Report for the practical assignment

Part 1: USER-ITEM

INFDTA01-1 – Data Science (Recommender systems) – 2015/16

**Important**: you can work in groups of *two* students, but the final report is *individual*. The *code* can be the same for the two students working in pair, but *explanations/answers* must be different from each other. The code must be readable (by placing comments and naming well the variables).

**Delivery:** Fill this document with your answers (code, explanations, etc…) and upload a zipped folder (containing *this document* and *all your code*) on N@tschool *before* the deadline specified in the modulewijzer.

**Evaluation:** Each of the numbered sections below gives up to 1 point. The total number of points for this part of the assignment is, thus, 12. The final grade of the whole practical assignment is obtained combining the points of Part 1 and Part 2 (see formula in the modulewijzer).

Student name: \_\_Zahey Boukich\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

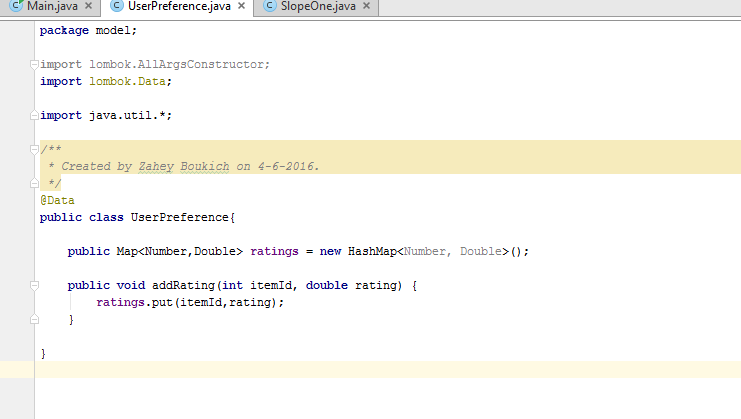
Student number: \_\_\_\_\_\_0780221\_\_\_\_\_\_\_\_\_\_\_\_\_

Class (3A/B/C): \_\_\_\_INF3B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Language (Java, C#, …): \_\_\_\_\_\_\_\_JAVA\_\_\_\_\_\_\_\_\_\_\_

[ If code was developed in pairs, name of the other student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ]

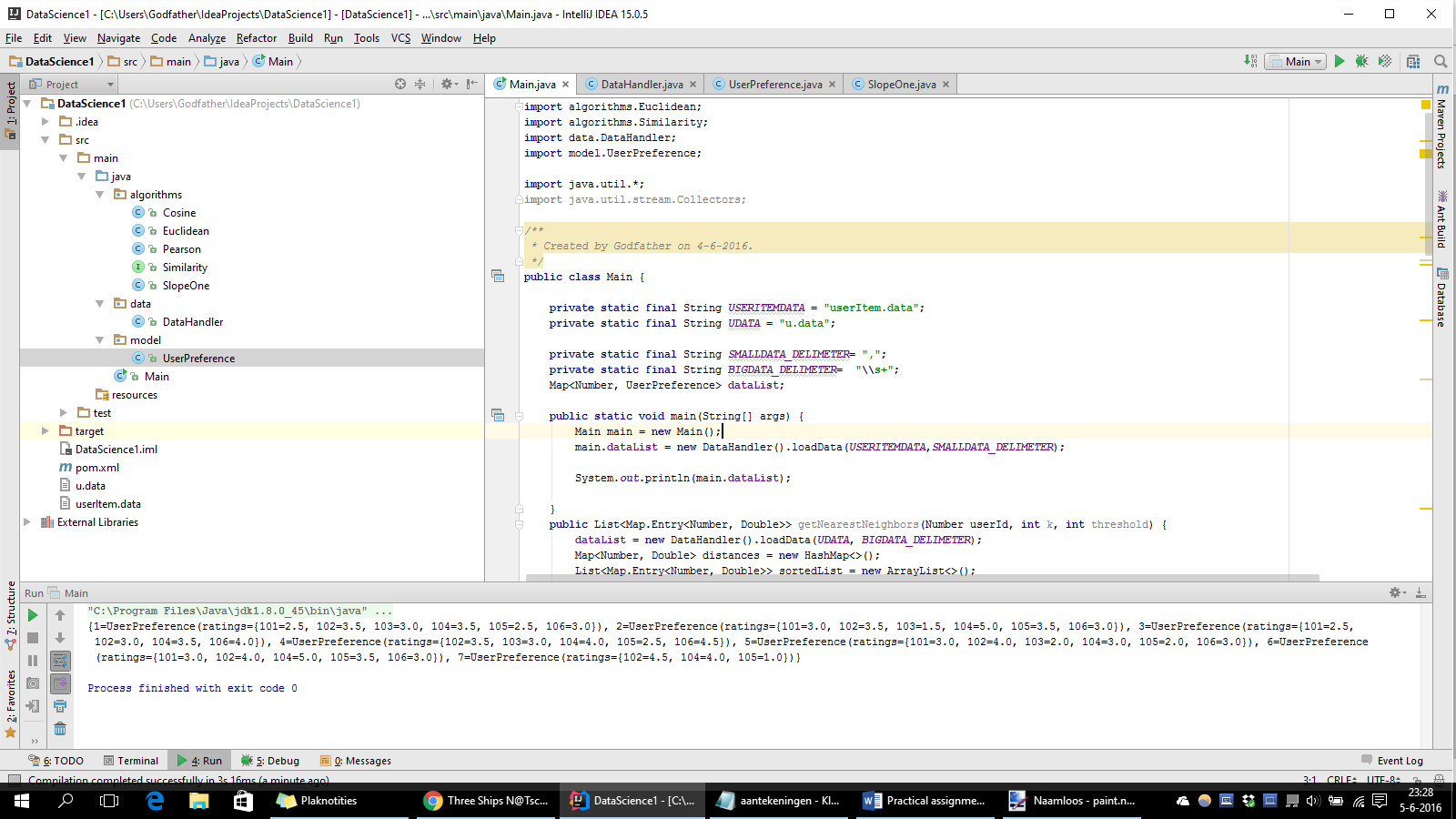
# Importing data

*Copy and paste here the code of the* ***UserPreference*** *class (which contains the ratings associated to a user)*

Which data structure did you use to store the ratings of items inside the UserPreference class? Why did you choose that in particular?

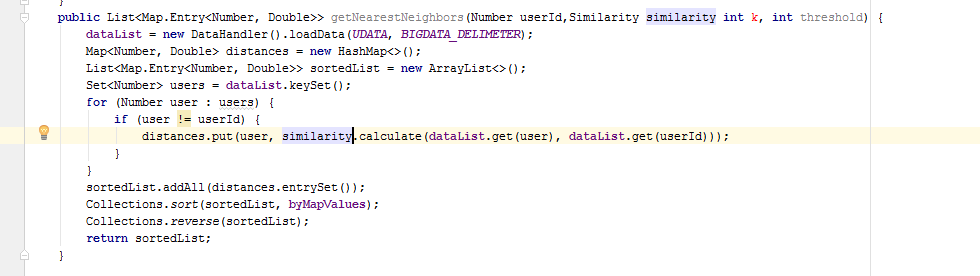
**I store them in hashmap because then it s easy to look up values by the keys a userId based on the itemId or ratingid. An its O(1) look up insert and removal exaclty what needed in this case because of the big data.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Show with a screenshot the result of importing the small dataset (userItem.data) into the hash map (i.e., a print of what is inside the hash map)*

**

# Nearest neighbours

*Copy and paste here the code of the* ***nearest neighbours’*** *algorithm*



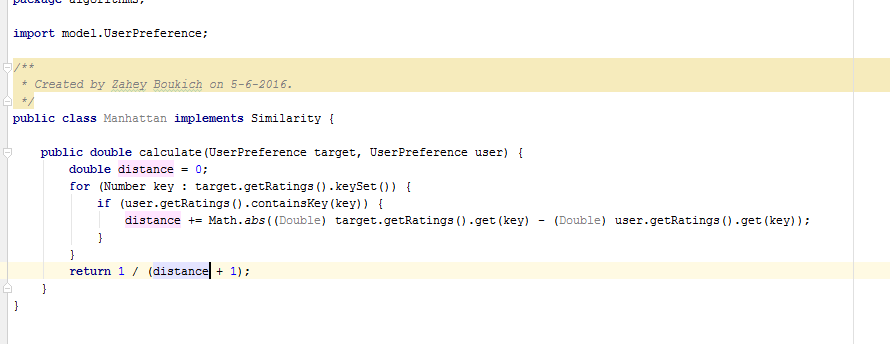
# Relationship between algorithm and similarity measures

Is the algorithm generic with respect to the similarity/distance measure used (Pearson, Cosine, etc.)? If yes, explain how you achieved that.

\_By creating a interface with calculate abstract method this way new algorithm can implement the interface to be a algorithm. This gives the flexibility to use the interface as a parameter in methods to achieve polymorphism and decoupling and change algorithm at runtime in case of this assignment\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Similarity measures

*Copy and paste here the code containing the computation of the three* ***similarity measures*** *(Pearson, Cosine, Euclidean)*

**

# Results of nearest neighbours

Consider the small dataset imported before (*userItem.data*). If we focus on user **7** (target user) and we compute its **3** nearest neighbours (with an initial similarity threshold of ), which are the results?

***PEARSON***

Nearest neighbour 1: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 2: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 3: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

***COSINE***

Nearest neighbour 1: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 2: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 3: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

***EUCLIDEAN***

Nearest neighbour 1: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 2: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

Nearest neighbour 3: \_\_\_\_\_\_ with similarity \_\_\_\_\_\_\_\_\_\_\_\_

# Comment on Pearson result

What is the **Pearson** coefficient of similarity between users **3** and **4**? *Why* does it have that value?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Predicting ratings

*Copy and paste here the code that, given a target user and a product id, computes the* ***predicted rating*** *of that product for that user*

# Results of predicting ratings

Given the 3 nearest neighbours of user **7** computed with the **Pearson** similarity measure, predict the ratings that user 7 would give to items **101**, **103** and **106**:

Predicted rating for item 101: \_\_\_\_\_\_\_\_\_\_\_\_\_

Predicted rating for item 103: \_\_\_\_\_\_\_\_\_\_\_\_\_

Predicted rating for item 106: \_\_\_\_\_\_\_\_\_\_\_\_\_

Given the 3 nearest neighbours of user **4** computed with the **Pearson** similarity measure, predict the rating that user 4 would give to item **101**:

Predicted rating for item 101: \_\_\_\_\_\_\_\_\_\_\_\_\_

# Adding and updating ratings

Suppose that user **7** rates the item **106** with **2.8**. Add this rating into your dataset and update the nearest neighbours of user 7 (using **Pearson**). Compute again the predicted ratings of the other items (**101** and **103**):

Predicted rating for item 101: \_\_\_\_\_\_\_\_\_\_\_\_\_

Predicted rating for item 103: \_\_\_\_\_\_\_\_\_\_\_\_\_

Suppose that user **7** changes idea and rates the item **106** with **5**. Update the rating into your dataset and the nearest neighbours of user 7 (using **Pearson**). Compute again the predicted ratings of other items (**101** and **103**):

Predicted rating for item 101: \_\_\_\_\_\_\_\_\_\_\_\_\_

Predicted rating for item 103: \_\_\_\_\_\_\_\_\_\_\_\_\_

# Importing another dataset

*Show with a screenshot the result of importing the MovieLens100k dataset into the hash map (i.e., a print of what is inside the hash map). NOTE: print only a few items, not all of them!!!!!!!!*

# ) Computing the top recommendations

*Copy and paste here the code which finds the* ***n top recommendations*** *for a target user*

# ) Results and analysis of top recommendations

Consider user **186** and use your algorithm (with the following parameters: **25** nearest neighbours, **0.35** threshold, **Pearson** similarity) to compute the **8** top recommendations for him:

Recommendation 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 4: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 5: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 6: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 7: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 8: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Based on the results above, do you think it could be better to compute the predicted rating only for movies which were rated by *more than one* nearest neighbour (i.e., at least two or three)? Why?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 12) New results of top recommendations

If yes, modify your algorithm to compute the predicted ratings considering only products rated by *at least* **3** neighbours. Execute again the program and put here the updated results:

Recommendation 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 4: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 5: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 6: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 7: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Recommendation 8: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with predicted rating: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_