



Au fil des années, une multitude de chansons se sont placées en tête des classements musicaux les plus célèbres. On constate que chaque époque a été marquée par un courant musical.

Pour quelles raisons une chanson plaît-elle à un public précis ? Pourquoi certaines chansons ne rencontreront-elles jamais aucun succès avec ce même public ?

A notre époque la musique est principalement consommée sur en ligne et l'industrie musical investit de plus en plus pour comprendre les données des utilisateurs en ligne.

Cette grande quantité de données permet d'étudier l'impact d'un artiste ou d'une chanson et proposer des titres musicaux à tous les types d'utilisateurs.

Il est désormais possible d'extraire la probabilité qu'une chanson plaise à un utilisateur grâce à **SOUNDSGOOD**.

En lisant cet article, vous découvrirez comment **SOUNDSGOOD** a été conçu.

Étape 1 : le choix du Dataset

Nous avons sélectionné le dataset Spotify Song Attributes qui provient de l'API Spotify. Il contient des données sur les 2700 titres qu'un utilisateur a aimé ou non.

On retrouve dans ce dataset 16 colonnes différentes et 13 d'entre elles sont des attributs de la chanson. Il y a une colonne pour le nom de la chanson, une autre colonne pour le nom de l'artiste et une dernière colonne appelée "target" qui est le label de la chanson : chacune est labélisée avec un "1" lorsque l'utilisateur les aime et "0" pour celles qu'il n'aime pas.

Voici les 13 attributs de la chanson :

- *Tempo* indique le rythme de la chanson.
- *Energy* indique les chansons qui sont perçues comme rapides, fortes et bruyantes. Par exemple une chanson rock aura une note plus élevée sur l'échelle de l'énergie qu'une chanson classique.
- *Danceability* correspond à l'aptitude à danser sur le morceau de musique, plus la note est élevée, plus il est simple de danser sur cette chanson.
- *Loudness* correspond à l'intensité sonore de la chanson.
- *Liveness* indique la présence d'un public lors de l'enregistrement. Plus la note de vivacité est élevée, plus il y a des chances que la chanson ait été jouée en direct.
- *Instrumentalness* est la mesure qui définit l'utilisation d'instruments dans la musique.
- *Valence* indique la positivité ressentie lors de l'écoute de la chanson.

- *Duration_ms* correspond à la durée de la chanson.
- *Acousticness* indique si le son est acoustique.
- *Speechiness* détecte la présence de mots parlés dans une chanson : plus il est élevé et plus il y a de mots parlés.
- *Key* la clé musicale globale estimée de la chanson.
- *Mode* indique la modalité d'un morceau (majeure ou mineure) d'un morceau.
- *Time_signature* est une estimation de la signature rythmique globale (nombre de battements) d'une piste.

Étape 2 :Data Cleaning

Beaucoup de features ont des similitudes. Par exemple, *instrumentalness* est la mesure qui définit l'utilisation d'instrument dans la musique, tandis que la *speechiness* mesure l'utilisation de la voix comme nous l'avons évoqué précédemment.

Les deux features mesurent donc la musicalité d'une chanson.

Ainsi nous avons créé des catégories de features :

- Class Labels : *target*
- Interval : *acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence*
- Time : *duration_ms*
- Numerical : *tempo*
- Ordinal : *key, time_signature*
- Binary : *mode*
- String : *song_title & artist*

A l'aide de la méthode binning, nous allons créer de nouvelles features à partir des familles que nous avons créées précédemment.

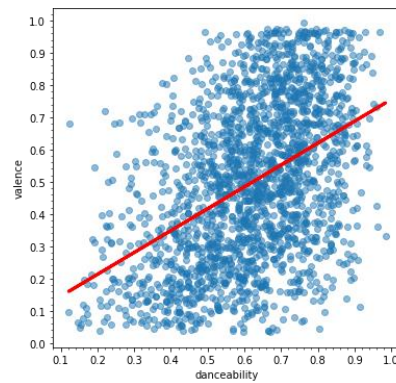
- instru_ind : L'instrumentale d'une musique et ses paroles
- beat_ind : L'énergie et le beat d'une musique
- melody_ind : Toute la mélodie d'une musique
- env_ind : L'ambiance d'une musique
- valence

Nous avons également décidé de nettoyer notre dataset afin de conserver uniquement les données nécessaires. Pour ce faire, nous avons remplacé les caractères string des colonnes *song_title*, *artiste* par la valeur nulle.

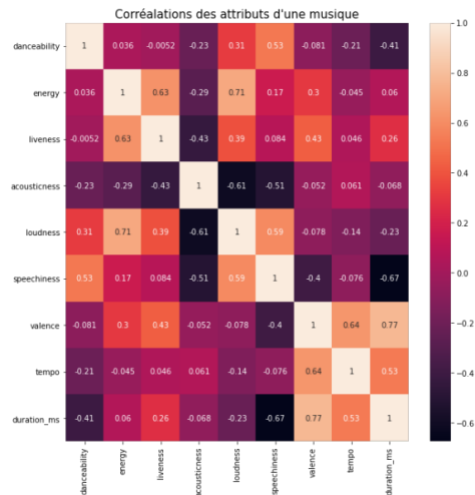
Étape 3 : Datavisualisation

Cette étape nous a permis d'avoir une vue d'ensemble sur nos données et de comprendre les corrélations existantes entre elles. Nous avons procédé à une analyse des features de manière graphique.

Corrélation entre la danceability et la mood d'une musique

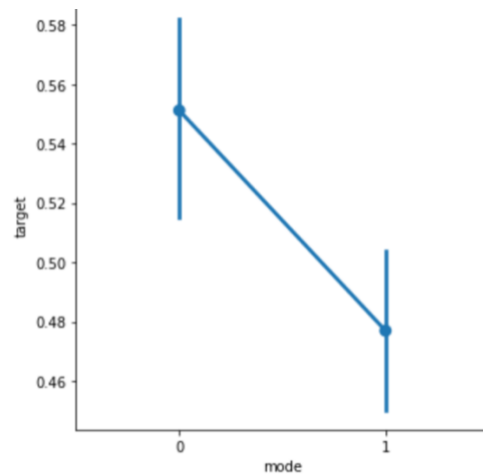


Grâce au graphe ci-dessus, on constate que plus la danceability est élevée et plus la valence l'est également : ces deux features sont donc corrélées.



La Heatmap ci-dessus montre la corrélation entre les attributs : il existe une forte relation linéaire entre la loudness et la valence par exemple.

Pour terminer, nous avons analysé la courbe représentative de target en fonction du mode.



On constate que plus le *mode* (qui indique la modalité d'un morceau (majeure ou mineure) d'un morceau) augmente et plus la *target* (qui indique si le user apprécie ou non la chanson) diminue.

Par la suite, il était important d'étudier la valeur de la *target* en fonction des valeurs de nos features. Nous avons créé des intervalles de données de valeurs pour chaque feature avec la valeur de target associée. Nous obtenons les résultats suivants :

target		tempo_band		target		loudness_band		liveness_band		target			
valence_band		0	(47.688, 90.727]	0.413043	(-33.13, -24.9]	0.090909	0	(0.0178, 0.256]	0.499348				
(0.0338, 0.274]	0.423246	1	(90.727, 133.595]	0.529218	(-24.9, -16.702]	0.220000	1	(0.256, 0.494]	0.517808				
(0.274, 0.513]	0.490196	2	(133.595, 176.463]	0.485323	(-16.702, -8.504]	0.673660	2	(0.494, 0.731]	0.510870				
(0.513, 0.753]	0.539405	3	(176.463, 219.331]	0.557377	(-8.504, -0.307]	0.470858	3	(0.731, 0.969]	0.692308				
(0.753, 0.992]	0.579365												
acoustic_band	target	energy_band		target		dance_band		target		speech_band		target	
0	(-0.000992, 0.249]	0	(0.0138, 0.261]	0.152381	0	(0.121, 0.338]	0.469565	0	(0.0223, 0.221]	0.481298			
1	(0.249, 0.498]	1	(0.261, 0.506]	0.583333	1	(0.338, 0.553]	0.377820	1	(0.221, 0.42]	0.740113			
2	(0.498, 0.746]	2	(0.506, 0.752]	0.555270	2	(0.553, 0.769]	0.506442	2	(0.42, 0.618]	0.650000			
3	(0.746, 0.995]	3	(0.752, 0.998]	0.480460	3	(0.769, 0.984]	0.703601	3	(0.618, 0.816]	0.500000			
instru_band	target												
0	(-0.000976, 0.244]	0.467635											
1	(0.244, 0.488]	0.724138											
2	(0.488, 0.732]	0.702128											
3	(0.732, 0.976]	0.644809											

Nous avons ainsi une vision des indicateurs qui permettent au score de la *target* d'augmenter et ceux qui le font diminuer.

Étape 4 : nos algorithmes de prédiction

Nous allons baser notre travail sur les trois modèles de prédiction suivants :

- Regression logistique qui est utilisée pour modéliser la probabilité de l'existence d'une certaine classe ou d'un certain événement. Dans notre étude, cet algorithme nous permettra de modéliser la probabilité qu'une chanson plaise à un utilisateur.
- Decision Tree Classifier utilise un arbre de décision pour passer des observations sur un élément aux conclusions sur la valeur cible de l'élément.
- K-NN est un algorithme dont le principe consiste en effet à choisir les **k** données les plus proches du point étudié afin d'en prédire sa valeur.

On obtient les résultats suivants :

Training test

1. Logistic Regression: 65.98
2. Decision Tree Classifier: 89.23
3. K-NN: 74.27

Result test

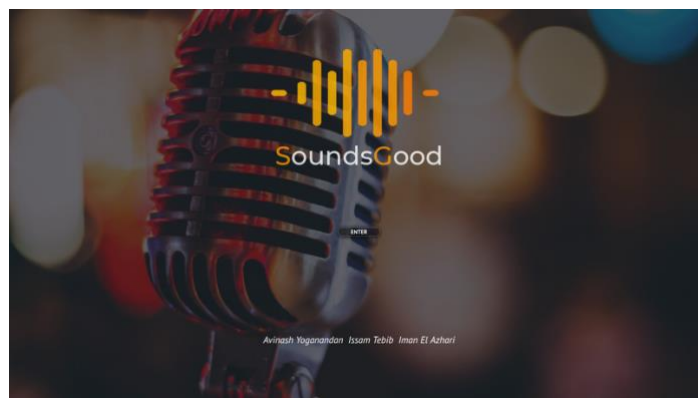
1. Logistic Regression: 61.39
2. Decision Tree: 61.06
3. K-NN: 61.22

K-Kross validation

1. Logistic Regression: 57.38
2. Decision Tree Classifier: 57.07
3. K-NN: 60.22

Étape 5 : Création de l'interface SoundsGood

Nous avons créé une interface ludique et simple d'utilisation à laquelle nous avons implémenté notre modèle de prédiction. Ainsi, nous avons réalisé une application en React que nous avons déployée sur Heroku.



Grâce à l'application SoundsGoods, vous pouvez en quelques clics connaître le potentiel d'une chanson à être appréciée par un utilisateur.

Tout d'abord il faut d'entrer le titre de la chanson de notre choix dans le champ dédié puis la plausibilité que la chanson plaise à notre utilisateur sera renvoyée en fonction des différents modèles de prédiction avec lesquels nous avons travaillé.

Rejoignez-nous vite sur SoundsGood !

