# **REPORT METODI INFORMATICI PER LA GESTIONE AZIENDALE**

Davide Zangari 844760 giugno 2022

Progetto singolo: devono essere svolti tutti i passi di analisi del “progetto base” più a scelta 1 analisi aggiuntiva: ho svolto tutti i passi di analisi del progetto base e l’analisi aggiuntiva numero 3.

INTRODUZIONE AL PROBLEMA

Ho importato dal sito MovieLens (www.movielens.umn.edu) i file u.data.csv, u.user.csv e u.item.csv che contengono dati raccolti tra il 20 Marzo 1996 e il 24 Settembre 2018 riguardanti 943 utenti, 1682 film, e le relative recensioni (circa 100000). Ogni utente ha recensito almeno 20 film con voti che vanno da 1 a 5. Questo set di dati è stato raccolto dal progetto GroupLens all’università del Minnesota. Sono presenti anche numerose informazioni demografiche sugli user (età, genere , occupazione...)

Ho creato un Recommender System che suggerisce i film che potrebbero piacere per ciascun utente, prevedendo la valutazione che potrebbero dare ai film; quindi, ho raggruppato gli utenti in diversi cluster in base alle proprie preferenze relative ai film, basandomi sulle recensioni.

**PASSO DI ANALISI BASE NUMERO 1: ANALISI ESPLORATIVA**

Ho importato alcune librerie, ad esempio:

Pandas

Numpy

Seaborn

Surprise

Matplotlib.pyplot

Nltk

Ho letto i file .csv scaricati da MovieLens, e ho svolto l’analisi esplorativa calcolando alcune statistiche descrittive per trarre conclusioni in merito alle popolazioni statistiche, dopo aver preprocessato i dati (cioè ho ispezionato i dati, controllato i valori nulli, controllato i tipi di dati, etc)

Queste le statistiche descrittive dei dati .csv importati da MovieLens:

u.data.csv shape(100000, 4)

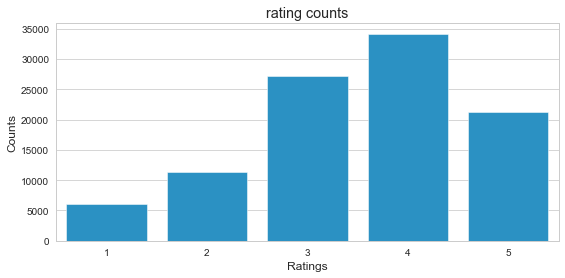
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **user\_id** | **movie\_id** | **rating** | **timestamp** |
| **count** | 100000.00000 | 100000.000000 | 100000.000000 | 1.000000e+05 |
| **mean** | 462.48475 | 425.530130 | 3.529860 | 8.835289e+08 |
| **std** | 266.61442 | 330.798356 | 1.125674 | 5.343856e+06 |
| **min** | 1.00000 | 1.000000 | 1.000000 | 8.747247e+08 |
| **25%** | 254.00000 | 175.000000 | 3.000000 | 8.794487e+08 |
| **50%** | 447.00000 | 322.000000 | 4.000000 | 8.828269e+08 |
| **75%** | 682.00000 | 631.000000 | 4.000000 | 8.882600e+08 |
| **max** | 943.00000 | 1682.000000 | 5.000000 | 8.932866e+08 |

u.item-csv shape(1682, 24) ha molte colonne che rappresentano principalmente la tipologia di film, qui ne visualizzo 6 per leggibilità

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **movie\_id** | **video\_release\_date** | **unknown** | **Action** | **Adventure** | **Animation** |
| **count** | 1682.000000 | 0.0 | 1682.000000 | 1682.000000 | 1682.000000 | 1682.000000 |
| **mean** | 841.500000 | NaN | 0.001189 | 0.149227 | 0.080262 | 0.024970 |
| **std** | 485.695893 | NaN | 0.034473 | 0.356418 | 0.271779 | 0.156081 |
| **min** | 1.000000 | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 421.250000 | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 841.500000 | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 1261.750000 | NaN | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **max** | 1682.000000 | NaN | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

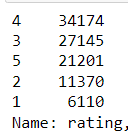
u.user.csv shape(943, 5)

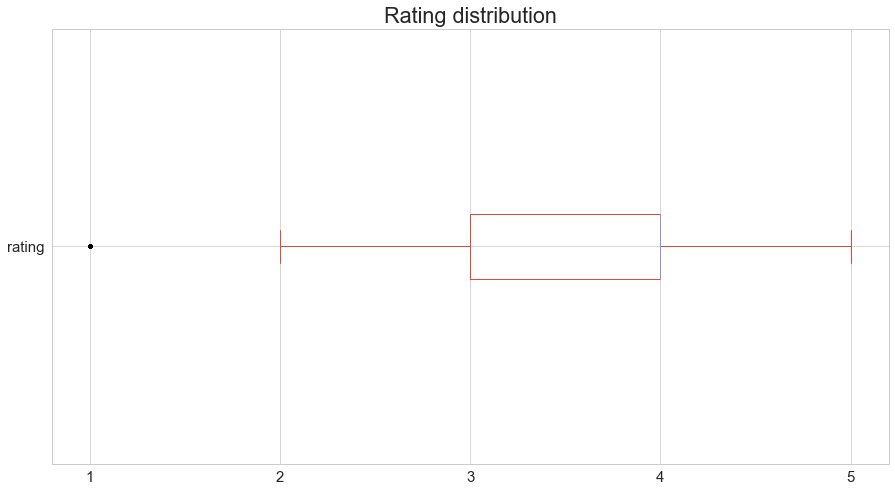
|  |  |  |
| --- | --- | --- |
|  | **user\_id** | **age** |
| **count** | 943.000000 | 943.000000 |
| **mean** | 472.000000 | 34.051962 |
| **std** | 272.364951 | 12.192740 |
| **min** | 1.000000 | 7.000000 |
| **25%** | 236.500000 | 25.000000 |
| **50%** | 472.000000 | 31.000000 |
| **75%** | 707.500000 | 43.000000 |
| **max** | 943.000000 | 73.000000 |



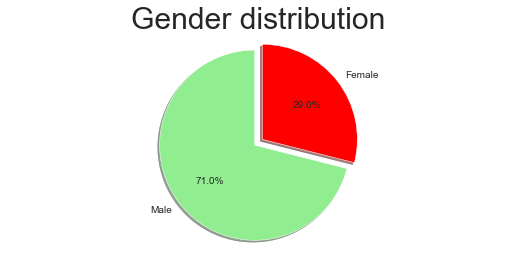
Qui sopra riporto il conteggio delle valutazioni ricevute. Salta all’occhio che ci sono molte valutazioni di 4 su 5 e poche valutazioni col punteggio minimo di 1 su 5 (circa 5000 su 100000), più precisamente:

Rating Counts

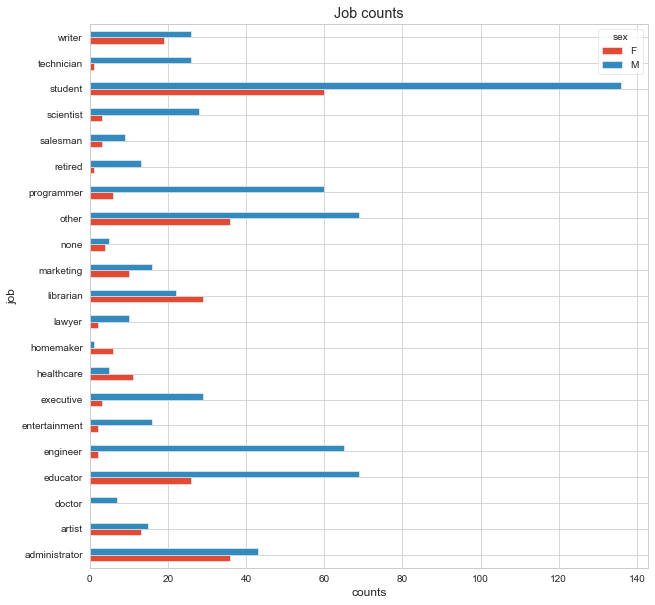


 4 è il voto più frequente

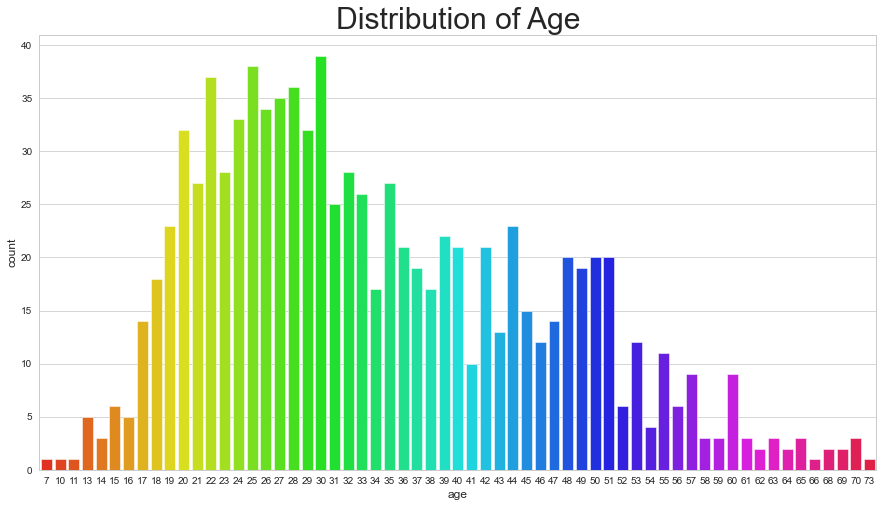
Sotto ho visualizzato la distribuzione maschi-femmine (670 contro 273) notando che ci sono molti più maschi che mettono recensioni in questi dati e la loro età media è simile (33.8 per le femmine; 34.1 per i maschi)

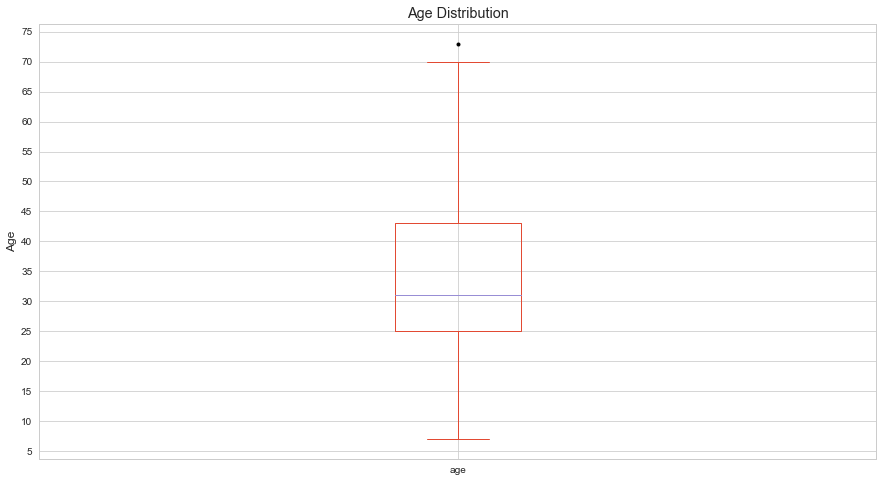


Ho inoltre riportato il conteggio per genere degli utenti in base ai loro lavori



Da quest’altro grafico posso notare che il database è ricco di recensioni di utenti relativamente giovani (tra 17 e 35 anni). Invece possiede poche recensioni da parte di utenti over 55 e under 17





Il boxplot per la distribuzione dell’età è parecchio informativo, noto che l’età mediana è 31 anni (per cui metà degli utenti ha meno di 31 anni), il primo quartile è a 25 anni (per cui 1 su 4 ha meno di 25 anni), il terzo quartile sui 43. La presenza di utenti over 70 è considerata anomala poiché c’è un outlier in corrispondenza dei 73 anni e il baffo superiore termina a 70 anni.

Ho esplorato per curiosità i dati e ho visto i rating dell’utente 1 ai primi 20 film

user\_id movie\_id rating  
1 1 5.0  
 2 3.0  
 3 4.0  
 4 3.0  
 5 3.0  
 6 5.0  
 7 4.0  
 8 1.0  
 9 5.0  
 10 3.0  
 11 2.0  
 12 5.0  
 13 5.0  
 14 5.0  
 15 5.0  
 16 5.0  
 17 3.0  
 18 4.0  
 19 5.0  
 20 4.0

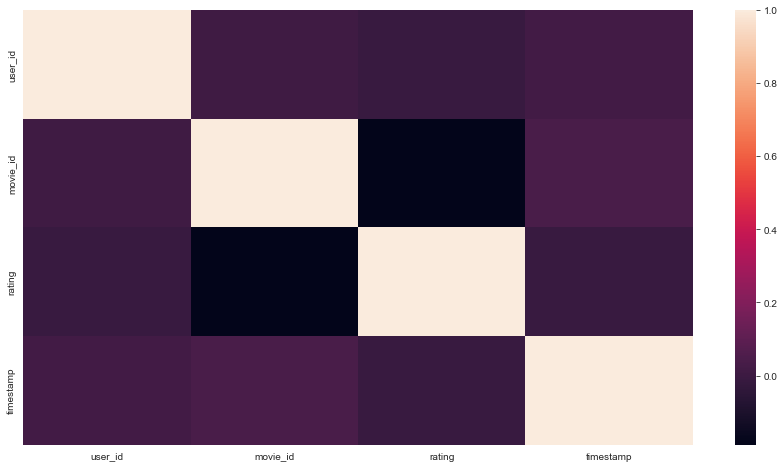
Ho visualizzato i voti più ricorrenti dei primi 10 utenti:

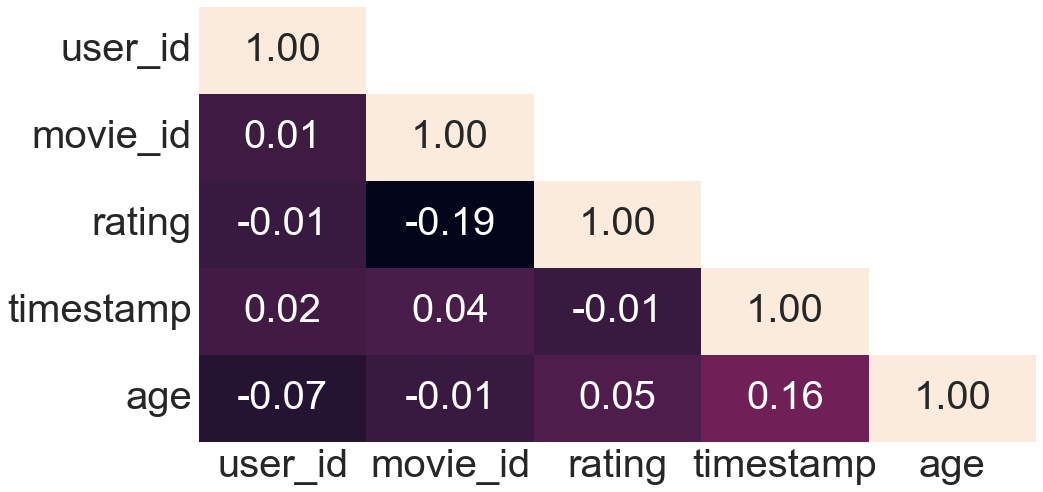
User\_id rating  
1 4  
2 4  
3 3  
4 5  
5 3  
6 4  
7 4  
8 4  
9 4  
10 4

Da questo scatterplot ho notato che i primi film per id hanno più recensioni



Nell’analisi di correlazione non ho trovato correlazioni positive significative tra i dati, come mi aspettavo, per cui concludo che le variabili sono tra loro indipendenti. Una correlazione di 1 ci indica che c’è una perfetta correlazione positiva tra le variabili, 0 indica che le variabili non sono correlate, -1 ci indica che c’è una perfetta correlazione negativa.





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **user\_id** | **movie\_id** | **rating** | **timestamp** |
| **user\_id** | 1.000000 | 0.010377 | -0.009371 | 0.019082 |
| **movie\_id** | 0.010377 | 1.000000 | -0.189124 | 0.041867 |
| **rating** | -0.009371 | -0.189124 | 1.000000 | -0.012003 |
| **timestamp** | 0.019082 | 0.041867 | -0.012003 | 1.000000 |

**PASSO DI ANALISI DI BASE NUMERO 2: FILLING DELLA MATRICE DI RATING**

Ho usato l'algoritmo user-based knn e la cosine similarity come misura della somiglianza. Ho diviso la matrice di rating in train (80% dei dati) e test set (20%), allenato l'algoritmo per il filling della matrice usando il train set e poi effettuato le predizioni sul test set.

La cosine similarity rappresenta il coseno dell’angolo compreso tra due vettori per cui varia tra –1 e +1, ma se le componenti del vettore sono positive allora varia nell’intervallo [0, +1]

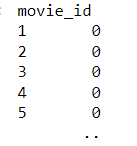
Ho visualizzato la matrice di rating e notato che contiene moltissimi valori NaN (not a number) indicanti recensioni mancanti; infatti è logico pensare che difficilmente un utente possa aver valutato tutti i film presenti in un catalogo.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **movie\_id** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **...** |
| **user\_id** |  |  |  |  |  |  |  |  |  |  |  |
| **1** | 5.0 | 3.0 | 4.0 | 3.0 | 3.0 | 5.0 | 4.0 | 1.0 | 5.0 | 3.0 | ... |
| **2** | 4.0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 2.0 | ... |
| **3** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... |
| **4** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... |
| **5** | 4.0 | 3.0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... |

Per riempire la matrice l’ho fatta scorrere e ogni volta che trovavo un NaN lo sostituivo con la predizione dell’algoritmo knn per quella specifica coppia film-utente e approssimando il valore per intero (poiché la predizione restituisce un numero razionale) usando la funzione round()

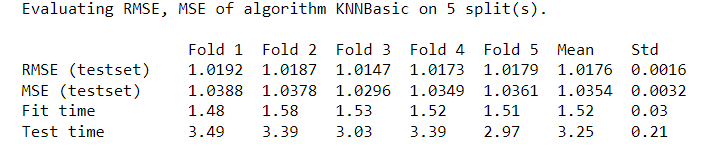
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **movie\_id** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **...** |
| **user\_id** |  |  |  |  |  |  |  |  |  |  |  |
| **1** | 5.0 | 3.0 | 4.0 | 3.0 | 3.0 | 5.0 | 4.0 | 1.0 | 5.0 | 3.0 | ... |
| **2** | 4.0 | 4.0 | 3.0 | 4.0 | 4.0 | 3.0 | 4.0 | 4.0 | 4.0 | 2.0 | ... |
| **3** | 4.0 | 4.0 | 3.0 | 3.0 | 4.0 | 3.0 | 4.0 | 4.0 | 4.0 | 4.0 | ... |
| **4** | 4.0 | 4.0 | 3.0 | 3.0 | 4.0 | 3.0 | 4.0 | 4.0 | 4.0 | 4.0 | ... |
| **5** | 4.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 4.0 | 4.0 | 4.0 | 4.0 | ... |

Ho poi controllato l’assenza di valori nulli visualizzando le prime righe della matrice e controllando con le funzioni .isnull().sum()



**PASSO DI ANALISI BASE NUMERO 3: PERFORMANCE DELL’ALGORITMO KNN CON MSE E RMSE**

Dopo aver creato le predizioni e riempito la matrice di rating, ho calcolato le metriche di performance dell’algoritmo knn usando la crossvalidation



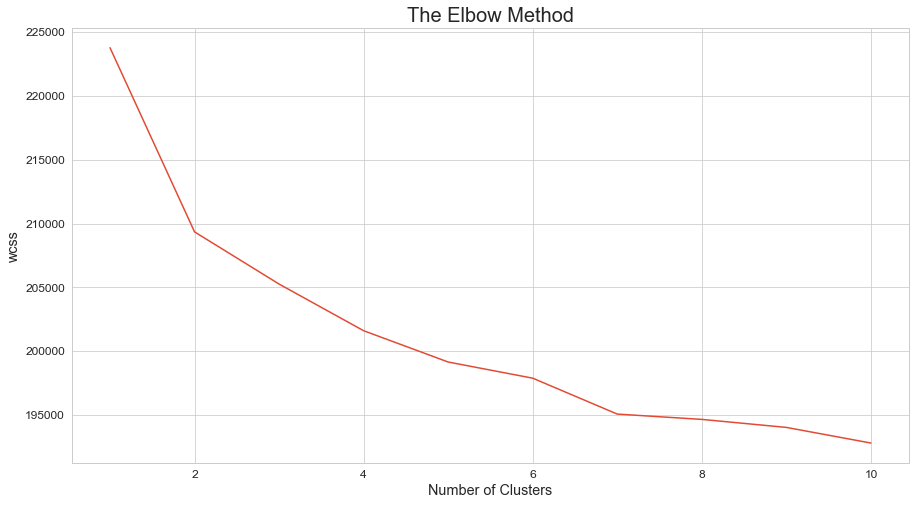
Ho notato una bassa deviazione standard e valori relativamente bassi degli MSE e RMSE medi utilizzando 5 fold.

L’MSE è l'errore quadratico medio, indica la discrepanza quadratica media fra i valori dei dati osservati ed i valori dei dati stimati.

L’RMSE è una metrica che corrisponde alla radice quadrata dell’MSE

**PASSO DI ANALISI BASE NUMERO 4: SEGMENTAZIONE DEGLI UTENTI CON KMEANS**

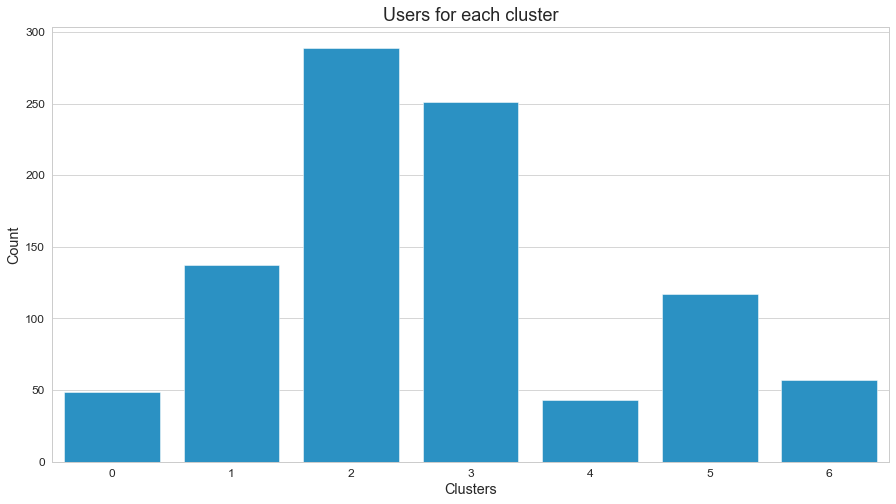
Trovo il numero di k cluster ottimale attraverso il metodo del gomito.



Il gomito non è perfettamente visibile, per cui ho scelto k=7

Ho calcolato le silhouette con sktlearn per avere un valore della bontà dei cluster. Con k=7 ho silhouette 0.07 che è il valore migliore ottenuto testando manualmente con k compreso tra 6 e 10

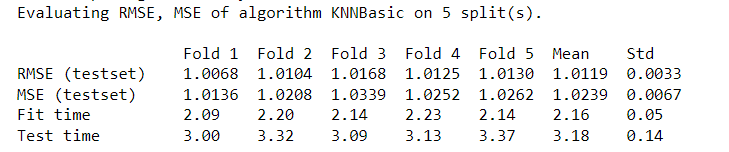
E ho visualizzato il numero di utenti per cluster:



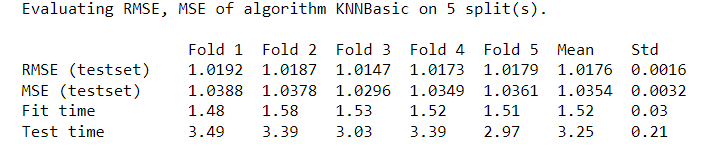
**ANALISI AGGIUNTIVA NUMERO 3: FILLING DELLA MATRICE DI RATING CON KNN E CORRELAZIONE DI PEARSON, CONFRONTARE I RISULTATI AVUTI CON LA COSINE SIMILARITY**

Ho riempito la matrice di rating che conteneva i NaN con le valutazioni intere ricavate dalle predizioni dell’algoritmo knn usando l’indice di Pearson come misura di similarità.

Ho usato la crossvalidation con 5 fold per calcolare RMSE e MSE



Ottenendo risultati molto simili a quelli ottenuti con la cosine similarity:



Ho calcolato le metriche di performance delle previsioni pure (con test set) sfruttando la libreria Surprise senza usare la crossvalidation per fare un confronto e anche per salvare nelle variabili i risultati delle misurazioni.

La libreria Surprise fornisce una infrastruttura per testare e sviluppare algoritmi di raccomandazione, tra cui:

▪️ Dataset

▪️ Algoritmi

▪️ Metodi di valutazione

accuracy con Surprise e Pearson: RMSE: 0.8222 MSE: 0.6760

accuracy con Surprise e cosine similarity: RMSE: 0.9244 MSE: 0.8544

Dunque ho meno errori usando l’indice di Pearson nell’algoritmo knn.

**CONCLUSIONI**

Lo sviluppo di questo progetto è stato interessante perché è un esempio concreto dell’utilità dell’informatica e della statistica nel mondo reale. Dall’analisi dei dati scaricati da MovieLens si evince che si possono raccogliere molte informazioni relative agli utenti che recensiscono film ma riuscire a predire correttamente le valutazioni future di un utente è più complicato, in particolare quando questo ha poche o zero recensioni, per cui più dati si hanno a disposizione e maggiore potrà essere l’accuratezza del sistema di raccomandazione. Il processo di clustering è ancora più delicato e sensibile alla qualità e quantità dei nostri dati.