Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 02/03/2022
Practica No.2 Reglas de Asociación (Algoritmo Apriori)

Objetivo.

Analizar las transacciones y obtener reglas significativas (patrones) de los productos vendidos en un comercio minorista en Francia. Los datos son transacciones de un comercio de un periodo de una semana (7 días).

Características.

- Ítems (120 productos).
- 7500 transacciones.

Las reglas de asociación sirven para determinar la posibilidad de que a un antecedente (una compra, una reproducción de video, etc) le siga determinado consecuente del mismo tipo. Para ello definimos tres principales parámetros los cuales son.

- Soporte mínimo $support = \frac{Frecuency(A,B)}{N}$
- Confianza mínima $confidence = \frac{Frecuency(A,B)}{Frecuency(A)}$
- Elevación mínima $lift = \frac{support}{support(A)\ x\ support(B)}$

Desarrollo.

En esta práctica a diferencia de la anterior se analizarán datos transaccionales correspondientes a la compra de artículos en un supermercado, los cuales lucen de la siguiente manera después de importarlos en un inicio.

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN

Todos estos datos deben pasar por una etapa de procesamiento donde se explora como es que se encuentran distribuidos los datos con base en su frecuencia de aparición para esto se tiene que contabilizar en cada transacción cuantas veces aparece cada uno de los artículos, después asignar a cada uno de los artículos su porcentaje de aparición basándonos en su frecuencia previamente calculada y el número total de datos por cada transacción, es decir de compras, dentro del conjunto de datos.

Por ejemplo, para la transacción con identificador 72 “Mineral Water” se tiene una Frecuencia de 1788, es decir que dentro de las transacciones de este conjunto de datos el agua mineral se vendió 1788 veces. Además, cuenta con un porcentaje de 0.06 el cual resulta de sacar la relación entre su frecuencia y el número total de transacciones (1788 / 29363 = 0.06).

Numero total de compras: 29363

	Item	Frecuencia	Porcentaje
0	asparagus	1	0.000034
112	water spray	3	0.000102
77	napkins	5	0.000170
34	cream	7	0.000238
11	bramble	14	0.000477
...	...	...	...
25	chocolate	1230	0.041889
43	french fries	1282	0.043660
100	spaghetti	1306	0.044478
37	eggs	1348	0.045908
72	mineral water	1788	0.060893

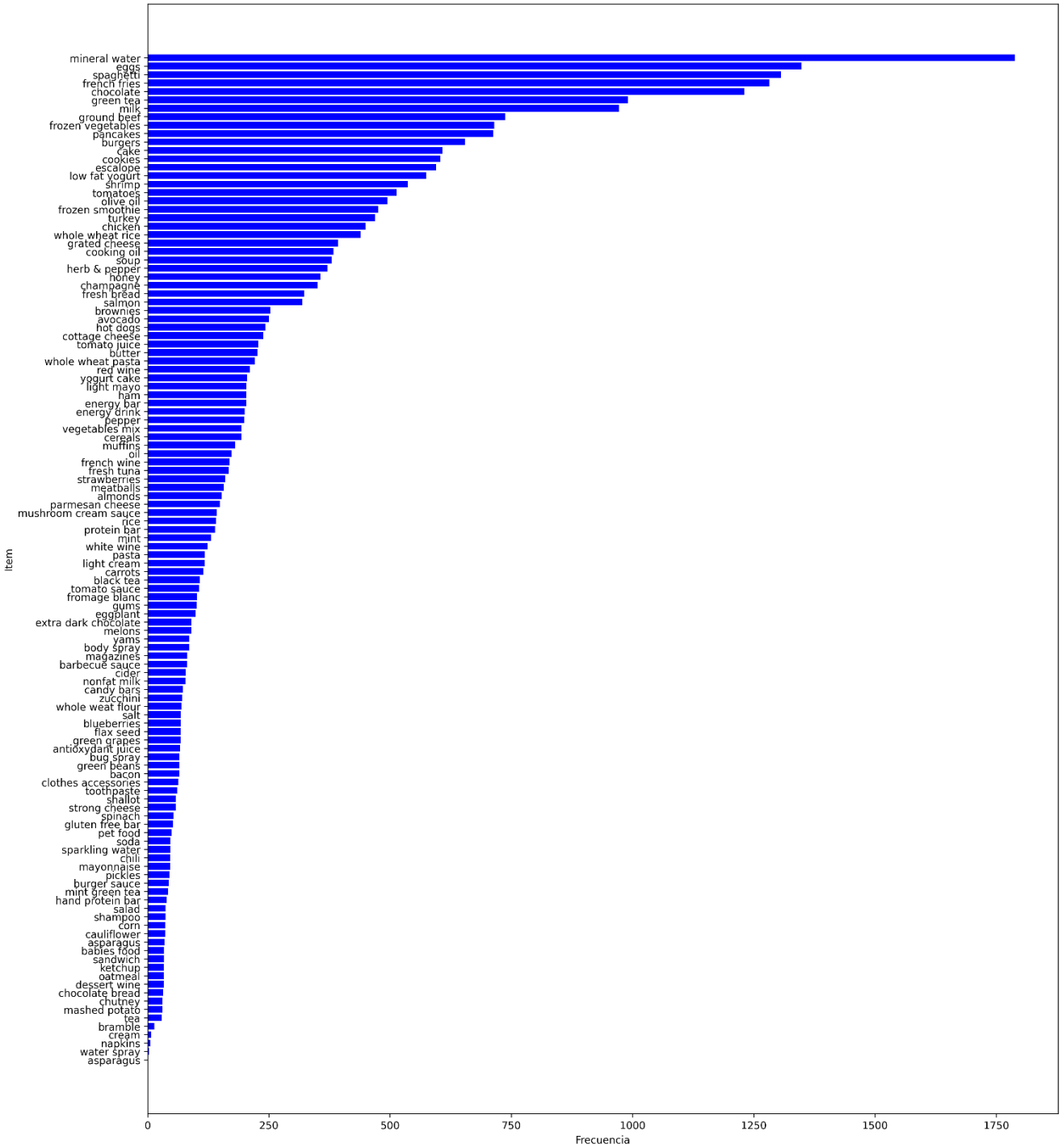
120 rows × 3 columns

Basándonos en esta tabla de frecuencias se grafico con ayuda de la librería `matplotlib` una grafica de barras horizontal donde pudiéramos apreciar un poco mejor como es que se vendieron cada uno de estos artículos. En ella podemos ver que los artículos mas vendidos son el agua mineral, huevos y spaghetti por lo cual es muy probable que veamos estos artículos dentro de nuestras reglas, aunque esto no quiere decir que los veamos juntos, mientras que los que se vendieron menos son los espárragos, los rociadores de agua y las servilletas de tela lo cual representa que muy difícilmente veremos estos artículos dentro de las reglas de asociación que se presenten con el algoritmo Apriori.

La grafica de frecuencias se muestra en la siguiente página.

Cervantes Rubí Brandon

Inteligencia Artificial 2022-2, G.3, 02/03/2022
Practica No.2 Reglas de Asociación (Algoritmo Apriori)



En este momento los datos son un `dataframe` de pandas para facilidad a la hora de mostrar estos datos, sin embargo, el algoritmo `apriori` acepta como entrada una lista de listas representando los artículos por cada transacción por lo cual hay que hacer una transformación a estos datos de la siguiente manera.

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

Con esto ahora es posible la aplicación del algoritmo mandando a llamar a la función apriori que se importó previamente y pasándole como parámetros la lista de listas de transacciones que se leyeron, el soporte mínimo, la confianza mínima y el nivel mínimo de elevación que cada una de las reglas a ser considerada deberá de cumplir para poder sobrevivir a la poda.

Configuración 1.

Se nos pide considerar artículos que hayan sido comprados por lo menos 35 veces en una semana, es decir que aparezcan por lo menos 35 veces dentro de las 29367 transacciones que se hicieron, un numero de compras que la mayoría de los artículos cumplen, esto representa un soporte mínimo del 0.45%. Además de una confianza mínima del 20% y un nivel de elevación igual o mayor a 3.

Con esto se obtuvieron 24 reglas significativas. Al tener un soporte mínimo tan bajo, esto da lugar a que muchas reglas puedan ser generadas por el algoritmo y bajo este criterio podemos ser un poco mas exigentes incrementando el índice de elevación colocándolo en 30% para quedarnos solamente con las reglas que incrementen más las posibilidades de compras cruzadas.

	items	support	ordered_statistics
0	(chicken, light cream)	0.004533	[((light cream), (chicken), 0.2905982905982905...
1	(escalope, mushroom cream sauce)	0.005733	[((mushroom cream sauce), (escalope), 0.300699...
2	(pasta, escalope)	0.005866	[((pasta), (escalope), 0.3728813559322034, 4.7...
3	(herb & pepper, ground beef)	0.015998	[((herb & pepper), (ground beef), 0.3234501347...

Analizando la primera regla se tiene que los usuarios que compren **pollo** también compren **crema ligera**, lo cual tiene sentido ya que se puede ver un perfil saludable. En lugar de comprar carnes rojas compran pollo y en lugar de comprar mayonesa o crema entera prefieren comprar crema ligera junto con el pollo.

Por el lado de los índices que nos arroja el algoritmo tenemos un soporte de 0.45% que indica que tan importante es la regla dentro del conjunto de datos cumpliendo con el índice mínimo que pasamos como parámetro, una confianza del 29% indicando que se trata de una regla fiable debido a que se encuentra por encima del 20% que se pasó como parámetro y una elevación de 4.8 indicando una relación positiva entre estos dos artículos además de un incremento en las posibilidades en casi 5 veces de que si se compra pollo también se comprara crema ligera y viceversa.

Configuración 2.

Para esta ocasión se decidió subir el índice de compra diaria a por lo menos 30 veces en un día, es decir 210 veces en una semana resultando en un soporte mínimo de 2.8%, además de una confianza mínima del 25% y una elevación apenas por encima del indicativo de relación positiva, es decir de 1.01. Se obtuvieron 10 reglas significativas las cuales cumplieron con estos valores mostrándose a continuación.

	items	support	ordered_statistics
0	(burgers, eggs)	0.028796	[((burgers), (eggs), 0.33027522935779813, 1.83...
1	(mineral water, chocolate)	0.052660	[((chocolate), (mineral water), 0.321399511798...
2	(eggs, mineral water)	0.050927	[((eggs), (mineral water), 0.28338278931750743...
3	(mineral water, frozen vegetables)	0.035729	[((frozen vegetables), (mineral water), 0.3748...
4	(mineral water, ground beef)	0.040928	[((ground beef), (mineral water), 0.4165535956...
5	(ground beef, spaghetti)	0.039195	[((ground beef), (spaghetti), 0.39891451831750...
6	(mineral water, milk)	0.047994	[((milk), (mineral water), 0.3703703703703704,...
7	(spaghetti, milk)	0.035462	[((milk), (spaghetti), 0.27366255144032925, 1....
8	(mineral water, pancakes)	0.033729	[((pancakes), (mineral water), 0.3548387096774...
9	(mineral water, spaghetti)	0.059725	[((mineral water), (spaghetti), 0.250559284116...

Analizando la primera de ellas tenemos que las personas que compran **hamburguesas** también son propensas a comprar **huevos**, a lo que le podemos ver sentido ya que son comidas de preparación rápida. Los índices que arroja esta regla son con un soporte de 2.8 que es justo el limite que se ingreso representando la importancia requerida dentro del conjunto de datos, una confianza del 33% lo cual indica que es una regla fiable y por último una elevación de 1.83 que representa un incremento en las posibilidades a casi el doble para que las personas que compren hamburguesas también compren huevos y viceversa.

Propuesta de Configuración 3.

Para esta configuración se decidió bajar el soporte mínimo con respecto a la configuración anterior a solo 10 compras por día, es decir 70 a la semana resultando en un soporte mínimo de 0.93%, una confianza mínima de 30% para quedarnos con reglas fiables dentro del conjunto y una elevación mínima de 2, es decir que incremente en 2 las posibilidades de compra cruzada para las reglas de asociación que se generen con el algoritmo.

Se obtuvieron 13 reglas las cuales se pueden ver a continuación.

	items	support	ordered_statistics
0	(herb & pepper, ground beef)	0.015998	[((herb & pepper), (ground beef), 0.3234501347...
1	(ground beef, spaghetti)	0.039195	[((ground beef), (spaghetti), 0.39891451831750...
2	(soup, milk)	0.015198	[((soup), (milk), 0.3007915567282322, 2.321231...
3	(whole wheat pasta, milk)	0.009865	[((whole wheat pasta), (milk), 0.3348416289592...
4	(pepper, spaghetti)	0.009865	[((pepper), (spaghetti), 0.37185929648241206, ...
5	(spaghetti, red wine)	0.010265	[((red wine), (spaghetti), 0.36492890995260663...
6	(eggs, mineral water, ground beef)	0.010132	[((eggs, ground beef), (mineral water), 0.5066...
7	(mineral water, frozen vegetables, milk)	0.011065	[((mineral water, frozen vegetables), (milk), ...
8	(mineral water, ground beef, milk)	0.011065	[((ground beef, milk), (mineral water), 0.5030...
9	(ground beef, spaghetti, milk)	0.009732	[((ground beef, milk), (spaghetti), 0.44242424...
10	(mineral water, ground beef, spaghetti)	0.017064	[((mineral water, ground beef), (spaghetti), 0...
11	(mineral water, olive oil, spaghetti)	0.010265	[((mineral water, olive oil), (spaghetti), 0.3...
12	(tomatoes, mineral water, spaghetti)	0.009332	[((tomatoes, mineral water), (spaghetti), 0.38...

Analizando la primera de ellas tenemos “herb & pepper”, es decir pimienta con hierbas, y “ground beef”, que se traduce como carne molida. Lo cual tiene sentido ya que normalmente este tipo de carne no se prepara sola y se le pone, entre otras cosas, pimienta para mejorar su sabor y su olor.

En cuanto a los índices que nos arroja nuestro algoritmo tenemos un soporte de 1.6% indicando que se trata de una regla importante dentro de nuestro análisis ya que supera por mucho el índice de 0.93% solicitado, es decir que se vende mas de lo que habíamos considerado como mínimo. Además, se cuenta con una confianza del 32% confirmando la fiabilidad de la regla y por último una elevación de 3.29%, índice que indica un incremento en las posibilidades de mas del triple de que si se compra pimienta con hierbas también se compre carne molida y viceversa.

Conclusiones.

En conclusión, esta práctica ayudo a reforzar los conocimientos adquiridos para emplear el algoritmo apriori obteniendo reglas de asociación confiables dentro de un conjunto de datos transaccionales de los productos vendidos de un supermercado en 7 días. Para ello debemos de definir 3 parámetros, el primero es el soporte basado en el número de artículos vendidos a la semana que queremos en nuestras reglas, un nivel de confianza que mientras mas alto sea indicara que la regla es más fiable, sin embargo, estos índices normalmente se encuentran por encima del 20% y llegando a mas de 40% para algunas reglas generadas para este conjunto de datos. Por último, está la elevación donde para este caso nos podemos poner un poco mas exigentes que en la práctica anterior y subir el índice por encima de 2 indicando el incremento al doble de las posibilidades debido a que estamos tratando con artículos los cuales van a tener un alto grado de redundancia, es decir, que se van a repetir mucho las compras ya que hablamos de un supermercado de venta general. Además de que la implementación de este tipo de reglas en datos transaccionales se vuelve sencilla gracias a las bibliotecas desarrolladas para Python.