
RECOMMANDER SYSTEM CON COLLABORATIVE FILTERING

Università degli Studi di Salerno

Prof. Giuseppe Polese
Dott. Giandomenico Solimando
Dott.ssa Eleonora Calò
Alessandro Clericuzio
Francesco Pio di Pippa
Christion Lo Conte

Contents

1	Introduzione	3
1.1	Reasearch Questions	3
1.2	Struttura del documento	3
2	Stato dell'arte	4
2.1	Overview sui sistemi di raccomandazione	4
2.2	Tecnica del Collaborative Filtering	4
2.3	Problema del cold start	5
2.4	Tecniche di Clustering	6
3	Metodologia	6
3.1	MovieLens 100k Dataset	7
3.1.1	Struttura dei dati	7
3.1.2	Statistiche principali	7
3.2	Pipeline Sperimentale	7
3.2.1	Preprocessing e Suddivisione del Dataset	7
3.2.2	Costruzione della Matrice dei Rating e di Similarità	8
3.2.3	Calcolo delle Predizioni	8
3.3	Approcci a Confronto	9
3.3.1	User-Based	9
3.3.2	Hybrid	9
3.4	Metriche di valutazione	10
4	Risultati	12
4.1	Statistiche descrittive della matrice	12
4.2	Analisi delle similarità	12
4.3	Valutazione globale delle predizioni nello scenario ideale (senza cold start) . .	13
4.3.1	Approccio User-Based	14
4.3.2	Approccio Hybrid	14
4.4	Valutazione Top-K	15
4.4.1	Approccio User-Based	15
4.4.2	Approccio Hybrid	17
4.5	Scenario con Cold Start (nuovi utenti)	17
4.5.1	Metodologia	18
4.5.2	Selezione numero di cluster	18
4.5.3	Metriche di valutazioni finali	19
4.5.4	Visualizzazione 3D	20
4.6	Scenario con Cold Start (nuovi item)	20
4.6.1	Metodologia	20
4.6.2	Metriche di similarità	20
4.6.3	Selezione del numero di cluster	20
4.6.4	Visualizzazione 3D	21
4.6.5	Conclusioni	21
5	Conclusioni	22

1 Introduzione

Negli ultimi anni, i sistemi di raccomandazione hanno assunto un ruolo fondamentale nella personalizzazione dell'esperienza utente in svariati ambiti applicativi, dall'e-commerce ai servizi di streaming, passando per piattaforme social ed educative. Questi sistemi mirano a fornire suggerimenti mirati e rilevanti basandosi sull'analisi delle preferenze individuali e dei comportamenti precedenti degli utenti. Tra le tecniche più diffuse e consolidate vi è il Collaborative Filtering (CF), che propone raccomandazioni sfruttando le similarità comportamentali tra utenti.

Tuttavia, il CF presenta alcune criticità significative, tra cui il problema del cold start, ossia la difficoltà nel generare raccomandazioni efficaci per nuovi utenti o nuovi item che non hanno ancora ricevuto sufficienti interazioni. Questo lavoro propone un'analisi di un sistema User-Based Collaborative Filtering arricchito con tecniche avanzate di clustering per superare le limitazioni del cold start, utilizzando rappresentazioni semantiche continue delle caratteristiche utente e item. Inoltre, è stato esplorato un approccio Hybrid, che integra metodologie User-Based e Item-Based Collaborative Filtering, al fine di combinare la flessibilità nella personalizzazione con la robustezza e la scalabilità nella generazione di raccomandazioni.

1.1 Research Questions

Alla luce delle problematiche evidenziate, il presente lavoro si propone di rispondere alle seguenti **domande di ricerca**:

1. **Quali sono le principali criticità e limitazioni del Collaborative Filtering e come possono essere mitigate attraverso strategie ibride o tecniche di pre-elaborazione dei dati?**
2. **Quanto è efficace il clustering basato su metadati utente nel mitigare il problema del *cold start* in un sistema di raccomandazione user-based?**

1.2 Struttura del documento

- **Capitolo 2: Stato dell'arte**

Analizza le tecniche di raccomandazione esistenti, in particolare il Collaborative Filtering, il problema del cold start e le metodologie di clustering. Fornisce una panoramica teorica dettagliata delle diverse tecniche disponibili e degli approcci più recenti per affrontare le principali criticità.

- **Capitolo 3: Metodologia**

Descrive il dataset MovieLens 100k utilizzato per gli esperimenti e illustra la pipeline sperimentale adottata, dalla fase di preprocessing fino al calcolo delle predizioni. Presenta inoltre i dettagli degli approcci implementati, ossia User-Based e Hybrid, e le metriche scelte per la valutazione delle performance.

- **Capitolo 4: Risultati**

Riporta i risultati degli esperimenti condotti in vari scenari, inclusi quelli con e senza cold start. Analizza statistiche descrittive, matrici di similarità e valutazioni globali

delle predizioni effettuate. Include inoltre confronti dettagliati tra i due approcci adottati (User-Based e Hybrid) e una valutazione approfondita delle raccomandazioni Top-K.

- **Capitolo 5: Conclusioni**

Riassume i principali risultati ottenuti, confermando l'efficacia dell'approccio proposto nell'affrontare il cold start. Suggerisce inoltre possibili sviluppi futuri del lavoro, come l'utilizzo di tecniche avanzate di deep learning o approcci ibridi più complessi.

2 Stato dell'arte

2.1 Overview sui sistemi di raccomandazione

Un **sistema di raccomandazione** [1] è uno strumento progettato per facilitare l'accesso personalizzato a grandi quantità di informazioni, suggerendo contenuti, prodotti o servizi che meglio rispondono ai gusti e alle esigenze di un singolo utente. Questi sistemi analizzano dati provenienti da diverse fonti (come comportamenti passati, preferenze esplicite, interazioni sociali o caratteristiche degli oggetti) per generare raccomandazioni rilevanti e pertinenti. Il loro utilizzo è particolarmente vantaggioso in contesti caratterizzati da un'ampia offerta, come l'e-commerce, l'intrattenimento digitale o le piattaforme di informazione, dove aiutano a ridurre il sovraccarico informativo, migliorare l'esperienza utente e aumentare l'efficacia delle interazioni, sia in termini di soddisfazione dell'utente che di obiettivi di business (es. conversioni, fidelizzazione).

2.2 Tecnica del Collaborative Filtering

Il *Collaborative Filtering* (CF) è una delle tecniche più popolari e diffuse nei sistemi di raccomandazione. La sua idea di base consiste nel suggerire oggetti (film, libri, prodotti, ecc.) a un utente sulla base delle preferenze espresse da altri utenti simili [2]. A differenza dell'approccio content-based [3], il CF non richiede conoscenza esplicita delle caratteristiche degli item, ma sfrutta unicamente il comportamento osservato (es. voti, acquisti, clic).

Le tecniche CF si dividono principalmente in tre categorie: **memory-based**, **model-based** e **ibridi** [4]. Gli approcci *memory-based*, come il *user-based* e *item-based*, calcolano direttamente le similitudini tra utenti o tra oggetti basandosi sulla matrice di valutazioni. Questi metodi sono semplici da implementare, ma soffrono di problemi di scalabilità e scarsa densità dei dati. Le tecniche *model-based* costruiscono un modello predittivo usando algoritmi di apprendimento automatico, come reti bayesiane, clustering, regressione, modelli a fattori latenti (es. SVD) e metodi basati su processi decisionali di Markov. Infine, i sistemi *ibridi* combinano CF con approcci content-based o altre strategie per migliorarne l'efficacia e ridurre le limitazioni, in particolare nei casi di *cold start* e dati sparsi.

Formalizzazione del problema. In uno scenario tipico di CF, si considerano un insieme di m utenti $U = \{u_1, u_2, \dots, u_m\}$ e un insieme di n item $I = \{i_1, i_2, \dots, i_n\}$. Ogni utente u_i ha espresso opinioni su un sottoinsieme di item $I_{u_i} \subseteq I$, che può essere anche vuoto.

Le opinioni possono essere:

- **Esplicite:** valutazioni fornite direttamente dall'utente su una scala numerica (es. da 1 a 5);
- **Implicite:** derivate dal comportamento dell'utente (click, acquisti, tempo di permanenza, ecc.).

In questo lavoro consideriamo le prime.

Dato un utente attivo $u_a \in U$, il compito dell'algoritmo di CF è stimare la *piacevolezza* di uno o più item non ancora valutati. Questa può essere espressa in due forme principali:

- **Predizione** ($P_{a,j}$): valore numerico previsto che rappresenta il gradimento dell'item $i_j \notin I_{u_a}$ per l'utente attivo. Il valore è nello stesso intervallo delle valutazioni esplicite.
- **Raccomandazione Top-N** ($I_r \subset I$): lista ordinata di N item che si prevede piacciono di più all'utente attivo. È richiesto che $I_r \cap I_{u_a} = \emptyset$, ossia che gli item raccomandati non siano già stati valutati.

Matrice delle valutazioni. Il processo di collaborative filtering viene comunemente rappresentato tramite una **matrice delle valutazioni** $A \in \mathbb{R}^{m \times n}$, in cui ogni riga rappresenta un utente e ogni colonna un item. L'elemento $a_{i,j}$ rappresenta il punteggio di preferenza assegnato dall'utente i all'item j .

Esistono tre principali approcci al collaborative filtering:

- **User-based CF:** raccomanda gli item che sono piaciuti ad utenti simili all'utente target. Il presupposto è che utenti con preferenze simili in passato tenderanno a concordare anche in futuro. Per implementare questo approccio, si costruisce una *matrice di similarità tra utenti*, in cui ogni cella rappresenta il grado di similarità tra due utenti [5].
- **Item-based CF:** propone item simili a quelli che l'utente ha già valutato positivamente. Questo approccio è spesso preferito in contesti ad alta scalabilità, grazie alla maggiore stabilità dei profili degli item rispetto a quelli degli utenti [6].
- **Hybrid CF:** questo approccio combina i risultati ottenuti dal *user-based* e dall'*item-based CF*, con l'obiettivo di migliorare la qualità delle raccomandazioni e superare le limitazioni di ciascun metodo individuale [7].

2.3 Problema del cold start

Il Collaborative Filtering, pur essendo uno dei paradigmi più diffusi per la personalizzazione delle raccomandazioni, soffre di numerose **criticità**, tra cui la forte *sparsità* della matrice **utente-oggetto** e la conseguente instabilità delle misure di similarità, problemi di *scalabilità* nel calcolo delle distanze su grandi dataset, nonché la vulnerabilità ad attacchi di tipo “shilling” e a utenti *gray sheep* con gusti atipici. Tuttavia, tra tutte, la più insidiosa è il cosiddetto *cold start*, che si manifesta sia nel caso di un utente appena registrato, privo di storico di valutazioni, sia di un nuovo oggetto senza feedback, causando raccomandazioni di qualità scarsa e bassa copertura del sistema. A fronte di ciò, la letteratura ha proposto varie soluzioni: approcci ibridi content-based e memory/model-based per stimare valutazioni iniziali sui nuovi elementi [8, 9], strategie di *learning* attivo che sollecitano pochi feedback

mirati dall'utente [10, 11], e metodi di *matrix factorization* con regolarizzazione per produrre embedding significativi anche con dati scarsi [12, 13]. Queste soluzioni, spesso in ensemble, rappresentano l'avanguardia per mitigare efficacemente il problema del cold start.

2.4 Tecniche di Clustering

Il **clustering** è una tecnica di apprendimento non supervisionato che consente di raggruppare oggetti simili in insiemi (cluster), sulla base delle loro caratteristiche, senza usare etichette predefinite. Nei sistemi di raccomandazione content-based, il clustering può essere applicato sia agli *item* sia agli *utenti*, con l'obiettivo di semplificare la ricerca di contenuti rilevanti, migliorare l'efficienza computazionale e rendere le raccomandazioni più generalizzabili. Le principali tecniche includono:

- **K-means**: algoritmo particolarmente adatto a dati numerici. Parte da k centroidi iniziali e assegna ogni oggetto al centroide più vicino secondo una misura di distanza (tipicamente euclidea). I centroidi vengono aggiornati iterativamente fino alla convergenza. Richiede di specificare il numero k di cluster a priori.
- **Agglomerative Hierarchical Clustering**: costruisce una gerarchia di cluster unendo ricorsivamente i più simili. Produce un dendrogramma da cui si può selezionare il livello di granularità più adatto. Non richiede di fissare a priori il numero di cluster ed è utile per esplorare strutture multilivello nei dati.
- **K-modes**: estensione di K-means per dati categorici, come genere, lingua, piattaforma, categoria, ecc. Al posto della media (non definita per dati discreti), utilizza la *moda* per rappresentare i centroidi dei cluster e impiega una misura di dissimilarità basata sul numero di attributi discordanti. È particolarmente utile in contesti reali dove gli item sono descritti da etichette testuali o valori discreti, come film con genere, regista, attori, o prodotti con marca e categoria.

Nel contesto dei sistemi di raccomandazione, il clustering consente di proporre contenuti che appartengono allo stesso gruppo di quelli apprezzati dall'utente, anche in assenza di valutazioni esplicite. Inoltre, permette di gestire meglio la *sparsità* dei dati, migliorare la *scalabilità* del sistema e supportare il *cold start* per nuovi utenti o item attraverso l'assegnazione a cluster esistenti.

3 Metodologia

La pipeline implementata ha permesso di confrontare due approcci CF in scenari realistici. L'approccio User-Based ha richiesto strategie specifiche per gestire il cold start, come il clustering basato sui metadati. Il modello Hybrid, invece, ha privilegiato la fusione delle due fonti collaborative, risultando più robusto in condizioni ideali. Le valutazioni tramite metriche continue, binarie e Top-K hanno fornito una panoramica completa delle prestazioni dei modelli.

3.1 MovieLens 100k Dataset

Per lo sviluppo del progetto è stato utilizzato il dataset **MovieLens 100k** [14], un insieme di dati ampiamente utilizzato nella ricerca sui sistemi di raccomandazione. Rilasciato dal *GroupLens Research Project* dell'Università del Minnesota, contiene 100.000 valutazioni esplicite fornite da 943 utenti su 1.682 film. È particolarmente adatto per sperimentazioni su algoritmi di filtraggio collaborativo e tecniche ibride.

3.1.1 Struttura dei dati

Il dataset è composto da diversi file testuali, tra cui:

- **u.data**: contiene le valutazioni con la struttura `user_id`, `item_id`, `rating`, `timestamp`. Le valutazioni sono interi da 1 a 5.
- **u.item**: include informazioni sui film (ID, titolo, data di uscita, e codifiche binarie per 19 generi cinematografici).
- **u.user**: descrive gli utenti mediante età, genere, professione e CAP.
- **u.genre**, **u.occupation**: forniscono la lista completa dei generi e delle professioni presenti.

3.1.2 Statistiche principali

- Numero di utenti: 943
- Numero di film: 1.682
- Numero totale di valutazioni: 100.000 (opinioni esplicite)
- Valutazioni per utente: minimo 20

Il dataset è disponibile pubblicamente per scopi di ricerca e didattica presso il sito ufficiale <https://grouplens.org/datasets/movielens/100k/>.

3.2 Pipeline Sperimentale

L'obiettivo del progetto è stato quello di analizzare il comportamento e le prestazioni di un sistema di raccomandazione basato su Collaborative Filtering (CF), con particolare attenzione al problema del cold start per i nuovi utenti. Sono stati confrontati due approcci principali: **User-Based** e **Hybrid**, ciascuno valutato in scenari con e senza cold start.

3.2.1 Preprocessing e Suddivisione del Dataset

In entrambi gli approcci è stato utilizzato il file `u.data`, contenente le valutazioni utente–film. La colonna `timestamp` è stata rimossa in quanto non rilevante per la raccomandazione. Il dataset è stato quindi suddiviso in:

- **training set**: 70% delle interazioni totali;
- **test set**: 30% rimanenti.

Per lo **scenario ideale** (senza cold start), sono state eliminate dal test set tutte le occorrenze di utenti o film non presenti nel training set, per evitare situazioni di cold start.

3.2.2 Costruzione della Matrice dei Rating e di Similarità

A partire dal **train set**, è stata costruita una matrice dei rating conosciuti, dove sulle righe vengono rappresentati gli utenti e sulle colonne i film, mentre le celle di questa matrice sono le valutazioni conosciute. Inoltre è stata costruita una matrice delle similarità, fondamentale per poi effettuare la predizione dei rating.

Sono stati applicati due diversi metodi:

- **Similarità degli utenti:** la similarità tra utenti viene calcolata tramite la **correlazione di Pearson**, dopo aver centrato la matrice dei rating rispetto alla media di ciascun utente:

$$R_{u,i}^{\text{centered}} = R_{u,i} - \bar{r}_u \quad (1)$$

Centra la matrice di rating rispetto alla media.

$$\text{sim}_{\text{user}}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (2)$$

dove I_{uv} è l'insieme di item valutati sia da u che da v .

- **Similarità degli item** (solo nel modello Hybrid): è stata effettuata una trasposizione della matrice e la similarità tra item è calcolata usando la **cosine similarity**, applicata alla matrice trasposta R^T , in cui ogni item è rappresentato come vettore nel dominio degli utenti. I valori mancanti vengono riempiti con 0:

$$\text{sim}_{\text{item}}(i, j) = \frac{\sum_{u \in U_{ij}} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U_{ij}} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U_{ij}} r_{u,j}^2}} \quad (3)$$

dove U_{ij} è l'insieme di utenti che hanno valutato entrambi gli item i e j .

3.2.3 Calcolo delle Predizioni

Le predizioni dei rating sono state calcolate utilizzando la similarità tra utenti (User-Based) o tra item (Hybrid). In entrambi i casi, la stima del rating si ottiene come media pesata delle valutazioni esistenti. La predizione del rating dell'utente u per l'item i è data da:

$$\hat{r}_{u,i}^{\text{user}} = \bar{r}_u + \frac{\sum_{v \in N_u(i)} \text{sim}_{\text{user}}(u, v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u(i)} |\text{sim}_{\text{user}}(u, v)|} \quad (4)$$

dove $N_u(i)$ è il gruppo di utenti simili a u che hanno valutato l'item i .

Mentre, nel modello ibrido, la predizione del rating che l'utente u attribuirebbe all'item i è calcolata come:

$$\hat{r}_{u,i}^{\text{item}} = \frac{\sum_{j \in N_i(u)} \text{sim}_{\text{item}}(i, j) \cdot r_{u,j}}{\sum_{j \in N_i(u)} |\text{sim}_{\text{item}}(i, j)|}$$

dove $N_i(u)$ è l'insieme di item simili a i già valutati dall'utente u .

Nel modello **Hybrid**, le predizioni finali sono ottenute mediante una combinazione lineare dei due punteggi stimati. Le due predizioni (user-based e item-based) sono combinate con un parametro di fusione α , secondo la formula:

$$\hat{r}_{u,i}^{\text{hybrid}} = \alpha \cdot \hat{r}_{u,i}^{\text{user}} + (1 - \alpha) \cdot \hat{r}_{u,i}^{\text{item}} \quad (5)$$

dove $\alpha \in [0, 1]$ è un parametro di bilanciamento che può essere ottimizzato empiricamente. Questo tipo di sistema ibrido mira a ottenere il meglio da entrambi gli approcci: la flessibilità e la personalizzazione del modello user-based e la robustezza e scalabilità del modello item-based.

3.3 Approcci a Confronto

3.3.1 User-Based

Scenario A: senza cold start. Dopo aver garantito che tutti gli utenti e i film nel test set fossero già presenti nel training, è stata calcolata la matrice delle similarità utente tramite correlazione di Pearson. Le predizioni si basano su una media pesata dei rating espressi dagli utenti più simili.

Scenario B: con cold start (nuovo utente). Per simulare casi di cold start, un gruppo di utenti è stato rimosso dal training. Le loro interazioni sono state spostate nel test set. Per predire i rating dei nuovi utenti, è stata adottata una strategia basata su clustering utilizzando i metadati presenti nel file `u.user`. Gli utenti sono stati raggruppati per somiglianza, e le predizioni sono state calcolate come media dei rating degli altri membri del cluster:

$$\hat{r}_{u,i} = \frac{1}{|C_c|} \sum_{v \in C_c} r_{v,i} \quad (6)$$

Dove:

- $\hat{r}_{u,i}$ è la predizione del rating dell'utente u per l'item i ,
- C_c è l'insieme degli utenti nel cluster c ,
- $|C_c|$ è il numero di utenti nel cluster c ,
- $r_{v,i}$ è il rating dato dall'utente v all'item i .

3.3.2 Hybrid

Scenario ideale (senza cold start). Il modello Hybrid ha unito le predizioni User-Based e Item-Based attraverso un parametro α . Sono stati esplorati diversi valori per α per identificare il miglior compromesso tra le due fonti di informazione. Le predizioni finali sono state quindi arrotondate per riflettere la natura discreta dei rating.

Nel caso del modello Hybrid, non è stata implementata una specifica strategia per affrontare direttamente il problema del cold start, poiché il focus principale è stato sull'integrazione tra le due componenti collaborative.

3.4 Metriche di valutazione

Per valutare l'efficacia del sistema di raccomandazione sono state adottate diverse metriche, ciascuna delle quali fornisce una prospettiva specifica sulle performance del modello. Le metriche utilizzate sono le seguenti:

- **RMSE (Root Mean Square Error)**: misura la differenza tra i valori predetti dal sistema di raccomandazione e i valori reali. È particolarmente sensibile agli errori più grandi, penalizzando maggiormente le predizioni molto distanti dal valore atteso. Viene calcolato come:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{r}_i - r_i)^2}$$

dove \hat{r}_i è il rating predetto, r_i il rating reale e n il numero totale di predizioni.

- **MAE (Mean Absolute Error)**: rappresenta la media degli errori assoluti tra le predizioni e i valori reali. A differenza dell'RMSE, tratta in modo uniforme tutti gli errori, senza penalizzare maggiormente quelli più grandi.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{r}_i - r_i|$$

- **Accuracy**: è una metrica generica che misura la proporzione di predizioni corrette sul totale delle predizioni effettuate. Tuttavia, in ambito di raccomandazione ha senso solo in contesti binari (es. like/dislike).
- **Precision**: misura la percentuale di elementi raccomandati che sono effettivamente rilevanti per l'utente. È utile per valutare la qualità delle raccomandazioni.

$$\text{Precision} = \frac{\text{raccomandazioni rilevanti}}{\text{raccomandazioni totali}}$$

- **Recall**: indica la capacità del sistema di raccomandazione di recuperare tutti gli elementi rilevanti. È una misura di completezza.

$$\text{Recall} = \frac{\text{raccomandazioni rilevanti}}{\text{elementi rilevanti totali}}$$

Per valutare le raccomandazioni nei primi K elementi suggeriti, sono state adottate anche metriche specifiche per ranking:

- **Precision@K**: misura la precisione considerando solo i primi K elementi raccomandati. Indica quanta parte dei top- K suggerimenti è effettivamente rilevante per l'utente.

$$\text{Precision@K} = \frac{\text{elementi rilevanti nei top-K}}{K}$$

- **Recall@K**: quantifica la frazione di elementi rilevanti che appaiono nei primi K suggerimenti, rispetto a tutti gli elementi rilevanti disponibili per l'utente.

$$\text{Recall@K} = \frac{\text{elementi rilevanti nei top-K}}{\text{elementi rilevanti totali}}$$

- **Average Precision@K (AP@K)**: valuta sia la rilevanza che la posizione degli elementi nei top- K . Viene calcolata come la media delle precisioni calcolate in corrispondenza delle posizioni in cui si trovano elementi rilevanti nella top- K .
- **Mean Average Precision@K (MAP@K)**: è la media dell'AP@K calcolata su tutti gli utenti. Fornisce una misura complessiva della qualità delle raccomandazioni a livello globale, tenendo conto sia della rilevanza che dell'ordinamento.

$$\text{MAP@K} = \frac{1}{N} \sum_{u=1}^N \text{AP@K}_u$$

dove N è il numero totale di utenti e AP@K_u è l'Average Precision@K per l'utente u .

Queste metriche permettono di valutare il sistema di raccomandazione sia in termini di accuratezza predittiva (RMSE, MAE) sia in termini di qualità del ranking e rilevanza delle raccomandazioni (Precision, Recall, MAP@K).

Per la valutazione delle metriche di ranking in termini di accuracy, precision, recall, così come per Precision@K, Recall@K, AP@K e MAP@K, è stato necessario definire un criterio per determinare se un film possa essere considerato rilevante per l'utente. A tal fine, è stata introdotta **una soglia di qualità (threshold)**: un film è stato considerato *rilevante* (o “buono”) se il rating assegnato dall'utente è maggiore o uguale a una soglia predefinita. Questa soglia permette di trasformare il problema in un contesto binario, necessario per l'applicazione di metriche di classificazione e ranking.

Valutazione dei cluster

Nel caso in cui il sistema di raccomandazione preveda una fase di clustering degli utenti o degli item, è fondamentale valutare la qualità dei cluster generati. A tal fine, sono stati adottati i seguenti criteri:

- **Silhouette Score**: è una metrica che valuta quanto un punto è ben assegnato al proprio cluster rispetto agli altri cluster. Assume valori tra -1 e 1: valori vicini a 1 indicano una buona separazione tra i cluster, mentre valori vicini a 0 o negativi suggeriscono sovrapposizione o assegnazioni errate. È definito come:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (7)$$

dove $a(i)$ è la distanza media tra il punto i e tutti gli altri punti dello stesso cluster, e $b(i)$ è la distanza minima tra i e i punti di un altro cluster. Il valore finale è ottenuto come media dei valori $s(i)$ di tutti i punti.

- **Valutazione visiva**: oltre alle metriche quantitative, è stata effettuata un'analisi qualitativa dei cluster tramite rappresentazione grafica in 2D e 3D (mediante tecnica di riduzione dimensionale con PCA). Questi grafici permettono di verificare visivamente la coesione interna dei cluster e la loro separazione reciproca, facilitando l'interpretazione e la validazione del processo di clustering.

4 Risultati

In questo capitolo vengono presentati e analizzati i risultati ottenuti dai modelli User-Based e Hybrid nei diversi scenari considerati. Le valutazioni sono state effettuate utilizzando le metriche descritte nella sezione precedente, sia per la predizione dei rating sia per la qualità delle raccomandazioni nei Top-K suggerimenti.

4.1 Statistiche descrittive della matrice

Dalla matrice di train R_{train} (matrice user-item) vengono estratti:

- Numero di utenti: $n_{\text{users}} = 943$;
- Numero di item: $n_{\text{items}} = 1682$;
- Numero di valutazioni presenti (non nulli): $\text{nnz} = 100000$;
- Sparsità: 94% (vedi Figura 1).

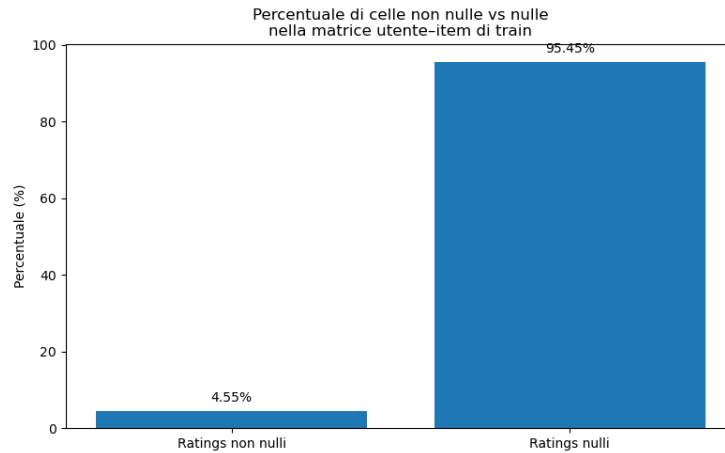


Figure 1: Distribuzione delle celle non nulle vs nulle nella matrice utente-item di train.

4.2 Analisi delle similarità

1. User-based

La distribuzione dei coefficienti di *Pearson* tra coppie di utenti mostra una concentrazione attorno a valori moderatamente positivi, suggerendo che molti utenti condividono solo parzialmente le stesse preferenze.

Coefficienti più alti (vicini a 1) indicano utenti con comportamenti molto simili, mentre valori negativi evidenziano gusti divergenti. Una distribuzione centrata attorno allo zero è comune in contesti con alta eterogeneità di gusti.

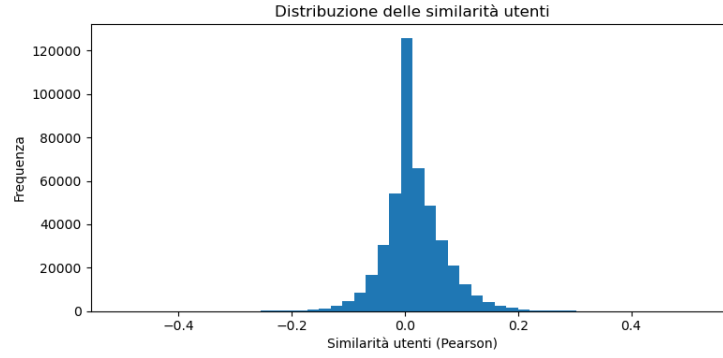


Figure 2: Distribuzione dei coefficienti di similarità utente-utente (Pearson).

2. **Item-based** La similarità tra item è calcolata usando la *coseno-similarity*, che valuta quanto due film siano stati apprezzati dagli stessi utenti. Valori prossimi a 1 indicano film spesso votati insieme (tipicamente per genere o popolarità), mentre valori bassi suggeriscono film con audience distinte. La maggior parte delle coppie presenta similarità bassa, coerente con la natura altamente sparsa del dataset.

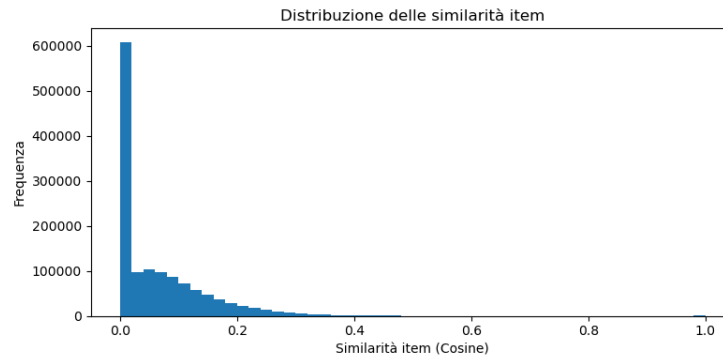


Figure 3: Distribuzione delle similarità item-item (coseno).

4.3 Valutazione globale delle predizioni nello scenario ideale (senza cold start)

Nel contesto privo di problemi di cold start, entrambi i modelli hanno potuto sfruttare pienamente i dati presenti nel training set.

Per ogni approccio (User-Based, Hybrid) sono state calcolate:

- *RMSE* e *MAE* sul test set pulito;
- *Accuracy*, *Precision* e *Recall* binarie con threshold 3 e 4, per .

Le tabelle seguenti riportano le performance dei modelli User-Based e Hybrid rispetto alle metriche predittive e di ranking.

4.3.1 Approccio User-Based

Table 1: Valutazione dei rating con MAE ed RMSE predetti nello scenario ideale (senza cold start) utilizzando l'approccio cf user-based

RMSE	MAE
1.06	0.78

Table 2: Valutazione dei rating con Accuracy, Precision e Recall predetti nello scenario ideale (senza cold start) utilizzando l'approccio cf user-based con threshold pari a 3 e 4

Threshold	Accuracy	Precision	Recall
3	0.83	0.83	0.99
4	0.64	0.67	0.70

Il modello user-based mostra buone prestazioni predittive: $RMSE = 1.06$ e $MAE = 0.78$ indicano una discreta accuratezza nella stima dei rating.

Con threshold 3, il sistema ottiene ottimi risultati di classificazione (accuracy e precision = 0.83, recall = 0.99), evidenziando un'elevata capacità di individuare contenuti apprezzati dagli utenti. Con threshold 4, le metriche calano (accuracy = 0.64, precision = 0.67, recall = 0.70), ma restano accettabili, segnalando un comportamento più selettivo.

Nel complesso, l'approccio user-based si dimostra efficace, soprattutto in scenari con dati completi e ben distribuiti.

4.3.2 Approccio Hybrid

Table 3: Valutazione MAE e MRSE nello scenario ideale (senza cold start) utilizzando l'approccio cf Hybrid

Metrica	Alpha	MAE	MSRE
MAE minima	1	1.065	0.7855
MRSE minima	1	1.065	0.7855

Sia per la MAE minima che per la RMSE minima, il modello ha ottenuto valori positivi, pari a 1.065 e 0.7855 rispettivamente.

Questi valori, combinati con le buone prestazioni di precisione e recall, suggeriscono che il modello è in grado di fornire previsioni numeriche accurate sui rating.

Table 4: Metriche di valutazione nello scenario ideale (senza cold start) utilizzando l’approccio cf Hybrid

Metrica	Alpha	Threshold	Accuracy	Precision	Recall
Precision massima	0.9	3	0.83	0.83	0.98
Recall massima	0.9	3	0.83	0.83	0.99
Accuracy massima	0.9	3	0.83	0.83	0.99

Dalla tabella, si osserva che i valori di Accuracy, Precision e Recall sono massimizzati contemporaneamente per lo stesso set di parametri ($\alpha = 0.9$, $\text{threshold} = 3$). Questo suggerisce un buon bilanciamento tra la capacità del modello di identificare correttamente gli item rilevanti e la sua robustezza nel ridurre falsi positivi e negativi.

I valori raggiunti (Accuracy = 0.83, Precision = 0.83, Recall = 0.98–0.99) indicano un comportamento molto positivo del modello nell’identificare correttamente i rating.

4.4 Valutazione Top-K

Sono stati valutati anche le metriche *Precision@K*, *Recall@K* e *MAP@K*. Nello specifico sono state calcolate per $K \in \{5, 10, 15, 20\}$.

4.4.1 Approccio User-Based

Table 5: Valutazione delle top-k raccomandazioni per l’utente 1 su diversi valori di k e *threshold* utilizzando l’approccio user-based

k	threshold	precision@k	recall@k
5	3	0.40	0.03
10	3	0.60	0.09
15	3	0.66	0.15
20	3	0.70	0.21
5	4	0.20	0.02
10	4	0.30	0.07
15	4	0.40	0.14
20	4	0.45	0.21

In generale, si osserva (tabella 5) che aumentando il valore di k , ovvero il numero di raccomandazioni restituite all’utente 1, entrambe le metriche tendono a migliorare. Questo è coerente con l’aspettativa: allargando il numero di item consigliati, cresce la probabilità di includere quelli rilevanti. Tuttavia, ciò può talvolta comportare una perdita di precisione, anche se in questo caso la precisione resta relativamente elevata anche per valori alti di k . Ad esempio, con $k = 20$ e soglia di rilevanza impostata a 3, si ottiene il miglior bilanciamento tra precisione (0.70) e richiamo (0.21), suggerendo che il sistema riesce a proporre molte raccomandazioni corrette e a coprire una discreta parte degli item rilevanti.

Un altro elemento importante riguarda il threshold, ovvero la soglia a partire dalla quale un item viene considerato rilevante. Con una soglia più bassa (ad esempio, 3), il sistema

ottiene risultati migliori sia in termini di precision che di recall. Al contrario, quando la soglia viene alzata a 4, il sistema fatica di più a individuare e raccomandare item con valutazioni elevate, facendo calare le performance.

Table 6: Valori di MAP@k per combinazioni di k e *threshold* utilizzando l'approccio user-based

k	threshold	MAP@k
5	3	0.30
5	4	0.25
10	3	0.49
10	4	0.39
15	3	0.59
15	4	0.46
20	3	0.65
20	4	0.50

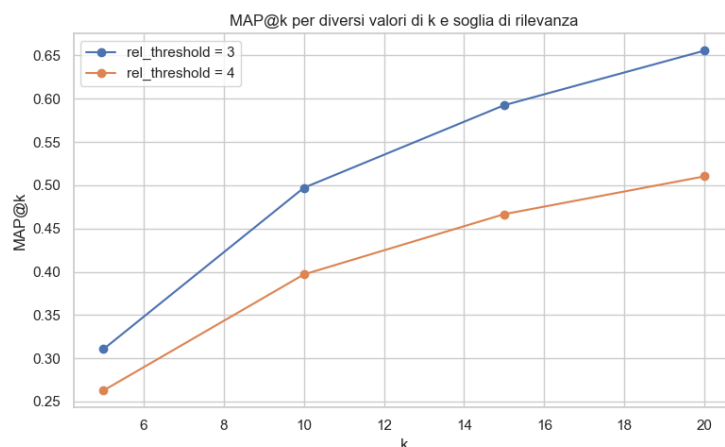


Figure 4: Andamento metrica MAP@K al variare di k e threshold per l'approccio cf user-based

Dalla tabella 6 e dal grafico 4 emerge chiaramente che, al crescere del valore di k (cioè del numero di raccomandazioni generate), la MAP@k tende ad aumentare per entrambe le soglie di rilevanza considerate (3 e 4). Questo comportamento è coerente con il fatto che un numero maggiore di raccomandazioni offre più opportunità di includere item rilevanti in posizioni alte della lista, migliorando la precisione media cumulativa.

Tuttavia, la metrica risente fortemente della soglia di rilevanza adottata. Quando il threshold è impostato a 3, si ottengono valori di MAP@k sistematicamente superiori rispetto a quando la soglia è più alta (4). Questo indica che, se si considerano come rilevanti anche gli item con valutazioni medio-alte, il sistema è in grado di offrire raccomandazioni molto più efficaci, raggiungendo un valore massimo di 0.65 per $k = 20$. Al contrario, con soglie più restrittive, il sistema ha più difficoltà a trovare item rilevanti secondo il criterio adottato, con una MAP@k che si ferma a 0.50 per lo stesso valore di k .

Il grafico rafforza questa osservazione, mostrando due curve separate con un divario

costante tra le soglie, a testimonianza della sensibilità del sistema rispetto al criterio di rilevanza adottato. È anche evidente che la crescita di $MAP@k$ è più marcata nei primi valori di k (da 5 a 10), segno che l'inclusione dei primi pochi item nella lista di raccomandazione ha un forte impatto sulla precisione media cumulativa.

4.4.2 Approccio Hybrid

Table 7: Combinazioni ottimali dei parametri per massimizzare le diverse metriche di valutazione con approccio hybrid

Metrica	Alpha	Threshold	k	Precision@K	Recall@K	MAP@K
Precision@K massima	1.0	3.0	5	0.8407	0.3435	0.3099
Recall@K massima	0.9	3.0	20	0.6424	0.7449	0.6543
MAP@K massima	1.0	3.0	20	0.6423	0.7449	0.6546

Al seguito viene mostrato un grafico 3D sull'andamento della metrica $MAP@K$ al variare di Threshold, di alpha e di K.

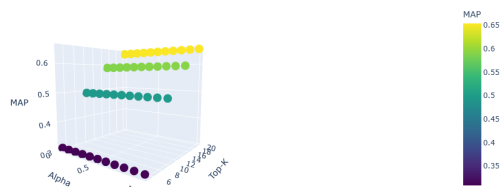


Figure 5: Andamento metrica $MAP@K$ al variare di alpha, k e threshold

Considerazioni finali Per tutte le metriche di valutazione considerate (*Accuracy*, *Precision*, *Recall*, *Precision@K*, *Recall@K*, *MAP@K*), i valori ottenuti risultano essere pari o molto prossimi a 1. Questi risultati suggeriscono che **l'approccio user-based si comporta complessivamente meglio rispetto a quello item-based**, soprattutto nello scenario ideale in cui non si presentano problematiche di *cold start*.

4.5 Scenario con Cold Start (nuovi utenti)

Per la gestione del problema del cold start per i nuovi utenti, è stato utilizzato il file `u.user` del dataset, poiché contiene metainformazioni descrittive sugli utenti. In particolare, è stato affrontato il problema della presenza di variabili categoriche (come *gender*, *occupation* e *zip_code*) che non possono essere direttamente utilizzate da algoritmi di clustering in quanto non numeriche. Inoltre, per stimare le preferenze dei nuovi utenti in assenza di dati storici, sono stati testati sia un approccio globale al calcolo delle similarità, basato sull'intera popolazione utente, sia un approccio locale, che considera solamente un sottoinsieme di utenti simili in base alle metainformazioni. I risultati ottenuti mostrano che l'approccio locale offre performance leggermente migliori rispetto a quello globale per quanto riguarda

le metriche predittive generali, suggerendo che una personalizzazione più mirata (basata su gruppi omogenei di utenti) può portare a raccomandazioni più efficaci in fase di cold start. Tuttavia, per le metriche di ranking Top@K è stato osservato un lieve peggioramento, probabilmente dovuto alla minore varietà di item suggeriti. Nel seguito, ci concentreremo sulla versione globale, attribuendo maggiore importanza a questo tipo di valutazione.

4.5.1 Metodologia

Gli algoritmi di clustering utilizzati sono principalmente *KMeans* e *Agglomerative Clustering*. In entrambi i casi, i risultati migliori sono stati ottenuti utilizzando l’encoding *Word2Vec*, che ha permesso di mappare ciascuna categoria in uno spazio vettoriale semantico, catturando relazioni latenti tra le etichette e rendendo il processo di clustering più efficace (Tabelle 9 e 8). I punteggi migliori di *silhouette score* sono stati pari a 0,62 con 2 cluster per *KMeans* e 0,54 con 3 cluster per *Agglomerative Clustering*. Nonostante il punteggio leggermente inferiore, è stato scelto l’algoritmo *Agglomerative Clustering* in quanto, con 3 cluster, consente una segmentazione più significativa degli utenti simili, sfruttando appieno la rappresentazione fornita da *Word2Vec*.

Table 8: Tecniche di encoding testate con Agglomerative Clustering

Encoding	Silhouette Score	K cluster
OneHot Encoding	0,22	2
Label Encoder	0,3	2
Binary Encoder	0,13	3
HASHER	0,2	10
Target encoder	0,49	6
Word2vec	0,54	3

Table 9: Tecniche di encoding testate con clustering KMeans

Encoding	Silhouette Score	K cluster
Label Encoder	0,32	7
Binary Encoder	0,13	2
HASHER	0,23	10
Target encoder	0,49	6
Word2vec	0,62	2

I vettori generati tramite *Word2Vec* sono stati utilizzati come input per l’*Agglomerative Clustering*, impiegando la distanza euclidea come metrica di similarità e la strategia di linkage *ward*, che minimizza la varianza intra-cluster durante la fusione dei gruppi.

4.5.2 Selezione numero di cluster

Il numero ottimale di cluster è stato determinato empiricamente sulla base del *silhouette score*. Infine, le raccomandazioni sono state generate considerando esclusivamente gli utenti appartenenti allo stesso cluster dell’utente target.

4.5.3 Metriche di valutazioni finali

Un primo confronto è stato effettuato tra lo scenario ideale (dove tutte le informazioni sugli utenti sono note) e lo scenario con cold start, in cui si è fatto uso del clustering per raggruppare utenti simili con i metadati a disposizione. I valori delle metriche principali – RMSE, MAE, Accuracy, Precision e Recall – risultano pressoché invariati tra i due contesti. In particolare, il sistema con clustering ha fatto registrare un RMSE di 1.04 e un MAE di 0.77, valori leggermente migliori rispetto allo scenario ideale (1.06 e 0.78). Anche Accuracy e Precision restano stabili a 0.83, mentre il Recall cala solo marginalmente da 0.99 a 0.98.

Questi risultati suggeriscono che il clustering degli utenti, effettuato sulla base delle rappresentazioni semantiche ottenute tramite Word2Vec, consente di compensare efficacemente la mancanza di dati, mantenendo elevata la qualità delle raccomandazioni.

Table 10: Confronto tra le valutazioni dei rating predetti nello scenario ideale (senza cold start) e nello scenario con cold start, utilizzando l’approccio cf user-based con threshold = 3 (il threshold ci indica se l’item è considerato buono o meno)

Modello	RMSE	MAE	Accuracy	Precision	Recall
User-Based (Clustering)	1.04	0.77	0.83	0.83	0.98
User-Based (Caso ideale)	1.06	0.78	0.83	0.83	0.99

Un’ulteriore valutazione è stata condotta utilizzando la metrica MAP@K. I test sono stati effettuati variando sia il numero di raccomandazioni restituite (K) sia il threshold che definisce un item come “buono” (ossia, meritevole di raccomandazione).

L’utilizzo del clustering non introduce rumore o distorsioni nel processo di raccomandazione, ma anzi fornisce un contesto coerente su cui operare, permettendo una selezione più mirata degli utenti simili. Interessante notare come infatti le performance siano identiche sia con il clustering dei dati (CD) sia senza.

Table 11: MAP@K ottenuto considerando lo scenario di cold start, variando diversi K e threshold

K	Threshold	MAP@K (con CD)	MAP@K (senza CD)
5	3	0.30	0.30
5	4	0.25	0.25
10	3	0.49	0.49
10	4	0.39	0.39
15	3	0.59	0.59
15	4	0.46	0.46
20	3	0.65	0.65
20	4	0.51	0.50

4.5.4 Visualizzazione 3D

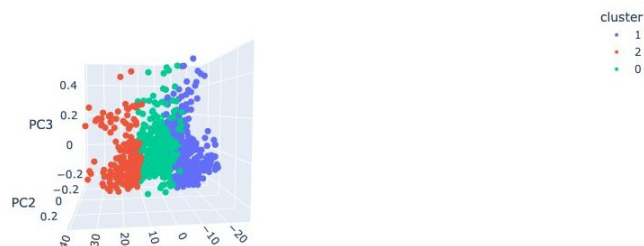


Figure 6: Clustering ottenuto utilizzando Agglomerative con encoding Word2Vec

4.6 Scenario con Cold Start (nuovi item)

Il dataset utilizzato per il clustering degli item è stato il MovieLens 100k, in particolare il file `u.item` contenente informazioni sui film e i relativi generi. Sono stati esclusi i campi `IMDb_url`, `video_release_date` e `release_date` in quanto non pertinenti all'analisi basata sui generi.

Le variabili considerate per il clustering sono 19 generi binari (0/1) indicanti la presenza o meno di un dato genere per ciascun film.

4.6.1 Metodologia

Sono stati testati diversi metodi di clustering, tra cui *K-Modes* e clustering gerarchico agglomerativo, utilizzando come features esclusivamente i generi dei film. Successivamente, integrando anche la variabile `release_date`, è stato applicato il metodo *K-Prototypes*, più adatto a dati misti numerici e categorici. Nel seguito viene illustrato il caso migliore, ottenuto con il metodo *K-Modes* sulle sole feature di genere.

4.6.2 Metriche di similarità

Data la natura binaria dei dati, sono state valutate diverse metriche di distanza/similarità, quali: Hamming, Jaccard, Dice, Sokal-Michener e Russell-Rao.

Per ciascuna metrica è stata calcolata la matrice di distanza tra gli item.

4.6.3 Selezione del numero di cluster

Il numero ottimale di cluster k è stato individuato tramite il coefficiente di Silhouette, calcolato sulle matrici di distanza per valori di k compresi tra 2 e 10. Il valore di k che ha massimizzato il coefficiente di Silhouette è stato scelto come parametro ottimale.

Di seguito sono riportati i grafici dello score di Silhouette in funzione di k per alcune delle metriche analizzate.

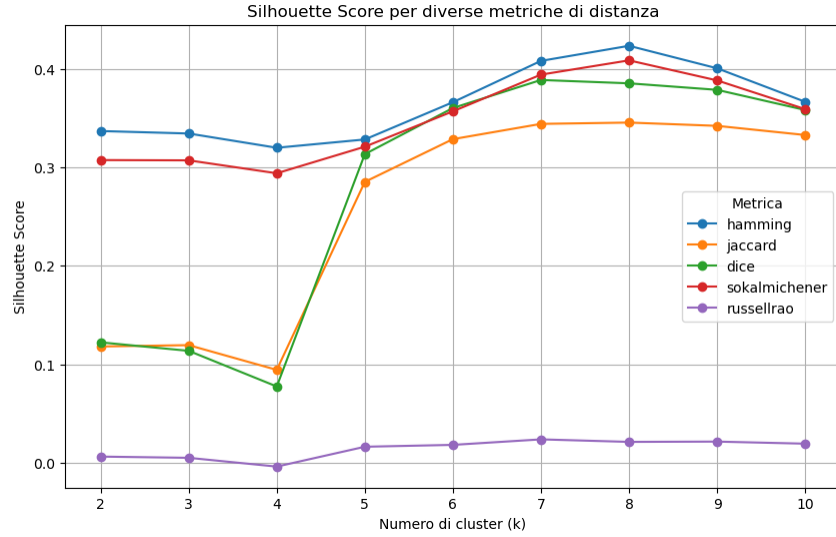


Figure 7: Coefficiente di Silhouette in funzione del numero di cluster e con le metriche.

Il picco massimo si ha con un valore della silhouette pari a 0.4232 con $k = 8$ cluster.

4.6.4 Visualizzazione 3D

La riduzione dimensionale è stata effettuata con UMAP a tre componenti sui dati binari, consentendo la visualizzazione spaziale dei cluster individuati dal metodo K-Modes.

La rappresentazione tridimensionale interattiva, realizzata con Plotly, mostra i film colorati secondo il cluster di appartenenza.

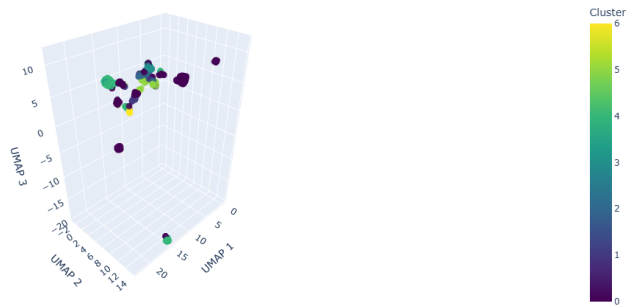


Figure 8: Visualizzazione 3D dei cluster K-Modes su embedding UMAP.

4.6.5 Conclusioni

Dall'analisi del grafico dei silhouette score per le diverse metriche di distanza emerge che il valore massimo ottenuto è pari a 0.4232. Questo valore, inferiore a 0.5, indica che la separazione tra i cluster ottenuti non è particolarmente netta. Di conseguenza, si può affermare che il clustering tramite l'algoritmo K-Modes sulle categorie di genere dei film non consente una distinzione ottimale dei gruppi. Ciò suggerisce che le caratteristiche considerate potrebbero non essere sufficientemente discriminanti o che la struttura dei dati non si presta a una partizione chiara in cluster ben definiti.

5 Conclusioni

Il presente lavoro ha approfondito lo sviluppo e la valutazione di un sistema di raccomandazione basato su Collaborative Filtering, esaminando due scenari ipotetici, con e senza il problema del cold start. Dall'analisi comparativa tra approcci User-Based e Hybrid nello scenario ideale che, nel nostro caso, l'approccio User-Based è predominante. Dopo questa comparazione, è emerso che l'integrazione dei metadati utente attraverso tecniche di clustering rappresenta una soluzione efficace per la gestione dei nuovi utenti privi di storico (Cold start).

L'adozione di tecniche avanzate di encoding, come Word2Vec, ha permesso di rappresentare semanticamente variabili categoriche, migliorando la qualità del clustering. In particolare, l'algoritmo di Agglomerative Clustering ha garantito una segmentazione significativa degli utenti, risultando più efficace rispetto all'approccio KMeans, nonostante un silhouette score leggermente inferiore.

In fase di valutazione, l'approccio locale al calcolo delle similarità, limitato agli utenti appartenenti allo stesso cluster, ha prodotto risultati leggermente migliori rispetto a quello globale per le metriche predittive generali, come RMSE e MAE, grazie a una maggiore omogeneità all'interno dei gruppi. Tuttavia, le metriche di ranking Top-K (come Precision@K e MAP@K) hanno mostrato una performance superiore nell'approccio globale, probabilmente a causa di una maggiore varietà di item considerati e di un miglior posizionamento degli elementi rilevanti nella parte alta delle raccomandazioni.

Infine, è stato analizzato anche il problema del cold start per i nuovi item, applicando tecniche di clustering. Tuttavia, i risultati ottenuti sono stati meno soddisfacenti, con silhouette score relativamente bassi, probabilmente a causa della scarsa qualità e varietà delle informazioni disponibili nei dati sui film. Ciò suggerisce che, per migliorare il clustering degli item, sarebbero necessarie feature più ricche o diversificate.

Nel complesso, i risultati ottenuti dimostrano che il clustering basato su rappresentazioni semantiche dei metadati è una strategia valida per mitigare le limitazioni intrinseche del Collaborative Filtering puro, offrendo un sistema robusto e adattabile a scenari reali.

References

- [1] R. Burke, A. Felfernig, and M. H. Göker, “Recommender systems: An overview,” *Ai Magazine*, vol. 32, no. 3, pp. 13–18, 2011.
- [2] Z.-D. Zhao and M.-S. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” in *2010 third international conference on knowledge discovery and data mining*. IEEE, 2010, pp. 478–481.
- [3] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The adaptive web: methods and strategies of web personalization*. Springer, 2007, pp. 325–341.
- [4] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in artificial intelligence*, vol. 2009, no. 1, p. 421425, 2009.
- [5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 230–237.
- [6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.
- [7] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [8] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, 2002, pp. 253–260.
- [9] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, “Addressing cold-start problem in recommendation systems,” in *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC ’08)*. ACM Press, 2008, pp. 208–211.
- [10] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, “Getting to know you: Learning new user preferences in recommender systems,” in *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI ’02)*. ACM, 2002, pp. 127–134.
- [11] N. Houlsby, J. M. Hernández-Lobato, and Z. Ghahramani, “Cold-start active learning with robust ordinal matrix factorization,” in *Proceedings of the 31st International Conference on Machine Learning (ICML ’14)*, vol. 32. PMLR, 2014, pp. 766–774.
- [12] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, “Facing the cold start problem in recommender systems,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

- [13] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, “A meta-learning perspective on cold-start recommendations for items,” in *Advances in Neural Information Processing Systems 31 (NeurIPS '17)*, 2017.
- [14] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” <https://grouplens.org/datasets/movielens/>, 2015, aCM Transactions on Interactive Intelligent Systems (TiiS), 5(4), Article 19.