Analyse et Clustering de Données Spotify

Thème du Projet : Exploration et segmentation des données musicales Spotify

Team :

- BENHAMMADI Ilyas P2513238.
- KACEMI Oualid P2413510.
- GUELFOUT Oussama P2310587.

Table of Contents

1. Introduction

Spotify est la plus grande plateforme de streaming musical au monde avec plus de 500 millions d'utilisateurs. Chaque jour, des milliards de chansons sont écoutées. Spotify analyse les caractéristiques audio de chaque morceau pour recommander de la musique, créer des playlists, et comprendre les tendances musicales.
Ce projet analyse un ensemble de données de Spotify pour découvrir des patterns dans la musique - comment les chansons se regroupent, quelles caractéristiques audio rendent une chanson populaire, et comment différents genres se distinguent les uns des autres.

2. Le Jeu de Données

Nous avons travaillé avec deux fichiers CSV depuis Kaggle, contenant des informations sur les artistes et les chansons de Spotify. Le code complet de ce projet est disponible sur GitHub : `https://github.com/Mitchi-02/spotify-data-analysis-case-study`

2.1. Données des Artistes (artists.csv)

Ce fichier contient des statistiques globales sur les artistes :

- Noms des artistes (de Drake à The Weeknd)
- Total de streams cumulés (en tant qu'artiste principal)
- Nombre de morceaux publiés par artiste
- Nombre de morceaux avec plus d'1 milliard de streams
- Nombre de morceaux avec plus de 100 millions de streams
- Nombre de collaborations en featuring (artistes invités)

Ces données nous permettent de voir quels artistes dominent Spotify et comment leur succès se manifeste en termes de streams et de nombre de hits.

2.2. Données des Morceaux (tracks.csv)

Ce fichier contient les détails de chaque chanson :

- Informations de base : nom du morceau, artiste(s), genre, date de sortie

- Score de popularité (0-100) calculé par Spotify basé sur les streams récents
- Caractéristiques audio qu'un algorithme Spotify calcule pour chaque chanson

Les caractéristiques audio sont particulièrement importantes car elles permettent à Spotify de comprendre la nature musicale de chaque track, indépendamment de qui l'a créée.

Caractéristique	**Ce qu'elle mesure**
danceability	Facilité à danser (échelle 0-1)
Energy	Intensité et activité de la chanson (0-1)
Loudness	Volume global en décibels
Speechiness	Proportion de paroles vs chant (0-1)
Acousticness	Instruments acoustiques vs électroniques (0-1)
Instrumentalness	Pas de voix vs voix dominante (0-1)
Valence	Positivité/joie de la chanson (0-1)
Tempo	Vitesse en battements par minute (BPM)

3. Nettoyage des Données

Avant de pouvoir analyser les données, il a fallu les nettoyer. Les données du monde réel ne sont jamais parfaites, et les données de Spotify ne font pas exception.

3.1. Problèmes rencontrés et solutions

1. Nombres avec virgules (malformat numérique)

Les comptages de streams avaient des virgules comme séparateurs de milliers. Par exemple, Drake avait "50,162,292,808" streams - c'est 50 milliards de streams ! Mais Python lit ça comme du texte, pas comme un nombre. Impossible de faire des calculs avec.

Solution : Enlever les virgules et convertir en nombre entier. Maintenant on peut faire des comparaisons et des calculs statistiques sur les streams.

2. Valeurs manquantes dans les caractéristiques audio

Certaines chansons n'avaient pas de données pour quelques caractéristiques audio de Spotify. C'est rare mais ça arrive - peut-être parce que la chanson est très ancienne ou parce que Spotify n'a pas pu analyser le fichier audio.

Solution : Remplir les valeurs manquantes avec la **médiane** de la colonne. La médiane est meilleure que la moyenne car elle n'est pas affectée par les valeurs extrêmes. Par exemple, la danceability médiane était 0.580 - si une chanson n'avait pas cette info, on utilise 0.580.

Au lieu de supprimer des chansons complètes à cause de quelques valeurs manquantes, on les garde dans l'analyse.

3. Noms de colonnes bizarres

Certains en-têtes de colonnes avaient des noms problématiques. Par exemple, "Artist Name" au lieu d'un nom court comme "artist". En Python, accéder à une colonne avec un nom long et des espaces, c'est compliqué.

Solution : Renommer les colonnes en noms simples et standards. Maintenant le code est plus lisible et plus facile à maintenir.

3.2. Nouvelles caractéristiques créées

- Scores de popularité normalisés (échelle 0-100) basés sur les comptages de streams
- Nombre d'artistes par morceau (solo vs collaborations)
- Durée des chansons convertie de millisecondes en minutes pour la lisibilité

4. Analyse du Clustering

Spotify a des millions de chansons. Comment recommander la bonne chanson au bon utilisateur ? Une approche : grouper les chansons par similarité. Si un utilisateur aime une chanson dans un groupe, il aimera probablement d'autres chansons du même groupe.

Le clustering automatise ce processus. Au lieu de catégoriser manuellement chaque chanson, les algorithmes trouvent automatiquement des groupes de chansons similaires basés sur leurs caractéristiques audio.

Nous avons testé deux algorithmes : K-Means et DBSCAN.

4.1. K-Means Clustering

K-Means est un algorithme classique et populaire. L'idée de base : regrouper les chansons en K clusters de sorte que les chansons dans le même cluster soient aussi similaires que possible.

4.1.1. Comment ça marche

D'abord, nous avons standardisé les données avec StandardScaler. C'est important car les caractéristiques ont des échelles différentes - le tempo va de 60 à 200 environ, mais la danceability n'est que de 0 à 1. Sans standardisation, les nombres plus grands dominent l'analyse.

Ensuite, nous avons utilisé la “méthode du coude” pour déterminer combien de clusters utiliser. Nous avons essayé différentes valeurs de K (de 1 à 10) et tracé les résultats. Le graphique ressemble à un coude, et là où il se plie est généralement la meilleure valeur de K. Pour ce jeu de données, K=4 semblait bien fonctionner.

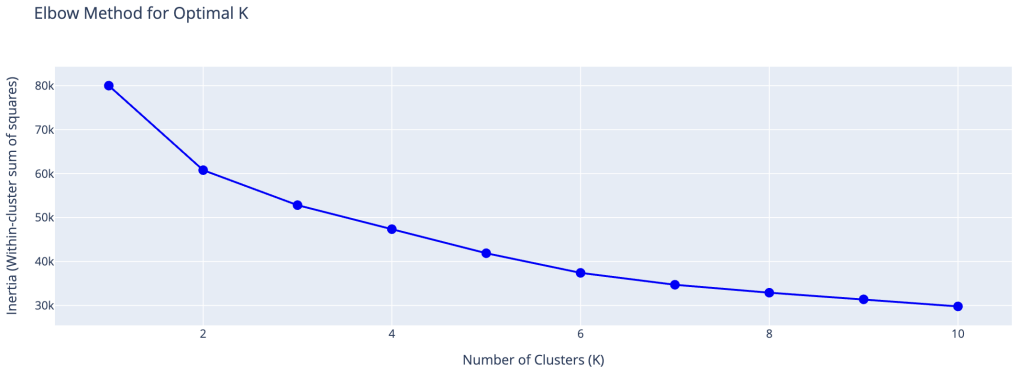

Figure 1: Méthode du coude pour déterminer K optimal

4.1.2. Résultats

Les 4 clusters qui ont émergé avaient du sens musicalement :

- **Cluster 0 - Chansons Chill/Acoustiques** : Musique folk, chansons à la guitare acoustique.
- **Cluster 1 - Énergique/Pop** : Musique pop radio typique.
- **Cluster 2 - Vocal/Mainstream** : La plupart des morceaux populaires tombaient ici.
- **Cluster 3 - Électronique/Instrumental** : EDM et musique électronique.

Fait intéressant, le cluster pop énergique avait la popularité moyenne la plus élevée.

4.1.3. Visualisation avec PCA

On ne peut pas tracer des données à 8 dimensions, donc nous avons utilisé PCA (Analyse en Composantes Principales) pour les réduire à 2 dimensions. PCA trouve les deux directions qui capturent le plus de variation dans les données. Ce n’est pas parfait mais ça permet de voir les clusters sur un graphique 2D.

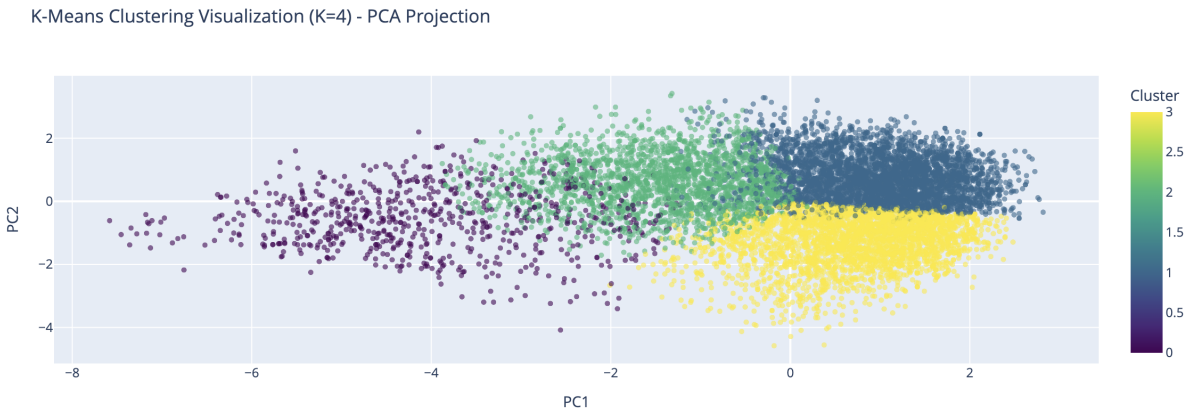

Figure 2: Visualisation K-Means avec PCA

4.2. DBSCAN Clustering

DBSCAN est assez différent de K-Means. Au lieu de forcer chaque chanson dans un des K groupes, DBSCAN cherche des **régions denses** de données et appelle tout le reste "bruit" ou valeurs aberrantes. Pourquoi c'est utile ? K-Means force chaque chanson quelque part, même si elle est complètement bizarre. DBSCAN dit : "non, cette chanson est trop différente de tout le reste, c'est une valeur aberrante".

4.2.1. Paramètres

- `eps=0.5` : Quelle proximité les points doivent avoir pour compter comme voisins
- `min_samples=10` : Combien de voisins vous devez avoir pour être considéré comme un point "core"

4.2.2. Résultats

DBSCAN a identifié 13 clusters distincts dans l'ensemble de données. Le cluster de bruit (noté -1) contient environ 8400 morceaux, représentant les chansons qui ne correspondent à aucun pattern dense. Le cluster 0 est le plus grand parmi les clusters réguliers avec environ 1200 morceaux, tandis que les autres clusters (1 à 12) contiennent des groupes plus petits. Cette distribution révèle que la majorité des chansons tombent dans des patterns réguliers, avec une portion significative étant des chansons expérimentales ou de niche.

La détection de ces valeurs aberrantes est particulièrement utile car elle met en évidence les chansons qui sont vraiment différentes de tout le reste du dataset, permettant une meilleure compréhension de la diversité musicale dans Spotify.

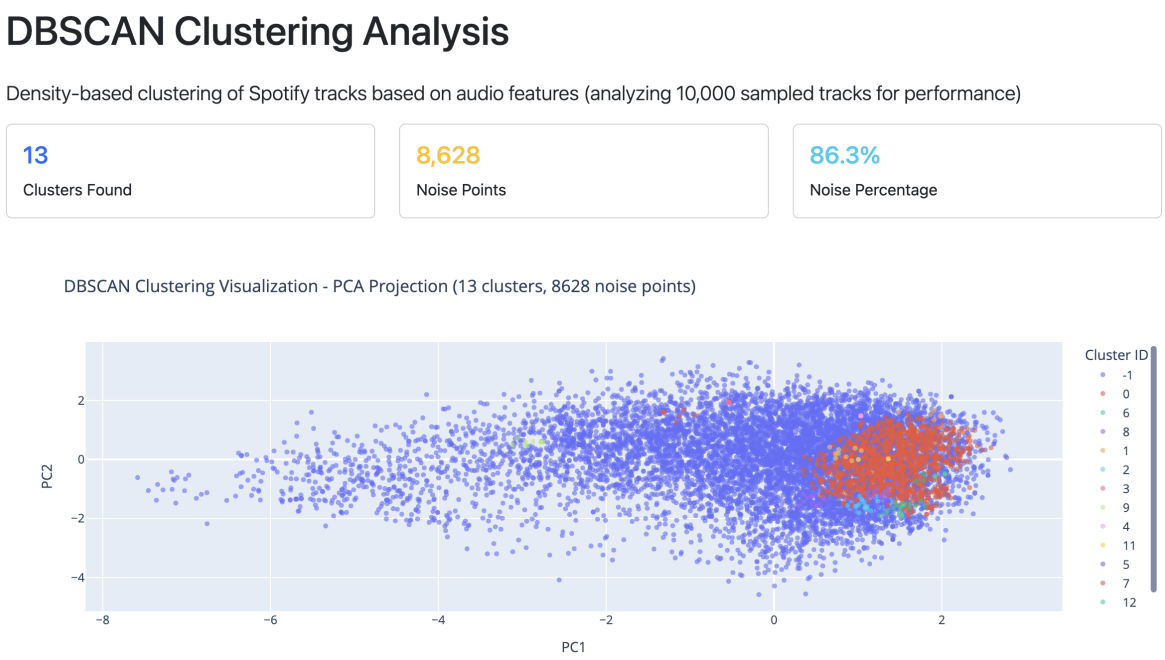

Figure 3: Visualisation DBSCAN avec détection de bruit

5. Analyse de Corrélation

Nous voulions voir quelles caractéristiques audio sont liées les unes aux autres. Les mesures de corrélation vont de -1 (relation négative parfaite) à +1 (relation positive parfaite).

5.1. Corrélations fortes trouvées

- **Énergie et Loudness** ($r \approx 0.76$) : Fait totalement sense - les chansons fortes semblent plus énergiques.

- **Énergie et Acousticité** ($r \approx -0.73$) : Les chansons acoustiques tendent à être moins énergiques. La production électronique augmente l'énergie.

- **Valence et danceability** ($r \approx 0.48$) : Les chansons joyeuses sont plus dansantes.

- **Speechiness et Instrumentalité** (négatif) : Plus de paroles signifie moins de contenu instrumental.

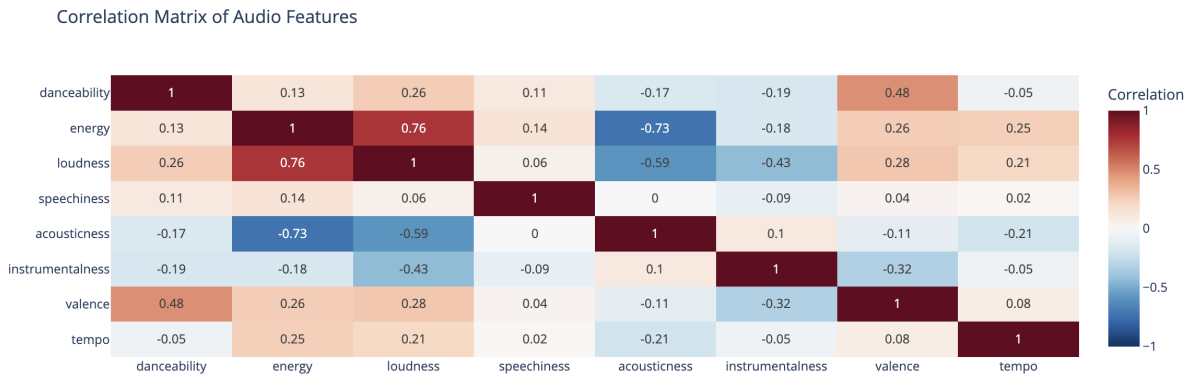

Figure 4: Matrice de corrélation des caractéristiques audio

5.2. Découverte sur la popularité

La popularité n'était pas fortement corrélée avec une seule caractéristique audio (toutes les corrélations en dessous de 0.3). Cela suggère que ce qui rend une chanson populaire n'est pas juste une question de caractéristiques audio - c'est probablement plus une question de marketing, de célébrité de l'artiste, de timing, de médias sociaux, etc.

6. Analyse de Genre

Nous avons regardé quels genres sont les plus populaires.

1. Pop Film
2. Pop
3. Progressive House

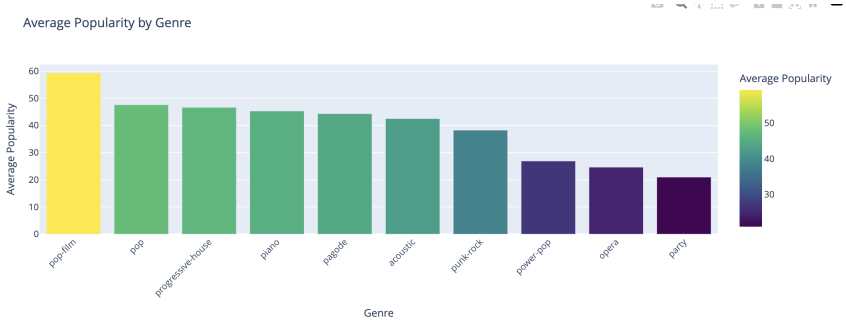

Figure 5: Genres les plus populaires

6.1. Signatures audio par genre

Chaque genre a une “signature” distincte :

- **Pop** : Haute danceability, haute valence (joyeux), énergie haute
- **Rock** : Haute énergie, danceability plus faible
- **Acoustique**: énergie, danceability, valence plus faible

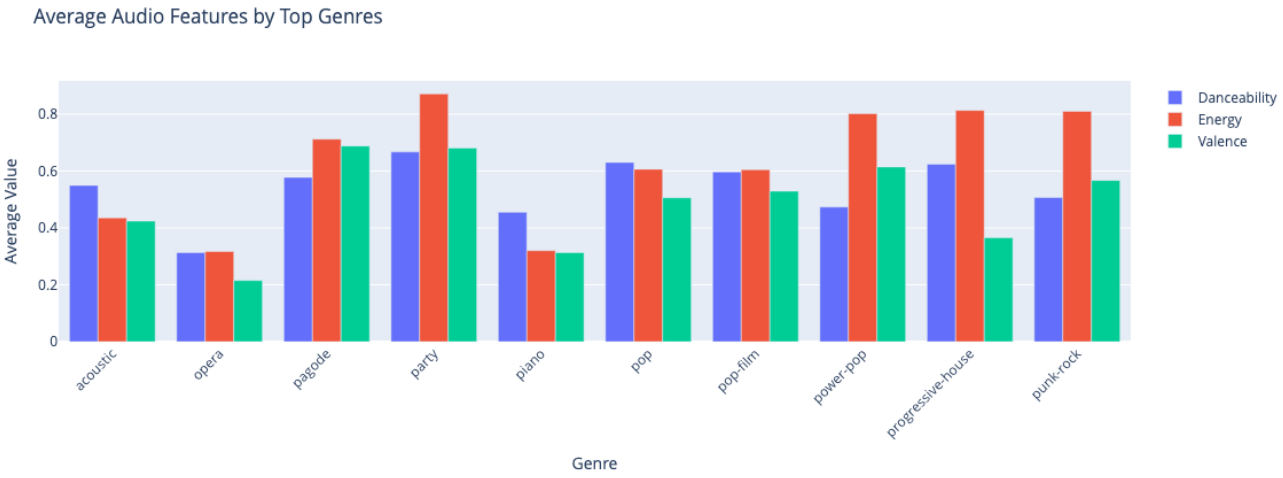

Figure 6: Signatures audio par genre

7. Dashboard Interactif

Nous avons créé un dashboard Dash en Python qui permet d’explorer les données de manière interactive. Le dashboard contient 7 pages différentes :

- Overview : Vue d’ensemble des données
- K-Means : Visualisation du clustering K-Means
- DBSCAN : Visualisation du clustering DBSCAN
- Correlation : Matrice de corrélation interactive
- Genre Analysis : Analyse des genres (plus communs, plus populaires, popularité moyenne)
- Top Charts : Les artistes et morceaux les plus populaires

7.1. Bibliothèques utilisées

Le dashboard utilise les principales bibliothèques suivantes :

- **Dash** : Framework web pour créer l'interface interactive du dashboard
- **Plotly** : Bibliothèque de visualisation pour créer des graphiques interactifs
- **Pandas** : Manipulation et traitement des données, chargement des fichiers CSV
- **NumPy** : Calculs numériques et opérations matricielles
- **Scikit-learn** : Algorithmes de machine learning (clustering K-Means et DBSCAN, PCA, StandardScaler)
- **SciPy** : Calculs scientifiques avancés
- **Statsmodels** : Analyses statistiques

8. Conclusion

8.1. Insights techniques

Ce projet nous a permis de :

- **Nettoyer des données réelles** : Comprendre les problèmes courants quand on travaille avec des données du monde réel
- **Appliquer des algorithmes de clustering** : Comprendre les différences entre K-Means (contraint) et DBSCAN (basé sur la densité)
- **Analyser les corrélations** : Voir quelles caractéristiques audio sont liées les unes aux autres
- **Visualiser les données** : Utiliser PCA pour réduire 8 dimensions à 2 dimensions pour la visualisation
- **Créer une interface interactive** : Construire un dashboard en Dash pour explorer les données

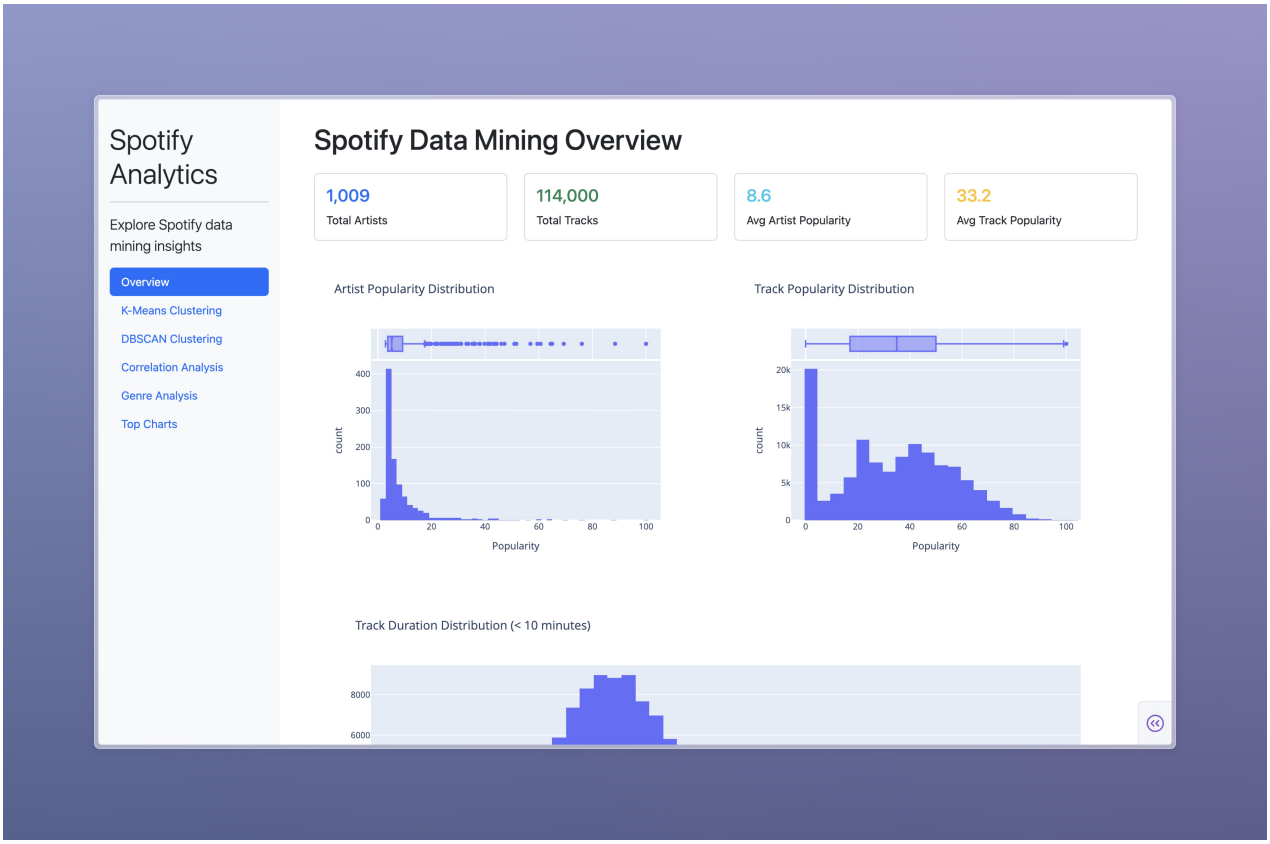

Figure 7: Page d'overview du dashboard

8.2. Insights musicaux

Les résultats sont intéressants du point de vue musical :

- La musique sur Spotify peut être groupée de manière **très significative** basée uniquement sur les caractéristiques audio. Cela confirme que Spotify comprend vraiment les propriétés musicales.
- Différents **genres ont des "signatures" audio distinctes**. La pop n'a pas les mêmes caractéristiques que l'électronique ou le rock.
- La popularité n'est **pas** fortement liée aux caractéristiques audio. Ce qui rend une chanson populaire est probablement plus complexe - marketing, artiste connu, moment du lancement, virality sur les réseaux sociaux.
- DBSCAN trouve environ 20-40% de chansons comme "bruit" - ce sont probablement des chansons très expérimentales ou de niche. C'est intéressant car Spotify a une énorme diversité musicale.

8.3. Applications pratiques

Ces techniques sont utilisées réellement par des services comme Spotify :

- Le clustering aide à créer des playlists auto-générées (“Discover Weekly”)
- L’analyse de corrélation aide Spotify à comprendre quelles caractéristiques rendre recommandables
- L’identification d’outliers (valeurs aberrantes) aide à trouver des chansons vraiment innovantes