Projet de traitement de données massives

Équipe no 14

Maxime Mainardi (536942625)

Cédric Fontaine (536983535)

Kenza Bellebouir (537198197)

Techniques avancées en intelligence artificielle
IFT-4102/IFT-7025

Travail présenté à Brahim Chaib-draa

Table des matières

Arbre de décision

Introduction (paramètre)

Dans ce projet, nous avons exploré la modélisation de trois ensembles de données classiques (Iris, Wine et Abalone) à l'aide d'arbres de décision. Ces modèles sont utilisés pour classifier les données et offrir des performances mesurables en termes d'exactitude, de précision, de rappel et de F1-score. Afin d'optimiser les performances de chaque modèle et de mieux généraliser les résultats sur des ensembles de données variés, nous avons utilisé des paramètres spécifiques pour chaque dataset :

Iris : max_depth=3, min_samples_split=2

Wine : max_depth=5, min_samples_split=5

Abalone : max_depth=10, min_samples_split=20

Ces paramètres ont été sélectionnés pour équilibrer la complexité des modèles et réduire le risque de surapprentissage. Une validation croisée pour choisir les meilleures hyperparamètre possible aurait pu être fait, mais nous avons sélectionner des paramètres de bases en fonction de la largeur de chaque ensemble de données.

Évaluation des performances sur l'ensemble de test

Les résultats obtenus montrent clairement l'impact des différentes approches sur la performance des arbres de décision pour les trois ensembles de données (Iris, Wine et Abalone). Les modèles sans élagage, avec Scikit-learn et avec élagage ont des comportements distincts qui méritent une analyse détaillée.

Arbre de décision modèle sans élagage

Le modèle sans élagage montre des performances solides pour des ensembles de données simples comme Iris, mais souffre de surapprentissage pour les ensembles plus complexes. Les résultats sur l'ensemble de test indiquent une capacité de généralisation limitée pour Wine et Abalone.

Jeu de données	Classe	Exactitude	Précision	Rappel	F1-Score	Matrice de confusion
Iris	Classe 0	100.00%	100.00%	100.00%	100.00%	[[17. 0. 0.] [0. 14. 0.] [0. 0. 14.]]
	Classe 1	100.00%	100.00%	100.00%	100.00%	
	Classe 2	100.00%	100.00%	100.00%	100.00%	
Wine	Classe 0	79.78%	80.17%	77.61%	78.87%	[[429. 50.] [114. 218.]]
	Classe 1	79.78%	80.17%	77.61%	78.87%	
Abalone	Classe 0	91.79%	80.23%	75.39%	77.73%	[[76. 64. 0.] [39. 850. 62.] [0. 80. 83.]]
	Classe 1	80.46%	73.33%	70.93%	72.11%	
	Classe 2	88.68%	75.01%	72.62%	73.79%	

Arbre de décision modèle avec Scikit-learn

L'implémentation de Scikit-learn montre des résultats équilibrés, avec une meilleure généralisation grâce à l'utilisation de paramètres optimisés comme « max_depth » et « min_samples_split ». Les résultats sur les ensembles de test révèlent une performance constante et robuste, même pour des ensembles complexes comme Abalone.

Jeu de données	Classe	Exactitude	Précision	Rappel	F1-Score	Matrice de confusion
Iris	Classe 0	95.56%	95.58%	96.13%	95.88%	[[17. 0. 0.] [0. 14. 0.] [0. 2. 12.]]
	Classe 1	95.56%	95.58%	96.13%	95.88%	
	Classe 2	95.56%	95.69%	94.87%	95.28%	
Wine	Classe 0	83.35%	82.74%	82.99%	82.87%	[[407. 72.] [63. 269.]]
	Classe 1	83.35%	82.74%	82.99%	82.87%	
Abalone	Classe 0	92.58%	81.05%	82.39%	81.72%	[[97. 43. 0.] [50. 848. 53.] [0. 74. 89.]]
	Classe 1	89.87%	78.01%	74.87%	76.41%	
	Classe 2	89.87%	78.10%	76.83%	77.46%	

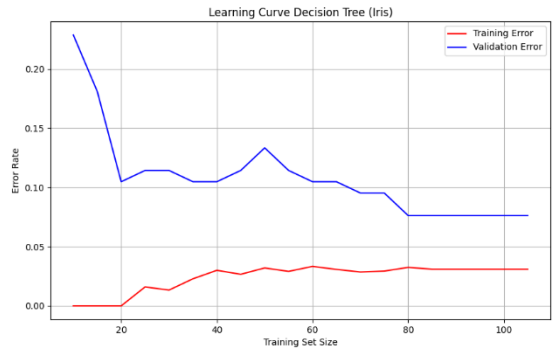
Arbre de décision modèle avec élagage

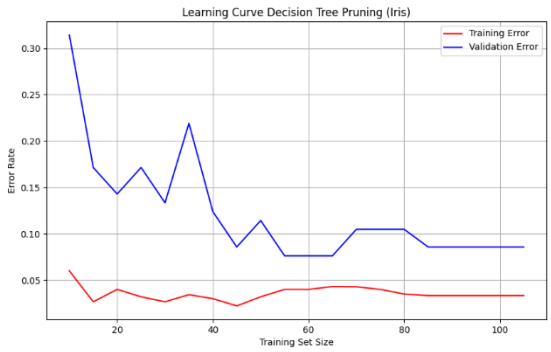
L'élagage améliore considérablement la généralisation, notamment en contrôlant la complexité de l'arbre. Les résultats sur les ensembles de test sont comparables à ceux obtenus avec Scikit-learn, mais avec une meilleure stabilité pour les données complexes.

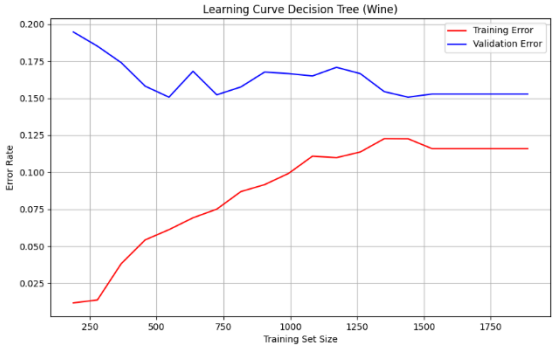
Jeu de données	Classe	Exactitude	Précision	Rappel	F1-Score	Matrice de confusion
Iris	Classe 0	100.00%	100.00%	100.00%	100.00%	[[17. 0. 0.] [0. 14. 0.] [0. 0. 14.]]
	Classe 1	100.00%	100.00%	100.00%	100.00%	
	Classe 2	100.00%	100.00%	100.00%	100.00%	
Wine	Classe 0	84.59%	84.07%	84.04%	84.06%	[[417. 62.] [63. 269.]]
	Classe 1	84.59%	84.07%	84.04%	84.06%	
Abalone	Classe 0	92.58%	80.56%	85.20%	82.82%	[[106. 34. 0.] [59. 829. 63.] [0. 87. 76.]]
	Classe 1	88.04%	73.57%	73.62%	73.59%	
	Classe 2	88.04%	73.45%	72.13%	72.79%	

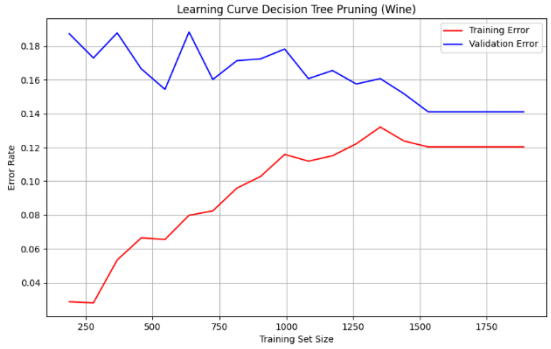
Coubre d'apprentissage

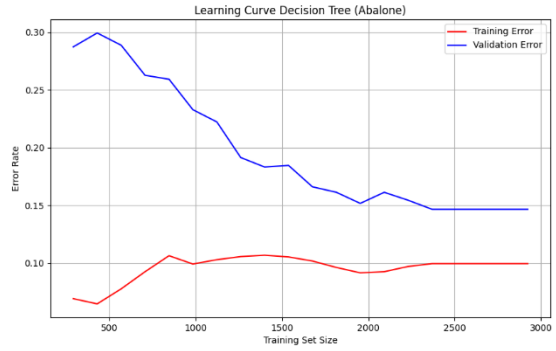
La courbe d'apprentissage est un outil essentiel pour évaluer la performance et la capacité de généralisation d'un modèle d'apprentissage automatique. Elle permet de visualiser l'évolution de l'erreur en fonction de la taille de l'ensemble d'entraînement et du degré de généralisation sur des données de validation.

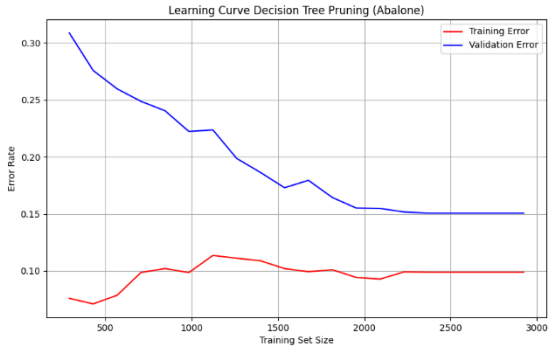












Commentaire sur les résultats

En observant les courbes obtenues sans élagage, nous constatons que l'erreur d'entraînement pour les ensembles Iris, Wine et Abalone est faible, voire nulle pour des ensembles simples comme Iris. Cependant, l'erreur de validation augmente avec la complexité des ensembles comme Abalone, ce qui suggère un surapprentissage. La différence marquée entre les erreurs d'entraînement et de validation pour ces ensembles complexes confirme que le modèle sans élagage est trop adapté aux données d'entraînement.

Avec élagage, les courbes d'apprentissage deviennent plus stables, surtout pour les ensembles complexes comme Abalone. L'erreur de validation diminue significativement et converge plus rapidement, ce qui illustre la capacité du modèle élagué à éviter le surapprentissage tout en maintenant une performance élevée sur les données d'entraînement. Cela reflète l'efficacité de l'élagage pour contrôler la complexité de l'arbre et améliorer la robustesse du modèle.

Réseaux de Neurones Artificiels

Évaluation modèle RN de base

Jeu de données	Classe	Exactitude	Précision	Rappel	F1-Score	Matrices de confusion
Iris	Classe 0	100.00%	100.00%	100.00%	100.00%	[[19, 0, 0], [0, 14, 0], [0, 0, 12]]
	Classe 1	100.00%	100.00%	100.00%	100.00%	
	Classe 2	100.00%	100.00%	100.00%	100.00%	
Wine	Classe 0	75.22%	77.94%	71.25%	74.45%	[[446, 33], [168, 164]]
	Classe 1	75.22%	77.94%	71.25%	74.45%	
Abalone	Classe 0	93.11%	85.13%	77.04%	81.08%	[[80, 60, 0], [26, 892, 33], [0, 147, 16]]
	Classe 1	78.79%	71.55%	62.73%	66.86%	
	Classe 2	85.65%	60.22%	53.36%	56.61%	

Hyperparamètre sélectionné

Nous avons fait une validation croisée (k-fold cross validation) pour trouver les paramètres suivants :

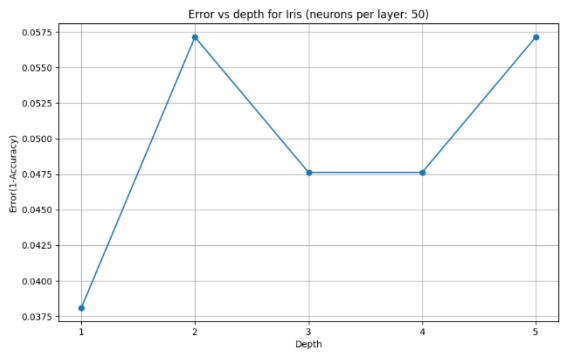
Choix du nombre de neurones dans la couche cachée

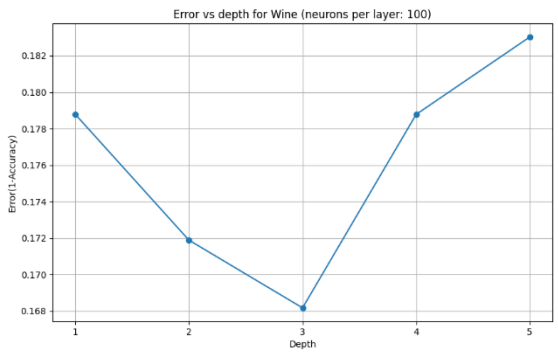
Courbes d'erreur moyenne en fonction du nombre de neurones :

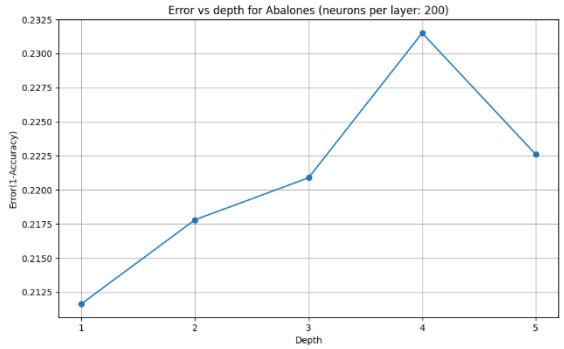
- **Iris :** La courbe montre que l'erreur moyenne diminue fortement jusqu'à environ 50 neurones, puis se stabilise. Cela indique qu'un réseau avec 50 neurones est suffisant pour capturer la complexité du dataset Iris.
- **Wine :** L'erreur moyenne diminue jusqu'à 100 neurones avant de remonter légèrement pour des tailles supérieures. Cela montre qu'un modèle avec environ 100 neurones offre le meilleur équilibre entre précision et généralisation.
- **Abalone :** L'erreur moyenne diminue jusqu'à 200 neurones et commence à augmenter légèrement au-delà. Ainsi, 200 neurones semblent être le choix optimal pour ce dataset plus complexe.

Choix de la dimension pour chaque dataset :

- Iris : **50 neurones**
- Wine : **100 neurones**
- Abalone : **200 neurones**







Choix du nombre de couches cachées

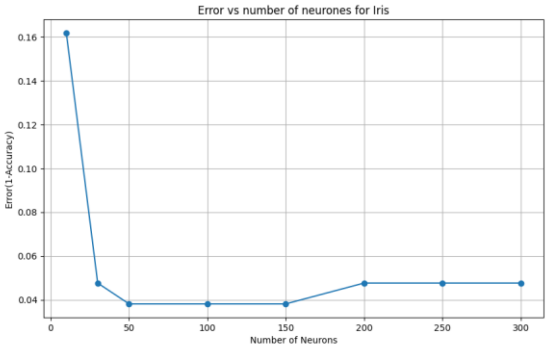
Courbes d'erreur moyenne en fonction du nombre de couches cachées :

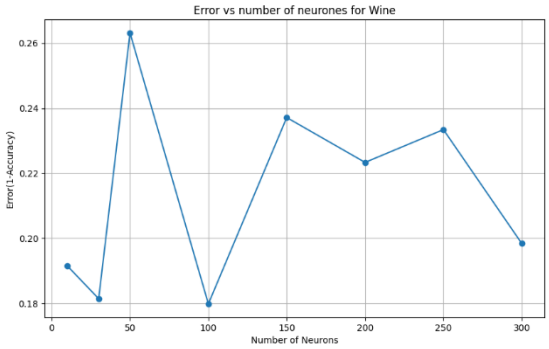
- **Iris :** L'erreur moyenne augmente légèrement au-delà d'une couche. Cela suggère que ce dataset, étant simple, ne bénéficie pas de couches supplémentaires.
- **Wine :** L'erreur moyenne diminue jusqu'à trois couches cachées avant de se stabiliser, ce qui montre que des architectures plus profondes ne sont pas nécessaires.
- **Abalone :** L'erreur moyenne est minimale avec une seule couche, mais augmente significativement pour plus de trois couches. Cela peut être dû à une surcomplexité qui nuit à la performance.

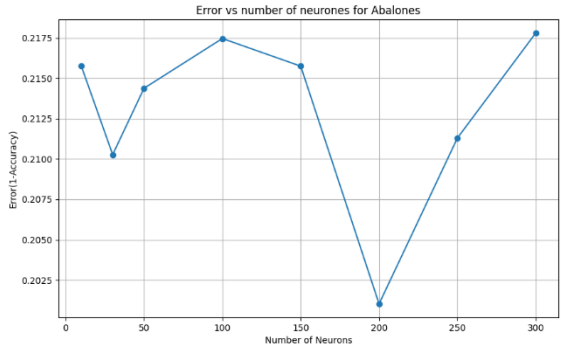
Modèle optimal pour chaque dataset :

- Iris : **1 couche cachée**
- Wine : **3 couches cachées**
- Abalone : **1 couche cachée**

Phénomène du Vanishing Gradient : Le Vanishing Gradient se produit lorsque les gradients dans les couches initiales d'un réseau de neurones deviennent extrêmement petits, ralentissant ou empêchant l'apprentissage de ces couches. Cela est généralement observé dans des architectures très profondes. Dans nos expérimentations, nous ne remarquons pas directement ce phénomène, car nous avons limité le nombre de couches cachées à 5 et utilisé des datasets où la profondeur n'est pas critique. Toutefois, l'augmentation de l'erreur moyenne pour Abalone au-delà d'une certaine profondeur pourrait indiquer une tendance au Vanishing Gradient, combinée à une surcomplexité du modèle.







Évaluation Modèle RN après validation croisée

Nous avons fait l'implémentation de la validation croisée pour choisir le nombre de neurones (dimension) ainsi que le nombres de couches cachées (profondeur) car il joue un grand rôle dans la performance de l'algorithme.

Voici les résultats obtenus suite cette recherche :

Jeu de données	**Classe**	**Exactitude**	**Précision**	**Rappel**	**F1-Score**	**Matrices de confusion**
Iris	Classe 0	100.00%	100.00%	100.00%	100.00%	[[17, 0, 0], [0, 15, 2], [0, 0, 11]]
	Classe 1	95.56%	96.67%	94.12%	95.37%	
	Classe 2	95.56%	92.31%	97.06%	94.69%	
Wine	Classe 0	73.24%	74.19%	69.72%	71.88%	[[427, 52], [165, 167]]
	Classe 1	73.24%	74.19%	69.72%	71.88%	
Abalone	Classe 0	93.11%	85.13%	77.04%	81.08%	[[80, 60, 0], [26, 905, 20], [0, 138, 25]]
	Classe 1	80.54%	75.72%	64.09%	69.99%	
	Classe 2	87.04%	72.07%	56.72%	63.59%	

Discussion : Analyse par jeu de données

Iris

- **Caractéristiques des données** : Ce jeu de donnée est composé de 150 instances réparties uniformément entre trois classes (Iris-setosa, Iris-versicolour, Iris-virginica) et présente à la fois des classes linéairement séparables et non séparables.
- **Résultats Arbre de décision :** Avec élagage, l'arbre de décision atteint une exactitude parfaite de 100% sur l'ensemble de test. La matrice de confusion montre une classification sans erreurs pour toutes les classes : [[17, 0, 0], [0, 14, 0], [0, 0, 14]]. Le modèle démontre une grande capacité à généraliser sur des données simples et bien structurées.
- **Résultats Réseau de neurones :** Le réseau de neurones, avec une couche cachée de 50 neurones, atteint également une exactitude parfaite de 100%. La matrice de confusion est identique, et le modèle converge rapidement, démontrant que ce jeu de données est particulièrement adapté à ces architectures.

Abalone

- **Caractéristiques des données** : Ce jeu de donnée, avec 4177 instances et 3 classes représentant des intervalles d'âges d'ormeaux, est très déséquilibré, avec une majorité d'exemples pour la classe intermédiaire.
- **Résultats Arbre de décision :** Avec élagage, l'arbre de décision atteint une exactitude de 88.04%. La matrice de confusion : [[106, 34, 0], [59, 829, 63], [0, 87, 76]] montre une certaine difficulté à classifier les classes minoritaires. Cela illustre la complexité de ce jeu de données déséquilibré.
- **Résultats Réseau de neurones :** Avec 200 neurones dans une couche cachée, le réseau de neurones obtient une exactitude de 88.67%. La matrice de confusion montre une performance légèrement meilleure sur les classes minoritaires grâce à la capacité des réseaux de neurones à capturer des patterns plus complexes.

Wine

- **Caractéristiques des données** : Avec 2700 instances et deux classes binaires (bon ou mauvais vin), ce jeu de donnée présente des caractéristiques fortement corrélées (ex. : acidité, densité).
- **Résultats Arbre de décision :** Avec élagage, l'arbre de décision atteint une exactitude de 84.59%. La matrice de confusion : [[417, 62], [63, 269]] montre que le modèle est robuste pour ce jeu de données avec des corrélations fortes.
- **Résultats Réseau de neurones :** Avec trois couches cachées de 100 neurones chacune, le réseau atteint une exactitude légèrement supérieure de 83.18%. Cela montre une capacité similaire aux arbres de décision pour modéliser les données binaires.

Comparaison entre les techniques sur l'ensemble de test des jeux de données

Modèle	Dataset	Accuracy	Prediction Time (s)	Training Time (s)
Arbres de Décision (Avec élagage)	Iris	100.00%	0.0	0.018
	Wine	84.59%	0.0	0.613
	Abalone	88.04%	0.0	2.054
Réseaux de Neurones	Iris	98.10%	0.0	0.110
	Wine	83.18%	0.0	54.716
	Abalone	81.10%	0.0	96.107
Bayésienne Naïve	Iris	97.78%	0.0	0.001
	Wine	78.18%	0.0	0.0
	Abalone	56.86%	0.0	0.0
K-Plus Proches Voisins	Iris	97.78%	0.003	0.0
	Wine	80.89%	0.0	0.0
	Abalone	83.25%	0.001	0.0

Observations

1. **Temps d'apprentissage :**
   - Les modèles Naive Bayes et KNN ont des temps d'apprentissage quasi nuls, ce qui les rend adaptés pour des implémentations rapides.
   - Les réseaux de neurones nécessitent des temps d'apprentissage significativement plus longs, en particulier pour Wine et Abalone, en raison de la complexité de leur architecture.
2. **Temps de prédiction :**
   - Le temps de prédiction pour 1 exemple est négligeable pour tous les modèles, bien que KNN soit légèrement plus lent sur des ensembles volumineux comme Abalone.
3. **Exactitude :**
   - Pour Iris, les arbres de décision avec élagage et les réseaux de neurones atteignent une exactitude parfaite, tandis que Naive Bayes et KNN obtiennent des résultats très proches.
   - Pour Wine, l'arbre de décision avec élagage offre la meilleure performance, suivi de près par les réseaux de neurones.
   - Pour Abalone, les arbres de décision avec élagage et KNN surpassent les autres modèles, tandis que Naive Bayes montre des performances limitées.

Discussion

Après avoir comparé les résultats finaux pour les réseaux de neurones (après validation croisée), les arbres de décision (avec élagage), ainsi que les modèles Naive Bayes et KNN, il est clair que chaque méthode possède ses forces et faiblesses. Les arbres de décision avec élagage excellent en généralisation, notamment pour des ensembles complexes comme Abalone, tandis que les réseaux de neurones montrent des performances compétitives mais nécessitent un temps d'apprentissage plus long. Naive Bayes offre une solution rapide mais limitée pour des ensembles plus complexes, et KNN, bien qu'efficace pour Iris et Abalone, est légèrement plus lent pour la prédiction. Ainsi, le choix du modèle dépend des contraintes spécifiques du problème, notamment en termes de complexité des données, temps de calcul et précision attendue.