M1 Informatique AMU – TP4 ISD — 2023/24 — Rapport. Date : 14 novembre 2023.
ZEMMOURI Yasmine, Groupe 3 TP# 4.

TP4 - Pratique du Clustering

Partie 1 : Prise en main des données Unpopular Spotify Songs USS.

Etude des statistiques descriptives du jeu de données USS :

La méthode 'info()' de pandas présente des informations d'un DataFrame telles que le nombre total d'entrées, le nombre de valeurs non nulles, les types de données de chaque colonne et l'utilisation de la mémoire.

En appliquant cette méthode sur nos données, on aura les informations présentées sur la figure adjacente.

On peut voir que notre DataFrame contient 18 attributs, dont 9 de type réel (float), 4 de type entier (int), 4 de type objet et un booléen.
On peut également noter le nom de chaque attribut, son type et le nombre de valeurs non-nulles.

A partir de ces informations, on peut facilement constater quels attributs contiennent des valeurs manquantes : le compte de valeurs non nulles sera différent au nombre de lignes (d'examples) du DataFrame [dans ce cas, seul l'attribut 17 (Unnamed : 17) répond à cette contrainte).

```
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      4073 non-null   float64
 1   energy            4073 non-null   float64
 2   key               4073 non-null   int64
 3   loudness          4073 non-null   float64
 4   mode              4073 non-null   int64
 5   speechiness       4073 non-null   float64
 6   acousticness      4073 non-null   float64
 7   instrumentalness  4073 non-null   float64
 8   liveness          4073 non-null   float64
 9   valence           4073 non-null   float64
 10  tempo             4073 non-null   float64
 11  duration_ms       4073 non-null   int64
 12  explicit          4073 non-null   bool
 13  popularity        4073 non-null   int64
 14  track_name        4073 non-null   object
 15  track_artist      4073 non-null   object
 16  track_id          4073 non-null   object
 17  Unnamed: 17       1 non-null      object
dtypes: bool(1), float64(9), int64(4), object(4)
memory usage: 545.1+ KB
```

Une autre façon de déterminer le nombre de valeurs manquantes est d'effectuer la somme des valeurs nulles pour chaque colonne du DataFrame.

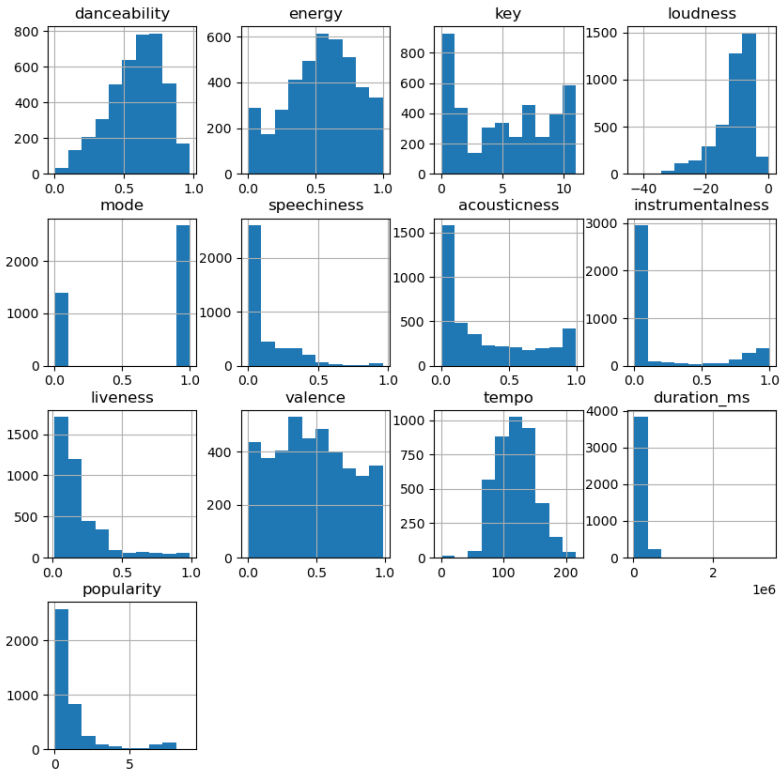
La méthode 'describe()'de pandas quant à elle fournit les statistiques de chaque attribut numérique d'un DataFrame. Cela inclut la moyenne, l'écart type, les valeurs minimales et maximales, et les quartiles (25%, 50%, 75%) :

	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	
count	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4073.000000	4
mean	0.584909	0.545931	5.171372	-10.877235	0.658237	0.142364	0.327200	0.199610	0.198481	0.468564	119.483633	2
std	0.195275	0.253181	3.571346	6.082619	0.474359	0.169173	0.334721	0.348140	0.177769	0.273403	30.626790	1
min	0.000000	0.000020	0.000000	-43.046000	0.000000	0.000000	0.000000	0.000000	0.016500	0.000000	0.000000	1
25%	0.460000	0.372000	2.000000	-13.181000	0.000000	0.038600	0.030200	0.000000	0.096300	0.249000	96.512000	1
50%	0.608000	0.565000	5.000000	-9.338000	1.000000	0.060300	0.194000	0.000037	0.124000	0.460000	119.948000	1
75%	0.732000	0.736000	8.000000	-6.693000	1.000000	0.195000	0.589000	0.219000	0.241000	0.684000	139.993000	2
max	0.973000	1.000000	11.000000	0.416000	1.000000	0.962000	0.996000	1.000000	0.990000	0.987000	215.983000	3

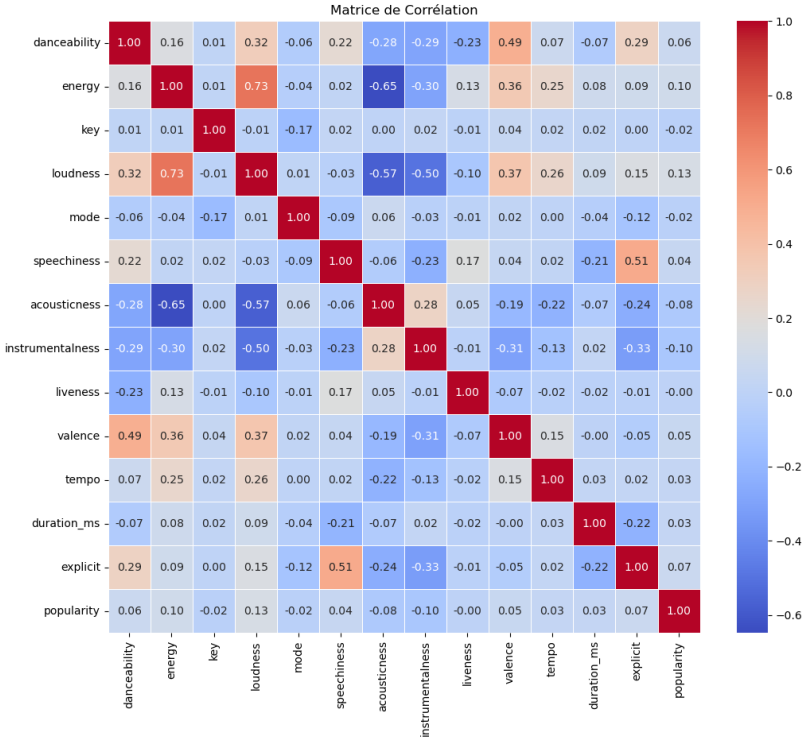
Afin de visualiser les résultats précédents et la distribution de chaque colonne, on peut faire appel aux histogrammes :

On remarque que :

Les valeurs des données sont distribuées plutôt asymétriquement.
Par exemple, pour les attributs ‘danceability’, ‘energy’, ‘loudness’, ‘valence’ et ‘tempo’ sont asymétrique à droite : la majorité des valeurs sont >0,5.
Ce qui est le contraire pour les autres attributs, qui ont une distribution essentiellement focalisées sur une ou deux valeurs.
En étudiant l’échelle des données de chaque attribut, on peut conclure qu’une standardisation des valeurs est nécessaires : prenons par exemple l’attribut ‘duration_ms’ et ‘energy’ : le premier est compris entre 17 951 et 3 408 890, le second entre 0 et 1.La standardisation permet de ramener ces valeurs à une échelle commune.



Matrice de corrélation :

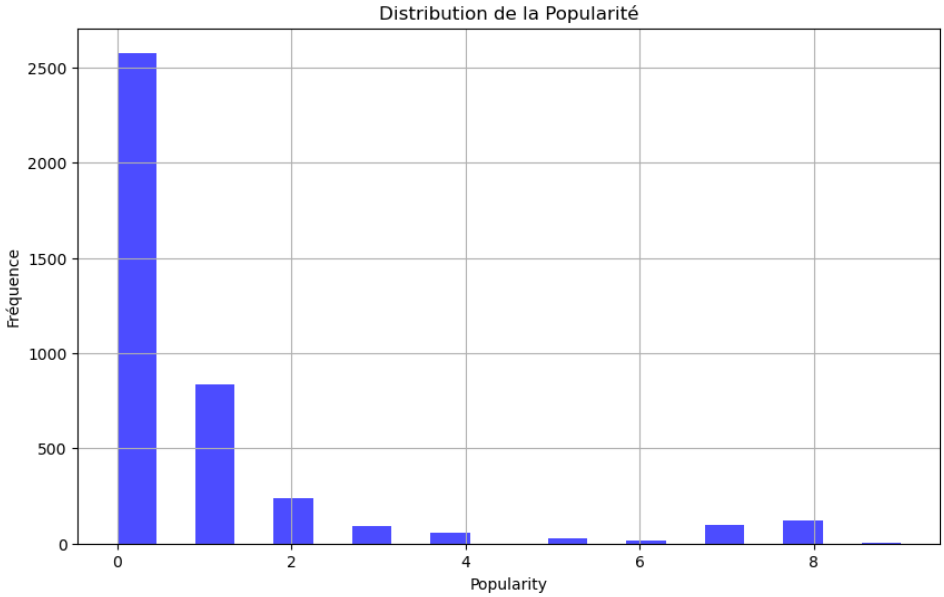


A partir de la matrice de corrélation obtenue, on peut voir que les attributs ‘key’ et ‘popularity’ sont faiblement corrélés avec tous les autres attributs, cela signifie qu'il y a une faible corrélation linéaire entre ces attributs et les autres. Cela pourrait indiquer que 'key' et la popularité ne sont pas fortement influencées par des relations linéaires avec d'autres attributs tels que la danseabilité, l'énergie, le tempo, etc.

Cependant, on peut remarquer que certains attributs sont fortement corrélés entre eux, tels que les attributs ‘energy’ et ‘loudness’, ou encore ‘energy’ et ‘acousticness’.

Partie 2 : Préparations des données et classification.

Le problème peut être caractérisé comme un problème de classification supervisée. La variable cible à prédire est "popularity", une valeur discrète représentant le niveau de popularité d'une musique sur une échelle de 0 à 11. Il s'agit d'une tâche de classification multiclasse, où chaque classe correspond à une plage de valeurs de popularité. L'objectif est de développer un modèle capable de prédire la popularité d'une chanson en fonction de certaines caractéristiques musicales. On peut observer la distribution de chaque classes dans la figure suivante :




```
Classe 0 : 2576 occurrences.
Classe 1 : 836 occurrences.
Classe 2 : 240 occurrences.
Classe 3 : 93 occurrences.
Classe 4 : 56 occurrences.
Classe 5 : 30 occurrences.
Classe 6 : 16 occurrences.
Classe 7 : 99 occurrences.
Classe 8 : 123 occurrences.
Classe 9 : 4 occurrences.
```

- ✓ La colonnes jugées ‘à supprimer’ sont : ‘track_name’, ‘track_artist’, ‘track_id’, ‘Unnamed: 17’, ‘key’, ‘liveness’, ‘mode’, ‘explicit’, ‘acousticness’ et ‘loudness’ pour les raisons suivantes :
  - L’attribut ‘Unnamed : 17’ ne contient que des valeurs nulles (appart une), il ne sera d’aucune utilité pour une tache de classification.
  - Les attributs ‘track_id’, ‘track_name’ et ‘track_artist’ sont des attributs catégoriques qui n’auront pas un grand impact pout déterminer le degré de popularité d’une chanson.
  - Les attributs ‘key’, ‘liveness’ et ‘mode’ dont ***très faiblement*** corrélées avec l’attribut cible ‘popularity’, ce qui signifie qu’ils ont peu d’influence sur la popularité, et ainsi sur la classification.
  - Les attributs ‘acousticness’ et ‘loudness’ sont ***fortement*** corrélés avec ‘energy’, ce qui signifie que qu’on pourrait calculer l’un avec l’autre.

Classification après standardisation :

- ✓ Les performances d'un classifieur de type arbre de décision avec une profondeur de 5 :
  - Taux d'erreur : 0.3685208604326251 scores : 0.6314791395673749 Temps : 0.2533700466156006 s.

Ce qui est moyennement bien pour un classifieur.

- ✓ Les performances d'un classifieur de type k plus proches voisins avec k=log(n) :
  - Taux d'erreur : 0.3972467119525943 scores : 0.6027532880474057 Temps : 0.2520167827606201 s.
- ➔ La distance utilisée est la distance par défaut : euclidienne.

Les résultats sont plus ou moins similaires, l'arbre de décision reste néanmoins plus performant.
Il est également important de noter que le temps d'apprentissage est assez similaire entre les deux modèles, ce qui suggère que, du point de vue du temps de calcul, ils sont tous deux assez efficaces.
L'exploration de méthodes d'apprentissage profond pourraient être envisagée pour pousser davantage les performances.

Colonnes discriminantes :

Lorsque l'arbre de décision est analysé en utilisant les valeurs d'importance des caractéristiques ('feature_importances_'), on constate que les six colonnes les plus discriminantes sont 'energy', 'valence', 'duration_ms', 'speechiness', 'danceability' et 'instrumentalness'. Cela montre que ces attributs jouent un rôle crucial dans la prise de décision du modèle et sont les plus influents pour prédire la popularité.

Pour confirmer cette observation, une expérimentation avec le classifieur des k-plus proches voisins (KPP) a été réalisée en variant la valeur de k. Les résultats montrent que les colonnes absentes n'ont pas d'influence sur les performances du modèle, et même dans certains cas, une légère amélioration a été notée. Le meilleur score obtenu était de 62.48% pour k=20.

Cela renforce l'idée que la sélection appropriée des caractéristiques est importante pour la construction d'un modèle efficace. En éliminant les colonnes moins discriminantes, on peut simplifier le modèle tout en maintenant ou améliorant les performances, ce qui peut être bénéfique en termes de complexité et de mémoire.

- ➔ On peut transformer ce problème de classification en un problème de régression en considérant l'attribut 'popularity' comme étant continu et non catégorique. Il suffira d'utiliser un modèle de régression tels que la régression linéaire ou des arbres de régression.

Utilisation des ces sources pour ce TP :

- https://mljar.com/blog/visualize-decision-tree/
- https://medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://chartio.com/learn/charts/histogram-complete-guide/