positive" | "neutral" | "negative"
}
```

Creare una collezione separata per ogni livello di punteggio (`score` da 1 a 5). I record che non presentano testo o hanno testo vuoto (campi `summary` e `text`) vanno inseriti in una collezione "`no_text`".

2) Sentiment analysis. Stimare il "sentiment" degli utenti su ciascun prodotto partando dai campi `summary` e `text` utilizzando un'apposita libreria Python (ad es. VADER), e salvare il valore nel campo `sentiment` (non presente nel dataset originario).

3) Interrogazioni analitiche. Effettuare le seguenti interrogazioni sulle collezioni create in precedenza:

- Trovare il numero di recensioni per ciascun valore di `score` e visualizzarlo in un istogramma
- Trovare i 5 prodotti con la media di `score` più alta
- Trovare i 5 utenti che hanno scritto più recensioni

- Calcolare il numero di recensioni positive, neutre e negative per ogni valore di `score` (usando l'attributo `sentiment`)
- Trovare i prodotti che hanno sia un alto punteggio medio (`score >= 4.5`) che recensioni con almeno il 10% di sentiment negativo

Dopo aver svolto tutte le interrogazioni precedenti, discutere come la scelta di suddividere i record in collezioni distinte per livello di `score` influisca: (a) sulla complessità delle query (ad esempio quando si devono aggregare dati provenienti da più collezioni); (b) sulle prestazioni delle operazioni di lettura e sulle risorse utilizzate (tempi di risposta); (c) sulla manutenibilità e scalabilità del database in caso di crescita del dataset.
Confrontare questa soluzione con quella di mantenere un'unica collezione "reviews" indicizzata sui campi `score`, `product_id` e `sentiment`. Spiegare in quali casi la suddivisione in più collezioni può essere utile (ad esempio per partizionare logicamente o parallelizzare query), e in quali scenari è invece preferibile utilizzare indici per ottimizzare le stesse interrogazioni.

***4) Sistema di raccomandazione*.** Si supponga che un utente abbia recensito positivamente un prodotto con `product_id = P12345`. Creare un algoritmo che suggerisca tre prodotti simili, basandosi su:

- co-occorrenza di utenti (utenti che hanno recensito anche altri prodotti);
- media dei sentiment positivi associati ai prodotti;
- somiglianza lessicale tra i testi delle recensioni (ad es. con `TfidfVectorizer` e `cosine_similarity`).

Classificare il tipo di sistema di raccomandazione implementato (ad es. collaborative filtering, content-based, ibrido).

***5) Analisi avanzata*.** Analizzare l'evoluzione del sentiment nel tempo per i 5 prodotti con la media di `score` più alta:

a) Per ciascuno di questi prodotti, calcolare l'andamento mensile della proporzione di recensioni positive, neutre e negative nel tempo
b) Visualizzare i risultati in un grafico multilinea (una linea per ciascun tipo di sentiment)
c) Identificare automaticamente (tramite script Python o pipeline di aggregazione) eventuali anomalie o cambiamenti significativi nel sentiment (ad esempio un improvviso aumento del sentiment negativo > 20% rispetto al mese precedente)
d) Discutere le possibili cause di tali variazioni e come un sistema di raccomandazione potrebbe adattarsi a questi cambiamenti nel tempo (ad esempio riducendo la raccomandazione di prodotti che mostrano un trend negativo recente)

Dettagli sulla consegna – Predisporre un notebook Python (o un file .py) in cui è riportato lo svolgimento di tutti gli esercizi e un file PDF in cui sono riportati i commenti sui risultati laddove esplicitamente richiesto. Includere tutti i file in uno zip e caricare l'archivio su Moodle.