Week 8 – Unsupervised Learning

EXPLORING TEXT CLUSTERING OF SPOTIFY FEATURES.

Data: https://www.kaggle.com/datasets/thedevastator/spotify-tracks-genre-dataset

Intro

Using a large Spotify dataset, I wanted to see how well unsupervised learning would cluster songs based on their audio features. I predict that it will do well as the audio features are calculated by Spotify to be used in their recommendation algorithms. I want to see how it works.

Exploring the data

	Unnamed: 0	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
Unnamed: 0	1.000000	0.032142	-0.032743	-0.054736	0.003444	-0.055994	-0.005520	-0.027307	0.005107	-0.084952	0.076840	-0.070286	0.033639	0.053111	-0.025824	-0.021115
popularity	0.032142	1.000000	-0.007101	0.044082	0.035448	0.001056	-0.003853	0.050423	-0.013931	-0.044927	-0.025472	-0.095139	-0.005387	-0.040534	0.013205	0.031073
duration_ms	-0.032743	-0.007101	1.000000	-0.065263	-0.073426	0.058523	0.008114	-0.003470	-0.035556	-0.062600	-0.103788	0.124371	0.010321	-0.154479	0.024346	0.018225
explicit	-0.054736	0.044082	-0.065263	1.000000	0.122507	0.096955	0.004484	0.108588	-0.037212	0.307952	-0.094403	-0.103404	0.032549	-0.003381	-0.002816	0.038386
danceability	0.003444	0.035448	-0.073426	0.122507	1.000000	0.134325	0.036469	0.259077	-0.069219	0.108626	-0.171533	-0.185606	-0.131617	0.477341	-0.060450	0.207218
energy	-0.055994	0.001056	0.058523	0.096955	0.134325	1.000000	0.048006	0.761690	-0.078362	0.142509	-0.733906	-0.181879	0.184796	0.258934	0.247851	0.187126
key	-0.005520	-0.003853	0.008114	0.004484	0.036469	0.048006	1.000000	0.038590	-0.135916	0.020418	-0.040937	-0.006823	-0.001600	0.034103	0.010917	0.015065
loudness	-0.027307	0.050423	-0.003470	0.108588	0.259077	0.761690	0.038590	1.000000	-0.041764	0.060826	-0.589803	-0.433477	0.076899	0.279848	0.212446	0.191992
mode	0.005107	-0.013931	-0.035556	-0.037212	-0.069219	-0.078362	-0.135916	-0.041764	1.000000	-0.046532	0.095553	-0.049955	0.014012	0.021953	0.000566	-0.024092
speechiness	-0.084952	-0.044927	-0.062600	0.307952	0.108626	0.142509	0.020418	0.060826	-0.046532	1.000000	-0.002186	-0.089616	0.205219	0.036635	0.017273	-0.000011
acousticness	0.076840	-0.025472	-0.103788	-0.094403	-0.171533	-0.733906	-0.040937	-0.589803	0.095553	-0.002186	1.000000	0.104027	-0.020700	-0.107070	-0.208224	-0.176138
instrumentalness	-0.070286	-0.095139	0.124371	-0.103404	-0.185606	-0.181879	-0.006823	-0.433477	-0.049955	-0.089616	0.104027	1.000000	-0.079893	-0.324312	-0.050330	-0.082580
liveness	0.033639	-0.005387	0.010321	0.032549	-0.131617	0.184796	-0.001600	0.076899	0.014012	0.205219	-0.020700	-0.079893	1.000000	0.019086	0.000600	-0.023651
valence	0.053111	-0.040534	-0.154479	-0.003381	0.477341	0.258934	0.034103	0.279848	0.021953	0.036635	-0.107070	-0.324312	0.019086	1.000000	0.078273	0.133686
tempo	-0.025824	0.013205	0.024346	-0.002816	-0.060450	0.247851	0.010917	0.212446	0.000566	0.017273	-0.208224	-0.050330	0.000600	0.078273	1.000000	0.066641
time_signature	-0.021115	0.031073	0.018225	0.038386	0.207218	0.187126	0.015065	0.191992	-0.024092	-0.000011	-0.176138	-0.082580	-0.023651	0.133686	0.066641	1.000000

Figure 1: Heatmap of correlation between each feature.

I began with a lovely heatmap. I dropped all non-numerical columns here. There were a couple of features I looked up to see what the values meant on the data sheet such as mode and key, both of which have been given an integer value.

To begin, I immediately removed anything that was highly correlated. From reading the datasheet I felt that both pairs, 'Energy and Loudness' and 'Danceability and Valence' expressed similar things about the music and as they both had high correlations, I chose to remove one of each. I did this in hopes of it removing unnecessary data so that it would improve the model. 'energy' and 'acousticness' correlate quite highly across the entire dataset in comparison to the other features suggesting they are quite important.

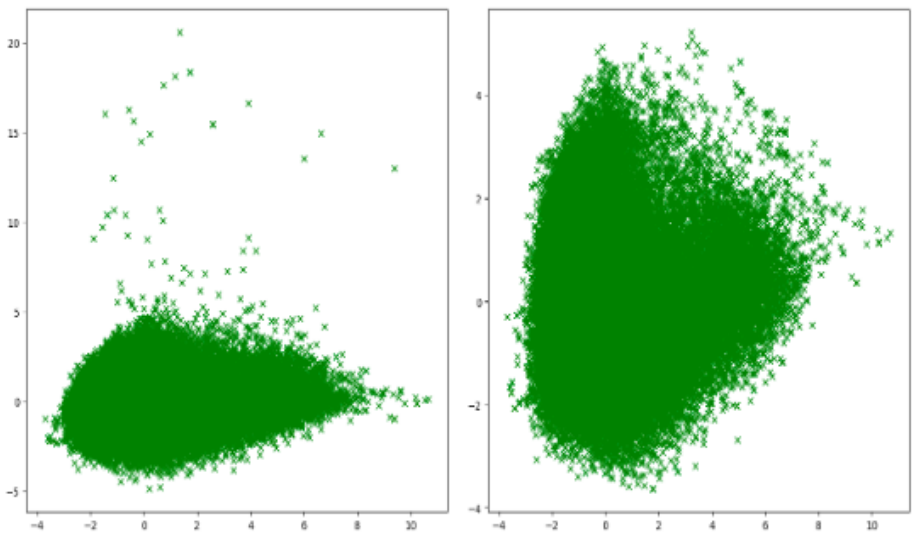

Figure 2: 2D Representation of data after PCA Dimensionality Reduction. Left, before resolving data issue. Right, after.

Going forward I ran lots of test to then realise that there was a feature that was incorrectly scaled/ not necessary. This made a big difference to my results, as you can see between these two graphs. Removing the duration feature gave more spread across the graph perhaps due to more focus on just the audio features.
There is a good use of PCA here as I have a data set with many features. PCA finds key patterns in the data where the variability or information is present, and it compresses the data into various bits that capture the most info, 'principal components'. It then only keeps the most important dimensions and discards less important ones making the data easier to work with. The result is a lower-dimensional representation of the data that still maintains a lot of its original information[1]. Doing this makes it possible to plot lots of information onto a 2D plot – Pretty cool!

My graph (see Figure 7) shows a lot of the songs grouped together. Whilst this could imply that lots of songs have very similar features, I think it is also worth noting that the dataset contains 114,000 songs, which is a lot of ticks on a 2D plot! Overall, the final PCA plot (on the right) shows that there is a spread of data and lots of it across said spread.

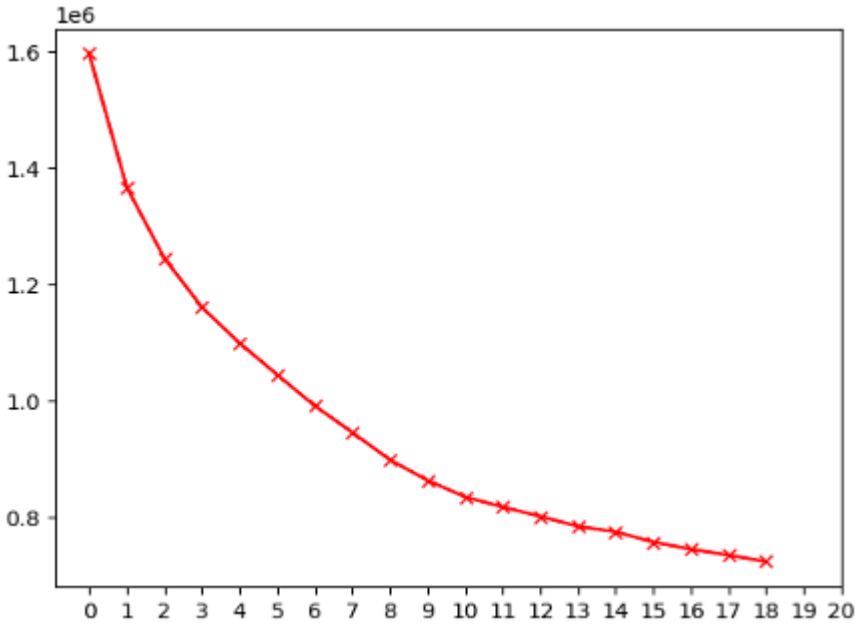

Figure 3: Elbow Plot

After scaling the data, it was plotted using an Elbow Plot to help find the optimal number of clusters. This technique shows where the rate of variance decreases. Where it does so sharply suggests a good k-means value[2]. Looking at my graph it has quite a steady decline, I settled on a k-means of 4 after examining the graph and experimenting through trial and error.

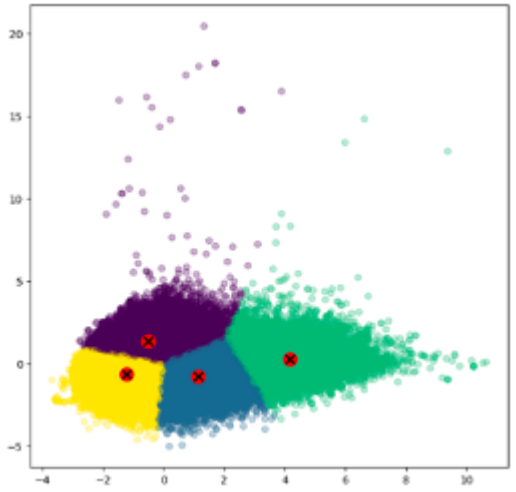

Figure 5: K-means 4, Dimensions 2. Silhouette Score 0.3815816680338216 Before removing invalid data.

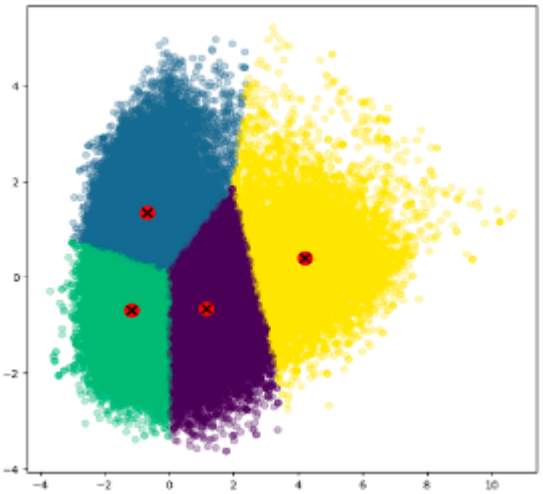

Figure 5: K-means 4, Dimensions 2 Silhouette Score:0.38581822915037844 After removing invalid data.

Clustering

I ran lots of tests exploring the Silhouette Scores and cluster plots before finding the scaling error, so here are some more examples of the data before and after. Interestingly the Silhouette Scores remained almost unchanged.

I chose to look at the Silhouette Scores as I was finding it difficult to decide where to leave the parameters for the model based solely on the visual information. The Silhouette Score gives a value that describes the spread of the clusters, with 1 being the best and -1 the worst and a score of 0 indicates overlapping[3]. After various tests of trial and error I stuck with a K-means of 4 as this gave me the best Silhouette Score of 0.39.

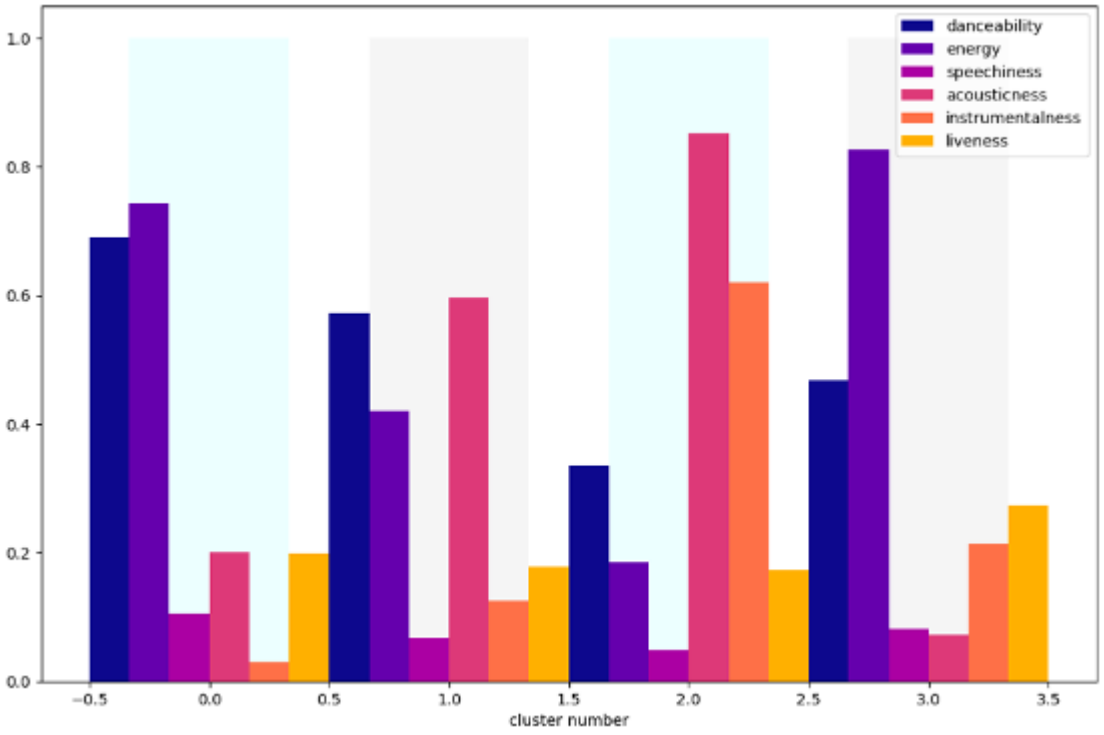

Figure 6: Plot of feature averages in each cluster. K-means 4, dimensions 2. With final data set.

With my final parameters set I plotted the clusters as a bar graph to have a look at what features were most prominent in each. At first glance I can expect Cluster0 to have high energy dance genres and Cluster2 to be the opposite with more acoustic and instrumental genres. There is lovely variation within each cluster, so I think it has done quite a good job so far!

Deeper look into the clusters.

	popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
count	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000	33443.000000
mean	33.278324	0.468975	0.825602	5.403223	-5.880927	0.593458	0.081639	0.073521	0.214427	0.273279	0.339261	137.013163	3.920731
std	20.373938	0.140559	0.142836	3.562011	2.518661	0.491195	0.076415	0.140796	0.335030	0.228386	0.197791	29.034120	0.353664
min	0.000000	0.051300	0.222000	0.000000	-22.398000	0.000000	0.022600	0.000000	0.000000	0.013300	0.000000	36.950000	0.000000
25%	19.000000	0.377000	0.728000	2.000000	-7.280000	0.000000	0.040200	0.000628	0.000001	0.107000	0.183000	119.898500	4.000000
50%	34.000000	0.475000	0.867000	6.000000	-5.567000	1.000000	0.056400	0.008620	0.001130	0.188000	0.317000	135.984000	4.000000
75%	49.000000	0.580000	0.945000	9.000000	-4.139000	1.000000	0.093000	0.071800	0.399000	0.354000	0.475000	158.070000	4.000000
max	98.000000	0.970000	1.000000	11.000000	4.532000	1.000000	0.962000	0.947000	0.999000	1.000000	0.988000	222.605000	5.000000

Figure 7: Cluster3 column descriptions.

With the first run of tests this is where I discovered my error! I looked at each clusters description to discover that the ‘duration’ feature was looking very odd. The duration of the songs before being scaled is represented in milliseconds. I don’t think the duration of the song will have an impact on what genre it falls into, so I went back and dropped the column to rerun the tests. Looking at the comparisons of the graphs before and after this decision I think it was the correct one to make. After going back and updating everything it was finally time to investigate the quality of the clusters and see if it had grouped by genre as expected.

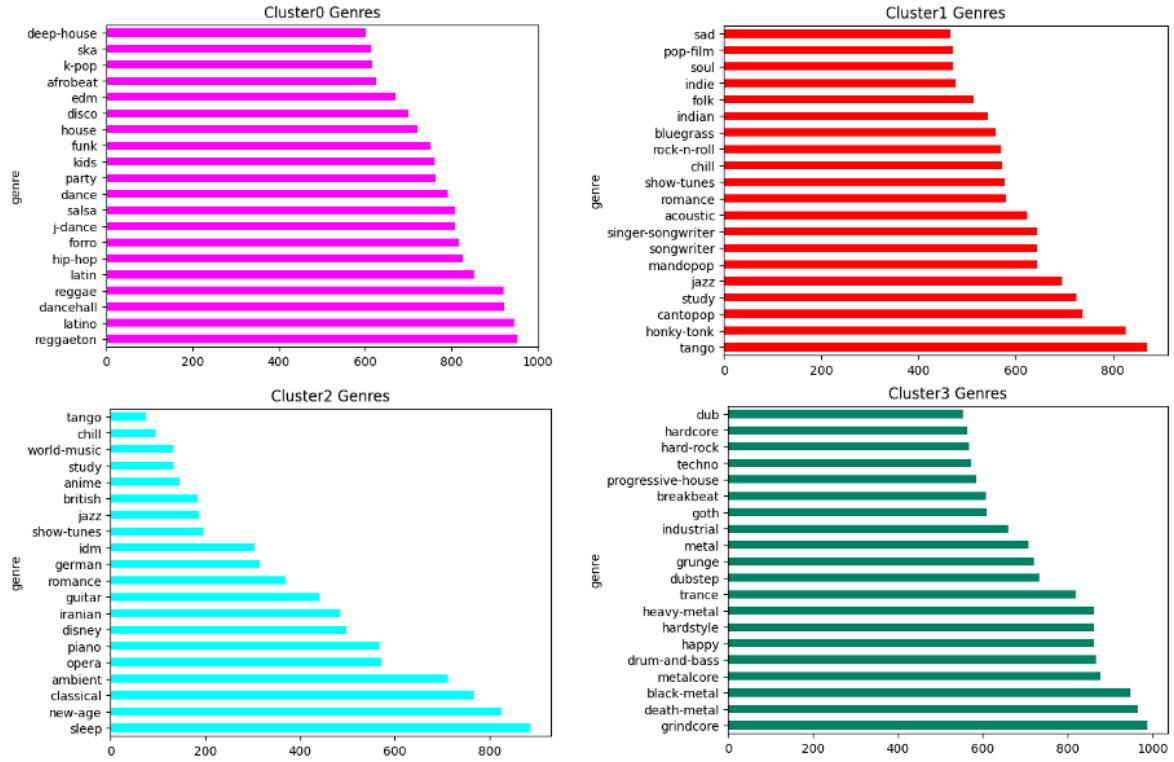

Figure 8 : Top 20 Genres across each cluster.

The grouped genres appear to make a lot of sense, phew! Upbeat genres such as disco, dance and salsa are together in Cluster0, with more relaxed and laid-back genres such as sleep, classical and piano in Cluster2. Cluster3 is filled with heavy genres such as death metal and grunge. This aligns well with my initial assumptions! The clusters vary in size, the largest being Cluster0 with 42,901 songs and the smallest is Cluster2 with 9,282 songs. The others are larger with over 20,000 and 30,000 respectively. I don't think the variation in size is a problem, it could just be that there are less genres like 'Honky-tonk' and 'Cantopop' in the dataset, which makes sense. Looking at Cluster3 in more detail, it has a very low 'acousticness' mean, and a very high 'energy' mean which relates well to the clustered genre types. 'Heavy-metal' and 'Drum-and-bass' are both genres that are high in energy with broad sounds that are far from acoustic. Looking at the 'mode' mean, which is whether the song is minor (0) or major (1), it appears there is an even distribution of modes across this cluster, with the other clusters they tend towards major keys. 'Valance', how positive the song is, is low here but the lowest average is in Cluster2 where the genres are more ambient and slower.

Reflection

Overall, I think the model has clustered very well! I stumbled a bit along the way and struggled with settling on final parameters however, I have learnt a lot about the process of exploring. I think this was a strong technique to use for this data set and task and that if was very effective in helping me reach the objective. I now have a very strong understanding of how to use the clustering technique and how it can be applied in real-life scenarios. Going back through the code upon realising I wanted to remove the 'duration' feature at first was

very frustrating but in the end, it showed me the big difference selecting only the necessary features can make. It made my plots more readable and improved the mode. I am interested in taking this exercise further and looking at a larger number of clusters, to see if the model is consistent with clustering the genres with more clustering groups available. I'd also like to take my own music that has been released and seeing how Spotify sees the music in terms of its audio features and explore how that impacts where my songs are recommended through their algorithm.

---

[1] PCA using python: A tutorial (no date) Built In. Available at: https://builtin.com/machine-learning/pca-in-python (Accessed: 08 March 2024).

[2] Saji, B. (2024) Elbow method for finding the optimal number of clusters in K-means, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/ (Accessed: 08 March 2024).

[3] Sklearn.metrics.silhouette_score (no date) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html (Accessed: 09 March 2024).