Casamian Marius /(cm205773)

Introduction à l'Intelligence Artificielle

Licence 2 — Portail Sciences et Technologies

Travaux dirigés No 6 : Clustering

Objectif du TP :

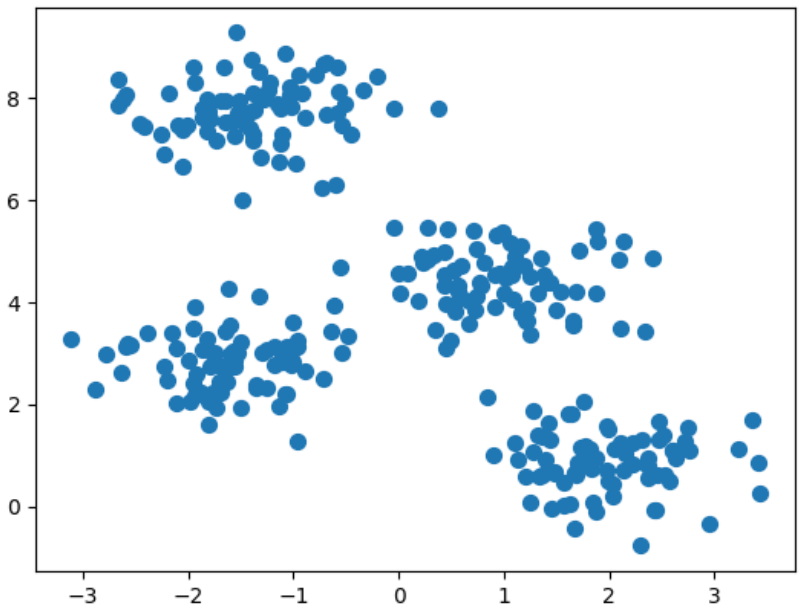
- Algorithme de clustering K-Means, Scatter(), PCA, Pandas avec Python
- Jeu de données de vins pour classification

Réalisation du TP :

Partie I/

(Code Python disponible dans le dossier zippé)

La fonction Partie1() comporte la première partie du TP, ce code génère des données, les affiche, applique l'algorithme K-means pour trouver les clusters, affiche les clusters attribués par K-means, les centres des clusters, et enfin compare les classes réelles avec celles prédites par K-means.

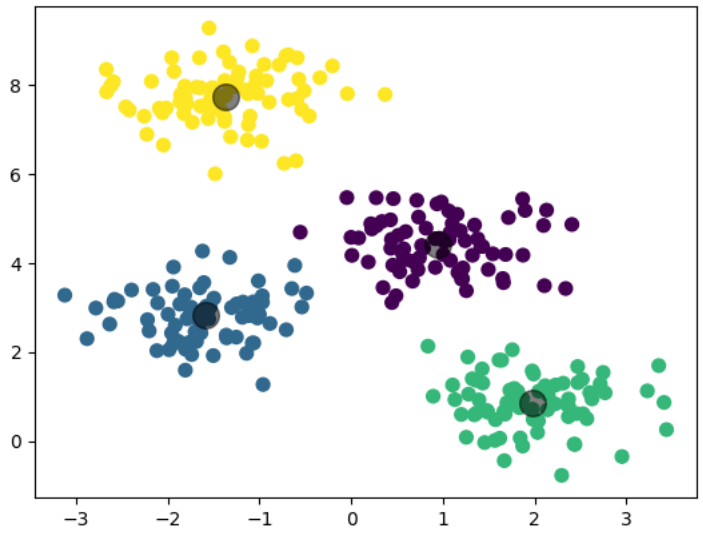


Titre : Coordonnées des points du jeu de données.

Comment l'interpréter ?

Il s'agit de données générées aléatoirement par make_blobs() que l'on visualise, on remarque pour le moment 4 «types » de groupes.

L'algorithme K-Means est ensuite utilisé pour effectuer un clustering du jeu de données « X » en 4 clusters et visualise son résultat à travers un graphique ci-dessous.



Titre : Segmentation en 4 clusters avec K-Means

Comment l'interpréter ?

Les points du jeu de données sont colorés en fonctions des clusters prédits par l'algorithme K-Means. Chaque groupe de couleurs représente un cluster.
Les centres des clusters sont affichés en noir et correspondent aux moyennes des points dans chaque cluster prédit.

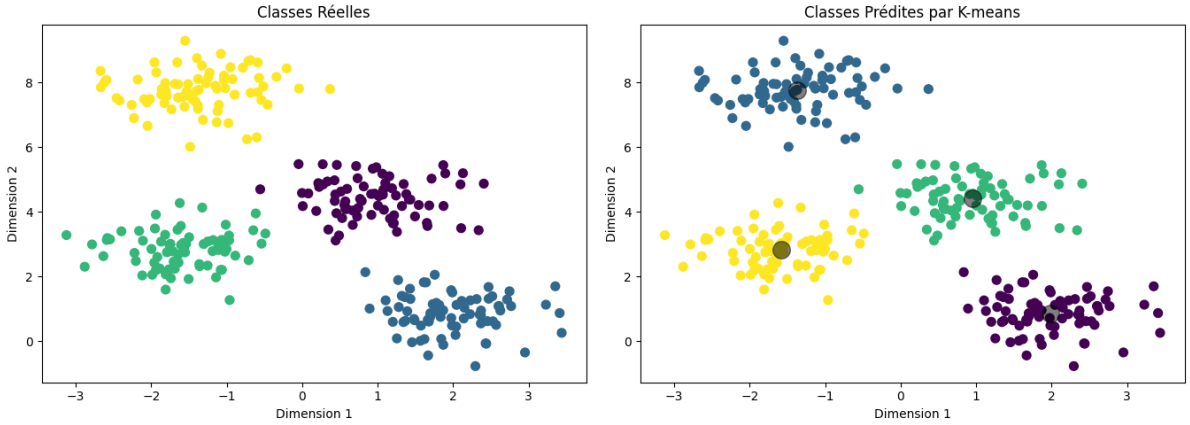
Si les clusters sont bien définis, les points du même cluster devraient être proches les uns des autres, et les centres des clusters devraient se situer au milieu des groupes de points. On remarque bien que c'est le cas ici.

Ensuite on vérifie les contenues des variables X et y : (Ex1/2.1/3)

```
X classes:  [1 3 0 3 1 1 2 0 3 3 2 3 0 3 1 0 0 1 2 2 1 1 0 2 2 0 1 0 2 0 3 3 0 3 3 3 3
 3 2 1 0 2 0 0 2 2 3 2 3 1 2 1 3 1 1 2 3 2 3 1 3 0 3 2 2 2 3 1 3 2 0 2 3 2
 2 3 2 0 1 3 1 0 1 1 3 0 1 0 3 3 0 1 3 2 2 0 1 1 0 2 3 1 3 1 0 1 1 0 3 0 2
 2 1 3 1 0 3 1 1 0 2 1 2 1 1 1 1 2 1 2 3 2 2 1 3 2 2 3 0 3 3 2 0 2 0 2 3 0
 3 3 3 0 3 0 1 2 3 2 1 0 3 0 0 1 0 2 2 0 1 0 0 3 1 0 2 3 1 1 0 2 1 0 2 2 0
 0 0 0 1 3 0 2 0 0 2 2 2 0 2 3 0 2 1 2 0 3 2 3 0 3 0 2 0 0 3 2 2 1 1 0 3 1
 1 2 1 2 0 3 3 0 0 3 0 1 2 0 1 2 3 2 1 0 1 3 3 3 3 2 2 3 0 2 1 0 2 2 2 1 1
 3 0 0 2 1 3 2 0 3 0 1 1 2 2 0 1 1 1 0 3 3 1 1 0 1 1 1 3 2 3 0 1 1 3 3 3 1
 1 0 3 2]
y_kmeans classes:  [1 2 3 2 1 1 0 3 2 2 0 2 3 2 1 3 3 1 0 0 1 1 3 0 0 3 1 3 0 3 2 2 3 2 2 2 2
 2 0 1 3 0 3 3 0 0 2 0 2 1 0 1 2 1 1 0 2 0 2 1 2 3 2 0 0 0 2 1 2 0 3 0 2 0
 0 2 0 3 1 2 1 3 1 1 2 3 1 3 2 2 3 1 2 0 0 3 1 1 3 0 2 1 2 1 3 1 1 3 2 3 0
 0 1 2 1 3 2 1 1 3 0 1 0 1 1 1 1 0 1 0 2 0 0 1 2 0 0 2 3 2 2 0 3 0 3 0 2 3
 2 2 2 3 2 3 1 0 2 0 1 3 2 3 3 1 3 0 0 3 1 3 3 2 1 3 0 2 1 1 3 0 1 3 0 0 3
 3 3 3 1 2 3 0 3 3 0 0 0 3 0 2 3 0 1 0 3 2 0 2 3 2 3 0 3 3 2 0 0 1 1 3 2 1
 1 0 1 0 3 2 2 3 3 2 3 1 0 3 1 0 2 0 1 3 1 2 2 2 2 0 0 2 3 0 1 3 0 0 0 1 1
 2 3 3 0 1 2 0 3 2 3 1 1 0 0 3 1 1 1 3 2 2 1 1 3 1 1 1 2 0 2 3 1 1 2 2 2 1
 1 3 2 0]
```

Pour mieux comprendre ce résultat rien de mieux qu'un graphique !

(Optionnel)On rajoute un code supplémentaire pour mieux visualiser les deux sous-graphiques.



Titre : Répartition des classes réelles et prédites par K-Means

On remarque que les clusters sont positionnés aux mêmes endroits dans le graphique des classes réelles, mais les couleurs sont différentes dans le graphique des classes prédites. Il s'agit en fait d'une permutation des étiquettes de classes par K-Means. L'algorithme a tout de même réussi à regrouper les points spatialement mais les couleurs et numéros ne peuvent pas correspondre fidèlement à ceux des classes réelles. K-Means reste un succès pour cette structure car il identifie bien les groupements, et des similitudes dans les données sont visibles (spatiale par exemple).

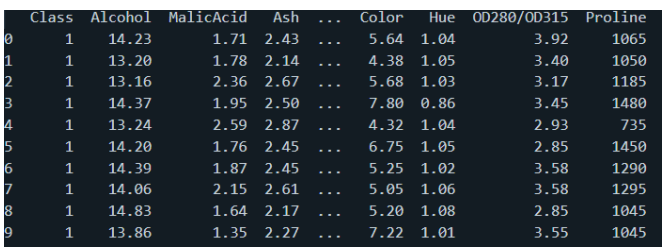
Partie II/

I/

<u>(Code Python disponible dans le dossier zippé)</u>

Dans cette partie :

Analyse du jeu de données :

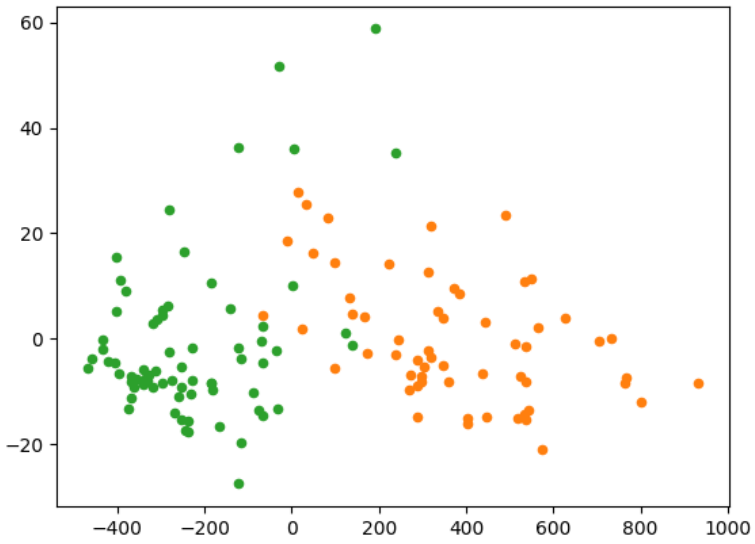
	Affiche les 10 premiers éléments du jeu de données.

	Class	Alcohol	MalicAcid	Ash	...	Color	Hue	OD280/OD315	Proline
0	1	14.23	1.71	2.43	...	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	...	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	...	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	...	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	...	4.32	1.04	2.93	735
5	1	14.20	1.76	2.45	...	6.75	1.05	2.85	1450
6	1	14.39	1.87	2.45	...	5.25	1.02	3.58	1290
7	1	14.06	2.15	2.61	...	5.05	1.06	3.58	1295
8	1	14.83	1.64	2.17	...	5.20	1.08	2.85	1045
9	1	13.86	1.35	2.27	...	7.22	1.01	3.55	1045

[10 rows x 14 columns] Class Alcohol MalicAcid Ash ... Color Hue OD280/OD315 Proline 168 3 13.58 2.58 2.69 ... 8.660000 0.74 1.80 750 169 3 13.40 4.60 2.86 ... 8.500000 0.67 1.92 630 170 3 12.20 3.03 2.32 ... 5.500000 0.66 1.83 510 171 3 12.77 2.39 2.28 ... 9.899999 0.57 1.63 470 172 3 14.16 2.51 2.48 ... 9.700000 0.62 1.71 660 173 3 13.71 5.65 2.45 ... 7.700000 0.64 1.74 740 174 3 13.40 3.91 2.48 ... 7.300000 0.70 1.56 750 175 3 13.27 4.28 2.26 ... 10.200000 0.59 1.56 835 176 3 13.17 2.59 2.37 ... 9.300000 0.60 1.62 840 177 3 14.13 4.10 2.74 ... 9.200000 0.61 1.60 560	Affiche les 10 derniers éléments du jeu de données.
<class 'pandas.core.frame.DataFrame'> RangeIndex: 178 entries, 0 to 177 Data columns (total 14 columns): # Column Non-Null Count Dtype 0 Class 178 non-null int64 1 Alcohol 178 non-null float64 2 MalicAcid 178 non-null float64 3 Ash 178 non-null float64 4 AlcalinityAsh 178 non-null float64 5 Magnesium 178 non-null int64 6 PhenolsTotal 178 non-null float64 7 Flavanoids 178 non-null float64 8 NonflavanoidPhenols 178 non-null float64 9 Proanthocyanins 178 non-null float64 10 Color 178 non-null float64 11 Hue 178 non-null float64 12 OD280/OD315 178 non-null float64 13 Proline 178 non-null int64 dtypes: float64(11), int64(3) memory usage: 19.6 KB	Affiche toutes les informations du jeu de données.
Class 0 Alcohol 0 MalicAcid 0 Ash 0 AlcalinityAsh 0 Magnesium 0 PhenolsTotal 0 Flavanoids 0 NonflavanoidPhenols 0 Proanthocyanins 0 Color 0 Hue 0 OD280/OD315 0 Proline 0	Affiche les valeurs nulles du jeu de données.
dtype: int64	Les types des variables dans le jeu de données
[1 2 3]	Les labels du DataFrame

II/

Réduction de la dimension des données avec PCA



Titre : Graphique de Clustering des Données de Vin avec K-means

Chaque point dans le graphique représente une observation dans les données de vin.

Les points sont colorés en fonction des clusters prédits par K-means (3 clusters).

La PCA est utilisée pour réduire les dimensions des données à deux composantes principales afin de visualiser les clusters dans un espace bidimensionnel.

Le graphique montre la répartition des données en fonction des clusters prédits.

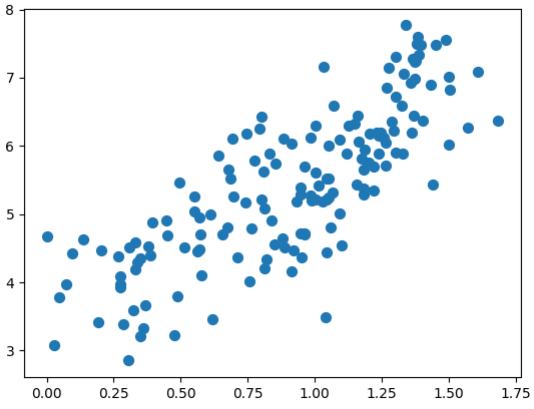
Interprétation Potentielle :

L'algorithme K-means a réussi à identifier des structures de cluster significatives car les points de données sont regroupés distinctement.

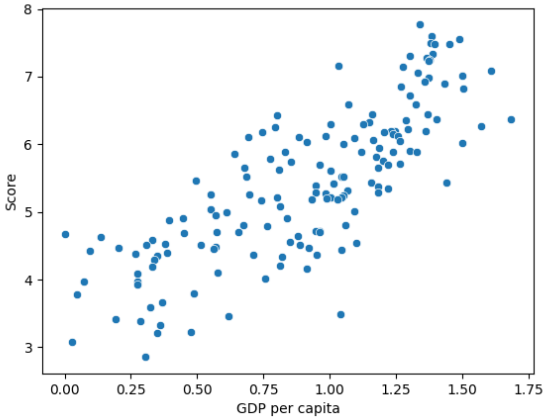
(3.2/27) labels_unique = [0,1,2]

III/ Scatter()

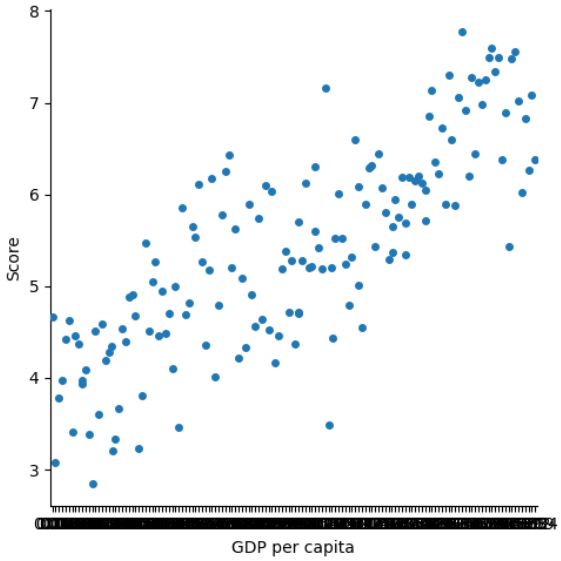
Voici les trois graphiques en nuages de points



Titre : Richesse pays / impression d'être heureux



Titre : Graphique améliorer



Titre : Visualiser par catégorie

Conclusion du TP :

Nous avons vu l'algorithme K-Means, appris à visualiser des données sur des graphiques, utiliser des librairies de données (Pandas, matplotlib…), utilisé la réduction PCA et répondu aux questions du TP.