



Laurea Triennale in Informatica - Università di Salerno
Corso di Fondamenti di Intelligenza Artificiale
Professore F. Palomba

Sommario

1	Introduzione: sistema attuale e sistema proposto	2
2	Descrizione dell'agente	2
2.1	Obiettivi	2
2.2	Specifica PEAS	3
2.3	Analisi del problema	4
3	Raccolta, analisi e preprocessing dei dati	4
3.1	Scelta del dataset	4
3.2	Analisi e scrematura del dataset	4
3.2.1	Tabella Movies	4
3.2.2	Tabella Links	4
3.2.3	Tabella Genome-Scores	4
3.2.4	Tabella Genome-Tags	4
3.2.5	Tabella Ratings	4
3.2.6	Tabella Tags	4
3.3	Preprocessing dei Dati	5
3.3.1	Caricamento dei dati	5
3.3.2	Unione dei Dataset	5
3.3.3	Preprocessing per il Clustering	5
4	Algoritmo di clustering	5
4.1	Scelta dell'algoritmo di clustering	5
4.1.1	K-Means	5
4.1.2	Scelta del numero di Cluster	5
4.1.3	Implementazione dell'algoritmo KMeans	5
5	Integrazione con il sistema	5
5.1	Architettura e Funzionalità Principali	5
5.2	Logica Implementata:	5
5.2.1	Risposta e Error Handling	5
5.3	Logica Interna e Dettagli Tecnici	5
5.4	Implementazione dello script:	5
6	Glossario	5



1 Introduzione: sistema attuale e sistema proposto

Oggi, con il cambiamento dello stile di vita delle persone, che preferiscono sempre più restare a casa, e a seguito della pandemia, le piattaforme di streaming hanno ottenuto un successo crescente. Già ampiamente utilizzate, queste piattaforme permettono di guardare film e serie TV su qualsiasi dispositivo connesso a Internet. Con l'aumento della loro popolarità, hanno sentito l'esigenza di rendere l'esperienza dell'utente sempre più personalizzata, introducendo nuove funzionalità.

Inizialmente, i problemi principali erano due: da un lato, l'utente accedeva alla piattaforma sapendo già cosa guardare, limitandosi semplicemente a visualizzare i contenuti scelti per poi uscire; dall'altro lato, numerosi utenti, incerti su cosa guardare, si trovavano disorientati dalla vasta offerta disponibile e, di conseguenza, spesso abbandonavano l'idea di selezionare un contenuto.

A partire dal 2006, con l'introduzione dei primi algoritmi di raccomandazione su Netflix, è iniziata una trasformazione che ha rivoluzionato il modo in cui gli utenti fruiscono dei contenuti. Inizialmente, questi algoritmi si limitavano a proporre i contenuti più popolari, più visti e le ultime uscite. Successivamente, si sono evoluti, ponendo l'utente al centro dell'esperienza. Oggi, i contenuti vengono consigliati sulla base delle preferenze individuali, tenendo conto non solo dei contenuti già visti e dei generi preferiti, ma anche di attori, registi e altre caratteristiche, come le case di produzione.

Utilizzando piattaforme di streaming come Netflix o Prime Video, si può apprezzare il progresso degli algoritmi di intelligenza artificiale, che rendono l'esperienza dell'utente ancora più immersiva e dinamica.

Il progetto proposto mira a realizzare una web app ottimizzata per smart TV, che consenta agli utenti di scoprire nuovi contenuti da guardare. L'app offrirà la possibilità di accedere a informazioni dettagliate sui film, inclusi trailer e indicazioni sulle piattaforme dove sono disponibili. In particolare, il sistema sfrutterà i dati derivati dalla lista di preferiti dell'utente per consigliare contenuti in linea con i suoi gusti, considerando elementi come genere, cast, produzione e anno di uscita. Il tutto sarà supportato da tecniche di intelligenza artificiale, simili a quelle già adottate dalle principali piattaforme di streaming.

2 Descrizione dell'agente

2.1 Obiettivi

Lo scopo del progetto è quello di realizzare un agente intelligente che sia in grado di:

- Generare un insieme di film basandosi sulla lista dei preferiti dell'utente;
- Consigliare una lista di contenuti correlati nella schermata di dettaglio del singolo film;
- Nel consigliare i film, non deve tenere conto di una singola caratteristica, ma di tutte le informazioni che possono essere utilizzate per personalizzare al meglio la lista di film.



2.2 Specifica PEAS

Diamo ora la specifica PEAS dell'agente:

PEAS	Descrizione
Performance	La misura di performance dell'agente è la sua capacità di avvicinarsi quanto più possibile a una situazione ideale nella quale vengano mostrati agli utenti esattamente i film che loro desiderano guardare e ai quali sono interessati.
Environment	<p>L'ambiente in cui opera l'agente è lo spazio dell'utente dell'app, con le sue preferenze, unito a quello dei possibili film e le loro caratteristiche. L'ambiente è:</p> <ul style="list-style-type: none">• Dinamico, in quanto nel corso delle elaborazioni dell'agente, l'utente aggiunge un film alla sua lista di preferiti, cambiando in tal modo le sue preferenze;• Sequenziale, in quanto le preferenze passate dell'utente influenzano le decisioni future dell'agente;• L'ambiente è Discreto perchè gli stati (film) e le azioni (suggerimenti) sono limitati e finiti, basati su attributi come genere, cast e tag, che hanno un numero definito di valori;• Completamente osservabile, in quanto si ha accesso a tutte le informazioni relative al catalogo di contenuti e alle preferenze dell'utente in ogni momento;• Non deterministico, in quanto lo stato dell'ambiente cambia indipendentemente dalle azioni dell'agente;• Non noto, in quanto l'agente non può conoscere a priori il risultato esatto delle sue azioni, in termini di efficienza;• Stocastico e unico, in quanto l'unico agente che opera in questo ambiente è quello in oggetto.
Actuators	Gli attuatori dell'agente consistono nella lista dei film consigliati sulla base delle preferenze dell'utente e il relativo carrello che li mostra.
Sensors	I sensori dell'agente consistono nel bottone per aggiungere un film ai preferiti e la pagina dei dettagli del singolo film.

Tabella 1: Specifica PEAS dell'agente



2.3 Analisi del problema

L'intenzione di sviluppare questo agente nasce dallo studio dell'API di TMDb, che offre algoritmi di raccomandazione. Tuttavia, sono state individuate due principali limitazioni:

- L'API richiede la registrazione dell'utente, mentre si è scelto di rendere il sistema accessibile senza obbligo di login, consentendo all'utente di usufruire delle funzionalità anche senza registrarsi.
- TMDb è soggetto al "cold start problem", ovvero l'algoritmo ha difficoltà a fare raccomandazioni accurate quando l'utente non ha ancora espresso preferenze. Per ovviare a questo, è stata implementata una fase di "User Onboarding", durante la quale all'utente viene chiesto di indicare le sue preferenze all'avvio dell'applicazione.

Si è scelto di interpretare il problema come un problema di **Clustering**. L'idea è stata di analizzare la lista dei film preferiti dall'utente, estrarre le informazioni rilevanti da questi, e generare una lista di film che rispecchiassero i suoi gusti, suggerendo contenuti che potessero piacergli.

3 Raccolta, analisi e preprocessing dei dati

3.1 Scelta del dataset

Andando a parlare del dataset necessario per la creazione del modello di machine learning, si potevano considerare due possibili approcci:

1. **Creare** un dataset da zero, raccogliendo informazioni su tutti i film, anche quelli più vecchi, e concentrandosi sugli aspetti salienti come generi, attori, anno di uscita, tag, ecc.;
2. **Cercare** un dataset già formato e adattarlo alle specifiche esigenze del progetto.

La prima soluzione risultava impraticabile, dato l'enorme numero di film presenti nella storia e la varietà di caratteristiche che sarebbe stato necessario raccogliere e inserire. Tale approccio avrebbe portato alla creazione di un dataset poco popolato e privo di molte informazioni importanti.

Si è quindi deciso di procedere cercando in rete un dataset già esistente. Dopo aver considerato diverse opzioni, si è scelto il dataset MovieLens, che si è rivelato il più adatto alle necessità del progetto.

3.2 Analisi e scrematura del dataset

3.2.1 Tabella Movies

3.2.2 Tabella Links

3.2.3 Tabella Genome-Scores

3.2.4 Tabella Genome-Tags

3.2.5 Tabella Ratings

3.2.6 Tabella Tags



3.3 Preprocessing dei Dati

3.3.1 Caricamento dei dati

3.3.2 Unione dei Dataset

3.3.3 Preprocessing per il Clustering

4 Algoritmo di clustering

4.1 Scelta dell'algoritmo di clustering

4.1.1 K-Means

4.1.2 Scelta del numero di Cluster

4.1.3 Implementazione dell'algoritmo KMeans

5 Integrazione con il sistema

5.1 Architettura e Funzionalità Principali

5.2 Logica Implementata:

5.2.1 Risposta e Error Handling

5.3 Logica Interna e Dettagli Tecnici

5.4 Implementazione dello script:

6 Glossario