Résumé des méthodes de clustering (K-means, CAH, DBSCAN) :

Table des matières

1. Introduction

Le datascientist, dans son travail doit réaliser plusieurs types d'étude sur les jeux de données qui l'intéresse. Parmi elles, les clusterings permettent, de manière non-supervisé, de mettre en lumière les potentielles corrélations entre les différentes variables du jeu de donnée.

Nous verrons donc tout d'abord la réalisation de l'import des données puis la manière dont un data analyste réalise des clusterings et en tire des conclusions.

Dans le cadre de ce TP, les packages « Pandas », « sklearn », « Numpy », « Seaborn », « Scipy » et « Matplotlib » ont été utilisés. Les jeux de donnée Test_Cluster_Random.txt, Test_Cluster_Melanges.txt, Test_Cluster_Distincts.txt, Test_Cluster_Atypiques.txt, représentant des données Mélangé, distinctes, aléatoire et atypiques, composera la base de notre étude.

2. Import des données

Afin d'apporter les données à étudier dans python, on utilise le package Pandas qui simplifie la création et la manipulation de dataframe.

Suite à l'import des packages,

```
# import libraries
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import DBSCAN
```

Il suffit d'utiliser read_csv, une fonction de pandas, pour lire un csv et le récupérer sous forme de dataframe. Exemple :

```
dist = pd.read_csv("Data\Test_Clusters_Distincts.txt", sep=" ", header=None)
atyp = pd.read_csv("Data\Test_Clusters_Atypiques.txt", sep=" ", header=None)
mel = pd.read_csv("Data\Test_Clusters_Melanges.txt", sep=" ", header=None)
rand = pd.read_csv("Data\Test_Clusters_Random.txt", sep=" ", header=None)
dist.head()
```

	0	1	2	3	4	5
0	0.112213	0.614523	-0.120554	-0.076287	0.844718	0.376434
1	-0.404110	-0.042121	0.574591	-0.204818	0.360084	1.398744
2	-0.018190	0.263355	0.429380	0.755978	0.301715	0.449145
3	1.693847	1.382552	0.834288	1.245551	1.641393	1.114700
4	0.721389	-0.407652	0.573885	0.201700	0.742344	2.211372

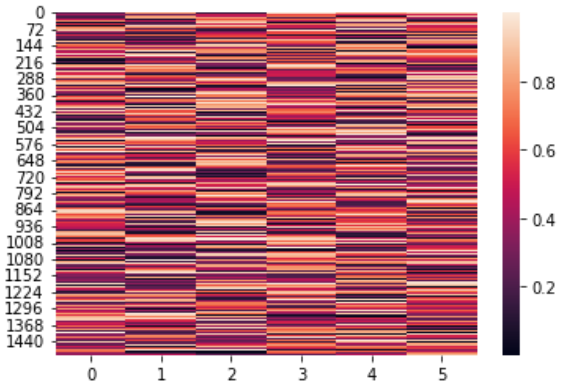
3. Étude de cas : Utilisation des méthodes de clusterings

Afin de déterminer visuellement la pertinence d'une méthode de clustering il est possible de réaliser plusieurs graphiques.

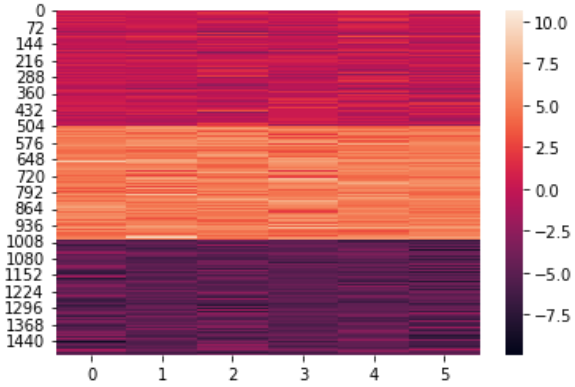
3.1. Graphiques

En utilisant la fonction « Heatmap » proposé par le package seaborn on peut rapidement voir que le jeu donnée Test_Clusters_Random présente moins de pertinence pour la méthode des clusterings que Test_Clusters_Distincts :

```
sns.heatmap(rand)
```




```
sns.heatmap(dist)
```



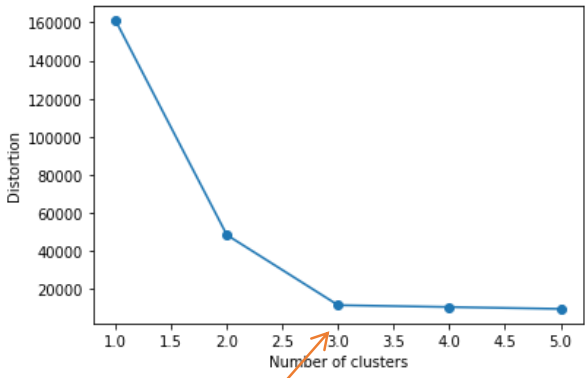
3.2. K-means

L'utilisation de la méthode des k-means est rendu possible grâce au package sklearn. On réalise un préalable un elbow plot dans le but de définir le nombre de cluster optimisé :

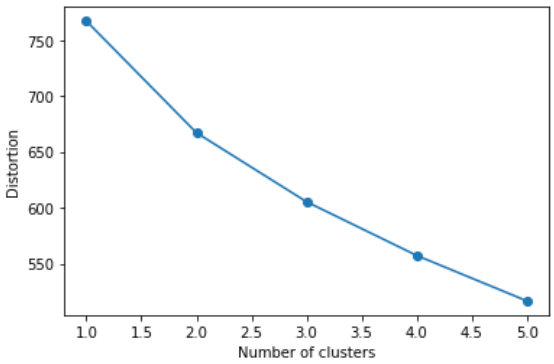
Exemple sur les dataframes Random et Distincts :

```
km = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=300, random_state=0) # kmeans vient du package sklearn
```

Graphique elbow sur Distincts



Graphique elbow sur Random



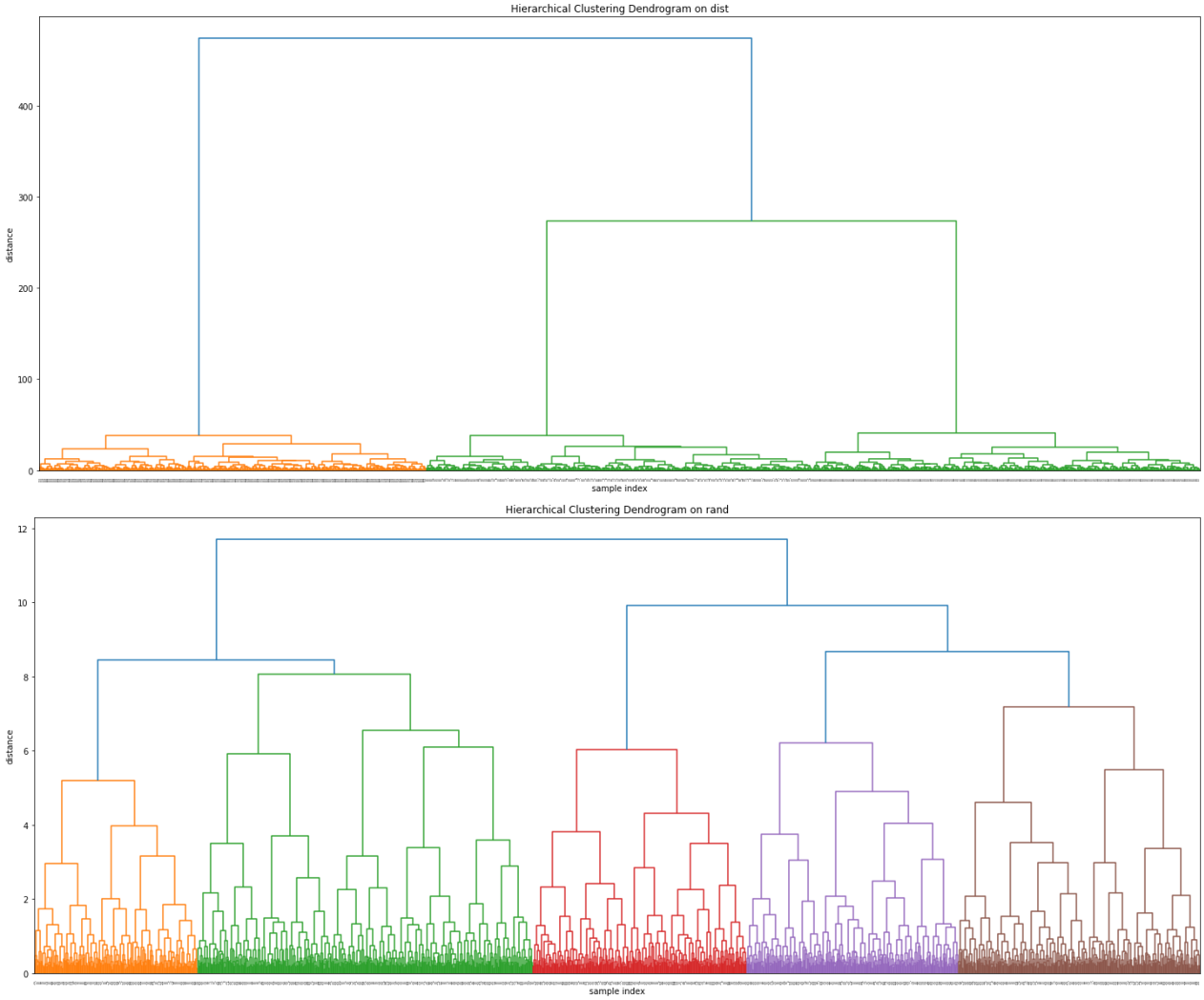
Indication du nombre de cluster pertinent

3.3. Réalisation des graphiques des différents clusterings

Plusieurs graphiques sont réalisables dans l'études des variables par clusterings. Nous allons voir ici les résultats et leurs interprétations sur les jeux de données distinctes et aléatoires pour les dendrogrammes.

3.3.1. Dendrogramme

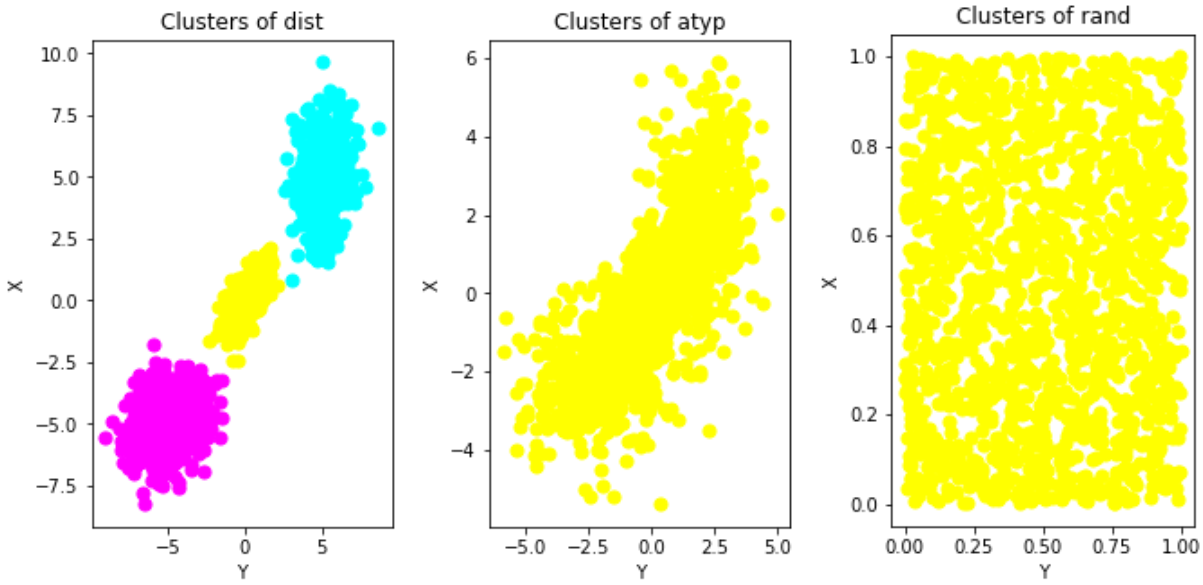
La réalisation d'un dendrogramme permet de visualiser la réduction de la variance du modèle pour chaque cluster ajouté.



On remarque ici que sur le premier dendrogramme (variables distincts), la variance varie fortement à chaque nouveau cluster jusqu'au 3ieme. Sur le second à l'inverse, la variance diminue plus lentement mais de manière continue.

3.3.2. DBSCAN

La méthode du DBSCAN permet de réaliser des clusterings plus précis mais surtout plus tolérant face aux structures convexes et linéaires. Cependant elle utilise une méthode de « frontière ».



On remarque que cette méthode perd en efficacité lorsque les données sont très proches. Ici, dans les cas différents des données distinctes, un seul cluster est toujours réalisé.