

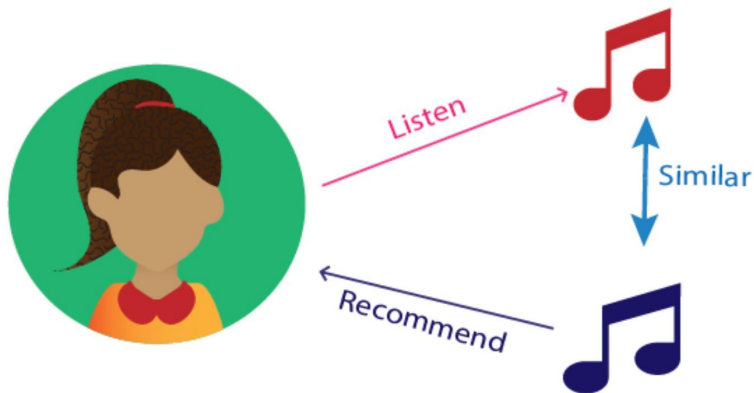
Musical Recommender

Muggles, or Mia and the Muggles

Bryan Fenchel, Mia Kobayashi,
Alex Zeng-Yang

Goal

- Goal 1:
 - Build a model that accepts a song as input and outputs a list of song recommendations
 - Inspo: Jenny Tang's book recommender practical as a starting point (but for music & make it **so much** better)
 - github.com/Jennytang1224/BookRecommender
 - Desired model is collaborative filtering using (KNN)
 - Known to utilize cosine similarity
- Goal 2:
 - Implement collaborative filtering model with implicit (a production ready collaborative filtering python library)
 - Pass the user id, user artist matrix, and number of new artists we'd like to recommend to the user.
 - Matrix factorization with alternating least squares



Intro to Recommendation Systems:

Recommenders are everywhere! Netflix, Amazon, Spotify, dating sites, news, github...

Before e-commerce stores,

Limited inventory -> best sellers.

E-commerce changed everything!

Unlimited inventory -> niche products.

Is more always better?

The Tasting Booth Experiment

When Choice is Demotivating: Can One Desire Too Much of a Good Thing?

Sheema S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

6 jam samples

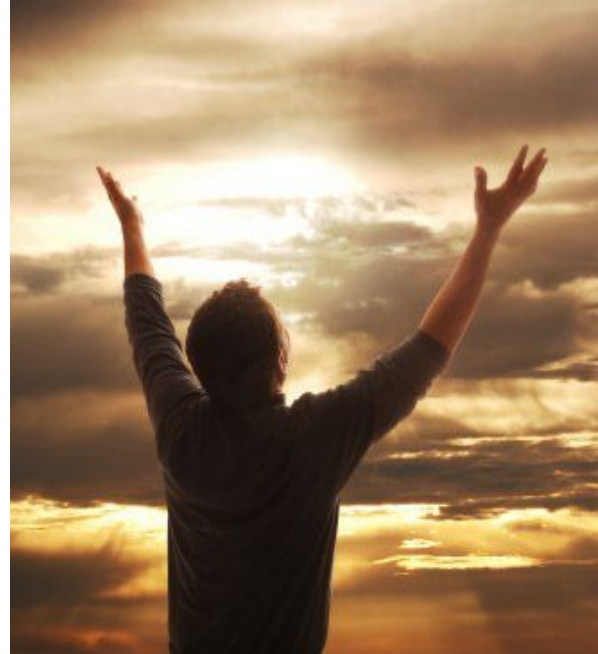
24 jam samples

"30% of the consumers in the limited-choice condition subsequently purchased a jar of jam; in contrast, only 3% of the consumers in the extensive-choice condition did so"

Too Much Variety is overwhelming!

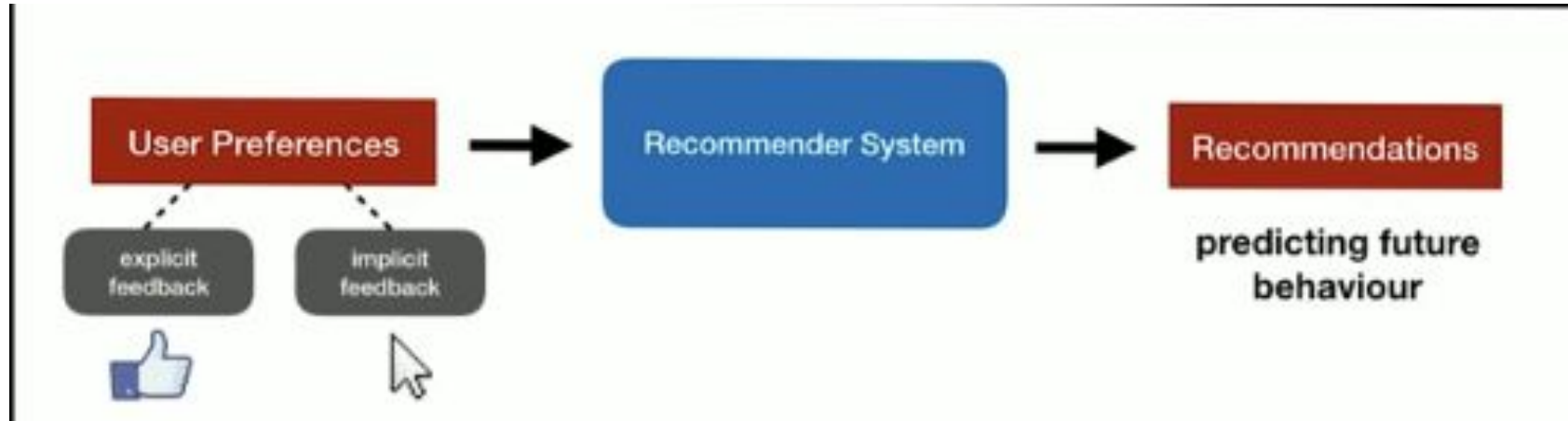
Can't you just show me what is relevant to my needs????!!

YES! Thanks to Recommendation systems.



So how does it work?

Did you like something? Did you binge something?



Two common approaches...

Collaborative filtering

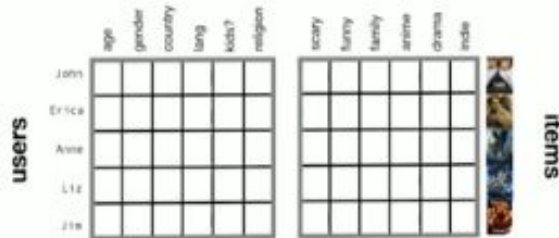


	1	2	3	4	5	6	7	8
John	1		3		5		5	
Erica			5		2		4	
Anne	5	2		1		4	3	2
Liz	3		4	3	4		5	
John	5	2		1		4		3

similar users like
similar things

- “Because you watched Movie X”
- “Customers who bought this item also bought”

Content-based filtering



	age	gender	country	lang	kids?	religion
John						
Erica						
Anne						
Liz						
John						

	scary	funny	family	anime	drama	indie

user and item features

- user features: age, gender, spoken language
- item features: movie genre, year of release, cast

Which one should you use?

Collaborative filtering requires a lot of data! If too sparse, not a good option.
Collaborative filtering does not work WELL for new items and new users!

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0



users

items							
		5.0			4.5		3.0
		3.0		4.0	2.5		3.0
2.0						1.0	
	3.0	3.5		5.0	4.5		
1.5	2.0			4.5		2.0	

Calculate Matrix Sparsity

$$\text{sparsity} = \frac{\text{\# ratings}}{\text{total \# elements}}$$

How do you deal with bias?

Normalization

- Optimists → rate everything 4 or 5
- Pessimists → rate everything 1 or 2
- Need to normalize ratings by accounting for user and item bias
- Mean normalization
 - subtract b_i from each user's rating for given item i

$$b_{ui} = \mu + b_i + b_u$$

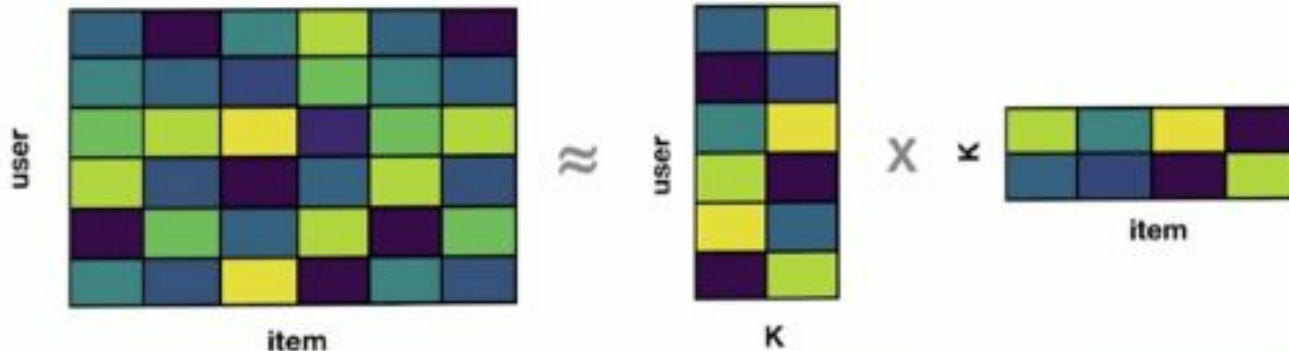
Diagram illustrating the components of the mean normalization equation:

- b_{ui} : user-item rating bias
- μ : global avg
- b_i : item's avg rating
- b_u : user's avg rating

Matrix Factorization!

- factorize the user-item matrix to get 2 latent factor matrices:
 - user-factor matrix
 - item-factor matrix
- missing ratings are predicted from the inner product of these two factor matrices

$$X_{mn} \approx P_{mk} \times Q_{nk}^T = \hat{X}$$



Potential Data

- Million Song Dataset
 - Downloaded from: [Kaggle](#)
 - 2 files:
 - 10000.txt
 - 3 columns: user_id, song_id, listen_count
 - song_data.csv
 - 5 columns: song_id, title, release, artist_name, year
- Music4All?
 - Has potential, waiting on access from owners
 - Contains 15,602 anonymous users, their listening histories, and 109,269 songs represented by their audio clips, lyrics, and 16 other metadata/attributes
- We want to wait for access to the Music4All database before deciding which dataset to use

Our Data Journey:

A contact in Vienna, **Marta Moscati**, working on her phd in music recommendation systems, suggested a data set.

Music-4-All is not available to all.

- First, in order to gain access to the database, we had to sign NDA's with the creators of the extensive 50gb database.

Our Data Journey (NDA Edition):

Music4All Database

Disclosure and Confidentiality Agreement

This agreement presents the terms and conditions to obtain the "Music4All database", which are listed below:

1. The Music4All database will be used for research purposes only. This database will NOT be shared/used for any commercial purpose or activity, without requesting written permission from the database owners.
2. You ensure that appropriate reference will be made to the database and its paper (mentioned below), in any research publication that involves reference/results/data in this database.

□ Igor André Pegoraro Santana and Fabio Pinelli and Juliano Donini and Leonardo Calharin and Rafael Blazus Mangolin and Yandre Maldonado e Gomes da Costa and Valéria Delisandro Faltim and Marcos Aurélio Domingues. Music4All: A New Music Database and its Applications. In: 27th International Conference on Systems, Signals and Image Processing (IWSSIP 2020), 2020, Niterói, Brazil. p. 399-404.

3. We understand that database will be shared on 'as it is' basis and no liability of any kind would be passed to the owners of the database.

By signing this document you agree with previous terms and conditions.

We count on your understanding and are at your disposal if you require any further information from our part. For any additional information, please send an e-mail to contact4music4all@gmail.com.

Date: 15 / 11 / 2022

Mia Kobayashi, student, University of San Francisco
Please replace this part with your name,
position and
institution.

Date: 15 / 11 / 2022

Mia Kobayashi, student, University of San Francisco
Please replace this part with your name,
position and
institution.

The Data: Music4All

(A) goal of the music information retrieval (MIR) community:

Research new methods and create new systems that can efficiently and effectively retrieve and recommend songs from large databases of music content.

In hopes of contributing to the MIR community, there exists Music4All, a new music database which contains metadata, tags, genre information, 30-second audio clips, lyrics, and so on.

- Features used for recommendation system:
 - Base recommendation system (Collaborative Filtering utilizing KNN and Cosine Similarity):
 - User ID, Song ID, Song count (number of listens for song per user)
 - Also, created a dictionary of {song ID: [song name, song artist]}

The Data:

Full (2597382 rows × 20 columns) dataset from all .csv files provided from *Music4All* database (with song_count addition):

	user	song_id	timestamp	artist	song	album_name	tags	genres	lang	spotify_id	popularity	release	danceability	energy	key	mode	valence	tempo	duration_ms	song_count
0	user_007XlJOr	DaTQ53TUmfP93FSr	2019-02-26 18:09	Mitski	Your Best American Girl	Puberty 2	2016,somafm,bagel,indie rock,noise pop,indie r...	indie rock,noise pop,indie rock,dream pop	en	172rW45GEnGoJUuWfm1drt	55.0	2016	0.360	0.257	7.0	1.0	0.130	76.972	212184	2
1	user_02DWuQOR	DaTQ53TUmfP93FSr	2019-03-05 19:46	Mitski	Your Best American Girl	Puberty 2	2016,somafm,bagel,indie rock,noise pop,indie r...	indie rock,noise pop,indie rock,dream pop	en	172rW45GEnGoJUuWfm1drt	55.0	2016	0.360	0.257	7.0	1.0	0.130	76.972	212184	12
2	user_0BZUk6bj	DaTQ53TUmfP93FSr	2019-03-15 12:17	Mitski	Your Best American Girl	Puberty 2	2016,somafm,bagel,indie rock,noise pop,indie r...	indie rock,noise pop,indie rock,dream pop	en	172rW45GEnGoJUuWfm1drt	55.0	2016	0.360	0.257	7.0	1.0	0.130	76.972	212184	2
3	user_0PJuaOVH	DaTQ53TUmfP93FSr	2019-02-28 21:34	Mitski	Your Best American Girl	Puberty 2	2016,somafm,bagel,indie rock,noise pop,indie r...	indie rock,noise pop,indie rock,dream pop	en	172rW45GEnGoJUuWfm1drt	55.0	2016	0.360	0.257	7.0	1.0	0.130	76.972	212184	1
4	user_0QbKRt8m	DaTQ53TUmfP93FSr	2019-02-06 08:06	Mitski	Your Best American Girl	Puberty 2	2016,somafm,bagel,indie rock,noise pop,indie r...	indie rock,noise pop,indie rock,dream pop	en	172rW45GEnGoJUuWfm1drt	55.0	2016	0.360	0.257	7.0	1.0	0.130	76.972	212184	1
...

Simple dataset used to formulate model:

	user	song_id	song_count
0	user_007XlJOr	DaTQ53TUmfP93FSr	2
1	user_02DWuQOR	DaTQ53TUmfP93FSr	12
2	user_0BZUk6bj	DaTQ53TUmfP93FSr	2
3	user_0PJuaOVH	DaTQ53TUmfP93FSr	1
4	user_0QbKRt8m	DaTQ53TUmfP93FSr	1
...

kNN / Collaborative Filtering | Cosine Similarity

- Item-Based Collaborative Filtering
 - In this approach, similarities between pair of items are computed using cosine similarity metric
- Process:
 - Find the common / existing values between two vectors
 - Apply equation:

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

https://en.wikipedia.org/wiki/Cosine_similarity

Recommender: The Model

```
from sklearn.neighbors import NearestNeighbors
```

```
metric = 'cosine'  
algorithm = 'brute'  
k = 5 #number of recommendations
```

```
modelKnn = NearestNeighbors(metric = metric, algorithm = algorithm)  
modelKnn.fit(countMatrixPivot.values)
```

```
▼ NearestNeighbors  
NearestNeighbors(algorithm='brute', metric='cosine')
```

```
#function to return recommended songs  
def makeRecKNN(k, sName, sArtist):  
    try:  
        songKey = correspondingKey(sName, sArtist)  
        song = countMatrixPivot.loc[songKey]  
    except KeyError as e:  
        print('Cannot find the song', e, 'in database!')  
        return  
  
    similarities = []  
    indices = []  
  
    distance, indice = modelKnn.kneighbors([song.values], n_neighbors = k + 1)  
  
    songNamesArtist = np.array(idToName(countMatrixPivot.iloc[indice[0]].index.values))  
    songNames = songNamesArtist[:, 0]  
    songArtists = songNamesArtist[:, 1]  
  
    similarities = 1 - distance.flatten()  
  
    recommended_books = pd.DataFrame({  
        'Song Name' : songNames,  
        'Song Artist' : songArtists,  
        'similarities': similarities}).sort_values(by = 'similarities', ascending = False)  
  
    return recommended_books
```


Recommender: In Action

```
makeRecKNN(5, 'Burn It To The Ground', 'Nickelback')
```

	Song Name	Song Artist	similarities
0	Burn It To The Ground	Nickelback	1.000000
1	Lullaby	Nickelback	0.503034
2	Take Over Control (Radio Edit)	Afrojack	0.453862
3	Rockstar	Nickelback	0.441870
4	Photograph	Nickelback	0.407601
5	New Bohemia	TransViolet	0.399150

Recommender: In Action

```
makeRecKNN(5, 'Bohemian Rhapsody - 2011 Remaster', 'Queen')
```

	Song Name	Song Artist	similarities
0	Bring Back That Leroy Brown	Queen	1.000000
1	In The Lap Of The Gods...Revisited	Queen	0.744686
2	Misfire	Queen	0.725908
3	Brighton Rock	Queen	0.687961
4	White Queen (As It Began)	Queen	0.681188
5	She Makes Me (Stormtrooper in Stilettos)	Queen	0.662266

Recommender: In Action

```
makeRecKNN(5, 'Money, Money, Money', 'ABBA')
```

	Song Name	Song Artist	similarities
0	Ring Ring	ABBA	1.000000
1	Tack för en underbar vanlig dag	Agnetha Fältskog	0.662541
2	Words Of Love	The Mamas & the Papas	0.592157
3	Mi Rubi L'anima	Laura Pausini	0.569803
4	Se Me Esta Escapando	ABBA	0.569803
5	Gimme! Gimme! Gimme! (A Man After Midnight) - ...	ABBA	0.548293

That's all cool, but...is it precise????????

Precision@K

- of the top K recommendations, what proportion are relevant to the user?
- Of top 10, top 5... how relevant are they? Tune Hyperparameters...

Alternating Least Square's Hyperparameters

- k (# of factors)
- λ (regularization parameter)

Goal 2: Matrix Factorization with ALS

- A mathematical tool for playing around with matrices
 - Applicable in many scenarios where one would like to find out something hidden under the data
- ALS
 - Pass the user id, user artist matrix, and number of new artists we'd like to recommend to the user...
- COLLABORATIVE FILTERING... THEN... CONTENT BASED FILTERING... THEN... HYBRID?

	Feature 1	Feature 2
User 1	?	?
User 2	?	?
User 3	?	?
User 4	?	?
User 5	?	?

X

	Item 1	Item 2	Item 3	Item 4	Item 5
Feature 1	?	?	?	?	?
Feature 2	?	?	?	?	?

=

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0?	3	0?	3	0?
User 2	4	0?	0?	2	0?
User 3	0?	0?	3	0?	0?
User 4	3	0?	4	0?	3
User 5	4	3	0?	4	0?

Matrix Factorization: Resources, Papers, Code

LFM-2b Dataset Corpus of Music Listening Events for Music Recommendation & Retrieval :

More than two billion listening events, intended to be used for various music retrieval and recommendation tasks.

RecSys'22 paper ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations

Alessandro karapostK on Github :

“This repository hosts the code and the additional materials for the paper "ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations" by Alessandro B. Melchiorre, ...”

[How to Design and Build a Recommendation System Pipeline in Python \(Jill Cates\)](#)

[Build a Spotify-Like Music Recommender System in Python - THE SOUND OF AI](#)

TODO: Pursue Aspirations