Instructions

Rendre un rapport PDF de 4 pages max + codes sous fic(s).py, pour répondre aux sections 1 et 2. Si vous ré-utilisez du code ou du texte, cela doit être clairement lisible dans votre rendu : il faut citer leur source et leur auteur, avec si possible un lien internet s'il existe, ou des références permettant de le(s) retrouver.

Si vous partagez votre code/rapport publiquement, ou de façon privée, signalez le nous (en première page de votre rapport, en indiquant les canaux de diffusion utilisés). Cela peut vous éviter un 0/20 si nous retrouvons – même partiellement – vos sources dans des rendus du MINF1.

1 Classification non-supervisée (commencé au TDM précédent)

Ici, on cherchera à identifier des regroupements de chansons aux caractéristiques similaires, et à étudier ces regroupements au regard des caractères de non-popularité (colonne `popularity`), et au regard des octaves/harmoniques des chansons (colonne `key`).

1. Produire un *dataframe* issu de l'initial, qui enlève les colonnes `popularity` et `key` (tout en conservant ces informations que nous utiliserons plus tard comme supervision), et sans toutes les colonnes enlevées dans la section précédente. Quelles sont les colonnes descriptives restantes ?
2. Nous voulons retrouver les classes de non-popularité à partir des autres colonnes. Combien y en a-t-il ? Quel est le nombre de clusters que nous voulons définir ?
3. Lancer l'apprentissage d'un clustering sur toutes les données, avec autant de clusters que nécessaires pour résoudre ce problème (justifier ce nombre). Quels sont les mesures pour évaluer la qualité du clustering obtenu (1. comme si nous avions accès à l'impopularité de chaque chanson, et 2. comme si nous n'y avions pas accès)

2 Mesures de qualité d'un regroupement

Dans cette partie, l'objectif est d'apprendre un clustering des chansons, en essayant de retrouver les clusters de popularité, et de mesurer sa qualité à l'aide de plusieurs moyens.

2.1 Qualité d'un clustering k-moyennes

Répéter dix fois les étapes suivantes (en garantissant des splits différents), et indiquer la moyenne des scores obtenus.

1. Découper le jeu de donnéees en 70% jeu d'apprentissage et 30% jeu de test ;
2. Calculer un regroupement en k clusters, k étant le nombre déterminé dans la section précédente
3. Déterminer, pour chaque élément du jeu test, le cluster auquel il appartient.
4. Calculer, sur le jeu d'apprentissage : (a) le score de Davies Bouldin, (b) la silhouette moyenne, et (c) le rand index entre les vrais clusters des exemples – popyularité – et ceux que le clustering leur attribue.
5. Calculer, sur le jeu de test : (a) le score de Davies Bouldin, (b) la silhouette moyenne, et (c) le rand index entre les vrais clusters des exemples et ceux que le clustering leur attribue.

Commenter quant à la qualité du clustering moyen obtenu.

2.2 Qualité d'un clustering mélange de gaussiennes

Il existe des dizaines d'algorithmes de clustering, et pour avoir une idée de l'impact de chacun sur des jeux de données artificielles, voir `https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py`.
Dans cette section, nous appliquerons l'algorithme des mélanges de gaussiennes, qui cherche à maximiser la vraisemblance d'appartenance des exemples aux clusters. On utilisera cet algorithme avec k composants (k distributions).
Procéder aux mêmes expériences que dans le cas de k-means.

2.3 Comparaison des deux méthodes

1. Apprendre un regroupement k-means, et un regroupement *mélange de gaussiennes*, avec toutes les données. Quel est le rand index entre les deux clustering obtenus ? Interpréter et commenter.
2. Quelle est la meilleure méthode de clustering parmi les deux testées, dans le cas du jeu de données spotify ?
3. Concernant le regroupement obtenu par cette méthode, dans combien de clusters les exemples de popularité 3 se retrouvent-ils ? Idem pour autres indices de popularité ?
4. Plus généralement, quel est le rand index de ce regroupement par rapport à la vérité terrain (véritable popularité de chaque exemple) ? Que pouvez-vous conclure sur le lien entre popularité et les attributs décrivant les chansons ?
5. Dans le jeu de données initiales, indiquer les chansons les plus proches de chacun des centres des clusters.
6. Comment pourrions-nous procéder pour savoir quel(s) attribut(s) sont prépondérants dans les clusters ?

3 Annexe : dataset description

See the source `https://www.kaggle.com/datasets/estienneggx/spotify-unpopular-songs/data`