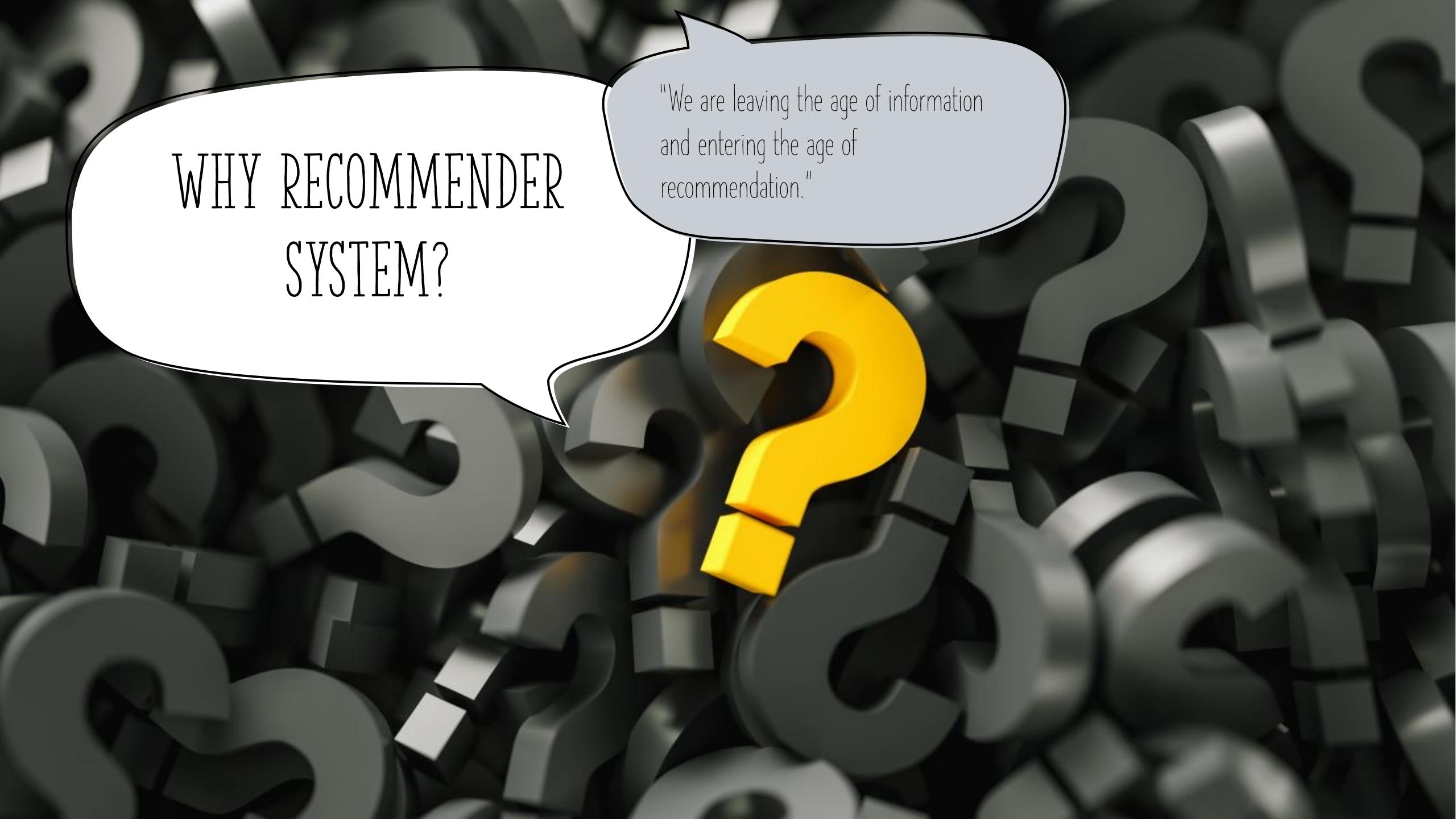
MUSIC RECOMMENDER SYSTEM

Arvind Suriakanth, Chieh Wu,
Pragyna Veeramallu, Sai Kumar,
Sina Saba

FANTASTIC FIVE

Group 5


WHY RECOMMENDER SYSTEM?
"We are leaving the age of information and entering the age of recommendation."

RECOMMENDER SYSTEM

Music dataset is too big while life is too short! You need someone to teach you how to manage and give you wise suggestions according to your taste.

Recommender systems are a way of suggesting like or similar items and ideas to a user specific way of thinking.

Recommender systems try to automate aspects of a completely different information discovery model where people try to find other people with similar tastes and then ask them to suggest new things.

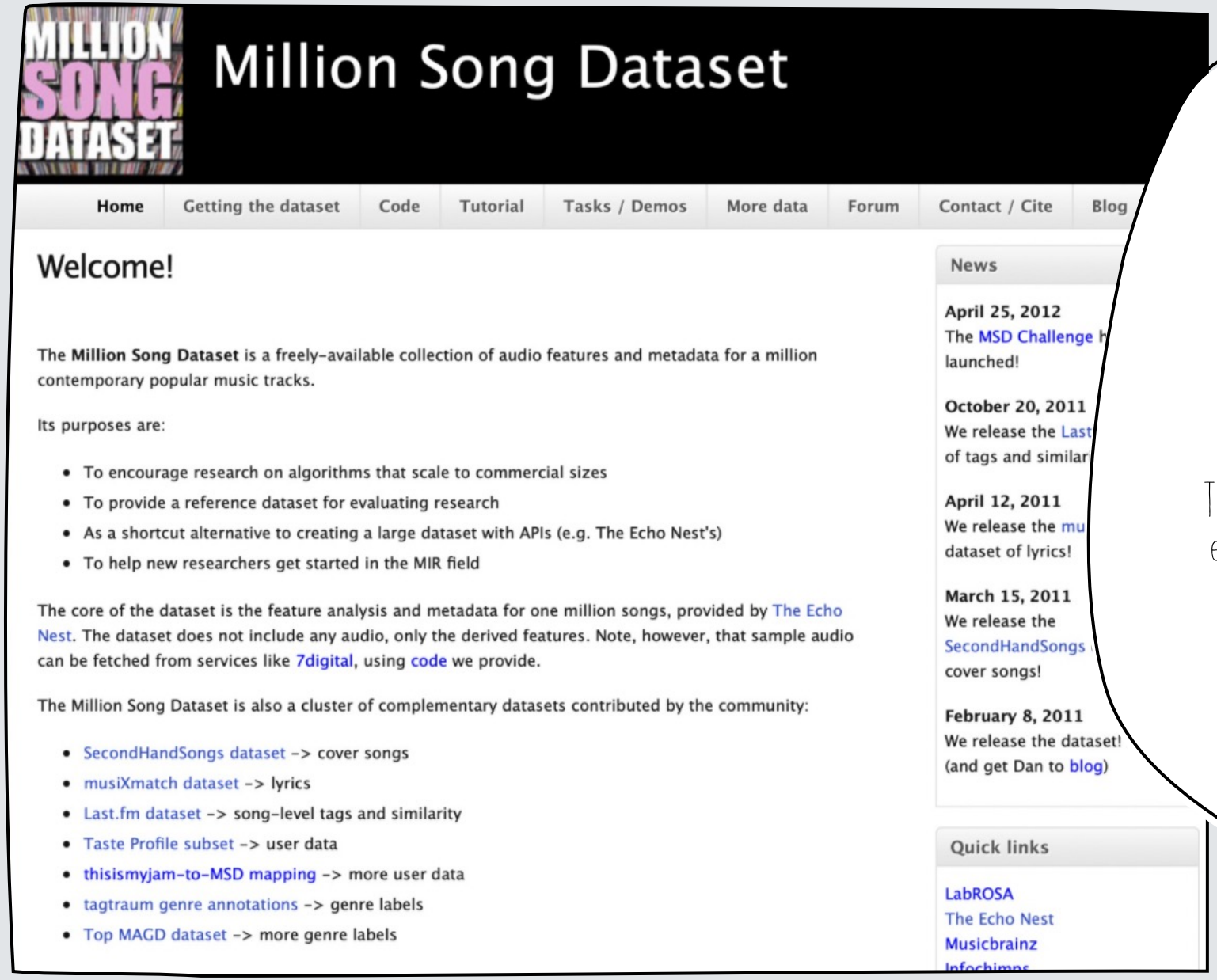
Search:

User → Items

Recommend:

Items → User


DATASET



We used the database provided by Kaggle competition, refer

https://www.kaggle.com/c/msdchallenge/data

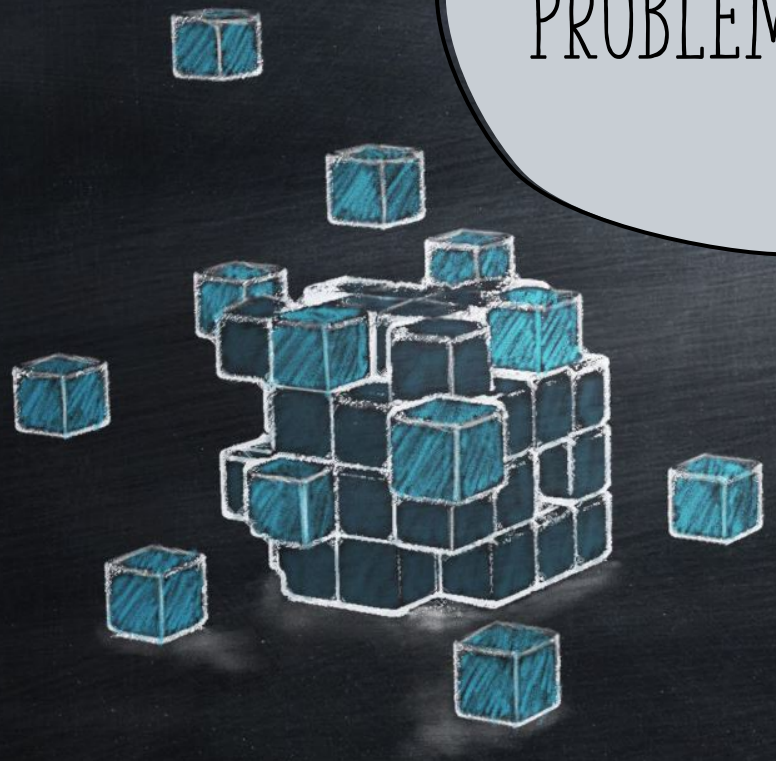
It includes metadata (e.g., artist identifiers, tags, etc) , audio content analysis and standardized identifiers.

The data is open; meta-data, audio – content analysis, etc. are available for all the songs. It is also very large and contains around 48 million (userid, songid, play count) triplets collected from histories of over one million users and metadata of millions of songs.

OUR TASK


User's listening history &music information
Music Recommender System
Prediction of songs that user will listen to

PROBLEMS AND CHALLENGES



Problems:

- Inconclusive user feedback forms
- Finding users to take the feedback surveys

Challenges:

- Cold start problem
- Automatic playlist continuation
- Non user - specific parameterization
- Dimensionality for SVD

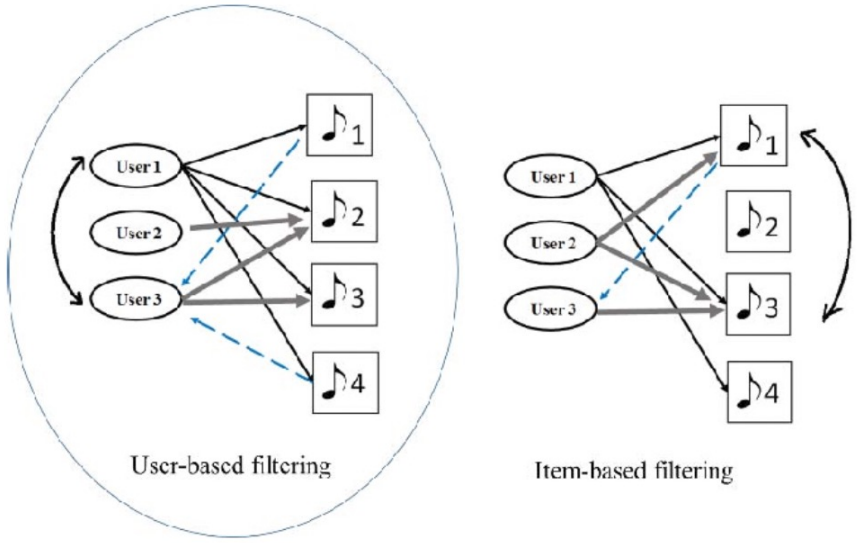

MOTIVATION

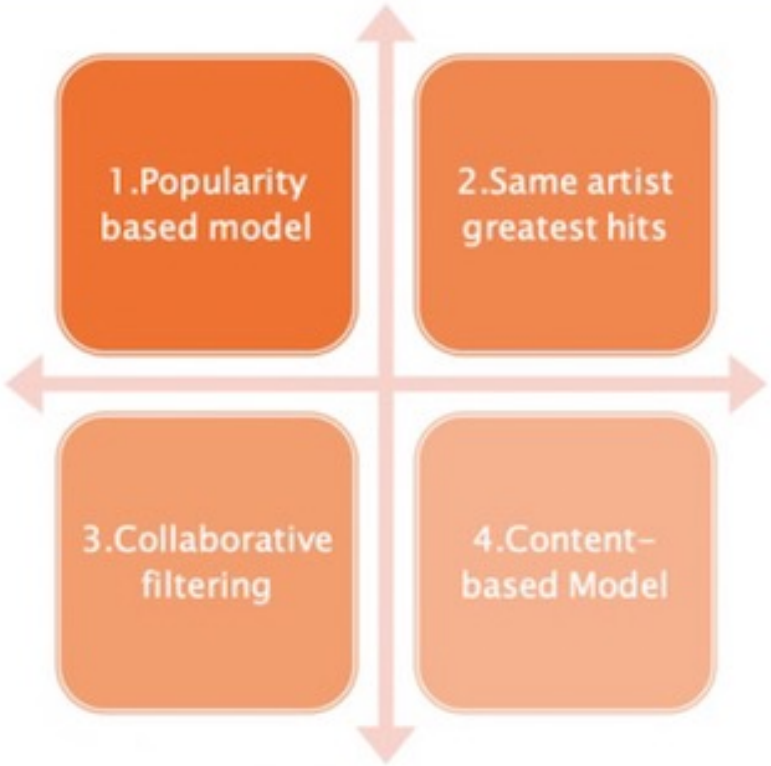
People today have unparalleled access to music collections because to the growth of digital content delivery. Commercial music archives easily surpass 15 million tracks, far beyond any single person's listening capacity. People might become overwhelmed when faced with millions of tunes to pick from. As a result, both music service providers and users benefit from an effective music recommender system. Customers will no longer have to struggle to decide what to listen to, and music firms will be able to sustain and attract new users by boosting user happiness.

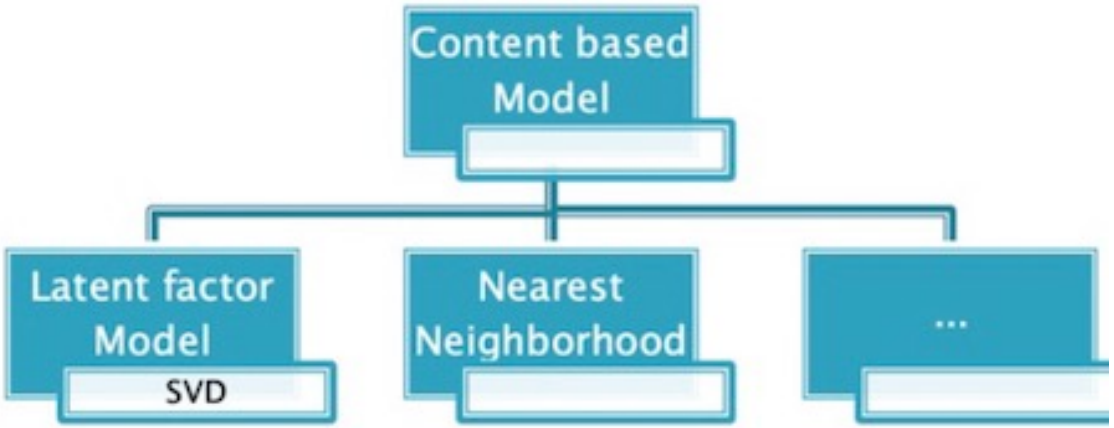
EXISTING RELATED APPROACHES

We see that Yading Song and Marcus Pearce explained fundamental metadata-based model and used two popular music recommender approaches: collaborative filtering and content-based model in their article "A Survey of Music Recommendation Systems and Future Perspectives." They used Collaborative filtering to recommend the items via choices of other similar users.

Though they have had great success, they have clear flaws such as popularity bias and human effort. Furthermore, a hybrid model would outperform a single model since it combines the benefits of both methodologies. Their effort, however, fell short of providing user-centric music choices.



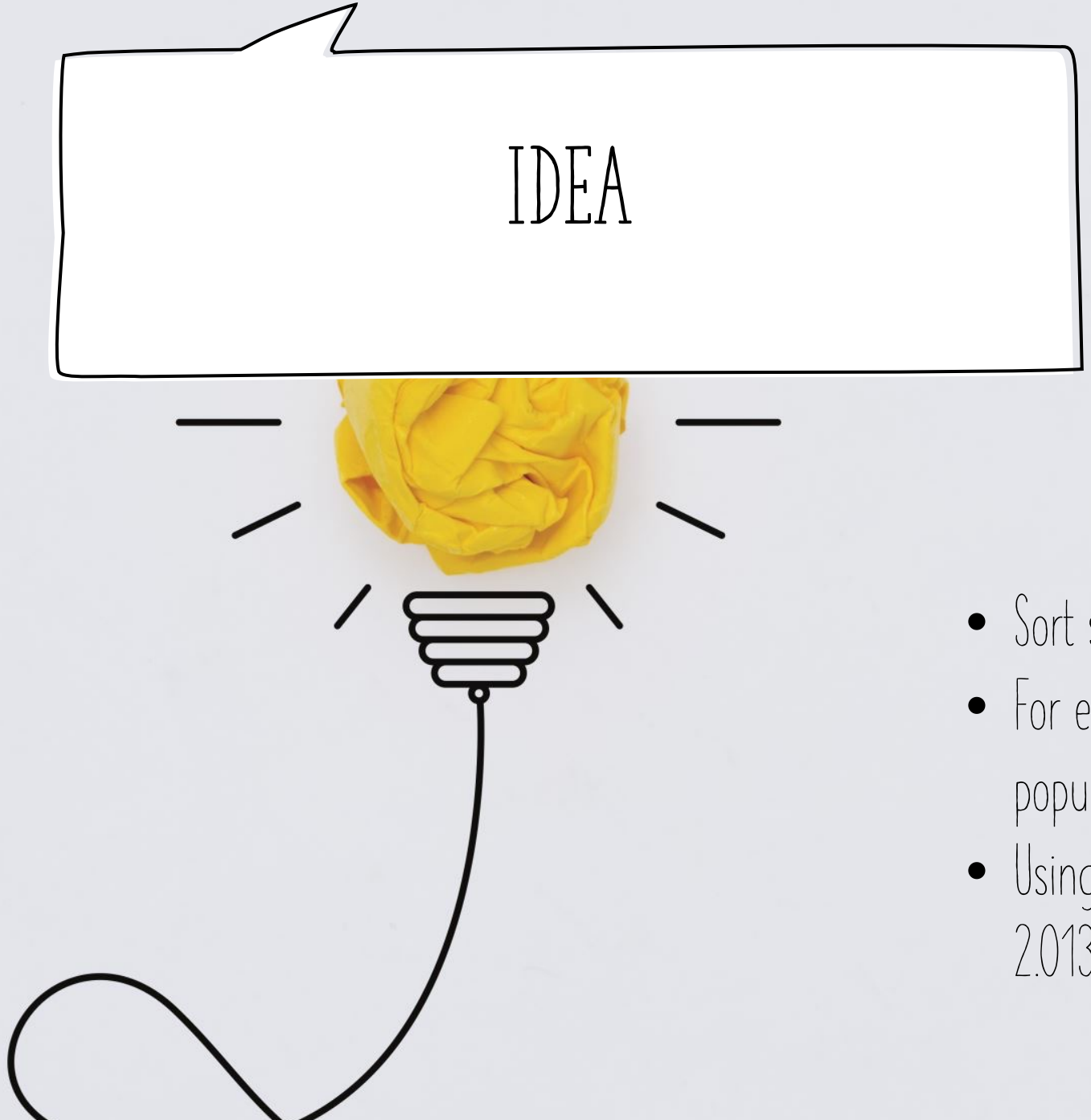

1.Popularity based model
2.Same artist greatest hits
3.Collaborative filtering
4.Content-based Model


Content based Model
Latent factor Model
SVD
Nearest Neighborhood
...

OUR METHOD

1. POPULARITY BASED MODEL

IDEA



- Sort songs by popularity in a decreasing order.
- For each user, recommend the songs in order of popularity, except those already in the user's profile
- Using popularity-based model, we got an accuracy of 2.0138

PROS & CONS

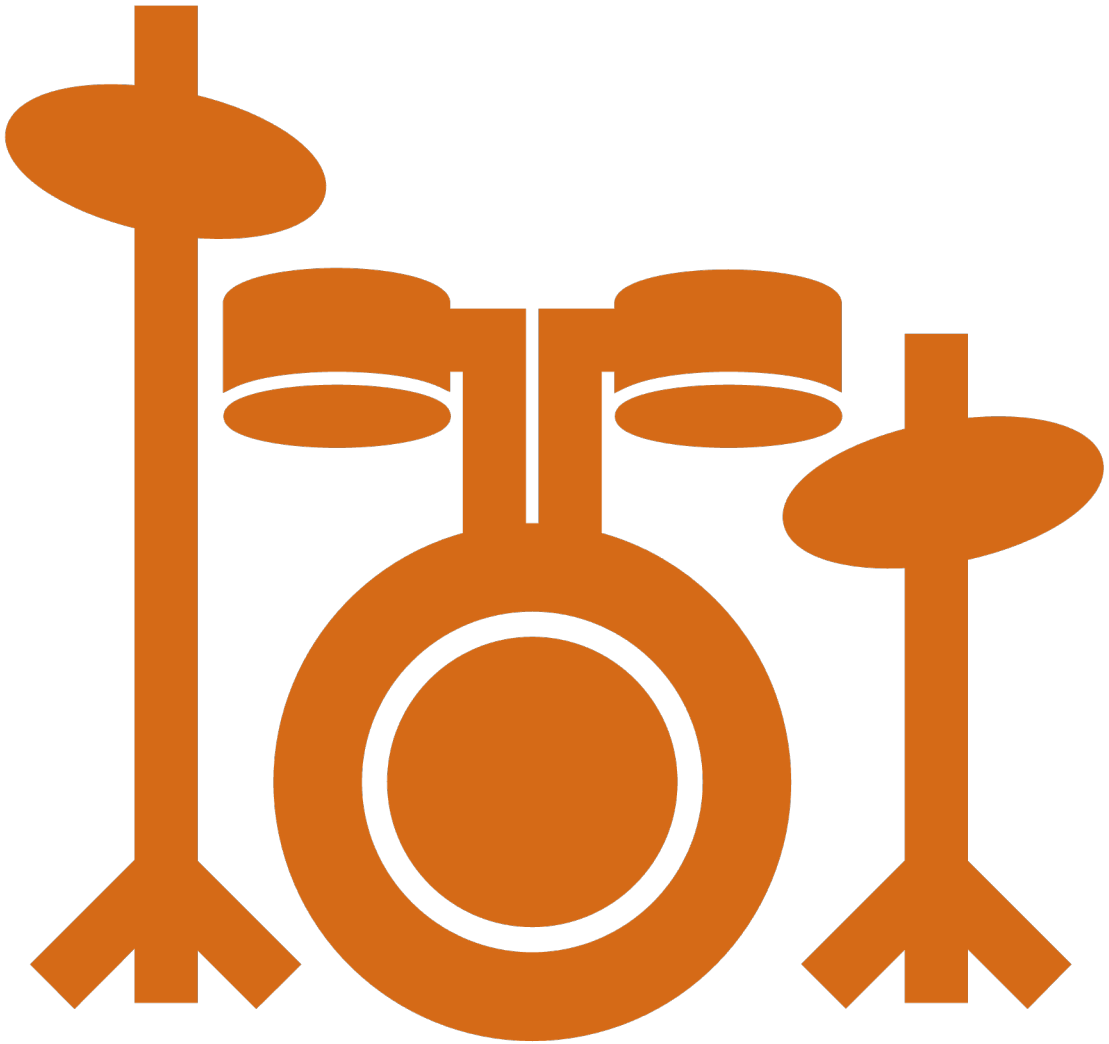
Pros:

- Idea is simple
- Easy to implement
- Served as baseline

Cons:

- Not personalized (users and songs' information is not considered)
- Some songs will never be listened

2. SAME ARTIST GREATEST HITS



IDEA

- Sort songs by popularity in a decreasing order
- For each user, the ranking of songs is re-ordered to place songs by artists
- recommend the songs in the new order, except those already in the user's profile

PROS & CONS

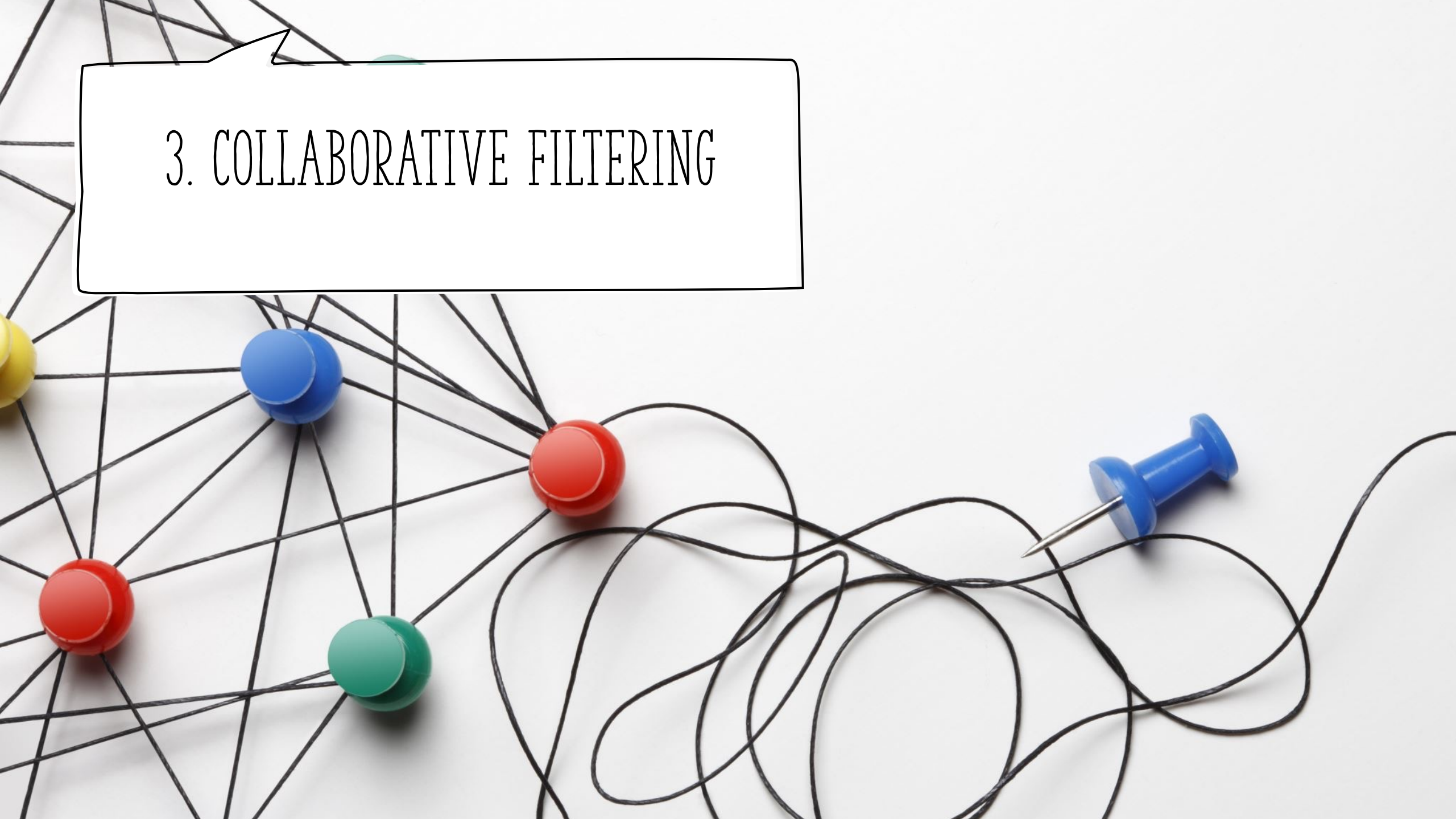
Pros:

- Idea is simple
- Easy to implement
- Minimum personalized

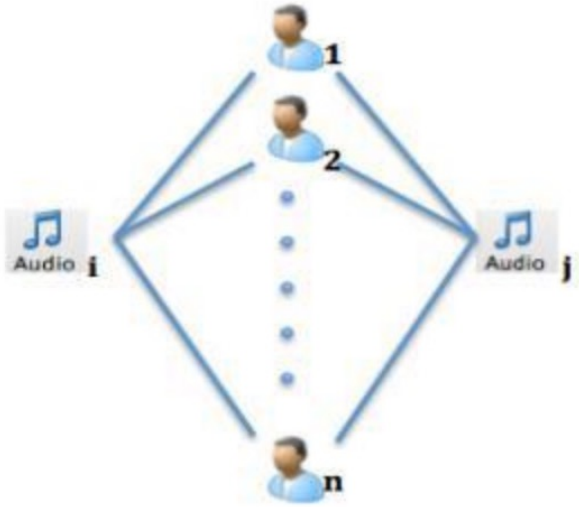
Cons:

- Only single-meta-data is used.
- Maximally conservative: doesn't explore the space beyond songs with which the user is likely already familiar

3. COLLABORATIVE FILTERING



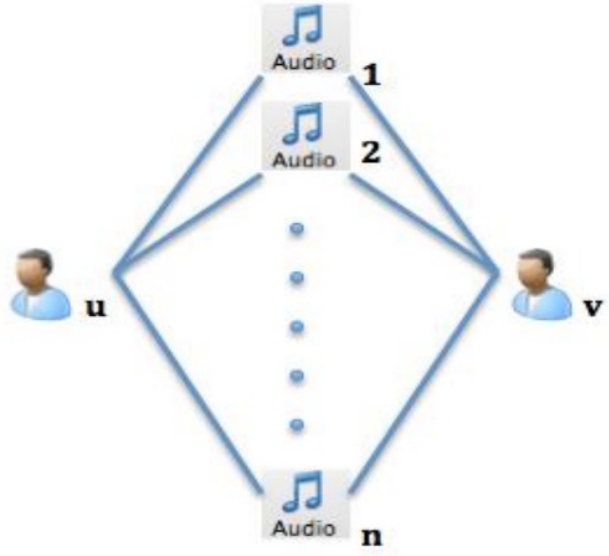
Songs that are often listened by the same user tend to be similar and are more likely to be listened together in future by some other user.



Item - based

Using collaborative filtering model, we got an accuracy of 8.2117 which marks the highest of all models.

Users who listen to the same songs in the past tend to have similar interests and will probably listen to the same songs in future.

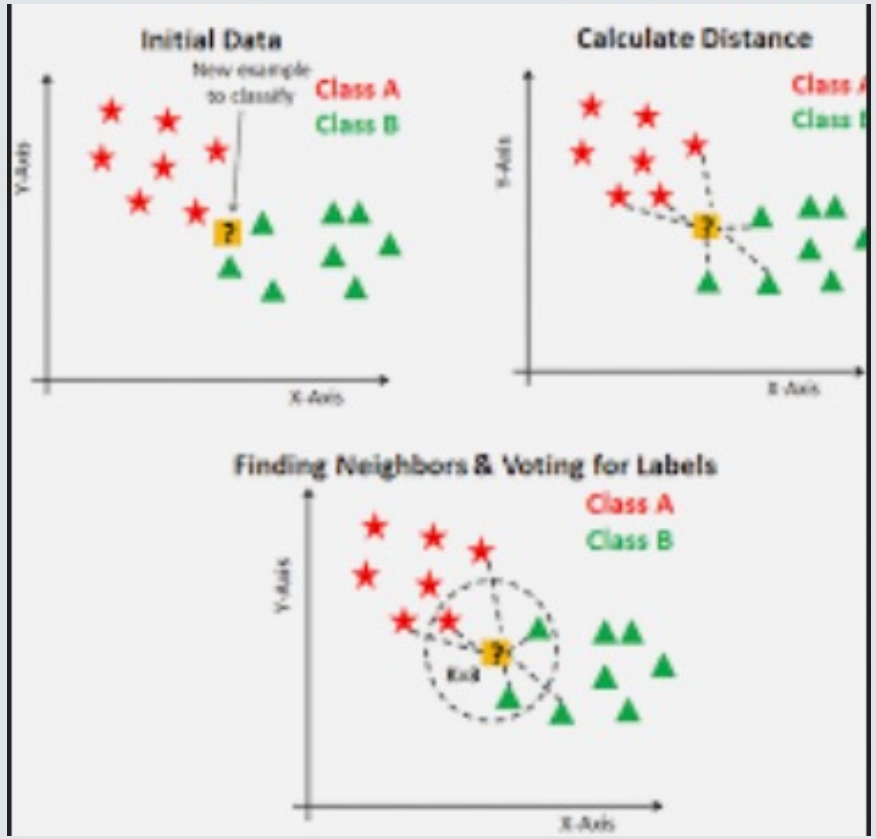


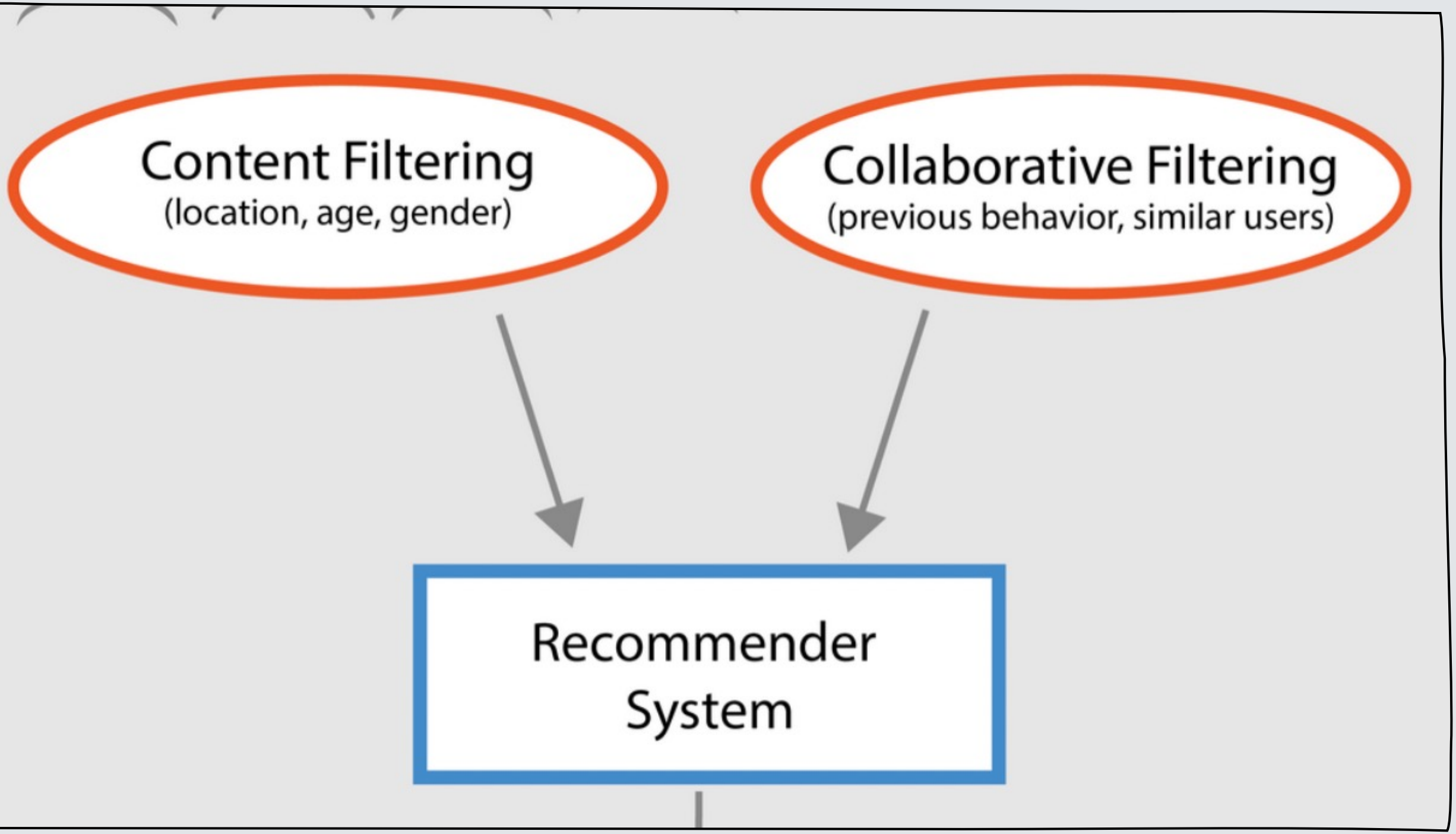
User - based

K - NEAREST NEIGHBOR TECHNIQUE

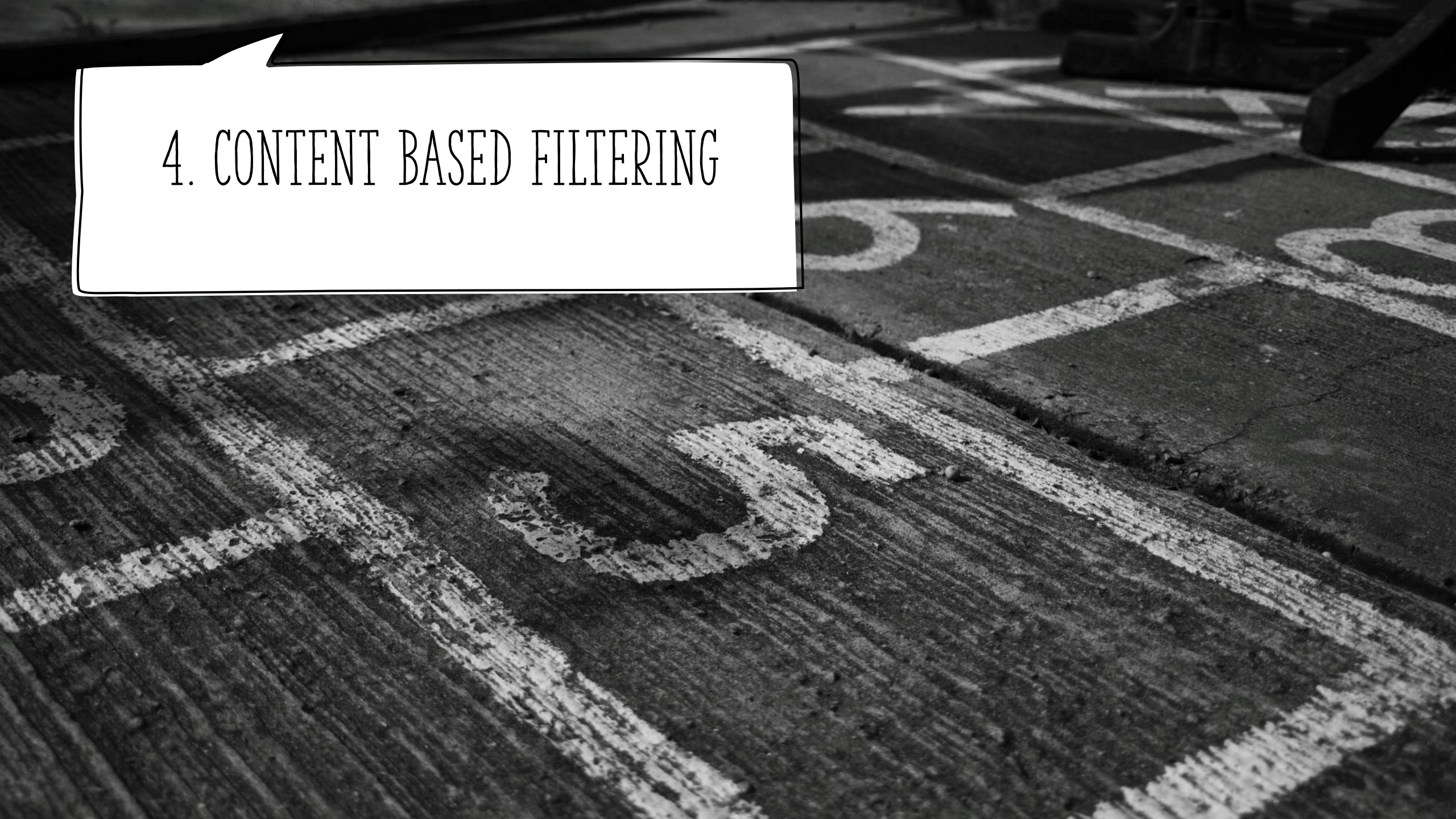
- K-NN is considered the standard method when it comes to both user-based and item-based collaborative filtering approaches.
- Create a space of songs according to songs features. We find out neighborhood of each song.
- The algorithm will calculate the distance between the target song and every other song in the dataset, it will then make a ranking according to their distances. Finally, it will return the top k nearest neighbor songs as song recommendations.

Using K-NN model, we got an accuracy of 0.6867




Content Filtering
(location, age, gender)
Collaborative Filtering
(previous behavior, similar users)
Recommender System

4. CONTENT BASED FILTERING



- Based on item's description and user's preference profile.
- Not based on choices of other users with similar interests.
- We make recommendations by looking for music whose features are very similar to the tastes of the user.

- Similarity != recommendation (no notion of personalization)
- Majority of songs have too few listeners, so difficult to "collaborate"

Using SVD model, we got an accuracy of 3.18

SVD'S LATENT FACTOR MODEL

- Meta-data is fully used, all the information is well explored
- Listening histories are influenced by a set of factors specific to the domain (e.g., Genre, artist).
- These factors are in general not obvious, and we need to infer those so-called latent factors from the data.
- Users and songs are characterized by latent factors.
- Personalized

MATRIX M

1	0	1	1	0	0	...	
1	1	0	0	0	0		
0	1						
...							

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1						
User 2						
User 3						
User 4						

To handle such a large amount of data, we build a sparse matrix from user-song triplets and directly operate on the matrix, instead of looping over millions of songs and users. We used truncated SVD for dimensionality reduction.

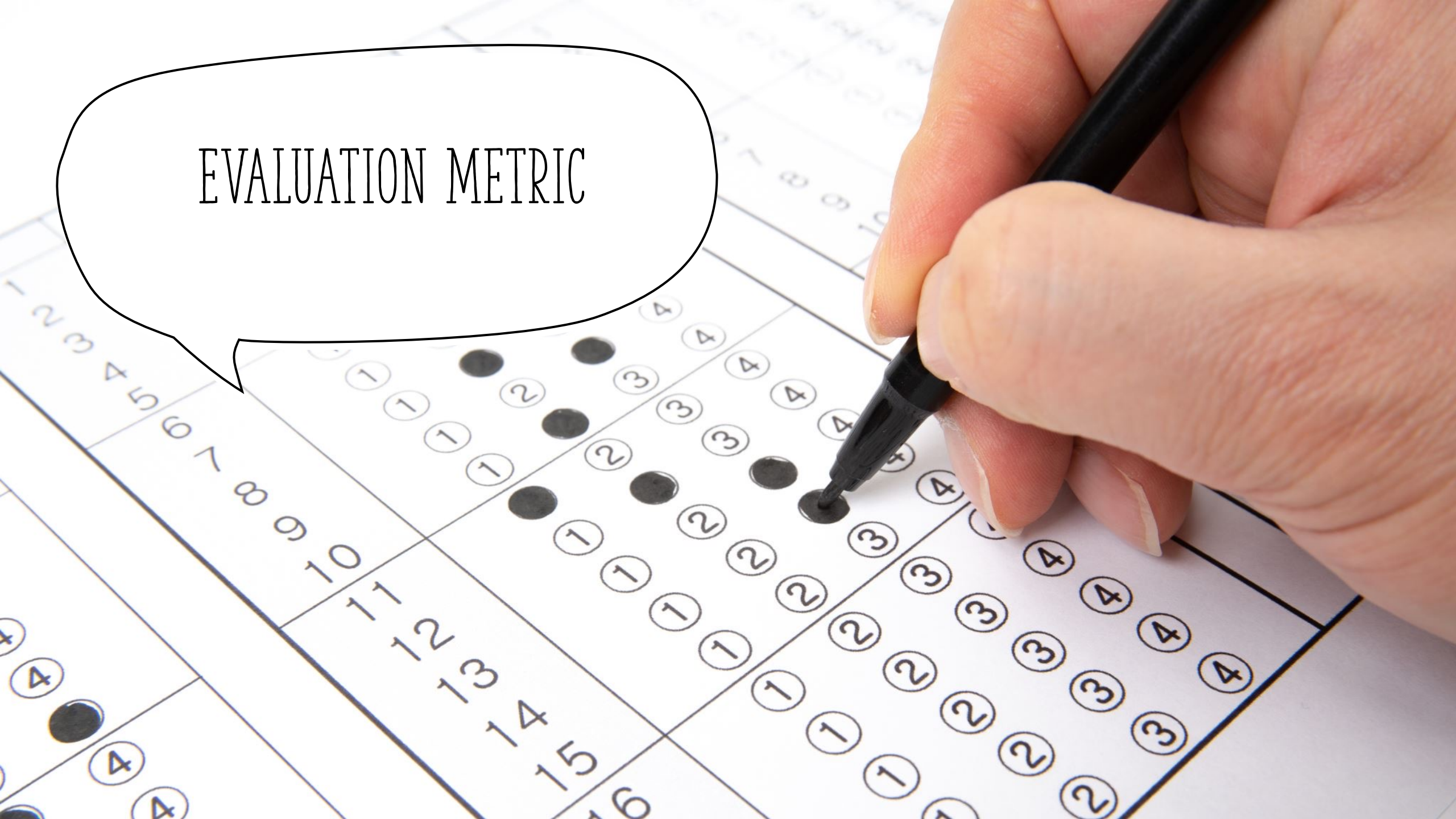
- Firstly, we decompose Matrix M into latent feature space that relates user to songs.

$M = U^T . V,$

U = user factors, V = item factors

- Then, for each user, a personalized recommendation is giving by ranking following item for each song $W_i = U^t_u . V_i$

EVALUATION METRIC



MEAN AVERAGE PRECISION

We used mean Average Precision (mAP) as our evaluation metric.

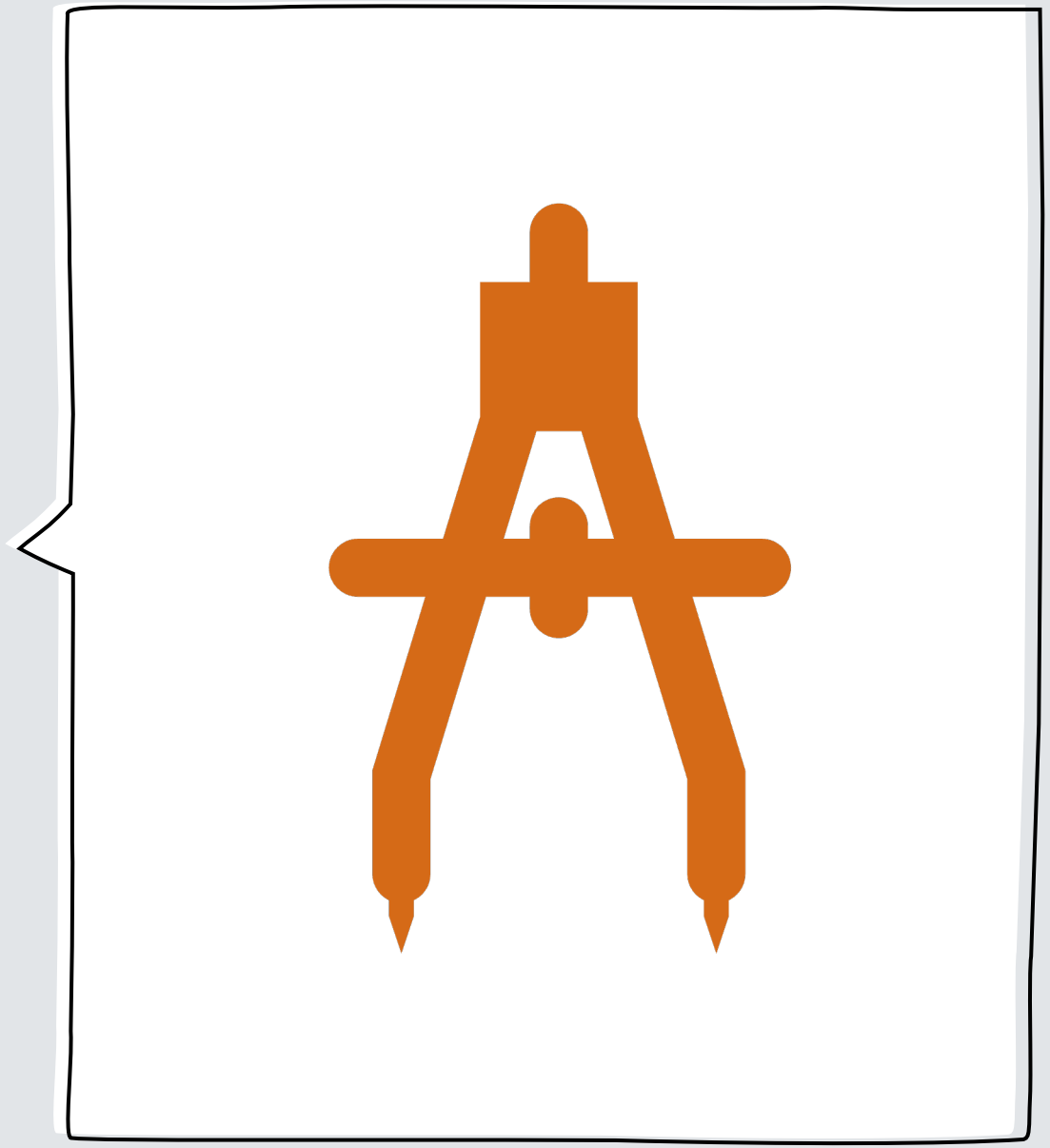
- Precision is much more important than recall because false positives can lead to a poor user experience.
- Our metric gives the average of the proportion of correct recommendations giving more weight to the top recommendations.

STEPS IN COMPUTING MEAN AVERAGE PRECISION

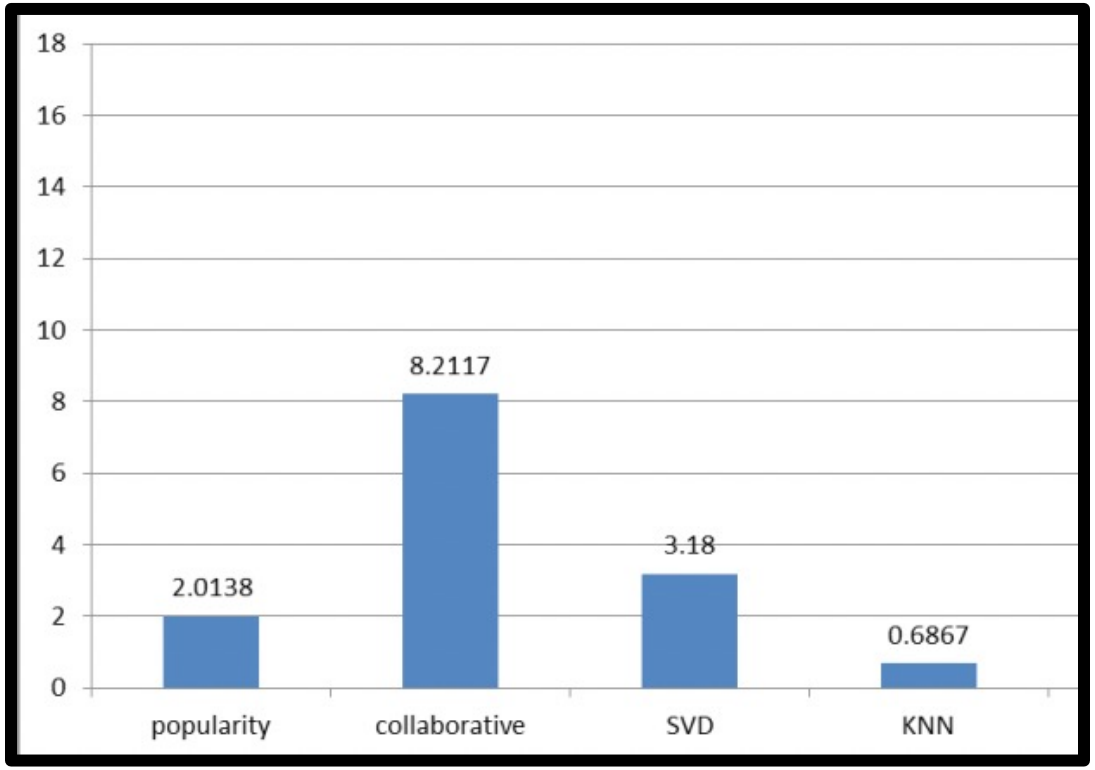
There are three steps of computing mAP:

1. Firstly, precision at each k is calculated. It gives the proportion of correct recommendations within the top-k of the predicted rankings.
2. Then, for each user, average precision at each k is evaluated.
3. Finally, mean of all the users is taken.

$$mAP = \frac{1}{m}\sum_{u=1}^{m} AP(u, r_u)$$

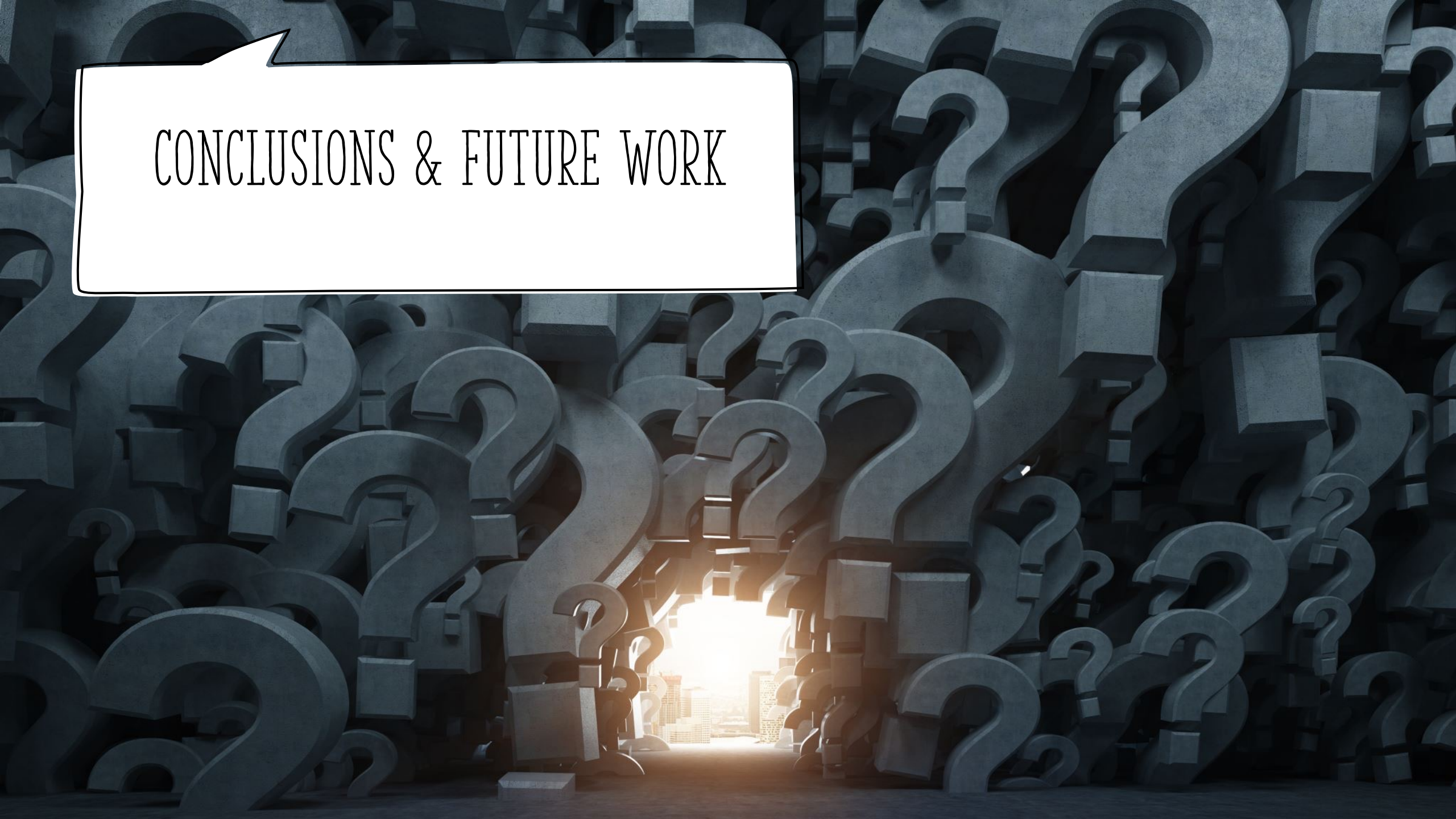


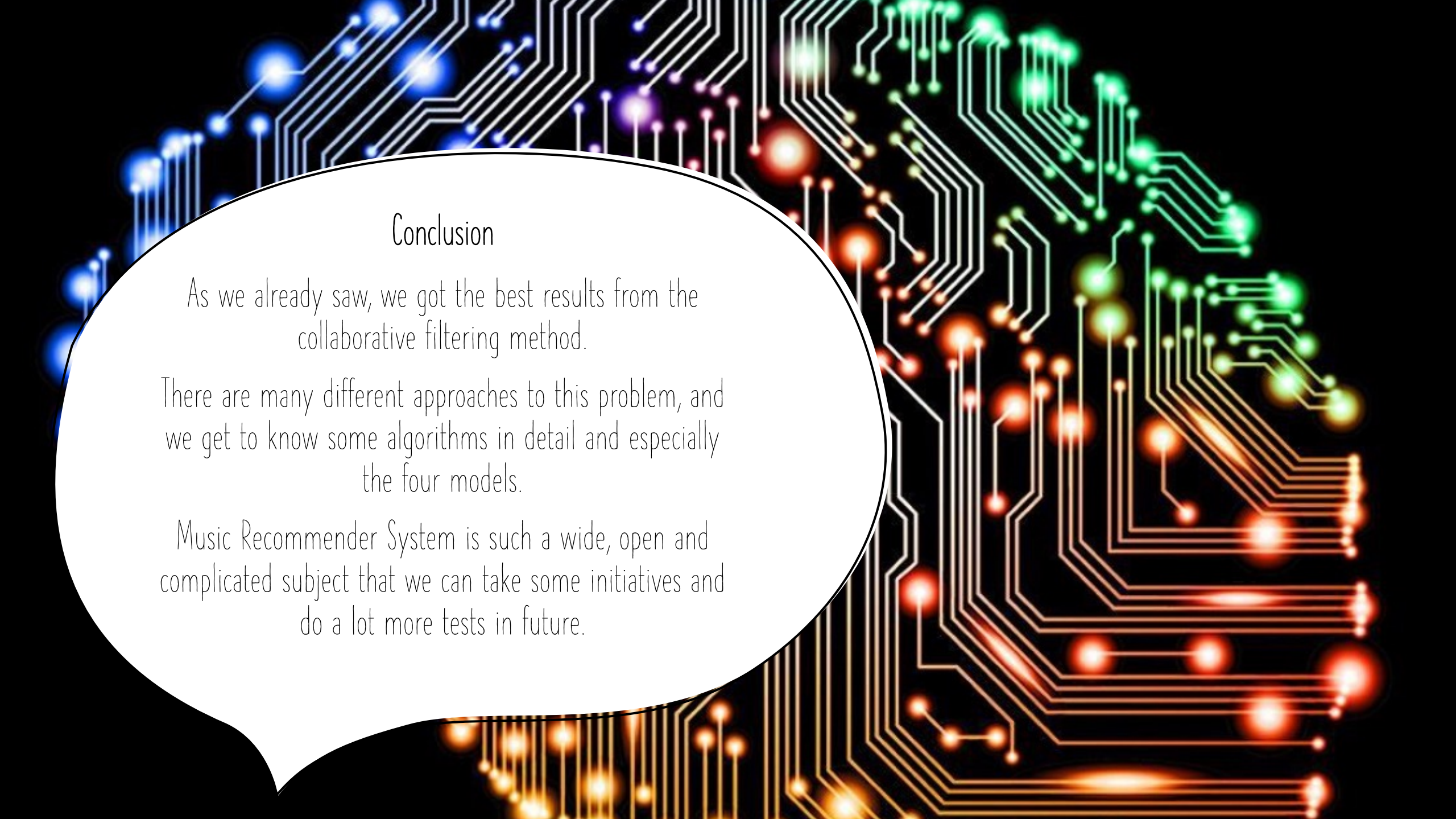
RESULTS & OBSERVATIONS



- We got best results for memory based collaborative filtering algorithm.
- Our SVD based latent factor model gives better results than popularity-based model. It lags collaborative filtering algorithm because the matrix was too sparse which prevented objective functions to converge to global optimum.
- Our K-NN model did not work well and performs worse than even the popularity model.
- The reason behind this is that we have the features of only 10,000 songs, which is less than 3 % of the whole dataset so only some of these 10,000 songs could be recommended.

CONCLUSIONS & FUTURE WORK





Conclusion

As we already saw, we got the best results from the collaborative filtering method.

There are many different approaches to this problem, and we get to know some algorithms in detail and especially the four models.

Music Recommender System is such a wide, open and complicated subject that we can take some initiatives and do a lot more tests in future.

FUTURE WORK

- Run the algorithms on a distributed system, like Hadoop or Condor, to parallelize the computation, decrease the runtime and leverage distributed memory to run the complete MSD.
- Combine different methods and learn the weightage for each method according to the dataset


THANK YOU


ANY QUESTIONS?