École Nationale Supérieure d'Arts et Métiers

IATD-SI

RETIS

Arbres de Régression avec Modèles Linéaires dans les Feuilles

Rapport de Projet

KDD

Nkira Mohamed Reda - Boudrika Ilias - Es-saaidi Youssef

Module :
KDD
S7 - 4ème Année

Année Universitaire :
2025 - 2026

Table des matières

1 Introduction

1.1 Contexte et Motivation

L'apprentissage automatique pour la régression représente un domaine fondamental de l'intelligence artificielle, avec des applications allant de la prédiction financière à la modélisation de systèmes physiques complexes. Parmi les nombreuses approches existantes, les **arbres de décision** se distinguent par leur interprétabilité et leur capacité à capturer des relations non-linéaires.

Cependant, les arbres de régression classiques (CART) présentent une limitation majeure : ils effectuent des prédictions constantes dans chaque feuille, correspondant à la moyenne des observations. Cette approche, bien que robuste, ne capture pas les tendances linéaires locales qui peuvent exister au sein des partitions de l'espace des caractéristiques.

1.2 Objectifs du Projet

Ce projet vise à implémenter et évaluer l'algorithme **RETIS** (Regression Tree Induction System), proposé par Karalič en 1992. Les objectifs principaux sont :

1. Comprendre les fondements théoriques de RETIS et ses différences avec les arbres classiques
2. Implémenter l'algorithme de manière fidèle à la publication originale
3. Développer un système d'élagage basé sur des tests statistiques (F-test)
4. Intégrer une régularisation Ridge pour réduire le surapprentissage
5. Évaluer les performances sur des problèmes de régression et de classification

1.3 Organisation du Rapport

Ce rapport est structuré comme suit : la Section 2 présente les fondements théoriques de RETIS. La Section 3 détaille l'approche méthodologique et les choix d'implémentation. La Section 4 présente les résultats expérimentaux. Enfin, la Section 5 conclut ce travail et propose des perspectives.

2 Fondements Théoriques

2.1 Arbres de Régression Classiques (CART)

2.1.1 Principe Général

L'algorithme CART (Classification and Regression Trees), introduit par Breiman et al. (1984), construit un arbre binaire en partitionnant récursivement l'espace des caractéristiques. Pour la régression, chaque feuille prédit une valeur constante :

$$\hat{y}_{\text{feuille}} = \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{1}$$

où n est le nombre d'échantillons dans la feuille et y_i les valeurs cibles correspondantes.

2.1.2 Critère de Division

La division optimale minimise la somme des erreurs quadratiques (SSE) dans les nœuds enfants :

$$\text{SSE}_{\text{total}} = \sum_{i \in \text{gauche}} (y_i - \bar{y}_{\text{gauche}})^2 + \sum_{i \in \text{droite}} (y_i - \bar{y}_{\text{droite}})^2 \tag{2}$$

2.1.3 Limitations

Les arbres CART souffrent de plusieurs limitations :

- **Discontinuités :** Les prédictions sont en escalier, créant des discontinuités aux frontières
- **Approximation grossière :** Les relations linéaires locales ne sont pas capturées
- **Profondeur excessive :** Nécessité d'arbres profonds pour approximer des fonctions lisses

2.2 RETIS : Modèles Linéaires dans les Feuilles

Définition : RETIS

RETIS (Regression Tree Induction System) est un algorithme d'arbre de régression qui ajuste un **modèle de régression linéaire** dans chaque feuille de l'arbre, plutôt qu'une simple moyenne.

2.2.1 Formulation Mathématique

Dans chaque feuille ℓ, RETIS ajuste un modèle linéaire :

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \mathbf{x}^T \boldsymbol{\beta} \tag{3}$$

où $\mathbf{x} = [1, x_1, \ldots, x_p]^T$ est le vecteur augmenté des caractéristiques et $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]^T$ le vecteur des coefficients.

Les coefficients sont estimés par la méthode des moindres carrés :

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \tag{4}$$

2.2.2 Critère de Division Modifié

La division optimale minimise désormais la somme des erreurs résiduelles des modèles linéaires :

$$\text{SSE}_{\text{split}} = \text{SSE}^{\text{lin}}_{\text{gauche}} + \text{SSE}^{\text{lin}}_{\text{droite}} \tag{5}$$

où $\text{SSE}^{\text{lin}} = \sum_i (y_i - \hat{y}_i^{\text{lin}})^2$ représente l'erreur du modèle linéaire ajusté.

2.3 Élagage Statistique par F-test

2.3.1 Motivation

La croissance non contrôlée de l'arbre conduit au surapprentissage. Karalič (1992) propose un élagage basé sur le **test F de Fisher**, qui évalue si la réduction d'erreur apportée par une division est statistiquement significative.

2.3.2 Formulation du Test F

Soit un nœud parent avec erreur SSE_p et df_p degrés de liberté, et ses enfants avec erreur combinée SSE_c et df_c degrés de liberté. La statistique F est :

$$F = \frac{(\text{SSE}_p - \text{SSE}_c)/(\text{df}_p - \text{df}_c)}{\text{SSE}_c/\text{df}_c} \tag{6}$$

2.3.3 Règle de Décision

La division est conservée si $F > F_\alpha(\text{df}_p - \text{df}_c, \text{df}_c)$, où F_α est la valeur critique au niveau de signification α (typiquement 0.01 ou 0.05).

Degrés de Liberté

Pour un modèle linéaire avec p caractéristiques et n observations :

$$\text{df} = n - (p + 1) \tag{7}$$

Le terme $(p + 1)$ représente l'intercept plus les p coefficients.

2.4 Régularisation Ridge (L2)

2.4.1 Problème du Surapprentissage

Les modèles linéaires dans les feuilles peuvent surapprendre, surtout lorsque le nombre d'échantillons est faible par rapport au nombre de caractéristiques. La régularisation Ridge ajoute une pénalité sur la norme des coefficients.

2.4.2 Formulation

Au lieu de minimiser uniquement l'erreur quadratique, on minimise :

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{8}$$

où $\lambda > 0$ est le paramètre de régularisation.

2.4.3 Solution Analytique

Les coefficients régularisés sont donnés par :

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^T \mathbf{y} \tag{9}$$

où $\mathbf{I}_p$ est la matrice identité (sans régulariser l'intercept).

3 Approche Méthodologique

3.1 Architecture de l'Implémentation

L'implémentation suit une architecture modulaire comprenant :

TABLE 1 – Composants principaux de l'implémentation

Composant	Description
`RETIS`	Classe principale implémentant l'arbre de régression
`RETISNode`	Structure de données pour les nœuds de l'arbre
`LinearRegressionCustom`	Régression linéaire avec support Ridge
`RETISOptimizer`	Recherche d'hyperparamètres par validation croisée
`CustomMetrics`	Suite complète de métriques d'évaluation

3.2 Algorithme de Construction

La construction de l'arbre RETIS suit un processus en deux phases :

3.2.1 Phase 1 : Croissance de l'Arbre

Algorithm 1 Croissance de l'arbre RETIS

procedure GROWTREE(X, y, depth)
    Ajuster un modèle linéaire sur (X, y)
    Calculer l'erreur résiduelle $\text{SSE}_{\text{parent}}$
    if critères d'arrêt satisfaits **then**
        return nœud feuille avec modèle linéaire
    end if
    Trouver la meilleure division (j^*, t^*)
    Partitionner les données en gauche/droite
    return nœud interne avec enfants récursifs
end procedure

3.2.2 Phase 2 : Élagage Post-hoc

Algorithm 2 Élagage par F-test

```
procedure PruneTree(nœud, X, y)
    if nœud est une feuille then
        return
    end if
    Élaguer récursivement les enfants
    Calculer la statistique F
    if F < F_critique then
        Convertir le nœud en feuille
    end if
end procedure
```

3.3 Hyperparamètres

Les hyperparamètres clés de RETIS sont présentés dans le Tableau 2.

Table 2 – Hyperparamètres de RETIS et valeurs par défaut

Paramètre	Défaut	Description
`max_depth`	8	Profondeur maximale de l'arbre
`min_samples_split`	20	Échantillons minimum pour diviser
`min_samples_leaf`	10	Échantillons minimum par feuille
`significance_level`	0.01	Niveau de signification pour le F-test
`min_error_reduction`	0.01	Réduction d'erreur relative minimale
`m_estimate`	2.0	Paramètre de régularisation Ridge (λ)
`account_for_split_cost`	True	Ajuster les ddl pour la recherche de seuil

3.4 Stratégies Anti-Surapprentissage

Plusieurs mécanismes sont implémentés pour contrôler le surapprentissage :

1. **Régularisation Ridge :** Pénalise les coefficients de grande magnitude dans les modèles linéaires locaux
2. **Élagage statistique :** Utilise le F-test pour ne conserver que les divisions statistiquement significatives
3. **Contraintes de taille :** Impose des tailles minimales de nœuds et de feuilles

4. **Réduction d'erreur minimale :** Exige une amélioration relative minimale pour accepter une division
5. **Limitation des seuils :** Échantillonne un nombre limité de seuils candidats pour éviter le surapprentissage aux données de bruit

3.5 Extension à la Classification

RETIS est naturellement un algorithme de régression. Pour l'adapter à la classification, nous utilisons une approche **One-vs-Rest** (OvR) :

— Pour un problème à K classes, entraîner K modèles RETIS
— Chaque modèle prédit la probabilité d'appartenance à une classe
— La prédiction finale est la classe avec la probabilité maximale

4 Résultats Expérimentaux

4.1 Protocole Expérimental

4.1.1 Données Utilisées

Les expériences ont été menées sur des données synthétiques générées avec des caractéristiques contrôlées :

— **Régression :** 1000 échantillons, 10 caractéristiques, relation linéaire avec bruit gaussien
— **Classification :** 1000 échantillons, 10 caractéristiques, 3 classes

4.1.2 Métriques d'Évaluation

Régression :

— MSE (Mean Squared Error) : $\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2$
— RMSE (Root MSE) : $\sqrt{\text{MSE}}$
— MAE (Mean Absolute Error) : $\frac{1}{n}\sum_i |y_i - \hat{y}_i|$
— R^2 (Coefficient de détermination) : $1 - \frac{\text{SSE}}{\text{SST}}$

Classification :

— Accuracy : Proportion de prédictions correctes
— Précision, Rappel, F1-score (macro et micro)
— AUC-ROC (Area Under ROC Curve)

4.1.3 Validation Croisée

Une validation croisée à 5 plis avec mélange aléatoire est utilisée pour :

- Sélectionner les meilleurs hyperparamètres
- Estimer la performance de généralisation
- Évaluer la stabilité des modèles

4.2 Résultats en Régression

4.2.1 Performances du Modèle

Résultats - Régression

Métrique	Entraînement	Test
MSE	2.53	5.10
RMSE	1.59	2.26
MAE	1.24	1.77
R^2	0.885	0.745
Variance Expliquée	0.885	0.748

4.2.2 Analyse du Surapprentissage

Un indicateur clé de la qualité du modèle est l'écart entre les performances d'entraînement et de test :

$$\Delta R^2 = R^2_{\text{train}} - R^2_{\text{test}} = 0.885 - 0.745 = 0.140 \tag{10}$$

Conclusion : Généralisation

Un écart ΔR^2 de 0.14 indique un léger surapprentissage, typique des modèles avec des données bruitées. Le modèle reste performant avec un R^2 test de **0.745**.

4.2.3 Comparaison avec les Baselines

TABLE 3 – Comparaison avec les prédicteurs naïfs

Modèle	MSE	RMSE	MAE	R^2	MAPE
RETIS	5.10	2.26	1.77	**0.745**	133.42
Prédicteur Moyenne	20.32	4.51	3.77	-0.016	274.87
Prédicteur Médiane	20.64	4.54	3.79	-0.032	323.00

RETIS surpasse significativement les baselines naïves avec une réduction de MSE de **75%** par rapport au prédicteur moyenne.

4.3 Résultats en Classification

4.3.1 Performances du Modèle

Résultats - Classification

Métrique	Entraînement	Test
Accuracy	0.881	0.758
Précision (macro)	0.882	0.749
Rappel (macro)	0.869	0.738
F1-score (macro)	0.876	0.744
AUC (macro)	0.970	0.869

4.3.2 Analyse du Surapprentissage

$$\Delta\text{Accuracy} = \text{Acc}_{\text{train}} - \text{Acc}_{\text{test}} = 0.881 - 0.758 = 0.123 \tag{11}$$

Conclusion : Généralisation

Un écart d'accuracy de **12%** est typique pour des données bruitées. Le modèle atteint une accuracy test de **75.8%**, significativement supérieure aux baselines.

4.3.3 Comparaison avec les Baselines

TABLE 4 – Comparaison avec les classifieurs naïfs

Modèle	Accuracy	Précision	Rappel	F1	AUC
RETIS	**0.758**	**0.749**	**0.738**	**0.744**	**0.869**
Aléatoire	0.325	0.328	0.334	0.331	0.000
Classe Majoritaire	0.642	0.214	0.333	0.261	0.000

4.4 Validation Croisée

4.4.1 Régression

TABLE 5 – Scores de validation croisée (5 plis) - Régression

Métrique	**Valeur**
MSE moyen	4.12
Écart-type MSE	± 0.33

4.4.2 Classification

TABLE 6 – Scores de validation croisée (5 plis) - Classification

Métrique	**Valeur**
Accuracy moyenne	0.725
Écart-type Accuracy	± 0.025

La faible variance entre les plis témoigne de la stabilité du modèle.

4.5 Structure de l'Arbre

Les arbres construits présentent les caractéristiques suivantes :

TABLE 7 – Statistiques structurelles des arbres

Caractéristique	**Régression**	**Classification**
Profondeur	3-4	4-5
Nombre de feuilles	4-10	6-12
Échantillons/feuille (moy.)	∼60	∼50

Les arbres restent compacts grâce à l'élagage statistique, ce qui favorise l'interprétabilité.

4.6 Effet de la Régularisation

L'impact du paramètre de régularisation Ridge (λ ou `m_estimate`) est illustré ci-dessous :

TABLE 8 – Impact de la régularisation sur le surapprentissage

λ	R^2_{train}	R^2_{test}	ΔR^2
0.0	0.786	-161.86	162.65 (surapprentissage)
1.0	0.512	0.089	0.42
2.0	0.388	0.121	0.27
5.0	0.201	0.145	0.06

Une régularisation plus forte réduit drastiquement le surapprentissage au prix d'une légère sous-adaptation.

5 Discussion

5.1 Avantages de RETIS

1. **Interprétabilité :** La structure arborescente permet de comprendre les règles de décision, tandis que les modèles linéaires locaux révèlent les relations au sein de chaque partition.
2. **Flexibilité :** Capture à la fois les non-linéarités globales (via les divisions) et les tendances linéaires locales (via les modèles des feuilles).
3. **Robustesse :** L'élagage statistique et la régularisation Ridge contrôlent efficacement le surapprentissage.
4. **Pas de dépendances :** L'implémentation repose uniquement sur NumPy, sans nécessiter de bibliothèques externes d'apprentissage automatique.

5.2 Limitations

1. **Complexité computationnelle :** L'ajustement d'un modèle linéaire à chaque division candidate augmente le temps de calcul.
2. **Sensibilité aux hyperparamètres :** Le choix du niveau de signification et du paramètre de régularisation influence significativement les performances.
3. **Données de faible dimension :** RETIS est particulièrement adapté aux problèmes avec un nombre modéré de caractéristiques.

5.3 Comparaison avec M5

RETIS est conceptuellement proche de l'algorithme M5 (Quinlan, 1992), mais présente des différences notables :

TABLE 9 – Comparaison RETIS vs M5

Aspect	RETIS	M5
Modèles	À tous les nœuds	Uniquement aux feuilles
Élagage	F-test statistique	Erreur avec pénalité
Régularisation	Ridge intégrée	Post-traitement

6 Conclusion et Perspectives

6.1 Synthèse

Ce projet a permis d'implémenter avec succès l'algorithme RETIS tel que décrit par Karalič (1992), enrichi de plusieurs améliorations :

— Régularisation Ridge intégrée aux modèles linéaires locaux
— Élagage post-hoc basé sur le F-test avec option de prise en compte du coût de recherche de seuil
— Extension à la classification multiclasse par approche One-vs-Rest
— Suite complète d'évaluation avec validation croisée et comparaison aux baselines

Les résultats expérimentaux démontrent que RETIS :
— Surpasse les prédicteurs naïfs sur les tâches de régression et classification
— Généralise correctement grâce aux mécanismes anti-surapprentissage
— Produit des arbres compacts et interprétables

6.2 Perspectives

Plusieurs extensions pourraient enrichir ce travail :

1. **Sélection de variables :** Implémenter une sélection pas-à-pas des caractéristiques dans les modèles linéaires locaux
2. **Méthodes d'ensemble :** Combiner plusieurs arbres RETIS (forêts aléatoires, boosting)
3. **Régularisation adaptative :** Ajuster automatiquement λ en fonction de la taille du nœud
4. **Parallélisation :** Accélérer la recherche de divisions par calcul parallèle
5. **Données réelles :** Valider les performances sur des jeux de données de benchmark (UCI, Kaggle)

Références

[1] Karalič, A. (1992). *Employing Linear Regression in Regression Tree Leaves*. Proceedings of ECAI-92, Vienna, Austria.

[2] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. CRC Press.

[3] Quinlan, J. R. (1992). *Learning with Continuous Classes*. Proceedings of the 5th Australian Joint Conference on Artificial Intelligence.

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

[5] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

A Annexe : Formules Mathématiques Détaillées

A.1 Régression Linéaire par Moindres Carrés

Soit $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ la matrice de design augmentée (avec colonne de 1) et $\mathbf{y} \in \mathbb{R}^n$ le vecteur cible. La solution des moindres carrés est :

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{12}$$

A.2 Régression Ridge

La solution régularisée avec paramètre λ :

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}_{-0}\|_2^2 \right\} \tag{13}$$

où $\boldsymbol{\beta}_{-0}$ exclut l'intercept de la pénalisation.

A.3 Test F pour la Comparaison de Modèles

Pour comparer un modèle réduit (parent) et un modèle complet (enfants) :

$$F = \frac{(\text{SSE}_{\text{réduit}} - \text{SSE}_{\text{complet}})/(k_{\text{complet}} - k_{\text{réduit}})}{\text{SSE}_{\text{complet}}/(n - k_{\text{complet}})} \tag{14}$$

où k représente le nombre de paramètres du modèle.

Sous H_0 (le modèle réduit est suffisant), F suit une loi de Fisher $\mathcal{F}(k_{\text{complet}} - k_{\text{réduit}}, n - k_{\text{complet}})$.

A.4 Coefficient de Détermination

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{15}$$

où SSE = somme des carrés des résidus et SST = somme totale des carrés.