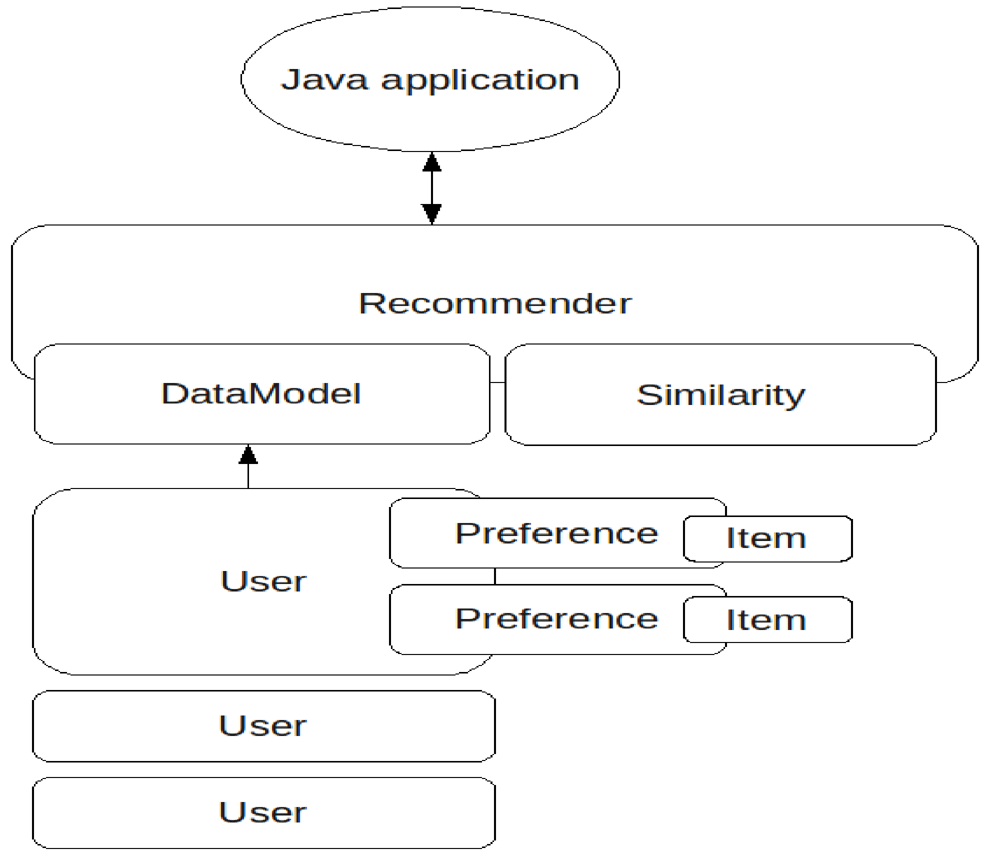
BTP PRESENTATION-PHASE III



Rahul Sihag (200901093)
Devender Goyal (200901072)

Tools Used

- Cygwin
- Java SDK 6u23 x64
- Eclipse 3.6(helios) SR1 x64
- Maven 3.0.2
- Hadoop 0.21.0

Working of Recommendation System

$$\begin{bmatrix} 5 & 3 & 4 & 4 & 2 & 2 & 1 \\ 3 & 3 & 3 & 2 & 1 & 1 & 0 \\ 4 & 3 & 4 & 3 & 1 & 2 & 0 \\ 4 & 2 & 3 & 4 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 & 2 & 1 & 1 \\ 2 & 1 & 2 & 2 & 1 & 2 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 0 \\ 0 \\ 4 \\ 4.5 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 40 \\ 18.5 \\ 24.5 \\ 40 \\ 26 \\ 16.5 \\ 15.5 \end{bmatrix}$$

- S is the similarity matrix between items
- U is the user's preferences for items
- R is the predicted recommendations

Injecting Domain specific information

- Take advantage of what we know about the content i.e gender, wealth , North – South, graduation etc.

Example :

Create **custom similarity metric** for Gender. :

- Two Urban profiles assign similarity value 1.
- Two Rural values assign similarity value 1.
- Urban - Rural profile assign value -1.
- If Information not disclosed Assign value 0.

- Similarly create preferences for other parameters like urban – rural ,North- south , graduation etc

Injecting Domain specific information

- Problem : Mahout does not provide content based recommendation system.
- Solution : Mahout has various expansion points and APIs that allow us to do that.

Idrescorer : Allow us to modify recommendations . If a person is viewing all the profiles of college graduates then recommend him the profile of graduate persons only.

Similarly it is used to filter out mystery novels and books etc. Below a very simple implementation

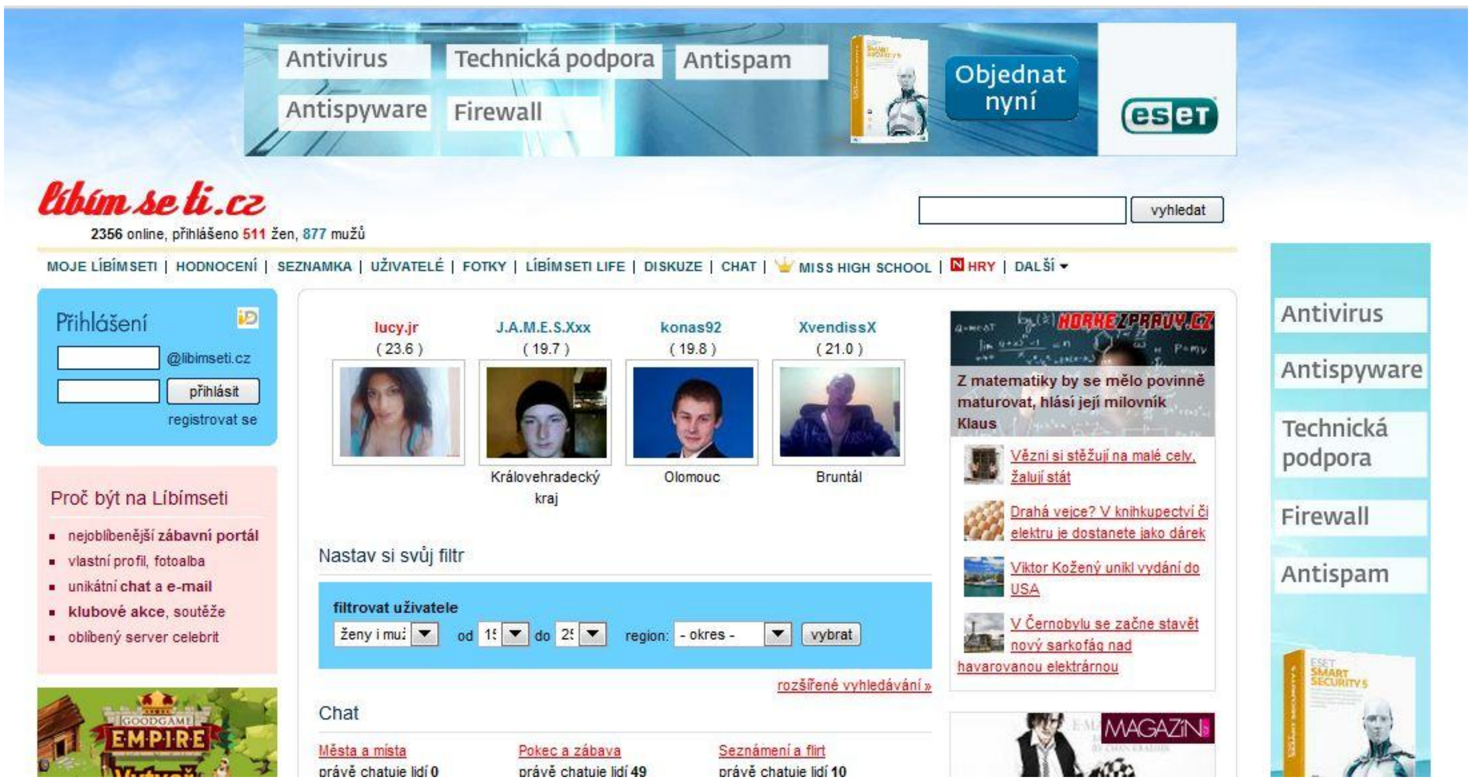
```
public class GenreRescorer implements IDRescorer {

  private final Genre currentGenre;

  public GenreRescorer(Genre currentGenre) {
    this.currentGenre = currentGenre;
  }

  public double rescore(long itemID, double originalScore) {
    Book book = BookManager.lookupBook(itemID);
    if (book.getGenre().equals(currentGenre)) {
      return originalScore * 1.2;
    }
    return originalScore;
  }
```

Assume BookManager exists

Boost estimate by 20%

Don't change anything else

Libimseti dataset : 17,359,346 anonymous ratings of 168,791 profiles made by 135,359 LibimSeTi users



Average absolute difference between Estimated and actual values based on different similarity metrics and using the nearest-n-classifier.

Similarity	n = 1	n = 2	n = 4	n = 8	n = 16	n = 32	n = 64	n = 128
Euclidean	1.17	1.12	1.23	1.25	1.25	1.33	1.48	1.43
Pearson	1.3	1.19	1.27	1.3	1.26	1.35	1.38	1.47
Log-likelihood	1.33	1.38	1.33	1.35	1.33	1.29	1.33	1.49
Tanimoto	1.32	1.33	1.43	1.32	1.3	1.39	1.37	1.41

Average absolute difference between Estimated and actual values while evaluating the user based recommender based on different similarity metrics and using the threshold based recommender.

Similarity	t = 0.95	t = 0.85	t = 0.9	t = 0.8	t = 0.75	t = 0.7
Euclidean	1.33	1.37	1.39	1.43	1.41	1.47
Pearson	1.47	1.4	1.42	1.4	1.38	1.37
Log-likelihood	1.37	1.46	1.56	1.52	1.51	1.43
Tanimoto	NaN	NaN	NaN	NaN	NaN	NaN

NAN = > Not a Number

Anonymous user problem

The problem of making recommendations to anonymous users, starting from no data is called the **Cold Start Problem.**

Approach I : Give a general predefined list of products to recommend. It is generally better than nothing and simple way of recommending.

Approach 2 : Promote these users to real users on their first visit and assign ID and track their activities based on a web session.

Problem : Exploded the number of user , users may not return (so computation going to waste)

Solution : Aggregate anonymous users, treat them as if they were a single user.

This simplifies the process of storing , based on assumption that all such users behave similarly.

The set of recommendations is stored and is computed periodically instead upon every request.

Custom Recommender

- ❖ Recommend more effectively

- ❖ Includes various preferences like food type, movies for entertainment compatibility, gender information, liking for pets, and the region from which the person belongs, the regions were divided into east, west, north and south.

- ❖ Takes into account both user based and item based recommendations.

- ❖ Different preferences were given different weights.

Custom Recommender

Sample Data Showing the location of different users

```
101,N
102,E
103,E
104,S
105,S
106,W
107,E
```

Below is shown the final output

```
For userNo - 1 ,RecommendedItem[item:106, value:6.0]
For userNo - 2 ,RecommendedItem[item:103, value:9.67399]
For userNo - 3 ,RecommendedItem[item:102, value:7.2378254]
For userNo - 4 ,RecommendedItem[item:105, value:2.8]
For userNo - 5 ,RecommendedItem[item:107, value:10.249556]
```

References

- Mahout's website, wiki and mailinglist
    - http://mahout.apache.org
    - user@mahout.apache.org
- B. Sarwar et al: „Itembased collaborative filtering recommendation algorithms“, 2001
- Manning's Early Access Program
  http://manning.com/owen
- Apache mahout Documentation and Mahout wiki

Thank You