Final Project

Chi Zhang

5/6/2020

Reading Streaming Music: Approaches to Analysis in 2019 Pop Music

This report was made by Chi Zhang, UTEID: cz6753.

Overview

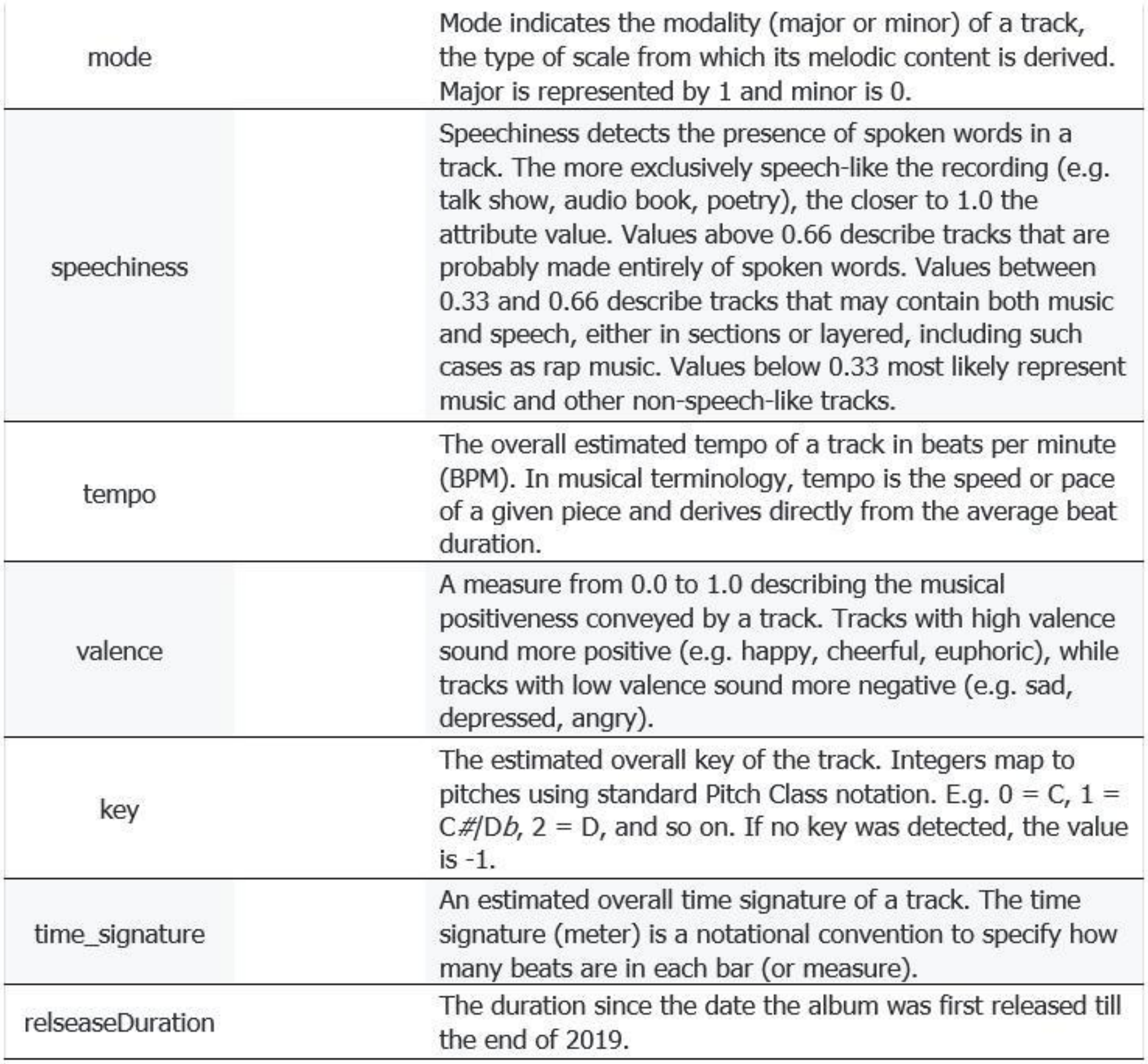
When you're stuck at home, music can provide a sense of order amid the coronavirus chaos. Streaming music is cheap or even free (in the case of Pandora and Spotify) and outpaces any physical format when it comes to ease and convenience. There is no denying that streaming music is now faced with the biggest development opportunity since streaming services accounted for nearly 80% of all music revenue in past 2019, in accordance to a year-end report from the RIAA. Therefore, it is of importance to understand user behavior and preference when interacting with streaming music services. To accomplish this objective, we collected the data of the 2019 popular music database from Pandora, one of the widely-used media service providers, and tried to build the best predictive model possible for streams of songs through both linear and non-linear methods for comprehensiveness. More than that, I also segmented the songs into five groups through unsupervised algorithms and estimated the popularity trend for various groups throughout the year. By doing so, the result of this project could help streaming music servers improve their playlist song recommendations so that better meet users' needs and provide the basis for the optimization of future development roadmap.

Data Sources

The Pandora Top Spins Chart is a record chart ranking the 100 tracks that have been streamed the most over the course of the past week. It is published weekly by Pandora. From https://www.nextbigsound.com/charts/top-spins, we downloaded the weekly data of the top 100 songs in the US. The data in year 2019 gives us access to 1,502 different songs. Given Pandora's public API, I had access to the data on the song features, artists and album information. By using web crawler, I could extract those target data from the web page and store them in .csv format. In the end, the formal dateset ultimately contains the following variables(attached with corresonding descriptions):

Table1: Variables Description

Variable	Description
id	song ID
duration	The duration (in seconds) of the track.
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
instrumentalness	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly vocal. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.



mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C*♯*/D*♭*, 2 = D, and so on. If no key was detected, the value is -1.
time_signature	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
relseaseDuration	The duration since the date the album was first released till the end of 2019.

Predictive Model Building

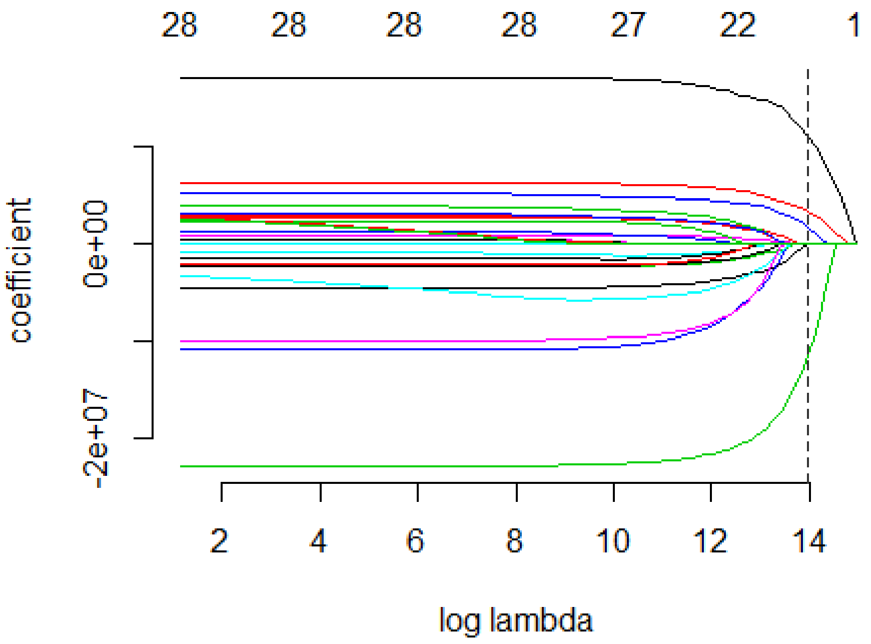
In this part I intended to build a predictive model for the streams of songs in 2019. I selected between linear regression and decision tree models, using methods such as stepwise selection, lasso regression and random forests to make sure the robustness of prediction results. For the first model, I started with the null model by regressing streams on one, followed by running stepwise selection within 25 song feature variables and eventually obtained the final model. For the second model, I began with the medium model by regressing streams on all other 25 variables, and used stepwise method to choose variables within all the 25 song features and their interactions. By doing so, the two selected models are shown below. Noted that I had 5 and 31 significant coefficients, respectively in the first and second model.

```
## [1] "model 1: "
## Streams ~ danceability + speechiness + explicitTRUE + key8

## [1] "model 2: "

## Streams ~ duration_ms_x + acousticness + danceability + energy +
##     liveness + loudness + mode + speechiness + valence + key6 +
##     key8 + key10 + explicitTRUE + relseaseDuration + explicitTRUE:relsease
Duration +
##     valence:explicitTRUE + duration_ms_x:key8 + energy:liveness +
##     acousticness:liveness + mode:speechiness + mode:key10 + speechiness:ex
plicitTRUE +
##     liveness:key6 + acousticness:energy + valence:key6 + mode:key6 +
##     danceability:key8 + key8:explicitTRUE + valence:key8 + speechiness:key
8
```

I then used the Lasso model to assemble the best predictive model possible for streams. When using this method to select two models above, I did not allow for the interaction terms in the third model, but included them in the forth model.

For the third model, viewed from the path plot below I could find that minimum AIC occurs at segment 8, where there are 6 coefficients in this model.

Figure1: Pathplot of Lasso (Model 3)

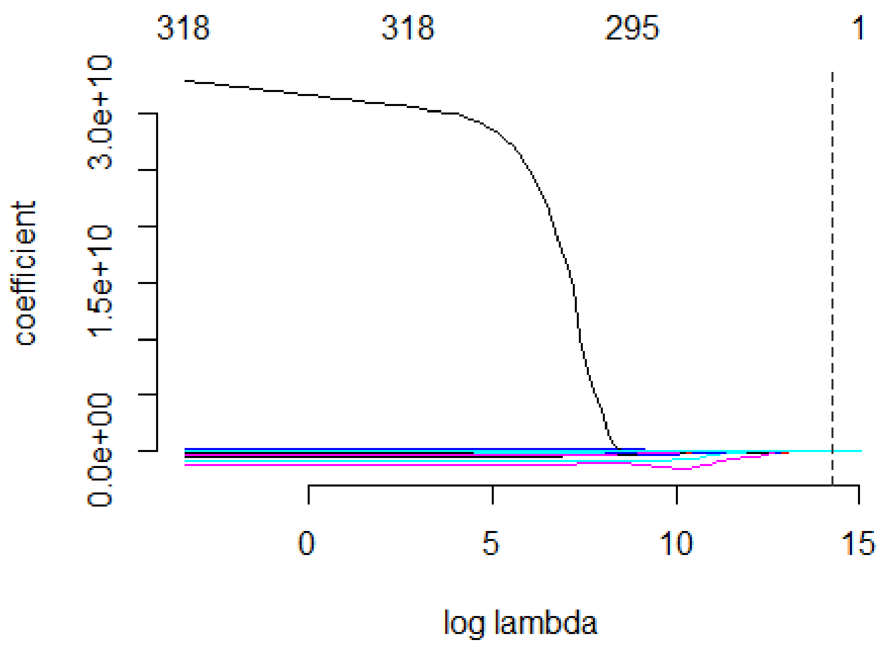


Thus, I used the model at the segment 8 and chose 6 coefficients. The specific model is shown below.

```
## [1] "model 3: "
## Streams ~ danceability + speechiness + key8 + explicitTRUE +
##     relseaseDuration
```

For the forth model, viewed from the path plot below I could see that minimum AIC occurs at segment 5, where there are 8 coefficients in the model.

Figure2: Pathplot of Lasso (Model 4)



Thus, I used the model at the segment 5 and chose 8 coefficients. The specific model is shown below.

```
## [1] "model 4: "

## Streams ~ danceability + danceability:time_signature4 + danceability:expli
citTRUE +
##     energy:speechiness + speechiness:relseaseDuration + key8:explicitTRUE
+
##     time_signature4:explicitTRUE
```

Afterwards, I used the decision tree models to assemble the best predictive model possible for streams. I tried the random forest model and the boosting model on the dataset, which gave me 2 non-linear models: the fifth model and the sixth model.

Table2: RMSE of Different Models

Model	CV
Model 1	34828786.1178673

Model 2	34648900.5959989
Model 3	34813629.5545773
Model 4	34820894.1844114
Model 5	34571289.8694528
Model 6	34925982.6704777

Lastly, I used k-fold cross validation in order to compare all 6 models above. I found that the CVs of the second model has the minimum CV, and therefore it is the best predictive model possible for streams. The advantage of a linear model is that a linear model with interactions is much easier to interpret than the non-linear models.

The second best model was the fifth model, which came from the random forest method. The random forest model has one advantage over the linear regression: it will only give us positive predictions. As a result, I used both the second model and the fifth model to do the predictions.

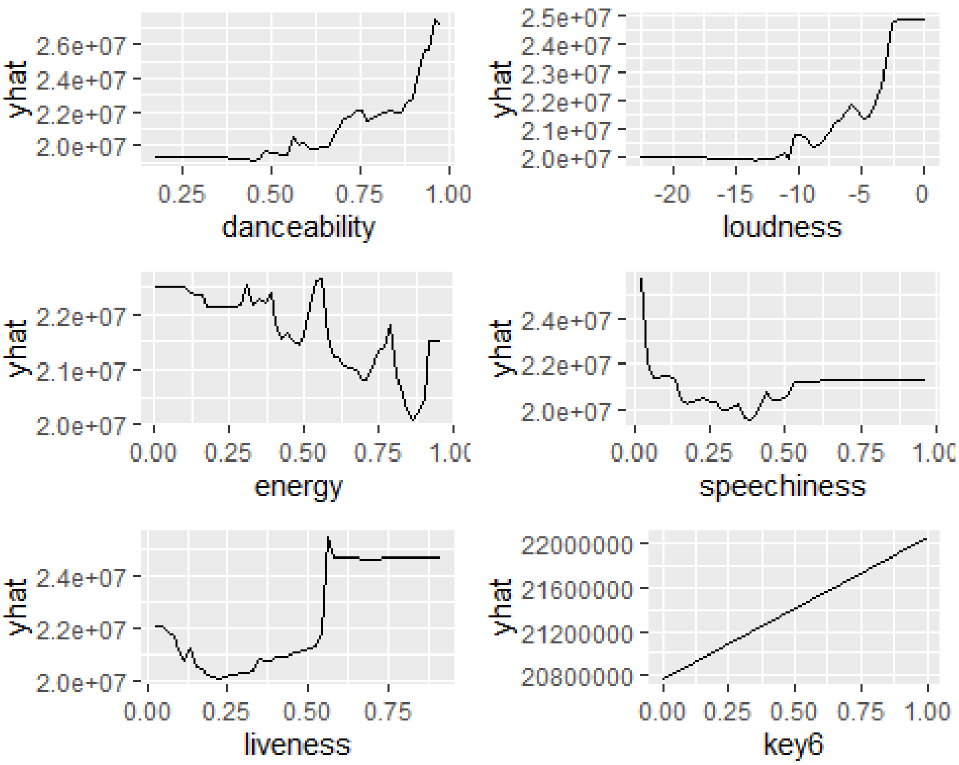
Table3: Coefficients of Model 2

	coefficients.Estimate	coefficients.Std..Error	coefficients.t.value	coefficients.Pr…t..
(Intercept)	1.100878e+07	1.469573e+07	0.7491141	0.4539222
duration_ms_x	-2.814570e+01	2.102728e+01	-1.3385325	0.1809539
acousticness	3.630936e+06	1.372552e+07	0.2645391	0.7914059
danceability	2.152083e+07	8.260827e+06	2.6051664	0.0092864
energy	7.257964e+06	1.443598e+07	0.5027690	0.6152108
liveness	1.390366e+08	3.865513e+07	3.5968465	0.0003340
loudness	1.028951e+06	5.962502e+05	1.7257029	0.0846354
mode	4.994066e+06	3.120765e+06	1.6002700	0.1097785
speechiness	2.173976e+07	1.807288e+07	1.2028940	0.2292335
valence	-1.725879e+07	7.844637e+06	-2.2000743	0.0279748
key6	7.753823e+05	1.109431e+07	0.0698901	0.9442917
key8	-3.361978e+07	2.474442e+07	-1.3586816	0.1744802
key10	-6.252876e+06	4.352800e+06	-1.4365179	0.1510925
explicitTRUE	-7.452120e+07	2.617778e+07	-2.8467348	0.0044855
relseaseDuration	2.460912e+01	3.417435e+02	0.0720105	0.9426045

explicitTRUE:relsease Duration	9.391623e+03	3.364228e+03	2.7916128	0.0053204
valence:explicitTRUE	2.590461e+07	9.317731e+06	2.7801418	0.0055109
duration_ms_x:key8	2.293418e+02	7.639888e+01	3.0019006	0.0027335
energy:liveness	-1.796170e+08	5.143930e+07	-3.4918241	0.0004956
acousticness:liveness	-1.302234e+08	3.808985e+07	-3.4188475	0.0006481
mode:speechiness	-2.548602e+07	1.465146e+07	-1.7394866	0.0821828
mode:key10	1.678909e+07	6.905583e+06	2.4312336	0.0151802
speechiness:explicitTRUE	-3.510896e+07	1.897601e+07	-1.8501760	0.0645120
liveness:key6	7.234027e+07	3.311980e+07	2.1842002	0.0291234
acousticness:energy	3.846649e+07	2.251290e+07	1.7086422	0.0877529
valence:key6	-3.197385e+07	1.893489e+07	-1.6886205	0.0915289
mode:key6	1.263361e+07	7.858295e+06	1.6076780	0.1081454
danceability:key8	-5.217979e+07	2.412374e+07	-2.1630061	0.0307202
key8:explicitTRUE	2.971791e+07	9.221327e+06	3.2227365	0.0013008
valence:key8	3.968255e+07	1.643568e+07	2.4144145	0.0158961
speechiness:key8	-4.936280e+07	2.701347e+07	-1.8273400	0.0678749

From the second model, I could clearly see that danceability, energy, liveness, loudness, mode, speechiness and key 6 have positive effects on streams, which means the more these factors used in the song, the more people the song will be played. Also, I intend to pay attention to release duration of the track . The longer the release duration is, the song will be played by less people, which means users prefer to play latest songs on.

Figure3: Partial Dependence Plot (Model 5)



Last but not the least, I plot the partial dependence for each variable contained in the fifth model, and the results seem similar to those derived from the second model, which guarantee the robustness of results. In conclusion, both selected linear model(the second model) and the decision tree model(the fifth model) provided me with similar results.

PCA and Clustering

General methodologies

In this section I would like to segment the 1,502 songs into groups with similar features in order to recommend to listeners who share the same interests/taste. For the reason of reducing unnecessary noises and computations, I first reduced the initial 25 variables by PCA. Next, I clustered them into groups with similar principle components, and based on the features in each principal component and the actual songs in each cluster, I were able to describe them in secular terminologies such as “genre”.

Part 1: PCA

In this part, I would like to use PCA to balance between the amount of computation load and explanatory variability, while eliminating as much noise as possible from the data. After demeaning and scaling of the data with standard deviation, I calculated the the loading matrix/scores matrix in order to derive the proportion of variance explained (PVE) and decide the number of principal components needed.

Table 4: PCA Components

ID	Standard deviation	Proportion of Variance	Cumulative Proportion
PC1	2.80212353537273	0.112084941414909	0.1121
PC2	1.77344288190906	0.0709377152763623	0.183
PC3	1.46727876624167	0.058691150649667	0.2417
PC4	1.44279628131763	0.0577118512527051	0.2994
PC5	1.2024934848242	0.0480997393929679	0.3475
PC6	1.18118422033003	0.0472473688132011	0.3948
PC7	1.16471426274806	0.0465885705099224	0.4414
PC8	1.11901808461181	0.0447607233844724	0.4861
PC9	1.10874235654547	0.0443496942618188	0.5305
PC10	1.08797663277924	0.0435190653111695	0.574
PC11	1.08178043639889	0.0432712174559556	0.6173
PC12	1.06412510961753	0.0425650043847012	0.6598
PC13	1.05315275540945	0.0421261102163779	0.702
PC14	1.03932598862547	0.0415730395450188	0.7435
PC15	1.00612785222318	0.0402451140889271	0.7838
PC16	0.947189585246249	0.03788758340985	0.8217
PC17	0.892958210859255	0.0357183284343702	0.8574
PC18	0.821043647001771	0.0328417458800708	0.8902
PC19	0.756342698063931	0.0302537079225572	0.9205
PC20	0.647404260335455	0.0258961704134182	0.9464
PC21	0.604914885182242	0.0241965954072897	0.9706
PC22	0.419178980270296	0.0167671592108118	0.9873
PC23	0.209254918185628	0.00837019672742513	0.9957
PC24	0.100442731001129	0.00401770924004517	0.9997
PC25	0.00698743489964209	0.000279497395985684	1

Table 4 reports that the first 20 principle components explain more than 90% of the variability. and hence I believe that these 20 principle components would keep the computation load low and eliminate some of the noises, while keeping the majority of the variability. Clustering would further group the songs based on these 20 principle components.

Part 2: Clustering

K-means++ clustering was used to determine the market segments. 3 types of supporting analysis were used to help me determine the number of K (centroids): Elbow plot(SSE), CH index and Gap statistics.

Figure 4: SSE Grid vs K

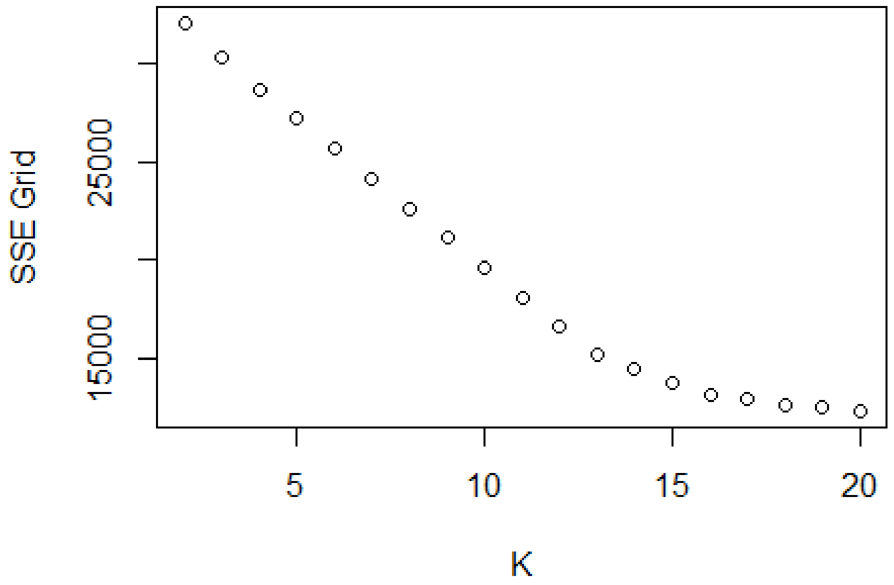


Figure5: CH Grid vs K

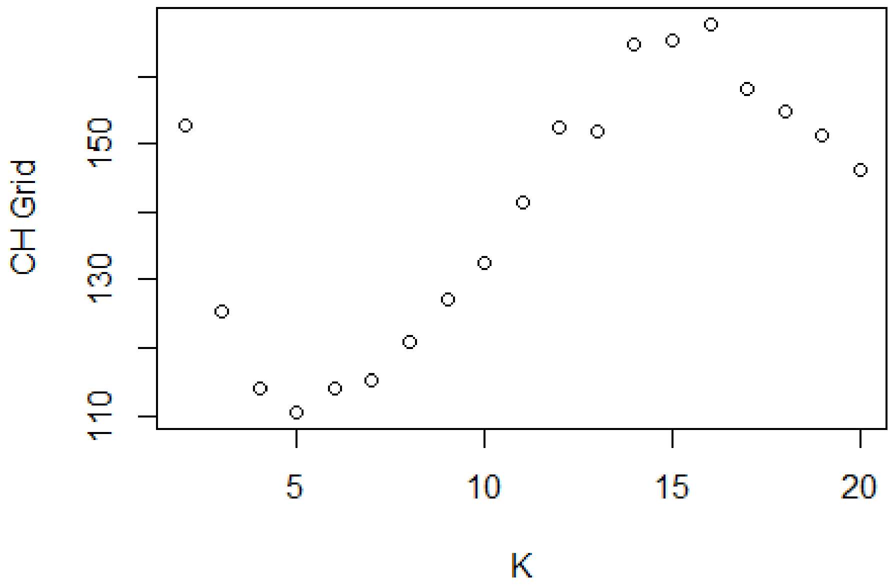
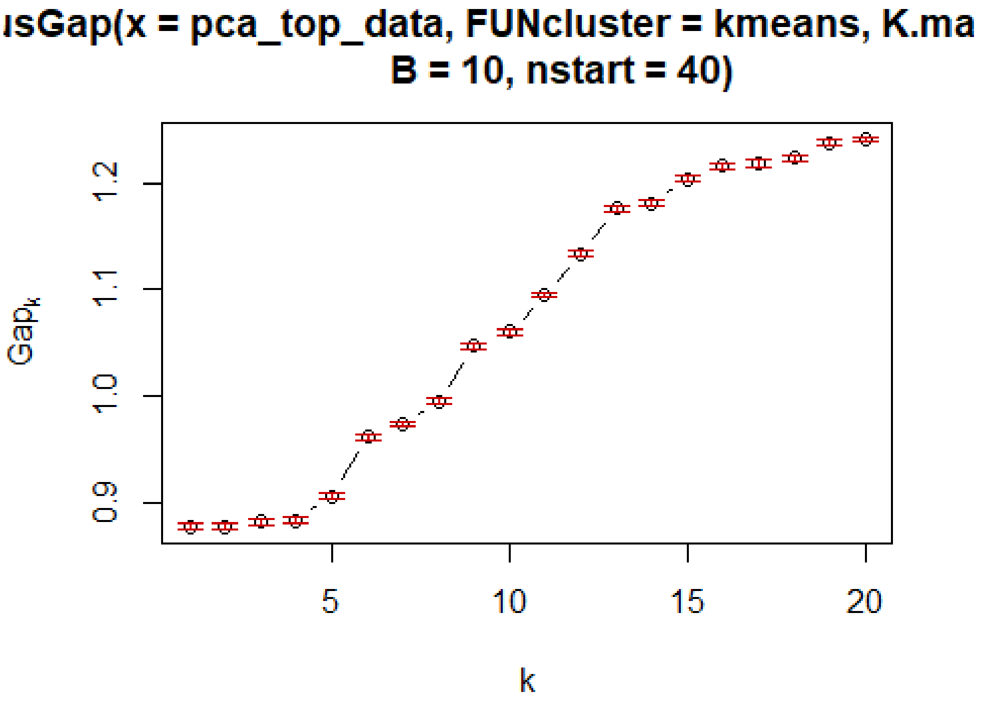


Figure6: Gap vs K



As shown above, both Elbow plot and CH index returned K=16 and Gap statistics returned K=4. Clustering 16 segments would not show the distinct differences among them as I now only have 20 principle components to allocate. So I selected K=4 as my anchor and explored the nearby Ks to see which one provides me with the best explanation for each cluster. For best explanation, I considered the following 2 categories.

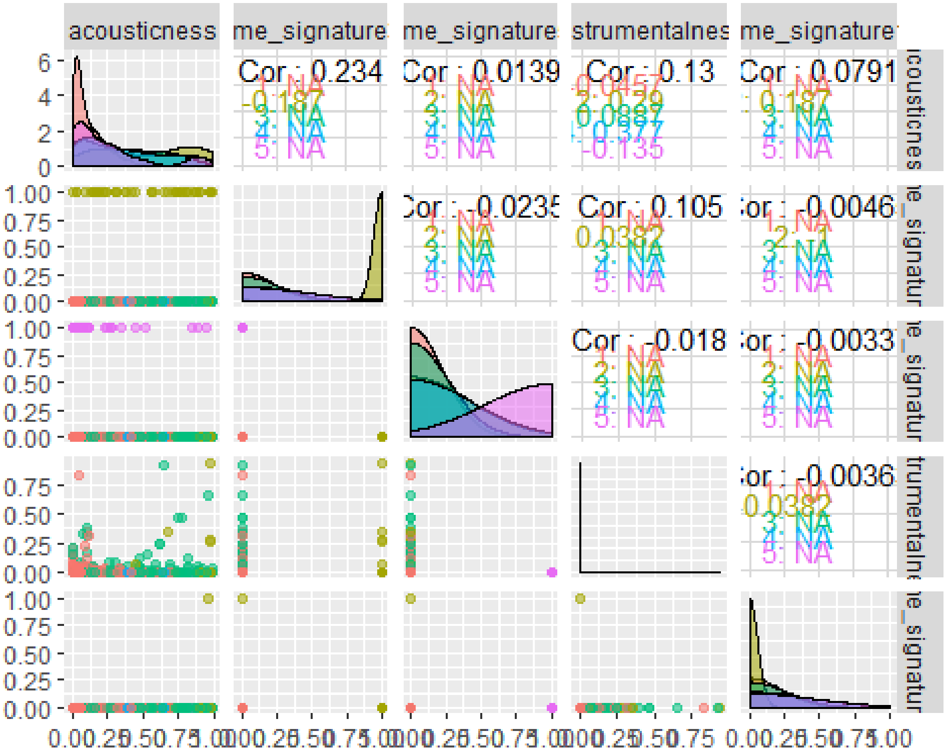
- Clusters that have songs with clear and unique distribution in any of the 20 features.
- Clusters that have songs with clear genre by their artist name and actual music.(I have played a considerable quantity of sample size from each cluster on video music providers such as YouTube, for confirmation)

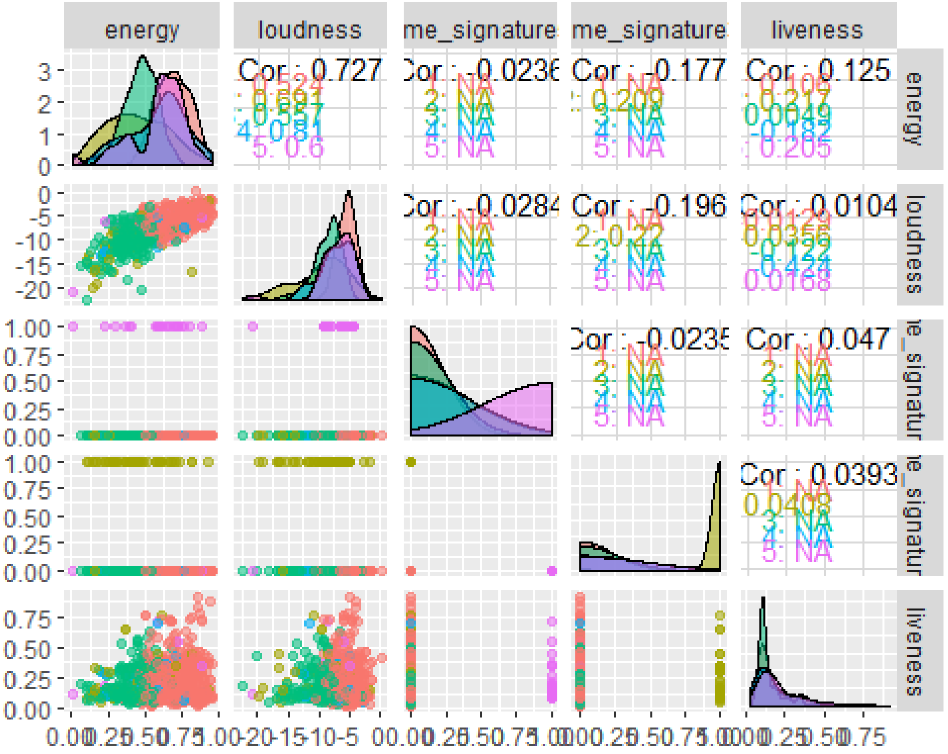
As the result, I eventually picked K = 5.

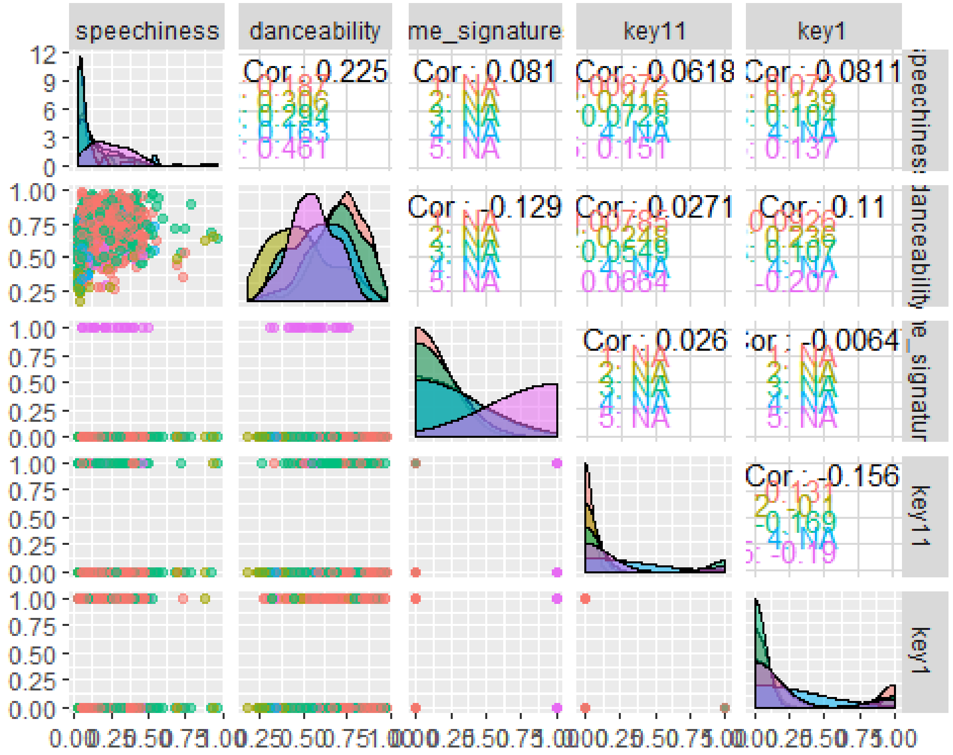
Catrgory 1: Song market segments breakdown by distribution of features

After 5 clusters were determined, first I reversed the principle components into the original features to determine cluster characteristics. Then I showed some of the cluster identifiable distributions and the summary of each cluster below.

Figure 7: cluster identifiable distributions


acousticness
me_signature
me_signature
strumentalnes
me_signature
Cor : 0.234
Cor : 0.0139
Cor : 0.13
Cor : 0.0791
Cor : -0.0235
Cor : 0.105
Cor : -0.0046
Cor : -0.018
Cor : -0.0033
Cor : -0.0036
1: NA
2: -0.187
3: NA
4: NA
5: NA
0.0457
2: 0.29
0.0887
-0.377
-0.135
0.0382
0.0382
1.00
0.75
0.50
0.25
0.00
0 2 4 6
0.00 0.25 0.50 0.75 1.00
coustícnes
ie_signatur
trumentalne


energy
loudness
me_signature
me_signature
liveness
Cor : 0.727
Cor : -0.0236
Cor : -0.177
Cor : 0.125
Cor : -0.0284
Cor : -0.196
Cor : 0.0104
Cor : -0.0235
Cor : 0.047
Cor : 0.0393
1: 0.524
2: 0.691
3: 0.557
4: 0.81
5: 0.6
2: 0.209
2: 0.22
0.106
0.217
0.0049
-0.182
0.205
0.0129
0.0356
-0.122
-0.424
0.0168
0.0408
1: NA
2: NA
3: NA
4: NA
5: NA
0 1 2 3
0 -5 -10 -15 -20
1.00 0.75 0.50 0.25 0.00
0.00 0.25 0.50 0.75 1.00
-20 -15 -10 -5 0
0.00 0.25 0.50 0.75
energy
loudness
ie_signatur
liveness



- Cluster 1: High in energy, high in loudness, high danceability, low speechiness, considerate amount of G key, low acousticness

- Cluster 2: Many 5 quarter time signature songs, high in energy

- Cluster 3: Many songs with high energy, high on loudness

- Cluster 4: Many songs with high on loudness, high danceability, considerable amount of B flat key

- Cluster 5: Many 3 quarter time signature songs, low speechiness

Category 2: Song market segments breakdown by genre

Since I have the full list of song names and artist names available in each cluster, I could actually listen to the songs and categorize them manually by the music genre standard as in pop, rock, rap, etc. If my cluster characteristics determined by K-means++ show close resemblance of the music genre, then the recommendation system could be effective, at least to the extent of traditional music listeners with distinct preference over specific genre.

- Cluster 1: Many songs with electronically altered/amplified sounds, very rhythmic, but genre varying from pop to rap to country, etc. Typical examples would be Suge (Yea Yea) by DaBaby, Caro by Bad Bunny and Spy Kid by Chief Keef & Zaytoven.

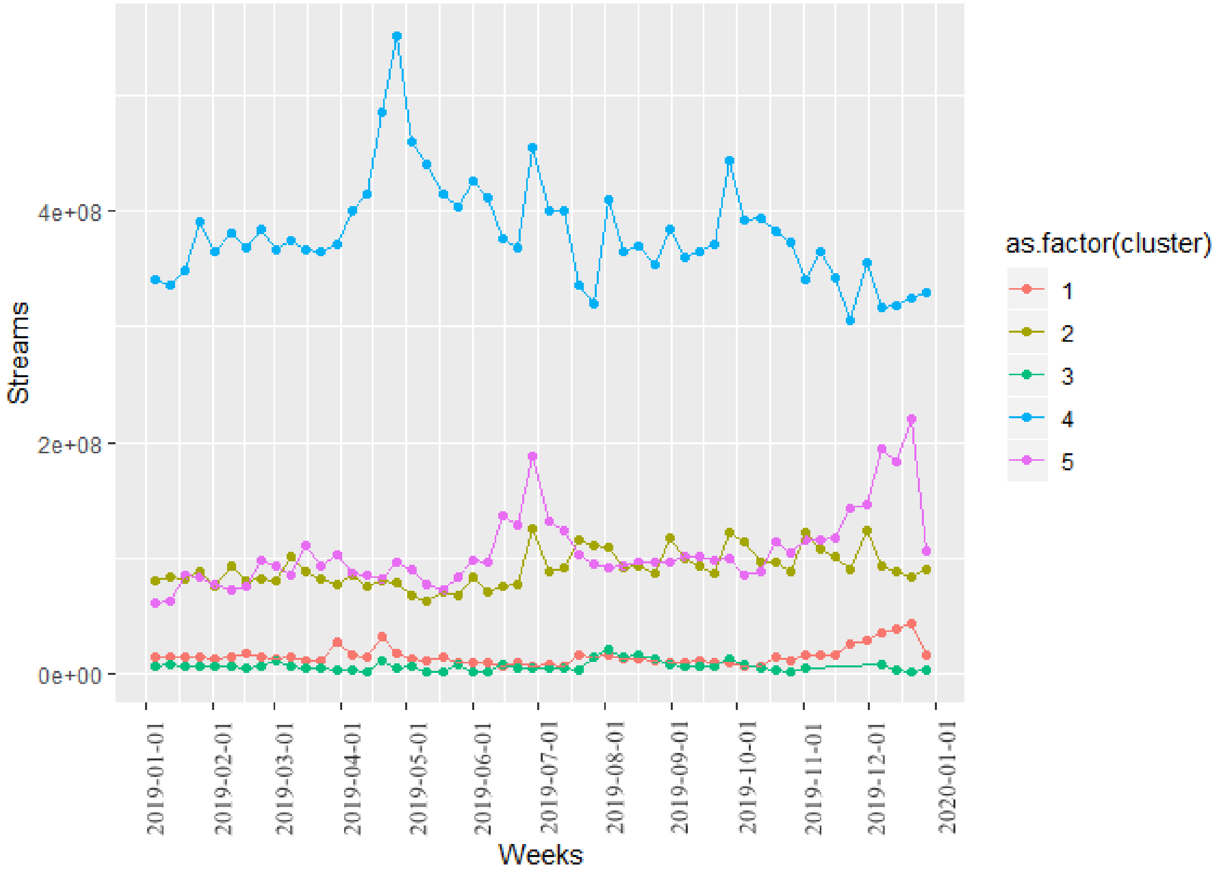
- Cluster 2: Indeed many songs with 5/4 time signature, high energy and rhythmic, but clearly sets apart different vibe compared cluster 1, perhaps due to the different time signature. Typical examples would be Higher by DJ Khaled, Safety, and That's Mine by GASHI.

- Cluster 3: Genre varies a lot in this cluster, as shown in the very different artists such as Drake, Kendrick Lamar, Taylor Swift, XXXTENTACION and Queen. I did realize that out of the many rap songs in this cluster, most of them were the slower ones. For example, Me! by Taylor Swift and Who Needs Love by Trippie Redd.
- Cluster 4: Songs in B flat key stands out, such as Midnight In Prague by Lil Xan and Say Something by Justin Timberlake, which make this cluster a different vibe than others.
- Cluster 5: Many indie and pop songs with long vowel sounds, typical examples would be When We Were Young by Hollow Coves, Hay-on-Wye by Matthew Frederick and I Keep on Telling Myself by Dve Thomas Junior.

Trend in Popularity

I also calculated the total streams of different song clusters by time. The following graph demonstrates the trend in the total streams of different categories.

Figure8: trend in the Total Streams



From this graph it is demonstrated that the stream of five types of songs does not change too much in a year. Cluster 4 music has more streams overall, due to the fact that there are more songs in this categories. There is a peak in the end of April in 2019 for cluster 4, and then the streams goes back to normal. From this graph I can also see that at the end of the year cluster 4 music is not as popular as in the middle of the year, but type 5 music becomes more and more popular, especially in June and the end of the year. The popularity of cluster 1, cluster 2 and cluster 3 music doesn't vary too much throughout the whole year.

Conclusion

In each age, the popularity of songs reflects people's preference over different music, which may also differ from each era. To predict the success of a song, taking the contemporaneous music preference into account is of significance. In streaming era, users are insane about music with elements of danceability, energy, liveness and so on. It seems that they are more likely to pursue the latest music. As a result, in order to predict the song's popularity trend in 2020, the first thing we need to do is to gather the information of users' music preference next year.

Traditional music listeners explore songs by specific genre and artists. This confirmation bias, typically nurtured through years of artificial genre segmentation by media and artist reputation, could limit listeners from the songs that they really want to be exposed to. The question of "why are we attracted to certain songs" is a philosophical discussion that is beyond the scope of our project here, but given the data from Pandora and the clustering method, I perhaps to show that key, time signature and speed of the songs are some of the contributing factors to the inner biological working of what to like and dislike. Then, my basic recommendation system, most likely already used by streaming music service providers like Pandora, Spotify, Apple Music, etc., could recommend songs not by mere genre and artist names, but also by specific keys and time signatures each listener is attracted to, subconsciously.