
Programmazione 1

Esercitazione 7

Cognome:

Nome:

Matricola:

Questa è la prima di due esercitazioni che vi guideranno alla scrittura di un programma che possa fare delle semplici previsioni sui voti che un utente potrebbe dare a dei film che non ha ancora valutato.

L'obiettivo didattico dell'esercitazione è di far pratica nello scrivere programmi per la lettura di file di testo, nella costruzione di liste di liste o tuple, e nell'uso di dizionari. Si consiglia pertanto di ripassare il contenuto del notebook Lab 10.

Per questa esercitazione si devono usare i seguenti file di testo (i dati sono reali):

1. **u.user**: contiene le informazioni sugli utenti.
2. **u.item**: contiene le informazioni sui film.
3. **u1.base**: contiene le valutazioni degli utenti ai film, su una scala da 1 a 5.

L'esercitazione va svolta andando completare il codice per le funzioni contenute nel file `Testing_es_7.py` disponibile sul sito. Gli esercizi da svolgere in ordine sono:

1. Scrivere una funzione `ParseUsers(filename)` che legge il file di testo chiamato `filename` e restituisce una lista di tuple, in cui ogni tupla contiene le informazioni di un utente. Il file di input contiene una riga per ogni utente con le informazioni seguenti:

```
user id | age | gender | occupation | zip code
```

2. Scrivere una funzione `CountGender(Ls)` che prende in input la lista di tuple `Ls` contenente le informazioni degli utenti e conta il numero di utenti per ogni *gender*.

Il risultato dovrebbe essere 273 utenti di sesso femminile e 670 di sesso maschile.

3. Scrivere una funzione `CountOccupation(Ls)` che prende in input la lista di tuple `Ls` contenente le informazioni degli utenti e conta il numero di utenti per ogni tipo di occupazione. Il conteggio viene memorizzato in un dizionario in cui la chiave è il tipo di occupazione, mentre il valore è il rispettivo numero di utenti.

4. Scrivere una funzione **CountAge(Ls)** che prende in input la lista di tuple **Ls** contenente le informazioni degli utenti e conta il numero di utenti per le seguenti fasce di età: ($x < 18, 18 \leq x < 25, 25 \leq x < 40, 40 \leq x < 65, x > 65$). La funzione deve restituire una lista con i valori per le relative fasce di età.

5. Scrivere una funzione **ParseFilm(filename)** che legge il file **filename** e restituisce un dizionario, in cui ogni chiave è l'id di un film e il rispettivo valore è lista con le informazioni sul film. Il file di input contiene una riga per ogni film, e ogni riga contiene in ordine le informazioni seguenti:

```
movie id|movie title|release date|video release date|IMDb URL|unknown|Action|
Adventure|Animation|Children's|Comedy|Crime|Documentary|Drama|Fantasy|Film-Noir|
Horror|Musical|Mystery|Romance|Sci-Fi|Thriller|War|Western
```

6. Scrivere una funzione **CountItem(Fs, i)**: che prende in input un dizionario **Fs** come restituito dalla funzione precedente, e un indice $i > 5$, e conta il numero di film con il genere corrispondente all' i -esimo elemento della lista che descrive un film. Per esempio, il numero di film thriller è 251.
7. Scrivere una funzione **CountYears(Fs)**: che prende in input un dizionario **Fs** come restituito dalla funzione precedente, e conta il numero di film usciti in ogni anno. Per contare l'anno, si deve elaborare il campo **release date** estraendone l'anno (si consiglia di usare la funzione **split**). Per esempio, si dovrebbe ottenere che nel 1996 sono usciti 355 film.

8. Scrivere una funzione **ParseRating(filename)** che legge il file **filename** e restituisce una lista di liste, una per ogni rating. Il file di input contiene le informazioni seguenti:

```
user id | item id | rating | timestamp
```

9. Scrivere una funzione **CountFiled(Ls, i)** che prende in input una lista di rating e un indice $i < 3$, e per ogni elemento della lista conta il numero di occorrenze diverse. Questa funzione può essere usata per contare il numero di volte che ciascun voto è stato dato (usata con $i = 2$), il numero di voti dati da ciascun utente ($i = 0$), oppure il numero di voti ricevuti da ciascun film ($i = 1$).
10. Scrivere una funzione **PrintTopFilm(Ds, top, Ts)** che prende in input un dizionario che ha una chiave per ogni film con il valore del numero di voti ricevuti, un numero intero **top**, e il dizionario che mappa ogni id di un film nelle sua lista di informazioni, e stampi a video il titolo e il numero di valutazioni per i primi **top** film più votati.