

## Recomendaciones de películas a Usuario

método 1, binarios:

Interestelar
Mr. Nobody
Lucy
3 metros sobre el cielo
La la land
El conjuro
REC
Eso (It)
El exorcismo de Emily Rose
Coco
South park
Mr. Bean
Scary movie
Superbad
El laberinto del fauno

Las recomendaciones están basadas en las películas que el usuario 13 tiene bien calificadas y el usuario 1 no tiene bien evaluadas. Hay que mencionar que la película ‘Scary Movie’ ya había sido vista, sin embargo, la evaluación de la misma fue baja, aún así se recomienda esta al usuario1

Método 2, binarios:

Interestelar
Diario de una pasión (The notebook)
3 metros sobre el cielo
La la land
El conjuro
Eso (It)
El exorcismo de Emily Rose
Coco
South park
Mr. Bean
Una esposa de mentiras
Scary movie
El laberinto del fauno

En esta ocasión las sugerencias no vienen de un solo usuario, sino de los 5 más parecidos. Estas recomendaciones siguen incluyendo la película ‘Scary Movie’, pero omiten otras como ‘Super Bad’ y ‘Mr Nobody’

#### Método 1, Dummies:

E.T._4.0
Interestelar_5.0
Mr. Nobody_5.0
V de Vendetta_3.0
El Origen (Inception)_4.0
Matrix_3.0
Lucy_4.0
3 metros sobre el cielo_4.0
Titanic_4.0
La la land_5.0
Orgullo y prejuicio_1.0
El conjuro_3.0
La monja_2.0
REC_4.0
Eso (It)_4.0
El exorcismo de Emily Rose_4.0
Coco_4.0
Intensa-mente_5.0
Big hero 6 (Grandes heroes)_1.0
Shrek_5.0

En este punto se tenían dos opciones; la primera de ellas consistía en utilizar recomendaciones basadas en los emparejamientos de Dummies, sin embargo se rechazó este mecanismo debido a que recomienda prácticamente todas las películas en las que se haya tenido un resultado distinto

entre el recomendador y el recomendado. La segunda de ellas era utilizar las recomendaciones normales de la persona recomendada por el pareo de Dummies, en este sentido se recomendaron exactamente las mismas películas debido a que el usuario recomendador era el mismo en las dos ocasiones.

#### Método 2, Dummies:

Toy story_5.0
Star Wars_5.0
Spider-Man_5.0
South park_3.0
Shrek_5.0
Orgullo y prejuicio_1.0
Narnia_4.0
Interestelar_5.0
Intensa-mente_5.0
Harry Potter_5.0
Eso (It)_4.0
Diario de una pasión (The notebook)_5.0
Deadpool_5.0
Big hero 6 (Grandes heroes)_1.0

Para este método de recomendaciones sse encuentra un poco más de homogeneidad en las recomendaciones (la mayoría se recomiendan con 5 o 4 estrellas), sin embargo puede ser recomendada una misma película que se haya evaluado anteriormente con buena calificación. (suponga el caso de que se evalua Toy Story con 4 y aún así se le sugiere ver Toy Story hasta que se convenza de evaluarla en 5...) Caso similar ocurre con películas ‘orgullo y prejuicio’ y ‘Big hero 6’; películas que seguramente fueron evaluadas con una calificación superior a 1 y aún así se le sugiere verlas (con muy baja recomendación)

A simple vista no parece mejorar demasiado la recomendación de películas, sin embargo esto se puede deber a que no están siendo procesados de forma apropiada los resultados.

```

9 import pandas as pd
10 import numpy as np
11 import sklearn.metrics as skm
12 import scipy.spatial.distance as sc
13
14 #%% Leer datos
15 data = pd.read_excel('../data/Test de peliculas(1-16).xlsx', encoding='latin_1')
16
17 #%% Tomar solamente las columnas con calificación de películas.
18 pel = pd.DataFrame()
19 for i in range((len(data.T)-5)//3):
20     pel = pel.append(data.iloc[:,6+i*3])
21 pel = pel.T
22 print(pel)
23
24 #%% convertir a binarios
25 pel_n = pel.copy()
26 pel_n[pel_n<3] = 0
27 pel_n[pel_n>=3] = 1
28
29 #%% calcular Jaccard para los usuarios.
30 D1 = sc.squareform(sc.pdist(pel_n, 'jaccard'))
31
32 Isim1 = 1-D1
33
34 #%% Elegir usuario y tomar los usuarios más parecidos a él.
35 user = 1
36 Isim_user = Isim1[user]
37 indx_user1 = np.argsort(Isim_user)
38

```

```

39 #%% Recomendar las películas que el usuario no ha visto, pero el recomendador sí.
40 USER = pel_n.loc[user]
41 USER_sim1 = pel_n.loc[indx_user1[-2]]
42
43 indx_recomend1_1 = (USER_sim1==1)&(USER==0)
44 recomend1_1 = list(USER.index[indx_recomend1_1])
45
46 #%% Recomendar por segundo método
47 USER_sim1 = np.mean(pel_n.loc[indx_user1[-6:-1]],axis = 0)
48 USER_sim1[USER_sim1<=.5]=0
49 USER_sim1[USER_sim1>.5]=1
50
51 indx_recomend1_2 = (USER_sim1==1)&(USER==0)
52 recomend1_2 = list(USER.index[indx_recomend1_2])
53
54
55

```

```

62
63 ### Ahora con Dummies.
64 pel_d = pd.get_dummies(pel[pel.columns[0]],prefix=pel.columns[0])
65 for i in pel.columns[1:]:
66     tmp = pd.get_dummies(pel[i],prefix=i)
67     pel_d = pel_d.join(tmp)
68
69 ### Jaccard
70 D2 = sc.squareform(sc.pdist(pel_d, 'jaccard'))
71
72 Isim2 = 1-D2
73
74 ###
75 indx_user2 = np.argsort(Isim_user)
76
77 ###
78 USER = pel_d.loc[user]
79 USER_sim2 = pel_d.loc[indx_user2[-2]]
80
81 indx_recomend2_1 = (USER_sim2==1)&(USER==0)
82 recomend2_1 = list(USER.index[indx_recomend2_1])
83
84 ###
85 USER_sim2 = np.mean(pel_d.loc[indx_user2[-6:-1]],axis = 0)
86 USER_sim2[USER_sim2<=.5]=0
87 USER_sim2[USER_sim2>.5]=1
88
89 indx_recomend2_2 = (USER_sim2==1)&(USER==0)
90 recomend2_2 = list(USER.index[indx_recomend2_2])
91

```