DODO.NET

INDICE

Alfonso Cuomo

Gioacchino Saraceno

Grazia Margarella

Simone Farina

INTRODUZIONE

COSA VOGLIAMO OTTENERE

Ciò che intendiamo creare è un sistema di suggerimenti per gli utenti di un e-commerce che vende libri, in modo tale da consigliare i libri che più si addicono al gusto dell'utente attraverso un modello di intelligenza artificiale in grado di apprendere le preferenze di un utente tramite le interazioni con i prodotti.

PROBLEMA DA RISOLVERE

Il nostro problema corrisponde al come determinare l'insieme di libri da consigliare all'utente, sfruttando le informazioni sui libri e gli acquisti dell'utente.

Un esempio: Se un utente è solito acquistare i libri di una determinata categoria, di un determinato autore o di un determinato periodo, l'algoritmo dovrà suggerirgli i libri inerenti alle caratteristiche sopra citate.

LINK REPOSITORY GITHUB

https://github.com/graziamargarella/Progetto_FIA_Dodo.net

L'AGENTE

SPECIFICA PEAS

Dal punto di vista dell'algoritmo, l'ambiente comprenderà i dati relativi all'e-commerce.

Ciò che dovrà restituire l'agente all'utente saranno i libri consigliati sul terminale.

In seguito, verrà mostrata una tabella esplicativa della specifica PEAS dell'agente:

	PEAS
PERFORMANCE	Percentuale di consigli pertinenti
AMBIENTE	E-commerce
ATTUATORI	Visualizzazione dei libri consigliati
SENSORI	Collegamento al database

PROPRIETÀ DELL'AMBIENTE

In seguito, viene mostrata una tabella che specifica quali sono le proprietà dell'ambiente in cui l'agente opera:

	AMBIENTE
COMPLETAMENTE OSSERVABILE	I sensori possono accedere in qualsiasi momento a tutte le informazioni dell'ambiente
NON DETERMINISTICO	Lo stato successivo dell'ambiente non è determinato solamente dallo stato corrente e dall'azione dell'agente
SEQUENZIALE	Ogni azione eseguita dall'agente avrà un impatto sulle azioni successive ad essa
DINAMICO	I libri e gli utenti presenti nel bookstore online saranno soggetti a continui cambiamenti
DISCRETO	L'agente viene chiamato solo un numero determinato di volte
AGENTE SINGOLO	Nel sistema opererà un singolo agente.

SCELTE PROGETTUALI

SCELTA DELL'LINGUAGGIO

Il codice utilizzato per le varie fasi dello sviluppo è in linguaggio Python e si basa su alcune librerie come panda e numpy per l'analisi e la manipolazione dei dati, matplotlib e seaborn per la visualizzazione dei dati, scikit-learn per l'implementazione degli algoritmi di apprendimento.

DATASET UTILIZZATO

LIBRI

Dal momento che gli algoritmi di apprendimento si basano sui dati, questi sono stati il primo focus nella fase di sviluppo.

Inizialmente si sono testati due approcci:

- utilizzare dei dataset di grandi dimensioni reperiti online;
- utilizzare un dataset ridotto prodotto appositamente per l'e-commerce.

Nel primo caso, il vantaggio immediato della grande mole di dati è stato messo in dubbio dopo un'analisi della qualità del dataset. I valori nulli erano molti, tanto che eliminando i datapoint non completi si era ridotto il dataset del 10%.

Inoltre, altro punto critico, era quello della suddivisione dei libri in categorie non rilevanti. La maggior parte delle categorie individuate avevano una cardinalità compresa tra 1 e 5 libri. Su un dataset di 6000 libri è evidente quanto un'informazione come la categoria non sia affatto utilizzabile.

Per ovviare a problemi del genere si è optato per la seconda scelta. Il dataset utilizzato era inizialmente composto da:

- ISBN (intero)
- Titolo (stringa)
- Autore (stringa)
- Editore (stringa)
- Prezzo (float)
- Anno di pubblicazione (intero)
- Categoria (stringa)
- Basato su storia vera (booleano)
- Edizione illustrata (booleano)
- Appartenente a una saga (booleano)

- Numero pagine (intero)

In un primo script Python, creazione_dataset.py, dopo aver importato il file csv ottenuto da una query SQL dal database del sistema, si è scelto di eliminare le colonne non rilevanti, quali titolo e descrizione. In seguito, sono stati sostituiti i dati testuali in numeri interi, processabili successivamente.

Per aggiungere altre informazioni sono stati inoltre generati i valori:

- del numero di recensioni positive ricevute nel sistema (intero),
- del numero di acquisti avvenuti nel sistema (intero),

Il tutto esportato in un altro file csv.

UTENTI

Per quanto riguarda i dati dell'utente, dallo stesso database del sistema è stata eseguita una query che restituiva gli ISBN dei libri con cui ha interagito l'utente. Nel nostro caso abbiamo simulato il funzionamento dell'algoritmo su quattro utenti diversi. La cardinalità di questi insiemi varia tra i 10 e 20 libri.

I dati dei singoli utenti sono contenuti nei file del tipo *utente_X.txt* dove X corrisponde all' utente (X compreso tra 1 e 4 inclusi).

Nello script *creazione_dataset.py* si è inserita una colonna al dataset precedentemente ottenuto contenente un booleano, vero se l'utente inserito in input ha interagito con un determinato titolo, falso altrimenti.

I risultati sono contenuti nei file *dataset_utente_X.txt* (dove X indica l'utente compreso tra 1 e 4 inclusi).

Nello stesso script si genera inoltre il file *ISBN_index.txt* che associa ogni ISBN all'indice del dataset finale, e ci sarà utile in seguito per ricostruire le informazioni di titolo e autore.

SCELTA DELL'ALGORITMO E DESCRIZIONE

Per quanto riguarda la scelta dell'algoritmo di apprendimento sono stati presi in considerazione sia gli algoritmi di apprendimento supervisionato, in particolare la classificazione, che gli algoritmi di apprendimento non supervisionato, in particolare il clustering.
La scelta definitiva è andata sul clustering, dal momento che la struttura dei dati sarebbe stata difficilmente gestibile da un algoritmo di classificazione per via di una mancata definizione chiara della label.

Il ***clustering*** permette dunque di raggruppare i singoli datapoint in insiemi di elementi simili in base a dei pattern ritrovati nei dati. Un'ipotesi di come consigliare, dunque, una lista di libri in output all'utente è quello di selezionare il cluster contenente il maggior numero di libri con cui l'utente ha già interagito.

L'algoritmo di base da cui si è pensato di iniziare è il ***k-means***. Per poter operare su un qualsiasi algoritmo di clustering bisogna scalare le features più grandi che possono avere più peso rispetto ad altre più piccole ma più rilevanti.

Per scalare i dati è stato utilizzato il package *sklearn.preprocessing*. Il metodo di scaling utilizzato è lo StandardScaler().
Esso è un metodo di standardizzazione delle feature che si ottiene rimuovendo la media e dividendo per la deviazione standard.

$$z = \frac{x - \mu}{\sigma}$$

Questo serve per ridistribuire i dati in una distribuzione normale standardizzata.

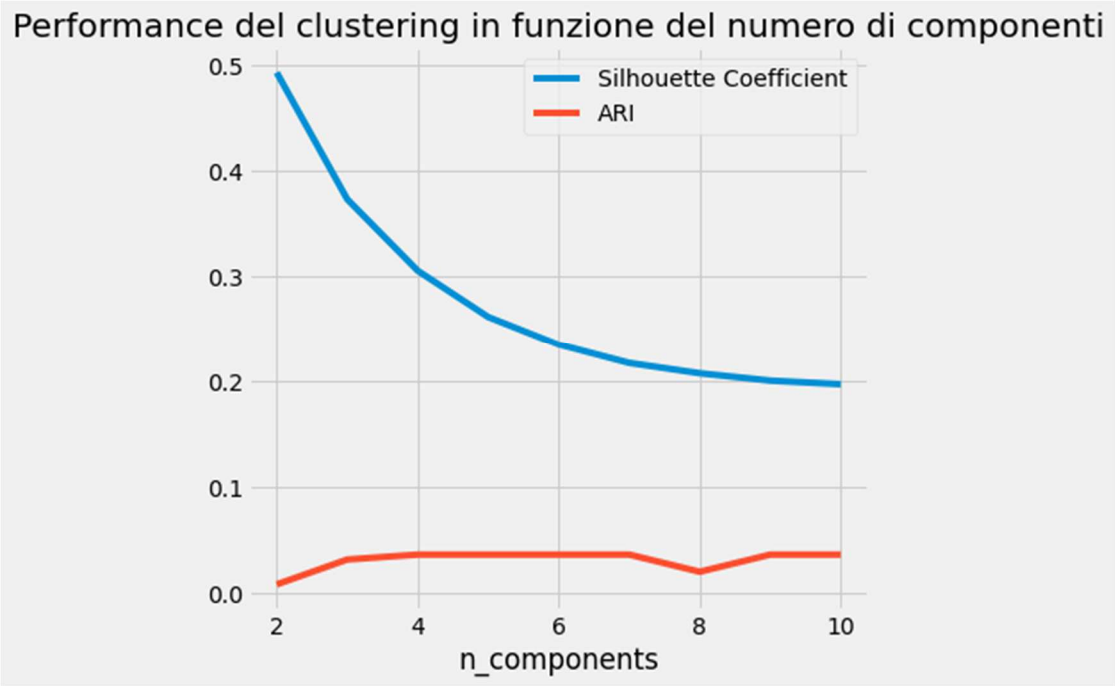
Per visualizzare dunque i risultati del cluster in un piano cartesiano si è utilizzata la PCA, ossia l'***analisi delle componenti principali***, che permette di ridurre le dimensioni del dataset, cercando un nuovo insieme di variabili con dimensione minore ma senza perdere il contenuto informativo del dataset iniziale.
L'implementazione di questo metodo è stata effettuata utilizzando il package *sklearn.decomposition*.

Il k-means, dal package *sklearn.cluster*, ha come problema la scelta del numero iniziale dei cluster. Per questo motivo sono state effettuate varie prove per trovare il k più adatto, e si è cercato di analizzare quanto una diversa inizializzazione dei centroidi potesse influire sulla scelta dei dati.

Per stimare la qualità dei cluster risultanti abbiamo utilizzato le seguenti metriche di valutazione:

- silhouette coefficient
- ARI (Adjusted Rand Index)

Il numero di cluster risultante il migliore nella maggior parte dei casi è compreso tra 2 e 3, quindi la versione finale ha k = 3.

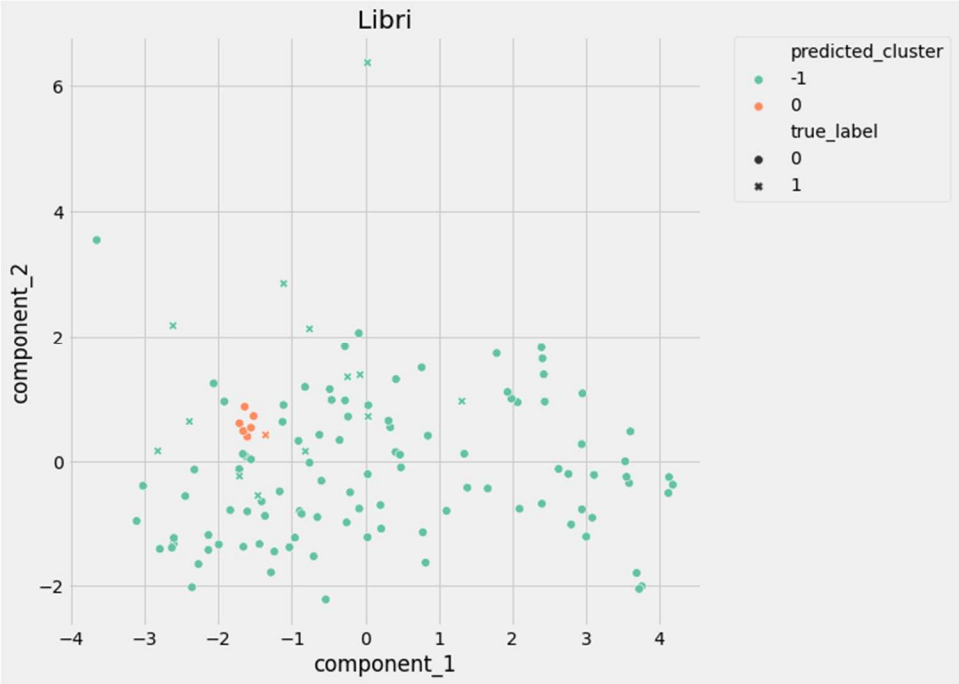


Questo tipo di grafico mostra l'andamento delle due metriche scelte precedentemente in base al numero di cluster. Il primo, che rappresenta il Silhouette Coefficient, decresce linearmente poiché dipende dalla distanza tra i datapoint. La seconda, ossia la ARI, non migliora significativamente al crescere dei cluster; ciò significa che dal primo risulta essere il k migliore il numero 2 mentre il secondo dà un piccolo miglioramento (seppur impercettibile) con il valore 3.

Come si evince anche dal grafico, la purezza dei cluster non è eccellente, dal momento che questi numeri dovrebbero tendere a 1. Purtroppo, questo è dovuto alla piccola quantità di dati su cui è stato testato l'algoritmo, il quale con altre accortezze potrebbe essere sicuramente migliore.
Il progetto, infatti, si presta molto all'espansione per via della struttura tramite pipeline e della semplicità di utilizzo degli algoritmi già implementati in sklearn.

CONFRONTO CON ALTRI ALGORITMI

Un altro algoritmo di clustering utilizzabile è il DBSCAN, ma in questo caso non risulterebbe efficiente dal momento che i dati non hanno una forma definita, come è evidente dallo screenshot seguente:



Questi sono le silhouette score e le ARI dell' algoritmo DBSCAN:

```
silhouette_score(preprocessed_data, predicted_labels_DB)
```

```
-0.11964472337335649
```

```
adjusted_rand_score(labels, predicted_labels)
```

```
0.008709306040733993
```

In questo modo, questo algoritmo dividerebbe in maniera casuale i dati senza apportare alcun raggruppamento rilevante.

IMPLEMENTAZIONE

Come è visibile nel Jupyter Notebook "model_notebook" presente nella repository, si è iniziato il processo utilizzando i dati prodotti dallo script "*creazione dataset.py*", presente nella cartella *data/script*.
Dopo di che si selezionano le feature e la label (che in questo caso indica se l'utente ha interagito o meno con il prodotto).

Definiamo dunque la pipeline di lavoro:

- Definiamo un preprocessore, composto dallo StandardScaler e la PCA, che riduce in due componenti i dati:

```
preprocessor = Pipeline(
    [
        ("scaler", StandardScaler()),
        ("pca", PCA(n_components=2, random_state=0)),
    ]
)
```

- Definiamo l'algoritmo di clustering:

```
clusterer = Pipeline(
   [
       (
           "kmeans",
           KMeans(
               n_clusters=n_clusters,
               init="random",
               n_init=50,
               max_iter=300,
               random_state=0,
           ),
       ),
   ]
)
```

  - n_clusters è una variabile che in questo caso assume valore 3;
  - il metodo di inizializzazione è 'random', dunque si sceglieranno n_clusters righe casualmente dai dati per definire i centroidi iniziali;
  - n_init è il numero di volte che l'algoritmo k-means verrà riprodotto con una scelta diversa dei centroidi. Il risultato finale scelto sarà l'output migliore in base all'inerzia dei clusters, ossia quanto possono essere considerati coerenti internamente i singoli clusters;
  - max_iter è il numero massimo di iterazioni che può eseguire l'algoritmo per una singola esecuzione.

- Dichiariamo la composizione della pipeline:

```
pipe = Pipeline(
    [
        ("preprocessor", preprocessor),
        ("clusterer", clusterer)
    ]
)
```

- Eseguiamo il metodo fit sulle features.

Una prima analisi ottenuta dal clustering è la seguente, che è composta dai dati pre-processati e le label risultanti dall'esecuzione:

```
preprocessed_data = pipe["preprocessor"].transform(features)
```

```
predicted_labels = pipe["clusterer"]["kmeans"].labels_
```

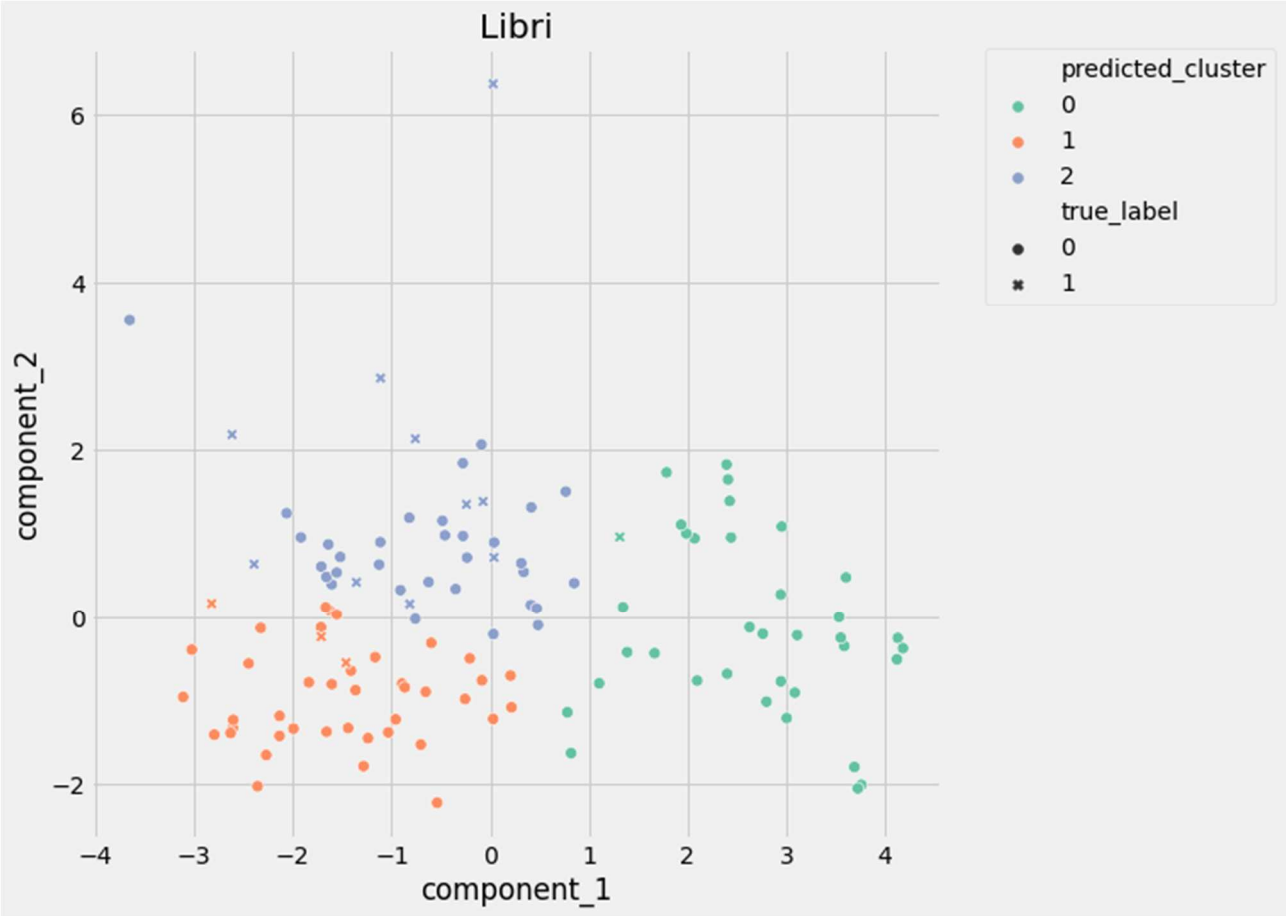
```
silhouette_score(preprocessed_data, predicted_labels)
```

```
0.3978327487551547
```

```
adjusted_rand_score(labels, predicted_labels)
```

```
0.008709306040733993
```

Per avere un'idea migliore dei dati li visualizziamo nei tre cluster e con le label originarie:



Il processo ha ancora dei dati non descritti correttamente, ma la suddivisione sembra comunque accettabile.
I titoli che andranno in output nel sistema finale saranno presi dal cluster con il maggior numero di label 1, quindi ad esempio, in questo caso, il cluster 0 in verde.

I libri con cui ha interagito l'utente selezionato sono:

	ISBN	Titolo	Autore	Editore	Prezzo	Anno	Nome_Categoria
15	9788245335872	L isola di Arturo	Elsa Morante	Einaudi	13.0	2019	Narrativa italiana
34	9788804670520	Il mondo nuovo	Aldous Huxley	Mondadori	14.0	2016	Fantascienza
43	9788804726685	Il tempo della clemenza	John Grisham	Mondadori	22.0	2020	Thriller
44	9788804727880	Illuminismo adesso	Steven Pinker	Mondadori	18.0	2020	Saggistica
66	9788820067748	Pet Sematary	Stephen King	Sperling & Kupfer	20.0	2019	Horror
69	9788822719713	La guerra dei mondi	G.W. Wells	Newton Compton Editore	4.9	2018	Fantascienza
72	9788830104716	Il Signore degli Anelli	J.R.R. Tolkien	Bompiani	50.0	2020	Fantasy
76	9788834738955	La svastica sul sole	Philip K. Dick	Fanucci Editore	16.0	2019	Fantascienza
84	9788845298752	Homo Deus	Yuval Noah Harari	Bompiani	16.0	2018	Saggistica
85	9788845932984	Essere una macchina.	Mark O Connell	Adelphi	19.0	2018	Saggistica

E i consigli sono:

27	9788804664994	Ciclo delle Fondazioni. Prima Fondazione-Fonda...	Isaac Asimov	Mondadori	16.0	2017	Fantascienza
30	9788804666905	Alexandros. La trilogia	Valerio Massimo Manfredi	Mondadori	17.0	2016	Narrativa storica
36	9788804676379	Storie della buonanotte per bambine ribelli. 1...	Favilli/Cavallo	Mondadori	20.0	2017	Ragazzi
38	9788804688846	Storie della buonanotte per bambine ribelli 2	Favilli/Cavallo	Mondadori	20.0	2018	Ragazzi
41	9788804717638	Gli invisibili	Valerio Varesi	Mondadori	16.0	2019	Thriller
42	9788804721871	La nona casa	Leigh Bardugo	Mondadori	20.0	2020	Narrativa straniera
45	9788804730422	Storie della buonanotte per bambine ribelli. 1...	Favilli	Mondadori	20.0	2020	Ragazzi
48	9788806237530	Peccato mortale. Indagine del commissario De Luca	Carlo Lucarelli	Einaudi	17.0	2018	Gialli
50	9788806242442	L inverno piu nero. Indagine del commissario D...	Carlo Lucarelli	Einaudi	18.0	2020	Gialli
57	9788811608776	A sangue freddo	Truman Capote	Rizzoli	18.0	2020	Narrativa Straniera
59	9788811671572	L estate dell innocenza	Clara Sanchez	Garzanti	15.0	2020	Rosa
61	9788811682691	Il linguaggio segreto dei fiori	Vanessa Diffenbaugh	Rizzoli	10.0	2013	Rosa
64	9788817106825	Dio di illusioni	Donna Tartt	BUR	13.0	2014	Gialli
67	9788820068288	Se scorre il sangue	Stephen King	Sperling & Kupfer	20.0	2019	Horror

CONCLUSIONI

CONSIDERAZIONI FINALI

Il progetto, in definitiva, è solamente ad uno stato iniziale che può essere migliorato in futuro con un aumento considerevole dei dati e rianalizzando le performance per avere scelte più mirate.

Inizialmente il progetto prevedeva una integrazione con il sistema di Ingegneria del Software, infatti era stato concepito per essere integrato con il sito di e-commerce Dodo.net basato su una libreria indipendente. I dati utilizzati erano e sono tutt'ora presenti nel database utilizzato a supporto del sito del progetto iniziale ma per via di alcune difficoltà tecniche non si è più proceduto all'implementazione del sistema principale, per cui abbiamo scisso in due sistemi. Il modulo di Intelligenza Artificiale è però facilmente utilizzabile dal momento che porta come risultato finale un file .csv, facilmente interpretabile anche in altri linguaggi.