1. Objetivo del laboratorio

Desarrollar de forma autónoma ***un Notebook*** *que permitan usar algoritmos para obtener reglas de asociación y patrones secuenciales.*

2. Elementos a utilizar:

- Lenguaje Python
- Librería numérica *NumPy*, *pandas*, *scikit-learn*, *mlxtend*, *apyori* y gráfica *Matplotlib*
- Entorno Anaconda
- Editor Jupyter

3. Práctica 1 (reglas de Asociación)

Objetivo (5 puntos)

Usa la librería *mlxtend* o *apyori*, que nos permiten solucionar todos los problemas relacionados con las reglas de asociación. Para ello utilizaremos el algoritmo Apriori, y una serie de métodos para obtener la distinta información que este genera, aplicándolos al Dataset BlackFriday.csv que se ha proporcionado.

La librería se usará de la siguiente manera:

1) Empezaremos obteniendo los itemsets frecuentes para k=1. En este punto necesitaremos obtener el soporte de los itemset. Por lo tanto, se tendrá que usar un método que dado un itemset devuelva su soporte.
2) Para k≥2.
    a. Mostrar los itemset frecuentes candidatos y su soporte.
3) Repetir el proceso 2 hasta que no se generen nuevos itemsets frecuentes.
4) Mostrar todas las posibles reglas con la confianza de cada una de ellas.
5) Listar todas las reglas que sean de alta confianza.
6) Usar los siguientes métodos. Dado un antecesor, devolver todas las reglas que contengan a dicho antecesor. Dado un umbral mínimo devolver todas las reglas que cumplan con dicha confianza.
7) Utiliza las representaciones gráficas que consideres adecuadas para representar las reglas obtenidas y obtener conclusiones a partir de los datos.

A partir de lo anterior de pide:

- Prueba al menos tres configuraciones de soporte y frecuencia para cada género, edad y tipo de producto. (1 punto)

- ¿Qué diferencias hay entre usar soporte y frecuencia? Respalda la respuesta con datos (1 punto)

- ¿Qué tipo reglas desaparecen según la configuración y categorías (género, edad y tipo de producto) consideradas? ¿Por qué? (1 punto)

- Para una de las configuraciones, interpreta algunas de las reglas que te hayan resultado interesantes obtenidas usando la clase *association_rules* y un par de configuraciones cambiando la variable *metric* y *min_threshold*. Justifica los resultados. (1 punto)

- Dadas las mejores configuraciones. ¿Existen reglas o patrones que se repitan? ¿Podemos generalizar de alguna manera como se comportan los clientes? Expón las conclusiones respaldadas con datos (1 punto)

4. Práctica 2 (Patrones Secuenciales)

Objetivo (5 puntos)

El conjunto de datos "splice.data" contiene los datos de secuencias de genes (DNA) asociados a clases EI (empalme exon-intron), IE (empalme intron-exon) y N (muestra sin empalme). Por tanto, para diferentes clases tenemos recogida una secuencia de genes (donde el orden sí importa) para cada instancia/muestra. Analizando esta información se pueden extraer conclusiones sobre en qué orden aparecen los genes y extraer conclusiones acerca de enfermedades relacionadas con alteraciones genéticas. Este estudio se puede llevar a cabo aplicando el algoritmo Generalized Sequential Patterns utilizando la implementación de disponible en la librería *gsppy*.

- Para la columna que contiene las secuencias de genes, generar una nueva columna (con la que trabajaremos posteriormente). Esta nueva columna debe contener los primeros dígitos de los genes siento este número aleatorio contenido en el intervalo [4-10], es decir, cortaremos las secuencias para que su longitud sea aleatoria entre 4 y 10 dígitos. (1 punto)
- Comprueba que los datos son correctos: no hay clases vacías ni contenido repetido para la misma clase. (2 puntos)
- Prueba al menos dos configuraciones de soporte diferentes. (1 punto)
- Para una de ellas, interpreta algunos de los patrones secuenciales que te resulten curiosos. (1 punto)

Librería gsppy:

Podéis encontrar su documentación en el siguiente enlace: https://pypi.org/project/gsppy/

5. Forma de entrega del laboratorio:

La entrega consistirá en un fichero comprimido RAR con nombre LAB02-GRUPOxx.RAR subido a la tarea **LAB1** que **contenga únicamente**

1. **Por cada práctica** un notebook de Jupyter (archivos con extensión **.ipynb**).
2. La **memoria del laboratorio** que se irá construyendo en el Notebook de manera que se explique todo lo que se hace.

Las entregas que no se ajusten exactamente a esta norma NO SERÁN EVALUADAS.

6. Rúbrica de la Práctica:

1. IMPLEMENTACIÓN: Multiplica la nota del trabajo por 0/1

Siendo una práctica de Data Mining, todos los aspectos de programación se dan por supuesto. La implementación será:

- Original: Código fuente no copiado de internet. Grupos con igual código fuente serán suspendidos
- Correcta: El programa funciona y ejecuta correctamente todo lo planteado en los apartados de cada práctica.
- Comentada: Inclusión (**obligatoria**) de comentarios.
- En las gráficas que se realicen proporciona todos los datos que creas necesarios.

2. MEMORIA DEL LABORATORIO

Obligatorio redacción clara y correcta ortográfica/gramaticalmente. Cada paso que se haga tiene que estar justificado.