1.4 Assignment

Benjamin Melitz, Daniel Jang, Jimin Hong, Tanush Deka

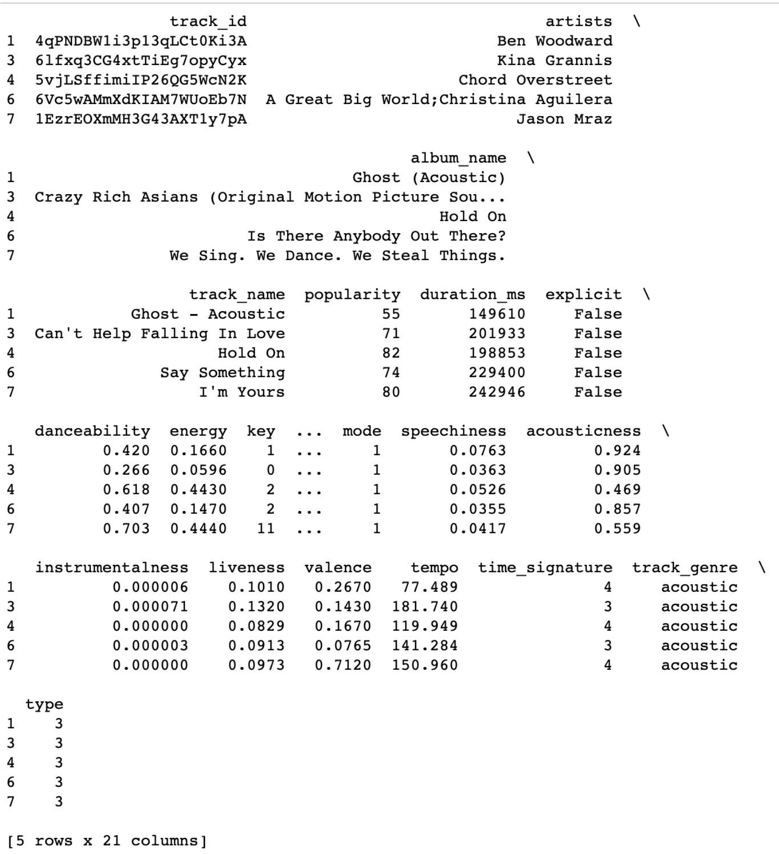
Original Model

The original model used the kmeans library to make 5 clusters of spotify songs. The user inputs their favorite songs by id, and then the model finds songs in the same group to suggest to the user.

Output of the model after song ids are input:

Pros: Used unsupervised learning to make clusters

Cons: Only 5 clusters, Unformatted Output



```
                 track_id                              artists  \
1  4qPNDBW1i3p13qLCt0Ki3A                         Ben Woodward   
3  6lfxq3CG4xtTiEg7opyCyx                         Kina Grannis   
4  5vjLSffimiIP26QG5WcN2K                     Chord Overstreet   
6  6Vc5wAMmXdKIAM7WUoEb7N  A Great Big World;Christina Aguilera   
7  1EzrEOXmMH3G43AXT1y7pA                           Jason Mraz   

                                          album_name  \
1                                   Ghost (Acoustic)   
3  Crazy Rich Asians (Original Motion Picture Sou...   
4                                            Hold On   
6                        Is There Anybody Out There?   
7                We Sing. We Dance. We Steal Things.   

                   track_name  popularity  duration_ms  explicit  \
1            Ghost - Acoustic          55       149610     False   
3  Can't Help Falling In Love          71       201933     False   
4                     Hold On          82       198853     False   
6               Say Something          74       229400     False   
7                   I'm Yours          80       242946     False   

   danceability  energy  key  ...  mode  speechiness  acousticness  \
1         0.420  0.1660    1  ...     1       0.0763         0.924   
3         0.266  0.0596    0  ...     1       0.0363         0.905   
4         0.618  0.4430    2  ...     1       0.0526         0.469   
6         0.407  0.1470    2  ...     1       0.0355         0.857   
7         0.703  0.4440   11  ...     1       0.0417         0.559   

   instrumentalness  liveness  valence    tempo  time_signature  track_genre  \
1          0.000006    0.1010   0.2670   77.489               4     acoustic   
3          0.000071    0.1320   0.1430  181.740               3     acoustic   
4          0.000000    0.0829   0.1670  119.949               4     acoustic   
6          0.000003    0.0913   0.0765  141.284               3     acoustic   
7          0.000000    0.0973   0.7120  150.960               4     acoustic   

  type
1   3
3   3
4   3
6   3
7   3

[5 rows x 21 columns]
```

Model Enhancements: Outlier Removal

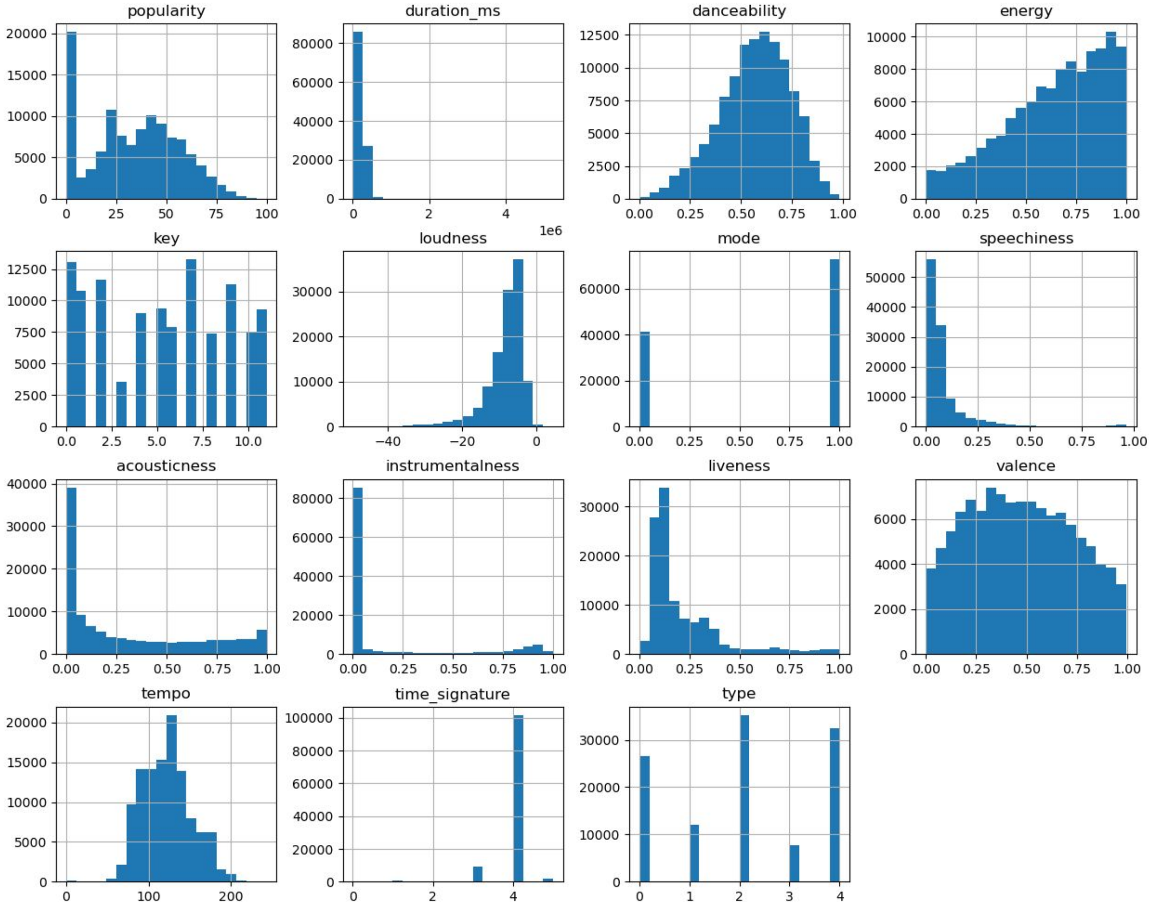
The first enhancement was removing outliers from the dataset. This allowed for better representation of the data, and more practical clusters of songs. Using the scipy library, we calculated the z scores of the ‘tracks’ dataframe using the scipy.stats.zscore function. Outliers were any entries where at least one column had a z score of more than 4 or less than -4 (~1.26% of entries). Without outliers, the model does not have to make small groups for a couple outlier songs and can be focus on the majority of songs

Model Enhancements: Other Models

The next enhancement was using other models, like DBSCAN and agglomerative clustering. The agglomerative clustering model took too long on the large dataset, so we used the Birch algorithm in sklearn.cluster instead.

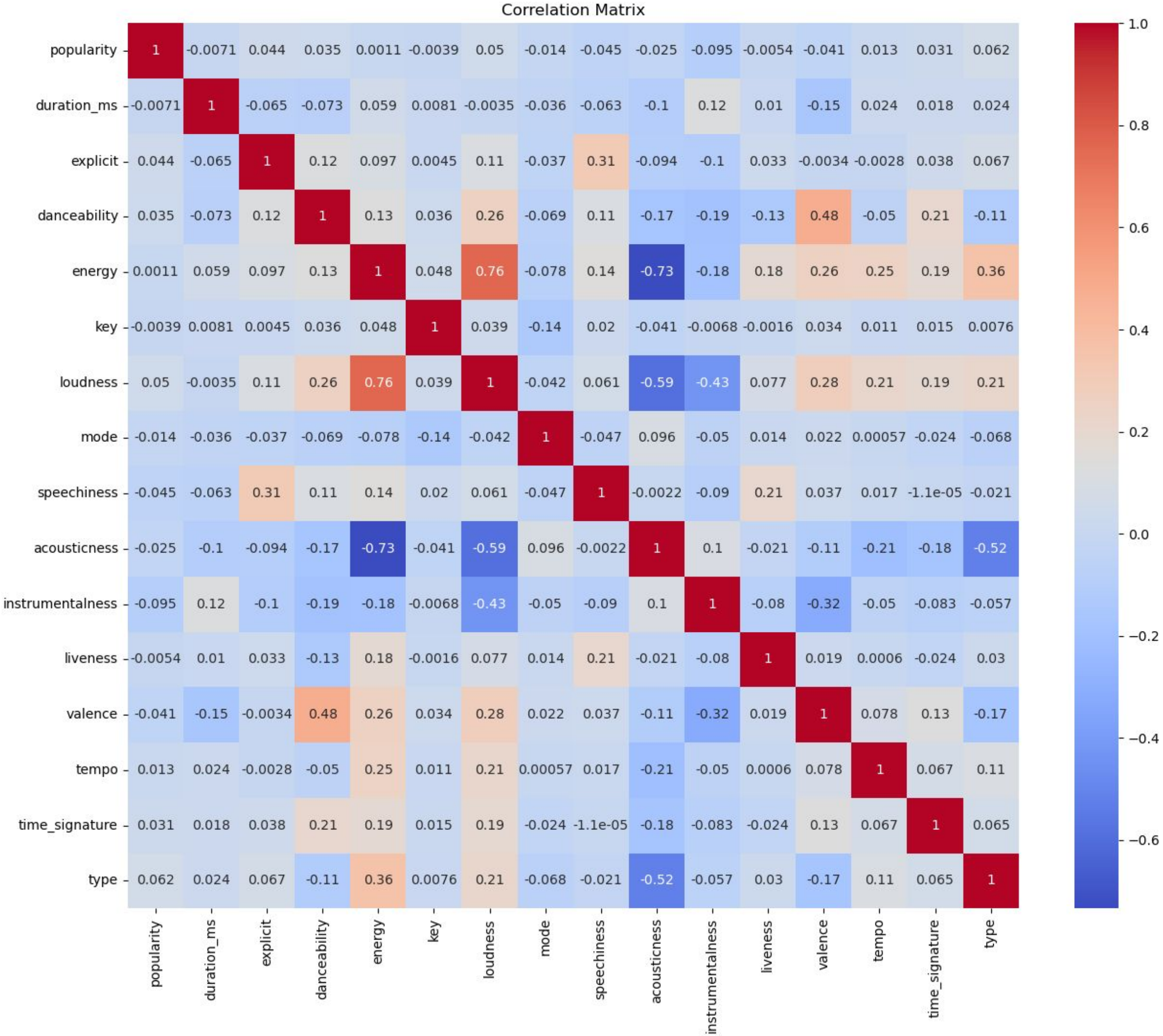
Feature Distributions

Plots of the columns from the spotify dataset



Correlation Matrix

Correlation matrix of the columns from the spotify dataset



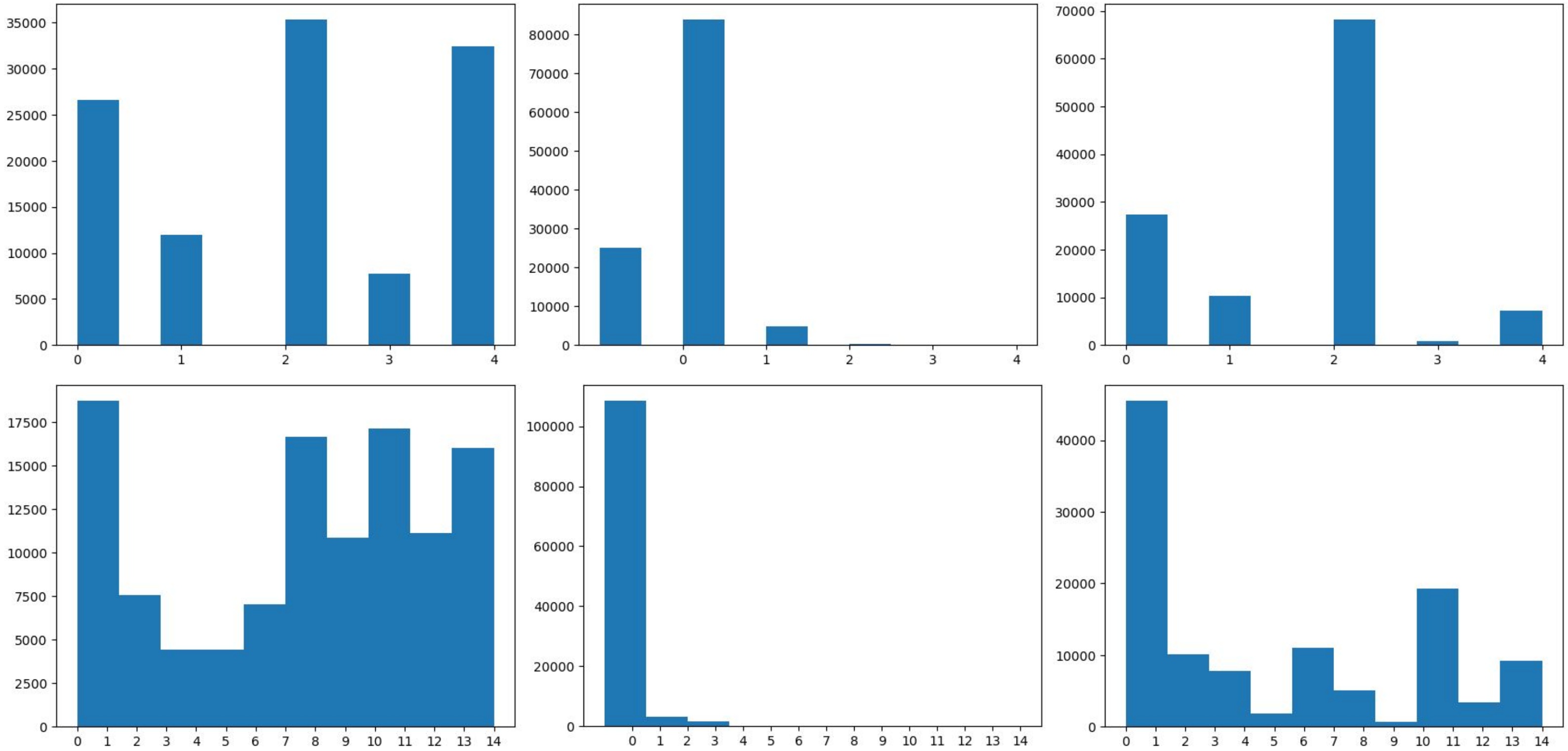
	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	type
popularity	1	-0.0071	0.044	0.035	0.0011	-0.0039	0.05	-0.014	-0.045	-0.025	-0.095	-0.0054	-0.041	0.013	0.031	0.062
duration_ms	-0.0071	1	-0.065	-0.073	0.059	0.0081	-0.0035	-0.036	-0.063	-0.1	0.12	0.01	-0.15	0.024	0.018	0.024
explicit	0.044	-0.065	1	0.12	0.097	0.0045	0.11	-0.037	0.31	-0.094	-0.1	0.033	-0.0034	-0.0028	0.038	0.067
danceability	0.035	-0.073	0.12	1	0.13	0.036	0.26	-0.069	0.11	-0.17	-0.19	-0.13	0.48	-0.05	0.21	-0.11
energy	0.0011	0.059	0.097	0.13	1	0.048	0.76	-0.078	0.14	-0.73	-0.18	0.18	0.26	0.25	0.19	0.36
key	-0.0039	0.0081	0.0045	0.036	0.048	1	0.039	-0.14	0.02	-0.041	-0.0068	-0.0016	0.034	0.011	0.015	0.0076
loudness	0.05	-0.0035	0.11	0.26	0.76	0.039	1	-0.042	0.061	-0.59	-0.43	0.077	0.28	0.21	0.19	0.21
mode	-0.014	-0.036	-0.037	-0.069	-0.078	-0.14	-0.042	1	-0.047	0.096	-0.05	0.014	0.022	0.00057	-0.024	-0.068
speechiness	-0.045	-0.063	0.31	0.11	0.14	0.02	0.061	-0.047	1	-0.0022	-0.09	0.21	0.037	0.017	-1.1e-05	-0.021
acousticness	-0.025	-0.1	-0.094	-0.17	-0.73	-0.041	-0.59	0.096	-0.0022	1	0.1	-0.021	-0.11	-0.21	-0.18	-0.52
instrumentalness	-0.095	0.12	-0.1	-0.19	-0.18	-0.0068	-0.43	-0.05	-0.09	0.1	1	-0.08	-0.32	-0.05	-0.083	-0.057
liveness	-0.0054	0.01	0.033	-0.13	0.18	-0.0016	0.077	0.014	0.21	-0.021	-0.08	1	0.019	0.0006	-0.024	0.03
valence	-0.041	-0.15	-0.0034	0.48	0.26	0.034	0.28	0.022	0.037	-0.11	-0.32	0.019	1	0.078	0.13	-0.17
tempo	0.013	0.024	-0.0028	-0.05	0.25	0.011	0.21	0.00057	0.017	-0.21	-0.05	0.0006	0.078	1	0.067	0.11
time_signature	0.031	0.018	0.038	0.21	0.19	0.015	0.19	-0.024	-1.1e-05	-0.18	-0.083	-0.024	0.13	0.067	1	0.065
type	0.062	0.024	0.067	-0.11	0.36	0.0076	0.21	-0.068	-0.021	-0.52	-0.057	0.03	-0.17	0.11	0.065	1

Model Cluster Comparison

KMeans DBSCAN Birch

5 Clusters

15 Clusters



Number of Cluster Comparisons

We tried the silhouette method, but it took too long. Here are results from the elbow test with kmeans. The elbow test takes the number of clusters where the distortion point levels off. Distortion point measures distance from data points to center of cluster. We chose 15 clusters

5 clusters with value of 423432.3, 10 clusters with value of 267103.8

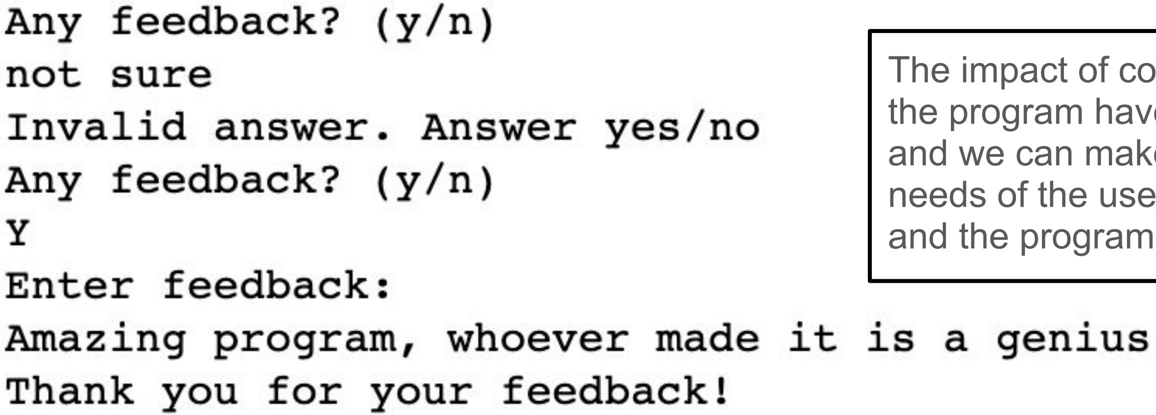
15 clusters with value of 221535.1, 20 clusters with value of 195752.4

25 clusters with value of 177833.0, 30 clusters with value of 165197.3

35 clusters with value of 154795.6, 40 clusters with value of 146217.4

User Feedback

To collect feedback, first we ask the user if they want to submit feedback. If the user enters one of the following answers: y, yes, n, no (in any case) then it accepts the answer, otherwise it asks again. If the user enters y or yes, the program asks for feedback and adds the feedback to a ‘feedback.txt’ file. If no ‘feedback.txt’ file exists, one is created. Example interaction:



```
Any feedback? (y/n)
not sure
Invalid answer. Answer yes/no
Any feedback? (y/n)
Y
Enter feedback:
Amazing program, whoever made it is a genius
Thank you for your feedback!
```

The impact of collecting feedback means that the users of the program have a voice on how to fix/update the program, and we can make changes to the program to meet the needs of the users. It provides interaction between the user and the programmer/s

Future Enhancements

The program could take a playlist link and recommend songs from the link

The program could give a nice interface or GUI for the user

The program could choose genres of music to include/exclude in the recommendations

The program could ask the user if the recommendations were good or not, and learn from the feedback to include or exclude songs like the ones it recommended