SocialBook

Progetto Fondamenti di Intelligenza Artificiale

(2020/2021)

Studenti:

Barbato Alessia (0512105858)

Proietto Angelica (0512105762)

Russo Luca (0512105840)

1.Introduzione

SocialBook si propone come un social innovativo che permette la nascita di una vasta community unita dalla passione comune per i libri, dando agli utenti la possibilità di interagire tra loro.

Ogni utente, quando si registra, inserirà alcune informazioni personali di vario tipo, in particolare sui suoi interessi e preferenze letterarie, in modo da ricevere dei suggerimenti di altri utenti “simili”, ovvero utenti che appartengono allo stesso gruppo.

Il primo passo nell’analisi del problema consiste sicuramente nella definizione dell’ambiente, che si trova immediatamente al punto successivo.

2.Definizione dell’ambiente

L’ambiente viene definito tramite la rappresentazione schematica PEAS(*Performance, Environment, Actuator, Sensors*):

• **Prestazioni:** Le prestazioni dell’agente vengono valutate in base all’accuratezza con cui consiglia ad un utente altri utenti a cui potrebbe essere interessato.

• **Ambiente:** L’ambiente in cui l’agente opera è composto da utenti registrati a una piattaforma riguardante la tematica “libri”, ed è completamente osservabile, deterministico, episodico, statico e discreto. Inoltre, vi è presente un singolo agente.

• **Attuatori:**

• **Sensori:** I sensori dell’agente sono costituiti da un questionario di campionamento e da quello proposto all’utente che decide di iscriversi.

3.Raccolta dati e introduzione al pre-processing

Il dataset è la collezione dei dati (campioni), ognuno con le proprie caratteristiche, che rappresentano il dominio di interesse; viene utilizzato per addestrare e successivamente testare l’agente intelligente progettato.

Per la natura dell’agente, il cui obiettivo è quello di “apprendere” e saper suddividere gli utenti in base alla similarità tra essi, si è reso necessario effettuare una sorta di indagine per poter raccogliere i dati di partenza, in modo da garantire il funzionamento efficace del modulo che abbiamo costruito.

Per fare ciò, ci siamo serviti di un questionario sottoposto a quante più persone possibili, in modo da avere un dataset abbastanza ampio e non rischiare di mandare l’agente in overfitting.

Il nostro questionario (Socialbook (google.com)) raccoglie informazioni sia personali che riguardanti l’ambiente letterario.

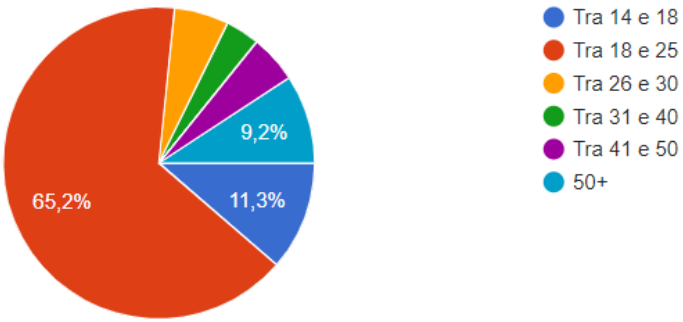
Il lettore, infatti, è caratterizzato dalle seguenti informazioni: l’età, l’essere genitori o meno di figli minorenni, l’occupazione, gli hobby, la quantità di libri letti in un anno, i generi letterari preferiti, la preferenza sul numero di pagine di un libro

(inferiore/superiore a una certa soglia, pari a 400) e i criteri in base ai quali effettua la scelta del nuovo libro da leggere.

Purtroppo, non tutti i dati raccolti sono stati significativi e utili per il problema in questione, a causa di una scarsa distribuzione delle risposte. Di seguito, sono stati allegati degli esempi.

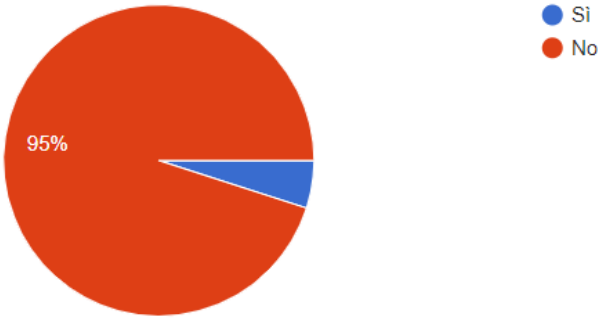
Quanti anni hai?

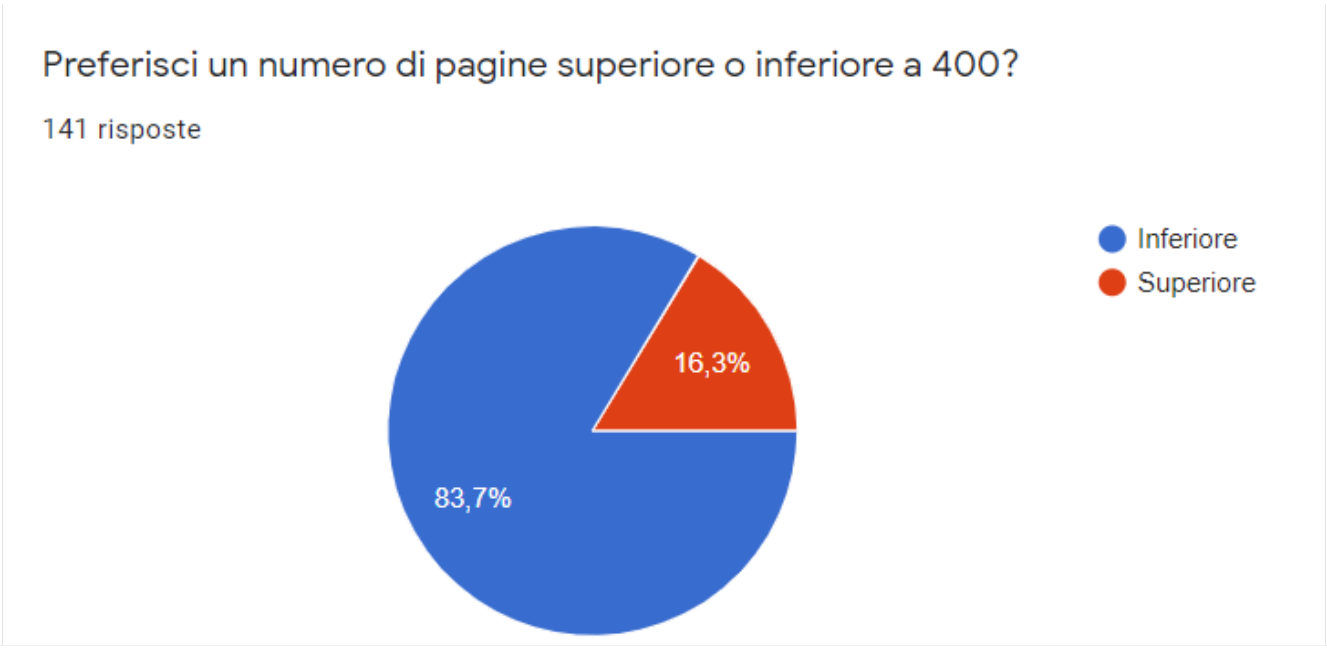
141 risposte

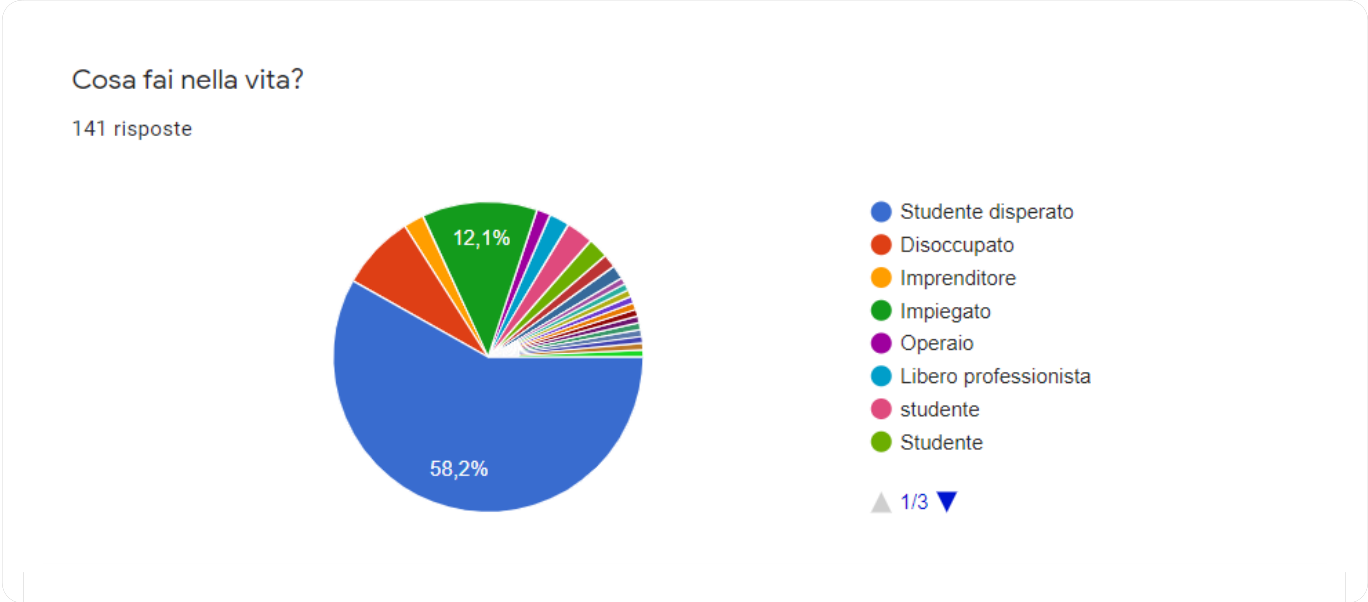


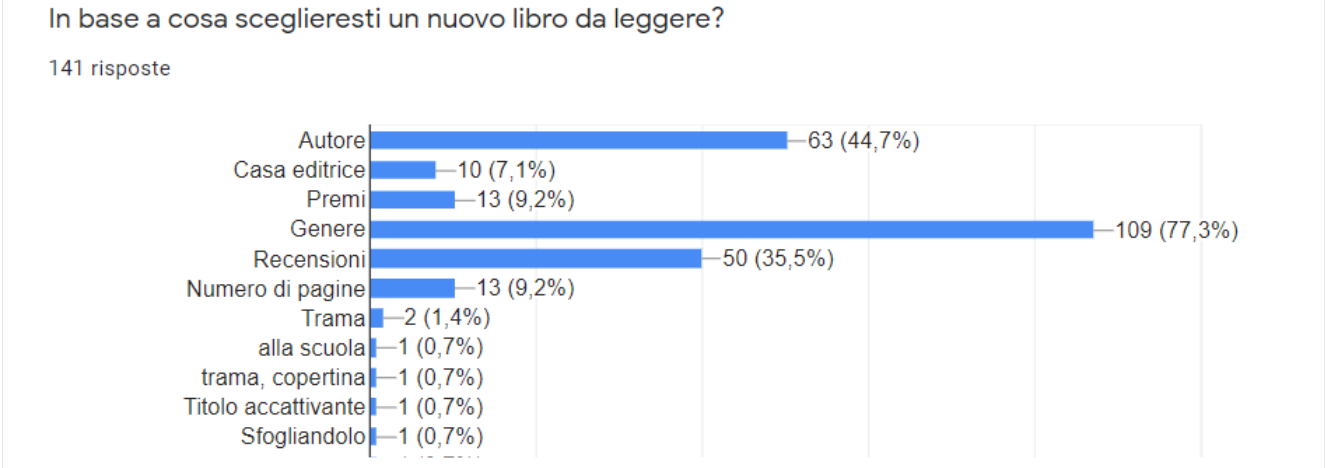
Hai figli minorenni?

141 risposte









Nell'ultimo caso, abbiamo deciso di eliminare questa caratteristica per due ragioni:

- la preponderanza della risposta "genere" rispetto alle altre;
- la presenza della caratteristica "genere preferito" che può causare ridondanza.

Per questo motivo, si è deciso di filtrare il dataset di origine, con lo scopo di ridurre lo spazio di dimensione dei dati in input, andando a considerare solo le caratteristiche realmente utili per la risoluzione del problema, ovvero: hobby, numero di libri letti all'anno e genere preferito.

4. Definizione della tipologia di apprendimento

La profilazione utenti in questo caso si basa su campioni non associabili ad etichette. Dunque, l'unico approccio disponibile per questo tipo di apprendimento (non supervisionato) è quello del clustering, che restituisce un set di cluster contenenti degli oggetti raggruppati a seconda della similarità reciproca.

Gli algoritmi scelti e considerati per la realizzazione dell'agente intelligente in questione dovranno partizionare il dataset fino a copertura completa. Per soddisfare questa esigenza si è scelto di confrontare gli algoritmi *K-MEANS* e *DBSCAN*, entrambi esclusivi, agglomerativi e seriali.

Per questa ragione, è necessario stabilire due criteri:

- criterio di "bontà": coefficiente di forma;
- criterio di similitudine.

4.1 Criterio di similitudine

Il passaggio successivo è stato quello di associare dei valori (numerici) alle varie opzioni di ogni caratteristica non numerica in base all'affinità tra loro.

```python
#creiamo dizionario generi
genre_dic = {
    'Fumetti': 1,
    'Avventura': 4,
    'Fantasy': 6,
    'Fantascienza': 10,
    'Horror': 15,
    'Psicologico ': 20,
    'Psicologico': 20,
    'Thriller': 22,
    'Giallo': 24,
    'Gangster': 26,
    'Drammatico': 34,
    'Romantico': 38,
    'Poesia': 40,
    'Classici': 42,
    'Storico': 45,
    'Saggi': 48,
    'Scienza': 50,
    'Romanzo di formazione': 56,
    'Motivazionale': 57,
    'Attualit\ufffd ': 60,
    'Satira': 62

}
```

```
#creiamo dizionario hobby
hobby_dic = {
    'Videogiochi': 1,
    'Serie tv/film': 4,
    'Leggere': 8,
    'Arte': 10,
    'Suonare': 12,
    'Cantare': 14,
    'Ballare': 18,
    'Sport': 24,
    'Sport ': 24,
    'Viaggiare': 34,
    'Volontariato': 38,
    'Cucinare': 46,
    'Beauty': 50
}

#creiamo dizionario del numero di libri medio che l'utente legge
libri_dic = {
    'Nessuno': 0,
    'Tra 1 e 3': 1,
    'Tra 3 e 5': 2,
    'Pi\ufffd di 5 ': 3,
    ' Pi\ufffd di 5': 3,
    'Pi\ufffd di 5': 3
}
```

Ciò è stato fatto per permettere il corretto funzionamento degli algoritmi di clustering che si è deciso di utilizzare.

4.2 Problem Solved

Per risolvere problemi di formattazione nei dizionari sono state ripetute delle chiavi con la sola differenza di un carattere (blank space), nonostante sia una soluzione meno efficiente.

Per rendere omogenei i valori relativi alle caratteristiche generi preferiti e hobby, sono stati utilizzati dei meccanismi matematici in quanto il numero di possibili scelte del lettore dovevano essere comprese in un range di opzioni (da 1 a 5 per gli hobby e da 1 a 6 per i generi).

I dati sono stati poi scalati e normalizzati affinché la loro distribuzione assomigliasse ad una gaussiana.

5. Algoritmi K-MEANS e DBSCAN

5.1 DBSCAN

La tecnica PCA (Principal Componet Analisys) è stata utilizzata in modo da ridurre le caratteristiche di ogni campione del dataset da 3 a 2, individuando quelle principali.

La logica dell'algoritmo DBSCAN è basata sulla densità dei campioni, l'obbiettivo è, infatti, quello di creare cluster densi e correlati in cui i punti sono collegati tra loro in una certa superficie.

Per l'esecuzione dell'algoritmo è necessario stabilire 2 parametri:

• MinPts: il numero minimo di punti per considerare l'intorno di un punto denso;
• *e*: la distanza che definisce l'intorno circolare di ogni punto dai suoi vicini.

Sono stati considerati anche i punti di rumore, ovvero quelli per i quali il numero di punti nell'intorno di raggio *e* è minore di MinPts.

Per la selezione dei due parametri sono state effettuate varie sessioni di testing in modo da scegliere la combinazione ottimale.

e	**MinPts**	**Coefficiente di Forma**	**Numero cluster stimati**	**Punti di rumore**
0,2	3	-0,014	3	2
0,3	4	0,163	1	1
0,2	4	-0,065	5	5
0,2	6	-0,049	5	17
0,2	7	-0,104	6	21
0,4	14	-0,063	2	1

La scelta dei parametri che ha permesso di ottenere un coefficiente di forma relativamente buono, un numero di cluster stimati equilibrato e un numero di punti di rumore accettabile è ricaduta sul record evidenziato.

5.2 K-MEANS

A differenza del DBSCAN, che risolveva il problema della preselezione del numero di cluster, per l'esecuzione di questo algoritmo è stato necessario settare tale parametro, scelto confrontando vari risultati nella fase di testing in combinazione al valore del seme della funzione *random.seed(k).*

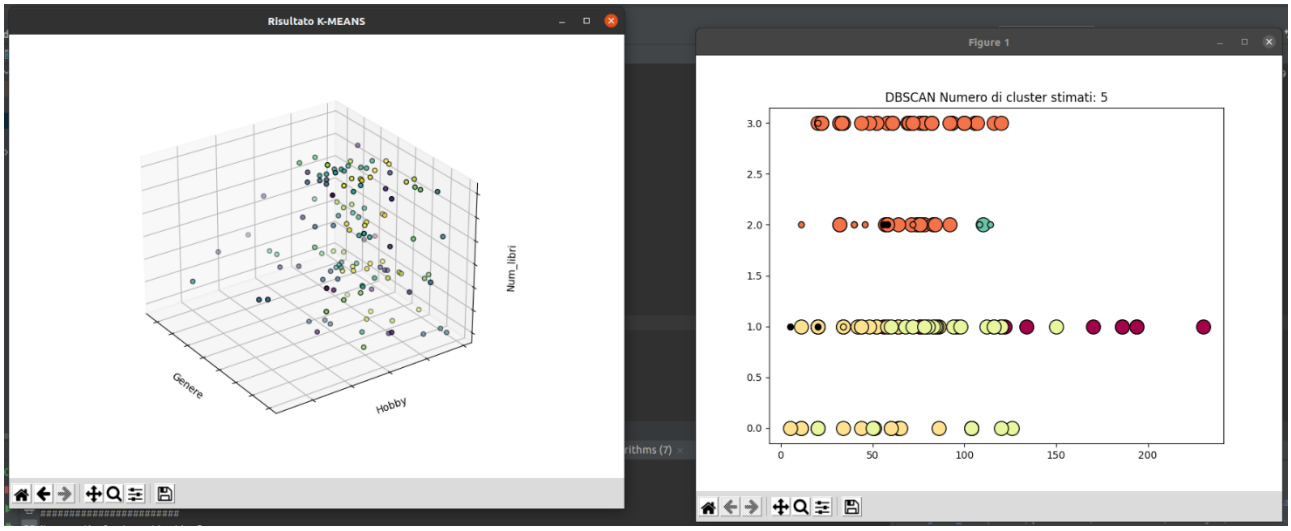
Di seguito sono mostrate le varie scelte effettuate.

Seed	**Numero di cluster**	**Coefficiente di forma**
10	15	0,393
12	15	0,386
11	15	0,396
11	20	0,424
12	20	0,425
12	12	0,391

20	20	0,407
21	21	0,429

La scelta dei parametri è stata dettata dalla combinazione di parametri che ha portato al miglior coefficiente di forma.

5.3 Risultati finali e osservazioni



Come si puo' evincere dalle immagini la distribuzione nel DBSCAN è lineare e questo fa presupporre che in nessun modo l'algoritmo possa essere una soluzione ottimale in questo contesto.

Mettendo a confronto i risultati ottenuti e valutati secondo il criterio del coefficiente di forma, si è constatato che l'algoritmo K-MEANS sia stato migliore del DBSCAN.

Queste osservazioni, sommate alla consapevolezza dell'efficienza del K-MEANS a livello di calcolo, hanno indotto alla scelta definitiva del K-MEANS come algoritmo di apprendimento per i fini preposti al progetto SocialBook.

Link al repository: https://github.com/AleBarbados/NewSocialBookRepo/tree/fia