Laboratorio 1 – Grupo 09

Alejandro García Flores, Francisco José Correa Rozo, Camilo Esteban Rozo Benítez
{a.garcia13, fj.correa10, ce.rozob}@uniandes.edu.co
Fecha de presentación: febrero 16 de 2023

Tabla de contenido

1. Entendimiento de los datos

En ratings encontramos 943 usuarios con id único, que han generado 100,000 calificaciones para 1682 ítems (películas) distintos, de las cuales los 5 primeros acumulan cerca de 500 calificaciones o más, mientras que los 143 últimos sólo cuentan con 1 rating.

```
 1 ratings.value_counts()
```

```
user_id  item_id  rating  timestamp
1        1        5       874965758    1
606      91       5       880926610    1
         144      4       880924664    1
         138      3       880927923    1
         135      5       880926245    1
                                       ..
311      747      3       884364502    1
         739      4       884365823    1
         735      4       884366637    1
         732      4       884365617    1
943      1330     3       888692465    1
Length: 100000, dtype: int64
```

FIGURA 2. CONTEO DE RATINGS

```
1 print(ratings.item_id.value_counts())
2 print('-----')
3 print(len(ratings.item_id.value_counts()))
```

```
50      583
258     509
100     508
181     507
294     485
       ...
852       1
1505      1
1653      1
1452      1
1641      1
Name: item_id, Length: 1682, dtype: int64
-----
1682
```

FIGURA 1. CONTEO DE RATINGS POR PELÍCULA

La distribución de los ratings está sesgada hacia la derecha. Esto indica que aproximadamente el 80% de los ratings ha sido igual o superior a 3. Esto nos indica que los items calificados suelen ser mejor recibidos por los usuarios. Una hipótesis para explicar esto es que los usuarios suelen ver películas que intuitivamente creen que van a ser de su agrado o que, precisamente, su elección de consumo está influenciada por una recomendación ya sea voz a voz, o a través de un algoritmo. Otra hipótesis podría ser que los usuarios suelen calificar con más frecuencia aquellos items que en efecto fueron de su agrado

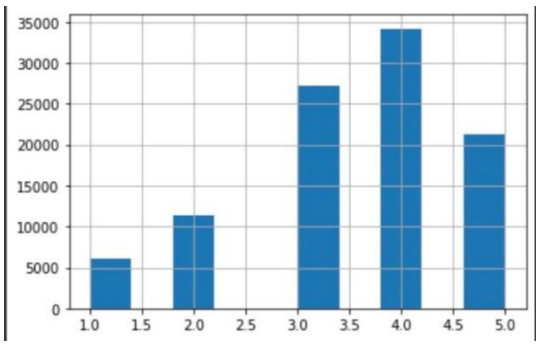

FIGURA 3. HISTOGRAMA DE CANTIDAD DE PELÍCULAS POR VALOR DE RATING

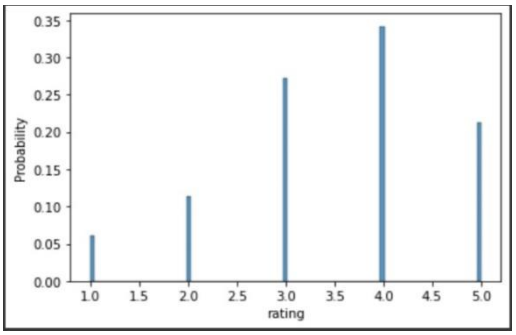

FIGURA 4. HISTOGRAMA DE DENSIDAD DE RATINGS

El número de ratings que otorga cada usuario muestra una 'cola' larga, lo que nos indica que la mayoría de los usuarios califica en promedio entre 20 y 100 veces (la cola está truncada a la izquierda ya que se excluyen usuarios con menos de 20 calificaciones) y hay unos pocos usuarios que califican más de 500 veces. Se identifica que más del 37% de usuarios no supera los 50 ratings y son relativamente pocos los que superan los 100 ratings, por lo que la matriz de utilidad, a crear sería dispersa, que concuerda con lo visto en clase

user_id	mean	count
405	1.834464	737
655	2.908029	685
13	3.097484	636
450	3.864815	540
276	3.465251	518
...	...	...
685	2.050000	20
475	3.600000	20
36	3.800000	20
732	3.700000	20
596	3.600000	20

943 rows × 2 columns

FIGURA 5. CONTEO DE RATINGS POR USUARIO

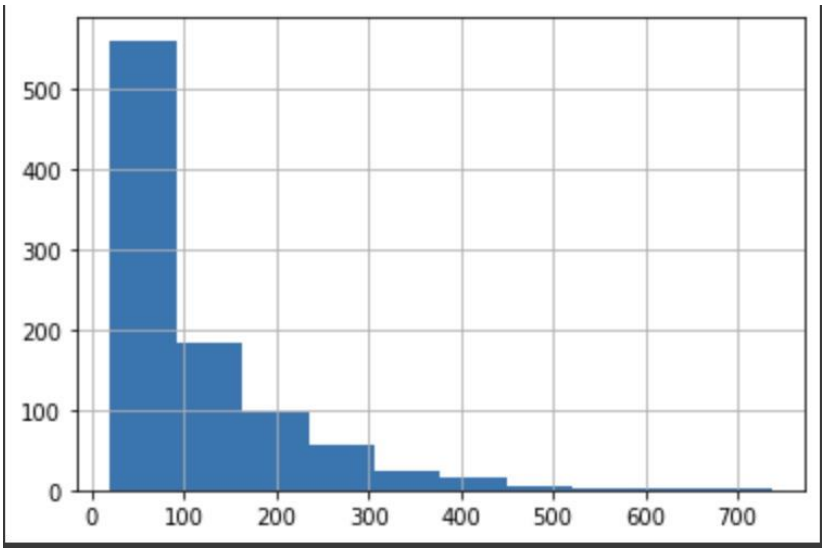

FIGURA 6. HISTOGRAMA DE CANTIDAD DE RATINGS POR USUARIO

Cuando examinamos el rating promedio otorgado por cada usuario, vemos una distribución de estos ratings más normalizada, centrada entre 3.5 y 4.0. Con alrededor del 75% de usuarios ubicados entre 3 y 4, esto puede ser causado, porque los usuarios puedan tener una tendencia a calificar las películas hacia los extremos (dónde vemos que este puede ser el caso para algunos usuarios puesto que el histograma de calificaciones en general sí muestra que estas áreas no están vacías). Lo que al final causa que el promedio sea cercano a la mediana. Junto con otros usuarios, que según el histograma de calificaciones son la mayoría, tienen la tendencia a no dar ratings cercanos a extremos, sino más bien neutrales, esto se confirma ya que alrededor del 60% de las calificaciones de películas son 3 o 4.

	user_id	rating
0	1	3.610294
1	2	3.709677
2	3	2.796296
3	4	4.333333
4	5	2.874286
...	...	...
938	939	4.265306
939	940	3.457944
940	941	4.045455
941	942	4.265823
942	943	3.410714

943 rows × 2 columns

FIGURA 7. RATING PROMEDIO POR USUARIO

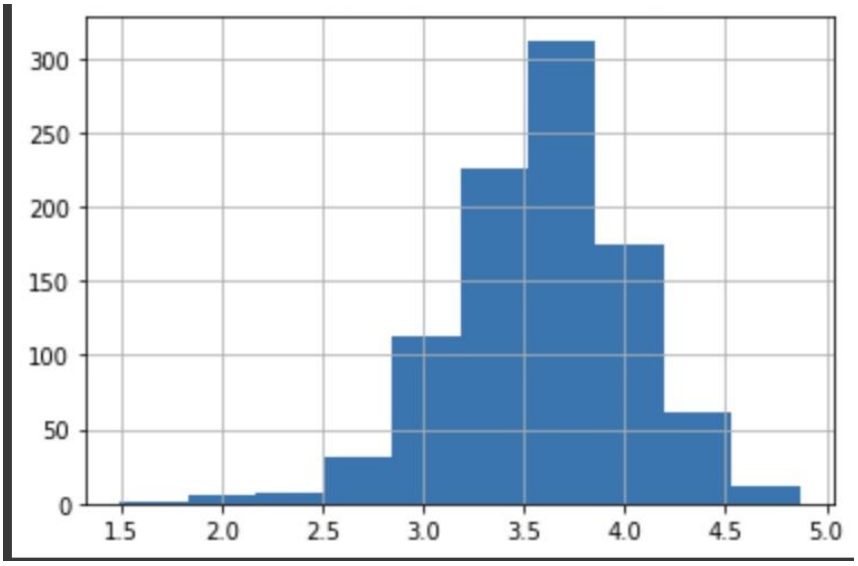

FIGURA 8. DISTRIBUCIÓN DE RATING PROMEDIO POR USUARIO

Finalmente, observamos que los items con más calificaciones (Legends of the Fall, George of the Jungle, Heavy Metal, GoodFellas y Breakdown) son los más calificados y acumulan un total de 2592 calificaciones. El top-100 de items, acumula cerca del 15% de las calificaciones en el dataset y el top-300, cerca del 47%, por lo que claramente se evidencia el efecto de cola larga.

Nombre	Cantidad de calificaciones
Legends of the Fall (1994)	583
George of the Jungle (1997)	509
Heavy Metal (1981)	508
GoodFellas (1990)	507
Breakdown (1997)	485
Marvin's Room (1996)	481
Evita (1996)	478
GoldenEye (1995)	452
In & Out (1997)	431
Cable Guy, The (1996)	429

FIGURA 9. TOP PELÍCULAS CON MAYOR CANTIDAD DE RATINGS

```
[85]  1 mt['count'][0:4].sum()
      882
[91]  1 mt['count'][0:100].sum()
      14892
      1 mt['count'][0:300].sum()
      47063
```

FIGURA 10. CANTIDAD DE RATINGS DE LOS TOP 5, 100 Y 300

2. Creación de lista de recomendación no personalizadas

Generamos una lista de 10 recomendaciones no personalizadas, basada únicamente en el promedio de la calificación otorgada a los items, tal como se presenta a continuación:

```
1 list_no_per = mt.sort_values(by='mean', ascending=False)
2 list_no_per.pop('var')
3 list_no_per[0:10].sort_values(by='count', ascending=False
```

item_id	mean	count	movie title
1293	5.0	3	Ayn Rand: A Sense of Life (1997)
1189	5.0	3	That Old Feeling (1997)
1500	5.0	2	Prisoner of the Mountains (Kavkazsky Plennik) ...
1467	5.0	2	Cure, The (1995)
814	5.0	1	One Fine Day (1996)
1599	5.0	1	Guantanamera (1994)
1201	5.0	1	Maybe, Maybe Not (Bewegte Mann, Der) (1994)
1122	5.0	1	Last Time I Saw Paris, The (1954)
1653	5.0	1	Chairman of the Board (1998)
1536	5.0	1	Cosi (1996)

FIGURA 11. MEJORES PELÍCULAS SEGÚN SU RATING PROMEDIO

Esta aproximación no funciona bien cuando existen ítems con pocas calificaciones[1] (o una única) ya que no se puede evidenciar que la calificación sobre la que se basa la recomendación sea significativa, tal como se evidencia en la lista presentada anteriormente.

[1] https://www.evanmiller.org/how-not-to-sort-by-average-rating.html.

Para corregir esto, calculamos el límite inferior al 95% de confianza de la calificación de cada una de las películas usando el score Wilson para un parámetro de Bernoulli. Para ello, clasificamos 'ratings positivos' como aquellos mayores a 3 y calculamos el parámetro usando la suma de estos 'ratings positivos' sobre el total de las calificaciones para cada item. El resultado es el siguiente:

item_id	count	mean	pos	movie title	low_bound
64	283	4.445230	255	What's Eating Gilbert Grape (1993)	0.860720
479	179	4.251397	162	North by Northwest (1959)	0.853178
318	298	4.466443	265	Everyone Says I Love You (1996)	0.848560
98	390	4.289744	344	Snow White and the Seven Dwarfs (1937)	0.846252
483	243	4.456790	216	Maltese Falcon, The (1941)	0.843166
603	209	4.387560	185	It Happened One Night (1934)	0.834823
427	219	4.292237	193	Harold and Maude (1971)	0.831730
50	583	4.358491	501	Legends of the Fall (1994)	0.828769
357	264	4.291667	230	Spawn (1997)	0.825420
12	267	4.385768	232	Mighty Aphrodite (1995)	0.823148

FIGURA 12. MEJORES PELÍCULAS SEGÚN EL SCORE WILSON USANDO RATINGS > 3 COMO POSITIVOS

	mean	positive votes	movie title	total votes	wilson_lower_bound
477	4.481328	241	Maltese Falcon, The (1941)	243	0.970493
474	4.297753	178	Apartment, The (1960)	179	0.969039
596	4.410628	207	It Happened One Night (1934)	209	0.965786
478	4.226277	137	My Fair Lady (1964)	138	0.960097
177	4.362903	124	Clockwork Orange, A (1971)	125	0.956075
425	4.339535	215	Harold and Maude (1971)	219	0.953987
492	4.213333	150	Cat on a Hot Tin Roof (1958)	152	0.953299
193	4.110169	236	Terminator, The (1984)	241	0.952361
63	4.514493	276	What's Eating Gilbert Grape (1993)	283	0.949832
521	4.068063	191	Killing Fields, The (1984)	195	0.948453

FIGURA 13. MEJORES PELÍCULAS SEGÚN EL SCORE WILSON USANDO RATINGS ≥3 COMO POSITIVOS

3. Creación de modelo de filtrado colaborativo basado en similitud con usuarios o items cercanos

1. **Item item**
   Al usar la configuración inicial del notebook (KNN con 20 vecinos y 2 mínimo, basado en ítems), se obtuvo un RMSE de 1.045, por lo que el sistema encuentra ratings que estan una estrella por encima o por debajo del rating del usuario, tal cual lo menciona el notebook, esto porque la escala de ratings va de 0 a 5 y el RMSE resulta muy cercano a 1. En este contexto, consideramos que la medida da pie a considerar el modelo como aceptable, ya que desfasarse por un valor de 1, positivamente, en realidad termina también en una buena recomendación, a pesar de que recomendar "mal" en un rating 1 unidad menor, podría causar que el usuario sienta un poco de desilusión
2. **Usuario usuario**

```
# se crea un modelo knnbasic item-item con similitud coseno
sim_options = {'name': 'cosine',
               'user_based': False  # calcule similitud item-item
               }
algo = KNNBasic(k=20, min_k=2, sim_options=sim_options)
✓ 0.0s
```

FIGURA 14. CONFIGURACIÓN POR DEFECTO DEL MODELO

Al modificar el parámetro "user_based", para entrenar el algoritmo con una aproximación usuario-usuario, se observan resultados ligeramente mejores suponiendo la posibilidad de calificar películas usando decimales, sin embargo, al no ser el caso, ambos modelos podrían idénticos en práctica.

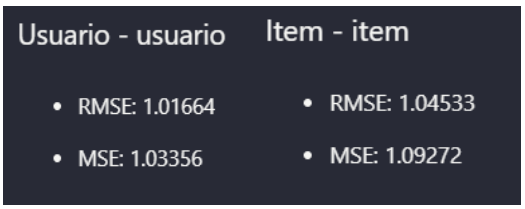

FIGURA 15. COMPARACIÓN RMSE ENTRE USUARIO-USUARIO E ITEM-ITEM

Luego, al cambiar la cantidad de vecinos se observa el mismo comportamiento de minúsculas mejorías. De todas maneras, se puede observar que aumentar el *k* sí resulta en una mejoría hasta cierto punto, en el que luego empeora rápidamente, mostrando que el modelo presenta sobreajuste al aumentar demasiado la cantidad de vecinos.

k	RMSE	MSE
10	1.03857	1.07863
20	1.01664	1.03356
50	1.01620	1.03267
75	1.05073	1.10404
100	1.05164	1.10594

FIGURA 16. TABLA, RMSE SEGÚN LA CANTIDAD DE VECINOS

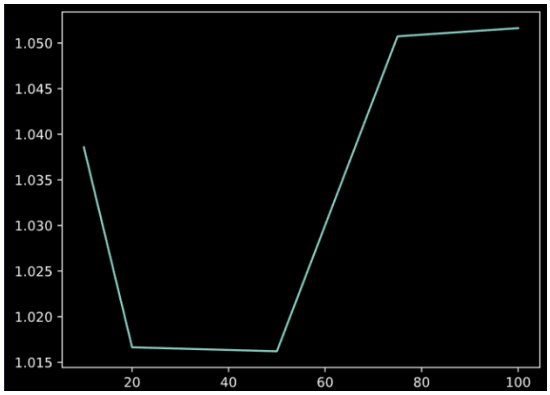

FIGURA 17. CANTIDAD DE VECINOS VS RMSE

4. Generando listas de predicciones para los usuarios

Para esto se generaron dos usuarios, uno que le gustaban las 106 películas más populares con un puntaje promedio, igual al esperado, y se esperaba que el recomendador diera las siguientes 10 películas más populares como recomendación.

```
new_user_data = {'user_id': [ratings['user_id'].max() + 1] * math.floor(num_items_rated.mean()),
                 'item_id': mainstream_items[0:math.floor(num_items_rated.mean())],
                 'rating': [int(round(rec_lb[rec_lb['item_id'] == item_id]['mean'].iloc[0])) for item_id in mainstream_items[0:math.floor(num_items_rated.mean())]]}
```

5. Otros frameworks de filtrado colaborativo

Estuvimos trabajando con la librería de lenskit y, aunque logramos reproducir los puntos del tutorial con nuestra base de datos, no nos fue posible reproducir las reproducciones para comparar el desempeño de ambos modelos. Encontramos muy poca documentación para resolver los errores / bugs que se generaban.

Aún así, logramos generar una lista de recomendaciones generalizada y que se puede especificar con usuario. Esta lista incluye las 100 recomendaciones mejor rankeadas según el modelo.

```
all_recs = pd.concat(all_recs, ignore_index=T
all_recs.head()
```

1] ✓ 0.0s

	item	score	user	rank	Algorithm
0	1449	5.154592	2	1	ItemItem
1	1639	5.014940	2	2	ItemItem
2	1594	4.952485	2	3	ItemItem
3	318	4.867934	2	4	ItemItem
4	169	4.793175	2	5	ItemItem

```
all_recs[(all_recs['user'] == 154)]
```

✓ 0.0s

	item	score	user	rank	Algorithm
3200	169	4.899779	154	1	ItemItem
3201	1449	4.832287	154	2	ItemItem
3202	127	4.707132	154	3	ItemItem
3203	513	4.688831	154	4	ItemItem
3204	474	4.660902	154	5	ItemItem
...	...	...	...	...	...
3295	656	4.204257	154	96	ItemItem
3296	1251	4.200665	154	97	ItemItem
3297	481	4.199653	154	98	ItemItem
3298	313	4.199425	154	99	ItemItem
3299	1021	4.194008	154	100	ItemItem

100 rows × 5 columns