

Projet de Décision Collective, Décision Multicritère
Sujet : Recommandation de Musique

Sepanta FARZOLLAHI

Mohamed El Amine ROUIBI

Responsable : Hassan AISSI



17 janvier 2025

Table des matières

1	Introduction	1
1.1	Le rôle des recommandations musicales	1
1.2	Les playlists et leur importance	1
1.3	Objectifs du projet	2
1.4	Méthodologie adoptée	2
2	Dataset et Prétraitement	3
2.1	Choix du dataset	3
2.2	Prétraitement des données	3
2.2.1	Description du dataset	4
2.2.2	Enrichissement des données par diversification des genres	5
2.2.3	Clustering et préparation finale des données	5
3	Choix et Organisation des Critères	7
3.1	Critères retenus	7
3.2	Cohérence des critères	8
3.3	Hierarchie des critères	8
4	Algorithme de Recommandation	10
4.1	Algorithmes testés	10
4.1.1	Agrégation lexicographique	10
4.1.2	Méthode de Condorcet	11
4.2	Algorithme choisi	12
4.2.1	Somme pondérée	12
5	Résultats et Analyse	15
5.1	Structure du code	15
5.2	Exécution avec un exemple pratique	16
5.3	Analyse des résultats	17
6	Améliorations et Perspectives	20
6.1	Contraintes et axes d'amélioration	20
6.2	Exploration d'approches alternatives	21
6.2.1	Méthodes de surclassement (ELECTRE III)	21
6.2.2	Exploration de KNN (K-Nearest Neighbors)	21
7	Conclusion	23

Chapitre 1

Introduction

L'évolution des technologies de l'information et de la communication a profondément transformé l'industrie musicale ces dernières années. Avec l'essor de plateformes de streaming comme *Spotify*, *Apple Music*, et *Deezer*, la manière dont les utilisateurs découvrent et écoutent de la musique a changé radicalement. Ces plateformes utilisent des algorithmes sophistiqués de recommandation pour personnaliser les expériences d'écoute en fonction des préférences individuelles, permettant ainsi de découvrir de nouveaux morceaux, artistes et genres musicaux.



1.1 Le rôle des recommandations musicales

Les systèmes de recommandation jouent un rôle central dans ces plateformes, offrant une sélection de morceaux adaptés aux goûts de chaque utilisateur. Ces systèmes s'appuient sur diverses techniques telles que la *collaborative filtering*, *content-based filtering*, ou des méthodes hybrides qui combinent plusieurs approches. Leur objectif est de proposer des contenus musicaux qui non seulement correspondent aux préférences des utilisateurs, mais aussi de leur faire découvrir de nouveaux morceaux susceptibles de les intéresser. Le système de recommandation de Spotify, par exemple, analyse des millions de morceaux en fonction de critères variés tels que le genre musical, l'artiste, le tempo, et même les émotions transmises par les chansons. Ces analyses sont ensuite utilisées pour générer des listes de lecture, ou *playlists*, qui répondent aux attentes des utilisateurs.

1.2 Les playlists et leur importance

Les playlists jouent un rôle essentiel dans l'écosystème des services de streaming musical. Ces listes de morceaux, souvent créées à la main ou de manière algorithmique, permettent aux utilisateurs de vivre des expériences d'écoute sur mesure. Que ce soit pour une soirée entre amis, une séance de sport ou simplement une écoute tranquille à la maison, les playlists offrent une sélection musicale cohérente et adaptée à chaque moment. Par ailleurs, les playlists personnalisées, comme *Discover Weekly* sur Spotify¹, utilisent les données de l'utilisateur pour

1. Plus d'informations sur l'algorithme de *Discover Weekly* : <https://medium.com/the-sound-of-ai/spotify-s-discover-weekly-explained-breaking-from-your-music-bubble-or-maybe-not-b506da144123>.

créer une sélection de morceaux qu'il est susceptible d'aimer, basées sur son historique d'écoute et ses interactions avec la plateforme.

Les plateformes de streaming telles que Spotify permettent aussi aux utilisateurs de créer leurs propres playlists, les partageant avec leurs amis ou le public. Cela renforce l'aspect social de la musique, en permettant à chacun d'exprimer ses goûts tout en découvrant de nouvelles musiques grâce aux sélections des autres.

Dans ce contexte, il est crucial pour les systèmes de recommandation d'offrir des résultats de haute qualité, adaptés aux goûts des utilisateurs tout en offrant un certain degré de surprise pour encourager la découverte. Ce projet s'inscrit dans cette dynamique, en étudiant des critères de similarité pour calculer des recommandations de morceaux et générer des playlists intelligentes.

1.3 Objectifs du projet

L'objectif de ce projet est de développer un système de recommandation musicale basé sur plusieurs critères de similarité entre les morceaux. Nous chercherons à explorer différents critères comme l'artiste, le genre musical, la date de sortie, les sujets abordés dans les morceaux et d'autres fonctionnalités afin de générer des playlists personnalisées. L'idée est d'améliorer la qualité des recommandations en affinant les critères de similarité et en proposant une expérience plus précise et diversifiée pour l'utilisateur.

Concrètement, à partir d'une chanson donnée par l'utilisateur, reflétant ses préférences musicales, le système de recommandation génère automatiquement une playlist composée de morceaux similaires. Cette approche permet non seulement de proposer des titres en adéquation avec les goûts de l'utilisateur, mais aussi de lui faire découvrir de nouvelles chansons et d'enrichir son expérience musicale.

1.4 Méthodologie adoptée

Pour ce projet, un système de recommandation fondé sur l'analyse de plusieurs critères a été développé. Les métadonnées des morceaux, telles que l'artiste, le genre musical et la date de sortie, ont été exploitées pour mesurer la similarité entre les chansons. De plus, des critères complémentaires comme les thèmes abordés dans les paroles et certaines caractéristiques audio (danseabilité, énergie, etc.) ont été intégrés afin d'enrichir et d'affiner les recommandations.

Une particularité de ce projet est de formuler la génération de playlists comme un problème de décision et de classement. En effet, lorsqu'un utilisateur sélectionne une chanson donnée comme point de départ, l'objectif du système est de proposer une playlist cohérente, organisée selon des critères pertinents. Ce processus peut être assimilé à un problème de rangement, où les morceaux doivent être classés de manière optimale en fonction de leur proximité ou complémentarité avec la chanson initiale.

Cette approche vise non seulement à offrir des recommandations pertinentes et alignées avec les goûts de l'utilisateur, mais aussi à maintenir une diversité musicale et à faciliter la découverte de nouvelles œuvres. Le système sera évalué sur sa capacité à résoudre efficacement ce problème de classement tout en préservant l'expérience utilisateur.

Chapitre 2

Dataset et Prétraitement

Le bon choix du dataset et un prétraitement adéquat sont des étapes essentielles pour garantir la performance du système de recommandation. La qualité des données utilisées, ainsi que les transformations appliquées avant leur traitement, influencent directement la pertinence et l'efficacité des résultats obtenus.

2.1 Choix du dataset

Le choix du dataset constitue une étape cruciale dans ce projet, car il impacte directement la pertinence et la qualité des recommandations générées. À l'origine, nous avons envisagé d'exploiter l'API de Spotify pour collecter les informations nécessaires sur les morceaux. Cependant, cette solution s'est avérée limitée pour notre projet. Bien que l'API de Spotify fournisse des métadonnées intéressantes, comme l'énergie ou la danseabilité d'une chanson, elle ne permet pas d'accéder à certaines informations essentielles, telles que le genre musical ou la langue des chansons. Ces limitations ont rendu cette option peu adaptée à nos besoins.

Face à cette contrainte, nous avons décidé de rechercher un dataset préexistant. Après avoir exploré plusieurs sources et comparé différents jeux de données, nous avons opté pour le *Music Dataset : 1950 to 2019*, disponible sur Kaggle [Shahane \(2019\)](#). Ce dataset se distingue par sa richesse en informations, incluant des attributs clés tels que le genre, l'artiste, la date de sortie, ainsi que d'autres caractéristiques essentielles des morceaux de musique. Bien qu'il manque certaines données supplémentaires, comme la langue des chansons ou le nom de l'album, il reste néanmoins l'un des ensembles de données les plus complets que nous avons trouvés.

Cette combinaison de diversité et de richesse d'attributs en fait un choix pertinent et adapté aux besoins de notre projet, notamment pour la création de recommandations musicales basées sur des critères variés.

2.2 Prétraitement des données

Le prétraitement des données est une étape fondamentale pour garantir que les informations utilisées dans le modèle de recommandation soient cohérentes et exploitables. Il comprend la gestion des caractéristiques initiales du dataset, ainsi que les étapes de nettoyage, transformation et préparation des données.

2.2.1 Description du dataset

Le dataset utilisé, intitulé *Music Dataset : 1950 to 2019* [Shahane \(2019\)](#), contient un total de 28 372 chansons provenant de 5 426 artistes différents. Voici un exemple pour mieux illustrer les données présentes dans ce dataset, avec la chanson *Sympathy for the Devil* des *Rolling Stones*.

artist_name	the rolling stones
track_name	sympathy for the devil
release_date	1966
genre	rock
lyrics	allow introduce wealth taste long...
len	107
dating	0.0009233610397165044
violence	0.3116658473845667
world/life	0.0009233611312163228
night/time	0.06971804597269644
shake the audience	0.05107529508658665
family/gospel	0.000923361056407817
romantic	0.08713766591894917
communication	0.06871796362851121
obscene	0.18890433075293298
music	0.0009233610560024295
movement/places	0.21170051848888077
light/visual perceptions	0.0009233610728039011
family/spiritual	0.00092336105294547
like/girls	0.0009233610773345778
sadness	0.0009233610762139538
feelings	0.0009233610393121782
danceability	0.4248889851619192
loudness	0.6996641284003794
acousticness	0.16465779584116047
instrumentalness	$4.02834008097166 \times 10^{-5}$
valence	0.8773701566364386
energy	0.8898864509081764
topic	violence
age	0.7714285714285715

Table 2.1 : Les données présentes pour la chanson *Sympathy for the Devil*

Comme on peut le voir pour cette chanson, le dataset regroupe plusieurs colonnes fournissant des informations variées sur chaque morceau :

- **artist_name** : Nom de l'artiste.
- **track_name** : Nom de la chanson.
- **release_date** : Année de sortie, variant de 1950 à 2019.
- **genre** : Genre musical attribué à chaque chanson. Les genres disponibles dans le dataset sont les suivants :
 - pop, country, blues, jazz, reggae, rock, hip hop.

- lyrics : Extrait partiel des paroles de la chanson, non complet.
- len : Longueur des paroles partielles fournies (lyrics).
- dating à feelings : Seize colonnes représentant des sujets thématiques liés aux chansons, avec des valeurs comprises entre 0 et 1, reflétant la présence relative de chaque thème. Les sujets sont les suivants :
 - dating, violence, world/life, night/time, shake the audience, family/gospel, romantic, communication, obscene, music, movement/places, light/visual perceptions, family/spiritual, like/girls, sadness, feelings.
- danceability à energy : Six colonnes représentant les caractéristiques audio des morceaux, également exprimées entre 0 et 1. Ces caractéristiques sont :
 - danceability, loudness, acousticness, instrumentality, valence, energy.
- topic : Colonne existante indiquant le sujet dominant parmi les 16 sujets mentionnés précédemment.
- age : Colonne redondante par rapport à release_date, exprimant l'ancienneté relative de la chanson sous forme normalisée (valeurs entre 0 et 1).

Bien que nous ne détaillons pas cette étape, à chaque artiste sera associé un identifiant unique. De même, un identifiant unique sera attribué à chaque chanson, ainsi qu'à chacun des sujets thématiques allant de dating à feelings (de 1 à 16), et pour chaque caractéristique audio, un identifiant sera attribué (de 1 à 6) pour des colonnes danceability à energy.

2.2.2 Enrichissement des données par diversification des genres

Afin d'améliorer la précision des recommandations musicales, nous avons enrichi le dataset d'origine en intégrant une classification plus détaillée des genres musicaux. Le dataset initial ne proposait que 7 genres principaux (pop, country, blues, jazz, reggae, rock, hip hop), ce qui limitait la diversité des suggestions.

Pour pallier cette limitation, des données issues du dataset *Music Recommendation System using Spotify Dataset*, disponible sur Kaggle [Mavani \(2023\)](#), ont été utilisées. Ce dataset répertorie de nombreux genres/sous-genres dans lesquels un artiste a déjà produit des morceaux. Ainsi, chaque chanson n'est plus restreinte à un genre unique, mais est associée à l'ensemble des genres explorés par son interprète.

Cette association multiple des genres permet de mieux refléter la diversité musicale d'un artiste et d'affiner les recommandations. Bien que cette méthode puisse introduire certaines imprécisions — toutes les chansons d'un artiste ne partageant pas nécessairement les mêmes genres — elle améliore significativement la cohérence et la pertinence des suggestions générées par l'algorithme.

Cette démarche a permis de surmonter les limites du dataset initial en offrant une vision plus riche et plus nuancée des préférences musicales. La manière dont ces données sont utilisées sera détaillée dans la Section [5.2](#).

2.2.3 Clustering et préparation finale des données

Le clustering a été réalisé en regroupant les colonnes par thématiques pertinentes. Les colonnes allant de dating à feelings ont été considérées comme des sujets (*topics*) abordés dans les chansons et ont été combinées en un cluster relatif aux topics. De même, les colonnes

allant de *danceability* à *energy*, représentant les caractéristiques audio (*features*), ont été regroupées pour former un cluster distinct correspondant aux *features* musicales.

Ce regroupement permet de mieux structurer les données en distinguant le contenu sémantique des chansons de leurs propriétés acoustiques. Les *topics* reflètent les thèmes abordés dans les paroles, tels que les émotions, les relations ou les valeurs, tandis que les *features* capturent des aspects mesurables comme le rythme ou l'énergie sonore. Cette séparation facilite l'analyse et rend plus efficaces les calculs de similarité, en permettant de travailler sur chaque cluster de manière indépendante.

Pour simplifier les manipulations et optimiser le temps d'accès, les colonnes nécessaires pour les calculs et l'algorithme ont été extraites et enregistrées dans un nouveau fichier. Ce fichier réduit contient uniquement les informations essentielles : *track_id*, *artist_id*, *release_date*, *genre*, ainsi que les trois principaux sujets (*top 3 topics*) parmi les 16 possibles et les trois principales caractéristiques audio (*top 3 features*) parmi les 6 disponibles. Cette réduction a permis de simplifier l'exploitation du dataset, qui comptait à l'origine plus de 30 colonnes.

Les raisons justifiant le choix de ces colonnes, directement liées aux critères retenus pour l'analyse, seront détaillées dans le Chapitre 3.

Chapitre 3

Choix et Organisation des Critères

L'algorithme de recommandation présenté dans le Chapitre 4, repose sur un ensemble de critères visant à évaluer la similarité entre les chansons et à générer des recommandations pertinentes. Ici, nous présentons les critères retenus pour cette approche, leur cohérence dans le cadre de notre modèle, ainsi que la manière dont nous avons organisé et hiérarchisé ces critères pour optimiser les résultats des recommandations.

3.1 Critères retenus

Le choix des colonnes retenues dans le fichier optimisé repose sur leur pertinence pour capturer les similitudes entre les morceaux et sur leur utilité dans les calculs de recommandation. Nous avons sélectionné cinq critères qui permettent de définir une représentation compacte et pertinente des morceaux de musique. Ces critères sont les suivants :

- `artist_id` : Identifiant unique de l'artiste, permettant d'analyser les préférences liées à un artiste spécifique.
- `release_date` : Date de sortie de la chanson, introduisant un aspect temporel pour identifier des tendances musicales.
- `genre` : Identifiants des genres musicaux, essentiel pour regrouper les morceaux selon leurs styles.
- `topics` : Trois des 16 thèmes dominants associés à chaque chanson, capturant les sujets abordés dans les paroles des morceaux.
- `features` : Trois des 6 caractéristiques audio les plus dominantes, représentant des aspects sonores et rythmiques de la musique.

Ces cinq critères ont été retenus car ils capturent les informations essentielles pour établir des liens significatifs entre les morceaux de musique. Le `artist_id` permet de lier les morceaux à des préférences d'artistes, le `release_date` ajoute une dimension temporelle qui est cruciale pour suivre l'évolution des tendances musicales. Le `genre` est essentiel pour regrouper les morceaux par style, tandis que les `topics` et `features` fournissent des informations détaillées sur les thèmes abordés et les caractéristiques audio des chansons. La réduction à ces cinq critères garantit une représentation suffisamment précise des morceaux, tout en simplifiant les calculs et en optimisant le temps d'accès aux données.

3.2 Cohérence des critères

Les cinq critères sélectionnés (`artist_id`, `release_date`, `genre`, `topics`, `features`) respectent les propriétés fondamentales assurant la cohérence du modèle de recommandation. Ces propriétés garantissent la robustesse et la pertinence des comparaisons entre morceaux.

Exhaustivité : L'ensemble des cinq critères est suffisant pour comparer tout couple de morceaux sans perte d'information. Pour deux morceaux a et b , si tous les critères sont égaux, c'est-à-dire $g_j(a) = g_j(b)$, $\forall j \in F$, alors les préférences entre a et b restent cohérentes dans toute situation de comparaison. Cela assure que chaque morceau est évalué de manière complète selon les dimensions essentielles.

Cohésion : Les critères sont également cohérents entre les préférences locales (sur chaque critère) et la préférence globale. Si un morceau a^+ est amélioré sur tous les critères par rapport à un morceau a et qu'inversement a^- est dégradé, alors a^+ doit naturellement être préféré à a^- :

$$g_j(a^+) \geq g_j(a) \geq g_j(a^-), \forall j \in F \implies a^+ \succ a^-.$$

Cela garantit que l'amélioration d'un morceau sur plusieurs critères se traduit par une meilleure recommandation globale.

Non-redondance : Chaque critère apporte une information unique et indispensable. La suppression de l'un d'entre eux remettrait en cause l'exhaustivité ou la cohésion du modèle. Par exemple, retirer `genre` réduirait la capacité à distinguer des morceaux de styles différents, et négliger les `topics` limiterait la prise en compte des contenus thématiques. Ainsi, tous les critères contribuent de manière essentielle à la qualité des recommandations.

En respectant ces trois propriétés, les cinq critères retenus assurent une base solide et cohérente pour comparer et recommander efficacement les morceaux de musique.

3.3 Hiérarchie des critères

Pour affiner la qualité des recommandations musicales, il a été essentiel d'établir une hiérarchie des critères en fonction de leur importance perçue dans le processus de suggestion. Cette hiérarchisation repose sur plusieurs sources d'analyse : des recherches approfondies dans la littérature scientifique (Stetler (2022), Zangerle et al. (2019) et Saragih (2023)), des échanges avec des auditeurs de musique issus de notre entourage personnel et universitaire, ainsi que notre propre réflexion sur les mécanismes de recommandation musicale.

L'ordre de priorité des critères a été défini comme suit :

1. **Genre (`genre`)** : Les genres musicaux sont apparus comme le critère dominant dans les préférences des utilisateurs. Ils constituent généralement le premier filtre lorsqu'un auditeur recherche de nouvelles chansons, car ils délimitent immédiatement un univers sonore spécifique (ex. : jazz, pop, rock).
2. **Caractéristiques audio (`features`)** : Les aspects sonores des morceaux, tels que la danseabilité, l'énergie ou la valence, influencent directement l'expérience d'écoute. Ces caractéristiques permettent de capturer des nuances musicales qui dépassent les simples classifications par genre, offrant ainsi des recommandations plus fines et adaptées aux préférences rythmiques et émotionnelles des utilisateurs.

3. **Artistes (artist_id) et Sujets (topics)** : Ces deux critères sont jugés d'importance équivalente. Les utilisateurs peuvent être attachés à certains artistes pour leur style ou leur interprétation, tandis que les sujets abordés dans les paroles influencent également les préférences d'écoute. Ces deux aspects contribuent à enrichir la pertinence des recommandations.
4. **Date de sortie (release_date)** : Ce critère a été classé en dernier. Bien que la nouveauté ou la nostalgie puissent influencer certaines écoutes, la date de sortie ne détermine pas directement la similarité musicale entre deux morceaux. Son rôle est principalement contextuel, par exemple pour recommander des chansons récentes ou des classiques.

Cette hiérarchie permet de guider les algorithmes de recommandation en mettant davantage l'accent sur les critères les plus déterminants dans la perception des utilisateurs. Elle reflète un équilibre entre les tendances observées dans la littérature scientifique et les préférences exprimées par des auditeurs réels.

Chapitre 4

Algorithme de Recommandation

L'élaboration d'un système de recommandation efficace repose sur le choix d'un algorithme capable de combiner intelligemment plusieurs critères de similarité. Différentes méthodes ont été explorées et comparées afin d'identifier celle offrant les résultats les plus pertinents et équilibrés. Cette analyse a conduit à la sélection d'une approche optimale, garantissant des recommandations musicales adaptées aux préférences de l'utilisateur.

4.1 Algorithmes testés

Plusieurs algorithmes ont été évalués afin de déterminer leur capacité à générer des recommandations musicales pertinentes. Chacun de ces algorithmes présente des mécanismes de décision distincts pour combiner différents critères de similarité.

4.1.1 Agrégation lexicographique

Dans un premier temps, l'approche d'agrégation lexicographique qui est non compensatoire a été envisagée. Cette méthode semblait la plus adaptée au regard des notions étudiées en cours, notamment parce que nous disposions d'un ordre d'importance clair pour les critères de recommandation. Ce type d'algorithme permet de discriminer les suggestions en se basant sur les premiers critères les plus importants. Si les k premiers critères suffisent à départager les choix, les $n - k$ critères restants peuvent être négligés.

Cependant, la mise en œuvre de cette approche s'est révélée inefficace, en partie à cause des imperfections du dataset. Par exemple, pour le critère `artist_id`, il était impossible de maximiser les valeurs de manière cohérente. L'ordre des identifiants des artistes ne suivait aucune logique : un `artist_id` lié au genre pop pouvait être immédiatement suivi par un autre associé au country, sans relation pertinente entre eux. Cette absence de structure rendait difficile la comparaison directe des artistes. De plus, de nombreux artistes étaient associés à plusieurs genres musicaux, brouillant davantage les correspondances.

Pour pallier ces incohérences, des conditions supplémentaires auraient été nécessaires, ce qui aurait alourdi le code et potentiellement réduit son efficacité. Ce problème ne se limitait pas au critère `artist_id`. Les critères `topics` et `features` posaient également des difficultés. En sélectionnant les trois valeurs dominantes, nous avons constaté que des chansons pouvaient partager les mêmes types de `topics` ou de `features`, mais avec des valeurs très éloignées. Par exemple, une chanson avec des valeurs $[0.9, 0.8, 0.7]$ pouvait être jugée plus proche d'une autre avec $[0.6, 0.5, 0.4]$, malgré des écarts significatifs, alors qu'une chanson avec des valeurs $[0.9, 0.8, 0.75]$ et partageant deux `topics` identiques aurait été plus pertinente.

Nous avons alors envisagé de ne plus imposer le même ordre pour les topics et features afin de privilégier les similitudes de types, mais les écarts de valeurs persistaient, compromettant la qualité des recommandations. Même en ajoutant des conditions pour limiter les transitions incohérentes entre genres, les résultats restaient insatisfaisants.

Le problème majeur provenait du dataset lui-même. Bien qu'il soit le plus adapté parmi ceux disponibles pour ce problème de décision multicritère, il comportait des incohérences internes. Certaines chansons se voyaient attribuer des valeurs élevées pour des features ou topics spécifiques à un genre différent du leur. De plus, certaines valeurs attribuées ne correspondaient pas à la réalité : des chansons joyeuses affichaient parfois des niveaux de sadness très élevés.

En raison de ces limitations et d'autres difficultés rencontrées avec ce dataset, l'approche par agrégation lexicographique a été abandonnée. Ces problèmes ont également restreint nos choix d'algorithmes, affectant la performance de toute méthode testée.

4.1.2 Méthode de Condorcet

Dans un second temps, la Méthode de Condorcet (règle de la majorité), également non-compensatoire, a été testée. Des poids w_j ont été attribués selon la hiérarchie définie : genre = 5, features = 4, topics = 3, artist = 3, et date de sortie = 2. Initialement, pour simplifier le problème, seules les chansons du même genre et de la même année de sortie que la chanson donnée ont été considérées. Parmi elles, celles partageant les mêmes top 3 topics et top 3 features, même dans un ordre différent, ont été comparées.

Pour évaluer la préférence entre deux chansons, le nombre de critères les plus proches de ceux de la chanson d'origine était comptabilisé. Cependant, cette approche a rapidement révélé des intransitivités (paradoxe de Condorcet). Par exemple, considérons trois chansons A , B et C qui partagent toutes le même genre et la même date de sortie :

- A partage les mêmes top 3 features mais seulement deux topics avec la chanson de référence.
- B partage les mêmes top 3 topics mais seulement deux features.
- C partage exactement deux features et deux topics.

Selon la pondération :

- $A \succ B$ car les features ont un poids plus élevé que les topics.
- $B \succ C$ car B correspond mieux aux topics que C .
- Cependant, $C \succ A$ car la combinaison de deux features et deux topics est globalement plus équilibrée que la seule correspondance complète sur les features de A .

Ce cycle de préférences incohérent ($A \succ B$, $B \succ C$, mais $C \succ A$) illustre le paradoxe de Condorcet. Cette intransitivité complique la création d'un classement cohérent des chansons, réduisant l'efficacité de cette méthode pour générer des recommandations stables et pertinentes.

Face à ces incohérences, il a été conclu que cette méthode n'était pas adaptée. Si des problèmes d'intransitivité apparaissent déjà dans un cas simplifié (même année de sortie), leur impact ne pourrait qu'empirer dans un contexte plus complexe.

4.2 Algorithme choisi

Après avoir exploré plusieurs méthodes de recommandation, il est apparu pertinent d'adopter une approche fondée sur la somme pondérée.

4.2.1 Somme pondérée

Bien que cette méthode soit connue pour son caractère compensatoire (c'est-à-dire que des similarités faibles dans certains critères peuvent être compensées par des similarités fortes dans d'autres), cet inconvénient ne s'est pas avéré problématique dans notre cas. Les poids attribués à chaque critère ont été fixés de manière réfléchie et justifiée, ce qui a permis d'obtenir des résultats cohérents et pertinents sans nécessiter d'ajustements supplémentaires.

Nous avons choisi de ne pas adapter les algorithmes cités dans la Section 4.1 pour chaque critère de manière trop complexe, car une telle approche aurait introduit une surcharge et une variabilité qui pourraient diminuer la stabilité du système de recommandation. À la place, nous avons opté pour l'approche la plus populaire et éprouvée de somme pondérée, car elle offre une simplicité et une efficacité remarquables tout en restant flexible.

Avant de calculer la similarité pondérée entre les chansons, nous définissons un ensemble de poids pour chaque critère. Ces poids permettent de prioriser certains critères par rapport à d'autres en fonction de leur importance dans le processus de recommandation. Les poids sont ajustables, ce qui permet de tester différentes configurations et de déterminer l'impact relatif de chaque critère.

Voici les poids attribués à chaque critère dans notre approche :

$$w = \begin{cases} w_{\text{artiste}} : 3 \\ w_{\text{genre}} : 6 \\ w_{\text{temps}} : 2 \\ w_{\text{topics}} : 3 \\ w_{\text{features}} : 5 \end{cases}$$

Ces poids sont ensuite utilisés pour calculer une somme pondérée des similarités entre les chansons. Le calcul de cette somme pondérée est effectué en multipliant chaque mesure de similarité par son poids respectif, ce qui permet de déterminer un score global de similarité entre deux chansons. Ce score global est ensuite utilisé pour établir les recommandations.

Chaque chanson u est évaluée sur chaque critère i à l'aide de la fonction sim_i , qui génère une évaluation $g_i(u)$:

$$g_i(u) = sim_i(input, u)$$

où $input$ désigne la chanson donnée (le problème de décision étant de trouver les chansons similaires à cette chanson donnée) et u représente une autre chanson du dataset. La fonction sim_i évalue la chanson u selon le critère i (tel que l'artiste, le genre, la date de sortie, les thèmes abordés, ou les caractéristiques audio comme la danseabilité ou l'énergie). Ce vecteur $g_i(u)$ quantifie la similarité de u par rapport à la chanson donnée selon ce critère.

L'évaluation globale de chaque chanson u est ensuite obtenue par la somme pondérée des évaluations de tous les critères. Le vecteur de préférence w_i est utilisé pour pondérer chaque critère i :

$$S(u) = \sum_{i=1}^n w_i \cdot g_i(u)$$

où w_i est le vecteur de préférence pour le critère i , et $g_i(u)$ est l'évaluation de la chanson u selon le critère i . Ce score global $S(u)$ permet de mesurer la pertinence de la chanson u par rapport à la chanson donnée.

Pour comparer deux chansons u et v , on vérifie leurs scores globaux :

$$S(u) = \sum_{i=1}^n w_i \cdot g_i(u) \quad \text{et} \quad S(v) = \sum_{i=1}^n w_i \cdot g_i(v)$$

Cette mesure permet de classer les chansons en fonction de leurs score, où un score plus grand indique une meilleure similarité et justifie un meilleur classement de la chanson dans la liste de recommandations.

Cette approche combine plusieurs critères de similarité entre les chansons :

- **Similarité des artistes (artist_id)** : La similarité entre deux chansons dépend de l'égalité de leurs artistes. Si les artistes sont identiques, la similarité est maximale avec une valeur de 0,5, sinon elle est nulle. Cette valeur est pondérée par le poids w_a attribué à ce critère.

$$sim_{\text{artiste}} = \begin{cases} 0,5 & \text{si } A_u = A_v \\ 0 & \text{sinon} \end{cases}$$

où A_u et A_v désignent les artistes des chansons u et v .

- **Similarité des genres musicaux (genre)** : La similarité entre deux chansons est évaluée par l'intersection de leurs genres respectifs. Si les ensembles de genres sont de même taille, la similarité est maximale. Sinon, elle est ajustée pour pénaliser les ensembles plus grands. Cette méthode offre une évaluation équilibrée de la proximité des genres.

La similarité des genres est donnée par :

$$sim_{\text{genre}} = \begin{cases} |G_u \cap G_v| & \text{si } |G_u| = |G_v| \\ |G_u \cap G_v| - 0,5 \cdot (\max(|G_u|, |G_v|) - |G_u \cap G_v|) & \text{sinon} \end{cases}$$

où G_u et G_v sont les ensembles des genres des chansons u et v , et $|G_u \cap G_v|$ représente le nombre de genres communs.

- **Similarité temporelle (release_date)** : La différence entre les dates de sortie des chansons est pénalisée par une fonction logistique décroissante, favorisant les chansons sorties à des périodes proches. La similarité temporelle est donnée par :

$$sim_{\text{temps}} = \frac{1}{1 + e^{\beta \cdot \Delta t}}$$

où $\Delta t = |d_u - d_v|$ est la différence absolue entre les dates de sortie d_u et d_v , et $\beta = 0,1$ contrôle la décroissance. Le poids associé est ajusté par $\log(\Delta t + 1)$ pour atténuer l'effet des écarts importants.

- **Similarité des sujets (topics)** : La similarité thématique entre deux chansons est basée sur l'intersection de leurs trois principaux sujets (top 3). Plus le nombre de sujets communs est élevé, plus la similarité est forte. Elle est définie par :

$$sim_{\text{topics}} = |T_u \cap T_v|$$

où T_u et T_v sont les ensembles des sujets des chansons u et v . Cette mesure met directement en valeur le nombre de thèmes partagés.

- **Similarité des caractéristiques sonores (features)** : La similarité entre les caractéristiques audio de deux chansons est calculée en comptant les attributs communs parmi les trois principales caractéristiques (top 3). Un bonus est ajouté si les caractéristiques correspondent dans le même ordre. La formule est la suivante :

$$sim_{\text{features}} = |F_u \cap F_v| + 0.5 \cdot \delta_1 + 0.5 \cdot \delta_2 + 1 \cdot \delta_3$$

où F_u et F_v sont les ensembles des caractéristiques sonores des chansons u et v , et $\delta_i = 1$ si la i^{e} caractéristique est identique et que la $(i - 1)^{\text{e}}$ caractéristique était également identique, sinon $\delta_i = 0$. Cette approche valorise les correspondances exactes tout en prenant en compte les similitudes générales.

Ces approches différenciées assurent une évaluation équilibrée et pertinente des similarités, en tenant compte des spécificités de chaque critère. Cette diversité méthodologique permet d'optimiser la qualité des recommandations musicales.

Une fois les similarités calculées selon les critères précédemment définis, le système de recommandation exploite ces résultats pour générer automatiquement une playlist cohérente et variée. À partir d'une chanson initiale, les morceaux les plus pertinents sont sélectionnés en fonction de leur score global de similarité. Ce processus itératif permet de constituer progressivement une playlist complète pouvant contenir jusqu'à 100 chansons, sans nécessiter de recalcul des poids à chaque étape. Cette approche assure une recommandation fluide et optimisée, tout en garantissant la diversité musicale et la cohérence globale de la playlist.

Chapitre 5

Résultats et Analyse

L'implémentation du système de recommandation repose sur l'exécution de l'algorithme permettant de générer des playlists basées sur des critères précis. Les résultats obtenus sont ensuite analysés pour évaluer la pertinence des recommandations.

5.1 Structure du code

Avant d'exécuter le système de recommandation, il est important de comprendre la structure du code. Le projet est organisé en deux répertoires principaux : `src` et `data`.

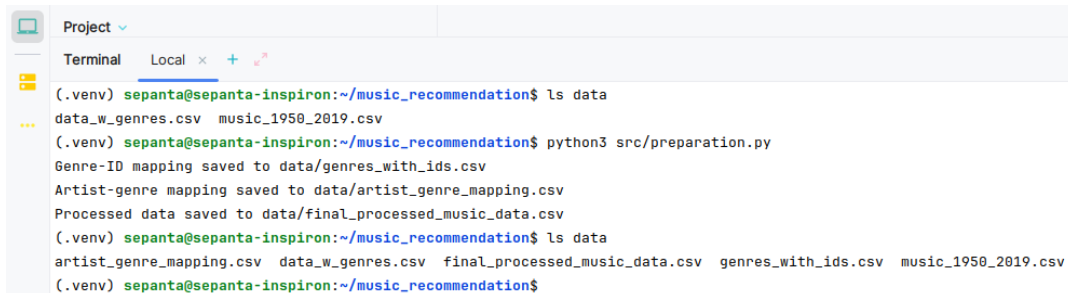
Au départ, les répertoires sont structurés comme suit :

- **Répertoire `src`** : Ce répertoire contient les fichiers Python nécessaires à la préparation des données et à l'exécution du système de recommandation. Il comprend les fichiers suivants :
 - `preparation.py` : Ce fichier est chargé de préparer et de générer les nouveaux fichiers CSV nécessaires au système de recommandation. Il prend en entrée les fichiers bruts et les transforme pour créer des ensembles de données propres à utiliser dans le processus de recommandation.
 - `recommendation.py` : Ce fichier contient l'algorithme de recommandation proprement dit. Il utilise les fichiers CSV générés par `preparation.py` pour calculer les similarités entre les chansons en fonction de plusieurs critères, puis génère une playlist cohérente en fonction des recommandations calculées.
- **Répertoire `data`** : Ce répertoire contient les fichiers de données nécessaires à l'algorithme de recommandation. Les fichiers présents dans ce répertoire sont les suivants :
 - `music_1950_2019.csv` : Ce fichier contient des informations sur un large éventail de chansons et d'artistes, selon [Shahane \(2019\)](#). Il fournit une base de données riche en informations sur la musique depuis 1950 jusqu'en 2019.
 - `data_w_genres.csv` : Ce fichier, tiré de [Mavani \(2023\)](#), contient des informations sur les genres musicaux associés à chaque chanson. Il est utilisé pour enrichir les critères de similarité, notamment la similarité des genres.

Dans la Section 5.2, pour mieux comprendre comment fonctionne le processus de recommandation, nous allons détailler ce processus étape par étape à travers un exemple concret. Nous observerons les nouveaux fichiers créés, ainsi que les différentes étapes du traitement, depuis la préparation des données jusqu'à la génération de la playlist recommandée.

5.2 Exécution avec un exemple pratique

Avant de commencer le processus de recommandation, il est nécessaire d'exécuter le fichier `preparation.py`¹.



```

Project
Terminal Local x +
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ ls data
data_w_genres.csv  music_1950_2019.csv
...
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ python3 src/preparation.py
Genre-ID mapping saved to data/genres_with_ids.csv
Artist-genre mapping saved to data/artist_genre_mapping.csv
Processed data saved to data/final_processed_music_data.csv
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ ls data
artist_genre_mapping.csv  data_w_genres.csv  final_processed_music_data.csv  genres_with_ids.csv  music_1950_2019.csv
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$

```

Comme illustré dans l'image ci-dessus, après l'exécution, trois nouveaux fichiers sont créés dans le répertoire `data`. Ces fichiers servent de base de données pour le système de recommandation et sont décrits ci-dessous :

- **genres_with_ids.csv** : Ce fichier contient une liste de tous les genres/sous-genres possibles trouvés dans le fichier `data_w_genres.csv`. À chaque genre est associé un identifiant unique. Au total, 2972 genres/sous-genres (identifiants) différents sont présents dans ce fichier.
- **artist_genre_mapping.csv** : Ce fichier établit une association entre les genres présents dans `genres_with_ids.csv` et les artistes du fichier `data_w_genres.csv`. Pour chaque artiste, on y trouve les genres qu'il a déjà interprétés dans ses chansons. Ce fichier permet d'enrichir les informations de genre pour chaque chanson et artiste.
- **final_processed_music_data.csv** : Ce fichier est le seul utilisé par `recommendation.py` pour générer des recommandations de chansons (playlists). Il est basé sur le fichier `music_1950_2019.csv`, mais a été modifié pour inclure plusieurs enrichissements.
 - Chaque artiste a maintenant un identifiant unique.
 - Chaque chanson a également un identifiant unique.
 - Les genres des artistes ont été repris à partir du fichier `artist_genre_mapping.csv`. Pour les artistes absents de ce fichier, les genres originaux présents dans `music_1950_2019.csv` ont été conservés.
 - Ce fichier contient également les trois principaux sujets (`topics`) associés à chaque chanson, ainsi que les trois principales caractéristiques sonores (`features`), avec leurs valeurs associées.

Ces trois fichiers permettent au système de recommandation de fonctionner avec des données plus structurées et enrichies, facilitant ainsi la génération de playlists plus précises et pertinentes.

Après la création des fichiers mentionnés précédemment, voici un exemple des données présentes dans le fichier `final_processed_music_data.csv` pour la même chanson présentée dans la Table 2.1.

1. Vérifiez que `pandas` est bien installé. Si ce n'est pas le cas, consultez https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html.

artist_name	the rolling stones
artist_id	4375
track_name	sympathy for the devil
track_id	23723
release_date	1966
genre	{376, 617, 2340, 47}
topic_1	(2, 0.3116658473845667)
topic_2	(11, 0.2117005184888807)
topic_3	(9, 0.1889043307529329)
feature_1	(6, 0.8898864509081764)
feature_2	(5, 0.8773701566364386)
feature_3	(2, 0.6996641284003794)

Table 5.1 : Les données présentes pour la chanson *Sympathy for the Devil* dans `final_processed_music_data.csv`

Maintenant que les données ont été préparées, la phase de recommandation peut être lancée en exécutant le fichier `recommendation.py`. Lors de cette exécution, il est demandé de spécifier l’ID de la chanson (`track_id`) à partir de laquelle la playlist sera générée. Les ID des chansons sont accessibles dans le fichier `final_processed_music_data.csv`.

En reprenant l’exemple précédent avec la chanson *Sympathy for the Devil* des *Rolling Stones*, son `track_id` est 23723, comme mentionné dans la Table 5.1. En utilisant cet ID, le système de recommandation génère automatiquement un nouveau fichier CSV nommé `generated_playlist_from_rolling_stones_sympathy_for_the_devil.csv` dans le répertoire `data`.



```

(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ ls data
artist_genre_mapping.csv  data_w_genres.csv  final_processed_music_data.csv  genres_with_ids.csv  music_1950_2019.csv
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ python3 src/recommendation.py
Entrez l'ID du morceau de départ (entre 1 et 28372) : 23723
Playlist saved to data/generated_playlist_from_the_rolling_stones_sympathy_for_the_devil.csv
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$ ls data
artist_genre_mapping.csv  data_w_genres.csv  final_processed_music_data.csv  generated_playlist_from_the_rolling_stones_sympathy_for_the_devil.csv  genres_with_ids.csv  music_1950_2019.csv
(.venv) sepanta@sepanta-inspiron:~/music_recommendation$

```

Ce fichier contient la liste des 100 chansons recommandées, triées selon leur pertinence. Comme dans `final_processed_music_data.csv`, chaque chanson est accompagnée de ses informations détaillées : ID de la chanson, nom de la chanson, nom de l’artiste, genres musicaux, date de sortie, top 3 topics et top 3 features. De plus, ce fichier inclut une colonne supplémentaire : le score de similarité attribué à chaque chanson. Ce score est calculé par l’algorithme de recommandation et reflète le degré de similarité avec la chanson de départ. Les chansons sont ordonnées de manière décroissante selon ce score : la première chanson de la liste présente la plus forte similarité, tandis que les suivantes montrent une similarité décroissante.

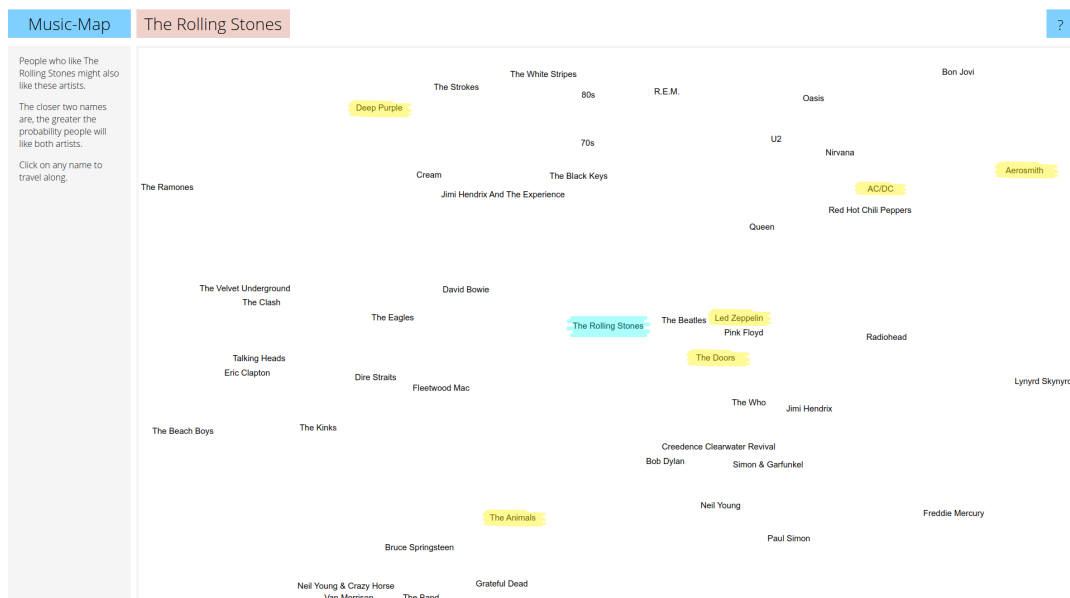
5.3 Analyse des résultats

Pour illustrer l’efficacité du système de recommandation, voici un aperçu des dix premières chansons générées dans la playlist recommandée (pour *Sympathy for the Devil*). Pour des raisons de simplicité, seules les informations essentielles sont présentées : le score de similarité, le nom de la chanson, l’ID de la chanson et le nom de l’artiste.

Score	Track ID	Artist	Song
	23723	the rolling stones	sympathy for the devil
50.33112673275636	24106	aerosmith	walk this way
49.72583194147707	24927	van halen	inside
47.33112673275636	24084	led zeppelin	trampled under foot
47.07139192074038	24447	ac/dc	shoot to thrill
46.092337717028514	16212	santana	back in black
44.29177458908532	23884	the doors	peace frog
44.0	13200	the animals	boom boom (bonus track)
43.72583194147707	24997	poison	look what the cat dragged in
43.57918735057182	25212	white lion	radar love
41.36239137933565	13681	deep purple	highball shooter

Pour évaluer la pertinence et l'optimalité des recommandations générées par notre algorithme, une comparaison a été réalisée à l'aide du site *Music Map*². Ce site permet d'obtenir une visualisation graphique des artistes les plus similaires à un artiste donné, en se basant sur des critères tels que les genres musicaux, les thématiques abordées et d'autres caractéristiques musicales.

En saisissant le nom de l'artiste *The Rollings Stones* sur *Music Map*, la plateforme a généré une cartographie des artistes jugés proches. Cette représentation offre un point de comparaison pertinent avec les résultats produits par notre système de recommandation.



Comme le montre cette cartographie, parmi les dix premières chansons de la playlist générée à partir de la chanson *Sympathy for the Devil* des *Rolling Stones*, 6 chansons sont interprétées par des artistes présents sur cette représentation. Plus précisément, 5 des 7 premières recommandations partagent des similarités avec des artistes figurant sur cette cartographie. Ce résultat témoigne de la pertinence de notre système de recommandation et de sa capacité à identifier des artistes proches musicalement.

2. URL du site *Music Map* : www.music-map.com

Ce constat met en avant la robustesse de l'algorithme, malgré certaines limitations liées aux données. En effet, si le dataset avait contenu des informations plus détaillées sur chaque chanson (par exemple, des caractéristiques musicales plus fines), les recommandations auraient pu gagner en précision et en cohérence.

De plus, certains artistes figurant sur la cartographie de *Music Map* ne sont pas présents dans notre dataset. Leur inclusion aurait probablement permis d'obtenir des recommandations encore plus pertinentes, voire parfaitement alignées avec les similarités musicales attendues. Ce résultat souligne le potentiel d'amélioration du système en enrichissant le jeu de données utilisé.

Les résultats obtenus sont cohérents avec les attentes. Les recommandations générées reflètent bien les similarités entre les chansons selon les critères sélectionnés, tels que l'artiste, les genres, les sujets et les caractéristiques audio. Malgré les limites des données, l'algorithme montre une bonne capacité à proposer des chansons pertinentes pour les utilisateurs.

Chapitre 6

Améliorations et Perspectives

Le système de recommandation développé a permis d'obtenir des résultats intéressants, mais plusieurs pistes d'amélioration peuvent être envisagées pour affiner les recommandations et mieux répondre aux attentes des utilisateurs. L'exploration de nouvelles approches et l'intégration de critères supplémentaires peuvent offrir des opportunités pour améliorer la pertinence et la personnalisation des playlists générées.

6.1 Contraintes et axes d'amélioration

Bien que l'algorithme de recommandation proposé ait montré des résultats satisfaisants, plusieurs limites liées aux données disponibles méritent d'être soulignées.

L'ajout de nouveaux critères, notamment plus variés ou basés sur l'interaction de l'utilisateur avec les chansons, aurait sans doute permis d'affiner les recommandations et de rendre le système plus pertinent. Par exemple, la prise en compte de la langue des morceaux aurait pu mieux cibler les préférences culturelles et linguistiques des utilisateurs, contribuant ainsi à des recommandations plus personnalisées. De plus, intégrer la durée des chansons comme critère aurait permis d'adapter les playlists à différents contextes d'écoute (courtes chansons pour des trajets rapides, morceaux plus longs pour des sessions de concentration ou de détente). Cependant, les données disponibles ne permettaient pas une telle personnalisation, et il n'a pas été possible d'explorer d'autres dimensions qui auraient pu enrichir le dataset.

Comme mentionné dans la Sous-section [2.2.2](#), pour une chanson donnée, tous les genres associés à son artiste ont été pris en compte, même si la chanson elle-même ne correspondait pas nécessairement à tous ces genres. De plus, le dataset d'origine attribuait un seul genre à chaque chanson, souvent peu spécifique. Il aurait été préférable d'avoir plusieurs genres/sous-genres plus précis directement liés à chaque morceau plutôt qu'à l'artiste. Cela aurait permis d'améliorer la cohérence des recommandations en affinant la classification musicale.

En outre, l'algorithme repose sur des critères statiques et ne prend pas en compte l'évolution dynamique des préférences des utilisateurs. Ainsi, le système pourrait bénéficier d'une prise en charge plus fluide des changements dans les goûts musicaux au fil du temps.

Enfin, l'approche actuelle favorise plutôt les chansons similaires aux goûts déjà identifiés, réduisant la découverte de nouveaux styles ou artistes. Un équilibre entre pertinence et exploration aurait pu enrichir l'expérience utilisateur.

6.2 Exploration d'approches alternatives

L'exploration de différentes méthodes d'optimisation et d'algorithmes de recommandation permet d'envisager des alternatives potentielles à notre approche actuelle. Ces méthodes, telles que les techniques de surclassement et les modèles basés sur KNN et le Machine Learning, offrent des perspectives intéressantes pour améliorer la précision et la personnalisation des recommandations musicales.

6.2.1 Méthodes de surclassement (ELECTRE III)

Étant donné que la problématique de recommandation musicale se rapproche du problème de rangement (P_γ), une approche naturelle à explorer est celle des méthodes de classement multicritères telles que la méthode ELECTRE III. Cette méthode est particulièrement adaptée lorsqu'il s'agit de comparer des alternatives sur la base de plusieurs critères et de déterminer un ordre de préférence.

ELECTRE III repose sur l'évaluation des matrices de concordance et de discordance pour effectuer des classements. Les matrices de concordance représentent la mesure dans laquelle un critère soutient une préférence entre deux alternatives, tandis que les matrices de discordance indiquent la mesure dans laquelle un critère contrevient à cette préférence. L'approche s'appuie sur des seuils pour décider si une alternative surclasse une autre en fonction des critères donnés.

Nous avons commencé à implémenter cette méthode en calculant les matrices de concordance et de discordance partielles et globales. Toutefois, étant donné que notre dataset contient plus de 28 372 chansons, le calcul de ces matrices est devenu rapidement intractable. Chaque matrice de concordance et de discordance nécessitait une évaluation pour chaque paire de chansons, ce qui entraînait une complexité considérable et des délais de calcul importants.

Au fur et à mesure de l'implémentation, nous avons observé que la version initiale de l'algorithme ELECTRE III donnait des résultats moins cohérents que l'approche hybride avec somme pondérée que nous avons choisie. Bien que la méthode ELECTRE III soit théoriquement intéressante pour un problème de classement, dans la pratique, les résultats obtenus avec notre implémentation ne correspondaient pas aux attentes en termes de précision et de pertinence des recommandations.

Après avoir confronté les résultats obtenus avec l'algorithme hybride à ceux générés par ELECTRE III, nous avons décidé de maintenir l'approche hybride pour les recommandations musicales. Cependant, nous avons réalisé qu'il serait pertinent d'explorer des améliorations sur notre modèle existant.

En particulier, l'implémentation d'une relation de surclassement pourrait être envisagée. En utilisant les critères existants, nous pourrions mieux exploiter les relations de surclassement et de sous-classification entre les chansons, en ajustant dynamiquement les pondérations des critères en fonction des retours utilisateurs et des préférences contextuelles. Cela permettrait de maintenir un équilibre entre la cohérence des recommandations et la diversité des suggestions proposées.

6.2.2 Exploration de KNN (K-Nearest Neighbors)

Une autre approche potentielle pour améliorer notre système de recommandation est l'utilisation de l'algorithme K-Nearest Neighbors (KNN)¹, largement utilisé dans les systèmes de

1. Pour plus d'informations sur KNN : <https://www.ibm.com/fr-fr/topics/knn>

recommandation, en particulier pour les tâches de classement et de prédiction. L'idée derrière KNN est simple : pour une chanson donnée, l'algorithme identifie ses K chansons les plus proches dans l'espace des caractéristiques (en termes de similarité), puis utilise ces chansons proches pour faire une recommandation.

Dans notre cas, étant donné que nous avons une riche représentation des chansons sous forme de différents critères, KNN pourrait être utilisé pour trouver des chansons similaires à celles qu'un utilisateur a déjà écoutées. Ces chansons similaires pourraient alors être proposées à l'utilisateur en fonction de ses préférences passées.

L'algorithme KNN repose sur une mesure de distance, souvent la distance Euclidienne ou la similarité cosinus, pour quantifier la proximité entre deux chansons. L'une des difficultés principales avec KNN, surtout pour un dataset volumineux comme le nôtre, est l'efficacité du calcul. Calculer la distance entre chaque paire de chansons peut devenir prohibitif en termes de temps de calcul, en particulier avec plus de 28 000 chansons dans notre dataset. Cela nécessite l'optimisation de l'algorithme pour accélérer la recherche des plus proches voisins.

Chapitre 7

Conclusion

Ce projet a permis de concevoir un algorithme de recommandation de chansons capable de proposer des suggestions personnalisées en s'appuyant sur une analyse fine des préférences musicales. Après avoir exploré plusieurs méthodes d'agrégation telles que l'agrégation lexicographique et la méthode de Condorcet, une approche basée sur la somme pondérée s'est imposée comme la solution la plus équilibrée et performante. Cette méthode a su combiner efficacement divers critères de similarité pour offrir des recommandations pertinentes et adaptées.

Les résultats obtenus témoignent de la capacité de cet algorithme à capter les nuances des goûts musicaux en exploitant les critères des morceaux tels que leurs caractéristiques sonores et les genres associés. L'algorithme permet de générer des playlists de 100 chansons, offrant une variété de morceaux en fonction des préférences spécifiques des utilisateurs. Néanmoins, ce travail ouvre la voie à de nombreuses perspectives d'amélioration. L'ajustement des pondérations, l'intégration de nouveaux critères ou encore l'utilisation de techniques plus avancées pourraient renforcer la qualité des recommandations.

Bien que d'autres alternatives aient été possibles, compte tenu du dataset disponible et des ajustements nécessaires, l'approche choisie s'est révélée être une solution cohérente et efficace. Ces expérimentations ont permis de démontrer que l'algorithme actuel représente une solution robuste et bien adaptée aux besoins de recommandation musicale. Les résultats obtenus semblent cohérents avec les attentes, et cette approche paraît prometteuse pour des développements futurs.

Ainsi, cette première étape dans la conception d'un système de recommandation intelligent constitue une base solide, offrant de nombreuses opportunités pour affiner et enrichir les recommandations, dans le but ultime de proposer une expérience musicale toujours plus immersive et personnalisée.

Le code source et d'autres détails sont disponibles sur le dépôt GitHub du projet ¹. Bien que toutes les approches testées n'aient pas été `commit`, le dépôt inclut les versions principales du code utilisées pour générer les recommandations.

1. URL du dépôt GitHub : https://github.com/sepanta007/Music_Recommendation

Bibliographie

- Mavani, V. (2023), 'Music recommendation system using spotify dataset'.
URL: <https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset>
- Saragih, H. S. (2023), 'Predicting song popularity based on spotify's audio features : insights from the indonesian streaming users', *Journal of Management Analytics* .
URL: <https://doi.org/10.1080/23270012.2023.2239824>
- Shahane, S. (2019), 'Music dataset : 1950 to 2019'.
URL: <https://www.kaggle.com/datasets/saurabhshahane/music-dataset-1950-to-2019>
- Stetler, R. (2022), 'Exploring music genres : A study of optimal differentiation by feature'.
URL: <https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset>
- Zangerle, E., Huber, R., Vötter, M. and Yang, Y.-H. (2019), 'Hit songprediction : Leveraging low- and high-level audio features'.
URL: <https://archives.ismir.net/ismir2019/paper/000037.pdf>