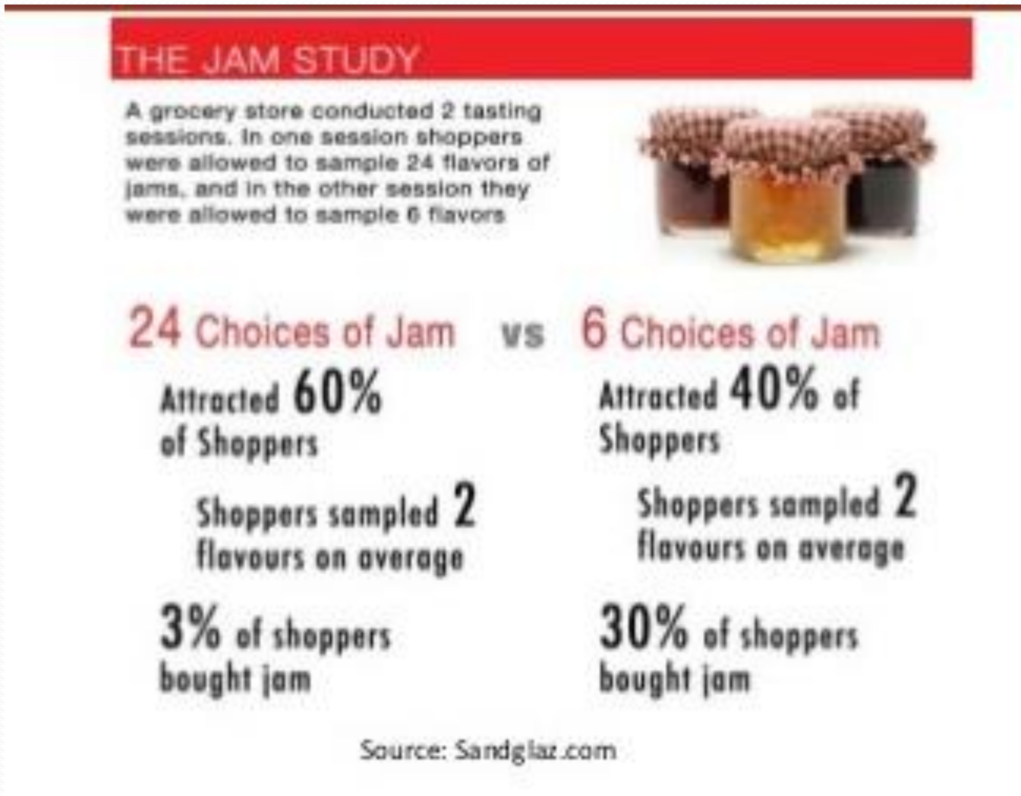
Recommendations…

- We are leaving the age of Information and entering the age of Recommendation.
- The world move from "one size fits all" solutions to personal tailor made solutions.

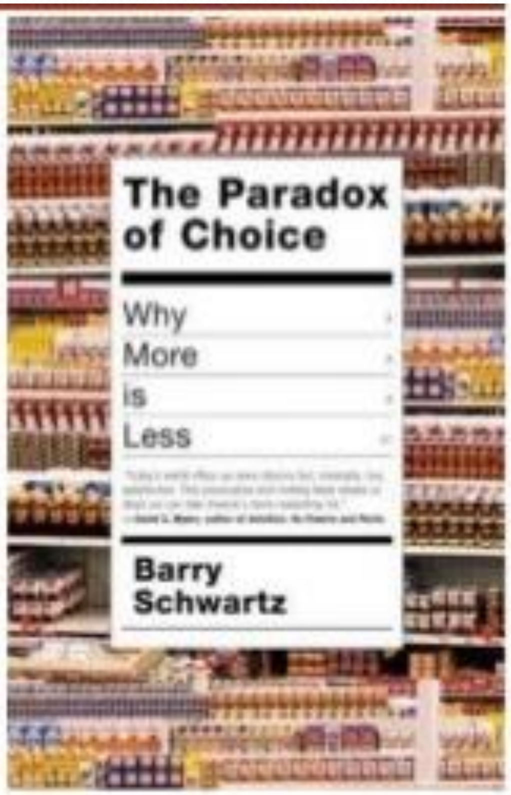


THE JAM STUDY

A grocery store conducted 2 tasting sessions. In one session shoppers were allowed to sample 24 flavors of jams, and in the other session they were allowed to sample 6 flavors

24 Choices of Jam	vs	6 Choices of Jam
Attracted 60% of Shoppers		Attracted 40% of Shoppers
Shoppers sampled 2 flavours on average		Shoppers sampled 2 flavours on average
3% of shoppers bought jam		30% of shoppers bought jam

Source: Sandglaz.com



The Paradox of Choice

Why More is Less

Barry Schwartz

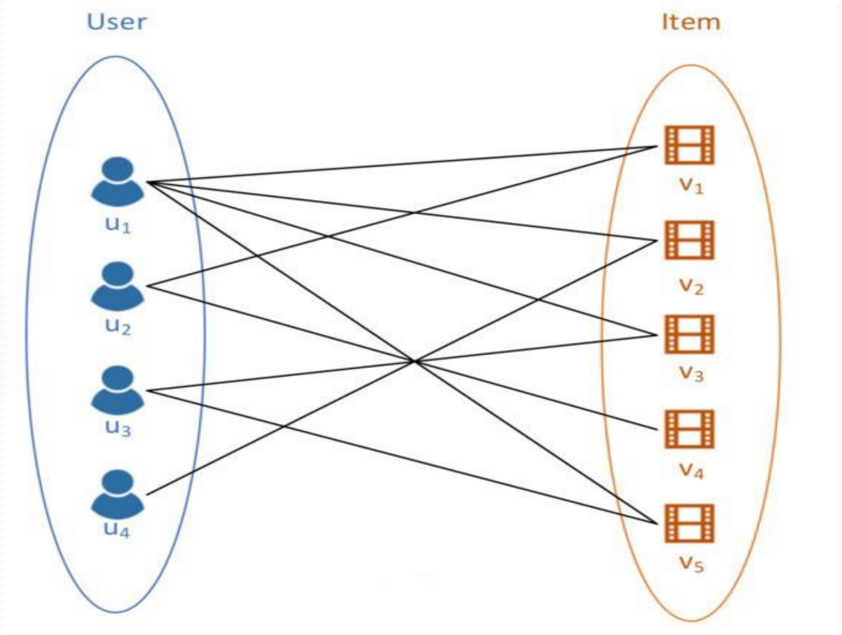
Definition

- Recommender systems can be defined as programs which attempt to recommend the most suitable items to particular users by predicting a user’s interest in an item [1].
- The most important feature of a Recommender System is its ability to “guess” a user’s preferences and interests by analysing the behaviour of the user to generate personalized recommendations [2].
- Example of Recommendation System:

facebook	“People you may know”
NETFLIX	“Other movies you may like”
LinkedIn	“Jobs you may be interested in”
amazon	“Customer who bought this item also bought this”

Recommendation System as Graph

- A Recommendation System can be represented as a **Bipartite Graph** with the notation **G = (V , E),** where G is the whole network, V stands for the set of vertices which comprises two distinct set of vertices users and items, i.e. V= {U,I} and E is the set of all edges between users and items.



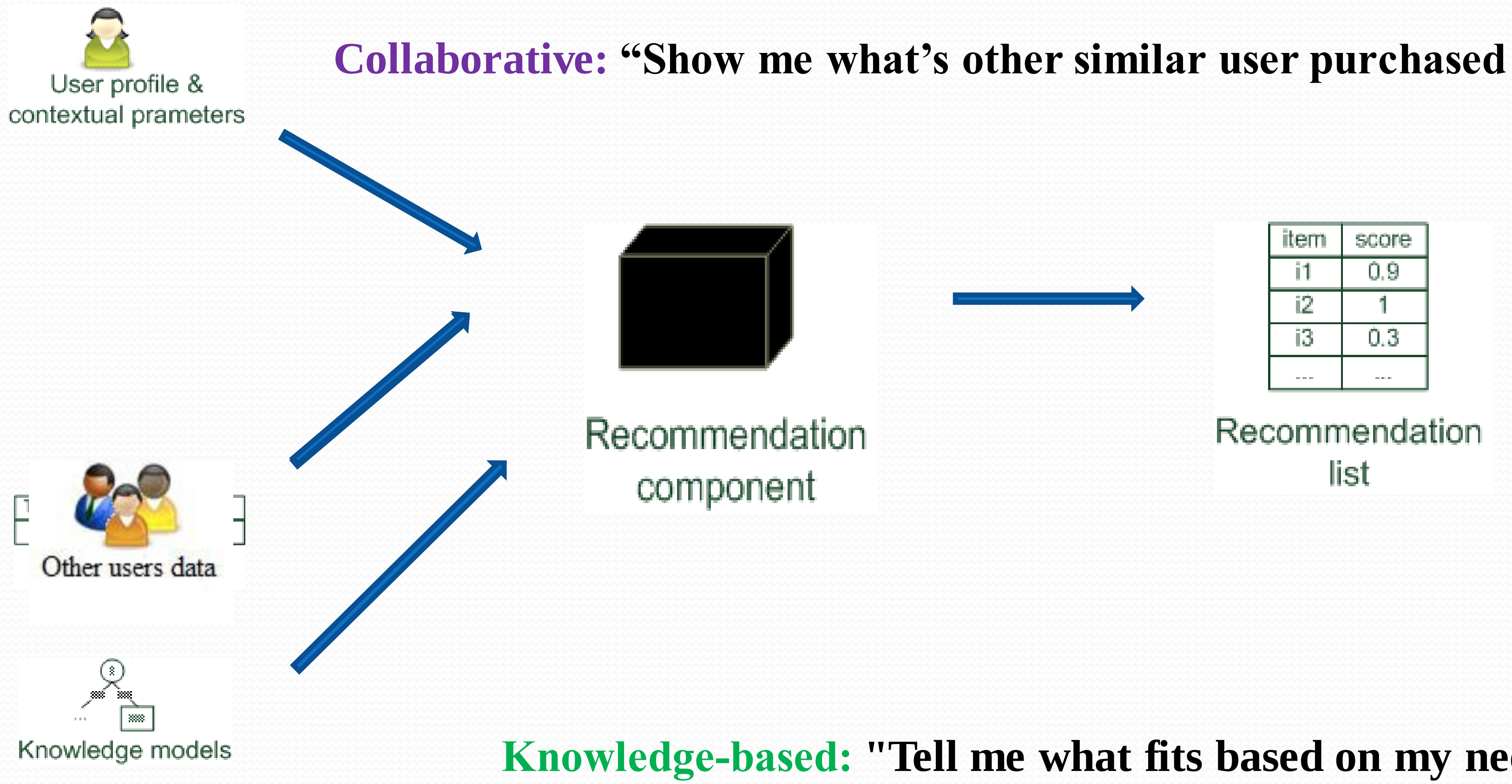
Here the edge (u1,v2) means the user u1 has liked or purchased the item v2.

Paradigms of Recommender System

Recommender systems reduce information overload by estimating relevance.

Content-based: **"Show me more of the same what I've liked"**

Collaborative: **“Show me what’s other similar user purchased before”**



item	score
i1	0.9
i2	1
i3	0.3
...	...

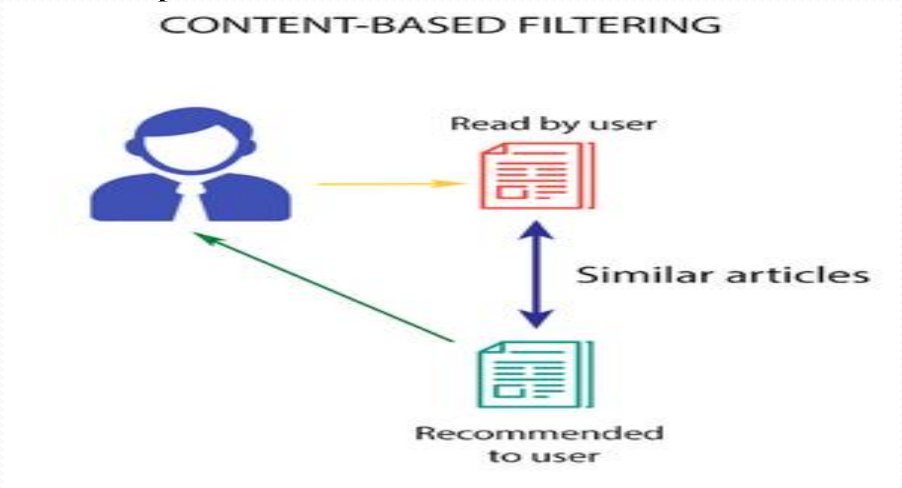
Knowledge-based: **"Tell me what fits based on my needs"**

Different types of Recommendation Techniques

- The purpose of a recommender system is to suggest relevant items to users.
- To achieve this task, traditional recommendation techniques can be primarily classified into two categories:

1. Content based Filtering:

- ➢ Here recommendation is made based on product description.
- ➢ This algorithm recommends products which are similar to the ones that a user has liked in the past [1].



Content based filtering

- It saves all the information related to each user in a vector form, called ***profile vector***.
- Suppose, a user watched a movie that falls under the category: Action, Thriller and Sci-Fic.

  So, the profile vector of the user is: (1,1,1) in order of (Action, Thriller, Sci-Fic).
- All the information related to item is also stored in another vector called the ***item vectors.***
- Suppose we have another two movies to recommend with the following category:

  Movie-1: (Action, Comedy) and Movie-2: (Thriller, Sci-Fic)

  So, item vector of Movie-1 is: (1,0,0) and Movie-2 is: (0,1,1) in order of (Action, Thriller, Sci-Fic).

Content based filtering

- The content-based filtering algorithm finds the similarity between the profile vector and item vector.
- Suppose A is the profile vector and B is the item vector, then the similarity between them can be calculated as:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

- A perfect similarity will have a score of 1 (or an angle of 0) and no similarity will have a score of 0 (or an angle of 90 degrees).

- Sim(user, Movie-1) = (1*1+ 1*0+1*0) / (√(1^2+1^2+1^2)*√(1^2+0^2+0^2))
  = 1/√3 = .577

  Sim(user, Movie-2) = (1*0+ 1*1+1*1) / (√(1^2+1^2+1^2)*√(0^2+1^2+1^2))
  = 2/√6 = .816

Hence, **Movie-2 will be recommended to the user**.

Other methods that can be used to calculate the similarity are:

- **Euclidean Distance**: Similar items will lie in close proximity to each other if plotted in n-dimensional space. So, we can calculate the distance between items and based on that distance, recommend items to the user. The formula for the Euclidean distance is given by:

$$\text{Euclidean Distance} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2}$$

- **Pearson's Correlation**: It tells us how much two items are correlated. Higher the correlation, more will be the similarity. Pearson's correlation can be calculated using the following formula:

$$sim(u, v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

Different types of Recommendation Techniques

2. Collaborative Filtering:

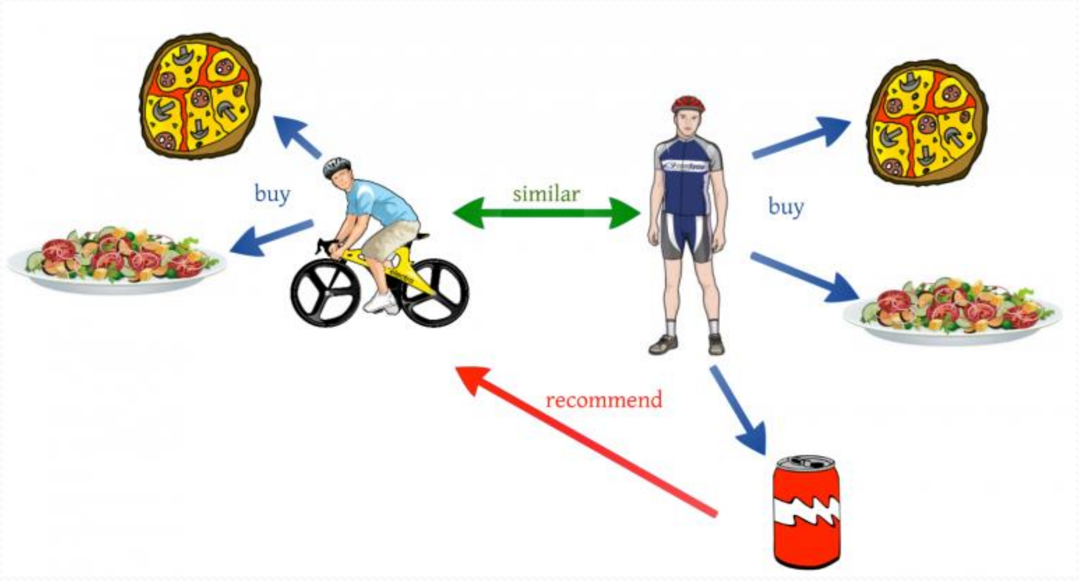
- Collaborative Filtering (CF) is a technique to predict a user's taste and find the items that a user might prefer on the basis of information collected from various other users having similar tastes or preferences[2].
- Collaborative Filtering can be of two types:

1. **Memory based methods:** Here ratings of user-item combinations are predicted on the basis of their neighbourhoods. These neighbourhoods can be defined in one of two ways:
   - User-User Collaborative Filtering
   - Item-Item Collaborative Filtering

2. **Model based methods:** Here machine learning is used to find user ratings of unrated items.

User-User Collaborative filtering

- **Users having higher correlation will tend to be similar.** Based on this, recommendations are made.
- Here products are recommended based on similarity between two users on the basis of products that they have liked or bought previously.



User-User Collaborative filtering

- Consider the user-movie rating matrix:

User/ Movie	M1	M2	M3	M4	M5	Mean Rating
A	4	1	-	4	-	3
B	-	4	-	2	3	3
C	-	1	-	4	4	3

Find the similarity between users: (A, C) and (B, C)?

Common movies rated by A and C are: M2 and M4

B and C are : M2, M4 and M5

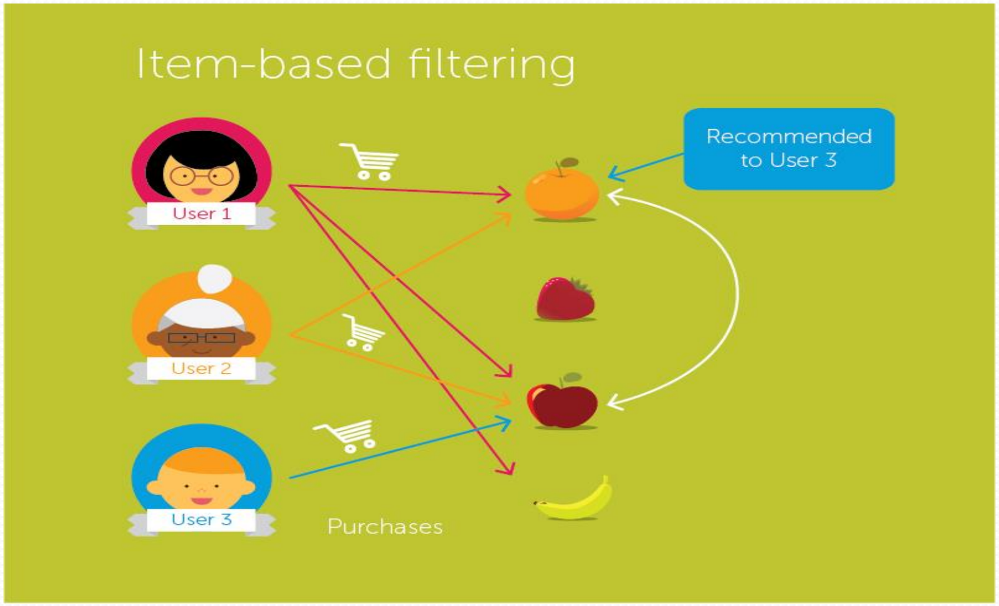
$$r_{AC} = [(1-3)*(1-3) + (4-3)*(4-3)]/[((1-3)^2 + (4-3)^2)^{1/2} * ((1-3)^2 + (4-3)^2)^{1/2}] = 1$$

$$r_{BC} = [(4-3)*(1-3) + (2-3)*(4-3) + (3-3)*(4-3)]/[((4-3)^2 + (2-3)^2 + (3-3)^2)^{1/2} * ((1-3)^2+(4-3)^2 + (4-3)^2)^{1/2}] = -0.866$$

- **The correlation between user A and C is more than the correlation between B and C.**
- Hence users A and C have more similarity and the movies liked by user A will be recommended to user C and vice versa.

Item-Item collaborative filtering

- This algorithm finds the similarity between each item pair and based on that, it will recommend similar items which are liked by the users in the past.
- In this algorithm, instead of taking the weighted sum of ratings of "user-neighbors", we take the weighted sum of ratings of "item-neighbors".



Item-Item Collaborative Filtering

- **Consider the user-movie rating matrix:**

User/Movie	M1	M2	M3	M4	M5
A	4	1	2	4	4
B	2	4	4	2	1
C	-	1	-	3	4
Mean item rating	3	2	3	3	3

- Let us find the similarity between movies (M1, M4) and (M1, M5). Common users who have rated movies M1 and M4 are A and B while the users who have rated movies M1 and M5 are also A and B.

$$C_{14} = [(4-3)*(4-3) + (2-3)*(2-3)] / [((4-3)^2 + (2-3)^2)^{1/2} * ((4-3)^2 + (2-3)^2)^{1/2}] = 1$$

$$C_{15} = [(4-3)*(4-3) + (2-3)*(1-3)] / [((4-3)^2 + (2-3)^2)^{1/2} * ((4-3)^2 + (1-3)^2)^{1/2}] = 0.94$$

- The similarity between movie M1 and M4 is more than the similarity between movie M1 and M5. So based on these similarity values, if any user searches for movie M1, they will be recommended movie M4 and vice versa.

Problems in Traditional Recommendation Techniques

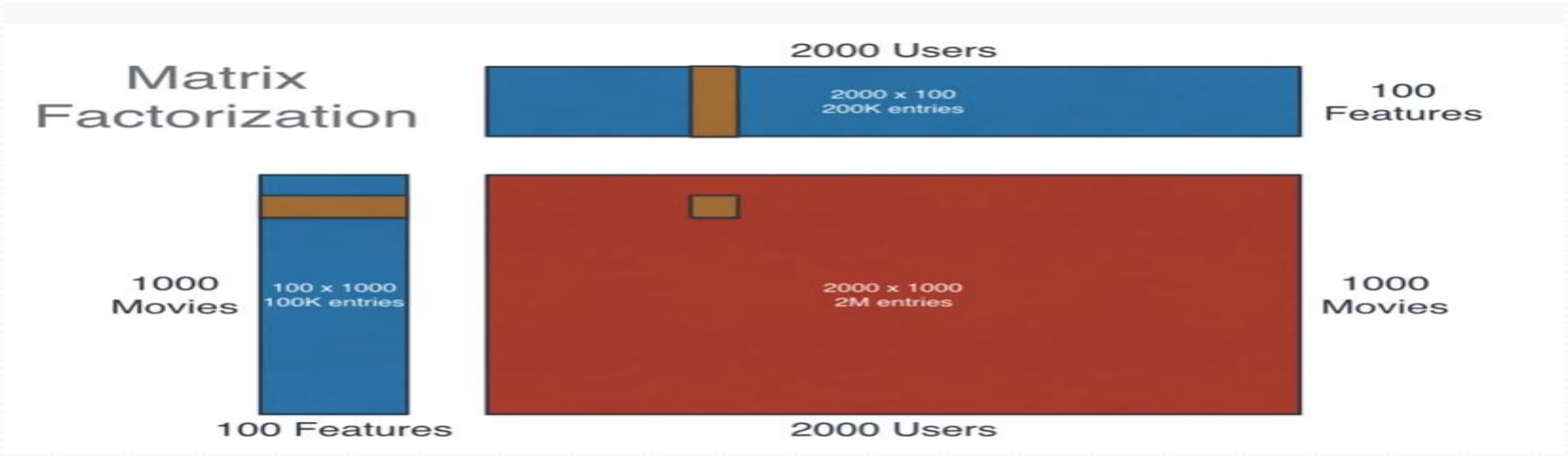
- The user/item rating matrix is typically **very sparse**.

- Another related problem is the **cold-start problem**.

- Content based technique does not consider customers’ attitude towards the product, which limits the accuracy of recommendation.

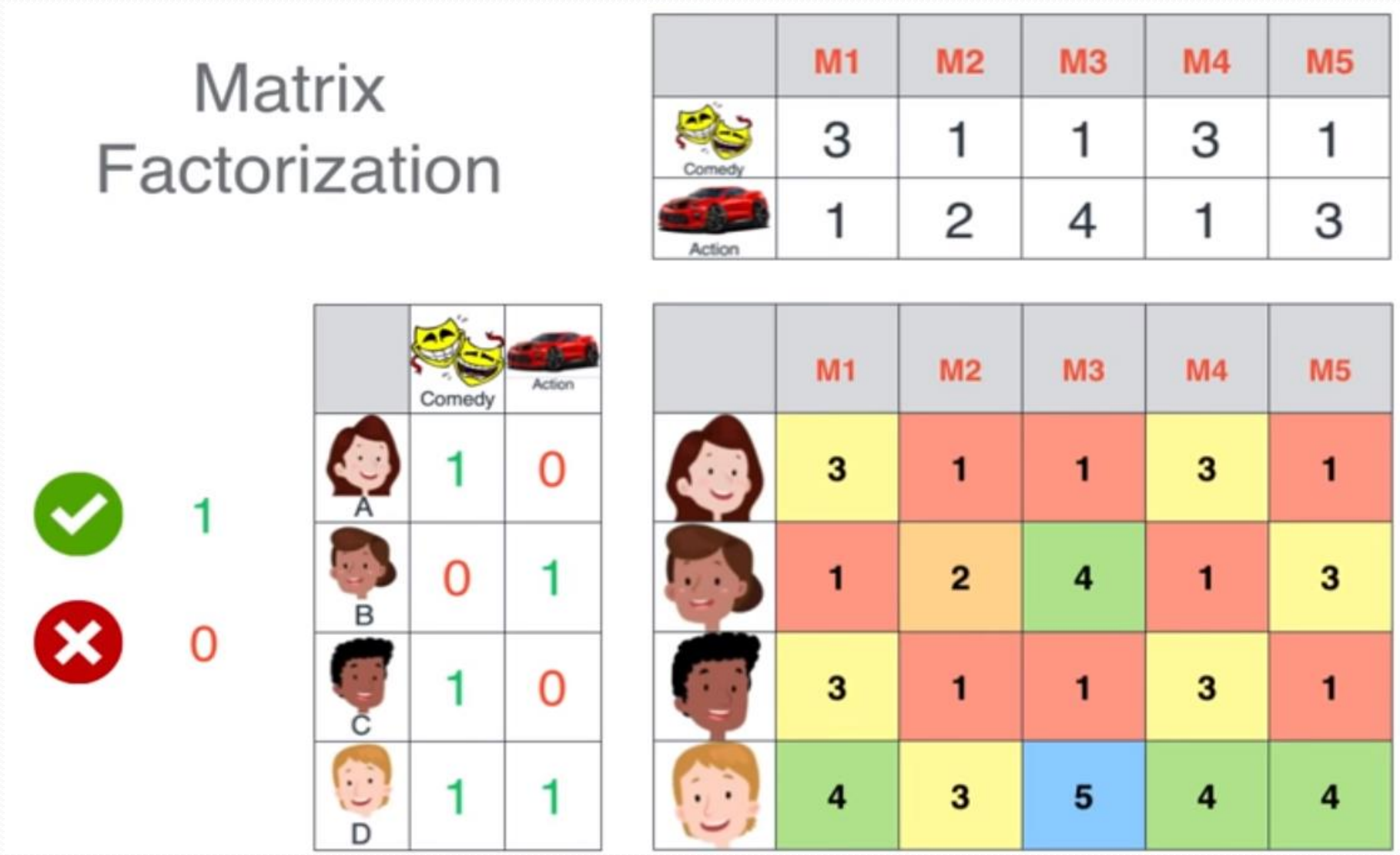
- On the other hand, Collaborative Filtering technique dose not consider the product attributes which are pivotal for a recommender system.

Matrix Factorization for Recommendation



- MF is the technique through which we can found latent features.
- Let, size of the original user-movie matrix (R) is (m,n)
- It reduce the dimensionality of the original matrix by decomposing it into two matrices user (P) and movie (U) having dimensions of (m,k) and (k,n)
- Mathematically: $R_{u,i} = P_u . Q_i^T$
- A ratting $R_{u,i}$ can be estimated by dot product of user P_u and item Q_i

Matrix Factorization for Recommendation



Matrix Factorization

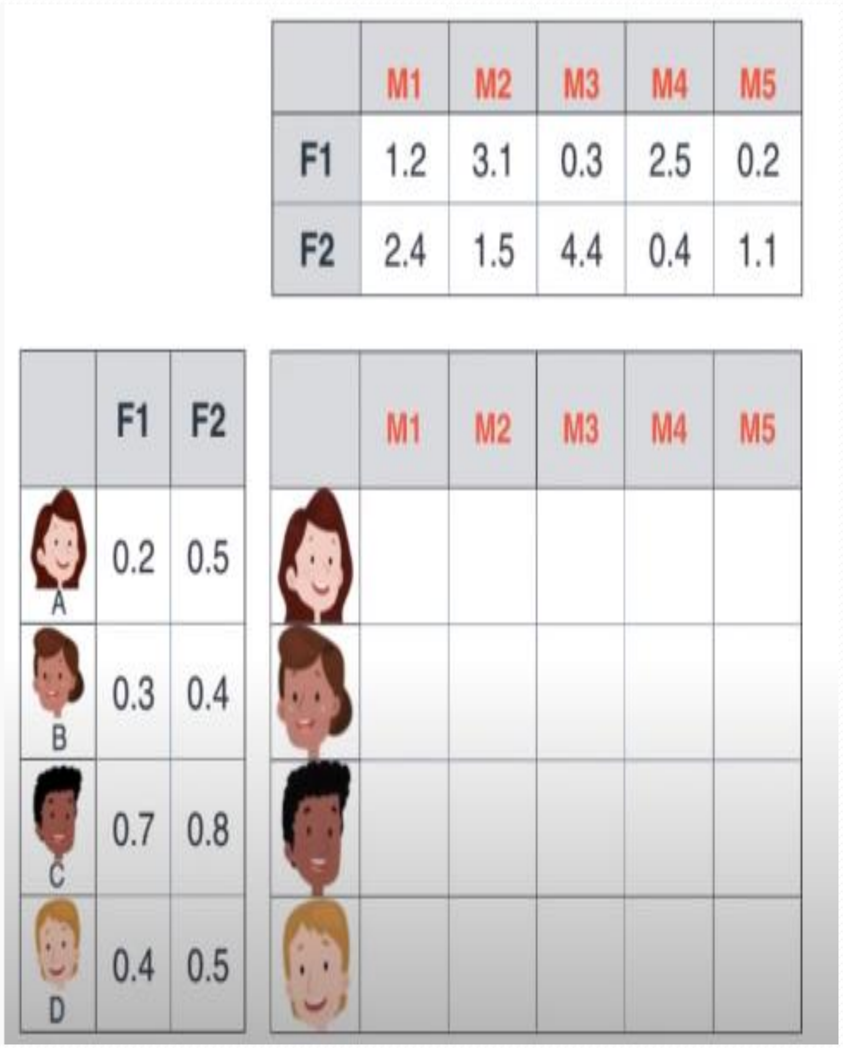
	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

✓	1
✗	0

	Comedy	Action
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
A	3	1	1	3	1
B	1	2	4	1	3
C	3	1	1	3	1
D	4	3	5	4	4

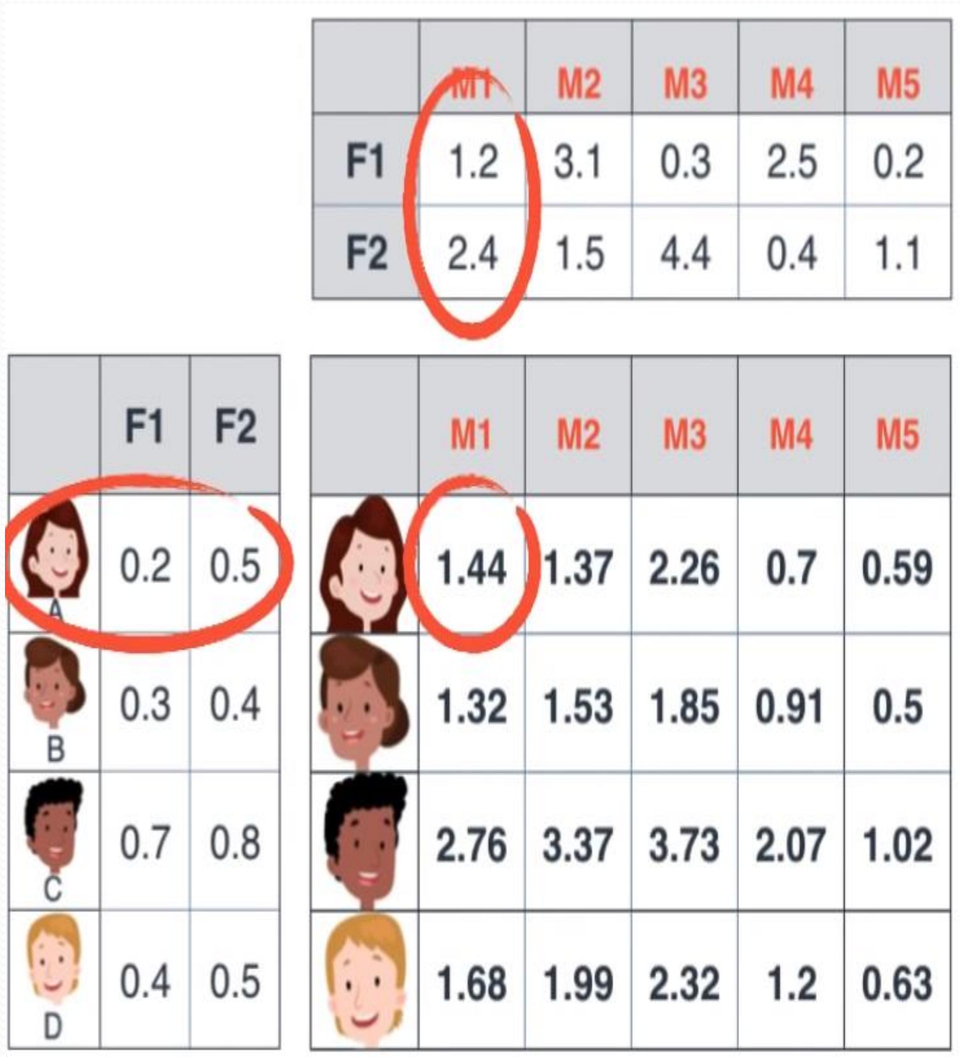
Matrix Factorization for Recommendation



	M1	M2	M3	M4	M5
F1	1.2	3.1	0.3	2.5	0.2
F2	2.4	1.5	4.4	0.4	1.1

	F1	F2
A	0.2	0.5
B	0.3	0.4
C	0.7	0.8
D	0.4	0.5

	M1	M2	M3	M4	M5

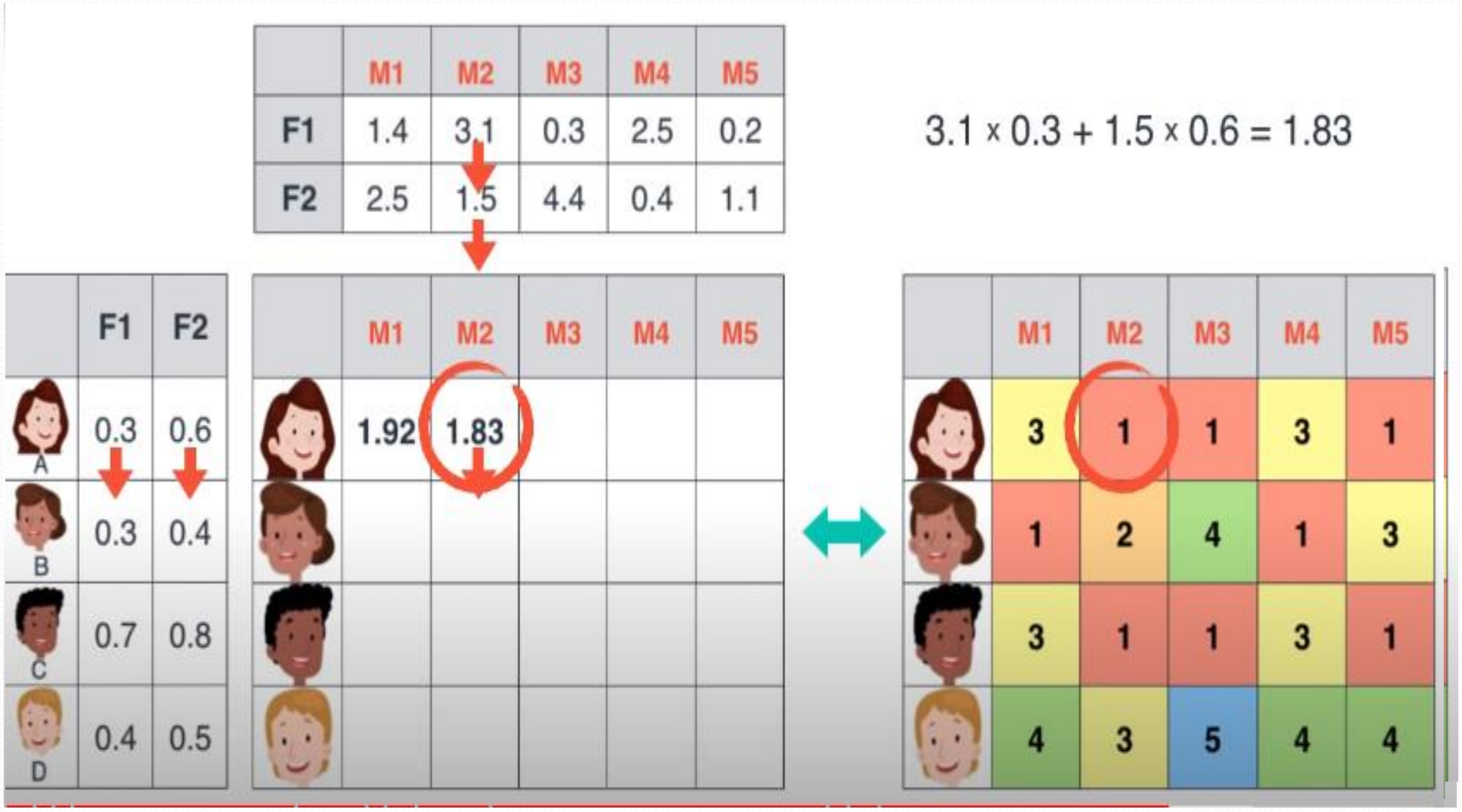


	M1	M2	M3	M4	M5
F1	1.2	3.1	0.3	2.5	0.2
F2	2.4	1.5	4.4	0.4	1.1

	F1	F2
A	0.2	0.5
B	0.3	0.4
C	0.7	0.8
D	0.4	0.5

	M1	M2	M3	M4	M5
	1.44	1.37	2.26	0.7	0.59
	1.32	1.53	1.85	0.91	0.5
	2.76	3.37	3.73	2.07	1.02
	1.68	1.99	2.32	1.2	0.63

Gradient Descent in Matrix Factorization



	M1	M2	M3	M4	M5
F1	1.4	3.1	0.3	2.5	0.2
F2	2.5	1.5	4.4	0.4	1.1

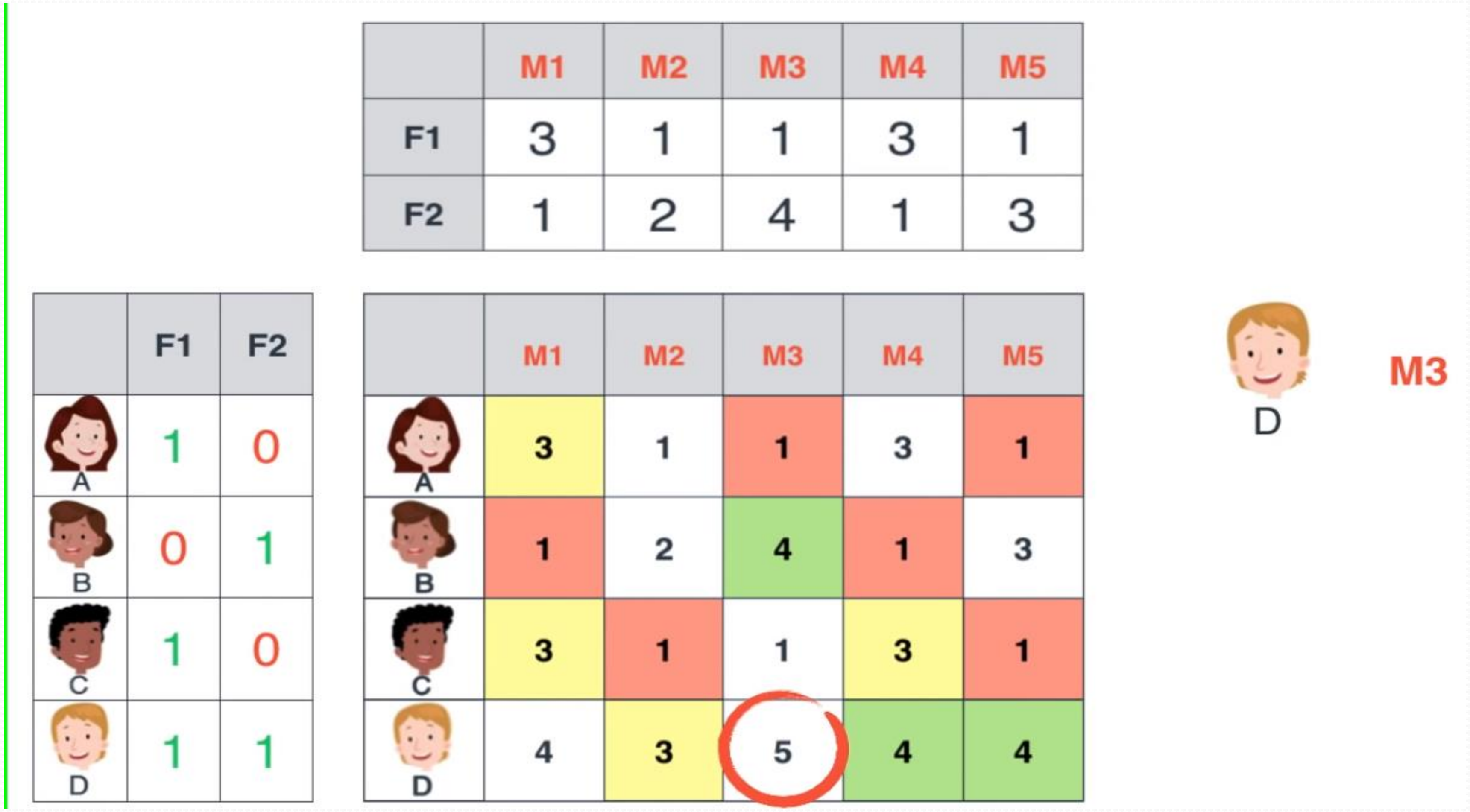
$$3.1 \times 0.3 + 1.5 \times 0.6 = 1.83$$

	F1	F2
A	0.3	0.6
B	0.3	0.4
C	0.7	0.8
D	0.4	0.5

	M1	M2	M3	M4	M5
	1.92	1.83			

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

How Recommendation Works?



	M1	M2	M3	M4	M5
F1	3	1	1	3	1
F2	1	2	4	1	3

	F1	F2
A	1	0
B	0	1
C	1	0
D	1	1

	M1	M2	M3	M4	M5
A	3	1	1	3	1
B	1	2	4	1	3
C	3	1	1	3	1
D	4	3	5	4	4