Introduction à la Science des données
Arbre de décision

V. Bertret

[1] Université de Rennes 1/UFR Mathématiques

Outline

Contexte/Rappel sur l'apprentissage supervisé

Apprentissage supervisé : Apprendre à partir de données **labélisées/étiquetées**.

Jeu de données

- Données d'entrées (X), les régresseurs/features
- Données de sorties (Y), les réponses
- Un ensemble de paires (x, y), representant chacune un exemple

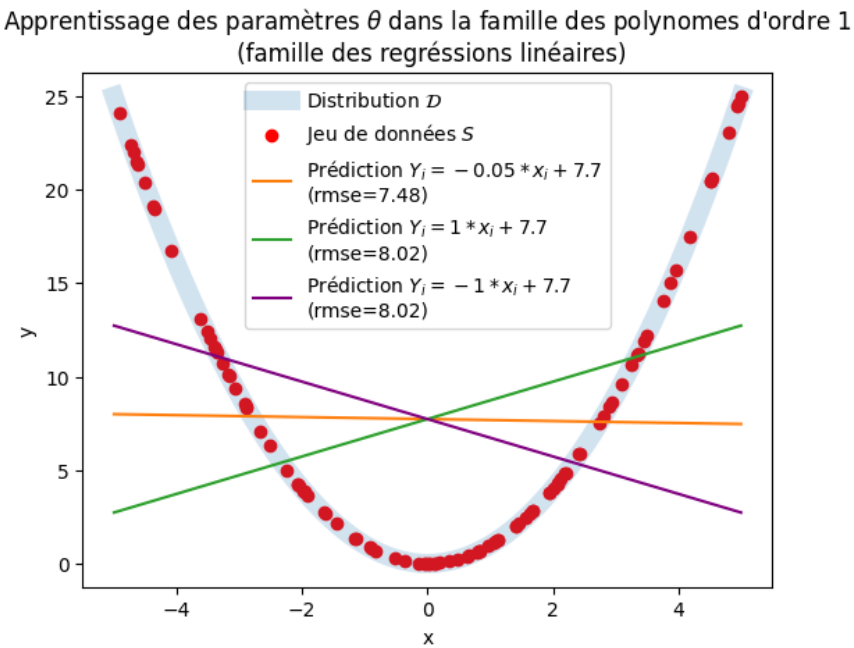
Objectif : Expliquer Y à partir de X (trouver les meilleurs paramètres θ tel que $Y = h_\theta(X)$ avec h_θ une certaine famille de fonction)

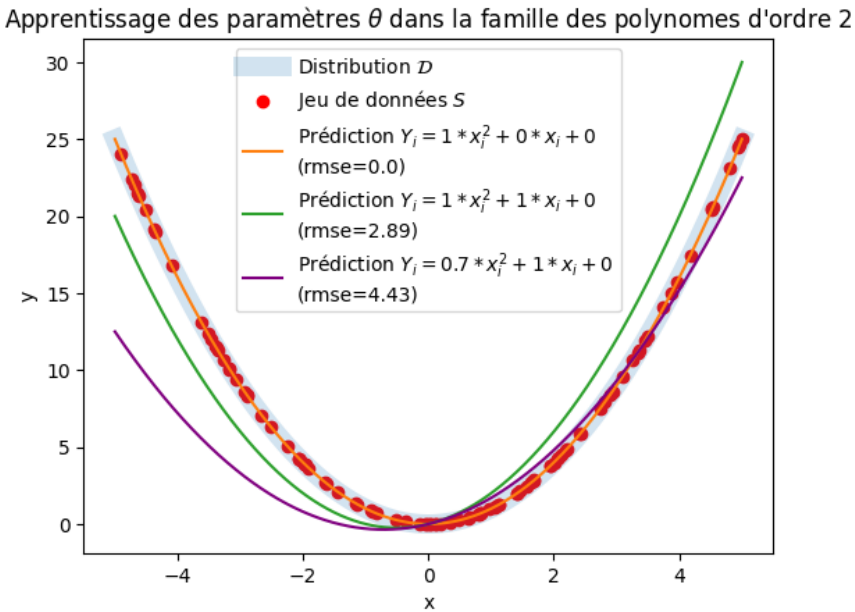
Exemples :

- Données
  - Y : spam/non spam, X : type d'adresse d'expéditeur, nombre d'adresses, fréquence des mots du titre, fréquence des mots du message, nombre de mots dans le message, etc.
  - Y : connection normale/connection attaque , X : temps de connection, heure de la connection, historique de cliques, etc.
- Modèle
  - Modèle linéaire : $f(x) = ax + b$. Deux paramètres à estimer (régression linéaire)
  - Modèle non-linéaire : $f(x) = ax + b\sin(cx) + d$. Quatre paramètres à estimer
  - K plus proches voisins.
  - ...

Compléments Apprentissage et Validation

- **Apprentissage** des meilleurs paramètres θ dans **une famille de classe donnée**.
- **Choix** de la **meilleur famille** en comparant les résultats des **meilleurs fonctions** de chaque famille (**choix du bon compromis biais-variance**).





- Une **famille** est représentée ici par un **ordre de polynôme** spécifique (ordre 1 ou 2).

Compléments métrique classification

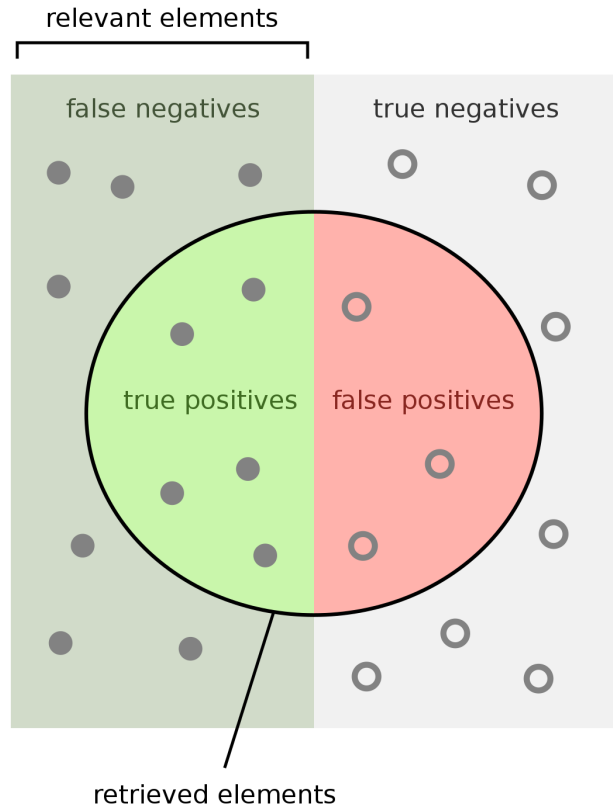
Métrique classique : **Exactitude** (accuracy)

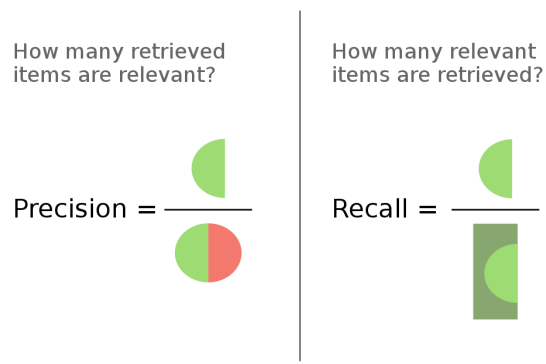
$$\mathcal{L}_S(h) = \sum_{i=1}^{n} \mathbb{1}_{\{y_i = h(x_i)\}}$$

Problème : ensemble de données non équilibré (plus d'importance à la classe la plus représentée)

Solution :

- *Précision* : nombre d'éléments pertinents sélectionnés sur l'ensemble des éléments sélectionnés
- *Rappel* : le nombre d'éléments pertinents retrouvés au regard du nombre d'éléments pertinents que possède l'ensemble de données.

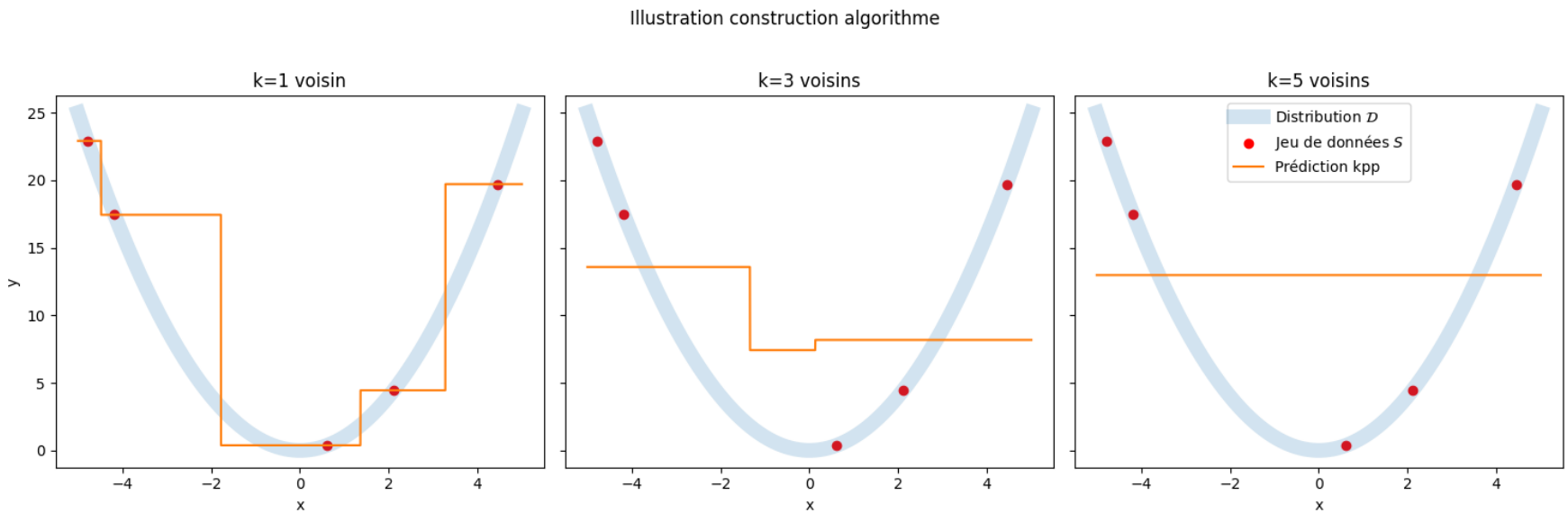




Compléments k plus proches voisins

Algorithme particulier :

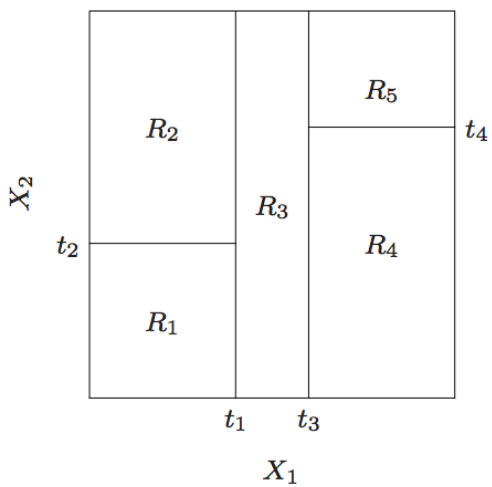
- **Pas** de phase d'**apprentissage**.
- **Pas** d'ensemble de paramètres θ à apprendre.
- Plus il y a de données, meilleurs sont les résultats.

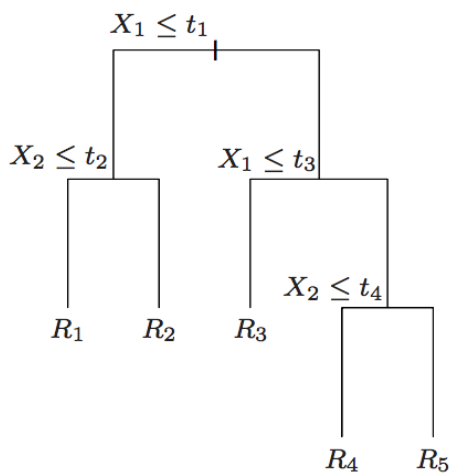


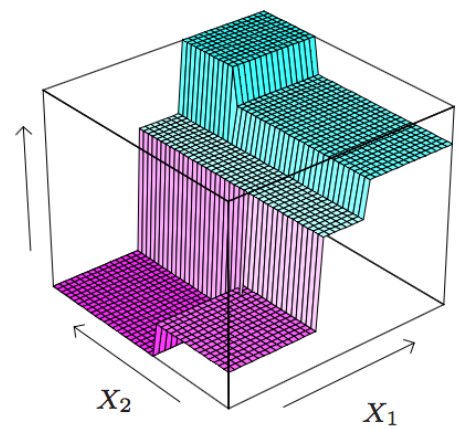
Principe des arbres de décision

- Le principe des algorithmes d'arbre de décision est de **partitionner l'espace en rectangles** dans lesquels la variable à prédire est **homogène** (faible variance) et d'ajuster un modèle **(très) simple** dans chaque région.
- Le concept est simple mais souvent efficace.
- Par exemple, considérons un problème de régression pour prédire une réponse continue $Y \in \mathcal{Y}$ à partir de 2 entrées X_1 et X_2 prenant chacune ses valeurs dans l'intervalle unité. Dans chaque cellule R_m, Y est prédit par la valeur moyenne c_m des valeurs observées dans la cellule.

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \mathbb{I}_{R_m}(x_1, x_2)$$







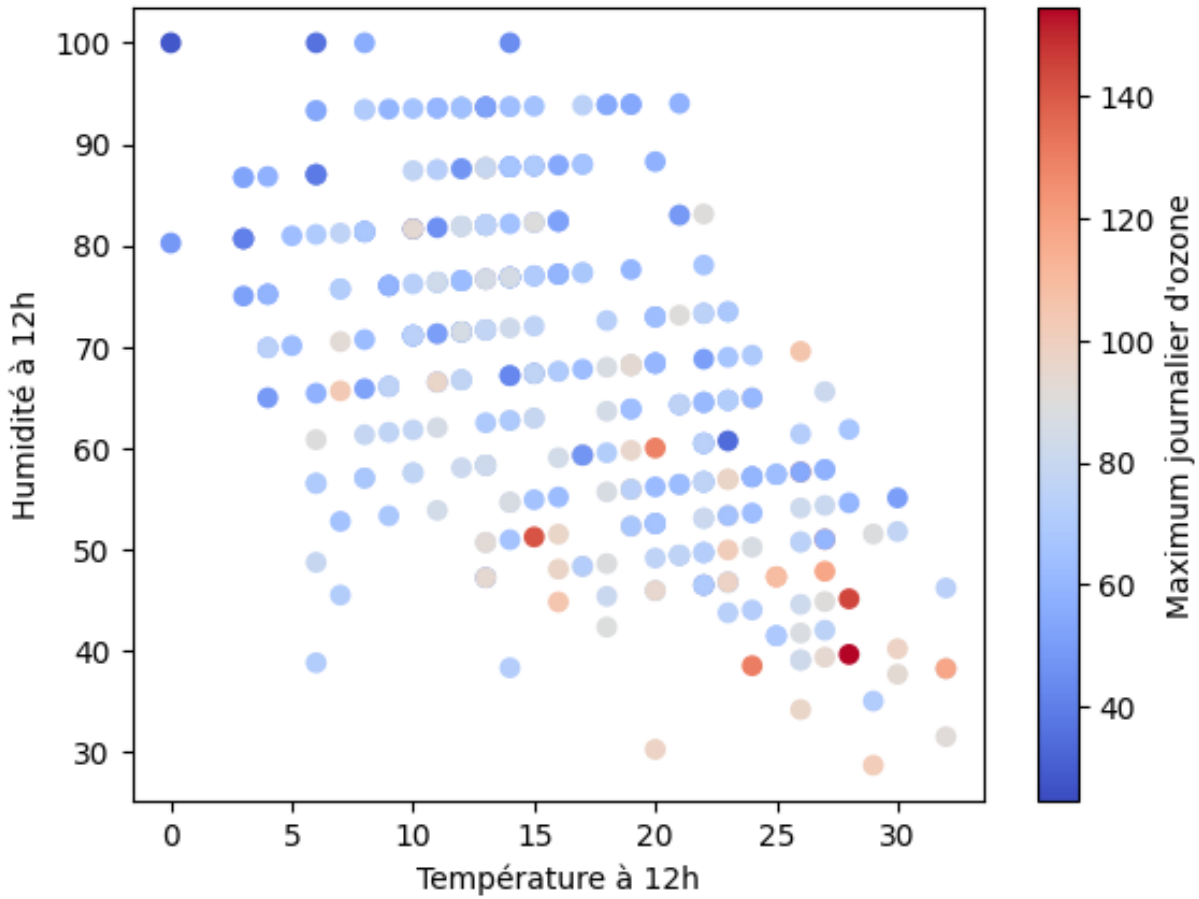
Figures from Hastie's book.

Outline

Outline

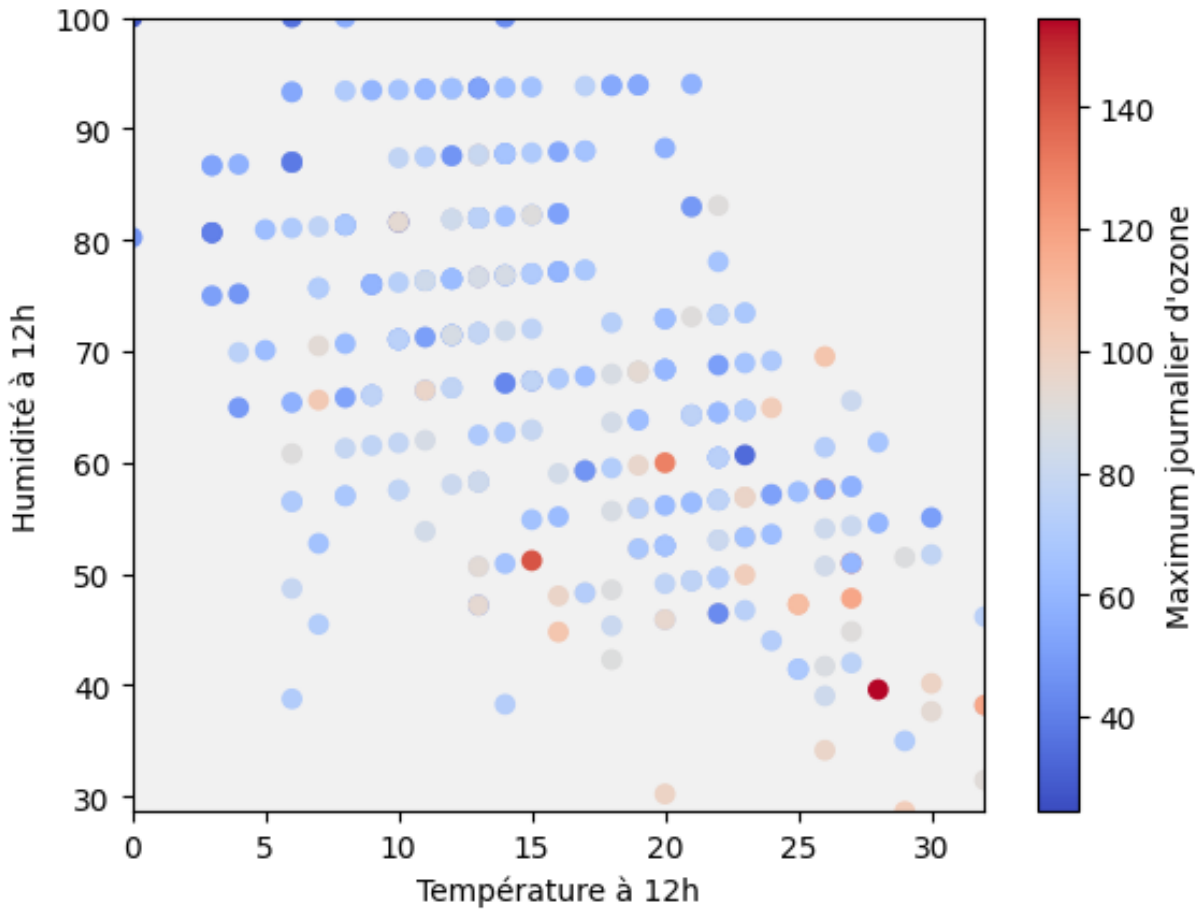
Jeu de données

- Maximum d'O3 journalier en fonction de la température et de l'humidité à 12h.



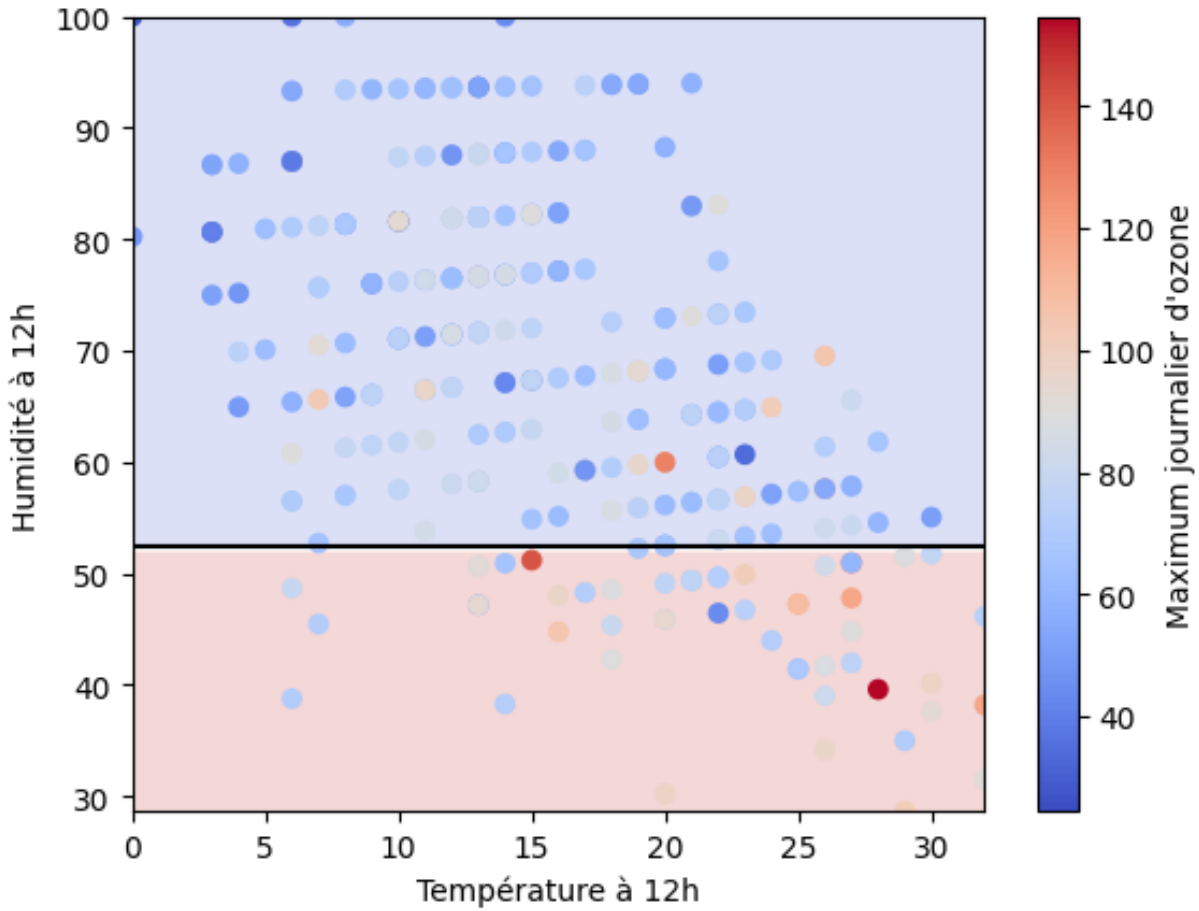
Jeu de données

- **Règle de décision** : $h(x) = \bar{y} = 72.78$



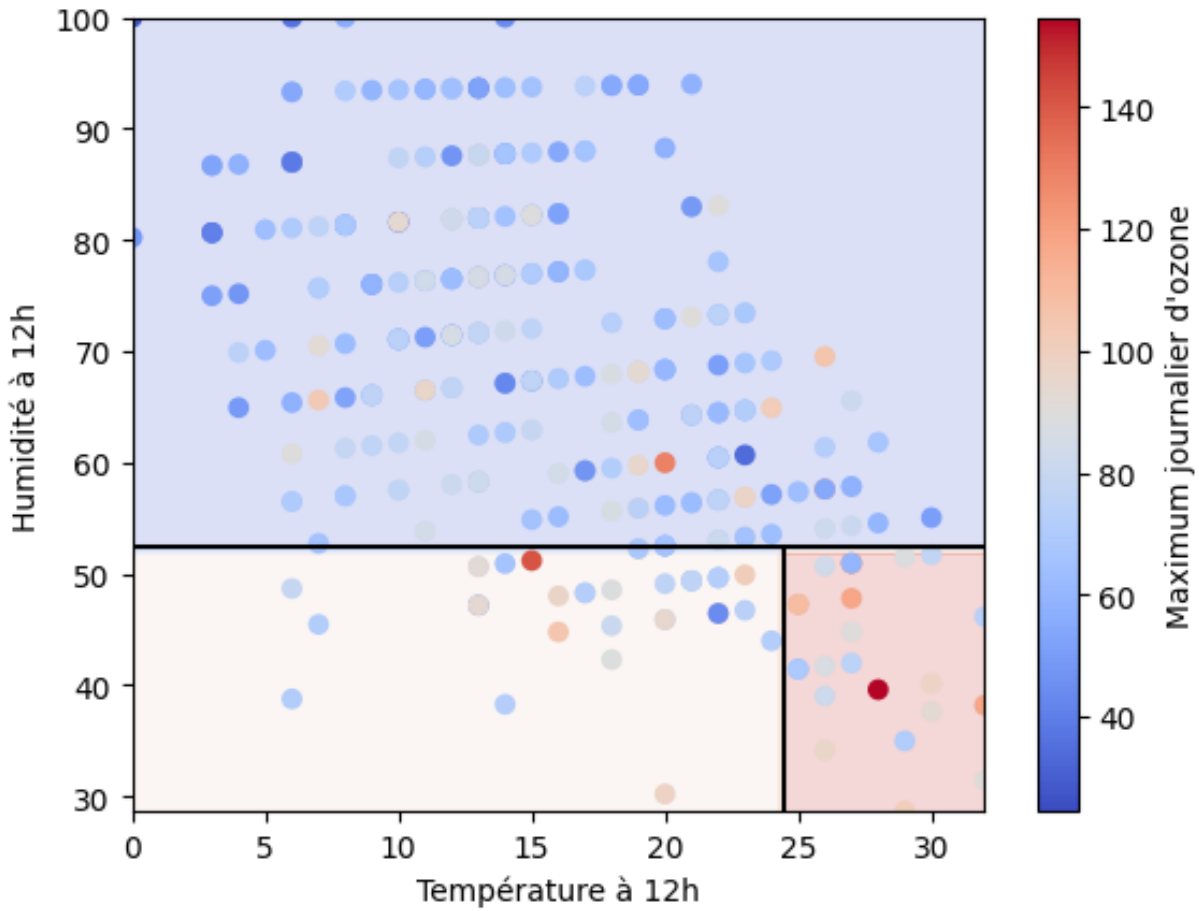
Construction première partition

- **Première partition** : $x_{humidite,12h} \leq 52.4$
- **Règle de décision** : $h(x) = 86.5 * \mathbb{1}_{\{x_{humidite,12h} \leq 52.4\}} + 67 * \mathbb{1}_{\{x_{humidite,12h} > 52.4\}}$



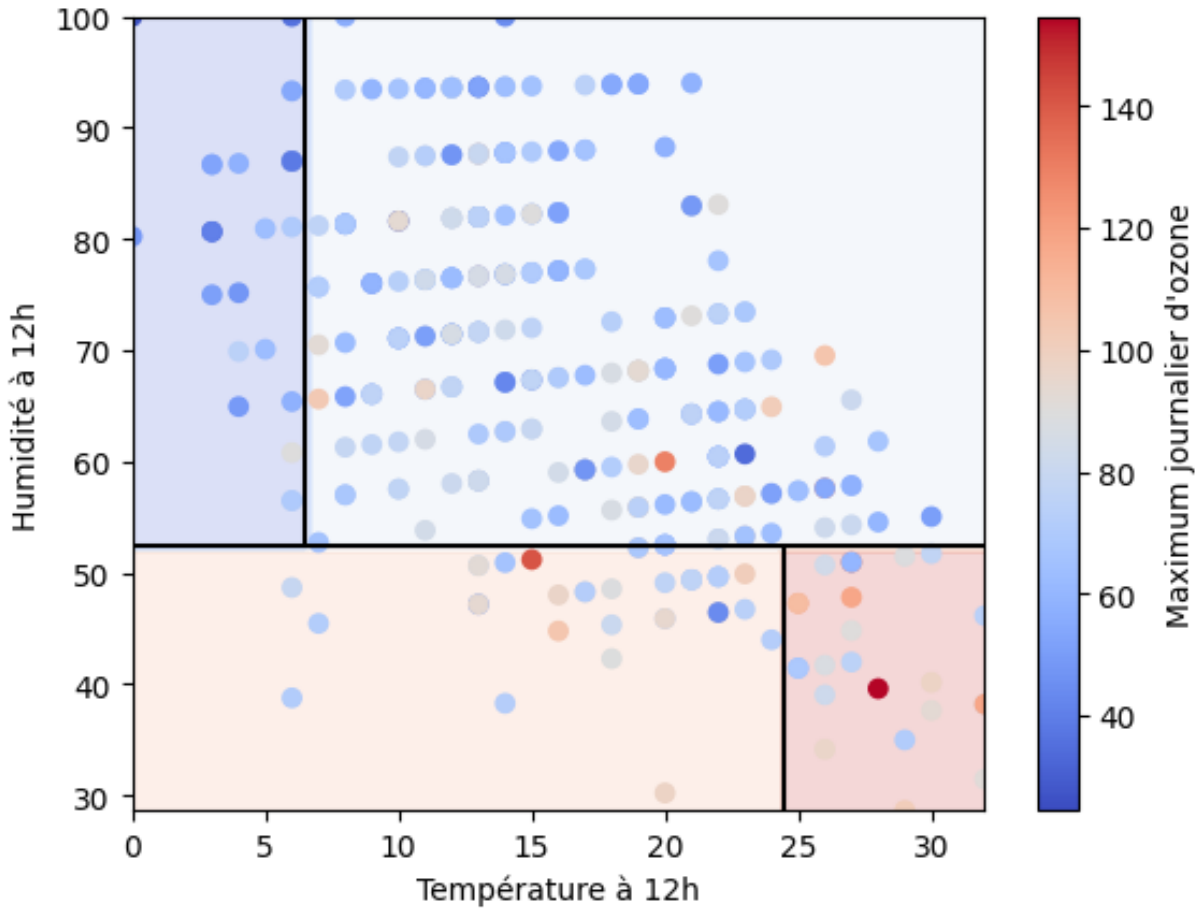
Construction deuxième partition

- **Deuxième partition** : $x_{humidite,12h} \leq 52.4$ et $x_{temperature,12h} \leq 24.50$
- **Règle de décision** : $h(x) = (82 * \mathbb{1}_{\{x_{temperature,12h} \leq 24.50\}} + 93 * \mathbb{1}_{\{x_{temperature,12h} \leq 24.50\}})\mathbb{1}_{\{x_{humidite,12h} \leq 52.4\}} + 67 * \mathbb{1}_{\{x_{humidite,12h} > 52.4\}}$

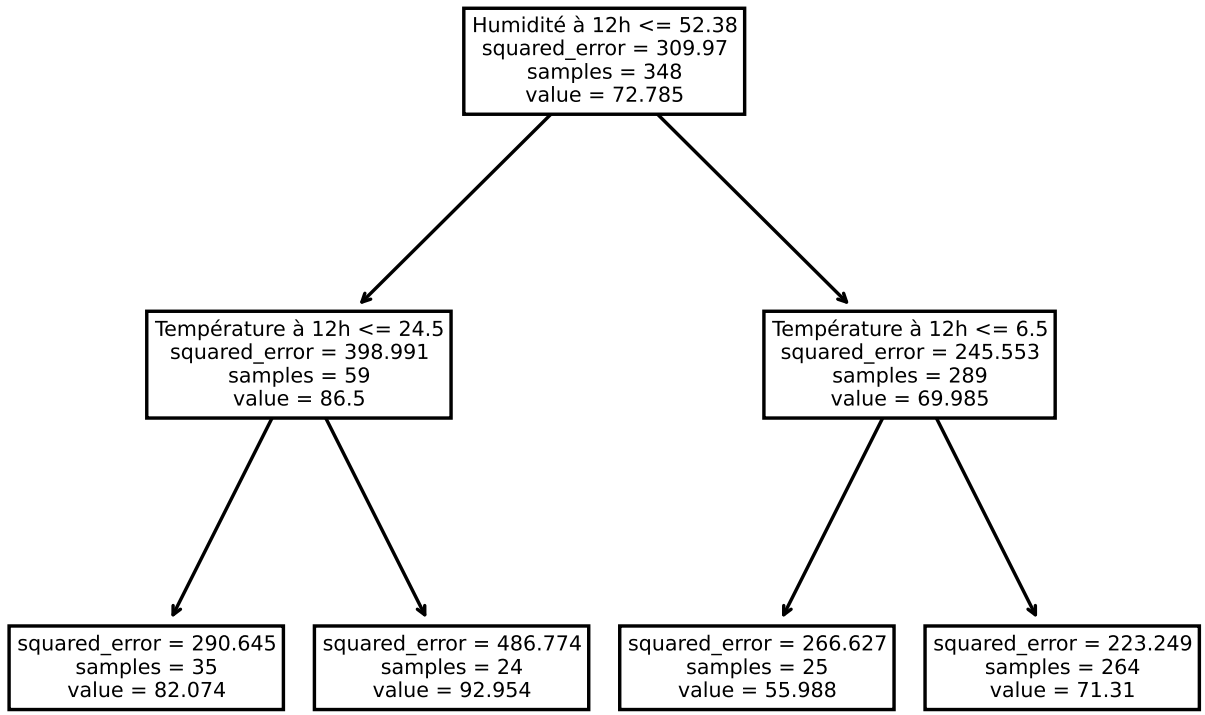


Construction troisième partition

- Nouvelle partition et nouvelle règle de décision.
- Arbre de profondeur 2.



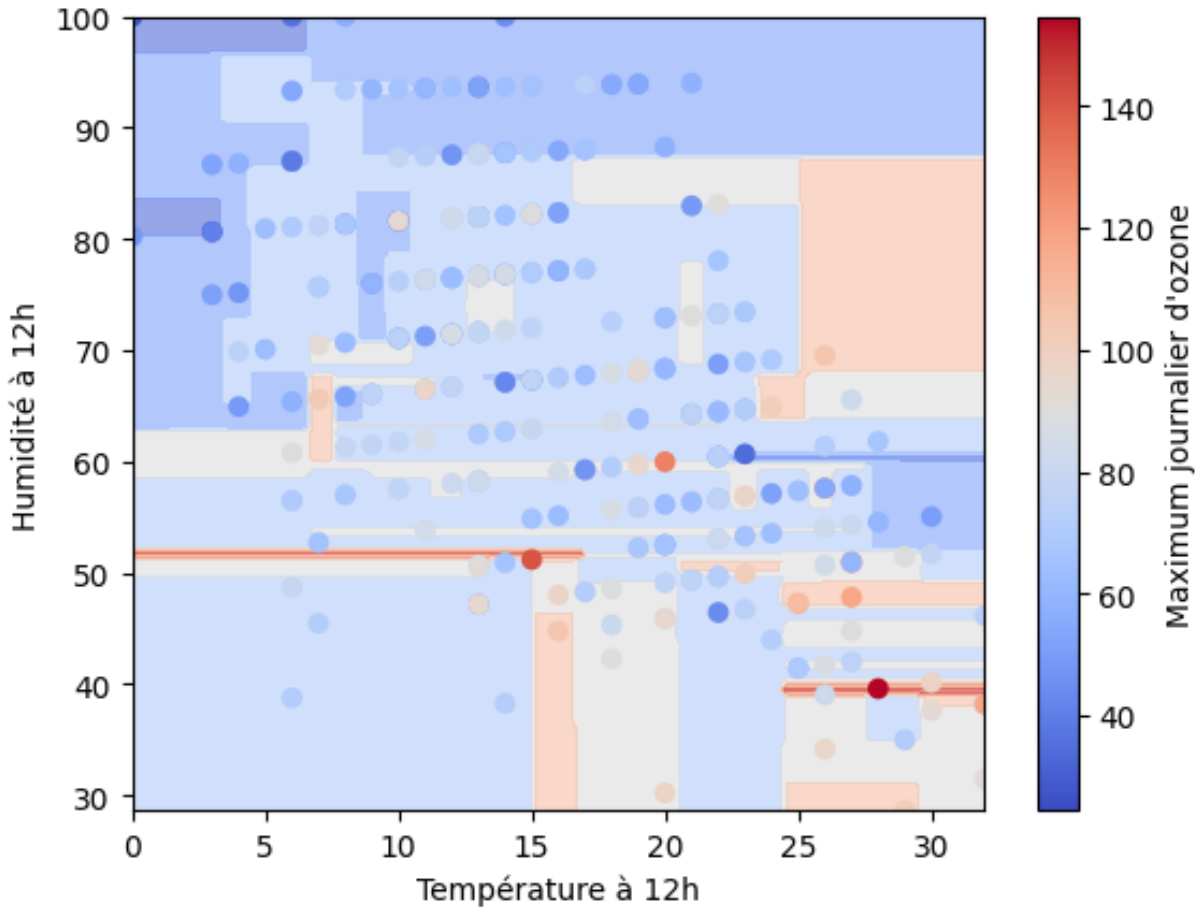
Représentation graphique



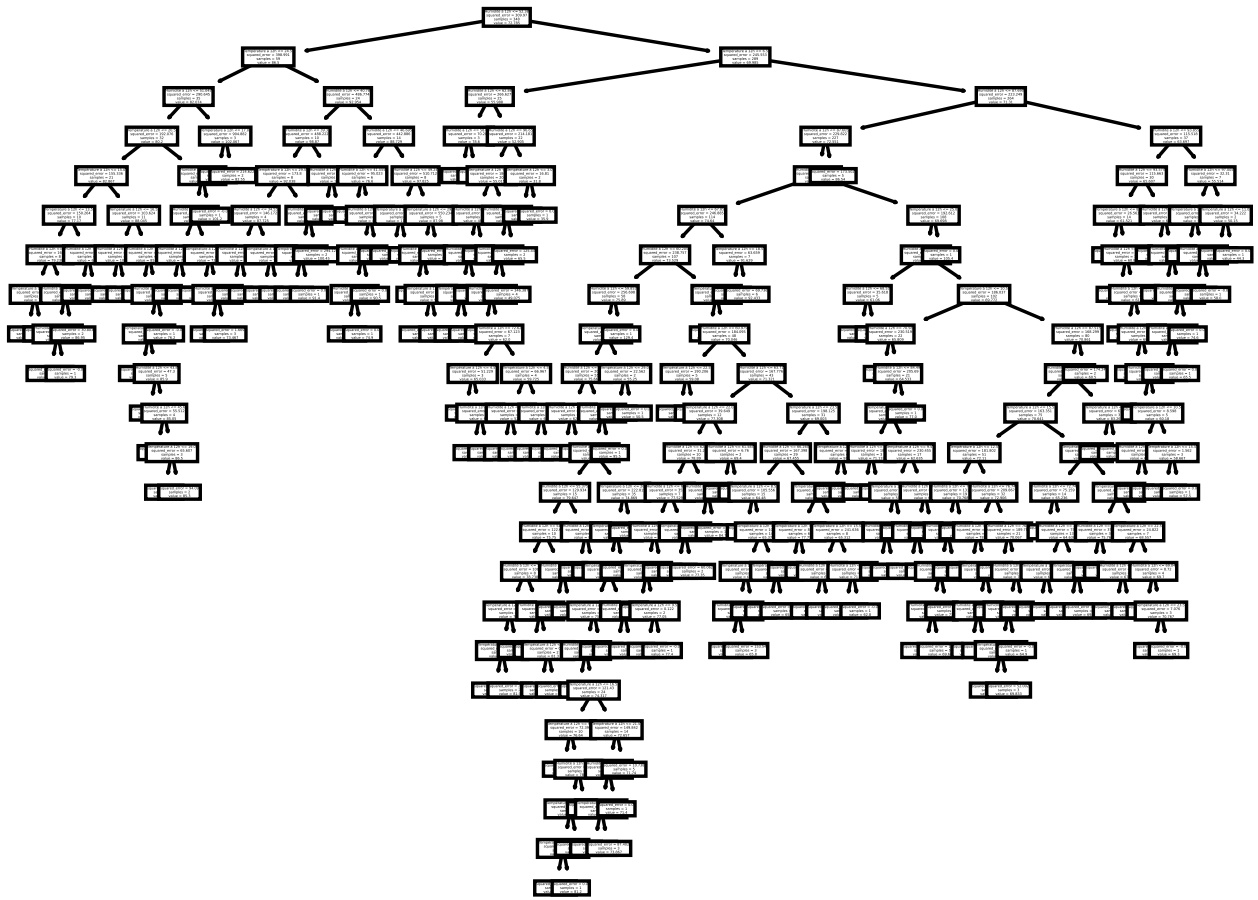
- Dans chaque **noeud**, on a le **nom** de la variable qui **divise** ainsi que son **seuil de division**, l'erreur aux moindres carrés et le nombre d'observation de la sous-région ainsi que la prédiction associée.
- Les noeuds **terminaux** sont appelés **"feuilles"**.

Jusqu'où s'arrêter ?

- 184 divisions.
- Arbre de profondeur 22.



Représentation graphique

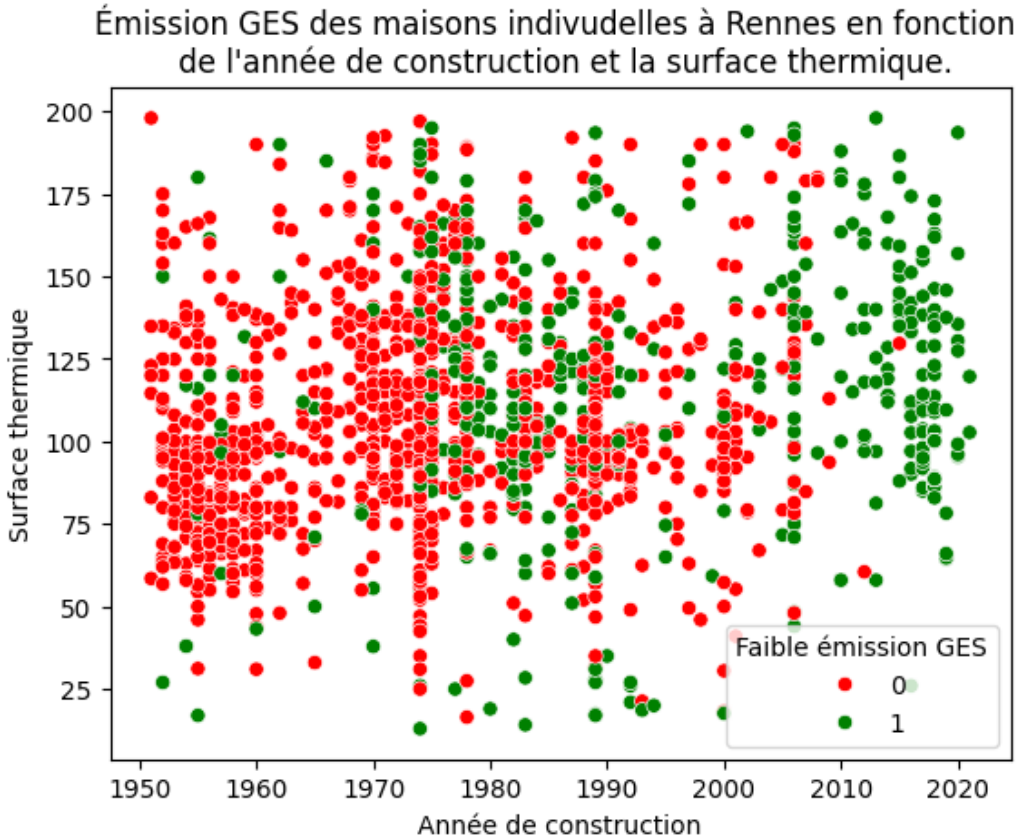


- Arbre **très très grand** !

Outline

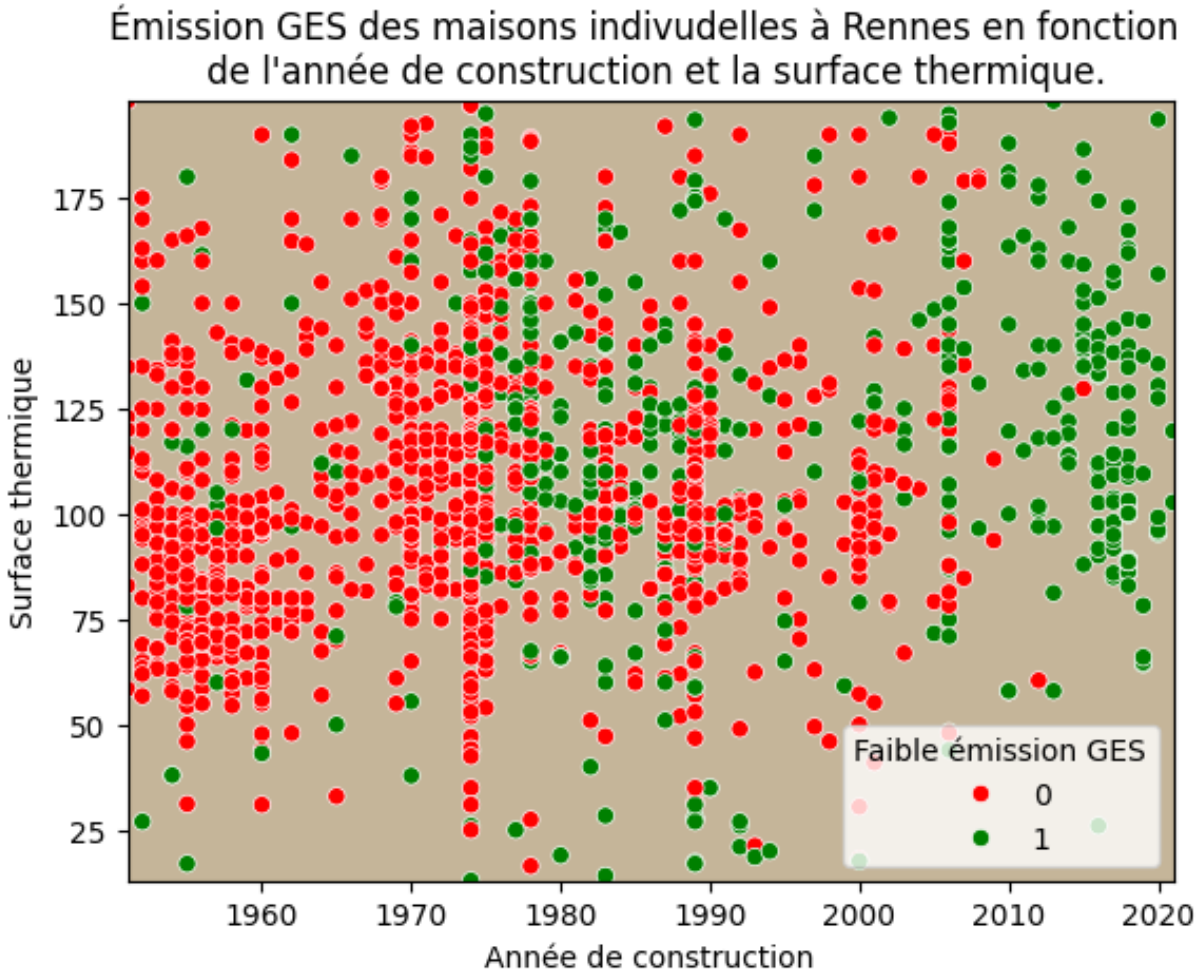
Jeu de données

- Émission GES des maisons individuelles à Rennes en fonction de l'année de construction et la surface thermique.



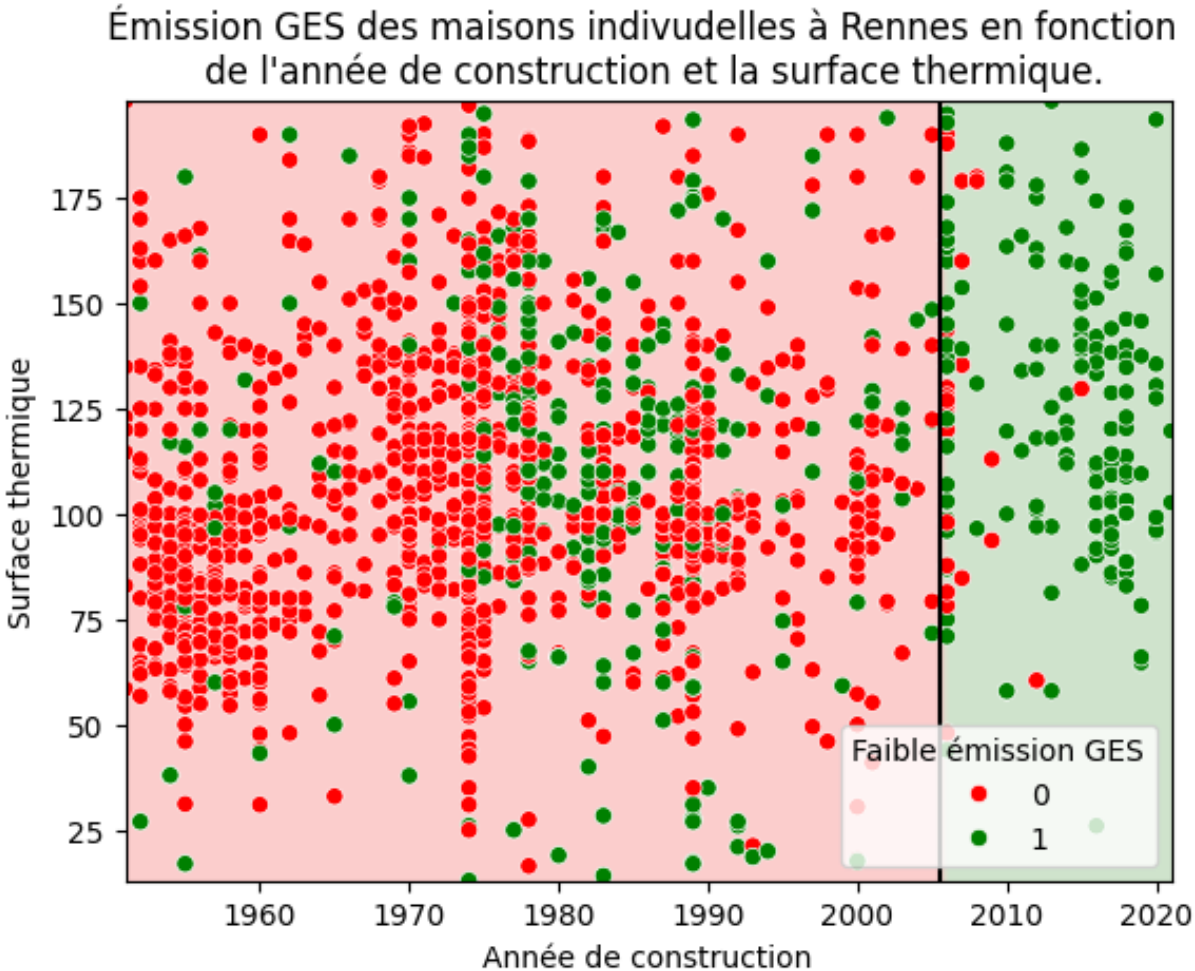
Jeu de données

- **Règle de décision** : $h(x) = \bar{y} = (probabilit d' trefaiblemission)$

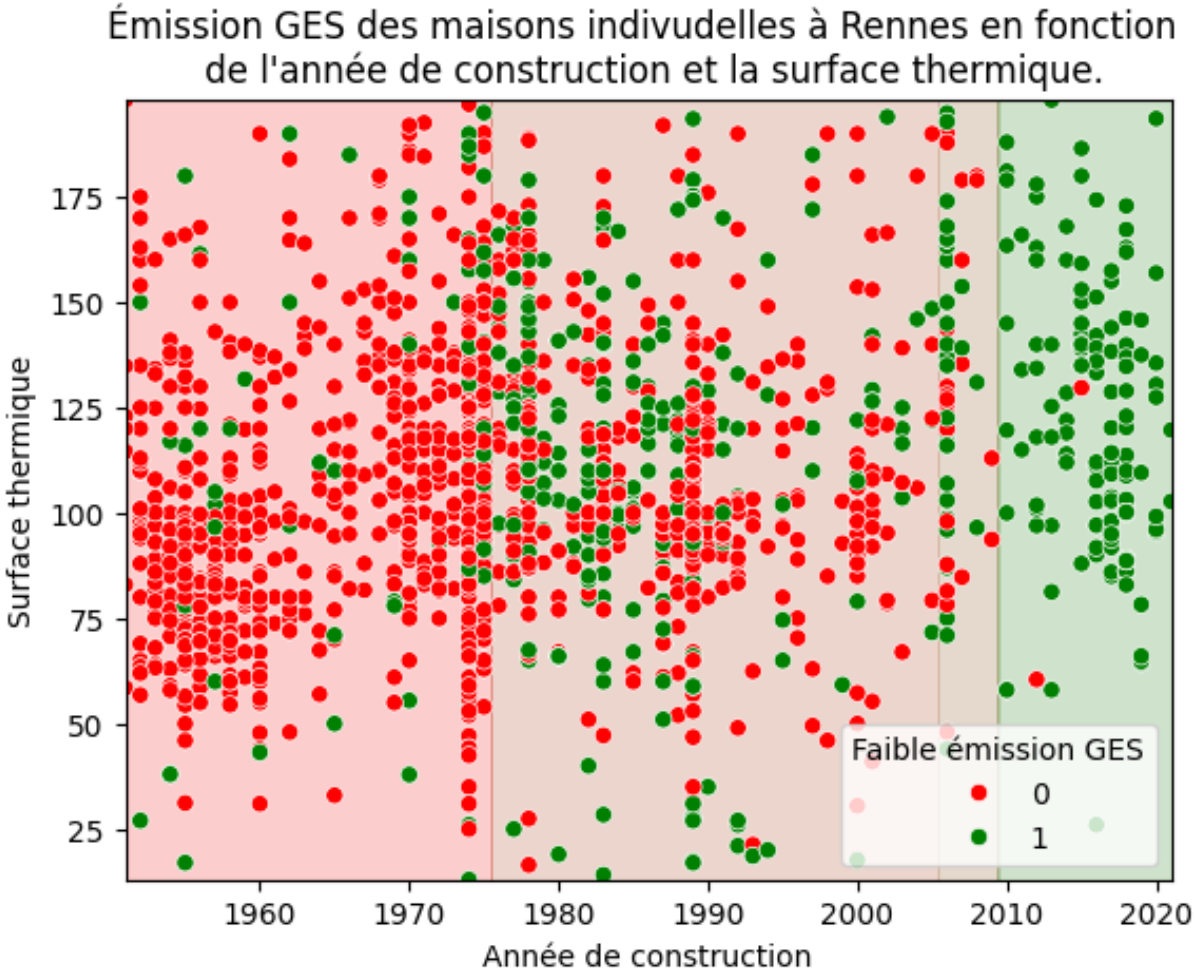


Construction première partition

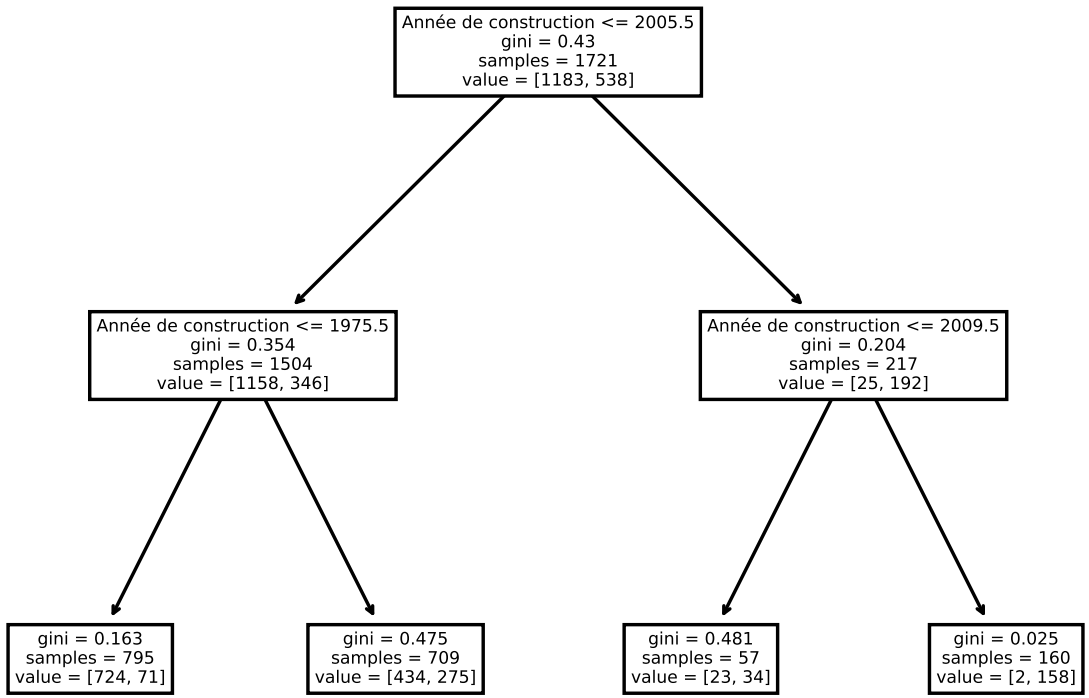
- **Première partition** : $x_{annee} \leq 2005.5$
- **Règle de décision** : $h(x) = 0.23 * \mathbb{1}_{\{x_{annee} \leq 2005.5\}} + 0.88 * \mathbb{1}_{\{x_{annee} > 2005.5\}}$



Arbre de pronfondeur 2



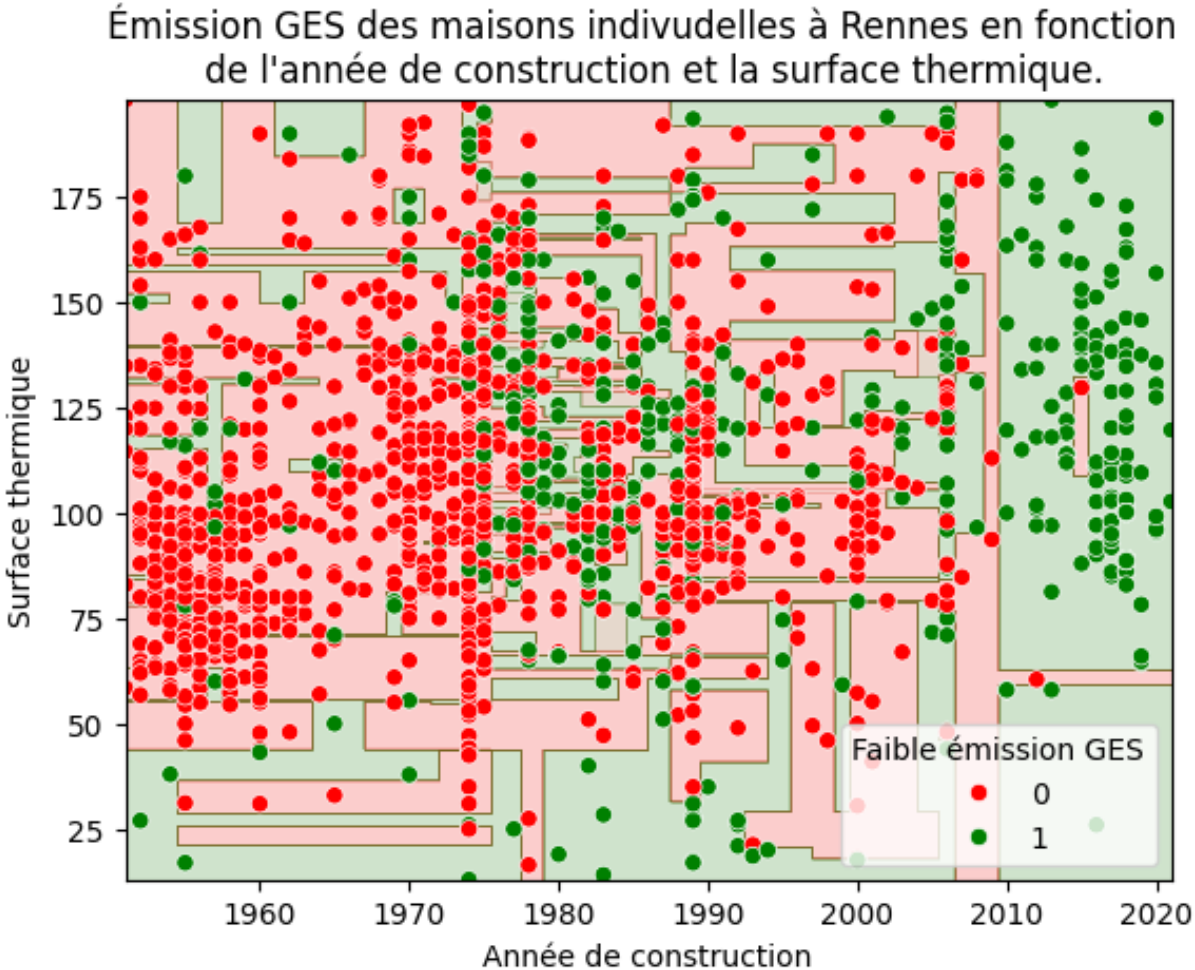
Représentation graphique



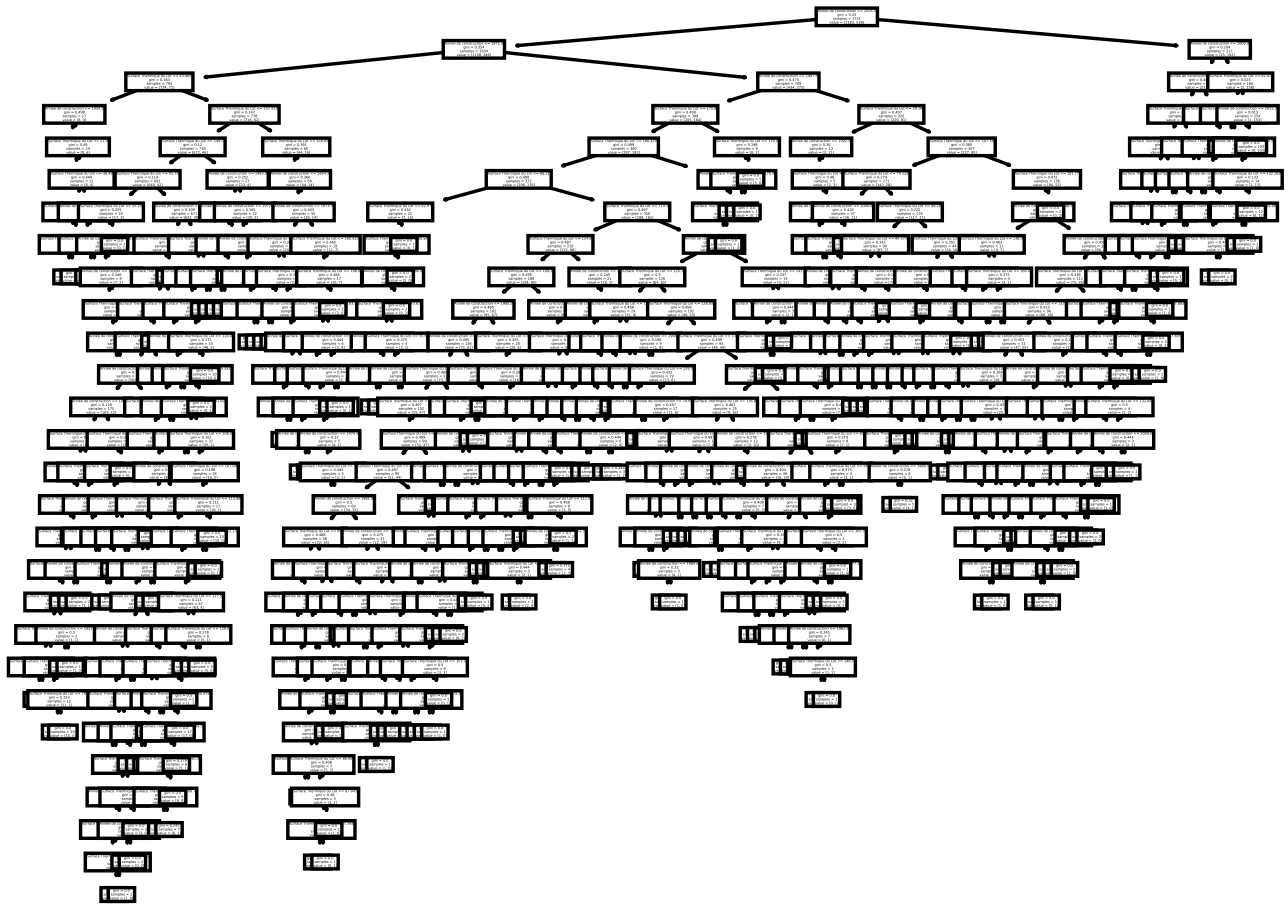
- Dans chaque **noeud**, on a le **nom** de la variable qui **divise** ainsi que son **seuil de division**, le critère de gini et le nombre d'observation correspondant à chaque classe.
- Les noeuds **terminaux** sont appelés **"feuilles"**.

Jusqu'où s'arrêter ?

- 446 divisions.
- Arbre de profondeur 27.



Représentation graphique



- Arbre **très très grand** !

Outline

Outline

Comment construire l'arbre ?

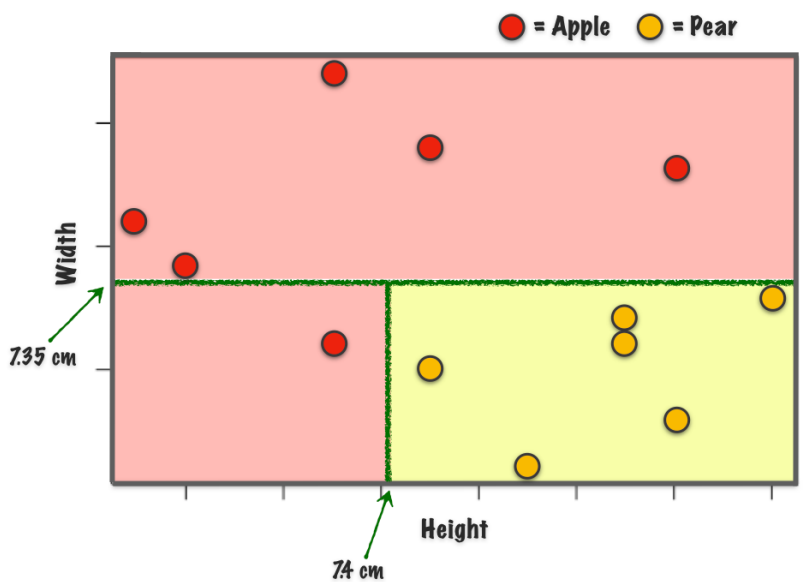
Objectifs :

- **Partitionner** l'espace en rectangle afin d'obtenir des **modèles simples** dans chaque région(moyenne, vote à la majorité)
- **Obtenir** des régions finales (ou feuilles) **pures**, c'est à dire avec une seule classe ou des valeurs proches.

Problème : Sur un jeu de **données** beaucoup plus **important**, il sera **compliqué** d'utiliser une méthode visuelle/graphique pour découper les régions.

Idée n°1 : Test de tous les partitionnements possibles et prendre celle qui a l'erreur la plus faible.

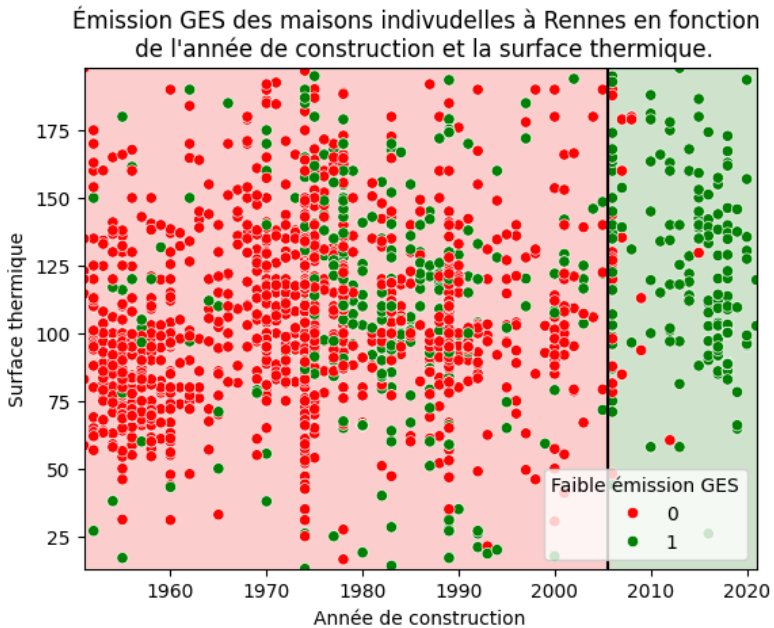
Problème : Problème NP-difficile ...



Construction d'un arbre de décision : algorithme glouton/heuristique

Idée n°2 : Approche itérative et heuristique.

- Les arbres sont estimés par un algorithme itératif qui construit une suite de partitions imbriquées.



- A chaque itération, l'algorithme choisit la variable et le seuil qui permettent d'obtenir une nouvelle division pour une sous région de la partition.
- **Critère de division** : Séparation en 2 régions les plus **homogènes** possibles.

Quand la partition **finale** ($\{R_m\}_{\{m=1,\dots,M\}}$) est obtenue, la **fonction de décision** est :

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbb{I}_{R_m}(\mathbf{x}) \text{ avec , } \begin{cases} c_m = \frac{1}{N_m} \sum_{\{i|\mathbf{x}_i \in R_m\}} y_i & \text{pour la régression} \\ c_m = \arg\max_{k \in \mathcal{Y}} \frac{1}{N_m} \sum_{i|\mathbf{x}_i \in R_m} \mathbb{I}(y_i = k) & \text{pour la classification} \end{cases}$$

Une itération de l'algorithme

- Considérons toutes les données de la région R_m, une variable de division x_j et un seuil s, on définit les sous régions

$$R_{m1}(j,s) = \{\mathbf{x} \in R_m | x_j \leq s\} \text{ and } R_{m2}(j,s) = \{\mathbf{x} \in R_m | x_j > s\}$$

- On cherche alors la variable x_{j*} et le seuil s^* qui maximise le gain d'information.
  - Le gain d'information est la différence entre l'impureté de la région à partitionner - la somme des impuretés des 2 nouvelles régions.
  - Il faut trouver le réel x_{j*} et s^* qui divisera au mieux la population de la région en deux ensembles les plus homogènes possibles.
  - Mathématiquement, en dénotant $R \in \mathcal{X} \rightarrow \text{Imp}(R)$ comme l'impureté de la région R,

$$(j^*, s^*) = arg \max_{j,s} Gain(x_j, s)$$

$$arg \max_{j,s} Imp(R_m) - \left(\frac{|R_{m1}(j,s)|}{|R_m(j,s)|} Imp(R_{m1}(j,s)) + \frac{|R_{m2}(j,s)|}{|R_m(j,s)|} Imp(R_{m2}(j,s)) \right)$$

- Pour chaque variable, le choix du seuil s est (numériquement) très rapide et la sélection de la meilleure paire (j,s) est faisable.

Outline

Variance

Objectif : Définir des mesures d'**impuretés**

En fonction de la **tâche** effectuée, les **indices** sont **différents** :

- **Régression** : Variance.
- **Classification** : Gini, Entropie.

La **variance** est une mesure de **dispersion** parfaitement adapté pour mesurer l'homogéinité d'une région.

Mathématiquement, en notant N_m le nombre d'éléments dans le région R_m, il s'exprime par

$$Imp^{variance}(R_m) = \sum_{\{i|\mathbf{x}_i \in R_m\}} (y_i - c_m)^2$$

avec $c_m = \frac{1}{N_m} \sum_{\{i|\mathbf{x}_i \in R_m\}} y_i$.

Critère de Gini

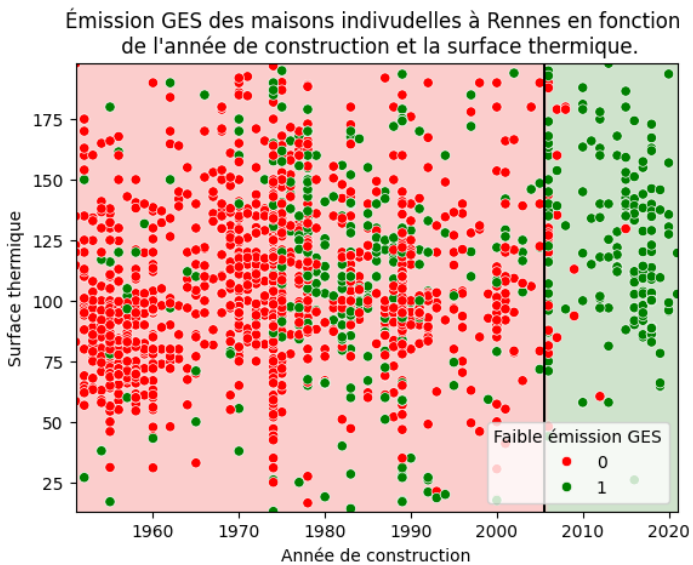
- L'**indice de Gini** est une mesure d'**inégalité**. Il a été introduit pour mesurer des inégalités de revenu.
  - Il est égal à 0 en cas de totale égalité (homogénéité parfaite) et 1 en cas de totale inégalité.

- Mathématiquement, en notant K le nombre de classe et N_m le nombre d'éléments dans le région R_m, il s'exprime par

$$Imp^{gini}(R_m) = 1 - \sum_{k=1}^{K} p_{mk}^2$$

avec $p_{mk} = \frac{1}{N_m} \sum_{i|x_i \in R_m} \mathbb{I}\{y_i = k\}$.

Dans le cas d'une variable dichotomique, l'indice de Gini calcule une variance. Il est donc très proche du critère utilisé en régression.



- Indice de Gini R_0 : 0.43
- Indice de Gini R_1 : 0.354
- Indice de Gini R_2 : 0.204

Critère d'Entropie

- L'**entropie** peut être interprétée comme une **mesure de désordre**.
  - Une entropie nulle correspond à un milieu parfaitement ordonné (ie une sous région parfaitement homogène).

- Mathématiquement, en notant K le nombre de classe et N_m le nombre d'éléments dans le région R_m, il s'exprime par

$$Imp^{entropie}(R_m) = \sum_{k=1}^{K} p_{mk} \log p_{mk}$$

  avec $p_{mk} = \frac{1}{N_m} \sum_{i|x_i \in R_m} \mathbb{I}_{\{y_i = k\}}$.

- En général, on n'utilise pas l'erreur de **classification** car elle est moins **sensible** et a tendance à **sélectionner** des **divisions** qui conduisent à **des nœuds qui ne sont pas purs** (problème d'équilibre des classes.

Outline

Critères d'arrêt/Élagage

Les divisions sont répétées jusqu'à atteindre au moins un critère d'arrêt.

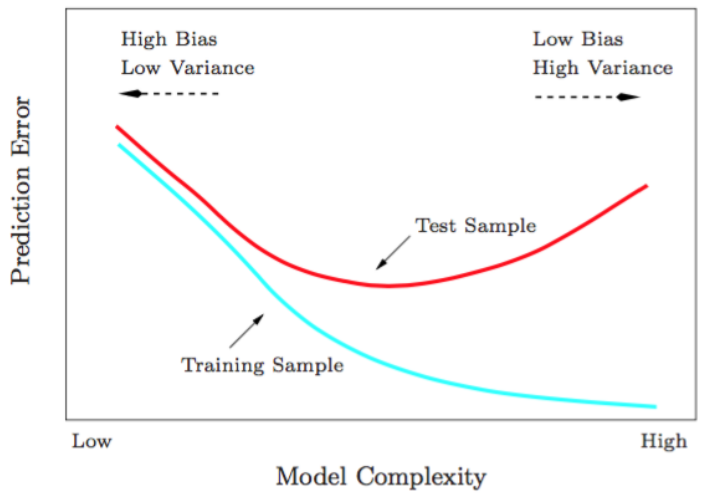
Les critères d'arrêt classiques sont

- le **nombre minimum d'observations** pour faire une division : si le nombre d'observations est faible l'estimation des variances risque d'être biaisée et bruitée ;
- le **nombre minimum d'observations par feuille** : si le nombre d'observations est faible la prédiction de la variable cible sera peu précise (forte variance d'estimation) ;
- la **profondeur maximum** : un arbre trop profond conduit à une situation de sur-apprentissage.

Une alternative consiste à considérer un **ensemble de validation** pour faire de la **validation croisée**. On **arrête** alors l'algorithme lorsque l'**erreur** de prédiction de l'ensemble de validation ne **décroît plus** assez.

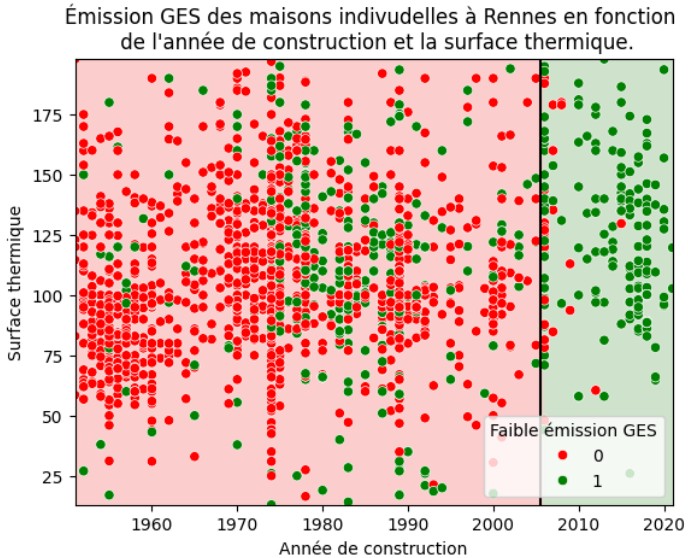
Une autre façon de poser la question : Quelle est la bonne taille pour un arbre de décision ?

- Compromis biais-variance : un arbre **profond** risque de faire du **sur-apprentissage** tandis qu'un arbre trop **petit** ne va pas permettre de capturer les structures importantes (**biais important**).

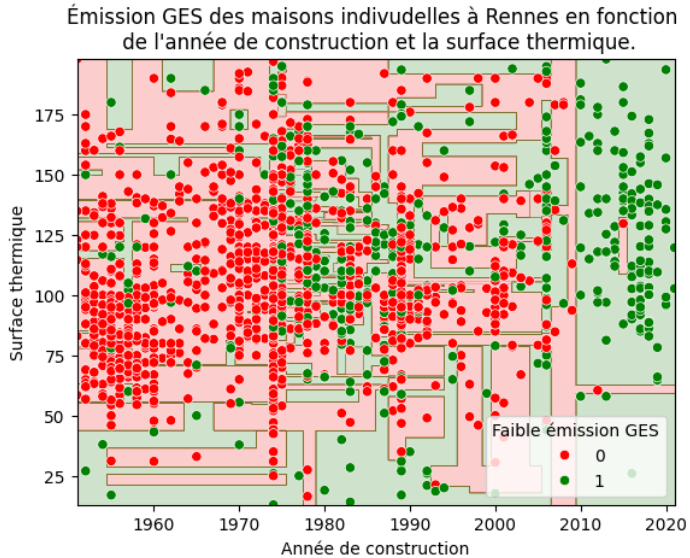


- La profondeur de l'arbre contrôle le compromis entre le biais et la variance
  - **Profondeur faible** (= peu de divisions) conduit à des arbres avec une **faible variance** mais un **biais important**
  - **Profondeur importante** (= beaucoup de divisions) conduit à des arbres avec une **forte variance** et un **biais faible**.

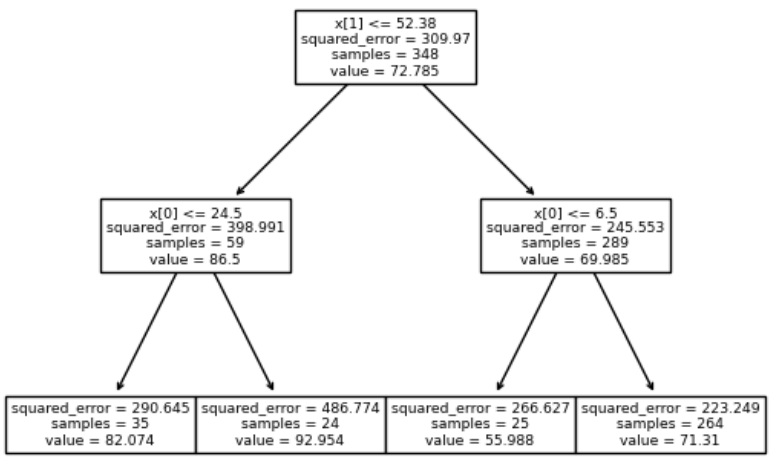
Exemple



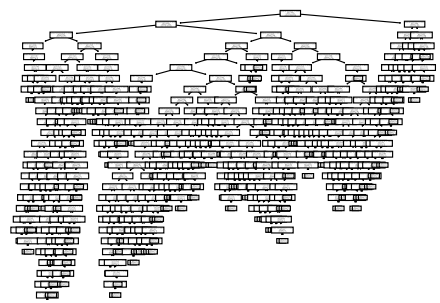
(a) Règle de décision arbre de profondeur 1



(b) Règle de décision arbre de profondeur 27



(c) Représentation graphique de l'arbre de profondeur 1



(d) Représentation graphique de l'arbre de profondeur 27

Optimiser la profondeur d'un arbre de décision

- La stratégie classique est de construire un arbre trop profond $\mathbb{T}_0$, en stoppant par exemple l'algorithme lorsque le nombre minimum d'observations par nœud est atteint. Puis l'arbre est élagué pour optimiser un critère coût-complexité.
- Le critère coût-complexité est défini par

$$C_\alpha(\mathbb{T}) = \textit{erreur}(\mathbb{T}) + \underbrace{\alpha|\mathbb{T}|}_{\text{complexity}}$$

avec $|\mathbb{T}|$ définit la profondeur de l'arbre.

- L'idée est de trouver pour chaque α, le sous arbre $\mathbb{T}_\alpha$ de $\mathbb{T}$ qui minimise $C_\alpha(\mathbb{T})$.
- α est appelé complexity parameter et permet d'avoir un **unique** paramètre à tuner.
- En pratique, on sélectionne α par validation croisée.

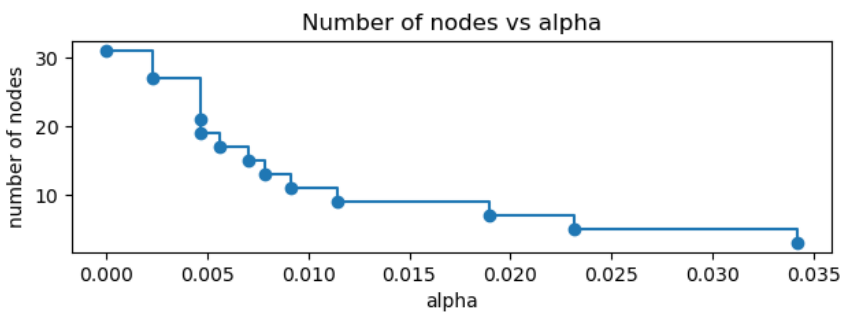


Figure – Évolution du nombre de noeuds avec α

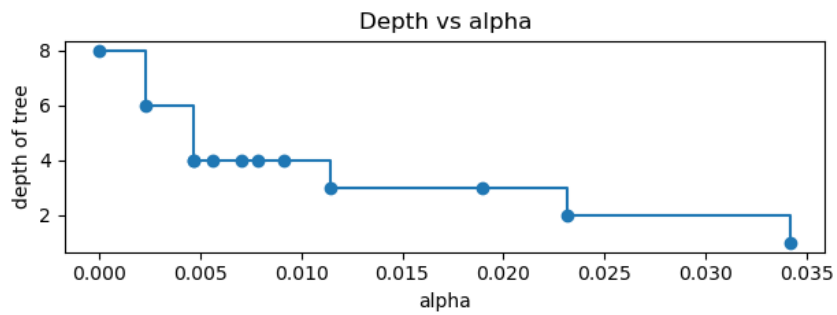


Figure – Évolution de la profondeur avec α

Exemples d'élagage par validation croisée

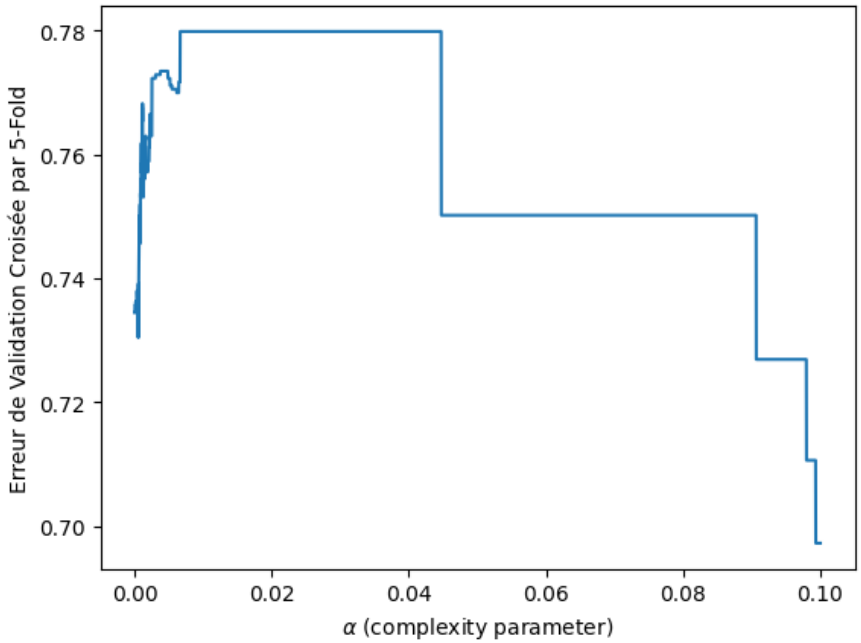


Figure – Évolution de l'erreur par validation croisée 5-Fold en fonction de α

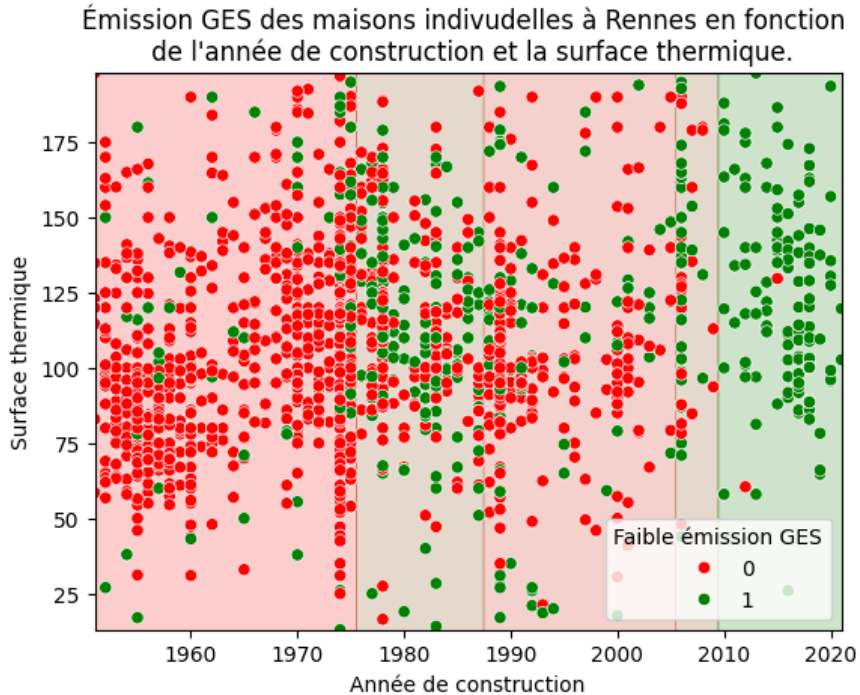


Figure – Visualisation de la règle de décision du meilleur arbre obtenu.

- Arbre **peu profond** et décision sur une **seule variable**.
- Pas de **surajustement** mais la deuxième variable est inutile. Il serait intéressant de créer/trouver de nouvelles **variables** plus **complexes**/**corrélés** avec la réponse.

Exemples d'élagage par validation croisée

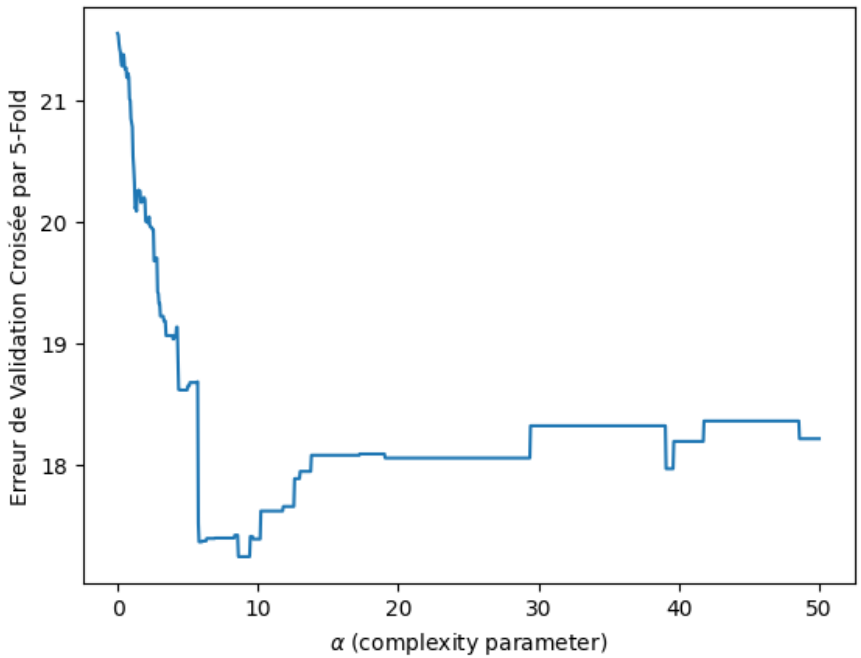


Figure – Évolution de l'erreur par validation croisée 5-Fold en fonction de α

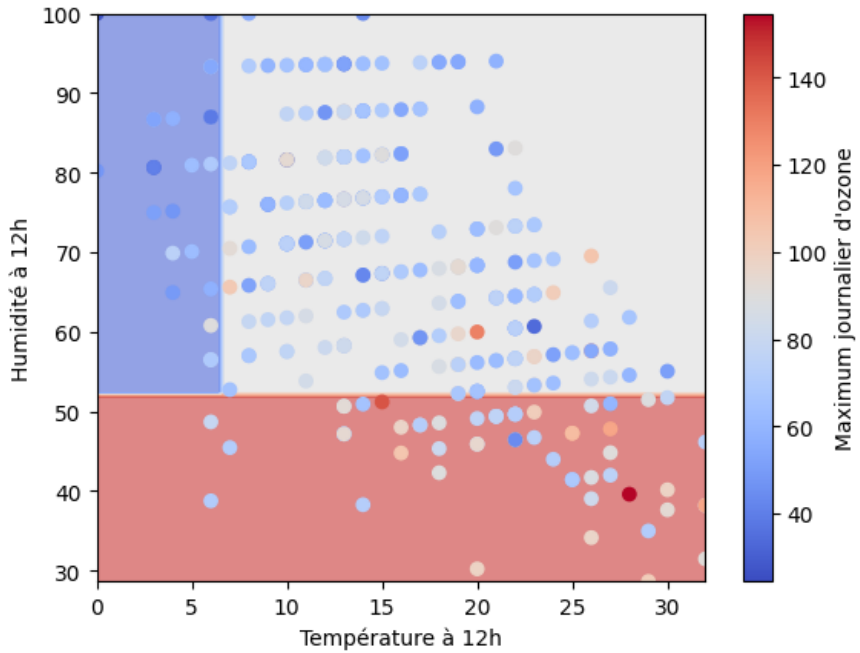


Figure – Visualisation de la règle de décision du meilleur arbre obtenu.

- Arbre **peu profond** mais décision sur les 2 **variables**.
- Pas de **surajustement** mais il faudrait sûrement ici **rajouter des variables** pour diminuer le biais.

Outline

Remarques de conclusion

Les arbres de décision ont un certain nombre d'avantages :

- Ce sont des modèles **non paramétriques** (très proches du modèle des ppv) qui permettent de prendre en compte des **relations non linéaires** entre les variables.
- Prédicteur **intuitif** : Ressemble à la manière de prendre des décisions des humains.
- Les arbres nécessitent **peu ou pas de pré-traitement**.
  Par exemple, il est inutile de faire des transformations marginales. En particulier, les outliers des variables prédictives ont relativement peu d'impact sur le choix des divisions.
- Les arbres prennent en compte les **variables explicatives catégorielles à plus de 2 modalités**. Voir J. Friedman, Hastie, and Tibshirani (2001).
- Les valeurs manquantes ne posent pas non plus de problème Une solution classique consiste à créer pour chaque variable une catégorie "missing".

Mais, les **performances** prédictives des arbres de décision sont souvent un peu **en dessous** de celles d'autres algorithmes. Ceci est dû au fait que le arbres sont composés par des **règles très simples** qui conduisent à des décisions qui ne sont pas lisses.
De plus les **arbres profonds** ont une **forte variance** et les arbres **peu profonds** un **fort biais**.

Références

Breiman, Leo. 1984. Classification and Regression Trees. Routledge.

Breiman, Leo, and Ross Ihaka. 1984. Nonlinear Discriminant Analysis via Scaling and Ace. Department of Statistics, University of California.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. The Elements of Statistical Learning. Vol. 1. Springer Series in Statistics New York, NY, USA :

Shalev-Shwartz, S., Ben-David, S. (2014). Understanding Machine Learning - From Theory to Algorithms.. Cambridge University Press. ISBN : 978-1-10-705713-5