# Playlist Recommendation Engine

Team: Aydin Ozbek, Dane Miyata, Kashish Mehta, Kristina Knowles, Mario Gomez

Fall 2023
Erdős Institute Data Science Bootcamp

GitHub
Executive Summary

# Music for...

Mood Regulation

Functions

Soundtrack for activities

Goal: Build a platform-agnostic playlist recommendation engine that takes in a user prompt and generates a novel corresponding playlist

# Data: Million Playlist Dataset (Spotify)


```
{
  "name": "christmas",
  "collaborative": "false",
  "pid": 96,
  "modified_at": 1480464000,
  "num_tracks": 19,
  "num_albums": 12,
  "num_followers": 1,
  "tracks": [
      {
          "pos": 0,
          "artist_name": "Mariah Carey",
          "track_uri": "spotify:track:0bYg9bo50gSsH3LtXe2SQn",
          "artist_uri": "spotify:artist:4iHNK0tOyZPYnBU7nGAgpQ",
          "track_name": "All I Want for Christmas Is You",
          "album_uri": "spotify:album:61ulfFSmmxMhc2wCdmdMkN",
          "duration_ms": 241106,
          "album_name": "Merry Christmas"
      },
      {
          "pos": 1,
          "artist_name": "Michael Bubl\u00e9",
          "track_uri": "spotify:track:6Lh85Pe18pqvNdDoWgTVUC",
          "artist_uri": "spotify:artist:1GxkXlMwML1oSg5eLPiAz3",
          "track_name": "Blue Christmas",
          "album_uri": "spotify:album:3CKVXhODttZebJAzjUs2un",
          "duration_ms": 221706,
          "album_name": "Christmas"
      },
      {
          "pos": 2,
          "artist_name": "The Cheetah Girls",
          "track_uri": "spotify:track:3A8Do36GezqgUVCd7f6GOR",
          "artist_uri": "spotify:artist:4ntkql3f3ect7NDRUJ7aAY",
          "track_name": "Cheetah-licious Christmas",
          "album_uri": "spotify:album:2dCaJ5ten3q3AbJ2VJZ8MY",
          "duration_ms": 211733,
          "album_name": "The Cheetah Girls: A Cheetah-licious Christmas"
      },
      •••
      {
          "pos": 17,
          "artist_name": "Michael Bubl\u00e9",
          "track_uri": "spotify:track:3pRTjSYlsfL3hqG43VolZl",
          "artist_uri": "spotify:artist:1GxkXlMwML1oSg5eLPiAz3",
          "track_name": "Silver Bells (feat. Naturally 7) - Bonus Track",
          "album_uri": "spotify:album:1l4DfcI1lGAmnDGDIWgiNA",
          "duration_ms": 187893,
          "album_name": "Christmas"
      },
      {
          "pos": 18,
          "artist_name": "Pentatonix",
          "track_uri": "spotify:track:2U9kDk5mlHYunC7PvbZ8KX",
          "artist_uri": "spotify:artist:26AHtbjWKiwYzsoGoUZq53",
          "track_name": "That's Christmas to Me",
          "album_uri": "spotify:album:082VlX7cBth0o8xqDGclNn",
          "duration_ms": 182173,
          "album_name": "That's Christmas To Me (Deluxe Edition)"
      }
  ],
  "num_edits": 5,
  "duration_ms": 3679250,
  "num_artists": 10
}
```


sample playlist structure

1 million playlists

2.26 million tracks

1 thousand json files

33 GB

https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge

Original Dataset

33 GB

SQL Database

1.6 GB

# Database Architecture

**PLAYLISTS**

PID
Name
Number of Followers
Number of Albums
Number of Artists
Number of Tracks
Number of Edits
Collaborative
Modified At
Duration

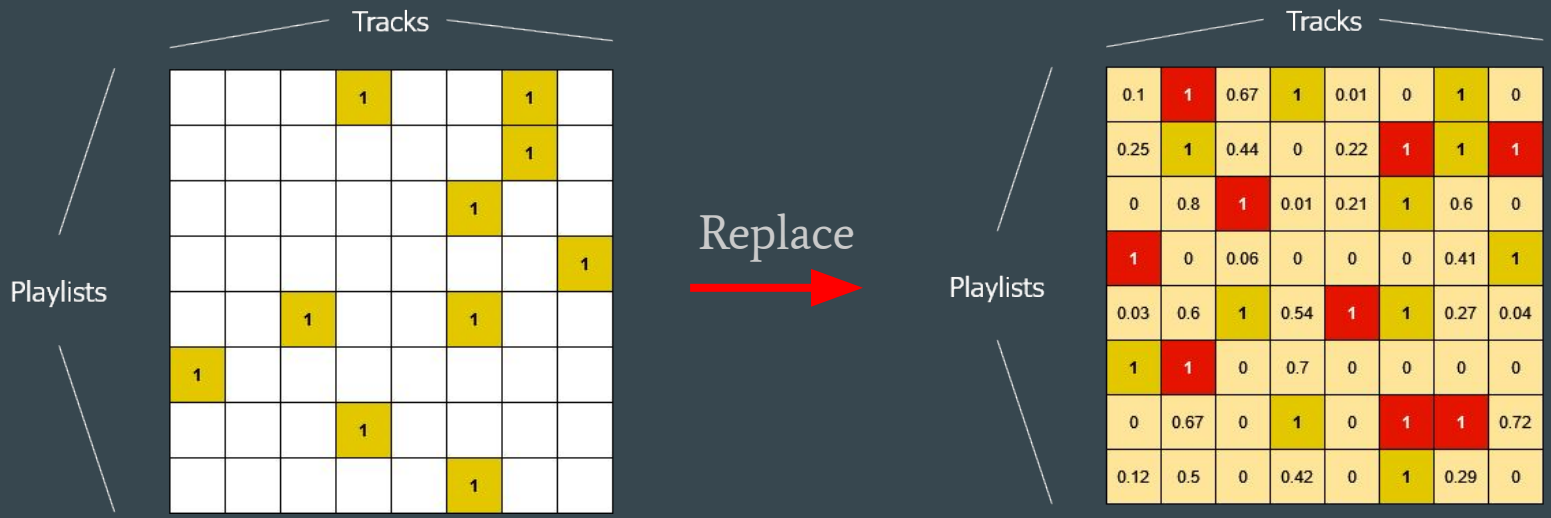
**PAIRINGS**

PID
TID
POS

**TRACKS**

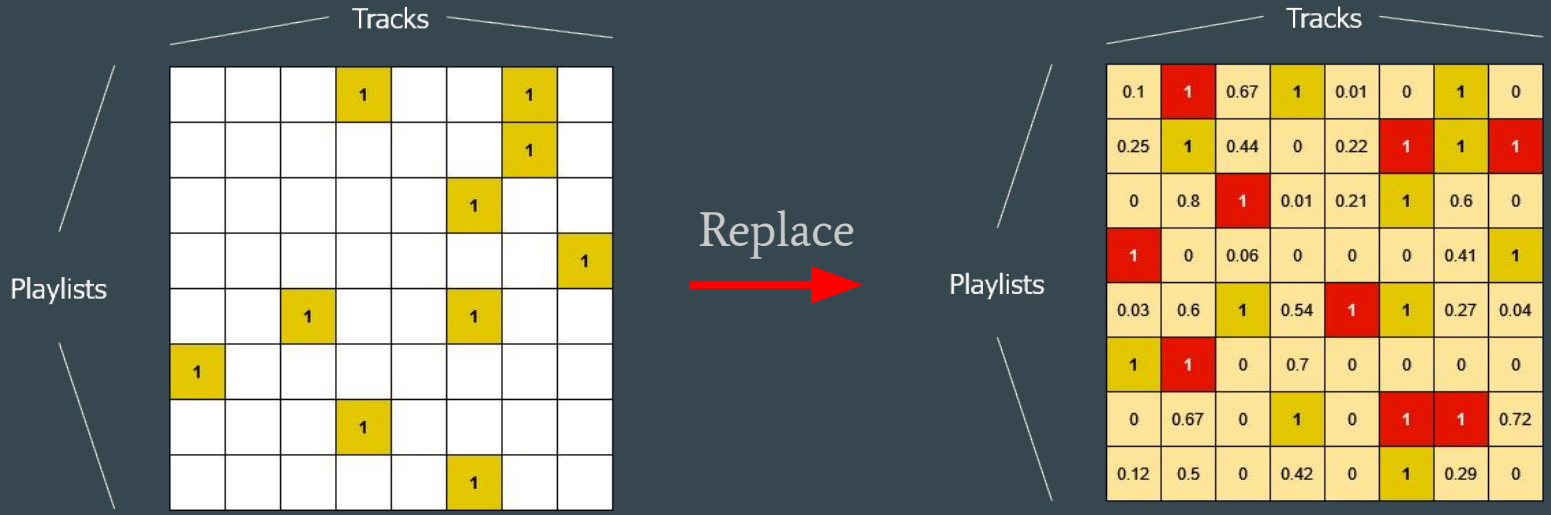
TID
Track Name
Artist Name
Album Name
Track URI
Album URI
Artist URI
Duration

# Matrix Factorization: FunkSVD



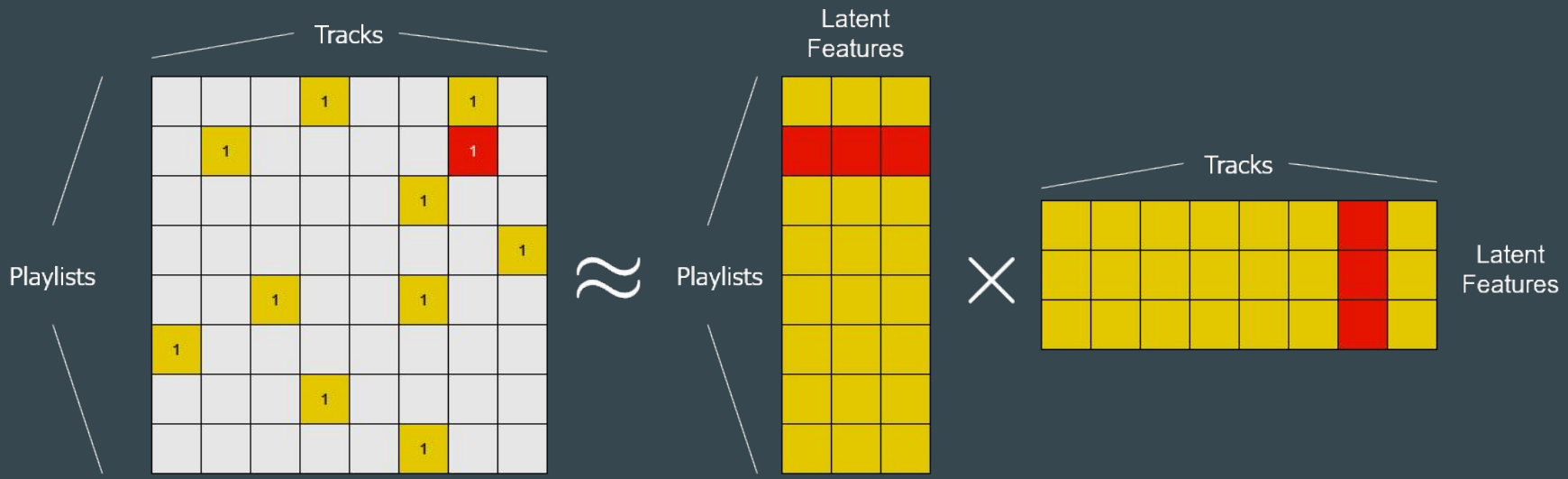

- 1 = track is in playlist
- 0 = track not in playlist

# Matrix Factorization: FunkSVD

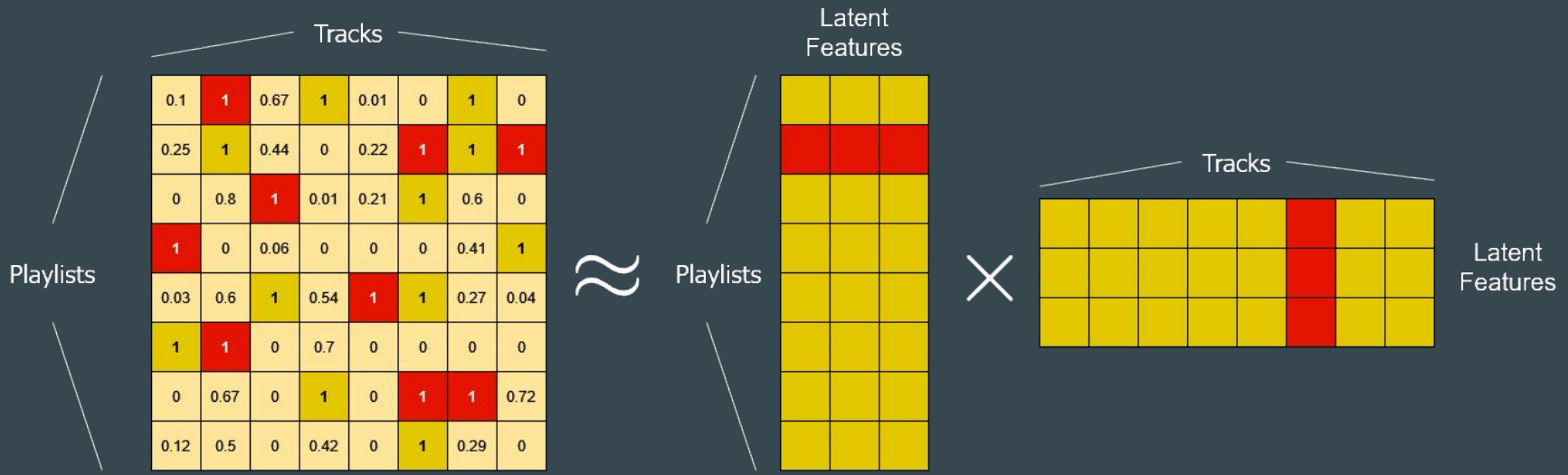



- 1 = track is in playlist
  ~~0 = track not in playlist~~
- 0 = missing value

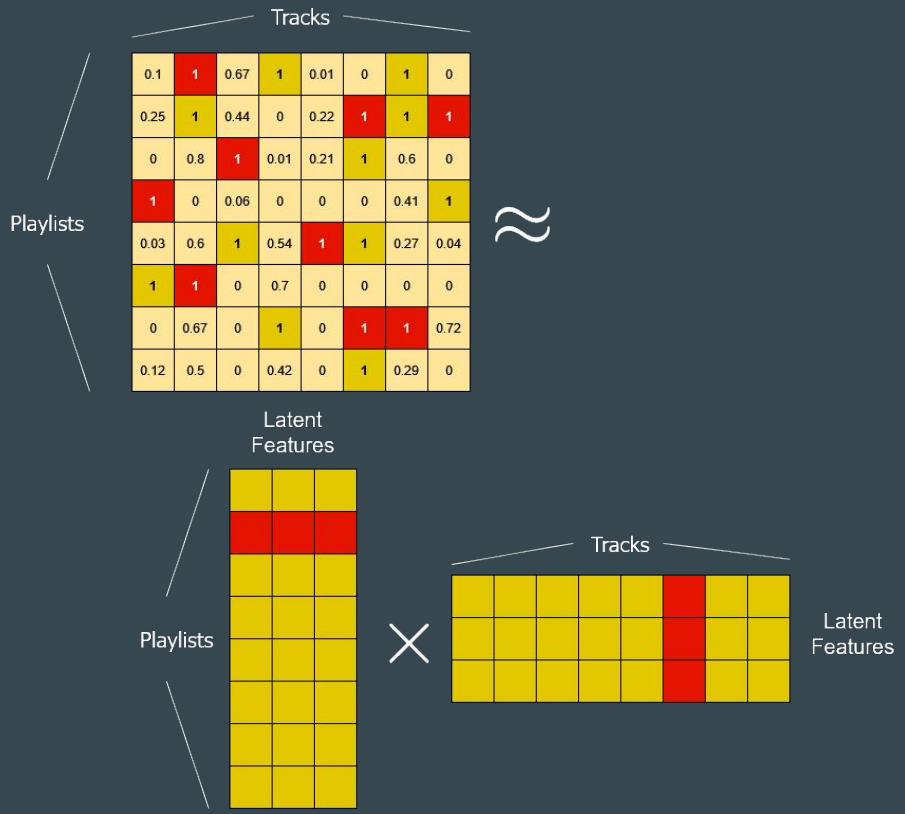
# Matrix Factorization: FunkSVD



# Matrix Factorization: FunkSVD



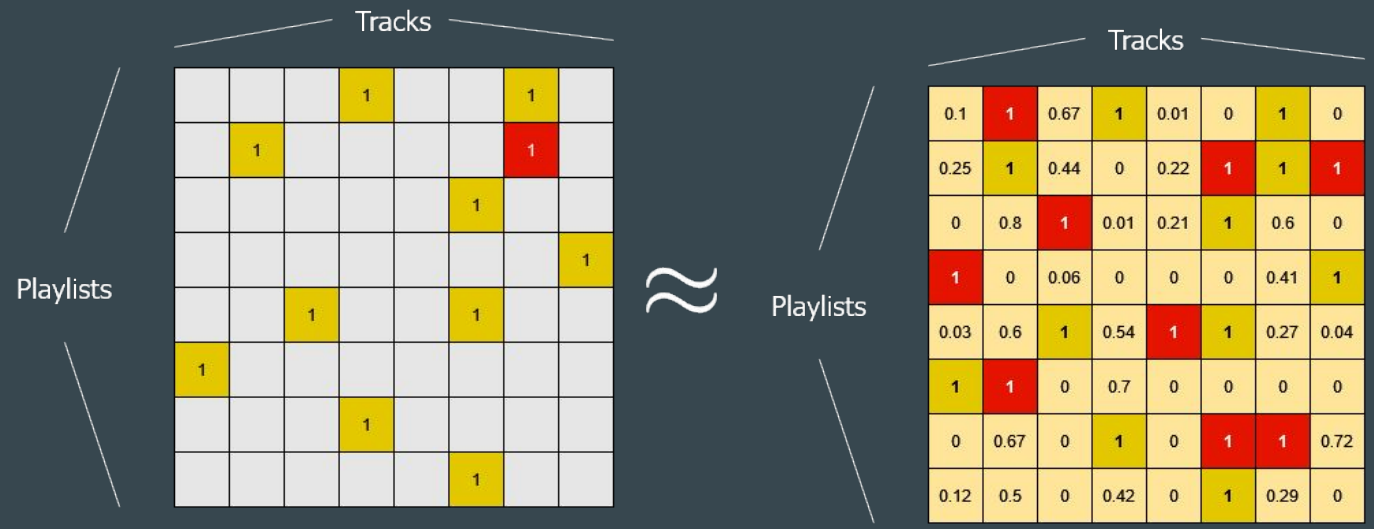
# Hyper-Parameters and Training




- Used gradient descent to minimize:
  - MSE (restricted to 1 entries of original matrix)
  - Plus L2 regularization

- Used a 70/15/15 train/val/test split to choose:
  - Number of latent features
  - L2 regularization factor
  - Step size in gradient descent

# Matrix Factorization: FunkSVD




MSE_val = 0.03603
MSE_test = 0.03512

**We recover 1s up to an error of 0.035**

# Big Drawback

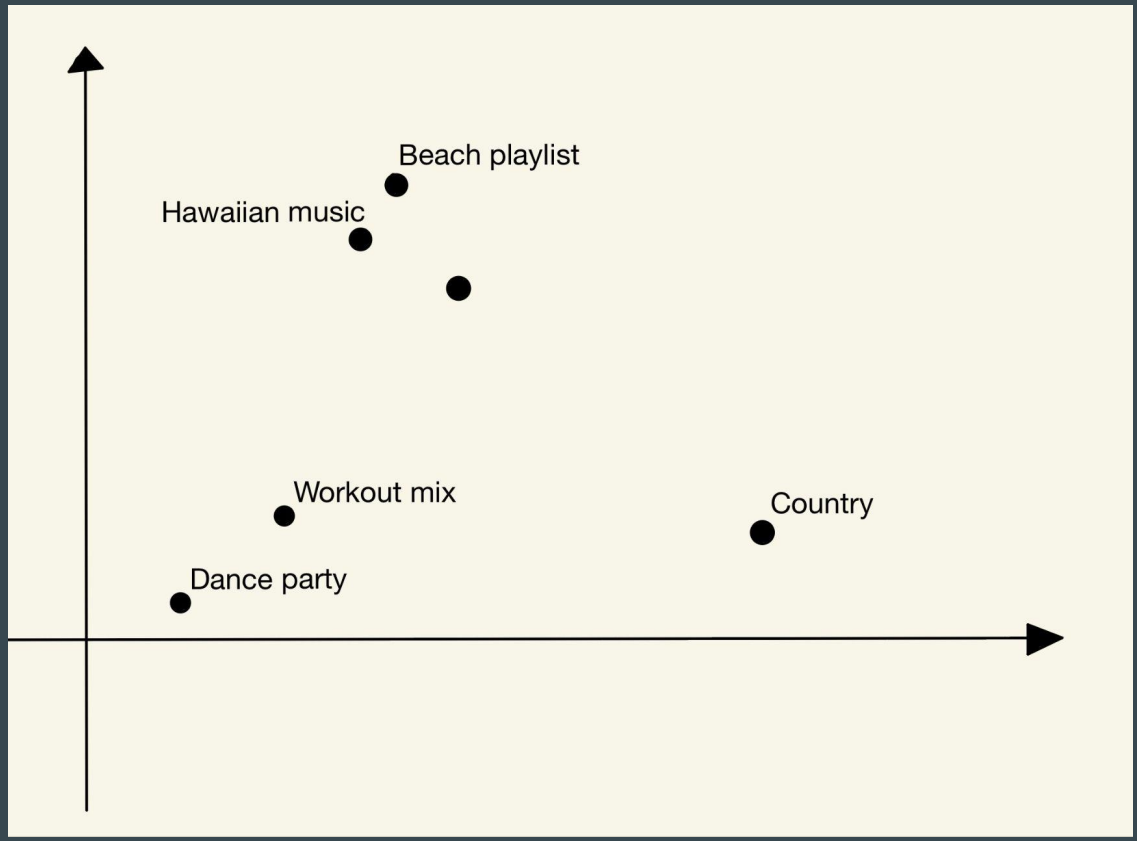
FunkSVD can only expand playlists, not create new ones

We need Seed Tracks!

# Semantic Search

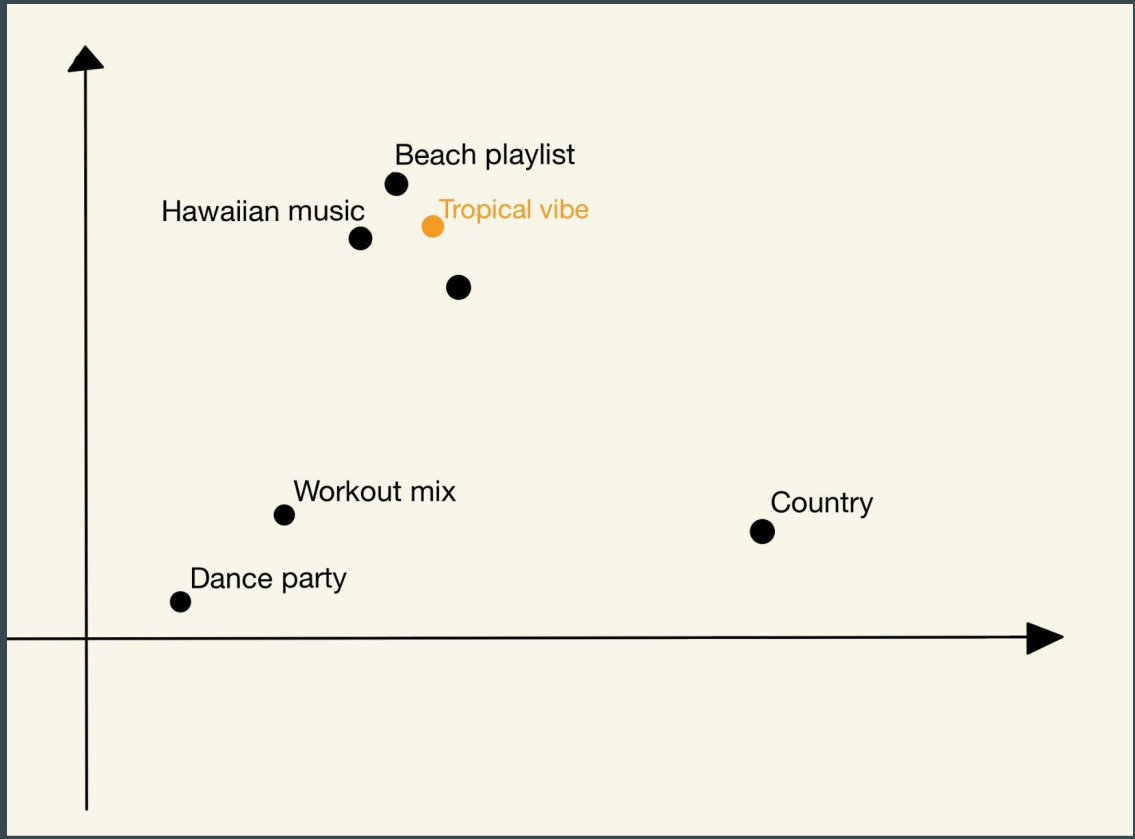
# Semantic Search

- SBERT:
  - Embed playlist titles



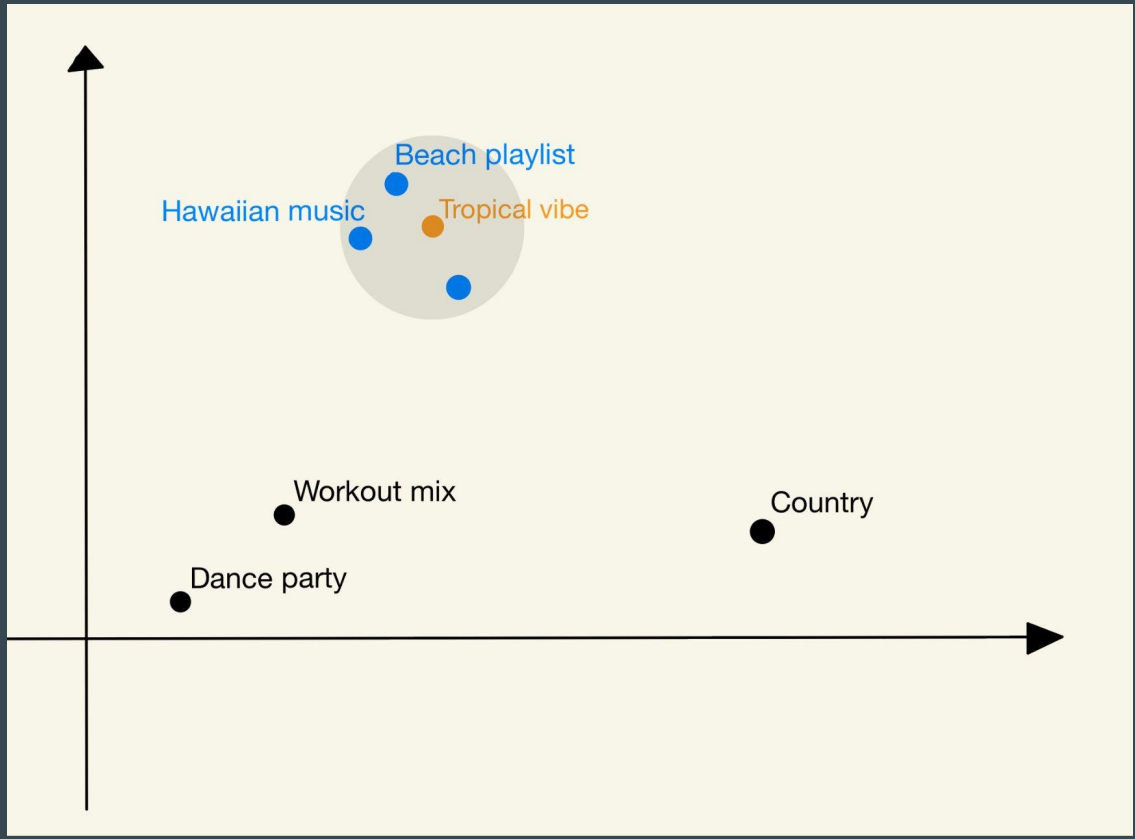
# Semantic Search

- SBERT:
  - Embed playlist titles
  - Embed user prompt



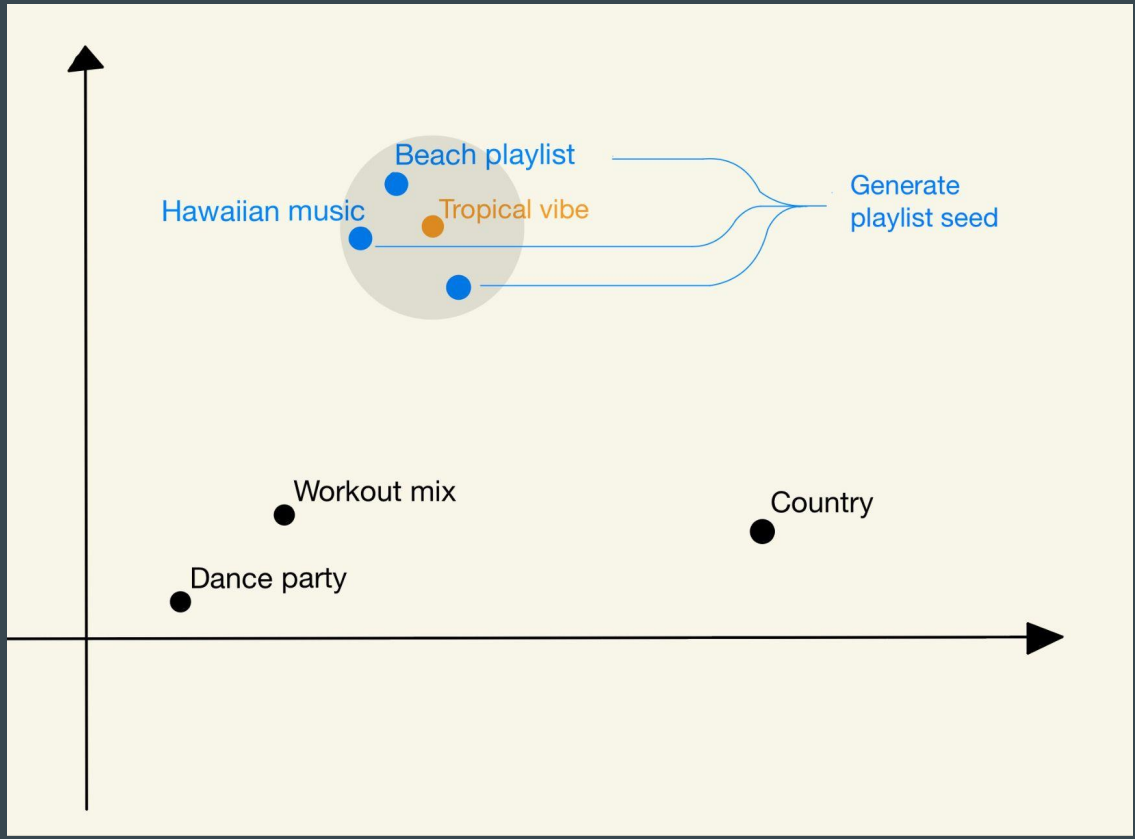
# Semantic Search

- SBERT:
  - Embed playlist titles
  - Embed user prompt
- ANNOY
  - Find nearest neighbors



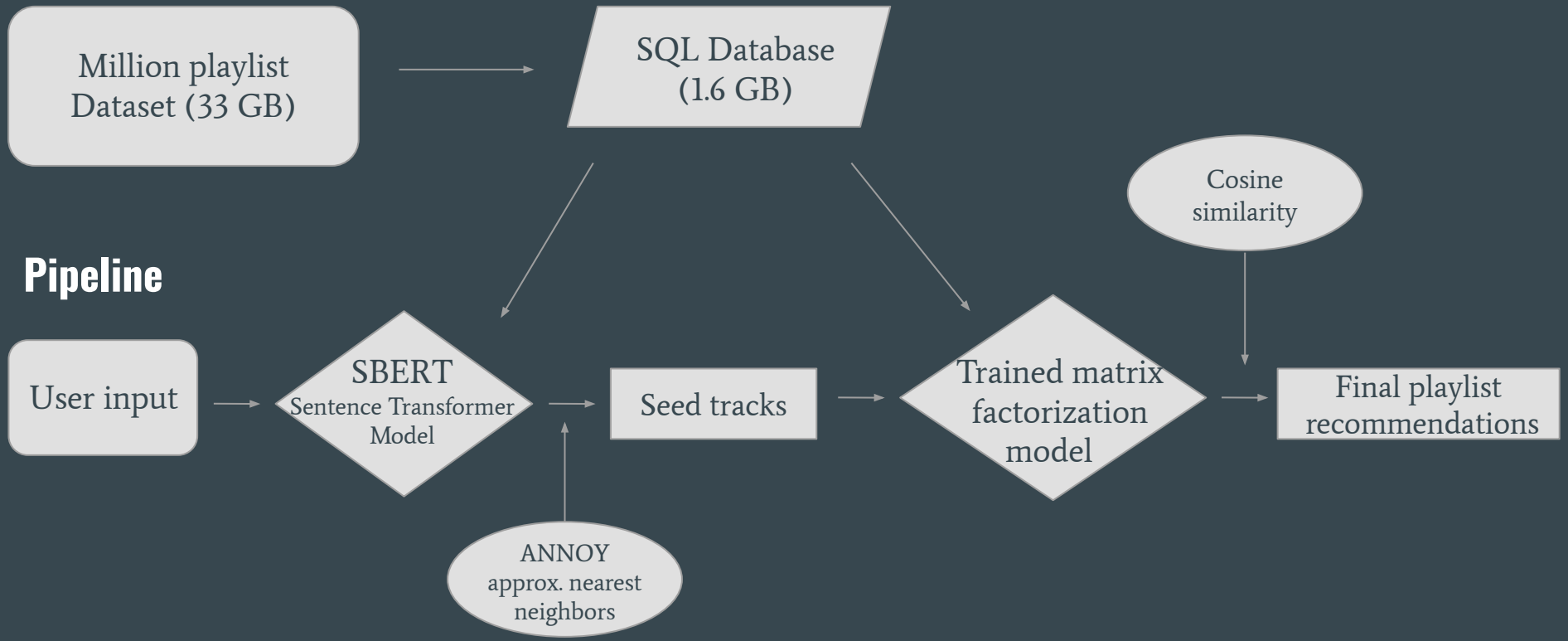
# Semantic Search

- SBERT:
  - Embed playlist titles
  - Embed user prompt
- ANNOY
  - Find nearest neighbors
- Curate songs from playlists to seed the matrix factorization model



# Workflow

## Data Flow



# Demo

# Future Directions

Deploy engine and get user feedback to improve recommendations

Add additional layers, such as the MuSe Music Sentiment database to improve recommendations based on emotion/mood

Incorporate track metadata, such as genre, energy, loudness, or tempo to fine-tune recommendations

# Thanks!

Special thanks to our mentor Matthew Graham

## Resources


Chen, C.-W., Lamere, P., Schedl, M., & Zamani, H. (2018). Recsys challenge 2018: Automatic music playlist continuation. Proceedings of the 12th ACM Conference on Recommender Systems, 527–528. https://doi.org/10.1145/3240323.3240342

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE international conference on data mining (pp. 263-272). IEEE.