MOvie recommender system

Zagaria Riccardo

Matricola: 760475

A.A. 2024 - 2025

E-mail: [r.zagaria18@studenti.uniba.it](mailto:r.zagaria18@studenti.uniba.it)

Link GitHub: https://github.com/venicebeachl/IconProject.git

Sommario

[1) Introduzione 2](#_Toc187227300)

[1.1) Requisiti funzionali 2](#_Toc187227301)

[2) Creazione del Dataset 3](#_Toc187227302)

[2.2) Descrizione delle Feature 3](#_Toc187227303)

[3) Ricerca 4](#_Toc187227304)

[3.1) Algoritmo Random Walk 4](#_Toc187227305)

[3.2) Risultati di Precisione e Recall 4](#_Toc187227306)

[3.3) Possibili Miglioramenti 5](#_Toc187227307)

[4) Apprendimento Supervisionato 6](#_Toc187227308)

[4.1) Implementazione del K-Fold Cross Validation 6](#_Toc187227309)

[4.2) Risultati della K-Fold Cross Validation 6](#_Toc187227310)

[4.3) Possibili Miglioramenti 7](#_Toc187227311)

[5) Ragionamento Probabilistico 8](#_Toc187227312)

[5.1) Processo di Implementazione 8](#_Toc187227313)

[5.2) Valutazione delle Performance e Analisi dei Risultati 9](#_Toc187227314)

[5.3) Conclusioni e Sviluppi Futuri 10](#_Toc187227315)

[6) Ragionamento Logico 11](#_Toc187227316)

[6.1) Processo di Implementazione 11](#_Toc187227317)

[6.2) Valutazione delle Performance e Analisi dei Risultati 12](#_Toc187227318)

[6.3) Conclusioni e Sviluppi Futuri 12](#_Toc187227319)

[7) Confronto delle Soluzioni 14](#_Toc187227320)

[7.1) Soluzione Selezionata 15](#_Toc187227321)

[7.2) Sviluppi Futuri 15](#_Toc187227322)

[Riferimenti Bibliografici 16](#_Toc187227323)

# Introduzione

L’obiettivo di questo progetto è quello di sviluppare un Recommender System in grado di consigliare dei film all’utente. L’origine delle raccomandazioni può essere Content-Based, ovvero basato sulla similarità tra due o più prodotti, film in questo caso, o User-Based, ovvero si basa sulla similarità tra due o più utenti.

## Requisiti funzionali

Il progetto è stato realizzato in Python, ad oggi uno dei linguaggi di programmazione più utilizzati, poiché questo linguaggio mette a disposizione numerose librerie che permettono di esaminare facilmente i dati, o di realizzare facilmente dei grafici, utili per analizzare statisticamente i risultati ottenuti da determinate soluzioni.

La versione di Python utilizzata è la versione 3.12.8. L’IDE utilizzato per lo sviluppo del progetto è, invece, Visual Studio Code. Si può, però, utilizzare un IDE qualsiasi.

Tra le librerie di Python utilizzate, troviamo invece:

* **matplotlib**, per la realizzazione dei grafici;
* **numpy**, per la gestione degli array;
* **pandas**, per l’importazione dei dataset nel formato *.csv*;
* **seaborn**, per la realizzazione di altri grafici;
* **sklearn**, per l’apprendimento automatico;
* **os**, per l’interazione con il sistema operativo;
* **cornac**, per lo sviluppo di recommender system.
* **Imblearn**, per l’oversampling tramite SMOTE;

# Creazione del Dataset

Il dataset utilizzato è il MovieLens 100K.

MovieLens 100K è stato preprocessato per unificare informazioni su utenti, film e valutazioni. Questo processo ha incluso i seguenti passaggi chiave:

* **Caricamento dati**: I dataset relativi agli utenti, ai film e alle valutazioni sono stati importati da file separati.
* **Unione dei dati**: Attraverso operazioni di merge, sono state integrate le informazioni demografiche degli utenti, i dettagli sui film (titoli e generi) e le valutazioni esplicite fornite dagli utenti.
* **Gestione dei dati mancanti**: Sono stati identificati e trattati i valori mancanti per garantire l'integrità e la qualità del dataset finale.
* **Trasformazione e pulizia**: I dati sono stati convertiti in un formato adatto per le analisi successive, eliminando le colonne non necessarie e standardizzando i tipi di dati.

I dati mancanti sono stati gestiti utilizzando tecniche come l'imputazione basata su medie o modalità per le colonne numeriche e categoriche, rispettivamente. In alcuni casi, i record con dati mancanti in campi critici sono stati rimossi per mantenere l'integrità del dataset. Successivamente, i dati sono stati trasformati in un formato adatto per l'analisi, standardizzando i valori e garantendo la coerenza tra le colonne.

## 2.2) Descrizione delle Feature

Durante il progetto, sono state selezionate feature rilevanti per ogni macroarea al fine di ottimizzare le prestazioni dei modelli:

* **Età e Genere**: Utilizzate nelle reti bayesiane per stimare la probabilità che un utente valuti positivamente un film, integrando informazioni demografiche.
* **Generi dei Film**: Usati per identificare le preferenze degli utenti sia nel Logic-Based Recommender che in Random Walk. Questi hanno permesso di individuare film con caratteristiche simili a quelli già apprezzati.
* **Similarità tra Utenti e Film**: Calcolata tramite metriche come Pearson o Cosine, queste sono state utilizzate nel KNN per raccomandare film basati sulle preferenze degli utenti più simili.
* **Valutazioni Esplicite**: Fondamentali in tutte le tecniche per addestrare e valutare i modelli, consentendo di identificare pattern di comportamento e feedback.

Queste feature sono state scelte per bilanciare la semplicità e l'efficacia, considerando i vincoli del dataset MovieLens 100K e gli obiettivi del progetto.

# Ricerca

L'obiettivo principale della macroarea della ricerca è stato esplorare tecniche basate sulla rappresentazione grafica per identificare film potenzialmente interessanti per gli utenti. Dopo una valutazione approfondita delle soluzioni possibili, è stato scelto l'approccio del **Random Walk**, in quanto in grado di sfruttare la similarità tra i film all'interno di un grafo.

Nel grafo, i nodi rappresentano i film, mentre le connessioni tra i nodi sono basate sulla similarità tra i film stessi, calcolata in base ai generi cinematografici. Per ottenere questa similarità, è stata impiegata la tecnica del **TF-IDF** (Term Frequency-Inverse Document Frequency) combinata con la **cosine similarity**, garantendo una rappresentazione accurata e dettagliata dei generi. Questa scelta ha permesso di catturare le affinità tra i film, anche quando non strettamente legati da una semplice somiglianza nel titolo o trama.

## 3.1) Algoritmo Random Walk

L'algoritmo di **Random Walk** inizia da un set di film che l'utente ha valutato positivamente, e successivamente esplora il grafo seguendo le connessioni tra i film simili. Per migliorare la coerenza delle raccomandazioni, è stato introdotto un **bias** che favorisce i film già apprezzati dall'utente. Questo bias consente di orientare la ricerca verso i gusti personali dell'utente, migliorando così la pertinenza delle raccomandazioni. Inoltre, per contrastare problemi di bassa precisione, è stato ridotto il numero di passi del **Random Walk**, limitando la ricerca ai film che sono apparsi più frequentemente durante il cammino.

Un altro passo importante è stato l'inclusione di un filtro che seleziona solo i film con alta frequenza e che sono fortemente simili ai generi preferiti dall'utente, migliorando il **richiamo** del sistema.

## 3.2) Risultati di Precisione e Recall

Le performance del metodo sono state valutate utilizzando le metriche **Precision** e **Recall**, che sono utilizzate per confrontare i film raccomandati con quelli effettivamente apprezzati dall'utente. I valori ottenuti sono i seguenti:

* **Precisione**: 0.67
* **Recall**: 0.06

Questi risultati sono stati rappresentati graficamente per una migliore comprensione delle prestazioni.

Immagine che contiene testo, schermata, diagramma, Rettangolo

Descrizione generata automaticamente

**Discussione dei Risultati**

I risultati mostrano che il valore di **Precision** è abbastanza alto, indicando che circa il 67% dei film raccomandati sono stati effettivamente apprezzati dall'utente. Tuttavia, il valore di **Recall** molto basso (0.06) suggerisce che il sistema non è riuscito a coprire una vasta gamma di film che potrebbero essere di interesse per l'utente. Questo potrebbe essere dovuto al fatto che l'algoritmo si concentra troppo sui film più simili a quelli già apprezzati, senza esplorare sufficientemente nuove possibilità.

In particolare, l'introduzione del bias per favorire i film già apprezzati ha sicuramente migliorato la precisione, ma potrebbe aver limitato la varietà delle raccomandazioni. Inoltre, la limitazione dei passi del Random Walk ha ristretto troppo l'esplorazione del grafo, riducendo la possibilità di scoprire film che potrebbero essere interessanti ma non immediatamente simili ai film già valutati.

## 3.3) Possibili Miglioramenti

Per migliorare i risultati, una strategia possibile sarebbe quella di integrare altre tecniche di apprendimento supervisionato o di esplorazione più ampia dei grafi, che possano bilanciare meglio la **Precision** e il **Recall**. Un'altra possibilità sarebbe quella di ottimizzare ulteriormente il processo di selezione dei film simili, utilizzando una **normalizzazione** delle similarità o un sistema di **pesatura dinamica** dei generi, in modo da considerare con maggiore attenzione le preferenze espresse dall'utente.

# Apprendimento Supervisionato

L'obiettivo principale della macroarea dell'apprendimento supervisionato è stato migliorare la capacità predittiva del sistema di raccomandazione attraverso l'utilizzo di metodi di validazione robusti. In particolare, è stata adottata la tecnica del K-Fold Cross Validation per testare l'affidabilità dei modelli e garantire una valutazione accurata delle prestazioni su diversi sottoinsiemi dei dati.

La K-Fold Cross Validation è una tecnica che suddivide il dataset in K sottoinsiemi, o "fold", utilizzando alternativamente ogni fold come set di test e il restante come set di addestramento. Questo approccio consente di ridurre il rischio di overfitting e fornisce una stima più robusta della performance del modello, in quanto ogni esempio del dataset viene utilizzato sia per l'addestramento che per il test. In questo progetto, il K-Fold è stato utilizzato per validare il modello di apprendimento supervisionato impiegato nel sistema di raccomandazione, in modo da ottimizzare le previsioni dei film da consigliare in base alle preferenze degli utenti.

## 4.1) Implementazione del K-Fold Cross Validation

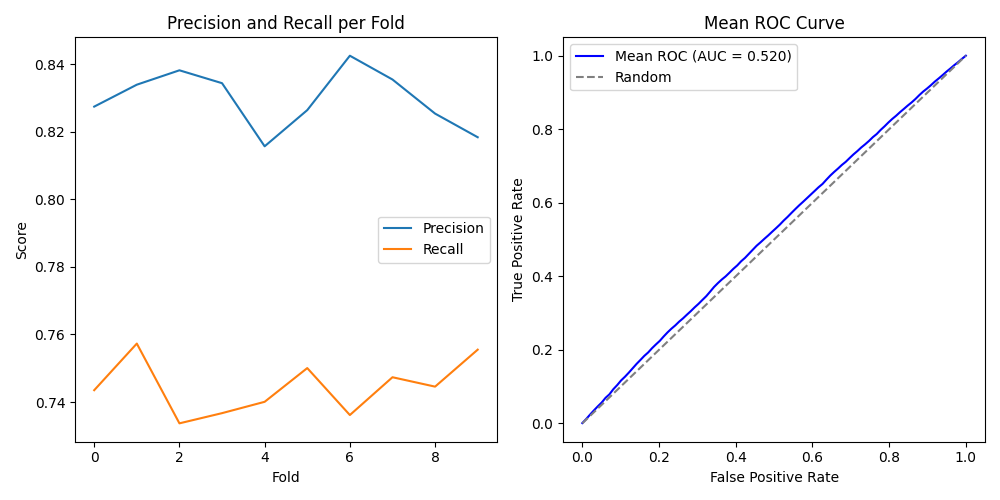
La K-Fold Cross Validation è una tecnica che suddivide il dataset in K sottoinsiemi, o "fold", utilizzando alternativamente ogni fold come set di test e il restante come set di addestramento. Nel nostro caso, è stato adottato un valore di K=10, il che ha comportato l'addestramento e il test del modello su 10 diversi sottoinsiemi dei dati. Questo approccio ha consentito di utilizzare ogni dato sia per l'addestramento che per il test, fornendo una stima complessiva più robusta della performance del sistema di raccomandazione.

## 4.2) Risultati della K-Fold Cross Validation

La K-Fold Cross Validation ha mostrato performance variabili a seconda dei fold, ma ha consentito di ottenere una valutazione più precisa del modello. I risultati principali, calcolati su tutte le iterazioni di test, sono stati i seguenti:

* **Precisione Media:** 0.831
* **Recall Medio:** 0.741
* **AUC Medio (Area Under Curve):** 0.525

Questi risultati suggeriscono che il sistema ha una buona capacità di raccomandare film con alta precisione, riuscendo a identificare correttamente circa l'83% dei film che l'utente potrebbe effettivamente apprezzare, mentre ha anche una buona capacità di recupero (recall), riuscendo a individuare circa il 74% dei film rilevanti. Tuttavia, l'AUC medio relativamente basso (0.525) indica che il modello potrebbe migliorare nella separazione tra film rilevanti e non rilevanti.



## 4.3) Possibili Miglioramenti

Per migliorare ulteriormente i risultati del sistema di raccomandazione, potrebbero essere adottate diverse strategie:

* **Ottimizzazione dei parametri:** La ricerca a griglia (grid search) potrebbe essere utilizzata per affinare ulteriormente i parametri del modello KNN, così come per esplorare altre configurazioni che potrebbero portare a un miglioramento delle performance.
* **Modelli più avanzati:** L'adozione di modelli più complessi come reti neurali (es. MLP) o metodi di ensemble potrebbe migliorare ulteriormente la qualità delle previsioni, combinando i punti di forza di più modelli.
* **Pesatura delle caratteristiche:** Esplorare metodi avanzati di pesatura delle caratteristiche degli utenti e degli oggetti potrebbe portare a previsioni più personalizzate e precise, migliorando la capacità del sistema di adattarsi alle diverse preferenze degli utenti.

In generale, l'approccio di K-Fold Cross Validation ha permesso di ottenere una valutazione solida delle prestazioni del sistema, ma c'è ampio margine per miglioramenti, specialmente attraverso una maggiore ottimizzazione dei modelli e l'integrazione di tecniche più avanzate.

# Ragionamento Probabilistico

Il ragionamento probabilistico rappresenta una componente fondamentale del sistema di raccomandazione sviluppato. In questo progetto, è stato implementato utilizzando un modello Random Forest per prevedere se un utente valuterà positivamente un film. Questa metodologia si basa su un approccio supervisionato che sfrutta tecniche di classificazione avanzate e ottimizzazioni mirate per migliorare la qualità delle previsioni.

L'obiettivo principale era quello di integrare informazioni demografiche e preferenze sui generi cinematografici per fornire previsioni personalizzate. Il processo di implementazione è stato suddiviso in diverse fasi, ciascuna progettata per garantire la massima efficienza e robustezza del modello.

## 5.1) Processo di Implementazione del Grid Search

La realizzazione del ragionamento probabilistico è iniziata con la creazione di un dataset unificato, che combina informazioni sugli utenti, i film e le valutazioni. Questi dati sono stati pre-processati e trasformati per renderli adatti all'analisi:

1. **Preparazione del Dataset**:  
   I dati iniziali sono stati organizzati in modo da includere sia attributi demografici, come l'età e il genere, sia dettagli relativi ai film, come i generi cinematografici principali (*Action*, *Comedy*, *Drama*, *Romance*, *Sci-Fi*). La variabile target è stata definita in termini di una classificazione binaria:
   * **Positive**: valutazioni pari o superiori a 4;
   * **Negative**: valutazioni inferiori a 4.

Questa suddivisione ha permesso di concentrare l'analisi sul comportamento degli utenti e sulle loro preferenze, con l'obiettivo di individuare schemi predittivi.

1. **Gestione dello Squilibrio delle Classi**:  
   Per affrontare il problema dello squilibrio tra valutazioni positive e negative, è stata utilizzata la tecnica **SMOTE** (Synthetic Minority Oversampling Technique). Questo approccio ha generato campioni sintetici per la classe minoritaria, garantendo un training più equilibrato del modello.
2. **Ottimizzazione del Modello**:  
   Per migliorare la performance del modello, è stata implementata una **Grid Search**. Questo processo ha permesso di testare diverse configurazioni di iperparametri del Random Forest, tra cui:
   * Numero di alberi nella foresta (*n\_estimators*).
   * Profondità massima degli alberi (*max\_depth*).
   * Numero minimo di campioni necessari per uno split o una foglia.
   * Metodo di selezione delle feature (*max\_features*).

## 5.2) Valutazione delle Performance e Analisi dei Risultati

Il modello è stato valutato utilizzando un dataset di test separato. Le metriche principali calcolate includono:

* + **Accuratezza**: misura della percentuale di previsioni corrette.
  + **Precisione**: capacità del modello di evitare falsi positivi.
  + **Recall**: capacità del modello di identificare correttamente le valutazioni positive.

Immagine che contiene testo, schermata, Rettangolo, diagramma

Descrizione generata automaticamente

I risultati ottenuti mostrano una performance moderata, con un'accuratezza complessiva del 58% e valori di precisione e recall attorno a 0.6. Sebbene queste metriche indichino un equilibrio tra falsi positivi e falsi negativi, evidenziano anche margini di miglioramento, soprattutto in termini di discriminazione tra le classi.

Immagine che contiene testo, schermata, diagramma, numero

Descrizione generata automaticamente

* **True Positive e True Negative** suggeriscono una buona capacità del modello di riconoscere correttamente entrambe le classi.
* **False Positive e False Negative** indicano aree di miglioramento, poiché il modello fatica a distinguere alcune valutazioni.

L'ottimizzazione degli iperparametri tramite Grid Search ha migliorato la robustezza del modello, selezionando configurazioni che massimizzano l'accuratezza. Tuttavia, l'aggiunta di nuove feature o l'adozione di modelli alternativi potrebbe ulteriormente incrementare le performance.

## 5.3) Conclusioni e Sviluppi Futuri

L'approccio probabilistico implementato ha dimostrato la sua validità nel contesto del sistema di raccomandazione. Tuttavia, ulteriori miglioramenti possono essere ottenuti esplorando nuove direzioni:

* **Feature Engineering Avanzato**: includere variabili contestuali o temporali per affinare le previsioni.
* **Espansione del Dataset**: utilizzare dataset più ampi per aumentare la generalizzazione del sistema.

# Ragionamento Logico

Il ragionamento logico costituisce un elemento chiave del sistema di raccomandazione sviluppato, volto a generare suggerimenti personalizzati basandosi sui generi dei film apprezzati dagli utenti. Questa metodologia utilizza una combinazione di regole logiche per dedurre le preferenze di genere di un utente e generare raccomandazioni pertinenti.

Il ragionamento logico, in questo caso, si concentra su due aspetti principali:

1. Identificare i generi cinematografici preferiti dall'utente basandosi su film precedentemente apprezzati.
2. Generare raccomandazioni che condividano almeno due generi predominanti tra quelli preferiti dall'utente.

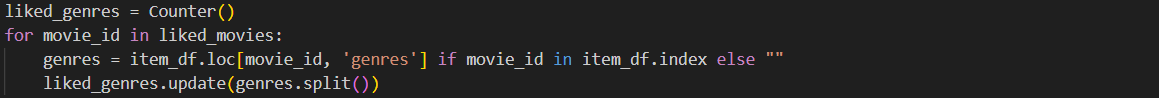
## 6.1) Processo di Implementazione

La realizzazione del ragionamento logico è stata suddivisa in diverse fasi operative, ognuna delle quali è stata progettata per massimizzare la precisione e la rilevanza delle raccomandazioni:

1. **Preparazione del Dataset:**  
   I dati relativi ai film includono attributi binari per i generi cinematografici (ad esempio, Action, Drama, Comedy) e sono stati trasformati per costruire una rappresentazione testuale dei generi associati a ciascun film. Nel caso di film senza generi definiti, è stata applicata una gestione specifica per evitare raccomandazioni prive di significato.



1. **Identificazione dei Generi Preferiti:**  
   Il sistema analizza i film apprezzati dall'utente per individuare i generi predominanti. È stato utilizzato un approccio basato su conteggi ponderati, identificando i generi con più di una occorrenza come principali candidati per le raccomandazioni.



1. **Generazione delle Raccomandazioni:**  
   I film raccomandati devono condividere almeno due generi con quelli preferiti dall'utente. Inoltre, le raccomandazioni vengono ordinate sulla base della valutazione media degli utenti, quando disponibile, per garantire maggiore qualità nei suggerimenti.



1. **Fallback Logico:**  
   Nel caso in cui i generi preferiti dell'utente risultassero poco definiti (ad esempio, "unknown"), il sistema applica una logica di fallback, generando raccomandazioni basate sui film con le valutazioni più alte.



## 6.2) Valutazione delle Performance e Analisi dei Risultati

Il sistema è stato testato su un dataset di esempio e le sue prestazioni sono state misurate utilizzando precisione e richiamo:

* **Precisione:** indica la proporzione di film raccomandati che sono effettivamente apprezzati dall'utente.
* **Richiamo:** misura la capacità del sistema di individuare correttamente i film apprezzati.

Immagine che contiene testo, schermata, diagramma, Rettangolo

Descrizione generata automaticamente

Questi valori mostrano una forte capacità del modello di identificare correttamente le preferenze degli utenti (richiamo elevato), ma evidenziano un margine di miglioramento nella riduzione dei falsi positivi.

## 6.3) Conclusioni e Sviluppi Futuri

Il ragionamento logico implementato ha dimostrato di essere efficace per raccomandare film basati sui generi preferiti, mantenendo un equilibrio tra semplicità e accuratezza. Tuttavia, ulteriori miglioramenti potrebbero essere apportati:

* **Ottimizzazione del Metodo di Selezione dei Generi:** Includere un'analisi più approfondita delle correlazioni tra i generi.
* **Espansione dei Criteri di Fallback:** Considerare anche il contesto temporale o geografico delle valutazioni.
* **Integrazione con Modelli Probabilistici:** Combinare il ragionamento logico con metodi avanzati di apprendimento automatico per migliorare la precisione delle previsioni.

# Confronto delle Soluzioni

Per identificare l'approccio migliore per il sistema di raccomandazione, sono state considerate quattro macroaree: Ricerca, Apprendimento Supervisionato, Ragionamento Probabilistico e Ragionamento Logico. Ciascuna soluzione è stata implementata e valutata utilizzando metriche standard come precision, recall e accuratezza. Di seguito, una tabella che riassume i principali risultati e caratteristiche di ogni soluzione:

| **Soluzione** | **Precision** | **Recall** | **Punti di Forza** | **Limitazioni** |
| --- | --- | --- | --- | --- |
| **Random Walk** | 0.67 | 0.06 | Sfrutta la similarità tra i film in un grafo; buona precisione. | Recall molto basso; esplorazione limitata. |
| **K-Fold Cross Validation** | 0.83 | 0.74 | Solido metodo di validazione; consente una stima robusta delle performance. | Richiede risorse computazionali elevate; precisione migliorabile in alcuni scenari. |
| **Grid Search con Random Forest** | 0.60 | 0.60 | Integrazione di SMOTE per bilanciare le classi; robustezza garantita da GridSearchCV. | Performance moderate; dipendenza da feature predefinite. |
| **Ragionamento Logico** | 0.40 | 1.00 | Recall molto elevato; semplice implementazione logica basata sui generi preferiti. | Precisione bassa; fallisce in caso di dati incompleti o generi sconosciuti. |

## 7.1) Soluzione Selezionata

**K-Fold Cross Validation** è stata scelta come la soluzione migliore per il sistema di raccomandazione. Le ragioni includono:

* **Equilibrio tra precision e recall:** L'alta precisione (0.83) garantisce raccomandazioni pertinenti, mentre un recall di 0.74 assicura una buona copertura delle preferenze.
* **Robustezza metodologica:** La validazione incrociata riduce il rischio di overfitting, fornendo una stima accurata delle prestazioni.
* **Flessibilità:** Il metodo può essere ulteriormente migliorato tramite l'integrazione di feature più complesse o modelli avanzati.

Nonostante il tempo computazionale richiesto, i benefici in termini di accuratezza e affidabilità lo rendono il candidato ideale per implementazioni pratiche.

## 7.2) Sviluppi Futuri

Nonostante le solide performance ottenute con il K-Fold Cross Validation, sono possibili ulteriori miglioramenti per potenziarne l'efficacia. Di seguito alcune proposte di sviluppi futuri:

1. **Feature Engineering Avanzato**:

* Integrare nuove variabili, come informazioni contestuali (ad esempio, tempo di visualizzazione o preferenze temporali degli utenti) o dati derivati da fonti esterne (come recensioni o classifiche di film).

1. **Iperparametri Dinamici**:

* Utilizzare metodi avanzati di ottimizzazione iperparametrica, come Random Search o Bayesian Optimization, per migliorare ulteriormente le prestazioni del modello.

1. **Espansione del Dataset**:

* Integrare dataset più grandi o aggiornati per aumentare la generalizzazione del modello e testare la sua robustezza su un numero maggiore di utenti e film.

1. **Implementazione di Modelli Avanzati**:

* Sperimentare algoritmi di apprendimento profondo, come reti neurali, per catturare relazioni complesse tra le feature.
* Combinare il K-Fold con ensemble di modelli per ottenere previsioni più accurate e robuste.

# Riferimenti Bibliografici

**Ricerca**: D. Poole, A. Mackworth: Artificial Intelligence: Foundations of Computational Agents. 3/e. Cambridge University Press (Ch. 3)

**Apprendimento** **Supervisionato**: D. Poole, A. Mackworth: Artificial Intelligence: Foundations of Computational Agents. 3/e. Cambridge University Press (Ch. 7)

**Ragionamento** **Logico**: D. Poole, A. Mackworth: Artificial Intelligence: Foundations of Computational Agents. 3/e. Cambridge University Press (Ch. 5)

**Ragionamento** **Probabilistico**: D. Poole, A. Mackworth: Artificial Intelligence: Foundations of Computational Agents. 3/e. Cambridge University Press (Ch. 9-10)