Execise 1

Method --- programmed with numpy Solution:

Task 1: Cosine similarity between 1 and 2 --- with alpha = 1 and beta = 1: 0.9999973 (1st) Cosine similarity between 1 and 3 --- with alpha = 1 and beta = 1: 0.9999953 (2nd) Cosine similarity between 2 and 3 --- with alpha = 1 and beta = 1: 0.9999878 (3rd)

Task 2 Cosine similarity between 1 and 2 --- with alpha = 0.01 and beta = 0.5: 0.9908 (2nd) Cosine similarity between 1 and 3 --- with alpha = 0.01 and beta = 0.5: 0.9915 (1st) Cosine similarity between 2 and 3 --- with alpha = 0.01 and beta = 0.5: 0.9691 (3rd)

Task 3 The fair alpha value is: 0.002054794520547945 The fair beta value is: 0.1875 Cosine similarity between 1 and 2 --- with alpha = 0.002054794520547945 and beta = 0.1875: 0.9943 (1st) Cosine similarity between 1 and 3 --- with alpha = 0.002054794520547945 and beta = 0.1875: 0.9956 (2nd) Cosine similarity between 2 and 3 --- with alpha = 0.002054794520547945 and beta = 0.1875: 0.9822 (3rd)

Execise 2

How can a competitor - in principle - try to steal the valuable data for recommendation from this website?

He could try to extract the rating values by liking items and determining the recommended items. With the main goal of recreating the data from on which those recommendations are calculated in the first place.

Does this work better when the web shop implemented a content-based or a collaborative filtering system?

This would work better with a collaborative system as the content based one is focused on the prefernces of the user, which the competitor would than define himself by the first likes that he chooses. A collaborative system on the other hand draws conclusions fro the whole user/item base.

What data would the competitor be able to infer?

- Clusters of simular items
- List of popular items (recommended quite heavely)
- List of unpopular item (rarely recommended)

Would this technique have an impact on the recommendation system, i.e., would this attack create a bias on the data?

Only if it is a small system with a very limited amount of users. Because only than the impact of single Preference is high. This however does not apply when comparing between items. Or when working againt a collaborative system. In which case the attacker would only confuse his own recommenations.

Why is this attack probably not viable in any case?

The target system would need to be in a quite extrem dependencie for singular user profile. This would speek against the recommender system heavily, in which case it is probably not even giving good recommendations in the very first place.

Exercise 3

a)

Let's start by rewriting A,B and C as sets. $$A = \{ 1,2,4,5,7,8 \}$$ $$B = \{ 2,3,4,5,6,7 \}$$ $$C = \{ 1,3,4,6,7,8 \}$$

From this we can compute the jacardi-distances:

A and B:

- size_of_int= 4
- size_of_union = 8
- $D_{Jac}=1-\frac{4}{8}=0.5$

B and C:

- size_of_int= 4
- size_of_union = 8
- $D_{Jac}=1-\frac{4}{8}=0.5$

C and A:

- size_of_int= 4
- size_of_union = 8
- $D_{Jac}=1-\frac{4}{8}=0.5$

b)

Let's start by rewriting A,B and C as vectors. $$A = \{ 4,5,0,5,1,0,3,2 \}$$ $$B = \{ 0,3,4,3,1,2,1,0 \}$$ $$C = \{ 2,0,1,3,0,4,5,3 \}$$

Calculated using the same python script as in execise 1:
Cosine similarity between 1 and 2 : 0.6010407640085653
Cosine similarity between 1 and 3 : 0.6149186938124421
Cosine similarity between 2 and 3 : 0.5138701197773616

c)

Let's start by rewriting A,B and C as sets. $$A = \{ 1,2,4,7\}$$ $$B = \{ 2,3,4 \}$$ $$C = \{ 4,6,7 \}$$

A and B:

- size_of_int= 2
- size_of_union = 5
- $D_{Jac}=1-\frac{2}{5}=0.6$

B and C:

- size_of_int= 1
- size_of_union = 5
- $D_{Jac}=1-\frac{1}{5}=0.8$

C and A:

- size_of_int= 2
- size_of_union = 5
- $D_{Jac}=1-\frac{2}{5}=0.6$

d)

Let's start by rewriting A,B and C as vectors. $$A = { 1,1,0,1,0,0,1,0 }$$ $$B = { 0,1,1,1,0,0,0,0 }$$ $$C = { 0,0,0,1,0,1,1,0 }$$

Cosine similarity between 1 and 2 : 0.5773502691896258
Cosine similarity between 1 and 3 : 0.5773502691896258
Cosine similarity between 2 and 3 : 0.33333333333333337

e)

Lets rewirte the matrix (subtracting: 3,33 for A, 2,33 for B and 3 for C) $$A = { 0.77, 1.77, 0, 1.77, -2.33, 0, -0.33, -1.33}$$ $$B = { 0, 0.77, 1.77, 0.77, -1.33, -0.33, -1.33, 0 }$$ $$C = { -1, 0, -2, 0, 0, 1, 2, 0 }$$

f)

Cosine similarity between 1 and 2 : 0.5896656091142637
Cosine similarity between 1 and 3 : -0.12014940271473741
Cosine similarity between 2 and 3 : -0.731660124151719

Exercise 4

Student raiting website

- Users: The students will be the different users
- Items: The system knows different kinds of items: lectures, exams, seminars and professors
- other likes/dislikes: passing a exam, not finishing a course (dislike), practicals with certain professors, thesis with a certain professors

Art sharing network

- Users: the artists/art enjoyers
- Items: the art pieces
- other likes/dislikes: not liking a image(dislike), commenting on a picture(possibly with sentiment analysis), liking a user profile(small apreciation for all said artist pictures)

Dating profile

- Users: the users signed up to the plattform

- Items: the attribute-values of the other users // the other users
- other likes/dislikes: block(dislike), number of messages send(like),

This case is somewhat special since users are also the items. Without the description of there dream partner and thier own answering of some questions this system would highly suffer from the new users problem.