HW4 – Recommender System

Team Name(s):

Rank & RMSE score:

Approach:

Read from the necessary .dat files (movie\_directors.dat, movie\_actors.dat, movie\_genres.dat) and create pandas DataFrames out of them.

Extract

Trial and Error stuff

Research stuff

1 tried dataframe manipulation

2 tried to optimize through multiprocessing

3 tried to optimize through cython

4 researched ways to optimize dataframes

5 decided to go with manipulation of .csv files

Initially, the utility matrix, with rows as user-movie pairs and columns as directors, actors, and genres, was created using the pandas DataFrames. This was done by reading the .dat files, movie\_directors.dat, movie\_actors.dat, and movie\_genres.dat, and placing them in DataFrames. These individual frames were then merged together.

The algorithm that we designed populated the utility matrix with a ‘1’ if the attribute was present in the movie that the user had watched. However, to iterate and populate the utility matrix using all 641,700 entries in the training set was extremely costly in run-time by using this approach. In attempting to optimize our solution, we found that the pandas DataFrames are have a high overhead despite the convenience factor of using these libraries. Even utilizing all four cores on with the multiprocessing library still yielded a run time that was not efficient enough for our preferences. In addition, we attempted to use Cython, which is C extensions for python. Based on our research, this new technology would allow us to have the performance benefits of the C language while writing code in the python language. Unfortunately, we gave up on using Cython due to the myriad of compile time errors. Due to the time constraint and the approaching dead line, we decided that the time improvements given by Cython were not worth spending more time trying to learn how to use the new technology.

Because of

1. Your Approach
2. The rationale behind your approach, the content

features you used if any and what worked, what did not.  
Archive your parent folder (.zip or .tar.gz) and submit via Blackboard for HW4.

Run time for item creation profile: ~ 15mins

Run time for utility matrix creation: ~ 2hr 15min