**CS 435**

**Introduction to Data Mining**

**Final Project Report**

**May 10th, 2018**

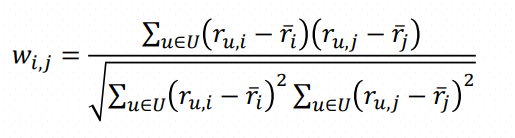
**Bunyamin ISERI**

Recommender systems are used for identifying user preferences and showing new products that the user may like. These systems are used in e-commerce websites like Amazon, eBay, or streaming sites like HBO, Hulu, Netflix, etc. There are various alternatives for recommendation systems where some of them works better in different conditions. The dataset we used for the project has 943 users and 1682 items which means more than 1,5 million data for ratings. Since we have that much data, I used one of the technique that has been used in Netflix’s recommendation system competition.

The technique that I used is Collaborative Filtering . There are two types of this technique: user-to-user matching and item-to-item matching. According to my research, I found that item-to-item matching will be better for large datasets which has large amount of missing data.

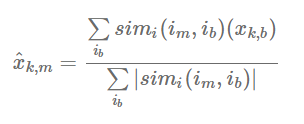
In my code, I started with putting all rating values into 943x1682 array to make my operations easier. I set the default value “0” for missing values. I also calculated the average rating for each item and saved them into “avg\_array” which has the length of 1682. These average values will be used in other operations while predicting.

In the second step, I need to create a weight matrix and fill it with the weights between all items. This weight will show how much an item is similar to another item. Since it will show all weights between items, weight array is 1682x1682. I scanned for weight matrix with 2 nested for loops to and made the operation if the value is zero and item numbers are different. Because items should not have weight with themselves. I designed the algorithm to calculate weights with this formula:



Basically, the formula works in the set of users (U) which have given ratings (r) for both items. I looked for the ratings of both items and if they are different than zero value, it means that a user has given rating for both items and I saved these users in “user\_list” in my code. After that I made the operations according to the formula to find all weights between all items.

For the prediction step of the algorithm I used the formula below:



The formula basically sums all weight values for an item multiplied by the rating value of that user for other items divided by the sum of absolute value of all weights. For example if we are trying to predict the rating of user5 to item6. First we will multiply weight between item6 and item1, and multiply it by rating that user5 has given to item1. We will add this to “a” value. Next, we will multiply weight of items 6&2 and multiply it with rating of user5 to item2. We will add this value to “a” again. This will continue for all 1682 items. After calculating the top part, we will sum up all absolute values of weights for item5 (because weight can be lower than zero) and this will give the bottom part of the equation. When we divide both values, we will get the prediction rating for user5 to item6.

Additionally, in my code I added the average ratings of each items to my predictions to make normalization. Also I set the values higher than 5 to 5 and lower than 1 to 1 for not to cross the bounds of 1-5.