PRÁCTICA 1: APRENDIZAJE AUTOMÁTICO

PARTE A: análisis y procesamiento de un dataset

En esta parte usaremos el dataset titanic.csv. Este dataset contiene los siguientes atributos:

- PassengerId: Identificador único de cada pasajero. (Numérico)
- Survived: Indica si el pasajero sobrevivió (1) o no (0). (Binario)
- Pclass: Clase del billete del pasajero (Ordinal):
    - 1 = Primera clase
    - 2 = Segunda clase
    - 3 = Tercera clase
- Name: Nombre completo del pasajero. (Texto)
- Sex: Género del pasajero (male o female). (Categoría)
- Age: Edad del pasajero en años. Puede contener valores nulos. (Numérico)
- SibSp: Número de hermanos o cónyuges que viajaban con el pasajero. (Numérico)
- Parch: Número de padres o hijos que viajaban con el pasajero. (Numérico)
- Ticket: Código del billete del pasajero. (Texto)
- Fare: Tarifa pagada por el billete. (Numérico)
- Cabin: Número de la cabina en la que se alojó el pasajero. Puede contener valores nulos. (Texto)
- Embarked: Puerto en el que el pasajero abordó el Titanic (Categoría):
    - C = Cherburgo
    - Q = Queenstown
    - S = Southampton

Sobre este dataset, hay que realizar las siguientes tareas:

1. Carga el dataset a analizar en la parte A (titanic.csv) con pandas y elimina los valores nulos y duplicados. ¿Cuántas filas se han borrado? ¿En qué beneficia su eliminación? Con reset_index resetea los índices para evitar problemas en los ejercicios siguientes.
2. Antes de empezar, determina si hay algún atributo no nos va a resultar útil y por qué.
3. Relaciones entre atributos: dibuja diagramas de dispersión y calcula coeficientes de correlación. ¿Cuáles son los atributos que están más relacionados y qué podemos interpretar?
4. Atributos numéricos:
    a. Calcula media, desviación típica, valores mínimos, máximos, etc. de los atributos numéricos. Describe estos valores para cada variable.
5. Atributos categóricos:

    a. Dibuja histogramas, diagramas de barras o de tartas para determinar las frecuencias de los valores de los atributos categóricos. Indica el número de valores distintos para cada atributo y el valor más frecuente para cada atributo. ¿Qué atributos están balanceados y cuáles no?
6. Determina si hay outliers. Fíjate en las gráficas y descripciones que has empezado antes.
7. Convierte los atributos categóricos en valores numéricos usando OneHotEncoder (o getdummies) y LabelEnconder. Observa las diferencias y discute cuál sería la mejor opción cuando el dataset sea usado en un modelo de IA.
8. Normaliza y estandariza el dataset. Observa los resultados. Discute qué opción sería mejor usar.

PARTE B: evaluación de modelos de AA

En esta parte vamos a utilizar la librería surprise (https://surprise.readthedocs.io/en/stable/) que implementa algoritmos y otros procedimientos para desarrollar y evaluar algoritmos de sistemas de recomendación. Asegúrate de tenerla instalada antes de empezar a hacer las tareas.

Antes de empezar: determina un valor para una variable SEED (el que sea). Esta variable se la vamos a pasar a los modelos para que cada vez que se ejecute el código salgan los mismos resultados. De esta forma os aseguráis de que los resultados que os salgan serán los mismos que me salgan a mí al ejecutar el código. Si no hacéis esto, vuestro análisis puede no tener ningún sentido en mi ejecución y os arriesgáis al no apto.

1. Carga el dataset de MovieLens de 100K. MovieLens es un dataset muy usado en el desarrollo y evaluación de sistemas de recomendación, y el dataset de 100k contiene 100K interacciones entre usuarios y productos, almacenando qué usuario ha puntuado qué película y con qué rating.
2. Divide el dataset en una partición simple, donde el conjunto de entrenamiento sea el 75% de las interacciones y el resto forme parte del conjunto de evaluación. Aquí tenéis que pasarle random_state = SEED a la función.
3. Vamos a evaluar distintos algoritmos de recomendación:
    a. Filtrado colaborativo basado en vecinos (KNNBasic en la librería: https://surprise.readthedocs.io/en/stable/knn_inspired.html), tanto

basado en usuarios como en productos. Utilizaremos la métrica de similitud de Pearson.
    b. Filtrado colaborativo basado en modelos usando factorización de matrices(https://surprise.readthedocs.io/en/stable/matrix_factorization.html), usando los algoritmos de SVD y NMF.
4. Cada uno de estos algoritmos se entrenarán con el conjunto de entrenamiento. Aquí tenéis que pasarle random_state = SEED a cada uno de los modelos.
5. Después se obtendrán las predicciones que todos los algoritmos obtienen para el conjunto de evaluación. Muestra el resultado de 5 predicciones e interpreta los resultados.
6. Crea una tabla con los valores que se obtienen para las métricas de evaluación RMSE, precision, recall, y NDCG (para NDCG, k es el tamaño de la lista de recomendación y será k = 10). Surprise solo implementa RMSE. Las demás las podéis encontrar en sklearn usando precision_score, recall_score, ndcg_score.
    a. **IMPORTANTE**: solo las películas cuyo rating sea superior a 4 serán consideradas relevantes (incluyendo en la lista de películas recomendadas por el modelo).
7. Explica cada uno de los resultados obtenidos y qué significado tienen. Determina cuál podría ser el mejor método recomendador a utilizar.

Entrega

Se entregarán dos notebooks por separado (en un archivo comprimido), uno por cada parte, en la entrega habilitada en el campus virtual. Recuerda que todas las celdas deben funcionar para que sean corregidas. **Muy importante**: contesta a las preguntas del enunciado y comenta el código y los resultados. Sin las respuestas a las preguntas y el análisis de los resultados, la práctica no conseguirá el apto.