Produisez une étude de marché avec Python

Antoine Jeambourquin
Projet 9

Contexte

L'objectif principal est d'identifier les pays potentiels pour nos exportations de poulets. Pour réaliser cette étude on va sélectionner des données en provenance de la FAO.

Après un traitement, nettoyage et une préparation de nôtre jeu de données. Je vais utiliser plusieurs technique de clustering afin de trouver un groupe de pays pertinent pour y cibler les exportations de poulet.

Après avoir réduit les dimensions de notre jeu de données grâce à une PCA (Principal Component Analysis) je vais appliquer l'algorithme de clustering Kmeans et faire une CAH (Classification Ascendante Hiérarchique).

Sélection des données pertinentes pour l'étude

- Un fichier avec des données de production, exportations, importations, disponibilité alimentaire, et disponibilité en protéines par pays en 2017.
- Un fichier avec la population par pays en 2017.
- Un fichier avec le pourcentage de personnes sous alimenté par pays en 2017.
- Un fichier avec le PIB par habitant par pays en 2017.
- Un fichier avec les données de stabilité politique en 2017.

Notebook de préparation, nettoyage et analyse exploratoire

Préparation et nettoyage

Pour bien commencer l'étude de marché :

- ➔ Sélectionner le données qui sont pertinentes pour notre analyse et supprimer les information superflues.

- ➔ Créer un jeu de données "prêt à l'emploi" pour notre future analyse grâce à des jointures.

- ➔ Uniformiser nos unités de grandeur si possible.

- ➔ Pondéré nos données par la population afin de pouvoir comparer les données de pays avec des populations très différentes.Dans cette logique nous allons donc supprimer l'Inde et la Chine qui "écrasent" les autres pays par leur population importante.

Tri et jointures de nos données pour garder un jeu propre et cohérent

Nous allons pondérer chacune de nos colonnes par la population afin de rendre la comparaison entre pays plus pertinente.

Nous avons des données qui sont influencé fortement par la population de nos pays, la chine et l'inde peuvent donc poser problème car leur population écrase le reste des pays et rend l'analyse difficile.

```
# pondération par la population de nos colonnes qui ne le sont pas déjà
data['Disponibilité intérieure'] = data['Disponibilité intérieure'] / data['Population']
data['Importations - Quantité'] = data['Importations - Quantité'] / data['Population']
data['Exportations - Quantité'] = data['Exportations - Quantité'] / data['Population']
data['Production'] = data['Production'] / data['Population']
data['Disponibilité de protéines en quantité (g/personne/jour)'] = data['Disponibilité de protéines en quantité (g/personne/jour)'] * 365
```

Mise à niveau des données

Nous allons ensuite exprimé chacune de nos colonnes dans des ordres de grandeur équivalent si possible.

```
#transformation en Kg de toutes les données
data['Disponibilité intérieure'] = data['Disponibilité de protéines en quantité (g/personne/jour)']* 1000000*365
data['Importations - Quantité'] = data['Importations - Quantité'] * 1000000
data['Production'] = data['Production']  * 1000000
data['Exportations - Quantité'] = data['Exportations - Quantité']  * 1000000

# Multiplier par 1000 pour respecter l'unité
data['Population'] = data['Population']*1000

# Mettre à la bonne échelle poyr respecter l'unité de la donnée
df_sous_nutrition['Nombre de personnes sous-alimentées'] = df_sous_nutrition['Nombre de personnes sous-alimentées']*1000000
data.head()
```

Préparation et nettoyage

Dans un deuxième temps :

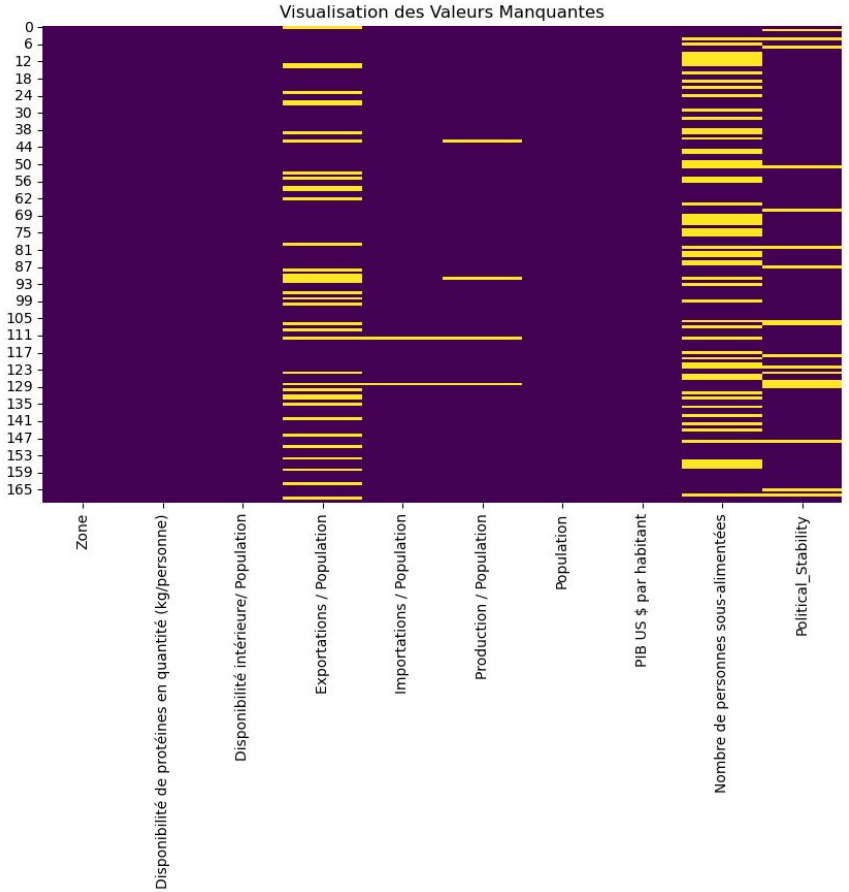
- ➔ Traiter les valeurs manquantes selon leurs volumes et en adaptant la méthode selon la métrique.

- ➔ Étudier nos outliers et appliquer une fonction logarithme pour limiter leur impact, (transformer des données asymétriques et stabiliser la variance)

- ➔ Normaliser le jeu de données (équilibre les échelles entre les variables)

Gestion des valeurs manquantes

- ➔ La colonne nombre de personnes sous alimenté contient trop de valeurs nuls et ne va pas importer assez d ‘information
- ➔ la colonne exportation va être rempli par des zéros en partant du principe que ce sont des pays non exportateur
- ➔ De même pour les importations
- ➔ le reste des valeurs manquantes vont être remplacé par des 0.



Limiter l'impact des outliers

Afin de réduire l'impact des outlier on va utiliser une fonction logarithmique sur chacune de nos colonnes.

- réduction de l'effet des valeurs aberrantes
- Normalisation de la distribution

Etude outliers

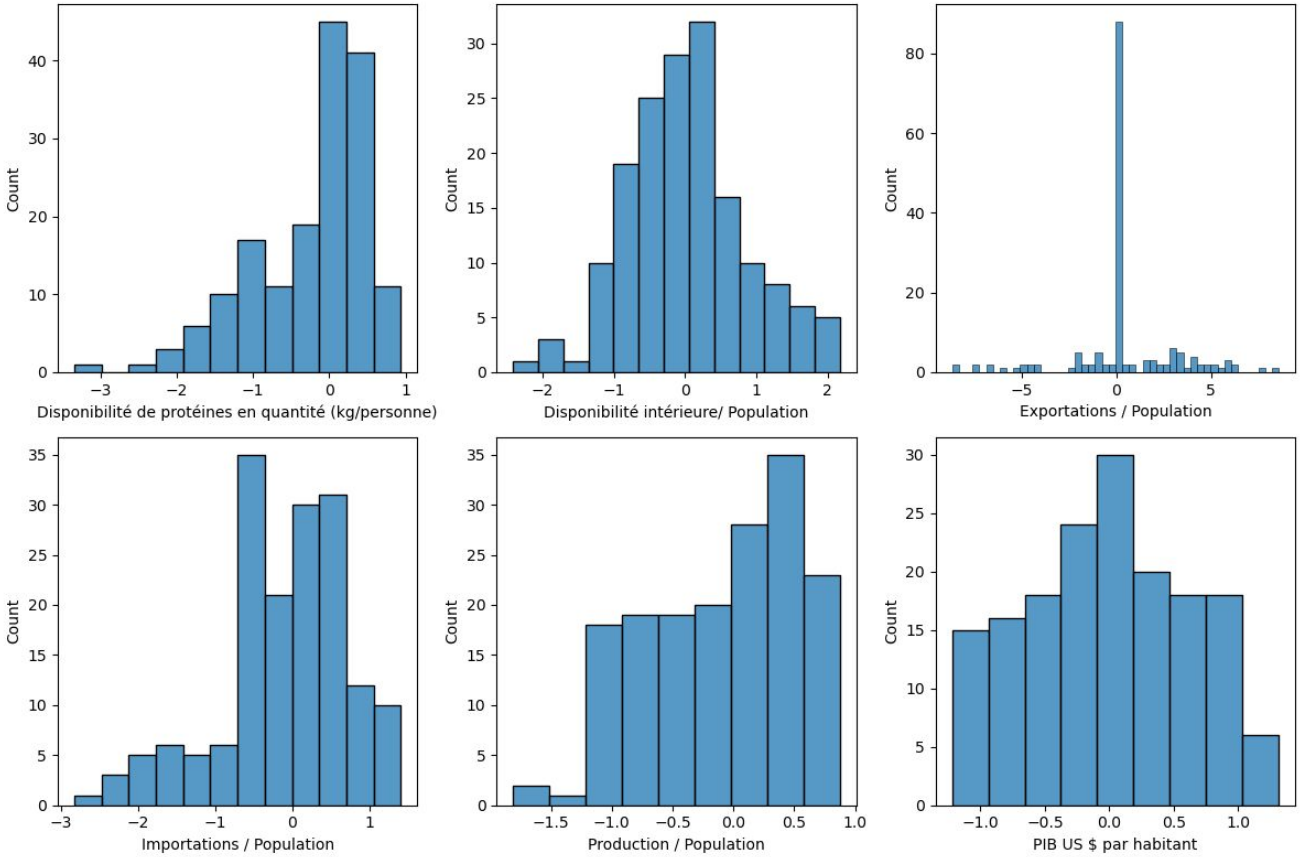
```
fc.etude_outliers(data,  seuil=2.0)
```

```
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Disponibilité
de protéines en quantité (kg/personne)' est 3.59%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Disponibilité
intérieure/ Population' est 4.19%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Exportations /
Population' est 4.79%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Importations /
Population' est 5.99%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Production / P
opulation' est 5.39%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'PIB US $ par h
abitant' est 5.99%
Le pourcentage de valeurs considérées comme des outliers en utilisant le Z-score au seuil 2.0 dans la colonne 'Political_Stab
ility' est 5.99%
```

Normalisation des données

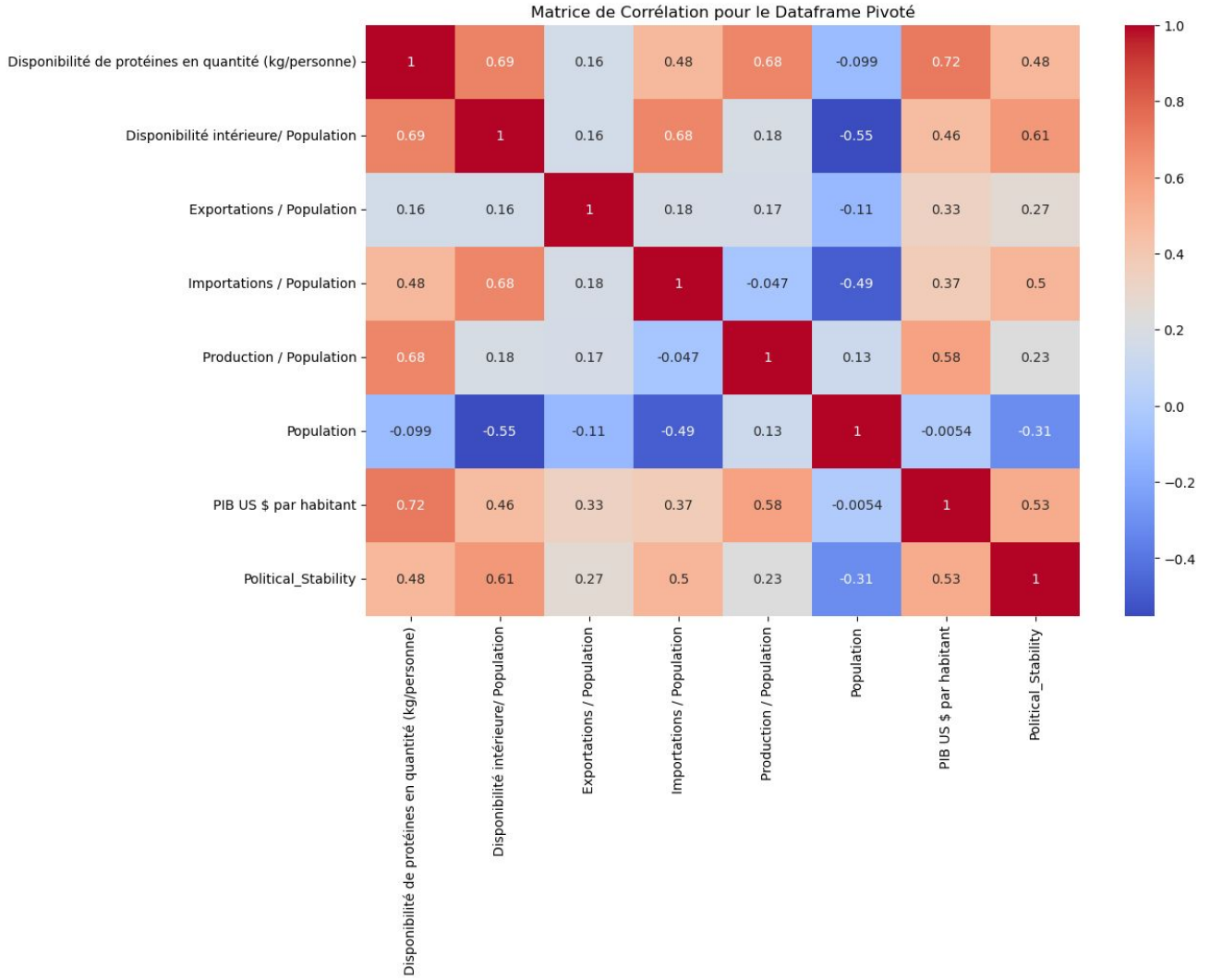
On va utiliser 3 méthodes de normalisation différentes afin de créer 3 dataframes à étudier par la suite dans notre feature engineering:

- Robust
- Standard
- MinMax



Matrice de Corrélation

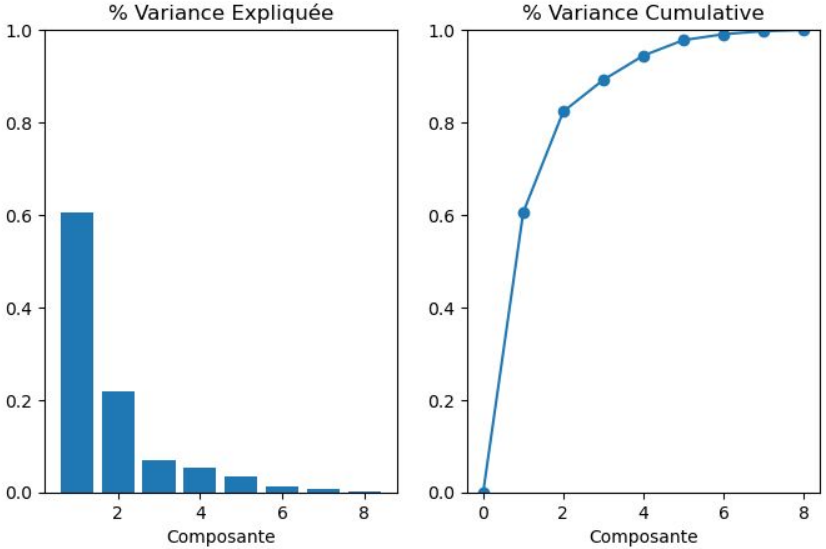
La matrice de corrélation permet d'analyser les relations entre nos différentes variables.

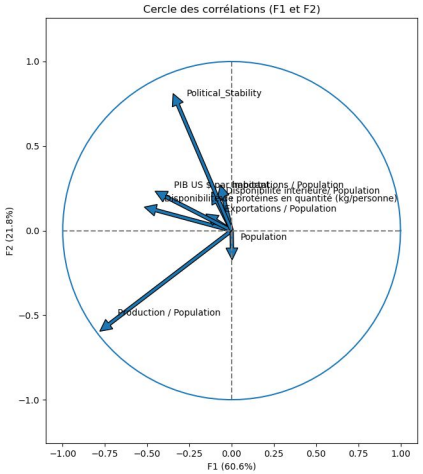


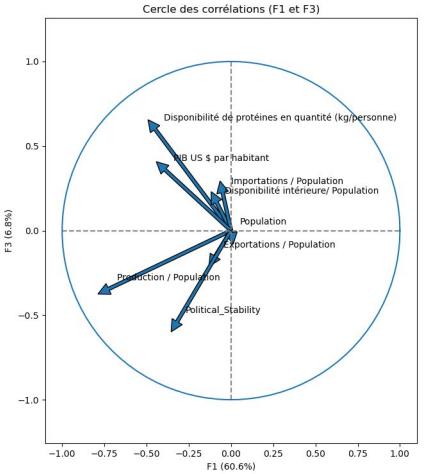
Notebook, clustering et visualisations

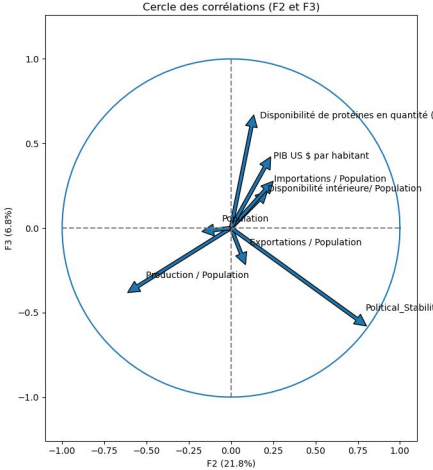
PCA

Pour réduire les dimensions du jeu de données nous allons appliquer une PCA

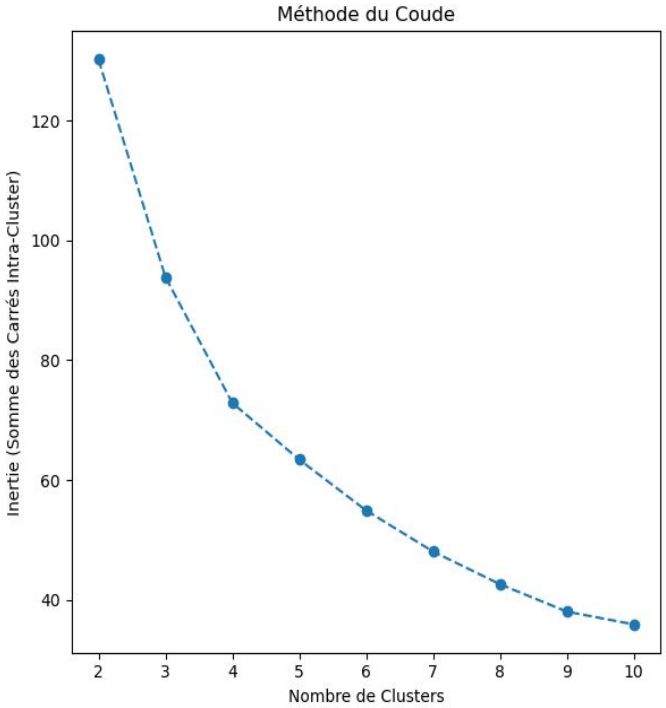


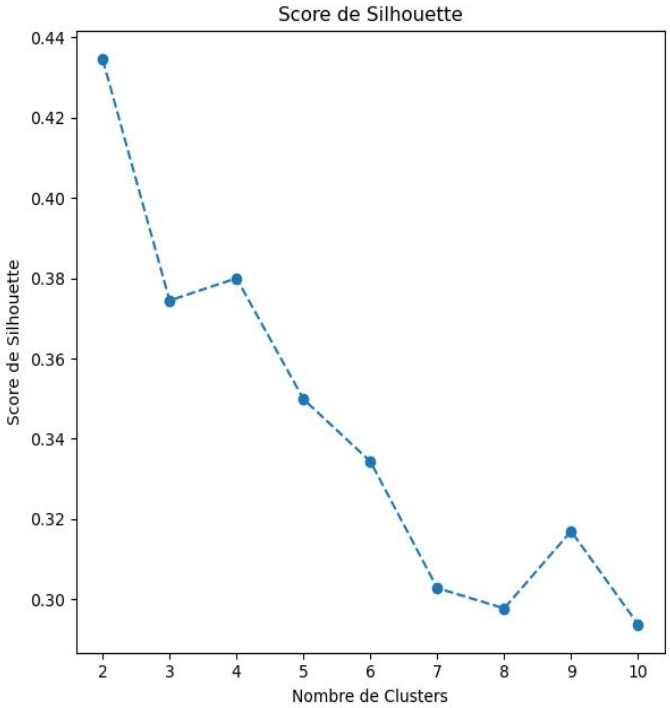






Nombre de clusters





Pour trouver le nombre de clusters optimal pour notre jeu de données, nous allons utiliser la méthode du coude et le score de silhouette.

Dans notre cas nous allons privilégier l'aspect de l'analyse métier.
Le nombre de clusters optimal au vu des résultats est 4.

Cependant notre but est de trouver un petit group ede pays susceptible d'être de bonnes cibles pour nos exportations.

Pour rendre la décision plus facile et l'analyse métier plus fine le clustering sera effectué avec 9 clusters.

Afin de trouver des clusters pertinent pour notre étude, le feature engineering va être un élément essentiel.

Feature enginering

```
nb_clusters = 9
```

[14] ✓ 0.0s Python

```
# selection du poids de la colonne
poids_prod = 2
poids_exp = 1
poids_imp = 1
poids_dispo_int = 1
poids_disp_prot = 3
poids_pib = 2
poids_politique = 3
```

[15] ✓ 0.0s Python

```
# choix du dataframe
# data = robust
# data = standard
data = minmax
```

[16] ✓ 0.0s Python

```
#Les colonnes à qui accorder le plus d'importance
data['Production / Population'] = np.exp(data['Production / Population']) * poids_prod
data['Exportations / Population'] = np.exp(data['Exportations / Population'])* poids_exp
data['Importations / Population'] = data['Importations / Population']* poids_imp
data['Disponibilité de protéines en quantité (kg/personne)'] = data['Disponibilité de protéines en quantité (kg/personne)'] * poids_disp_prot
data['PIB US $ par habitant'] = data['PIB US $ par habitant'] * poids_pib
data['Disponibilité intérieure/ Population'] = data['Disponibilité intérieure/ Population'] * poids_dispo_int
data['Political_Stability'] = data['Political_Stability'] * poids_politique
```

[19] ✓ 0.0s Python

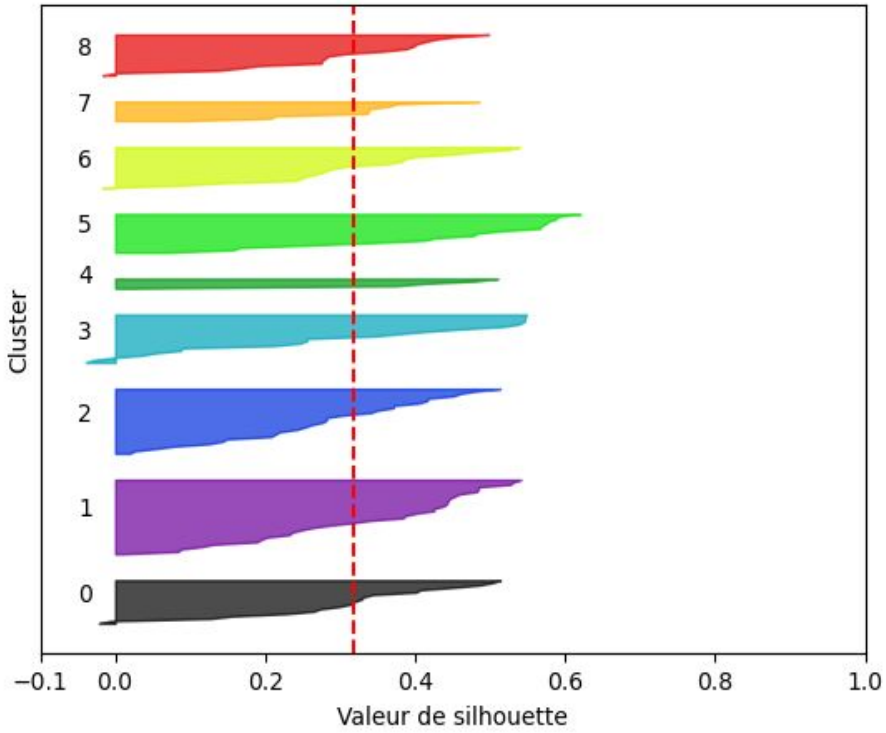
Le feature engineering va nous permettre de jouer avec tous les éléments de notre jeu de donnée afin de mettre en avant une distribution qui soit le plus étalé possible.

En attribuant un poids à une colonne on renforce ou non son importance dans le jeu de données et donc on influe sur la PCA et le clustering.

Un score de silhouette moyen faible

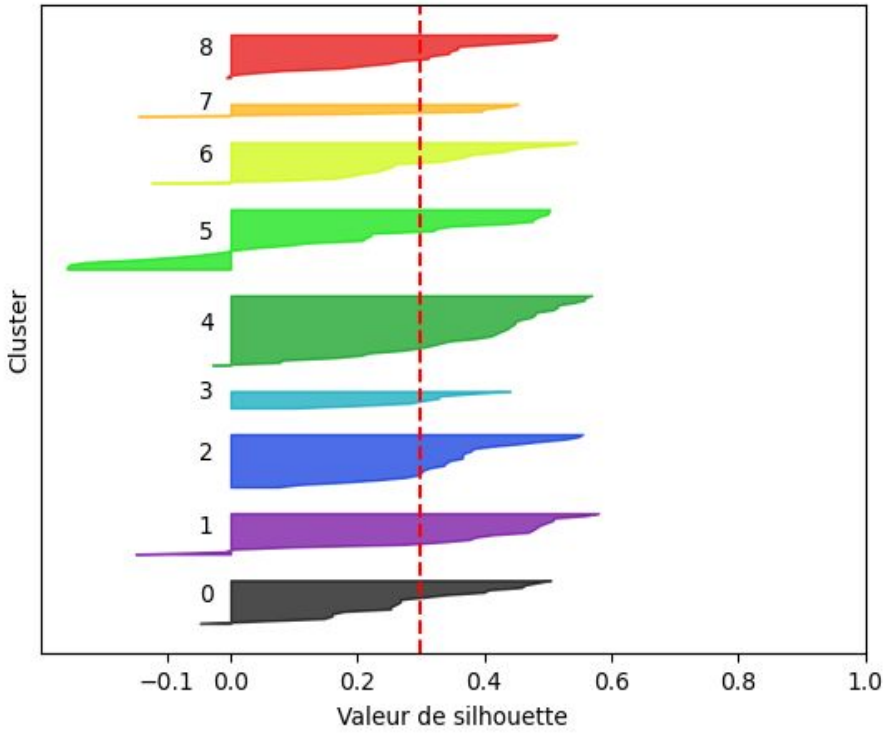
Kmeans

Score de silhouette moyen : 0.31700464694928726



ACH

Score de silhouette moyen : 0.2993195163227867

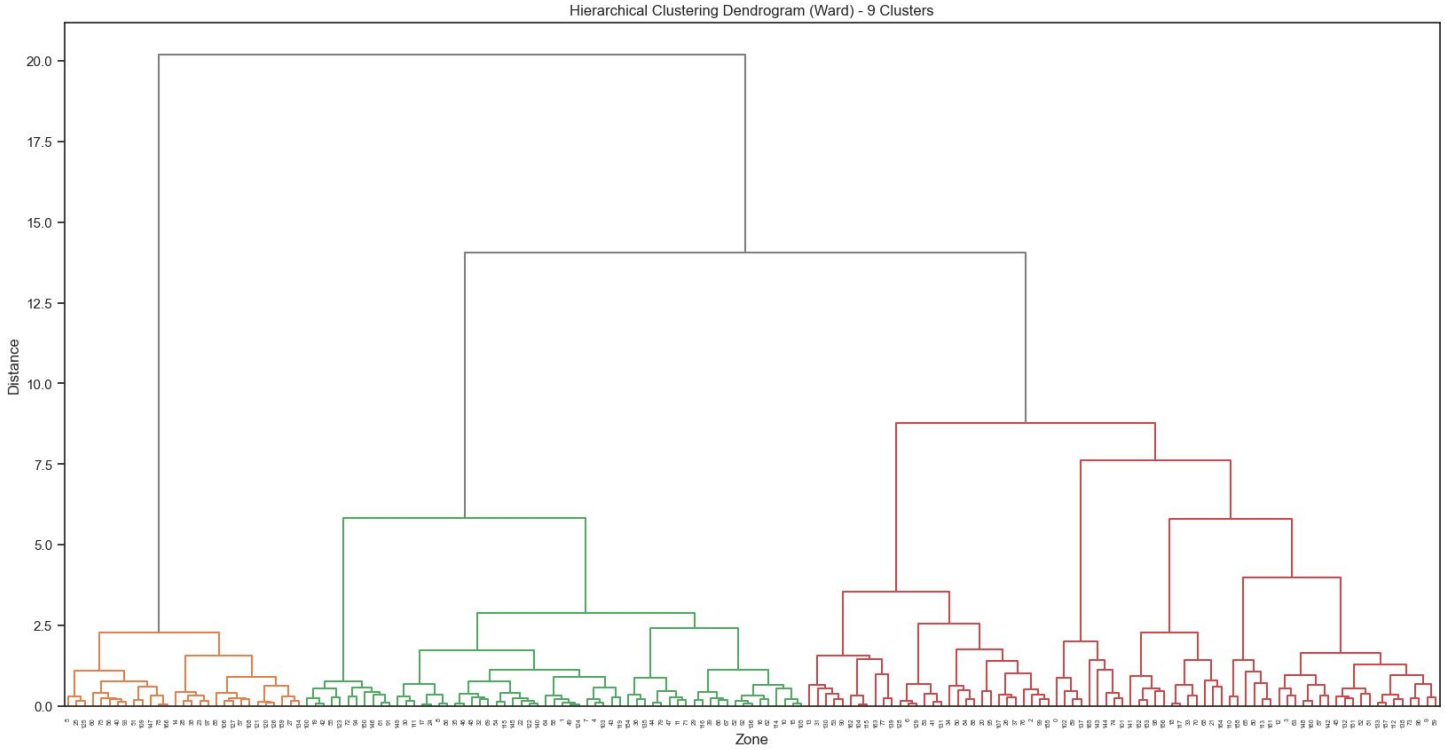


Classification Ascendante Hiérarchique

Sur le dendrogramme on retrouve les branches qui définissent nos clusters.

On va donc prendre en compte nos branches pour trouver les 9 clusters qui nous intéressent.

Le clustering s'effectue du bas vers le haut. Au départ chaqeu point est considéré comme un cluster puis on va fusionner les plus similaires jusqu'en haut.

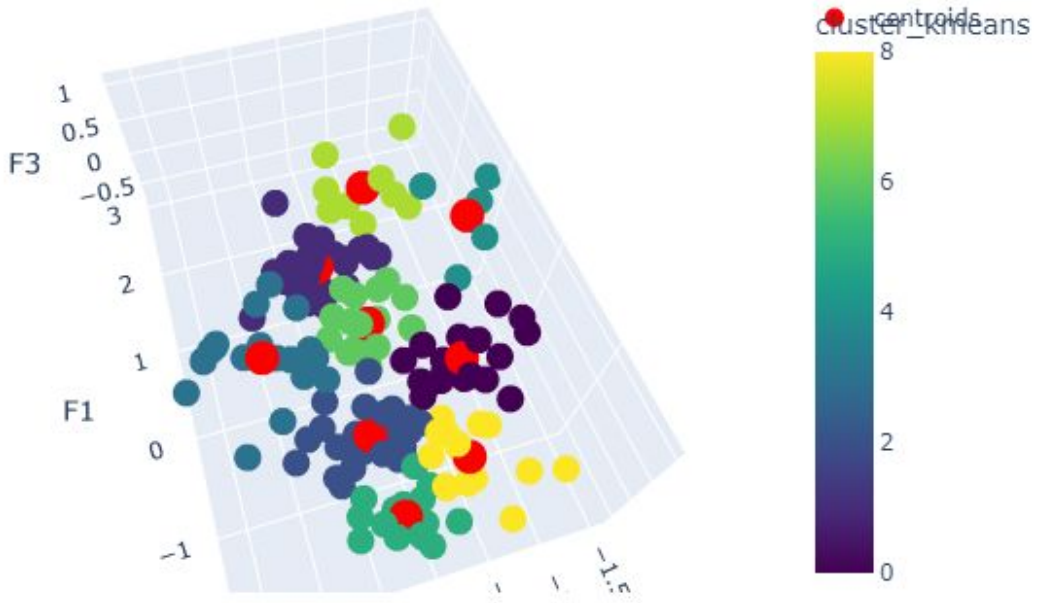


Kmeans

voici notre résultat du clustering K Means.

Les points rouges sont les centroids de chacun de nos clusters.

Nos clusters reste proche les uns des autres ce qui explique notre score de silhouette moyen faible.



Evaluation du clustering

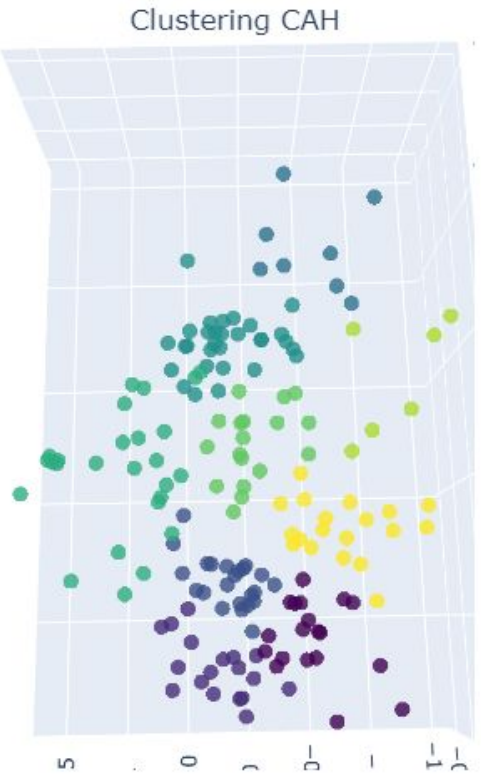
On va maintenant comparer nos deux méthodes de clustering.

On peut voir que les clusters ne sont pas identiques sur les graphqiues.

L'indice de rand permet mesurer la similarité entre deux ensembles de données.

Indice de Rand : 0.81

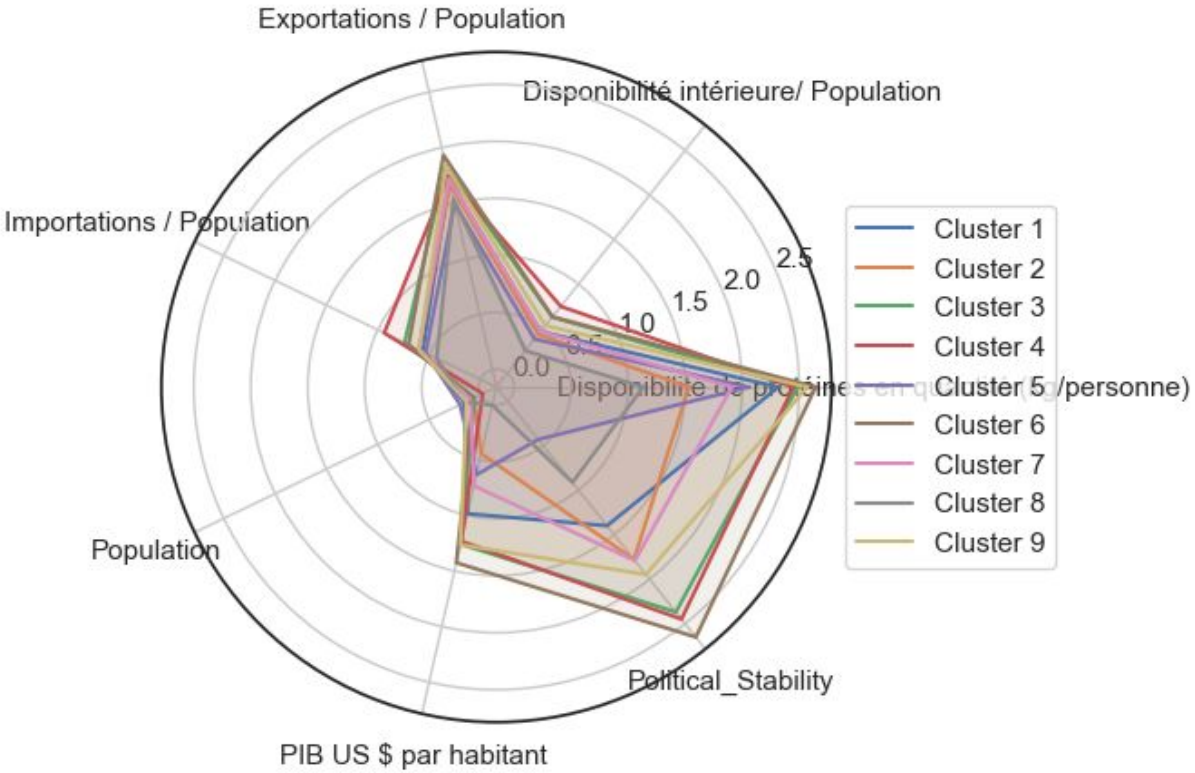
Un indice de rand proche de 1 indique que nos clusters sont relativement équivalent.



Evaluation du clustering

Grâce à ce graphique on peut voir que les différenciations se font principalement :

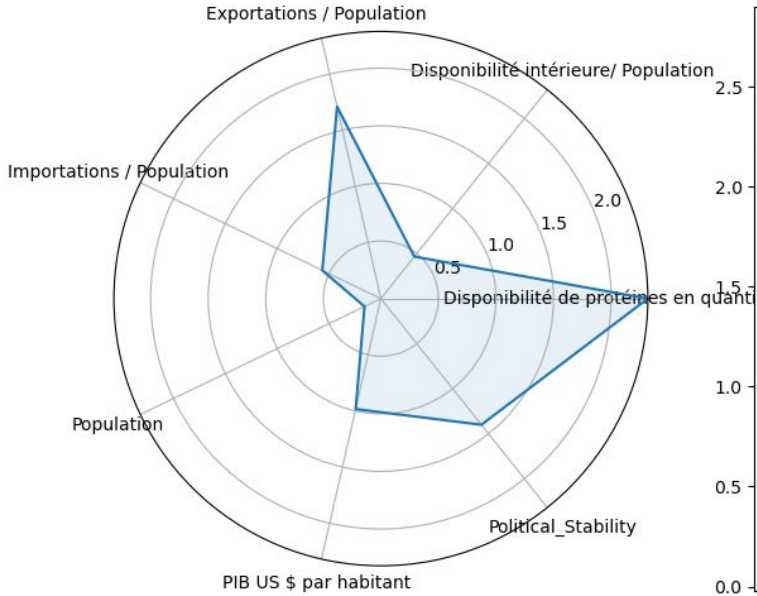
- La stabilité politique
- La disponibilité en protéines
- le PIB

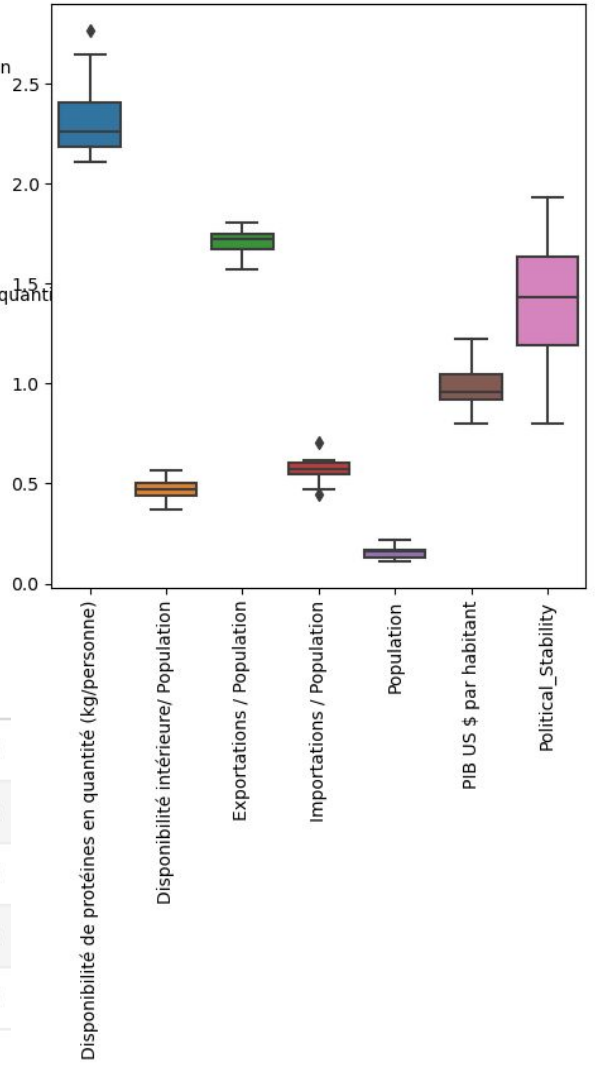


CLuster Kmeans 1

- Caractéristiques distinctives : Disponibilité de protéines, disponibilité intérieure, et exportations par habitant positives. Importations par habitant négatives. PIB par habitant et stabilité politique en dessous de la moyenne globale.

- Interprétation potentielle : Ce cluster représente des régions avec une forte disponibilité de protéines, des exportations significatives, mais une stabilité politique et un PIB par habitant relativement bas.



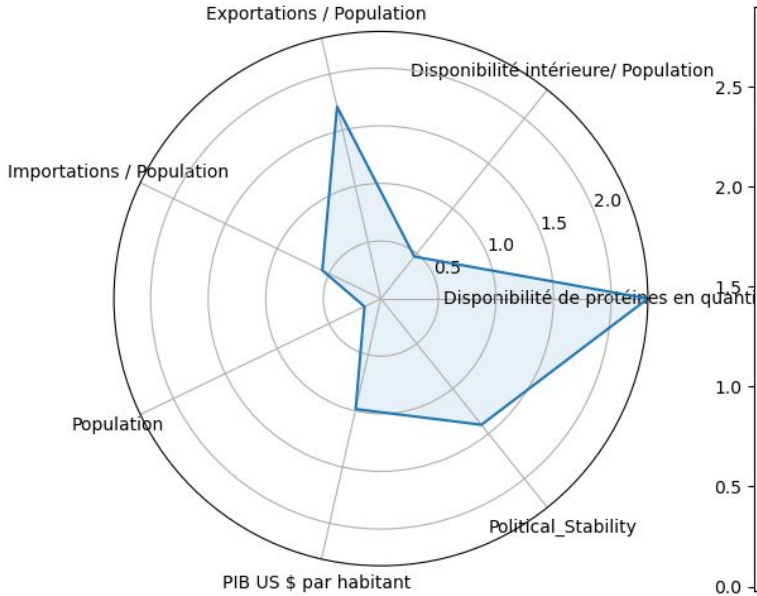


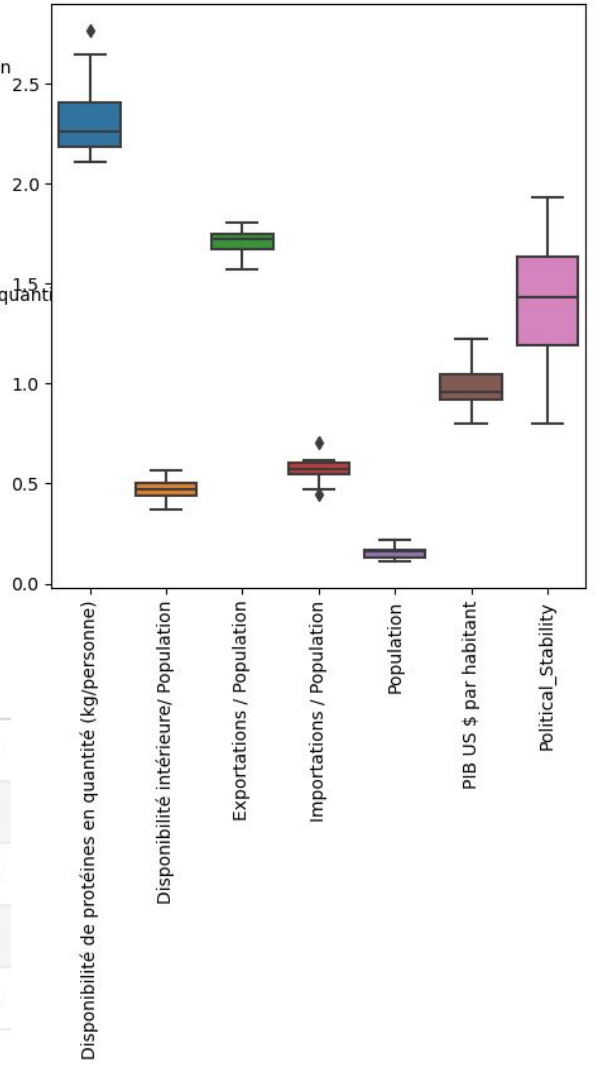
Pays	cluster_kmeans	cluster_dd
Bosnie-Herzégovine	0	9
Colombie	0	9
El Salvador	0	9
Guatemala	0	9
Honduras	0	9

Cluster Kmeans 5

- Caractéristiques distinctives : Disponibilité de protéines, disponibilité intérieure, et exportations par habitant positives. Importations par habitant négatives. PIB par habitant et stabilité politique en dessous de la moyenne globale.

- Interprétation potentielle : Ce cluster représente des régions avec une forte disponibilité de protéines, des exportations significatives, mais une stabilité politique et un PIB par habitant relativement bas.

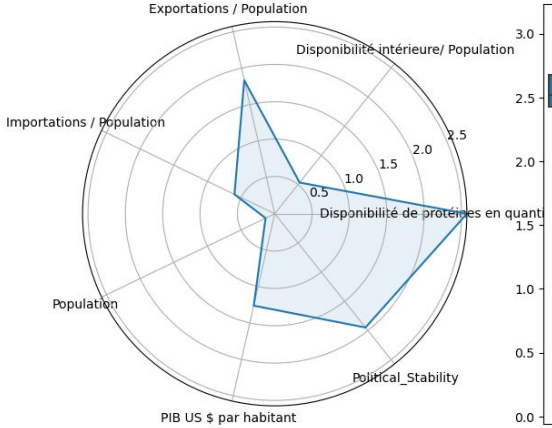


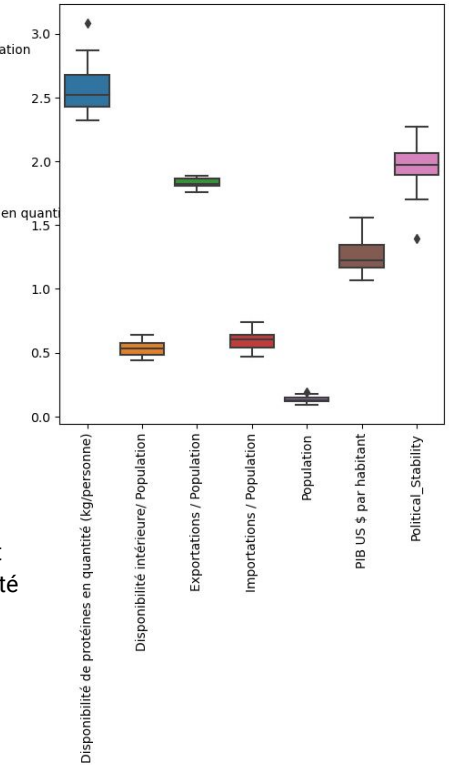


Pays	cluster_kmeans	cluster_dd
Bosnie-Herzégovine	0	9
Colombie	0	9
El Salvador	0	9
Guatemala	0	9
Honduras	0	9

Choix du cluster 9

	Pays	cluster_kmeans	cluster_dd
1	États-Unis d'Amérique	8	1
2	Trinité-et-Tobago	8	1
3	République dominicaine	8	1
4	Royaume-Uni de Grande-Bretagne et d'Irlande du...	8	1
5	Pérou	8	1
6	Panama	8	1
7	Malaisie	8	1
8	Israël	8	1
9	Iran (République islamique d')	8	1
10	Guyana	8	1
11	Fédération de Russie	8	1
12	Espagne	8	1
13	Bélarus	8	1
14	Brésil	8	1
15	Bolivie (État plurinational de)	8	1
16	Belize	8	1
17	Argentine	8	1
18	Afrique du Sud	8	1





- Caractéristiques distinctives : Disponibilité de protéines, disponibilité intérieure, exportations, et PIB par habitant positives. Importations et stabilité politique proches de la moyenne globale.

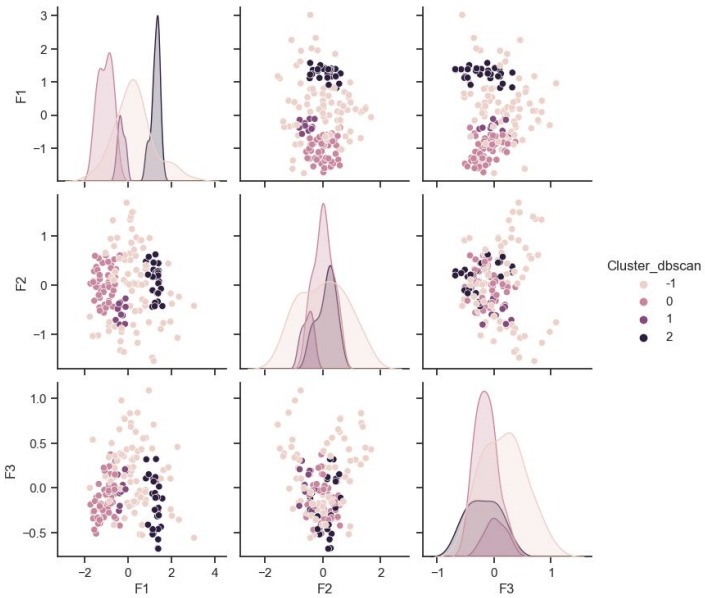
- Interprétation potentielle : Ce cluster représente des régions avec une forte disponibilité de protéines, des exportations significatives, un PIB par habitant élevé, et des importations et une stabilité politique relativement équilibrées.

Dbscan

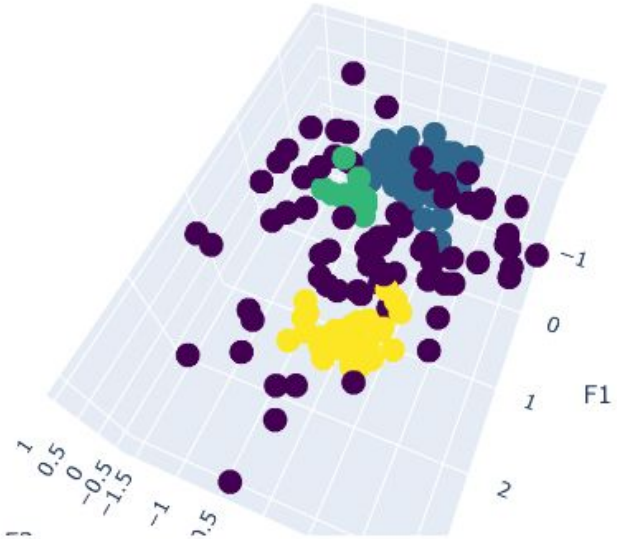
Au cours de mes recherches j'ai découvert le Dbscan que j'ai voulu essayer sur notre jeu de données.

J'obtiens donc ce clustering avec comme paramètre.

eps=0.3
min_samples=5



Graphique des clusters en 3D



Zone	Dbscan
Bosnie-Herzégovine	1
El Salvador	1
Guatemala	1
Honduras	1
Jordanie	1
Maroc	1
Mexique	1
Nicaragua	1
République de Moldova	1
Thaïlande	1
Tunisie	1