# Assignment 2:
# LSH for the Netflix data

Wojtek Kowalczyk

# Data: extract from the 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)** of 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 expensive: 5.000.000.000 pairs)

# To Do:

1. Implement the naïve algorithm that calculates similarities of all pairs of users and run some tests to estimate the total run time of this “exact” algorithm (you don’t have to wait till it is finished!)
   Your program should only print the final estimate of the total runtime).

2. Implement the LSH algorithm and apply it to the data.
   - Tune it (signature length, number of bands, number of rows per band)
   - Randomize, optimize, benchmark, polish the code, …
   - Deliver your code, including lines which:
     1. load the file user_movie.npy (don’t include this file in your submission!)
     2. dump results to a text file ans.txt (just a csv list of records: user1, user2)
     3. set the random seed to a specific value, eg.: np.random.seed(seed=17)

# Evaluation criteria:

Your code (notebooks) will be evaluated using the usual criteria:

- code readability, elegance
- comments
- the number of generated pairs in the first 15 minutes of the run

Submit two notebooks (as a .zip file): *time_estimate.ipynb* and *lsh.ipynb*

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

**Additional hints:** Content->Old->Assignments>A3_instructions.pdf

(note that in 2019 instructions are different!)

***Deadline: over 2 weeks (Tuesday, 8th October, 23:59)***