Université de Lorraine

Faculté des Sciences et Technologies

Département de Mathématiques

Master II en Mathématiques Appliquées

Parcours : Ingénierie Mathématique pour la Science des Données

Projet FDEC

La fouille de données au service du développement durable

Réalisé par :

KHATIR ISSLAM
MORSLI MARWA

02 février 2022

Table des matières

1. Résumé

Ce projet propose un défi dont le contexte est la gestion des espaces verts pour la ville de Grenoble, et notamment des arbres qui y sont présents. L'objectif est de proposer un modèle basé sur des données fournies qui permettrait de prédire au mieux les arbres avec défauts, ainsi que la localisation potentielle du défaut. Plusieurs algorithmes de classification supervisée ont été expérimentés pour répondre à la tâche numéro 1 du défi. Les performances ont été évaluées par validation croisée. Cela nous a permis de sélectionner les meilleurs classifieurs uni-label et multi-label. Nous avons également exploré la tâche numéro 2 du défi. D'une part, des règles d'association ont été recherchées. D'autre part, le jeu de données a été enrichi avec des connaissances telles que des données climatiques (pluviométrie, température, vent) ou des données taxonomiques dans le domaine de la botanique (famille, ordre, super-ordre). En outre, des données géographiques et cartographiques sont exploitées dans un outil de visualisation d'une partie des données sur les arbres.

2. Introduction

Les deux tâches du défi vert de Grenoble ont été abordées. La première tâche de prédiction, visant à prédire si les arbres présentent des défauts ou non, est un problème de classification supervisée. Dans un premier temps, les données ont été analysées afin de s'assurer d'un corpus d'apprentissage le plus exploitable possible. Dans un second temps, quelques algorithmes de classification sélectionnés ont été testés et évalués sur le jeu de données. La seconde tâche est quant à elle axée sur une meilleure connaissance de l'état ainsi que de l'évolution du "parc végétal" de Grenoble.

3. Préparation des données

3.1 Analyse exploratoire des données

L'enjeu du travail présenté dans ce rapport est, pour la ville de Grenoble, de prédire l'apparition des défauts parmi les arbres du parc. Une partie du rapport expliquera notre approche sur les 2 types de prédiction proposés dans la tâche 1 (unilabel et multilabel). Une seconde partie proposera une connaissance de l'état du « parc végétal » de Grenoble et comprendre son évolution et fournir des préconisations pour faciliter son entretien. Avant cela, nous étudierons comment nous avons préparé les données afin de les intégrer à nos modèles.

Les données fournies constituent un corpus composé de 15 375 instances d'arbres, décrites par 34 attributs. Certains attributs décrivent l'arbre *(Code, DiamètreArbreÀUnMètre, Année DePlantation, Espèce, Genre_Bota…*), son emplacement *(Adr_Secteur, Trottoir, Fréquenta tionCible,* coordonnées géographiques sur un plan…*)*, des informations établies à l'occasion de diagnostics *(AnnéeRéalisationDiagnostic, NoteDiagnostic, AnnéeTravauxPréconisésDiag, Remarques…*), la présence et la (ou les) localisation(s) d'un défaut *(Défaut, Collet, Houppier,Racine, Tronc*). Les derniers attributs constituent des informations de classe qu'il s'agit de prédire dans le cadre de la première tâche du défi.

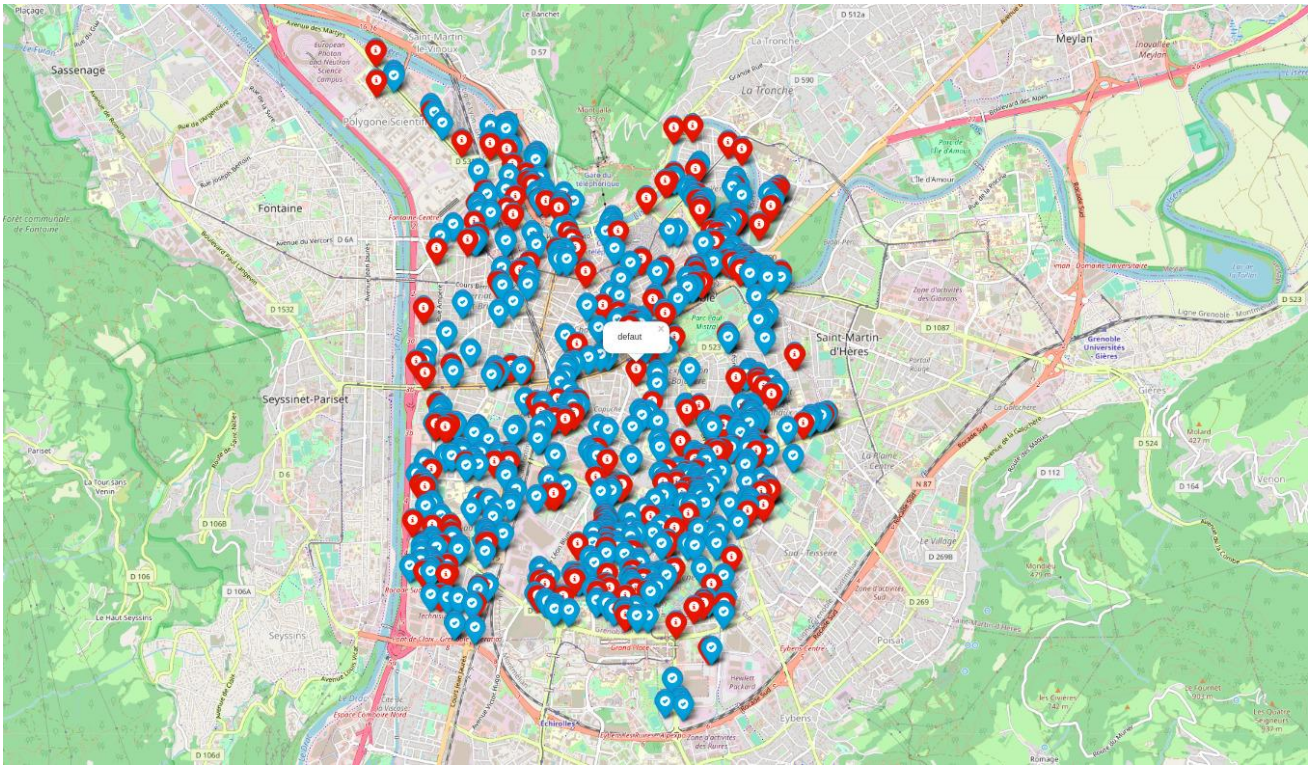

FIG. 1 – Positionnement des arbres sans défaut (en bleu) et avec défauts (en rouge)

3.2 Types des données existants

3.2.1 Données numériques

Un certain nombre de champs (7 au total) possède des valeurs numériques *(ADR_SECTEUR,ANNEEDEPLANTATION,ANNEEREALISATIONDIAGNOSTIC, ANNEETRAVAUXPRECONISESDIAG, IDENTIFIANTPLU, coord_x, coord_y)* pour lesquels aucun formatage n'est nécessaire.

3.2.2 Données catégorielles

Un certain nombre de champs (22 au total) posséde des valeurs catégorielles *(Code, Code_parent, Code_parent_DESC, DIAMETREARBREUUNMETRE , ESPACE, FREQUENTATIONCIBLE , GENRE_BOTA, INTITULEPROTECTIONPLU , NOTEDIAGNOSTIC,PRIORITEDERENOUVELLEMENT, RAISONDEPLANTATION, SOUS_CATEGORIE,SOUS_CATEGORIE_DESC,STADEDEDEVELOPPEMENT,STADEDEVELOPPEMENTDIAG,TRAITEMENTCHENILLES,TRAVAUXPRECONISESDIAG, TROTTOIR, TYPEIMPLANTATIONPLU, VARIETE, VIGUEUR)*

3.2.3 Données textuelles

Seul le champ REMARQUES a retenu notre attention en tant que donnée textuelle.

Les variables *(DEFAUT, Collet,Houppier, Racine, Tronc)* sont considérés comme qualitatives.

Les variables *(Code,IDENTIFIANTPLU,INTITULEPROTECTIONPLU , TYPEIMPLANTATIONPLU)* sont des identifiants associés à la base donc sont considérés des variables factorielles.

3 .3 Problème des valeurs manquantes

On remarque que les variables qui ont un haut pourcentage (plus que 50%) de valeurs manquantes sont : *RAISONDEPLANTATION* (98.5%), *IDENTIFIANTPLU* (97.7%), *INTITULEPROTECTIONPLU* (97.7%), *TYPEIMPLANTATIONPLU* (97.7%), *TRAITEMENTCHENILLES* (92.9%), *VARIETE* (85.9%), *REMARQUES* (72.7%).

On remarque que les variables qui ont un pourcentage de valeurs manquantes ne dépassant pas les 50% des observations : *TRAVAUXPRECONISESDIAG* (29.4%), *ANNEETRAVAUXPRECONISESDIAG* (29.3%), *ESPECE* (6.6%), *PRIORITEDERENOUVELLEMENT* (0.8%), *DIAMETREARBREAUNMETRE* (0.4%), *STADEDEDEVELOPPEMENT* (0.3%), *NOTEDIAGNOSTIC* (0.3%), *STADEDEVELOPPEMENTDIAG* (0.1%), *ANNEEREALISATIONDIAGNOSTIC* (0.1%), *VIGUEUR* (0.1%).

Avant de traiter l'objectif du premier Défi 1 qu'est la classification supervisée, on essayera de résoudre ce problème de valeurs manquantes (comme vu en cours).

3.4 Corrélation et indépendance

3.4.1 Corrélation entre *DEFAUT* et les autres variables

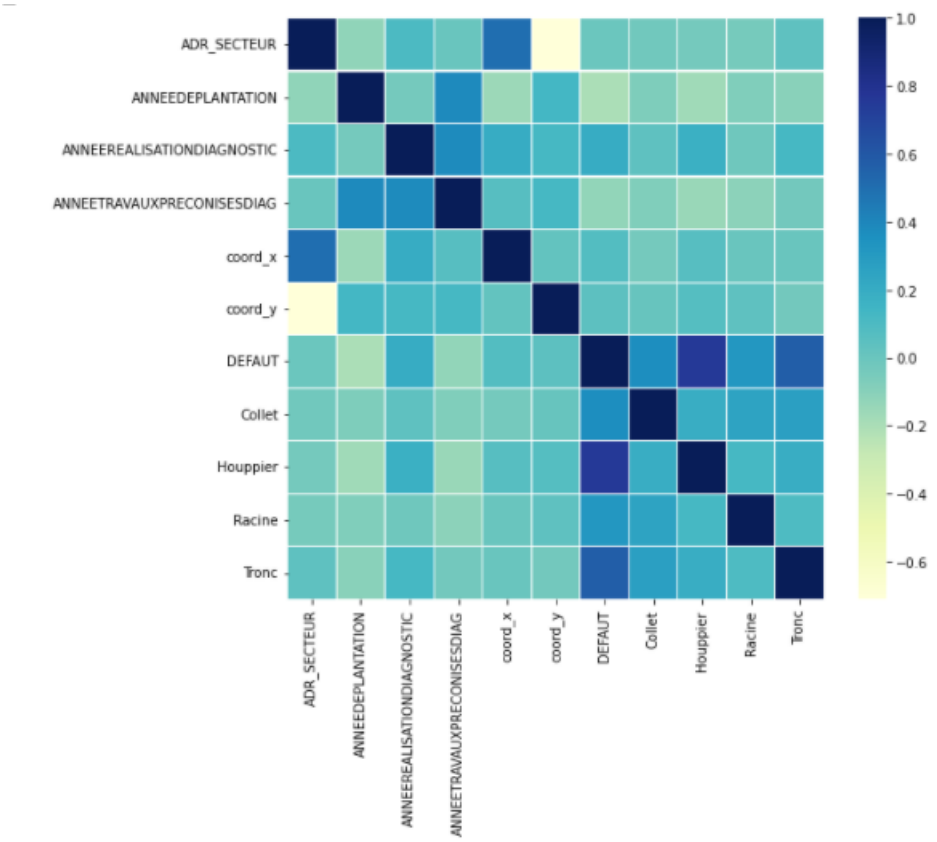

FIG. 2 – Corrélation entre Variables Quantitatives

On remarque que la variable la plus corrélé positivement avec la variable *DEFAUT*, est la variable *ANNEEREALISATIONDIAGNOSTIC*, appart les variables *Collet*, *Houppier*, *Racine*, *Tronc* bien sûr.

3.4.2 Dépendance entre Variables Qualitatives et *DEFAUT*

- Pour un seuil fixé à 1%, il y a indépendance entre *RAISONDEPLANTATION* et *DEFAUT*.
- Pour un seuil fixé à 1%, il y a indépendance entre *TRAITEMENTCHENILLES* et *DEFAUT*.
- Pour un seuil fixé à 1%, il n'y a pas indépendance entre *VARIETE* et *DEFAUT*.
- Pour un seuil fixé à 1%, il n'y a pas indépendance entre *REMARQUES* et *DEFAUT*.

4. Détection et traitement des valeurs manquantes

4.1 Par suppression

Vu que les variables *IDENTIFIANTPLU*, *INTITULEPROTECTIONPLU*, *TYPEIMPLANTATIONPLU*, n'apportent aucune information importante pour détecter un défaut ou non d'un arbre vu qu'elles sont que des identifiants, on va les supprimer.

Après l'étude d'indépendance, on décide de supprimer la variables *TRAITEMENTCHENILLES* et *RAISONDEPLANTATION*.

On va supprimer aussi les variables *Collet*, *Houppier*, *Racine*, *Tronc* sinon on tombe sur un sur-apprentissage.

4.2 Par imputation

On traite les valeurs manquantes de *VARIETE*, *ESPECE*, *TRAVAUXPRECONISESDIAG*, *STADEDEDEVELOPPEMENT*, *PRIORITEDERENOUVELLEMENT*, *DIAMETREARBREAUNMETRE*, *NOTEDIAGNOSTIC*, *STADEDEVELOPPEMENTDIAG*, *ANNEEREALISATIONDIAGNOSTIC*, *VIGUEUR* par *'Most Common Class'* qui renvoie une liste des éléments 'n' supérieurs du plus commun au moins commun, comme spécifié le paramètre 'n'.

L’attribut Remarques comporte du texte libre, décrivant des informations notées sur les arbres, par des techniciens ou botanistes, lors des diagnostics. Malheureusement, avec 1 684 valeurs, dont 1 291 valeurs uniques (ne concernant qu’un arbre), cet attribut est difficilement exploitable en l’état, on traite ses valeurs manquantes de par la création d'une classe *'Rien'*

Finalement,
On traite les valeurs manquantes de *ANNEETRAVAUXPRECONISESDIAG* par imputation par médian, cette technique consiste à remplacer les données manquantes par des estimations statistiques des valeurs manquantes. L’objectif de toute technique d’imputation est de produire un ensemble complet de données qui peut être utilisé pour former des modèles d’apprentissage automatique.

5. Choix des meilleurs classifieurs uni-label et multi-label

5.1 Méthodologie

Une fois le corpus préparé, les expériences de classification peuvent être menées. Les métriques d’évaluation qui ont été fournies sur les classifieurs de référence sont l’exactitude, la précision micro et macro, et le rappel micro et macro, Nous avons donc considéré le problème de classification multi-label comme quatre problèmes de classification uni-label.

Méthodes d’ensemble : les méthodes d’ensemble pour la classification nous semblent pertinentes en raison du nombre important d’instances dans le jeu de données et de la difficulté de la tâche de prédiction de la localisation d’un défaut. Les programmes RandomForest et Ada BoostM1, procédant par bagging ou par boosting, et arbre de décision ont été utilisés.

Méthodes bayésiennes : les méthodes bayésiennes étant généralement performantes, les programmes de classification NaiveBayes a été testé.

- NaiveBayes : En statistique, les classificateurs de Bayes naïfs sont une famille de simples « classificateurs probabilistes » basés sur l'application du théorème de Bayes avec des hypothèses d'indépendance fortes (naïves) entre les caractéristiques. Ils sont parmi les modèles de réseau bayésiens les plus simples, mais couplés à l'estimation de la densité du noyau, ils peuvent atteindre des niveaux de précision plus élevés. Les classificateurs De Bayes naïfs sont hautement évolutifs, nécessitant un certain nombre de paramètres linéaires dans le nombre de variables (caractéristiques/prédicteurs) dans un problème d'apprentissage. L'apprentissage de la probabilité maximale peut être effectué en évaluant une expression de forme fermée, qui prend du temps linéaire, plutôt que par approximation itérative coûteuse comme utilisé pour de nombreux autres types de classificateurs. Dans la littérature statistique, les modèles de Bayes naïfs sont connus sous une variété de noms, y compris les Bayes simples et les Bayes de l'indépendance.

6. Défi 1 : prédiction des défauts de l'arbre

6.1 Classification uni-label

Résultats pour la prédiction uni-label :

	Exactitude	Précision	Rappel	F-mesure
Référence	0.86	0.82	0.72	0.762
RandomForest	0.883	0.84	0.75	0.881
RandomForest(CV 10 folds)	0.884	0.842	0.757	0.881
AdaBoostM1	0.834	0.735	0.728	0.833
AdaBoostM1(CV 10 folds)	0.834	0.732	0.735	0.883
ArbreDeDécision	0.832	0.734	0.733	0.832
ArbreDeDécision(CV 10 folds)	0.834	0.729	0.735	0.832
NativeBayes	0.802	0.680	0.740	0.804
NativeBayes(CV 10 folds)	0.802	0.681	0.741	0.804

On remarque qu'avec le RandomForest on a une exactitude et une précision plus proche de l'exactitude et la précision référence alors que si on parle de rappel la meilleur résultat est donné par le AdaBoostM1, alors qu' avec le NativeBayes on trouve une F-mesure proche de la F-mesure référence. Mais en général es résultats sont plus ou moins proches entre eux. Et on peut dire que tout algorithme ne donne pas des valeurs proches de la variable F-mesure.

6.2 Classification multi-label

Résultats pour la prédiction multi-label :

	Exactitude	Précision	Rappel
Référence	0.86	0.64	0.37

Test d'exactitude :

	Collet	Houppier	Racine	Tranc
RandomForest	0.943	0.892	0.956	0.892
AdaBoostM1	0.915	0.854	0.935	0.850
ArbreDeDécision	0.919	0.847	0.936	0.852
NaiveBayes	0.901	0.789	0.867	0.821

Test de précision :

	Collet	Houppier	Racine	Tranc
RandomForest	0.943	0.893	0.956	0.893
AdaBoostM1	0.917	0.851	0.930	0.850
ArbreDeDécision	0.919	0.855	0.936	0.851
NaiveBayes	0.901	0.789	0.867	0.821

Test de rappel :

	Collet	Houppier	Racine	Tranc
RandomForest	0.942	0.893	0.955	0.893
AdaBoostM1	0.917	0.850	0.934	0.850
ArbreDeDécision	0.919	0.851	0.935	0.854
NaiveBayes	0.901	0.789	0.867	0.821

En classification multi-label, qui consiste en un problème d'apprentissage où plusieurs classes peuvent être affectées simultanément à un exemple, on trouve usuellement deux grandes catégories d'approches :

— Transformer le problème pour le convertir en un problème pouvant être résolu par un algorithme de classification binaire ou multiclasses usuel ;
— Utiliser des algorithmes multiclasses transformés, qui peuvent résoudre directement le problème multi-label.

Nous, on a utilisé le Pipelining en Python, qui consiste à appliquer séquentiellement une liste de transformateurs (modélisation de données), puis un estimateur final (modèle ML). Les étapes de transformation doivent implémenter fit() et transform().La dernière étape, l'estimateur, doit mettre en œuvre fit() et predict().L'estimateur doit implémenter fit() cependant, pas nécessairement implémenter predict.

En bref, les pipelines sont configurés avec la fonctionnalité ajuster/transformer/prédire, de sorte que nous pouvons adapter l'ensemble du pipeline aux données d'entraînement et les transformer en données de test sans avoir à le faire individuellement pour tout ce que vous faites.

On remarque qu'avec le NativeBayes on trouve les mêmes résultats que ça soit dans l'exactitude, la précision ou le rappel, c'est-à-dire que il est pas le meilleur classifier pour le multi-label.

On remarque aussi que les valeurs obtenues sont un peu élevés comparèrent à les valeurs référence surtout en ce qui concerne la précision et le rappel.

Meilleur classifier pour les variables Collet, Houppier, Racine est AdaBoostM1, pour la variable Tronc est ArbreDeDécision.

7. Défi 2 : Connaitre mieux l'état du « parc végétal » de Grenoble

7.1 Objectif du défi

La seconde tâche, plus ouverte, vise à mieux connaitre l'état du « parc végétal » de Grenoble, mieux comprendre son évolution et fournir des préconisations pour faciliter son entretien. Pour cette seconde tâche, il est possible d'avoir recours à des données externes, de proposer des possibilités de visualisation…

7.2 Visualisation de la répartition des arbres de Grenoble selon différents critères

Les arbres étant plantés dans un environnement urbain, divers facteurs pourraient expliquer certains types de défaut. Un outil de visualisation a été développé afin de permettre de visualiser en contexte urbain les arbres du corpus. Cet outil requiert un fichier (au format CSV) comprenant la latitude, la longitude, la présence d'un défaut ou non, ainsi que la localisation d'un défaut au collet, sur le houppier, à la racine ou au tronc. L'outil permet de sélectionner un type de défaut, un nombre d'arbres présentant ce défaut et un nombre d'arbres ne présentant pas de défaut. Ces arbres sont alors affichés sur un fond de carte de la ville de Grenoble.

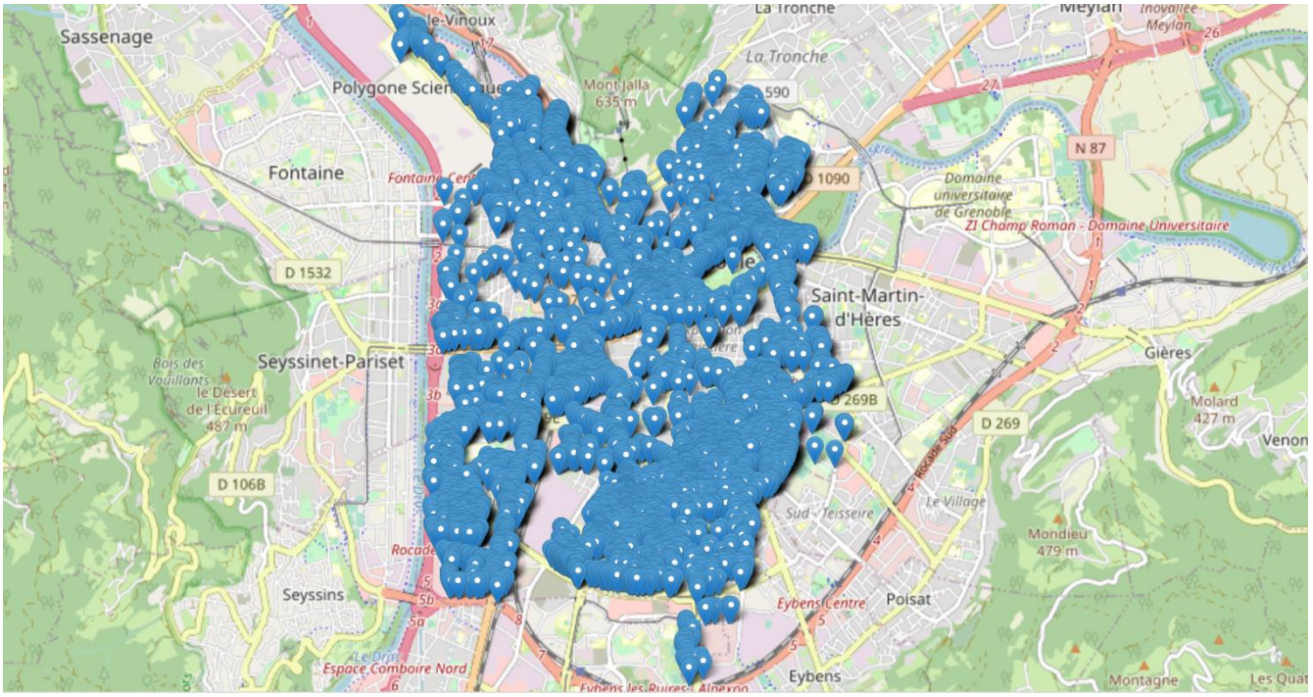

FIG. 3 – Positionnement des arbres sans et avec défauts

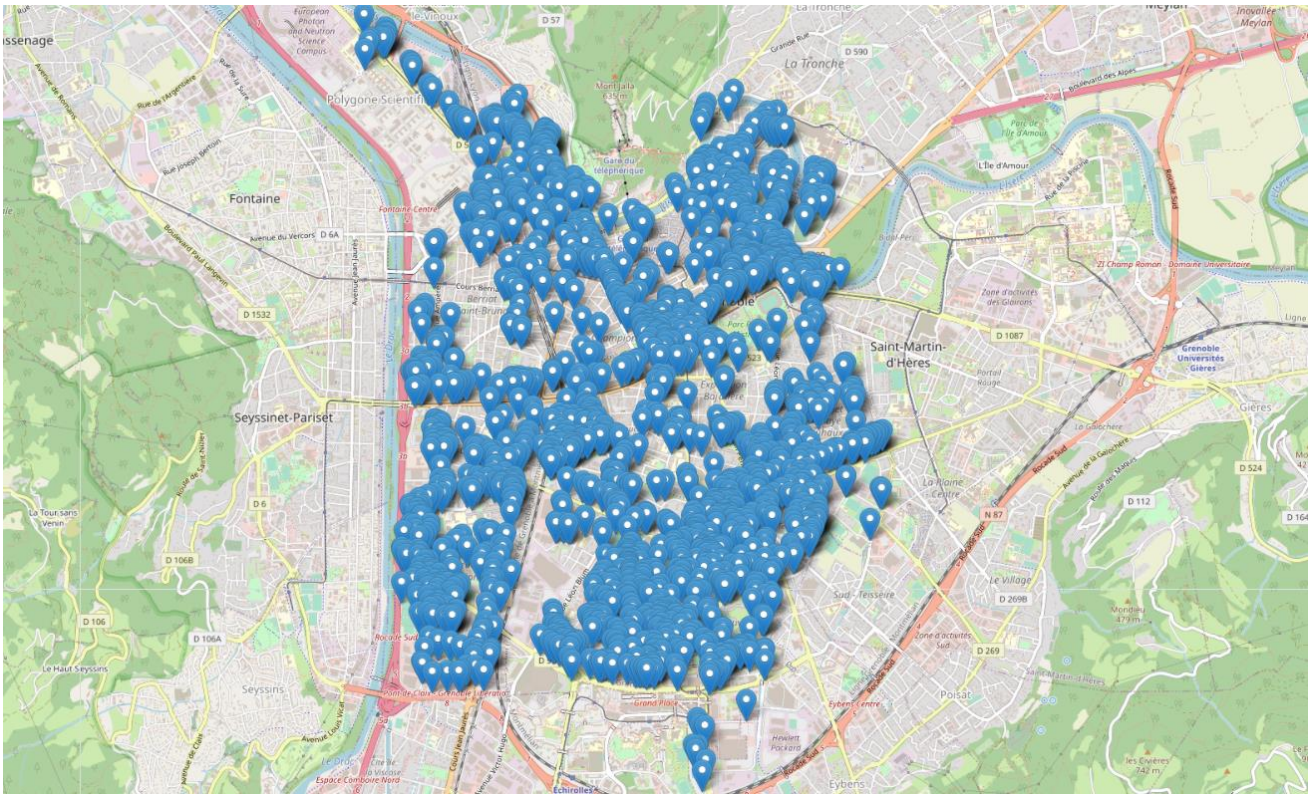

FIG. 4 – Positionnement des arbres avec défauts

FIG. 5 – Positionnement des arbres avec défauts dans le collet

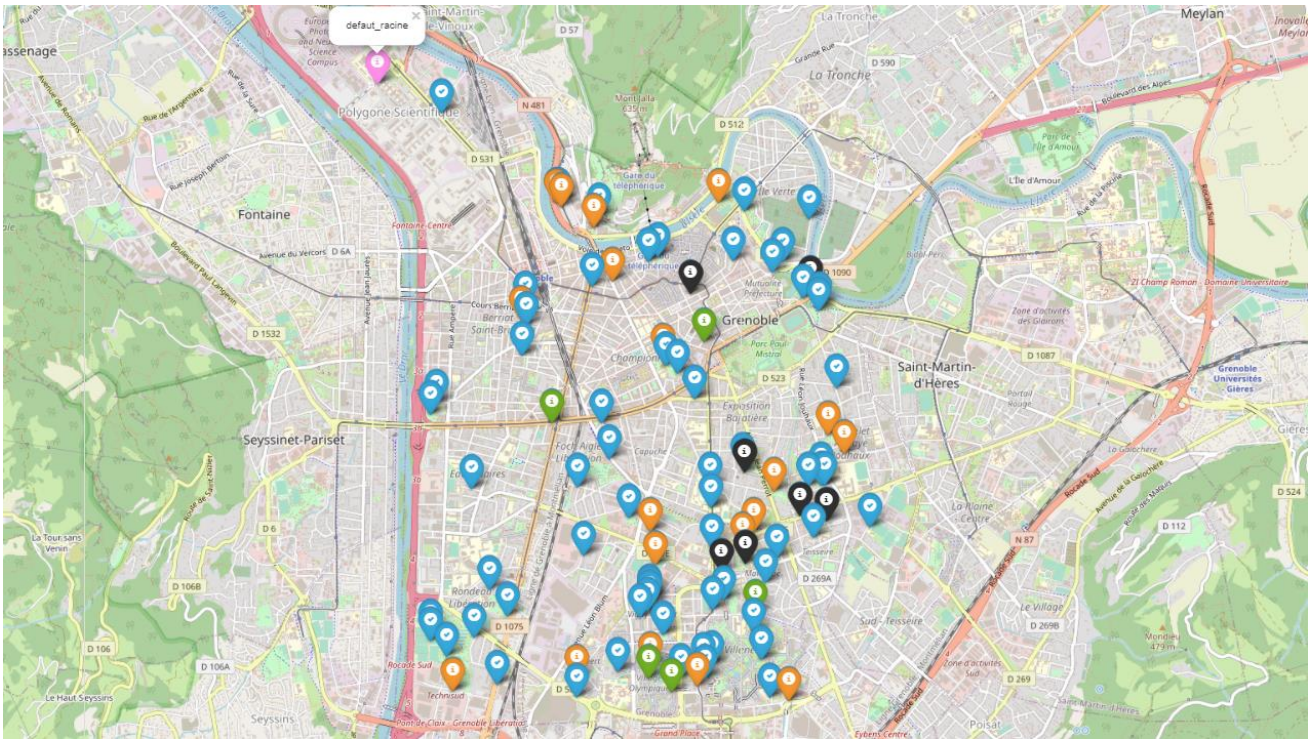

FIG. 6 – Positionnement des arbres avec des défauts dans différents endroits

Cet outil présente des avantages. S'agissant d'une vraie carte, mise à jour régulièrement, il fournit une interface réaliste permettant aux techniciens et botanistes de mieux cerner la situation du parc végétal. Le système d'échantillonnage permet d'étudier des phénomènes à petite échelle. La possibilité de zoomer permet de considérer l'étendue géographique du territoire.

8. Nuage de points géographique sur la variable DEFAUT

Dans un nuage de points géographique, chaque ligne qui réfère à une arbre avec défaut est représenté par un symbole sur la carte.

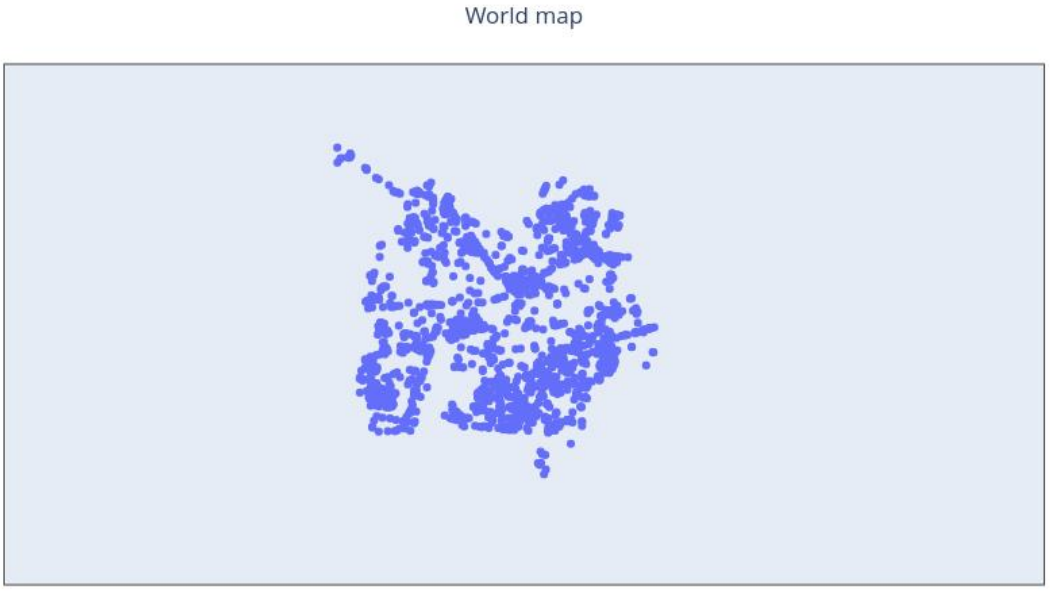

FIG. 7 – 8. Nuage de points géographique sur la variable DEFAUT dans world map