Data Mining

Project

Master 2 Data Science

Auteurs	: BORIE Clarence – SOUIBKI Heidi - SAKINI Oussama
Date	: 14/11/2024

A. Table des matières

B. Table des Figures 3
C. Table des Tableaux 4
D. Introduction 5
E. Description du Projet 6
F. Méthodologie 7
1. Dataset 7
2. Design du modèle 7
3.1. Prepocessing 7
3.2. Réduction de dimensionnalité 8
3.3. Clustering 10
3.4. Régression 10
3.5. Collaborative filtering 10
3.6. Recommandation 11
4. Implémentation 11
G. Résultats 12
1. Réduction de dimensionnalité 12
2. Clustering 12
3. Régression 13
4. Collaborative filtering 13
H. Conclusion et Perspectives 15
I. Références 16
J. Appendice 17

B. Table des Figures

Figure 1: Méthodologie employée pour la recommandation musicale 6
Figure 2: Graphique représentant la projection des données lors d'un PCA [3] 9
Figure 3: Singular Value Decomposition Matrices [4] 9
Figure 4: Méthode du coude pour déterminer le nombre de cluster pour le K-Means 12
Figure 5: Score de silhouette du K-Means 12
Figure 6: Visualisation des clusters K-Means après PCA 13
Figure 7: Output des scores de similarité les plus élevés pour 5 morceaux 14
Figure 8: Output des associations des indices correspondants aux morceaux 14
Figure 9: Liste finale des recommandations musicales 14

C. Table des Tableaux

Tableau 1: Variables utilisées pour le clustering et la régression 7
Tableau 2: Comparaison des modèles de clustering 10
Tableau 3: Comparaison des modèles de régression 10
Tableau 4: Variables présentent dans le dataset 17

D. Introduction

Ce projet s'inscrit dans le cadre du module de Data Mining du master 2 en Data Sciences de L'université Lyon 1. L'objectif principal est d'identifier des morceaux susceptibles de correspondre aux goûts d'un utilisateur, en utilisant les différentes informations que nous pourrions avoir sur les morceaux qu'il aurait pu apprécier et en exploitant les caractéristiques audio propres à chacun de ces morceaux. Il aborde différentes méthodes de Data Mining vues en cours utilisées pour la recommandation musicale, notamment des méthodes de clustering, de régressions et de filtrage collaboratifs.

Ce rapport est structuré en plusieurs sections. Il débute par une introduction pour présenter le contexte et les objectifs du projet. Une description du projet suit, donnant un aperçu du système développé. La section méthodologie détaille les étapes clés, de la gestion des données au design du modèle et à son implémentation, en expliquant les techniques utilisées, comme le clustering, la régression ou le filtrage collaboratif. Les résultats obtenus sont ensuite analysés, avant une conclusion abordant les perspectives d'amélioration.

E. Description du Projet

Ce projet met en lumière différentes approches permettant d'aboutir à de la recommandation musicale comme décrit dans la Figure 1. Pour ce faire, il est nécessaire de choisir un dataset riche, avec ici environ 30 000 morceaux différents, et des variables pertinentes qui permettront de définir, puis de classer chaque morceaux en fonction de ceux-ci.

Une fois le dataset sélectionné, une étape de préparation des données doit avoir lieu. À l'issue de celle-ci, nous obtiendrons des données nettoyées et standardisées. Cette étape est primordiale et ses résultats doivent être adaptés aux algorithmes de recommandations que nous utiliserons.
Afin de créer le modèle de recommandation, deux approches différentes mais complémentaires ont été étudiées, à savoir : le clustering ou la regression et le collaborative filtering.

Pour la partie clustering, trois modèles différents ont été utilisés et comparés, pour chacun desquels ont été appliquées en amont deux méthodes de simplification, facilitant le traitement, le stockage, l'interprétation, et la cohérence du résultat du clustering.

Pour la régression, les mêmes méthodes de simplifications ont été appliquées à deux modèles de régression différents.

Pour le collaborative filtering il faut faire appel à une matrice d'interactions puis de similarité, afin de faire des recommandations pertinentes basées sur les musiques ayant les variables les plus similaires à la musique appréciée.

Enfin, pour recommander à l'utilisateur des morceaux similaires à ceux qu'il aime, le collaborative filtering va être combiné au clustering ou à la regression. En effet les morceaux qui ressortiront du collaborative filtering seront ensuite localisés dans les clusters ou classés dans une liste, afin de recommander de nouveaux morceaux à l'utilisateur en élargissant et affinant la recommandation.

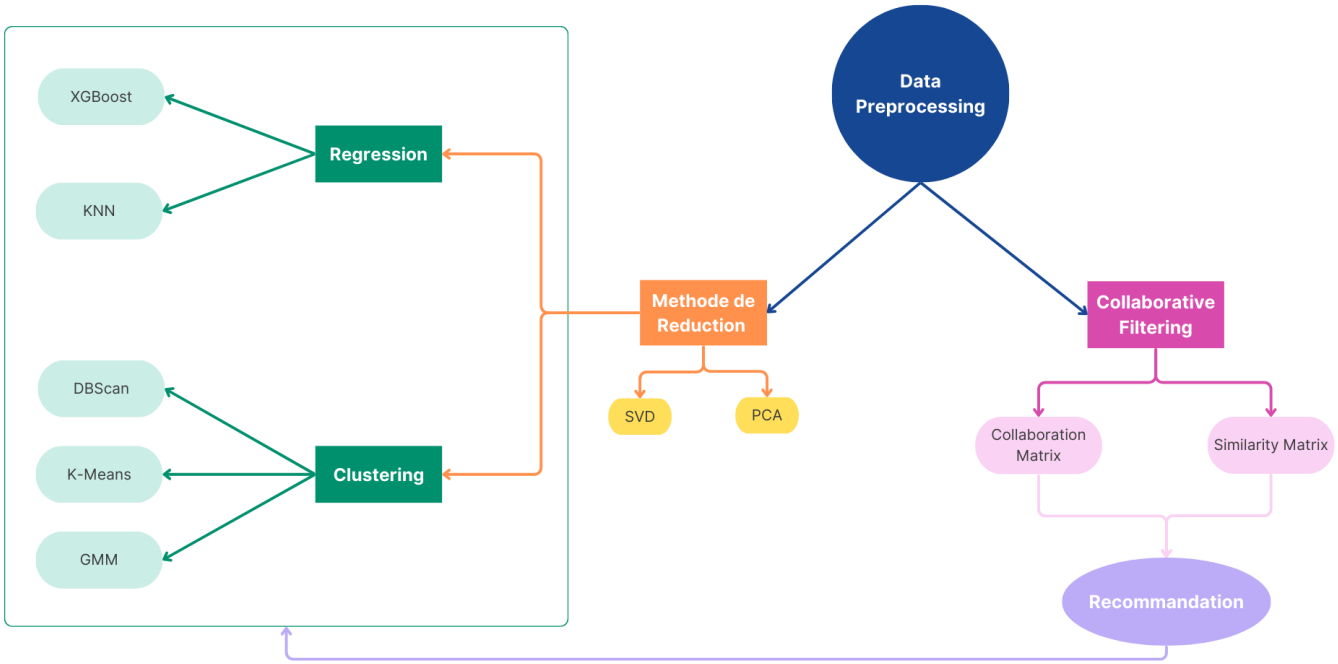

Figure 1: Méthodologie employée pour la recommandation musicale

F. Méthodologie

1. Dataset

Le dataset utilisée lors de ce projet est disponible en open source sur Kaggle, et a été créée par Joakim Arvidsson et contient plus de 30 000 morceaux provenant d'une API Spotify [1]. Ce dataset contient un total de 23 variables qui permettent de décrire et catégoriser chaque musique en fonction de ces variables.

Bien que le nom du morceau, de l'artiste ou de l'album soient importants, ce ne sont pas ces variables qui permettront de classifier les morceaux. En effet, une sélection de variables est faite en amont, ne gardant que celles qui décrivent les techniques et émotions d'une piste audio. Elles permettent ainsi d'identifier les caractéristiques clés de l'expérience d'écoute qui peuvent influencer les préférences des utilisateurs et ainsi recommander des morceaux similaires. Voici comment chacune contribue efficacement à un système de recommandation : danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence.

Tableau 1: Variables utilisées pour le clustering et la régression

Caractéristique	Définition	Plage de valeurs	Utilisation	Exemple d'utilisation
Danceability	Potentiel de danse	0.0 - 1.0	Identifier les morceaux dansants	Créer une playlist pour une soirée dansante
Energy	Intensité et dynamisme	0.0 - 1.0	Mesurer l'intensité	Recommander des chansons pour faire du sport
Key	Tonalité musicale	0 - 11	Influencer l'émotion	Créer une playlist mélancolique en sélectionnant des morceaux en mode mineur
Loudness	Volume	dB	Créer des playlists homogènes	Ajuster le volume d'une playlist pour une écoute confortable
Mode	Mode majeur ou mineur	0 (mineur) - 1 (majeur)	Ajuster l'humeur	Créer une playlist pour se détendre avec des morceaux en mode majeur
Speechiness	Présence de paroles	0.0 - 1.0	Distinguer chansons et podcasts	Recommander des chansons sans paroles pour se concentrer
Acousticness	Niveau d'acoustique	0.0 - 1.0	Identifier les morceaux acoustiques	Créer une playlist avec des sons naturels
Instrumentalness	Absence de paroles	0.0 - 1.0	Séparer morceaux instrumentaux et chansons	Créer une playlist pour se relaxer sans paroles
Liveness	Ambiance live	0.0 - 1.0	Évaluer l'ambiance live	Créer une playlist pour se sentir comme à un concert
Valence	Émotion positive	0.0 - 1.0	Déterminer la vitesse	Créer une playlist énergique pour faire de l'exercice

2. Design du modèle

3.1. Prepocessing

Le préprocessing des données est une étape clé pour nettoyer et préparer les données brutes, en corrigeant les erreurs et en réduisant les biais, afin d'améliorer les performances des systèmes de recommandation [2].

3.1.1. Suppression des lignes vides

Certaines lignes du dataset sont vides, cela étant indiqué par « NaN ». Celles-ci peuvent introduire du bruit et ainsi réduire la qualité des résultats. Supprimer ces colonnes permet donc d'éviter les erreurs et de s'assurer que le modèle utilise une base fiable et complète. De ce fait, 5 lignes ont été supprimées.

3.1.3. Suppression des doublons pour le clustering

Dans le dataset certains titres sont présents plusieurs fois. La présence de ces doublons peut biaiser l'analyse car certaines musiques, étant répertoriées plusieurs fois, auront plus de poids dans la décision du morceau à recommander

pour lors du clustering. En effet, cela leur donne une influence plus importante, faussant les résultats et les calculs de similarité. Afin de les repérer et de les supprimer, il a fallu regarder si le même nom de la variable « track_id » revenait plusieurs fois dans le tableau. Ainsi, 4476 doublons ont été repérés et ont pu être supprimés de la data frame.

Cependant, ces doublons seront utilisés dans un dataset spécial pour la partie collaborative filtering. En effet cette étape se reposera sur les morceaux ayant une occurrence supérieure à 5 dans la dataset, afin de pouvoir proposer des recommandations les plus précises et pertinentes.

3.1.2. Sélection de variables

Il est nécessaire dans ce projet de ne garder que les variables pertinentes afin de réduire la complexité de l'analyse, et d'améliorer la précision et la pertinence des recommandations faites par le modèle. Ainsi, seules « Danceability », « Key », « Loudness », « Mode », « Speechiness », « Acousticness », « Instrumentalness », « Liveness », « Valence » sont gardées.

3.1.4. Standardisation des données

La standardisation est une étape majeure du prétraitement de données car elle permet de mettre toutes les variables à la même échelle, ce qui est essentiel pour les modèles sensibles à l'échelle des données, comme les modèles de clustering utilisés dans ce projet. Pour ce faire, StandardScaler, un outil de la bibliothèque Scikit-learn, est utilisé. En standardisant, chaque variable est centrée réduite en calculant la moyenne et l'écart-type de chaque variable, puis en transformant chaque valeur pour qu'elle ait une moyenne de 0 et un écart-type de 1 [3].

Cela facilite l'optimisation et améliore les comparaisons entre les données. De plus, les données transformées sont optimisées pour les modèles qui reposent sur des distributions gaussiennes, comme le Principal Component Analysis (PCA) qui sera utilisé avant le clustering.

3.2. Réduction de dimensionnalité

La réduction de dimensionnalité simplifie les données en les projetant dans un espace de moindre dimension tout en préservant l'essentiel de l'information. Elle réduit la complexité, accélère les calculs et limite le sur-apprentissage, avec des méthodes comme PCA ou SVD par exemple [4].

3.2.1. Principal Component Analysis

Cette méthode de réduction de dimensionnalité illustrée sur la Figure 2, permet de préserver la variance maximale dans les données, en les projetant sur un sous espace de dimensions réduite. La variance représente la dispersion des données, donc en maximisant la variance, il est assuré de préserver les aspects les plus significatifs et distinctifs des données.

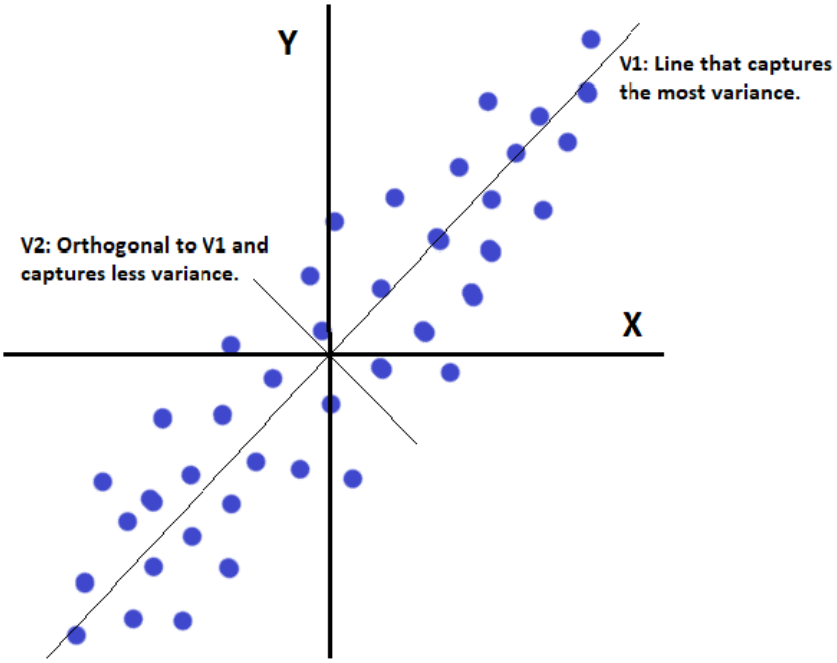

Figure 2: Graphique représentant la projection des données lors d'un PCA [3]

PCA permet de filtrer le bruit et les variations indésirables en supprimant les composantes qui ne capturent qu'une faible part de la variance, car elles sont souvent liées à des éléments aléatoires ou non significatifs. Elle est également utile pour détecter les valeurs aberrantes. Après transformation dans l'espace des composantes principales, les observations trop à l'écart du reste des données, peuvent être interprétées comme des points atypiques ou des anomalies [5].

3.2.2. Singular Value Decomposition

Dans un système de recommandation, il peut y avoir une matrice où chaque ligne représente un utilisateur, et chaque colonne représente un élément, comme ici un morceau. Les valeurs dans cette matrice représentent les évaluations ou les interactions des utilisateurs avec ces éléments. Cette méthode permet également de réduire la dimensionnalité en décomposant une matrice en trois sous-matrices, comme le montre la Figure 3, il y a la matrice des utilisateurs U, des valeurs singulières Σ et celle des éléments V* [6].

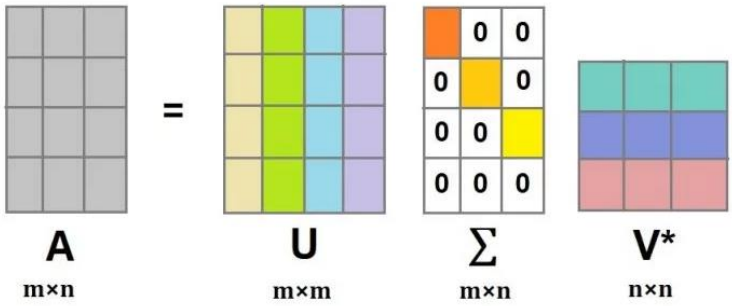

Figure 3: Singular Value Decomposition Matrices [4]

Il faut reconstruire la matrice en les multipliant, et les valeurs manquantes de la matrice, à savoir les évaluations de certains morceaux non fournis par l'utilisateur, pourront être prédites, et donc seront utilisées pour la recommandation.

Cependant, la SVD présente des inconvénients, notamment un coût de calcul élevé pour les grandes matrices, des difficultés avec le problème de démarrage à froid lorsqu'un nouveaux utilisateurs ou éléments entre dans le dataset, et des problèmes de sparsité des données.

3.3. Clustering

Le clustering est une technique d'apprentissage non supervisé qui consiste à regrouper des objets similaires en clusters, de sorte que les éléments d'un même groupe soient plus proches les uns des autres que ceux d'autres groupes. Dans ce projet, plusieurs méthodes seront appliquées puis comparées comme dans le Tableau 2. Celles-ci sont K-means [7] [8], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9], Gaussian Mixture Model (GMM) [10].

Pour certaines méthodes, comme ici K-Means, la méthode du coude est utilisée afin de trouver au préalable le nombre optimal de clusters à former à imposer au modèle. Celle-ci analyse la variance dans un cluster en fonction du nombre de cluster. Le « coude » de la courbe désigne la valeur pour laquelle ajouter un cluster supplémentaire n'améliore plus significativement la variance [11].

Pour d'autres comme GMM, il faut calculer le Critère d'Information Bayésien (BIC), utilisé pour la sélection de modèles statistiques. Il évalue la qualité d'ajustement d'un modèle tout en tenant compte de sa complexité afin de réduire le risque de surapprentissage. Un BIC plus faible indique un modèle mieux adapté. Ainsi, pour sélectionner un modèle, il suffit de choisir celui qui présente le score BIC le plus bas parmi les options disponibles [12].

Tableau 2: Comparaison des modèles de clustering

Méthode	Description	Forces	Faiblesses
K-means	Partitionne les données en k clusters	Simple à implémenter, efficace pour de grands datasets	Nécessite de connaître le nombre de clusters a priori, sensible aux valeurs aberrantes et aux initialisations
DBSCAN	Identifie les clusters de densité	Détecte les valeurs aberrantes automatiquement, ne nécessite pas de connaître le nombre de clusters	Sensible aux paramètres de rayon de voisinage et de densité minimale , peut avoir du mal avec des clusters de densités très différentes
Gaussian Mixture Model (GMM)	Modélise les données comme un mélange de gaussiennes	Représentation plus flexible des données, prend en compte la structure de covariance	Plus complexe à implémenter et à interpréter que K-means

3.4. Régression

Un modèle de régression est une méthode d'apprentissage supervisé permettant de prédire une valeur continue en fonction d'une ou plusieurs variables. Elle établit une relation entre la variable cible dépendante et les variables explicatives indépendantes. Dans ce projet sont utilisés K-Nearest Neighbors (KNN) [13], et Extreme Gradient Boosting (XGBoost) [14], et ils sont comparé Tableau 3 dans le suivant.

Tableau 3: Comparaison des modèles de régression

Méthode	Description	Forces	Faiblesses	Utilisation typique
K-Nearest Neighbors (kNN)	Classifie un point en fonction de ses k voisins les plus proches	Simple à comprendre et à implémenter, ne nécessite pas d'entraînement	Lent sur de grands datasets, sensible au bruit, choix du paramètre k difficile	Classification, régression, recommandation
XGBoost	Algorithme de boosting par gradient	Très performant pour de nombreux problèmes, parallélisation efficace	Nécessite un réglage fin des hyperparamètres, moins interprétable que d'autres modèles	Classification, régression, classement, prédiction de séries temporelles

3.5. Collaborative filtering

Le collaborative filtering (filtrage collaboratif) est une technique populaire dans les systèmes de recommandation, utilisée pour proposer des éléments à un utilisateur en s'appuyant sur les interactions des autres utilisateurs avec ces éléments. Cette approche repose sur deux principales matrices : la collaboration matrix (matrice de collaboration) et la similarity matrix (matrice de similarité). Il existe deux types de filtrage collaboratif : d'utilisateur à utilisateur ou d'éléments à éléments [15].

Un avantage majeur du filtrage collaboratif est qu'il fait des recommandations personnalisées sans nécessiter de données sur le contenu des éléments. Cependant, il a du mal à faire des recommandations précises pour les nouveaux utilisateurs ou éléments, c'est ce qu'on appelle le démarrage à froid. De plus, il souffre de la sparsité des données, car la majorité des utilisateurs interagissent avec peu d'éléments. Enfin, il peut devenir coûteux en calcul sur de très grandes bases de données.

3.5.1. Collaboration matrix

Cette matrice représente les interactions brutes entre utilisateurs et éléments. Les valeurs dans cette matrice (comme des évaluations, des clics, ou des achats) servent de base pour déterminer les similarités, qu'il s'agisse de similarités entre utilisateurs ou entre éléments. Elle fournit les données d'interactions nécessaires pour calculer les scores de similarité dans la matrice de similarité. Deux types de filtrage collaboratif s'appuient sur cette matrice : filtrage collaboratif d'utilisateur à utilisateur ou d'éléments à éléments [16].

3.5.2. Similarity matrix

Cette matrice est calculée à partir des données de la matrice de collaboration et peut donc être construite de deux manières : filtrage collaboratif d'utilisateur à utilisateur ou d'éléments à éléments. Est utilisé dans ce projet le filtrage d'éléments à éléments. Elle quantifie les liens entre utilisateurs ou éléments en fonction de leurs préférences similaires. Enfin, elle permet d'estimer les interactions manquantes dans la matrice de collaboration en se basant sur les relations entre utilisateurs ou entre éléments [17].

Pour un filtrage basé sur les utilisateurs, la matrice de similarité compare les utilisateurs deux à deux, en mesurant leur degré de similarité (par exemple avec le cosinus de similarité ou la corrélation de Pearson). Dans un filtrage basé sur les éléments, la similarité est mesurée entre les éléments eux-mêmes, en fonction des interactions des utilisateurs.

3.6. Recommandation

Le système de recommandation combine les résultats du collaborative filtering et du clustering pour proposer des morceaux pertinents [17]. Dans un premier temps, le collaborative filtering génère une liste de morceaux classés par ordre de similarité avec le morceau de départ. Cette liste est composée uniquement des morceaux dupliqués dans le dataset d'origine. Les cinq morceaux les plus similaires sont sélectionnés à cette étape.

Ensuite, le cluster majoritaire est le genre dominant parmi ces cinq morceaux sont identifiés. Le clustering, appliqué à l'ensemble des morceaux du dataset, permet d'élargir le spectre des recommandations en incluant des morceaux similaires issus d'autres parties des données. Quatre morceaux sont alors choisis aléatoirement parmi ceux qui appartiennent au même cluster et au même genre. Ils sont ajoutés à la liste de recommandations, qui contient désormais neuf morceaux.

Enfin, un dixième morceau est recommandé. Celui-ci est sélectionné dans le cluster majoritaire parmi les cinq premiers morceaux, mais il doit appartenir à un genre différent. L'objectif est de diversifier les suggestions tout en maintenant une certaine cohérence, en proposant un morceau qui s'éloigne légèrement des habitudes de l'auditeur.

4. Implémentation

L'implémentation du projet a été réalisée en Python à l'aide de Jupyter Notebook, permettant une analyse interactive des données et la section et organisation les étapes. Plusieurs bibliothèques ont été utilisées pour optimiser les calculs et la visualisation : NumPy pour les calculs numériques, Pandas pour la manipulation et l'analyse des données, Matplotlib pour la création de graphiques détaillés, et Seaborn pour des visualisations statistiques avancées. Ces outils ont permis de structurer efficacement le pipeline et d'obtenir des représentations claires des résultats.

G. Résultats

1. Réduction de dimensionnalité

La réduction de dimensionnalité à l'aide des méthodes de PCA et de SVD nous a permis de passer de neufs dimensions, à savoir les colonnes décrivant les techniques et émotions d'une piste audio à seulement deux dimensions. Ces deux dimensions permettent une bien meilleure visualisation des données tout en conservant le maximum de la variance. La perte d'information liée à la réduction de dimensionnalité est ainsi minimisée.

2. Clustering

Une fois les données réduites en deux dimensions, nous avons pu appliquer différentes méthodes de clustering : K-Means, DBscan, Gaussian Mixtures. DBscan est particulièrement difficile à paramétrer et n'a pas été capable de créer des clusters mais uniquement d'isoler certains points. Ainsi, les résultats les plus intéressants sont ceux produits par K-Means.

La méthode du coude affiché sur la Figure 4 n'a pas donné de résultat pertinent pour choisir le nombre de clusters, car la courbe est lisse et ne présente aucun angle particulier. Ainsi, il a été choisi de conserver 6 clusters correspondant aux 6 genres principaux présents dans les données.

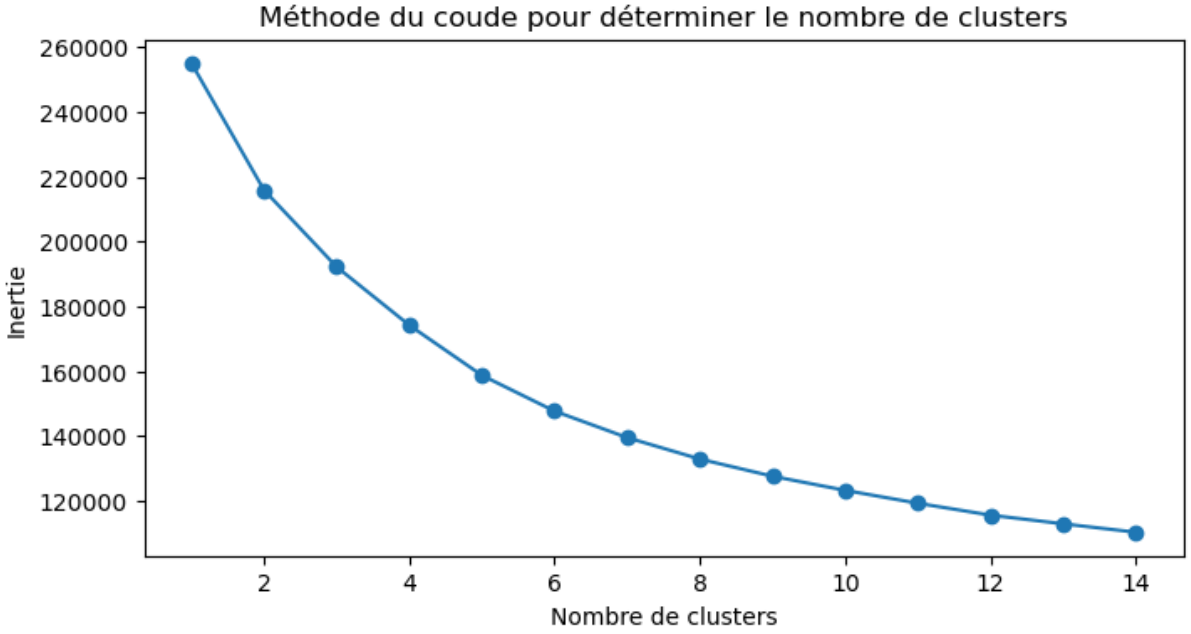

Figure 4: Méthode du coude pour déterminer le nombre de cluster pour le K-Means

Le score de silhouette étant de 0.33 sur une échelle allant de [-1 ;1] comme indiqué Figure 5, il est possible de conclure que le clustering soit l'association d'un morceau à un cluster, est réussi et précis.

```
Score de silhouette pour PCA : 0.33
```

Figure 5: Score de silhouette du K-Means

Ainsi la Figure 6 permet de visualiser les 6 clusters comme choisi, et chaque morceau fait partie d'un cluster. Ici, les clusters sont bien séparés ce qui montre que l'algorithme K-means a réussi à identifier des sous-populations dans les données. En effet, chaque cluster représente une sous-population, ici chaque couleur correspond à un cluster différent représentant un des 6 genres musicaux. Et chaque cluster regroupe des points ayant des caractéristiques similaires.

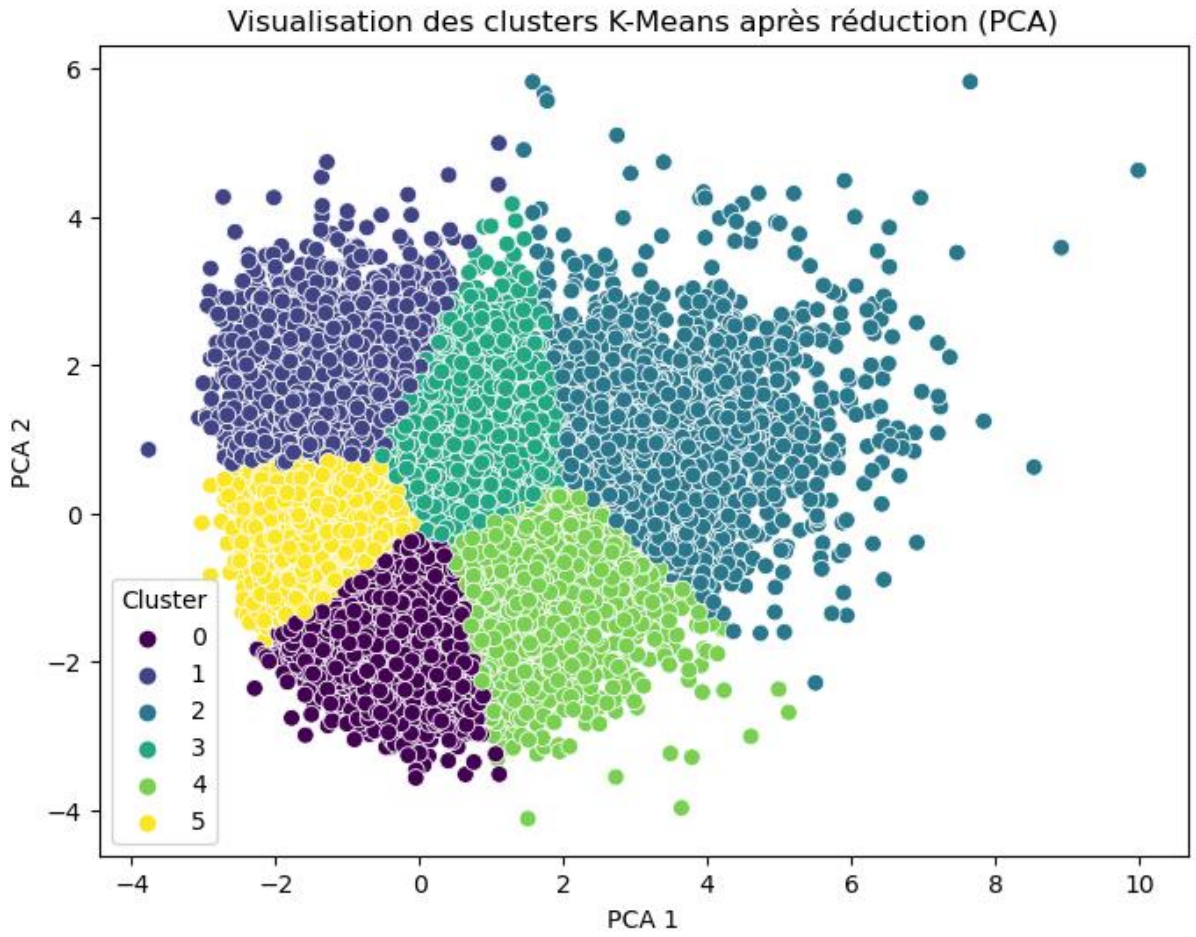

Figure 6: Visualisation des clusters K-Means après PCA

3. Régression

Les méthodes de régressions telles que KNN et XGBoost ne sont pas les plus adaptées pour le problème de recommandation musicale. En effet, la musique se basant sur plusieurs variables (danceability, energy, etc...) la modélisation avec une régression est difficile. Par exemple, KNN ne se base que sur le genre musical pour la recommandation, là ou un modèle de clustering regroupe les morceaux et en recommande en prenant en compte toutes les caractéristiques requises similaires .

De plus, XGBoost n'est simplement pas adapté au critères de ce projet puisqu'il obtient une précision de recommandation de 0.56, ce qui est bien trop faible pour que celle-ci soit précise et adaptée.

4. Collaborative filtering

Afin d'effectuer le collaborative filtering, il a fallu construire la collaboration matrix et similariity matrix.

Ainsi, la collaborative matrix obtenue est une 70*91. La matrice contient donc 70 lignes représentants les morceaux ayants une occurrence dans le dataset supérieur à 5. Elle contient également 91 colonnes, étan toutes les playlists différentes présentes dans le dataset.

La similarity matrix obtenue est une 70*70, il y a donc autat de lignes que de colonnes puisque cette matrice est basée sur les 70 morceaux également utilisés dans la collaborative matrix. En effet cette similarity matrix est basée sur les interactions éléments-éléments.

5. Recommandation

Afin de tester la qualité des recommandations de ces méthodes, un morceau a été choisi au hasard, puis placé en input de la matrice de similarité afin que celle-ci recommande des morceaux similaires. Il d'abord a fallu avoir enlevé de la liste de recommandation proposée en output le morceau sélectionné en input car la similarité est de 1 puisqu'il s'agit du même morceaux. Plus largement, cette étape permettra de ne jamais proposer un morceau déjà écouter par l'utilisateur. Puis, la matrice propose un classement des 69 morceaux restants dans le dataset, avec un score de similarité allant de 0 à 0.909. Pour avoir les résultats les plus pertinent, sels les 5 premiers morceaux sont gardés pour pouvoir être proposés à l'auditeur comme indiqué Figure 7.

```
track_id
2plLJpUcYPFrl1sW2pMG63    0.909091
7LzouaWGFCy4tkXDOOnEyM    0.836242
0qaWEvPkts34WF68r8Dzx9    0.836242
0rIAC4PXANcKmitJfoqmVm    0.836242
3HVWdVOQ0ZA45FuZGSfvns    0.818182
Name: 0bMbDctzMmTyK2j74j3nF3, dtype: float64
```

Figure 7: Output des scores de similarité les plus élevés pour 5 morceaux

Il faut ensuite identifier le cluster cible, dans lequel le plus de morceaux similaires seront présents. Pour cela il faut récupérer les indices des 5 morceaux comme sur la Figure 8 afin de les repérer dans le dataset original contenant aussi tous les autres morceaux. Grace à cet indice, il est possible de retrouver les clusters auxquels les 5 morceaux appartiennent.

```
141     0qaWEvPkts34WF68r8Dzx9
661     0rIAC4PXANcKmitJfoqmVm
1247    7LzouaWGFCy4tkXDOOnEyM
1315    2plLJpUcYPFrl1sW2pMG63
4655    3HVWdVOQ0ZA45FuZGSfvns
Name: track_id, dtype: object
```

Figure 8: Output des associations des indices correspondants aux morceaux

Il a été choisi de filtrer les chansons appartenant au cluster cible et ayant le même genre prioritaire, puis au genre secondaire. De ce fait, 4 morceaux similaires appartenant au même cluster seront recommandés en plus des 5 morceaux recommandés par la similarity matrix.
Les 9 premiers morceaux appartiennent donc au genre prioritaire, et un 10ème morceau est choisi comme le montre la Figure 9, celui-ci appartenant au même cluster mais à un genre secondaire, afin de diversifier la recommandation.

	track_name	track_artist
141	Turn Me On (feat. Vula)	Riton
661	Motivation	Normani
1247	Liar	Camila Cabello
1315	Lights Up	Harry Styles
4655	I Don't Care (with Justin Bieber)	Ed Sheeran
16207	Messiah	Klingande
16208	Never Alone	Felix Jaehn
16210	All or Nothing - Sultan + Shepard Remix	Lost Frequencies
16211	Are You Mine	Alex Schulz

Figure 9: Liste finale des recommandations musicales

H. Conclusion et Perspectives

Pour conclure, ce projet a démontré l'efficacité des systèmes de recommandation musicale basés sur l'apprentissage automatique, en utilisant des techniques comme la réduction de dimensionnalité, et l'association du collaborative filtering et du clustering pour personnaliser les suggestions musicales. Parmi les différences méthodes de réduction de dimensionalité, c'est la méthode du PCA qui s'est avéré etre la plus efficace. Quant au clustering, GMM et K-Means ont été les plus performants, bien que la recommandation se soit finalement basée sur K-Means. De plus il s'est avéré que la régression n'était pas la méthode la mieux adapté à la recommandation, ne fournissant pas de résultats exploitables.

Des pistes d'amélioration incluent l'utilisation de réseaux neuronaux profonds pour des analyses plus complexes, l'intégration de données contextuelles pour rendre les recommandations plus dynamiques, et l'application de méthodes hybrides pour renforcer la robustesse du système. De plus, des techniques comme le transfert learning pourraient être explorées pour résoudre le problème du démarrage à froid. L'expansion du système à plus d'utilisateurs et de données permettrait d'améliorer la diversité et l'efficacité des recommandations. Ces perspectives visent à rendre les systèmes de recommandation musicale encore plus personnalisés et adaptés aux besoins des utilisateurs.

I. Références

[1] J. Arvidsson, «30000 Spotify Songs,» Kaggle, 2023. [En ligne]. Available: https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs.

[2] J. Robert, «Préprocessing: Qu'est-ce que c'est ? Comment ça marche ?,» DataScientest, 11 04 2022. [En ligne]. Available: https://datascientest.com/guide-du-data-preprocessing.

[3] Opendatasoft, «Données standardisées,» Opendatasoft, s.d.. [En ligne]. Available: https://www.opendatasoft.com/fr/glossaire/donnees-standardisees/#:~:text=Les%20donn%C3%A9es%20standardis%C3%A9es%20sont%20des,leur%20permettant%20d'%C3%AAtre%20compar%C3%A9es..

[4] R. Kassel, «Réduction de dimension : Comment ça fonctionne ?,» DataSientest, 16 06 2023. [En ligne]. Available: https://datascientest.com/reduction-de-dimension.

[5] J. Frost, «Principal Component Analysis Guide & Example,» 2024. [En ligne]. Available: https://statisticsbyjim.com/basics/principal-component-analysis/.

[6] A. Wasnik, «Singular Value Decomposition (SVD) in Python,» AskPython, 2020. [En ligne]. Available: https://www.askpython.com/python/examples/singular-value-decomposition.

[7] DataScientest, «K-Means Clustering in Machine Learning: A Deep Dive,» DataScientest, 10 09 2023. [En ligne]. Available: https://datascientest.com/en/k-means-clustering-in-machine-learning-a-deep-dive.

[8] N. Singh, «What is K-Means Clustering?,» Medium, 09 07 2020. [En ligne]. Available: https://ai.plainenglish.io/what-is-k-means-clustering-3060791cb589.

[9] R. Yehoshua, «DBSCAN: Density-Based Clustering,» Medium, 17 10 2023. [En ligne]. Available: https://ai.plainenglish.io/dbscan-density-based-clustering-aaebd76e2c8c.

[10] Scikit Learn, «Gaussian mixture models,» Scikit Learn, 2024. [En ligne]. Available: https://scikit-learn.org/1.5/modules/mixture.html.

[11] É. Blent, «Algorithme k-means : comment ça marche ?,» Blent, 25 01 2022. [En ligne]. Available: https://blent.ai/blog/a/k-means-comment-ca-marche.

[12] Statistical & Financial Consulting by Stanford PhD, «BAYESIAN INFORMATION CRITERION,» Statistical & Financial Consulting by Stanford PhD, s.d.. [En ligne]. Available: https://stanfordphd.com/BIC.html.

[13] J. Robert, «Qu'est ce que l'algorithme KNN ?,» DataScientest, 19 11 2020. [En ligne]. Available: https://datascientest.com/knn.

[14] Nvidia, «XGBoost,» Nvidia, [En ligne]. Available: https://www.nvidia.com/en-us/glossary/xgboost/.

[15] Evelyn, «Collaborative Filtering in Recommender System: An Overview,» Medium, 04 11 2023. [En ligne]. Available: https://medium.com/@evelyn.eve.9512/collaborative-filtering-in-recommender-system-an-overview-38dfa8462b61.

[16] Data Mesh Architecture , «What is Collaborative Matrix Model,» Weaver, 31 07 2020. [En ligne]. Available: https://weaver.com.sg/collaborative-matrix-model/.

[17] C. Borodescu, «The anatomy of high-performance recommender systems – Part IV,» Algolia, [En ligne]. Available: https://www.algolia.com/blog/ai/the-anatomy-of-high-performance-recommender-systems-part-iv/.

[18] «5 mins Recommender systems: Matrix Factorization Collaborative Filtering,» Medium, 10 05 2024. [En ligne]. Available: https://medium.com/@omar.tafsi_46401/5-mins-recommender-systems-matrix-factorization-collaborative-filtering-8eead5fbe18c.

J. Appendice

Tableau 4: Variables présentent dans le dataset

Variable	Class	Description
track_id	character	Unique ID of the song
track_name	character	Name of the song
track_artist	character	Artist of the song
track_popularity	double	Song popularity (0-100), higher is better
track_album_id	character	Unique ID of the album
track_album_name	character	Name of the album
track_album_release_date	character	Date when the album was released
playlist_name	character	Name of the playlist
playlist_id	character	ID of the playlist
playlist_genre	character	Genre of the playlist
playlist_subgenre	character	Subgenre of the playlist
danceability	double	Danceability (0.0-1.0), higher is more danceable
energy	double	Energy (0.0-1.0), higher is more energetic
key	double	Estimated overall key of the track (-1 to 11)
loudness	double	Overall loudness of a track in decibels (dB)
mode	double	Mode of the track (0 = Minor, 1 = Major)
speechiness	double	Speechiness (0.0-1.0), higher is more speech-like
acousticness	double	Acousticness (0.0-1.0), higher is more acoustic
instrumentalness	double	Instrumentalness (0.0-1.0), higher is more instrumental
liveness	double	Liveness (0.0-1.0), higher is more likely live
valence	double	Valence (0.0-1.0), higher is more positive
tempo	double	Tempo of the track in beats per minute (BPM)
duration_ms	double	Duration of the song in milliseconds