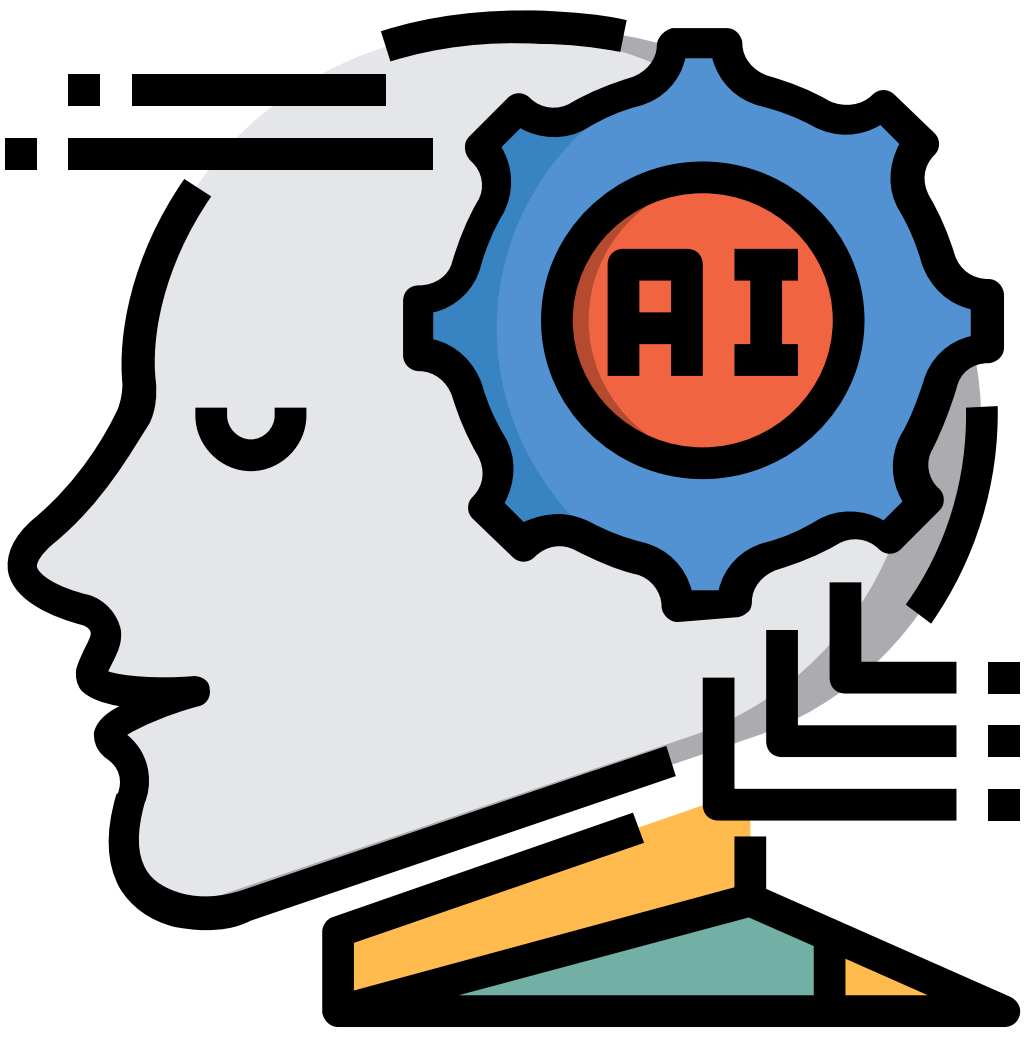
Universidad Nacional Autónoma de México
Facultad de Ingeniería

Inteligencia Artificial

PRÁCTICA 1. REGLAS DE ASOCIACIÓN

Casasola García Oscar
316123747
oscar.casasola.g7@gmail.com
Grupo 03



Profesor: Dr. Guillermo Gilberto Molero Castillo
Semestre 2022-1

Contenido

Objetivo

- ✓ Obtener reglas de asociación a partir de datos obtenidos de una plataforma de películas, donde los clientes pueden rentar o comprar este tipo de contenidos.

Características

- Por lo general, existe un patrón en lo que ven los clientes. Por ejemplo, superhéroes en la categoría para niños.
- En este sentido, se pueden generar más ganancias, si se puede identificar la relación entre las películas. Esto es, si las películas A y B se rentan juntas, este patrón se puede aprovechar para aumentar las ganancias.
- Las personas que rentan una de estas películas pueden ser empujadas a rentar o comprar la otra, a través de campañas o sugerencias dentro de la plataforma.
- En este sentido, cada vez es común familiarizarse con los motores de recomendación en Netflix, Amazon, por nombrar los más destacados.

Preparación del entorno de ejecución

1) Importar las bibliotecas necesarias

```
!pip install apyori # pip es un administrador de paquetes de Python. Se instala el paquete Apyori
```

```
import pandas as pd                # Para la manipulación y análisis de los datos
import numpy as np                 # Para crear vectores y matrices n dimensionales
import matplotlib.pyplot as plt    # Para la generación de gráficas a partir de los datos
from apyori import apriori
```

2) Importar los datos

Fuente de datos: movies.csv

```
from google.colab import files
files.upload()

DatosMovies = pd.read_csv('movies.csv')
DatosMovies #Visualizamos los datos cargados
```

	The Revenant	13 Hours	Allied	Zootopia	Jigsaw	Achorman	Grinch	Fast and Furious	Ghostbusters	Wolverine	Mad Max	John Wick	La La Land	The Good Dunosaur	Ninja Turtles	The Good Dunosaur Bad Moms	G
0	Beirut	Martian	Get Out	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	Deadpool	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	X-Men	Allied	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
3	Ninja Turtles	Moana	Ghost in the Shell	Ralph Breaks the Internet	John Wick	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
4	Mad Max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7454	Big Sick	Looper	Hulk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
7455	Beirut	Intern	Get Out	Hotel Transylvania	Mamma Mia	John Wick	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

Observaciones:

1) Se observa que el encabezado es la primera transacción.
2) NaN indica que esa película no fue rentada o comprada en esa transacción.

```
DatosMovies = pd.read_csv('movies.csv', header=None) #Los primeros datos nos lo toma como datos y no como un encabezado

DatosMovies
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	The Revenant	13 Hours	Allied	Zootopia	Jigsaw	Achorman	Grinch	Fast and Furious	Ghostbusters	Wolverine	Mad Max	John Wick	La La Land	The Good Dunosaur	Ninja Turtles	The Good Dunosaur Bad Moms	2 Guns	In
1	Beirut	Martian	Get Out	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	Deadpool	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	X-Men	Allied	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	Ninja Turtles	Moana	Ghost in the Shell	Ralph Breaks the Internet	John Wick	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7455	Big Sick	Looper	Hulk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7456	Beirut	Intern	Get Out	Hotel Transylvania	Mamma Mia	John Wick	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Procesamiento de los datos

Exploración de los ítems para contabilizarlos y mostrar la frecuencia

Antes de ejecutar el algoritmo es recomendable observar la distribución de la frecuencia de los elementos.

```
#Se incluyen todas las transacciones en una sola lista
Transacciones = DatosMovies.values.reshape(-1).tolist() #-1 significa 'dimensión desconocida' (recomendable) o: 7460*20=149200
Transacciones
```

```
['The Revenant',
 '13 Hours',
 'Allied',
 'Zootopia',
 'Jigsaw',
 'Achorman',
 'Grinch',
 'Fast and Furious',
 'Ghostbusters',
 'Wolverine',
 'Mad Max',
 'John Wick',
 'La La Land',
 'The Good Dunosaur',
 'Ninja Turtles',
 'The Good Dunosaur Bad Moms',
 '2 Guns',
 'Inside Out',
 'Valerian',
 'Spiderman 3',
 'Beirut',
 'Martian',
 'Get Out',
 nan,
 nan,
 nan,
```

```
#Se crea una matriz (dataframe) usando la lista y se incluye una columna 'Frecuencia'
ListaM = pd.DataFrame(Transacciones)
ListaM['Frecuencia'] = 0 #Valor temporal
ListaM
```

	0	Frecuencia
0	The Revenant	0
1	13 Hours	0
2	Allied	0
3	Zootopia	0
4	Jigsaw	0
...	...	...
149195	NaN	0
149196	NaN	0
149197	NaN	0
149198	NaN	0
149199	NaN	0

149200 rows × 2 columns

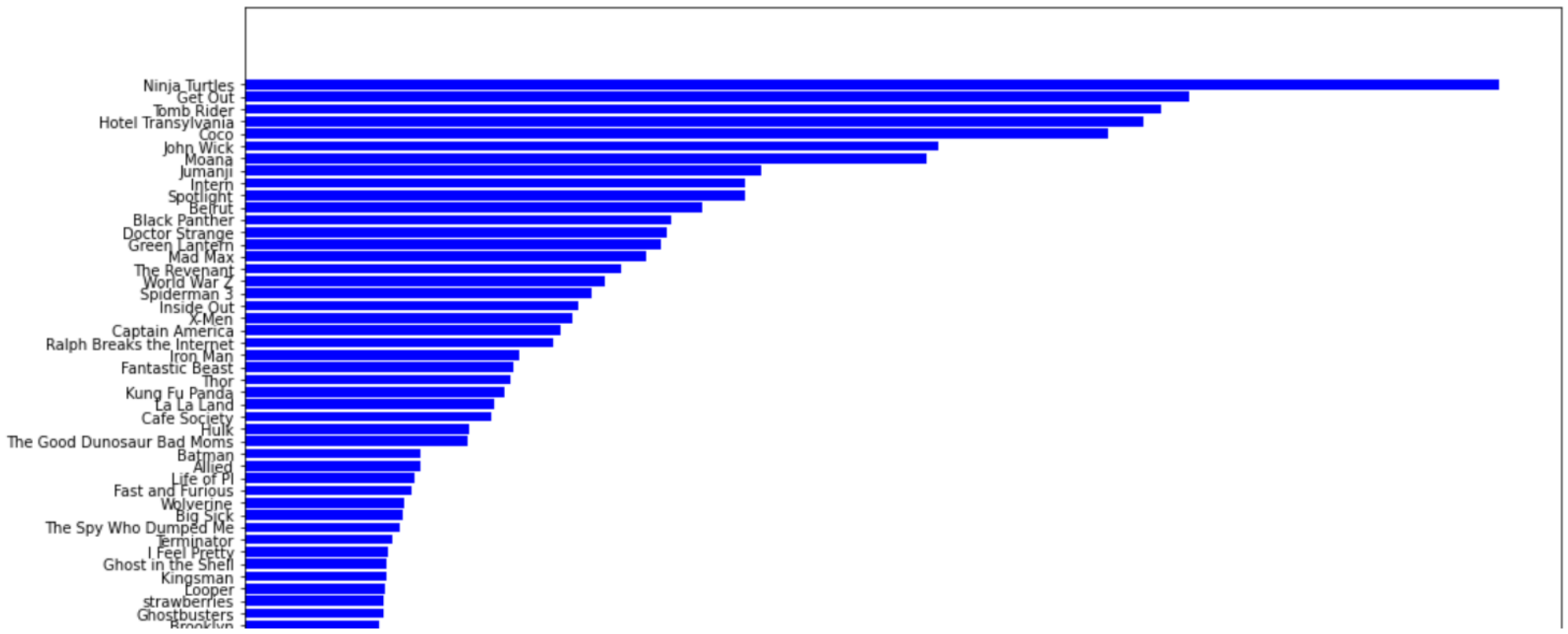
```
#Se agrupa los elementos
ListaM = ListaM.groupby(by=[0], as_index=False).count().sort_values(by=['Frecuencia'], ascending=True) #Conteo
ListaM['Porcentaje'] = (ListaM['Frecuencia'] / ListaM['Frecuencia'].sum()) #Porcentaje
ListaM = ListaM.rename(columns={0 : 'Item'})

#Se muestra la lista de las películas menos populares a las más populares
ListaM
```

	Item	Frecuencia	Porcentaje
106	Vampire in Brooklyn	3	0.000102
63	Lady Bird	5	0.000171
34	Finding Dory	7	0.000239
11	Bad Moms	14	0.000477
118	water spray	29	0.000989
...	...	...	...
25	Coco	1229	0.041915
44	Hotel Transylvania	1280	0.043655
103	Tomb Rider	1305	0.044507
37	Get Out	1346	0.045906
75	Ninja Turtles	1786	0.060912

119 rows × 3 columns

```
# Generamos un gráfico de barras
plt.figure(figsize=(16,20))
plt.ylabel('Item')
plt.xlabel('Frecuencia')
plt.barh(ListaM['Item'], width=ListaM['Frecuencia'], color='blue')
plt.show()
```



Preparación

La función Apriori de Python requiere que el conjunto de datos tenga la forma de una lista de listas, donde cada transacción es una lista interna dentro de una gran lista.

Los datos actuales están en un dataframe de Pandas, por lo que, se requiere convertir en una lista.

```
#Se crea una lista de listas a partir del dataframe y se remueven los 'NaN'
#level=0 especifica desde el primer índice
MoviesLista = DatosMovies.stack().groupby(level=0).apply(list).tolist()
MoviesLista
```

```
 'Spotlight'],
['Jumanji', 'Ninja Turtles', 'Moana', 'Get Out', "Olalf's Frozen Adventure"],
['The Revenant', 'Avengers', 'John Wick'],
['Inside Out'],
['Kung Fu Panda', 'Intern', 'Ninja Turtles', 'Pop Star', 'Brooklyn'],
['X-Men',
 'World War Z',
 'Tomb Rider',
 'Moana',
 'Despicable Me',
 'Get Out',
 'La La Land',
 'Black Panther',
 'John Wick',
 'Hotel Transylvania',
 'Batman',
 'Wolverine'],
['Kung Fu Panda',
 'Terminator',
 'Tomb Rider',
 'Get Out',
 'Ralph Breaks the Internet',
 'Coco',
 'Hotel Transylvania',
 'Green Lantern',
 'Doctor Strange'],
['John Wick'],
['Iron Man'
```

Aplicación del algoritmo

Configuración 1

Obtener reglas para aquellas películas que se hayan rentado al menos 10 veces en un día (70 veces en una semana):

i) El soporte mínimo se calcula de 70/7460 = 0.00938 (1%).
ii) La confianza mínima para las reglas de 30%
iii) La elevación de 2.

Observación: Estos valores se eligen arbitrariamente, por lo que, se recomienda probar valores y analizar la diferencia en las reglas.

```
ReglasC1 = apriori(MoviesLista,
                   min_support=0.01,
                   min_confidence=0.3,
                   min_lift=2)
```

Se convierte las reglas encontradas por la clase apriori en una lista, puesto que es más fácil ver los resultados.

```
ResultadosC1 = list(ReglasC1)
print(len(ResultadosC1)) #Total de reglas encontradas(9)
ResultadosC1
```

```
[RelationRecord(items=frozenset({'Jumanji', 'Kung Fu Panda'}), support=0.0160857908847185, ordered_statistics=[OrderedStatistic(items_base=frozenset({'
 RelationRecord(items=frozenset({'Jumanji', 'Tomb Rider'}), support=0.03941018766756032, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Ju
 RelationRecord(items=frozenset({'Moana', 'Thor'}), support=0.015281501340482574, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Thor'}),
 RelationRecord(items=frozenset({'Tomb Rider', 'Terminator'}), support=0.01032171581769437, ordered_statistics=[OrderedStatistic(items_base=frozenset({
 RelationRecord(items=frozenset({'Jumanji', 'Ninja Turtles', 'Get Out'}), support=0.010187667560321715, ordered_statistics=[OrderedStatistic(items_base
 RelationRecord(items=frozenset({'Moana', 'Ninja Turtles', 'Intern'}), support=0.011126005361930294, ordered_statistics=[OrderedStatistic(items_base=fr
 RelationRecord(items=frozenset({'Jumanji', 'Ninja Turtles', 'Moana'}), support=0.011126005361930294, ordered_statistics=[OrderedStatistic(items_base=f
 RelationRecord(items=frozenset({'Jumanji', 'Ninja Turtles', 'Tomb Rider'}), support=0.017158176943699734, ordered_statistics=[OrderedStatistic(items_b
 RelationRecord(items=frozenset({'Ninja Turtles', 'Tomb Rider', 'Spiderman 3'}), support=0.01032171581769437, ordered_statistics=[OrderedStatistic(iten
```

```
pd.DataFrame(ResultadosC1)
```

	items	support	ordered_statistics
0	(Jumanji, Kung Fu Panda)	0.016086	[((Kung Fu Panda), (Jumanji), 0.32345013477088...
1	(Jumanji, Tomb Rider)	0.039410	[((Jumanji), (Tomb Rider), 0.3994565217391304,...
2	(Moana, Thor)	0.015282	[((Thor), (Moana), 0.3007915567282322, 2.31092...
3	(Tomb Rider, Terminator)	0.010322	[((Terminator), (Tomb Rider), 0.36492890995260...
4	(Jumanji, Ninja Turtles, Get Out)	0.010188	[((Jumanji, Get Out), (Ninja Turtles), 0.50666...
5	(Moana, Ninja Turtles, Intern)	0.011126	[((Ninja Turtles, Intern), (Moana), 0.30970149...
6	(Jumanji, Ninja Turtles, Moana)	0.011126	[((Jumanji, Moana), (Ninja Turtles), 0.5030303...
7	(Jumanji, Ninja Turtles, Tomb Rider)	0.017158	[((Jumanji, Ninja Turtles), (Tomb Rider), 0.41...
8	(Ninja Turtles, Tomb Rider, Spiderman 3)	0.010322	[((Ninja Turtles, Spiderman 3), (Tomb Rider), ...

A partir de la siguiente tabla, se puede concluir que hay una correlación entre películas, ya que existen usuarios que compraron o compraron la película de Jumanji y Kung Fu Panda, Jumanji y Tomb Rider, Moana y Thor, Tomb Rider y Terminator, y así sucesivamente. Esto es lógico ya que estas películas tienen una relación de género, por ejemplo, las películas de Jumanji y Kung Fu Panda pertenecen al género infantil, por lo que si una persona ve la película de Jumanji es más probable que se también vea la de Kung Fu Panda y viceversa.

Son 9 reglas. A manera de ejemplo se imprime la primera regla:

```
print(ResultadosC1[0])
```

Primera regla:

RelationRecord(items=frozenset({'Jumanji', 'Kung Fu Panda'}), support=0.0160857908847185, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Kung Fu Panda'}), items_add=frozenset({'Jumanji'}), confidence=0.3234501347708895, lift=3.2784483768897226)])

Presentando los datos:

```
for item in ResultadosC1:
  #El primer índice de la lista
  Emparejar = item[0]
  items = [x for x in Emparejar]
  print("Regla: " + str(item[0]))

  #El segundo índice de la lista
  print("Soporte: " + str(item[1]))

  #El tercer índice de la lista
  print("Confianza: " + str(item[2][0][2]))
  print("Lift: " + str(item[2][0][3]))
  print("====================================")
```

Número de regla	Regla	Soporte	Confianza	Lift
1	('Kung Fu Panda', 'Jumanji')	1.6085790884718498 %	32.345013477088955 %	3.2784483768897226
2	('Tomb Rider', 'Jumanji')	3.9410187667560317 %	39.94565217391304 %	2.283483258370814
3	('Moana', 'Thor')	1.5281501340482573 %	30.07915567282322 %	2.3109217437617016
4	('Tomb Rider', 'Terminator')	1.032171581769437 %	36.492890995260666 %	2.0861070254762035
5	(' Ninja Turtles ', 'Get Out', 'Jumanji')	1.0187667560321716 %	50.66666666666666 %	2.1163120567375886
6	('Intern', 'Moana', 'Ninja Turtles')	1.1126005361930293 %	30.970149253731343 %	2.37937500960696
7	('Moana', 'Ninja Turtles', 'Jumanji')	1.1126005361930293 %	50.3030303030303 %	2.1011232142251175
8	('Ninja Turtles', 'Tomb Rider', 'Jumanji')	1.7158176943699734 %	41.69381107491857 %	2.383416326581552
9	('Ninja Turtles', 'Spiderman 3', 'Tomb Rider')	1.032171581769437 %	37.19806763285024 %	2.1264182723453087

Conclusiones Configuración 1

- ✓ **Primera regla**

La primera regla contiene dos elementos: **'Kung Fu Panda'** y **'Jumanji'** que se vieron juntos.

Esto tiene sentido, las personas que ven películas familiares, en este caso de corte infantil, suelen ver también más películas del mismo tipo, como Kung Fu Panda (2016) y Jumanji (2017).

El soporte es de 0.016 (1.6%), la confianza de 0.32 (32%) y elevación de 3.27, esto representa que existe 3 veces más posibilidades de que los que vean Kung Fu Panda miren también Jumanji, o viceversa. Es decir, si una persona ve Kung Fu Panda, es más probable que se le recomiende Jumanji.

- ✓ **Segunda regla**

Para la segunda regla se tienen dos elementos: **'Tomb Rider'** y **'Jumanji'**, el cual se vieron juntos.

Esto tiene sentido, porque personas que ven películas del género de Acción y Aventura, suelen ver también más películas del mismo tipo, como Tomb Rider y Jumanji.

El soporte es de 0.0394 (3.94%), la confianza de 0.3994 (39.94%) y elevación de 2.2834, esto representa que existe 2 veces más posibilidades de que los que vean Tomb Rider miren también Jumanji, o viceversa. Es decir, si una persona ve Tomb Rider, es más probable que se le recomiende Jumanji.

Siguiendo este mismo patrón podemos seguir deduciendo las reglas:

- ✓ **Cuarta regla**

Para la cuarta regla contiene dos elementos: **'Tomb Rider'** y **'Terminator'**. Esto tiene sentido porque las personas que ven películas del género de Acción y Ciencia Ficción (Tomb Rider), también tienen gustos por las películas de géneros similares, como Terminator, que pertenece a los mismos géneros.

En esta regla, el soporte es de 0.0103 (1.03 %), la confianza de 0.3649 (36.49 %) y una elevación de 2.08, esto representa que existen 2 veces más probabilidades de que las personas que vean Tomb Rider miren también Terminator, o viceversa.

De esta manera y siguiendo el mismo patrón, se pueden sacar conclusiones para cada una de las nueve reglas mostradas en esta configuración.

Configuración 2

Obtener reglas para aquellas películas que se hayan rentado al menos 30 veces en un día (210 por semana):

i) El soporte mínimo se calcula de 210/7460 = 0.028 (2.8%).
ii) La confianza mínima para las reglas de 30%.
iii) La elevación de al menos 1.

Observación: Estos valores se eligen arbitrariamente, por lo que, se recomienda probar valores y analizar la diferencia en las reglas.

```
ReglasC2 = apriori(MoviesLista,
                   min_support=0.028,
                   min_confidence=0.3,
                   min_lift=1.1)
```

Se convierte las reglas encontradas por la clase apriori en una lista, puesto que es más fácil ver los resultados.

```
ResultadosC2 = list(ReglasC2)
print(len(ResultadosC2)) #Total de reglas encontradas(8)
ResultadosC2
```

```
[RelationRecord(items=frozenset({'Beirut', 'Get Out'}), support=0.028954423592493297, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Beiru
 RelationRecord(items=frozenset({'Coco', 'Ninja Turtles'}), support=0.05294906166219839, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Co
 RelationRecord(items=frozenset({'Ninja Turtles', 'Intern'}), support=0.035924932975871314, ordered_statistics=[OrderedStatistic(items_base=frozenset({
 RelationRecord(items=frozenset({'Jumanji', 'Ninja Turtles'}), support=0.04115281501340483, ordered_statistics=[OrderedStatistic(items_base=frozenset({
 RelationRecord(items=frozenset({'Jumanji', 'Tomb Rider'}), support=0.03941018766756032, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Ju
 RelationRecord(items=frozenset({'Moana', 'Ninja Turtles'}), support=0.04825737265415549, ordered_statistics=[OrderedStatistic(items_base=frozenset({'N
 RelationRecord(items=frozenset({'Ninja Turtles', 'Spotlight'}), support=0.0339142091152815, ordered_statistics=[OrderedStatistic(items_base=frozenset(
 RelationRecord(items=frozenset({'Ninja Turtles', 'Tomb Rider'}), support=0.060053619302949064, ordered_statistics=[OrderedStatistic(items_base=frozens
```

```
pd.DataFrame(ResultadosC2)
```

	items	support	ordered_statistics
0	(Beirut, Get Out)	0.028954	[((Beirut), (Get Out), 0.3312883435582822, 1.8...
1	(Coco, Ninja Turtles)	0.052949	[((Coco), (Ninja Turtles), 0.32166123778501626...
2	(Ninja Turtles, Intern)	0.035925	[((Intern), (Ninja Turtles), 0.375350140056022...
3	(Jumanji, Ninja Turtles)	0.041153	[((Jumanji), (Ninja Turtles), 0.41711956521739...
4	(Jumanji, Tomb Rider)	0.039410	[((Jumanji), (Tomb Rider), 0.3994565217391304,...
5	(Moana, Ninja Turtles)	0.048257	[((Moana), (Ninja Turtles), 0.3707518022657054...
6	(Ninja Turtles, Spotlight)	0.033914	[((Spotlight), (Ninja Turtles), 0.355337078651...
7	(Ninja Turtles, Tomb Rider)	0.060054	[((Tomb Rider), (Ninja Turtles), 0.34329501915...

A partir de la siguiente tabla, se puede concluir que hay una correlación entre películas, ya que existen usuarios que compraron o compraron la película de Beirut y Get Out, Coco y Ninja Turtles, Ninja Turtles e Intern, Jumanji y Ninja Turtles, y así sucesivamente. Esto es lógico ya que estas películas tienen una relación de género, por ejemplo, las películas de Beirut y Get Out pertenecen al género de suspenso, por lo que si una persona ve la película de Beirut es más probable que se también vea la de Get Out y viceversa.

Son 8 reglas. A manera de ejemplo se imprime la primera regla:

```
print(ResultadosC2[0])
```

Primera regla:

RelationRecord(items=frozenset({'Beirut', 'Get Out'}), support=0.028954423592493297, ordered_statistics=[OrderedStatistic(items_base=frozenset({'Beirut'}), items_add=frozenset({'Get Out'}), confidence=0.3312883435582822, lift=1.836115187923317 3)])

Presentando los datos:

```
for item in ResultadosC2:
  #El primer índice de la lista
  Emparejar = item[0]
  items = [x for x in Emparejar]
  print("Regla: " + str(item[0]))

  #El segundo índice de la lista
  print("Soporte: " + str(item[1]))

  #El tercer índice de la lista
  print("Confianza: " + str(item[2][0][2]))
  print("Lift: " + str(item[2][0][3]))
  print("====================================")
```

Número de regla	Regla	Soporte	Confianza	Lift
1	('Beirut', 'Get Out')	2.89544235924933 %	33.12883435582822 %	1.8361151879233173
2	('Coco', 'Ninja Turtles')	5.294906166219839 %	32.16612377850163 %	1.3435570178478284
3	('Ninja Turtles', 'Intern')	3.5924932975871315 %	37.53501400560224 %	1.5678118951948081
4	('Jumanji', 'Ninja Turtles')	4.115281501340482 %	41.71195652173913 %	1.742279930863236
5	('Jumanji', 'Tomb Rider')	3.9410187667560317 %	39.94565217391304 %	2.283483258370814
6	('Moana', 'Ninja Turtles')	4.825737265415549 %	37.075180226570545 %	1.5486049523528347
7	('Ninja Turtles', 'Spotlight')	3.39142091152815 %	35.53370786516854 %	1.4842187047825157
8	('Ninja Turtles', 'Tomb Rider')	6.005361930294907 %	34.32950191570881 %	1.4339198448554744

Conclusiones Configuración 2

- ✓ **Primera regla**

La primera regla contiene dos elementos: **'Get Out'** y **'Beirut'** que se vieron juntos.

Esto también tiene sentido, ya que las personas que ven películas de espionaje, como Beirut (2018), tienen gustos afines con películas de terror, como Get Out (2017).

El soporte es de 0.028 (2.8%), la confianza de 0.33 (33%) y una elevación de 1.83, esto representa que existe casi 2 veces más probabilidades de que los que vean Get Out miren también Beirut, o viceversa.

- ✓ **Segunda regla**

Para la segunda regla se tienen dos elementos: **'Coco'** y **'Ninja Turtles'**, el cual se vieron juntos.

Esto tiene sentido, porque personas que ven películas del género de Acción y Aventura, suelen ver también más películas del mismo tipo, como Coco y Ninja Turtles.

El soporte es de 0.0529 (5.29%), la confianza de 0.3216 (32.16%) y elevación de 1.3435, por ende se puede observar que las posibilidades de que las personas que vean Coco y que miren también las Tortugas Ninja, son más de una.

Siguiendo este mismo patrón podemos seguir deduciendo las reglas:

- ✓ **Cuarta regla**

La cuarta regla contiene dos elementos: **'Jumanji'** y **'Ninja Turtles'**. Esto tiene sentido porque las personas que ven películas del género de Aventura y Comedia (Jumanji), también tienen gustos por las películas de géneros similares, como las Tortugas Ninja que pertenece a los géneros de Acción y Aventura.

En esta regla, el soporte es de 0.041 (4.1 %), la confianza de 0.417 (41.7 %) y una elevación de 1.74, esto representa que existe casi 2 veces más probabilidades de que las personas que vean Jumanji miren también las Tortugas Ninja, o viceversa.

De esta manera se puede seguir el mismo patrón y sacar conclusiones para cada una de las ocho reglas mostradas en esta configuración.

Link de Google Colab

OCG-Práctica1-RAsociación.ipynb - Colaboratory (google.com)