

Rapport de Projet en technique d'apprentissage artificiel :

Systeme de
recommandation de
playlist Spotify



Aaron Amani et Willhem Liban

Sommaire

- Introduction
- Problématique
- Méthodologie
- Conclusion

Introduction

Dans le cadre du module Technique d'Apprentissage Artificiel, nous devons réaliser un projet d'apprentissage automatique. Pour ce faire nous avons décidé de réaliser un système de recommandation.

Problématique

Comment proposer à l'utilisateur de la musique qu'il aime?

Pour répondre à cette problématique, on réalise ce système de recommandation.

Méthodologie

Création de dataset

Tout d'abord on va créer 2 playlist l'une contenant la musique "aimé" et l'autre contenant la playlist "pas aimé".

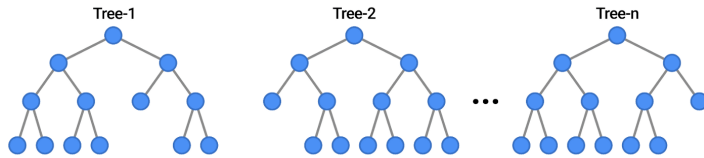
A l'aide de [l'api spotify](#) et de l'extension [spotipy](#) on récupère les 2 playlist. Tout d'abord on récupère les ID de chaque chansons à l'aide desquelles on récupère les données de chaque titre (tempo, duration, danceability...).

Ensuite on réunit ces 2 playlists dans un seul data frame en y ajoutant un champ "liked" qui permet de savoir si la chanson a été aimée ou non. Ce champ est donc la cible à prédire.

Entraînement

Parmi les données, ou features, que l'on récupère on détermine celle qui sont importantes. Pour cela on utilise l'algorithme Random Forest.

EXAMPLES

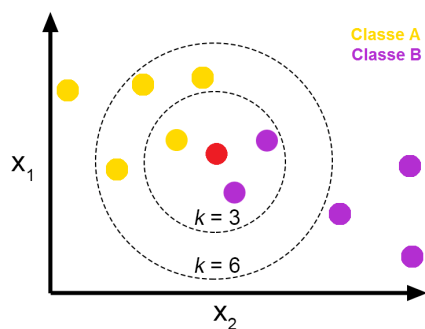


Les 3 features qui apparaissent en bas du classement sont “instrumentalness”, “mode” et “time_signature”. Nous n’avons donc pas utilisé ces données pour entraîner notre modèle.

```

1) liveness          0.156651
2) tempo             0.102514
3) loudness          0.100957
4) duration_ms       0.099219
5) danceability       0.098334
6) speechiness       0.083991
7) acousticness      0.082923
8) energy            0.080492
9) valence            0.080201
10) key              0.058677
11) instrumentalness  0.040053
12) mode             0.012251
13) time_signature   0.003736
  
```

Ensuite l’algorithme KNN permet de classifier. On entraîne le modèle sur le dataset créé. Le score du modèle est de 84%.



Test

Pour finir on récupère une autre playlist pour la faire passer dans le modèle entraîné. Et on retourne un pourcentage de sur la playlist représentant le nombre de “like” sur le nombre total de chansons.

```

Score du model : 0.965 .
Vous allez aimer 80/100 chansons.
Poucentage que vous aimez cette playlist est a 80.0 %.
Details des chanson aimé et ou pas ci-dessous :

```

	name	artist	liked
0	HOTEL LOBBY (Unc & Phew)	Quavo	1
1	God's Plan	Drake	1
2	Laugh Now Cry Later (feat. Lil Durk)	Drake	1
3	All Mine	Kanye West	1
4	XO Tour Llif3	Lil Uzi Vert	0
..
95	Faucet Failure	Ski Mask The Slump God	1
96	Hot N*gga	Bobby Shmurda	1
97	Splashin	Rich The Kid	1
98	PTSD	G Herbo	0
99	Wassup (feat. Future)	Lil Uzi Vert	1

```

Score du model : 0.965 .
Vous allez aimer 9/100 chansons.
Poucentage que vous aimez cette playlist est a 9.0 %.
Details des chanson aimé et ou pas ci-dessous :

```

	name	artist	liked
0	Numb	Linkin Park	0
1	In the End	Linkin Park	0
2	Chop Suey!	System Of A Down	0
3	Young and Lit	KAYGLIZZY	1
4	Bring Me To Life	Evanescence	0
..
95	The Resistance	Skillet	0
96	Square Hammer	Ghost	0
97	Fear of the Dark - 2015 Remaster	Iron Maiden	0
98	Your Betrayal	Bullet For My Valentine	0
99	Nobody's Fool	Cinderella	0

Conclusion

En perspective, on souhaite pouvoir créer des playlists pour l'utilisateur. On prendra en compte les artistes en utilisant la fonctionnalité related artist de l'api spotify qui permet de trouver artistes lié aux artist présent dans le dataset puis de passer par exemple les titres récents dans le modèle afin de déterminer si la musique va être aimée ou non et l'ajouter automatiquement dans une nouvelle playlist.