Question 3)

Recommending based on user based collaborative filtering,

Image of MovieLense dataset for 100 user 100 movies and ratings

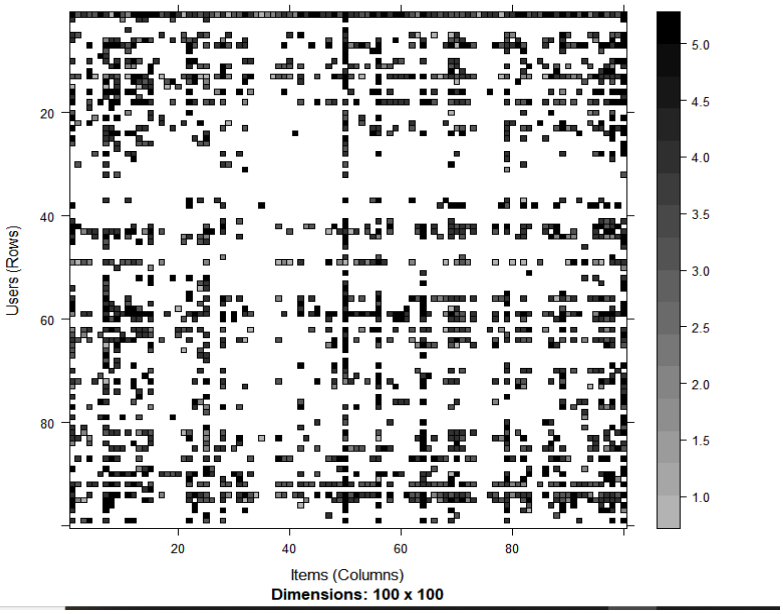
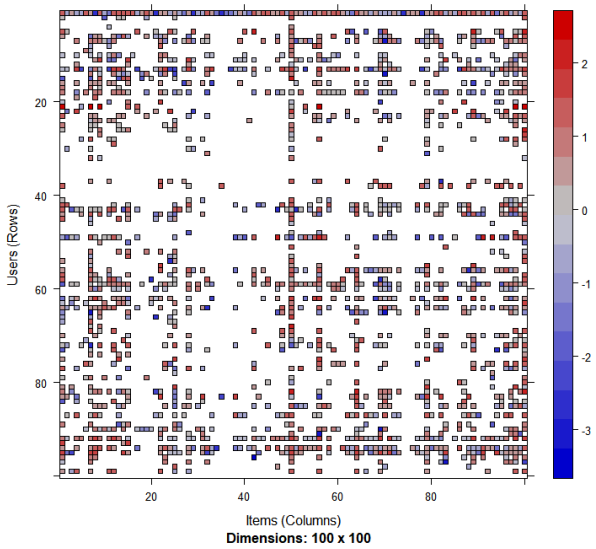


Image of normalized MovieLense dataset



The data set for training is created using 75% of the data and 25% of the data is used for testing.

```
> eval <- evaluationScheme(R_Normalize, method = "split", given = 3, train = 0.75, goodRating = 4)
> eval
Evaluation scheme with 3 items given
Method: 'split' with 1 run(s).
Training set proportion: 0.750
Good ratings: >=4.000000
Data set: 943 x 1664 rating matrix of class 'realRatingMatrix' with 99392 ratings.
Normalized using center on rows.
```

- Recommendation system ->**User based collaborative filtering approach**
- **Cosine similarity** is used
- Function predict() is used on the known ratings
- Model learnt data of 707 users that is 75% of total users
- Prediction is made on 25% of the known test data

```
> userbased_model
Recommender of type 'UBCF' for 'realRatingMatrix'
learned using 707 users.
> user_rating <- predict(userbased_model, getData(eval, "known"), type = "ratings")   #Make predictions on ratings
> user_rating
236 x 1664 rating matrix of class 'realRatingMatrix' with 378708 ratings.
>
```

Predicted ratings for 1st 5 users and movies are shown below:

```
> getRatingMatrix(user_rating)[1:5,1:5]
5 x 5 sparse Matrix of class "dgCMatrix"
   Toy Story (1995) GoldenEye (1995) Four Rooms (1995) Get Shorty (1995) Copycat (1995)
8         0.7087328      0.511736418        0.52365518         0.4514746     0.52750612
25        .              .                  .                  .             .
27        0.6365097      0.425937780        0.30918228         0.4797549     0.38636604
32        0.3264952      0.006043162        0.01197689         0.1220685     0.02698086
36       -0.9221379     -1.242261866       -1.24206812        -1.3624935    -1.22119323
```

Calculation of error using unknown ratings:

Error between known and unknown set is calculated as follows:

```
> pred_error <- rbind(UBCF = calcPredictionAccuracy(user_rating, getData(eval, "unknown")))  #Calculate error for qu
estion 3
> pred_error
         RMSE      MSE       MAE
UBCF 1.168049 1.364339 0.9256433
>
```

- Root mean Square Error (RMSE) for the method is 1.68049
- Mean Average Error is 0.9256433

Question 4) Testing performance using **cross validation**

Data is divided into 5 parts as given in the question :

3parts < - training set

1part<- testing, 1part <- validation

```
> scheme_1 <- evaluationScheme(MovieLense, method = "cross", k = 5, given = 3, goodRating = 4)
> scheme_1
Evaluation scheme with 3 items given
Method: 'cross-validation' with 5 run(s).
Good ratings: >=4.000000
Data set: 943 x 1664 rating matrix of class 'realRatingMatrix' with 99392 ratings.
>
```

User based collaboration filtering Recommender system is created using **cosine similarity**

Training the dataset:

```
> userbased_model_4 <- Recommender(getData(scheme_1, "train"), "UBCF", param = list(method = "cosine",nn = 50))
> userbased_model_4
Recommender of type 'UBCF' for 'realRatingMatrix'
learned using 752 users.
```

Prediction on the known ratings in the test dataset:

```
> pred_rating_4 <- predict(userbased_model_4, getData(scheme_1, "known"), type = "ratings")
> as(pred_rating_4,"matrix")[1:10,1:2]
   Toy Story (1995) GoldenEye (1995)
1          4.138605          3.972549
2          3.739158          3.636776
17         2.873343          2.958476
18         3.475680          3.228391
19         4.000000          4.000000
26         2.923625          2.584922
29         3.764436          3.644178
33         3.449760          3.316200
47         2.856153          2.577535
51         2.594513          2.312590
```

Sparse Matrix for ratings for cross validation for 2 movies and 10 users

Error with unknown ratings is calculated:

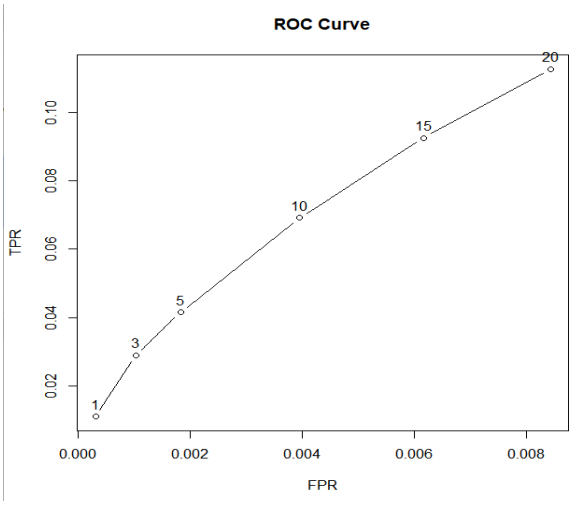
```
> pred_error_4 <- rbind(UBCF = calcPredictionAccuracy(pred_rating_4, getData(scheme_1, "unknown")))
> pred_error_4
         RMSE      MSE       MAE
UBCF 1.133309 1.284389 0.9032277
>
```

- Root mean Square Error (RMSE) is 1.13333
- Mean Average error (MAE) is 0.9032277

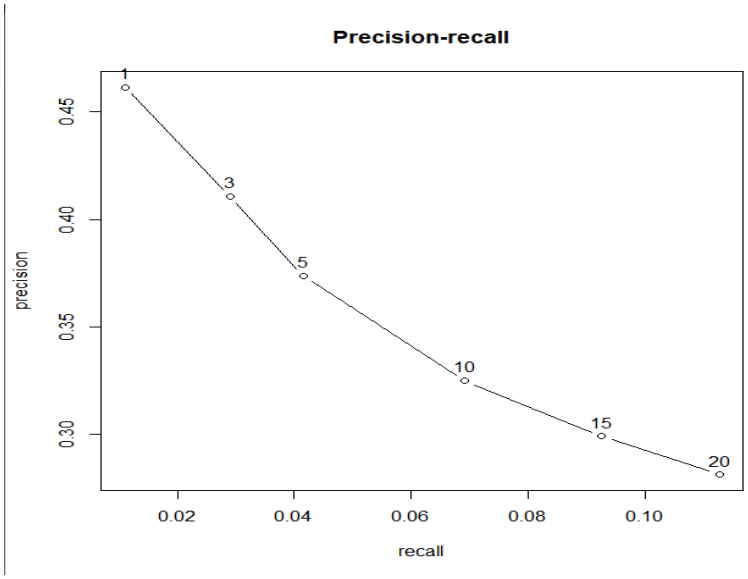
Testing Performance of the system:-

We have the Following Roc measures and curve for the recommendations

```
> results_4 <- evaluate(scheme_1, method = "UBCF", n = c(1,10,13,18,20,24), param = list(method = "cosine"))
UBCF run fold/sample [model time/prediction time]
         1  [0.01sec/1.14sec]
         2  [0.01sec/0.86sec]
         3  [0.02sec/1.17sec]
         4  [0.01sec/0.86sec]
         5  [0sec/1.03sec]
> avg(results_4)
          TP        FP       FN       TN precision     recall        TPR          FPR
1  0.4251309  0.539267 56.22094 1603.815 0.4408414 0.01063407 0.01063407 0.0003322826
10 3.1926702  6.451309 53.45340 1597.903 0.3310394 0.07070170 0.07070170 0.0039840793
13 3.9151832  8.621990 52.73089 1595.732 0.3122817 0.08457794 0.08457794 0.0053265465
18 5.0471204 12.312042 51.59895 1592.042 0.2907335 0.10470205 0.10470205 0.0076090765
20 5.4701571 13.817801 51.17592 1590.536 0.2835876 0.11263336 0.11263336 0.0085413384
24 6.2282723 16.917277 50.41780 1587.437 0.2690846 0.12593037 0.12593037 0.0104609228
```



Precision Plot of the



Performance:

- True positive ratio(TPR) and False positive ratio(FPR) is very low for cross validation. Therefore , Performance of system is not good.
- Also true negative and false negative for the system is very high which is not good for the system.
- Root Mean Square Error(RMSE) of Cross validation is much smaller than Model in Question 3 , Therefore Cross validation performs better. Since RMSE penalizes larger errors stronger than smaller errors, therefore in case of cross validation model larger error are relatively low.
- MAE (Mean Average error) for cross validation is also relatively low for cross validation