IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

EXCALIBUR IP, LLC **Plaintiff,** **v.** **SPOTIFY AB, and SPOTIFY USA, Inc.** **Defendants.**	**Civil Action No. _____** **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Excalibur IP, LLC ("Excalibur" or "Plaintiff") files this Complaint for patent infringement against Defendants Spotify AB and Spotify USA, Inc. (collectively "Spotify") for infringement of U.S. Patent No. 8,392,148 ("the '148 Patent"), U.S. Patent No. 8,160,840 ("the '840 Patent"), U.S. Patent No. 7,454,509 ("the '509 Patent"), and U.S. Patent No. 8,352,331 ("the '331 Patent") (collectively the "Patents-in-Suit"), pursuant to 35 U.S.C. § 271 (copies of the Patents-in-Suit are attached as Exhibits A, B, C, and D), and alleges the following:

I. NATURE OF THE ACTION

1. This is an action for patent infringement arising under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq*.

II. PARTIES

2. Plaintiff Excalibur is a limited liability company organized and existing under laws of the State of Delaware.

3. Upon information and belief, Defendant Spotify AB is a corporation (*Aktiebolag*), registration no. 556703-7485, organized under the laws of Sweden, with its headquarters and principal place of business in Sweden, located at Reeegeringsgatan 19, SE-111 53 Stockholm, Sweden. Spotify AB may be served by mail or by process under the Hague Convention. *See Water Splash, Inc. v. Menon*, 137 S. Ct. 1504 (2017). Upon information and belief, Spotify AB makes, sells and offers to sell products and services throughout the world including the United States, and introduces products and services into the stream of commerce and that incorporate infringing technology knowing that they would be made, used and sold in this judicial district and elsewhere in the United States in an infringing manner.

4. Upon information and belief, Spotify USA, Inc., is a corporation organized and existing under the laws of Delaware, with a place of business at 45 West 18th St., 7th Floor, New York, NY 10011, and can be served through its Delaware registered agent, National Registered Agents, Inc., 160 Greentree Dr., Suite 101, Dover, DE 19904. Upon information and belief, Spotify USA, Inc., makes, sells and offers to sell products and services throughout the United States, including in this judicial district, and introduces products and services into the stream of commerce and that incorporate infringing technology knowing that they would be made, used and sold in this judicial district and elsewhere in the United States in an infringing manner.

III. JURISDICTION AND VENUE

5. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §271.

6. This Court has subject matter jurisdiction of such federal question claims pursuant to 28 U.S.C. §§ 1331 and 1338(a).

7. Venue is proper in this judicial district under 28 U.S.C. §§ 1391 and/or 1400(b). On information and belief, Spotify USA, Inc. is incorporated in the State of Delaware. Venue is proper against Spotify AB because suits against foreign entities are proper in any judicial district.

8. On information and belief, each Defendant is subject to this Court's general and specific personal jurisdiction because each Defendant has sufficient minimum contacts within the State of Delaware and this District, pursuant to due process and/or the Delaware Long Arm Statute (10 Del. Code § 3104), because each Defendant purposefully availed itself of the privileges of conducting business in the State of Delaware and in this District, because each Defendant regularly conducts and solicits business within the State of Delaware and within this District, and because Plaintiff's causes of action arise directly from each Defendant's business contacts and other activities in the State of Delaware and this District. Further, this Court has personal jurisdiction over Spotify USA, Inc., because it is incorporated in Delaware and has purposely availed itself of the privileges and benefits of the laws of the State of Delaware.

9. In other patent infringement actions, such as *CRFD Research, Inc. v. Spotify USA Inc., et al.*, No. 1:14-cv-315 (D. Del.), Spotify has admitted that it is subject to personal jurisdiction in this Court.

IV. THE ASSERTED PATENTS

10. On March 5, 2013, the United States Patent and Trademark Office issued U.S. Patent No. 8,392,148 ("the '148 Patent") to Caruso et al., entitled "Comparison of data signals using characteristic electronic thumbprints extracted therefrom," after full and fair examination. Plaintiff is the assignee of all rights, title, and interest in and to the '148 Patent and possesses all rights of recovery under the '148 Patent, including the right to recover damages for present, past, and future infringement. A true and correct copy of the '148 Patent is attached as Exhibit A. The '148 Patent is valid and enforceable.

11. On April 17, 2012, the United States Patent and Trademark Office issued U.S. Patent No. 8,160,840 ("the '840 Patent") to Caruso et al., entitled "Comparison of data signals using characteristic electronic thumbprints extracted therefrom," after full and fair examination. Plaintiff is the assignee of all rights, title, and interest in and to the '840 Patent and possesses all rights of recovery under the '840 Patent, including the right to recover damages for present, past, and future infringement. A true and correct copy of the '840 Patent is attached as Exhibit B. The '840 Patent is valid and enforceable.

12. On November 18, 2008, the United States Patent and Trademark Office issued U.S. Patent No. 7,454,509 ("the '509 Patent") to Boulter et al., entitled "Online playback system with community bias," after full and fair examination. Plaintiff is the assignee of all rights, title, and interest in and to the '509 Patent and possesses all rights of recovery under the '509 Patent, including the right to recover damages for present, past, and future infringement. A true and correct copy of the '509 Patent is attached as Exhibit C. The '509 Patent is valid and enforceable.

13. On January 8, 2013, the United States Patent and Trademark Office issued U.S. Patent No. 8,352,331 ("the '331 Patent") to Dunning et al., entitled "Relationship discovery engine," after full and fair examination. Plaintiff is the assignee of all rights, title, and interest in and to the '331 Patent and possesses all rights of recovery under the '331 Patent, including the right to recover damages for present, past, and future infringement. A true and correct copy of the '331 Patent is attached as Exhibit D. The '331 Patent is valid and enforceable.

14. The inventions of the Patents-in-Suit were not well-understood, routine, or conventional at the time of the inventions.

15. The inventions of the ’148 and ’840 patents resolve technical problems related to the extraction of a characteristic thumbprint from a data signal, such as an audio data file, and further to the comparison or matching of such thumbprints. The claims of the ’148 and ’840 patents do not merely recite the performance of some business practice known from the pre-computer world along with the requirement to perform it on a computer. Instead, the claims of the ’148 and ’840 patents recite one or more inventive concepts that are rooted in computerized technology, and overcome problems specifically arising in that realm.

16. The ’148 and ’840 patents relate generally to the extraction of a characteristic thumbprint from a data signal, such as an audio data file, and further to the comparison or matching of such thumbprints. ’148 Patent at 1:21-24. Because of the variations in file formats, compression technologies, and other methods of representing data, the problem of identifying a data signal or comparing it to others raises significant technical difficulties. *Id*. at 1:25-28. Prior art techniques often identified and tracked media by attaching metadata, watermarks, or some other code that contains identification information for the media. *Id*. at 1:38-41. However, this attached information is often incomplete, incorrect, or both. *Id*. at 1:41-43. At the time of the invention, a need existed to efficiently identify a data signal and compare it to other signals. *Id*. at 2:12-14.

17. The inventive solutions of the claimed inventions of the ’148 and ’840 patents include methods and apparatus to extract a characteristic thumbprint from a data signal based on the content of that signal. *Id*. at 2:18-20. This thumbprint can be matched against a set of reference thumbprints to determine the identity of the signal or the similarity between two signals. *Id*. at 2:20-22. These inventive solutions overcome one or more problems of the prior

art. As detailed by the specification, the prior techniques suffered drawbacks such that a new and novel solution was required.

18. A person of ordinary skill in the art reading the '148 and '840 patents and their claims would understand that the patent's disclosure and claims are drawn to solving a specific, technical problem arising in the identification or comparison of data signals. Moreover, a person of ordinary skill in the art would understand that the claims' subject matter presents advancements in the field of digital signal processing. The claims do not preempt all types of digital signal processing. For example, the claims do not preempt use of the techniques taught in the prior art cited on the face of the '148 and '840 patents. Instead, the claims recite the specific novel solution taught in the specification generally relating to the characteristic thumbprint from a data signal.

19. In light of the foregoing, a person of ordinary skill in the art would understand that the claims of the '148 and '840 patents are directed to a specific improvement for digital signal processing, including for the efficient identification of audio files using a characteristic thumbprint based on the content of a data signal. Accordingly, each claim of the '148 and '840 patents recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

20. The inventions of the '509 patent resolve technical problems related to biasing narrowcast transmissions according to audience/individual preferences, such as for an Internet audio data stream. The claims of the '509 patent do not merely recite the performance of some business practice known from the pre-computer world along with the requirement to perform it on a computer. Instead, the claims of the '509 patent recite one or more inventive concepts that

are rooted in computerized and Internet technology, and overcome problems specifically arising in that realm.

21. The ’509 patent relates generally to biasing narrowcast transmissions according to audience/individual preferences, such as for an audio data stream. ’509 Patent at 2:61-3:11. As entertainment data streams are particularly susceptible to strong personal preferences, the present invention resolves a need for providing dynamic accommodation of expressed preferences in a community of subscribers or listeners while complying with applicable copyright law. *Id*. at 2:52-57. At the time of the invention, the inventors identified a need for the LAUNCHcast Internet radio service to provide users with an enhanced narrowcast transmission for the listener’s or customer’s enjoyment. *Id*. at 2:15-41. Information derived from user accounts form the basis of a community and collateral preferences allow other subscribing individuals to enjoy the benefit of wider-ranging tastes according to the preferences expressed by the other members of the community. *Id*. at Abstract.

22. The inventive solutions of the claimed inventions of the ’509 patent include methods and apparatus to define an individual data stream for a user by filtering content using stored preferences of members of a community. These inventive solutions overcome one or more problems of the prior art. The prior techniques suffered drawbacks such that a new and novel solution was required. For example, prior art cited by the patent examiner related to “TuneTo.com” focused on matching a user to a set of pre-existing channels, as opposed to defining a different individual data stream for each user.

23. A person of ordinary skill in the art reading the ’509 patent and its claims would understand that the patent’s disclosure and claims are drawn to solving a specific, technical problem arising in the definition of a narrowcast data stream. Moreover, a person of ordinary

skill in the art would understand that the claims' subject matter presents advancements in the field of data stream transmission. The claims do not preempt all types of data stream transmission. For example, the claims do not preempt use of the techniques taught in the prior art cited on the face of the '509 patent. Instead, the claims recite the specific novel solution taught in the specification generally relating to defining an individual data stream for a user by filtering content using stored preferences of members of a community.

24. In light of the foregoing, a person of ordinary skill in the art would understand that the claims of the '509 patent are directed to a specific improvement for data stream transmission, including for defining an individual data stream for a user by filtering content using stored preferences of members of a community. Accordingly, each claim of the '509 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

25. The inventions of the '331 patent resolve technical problems related to discovering relationships among items and recommending items, such as audio tracks, artists, or albums, without requiring undue effort on the part of the user. The claims of the '331 patent do not merely recite the performance of some business practice known from the pre-computer world along with the requirement to perform it on a computer. Instead, the claims of the '331 patent recite one or more inventive concepts that are rooted in computerized and Internet technology, and overcome problems specifically arising in that realm.

26. The '331 patent relates generally to discovering relationships among items and recommending items, such as audio tracks, artists, or albums, by observing user behavior. '331 Patent at 5:66-6:12. The '331 patent provides improved data analysis by avoiding inaccurate assumptions regarding distribution of user preferences. *Id*. at 6:13-15. Conventional user profile

generation techniques, based on user purchases, did not include such a mechanism for determining the degree of satisfaction of a user by observing the user's behavior, since a user does not tend to make repeated purchases of a particular item even if he or she enjoys the item. *Id*. at 6:54-59. Thus, by contrast to conventional monitoring of online purchases, the '331 patent facilitates development of a user profile that indicates the degree to which various items are preferred. *Id*. at 6:59-62. At the time of the invention, a need existed of discovering relationships among items, that is not obtrusive to users and that leads to accurate recommendations based on user preferences. *Id*. at 5:55-58.

27. The inventive solutions of the claimed inventions of the '331 patent include methods and apparatus to discover relationships among items. *Id*. at 5:66-6:2. The identified relationships can be used to recommend items to a user. *Id*. at 6:2-5. These inventive solutions overcome one or more problems of the prior art. As detailed by the specification, the prior techniques suffered drawbacks such that a new and novel solution was required. *Id*. at 1:55-5:62.

28. A person of ordinary skill in the art reading the '331 patent and their claims would understand that the patent's disclosure and claims are drawn to solving a specific, technical problem arising in the discovery of relationships among items by observing user behavior. Moreover, a person of ordinary skill in the art would understand that the claims' subject matter presents advancements in the field. The claims do not preempt all types of discovering user preferences. For example, the claims do not preempt use of the techniques taught in the prior art cited on the face of the '331 patent or the prior art discussed in the specification. Instead, the claims recite the specific novel solution taught in the specification generally relating to the discovery of relationships among items by observing user behavior (e.g., by scoring user logs).

29. In light of the foregoing, a person of ordinary skill in the art would understand that the claims of the '331 patent are directed to a specific improvement for discovering relationships, including by scoring user logs. Accordingly, each claim of the '331 patent recites a combination of elements sufficient to ensure that the claim in practice amounts to significantly more than a patent on an ineligible concept.

V. CLAIMS FOR PATENT INFRINGEMENT

30. The allegations provided below are exemplary and without prejudice to Plaintiff's infringement contentions provided pursuant to the Court's scheduling order and local rules. In providing these allegations, Plaintiff does not convey or imply any particular claim constructions or the precise scope of the claims. Plaintiff's claim construction contentions regarding the meaning and scope of the claim terms will be provided under the Court's scheduling order and local rules. As detailed below, each element of at least one claim of each of the Patents-in-Suit is literally present in the accused products. To the extent that any element is not literally present, each such element is present under the Doctrine of Equivalents. While publicly available information is cited below, Plaintiff also intends to rely on other forms of evidence to show infringement at trial, including but not limited to source code, non-public technical documents, testimony from Spotify engineers, and expert testimony.

COUNT I: PATENT INFRINGEMENT OF THE '148 CARUSO ET AL. PATENT

31. Paragraphs 1-30 are incorporated by reference as if fully stated herein.

32. Spotify has been and is now directly infringing and/or indirectly infringing the '148 Patent by way of inducement and/or contributory infringement, literally and/or under the Doctrine of Equivalents, in violation of 35 U.S.C. § 271, including by making, using, selling, and/or offering for sale in the United States or importing into the United States infringing fingerprinting products and/or services. For example, on information and belief, Spotify uses the

infringing fingerprinting products and/or services for both its Premium Service and Ad-Supported Service. As explained below, these fingerprinting products and/or services are covered by at least one claim of the '148 Patent, including, but not limited to, Claim 1.

33. Upon information and belief, Spotify derives revenue from the activities relating to these infringing fingerprinting products and/or services.

34. Claim 1 of the '148 patent recites a "method" comprising "obtaining, by a processor, a data signal." On information and belief, Spotify practices a method that comprises obtaining, by a processor, a data signal. For example, Spotify's Echoprint obtains, by a processor, a data signal, as shown in the examples below.

3. MATCHING QUERY CODES TO SONGS

We have a database of known tracks, with each track T consisting of an ID and metadata (artist, album, track name). Performing a match involves taking an unknown audio query Q and finding the corresponding track in the reference database.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 2.

3.1.3 Lookups

Lookups are performed in two steps. The same fingerprinting process is performed on the query signal, resulting in a set of `offset hash` pairs. For the first step, the time values

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 31.

35. Claim 1 of the '148 patent recites a "method" comprising "determining, by the processor, a digital thumbprint of the data signal." On information and belief, Spotify practices a method that comprises determining, by the processor, a digital thumbprint of the data signal. For

example, Spotify's Echoprint determines, by the processor, a digital thumbprint of the data signal, as shown in the examples below.

1. MUSIC FINGERPRINTING

"Fingerprinting" of audio files [1, 2, 4] is becoming a necessary feature for any large scale music understanding service or system. Online music stores want to resolve existing user catalog against their cloud storage to save bandwidth. Music indexing tools want to adjust poor metadata.

***

In the next section we present more detail on how fingerprint codes are generated from the source audio. In the following section, we describe the indexing and matching scheme in the server. Finally, we present some results.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

More recently, The Echo Nest [12] released two fingerprinting systems, ENMFP (Ellis et al. 2010), and the open source Echoprint (Ellis, Whitman, and Porter 2011), which is discussed in further detail in Section 3.1, for use by music developers to accurately link audio and music metadata in the Echo Nest developer ecosystem. The fingerprinting services allow developers to upload a recording's fingerprint and gain access to metadata, analysis information, and cross-referenced metadata to a number of other music services on the Internet. The fingerprinting algorithm and software for the lookup server are released under open-source licenses, allowing developers to set up independent fingerprinting servers for private use.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 23.

36. Claim 1 of the '148 patent recites a "method" comprising "comparing, by the processor, the digital thumbprint with a plurality of reference thumbprints." On information and belief, Spotify practices a method that comprises comparing, by the processor, the digital thumbprint with a plurality of reference thumbprints. For example, Spotify's Echoprint compares, by the processor, the digital thumbprint with a plurality of reference thumbprints, as shown in the examples below.

3. MATCHING QUERY CODES TO SONGS

We have a database of known tracks, with each track T consisting of an ID and metadata (artist, album, track name). Performing a match involves taking an unknown audio query Q and finding the corresponding track in the reference database.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 2.

3.1 Echoprint

The Echoprint algorithm works by finding *onsets*—points in time where musical notes occur. Features are created by calculating the difference in time between subsequent onsets and creating a hash of these time values. Matching recordings are found by looking for identical hashes in the reference database.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 28.

3.1.3 Lookups

Lookups are performed in two steps. The same fingerprinting process is performed on the query signal, resulting in a set of `offset hash` pairs. For the first step, the time values are discarded. The inverted index is searched to find all 60 second sub-recordings that contain a hash value that is present in the query. These sub-recordings are ordered by the number of times a recording hash matches a query hash. The 15 sub-recordings with the highest number of matching hashes are returned. If more than one sub-recording for the same recording is returned from this stage, all but one of them are discarded.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 31.

37. Claim 1 of the '148 patent recites a "method" comprising "determining, by the processor, a match of the digital thumbprint with at least one of the plurality of reference thumbprints when a plurality of different sections of the digital thumbprint that have start times differing by a given amount match a plurality of different sections of one or more reference thumbprints whose start times differ by approximately the given amount." On information and belief, Spotify practices a method that comprises determining, by the processor, a match of the digital thumbprint with at least one of the plurality of reference thumbprints when a plurality of different sections of the digital thumbprint that have start times differing by a given amount match a plurality of different sections of one or more reference thumbprints whose start times differ by approximately the given amount. For example, Spotify's Echoprint determines, by the processor, a match of the digital thumbprint with at least one of the plurality of reference thumbprints when a plurality of different sections of the digital thumbprint that have start times differing by a given amount match a plurality of different sections of one or more reference thumbprints whose start times differ by approximately the given amount, as shown in the examples below.

> A 30 second query has about 800 hash keys. The query server returns the documents with the most matches of each code term in the query. In practice we find that there is rarely one document with significantly more matches than all other documents in the index, however, the top matches (we use 15) in this metric will contain the actual match if it exists. We compute a histogram of all time offset t differences per matching key in the result set. We then use the total of the top two histogram buckets to inform the "true score." This allows us to ensure that the codes occur in order even if Q is from a different section of the song and thus has a different absolute time offset.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 2.

> The final score of each candidate recording is calculated by trying to fit the query hashes to the recordings. This is done by calculating the time difference between the onset time in the query and the onset time in the recording for each hash in the recording, and keeping a sum of the number of times each time difference occurs. If a query fingerprint is similar to a recording fingerprint then this offset will be the same for all matching hashes. The reference recording with the highest number of identical time offsets is chosen as the matching query as long as it has more than twice the number of matching offsets as the next recording.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 31.

38. Spotify has knowledge of the '148 Patent by way of this complaint and, to the extent they do not cease their infringing activities, their infringement is and continues to be willful and deliberate.

39. On information and belief, Spotify actively, knowingly, and intentionally induces infringement of one or more claims of the '148 patent under 35 U.S.C. § 271(b) by actively

encouraging others to make, use, offer to sell, sell, and/or import the infringing fingerprinting products and/or services in this judicial district and elsewhere in the United States. For example, Spotify actively instructs, promotes, and encourages the use of the infringing features of the Echoprint software in marketing materials, technical specifications, web pages on its website (e.g., http://the.echonest.com/), press releases, by providing infringing source code (e.g., https://github.com/spotify/echoprint-codegen) and through its sales and distribution channels that encourage infringing use, sales, offers to sell, and importation of the infringing products and/or services. For instance, to the extent Spotify's infringing fingerprinting products and/or services implicate third party platforms, Spotify induces infringement by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that its actions specifically intend and persuade others to engage in conduct Spotify knows is infringement.

40. On information and belief, Spotify also contributorily infringes the '148 patent under 35 U.S.C. § 271(c) because there is no substantial non-infringing use of Spotify's infringing fingerprinting products and/or services. For example, the infringing features of the Echoprint software are a material component for use in practicing claim 1 of the '148 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use, and Spotify's providing of the same results in direct infringement by others. For instance, to the extent Spotify's infringing fingerprint products and/or services implicate third party platforms, Spotify contributorily infringes by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint

that the infringing features of the Echoprint software are especially made or especially adapted for use in infringement of the ’148 patent.

41. To the extent Spotify’s infringing fingerprinting products and/or services implicate third party platforms, Spotify is vicariously liable because Spotify controls and directs the actions of the third party.

42. Plaintiff has no adequate remedy at law against Spotify’s acts of infringement, and, unless Spotify is enjoined from its infringement of the ’148 Patent, Plaintiff will suffer irreparable harm.

43. Spotify, by way of its infringing activities, has caused and continues to cause Plaintiff to suffer damages, the exact amount to be determined at trial.

COUNT II: PATENT INFRINGEMENT OF THE ’840 CARUSO ET AL. PATENT

44. Paragraphs 1-43 are incorporated by reference as if fully stated herein.

45. Spotify has been and is now directly infringing and/or indirectly infringing the ’840 Patent by way of inducement and/or contributory infringement, literally and/or under the Doctrine of Equivalents, in violation of 35 U.S.C. § 271, including by making, using, selling, and/or offering for sale in the United States or importing into the United States infringing fingerprinting products and/or services. For example, on information and belief, Spotify uses the infringing fingerprinting products and/or services for both its Premium Service and Ad-Supported Service. As explained below, these fingerprinting products and/or services are covered by at least one claim of the ’840 Patent, including, but not limited to, Claim 1.

46. Upon information and belief, Spotify derives revenue from the activities relating to these infringing fingerprinting products and/or services.

47. Claim 1 of the '840 patent recites a "method for extracting a digital thumbprint from a data signal." On information and belief, Spotify practices a method for extracting a digital thumbprint from a data signal. For example, Spotify's Echoprint extracts a digital thumbprint from a data signal, as shown in the examples below.

1. MUSIC FINGERPRINTING

"Fingerprinting" of audio files [1, 2, 4] is becoming a necessary feature for any large scale music understanding service or system. Online music stores want to resolve existing user catalog against their cloud storage to save bandwidth. Music indexing tools want to adjust poor metadata.

***

In the next section we present more detail on how fingerprint codes are generated from the source audio. In the following section, we describe the indexing and matching scheme in the server. Finally, we present some results.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

More recently, The Echo Nest [12] released two fingerprinting systems, ENMFP (Ellis et al. 2010), and the open source Echoprint (Ellis, Whitman, and Porter 2011), which is discussed in further detail in Section 3.1, for use by music developers to accurately link audio and music metadata in the Echo Nest developer ecosystem. The fingerprinting services allow developers to upload a recording's fingerprint and gain access to metadata, analysis information, and cross-referenced metadata to a number of other music services on the Internet. The fingerprinting algorithm and software for the lookup server are released under open-source licenses, allowing developers to set up independent fingerprinting servers for private use.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 23.

48. Claim 1 of the '840 patent recites a "method" comprising "filtering the data signal to produce a plurality of filtered signals, at least two of the filtered signals having a different frequency range." On information and belief, Spotify practices a method that comprises filtering the data signal to produce a plurality of filtered signals, at least two of the filtered signals having a different frequency range. For example, Spotify's Echoprint filters the data signal to produce a plurality of filtered signals, at least two of the filtered signals having a different frequency range, as shown in the examples below.

2. GENERATING CODES FROM AUDIO

To achieve greater robustness to the kind of spectral modifications and noise encountered in OTA recordings, Echoprint relies only on the relative timing between success beat-like onsets detected in the audio. Onset detection is performed independently in 8 frequency bands, corresponding to the lowest 8 bands in the MPEG-Audio 32 band filterbank (hence, nominally spanning 0 to 5512.5 Hz). The magnitude of the complex band-pass signal in each band is compared to an exponentially-decaying threshold, and an onset recorded when the signal exceeds the threshold, at which point the threshold is increased to 1.05 × the new signal peak. An adaptive algorithm takes a target "inter-onset-interval" (IOI), and decreases the threshold decay rate when actual IOIs are shorter, or increases the rate of decay if they are longer. The target onset rate for Echoprint is 1 onset per second per band.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

> Once the audio has been downsampled and whitened it is transformed into the frequency domain. Echoprint uses a 128 band cosine filter bank to perform this transform (Ramstad and Tanem 1991). The filterbank is moved over the signal with a hop size of 32 samples. The resulting frequency bands are grouped into eight equally spaced bins by summing the absolute difference of adjacent bands. The eight bins are equally spread out from 0 Hz to 5512.5 Hz.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 29.

49. Claim 1 of the '840 patent recites a "method" comprising "computing a measure of power contained within at least a segment of each of a plurality of the filtered signals." On information and belief, Spotify practices a method that comprises computing a measure of power contained within at least a segment of each of a plurality of the filtered signals. For example, Spotify's Echoprint computes a measure of power contained within at least a segment of each of a plurality of the filtered signals, as shown in the examples below.

> **2. GENERATING CODES FROM AUDIO**
>
> To achieve greater robustness to the kind of spectral modifications and noise encountered in OTA recordings, Echoprint relies only on the relative timing between success beat-like onsets detected in the audio. Onset detection is performed independently in 8 frequency bands, corresponding to the lowest 8 bands in the MPEG-Audio 32 band filterbank (hence, nominally spanning 0 to 5512.5 Hz). The magnitude of the complex band-pass signal in each band is compared to an exponentially-decaying threshold, and an onset recorded when the signal exceeds the threshold, at which point the threshold is increased to 1.05 × the new signal peak. An adaptive algorithm takes a target "inter-onset-interval" (IOI), and decreases the threshold decay rate when actual IOIs are shorter, or increases the rate of decay if they are longer. The target onset rate for Echoprint is 1 onset per second per band.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

3.1.2 Hashing and storage

Echoprint hashes are calculated based on the time difference between musical onsets in each band. The first step of the hashing process is to detect the onsets in the audio signal. Onsets are detected in each frequency band independently. In each band, an envelope follower is used to measure the amplitude of the band. When the amplitude reaches a threshold, an onset is registered. After an onset has been detected, 128 samples must pass before the next onset. The amplitude of the onset is multiplied by an exponentially decaying curve to calculate a new threshold value. Subsequently detected onsets must exceed this threshold in order to be counted. The multiplier decay is adaptive to the number of onsets that are being detected. Echoprint has a target of generating one onset per second in each frequency band. If onsets are being generated at more than this rate, the multiplier decay is increased, resulting in a larger threshold to exceed. If the rate of onsets decreases too much, the multiplier is decreased to compensate.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 29.

50. Claim 1 of the '840 patent recites a "method" comprising "determining characteristic data for the data signal, by a computer processor, wherein the characteristic data for the data signal is determined based on the computed measures of power." On information and belief, Spotify practices a method that comprises determining characteristic data for the data signal, by a computer processor, wherein the characteristic data for the data signal is determined based on the computed measures of power. For example, Spotify's Echoprint determines characteristic data for the data signal, by a computer processor, wherein the characteristic data for the data signal is determined based on the computed measures of power, as shown in the examples below.

2. GENERATING CODES FROM AUDIO

To achieve greater robustness to the kind of spectral modifications and noise encountered in OTA recordings, Echoprint relies only on the relative timing between success beat-like onsets detected in the audio. Onset detection is performed independently in 8 frequency bands, corresponding to the lowest 8 bands in the MPEG-Audio 32 band filterbank (hence, nominally spanning 0 to 5512.5 Hz). The magnitude of the complex band-pass signal in each band is compared to an exponentially-decaying threshold, and an onset recorded when the signal exceeds the threshold, at which point the threshold is increased to 1.05 × the new signal peak. An adaptive algorithm takes a target "inter-onset-interval" (IOI), and decreases the threshold decay rate when actual IOIs are shorter, or increases the rate of decay if they are longer. The target onset rate for Echoprint is 1 onset per second per band.

Pairs of successive IOIs in each band, quantized into units of 23.2 ms, are combined to make a hash. To provide robustness against spurious or missed onsets, each onset is considered along with its four successors. Six different hashes (IOI pairs) are created by choosing all possible pairs of succeeding onsets from the four.

Thus, the overall hash rate is approximately 8 (bands) × 1 (onset per second) × 6 (hashes per onset) ≈ 48 hashes/sec. Since onsets are approximately one second apart, the quantized IOIs are of the order of 1/.0232 = 43, or around 5-6 bits, and a pair of onsets constitutes around 12 bits of information. This is combined with the 3 bit band index to generate the raw hash, which is then stored along with the time it occurs within the file.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

3.1.2 Hashing and storage

Echoprint hashes are calculated based on the time difference between musical onsets in each band. The first step of the hashing process is to detect the onsets in the audio signal. Onsets are detected in each frequency band independently. In each band, an envelope follower is used to measure the amplitude of the band. When the amplitude reaches a threshold, an onset is registered. After an onset has been detected, 128 samples must pass before the next onset. The amplitude of the onset is multiplied by an exponentially decaying curve to calculate a new threshold value. Subsequently detected onsets must exceed this threshold in order to be counted. The multiplier decay is adaptive to the number of onsets that are being detected. Echoprint has a target of generating one onset per second in each frequency band. If onsets are being generated at more than this rate, the multiplier decay is increased, resulting in a larger threshold to exceed. If the rate of onsets decreases too much, the multiplier is decreased to compensate.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 29.

To encode the onsets to numerical values, the algorithm considers the time of each onset (o) and the time of its four successors ($s1$–$s4$). A hash value is created by taking the time delta between pairs of the five onsets (Figure 3.2), and the band in which the onsets occur.

Hash	1	2	3	4	5	6
bytes 1–2	$s1 - o$	$s1 - o$	$s2 - o$	$s1 - o$	$s2 - o$	$s3 - o$
bytes 3–4	$s2 - s1$	$s3 - s1$	$s3 - s2$	$s4 - s1$	$s4 - s2$	$s4 - s3$
byte 5	band	band	band	band	band	band

Figure 3.2 Calculating the time delta between pairs of onsets to create six hashes.

The two hash values and band index are stored in a 40-bit (5 byte) number (two bytes for each delta and 1 byte for the band index). The number is reduced to a 32-bit integer with the MurmurHash algorithm (Appleby 2009).

Each onset and set of successors generates six hashes. By pairing onsets and successors, the algorithm adds robustness against the failure of onsets to be detected. If one onset is missed then there will still be some matching hashes at that point in time. This hashing method results in approximately 48 hashes per second of audio (8 bands, 1 onset per second, 6 hashes per onset).

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 30.

51. Claim 1 of the '840 patent recites a "method" comprising "writing a digital thumbprint to a non-transitory computer-readable storage medium, the digital thumbprint comprising the determined characteristic data for the data signal." On information and belief, Spotify practices a method that comprises writing a digital thumbprint to a non-transitory computer-readable storage medium, the digital thumbprint comprising the determined characteristic data for the data signal. For example, Spotify's Echoprint writes a digital thumbprint to a non-transitory computer-readable storage medium, the digital thumbprint comprising the determined characteristic data for the data signal, as shown in the examples below.

2. GENERATING CODES FROM AUDIO

To achieve greater robustness to the kind of spectral modifications and noise encountered in OTA recordings, Echoprint relies only on the relative timing between success beat-like onsets detected in the audio. Onset detection is performed independently in 8 frequency bands, corresponding to the lowest 8 bands in the MPEG-Audio 32 band filterbank (hence, nominally spanning 0 to 5512.5 Hz). The magnitude of the complex band-pass signal in each band is compared to an exponentially-decaying threshold, and an onset recorded when the signal exceeds the threshold, at which point the threshold is increased to 1.05 × the new signal peak. An adaptive algorithm takes a target "inter-onset-interval" (IOI), and decreases the threshold decay rate when actual IOIs are shorter, or increases the rate of decay if they are longer. The target onset rate for Echoprint is 1 onset per second per band.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 1.

3. MATCHING QUERY CODES TO SONGS

We have a database of known tracks, with each track T consisting of an ID and metadata (artist, album, track name). Performing a match involves taking an unknown audio query Q and finding the corresponding track in the reference database.

To build the database, each track is split into 60 second segments, with adjacent sections overlapping by 30 seconds. This helps to remove bias introduced when longer songs provide more matches for a set of query hashes.

The codes for a 60 second segment are represented as terms of a document D in an inverted index. The combination of the unique track ID plus the segment number is used as the document ID. Our underlying data store uses Apache Solr with a custom query handler to provide a fast lookup of a code query to list of document IDs.

A 30 second query has about 800 hash keys. The query server returns the documents with the most matches of each code term in the query. In practice we find that there is rarely one document with significantly more matches than all other documents in the index, however, the top matches (we use 15) in this metric will contain the actual match if it exists. We compute a histogram of all time offset $\mathbf{t}$ differences per matching key in the result set. We then use the total of the top two histogram buckets to inform the "true score." This allows us to ensure that the codes occur in order even if Q is from a different section of the song and thus has a different absolute time offset.

See, e.g., https://www.ee.columbia.edu/~dpwe/pubs/EllisWP11-echoprint.pdf at 2.

3.1.2 Hashing and storage

Echoprint hashes are calculated based on the time difference between musical onsets in each band. The first step of the hashing process is to detect the onsets in the audio signal. Onsets are detected in each frequency band independently. In each band, an envelope follower is used to measure the amplitude of the band. When the amplitude reaches a threshold, an onset is registered. After an onset has been detected, 128 samples must pass before the next onset. The amplitude of the onset is multiplied by an exponentially decaying curve to calculate a new threshold value. Subsequently detected onsets must exceed this threshold in order to be counted. The multiplier decay is adaptive to the number of onsets that are being detected. Echoprint has a target of generating one onset per second in each frequency band. If onsets are being generated at more than this rate, the multiplier decay is increased, resulting in a larger threshold to exceed. If the rate of onsets decreases too much, the multiplier is decreased to compensate.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 29.

To store hashes, they are paired with the time that the onset occurs in the audio query. The `offset hash` pairs for a single recording are split into a number of sub-recordings, each 60 seconds long, overlapping with the previous sub-recording by 30 seconds. Hashes are split in this manner because the matching component of the system scores recordings by the number of times a hash in the recording matches hashes in the query. If the hashes were not split, then long recordings would receive an unfair advantage at the lookup stage because recordings with repeated content could generate the same hash value at many points in the recording. The hash values are stored in an inverted index, mapping a hash to a sub-recording id and the point in time at which the hash occurs. The Echoprint server application uses Apache Solr [9], a fast text search engine, to store the hash index. The full set of `offset hash` values for each recording are stored in a separate database for use in the lookup process.

See, e.g., http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1544123706348~456 at 30-31.

52. Spotify has knowledge of the '840 Patent by way of this complaint and, to the extent they do not cease their infringing activities, their infringement is and continues to be willful and deliberate.

53. On information and belief, Spotify actively, knowingly, and intentionally induces infringement of one or more claims of the '840 patent under 35 U.S.C. § 271(b) by actively encouraging others to make, use, offer to sell, sell, and/or import the infringing fingerprinting products and/or services in this judicial district and elsewhere in the United States. For example, Spotify actively instructs, promotes, and encourages the use of the infringing features of the Echoprint software in marketing materials, technical specifications, web pages on its website (e.g., http://the.echonest.com/), press releases, by providing infringing source code (e.g., https://github.com/spotify/echoprint-codegen) and through its sales and distribution channels that encourage infringing use, sales, offers to sell, and importation of the infringing products and/or services. For instance, to the extent Spotify's infringing fingerprint products and/or services implicate third party platforms, Spotify induces infringement by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that its actions specifically intend and persuade others to engage in conduct Spotify knows is infringement.

54. On information and belief, Spotify also contributorily infringes the '840 patent under 35 U.S.C. § 271(c) because there is no substantial non-infringing use of Spotify's infringing fingerprinting products and/or services. For example, the infringing features of the Echoprint software are a material component for use in practicing claim 1 of the '840 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use,

and Spotify's providing of the same results in direct infringement by others. For instance, to the extent Spotify's infringing fingerprint products and/or services implicate third party platforms, Spotify contributorily infringes by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that the infringing features of the Echoprint software are especially made or especially adapted for use in infringement of the '840 patent.

55. To the extent Spotify's infringing fingerprinting products and/or services implicate third party platforms, Spotify is vicariously liable because Spotify controls and directs the actions of the third party.

56. Plaintiff has no adequate remedy at law against Spotify's acts of infringement, and, unless Spotify is enjoined from its infringement of the '840 Patent, Plaintiff will suffer irreparable harm.

57. Spotify, by way of its infringing activities, has caused and continues to cause Plaintiff to suffer damages, the exact amount to be determined at trial.

COUNT III: PATENT INFRINGEMENT OF THE '509 BOULTER ET AL. PATENT

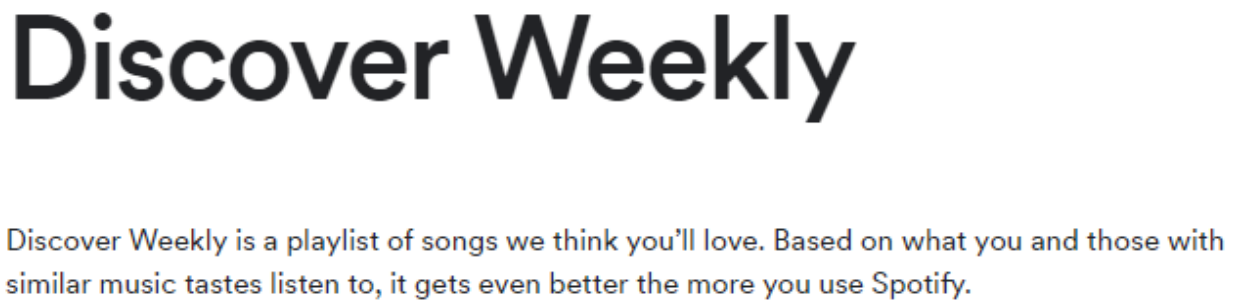
58. Paragraphs 1-57 are incorporated by reference as if fully stated herein.

59. Spotify has been and is now directly infringing and/or indirectly infringing the '509 Patent by way of inducement and/or contributory infringement, literally and/or under the Doctrine of Equivalents, in violation of 35 U.S.C. § 271, including by making, using, selling, and/or offering for sale in the United States or importing into the United States infringing personalized playlist products and/or services. For example, on information and belief, Spotify uses the infringing personalized playlist products and/or services for both its Premium Service

and Ad-Supported Service. As explained below, these personalized playlist products and/or services are covered by at least one claim of the '509 Patent, including, but not limited to, Claim 43.

60. Upon information and belief, Spotify derives revenue from the activities relating to these infringing personalized playlist products and/or services.

61. Claim 43 of the '509 patent recites a "computer-implemented method" comprising "determining a first community of members by filtering a data store of preferences for data stream content, said data stream content comprising data stream elements, said members of said first community having at least one stored preference in common." On information and belief, Spotify practices a method that comprises determining a first community of members by filtering a data store of preferences for data stream content, said data stream content comprising data stream elements, said members of said first community having at least one stored preference in common. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify determines a first community of members by filtering a data store of preferences for data stream content, said data stream content comprising data stream elements, said members of said first community having at least one stored preference in common, as shown in the examples below.



See, e.g., https://support.spotify.com/us/using_spotify/playlists/discover-weekly/

Daily Mix

Introducing the music you love, minus the effort.

Your Daily Mixes - each corresponding to a style of music you've been into - is loaded with your favorite tracks and artists, plus a few recommendations. Watch as your Daily Mixes change as your taste changes.

See, e.g., https://support.spotify.com/us/article/daily-mix/

Release Radar

Release Radar is a playlist of new releases recommended just for you.

Updated every Friday with music released over the past few weeks, it includes artists you listen to the most and some new discoveries based on your recent listening.

See, e.g., https://support.spotify.com/us/article/release-radar/

Spotify Radio

Spotify Radio picks the music for you, so you can sit back and listen to the music you love.

Create a Radio station with any song, album, artist, or playlist, and Spotify will play music based on that.

See, e.g., https://support.spotify.com/us/using_spotify/discover_music/spotify-radio/

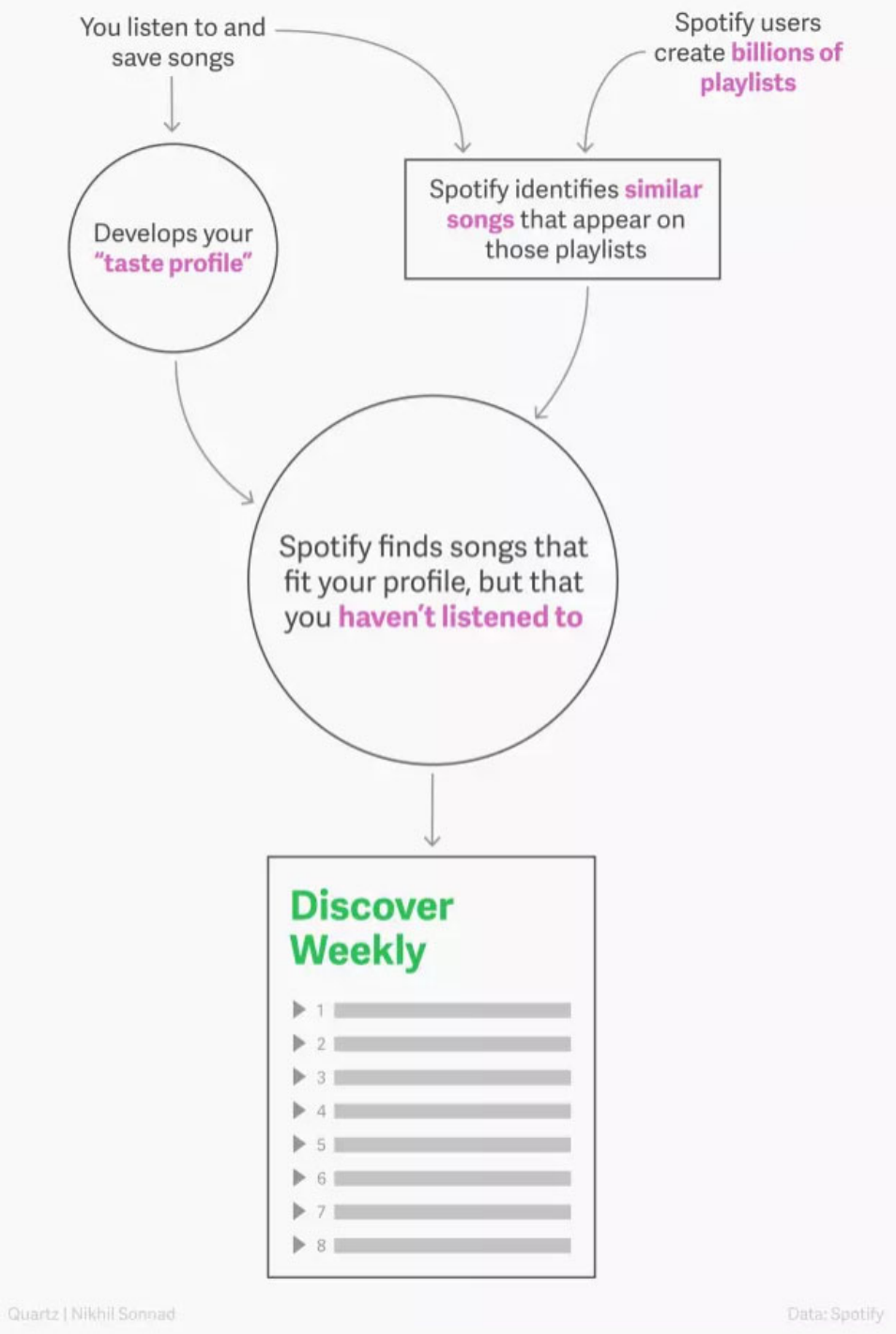
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

> 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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

62. Claim 43 of the '509 patent recites a "computer-implemented method" comprising "determining data stream elements of said data stream content by filtering said data stream content using said stored preferences of said members of said first community, each determined data stream element being preferred by said first member community." On information and belief, Spotify practices a method that comprises determining data stream elements of said data stream content by filtering said data stream content using said stored preferences of said members of said first community, each determined data stream element being preferred by said first member community. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify determines data stream elements of said data stream content by filtering said data stream content using said stored preferences of said members of said first community, each determined data stream element being preferred by said first member community, as shown in the examples below.

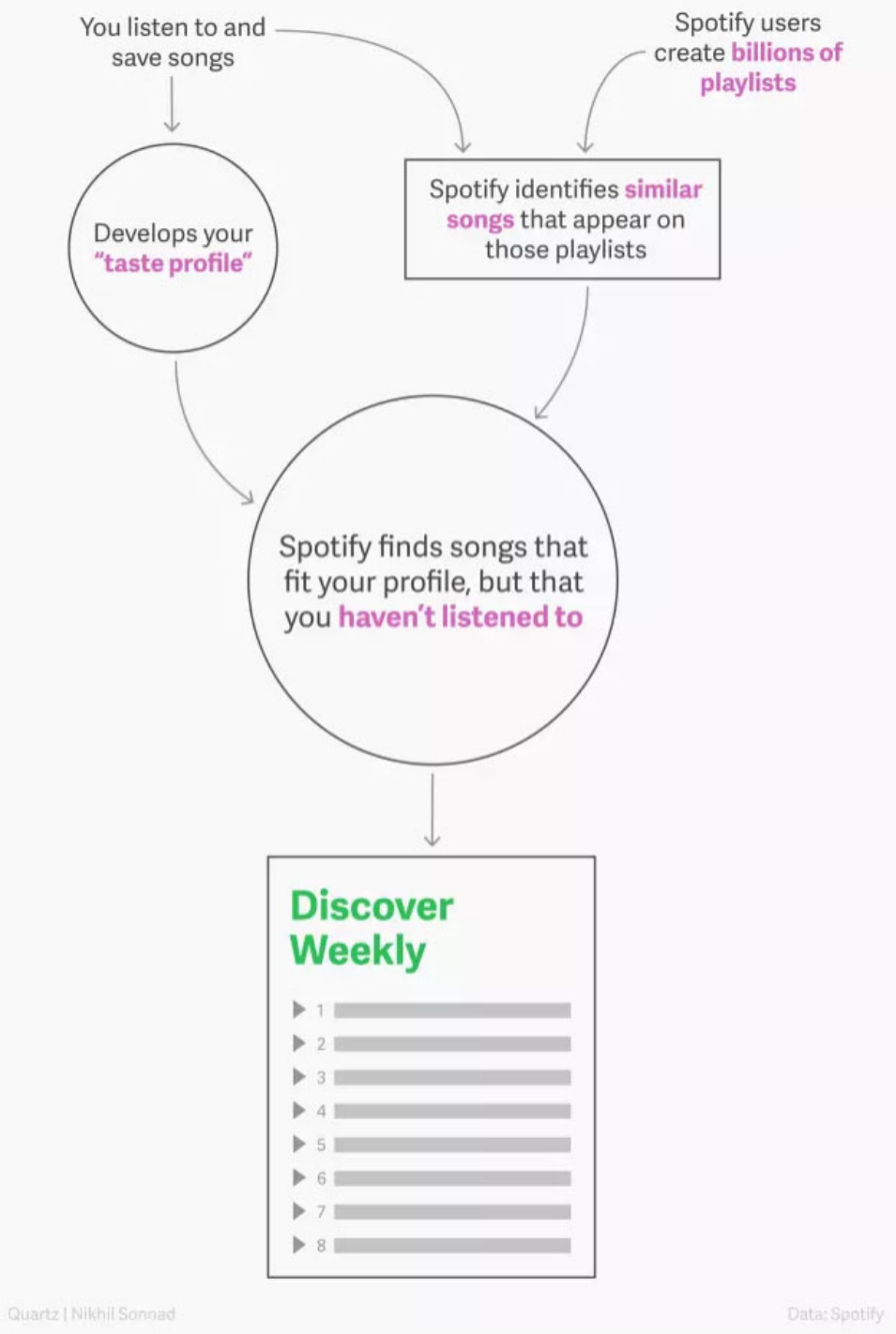
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



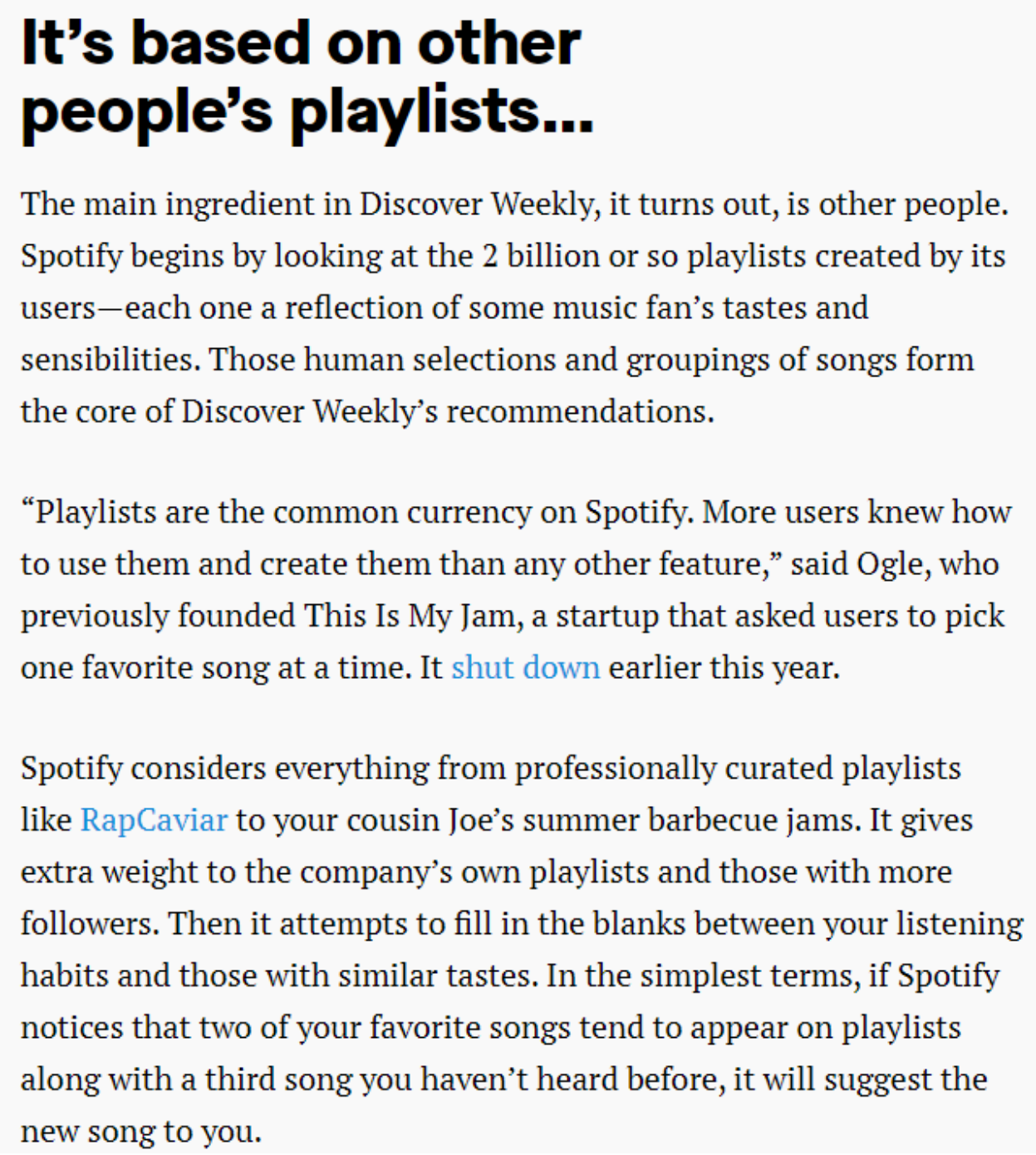
See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

> 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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

63. Claim 43 of the '509 patent recites a "computer-implemented method" comprising "defining a different individual data stream for each user of a plurality of users, said defining comprising, for each individual data stream, further filtering said determined data stream elements using said stored preferences of said members of said first community to select at least one data stream element from said determined data stream elements for inclusion in said individual data stream." On information and belief, Spotify practices a method that comprises defining a different individual data stream for each user of a plurality of users, said defining comprising, for each individual data stream, further filtering said determined data stream elements using said stored preferences of said members of said first community to select at least one data stream element from said determined data stream elements for inclusion in said individual data stream. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify defines a different individual data stream for each user of a plurality of users, said defining comprising, for each individual data stream, further filtering said determined data stream elements using said stored preferences of said members of said first community to select at least one data stream element from said

determined data stream elements for inclusion in said individual data stream, as shown in the examples below.



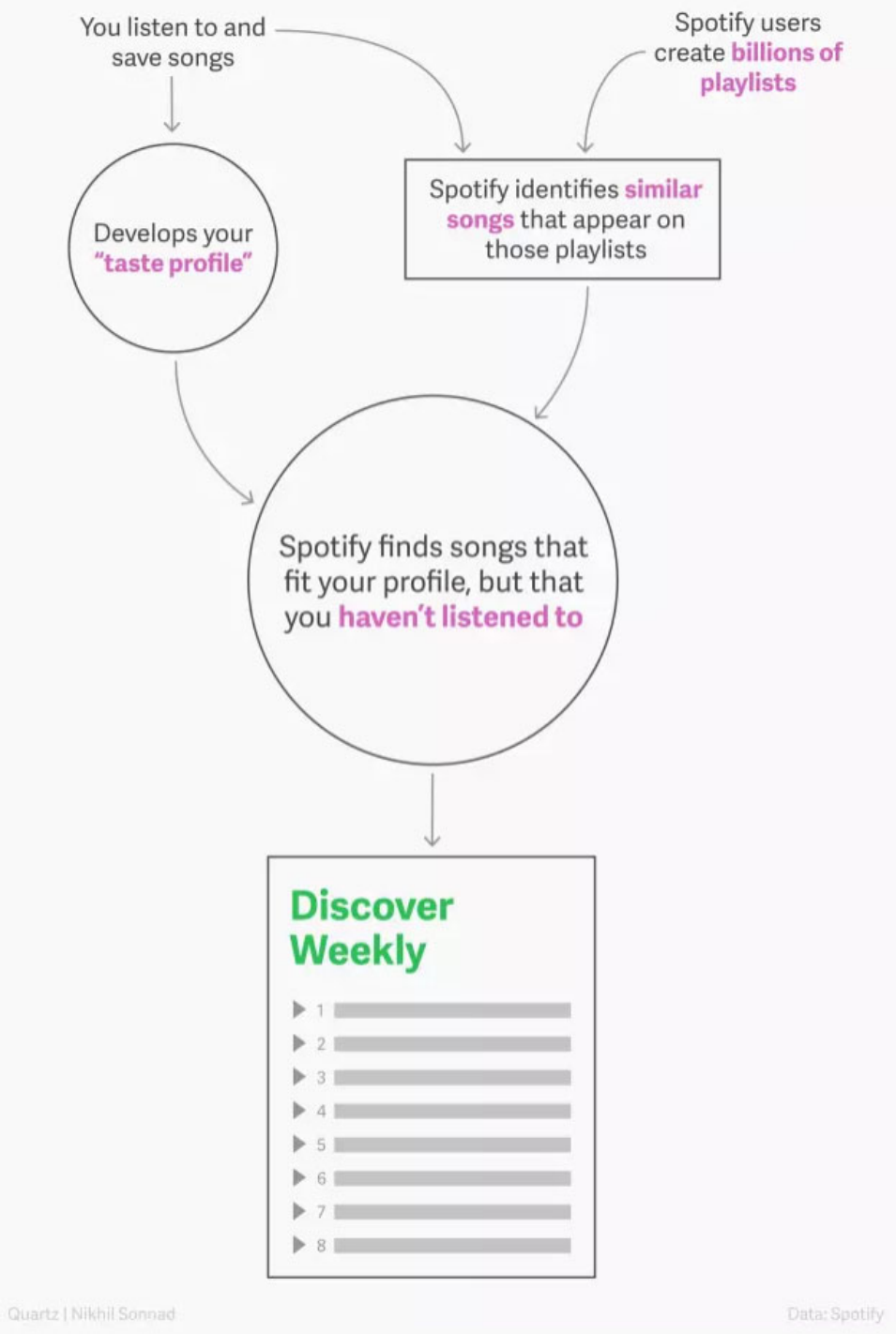
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

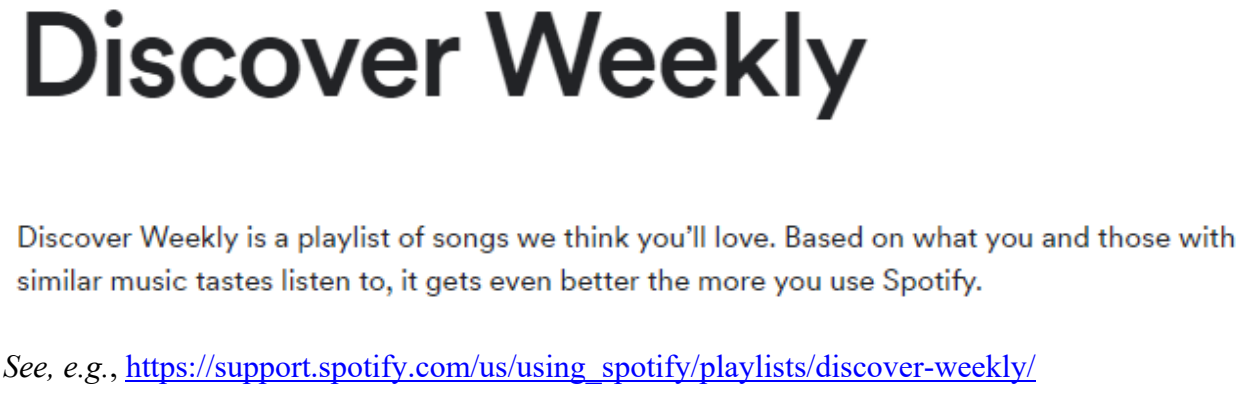
See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



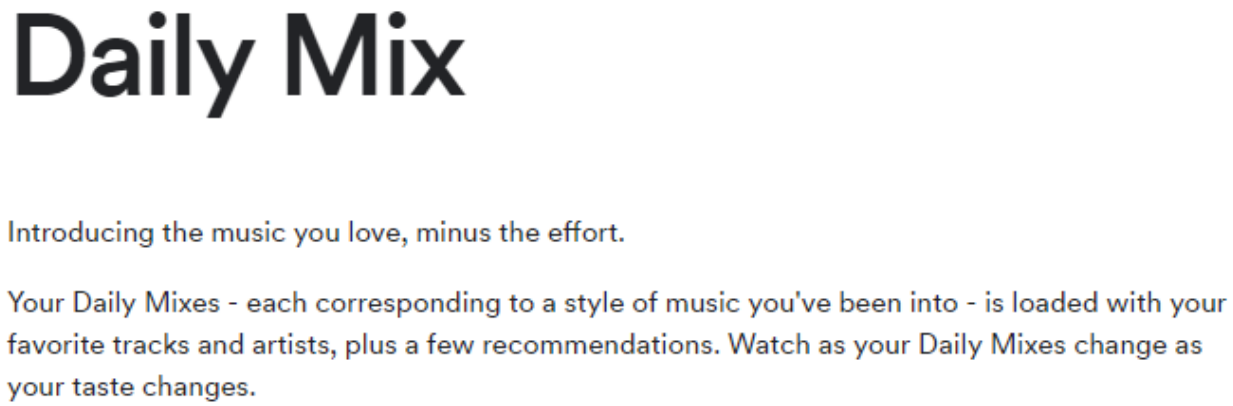
See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

64. Claim 43 of the '509 patent recites a "computer-implemented method" comprising "transmitting said individual data stream to a user computer." On information and

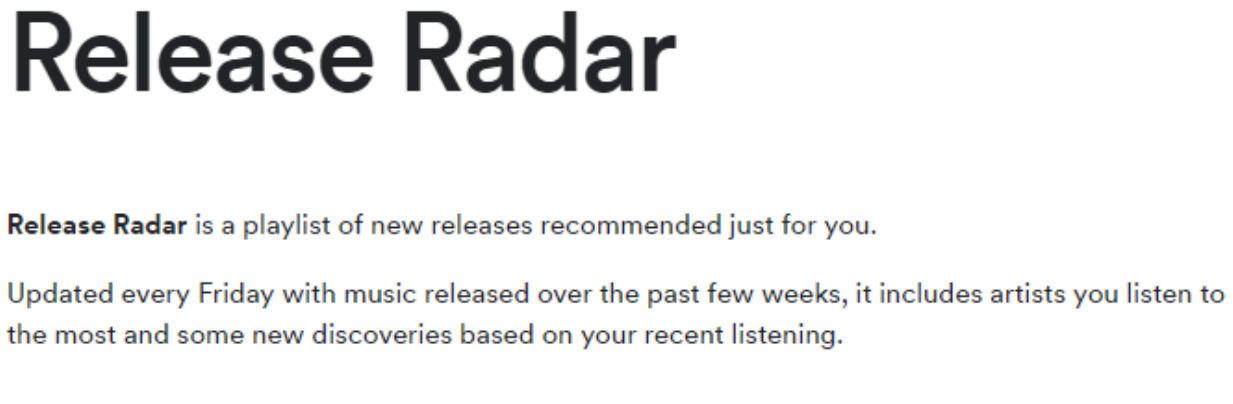
belief, Spotify practices a method that comprises transmitting said individual data stream to a user computer. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify transmits an individual data stream to a user computer, as shown in the examples below.



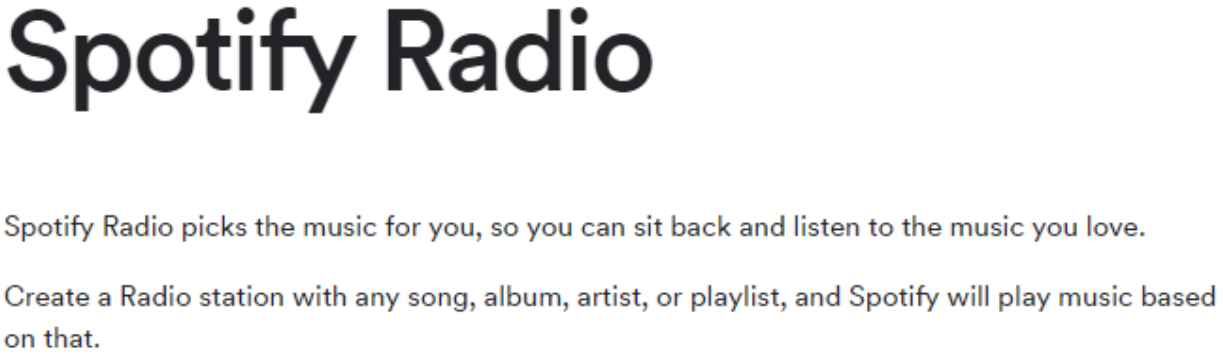
See, e.g., https://support.spotify.com/us/using_spotify/playlists/discover-weekly/



See, e.g., https://support.spotify.com/us/article/daily-mix/



See, e.g., https://support.spotify.com/us/article/release-radar/



See, e.g., https://support.spotify.com/us/using_spotify/discover_music/spotify-radio/

65. Spotify has knowledge of the '509 Patent by way of this complaint and, to the extent they do not cease their infringing activities, their infringement is and continues to be willful and deliberate.

66. On information and belief, Spotify actively, knowingly, and intentionally induces infringement of one or more claims of the '509 patent under 35 U.S.C. § 271(b) by actively encouraging others to make, use, offer to sell, sell, and/or import the infringing personalized playlist products and/or services in this judicial district and elsewhere in the United States. For example, Spotify actively instructs, promotes, and encourages the use of the infringing features of its software by providing infringing source code and through its technical support, sales and distribution channels that encourage infringing use, sales, offers to sell, and importation of the infringing products and/or services. For instance, to the extent Spotify's infringing personalized playlist products and/or services implicate third party platforms, Spotify induces infringement by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that its actions specifically intend and persuade others to engage in conduct Spotify knows is infringement.

67. On information and belief, Spotify also contributorily infringes the ’509 patent under 35 U.S.C. § 271(c) because there is no substantial non-infringing use of Spotify’s infringing personalized playlist products and/or services. For example, the infringing features of Spotify’s software are a material component for use in practicing claim 43 of the ’509 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use, and Spotify’s providing of the same results in direct infringement by others. For instance, to the extent Spotify’s infringing personalized playlist products and/or services implicate third party platforms, Spotify contributorily infringes by providing Spotify’s software to the third party platform and controlling and directing the actions of the third party’s direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that the infringing features of its software are especially made or especially adapted for use in infringement of the ’509 patent.

68. To the extent Spotify’s infringing personalized playlist products and/or services implicate third party platforms, Spotify is vicariously liable because Spotify controls and directs the actions of the third party.

69. Plaintiff has no adequate remedy at law against Spotify’s acts of infringement, and, unless Spotify is enjoined from its infringement of the ’509 Patent, Plaintiff will suffer irreparable harm.

70. Spotify, by way of its infringing activities, has caused and continues to cause Plaintiff to suffer damages, the exact amount to be determined at trial.

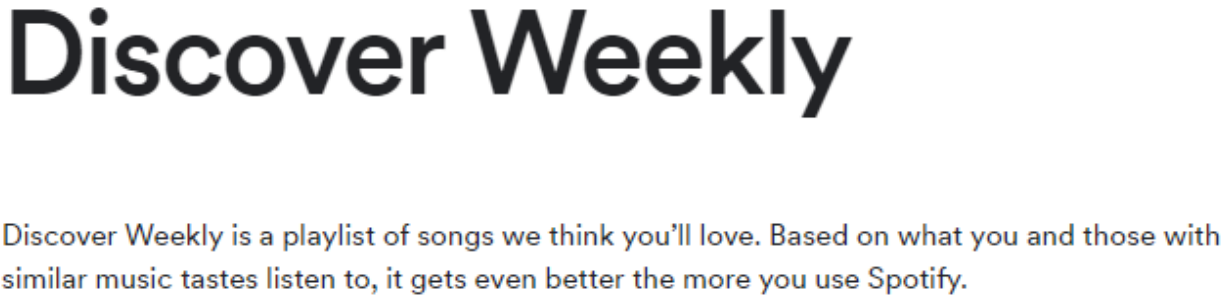
COUNT IV: PATENT INFRINGEMENT OF THE ’331 DUNNING ET AL. PATENT

71. Paragraphs 1-70 are incorporated by reference as if fully stated herein.

72. Spotify has been and is now directly infringing and/or indirectly infringing the '331 Patent by way of inducement and/or contributory infringement, literally and/or under the Doctrine of Equivalents, in violation of 35 U.S.C. § 271, including by making, using, selling, and/or offering for sale in the United States or importing into the United States infringing personalized playlist products and/or services. For example, on information and belief, Spotify uses the infringing personalized playlist products and/or services for both its Premium Service and Ad-Supported Service. As explained below, these personalized playlist products and/or services are covered by at least one claim of the '331 Patent, including, but not limited to, Claim 1.

73. Upon information and belief, Spotify derives revenue from the activities relating to these infringing personalized playlist products and/or services.

74. Claim 1 of the '331 patent recites a "computer-implemented method of discovering relationships between items." On information and belief, Spotify practices a computer-implemented method of discovering relationships between items. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify discovers relationships between items, as shown in the examples below.



See, e.g., https://support.spotify.com/us/using_spotify/playlists/discover-weekly/

Daily Mix

Introducing the music you love, minus the effort.

Your Daily Mixes - each corresponding to a style of music you've been into - is loaded with your favorite tracks and artists, plus a few recommendations. Watch as your Daily Mixes change as your taste changes.

See, e.g., https://support.spotify.com/us/article/daily-mix/

Release Radar

Release Radar is a playlist of new releases recommended just for you.

Updated every Friday with music released over the past few weeks, it includes artists you listen to the most and some new discoveries based on your recent listening.

See, e.g., https://support.spotify.com/us/article/release-radar/

Spotify Radio

Spotify Radio picks the music for you, so you can sit back and listen to the music you love.

Create a Radio station with any song, album, artist, or playlist, and Spotify will play music based on that.

See, e.g., https://support.spotify.com/us/using_spotify/discover_music/spotify-radio/

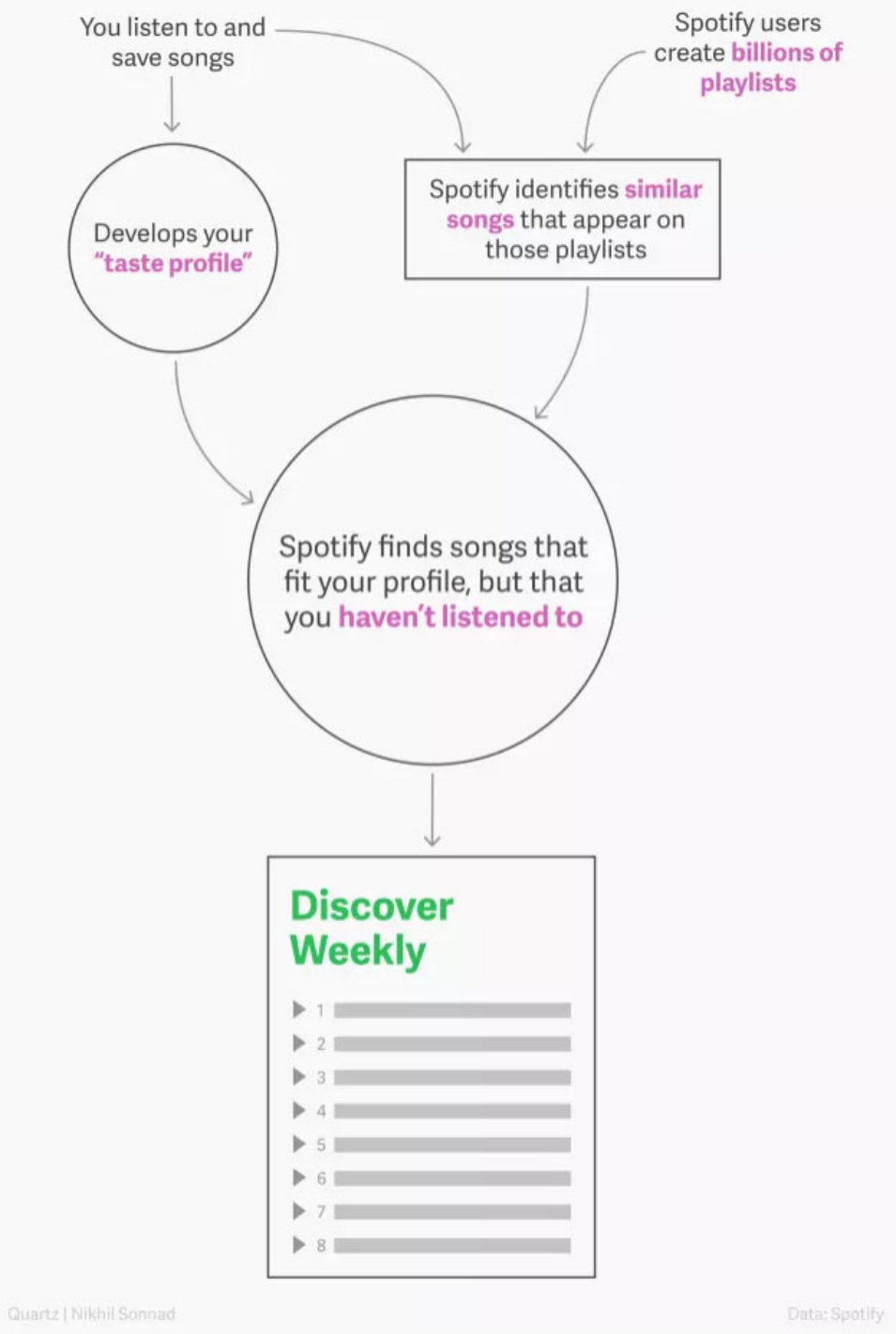
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

> 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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

75. Claim 1 of the '331 patent recites a "computer-implemented method" comprising "accepting, in a computer, item selections detected from a plurality of users." On information and belief, Spotify practices a method that comprises accepting, in a computer, item selections detected from a plurality of users. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify accepts, in a computer, item selections detected from a plurality of users, as shown in the examples below.

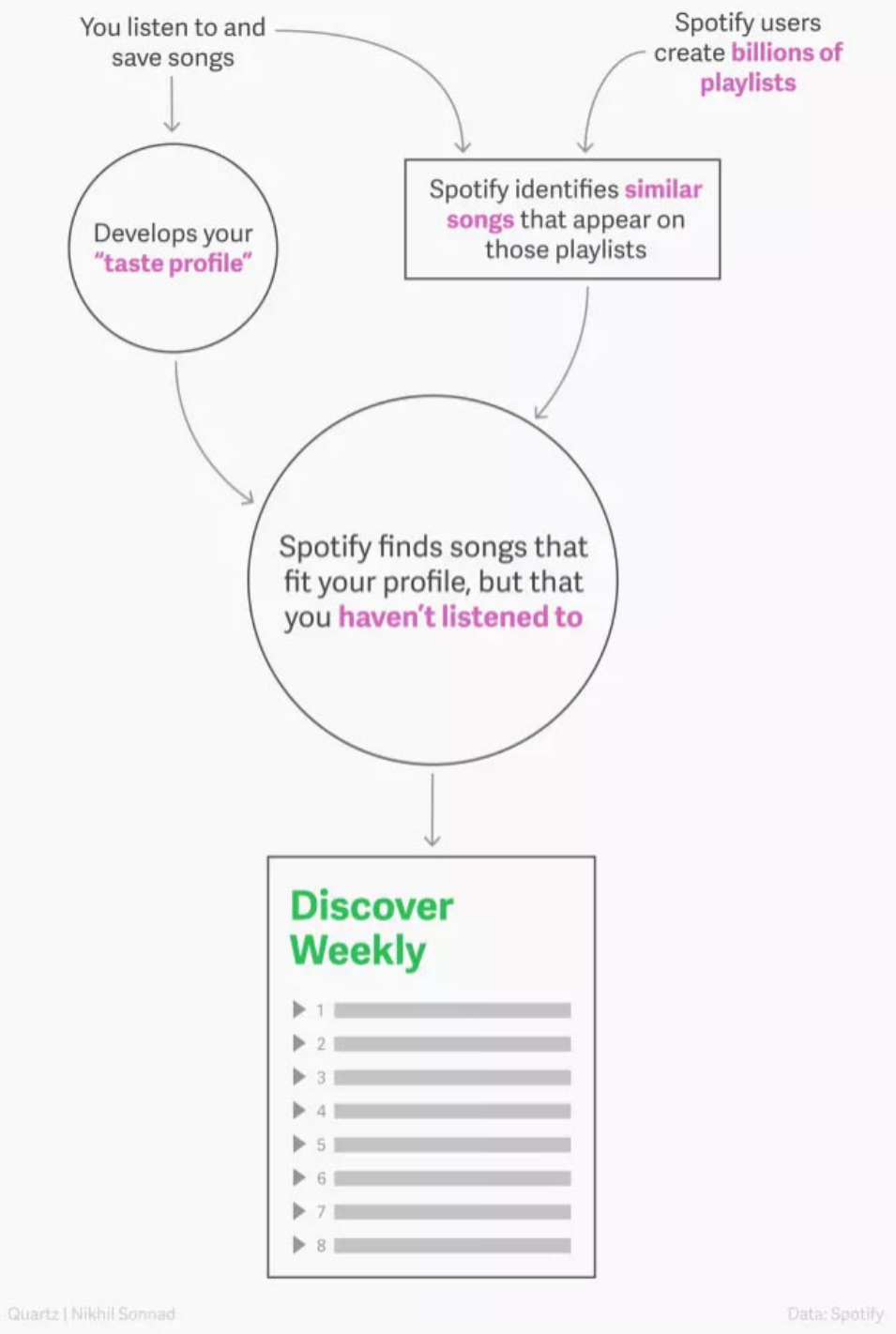
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

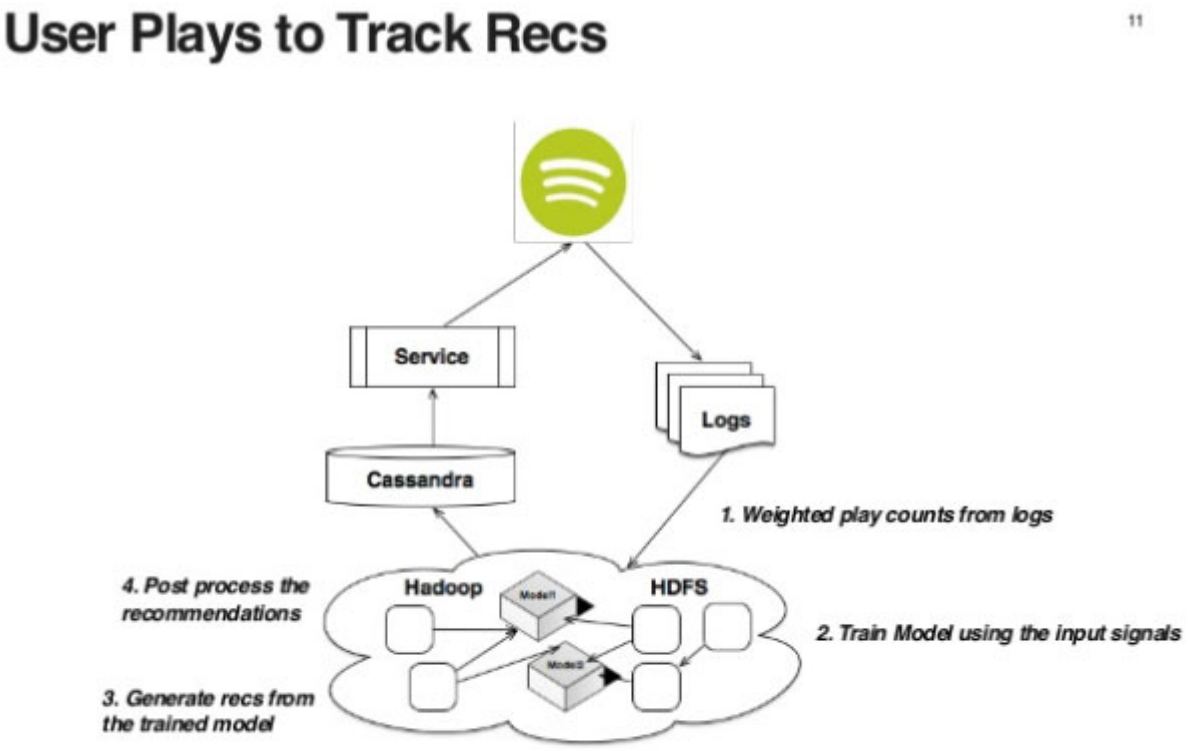
"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

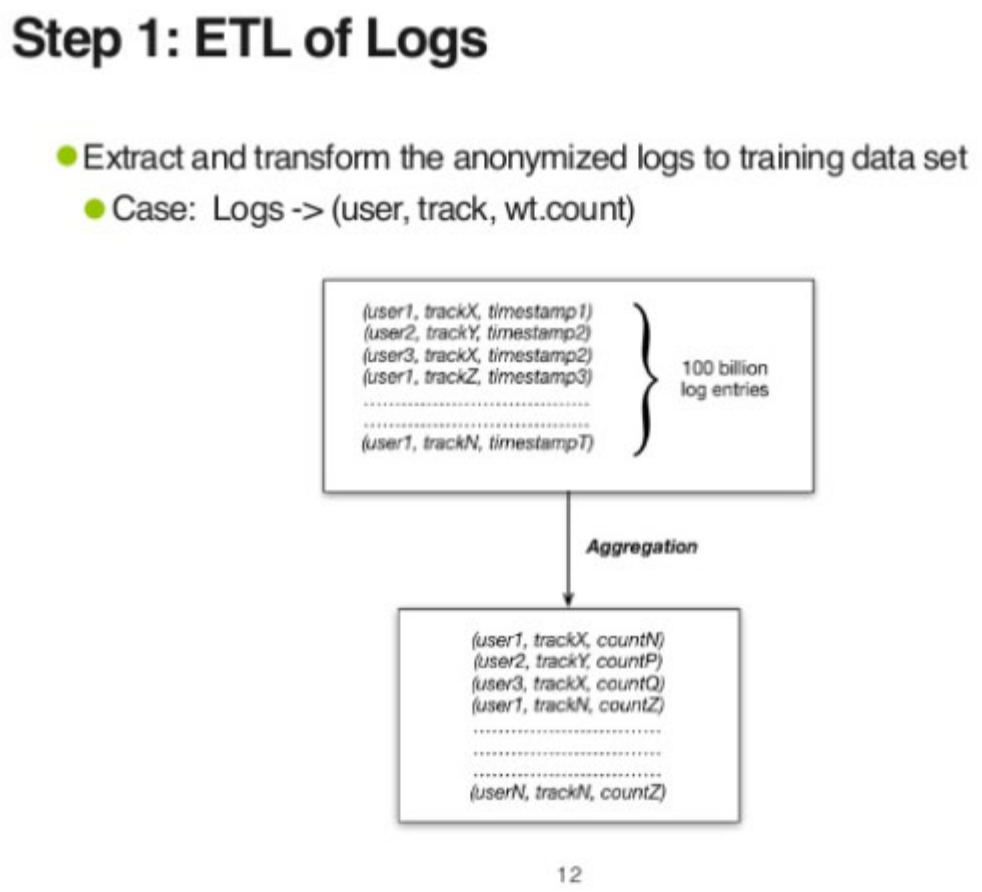
See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

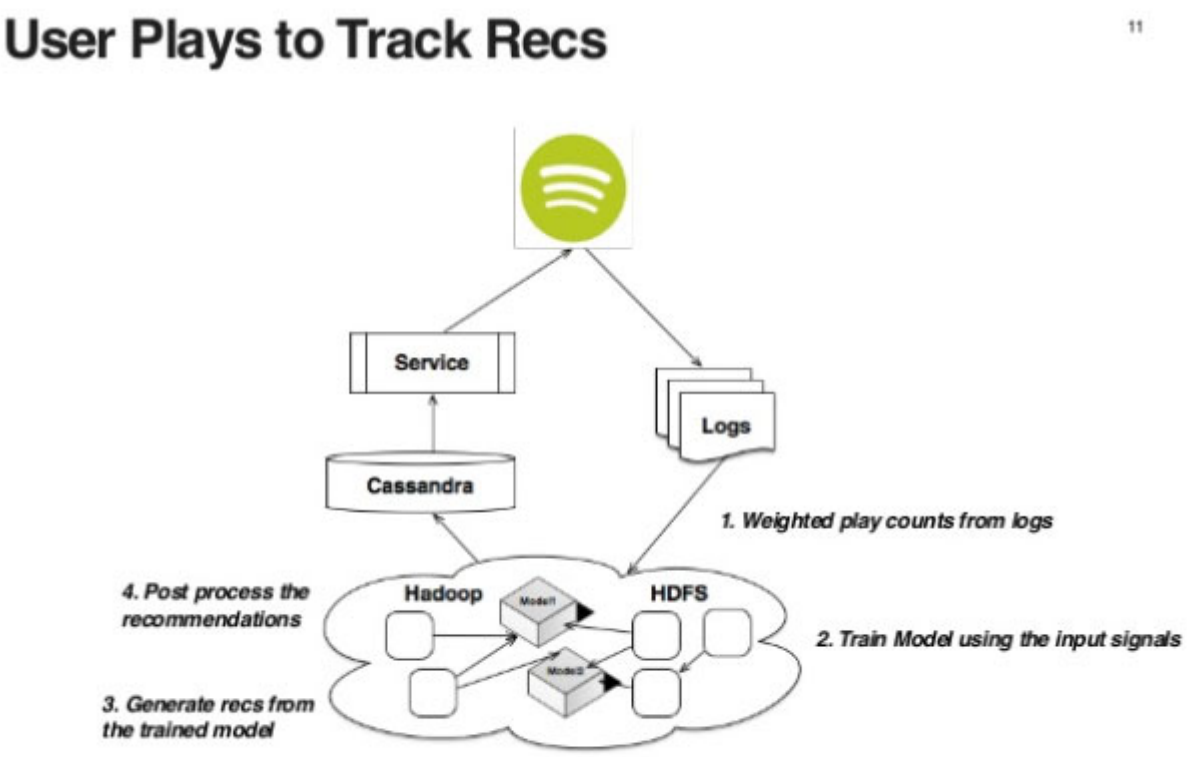


See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

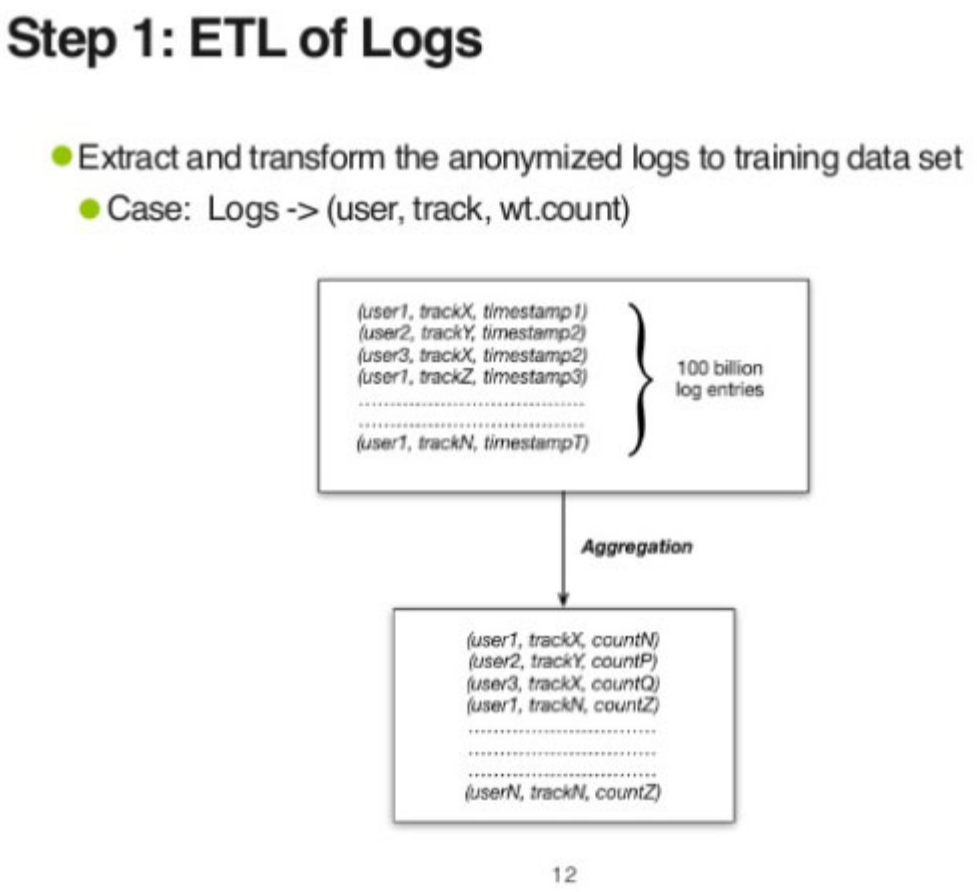


See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

76. Claim 1 of the '331 patent recites a "computer-implemented method" comprising "generating, in the computer, a log for each user, each log containing identifiers corresponding to detected user item selections." On information and belief, Spotify practices a method that comprises generating, in the computer, a log for each user, each log containing identifiers corresponding to detected user item selections. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify generates, in the computer, a log for each user, each log containing identifiers corresponding to detected user item selections, as shown in the examples below.



See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify



See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

77. Claim 1 of the '331 patent recites a "computer-implemented method" comprising "accepting, in the computer, a query including at least one query item identifier." On information and belief, Spotify practices a method that comprises accepting, in the computer, a query including at least one query item identifier. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify accepts, in the computer, a query including at least one query item identifier, as shown in the examples below.

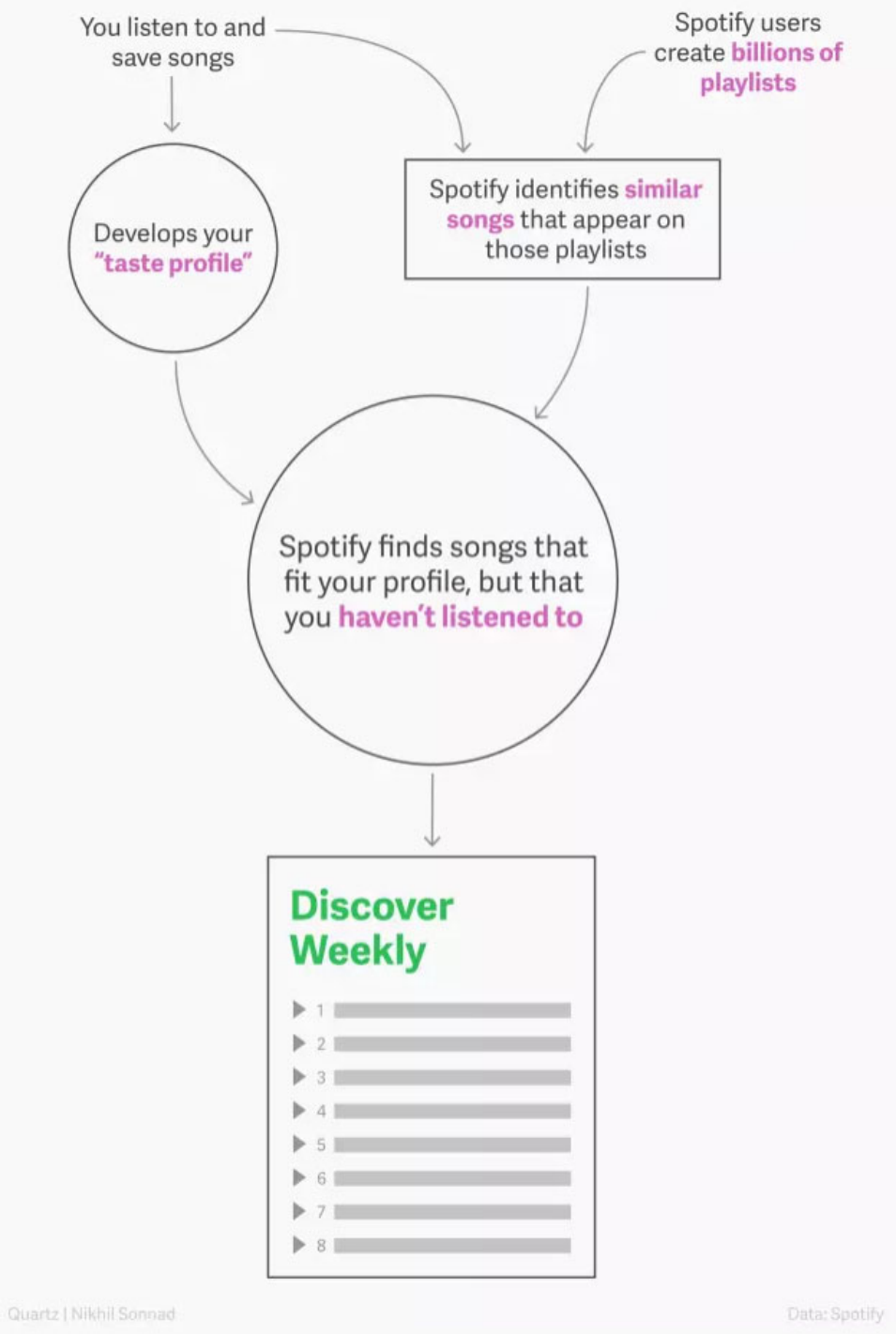
It's based on other people's playlists...

The main ingredient in Discover Weekly, it turns out, is other people. Spotify begins by looking at the 2 billion or so playlists created by its users—each one a reflection of some music fan's tastes and sensibilities. Those human selections and groupings of songs form the core of Discover Weekly's recommendations.

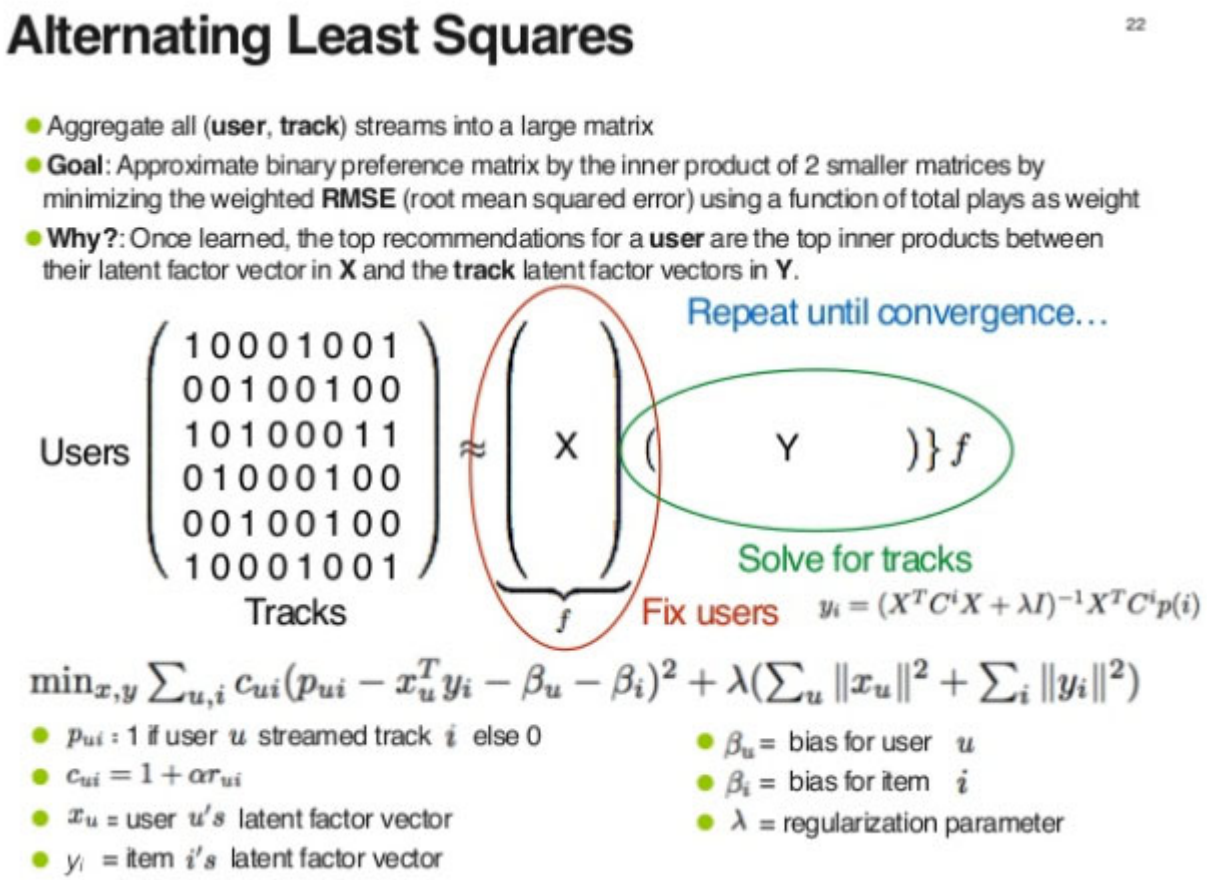
"Playlists are the common currency on Spotify. More users knew how to use them and create them than any other feature," said Ogle, who previously founded This Is My Jam, a startup that asked users to pick one favorite song at a time. It shut down earlier this year.

Spotify considers everything from professionally curated playlists like RapCaviar to your cousin Joe's summer barbecue jams. It gives extra weight to the company's own playlists and those with more followers. Then it attempts to fill in the blanks between your listening habits and those with similar tastes. In the simplest terms, if Spotify notices that two of your favorite songs tend to appear on playlists along with a third song you haven't heard before, it will suggest the new song to you.

See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/



See, e.g., https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/

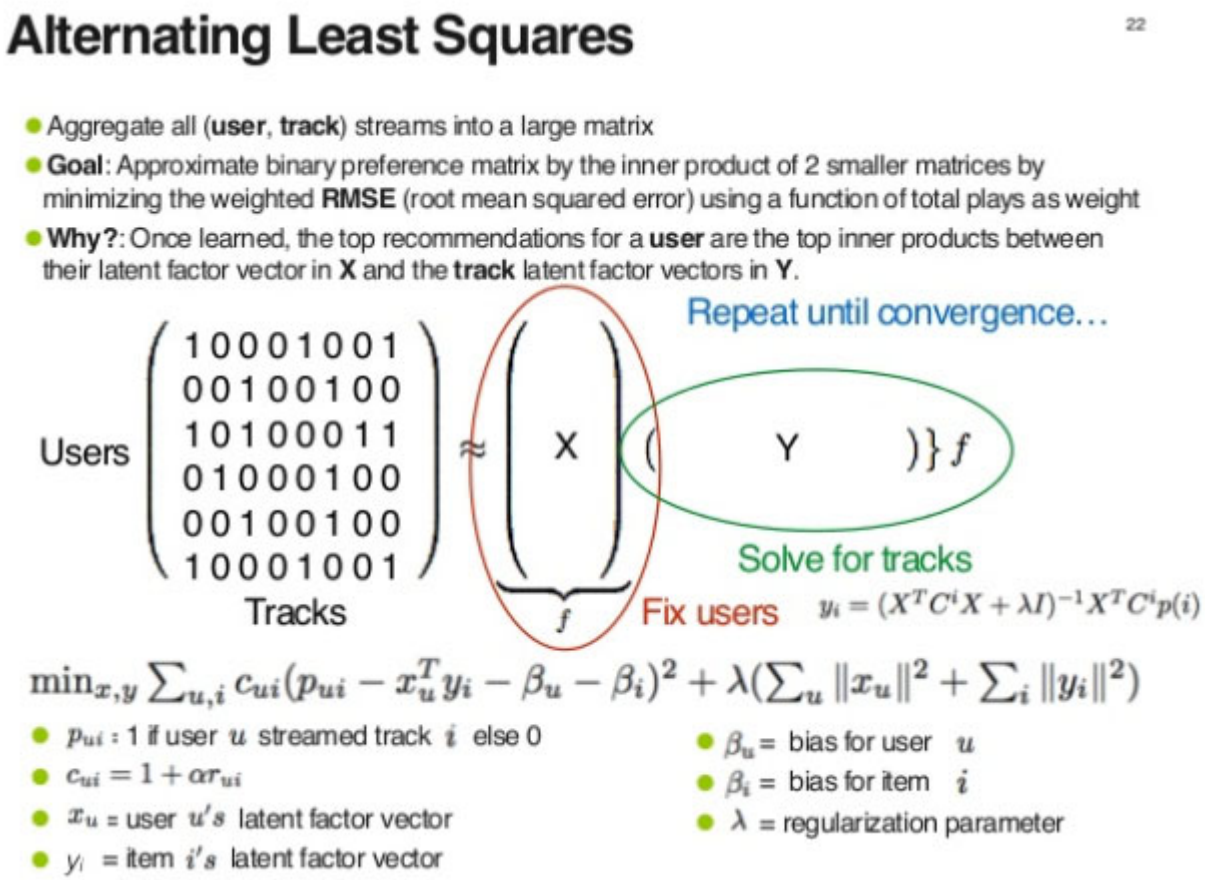


See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

> 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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

78. Claim 1 of the '331 patent recites a "computer-implemented method" comprising "scoring, in the computer, each of the user logs, the scoring for each user log being responsive to a frequency of occurrence of the at least one query item identifier in the user log, a frequency of occurrence of the at least one query item identifier in all of the user logs and a query weight for the at least one query item identifier in the query, so as to generate a user log score for each user log based exclusively on detected user item selections and the at least one query item." On information and belief, Spotify practices a method that comprises scoring, in the computer, each of the user logs, the scoring for each user log being responsive to a frequency of occurrence of the at least one query item identifier in the user log, a frequency of occurrence of the at least one query item identifier in all of the user logs and a query weight for the at least one query item identifier in the query, so as to generate a user log score for each user log based exclusively on detected user item selections and the at least one query item. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify scores, in the computer, each of the user logs, the scoring for each user log being responsive to a frequency of occurrence of the at least one query item identifier in the user log, a frequency of occurrence of the at least one query item identifier in all of the user logs and a query weight for the at least one query item identifier in the query, so as to generate a user log score for each user log based exclusively on detected user item selections and the at least one query item, as shown in the examples below.



See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

Annoy

- 70 million users, at least 4 million tracks for candidates per user
- Brute Force Approach:
  - O(70M x 4M x 10) ~= 0(3 peta-operations)!
- **A**pproximate **N**earest **N**eighbor **O**h **Y**eah!
  - Uses Local Sensitive Hashing
  - Clone: https://github.com/spotify/annoy

28

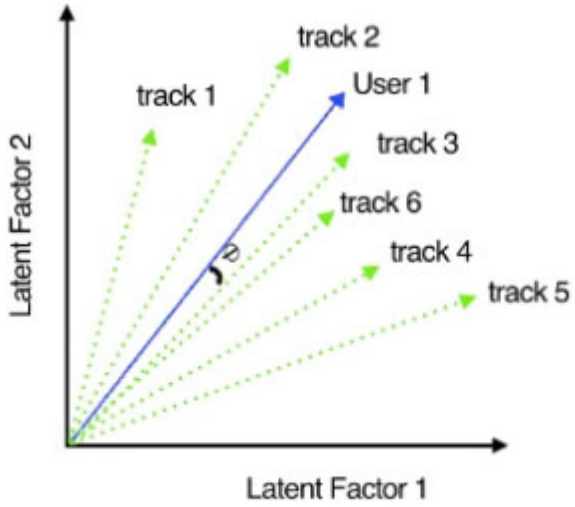
See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

We use it at Spotify for music recommendations. After running matrix factorization algorithms, every user/item can be represented as a vector in f-dimensional space. This library helps us search for similar users/items. We have many millions of tracks in a high-dimensional space, so memory usage is a prime concern.

See, e.g., https://github.com/spotify/annoy

24

Recommendations via Dot Product!



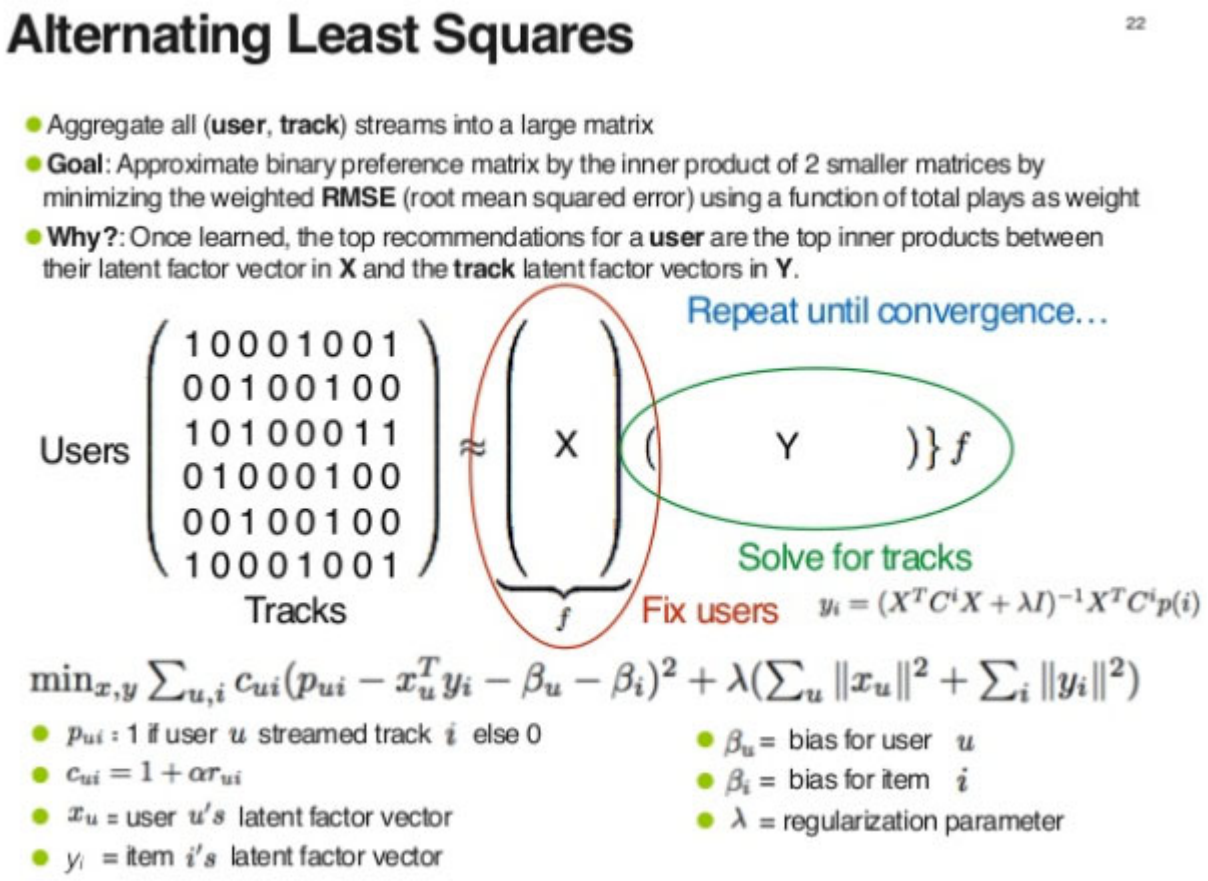
See, e.g., https://www.slideshare.net/vidhyamurali/cf-models-for-music-recommendations-at-spotify

> 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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

79. Claim 1 of the '331 patent recites a "computer-implemented method" comprising "determining, in the computer, at least one result item, responsive to the scoring of each user log,

so as to discover at least one relationship based exclusively on detected user item selections and the at least one query item." On information and belief, Spotify practices a method that comprises determining, in the computer, at least one result item, responsive to the scoring of each user log, so as to discover at least one relationship based exclusively on detected user item selections and the at least one query item. For example, in order to create personalized playlists (e.g., Discover Weekly, Daily Mix, Release Radar, Spotify Radio, etc.), Spotify determines, in the computer, at least one result item, responsive to the scoring of each user log, so as to discover at least one relationship based exclusively on detected user item selections and the at least one query item, as shown in the examples below.



See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

Annoy

- 70 million users, at least 4 million tracks for candidates per user
- Brute Force Approach:
  - O(70M x 4M x 10) ~= 0(3 peta-operations)!
- **A**pproximate **N**earest **N**eighbor **O**h **Y**eah!
  - Uses Local Sensitive Hashing
  - Clone: https://github.com/spotify/annoy

28

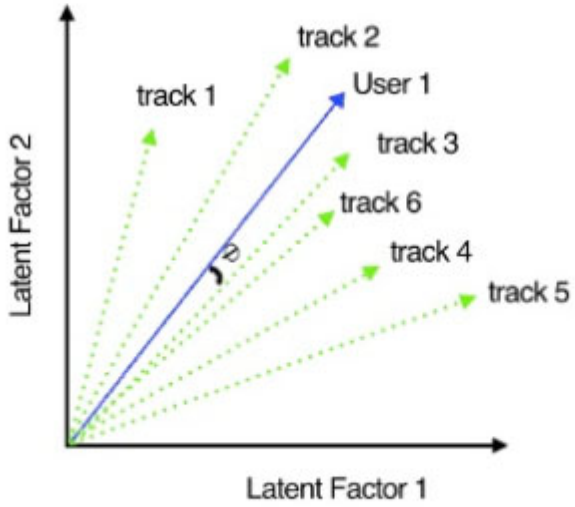
See, e.g., https://www.slideshare.net/vidhyamurali/building-data-pipelines-for-music-recommendations-at-spotify

We use it at Spotify for music recommendations. After running matrix factorization algorithms, every user/item can be represented as a vector in f-dimensional space. This library helps us search for similar users/items. We have many millions of tracks in a high-dimensional space, so memory usage is a prime concern.

See, e.g., https://github.com/spotify/annoy

24

Recommendations via Dot Product!



See, e.g., https://www.slideshare.net/vidhyamurali/cf-models-for-music-recommendations-at-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.

See, e.g., https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe

80. Spotify has knowledge of the '331 Patent by way of this complaint and, to the extent they do not cease their infringing activities, their infringement is and continues to be willful and deliberate.

81. On information and belief, Spotify actively, knowingly, and intentionally induces infringement of one or more claims of the '331 patent under 35 U.S.C. § 271(b) by actively encouraging others to make, use, offer to sell, sell, and/or import the infringing personalized playlist products and/or services in this judicial district and elsewhere in the United States. For example, Spotify actively instructs, promotes, and encourages the use of the infringing features of its software by providing infringing source code and through its technical support, sales and distribution channels that encourage infringing use, sales, offers to sell, and importation of the infringing products and/or services. For instance, to the extent Spotify's infringing personalized playlist products and/or services implicate third party platforms, Spotify induces infringement by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that its actions specifically intend and persuade others to engage in conduct Spotify knows is infringement.

82. On information and belief, Spotify also contributorily infringes the '331 patent under 35 U.S.C. § 271(c) because there is no substantial non-infringing use of Spotify's infringing personalized playlist products and/or services. For example, the infringing features of Spotify's software are a material component for use in practicing claim 1 of the '331 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use, and Spotify's providing of the same results in direct infringement by others. For instance, to the extent Spotify's infringing personalized playlist products and/or services implicate third party

platforms, Spotify contributorily infringes by providing Spotify's software to the third party platform and controlling and directing the actions of the third party's direct infringement. To the extent Spotify does not cease its infringing activities, Spotify has knowledge by way of this complaint that the infringing features of its software are especially made or especially adapted for use in infringement of the '331 patent.

83. To the extent Spotify's infringing personalized playlist products and/or services implicate third party platforms, Spotify is vicariously liable because Spotify controls and directs the actions of the third party.

84. Plaintiff has no adequate remedy at law against Spotify's acts of infringement, and, unless Spotify is enjoined from its infringement of the '331 Patent, Plaintiff will suffer irreparable harm.

85. Spotify, by way of its infringing activities, has caused and continues to cause Plaintiff to suffer damages, the exact amount to be determined at trial.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for the following relief:

86. A judgment in favor of Plaintiff that Spotify, has infringed, directly and indirectly, by way of inducement and/or contributory infringement, literally and/or under the doctrine of equivalents, the Patents-in-Suit;

87. A permanent injunction, enjoining Spotify and its officers, directors, agents, servants, employees, affiliates, divisions, branches, subsidiaries, parents, and all others acting in concert or privity with any of them from infringing, inducing the infringement of, or contributing to the infringement of the Patents-in-Suit;

88. An award of damages to which Plaintiff is entitled under 35 U.S.C. § 284 and 35 U.S.C. § 154(d) for Spotify's past infringement and any continuing or infringement post-trial up until the date Spotify is finally and permanently enjoined from further infringement and a final judgment is entered, including both compensatory damages and treble damages for willful infringement;

89. A judgment and order against Spotify for exemplary damages as determined by the trier of fact;

90. A judgment that Spotify's infringement has been willful;

91. Pre- and post-judgment interest as allowed by law on any damages awarded to Plaintiff;

92. A judgment and order requiring Spotify to pay the costs of this action (including all disbursements), as well as attorneys' fees as provided by 35 U.S.C. § 285;

93. A judgment and order requiring that, in the event a permanent injunction preventing future infringement is not granted, Spotify pay to Plaintiff compulsory ongoing licensing fees, as determined by the Court in equity; and

94. Such other and further relief in law or in equity to which Plaintiff may be justly entitled.

DEMAND FOR JURY TRIAL

Plaintiff demands a trial by jury of any and all issues triable of right before a jury, except for future patent infringement, which is an issue in equity to be determined by the Court.

Dated: January 28, 2019

Respectfully submitted,

Of Counsel:

FARNAN LLP

/s/ Michael J. Farnan

Steven J. Pollinger
Seth R. Hasenour
MCKOOL SMITH, P.C.
300 W. 6th Street, Suite 1700
Austin, Texas 78701
Telephone: (512) 692-8704
Facsimile: (512) 692-8744
spollinger@mckoolsmith.com
shasenour@mckoolsmith.com

Brian E. Farnan (Bar No. 4089)
Michael J. Farnan (Bar No. 5165)
919 N. Market St., 12th Floor
Wilmington, DE 19801
Telephone: (302) 777-0300
Fax: (302) 777-0301
bfarnan@farnanlaw.com
mfarnan@farnanlaw.com

Attorneys for Plaintiff Excalibur IP, LLC