



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

1° LUGLIO 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.
- **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 (Netflix_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 corrispondenza 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. Interroga il sistema
  3. Esci
--> 2
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. Interroga il sistema
  3. Esci
--> 2
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. Interroga il sistema
3. Esci
--> 2
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. Interroga il sistema
3. Esci
--> 2
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 base 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. 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.

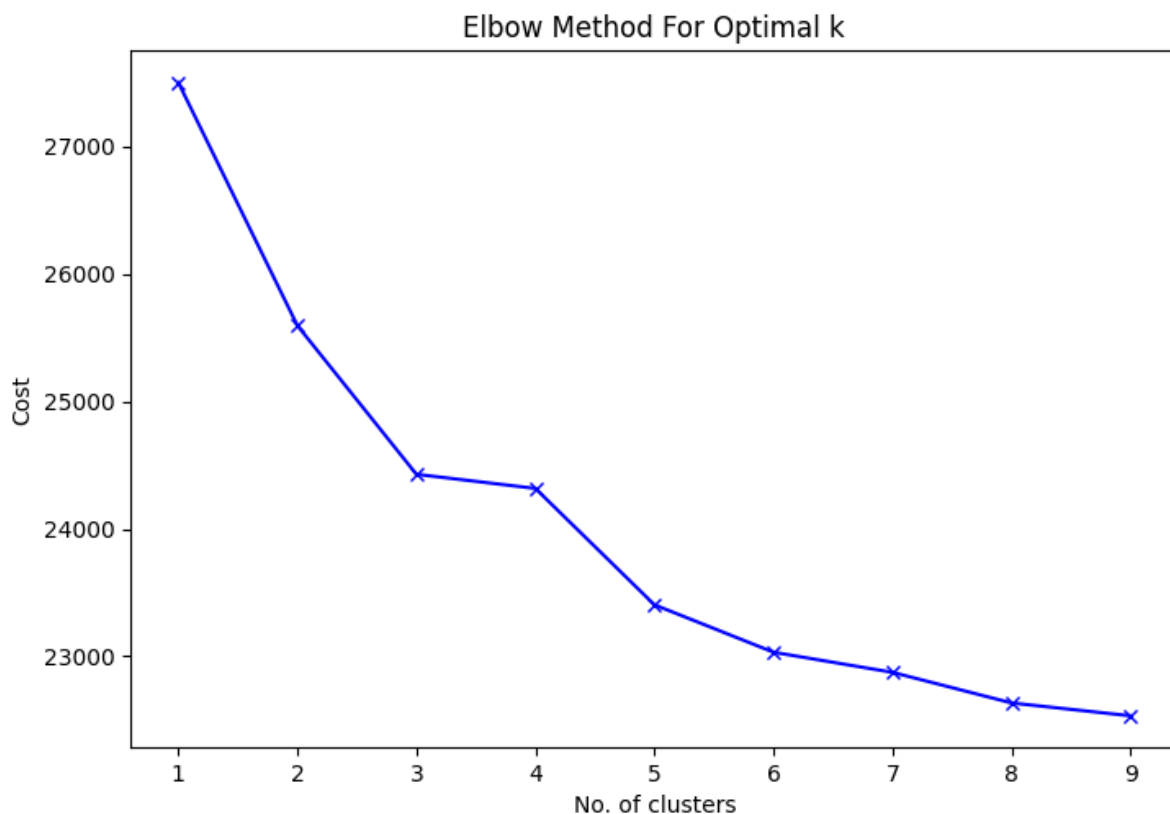
a. CLUSTER

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.



b. 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. Interroga il sistema
  3. 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 comedies
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
the matrix revolutions
the matrix reloaded
tremors 3: back to perfection
spectral
superman returns
skyline
terminator 3: rise of the machines
```

Infine, riportiamo un esempio di esecuzione completa del programma in cui viene utilizzato sia il Recommender system che la Base di conoscenza:

```
Benvenuto in UnibaVision!
Scegli come proseguire:
  1. lasciati suggerire un nuovo film sulla base di un altro che hai apprezzato
  2. Interroga il sistema
  3. 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 comedies
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
hubie halloween
welcome to willits
antidote
house of the witch
thriller
```

Per dettagli sulle raccomandazioni restituite, digitare kb:

kb

Il cluster di appartenenza è il valore di choice: 3

Le metriche restituite tra tutti i cluster sono le seguenti: [507639, 413631, 534991]

Le singole metriche di similarità restituite per il cluster 3 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