Music Recommendation in Spotify

Boxun Zhang

Spotify®

About me

- Data scientist at Spotify
  - Big hype nowadays
  - Build models of user behavior
  - Develop algorithms
  - Design A/B tests

- Ph.D. in CS from TU Delft (NL)
  - Studied user behavior in P2P systems
  - Interned at Spotify

Outline

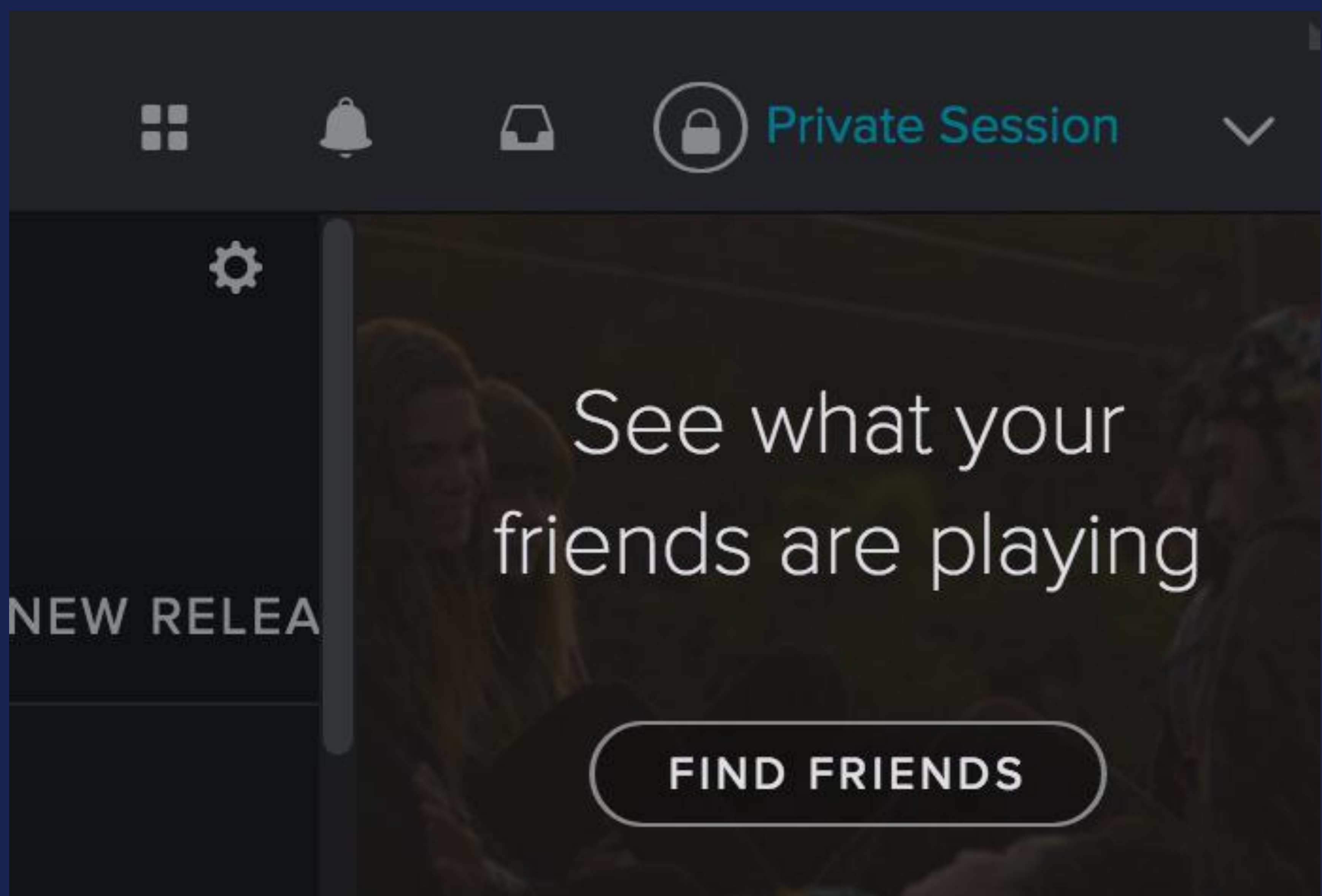
- Spotify basics
- Machine learning at Spotify
- Music recommendation
- Collaborative filtering
  - Latent factor model
  - Approximate nearest neighbor search
- Future work

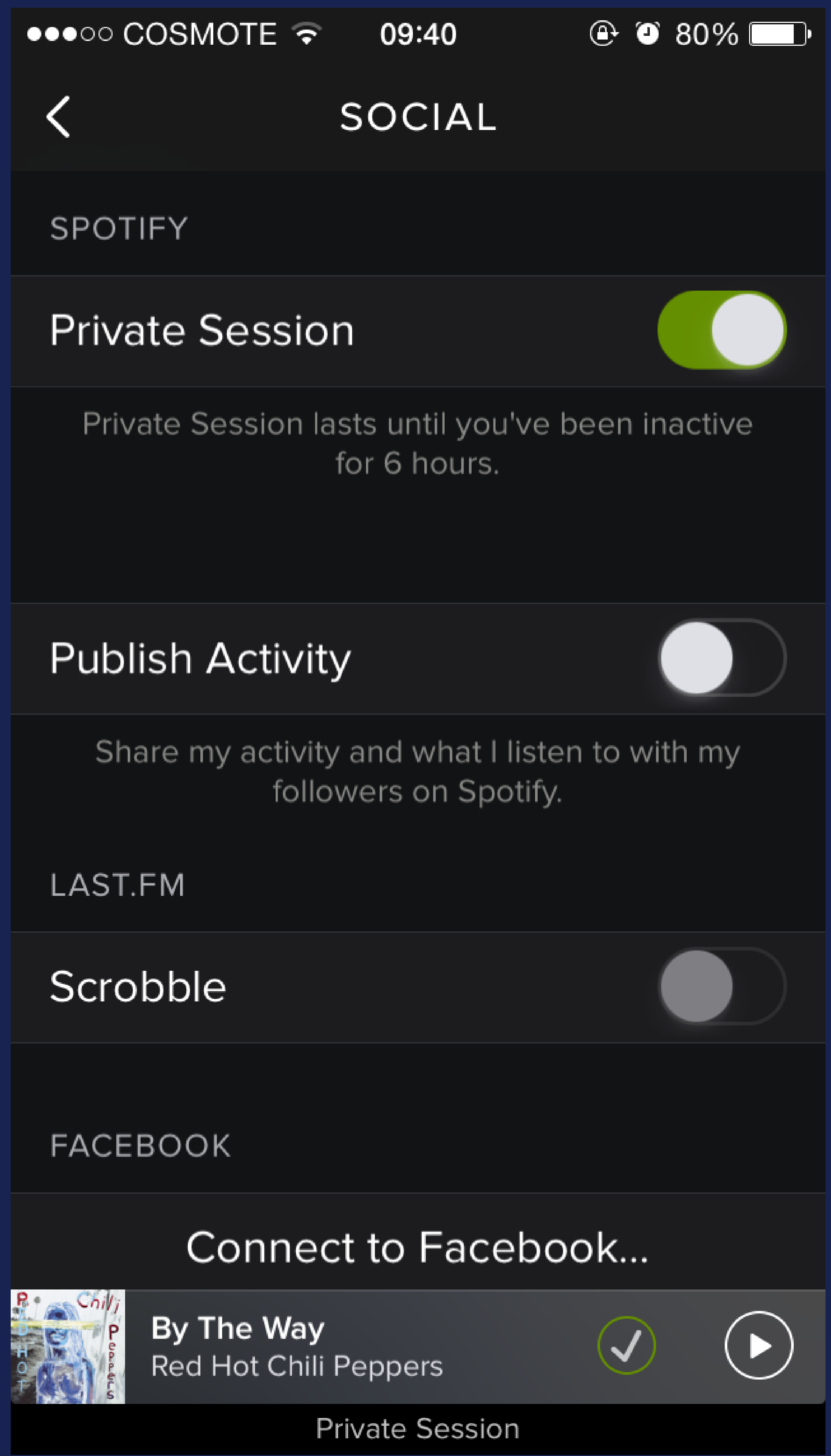
Spotify basics

- A popular music streaming service
  - 60M+ active users
  - 30M+ songs
  - 1.5B+ user-generated playlists
  - Multi-platform, now also on PlayStation
  - Available in 58 countries

Privacy

- Private session ☺





Machine learning at Spotify

- User segmentation
- Churn/conversion prediction
- Ads clicking
- Automatic playlist generation
- Related artists
- Music recommendation

Music recommendation

- Help users to discover good music
  - Search: requires lots of efforts
  - Browse: good curated playlists, but not personalized
  - Discover: personalized recommendations

Not that trivial for our large catalog and user base

Collaborative filtering

- Predict user rating on items
  - Popular strategy for recommender systems
  - Exploits user interactions with items, songs or videos
  - Domain-free
  - Suffers from the *cold start* problem
- Memory-based approach
- Model-based approach

Latent factor model

- Proved to be more effective in the Netflix prize

- How it works
  - Build user-item interaction matrix [users, items]
  - Map user/item vectors to a latent factor space
  - The latent factor space should have much lower dimensions
  - Approximate users' ratings using latent vectors

From video to music

- Implicit user feedback in Spotify
  - Binary rating of songs: 1 if streamed, otherwise 0

- Repetitive consumption
  - An ad-hoc weight on user rating

Compute latent vectors

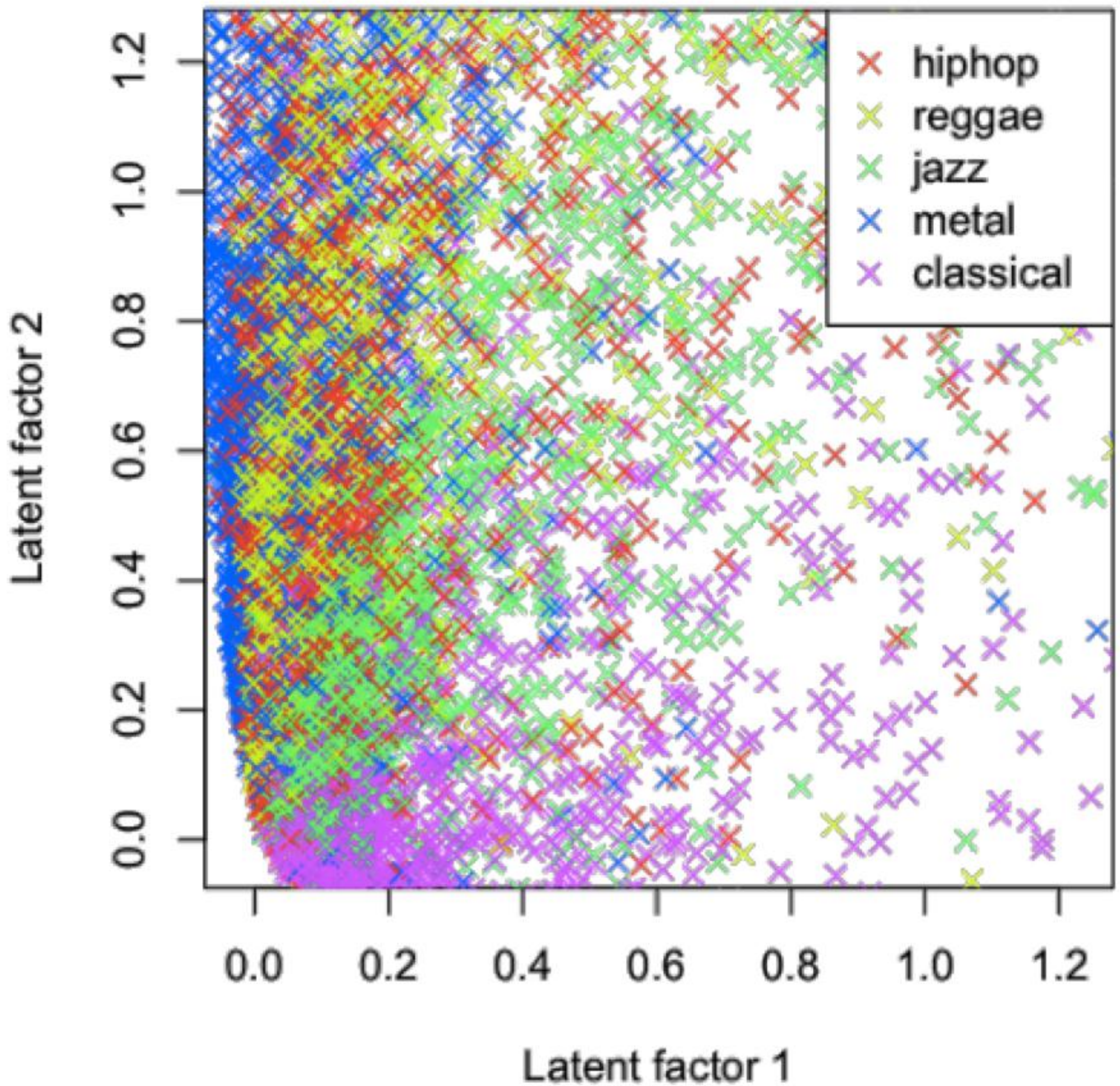
- Minimize the loss function below
  - r_{ui}: 1 if a track if streamed, otherwise 0
  - p_u: user vector
  - q_i: item vector
  - c_{ui}: ad-hoc weight to consider repetitive consumption $1+\alpha \cdot plays_{ui}$
  - λ: regularization penalty

$$\sum_{u,i} c_{ui}(r_{ui} - q_i^T p_u)^2 + \lambda\left(\sum_u \|p_u\|^2 + \sum_i \|q_i\|^2\right)$$

Compute latent vectors, cont.

- Alternating least squares
  - Cost function becomes quadratic when fixing either user factors or item factors
  - Minimize the cost function iteratively until convergent
  - Linear run-time complexity in each iteration
  - Support parallelization in e.g., Hadoop

- Spotify matrix
  - 40 latent factors
  - Computation converges within ~20 iterations (a few hours)
  - On our Hadoop cluster of ~1,300 nodes


hiphop
reggae
jazz
metal
classical
Latent factor 2
Latent factor 1
0.0
0.2
0.4
0.6
0.8
1.0
1.2

The *real* reality

- It’s not only the latent factor model
- We use an ensemble model to approximate user ratings
  - include *some* other information

Find recommendations

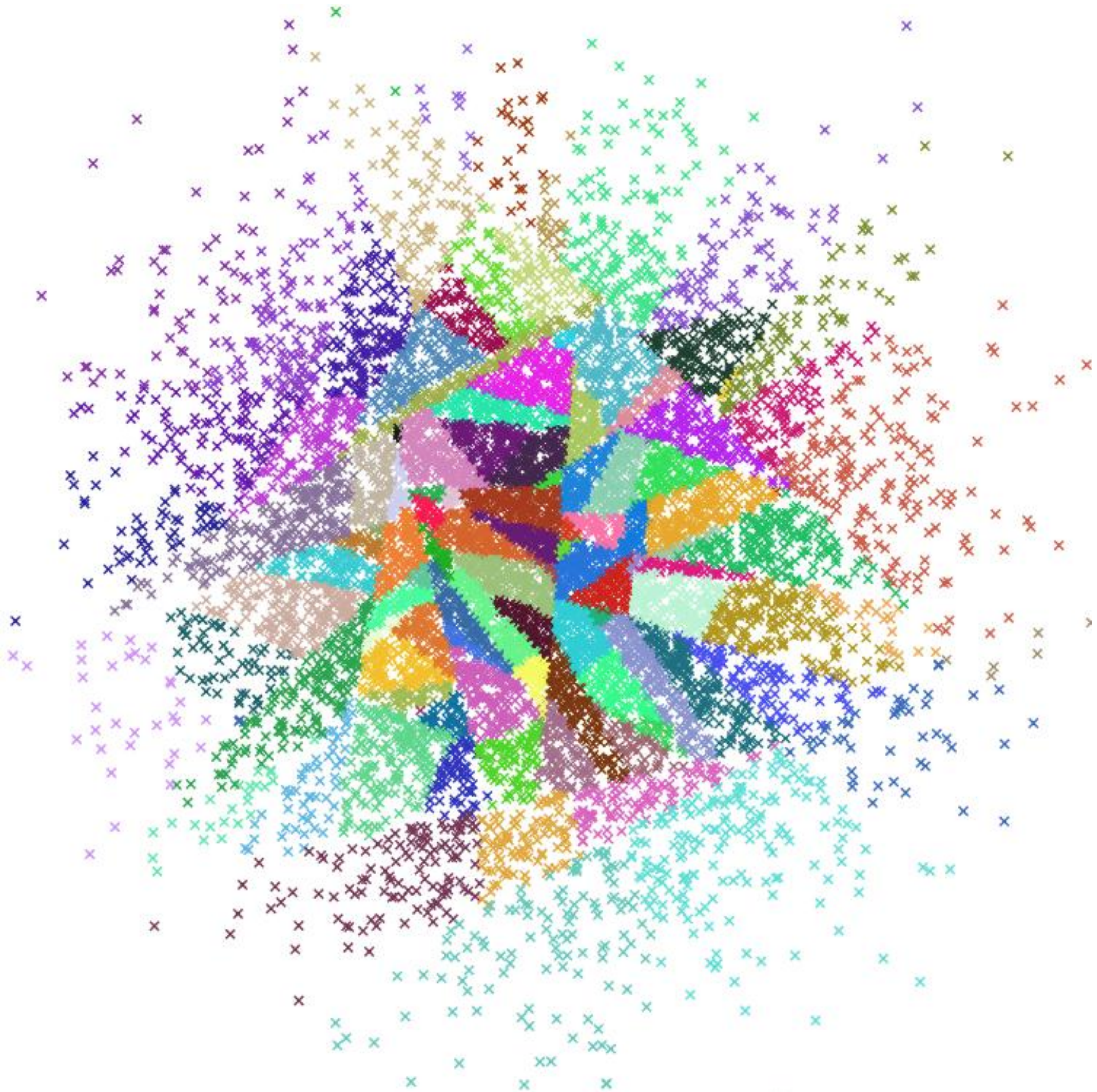
- There are 30M+ songs out there
  - 20K+ songs added every day
  - Brute-force? Too slow, and NOT cool!
  - Use (Approximate) Nearest Neighbor (ANN) search

Annoy

- Locality-sensitive hashing
  - Vectors close to each other are still close nearby after been projected to a space with lower dimensionality or a hyperplane

- Build a tree with intermediate nodes being random hyperplanes
  - Nearby vectors likely to be on the same side
  - Better approximation with several trees
  - Very fast query

www.github.com/spotify/annoy



Future work

- Include bias and temporal patterns into latent factor model
- Improve evaluation of recommender system
- Echo Nest: Signal processing
- Deep learning, maybe

Since two days ago

- Not only music any more
  - Video
  - Podcast
  - News

- Context-based recommendations

- Running


Thank you
Spotify