

Homework #1 - MongoDB

Analisi delle Recensioni Amazon Fine Food

Indice

Indice.....	1
1. Introduzione	2
2. Caricamento dei Dati.....	3
2.1 Acquisizione del Dataset	3
2.2 Struttura dei Documenti.....	3
2.3 Strategia di Partizionamento	3
2.4 Risultati del Caricamento.....	4
3. Sentiment Analysis.....	5
3.1 Scelta della Libreria: VADER.....	5
3.2 Metodologia di Classificazione	5
4. Interrogazioni Analitiche	6
4.1 Query 1: Distribuzione per Score	6
4.2 Query 2: Top 5 Prodotti per Media Score	7
4.3 Query 3: Top 5 Utenti per Numero di Recensioni.....	7
4.4 Query 4: Sentiment per Score	7
4.5 Query 5: Prodotti Controversi	8
4.6 Discussione Architetturale: Collezioni Separate vs. Collezione Unica	8
Impatto sulla Complessità delle Query	8
Impatto sulle Prestazioni	8
Impatto sulla Manutenibilità e Scalabilità	8
5. Sistema di Raccomandazione.....	9
5.1 Componenti del Sistema	9
Componente 1: Collaborative Filtering (40%).....	9
Componente 2: Content-Based Filtering (30%)	9
Componente 3: Sentiment-Based Ranking (30%).....	9
5.2 Formula di Combinazione.....	9
6. Analisi Temporiale del Sentiment	10
6.1 Metodologia	10
6.2 Risultati.....	10
6.3 Possibili Cause di Variazioni nel Sentiment	11
6.4 Adattamento del Sistema di Raccomandazione	11

1. Introduzione

Questo elaborato presenta l'analisi di un dataset di recensioni di prodotti alimentari venduti su Amazon, noto come "Amazon Fine Food Reviews". Il dataset, disponibile su Kaggle, contiene oltre 500.000 recensioni scritte da utenti nel corso di diversi anni, rappresentando una ricca fonte di informazioni sul sentimento dei consumatori e sulle loro preferenze.

L'obiettivo del progetto è stato triplice: (1) progettare e implementare una struttura dati efficiente in MongoDB per la gestione delle recensioni, (2) applicare tecniche di Natural Language Processing per l'analisi del sentimento, e (3) sviluppare un sistema di raccomandazione ibrido in grado di suggerire prodotti simili basandosi su molteplici criteri.

L'elaborato si articola in cinque parti principali: caricamento e organizzazione dei dati, sentiment analysis con VADER, interrogazioni analitiche, implementazione del sistema di raccomandazione, e analisi dell'evoluzione temporale del sentimento.

2. Caricamento dei Dati

2.1 Acquisizione del Dataset

Il dataset è stato scaricato automaticamente da Kaggle utilizzando la libreria **kagglehub**, che permette il download programmatico di dataset direttamente all'interno del notebook. Questa scelta garantisce la riproducibilità dell'esperimento e automatizza il processo di acquisizione dei dati.

2.2 Struttura dei Documenti

Ogni documento inserito in MongoDB segue la struttura specificata dalla consegna, con i seguenti campi:

Campo	Descrizione
product_id	Identificativo univoco del prodotto Amazon (ASIN)
user_id	Identificativo univoco dell'utente che ha scritto la recensione
profile_name	Nome del profilo pubblico dell'utente
score	Punteggio assegnato dall'utente (1-5 stelle)
summary	Titolo/riassunto breve della recensione
text	Testo completo della recensione
time	Timestamp della recensione (convertito in formato datetime)
sentiment	Classificazione del sentiment: "positive", "neutral" o "negative" (calcolato successivamente)

2.3 Strategia di Partizionamento

Come richiesto dalla consegna, i documenti sono stati distribuiti in 6 collezioni separate:

- **score_1 - score_5**: una collezione per ogni livello di punteggio (1-5 stelle)
- **no_text**: collezione dedicata ai record privi di contenuto testuale (summary e text vuoti o assenti)

2.4 Risultati del Caricamento

Il processo di caricamento ha elaborato **568.454 record**, distribuiti come segue:

Collezione	Documenti	Percentuale
score_1	52,268	9.2%
score_2	29,769	5.2%
score_3	42,640	7.5%
score_4	80,655	14.2%
score_5	363,122	63.9%
no_text	0	0%

La distribuzione mostra una forte asimmetria verso i punteggi elevati, con quasi il 64% delle recensioni a 5 stelle. Questo è un pattern comune nei dataset di recensioni online, dove gli utenti tendono a recensire principalmente quando sono molto soddisfatti. La collezione no_text risulta vuota, indicando che tutte le recensioni nel dataset contengono almeno un campo testuale.

3. Sentiment Analysis

3.1 Scelta della Libreria: VADER

Per l'analisi del sentimento è stata utilizzata la libreria **VADER (Valence Aware Dictionary and sEntiment Reasoner)**, sviluppata specificamente per l'analisi di testi brevi come recensioni e post sui social media. VADER presenta diversi vantaggi rispetto ad altre soluzioni:

1. **Ottimizzazione per testi brevi:** a differenza di modelli basati su machine learning che richiedono grandi quantità di dati di training, VADER è basato su un lexicon pre-costruito e regole grammaticali, risultando particolarmente efficace su testi concisi come i riassunti delle recensioni.
2. **Sensibilità al contesto:** VADER riconosce modificatori come negazioni ("non buono"), intensificatori ("molto buono"), e punteggiatura enfatica ("buono!!!"), adattando il punteggio di conseguenza.
3. **Velocità di elaborazione:** non richiedendo training o inferenza su modelli complessi, VADER può processare centinaia di migliaia di documenti in tempi ragionevoli.
4. **Nessuna necessità di training:** il lexicon pre-costruito e validato empiricamente evita il rischio di overfitting su dati specifici del dominio.

3.2 Metodologia di Classificazione

VADER produce quattro punteggi per ogni testo analizzato: *positive*, *negative*, *neutral* (proporzioni normalizzate) e *compound* (punteggio aggregato da -1 a +1). La classificazione finale utilizza il compound score con le seguenti soglie standard:

- **Positive:** compound ≥ 0.05
- **Negative:** compound ≤ -0.05
- **Neutral:** $-0.05 < \text{compound} < 0.05$

Per ogni documento, l'analisi è stata effettuata concatenando i campi *summary* e *text*, in modo da catturare il sentimento complessivo della recensione.

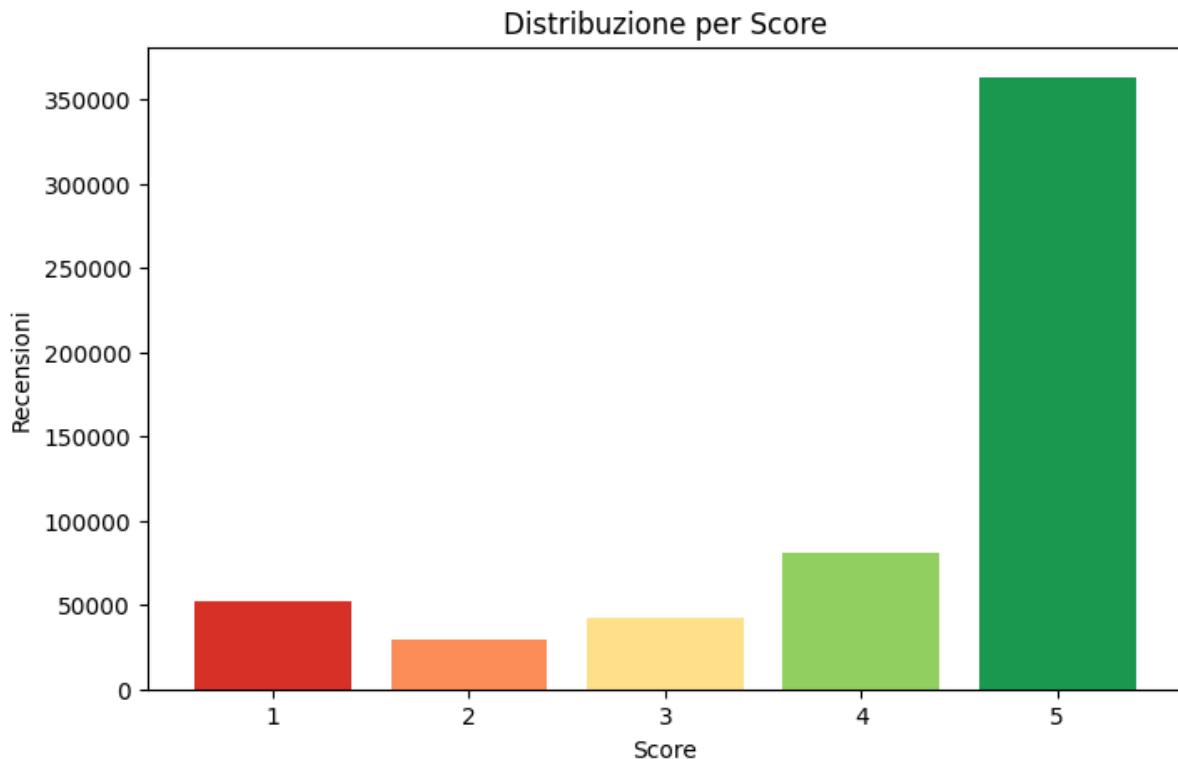
Sentiment	Documenti
Positive	507,424 (89.3%)
Negative	53,875 (9.5%)
Neutral	7,155 (1.2%)

La forte predominanza di sentimento positivo (89.3%) è coerente con la distribuzione degli score, dove il 78% delle recensioni ha punteggio 4 o 5 stelle. Tuttavia, è interessante notare che la percentuale di sentimento negativo (9.5%) è leggermente superiore alla percentuale di recensioni a 1 stella (9.2%), suggerendo che alcune recensioni con score medio-alto contengono comunque elementi critici nel testo.

4. Interrogazioni Analitiche

4.1 Query 1: Distribuzione per Score

La prima query calcola il numero di recensioni per ciascun valore di score, visualizzando i risultati in un istogramma. L'implementazione sfrutta il partizionamento in collezioni separate, permettendo di ottenere il conteggio tramite una semplice operazione `count_documents({})` su ciascuna collezione.



L'istogramma conferma la distribuzione J-shaped tipica delle recensioni online, con una forte concentrazione sui punteggi estremi (soprattutto 5 stelle) e minore rappresentazione dei punteggi intermedi. Questo pattern è ben documentato nella letteratura sull'e-commerce.

4.2 Query 2: Top 5 Prodotti per Media Score

Per identificare i prodotti con la migliore valutazione media, è stata implementata un'aggregazione cross-collection che:

- | | |
|--|-------------------------------|
| 1. Scorre tutte le collezioni score_1 - score_5 | 1. B005EL6VOY - 5.00 (33 rec) |
| 2. Accumula punteggi e conteggi per ciascun product_id | 2. B000RL53PE - 5.00 (33 rec) |
| 3. Calcola la media solo per prodotti con almeno 5 recensioni (filtro per evitare bias) | 3. B001IZ9ME6 - 5.00 (31 rec) |
| 4. Ordina per media decrescente, usando il numero di recensioni come criterio secondario | 4. B002900XGQ - 5.00 (30 rec) |
| | 5. B00376ZEY6 - 5.00 (30 rec) |

4.3 Query 3: Top 5 Utenti per Numero di Recensioni

Questa query identifica gli utenti più attivi, aggregando il conteggio delle recensioni per user_id su tutte le collezioni. I risultati mostrano che alcuni "super-reviewer" hanno

- | | |
|-----------------------------|---|
| 1. A30XHLG6DIBRW8 - 448 rec | scritto centinaia di recensioni, |
| 2. A1YUL9PCJR3JTY - 421 rec | rappresentando una piccola ma influente |
| 3. AY12DBB0U420B - 389 rec | porzione della community. |
| 4. A281NPSIMI1C2R - 365 rec | |
| 5. A1Z54EM24Y40LL - 256 rec | |

4.4 Query 4: Sentiment per Score

Per ogni collezione score_X, viene eseguita una pipeline di aggregazione con operatore \$group sul campo sentiment. I risultati rivelano pattern interessanti:

- **Score 5:** predominanza quasi assoluta di sentiment positivo, come atteso
- **Score 3-4:** mix di sentiment, con presenza significativa di positivi anche negli score medi
- **Score 1-2:** predominanza di sentiment negativo, ma con sorprendente presenza di positivi (circa 45-50%)

La discrepanza tra score numerico e sentiment testuale suggerisce che alcuni utenti assegnano punteggi bassi pur esprimendo apprezzamento nel testo (es. "Buon prodotto ma troppo costoso"), mentre altri danno punteggi alti nonostante critiche ("Ottimo sapore ma confezione danneggiata").

4.5 Query 5: Prodotti Controversi

Questa query identifica i prodotti "controversi": quelli con alto punteggio medio (≥ 4.5) ma con almeno il 10% di sentimento negativo nelle recensioni (almeno 10 recensioni considerate). Questi prodotti meritano attenzione particolare perché il punteggio numerico elevato potrebbe mascherare problemi sottostanti evidenziati nei testi.

Trovati 460 prodotti controversi:

1. B000FBMFDO
Media: 4.62, Negative: 62.9%, Reviews: 97
2. B000EP0BYC
Media: 4.54, Negative: 30.8%, Reviews: 13
3. B001CD0B40
Media: 5.00, Negative: 30.8%, Reviews: 13
4. B0007RL9RY
Media: 4.70, Negative: 30.4%, Reviews: 23
5. B001JU7YJA
Media: 4.50, Negative: 30.0%, Reviews: 10

4.6 Discussione Architetturale: Collezioni Separate vs. Collezione Unica

Impatto sulla Complessità delle Query

La suddivisione in collezioni separate per score presenta vantaggi e svantaggi significativi in termini di complessità delle query:

- **Vantaggio:** query su singolo score sono immediate e naturali (es. "tutte le recensioni a 5 stelle" = query sulla collezione score_5)
- **Svantaggio:** aggregazioni cross-score richiedono operatori come \$unionWith o elaborazione applicativa multipla, aumentando la complessità del codice

Impatto sulle Prestazioni

Dal punto di vista delle prestazioni:

- **Collezioni separate:** query selettive su singolo score scansionano meno documenti, con tempi di risposta potenzialmente inferiori. Tuttavia, aggregazioni globali subiscono overhead di coordinamento.
- **Collezione unica indicizzata:** con indici composti (score, product_id, sentiment), le query più complesse possono raggiungere prestazioni superiori.

Impatto sulla Manutenibilità e Scalabilità

- **Collezioni separate:** maggiore complessità operativa (backup, indici, policy da gestire per ogni collezione), ma flessibilità nel distribuire collezioni su nodi diversi.
- **Collezione unica:** gestione centralizzata più semplice.

> Per carichi analitici complessi e dataset in crescita, una collezione unica è generalmente preferibile. La suddivisione in collezioni può essere utile in scenari specifici: partizionamento logico per team diversi, politiche di retention differenziate, o quando le query sono quasi esclusivamente filtrate per score.

5. Sistema di Raccomandazione

Il sistema implementato è di tipo **ibrido**, combinando tre approcci complementari per superare i limiti di ciascuna tecnica individuale e sfruttarne i punti di forza.

5.1 Componenti del Sistema

Componente 1: Collaborative Filtering (User-Based) (40%)

Tecnica: Similarità di Jaccard sugli insiemi di utenti

Razionale: Se molti utenti hanno recensito sia il prodotto target che un altro prodotto, è probabile che i due prodotti siano correlati.

Limitazioni: soffre del problema del "cold start" per prodotti nuovi con pochi utenti.

Il collaborative filtering è generalmente il metodo più affidabile nei sistemi di raccomandazione.

Componente 2: Content-Based Filtering (30%)

Tecnica: TF-IDF + Cosine Similarity sui pesi dei testi delle recensioni

Razionale: Prodotti descritti con terminologia simile nelle recensioni (es. "croccante", "biologico", "senza glutine") sono probabilmente simili nelle caratteristiche.

Limitazioni: non cattura relazioni semantiche profonde (limitandosi alle 1000 parole più note della lingua inglese).

Componente 3: Sentiment-Based Ranking (30%)

Tecnica: Proporzione di recensioni con sentimento positivo

Razionale: A parità di altre condizioni, è preferibile raccomandare prodotti con feedback prevalentemente positivo.

Limitazioni: non considera la similarità con il prodotto di partenza, ma solo la qualità assoluta.

5.2 Formula di Combinazione

Il punteggio finale per ciascun prodotto candidato è calcolato come:

$$\text{Score} = 0.4 \times \text{Jaccard} + 0.3 \times \text{Sentiment} + 0.3 \times \text{TF-IDF}$$

6. Analisi Temporale del Sentiment

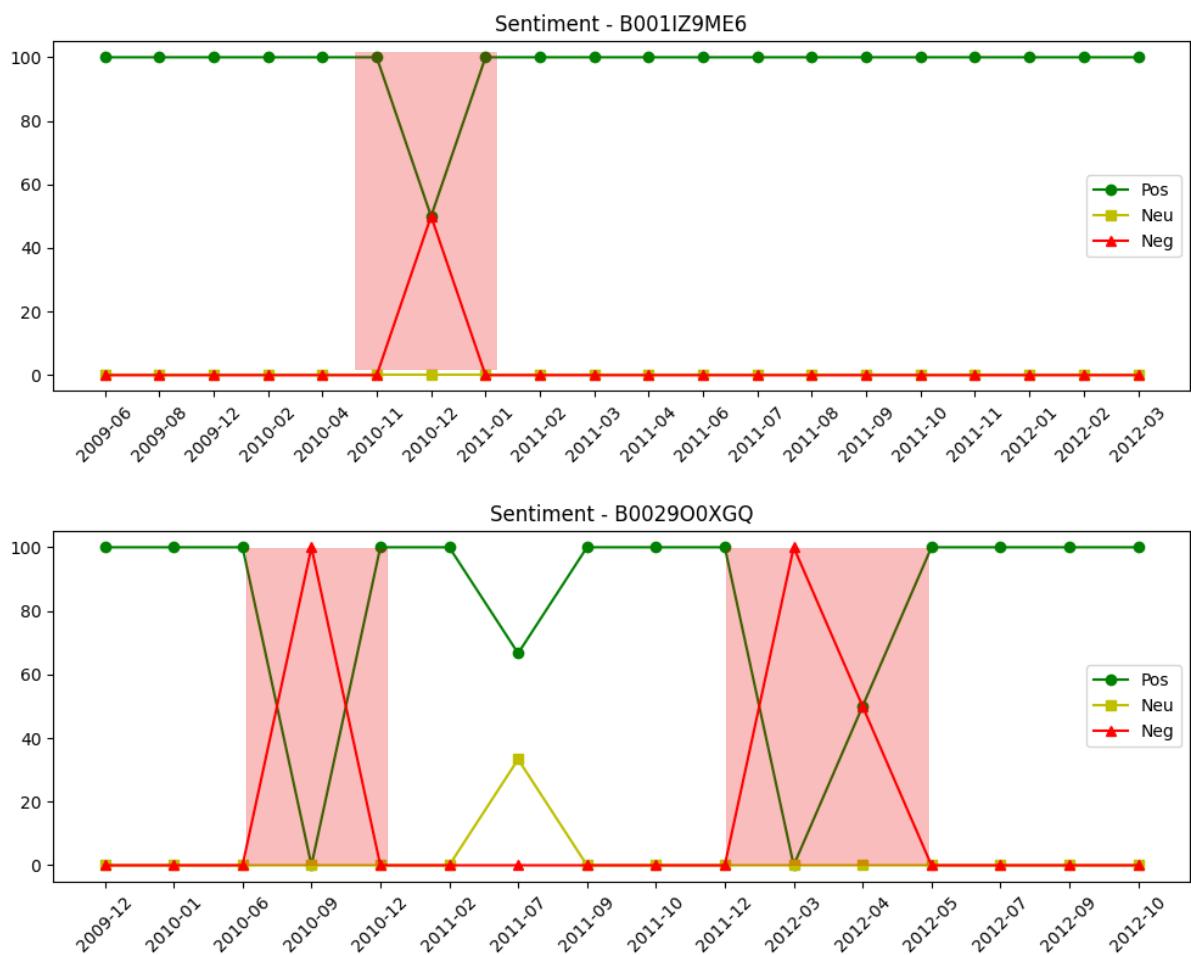
6.1 Metodologia

L'analisi dell'evoluzione temporale del sentiment è stata condotta sui top 5 prodotti per media score identificati nella Parte 4.2, seguendo questi step:

1. **Aggregazione mensile:** per ogni prodotto, le recensioni sono state raggruppate per anno-mese usando il campo time.
2. **Calcolo proporzioni:** per ogni mese, è stata calcolata la percentuale di recensioni positive, neutre e negative.
3. **Visualizzazione:** grafici multilinea con matplotlib per visualizzare i trend.
4. **Anomaly detection:** identificazione automatica di aumenti del sentiment negativo superiori al 20% rispetto al mese precedente.

6.2 Risultati

Per 2/5 prodotti analizzati **sono state rilevate anomalie significative** (aumenti > 20% del sentiment negativo).



6.3 Possibili Cause di Variazioni nel Sentiment

Quando si verificano variazioni significative nel sentiment, le cause possono essere molteplici:

- **Cambiamenti nel prodotto:** modifiche alla formula, alla confezione o al fornitore possono alterare la qualità percepita.
- **Problemi logistici:** ritardi nelle spedizioni, prodotti danneggiati o scaduti durante il trasporto.
- **Fattori stagionali:** alcuni prodotti alimentari possono avere variazioni di qualità stagionali (es. prodotti freschi).
- **Cambiamenti nella base utenti:** espansione a nuovi segmenti di mercato con aspettative diverse.
- **Eventi esterni:** notizie negative sul brand, richiami di prodotto, o trend sui social media.

6.4 Adattamento del Sistema di Raccomandazione

Un sistema di raccomandazione efficace non può basarsi su valutazioni statiche, ma deve essere in grado di adattarsi dinamicamente alle variazioni del sentiment degli utenti. Il sentiment associato a un prodotto può infatti cambiare nel tempo per molteplici ragioni, come modifiche al prodotto stesso (ad esempio variazioni nella formula, nel packaging o nel fornitore), problemi logistici quali ritardi o prodotti danneggiati, fattori stagionali che incidono soprattutto sui prodotti freschi, cambiamenti nella base utenti con aspettative diverse, oppure eventi esterni come notizie negative, richiami ufficiali o trend sui social media.

Per gestire efficacemente queste variazioni, il sistema di raccomandazione può adottare diverse strategie di adattamento.

1. **Decay temporale del sentiment:** pesare maggiormente le recensioni recenti nel calcolo del sentiment score, riducendo l'influenza di recensioni datate;
2. **Penalizzazione trend negativi:** se un prodotto mostra un aumento >20% del sentiment negativo nell'ultimo mese, ridurre il suo score di raccomandazione proporzionalmente;
3. **Alert proattivi:** notificare gli utenti che hanno acquistato prodotti ora in trend negativo, suggerendo alternative.