Relazione Caso di Studio

Ingegneria Della Conoscenza

UnibaVision

Pasquale Laterza mat. 763611 p.laterza10@studenti.uniba.it

Giammarco Giovinazzi mat. 760675 g.giovinazzi4@studenti.uniba.it

Repository GitHub:

https://github.com/giamma30/Ingegneria_della_Conoscenza_2023-2024.git

AA 2023-2024

INTRODUZIONE

L’obiettivo del caso di studio è quello di sfruttare in modo approfondito le informazioni relative a film e serie TV presenti sulla piattaforma Netflix. Questo studio si propone di analizzare vari aspetti dei contenuti disponibili, come il genere, il cast, la regia.

In particolare, il progetto è suddiviso in tre sezioni principali:

- **Preprocessing**: mira ad adattare i dati per renderli più adatti all'uso successivo.
- **Base di conoscenza**: permette all'utente di porre domande sulle conseguenze logiche e di ricevere risposte dalla macchina stessa grazie all’utilizzo delle spiegazioni a livello di conoscenza.
- **Classificazione**: Individua il classificatore più performante per la predizione del genere di un film fornito dall’utente.
- **Recommender system**: suggerisce all'utente film simili a quello fornito, utilizzando tecniche di clustering basate sull'apprendimento non supervisionato.

Strumenti utilizzati nel progetto:

Il gruppo ha deciso di utilizzare come linguaggio Python (www.Python.org). Come servizio di hosting è stato scelto GitHub, per gli ottimi sistemi di collaborazione ove risiede la repository del progetto.

Tutte le librerie e versioni utilizzate sono definite nel file “requirements.txt”.

In particolare:

- **Pandas**: libreria utile per la manipolazione e l'analisi dei dati, utilizzata nella sezione di preprocessing.
- **scikit-learn**: libreria per tecniche di apprendimento, utilizzata nelle sezioni di classificazione e clusterizzazione.
- **matplotlib**: libreria per la creazione di grafici, utilizzata nelle sezioni di classificazione e clusterizzazione.

- **kmodes**: libreria per la clusterizzazione mediante l'algoritmo K-Modes, utilizzata nella sezione di clusterizzazione.
- **fuzzywuzzy**: libreria per calcolare la similarità tra stringhe, utilizzata nella sezione di clusterizzazione.
- **numpy**: libreria per eseguire calcoli su vettori e matrici, utilizzata in tutte le sezioni del progetto.

1. PREPROCESSING

I dataset utilizzati nel caso di studio sono stati reperiti dal sito Kaggle in formato csv e sono i seguenti:
- Dataset film Netflix (Netflix_film.csv)
- Dataset film e serie tv Netflix (Netfix_serie_film.csv)
- Dataset IMDB ratings film (IMDb_valutazioni.csv)

Per rendere i dati adatti e conformi alle operazioni da svolgere successivamente, sono state effettuate diverse operazioni di preprocessing:

- Unificazione dei tre dataset per ottenere uno unico finale;
- Eliminazione delle colonne ritenute superflue ai fini del progetto;
- Rimozione dei duplicati;
- Discretizzazione della colonna year, sostituendola con la colonna year_range;
- Riduzione dei generi associati a ciascun film, mantenendone uno unico per ciascuno effettuando la scelta sulla base delle occorrenze dei generi stessi nel dataset e optando per quelli che risultano maggiormente citati;
- Riduzione degli attori presenti nella colonna cast, mantenendone uno unico per ciascun film;
- Inserimento del valore ‘Movie’ nella colonna type per le row che presentavano un valore nullo proveniente dal dataset contenente unicamente film;
- Conversione dei valori della colonna genres da categorici a numerici mediante metodo di conversione delle dummy variables, utile per la successiva operazione di imputation;
- Ridenominazione dei valori nella colonna genres;
- Conversione dei valori nella colonna type, da categorici a numerici mediante tecnica di conversione del label encoder, utile per la successiva operazione di imputation;

- Conversione dei valori nella colonna year_range, da categorici a numerici

mediante tecnica di conversione del label encoder, utile per la successiva operazione di imputation;

- Conversione dei valori nella colonna title, da categorici a numerici mediante tecnica di conversione del label encoder, utile per la successiva operazione di imputation;
- Riduzione dei valori presenti nella colonna country, mantenendone uno unico per ciascun film e conversione degli stessi da categorici a numeri mediante tecnica di conversione del label encoder;
- Feature imputation per i valori della colonna ratings mancanti tramite KNNImputer;
- Standardizzazione dei valori della colonna ratings;
- Feature imputation per i valori della colonna genre mancanti tramite hot-deck imputation;
- Eliminazione delle row con informazioni mancanti su cui l'operazione di values imputation era impossibile da effettuare;

2. BASE DI CONOSCENZA

Una base di conoscenza è un insieme di informazioni strutturate e organizzate che rappresentano la conoscenza su un determinato dominio. Questa conoscenza può includere fatti, regole, concetti, relazioni e vincoli che descrivono il mondo reale o un aspetto specifico di esso.

Quindi, la base di conoscenza o KB è definibile come un insieme di assiomi, cioè delle proposizioni che possono essere asserite essere vere.

La base di conoscenza viene impiegata nel caso di studio per facilitare un'interazione dinamica tra l'utente e il sistema, permettendo uno scambio di domande e risposte specifiche al dominio di interesse, ovvero quello dei film e delle serie TV. Questo scambio di informazioni consente di ottenere risposte pertinenti e dettagliate riguardo a vari aspetti di film e serie.

In pratica, la base di conoscenza funge da supporto informativo che raccoglie e organizza una vasta quantità di dati riguardanti il mondo cinematografico e televisivo. Gli utenti possono porre domande specifiche, e il sistema è in grado di elaborare queste richieste attingendo alle informazioni presenti nella KB, fornendo risposte accurate e rilevanti.

Nello specifico, l'utente può avanzare le seguenti richieste:

- Confermare la corrisponde tra titolo e genere relativi ad un film, attraverso la funzione **askGenereDaTitolo**, che accetta in input entrambi i dati e restituisce in output una risposta affermativa o negativa;
  askGenereDaTitolo(titolo, genere) <=> titolo_genere;

Esempio di funzionamento askGenereDaTitolo(“titolo”,”genere”) ottimale:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 3
INIZIAMO!

1) Dato un titolo e un genere in input, la KB è in grado di dirti se il titolo corrisponde al genere indicato
grazie alla funzione askGenereDaTitolo, rispondendo YES se effettivamente corrisponde, altrimenti NO

2) Dati due titoli in input, la KB è in grado di dirti se il genere dei due film è lo stesso oppure no grazie
alla funzione askStessoGenere, rispondendo YES se corrispondono, NO altrimenti

Digitare il numero della funzione che si vuole eseguire : 1
Digitare il titolo del film: all about love
Digitare il genere del film: romantic
YES
Digitare how per la spiegazione: how
askGenereDaTitolo(all about love,romantic) <=> all about love_romantic
Digitare 'how i' specificando al posto di i il numero dell'atomo : how 1
all about love_romantic <=> True
```

Esempio di funzionamento askGenereDaTitolo(“titolo”,”genere”) non Ottimale:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 3
INIZIAMO!

1) Dato un titolo e un genere in input, la KB è in grado di dirti se il titolo corrisponde al genere indicato
grazie alla funzione askGenereDaTitolo, rispondendo YES se effettivamente corrisponde, altrimenti NO

2) Dati due titoli in input, la KB è in grado di dirti se il genere dei due film è lo stesso oppure no grazie
alla funzione askStessoGenere, rispondendo YES se corrispondono, NO altrimenti

Digitare il numero della funzione che si vuole eseguire :
1
Digitare il titolo del film: anon
Digitare il genere del film: dramas
NO
Digitare how per la spiegazione: how
askGenereDaTitolo(anon,dramas) <=> anon_dramas
Digitare 'how i' specificando al posto di i il numero dell'atomo : how 1
anon_dramas <=> False
```

- Verificare se film diversi appartengano ad uno stesso genere, mediante l'utilizzo della funzione **askStessoGenere**, che accetta in input i titoli dei film in questione;

  askStessoGenere(titolo1, titolo2) <=>
  titolo1_primoGenere and
  titolo2_secondoGenere and
  stessoGenere(primoGenere, secondoGenere),
  dove stessoGenere("genere1","genere2") indica se i generi presenti come parametri sono o meno uguali tra loro.

Esempio di funzionamento di askStessoGenere("titolo1","titolo2") caso ottimale:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 3
INIZIAMO!

1) Dato un titolo e un genere in input, la KB è in grado di dirti se il titolo corrisponde al genere indicato
grazie alla funzione askGenereDaTitolo, rispondendo YES se effettivamente corrisponde, altrimenti NO

2) Dati due titoli in input, la KB è in grado di dirti se il genere dei due film è lo stesso oppure no grazie
alla funzione askStessoGenere, rispondendo YES se corrispondono, NO altrimenti

Digitare il numero della funzione che si vuole eseguire : 2
Digitare il titolo del primo film: valor
Digitare il titolo del secondo film: cargo
YES
Digitare how per la spiegazione: how
askStessoGenere(valor,cargo) <=> valor_dramas and cargo_dramas and generiUguali(dramas,dramas)
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 1
valor_dramas <=> True
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 2
cargo_dramas <=> True
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 3
generiUguali(dramas,dramas) <=> True
```

Esempio di funzionamento di askStessoGenere(“titolo1”,”titolo2”) caso non ottimale:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 3
INIZIAMO!

1) Dato un titolo e un genere in input, la KB è in grado di dirti se il titolo corrisponde al genere indicato
grazie alla funzione askGenereDaTitolo, rispondendo YES se effettivamente corrisponde, altrimenti NO

2) Dati due titoli in input, la KB è in grado di dirti se il genere dei due film è lo stesso oppure no grazie
alla funzione askStessoGenere, rispondendo YES se corrispondono, NO altrimenti

Digitare il numero della funzione che si vuole eseguire : 2
Digitare il titolo del primo film: intelligence
Digitare il titolo del secondo film: nails
NO
Digitare how per la spiegazione: how
askStessoGenere(intelligence,nails) <=> intelligence_dramas and nails_horror and generiUguali(dramas,horror)
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 1
intelligence_dramas <=> True
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 2
nails_horror <=> True
Digitare 'how i' specificando in i il numero dell'atomo per ulteriori informazioni: how 3
generiUguali(dramas,horror) <=> False
```

Abbiamo utilizzato la spiegazione a livello di conoscenza che è un approccio che consente a un sistema basato su conoscenza di fornire ragionamenti e giustificazioni per le sue risposte.

Per ogni query che viene eseguita, ossia ogni interrogazione posta in modo tale da sapere se una proposizione sia conseguenza logica della base di conoscenza, la KB risponderà con YES oppure NO a seconda del tipo di clausola che le viene presentata.

Inoltre, si potrà chiedere la motivazione secondo la quale si è ottenuto un determinato risultato attraverso l’operatore **how** - in questo modo, la KB potrà fornire la motivazione alla base della restituzione di una certa risposta rispetto ad un’altra mostrando le clausole utilizzate per dedurre la risposta.

Infine, l’utente ha la possibilità di richiedere una prova per ogni atomo nel corpo di una clausola.

3. CLASSIFICAZIONE

Uno degli scopi principali del Machine Learning è la classificazione, cioè il problema di indentificare la classe di un nuovo obiettivo sulla base di conoscenza estratta da un training set. Un sistema che classifica è detto classificatore. I classificatori estraggono dal dataset un modello che utilizzano poi per classificare le nuove istanze. Il processo di classificazione si può dividere in tre fasi: Addestramento, Stima dell'accuratezza e Utilizzo del Modello.
Per lo scopo del nostro progetto abbiamo deciso di suddividere i dati in un insieme di training e un insieme di test fissando quest'ultimo al 30%.
La classificazione nel caso di studio è stata utilizzata con lo scopo di predire il genere di un film fornito dall'utente. La variabile target sulla quale effettuare la predizione sarà quindi il "Genere".
Per ottenere il risultato migliore sono stati messi a confronto tre modelli di classificatori:

- KNN
- Random Forest
- Bagging

a. KNN

Uno degli algoritmi più conosciuti nel machine learning è il K-Nearest Neighbors (KNN). Questo classificatore restituisce come output il genere di appartenenza del film dato in input, basando la classificazione sulla pluralità dei voti dei suoi vicini, cioè viene assegnata la classe più presente tra i k film più simili ritrovati, calcolati per similarità dal film da definire dato in input.
È la tecnica più semplice che si può applicare, spesso efficace ma lenta e richiede molta memoria poichè il costo di calcolo è quadratico.

b. RANDOM FOREST

L'RF o Random Forest Classifier è largamente utilizzato per classificazione, regressione e altri task, funziona costruendo una moltitudine di alberi di decisione. Per la classificazione l'output è la classe selezionata dalla maggior parte degli alberi.
La foresta generata dall'algoritmo è addestrata attraverso aggregazione di tipo begging o bootstrap.
L'algoritmo stabilisce il risultato sulla base di predizioni dei decision trees. Esso predice prendendo la media dell'output dei vari alberi; aumentando il numero di alberi si aumenta la precisione del risultato.
Il Random Forest elimina i limiti dell'algoritmo Decision Tree infatti riduce l'overfitting dei dataset e aumenta la precisione.

c. BAGGING

Il Bagging Classifier si basa sull'addestrare più modelli dello stesso tipo, ciascuno su sottoinsiemi casuali del dataset originale e quindi aggrega le loro previsioni individuali (mediante voto o media) per formare una previsione finale. Ogni weak learner viene addestrato in parallelo con un set di addestramento che viene generato estraendo casualmente, con sostituzione, N esempi (o dati) dal dataset originale (dove N è la dimensione del dataset). Il training set per ciascuno dei classificatori di base è indipendente l'uno dall'altro.
Il bagging viene usato soprattutto quando l'obiettivo è ridurre la varianza(overfitting) del classificatore, in modo da evitare che si abbia un'ottima precisione sui dati di addestramento e alte percentuali di errore sui dati di test. Gli stimatori maggiormente considerati sono gli alberi di decisione, definiti in molti casi come base learner del bagging classifier.

d. RISULTATI CLASSIFICATORI

Per valutare le performance di ogni classificatore si è svolto un lavoro di tuning dei parametri. A tale scopo ci si è serviti del metodo GridSearchCV della libreria model-selection del package Sklearn fornito da Python.
Di seguito riportiamo i risultati migliori ottenuti per ogni classificatore:

- KNN best params {metric: 'manhattan', n_neighbors: 1, weights: 'uniform'}
- RF best params {max_features: 'sqrt', n_estimators: 100}
- BAGGING best params {n_estimators: 10}

Sono state effettuate più prove con diversi parametri per ogni classificatore per ottenere la corrispondenza migliore.

	precision	recall	f1-score	support
anime	0.96	1.00	0.98	555
cult	0.89	0.95	0.92	532
fantasy	1.00	1.00	1.00	537
action	0.91	1.00	0.95	560
documentary	0.95	1.00	0.97	553
nature	0.93	1.00	0.96	584
romantic	0.90	0.97	0.94	571
sport	0.95	1.00	0.97	520
thrillers	0.99	1.00	1.00	554
kids	0.93	1.00	0.97	538
dramas	0.93	1.00	0.96	557
horror	0.83	0.84	0.83	568
standup	0.84	0.87	0.85	560
commedies	0.77	0.70	0.74	556
musical	0.61	0.27	0.38	562
accuracy			0.90	8307
macro avg	0.89	0.91	0.89	8307
weighted avg	0.89	0.90	0.89	8307

Figura 1: KNN

	precision	recall	f1-score	support
anime	0.97	1.00	0.98	555
cult	0.90	0.95	0.92	532
fantasy	1.00	1.00	1.00	537
action	0.93	1.00	0.96	560
documentary	0.96	1.00	0.98	553
nature	0.92	1.00	0.95	584
romantic	0.90	0.97	0.93	571
sport	0.93	1.00	0.97	520
thrillers	0.99	1.00	1.00	554
kids	0.90	1.00	0.95	538
dramas	0.94	1.00	0.97	557
horror	0.82	0.83	0.83	568
standup	0.85	0.86	0.85	560
commedies	0.76	0.70	0.73	556
musical	0.68	0.30	0.41	562
accuracy			0.91	8307
macro avg	0.90	0.91	0.90	8307
weighted avg	0.90	0.91	0.89	8307

Figura 2: RF

	precision	recall	f1-score	support
anime	0.97	1.00	0.98	555
cult	0.87	0.94	0.90	532
fantasy	0.99	1.00	1.00	537
action	0.93	0.99	0.96	560
documentary	0.96	1.00	0.98	553
nature	0.92	0.99	0.96	584
romantic	0.91	0.97	0.94	571
sport	0.93	1.00	0.96	520
thrillers	0.98	1.00	0.99	554
kids	0.92	1.00	0.96	538
dramas	0.93	1.00	0.96	557
horror	0.82	0.82	0.82	568
standup	0.86	0.85	0.85	560
commedies	0.75	0.73	0.74	556
musical	0.63	0.27	0.37	562
accuracy			0.90	8307
macro avg	0.89	0.90	0.89	8307
weighted avg	0.89	0.90	0.89	8307

Figura 3: Bagging

L’esito di questo confronto ci ha portato a scegliere il Random Forest come classificatore per la predizione del genere.

Di seguito si riporta un esempio di funzionamento del classificatore:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 2
INIZIAMO!

Inserire il nome del film o serie TV che hai apprezzato: ghost
ghost è un film? (s/n)
 -> s
Inserire il paese di produzione:
 -> thailand
Inserire l`anno di rilascio:
 -> 2016
Inserire un membro del cast:
 -> scout taylor-compton
Inserire un voto da 1 a 10 sul film/serie TV:
 -> 9
Il genere del film o serie TV da te inserito è horror
```

4. CLUSTERING

Il clustering è una metodologia di apprendimento non supervisionato che consente di identificare e raggruppare elementi simili appartenenti a dataset di grandi dimensioni, creando cluster ossia gruppi di questi ultimi che risultano conformi ad elementi medi, detti centroidi.

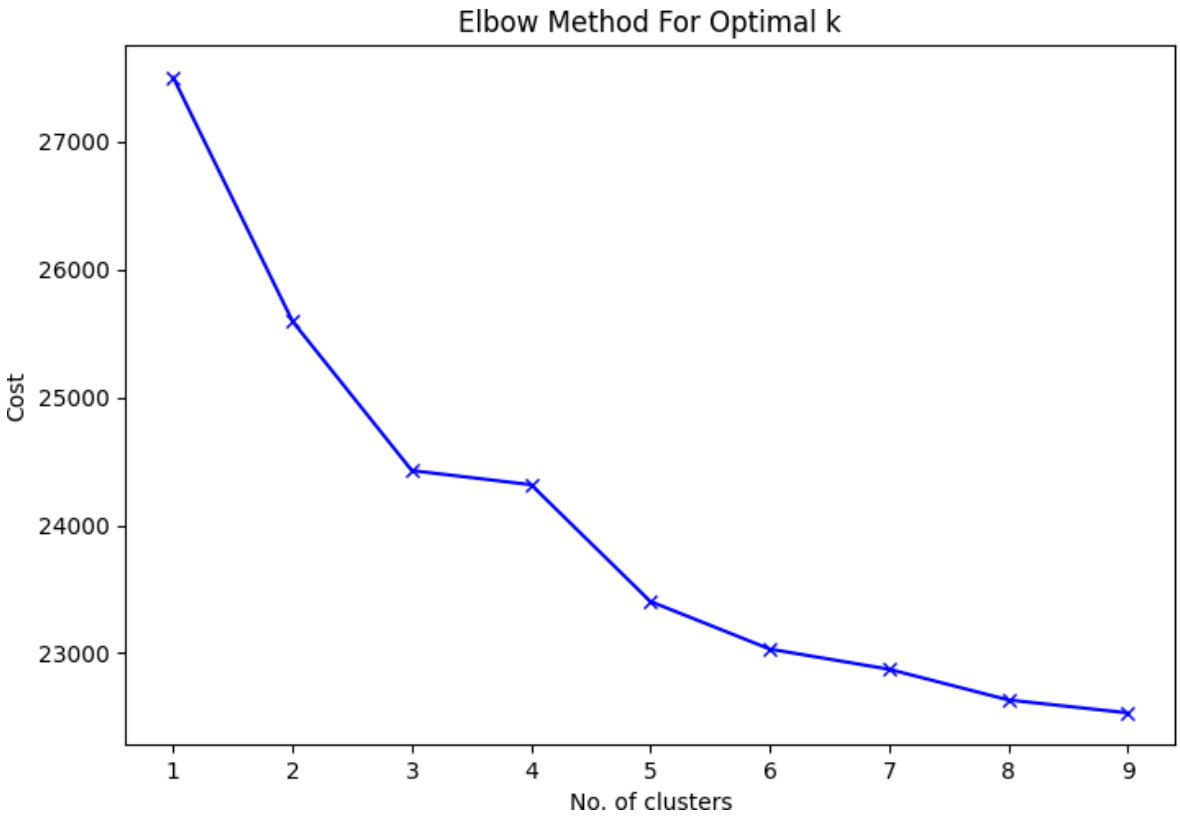
Nello specifico, abbiamo scelto di adottare questa tecnica al fine di poter individuare delle nuove similarità e correlazioni tra i dati che non dipendessero unicamente dal genere dei film interessati, per poterle poi sfruttare alla base di un recommender system.

e. CLUSTER

Considerando la forte presenza di dati categorici nel dataset, abbiamo deciso di optare per l'algoritmo K- Modes.
Questo algoritmo estende l'algoritmo K-Means utilizzando una misura di similarità dedita ad elementi categorici, sostituendo l'utilizzo della media con l'utilizzo della moda ed utilizzando un metodo frequency-based utile per minimizzare la funzione di costo.

Si è scelto di individuare 3 cluster, centroidi, sfruttando il 'metodo del gomito', ossia un metodo empirico utile a trovare il numero ottimale di cluster per un dataset all'interno di un range determinato.

Nello specifico, il range scelto è rimasto limitato al di sotto del numero di generi dei film presenti, in modo tale da poter individuare correlazioni non fortemente legate a questi.



f. RECOMMENDER SYSTEM

Per quanto riguarda il sistema di raccomandazione, è stato adottato un approccio basato sui contenuti, incrociando gli attributi dei film e delle serie tv presenti nel dataset con uno apprezzato e fornito dall'utente stesso.
Nello specifico, all'utente sono richieste informazioni inerenti al film da lui apprezzato che vengono sfruttate per individuare il cluster più simile e, in questo modo, è possibile ricavare una lista di film consigliabili all' utente(la top 10), sulla base della similarità tra quello fornito e quelli presenti nel cluster risultato più simile.

In particolare, a seguito della clusterizzazione, per calcolare le similarità si è utilizzata la libreria sopra-citata FuzzyWuzzy che utilizza come metrica la distanza di Levenshtein, ossia una metrica in grado di misurare la differenza tra due sequenze di caratteri basandosi sul numero minimo di modifiche necessarie di un singolo carattere per trasformare la parola con quella con cui viene confrontata.

Di seguito è riportato un esempio di utilizzo del Recommender system:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 1
INIZIAMO!

Inserire il nome del film o serie TV che hai apprezzato: spider-man
spider-man è un film? (s/n)
 -> s
Inserire il paese di produzione:
 -> united states
Inserire l`anno di rilascio:
 -> 2002
Inserire un membro del cast:
 -> tobey maguire
Inserire un voto da 1 a 10 sul film/serie TV:
 -> 9
Inserisci il genere, scegliendo tra questi:
1 action
2 anime
3 commedies
4 cult
5 documentary
6 dramas
7 fantasy
8 horror
9 kids
10 musical
11 nature
12 romantic
13 sport
14 stand-up
15 thrillers
 ->7
```

```
Ti consigliamo di guardare:

spider-man 3
the time machine
ghost rider
underworld
the matrix revolutions
the matrix reloaded
spectral
tremors 3: back to perfection
superman returns
the core
```

Infine, riportiamo un esempio di esecuzione completa del programma:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
 1. Lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
 2. Scopri il genere di un film o serie TV
 3. Interroga il sistema
 4. Esci
 --> 1
INIZIAMO!

Inserire il nome del film o serie TV che hai apprezzato: countdown
countdown è un film? (s/n)
 -> s
Inserire il paese di produzione:
 -> united states
Inserire l`anno di rilascio:
 -> 2019
Inserire un membro del cast:
 -> Elizabeth Lail
Inserire un voto da 1 a 10 sul film/serie TV:
 -> 8
Inserisci il genere, scegliendo tra questi:
1 action
2 anime
3 commedies
4 cult
5 documentary
6 dramas
7 fantasy
8 horror
9 kids
10 musical
11 nature
12 romantic
13 sport
14 stand-up
15 thrillers
 ->8
```

```
Ti consigliamo di guardare:

deadcon
tales from the hood 2
holidays
haunting on fraternity row
cult of chucky
final destination 3
hubie halloween
welcome to willits
house of the witch
antidote
```

```
Per dettagli sulle raccomandazioni restituite, digitare kb:
kb
Il cluster di appartenenza è il valore di choice: 2
Le metriche restituite tra tutti i cluster sono le seguenti: [88109, 1112880, 255272]

Le singole metriche di similarità restituite per il cluster 2 sono:
        type                      title  similarity
1409  movie                    deadcon         398
5124  movie       tales from the hood 2         376
2375  movie                   holidays         369
2268  movie  haunting on fraternity row         367
1319  movie              cult of chucky         366
1846  movie         final destination 3         366
2439  movie             hubie halloween         364
6456  movie          welcome to willits         363
2422  movie          house of the witch         362
484   movie                    antidote         362
```