****

**Project Report**

**Session:** 2021-2022

**Sub Session:** Semester I (Jul-Dec)

**Course:** CS-4121 (Web Intelligence & Algorithms)

**Submitted To:**

Prof. Deepika Prakash

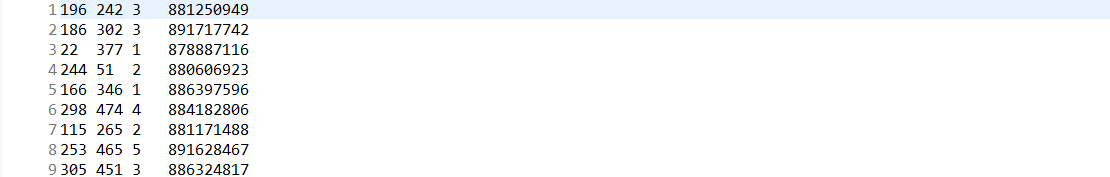
**Submitted By :**

**Name:** Swattik Chakrabarty

**Enr. No.:** BT18GCS085

**Section:** D4

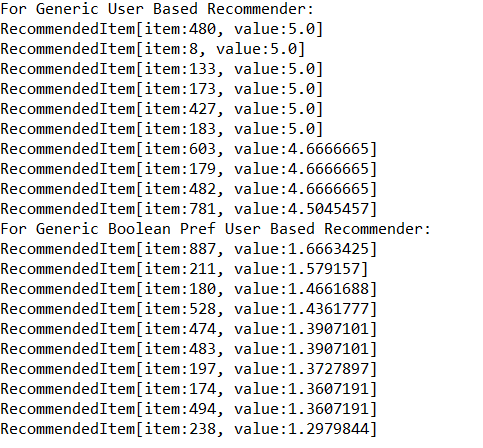
**Data:** The Data was collected from the group lens dataset website. The size of the dataset was 100,000. It was stored in the u.data file. The format of the dataset was not in csv (as required by the data model and recommendation engine)and contained extra unnecessary columns. It looked like this:



The dataset was converted to our desired format and the unnecessary formats were removed by running the **MovieDataConvert.java** file. After running it, all the data was stored in the desired format in **movies.csv**:



Two recommenders were created: **GenericUserBasedRecommender** and **GenericBooleanPrefUserBased** **Recommender**. Initially both the recommenders were run and evaluated on different UserSimilarity: Pearson Correlation for GenericUserBased and LogLikelihood for GenericBooleanPref. A UserSimilarity implementation provides some notion of how similar two users are; this could be based on one of many possible metrics or calculations. The UserNeighborhood was same for both the recommenders: NearestUserNeighborhood. 10 nearest users were used for both the recommenders. The difference noted between the recommenders according to recommendations for user 2 if 10 recommendations were predicted then:



Here we see that as Generic User Based Recommender considers preference value while recommending items it only recommends items which have a preference value of 4.5 or higher, which is the highest for this specific user among all the recommendations possible. But as Generic Boolean Pref User Based Recommender is not bound by preference values hence it recommends items completely different for the exact same user using the same dataset. The value it gives in its output is incomprehensible as it does not recommend users items according to preference values nor does it assign to items while recommending them. These values are arbitrary. It weights items that are associated with many other similar users, and with users that are more similar, more heavily. It doesn’t produce a weighted average like Generic User Based Recommender.

Apart from the above difference that we discovered we also discovered differences based on performance evaluators like **Average Absolute Difference, Precision and Recall.**

**Average absolute Difference Evaluator** is an evaluator that when implemented gives out a value which means, on average, the recommender estimates a preference that deviates from the actual preference by that particular value. It is because of this we could not compare the recommenders on this evaluator as in Generic Boolean Pref User Based recommender, the DataModel pretends that

all preference values are 1.0. The average difference between estimated and actual preferences in a world where every preference value is 1 will be 0.Hence the test itself is invalid because it will only ever result in 0.

**Precision** means the proportion of top results that are relevant, for some definition of relevant. Here we are finding out Precision at 10, which implies that this proportion is judged from the top 10 results.

**Recall** is the proportion of all relevant results included in the top results.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Average**  **Absolute**  **Difference** | **Precision** | **Recall** |
| **GenericUserBased**  **Recommender**  **(Pearson Correlation Similarity)** | 0.9897797 | 0.03319999 | 0.03897619 |
| **GenericBoolean**  **PrefUserBased**  **Recommender**  **(Log Likelihood Similarity)** | Invalid | 0.243100 | 0.30532 |

***Table :***  *Both the recommenders were compared and the*

*differences were noted.*

As we wanted to compare the recommender engines on exactly the same parameters we used Tanimoto Coefficient similarity which works with both recommenders and we noted the differences in the below table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Average**  **Absolute**  **Difference** | **Precision** | **Recall** |
| **GenericUserBased**  **Recommender** | 0.8573568 | 0.1063999 | 0.1529714 |
| **GenericBoolean**  **PrefUserBased**  **Recommender** | Invalid | 0.243529 | 0.317164 |

***Table :***  *With Tanimoto Coefficient Similarity*

Here we see that although the result for Generic Boolean Pref User Based Recommender is almost the same with little increase in performance in Recall but a significant increase in performance was noticed in all 3 evaluators for Generic User Based Recommender.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*