Università degli Studi di Milano - Bicocca

Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea in Informatica

Progetto di Metodi Informatici per la Gestione Aziendale

Sviluppo di diverse tipologie di Recommendation System

Autori

Matias Maciej Bonoli
Andrea Vasciminno 904899

Anno Accademico 2024–2025

Indice

1 Introduzione

1.1 Obiettivi del Progetto

L 'obiettivo principale è stato quello di analizzare un vasto set di dati di recensioni di prodotti Amazon per implementare e confrontare diverse metodologie di raccomandazione.

Il progetto si è articolato in tre livelli di complessità crescente: **Base**, **Intermedio** e **Avanzato**, ognuno dei quali ha esplorato un aspetto differente delle tecniche di analisi dei dati.

Il progetto base si è concentrato esclusivamente sul *collaborative filtering*, un approccio che sfrutta i dati di interazione utente-prodotto per identificare preferenze e generare raccomandazioni incrociate.

Il progetto intermedio ha integrato questo approccio con il *content-based filtering*, analizzando le caratteristiche testuali dei prodotti per suggerire articoli affini a quelli già apprezzati.

Infine, il progetto avanzato ha aggiunto una componente di *sentiment analysis*, che ha permesso di classificare le recensioni in base al sentimento espresso (positivo, negativo, neutro) per ottenere una comprensione ancora più profonda del feedback degli utenti.

L'analisi è stata condotta sul set di dati **Amazon Reviews 2023**, specificamente sulla categoria *"CDs and Vinyl"*, che ha fornito una base di dati significativa per eseguire analisi complete e per testare l'efficacia dei diversi modelli.

1.2 Metodologia e Strumenti

Il processo di lavoro è iniziato con la preparazione e la pulizia dei dati, essenziale per la qualità delle analisi successive. Successivamente, è stata condotta un'esplorazione approfondita per identificare pattern, correlazioni e caratteristiche specifiche del dataset.

La fase finale ha previsto la valutazione comparativa dei diversi modelli per determinarne l'efficacia.

Per lo sviluppo del progetto sono state utilizzate diverse librerie Python, tra cui:

- **pandas** e **NumPy** per la manipolazione e l'analisi numerica dei dati;
- **scikit-learn** per l'implementazione di algoritmi di machine learning;
- **surprise** per la costruzione di sistemi di raccomandazione collaborativi;

- **NLTK** per il pre-processing del linguaggio naturale;
- **Hugging Face** per l'utilizzo di modelli neurali avanzati.

2 Dataset e Esplorazione

2.1 Informazioni sui dati

Il dataset scelto per questo progetto è Amazon Reviews 2023, una raccolta di recensioni di prodotti Amazon organizzate per categorie merceologiche. Tra le diverse categorie disponibili, è stata selezionata "CDs and Vinyl", che presenta dimensioni significative per l'analisi: 4,8 milioni di recensioni riguardanti 701.000 prodotti, scritte da 1,8 milioni di utenti.

Come specificato nelle istruzioni del progetto, ogni categoria è composta da due dataset distinti: uno contenente le recensioni degli utenti (*user reviews*) e uno con i metadati dei prodotti (*item metadata*).

2.1.1 Dataset delle recensioni (User Reviews)

Dal dataset delle recensioni sono stati estratti i seguenti attributi principali:

- **rating**: la valutazione numerica assegnata dall'utente al prodotto, espressa su una scala da 1 a 5
- **user_id**: identificativo univoco dell'utente che ha scritto la recensione
- **parent_asin**: codice identificativo della famiglia di prodotti a cui appartiene l'articolo recensito

2.1.2 Dataset dei metadati (Item Metadata)

Per quanto riguarda i metadati dei prodotti, sono stati considerati questi campi:

- **title**: il nome commerciale del prodotto
- **description**: la descrizione dettagliata del prodotto
- **parent_asin**: codice identificativo della famiglia di prodotti, utilizzato per collegare i due dataset

L'utilizzo del campo `parent_asin` come chiave di collegamento permette di associare le recensioni ai rispettivi prodotti e di integrare le informazioni provenienti dai due dataset per le analisi successive.

2.2 Struttura dei Dati

2.2.1 User Reviews

Dopo aver caricato il dataset *user reviews*, le prime 5 righe del dataframe sono le seguenti:

	rating	parent_asin	user_id
0	5	B002MW50JA	AGKASBHYZPGTEPO6LWZPVJWB2BVA
1	5	B008XNPN0S	AGKASBHYZPGTEPO6LWZPVJWB2BVA
2	3	B00IKM5N02	AGKASBHYZPGTEPO6LWZPVJWB2BVA
3	3	B00006JKCM	AEVWAM3YWN5URJVJIZZ6XPD2MKIA
4	5	B00013YRQY	AFWHJ6O3PV4JC7PVOJH6CPULO2KQ

Tabella 1: Prime 5 righe del dataset user reviews

Le statistiche descrittive del campo `rating` sono riportate nella tabella seguente:

	rating
count	4.827.273
mean	4,50
std	1,00
min	1,00
max	5,00

Tabella 2: Statistiche descrittive del campo rating

Si nota come la media abbia una forte tendenza verso valutazioni positive; tuttavia, la deviazione standard mostra comunque una certa variabilità nelle valutazioni.

Ulteriori statistiche del dataset includono:

- La media di recensioni per **prodotto** è di 6,88
- La media di recensioni lasciate da ogni **utente** è di 2,75

2.2.2 Item Metadata

Dopo aver caricato il dataset *item metadata*, le prime 5 righe del dataframe sono le seguenti:

N	Title	Description	Parent ASIN
0	Release Some Tension	Swv Release Some Tension	B000002X4C
1	Rio Angie	Shrimp City Slim (aka Gary Erwin, b. 1953, Chicago) has long been associated with the Southeastern blues scene. However, he's always written songs in a variety of styles and 'Rio Angie' touches on that side of the artist. This solo piano CD contains improvisations on eleven original themes from 1971-1980.	B00902T10Y
2	Lost in Love	(Nessuna descrizione)	B00000DALY
3	Somewhere in Time	The 1980 soundtrack to the now classic motion picture starring the late Christopher Reeve and Jane Seymour was composed by veteran composer John Barry and is one of his best loved works. This soundtrack has been digitally remastered.	B0000086D1
4	Kimmon Waldruff	Solo acoustic fingerstyle guitar.	B000S6W7KC

Tabella 3: Dataset musicale - Prime 5 righe

2.3 Pulizia dei dati

Per motivi computazionali e di precisione dei risultati, è stato deciso di ridurre la dimensione del dataset delle recensioni, considerando solamente prodotti con più di 22 recensioni e utenti che hanno lasciato almeno 12 recensioni. In questo modo si vanno a considerare solamente le recensioni che possano avere un senso dal punto di vista temporale e di affidabilità. Difatti, un dataset con milioni di istanze come quello fornitoci potrebbe sembrare ideale per l'analisi dei dati, ma nella realtà dei fatti, invece, tende a richiedere sforzi computazionali enormi. Per questo motivo si

è adottata una riduzione di questo tipo per il dataset. Il dataset risultante presenta le seguenti caratteristiche:

- 112.140 recensioni
- 2.601 prodotti recensiti
- 4.219 utenti che hanno recensito

Le statistiche descrittive del campo `rating` dopo il filtraggio sono riportate nella tabella seguente:

	rating
count	112.140
mean	4,25
std	1,14

Tabella 4: Statistiche descrittive del campo rating dopo il filtraggio

È possibile notare come, rimuovendo gli utenti meno attivi, la media si sia abbassata (da 4,50 a 4,25) e la deviazione standard sia aumentata (da 1,00 a 1,14), mostrando un range più ampio e una distribuzione più equilibrata delle valutazioni.

Dal dataset dei metadati sono stati mantenuti solamente i prodotti che compaiono anche nel dataset delle recensioni filtrato, ottenendo quindi 2.601 prodotti con le relative informazioni sui titoli e descrizioni.

2.4 Esplorazione dei dati

Analizziamo più nel dettaglio la struttura del dataset per comprendere le caratteristiche principali dei dati raccolti.

2.4.1 Distribuzione dei Rating

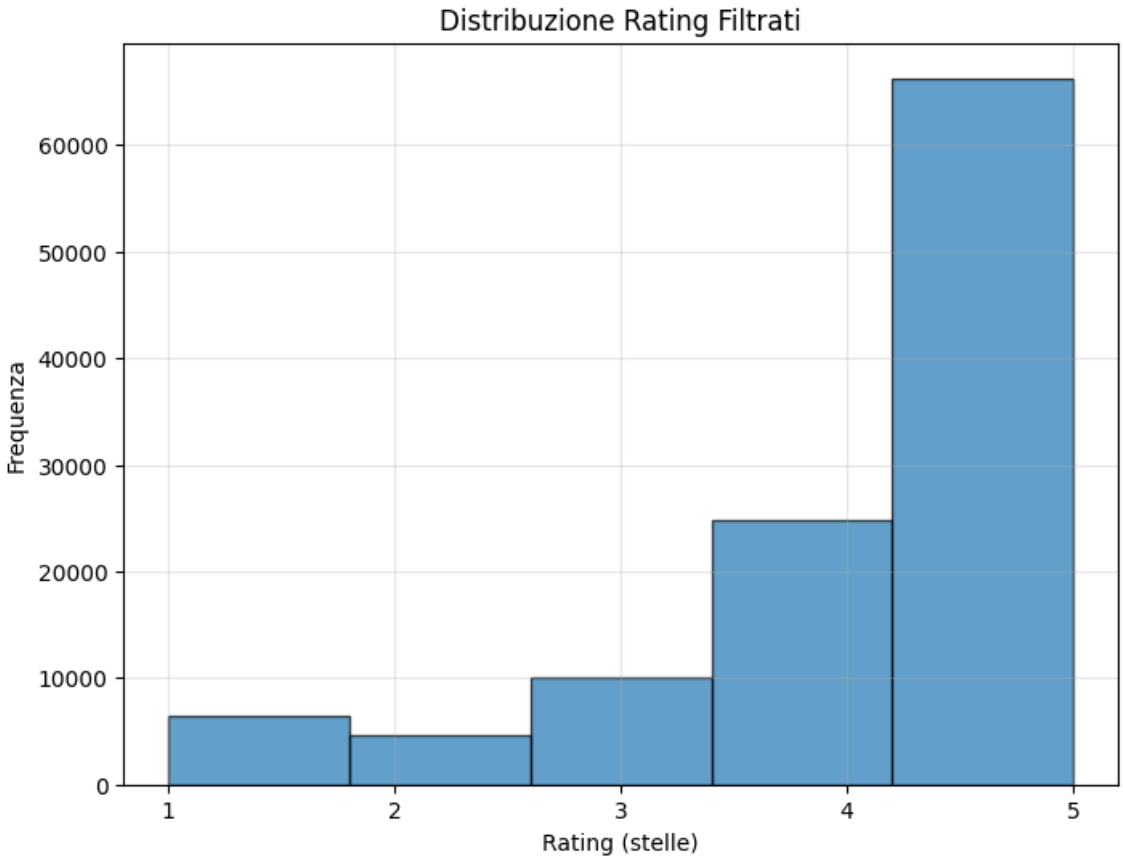

Figura 1: Distribuzione dei rating nel dataset

Come già evidenziato, la media dei rating è di 4,25. La distribuzione dettagliata delle valutazioni è riportata nella tabella seguente:

Rating	Count	Percentuale
1	6.404	6,86%
2	4.595	4,65%
3	10.064	9,91%
4	24.818	22,18%
5	66.259	56,40%

Tabella 5: Distribuzione dei rating nel dataset filtrato

Il grafico mostra chiaramente come la maggior parte delle recensioni sia concentrata sui valori più alti della scala, con le 5 stelle che rappresentano quasi il 60% del totale. È evidente una netta preferenza degli utenti per valutazioni positive, mentre le valutazioni intermedie (2 e 3 stelle) sono meno frequenti rispetto agli estremi.

2.4.2 Distribuzione di Prodotti e Utenti

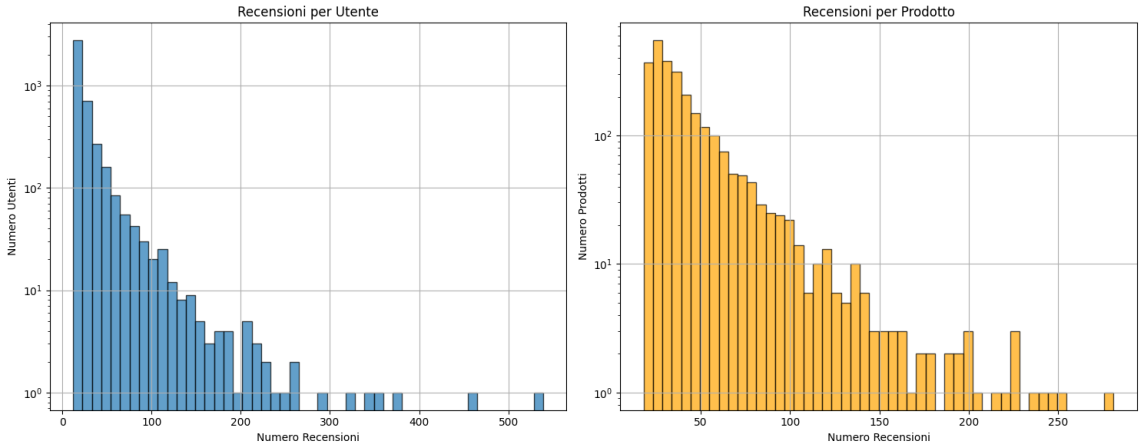

Figura 2: Distribuzione delle recensioni per utenti e prodotti

I due istogrammi rivelano pattern interessanti nel comportamento di utenti e prodotti:

Attività degli Utenti:

- **Media** recensioni per utente: 26,58

- **Mediana**: 18,0
- **Massimo**: 539 recensioni

Il grafico delle recensioni per utente mostra una distribuzione fortemente asimmetrica. La maggior parte degli utenti ha scritto relativamente poche recensioni, con una concentrazione significativa nelle prime classi dell'istogramma. Tuttavia, esiste una "coda lunga" di utenti molto attivi che hanno recensito centinaia di prodotti, come evidenziato dal valore massimo di 539 recensioni.

Popolarità dei Prodotti:

- **Media** recensioni per prodotto: 43,11
- **Mediana**: 34,0
- **Massimo**: 281 recensioni

Anche la distribuzione delle recensioni per prodotto presenta una forma simile, con molti prodotti che ricevono un numero relativamente basso di recensioni e alcuni che invece risultano particolarmente popolari. Il picco del grafico si concentra nelle prime classi, indicando che la maggior parte dei prodotti nel dataset ha ricevuto tra le 20 e le 50 recensioni circa.

2.4.3 Analisi di Correlazione

È stata condotta un'analisi di correlazione per identificare le relazioni tra le variabili principali del dataset: i singoli rating, il numero di recensioni per utenti e prodotti, e i rating medi.

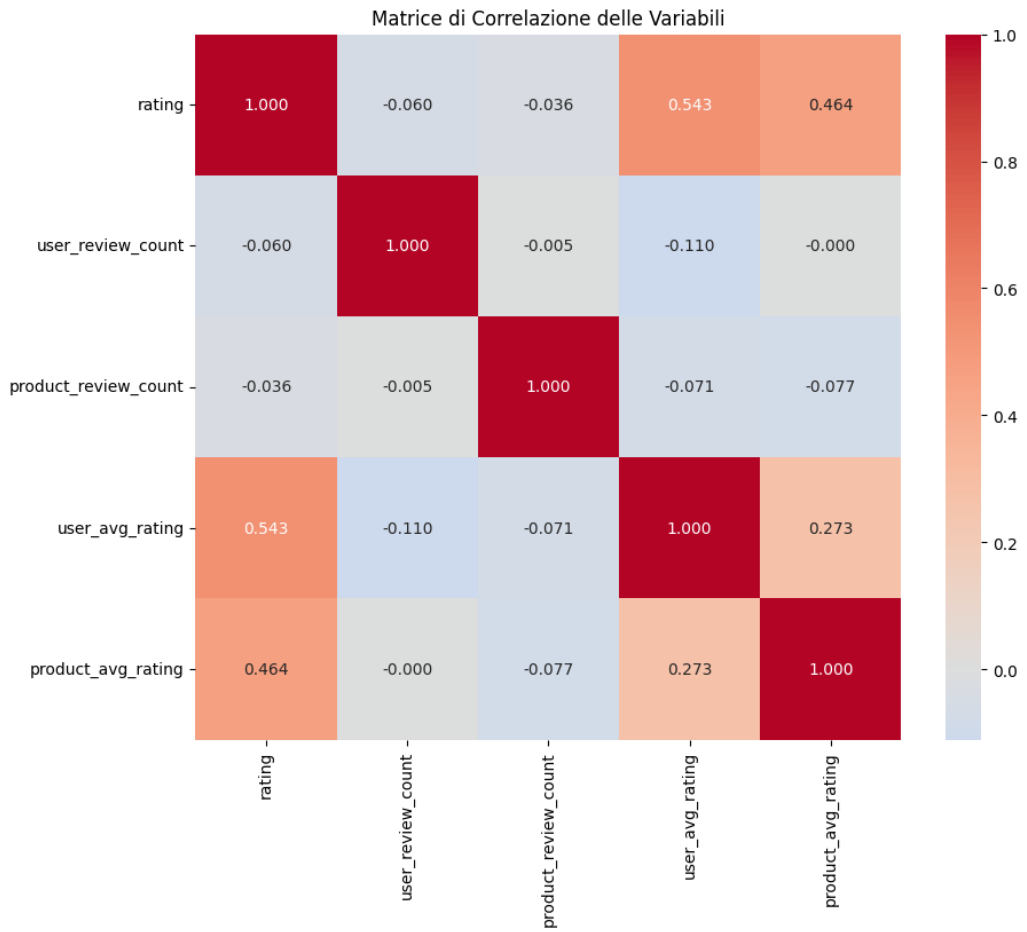

Figura 3: Matrice di correlazione delle variabili principali

I risultati evidenziano alcune relazioni significative. Si osserva una correlazione moderata tra il rating individuale e la media generale dell'utente (0.543), indicando che gli utenti mantengono un comportamento valutativo coerente. Analogamente, esiste una correlazione positiva tra i singoli rating e la media del prodotto (0.464), suggerendo che i prodotti con buona reputazione tendono a ricevere valutazioni elevate.

Un aspetto rilevante è che il numero di recensioni scritte dagli utenti mostra correlazioni molto deboli con i rating assegnati. Questo indica che l'attività di recensione non influisce sui pattern delle valutazioni.

3 Progetto Base: Sistema di Raccomandazione con Collaborative Filtering

Dopo l'analisi esplorativa, il progetto si è concentrato sull'implementazione di un sistema di raccomandazione basato su *collaborative filtering*, con l'obiettivo di identificare la configurazione ottimale degli algoritmi e fornire raccomandazioni personalizzate agli utenti.

3.1 Preparazione dei Dati e Split del Dataset

Il dataset è stato preparato per l'addestramento e la valutazione dei modelli utilizzando la libreria **Surprise**, specializzata per sistemi di raccomandazione. La suddivisione è stata effettuata seguendo lo standard 80/20:

- **Training set:** 89.712 osservazioni (80%)
- **Test set:** 22.428 osservazioni (20%)

È stato eseguito un *sanity check* per verificare l'integrità dei dati durante la conversione e assicurare che non vi fossero perdite di informazioni nel processo.

3.2 Implementazione e Ottimizzazione dell'Algoritmo K-NN

3.2.1 Test Iniziale e Configurazione Base

L'implementazione iniziale dell'algoritmo K-NN con similarità MSD (*Mean Squared Difference*) ha prodotto i seguenti risultati baseline:

Metrica	**Valore**
MSE	0.9647
RMSE	0.9822

Un test di predizione singola ha mostrato la tendenza dell'algoritmo a sovrastimare i rating bassi, come evidenziato dal caso: rating reale = 1.00, rating predetto = 5.00 con k effettivo = 40.

3.2.2 Ottimizzazione del Parametro K

Per identificare il valore ottimale di K, è stata condotta un'analisi sistematica testando valori da 5 a 40 con incrementi di 5. Il metodo del gomito ha identificato $\mathbf{K = 15}$ come punto ottimale, rappresentando il miglior compromesso tra bias e varianza del modello.

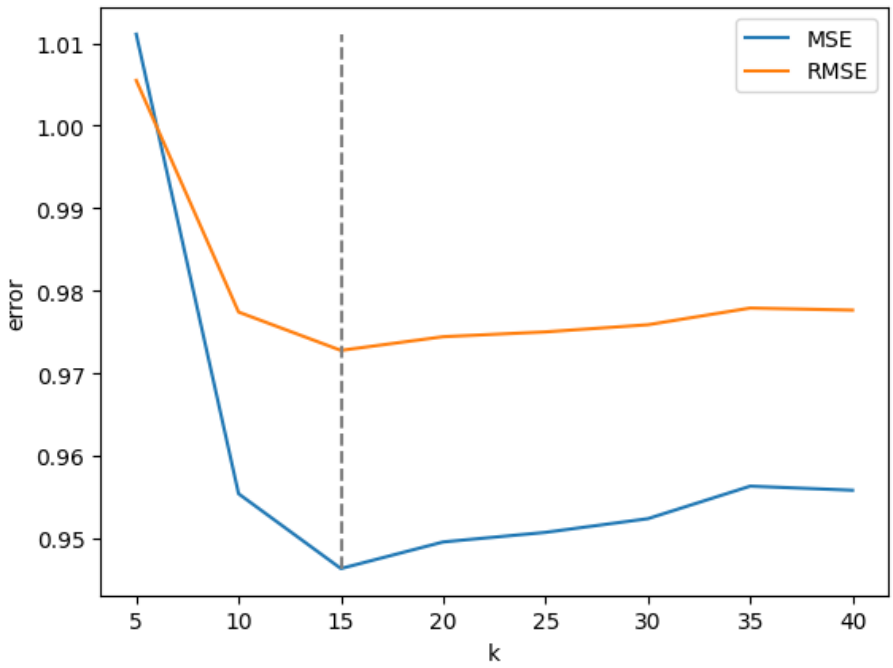

Figura 4: Analisi dell'errore (MSE e RMSE) in funzione del parametro K

3.2.3 Cross-Validation e Configurazione Ottimale

Una ricerca esaustiva tramite *grid search* con cross-validation è stata effettuata sui seguenti parametri:

- **K:** range(15, 45, 5)
- **Metriche di similarità:** cosine, pearson
- **Approccio:** user-based, item-based

La configurazione ottimale identificata è stata:

Parametro	Valore Ottimale
K	40
Similarità	Cosine
Approccio	Item-based
MSE	0.9705
RMSE	0.9851

3.3 Implementazione e Ottimizzazione dell'Algoritmo SVD

3.3.1 Test Iniziale

L'implementazione iniziale dell'algoritmo SVD ha mostrato prestazioni superiori rispetto al K-NN:

Metrica	Valore
MSE	0.8232
RMSE	0.9073

3.3.2 Ottimizzazione tramite Grid Search

La ricerca della configurazione ottimale ha esplorato i seguenti iperparametri:

- **n_factors:** 80, 100, 120, 140
- **n_epochs:** 10, 20, 30, 40
- **biased:** True, False
- **lr_all:** 0.005, 0.01, 0.02
- **reg_all:** 0.02, 0.05, 0.1, 0.2

La configurazione ottimale ha prodotto un significativo miglioramento delle prestazioni:

Parametro	Valore Ottimale
n_factors	120
n_epochs	40
biased	True
lr_all	0.02
reg_all	0.2
MSE	**0.7793**
RMSE	**0.8828**

3.4 Confronto delle Performance

Il confronto finale tra i due algoritmi ottimizzati mostra la superiorità dell'approccio SVD:

Algoritmo	**MSE**	**RMSE**
K-NN (ottimizzato)	0.9705	0.9851
SVD (ottimizzato)	**0.7793**	**0.8828**
Miglioramento	19.7%	10.4%

3.5 Generazione delle Raccomandazioni

3.5.1 Processo di Filling della Matrice

Utilizzando le configurazioni ottimali, sono state generate le raccomandazioni per tutti gli utenti. Il processo ha seguito questi step:

1. Predizione dei rating per tutti gli item non recensiti
2. Esclusione degli item già valutati dall'utente
3. Ordinamento per rating predetto decrescente
4. Selezione dei top-20 item per utente

3.5.2 Confronto delle Raccomandazioni K-NN vs SVD

Per illustrare le differenze tra i due approcci, riportiamo le raccomandazioni generate per l'utente di test AGTMZCWIWBH43TCW7UKG2YV2EKKA:

Raccomandazioni K-NN (Item-based, Cosine Similarity):

#	**Album Raccomandato**	**Rating**
1	Reroute to Remain	5.00
2	Buffalo Springfield Again	5.00
3	Carnival of Souls: The Final Sessions	5.00
4	Wheels Of Fire	5.00
5	Mystery to Me	5.00
6	Greatest Hits	5.00
7	Freak Out	5.00
8	One More From the Road	5.00
9	Save the Turtles: The Turtles Greatest Hits	5.00
10	Jailbreak	5.00
11	Hello Nasty (Clean Version)	5.00
12	Great Milenko (Explicit Lyrics)	5.00
13	Oceania	5.00
14	Coverdale & Page	5.00
15	Best of	5.00
16	Some Girls	5.00
17	Blizzard of Ozz	5.00
18	Silver & Gold	5.00
19	A Decade of Hits 1969-1979	5.00
20	White Light/White Heat	5.00

Raccomandazioni SVD (Matrix Factorization):

#	**Album Raccomandato**	**Rating**
1	Red: 30th Anniversary Editions	5.00
2	Aja Remastered	5.00
3	Fear Of A Black Planet (explicit lyrics)	5.00
4	Hot Rocks	5.00
5	Rickie Lee Jones	5.00
6	Chronicle: 20 Greatest 1976	5.00
7	The Cars	5.00
8	Let It Bleed	5.00
9	Ace Of Spades	5.00
10	Thriller	5.00
11	Time Out	5.00
12	Hunky Dory	5.00
13	In the Nightside Eclipse	5.00
14	Sons of Northern Darkness	5.00
15	A Charlie Brown Christmas: Original Sound Track	5.00
16	Mezzanine	5.00
17	In Step	5.00
18	Innervisions Remastered	5.00
19	Graceland	5.00
20	Symbolic	5.00

Entrambi gli algoritmi assegnano il rating massimo (5.00) a tutti i top-20 item, ma la differenza sostanziale sta nella selezione e nell'ordinamento degli album raccomandati.

3.5.3 Matrice delle Raccomandazioni per l'Intero Dataset

La generazione delle raccomandazioni è stata estesa a tutti gli utenti del dataset. Un esempio della struttura della matrice finale (prime 5 righe) mostra la consistenza del processo:

User ID	Top-3 Raccomandazioni	Rating
AFW2PDT3AMT4X3PYQG7FJZH5FXFA	1. Ultimate Rascals, The	5.0
	2. Carole King Tapestry	5.0
	3. Kind Of Blue	5.0
AE7BV6IMNPZ3F266H7PXMH3BZQNQ	1. Innervisions Remastered	5.0
	2. Dirt (Explicit Lyrics)	5.0
	3. At Folsom Prison	5.0
AGTMZCWIWBH43TCW7UKG2YV2EKKA	1. Innervisions Remastered	5.0
	2. Dirt (Explicit Lyrics)	5.0
	3. In Step	5.0
AFCU2ZFZ2ZLMM5YX2MXUOV52WMKQ	1. At Folsom Prison	5.0
	2. Message In A Box: Complete Recordings	5.0
	3. Fear Of A Black Planet	5.0
AGWDYYVVWM3DC3CASUZKXK67G6IA	1. The Immaculate Collection	5.0
	2. Harem	5.0
	3. Scarlet's Walk	5.0

3.6 Segmentazione degli Utenti tramite Clustering

3.6.1 Determinazione del Numero Ottimale di Cluster

L'algoritmo K-Means è stato applicato sulla matrice di rating completata. Il metodo del gomito ha suggerito $\mathbf{K = 5}$ come numero ottimale di cluster.

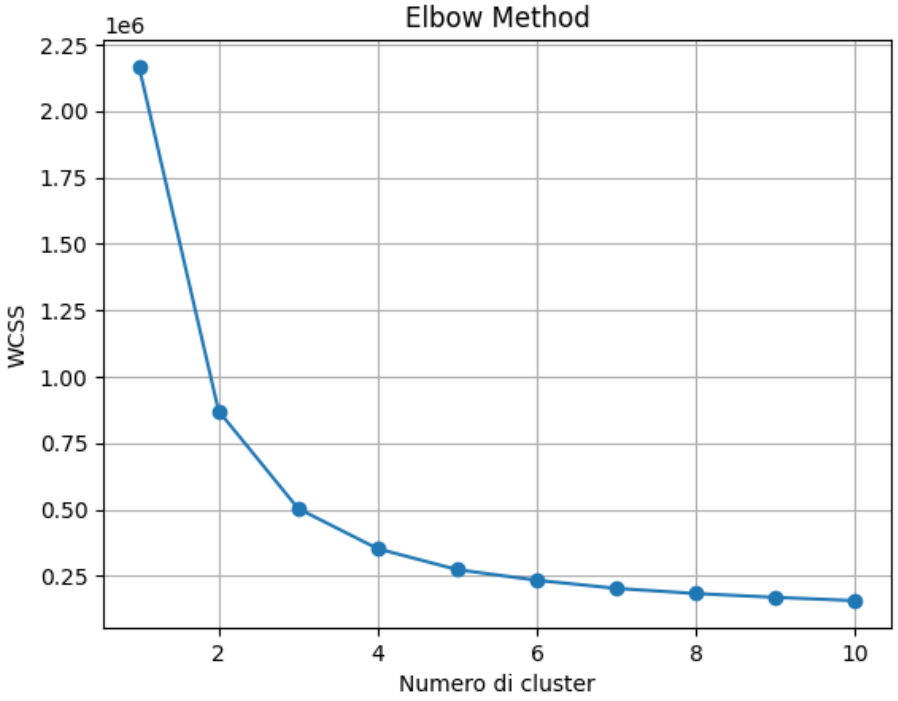

Figura 5: Metodo del gomito per la determinazione del numero di cluster

3.6.2 Configurazione K-Means

I parametri utilizzati per il clustering sono stati:

- **n_clusters:** 5
- **init:** k-means++
- **max_iter:** 300
- **n_init:** 10
- **random_state:** 42

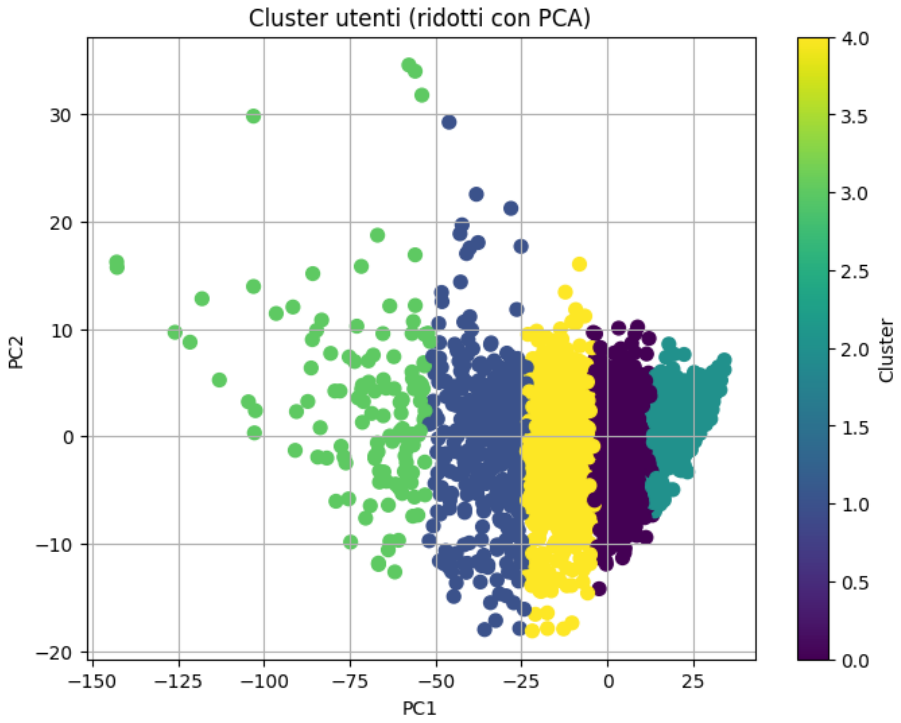

Figura 6: Visualizzazione dei cluster tramite riduzione PCA a 2 componenti

3.6.3 Interpretazione dei Cluster

L'analisi dei cluster ha rivelato cinque profili distinti di utenti:

Cluster	Utenti	Rating Medio	Std	Profilo
0 (Viola)	1307 (31.0%)	4.370	0.373	Utenti positivi
1 (Blu)	417 (9.9%)	3.608	0.448	Utenti moderati
2 (Verde)	1403 (33.3%)	4.679	0.328	Utenti entusiasti
3 (Giallo)	125 (3.0%)	2.903	0.554	Utenti critici
4 (Teal)	967 (22.9%)	4.027	0.408	Utenti equilibrati

Analisi dettagliata dei cluster:

- **Cluster 2 (Teal):** Il gruppo più generoso nelle valutazioni (33.3% degli utenti), con rating medio di 4.679 e bassa variabilità (0.328). Questi utenti tendono a valutare molto positivamente i prodotti, con molti item che ricevono il punteggio massimo di 5.0.

- **Cluster 0 (Viola):** Secondo gruppo per dimensione (31.0%), con valutazioni tendenzialmente positive (media 4.370) e consistenti (std 0.373).
- **Cluster 4 (Giallo):** Rappresenta il 22.9% degli utenti con un approccio bilanciato (media 4.027), mostrando maggiore discriminazione nelle valutazioni.
- **Cluster 1 (Blu):** Gruppo più piccolo tra i principali (9.9%), con rating medio di 3.608 e maggiore variabilità (0.448), indicando un approccio più critico e selettivo.
- **Cluster 3 (Verde):** Il gruppo più esiguo (3.0%) ma distintivo, con rating medio di 2.903 e la più alta variabilità (0.554). Rappresenta gli utenti più critici del sistema.

3.7 Conclusioni del Progetto Base

Il sistema di raccomandazione implementato ha raggiunto i seguenti obiettivi:

1. **Identificazione della configurazione ottimale:** SVD con n_factors=120, n_epochs=40, biased=True ha dimostrato prestazioni superiori con RMSE di 0.8828
2. **Filling efficace della matrice:** Completamento della matrice sparsa con predizioni affidabili per tutti gli utenti
3. **Segmentazione significativa:** Identificazione di 5 profili utente distinti che permettono strategie di raccomandazione personalizzate
4. **Sistema di raccomandazione funzionale:** Generazione di top-10 raccomandazioni personalizzate per ogni utente basate sui pattern di preferenza identificati

Il confronto tra K-NN e SVD ha evidenziato la superiorità degli approcci basati su fattorizzazione matriciale per questo dataset, con un miglioramento del 19.7% in termini di MSE rispetto all'approccio basato su vicinanza.

4 Progetto Intermedio: Sistema di Raccomandazione Content-Based

L'obiettivo del progetto intermedio è stato quello di estendere il sistema di raccomandazione del progetto base, introducendo un approccio *content-based* che sfrutta le caratteristiche intrinseche dei prodotti per generare raccomandazioni personalizzate. A differenza del *collaborative filtering*, questo metodo utilizza le informazioni testuali dei prodotti — in particolare i campi *title* e *description* — per identificare e suggerire articoli simili a quelli già apprezzati dall'utente.

4.1 Preparazione e Pre-processing dei Dati Testuali

Per trasformare le informazioni testuali non strutturate in rappresentazioni numeriche utilizzabili dai modelli di machine learning, è stata implementata una pipeline completa di Natural Language Processing (NLP). Per illustrare l'evoluzione del testo attraverso le varie fasi, consideriamo l'esempio del prodotto "October Rust" (album dei Type O Negative).

4.1.1 Combinazione dei Campi Testuali

Il primo passo ha previsto l'unione dei campi *title* e *description* del dataset dei metadati in un unico campo testuale combinato. Questa operazione ha permesso di creare un corpus più ricco e informativo per ciascun prodotto.

Esempio di trasformazione:

- *Title*: "October Rust"
- *Description*: "October Rust is the fourth studio album by Type O Negative. It was released in 1996. This is the first album with Johnny Kelly credited as the band's drummer..."
- *Combined text*: "October Rust October Rust is the fourth studio album by Type O Negative. It was released in 1996..."

4.1.2 Tokenizzazione

Il testo combinato è stato successivamente suddiviso in unità elementari (token) utilizzando `word_tokenize` di NLTK. Questo processo ha trasformato le stringhe di testo continuo in liste di parole individuali.

Risultato dopo tokenizzazione:

```
['October', 'Rust', '[', '``', 'October', 'Rust', 'is', 'the',
'fourth', 'studio', 'album', 'by', 'Type', 'O', 'Negative', '.',
'It', 'was', 'released', 'in', '1996', '.', 'This', 'is', 'the',
'first', 'album', 'with', 'Johnny', 'Kelly', 'credited', 'as',
'the', 'band', "'s", 'drummer', ',', 'although', 'programmed',
'drums', 'are', 'used', 'on', 'the', 'album', '.', ...]
```

4.1.3 Rimozione di Stopwords e Punteggiatura

Per migliorare la qualità della rappresentazione testuale, sono stati rimossi elementi non informativi utilizzando il dizionario di stopwords inglesi di NLTK e filtrando tutti i caratteri di punteggiatura.

Risultato dopo rimozione stopwords e punteggiatura:

```
['October', 'Rust', 'October', 'Rust', 'fourth', 'studio',
'album', 'Type', 'Negative', 'released', '1996', 'first',
'album', 'Johnny', 'Kelly', 'credited', 'band', 'drummer',
'although', 'programmed', 'drums', 'used', 'album', 'October',
'Rust', 'ballads', 'less', 'doom', 'metal', 'sound', 'previous',
'subsequent', 'albums']
```

Si nota come siano state eliminate parole comuni ("is", "the", "was", "by") e segni di punteggiatura, mantenendo solo i termini con valore semantico.

4.1.4 Lemmatizzazione

L'ultimo passaggio del pre-processing ha previsto la riduzione dei token alla loro forma base attraverso il `WordNetLemmatizer` di NLTK.

Risultato finale dopo lemmatizzazione:

```
['October', 'Rust', 'October', 'Rust', 'fourth', 'studio',
'album', 'Type', 'Negative', 'released', '1996', 'first',
'album', 'Johnny', 'Kelly', 'credited', 'band', 'drummer',
'although', 'programmed', 'drum', 'used', 'album', 'October',
'Rust', 'ballad', 'less', 'doom', 'metal', 'sound', 'previous',
'subsequent', 'album']
```

Le trasformazioni principali includono: "drums" $\rightarrow$ "drum", "ballads" $\rightarrow$ "ballad", "albums" $\rightarrow$ "album". Questo processo ha normalizzato le varianti morfologiche riducendo la dimensionalità del vocabolario.

4.2 Tecniche di Embedding

Dopo il pre-processing, il testo pulito è stato convertito in rappresentazioni numeriche mediante due tecniche distinte.

4.2.1 Embedding basato su TF-IDF

La prima tecnica implementata è stata TF-IDF (Term Frequency-Inverse Document Frequency), un metodo classico basato sulla frequenza dei termini che cattura l'importanza di una parola in un documento rispetto all'intero corpus.

I token lemmatizzati sono stati prima riconvertiti in stringhe unendo i termini con spazi. Successivamente, è stato applicato il vettorizzatore TF-IDF con i seguenti parametri ottimizzati:

- **max_features=5000**: limita il vocabolario alle 5000 features più rilevanti per bilanciare ricchezza informativa e complessità computazionale
- **ngram_range=(1,2)**: cattura sia singole parole (unigrams) che coppie di parole consecutive (bigrams), permettendo di identificare espressioni composte come "studio album" o "doom metal"
- **min_df=2**: filtra termini che appaiono in meno di 2 documenti, eliminando parole troppo rare
- **max_df=0.95**: esclude termini presenti in più del 95% dei documenti, rimuovendo parole troppo comuni non catturate dalle stopwords

Il risultato è una matrice sparsa dove ogni prodotto è rappresentato da un vettore di 5000 dimensioni, con valori che riflettono l'importanza relativa di ciascun termine.

4.2.2 Embedding basato su Transformer

La seconda tecnica ha utilizzato un modello neurale pre-addestrato basato su architettura transformer, specificamente il modello `average_word_embeddings_komninos` dalla libreria Sentence Transformers.

Questo approccio genera embedding densi di dimensionalità fissa che catturano relazioni semantiche e contestuali profonde tra le parole. A differenza di TF-IDF, i transformer possono comprendere sinonimi, relazioni semantiche e contesto, producendo rappresentazioni più sofisticate del contenuto testuale.

4.3 Implementazione del Sistema di Raccomandazione con K-NN

Per generare le raccomandazioni è stato utilizzato l'algoritmo K-Nearest Neighbors (K-NN) in modalità regressione, applicato agli embedding ottenuti dalle due tecniche.

Il processo di raccomandazione si articola nei seguenti passaggi:

1. **Filtraggio utenti**: sono stati selezionati solo gli utenti con almeno 25 rating per garantire sufficiente informazione storica
2. **Creazione dataset personalizzato**: per ogni utente, sono stati recuperati i prodotti valutati e i corrispondenti embedding
3. **Split train/test**: i dati sono stati divisi con proporzione 80/20 per training e validazione
4. **Training K-NN**: il modello è stato addestrato utilizzando la metrica coseno per la similarità, con un numero di vicini pari a min(40, dimensione training set)
5. **Predizione e valutazione**: sono stati predetti i rating sul test set e calcolato l'errore quadratico medio (MSE)

4.4 Risultati e Valutazione Critica

4.4.1 Performance delle Tecniche di Embedding

I risultati ottenuti mostrano performance comparabili tra le due tecniche:

- **TF-IDF**: MSE medio = 1.03, RMSE medio = 1.02
- **Transformer**: MSE medio = 1.01, RMSE medio = 1.00

Contrariamente alle aspettative iniziali, il modello transformer ha mostrato solo un miglioramento marginale rispetto a TF-IDF. Questo risultato può essere attribuito a diversi fattori:

1. **Natura dei testi**: titoli e descrizioni di prodotti sono testi relativamente brevi e descrittivi, dove le parole chiave hanno un ruolo predominante rispetto alle relazioni semantiche complesse

2. **Dimensione del dataset**: con un corpus limitato, i vantaggi dei transformer nel catturare pattern complessi potrebbero non emergere completamente

3. **Overhead computazionale**: i transformer richiedono risorse computazionali significativamente maggiori per un miglioramento minimo delle performance

4.5 Confronto con Collaborative Filtering

4.5.1 Performance a Confronto

Il collaborative filtering con SVD ottiene risultati significativamente migliori del content-based:

Approccio	Algoritmo	MSE	RMSE	Δ vs Best CF
Collaborative Filtering	K-NN	0.971	0.985	+24.6%
	SVD	**0.779**	**0.883**	**baseline**
Content-Based	TF-IDF + K-NN	1.030	1.020	+32.2%
	Transformer + K-NN	1.010	1.000	+29.7%

Tabella 6: Confronto diretto delle metriche di errore tra gli approcci

L'SVD supera il miglior content-based (Transformer) del 23% in termini di MSE, dimostrando la superiorità del collaborative filtering quando sono disponibili sufficienti dati di interazione.

4.5.2 Vantaggi e Limitazioni

Collaborative Filtering:

- **Pro:** Accuratezza superiore (RMSE 0.883), scoperta di pattern latenti, raccomandazioni diversificate
- **Contro:** Non gestisce nuovi prodotti senza rating, richiede massa critica di utenti, raccomandazioni non interpretabili

Content-Based:

- **Pro:** Funziona con nuovi prodotti, non richiede molti utenti, raccomandazioni spiegabili
- **Contro:** Accuratezza inferiore (RMSE 1.000), tende a raccomandare item molto simili, dipende dalla qualità delle descrizioni

4.6 Conclusioni

Il collaborative filtering con SVD rimane l'approccio migliore per accuratezza predittiva pura, con performance superiori del 23% rispetto al content-based. Tuttavia, il content-based risulta indispensabile per gestire il problema del cold start e fornire raccomandazioni interpretabili.

La soluzione ottimale per un sistema reale sarebbe un approccio ibrido che utilizzi:

- SVD come motore principale per utenti e prodotti con storia sufficiente
- Content-based per nuovi prodotti e utenti
- Ponderazione dinamica basata sulla disponibilità dei dati

Questo permetterebbe di combinare l'accuratezza del collaborative filtering con la robustezza del content-based, mitigando le debolezze di entrambi gli approcci.

5 Progetto Avanzato: Sentiment Analysis sulle Recensioni

Il progetto avanzato estende le funzionalità del sistema di raccomandazione attraverso l'implementazione di tecniche di Natural Language Processing per l'analisi del sentiment delle recensioni utente. L'obiettivo principale consiste nel predire automaticamente il sentiment delle recensioni basandosi sul loro contenuto testuale, classificandole in tre categorie: negative, neutre e positive.

5.1 Preparazione dei Dati e Trasformazione del Target

La prima fase del progetto ha richiesto la trasformazione dei rating numerici in classi di sentiment discrete, seguendo la mappatura specificata nei requisiti:

- **Sentiment negativo**: rating 1-2
- **Sentiment neutro**: rating 3
- **Sentiment positivo**: rating 4-5

L'analisi della distribuzione risultante ha evidenziato un forte sbilanciamento del dataset:

- Recensioni positive: 91.077 (81,2%)
- Recensioni negative: 10.999 (9,8%)
- Recensioni neutre: 10.064 (9,0%)

Questo sbilanciamento rappresenta una sfida significativa per l'addestramento di modelli di classificazione efficaci, richiedendo strategie specifiche di bilanciamento durante la fase di training.

5.2 Processamento del Testo con Tecniche NLP

Il preprocessing dei campi testuali `title` e `text` delle recensioni ha seguito una pipeline strutturata di Natural Language Processing:

5.2.1 Pipeline di Preprocessing

1. **Tokenizzazione**: segmentazione del testo in unità discrete (token)
2. **Normalizzazione**: conversione in minuscolo per uniformità
3. **Rimozione stopwords**: eliminazione di parole comuni non informative
4. **Rimozione punteggiatura**: pulizia da caratteri non alfabetici
5. **Lemmatizzazione**: riduzione delle parole alla loro forma base

Il preprocessing ha processato con successo 112.140 recensioni, preparando il testo per le successive fasi di embedding.

5.3 Tecniche di Embedding

Come da specifica, sono state implementate due tecniche complementari di embedding per la rappresentazione vettoriale del testo:

5.3.1 TF-IDF: Approccio Basato sulla Frequenza

L'embedding TF-IDF è stato configurato con i seguenti parametri:

- `max_features`: 3000 feature più rilevanti
- `ngram_range`: (1, 2) per catturare unigrammi e bigrammi
- `min_df`: 2 documenti minimi per feature
- `max_df`: 0.95 per escludere termini troppo comuni

Questa configurazione ha prodotto una matrice di 112.140 $\times$ 3000 feature, rappresentando ogni recensione come vettore sparso basato sulla frequenza pesata dei termini.

5.3.2 Transformer Embeddings: Approccio Neurale

Per l'approccio neurale è stato utilizzato il modello `average_word_embeddings_komninos`, che genera rappresentazioni dense di dimensione 300. Questa tecnica cattura relazioni semantiche più profonde tra le parole, producendo embeddings contestualizzati per ogni recensione.

5.4 Modelli di Classificazione

Per entrambe le tecniche di embedding è stato utilizzato un **Random Forest Classifier** con le seguenti configurazioni chiave:

- `n_estimators`: 200 alberi
- `class_weight`: "balanced" per gestire lo sbilanciamento
- `max_features`: "sqrt" per la selezione delle feature
- `min_samples_split`: 5
- `min_samples_leaf`: 2
- `random_state`: 42 per riproducibilità

Il parametro `class_weight="balanced"` è stato cruciale per mitigare l'impatto dello sbilanciamento del dataset, assegnando pesi inversamente proporzionali alla frequenza delle classi.

5.5 Risultati e Valutazione delle Performance

5.5.1 Performance con TF-IDF

Il modello basato su TF-IDF ha raggiunto le seguenti metriche:

Metrica	**Valore Globale**	**Negative**	**Neutral**	**Positive**
Accuracy	0.857	-	-	-
RMSE	0.563	-	-	-
MAE	0.201	-	-	-
Precision	-	0.77	0.71	0.86
Recall	-	0.48	0.08	0.99
F1-Score	-	0.60	0.15	0.92

Tabella 7: Metriche di valutazione per il modello TF-IDF

La matrice di confusione ha rivelato che:

- Le recensioni positive sono state classificate con alta accuratezza (98.7% recall)
- Le recensioni negative hanno mostrato performance moderate (48.5% recall)
- Le recensioni neutre hanno sofferto di scarsissimo recall (8.2%)

5.5.2 Performance con Transformer Embeddings

Il modello basato su embeddings neurali ha ottenuto:

Metrica	Valore Globale	Negative	Neutral	Positive
Accuracy	0.830	-	-	-
RMSE	0.646	-	-	-
MAE	0.252	-	-	-
Precision	-	0.87	0.72	0.83
Recall	-	0.18	0.04	1.00
F1-Score	-	0.29	0.08	0.91

Tabella 8: Metriche di valutazione per il modello con Transformer Embeddings

Sorprendentemente, gli embeddings neurali hanno mostrato performance inferiori rispetto a TF-IDF, con un calo particolarmente marcato nel recall delle classi minoritarie.

5.6 Analisi Critica e Limitazioni

L'analisi dei risultati evidenzia diverse criticità:

5.6.1 Problema dello Sbilanciamento

Nonostante l'utilizzo di pesi bilanciati, il forte sbilanciamento del dataset (81.2% positive) ha portato il modello a favorire eccessivamente la classe maggioritaria. Le classi neutre e negative, rappresentando solo il 19% dei dati, non hanno fornito esempi sufficienti per un apprendimento robusto.

5.6.2 Ambiguità della Classe Neutra

Le recensioni con rating 3 presentano caratteristiche linguistiche ambigue, spesso contenendo elementi sia positivi che negativi. Questa ambiguità intrinseca rende la loro classificazione particolarmente complessa, come evidenziato dal recall estremamente basso (4-8%) in entrambi gli approcci.

5.6.3 Limitazioni degli Embeddings

- **TF-IDF**: pur mostrando performance superiori, manca di comprensione semantica contestuale

- **Transformer**: gli embeddings pre-addestrati potrebbero non essere ottimizzati per il dominio specifico delle recensioni prodotto

6 Conclusioni e Interpretazione Sintetica dei Risultati

Il progetto ha permesso di esplorare e confrontare in modo sistematico diversi approcci alla raccomandazione e all'analisi dei dati in un contesto di e-commerce. Ogni fase ha aggiunto un tassello al quadro complessivo, offrendo le seguenti conclusioni chiave:

- **Collaborative Filtering:** Gli algoritmi di *matrix factorization*, come SVD, si sono dimostrati particolarmente efficaci per la predizione dei rating, superando i modelli basati sul vicinato come K-NN. L'implementazione di algoritmi di clustering ha permesso di segmentare gli utenti in profili con preferenze distinte, un'informazione preziosa per personalizzare ulteriormente le raccomandazioni.
- **Content-Based Filtering:** Il confronto tra TF-IDF e un modello transformer ha mostrato l'importanza di scegliere lo strumento giusto per il problema. Per i metadati dei prodotti in questo set di dati, TF-IDF ha offerto un'ottima combinazione di accuratezza ed efficienza, dimostrando che non sempre un modello più complesso garantisce prestazioni migliori.
- **Sentiment Analysis:** L'utilizzo di un *RandomForestClassifier* per l'analisi del sentiment ha evidenziato le sfide pratiche di lavorare con dati reali. Nonostante una buona performance complessiva, la difficoltà nel classificare le recensioni "neutre" ha sottolineato i limiti degli algoritmi quando si affrontano problemi di sbilanciamento delle classi e ambiguità semantica.