

Ecole Publique d'Ingénieurs en 3 ans

Rapport de fin de projet – Projet 2A

ANALYSE DE DONNEES MUSICALES

15 avril 2021

MICHEL Nathan, ANCETTE Raphaël,
GUIGNOL Camille, RIO Arnaud

Tuteur école : **M. ROSENBERGER**



www.ensicaen.fr

TABLE DES MATIERES

Introduction	3
1. Présentation du projet	3
1.1. Présentation générale	3
1.2. Objectifs du projet	3
2. Outils et organisation	3
2.1. Outils et langages	3
2.2. Méthodologie	4
2.3. Sprints et répartition des tâches	4
3. Travail réalisé	5
4. Difficultés rencontrées	8
5. Bilan et améliorations futures	9
Conclusion	9

Introduction

D'après l'Institut National de la Statistique et des Etudes Economiques (INSEE), plus de 41 milliards de musiques ont été écoutées en flux (*streaming*) en France en 2020¹. Cela souligne l'importance de la musique pour les Français. Il serait donc intéressant d'obtenir des informations sur les musiques que l'on écoute et celles qui pourraient nous plaire. En fonction de notre humeur ou de nos écoutes récentes, nous pourrions découvrir des musiques appropriées à l'instant, et aussi de nouveaux genres ou de nouveaux artistes susceptibles de nous plaire, diversifiant ainsi notre playlist personnelle.

1. Présentation du projet

1.1. Présentation générale

Le but de notre projet est donc d'obtenir le genre d'une musique donnée à l'aide d'outils de fouille de données et d'apprentissage machine en se basant sur des données calculées et sa représentation sous forme d'image.

1.2. Objectifs du projet

Le premier objectif du projet a été d'implémenter différents algorithmes de classification de musique par genre. Pour cela, nous avons réalisé en amont un kick-off pour présenter notre projet au tuteur et voir si celui-ci est viable, ainsi qu'un état de l'art pour réunir ce qui a déjà été fait sur le sujet ou des sujets proches (algorithmes et jeux de données utilisés, résultats obtenus, ...).

Le but a été également de réaliser une application mobile capable de lire des musiques et qui simplifierait l'utilisation des algorithmes de classification afin de proposer des playlists adaptées à l'utilisateur. A terme, nous avons envisagé de mettre en place un système de recommandation basé sur les écoutes de l'utilisateur.

2. Outils et organisation

2.1. Outils et langages

Concernant la gestion de versions, nous avons utilisé Gitlab pour partager le code aux autres membres de l'équipe. En raison du contexte sanitaire, nous avons également utilisé principalement Discord pour la communication entre les différents acteurs du projet (cela comprend également le tuteur), pour demander de l'aide, pour faire des points réguliers ou pour informer les autres de l'avancée des tâches, aussi bien à l'écrit qu'à l'oral.

¹ Source : <https://www.insee.fr/fr/statistiques/4238548?sommaire=4238635>

Pour la réalisation technique du projet, nous pouvons la séparer en deux parties : d'une part l'apprentissage machine et l'intelligence artificielle, et d'autre part la conception de l'application mobile.

Pour la première partie, nous avons utilisé le langage Python et plusieurs bibliothèques dont les principales sont TensorFlow (intelligence artificielle) et Librosa (traitement de son), le tout sous l'environnement de développement PyCharm. De plus, nous avons utilisé le jeu de données GTZAN afin d'entraîner notre algorithme. Enfin, nous nous sommes servis du logiciel Orange pour tester les résultats d'autres algorithmes de manière plus pratique et graphique.

La seconde partie a été développée en Java sous l'environnement de développement Android Studio.

2.2. Méthodologie

Le développement s'est fait selon la méthode Agile où l'auto-organisation était essentielle. Ainsi, la communication entre les différents acteurs du projet fut primordiale. Nous nous réunissions régulièrement pour nous tenir au courant de l'avancée dans les tâches de chacun. De plus, nous faisons un point avec notre tuteur régulièrement et surtout au début du projet afin de nous conseiller et de valider l'avancement du projet et de nous éviter de mal orienter notre travail.

2.3. Sprints et répartition des tâches

Nous avons gardé la répartition définie dans le diagramme de Gantt au moment du kick-off, tout en gardant une certaine liberté et flexibilité. Nous avons également fait en sorte de répartir efficacement les tâches en fonction des points forts et des connaissances des technologies de chacun.

Dans le cadre d'une méthodologie agile, nous avons défini ci-dessous les différents sprints réalisés et les membres ayant travaillé sur ces tâches.

SPRINT**MEMBRE DE L'EQUIPE**

SEMESTRE 1	
Recherche	Tous les membres
Etat de l'art	Tous les membres
Nettoyage des données	MICHEL Nathan
Calcul des spectrogrammes	ANCETTE Raphaël
Sauvegarde et chargement	ANCETTE Raphaël, GUIGNOL Camille
Interface de l'application mobile	RIO Arnaud
Apprentissage (CNN ²)	GUIGNOL Camille
Lecture de musiques sur l'application mobile	RIO Arnaud

SEMESTRE 2

Nettoyage des données	MICHEL Nathan
Réduction du surapprentissage (CNN)	ANCETTE Raphaël, GUIGNOL Camille
Choix des activités	RIO Arnaud
Mise en place d'un k-NN ³ (avorté)	ANCETTE Raphaël, GUIGNOL Camille
Recherche de musiques sur le téléphone	RIO Arnaud
Augmentation du nombre de données	ANCETTE Raphaël
Test de différents algorithmes (Orange)	GUIGNOL Camille
Recherche sur l'utilisation des API	RIO Arnaud

3. Travail réalisé

Après deux semestres de travail, nous avons en grande partie atteint les objectifs que nous nous étions fixés.

² En apprentissage machine, un réseau neuronal convolutif (ou CNN) est un type de réseau de neurones artificiels dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Ces réseaux de neurones sont particulièrement efficaces dans la reconnaissance d'image et de vidéo.

³ La méthode des k plus proches voisins (ou k-NN) est une méthode d'apprentissage supervisé qui consiste à calculer la distance euclidienne entre plusieurs données afin de les classer.

Le premier semestre a surtout tourné autour de la recherche et du début du développement. Nous avons mis en place une première version de notre réseau neuronal convolutif (CNN) et de notre application mobile.

Les CNN étant particulièrement efficaces sur des images, nous avons représenté les musiques sous forme de spectrogrammes. Cependant, comme nous utilisons un grand nombre de données, le calcul de ces derniers était alors très chronophage. Ainsi, nous avons décidé de les sauvegarder grâce à la bibliothèque Pickle⁴ afin de pouvoir les réutiliser rapidement lors de l'apprentissage du CNN.

A l'issue de ce semestre, nous avons obtenu un CNN ayant une précision d'environ 50 à 60 % et souffrant d'un fort surapprentissage (voir figure ci-dessous).

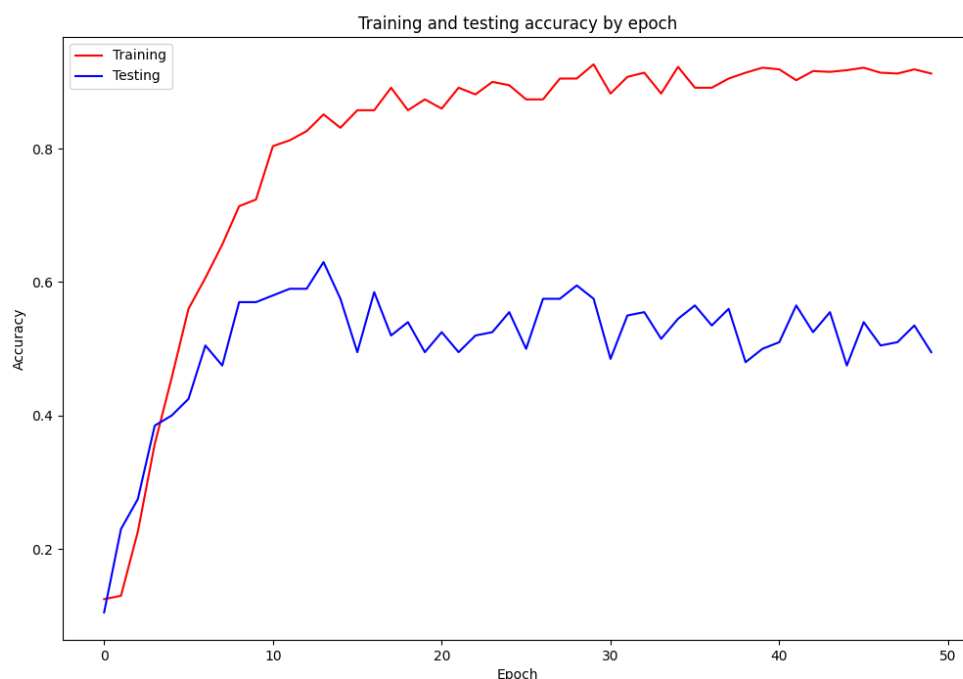


Figure 1 : Courbe d'évolution de la précision du 1er CNN par cycle d'apprentissage

Le souci que pose le surapprentissage est que le CNN devient très spécialisé sur un certain type de données. Ainsi, quand il rencontre une musique qu'il n'a jamais vu, il aura des difficultés à la classer. Cela explique l'écart si important entre les deux courbes : la courbe d'apprentissage en rouge a une très forte précision alors que la courbe de test en bleu est assez faible.

L'application mobile quant à elle permettait de pouvoir lire et écouter quelques musiques présentes sur le téléphone.

⁴ Pickle est un module de Python permettant de sauvegarder n'importe quel objet Python dans un fichier au format binaire et de récupérer leurs valeurs ultérieurement.

Nous avons pu ensuite poursuivre le projet sereinement avec pour objectif d'améliorer la base réalisée pour obtenir des résultats plus satisfaisants aussi bien sur l'aspect apprentissage machine que sur celui de l'application mobile.

Ainsi, les paramètres du CNN ont d'abord été modifiés dans le but d'éviter un surapprentissage. De plus, un CNN étant d'autant plus efficace avec un grand nombre de données, les musiques utilisées en entrée ont été découpées afin d'avoir plus d'échantillons et donc un apprentissage plus efficace et correct. Grâce à ce procédé, nous avons pu passer de 1000 musiques à 9000.

On a finalement obtenu une précision finale de 85 % sans surapprentissage (voir figure ci-dessous).

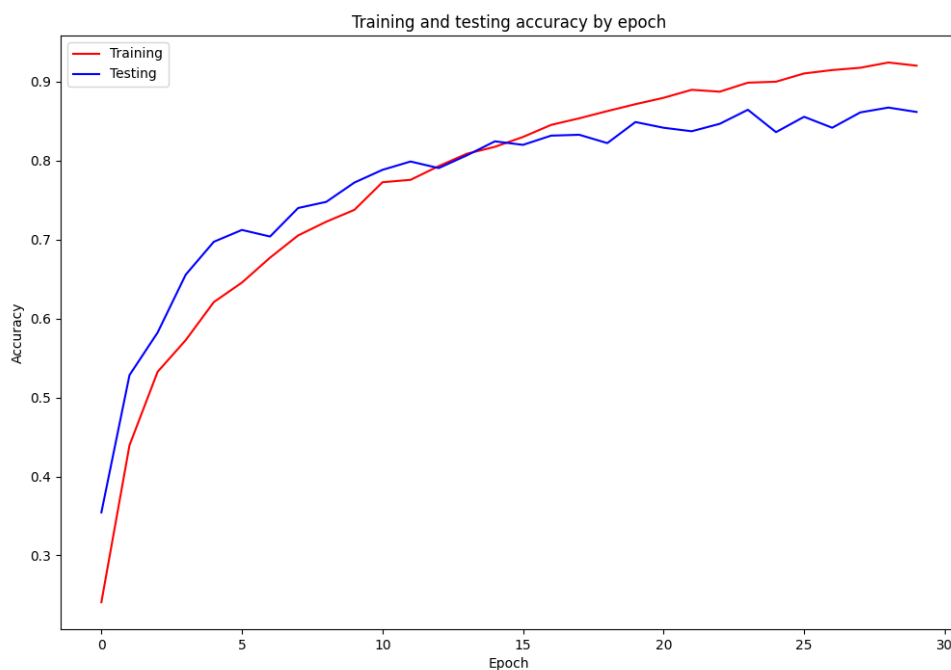


Figure 2 : Courbe d'évolution de la précision du 2nd CNN par cycle d'apprentissage

Enfin, d'autres algorithmes d'apprentissage machine ont été testés avec d'autres paramètres décrivant la musique tels que les paramètres MFCC, *Zero-Crossing Rate*, Centroïde Spectral, etc. (voir figure ci-dessous).

Evaluation Results					
Model	AUC	CA	F1	Precision	Recall
kNN	0.662	0.238	0.164	0.293	0.238
Tree	0.640	0.349	0.346	0.345	0.349
Stack	0.939	0.682	0.682	0.682	0.682
SVM	0.940	0.645	0.642	0.647	0.645
Neural Network	0.926	0.632	0.626	0.625	0.632
AdaBoost	0.853	0.497	0.486	0.483	0.497

Figure 3 : Précision des différents algorithmes d'apprentissage machine testés

Nous pouvons voir que ces algorithmes sont bien moins efficaces que notre CNN. C'est pour cela que nous avons décidé de ne garder que le CNN.

Enfin, le développement de l'application mobile s'est poursuivi en rajoutant différentes fonctionnalités telles que l'amélioration de l'interface et de l'ergonomie, la recherche de l'intégralité des musiques sur le téléphone et la recherche d'une musique particulière de l'utilisateur. La dernière étape a été d'essayer de relier celle-ci au code Python afin de classer en temps réel les différentes musiques écoutées par l'utilisateur.

4. Difficultés rencontrées

La première difficulté que nous avons rencontrée concerne l'utilisation des CNN. En effet, c'est un type de réseaux de neurones que nous n'avons pas approfondi en cours. Nous avons dû prendre du temps pour nous former à cette nouvelle technologie, ce qui nous a grandement retardé sur l'implémentation de la classification.

De plus, nous avons rencontré une deuxième difficulté sur l'algorithme à utiliser pour améliorer la précision. En effet, nous pensions que la mise en place d'un k-NN en sortie d'un CNN serait plus efficace qu'un perceptron⁵. Nous y avons passé beaucoup de temps pour finalement obtenir un résultat assez médiocre. Cette perte de temps est due à notre manque de communication avec le tuteur. En effet, si nous l'avions contacté dès le départ, il nous aurait réorienter vers une meilleure solution plus tôt.

La liaison entre l'application mobile et la classification par apprentissage machine nous a également posé quelques soucis. Après avoir réfléchi à plusieurs options qui se sont avérées difficilement réalisables et inefficaces (comme intégrer directement le code Python dans l'application), nous avons finalement conclu qu'il aurait été probablement plus judicieux de

⁵ Le perceptron est une méthode d'apprentissage supervisé qui à partir d'une entrée caractérisée par plusieurs paramètres associés à leur poids respectif donne en sortie une classification

passer par une API⁶ embarquant les outils d'intelligence artificielle que l'application mobile pourrait alors interroger.

Enfin, le nettoyage des données n'a pas pu être implémenté. En effet, travailler sur ce projet en parallèle des cours habituels, des examens et de nombreux projets, il est parfois difficile d'avancer autant qu'on le souhaiterait. Certaines périodes étaient très chargées en travail personnel, ce qui explique un avancement plus lent. L'implémentation du nettoyage des données aurait pu améliorer la précision du CNN en lui fournissant des données plus exactes.

5. Bilan et améliorations futures

Après ces deux semestres de travail, il est évident que le projet peut être encore amélioré.

Tout d'abord, nous pourrions améliorer davantage la précision de notre CNN en nettoyant les données et en augmentant drastiquement leur nombre. Il aurait également été possible de combiner d'autres outils afin d'obtenir une précision de classification encore plus élevée, même si celle résultante est déjà largement satisfaisante.

Ensuite, on pourrait chercher à améliorer l'application pour la rendre encore plus ergonomique. On pourrait également ajouter de nouvelles fonctionnalités comme un système d'évaluation par l'utilisateur de la classification proposée même si cela n'est pas l'objectif principal du projet.

À court terme, et dans la continuation de ce projet, il faudrait également mettre en place le lien entre l'algorithme de classification des musiques et l'application mobile. Cela nous permettrait d'implémenter et d'approfondir le système de génération de playlists adaptées à l'utilisateur en fonction du genre afin d'éviter que ce dernier n'écoute en boucle les mêmes morceaux d'une playlist qu'il aurait lui-même créée. Dans la même optique, on pourrait aussi permettre à l'utilisateur d'évaluer la pertinence des playlists proposés afin d'améliorer la création des celles-ci.

Conclusion

Pour conclure, notre groupe était plutôt harmonieux et hétéroclite. Chacun a su apporter sa pierre à l'édifice et nous avons réussi à tirer le meilleur de chacun pour faire avancer le projet et atteindre nos objectifs. Malgré quelques soucis de communication, cela nous permis de nous rendre compte de la réelle importance du travail d'équipe. Nous nous sommes beaucoup entraïdés, en travaillant à plusieurs pour résoudre le problème que rencontrait l'un d'entre

⁶ Une interface de programmation d'application (ou API) est une solution informatique permettant à des applications de communiquer entre elles et de s'échanger mutuellement des services et/ou des données.

nous. La technique du *pair programming*⁷ a beaucoup servi à corriger des erreurs lors du développement. Cependant, les difficultés rencontrées ont fait que nous avons dû revoir nos objectifs à la baisse.

⁷ Le *pair programming* est une méthode de travail dans laquelle deux développeurs travaillent ensemble sur un même poste de travail, une s'occupant de rédiger le code et l'autre de l'assister en décelant des erreurs et en suggérant des alternatives de développement.



Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04

