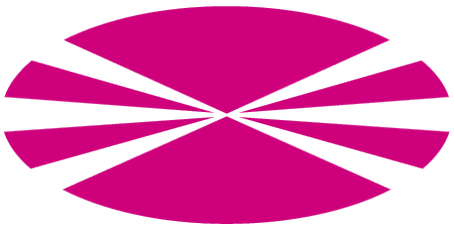

UNIVERSIDADE DA CORUÑA

PRÁCTICA DE SISTEMAS DE RECOMENDACIÓN

Diego Fernández Iglesias
diego.fernandez@udc.es

El bloque de Recomendación de la asignatura de RIWS consta de una parte teórica y una parte práctica. Esta práctica constituye la única entrega para la evaluación de este bloque (0'75 puntos).

Para la práctica se hará uso del paquete *scikit-surprise.* Este paquete puede ser descargado con pip (previamente se necesita numpy), con conda (disponible en el canal conda-forge) o directamente del repositorio de GitHub (https://github.com/NicolasHug/surprise.git).

La entrega de la práctica constará de dos ficheros:

- Un *script* implementado en Python (versión 3.10 o superior). La ejecución del *script* debe permitir responder a los ejercicios planteados en esta práctica.

- Un informe en pdf en el que se comentarán las soluciones a los ejercicios.

Se recomienda analizar los ficheros **example_surprise.py** y **example_grid_search.py** antes de implementar el script a entregar.

Para esta práctica escogeremos tres algoritmos disponibles en el paquete *scikit-surprise*: **NormalPredictor, KNNWithZScore y SVD**. Se realizarán los siguientes pasos de manera secuencial, y la salida de un punto será la entrada del punto siguiente:

1. Usando la librería de Pandas, crear un *DataFrame* a partir del fichero ml-latest-small/ratings.csv accesible desde la siguiente URL: http://files.grouplens.org/datasets/movielens/ml-latest-small.zip. Tener en cuenta que las puntuaciones varían entre 0,5 y 5. Realizar una exploración inicial indicando el número de usuarios, productos y puntuaciones que hay en el *DataFrame* creado. Comprobar si existen valores vacíos (NA) o muestras duplicadas.

2. Eliminar del *dataset* los productos con menos de 10 puntuaciones, ¿cuántos productos quedan? A continuación, sobre el *dataset* obtenido, eliminar los usuarios con menos de 20 puntuaciones, ¿cuántos usuarios quedan? ¿y cuántos productos? ¿de qué tamaño es ahora la matriz?

3. Representar en un histograma el número de puntuaciones por usuario. Hacer lo mismo con el número de puntuaciones por producto.

4. Representar en un histograma la media de puntuaciones por usuario. Hacer lo mismo con la media de puntuaciones por producto.

5. Representar en un diagrama de barras la distribución de las puntuaciones.

6. Crear un objeto de tipo *surprise.Dataset* a partir del *DataFrame* empleado previamente y definir una semilla para que todo el código sea reproducible. Para la experimentación se usará una validación cruzada con 5 *folds*. Para definir los *folds* se empleará la función *set_my_folds* proporcionada (con shuffle = True, tal y como está definido por defecto).

7. Realizar un proceso de validación con *GridSearchCV* (considerando measures = ["mae"], cv = 3 y n_jobs = -1) para determinar los valores de los hiperparámetros del algoritmo KNNWithZScore. En concreto, se estudiarán los siguientes parámetros: *k* (valores 25, 50 y 100), *min_k* (valores 1, 2 y 5). Se empleará Pearson como media de similitud. El resto de parámetros se dejarán con sus valores por defecto.

8. Considerando la tarea de predicción y la métrica MAE, comparar los algoritmos **NormalPredictor** (con sus parámetros por defecto)**, KNNWithZScore** (con los mejores hiperparámetros definidos en el proceso de grid search) y **SVD** (estableciendo n_factors = 25). ¿Cuál es el algoritmo que mejor se comporta y por qué? Comparar el funcionamiento de los algoritmos. Justificar la respuesta.

9. Con respecto a la tarea de recomendación, se tendrán en cuenta listas de recomendación de tamaños 1, 2, 5 y 10, el umbral de relevancia será de 4 y se usarán exactamente los mismos algoritmos que en apartado anterior, con los mismos conjuntos de entrenamiento y test (definidos con la función *set_my_folds*)**.** ¿Cuál es el algoritmo que mejor se comporta en este caso? Justificar las respuestas tomando como base una gráfica de precision-recall. Para el cálculo de las métricas, se partirá del código disponible en: https://github.com/NicolasHug/Surprise/blob/master/examples/precision_recall_at_k.py

Para entregar el ejercicio se empleará la actividad de moodle denominada "Sistemas de Recomendación. Tarea". Se entregará tanto un script de Python llamado script-sistrec.py con el código implementado, como un informe en pdf llamado informe-sistrec.pdf respondiendo a las preguntas propuestas. La fecha límite para esta entrega será **el jueves 14 de diciembre de 2023 a las 23:59.**