Prédiction de défauts dans les arbres du parc végétal Grenoblois et préconisations pour les futures plantations

Yelen Per*, Kevin Dalleau**, Malika Smail-Tabbone***

*LORIA UMR 7503, CNRS, yelen.per@loria.fr
**LORIA UMR 7503, Université de Lorraine, kevin.dalleau@loria.fr
***LORIA UMR 7503, Université de Lorraine, malika.smail@loria.fr

Résumé. Nous décrivons dans cet article notre réponse au défi EGC 2017. Une analyse exploratoire des données a tout d'abord permis de comprendre les distributions des différentes variables et de détecter de fortes corrélations. Nous avons défini deux variables supplémentaires à partir des variables du jeu de données. 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. Autant sur la tâche uni-label que multi-label, le meilleur classifieur dépasse les références d'environ 2%. 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.

1 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. Les contributions proposées sont au nombre de trois :

— une prise en compte de données climatiques ou de données sur la classification botanique des arbres ;
— une recherche de règles d'association ;
— un outil de visualisation du parc arboricole Grenoblois.

Le logiciel libre WEKA, offrant une implémentation des principaux algorithmes de classification, a été utilisé dans ce travail (Witten et Frank (2005)). Des programmes complémentaires ont été écrits principalement pour l'ingénierie des données.

2 Analyse exploratoire des données

Les données fournies constituent un corpus composé de 15 375 instances d'arbres, décrites par 34 attributs. Certains attributs décrivent l'arbre (*Code, DiamètreArbreÀUnMètre, AnnéeDePlantation, Espèce, Genre_Bota*...), son emplacement (*Adr_Secteur, Trottoir, FréquentationCible*, 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.

2.1 Nettoyage et typage des données

Une première phase a consisté à coder de façon systématique l'absence de valeurs (N/A, chaînes vides...). Des caractères spéciaux ont été retirés de l'attribut *Remarques*. Une deuxième phase a porté sur le typage des données en fonction de leur nature. Le logiciel WEKA détecte deux types d'attributs à partir de leurs valeurs : les attributs de type nominal (i.e. un nom représente une catégorie) et ceux de type numérique. Des conversions ont été effectuées pour améliorer l'exploitabilité du corpus. Ainsi, l'attribut *Adr_Secteur* (figure 1), dont les valeurs sont des nombres compris entre 1 et 6, sans relation d'ordre particulière, a été transformé en attribut nominal. De même, les variables de classe *Défaut, Collet, Houppier, Racine, Tronc*, à valeurs binaires, ont été transformées en attributs nominaux (cela permet d'appliquer des programmes de classification). La variable *DiamètreArbreÀUnMètre* a été transformée en attribut numérique en considérant le centre de l'intervalle (distribution en figure 2). Les valeurs de la variable *PrioritéDeRenouvellement*, nominales au départ, ont été codifiées sous forme d'intervalles (exemple : la valeur "moins de 5 ans" est transformée en intervalle 1-5).

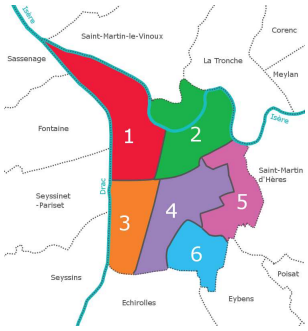

FIG. 1 – *Secteurs géographiques de Grenoble.*

2.2 Suppression d'attributs redondants

L'attribut *Code* identifiant de façon unique un arbre, a été supprimé avant l'application des programmes de classification. De même, la variable *Sous_Catégorie*, codifiant la variable *Sous_Catégorie_Description*, a été supprimée. Les variables *Code_Parent* et

Code_Parent_Desc sont aussi redondantes, et décrivent, pour un arbre, la station parente à laquelle il est rattaché. Ces deux variables ont été supprimées du corpus pour deux raisons. D'une part, le nombre de valeurs distinctes est très important (près de 1 140). D'autre part, l'information de la station d'un arbre, étant de nature géographique, elle est redondante avec les variables *Adr_Secteur* ainsi que *coord_x* et *coord_y*.

2.3 Création d'attributs supplémentaires

Deux attributs ont été créés par aggrégation, pour chacun, de deux attributs du corpus.

Le premier attribut, appelé *EvolutionJsqDiag*, est la différence entre le stade de développement initial de l'arbre et son stade de développement lors du diagnostic. Autrement dit il reflète l'évolution de l'arbre, de sa plantation jusqu'au diagnostic. Il se calcule de la façon suivante :

$$EvolutionJsqDiag = StadeDeDéveloppement - StadeDéveloppementDiag,$$

les valeurs des variables *StadeDeDéveloppement* et *StadeDévelopementDiag* étant codifiées par les entiers 1, 2 et 3, correspondant respectivement aux valeurs "arbre jeune", "arbre adulte" et "arbre viellissant". Une différence positive, nulle ou négative, correspond alors à une évolution positive, constante ou négative, du développement de l'arbre.

Le second attribut, appelé *NbAnnéesAvantProchainDiag* est également une différence, qui mesure, par un nombre d'années, la nécessité plus ou moins importante d'un prochain diagnostic, conséquence d'éventuels défauts ayant endommagé l'arbre. Il se définit de la façon suivante :

$$NbAnnéesAvantProchainDiag = AnnéeTravauxPréconisésDiag - AnnéeRéalisationDiagnostic.$$

Nous avons vérifié que ces deux attributs supplémentaires contribuent à la prédiction (par exemple dans les règles de classification données dans la section 3.2).

2.4 Structuration d'un attribut textuel

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. Nous distinguons trois façons de considérer cet attribut :

- les valeurs de l'attribut sont laissées telles quelles ;
- les valeurs de l'attribut sont transformées en vecteurs de mots en utilisant le filtre *StringToWordVector* du logiciel WEKA. Ce filtre crée un dictionnaire formé des mots contenus dans le texte des remarques, avec leur fréquence d'apparition. Les mots ayant une fréquence d'apparition minimale (de 50 dans notre cas) sont ensuite transformés en attributs. Chacun de ces attributs va contenir, pour chaque remarque, la valeur 1 ou 0 selon que cette dernière contienne ou non le mot en attribut. Finalement, la variable *Remarques* est supprimée du corpus ;
- pour chaque remarque, les TF-IDF des termes composant le texte de la remarque sont calculés. La somme de ces valeurs forme la valeur d'un attribut numérique représentant l'importance des mots de la remarque selon la méthode de pondération TF-IDF. Les remarques longues, composées de termes rares, sont privilégiées.

2.5 Retour sur la qualité des données

Globalement, les données sont de bonne qualité, puisque nous n'avons pas repéré d'erreurs de saisie ou d'incohérences. Certains attributs ont un fort taux de valeurs manquantes, mais cela ne constitue pas forcément un problème lorsque les programmes de classification sont capables de s'accomoder de ces données manquantes.

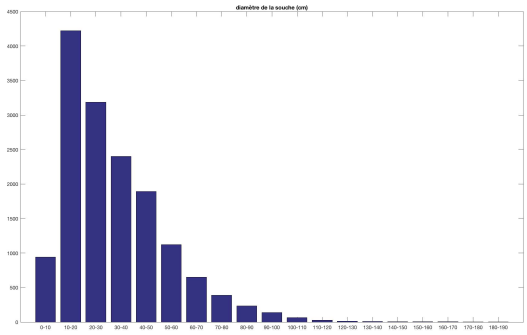

FIG. 2 – *Distribution des valeurs de la variable* DiamètreArbreÀUnMètre.

3 Choix des meilleurs classifieurs uni-label et multi-label

3.1 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 (Yang et Liu (1999)). Nous avons donc considéré le problème de classification multi-label comme quatre problèmes de classification uni-label. Quelques algorithmes de classification supervisée ont été expérimentés, selon trois configurations différentes, correspondant à la manière de gérer l'attribut *Remarques*. Nous avons choisi de tester un à deux algorithmes de quatre catégories différentes.

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 *AdaBoostM1*, procédant par bagging ou par boosting, ont été utilisés (Breiman (2001); Freund et Schapire (1996)).

Méthodes produisant un modèle de classification explicite. Les méthodes de classification produisant un modèle explicite, tel qu'un arbre de décision ou des règles de classification, sont utilisées de façon complémentaire pour permettre aux experts du domaine de comprendre la façon dont la prédiction se fait sur la base des attributs. Les programmes *J48* et *JRip* ont été utilisés (Quinlan (1993); Cohen (1995)).

Méthodes bayésiennes. Les méthodes bayésiennes étant généralement performantes, les programmes de classification *NaiveBayes* et *BayesNet* ont été testés.

Méthodes à base d'instances. La méthode des plus proches voisins (programme *IBk*) a été considérée car elle privilégie une approche locale pour la classification qui s'oppose à la construction d'un modèle global recherché avec les autres méthodes (Aha et al. (1991)).

La plupart des programmes de classification ont été utilisés avec les paramètres par défaut préconisés dans le logiciel WEKA. Le tableau ci-dessous récapitule les choix qui ont été faits lorsque ce n'est pas le cas.

AdaBoostM1	
classifier	*RandomTree*
RandomTree.breakTiesRandomly	*True*
J48	
confidenceFactor	0.6
JRip	
minNo	9.0
optimizations	5

FIG. 3 – *Paramètres modifiés.*

3.2 Résultats

Les figures 4 et 5 ci-dessous décrivent les résultats des trois expériences pour les deux problèmes de classification uni-label et multi-label (la F-mesure est la moyenne harmonique de la précision et du rappel). Les scores obtenus, sur la base d'une validation croisée à 10 plis, sont comparés par rapport aux scores de référence fournis. Les scores surlignés en gras dépassent les scores de référence. Les scores avec une police plus grande, représentent, pour une expérience, la meilleure valeur de chaque catégorie de score.

En guise d'éléments explicites de classification, nous présentons le texte de quelques règles construites par le programme *JRip* pour la prédiction de la présence d'un défaut sur un arbre (les nombres entre parenthèses correspondent aux nombres d'instances du corpus d'apprentissage pour lesquelles la règle s'applique et aux nombres de prédictions erronées) :

1. *NoteDiagnostic* = 'Arbre Davenir Incertain' $\wedge$ *AnnéeRéalisationDiagnostic* $\geq 2015 \rightarrow$ *Défaut* (1776/189).
2. *PrioritéDeRenouvellement* $\leq 15 \wedge$ *PrioritéDeRenouvellement* $\leq 7.5 \rightarrow$ *Défaut* (876/77).
3. *PrioritéDeRenouvellement* $\leq 15 \wedge$ *NoteDiagnostic* = 'Arbre Davenir Incertain' $\wedge$ *AnnéeRéalisationDiagnostic* $\geq 2014 \wedge$ *NbAnnéesAvantProchainDiag* $\leq 1 \wedge$ *coord_y* $\leq 4224826.76743 \rightarrow$ *Défaut* (243/26).
4. *PrioritéDeRenouvellement* $\leq 15 \wedge$ *NbAnnéesAvantProchainDiag* $\leq 2 \wedge$ *coord_x* $\leq 1914782.38638 \wedge$ *NoteDiagnostic* = 'Arbre a abattre dans les 10 Ans' $\rightarrow$ *Défaut* (26/2).

5. *DiamètreArbreÀUnMètre* $\geq$ 35 $\wedge$ *Sous_Catégorie_Description* = 'Arbre despaces ouverts' $\wedge$ *NbAnnéesAvantProchainDiag* $\leq$ 1 $\wedge$ *coord_x* $\leq$ 1914220.45606 $\wedge$ *coord_x* $\geq$ 1913201.16333 $\rightarrow$ *Défaut* (101/15).
6. *DiamètreArbreÀUnMètre* $\geq$ 35 $\wedge$ *NbAnnéesAvantProchainDiag* $\leq$ 2 $\wedge$ *Sous_Catégorie_Description* = 'Arbre despaces ouverts' $\wedge$ *TravauxPréconisésDiag* = 'Taille de bois mort' $\rightarrow$ *Défaut* (60/10).

	Exactitude	Précision	Rappel	F-mesure
Référence	86.0	82.0	72.0	76.7
Expérience n° 1				
BayesNet	81.7	72.6	69.9	71.2
NaiveBayes	81.1	71.0	70.8	70.9
J48	84.8	**87.2**	62.5	72.8
JRip	85.6	**84.6**	68.0	75.4
IBk	81.5	71.3	**72.3**	71.8
RandomForest	**87.6**	**86.9**	**72.7**	**79.2**
AdaBoostM1	84.7	78.6	**73.0**	75.7
Expérience n° 2				
BayesNet	81.3	72.1	69.5	70.8
NaiveBayes	80.6	69.8	71.4	70.6
J48	**86.5**	**82.9**	**73.7**	**78.0**
JRip	85.7	**84.7**	68.4	75.7
IBk	81.8	72.1	71.5	71.8
RandomForest	**87.5**	**85.4**	**74.4**	**79.5**
AdaBoostM1	84.9	78.8	**73.3**	76.0
Expérience n° 3				
BayesNet	81.4	71.7	70.7	71.2
NaiveBayes	80.5	69.8	70.7	70.2
J48	**86.3**	**83.4**	**72.1**	**77.3**
JRip	85.9	**84.0**	70.0	76.4
IBk	75.9	60.8	**73.2**	66.4
RandomForest	**87.2**	**84.6**	**74.3**	**79.1**
AdaBoostM1	84.7	78.5	**72.8**	75.5

FIG. 4 – *Résultats des expériences de classification uni-label.*

3.3 Comparaison des classifieurs et des expériences et interprétations

L'examen des résultats quantitatifs des différents classifieurs construits semblent montrer des performances variables dans les 3 expériences et une amélioration des meilleures performances dans la troisième expérience (dans laquelle les mots de l'attribut textuel *Remarques* ont été analysés et agrégés en un nombre). Néanmoins, les performances des différents classifieurs dans les trois expériences restent proches et nous avons besoin de nous assurer que les différences ne sont pas simplement dûes aux erreurs d'estimation. Nous avons donc réalisé un test de Student (*t*-test) apparié pour comparer 100 valeurs de chaque métrique obtenues par chaque classifieur pour 10 répétitions de validations croisées à 10 plis. Les

tests ont été appliqués sur les métriques exactitude et F-mesure comme une aggrégation de la précision et du rappel avec un intervalle de confiance de 95%.

Tout d'abord, nous avons testé la significativité des différences obtenues par l'algorithme *RandomForest* sur les trois jeux de données correspondant aux trois expériences selon le traitement réservé à l'attribut textuel *Remarques*. Pour cela nous avons utilisé comme *baseline* le jeu de données brutes (sans traitement du champ textuel). Comme le tableau 6 l'indique, l'algorithme *RandomForest* ne donne pas de meilleurs résultats statistiquement significatifs sur les jeux de données des expériences 1 et 2 pour l'exactitude ni pour la F-mesure, dans l'intervalle de confiance considéré.

	Micro			Macro		
	Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
Référence	70.0	47.0	56.2	64.0	37.0	46.9
Expérience n° 1						
BayesNet	43.1	**56.8**	49.0	38.2	**53.3**	44.6
NaiveBayes	40.5	**59.3**	48.1	36.2	**55.6**	43.9
J48	66.1	40.8	50.4	58.3	28.3	38.1
JRip	66.0	46.6	54.6	56.3	33.2	41.8
IBk	54.4	**53.5**	54.0	47.1	**46.6**	46.8
RandomForest	**79.6**	42.4	55.4	**75.5**	31.7	44.7
AdaBoostM1	64.4	**52.6**	**57.9**	57.9	**44.9**	**50.6**
Expérience n° 2						
BayesNet	42.7	**55.6**	48.3	37.8	**52.3**	43.8
NaiveBayes	38.6	**58.7**	46.6	35.1	**54.9**	42.8
J48	65.5	**48.9**	56.0	57.5	**39.0**	46.5
JRip	66.0	**47.9**	55.5	56.0	34.1	42.4
IBk	55.1	**53.0**	54.0	48.3	**46.4**	**47.3**
RandomForest	**76.0**	**47.3**	**58.3**	**70.1**	**38.2**	**49.5**
AdaBoostM1	66.0	**51.9**	**58.1**	59.8	**44.5**	**51.0**
Expérience n° 3						
BayesNet	42.9	**58.0**	49.3	37.8	**54.0**	44.5
NaiveBayes	39.9	**58.8**	47.5	35.6	**54.8**	43.2
J48	65.8	**47.1**	54.9	57.3	36.9	44.8
JRip	65.3	**47.9**	55.3	56.6	34.7	43.0
IBk	46.3	**52.8**	49.4	40.4	**45.9**	43.0
RandomForest	**74.5**	**48.2**	**58.5**	**68.8**	**39.4**	**50.1**
AdaBoostM1	64.4	**51.7**	**57.4**	57.8	**44.3**	**50.1**

FIG. 5 – *Résultats des expériences de classification multi-label.*

Nous avons ensuite comparé les performances des 7 algorithmes étudiés sur le jeu de données brutes (expérience 1). Les résultats, consignés dans le tableau 7, confirment que l'algorithme *RandomForest* présente de meilleurs résultats, statistiquement significatifs pour l'exactitude et la F-mesure dans l'intervalle de confiance considéré, que les autres algorithmes considérés. Nous constatons également qu'à l'occasion de ces tests statistiques, une meilleure estimation des métriques (F-mesure et exactitude) a été faite grâce à la répétition des validations croisées et fournit des résultats plus favorables que ceux présentés dans la section précédente, obtenus avec une seule validation croisée.

	Jeu d'expérience n° 1	Jeu d'expérience n° 2	Jeu d'expérience n° 3
F-mesure	**91**	91*	91
Exactitude	**87.3**	87	87.3

* dégradation statistiquement significative

FIG. 6 – *Résultats du test t sur 100 expériences de classification uni-label avec l'algorithme RandomForest sur les trois jeux de données (le premier jeu de données sert de baseline).*

Algorithme	F-mesure	Exactitude
RandomForest	**91**	**87.2**
BayesNet	87*	81.6*
NaiveBayes	86*	81*
JRip	90*	85.6*
J48	90*	84.5*
IBk	86*	81.5*
AdaBoostM1	89*	84.8

* dégradation statistiquement significative

FIG. 7 – *Résultats du test t sur 100 expériences de classification uni-label pour comparer l'algorithme RandomForest (baseline) avec les six autres algorithmes.*

Pour ce qui est de la classification multi-label, nous avons comparé, pour chaque classe, les performances en termes d'AUC (aire sous la courbe ROC) et de F-mesure de l'algorithme *RandomForest* avec les six autres algorithmes. Les résultats du test *t* montrent également qu'aucun algorithme n'améliore de façon significative les performances de l'algorithme *RandomForest*.

La méthode d'ensemble de classifieurs *RandomForest* est donc celle qui fournit le meilleur modèle de prédiction pour les problèmes posés. Ce sont ces classifieurs qui ont été utilisés pour réaliser les prédictions sur l'ensemble de test. Les treize attributs les plus prédictifs d'un défaut sont les suivants : *Adr_Secteur*, *AnnéeDePlantation*, *AnnéeRéalisationDiagnostic*, *AnnéeTravauxPréconisésDiag*, *DiamètreArbreÀUnMètre*, *NoteDiagnostic*, *PrioritéDeRenouvellement*, *StadeDeDéveloppement*, *TravauxPréconisésDiag*, *NbAnnéesAvantProchainDiag*, *coord_x*, *coord_y*.

4 Analyses exploratoires des données et aide à la décision

Pour répondre à la seconde tâche du défi, nous avons exploré trois pistes. La première consiste à enrichir les données du corpus avec des données externes climatologiques et botaniques, et de vérifier si cela permet de réaliser une meilleure prédiction d'un défaut et de ses localisations sur l'arbre. La deuxième consiste à rechercher des combinaisons fréquentes voire des associations de défauts avec certaines caractéristiques des arbres. La dernière consiste à visualiser les données du corpus enrichies avec des données urbaines afin d'aider les décideurs à comprendre certains phénomènes.

4.1 Apport de données externes à la classification supervisée

Des données climatiques [1] ont été agrégées, pour chaque arbre, dans la période comprise entre sa plantation et le diagnostic. Ces données sont relatives à la pluviométrie, la température, l'ensoleillement

1. www.prevision-meteo.ch

et au vent. Une fois ces données intégrées au corpus, les expériences ont été refaites avec les 2 meilleurs classifieurs et l'expérience n° 3. Les résultats de classification uni-label et multi-label sont respectivement décrits dans les figures 8 et 9. On y constate une très lègère amélioration du rappel et de la F-mesure pour *AdaBoostM1*, au détriment d'une très fine diminution de la précision, pour les 2 classifieurs. En somme, l'apport d'information pour la prédiction de la présence de défaut peut être considéré comme nul. Pour ce qui est de la prédiction de la localisation d'un défaut, les valeurs de la F-mesure augmentent légèrement. Finalement, l'intérêt de la prise en compte de données climatiques est certes minime, mais présent.

	Exactitude	Précision	Rappel	F-mesure
Expérience de référence				
RandomForest	87.2	84.6	74.3	79.1
AdaBoostM1	84.7	78.5	72.8	75.5
Expérience avec les données climatiques				
RandomForest	**87.2**	84.5	**74.3**	**79.1**
AdaBoostM1	**84.7**	78.4	**73.0**	**75.6**

FIG. 8 – *Résultats des classifications uni-label avec prise en compte de données climatiques.*

	Micro			Macro		
	Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
Expérience de référence						
RandomForest	74.5	48.2	58.5	68.8	39.4	50.1
AdaBoostM1	64.4	51.7	57.4	57.8	44.3	50.1
Expérience avec les données climatiques						
RandomForest	73.4	**49.3**	**58.9**	67.3	**40.4**	**50.5**
AdaBoostM1	63.5	**52.1**	57.3	57.0	**44.9**	**50.2**

FIG. 9 – *Résultats des classifications multi-label avec prise en compte de données climatiques.*

Des données de classification botanique ont été recherchées. Une base de données (accessible par le site `www.tropicos.org`) a permis d'associer les genres botaniques présents dans le corpus à trois propriétés (famille, ordre et super-ordre). Les 107 genres botaniques ont ainsi été agrégés en 42 familles, 21 ordres et 6 super-ordres. Comme pour les données climatiques, les 2 meilleurs classifieurs ont été évalués sur la base de l'expérience n° 3. Les figures 10 et 11 illustrent les résultats obtenus. Nous constatons que malgré une baisse des performances pour la classification uni-label, seul le classifieur *AdaBoostM1* présente de meilleures performances. En conclusion, l'apport des données taxonomiques ne semble pas clair, probablement à cause d'une certaine redondance avec les données du corpus (espèce, genre botanique, variété).

4.2 Recherche d'associations intéressantes

En nous focalisant sur un sous-ensemble intéressant de caractéristiques des arbres, nous avons voulu explorer les possibles associations présentes dans les données et susceptibles d'aider les décideurs par rapport aux choix de plantation. Nous avons recherché les règles d'association grâce au programme *Apriori* (Agrawal et Srikant (1994)). Nous présentons ci-après quelques exemples de règles d'association à support (supp) relativement modeste mais présentant une bonne confiance (conf). Nous avons également

	Exactitude	Précision	Rappel	F-mesure
Expérience de référence				
RandomForest	87.2	84.6	74.3	79.1
AdaBoostM1	84.7	78.5	72.8	75.5
Expérience avec les données de classification botanique				
RandomForest	87.0	84.4	73.6	78.6
AdaBoostM1	84.4	77.9	72.7	75.2

FIG. 10 – *Classifications uni-label exploitant les données de classification botanique.*

	Micro			Macro		
	Précision	Rappel	F-mesure	Précision	Rappel	F-mesure
Expérience de référence						
RandomForest	74.5	48.2	58.5	68.8	39.4	50.1
AdaBoostM1	64.4	51.7	57.4	57.8	44.3	50.1
Expérience avec les données de classification botanique						
RandomForest	74.0	48.1	58.3	68.5	**39.4**	**50.1**
AdaBoostM1	**64.7**	**51.9**	**57.6**	**58.8**	**44.8**	**50.9**

FIG. 11 – *Classifications multi-label exploitant les données de classification botanique.*

calculé la mesure de lift définie comme le rapport entre la probalité jointe des parties gauche et droite et le produit des probalités des deux parties. Rappelons qu'un lift supérieur à 1 traduit une corrélation positive entre les deux parties de la règle et donc le caractère significatif de l'association. Ceci est vrai pour l'ensemble des règles présentées ici :

1. *Adr_Secteur* $= 2 \wedge$ *Genre_Bota* = 'Populus' $\wedge$ *Trottoir* = 'non' $\rightarrow$ *Défaut* (supp : 97 ; conf : 0.85 ; lift : 2.6).
2. *Adr_Secteur* $= 5 \wedge$ *Genre_Bota* = 'Populus' $\wedge$ *Trottoir* = 'non' $\rightarrow$ *Défaut* (supp : 94 ; conf : 0.8 ; lift : 2.45).
3. *Genre_Bota* = 'Populus' $\wedge$ *Trottoir* = 'non' $\wedge$ *Variété* = 'Italica' $\rightarrow$ *Défaut* (supp : 174 ; conf : 0.72 ; lift : 2.2).
4. *Adr_Secteur* $= 2 \wedge$ *Genre_Bota* = 'Platanus' $\wedge$ *Sous_Catégorie_Description* = 'Arbre de voirie' $\wedge$ *Trottoir* = 'oui' $\wedge$ *Collet* $\rightarrow$ *Houppier* (supp : 78 ; conf : 0.93 ; lift = 4.3).
5. *Adr_Secteur* $= 2 \wedge$ *Espèce* = 'acerifolia' $\wedge$ *Genre_Bota* = 'Platanus' $\wedge$ *Sous_Catégorie_Description* = 'Arbre de voirie' $\wedge$ *Trottoir* = 'oui' $\wedge$ *Collet* $\rightarrow$ *Houppier* (supp : 78 ; conf : 0.93 ; lift : 4.3).
6. *Genre_Bota* = 'Platanus' $\wedge$ *Collet* $\rightarrow$ *Houppier* (supp : 171 ; conf : 0.83 ; lift : 3.8).

Ces règles permettent de suggérer des hypothèses intéressantes à valider par des études plus poussées. Ainsi les règles 1 et 2 montrent que la probabilité pour un peuplier de présenter un défaut sachant qu'il est planté loin d'un trottoir dans le secteur 2 ou 5 est forte. De même la règle 4 montre que la probabilité d'un défaut au houppier pour un platane touché au collet et planté comme arbre de voirie dans le secteur 2 à proximité d'un trottoir est forte. Si certaines de ces règles sont validées, les botanistes chargés des plantations peuvent en tenir compte par exemple en évitant les mauvaises associations entre secteurs et variétés botaniques.

4.3 Visualisation en contexte urbain des arbres avec leurs défauts

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 lattitude, 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 [2]. La figure 12 représente une capture d'écran de l'outil, dans lequel sont représentés 50 arbres avec un défaut à la racine, parmi 50 arbres sans défaut.

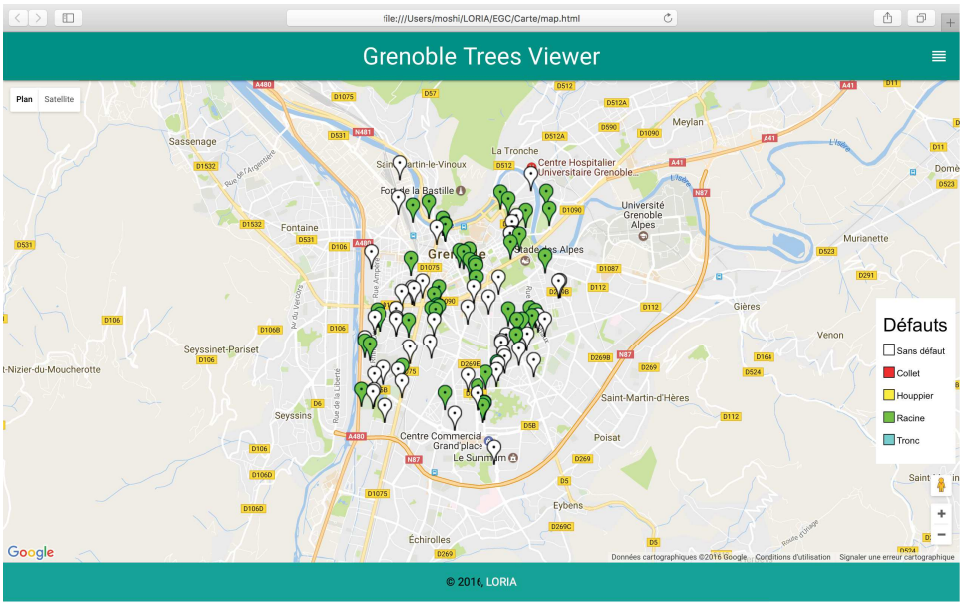

FIG. 12 – *Capture d'écran de l'outil de visualisation avec 50 arbres présentant un défaut à la racine, parmi 50 arbres sans défaut.*

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.
La manipulation de l'outil a par exemple permis de faire quelques constats intéressants :

- les arbres avec un défaut au niveau du tronc sont nombreux à proximité de grandes voies (autoroutes, avenues, boulevards...) ;
- les arbres avec un défaut au niveau de la racine sont nombreux à proximité d'intersections de voies ;
- les arbres avec un défaut sur le houppier sont nombreux dans des lieux à fréquentation publique importante (parcs, aires de jeux...).

D'autres cas d'utilisation pourraient enrichir cet outil de visualisation tels que l'utilisation d'un formulaire qui servirait de support pour la saisie de divers critères de sélection des arbres à afficher.

2. `https://developers.google.com/maps/documentation/javascript/?hl=fr`

5 Conclusion

Nous avons répondu au défi EGC 2017 en testant différentes méthodes de fouille de données que nous avons jugées pertinentes et complémentaires pour l'exploitation des données fournies dans le cadre des deux tâches du défi. Les classifieurs construits nous semblent assez précis bien que l'amélioration des performances par rapport à la référence soit relativement modeste. Les éléments de connaissance extraits nous semblent susceptibles d'apporter aux décideurs une certaine compréhension des phénomènes de dégradation des arbres en milieu urbain. Nous pensons que l'outil de visualisation des arbres en contexte urbain pourrait gagner à intégrer d'autres cas d'utilisation suggérés par les experts.

Références

Agrawal, R. et R. Srikant (1994). Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, et C. Zaniolo (Eds.), *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pp. 487–499. Morgan Kaufmann.

Aha, D. W., D. F. Kibler, et M. K. Albert (1991). Instance-based learning algorithms. *Machine Learning 6*, 37–66.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Cohen, W. W. (1995). Fast effective rule induction. In A. Prieditis et S. J. Russell (Eds.), *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pp. 115–123. Morgan Kaufmann.

Freund, Y. et R. E. Schapire (1996). Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, pp. 148–156. Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann.

Witten, I. et E. Frank (2005). *Data Mining : Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann.

Yang, Y. et X. Liu (1999). A re-examination of text categorization methods. In *SIGIR '99 : Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pp. 42–49. ACM.

Summary

We describe in this paper our response to the EGC Challenge 2017. Exploratory data analysis has first lead to understand the distribution of variables and detect strong correlations. We then defined two new variables combining dataset variables. Several classification algorithms have been experimented for the first task of the challenge. Performances have been evaluated by 10-fold cross validation. It has resulted in selecting the best unilabel and multilabel classifiers. On both unilabel and multilabel levels, the best classifier outperforms the reference scores by approximately 2%. We also explored the second task of the challenge. On one hand, association rules have been searched. On the other hand, the initial dataset has been enriched with domain knowledge such as climate data (rainfall, temperature, wind) or taxonomic data in the field of botany. Furthermore, geographical and cartographic data have been used in a visualisation tool for representing trees.