

---

## Programmazione 1

## Esercitazione 8

Cognome:

Nome:

Matricola:

---

Questa è la seconda esercitazione per arrivare all'implementazione di un programma che possa fare delle semplici previsioni sui voti che un utente potrebbe dare a dei film che non ha ancora valutato.

Rispetto alla prima esercitazione si deve usare anche file di testo:

1. `u1.test`: contiene le valutazioni degli utenti ai film, su una scala da 1 a 5.

L'esercitazione va svolta in parte riprendendo le soluzioni dell'esercitazione 7, ma scrivendo un nuovo file python. Scrivere in coda al file all'interno del blocco `if __name__ == "__main__":` l'insieme di chiamate alle vostre funzioni che permettano di controllarne il corretto funzionamento.

1. Scrivere la funzione `ParseRating(filename)` che legge il file `filename` e restituisce un dizionario con una chiave per ogni coppia (user id, item id), e come valore il rispettivo rating. Il rating deve essere memorizzato come un numero intero. Il file di input contiene le informazioni seguenti:

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

2. Scrivere una funzione `ComputeAverage(Ls)` che prende in input un dizionario di valutazioni (come letto all'esercizio 1) e calcola la media aritmetica di tutte le valutazioni. La funzione deve restituire la media.

3. Scrivere una funzione `ComputeItemAverage(Ls)` che prende in input un dizionario di valutazioni e calcola per ciascun film (in genere si parla di *item*) la media aritmetica di tutte le valutazioni. La funzione deve restituire un dizionario con una chiave per ogni film id e come valore la media di voti ricevuti da quel film.
4. Scrivere una funzione `ComputeUserAverage(Ls)` che prende in input un dizionario di valutazioni e calcola per ciascun utente la media aritmetica di tutte le valutazioni. La funzione deve restituire un dizionario con una chiave per ogni user id e come valore la media di voti dati da quell'utente.
5. Scrivere una funzione `ComputeUserTypeAverage(Ls, Us)` che prende in input un dizionario di valutazioni e un dizionario con i dati degli utenti e calcola per ciascuna categoria di utente la media aritmetica di tutte le valutazioni. La funzione deve restituire un dizionario con una chiave per ciascuna categoria di utente e come valore la media dei voti dati da quella categoria.

6. Scrivere una funzione `PredictAvg(Ls)` che prende in input il dizionario di valutazioni da predire e per ogni coppia (utente, film) per cui si deve fare una previsione, restituisce la media globale. Usare la funzione passandogli in input la lista di valutazioni contenuta nel file `u1.test`. In questo caso, la funzione deve restituire un dizionario, in cui la chiave è la coppia (id utente, id item), e il valore è il rating predetto, ovvero la media globale, opportunamente arrotondata (non si possono predire frazioni di voto).

7. Scrivere una funzione `RMSE(Ls)` che calcola il *Root Mean Square Error (RMSE)* delle vostre predizioni. Ovvero, date  $n$  predizioni  $\hat{y}_i$  e dati i valori reali delle rispettive osservazioni  $y_i$ , si calcola l'errore complessivo con la formula:

$$\text{RMSE}(\hat{y}, y) = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

8. Valutare il RMSE per le predizioni ottenute con la funzione `PredictAverage` usando le valutazioni contenute nel file di test.

9. Scrivere le funzioni `PredictAvgItem(Ls)`, `PredictAvgUser(Ls)` e `PredictAvgCategory(Ls)`, che effettuano le predizioni di voto in base alle rispettive medie calcolate con le soluzioni degli esercizi precedenti. Usando come funzione di valutazione la RMSE, confrontare le prestazioni dei 4 semplici approcci proposti. Se un film non ha ricevuto nessun voto oppure un utente non ha dato nessun voto il rispettivo valore predetto potrebbe essere la media globale.
10. **CHALLENGE 5:** Elaborare un metodo per ottenere il valore di RMSE il più piccolo possibile. Il metodo proposto dovrebbe ottenere dei risultati migliori di quelli ottenuti con i 4 metodi precedenti, chiamati *baselines*. Testare il vostro metodo usando i seguenti dati: elaborare il modello di previsione con il file `uX.base`, con  $X \in \{1, \dots, 5\}$ , e testare il modello con i rispettivi `uX.test` file, con  $X \in \{1, \dots, 5\}$ .

**NOTA:** non è possibile incrociare i file di base con quelli di test.

**Facoltativo:** Mandare la vostra soluzione per email al docente. Riportare nell'email il valore di RMSE ottenuto per i cinque diversi data set.