Team Stuxnet Project Final Report
Million Playlist Dataset Challenge

Mudit Soni — 2017EE10463
Yash Malviya — 2016CS50403
Shivani Choudhary — 2020SRZ8250

1 Project description

The spotify million playlist dataset (MPD) challenge is a continuation of the 2018 ACM Recommender Systems Challenge. The dataset consists of one million spotify user-generated playlists and the task is automatic playlist continuation (APC), a form of more general sequential recommendation.

Given a playlist containing some metadata and seed tracks (0, 1, 5, 10, 25, or 100), the task is to recommend upto 500 candidate tracks ordered by relevance in decreasing order, thereby *continuing* the playlist.

The challenge had two distinct tracks -

1. **Main Track**: Only the provided dataset (MPD) can be used for training of models.
2. **Creative Track**: Using external resources (along with MPD), such as public datasets, for solving the same task.

To asses the quality of playlist continuation three metrics average across all the playlist in challenge dataset was used: R-precision, normalized discounted cumulative gain (NDCG), and recommended songs clicks.

2 Technical Outline

Under this project, we have tried to solve the challenge using Collaborative filtering and Content based filtering approach. There are different tracks in the challenge. Out of which most difficult one was the case where no seed track was given, called as cold start problem. Below is the details of the different approaches. There were limited information was available for the tracks in MPD dataset. We have augmented the dataset with Spotify API.

2.1 Preprocessing

For data augmentation, we have used spotify api [4] for python to get additional track information. The audio features we added are:

1. **Acousticness**: A confidence measure in range $[0, 1.0]$ whether the track is acoustic.
2. **Danceability**: A value from 0 to 1.0 giving how danceable the track is.
3. **Energy**: Measure in range $[0, 1.0]$ based on loudness, speed and noisiness of the audio.
4. **Instrumentalness**: Measure in range $[0, 1.0]$ giving whether the track has no vocals.
5. **Liveness**: Detects presence of audience in the audio.
6. **Speechiness**: Detects presence of spoken words. Speechiness above 0.66 means the audio is entirely made of spoken words, 0.33 to 0.6 means audio contains both speech and music and below 0.33 implies little to no speech.
7. **Valence**: Value in range $[0, 1.0]$ giving positiveness in the track.
8. **Tempo**: The overall tempo in beats per minute.

A playlist-track matrix (PTM) sparse matrix is obtained for further applications. We have used sparse matrices for storing intermediate results.

2.2 Algorithms

Three different algorithms have been used for obtaining recommendations. A weighted interpolation over their scores gives the final track scores which are used for playlist continuation.

2.2.1 Collaborative filtering: Track Based

We have applied item-item collaborative filtering over the PTM, where items represent tracks. First BM25 normalization is applied to PTM which has rows and tracks as columns.

Similarity between two tracks i and j is defined as:

$$s_{ij} = r_i * r_j$$

Score for track i for playlist u is calculated as:

$$r_{ui} = \sum_{j \in I(u)}^{KNN} r_{uj} * (s_{ij})^p$$

2.2.2 Content Based Filtering: Track Based

First BM25 normalization is applied to Track Content Matrix. The similarity between two tracks is defined as:

$$s_{ij} = f_i * f_j$$

where f are feature vectors formed by metadata information apart from PTM.

Score for track i for playlist u is calculated as:

$$r_{ui} = \sum_{j \in I(u)}^{KNN} r_{uj} * (s_{ij})^p$$

3 Implementation Details

Content based filtering data in our approach primarily relies on the track to track similarity and playlist of track mapping. Track similarity is based on the Album ID, Artist ID and augmented features. There can be varied degree of correlation among the augmented features. In order to ascertain that we have applied PCA and selected the n-Principal components that can explain at least 80% of the variance for the data. The matrix in relation to AlbumID and ArtistID are BM25 normalized. To make a single matrix with all the feature, vectors corresponding to each feature and Principal components are horizontally stacked.

Album vec	Aritst Vec	PCA vec

Table 1: Feature vector of Track

The similarity score between two tracks was calculated using Cosine similarity. Similarity amongst tracks are calculated with the vector shown above. The track-track similarity matrix acts as a basis for the seed tracks in the playlist to calculate the similarity and extract the K-Nearest neighbours (KNN) that will be the sample retrieval for a playlist. Playlist are mapped with the set of tracks present in the playlist from the MPD dataset. Playlist based features are of prime importance in case of the cold start problem where no seed tracks are present. In this scenario, we have applied NLP based approach on the title of the playlist along with two diversity score for album and artist. These scores are created using:

$$Albumdiversity = \log_2 \frac{NumberofAlbum}{Numberofsongs} \tag{1}$$

$$Artistdiversity = \log_2 \frac{NumberofArtist}{Numberofsongs} \tag{2}$$

On a careful analysis of these two features. One of the Principal component(PC) has address 99% of the variance. This lead to the substitution of these two features with the PC. Under the

NLP approach, we mapped the each of the playlist name in token space that was generated by stemming the name. The PC feature was used to select the top-K playlist out of the suggestion pool. Suggestion pool is generated using partial name matching using edit distance or full name matching.

Difference combinations has generated following models:

- Pure content based
- NLP approximate matching
- NLP absolute matching

4 Resources Used

We have used HPC to run our retrieval system. Mostly we have used 64gb ram with 2-4 cpu cores.

5 Datasets

We are using the *Million Playlist Dataset (MPD)* same as the one used in 2018 ACM Recommender Systems Challenge. A training dataset with 1 million user created playlists (complete) is provided. These playlists were created between January 2010 and October 2017. For each playlist, its title, list of tracks, album, artist names, and some additional metadata such as Spotify URIs and the playlist's number of followers is included in the dataset. Example of such playlist entry would be -

```
{
    "name": "musical",
    "collaborative": "false",
    "pid": 5,
    "modified_at": 1493424000,
    "num_albums": 7,
    "num_tracks": 12,
    "num_followers": 1,
    "num_edits": 2,
    "duration_ms": 2657366,
    "num_artists": 6,
    "tracks": [
        {
            "pos": 0,
            "artist_name": "Degiheugi",
            "track_uri": "spotify:track:7vqa3sDmtEaVJ2gcvxtRID",
            "artist_uri": "spotify:artist:3V2paBXEoZIAhfZRJmo2jL",
            "track_name": "Finalement",
            "album_uri": "spotify:album:2KrRMJ9z7Xjoz1Az4O6UML",
            "duration_ms": 166264,
            "album_name": "Dancing Chords and Fireflies"
        },
        // 11 tracks omitted
    ],
}
```

A separate challenge dataset is there to evaluate the quality of algorithms. This challenge dataset was composed of 10 thousand incomplete playlists, these playlist were of these type of 10 scenarios. Out of these, we have presented results of the following 7 selected scenarios -

1. Title only no tracks
2. Title and first track
3. Title and first 5 tracks
4. Title and first 10 tracks
5. Title and random track

6. Title and 5 random tracks
7. Title and 10 random tracks

6 Metrics

6.1 R-precision

Defined as weighted sum of track-level R-precision and artist level R-precision.

$$Rprec = Rprec_t + 0.25 * Rprec_a$$

where

$$Rprec_t = \frac{G_t \cap R_{t_{1:|G_t|}}}{G_t}$$

and

$$Rprec_a = \frac{G_a \cap R_a}{G_a}$$

6.2 Normalized Discounted Cumulative Gain

This is the well know metric he have studied in class

6.3 Recommended songs clicks

$$clicks = \frac{argmin_i\{R_{t_i} : R_{t_i} \in G_t\} - 1}{10}$$

where R_{t_i} is the track that occupies the i^{th} index in the ordered list of recommended tracks R_t

7 Results

The evaluation was done on 3 metrics R-precision, nDCG and recommended song clicks on a test dataset prepared using last 5000 playlists from Million Playlist Dataset. The test dataset consists of 1000 samples of each of the 10 scenarios mentioned in Datasets section. The final score is given as average of all three metrics over 10,000 samples in the test dataset.

For Content based filtering have run our system with test and train data separation in the ratio of 0.1. Randomly 10% data was selected as test set. As per the challenge statement, the MPD set contains all the tracks even those are mentioned in the challenge set. To emulate the same behaviour the track to tracks similarity matrix has pairwise similarity score for all the track. Test set playlist information is missing from the playlist CSV to ensure a proper separation between test and train sets. Test playlist was further separated into different tracks based on the number of seed tracks.

- Playlist with 1st track
- Playlist with first 5 tracks
- Playlist with first 10 tracks
- Playlist with 1 random track
- Playlist with 5 random tracks
- Playlist with 10 random tracks
- Playlist with no seed track

Judgment for each of the tracks are comprised of rest of the track from the playlist in the same order as in the playlist. For this we have selected playlist with more than 30 tracks. The nDCG,R-precision scores for the iterations are

7.1 Collaborative Filtering Model

The results of collaborative filtering only model are given in table 2. Here we have only provided average scores of scenarios with at least one track as collaborative filtering cannot work otherwise.

Metric(Average)	**Score**
R-Precision	0.044
nDCG	0.072
Song Clicks	2.199

Table 2: Scores for Collaborative Filtering Model

7.2 Content-based Filtering Model

The results of content-based model are given in table 3

Scenario	**nDCG Score**	**R-precision Score**
First Track	0.136036	0.073826
First 5 Tracks	0.268092	0.140807
First 10 Tracks	0.365528	0.220056
First 10 Tracks	0.365528	0.220056
Random Track	0.139257	0.072197
5 Random Tracks	0.318004	0.183987
10 Random Tracks	0.444056	0.291902
Title only (Partial)	0.0159	0.004427
Title only (Full)	0.001047	0.000329
Average Score	0.21099	0.123441

Table 3: Scores for Content-based Filtering Model across different scenarios

7.3 State-of-the-art Results

The results of state-of-the-art model by team vl6 [3] in the ACM RecSys Challenge 2018 are given in the table 4. Comparison and conclusion are given in the following Conclusion section.

Scenario	**nDCG Score**	**R-precision Score**	**Song Clicks Score**
First Track	0.3065	0.1497	3.527
First 5 Tracks	0.3773	0.2032	0.889
First 10 Tracks	0.3978	0.2094	0.439
Title only (Full)	0.2044	0.0979	10.746
Combined Score	0.2241	0.3946	1.7839

Table 4: Scores for State-of-art Model across different scenarios

8 Conclusion

In this project we have created two different types of recommender models namely, content-based filtering and collaborative filtering models. We observed that the average scores for our content-based model were better than that for collaborative model on all three metrics. Further comparison with state-of-the-art model revealed that our scores were closer to state-of-art and even surpassed them in some cases as the number of available tracks increased. Here, we have separately developed and presented both models, but in future they could be combined to produce better results. Further, modern machine learning techniques could be added to further enhance the performance.

References

[1] Challenge Description:
`https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge`

[2] Team Creamy Fireflies approach: `https://dl.acm.org/doi/10.1145/3267471.3267475`

[3] Best performing approaches and analysis of the RecSys 2018 challenge:
`https://dl.acm.org/doi/abs/10.1145/3344257`

[4] Spotify API: `https://developer.spotify.com/documentation/web-api`