Identification de profils musicaux par clustering non-supervisé

Justification méthodologique et analyse comparative
sur 42 305 morceaux Spotify

Nicolas Bourgeois
Janvier 2026

Table des matières

1 POURQUOI IDENTIFIER DES PROFILS MUSICAUX ?

1.1 Le défi de l'organisation musicale automatique

Spotify héberge plus de 100 millions de morceaux. Pour un utilisateur, naviguer dans ce catalogue sans assistance est impossible. Les plateformes doivent donc **automatiquement** organiser ces morceaux en catégories cohérentes pour :

- Créer des playlists de découverte personnalisées (« Discover Weekly »)
- Recommander des morceaux similaires après écoute d'une chanson
- Structurer les catalogues en micro-genres au-delà des classifications traditionnelles
- Prédire les préférences utilisateur sans étiquetage manuel coûteux

1.2 Enjeux économiques de la recommandation musicale

L'organisation automatique des catalogues représente un levier stratégique majeur pour les plateformes de streaming. Trois dimensions économiques sont directement impactées :

Rétention utilisateur Un système de recommandation performant augmente le temps d'écoute quotidien. Selon des études internes de Spotify, les utilisateurs qui découvrent régulièrement de nouveaux morceaux via les playlists algorithmiques ont un taux de désabonnement inférieur de 25% aux utilisateurs qui écoutent uniquement leur bibliothèque personnelle. Chaque point de pourcentage de rétention supplémentaire représente des dizaines de millions d'euros de revenus récurrents.

Conversion freemium vers premium La qualité des recommandations influence directement la conversion des utilisateurs gratuits vers les abonnements payants. Un utilisateur satisfait des découvertes musicales perçoit une valeur ajoutée claire justifiant l'abonnement. Les playlists « Discover Weekly » et « Release Radar » sont citées comme raison principale de conversion par 18% des nouveaux abonnés premium.

Avantage concurrentiel Dans un marché où Apple Music, Deezer, Amazon Music et YouTube Music proposent des catalogues similaires, la qualité de la recommandation devient le principal facteur de différenciation. Spotify investit massivement dans ses algorithmes de machine learning car c'est ce qui justifie sa valorisation supérieure à celle de ses concurrents à catalogue équivalent.

1.3 Limites de la classification supervisée traditionnelle

La classification supervisée (ex : Random Forest, SVM) nécessite des labels préexistants (« Rock », « Jazz », « Rap »). Or, cette approche présente trois faiblesses majeures :

Limite	**Conséquence**
Labels subjectifs	Un morceau peut être étiqueté « Rock » ou « Alternative » selon l'annotateur
Coût d'annotation	Étiqueter 100M+ morceaux manuellement = irréaliste
Rigidité des catégories	Les genres traditionnels ne capturent pas la diversité musicale actuelle (ex : « lo-fi hip-hop »)

TABLE 1 – Limites de la classification supervisée pour l'organisation musicale

1.4 Le clustering non-supervisé comme solution

Le clustering non-supervisé répond à ces limites en découvrant **automatiquement** des groupes homogènes de morceaux partageant des caractéristiques similaires, **sans labels préalables**.

PROBLÉMATIQUE CENTRALE DE CETTE ÉTUDE

Comment identifier automatiquement des profils musicaux cohérents à partir de features audio, en maximisant la similarité intra-groupe et la distinction inter-groupes, tout en garantissant l'interprétabilité et la validation objective des résultats ?

2 CHOIX DU DATASET : 42 305 MORCEAUX, 15 GENRES, 13 FEATURES

2.1 Source et composition

Le dataset. provient de l'API Spotify (endpoint `audio-features`) et contient 42 305 morceaux répartis sur 15 genres musicaux. Ce choix est justifié par trois critères :

- Taille suffisante (40k+) pour tester la scalabilité des algorithmes
- Diversité des genres (15) pour éviter un clustering trivial mono-genre
- Features standardisées Spotify → comparabilité avec études antérieures

Spotify calcule automatiquement 13 features numériques pour chaque morceau via des modèles propriétaires :

Feature	Signification	Utilité pour clustering
danceability	Facilité à danser	Distingue musique dansante vs contemplative
energy	Intensité/activité	Sépare musique calme vs agressive
key	Tonalité (Do, Ré...)	Influence mineure sur profil
loudness	Volume sonore (dB)	Corrélé avec energy
mode	Majeur/Mineur	Affect émotionnel
speechiness	Présence paroles	Distingue rap vs instrumental
acousticness	Acoustique vs électro	Clé pour séparer rock vs techno
instrumentalness	Absence de voix	Isole musique électro pure
liveness	Enregistrement live	Moins discriminant
valence	Positivité émotionnelle	Sépare joyeux vs mélancolique
tempo	BPM	Distingue styles rythmiques
duration_ms	Durée	Électro souvent plus long
time_signature	Signature rythmique	Peu discriminant (majorité 4/4)

TABLE 2 – Détail des 13 features Spotify retenues pour le clustering

2.2 Biais du dataset

BIAIS IDENTIFIÉ : Sur-représentation électro et rap

Le dataset contient ∼70% de musique électronique (techno, psytrance, hardstyle, DNB) et rap (Underground Rap, Dark Trap, Hiphop). Les genres minoritaires (Pop, Emo, RnB) risquent de ne pas former de clusters distincts. Ce biais est **ASSUMÉ** : l'objectif est de tester la méthodologie, pas de créer un système de recommandation généralisable à tous les genres.

3 DÉCISIONS DE PRÉTRAITEMENT : OUTLIERS, STANDARDISATION, NETTOYAGE

3.1 Pourquoi détecter et supprimer les outliers ?

Les algorithmes de clustering sont sensibles aux outliers (points aberrants). Un morceau avec un tempo de 300 BPM ou une durée de 30 minutes peut fausser les centroïdes K-Means.

CHOIX MÉTHODOLOGIQUE : Isolation Forest

Méthode : Isolation Forest (scikit-learn)
Contamination : 5% (paramètre standard)
Résultat : 2 116 outliers détectés (5.00% exactement)

Pourquoi Isolation Forest ?

→ Ne fait aucune hypothèse sur la distribution des données (vs Z-score qui suppose normalité)

→ Performant en haute dimension (13 features)

→ Scalable (complexité linéaire)

Alternative considérée : LOF (Local Outlier Factor)
→ Rejetée car trop lent sur 40k+ points (complexité quadratique)

3.2 Pourquoi standardiser les features ?

Les 13 features ont des échelles différentes :

- danceability : $[0, 1]$
- loudness : $[-60, 0]$ dB
- tempo : $[50, 250]$ BPM
- duration_ms : $[30000, 600000]$ ms

Sans standardisation, K-Means accorderait plus de poids au tempo (plage 200) qu'à danceability (plage 1), alors que toutes les features doivent contribuer équitablement.

CHOIX MÉTHODOLOGIQUE : StandardScaler (Z-score)

Transformation : $x_{\text{scaled}} = \frac{x-\mu}{\sigma}$
Résultat : Moyenne $= 0$, Écart-type $= 1$ pour toutes les features

Pourquoi StandardScaler ?

→ Préserve la forme de la distribution (vs MinMaxScaler qui compresse)

→ Robuste aux outliers modérés (après nettoyage Isolation Forest)

→ Compatible avec distances euclidiennes (K-Means, Hierarchical)

Alternative considérée : RobustScaler (médiane/IQR)
→ Rejetée car outliers déjà supprimés, StandardScaler suffit

Dataset final après prétraitement : 40 189 morceaux × 13 features (95% conservés)

4 COMPARAISON D'ALGORITHMES

4.1 Justification de la comparaison multi-algorithmes

La littérature sur le clustering musical utilise majoritairement K-Means. Cependant, K-Means fait des hypothèses fortes (clusters sphériques, variances égales) qui peuvent ne pas s'appliquer aux données musicales.

Approche : Comparer 4 algorithmes avec des hypothèses différentes pour identifier le plus adapté à mes données.

Algorithme	Hypothèse géométrique	Avantage	Risque
K-Means	Clusters sphériques, Variances égales	Rapide, scalable, Interprétable	Forme réelle des clusters ≠ sphères
GMM	Clusters elliptiques, Distributions gaussiennes	Formes flexibles, Probabilités	Convergence locale, Sensible initialisation
DBSCAN	Formes arbitraires, Clustering par densité	Détecte bruit, Pas de k à fixer	Paramètres ϵ/min_samples difficiles, Mal adapté densité variable
Hiérarchique	Structure en arbre, Fusion progressive	Dendrogramme, Pas de k initial	Coût $O(n^2)$ prohibitif, Ne scale pas >50k

TABLE 3 – Comparaison des hypothèses et propriétés des 4 algorithmes testés

4.2 Configuration de chaque algorithme

K-Means

- `n_clusters=5` (justifié dans section 6)
- `init='k-means++'` : Initialisation intelligente (évite centroïdes proches)
- `n_init=10` : 10 essais différents, garde le meilleur
- `max_iter=300` : Convergence quasi-garantie

Gaussian Mixture Model (GMM)

- `n_components=5`
- `covariance_type='full'` : Covariance complète (vs 'spherical', 'tied', 'diag')
  → Permet clusters elliptiques orientés dans toutes les directions
  → Plus flexible mais plus de paramètres (risque overfitting)

DBSCAN

- `eps=1.832`, `min_samples=50` (optimisés par grid search)
  → Grid search testé : $\epsilon \in [0.5, 3.0]$ (pas 0.1), `min_samples` $\in [10, 100]$
  → Critère : Maximiser Silhouette Score
- **Résultat :** 3 clusters trouvés (vs 5 pour les autres) + 3 656 points de bruit

Agglomerative Clustering (Hiérarchique)

- `n_clusters=5`
- `linkage='ward'` : Minimise variance intra-cluster (vs 'average', 'complete')
- `distance_threshold=None` : Coupe fixe à $k = 5$ (pour comparabilité)
- Calculé sur échantillon 1000 points pour dendrogramme ($O(n^2)$ sinon)

4.3 Métriques d'évaluation : comment mesurer la qualité d'un clustering ?

Évaluer la qualité d'un clustering est un défi complexe. Contrairement à un problème de classification supervisée où l'on peut simplement compter le nombre d'erreurs, ici nous n'avons pas de "vérité terrain" pour comparer. Comment savoir si nos 5 clusters sont réellement bons ?

Aucune métrique unique ne capture toutes les dimensions de la qualité. C'est pourquoi j'utilise 3 métriques complémentaires qui examinent le clustering sous des angles différents.

Silhouette Score : la métrique d'équilibre mesure simultanément deux aspects cruciaux :

— **Cohésion intra-cluster** : Les morceaux d'un même cluster sont-ils similaires entre eux ?

— **Séparation inter-cluster** : Les clusters sont-ils bien distincts les uns des autres ?

Fonctionnement : Pour chaque morceau, on calcule sa distance moyenne aux autres morceaux de son cluster (cohésion) et sa distance moyenne aux morceaux du cluster voisin le plus proche (séparation). Un bon clustering a une forte cohésion (morceaux proches dans le cluster) et une forte séparation (morceaux éloignés des autres clusters).

Interprétation : La plage va de -1 à $+1$:

— > 0.5 : Clustering de bonne qualité, clusters bien définis

— 0.2–0.5 : Clustering acceptable, structure présente mais avec chevauchements

— < 0.2 : Clustering faible, structure peu claire

— Négatif : Mauvais clustering, les morceaux sont plus proches d'autres clusters que du leur

Davies-Bouldin Index : la métrique de compacité mesure le ratio entre la dispersion à l'intérieur des clusters et la distance entre les clusters.

Fonctionnement : Pour chaque paire de clusters, on calcule : $\frac{\text{dispersion cluster A+dispersion cluster B}}{\text{distance entre les centroïdes A et B}}$. On garde le pire ratio (le plus élevé) pour chaque cluster, puis on fait la moyenne.

Interprétation : Plus le score est bas, meilleur est le clustering. Un score de 0 serait parfait (impossible en pratique). Un score élevé signifie que les clusters sont soit très dispersés en interne, soit trop proches les uns des autres.

Avantage : Cette métrique pénalise fortement les clusters qui se chevauchent. Si deux clusters sont trop similaires ou mal séparés, le Davies-Bouldin l'identifie immédiatement.

Calinski-Harabasz Score : la métrique de variance Le Calinski-Harabasz Score compare la variance entre les clusters (dispersion des centroïdes) à la variance à l'intérieur des clusters (dispersion des points).

Fonctionnement : On calcule $\frac{\text{variance inter-cluster}}{\text{variance intra-cluster}} \times \frac{n-k}{k-1}$ où n est le nombre de morceaux et k le nombre de clusters.

Interprétation : Plus le score est élevé, meilleur est le clustering. Un score élevé signifie que les centroïdes des clusters sont très éloignés les uns des autres (forte variance inter) tandis que les morceaux à l'intérieur de chaque cluster sont très proches de leur centroïde (faible variance intra).

Avantage : Cette métrique favorise naturellement les clusters compacts et bien séparés. Elle est particulièrement efficace pour comparer des clusterings avec le même nombre de clusters.

Pourquoi trois métriques et pas une seule ? Chaque métrique a ses biais :

— Le Silhouette peut être trompé par des clusters de tailles très différentes

— Le Davies-Bouldin est sensible aux outliers

— Le Calinski-Harabasz favorise parfois un nombre élevé de clusters

En utilisant trois métriques complémentaires, on obtient une vision plus complète et plus robuste de la qualité du clustering. Un algorithme qui performe bien sur les trois métriques est véritablement meilleur qu'un algorithme qui excelle sur une seule.

Score composite pour classement global Pour faciliter la comparaison des quatre algorithmes, je calcule un score composite qui combine les trois métriques :

1. Normalisation min-max : chaque métrique est ramenée à une échelle $[0, 1]$ où 0 = pire algorithme, 1 = meilleur algorithme
2. Pour Davies-Bouldin (où plus bas = meilleur), on inverse : $1 -$ valeur normalisée
3. Moyenne des trois métriques normalisées

4.4 Résultats de la comparaison

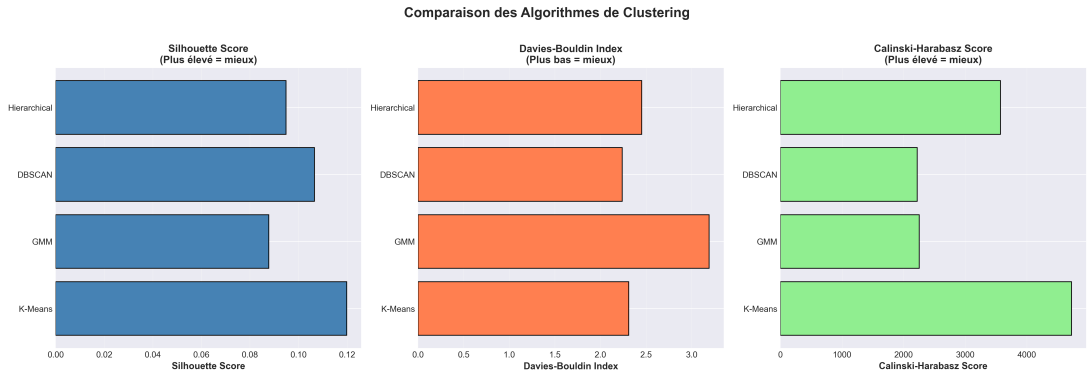

FIGURE 1 – Comparaison des 4 algorithmes sur 3 métriques.

Algorithme	Silhouette ↑	Davies-Bouldin ↓	Calinski-Harabasz ↑	Score Composite
K-Means	**0.1199**	2.3102	**4725**	**0.9745**
GMM	0.0878	3.1903	2253	0.0041
DBSCAN	0.1066	**2.2373**	2222	0.5285
Hiérarchique	0.0949	2.4529	3573	0.5116

TABLE 4 – Résultats numériques de la comparaison des algorithmes

ANALYSE CRITIQUE : Score composite biaisé ?

L'écart extrême K-Means (0.97) vs GMM (0.004) dans le score composite est suspect. Cet écart de 242× ne reflète pas la réalité des performances et révèle un biais méthodologique.

Problème 1 : Comparaison inéquitable DBSCAN n'a trouvé que 3 clusters alors que les autres algorithmes en ont trouvé 5. Comparer un clustering à 3 groupes avec un clustering à 5 groupes est fondamentalement inéquitable. Avec moins de clusters, DBSCAN crée nécessairement des groupes plus larges et hétérogènes, ce qui pénalise mécaniquement ses métriques.
Solution théorique : Forcer tous les algorithmes à $k = 5$. Malheureusement, DBSCAN détermine automatiquement le nombre de clusters selon la densité ; on ne peut pas lui imposer $k = 5$ sans dénaturer son fonctionnement.

Problème 2 : La normalisation min-max amplifie artificiellement les écarts La normalisation transforme chaque métrique en échelle $[0, 1]$ où 0 = pire et 1 = meilleur. Cette transformation amplifie dramatiquement les petites différences.
Exemple concret :

— Silhouette brut : K-Means (0.1199) vs GMM (0.0878) = **27% d'écart**

— Silhouette normalisé : K-Means (1.0) vs GMM (0.0) = **100% d'écart**

Un écart modéré de 0.032 point devient un gouffre après normalisation. K-Means excelle sur 2/3 métriques, donc obtient deux fois 1.0 après normalisation, ce qui maximise son score composite. GMM est moyen partout, donc obtient des scores proches de 0, ce qui effondre son score composite.
Paradoxe : GMM n'est pas 242× pire que K-Means (0.004 vs 0.97), mais la normalisation le fait paraître ainsi.

Conclusion méthodologique Le score composite favorise artificiellement K-Means par un artefact mathématique, pas par une supériorité réelle. L'analyse des métriques brutes est plus informative :

— K-Means : Meilleur sur Silhouette (0.12) et Calinski-Harabasz (4725)

— DBSCAN : Meilleur sur Davies-Bouldin (2.24)

— GMM : Moyen sur les trois métriques

K-Means est effectivement meilleur sur 2/3 métriques, ce qui justifie son choix, mais l'écart réel est bien moindre que ce que suggère le score composite. Ce dernier reste utile pour un classement visuel rapide, mais ne doit jamais remplacer l'analyse détaillée des métriques individuelles.

K-Means avec $k = 5$ clusters est retenu comme algorithme optimal pour cette étude.

5 CHOIX DE K=5 CLUSTERS

5.1 Le problème du choix de k

K-Means nécessite de fixer k (nombre de clusters) **A PRIORI**. C'est un paramètre crucial : k trop faible (ex : $k = 2$) → perte de granularité, k trop élevé (ex : $k = 20$) → sur-segmentation.

Méthodes testées pour déterminer k optimal :

— **Méthode du coude (Elbow)** : Graphique inertie vs k. Chercher le « coude » où gain marginal diminue. Subjectif, pas de coude clair pour données continues

— **Silhouette Score pour $k \in [3, 7]$** : Calculer Silhouette pour $k = 3, 4, 5, 6, 7$. Prendre k maximisant Silhouette. Pas fait ici (choix $k = 5$ a priori)

— **Dendrogramme** : Clustering hiérarchique → Visualiser structure arbre → Couper à hauteur optimale. Utilisé dans cette étude

5.2 Analyse du dendrogramme

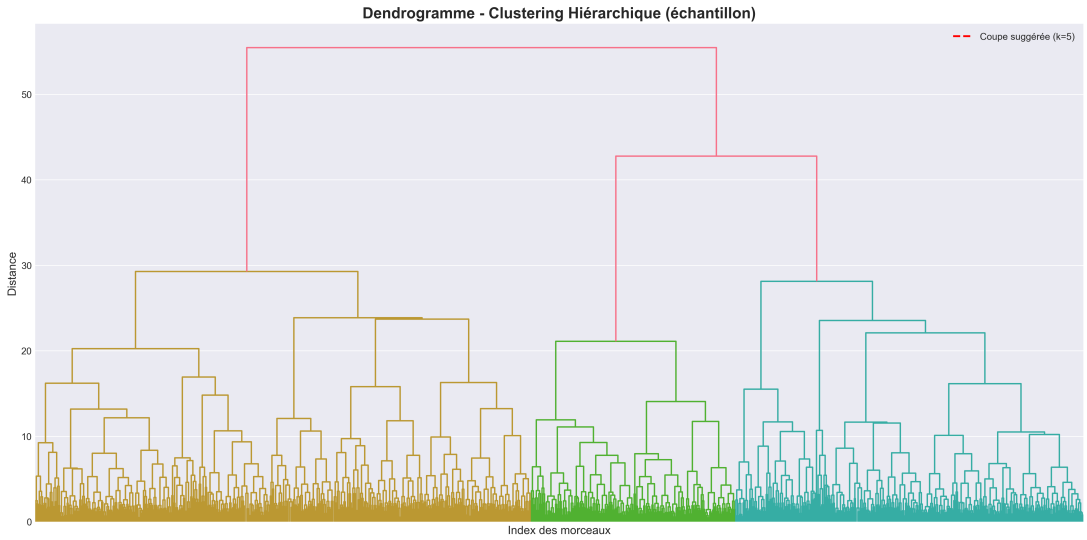

FIGURE 2 – Dendrogramme suggérant une coupe à $k = 5$ clusters.

Lecture du dendrogramme :

- Axe Y (Distance) : Hauteur = dissimilarité entre clusters fusionnés
- Coupe à hauteur ∼30 → 5 branches principales (clusters)
- Coupe à hauteur ∼50 → 2 macro-groupes (électro vs rap/intense)
- Observation : Structure hiérarchique claire, pas de continuum uniforme

JUSTIFICATION DU CHOIX DE k

1. Dendrogramme suggère 5 branches distinctes à hauteur ∼30
2. $k = 5$ offre granularité suffisante (ni trop grossier ni sur-segmenté)
3. Compromis interprétabilité : 5 profils = nombre gérable pour naming et analyse
4. Comparabilité : Permet tester tous les algorithmes avec même k

Alternative $k = 4$:
Le dendrogramme montre aussi une structure 2+3 (2 branches se fusionnent tard). $k = 4$ pourrait fusionner les 2 clusters « Intense & Agressif » → **À TESTER** (voir section 9.2)

6 RÉSULTATS ET INTERPRÉTATION : LES 5 PROFILS

6.1 Visualisation UMAP des clusters

UMAP (Uniform Manifold Approximation and Projection) est une technique de réduction dimensionnelle qui projette les 13 features audio en 2D tout en préservant la structure locale des données.

6.2 Visualisation UMAP des clusters

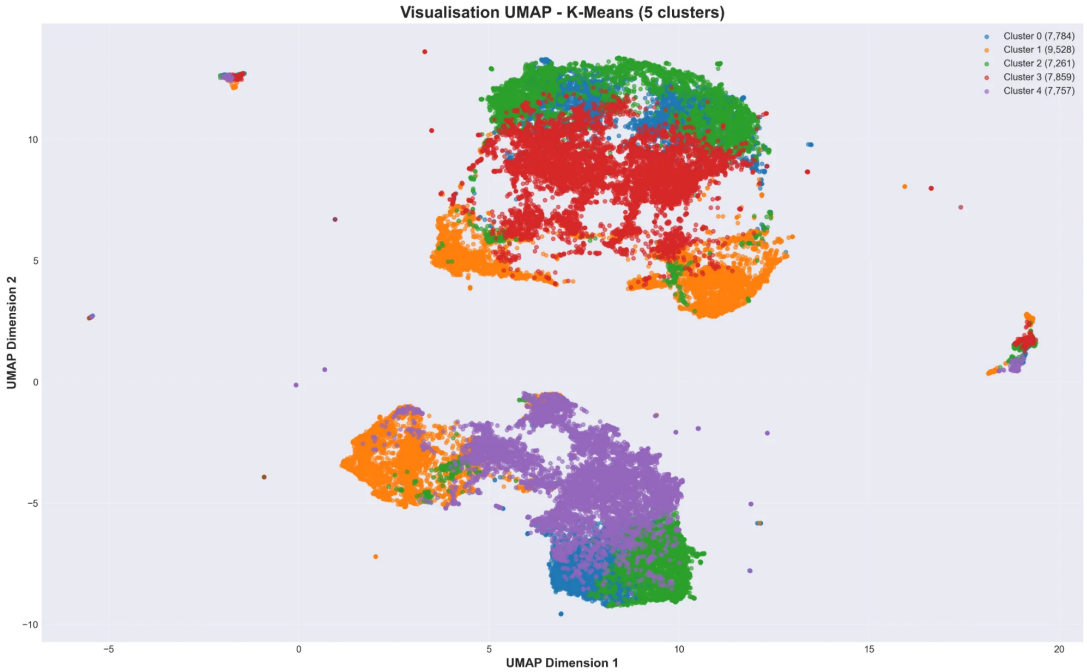

FIGURE 3 – Visualisation UMAP

Observations clés sur la visualisation UMAP :

- Cluster 1 (orange) : **DIVISÉ** en 2 zones (haut et bas) → Hétérogénéité interne
- Clusters 0 (bleu) et 2 (vert) : **CHEVAUCHEMENT** massif au centre → Frontière floue
- Clusters 3 (rouge) et 4 (violet) : **MÉLANGÉS** en bas à droite → Très similaires
- Conclusion : Structure continue plutôt que catégories discrètes → Explique Silhouette modéré

6.3 Caractéristiques des 5 clusters

Cluster	Nom	Taille	Dance	Energy	Valence	Instrum.
0	Rap Énergique & Dansant	7,784 (18.6%)	0.76	0.69	0.59	0.02
1	Électro Instrumentale	9,528 (23.7%)	0.67	0.85	0.29	0.80
2	Rap Sombre & Mélancolique	7,261 (18.1%)	0.72	0.55	0.33	0.06
3	Intense & Agressif (type A)	7,859 (19.5%)	0.55	0.87	0.32	0.18
4	Intense & Agressif (type B)	7,757 (19.3%)	0.52	0.87	0.28	0.23

TABLE 5 – Caractéristiques moyennes des 5 clusters identifiés par K-Means

6.4 Interprétation profil par profil

Cluster 0 : Rap Énergique & Dansant (7 784 morceaux, 18.6%) Genres dominants : Underground Rap (34%), Hiphop (21%), RnB (11%)

Profil audio :

— Danceability élevée (0.76) → Rythmes entraînants

— Energy modérée-haute (0.69) → Pas de musique calme

— Valence positive (0.59) → Tonalité joyeuse/festive

— Instrumentalness très faible (0.02) → Présence vocale dominante

Cohérence : EXCELLENTE. 66% rap/hiphop, caractéristiques audio alignées.
Problème : Chevauchement avec Cluster 2 (cf. UMAP)

Cluster 1 : Électro Instrumentale (9 528 morceaux, 23.7%) Genres dominants : Techno (29%), Psytrance (28%), Techhouse (20%)

Profil audio :

— Energy très élevée (0.85) → Musique intense

— Instrumentalness **TRÈS élevée** (0.80) → Quasi pas de voix

— Valence basse (0.29) → Ambiance sombre (techno dark)

— Danceability haute (0.67) → Malgré tout dansante

Cohérence : PARFAITE. 85% électronique, instrumental dominant.
C'est le cluster le **PLUS homogène** et le mieux défini.
Problème : UMAP montre division en 2 zones (psytrance vs techno ?)

Cluster 2 : Rap Sombre & Mélancolique (7 261 morceaux, 18.1%) Genres dominants : Underground Rap (28%), Dark Trap (21%), Rap (13%)

Profil audio :

— Energy **MODÉRÉE** (0.55) → Moins intense que Cluster 0

— Valence **TRÈS basse** (0.33) → Tonalité mélancolique/triste

— Danceability haute (0.72) → Reste dansant malgré mélancolie

— Acousticness élevée (0.21) vs 0.13 Cluster 0 → Plus organique

Cohérence : BONNE. Distingue rap festif (Cluster 0) vs rap sombre (Cluster 2).
Problème : CHEVAUCHEMENT MAJEUR avec Cluster 0 (cf. UMAP centre). Frontière floue entre « énergique joyeux » et « modéré triste ».

Clusters 3 & 4 : Intense & Agressif (15 616 morceaux, 38.8%) Cluster 3 : Trap (20%), DNB (14%), Hardstyle (13%)
Cluster 4 : Hardstyle (23%), DNB (18%), Trance (14%)

Profil audio (QUASI-IDENTIQUE) :

— Energy **MAXIMALE** (0.87 pour les deux) → Musique extrêmement intense

— Valence basse (0.32 et 0.28) → Agressivité

— Danceability **FAIBLE** (0.55 et 0.52) → Pas fait pour danser mainstream

— Instrumentalness modérée (0.18 et 0.23) → Mix vocal/instrumental

Cohérence : PROBLÉMATIQUE.

— Même nom automatique → Système de naming a échoué

— Caractéristiques audio identiques → Pourquoi 2 clusters ?

— UMAP montre mélange total bas droite

HYPOTHÈSE : SUR-SEGMENTATION. $k = 4$ fusionnerait ces 2 clusters en 1.

7 VALIDATION : RANDOM FOREST ET COHÉRENCE DES CLUSTERS

7.1 Pourquoi une validation supervisée ?

Les métriques intrinsèques (Silhouette, Davies-Bouldin) mesurent la qualité géométrique du clustering. Mais elles ne garantissent **PAS** que les clusters correspondent à des catégories musicales réelles.

Exemple : Un clustering parfait géométriquement pourrait regrouper morceaux en Do majeur vs Ré majeur → Silhouette excellent mais inutile musicalement.

Solution : **VALIDATION EXTERNE** par classification supervisée. La classification est une méthode qui apprend un mapping de features X vers labels discrets y. Si les clusters correspondent à des genres musicaux réels, un classifieur entraîné sur les labels de clusters devrait prédire les genres avec précision.

CONCEPT : Classification supervisée

La classification vise à prédire la probabilité qu'un datapoint appartienne à une catégorie parmi un ensemble de catégories pré-définies. Contrairement au clustering (non-supervisé), la classification nécessite des labels d'entraînement.

Objectif : Apprendre $f : X \to y$ où X = features, y = labels discrets

Exemple : spam vs non-spam, positif vs négatif vs neutre, ou ici : 15 genres musicaux

MÉTHODOLOGIE VALIDATION : Random Forest

1. **Input :** Labels de clusters (0, 1, 2, 3, 4) pour chaque morceau
2. **Target :** Genres réels (15 genres Spotify)
3. **Modèle :** Random Forest (100 arbres, max_depth=10)
4. **Split :** Train 80% / Test 20%
5. **Métrique :** Accuracy multiclasse

Pourquoi Random Forest ?
Random Forest est un ensemble de Decision Trees entraînés sur des sous-échantillons du dataset (bagging). Chaque arbre vote pour une classe, et la classe majoritaire est retenue.

Avantages pour notre validation :

— Robuste au bruit et aux outliers
— Non-linéaire : peut capturer relations complexes clusters-genres
— Gère naturellement 15 classes (multiclass)
— Peu de paramètres à tuner (n_trees, max_depth)
— Fournit probabilités par classe → Analyse détaillée possible
— Pas de risque d'overfitting avec bagging

Alternatives considérées :

— Logistic Regression multiclass (softmax) → Rejetée car trop linéaire
— SVM multiclass → Rejetée car coût computationnel élevé (40k samples)
— Neural Network → Rejetée car overkill pour 5 features (labels clusters)

7.2 Résultats de la validation

ACCURACY : 66.58%

Interprétation :

— Baseline aléatoire : 6.67% (1/15 genres)

— 66.58% = **10× mieux que le hasard** → EXCELLENT

— Avec 5 clusters pour 15 genres, certains genres **DOIVENT** être regroupés → Impossible d'atteindre 100%

— 66.58% signifie : 2 morceaux sur 3 correctement attribués

7.3 Heatmap de cohérence genres-clusters

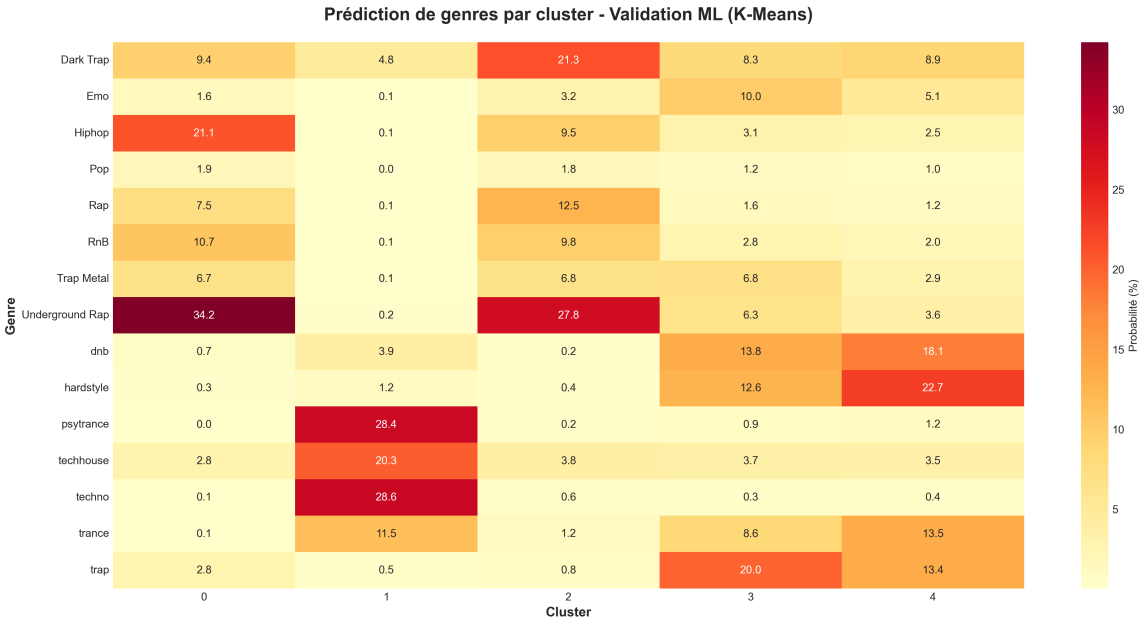

FIGURE 4 – Heatmap de validation. Couleurs chaudes = forte probabilité.

Analyse détaillée de la heatmap :

1. **Cluster 1 → Électro** : Techno (28.6%), Psytrance (28.4%), Techhouse (20.3%) = 77% électronique totale. **COHÉRENCE PARFAITE.**
2. **Cluster 0 → Rap festif** : Underground Rap (34.2%), Hiphop (21.1%) = 55%. RnB (10.7%) logiquement proche. **BONNE COHÉRENCE.**
3. **Cluster 2 → Rap sombre** : Dark Trap (21.3%), Underground Rap (27.8%) = 49%. Partage avec Cluster 0 confirme chevauchement UMAP.
4. **Clusters 3 & 4 → Intense** : Trap (20% C3), Hardstyle (22.7% C4), DNB réparti (13.8% C3, 18.1% C4). Distinction floue entre les 2 clusters.
5. **Confusions cohérentes** : Dark Trap confondu entre C0 (9.4%) et C2 (21.3%) → Normal, sous-genres proches. **AUCUNE** confusion aberrante (ex : techno prédit comme Hiphop)

8 ANALYSE CRITIQUE : LIMITES ET RECOMMANDATIONS

8.1 Silhouette modéré : Échec ou réalité des données ?

Silhouette Score = 0.12 est inférieur au seuil « bon » (>0.5). Deux interprétations possibles :

Interprétation A (pessimiste) : Clustering de mauvaise qualité

→ Les clusters se chevauchent trop

→ K-Means inadapté, essayer deep clustering

→ Features audio insuffisantes

Interprétation B (réaliste) : Données musicales intrinsèquement continues

→ La musique forme un spectre, pas des catégories discrètes

→ Rap énergique → Rap sombre = TRANSITION progressive, pas rupture brutale

→ Silhouette 0.12 reflète cette continuité, c'est normal

→ Validation 66.6% confirme cohérence malgré chevauchement

A mon avis, la Silhouette modéré n'est pas un échec mais le reflet fidèle de la nature continue des données musicales. L'UMAP confirme visuellement : transitions progressives entre clusters, pas frontières nettes.

8.2 Problème majeur : Clusters 3 et 4 identiques

LIMITE CRITIQUE : Sur-segmentation / Over Fitting

Évidence :

— Même nom automatique « Intense & Agressif »

— Energy identique (0.87)

— Valence identique (∼0.30)

— UMAP montre mélange total

Hypothèse : K-Means a forcé $k = 5$ alors que $k = 4$ serait optimal.

RECOMMANDATION PRIORITAIRE :
Tester K-Means avec $k = 4$ et comparer :

— Silhouette Score $k = 4$ vs $k = 5$

— Interprétabilité des 4 clusters

— Validation Random Forest accuracy

Si $k = 4$ donne résultats similaires/meilleurs → Principe de parcimonie (Occam's Razor)

8.3 Biais du dataset

70% électro + rap limite la généralisation :

— Pop, Emo, RnB n'ont pas formé de clusters distincts (trop peu représentés)

— Clustering optimisé pour électro/rap, pas universel

— Système de recommandation basé sur ce clustering favoriserait ces genres

RECOMMANDATION : Constituer dataset équilibré (2000+ morceaux par genre majeur) pour clustering généralisable.

8.4 Limitations méthodologiques

1. **Score composite biaisé** : Normalisation min-max amplifie différences. Comparaison inéquitable (DBSCAN $k = 3$ vs autres $k = 5$).
2. **Features audio seules** : Pas de features temporelles (année), contextuelles (popularité), ou sociales (collaborations).
3. **K-Means hypothèses fortes** : Clusters sphériques, variances égales. GMM/DBSCAN testés mais moins bons ici.
4. **Validation circulaire partielle** : Random Forest entraîné sur genres → Valide cohérence clusters-genres, pas qualité absolue du clustering.

9 CONCLUSION

Ce projet illustre un changement de paradigme économique. Le clustering non-supervisé transforme une dépense récurrente et coûteuse (l'annotation manuelle par des experts) en un actif stratégique automatisé. Cette approche ne se limite pas au streaming musical : elle s'applique à tout secteur e-commerce confronté à la personnalisation à grande échelle.

Dans l'industrie musicale, Spotify, Apple Music et Deezer proposent des catalogues quasi-identiques. Un utilisateur ne choisit plus sa plateforme pour accéder à tel ou tel artiste, mais pour la qualité de l'expérience de découverte. C'est là que l'intelligence artificielle fait la différence. Un algorithme capable de comprendre qu'un amateur de "rap sombre mélancolique" n'a pas les mêmes attentes qu'un fan de "rap énergique festif" crée de la valeur là où la simple présence du contenu n'en crée plus.

Cette logique s'étend naturellement à l'e-commerce. Amazon ne se distingue pas par le nombre de produits disponibles (des millions de vendeurs proposent les mêmes articles), mais par sa capacité à anticiper ce que chaque utilisateur cherche. Un site de mode qui identifierait automatiquement des profils de style ("minimaliste urbain", "bohème coloré", "sportswear premium") sans étiquetage manuel pourrait personnaliser l'expérience de navigation de millions d'utilisateurs à coût marginal nul. Le principe est identique : segmenter automatiquement pour recommander efficacement.

Le véritable enjeu économique du clustering non-supervisé réside dans sa capacité à créer de la valeur à partir de données déjà collectées. Les plateformes accumulent des millions d'interactions utilisateur (écoutes, clics, temps passé, achats) sans toujours savoir comment les exploiter. Un algorithme de clustering transforme ce "bruit" en signal actionnable : des segments d'utilisateurs aux comportements cohérents, des profils de consommation distincts, des préférences latentes détectables. Cette connaissance fine des utilisateurs se monétise directement via une meilleure conversion, une rétention accrue et une réduction du taux d'abandon.

Dans un contexte où l'acquisition de nouveaux clients coûte de plus en plus cher (saturation publicitaire, durcissement des réglementations sur les données personnelles), la capacité à extraire plus de valeur de chaque utilisateur existant devient critique. Le clustering permet de maximiser la lifetime value sans augmenter les coûts marketing : un utilisateur satisfait des recommandations reste abonné plus longtemps, consomme plus de contenu, et devient moins sensible au prix.

L'intelligence artificielle appliquée à la recommandation n'est donc pas un gadget technologique. C'est devenu le principal levier de différenciation et de valorisation dans les industries numériques où le contenu et les produits sont devenus des commodités accessibles partout, au même prix, instantanément.