

ATDN2 TP1

Remy XU M1 OIVM

26 March 2025

Partie 1 : Analyse exploratoire des données

Avant de commencer :

Analysons notre jeu de données sur l'élevage de poulet dans le fichier donnees_elevage_poulet.csv.

On affiche les valeurs de notre jeu de données:

	A	B	C	D	E	F	G	H
1	Poids_poulet_g	Nourriture_consommee_g_jour	Temperature_enclos_C	Humidite_%	Age_poulet_jours	Gain_poids_jour_g	Taux_survie_%	Cout_elevage_FCFA
2	3974		52.27.6	79.3		24.12.0	81.1	2682
3	1660		152.31.7	62.5		42.12.2	89.1	6626
4	2094		186.30.1	64.8		29.18.8	90.4	8424
5	1930		111.29.2	87.0		63.13.8	92.9	1933
6	1895		100.26.1	78.2		21.5.5	93.0	4598
7	3892		108.26.1	60.3		32.18.7	87.3	2980
8	2438		167.25.4	63.0		59.8.7	91.2	4056
9	2969		145.31.1	79.9		21.13.7	89.6	9984
10	1266		162.29.2	60.2		103.7.5	97.7	1791
11	2038		111.30.0	64.8		84.5.5	90.6	6390
12	1130		101.25.1	76.5		82.9.7	88.8	7635
13	2282		61.31.8	80.8		92.16.7	88.1	3255
14	2935		88.30.8	79.6		36.9.2	91.4	4610
15	3971		179.26.5	66.7		28.8.3	96.1	2607
16	3719		180.26.3	81.4		94.8.2	90.8	6179
17	930		162.26.3	67.1		34.12.7	93.2	7573
18	2485		150.27.1	69.8		43.19.6	94.8	1984
19	1569		162.28.7	82.4		57.11.9	90.4	6776
20	3191		130.28.0	79.5		54.13.4	88.5	4676
21	2315		162.27.0	85.5		113.17.9	97.5	4380
22	3653		51.29.3	79.7		114.13.0	88.4	9711
23	3233		179.26.0	77.0		68.7.8	89.2	9782
24	2015		103.27.0	62.8		88.9.5	99.8	2050
25	1755		136.27.6	71.0		81.9.6	80.0	4633
26	3124		178.28.2	68.0		79.11.0	83.7	7342
27	1964		196.30.5	67.3		69.11.4	87.7	1959
28	1259		175.26.4	89.2		97.17.0	98.7	7059
29	821		179.28.6	71.8		94.10.2	81.4	9884
30	3100		102.29.1	86.8		28.12.0	80.2	6655
31	1547		117.25.3	78.9		53.14.4	81.1	5473
32	3704		172.29.3	83.8		95.10.7	81.8	3946
33	1274		194.28.2	75.1		118.17.5	80.8	6725
34	1882		87.25.5	77.3		54.13.8	89.6	1526
35	3358		73.31.6	74.8		20.9.4	91.6	4224
36	2847		118.31.8	65.9		59.15.7	85.4	2875
37	3547		165.30.7	81.7		83.12.9	88.0	2763
38	1775		147.27.1	68.4		41.13.0	81.8	9712
39	2606		188.25.7	60.7		79.12.2	86.7	6797
40	969		193.29.8	79.4		83.12.5	90.5	8099
41	3805		146.28.1	65.3		112.16.5	94.6	6109
42	3534		173.25.9	88.2		91.6.5	80.1	7644
43	3805		119.28.5	88.6		30.10.0	89.3	1116
44	1362		142.25.2	87.4		33.6.1	85.9	3433

Figure 1: Affichage de notre jeu de données

Analysons les variables. On a les suivantes :

- Poids_poulet_g
- Nourriture_consommee_g_jour
- Temperature_enclos_C
- Humidite_%
- Age_poulet_jours
- Gain_poids_jour_g
- Taux_survie_%
- Cout_elevage_FCFA

On remarque qu'il y a que des variables quantitatives. Il n'y a donc pas besoin de transformer les variables qualitatives en étiquettes (label).

Exercice 1.1

Voici les données statistiques telles que la moyenne, la médiane, l'écart type, la variance, et les quantiles (25%, 50%, 75%) des variables poids, nourriture et température :

```

Moyenne du poids : 2509.58
Moyenne de la nourriture consommee : 129.745
Moyenne de la temperature de l'enclos : 28.389

Mediane du poids : 2481.5
Mediane de la nourriture consommee : 135.5
Mediane de la temperature de l'enclos : 28.5

Ecart-type du poids : 898.4368746263937
Ecart-type de la nourriture consommee : 44.00616648200808
Ecart-type de la temperature de l'enclos : 2.0657238623245084

Variance du poids : 807188.8176884422
Variance de la nourriture consommee : 1936.542688442211
Variance de la temperature de l'enclos : 4.2672150753768845

Quantiles du poids :
0.25    1810.75
0.50    2481.50
0.75    3356.50
Name: Poids_poulet_g, dtype: float64
Quantiles de la nourriture consommee :
0.25     95.75
0.50    135.50
0.75    165.25
Name: Nourriture_consommee_g_jour, dtype: float64
Quantiles de la temperature de l'enclos :
0.25     26.6
0.50     28.5
0.75     30.3
Name: Temperature_enclos_C, dtype: float64

```

Figure 2: Données statistiques du poids, nourriture, et température

Exercice 1.2

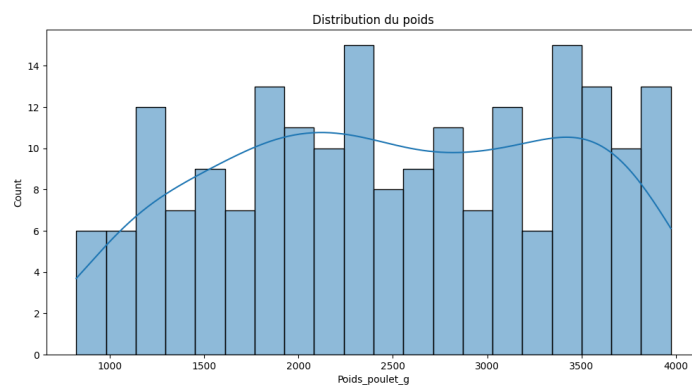


Figure 3: Histogramme poids

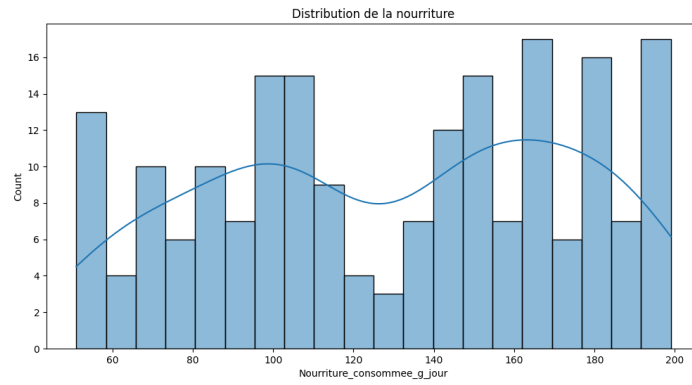


Figure 4: Histogramme nourriture

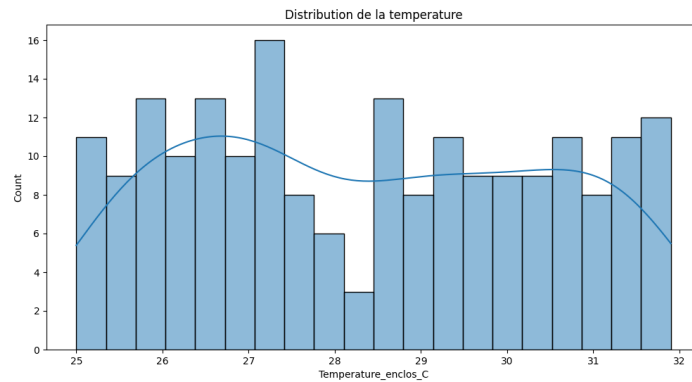


Figure 5: Histogramme temperature

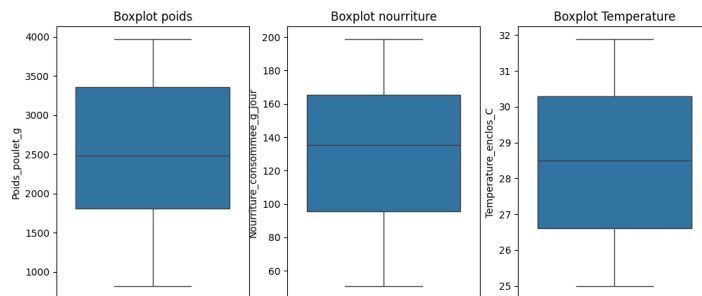


Figure 6: Boites a moustaches du poids, de la nourriture, et de la température

On remarque qu'il n'y a pas d'outliers detectes pour l'ensemble des trois donnees.

Les valeurs sont majoritairement reparties entre :

- Poids : 1800 et 3300 pour une valeur mediane de 2500. $\pm \frac{800}{2500} = 32\%$
- Nourriture : 95 et 165 pour une valeur mediane de 137. $\pm \frac{30}{137} = 21,9\%$
- Temperature : 26,7 et 30,3 pour une valeur mediane de 28,5. $\pm 1,8 \frac{1,8}{28,5} = 6,3\%$

ON remarque alors que les valeurs des poids et de nourriture sont assez disperses contrairement a la temperature.

Question 2.1

Nous avons utilise la detection d'outliers avec les ecarts interquartiles precedemment grace aux boxplots. Maintenant, utilisons le Z score pour identifier les outliers.

Question 2.2

Voici la nouvelle boîte à moustaches après avoir standardisé les valeurs:

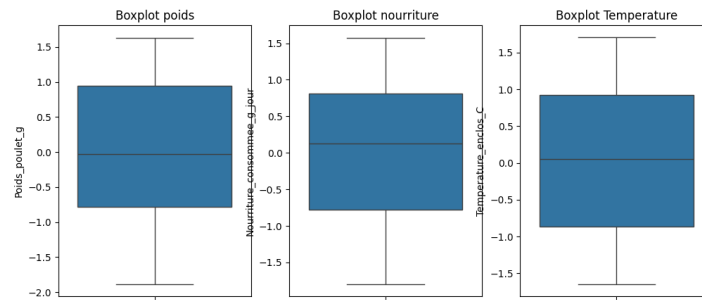


Figure 7: Boîtes à moustaches du poids, de la nourriture, et de la température standardisées

Grace à la standardisation, on ne remarque pas d'outliers. Dans le cas où on en aurait, on aurait dû les enlever car les valeurs aberrantes rajoutent trop de biais aux autres données.

Question 3.1

```
ShapiroResult(statistic=0.9568221670349863, pvalue=9.098264233228524e-06)
ShapiroResult(statistic=0.9448708208372757, pvalue=6.230563751996703e-07)
ShapiroResult(statistic=0.943209717135969, pvalue=4.4060638371198676e-07)
```

Figure 8: Tests Shapiro-Wilk sur le poids, la nourriture et la température

On remarque que les p-values des 3 colonnes donnent des résultats très faibles : 10^{-6} , 10^{-7} et 10^{-7}

Cela montre que les données n'ont pas une répartition normale. (Elle ne ressemble pas à une loi normale)

Question 3.2

```
TtestResult(statistic=array([-1.46652435, -0.46381962, -0.39245999]), pvalue=array([0.15073292, 0.64542304, 0.69691116]), df=array([38., 38., 38.]))
f_onewayresult(statistic=array([2.15069367, 0.21512864, 0.15402484]), pvalue=array([0.15073292, 0.64542304, 0.69691116]))
```

Figure 9: Tests Student et ANOVA sur le poids, la nourriture et la température

On remarque que les p-values du test de student, effectuée sur deux populations de 20 poules aléatoires, montrent que les deux ensembles ne sont pas significativement différents. (p-value bien supérieure à 0.5)

Les p-values sur le test ANOVA sont identiques que ceux du test de student. La conclusion serait alors identique.

Question 4.1

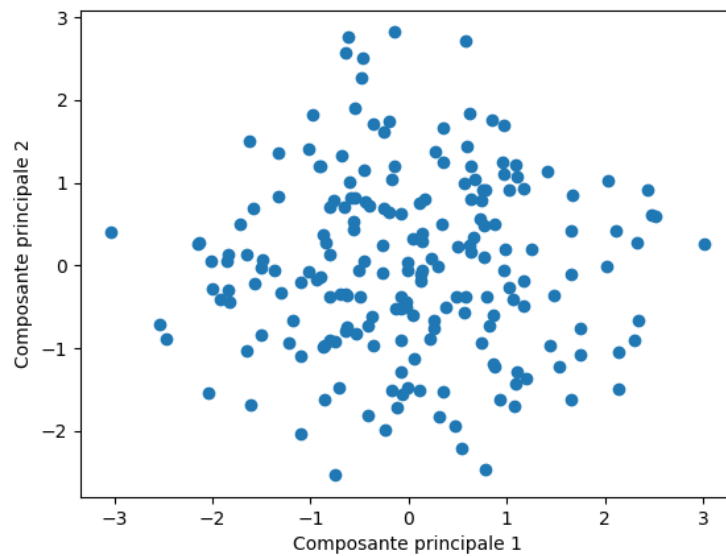


Figure 10: ACP sur le dataframe sans sklearn

Question 4.2

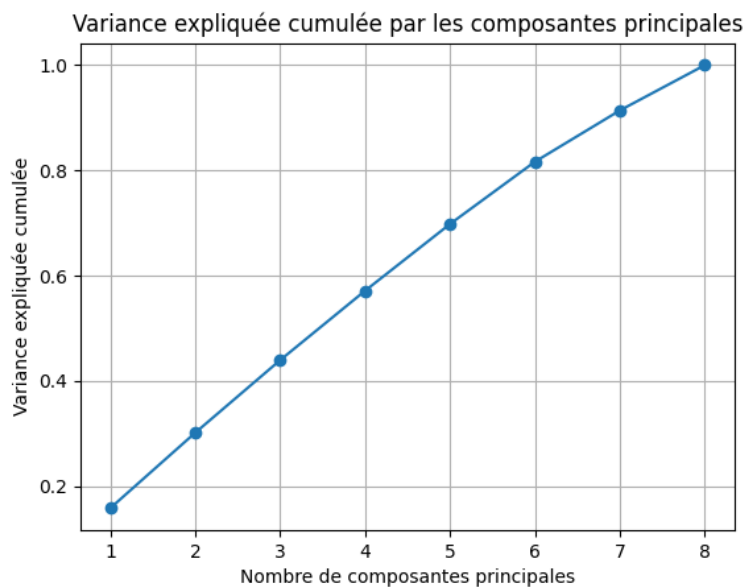


Figure 11: Tests de la variance cumulée en fonction du nombre de composantes

En General, nous decidons de garder environ 80% a 90% de la variance. Ici, on observe que 6 composantes permettent d'obtenir cela.

Question 5.1

KernelPCA avec noyau linéaire : Ce noyau est équivalent à l'ACP classique, car il n'ajoute pas de transformation non linéaire aux données.

KernelPCA avec noyau RBF (Radial Basis Function) : Ce noyau transforme les données dans un espace de plus grande dimension, permettant de capturer des relations non linéaires complexes.

KernelPCA avec noyau polynomial : Ce noyau permet également de projeter les données dans un espace de plus grande dimension avec une transformation polynomiale.

Question 5.2

L'ACP classique (lorsque les données sont linéairement séparables) peut suffire à bien représenter les données dans un espace réduit. Cependant, si les données ne sont pas linéairement séparables (par exemple, dans des situations de classification complexe), KernelPCA avec des noyaux non linéaires (RBF ou polynomial) peut offrir une meilleure séparation des données.

KernelPCA est particulièrement utile dans des cas où les relations entre les données sont complexes et non linéaires, par exemple lorsqu'il existe des structures sous-jacentes difficiles à capturer avec l'ACP classique.

Question 6.1

```
Categories (3, int64): [0 < 1 < 2]
Accuracy: 0.50
F1-score: 0.50
Humidite_%: 0.1646
Poids_poulet_g: 0.1638
Cout_elevage_FCFA: 0.1452
Gain_poids_jour_g: 0.1445
Nourriture_consommee_g_jour: 0.1408
Age_poulet_jours: 0.1244
Temperature_enclos_C: 0.1168
```

Figure 12: Resultats classification

Humidité (%) : L'humidité a une importance de 0.1646, ce qui en fait le facteur le plus influent. L'humidité dans l'enclos peut affecter directement la santé des poulets, notamment en facilitant la propagation de maladies respiratoires.

Poids poulet (g) : Avec une importance de 0.1638, le poids des poulets est également un facteur important. Un poids insuffisant pourrait être lié à une mauvaise alimentation ou à des conditions de vie stressantes, ce qui réduit la survie.

Coût de l'élevage (FCFA) : Ce facteur a une importance de 0.1452. Le coût d'élevage peut affecter la qualité des conditions de vie, notamment en ce qui concerne la qualité de la nourriture, les équipements et les soins de santé. Un coût plus élevé pourrait être associé à des conditions plus favorables pour les poulets.

Gain de poids par jour (g) : L'importance de 0.1445 indique que le gain de poids est un bon indicateur de la santé des poulets. Si les poulets ne prennent pas suffisamment de poids, cela peut indiquer des problèmes alimentaires ou de santé.

Nourriture consommée par jour (g) : À 0.1408, cette variable montre l'impact de l'alimentation quotidienne sur la survie. Une quantité insuffisante de nourriture peut affecter la croissance et la santé des poulets.

Âge du poulet (jours) : Avec une importance de 0.1244, l'âge peut aussi être un facteur de survie. Les jeunes poulets peuvent être plus vulnérables aux maladies et au stress.

Température de l'enclos (°C) : À 0.1168, la température a un impact modéré mais important. Des températures trop extrêmes, trop chaudes ou trop froides, peuvent affecter la santé des poulets.

Exercice 7.1

daBoost :

- Fonctionne bien sur des modèles simples et rapides.

- Moins performant si les relations entre les variables sont complexes.

- Sensible au bruit dans les données.

Gradient Boosting :

- Plus robuste et efficace pour capturer des relations complexes.

- Moins sensible aux valeurs aberrantes que AdaBoost.

- Peut être plus précis si les bons hyperparamètres sont choisis.

Exercice 7.2

AdaBoost fonctionne en attribuant des poids aux erreurs et en se concentrant davantage sur les points mal prédits à chaque itération.

Si un outlier est présent, AdaBoost va lui attribuer un poids élevé et essayer de l'ajuster fortement, ce qui peut déstabiliser le modèle.

Résultat : Risque de surajustement si les outliers sont nombreux ou extrêmes.

Contrairement à AdaBoost, Gradient Boosting réduit progressivement l'erreur résiduelle, ce qui lui permet de mieux gérer les données bruitées.

Les erreurs dues aux outliers sont diluées au fil des itérations, empêchant le modèle de leur accorder trop d'importance.

De plus, certaines variantes comme Huber loss ou quantile regression (dans `sklearn.ensemble.GradientBoostingRegressor`) aident à mieux gérer les valeurs extrêmes.