Descrizione dell'agente

Lo scopo del progetto è quello di creare un agente capace di:

-Raccomandare dei film sulla base di precedenti film visualizzati, in modo da

consigliare un contenuto possibilmente interessante

------------------------------------

Specifiche PEAS:

-Performance:

La misura di performance dell'agente è la sua capacità di avvicinarsi quanto piu possibile

al mostrare i film piu simili rispetto a quelli gia visti.

Enviroment:

L'ambiente in cui opera l'agente è costituito dal dataset e il film visualizzato

dall'utente:

- Statico, in quanto l'ambiente non varia durante la elabrazione dell'utente

- Episodico, perchè non ci sono influenze dai precedenti "episodi" di elaborazione dell'agente

- Completamente osservabile, in quanto si ha accesso a tutti i film consigliabili e quello selezionato.

- Deterministico o stocastico?

- Discreto

- Singolo agente

Attuatori:

L'agente ha un attuatore che consiste nel fornire una lista di film relativa a quello visualizzato

dall'utente.

Sensori:

L'agente ha un sensore che consiste nel raccogliere il film visualizzato dall'utente sul quale

basare le sue azioni.

----------------------------------------

2. Analisi del problema

Inizialmente abbiamo approcciato al problema pensando che un classificatore fosse il miglior algoritmo

da utilizzare. Dopo aver iniziato a sviluppare l'algoritmo abbiamo cambiato rotta in quanto non disponevamo

di profili utente, né di un sistema dotato di interfaccia grafica, successivamente abbiamo adottato un algoritmo

di clustering che ci è sembrato piu adatto in relazione alle nostre risorse.

- Approfondire scelta del classificatore e problemi incontrati

- Approfondire scelta clustering e perchè funziona meglio nel nostro caso

------------------------------------------------------------

3.1 Raccolta, analisi e preprocessing dei dati

Scelta del dataset:

Per quanto riguarda la scelta del dataset del modello di machine learning abbiamo optato per la selezione

di un dataset gia fornito da kaggle. Per quanto riguarda la nostra idea di modulo IA c'erano 3 diversi dataset

idonei, abbiamo selezionato quello piu consistente.

Non abbiamo valutato di creare un dataset da 0 in quanto troppo dispendioso per il nostro budget, avendo anche trovato

degli ottimi dataset online.

------------------------------------------------------------

3.2 Analisi e scrematura del dataset

Il dataset è disponibile al seguente link (kaggle.com / imdb) e contiene 1000 voci tra film e show televisivi

selezionati da IMDB come classifica dei "top 1000". È formato da 16 colonne, relative ai film, trame, generi, attori, votazioni etc.

Inizialmente abbiamo trovato vari dataset possibilmente adatti al nostro agente, circa 3, quello di imdb è stato selezionato valutando

la qualità e usabilità dei dati.

Dimensione: 438 Kb

Usability: 10.0 (kaggle)

Data governance: CC0: Public Domain

ALGORITMO DI CLASSIFICAZIONE

Il primo approccio è stato quello di cercare una correlazione tra le varie colonne.

La scelta è stata di cercare una correlazione tra le colonne di voto di critica e

pubblico. Essondoci una scarsa correlazione tra voto di critica e pubblico abbiamo escluso inizialmente

Un possibile utilizzo di algoritmo di clustering, utilizzando uno scatterplot per rappresentare la distribuzione delle due colonne su un piano cartesiano.

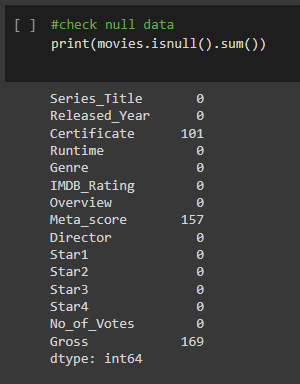
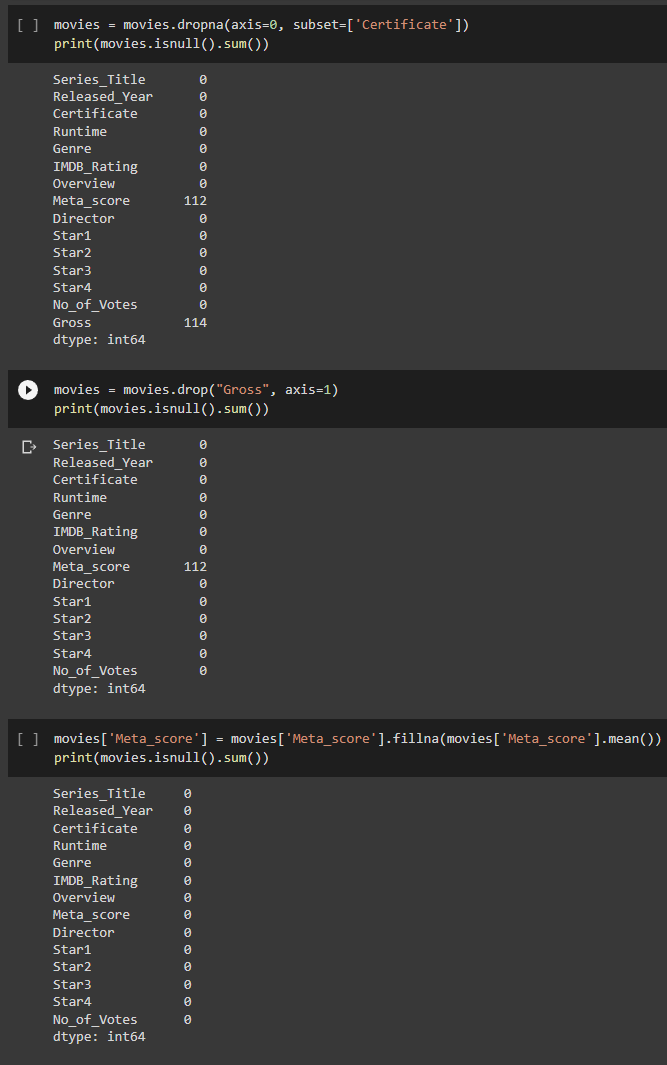
Ci siamo dedicati successivamente alla data preparation:

-Data cleaning:

La pulizia dei dati è stata eseguita per eliminare i valori nulli contenuti in alcune colonne del dataset, utilizzando un due semplici soluzioni, quali l’eliminazione delle righe e delle colonne.

Abbiamo eliminato la colonna “Gross” (Budget del film) e “Poster\_Link” (URL per copertina film), perché non ci risultava utile nell’algoritmo prefissato e eliminato le righe “Certificate” (classificazione film per categorie di età) in quanto non avrebbe compromesso il dataset essendo solo un 10% di valori mancanti.

Rimangono valori mancanti solo nella colonna Metascore, in questo caso abbiamo riempito i valori mancanti facendo una media dei voti di tutta la colonna.

-Feature scaling:

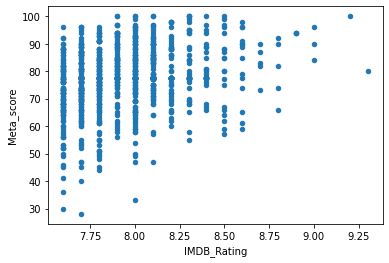
Per favorire l’individuazione di una correlazione tra le colonne interessate, Meta\_score e IMDB\_Rating rispettivamente pubblico e critica, abbiamo “scalato” i valori di Meta\_score per renderli piu adatti a un confronto.

-Feature Selection:

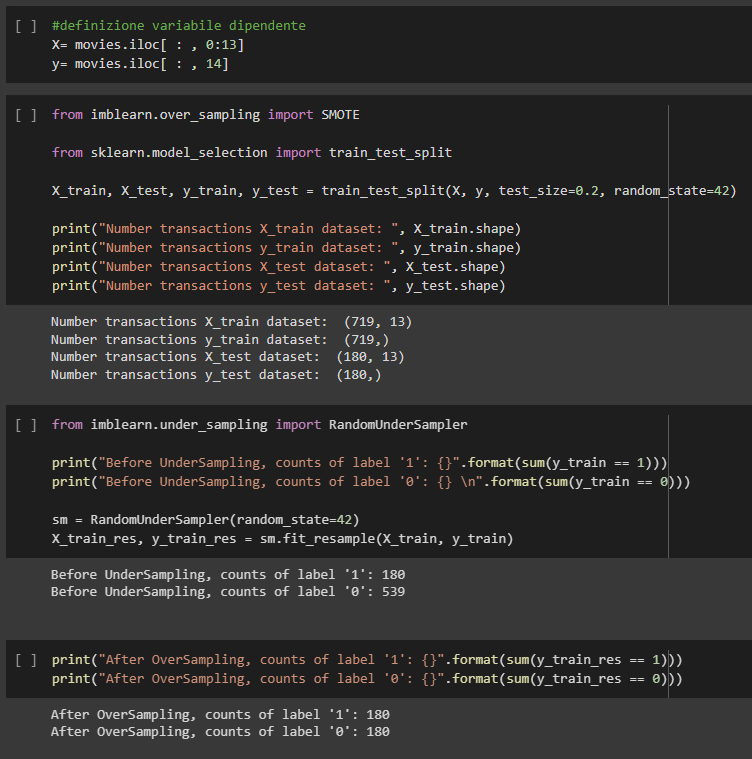
Quindi abbiamo utlizzato queste colonne di rating come riferimento per estrarre dati che ci aiutassero nella classificazione di un possibile film idoneo.

-Data Balancing:

La variabile target è stata riempita provvisoriamente riempita con dati fittizzi, proponendoci di risolvere la questione in seguito. In quanto già abbastanza bilanciato, abbiamo effettuato







Nel corso dello sviluppo abbiamo incontrato diverse problematiche.

-Algoritmo selezionato Naive Bayes

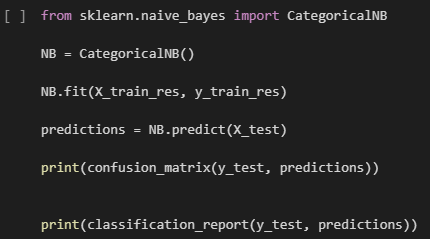
-Vedere slide

-Valutare algoritmo clustering

Successivimente abbiamo deciso di utilizzare un classificatore con classificazione probabilista, adottando

Naive Bayes con le colonne relative alla votazione del pubblico e della critica.

Naive Bayes ha dato errore.  
Abbiamo convenuto che l’errore ottenuto fosse dovuto alla presenza di variabili “string”, di conseguenza abbiamo pensato di convertire le stringhe in variabili int categoriche.  
Durante la risoluzione del bug ci siamo convinti che il tipo di output da noi desiderato si sposasse meglio con un algoritmo di clustering, abbiamo dunque virato verso questo tipo di algoritmo di machine learning.



-Algoritmo

ALGORITMO DI CLUSTERING

Il risultato non era soddisfacente, così abbiamo provato a ritornare su un algoritmo di clustering scegliendo

stavolta tutti i dati in stringhe.

# Abbiamo pensato di convertire le stringhe in variabili int categoriche, per poi convenvenire che il problema che ci stiamo ponendo non sia di classificazione ma di clustering.

