



Search



Write



This member-only story is on us. [Upgrade](#) to access all of Medium.

♦ Member-only story

How Does Spotify Know You So Well?

A software engineer explains the science behind personalized music recommendations



Sophia Ciocca · [Follow](#)

9 min read · Oct 10, 2017

80K

187



...

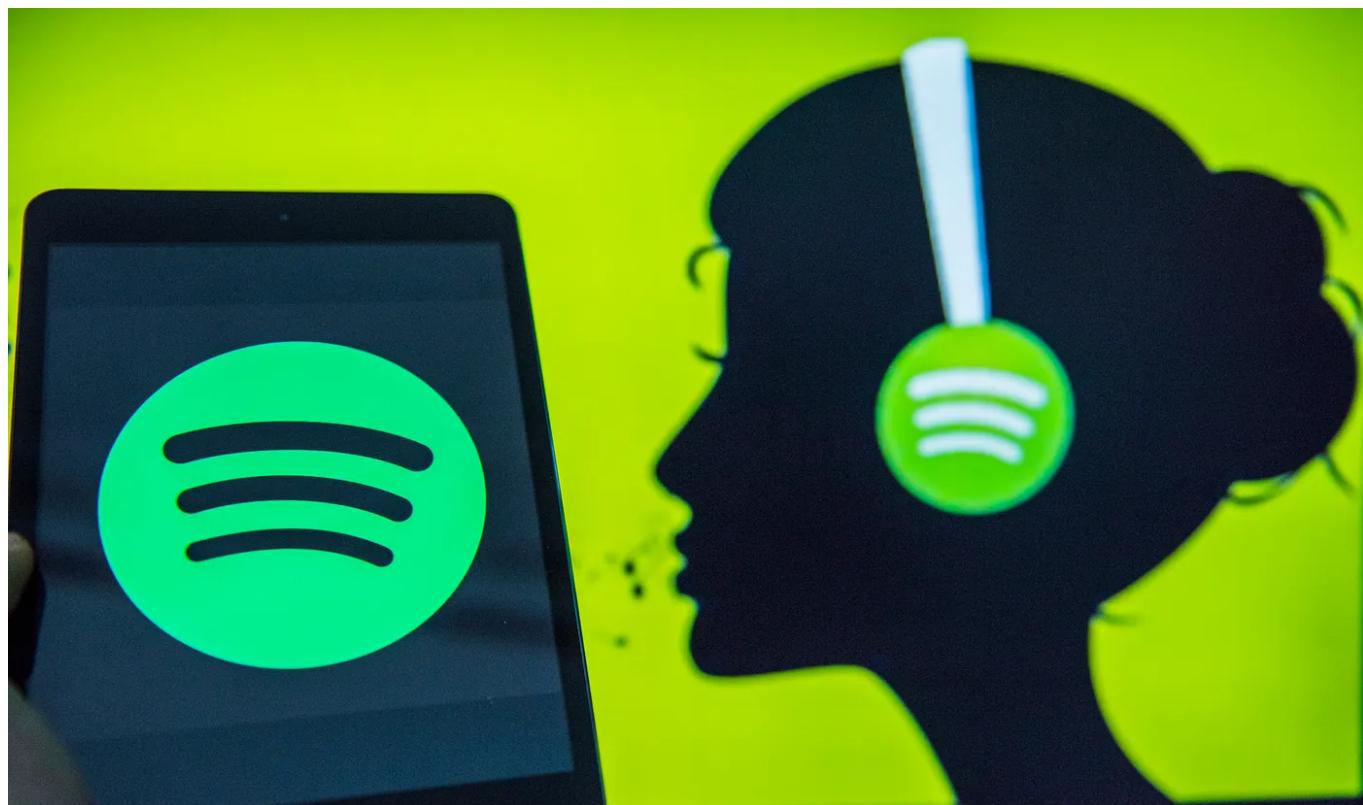


Photo by studioEAST/Getty Images

This Monday — just like every Monday before it — over 100 million Spotify users found a fresh new playlist waiting for them called Discover Weekly. It's a custom mixtape of 30 songs they've never listened to before but will probably love, and it's pretty much magic.

I'm a huge fan of Spotify, and particularly Discover Weekly. Why? It makes me feel *seen*. It knows my musical tastes better than any person in my entire life ever has, and I'm consistently delighted by how satisfyingly *just right* it is every week, with tracks I probably would never have found myself or known I would like.

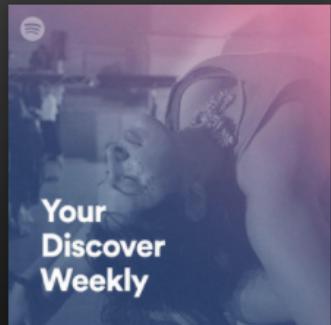
For those of you who live under a soundproof rock, let me introduce you to my virtual best friend:



Search



Sophia Ciocca



MADE FOR SOPHIA

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your favourites!

Made for Sophia Ciocca by Spotify • 30 songs, 2 hr 3 min

PLAY

FOLLOWING

...

FOLLOWER
1

Q Filter

Download

TITLE	ARTIST	ALBUM	
+ To Hugo	Clogs	The Creatures In Th...	2 days ago
+ Little Worlds	Mandolin Orange	Such Jubilee	2 days ago
+ Quiet Voices	Mike Vass	In the Wake of Neil ...	2 days ago
+ Sometimes	Goldmund	Sometimes	2 days ago
+ Sileo	Rhian Sheehan	Stories From Elsewh...	2 days ago
+ Hollow Home Rd	Brolly	Hollow Home Rd	2 days ago
+ Marigold	Mother Falcon	You Knew	2 days ago
+ Things Happen	Dawes	All Your Favorite Ba...	2 days ago
+ Sliding Down	Edgar Meyer, Mike ...	The Best of Edgar M...	2 days ago
+ Celeste	Pete Kuzma	Equilibrium	2 days ago

Fee +



3:43 ————— 5:49

A Spotify Discover Weekly playlist — specifically, mine.

As it turns out, I'm not alone in my obsession with Discover Weekly. The user base goes crazy for it, which has driven Spotify to rethink its focus, and invest more resources into algorithm-based playlists.

Ever since Discover Weekly debuted in 2015, I've been dying to know how it works (What's more, I'm a Spotify fangirl, so I sometimes like to pretend that I work there and research their products.) After three weeks of mad Googling, I feel like I've finally gotten a glimpse behind the curtain.

So how does Spotify do such an amazing job of choosing those 30 songs for each person each week? Let's zoom out for a second to look at how other music services have tackled music recommendations, and how Spotify's doing it better.

A Brief History of Online Music Curation



last.fm

Back in the 2000s, Songza kicked off the online music curation scene using **manual curation** to create playlists for users. This meant that a team of “music experts” or other human curators would put together playlists that they just thought sounded good, and then users would listen to those playlists. (Later, Beats Music would employ this same strategy.) Manual curation worked alright, but it was based on that specific curator’s choices, and therefore **couldn’t take into account each listener’s individual music taste**.

Like Songza, Pandora was also one of the original players in digital music curation. It employed a slightly more advanced approach, instead **manually tagging attributes** of songs. This meant a group of people listened to music, chose a bunch of descriptive words for each track, and tagged the tracks accordingly. Then, Pandora’s code could simply filter for certain tags to make playlists of similar-sounding music.

Around that same time, a music intelligence agency from the MIT Media Lab called The Echo Nest was born, which took a radical, cutting-edge approach

to personalized music. The Echo Nest used algorithms to analyze the audio and textual content of music, allowing it to perform music identification, personalized recommendation, playlist creation, and analysis.

Finally, taking another approach is Last.fm, which still exists today and uses a process called **collaborative filtering** to identify music its users might like, but more on that in a moment.

So if that's how other music curation services have handled recommendations, how does Spotify's magic engine run? How does it seem to nail individual users' tastes so much more accurately than any of the other services?

Spotify's Three Types of Recommendation Models

Spotify doesn't actually use a single revolutionary recommendation model. Instead, they mix together some of the best strategies used by other services to create their own uniquely powerful discovery engine.

To create Discover Weekly, there are three main types of recommendation models that Spotify employs:

1. **Collaborative Filtering** models (i.e. the ones that Last.fm originally used), which analyze both *your* behavior and *others'* behaviors.
2. **Natural Language Processing (NLP)** models, which analyze *text*.
3. **Audio** models, which analyze the *raw audio tracks themselves*.

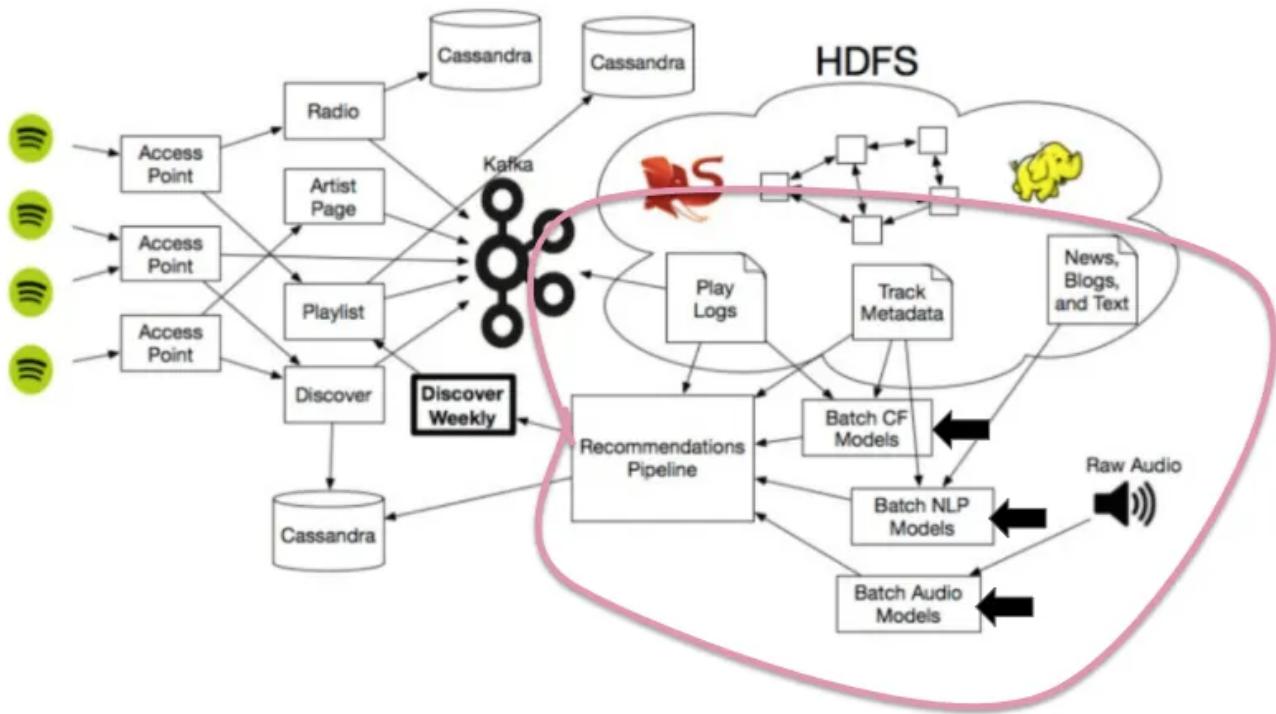


Image source: [Ever Wonder How Spotify Discover Weekly Works? Data Science](#), via Galvanize.

Let's dive into how each of these recommendation models work!

Recommendation Model #1: Collaborative Filtering



First, some background: When people hear the words “collaborative filtering,” they generally think of Netflix, as it was one of the first companies to use this method to power a recommendation model, taking users’ star-based movie ratings to inform its understanding of which movies to recommend to other similar users.

After Netflix was successful, the use of collaborative filtering spread quickly, and is now often the starting point for anyone trying to make a recommendation model.

Unlike Netflix, Spotify doesn't have a star-based system with which users rate their music. Instead, Spotify's data is **implicit feedback** — specifically, the **stream counts** of the tracks and additional streaming data, such as whether a user saved the track to their own playlist, or visited the artist's page after listening to a song.

But what *is* collaborative filtering, truly, and how does it work? Here's a high-level rundown, explained in a quick conversation:



Image source: [Collaborative Filtering at Spotify](#), by Erik Bernhardsson, ex-Spotify.

What's going on here? Each of these individuals has track preferences: the one on the left likes tracks P, Q, R, and S, while the one on the right likes

tracks Q, R, S, and T.

Collaborative filtering then uses that data to say:

“Hmmm... You both like three of the same tracks – Q, R, and S – so you are probably similar users. Therefore, you’re each likely to enjoy other tracks that the other person has listened to, that you haven’t heard yet.”

Therefore, it suggests that the one on the right check out track P – the only track not mentioned, but that his “similar” counterpart enjoyed – and the one on the left check out track T, for the same reasoning. Simple, right?

But how does Spotify actually use that concept in practice to calculate millions of users’ suggested tracks based on millions of other users’ preferences?

With matrix math, done with Python libraries!

$$\begin{matrix} & \text{Users} \\ \left(\begin{array}{cccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right) \\ & \text{Songs} \end{matrix}$$

In actuality, this matrix you see here is *gigantic*. Each row represents one of Spotify’s 140 million users – if you use Spotify, you yourself are a row in this matrix – and each column represents one of the 30 million songs in Spotify’s database.

Then, the Python library runs this long, complicated matrix factorization formula:

$$\min_{x,y} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i - \beta_u - \beta_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

Some complicated math...

When it finishes, we end up with two types of vectors, represented here by X and Y. **X is a *user vector***, representing one single user's taste, and **Y is a *song vector***, representing one single song's profile.

$$\text{Users} \begin{pmatrix} 10001001 \\ 00100100 \\ 10100011 \\ 01000100 \\ 00100100 \\ 10001001 \end{pmatrix} \approx \underbrace{\begin{pmatrix} x \\ x \\ x \\ x \\ x \\ x \end{pmatrix}}_{\text{user vector}} \underbrace{\left(\begin{array}{c|c} Y & Y \\ Y & Y \end{array} \right) \{ f \}}_{\text{song vector}}$$

The User/Song matrix produces two types of vectors: user vectors and song vectors. Image source: [From Idea to Execution: Spotify's Discover Weekly](#), by Chris Johnson, ex-Spotify.

Now we have 140 million user vectors and 30 million song vectors. The actual content of these vectors is just a bunch of numbers that are essentially meaningless on their own, but are hugely useful when compared.

To find out which users' musical tastes are most similar to mine, collaborative filtering compares my vector with all of the other users' vectors, ultimately spitting out which users are the closest matches. The same goes for the Y vector, *songs*: you can compare a single song's vector with

all the others, and find out which songs are most similar to the one in question.

Collaborative filtering does a pretty good job, but Spotify knew they could do even better by adding another engine. Enter NLP.

Recommendation Model #2: Natural Language Processing (NLP)

The second type of recommendation models that Spotify employs are **Natural Language Processing (NLP) models**. The source data for these models, as the name suggests, are regular ol' words: track metadata, news articles, blogs, and other text around the internet.



Natural Language Processing, which is the ability of a computer to understand human speech as it is spoken, is a vast field unto itself, often harnessed through sentiment analysis APIs.

The exact mechanisms behind NLP are beyond the scope of this article, but here's what happens on a very high level: Spotify crawls the web constantly looking for blog posts and other written text about music to figure out what people are saying about specific artists and songs — which adjectives and what particular language is frequently used in reference to those artists and songs, and which other artists and songs are also being discussed alongside them.

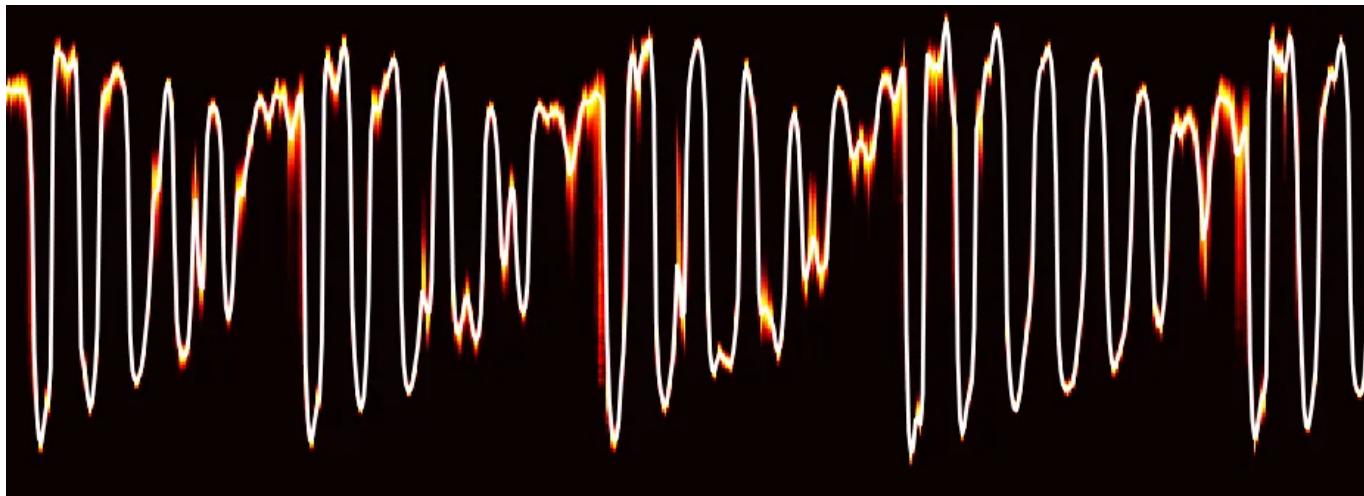
While I don't know the specifics of how Spotify chooses to then process this scraped data, I can offer some insight based on how the Echo Nest used to work with them. They would bucket Spotify's data up into what they call "cultural vectors" or "top terms." Each artist and song had thousands of top terms that changed on the daily. Each term had an associated weight, which correlated to its relative importance — roughly, the probability that someone will describe the music or artist with that term.

n2 Term	Score	np Term	Score	adj Term	Score
dancing queen	0.0707	dancing queen	0.0875	perky	0.8157
mamma mia	0.0622	mamma mia	0.0553	nonviolent	0.7178
disco era	0.0346	benny	0.0399	swedish	0.2991
winner takes	0.0307	chess	0.0390	international	0.2010
chance on	0.0297	its chorus	0.0389	inner	0.1776
swedish pop	0.0296	vous	0.0382	consistent	0.1508
my my	0.0290	the invitations	0.0377	bitter	0.0871
s enduring	0.0287	voulez	0.0377	classified	0.0735
and gimme	0.0280	something's	0.0374	junior	0.0664
enduring appeal	0.0280	priscilla	0.0369	produced	0.0616

"Cultural vectors" or "top terms," as used by the Echo Nest. Image source: [How music recommendation works — and doesn't work](#), by Brian Whitman, co-founder of The Echo Nest.

Then, much like in collaborative filtering, the NLP model uses these terms and weights to create a vector representation of the song that can be used to determine if two pieces of music are similar. Cool, right?

Recommendation Model #3: Raw Audio Models



First, a question. You might be thinking:

Sophia, we already have so much data from the first two models! Why do we need to analyze the audio itself, too?

First of all, adding a third model further improves the accuracy of the music recommendation service. But this model also serves a secondary purpose: unlike the first two types, **raw audio models take *new songs* into account**.

Take, for example, a song your singer-songwriter friend has put up on Spotify. Maybe it only has 50 listens, so there are few other listeners to collaboratively filter it against. It also isn't mentioned anywhere on the internet yet, so NLP models won't pick it up. Luckily, raw audio models don't discriminate between new tracks and popular tracks, so with their help, your friend's song could end up in a Discover Weekly playlist alongside popular songs!

But how can we analyze *raw audio data*, which seems so abstract?

With convolutional neural networks!

Convolutional neural networks are the same technology used in facial recognition software. In Spotify's case, they've been modified for use on audio data instead of pixels. Here's an example of a neural network architecture:

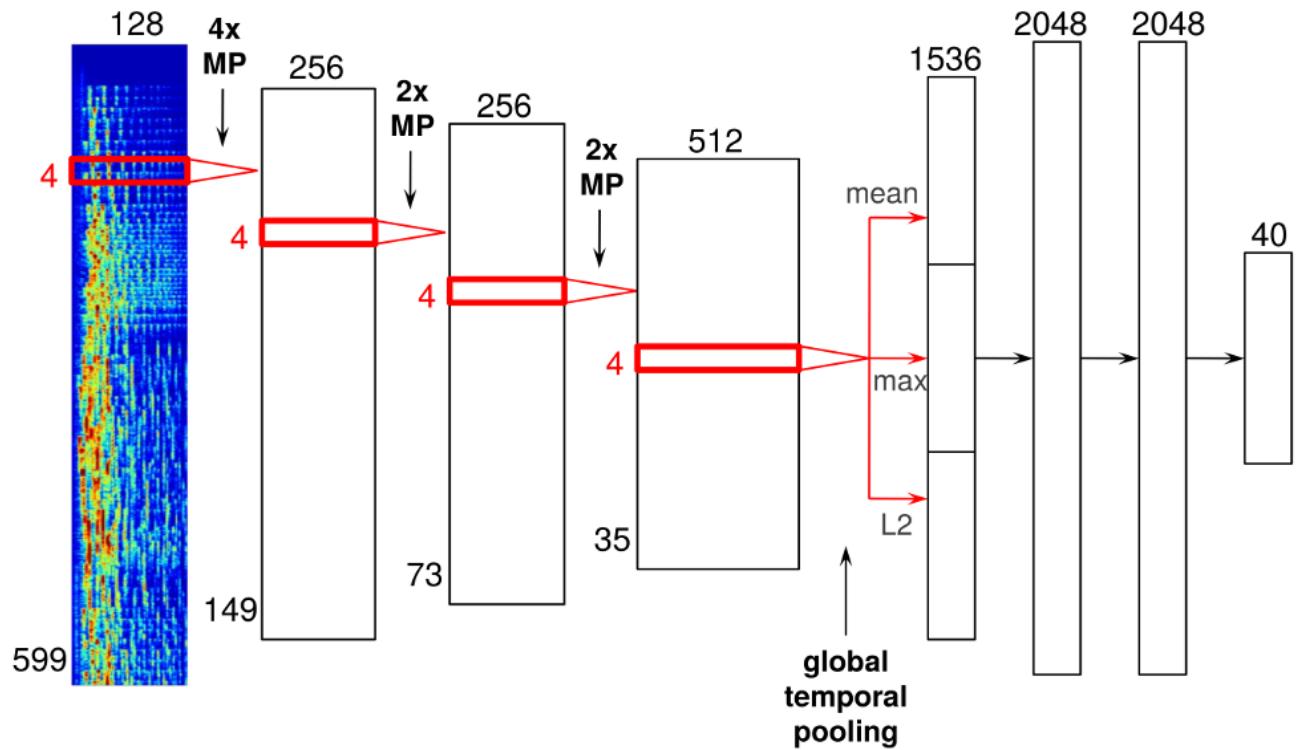


Image source: [Recommending music on Spotify with deep learning](#), Sander Dieleman.

This particular neural network has four *convolutional layers*, seen as the thick bars on the left, and three dense layers, seen as the more narrow bars on the right. The inputs are time-frequency representations of audio frames, which are then concatenated, or linked together, to form the spectrogram.

The audio frames go through these convolutional layers, and after passing through the last one, you can see a “global temporal pooling” layer, which pools across the entire time axis, effectively computing statistics of the learned features across the time of the song.

After processing, the neural network spits out an understanding of the song, including characteristics like estimated time signature, key, mode, tempo, and loudness. Below is a plot of data for a 30-second snippet of “Around the World” by Daft Punk.

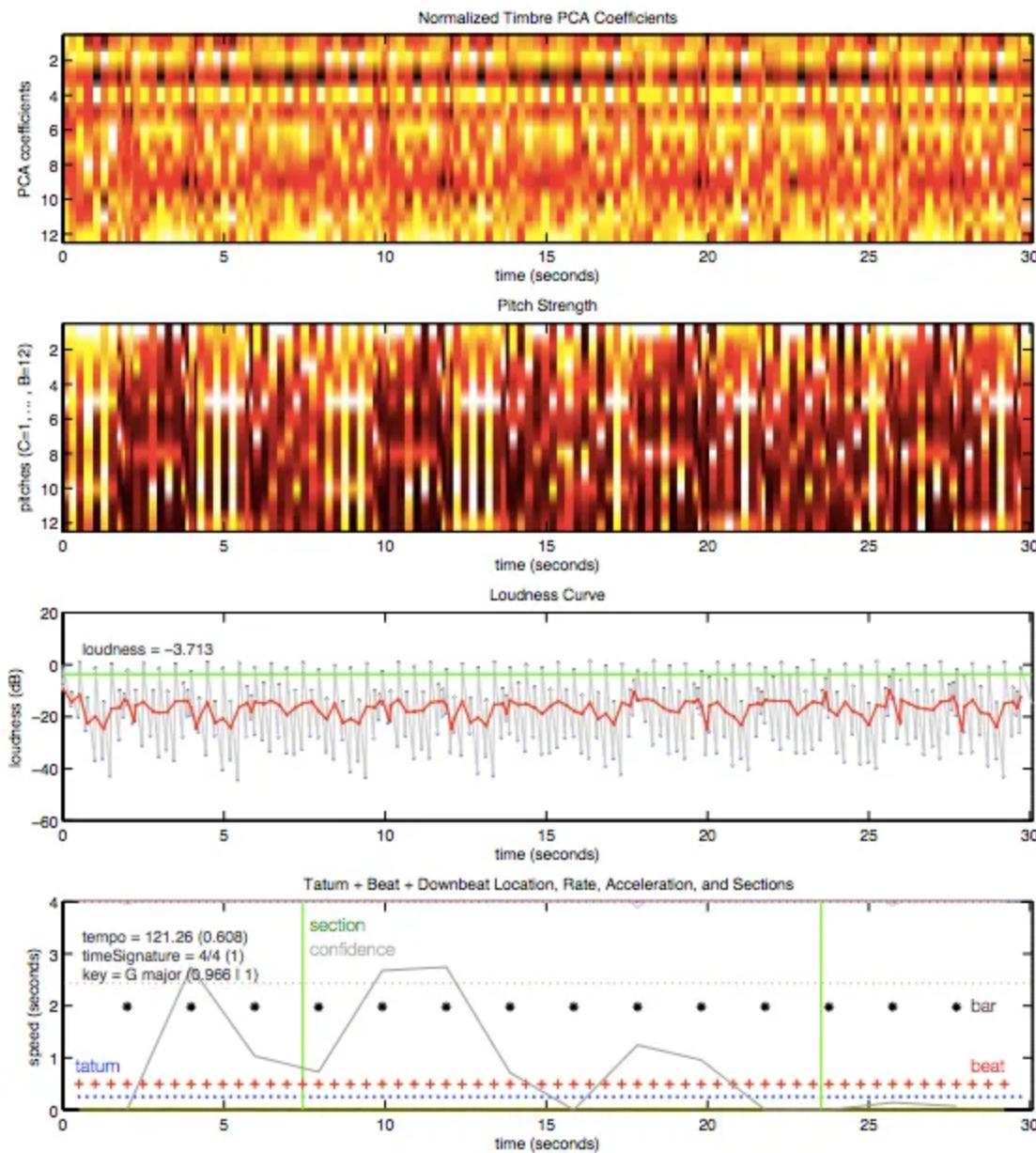
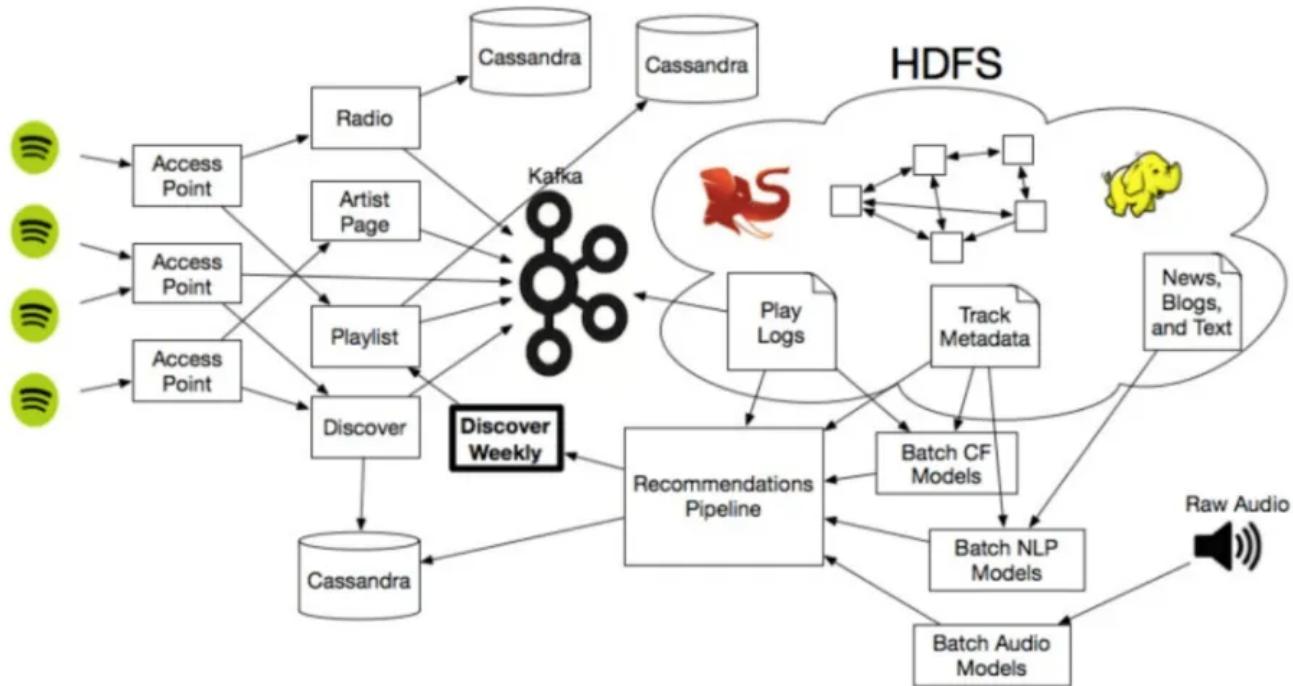


Image source: [Tristan Jehan & David DesRoches, via The Echo Nest](#).

Ultimately, this reading of the song’s key characteristics allows Spotify to understand fundamental similarities between songs and therefore which users might enjoy them, based on their own listening history.

That covers the basics of the three major types of recommendation models feeding Spotify's Recommendations Pipeline, and ultimately powering the Discover Weekly playlist!



Of course, these recommendation models are all connected to Spotify's larger ecosystem, which includes giant amounts of data storage and uses *lots of Hadoop clusters* to scale recommendations and make these engines work on enormous matrices, endless online music articles, and huge numbers of audio files.

I hope this was informative and piqued your curiosity like it did mine. For now, I'll be working my way through my own Discover Weekly, finding my new favorite music while appreciating all the machine learning that's going on behind the scenes. ♪

Note (March 2020): I wrote this article nearly three years ago, so information may be outdated. It was merely what I found publicly on the internet and pieced together at the time.

Thanks also to [ladycollective](#) for reading this article and suggesting edits.

Music

Spotify

Machine Learning

Tech

Artificial Intelligence



Written by Sophia Ciocca

10.9K Followers

Warrior for authenticity. Uncovering my truest self & documenting the journey.

<http://sophiaciocca.com>

Follow



More from Sophia Ciocca



 Sophia Ciocca in NYT Open

Building a Text Editor for a Digital-First Newsroom

An inside look at the inner workings of a technology you may take for granted

10 min read · Apr 12, 2018

 14.9K  45



 Sophia Ciocca in Personal Growth

F*ck impostor syndrome — I'm finally learning to code.

I'm really doing it: Starting next week, I will be a student at the Grace Hopper Program, an...

8 min read · Apr 10, 2017

 1.3K  38

 Sophia Ciocca in Personal Growth

2016: The Year I Burned Down and Began to Rebuild Myself

This year, y'all. This year.

6 min read · Dec 30, 2016

 1K  28



 Sophia Ciocca

The key to curing binge eating isn't what you think

How I achieved freedom around food, for good.

9 min read · Jul 15, 2020

 353 

See all from Sophia Ciocca

Recommended from Medium



 Cassie Kozyrkov

Why I quit my job as Google's Chief Decision Scientist

What's it like to go from being Chief Decision Scientist at Google to being, well... just me?

★ · 7 min read · Oct 28

 6.6K  92

 Zach Qui... in Pipeline: Your Data Engineering Res...

3 Data Science Projects That Got Me 12 Interviews. And 1 That Got...

3 work samples that got my foot in the door, and 1 that almost got me tossed out.

★ · 7 min read · Aug 29, 2022

 3.7K  49

Lists



Predictive Modeling w/ Python

20 stories · 562 saves



Natural Language Processing

800 stories · 367 saves



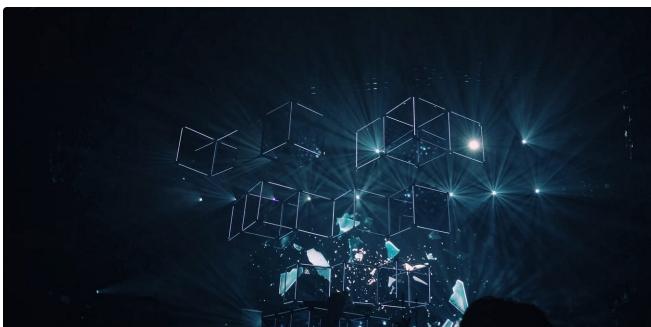
AI Regulation

6 stories · 175 saves



ChatGPT prompts

27 stories · 609 saves



 Virat Patel

 Barack Obama 

I applied to 230 Data science jobs during last 2 months and this is...

A little bit about myself: I have been working as a Data Analyst for a little over 2 years....

◆ · 3 min read · Aug 11

👏 2.3K

💬 48



 Ashley Mayer

We need more women founders on offense

A response to Glossy from Glossier's former comms lead

5 min read · 5 days ago

👏 3.3K

💬 83



What I'm Reading on the Rise of Artificial Intelligence

Earlier this week, President Biden issued an executive order on artificial intelligence—a...

2 min read · 3 days ago

👏 7.2K

💬 148



 MargaretEfron in Learning Data

5 Things I Wish I Knew Before My First Job as a Data Analyst

I'm five months into my first Data Analyst job, and there are many surprises I wish someone...

5 min read · Aug 23

👏 1.5K

💬 36



[See more recommendations](#)