Probabilistic-Matrix-Factorization

* [Paper](https://proceedings.neurips.cc/paper_files/paper/2007/hash/d7322ed717dedf1eb4e6e52a37ea7bcd-Abstract.html)
* [Github](https://github.com/fuhailin/Probabilistic-Matrix-Factorization/tree/master)

1. Entorno y ejecución
   1. Clonar repositorio  
      git clone <https://github.com/fuhailin/Probabilistic-Matrix-Factorization.git>
   2. Crear el entorno  
      python -m venv .venv
   3. Instalar dependencias  
      pip install -r requirements.txt
   4. Ejecutar  
      python RunExample.py
2. Estructura del código
   1. LoadData.py
      1. Carga data/ml-100k (ratings) y los convierte a una matriz / lista de tripletas (user, item, rating).
      2. Funciones
         1. load\_rating\_data (file\_path)  
            Carga *u.data* línea a línea, extrae usuario, película, rating (ignora timestamp), y devuelve un np.array de tripletas.
         2. spilt\_rating\_dat (data, size=0.2)  
            Parte aleatoriamente las tripletas en train/test según el umbral size.   
            No garantiza balance por usuario, pero es suficiente para el ejemplo.
   2. ProbabilisticMatrixFactorization.py
      1. Implementa PMF
   3. RunExample.py
      1. Usa LoadData para cargar y dividir los datos.
      2. Crea el modelo PMF, lo entrena y muestra.
   4. test\_pmf.py
      1. Pruebas
3. Base de Datos MovieLens 100k
   1. Se fichero se llama *u.data* y se encuentra en la carpeta *data\ml-100k.*
   2. El dataset MovieLens 100k fue creado por GroupLens (University of Minnesota) y contiene valoraciones de películas de 943 usuarios sobre 1682 títulos.
   3. Consta de 4 columnas:

|  |  |  |  |
| --- | --- | --- | --- |
| User\_ide | Item\_id | rating | timestamp |
| ID del usuario  (de 1 a 943) | ID de la película (de 1 a 1682) | Valoración que ese usuario dio a esa película (entero 1–5) | marca de tiempo UNIX |
| *196* | *242* | *3* | *881250949* |

1. Explicación de PMF
   1. Técnica usada por el repositorio: PMF:

Probabilistic Matrix Factorization de Mnih & Salakhutdinov (NIPS 2007).

* 1. Es un modelo probabilístico que representa a cada usuario e ítem (película) con vectores.
  2. Ejemplo real explicativo   
      301 98 4 882075827

301 790 4 882078621

301 401 4 882078040

301 56 4 882076587

301 79 5 882076403

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario\Película | 56A | 79R | 98M |
| 301 | 4 | 5 | 4 |
| 302 | ? | ? | 5 |
| 303 | ? | 3 | ? |

* + 1. Predecir ‘?’ para recomendar
    2. PMF asume que cada usuario y cada ítem se pueden representar como un vector:
       1. Usuario = vector de gustos

u301 = [4, 5, 4]

* + - 1. Ítem = vector de características   
         vPel = [nivel de Acción, nivel de Romance, nivel de Miedo]  
         V56 = [0.2, 1.0, 0.1]
* Acción → la película tiene muy poca acción.
* Romance → la película tiene muchísimo romance (su género principal).
* Miedo → la película casi no tiene terror.
* “La película 56 es casi toda romance, con un toque muy pequeño de acción y miedo.”