**HW4 – Recommendation System**   
Submitted by: Prachi Hada

**Part A: Data pre-processing**

To create movies.csv from Movie Night dataset:

1. Removed all columns except - id, rating, movie name from [dataset](https://docs.google.com/spreadsheets/d/1DPZkamfbd-V7rfg25GsuvKRvj9T4J0FQ1cKL-VKIpqo/edit#gid=1774088842)
2. Since id were not standardized, so used last 3 digits and gave movie names numeric id.
3. Denormalized dataset (flattened the file) so that input dataset has 3 columns - userid, movieId and, rating. There are 1113 records (53 users \* 21 movies).
4. Checked if the value is missing. Removed records with missing ratings. There are 637 records after removing missing value.
5. Created a csv file without header for mahout and with header for Spark.
6. After I have ran various algorithms, I have performed same processing using normalized (mean-centered) data set which is preprocessed using R.

Other checks performed during preprocessing:

1. Check for duplicates for (userid, movieid) - No duplicate combination were found. Checked using Excel (Select columns, Data->Remove Duplicates)
2. Data Validation was done to check that value of rating column lies between 1 to 5.

To create movies21.csv from Movie Night dataset for Matrix Factorization:

1. Selected movie name from top row of Movie Night dataset, pivoted the horizontal column into vertical
2. Added another column movieId.

**Part B:**

Algorithms performed for User-based and Item based Collaborative Filtering:

1. Pearson Correlation Similarity
2. Log Likelihood Similarity
3. Tanimoto Coefficient Similarity
4. Euclidean Distance Similarity

For testing these algorithms, the factor which has been modified is training split – 0.7, 0.8, 0.9.

For User based Collaborative filtering, neighborhood has been used as 3.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Algorithm Name | Training % | Evaluation % | Neighbourhood | RMSE | Precision | Recall | F-Score |
| User | PearsonCorrelation | 0.7 | 1 | 3 | 1.256732 | 0.95 | 0.45 | 0.610714 |
| User | PearsonCorrelation | 0.8 | 1 | 3 | 1.224452 | 0.95 | 0.45 | 0.610714 |
| User | PearsonCorrelation | 0.9 | 1 | 3 | 1.192319 | 0.95 | 0.45 | 0.610714 |
| User | LogLikelihood | 0.7 | 1 | 3 | 1.3925 | 0.957672 | 0.616667 | 0.750238 |
| User | LogLikelihood | 0.8 | 1 | 3 | 1.351347 | 0.957672 | 0.616667 | 0.750238 |
| User | LogLikelihood | 0.9 | 1 | 3 | 1.474438 | 0.957672 | 0.616667 | 0.750238 |
| User | TanimotoCoefficient | 0.7 | 1 | 3 | 1.411682 | 0.981481 | 0.716667 | 0.828426 |
| User | TanimotoCoefficient | 0.8 | 1 | 3 | 1.272747 | 0.981481 | 0.716667 | 0.828426 |
| User | TanimotoCoefficient | 0.9 | 1 | 3 | 1.243418 | 0.981481 | 0.716667 | 0.828426 |
| User | EuclideanDistance | 0.7 | 1 | 3 | 1.148566 | 0.972222 | 0.633333 | 0.767013 |
| User | EuclideanDistance | 0.8 | 1 | 3 | 1.199972 | 0.972222 | 0.633333 | 0.767013 |
| User | EuclideanDistance | 0.9 | 1 | 3 | 0.978019 | 0.972222 | 0.633333 | 0.767013 |
| Item | LogLikelihood | 0.7 | 1 |  | 1.163044 | 0.95 | 0.95 | 0.95 |
| Item | LogLikelihood | 0.8 | 1 |  | 1.129377 | 0.95 | 0.95 | 0.95 |
| Item | LogLikelihood | 0.9 | 1 |  | 0.919826 | 0.95 | 0.95 | 0.95 |
| Item | TanimotoCoefficient | 0.7 | 1 |  | 1.079804 | 0.966667 | 0.966667 | 0.966667 |
| Item | TanimotoCoefficient | 0.8 | 1 |  | 1.068958 | 0.966667 | 0.966667 | 0.966667 |
| Item | TanimotoCoefficient | 0.9 | 1 |  | 0.969215 | 0.966667 | 0.966667 | 0.966667 |
| Item | PearsonCorrelation | 0.7 | 1 |  | 1.528388 | 0.95 | 0.95 | 0.95 |
| Item | PearsonCorrelation | 0.8 | 1 |  | 1.293239 | 0.95 | 0.95 | 0.95 |
| Item | PearsonCorrelation | 0.9 | 1 |  | 1.236085 | 0.95 | 0.95 | 0.95 |
| Item | EuclideanDistance | 0.7 | 1 |  | 1.108224 | 0.95 | 0.95 | 0.95 |
| Item | EuclideanDistance | 0.8 | 1 |  | 1.127989 | 0.95 | 0.95 | 0.95 |
| Item | EuclideanDistance | 0.9 | 1 |  | 1.02802 | 0.95 | 0.95 | 0.95 |

**User Based Collaborative Filtering best Algorithm: Euclidean Distance Similarity**

It has the lowest RMSE for all the splits. It also has good F-score. Movies recommended to me using Euclidean Similarity are: Movie Id 2 (Fight Club) (Recommendation Score 4.5) and Movie Id 7(Hangover) (Recommendation Score 3.8). Following table (extracted from bigger table) is sorted on RMSE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EuclideanDistanceSimilarity | 0.9 | 1 | 3 | 0.978019 | 0.972222 | 0.633333 | 0.767013 |
| EuclideanDistanceSimilarity | 0.7 | 1 | 3 | 1.148566 | 0.972222 | 0.633333 | 0.767013 |
| PearsonCorrelationSimilarity | 0.9 | 1 | 3 | 1.192319 | 0.95 | 0.45 | 0.610714 |
| EuclideanDistanceSimilarity | 0.8 | 1 | 3 | 1.199972 | 0.972222 | 0.633333 | 0.767013 |

**Item Based Collaborative Filtering best Algorithm: Tanimoto Coefficient Similarity**

LogLikelihoodSimilarity has smallest RMSE (0.91) when training data percentage is 0.9

But TanimotoCoefficientSimilarityhas RMSE (0.96) when training data percentage is 0.9 which is next lowest RMSE. Not only this, it has low RMSE for 0.8, 0.7 split and has higher F-score than LogLikelihoodSimilarity. Following table (extracted from bigger table) is sorted on RMSE.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LogLikelihoodSimilarity | 0.9 | 1 |  | 0.919826 | 0.95 | 0.95 | 0.95 |
| TanimotoCoefficientSimilarity | 0.9 | 1 |  | 0.969215 | 0.966667 | 0.966667 | 0.966667 |
| EuclideanDistanceSimilarity | 0.9 | 1 |  | 1.02802 | 0.95 | 0.95 | 0.95 |
| TanimotoCoefficientSimilarity | 0.8 | 1 |  | 1.068958 | 0.966667 | 0.966667 | 0.966667 |
| TanimotoCoefficientSimilarity | 0.7 | 1 |  | 1.079804 | 0.966667 | 0.966667 | 0.966667 |

Using TanimotoCoefficientSimilarity, movies similar to Movie id 1 are as follows:

ItemID 1,MyRecommendedItem 3,RecommendationScore 0.70454544

ItemID 1,MyRecommendedItem 15,RecommendationScore 0.63829786

ItemID 1,MyRecommendedItem 8,RecommendationScore 0.63829786

ItemID 1,MyRecommendedItem 5,RecommendationScore 0.63265306

ItemID 1,MyRecommendedItem 11,RecommendationScore 0.60465115

**Part C:**

For this I have changed the split for 3 different ranks (features). After sorting data on RMSE, the best model is rank 20 on training data, validation data, test data split of 0.8, 0.1, 0.1 respectively.

Following table is sorted on RMSE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Training | Validation | Test | Rank | RMSE |
| Matrix Factorization | 0.8 | 0.1 | 0.1 | 20 | 0.987581 |
| Matrix Factorization | 0.8 | 0.1 | 0.1 | 15 | 0.995047 |
| Matrix Factorization | 0.8 | 0.1 | 0.1 | 12 | 1.005887 |
| Matrix Factorization | 0.6 | 0.2 | 0.2 | 20 | 1.10546 |
| Matrix Factorization | 0.6 | 0.2 | 0.2 | 15 | 1.109112 |
| Matrix Factorization | 0.6 | 0.2 | 0.2 | 12 | 1.110865 |
| Matrix Factorization | 0.7 | 0.2 | 0.1 | 20 | 1.135408 |
| Matrix Factorization | 0.7 | 0.2 | 0.1 | 12 | 1.141068 |
| Matrix Factorization | 0.7 | 0.2 | 0.1 | 15 | 1.148748 |

Movies recommended for me for best RMSE are as follows:

|user|movieId| title| rating|

| 858| 2| Fight Club| 4.840835691412698|

| 858| 12| Pulp Fiction| 4.666836407620735|

| 858| 9|The big Lebowski| 4.653912486440795|

| 858| 10| Almost Famous|4.5528879725719555|

| 858| 19| Ghostbusters| 4.427450639981219|