**Documentation**

**Job recommendation system**

**University of Miami**

This document describes all the .txt and .exe files used or generated by the system

**X (used)**

Matrix = Jobs X features

**Y (used)**

Matrix = Jobs X user

**R (used)**

Matrix = Jobs X user

* The R should just show 1 or 0 (binary approach) according with Y. It means that if Y has some measured job, the R will be 1 in that cell. 0 Otherwise.

**Job\_names (used)**

Just a column vector with the job’s name.

\* For our experiment with the Portuguese expressions, we are using the file above.

**Expressions (used)**

The name for each expression

\* The ID of the expression (not the expression anymore)

**User\_table (used)**

Name for each user with their respective proficiency (self-rating).

**Results (generated)**

Column 1 is a Job from the TOP 10(or less if TOP 10 is not applicable) jobs compared with each job (column 3) that has a similarity equal or greater than 70%.

Column 2 is the job difficult.

Column 3 is the predicted rate for our collaborative filtering for recommended jobs.

Column 4 is the job that is being compared with the job at column 1.

Column 5 is the original rating by employer for each job in column 4.

Column 6 is the similarity in percentage. Just for jobs with similarity equal or greater than 70%.

Column 7 is the user number (id).

**Average (generated)**

Here we have the top 10 jobs with an average between the job and the others jobs with similarity greater than 70%

Column 1 is the job name from the TOP 10 jobs (or less if TOP 10 is not applicable).

Column 2 is the job difficult.

Column 3 is the predicted rating for the recommended job.

Column 4 is the average of the ratings among the TOP 10 jobs which have similarity equal or greater than 70% to the recommended job.

Column 5 is the average of the percentage among the TOP 10 jobs which have similarity equal or greater than 70% to the recommended job.

Column 6 shows the difficulty of each job related with the jobs the user have been done.

RATING AVGS TOTAL row shows the total average from the similarities from the top 10 jobs for the actual user.

PERCENTAGE AVGS TOTAL row shows the average from all the similarities.

JOB DIFFICULTY AVGS TOTAL is calculated according with the column 2

SIMILAR JOBS DIFFICULTY AVGS TOTAL row shows the average from the difficulties calculated in the column 3

SELF INACCURACY is how much, in percentage, a user is compared with he/she put in your self-evaluation. If positive, the user is above of his/her self-evaluation (This group is not good). If negative, the user is bellow of his/her self-evaluation (GREAT).

AVERAGES FOR DIFFICULTIES

\* In the last two rows, we have the total system statistics for all users in the dataset.

**IDandAVG (generated, to be used at the Server)**

Here we write the users ID and their calculated rating.

Column 1 is the user ID.

Column 2 is the calculated rating by our software.

**Difficulty (generated)**

On this file we have the average difficulty for the recommended jobs, the average difficulty of the similar jobs to the recommended ones, and the self-rating for each one of the users.

\*each line speaks about a user

Column 1 is the average difficulty for the recommended jobs.

Column 2 is the average difficulty of the similar jobs (to the recommended ones).

Column 3 is self-rating for each one of the users.

\*there is an excel file called DIFFICULTY that has the same data as the .txt file, but also the following graphs: Column 1 vs. Column 2, Column 1 vs. Column 3, and Column 2 vs. Column 3.

**EXCEL FILE: STATISTIC**

***Community inaccuracy:***

The community inaccuracy according to the self-evaluation for the five experiments.

***Similarity***

The similarity correspond to the percentage of similar jobs that were recommended for the five experiments.

***Self and software ratings***

There are five graphs that have the amount of people with their own self-evaluation (before) and the evaluation given by the system (after) for them.

***Self and software ratings 2***

There are five graphs that have the amount of people according with the systems ratings. So, we can compare the amount of each evaluation group before and after the software execution.

***System***

This graph shows us the behavior of a job for the first 10 users according with the quantity of data that we have. I our case we have for 11(10 people + Google), 21, 31, 41 and 53. All the statistics are considering the Google user.

**EXCEL FILE: NEW FINDS RECOMMENDATIONS**

Column 1: is the job description (the expression to be translated).

Column 2: Job ID. They are the job ids for the recommended jobs, in our case there were 10 recommended jobs.

Column 3: This is the number of time the job listed in column 1 were recommended to different users. The total number of users is 53 and the total number of recommendations is 342.

Column 4: This column is the percentage that the recommended job had according to the total of 342 recommendations.

This table can be seen easily in a pie graph, were you can see the percentages to the recommended jobs over the 342 total recommendations.

After that, we have 3 pieces of data.

The first is the total number of recommendations, which are 342. Considering that we have 53 users and we are considering the top 10 recommendations, we should have 530 recommended jobs, but the value 342 is right because we are considering for the recommendations only the jobs that had jobs >= 70% similar for each user.

The second is the number of new expressions, which means that for each one of the users, new jobs were recommended, jobs that didn’t do before. The total number considering all users was 249.

The last piece is the number of recommended jobs that the users already did, and were recommended for them to do it again. The sum of all jobs for all users was 93.

This information can be seen in a pie graph considering the percentages, were 73% is about the new jobs recommended, and 27% is about the already done recommended jobs above the total of 342 recommendations.

This is good result because with addition of new jobs on the platform, they will not suffer of starvation, they will be recommended to users if the users have done similar jobs to them.