COMP9727 Music Recommendation 2024 T2

Aditi Sachan (z5379867)

Lingyue Feng (z5446236)

Phu Dien Ma(z5432091)

Pyae Sone Oo(z5271704)

Recommendation Problems

- Personalization
- Cold Start Problem
- Exploration vs Exploitation
- Dynamic Adaption

Deployment within larger system

- Backend Integration
  - Data collection
  - Data Processing
  - Recommendation Engine

- Frontend Integration
  - User Interface
  - User Feedback

Deployment within larger system

- Recommendation Workflow
  - Initialization
  - Daily Recommendations
  - User Interaction
- Scalability and Performance
  - Cloud Infrastructure
  - API Integration
  - Caching and Optimization
- User Engagement and Retention
  - Personalized Experience
  - Social Features

Competitor Analysis

- Exploration & Exploitation Balance
- Cold Start Solution
- Real-time Adaption
- User Engagement

User Inputs and Recommended Outputs

- User Profile Information
- Top Artists and Tracks
- Liked Songs
- Playlists
- Track Interactions
- Daily Playlist
- Exploratory Recommendations

Utilizing User Feedback in Recommender Systems

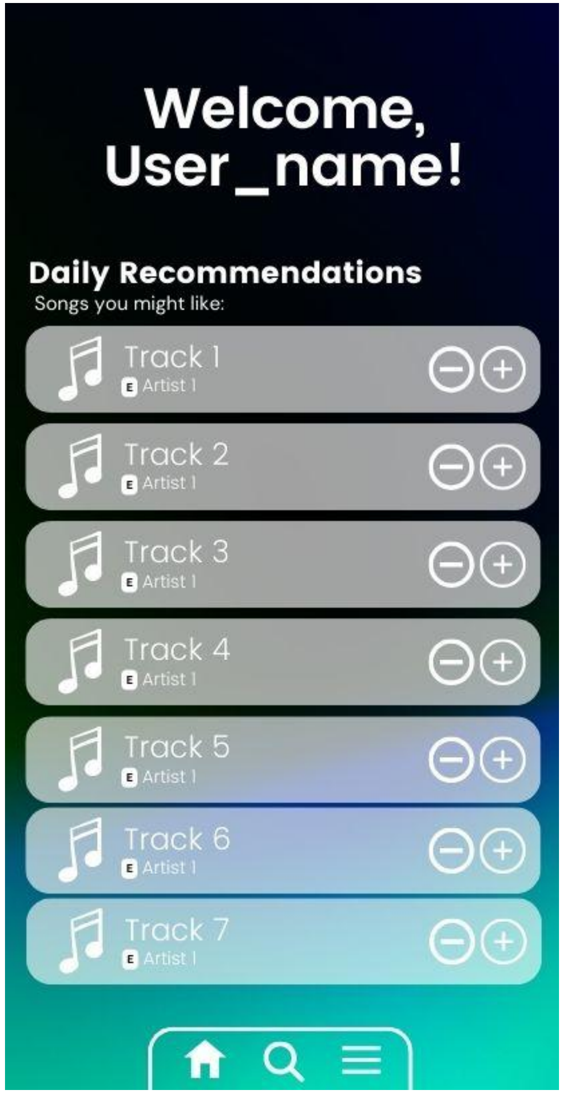
- Interaction Options
- Feedback Collection
- Usage of Feedback

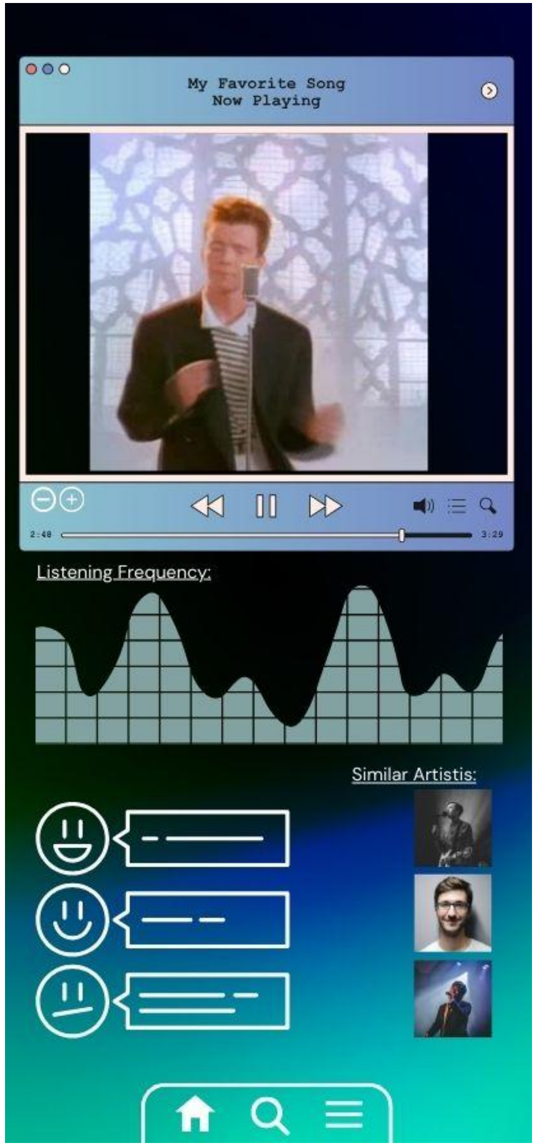
Defining Recommendation Problems

- Prediction
- Rating Estimation
- Ranking

Using UI Mockups to Illustrate System Usage

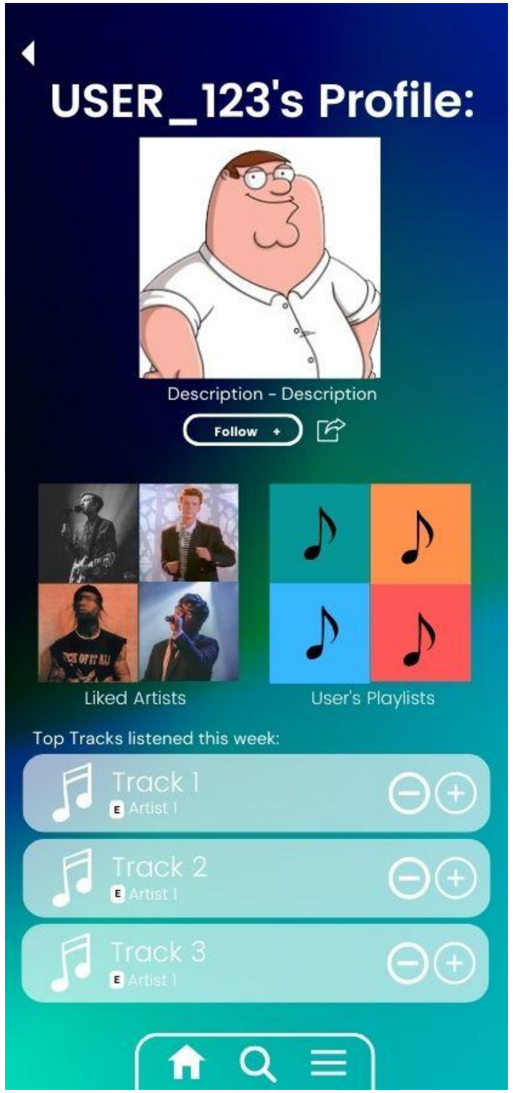
Daily Recommendation Screen





Interaction Feedback Screen

User Profile Screen



DATASET

Dataset(s) Used

Primary Dataset: Music4All Database

Files Included:

- `listening_history.csv`: User listening history (user, song, timestamp)
- `id_tags.csv`: Song IDs and associated tags
- `id_genres.csv`: Song IDs and associated genres
- `id_information.csv`: Song IDs with artist, song name, album name
- `id_metadata.csv`: Song attributes from Spotify API
- `id_lang.csv`: Song IDs and language of lyrics

Basic Characteristics of the Dataset(s)

Listening History:

- Attributes: *user, song, timestamp*

Song Information:

- Attributes: *song ID, artist, song name, album name, tags, genres, language, metadata Popularity, Release, Danceability, Energy, Key Mode, Valence, Tempo, Duration_ms*

Basic Characteristics of the Dataset(s)

Feature Extraction:

- **Attributes Used**: tags, genres, language, metadata, danceability, energy, valence, tempo
- **Techniques**:
  - **Textual Features**: Use TF-IDF Vectorizer for tags, genres and languages.
  - **Numerical Features**: Use normalization for danceability, energy, valence, and tempo.

Linking Different Datasets

Integration Strategy:

- Linking using song IDs across different files
- Example: `id_information.csv` linked with `id_genres.csv` and `id_tags.csv` using song IDs
- Song names in reference to that song id, retrieved from `id_information.csv`

FIRST FEW ROWS AND SUMMARY OF DATASETS

Listening History Dataset

Listening History: **First few rows**

	user	song	timestamp
0	user_007XIjOr	DaTQ53TUmfP93FSr	2019-02-20 12:28
1	user_007XIjOr	dGeyvi5WCOjDU7da	2019-02-20 12:35
2	user_007XIjOr	qUm54NYOjeFhmKYx	2019-02-20 12:48
3	user_007XIjOr	FtnuMT1DlevSR2n5	2019-02-20 12:52
4	user_007XIjOr	LHETTZcSZLeaVOGh	2019-02-20 13:09

Listening History **Summary:**

	user	song	timestamp
count	5109592	5109592	5109592
unique	14127	99596	122340
top	user_N9OKtRH0	32m5suoC94ytD8Ed	2019-02-08 20:35
freq	500	82871	175

Tags Dataset

ID Tags: **First few rows**

	id	tags
0	0009fFIM1eYThaPg	pop,british,female vocalists,dance,cheryl cole
1	0010xmHR6UICBOYT	instrumental hip-hop,underground hip hop,instr...
2	002Jyd0vN4HyCpqL	hard rock,rock,classic rock,american artist
3	006TYKNjNxWjfKjy	symphonic metal,power metal,symphonic power metal
4	007LIJOPQ4Sb98qV	post-punk,new wave,1985

ID Tags **Summary**:

	id	tags
count	109269	109269
unique	109269	80247
top	0009fFIM1eYThaPg	pop
freq	1	338

Genres Dataset

ID Genres: **First few rows**

	id	genres
0	0009fFIM1eYThaPg	pop
1	0010xmHR6UICBOYT	underground hip hop
2	002Jyd0vN4HyCpqL	hard rock,rock,classic rock
3	006TYKNjNxWjfKjy	symphonic metal,power metal,symphonic power metal
4	007LIJOPQ4Sb98qV	post-punk,new wave

ID Genres **Summary:**

	id	genres
count	109269	109269
unique	109269	23520
top	0009fFIM1eYThaPg	pop
freq	1	6092

Information Dataset

```
ID Information: First few rows
                 id               artist                song  \
0  0009fFIM1eYThaPg               Cheryl          Rain on Me
1  0010xmHR6UICBOYT             Oddisee       After Thoughts
2  002Jyd0vN4HyCpqL    Blue Öyster Cult              ME 262
3  006TYKNjNxWjfKjy            Rhapsody   Flames of Revenge
4  007LIJOPQ4Sb98qV       The Chameleons           Nostalgia

                                       album_name
0                                         3 Words
1                             The Beauty in All
2                               Secret Treaties
3                    Legendary Years (Re-Recorded)
4  What Does Anything Mean? Basically (2009 Remas...
```

```
ID Information Summary:
                      id  artist    song
album_name
count             109269  109269  109269
109269
unique            109269   16269   87915
38363
top     0009fFIM1eYThaPg   Queen   Intro
Greatest Hits
freq                   1     264      65
175
```

Metadata Dataset

```
ID Metadata: First few rows
                   id              spotify_id  popularity  release  \
0  0009fFIM1eYThaPg  3eObKIfHKJ1nAPh0wTxFCc        12.0     2009
1  0010xmHR6UICBOYT  27szvF97Tu95GxN98N52fy        46.0     2013
2  002Jyd0vN4HyCpqL  273lBFpxUCwisTpdnF9cVb        31.0     1974
3  006TYKNjNxWjfKjy  1qZgergQ41vaD4zBf3AKXR        33.0     2017
4  007LIJOPQ4Sb98qV  6rVxJ3sN3Cz40MSLavbG1K        19.0     2009

   danceability  energy  key  mode  valence    tempo  duration_ms
0         0.635   0.746  6.0   1.0    0.548  110.973       229947
1         0.591   0.513  7.0   0.0    0.263  172.208       325096
2         0.319   0.925  2.0   1.0    0.658  157.630       285693
3         0.432   0.979  7.0   1.0    0.162   90.008       332867
4         0.357   0.708  9.0   1.0    0.470  123.904       326067
```

```
ID Metadata Summary:

         popularity        release   danceability         energy  \
count  109269.000000  109269.000000  109269.000000  109269.000000
mean       35.080608    2005.813488       0.520449       0.667162
std        14.756258      14.335056       0.173008       0.241372
min         0.000000    1013.000000       0.000000       0.000000
25%        25.000000    2001.000000       0.403000       0.502000
50%        34.000000    2011.000000       0.528000       0.712000
75%        45.000000    2016.000000       0.645000       0.872000
max        95.000000    2019.000000       0.988000       1.000000

                 key           mode        valence          tempo  \
count  109269.000000  109269.000000  109269.000000  109269.000000
mean        5.284079       0.624669       0.445504     122.753032
std         3.560797       0.484210       0.252159      28.997936
min         0.000000       0.000000       0.000000       0.000000
25%         2.000000       0.000000       0.236000     100.194000
50%         5.000000       1.000000       0.424000     121.079000
75%         9.000000       1.000000       0.639000     140.047000
max        11.000000       1.000000       0.998000     242.903000

        duration_ms
count  1.092690e+05
mean   2.425046e+05
std    1.003360e+05
min    7.229000e+03
25%    1.934810e+05
50%    2.275330e+05
75%    2.715330e+05
max    4.995315e+06
```

Language Dataset

ID Language: **First few rows**

```
                 id lang
0  0009fFIM1eYThaPg   en
1  0010xmHR6UICBOYT   en
2  002Jyd0vN4HyCpqL   en
3  006TYKNjNxWjfKjy   en
4  007LIJOPQ4Sb98qV   en
```

ID Language **Summary**:

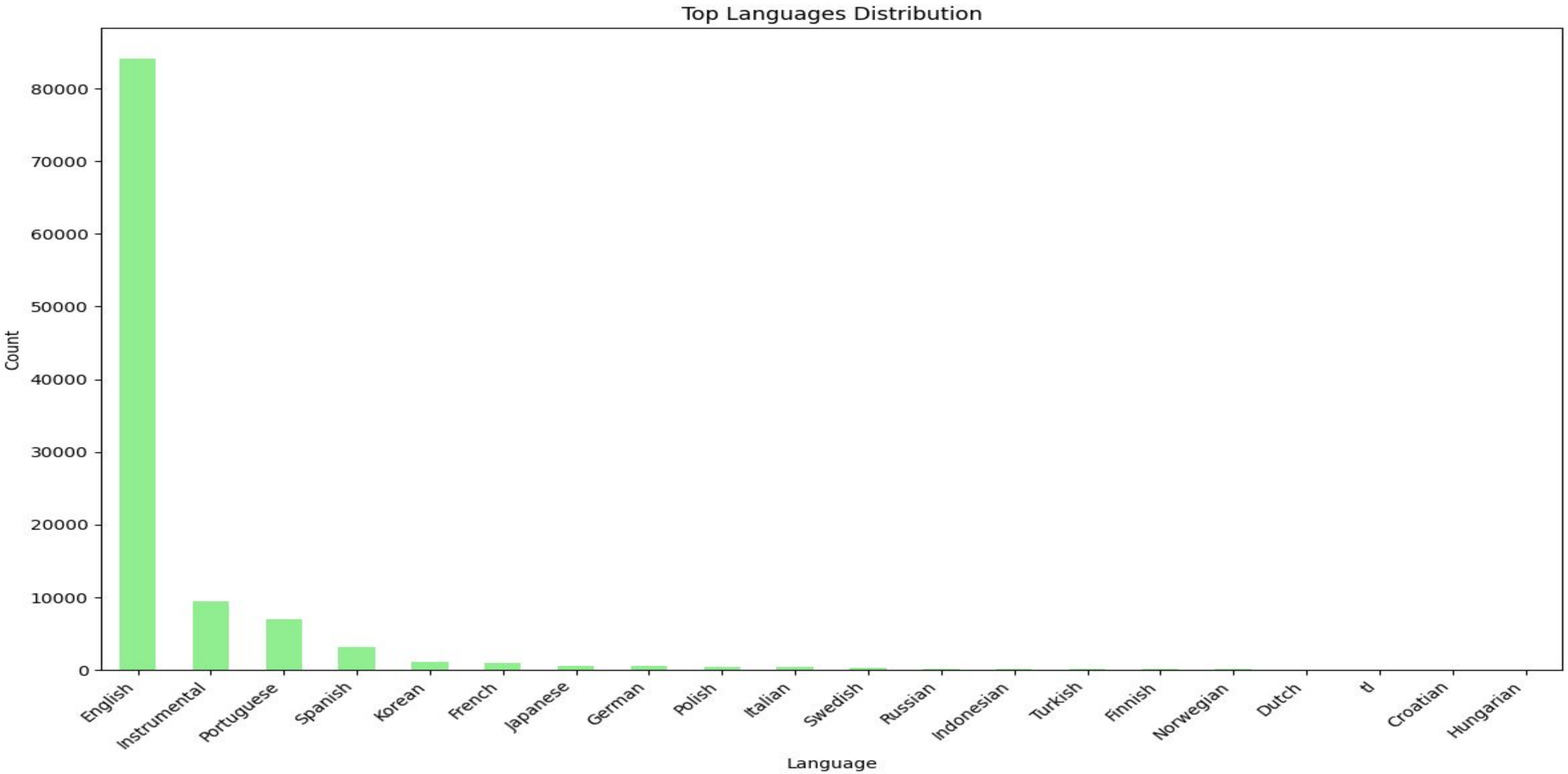
```
                      id    lang
count             109269  109269
unique            109269      46
top       0009fFIM1eYThaPg    en
freq                   1   84103
```

VISUALISATION

Language

Language Analysis:

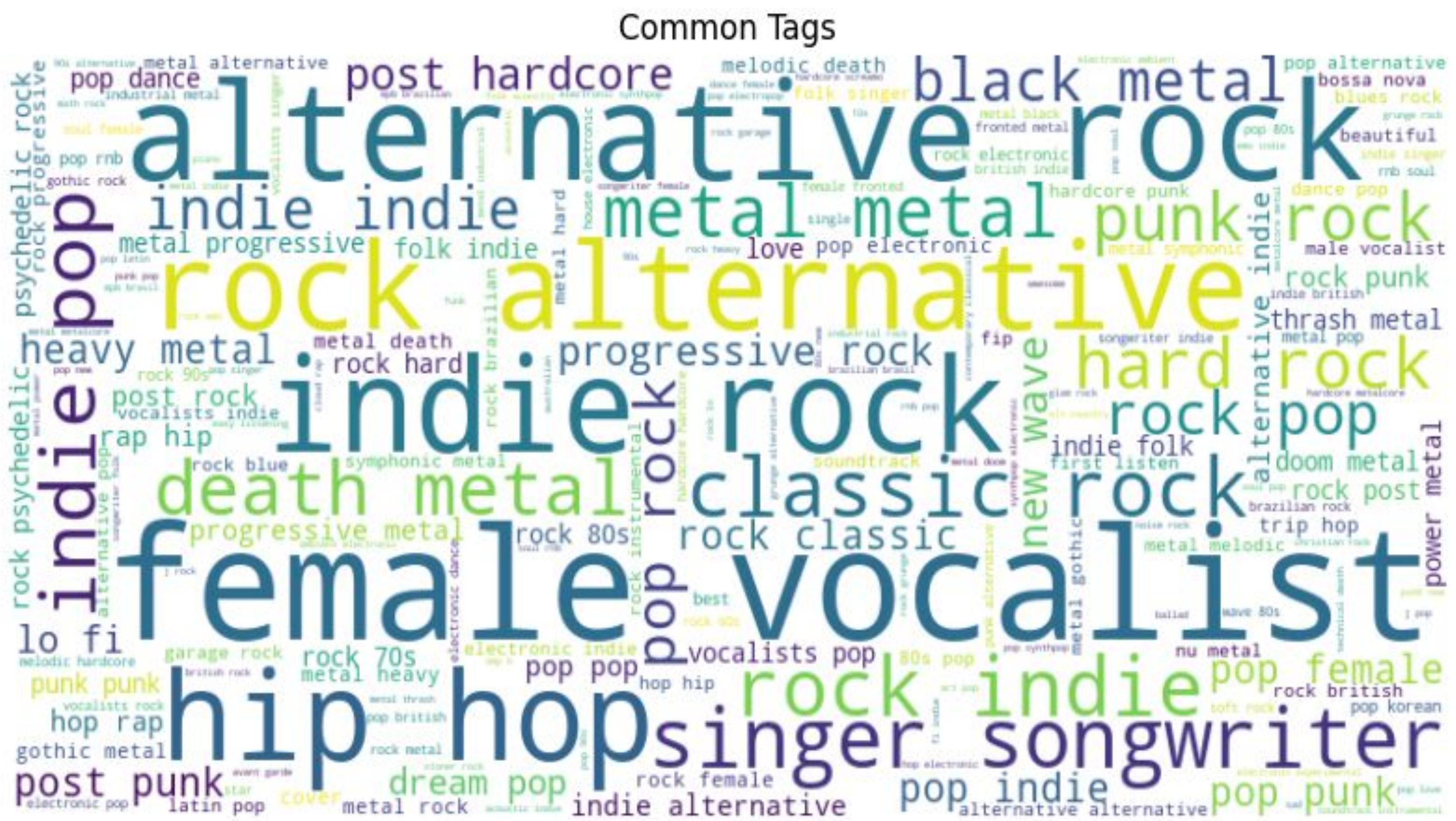
- Bar chart of Language distribution



Tags

Tag Analysis:

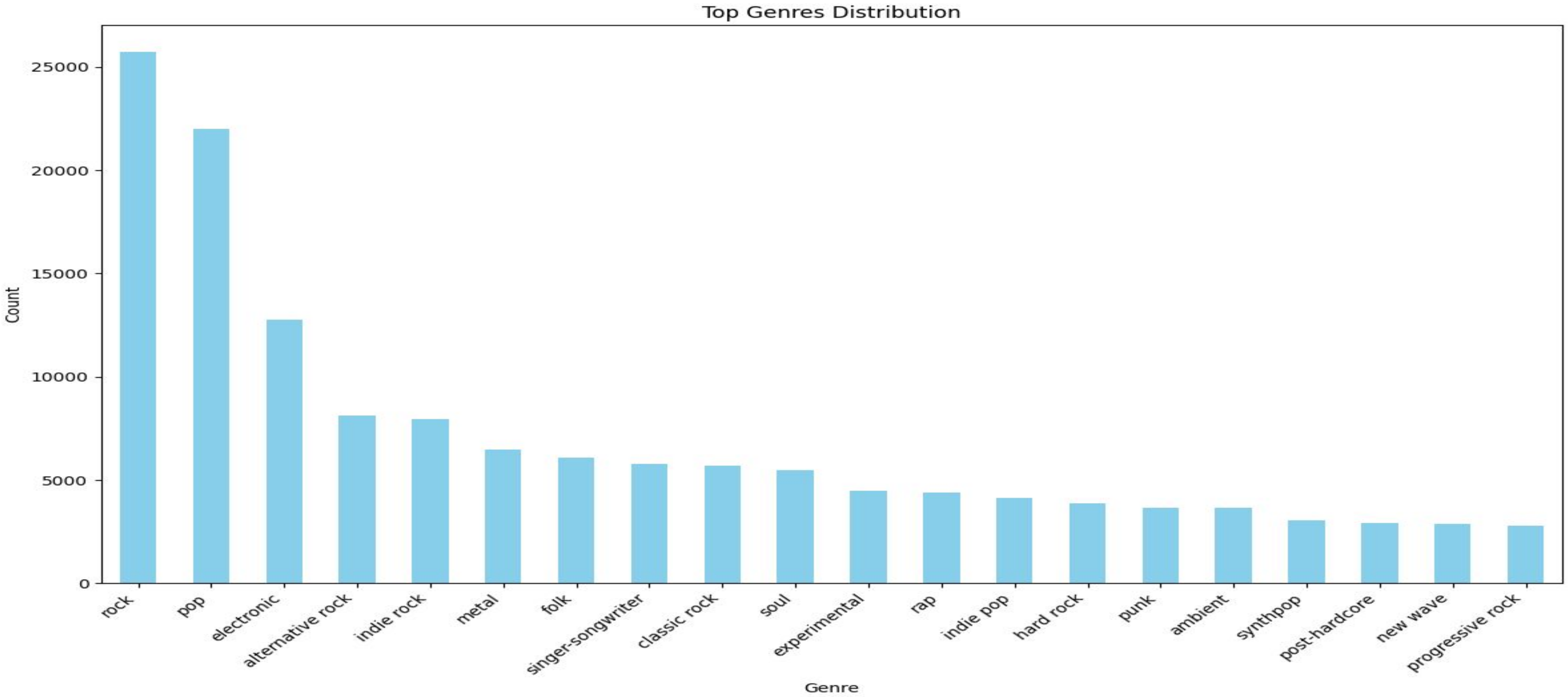
- Common tags (Word cloud)



Genres

Genre Analysis:

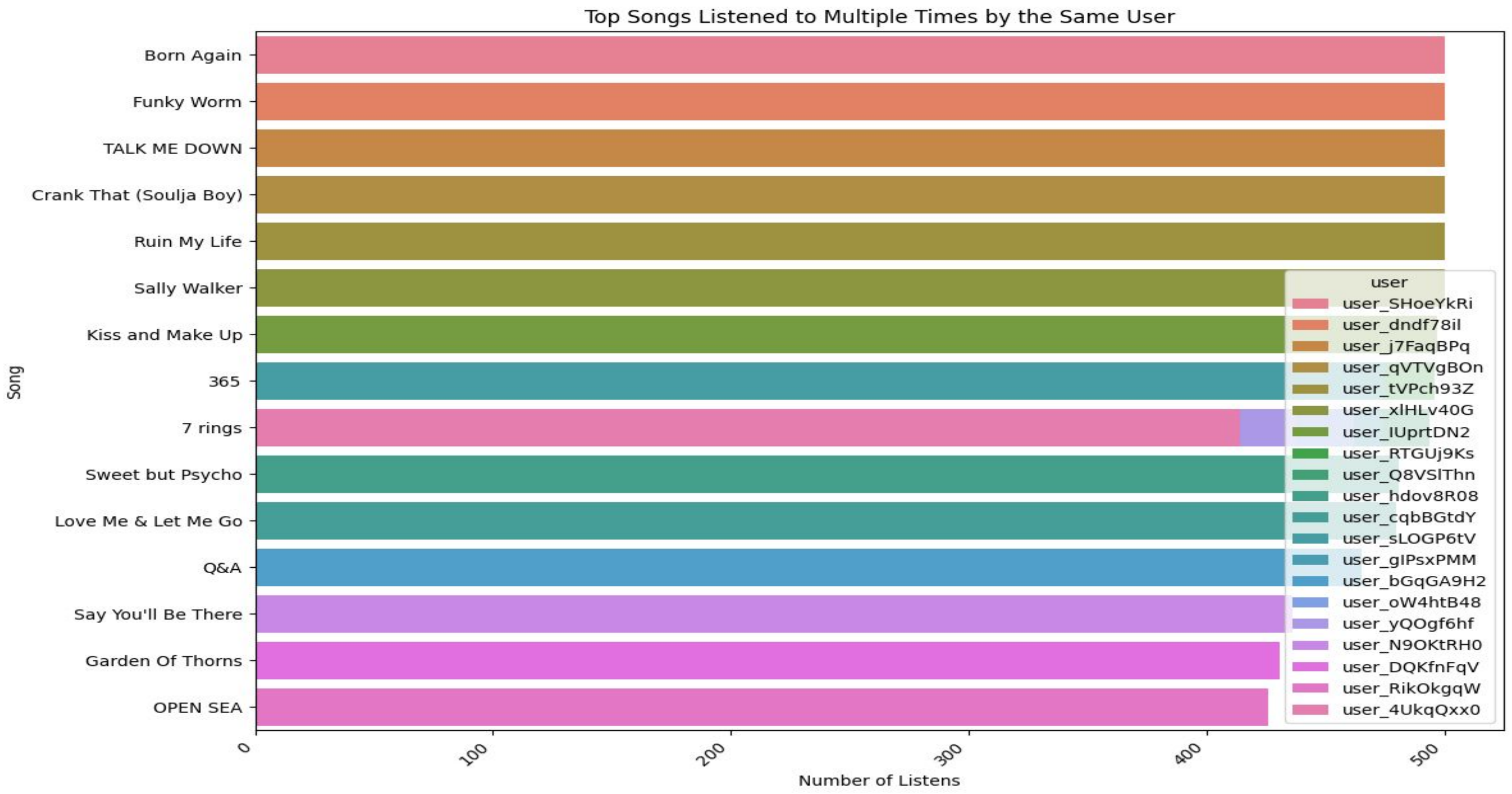
- Bar chart of genre distribution



Listening History

A song is considered **"liked"** if a user listens to it more than once. (2 times or more)

- Bar chart - Top songs listened multiple times by the same user.



Strengths of the Dataset(s)

Comprehensive Data:

- **Rich Metadata and User Interaction History**:

  a. **Impact on Content-Based Filtering:**

     - Detailed item profiles for accurate recommendations
     - Effectively captures user preferences

  b. **Impact on Collaborative Filtering:**

     - Enhanced user-item interactions leading to better similarity calculations
     - Improves the robustness of user-based recommendations

Strengths of the Dataset(s)

- **Diverse Genres and Tags**:
  a. **Impact on Content-Based Filtering:**
     - Captures diverse user tastes
     - Enables personalized recommendations

  b. **Impact on Collaborative Filtering:**
     - Expands user-item matrix
     - Enhances recommendation diversity.

Strengths of the Dataset(s)

Real-World Relevance

- **Reflects Actual User Behavior and Preferences**:
  a. **Impact on Content-Based Filtering:**
     - Aligns recommendations with real user behavior
     - Models genuine user interests accurately

  b. **Impact on Collaborative Filtering:**
     - Enhances user-user and item-item similarity relevance
     - Improves effectiveness of collaborative filtering models

Weaknesses of the Dataset(s)

Data Sparsity:

- **Uneven distribution of user interactions**
  a. **Impact on Content-Based Filtering:**
     - Limited data for some users
     - Cold start problem

  b. **Impact on Collaborative Filtering:**
     - Sparse user-item matrix
     - Low precision and recall

- **Potential gaps in user listening history**
  a. **Impact on Content-Based Filtering:**
     - Incomplete profiles

  b. **Impact on Collaborative Filtering:**
     - Inaccurate similarity calculations

Weaknesses of the Dataset(s)

Noise in Tags:

- **Variability in user-generated tags**
  a. **Impact on Content-Based Filtering:**
     - Inconsistent item profiles
     - Noise in recommendations
  b. **Impact on Collaborative Filtering:**
     - Indirect impact on hybrid models

- **Inconsistent Tagging**:
  a. **Impact on Content-Based Filtering**:
     - Ambiguous item profiles
  b. **Impact on Collaborative Filtering**:
     - Data quality issues in hybrid models

Dataset - Time Based Split

Purpose

- Predict future listening behavior.
- Simulate real-world usage with unseen data.

Advantages

1. **Realistic Simulation**
   - Mimics real-world scenarios.
2. **New Song Recommendations**
   - Suggests new songs, not just known ones.
3. **Avoid Overfitting**
   - Separates training and testing data by time.

Models & Methods

Exploration & Exploitation Model

Cold Start - Purely Exploration

Popularity-Based Recommendation

Why?

- Lack of User Data
- Simplicity and Effectiveness
- Baseline Performance

Method

- Recommends 30 popular songs as baseline

Implementation

- Data Collection
- Extracting Top Songs
- Matching Song Details
- Formatted Output

EXPLOITATION MODEL

- Content
- Collaborative
- Hybrid

Modeling-Content based filtering

why content based filtering is a good approach ?

- Personalized
- Help user discover the less popular songs.

Modeling-Content based filtering

Process of training models:

Data preprocessing → User profile → Similarity matrix → Recommendation

Modeling-Content based filtering

Process of training models: ['popularity', 'release', 'danceability', 'energy', 'key', 'mode', 'valence', 'tempo', 'duration_ms']

Data processing

Extracting and scaling numeric value for song

```
min_max_scaler = MinMaxScaler()
scaled_df=scalling_features(min_max_scaler)
```

Converting the other data into tf-idf

```
tags_TF=convert_to_TF(data_frames["id_tags"]["tags"])
genres_TF=convert_to_TF(data_frames["id_genres"]["genres"])
lang_TF=convert_to_TF(data_frames["id_lang"]["lang"])
```

Combining into a songs profiles

```
combined_df = pd.concat([scaled_df, tags_TF, genres_TF, lang_TF], axis=1)
```

Modeling-Content based filtering

Process of training models:

User profile

- User Interaction Filtering: filter out songs listened more than twice
- Combining into a Feature Weighting based on interaction counts
- User Profile Calculation

Modeling-Content based filtering

Process of training models:

Similarity matrix

- **Choosing cosine similarly as main similarity method.**
- **Generating matrix with with rows are users and columns are songs**
- **Combining into a songs profiles**

Modeling-Content based filtering (user-based)

Issues with Content based method?

– Known information might not perfectly characterize the items.

Modeling-Collaborative filtering

Listening_History

#	Column	Dtype
0	user	object
1	song	object
2	timestamp	object

	user	song	timestamp
0	user_007XIj0r	DaTQ53TUmfP93FSr	2019-02-20 12:28
1	user_007XIj0r	dGeyvi5WCOjDU7da	2019-02-20 12:35
2	user_007XIj0r	qUm54NYOjeFhmKYx	2019-02-20 12:48
3	user_007XIj0r	FtnuMT1DlevSR2n5	2019-02-20 12:52
4	user_007XIj0r	LHETTZcSZLeaVOGh	2019-02-20 13:09
5	user_007XIj0r	LHETTZcSZLeaVOGh	2019-02-20 13:13
6	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:33
7	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:39
8	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:45
9	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:51
10	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:58
11	user_007XIj0r	QDtAFWCvEaRwFPnC	2019-02-20 14:15
12	user_007XIj0r	MFVaxfkxZAUc4C8U	2019-02-20 14:20
13	user_007XIj0r	DhLocRB1CrB5ikDO	2019-02-
14	user_007XIj0r	zkJb2QRrowAMNqvr	2019-02-
15	user_007XIj0r	iMZ6ZUqtQ1UVGOoW	2019-02-
16	user_007XIj0r	lHdsYHWV2aB5qYvE	2019-02-
17	user_007XIj0r	D098jupnftqfLnvL	2019-02-
18	user_007XIj0r	dwpODe8SVpfU7u1W	2019-02-
19	user_007XIj0r	jH2ACbKWbK1A1GLB	2019-02-

	user	song	timestamp
5109572	user_zzWscYTy	62JzqGi7XyO8pvUZ	2019-01-10 12:14
5109573	user_zzWscYTy	mkasX8AtFbvUO2So	2019-01-10 12:29
5109574	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 12:55
5109575	user_zzWscYTy	h1Nu3eaJHofZvwU0	2019-01-10 12:59
5109576	user_zzWscYTy	oaRnInagCUa6Vr0y	2019-01-10 13:54
5109577	user_zzWscYTy	eonPnBqOTrrDaFrP	2019-01-10 14:17
5109578	user_zzWscYTy	1cUEwqcqB0tnw3YA	2019-01-10 14:21
5109579	user_zzWscYTy	z7gqPfuaCyCR9Myt	2019-01-10 14:42
5109580	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 14:46
5109581	user_zzWscYTy	mvUaP8k67qOFfA65	2019-01-10 14:53
5109582	user_zzWscYTy	xiTyf8gVM8tYnzZb	2019-01-10 14:56
5109583	user_zzWscYTy	MzdRliPkcseDAw18	2019-01-10 15:06
5109584	user_zzWscYTy	gbm4IbnitbCXQwCb	2019-01-10 15:24
5109585	user_zzWscYTy	13avOnDUqkbSzRyI	2019-01-10 15:36
5109586	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 15:48
5109587	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 15:57
5109588	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 16:21
5109589	user_zzWscYTy	m401iLh6fC43xjRy	2019-01-10 16:48
5109590	user_zzWscYTy	mvUaP8k67qOFfA65	2019-01-10 21:13
5109591	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 21:16

"Wisdom of crowds" ?

Approach 1. Similar users' taste

Approach 2. Extracting latent Factors (matrix decomposition)

Modeling-Collaborative filtering

Approach 1. Similar users' taste

1. **listening history** → 2. **Latent Ratings** → 3. **User similarity matrix** → 4. **Recommendation**

	user	song	timestamp
0	user_007XIj0r	DaTQ53TUmfP93FSr	2019-02-20 12:28
1	user_007XIj0r	dGeyvi5WCOjDU7da	2019-02-20 12:35
2	user_007XIj0r	qUm54NYOjeFhmKYx	2019-02-20 12:48
3	user_007XIj0r	FtnuMT1DlevSR2n5	2019-02-20 12:52
4	user_007XIj0r	LHETTZcSZLeaVOGh	2019-02-20 13:09
5	user_007XIj0r	LHETTZcSZLeaVOGh	2019-02-20 13:13
6	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:33
7	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:39
8	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:45
9	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:51
10	user_007XIj0r	zXMHUt57MEYgpgIz	2019-02-20 13:58
11	user_007XIj0r	QDtAFWCvEaRwFPnC	2019-02-20 14:15
12	user_007XIj0r	MFVaxfkxZAUc4C8U	2019-02-20 14:20
13	user_007XIj0r	DhLocRBlCrB5ikDO	2019-02-
14	user_007XIj0r	zkJb2QRrowAMNqvr	2019-02-
15	user_007XIj0r	iMZ6ZUqtQ1UVGOoW	2019-02-
16	user_007XIj0r	1HdsYHWV2aB5qYvE	2019-02-
17	user_007XIj0r	D098jupnftqfLnvL	2019-02-
18	user_007XIj0r	dwpODe8SVpfU7u1W	2019-02-
19	user_007XIj0r	jH2ACbKWbK1A1GLB	2019-02-

	user	song	timestamp
5109572	user_zzWscYTy	62JzqGi7XyO8pvUZ	2019-01-10 12:14
5109573	user_zzWscYTy	mkasX8AtFbvU02So	2019-01-10 12:29
5109574	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 12:55
5109575	user_zzWscYTy	h1Nu3eaJHofZvwU0	2019-01-10 12:59
5109576	user_zzWscYTy	oaRnInagCUa6Vr0y	2019-01-10 13:54
5109577	user_zzWscYTy	eonPnBqOTrrDaFrP	2019-01-10 14:17
5109578	user_zzWscYTy	1cUEwqcqB0tnw3YA	2019-01-10 14:21
5109579	user_zzWscYTy	z7gqPfuaCyCR9Myt	2019-01-10 14:42
5109580	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 14:46
5109581	user_zzWscYTy	mvUaP8k67qOFfA65	2019-01-10 14:53
5109582	user_zzWscYTy	xiTyf8gVM8tYnzZb	2019-01-10 14:56
5109583	user_zzWscYTy	MzdRliPkcseDAw18	2019-01-10 15:06
5109584	user_zzWscYTy	gbm4IbnitbCXQwCb	2019-01-10 15:24
5109585	user_zzWscYTy	13avOnDUqkbSzRyI	2019-01-10 15:36
5109586	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 15:48
5109587	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 15:57
5109588	user_zzWscYTy	5ZHgff3sjETIiedr	2019-01-10 16:21
5109589	user_zzWscYTy	m401iLh6fC43xjRy	2019-01-10 16:48
5109590	user_zzWscYTy	mvUaP8k67qOFfA65	2019-01-10 21:13
5109591	user_zzWscYTy	BBiswLufo26YQCT7	2019-01-10 21:16

	user	song	interactions	rating
0	user_007XIj0r	019Q5nZ079pL34qa	1	1.000000
1	user_007XIj0r	0247XD85scrr19o0	2	1.293631
2	user_007XIj0r	04iQE4VyxOcloEVY	4	1.663563
3	user_007XIj0r	0K3SrhMknjp3HCQg	3	1.501966
4	user_007XIj0r	0M1ouT1rCMQlu1Ps	8	2.089229

	user_1Pv2iFMy	user_2CsnB2wx	user_2Wntb3va	user_2yPi90hx	\
user_1Pv2iFMy	1.0	0.0	0.00000	0.000000	
user_2CsnB2wx	0.0	1.0	0.00000	0.000000	
user_2Wntb3va	0.0	0.0	1.00000	0.504650	
user_2yPi90hx	0.0	0.0	0.50465	1.000000	
user_31VIgRVv	0.0	0.0	0.08189	0.083473	

	user_31VIgRVv	user_45ezbtzI	user_5VV5WBxF	user_81YWI81X	\
user_1Pv2iFMy	0.000000	0.0	0.0	0.0	
user_2CsnB2wx	0.000000	0.0	0.0	0.0	
user_2Wntb3va	0.081890	0.0	0.0	0.0	
user_2yPi90hx	0.083473	0.0	0.0	0.0	
user_31VIgRVv	1.000000	0.0	0.0	0.0	

Modeling-Collaborative filtering (user-based)

Approach 1. Similar users' taste

Recommendation

1

#Predict unknown songs' ratings by similar users' ratings.

2

#Predict for all unknown songs and recommend top N items.

```
def get_predicted_rating(song, username, song_index, ratings_matrix, user_list, user_similarity_df, n=3):

    song_idx = song_index[song]
    user_interacted_indices =  ratings_matrix[:, song_idx].nonzero()[0]
    user_interacted = [user_list[idx] for idx in user_interacted_indices]
    similarities = user_similarity_df.loc[username, user_interacted]

    # Sort similarities
    sorted_indices = similarities.sort_values(ascending=False).index

    # Extracting the most similar N users excetpt the user itself
    top_n_user_ids = sorted_indices[sorted_indices != username][:n]
    score, rating_sum = 0, 0
    for user in top_n_user_ids:
        similarity = user_similarity_df.loc[username, user]
        rating = ratings_matrix[user_index[ user], song_idx]
        score += similarity * rating
        rating_sum +=rating
    if rating_sum == 0:
        return 0
    else:
        score = score/rating_sum
        return score
```

```
# Gather songs listened to by similar users but not by the specified user
user_idx = user_index[username]

user_songs = set(traindata[traindata['user'] == username]['song'])
all_songs = set( test_song_list)
unknown_songs = all_songs - user_songs


scores = []
songs_to_recommend = set()
for song in unknown_songs:
    scores.append((song, get_predicted_rating (song, username, song_index, ratings_matrix, user_list, user_similarity_d

scores.sort(key=lambda x: x[1], reverse=True)
top_n_songs_with_ratings = scores[:num_recommendations]
top_n_song_names = [song for song, rating in top_n_songs_with_ratings]
```

Modeling-Collaborative filtering (user-based)

Approach 1. Similar users' taste

Listening history 1 → Latent Ratings 2 → User similarity matrix 3 → Recommendation 4

	user	song	interactions	rating
0	user_007XIj0r	019Q5nZ079pL34qa	1	1.000000
1	user_007XIj0r	0247XD85scrr19o0	2	1.293631
2	user_007XIj0r	04iQE4VyxOcloEVY	4	1.663563
3	user_007XIj0r	0K3SrhMknjp3HCQg	3	1.501966
4	user_007XIj0r	0M1ouT1rCMQ1u1Ps	8	2.089229

	user_1Pv2iFMy	user_2CsnB2wx	user_2Wntb3va	user_2yPi90hx	\
user_1Pv2iFMy	1.0	0.0	0.00000	0.000000	
user_2CsnB2wx	0.0	1.0	0.00000	0.000000	
user_2Wntb3va	0.0	0.0	1.00000	0.504650	
user_2yPi90hx	0.0	0.0	0.50465	1.000000	
user_31VIgRVv	0.0	0.0	0.08189	0.083473	

	user_31VIgRVv	user_45ezbtzI	user_5VV5WBxF	user_81YWI81X	\
user_1Pv2iFMy	0.000000	0.0	0.0	0.0	
user_2CsnB2wx	0.000000	0.0	0.0	0.0	
user_2Wntb3va	0.081890	0.0	0.0	0.0	
user_2yPi90hx	0.083473	0.0	0.0	0.0	
user_31VIgRVv	1.000000	0.0	0.0	0.0	

Issues:

Interaction data is sparse

Loops through songs and users

Runs non-linearly

Modeling-Collaborative filtering (user-based)

Approach 2. Extracting latent Factors (SVD decomposition)

Listening history

1

User-item matrix

= $U \cdot \Sigma \cdot V^T$

2

U: Each row represents a user, and the columns correspond to the latent factors.

Σ: Diagonal matrix containing singular values, which represent the importance or weight of each latent feature.

V^T: Each row corresponds to an item and the columns correspond to the latent factors

$u \cdot \Sigma$:

Latent user factors for user u

$u \cdot \Sigma \cdot V^T$:

Scores for all items for user u

Modeling-Collaborative filtering (user-based)

Approach 2. Extracting latent Factors (SVD decomposition)

1

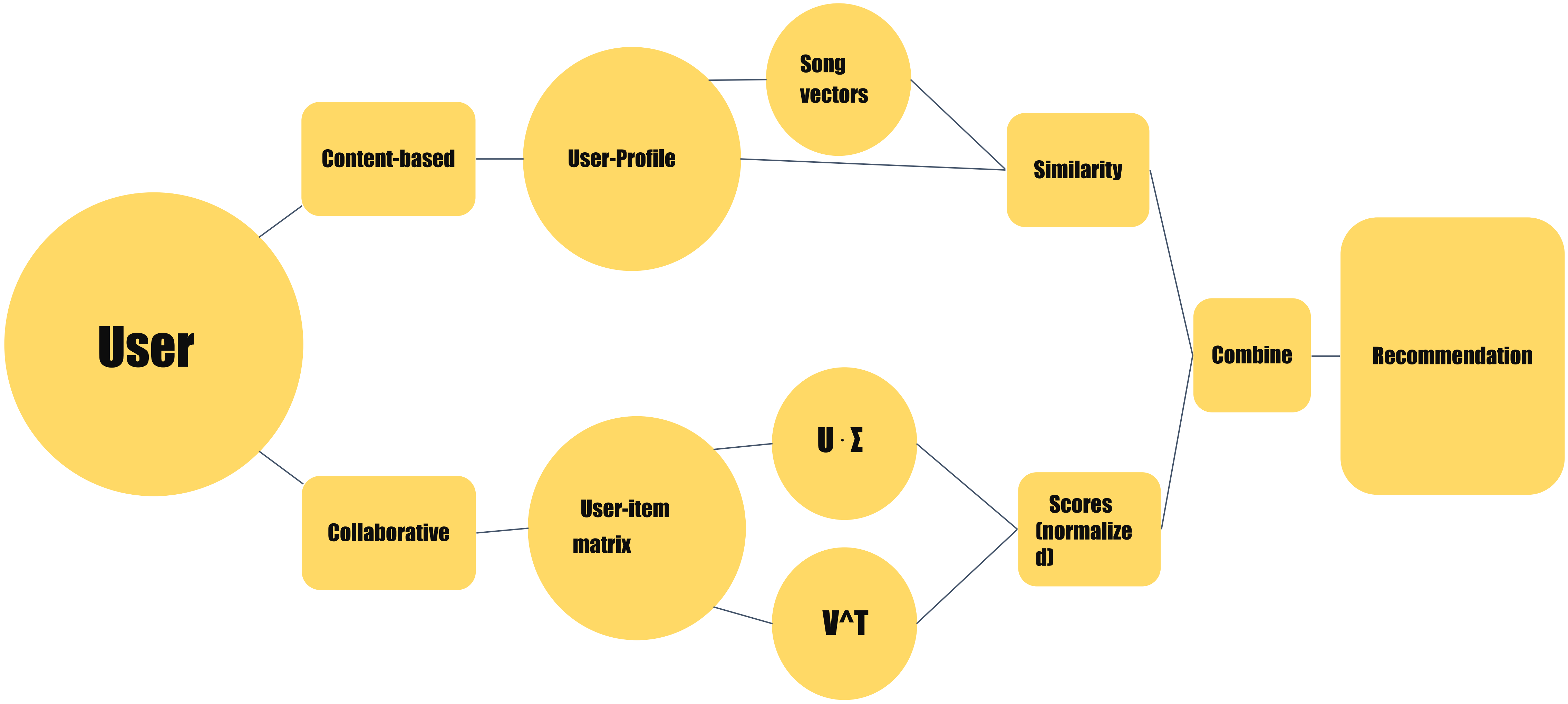
All “ratings” are calculated by one matrix multiplication.

Recommendation

```
def svd_recommendations(user_id, user_features, item_features, N=30):
    if user_id not in user_to_index:
        return []
    user_idx = user_to_index[user_id]
    user_vec = user_features[user_idx, :]
    scores = user_vec.dot(item_features.T)
    top_items = scores.argsort()[::-1]
    return [(song_ids[idx], scores[idx]) for idx in top_items if idx < len(song_ids)][:N]
```

Fast (Linear time complexity) and Precise

Modeling-Hybrid Model



Evaluations & Results

Evaluation & Results

- **Relevant Items**

	user	song	timestamp	count
0	user_007XIjOr	DaTQ53TUmfP93FSr	2019-02-20 12:28	1
1	user_007XIjOr	dGeyvi5WCOjDU7da	2019-02-20 12:35	1
2	user_007XIjOr	qUm54NYOjeFhmKYx	2019-02-20 12:48	1
3	user_007XIjOr	FtnuMT1DlevSR2n5	2019-02-20 12:52	1
4	user_007XIjOr	LHETTZcSZLeaVOGh	2019-02-20 13:09	1
5	user_007XIjOr	LHETTZcSZLeaVOGh	2019-02-20 13:13	1
6	user_007XIjOr	zXMHUt57MEYgpglz	2019-02-20 13:33	1
7	user_007XIjOr	zXMHUt57MEYgpglz	2019-02-20 13:39	1
8	user_007XIjOr	zXMHUt57MEYgpglz	2019-02-20 13:45	1

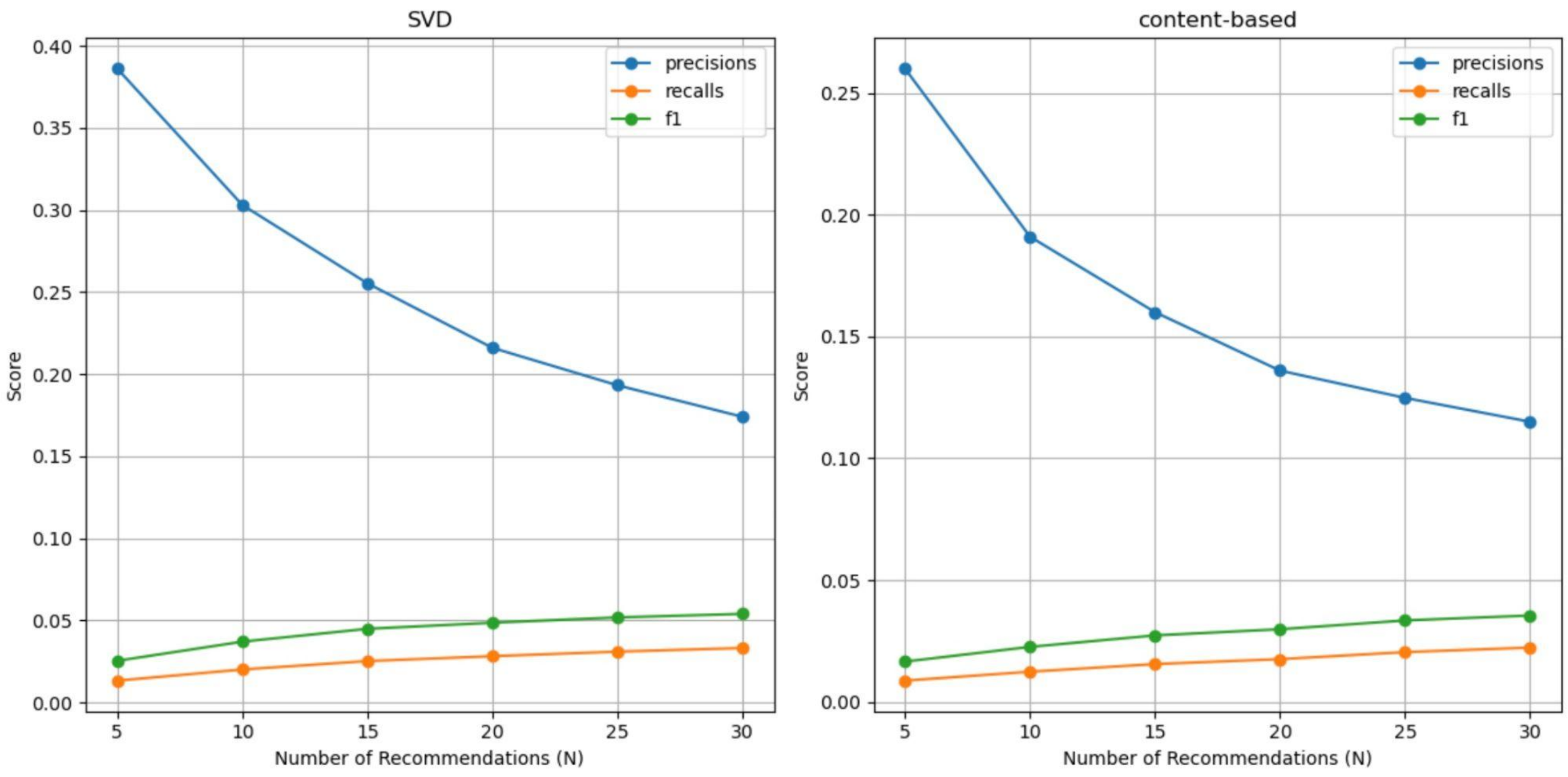
'Relevant items' - Songs user has interacted with ***(listened)*** more than once. (2 times or more)

- **Metrics**

Precisions : The accuracy of the recommendation
Recalls: The ability of retrieving relevant items
F1 scores: The balance between Precisions and Recalls

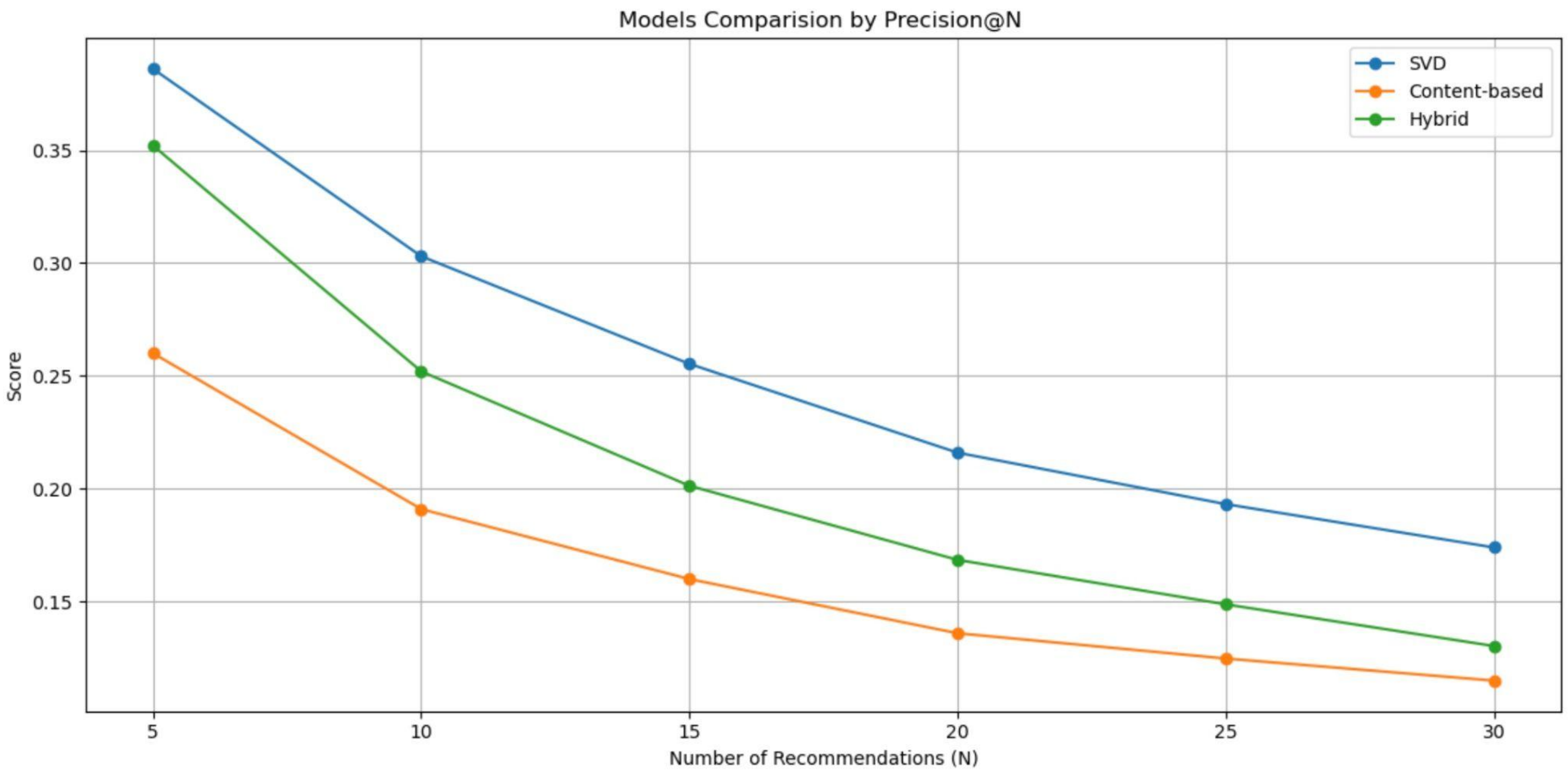
Evaluation & Results

We test the average scores for 100 users and different N values (Precision@N, Recall@N, F1@N):



1.The precision goes down as N increases
2.Recalls and F1's are increasing.
3.SVD: (0.17 - 0.35) content-based:(0.09 - 0.175).

Evaluation & Results



SVD outperforms the hybrid model .

Evaluation & Results

1.Both models have higher precisions when N is small

2.SVD performs better than content-based method.

3.SVD outperforms the hybrid model .

Limitations and Future Improvements

Limitations

- Scalability
- Recommendations based on Artists
- Contextual Awareness

Future work

- Adjusting weights of Exploration and Exploitation Models
- Real-time update
- Contextual Aware Recommendation
- Social and Collaborative Features

THANKS