



Fouilles de données et extraction de connaissances

---

Fouilles de données au service du développement durable

---

Moh-Ouali AMROUNI

Rayane BELKALEM

Valliamma-Raphael SOUPAYA

# 1 Introduction

Les services municipaux disposent aujourd’hui de données riches sur l’état des arbres, leur environnement et les interventions recommandées. L’objectif du projet est d’exploiter ces informations pour construire des modèles prédictifs utiles au suivi des espaces verts et à la planification de la maintenance.

## 2 Défi 1 : prédire la présence et la localisation d’un défaut

### 2.1 Jeu de données et variables cibles

Le jeu de données contient **15 375** observations. Chaque observation décrit un arbre au travers de variables numériques (coordonnées, mesures, etc.) et de variables catégorielles (type, environnement, diagnostic, etc.). Les variables cibles sont binaires :

- **Uni-label** : **DEFAULT** (0 : pas de défaut, 1 : défaut).
- **Multi-label** : **Collet**, **Houppier**, **Racine**, **Tronc**.

### 2.2 Prétraitement des données

Le prétraitement suit trois idées simples : (1) supprimer les colonnes non informatives, (2) gérer proprement les valeurs manquantes, et (3) encoder les variables catégorielles de manière compatible avec les modèles.

#### Gestion des valeurs manquantes

Dans le fichier brut, les valeurs manquantes sont notées "?". Elles ont été converties en **NaN** afin d’être reconnues par **pandas** et **scikit-learn**. Une analyse de complétude (heatmap + barplot) a permis d’identifier des variables très incomplètes. Nous supprimons les colonnes dont le taux de valeurs manquantes est supérieur à **70%**, car elles contiennent trop peu d’information exploitable et rendent l’imputation artificielle.

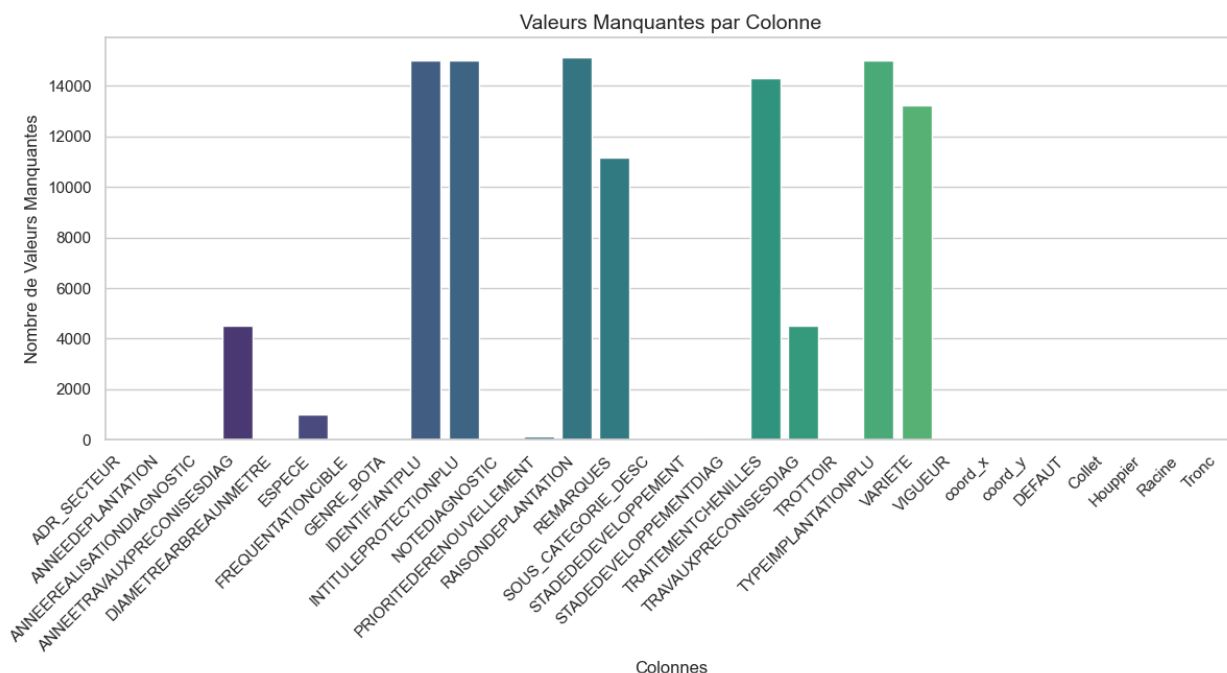


FIGURE 1 – Nombre de valeurs manquantes par variable (à exporter depuis le notebook).

#### Réduction / nettoyage de colonnes

Certaines colonnes de type identifiant (ex. **CODE**, **CODE\_PARENT**, descriptions associées) ont été supprimées, car elles contiennent quasiment autant de modalités que d’observations, et risquent d’introduire du bruit.

#### Imputation et encodage (sans fuite de données)

- Pour l’apprentissage, les traitements sont intégrés dans des **pipelines** afin d’éviter toute fuite de données :
- Variables numériques : imputation par la médiane.
  - Variables catégorielles : imputation par la modalité la plus fréquente.
  - Encodage : *Ordinal Encoding* (arbres, forêts, XGBoost), et *One-Hot Encoding* (régression logistique).

À noter : l'imputation/encodage « exploratoires » (moyenne/mode + label encoding) n'est utilisée que pour l'EDA (corrélations), pas pour entraîner les modèles.

### Vérification de cohérence des cibles

Nous avons vérifié la cohérence entre **DEFAULT** et l'union des 4 labels de localisation (**DEFAULT** = OR(**Collet**, **Houppier**, **Racine**, **Tronc**)) : aucune incohérence n'a été détectée sur l'ensemble des observations.

## 2.3 Tâche 1 : classification uni-label (DEFAULT)

### Protocole expérimental

La base est divisée en **80% entraînement** et **20% test** en conservant la proportion des classes (split stratifié). Plusieurs modèles sont comparés : régression logistique, arbre de décision, random forest, XGBoost, ainsi qu'une baseline (classe majoritaire).

L'évaluation est réalisée à l'aide de :

- métriques : accuracy, précision, rappel, F1 (pour la classe **DEFAULT=1**),
- AUC ROC et Average Precision (AP),
- validation croisée (5-fold) pour stabiliser la comparaison.

### Résultats comparatifs

Le tableau 1 synthétise les performances sur le jeu de test et en validation croisée. On observe que **Random Forest** et **XGBoost** dominent globalement, avec un AUC supérieur à 0.92, et un bon compromis précision/rappel sur la classe positive.

Modèle	Acc. test	Prec. (1)	Rec. (1)	F1 (1)	AUC test	Acc. CV (5)
Régression logistique	0.8374	0.7515	0.7470	0.7492	0.8858	0.8356
Arbre de décision	0.8371	0.7646	0.7210	0.7422	0.8070	0.8370
Random Forest	0.8823	0.8568	0.7660	0.8089	0.9275	0.8775
XGBoost	0.8797	0.8241	0.8010	0.8124	0.9299	0.8730

TABLE 1 – Tâche uni-label (DEFAULT) : comparaison des modèles. Les métriques précision/rappel/F1 concernent la classe positive (1).

## 2.4 Tâche 2 : classification multi-label (Collet, Houppier, Racine, Tronc)

### Approche

La multi-label est traitée via une stratégie **One-vs-All** : un classifieur binaire est entraîné pour chaque label. Les labels étant déséquilibrés (faible proportion de 1, surtout pour **Collet** et **Racine**), nous appliquons un **undersampling sur l'ensemble d'entraînement uniquement**, label par label.

### Niveau micro : résultats par attribut

Nous présentons ci-dessous les résultats « micro » (une table par attribut) comme dans le rapport précédent : accuracy, précision, rappel, AUC, et validation croisée (K=10) sur l'entraînement rééquilibré.

Modèle	Acc.	Prec.	Rec.	AUC	CV Acc (10)	CV AUC (10)
Régression logistique	76.94	17.00	75.14	83.83	77.20	84.79
Random Forest	79.19	19.44	80.66	87.43	79.54	87.39
XGBoost	78.83	19.56	83.43	87.46	79.07	86.38
Arbre de décision	80.91	19.61	72.38	83.58	75.06	80.97

TABLE 2 – Multi-label (micro) : performances pour **Collet** (valeurs en %).

Modèle	Acc.	Prec.	Rec.	AUC	CV Acc (10)	CV AUC (10)
Régression logistique	83.84	58.45	83.84	90.09	82.57	89.01
Random Forest	84.88	60.21	85.82	92.88	84.02	91.48
XGBoost	85.59	61.66	85.82	93.27	84.26	91.96
Arbre de décision	83.19	57.89	77.74	87.86	80.63	86.01

TABLE 3 – Multi-label (micro) : performances pour Houppier (valeurs en %).

Modèle	Acc.	Prec.	Rec.	AUC	CV Acc (10)	CV AUC (10)
Régression logistique	76.13	12.61	74.81	83.20	76.42	84.22
Random Forest	79.84	16.37	87.41	89.85	80.25	87.94
XGBoost	78.96	15.59	85.93	88.98	79.57	87.78
Arbre de décision	76.29	13.24	79.26	81.41	72.50	76.27

TABLE 4 – Multi-label (micro) : performances pour Racine (valeurs en %).

Modèle	Acc.	Prec.	Rec.	AUC	CV Acc (10)	CV AUC (10)
Régression logistique	80.91	39.84	69.75	82.07	77.75	84.37
Random Forest	78.57	37.39	77.37	85.54	80.27	86.94
XGBoost	79.87	39.36	79.45	85.83	79.16	86.69
Arbre de décision	83.45	44.24	67.44	81.92	78.18	83.06

TABLE 5 – Multi-label (micro) : performances pour Tronc (valeurs en %).

## Niveau macro : vision globale multi-label

Au niveau macro, on moyenne les performances sur les 4 labels afin d’obtenir une vision globale du système. On remarque un rappel élevé (lié à l’entraînement rééquilibré) mais une précision plus faible, particulièrement pour les labels rares.

Modèle	Accuracy (%)	Précision (%)	Rappel (%)	AUC (%)
Arbre de décision	80.96	33.75	74.20	83.69
Régression logistique	79.46	31.98	75.89	84.80
Random Forest	80.62	33.35	82.82	88.92
XGBoost	80.81	34.04	83.66	88.88

TABLE 6 – Multi-label (macro) : moyenne des performances sur les 4 attributs.

## Une autre méthode de classification multi-label

En complément de l’approche *un classifieur par attribut* (One-vs-All), nous avons testé deux méthodes multi-label :

- **MultiOutputClassifier** : entraîne un classifieur binaire par label via une interface unique (ici avec **Random Forest**).
- **Binary Relevance** : transforme le problème multi-label en plusieurs problèmes binaires indépendants (ici avec **Random Forest**).

Ces deux méthodes servent de **baselines multi-label** et permettent de vérifier la cohérence des résultats obtenus avec l’approche One-vs-All. Nous présentons ci-dessous des visualisations par label (ROC et matrices de confusion) pour comparer rapidement leur comportement.

Après avoir entraîné les deux approches, nous récupérons pour chaque arbre la probabilité prédite d’avoir un défaut sur chaque label (Collet, Houppier, Racine, Tronc). À partir de ces probabilités, nous calculons ensuite : (i) l’**AUC ROC** (capacité de classement globale) et (ii) l’**Average Precision (AP)** (plus informative quand les classes positives sont rares). Le tableau ci-dessous compare ces deux scores, label par label, pour **MultiOutput RF** et **Binary Relevance RF**.

Label	AUC_Multi	AP_Multi	AUC_BR	AP_BR
Collet	0.8961	0.5250	0.8961	0.5250
Houppier	0.9312	0.8314	0.9312	0.8314
Racine	0.9183	0.4387	0.9183	0.4387
Tronc	0.8679	0.6154	0.8679	0.6154

TABLE 7 – Comparaison MultiOutput RF vs Binary Relevance RF : AUC ROC et Average Precision (AP) par label.

Les deux méthodes donnent ici des résultats **identiques** (mêmes AUC et AP sur chaque label), ce qui est cohérent : dans les deux cas, on entraîne effectivement une Random Forest par label, avec une logique d'apprentissage très proche. On note que **Houppier** obtient les meilleurs scores (AUC et AP élevés), tandis que **Racine** et **Collet** restent plus difficiles, en particulier en AP, ce qui reflète le déséquilibre plus fort de ces labels (moins d'exemples positifs).

## 3 Défi 2 - Priorisation opérationnelle du parc d'arbres

### 3.1 Objectif

L'objectif est de mieux caractériser l'état du parc végétal de Grenoble à partir des variables disponibles (diagnostic, défauts, travaux préconisés, localisation), puis de proposer une priorisation actionnable : (1) identifier les **secteurs** les plus à risque/à traiter en priorité, et (2) fournir une liste d'**arbres prioritaires** par secteur, ainsi que des éléments de pilotage (types de travaux dominants, sur/sous-représentation dans l'urgence, visualisation spatiale).

### 3.2 Préparation des données et géolocalisation

Le notebook charge les données (`donnees-defi-egc.csv`) et effectue un nettoyage simple :

- Remplacement des valeurs "?" par NaN puis conversion de colonnes année en numérique (`ANNEEDEPLANTATION`, `ANNEEREALISATIONDIAGNOSTIC`, `ANNEETRAVAUXPRECONISESDIAG`).
- Construction d'un **GeoDataFrame** : chaque arbre devient un point (`coord_x`, `coord_y`) en EPSG:3945, puis conversion en WGS84 (EPSG:4326) pour la cartographie web.

Un indicateur multi-défauts est dérivé à partir des colonnes de localisation (`Collet`, `Houppier`, `Racine`, `Tronc`) :

$$\text{NB\_LOCALISATIONS\_DEFAULT} = \sum (\text{Collet}, \text{Houppier}, \text{Racine}, \text{Tronc}) ;$$

Enfin, un âge estimé est calculé :

$$\text{AGE\_ESTIME} = \text{ANNEEREALISATIONDIAGNOSTIC} - \text{ANNEEDEPLANTATION}$$

(en prenant 2025 comme année de référence si l'année de diagnostic est manquante, uniquement pour l'exercice).

### 3.3 Construction d'un score d'urgence (approche explicable)

Plutôt que d'entraîner un modèle de ML, le notebook met en place un **scoring interprétable** (explicable et facilement justifiable côté mairie). Les champs texte (`VIGUEUR`, `NOTEDIAGNOSTIC`, `PRIORITEDERENOUVELLEMENT`) sont normalisés (minuscule, trim) et utilisés via des règles simples.

Le score `SCORE_URGENCE` démarre à 0 et s'incrémente selon :

- +3 si `DEFAULT` = 1,
- +0 à +3 selon `NB_LOCALISATIONS_DEFAULT` (borné à 3),
- +2 si la vigueur contient "faible",
- +2 si la priorité de renouvellement contient "moins de 5" (ou <5),
- +2 si la note diagnostic contient "mauvais" / "danger" / "urgent",
- +1 si `AGE_ESTIME` ≥ 60 et encore +1 si ≥ 80.

Cette définition favorise une lecture opérationnelle : plus le score est élevé, plus l'arbre est supposé critique à court terme.

### 3.4 Agrégation et classement par secteur

Pour piloter à l'échelle de la ville, le notebook agrège les arbres par `ADR_SECTEUR` et calcule :

- `nb_arbres` (volume),
- `pct_default` (taux de défaut binaire `DEFAULT=1`),
- `pct_default_loc` (proportion avec au moins un défaut localisé),
- `score_moy` (moyenne du score),
- `age_moy`,
- `top_travaux` (travail le plus fréquent dans le secteur).

Les secteurs sont ensuite triés principalement par `score_moy` puis `pct_default` pour obtenir une **liste de secteurs prioritaires**. Le notebook met en évidence que certains secteurs (ex. 2 et 5) cumulent un score moyen élevé et une proportion importante d'arbres à problème, ce qui en fait des cibles naturelles pour un plan d'action.

### 3.5 Top arbres prioritaires par secteur

Au-delà du pilotage macro, le notebook génère une table **Top 10** par secteur : les arbres sont triés par `SCORE_URGENCE` décroissant, puis on conserve les 10 premiers de chaque `ADR_SECTEUR`, avec les informations utiles à l'action (`GENRE_BOTA`, `ESPECE`, âge, défauts, `TRAVAUXPRECONISESDIAG`, année des travaux, coordonnées). Cette sortie est directement exploitable pour préparer des tournées ou des interventions.

### 3.6 Validation qualitative du score via les travaux préconisés

Le notebook vérifie si le score « capte » des situations réputées lourdes/urgentes, en comparant `SCORE_URGENCE` aux types de travaux préconisés. Une table agrégée par `TRAVAUXPRECONISESDIAG` calcule notamment : effectif (`nb`), score moyen et taux de défaut. Les travaux typiquement critiques ressortent bien avec des scores très élevés, par exemple :

- **Abattage** : score moyen  $\approx 4.43$  ; % défaut  $\approx 93.2\%$ ,
- **Taille de mise en sécurité** : score moyen  $\approx 4.30$  ; % défaut  $\approx 90.7\%$ ,
- **Taille de bois mort** : score moyen  $\approx 2.34$  ; % défaut  $\approx 53.7\%$ .

Cela suggère que le score est cohérent : les interventions lourdes s'alignent avec des scores forts.

### 3.7 Urgence « critique » vs gestion de masse (Top 10%)

Le notebook isole ensuite les arbres appartenant au **Top 10%** des scores (`quantile 0.90`) et analyse la distribution des travaux. L'observation principale est la suivante :

- Les travaux **lourds** (mise en sécurité, abattage, contrôles spécialisés) sont fortement liés à l'urgence, mais restent **peu fréquents**.
- Le volume du Top 10% est surtout porté par des opérations **fréquentes** (formation, bois mort, contrôle), ce qui justifie une stratégie :
  - **Cas critique** : mise en sécurité / abattage / contrôles spécialisés,
  - **Gestion de masse** : formation / bois mort / contrôle.

### 3.8 Pilotage opérationnel : profils d'entretien par secteur

Pour comprendre « où on fait quoi », le notebook construit un tableau croisé (`pivot`) : `ADR_SECTEUR` en lignes, `TRAVAUXPRECONISESDIAG` en colonnes, et le nombre d'arbres en valeur. Il conserve les **8 travaux les plus fréquents** afin de dégager des profils d'entretien par secteur. C'est un livrable utile pour dimensionner les équipes/compétences et anticiper les besoins (matériel, planning).

### 3.9 Cartographie interactive (Folium)

Deux cartes interactives sont produites :

- Une carte affichant les **2000 arbres** les plus urgents (paramétrable), sous forme de `CircleMarker` avec popup (secteur, score, travaux).
- Une variante filtrée sur une condition d'urgence (ex. `score  $\geq 4$` ), pour ne visualiser que les points très critiques.

L'intérêt principal est de passer d'un classement tabulaire à une **lecture spatiale** : détection de grappes

---

d'arbres critiques, cohérence avec le découpage en secteurs, et support direct à la planification terrain.

### 3.10 Limites et améliorations possibles

- Le score repose sur des règles (poids fixes) : il est explicable, mais **arbitraire** et doit idéalement être calibré.
- Les champs texte sont exploités via des mots-clés : cela peut rater des formulations non prévues.

## 4 Conclusion

Dans ce projet, nous avons construit des modèles supervisés pour (i) prédire la présence d'un défaut (**DEFAULT**) et (ii) localiser ce défaut (**Collet**, **Houppier**, **Racine**, **Tronc**). Après un prétraitement standard (gestion des valeurs manquantes, sélection de variables, imputation et encodage via pipelines), les modèles d'ensemble **Random Forest** et **XGBoost** obtiennent les meilleures performances.

En uni-label, ils atteignent environ **0.88 d'accuracy** avec une **AUC proche de 0.93**, ce qui traduit une très bonne capacité de classement. En multi-label (One-vs-All), les résultats varient selon l'attribut : **Houppier** et **Tronc** sont les plus faciles, tandis que **Collet** et **Racine** restent plus difficiles en raison du déséquilibre des classes. Les méthodes alternatives (MultiOutputClassifier et Binary Relevance avec Random Forest) confirment ces tendances et donnent des résultats cohérents. Les figures détaillées (ROC/PR, matrices de confusion, arbre interprétable) sont fournies en annexe.

## A Annexe Défi 1

### Courbes ROC pour la tâche uni-label

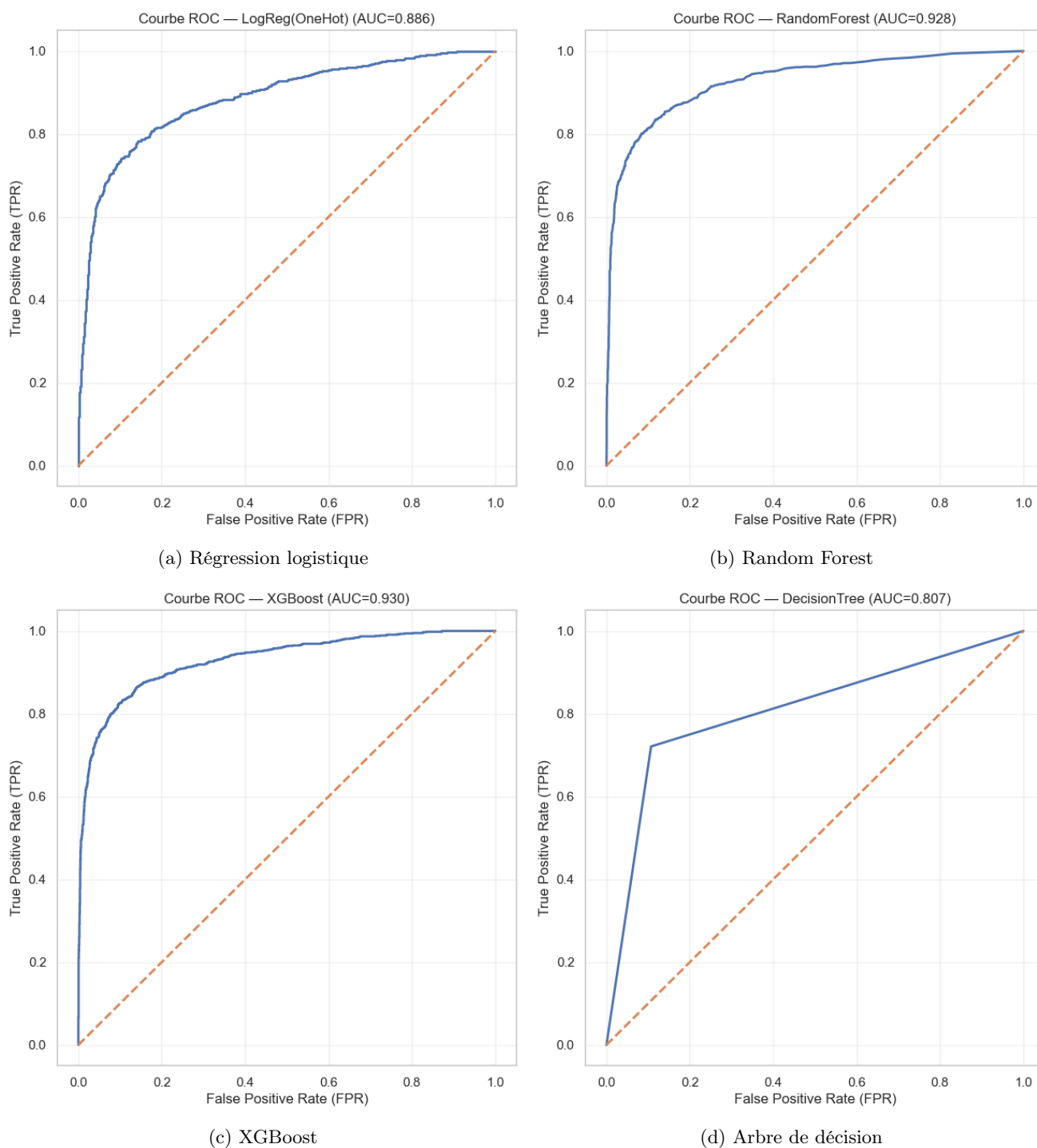


FIGURE 2 – Courbes ROC (uni-label) : une figure par modèle.

### Courbes ROC par attribut (grilles 2×2) pour la tâche multi-label

Pour chaque attribut (Collet, Houppier, Racine, Tronc), nous comparons les quatre modèles (régression logistique, random forest, XGBoost, arbre de décision) via des courbes ROC en grille 2×2. Plus la courbe est proche du coin supérieur gauche, meilleure est la séparation entre arbres avec et sans défaut.



### Courbes ROC — Collet

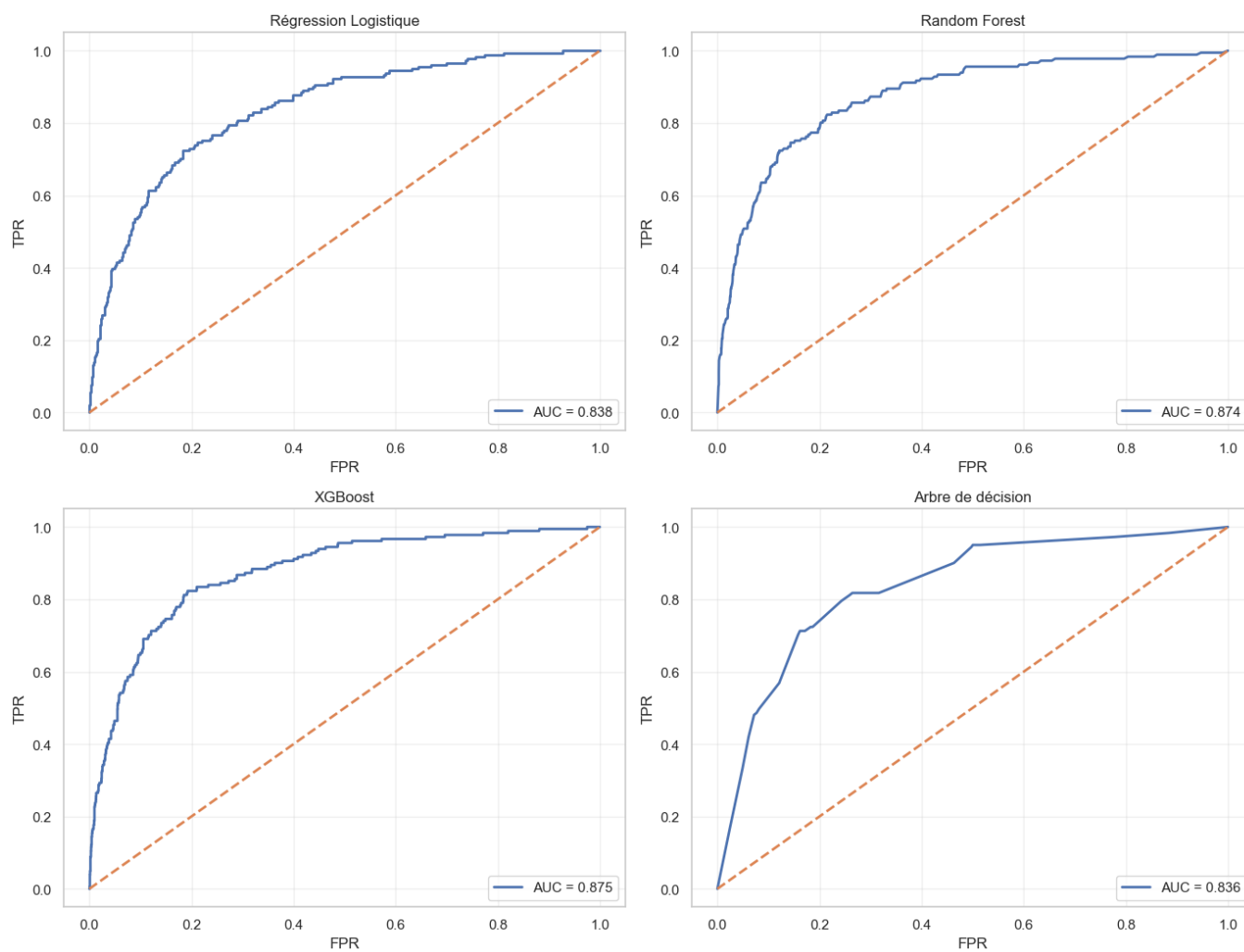


FIGURE 3 – ROC (2×2) pour Collet.

sur **Collet**, les meilleurs résultats sont obtenus par les modèles d'ensemble : **XGBoost** ( $AUC \approx 0.875$ ) et **Random Forest** ( $AUC \approx 0.874$ ), devant la régression logistique (0.838) et l'arbre (0.836). Les écarts restent modérés, mais les ensembles gardent un petit avantage.

### Courbes ROC — Houppier

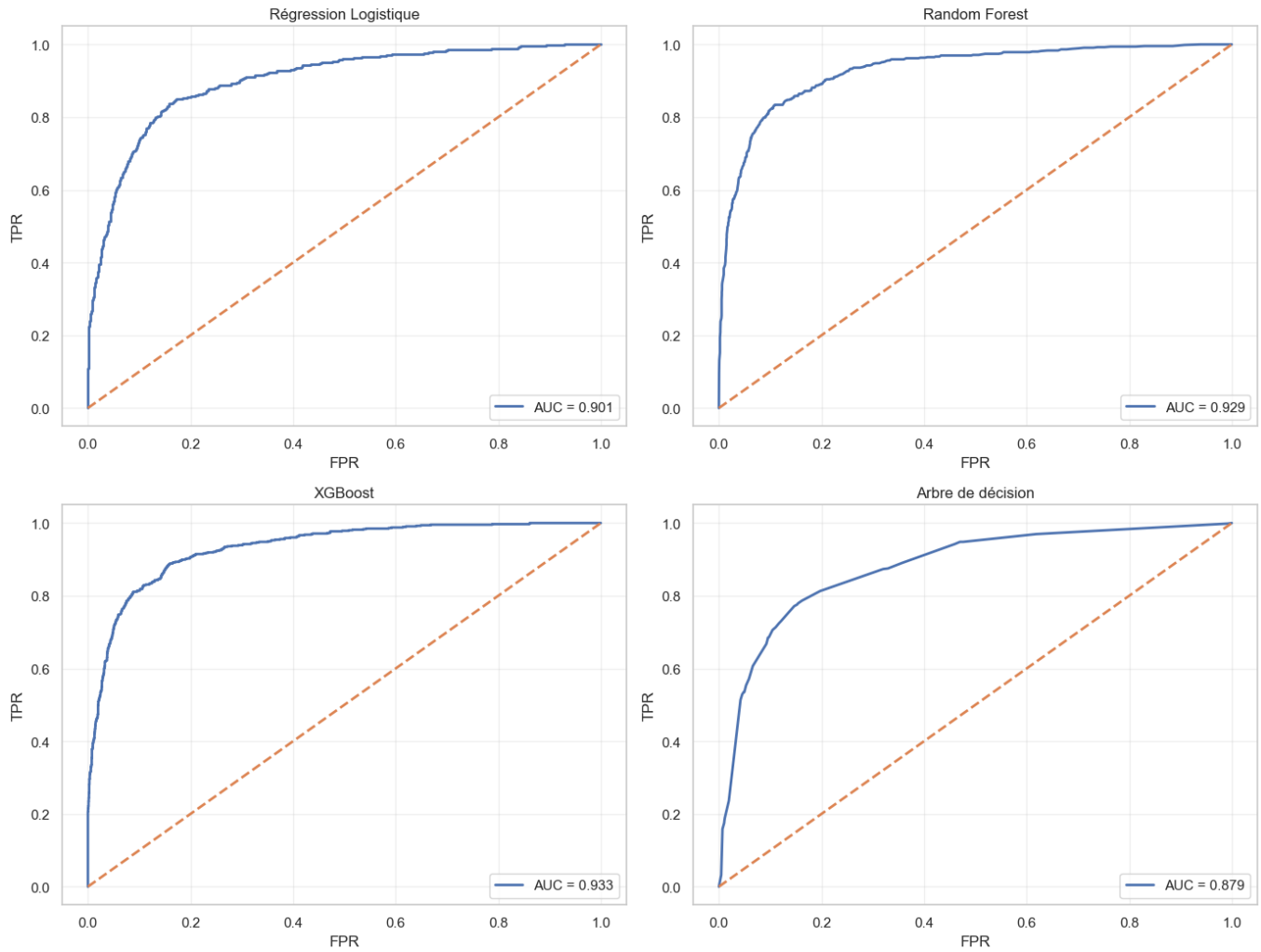


FIGURE 4 – ROC (2×2) pour Houppier.

Houppier est le label le plus « facile » ici : toutes les courbes sont nettement au-dessus de la diagonale. **XGBoost** ( $AUC \approx 0.933$ ) et **Random Forest** (0.929) dominent, et la régression logistique reste très compétitive (0.901). L'arbre de décision est un peu en retrait (0.879).

### Courbes ROC — Racine

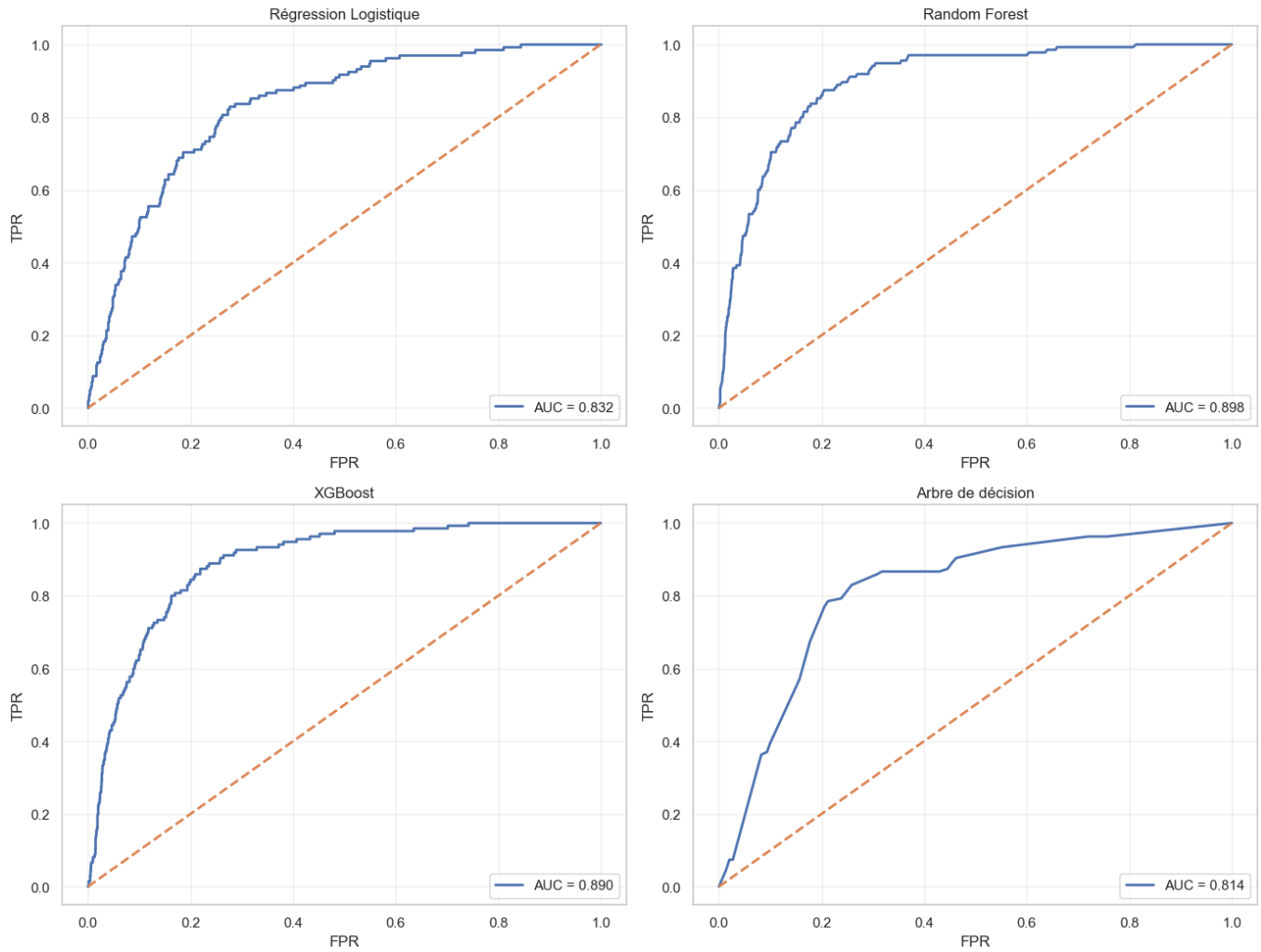


FIGURE 5 – ROC (2×2) pour **Racine**.

**Racine** montre l'avantage des ensembles de manière plus marquée : **Random Forest** ( $AUC \approx 0.898$ ) et **XGBoost** (0.890) dépassent clairement la régression logistique (0.832) et l'arbre (0.814). Cela confirme que les relations à apprendre sont probablement plus non-linéaires et/ou nécessitent des interactions entre variables.

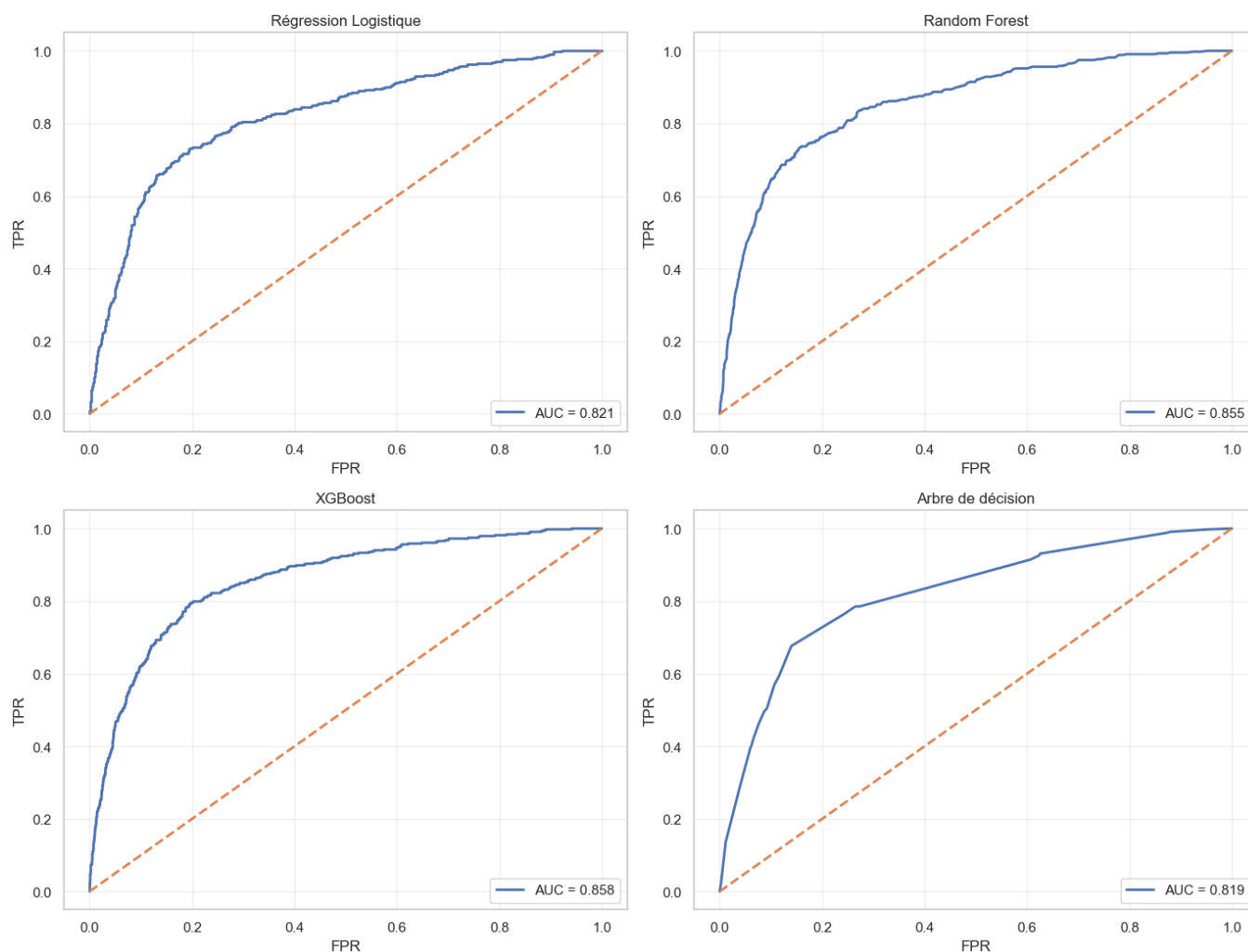


FIGURE 6 – ROC (2×2) pour Tronc.

sur Tronc, **XGBoost** ( $AUC \approx 0.858$ ) et **Random Forest** (0.855) sont légèrement meilleurs. La régression logistique (0.821) et l'arbre (0.819) restent corrects mais moins performants. On observe donc un gain régulier des modèles d'ensemble, sans écart massif.

### Interprétabilité : arbre de décision (profondeur limitée)

Afin de conserver un modèle interprétable, nous affichons un arbre de décision contraint en profondeur (`max_depth=3`, `min_samples_leaf=5`). Cela permet de visualiser les règles principales tout en évitant un arbre illisible.

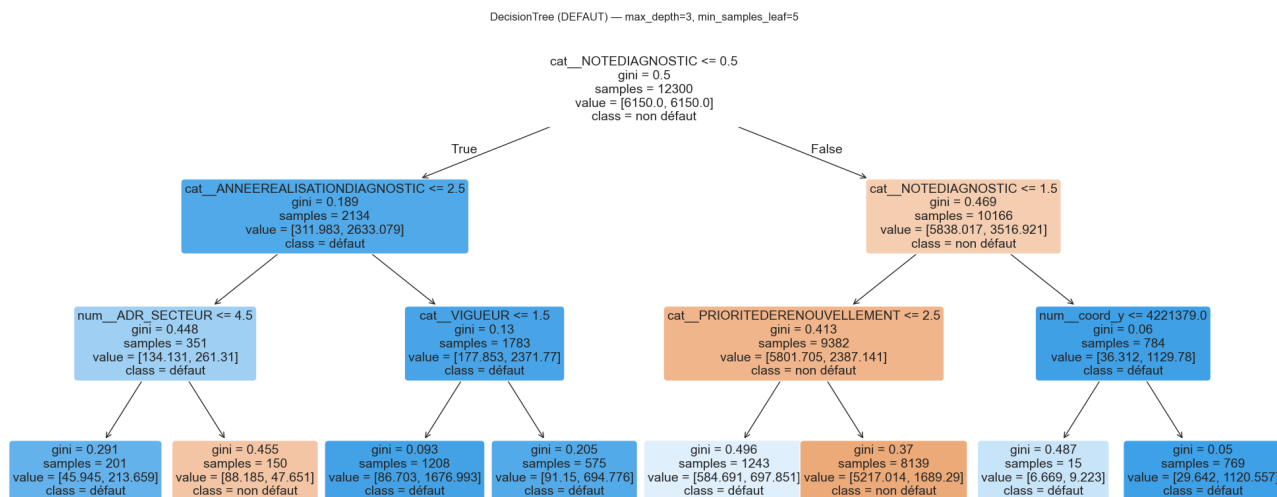


FIGURE 7 – Arbre de décision limité (uni-label).

Une autre méthode de classification multi-label (Courbes ROC et matrices de confusions)

Courbes ROC — MultiOutputClassifier (RandomForest) — 2x2

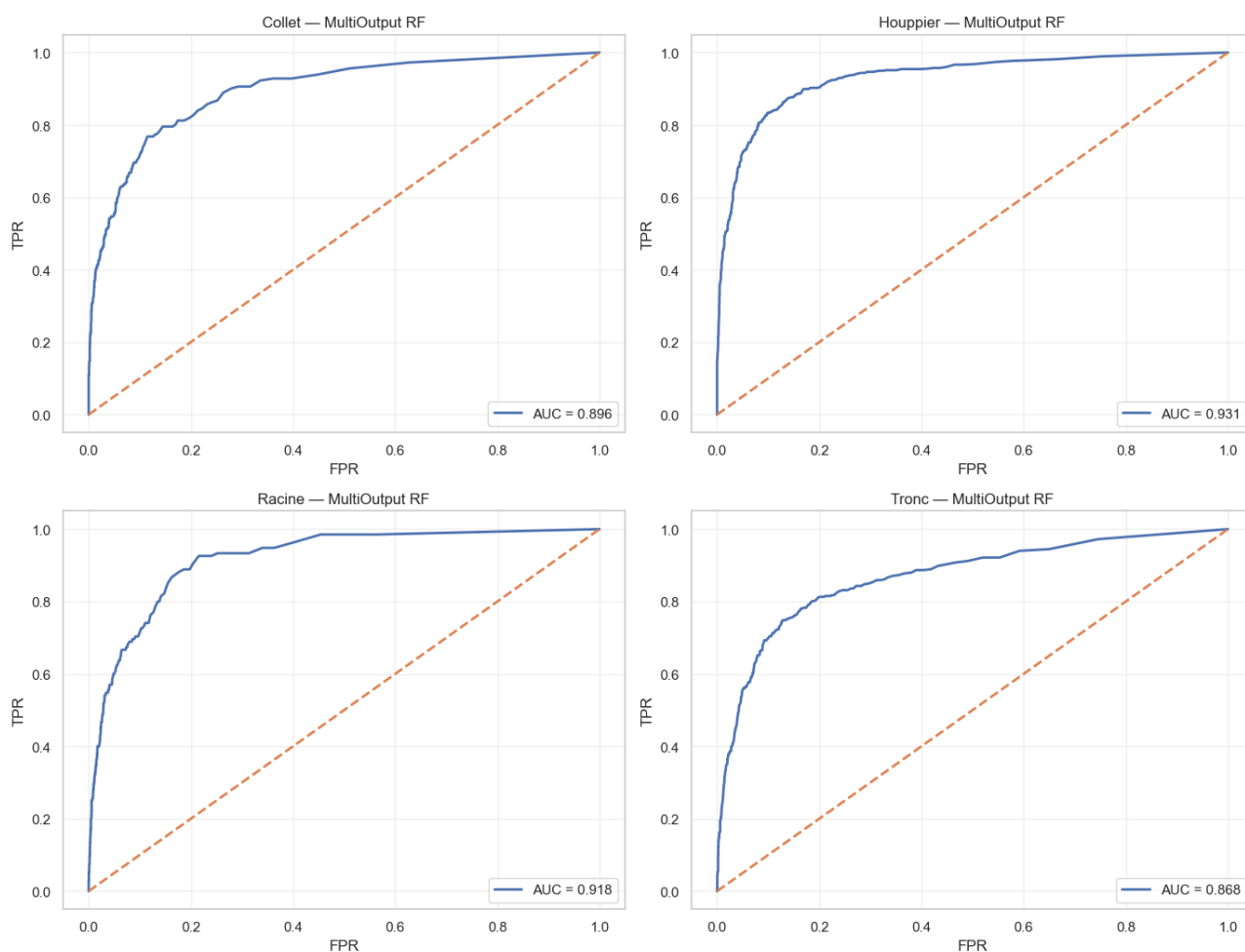


FIGURE 8 – Courbes ROC (2x2) par label — MultiOutputClassifier (Random Forest).

on observe une séparation globalement correcte, généralement meilleure sur les labels les plus fréquents (Houppier, Tronc) et plus difficile sur les labels rares (Collet, Racine).

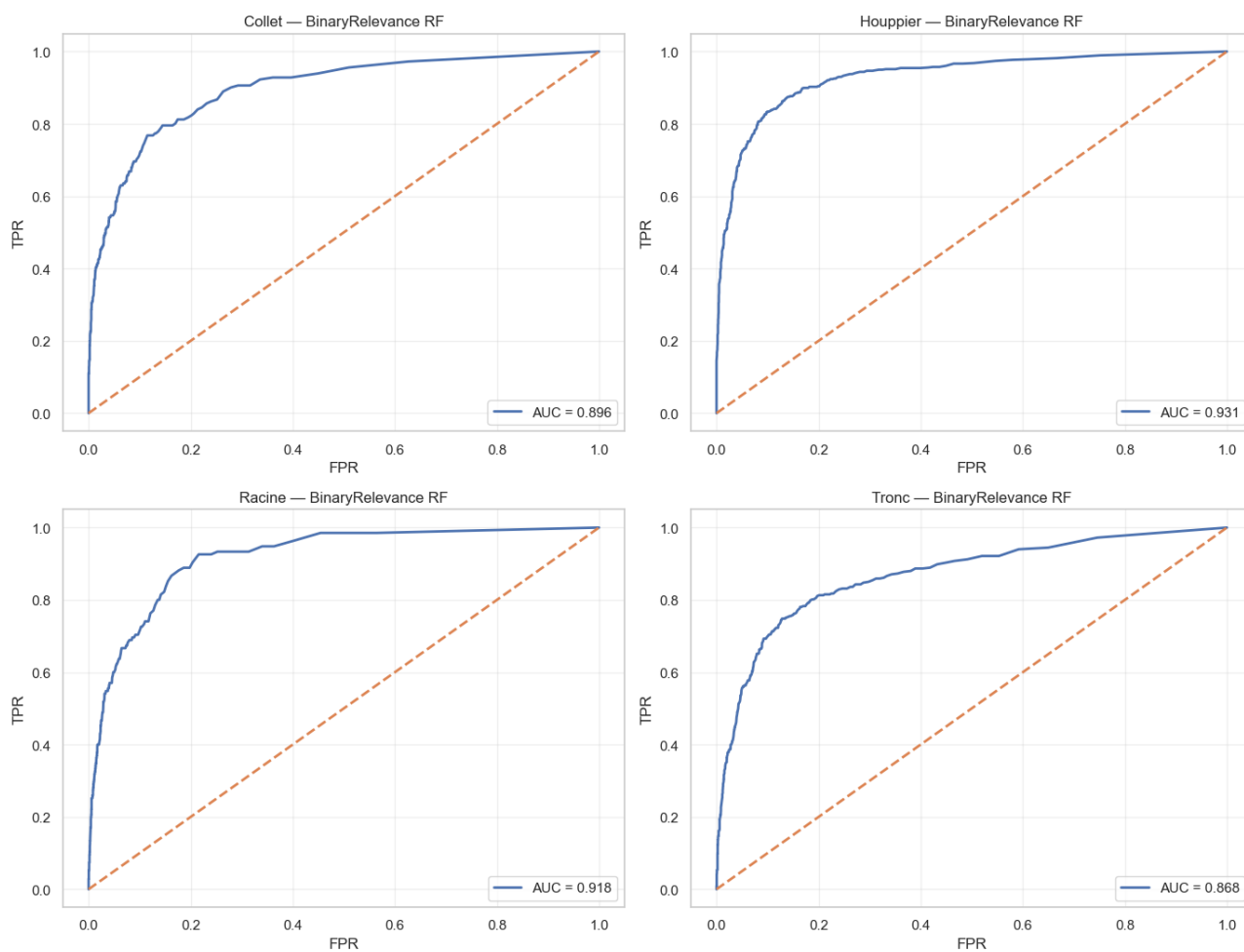


FIGURE 9 – Courbes ROC (2×2) par label — Binary Relevance (Random Forest).

les tendances sont proches de MultiOutput (même famille de modèle). Les différences observées proviennent surtout de la manière dont la méthode gère l'apprentissage label par label et de la sensibilité aux classes minoritaires.

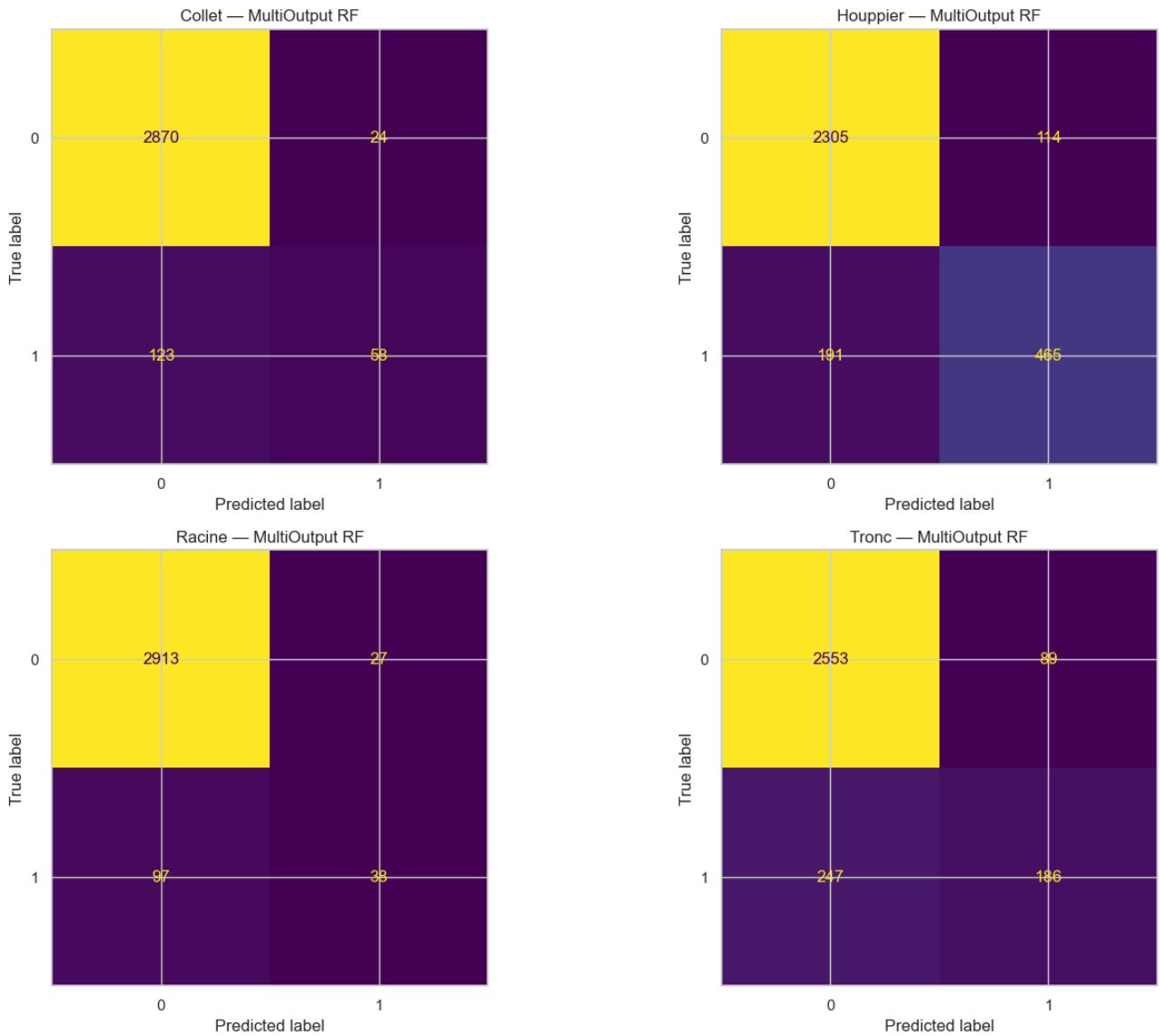


FIGURE 10 – Matrices de confusion (2×2) par label — MultiOutputClassifier (Random Forest).

pour les labels rares, on constate souvent davantage de faux positifs ou de faux négatifs. Cela illustre le compromis classique précision/rappel : améliorer le rappel sur des classes rares peut augmenter les fausses alertes.

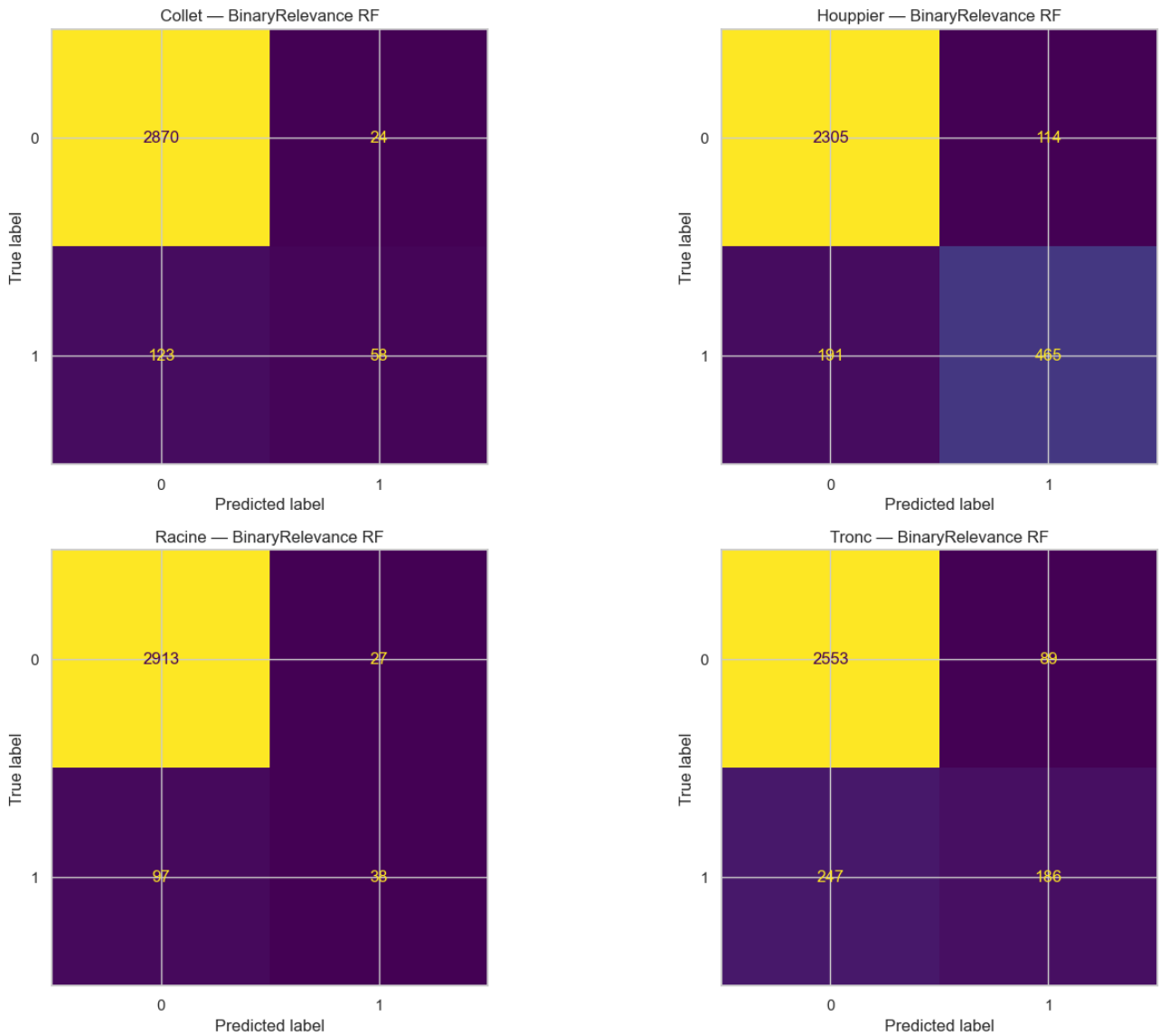


FIGURE 11 – Matrices de confusion (2×2) par label — Binary Relevance (Random Forest).

les matrices confirment les difficultés sur **Collet** et **Racine**. Les erreurs sont moins marquées sur **Houppier** et **Tronc**, où le nombre d'exemples positifs est plus élevé.



# B Annexe Défi 2

[52]:

	nb_arbres	pct_defaut	pct_defaut_loc	score_moy	age_moy	top_travaux
ADR_SECTEUR						
2	3113	43.559268	0.435593	1.954385	9.597816	Taille formation et mise au gabarit
5	2943	35.507985	0.355080	1.524295	9.724771	Taille formation et mise au gabarit
4	2162	31.082331	0.310823	1.340426	9.571693	Taille formation et mise au gabarit
6	2961	30.091185	0.300912	1.316447	9.488011	Taille formation et mise au gabarit
3	2518	27.958697	0.279587	1.261716	8.876886	Taille formation et mise au gabarit
1	1678	19.845054	0.198451	0.843862	7.523838	Controle

C'est une comparaison des secteurs sur des indicateurs de risque et d'entretien, pour cibler les zones prioritaires. Ici on remarque que les secteur 2 et 5 sont les deux prioritaires avec une combinaison de nombre d'arbre ayant des défauts et de score moyen élevé.

FIGURE 12 – Agrégation d'arbre avec défaut par secteur

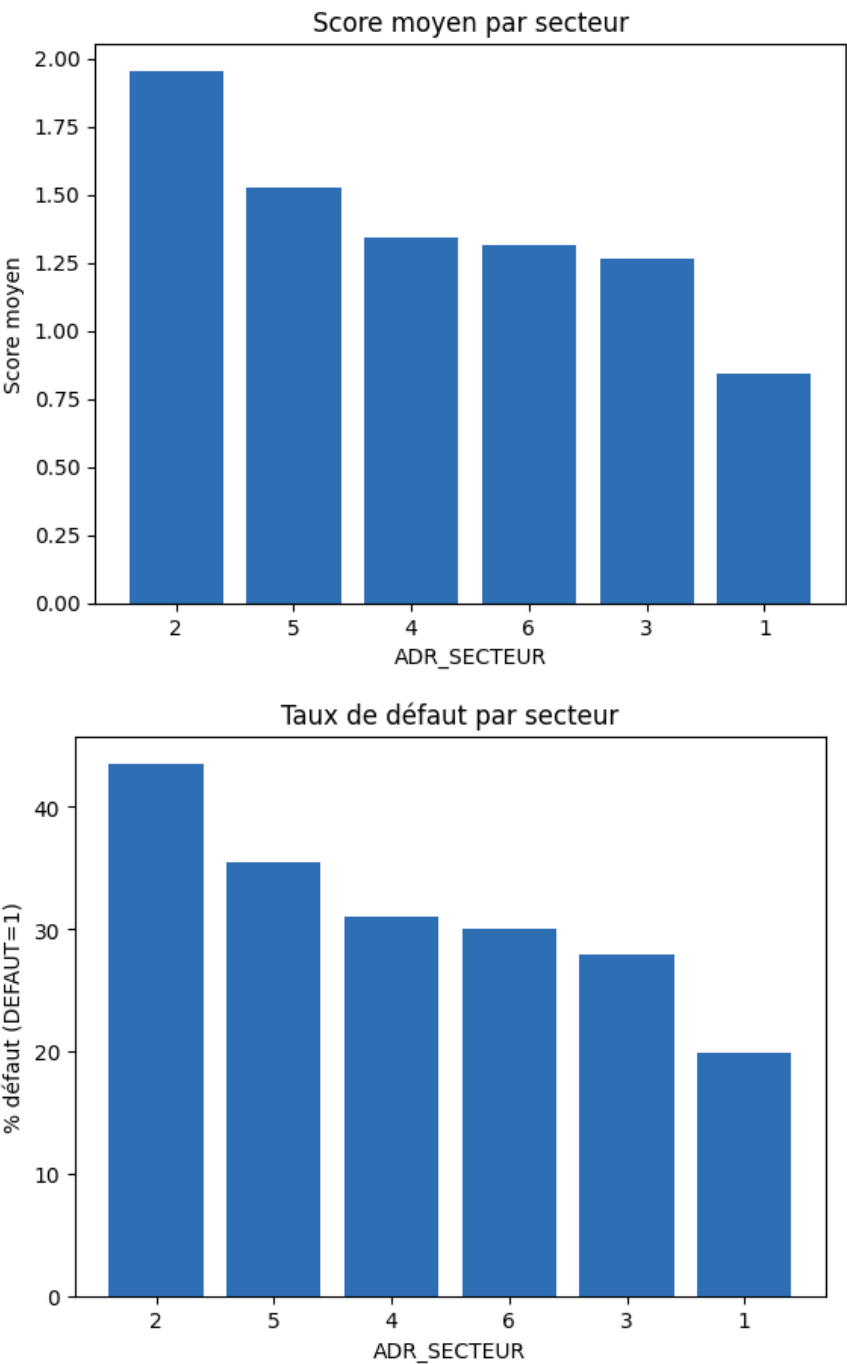


FIGURE 13 – Affichage des score élevé par secteur

[57]:

	ADR_SECTEUR	GENRE_BOTA	ESPECE	AGE_ESTIME	DEFAULT	NB_LOCALISATIONS_DEFAULT	TRAVAUXPRECONISESDIAG	ANNEETRAVAUXPRECONISESDIAG	SCC
12374	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
7189	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
8067	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
8993	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
12439	2	Platanus	acerifolia	11.0	1	3	Taille formation et mise au gabarit	2016.0	
9350	2	Populus	nigra	10.0	1	3	Taille de bois mort	2014.0	
421	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
12285	2	Platanus	acerifolia	11.0	1	3	Taille mise en sécurité	2016.0	
4551	2	Platanus	acerifolia	11.0	1	4	Contrôle	2017.0	
12369	2	Platanus	acerifolia	11.0	1	3	Contrôle	2017.0	

En plus de la priorisation par secteur précédente, on stock les 1à arbres prioritaire par secteurs.

FIGURE 14 – Top 10 arbres prioritaire par secteur (secteur 2 ici)

[61]:

	nb	score_moy	pct_default
TRAVAUXPRECONISESDIAG			
Taille formation et mise au gabarit	4616	1.341854	31.759099
Contrôle	2493	1.270758	27.396711
Taille de bois mort	1660	2.343976	53.674699
Taille cohabitation	564	1.168440	27.127660
Taille mise en sécurité	418	4.301435	90.669856
Taille déclaircie	365	0.534247	12.876712
Abattage	220	4.427273	93.181818
Aménagement pieds darbres	162	1.734568	38.271605
Contrôle tuteur attache ou protection	144	0.916667	22.222222
Taille de prolongement	108	1.314815	30.555556
Taille rideau	59	2.152542	52.542373
Arrosage	19	0.263158	5.263158
Emondage	8	2.000000	50.000000
Contrôle résistographe	7	4.428571	100.000000
Haubannage	7	4.285714	100.000000

On remarque qu'effectivement les travaux d'abattage, de taille de mise en sécurité ainsi que d'haubannage sont effectués sur les arbres ayant des scores élevés.

On remarque aussi une valeur similaire pour le contrôle résistographes.

FIGURE 15 – Travaux préconisés selon l'état de l'arbre

[62]:

TRAVAUXPRECONISESDIAG	
Taille formation et mise au gabarit	36.6
Taille de bois mort	22.3
Contrôle	17.1
Taille mise en sécurité	9.5
Abattage	5.1
Taille cohabitation	3.8
Aménagement pieds darbres	1.5
Taille déclaircie	1.2
Taille de prolongement	0.8
Contrôle tuteur attache ou protection	0.8
Name: proportion, dtype: float64	

Ici on se rend compte de quelque chose d'important comparé aux résultats d'avant :

Lorsque des travaux "lourds" sont prescrit, c'est quasiment lié à une situation urgente. Mais leur occurences est assez basse.

Lorsque l'on regarde le top 10% des arbres les plus urgents, le volume d'arbre dans ce top 10% est principalement porté par des opérations plus fréquentes tels que taille de formation, bois mort et contrôle.

Ainsi cela peut justifier une stratégie où :

- **Cas critique** : mise en sécurité / abattage / contrôles spécialisés
- **Gestion de masse** : formation / bois mort / contrôle

FIGURE 16 – Sur/sous représentation de travaux dans le top 10% d'arbre prioritaire

[63]:

TRAVAUXPRECONISESDIAG	Taille formation et mise au gabarit	Controle	Taille de bois mort	Taille cohabitation	Taille mise en sécurité	Taille déclaircie	Abattage	Aménagement pieds darbres
ADR_SECTEUR								
1	447	569	95	105	26	47	8	32
2	960	344	579	61	195	28	45	60
3	615	453	182	41	54	4	26	21
4	374	200	297	93	28	30	28	8
5	905	347	349	187	97	172	63	7
6	1315	580	158	77	18	84	50	34

FIGURE 17 – Occurrence de travaux selon les secteurs

[67]:

	Global_%	Top10%_%
TRAVAUXPRECONISESDIAG		
Taille formation et mise au gabarit	42.543779	36.640840
Controle	22.976959	17.070732
Taille de bois mort	15.299539	22.269433
Taille cohabitation	5.198157	3.824044
Taille mise en sécurité	3.852535	9.472632
Taille déclaircie	3.364055	1.174706
Abattage	2.027650	5.123719
Aménagement pieds darbres	1.493088	1.549613
Controle tuteur attache ou protection	1.327189	0.799800
Taille de prolongement	0.995392	0.824794

La taille de mise en sécurité (+5.56%), l'abattage (3.1%), la taille de bois mort (+6.96%) sont plus présent sur les cas urgents

La taille de formation (+5.94%), le contole (+5.9%) sont plus présent dans les as globaux

FIGURE 18 – Comparatif des travaux préconisé dans les cas global vs top 10% prioritaire