A Group Recommendation System for Music Using Implicit User Feedback

Te Lan (he/him)
MSc Data Science & AI for the Creative Industries

2022

A system recommending songs to a group of people at events

e.g. dinner parties, small gatherings, team-buildings…

Processing dataset of implicit user feedback (play counts)

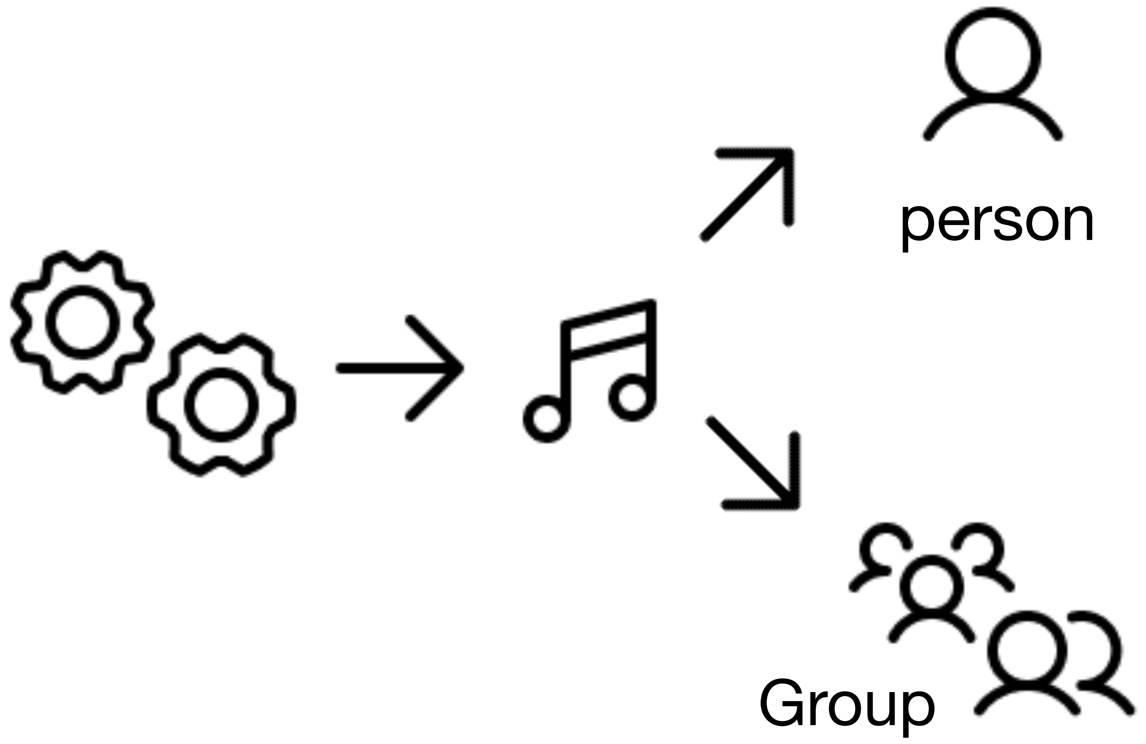
Training a model to learn the preferences from the dataset

Predict new songs that the group may like through different methods

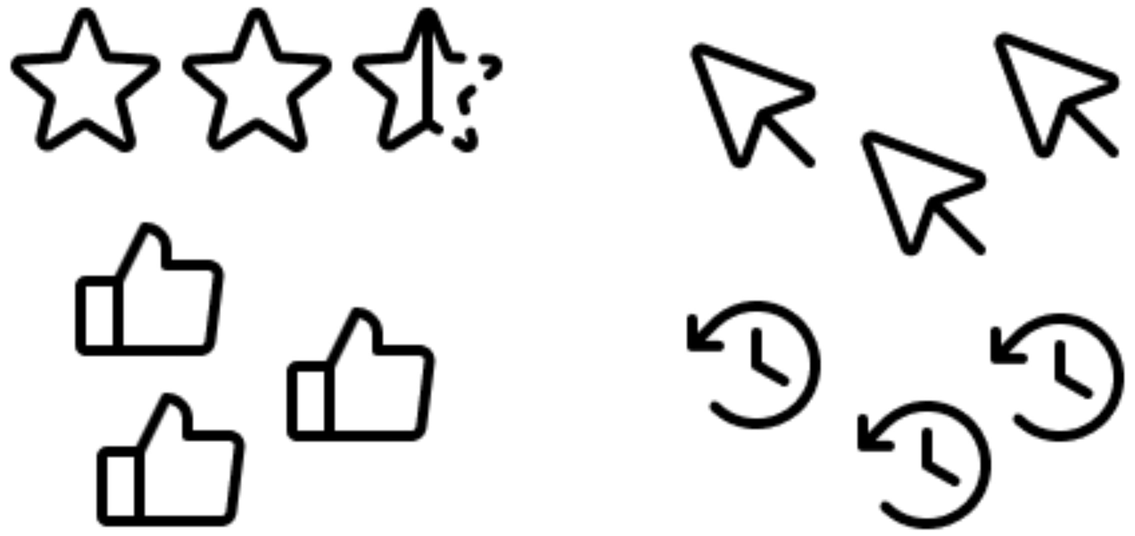
Generating explanations for predictions

Research background

Music Recommendation System & Group Recommendation System



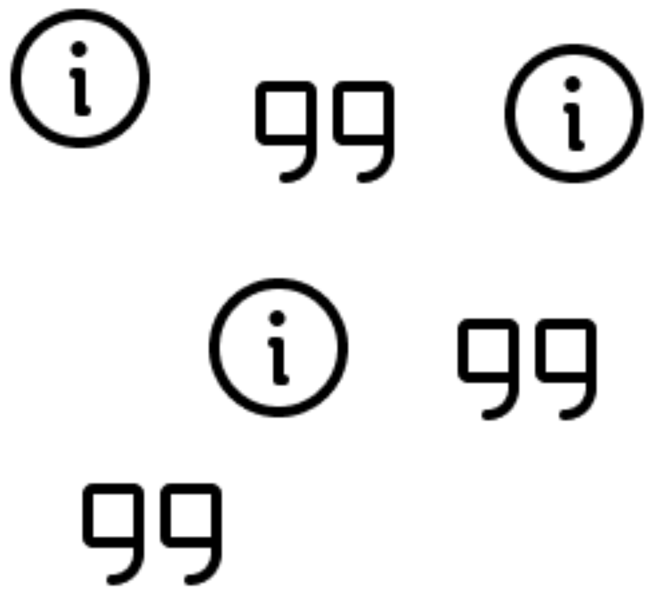
Explicit Feedback & Implicit Feedback as user data



Ratings
Likes

Clicks
Histories

Explainable Recommendation System



“Because you listened to…”

Technical challenges

1

Implementing implicit user feedback to an algorithm designed for explicit user feedback

Unfortunately, the majority of the work in matrix factorization is centered on high-quality explicit feedback datasets, in which users make their preferences known by directly rating subsets of accessible items on a fixed scale (Hu et al., 2008). However, these explicit ratings are not available in many real-world situations.

2

Assessing different group recommendation methods on new dataset

The group recommendation approaches can be roughly differentiated into two, the Pseudo-user approach (Kim and Lee, 2014) and the Consensus approach (Villavicencio et al., 2016; Kim and El Saddik, 2015).

3

Generating explanations for recommendations

It is acknowledged that explainability is essential between Human-AI interaction for the purpose of enhancing people's understanding of the system and building trust (Q. Vera et al, 2021).

Preparing the dataset

Implicit user feedback

Echo Nest
Taste Profile Dataset

*a subset of the Million Song Dataset

1,019,318
Users

384,546
Archived songs

48,373,586
user - song - play count triplets

	user_id	music_id	playcount
0	0d0f80a34807aab31a3521424d456d30bf2c93d9	SOAEHEX12A8C13EFA4	21
1	0d0f80a34807aab31a3521424d456d30bf2c93d9	SOZAFNE12AAF3B50E6	12
2	da681f423e8574d2581534fd138eae264dea0f5c	SOAQGES12A8C133FB5	9
3	da681f423e8574d2581534fd138eae264dea0f5c	SOBEJBH12AC468A455	26
4	da681f423e8574d2581534fd138eae264dea0f5c	SOBNCNF12A8C13F62E	12
...	...	...	...

0.5% Top active user (1,353)
2% Top songs (4,362)
41,062 triplets

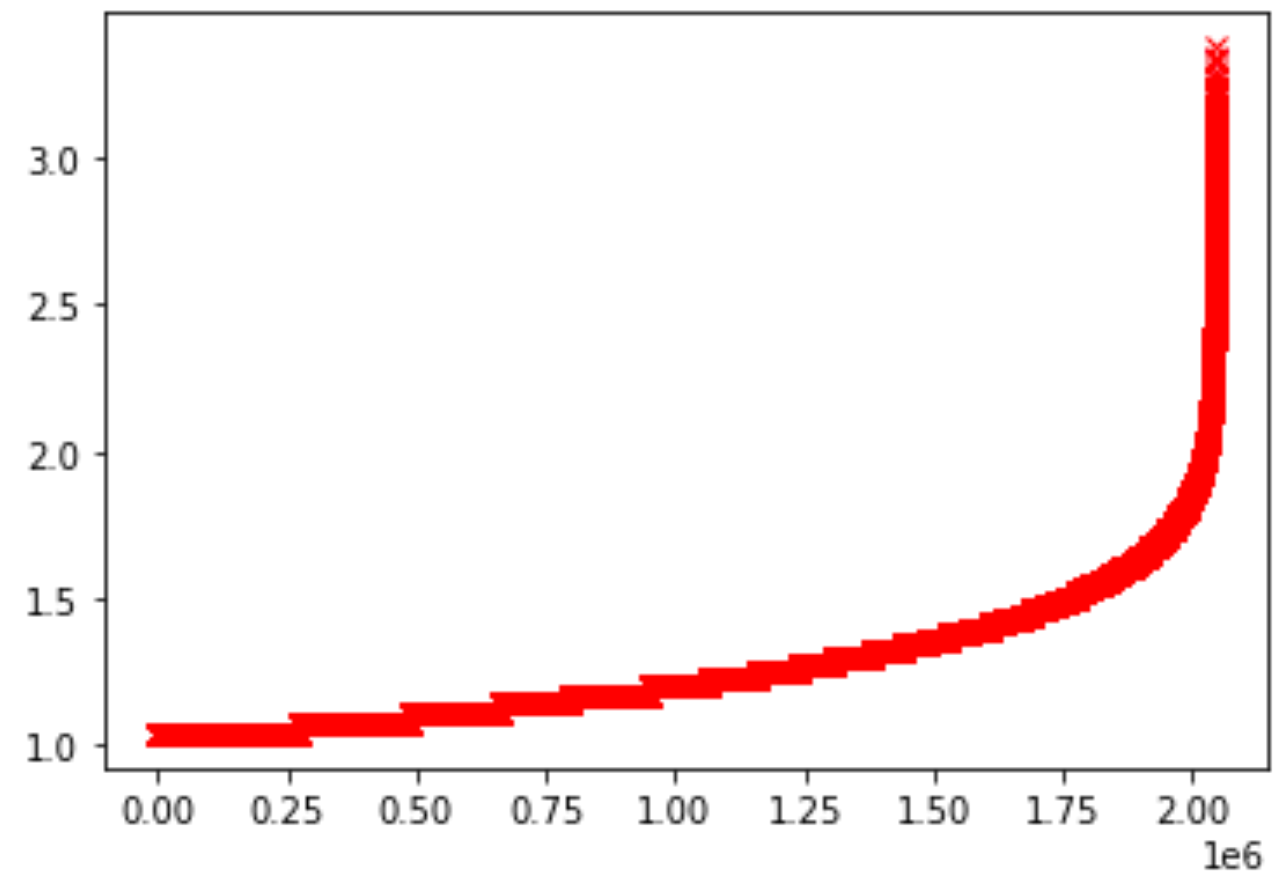
EN41K Dataset
Density: 0.07%

	user_id	music_id	ratings
0	0	0	4.000000
1	0	1	2.545455
2	1	2	4.000000
3	1	3	3.960265
4	1	4	3.920530
...	...	...	...

Preparing the dataset
continues…

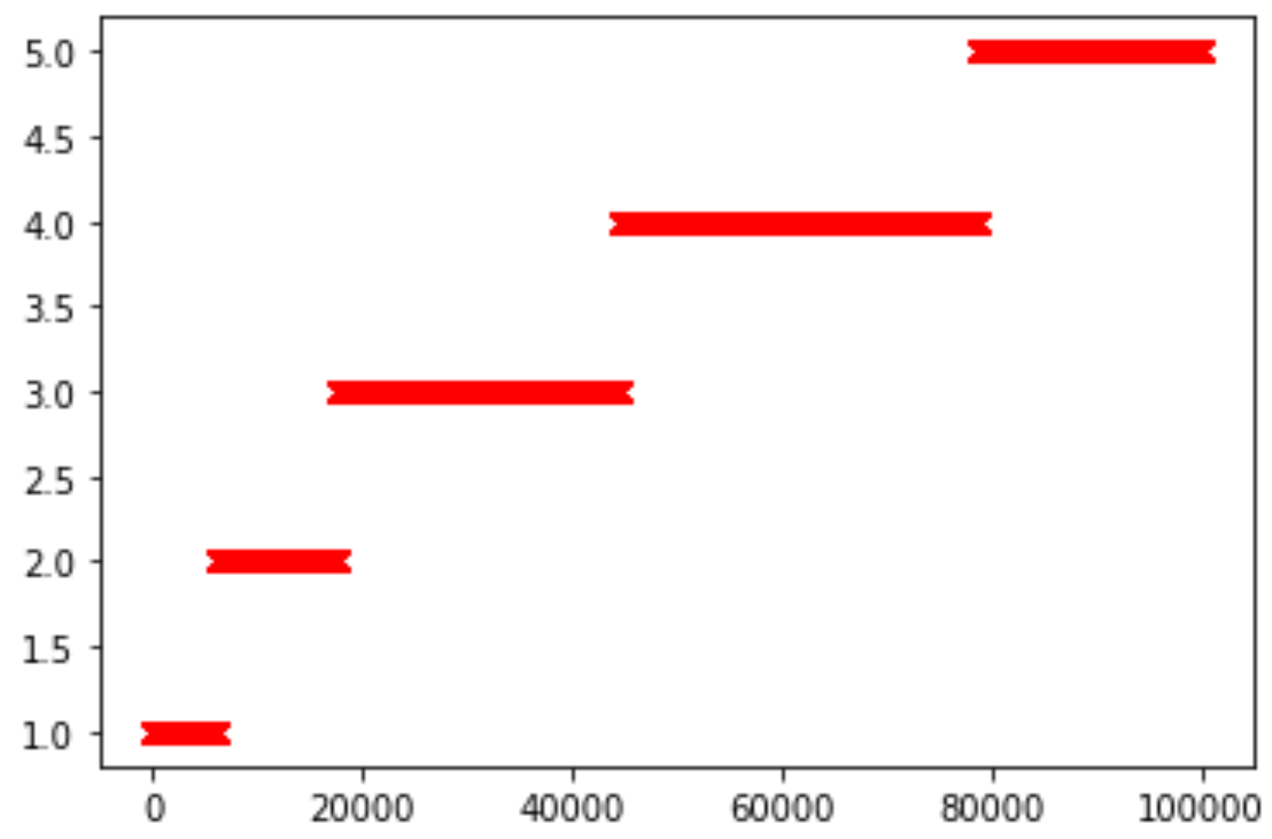
$$freq_{i,j} = \frac{count(i,j)}{\sum_{j'} count(i,j')}$$

$$r_{i,j} = 4 \cdot \left(1 - \sum_{k'=1}^{k-1} freq_{k'}(i)\right)$$



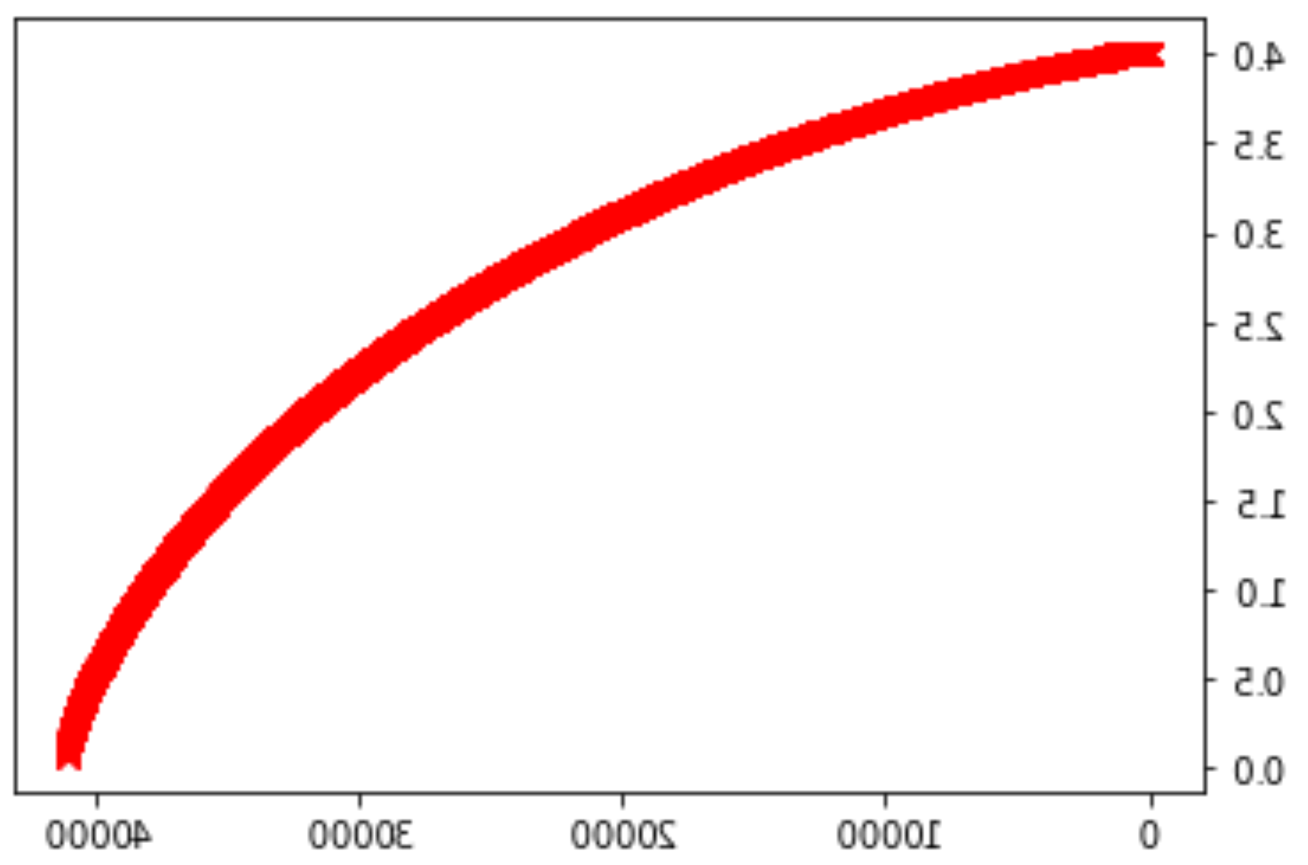
Original

Echo Nest Tast Profile Dataset
Play count distribution



Goal

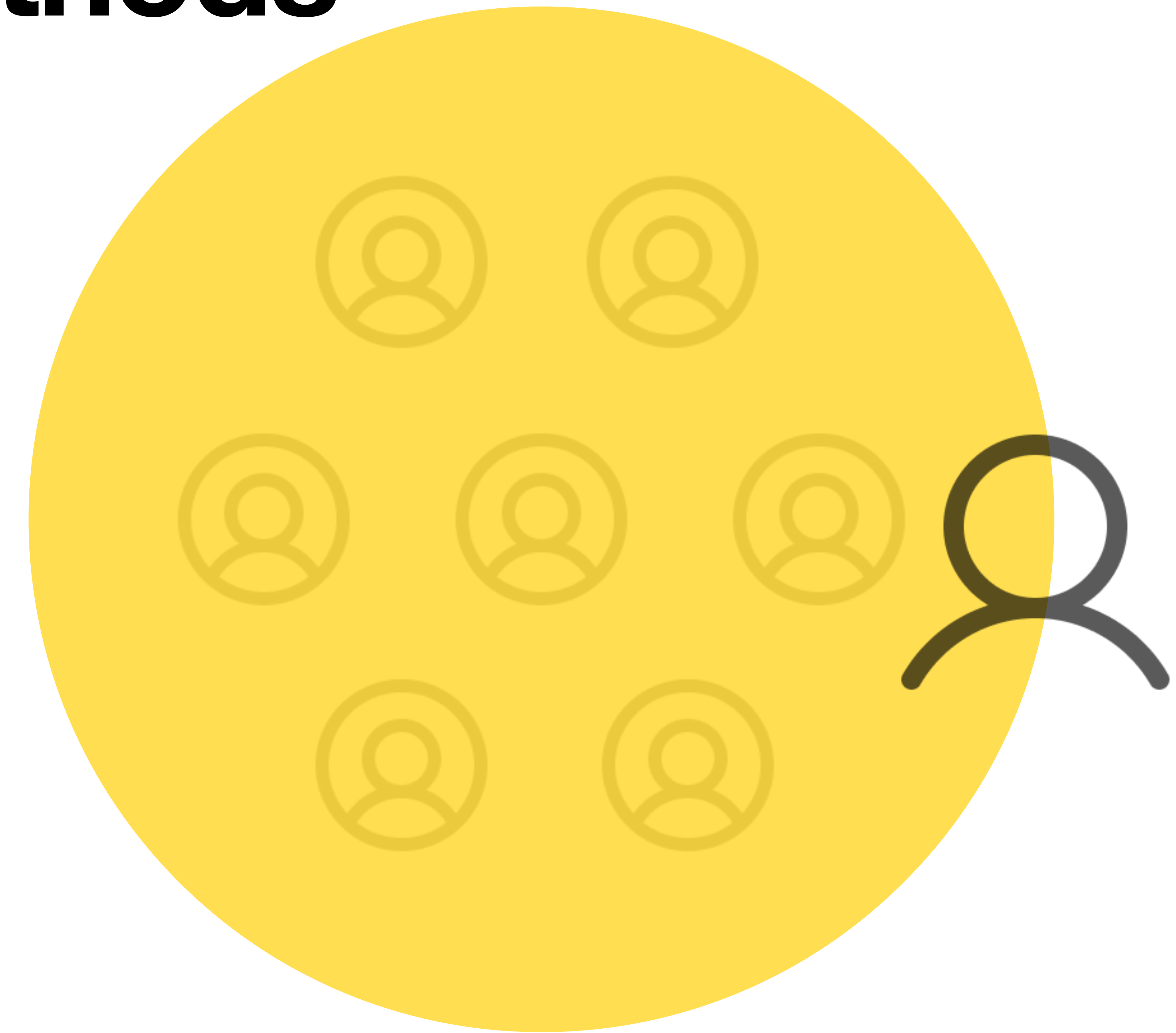
MovieLens 100K
Ratings distribution



Result

EN41K
Ratings distribution

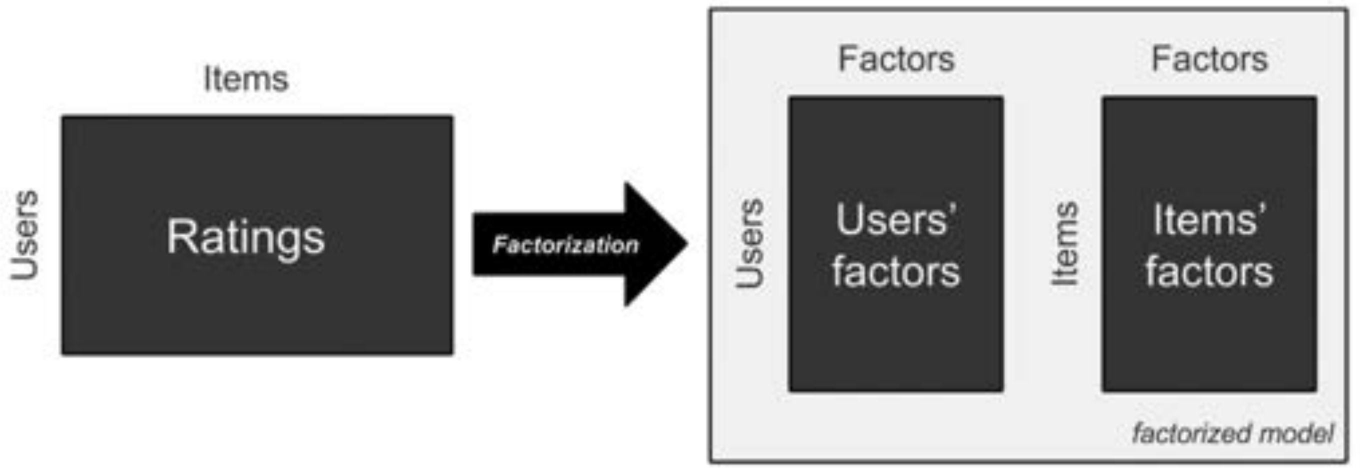
Group recommendation methods



Pseudo-user approach
(Virtual user)

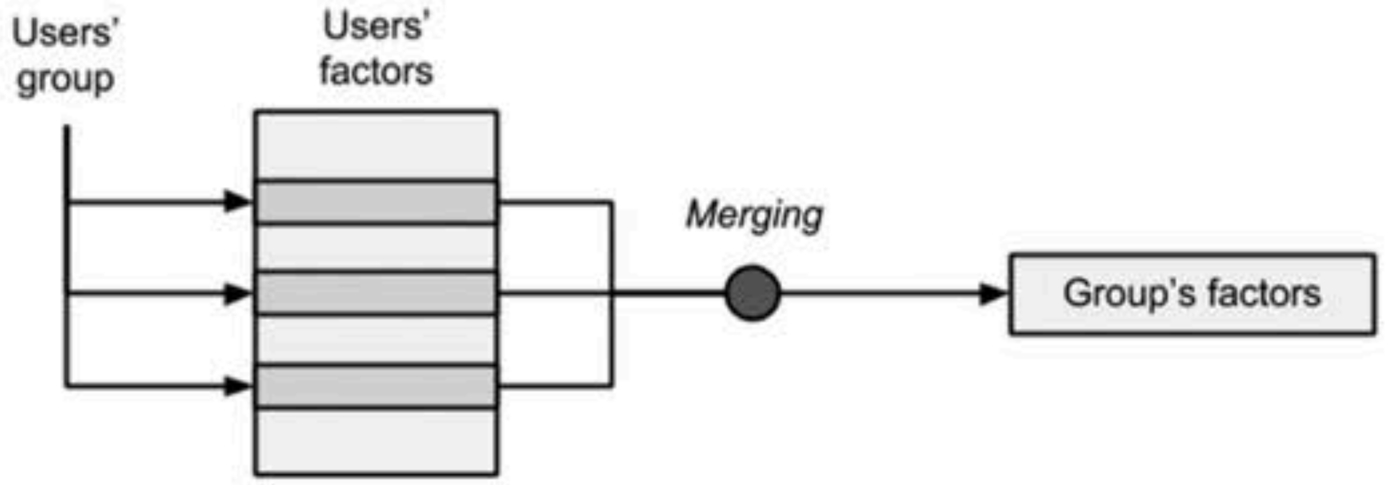
Consensus approach
(Simulating negotiation)

Group recommendation methods in detail

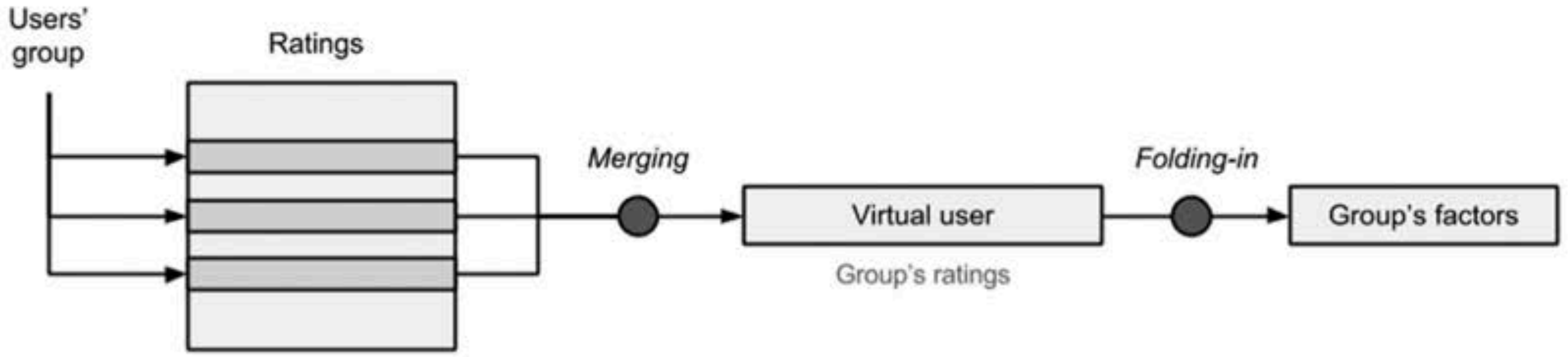


Matrix Factorisation

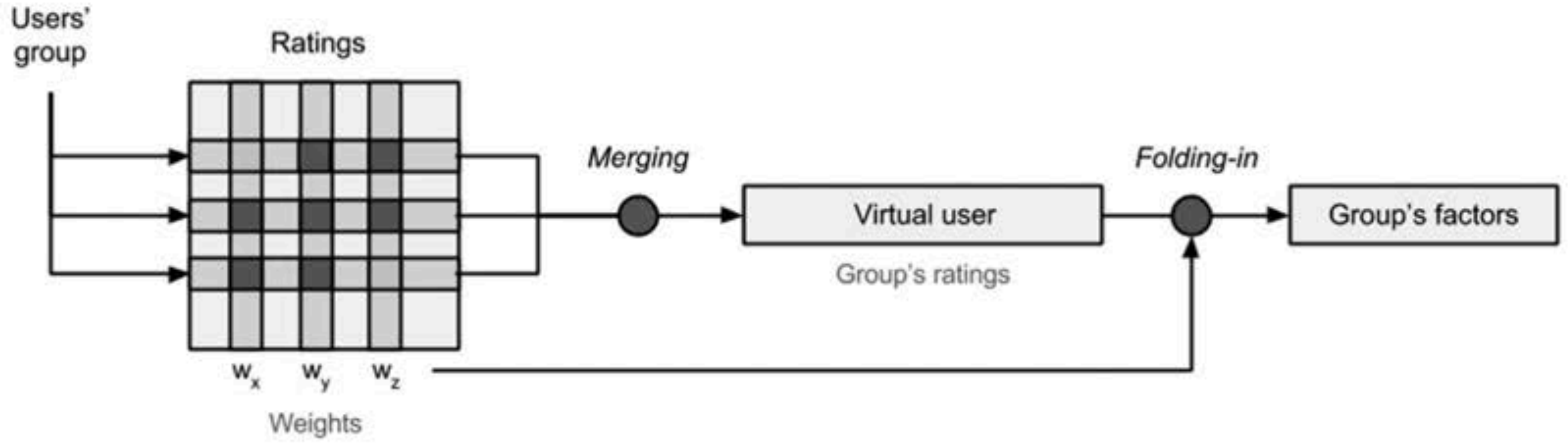
After Factorisation (AF)



Before Factorisation (BF)



Weighted Before Factorisation (WBF)



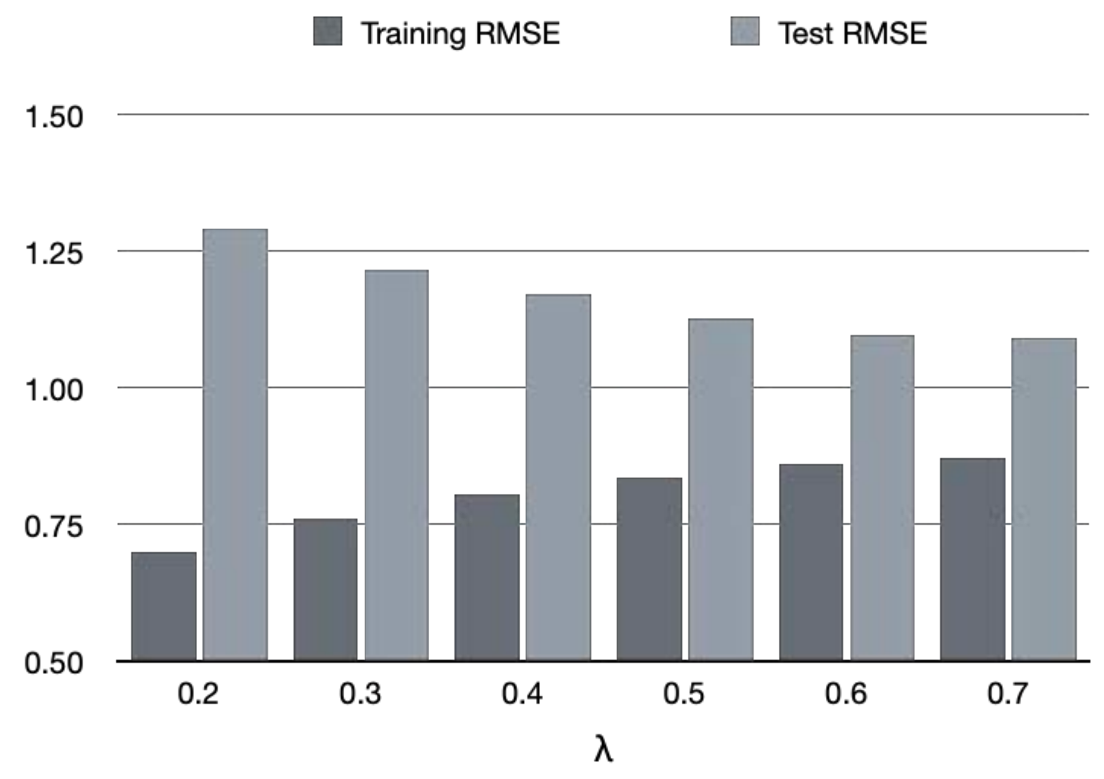
Training process review

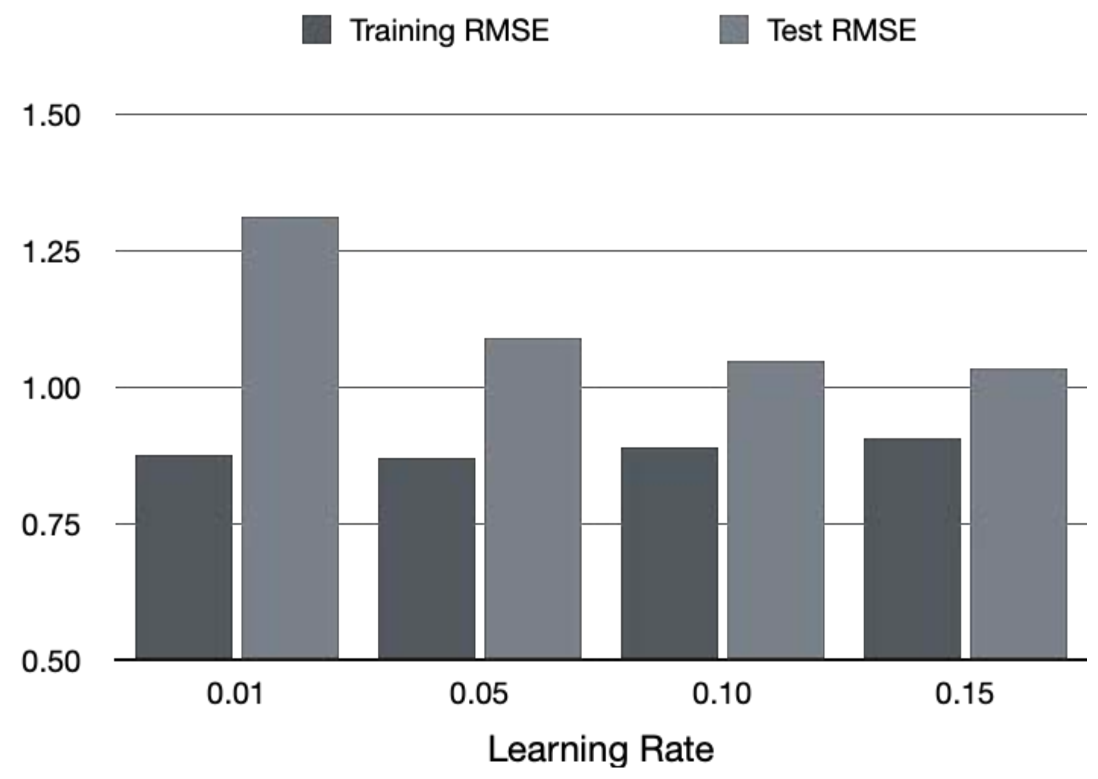
Optimising training parameters for Matrix Factorisation

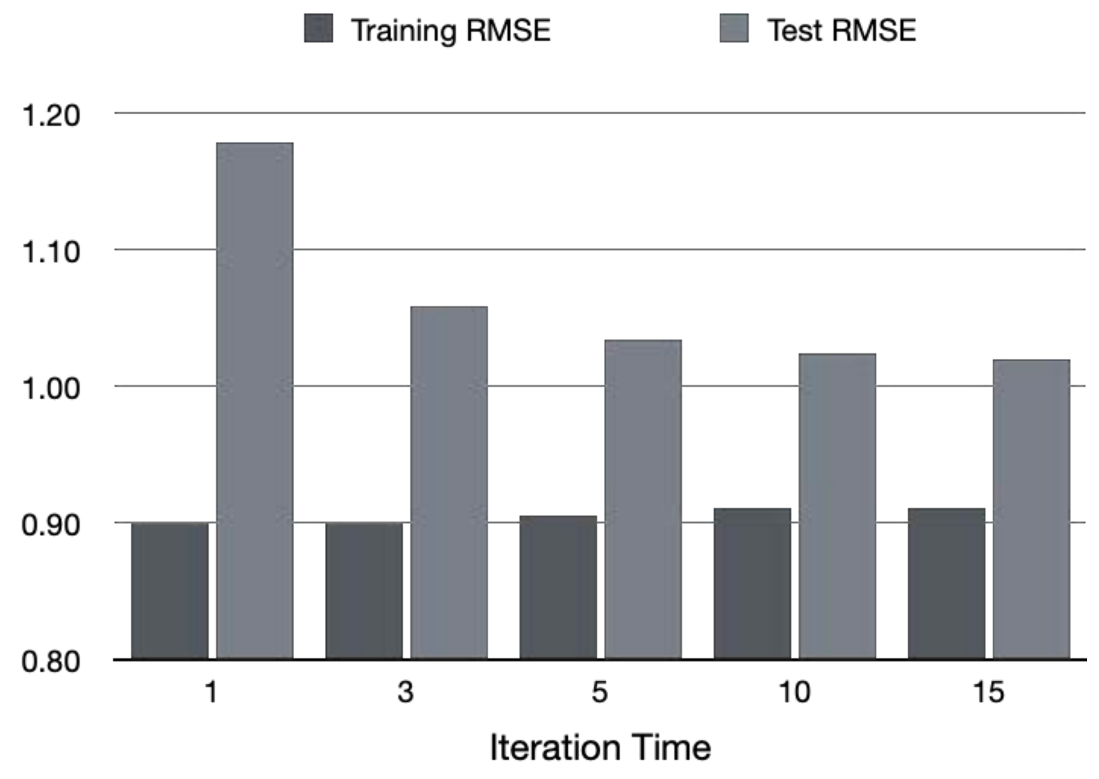
Parameter	Value
λ	0.7
Learning rate	0.15
Feature numbers	5
Iteration times	5

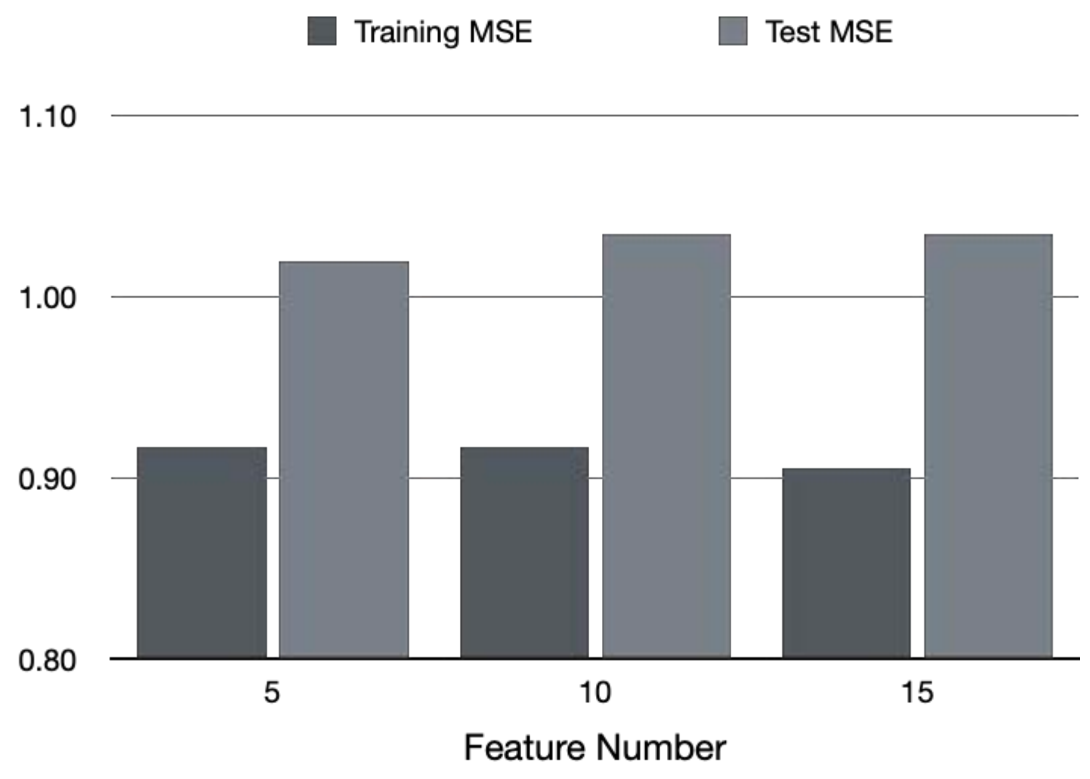
Training results

Dataset	Root Mean Square Error (RMSE)
Training	0.91447380953
Test	1.0147757779









Experiment setup

Group generation

Small group size: 3
Medium group size: 5
Large group size: 10
Group number: 10

Recommendation parameters

Recommendation number: 25
Ratings threshold: 3.0

```
******* Running for  small  groups *************
generated groups (only first 5 are getting printed here):
[310, 1058, 1164]
[398, 756, 1319]
[39, 1019, 1287]
[559, 618, 1078]
[305, 796, 1311]

******* Running for  medium  groups *************
generated groups (only first 5 are getting printed here):
[421, 509, 818, 949, 1270]
[214, 349, 661, 702, 1248]
[333, 709, 1147, 1242, 1259]
[380, 582, 647, 819, 1302]
[209, 303, 471, 1151, 1215]

******* Running for  large  groups *************
generated groups (only first 5 are getting printed here):
[278, 373, 396, 584, 709, 979, 1153, 1171, 1181, 1224]
[83, 221, 314, 554, 603, 646, 737, 975, 995, 1344]
[37, 660, 780, 785, 848, 922, 954, 1003, 1021, 1148]
[163, 238, 551, 869, 878, 1132, 1188, 1249, 1282, 1333]
[175, 247, 546, 601, 679, 1009, 1027, 1100, 1120, 1134]
```

```
group members:  [278, 373, 396, 584, 709, 979, 1153, 1171, 1181, 1224]
recommendation list:  [2732 3471  206 3696  287 3378 4125 1371 2531 1769 2189 1669 3339 2423
 1694 3870 1540 3714 2901 1724  451 3610 3168 1927   48]
```

System performance

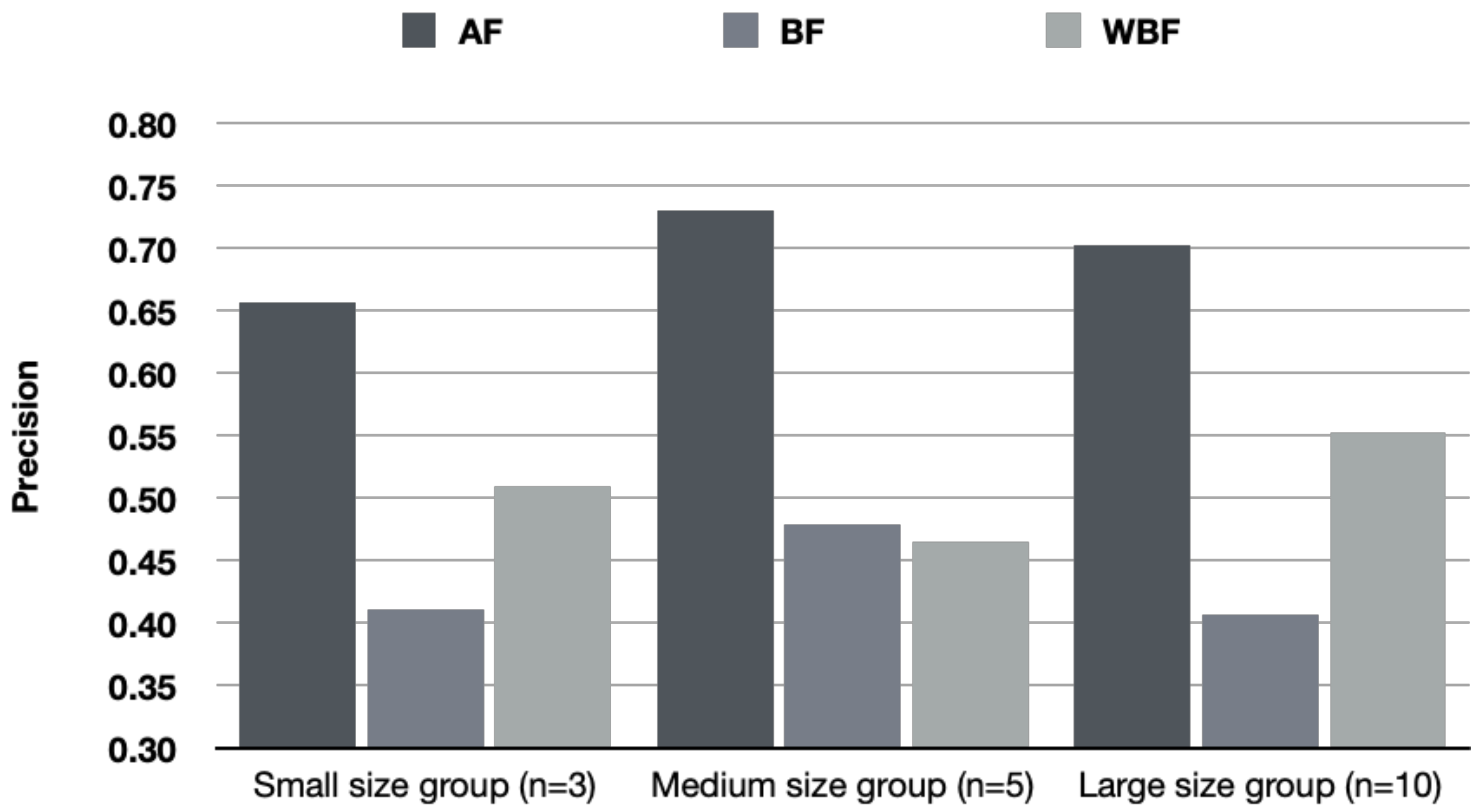
* Averaged statistics from 20 times experiment

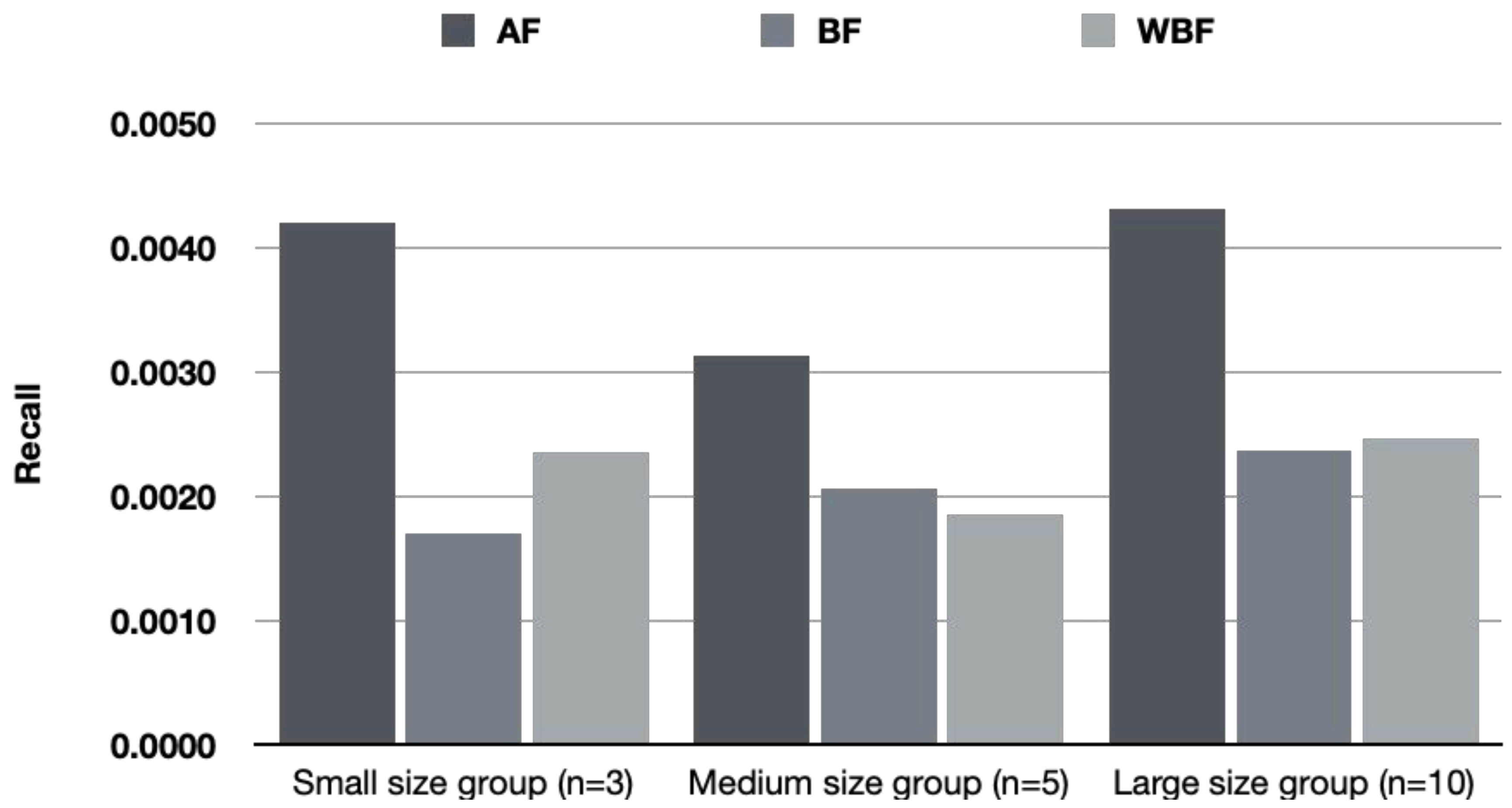
Precision The percentage of songs recommended to a group that would be liked by every group member.

$$precision_G = \frac{\#TP_G}{\#(TP_G \bigcup FP_G)}$$

Recall The ratio of recommended and liked songs to all songs that meet the same criteria.

$$recall_G = \frac{\#TP_G}{\#T_G}$$





Explaining recommendations

Explanation writing 1

“Top influencers for the playlist”

Sorting the similarities between the group vector and user vectors

```
group members:  [278, 373, 396, 584, 709, 979, 1153, 1171, 1181, 1224]
recommendation list:  [2732 3471  206 3696  287 3378 4125 1371 2531 1769 2189 1669 3339 2423
 1694 3870 1540 3714 2901 1724  451 3610 3168 1927   48]
{709: '89%',
 584: '88%',
 278: '84%',
 979: '83%',
 396: '82%',
 373: '80%',
 1224: '30%',
 1153: '30%',
 1171: '28%',
 1181: '28%'}
```

**User Index
(can be linked to user id)**

Similarity between user vector and group vector

Explaining recommendations continues…

Explanation writing 2

“Likely enjoyed by…”

comparing the predicted rating
between the vectors of every user
and every recommended music

```
music_top_enjoyed_user

{2732: [278, 396, 709, 373, 1224, 584, 979, 1181, 1171, 1153],
 3471: [1171, 1181, 278, 396, 1153, 709, 584, 373, 979, 1224],
 206: [278, 396, 709, 1224, 1181, 1153, 1171, 373, 584, 979],
 3696: [1171, 278, 396, 709, 979, 373, 584],
 287: [278, 1181, 396, 709, 1171, 373, 584, 1224, 1153, 979],
 3378: [278, 396, 709, 584, 1171, 979, 373],
 4125: [1181, 1153, 278, 373, 709, 1171, 396, 584, 1224, 979],
 1371: [278, 396, 1171, 709, 1153, 584, 373, 979, 1224, 1181],
 2531: [278, 1181, 396, 1171, 709, 373, 584, 1224, 1153, 979],
 1769: [1181, 278, 396, 709, 1171, 1224, 373, 584, 979, 1153],
 2189: [1153, 1181, 278, 1224, 396, 709, 373, 584, 979, 1171],
 1669: [1181, 1153, 278, 709, 396, 979, 373, 584, 1224],
 3339: [1224, 278, 396, 709, 373, 979, 584, 1153],
 2423: [1171, 278, 1181, 709, 396, 1224, 584, 373, 979],
 1694: [1181, 1153, 1224, 373, 278, 584, 709, 396, 979],
 3870: [1171, 1224, 278, 396, 709, 373, 584, 1181, 979],
 1540: [1171, 1181, 278, 396, 709, 584, 373, 979, 1153, 1224],
 3714: [278, 396, 979, 709, 373, 584],
 2901: [1181, 278, 709, 396, 373, 979, 584],
 1724: [1181, 278, 1153, 396, 709, 1224, 373, 584, 979, 1171],
 451: [278, 396, 1181, 1171, 709, 373, 584, 979, 1224],
 3610: [1181, 1224, 1153, 278, 709, 396, 979, 584, 373],
 3168: [278, 396, 1153, 1181, 709, 979, 373, 584, 1224],
 1927: [278, 1171, 396, 709, 1224, 373, 584, 1181, 979, 1153],
 48: [1224, 278, 396, 709, 1153, 979, 373, 584]}
```

Music Index **User Index**

Discussion

Converting play count data to rating data work is practicable for recommendation algorithm using Matrix Factorisation based Collaborative Filtering.

The precision rates using AF method is significantly higher in all three group sizes, with the best performance (precision = 0.729881) for medium size groups (n=5).

Explanations can be generated after the recommendation is performed. “Top influencers for the playlist” informs the representation of every group member in the playlist, “Likely enjoyed by…” tells why a song is recommended.

Discussion continues..

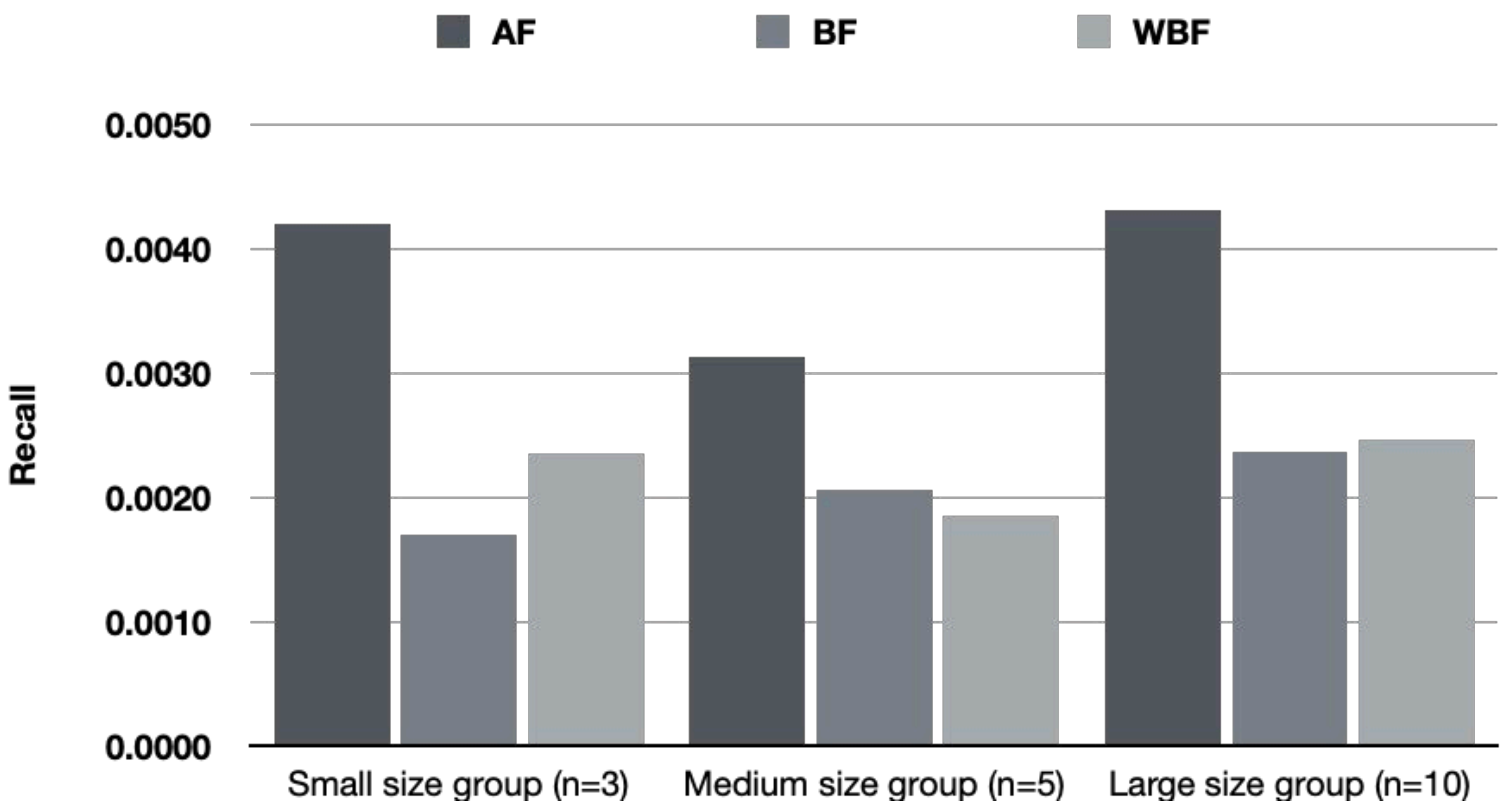
User trust and clarity

Factors and bias of group profile and group members, an example from a medium size group using AF method

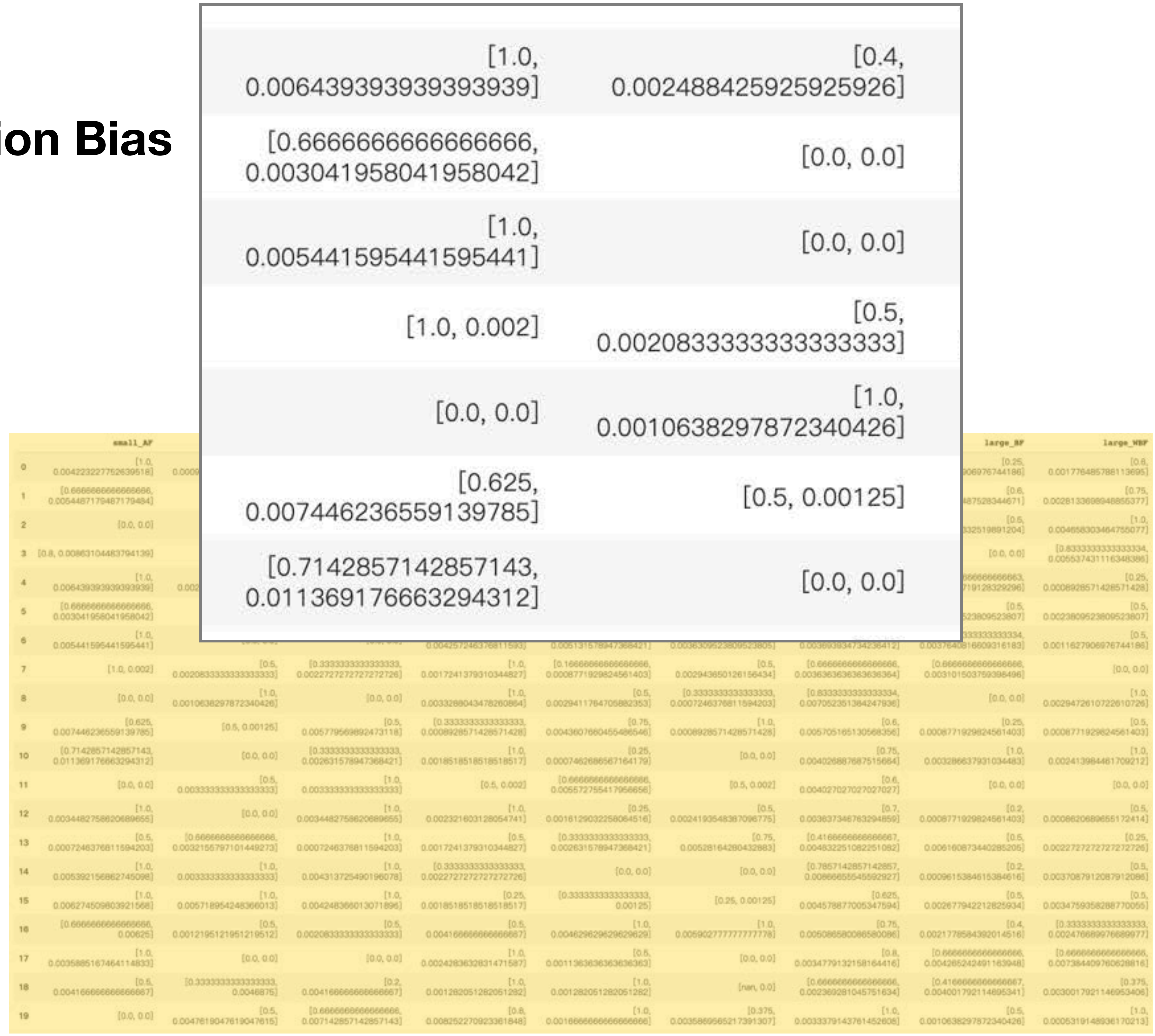
User Index	Similarity	F_1	F_2	F_3	F_4	F_5	b_u
(Group profile)	-	0.02101781	0.04496065	-0.07088136	0.00342788	-0.02857114	-0.04823289
734	93%	0.07451212	0.08319631	-0.05291501	-0.00177048	-0.02293285	-0.16143093
808	92%	-0.01362764	0.01744778	-0.11239957	0.00766848	-0.07983841	-0.13342482
512	92%	-0.02173481	0.01814226	-0.04460008	0.00639390	0.03168001	0.25780084
1134	48%	-0.64848781	0.18999331	0.06307645	0.73891721	-0.28622930	0.00000000
1270	45%	0.09736188	0.26146698	0.41860049	0.70812905	0.73932672	0.00000000

Discussion continues..

Low Recall



Selection Bias



[1.0, 0.006439393939393939]	[0.4, 0.002488425925925926]
[0.6666666666666666, 0.003041958041958042]	[0.0, 0.0]
[1.0, 0.005441595441595441]	[0.0, 0.0]
[1.0, 0.002]	[0.5, 0.0020833333333333333]
[0.0, 0.0]	[1.0, 0.0010638297872340426]
[0.625, 0.007446236559139785]	[0.5, 0.00125]
[0.7142857142857143, 0.011369176663294312]	[0.0, 0.0]

Conclusions

01 Utilising implicit user feedback like play counts by converting to ratings is practical given the limits of obtaining explicit user feedback real world.

02 The after factorisation method (AF) is proven to perform the best for all group sizes between three to ten people.

03 We can generate explanation to inform the reason for every suggested music as well as how much every group member is represented in the playlist.

Future work

Modifying the evaluation method

Applying a larger dataset

Optimising the sampling process.

User test that draw on how the explanation affect the mental model in users

It is acknowledged that explainability is essential between Human-AI interaction for the purpose of enhancing people's understanding of the system and building trust (Liao et al, 2021).

Thank you