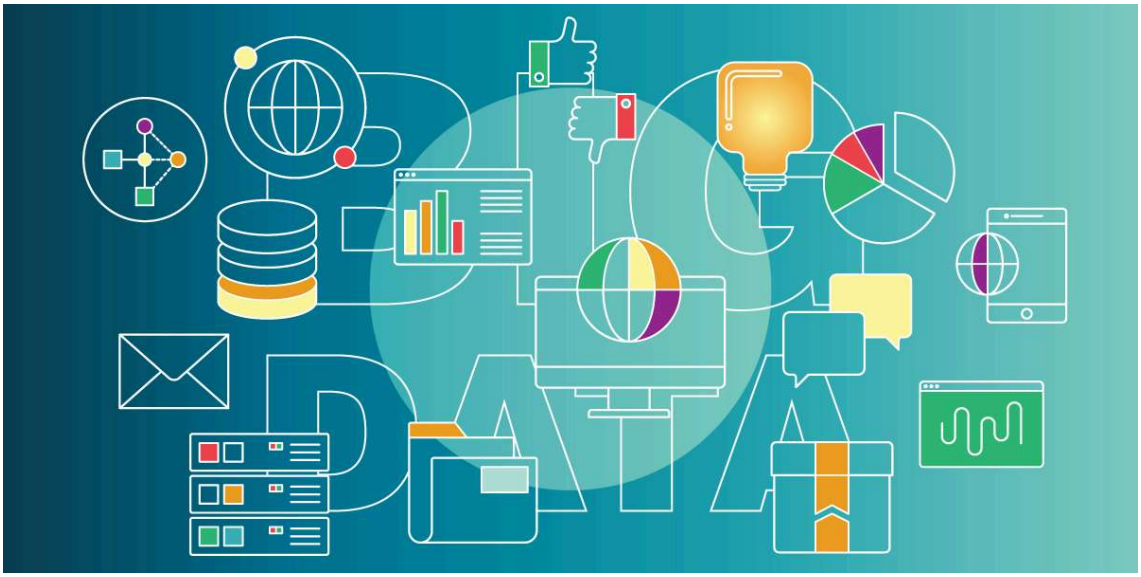
Διαχείριση Δεδομένων Μεγάλης Κλίμακας

Εργασία Εξαμήνου

Ονοματεπώνυμα / Α.Μ:
Άγγελος Τζώρτζης / ice18390094
Μιχάλης Κατσουλάκης / ice18390148
Γιάννης Παλάσης / ice18390175



Ημερομηνία παράδοσης: 21/06/2024

Εισαγωγή

Τα δεδομένα που κυκλοφορούν αυτή την στιγμή είναι άπειρα και αυξάνονται με εκθετικό ρυθμό. Λίγοι όμως ξέρουν να τα εκμεταλλευτούν πρός όφελος τους και να αντλήσουν ουσιώδες πληροφορίες. Παρακάτω θα δούμε ένα απλοποιημένο recommender system μουσικής, το οποίο με βάση κάποια χαρακτηριστικά τραγουδιών, μας προτείνει αυτά που έχουν τα χαρακτηριστικά που ορίζουμε. Γίνεται χρήση της μεθόδου clustering με τον αλγόριθμο k-means.

Ορισμός του προβλήματος και κίνητρο:

Θέλουμε να κάνουμε ένα πάρτυ και να φτιάξουμε ένα κατάλληλο playlist επιλέγοντας τραγούδια από ένα dataset που έχουμε. Όμως καθώς είναι μεγάλος ο αριθμός των τραγουδιών δεν θέλουμε και δεν είναι εφικτό να ακούσουμε ένα ένα τα τραγούδια και να επιλέξουμε εάν είναι κατάλληλα ή όχι. Οπότε με το σύστημα μας θα ορίσουμε κάποια χαρακτηριστικά που θέλουμε να έχουν τα τραγούδια μας και θα μας εμφανίζει ποια τραγούδια ανήκουν στην κατηγορία που θέλουμε. Ένα τέτοια σύστημα θα ήταν χρήσιμο σε εφαρμογές παραγωγής μουσικής (π.χ. Spotify, Apple Music, κλπ…) όπου ο χρήστης εισάγει τις προτιμήσεις του σε μουσική και δημιουργείται μία playlist με τραγούδια που ανήκουν στις κατηγορίες αυτές.

Περιγραφή του συνόλου δεδομένων:

Το σύνολο δεδομένων περιέχει τα 2000 δημοφιλέστερα τραγούδια στο Spotify απο τις χρονιές 2000-2019.

Τα τραγούδια περιγράφονται με τα εξής χαρακτηριστικά:

- artist: Όνομα του καλλιτέχνη/συγκροτήματος.
- song: Όνομα του τραγουδιού.
- duration_ms: διάρκεια του τραγουδιού σε milliseconds.
- explicit: Boolean τιμή που ορίζει ένα οι στίχοι του τραγουδιού είναι ακατάλληλοι για για μικρά παιδιά.
- year: Χρόνος που δημοσιεύτηκε το τραγούδι.
- popularity: Ορίζει την δημοσιότητα του τραγουδιού (τιμές 0-100).
- danceability: Ορίζει πόσο κατάλληλο είναι το τραγούδι για χορό (τιμές 0 - 1).
- energy: Αντιπροσωπεύει την ενέργεια και ένταση του τραγουδιού (τιμές 0 - 1).
- key: Ο μουσικός τόνος τον οποίο έχει το τραγούδι (παίρνει θετικές ακέραιες τιμές για κάθε διαφορετικό τόνο και -1 άν δέν έχει εντοπιστεί).
- loudness: Η ένταση του ήχου του τραγούδι σε db (τιμές -60 - 0).
- mode: Δείχνει αν το τραγούδι είναι σε μινόρε ή ματζόρε.
- speechiness: Εντοπίζει την ύπαρξη στίχων στο τραγούδι (τιμές 0 - 1).
- acousticness: Μέτρο που μας δείχνει εάν περιέχονται ακουστικά μουσικά όργανα (τιμές 0 - 1).
- instrumentalness: Εντοπίζει εάν το τραγούδι δεν περιέχει φωνητικά (τιμές 0-1).
- liveness: Εντοπίζει εάν υπάρχει κοινό στην καταγραφή του τραγουδιού (τιμές 0-1).
- valence: Μέτρο που περιγράφει τον θετικό τόνο του τραγουδιού (τιμές 0-1).
- tempo: Αριθμός BPM (beats per minute) του τραγουδιού.
- genre: Κατηγορία που ανήκει το τραγούδι.

Μέθοδος ανάλυσης δεδομένων και αποτελέσματα:

Όπως αναφέραμε και προηγουμένως χρησιμοποιούμε την μέθοδο του clustering με τον αλγόριθμο k-means. Επιλέχθηκε αυτός ο αλγόριθμος καθώς χρησιμοποιείται για την ομαδοποίηση στοιχείων με κοινά χαρακτηριστικά που είναι αυτό που θέλουμε να κάνουμε με τα τραγούδια στο σύνολο δεδομένων.

Αρχικά εισάγουμε τα τραγούδια από το σύνολο δεδομένων μας στο πρόγραμμα.

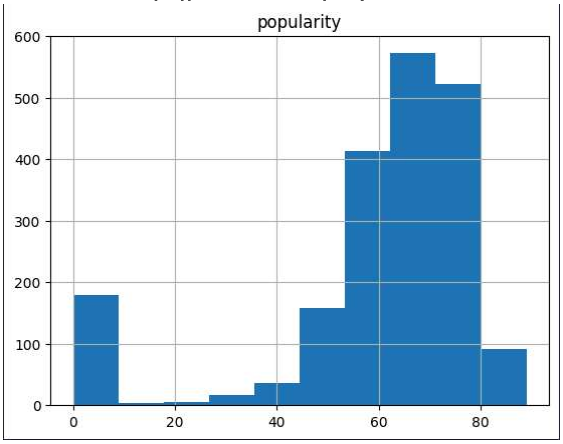
	artist	song	duration_ms	explicit	year	popularity	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	genre
0	Britney Spears	Oops!...I Did It Again	211160	False	2000	77	0.751	0.834	1	-5.444	0	0.0437	0.3000	0.000018	0.3550	0.894	95.053	pop
1	blink-182	All The Small Things	167066	False	1999	79	0.434	0.897	0	-4.918	1	0.0488	0.0103	0.000000	0.6120	0.684	148.726	rock, pop
2	Faith Hill	Breathe	250546	False	1999	66	0.529	0.496	7	-9.007	1	0.0290	0.1730	0.000000	0.2510	0.278	136.859	pop, country
3	Bon Jovi	It's My Life	224493	False	2000	78	0.551	0.913	0	-4.063	0	0.0466	0.0263	0.000013	0.3470	0.544	119.992	rock, metal
4	*NSYNC	Bye Bye Bye	200560	False	2000	65	0.614	0.928	8	-4.806	0	0.0516	0.0408	0.001040	0.0845	0.879	172.656	pop
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	Jonas Brothers	Sucker	181026	False	2019	79	0.842	0.734	1	-5.065	0	0.0588	0.0427	0.000000	0.1060	0.952	137.958	pop
1996	Taylor Swift	Cruel Summer	178426	False	2019	78	0.552	0.702	9	-5.707	1	0.1570	0.1170	0.000021	0.1050	0.564	169.994	pop
1997	Blanco Brown	The Git Up	200593	False	2019	69	0.847	0.678	9	-8.635	1	0.1090	0.0669	0.000000	0.2740	0.811	97.984	hip hop, country
1998	Sam Smith	Dancing With A Stranger (with Normani)	171029	False	2019	75	0.741	0.520	8	-7.513	1	0.0656	0.4500	0.000002	0.2220	0.347	102.998	pop
1999	Post Malone	Circles	215280	False	2019	85	0.695	0.762	0	-3.497	1	0.0395	0.1920	0.002440	0.0863	0.553	120.042	hip hop

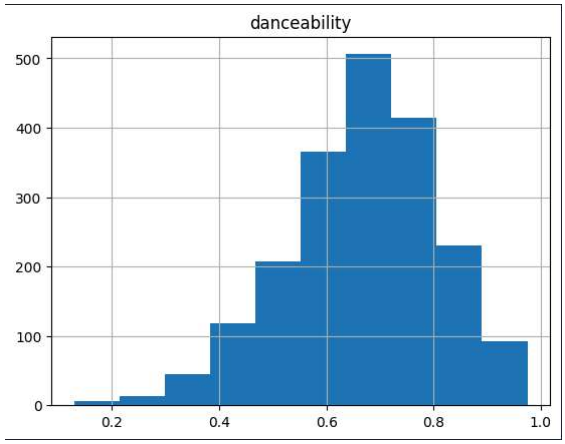
Καθώς για την δικιά μας ανάλυση θα χρειαστούν μόνο τα στοιχεία popularity, danceability, energy, valence, τα υπόλοιπα τα διαγράφουμε από το dataframe με τα τραγούδια.

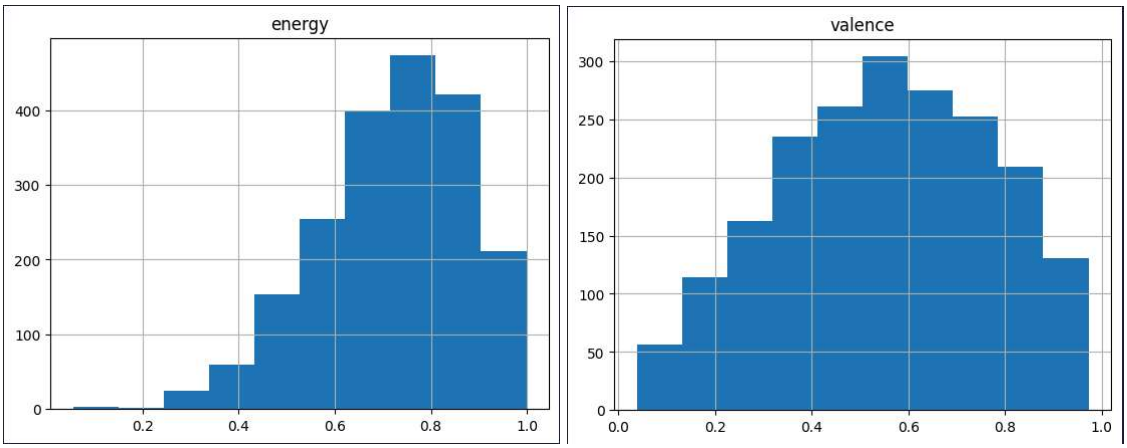
	artist	song	popularity	danceability	energy	valence
0	Britney Spears	Oops!...I Did It Again	77	0.751	0.834	0.894
1	blink-182	All The Small Things	79	0.434	0.897	0.684
2	Faith Hill	Breathe	66	0.529	0.496	0.278
3	Bon Jovi	It's My Life	78	0.551	0.913	0.544
4	*NSYNC	Bye Bye Bye	65	0.614	0.928	0.879
...	...	...	...	...	...	...
1995	Jonas Brothers	Sucker	79	0.842	0.734	0.952
1996	Taylor Swift	Cruel Summer	78	0.552	0.702	0.564
1997	Blanco Brown	The Git Up	69	0.847	0.678	0.811
1998	Sam Smith	Dancing With A Stranger (with Normani)	75	0.741	0.520	0.347
1999	Post Malone	Circles	85	0.695	0.762	0.553

2000 rows × 6 columns

Φτιάχνουμε και τα ιστογράμματα των χαρακτηριστικών που θέλουμε για να δούμε πώς είναι κατανεμημένα στα τραγούδια.



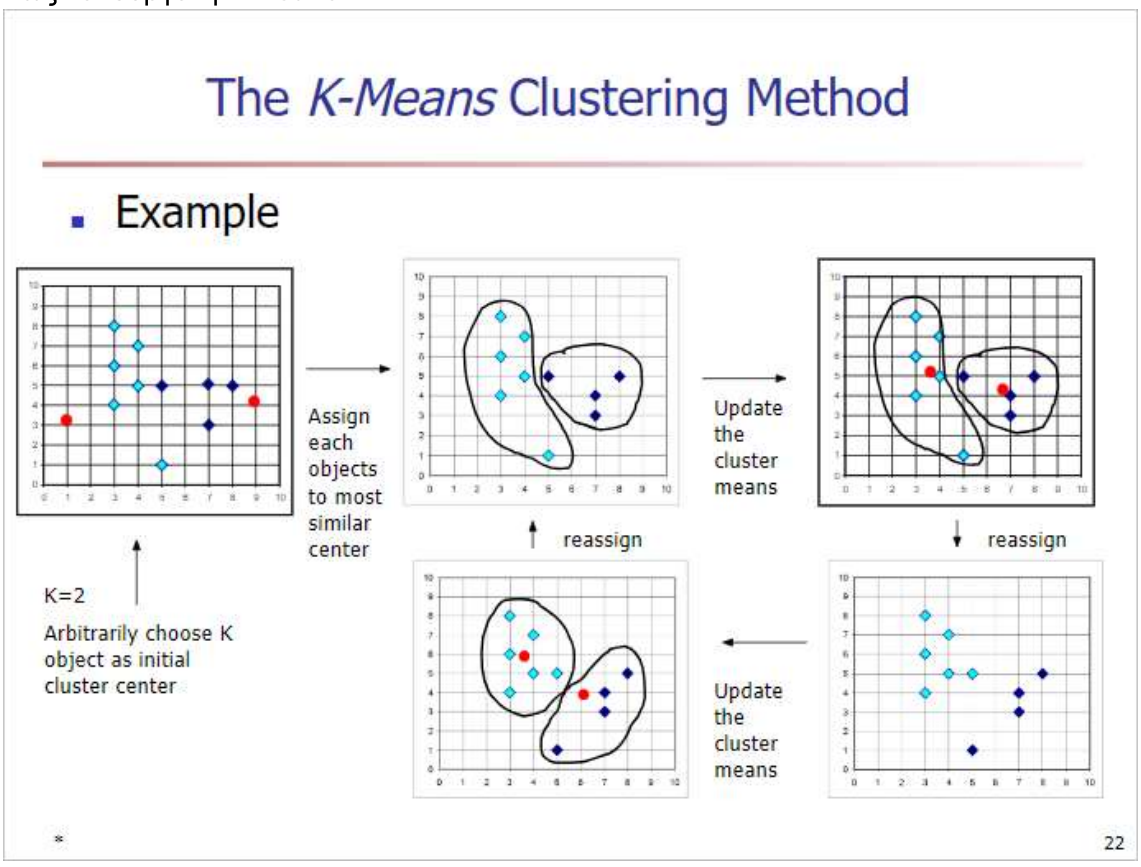




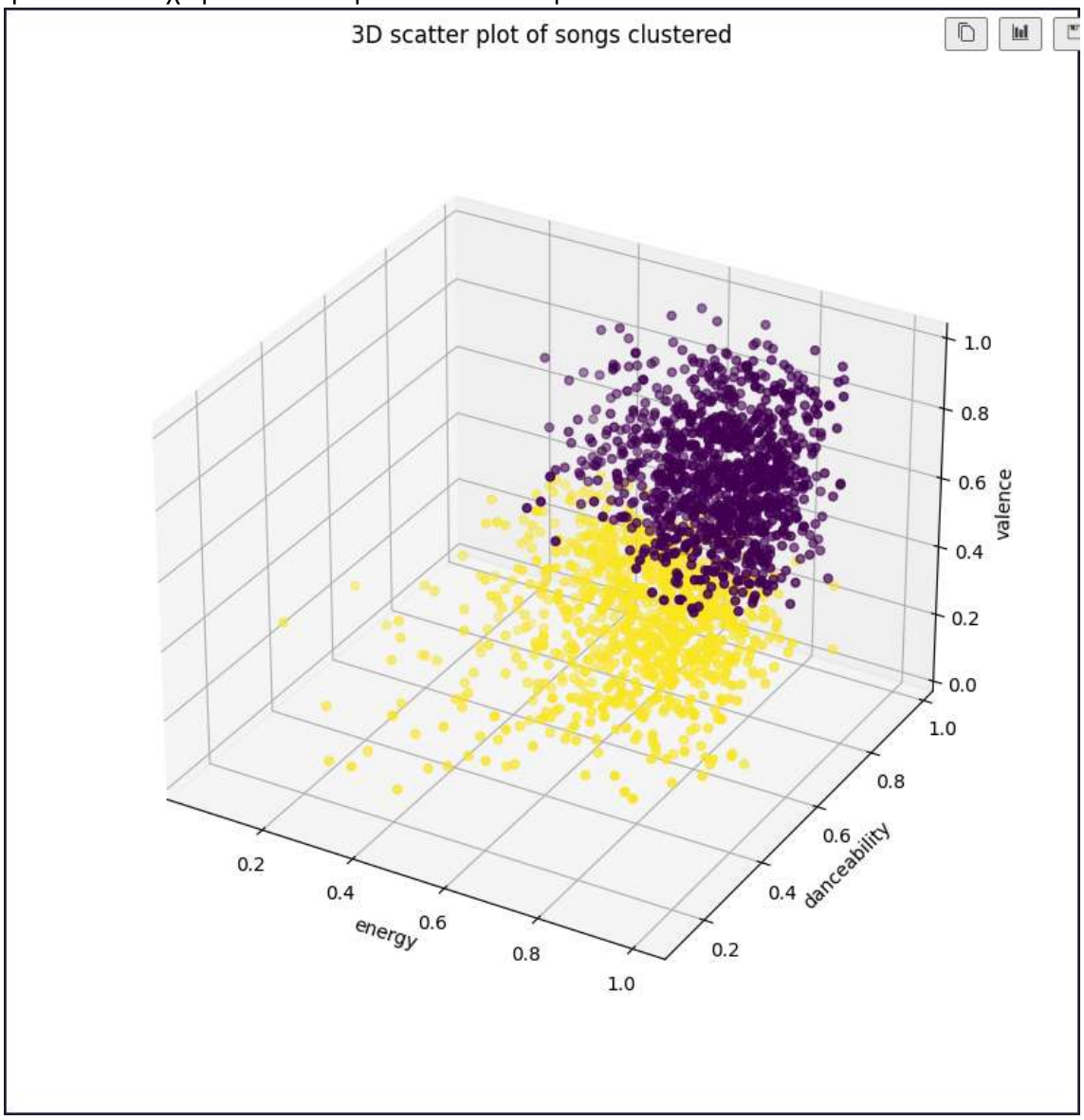
Επιλέξαμε να κάνουμε το clustering με βάση των στοιχείων energy, danceability, valence. Επιλέξαμε αυτά τα στοιχεία καθώς θέλουμε τα τραγούδια του πάρτυ να είναι κατάλληλα για χορό, να δίνουν ενέργεια σε αυτόν που τα ακούει και να έχουν ευχάριστη διάθεση.

Στην συνέχεια γίνεται κανονικοποίηση των δεδομένων που θέλουμε να χρησιμοποιήσουμε.

Πώς λειτουργεί η k-means:



Εφαρμόζουμε τον αλγόριθμο k-means με 2 clusters. Ο ένας θα αντιπροσωπεύει τα τραγούδια που έχουν υψηλή τιμή στα στοιχεία που ορίσαμε και ο άλλος αυτά που έχουν χαμηλή τιμή. Επίσης καθώς έχουμε 3 δεδομένα θα δείξουμε το scatterplot σε τρισδιάστατο χώρο ώστε να φαίνονται καλύτερα.



Στην συνέχεια βρίσκουμε στην κάθε συστάδα τον μέσο όρο των δεδομένων που θέλουμε για τα τραγούδια μας.

kmeans	popularity	danceability	energy	valence
0	59.526667	0.727390	0.764780	0.723974
1	60.254737	0.601175	0.671277	0.361270

Βλέπουμε ότι το cluster 0 έχει υψηλότερη τιμή και στα 3 πεδία που θέλουμε, άρα περιέχει και τα τραγούδια που θέλουμε.

Βάζουμε τα 100 πρώτα τραγούδια της πρώτης συστάδας σε ένα dataframe και τα κατατασουμε με βάση την τιμή danceability σε φθίνουσα σειρά. Επιλέγουμε το πεδίο danceability ώς βάση της κατάταξης καθώς έχουμε θέσει σαν προτεραιότητα το τραγούδι να είναι χορευτικό και να ταιριάζει στο πάρτυ μας. Επίσης υπάρχουν τραγούδια με χαμηλή βαθμολογία χορευτικότητας που μπήκαν στην συστάδα μας λόγο υψηλής τιμής στα άλλα πεδία.

	artist	song	popularity	danceability	energy	valence	kmeans
714	Timbaland	Give It To Me	70	0.975	0.711	0.815	0
425	Kelis	Trick Me	63	0.970	0.720	0.962	0
225	Missy Elliott	4 My People (feat. Eve)	49	0.969	0.701	0.905	0
602	Justin Timberlake	SexyBack (feat. Timbaland)	78	0.967	0.583	0.964	0
618	Ciara	Get Up (feat. Chamillionaire)	59	0.964	0.595	0.629	0
...	...	...	...	...	...	...	...
840	Dizzee Rascal	Dance Wiv Me - Radio Edit	68	0.878	0.746	0.792	0
552	Destiny's Child	Soldier (feat. T.I. & Lil' Wayne)	63	0.878	0.417	0.904	0
1900	Lil Nas X	Old Town Road - Remix	79	0.878	0.619	0.639	0
509	Akon	Bananza (Belly Dancer)	28	0.878	0.699	0.666	0
146	Jamiroquai	Little L	65	0.878	0.724	0.904	0

Βέβαια υπάρχει μια ενδιαφέρουσα παρατήρηση, πως το cluster που θεωρητικά περιέχει τα τραγούδια που δεν ταιριάζουν στο παρτυ έχει αρκετά τραγούδια που όσοι τα ξέρουν θα έλεγαν ταιριάζουν σε πάρτυ και έχουν υψηλή τιμή danceability και έχουν μείνει εκτός λόγο χαμηλής energy ή valence.

	artist	song	popularity	danceability	energy	valence	kmeans
1948	Cardi B	Money	73	0.950	0.590	0.219	1
1823	6ix9ine	FEFE	42	0.931	0.387	0.376	1
1243	Tyga	Rack City	61	0.929	0.339	0.273	1
1753	Migos	Bad and Boujee (feat. Lil Uzi Vert)	72	0.926	0.666	0.168	1
19	Dr. Dre	The Next Episode	82	0.922	0.909	0.309	1
...	...	...	...	...	...	...	...
1903	Ariana Grande	7 rings	83	0.778	0.317	0.327	1
233	Disturbing Tha Peace	Move Bitch	59	0.777	0.751	0.191	1
1526	Lost Frequencies	Are You With Me - Radio Edit	33	0.776	0.574	0.412	1
1695	Beyoncé	Sorry	67	0.775	0.598	0.356	1
834	Colby O'Donis	What You Got	61	0.775	0.641	0.305	1

Τέλος βρίσκουμε τις τιμές SSE (sum of squared errors) και silhouette coefficient για να διαπιστώσουμε πόσο ακριβής ήταν ο αλγόριθμος.

```
SSE(sum of squared errors): 127.89909312150701
Silhouette Coefficient: 0.355092483727035
```

Ο SSE μας δείχνει το τετράγωνο των αποστάσεων του κάθε σημείου ενός cluster από το κέντρο του. Όσο μικρότερη η τιμή τόσο κοντα είναι τα σημεία στο κέντρο και τόσο πιό ακριβής το clustering. Το silhouette coefficient μας δείχνει πόσο κοινά έχουν τα στοιχεία μιας συστάδας με την συστάδα που βρίσκονται και παίρνει τιμές από -1 εώς 1. Αρνητική τιμή σημαίνει ότι έχει γίνει λάθος συσταδοποίηση.

Έστω ότι τρέχουμε το ξανά την kmeans με τις ίδιες παραμέτρους, δεδομένα και τιμές και αλλάζουμε μονο το κ ώστε να έχουμε 100 clusters.

```
SSE(sum of squared errors): 9.475577120480581
Silhouette Coefficient: 0.25271620131906125
```
Βλέπουμε ότι ενώ το SSE έχει μειωθεί δραματικα, έχει μειωθεί και το silhouette coefficient που μας αποδεικνύει πώς μικρότερο SSE δεν σημαίνει απαραίτητα και καλύτερα αποτελέσματα.

Χρόνοι εκτέλεσης της K-means:
Για k = 2: 78.1 ms
Για k = 100: 1.7s

Συζήτηση/Κριτική αποτίμηση αποτελεσμάτων:
Ενώ η συστάδα μας περιέχει τα αποτελέσματα που θέλαμε με τις προδιαγραφές που δώσαμε, υπήρχαν και τραγούδια που θα ταίριαζαν στο πάρτι που δεν εντόπισε. Επίσης επιλέχθηκαν τραγούδια τα οποία δεν ταιριάζουν να ακουστούν σε πάρτι. Για αυτό επιλέχθηκαν μόνο τα πρώτα 100 τραγούδια από το cluster ώστε να είναι όσο πιο ακριβή γίνεται τα αποτελέσματα. Η δυσκολία οφείλεται στο γεγονός πως οι αλγόριθμοι συσταδοποίησης δέν έχουν σχεδόν ποτέ 100% ακρίβεια και ότι δεν υπήρχαν 2 ξεκάθαρες συστάδες να εντοπίσει το πρόγραμμα μας και υπήρχαν τιμές πολύ κοντά η μία στην άλλη με αποτέλεσμα να μην μπορεί να γίνει ξεκάθαρος διαχωρισμός. Ίσως η επιλογή άλλων πεδίων για τον διαχωρισμό να έφερνε καλύτερα αποτελέσματα εάν και τα τραγούδια που επιλέχθηκαν αμα τα ακούσουμε, θα διαπιστώσουμε ότι ταιριάζουν στον σκοπό της εργασίας αυτής. Επίσης ίσως θα μπορούσε να γίνει clustering χωρίς το energy καθώς έχει παρόμοιο ιστόγραμμα με το danceability και τα τραγούδια με υψηλή τιμή σε από αυτά τα πεδία έχουν και στο άλλο. Επίσης παρατηρείται μεγάλη αύξηση στο χρόνο εκτέλεσης για 100 συστάδες το οποίο είναι λογικό.

Συμπεράσματα:
Τα αποτελέσματα αν και κυρίως σωστά δέν ήταν πλήρες. Πιθανόν για πιο ολοκληρωμένο αποτέλεσμα να χρειάζεται άλλος αλγόριθμός clustering όπως ή DBSCAN. Επίσης σε μία πραγματική εφαρμογή θα γινόταν χρήση ενός μεγαλύτερου συνόλου δεδομένων και θα είχαμε πιο καθαρά αποτελέσματα αλλα δεν ήταν εφικτό λόγο της απαιτούμενης υπολογιστικής δύναμης. Παρόλα αυτά το σύστημα μας θα είναι αρκετά ικανοποιητικό, καθώς μπορεί να κάνει και επιπλέον αναλύσεις με τα υπολοιπα πεδία του συνόλου δεδομένων.

Εξωτερικές πηγές:
Dataset:
https://www.kaggle.com/datasets/paradisejoy/top-hits-spotify-from-20002019

Εικόνα για την εξήγηση της k-means: Διαφάνειες κ.Περικλή Ανδρίτσου:
INF2190H-Clustering.ppt
https://eclass.uniwa.gr/modules/document/?course=ICE359
Cluster Analysis