TP 1 : Analyse de Données et Méthodes d'Ensemble

ABCI Fella

28/03/2025

Partie 1 : Analyse exploratoire des données

Exercice 1 : Statistiques descriptives

	poids	nourriture	température
0	3.382026	1.764945	24.261636
1	2.700079	0.795672	24.521242
2	2.989369	0.818855	27.199319
3	3.620447	1.490819	26.310527
4	3.433779	0.848063	26.280263
..	...	...	...
95	2.853287	1.148536	27.273783
96	2.505250	1.431537	25.195450
97	3.392935	1.447051	26.165907
98	2.563456	1.848971	24.201102
99	2.700995	1.600958	25.740112

-Poids :

Moyenne : 2.53 kg
Médiane : 2.55 kg
Écart-type : 0.51

-> Les poids sont assez centrés autour de la moyenne ce qui nous donne une répartition équilibrée. La médiane et la moyenne sont très proches donc la distribution semble assez symétrique

-Nourriture :

Moyenne : 1.22 kg
Médiane : 1.21 kg
Écart-type : 0.31

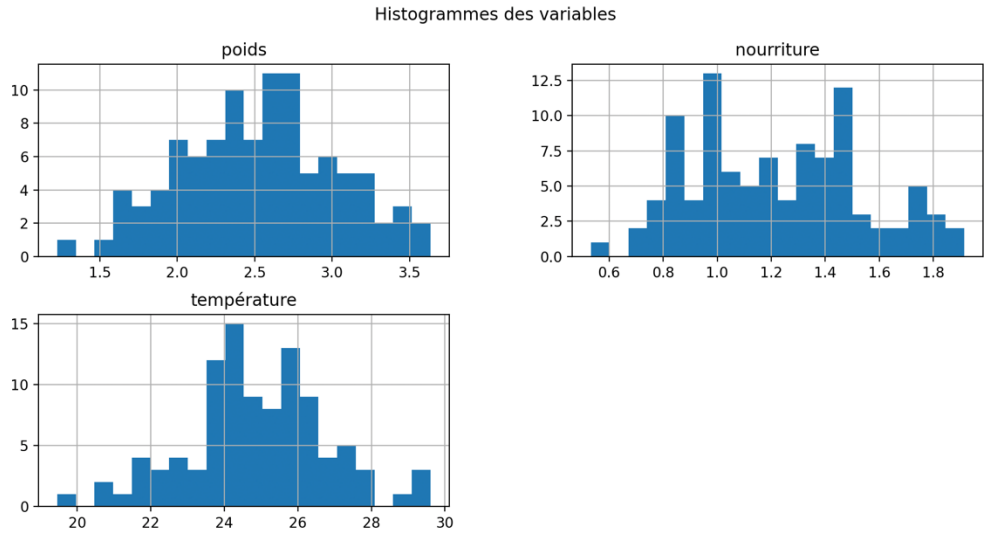
Les données sont un tout petit peu plus dispersées mais là aussi la moyenne et la médiane sont proches et la distribution est globalement équilibré même s'il y a quelques valeurs plus extrêmes

-Température :

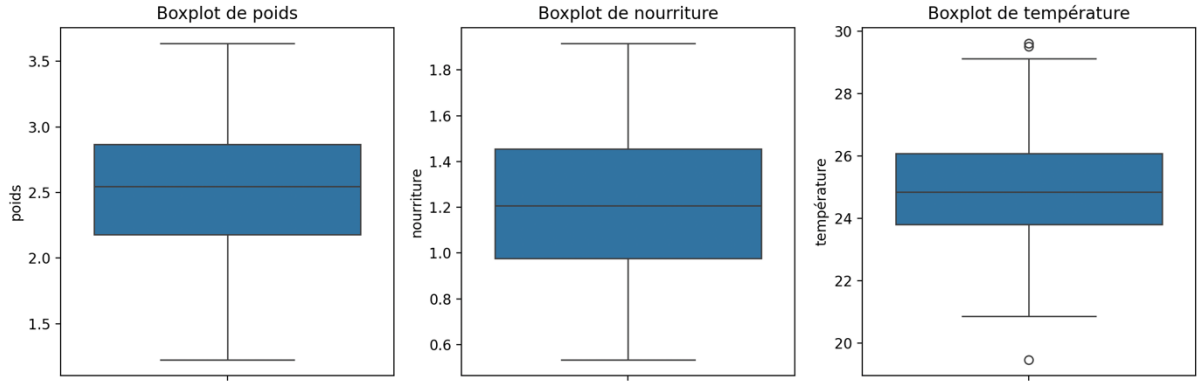
Moyenne : 24.88 °C
Médiane : 24.85 °C
Écart-type : 1.91

La température est concentrée autour de 25 °C. ici aussi la moyenne et médiane sont proches.

ça confirme la stabilité des données, il n'y a pas de grandes différences entre les valeurs basses et hautes



-Le poids et la température suivent une forme proche d'une courbe en cloche -> distribution normale, qui est cohérent avec les statistiques observées
-La variable nourriture a l'air un peu plus irrégulière avec plusieurs petits pics. ça pourrait indiquer des comportements différents parmi les poulets peut etre que certains mangent plus que d'autres



-Poids :
La majorité des poids se situent entre environ 2.0 et 3.0 kg il n'y a pas de valeurs aberrantes visibles La distribution a l'air équilibrée autour de la médiane (~2.5 kg)
-Nourriture :
Les valeurs de nourriture sont concentrées entre 0.9 kg et 1.5 kg. la boîte est légèrement étirée mais aucun outlier n'est visible
-Température :
On observe quelques outliers en dessous de 21 °C et au-dessus de 28 °C. ça peut correspondre à données inhabituels ou des erreurs de mesure.

Exercice 2 : Détection des outliers

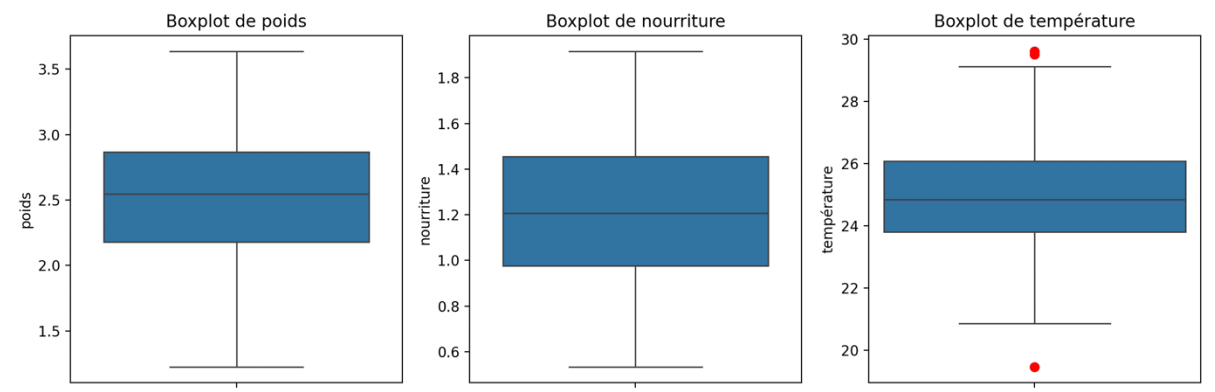
-Detection des outiers avec la méthode de l'écart interquartile (IQR) :
Outliers - Poids : 0

Outliers - Nourriture : 0
Outliers - Température : 3

-Avec la méthode du Z-score :
Z-Score - Outliers poids : 0
Z-Score - Outliers nourriture : 0
Z-Score - Outliers température : 0

-> Les deux methodes ne donnent pas les memes resultats : L'IQR est plus sensibles aux petites déviations ce qui explique pourquoi il detecte 3 outliers pour la temperature
Le Z-score ne considere pas une valeur comme aberrante si elle s'eloigne plus de 3 écart types de la moyenne, dans notre cas les temperatures extremes ne depassent pas ce seuil

Visualisation :



Les points rouges sont des outliers ils ne sont pas tres eloignés donc ce sont des valeurs inhabituelles mais coherentes donc a ne pas exclure

Exercice 3 : Tests paramétriques
Shapiro-wilk
Le test de Shapiro-wilk a été appliqué sur les trois variables :
poids ➜ Statistique = 0.9927, p-value = 0.8689
nourriture ➜ Statistique = 0.9735, p-value = 0.0416
température ➜ Statistique = 0.9911, p-value = 0.7545

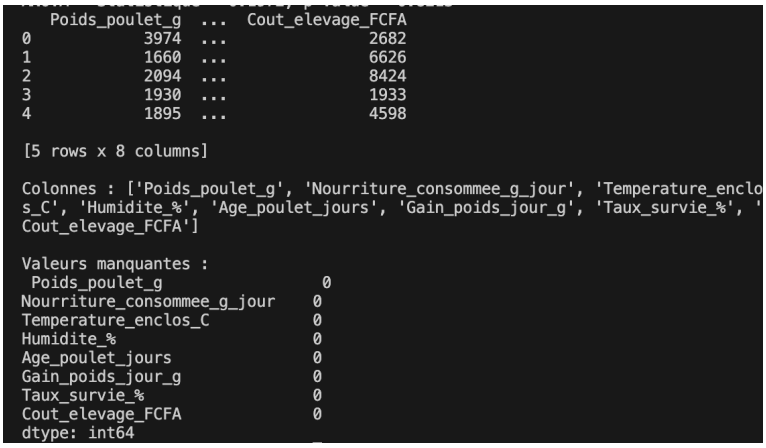
-> La p-value > 0.05 on considére que la distribution est normale (on ne rejette pas l'hypotherse de normalité)

Test de Student
Dans le test t de student permet de comparer les moyennes de poids entre deux groupes de poulets
Test t ➜ Statistique = -1.1196, p-value = 0.2656
-> la p-value est supérieure à 0.05 on considere que la difference n'es pas significative

ANOVA
ANOVA ➜ Statistique = 0.1972, p-value = 0.8213
->p-value<0.05 -> il n y a pas de lien fort entre temperature et poids

Partie 2 : Réduction de dimensionnalité

Exercice 4 : Analyse en Composantes Principales (ACP)



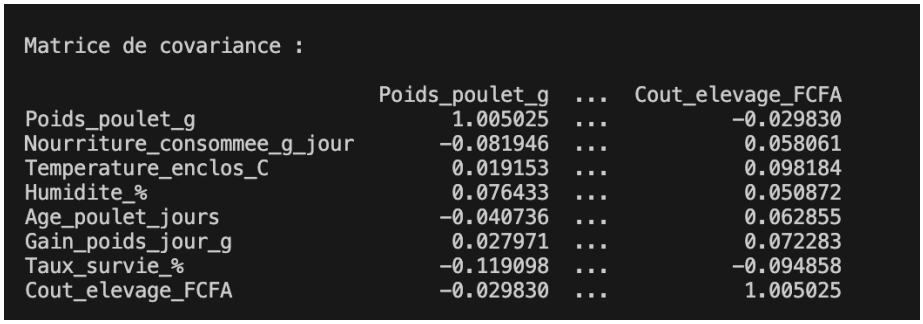
```
   Poids_poulet_g  ...  Cout_elevage_FCFA
0            3974  ...               2682
1            1660  ...               6626
2            2094  ...               8424
3            1930  ...               1933
4            1895  ...               4598

[5 rows x 8 columns]

Colonnes : ['Poids_poulet_g', 'Nourriture_consommee_g_jour', 'Temperature_enclo
s_C', 'Humidite_%', 'Age_poulet_jours', 'Gain_poids_jour_g', 'Taux_survie_%', '
Cout_elevage_FCFA']

Valeurs manquantes :
 Poids_poulet_g                0
Nourriture_consommee_g_jour    0
Temperature_enclos_C           0
Humidite_%                     0
Age_poulet_jours               0
Gain_poids_jour_g              0
Taux_survie_%                  0
Cout_elevage_FCFA              0
dtype: int64
```

-> Les données semblent propres sans valeurs manquantes
-> les données ont été centrées et réduites pour que chaque variable ait une moyenne nulle et une variance =1



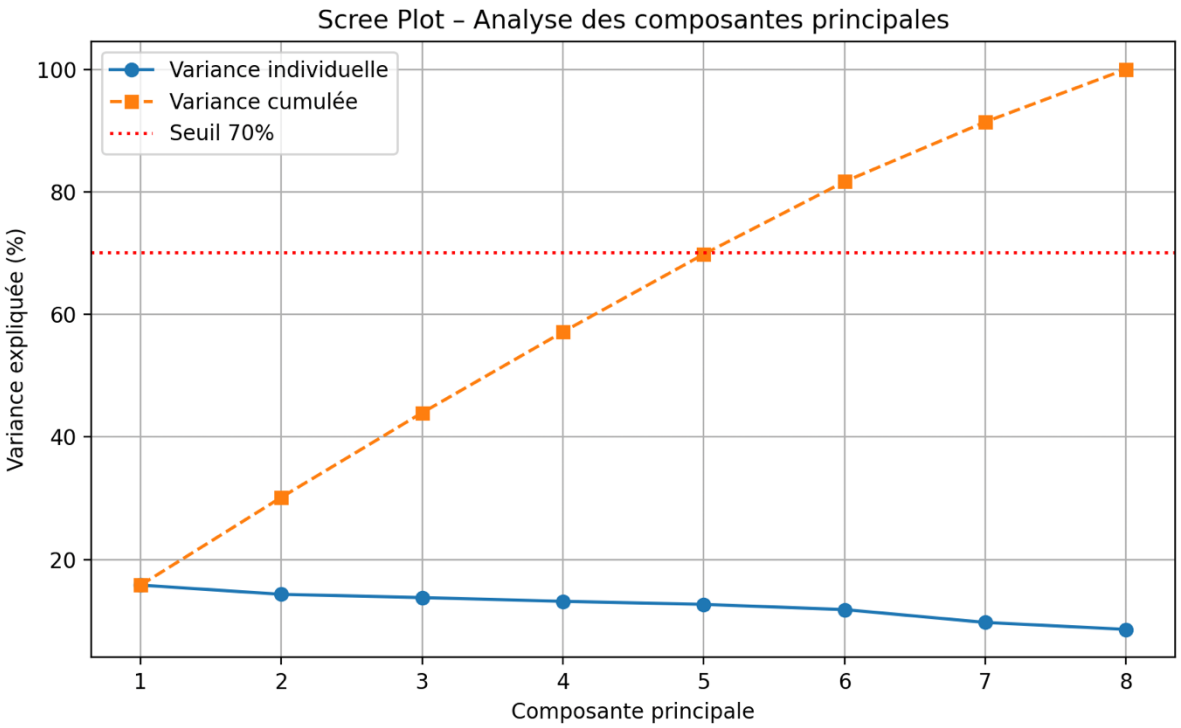
```
Matrice de covariance :

                           Poids_poulet_g  ...  Cout_elevage_FCFA
Poids_poulet_g                   1.005025  ...          -0.029830
Nourriture_consommee_g_jour     -0.081946  ...           0.058061
Temperature_enclos_C             0.019153  ...           0.098184
Humidite_%                       0.076433  ...           0.050872
Age_poulet_jours                -0.040736  ...           0.062855
Gain_poids_jour_g                0.027971  ...           0.072283
Taux_survie_%                   -0.119098  ...          -0.094858
Cout_elevage_FCFA               -0.029830  ...           1.005025
```

->On obtient les valeurs et vecteurs propres qui représentent l'importance et directions principales

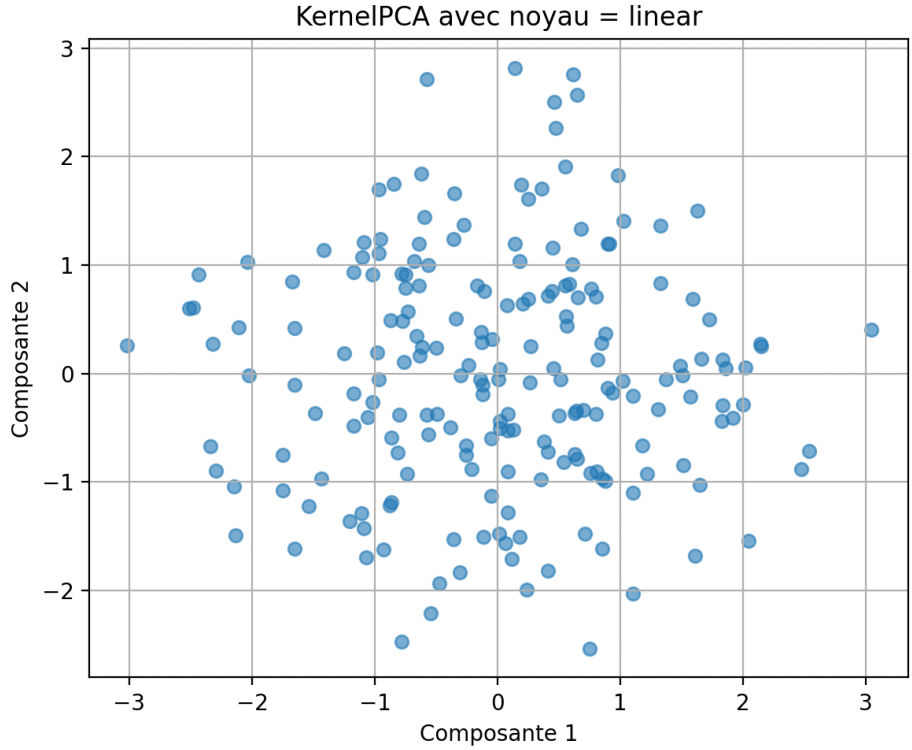
Les 2 premières composantes expliquent 30.16 %
Les 4 premières : 57.29 %
Les 5 premières : 70.03 %
Les 6 premières : 81.94 %

-> On garde les 5 premieres composantes car elles permettent de conserver environ 70% de la variance totale
Ce seuil est generalement consideré bon
Apres les 5 premieres composantes les gains deviennent plus faibles

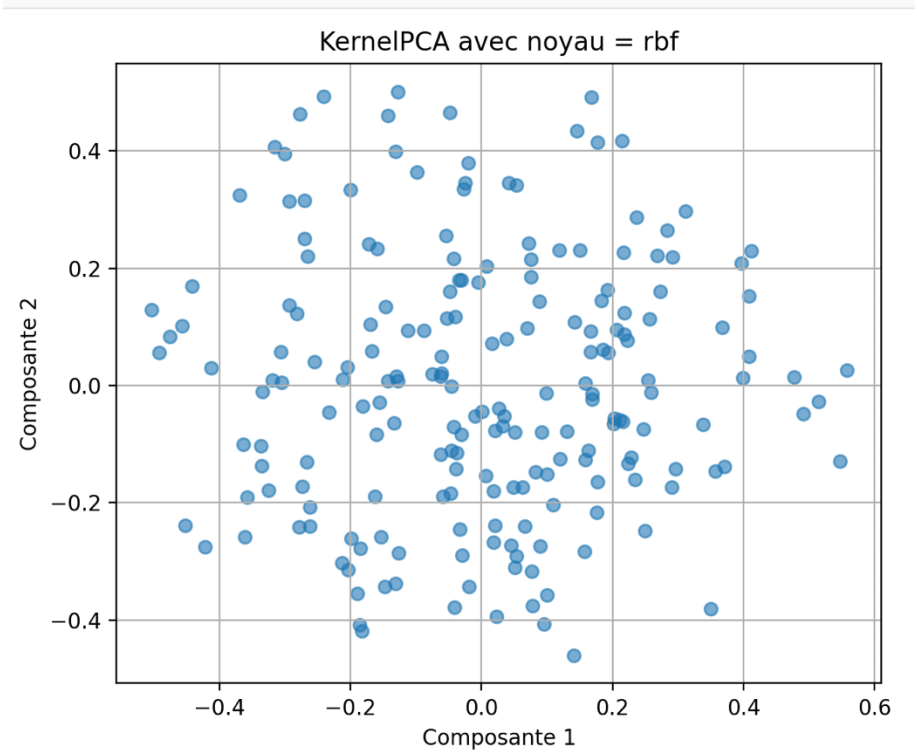


-> Ce plot montre la part de la variance expliqué par chaque composante
On voit un palier apartir du 5eme composant

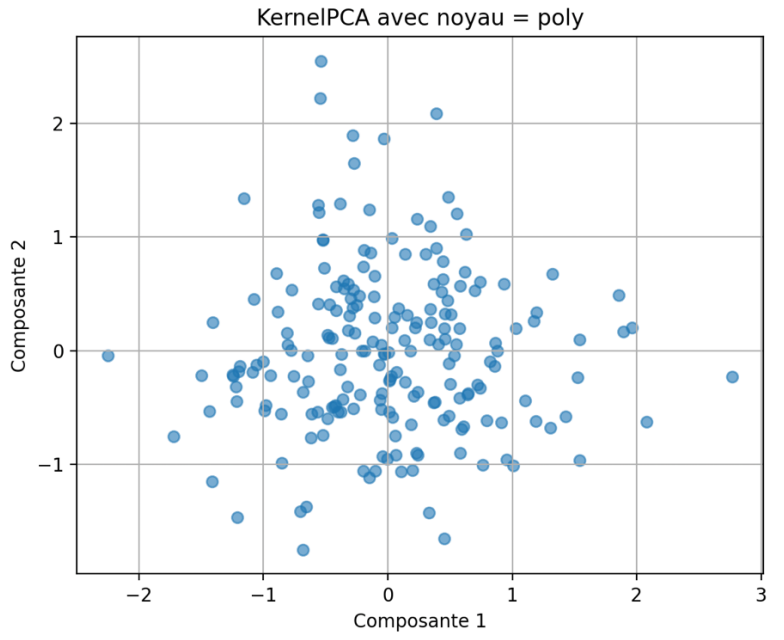
Exercice 5 : ACP à Noyau



-> on a un résultat presque identique a l'ACP classique, parce qu'on ne transforme pas l'espace des données, on conserve les relations lineaires



->le noyau RBF est non lineaire , donc ça transforme l'espace des données pour mieux faire ressortir des structures cachées
On observe une forme plus compacte que le noyau lineaire, les valeurs sont concentrés entre -0.5 et 0.5



->On observe une forme plus concentré autour de 0 avec quelques points eloignés
Il semble avoir plus de dipersion que dans le noyau RBF ce qui indique que le noyau polynomial transorme encore plus l'espace mais sans faire apparaitre de groupes tres nets non plus

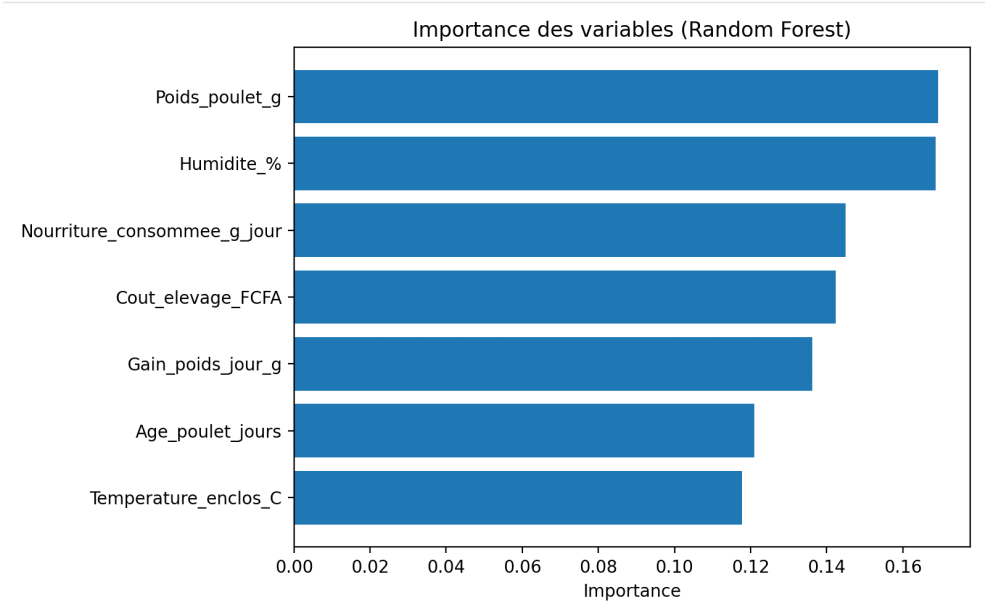
->Dans notre cas aucun noyau ne montre un clustering evident mais le RBF semble le plus adapté pour la separation non lineaire

Partie 3 : Méthodes d'ensemble

Exercice 6 : Bagging (3 points)

Identification des variables
Poids_poulet_g
Nourriture_consommee_g_jour
Temperature_enclos_C
Humidite_%
Age_poulet_jours
Gain_poids_jour_g
Cout_elevage_FCFA

->on a utilisé randomforestclassifier pour prédire si un lot de poulets aura un taux de survie >= 90%
Résultats :
Accuracy : 58.3 %
F1-score : 56.1 %
La précision et le rappel sont relativement équilibrés entre les deux classes
Classe 0 (faible survie) : f1-score = 60 %
Classe 1 (bonne survie) : f1-score = 56 %
->le modele n'est pas parfait mais on peut capturer quelques classes avec



les variables les plus importantes.
Poids_poulet_g et Humidité_% sont les deux variables les plus décisives
La nourriture consommée, le coût d'élevage et le gain de poids jouent aussi un rôle important

Des facteurs comme l'âge ou la température de l'enclos sont moins décisifs.

Exercice 7 : Boosting

AdaBoost vs Gradient Boosting :

AdaBoost
MSE : 21.446267588146572
R^2 : -0.20238767846351124

Gradient Boosting
MSE : 27.056741650228965
R^2 : -0.5169396094631744

->MSE indique l'écart moyen entre les valeurs prédites et réelles, plus il est bas mieux c'est
-> R^2 négatif signifie que le modele fait pire qu'une simple moyenne

Dans notre cas aucun des deux modeles parvient a bien predire le gain de poids

Q14. Les outliers peuvent beaucoup affecter les performances des algorithmes de boosting
Parceque :

- AdaBoost est très sensibles aux erreurs, il donne plus de poids aux observations al prédites, si un outlier est mal prédit Adaboost va s'acharner dessus er ça va deséquilibrer le modele
- Gradient boosting est un peu plus robuste car il corrige progressivement les erreus mais les valeurs aberrantes peuvent tout de meme l'influencer

->ça peut expliquer en partie les faibles scores qu'on a obtenus ici. Pour ameliorer on peut filtrer ou corriger les outliers avant entrainement des données