

Johannes Braun



DATA SCIENCE RETREAT

Final project, 30/04/15

braun.johannes@gmail.com
linkedin.com/in/johannesbraun1
johannesbraun.github.io
github.com/johannesbraun



KU LEUVEN







Sven Marquardt, Bouncer Berghain

Project:

Personal local electronic music radio

Scope :

- **Local:** Berlin
- **Radio:** Autoplay of generated playlists
- **Personal:** Responsive to feedback  
- **Music:** Focus on electronic music

Motivation :

- Scoop into Berghain

Agenda

1. Motivation
2. General approach & Data Engineering
3. Data Science:
 - Generating Recommendations
 - Measuring success
4. Launching the radio service
5. Demo

2. Approach & Data Engineering: Find out about local events



Resident Advisor (RA) is an online music magazine and community platform that's dedicated to showcasing electronic music, artists and events across the globe.

Top Berlin clubs

Berghain/Panorama Bar /
Watergate /
Club der Visionaere /
Tresor /
Weekend /
://about blank /
Salon Zur Wilden Renate /
Golden Gate /
Ritter Butzke /
Arena Club /
Suicide Circus /
Stattbad /

DE • Berlin • By week •

30/
Thu Apr 2015

Event	Location	Attending	Tickets
10 Years BNR at [ipse]	Boys Noize, Modeselektor, Spank Rock, Djedjotr...	391 Attending	
Into May - Umami, Andree Wischnewski, Jan Pyroman	at Suicide Circus	59 Attending	
Grüner in den Mai	at Neu West Berlin	25 Attending	
Tanz in den Mai, u.a. die Vögel, Marcus Meinhardt, Mat.Joe	at Ritter Butzke	184 Attending	
Tanz in den Mai / Klangost Spektakel	at Kosmonaut	102 Attending	
DIE DAMEN	at Watergate	79 Attending	



2. Approach & Data Engineering: Search for lineup on soundcloud



SoundCloud is an online audio distribution platform that enables its users to upload, record, promote, and share their originally-created sounds. The API exposes resources like sounds, sets and users

500 tracks found , 373 playlists found , 167 people found , and 5 groups found

A search bar containing the text "modeselektor".
A small search icon consisting of a magnifying glass symbol.

Modeselektor A small orange circular badge with a white checkmark.

Berlin / Germany

284,351 370

Following

A button labeled "Following" in red text.

2. Approach & Data Engineering: Extract soundcloud user preferences



API: Find the set of users that have liked
Modeselektor: Siriusmo "Itchy"

Modeselektor
Siriusmo "Itchy" taken from the forthcoming album "Enthusiast" (MONKEYTOWN033) Out June 14
2 years #Siriusmotronic 3:32

Liked Repost Add to playlist Share Buy ▶ 262,608 ❤ 6K ↵ 981 🗑 333

Modeselektor
Modeselektor - German Clap
3 years #Techno 0:45

Like Repost Add to playlist Share Buy ▶ 63,901 ❤ 482 ↵ 35 🗑 50

2. Approach & Data Engineering: Extract soundcloud user preferences

Screenshot of a SoundCloud repost page for the track "Itchy" by Siriusmo. The page shows the original post and a grid of 15 reposts from other users.

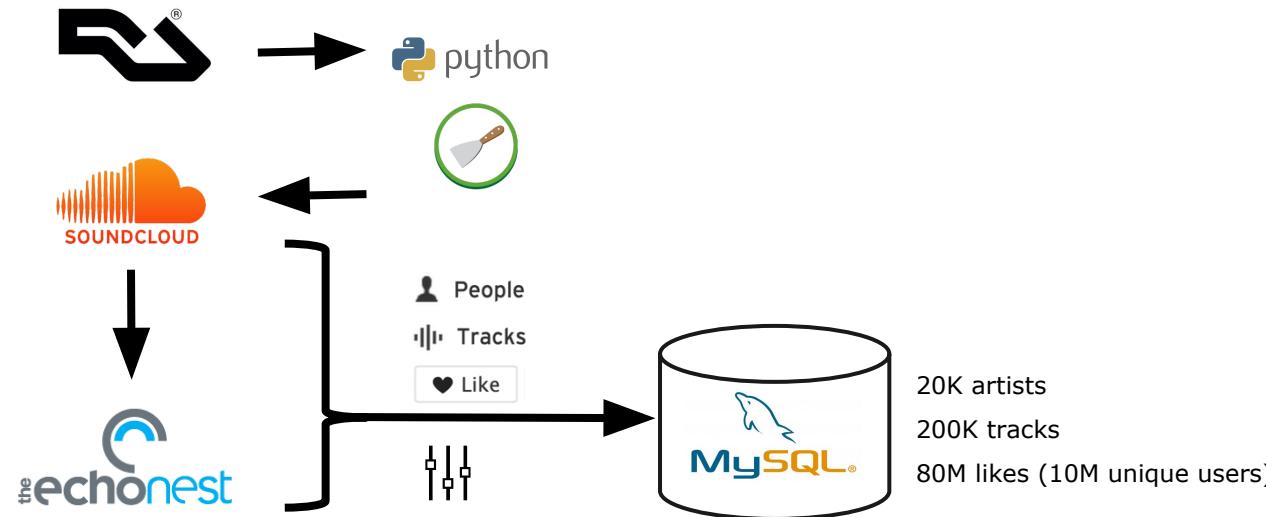
The main post (top left) includes:

- SoundCloud logo
- Artist: Siriusmo
- Title: "Itchy"
- Album: "Enthusiast" (M...)
- Reposts: 16
- Likes: 16

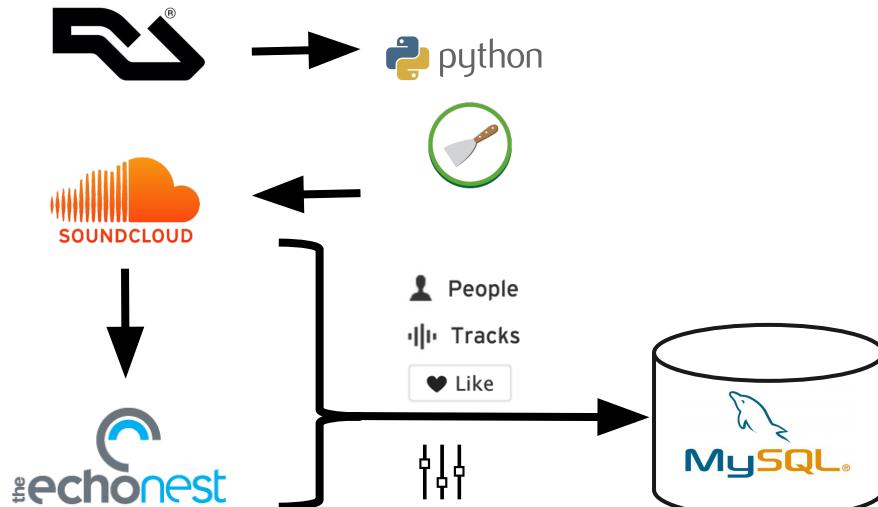
The repost grid (bottom right) shows 15 user profiles with their names, profile pictures, and repost counts:

User	Reposts	Likes
Philippe Renner	16	16
Bryan Oxylice	6	6
irrelevation	13	13
Schaddenhook	5,814	5,814
David Louthan	2,655	2,655
Teddy Cherry	443	443
PELITO DE ALACRÁN	895	895
KachiT	10	10
seneque	23	23
GGGGGG	9	9
CARNA_ONE	5	5
Jagan Stich Longaretti	33	33
boly	649	649
EMI	242	242
Piet Pietersen	23	23

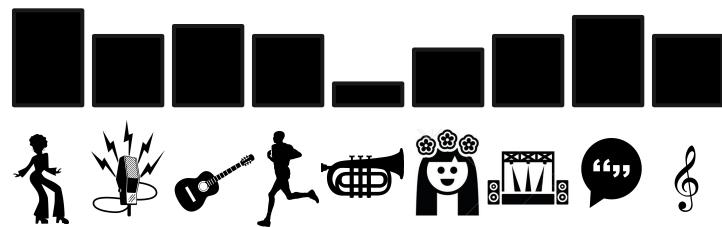
2. Approach & Data Engineering: Extracting Audio Features



2. Approach & Data Engineering: Extracting Audio Features



The Echo Nest is the industry's leading music intelligence company. The API offers a service for analyzing audio files:





Agenda

1. Motivation
2. Approach & Data Engineering
3. Data Science:
 - Generating Recommendations
 - Measuring success
4. Launching the radio service
5. Demo

3.1 Data Science: Recommendations

a) Collaborative Filtering



1) Find users that like *Modeselektor: Siriusmo "Itchy"*

A screenshot of a SoundCloud player. At the top left is the SoundCloud logo. To its right, the artist name "Modeselektor" is followed by the track title "Siriusmo 'Itchy'" and a description: "taken from the forthcoming album 'Enthusiast' (MONKEYTOWN033) Out June 14". To the right of the title is a timestamp "2 years" and the hashtag "#Siriusmotronic". Below the title is a waveform visualization of the track. Underneath the waveform is a horizontal bar showing various user profiles. A red box highlights the "6K" likes count, which is preceded by a heart icon and the text "Liked". Below the waveform are several interaction buttons: "Reposted" (with a retweet icon), "Add to playlist" (with a plus icon), "Share" (with a link icon), "Buy" (with a price tag icon), and the highlighted "6K" likes button. Further to the right are "981" reposts and "333" comments.

3.1 Data Science: Recommendations

a) Collaborative Filtering



- 1) Find users that like *Modeselektor: Siriusmo "Itchy"*
- 2) Find other favorites of these users

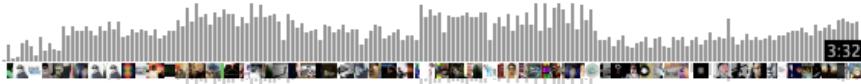


Modeselektor

Siriusmo "Itchy" taken from the forthcoming album
"Enthusiast" (MONKEYTOWN033) Out June 14

2 years

#Siriusmotronic



Heart Liked

Reposted

Add to playlist

Share

Buy 262,755

Heart 6K

1,981

333



Four Tet

Lion (Jamie xx remix)

2 years



Heart Like

Reposted

Add to playlist

Share

1,212,186 Heart 25K 1,981 804

3.1 Data Science: Recommendations

a) Collaborative Filtering



- 1) Find users that like *Modeselektor: Siriusmo "Itchy"*
- 2) Find other favorites of these users
- 3) Rank other tracks by # of co-occurrence



▶ Modeselektor
Siriusmo "Itchy" taken from the forthcoming album
"Enthusiast" (MONKEYTOWN033) Out June 14

2 years #Siriusmotronic

3:32

Liked Repost Add to playlist Share Buy 262,755 Heart 6K 981 333



▶ Four Tet
Lion (Jamie xx remix)

2 years

7:08

Like Repost Add to playlist Share 1,212,186 Heart 25K 3K 804

3.1 Data Science: Recommendations

a) Collaborative Filtering



- 1) Find users that like *Modeselektor: Siriusmo "Itchy"*
- 2) Find other favorites of these users
- 3) Rank other tracks by # of co-occurrence



▶ Modeselektor
Siriusmo "Itchy" taken from the forthcoming album
"Enthusiast" (MONKEYTOWN033) Out June 14

2 years #Siriusmotronic

212 3:32

Liked Repost Add to playlist Share Buy 262,755 6K 981 333



▶ Four Tet
Lion (Jamie xx remix)

2 years

1,212,186 25K 3K 804

Like Repost Add to playlist Share

Log-Likelihood-Ratio:
Normalizing co-occurrence

	A	!A
B	A & B 212	!A & B 5.8K
!B	A & !B 24.8K	!A & !B 10M

$$\text{LLR} = 525.2563$$

[http://tdunning.blogspot.de/2008/03/
surprise-and-coincidence.html](http://tdunning.blogspot.de/2008/03/surprise-and-coincidence.html)

3.1 Data Science: Recommendations

a) Collaborative Filtering



- 1) Find users that like *Modeselektor: Siriusmo "Itchy"*
- 2) Find other favorites of these users
- 3) Rank other tracks by # of co-occurrence



▶ Modeselektor
Siriusmo "Itchy" taken from the forthcoming album
"Enthusiast" (MONKEYTOWN033) Out June 14

2 years #Siriusmotronic

3:32

Liked Reposted Add to playlist Share Buy 262,755 Heart 6K 981 333



▶ dixon
in our wilderness

4 months #dixon

1:04:55

Like Repost Add to playlist Share 206,544 Heart 11K 3K 679

Log-Likelihood-Ratio:
Normalizing co-occurrence

	A	!A
B	A & B 86	!A & B 5.9K
!B	A & !B 9.9K	!A & !B 10M

$$\text{LLR} = 202.2691$$

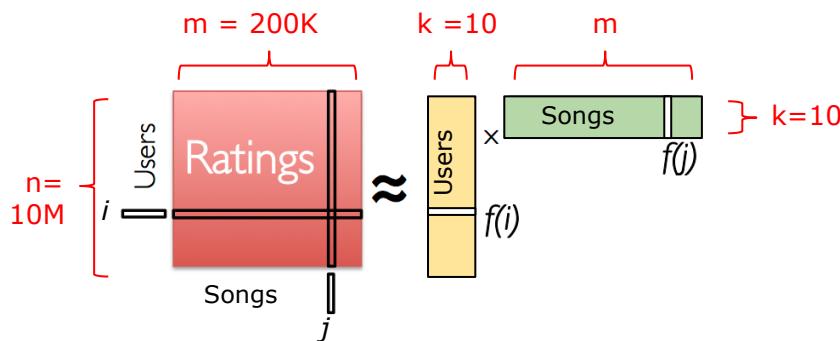
[http://tdunning.blogspot.de/2008/03/
surprise-and-coincidence.html](http://tdunning.blogspot.de/2008/03/surprise-and-coincidence.html)

3.1 Data Science: Recommendations

b) Matrix factorization

Alternating Least Squares (Apache Spark):

Assume small # of k latent factors determine preference:



3.1 Data Science: Recommendations b) Matrix factorization

Approximate ratings to hopefully
generalize to new users:

	Songs				
Users	1	1	1	0	0
0	0	0	1	0	0
0	1	0	0	1	1
1	0	1	0	0	1
0	0	0	0	1	0
1	1	0	0	0	0

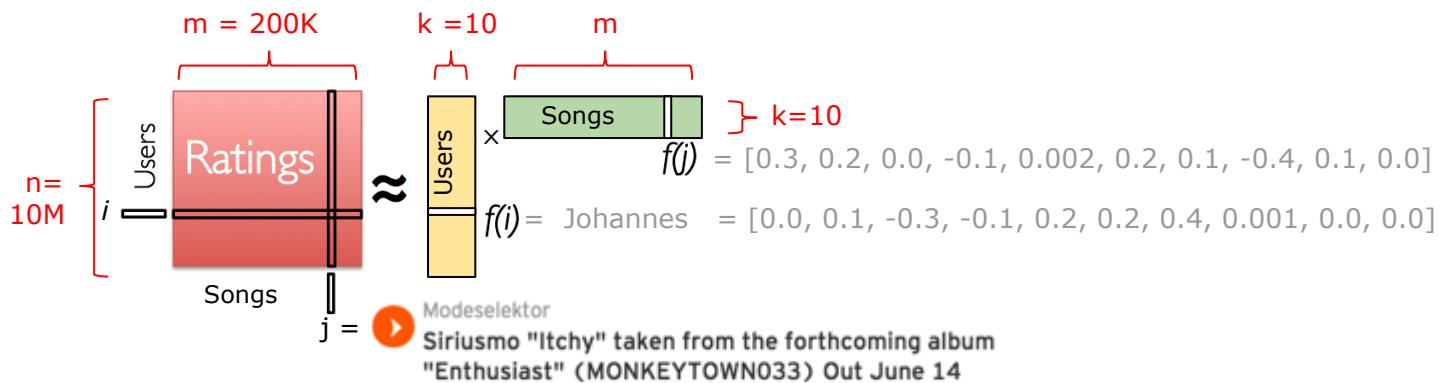
≈

0.96	0.99	0.99	0.38	0.93
0.44	0.39	0.98	-0.11	0.39
0.70	0.99	0.42	0.98	0.98
1.00	1.04	0.99	0.44	0.98
0.11	0.51	-0.13	1.00	0.57
0.97	1.00	0.68	0.47	0.91

3.1 Data Science: Recommendations b) Matrix factorization

Alternating Least Squares (Apache Spark):

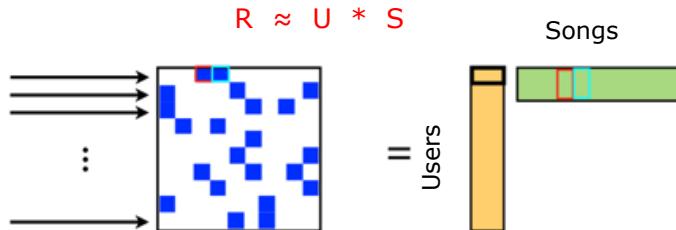
Assume small # of k latent factors determine preference:



3.1 Data Science: Recommendations

b) Matrix factorization: ALS in Spark

Alternating least squares: $\sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y)^2}{n}}$.
Parallel optimization

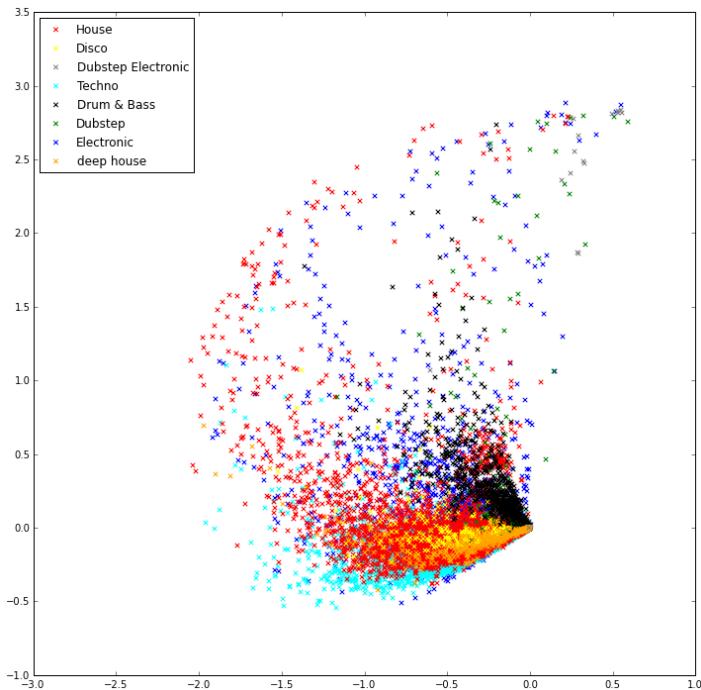


$$\text{Training error for first user} = (\text{blue square} - \text{orange bar}[red square]) + (\text{blue square} - \text{orange bar}[green square])$$

approximated rating for blue square

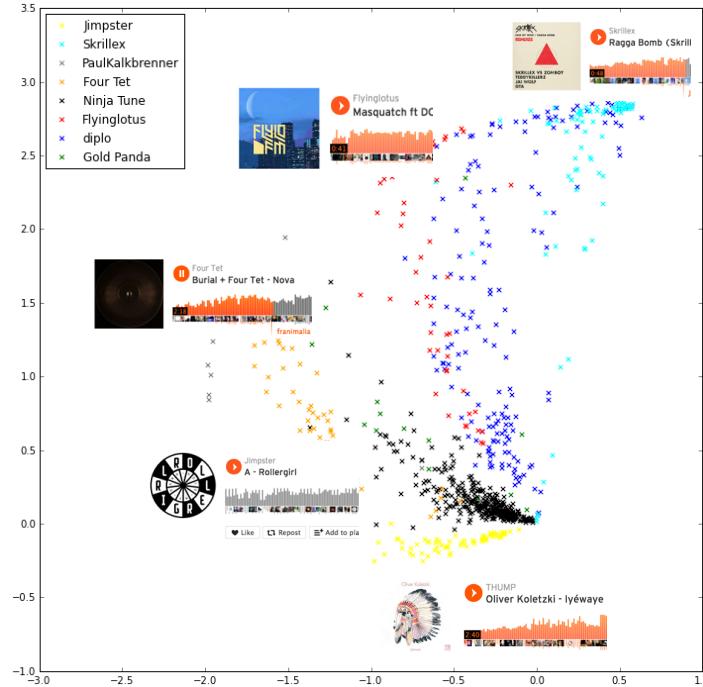
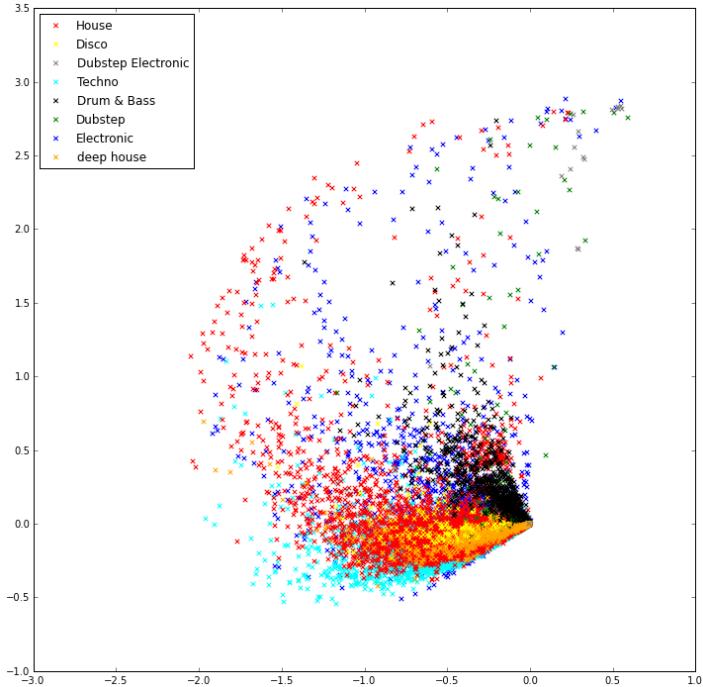
3.1 Data Science: Recommendations

b) Matrix factorization results: k=2



3.1 Data Science: Recommendations

b) Matrix factorization results: k=2



Skrillex: 0:35

<https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/178436255>

Electronic: Four Tet: 1:40

<https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/38720262>

Deep House: Jimpster 1:30

<https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/88568930>

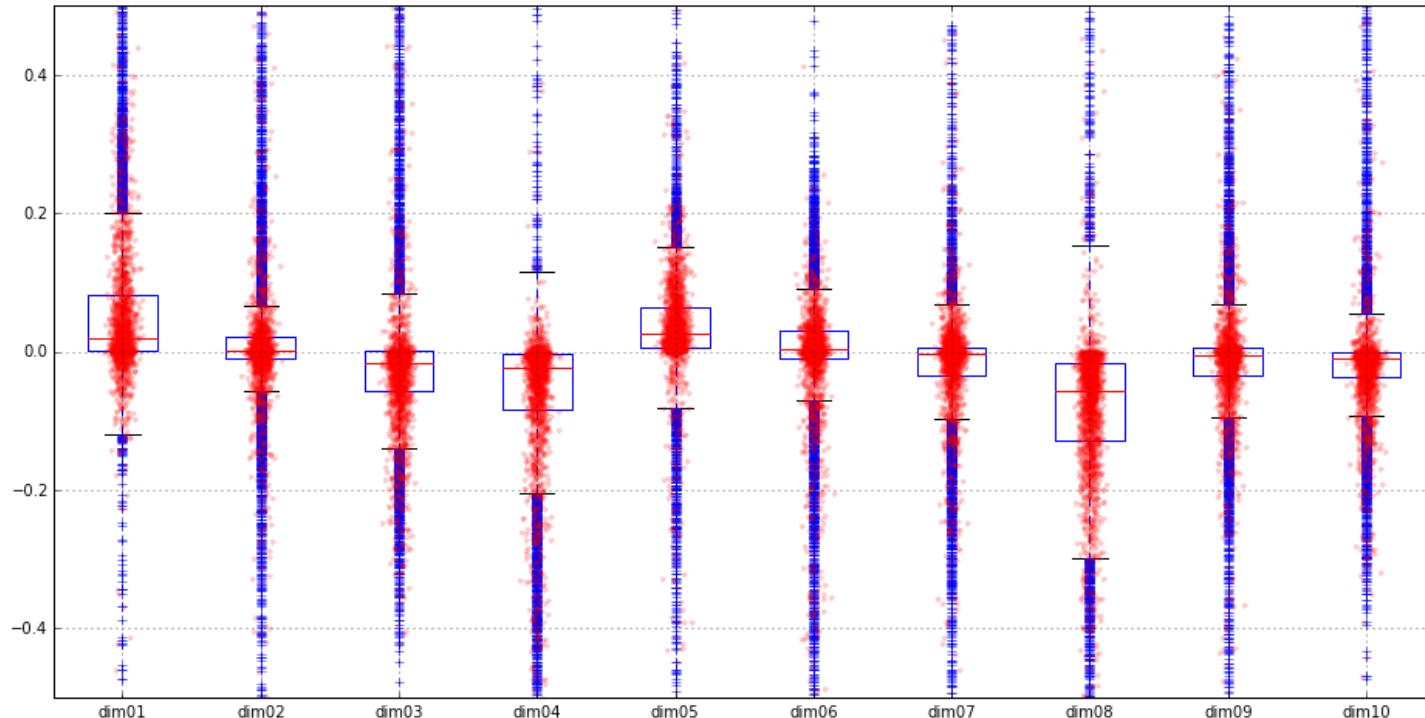
Techno: Oliver Koletzki 3:11

<https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/198557427>

backup links

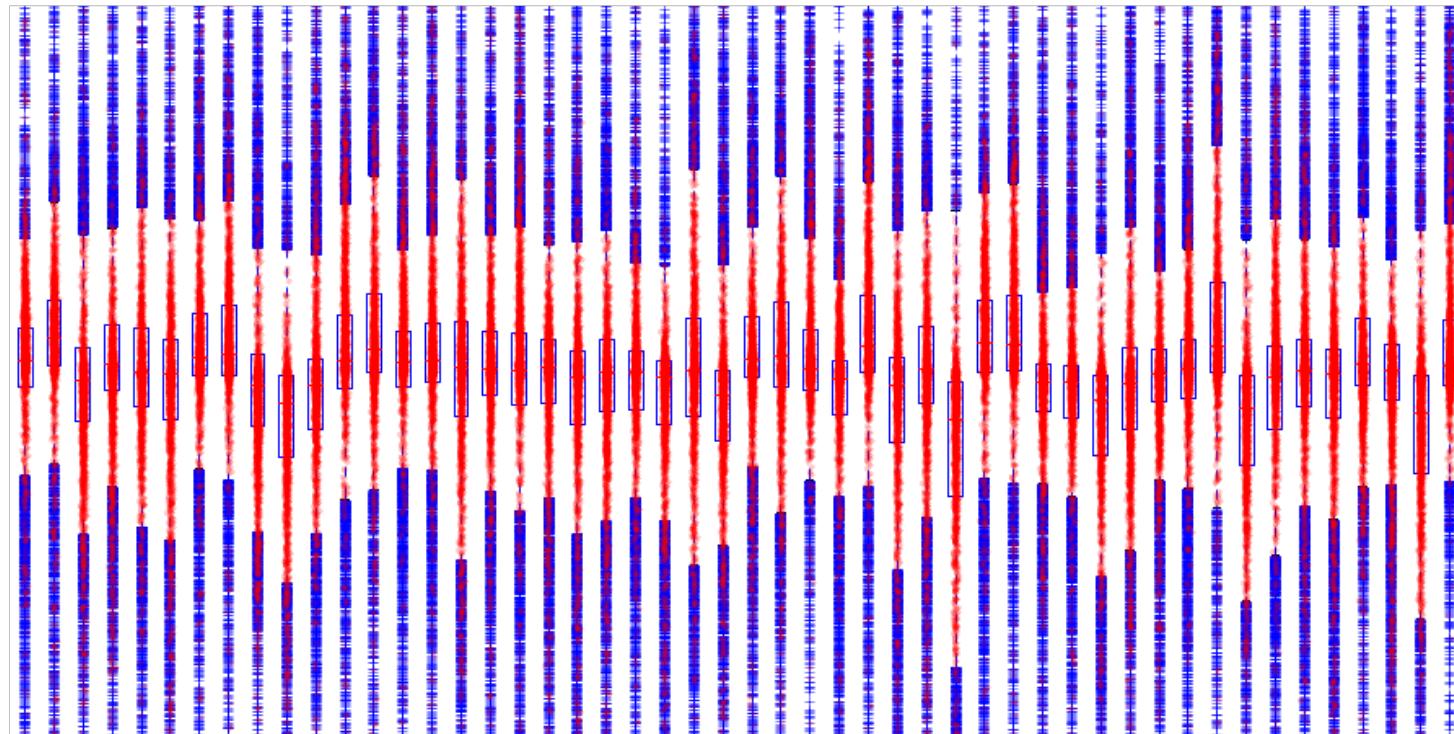
3.1 Data Science: Recommendations

b) Matrix factorization results: k=10



3.1 Data Science: Recommendations

b) Matrix factorization results: k=50





3.1 Data Science: Recommendations

c) Content based/ hybrid approach

- Same artist
- Same genre
- Related artists
- Audio similarity

3.1 Data Science: Recommendations: Audio similarity



Danceability: Describes how suitable a track is for dancing using a number of musical elements: tempo, rhythm stability, beat strength, and overall regularity.



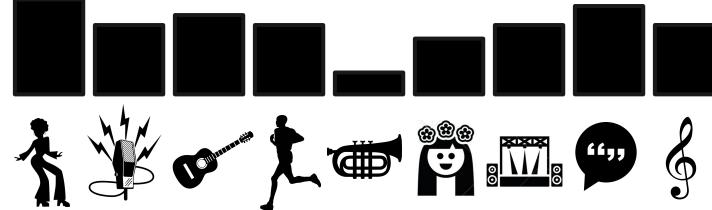
Energy: Represents a perceptual measure of intensity and powerful activity released throughout the track. Typical energetic tracks feel fast, loud, and noisy.



Acousticness: Represents the likelihood a recording was created by solely acoustic means such as voice and acoustic instruments as opposed to electronically such as with synthesized, amplified, or effected instruments.



Tempo: the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.



Instrumentalness: Represents a measure of how likely a song is to be all instrumental with no spoken words



Valence: Describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g., happy, cheerful, euphoric)



Liveness: Detects the presence of an audience in the recording.



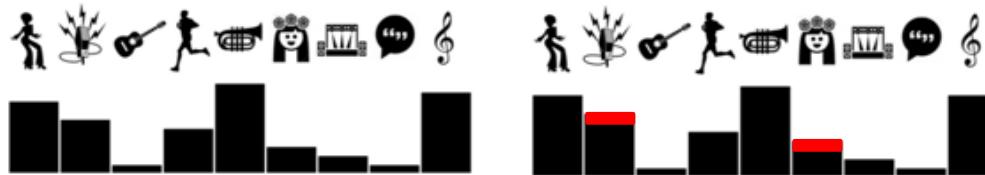
Speechiness: Speechiness indicates the likelihood the track contains spoken words. Tracks with singing have low speechiness AND low instrumentalness, whereas rap tracks could have low instrumentalness and MEDIUM speechiness.



Key: the estimated overall key of a track. The key identifies the tonic triad, the chord, major or minor, which represents the final point of rest of a piece.



3.1 Data Science: Recommendations: Audio similarity



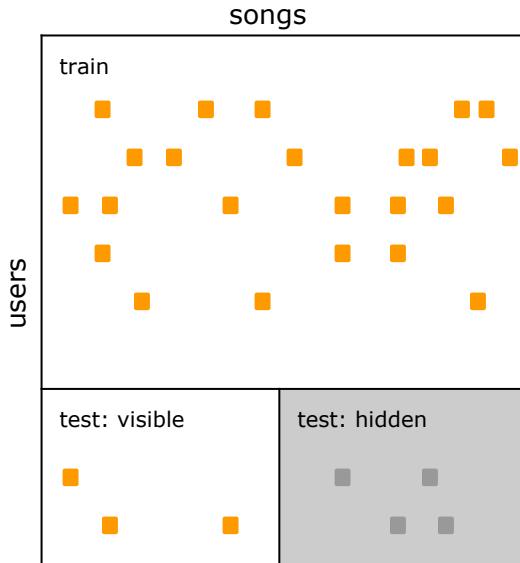


Agenda

1. Motivation
2. Approach & Data Engineering
3. Data Science:
 - Generating Recommendations
 - Measuring success
4. Launching the radio service
5. Demo

3.2 Data Science: Measuring success

Formulating the challenge



Predict which songs a user will like:

Given:

- 1) all favourites for 10M users
- 2) 50% of the favourites for 100K users

What is the task?

Predict the hidden 50%

3.2 Data Science: Measuring success

Formulating the challenge

userid 5506106	favourite tracks:	predicted tracks:	prediction score:
visible	a) Modeselektor - Siriusmo "Itchy"	-	-
visible	b) Four Tet - Lion (Jamie xx remix)	-	-
hidden	c) Jimpster - Hijackin' Berlin	e) Bonobo-Essential Mix	0.8
hidden	d) Four Tet - Final Plastic People	c) Jimpster - Hijackin' Berlin	0.6
		d) Four Tet - Final Plastic People	0.1

3.2 Data Science: Measuring success

Formulating the challenge

userid 5506106	favourite tracks:	predicted tracks:	prediction score:
visible	a) Modeselektor - Siriusmo "Itchy"	-	-
visible	b) Four Tet - Lion (Jamie xx remix)	-	-
hidden	c) Jimpster - Hijackin' Berlin	e) Bonobo-Essential Mix	0.8
hidden	d) Four Tet - Final Plastic People	c) Jimpster - Hijackin' Berlin	0.6
		d) Four Tet - Final Plastic People	0.1

$$\text{MSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y)^2}{n}}.$$

$$\text{AUC} = \text{sum} (\text{score}([c, d]) > \text{score}([r1, r2])) / 2$$

AUC may be viewed as the probability that a random positive item scores higher than a random negative one.

$$\text{MAP (n=3)} = (0 + 1/2 + 2/3) / 2 = 0.58333$$

Mean average precision for a set of users is the mean of the average proportion of correct recommendations for each user

3.2 Data Science: Measuring success

Formulating the challenge

userid 5506106	favourite tracks:	predicted tracks:	prediction score:
visible	a) Modeselektor - Siriusmo "Itchy"	-	-
visible	b) Four Tet - Lion (Jamie xx remix)	-	-
hidden	c) Jimpster - Hijackin' Berlin	e) Bonobo-Essential Mix	0.8
hidden	d) Four Tet - Final Plastic People	c) Jimpster - Hijackin' Berlin	0.6
		d) Four Tet - Final Plastic People	0.1

$$\text{MSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y)^2}{n}}.$$

$$\text{AUC} = \text{sum} (\text{score}([c, d]) > \text{score}([r1, r2])) / 2$$

AUC may be viewed as the probability that a random positive item scores higher than a random negative one.

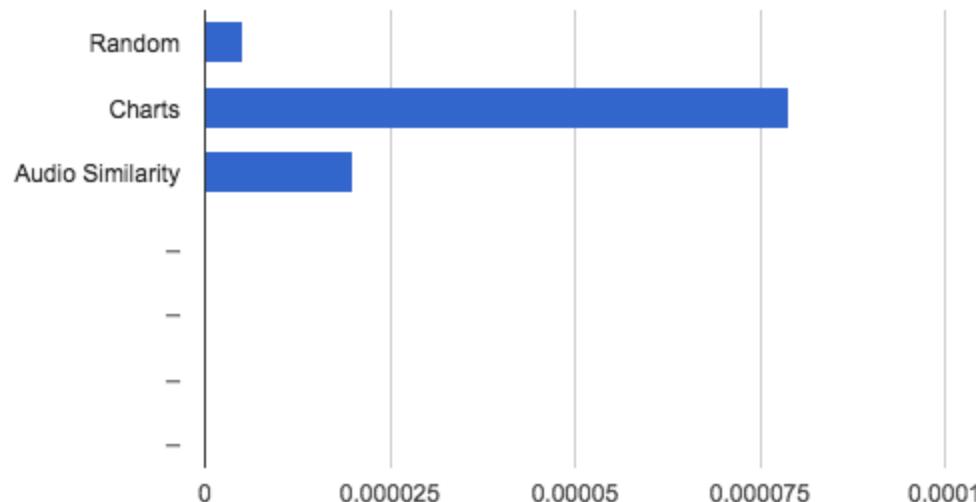
$$\text{MAP (n=3)} = (0 + 1/2 + 2/3) / 2 = 0.58333$$

Mean average precision for a set of users is the mean of the average proportion of correct recommendations for each user

3.2 Data Science: Measuring success

Performance overview

MAP (n=50)



Charts:



#1

#2

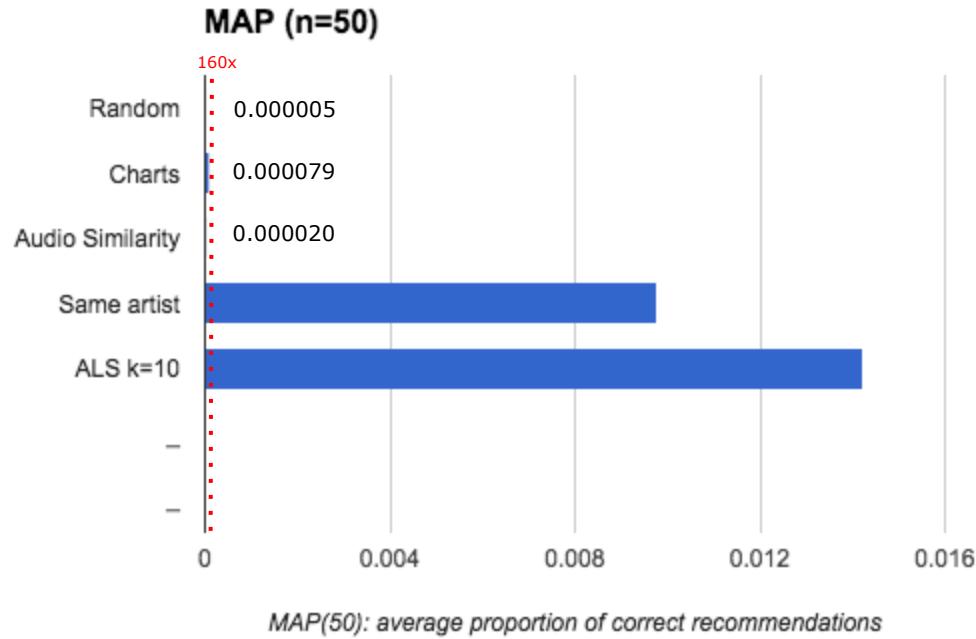
#3

... #200.000

MAP(50): average proportion of correct recommendations

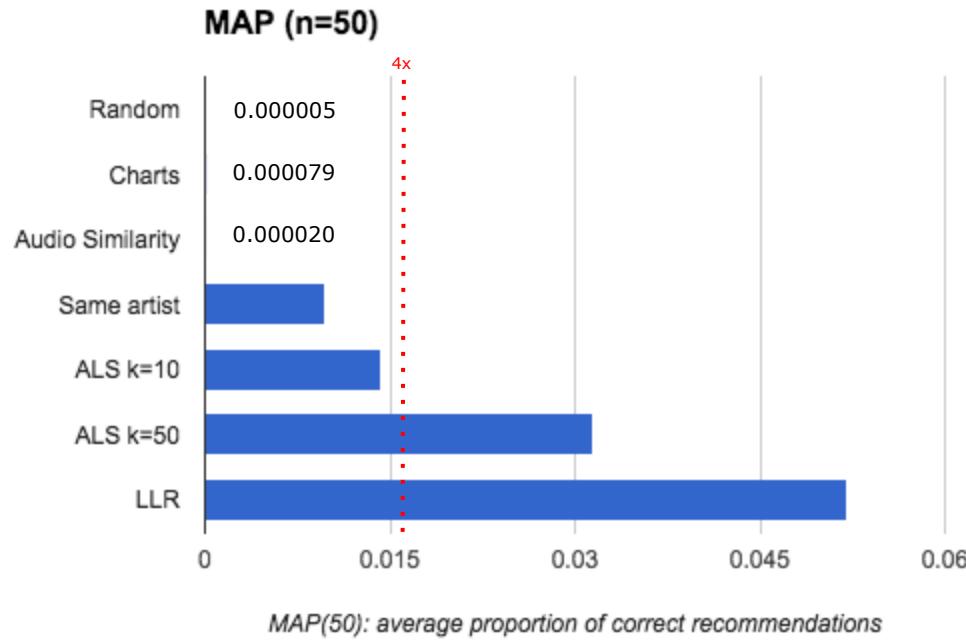
3.2 Data Science: Measuring success

Performance overview



3.2 Data Science: Measuring success

Performance overview



3.2 Data Science: Recommendations

c) House vs. Techno audio classifier

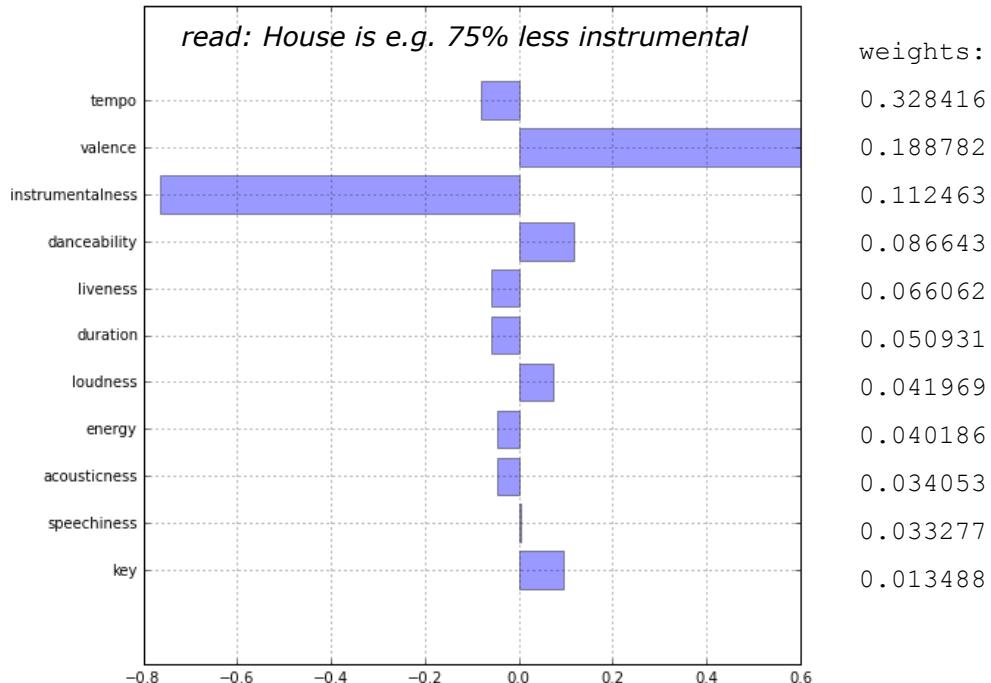
genre	precision	recall	f1-score
techno	0.82	0.80	0.81
house	0.83	0.84	0.83

Confusion matrix

Predicted label		Techno	House
True label	Techno	275	75
House	75	225	

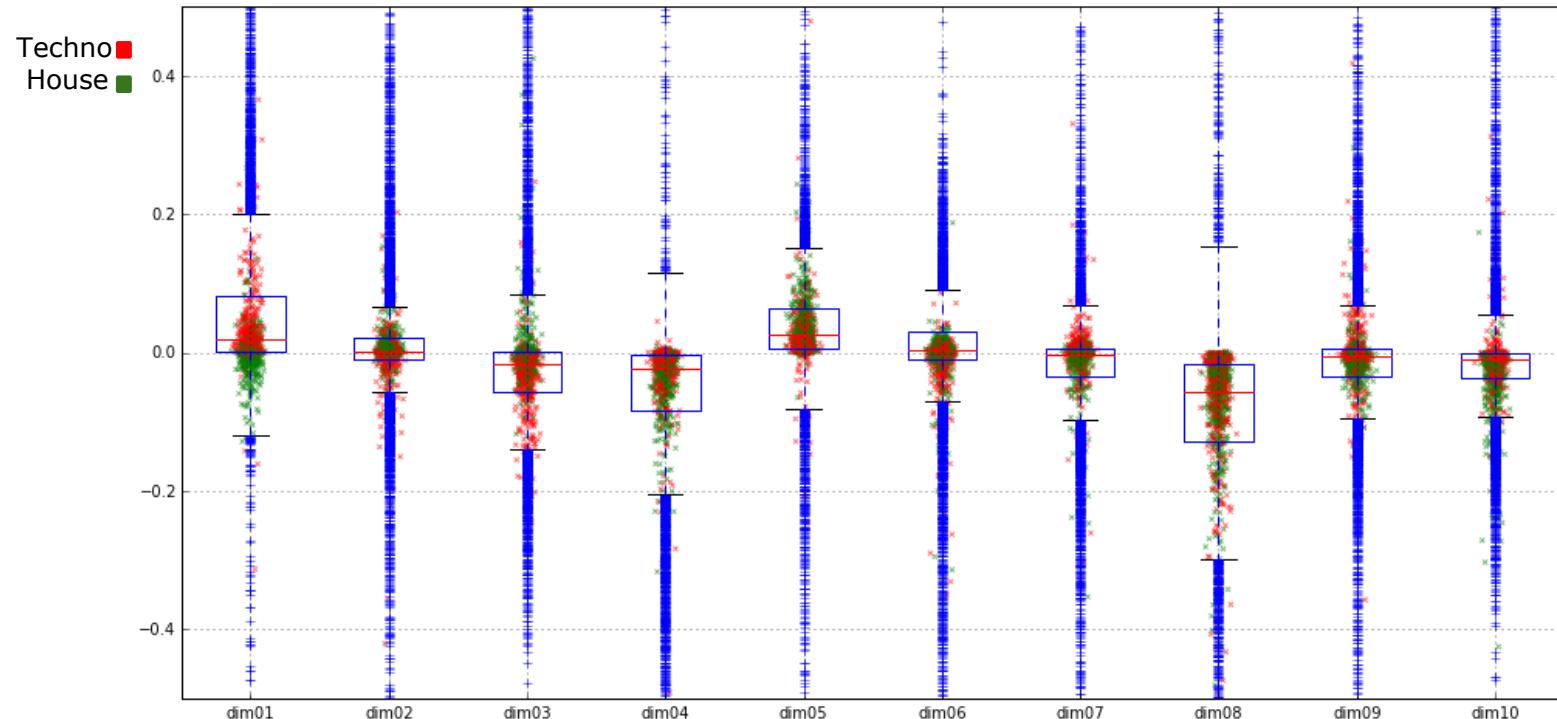
Predicted label

Random Forest classifier based on  features



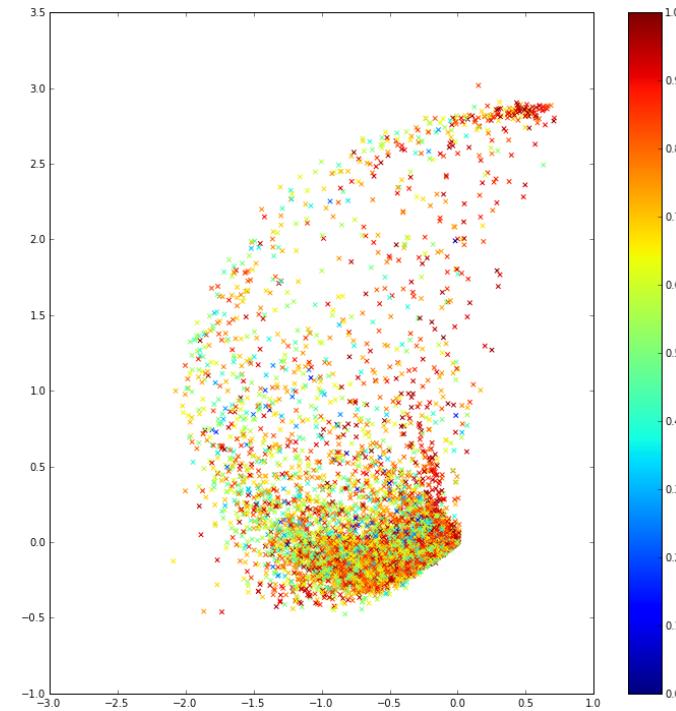
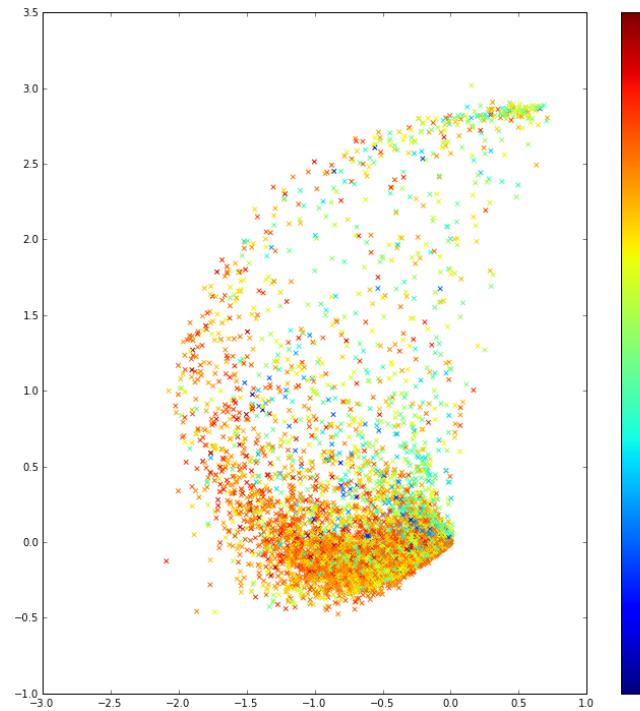
3.2 Data Science: Recommendations

c) Techno vs House in 10D



3.2 Data Science: Recommendations

c) Danceability vs Energy

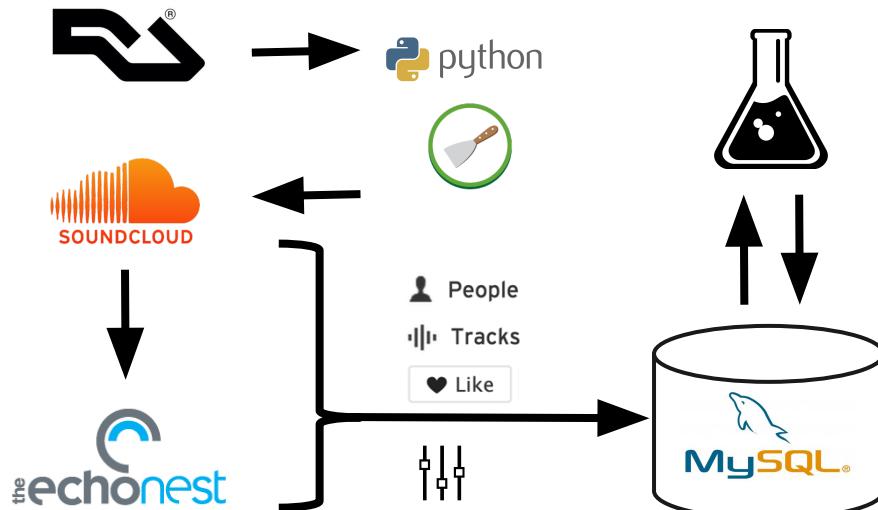




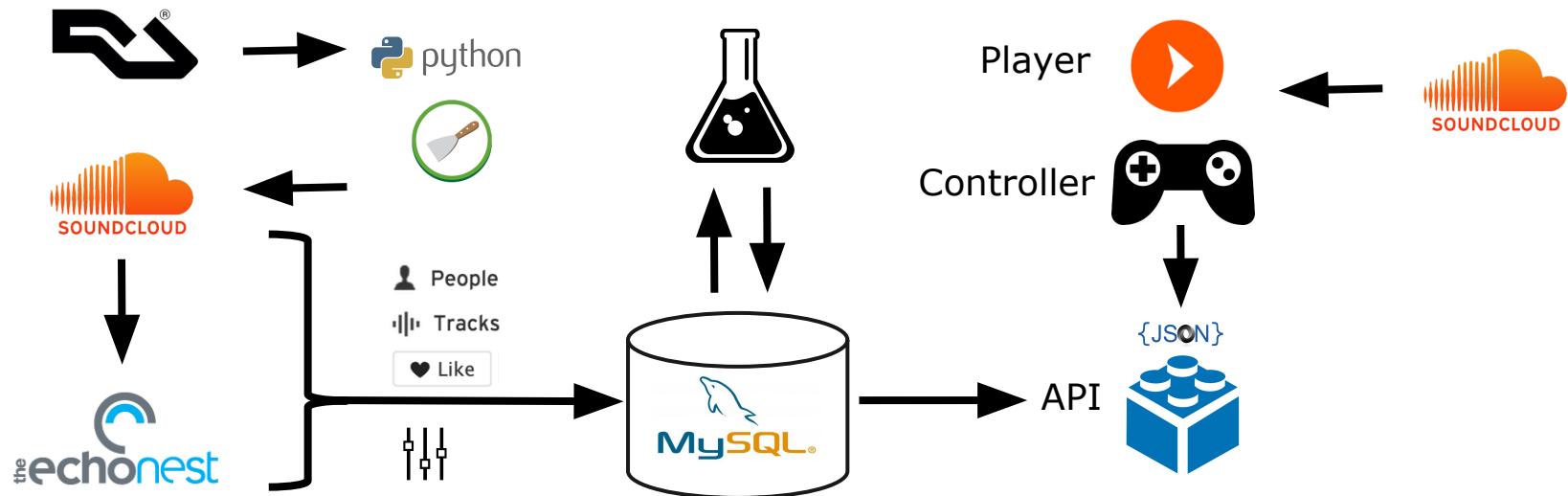
Agenda

1. Motivation
2. Approach & Data Engineering
3. Data Science:
 - Generating Recommendations
 - Measuring success
4. Launching the radio service
5. Demo

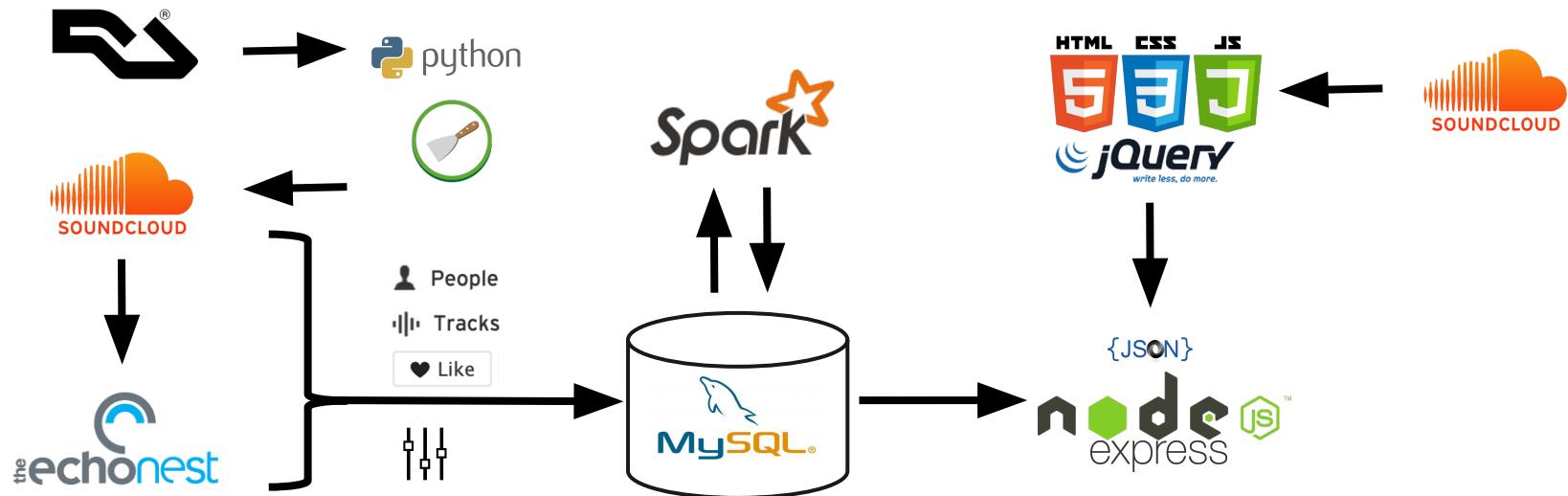
4. Launching the radio service Behind the scenes:



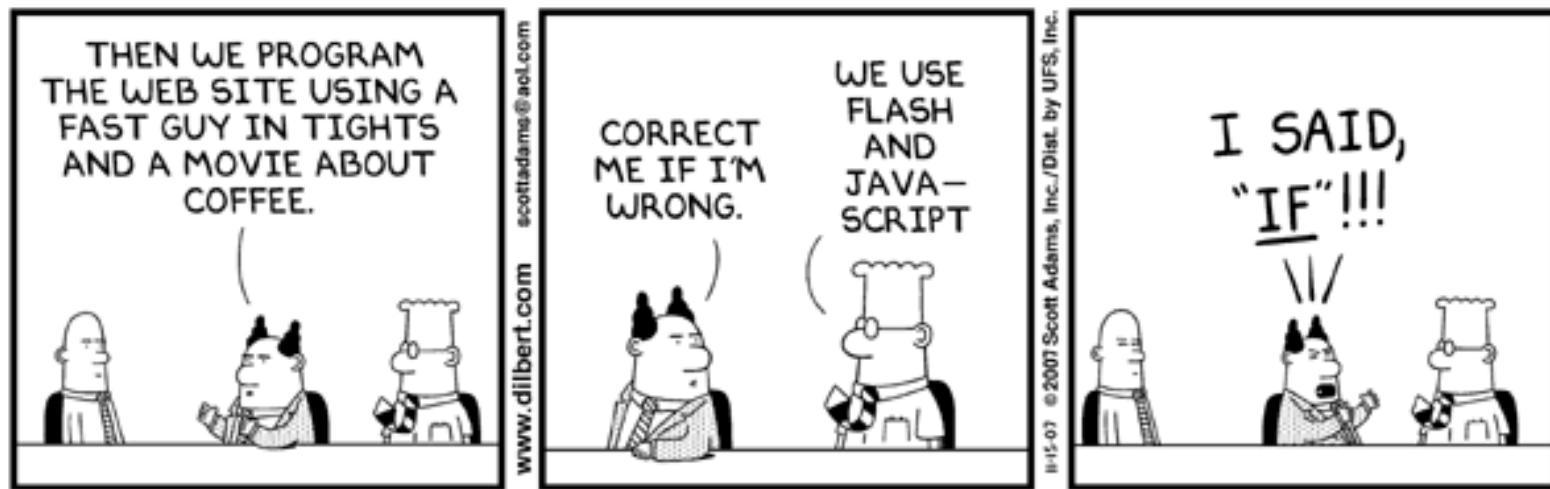
4. Launching the radio service Behind the scenes:



4. Launching the radio service Behind the scenes:



Demo



© Scott Adams, Inc./Dist. by UFS, Inc.

Project:

Personal local electronic music radio

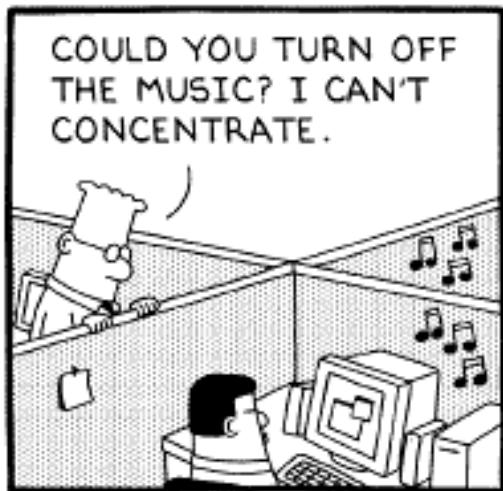
Scope :

- **Local:** Berlin
- **Radio:** Autoplay of generated playlists
- **Personal:** Responsive to feedback  
- **Music:** Focus on electronic music

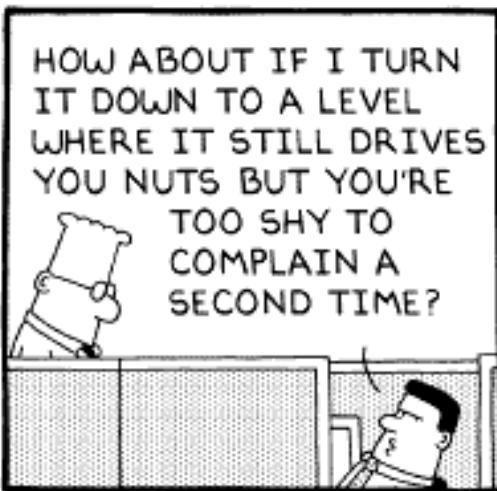
Motivation :

- Scoop into Berghain

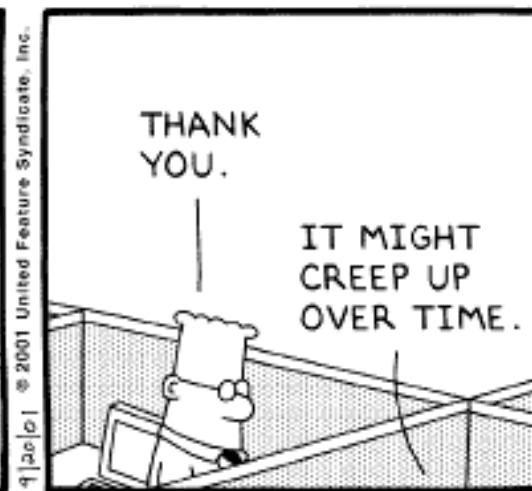
Demo



www.dilbert.com scottadams@aol.com



9/26/01 © 2001 United Feature Syndicate, Inc.



Copyright © 2001 United Feature Syndicate, Inc.

Johannes Braun



DATA SCIENCE RETREAT

Final project, 30/04/15

braun.johannes@gmail.com
linkedin.com/in/johannesbraun1
johannesbraun.github.io
github.com/johannesbraun



KU LEUVEN



```

def apk(actual, predicted, k=10):
    if len(predicted)>k:
        predicted = predicted[:k]

    score = 0.0
    num_hits = 0.0

    for i,p in enumerate(predicted):
        if p in actual and p not in predicted[:i]:
            num_hits += 1.0
            score += num_hits / (i+1.0)

    if not actual:
        return 1.0

    return score / min(len(actual), k)

def mapk(actual, predicted, k=10):
    return np.mean([apk(a,p,k) for a,p in zip(actual, predicted)])

```

visible

[166057101, 190972494, 74927480, 97610023, 181657050, 1]

`zip([actual], [predicted[0:10]])`

[([111221516, 162642747, 184927536, 112604679, 14309598

[117805508,

177358723,

111244653,

108807680,

142565321,

102378251,

168290694,

142738302,

145502026,

112604679])]

`mapk([actual], [predicted], 10)`

0.02

Mean Average Precision at rank k=10

2. Approach & Data Engineering: Extracting Audio Features



Danceability: Describes how suitable a track is for dancing using a number of musical elements: tempo, rhythm stability, beat strength, and overall regularity.



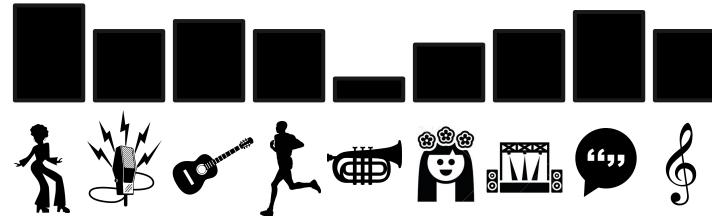
Energy: Represents a perceptual measure of intensity and powerful activity released throughout the track. Typical energetic tracks feel fast, loud, and noisy.



Acousticness: Represents the likelihood a recording was created by solely acoustic means such as voice and acoustic instruments as opposed to electronically such as with synthesized, amplified, or effected instruments.



Tempo: the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.



Instrumentalness: Represents a measure of how likely a song is to be all instrumental with no spoken words



Valence: Describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g., happy, cheerful, euphoric)



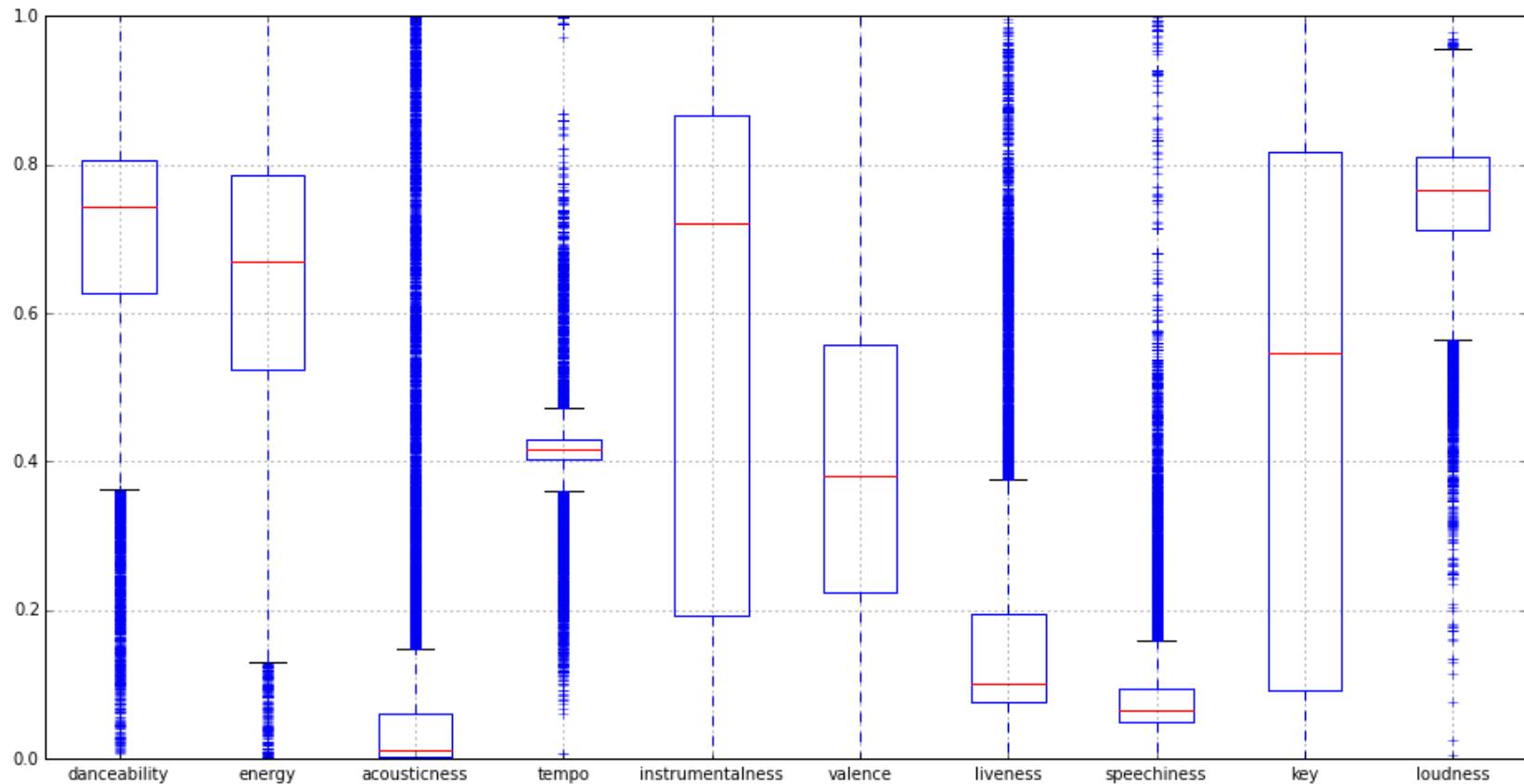
Liveness: Detects the presence of an audience in the recording.



Speechiness: Speechiness indicates the likelihood the track contains spoken words. Tracks with singing have low speechiness AND low instrumentalness, whereas rap tracks could have low instrumentalness and MEDIUM speechiness.



Key: the estimated overall key of a track. The key identifies the tonic triad, the chord, major or minor, which represents the final point of rest of a piece.



Distribution of Audio Features

3.4 Data Science: Extensions

- 1) Music math: jimpster - kalkbrenner = ??
- 2) Upload music track so it can be seen as combination of djs
- 3) Compare Spark ALS with
 - Vowpal Wabbit
 - MS Azure ML: Matchbox
- 4) Better hybrid approaches:
 - Rank aggregation
 - Mine comments
 - Mine echonest data in more depth
 - Integrate external knowledge of DJ labels etc.

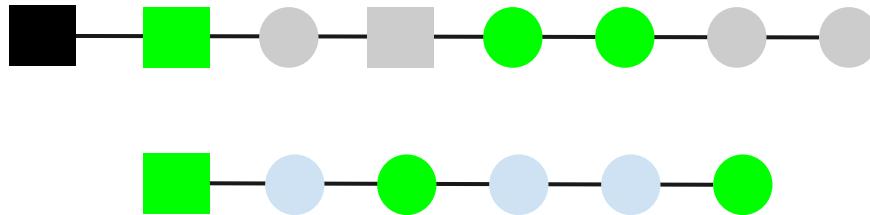
3.3 Data Science: Personalization

Update playlists with user feedback



3.3 Data Science: Personalization

Update playlists with user feedback



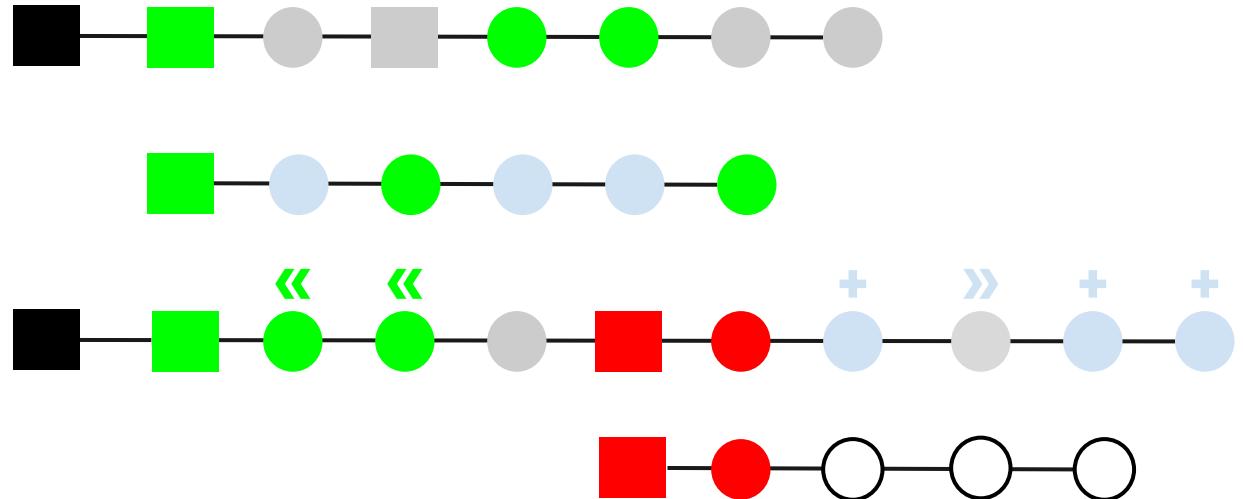
3.3 Data Science: Personalization

Update playlists with user feedback



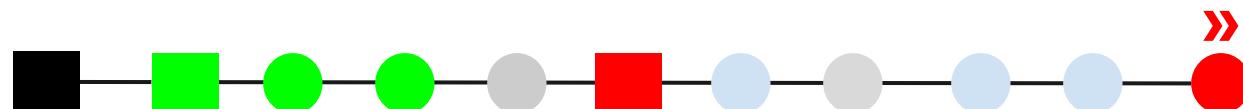
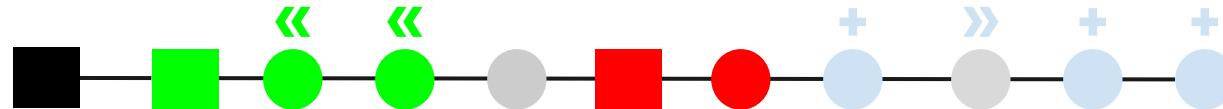
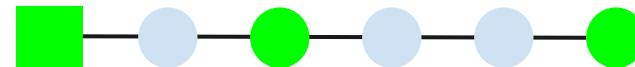
3.3 Data Science: Personalization

Update playlists with user feedback



3.3 Data Science: Personalization

Update playlists with user feedback



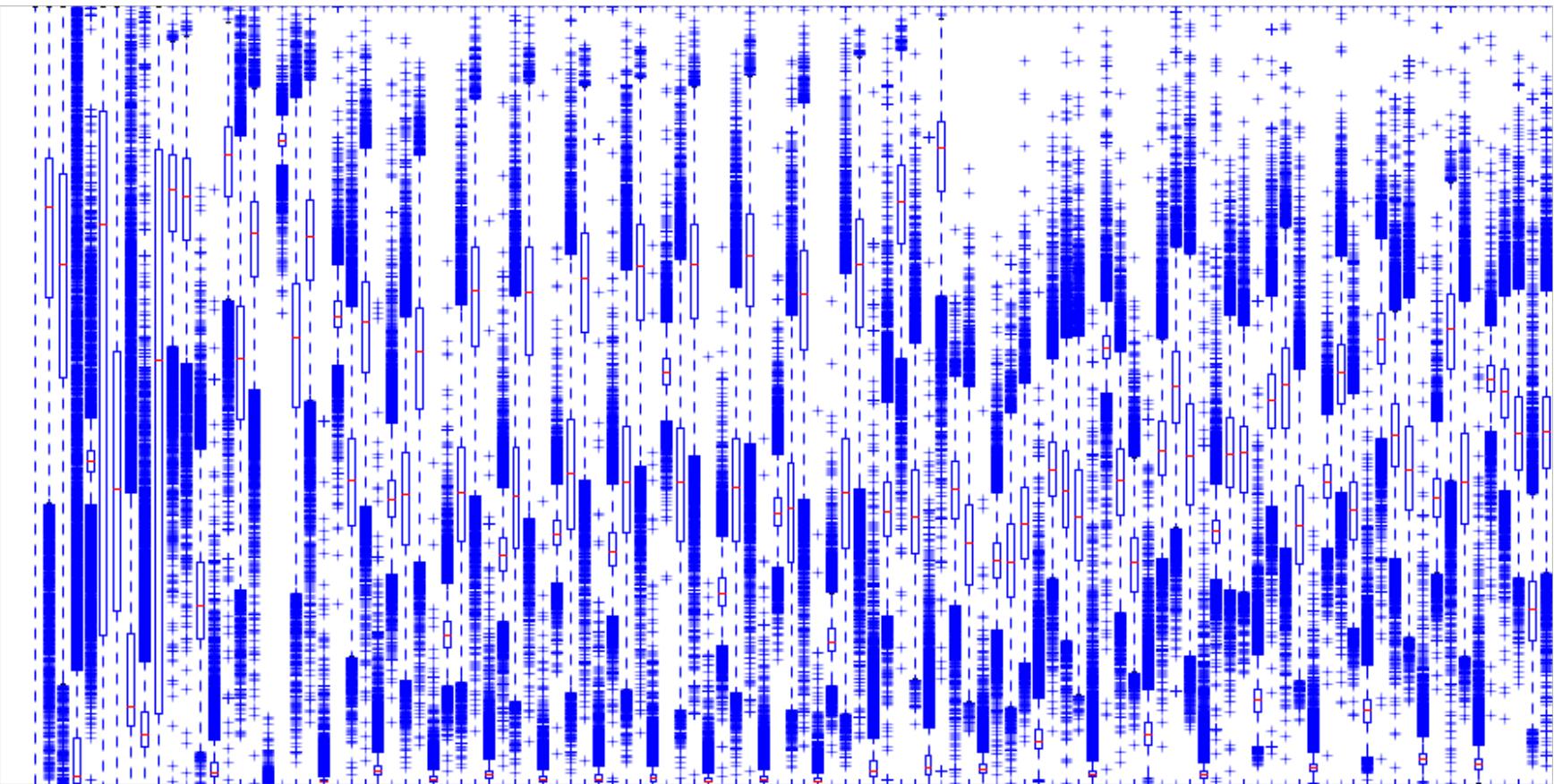
3.3 Data Science: Personalization

Update playlists with user feedback

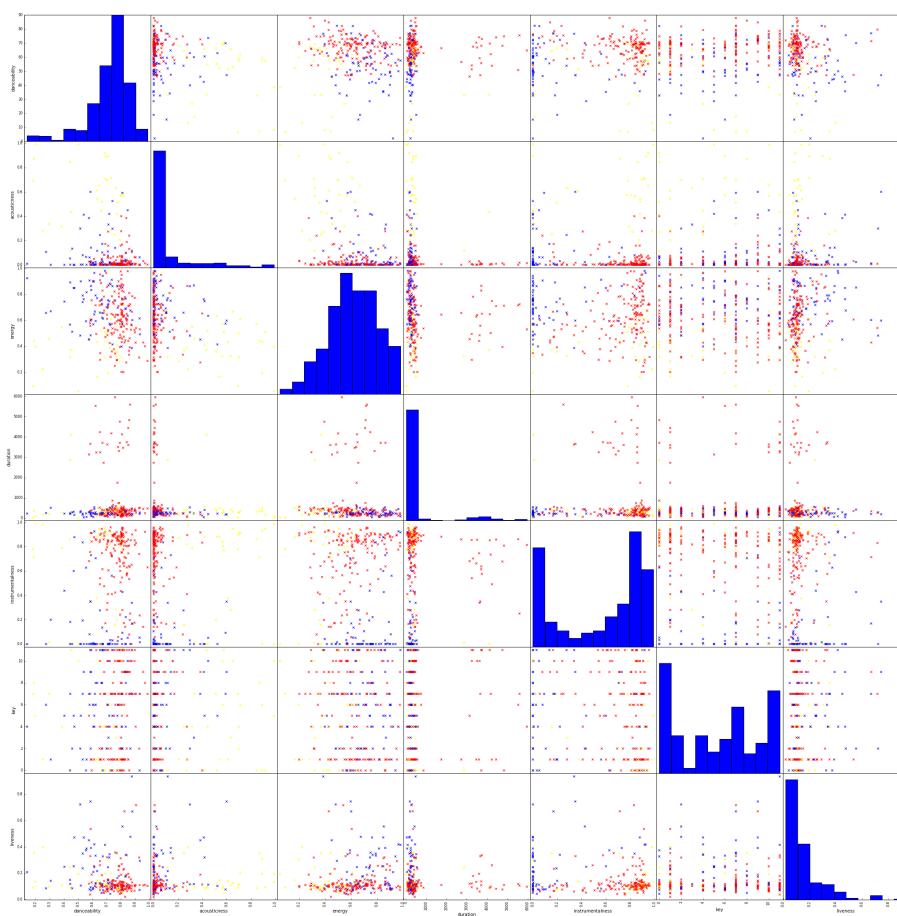
Naive update rule:

time decayed aggregated score of positive and negative recommendations

- LLR: neighborhood scores
- ALS: cosine similarities or aggregate user vector



Distribution of Audio Features incl. Timbre and Pitch features



Attempt to cluster songs by The Echo Nest Features

```
llr = function(k) {
  2 * sum(k) * (H(k) - H(rowSums(k)) - H(colSums(k)))
}
```

```
H = function(x) {
  N = sum(x) ;
  return (sum(x/N * log(x/N + (x==0))))
}
```

```
k = matrix(c(2,10,12,400),nrow=2)
k
```

$\begin{bmatrix} [1,] & [2,] & \text{colSums} \\ [1,] & k11: 2 & k12: 12 \quad 14 \\ [2,] & k22: 10 & k22: 400 \quad 410 \\ \text{rowSums} & 12 & 412 \quad 424 \end{bmatrix}$	$\begin{bmatrix} [1,] & [2,] & \text{colSums} \\ [1,] & k11: 2 & k12: 12 \quad 14 \\ [2,] & k22: 1 & k22: 400 \quad 401 \\ \text{rowSums} & 3 & 412 \quad 415 \end{bmatrix}$
--	--

```
llr(k)
> 3.704182
```

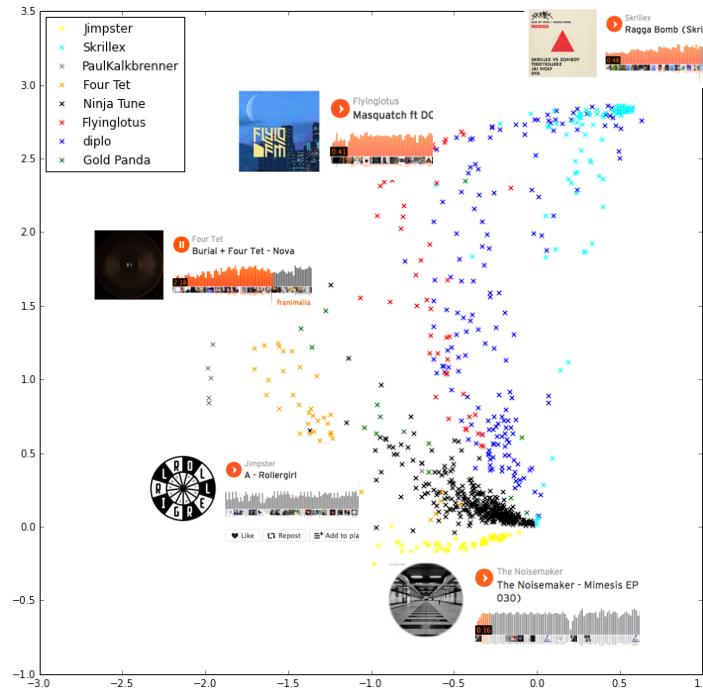
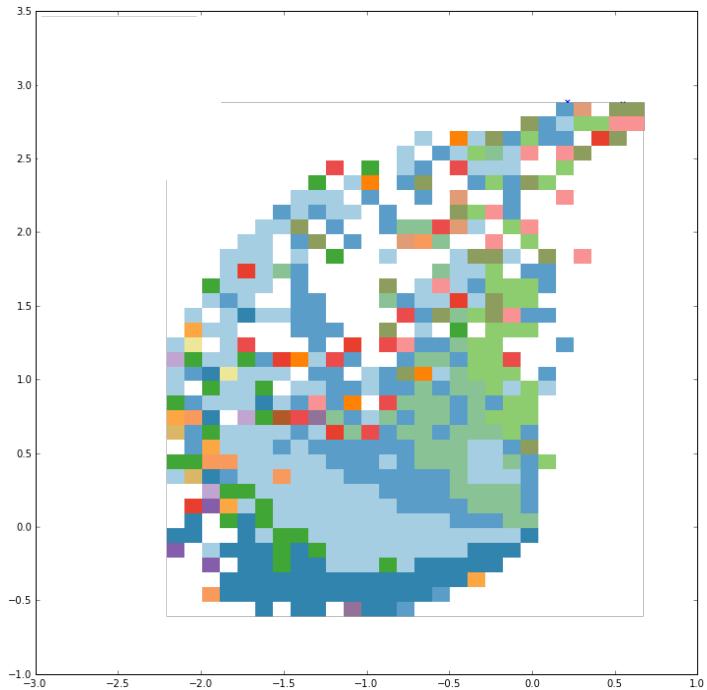
```
llr(k)
> 10.08757
```

	Song A	Everything but A
Song B	A and B together (k_11)	B, but not A (k_12)
Everything but B	A without B (k_21)	Neither A nor B (k_22)

Neighborhood approach: LLR

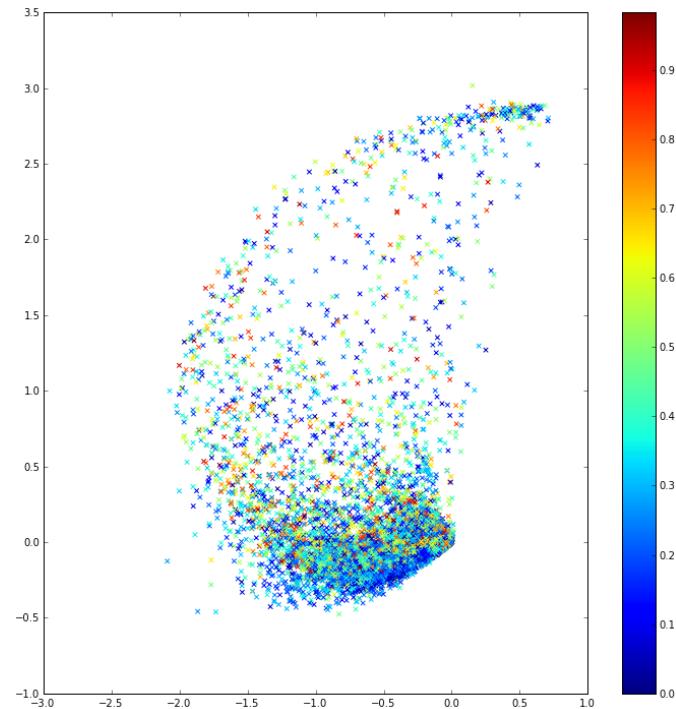
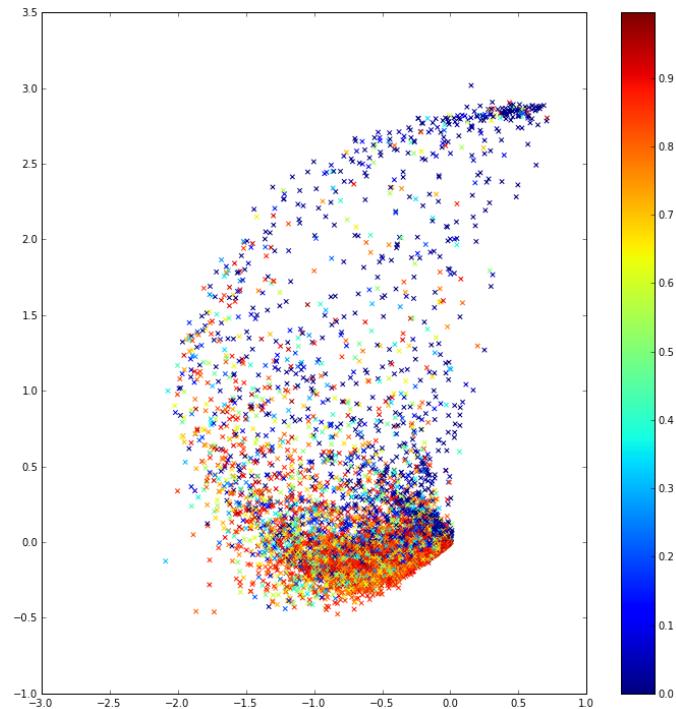
3.1 Data Science: Recommendations

b) Matrix factorization results: k=2



3.1 Data Science: Recommendations

c) Instrumentalness vs Valence



5.2 Truncated Mean Average Precision

Conformingly to the challenge, we used the truncated mAP (mean average precision) as the evaluation metric [10]. Let y denote a ranking over items, where $y(p) = i$ means that item i is ranked at position p . The mAP metric emphasizes the top recommendations. For any $k \leq N$, the *precision at k* (π_k) is defined as the proportion of correct recommendations within the top- k of the predicted ranking (assuming the ranking y does not contain the visible songs),

$$\pi_k(u, y) = \frac{1}{k} \sum_{p=1}^k r_{uy(p)}$$

For each user the (truncated) average precision is the average precision at each recall point:

$$AP(u, y) = \frac{1}{N_u} \sum_{p=1}^{N_u} \pi_k(u, y) r_{uy(p)}$$

where N_u is the smaller between N and the number of user u 's positively associated songs. Finally, the average of $AP(u, y_u)$'s over all users gives the mean average precision (mAP).

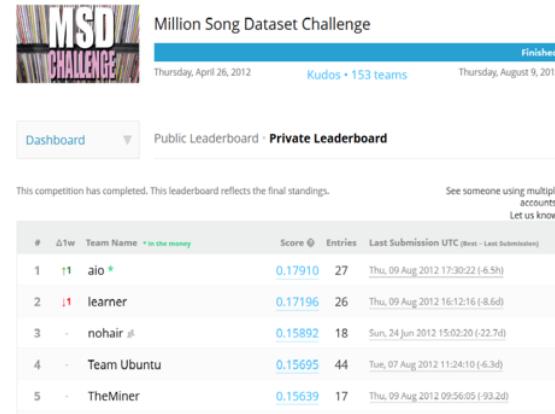


Figure 2: Screenshot of the final MSD challenge leaderboard.

Fabio Aiolfi, Winner of the Million Song Dataset challenge | Kaggle

7. Area under an ROC curve (AUC)

An ROC curve is a two-dimensional depiction of classifier performance. To compare classifiers we may want to reduce ROC performance to a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve, abbreviated AUC (Bradley, 1997; Hanley and McNeil, 1982). Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0. However, because random guessing produces the diagonal line between $(0,0)$ and $(1,1)$, which has an area of 0.5, no realistic classifier should have an AUC less than 0.5.

The AUC has an important statistical property: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. This is

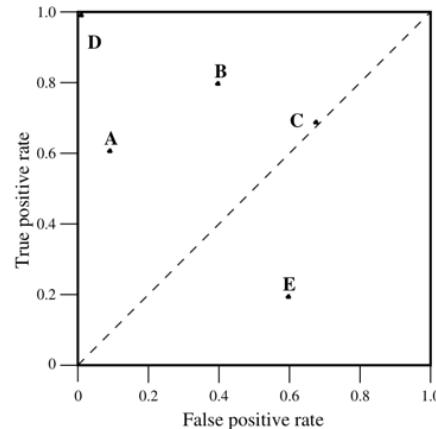


Fig. 2. A basic ROC graph showing five discrete classifiers.

Casting Recommendations as a Classification Problem

blasta.me



*Cartoon character "**Blaster**" from the Japanese animated series "Transformers: The Headmasters" (He transforms into a Ghetto blaster)*