**Predicting Movie rating based on User Based Collaborative Filtering(UBCF)**

*- by Ratul Ramchandani*

In this case study, we are trying to predict the ratings for the movies that the user has not rated. The predictions are derived from the similar movie viewing patterns of other users in the group who have rated the movies. This technique is also known as **User Based Collaborative Filtering (UBCF)**.

We will be using a the “ratings.txt” file which has the ratings for all the movies watched by a user, from the movielens dataset. It contains 100,000 lines of data of which the first few lines are seen below:

|  |  |  |  |
| --- | --- | --- | --- |
| **userId** | **movieId** | **rating** | **timestamp** |
| 1 | 31 | 2.5 | 1260759144 |
| 1 | 1029 | 3 | 1260759179 |
| 1 | 1061 | 3 | 1260759182 |
| 1 | 1129 | 2 | 1260759185 |
| 1 | 1172 | 4 | 1260759205 |
| 1 | 1263 | 2 | 1260759151 |

We will split the data into training and the test data by using stratified random sampling. After doing so, our predicted dataset would look like as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **userId** | **movieId** | **rating** | **predicted\_rating** |
| 1 | 31 | 2.5 | 2 |
| 1 | 1029 | 3 | 3 |
| 1 | 1061 | 3 | 3 |
| 1 | 1129 | 2 | 2 |
| 1 | 1172 | 4 | 3 |
| 1 | 1263 | 2 | 3 |

To achieve the prediction of the ratings, we will apply User Based Collaborative Filtering (UBCF) technique. An assumption that this approach uses is that the users with similar preferences will rate items similarly, consequently predicting the missing ratings for every user. It does so by first finding neighbouring users of similar patterns and then aggregating their ratings to form a prediction.

We will apply three popular similarity measures Jaccard, Pearson and Cosine similarity methods. The R-package: **Recommenderlab** has been used for this case study. The complete documentation can be found [here](https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf). The prediction algorithm will be modelled on the training data, which is 80% and then create a prediction for the remaining 20% which is will be used as the test data for every row ID.

The above steps are performed for all the three similarity metrics, one at a time. Post the predictions, we will calculate the Normalized Mean Absolute Error (NMAE) for all the models and do a comparison. The seed set here for the sampling is 12.

Below is the execution report:

|  |  |
| --- | --- |
| **Execution Report** | |
| Package used | Recommenderlab |
| Sampling method | Stratified Random Sampling |
| Number of records in data set | 100000 |
| Number of unique users | 671 |
| Number of unique movies | 9066 |
| Training set – % of records | 80% |
| Testing set – % of records | 20% |
| Sampling Seed | 12 |

NMAE scores for the 3 models:

|  |  |
| --- | --- |
| **Similarity Metric** | **NMAE** |
| Cosine | 0.1850611 |
| Jaccard | 0.1846834 |
| Pearson | 0.1852389 |

Analysis of different models:

Since the data NMAE scores, approx. 18% for all, turn out to be very similar for the given dataset, it is difficult to determine the best model. This translates to an absolute error value of around 0.8 rating on an average.

Details of attached files:

|  |  |
| --- | --- |
| **File name** | **Contents** |
| Ratul\_RS\_code.R | File containing R code |
| ratings.csv | MovieLens data set small |
| output\_cosine.csv | Ouput of predictions based on UBCF Cosine method |
| output\_jaccard.csv | Ouput of predictions based on UBCF Jaccard method |
| output\_pearson.csv | Ouput of predictions based on UBCF Pearson method |
| recordings.xlsx | Data for all the tables in the report. |

The github link for the files can be found here -- <https://github.com/creativecoderr/recommenderSystems.git>