Rapport de Projet : Classification de Genres Musicaux sur des Données de Playlists

Elyes Khalfallah Nadia Achatoui

16 décembre 2024

Résumé

Ce rapport présente une étude de classification de genres musicaux à partir de données de playlists issues de Spotify. Les approches explorées combinent des données quantitatives (caractéristiques audio, popularité, etc.) et des informations textuelles (titres des morceaux) encodées via des méthodes de type Sentence-BERT. Nous discutons le protocole d'apprentissage, les résultats obtenus, ainsi que les améliorations possibles.

Table des matières

1 Introduction

Dans le cadre de ce projet, nous nous sommes intéressés à la classification du genre musical de pistes issues de playlists. Le jeu de données utilisé provient de Kaggle (ici) et contient un ensemble de caractéristiques audio (telles que danseabilité, énergie, valence, etc.) ainsi que des métadonnées (titre, nom de la playlist, etc.).

L'objectif est de prédire le genre de chaque piste en exploitant deux types d'informations :

— Des données quantitatives (caractéristiques sonores, popularité)
— Des données textuelles (titres des morceaux) encodées en vecteurs sémantiques.

Le rapport se décompose comme suit :

— Dans un premier temps, nous présentons le jeu de données, la tâche de classification et les premiers résultats obtenus.
— Nous détaillons ensuite l'architecture des modèles (réseaux de neurones) employés, les variantes envisagées, ainsi que le protocole d'apprentissage et d'évaluation.
— Nous présentons ensuite les résultats finaux de manière synthétique (tableaux de performance, figures montrant l'évolution de la loss et de l'accuracy, etc.) et discutons la qualité de l'apprentissage.
— Enfin, nous concluons sur l'étude réalisée et suggérons des améliorations futures.

2 Description de l'architecture du code

Le projet est organisé en deux modules Python principaux et trois notebooks Jupyter pour les expérimentations. Le notebook `partie1.ipynb` contient l'intégralité du code concernant la partie 1 du projet.

Le fichier `fonctions1.py` contient les fonctions utilitaires de base, telles que l'encodage one-hot, le calcul de la perte d'entropie croisée, et la définition de la classe `MLP`, un réseau de neurones multi-couches avec options pour la normalisation par lot, le dropout et le clipping des gradients. Ce fichier contient les fonctions concernant le notebook `partie2.ipynb`

Le module `fonctions2.py` étend ces fonctionnalités en intégrant le traitement des données textuelles. Il inclut des fonctions pour charger un modèle Sentence-BERT pré-entraîné et encoder les titres des morceaux en vecteurs numériques. Ce fichier définit également une classe `SimpleMLP` adaptée à l'utilisation des embeddings textuels seuls ou en combinaison avec des caractéristiques numériques. Ce fichier contient les fonctions concernant le notebook `partie3.ipynb`

Le notebook `partie3.ipynb` orchestre le pipeline complet : chargement et prétraitement des données, encodage des titres, combinaison des données

numériques et textuelles, entraînement des modèles, et évaluation des performances. Des visualisations des courbes d'apprentissage permettent d'analyser la convergence et la généralisation des modèles. Cette architecture modulaire assure une organisation claire et facilite les expérimentations et les ajustements nécessaires.

3 Présentation du jeu de données et de la tâche de classification

Le jeu de données contient un ensemble de pistes musicales accompagnées de leurs caractéristiques numériques et d'informations textuelles. Les caractéristiques numériques incluent notamment :

— Danceability, energy, valence, mode, key, tempo, loudness, etc.

— Popularité et durée du morceau.

Les informations textuelles (titres des morceaux) fournissent un signal sémantique complémentaire. Les genres de playlists (cibles) sont distribués selon la figure 1.

La tâche consiste à prédire le genre à partir des données fournies. Nous avons d'abord testé des modèles classiques (SVM, Random Forest, Régression Logistique, etc) sur les seules données quantitatives. Les performances obtenues (voir Table 1) montrent une précision de l'ordre de 50%, valeur qui peut être améliorée.

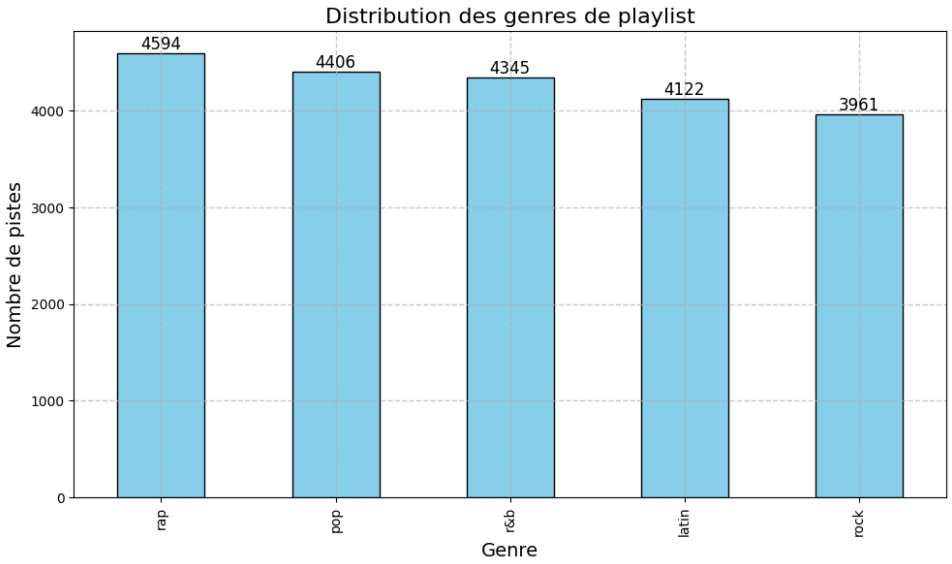

FIGURE 1 – Distribution des genres musicaux

TABLE 1 – Résultats préliminaires

Modèle	Accuracy (Val)
Random Forest (quantitatif seul)	57%
SVM (quantitatif seul)	51%
Régression Logistique (quantitatif seul)	50%
K-NN (quantitatif seul)	45%
Arbre de Décision (quantitatif seul)	44%

Par la suite, nous avons intégré les informations textuelles en encodant les titres des morceaux avec un modèle de type Sentence-BERT (*all-MiniLM-L6-v2*). Cette représentation textuelle de dimension d est combinée aux données quantitatives. L'ajout des données textuelles a permis d'améliorer les performances (voir section suivante).

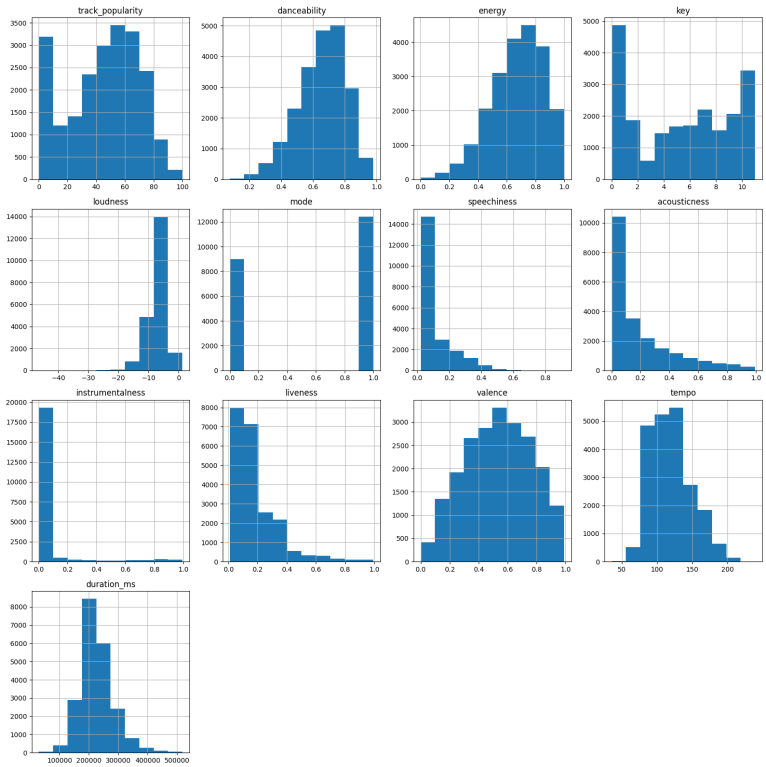

FIGURE 2 – Histogrammes des distributions des valeurs quantitatives dans le jeu de données (figure disponible dans le code)

Cette matrice de corrélation affiche les coefficients de corrélation entre les différentes variables numériques. Les valeurs vont de -1 (corrélation négative parfaite) à 1 (corrélation positive parfaite). On remarque des corré-

lations notables, par exemple "energy" et "loudness" (0.66), indiquant que des morceaux plus énergétiques tendent également à être plus forts. À l'inverse, "acousticness" et "energy" (-0.53) montrent une relation négative : des morceaux acoustiques sont généralement moins énergétiques. La plupart des autres variables présentent des corrélations faibles (proches de 0), ce qui suggère une faible dépendance entre elles. Globalement, les variables semblent relativement indépendantes, à l'exception de quelques relations modérées comme celles citées ci-dessus.

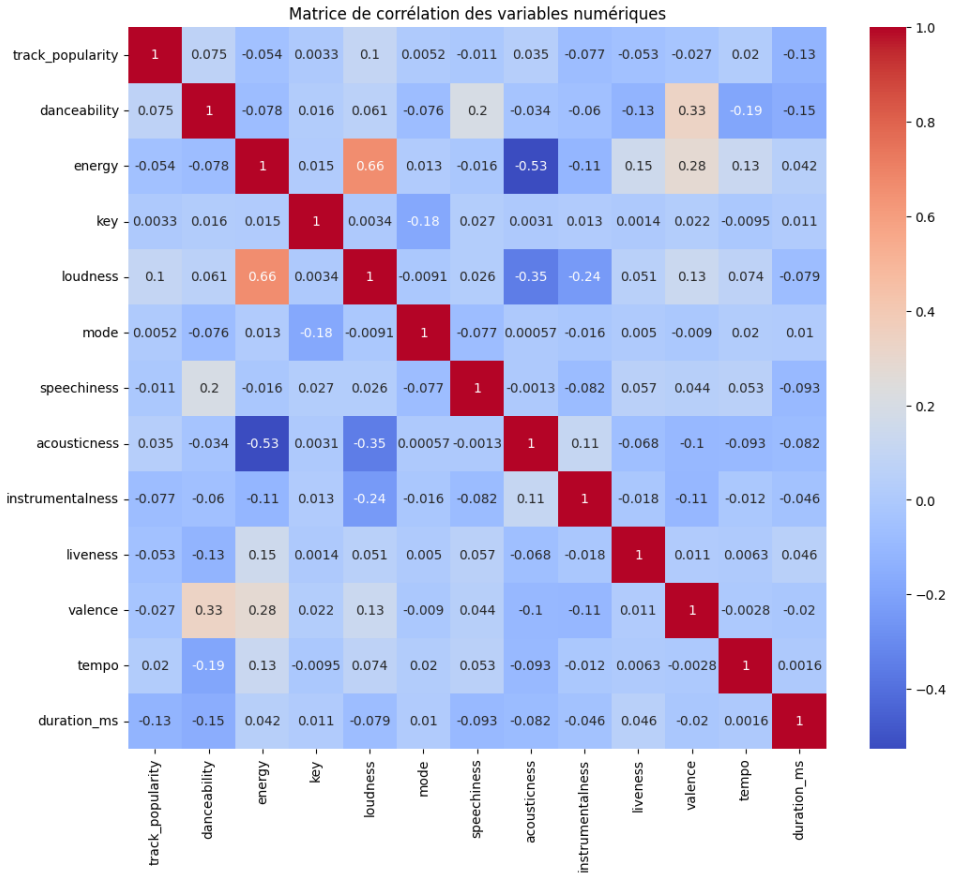

FIGURE 3 – Heatmap de corrélation des variables numériques

4 Architecture du réseau, protocole d'apprentissage et d'évaluation

Nous avons développé un réseau de neurones multi-couches (MLP) dont l'entrée est constituée soit des données quantitatives seules, soit des embeddings textuels seuls, puis la combinaison des deux. L'architecture de base (voir Figure 4) comprend :

— Une ou deux couches cachées ([512, 512], [128] neurones, etc.) avec fonction d'activation ReLU.
— Une couche de sortie softmax pour prédire la distribution sur les genres.

```python
# Architectures différentes a tester
layer_configs = [
    [],  # Pas de couche cachée
    [2],  # Une couche cachée avec 2 neurone
    [2, 4, 8, 16, 32, 64],  # etc
    # [2, 4, 8, 16, 32, 64, 32, 16, 8, 4, 2],  # etc
    [128],
    [512, 512],
    [64, 64, 64],
]

# Hyperparametres differentes a tester
batch_sizes = [32, 64, 256]
learning_rates = [0.01, 0.001, 0.0001]
dropout_rates = [0.0, 0.5]
use_batchnorm_options = [False, True]
clip_norm_values = [None, 1.0]

results = []  # pour stocker toutes les configurations testées
```

FIGURE 4 – Différentes architectures de MLP utilisées.

Les variantes envisagées incluent l'ajout de dropout, de batch normalization, ainsi que le taux d'apprentissage. Le protocole d'apprentissage consiste à séparer les données en ensembles *train*, *validation* et *test*, à normaliser les données quantitatives, et à entraîner le modèle sur un certain nombre d'époques (20 à 35), en monitorant la loss et l'accuracy sur l'ensemble de validation.

L'évaluation s'effectue principalement via l'accuracy, complétée par la visualisation des courbes d'apprentissage pour détecter d'éventuels problèmes de convergence ou de sur-apprentissage.

5 Résultats et analyse

Les résultats finaux montrent une amélioration des performances lorsque l'on combine les données quantitatives et textuelles. Malgré cela, les résultats restent comparables à l'application d'un SVM. Il est aussi important de noter que certaines données qualitatives contiennent intrinsèquement les genres musicaux en eux mêmes. Par exemple, la colonne "sous-genre de playlist" rend très facile de retrouver la donnée "genre" par la représentation vectorielle de son sens sémantique. Le tableau 2 résume les précisions obtenues sur le test set SANS utiliser les variables qualitatives évidentes, et le tableau 3 les précisions AVEC variables qualitatives évidentes :

TABLE 2 – Précision sur l'ensemble de test pour différents scénarios (SANS évidences, basé sur : `track_name`)

Modèle	Données utilisées	Accuracy (Test)
MLP (Quantitatif seul)	Quantitatif	55%
MLP (Texte seul)	Embeddings textuels	33%
MLP (Texte + Quantitatif)	Combinaison	58%

TABLE 3 – Précision sur l'ensemble de test pour différents scénarios (AVEC évidences, basé sur : `playlist_name`)

Modèle	Données utilisées	Accuracy (Test)
MLP (Quantitatif seul)	Quantitatif	55%
MLP (Texte seul)	Embeddings textuels	99%
MLP (Texte + Quantitatif)	Combinaison	99%

On constate que la combinaison des données permet de dépasser les performances obtenues avec les données quantitatives seules. Les figures de loss et d'accuracy (Figure 5) montrent une convergence satisfaisante dans la validation (quelques époques de plus aplatiront davantage ces courbes, mais ne changeront pas de manière significative les résultats), sans surapprentissage excessif.

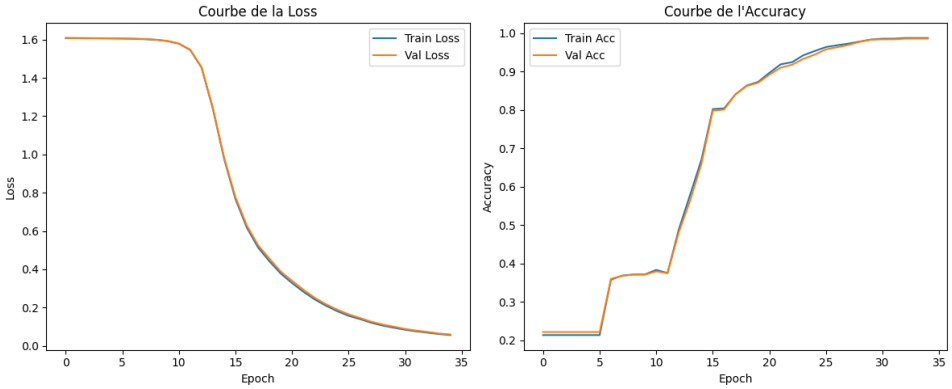

FIGURE 5 – Exemple de courbes d'apprentissage (loss et accuracy) sur l'entraînement et la validation.

6 Conclusion

Ce projet a permis de mettre en place une pipeline de classification de genres musicaux à partir de données de playlists. Après une première phase visant à exploiter les données quantitatives, nous avons intégré des informations textuelles via des embeddings de type Sentence-BERT. La combinaison des deux types de données a amélioré les résultats, démontrant une faible, mais existante complémentarité de l'information sémantique extraite des titres et des caractéristiques audio.

Les perspectives d'amélioration incluent l'exploration de réseaux plus complexes (CNN, LSTM pour d'autres types de données, Transformers), l'optimisation plus poussée des hyperparamètres, ou encore l'utilisation de techniques de régularisation pour limiter le sur-apprentissage. De plus, l'exploration d'autres types de représentations textuelles (modèles de langage plus récents) ou de features audio plus sophistiquées pourrait être envisagée.