08 Janvier 2024

M2 Ingénieries Mathématique pour la Science des Données

Projet Fouille de Données et Extraction de Connaissances

Encadré par :
Sabeur Aridhi
Laura Zanella Calzada
Élaboré par :
CHERGUI-ABBOUDA
Nouhaila
MARNISSI Yassine
ZAHOUANI Zineb

2023 — 2024

Table des matières

Table des figures

0.1 Introduction

Big Datext, entreprise Grenobloise spécialisée dans l'analyse prédictive, et la mairie de Grenoble se sont associées pour la mise en place et la diffusion d'une base de données pour un défi associé à une conférence nationale. Big Datext et les services de la Ville ont axé le défi sur les données relatives aux espaces verts. Le défi consiste en une prédiction visant à déterminer, à partir des données disponibles, si l'arbre a ou non un défaut et dans l'affirmative lequel, sachant qu'un arbre peut présenter un défaut à différents endroits : racine, tronc, collet, houppier.
Dans ce projet , nous allons travailler sur deux tâches : la première consiste à prédire si un arbre a un défaut ou pas , ce qui correspond à un problème de classification uni-label, la deuxième vise à prédire au mieux les localisations des défauts (problème de classification multilabel).

0.2 Le sujet et les prétraitements

0.2.1 Données disponibles

Notre jeu de données comprend 15 376 individus répartis sur 34 attributs, regroupés en différentes catégories :

- Attributs décrivant l'arbre :
  - Code
  - ESPECE
  - DIAMETREARBREAUNMETRE
  - etc ...
- Attributs décrivant l'emplacement de l'arbre :
  - ADR_SECTEUR
  - CODE_PARENT_DESC
  - etc ...
- Attributs fournissant des informations sur les diagnostics :
  - ANNEEREALISATIONDIAGNOSTIC
  - TRAVAUXPRECONISESDIAG
  - etc ...
- Attributs indiquant la présence ou non d'un défaut ainsi que sa localisation :
  - Défaut
  - Collet
  - Houppier
  - Racine
  - Tronc

0.2.2 Prétraitements

En observant les données, on voit que, bien qu'aucune valeur null ne soit trouvée dans l'ensemble, des valeurs ' ?' apparaissent dans les différentes colonnes là où il manque des informations. En cherchant quelle proportion de valeurs inconnues possèdent chaque colonne, on obtient :

RAISONDEPLANTATION	15145
TYPEIMPLANTATIONPLU	15014
INTITULEPROTECTIONPLU	15014
TRAITEMENTCHENILLES	14287
VARIETE	13212
REMARQUES	11176
TRAVAUXPRECONISESDIAG	4525
ANNEETRAVAUXPRECONISESDIAG	4511
ESPECE	1018
PRIORITEDERENOUVELLEMENT	127
DIAMETREARBREAUNMETRE	67
STADEDEDEVELOPPEMENT	51
NOTEDIAGNOSTIC	40
STADEDEVELOPPEMENTDIAG	13
VIGUEUR	11
ANNEEREALISATIONDIAGNOSTIC	8
FREQUENTATIONCIBLE	1
TROTTOIR	0
coord_y	0
coord_x	0
ADR_SECTEUR	0
SOUS_CATEGORIE_DESC	0
ANNEEDEPLANTATION	0
GENRE_BOTA	0
CODE_PARENT_DESC	0
CODE_PARENT	0
DEFAUT	0
dtype: int64	

FIGURE 1 – Total des valeurs manquantes pour chaque colonne

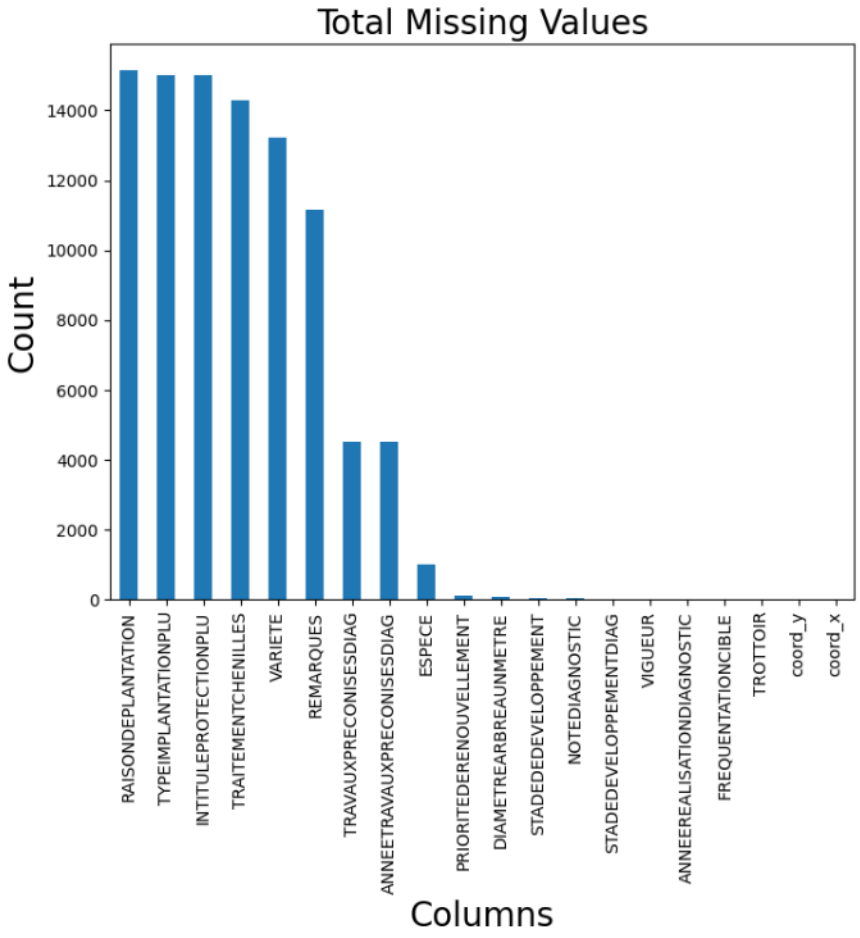

FIGURE 2 – Graphique du total des pourcentages de valeurs manquantes

Nous avons initié le processus en éliminant les colonnes :

— CODEIDENTIFIANTPLU
— SOUS_CATEGORIE
— CODE_PARENT
— STADEDEVELOPPEMENTDIAG

Ensuite, pour les colonnes restantes, une fois le seuil défini, nous avons supprimé celles qui présentaient plus de 90% d'informations manquantes.

Pour ces colonnes restantes contenant des valeurs manquantes, nous avons appliqué une imputation par mode, remplaçant ainsi les valeurs manquantes par la valeur la plus fréquemment observée dans chaque colonne respective. De plus, afin de mieux représenter ces valeurs manquantes, nous avons créé une catégorie spécifique que nous avons appelée `"aucun"`.

Cette approche globale vise à gérer de manière efficace les données manquantes tout en préservant l'intégrité et la signification des informations restantes dans le jeu de données.

Pour la colonne "remarque", nous avons mis en place un processus de prétraitement basé sur le traitement du langage naturel (NLP). Ce processus commence par l'utilisation d'outils NLP pour convertir les textes en minuscules, éliminer les mots vides (stopwords) et appliquer la racinisation des mots.

L'objectif est d'optimiser le texte pour une analyse ultérieure en éliminant le bruit inutile et en normalisant la représentation des mots. Une fois ce prétraitement effectué, nous avons une colonne "remarque" prête à être utilisée de manière plus efficace dans diverses tâches, telles que l'analyse de similarité ou le regroupement de remarques similaires. Cette approche permet d'améliorer la qualité et la cohérence de l'information contenue dans la colonne "remarque" de notre ensemble de données, facilitant ainsi son interprétation et son utilisation dans des analyses ultérieures.

	REMARQUES	classe_Remarques
0	Aucun	1
1	Aucun	1
2	Aucun	1
3	Enormément de grosses branches cassées	0
4	Aucun	1
...	...	...
15370	Aucun	1
15371	Aucun	1
15372	Aucun	1
15373	Réseau aérien passant audessus de larbre	0
15374	Aucun	1

[15375 rows x 2 columns]

Figure 3 – Résultats NLP sur la colonne Remarque

Après avoir prétraité les remarques en utilisant des techniques de traitement du langage naturel, nous avons normalisé les données numériques pour garantir une échelle uniforme. De plus, les données catégorielles dans les autres colonnes ont été encodées en utilisant une méthode de codage des labels, assignant des identifiants numériques uniques à chaque catégorie. Ces étapes de normalisation et d'encodage permettent d'obtenir un ensemble de données prêt à être utilisé dans des analyses et des modèles, assurant une représentation cohérente et adaptée aux algorithmes d'apprentissage automatique.

0.3 Prédiction de défauts

0.3.1 Classification uni-label

Après avoir effectué la préparation des données, nous avons utilisé les differents algorithmes de classifications supervisée et obtenu les résultats suivants :

Modèle	Accuracy	Precision	Recall	F1-score
KNN	77.82%	82%	87%	84%
Forêt aléatoire	86.34%	89%	91%	90%
Régression logistique	76.06%	78%	90%	84%
SVM	67.77%	68%	100%	81%
Naive Bayes	80.48%	80%	94%	87%
AdaBoost	83.80%	85%	93%	89%
Gradient Boosting	85.43%	86%	94%	90%

Table 1 – Résultats des différents modèles

Au regard des résultats ci-dessus, on remarque que les meilleurs modèles sont ceux obtenus par les algorithmes de forêt aléatoire, Adaboost et GradientBoosting.

0.3.2 Classification Multi-label

Au cours de notre étude, nous avons entrepris la tâche de prédire les défauts potentiels d'arbres en utilisant une approche de classification multi-label. Nous avons extrait et prétraité les données du fichier `.csv`, en nous concentrant sur les colonnes liées aux défauts au niveau du collet, du houppier, de la racine et du tronc.

Pour résoudre ce défi, nous avons choisi de construire des modèles spécifiques pour chaque type de défaut. Dans cette perspective, nous avons exploré trois approches de classification : `RandomForestClassifier`, `AdaBoostClassifier`, et `BaggingClassifier` avec un `DecisionTreeClassifier` comme modèle de base.

Jetons un coup d'œil au tableau global récapitulant les performances de ces modèles pour chaque classe de défaut : En analysant ces résultats, il est clair que `RandomForestClassifier` se

Méthode	Precision	Recall	F1-Score	Support
RandomForestClassifier	94.5%	88.7%	95.5%	89.2%
AdaBoostClassifier	93.5%	86.2%	94.9%	86.9%
BaggingClassifier	94.2%	87.5%	95.0%	89.1%

Table 2 – Résultats des différents modèles pour la classification multi-label

distingue avec des performances globalement solides, atteignant des scores équilibrés de précision, rappel, et F1-Score pour toutes les classes.

`AdaBoostClassifier` montre des performances inférieures, en particulier en ce qui concerne le rappel.

`BaggingClassifier` se situe entre les deux, offrant des performances intermédiaires.

En conclusion, notre approche de création de classifieurs distincts pour chaque classe de défaut a permis d'obtenir des résultats encourageants, mais il reste des possibilités d'amélioration. Nous pourrions explorer davantage les hyperparamètres, essayer d'autres techniques d'ensemble, et effectuer une validation croisée pour renforcer la robustesse de nos modèles.

0.4 Conclusion

En conclusion, notre projet de fouille de données pour prédire les défauts d'arbres a été réussi. Les modèles, notamment RandomForestClassifier, AdaBoostClassifier et BaggingClassifier, ont montré des performances encourageantes.

Malgré ces résultats positifs, des améliorations potentielles subsistent, soulignant l'importance continue de l'exploration des hyperparamètres. Ce projet offre des perspectives pour des applications futures dans la gestion durable des espaces verts urbains.

0.5 Annexe

Code source:cliquez ici