Relazione Caso di Studio
Ingegneria Della Conoscenza
Anime Recommender

Alessandro Ciafardini mat. 650816 a.ciafardini@studenti.uniba.it ,

Regina Grimaldi mat. 676934 r.grimaldi8@studenti.uniba.it ,

Giuseppe Forziati mat. 675477 g.forziati8@studenti.uniba.it

Repository GitHub:

16 Settembre 2021

Contenuti

1 Introduzione

Per la realizzazione di questo progetto ci si è ispirati ad un paper (Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor – Ahuja, Solanki, Nayyar) e constatato l'alto e sempre più crescente interesse del pubblico per gli anime si è deciso di optare per un progetto che riguardasse l'analisi di un dataset acquisito dal sito Kaggle: quest'ultimo è stato ottenuto tramite le API fornite dal sito MyAnimeList.net. Come ultimo fine del progetto abbiamo deciso di realizzare un Recommender System per incentivare gli utenti ad espandere la loro conoscenza di questo genere di intrattenimento.

1.1 Strumenti

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

1.2 Librerie

- Sklearn
  Scikit-learn (ex scikits.learn) è una libreria open source di apprendimento automatico per il linguaggio di programmazione Python. Contiene algoritmi di classificazione, regressione e clustering (raggruppamento) e macchine a vettori di supporto, regressione logistica, classificatore bayesiano, k-mean e DBSCAN, ed è progettato per operare con le librerie NumPy e SciPy.

- Pandas
  Nella programmazione per computer, Pandas è una libreria software scritta per il linguaggio di programmazione Python per la manipolazione e l'analisi dei dati. In particolare, offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali.

- MathPlotLib
  Matplotlib è una libreria per la creazione di grafici per il linguaggio di programmazione Python e la libreria matematica NumPy. Fornisce API orientate agli oggetti che permettono di inserire grafici all'interno di applicativi usando toolkit GUI generici, come WxPython, Qt o GTK.

2 Preprocessing

Il dataset direttamente acquisito da Kaggle (anime.csv) presentava dati mancanti (missing values e unknown values) e colonne non pertinenti allo scopo di questo progetto. Per questo motivo ci si è avvalsi della fase di Preprocessing per adattare meglio il dataset ai nostri fini. In particolare:

- È stato effettuato un processo di feature selection in questo caso eliminando alcune colonne, tali "English Name", "Rating", "Japanese Name", "Aired", "Premiered", "Licensors", "Ranked", "Popularity", "Members", "Favorites", "Watching", "Completed", "On-Hold", "Dropped", "Plan to Watch", "Score-10", "Score-9", "Score-8", "Score-7", "Score-6", "Score-5", "Score-4", "Score-3", "Score-2", "Score-1".
- Sono stati rimosse le missing values e i valori unknown eccetto per la colonna "Score".
- I valori unknown della colonna "Score" sono stati sostituiti con un valore neutro (5) per evitare di sbilanciare il dataset.
- Riduzione dei "Generi" ad un unico valore per riga
- Riduzione dei "Producers" ad un unico valore per riga.
- Riduzione degli "Studios" ad un unico valore per riga.
- È stato effettuato un lavoro di standardizzazione della colonna "Duration" per ottenere unicamente i minuti, in particolare laddove fossero state presenti ore si è convertita la quantità di ore in minuti e successivamente sommato il risultato ai minuti eventualmente già presenti.
- Per preparare il dataset alla classificazione si sono convertite le colonne con valori categorici in variabili numeriche

Il dataset così formattato si presenta in questo modo:

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Score	Episodes	Duration_format	Rating_format	Genre_format	Producers_format	Studios_format	Name_format	Type_format	Source_format
8.78	26	24	0	0	0	206	0	0	0
8.39	1	115	0	1	0	0	1	1	0
8.24	26	24	1	0	1	182	2	0	1
7.27	26	25	1	1	2	206	3	0	0
6.98	52	23	1	2	2	299	4	0	1
7.95	145	23	1	0	2	126	5	0	1
8.06	24	23	1	0	3	1	6	0	1
8.15	24	27	1	1	4	2	7	0	1
8.76	74	24	2	3	5	182	8	0	1
7.91	220	23	1	0	6	3	9	0	1
7.9	178	22	1	0	7	259	10	0	1
7.94	26	23	1	0	2	122	11	0	1
7.42	24	24	0	0	8	4	12	0	1
7.76	22	23	2	1	3	182	13	0	0
7.95	69	24	1	0	6	206	14	0	1
7.51	26	24	1	1	9	5	15	0	1
8.32	26	24	1	1	2	171	16	0	0
7.45	1	104	0	3	10	15	17	1	0

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 insime di test fissando quest'ultimo al 30%. La variabile target sulla quale effettuare la predizione sarà il genere.

Per ottenere il risultato migliore sono stati messi a confronto tre modelli di classificatori:

- KNN
- Random Forest
- Gaussian Naive Bayes

3.1 Classificatori

3.1.1 KNN

Uno degli algoritmi più conosciuti nel machine learning è il K-Nearest Neighbors (KNN) che, oltre alla sua semplicità, produce buoni risultati in un gran numero di domini. È un algoritmo di apprendimento supervisionato, il cui scopo è quello di predire una nuova istanza conoscendo i data points che sono separati in diverse classi.

Un oggetto è classificato in base alla maggioranza dei voti dei suoi k vicini. k è un intero positivo tipicamente non molto grande. Se k=1 allora l'oggetto viene assegnato alla classe del suo vicino.

Il suo funzionamento si basa sulla somiglianza delle caratteristiche, nel nostro caso viene calcolata la similarità dell'anime inserito dall'utente con gli altri anime presenti nel dataset. In questo modo si ottiene la variabile target da predire che è il genere.

3.1.2 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 aggragazione di tipo begging o bootstrap, il begging è un meta-algoritmo ensemble che migliora l'accuratezza degli algoritmi di Machine Learning.

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.

3.1.3 Gaussian Naive Bayes

L'algoritmo Naive Bayes è impiegato per problemi di classificazione binaria e multiclasse. E' così chiamato perchè il calcolo delle probabilità per ogni ipotesi è semplificato per rendere il problema trattabile.

Il Gaussian Naive Bayes è un estensione del Naive Bayes applicata ad attributi con valori reali asumendo solitamente un distribuzione gaussiana.

Si possono usare diverse funzioni per stimare la distribuzione dei dati ma la gaussiana risulta essere la più semplice poichè permette di stimare la media e la deviazione standard dai dati di training.

3.2 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:

- Bestparams KNN metric:manhattan, neighbor = 10, weights = distance
- Bestparams GAU var-smoothing = 0.0005
- Bestparams RF max-depth = 25, min-samples-leaf = 1, min-samples-split = 3, n-estimators = 1300, criterion = entropy

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

	precision	recall	f1-score	support
0	0.99	1.00	0.99	822
2	0.75	0.95	0.84	19
3	0.86	0.60	0.71	10
4	0.97	0.88	0.92	32
5	0.65	0.72	0.68	18
7	0.79	0.85	0.82	27
8	0.98	0.96	0.97	128
9	0.84	0.97	0.90	37
10	0.94	0.97	0.96	115
11	0.40	0.67	0.50	3
13	0.97	0.97	0.97	71
14	0.00	0.00	0.00	4
15	1.00	0.50	0.67	2
16	0.00	0.00	0.00	6
17	0.59	0.81	0.68	16
18	0.99	1.00	0.99	431
19	0.95	0.71	0.82	28
20	0.00	0.00	0.00	1
21	0.33	0.25	0.29	4
24	0.00	0.00	0.00	1
25	0.00	0.00	0.00	3
26	1.00	0.33	0.50	9
27	1.00	0.33	0.50	6
28	0.89	1.00	0.94	58
29	0.00	0.00	0.00	1
accuracy			0.96	1852
macro avg	0.64	0.58	0.59	1852
weighted avg	0.96	0.96	0.96	1852

Figura 1: KNN

	precision	recall	f1-score	support
0	1.00	1.00	1.00	822
2	0.76	0.84	0.80	19
3	0.64	0.90	0.75	10
4	1.00	0.97	0.98	32
5	0.68	0.83	0.75	18
7	0.87	1.00	0.93	27
8	0.99	1.00	1.00	128
9	0.90	0.97	0.94	37
10	0.98	0.99	0.99	115
11	0.25	0.33	0.29	3
13	0.96	1.00	0.98	71
14	0.00	0.00	0.00	4
15	1.00	0.50	0.67	2
16	0.00	0.00	0.00	6
17	0.72	0.81	0.76	16
18	1.00	1.00	1.00	431
19	1.00	0.93	0.96	28
20	0.00	0.00	0.00	1
21	0.50	0.25	0.33	4
23	0.00	0.00	0.00	0
24	0.00	0.00	0.00	1
25	0.00	0.00	0.00	3
26	1.00	0.33	0.50	9
27	1.00	0.67	0.80	6
28	0.97	1.00	0.98	58
29	0.00	0.00	0.00	1
accuracy			0.98	1852
macro avg	0.62	0.59	0.59	1852
weighted avg	0.97	0.98	0.97	1852

Figura 2: RF

Accuracy gau : 0.9422246220302376

	precision	recall	f1-score	support
0	0.99	0.98	0.99	822
2	0.67	0.21	0.32	19
3	0.71	0.50	0.59	10
4	1.00	0.81	0.90	32
5	0.62	0.89	0.73	18
7	0.77	1.00	0.87	27
8	0.94	0.93	0.93	128
9	0.92	0.89	0.90	37
10	0.94	0.96	0.95	115
11	0.29	0.67	0.40	3
13	0.99	0.97	0.98	71
14	0.00	0.00	0.00	4
15	1.00	0.50	0.67	2
16	0.09	0.17	0.12	6
17	0.67	0.88	0.76	16
18	0.99	0.97	0.98	431
19	0.70	0.93	0.80	28
20	0.00	0.00	0.00	1
21	0.00	0.00	0.00	4
23	0.00	0.00	0.00	0
24	0.00	0.00	0.00	1
25	0.00	0.00	0.00	3
26	0.57	0.44	0.50	9
27	0.83	0.83	0.83	6
28	0.88	0.97	0.92	58
29	0.00	0.00	0.00	1
accuracy			0.94	1852
macro avg	0.56	0.56	0.54	1852
weighted avg	0.94	0.94	0.94	1852

Figura 3: GAU

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

Qui di seguito si riporta un esempio di funzionamento del classificatore:

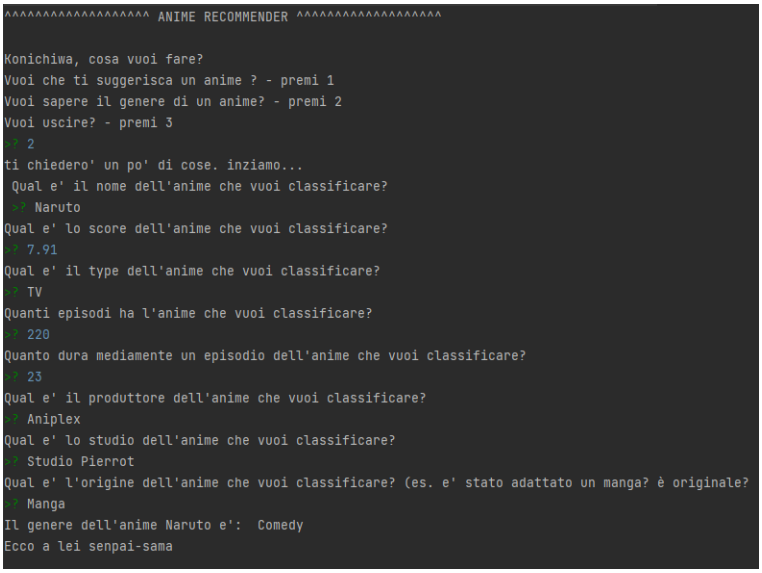

Figura 4: Esempio Classificatore

4 Clustering

Il clustering consiste in un insieme di metodi per raggruppare oggetti in classi omogenee. Un cluster è un insieme di oggetti che presentano tra loro delle similarità, ma che, per contro, presentano dissimilarità con oggetti in altri cluster. L'input di un algoritmo di clustering è costituito da un campione di elementi, mentre l'output è dato da un certo numero di cluster in cui gli elementi del campione sono suddivisi in base a una misura di similarità.

4.1 K-Means

Nel nostro progetto abbiamo applicato l'algoritmo K-Means.

Quest'ultimo ha lo scopo di suddividere un insieme di oggetti in k gruppi sulla base dei loro attributi.
L'algoritmo segue una procedura iterativa: inizialmente crea k partizioni e assegna i punti d'ingresso a ogni partizione o casualmente o usando alcune informazioni euristiche; quindi calcola il centroide di ogni gruppo; costruisce in seguito una nuova partizione associando ogni punto d'ingresso al gruppo il cui centroide è più vicino ad esso; infine vengono ricalcolati i centroidi per i nuovi gruppi e così via, finché l'algoritmo non converge.

4.2 Elbow Method

Applicando l'Elbow Method, "Metodo del Gomito", abbiamo scoperto che il numero di cluster più adatto per il dataset. L'Elbow Method è stato scelto poichè è un modo totalmente oggettivo per determinare il numero di cluster.

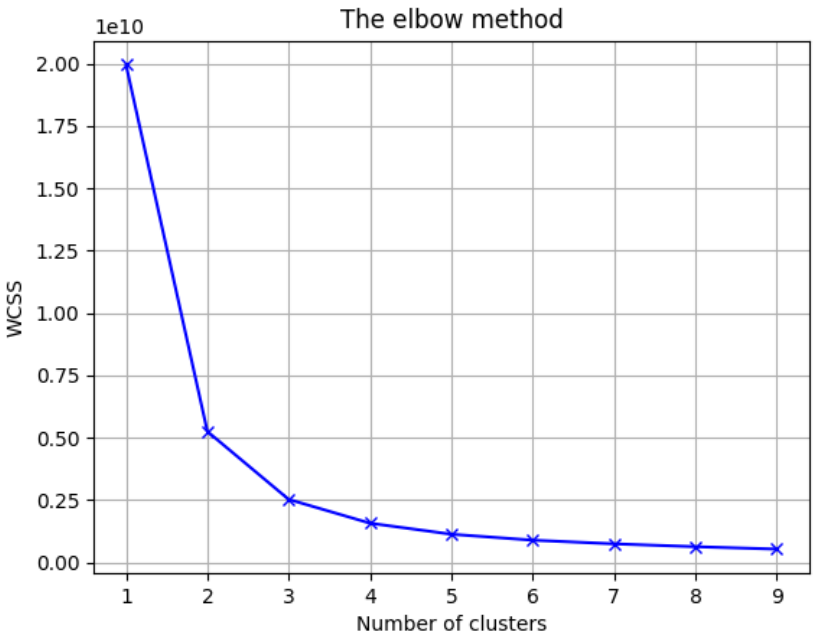

Figura 5: Metodo Del Gomito

Dal grafico si evince come i punti migliori ove effettuare il K-Means fossero 2 e 3. Abbiamo scelto di utilizzare la divisione in 3 cluster per avere una scelta più etereogenea.
L'utilizzo del clustering assieme al "classificatore" ci ha permesso di ottenere il risultato migliore nel Recommender che vedremo di seguito.

5 Recommender System

L'idea di creare un Recommender System per suggerire il prossimo anime da guardare proviene da un'analisi approfondita del paper di cui sopra.
Una volta effettuato il cluster sul dataset si è proceduto ad applicare la raccomandazione basata sula similarità degli anime.

In particolare si è scelto di utilizzare la similarità del coseno, metrica attraverso la quale è misurato il coseno dell'angolo tra due vettori proiettati in uno spazio multi-dimensionale. Più piccolo è l'angolo più alta sarà la similarità. Questo recommender si basa sulle risposte ottenute in seguito ad una serie di domande ove viene richiesto all'utente di inserire informazioni necessarie alla raccomandazione.

Di seguito riportiamo le domande iniziali del recommender:

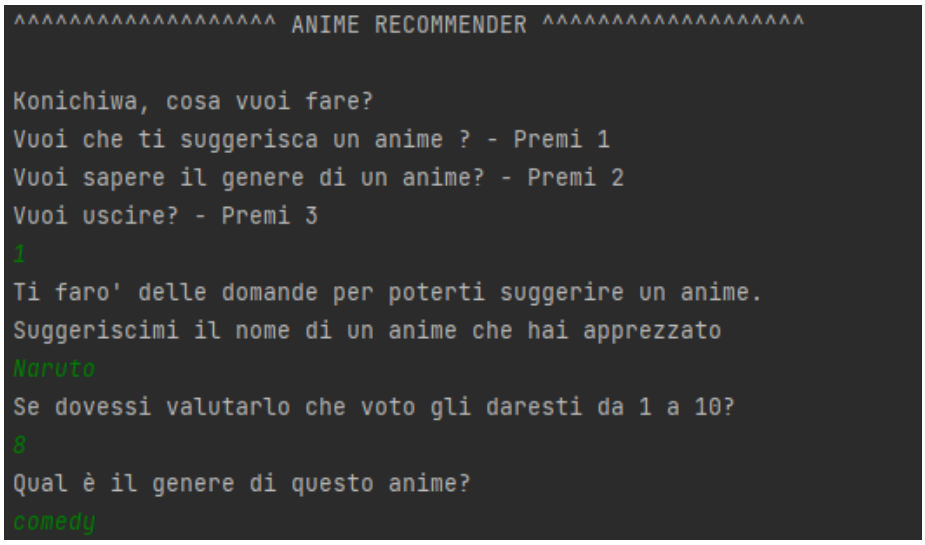

Figura 6: Start Recommender

Alcune domande potrebbero presentare delle difficoltà di comprensione per l'utente. Per questo motivo si è scelto di provvedere ad una opzione che spiegasse il significato della domanda posta:

```
Sai dirmi il tipo di questo anime? (Si o No)(se non sai cos'è il tipo digita 1)
1
Quando parliamo di tipo intendiamo se l'anime appartiene a una di queste categorie:
- TV      : l'anime viene trasmesso ad episodi
- Movie   : l'anime è un film
- ONA/OVA : l'anime è una puntata speciale di un anime o originale

Ora sai dirmi qual è il tipo? (Si o No)(se non sai cos'è il tipo digita 1)
si
Qual è il tipo?
tv
```

Figura 7: Spiegazione Domande

In altri casi l'utente potrebbe non sapere la risposta ad alcune domande. In questo caso il sistema semplicemente non acquisirà l'informazione per la query.

```
Sai dirmi da quanti episodi è formato? (Si o No)
no
Sai dirmi quanto dura mediamente un episodio?
no
```

Figura 8: Query senza informazioni

Infine il recommender produrrà in output una lista di dieci anime suggeriti sulla base delle preferenze espresse dall'utente.

Di seguito un esempio di risultato:

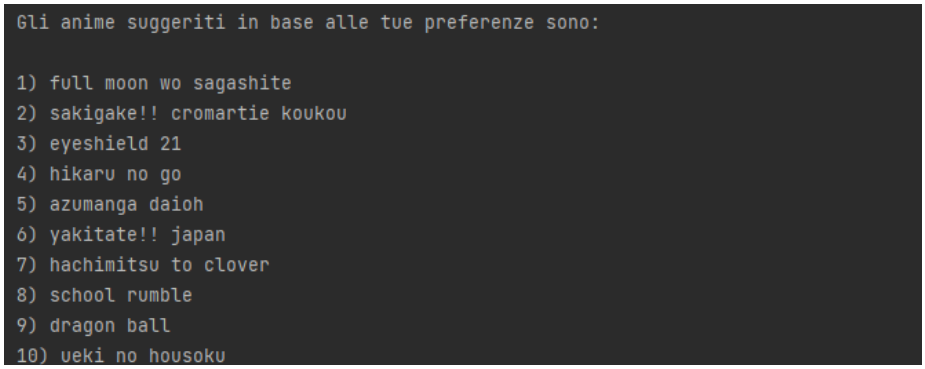

Figura 9: Risultato

Il recommender non si ferma dopo la predizione ma permette all'utente di scegliere una nuova opzione oppure di uscire.

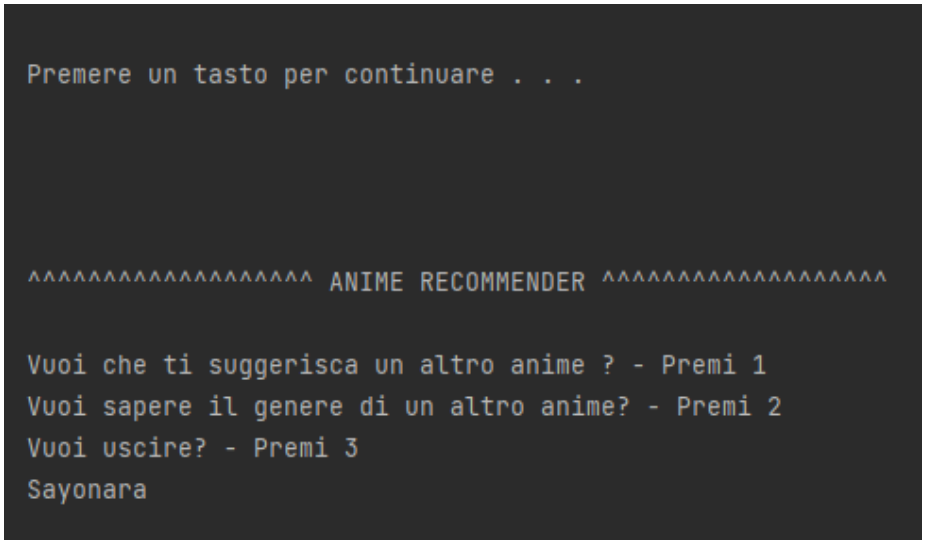

Figura 10: Menù

NB: Nel caso in cui un genere abbia pochi anime presenti nel dataset, il recommender suggerirà anime con generi simili.