Contenido

[Objetivo 1](#_Toc84105109)

[Características 1](#_Toc84105110)

[Preparación del entorno de ejecución 1](#_Toc84105111)

[1) Importar las bibliotecas necesarias 1](#_Toc84105112)

[2) Importar los datos 1](#_Toc84105113)

[Procesamiento de los datos 2](#_Toc84105114)

[Aplicación del algoritmo 4](#_Toc84105115)

[Configuración 1 4](#_Toc84105116)

[Conclusiones Configuración 1 5](#_Toc84105117)

[Configuración 2 6](#_Toc84105118)

[Conclusiones Configuración 2 7](#_Toc84105119)

[Link de Google Colab 7](#_Toc84105120)

# 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 (20 productos)
* 7500 transacciones

# Preparación del entorno de ejecución

## 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

## Importar los datos

Fuente de datos: store\_data.csv

from google.colab import files

files.upload()

DatosTransacciones = pd.read\_csv('store\_data.csv')

DatosTransacciones #Visualizamos los datos cargados

Calendario

Descripción generada automáticamente

**Observaciones:**

1. Se observa que el encabezado es la primera transacción.
2. NaN indica que ese producto no fue comprado en esa transacción.

DatosTransacciones = pd.read\_csv('store\_data.csv', header=None) #Los primeros datos nos lo toma como datos y no como un encabezado

DatosTransacciones.head(5)

Imagen de la pantalla de una computadora

Descripción generada automáticamente con confianza media

# 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 = DatosTransacciones.values.reshape(-1).tolist() #-1 significa 'dimensión desconocida'

#Se crea una matriz (dataframe) usando la lista y se incluye una columna 'Frecuencia'

Lista = pd.DataFrame(Transacciones)

Lista['Frecuencia'] = 1

#Se agrupa los elementos

Lista = Lista.groupby(by=[0], as\_index=False).count().sort\_values(by=['Frecuencia'], ascending=True) #Conteo

Lista['Porcentaje'] = (Lista['Frecuencia'] / Lista['Frecuencia'].sum()) #Porcentaje

Lista = Lista.rename(columns={0 : 'Item'})

#Se muestra la lista de los elementos que menos aparecen a los que más aparecen.

Lista

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

# Se genera un gráfico de barras

plt.figure(figsize=(16,20), dpi=300)

plt.ylabel('Item')

plt.xlabel('Frecuencia')

plt.barh(Lista['Item'], width=Lista['Frecuencia'], color='blue')

plt.show()

Imagen que contiene Gráfico

Descripción generada automáticamente

**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

TransaccionesLista = DatosTransacciones.stack().groupby(level=0).apply(list).tolist()

TransaccionesLista

Texto

Descripción generada automáticamente

# Aplicación del algoritmo

## Configuración 1

Obtener reglas para aquellos artículos que se compran al menos 5 veces al día, entonces, 5 x 7 = 35 veces en una semana, entonces:

1. El soporte mínimo se calcula de 35/7500 = 0.0045 (0.45%).
2. La confianza mínima para las reglas de 20%
3. La elevación de 3.

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

ReglasC1 = apriori(TransaccionesLista,

                   min\_support=0.0045,

                   min\_confidence=0.2,

                   min\_lift=3)

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: 24

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Son 24 reglas. A manera de ejemplo se imprime la primera regla:

print(ResultadosC1[0])

RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ordered\_statistics=[OrderedStatistic(items\_base=frozenset({'light cream'}), items\_add=frozenset({'chicken'}), confidence=0.29059829059829057, lift=4.84395061728395)])

**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: ", item[1]\*100, "%")

    #El tercer índice de la lista

    print("Confianza: ", item[2][0][2]\*100, "%")

    print("Lift: " + str(item[2][0][3]))

    print("=====================================")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número de regla** | **Regla** | **Soporte** | **Confianza** | **Lift** |
| 1 | ('chicken', 'light cream') | 0.4532728969470737 % | 29.059829059829056 % | 4.84395061728395 |
| 2 | ('escalope', 'mushroom cream sauce') | 0.5732568990801226 % | 30.069930069930066 % | 3.790832696715049 |
| 3 | ('pasta', 'escalope') | 0.5865884548726836 % | 37.28813559322034 % | 4.700811850163794 |
| 4 | ('herb & pepper', 'ground beef') | 1.5997866951073192 % | 32.345013477088955 % | 3.2919938411349285 |
| 5 | ('tomato sauce', 'ground beef') | 0.5332622317024397 % | 37.73584905660377 % | 3.840659481324083 |
| 6 | ('whole wheat pasta', 'olive oil') | 0.7998933475536596 % | 27.14932126696833 % | 4.122410097642296 |
| 7 | ('pasta', 'shrimp') | 0.5065991201173177 % | 32.20338983050847 % | 4.506672147735896 |
| 8 | ('frozen vegetables', 'shrimp', 'chocolate') | 0.5332622317024397 % | 23.255813953488374 % | 3.2545123221103784 |
| 9 | ('ground beef', 'spaghetti', 'cooking oil') | 0.4799360085321957 % | 57.14285714285714 % | 3.2819951870487856 |
| 10 | ('spaghetti', 'frozen vegetables', 'ground beef') | 0.8665511265164645 % | 31.100478468899524 % | 3.165328208890303 |
| 11 | ('milk', 'frozen vegetables', 'olive oil') | 0.4799360085321957 % | 20.338983050847457 % | 3.088314005352364 |
| 12 | ('mineral water', 'shrimp', 'frozen vegetables') | 0.7199040127982935 % | 30.508474576271183 % | 3.200616332819722 |
| 13 | ('olive oil', 'spaghetti', 'frozen vegetables') | 0.5732568990801226 % | 20.574162679425836 % | 3.1240241752707125 |
| 14 | ('spaghetti', 'shrimp', 'frozen vegetables') | 0.5999200106652446 % | 21.531100478468897 % | 3.0131489680782684 |
| 15 | ('spaghetti', 'frozen vegetables', 'tomatoes') | 0.6665777896280496 % | 23.923444976076556 % | 3.4980460188216425 |
| 16 | ('spaghetti', 'grated cheese', 'ground beef') | 0.5332622317024397 % | 32.25806451612903 % | 3.283144395325426 |
| 17 | ('herb & pepper', 'mineral water', 'ground beef') | 0.6665777896280496 % | 39.06250000000001 % | 3.975682666214383 |
| 18 | ('herb & pepper', 'spaghetti', 'ground beef') | 0.6399146780429277 % | 39.344262295081975 % | 4.004359721511667 |
| 19 | ('milk', 'olive oil', 'ground beef') | 0.4932675643247567 % | 22.424242424242426 % | 3.40494417862839 |
| 20 | ('spaghetti', 'shrimp', 'ground beef') | 0.5999200106652446 % | 52.32558139534884 % | 3.005315360233627 |
| 21 | ('milk', 'spaghetti', 'olive oil') | 0.7199040127982935 % | 20.30075187969925 % | 3.0825089038385434 |
| 22 | ('mineral water', 'soup', 'olive oil') | 0.5199306759098787 % | 22.543352601156073 % | 3.4230301186492245 |
| 23 | ('pancakes', 'spaghetti', 'olive oil') | 0.5065991201173177 % | 20.105820105820104 % | 3.0529100529100526 |
| 24 | ('milk', 'spaghetti', 'mineral water', 'frozen vegetables') | 0.4532728969470737 % | 28.813559322033893 % | 3.0228043143297376 |

### Conclusiones Configuración 1

* **Primera regla**

La primera regla contiene dos elementos **chicken** y **light cream** que exclusivamente se compran juntos.

* Esto tiene sentido, ya que las personas que compran crema ligera tienen cuidado con lo que comen, por lo que, es probable que compren pollo, en lugar de carne roja.
* El soporte es de 0.45%, la confianza de 29.05%, la elevación de 4.84, esto es, 4.84 veces más probabilidades de que compren crema ligera.
* **Segunda regla**

Para la segunda regla se tienen dos elementos: **escalope** y **mushroom cream sauce** que exclusivamente se compran juntos.

* Esto tiene sentido, ya que las personas que compran escalope (que es un corte de carne) suelen acompañar su platillo con salsa de champiñones, por lo que, es probable que compren estos dos productos por ser complementarios.
* El soporte es de 0.57%, la confianza de 30.06%, la elevación de 3.79, esto es, 3.79 veces más probabilidades de que compren salsa de champiñones si es que compraron escalope.
* **Tercera regla**

Para la tercera regla se tienen dos elementos: **pasta** y **escalope** que exclusivamente se compran juntos.

* Esto tiene sentido, ya que las personas que compran pasta, lo hacen por ser un acompañante de su platillo principal el cual en este caso sería el escalope, por lo que, es probable que compren estos dos productos por ser complementarios.
* El soporte es de 0.58%, la confianza de 37.288%, la elevación de 4.7, esto es, 4.7 veces más probabilidades de que compren escalope si es que compraron pasta.

De esta manera y siguiendo el mismo patrón, se pueden sacar conclusiones para cada una de las 24 reglas mostradas en esta configuración.

## Configuración 2

Obtener reglas para aquellos artículos que se compran al menos 30 veces al día, entonces, 30 x 7 = 210 veces en una semana, entonces:

1. El soporte mínimo se calcula de 210/7500 = 0.028 (2.8%).
2. La confianza mínima para las reglas de 25%.
3. La elevación mayor a 1.

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

ReglasC2 = apriori(TransaccionesLista,

                   min\_support=0.028,

                   min\_confidence=0.25,

                   min\_lift=1.01)

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(10)

ResultadosC2

Texto

Descripción generada automáticamente

pd.DataFrame(ResultadosC2)

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

print(ResultadosC2[0])

Primera regla:

RelationRecord(items=frozenset({'burgers', 'eggs'}), support=0.02879616051193174, ordered\_statistics=[OrderedStatistic(items\_base=frozenset({'burgers'}), items\_add=frozenset({'eggs'}), confidence=0.33027522935779813, lift=1.8378297443715457)])

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: ", item[1]\*100, "%")

    #El tercer índice de la lista

    print("Confianza: ", item[2][0][2]\*100, "%")

    print("Lift: " + str(item[2][0][3]))

    print("=====================================")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número de regla** | **Regla** | **Soporte** | **Confianza** | **Lift** |
| 1 | ('burgers', 'eggs') | 2.879616051193174 % | 33.027522935779814 % | 1.8378297443715457 |
| 2 | ('mineral water', 'chocolate') | 5.265964538061592 % | 32.13995117982099 % | 1.3483320682317521 |
| 3 | ('mineral water', 'eggs') | 5.092654312758299 % | 28.338278931750743 % | 1.188844688294532 |
| 4 | ('mineral water', 'frozen vegetables') | 3.572856952406346 % | 37.48251748251749 % | 1.57246288387228 |
| 5 | ('ground beef', 'mineral water') | 4.092787628316224 % | 41.65535956580732 % | 1.7475215442008991 |
| 6 | ('ground beef', 'spaghetti') | 3.9194774030129316 % | 39.89145183175034 % | 2.291162176033379 |
| 7 | ('mineral water', 'milk') | 4.799360085321957 % | 37.03703703703704 % | 1.5537741320739085 |
| 8 | ('spaghetti', 'milk') | 3.546193840821224 % | 27.366255144032927 % | 1.5717785592296398 |
| 9 | ('mineral water', 'pancakes') | 3.372883615517931 % | 35.483870967741936 % | 1.4886158620191963 |
| 10 | ('mineral water', 'spaghetti') | 5.972536995067324 % | 25.05592841163311 % | 1.4390851379453289 |

### Conclusiones Configuración 2

* **Primera regla**

La primera regla contiene dos elementos: **hamburguesas** y **huevos** que comúnmente se compran juntos.

* Tiene sentido, ya que algunas personas que compran hamburguesas consumen también huevos, como comida de preparación rápida.
* El soporte es de 0.028 (2.8%), la confianza de 0.33 (33%), la elevación de 1.83, esto es, hay casi 2 veces más probabilidades de que cuando se compren hamburguesas, se compren también huevos.
* **Segunda regla**

Para la segunda regla se tienen dos elementos: **agua mineral** y **chocolate**, que comúnmente se compran juntos.

* Tiene sentido, ya que algunas personas que compran agua mineral consumen también chocolate, a manera de un snack rápido y barato.
* El soporte es de 0.0526 (5.26%), la confianza de 0.321 (32.1%), la elevación de 1.34, esto es, hay 1.34 veces más probabilidades de que cuando se compre agua mineral, se compren también chocolates.
* **Tercera regla**

Para la tercera regla se tienen dos elementos: **agua mineral** y **huevos**, que comúnmente se compran juntos.

* Tiene sentido, ya que algunas personas que compran agua mineral consumen también huevos, a manera de una comida saludable o como parte de su compra de la despensa.
* El soporte es de 0.0509 (5.09%), la confianza de 0.2833 (28.33%), la elevación de 1.18, esto es, hay 1.18 veces más probabilidades de que cuando se compre agua mineral, se compren también huevos.

De esta manera y siguiendo el mismo patrón, se pueden sacar conclusiones para cada una de las 10 reglas mostradas en esta configuración.

# Link de Google Colab

🔗 [OCG-Práctica2-RAsociación.ipynb - Colaboratory (google.com)](https://colab.research.google.com/drive/1njwaKXhhodgt1VWzNvAZlqhc_Y9SLne7#scrollTo=85zJymTCWAJS)