Aix Marseille Université - Campus Luminy

UFR des Sciences

Rapport de TP

Master Informatique

Module ISD : Introduction à la Science des Données.

TP n°1 :

Prise en main de Python, pandas et de Scikit-Learn.

Réalisé par :

ZEMMOURI Yasmine G3.

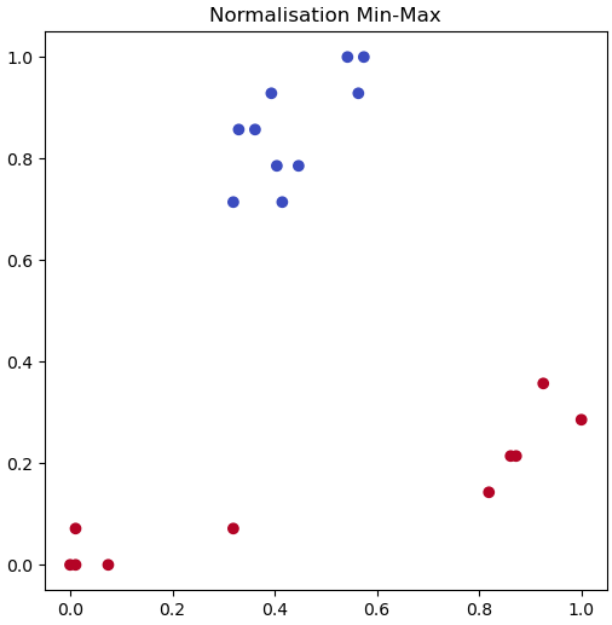
Partie 3 : Effet de normalisation :

- Pour implémenter les méthodes de normalisation vues en cours, je me suis servie de la documentation de la bibliothèque *numpy*, notamment des fonctions *numpy.mean(),* pour calculer la moyenne des données en entrées ainsi que d'autres fonctions.

Nuages de points normalisés pour chaque type de normalisation :

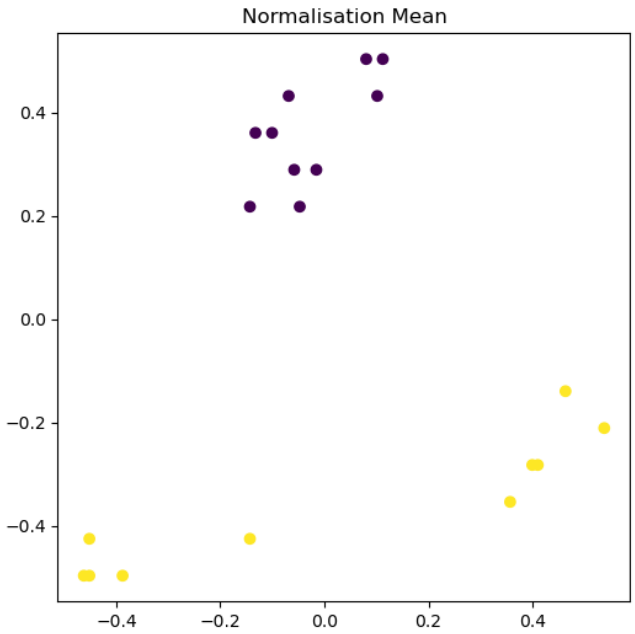
- ✓ Les graphiques de dispersion suivants représentent le jeu de données 'Fruits' normalisé de différentes manières. Chacun de ces graphiques met en évidence la distribution des points de données en fonction de deux attributs, Weight et Length, tout en les colorant en fonction de l'attribut 'Fruit'.

Méthode 1 : Normalisation min_max



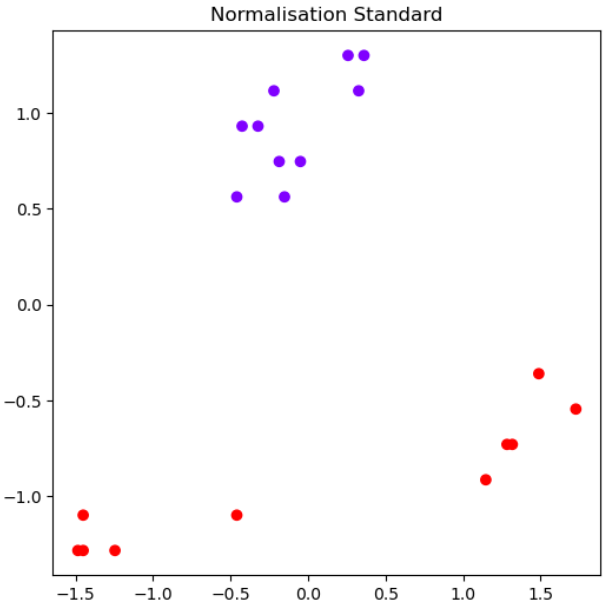
- ✓ Dans cette méthode, les valeurs de chaque attribut sont transformées pour qu'elles se situent dans une plage comprise entre 0 et 1. Le nuage de points montre une forte concentration de points autour de ces valeurs limites, mais ils varient beaucoup en dehors de ces bornes.

Méthode 2 : Normalisation moyenne



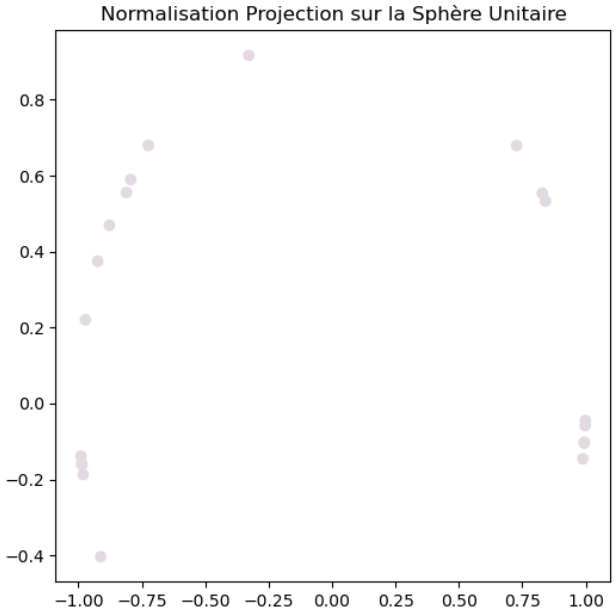
✓ Cette méthode soustrait la moyenne de chaque attribut et la divise par la plage des valeurs. Cela centre les données autour de zéro et les étend uniformément. Le graphe montre une dispersion homogène des points sur toute la plage de valeurs.

Méthode 3 : Standardisation



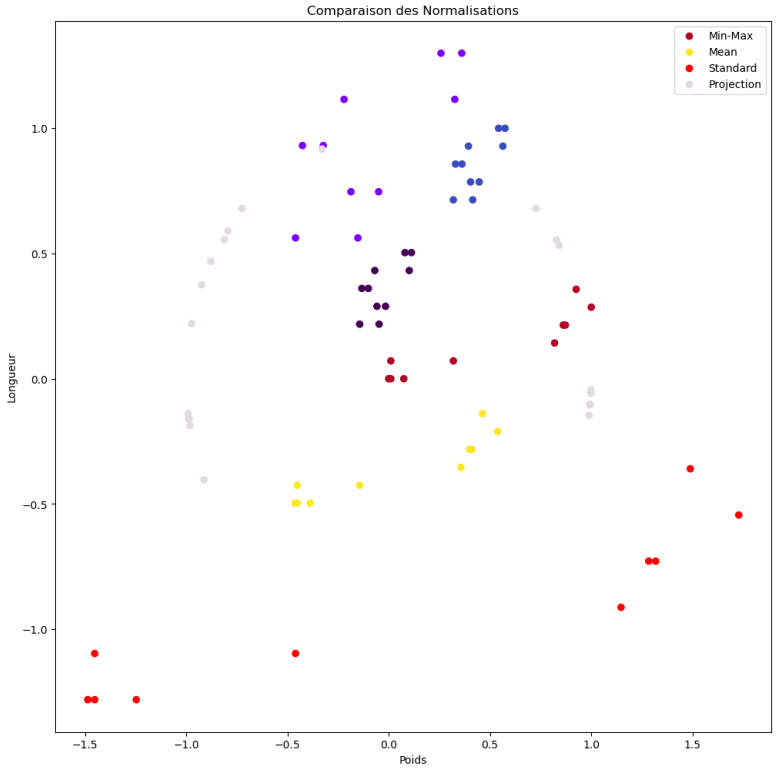
✓ Cette méthode centre les données autour de zéro en soustrayant la moyenne et en les divise par l'écart type. Le graphe montre une dispersion similaire à la normalisation par la moyenne, mais les valeurs sont ajustées en fonction de l'écart type.

Méthode 4 : Projection normalisée sur la surface de la sphère unitaire



✓ Cette méthode projette les données sur une sphère unitaire, préservant la direction des vecteurs. Les graphiques montrent une dispersion uniforme des points, formant une sphère.

Les quatre types en un seul nuage de points :



✓ La normalisation, quelle que soit la méthode choisie, préserve la dispersion naturelle des données. Cependant, ce qui change, c'est la gamme de valeurs possibles pour chaque caractéristique. Cette gamme peut varier selon la méthode de normalisation. Par conséquent, cela affecte la comparaison entre deux caractéristiques qui partagent le même ordre de grandeur, selon que leurs valeurs sont normalement distribuées ou non.

- ✓ La normalisation rend les données comparables de manière équitable en ajustant la portée des valeurs, tout en maintenant la variation naturelle.

Partie 5 : ACP

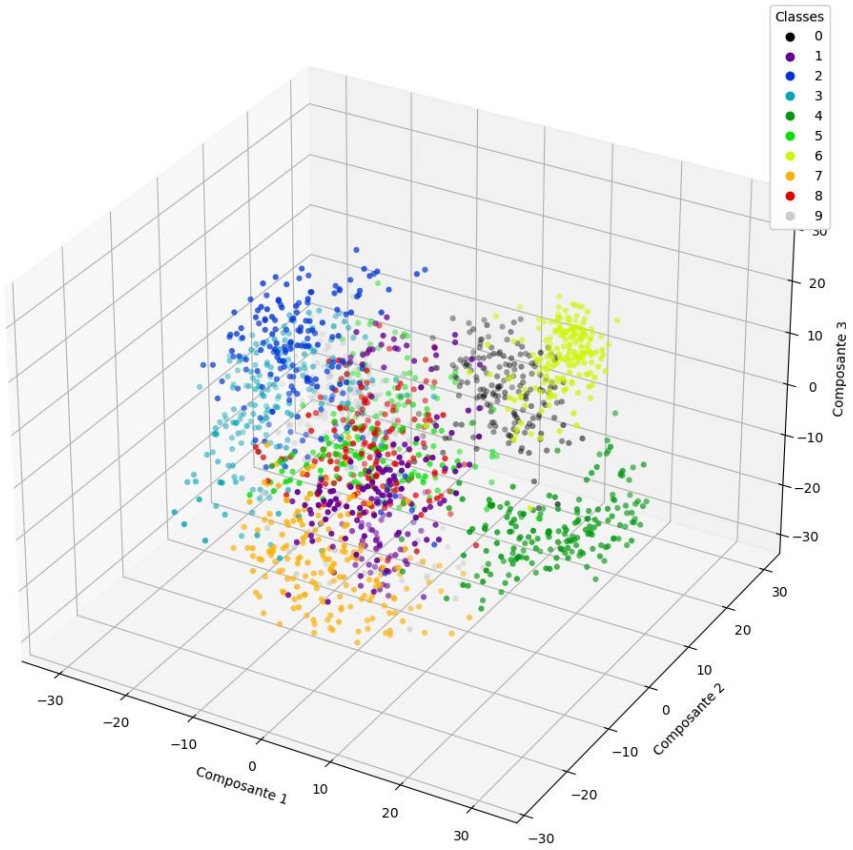
- ➢ Analyse du résultat obtenu après projection des données dans un espace de dimension 2 en utilisant l'ACP :

Certaines classes forment des groupes denses et homogènes, comme l'orange pour le chiffre 7 ou le violet pour le chiffre 1. En revanche, d'autres classes sont plus dispersées, par exemple, le vert pour le chiffre 5 ou le bleu clair pour le chiffre 3.

On observe également que certaines classes sont distinctes et ne se mélangent pas beaucoup avec les autres (par exemple, le chiffre 0 en noir), tandis que d'autres peuvent provoquer des confusions avec de nombreuses autres classes (par exemple, la classe 8 qui se superpose avec le bleu (classe 3) et le vert (classe 4 et 5)).

Il n'est donc pas possible de distinguer des frontières claires entre les 10 classes.

- ➢ En projetant les données dans un espace de dimension 3, on obtient le résultat suivant sur un espace 3D :



En utilisant trois composantes pour la visualisation, la séparation entre la plupart des clusters est plus nette, ce qui réduit les confusions entre les classes. L'ajout d'une troisième dimension offre une vue plus complète sur les relations et les différences entre les classes.

On peut observer que par exemple la classe 4 est nette par rapport aux autres classes. Il reste cependant une confusion pour les chiffres 8 et 5.

Partie 6 : Données manquantes

- En utilisant la documentation de numpy, il existe une fonction *numpy.isnan()* qui vérifie si une donnée est nulle ou pas. Son utilisation est bénéfique pour le calcul des données manquantes.
- Nous avons 1232 données manquantes globalement, 17 pour la première colonne, 13 pour la deuxième, 26 pour la troisième, 22 pour la quatrième...etc.
- Pour l'implémentation des méthodes de complétion stationnaire et algorithme KPP, la documentation de *sklearn* a été utile : il existe une librairie sklearn.impute contenant un ensemble de méthodes de complétion dont **SimpleImputer** et **KNNImputer**.
- Ces méthodes prennent en entrée le dataset creux, et renvoie un dataset sans données manquantes.
- Pour la méthode de complétion par combinaison linéaire, il suffit d'utiliser les nombreuses fonctions de numpy.
- ✓ En calculant l'écart moyen au carré (MSE) pour chaque méthode, la méthode des k-plus proches voisins présente la plus petite erreur, ce qui en fait la méthode la plus performante. Ensuite, la méthode basée sur la moyenne conditionnellement aux classes est la deuxième plus performante, suivie de la méthode stationnaire :
  - ➔ KNN Imputer avec k=5 : 0.06672579298831385.
  - ➔ Combinaison Linéaire : 0.1728107461701899.
  - ➔ Mode : 0.6365470228158041.
- ✓ Même en variant le nombre de voisins **k** de la méthode KPP, cette méthode reste la plus performante avec k=3 : 0.05151332467693069.
- ✓ La comparaison de ces méthodes dépend entièrement du jeu de données.