Assignment 3:
LSH and minhashing for Netflix data

Wojtek Kowalczyk

Data: extract from Netflix Challenge

- About **100.000** users that watched in total **17.770** movies;
- Each user watched between **300 and 3000** movies
- The file contains about **65.000.000 records (720 MB)** in the form:

<user_id, movie_id> : “user_id watched movie_id”

- Similarity between users: Jaccard similarity of sets of movies they watched:

jsim(S1, S2) = #intersect(S1, S2)/#union(S1, S2)

- **Task:**

find (with help of LSH) pairs of users whose jsim > 0.5

(**brute-force search** too slow: 5.000.000.000 pairs – about **2 months of cpu time**)

- **Data:** https://surfdrive.surf.nl/files/index.php/s/WwZqzkkHxg6KLlL

To Do:

- Implement minhashing and LSH
- Tune it (signature length, number of bands, number of rows per band)
- Randomize, optimize, benchmark, polish the code, …
- Deliver your code, including the main.py file which:
    1. loads the file user_movie.npy (don't include it in your submission!)
    2. runs computations
    3. dumps results to a text file: just a list of records: user1, user2
- main.py must explicitly set the random seed to a specific value, eg.:

  np.random.seed(seed=17) [details in instructions!]
- the total runtime < 30 minutes (at most!)

  (after 30 minutes the program will be killed!)

Grade:

- a working code that produces 10 valid pairs in < 30 min: **6.0**

- the key measure: the average number of found pairs per minute

- also taken into account:
  - the **total number of found pairs** (over 5 different runs),
  - the **median** run time (over 5 different runs),
  - code **readability**, elegance.

- No report required!

The limit of 30 minutes is quite generous:

- *10 minutes (single core, 8GB RAM) should be enough*
- *200 pairs/minute is possible*

Deadline: Tuesday, 23th October, 23:59