UNIVERSITE DE NANTES
LMJL
UNIVERSITE DE TOURS
CITERES-LAT

Rapport de stage

MASTER INGÉNIERIE STATISTIQUE

Sous le thème :

Utilisation de méthodes d'apprentissage supervisé pour prévoir la classe d'un individu en fonction de son profil céramique

---

Présenté par :

M. Barrion Michaël

Encadré par :

Mme Bellanger Lise

M. Husi Philippe

M. Coulon Arthur

Année universitaire 2023-2024

Sommaire

Introduction générale **3**

1 Contexte, problématique et objectifs **3**
1.1 Contexte . . . 3
1.2 Problématique et objectifs . . . 4

2 Prérequis archéologiques et travaux antérieurs **5**
2.1 Pré-requis archéologiques . . . 5
2.2 Travaux antérieurs . . . 6

3 Méthodologie **9**
3.1 Notations . . . 9
3.2 Méthodes des k plus proches voisins - kNN . . . 9
3.3 Arbres de décision (CART), forêts aléatoires et boosting . . . 11
3.3.1 Arbre de décision - CART . . . 11
3.3.2 Forêt aléatoire: . . . 13
3.3.3 Boosting (Boosted trees) : . . . 14
3.4 Méthodes paramétriques . . . 15
3.4.1 Régression logistique polytomique . . . 15
3.4.2 Analyse linéaire discriminante (LDA) . . . 16
3.4.3 Analyse quadratique disriminante (QDA) . . . 17
3.4.4 Support Vector Machine (SVM) . . . 17
3.5 Métriques . . . 18
3.6 Méthode de validation . . . 19

4 Résultats - Classification supervisée **20**
4.1 Application des méthodes de classification supervisée aux ensembles supplémentaires - Tunage des paramètres . . . 20
4.2 Stratégie de décision : apporter une réponse à l'utilisateur . . . 29
4.3 Résultats et discussion . . . 34

5 Limites et perspectives **47**
5.1 Difficultés, limites . . . 47
5.2 Perspectives . . . 51

Conclusion **52**

Remerciements **53**

Bibliographie **53**

Annexes **56**

Introduction générale

Le nombre de fouilles archéologiques, notamment avec le développement de l'archéologie préventive a énormément augmenté ces dernières années produisant un volume très important de données qui pour pouvoir être exploitées doivent être traitées à l'aide de méthodes adpatées. Parallèlement le développement de la micro informatique a mis à disposition d'importants moyens de calculs rendant possible par des méthodes de machine learning le traitement de données en particulier de données archéologiques. Effectué entre avril et août 2024 au sein du laboratoire de Mathématiques Jean Leray de l'Université de Nantes (LMJL, UMR 6629 CNRS, Nantes Université) et en collaboration avec l'équipe Laboratoire Archéologie et Terroires du laboratoire Cités, Territoires, Environnement (CITERES-LAT) de l'Université de Tours (CITERES-LAT UMR 7324 CNRS, Université de Tours), ce stage de Master 2 a eu pour objet la mise en oeuvre et la comparaison de différentes méthodes d'apprentissage supervisé appliquées à des données archéologiques issues du vaste projet de recherche en archéologie ModAThom.

On commencera par préciser le contexte et les objectifs de ce stage. Puis après avoir rappelé quelques concepts archéologiques clés, on exposera brièvement le travail conduit en amont du travail réalisé au cours de ce stage. On exposera les méthodes de classification supervisée utilisées, on analysera les résultats obtenus et enfin on se posera la question des limites et perspectives.

1 Contexte, problématique et objectifs

1.1 Contexte

[Husa],[Husb]
Le travail présenté ici s'inscrit dans le large projet ModAThom. ModAThom (MODèle explicatif de la fabrique urbaine d'Angkor Thom : archéologie d'une capitale disparue), est un projet d'étude archéologique de la ville de Angkor Thom, située dans l'actuel Cambodge. Ce projet qui regroupe plusieurs labaratoires, dont le laboratoire LAT-CITERES de l'Université de Tours, est cofinancé par l'Agence Nationale de la Recherche, a démarré en 2018. Angkor Thom, capitale du royaume Khmer entre le IX^{e} et le XVI^{e} siècle, fut une cité très importante de l'Asie du Sud-Est et constitue l'un des sites urbains les plus emblématiques de l'histoire mondiale de l'urbanisme. Ce projet qui est un projet d'archéologie urbaine, présente plusieurs volets parmi lesquels une remise en cause de la chronologie établie, la construction d'un modèle explicatif des transformations et évolutions de la ville qui s'appuiera sur l'analyse de la matérialité accumulée dans le sol urbain, à sa surface, sous la forme de données immobilières (architectures, îlots urbains, rues, canaux, bassins, etc.)

et mobilières (céramique, métal, os, etc.) et en particulier de la céramique khmer découverte, récoltée, inventoriée issue des nombreuses fouilles réalisées avec la Mission Archéologique Française d'Angkor Thom (MAFA) depuis 2009. Comme l'explique Philippe Husi, archéologue au laboratoire LAT-CITERES de l'Université de Tours et coordinateur du projet : "l'objectif du projet ModAThom est de construire un modèle explicatif de la formation et de la transformation d'Angkor Thom des conditions de sa naissance à son abandon (IX^{e}-XVI^{e} s.). Ce modèle, élaboré en grande partie à partir des sources archéologiques, aboutit à la remise en cause, au moins partielle, d'une chronologie existante, essentiellement fondée sur les études de la statuaire, les changements architecturaux et les dynasties des rois khmers relevant de l'épigraphie. Cette entrée archéologique permet de mieux percevoir l'organisation de la ville et sa relation avec le site d'Angkor, mais aussi dans la mesure du possible l'ouverture économique et culturelle de cette dernière sur le monde extérieur." Ce projet pluridisciplinaire comprend un volet statistique avec notamment l'emploi de méthodes de classification (supervisée et non supervisée) pour exploiter les nombreuses données recueillies.

1.2 Problématique et objectifs

Une collaboration engagée depuis de nombreuses années entre le Laboratoire de Mathématiques Jean Leray de Nantes et le Laboratoire Archéologie et Territoires a donné lieu notamment au développement d'un package R (SPARTAAS; [CBH21]). Une méthode de classification hiérarchique descendante (CAH) par compromis a été développée pour faciliter l'établissement de la chronologie et de la périodisation de sites archéologiques ([BCH21c]). Implémenté dans le package R SPARTAAS, elle permet de réaliser une partition à partir d'un compromis entre deux sources d'information. Cependant la qualité des données en archéologie nécessite un nettoyage important. Seul un petit nombre d'individus (moins de la moitié) ont été conservés pour être traités par la méthode de classification hiérarchique par compromis (désignée aussi sous le nom hclustcompro) pour s'assurer de la qualité de la classification ainsi réalisée. Se pose alors la question d'exploiter la grande quantité de données écartées. Il a été envisagé d'appliquer des méthodes de classification supervisée pour affecter une classe à ces individus écartés. Les objectifs de ce stage sont d'implémenter avec le logiciel R une série de méthodes de machine learning afin de réaliser une classification supervisée appliquée à des données archéologiques issues du projet ModAThom qui dans un premier temps ont été écartées en raison de leur qualité jugée insuffisante, d'étudier la fiabilité des résultats obtenus en comparant les différentes méthodes et en les confrontant aux avis d'un expert archéologue. Dans la perspective de développer un outil d'aide à la classification mis à la disposition de non statisticiens pour effectuer une tâche de classification supervisée d'individus d'un jeu de données

et en s'appuyant sur les résultats obtenus avec le jeu de données utilisé dans ce travail se poseront également les questions du choix des méthodes de classification à utiliser, de l'élaboration d'une stratégie pour fournir une réponse unique et de l'évaluation de la fiabilité de cette réponse.

2 Prérequis archéologiques et travaux antérieurs

2.1 Pré-requis archéologiques

Nous allons exposer brièvement quelques concepts fondamentaux d'archéologie utiles pour la compréhension du travail réalisé.
Stratification : La stratification est une méthodologie de fouilles importante et courante en archéologie et suit un protocole de relevés rigoureux et systématiques. Elle aboutit à l'identification de couches ou unités stratigraphiques qui se distinguent non seulement par leurs caractéristiques physiques et géologiques (essentiellement sédimentaires) mais aussi anthropiques. Cette identification est alors confirmée ou infirmée par l'étude du mobilier archéologique qu'elles contiennent. Une coupe stratigraphique met en évidence les différentes couches présentes qui sont autant de différents phénomènes liés à l'activité humaine.
Périodisation : Cela consiste à identifier des grands rythmes temporels de l'histoire d'un site archéologique, à étudier les temporalités distinctes (à l'aide de quantification céramique notamment) et finalement à établir une chronologie relative d'un espace. Pour établir cette périodisation, on utilise les unités stratigraphiques obtenues lors des fouilles.
Céramique : mot qui trouve son origine dans le mot grec "kearamos" qui signifie argile, le terme générique de céramique désignera l'ensemble des objets fabriqués en terre (argile) qui ont subi une transformation physico chimique irréversible au cours d'une cuisson à température plus ou moins élevée.

"La céramique est également considérée comme le premier "art du feu" à aparaître avant le travail du verre et du métal, à la fin de la préhistoire au Néolitihique. Utilitaire ou expression artistique, elle reflète les changements des modes de vie et témoigne des progrès techniques (maîtrise des quatre éléments naturels : la terre, l'eau, le feu et l'air). Elle restitue les coutumes, les habitudes alimentaires et les pratiques cultuelles d'un peuple à une époque donnée." (extrait de [Sèv])
Utilisée abondemment en tous temps et en tous lieux et pratiquement inaltérable, la céramique est omniprésente et on en retrouve partout des débris (tessons) en abondance. Elle constitue de ce fait une source d'information inestimable dans le domaine de l'archéologie et un instrument privilégié pour la datation. Elle fait l'objet d'un traitement méthodique et rigoureux (lavage, tri, inventaire, remontage), de l'élaboration d'une typologie (études des décors, formes, matériaux, techniques de

production) et entraîne la définition de groupes techniques. Un groupe technique regroupe les tessons qui vérifient un certain nombre de caractéristiques qui définissent ce groupe. Par exemple le groupe technique AT_TC_STS_1_5 utilisé dans notre jeu de données comprendra les tessons de céramiques en terre cuite sans traitement de surface, à pâte fine sableuse, rose, sans inclusion visible mais d'aspect doux et crayeux.
La céramique fait l'objet de différents modes de comptage : Nombre Minimum d'Individus, Nombre Typologique d'Individus, Nombre d'Individus par Forme, Nombre de Restes). Chacun d'entre eux a ses avantages et inconvénients. Le Nombre de Restes (NR) correspond au nombre de tessons retrouvés par groupe technique et c'est ce mode de comptage qui sera utilisé ici.
Les nombreux événements survenus au cours de temps d'origine naturelle (glissements de terrain par exemple) ou d'origine humaine (travaux, creusements, utilisation de matériaux prélevés dans un site et utilisés dans un autre etc) sont à l'origine de mélanges de tessons de céramique (phénomènes de redéposition) qui peuvent entraîner une pollution importante de certains ensembles stratigraphiques. Cette pollution rendra beaucoup plus difficile le positionnement chronologique relatif de l'ensemble pollué et sera à l'origine "d'erreurs" commises par les méthodes statistiques utilisées.

2.2 Travaux antérieurs

Ces travaux ont été réalisés par Lise Bellanger, Philippe Husi et Arthur Coulon. Les fouilles effectuées ont permis la constitution de 257 ensembles stratigraphiques. Ces ensembles ont été séparés en deux grandes catégories. D'une part 120 de ces ensembles ont été sélectionnés pour leur fiabilité chrono stratigraphique et la qualité des assemblages céramiques. Ils constituent ce qu'on appellera désormais les ensembles de référence. D'autre part les 137 autres ensembles ont été considérés moins fiables en raison d'assemblages céramiques perturbés. Ces derniers constituent les ensembles supplémentaires. Ces deux catégories d'ensemble vont faire l'objet d'un traitement statistique différent. Notons que la seule source mobilière utilisée est la céramique. Ainsi pour chacun des ensembles (de référence comme supplémentaires) une étude minutieuse a abouti au recensement des tessons de céramiques qui y ont été retrouvés et dont le nombre de restes (NR) a été classé par groupe technique parmi les 28 groupes techniques identifiés. Le résultat de ce recensement a donné lieu à deux tables de comptage (une première pour les ensembles de référence et une deuxième pour les ensembles supplémentaires). Une table de comptage est donc un tableau (T_{ij}) avec $1 \leq i \leq 120$ pour la première table, $1 \leq i \leq 137$ pour la 2ème table et $1 \leq j \leq 28$ pour les deux tables. La celluleT_{ij} contiendra le nombre de restes appartenant au groupe technique j retrouvés dans l'ensemble i.
Pour le pré-traitement de ces données d'origine céramique, il a d'abord été appliqué une

analyse factorielle des correspondances (AFC) à la table de comptage des ensembles de référence. Ceci afin d'obtenir une première visualisation de ces données qui permettra d'identifier certains phénomènes (effet Gutmann, présence de valeurs aberrantes) et de calculer les distances nécessaires à l'application de la méthode hclustcompro ([BCH21c]). A l'issue de cette AFC une première matrice D_1, matrice des distances, a été calculée à partir des quatre premiers axes résultants de l'AFC (57,6% de l'inertie totale). Le choix de quatre axes a été effectué après plusieurs essais.
La prise en compte des données d'origine stratigraphique a été faite à partir du calcul de D_2 matrice de dissimilarité qui traduit la relation sur/sous entre les ensembles : $D_{2ij} = 0$ si les ensembles i et j sont adjacents (c'est-à-dire en relation sur/sous) et $D_{2ij} = 1$ dans le cas contraire.

L'idée mise en oeuvre ici pour obtenir une périodisation du site est d'utiliser les ensembles de référence sur lesquels va être appliquée une méthode de classification non supervisée, la méthode hclustcompro pour déterminer les classes qui définiront cette périodisation. Cette méthode orginale a la particularité de prendre en compte deux sources de données. Ici, des données d'origine céramique et des données d'origine stratigraphique. Elle calcule d'abord la valeur d'un coefficient α qui détermine la proportion de chacune des deux matrices D_1 et D_2 qui intervient dans le calcul d'une matrice D_α.

$$D_\alpha = \alpha D_1 + (1 - \alpha) D_2$$

Le coefficient α est obtenu par minimisation du critère $|cor(D_\alpha^{coph}, D_1) - cor(D_\alpha^{coph}, D_2)|$ où D_α^{coph} désigne la matrice de distance cophénéntique obtenue à partir de la CAH réalisée sur D_α à α fixé. Puis elle effectue une classification hiérarchique ascendante sur cette matrice D_α (avec la méthode d'agrégation Ward2 retenue après plusieurs essais). La méthode hcluscompro a été appliquée à ces deux matrices. Elle a permis de calculer $\alpha = 0.55$, c'est-à-dire que l'information utilisée pour la classification est à 55% d'origine céramique et à 45% d'origine stratigraphique et a finalement permis la classification des ensembles de référence en 7 groupes (cf figure 1 et table 1).

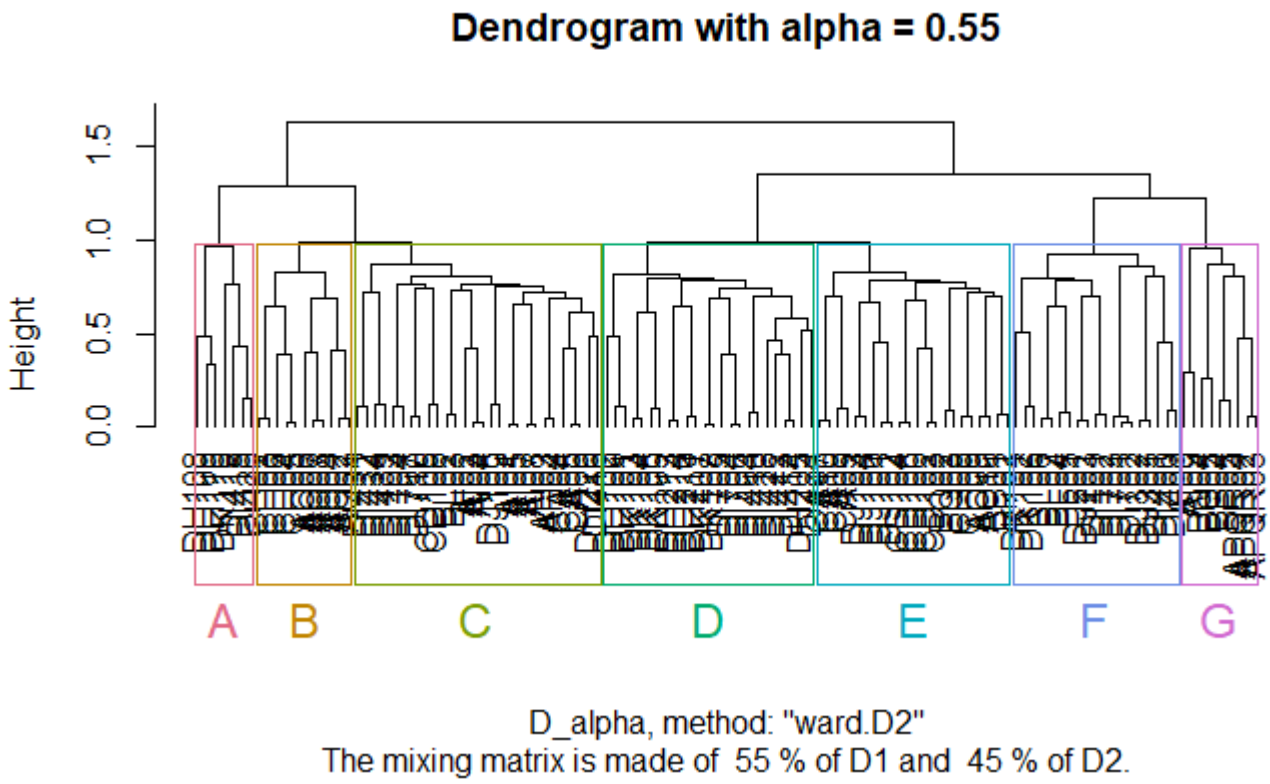

Figure 1: Dendogramme

Dans la suite, pour des raisons pratiques, les groupes seront désignés par les nombre de 1 à 7, 1 pour le groupe A, 2 pour le groupe B, etc.
De plus ces 7 groupes, résultats d"une démarche statistique, trouvent une interprétation archéologique cohérente. Des arguments archéologiques ont permis d'établir que la classe 3 regroupe les ensembles les plus anciens et représentent un premier faciès céramique, les classes 1 et 2 peuvent révéler un deuxième faciès, les classes 4 et 5 correpondent certainement à un troisième faciès, et les classes 6 et 7 semblent révéler un dernier grand faciès correspondant à la période la plus récente jusqu'à l'adandon du site. Ainsi certaines classes sont "proches" (1 et 2, 4 et 5, 6 et 7) mais les ensembles qui y ont été affectés par une méthode statistique peuvent également se distinguer par des arguments archéologiques plus fins rendant cette répartition en 7 groupes pertinente. Cependant les affectations à des groupes des ensembles supplémentaires par les différentes méthodes de classification supervisée devront ou pourront être interprétées à l'aune de ces proximités entre ces différents groupes.

Groupes	1	2	3	4	5	6	7
Effectifs	7	11	28	24	22	19	9

Table 1: Répartition des 120 ensembles de référence dans les 7 groupes

3 Méthodologie

La construction des sept classes obtenues et l'affectation des 120 ensembles supplémentaires à celles-ci constitue la première étape. Pour mettre en oeuvre la deuxième étape, l'étape de classement des 137 ensembles supplémentaires dans une des sept classes obtenues à l'issue de l'étape 1, nous avons utilisé différentes méthodes parmi les plus populaires de classification supervisée. De nature différente elles incluent des méthodes non paramétriques et des méthodes paramétriques. Dans cette partie, nous allons rappeler brièvement les principes de ces différentes méthodes.

3.1 Notations

Dans tout ce paragraphe, on considérera des individus notés $\mathbf{x_i}$ avec $i \in \mathbb{N}$, éléments d'un espace vectoriel à p dimensions, définis par leurs p composantes x_i^j avec $1 \leq j \leq p$, valeurs prises par les variables X^j pour un individu i, dites variables explicatives. De plus à chaque individu $\mathbf{x_i}$ correspond une classe y_i. On suppose disposer d'un ensemble d'apprentissage L constitué de n individus pour entraîner le modèle et d'un ensemble test T pour la validation du modèle.

3.2 Méthodes des k plus proches voisins - kNN

[HS04]
La méthode des k plus proches voisins (k Nearest Neighbors ou kNN) est une des méthodes les plus célèbres et les plus simples à mettre en oeuvre. Elle repose sur l'idée simple que des individus proches, au sens où la distance entre eux est faible, ont de bonnes chances d'appartenir à la même classe ou dit autrement qu'un individu à classer a raisonnablement de bonnes chances d'appartenir à la classe de la majorité de ses plus proches voisins. On ne considère parmi les distances possibles que les distances dites de Minkowski :

$$d_q(\mathbf{x_1}, \mathbf{x_2}) = (\sum_{i=1}^{p} |x_1^i - x_2^i|^q)^{\frac{1}{q}} \tag{3.1}$$

où $\mathbf{x_1}$ et $\mathbf{x_2}$ désignent des individus, p la dimension de l'espace des individus, q un nombre réel supérieur ou égal à 1.

Choisir la distance revient donc à choisir la valeur de l'exposant q, un nombre réel compris entre 1 et 2 (en effet pour $q < 1$, d_q n'est pas une distance et pour $q > 2$, les distances sont trop proches de d_2 pour qu'on ait intérêt à les différencier).
Une fois ces hyperparamètres déterminés, la classe d'un nouvel individu est donnée par la formule :

$$c(\mathbf{x}) = \underset{g\in\{1,...7\}}{\operatorname{argmax}} \sum_{i=1}^{k} \mathbf{1}_{y(i)=g} \tag{3.2}$$

Cette méthode présente plusieurs variantes. En particulier, il est raisonnable de penser que parmi les k voisins d'un nouvel individu à classer, la classe d'un voisin proche devrait compter davantage que celle d'un voisin plus éloigné dans la construction du vote majoritaire. C'est la méthode des k plus proches voisins pondérés, weighted kNN (wkNN). Dans cette méthode, la classe d'un nouvel individu n'est plus la classe simplement majoritaire, mais la classe gagnante d'un vote pour lequel chaque voix (la classe d'un des k plus proches voisins), se voit affecter un poids. Pour mettre en oeuvre cette variante de la méthode, on utilise une fonction noyau K qui doit vérifier les conditions suivantes :

- K(d)≥ 0

- K(d) atteint son maximum en d=0

- K(d) décroît de manière monotone pour d$\to \pm\infty$

De plus une fonction noyau nécessite soit une "largeur de fenêtre autorisée" si elle s'annule à une certaine distance du maximum, soit un paramètre de dispersion si K(d) est supérieur à 0 pour tout $d \in \mathbb{R}$. Dans la méthode wkNN, les deux sont sélectionnés automatiquement en fonction de la distance au premier voisin qui n'est pas pris en considération c'est-à-dire du k+1 ième voisin plus proche voisin $\mathbf{x_{(k+1)}}$. Cela est fait implicitement en standardisant toutes les distances par la distance du k+1 ième voisin :

$$D(\mathbf{x}) = \frac{d(\mathbf{x}, \mathbf{x_{(i)}})}{d(\mathbf{x}, \mathbf{x_{(k+1)}})}$$

Ainsi D ne prend que des valeurs comprises entre 0 et 1. De plus on ajoute une petite constante ϵ strictement positive à $d(\mathbf{x}, \mathbf{x_{(i)}})$ pour éviter d'avoir un poids égal à 0, ce qui pourrait arriver si un ou plusieurs des k voisins sont à la même distance que le k+1 ième. La classe d'un nouvel individu s'obtient alors par :

$$c(\mathbf{x}) = \underset{g\in\{1,...7\}}{\operatorname{argmax}} \sum_{i=1}^{k} K(D(\mathbf{x}, \mathbf{x_{(i)}}))\mathbf{1}_{y(i)=g} \tag{3.3}$$

Pour cette méthode, le noyau à utiliser fait partie des hyperparamètres. Remarquons que la méthode kNN peut être vue comme un cas particulier de wkNN (qui correspond au choix du noyau "rectangular").
Cette méthode selon la variante utilisée peut présenter jusqu'à 3 hyperparamètres à déterminer : le nombre k de voisins k (neighbors), l'exposant définissant la distance (dist_power) et le noyau de pondération K (weight_func).

Algorithme 1 wkNN

[HS04]

Require: $L = \{(y_i, \mathbf{x_i}), i = 1, ...N_L\}$ un ensemble d'apprentissage d'individus $\mathbf{x_i}$ appartenant à la classe y_i et $\mathbf{x}$ un nouvel individu dont on veut prédire la classe
- Trouver les k+1 plus proches voisins de $\mathbf{x}$
- Standardisation des distances via

$$D_{(i)}(\mathbf{x}) = \frac{d(\mathbf{x}, \mathbf{x_{(i)}})}{d(\mathbf{x}, \mathbf{x_{(k+1)}})}$$

- Transformation des distances normalisées en poids avec la fonction noyau K : $w_{(i)} = K(D_{(i)})$

Ensure: Calcul de la classe prédite pour $\mathbf{x}$ par $\hat{y} = \underset{g \in \{1,...7\}}{\operatorname{argmax}} \sum_{i=1}^{k} w_{(i)} \mathbf{1}_{y(i)=g}$

3.3 Arbres de décision (CART), forêts aléatoires et boosting

[Jam+13]

3.3.1 Arbre de décision - CART

[Bre+84]
La méthode "Arbre de décision" basée sur l'algorithme CART (Classification And Regresion Tree) consiste à partitionner par étapes l'espace des variables. Au terme de la procédure, on obtient une partition de l'espace des variables en plusieurs régions. Pour prédire la classe d'un nouvel individu, on déterminera la région à laquelle il appartient puis on lui affectera la classe majoritaire dans cette région, c'est-à-dire la classe la plus représentée parmi les classes des individus de l'ensemble d'entraînement qui se trouvent dans cette région.
Les différentes étapes de la partition peuvent être représentées par un arbre binaire. Les noeuds de l'arbre non terminaux ont chacun deux descendants (deux noeuds fils) et les noeuds terminaux (sans descendant) seront appelés feuilles. A la fin de la procédure, les feuilles représentent les régions de la partition.

Mais se pose la question de la taille de l'arbre. Si la taille est trop grande, il y a un risque de surapprentissage. Si au contraire elle est trop petite, la structure du jeu de données risque de ne pas être capturée.
L'algorithme se compose de deux étapes : une première étape durant laquelle on fait pousser le plus grand arbre possible (arbre maximal T_0) et une seconde étape dite d'élagage (pruning) durant laquelle on ajuste la taille de l'arbre en élaguant l'arbre maximal T_0.
A chaque étape de la partie de l'algorithme qui consiste à faire pousser un arbre, on divise un noeud en deux noeud fils. Pour ce faire, il faut déterminer une variable X^j et une valeur seuil s. Le noeud fils gauche contiendra alors les individus i tels que $x_i^j \leq s$ tandis que le noeud fils droit contiendra les individus tels que $x_i^j > s$. Le point crucial est le choix de la variable X^j à utiliser et de la valeur seuil s. Ce choix s'effectue de façon à minimiser une certaine fonction de coût :

$$f_{cout}(T, m) = N_{m_L} Q_{m_L}(T) + N_{m_R} Q_{m_R}(T)$$

où N_{m_L} et N_{m_R} désignent respectivement le nombre d'individus dans le fils gauche et le fils droit du noeud m et $Q_{m_L}(T)$ et $Q_{m_R}(T)$ sont des mesures d'impureté respectivement des fils gauche et droit du noeud m de l'arbre T.
Plusieurs choix sont possibles pour les mesures d'impureté. Le fréquent et classique index de Gini sera utilisé ici. Pour un noeud m représentant une région R_m avec N_m individus, soit $\hat{p}_{mk} = \frac{1}{N_m} \sum_{\mathbf{x_i} \in R_m} \mathbf{1}_{y_i = k}$ la proportion d'individus à la classe k au noeud m, l'index de Gini est défini par : $\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$ où K désigne le nombre total de classes.
Dans cette étape, on fait pousser un grand arbre T_0 en stoppant le processus seulement lorsqu'une certaine taille de noeud (c'est-à-dire un nombre minimum d'individus de l'ensemble d'entrainement sont présents dans le noeud) est atteinte. L'idée est d'obtenir des feuilles qui soient le plus homogène possible.
Pour l'étape d'élagage la stratégie adoptée va être de réduire la taille du grand arbre T_0 en utilisant le coût complexité.
On appelle "coût complexité" d'un arbre : $C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$ où $|T|$ désigne le nombre de feuilles de l'arbre T. On pénalise l'impureté de l'arbre par une fonction linéaire de nombre de feuilles de l'arbre.
L'idée est de trouver pour chaque α, le sous arbre $T_\alpha \subseteq T_0$ qui minimise $C_\alpha(T)$.
On peut montrer que pour chaque α il existe un unique arbre de taille minimale T_α qui minimise $C_\alpha(T)$. Pour le déterminer, on part de T_0, et on supprime le noeud

pour lequel l'augmentation de $\sum_{m=1}^{|T|} N_m Q_m(T)$ est la plus petite. On obtient un nouvel arbre sur lequel on réitère cette opération jusqu'à obtenir l'arbre "racine". On obtient ainsi une suite d'arbres et Breiman ([Bre+84] a montré que l'arbre cherché T_α est contenu dans cette suite.
Cette méthode présente trois hyperparamètres à déterminer : le coût complexité (cost_complexity), la profondeur de l'arbre (tree_depth), le nombre minimum d'indvidus requis dans un noeud pour effectuer une nouvelle division (min_n)

3.3.2 Forêt aléatoire:

[Bre01]
Les résultats donnés par un arbre de décision sont souvent décevants. De plus les arbres CART sont connus pour être relativement instables dans le sens où une petite modification du jeu d'entrainemet peut conduire à un arbre très différent. Les forêts aléatoires constituent un moyen d'améliorer significativement les résultats. L'idée consiste à faire pousser plusieurs arbres de décision selon la méthode CART. Chaque arbre prédira une classe et la prédiction de la forêt correspondra alors à la classe majoritaire (parmi les prédictions de tous les arbres de la forêt). Cette méthode se montre particulièrement performante lorsque les arbres sont aussi différents les uns des autres que possible afin que les erreurs des uns soient compensés par les autres. La méthode forêt aléatoire (Random Forest) utilisée présente deux caractéristiques importantes. Tout d'abord chaque arbre est entrainé avec des échantillons différents (par tirage au sort des individus mais avec répétitions possibles). Ensuite pour la construction de chaque arbre, la variable à utiliser pour chaque division d'un noeud est sélectionnée dans un sous ensemble tiré au hasard de l'ensemble des variables et de cardinal fixé (en classification ce nombre est souvent proche de $\sqrt{p}$.
Cette méthode présente trois hypermaramètres à déterminer : le nombre d'arbres de la forêt (trees), le cardinal du sous ensemble des variables dans lequel va être choisie au hasard celle utilisée pour une nouvelle division (mtry) et le nombre minimum requis d'individus pour procéder à une nouvelle division (min_n)

Algorithme 2 Forêt aléatoire

[HTF17]

Pour b=1 à B

1. tirer un échantillon taille N (avec remise et donc répétitions possibles) dans l'ensemble d'entraînement
2. faire pousser un arbre de la forêt T_b sur les données de cet échantillon en répétant récursivement les étapes suivantes pour les noeuds de l'arbre jusqu'à que la taille minimume soit atteinte
   (a) choisir au hasard m variables parmi les p
   (b) choisr parmi les m la variable qui permet la meileure division
   (c) diviser le noeud en 2 noeuds fils

Ensure: l'ensemble des arbres $\{T_b\}_b^B$.

Pour faire une prédicion pour un nouvel individu $\mathbf{x}$: $\hat{y} = \underset{g\in\{1,...7\}}{\text{argmax}} \sum_{i=1}^{B} T_b(\mathbf{x}) = g$

3.3.3 Boosting (Boosted trees) :

[FS+96], [CG16]

Le principe général et commun aux différents algorithmes de boosting existants consiste à construire itérativement un prédicteur fort à partir de plusieurs prédicteurs faibles (c'est-à-dire des prédicteurs qui font légèrement mieux que des prédictions faites purement au hasard). A chaque itération on prend en compte les performances du prédicteur fort courant et on se concentre sur les individus que celui-ci a mal classés en affectant des poids aux individus. Ainsi des individus mal classés à l'étape k recevront pour le calcul du prédicteur à l'étape k+1 des poids plus importants que les individus bien classés. Cette méthode peut s'appliquer à tous types de prédicteurs faibles (mais avec des performances différentes). Nous utiliserons ici comme prédicteurs faibles des arbres de décision qui se prêtent particulièrement bien à ce rôle. Plus précisément ([Sté15], le boosting est un algorithme itératif où l'estimateur $\hat{c}$ s'obtient à chaque étape m en minimsant l'expression $\hat{c}_m = \underset{c}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{C}_{m-1}(x_i) + c(x_i))$ avec L une fonction de perte, $\hat{C}_{m-1}(x_i) = \sum_{k=1}^{m-1} \alpha_k \hat{c}_k$ et c dans une famille donnée de fonctions (ici celle des arbres de décision).

On obtient à la fin de l'apprentissage l'estimateur $\hat{c} = \hat{C}_M = \sum_{k=1}^{m} \alpha_k \hat{c}_k$. On remplace souvent le modèle $\hat{C}_m = \hat{c}_{m-1} + \alpha_m \hat{c}_m$ par le modèle $\hat{C}_m = \hat{c}_{m-1} + \nu \alpha_m \hat{c}_m$ où ν compris

entre 0 et 1 est un taux d'apprentissage constant.
Notons que les méthodes de boosting ressemblent aux méthodes des forêts aléatoires en ce sens que dans les deux cas, on agrège plusieurs arbres de décision. Mais ces arbres sont indépendants dans le cas des forêts aléatoires et contrairement aux méthodes de boosting. Il existe plusieurs algorithme de boosting. C'est l'algorithme xgboost qui sera utilisé ici. Dans celui ci, il y a possiblement 8 hyperparamètres à tuner. Mais pour limiter le temps d'exécution, on ne tunera que 4 paramètres (les autres se verront affecter la valeur par défaut du packgage R utilisé).
Paramètres à déterminer : le nombre d'arbres (trees), le cardinal du sous ensemble des variables dans lequel va être choisie au hasard celle utilisée pour une nouvelle division (mtry), la taille minimale d'un noeud (min_n), le taux d'apprentissage (learn_rate)

3.4 Méthodes paramétriques

Dans le cas d'un problème de classification à K classes, on peut toujours partitionner l'espace des individus de l'ensemble d'entraînement en K parties, chaque partie étant associée à une classe. Pour réaliser une telle partition, il faut calculer des frontières de décision, c'est-à-dire des frontières qui délimitent les différentes régions. Dans bien des cas, on peut trouver des frontières de décision linéaires.
Certaines méthodes consistent à calculer des fonctions discriminantes f_k pour chaque classe k et à attribuer à un individu la classe k_0 pour laquelle la valeur $f_{k_0}(\mathbf{x})$ est la plus grande. D'autres à calculer les probabilités qu'un individu $\mathbf{x}$ appartienne à la classe k et à affecter l'individu à la classe pour laquelle il a la plus grande probabilité d'appartenance. Si ces fonctions discriminantes, ces probabilités ou une fonction monotone de celles ci sont linéaires alors on peut calculer des frontières de décision linéaires. Les méthodes paramétriques décrites ci-dessous entrent dans cette catégorie de méthodes (à l'exception l'analyse quadratique discriminante).

3.4.1 Régression logistique polytomique

[Kle+02]
On cherche à modéliser les probabilités $P(\mathbf{y_i} = g)$ avec $g \in \{1, ..., K\}$ où K est le nombre de classes différentes. On ne peut pas modéliser directement ces probabilités par une fonction linéaire des variables car alors il n'est pas garanti que la valeur obtenue soit comprise entre 0 et 1 comme doit l'être une probabilité. On utilise alors une fonction qui renvoie des valeurs dans l'intervalle [0,1]. Dans le cas de la régression logistique polytomique, l'approche consiste à se fixer une classe (par exemple la classe K) et à modéliser $log(\frac{P_g(\mathbf{x_i})}{P_K(\mathbf{x_i})})$ par une combinaison linéaire des variables : $log(\frac{P_g(\mathbf{x_i})}{P_K(\mathbf{x_i})}) = \beta_0^g + \sum_{j=1}^{p} \beta_g^j x_i^j$ où $g \in \{1, ..., K-1\}$ et $P_g(\mathbf{x_i})$ désigne la probabilité qu'à l'individu $\mathbf{x_i}$ d'appartenir à la classe g ($g \in \{1, ..., K\}$).

Il faut alors estimer les (p+1)(K-1) coefficients β_g^j (où p désigne le nombre de variables). On alors :

$$P(\mathbf{Y} = g|\mathbf{X} = x) = \frac{exp(\beta_g^0 + \beta_g^T x)}{1 + \sum_{g=1}^{K-1} exp(\beta_g^0 + \beta_g^T x)}, g \in \{1, ..., K-1\} \tag{3.4}$$

$$P(\mathbf{Y} = K|\mathbf{X} = x) = \frac{1}{1 + \sum_{g=1}^{K-1} + exp(\beta_g^0 \beta_g^T x)} \tag{3.5}$$

(puisque la somme des probabilités doit être égale à 1) où β_g est le vecteur des coefficients pour la classe g.
Aucun hyperparamètre à déterminer.

Régression logisitique pénalisée (Ridge, Lasso et Elastic net)
Le modèle de régresion logisitique polytomique admet des variantes qui consistent à pénaliser la norme du vecteur β des coefficients. On peut utiliser la norme L_2 (Ridge), la norme L_1(Lasso) où un "mélange" des 2 (Elastic net). Les coefficients sont calculés par maximum de vraisemblance en utilisant la vraisemblance conditionnelle de g sachant x et sont solutions du problème de maximisation $\underset{\beta_0,\beta}{max}(l(\beta_0, \beta) - \lambda||\beta||_i)$ où l désigne la fonction de vraisemblance, λ est un paramètre à déterminer, $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ et $||\beta||_2 = \sum_{j=1}^{p} \beta_j^2$. Dans le cas Elastic net le problème de maximisation devient $\underset{\beta_0,\beta}{max}(l(\beta_0, \beta) - \lambda(1-\alpha)||\beta||_2 + \alpha||\beta||_1))$
hyperparamètre à déterminer dans le cas Ridge et le cas Lasso : la quantité de pénalisation (penalty) λ.
Hyperparamètres à déterminer dans le cas Elastic net : la proportion de lasso (mixture) α et la quantité de pénalisation (penalty) λ.

3.4.2 Analyse linéaire discriminante (LDA)

Appelons π_g la probabilité a priori qu'un individu appartienne à la classe g, et f_g la fonction de densité de x pour un individu venant de la classe g. D'après le théorème de Bayes, on a $P(\mathbf{Y} = g|\mathbf{X} = x) = \frac{\pi_g f_g(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$ L'idée est d'affecter un nouvel individu à la classe pour laquelle la probabilité d'appartenance est la plus élevée. Mais plutôt que de calculer directement les probabilités $P(\mathbf{Y} = g|\mathbf{X} = x)$, on va chercher à estimer π_g et $f_g(\mathbf{x})$. En faisant l'hypothèse que les individus de chaque classe sont issus d'une distribution gaussienne multivariée et que les matrices de covariance Σ_g sont identiques pour toutes les classes, la classe à affecter à l'individu $\mathbf{x}$ est celle pour laquelle la fonction de score δ_g en x est la plus grande.

$$\delta_g(x) = x^T \Sigma^{-1} \mu_g - \frac{1}{2} \mu_g^T \Sigma^{-1} \mu_g + log \pi_g \tag{3.6}$$

Dans ce cas les frontières de décision $\delta_g(x) = \delta_{g'}(x)$ sont des équations linéaires en les variables X^j Le caractère linéaire des fonctions "frontières de décision" a été obtenu sous l'hypothèse que les données suivent une distribution normale avec une matrice de covariance commune pour chaque classe. Cela est rarement le cas dans la réalité. Mais même en l'absence de cette hypothèse strictement vérifiée, cette méthode donne très souvent d'excellents résultats.
Aucun hyperparamètre à déterminer.
On peut également utiliser une variante de la méthode LDA en pénalisant les coefficients avec une pénalité quadratique (Ridge)
Un paramètre à déterminer : la quantié de régularisation (penalty)

3.4.3 Analyse quadratique disriminante (QDA)

L'analyse discriminante quadratique est une alternative à l'analyse discriminantes linéaire. On fait toujours l'hypothèse que les individus sont distribués dans chaque classe selon une loi gaussienne multivariée mais on ne fait plus l'hypothèse d'une matrice de covariance commune à toutes les classes. La fonction score s'écrit alors:

$$\delta_g(f(x) = -\frac{1}{2}log|\Sigma_g| - \frac{1}{2}(x-\mu_g)^T\Sigma_g^{-1}(x-\mu_g) + log\pi_g \tag{3.7}$$

Les frontières de décision ainsi obtenues (d'équations $\delta_g(x) = \delta_{g'}(x)$) ne sont plus linéaires mais quadratiques en les variables.
Aucun hyperparamètre à déterminer

3.4.4 Support Vector Machine (SVM)

[VV+98]
Cette méthode est une méthode de classification binaire qui consiste à couper en 2 l'espace des variables à l'aide d'un hyperplan, chaque demi-espace représentant une classe. Cela suppose que les données du jeu d'entraînement soient bien séparables par un hyperplan. Dans ce cas, il existe une infinité d'hyperplans pouvant réaliser cette séparation. Pour obtenir un classifieur robuste on cherche l'hyperplan qui maximise la marge c'est-à-dire l'espace entre les individus des deux classes. Mais pour éviter les problèmes de surapprentissage on autorise certains individus à se trouver dans cet espace interclasse. Trouver cet hyperplan revient à trouver son équation c'est-à-dire les coefficients $\beta_0, \beta_1, ..\beta_p$ tels que l'équation de l'hyperplan s'écrive $\beta_0 + \sum_{j=1}^{p} \mathbf{X_j} = 0$.
Le problème s'écrit alors :
$Maximiser M$
$\beta_0,\beta_1,...,\beta_p,\epsilon_1,...,\epsilon_n$
sous les contraintes $\sum_{j=1}^{p} \beta_j^2 = 1$
$y_i(\beta_0 + \beta_1 x_i^1 + ... + \beta_p x_i^p \geq M(1-\epsilon_i)$ pour tout $i \in \{1, ..., n\}$

avec $\epsilon_i \geq 0$, $\sum_{i=1}^{n} \epsilon_i \leq C$ où C est un nombre positif représentant le coût, et M la largeur de la marge. Lorsqu'il n'existe pas de frontière linéaire, l'idée de cette méthode est de plonger les individus dans un espace de plus grande dimension en utilisant une fonction noyau dans lequel ils pourront être séparés par une frontière linéaire. La méthode SVM peut également être utilisée dans le cas où le nombre de classes K est plus grand que 2. Dans ce cas une approche est d'ajuster K modèles SVMs en comparant à chaque fois une des K classes avec les K-1 restantes. Si on appelle $\beta_{0g}, \beta_{1g}, ..., \beta pg$ les coefficients le modèle SVM qui compare la classe g (codée par 1) aux autres codées (par -1), on affecte à un nouvel individu **x**la classe g pour laquelle la quantité $\beta_{0g} + \beta_{1g}x^1 + ... + \beta_{pg}x^p$ est la plus grande.
Les hyperparamètres à déterminer sont le coût (cost) et la marge (margin)

3.5 Métriques

Lorsqu'on utilise une méthode de classification supervisée, il est bien sûr indispensable de mesurer ses performances pour pouvoir comparer diffférentes méthodes entre elles et se faire une idée de la confiance que l'on pourra accorder aux prédictions de notre modèle appliqué à de nouveaux individus.

Il existe de nombreuses "métriques" pour effectuer cette mesure. Dans ce travail nous avons retenu deux métriques parmi les plus répandues : la précision (accuracy) et la ROC-AUC.

- La précision correspond simplement aux taux de classements exacts calculé sur les données de l'ensemble test :

$$\frac{\text{Nombre d'individus de l'ensemble test ayant reçu une prédiction correcte}}{\text{Nombre total d'individus de l'ensemble test}}$$

La précision est donc un nombre réel compris entre 0 et 1. Plus elle sera proche de 1, meilleure sera la performance de la méthode.

- ROC-AUC : [AB94] [BD06]

On commence par tracer la courbe ROC (pour Receiver Operating Characteristic) qui constitue une bonne mesure de performance d'une méthode de classification *binaire* (c'est-à-dire à 2 classes que l'on notera +1 et -1).
On appellera :

- "vrai positif", un individu issu de la classe +1 correctement classé par la méthode
- "faux positif", un individu issu de la classe -1 incorrectement classé dans la classe +1 par la méthode
- "vrai négatif", un individu issu de la classe -1 correctement classé par la méthode

- "faux négatif", un individu issu de la classe +1 incorrectement classé dans la classe -1 par la méthode.

On désignera par VP, le nombre de vrais positifs, FP le nombre de faux positifs, VN le nombre de vrais négatifs et FN le nombre de faux négatifs.
Dans un modèle de classification supervisée binaire, on se fixe un seuil s (nombre réel compris entre 0 et 1) et on affecte un individu à la classe +1 dès que la fonction discriminante évaluée pour ce nouvel individu (souvent sa probabilité d'appartenance à la classe 1) dépasse le seuil s. On fait alors varier s de 0 à 1 et on construit le point de coordonnées (1-specificity,Sensitivity) c'est-à-dire $(\frac{FP(s)}{VN(s)+FP(s)}, \frac{VP(s)}{VP(s)+FN(s)})$ où les différentes quantités sont calculées sur l'ensemble test. L'ensemble des points ainsi obtenus constitue la courbe ROC de la méthode. On appelle alors ROC-AUC (pour Area Under the Curve) l'aire sous la courbe. Plus cette aire est proche de 1, meilleure est la méthode. Pour calculer la ROC-AUC dans le cas d'un problème de classification supervisée à K classes, K strictement supérieur à 1 (7 classes pour notre étude), on calcule pour chaque k compris entre 1 et K la ROC-AUC du test binaire où la classe +1 correspond à la classe k et la classe -1 correspond à toutes les classes autres que k et on effectue la moyenne.

3.6 Méthode de validation

Nous disposons d'un jeu de données constitué de 120 ensembles de référence (les individus). Ce dernier a été scindé en un jeu d'entraînement de 90 ensembles choisis au hasard parmi les 120 pour entraîner les modèles de classification supervisée. Les 30 ensembles restants constituent le jeu de données test destiné à évaluer la performance des modèles ainsi entaînés. La plupart des méthodes de classification supervisée utilisée nécessite la détermination d'hyperparamètres (tunage des paramètres). Nous avons procédé pour cela par validation croisée à "2 folds". Ce nombre de folds peut apparaître faible mais il s'est imposé à cause de la petite taille du jeu de données et du faible nombre d'ensembles (ie d'indvidus) du jeu d'entrainement dans certaines classes. Avec un plus grand nombre de folds il y avait de fortes chances que les effectifs de certaines classes dans ces folds soient très faibles (voire nuls) ce qui aurait entraîné des erreurs pour certains calculs lors du tunage des paramètres. Pour compenser le très faible nombre de folds et introduire de la variabilité qui rende le tunage des paramètres plus fiable, l'opération a été répétée plusieurs fois. Le jeu de données d'entraînement a été scindé en deux folds de 45 ensembles chacun choisis au hasard et cette opération a été répétée 5 fois. Nous disposons donc de 10 cas de figure par méthode. On verra cependant que pour une des méthodes utilisée il a fallu se contenter de 7 cas de figure.

4 Résultats - Classification supervisée

Remarque : à l'exception de la figure 2 tous les graphiques présentés dans cette partie répresentent pour chacune des quatre méthodes (une couleur par méthode), les probabilités d'affectation (ordonnées) à chacune des 7 classes (abscisses). En abscisse, les classes ont été représentées dans l'ordre chronologique, la classe 3 contenant les ensembles les plus anciens et la classe 7 les ensembles les plus récents).

4.1 Application des méthodes de classification supervisée aux ensembles supplémentaires - Tunage des paramètres

On a utilisé les packages Tidymodels et Tidyverse de R développés par Hadley Wickham et son équipe ([Wic14]). Tidymodels est en quelque sorte un meta package de R (au sens où il utilise un ensemble d'autres packages) particulièrement adapté à l'utilisation de modèles de machine learning.

On a appliqué un certain nombre de méthodes de classification supervisée parmi les plus répandues (avec parfois des variantes ne se différenciant que par le nombre d'hyperparamètres) et prises en charge par le package Tidymodels. Pour chacune des méthodes utilisées, il a fallu déterminer les hyperparamètres (tunage des paramètres) lorsqu'il y en avait et estimer les paramètres (dans le cas des méthodes paramétriques). On a utilisé pour cela le jeu de données (divisé en jeu d'entraînement et jeu test cf paragraphe 3.6) constitué des ensembles de référence. Ce jeu de données se compose de 120 individus (les ensembles de référence) et de 27 variables quantitatives qui correspondent aux 27 axes de l'AFC réalisée lors de la première étape. Chaque donnée (i,j) est la coordonnée de l'ensemble i sur l'axe j. La variable réponse correspond à la classe.

1 - kNN, k plus proches voisins

- package R : kknn
- paramètres fixés :
  1. Distance (exposant) : 2
  2. : Noyau : rectangular (pas de pondération)
- paramètre tuné :
  1. Nombre de voisins : 5
- grille pour le tunage : les nombres entiers de 1 à 20
- temps de tunage pour cette grille : 14 s

- mesures de performance :
    1. Accuracy train : 0.7778 (70/90)
    2. ROC_AUC train : 0.940
    3. Accuracy test : 0.5333 (16/30)
    4. ROC_AUC test : 0.863

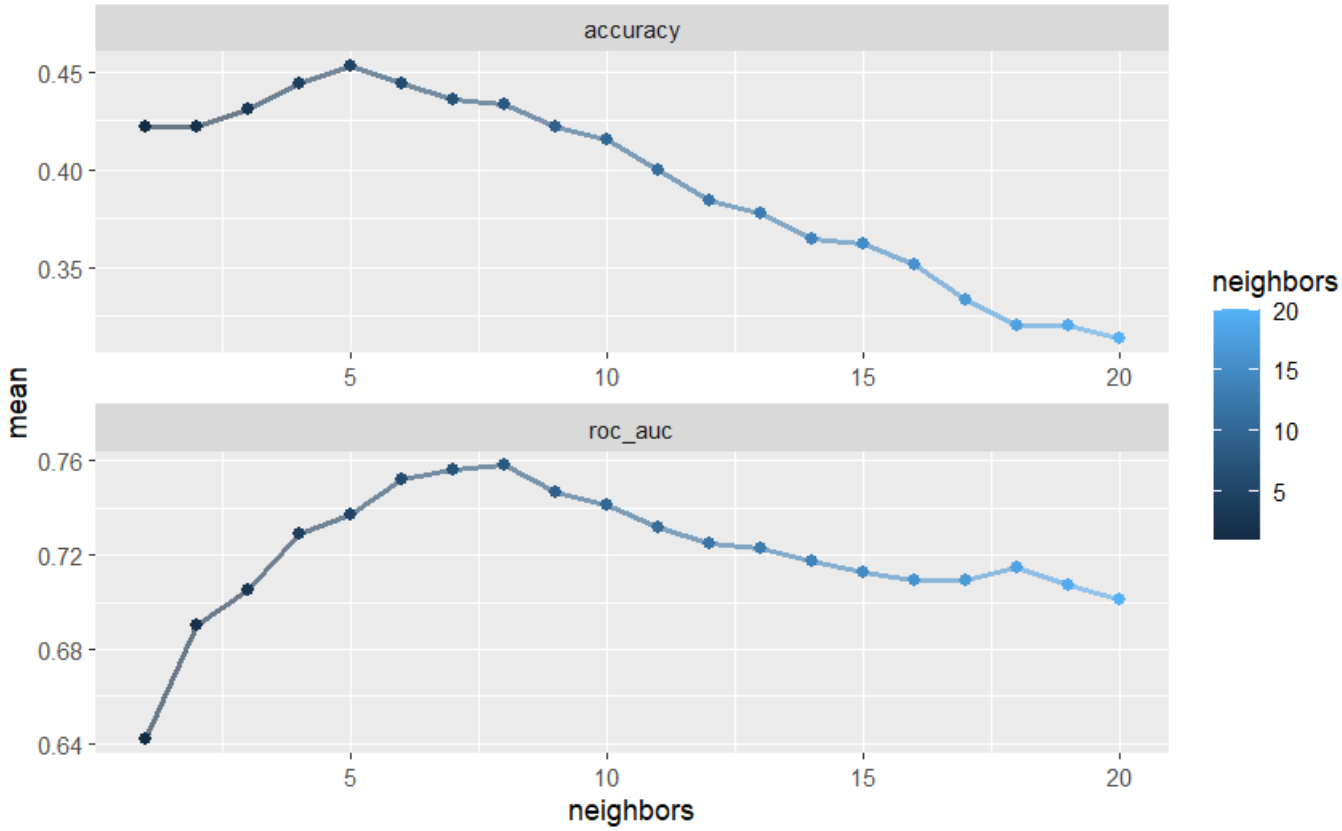

Figure 2: Résultats tuning

2 - kNNd, k plus proches voisins avec tunage de la distance en plus

- package R : kknn
- paramètre fixé :
    1. : Noyau : rectangular (pas de pondération)

- paramètres tunés :
    1. Nombre de voisins : 4
    2. Distance (exposant) : 1.56
- grille pour le tunage : 100 couples (nombre de voisins, exposant)
    1. Nombre de voisins : les nombres entiers de 1 à 10
    2. Exposant : 10 nombres décimaux entre 1 et 2
- temps de tunage pour cette grille : 1.2 min
- mesures de performance :
    1. Accuracy train : 0.8222 (74/90)
    2. ROC_AUC train : 0.948
    3. Accuracy test : 0.5667 (17/30)
    4. ROC_AUC test : 0.836

3 - wkNN, k plus proches voisins avec noyau de pondération

- package R : kknn
- paramètre fixé :
    1. : Distance (exposant) : 2
- paramètres tunés :
    1. Nombre de voisins : 9
    2. Noyau (fonction de pondération) : optimal
- grille pour le tunage : 100 couples (nombre de voisins, noyau)
    1. Nombre de voisins : les nombres entiers de 1 à 10
    2. Noyau(fonction de pondération) : les 10 noyaux proposés par le package R
- temps de tunage pour cette grille : 1.1 min
- mesures de performance :
    1. Accuracy train : 0.8889 (80/90)
    2. ROC_AUC train : 0.994
    3. Accuracy test : 0.5 (15/30)

4. ROC_AUC test : 0.873

4 - wkNNd, k plus proches voisins avec noyau de pondération et distance à tuner

- package R : kknn
- paramètre fixé : aucun
- paramètres tunés :
  1. Nombre de voisins : 8
  2. Noyau (fonction de pondération) : rank
  3. Distance (exposant) : 1
- grille pour le tunage : 1000 triplets (nombre de voisins, noyau,exposant)
  1. Nombre de voisins : les nombres entiers de 1 à 10
  2. Noyau(fonction de pondération) : les 10 noyaux proposés par le package R
  3. : Exposant : 10 nombres décimaux entre 1 et 2
- temps de tunage pour cette grille : 11 min
- mesures de performance :
  1. Accuracy train : 0.7778 (70/90)
  2. ROC_AUC train : 0.975
  3. Accuracy test : 0.5667 (17/30)
  4. ROC_AUC test : 0.884

5 - Arbre de décision (CART)

- package R : rpart
- paramètre fixé :
  1. Nombre minimum dans un noeud pour effectuer une nouvelle division, min_n : 2
- paramètres tunés :
  1. Coût complexité (cost_complexity) : 1e-10
  2. Profondeur de l'arbre (tree_depth) : 4
- grille pour le tunage : 25 couples (cost_complexity,tree_depth)

1. Coût_complexité : 5 nombres entre 1e-10 et 1
2. Profondeur de l'arbre : 5 entiers entre 1 et 15

- temps de tunage pour cette grille :20 s
- mesures de performance :

1. Accuracy train : 0.7 (63/90)
2. ROC_AUC train : 0.888
3. Accuracy test : 0.6 (18/30)
4. ROC_AUC test : 0.702

6 - Forêt aléatoire 1

- package R : ranger
- paramètre fixé :

1. Nombre d'arbres : 1000

- paramètres tunés :

1. Cardinal du sous ensemble de variables à considérer à chaque division, mtry : 19
2. Nombre minimum requis d'observations à chaque noeud pour être divisé, min_n : 2

- grille pour le tunage : 25 couples (mtry,min_n) gérés par le package R
- temps de tunage pour cette grille :1.5 min
- mesures de performance :

1. Accuracy train : 1 (90/90)
2. ROC_AUC train : 1
3. Accuracy test : 0.7667 (23/30)
4. ROC_AUC test : 0.919

7 - Forêt aléatoire 2

- package R : ranger
- paramètre fixé : aucun

- paramètres tunés :
    1. Nombre d'arbres (trees) : 1570
    2. Cardinal du sous ensemble de variables à considérer à chaque division, mtry : 7
    3. Nombre minimum requis d'observations à chaque noeud pour être divisé, min_n : 2
- grille pour le tunage : 64/125 triplets d'entiers (mtry,min_n,trees) gérés par le package R
- temps de tunage pour cette grille :2.7 min
- mesures de performance :
    1. Accuracy train : 1 (90/90)
    2. ROC_AUC train : 1
    3. Accuracy test : 0.8 (24/30)
    4. ROC_AUC test : 0.930

8 - Boosting

- package R : xgboost
- paramètres fixés :
    1. profondeur maximale d'un arbre (tree_depth) : 6
    2. loss_reduction : 0
    3. proportion d'observations échantillonnées (sample_size) : 1
    4. Nombre d'itérations sans amélioration avant arrêt (stop_iter) : inf
- paramètres tunés :
    1. Cardinal du sous ensemble de variables à considérer à chaque division, mtry : 11
    2. Nombre d'arbres (trees) : 1828
    3. Nombre minimum requis d'observations à chaque noeud pour être divisé, min_n : 4
    4. taux d'apprentissage (learn_rate) : 0.00426
- grille pour le tunage : 100 quadruplets (mtry,trees,min_n,learn_rate) gérés par le package R

- temps de tunage pour cette grille :6.7 min
- mesures de performance :
    1. Accuracy train : 0.9778 (88/90)
    2. ROC_AUC train : 1
    3. Accuracy test : 0.7 (21/30)
    4. ROC_AUC test : 0.886

9 - Régression logistique polytomique

- package R : nnet
- paramètre fixé : aucun
- paramètre tuné : aucun
- mesures de performance :
    1. Accuracy train : 1 (90/90)
    2. ROC_AUC train : 1
    3. Accuracy test : 0.5667 (17/30)
    4. ROC_AUC test : 0.916

10 - Régression logistique polytomique ridge

- package R : nnet
- paramètre fixé : aucun
- paramètre tuné :
    1. Quantité totale de régularisation (penalty) : 0.1
- Grille pour le tunage : 30 nombres décimaux entre 1e-4 et 1e-1
- Temps de tunage pour cette grille : 25 s
- mesures de performance :
    1. Accuracy train : 0.911 (82/90)
    2. ROC_AUC train : 0.992
    3. Accuracy test : 0.7667 (23/30)
    4. ROC_AUC test : 0.953

11 - Régression logistique polytomique Elastic net

- package R : glmnet
- paramètre fixé : aucun
- paramètres tunés
  1. Quantité totale de régularisation (penalty) : 0.000464
  2. proportion de lasso (mixture) : 1
- Grille pour le tunage : 100 couples de nombres décimaux (penalty,mixture)
  1. penalty : 10 nombres décimaux entre 1e-10 et 1
  2. mixture : 10 nombres décimaux entre 0 et 1
- Temps de tunage pour cette grille : 42 s
- mesures de performance :
  1. Accuracy train : 1 (90/90)
  2. ROC_AUC train : 1
  3. Accuracy test : 0.6333 (19/30)
  4. ROC_AUC test : 0.925

Remarque : Le nombre trop faible d'individus (ensembles) dans certaines classes pour certains fold a entraîné la génération d'erreurs par le package. Les calculs n'ont pu être effectués que sur 7 "modèles".

12 - Analyse linéaire discriminante (lda)

- package R : MASS
- paramètre fixé : aucun
- paramètre tuné : aucun
- mesures de performance :
  1. Accuracy train : 0.8667 (78/90)
  2. ROC_AUC train : 0.990
  3. Accuracy test : 0.6667 (20/30)
  4. ROC_AUC test : 0.925

Remarque : Le package MASS utilisé a renvoyé des erreurs : il semble que des valeurs trop proches de 0 rendaient l'inversion de matrices impossible. Aussi a t-il fallu conserver seulement 26 variables pour que cette méthode fonctionne sur ce jeu de données. L'analyse linéaie discriminante a donc été effectuée avec seulement 26 variables explicatives, les 26 premiers axes de l'AFC.

13 - Analyse linéaire discriminante avec pénalité (ldap)

- package R : mda
- paramètre fixé : aucun
- paramètre tuné :
    1. Quantité régularisation (penalty) : 1
- Grille pour le tunage : 20 nombres décimaux entre 1e-10 et 1
- Temps de tunage pour cette grille : 25 s
- mesures de performance :
    1. Accuracy train : 0.867 (78/90)
    2. ROC_AUC train : 0.988
    3. Accuracy test : 0.7333 (22/30)
    4. ROC_AUC test : 0.945

Remarque : comme pour la méthde précédente, l'analyse linéaire discriminante avec pénalité a été réalisée avec seulement 26 variables explicatives.

Le temps d'exécution du tunage des paramètres est un élément qui a été pris en compte. Avec les méthodes testées, il est de quelques secondes à 11 minutes, ce qui apparaît raisonnable.

Puis on a appliqué les méthodes avec les paramètres ainsi déterminés aux 137 ensembles supplémentaires afin d'obtenir pour chacun d'entre eux une prédiction de classe d'affectation. Le jeu de données utilisé ici se compose d'un tableau de 137 lignes et 27 colonnes. Chaque individu correspond à un ensemble supplémentaire. Chaque variable correspond à un axe obtenu lors de l'AFC réalisée sur les ensembles de référence lors du prétraitement de la première étape. Les données correspondent donc aux coordonnées des projections sur ces axes des ensembles supplémentaires. Pour les méthodes 12 et 13 il a fallu supprimer la dernière variable.

Pour chaque méthode testée, les packages R utilisés renvoient sous la forme d'un tableau (type tibble) la probabilité qu'a un ensemble d'appartenir à chaque classe ainsi que sa prédiction d'affectation qui correspond à la classe dont la probabilité d'appartenance est la plus grande.

Les méthodes de classification supervisée ont été appliquées pour obtenir une prédiction à la totalité des ensembles supplémentaires soit 137 ensembles. Mais le nombre de restes (NR) varie énormement (de 0 à 6612) d'un ensemble à l'autre. De plus les résultats pour des ensembles avec un nombre de restes trop faible n'ont pas de sens. Par exemple, les classes affectées par les différentes méthodes à l'ensemble CD204 pour lequel le nombre de restes vaut 0 n'a bien sûr aucun sens ! Il a été considéré que les résultats pour des ensembles avec un nombre de restes inférieur à 10 n'était pas interprétable. On a donc écarté les 28 ensembles dont le nombre de restes est inférieur à 10. La suite de l'étude porte sur les 109 ensembles restants pour lesquels le nombre de restes est supérieur à 10.

4.2 Stratégie de décision : apporter une réponse à l'utilisateur

On rappelle qu'à l'issue de l'étape de classification non supervisée, les 120 ensembles de référence ont été classifiés dans 7 classes dont certaines sont assez proches du point de vue de leur interprétation archéologique.

On peut constater (cf table 1), que certains groupes (groupe 1, groupe 7) ont des effectifs faibles. Le petit nombre d'éléments présents dans les groupes est une difficulté pour appliquer certaines méthodes dont il devra être tenu compte.

La table 2 montre les effectifs par classe pour chacune des méthodes de classification supervisée appliquée aux 120 ensembles de référence, ainsi que l'accuracy, calculée elle sur l'ensemble test. Sans surprise, les différentes méthodes montrent des performances (mesurées ici par l'acuracy) assez disparates. Certaines méthodes (les différentes variantes de la méthode des plus proches voisins en particulier) se montrent peu performantes avec une accuracy proche de 0.5. D'autre part on peut observer des variations dans les effectifs des classes assez importantes selon les méthodes. Ainsi les effectifs prédits de la classe 2 (qui contient réellement 11 ensembles) varie de 1 à 20 ! De plus, il faut garder à l'esprit que seuls les effectifs sont observés ici et qu'une coincidence parfaite entre l'effectif prédit par une méthode et l'effectif réel ne garantit pas nécessairement une bonne performance de la méthode (même si cela est certainement plus probable qu'en cas de différence importante). En effet, par exemple, la méthode boosting affecte conformément à la réalité 7 ensembles au groupe 1 mais on ne précise pas ici si ce sont bien les 7 ensembles réellement étiquetés classe 1.

D'autre part, il est notable que la méthode CART ne reconnaisse pas la classe 7, puisqu'elle n'y affecte aucun ensemble !

Méthode	GR 1	GR 2	GR 3	Gr 4	GR 5	GR 6	Gr 7	Accuracy
kNN	5	11	30	30	22	17	5	0.5333
kNNd	7	7	29	31	20	18	8	0.5667
wkNN	8	11	23	30	22	20	6	0.5
wkNNd	6	13	30	32	20	15	4	0.5667
Arbre(CART)	10	20	24	29	17	20	0	0.6
Forêt aléatoire 1	7	12	29	24	21	18	9	0.7667
Forêt aléatoire 2	7	12	29	24	22	18	8	0.8
boosting	7	11	29	24	22	19	8	0.7
Reg log	10	12	26	22	23	17	10	0.5667
Reg log ridge	5	11	29	25	24	18	8	0.7667
Reg log Elastic net	9	1	26	24	22	16	9	0.6333
LDA	6	14	29	22	27	15	7	0.6667
LDA pénalisée	6	14	29	23	26	16	6	0.7333

Table 2: Répartition des 120 ensembles de référence par groupe selon les méthodes

Le choix de la ou des méthodes à retenir pour affecter des ensembles supplémentaires à une classe est donc crucial dans la perspective de fournir un outil fiable d'aide à la classification.
Au regard de la nature (méthode paramétrique/non paramétrique) et surtout de leurs performances (accuracy en premier lieu et ROC-AUC), nous avons fait le choix de ne conserver que quelques unes des méthodes testées. Il nous a semblé trop hasardeux de ne retenir qu'une seule méthode. En effet, des résultats identiques obtenus avec différentes méthodes peut sembler fournir un critère de fiabilité pour une prévision. A contrario des résultats différents pour différentes méthodes peut constituer une alerte pour l'utilisateur et une incitation à approfondir les investigations et à utiliser des arguments autres que statistiques (ie arguments archéologiques ici). Mais, lorsqu'on applique les 13 méthodes testées aux 109 ensembles supplémentaires dont le nombre de restes (NR) est supérieur à 10, seuls 21.1% de ces ensembles reçoivent une seule et même prédiction (cf table 3). D'autre part pour un petit pourcentage d'entre eux, le nombre de prédictions peut aller jusqu'à 5 ! Cette possible grande variété de prédictions pour certains ensembles supplémentaires pose un problème dans le choix de la réponse à apporter à l'utilisateur de l'outil statistique. Pourtant plusieurs méthodes sont très proches mais certaines ont également une accuracy faible traduisant des mauvaises performances.

Nombre de prédictions différentes	1	2	3	4	5	6	7
Nombre d'ensembles (NR<10)	23	30	32	22	2	0	0
Pourcentage d'ensembles (NR<10)	21.1	27.5	29.4	20.2	1.8	0	0

Table 3: Résultats pour les 13 méthodes testées

Finalement en se fixant comme critère de ne retenir que les méthodes dont l'accuracy est supérieure à 0.7, nous sommes conduits à ne garder que 4 méthodes : forêt aléatoire 2, régression logistique pénalisée, lda pénalisée, boosting. Ces dernières présentent en outre l'avantage d'avoir une ROC-AUC supérieure à 0.9 (à l'exception de boosting dont la ROC-AUC est légèrement inférieure à 0.9) et d'être de nature différente (deux méthodes paramétriques et deux méthodes non paramétriques).

Méthode	Accuracy test	ROC-AUC test
Forêt aléatoire 2	0.8	0.93
Régression logistique pénalisée	0.7667	0.953
LDA pénalisée	0.7333	0.945
boosting	0.7	0.886

Table 4: Les méthodes retenues

Plusieurs cas de figure peuvent alors se présenter :

- cas 1 : on obtient une seule prédiction, les quatre méthodes retenues fournissent le même résultat

- cas 2 : on obtient deux prédictions différentes, soit un des résultats est fourni par trois méthodes et l'autre résultat par une seule (cas 2a), soit chacun des deux résultats est fourni par deux méthodes différentes (cas 2b)

- cas 3 : on obtient trois prédictions différentes, un des résultats est fourni par deux méthodes et chacun des deux autres par une seule méthode

- cas 4 : on obtient quatre prédictions différentes : chacune des quatre méthodes retenues fournit un résultat différent

Le pourcentage des ensembles parmi les 109 ensembles supplémentaires conservés (NR>10) qui ne reçoivent qu'une seule prédiction des quatre méthodes retenues atteint alors 50.5% tandis que celui des ensembles qui reçoivent deux prédictions différentes s'élève lui à 44% (soit un total significatif de 94.5% pour une ou deux réponses). Seuls 5.5% des 109 ensembles supplémentaires reçoivent trois réponses différentes et aucun ne reçoit quatre réponses différentes (cf table 5). De plus il ne faut pas perdre de vue que ces ensembles ont été sélectionnés parce qu'ils avaient été jugés peu fiables sur la base d'arguments archéologiques, que certains d'entre eux sont potentiellement très perturbés et que certaines des classes sont proches. Aussi que 94.5% des ensembles supplémentaires reçoivent une seule ou seulement deux prédictions différentes constitue un résultat intéressant.

Nombre de prédictions différentes	1	2	3	4
Nombre d'ensembles (NR<10)	55	48	6	0
Pourcentages d'ensembles (NR<10)	50.5	44	5.5	0

Table 5: Résultats pour les 4 méthodes retenues

L'objectif étant de fournir un outil statistique d'aide à la classification et donc de fournir une réponse à l'utilisateur, il reste à déterminer un mode de décision pour apporter une seule réponse c'est-à-dire une classe dans laquelle affecter chaque ensemble pour lequel on souhaite une prédiction.
Nous avons fait le choix fournir comme résultat final la "classe majoritaire". Plus précisément :

- cas 1 : on retient l'unique prévision
- cas 2a : on retient la classe prédite par trois des quatre méthodes
- cas 2b : on retient parmi les deux classes prédites (chacune par deux méthodes) celle qui a reçu la plus grande probabilité d'appartenance
- cas 3 : on retient la classe prédite par deux des quatre méthodes
- cas 4 : absent ici. Sinon le choix aurait été le même que pour le cas 2b

On espère en utilisant ainsi un panel de plusieurs méthodes relativement performantes en terme d'accuracy consolider la fiabilitié de la prédiction. Rappelons que chacun des modèles utilisés calcule en fait la probabilité qu'a un ensemble donné d'appartenir à l'une des 7 classes et lui attribue la classe qui présente la plus grande probabilité d'appartenance. Il est notable que les différentes méthodes se comportent différemment. Ainsi la méthode lda pénalisée affecte un grand nombre d'ensembles avec une probabilité très forte. Les résultats de cette méthodes sont souvent tranchés. Au contraire la méthode forêt aléatoire2 (et dans une moindre mesure la méthode boosting) donne souvent des probabilités d'affectation plus faibles et proches les unes des autres. Ces méthodes sont plus hésitantes. Parfois les différences entre les probabilités d'appartenir à différentes classes sont faibles et l'affectation finalement rendue par la méthode se joue à peu de choses. Ainsi la méthode lda pénalisée affecte 85 des 109 ensembles supplémentaires (soit dans presque 78% des cas) à une classe avec une probabilité d'appartenance supérieure ou égale à 0.75 ; alors que la méthode forêt aléatoire 2 affecte 44 des 109 ensembles supplémentaires (soit dans 40% des cas) avec une probabilité d'appartenance inférieure à 0.5 (cf table 6).

Probabilité maximale	Forêt aléatoire2	Boosting	Régression logistique pénalisée	LDA pénalisée
supérieure ou égal à 0.75	2	38	37	85
comprise entre 0.5 et 0.75	62	55	52	24
comprise entre 0.25 et 0.5	44	16	20	0
inférieure à 0.25	1	0	0	0

Table 6: Répartition des probabilités maximales par méthode

Cependant bien que la méthode forêt aléatoire2 soit la plus hésitante, elle est aussi celle qui présente la meilleure accuracy et présente donc un grand intérêt pour effectuer des prévisions fiables. Le choix de retenir ces quatre méthodes peut ainsi apparaître comme un compromis entre l'accuracy et la valeur de la probabilité d'affectation à la classe retenue (que l'on souhaite la plus grande possible). Ainsi la décision finale n'est jamais le résultat d'une seule méthode et lorsqu'une méthode hésite entre plusieurs classes, le choix peut être corroboré par une autre dont l'affectation est beaucoup plus tranchée.

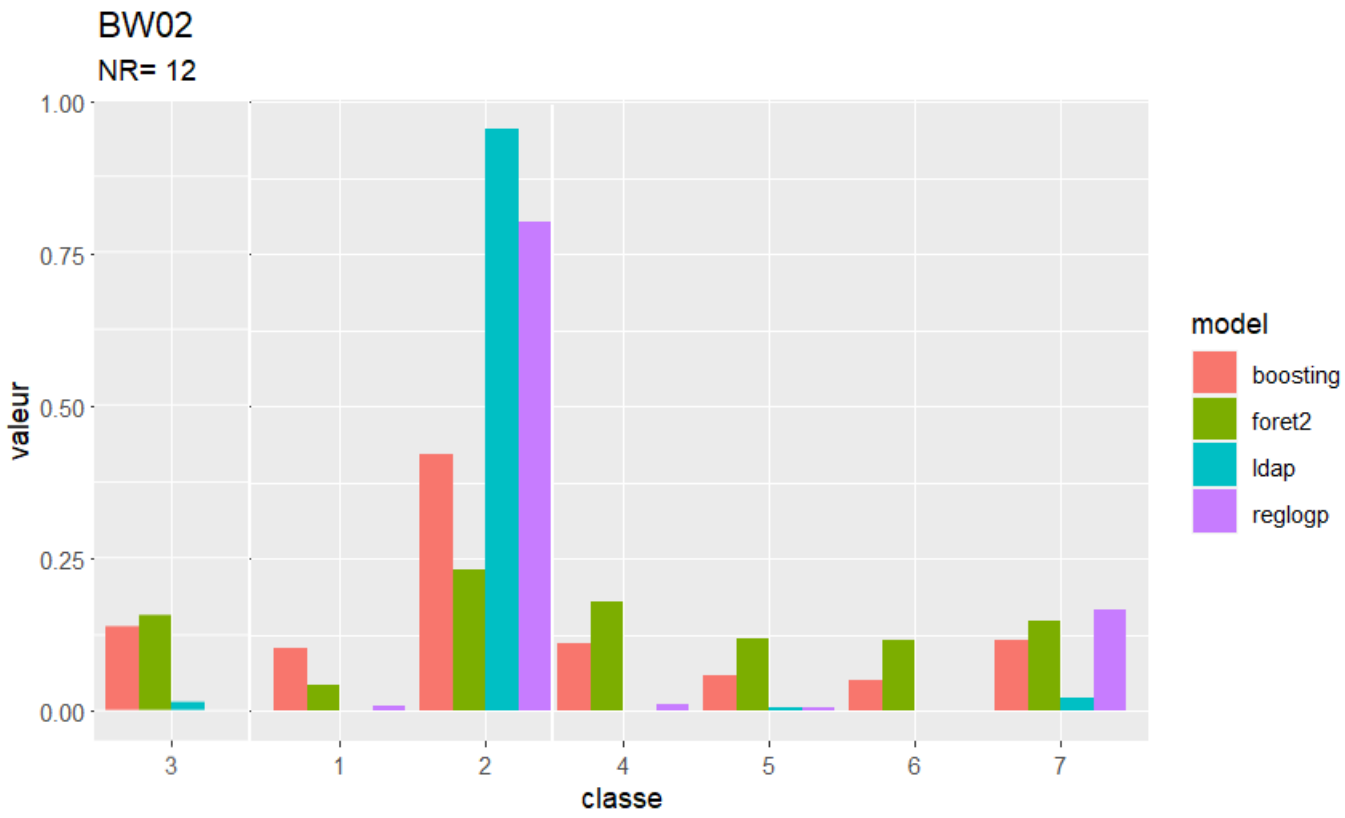

Figure 3: Résultats ensemble BW02

Ainsi la méthode forêt aléatoire 2 affecte l'ensemble BW02 à la classe 2 avec une probabilité environ égale à 0.23 (Figure 3) alors que cette méthode donne également une probabilité d'appartenance à la classe 4 environ égale à 0.18 donc très proche. Mais le choix de la classe 2 est largement confirmé par les autres méthodes et avec une probabilité très élevée dans le cas de la méthode lda pénalisée (environ égale à 0.96).

4.3 Résultats et discussion

Le choix de conserver quatre méthodes fournit plus d'informations que si on s'était contenté de la classe rendue par une seule méthode. En particulier, il permet de distinguer les ensembles supplémentaires pour lesquels les différentes méthodes effectuent des prédictions identiques de ceux où elles diffèrent ou se montrent "hésitantes". Il se présente alors différents cas de figure. Reprenons les cas définis dans le paragraphe précédent pour affecter une classe à un ensemble.

Cas 1 : les méthodes effectuent la même affectation.
Ce cas conserne 55 ensembles sur les 109 testés. Même dans ce cas de figure, le plus favorable, où les quatre méthodes effectuent la même affectation pour un ensemble, plusieurs situations peuvent se présenter. Pour une grande partie des 55 ensembles concernés (28 sur 55) chacune des 4 méthodes l'affecte à une classe avec une probabilité supérieure ou égale à 0.5 et parfois une probabilité très supérieure (supérieure à 0.8).

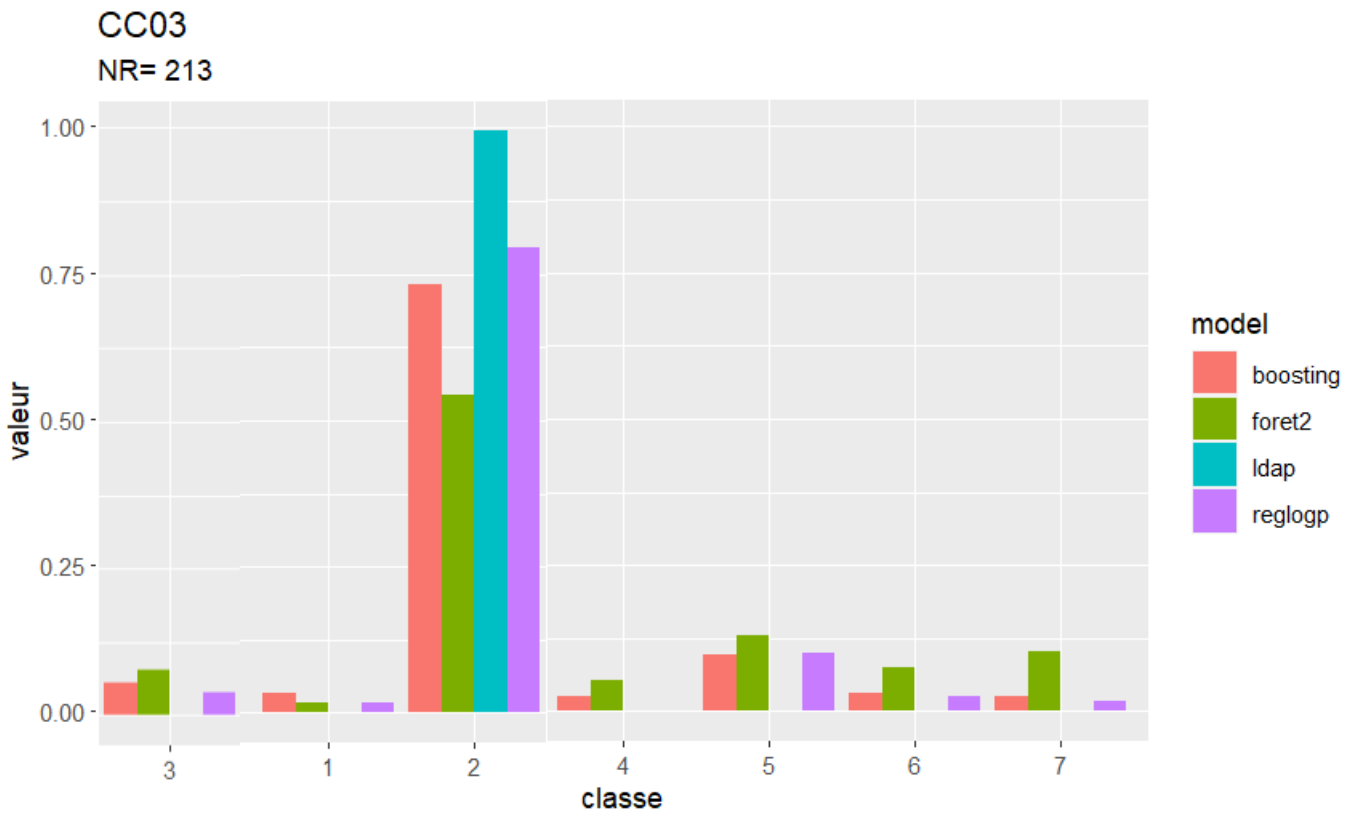

Figure 4: Résultats ensemble CC03

Ainsi dans le cas de l'ensemble CC03 (figure 4) les quatre méthodes affectent l'ensemble à la classe 2 avec une probabilité allant de environ 0.54 (pour la méthode forêt2) à environ 0.996 (pour la méthode ldap). La réponse rendue (classe 2) semble dans ce cas plutôt fiable.

Cas 2 : les différentes méthodes renvoient deux résultats différents
Ce cas concerne 48 des 109 ensembles testés (33 ensembles dans le cas 2a et 15 ensembles dans le cas 2b). C'est certainement le cas où l'affectation finale effectuée

selon la stratégie adoptée est le plus sujet à caution, en particulier dans le cas 2b où on ne peut pas utiliser l'argument de la majorité et lorsque les deux classes renvoyées par les différentes méthodes ne sont pas contigues. L'avis d'un expert avec des arguments autres que statistiques sera alors indispensable.

L'ensemble DI902 (figure 5) est un bon exemple de cas particulièrement ambigü où notre stratégie de classement pourrait être prise en défaut. Deux méthodes l'affectent à la classe 3 et deux autres à la classe 6, c'est-à-dire deux classes éloignées l'une de l'autre. C'est finalement la classe 6 qui sera délivrée avec notre stratégie avec une probabilité d'affectation à cette classe par la méthode ldap environ égale à 0.589. Mais la méthode boosting affecte cet ensemble à la classe 3 avec une probabilité environ égale à 0.583. Ces deux probabilités sont extrêment proches et l'affectation finale s'est jouée à très peu de choses. Finalement l'archéologue tranchera en faveur de la classe 6 (la classe fournie par l'outil stastistique) car il a identifié cet ensemble comme étant récent et contenant du matériel redéposé.

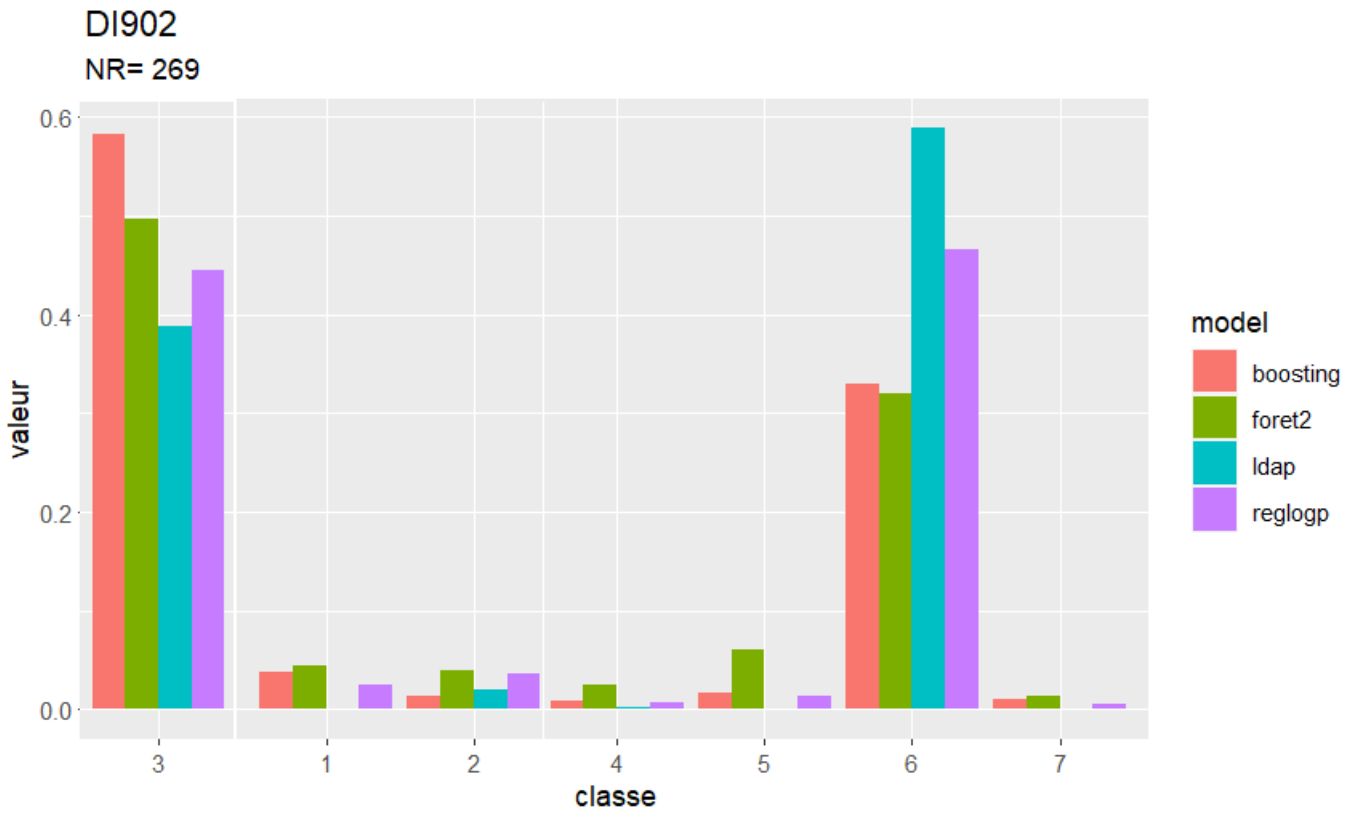

Figure 5: Résultats ensemble DI902

L'ensemble DI1000 (figure 6) constitue un autre exemple pour lequel les méthodes hésitent entre deux classes non contigues (bien que plus proches que dans l'exemple précédent). Là aussi l'archéologue aura des arguments en faveur de la classe 6, classe rendue par l'outil statistique, ce qui montre que la stratégie adoptée donne de bons résultats.

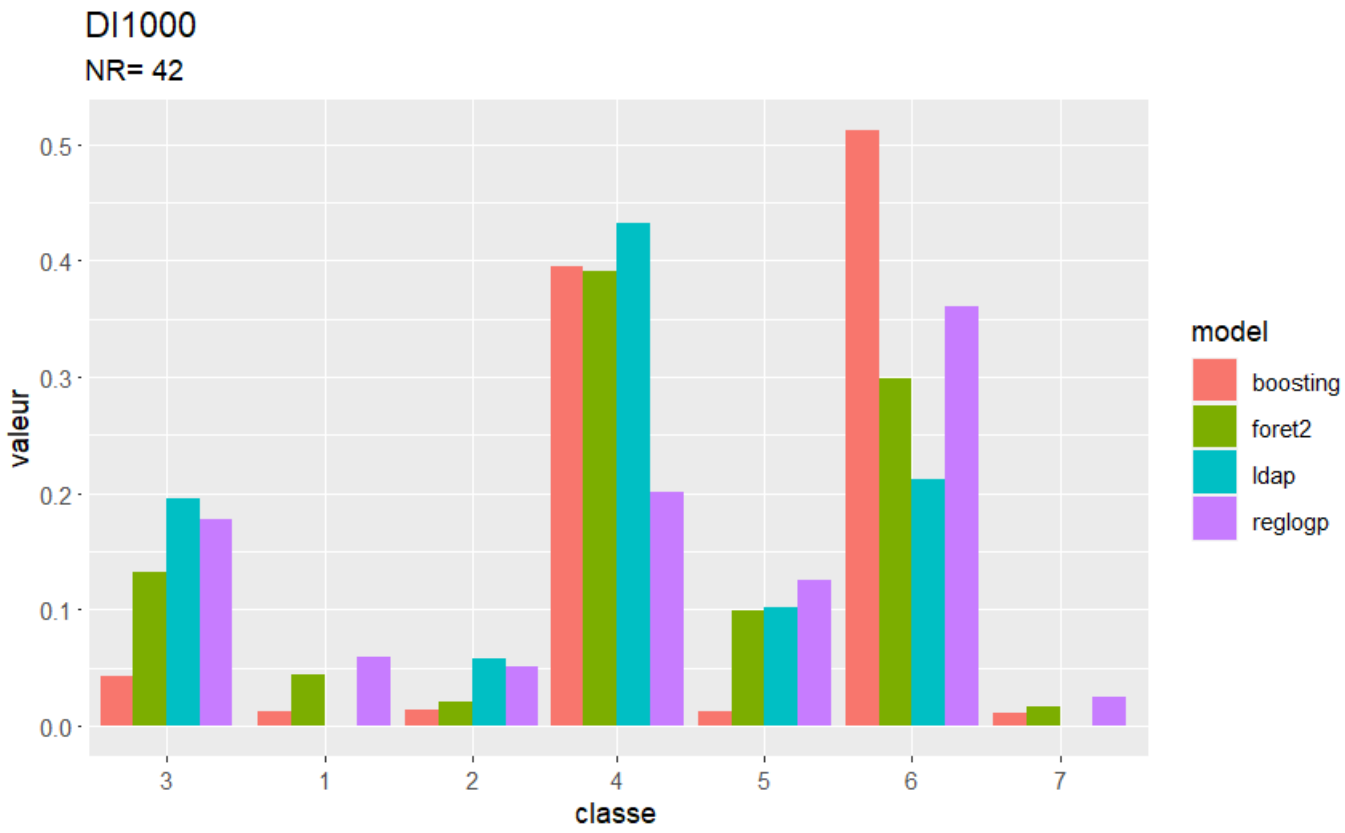

Figure 6: Résultats ensemble DI1000

Cependant les hésitations pour les ensembles entrant dans le cas 2b se font logiquement le plus souvent entre des classes voisines. C'est le cas par exemple de l'ensemble DH208 (figure 7). Deux méthodes l'affectent à la classe 4 alors que les deux autres méthodes l'affectent à la classe 5. Chacune des méthodes réalise son affectation avec une probabilité supérieure à 0.55 contre des probabilités pour les autres classes très inférieures. Bien que divergentes, les différentes méthodes délivrent une prévision assez sûre. Ces deux classes correspondant à un même faciès, cette hésitation entre les méthodes ne constitue alors pas toujours un réel problème pour l'archéologue.

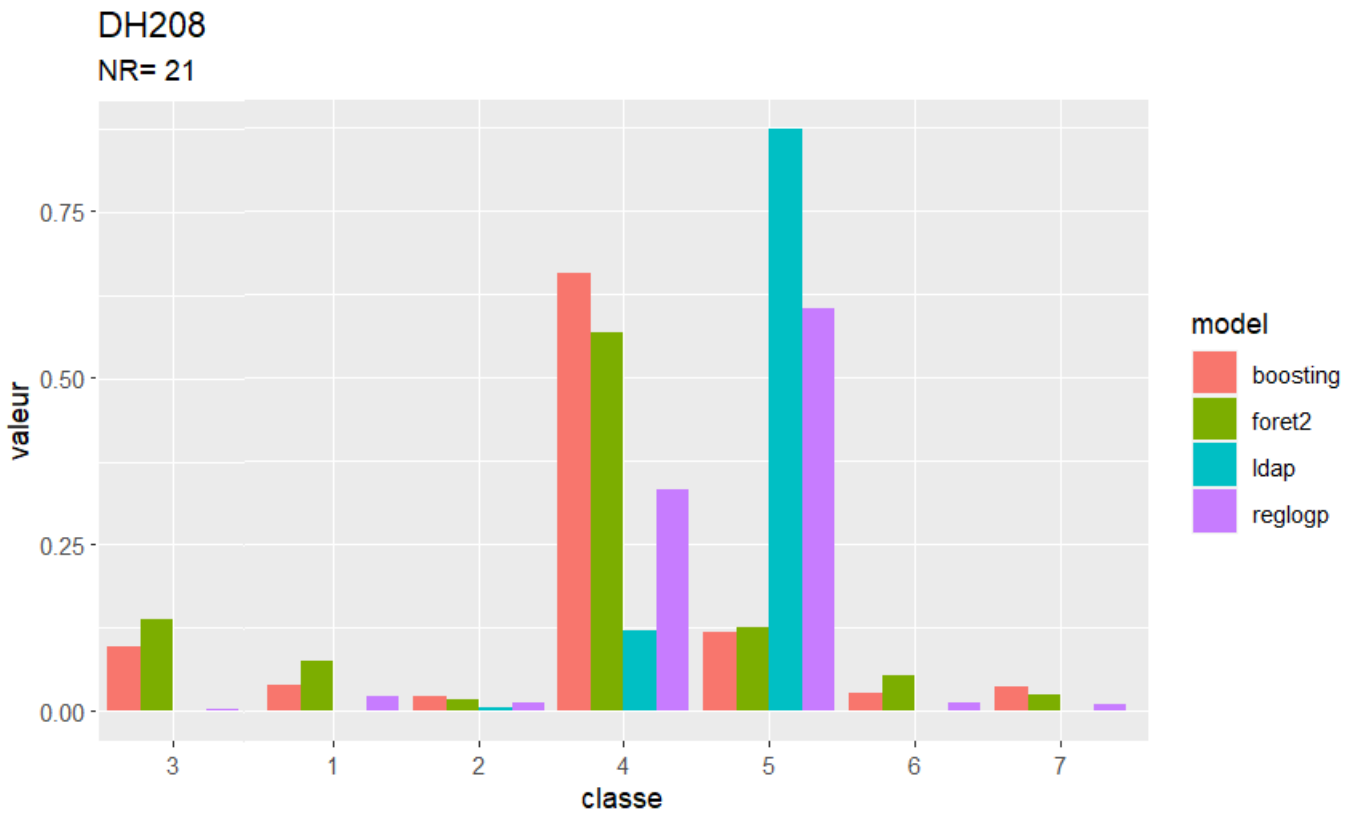

Figure 7: Résultats ensemble DH208

L'ensemble DI802 (figure 8) fournit un autre exemple d'ensemble pour lequel les quatre méthodes prédisent deux classes contigues. Deux méthodes l'affectent avec une forte probabilité à la classe 5 tandis que les autres l'affectent (avec des probabilités moindres) à la classe 4. Bien que ces deux classes correspondent au même faciès, il existe de solides arguments archéologiques qui impliquent que cet ensemble soit de la fin du faciès 3 et qu'il appartienne donc à la classe 5. Malgré des affectations différentes selon les méthodes, la classe délivrée par l'outil est bien celle qui est confirmée par l'archéologue.

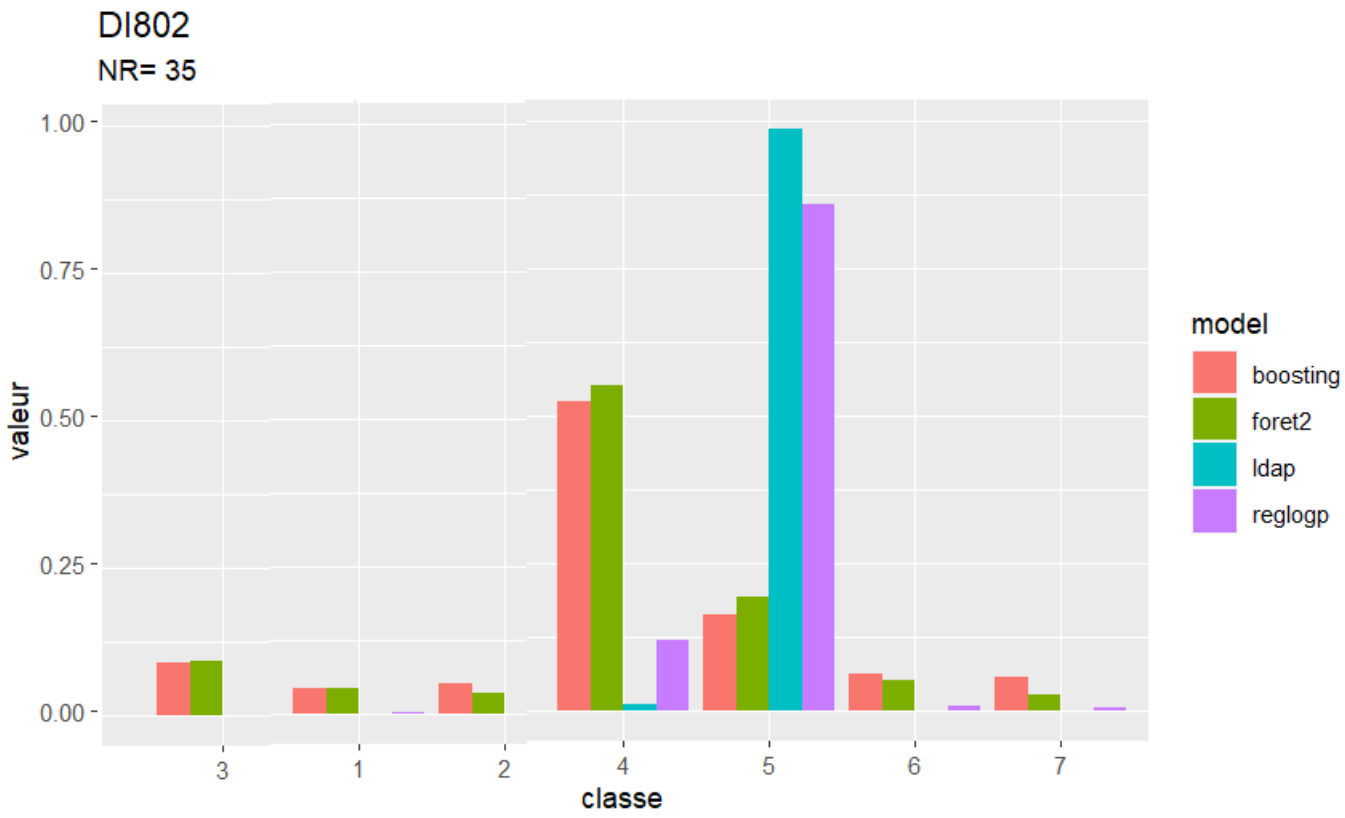

Figure 8: Résultats ensemble DI802

Pour les ensembles du cas 2a, la règle de la classe majoritaire s'applique. Souvent la 2ème classe (celle qui est prédite par une seule des quatre méthodes) est éloignée de la classe majoritaire. Mais pour beaucoup de ces ensembles la classe majoritaire renvoyée par l'outil statistique a été confirmée par l'archéologue. Ce qui confirme l'intérêt de la stratégie adoptée. Par exemple, pour l'ensemble AWBB18 (figure 9), trois classes l'affectent à la classe 2 et une à la classe 7 très éloignée. Mais cet ensemble correspond à un gros aménagement (creusement de grandes tranchés). Aussi l'archéologue opte en faveur de la classe 2 et explique l'affectation à une autre classe par une des méthodes par la présence probable de matériel intrusif justifiée par la nature de cet ensemble.

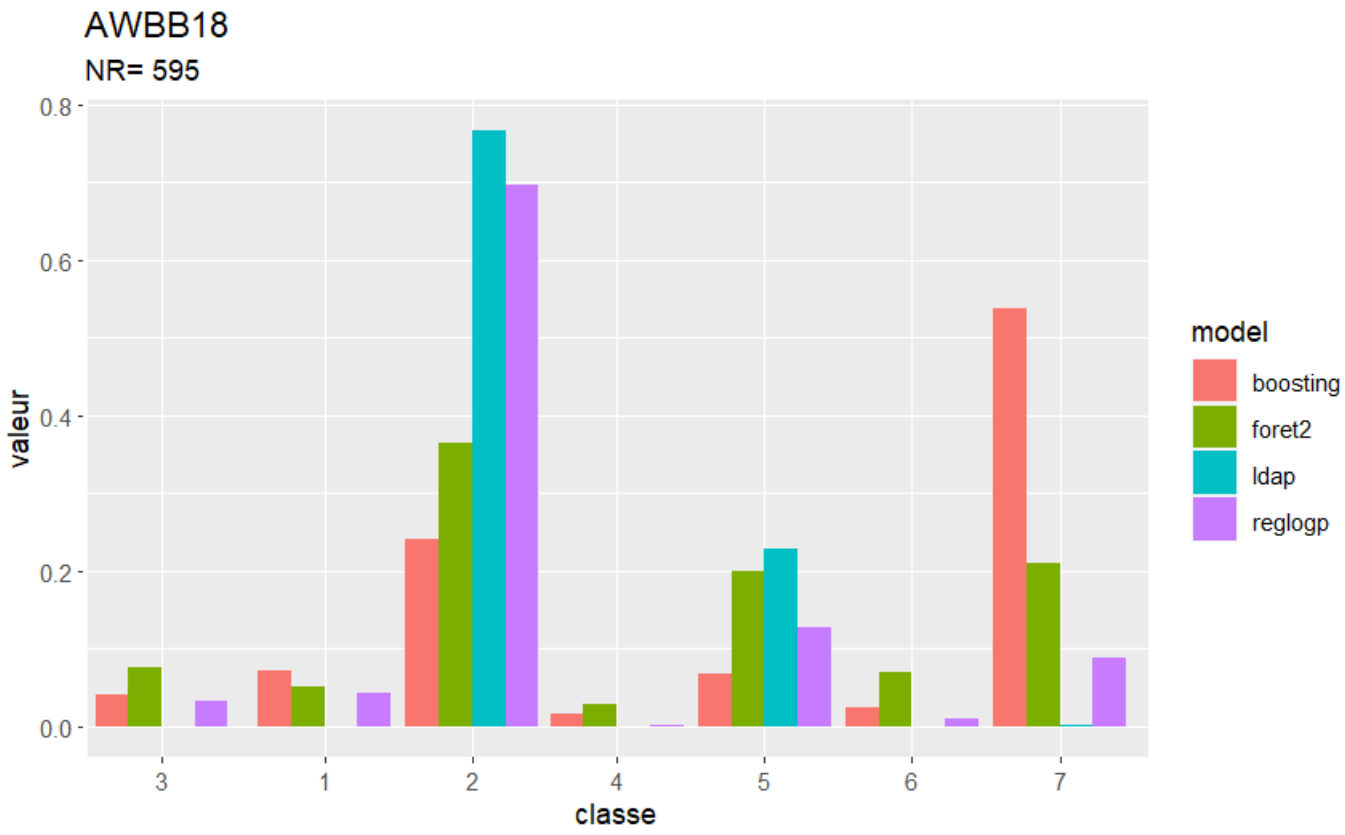

Figure 9: Résultats ensemble AWBB18

De même pour l'ensemble BE06 (figure 10) des arguments archéologiques valident son afectation à la classe 2 (la classe majoritaire). La prédiction en classe 7 effectuée par la méthode boosting étant très certainement due à la présence de matériel perturbateur. Cet exemple illustre la pertinence de la règle majoritaire.

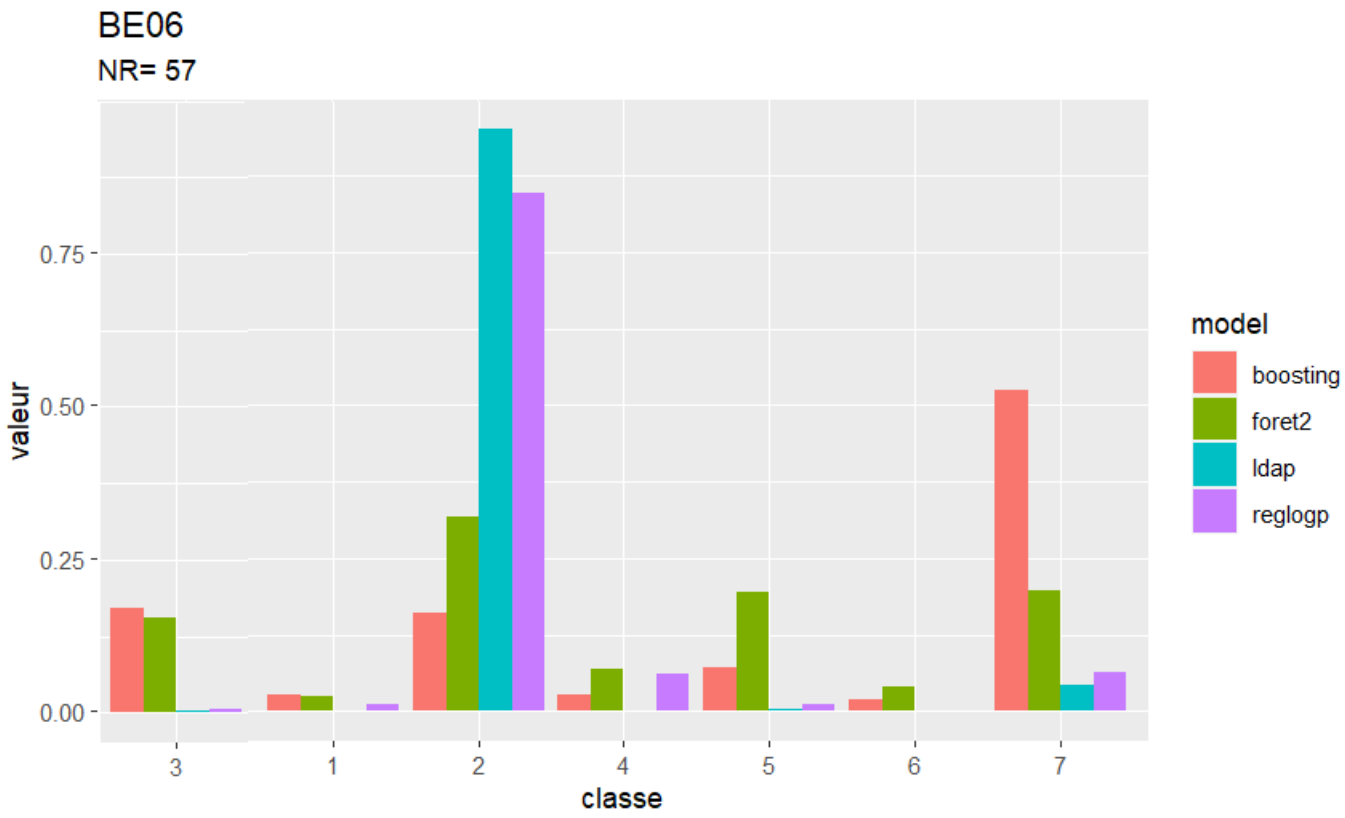

Figure 10: Résultats ensemble BE06

On trouve dans ce cas 2a également des ensembles pour lesquelles les deux prédictions sont effectuées dans des classes contigues mais n'appartenant pas au même faciès. C'est le cas de l'ensemble BM04 (figure 11). Ainsi trois méthodes l'affectent à la classe 6 et une à la classe 5. Mais archéologiquement il est obligatoirement de la classe 6 avec possiblement aussi un peu de matériel de la classe 5 redéposé.

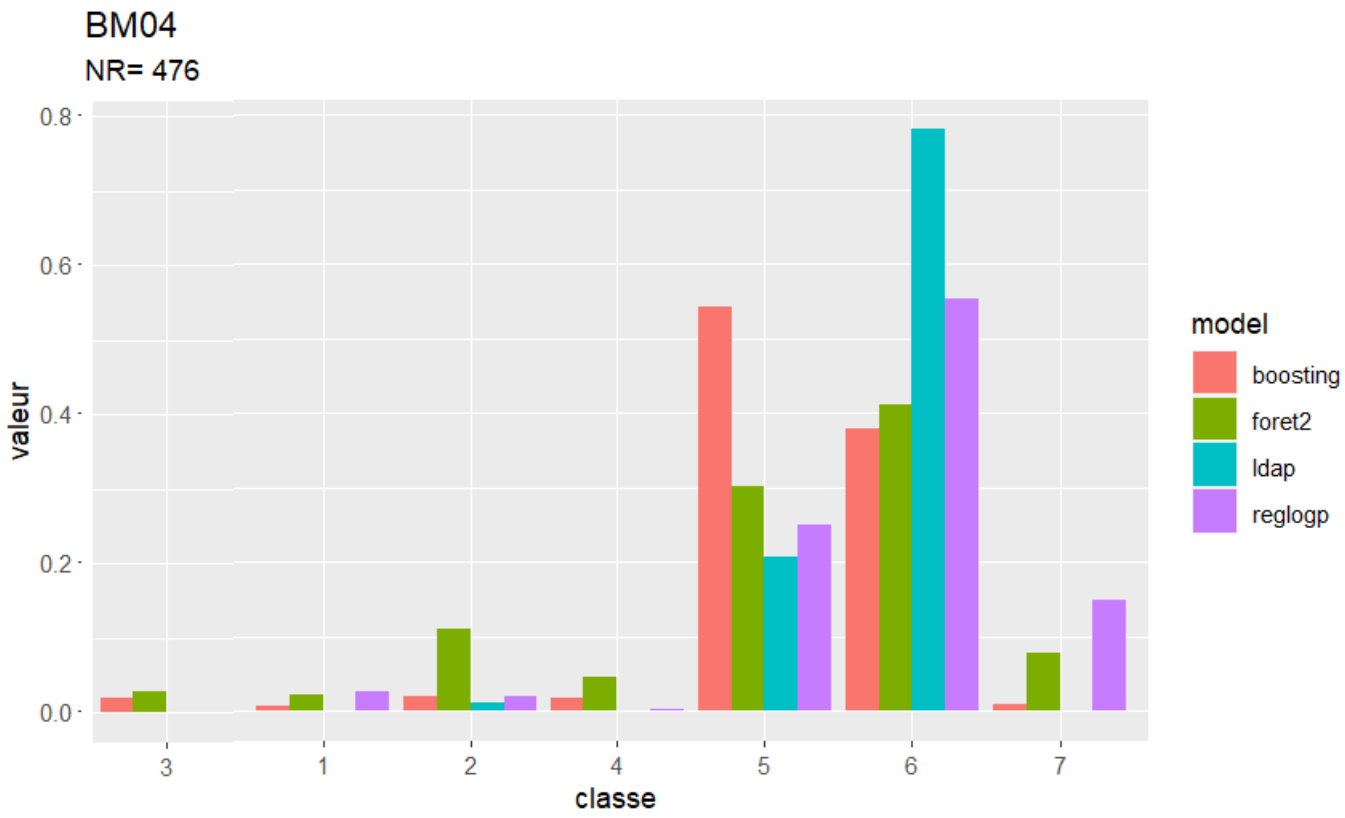

Figure 11: Résultats ensemble BM04

Dans tous les exemples montrés de ce cas 2 sans doute plus litigieux que les autres et pour les ensembles duquel l'avis d'un expert est nécessaire on remarquera que finalement l'archéologue a tranché en faveur de la réponse unique apportée par l'outil statistique. Ceci illustre que la stratégie adoptée pour rendre une réponse, même si elle est loin d'être infaillible, donne de bons résultats sur ce jeu de données. De plus les hésitations des différentes méthodes, et la classe minoritaire trouvent souvent elles aussi des explications archéologiques. On peut alors imaginer qu'elles sont elles mêmes (et pas seulement la réponse unique rendue) porteuses d'informations exploitables.

Cas 3 : les méthodes renvoient trois résultats différents.
Ce cas concerne seulement 6 ensembles sur les 109 testés.
Dans un cas (ensemble DI805, figure 12) les méthodes affectent l'ensemble à des classes contigues (classes 3, 1, 2) ce qui pourrait avoir du sens (l'ensemble pourrait se trouver à "l'interface" entre plusieurs classes).

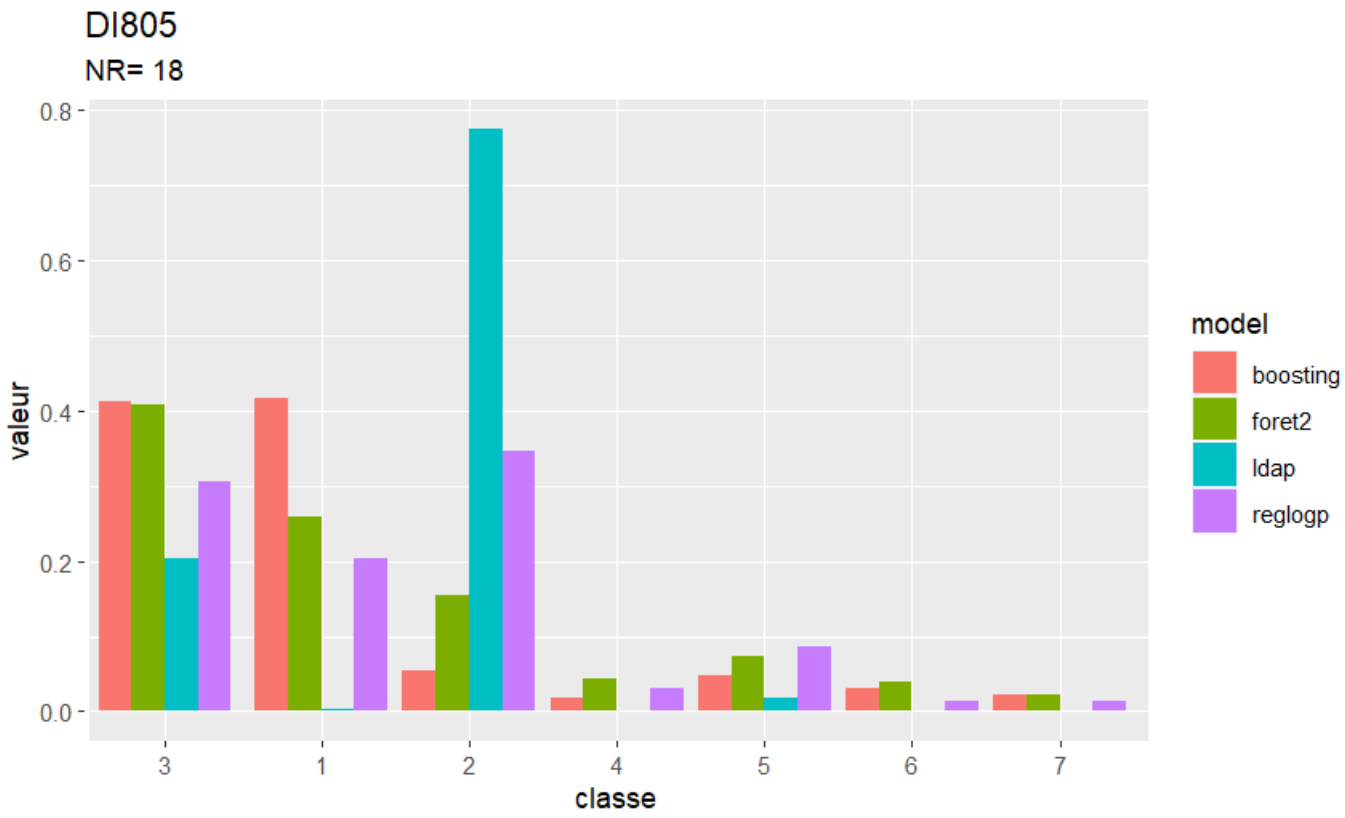

Figure 12: Résultats ensemble DI805

Mais dans d'autres cas (ensembles BJ07, figure 13 et CN02, figure 14) elles leur affectent des classes non contigues (classes 2, 5, 7) ou partiellement contigues (ensemble BE07 affecté aux classes 1, 2, 5, figure 15) ce qui n'a aucun sens d'un point de vue chronologique mais peut s'expliquer par des phénomènes de pollution des ensembles par redéposition.

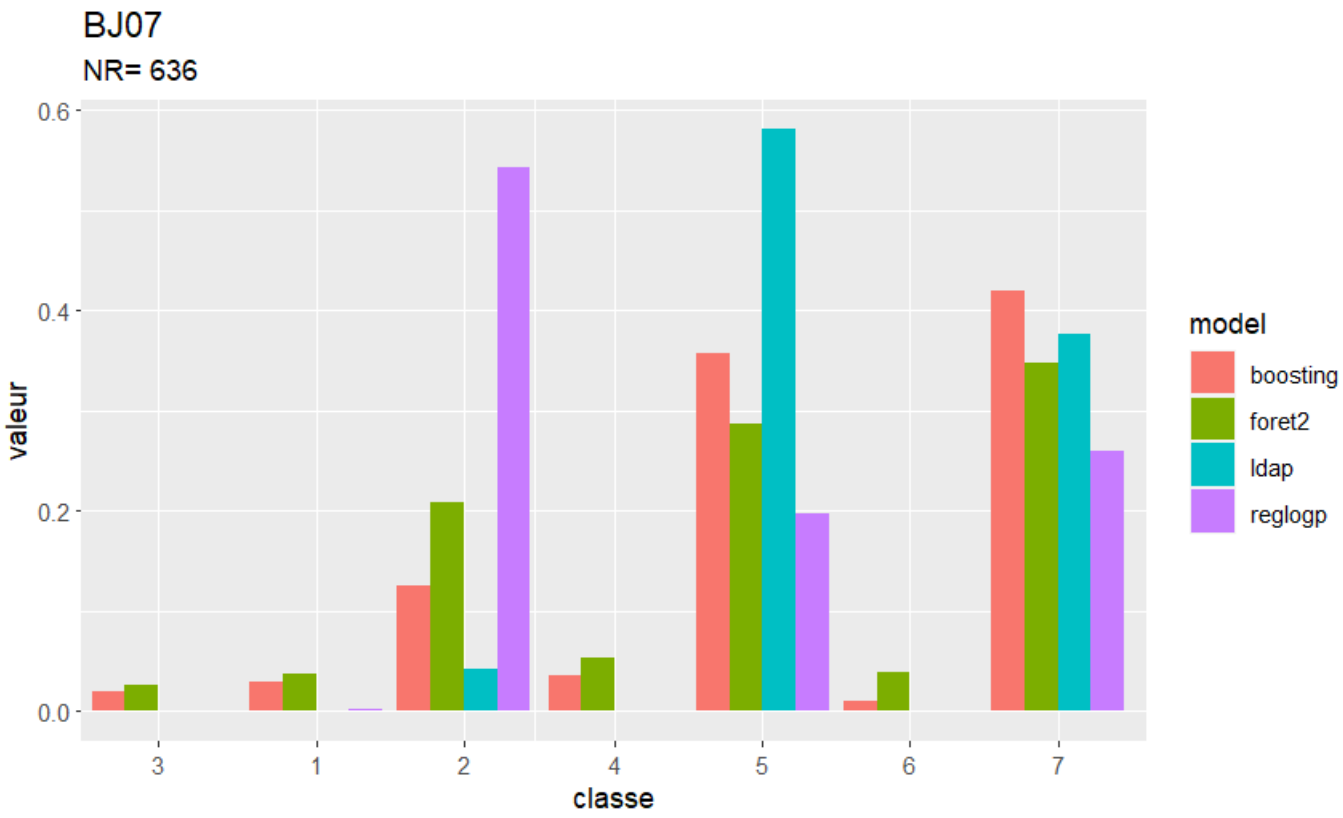

Figure 13: Résultats ensemble BJ07

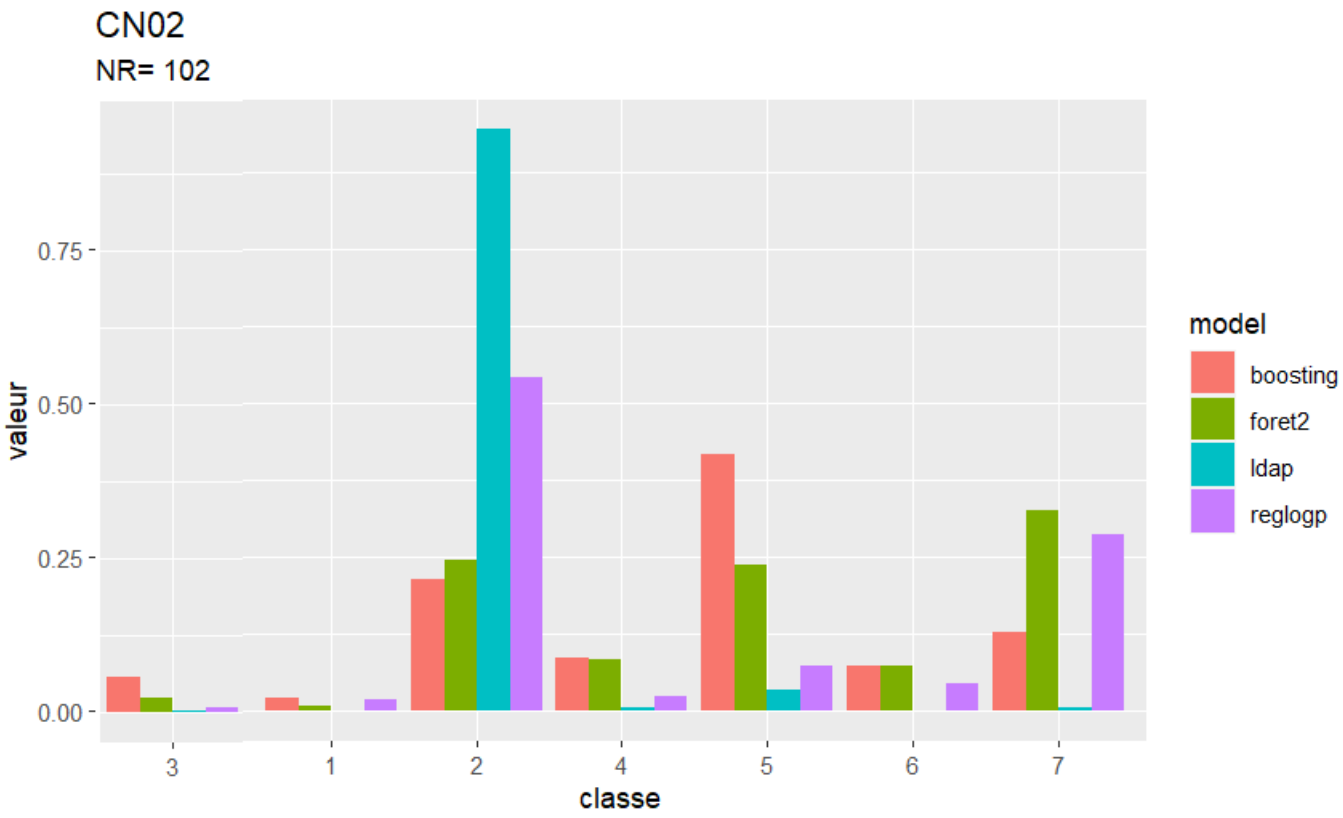

Figure 14: Résultats ensemble CCN02

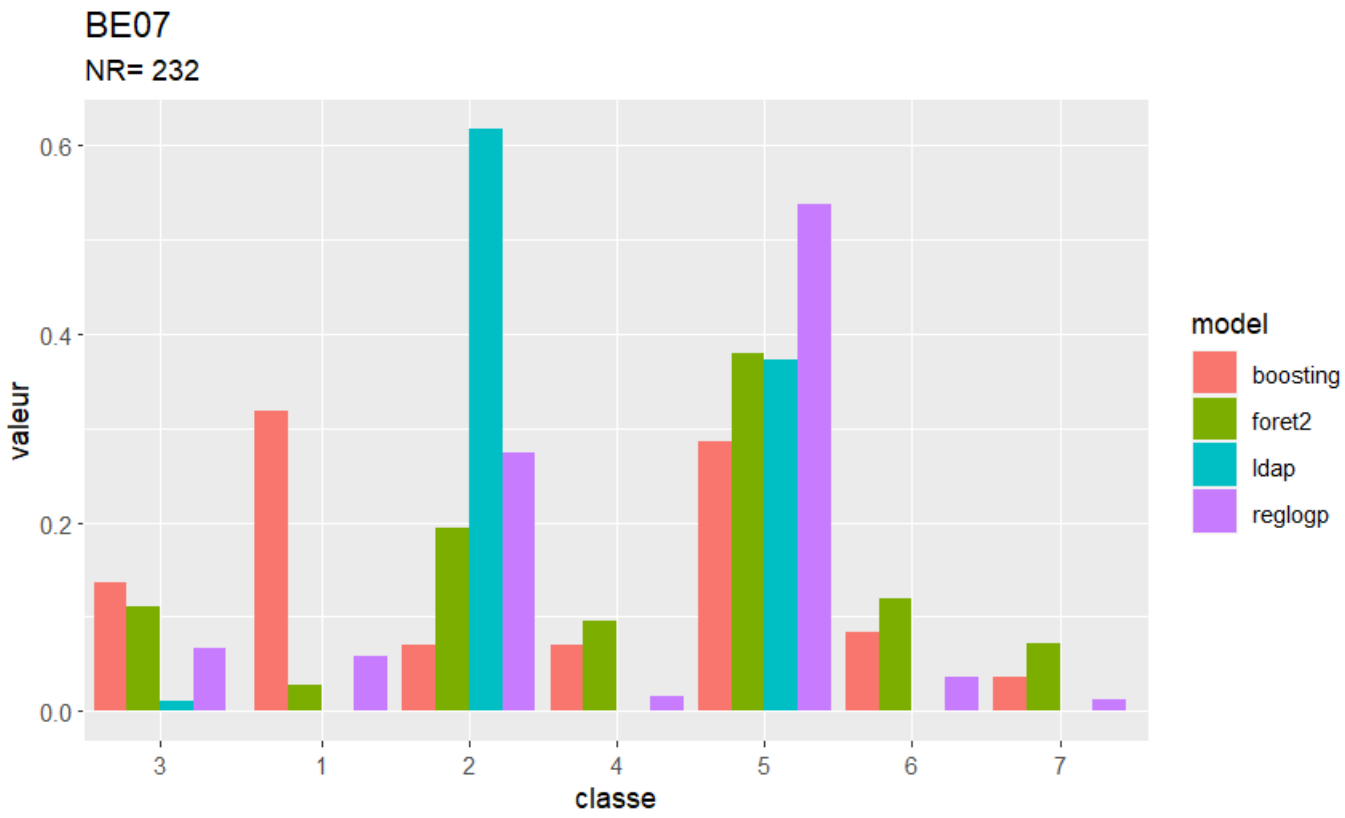

Figure 15: Résultats ensemble BE07

Ces ensembles ont été identifiés par l'archéologue commme des ensembles fortement perturbés ce qui explique les réponses peu cohérentes apportées par les différentes méthodes retenues. Ainsi, il a été relevé dans l'ensemble DI805 de la céramique chinoise importée. Concernant l'ensemble BE07, il suppose qu'il a pu être pollué par l'ensemble BE08 adjacent. L'ensemble CN02 correspond à un comblement de chenal et contient de ce fait beaucoup de matériel redéposé (la céramique apporte peu d'informations pour ce type d'ensemble). Enfin deux des six ensembles de ce cas 3 ont été écartés de l'étude archéologique car jugés peu interprétables. Cependant malgré toutes ces réserves, l'archéologue considère comme logiques les réponses qui ont été données par l'outil statistique pour trois des quatre ensembles conservés. En revanche il semble totalement échoué pour l'ensemble CN02 : il l'affecte à la classe 2 alors que selon la logique archéologique, il devrait être à la classe 6 ou la classe 7. Cependant on peut observer sur ces exemples que même dans le cas d'ensembles très perturbés, l'outil statistique peut réaliser de bonnes performances.

On voit que pour ce type d'ensemble les méthodes statistiques peinent légitimement

à leur affecter une classe. Pourtant dans certains cas ces affectations s'effectuent avec une probabilité assez élevée. Ainsi la méthode ldap affecte l'ensemble CN02 à la classe 2 avec une probabilité très élevée (environ égale à 0.95). Dans d'autres cas, les probabilités d'affectation à plusieurs classes sont voisines. Ainsi pour la méthode foret2, la probabilité que l'ensemble CN02 soit affecté à la classe 2 est légèrement supérieure à 0.24, la probabilité qu'il soit affecté à la classe 5 est légèrement inférieure à 0.24 et enfin celle qu'il le soit à la classe 7 est environ égale à 0.33 (ce qui confirme le caractère "problématique" de cet ensemble). Cette difficulté des différentes méthodes à effectuer une même prédiction (et qui peut peut-être s'expliquer par des sensibilités au bruit différentes selon les méthodes) apparaît grâce au choix de conserver pour la prédiction finale plusieurs méthodes et non une seule. Cela représente un avantage de ce choix et doit inciter l'utilisateur à regarder de plus près de tels ensembles et à convoquer des arguments de nature autre que statistique pour trancher mais à condition qu'il soit informé des hésitations des méthodes, ce qui implique que la réponse fournie par l'outil ne doit pas être seulement la classe finale prédite mais doit aussi intégrer le graphique des résultats. Remarquons enfin que si on fait le choix d'une stratégie consistant à affecter un ensemble à la classe qui a la plus forte probabilité toutes méthodes conservées confondues, on affecte l'ensemble BE07 à la classe 2 (au lieu de la classe 5 pour la "stratégie majoritaire") et on affecte l'ensemble BJ07 à la classe 5 (au lieu de la classe 7). La réponse conforme à la logique archéologique est obtenue par la stratégie alternative pour l'ensemble BE07 , mais par la stratégie majoritaire pour l'ensemble BJ07. Dans tous les cas la stratégie alternative ne permettrait pas d'identifer les "désaccords" entre les méthodes et donc le caractère potientiellement problématique de ces ensembles.

5 Limites et perspectives

5.1 Difficultés, limites

La première limite rencontrée concerne le nombre d'individus (ici le nombre d'ensembles stratigraphiques) du jeu de données utilisé. Certains algorithmes nécessitent d'avoir un nombre minimum d'individus dans chaque classe pour entraîner les méthodes de classification supervisée. Lorque le nombre d'individus est petit, cela oblige d'abord à adapter le mode de validation du modèle et en particulier à limiter le nombre de folds utilisés dans le processus de validation croisée. Mais même lorsqu'on se limite au nombre minimum de 2 folds, certaines méthodes n'ont pas pu être appliquées du tout et le tunage des paramètres n'a pas pu s'effectuer. Ce cas s'est produit ici avec les méthodes QDA et SVM.

D'autre part, les données mobilières caractérisant les ensembles supplémentaires qu'on a cherché à classer sont seulement d'origine céramique. Or, comme on l'a vu

avec l'ensemble CN02, il y certains ensembles pour lesquels la céramique apporte peu de choses. Les réponses données par les méthodes de classification supervisée ne peuvent pas alors être interprètées.

De plus, lié à la nature (archéologique) des données, le problème de pollutions des ensembles supplémentaires a été rencontré à de nombreuses reprises et est une des principales causes des erreurs ou hésitations dans les réponses apportées par les méthodes de classification superviée testées.

Ensuite, on a vu que conserver plusieurs méthodes pouvait rendre plus sûre la décision finale apportée (une unique classe prédite). Mais il est difficile de fixer une stratégie de décision. Plusieurs possibilités ont été envisagées (choisir la réponse qui a la plus forte probabilité d'affectation, ne faire participer au vote à la majorité pour chaque ensemble que les méthodes dont la probabilité d'affectation à la classe dépasse un certain seuil, mais alors à quelle valeur fixer ce seuil?). Il est apparu que chaque stratégie rencontrerait des contre exemples. On pourrait envisager de définir une sorte de mesure de fiabilité de la réponse apportée basée sur les résultats obtenus par les différentes méthodes conservées en l'enrichissant d'un code couleur ou d'un niveau de confiance. Par exemple, une prédiction verte serait considérée comme très fiable, une prédiction rouge serait à envisager avec méfiance. Mais définir une telle mesure apparaît difficile, rencontrerait inévitablement des contre exemples, ferait courir le risque de se perdre dans des multitudes de sous cas et finalement compliquerait sans doute inutilement la réponse finale rendue.

Finalement le plus pratique semble de délivrer à l'utilisateur non seulement la classe réponse mais aussi le graphique représentant les probabilités affectées par chaque méthode ainsi qu'un avertissement l'invitant garder un oeil critique sur la classe réponse apportée et à consulter le graphique. Une lecture même rapide de ce graphique devrait permettre à l'utilisateur de repérer des cas éventuellement litigieux, l'inciter à y regarder de plus près avec des arguments autres que statistiques et de questionner la cohérence de la réponse apportée avec le corpus de connaissances à disposition.

Pour certains des ensembles supplémentaires l'outil statistique renvoie une réponse qui peut apparaître comme sûre (car ces ensembles appartiennent au cas 1 ou au cas 2a) mais sont en réalité en totale contradiction avec l'expertise archéologique. Il existe donc des ensembles supplémentaires pour lesquels l'outil statistique est en échec.

Examinons trois exemples :

1er exemple : l'ensemble BU07 (figure 16).

Les quatre méthodes conservées classent cet ensemble dans le groupe 2 (donc ensemble du cas 1) et de manière assez sûre. Trois méthode sur quatre attribuent à cette classe une probabilité comprise entre 0.5 et 0.6 nettement supérieure aux probabilités d'affectation aux autres classes. Seule la méthode ldap se montre plus hésitante.

Pourtant l'archéologue a considéré que cet ensemble devait être classé dans la classe 4 qui non seulement n'est pas contigue à la classe réponse mais a également des probabilités d'affectation pour chacune des quatre méthodes très faibles. L'outil statistique n'a pas du tout détecté un classement dans le groupe 4.

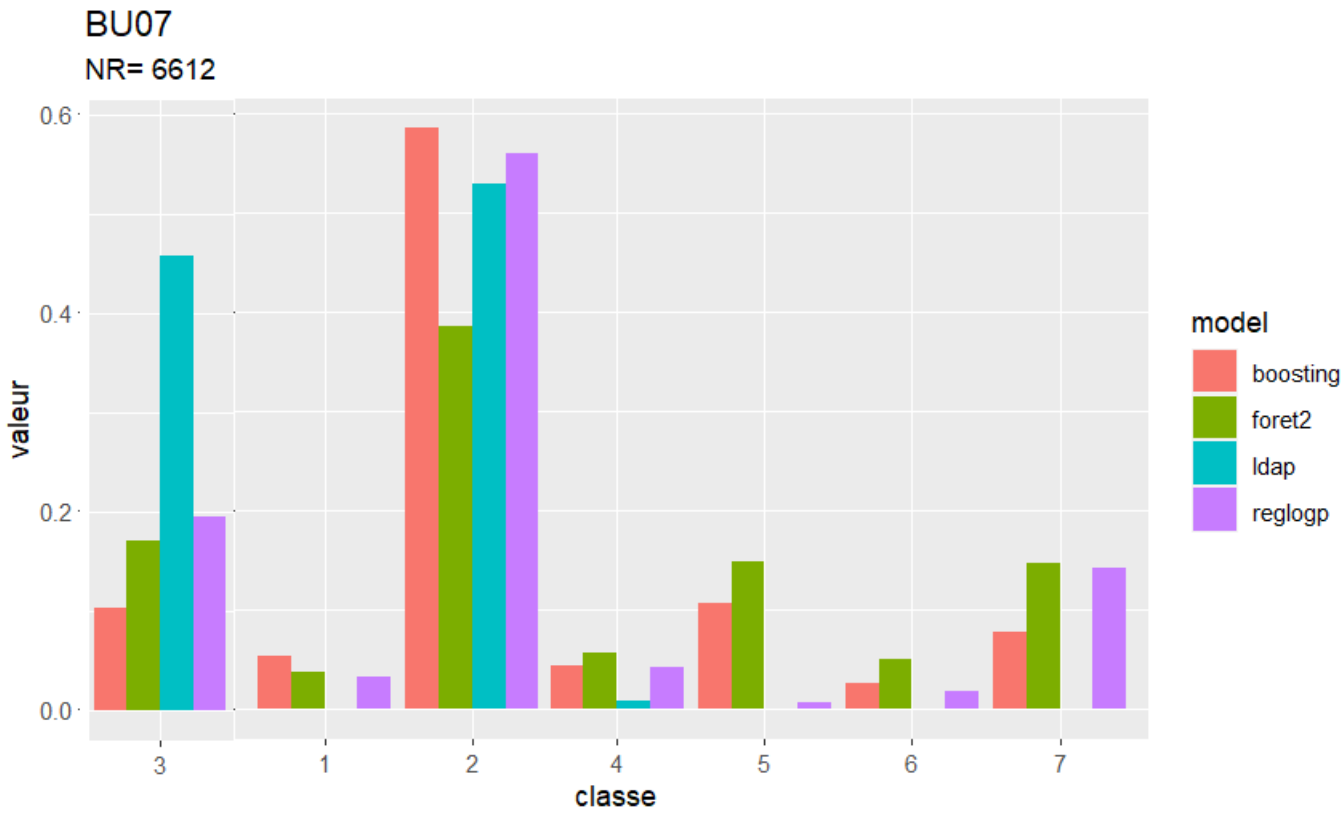

Figure 16: Résultats ensemble BU07

2ème exemple : l'ensemble BO02 (figure 17).
Les quatre méthodes conservées classent cet ensemble dans le groupe 7 (donc ensemble du cas 1) mais avec pour trois d'entre elles des probabilités soit légèrement inférieures, soit légèrement supérieure à 0.5. Seule la méthode ldap attribue à cette classe une probabilité d'affectation qui s'élève à presque 0.8. En considérant de plus les petites valeurs de probabilités d'affectation aux autres classes, la réponse fournie semble fiable. Pourtant cet ensemble provenant d'un premier comblement d'une douve intérieure, d'un point de vue archéologique il ne peut pas être aussi récent (et donc appartenir à la classe 7). Il est plus logique qu'il soit affecté à la classe 4, classe dont les probabilités d'affectation sont très petites pour chaque méthode.

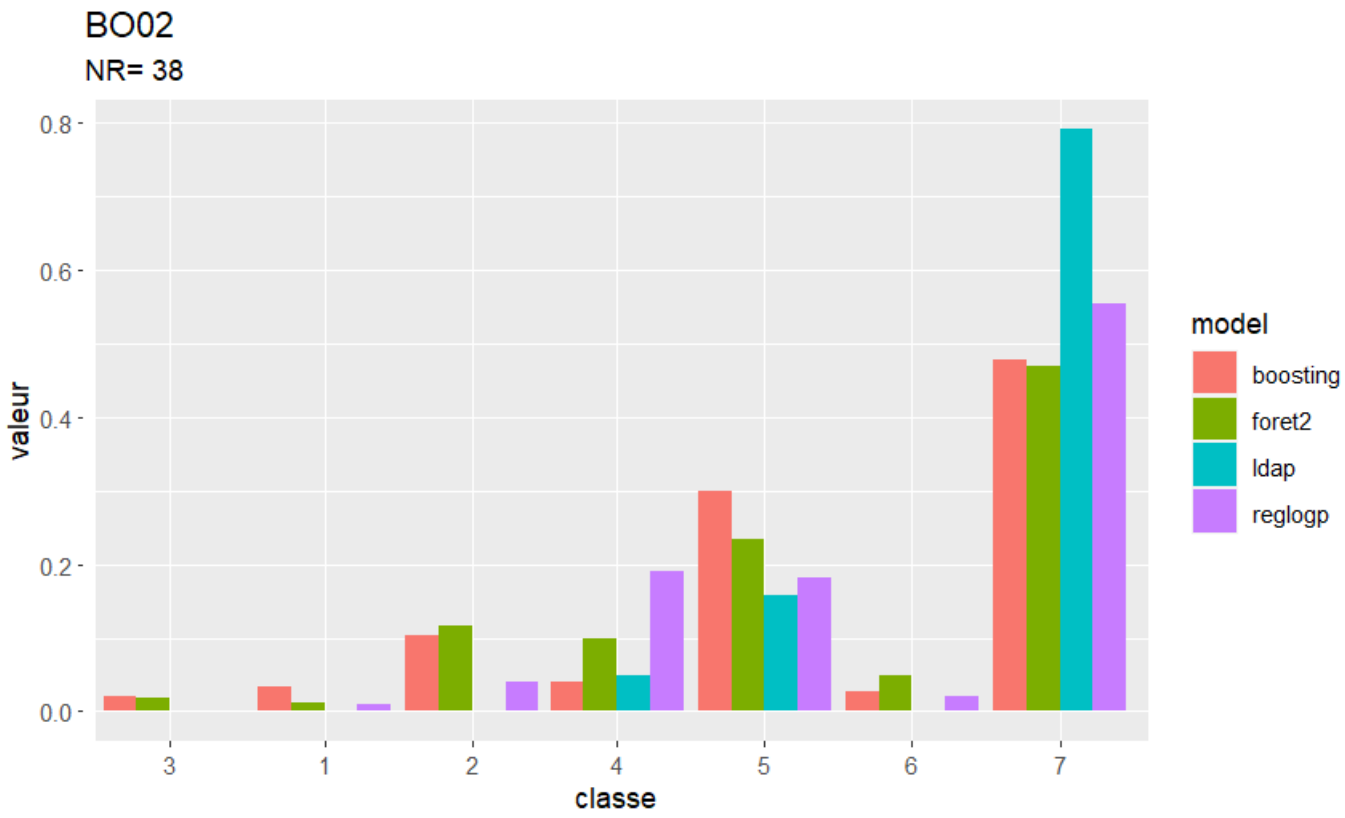

Figure 17: Résultats ensemble BO02

3ème exemple : l'ensemble CN03 (figure 18).
Cet ensemble est dans le cas 2a. Trois méthodes l'affectent (avec des probabilités élevées pour deux d'entre elles) à la classe 5 et une méthode (foret2) à la classe 2 mais avec une faible probabilité et une forte hésitation avec la classe 5. Cet ensemble est donc proche du cas 1 et la réponse apportée apparaît sûre. Mais cet ensemble a été identifié par l'archéologue comme pouvant être perturbé (présence au dessus du matériel relevant des classes 5 et 6, possibilité d'écoulement d'eau) et pense de ce fait que cet ensemble devrait être logiquement affecté à la classe 7. L'erreur commise ici semble bien s'expliquer par la présence identifiée de matériel relevant de la classe 5.

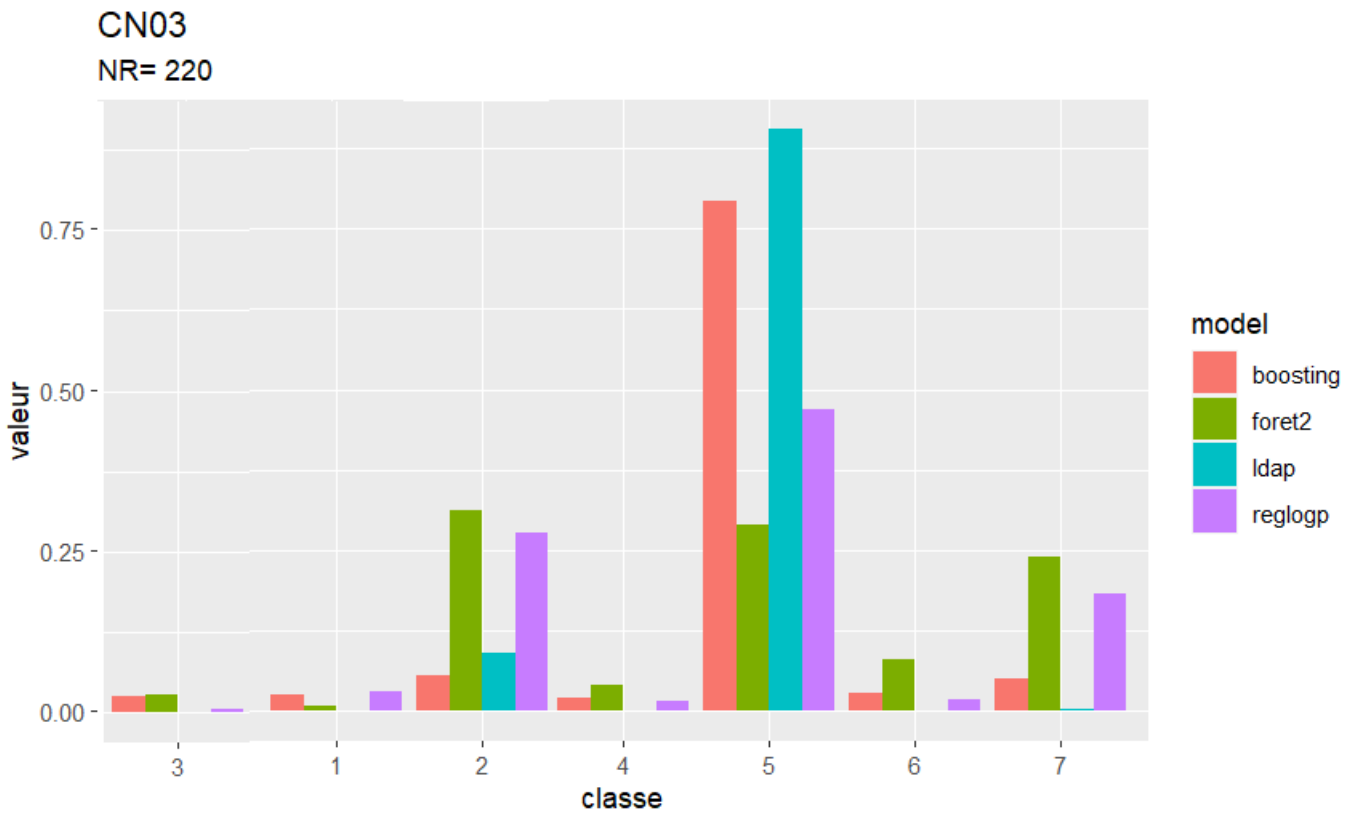

Figure 18: Résultats ensemble CN03

Ces trois exemples illustrent une des limites de l'outil statistique et la nécessité qu'un expert confirme ou valide la réponse rendue même lorsque celle ci apparaît relativement sûre.

5.2 Perspectives

Les tests réalisés au cours de ce travail ont donné des résultats prometteurs dans la perspective de mettre à disposition d'utilisateurs un outil d'aide à la classification supervisée. Il faudrait refaire des tests sur d'autres jeux de données en particulier des jeux de données pour lesquels les méthodes qui ont dues être écartées ici pourraient être utilisées. Il serait alors intéressant de comparer leurs performances avec celles qu'on a testées ici et de voir comment elles s'insèreraient dans un classement des méthodes destinées à choisir celles que l'on retient pour produire une réponse finale. Il serait également intéressant d'étudier des exemples qui se situent dans d'autres champs que l'archéologie et d'étudier la qualité des prédictions obtenues en suivant

la démarche utilisée ici. A l'issue de tout ce travail de tests, l'objectif final reste d'implémenter dans R un outil capable de réaliser un classement d'individus à un groupe aussi fiable que possible.

Conclusion

Au cours de ce travail, on a testé sur les ensembles supplémentaires, écartés de l'étape de classification non supervisée faute de qualité suffisante, 14 variantes de méthodes de classification supervisée (certaines ne se différenciant que par le nombre de paramètres à tuner) et on a comparé leurs performances en utilisant la métrique accuracy d'abord et en tenant compte également de la ROC-AUC. Les performances variées (et parfois très faibles de certaines méthodes) nous ont conduit finalement à en sélectionner quatre et à décider d'une stratégie pour rendre une unique classe d'affectation à chaque ensemble supplémentaire testé. Construire une réponse à partir d'un choix de méthodes à l'accuracy supérieure à 0.7 et selon la stratégie utilisée ("stratégie majoritaire") s'est avéré être un bon moyen pour obtenir de bons résultats dans le sens où dans la grande majorité des cas, c'est-à-dire des ensembles supplémentaires, la réponse apportée par l'outil statistique était validée par l'expert archéologue. Malgré la difficulté pour définir une mesure de fiabilité rigoureuse et utilisable, les choix effectués ont permis de réaliser une répartition des ensembles en plusieurs cas. La nature du cas dans lequel se trouve l'ensemble pour lequel on veut réaliser une prédiction peut constituer une indication de fiabilité. Mais une indication seulement, car on a vu qu'on pouvait trouver dans chaque cas des exemples où le résultat rendu par l'outil statistique était en contradiction avec la logique archéologique. En conclusion, les données écartées initialement peuvent être exploitées par des méthodes de classification supervisée, et il est permis d'envisager la création d'un outil d'aide à la classification supervisée qui pourra se montrer utile. Cependant, l'existence inévitable de cas litigieux (rappelons que les ensembles testés avaient été sélectionnés pour leur qualité moindre) qui met l'outil statistique en défaut doit inciter l'utilisateur à la prudence, à rester critique vis à vis des réponses fournies. Aussi il apparaît que concevoir un outil statistique qui ne renverrait qu'une classe réponse comporterait des risques mais que joindre à la réponse un graphique représentant les probabilités d'appartenir à chaque classe pour chaque méthode retenue permettrait de nuancer la seule réponse classe, d'aider à détecter des individus "problématiques" et serait finalement utilisable et pratique.

Remerciements

Je tiens à remercier chaleureusement Lise Bellanger, Philippe Husi et Arthur Coulon qui m'ont encadré pendant ce stage pour leur gentillesse et leur disponibilité, et l'accueil qu'ils m'ont fait au laboratoire LAT-CITERES de Tours. Travailler avec eux et bénéficier de leurs connaissances et compétences a toujours été un plaisir et une source d'enrichissement.

Bibliographie

[AB94] Douglas G Altman and J Martin Bland. "Diagnostic tests 3: receiver operating characteristic plots." In: *BMJ: British Medical Journal* 309.6948 (1994), p. 188.

[BCH21a] Lise Bellanger, Arthur Coulon, and Philippe Husi. "Determination of cultural areas based on medieval pottery using an original divisive hierarchical clustering method with geographical constraint (MapClust)". In: *Journal of Archaeological Science* 132 (2021), p. 105431.

[BCH21b] Lise Bellanger, Arthur Coulon, and Philippe Husi. "PerioClust: a simple hierarchical agglomerative clustering approach including constraints". In: *Data Analysis and Rationality in a Complex World 16*. Springer. 2021, pp. 1–8.

[BCH21c] Lise Bellanger, Arthur Coulon, and Philippe Husi. "Une méthode de classification ascendante hiérarchique par compromis: hclustcompro". In: *9e Conférence Internationale Francophone sur la Science des Données (CIFSD)*. 2021.

[BD06] Christopher D Brown and Herbert T Davis. "Receiver operating characteristics curves and related decision measures: A tutorial". In: *Chemometrics and Intelligent Laboratory Systems* 80.1 (2006), pp. 24–38.

[BH12] Lise Bellanger and Philippe Husi. "Statistical tool for dating and interpreting archaeological contexts using pottery". In: *Journal of archaeological science* 39.4 (2012), pp. 777–790.

[Bre+84] Leo Breiman et al. "Cart". In: *Classification and regression trees* (1984).

[Bre01] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[CBH21] A Coulon, L Bellanger, and P Husi. *SPARTAAS: Statistical Pattern Recognition and daTing using Archaeological Artefacts assemblageS. R package version 1.0. 0*. 2021.

[CG16] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[Cha+21] Theodore Chadjipadelis et al. *Data Analysis and Rationality in a Complex World*. Springer, 2021.

[FS+96] Yoav Freund, Robert E Schapire, et al. "Experiments with a new boosting algorithm". In: *icml*. Vol. 96. Citeseer. 1996, pp. 148–156.

[GP19] Robin Genuer and Jean-Michel Poggi. *Les forêts aléatoires avec R*. Presses universitaires de Rennes, 2019.

[HBT95] Trevor Hastie, Andreas Buja, and Robert Tibshirani. "Penalized discriminant analysis". In: *The Annals of Statistics* 23.1 (1995), pp. 73–102.

[HS04] Klaus Hechenbichler and Klaus Schliep. "Weighted k-nearest-neighbor techniques and ordinal classification". In: (2004).

[HTB94] Trevor Hastie, Robert Tibshirani, and Andreas Buja. "Flexible discriminant analysis by optimal scoring". In: *Journal of the American statistical association* 89.428 (1994), pp. 1255–1270.

[HTF17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. 2017.

[Husa] Husi. URL: `https://citeres.univ-tours.fr/lat/`.

[Husb] Husi. URL: `https://citeres.univ-tours.fr/contrat/modathom-modele-explicatif-de-la-fabrique-urbaine-dangkor-thom-archeologie-dune-capitale-disparue/`.

[Jam+13] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[KJ+13] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. Springer, 2013.

[Kle+02] David G Kleinbaum et al. *Logistic regression*. Springer, 2002.

[LW14] Cheng Li and Bingyu Wang. "Fisher linear discriminant analysis". In: *CCIS Northeastern University* 6 (2014).

[Red18] Chandan K Reddy. *Data clustering: algorithms and applications*. Chapman and Hall/CRC, 2018.

[Sam12] Richard J Samworth. "Optimal weighted nearest neighbour classifiers". In: (2012).

[Sèv] SèvreLimoges. *céramique*. URL: `https://www.sevresciteceramique.fr/musee/qu-est-ce-que-la-ceramique.html`.

[Sté15] TUFFERY Stéphane. *Modélisation prédictive et apprentissage statistique avec R*. Éditions Technip, 2015.

[Vap13] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[VV+98] Vladimir Naumovich Vapnik, Vlamimir Vapnik, et al. “Statistical learning theory”. In: (1998).

[Wic14] Hadley Wickham. “Tidy data”. In: *Journal of statistical software* 59 (2014), pp. 1–23.

Annexes

Schéma récapitulatif de traitement des données

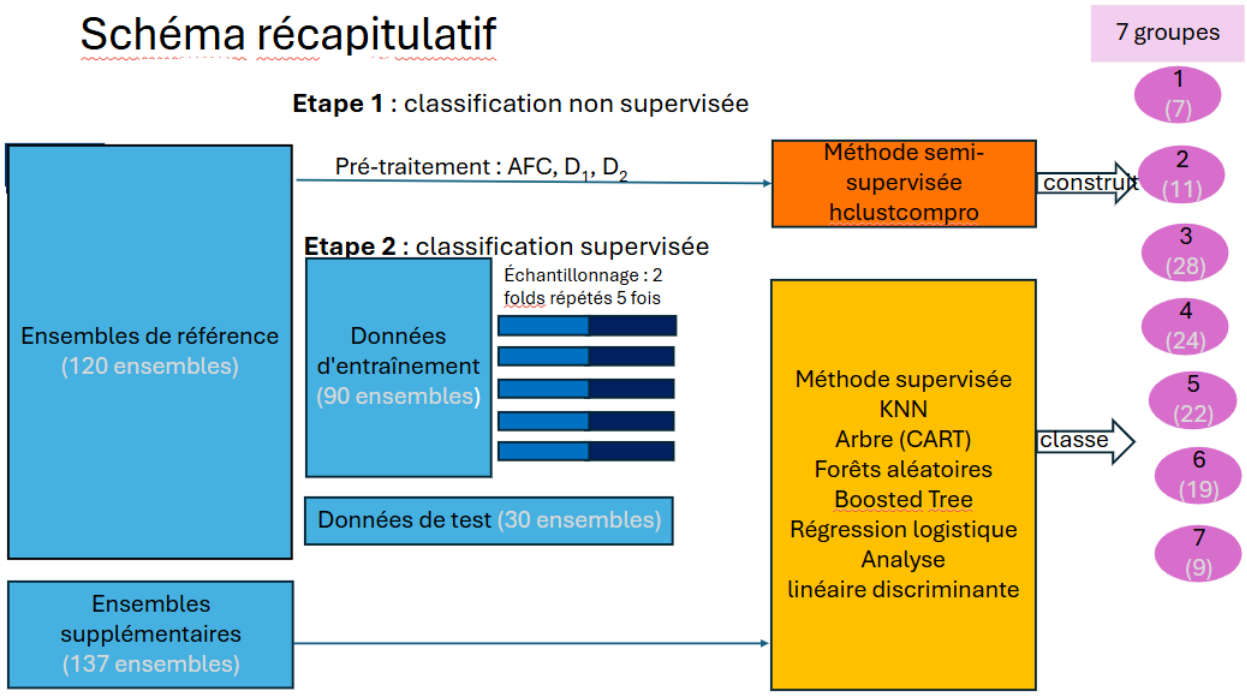

Figure 19: Schéma récapitulatif

Résultats pour les 109 ensemples supplémentaires avec NR>10

	NR	foret2	reglogp	ldap	boosting	classe retenue
CD202	104	2	2	2	2	2
DH104	272	4	4	4	4	4
DI1001	30	6	6	6	6	6
DI202	250	4	4	4	4	4
DI205	68	4	4	4	4	4
DI436	87	3	3	3	3	3
DI501	125	6	6	6	6	6
DI503	52	3	3	3	3	3
DI806	18	3	3	3	3	3
DI807	46	3	3	3	3	3
DI905	170	4	4	4	4	4
DJ300	165	6	6	6	6	6
BK02	14	3	3	3	3	3
BK03	69	3	3	3	3	3
BU03	490	3	3	3	3	3
BU05	675	3	3	3	3	3
BU06	2303	3	3	3	3	3
CF04	79	5	5	5	5	5
CR04	279	3	3	3	3	3
BE14	13	3	3	3	3	3
BJ05	95	5	5	5	5	5
BN03	98	3	3	3	3	3
CC04	2538	2	2	2	2	2
DI301	117	5	5	5	5	5
DI508	34	3	3	3	3	3
DI900	148	4	4	4	4	4
DK100	352	6	6	6	6	6
BK04	120	2	2	2	2	2
BU07	6612	2	2	2	2	2
CF02	257	5	5	5	5	5
CR03	34	6	6	6	6	6
BE11	116	5	5	5	5	5
BJ02	34	3	3	3	3	3
BJ12	31	5	5	5	5	5
BO06	48	6	6	6	6	6
BZ04	60	5	5	5	5	5

	NR	foret2	reglogp	ldap	boosting	classe retenue
BM02	116	6	6	6	6	6
BM05	379	6	6	6	6	6
CN04	352	5	5	5	5	5
CD201	26	5	5	5	5	5
DI1002	36	4	4	4	4	4
DI502	731	4	4	4	4	4
DI505	150	6	6	6	6	6
BU08	191	2	2	2	2	2
CA11	568	2	2	2	2	2
CA15	28	5	5	5	5	5
BE02	56	3	3	3	3	3
BE04	48	3	3	3	3	3
BW02	12	2	2	2	2	2
CC03	213	2	2	2	2	2
DI509	14	3	3	3	3	3
DI804	53	5	5	5	5	5
BU02	76	2	2	2	2	2
BO02	38	7	7	7	7	7
BZ01	118	7	7	7	7	7
DH208	21	4	5	5	4	5
DI1003	28	5	5	6	5	5
DI201	515	6	6	4	6	6
DI800	30	4	5	5	4	5
DI802	35	4	5	5	4	5
DI902	269	3	6	6	3	6
BK05	1624	6	3	3	6	6
CA12	376	5	5	5	2	5
CA13	465	5	5	5	2	5
CF01	29	7	5	7	5	7
BW05	44	7	5	5	5	5
BM03	161	6	7	7	6	7
BM04	476	6	6	6	5	6
CC02	145	2	3	2	2	2
DH209	2166	6	6	6	4	6
DI1000	42	4	6	4	6	6
DI200	92	6	6	4	6	6
DI419	58	3	2	2	3	2

	NR	foret2	reglogp	ldap	boosting	classe retenue
BU04	63	3	3	2	3	3
CA14	35	3	3	3	4	3
BE08	148	2	5	2	2	2
BJ03	105	5	5	1	5	5
BJ04	274	5	2	5	5	5
BJ08	107	7	2	2	7	2
BJ13	633	7	2	7	7	7
BN02	104	3	3	2	3	3
BN05	242	2	2	2	7	2
BO04	48	7	7	7	5	7
BZ02	48	4	3	3	4	3
BZ03	12	6	6	5	6	6
CN03	220	2	5	5	5	5
CC01	128	2	3	2	2	2
DI203	25	4	4	3	4	4
DI304	87	4	5	4	4	4
DI602	12	3	3	6	3	3
DI904	198	4	3	3	4	4
DI906	68	4	5	5	4	5
BE06	57	2	2	2	7	2
BE12	17	2	2	2	5	2
BJ06	75	7	2	2	7	2
BJ11	411	7	2	7	7	7
BN04	68	6	6	6	4	6
BO03	18	6	7	7	6	7
BW03	174	2	2	2	7	2
BW04	173	7	5	5	5	5
AWBB18	595	2	2	2	7	2
AWBB25	74	5	5	5	7	5
CD203	146	5	5	3	5	5
BJ07	636	7	2	5	7	7
BJ09	231	7	2	2	6	2
CN02	102	7	2	2	5	2
DI805	18	3	2	2	1	2
BE07	232	5	5	2	1	5
BJ10	16	2	2	5	7	2