Using PCA and K-Means to Predict Likeable Songs from Playlist Information

Caroline Langensiepen, Adam Cripps, Richard Cant
School of Science and Technology
Nottingham Trent University
Nottingham, United Kingdom
Caroline.Langensiepen@ntu.ac.uk

***Abstract*—Most recommendation systems for music rely on individual song ratings. Current song recommendation software that uses playlists has shown to either be inaccurate or suggest songs that are extremely like those in the playlist already. Furthermore, this recommendation software tends to rely very large numbers of records. AI models are used to overcome these limitations using substantially less data. A collaborative filtering approach using two different models (K-means and hierarchical clustering) is used to separate playlist data into clusters for comparison. After the data has been clustered, a Euclidean distance measure is used between the songs in the cluster and the average values of the songs in a single users playlist to make the final predictions. The use of normalisation and PCA enabled the K-means and hierarchical clustering models to form clusters efficiently. When tested on a small sample of users, the system recommended songs that were considered likeable by the users 60% of the time, while still finding songs that were generally diverse.**

Keywords–K-means, clustering, recommender systems, PCA

I. Introduction

Music streaming services such as Spotify and iTunes contain over 30 million songs, yet it has been estimated that on average around 90% of the music we listen to is music that we have heard before [1]. This suggests that either users prefer to stick with what they know, or that the recommendations they receive do not adequately introduce them to different music. The aim of this investigation was to assess whether artificial intelligence models could identify the features that determine whether a person likes a piece of music, whether they could accurately predict songs likeable by a user, and whether songs could be found that were diverse and not obvious. Additional aims were that these predictions could be made from playlists without the need for much larger databases of historical user listening data. Section 2 reviews current recommendation systems and work in this area, section 3 discusses the models used, section 4 presents the results and discussion and section 5 draws some conclusions and suggestions for further work.

II. Previous Work

Ahn [2] investigated the use of genre and popularity information to create a hybrid recommendation system, comparing the results with a collaborative filtering approach for cold starting (new users/information) and sparse data. They concluded that the proposed hybrid approach outperforms that of collaborative filtering specifically for those situations. Liu [3] explored radio station and music recommendation systems for use in cars. They suggested a system using preference records of users with similar backgrounds to produce recommendations. Bayesian and Neural Network (NN) approaches were used. It was found that the NN approach was more accurate after historical data had been captured, though there was little diversity in the recommended songs. The Bayesian model performed well and had greater diversity as it used others' music preferences. There were poor recommendations from a cold start with both attempted approaches, but the work was done on small data sets. Cilibrasi [4] investigated the effectiveness of hierarchical clustering based on the compression of strings that represented the music pieces to establish similarities in pieces of music. The authors discussed the importance of clustering to the development of recommendation systems, and sought to capture multiple distance metrics (Hamming, Euclidian, Lempel-Ziv etc.) in just one metric. They found hierarchical clustering to be the most informative, given that other models distorted the visual representations of results. Wolff, and Weyde [5] used relative user ratings to identify music similarity. They suggested that this performs better than absolute rating by avoiding some well-known problems. They used several models, evaluated the effectiveness of using different audio features and genre data and applied dimensionality reductions. They concluded that Timbral and Music Structural features were the most effective but that all features jointly outperformed any combinations of subsets of features. Li et al [6] worked on identifying music similarity from high dimensional data and from a diverse range of information sources. They used a semi-supervised model to make classifications. Results showed that feature level methods performed worse than content-only or lyric-only clustering methods.

A. Relevant Commercial Software

Deezer is a software platform that enables users to stream music and create playlists. Deezer claim that the more songs you listen to the better the song suggestions will be. There is currently no information relating to how the algorithms work or how successful they have been in recommending songs,

0-7695-6405-4/18/$31.00 ©2018 IEEE
DOI 10.1109/UKSim.2018.00017

but there is a general consensus on social media that they work well. This apparent success however may be because Deezer only suggests very similar songs to the current set and not more diverse ones. Gracenote provides REST API access to a database consisting metadata for many songs and is used by companies like Spotify, Apple and Amazon. Recently, Gracenote has also started to develop software to suggest radio, tracks and playlists through analysis of music descriptors such as genre, mood and era. There is currently no information regarding how successful this software is for suggesting playlists to its users. Spotify is an online music provider that gives its users the ability to listen to music consisting of millions of tracks across a multitude of platforms. Spotify also has a REST API for developers that allows the gathering of information about songs and playlists. Spotify also provides a feature called Discover Weekly which generates a playlist for the user based on their profile characteristics, others' playlists and the songs that the user repeatedly listens to. Quartz [7] conducted an interview with Spotify regarding Discover Weekly. Spotify explained that the software mainly works on other peoples playlists of which there are about 2 billion. If songs you like appear in another playlist then it will suggest songs from that playlist. It gives extra weight to playlists with more followers and those created by Spotify. Quartz noted that the same playlists are suggested to multiple people because trends in music have a big influence on what is suggested.

III. Proposed Approach

In order to determine whether AI approaches could provide improved recommendations for music while not requiring very large data repositories, two assumptions are made. Firstly, that music tastes can be identified by a set of song features. Identifying these is a significant challenge because of the high dimensionality of the tags/attributes that can be associated with a single piece of music. Secondly that a playlist captures the breadth of that users tastes, and that playlists can be separated from one another and clustered.

A. Data Acquisition and Understanding

Data was acquired as anonymously supplied individuals' playlists, and features extracted for each song using APIs from Spotify and Gracenote. Unfortunately the numbers acquired were small, so the work also relied on the publicly available Spotify playlists as notional users' preferences. The Spotify API provided information about duration, danceability, energy, tKey, loudness, mode, speechiness, acousticness, instrumentalness, liveness and valence. The Gracenote API provided information about year, language, album title, genres, moods, origins, eras, types and tempos. Once the data had been collected, a detailed overview of the data was compiled as shown in Table I and II, indicating that it was complex and had many dimensions. A playlist may have 20+ songs where each song had multiple genres, artists, types, origins and many other features.

B. Data Pre-processing

For most of the fields shown in Table I and II, the data was converted to a percentage for normalisation, because of the very different data values for each feature. For the later fields shown, separate features of High and Low were used, for example LoudnessLow, LoudnessHigh. Null or repeated playlists were removed. Because the data still had a very high number of dimensions, when clustering was attempted the results were extremely poor. Principal Component Analysis (PCA) was therefore applied to reduce the complexity further to a level that would be more usable by the models. Using 20 principle components covered just under 80% (0.798) of the variance in the data. Although 20 features is still a large number of dimensions, it is considerably better than the 63 produced from the original data. The results of the PCA are shown in Table 3 with the major positive and negative weights that contribute to each component. A total of 16 outliers were identified within the data but not removed. Outliers represented playlists that were different from the others.

C. Models

A collaborative filtering approach was taken using a simple K-means model [8] and evaluating it against a hierarchical clustering model. These two unsupervised learning models were chosen because there was no previous classified data to work from. The Orange datamining tool was used to apply the models to the public data. A simple decision tree algorithm was then used on the individuals' data to map it to clusters by imputing the class (cluster) from the features. For the K-means model various numbers of clusters were attempted. It was found that the best-defined clusters occurred when 9 clusters were used. The parameters were set to 100 re-runs with 1000 iterations. Initialising the cluster centroids can also have a big effect on the resulting model. This is because they may not reach the optimum positions if they are initialised poorly. The best results were found when the centroids were initialised using KMeans++ to carefully pick the initialisation points [9]. A hierarchical clustering model was also trained on the public data to form a comparison. A cosine distance metric was applied to the data and then the model was trained with complete linkage. A total of 9 clusters were formed. The data was not clustered in exactly the same way as the K-means model though the resulting clusters were very similar. After the clustering models had been trained on the public data it was combined with the individuals' data. The clusters for the individuals' data were imputed using a simple decision tree so that new data could be used with these models without having to constantly rerun the clustering algorithms. These new sets of data contained just the clusters and playlist ids.

TABLE I
INDIVIDUAL USER DATA (RAW)

Feature	Amount		
Playlists	36		
Male:Female:Unknown	19:14:2		
Manual:Spotify:Unknown	9:9:18		
18-25 : 26-50 : 50+	21 : 9 : 6		
Albums	319		
Artists	307		
Eras	23		
Genres	129		
Moods	87		
Origins	96		
Songs	322		
Tempos	392		
Types	9		
Feature	Min	Max	Average
Year	1971	2016	N/A
Duration	37733	590000	225912
Danceability	0.153	0.924	0.599
Energy	0.035	0.993	0.667
TKey	0	11	5.19
Loudness	-28.46	-0.839	-6.79
Mode	0	1	0.632
Speechiness	0.027	0.469	0.095
Acousticness	0	0.988	0.222
Instrumentalness	0	0.985	0.035
Liveness	0.032	0.894	0.171
Valence	0.036	0.980	0.510

TABLE II
PUBLIC DATA (RAW)

Feature	Amount		
Playlists	750		
Male:Female:Unknown	0:0:750		
Manual:Spotify:Unknown	0:750:0		
18-25 : 26-50 : 50+	N/A		
Albums	1781		
Artists	1629		
Eras	26		
Genres	205		
Moods	101		
Origins	166		
Songs	2057		
Tempos	2221		
Types	9		
Feature	Min	Max	Average
Year	1968	2017	N/A
Duration	68293	1009360	231598
Danceability	0.083	0.957	0.623
Energy	0.009	0.995	0.691
TKey	0	11	5.12
Loudness	-33.7	-0.919	-6.35
Mode	0	1	0.597
Speechiness	0.023	0.791	0.083
Acousticness	0	0.996	0.192
Instrumentalness	0	0.989	0.043
Liveness	0.017	0.969	0.181
Valence	0.032	0.976	0.505

An algorithm was written to make song predictions based on the individual's playlist averages and the distances between other songs in their identified cluster. For each playlist in the cluster, all songs were extracted from the database. The data was then normalised between the values of 0 and 1 using the function below. This reduces biasses in the data when computing the distances to make predictions.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

The average values for the user's songs were then calculated and 3 song predictions, "closest", "middle" and "furthest" made for each model using the Euclidean distance D as defined in equation 2. This is based on the following song features: duration, danceability, energy, tKey, loudness, mode, speechiness, acousticness, instrumentalness, liveness and valence. For each song in a cluster:

$$D = \sqrt{\sum_i (x_i - \overline{x}_i)^2} \tag{2}$$

Where the x_i are the features for a particular song and the $\overline{x}_i$ are the average features for the user's playlist.

The closest (minimum D) should be a song that the user is familiar with and should definitely like, whereas the middle (median D) and furthest (maximum D) are songs that may be less familiar to the user yet still likeable. The songs were also limited to the languages of the songs that were submitted to avoid songs being predicted that the user could not understand. For a set of users, the K-means and hierarchical cluster results were passed through these algorithms to make song predictions for the users. A further test was constructed to see the effect of removing one song from a playlist on its cluster assignment. The new data was then assigned to clusters by using the simple decision tree in the same way as the real data is assigned clusters. The resulting clusters were then compared to the unmodified data and the accuracy of the assignment was evaluated.

IV. RESULTS AND ANALYSIS

Once the data had been pre-processed it was passed through the AI models to cluster the playlists and identify which features are the most important to each cluster. Both K-means and hierarchical clustering models were successfully trained and outputted clear clusters in the data with high agreement.

A. K-Means

Due to the high dimensionality of the data it was difficult to visualise the K-means clusters on a 2D plane. A few figures are thus used to show the most useful generated 2D representations of the clustered data. The background colours show the cluster density and are coloured to match the representative cluster. Figure 1 shows the best representation generated of the data before PCA was applied which

TABLE III
PCA ANALYSIS

	Major Positive	Major Negative
PC1	20th Century (0.216), Alternative (0.171)	DanceabilityLow (-0.342), DurationLow (-0.304), EnergyLow (-0.298), LivenessLow (-0.292) SpeechinessLow (-0.269), ValenceLow (-0.256)
PC2	LowKey (0.295), AcousticnessLow (0.259)	EnergyHigh (-0.304), ValenceHigh (-0.303), DanceabilityHigh (-0.289), YearHigh (-0.262), YearLow (-0.261), HighKey (-0.225)
PC3	LoudnessLow (0.260), 21Century (0.241), LoudnessHigh (0.232), Energetic (0.231)	DurationHigh (-0.267), Mode (-0.207)
PC4	HipHopRap (0.397), Country (0.365),	Europe (-0.288), Dance (-0.208)
	NorthAmerica (0.236), Dramatic (0.232)	
PC5	AcousticnessHigh (0.357), Pop (0.292), Female (0.276), Slow (0.264)	EnergyLow (-0.226), Fast (-0.206)
PC6	InstrumentalnessLow (0.333), Calm (0.268), Male (0.262)	
PC7	Dramatic (0.338), YearLow (0.252), YearHigh (0.249), Mode (0.229)	DanceabilityHigh (-0.249), Passionate (-0.216), Reggae (-0.212)
PC8	Alternative (0.272), Male (0.246), Fast (0.241),	Medium (-0.277), Country (-0.257), R&BSoul (-0.213)
	NorthAmerica (0.238), 21StCentury (0.232)	
PC9	Africa (0.574), Blues (0.542)	
PC10	Australia (0.369), Aggressive (0.293)	Sad (-0.341), 20thCentury (-0.239) ValenceLow (-0.229), Regge (-0.216)
PC11	Fast (0.309), Jazz (0.303), R&BSoul (0.241)	AcousticnessLow (-0.275), Dark (-0.224)
PC12	Australia (0.481), Aggressive (0.379), Calm 0.249)	Energetic (-0.247), Happy (-0.211)
PC13	R&BSoul (0.477)	Mixed (-0.3719), InstrumentalnessHigh (-0.291)
PC14	Classical (0.328), Other (0.274)	Electronic (-0.263), Slow (-0.243), Sad (-0.231), Mixed (-0.216)
PC15	Other (0.369), Aggressive (0.288)	Calm (-0.371), Asia (-0.359), Reggae (-0.207)
PC16	SouthAmerica (0.317), Electronic (0.267), Calm (0.257), Female (0.232)	Mixed (-0.278), Aggressive (-0.268), InstrumentalnessLow (-0.208), Asia (-0.205)
PC17	SouthAmerica (0.338), Europe (0.226)	Dark (-0.262), Electronic (-0.237), Classical (-0.213), SpeechinesHigh (-0.207), NorthAmerica (-0.205)
PC18	Dark (0.434), Slow (0.359), Europe (0.259), SouthAmerica (0.246)	NorthAmerica (-0.254), Happy (-0.240), Rock (-0.217)
PC19	SouthAmerica (0.312), Male (0.310), 20thCentury (0.232)	Reggae (-0.316), Female (-0.269)
PC20	SouthAmerica (0.508), Other (0.308), InstrumentalnessLow (0.245)	Classical (-0.319), Energetic (-0.317), 20thCentury (-0.221)

was substantially more confusing than those produced after PCA. This showed how high dimensional data is difficult for the models to cope with and led to a convoluted clustering being performed.

After PCA was applied, the data was clustered again using K-means as shown in Figures 2,3,4. The K-means cluster representations show clear divisions and separations in the clusters. When looking at different sets of principal components, the various clusters diverge from one another. The clusters are projected onto a 2D plane to show this divergence and so that a more detailed analysis against the principal components can be made.

Figure 5 demonstrates the divergence of different clusters against the identified principal components. Table 4 shows the clusters and the features that Figure 5 show are most relevant to the music tastes of the playlists within.

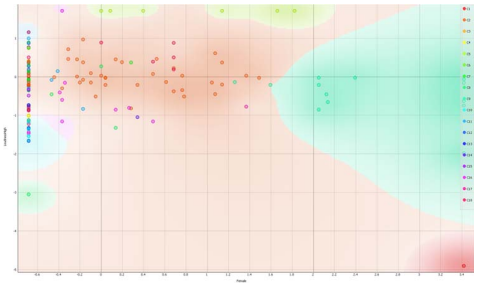

Figure 1. K-Means Before PCA

B. Hierarchical Clustering

A hierarchical clustering model was applied to the data with 9 clusters so that a comparison of the two models could

TABLE IV
K-MEANS PCA ANALYSIS

Cluster	Features Most Components	Identification Genre
1	PC6, PC8, PC9, PC11	Upbeat Fast
2	PC1, PC5, PC7, PC8, PC11,PC16, PC17, PC20	Mixed
3	PC3, PC5	Loud Energetic Pop
4	PC1, PC3	Loud Energetic Alternative
5	PC1, PC14, PC15, PC18	Classical Alternative Dark Aggressive
6	PC2, PC4, PC10, PC14	Aggressive Loud Energetic
7	PC3, PC4	Loud Energetic HipHop
8	PC1, PC2, PC5, PC7, PC16	Female Instrumental Pop
9	PC3, PC6	Male Loud Energetic

TABLE V
HIERARCHICAL PCA ANALYSIS

Cluster	Features Most Relevant	Identification Genre
1	PC20, PC10, PC12	Aggressive
2	PC1, PC7, PC8, PC13, PC14, PC15	Dramatic Alternative
3	PC4, PC7	Dramatic HipHop
4	PC6, PC12	Aggressive Male Instrumental
5	PC3, PC6, PC18	Dark Loud Energetic
6	PC3, PC5, PC6, PC12	Loud Aggressive Pop
7	PC1, PC2, PC4, PC5, PC15, PC13	Instrumental Pop/R&B
8	PC10, PC14, PC18	Dark Aggressive
9	PC1, PC4, PC10,	Dramatic Aggressive HipHop

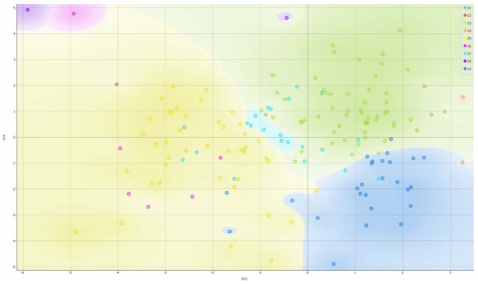

Figure 2. K-Means Cluster Representation 1

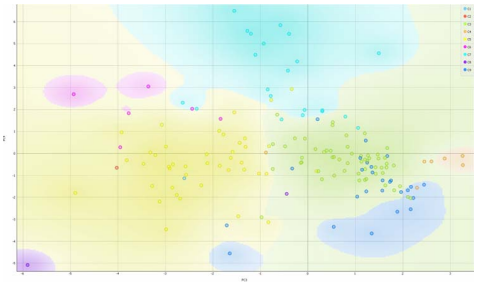

Figure 3. K-Means Cluster Representation 2

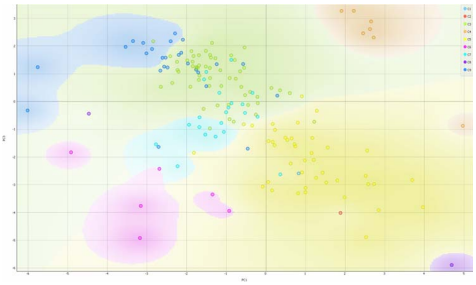

Figure 4. K-Means Cluster Representation 3

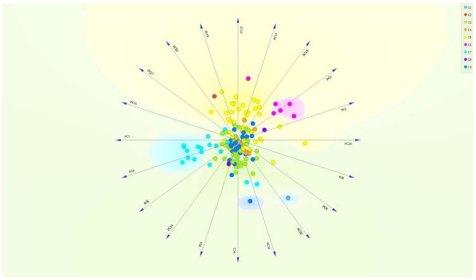

Figure 5. K-Means Clusters Mapped with PCA

be made. The Euclidean, Manhattan and Cosine distance metrics were all tested; the Cosine distance metric with complete linkage provided the best results. The clusters produced were evenly sized and they were easy to differentiate between. Much the same as K-Means, clustering of the data before PCA was applied resulted in very poor cluster results as shown in Figure 6.

Once the data had been mapped onto the PCA space, it was re-clustered and, as with K-means, the clusters were much more distinct than in the original form (Figure 7).

The resulting clusters were also projected onto a 2D plane against the PCA components for further analysis (Figure8). Again, a cluster table was created to identify the most

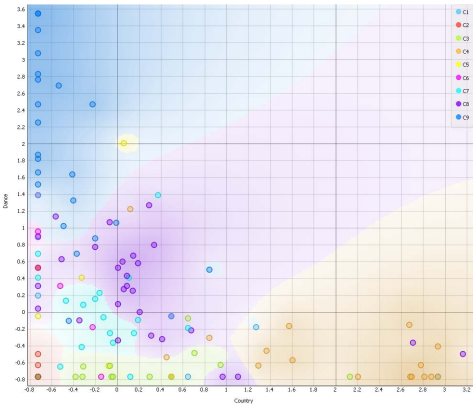

Figure 6. Hierarchical Clusters Before PCA

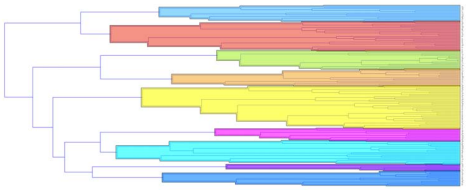

Figure 7. Hierarchical Clusters

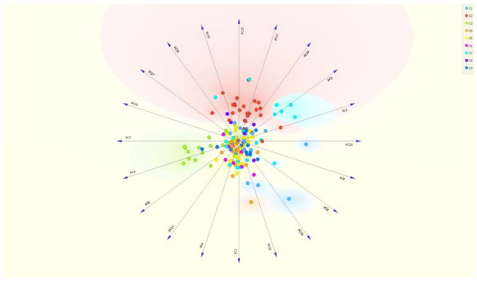

Figure 8. Hierarchical Clusters Mapped with PCA

prominent features as shown in Table 5.

When looking at the PCA projections it is obvious that both models clustered the data similarly, with only minor differences. This is a good indication that the data has been clustered well and that trends in the data have been identified by both models.

C. *"Leave one out" test*

The removal of one song and reclustering was only done on playlists with 20 songs or more. The result of this test showed 89% correctly classified for K-means data and 92% for hierarchical clustering. These results show that the proposed decision tree algorithm to assign clusters is accurate in assigning playlists to clusters, providing the playlist is large enough.

D. *User feedback on song predictions*

Users that had submitted playlists had 6 songs predicted (3 K-means and 3 hierarchical) and presented to them. The initial results were positive with around 60% of the songs predicted by K-means and 53% hierarchical clustering were liked by the participants. Some of the predictions were new to the users and after listening they often gave feedback that they liked the song. This shows that the methods used are predicting diverse songs. However only 5 participants gave feedback, so any conclusion from these results should be treated with caution. Most of the predicted songs differed between the two models and only on 2 occasions did the models agree and predict the same song.

V. Conclusions and Future Work

AI models and distance metrics achieved some success in identifying songs that users like based on a relatively small number of playlist submissions. K-means and hierarchical clustering models were used to separate playlists effectively into different clusters. Projections onto 2D images showed that the use of PCA with 20 components (maintaining around 80% variance) considerably improved the clustering models. There was a high degree of agreement between the two models which suggested that significant clusters had be identified within the data. The features that determine whether a person likes a piece of music were identified through mapping clusters against PCA components and evaluating the weights of the features that contribute the most to a particular component against the position of that cluster. Furthermore, it found that for each cluster there was a different set of features of high importance. New data that had not been clustered was assigned to clusters using a simple decision tree. The effectiveness of this assignment is tested through removing a song from each playlist and then testing to see if the playlists are assigned to the correct cluster. Larger playlists were clustered correctly 90% of the time across both models. The algorithms predicted songs with successfully based on a relatively small set of playlists and song information. This is in contrast to current recommendation software that relies on millions of playlists to make informed decisions. It has therefore demonstrated that songs can be predicted accurately based on a relatively small amount of historical data using both K-means and hierarchical clustering. After further data and more feedback is collected the models could be optimised, and the distance metric improved.

References

[1] Elizabeth Margulis "On Repeat - How Music Plays the Mind" Oxford University Press ISBN 9780199990825

[2] Ahn, H. 2006. Utilizing Popularity Characteristics for Product Recommendation. International Journal of Electronic Commerce. Volume 11 (Issue: 2) p. 59

[3] Liu, N. 2014. Car radio channel recommendation system. Design of an intelligent car radio and music player system Springer Science & Business Media Volume 72 (Issue: 2) p. 1341.

[4] Cilibrasi. R., Vitani. P., R de Wolf. 2004. Algorithmic clustering of music. Computer Society. Fourth International Conference on Web Delivering of Music.

[5] Wolff, D. and Weyde, T. 2013. Learning music similarity from relative user ratings. Springer Science & Business Media Volume 17 (Issue: 2) p. 109.

[6] Li. T. Ogihara, M. Peng, W. Shao, B. and Zhu, S. (2009) Music Clustering with Features from Different Information Sources Transactions on Multimedia Volume 11 (Issue: 3) p.447.

[7] "The magic that makes Spotifys Discover Weekly playlists so damn good" Available at:https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good/. Last Accessed: 18/2/2018

[8] MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281297

[9] Arthur, D., Vassilvitskii, S. 2006. K-means++: The Advantages of Careful Seeding Available at: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf. Last Accessed: 18/2/2018.