

# Università degli Studi di Milano Bicocca

Dipartimento di Informatica

Corso di Laurea Triennale



Progetto Metodi Informatici per la Gestione Aziendale

**ANNO ACCADEMICO 2023/2024**

Alessandro Isceri - 879309 & Luca Bolis - 879263

# Indice

<b>1</b>	<b>Executive Summary</b>	<b>1</b>
<b>2</b>	<b>Introduzione al Problema</b>	<b>2</b>
2.1	Obiettivo del Progetto . . . . .	2
2.2	Descrizione del Dataset . . . . .	2
2.3	Obiettivi dell'Analisi dei Dati . . . . .	4
2.4	Risultati Analisi Esplorativa . . . . .	4
2.4.1	Analisi Descrittiva . . . . .	4
2.4.2	Analisi di Correlazione . . . . .	6
2.4.3	Analisi Distribuzionale . . . . .	7
2.4.4	Distribuzione Temporale . . . . .	9
<b>3</b>	<b>Progetto Base</b>	<b>12</b>
3.1	Creazione Dataset Surprise . . . . .	12
3.2	Configurazione Ottimale KNN . . . . .	13
3.3	Configurazione Ottimale Matrix Factorization . . . . .	13
3.4	Segmentazione degli utenti: K-Means . . . . .	13
3.5	Collaborative Filtering Recommendation System . . . . .	15
3.6	KNN vs Matrix Factorization . . . . .	15
3.6.1	Confronto RMSE . . . . .	15
3.6.2	Confronto Raccomandazioni . . . . .	15
<b>4</b>	<b>Progetto Intermedio</b>	<b>18</b>
4.1	Riferimenti Teorici . . . . .	18
4.1.1	Tokenizzazione . . . . .	18
4.1.2	Embedding . . . . .	19
4.1.3	Transformer . . . . .	21
4.2	Processamento Attributi Testuali con NLP . . . . .	21

4.3	Predizione del Rating per Utente . . . . .	22
4.4	Valutazione dei Risultati . . . . .	22
4.5	Confronto tra Progetto Base ed Intermedio . . . . .	23
<b>5</b>	<b>Progetto Avanzato</b>	<b>25</b>
5.1	Preparazione dei Dati . . . . .	25
5.2	Processamento Attributi Testuali con NLP . . . . .	26
5.3	Valutazione dei Risultati . . . . .	26
5.3.1	TF-IDF . . . . .	27
5.3.2	BoW . . . . .	28
5.3.3	Transformer . . . . .	29
<b>6</b>	<b>Conclusione</b>	<b>30</b>

# 1. Executive Summary

Il principale obiettivo del progetto è la costruzione di sistemi di raccomandazione collaborative filtering e content based usando un dataset che contiene recensioni di Amazon su prodotti della categoria "Software". Un altro obiettivo è quello di effettuare sentiment analysis delle recensioni.

Per farlo il progetto è stato sviluppato come segue:

1. **analisi dei dati:** come in tutti i progetti che lavorano con tecniche di Machine Learning, una prima fase di analisi e valutazione del dataset è fondamentale. Per farlo abbiamo studiato le distribuzioni e le correlazioni tra variabili, sfruttando le strategie apprese durante il corso.
2. **progetto base:** sviluppo di un RecSys collaborative filtering sfruttando gli algoritmi KNN e Matrix Factorization; segmentazione degli utenti utilizzando l'algoritmo K-Means.
3. **progetto intermedio:** sviluppo di un RecSys content based, per sfruttare i dati testuali; con l'obiettivo di ottenere le migliori performance sono state implementate tutte le possibili combinazioni di Stemmer/Lemmatizer e Bag of Words/TF-IDF, e sono stati testati anche diversi modelli di Transformer.
4. **progetto avanzato:** analisi del sentiment delle recensioni con l'algoritmo KNN. Per analizzare i dati testuali delle recensioni, è stata seguita la stessa metodologia utilizzata nel progetto intermedio.

Nel sistema di raccomandazione collaborative filtering si ottiene un RMSE leggermente migliore utilizzando Matrix Factorization.

Il sistema di raccomandazione collaborative filtering e content based hanno un RMSE simile nonostante il primo non sfrutti le informazioni testuali. Questo può essere dovuto al fatto che il software presenta descrizioni molto diversificate tra loro, di conseguenza non è semplice trovare correlazioni tra i prodotti.

Nel progetto avanzato i Transformer si sono dimostrati la tecnica migliore: ottenendo risultati ottimi per quanto riguarda la sentiment analysis.

## 2. Introduzione al Problema

### 2.1 Obiettivo del Progetto

L'obiettivo principale del progetto è sviluppare due tipologie di Recommendation System, al fine di compararne le performance e mettere in pratica le nozioni acquisite durante il corso.

- collaborative filtering (progetto base)
- content based (progetto intermedio)

I dati su cui lavorano i Recommendation System sono delle recensioni di Amazon, in particolare verrà analizzato un set di recensioni su prodotti della categoria "Software".

Un altro obiettivo del progetto è utilizzare tecniche di Natural Language Processing (NLP) per classificare il sentiment delle recensioni (progetto avanzato).

### 2.2 Descrizione del Dataset

Come citato in precedenza, i dati utilizzati riguardano delle recensioni di Amazon del 2023.

In particolare, verranno analizzate delle recensioni rilasciate su prodotti della categoria "Software", che contiene 4.880.181 recensioni.

I dati sono contenuti in due file: **User Reviews** e **Item Metadata**.

Vengono riportate di seguito le strutture dei file:

User Reviews:

- rating: rating della recensione  $\in [1;5]$
- title: titolo della recensione
- text: testo della recensione
- images: lista di immagini della recensione
- asin: ID del prodotto

- parent\_asin: ID generale del prodotto
- user\_id: ID dell'utente che ha scritto la review
- timestamp: momento della pubblicazione della recensione
- verified\_purchase: booleano che rappresenta se l'acquisto è stato verificato o meno
- helpful\_vote: numero di helpful votes della review

#### Item Metadata:

- main\_category: categoria principale del prodotto, in questo caso "Software"
- average\_rating: rating del prodotto mostrato sulla pagina del prodotto
- rating\_number: numero di recensioni del prodotto
- features: features del prodotto
- descrizione: descrizione del prodotto
- prezzo: prezzo in dollari
- immagini: immagini del prodotto
- video: video del prodotto
- store: nome del venditore
- categorie: categorie gerarchiche del prodotto
- dettagli: dettagli del prodotto: materiali, brand, dimensione ecc.
- parent\_asin: ID generale del prodotto
- comprati\_insieme: articoli raccomandati

Nel progetto base verranno utilizzati solamente i dati presenti nel file User Reviews.

Nel progetto intermedio e avanzato verranno utilizzati sia i dati del file User Reviews che Item Metadata.

## 2.3 Obiettivi dell'Analisi dei Dati

L'analisi dei dati ha come obiettivo principale quello di analizzare i dati su cui dovranno lavorare i sistemi di raccomandazione, in particolare si dividerà in tre fasi:

1. **analisi descrittiva** per avere un overview della distribuzione dei dati
2. **analisi di correlazione** per vedere se esistono variabili correlate
3. **analisi distribuzionale** in cui verranno analizzate le distribuzioni dei dati con diversi grafici

Attraverso questa prima analisi, sarà possibile fare alcune considerazioni sui dati che potranno rivelarsi utili in futuro per giustificare i risultati ottenuti durante i tre progetti.

## 2.4 Risultati Analisi Esplorativa

### 2.4.1 Analisi Descrittiva

Di seguito vengono riportate alcune statistiche descrittive:

Tabella 2.1: Tabella delle statistiche descrittive calcolate sul dataset

Colonna	Count	Media	Dev Std	Min	25%	50%	75%	Max
rating	111988	3.91	1.39	1.00	3.00	5.00	5.000	5.00
timestamp	111988	01/2017	8.23e+10	09/2001	01/2015	08/2016	03/2019	04/2023
helpful_vote	111988	5.03	30.12	0.00	0.00	1.00	3.00	6178.00


Come si può notare dalla tabella 2.1, il dataset analizzato presenta rating mediamente piuttosto alti, accompagnati da una deviazione standard significativa.

È interessante osservare che il valore della mediana è già 5, il massimo rating ottenibile.

Le recensioni sono state raccolte tra settembre 2001 e aprile 2023.

Per quanto riguarda la colonna "helpful\_vote", si nota che in media una recensione riceve 5 voti utili, con una notevole volatilità nel numero di voti utili per recensione, come dimostra la deviazione

standard. Questo indica che alcune recensioni sono valutate molto utili dalla community, mentre altre lo sono meno.

 Bhatta's Reviews

★★★★★ **Full Five Stars!**

Reviewed in the United States on May 27, 2012

**Verified Purchase**

This is a very addictive fun and fast paced game. I must confess that I had side loaded this game several months ago when it was available on Google play and I loved it.

The game play is superb and has very good fast graphics on Kindle fire and hours of fun & excitement. Once you start playing this game I can assure you that you will never put it down - Beautifully done!

While I was very happy to see that it was finally released for Amazon App store. Understandably I could not replace my existing Google play application or update it. I do not want to start all over again (I have scores in the millions with various characters unlocked with hours of play) so I prevented from the APP store replacing my game and kept the old version which I got free from Google play.

This game has the best rating of any Android game ever. Glad it is officially available via Amazon APP store.

Thanks to the developer for making a fantastic free game.

Download it now. It is superb game!

Cheers!

6,150 people found this helpful

|

Figura 2.1: Esempio di una review molto utile

 Walter H.

★☆☆☆☆ **Not good**

Reviewed in the United States on January 8, 2020

**Verified Purchase**

Not for me

|

 Chris

★☆☆☆☆ **Ok**

Reviewed in the United States on December 1, 2019

**Verified Purchase**

Ok

|

Figura 2.2: Esempio di review non utili



## 2.4.2 Analisi di Correlazione

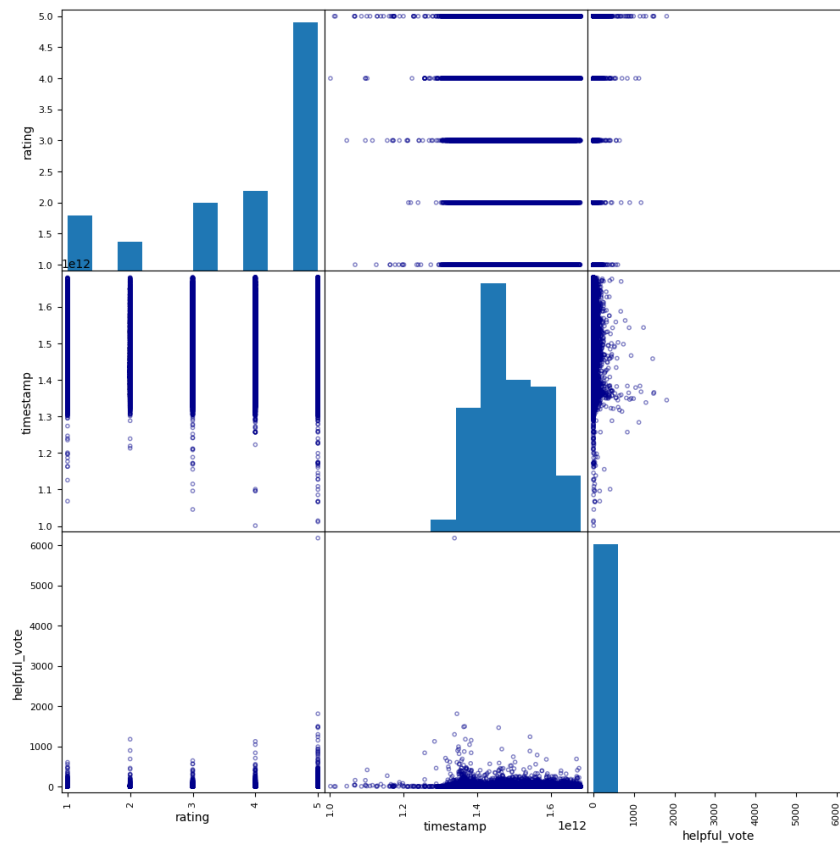


Figura 2.3: Correlazione tra variabili

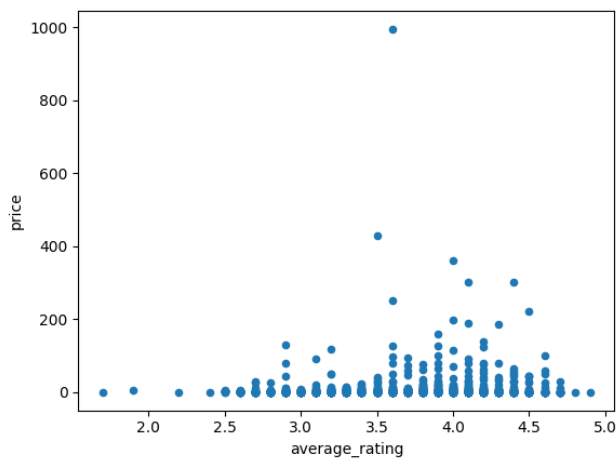


Figura 2.4: Correlazione tra "average\_rating" e "price"

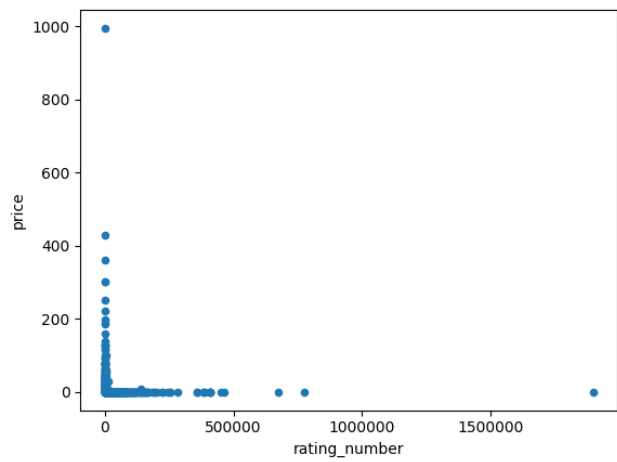


Figura 2.5: Correlazione tra "rating\_number" e "price"

Per quanto riguarda l'analisi di correlazione, dalla figura 2.3 si nota che non ci sono correlazioni significative tra le variabili. Tuttavia, si osserva una correlazione tra "helpful\_vote" e "rating"<sup>1</sup>.

La figura 2.4, che mostra la correlazione tra il rating medio e il prezzo di un articolo, indica che il rating degli articoli è indipendente dal loro prezzo.

Infine, dalla figura 2.5 emerge una correlazione leggermente negativa tra il prezzo e il numero di recensioni di un articolo, suggerendo che il numero di valutazioni tende a diminuire all'aumentare del prezzo.

## 2.4.3 Analisi Distribuzionale

### Distribuzione dei Rating

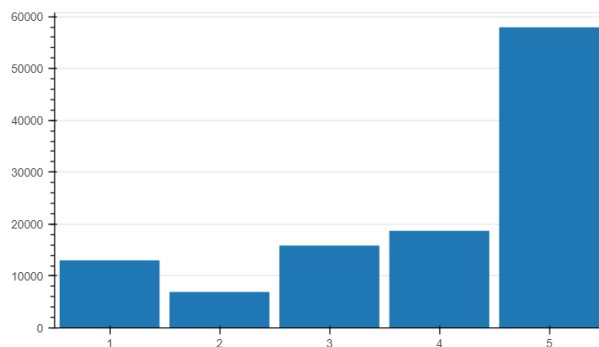


Figura 2.6: Distribuzione dei rating istogramma

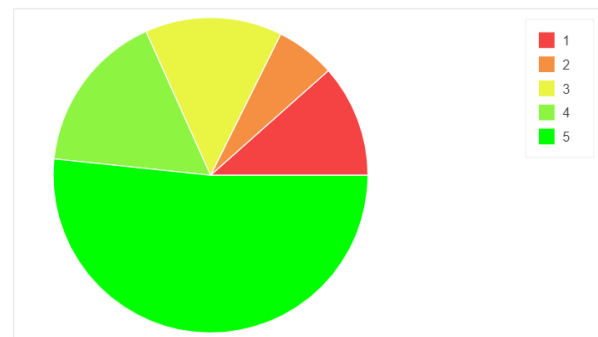


Figura 2.7: Distribuzione dei rating a torta

La distribuzione dei rating delle recensioni è fortemente sbilanciata verso il valore 5, indicando una predominanza di valutazioni molto positive. Questo sbilanciamento è significativo rispetto agli altri rating, che sono quasi equamente distribuiti, ad eccezione del rating 2, che risulta essere il meno popolare tra tutti. Questa tendenza potrebbe suggerire una soddisfazione generale degli utenti oppure una propensione a lasciare recensioni solo quando l'esperienza è stata estremamente positiva.

Analizzando più in dettaglio, si nota che i rating intermedi (3) e quelli bassi (1 e 2) sono presenti in

---

<sup>1</sup>al crescere del rating aumenta il numero di voti utili

quantità relativamente simili, tranne appunto per il rating 2, che è decisamente meno comune. Questo fenomeno può indicare che gli utenti sono più inclini a esprimere opinioni forti, specialmente se positive, mentre le valutazioni moderate sono meno frequenti.

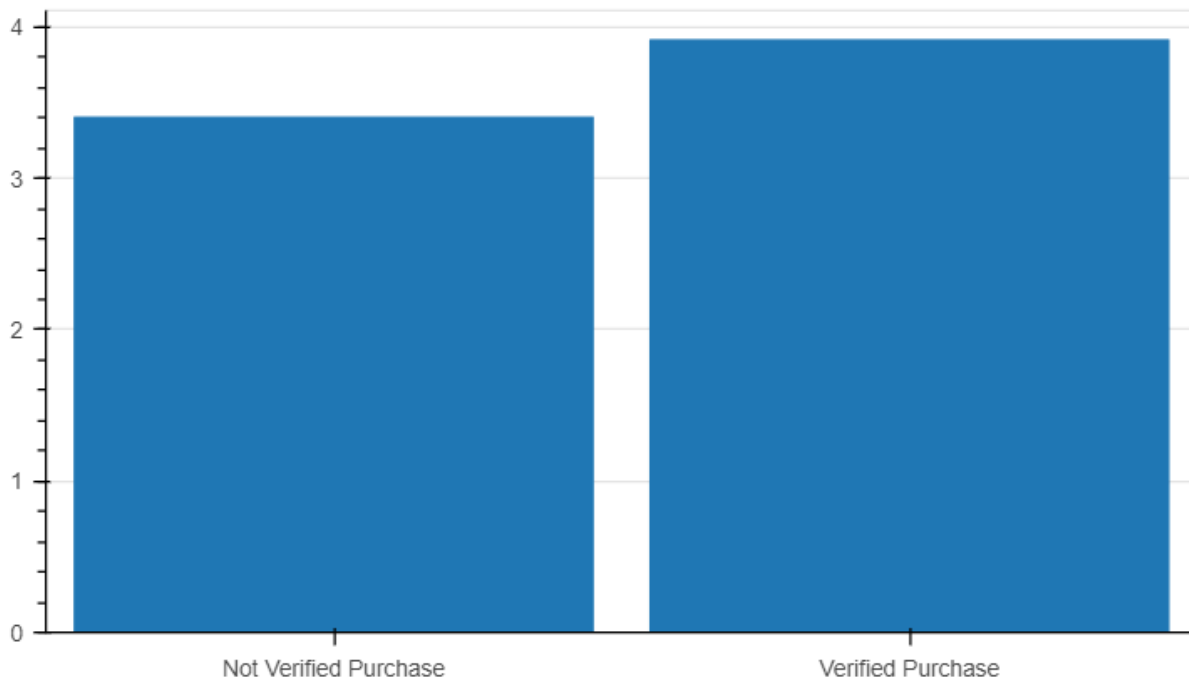


Figura 2.8: Media dei rating per "verified\_purchase"

In generale, osservando i dati, sembra emergere una tendenza secondo cui le recensioni associate ad acquisti verificati presentano mediamente valutazioni più elevate. Questo suggerisce che gli acquirenti che confermano la loro esperienza attraverso un acquisto effettivo tendono a dare valutazioni più positive rispetto a coloro che lasciano recensioni senza tale verifica.

Inoltre, la verifica degli acquisti può essere associata a una maggiore affidabilità delle recensioni stesse, poiché è più probabile che siano fornite da persone effettivamente coinvolte con il prodotto o il servizio in questione, ciò implica che gli utenti che hanno effettivamente provato un software, come indicato dagli acquisti verificati, mediamente sono molto soddisfatti dell'esperienza.

Tabella 2.2: Tabella che riporta il rating delle recensioni con più "helpful\_votes"

parent_asin	user_id	rating	#helpful_vote
B0086700CM	AHBZAI3V5AH4EGJQSCKPWREWPNAQ	5.0	6178
B008RA3X5E	AFOYRQEWLCJEFYV6I64L7S2WLETQ	5.0	1810
B00AB7HESI	AG7PCIWVZOWT5Q2WG7URPDKEZC2Q	5.0	1496
B00BQPEIT2	AFAHXM5DL4GKDNETZEW57LUIEWXQ	5.0	1485
B008XG1X18	AGI4OOLQVEGH3JBQCUET3OBTDLVA	5.0	1464
B007ZGO7EM	AEHOFUNZP6VT74RUDDCJ2VVIT56A	5.0	1297
B07FPS52BN	AF4FFRVBAKXVKJBFVKCMSTQYKGJQ	5.0	1238
B00AJAFNIA	AGYEEQLHIUB47CDI6X5M6VU4AT7Q	2.0	1177

Le recensioni con un gran numero di "helpful\_votes" tendono ad essere lunghe e dettagliate, riflettendo così un alto grado di coinvolgimento e interesse da parte degli utenti. Questo concetto è evidenziato anche nella figura 2.2, che mostra la recensione con il maggior numero di "helpful\_votes".

Si può notare che le recensioni con il maggior numero di "helpful\_votes" sono molto positive.

#### 2.4.4 Distribuzione Temporale

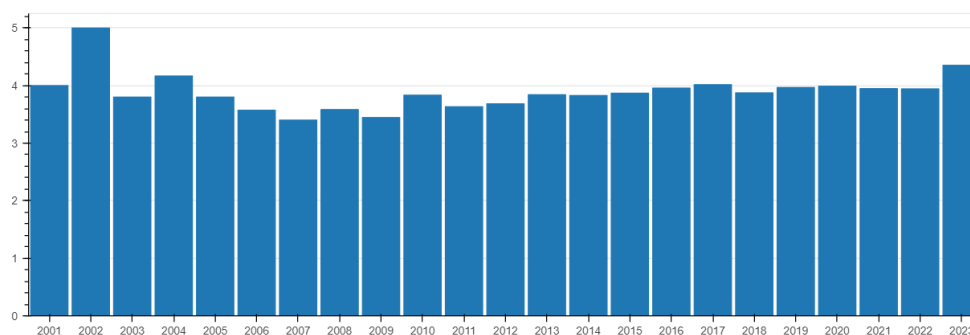


Figura 2.9: Media dei rating nel tempo

L'immagine 2.9 evidenzia che il rating medio dei software rimane generalmente stabile nel corso degli anni, come ci si potrebbe aspettare.



This is a very addictive **fun** and fast paced game. I must confess that I had side loaded this game several months ago when it was available on Google play and I **loved** it.

The game play is **superb** and has **very good** fast graphics on Kindle fire and hours of **fun & excitement**. Once you start playing this game I can assure you that you will never put it down - **Beautifully** done!

While I was very **happy** to see that it was finally released for Amazon App store. [...]

This game has the best rating of any Android game ever. **Glad** it is officially available via Amazon APP store.

Thanks to the developer for making a **fantastic** free game.

Download it now. It is **superb** game!

Cheers!

Figura 2.12: Analisi del sentiment di una review

Si noti come nella figura 2.12 sono state evidenziate le parole in base al loro sentiment e si vede che ci sono diverse parole con sentiment positivo.

Inoltre, un numero significativo di queste parole (o altre con sentiment positivo) sono presenti nella figura 2.11 che contiene le parole usate con più frequenza nelle recensioni, di conseguenza si nota un forte sentiment generale positivo nelle recensioni analizzate.

In conclusione, basandosi sia sulla distribuzione dei rating che delle parole si può affermare che il dataset è ricco di recensioni con sentiment positivo; dato che nel progetto avanzato sarà necessario classificare il sentiment delle recensioni, per migliorare i risultati dell'algoritmo di classificazione può essere una buona idea rimuovere alcune delle recensioni positive così da ottenere un dataset bilanciato.

## 3. Progetto Base

L'obiettivo del progetto base è lo sviluppo di un sistema di raccomandazione basato su collaborative filtering.

### 3.1 Creazione Dataset Surprise

Attualmente, il dataset assume la seguente forma:

Tabella 3.1: Dataset originale

user_id	parent_asin	rating
B00BQPEIT2	AG7PCIWVZOWT5Q2WG7URPDKEZC2Q	4.0
...	...	...
B008K6HN8I	AGYEEQLHIUB47CDI6X5M6VU4AT7Q	5.0

Questa configurazione è altamente efficiente nel memorizzare le recensioni: ogni recensione è identificata univocamente da una combinazione di "parent\_asin" e "user\_id", e a ciascuna recensione è associato un rating.

Tuttavia, per utilizzare algoritmi come KNN o Matrix Factorization, è necessario che la matrice assuma una diversa configurazione:

Tabella 3.2: Dataset Surprise

user_id	$item_1$	$item_2$	...	$item_n$
B00BQPEIT2	4.0	4.0	...	5.0
...	...	...	...	...
B008K6HN8I	5.0	3.0	...	2.0

La fase iniziale per sviluppare un Recommendation System basato su collaborative filtering, come illustrato in laboratorio, è costruire un Dataset Surprise: si tratta semplicemente di convertire il dataset dalla forma presentata nella matrice 3.1 in quella illustrata nella tabella 3.2.

## 3.2 Configurazione Ottimale KNN

Sono state esplorate diverse configurazioni per trovare quella ottimale di KNN sul dataset "Software".

Sono stati eseguiti diversi tentativi modificando gli iperparametri come il valore di  $k$ , la metrica di similarità e il tipo di KNN (item-based o user-based).

La configurazione più efficace individuata ha prodotto un Root Mean Squared Error (RMSE) di 1.13, con  $k = 40$ , utilizzando la metrica di similarità cosine e adottando l'approccio item-based.

## 3.3 Configurazione Ottimale Matrix Factorization

Analogamente a quanto fatto per KNN, è stato condotto un processo per individuare la configurazione ottimale dell'algoritmo di Matrix Factorization. Gli iperparametri presi in considerazione includono `n_factors`, `n_epochs`, `biased` e `random_state`.

La configurazione ottimale individuata ha prodotto un RMSE di 1.09, con i seguenti valori per gli iperparametri: `n_factors = 50`, `n_epochs = 20`, `biased = true` e `random_state = 0`.

## 3.4 Segmentazione degli utenti: K-Means

Per segmentare gli utenti è stato utilizzato l'algoritmo K-Means; l'idea di base dell'algoritmo è rappresentare ogni utente come un vettore in uno spazio a  $n$  dimensioni, dove  $n$  è il numero degli item. Ogni utente è quindi rappresentato dai rating assegnati agli  $n$  items<sup>1</sup>, e viene visualizzato come un punto nello spazio  $n$ -dimensionale.

L'algoritmo K-Means è un algoritmo di clustering che permette di radunare i punti in insiemi basati sulla loro vicinanza. Poiché l'obiettivo è raggruppare i punti basandosi sulla cosine similarity, il

---

<sup>1</sup>Non è necessario che un utente abbia valutato tutti gli item.



primo passo consiste nel normalizzare i valori dei vettori. Successivamente, si esegue l'algoritmo K-Means utilizzando la distanza euclidea come metrica di somiglianza.

Per trovare il miglior numero di cluster è stata usata la metrica di distorsione, derivata dall'inerzia.

Infatti:

$$\text{inerzia} = \sum_{i=1}^n d(x_i, c_j)^2 \quad (3.1)$$

Dove  $x_i$  rappresenta l' $i$ -esima istanza mentre  $c_j$  il cluster più vicino a quest'ultima e  $d(a, b)$  rappresenta la distanza euclidea tra il punto a e il punto b.

$$\text{distorsione} = \frac{\text{inerzia}}{k} = \frac{\sum_{i=1}^n d(x_i, c_j)^2}{k} \quad (3.2)$$

Dove  $k$  rappresenta il numero di clusters.

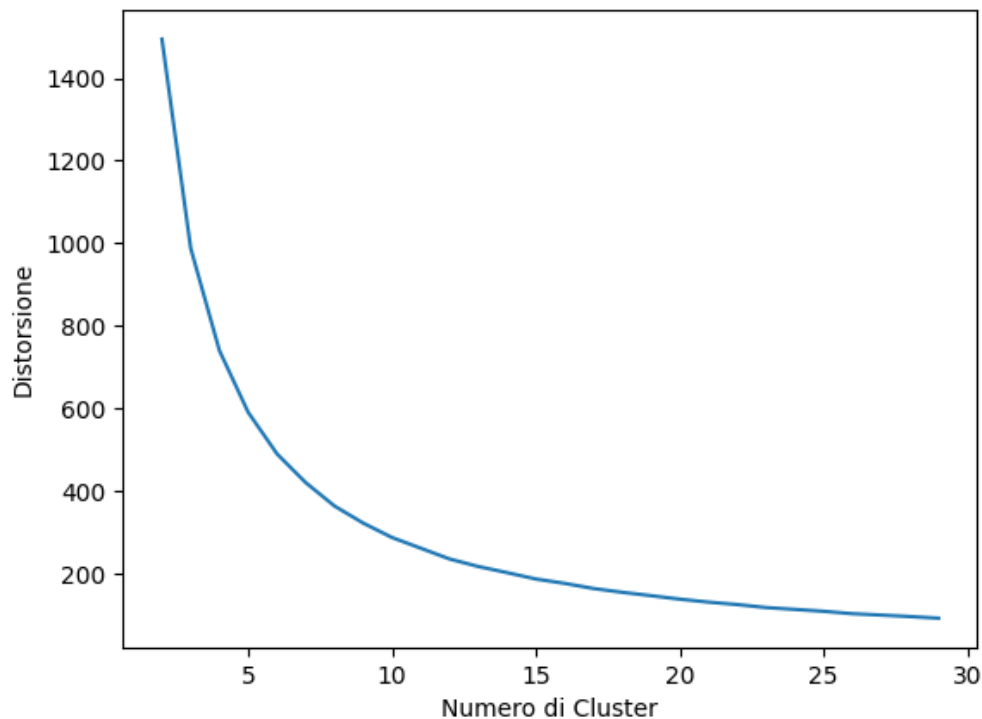


Figura 3.1: Elbow Method

Per trovare la miglior configurazione di K-Means (ovvero il miglior  $k$ ) è stato usato l'Elbow Method. Come si può notare dalla figura 3.1 il miglior numero di cluster è 15.

## 3.5 Collaborative Filtering Recommendation System

Per creare il sistema di raccomandazione, è stata sviluppata una funzione che, prendendo in input la lista degli utenti, la lista degli articoli, un numero  $nn$  e un modello utile a predire i rating mancanti, stampa a video per ogni utente i primi  $nn$  item raccomandati, basandosi sulla matrice dei rating.

È importante notare che non vengono raccomandati item che l'utente ha già valutato. La funzione scorre ogni utente e per ciascuno esamina gli articoli che non ha ancora valutato, assegnando loro un valore predetto del rating che viene calcolato utilizzando il modello selezionato. Successivamente, ordina i nuovi rating predetti in ordine decrescente e seleziona i primi  $nn$  item con i rating più alti, raccomandandoli all'utente.

## 3.6 KNN vs Matrix Factorization

### 3.6.1 Confronto RMSE

Algoritmo	RMSE
KNN	1.13
Matrix Factorization	1.09

Tabella 3.3: RMSE di KNN e Matrix Factorization

L'algoritmo Matrix Factorization, con la sua configurazione ottimale, ha prodotto un RMSE leggermente migliore di quello prodotto da KNN.

Matrix Factorization è noto per performare particolarmente bene anche su matrici sparse al contrario di KNN e ciò giustifica i risultati ottenuti.

### 3.6.2 Confronto Raccomandazioni

Di seguito viene illustrato come cambiano le raccomandazioni per un utente con i due sistemi di raccomandazione sviluppati.

Sono state analizzati un utente e le sue preferenze.

Item Code	Title	Rating	Tipologia
B078H2GSLC	Dillard	5	Applicazione
B086HSTQD3	Maps For Google	5	Applicazione
B0080S37DG	Smart System Info	5	Applicazione
B00SHV617K	The Island Castaway: Lost World	4	Gioco
B00IEG0JOY	OneDrive	4	Applicazione

Tabella 3.4: Articoli recensiti positivamente dall'utente

Item Code	Title	Predizione	Tipologia
B00EX4UXGU	Pocketmine server	5	Gioco
B06VWV5SKS	AirPlayMirror Receiver	4.5	Applicazione
B006RA0ZRU	Dino Digger	4.35	Gioco
B008GVC35U	Fraction calculator	4	Applicazione
B00PD79BR8	RPG Clicker	4	Gioco

Tabella 3.5: Articoli raccomandati usando KNN

Item Code	Title	Predizione	Tipologia
B00B5L5JRM	Zoom - One Platform to Connect	3.46	Applicazione
B00KZP2DTQ	Instagram	3.41	Applicazione
B004DLPXAO	Kindle for Android	3.40	Applicazione
B00F9F1G4U	Speedtest by Ookla	3.32	Applicazione
B01DUQOSRQ	Xfinity Stream (live TV)	3.30	Applicazione

Tabella 3.6: Articoli raccomandati usando Matrix Factorization

Analizzando gli articoli raccomandati da KNN e Matrix Factorization si può notare dalle tabelle 3.4, 3.5 e 3.6 che i due sistemi di raccomandazione hanno diverse proprietà:

- **Novità:** suggerire prodotti che un utente potrebbe non aver ancora visto; è soddisfatta da entrambi i sistemi di raccomandazione in quanto, come citato in precedenza, se un utente ha già valutato un articolo, esso non può essere raccomandato.
- **Rilevanza:** raccomandare prodotti che un utente ritiene più interessanti in base alle informazioni disponibili; difatti entrambi i Recommender System consigliano articoli che sono allineati alle preferenze espresse dall'utente, ad esempio Matrix Factorization consiglia Xfinity Stream (live TV), un servizio di streaming video simile a Dillard che è un prodotto per cui l'utente ha espresso una forte preferenza. KNN consiglia Pocketmine server che è un videogame proprio come The Island Castaway: Lost World.
- **Serendipità:** suggerire prodotti "inaspettati" che potrebbero piacere all'utente; entrambi i sistemi di raccomandazione (KNN e Matrix Factorization) rispettano questa proprietà. Infatti ambo i sistemi possono proporre prodotti che non sono perfettamente in linea con le preferenze esplicite dell'utente, ma che potrebbero comunque risultare graditi, arricchendo l'esperienza di scoperta di nuovi contenuti.

## 4. Progetto Intermedio

L'obiettivo del progetto intermedio è lo sviluppo di un sistema di raccomandazione content based.

### 4.1 Riferimenti Teorici

#### 4.1.1 Tokenizzazione

Per poter creare un sistema di raccomandazione content based, è necessario analizzare le componenti testuali degli articoli.

In questo progetto verranno esaminati il titolo e la descrizione di ogni articolo.

Per processare gli attributi testuali, la prima cosa da fare è creare una tokenizzazione di questi ultimi.

Il processo di tokenizzazione consiste nel suddividere una stringa<sup>1</sup> in token.

La suddivisione più semplice consiste nel dividere una stringa nelle sue parole, in questo caso, ogni parola rappresenterebbe un token.

Esistono delle tecniche che permettono di ridurre il numero di token, semplificando così la tokenizzazione finale.

#### Stemming

Lo Stemming è una tecnica molto utile per ridurre il numero di token presenti in una frase: permette dunque di risparmiare memoria e di avere un embedding più significativo in un secondo momento. L'idea di base dello Stemmer è quella di ricevere una parola (un token) in input e restituire in output la radice di quella parola.

In questo modo, il significato generale della frase rimane invariato, ma parole simili vengono ridotte alla stessa radice. Questo semplifica la tokenizzazione generale, riducendo l'overhead e mantenendo il significato originale della frase.

---

<sup>1</sup>Ad esempio la descrizione di un prodotto.

## Lemmatizing

Il Lemmatizer ha uno scopo simile a quello dello Stemmer, ma con una differenza fondamentale: mentre lo Stemmer restituisce la radice di una parola, spesso priva di significato logico, il Lemmatizer restituisce una parola completa e corretta (in inglese) con senso compiuto.

Il Lemmatizer si basa sul concetto di lemma, che è una forma base o rappresentativa di un insieme di parole correlate. Ad esempio, un lemma può rappresentare tutte le varianti di una parola, comprese le diverse coniugazioni e declinazioni.

In pratica, il Lemmatizer trasforma parole diverse ma simili nella loro forma base, il lemma, che funge da rappresentante per l'intero gruppo di parole derivate.

### 4.1.2 Embedding

Il secondo passo per utilizzare i dati testuali è creare degli embedding, in quanto il testo non elaborato non può essere direttamente utilizzato dagli algoritmi di Machine Learning.

Un embedding è una rappresentazione numerica<sup>2</sup> di un dato testuale.

In questo caso, gli embedding verranno costruiti a partire dai token generati dalle tecniche citate in precedenza.

## Bag of Words

Il primo metodo che è stato utilizzato per la creazione degli embedding è Bag of Words (BoW).

Il meccanismo è molto semplice:

1. Inizialmente, vengono uniti tutti i token prodotti nella fase precedente per creare un vocabolario dei token (un insieme che contiene tutti i token).
2. Dopodichè, per ogni descrizione, vengono contate le volte in cui ogni token appare in essa.
3. L'embedding di una descrizione è un vettore in cui la cella  $i$ -esima contiene il numero di volte che il token  $i$ -esimo è presente nella descrizione.

---

<sup>2</sup>Solitamente un vettore.

descrizione	<i>foglio</i>	...	<i>note</i>
Blocco note con fogli colorati. Fogli A4.	2	...	1
...	...	...	...
Spartito per imparare le note musicali.	0	...	1

Tabella 4.1: Esempio di embedding con BoW

## TF-IDF

Anche TF-IDF è un metodo che permette di creare embedding che si basa sulla frequenza delle parole. In particolare, TF-IDF assegna uno score ad ogni parola, calcolato correlando la frequenza della parola all'interno di un testo con la sua importanza. TF-IDF è composto da due componenti:

1. **Term Frequency - TF**: indica quante volte una parola compare all'interno di un testo.
2. **Inverse Document Frequency - IDF**: IDF misura l'importanza di una parola considerando la sua rarità nell'insieme dei documenti analizzati.

La Term Frequency si può rappresentare in 2 modi:

- Term frequency relativa alla lunghezza del testo:

$$TF(t, d) = \frac{\text{Numero di volte che il termine } t \text{ appare nel documento } d}{\text{Numero di parole nel documento } d} \quad (4.1)$$

- Term frequency scalata logaritmicamente:

$$TF(t, d) = \log(1 + \text{Numero di volte che il termine } t \text{ appare nel documento } d) \quad (4.2)$$

Per calcolare IDF viene introdotta la seguente formula:

$$IDF(t, D) = \log \frac{N}{|d \in D : t \in d|} \quad (4.3)$$

Dove  $N$  è il numero totale dei documenti, mentre il denominatore rappresenta il numero dei documenti che contengono il token  $t$ .

Dalla combinazione di questi strumenti matematici nasce la formula di TF-IDF:

$$TF-IDF = TF(t, d) \cdot IDF(t, D) \quad (4.4)$$

### 4.1.3 Transfomer

Mentre con TF-IDF e BoW è possibile costruire vettori sparsi<sup>3</sup>, questi approcci presentano seri svantaggi. Uno di questi riguarda la perdita dell'ordine delle parole, che comporta l'assenza di contesto e di significato.

Inoltre, essendo vettori sparsi, occupano molto spazio di memoria a causa della loro lunghezza.

I Transformer sono l'ultimo metodo che verrà presentato per creare degli embedding.

Questa tecnologia si basa sulle reti neurali e offre diverse funzionalità come ad esempio il meccanismo di attenzione, che consente di catturare efficacemente il contesto e le relazioni tra le parole.

Gli embedding creati con i Transformer sono decisamente più significativi, efficienti in termini di memoria e tendenzialmente producono risultati migliori quando utilizzati con algoritmi di Machine Learning.

## 4.2 Processamento Attributi Testuali con NLP

Per analizzare gli attributi testuali è stata creata una funzione chiamata "text\_processing" che riceve in input una lista di stringhe e un parametro "method", utile per scegliere se utilizzare uno Stemmer 4.1.1 o un Lemmatizer 4.1.1.

La funzione restituisce in output una matrice in cui ogni riga è una lista di token che rappresentano la descrizione corrispondente.

I token sono stati creati seguendo questo processo: si considera una descrizione, la si divide in base agli spazi, si estraggono le singole parole e, a seconda del metodo scelto, si applica il Lemmatizer o lo Stemmer per convertire le parole in token.

Se una parola è considerata una stop word<sup>4</sup>, allora viene esclusa.

Nelle stop word sono stati aggiunti i segni di punteggiatura in quanto sono poco significativi semanticamente e processarli richiede memoria e potenza computazionale.

Per processare il titolo e la descrizione di un prodotto insieme, sono state unite le due colonne della matrice, in modo tale da processare le informazioni contemporaneamente.

Uno sviluppo futuro potrebbe essere analizzare prima i titoli e in un secondo momento le descri-

---

<sup>3</sup>Con tanti 0.

<sup>4</sup>Le stop word sono parole "comuni" che non aggiungono significato semantico alla frase.



zioni, ottenendo due embedding: uno per i titoli e l'altro per le descrizioni.

Con questa metodologia sarebbe anche possibile pesare diversamente i titoli e le descrizioni, rendendo i titoli più importanti. Tuttavia, questo aumenterebbe l'overhead della rappresentazione e quindi l'utilizzo di memoria.

### 4.3 Predizione del Rating per Utente

Dopo aver costruito i vari embedding, per testarne l'efficacia, sarà necessario recuperare la lista degli item che sono stati valutati da un utente; per ogni item recuperare l'embedding dei suoi attributi testuali (titolo e descrizione).

Quindi si otterrà una matrice del tipo:

Tabella 4.2: Matrice che contiene gli embedding degli attributi testuali

$token_1$	...	$token_n$	user_id	parent_asin	rating
3	...	0	B00BQPEIT2	AG7PCIWVZOWT5Q2WG7URPDKEZC2Q	4.0
...	...	...	...	...	...
0	...	4	B00BQPEIT2	AGYEEQLHIUB47CDI6X5M6VU4AT7Q	5.0

Il passo successivo consiste nel dividere le righe della matrice appena ottenuta tra righe di train e di test; Il train set sarà utilizzato per allenare un modello (come KNN), mentre il test set verrà utilizzato per testarne le performance e calcolare l'RMSE riferito ad un utente.

Per calcolare l'RMSE del sistema di raccomandazione viene introdotta la seguente formula:

$$RMSE = \frac{\sum_{i=1}^n RMSE_i}{n} \quad (4.5)$$

Dove  $RMSE_i$  è l'RMSE relativo all'utente  $i$ -esimo, e  $n$  è il numero totale di utenti.

### 4.4 Valutazione dei Risultati

In questa sezione verranno presentati i risultati ottenuti.

Metodo usato	RMSE
Bow - Stemming	1.13
Bow - Lemmatizing	1.13
TF-IDF - Stemming	1.12
TF-IDF - Lemmatizing	1.12
Transformers	1.12

Tabella 4.3: RMSE arrotondati alla seconda cifra decimale

Nonostante siano stati testati vari modelli di Transformer, i risultati ottenuti sono pressochè identici e alla fine è stato scelto il modello "all-mpnet-base-v2". Pertanto, si può concludere che i Transformer non hanno portato a un miglioramento significativo delle performance rispetto a BoW e TF-IDF.

## 4.5 Confronto tra Progetto Base ed Intermedio

Di seguito viene riportata una tabella contenente gli RMSE calcolati nei due sistemi di raccomandazione.

Per quanto riguarda il sistema di raccomandazione content based viene riportato solamente l'RMSE ottenuto con i Transformer, in quanto non ci sono differenze significative nell'errore rispetto alle altre tecniche utilizzate.

Sistema di Raccomandazione	RMSE
Collaborative Filtering - KNN	1.13
Collaborative Filtering - Matrix Factorization	1.09
Content-Based - Transformer	1.12

Tabella 4.4: RMSE dei sistemi di raccomandazione implementati

Come si può notare dalla tabella 4.4, l'uso degli attributi testuali non ha portato a un miglioramento significativo nel sistema di raccomandazione content based. Questo risultato è probabilmente do-

vuto alla grande diversità del software disponibile su Amazon. Esistono infatti numerosi software diversi tra loro, con descrizioni decisamente differenti, rendendo difficile catturare le preferenze degli utenti.

In questo contesto catturare le preferenze di un utente attraverso l'analisi testuale di titoli e descrizioni dei prodotti può rivelarsi un task non banale.

## 5. Progetto Avanzato

L'obiettivo del progetto avanzato è effettuare una sentiment analysis sulle recensioni dei prodotti.

### 5.1 Preparazione dei Dati

Dato che il dataset analizzato non è etichettato per quanto riguarda il sentiment<sup>1</sup> per etichettarlo è stata effettuata una conversione da rating a classe.

Le recensioni sono state classificate in questo modo:

- Se il rating della recensione è pari a 4 o 5, allora la recensione ha sentiment positivo
- Se il rating della recensione è pari a 3, allora la recensione ha sentiment neutro
- Se il rating della recensione è pari a 1 o 2, allora la recensione ha sentiment negativo

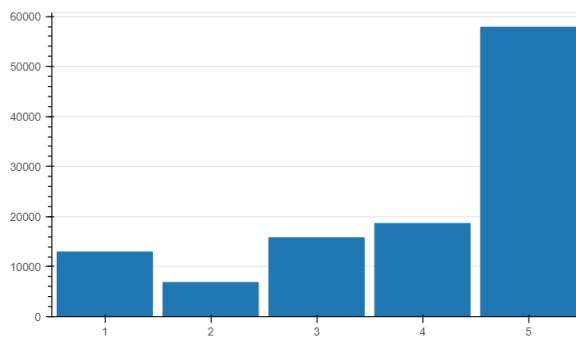


Figura 5.1: Distribuzione originale

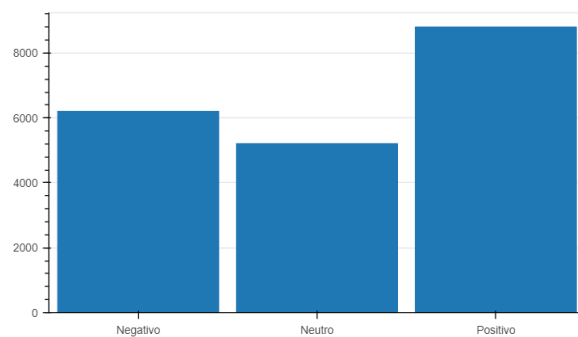


Figura 5.2: Distribuzione bilanciata

Come illustrato nella figura 5.1 il dataset è particolarmente sbilanciato verso i rating positivi (rating 4 e 5). Per bilanciare il dataset, è stato ridotto il numero di recensioni positive, con l'obiettivo di migliorare i risultati ottenuti nel processo di classificazione del sentiment, che predilige dataset ben bilanciati. Il nuovo dataset mostrato nella figura 5.2 presenta un numero di rating positivi, negativi e neutri pressochè uguale.

---

<sup>1</sup>Le recensioni hanno solo il rating, ma non hanno una classificazione in base al sentiment

## 5.2 Processamento Attributi Testuali con NLP

Analogamente a quanto fatto nel progetto intermedio, il primo passo per sviluppare il progetto avanzato ed effettuare una sentiment analysis delle recensioni consiste nel processare gli attributi testuali utilizzando tecniche di elaborazione del linguaggio naturale (NLP).

Per farlo verranno sfruttate le stesse tecniche utilizzate nel progetto intermedio nella sezione 4.1.

In questo caso, per ottenere il rispettivo embedding, sarà necessario processare gli attributi "titolo" e "descrizione" di ogni recensione. Successivamente, si potrà procedere con la classificazione del sentiment.

## 5.3 Valutazione dei Risultati

Per effettuare effettuare l'analisi del sentiment con la massima accuratezza possibile, sono state sperimentate tutte le combinazioni di Tokenizer ed Embedder presentate in laboratorio.

Questo approccio ha permesso di valutare l'efficacia di diverse tecniche di elaborazione del linguaggio naturale.

In questo modo, è stato possibile valutare le performance dei vari modelli di classificazione del sentiment.

Di seguito vengono riportate le tecniche utilizzate:

1. **BoW Lemmating**
2. **BoW Stemming**
3. **TF-IDF Lemmating**
4. **TF-IDF Stemming**
5. **Transformer** (modello: "ashok2216/gpt2-amazon-sentiment-classifier-V1.0")

Per trovare la combinazione migliore, sono state calcolate le metriche di **Precision**, **Recall**, **F1-score**, **Accuracy** e sono state create le corrispondenti **matrici di confusione**.

### 5.3.1 TF-IDF

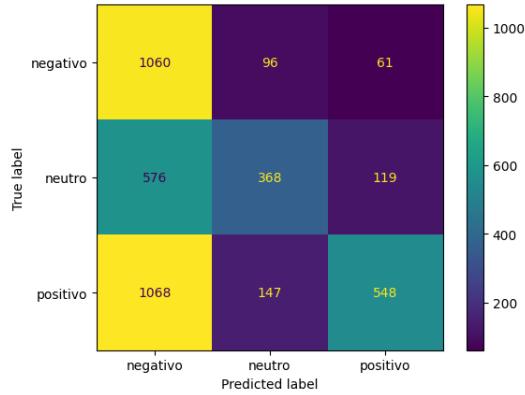


Figura 5.3: Matrice di confusione TF-IDF  
Lemmatizing

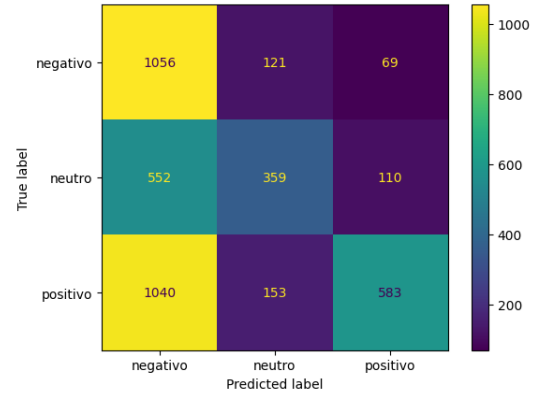


Figura 5.4: Matrice di confusione TF-IDF  
Stemming

	Precision	Recall	F1-Score	Support
<b>negativo</b>	0.39	0.87	0.54	1217
<b>neutro</b>	0.60	0.35	0.44	1063
<b>positivo</b>	0.75	0.31	0.44	1763
<b>Accuracy</b>			0.49	4043
<b>Macro Avg</b>	0.58	0.51	0.47	4043
<b>Weighted Avg</b>	0.60	0.49	0.47	4043

Tabella 5.1: Metriche TF-IDF Lemmatizing

	Precision	Recall	F1-Score	Support
<b>negativo</b>	0.40	0.85	0.54	1246
<b>neutro</b>	0.57	0.35	0.43	1021
<b>positivo</b>	0.77	0.33	0.46	1776
<b>Accuracy</b>			0.49	4043
<b>Macro Avg</b>	0.58	0.51	0.48	4043
<b>Weighted Avg</b>	0.60	0.49	0.48	4043

Tabella 5.2: Metriche TF-IDF Stemming

Come mostrato dalle figure 5.3 e 5.4 e dalle tabelle 5.1 e 5.2, TF-IDF si è dimostrata la strategia meno efficace sia con Lemmatizing che con lo Stemming.

TF-IDF ha prodotto risultati buoni sulla classe negativa ma pessimi per quanto riguarda la classificazione del sentiment positivo.

Tendenzialmente la classe neutra è quella più difficile da identificare quindi i risultati prodotti per il sentiment neutro possono essere considerati accettabili.

### 5.3.2 BoW

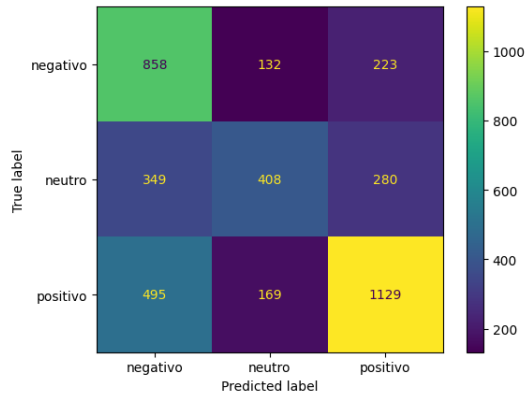


Figura 5.5: Matrice di confusione BoW  
Lemmatizing

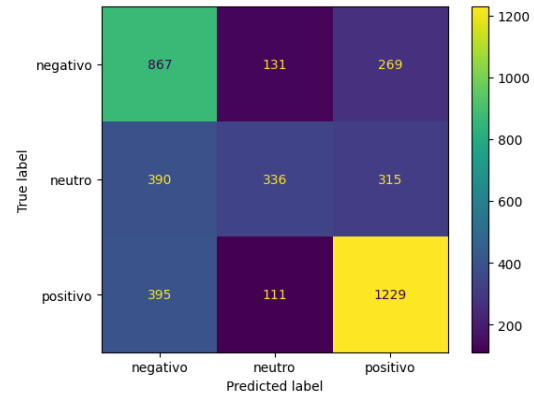


Figura 5.6: Matrice di confusione BoW  
Stemming

	Precision	Recall	F1-Score	Support
<b>negativo</b>	0.50	0.71	0.59	1213
<b>neutro</b>	0.58	0.39	0.47	1037
<b>positivo</b>	0.69	0.63	0.66	1793
<b>Accuracy</b>			0.59	4043
<b>Macro Avg</b>	0.59	0.58	0.57	4043
<b>Weighted Avg</b>	0.61	0.59	0.59	4043

Tabella 5.3: Metriche BoW Lemmatizing

	Precision	Recall	F1-Score	Support
<b>negativo</b>	0.52	0.68	0.59	1267
<b>neutro</b>	0.58	0.32	0.42	1041
<b>positivo</b>	0.68	0.71	0.69	1735
<b>Accuracy</b>			0.60	4043
<b>Macro Avg</b>	0.59	0.57	0.57	4043
<b>Weighted Avg</b>	0.61	0.60	0.59	4043

Tabella 5.4: Metriche BoW Stemming

Come mostrato dalle figure 5.5 e 5.6 e dalle tabelle 5.3 e 5.4, il modello BoW produce generalmente risultati migliori rispetto a TF-IDF. Per quanto riguarda la predizione del sentiment neutro, si nota una leggera differenza tra Lemmatizing e Stemming, con la lemmatizzazione che performa leggermente meglio. Inoltre, il sentiment positivo e negativo vengono classificati in modo decisamente più soddisfacente rispetto a TF-IDF.

### 5.3.3 Transformer

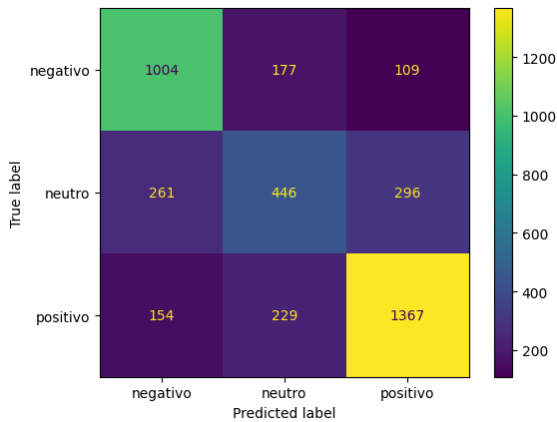


Figura 5.7: Matrice di confusione Transformer

	Precision	Recall	F1-Score	Support
<b>negativo</b>	0.71	0.78	0.74	1290
<b>neutro</b>	0.52	0.44	0.48	1003
<b>positivo</b>	0.77	0.78	0.78	1750
<b>Accuracy</b>				0.70
<b>Macro Avg</b>	0.67	0.67	0.67	4043
<b>Weighted Avg</b>	0.69	0.70	0.69	4043

Tabella 5.5: Metriche Transformer

Come si può notare dalla figura 5.7 e dalla tabella 5.5, l'utilizzo dei Transformer si rivela la strategia vincente, producendo risultati migliori nella classificazione di tutte le categorie. In particolare, l'incremento delle performance nella classificazione del sentiment neutro, sebbene lieve, è significativo, poiché questa categoria è risultata la più difficile da classificare anche con gli altri metodi. Le tabelle che riportano le metriche calcolate con le varie strategie confermano ulteriormente questi risultati: i Transformer ottengono i valori più alti del F1-score in tutte e tre le classi.

Inoltre, utilizzando i Transformer i tempi di esecuzione durante la fase di train e test del modello sono decisamente più bassi: ciò accade perchè i Transformer producono un embedding che è decisamente più piccolo rispetto a quello ottenuto con BoW e TF-IDF. Questo perchè gli embedding prodotti da BoW e TF-IDF hanno un numero di colonne pari al numero di token prodotti<sup>2</sup>, mentre i Transformer producono embedding che hanno una dimensionalità pari al numero di neuroni dell'ultimo layer della rete neurale su cui si basano<sup>3</sup>.

<sup>2</sup>In questo caso più di 80k colonne.

<sup>3</sup>In questo caso 770 colonne.



## 6. Conclusione

Di seguito vengono riportati i principali risultati ottenuti.

1. Per quanto riguarda la **distribuzione dei dati** si è notato<sup>1</sup> che il dataset è fortemente sbilanciato sul positivo; questa considerazione è stata fondamentale nel progetto avanzato: il dataset è stato ribilanciato per evitare che il task di classificazione producesse risultati scadenti.
2. Il sistema di raccomandazione **collaborative filtering** implementato utilizzando Matrix Factorization ottiene un RMSE<sup>2</sup> leggermente migliore di quello che usa KNN. Inoltre entrambi i sistemi di raccomandazione soddisfano le proprietà di novità, rilevanza e serendipità.
3. Il sistema di raccomandazione **content based** ottiene un RMSE leggermente peggiore di quello ottenuto con il sistema basato su collaborative filtering. Questo è probabilmente dovuto al fatto che il software ha descrizioni molto diversificate e di conseguenza non è facile basarsi su di esse per fare raccomandazioni.
4. Infine, per quanto riguarda la **sentiment analysis**, le metriche utilizzate per valutare i modelli sono state la matrice di confusione e Precision, Recall e F1-score; i risultati migliori sono stati ottenuti utilizzando i Transformer. Inoltre, utilizzando i Transformer i tempi di esecuzione durante la fase di train e test del modello sono decisamente più bassi: ciò accade perchè i Transformer producono un embedding che è decisamente più piccolo rispetto a quello ottenuto con BoW e TF-IDF. Questo perchè gli embedding prodotti da BoW e TF-IDF hanno un numero di colonne pari al numero di token prodotti, mentre i transformer producono embedding che hanno una dimensionalità pari al numero di neuroni dell'ultimo layer della rete neurale su cui si basano.

Nel complesso si può affermare che i risultati e le conclusioni ottenute sono soddisfacenti ed in linea con le nostre aspettative.

---

<sup>1</sup>Sia dalla distribuzione dei rating che delle parole.

<sup>2</sup>Di poco superiore ad 1.