

Spotify Song Recommendation

Spotify

Ben Sizelove

Problem Statement

In this project I plan to try and recommend to a user a list of potential songs they will like based off of a songs musical attributes. This dataset contains about 230,000 songs from 27 different genres. I found this data set from Kaggle.com however the original data was pulled from Spotify's api open to developers

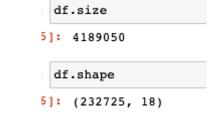
Data Characteristics

The following data attributes were used in the analysis presented here. Understanding the types of data points helped determine what cleaning was needed.

Variable Name Description object artist_name object track_name object track_id object int64 popularity acousticness float64 danceability float64 duration_ms int64 float64 energy float64 instrumentalness object key float64 liveness float64 loudness mode object float64 speechiness float64 object time signature float64 valence

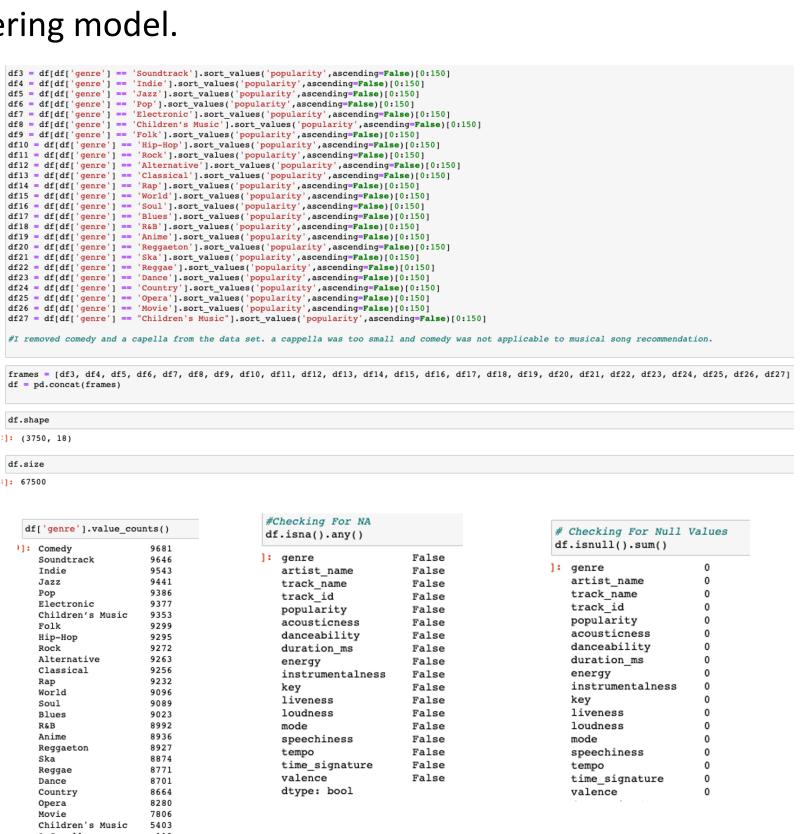
Initial Data

enre	artist_name	track_name	track_id	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valen
lovie	Henri Salvador	C'est beau de faire un Show	0BRjO6ga9RKCKjfDqeFgWV	0	0.611	0.389	99373	0.910	0.000	C#	0.3460	-1.828	Major	0.0525	166.969	4/4	0.8
lovie	Martin & les fées	Perdu d'avance (par Gad Elmaleh)	0BjC1NfoEOOusryehmNudP	1	0.246	0.590	137373	0.737	0.000	F#	0.1510	-5.559	Minor	0.0868	174.003	4/4	0.8
lovie	Joseph Williams	Don't Let Me Be Lonely Tonight	0CoSDzoNIKCRs124s9uTVy	3	0.952	0.663	170267	0.131	0.000	С	0.1030	-13.879	Minor	0.0362	99.488	5/4	0.3
lovie	Henri Salvador	Dis-moi Monsieur Gordon Cooper	0Gc6TVm52BwZD07Ki6tlvf	0	0.703	0.240	152427	0.326	0.000	C#	0.0985	-12.178	Major	0.0395	171.758	4/4	0.2
lovie	Fabien Nataf	Ouverture	0luslXpMROHdEPvSl1fTQK	4	0.950	0.331	82625	0.225	0.123	F	0.2020	-21.150	Major	0.0456	140.576	4/4	0.3

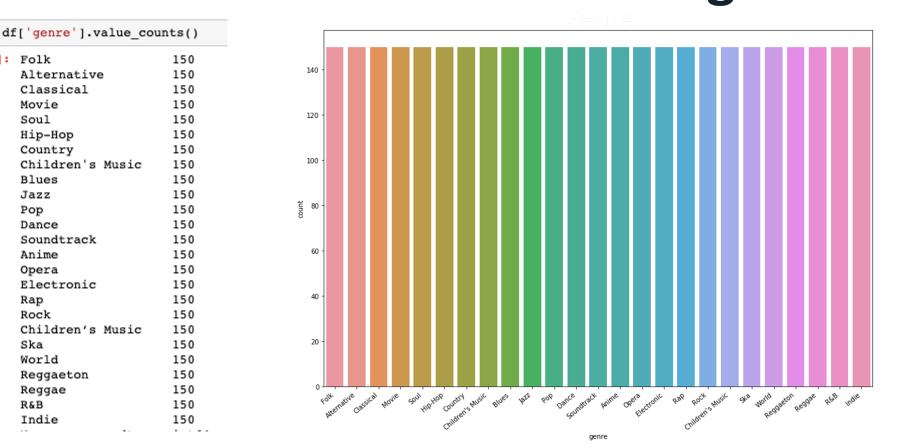


Data Cleaning

The majority of my cleaning at this stage was subsetting the data so that there was an equal amount of songs from each genre. This was important for when I would eventually run a clustering model.



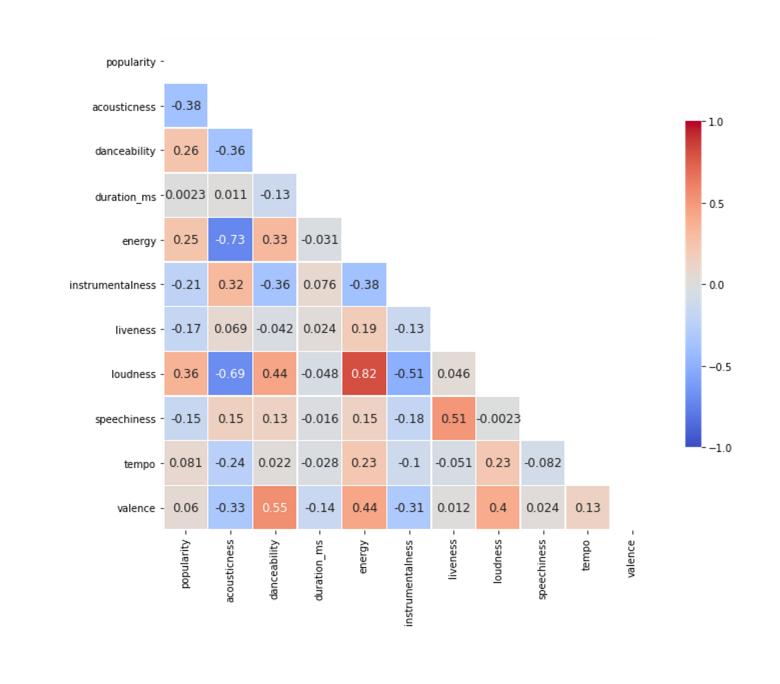
Data After Cleaning



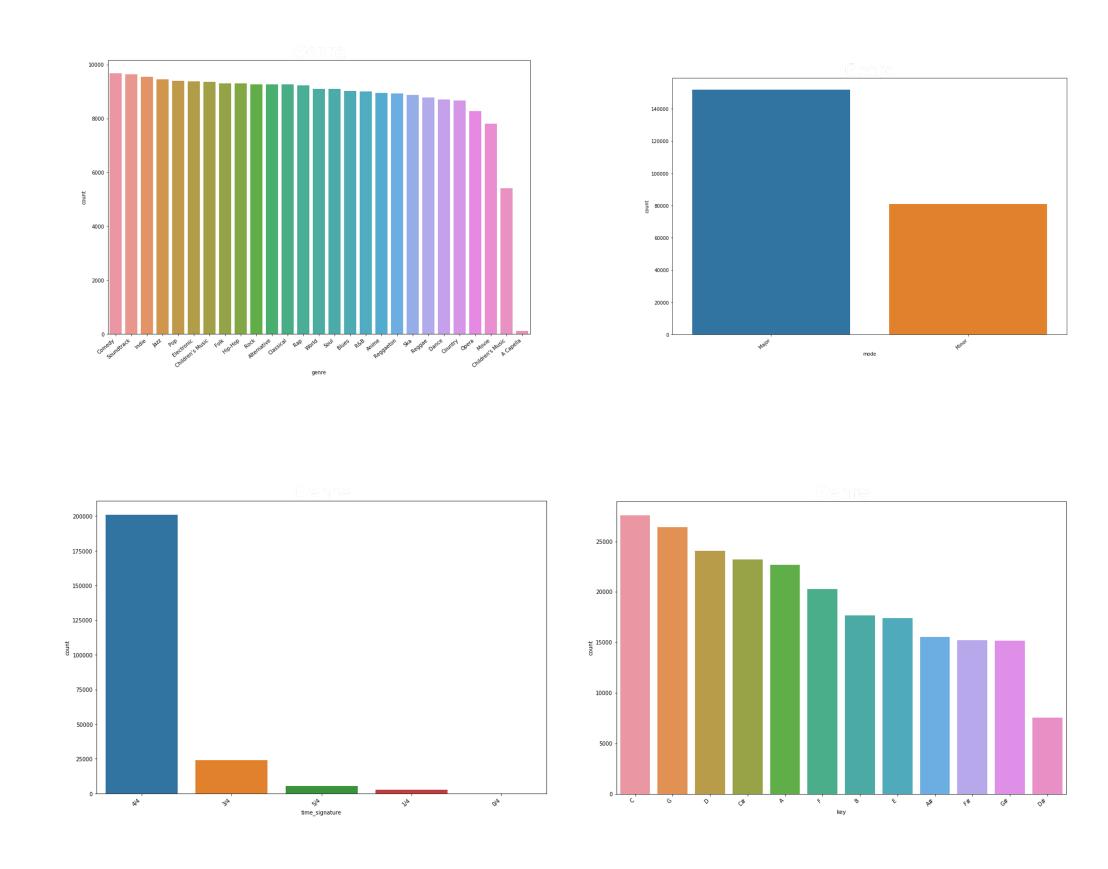
Data Exploration

df.	df.describe()														
:			-												
		popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence	cluster_		
	count	3750.000000	3750.000000	3750.000000	3.750000e+03	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000	3750.000000		
	mean	68.732000	0.347904	0.575938	2.281282e+05	0.563491	0.113447	0.177132	-9.071927	0.079439	117.793168	0.467727	3.945600		
	std	13.716771	0.344434	0.191047	8.027557e+04	0.256471	0.279143	0.150834	6.201948	0.078269	30.891447	0.259823	3.011444		
	min	27.000000	0.000010	0.060900	3.155600e+04	0.001440	0.000000	0.017600	-41.808000	0.022400	35.995000	0.022400	0.000000		
	25%	62.000000	0.039325	0.452000	1.853402e+05	0.391000	0.000000	0.092400	-10.599500	0.036100	93.989000	0.255000	2.000000		
	50%	71.000000	0.207500	0.596000	2.173070e+05	0.602000	0.00008	0.116500	-7.119000	0.047300	115.074000	0.460500	3.000000		
	75%	78.000000	0.648750	0.720000	2.586192e+05	0.765000	0.003520	0.204000	-5.213500	0.086750	138.137500	0.671000	5.000000		
	max	100.000000	0.996000	0.964000	1.240600e+06	0.995000	0.992000	0.986000	1.023000	0.779000	218.184000	0.985000	10.000000		

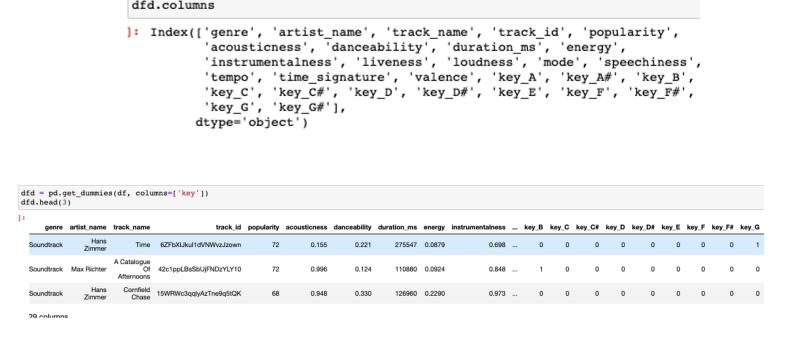
Correlation Detection



Understanding categorical attributes



Testing Categorical Variables as Attributes for Model



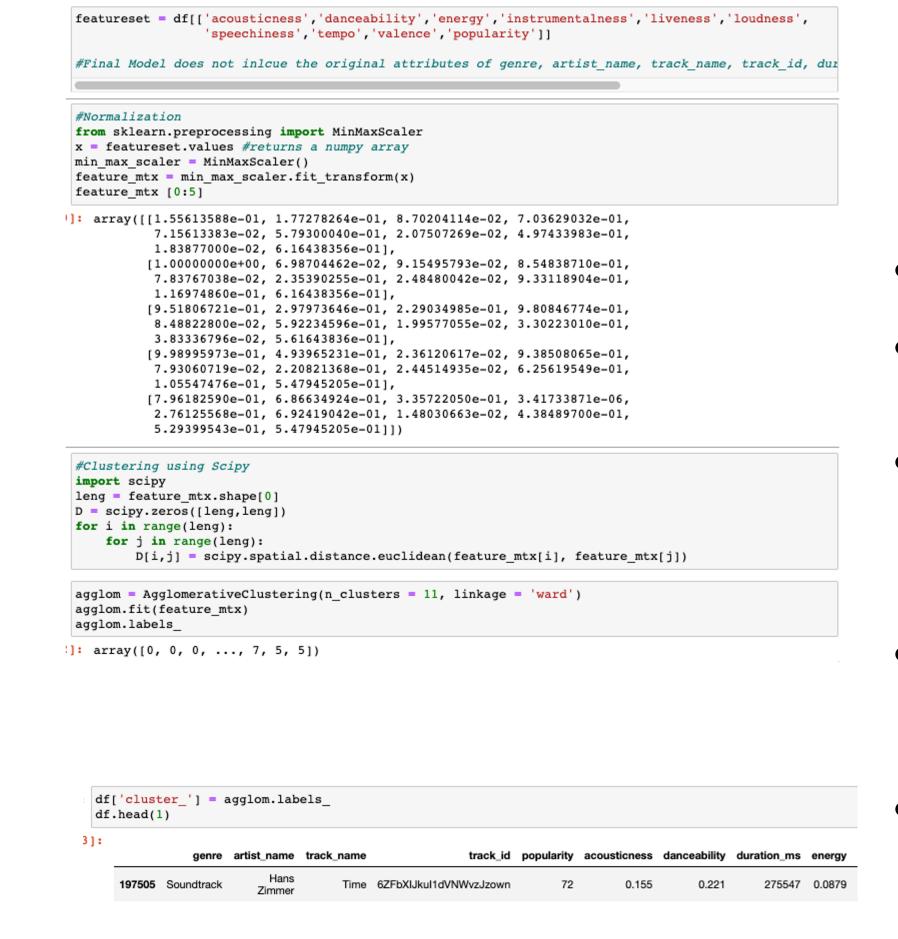
After exploring these variables I saw that they likely wouldn't be the best for clustering similar songs because these categorical attributed didn't differentiate between songs significantly.

MODEL SELECTION

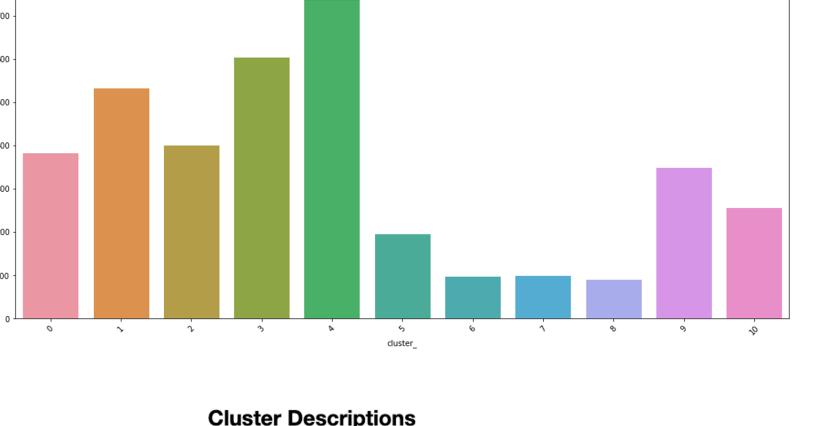
When selecting the model, I decided to use a agglomerative clustering model. The data explorations demonstrated there were attributes of songs that would lend themselves towards clustering.

Additionally, the majority of the predictor variables after cleaning were continuous which also leans to clustering. Also, for the model I had to reduce the size of the input data because this type of clustering has performance limitations for larger data sets

AHC Clustering



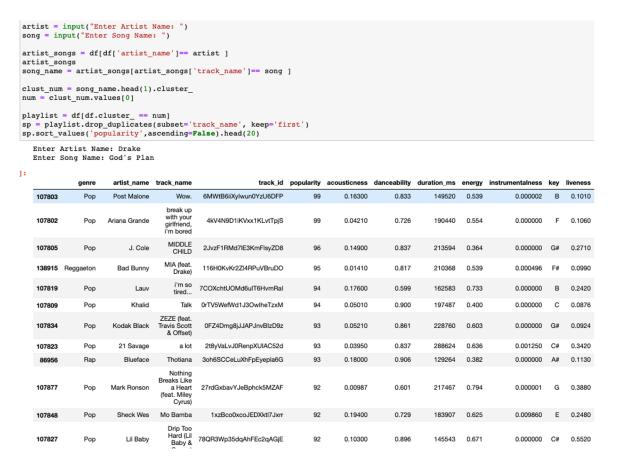
Results



How I describe the different clusters -Cluster 0: Instrumentals -Cluster 1: Slower and more relaxed pop songs ~ the feels -Cluster 2: Softer rock style songs -Cluster 3: Relaxed and more upbeat pop songs -Cluster 4: Upbeat and rythmic latin influenced songs -Cluster 5: Soft acoustics -Cluster 6: Electronic/Rock -Cluster 7: Anime/Childrens songs -Cluster 8: Faster pace non popular songs -Cluster 9: Popular Rap songs -Cluster 10: More agressive and serious pop songs

	popularity	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
luster_											
0	57.130548	0.882587	0.298614	251413.736292	0.150699	0.856796	0.136349	-21.835191	0.044502	101.153441	0.160004
1	67.073171	0.715047	0.543921	218411.911820	0.426391	0.019277	0.213853	-10.194094	0.067301	122.073977	0.472256
2	67.426434	0.047125	0.529716	229657.563591	0.850564	0.006563	0.192014	-4.746025	0.065772	128.667965	0.551676
3	75.758678	0.331110	0.639149	226252.380165	0.558936	0.005229	0.137344	-7.485140	0.086587	109.508625	0.465017
4	75.412720	0.098951	0.712591	222598.309878	0.726816	0.007689	0.152805	-6.069632	0.085998	124.267740	0.728788
5	49.510204	0.913959	0.371148	220860.678571	0.163349	0.029937	0.137917	-16.360301	0.045895	92.868061	0.235621
6	63.632653	0.089542	0.600694	250630.418367	0.663551	0.656520	0.179618	-8.430306	0.064705	123.105439	0.422050
7	34.606061	0.436332	0.777980	133019.050505	0.506313	0.000960	0.168798	-9.399778	0.090869	126.975919	0.819576
8	67.395604	0.068558	0.571495	274098.065934	0.745736	0.014144	0.721890	-6.152626	0.072954	119.805593	0.496822
9	80.968481	0.095911	0.744198	213640.020057	0.614670	0.005032	0.169700	-6.509648	0.157226	116.143576	0.352579
10	69.355469	0.087213	0.427422	268661.925781	0.711742	0.008507	0.151417	-5.902922	0.065725	132.661113	0.245745

Example Playlist Recommendation



CONCLUSIONS

- Agglomerative/Hierarchical clustering proved to be the best clustering algorithm for song recommendations
- When using agglomerative clustering, data size needs to be accounted for because it is resource intensive to custer many different rows of songs with a large number of attributes
- Clustering songs for recommendation performs better when there are an equal amount of each genre. This allows for one genre to not dominate the clustering as 'pop' normally would when including popularity as a variable because there are so many popular songs in the genre of 'pop'.
- -Categorical attributes such as genre are not as helpful for song recommendations because there are many different styles of songs in each genre so it would be limiting potential songs a user would like.
- This system would work better if it was able to process larger data size. A users song preferences would be better supported and each individual cluster would be populated with a lot more recommendations.
- Next step would be to experiment with other models that could provide recommendations with large data sets.
- It is hard to test the accuracy of a clustering model like this because a users preferences are entirely unique. But in my opinion this model did a good job at clustering songs of the similar vibe, mood, and feeling.