Relevance-based Music Recommendation using Clustering on Spotify Dataset

Koay Jing Zhi[1,a]

Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

[a]jzkoay@gmail.com

Keywords: Spotify, Music Streaming Platform, Music Recommendation, Recommendation System, Clustering

Abstract. This paper presented K-Means clustering algorithm on the *Spotify Dataset 1921-2020* to group music into different genres based on their similarity in characteristics. Music recommendation was made by suggesting artists who produced tracks that is similar to user's favourite genres. K value of 30 was found as a good starting point for optimal number of clusters using elbow method to calculate Sum of Squared Error (SSE) and K value. Future works includes applying Principal Component Analysis (PCA) for dimensionality reduction and cross-validation to justify quality of clusters.

Introduction

Nowadays, online music streaming service has emerged as the primary way of consuming music. Spotify, being one of the largest online music streaming platform provides millions tracks, podcast and other digital content from artists all over the world. It has changed the way consumers interact with musics, users can listening to algorithmically-generated playlists and tracks instead of discover music on their own from the vast amount of available contents.

As such, it has deliver a more customized and personalized experiences to users by suggesting music according to their habits and preferences. Recommendation system is essential for providing seamlessly music listening experience by presenting the most relevant music to the user. It provides a more engaging user experience for music consumption, thus increase the user conversion to subscription service and higher retention on the platform, which are closely related to business for music streaming service platforms. Apart from that, music recommendation systems also allow specific genres of artist to be discover and gain popularity in a niche market, and help new musicians to reach a bigger market.

Generally, existing music recommendation studies can be categorized into different groups depending on their approach and goal: popularity-based, relevance-based, and learner-based for ranking songs and display to users [1]; short-term and long-term goals which concern short-term user needs and long-term engagement [18]; Collaborative Filtering method, Natural Language Processing (NLP) and Raw Audio Features depending on the features they leverage [21, 23]. Their mutual objective is generating the most relevant content to users preferences which is helpful and meaningful.

The objective of this work is relevance-based music recommendation using clustering technique that make use of metadata and extracted audio features from tracks. This work will address the cold-start problem of recommendation where user profile is not available for personalization. Tracks' metadata refers to the track's name, artist, released year whereas the audio features extracted from tracks includes valence, acousticness, energy, instrumentalness, tempo, loudness, key and many else.

The rest of this paper is organized as follows: Section 2 provides a brief review of related works; Section 3 describe the methodology used in this work; Section 4 specify the experimental setup including dataset and parameters; Section 5 presents analysis of the experimental outcome obtained; and Section 6 is the conclusion that covers contribution of this paper and future work suggestion.

Related Works

In the early research, music recommendation system was recognized as composition of three key components: users, items, and user-item matching algorithms [27]. First, User profiling is conducted based on attributes such as their demographic, geographic, and psycho-graphic data, as well as their listening history. The second component - music item consists of metadata such as cover name, title, released and acoustic features from analysing the audio signal including beat, tempo, instrument and etc. Then, user-content matching algorithms can be broadly categorized into two popular approaches: collaborative filtering and content-based. The idea of collaborative filtering is finding out similar musical tastes between users by comparing their listening history. Due to the reliance on user's listening history, it work best with existence of long-term user profiles. While content-based approach leverage metadata and audio features to make recommendation, be it popularity-based, relevance-based, or learner-based ranking .

The study in music recommendation domain experienced a boom in recent years, a number of approaches such as hybridization, cross-domain recommendation, and active learning have been proposed [25]. Yet majority of research works aims to improve the prediction accuracy of an algorithm based on long-term user profiles. The reliance on usage data from user profiles for long-term recommendation rise cold start problems. Long-term music recommendation also concerns diversity in music consumption in the long term [1]. Cold start problem refers to when a new user registers to the system or a new item is added to the catalog and the recommendation system does not have sufficient data associated with these items and users. In this case, recommendation system cannot properly suggest existing items to a new user (new user problem) or suggest a new item to the existing users (new item problem) since there are no data of the new user or new item at all, and it takes time to populate the data [25]. Both collaborative filtering and content-based approaches have cold start problems but the later have only cold start problems for new users because they can extract features from newly added to the catalog and make recommendations.

Several work proposed hybridization method as solution to cold start problems. A hybrid recommendation system that combines content-based model to assist collaborative filtering algorithms using the derived acoustic features derived from tracks is presented in [10, 26]. Although hybridization can alleviate the cold start problem at a certain level, yet these approaches often more complex, computationally expensive, and lacking of transparency [5].

Another alternative to cold start problems is cross-domain recommendation techniques, which make use of user preferences in other domains from various sources [6, 11]. Additional information such as reading habits, recently watched movies, and even user's personality are exploited to enhance estimation of user's preferences without their long-term profiles [14, 12]. However, a major problem with cross-domain recommendation is the need of multiple user-related data source to estimate user's preferences [9, 7] and that rises concern of privacy and misuses of data collections [15].

Meanwhile, short-term recommendation do not make assumptions that such a user profile is available for content personalization. Short-term recommendation scenario are often referred as next-track music recommendation, playlist continuation, or session-based recommendation [18]. It does not require a user's profile to make suggestion to user, hence it is suitable for ad hoc recommendation to satisfy user short-term goal. A method called CAGH (Collocated Artists–Greatest Hits) that take artists in a given playlist as seed and recommend the most popular tracks of these artists and other similar artists is proposed for generate playlists automatically [3]. Besides, [17] presented a relevance-based playlist continuation using other metadata including playlist title, track's name, and album name; [19] use the similar features but with inverse document frequency (IDF) weighting and k-Nearest Neighbour (kNN) to define their similarity scores to recommend tracks; [20] used tf-idf weighting of metadata including playlist title, playlist description, number of followers of a playlist, track name, artist name, album name, track duration for tracks' ranking calculation with kNN. An approach determine user moods by extracting audio features from the listening history was proposed in [2].

As for Spotify, this company only reveal limited information regarding to their technology in publications. In some literature, it is revealed that Spotify exploited collaborative filtering models to analyze users behavior based on implicit feedback such as the user's interaction with its software including stream counts of track, add track to playlist, visit artists page after listening to a song [21, 23]. Some blog posts report that Spotify actually uses an ensemble of different techniques: collaborative filtering as mentioned above, NLP model for analysing playlist metadata and crawled content relevant to music including news articles, blogs, and etc from the Internet; machine learning techniques that learns representation of raw audio data [4, 8, 16, 24, 28].

It is noteworthy that a paper published by Spotify highlighted the importance of recommending content that users are likely to enjoy at the moment while simultaneously ensuring diversity in their music consumption in long-term, because greater content diversity is strongly associated with key user metrics such as conversion and retention rate to Spotify [1]. Beside diversity, the listening context, for instance the location or activity, and the listening purpose, for instance mood regulation or social boding also affect user satisfaction towards recommended content [25]. For example, users tend not to pick too recent tracks during road trip which is often done in a group where the playlist must satisfy all, and older tracks are more likely to acceptable by youths than vice-versa. Besides, there exist several challenges for music recommendation domain: tracks are shorter, usually listened to multiple times, typically consumed in sessions with other tracks, and relevance is highly context-dependent [13]

In contrast to the previous work, this paper employed a content-based model using metadata and extracted audio features from the tracks with K-Means Clustering, hence make recommendation based on the extracted pattern from the. K-Means clustering is a popular unsupervised Machine Learning (ML) algorithm that group a set of data into clusters by determining similarity between every data points with given k number of centroids. Compare to collaborative method, content-based model can avoid the new user cold start problem. In contrast to other content-based model presented in previous work including Deep Learning and kNN, the advantage of K-Means clustering is relatively simple to implement and guarantees convergence. Additionally, K-Means has great scalability to large dataset. However, the k value must be select carefully to ensure the quality of clusters and it is poor in handling outliers in the data.

The taxonomy of mainstream music recommendation work is shown in Figure 1. The differences and similarities of all related works are summarized as the table 1.

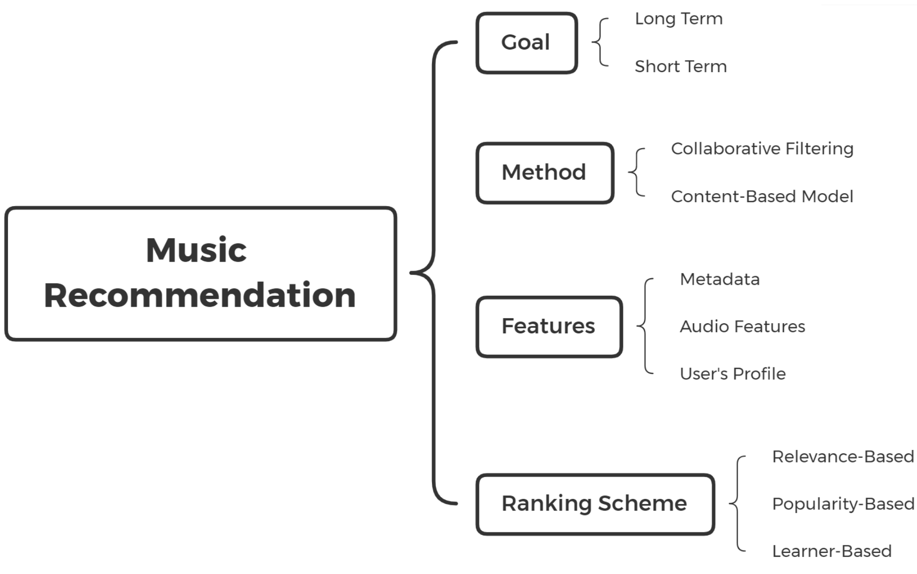

Fig. 1: Taxonomy of mainstream music recommendation methodology.

Table 1: Table of similarities and differences of all related works.

Papers	Method	Features	Goal
[17]	Relevance-based content-based model	Metadata	Short term
[19]	Relevance-based content-based model	Metadata (idf weighting)	Short term
[20]	Relevance-based content-based model	Metadata (tf-idf weighting)	Short term
[2]	Relevance-based content-based model	Audio features	Short term
[3]	Popularity-based content-based model	Metadata	Short term
[10, 26]	Hybrid of content-based and collaborative filtering	Audio features and user's profile	Long Term
Spotify	Hybrid of content-based (NLP, ML) and collaborative filtering	Metadata, audio features, and user's profile	Long term and short term
[6, 11]	Cross-domain collaborative filtering	User's profile from different source	Long term

Methodology

This work use K-Means clustering to extract the pattern from the dataset. All songs are categorized into different genres (clusters) based on their individual properties including both metadata and audio features. Elbow method is used to select the optimal K value that keep lower Sum of Squared Error (SSE) yet do not compromise the number of clusters [22]. Additionally, number of clusters to choose is not so obvious for this dataset where the dimension is high and cluster cannot easily be visualized.The implementation is using Python programming language in Google Colab environment.

First, pairwise correlation in the dataset variables is computed and visualized.. All numerical data are normalized using min-max normalization. Elbow method requires to draw a line plot between SSE and K value a.k.a the number of clusters, the plot is also known as Inertia Plot. In the Inertia Plot, the point where SSE value start decreasing in a linear manner is called as elbow point for estimating optimal K value.

For the recommendation demonstration, the data subset of *Spotify Dataset 1921-2020* named *data by artists* where the data is grouped by artists of the tracks is used. Two columns of user id and artist rating is randomly generated in the data subset. Then, those artists are clustered into 13 clusters. The recommendation is made by calculating user most favourite genres using their rating to artist and similarity between artists genres. An user is randomly picked to recommend artists to him based on his favourite genres. Then the recommended artists are compared to check if the suggested artists appears in his favourite genres list by sorting the his rating to artists.

Table 2: Table of audio features explanation from Spotify Documentation.

Audio Features	**Explanation**
Acousticness	A confidence measure of whether the track is acoustic.
Danceability	Describes how suitable a track is for dancing.
Energy	Represents a perceptual measure of intensity and activity.
Instrumentalness	Predicts whether a track contain no vocals.
Valence	Describing the musical positiveness conveyed by a track.
Tempo	Overall estimated tempo of a track in beats per minute.
Liveness	Detects the presence of an audience in the recording.
Loudness	Overall loudness of a track in decibels (dB).
Speechiness	Detects the presence of spoken words in a track.

Experimental Setup

In the experiment, the calculation of pairwise correlations of data variables are based on pearson coefficient. Min-max normalization is done for all numerical data attributes to range 0 to 1. Optimal K-value is estimated using SSE value at each K value ranging from 1 to 30. However, only ten thousand rows of data is used to calculate the SSE value at every K value to save computation power and time since the full dataset is large. Noteworthy that all random state are fixed with seeding to ensure reproducible results.

A dataset from Kaggle is used in this paper, namely *Spotify Dataset 1921-2020*, it contains more than 170, 000 songs released in between 1921 to 2020 collected from Spotify WebAPI. It is selected because of the up-to-date, rich in information, large size which is essential for data mining, not to mentioned it is publicly accessible. The metadata in this dataset are track name, artist name, release date, duration in milliseconds, popularity, and the extracted audio features includes acousticness, danceability, energy, instrumentalness, valence, tempo, liveness, loudness, speechiness in numerical values. The explanation of each audio features from Spotify are shown in Table 2.

Data Analysis

Data Exploration

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170653 entries, 0 to 170652
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   valence           170653 non-null  float64
 1   year              170653 non-null  int64  
 2   acousticness      170653 non-null  float64
 3   artists           170653 non-null  object 
 4   danceability      170653 non-null  float64
 5   duration_ms       170653 non-null  int64  
 6   energy            170653 non-null  float64
 7   explicit          170653 non-null  int64  
 8   id                170653 non-null  object 
 9   instrumentalness  170653 non-null  float64
 10  key               170653 non-null  int64  
 11  liveness          170653 non-null  float64
 12  loudness          170653 non-null  float64
 13  mode              170653 non-null  int64  
 14  name              170653 non-null  object 
 15  popularity        170653 non-null  int64  
 16  release_date      170653 non-null  object 
 17  speechiness       170653 non-null  float64
 18  tempo             170653 non-null  float64
dtypes: float64(9), int64(6), object(4)
```

Fig. 2: Shape of the dataset and datatype of attributes.

The dataset consist of 19 columns and more than 170,000 rows as shown in Figure 2. The features are metadata of tracks and audio features. Explanation of the audio features can refer to Table 2.

Correlation Analysis

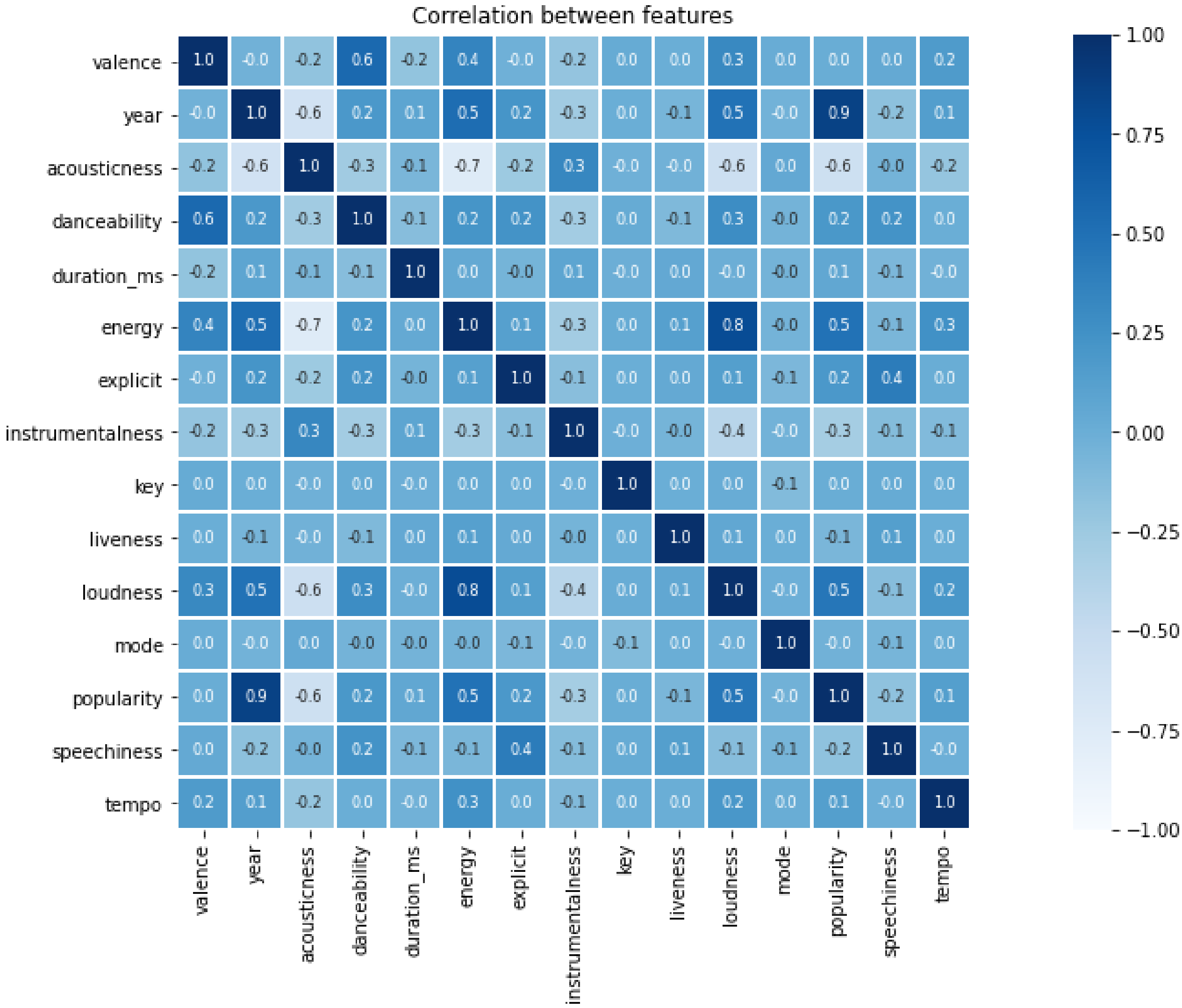

Fig. 3: Correlation Plot of the dataset.

Based on the outcome of the correlation analysis in Figure 3, it is found that year and popularity is highly correlated at 0.9 correlation., followed by energy and loudness at 0.8, valence and danceability at 0.6, hence popularity also correlated with energy and loudness. The possible explanation for year and popularity is due to less songs are released in the early days and recent song also have less popular because it takes time to propagate and gain popularity. Valence and danceability are related because positive tracks are more likely to be played in the context where a group of people are enjoying the music and dancing according to it, instead of playing sad songs. Regarding to energy and loudness correlation, it make sense since energetic tracks tend to be louder, and highly energetic songs are suitable for motivation purpose such as in the gym or stadium which have more chance to expose hence increase the popularity.

Visualization

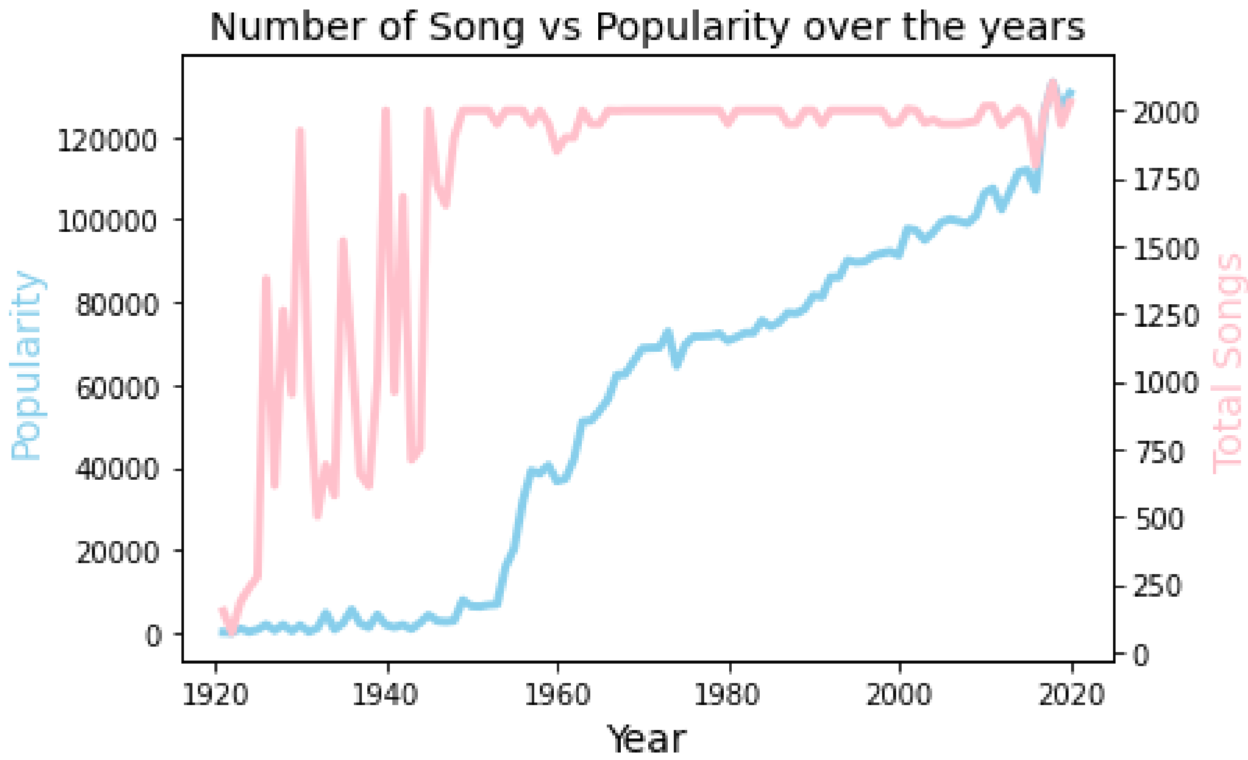

Fig. 4: The number of songs vs popularity over the years.

The visualization of number of songs and popularity over the years in Figure 4 indicates the assumption of year and popularity correlation made above are making some sense. Music streaming platform has empower newly released tracks to reach bigger market and gain popularity through music recommendation, hence it is a win-win situation where the user gets better music consumption experience and the artists also gains commercial value.

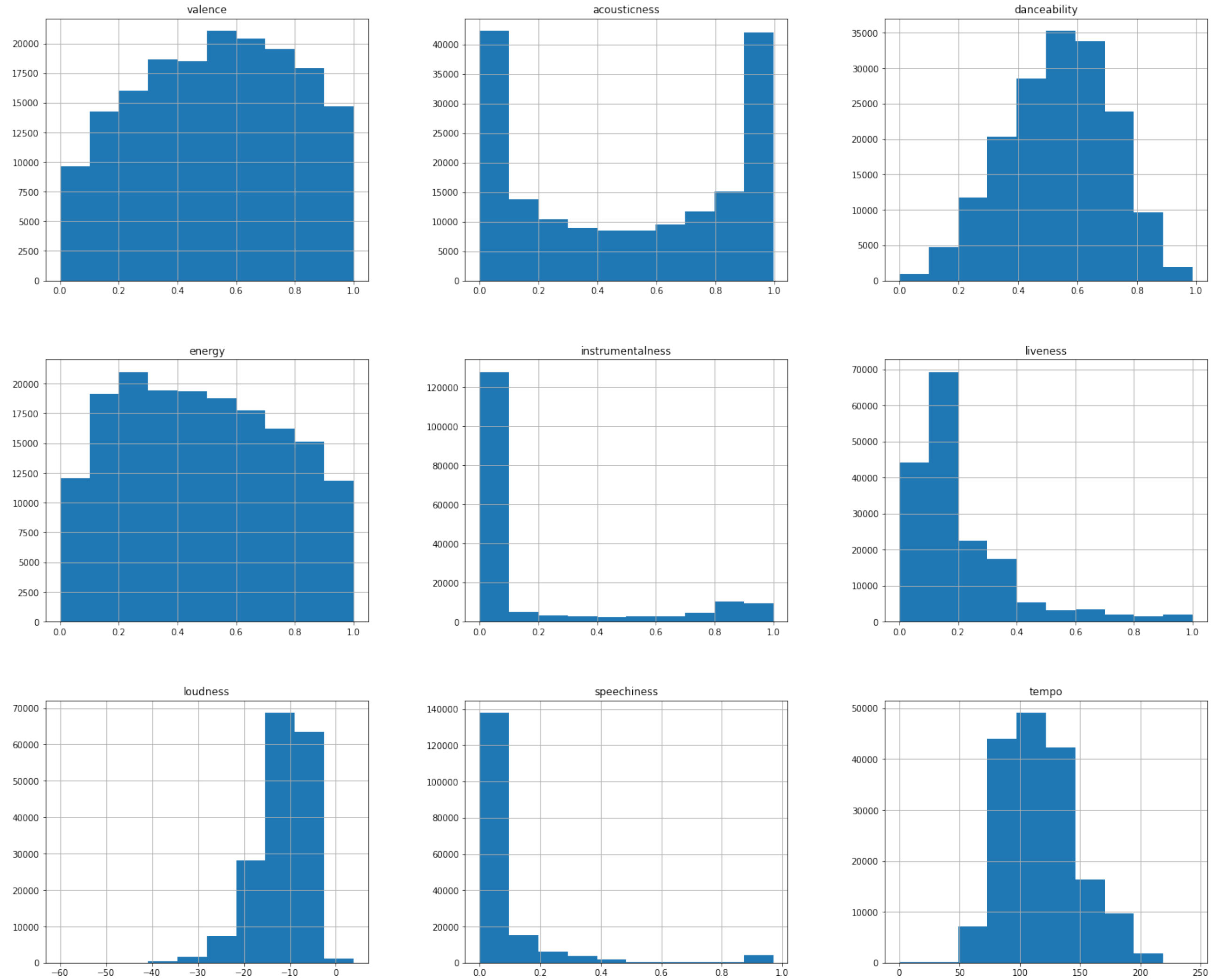

Fig. 5: Histogram of nine audio features.

From the histogram shown in figure 5, all audio features are in numerical value ranged from 0.0 to 1.0 except for loudness and tempo which has range of -60 to 0 and 0 to 250 respectively. Therefore, it is necessary to perform normalization for all the values for these two columns in order to make them compatible with other columns in the vector space.

In the histogram for instrumentalness, it is observed that majority of the value not larger than 0.1, hence the difference between mean and median of instrumentalness is quite large. The same goes to speechiness and liveness.

K-Means Clustering

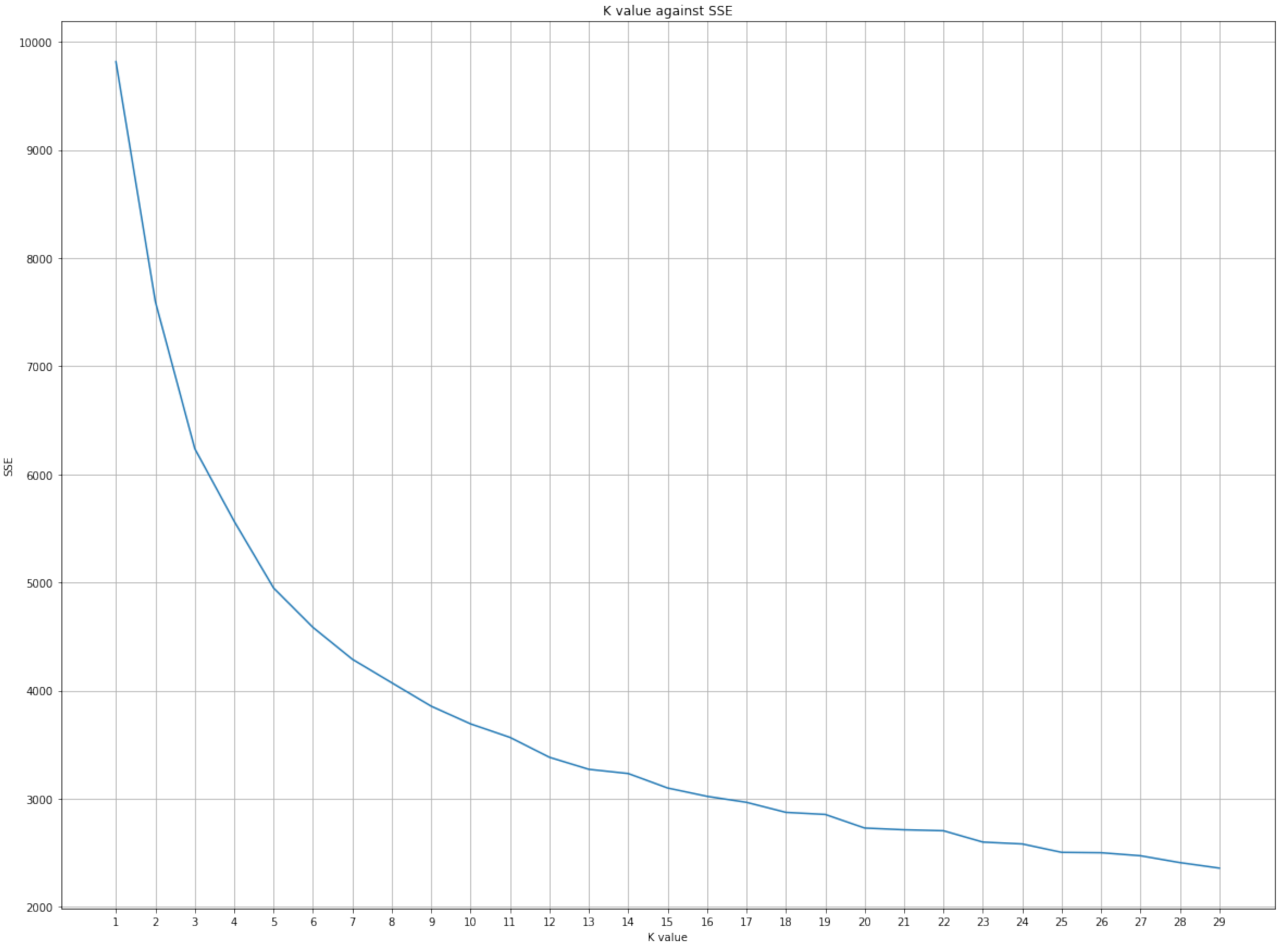

Fig. 6: K value against SSE (Inertia Plot).

One of the most challenging parts of K-Means clustering is deciding the optimal K value, in other words the number of clusters. Elbow methodology is used to determine K value by calculate SSE values from K = 1 and increase iteratively to K = 30. Inertia represents the SSE of data points to their cluster centroid. Small K value and low SSE indicates a successful cluster. Based on the inertia plot in Figure 6, the convergence point with elbow-like shape is not very distinct, since there is a slight curve at K = 20 hence twenty clusters seems reasonable the dataset. Noteworthy that the K-Means clustering used to plot Inertia Plot must set to a fixed seed number to make the model deterministic.

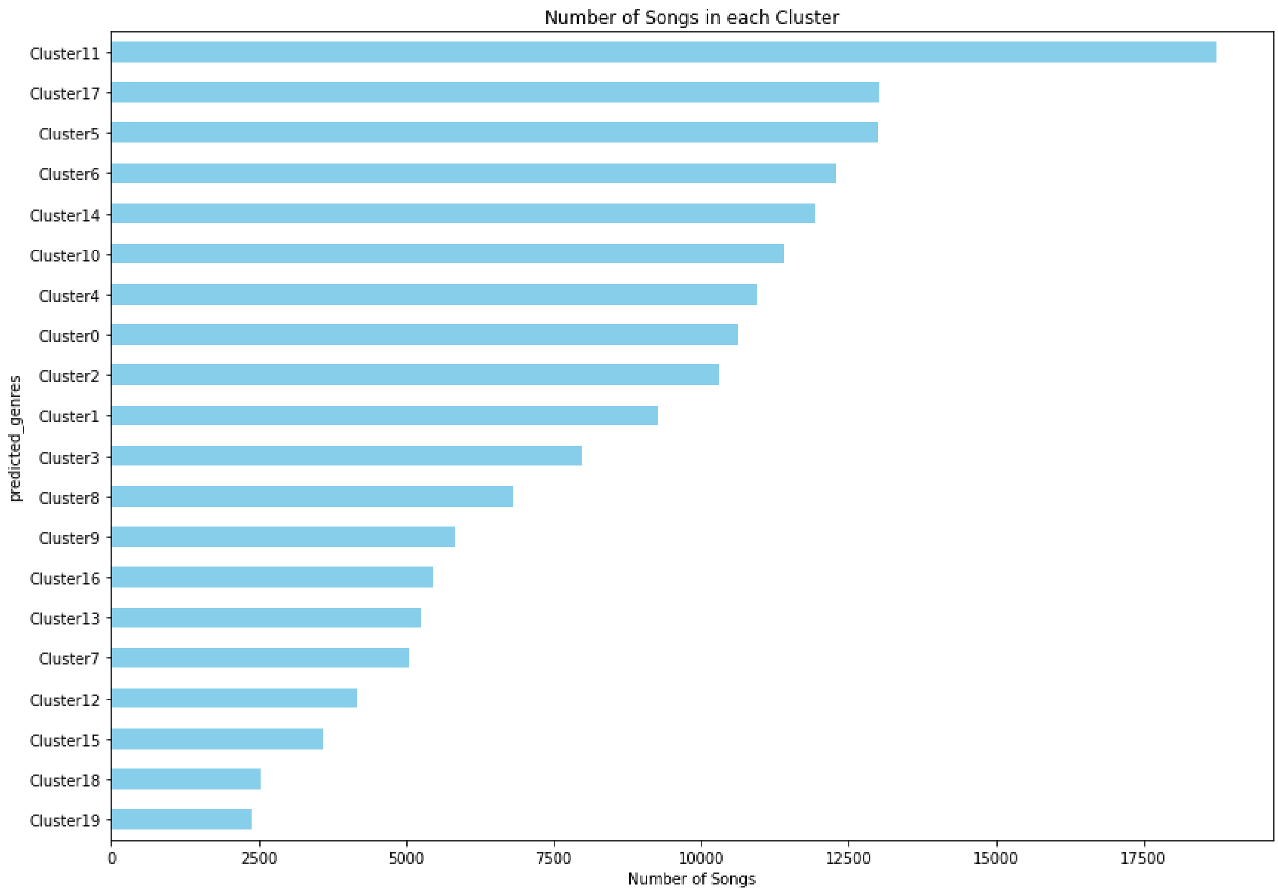

Fig. 7: Number of Songs in each cluster.

The dataset has high dimension, causing difficulties in visualizing all clusters. Therefore, number of songs in each clusters are illustrated instead of the cluster to provide a more intuitive idea regarding to the size of the clusters. Based on the Figure 7, cluster 11 being the largest cluster with approximate 40% more songs than second largest, which is cluster 17 and followed by cluster 5, while cluster 18 and cluster 19 are the smallest cluster with almost identical size.

Recommendation

	artists	mode	count	genres	user_id	rating
23561	Relyk	1.0	4	1	1071	1
8966	Erutan	0.0	2	3	1213	3
12356	Icona Pop	1.0	10	6	1119	4
3618	Bobby Van	0.0	3	3	1015	1
22229	Pest	0.0	2	16	1188	1
10597	Gerardo	0.0	2	7	1256	3
16347	Lee Dixon	1.0	4	12	1010	3
25723	Siegfried Vogel	1.0	1	14	1237	1
25754	Silke Avenhaus	1.0	2	18	1203	5
26388	Splinta	1.0	6	13	1313	5
24292	Ronald Cavaye	1.0	2	18	1029	2
18052	Malvina Reynolds	1.0	4	0	1279	5
15303	Ken Jennings	1.0	4	0	1309	3
32020	lil aaron	1.0	4	6	1011	3
31206	William Dickie	1.0	1	18	1152	3
12227	Hussein Fatal	1.0	1	6	1000	5
31888	Zyce	0.0	2	11	1345	2
3162	Bill Lee	1.0	18	0	1351	1
12714	J's Music	0.0	2	11	1208	4
15656	Klarino-Giorgos Anestopoylos	0.0	2	4	1319	5

Fig. 8: Dateset info (group by artists).

As mentioned in methodology, two columns *user id* and *ratings* is added into the dataset and insert with random generated value for demonstrating the music recommendation. The info of the dataset is as shown in Fig 9.

	artists	genres	rating	count
8551	Ella Fitzgerald	18	5	951
3525	Bob Marley & The Wailers	6	5	536
30187	U2	6	5	420
28541	The Kinks	6	5	388
29019	The Shadows	13	5	304

Fig. 9: Recommended artists based on clustered music genres.

The recommendation is made by determining user's favourite genres based on the clusters. K-Means clustering with K = 20 is performed on the dataset group by artists to group artists into each music genres depending on their tracks features, hence the given recommendation to user is the artists name. Then the recommended artists name can act as a seed to make music recommendation. For this picked user with user id 1111, the suggested artists are from genres (cluster) 18, 6, and 13 as shown in Fig 9.

The selected user id is used to query for his favourite genres in the dataset by sorting all his ratings to

```
array([ 6, 18,  1,  0, 11,  5, 13, 17,  7,  4,  3, 16,  2, 19,  8, 15, 14,
       10, 12,  9], dtype=int32)
```

Fig. 10: User's favourite genres based on randomly generated ratings.

artists from each genres. In Figure 10, the user's favourite genres are 6,18, 1, 0, 11, 5, 13, 17, 7, 4, 3, 16, 2, 19, 8, 15, 14, 10, 12 , 9 in total of 20 favourite genres. Compared to the recommended genres in cluster 18, 6, 13, the recommended artists are within user's favourite genres.

Result

K-Means clustering is used to find relation and patterns in the dataset, hence categorized into different genres (clusters). Based on the outcome, K-Means clustering is effective for clustering musics based on different audio features, and it is convenient to implement compared to other algorithms. By using elbow method, the optimal K value is found at K = 20.

This is a relevance-based recommendation which use similarity between two content to make suggestion, if two tracks are close in a vector space then they are likely to be identical genres. In other works, a person who listen music from a cluster might be delighted to hear similar characteristic music from the recommended artist. In this work, the recommendation is made only up to artists due to the time constraints, and using artists as seed for recommendation is effective because an artists tend to produce or create musics with similar characteristics.

Conclusion

This paper has presented K-Means clustering on the audio features and metadata in *Spotify Dataset 1921-2020*, to make recommendation based on musics genres relevancy. It is found that K = 20 is suitable for performing K-Means clustering on this dataset by using elbow method through inspecting the line plot of SSE and K value. The work shows that simple unsupervised ML algorithm like K-Means clustering is applicable for music recommendation purpose and easier to implement.

Future work may consider to test with bigger range of K value to further experiment if there is better choice of number of clusters. Besides, future work should try to reduce dimensionality of the data because it is found correlation between certain audio features, hence Principal Component Analysis could help to remove redundant information in the features and have better representation of musics genres to make recommendations more sophisticated and precise for a more complete music consumption experience. Additionally, cross-validation such as k-fold cross-validation could be use for more thoroughly evaluation of cluster quality by performing clustering repeatedly on splitted subset.

*

References

[1] Ashton Anderson et al. "Algorithmic Effects on the Diversity of Consumption on Spotify". In: *Proceedings of The Web Conference 2020*. ACM, Apr. 2020. DOI: 10.1145/3366423.3380281. URL: https://doi.org/10.1145/3366423.3380281.

[2] Mahta Bakhshizadeh et al. "Automated Mood Based Music Playlist Generation By Clustering The Audio Features". In: *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, Oct. 2019. DOI: 10.1109/iccke48569.2019.8965190. URL: https://doi.org/10.1109/iccke48569.2019.8965190.

[3] Geoffray Bonnin and Dietmar Jannach. "Automated Generation of Music Playlists: Survey and Experiments". In: *ACM Computing Surveys* 47.2 (Jan. 2015), pp. 1–35. DOI: 10.1145/2652481. URL: https://doi.org/10.1145/2652481.

[4] Clark Boyd. *How Spotify Recommends Your New Favorite Artist*. Nov. 2019. URL: https://towardsdatascience.com/how-spotify-recommends-your-new-favorite-artist-8c1850512af0.

[5] Robin Burke. "Hybrid Web Recommender Systems". In: *The Adaptive Web*. Springer Berlin Heidelberg, pp. 377–408. DOI: 10.1007/978-3-540-72079-9_12. URL: https://doi.org/10.1007/978-3-540-72079-9_12.

[6] Iván Cantador and Paolo Cremonesi. "Tutorial on cross-domain recommender systems". In: *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*. ACM Press, 2014. DOI: 10.1145/2645710.2645777. URL: https://doi.org/10.1145/2645710.2645777.

[7] Iván Cantador et al. "Cross-Domain Recommender Systems". In: *Recommender Systems Handbook*. Springer US, 2015, pp. 919–959. DOI: 10.1007/978-1-4899-7637-6_27. URL: https://doi.org/10.1007/978-1-4899-7637-6_27.

[8] Sophia Ciocca. *How Does Spotify Know You So Well?* Apr. 2020. URL: https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe.

[9] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. "Cross-Domain Recommender Systems". In: *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, Dec. 2011. DOI: 10.1109/icdmw.2011.57. URL: https://doi.org/10.1109/icdmw.2011.57.

[10] Justin Donaldson. "A hybrid social-acoustic recommendation system for popular music". In: *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*. ACM Press, 2007. DOI: 10.1145/1297231.1297271. URL: https://doi.org/10.1145/1297231.1297271.

[11] Ignacio Fernández-Tobías et al. "Alleviating the new user problem in collaborative filtering by exploiting personality information". In: *User Modeling and User-Adapted Interaction* 26.2-3 (Feb. 2016), pp. 221–255. DOI: 10.1007/s11257-016-9172-z. URL: https://doi.org/10.1007/s11257-016-9172-z.

[12] Bruce Ferwerda et al. "Personality Traits Predict Music Taxonomy Preferences". In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, Apr. 2015. DOI: 10.1145/2702613.2732754. URL: https://doi.org/10.1145/2702613.2732754.

[13] Casper Hansen et al. "Contextual and Sequential User Embeddings for Large-Scale Music Recommendation". In: *Fourteenth ACM Conference on Recommender Systems*. ACM, Sept. 2020. DOI: 10.1145/3383313.3412248. URL: https://doi.org/10.1145/3383313.3412248.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets". In: *2008 Eighth IEEE International Conference on Data Mining*. Ieee. 2008, pp. 263–272.

[15] Omar Hujran et al. “Big Data and Its Effect on the Music Industry”. In: *Proceedings of the 3rd International Conference on Software Engineering and Information Management*. ICSIM ’20. Sydney, NSW, Australia: Association for Computing Machinery, 2020, pp. 5–9. ISBN: 9781450376907. DOI: 10.1145/3378936.3378957. URL: https://doi.org/10.1145/3378936.3378957.

[16] Chris Johnson. *Algorithmic Music Recommendations at Spotify*. Jan. 2014. URL: https://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify.

[17] Surya Kallumadi, Bhaskar Mitra, and Tereza Iofciu. “A Line in the Sand: Recommendation or Ad-Hoc Retrieval? Overview of RecSys Challenge 2018 Submission by Team BachPropagate”. In: *Proceedings of the ACM Recommender Systems Challenge 2018*. RecSys Challenge ’18. Vancouver, BC, Canada: Association for Computing Machinery, 2018. ISBN: 9781450365864. DOI: 10.1145/3267471.3267478. URL: https://doi.org/10.1145/3267471.3267478.

[18] Iman Kamehkhosh, Geoffray Bonnin, and Dietmar Jannach. “Effects of recommendations on the playlist creation behavior of users”. In: *User Modeling and User-Adapted Interaction* 30.2 (May 2019), pp. 285–322. DOI: 10.1007/s11257-019-09237-4. URL: https://doi.org/10.1007/s11257-019-09237-4.

[19] Domokos M. Kelen et al. “Efficient K-NN for Playlist Continuation”. In: *Proceedings of the ACM Recommender Systems Challenge 2018*. RecSys Challenge ’18. Vancouver, BC, Canada: Association for Computing Machinery, 2018. ISBN: 9781450365864. DOI: 10.1145/3267471.3267477. URL: https://doi.org/10.1145/3267471.3267477.

[20] Malte Ludewig et al. “Effective Nearest-Neighbor Music Recommendations”. In: *Proceedings of the ACM Recommender Systems Challenge 2018*. RecSys Challenge ’18. Vancouver, BC, Canada: Association for Computing Machinery, 2018. ISBN: 9781450365864. DOI: 10.1145/3267471.3267474. URL: https://doi.org/10.1145/3267471.3267474.

[21] Mithun Madathil. “Music Recommendation System Spotify-Collaborative Filtering”. In: *Reports in Computer Music. Aachen University, Germany* (2017).

[22] D. Marutho et al. “The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News”. In: *2018 International Seminar on Application for Technology of Information and Communication*. 2018, pp. 533–538. DOI: 10.1109/ISEMANTIC.2018.8549751.

[23] Javier Pérez-Marcos and Vivian López Batista. “Recommender System Based on Collaborative Filtering for Spotify’s Users”. In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, July 2017, pp. 214–220. DOI: 10.1007/978-3-319-61578-3_22. URL: https://doi.org/10.1007/978-3-319-61578-3_22.

[24] Dieleman Sander. *Recommending music on Spotify with deep learning*. Aug. 2014. URL: https://benanne.github.io/2014/08/05/spotify-cnns.html.

[25] Markus Schedl et al. “Current challenges and visions in music recommender systems research”. In: *International Journal of Multimedia Information Retrieval* 7.2 (Apr. 2018), pp. 95–116. DOI: 10.1007/s13735-018-0154-2. URL: https://doi.org/10.1007/s13735-018-0154-2.

[26] Bo Shao et al. “Music Recommendation Based on Acoustic Features and User Access Patterns”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.8 (Nov. 2009), pp. 1602–1611. DOI: 10.1109/tasl.2009.2020893. URL: https://doi.org/10.1109/tasl.2009.2020893.

[27] Y. Song, S. Dixon, and M. Pearce. “A Survey of Music Recommendation Systems and Future Perspectives”. In: 2012.

[28] Valerio Velardo. *Spotify’s Discover Weekly explained-Breaking from your music bubble or, maybe not?* Feb. 2019. URL: https://medium.com/the-sound-of-ai/spotifys-

discover-weekly-explained-breaking-from-your-music-bubble-or-maybe-not-b506da144123.