

**THESE
POUR LE DIPLOME D'ETAT
DE DOCTEUR EN PHARMACIE**

**Soutenue publiquement le 12 septembre 2023
Par M. RIHANI Emir Kaïs**

**APPLICATION DE MODELES D'INTELLIGENCE ARTIFICIELLE
A LA CLASSIFICATION DES MACROMYCETES**

Membres du jury :

Président : Nom, Prenom, titre et lieu de fonction

Directeur, conseiller de thèse : Nom, Prenom, titre et lieu de fonction

Assesseur(s) : Nom, Prenom, titre et lieu de fonction

Faculté de Pharmacie de Lille
3 Rue du Professeur Laguesse – 59000 Lille
03 20 96 40 40
<https://pharmacie.univ-lille.fr>

Université de Lille

Président
Premier Vice-président
Vice-présidente Formation
Vice-président Recherche
Vice-présidente Réseaux internationaux et européens
Vice-président Ressources humaines
Directrice Générale des Services

Régis BORDET
Etienne PEYRAT
Christel BEAUCOURT
Olivier COLOT
Kathleen O'CONNOR
Jérôme FONCEL
Marie-Dominique SAVINA

UFR3S

Doyen
Premier Vice-Doyen
Vice-Doyen Recherche
Vice-Doyen Finances et Patrimoine
Vice-Doyen Coordination pluriprofessionnelle et Formations sanitaires
Vice-Doyen RH, SI et Qualité
Vice-Doyenne Formation tout au long de la vie
Vice-Doyen Territoires-Partenariats
Vice-Doyenne Vie de Campus
Vice-Doyen International et Communication
Vice-Doyen étudiant

Dominique LACROIX
Guillaume PENEL
Éric BOULANGER
Damien CUNY
Sébastien D'HARANCY
Hervé HUBERT
Caroline LANIER
Thomas MORGENROTH
Claire PINÇON
Vincent SOBANSKI
Dorian QUINZAIN

Faculté de Pharmacie

Doyen
Premier Assesseur et Assesseur en charge des études
Assesseur aux Ressources et Personnels
Assesseur à la Santé et à l'Accompagnement
Assesseur à la Vie de la Faculté
Responsable des Services
Représentant étudiant

Delphine ALLORGE
Benjamin BERTIN
Stéphanie DELBAERE
Anne GARAT
Emmanuelle LIPKA
Cyrille PORTA
Honoré GUISE

Professeurs des Universités - Praticiens Hospitaliers (PU-PH)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
Mme	ALLORGE	Delphine	Toxicologie et Santé publique	81
M.	BROUSSEAU	Thierry	Biochimie	82
M.	DÉCAUDIN	Bertrand	Biopharmacie, Pharmacie galénique et hospitalière	81
M.	DINE	Thierry	Pharmacologie, Pharmacocinétique et Pharmacie clinique	81
Mme	DUPONT-PRADO	Annabelle	Hématologie	82
Mme	GOFFARD	Anne	Bactériologie - Virologie	82
M.	GRESSIER	Bernard	Pharmacologie, Pharmacocinétique et Pharmacie clinique	81
M.	ODOU	Pascal	Biopharmacie, Pharmacie galénique et hospitalière	80
Mme	POULAIN	Stéphanie	Hématologie	82
M.	SIMON	Nicolas	Pharmacologie, Pharmacocinétique et Pharmacie clinique	81
M.	STAELS	Bart	Biologie cellulaire	82

Professeurs des Universités (PU)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
M.	ALIOUAT	El Moukhtar	Parasitologie - Biologie animale	87
Mme	AZAROUAL	Nathalie	Biophysique - RMN	85
M.	BLANCHEMAIN	Nicolas	Pharmacotechnie industrielle	85
M.	CARNOY	Christophe	Immunologie	87
M.	CAZIN	Jean-Louis	Pharmacologie, Pharmacocinétique et Pharmacie clinique	86
M.	CHAVATTE	Philippe	Institut de Chimie Pharmaceutique Albert Lespagnol	86
M.	COURTECUISSÉ	Régis	Sciences végétales et fongiques	87
M.	CUNY	Damien	Sciences végétales et fongiques	87
Mme	DELBAERE	Stéphanie	Biophysique - RMN	85
Mme	DEPREZ	Rebecca	Chimie thérapeutique	86
M.	DEPREZ	Benoît	Chimie bioinorganique	85
M.	DUPONT	Frédéric	Sciences végétales et fongiques	87

M.	DURIEZ	Patrick	Physiologie	86
M.	ELATI	Mohamed	Biomathématiques	27
M.	FOLIGNÉ	Benoît	Bactériologie - Virologie	87
Mme	FOULON	Catherine	Chimie analytique	85
M.	GARÇON	Guillaume	Toxicologie et Santé publique	86
M.	GOOSSENS	Jean-François	Chimie analytique	85
M.	HENNEBELLE	Thierry	Pharmacognosie	86
M.	LEBEGUE	Nicolas	Chimie thérapeutique	86
M.	LEMDANI	Mohamed	Biomathématiques	26
Mme	LESTAVEL	Sophie	Biologie cellulaire	87
Mme	LESTRELIN	Réjane	Biologie cellulaire	87
Mme	MELNYK	Patricia	Chimie physique	85
M.	MILLET	Régis	Institut de Chimie Pharmaceutique Albert Lespagnol	86
Mme	MUHR-TAILLEUX	Anne	Biochimie	87
Mme	PERROY	Anne-Catherine	Droit et Economie pharmaceutique	86
Mme	ROMOND	Marie-Bénédicte	Bactériologie - Virologie	87
Mme	SAHPAZ	Sevser	Pharmacognosie	86
M.	SERGHERAERT	Éric	Droit et Economie pharmaceutique	86
M.	SIEPMANN	Juergen	Pharmacotechnie industrielle	85
Mme	SIEPMANN	Florence	Pharmacotechnie industrielle	85
M.	WILLAND	Nicolas	Chimie organique	86

Maîtres de Conférences - Praticiens Hospitaliers (MCU-PH)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
M.	BLONDIAUX	Nicolas	Bactériologie - Virologie	82
Mme	DEMARET	Julie	Immunologie	82
Mme	GARAT	Anne	Toxicologie et Santé publique	81
Mme	GENAY	Stéphanie	Biopharmacie, Pharmacie galénique et hospitalière	81
M.	LANNOY	Damien	Biopharmacie, Pharmacie galénique et hospitalière	80

Mme	ODOU	Marie-Françoise	Bactériologie - Virologie	82
-----	------	-----------------	---------------------------	----

Maîtres de Conférences des Universités (MCU)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
M.	AGOURIDAS	Laurence	Chimie thérapeutique	85
Mme	ALIOUAT	Cécile-Marie	Parasitologie - Biologie animale	87
M.	ANTHÉRIEU	Sébastien	Toxicologie et Santé publique	86
Mme	AUMERCIER	Pierrette	Biochimie	87
M.	BANTUBUNGI-BLUM	Kadiombo	Biologie cellulaire	87
Mme	BARTHELEMY	Christine	Biopharmacie, Pharmacie galénique et hospitalière	85
Mme	BEHRA	Josette	Bactériologie - Virologie	87
M.	BELARBI	Karim-Ali	Pharmacologie, Pharmacocinétique et Pharmacie clinique	86
M.	BERTHET	Jérôme	Biophysique - RMN	85
M.	BERTIN	Benjamin	Immunologie	87
M.	BOCHU	Christophe	Biophysique - RMN	85
M.	BORDAGE	Simon	Pharmacognosie	86
M.	BOSC	Damien	Chimie thérapeutique	86
M.	BRIAND	Olivier	Biochimie	87
Mme	CARON-HOUDE	Sandrine	Biologie cellulaire	87
Mme	CARRIÉ	Hélène	Pharmacologie, Pharmacocinétique et Pharmacie clinique	86
Mme	CHABÉ	Magali	Parasitologie - Biologie animale	87
Mme	CHARTON	Julie	Chimie organique	86
M.	CHEVALIER	Dany	Toxicologie et Santé publique	86
Mme	DANEL	Cécile	Chimie analytique	85
Mme	DEMANCHE	Christine	Parasitologie - Biologie animale	87
Mme	DEMARQUILLY	Catherine	Biomathématiques	85
M.	DHIFLI	Wajdi	Biomathématiques	27
Mme	DUMONT	Julie	Biologie cellulaire	87
M.	EL BAKALI	Jamal	Chimie thérapeutique	86
M.	FARCE	Amaury	Institut de Chimie Pharmaceutique Albert Lespagnol	86

M.	FLIPO	Marion	Chimie organique	86
M.	FURMAN	Christophe	Institut de Chimie Pharmaceutique Albert Lespagnol	86
M.	GERVOIS	Philippe	Biochimie	87
Mme	GOOSSENS	Laurence	Institut de Chimie Pharmaceutique Albert Lespagnol	86
Mme	GRAVE	Béatrice	Toxicologie et Santé publique	86
Mme	GROSS	Barbara	Biochimie	87
M.	HAMONIER	Julien	Biomathématiques	26
Mme	HAMOUDI-BEN YELLES	Chérifa-Mounira	Pharmacotechnie industrielle	85
Mme	HANNOTHIAUX	Marie-Hélène	Toxicologie et Santé publique	86
Mme	HELLEBOID	Audrey	Physiologie	86
M.	HERMANN	Emmanuel	Immunologie	87
M.	KAMBIA KPAKPAGA	Nicolas	Pharmacologie, Pharmacocinétique et Pharmacie clinique	86
M.	KARROUT	Younes	Pharmacotechnie industrielle	85
Mme	LALLOYER	Fanny	Biochimie	87
Mme	LECOEUR	Marie	Chimie analytique	85
Mme	LEHMANN	Hélène	Droit et Economie pharmaceutique	86
Mme	LELEU	Natascha	Institut de Chimie Pharmaceutique Albert Lespagnol	86
Mme	LIPKA	Emmanuelle	Chimie analytique	85
Mme	LOINGEVILLE	Florence	Biomathématiques	26
Mme	MARTIN	Françoise	Physiologie	86
M.	MOREAU	Pierre-Arthur	Sciences végétales et fongiques	87
M.	MORGENROTH	Thomas	Droit et Economie pharmaceutique	86
Mme	MUSCHERT	Susanne	Pharmacotechnie industrielle	85
Mme	NIKASINOVIC	Lydia	Toxicologie et Santé publique	86
Mme	PINÇON	Claire	Biomathématiques	85
M.	PIVA	Frank	Biochimie	85
Mme	PLATEL	Anne	Toxicologie et Santé publique	86
M.	POURCET	Benoît	Biochimie	87
M.	RAVAUX	Pierre	Biomathématiques / Innovations pédagogiques	85

Mme	RAVEZ	Séverine	Chimie thérapeutique	86
Mme	RIVIÈRE	Céline	Pharmacognosie	86
M.	ROUMY	Vincent	Pharmacognosie	86
Mme	SEBTI	Yasmine	Biochimie	87
Mme	SINGER	Elisabeth	Bactériologie - Virologie	87
Mme	STANDAERT	Annie	Parasitologie - Biologie animale	87
M.	TAGZIRT	Madjid	Hématologie	87
M.	VILLEMAGNE	Baptiste	Chimie organique	86
M.	WELTI	Stéphane	Sciences végétales et fongiques	87
M.	YOUS	Saïd	Chimie thérapeutique	86
M.	ZITOUNI	Djamel	Biomathématiques	85

Professeurs certifiés

Civ.	Nom	Prénom	Service d'enseignement
Mme	FAUQUANT	Soline	Anglais
M.	HUGES	Dominique	Anglais
M.	OSTYN	Gaël	Anglais

Professeurs Associés

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
M.	DAO PHAN	Haï Pascal	Chimie thérapeutique	86
M.	DHANANI	Alban	Droit et Economie pharmaceutique	86

Maîtres de Conférences Associés

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
Mme	CUCCHI	Malgorzata	Biomathématiques	85
M.	DUFOSSEZ	François	Biomathématiques	85
M.	FRIMAT	Bruno	Pharmacologie, Pharmacocinétique et Pharmacie clinique	85
M.	GILLOT	François	Droit et Economie pharmaceutique	86
M.	MASCAUT	Daniel	Pharmacologie, Pharmacocinétique et Pharmacie clinique	86

M.	MITOUMBA	Fabrice	Biopharmacie, Pharmacie galénique et hospitalière	86
M.	PELLETIER	Franck	Droit et Economie pharmaceutique	86
M.	ZANETTI	Sébastien	Biomathématiques	85

Assistants Hospitalo-Universitaire (AHU)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
Mme	CUVELIER	Élodie	Pharmacologie, Pharmacocinétique et Pharmacie clinique	81
M.	GRZYCH	Guillaume	Biochimie	82
Mme	LENSKI	Marie	Toxicologie et Santé publique	81
Mme	HENRY	Héloïse	Biopharmacie, Pharmacie galénique et hospitalière	80
Mme	MASSE	Morgane	Biopharmacie, Pharmacie galénique et hospitalière	81

Attachés Temporaires d'Enseignement et de Recherche (ATER)

Civ.	Nom	Prénom	Service d'enseignement	Section CNU
Mme	GEORGE	Fanny	Bactériologie - Virologie / Immunologie	87
Mme	N'GUESSAN	Cécilia	Parasitologie - Biologie animale	87
M.	RUEZ	Richard	Hématologie	87
M.	SAIED	Tarak	Biophysique - RMN	85
M.	SIEROCKI	Pierre	Chimie bioinorganique	85

Enseignant contractuel

Civ.	Nom	Prénom	Service d'enseignement
M.	MARTIN MENA	Anthony	Biopharmacie, Pharmacie galénique et hospitalière

Faculté de Pharmacie de Lille

3 Rue du Professeur Laguesse – 59000 Lille

03 20 96 40 40

<https://pharmacie.univ-lille.fr>

L'Université n'entend donner aucune approbation aux opinions émises dans les thèses ; celles-ci sont propres à leurs auteurs.

J'adresse mes sincères remerciements à :

La Dream Team de tous ceux que je vais remercier

Table des matières

1	Liste des abréviations	17
2	Introduction	18
2.1	Propos liminaire	18
2.2	But de l'étude	18
2.3	Etat de l'art des lots de données mycologiques	19
3	Création du lot de données	20
3.1	Configuration matérielle et logicielle	20
3.2	Principes de conception d'un lot de données synthétiques	20
3.2.1	Principes généraux	20
3.2.2	Principes de génération des paramètres quantitatifs	21
3.2.3	Principes de génération des paramètres qualitatifs	27
4	Principes de l'apprentissage machine	28
4.1	Jeux de données	28
4.2	Modèles utilisés	29
4.2.1	Analyses discriminantes	29
4.2.2	Modèle additif généralisé	30
4.2.3	Arbres de décision	30
4.2.4	Forêts aléatoires	30
4.3	Optimisation par plans d'expérience (DOE)	30
4.4	Evaluation des performances des modèles	31
5	Apprentissage machine et classification binaire	34
5.1	Analyse exploratoire des données (EDA)	34
5.2	Modèles de la librairie <i>caret</i> : optimisation et sélection	37
5.2.1	Modèles d'analyse discriminante linéaire	38
5.2.2	Modèle additif généralisé	39
5.2.3	Modèles d'arbres de décision	39
5.2.4	Forêts aléatoires	42
5.3	Résultats	47
5.3.1	Protocole d'évaluation	47
5.3.2	Performances des modèles de forêts aléatoires	47
6	Apprentissage machine et classification multiclasse	49
6.1	Classification par familles	49
6.1.1	Modèles d'arbres de décision	49
6.1.2	Forêts aléatoires	51
6.1.3	Résultats	54
6.2	Classification par espèce	55

7	Robustesse de la classification	56
8	Références bibliographiques	57
A	Annexe : développement d'un algorithme de génération de lot synthétique	60
B	Annexe : outils d'analyse exploratoire des données (EDA)	61
C	Annexe : développement des algorithmes d'apprentissage machine	62
C.1	Initialisation	62
C.2	Création des jeux d'entraînement, optimisation et évaluation	62

1 Liste des abréviations

AI : *Artificial Intelligence* (intelligence artificielle)
AUC : *Area Under Curve* (aire sous la courbe)
CART : *Classification And Regression Tree* (arbre de classification et de régression)
CSV : *Comma Separated Values* ([fichier de] valeurs séparées par des virgules)
DOE : *Design of Experiments* (plan d'expériences)
EDA : *Exploratory Data Analysis* (analyse exploratoire des données)
ESE : *Enhanced Stochastic Evolutionary* ([algorithme] évolutionnaire stochastique amélioré)
GAM : *Generalized Additive Model* (modèle additif généralisé)
IA : Intelligence Artificielle
Kd-tree : *K-dimensional tree* (arbre k-dimensionnel)
LDA : *Linear Discriminant Analysis* (analyse linéaire discriminante)
LHS : *Latin Hypercube Sample* (échantillonnage par hypercube latin)
LOESS : *LOcally Estimated Scatterplot Smoothing* (régression locale)
NN : *Nearest Neighbour* (plus proche voisin)
NOLH : *Nearly Orthogonal Latin Hypercube* (hypercube latin quasi orthogonal)
PDA : *Penalized Discriminant Analysis* (analyse discriminante pénalisée)
RF : *Random Forest* (forêt aléatoire)
ROC : *Receiver Operating Characteristic* (fonction d'efficacité du récepteur)
XGB : *eXtreme Gradient Boosting*

2 Introduction

2.1 Propos liminaire

L'identification des macromycètes est un sujet difficile, ne devant évidemment pas être pris à la légère. Les espèces rencontrées varient considérablement d'un écosystème à un autre, d'un continent à un autre, et aucun lot de données ni ouvrage sur les champignons ne saurait couvrir toute la diversité du monde fongique.

Le lot de données mycologiques constitué dans cette étude, bien que constituant l'un des lots en libre accès les plus complets du domaine de la *data science*, n'est bien entendu pas exhaustif.

Ce lot se concentre exclusivement sur les champignons habituellement rencontrés au Nord de la France. Plusieurs genres, parfois très connus, ne sont pas présents, parmi lesquels nous pouvons par exemple citer le genre *psylocybe*, connu pour ses propriétés psychédéliques. Certains critères pourront également varier de manière considérable selon le stade de maturité du champignon : alors que les chapeaux vert-olive de l'*Amanita phalloides* mature sont faciles à reconnaître, les spécimens jeunes sont blancs et pourraient facilement être confondus avec des espèces comestibles (par exemple du genre *Agaricus*).

L'ingestion de certains de ces champignons est *mortelle*, même à de petites doses. Le diagnostic de l'intoxication fongique peut être difficile, et parfois trop tardif pour un traitement efficace. Des composés toxiques tels que les amanitines ne sont pas altérés ou détruits par cuisson ou congélation, et seront absorbés par l'intestin, avant de passer dans la circulation sanguine afin d'être filtrés par le foie, détruisant les cellules hépatiques, puis excrétées dans l'intestin, réabsorbées, refiltrées... chaque passe détruisant les cellules hépatiques ayant survécu à la précédente, dans un cycle connu sous le nom de réabsorption hépato-entérique.

Il ne faut jamais, *sous aucune circonstance*, utiliser ce type de lot de données afin de déterminer si un champignon est comestible ou non.

2.2 But de l'étude

L'identification des plantes et champignons est un problème de classification classique, qui est habituellement effectué manuellement à l'aide de clés d'identification. La plupart de ces clés sont basées sur un processus utilisant des arbres décisionnels, ce qui semble logique car rappelant la logique en arbre de l'évolution. Quoique séduisant, cet argument rencontre quelques limites :

La première limite est le nombre de chaînons manquants. Certaines espèces sont évidemment éteintes, ce qui signifie que certaines branches et nœuds de l'arbre phylogénétique sont manquants, ce qui peut compliquer l'analyse quand deux espèces apparentées ont un nombre élevé de chaînons

et nœuds communs manquants. Certaines similarités entre espèces peuvent également ne pas être identifiables de façon macroscopique.

La seconde limite, plus profonde, est la logique inhérente au processus évolutif. Deux phénomènes antagonistes sont en jeu : convergence et divergence évolutives. Ces deux phénomènes sont liés à la nécessaire adaptation des espèces à leurs environnements. La divergence évolutive explique par exemple la diversité des mammifères : les chauves-souris, baleines et chevaux sont apparentés, mais ont des aspects très dissemblables en raison de leur adaptation à des environnements très différents. D'un autre côté, la convergence évolutive explique la similarité entre l'aile de la chauve-souris et l'aile de l'abeille. Toutefois, malgré leur apparente dissimilarité, l'aile de la chauve-souris est plus proche de la main humaine ou de la nageoire de la baleine que de l'aile de l'abeille. La façon la plus fiable pour évaluer le processus évolutif et trouver les liens phylogénétiques de la manière la plus précise possible est l'analyse des génomes : les caractéristiques visibles peuvent être trompeuses. Malheureusement, ces caractéristiques sont souvent les seules aisément identifiables.

Le troisième problème est le critère principal de la classification. Ce critère peut être lié ou non au processus évolutif ou aux critères visibles, surtout si ce critère principal est vague. Le critère de comestibilité ou de non-comestibilité retenu pour les lots de données mycologiques usuellement utilisés en *data science* souffre de ce problème : il est essentiellement centré sur la toxicité contre les humains, or de nombreux mécanismes de toxicité peuvent exister, et une toxicité ou non-toxicité d'un métabolite fongique ou végétal peut être liée à des variations métaboliques très ténues entre une espèce et une autre.

Pour ces raisons parmi d'autres, la logique arborescente, bien qu'utilisée habituellement dans l'identification des champignons et des plantes, et souvent justifiée par la nature arborescente du processus évolutif, pourrait ne pas nécessairement être l'approche optimale à la classification des espèces basée sur des critères macroscopiques. Le but de cette étude est d'effectuer cette tâche de classification basée sur des indices visuels limités, et d'évaluer les performances relatives de différentes stratégies de classification.

2.3 Etat de l'art des lots de données mycologiques

Le tout premier lot de données mycologiques en libre accès mentionné en *data science* est probablement le *Mushroom Dataset* créé par Jeff Schlimmer en 1987.¹

Un lot de données plus conséquent a été publié par Dennis Wagner en 2021² et mis en libre accès sous le nom de *Secondary Mushroom Dataset*.

3 Création du lot de données

3.1 Configuration matérielle et logicielle

Le code d'apprentissage machine, les méthodes d'évaluation, ainsi que cette thèse ont été rédigés sur l'équipement suivant :

- CPU : AMD Ryzen 5 5600G
- RAM : 2x16 Go DDR4-3200
- SSD : Crucial P5 M2 NVMe
- OS : Xubuntu Linux 22.04.2 LTS
- R : version 4.2.2 (2022)
- RStudio : version 2022.7.2.576, "Spotted Wakerobin"
- Librairies : tidyverse³ (v1.3.2), microbenchmark⁴ (v1.4.9), SPlit⁵ (v1.2), MASS⁶ (v7.3.58.2), caret⁷ (v6.0.93), ?GGally?⁸ (v2.1.2), ?mda?⁹ (v0.5.3), rpart¹⁰ (v4.1.19), ?plyr?¹¹ (v1.8.8), C50¹² (v0.1.8), party¹³ (v1.3.11), ranger¹⁴ (v0.14.1), e1071¹⁵ (v1.7.13), rFens¹⁶ (v5.0.0), Rborist¹⁷ (v0.3.2), rmarkdown¹⁸ (v2.20), knitr¹⁹ (v1.41), ggpubr²⁰ (v0.6.0), DiceDesign²¹ (v1.9), DiceEval²² (v1.5.1), bookdown²³ (v0.32).

3.2 Principes de conception d'un lot de données synthétiques

3.2.1 Principes généraux

Un lot de données synthétiques est un lot de données généré par un algorithme, par opposition aux lots de données issus d'une collecte effectuée en "vie réelle".

Trois stratégies sont usuellement utilisées :

- Données factices (*dummy data*) : l'ensemble des données est généré aléatoirement.
- Données générées à partir de règles (*rule-based data*) : l'ensemble des données est généré suivant des lois définies au préalable (distribution, valeurs moyennes, minimales, maximales. . .)
- Données générées par intelligence artificielle (*AI generated*) : l'ensemble des données est généré suivant des lois extraites par l'IA suite à l'analyse d'un échantillon de données obtenues en "vie réelle".

Les données générées par ces stratégies peuvent être de type variés, que nous pouvons grossièrement regrouper en données alphanumériques (quantitatives et qualitatives) et en données d'imagerie.

Pour des raisons pratiques et de maturité des technologies disponibles à l'heure actuelle, la méthode retenue pour créer le lot de données exploité dans notre étude sera la génération de données alphanumériques à partir de règles, extraites d'ouvrages mycologiques de référence.^{24–26}

3.2.2 Principes de génération des paramètres quantitatifs

Dans le cadre de cette étude, les variables quantitatives générées aléatoirement sont :

- La longueur du stipe L_S ,
- Le diamètre du stipe D_S ,
- Le diamètre du chapeau D_C .

En première approximation, nous pouvons considérer que toutes ces valeurs sont intrinsèquement liées à la croissance du champignon. Ces trois variables peuvent, dans l'absolu, être susceptibles de varier indépendamment des autres au cours de la croissance du champignon, les variables L_S , D_S et D_C obéissant alors aux lois suivantes :

$$\begin{cases} L_S = L_{Smax} \cdot F_{Ls} \\ D_S = D_{Smax} \cdot F_{Ds} \\ D_C = D_{Cmax} \cdot F_{Dc} \end{cases}$$

Avec :

- L_{Smax} , D_{Smax} et D_{Cmax} les valeurs maximales de longueur de stipe, diamètre du stipe et diamètre de chapeau de chaque variété de champignon, extraites de la littérature,
- F_{Ls} , F_{Ds} , F_{Dc} des variables générées aléatoirement dans l'intervalle $]0; 1]$, et représentatives de la croissance du spécimen.

Toutefois, la recherche bibliographique sur la cinétique de croissance des sporophores n'a pas permis de distinguer de différences de la cinétique de croissance de chacun de ces trois paramètres. Nous supposons donc, en première approximation, que la croissance du stipe en longueur et en largeur, ainsi que la croissance du chapeau s'effectuent à des vitesses identiques. Nous obtenons par conséquent :

$$F_{Ls} = F_{Ds} = F_{Dc} = F_T$$

Avec F_T un facteur représentatif de la taille globale de chaque spécimen généré aléatoirement.

Ainsi, le problème de génération de nos trois variables aléatoires se simplifie en un problème de génération d'une seule variable aléatoire : le facteur de taille de chaque spécimen. Un certain nombre de distributions d'intérêt sont susceptibles d'être utilisées afin de générer des facteurs de taille F_T aléatoires, il convient donc de définir le cahier des charges de la distribution la plus adaptée au sujet de cette étude.

Les critères de sélection retenus afin de choisir la loi la plus appropriée sont :

- Efficience calculatoire,
- Distribution continue,
- Distribution bornée, ou aisément normalisable sur un intervalle $[0; 1]$,
- Distribution asymétrique.

Le premier critère n'est, en pratique, pas un facteur limitant, les temps de calcul pour la génération d'un nombre de facteurs de taille F_T suffisant étant typiquement inférieurs à 200 ms (pour 10^6 facteurs générés) avec la plupart des distributions d'intérêt (voir figure 1).

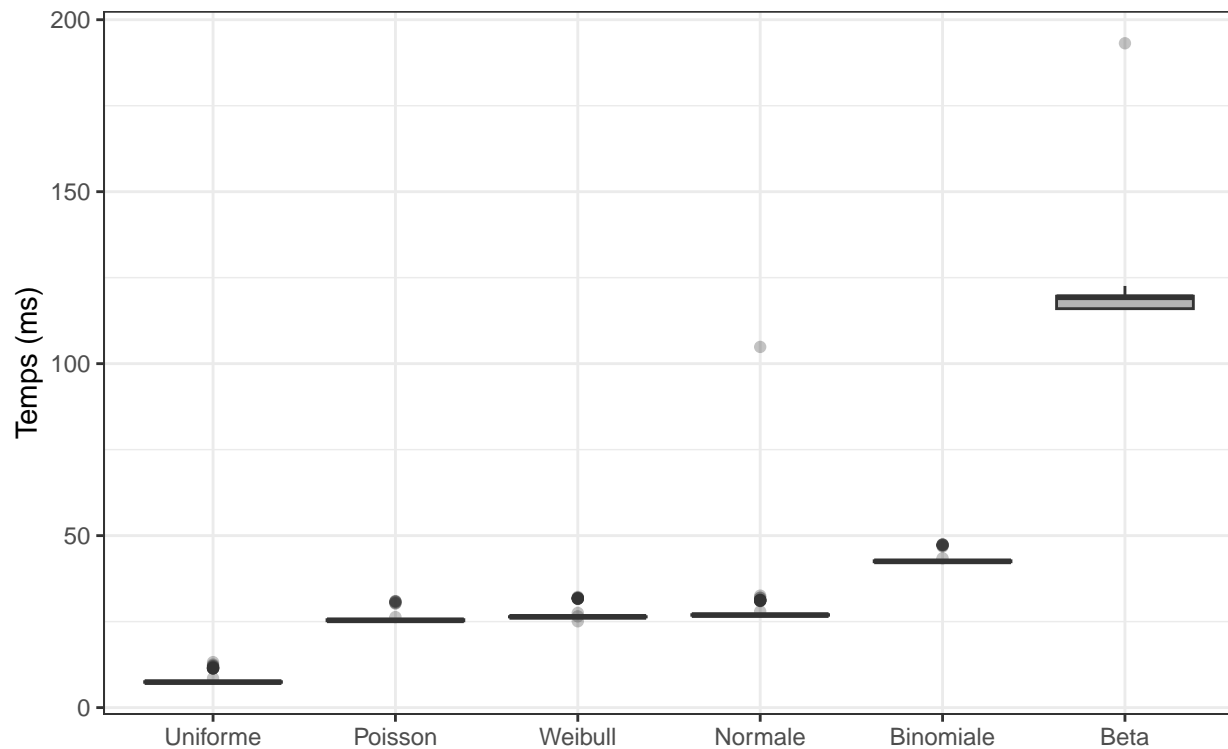


Figure 1: Temps de calcul des principales distributions d'intérêt pour $1e+06$ facteurs, (100 iter.)

Les critères de continuité et de normalité n'appellent que peu de commentaires. Ces critères permettent simplement de garantir la possibilité d'une infinité de valeurs dimensionnelles, dans l'intervalle considéré. Le critère de continuité proscribit toutefois l'utilisation de lois de distributions discrètes telles que la loi binomiale ou la loi de Poisson, et celui de normalité écarte des distributions telles que la loi de Weibull, dont la normalisation est parfois délicate.

Le critère d'asymétrie est un critère permettant de tenir compte des différents paramètres pouvant impacter la distribution de taille des spécimens prélevés, parmi lesquels :

- Différences de cinétique de croissance d'une famille à une autre,
- Particularités de la croissance fongique, notamment par la croissance hyphale,^{27,28}
- Probabilité de prélèvement variable selon la taille du spécimen (par difficulté de détection, considérations éthiques, intérêt mycologique ou gastronomique. . .).

Le premier paramètre évoqué précédemment n'a pu être exploité dans le cadre de cette étude en raison du manque de données concernant les cinétiques relatives de croissance des sporophores des différentes familles de macromycètes. Le modèle que nous proposons permet toutefois des développements ultérieurs dans ce domaine.

Les deux derniers paramètres permettent de supposer que la distribution de taille des spécimens d'une même espèce à l'issue d'une récolte en vie réelle ne sera pas symétrique, d'une part en raison de la rapidité de la croissance fongique, et d'autre part parce que le prélèvement se fera préférentiellement en épargnant les spécimens de petite taille.

Ainsi, la génération de la variable aléatoire F_T obéira idéalement à une loi de distribution asymétrique vers la droite ($G_1 < 0$). Ce critère d'asymétrie écarte par conséquent les lois de distribution symétriques telles que la loi normale ou la loi uniforme.



Figure 2: Exemples de distributions de la loi uniforme (à gauche), binomiale (au centre) et normale (à droite)



Figure 3: Exemples de distributions de la loi de Poisson (à gauche), de Weibull (au centre) et bêta (à droite)

En raison des contraintes imposées précédemment ainsi que de par sa grande polyvalence²⁹, la loi retenue dans le cadre de cette étude pour la génération des facteurs de taille aléatoires (F_T) est une loi bêta non-centrale, définie comme la fonction de distribution de :^{29,30}

$$X = \frac{\chi_{2\alpha}^2(\lambda)}{\chi_{2\alpha}^2(\lambda) + \chi_{2\beta}^2}$$

Avec, comme paramètres définis empiriquement pour cette étude :

$$\begin{cases} \alpha = 6 F_c & (shape1) \\ \beta = 4 & (shape2) \\ \lambda = F_c/2 & (ncp) \end{cases}$$

F_c est ici défini comme un facteur de croissance permettant de rendre compte de la cinétique de croissance de chaque variété d'une part, et du prélèvement préférentiel des spécimens de plus grande taille d'autre part, comme l'illustre la figure 4.

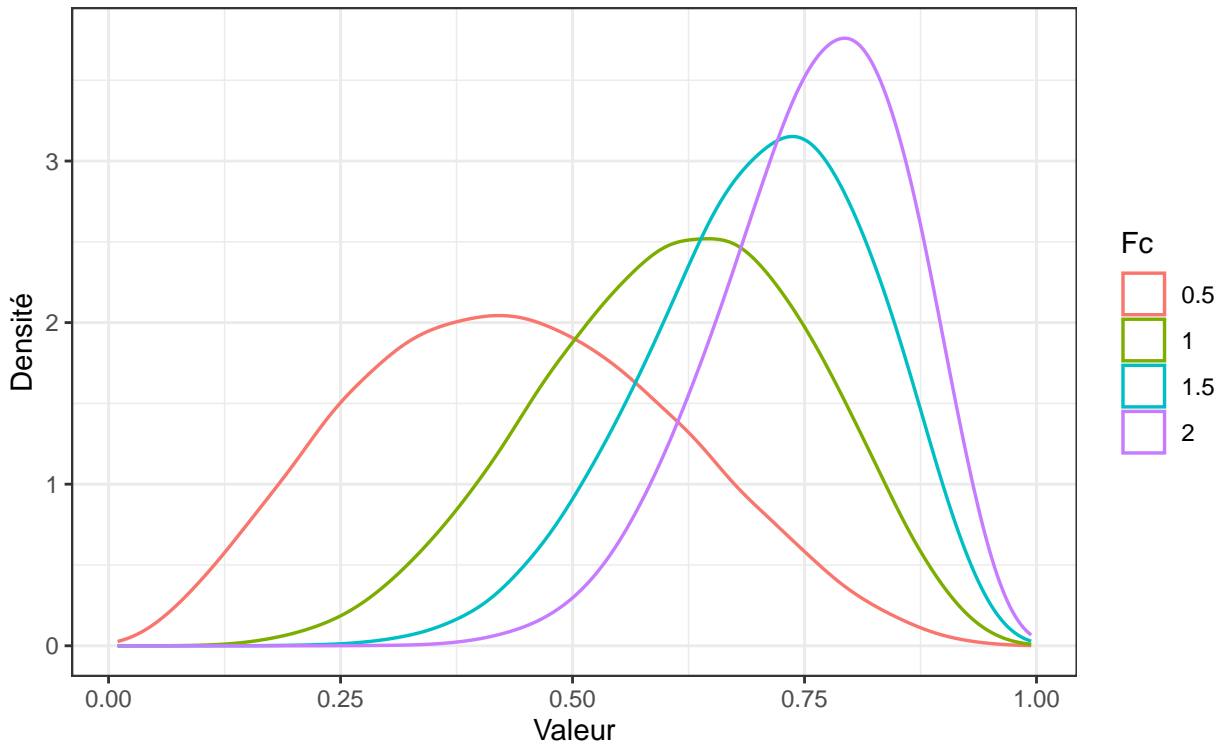


Figure 4: Distribution de différentes lois bêta, en fonction du facteur de croissance F_c

Le modèle défini à ce stade impose une stricte proportionnalité entre diamètre du chapeau D_c , diamètre du stipe D_s et longueur du stipe L_s .

Dans un souci de réalisme, il apparaît souhaitable d'améliorer ce modèle mathématique en y ajoutant un facteur de dispersion, afin de proposer le modèle suivant :

$$\begin{cases} L_S = L_{Smax} \cdot F_T \cdot \delta_{Ls} & \text{avec } \delta_{Ls} \sim \mathcal{N}(\mu = 1; \sigma = 0.05) \\ D_S = D_{Smax} \cdot F_T \cdot \delta_{Ds} & \text{avec } \delta_{Ds} \sim \mathcal{N}(\mu = 1; \sigma = 0.05) \\ D_C = D_{Cmax} \cdot F_T \cdot \delta_{Dc} & \text{avec } \delta_{Dc} \sim \mathcal{N}(\mu = 1; \sigma = 0.05) \end{cases}$$

L'impact de la dispersion sur la distribution des paramètres de taille L_S , D_S et D_C est illustré par les figures 5 et 6.

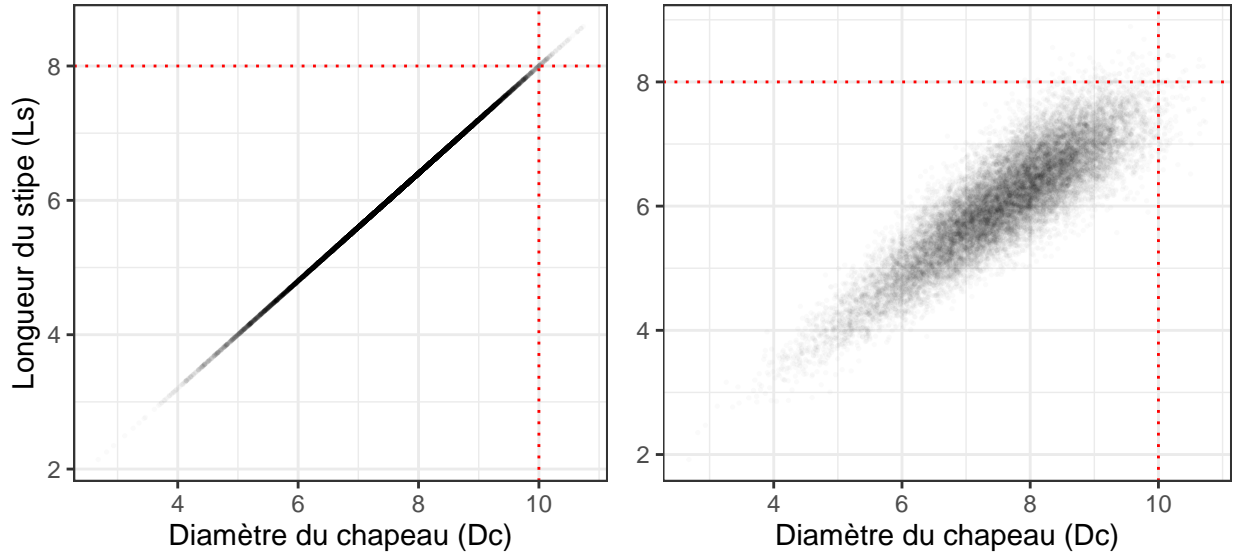


Figure 5: Nuages de points de 2 paramètres de taille (L_S et D_C), sans dispersion (à gauche) et avec dispersion (à droite), pour 20000 champignons

La dispersion ainsi créée permet ainsi de créer de légères variations des rapports entre les différents paramètres de taille, tout en se situant à proximité de la première bissectrice et majoritairement dans la zone 50-90% de la taille maximale comme illustré par la figure 6). Cette dispersion autorise par ailleurs l'existence d'une faible proportion de spécimens dépassant les valeurs dimensionnelles maximales généralement admises par la littérature.

[faire calcul sur R pour relier aux chiffres]

Une simulation de Monte Carlo unidimensionnelle effectuée sur 10^5 spécimens nous permet ainsi d'évaluer la proportion de spécimens "hors normes" dépassant la valeur dimensionnelle maximale à environ 0.4 % (cf. figure 7). La même simulation nous permet d'évaluer que la proportion de spécimens "exceptionnels", dépassant de plus de 10% cette valeur maximale, sera quant à elle inférieure à 0.5 %.

3.2.2.0.1 CALCULS A REVERIFIER, PAS COHERENT....

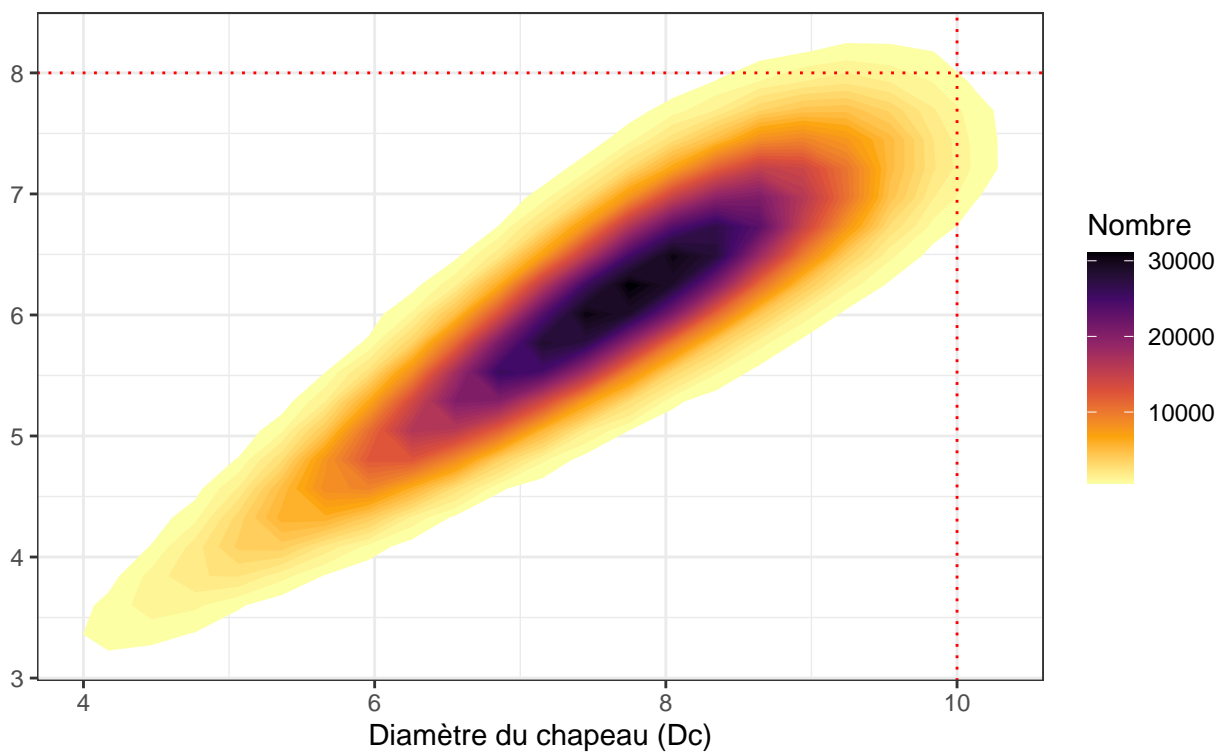


Figure 6: Diagramme de densité de 2 paramètres de taille, avec dispersion

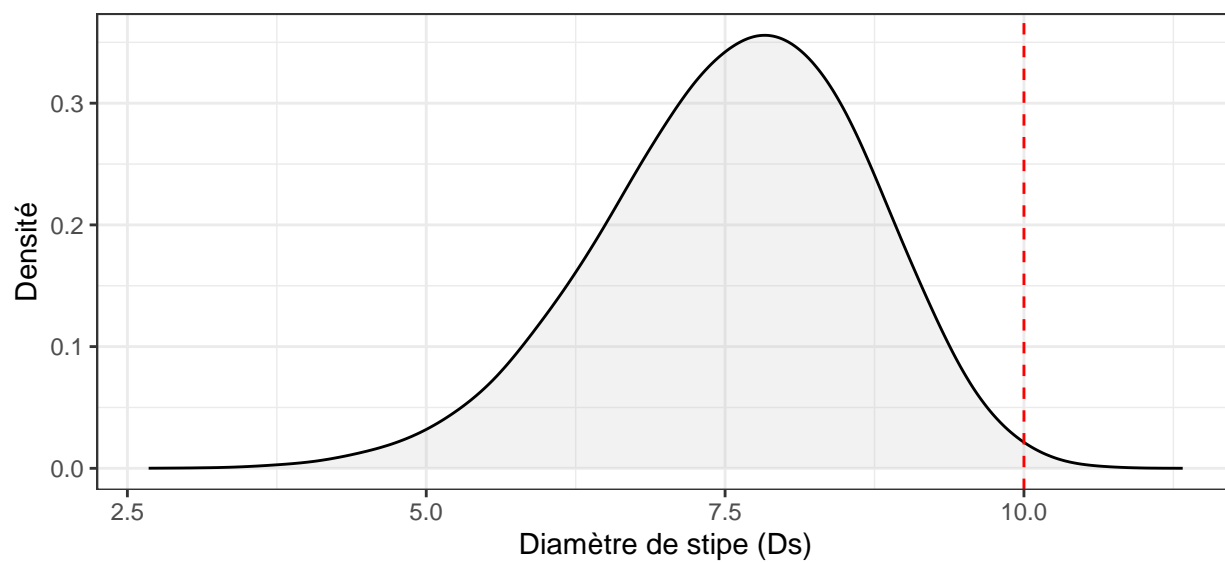


Figure 7: Distribution du diamètre de stipe D_s , pour $D_{smax} = 10$

3.2.3 Principes de génération des paramètres qualitatifs

La génération des paramètres qualitatifs, tels que la couleur des spores ou le type d'hyménophore, est nettement moins complexe que celle des paramètres quantitatifs.

L'ensemble des valeurs quantitatives possibles pour un critère et pour une variété donnée est insérée dans un vecteur de valeurs, et une valeur sera tirée aléatoirement parmi celles de ce vecteur pour caractériser chaque spécimen.

4 Principes de l'apprentissage machine

4.1 Jeux de données

Le déroulement de l'apprentissage machine se décompose conceptuellement en trois étapes principales, mettant en jeu trois lots de données distincts :

1. Entraînement : le modèle d'apprentissage est exposé à un *jeu de données d'entraînement* (*training data set*), censé être représentatif (cf. section 3.2.1) des données auquel le modèle sera exposé en utilisation réelle.
2. Validation : le modèle d'apprentissage développé à l'étape précédente, sera soumis à un *jeu de données de validation* (*validation data set*). Les prédictions (ex: comestibilité, espèce...) proposées par le modèle d'apprentissage sur la base des informations contenues dans le lot de données de validation (ex : dimensions, couleurs, morphologie du champignon...) sont comparées avec les valeurs réelles (ex : comestibilité, espèce...), ce qui permet d'évaluer les performances prédictives du modèle proposé en fonction des indicateurs retenus (spécificité, sensibilité, F1-score, temps de calcul...). Les étapes d'apprentissage et de validation sont répétées de manière itérative en explorant l'ensemble des paramètres de configuration du modèle (hyperparamètres) – idéalement en suivant un plan d'expériences – à fins d'optimisation.
3. Test : les performances du meilleur modèle (avec hyperparamètres optimaux), sélectionné à l'issue de l'étape de validation, sont évaluées vis-à-vis d'un *jeu de données test* (*test ou holdout data set*).

Certains contextes particuliers, tels que les tournois de *data science*, mettront en œuvre une quatrième fraction destinée au classement final des algorithmes (et des équipes). Certaines méthodes basées sur des plis multiples (*n-folds*) utilisent également une séparation flexible entre lots d'entraînement et de validation. Ces cas particuliers ne feront pas l'objet de cette étude.

La séparation entre étapes d'optimisation et de test peut sembler artificielle. Le problème est en partie lié à un flou sémantique : si l'étape initiale d'entraînement ou d'apprentissage ne pose que peu de problèmes conceptuels, l'étape intermédiaire, dite de *validation* correspond en réalité à une étape d'*optimisation* du modèle et de ses hyperparamètres. Par ailleurs, l'étape finale de *test* sera parfois qualifiée d'étape de *validation* dans la littérature, ce qui peut entretenir la confusion entre ces étapes.³¹

Une distinction sémantique plus nette entre phases d'*apprentissage*, d'*optimisation* et de *test* permet de comprendre plus aisément le fondement épistémologique de cette dernière phase pouvant parfois sembler superflue : l'optimisation effectuée lors de l'étape de validation aboutit à un modèle potentiellement biaisé (problème dit d'*overfitting*) vis-à-vis du jeu de données utilisé comme référence lors de cette étape. Seule une exposition du modèle à des données n'ayant jamais servi à son entraînement ou son optimisation permettra réellement d'évaluer avec précision son caractère prédictif, donc sa validité.

Dans un souci de clarté, nous utiliserons les termes lots et de phases d'entraînement, d'optimisation et d'évaluation dans la suite de cette étude.

Les phases d'entraînement, d'optimisation et d'évaluation utilisent chacune un lot de données spécifique. Chacun de ces lots de données est habituellement obtenu suite à dichotomies successives du lot de données initial, avec des proportions variables :

1. Découpage du jeu de données initial, en un jeu d'évaluation d'une part, et un jeu d'entraînement & optimisation d'autre part,
2. Découpage du jeu de données entraînement & optimisation, en un jeu d'entraînement et un jeu d'optimisation.

[Schéma Split Apprentissage/Optimisation/Validation]

Le rapport de taille entre jeux de données entraînement, optimisation, évaluation de cette étude suit la loi $p : \sqrt{p} : \sqrt{p} + 1$, avec p le nombre de coefficients du modèle.³²

Pour un nombre de coefficients compris entre 100 et 200, cette règle nous conduit à retenir :

$$\begin{cases} R_{entr} = 85 \pm 2\% \\ R_{opti} = 7 \pm 1\% \\ R_{eval} = 8 \pm 1\% \end{cases}$$

Ce rapport 85:7:8 peut, dans la pratique, être obtenu par réalisation de deux découpages successifs suivant des rapports 92:8.

Dans cette étude, la division de ces trois jeux de données utilise une méthode de découpage basée sur les points-supports³³ (*support-points based splitting*) exploitant un algorithme du plus proche voisin (NN : *Nearest Neighbour*), basé sur un arbre Kd (*k-dimensional tree*), afin d'optimiser la représentativité des jeux de données par rapport à ceux pouvant être obtenus par un découpage aléatoire.³⁴

4.2 Modèles utilisés

4.2.1 Analyses discriminantes

[discriminant correspondence analysis??]

<https://rpubs.com/markloessi/505575>

https://en.wikipedia.org/wiki/Altman_Z-score

<https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b>

https://scikit-learn.org/stable/modules/lda_qda.html

Cette étude propose plusieurs classifieurs linéaires s'appuyant sur des méthodes d'analyse discriminante :

- un modèle basé sur l'analyse discriminante linéaire (*Linear Discriminant Analysis*, LDA),
- un modèle basé sur l'analyse discriminante pénalisée (*Penalized Discriminant Analysis*, PDA).

L'analyse linéaire discriminante (LDA) est une méthode utilisée en statistiques et en data science pour trouver une combinaison linéaire d'éléments qui caractérisent des éléments, afin de créer un classifieur linéaire, ou d'effectuer des réductions de dimensionnalité. Cet algorithme fonctionne en créant des combinaisons linéaires (fonctions discriminantes) de prédicteurs. [A FINIR]

L'analyse discriminante pénalisée [A FINIR]

4.2.2 Modèle additif généralisé

gamLoess

4.2.3 Arbres de décision

rpart, Ctree???, c50tree, Rpartcost

4.2.4 Forêts aléatoires

rFerns, Rborist, ranger

4.3 Optimisation par plans d'expérience (DOE)

Certains modèles nécessiteront une optimisation de leurs hyperparamètres, qui relève du domaine des plans d'expérience. De nombreux plans et stratégies sont envisageables, le choix dépendra en partie des caractéristiques du processus à optimiser.

En effet, l'optimisation des paramètres d'un modèle informatique présente quelques particularités notables :

- La réalisation d'une expérience supplémentaire a un coût faible,
- Il peut exister plusieurs métriques coexistantes,
- La fonction de réponse peut s'avérer relativement complexe.

Ces particularités imposent d'explorer de manière méthodique la totalité de l'espace expérimental. Il existe une multitude de méthodes permettant de générer des plans expérimentaux, dits SFD (*Space Filling Design*), permettant d'optimiser l'occupation de l'espace expérimental. La méthode retenue pour cette étude sera celle des hypercubes latins, en raison de son utilisation répandue³⁵ et de sa simplicité conceptuelle.

La méthode des hypercubes latins est une extension du principe des carrés latins. Un carré latin est une grille $n \times n$, remplie de n éléments distincts arrangés de sorte que chaque ligne et chaque colonne ne contienne qu'un seul exemplaire de chacun des n éléments. Dans le domaine des plans d'expériences, l'application des carrés latins revient à diviser un domaine expérimental bidimensionnel en une grille $n \times n$, et à placer une expérience et une seule sur chaque ligne et chaque colonne.

L'application du concept de carré latin dans un domaine expérimental à trois dimensions aboutit au cube latin. La généralisation dans un espace n -dimensionnel mène au concept d'hypercube latin.

De nombreux plans expérimentaux basés sur les hypercubes latins peuvent être générés. Nous pouvons citer principalement trois types d'hypercubes latins :

- Aléatoires,
- Optimisés, afin d'optimiser l'occupation spatiale,
- Orthogonaux, visant à minimiser la corrélation des estimateurs des effets principaux.

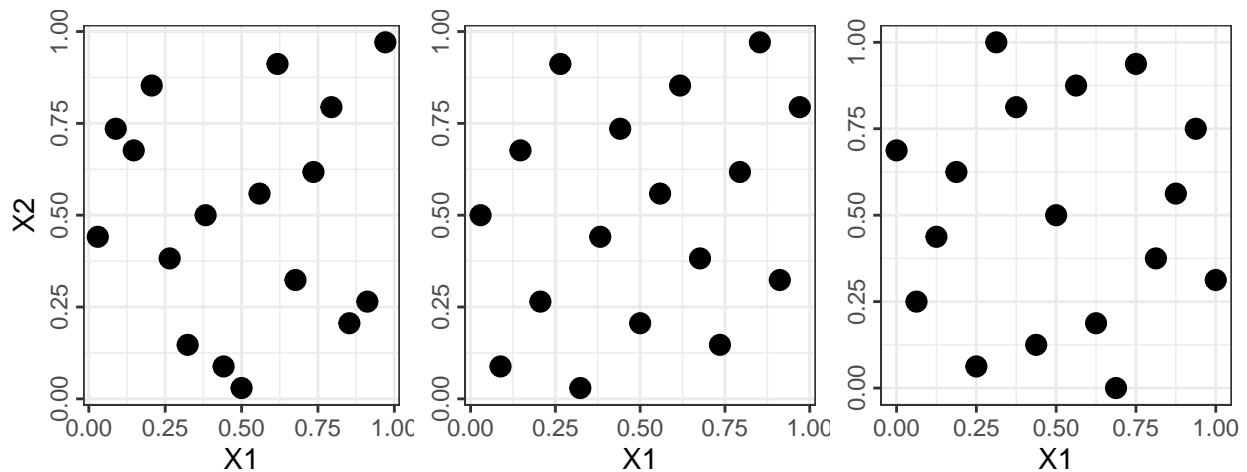


Figure 8: Carré latin aléatoire (à gauche), carré latin avec optimisation évolutive ESE maximin (au milieu), carré latin quasi-orthogonal (à droite)

4.4 Evaluation des performances des modèles

L'optimisation des modèles ainsi que la comparaison des performances des modèles entre eux implique nécessairement de définir quel sera le critère vis-à-vis duquel cette performance sera évaluée.

De nombreux critères sont utilisables, en fonction du cahier des charges, mais également du type de tâche effectuée : régression, classification binaire, classification multiclasse.

Dans une tâche de classification binaire, les critères usuels sont la spécificité, la sensibilité, et l'aire sous la courbe de fonction d'efficacité du récepteur (*AUC ROC*, parfois abrégé en *ROC*). Il conviendra bien évidemment, avant d'utiliser des indicateurs tels que la spécificité et la sensibilité, de définir la notion de test positif et test négatif.

D'autres indicateurs d'intérêt existent, nous retiendrons ici l'index *J* de Youden³⁶ pondéré, qui permet de d'ajuster les pondérations de la spécificité et la sensibilité au sein d'un index synthétique.³⁷ Cet indicateur présente un intérêt particulier lorsqu'il apparaît souhaitable de tenir compte de la différence d'impact entre un faux positif et un faux négatif, sans pour autant autoriser des sensibilités ou spécificités trop faibles.

En l'espèce, l'index *J* de Youden pondéré nous permet donc de construire un indice synthétique tenant compte du fait qu'il est plus grave de classer à tort comme comestible un champignon toxique que d'écarter à tort un champignon parfaitement comestible, sans pour autant autoriser le modèle à écarter un nombre inconsidéré de champignons comestibles.

L'index *J* de Youden pondéré est donné par :³⁷

$$J_w = 2 \cdot (w \cdot Sen + (1 - w) \cdot Spe) - 1 \quad \text{avec } w \in [0; 1]$$

Dans la classification binaire de cette étude, problème qui revient classiquement en mycologie à classer les espèces en fonction de leur toxicité, la valeur positive sera ici arbitrairement attribuée à la valeur "champignon toxique". Nous cherchons donc à maximiser la sensibilité de la détection, afin d'écarter les espèces toxiques, la spécificité – c'est-à-dire la capacité à ne pas écarter trop d'espèces comestibles – apparaissant alors comme un critère relativement secondaire. En établissant arbitrairement un index de Youden pondéré accordant dix ordres de grandeur entre l'importance accordée à la sensibilité et à la spécificité, nous pouvons établir $w = 10/11$.

Dans les tâches de classification multiclasse, d'autres indicateurs d'intérêt pourront être utilisés, tels que le kappa de Cohen, l'indice de Rand (*accuracy*), mais aussi la sensibilité et la spécificité moyennes.

Nous retiendrons dans le cadre de notre étude le kappa de Cohen, calculé à partir de la matrice de confusion, et donné par :

$$\kappa = \frac{\pi_0 - \pi_e}{1 - \pi_e}$$

Avec π_0 la probabilité d'accord entre notre modèle et la classe réelle du champignon, et π_e la probabilité d'un même accord résultant du pur hasard.

Landis et Koch ont défini des [INSERER SCHEMA MATRICE DE CONFUSION]

Source Kappa Cohen : <https://journals-sagepub-com.ressources-electroniques.univ-lille.fr/doi/abs/10.1177/001316446002000104>

Grille validité Kappa Cohen (Landis & Koch) : https://www-jstor-org.ressources-electroniques.univ-lille.fr/stable/2529310?searchText=The+Measurement+of+Observer+Agreement+for+Categorical+Data&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3DThe%2BMeasurement%2Bof%2BObserver%2BAGreement%2Bfor%2BCategorical%2BData&ab_segments=0%2Fbasic_search_gsv2%2Fcontrol&refreqid=fastly-default%3A60efe6a1ae467d1002cd4305a15de703

5 Apprentissage machine et classification binaire

Brouillon, le lot de données utilisé ici est le *Secondary Mushroom Dataset* de D.Wagner.

5.1 Analyse exploratoire des données (EDA)

La structure du lot de données est la suivante :

Le lot de données d'origine a été découpé deux fois avec un rapport 92/8, comme décrit dans la partie. (cf. § et p(ref!!!))

Toutes les distributions des variables du lot d'entraînement ont ensuite été tracées par histogrammes pour les variables numériques, et diagrammes en barres pour les variables alphabétiques et catégorielles.

Les diagrammes en barres n'ont rien illustré de particulièrement remarquable et n'ont pas été inclus dans le rapport. Toutefois, les distributions dimensionnelles sont plus intéressantes : à première vue, elles semblent suivre une courbe en cloche (fig. REF), avec une longue queue à droite. Une transformation logarithmique (fig. REF) montre plus nettement la forme de cette queue.

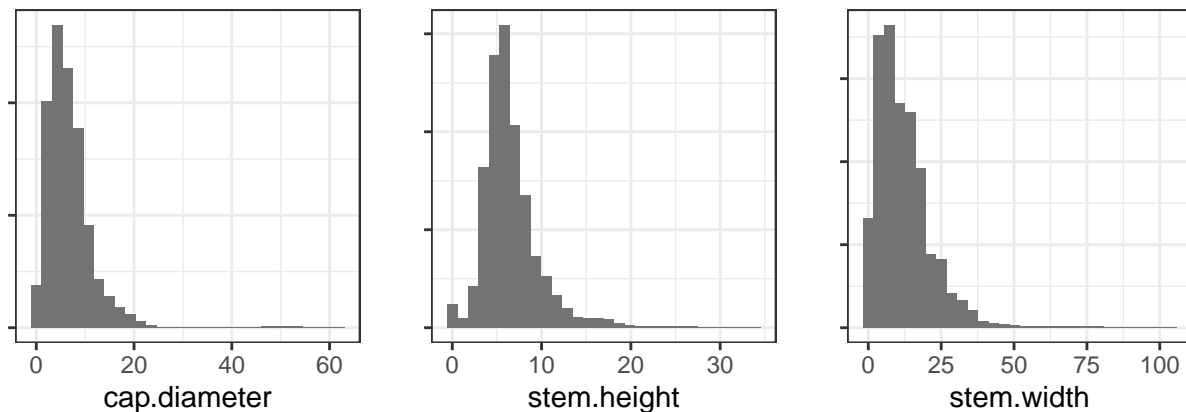


Figure 9: Distribution des diamètres de chapeau, longueur de stipe, diamètre de stipe

La distribution du diamètre du chapeau D_C a l'apparence d'une courbe en cloche, avec une longue queue à droite, mais est en réalité bimodale, avec un mode principal à 5cm, et un mode secondaire beaucoup plus petit pour $D_C \approx 50 \text{ cm}$. Cette taille exceptionnelle est attribuable à des variétés telles que *Polyporus squamosus*.²⁵

La distribution de la longueur de stipe L_S a également une forme de courbe en cloche avec une longue queue à droite, un mode principal à 5 cm et un mode secondaire à 0 cm. Cette valeur peut également sembler surprenante, mais certains champignons n'ont pas de stipe, ce qui explique cette valeur.

Table 1: Dataset structure

	Type	Niveaux
class	factor	2
cap.diameter	numeric	2571
cap.shape	factor	7
cap.surface	factor	12
cap.color	factor	12
does.bruise.or.bleed	factor	2
gill.attachment	factor	8
gill.spacing	factor	4
gill.color	factor	12
stem.height	numeric	2226
stem.width	numeric	4630
stem.root	factor	6
stem.surface	factor	9
stem.color	factor	13
veil.type	factor	2
veil.color	factor	7
has.ring	factor	2
ring.type	factor	9
spore.print.color	factor	8
habitat	factor	8
season	factor	4

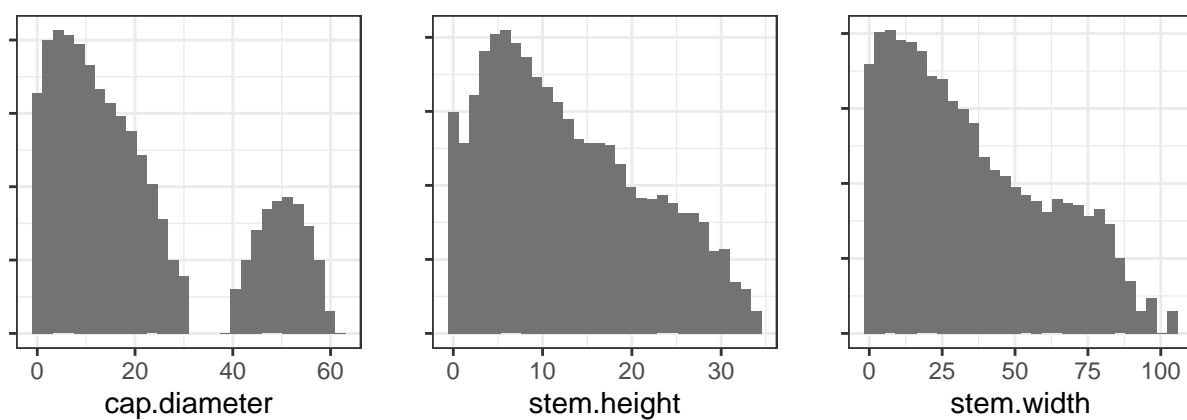


Figure 10: Distribution des diamètres de chapeau, longueur de stipe, diamètre de stipe (échelle log en Y)

La distribution du diamètre de stipe D_S a aussi l'apparence d'une courbe en cloche avec une longueur queue à droite, et un pic à $D_S \approx 10 - 15\text{mm}$. Dans toutes ces distributions, la longue queue à droite peut probablement s'expliquer par l'utilisation, dans le *Secondary Mushroom Dataset*, d'une distribution normale pour chaque variété, associée à l'impossibilité d'avoir des valeurs dimensionnelles négatives.

5.2 Modèles de la librairie *caret* : optimisation et sélection

La librairie *caret* fournit une plateforme très pratique et efficiente pour la modélisation de données et l'inférence par apprentissage machine. Cette section expliquera la stratégie utilisée pour l'évaluation de certains de ces modèles. Les modèles sélectionnés sont de types variés :

- Analyse discriminante linéaire (LDA) : *Linear Discriminant Analysis* (lda2), *Penalized Discriminant Analysis* (pda)
- Modèle additif généralisé (GAM) : *Generalized Additive Model using LOESS* (gamLoess)
- Modèle arborescent : *Classification And Regression Tree* (CART) (rpart, rpartCost), *Single C5.0 Tree* (ctree)
- Forêt aléatoire : *Random Ferns* (rferns), *Random Forest* (ranger, Rborist)

La première étape est de construire une fonction d'évaluation et de régression, afin de rationaliser notre travail. En effet, l'évaluation et la régression seront effectuées très fréquemment au cours de cette étude, il est donc souhaitable d'assigner une fonction dédiée à ces tâches. La librairie *caret* permet de la définir très simplement par :

```
set.seed(1)
tr_ctrl <- trainControl(classProbs = TRUE,
                        summaryFunction = twoClassSummary,
                        method = "cv", number = 10)
train(class ~ .,
      method = [METHODE],
      data = BI_lot_appr_opti,
      trControl = tr_ctrl,
      metric = 'Sens',
      tuneGrid = data.frame([PARAMETRES]))
```

La fonction *set.seed* assure la reproductibilité.

La fonction *trainControl* permet de contrôler divers paramètres d'entraînement et de validation à fins d'optimisation ; dans cet exemple, d'utiliser un critère tel que le ROC, la sensibilité ou la spécificité, et d'utiliser une validation croisée (*cross validation*) à 10 blocs (*10-folds*).

La fonction *train* lance le processus d'apprentissage et d'évaluation, tout en permettant l'utilisation de différents paramètres tels que la méthode de sélection des jeux d'entraînement et de validation, la métrique utilisée pour l'évaluation (sensibilité, spécificité, ROC...) ainsi que la grille des paramètres utilisés par le modèle.

Ce bloc de code a été inclus dans une fonction pour faciliter l'accessibilité du code et la reproductibilité. Cette fonction renvoie une liste qui peut ensuite être utilisée pour générer un graphique, ou pour extraire différentes données d'intérêt, contenues dans *.\$results* (pour une classification binaire : AUC de ROC, sensibilité, spécificité), *.\$bestTune* (paramètres donnant la

meilleure performance sur la métrique utilisée) et `.$finalModel` (informations diverses pouvant parfois être utilisées dans des graphiques, tels que des arbres décisionnels).

5.2.1 Modèles d'analyse discriminante linéaire

Les deux modèles d'analyse discriminante linéaire (LDA) choisis pour cette étude sont `lda2` (*Linear Discriminant Analysis*) et `pda` (*Penalized Discriminant Analysis*). Le modèle `lda2` dispose d'un hyperparamètre (*dimen*, nombre de fonctions discriminantes). Le modèle `pda` a également un unique hyperparamètre (*lambda*, pénalité de réduction des coefficients).

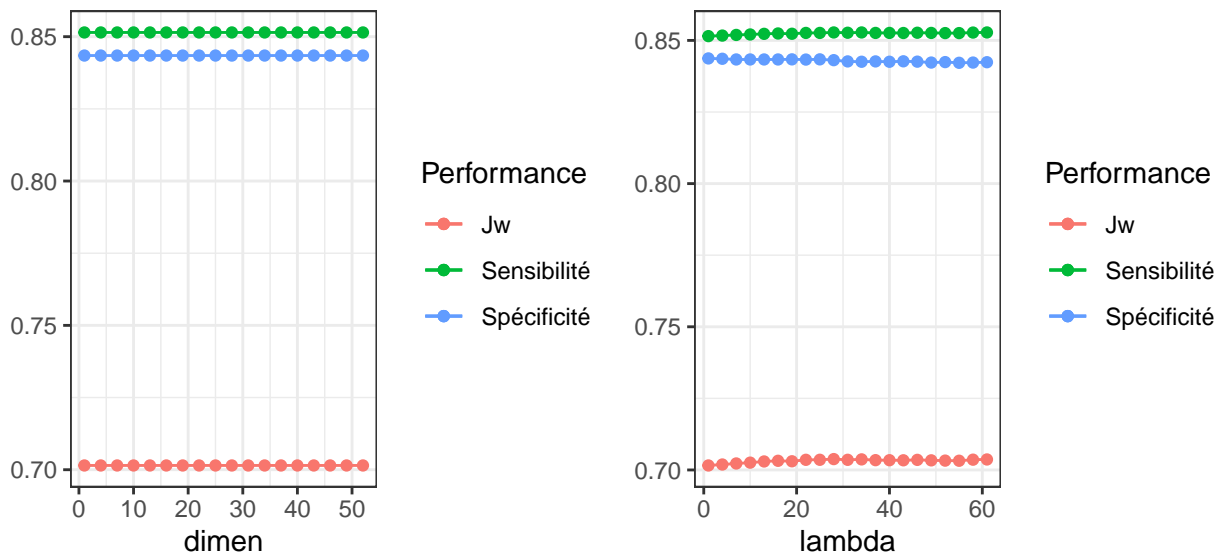


Figure 11: Performances des modèles `lda2` (à gauche) et `pda` (à droite)

Le paramètre *dimen* du modèle `lda2` ne semble pas avoir d'effet significatif sur la sensibilité ($Sens = 0.851$).

Le paramètre *lambda* du modèle `pda` impacte sa sensibilité de façon marginale, avec des *lambdas* faibles donnant une légère amélioration des résultats ($Sens_{max} = 0.853$).

Toutefois, la sensibilité de ces deux modèles reste malheureusement très insuffisante pour notre étude ($Sens \approx 0.85$), et leurs spécificités n'ont rien de remarquable non plus ($Spec \approx 0.843$).

Ces performances médiocres s'expliquent par le fonctionnement même des modèles LDA qui, s'ils peuvent analyser des données qualitatives à fins de classifications, ne peuvent le faire que si une quantification sous-jacente est possible, par exemple :

- Données binaires ou booléennes
- Données catégorielles basées sur des données numériques

5.2.2 Modèle additif généralisé

Le seul modèle additif généralisé choisi pour cette étude est gamLoess (*Generalized Additive Model using Locally Weighted Linear Regression*). La documentation de la librairie indique que le modèle gamLoess dispose de deux hyperparamètres : *span* (fraction de points utilisés dans l'environnement local) and *degree* (degré de linéarisation).

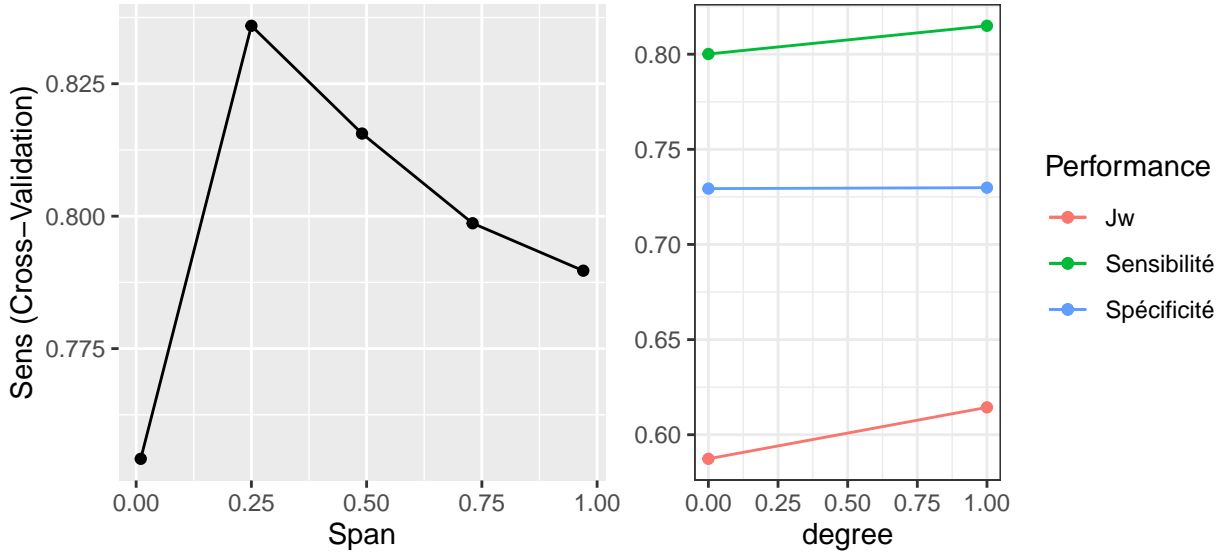


Figure 12: Performances de gamLoess en fonction du span (gauche) and du degré (droite)

L'hyperparamètre *degree* du modèle gamLoess ne semble pas avoir d'effet significatif sur la sensibilité, avec $\Delta Sens = 0.015$.

L'hyperparamètre *span* affecte marginalement la sensibilité de gamLoess, avec une valeur optimale de $span = 0.25$, aboutissant à une sensibilité maximale de $Sens_{max} = 0.836$.

La sensibilité de ce modèle n'atteint pas le critère posé pour notre étude. Le modèle additif généralisé s'est avéré inférieur aux modèles d'analyse discriminante linéaire, aussi bien en sensibilité (0.836 vs 0.852) qu'en spécificité (0.749 vs 0.843).

5.2.3 Modèles d'arbres de décision

Les modèles basés sur des arbres de décision ont un intérêt tout particulier pour cette étude, pour ceux raisons majeures :

- La logique en arbre de décision est habituellement usitée pour la classification manuelle des champignons,
- Les arbres de décision obtenus peuvent être tracés, et facilement interprétés par l'humain.

Table 2: Performance du modèle CART (rpart)

cp	ROC	Sens	Spec
0.00001	0.99901	0.99740	0.99724
0.00003	0.99899	0.99740	0.99724
0.00010	0.99897	0.99753	0.99696
0.00032	0.99866	0.99618	0.99632
0.00100	0.99649	0.99050	0.98964
0.00316	0.98421	0.96211	0.95438
0.01000	0.90019	0.92451	0.78669
0.03162	0.71483	0.60329	0.79837
0.10000	0.61919	0.54204	0.69633

Les premiers modèles présentés dans le cadre de notre étude sont deux modèles CART (*Classification And Regression Tree*). Le modèle CART le plus simple proposé dans notre étude (rpart) ne dispose que d'un seul hyperparamètre : cp (complexité).

Le modèle CART le plus simple n'atteint jamais la sensibilité requise $Spec = 1$. Toutefois, ce modèle s'en approche, et donne de très bons résultats globaux, avec ($Sens_{max} = 0.998$ et $Spec_{max} = 0.997$).

Le second modèle CART utilisé dans cette étude (rpartCost) associe des hyperparamètres de complexité (cp) et de coût ($Cost$). Les graphiques de sensibilité et de spécificité en fonction des hyperparamètres illustrent bien, dans leur partie supérieure ($cp \geq 0,05$) la notion classique de compromis entre sensibilité et spécificité : dans cette zone, toute amélioration de la sensibilité se fera inévitablement au détriment de la spécificité, et inversement.

En pratique, pour $cp \geq 0,05$, notre modèle d'IA basé sur ce type d'arbre de décision se montrera soit excessivement sévère, rejetant un nombre considérable de champignons comestibles (quadrant supérieur gauche, $cost \leq 1,5$), soit au contraire excessivement laxiste, admettant un nombre important de champignons non-comestibles (quadrant supérieur droit, $cost \geq 1,5$).

C'est dans la section inférieure de ces graphiques ($cp \leq 0,025$) que le modèle montre une performance acceptable tant en sensibilité qu'en spécificité.

Les performances maximales seront ici atteintes pour $cp = 0.000635$ et $Cost = 0.626$.

Table 3: Performance du modèle CART (rpartCost)

Cost	cp	Sens	Spec
0.626	0.00064	0.98678	0.9964

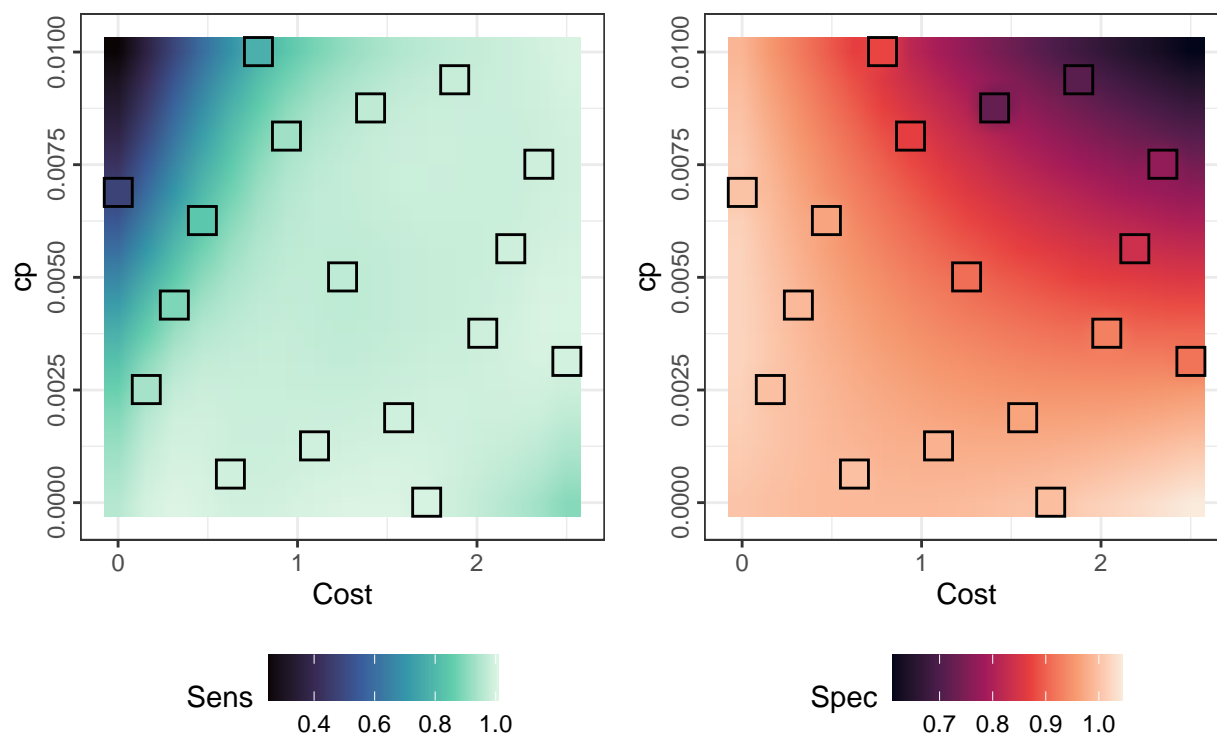


Figure 13: Sensibilité (à gauche) et spécificité (à droite) de `rpartCost` en fonction de la complexité et du coût (interpolation quadratique, points expérimentaux encadrés en noir)

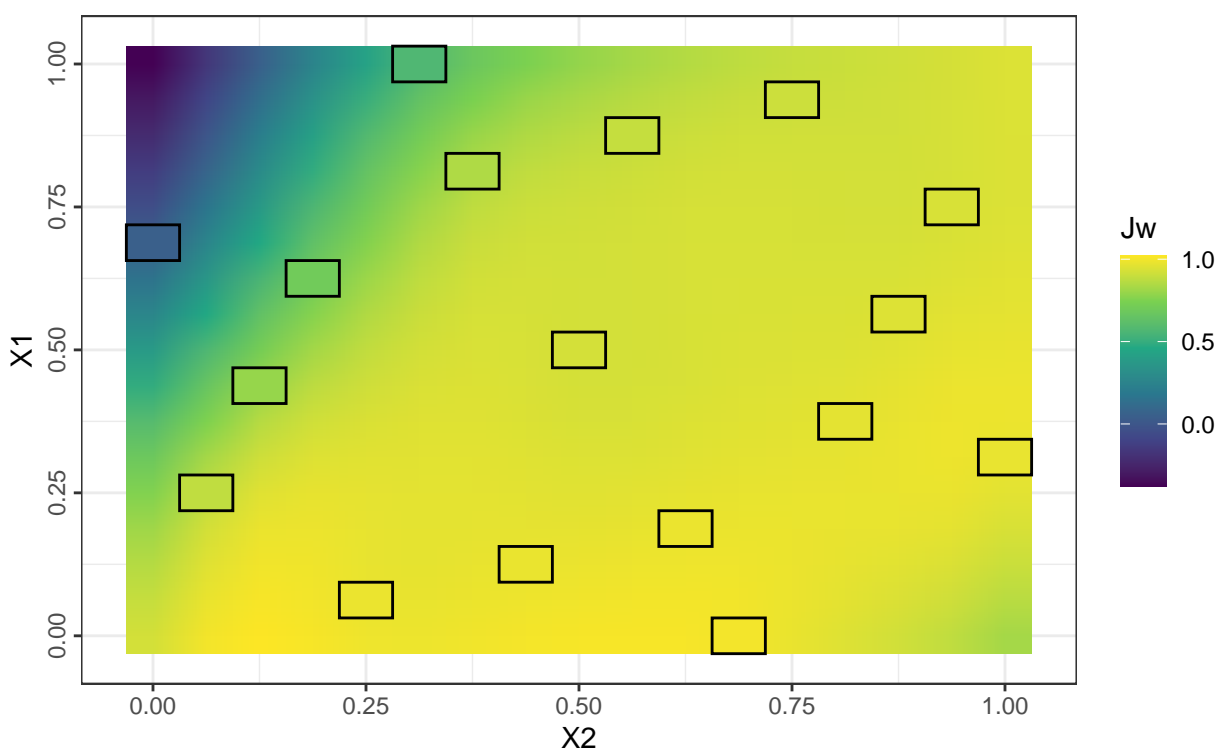


Figure 14: Performances (index de Youden pondéré 10:1) de `rpartCost` en fonction des paramètres réduits de complexité X_1 et de coût X_2 (interpolation quadratique, points expérimentaux encadrés en noir)

Les performances du modèle `rpartCost`, quoi qu'excellentes, ne permettent pas d'atteindre la sensibilité requise.

Le dernier modèle d'arbre décisionnel est `C5.0 tree` (`c50tree`). Ce modèle ne dispose d'aucun hyperparamètre.

De manière assez surprenante, bien que ne disposant d'aucun hyperparamètre, ce modèle a donné d'excellents résultats sans optimisation nécessaire, avec de très hautes sensibilité et spécificité. Toutefois, le modèle `C5.0 tree` n'a pas rempli l'objectif posé par le critère $Sens = 1$.

5.2.4 Forêts aléatoires

Table 4: Performance du modèle `C5.0 tree`

ROC	Sens	Spec
0.99943	0.99878	0.99872

Le premier modèle de forêt aléatoire évalué dans cette étude est le modèle de fougères aléatoires rFerns (*Random Ferns*). Ce modèle ne possède qu'un seul hyperparamètre, la profondeur (*depth*).

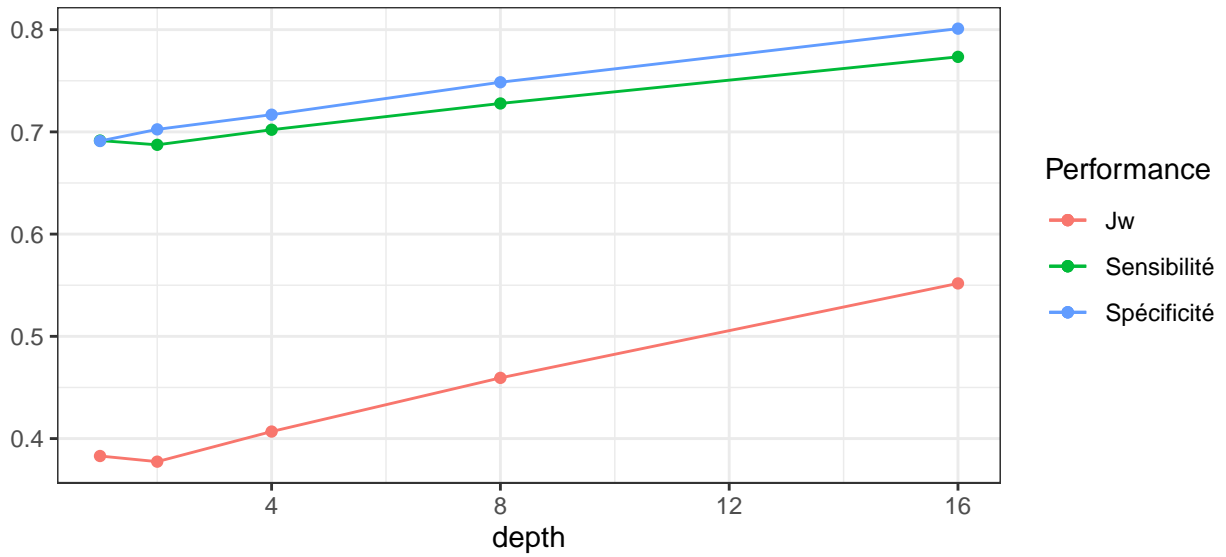


Figure 15: Performances du modèle de fougères aléatoires

Quoique très efficient sur le plan calculatoire, le modèle de fougères aléatoires a fourni des résultats assez peu satisfaisants, avec une sensibilité maximale de $Sens_{max} = 0.773$. La spécificité maximale n'est pas très élevée non plus ($Spec_{max} = 0.801$).

Le second modèle de forêt aléatoire évalué dans cette étude est Rborist. Deux hyperparamètres régissent ce modèle : le nombre de prédicteurs testés pour une scission (*predFixed*) et le nombre minimal de lignes-références distinctes avant de scinder un nœud (*minNode*).

En posant comme facteurs réduits :

- $X_1 \in [0; 1]$ pour le paramètre *minNode*,
- $X_2 \in [0; 1]$ pour le paramètre *predFixed*,

Nous pouvons modéliser la réponse avec un modèle quadratique

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_{12} \cdot X_1 \cdot X_2 + \beta_{11} \cdot X_1^2 + \beta_{22} \cdot X_2^2$$

Ici encore, les étapes d'optimisation ont montré des résultats prometteurs : sur ce modèle également, la spécificité requise a été atteinte sur plusieurs occurrences, en utilisant seulement une optimisation monoparamétrique.

Avec des paramètres optimaux (*predFixed* = 9 et *minNode* = 2), la performance est estimée à :

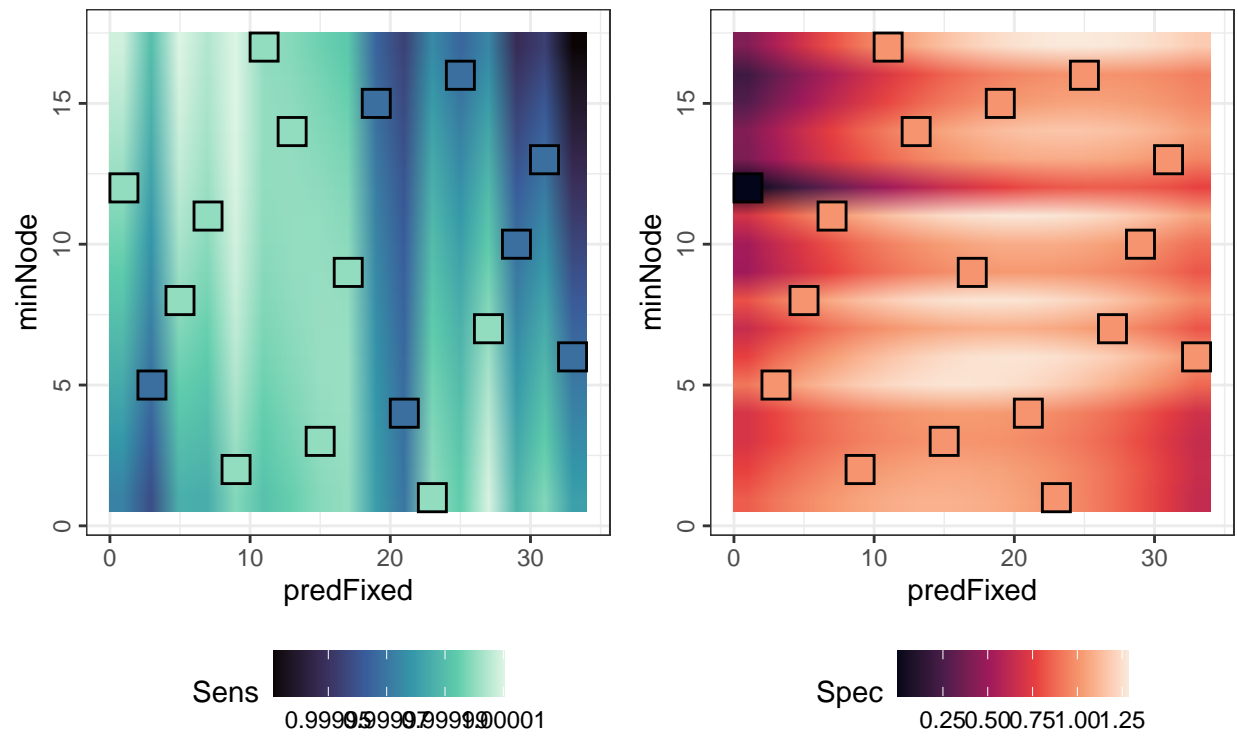


Figure 16: Sensibilité (à gauche) et spécificité (à droite) du modèle Rborist en fonction de ses deux hyperparamètres (interpolation quadratique, points expérimentaux encadrés en noir)

Table 5: Performances du modèle Rborist (hyperparamètres optimaux)

predFixed	minNode	ROC	Sens	Spec
9	2	1	1	0.99996

Ce modèle s'est comporté de façon excellente, donnant une sensibilité et une spécificité remarquables dans cette phrase d'évaluation.

Le dernier modèle de forêt aléatoire que nous évaluons dans cette étude est le modèle ranger. La documentation de la librairie caret mentionne trois hyperparamètres : la taille minimale de nœud (*min.node.size*), le nombre de caractéristiques à séparer à chaque nœud (*mtry*) et la règle contrôlant cette séparation (*splitrule*).

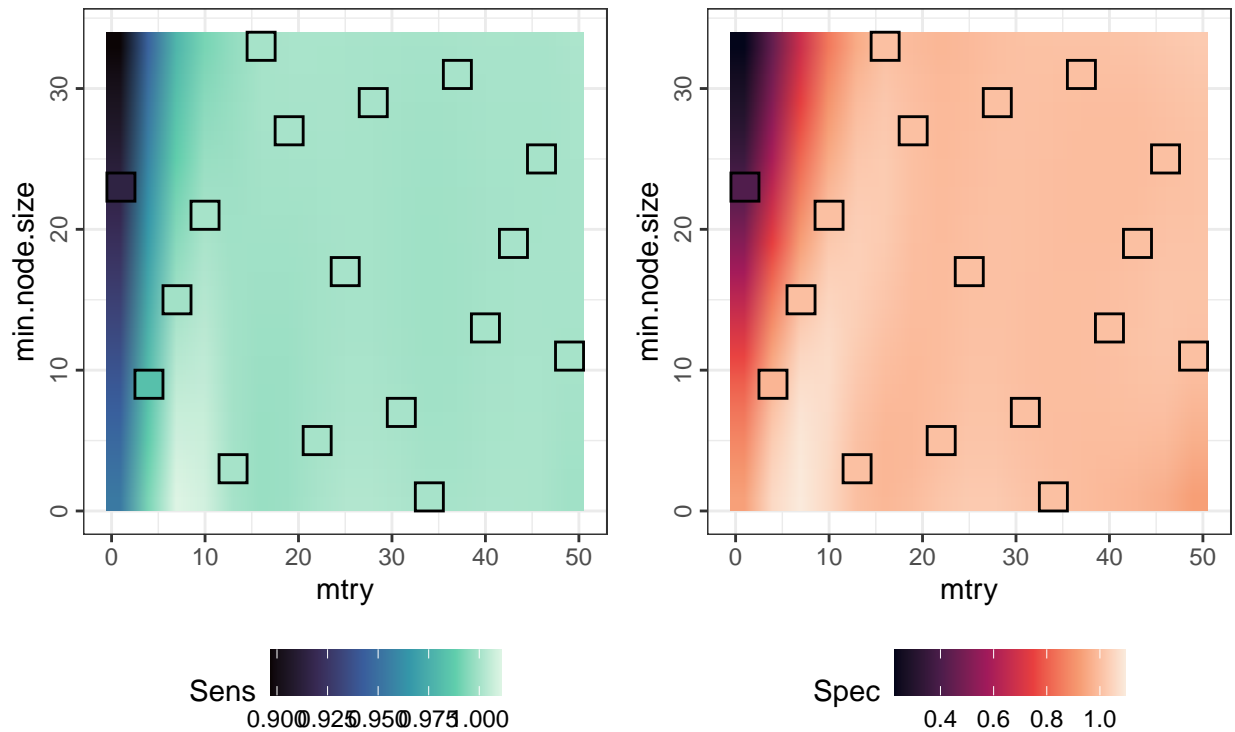


Figure 17: Sensibilité (à gauche) et spécificité (à droite) du modèle Ranger, en fonction des 2 hyperparamètres (algorithme de scission : gini)

Les étapes préliminaires de l'optimisation du modèle se sont avérées prometteuses : même avec un très faible nombre d'arbres ($n = 6$), la spécificité requise était déjà atteinte à plusieurs reprises lors de l'optimisation uniparamètre. L'optimisation de la totalité des hyperparamètres ($min.node.size = 27$, $mtry = 19$ et $splitrule = extratrees$) a donné d'excellents résultats.

Le modèle Ranger a donné des résultats similaires à ceux du modèle Rborist, avec une sensibilité et une spécificité excellentes.

Table 6: Performances du modèle Ranger (hyperparamètres optimaux)

mtry	min.node.size	splitrule	ROC	Sens	Spec
19	27	extratrees	1	0.99994	0.99996

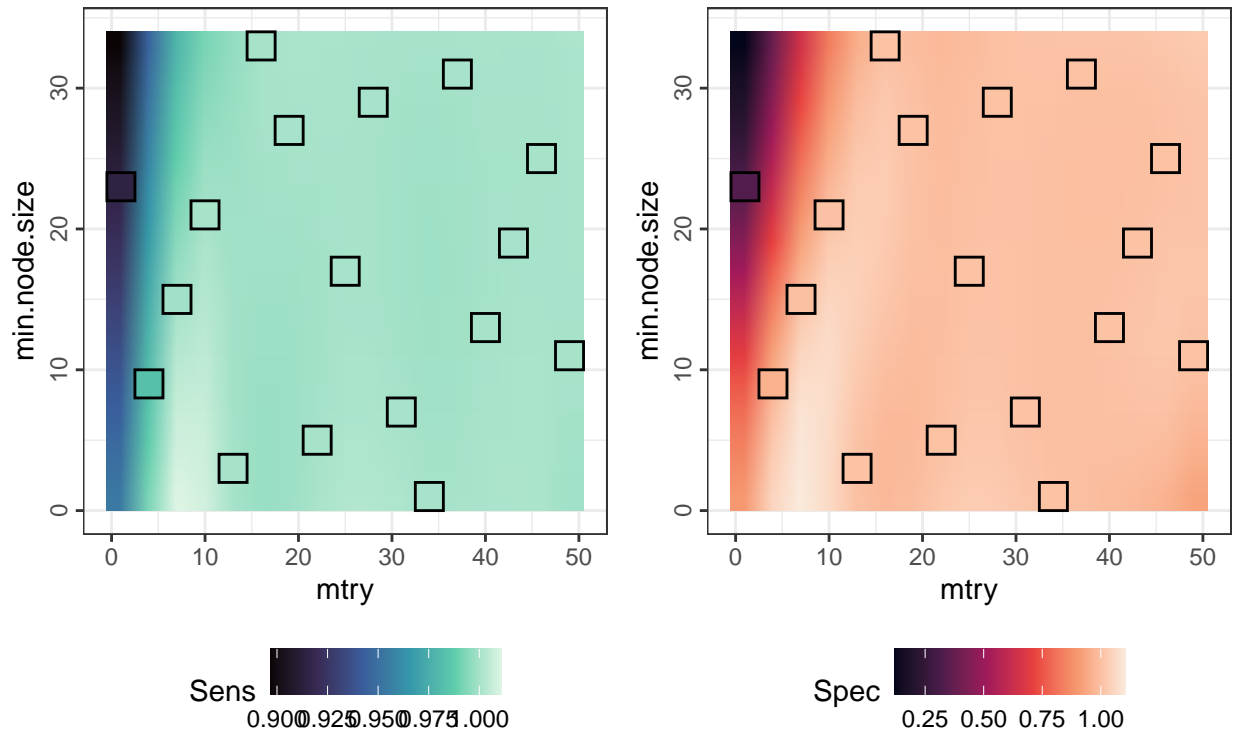


Figure 18: Sensibilité (à gauche) et spécificité (à droite) du modèle Ranger, en fonction des 2 hyperparamètres (algorithme de scission : extratrees)

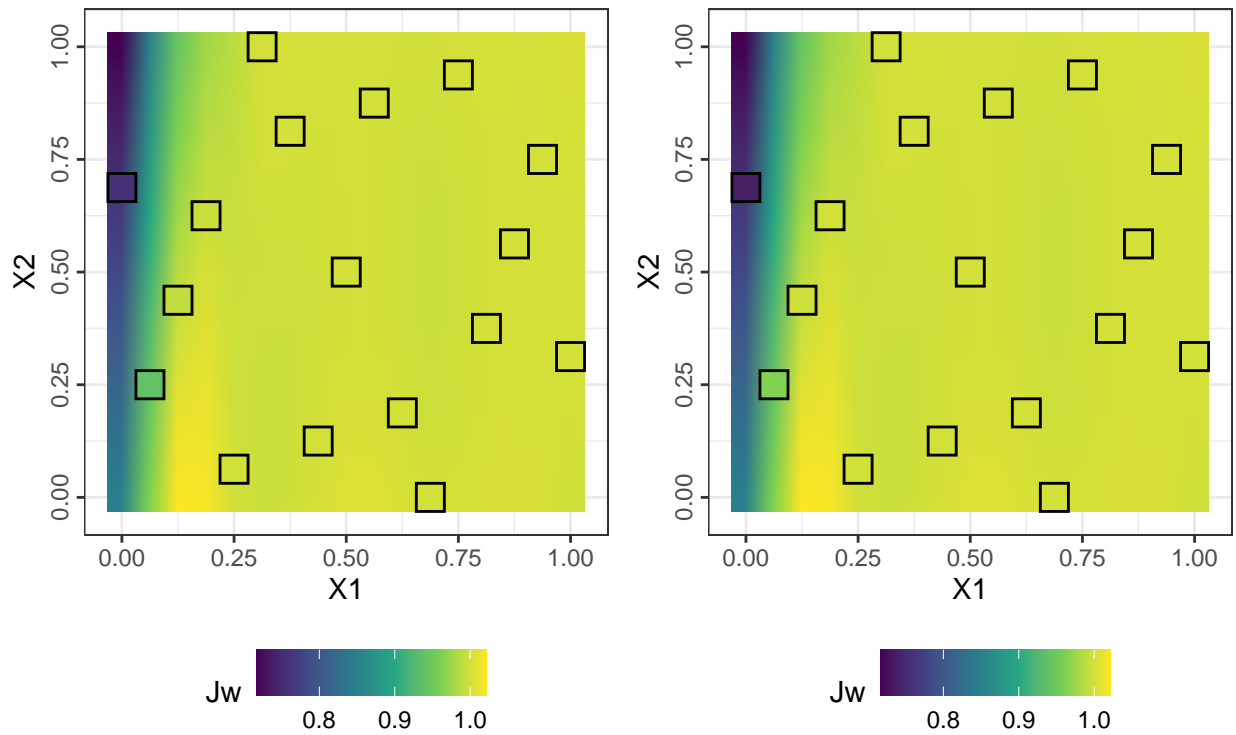


Figure 19: Performances du modèle Ranger en fonction de l'algorithme de scission (extratrees à gauche, gini à droite) et des paramètres réduits : caractéristiques à séparer $X1$ et taille minimale de noeud $X2$ (interpolation quadratique, points expérimentaux encadrés en noir)

Ces résultats soulignent un fait intéressant : tous les modèles de forêts aléatoires ne sont pas égaux. Notre étude montre une différence considérable en sensibilité et spécificité entre les forêts aléatoires de type rForns, Ranger et Rborist. Lors des étapes préliminaires de cette étude, d'autres modèles de forêts aléatoires disponibles dans la librairie caret se sont également montrés extrêmement inefficients sur le plan calculatoire, alors que d'autres se sont avérés sensiblement plus rapides et ont pu être retenus pour notre étude.

La vitesse des algorithmes sera l'objet de la section suivante. La durée d'exécution de n'importe quelle portion de code R pouvant facilement être mesurée par :

```
temps_debut <- Sys.time()
[Code à évaluer]
temps_fin <- Sys.time()
duree <- difftime(temps_fin, temps_debut)
```

5.3 Résultats

5.3.1 Protocole d'évaluation

Les modèles ayant atteint le critère de spécificité ($Spec = 1$) lors de l'étape d'optimisation ont été choisis pour l'évaluation. Les deux modèles choisis sont deux forêts aléatoires :

- Forêt aléatoire de type Ranger,
- Forêt aléatoire de type Rborist.

Tous les modèles ont été entraînés sur le jeu de données d'apprentissage, réglés avec les meilleurs hyperparamètres obtenus par mesure des performances face au jeu de données d'optimisation. Leurs performances face au jeu de données d'évaluation seront analysées avec les mêmes critères que précédemment :

- La sensibilité *doit* être supérieure à 0,9999.
- La spécificité doit être la plus haute possible.

5.3.2 Performances des modèles de forêts aléatoires

L'évaluation finale du modèle ranger donne la matrice de confusion suivante :

Table 7: Matrice de confusion du modèle Ranger

	toxique	comestible
toxique	2503	1
comestible	0	1992

Table 8: Performances des modèles Ranger et Rborist (évaluation)

	Sensibilité	Spécificité	F1 score	Durée (min)
Ranger	1	0.9995	0.9998	23.35
Rborist	1	1.0000	1.0000	3.38

La précision finale est égale à 0.9998, avec un intervalle de confiance à 95% de [0.9988 ; 1]. La forêt aléatoire de type Ranger a donné d'excellents résultats, en un temps très contenu (23.35 min), preuve de sa grande efficacité calculatoire.

La forêt aléatoire de type Rborist a donné des résultats similaires, avec une précision finale égale à 1, avec un intervalle de confiance à 95% de [0.9992 ; 1]. Le modèle Rborist, donnant des résultats sensiblement identiques à Ranger, s'est avéré extrêmement efficace sur le plan calculatoire (3.38 min).

6 Apprentissage machine et classification multiclasse

Brouillon, le lot de données utilisé ici est un lot synthétique créé par moi-même, mais à partir des données primaires du Secondary Mushroom Dataset de D.Wagner.

Etant données des performances qu'ont montré les différents modèles lors de la classification binaire, seuls les modèles basés sur les arbres décisionnels et les forêts aléatoires seront évalués dans cette section.

6.1 Classification par familles

6.1.1 Modèles d'arbres de décision

Les premiers modèles présentés dans le cadre de notre étude sont deux modèles CART (*Classification And Regression Tree*). Le modèle CART basique (rpart) n'a qu'un hyperparamètre : cp (complexité).

Le modèle CART le plus simple n'atteint jamais la sensibilité requise $Spec = 1$. Toutefois, ce modèle s'en approche, et donne de très bons résultats globaux, avec ($Sens_{max} = 0.999$ et $Spec_{max} = 1$).

Le second modèle CART utilisé dans cette étude (rpartCost) associe des hyperparamètres de complexité (cp) et de coût ($Cost$).

Les performances maximales sont atteintes pour $cp = 0.000635$ et $Cost = 0.626$.

Les performances du modèle rpartCost, quoi qu'excellentes, ne permettent pas d'atteindre la spécificité requise.

Table 9: Performance du modèle CART (rpart)

cp	ROC	Sens	Spec
0.00001	0.99976	0.99857	0.99844
0.00003	0.99976	0.99845	0.99844
0.00010	0.99974	0.99802	0.99846
0.00032	0.99942	0.99656	0.99757
0.00100	0.99788	0.99128	0.99395
0.00316	0.99311	0.97767	0.98033
0.01000	0.94754	0.91029	0.90549
0.03162	0.77033	0.93060	0.56222
0.10000	0.50000	0.00000	1.00000

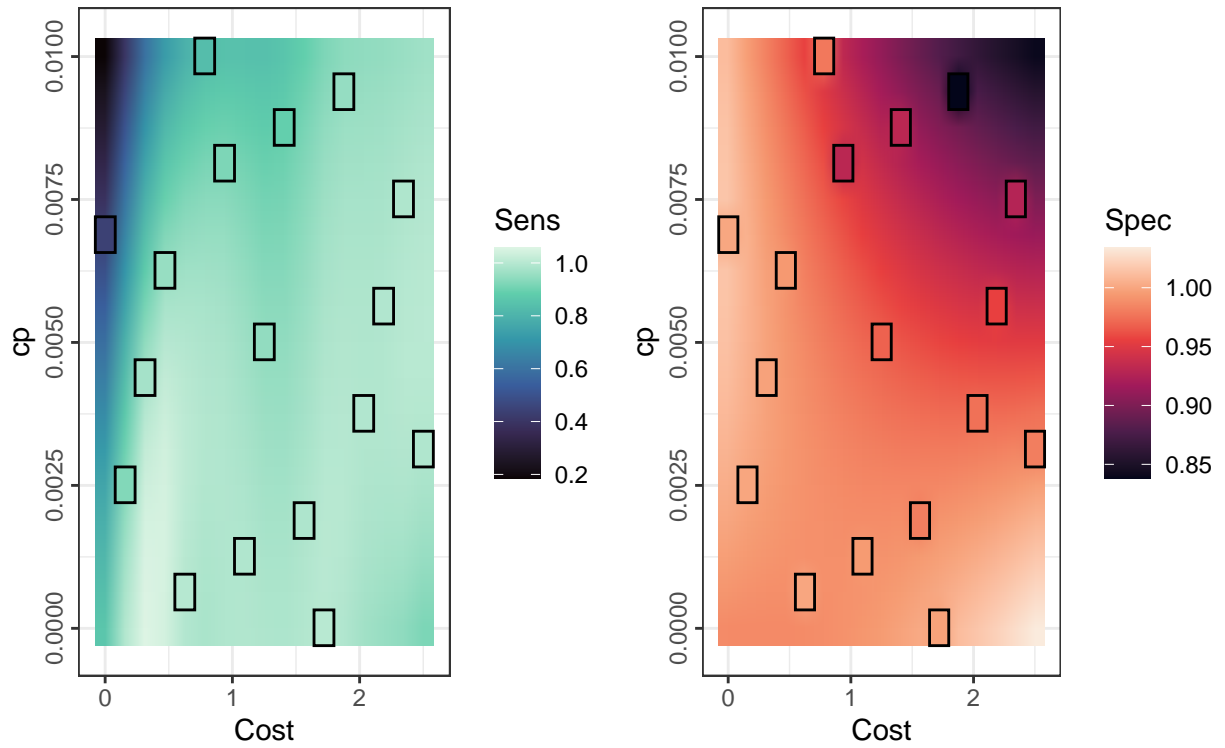


Figure 20: Sensibilité (à gauche) et spécificité (à droite) de `rpartCost` en fonction de la complexité et du coût (interpolation quadratique, points expérimentaux encadrés en noir)

Table 10: Performance du modèle CART (`rpartCost`)

Cost	cp	Sens	Spec
0.626	0.00064	0.99647	0.99965

Table 11: Performances du modèle Ranger (hyperparamètres optimaux)

mtry	min.node.size	splitrule	ROC	Sens	Spec
22	5	extratrees	1	1	1

6.1.2 Forêts aléatoires

Le premier modèle de forêt aléatoire évalué dans cette partie est le modèle ranger, qui possède trois hyperparamètres : la taille minimale de nœud (*min.node.size*), le nombre de caractéristiques à séparer à chaque nœud (*mtry*) et la règle contrôlant cette séparation (*splitrule*).

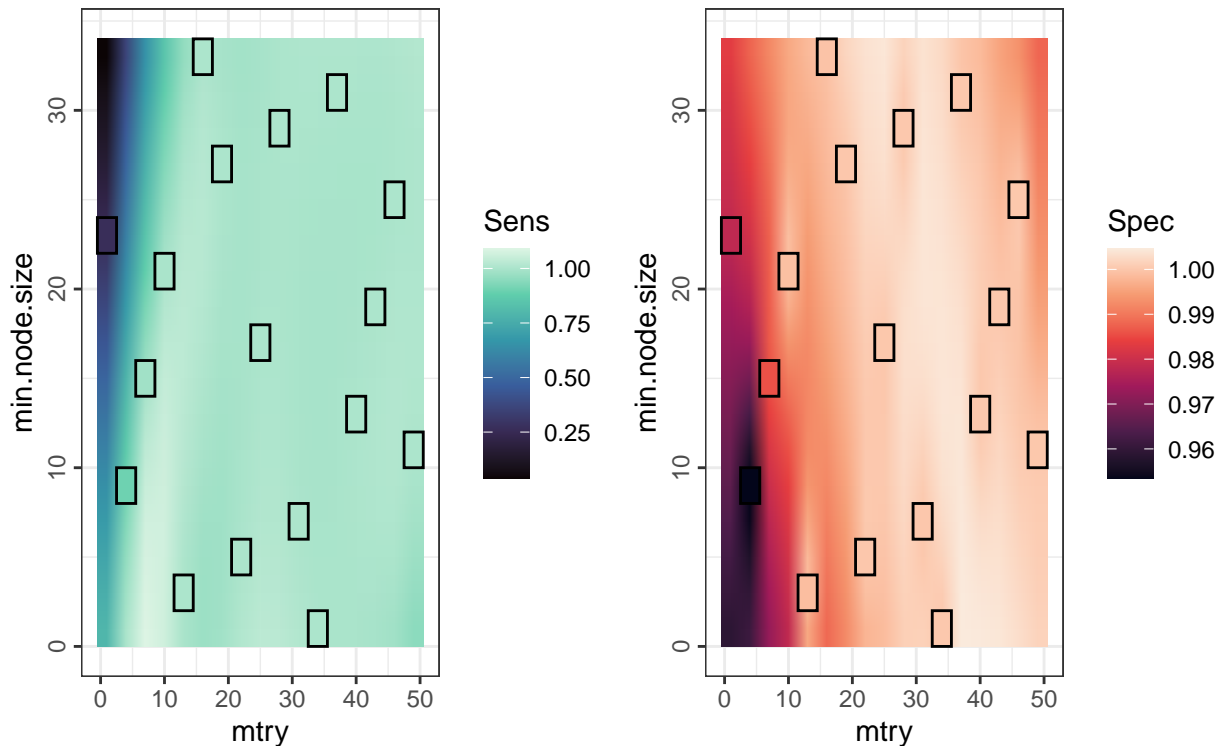


Figure 21: Sensibilité (à gauche) et spécificité (à droite) du modèle Ranger, en fonction des 2 hyperparamètres (algorithme de scission : gini)

Les étapes préliminaires de l'optimisation du modèle se sont avérées prometteuses : même avec un très faible nombre d'arbres ($n = 6$), la spécificité requise était déjà atteinte à plusieurs reprises lors de l'optimisation uniparamètre. L'optimisation de la totalité des hyperparamètres (*min.node.size* = 5, *mtry* = 22 et *splitrule* = extratrees) a donné d'excellents résultats.

Ce modèle s'est comporté de façon excellente, donnant une spécificité et une sensibilité remarquables dans cette phrase d'évaluation.

Le dernier modèle de forêt aléatoire est Rborist. Deux hyperparamètres régissent ce modèle : le nombre de prédicteurs testés pour une scission (*predFixed*) et le nombre minimal de lignes-références distinctes avant de scinder un nœud (*minNode*).

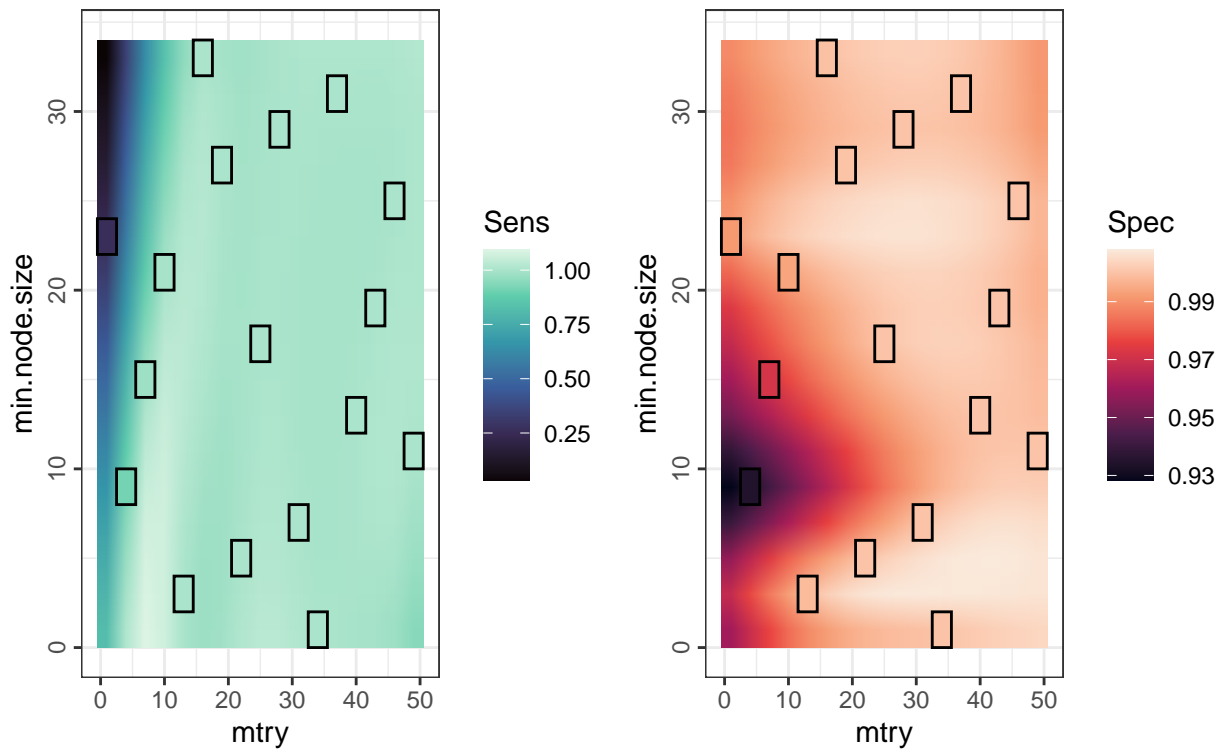


Figure 22: Sensibilité (à gauche) et spécificité (à droite) du modèle Ranger, en fonction des 2 hyperparamètres (algorithme de scission : extratrees)

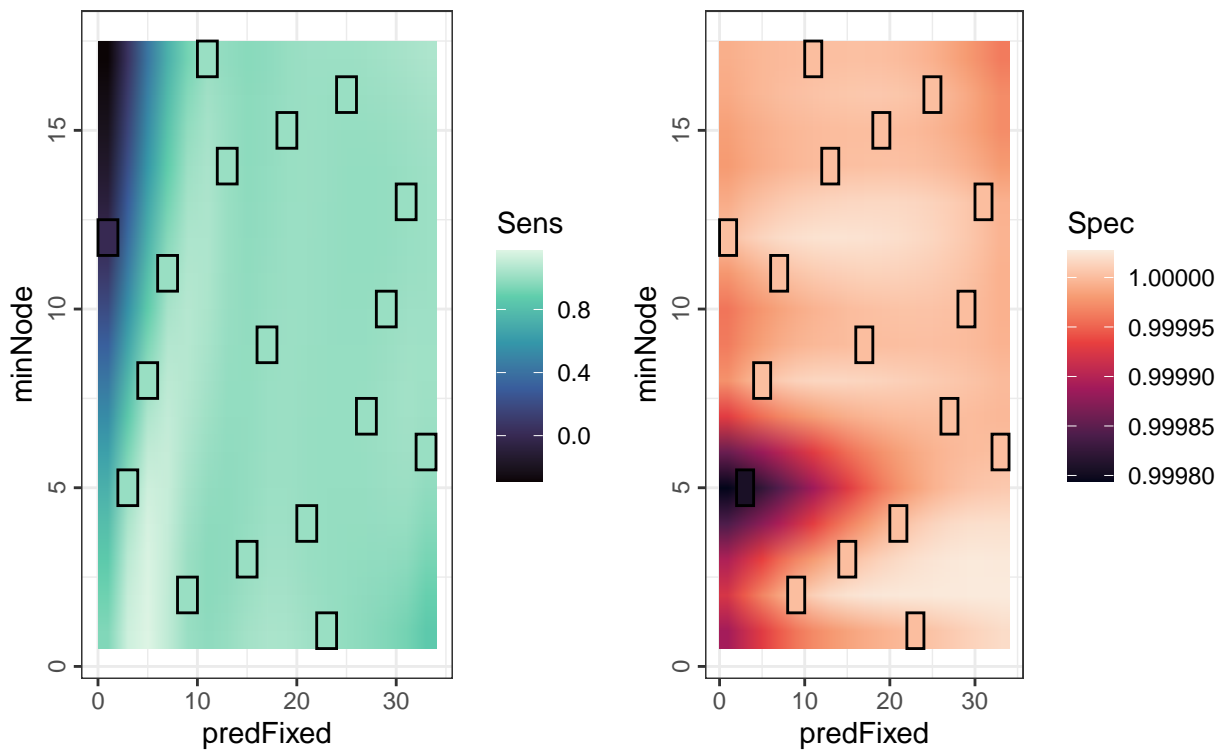


Figure 23: Sensibilité (à gauche) et spécificité (à droite) du modèle Rborist en fonction de ses deux hyperparamètres (interpolation quadratique, points expérimentaux encadrés en noir)

Ici encore, les étapes d'optimisation ont montré des résultats prometteurs : sur ce modèle également, la spécificité requise a été atteinte sur plusieurs occurrences, en utilisant seulement une optimisation monoparamétrique.

Avec des paramètres optimaux ($predFixed = 5$ et $minNode = 8$), la performance est estimée à :

Table 12: Performances du modèle Rborist (hyperparamètres optimaux)

predFixed	minNode	ROC	Sens	Spec
5	8	1	1	1

Le modèle Rborist a donné des résultats similaires à ceux du modèle Ranger, avec une sensibilité et une spécificité excellentes.

Ces résultats soulignent un fait intéressant : tous les modèles de forêts aléatoires ne sont pas égaux. Notre étude montre une différence considérable en sensibilité et spécificité entre les forêts aléatoires de type rFems, Ranger et Rborist. Lors des étapes préliminaires de cette étude, d'autres modèles de forêts aléatoires disponibles dans la librairie caret se sont également montrés extrêmement inefficients sur le plan calculatoire, alors que d'autres se sont avérés sensiblement plus rapides.

La vitesse des algorithmes sera l'objet de la section suivante. La durée d'exécution de n'importe quelle portion de code R peut facilement être mesurée par :

```
temps_debut <- Sys.time()
[Code à évaluer]
temps_fin <- Sys.time()
duree <- difftime(temps_fin, temps_debut)
```

6.1.3 Résultats

Les critères et le protocole de l'évaluation sont les mêmes que ceux évoqués précédemment.

L'évaluation finale du modèle ranger donne la matrice de confusion suivante :

La précision finale est égale à 0.9998, avec un intervalle de confiance à 95% de [0.9994 ; 1]. La forêt aléatoire de type Ranger a donné d'excellents résultats, en un temps très contenu (2.83 min), preuve de sa grande efficacité calculatoire.

La forêt aléatoire de type Rborist a donné des résultats similaires, avec une précision finale égale à 1, avec un intervalle de confiance à 95% de [0.9997 ; 1]. Le modèle Rborist, donnant des résultats sensiblement identiques à Ranger, s'est avéré extrêmement efficace sur le plan calculatoire (13.61 min).

Table 13: Matrice de confusion du modèle Ranger

	e	p
e	5568	2
p	0	7166

Table 14: Performances des modèles Ranger et Rborist (évaluation)

	Sensibilité	Spécificité	F1 score	Durée (min)
Ranger	1	0.99972	0.99982	2.83
Rborist	1	1.00000	1.00000	13.61

6.2 Classification par espèce

7 Robustesse de la classification

texte

8 Références bibliographiques

1. Schlimmer J. Mushroom Data Set. University of California. 1987; Disponible sur: <https://archive.ics.uci.edu/ml/datasets/Mushroom>
2. Wagner D, Heider D, Hattab G. Mushroom data creation, curation, and simulation to support classification tasks. Scientific Reports [Internet]. avr 2021 [cité 10 déc 2022];11(1):8134. Disponible sur: <https://www.nature.com/articles/s41598-021-87602-3>
3. Wickham H. tidyverse: Easily Install and Load the Tidyverse [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=tidyverse>
4. Mersmann O. microbenchmark: Accurate Timing Functions [Internet]. 2021. Disponible sur: <https://github.com/joshuaulrich/microbenchmark/>
5. Vakayil A, Joseph R, Mak S. SPlit: Split a Dataset for Training and Testing [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=SPlit>
6. Ripley B. MASS: Support Functions and Datasets for Venables and Ripley's MASS [Internet]. 2023. Disponible sur: <http://www.stats.ox.ac.uk/pub/MASS4/>
7. Kuhn M. caret: Classification and Regression Training [Internet]. 2022. Disponible sur: <https://github.com/topepo/caret/>
8. Schloerke B, Cook D, Larmarange J, Briatte F, Marbach M, Thoen E, et al. GGally: Extension to ggplot2 [Internet]. 2021. Disponible sur: <https://CRAN.R-project.org/package=GGally>
9. Trevor Hastie & Robert Tibshirani. Original R port by Friedrich Leisch S original by, Hornik K, code. BDRipleyBN has contributed to the upgrading of the. mda: Mixture and Flexible Discriminant Analysis [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=mda>
10. Therneau T, Atkinson B. rpart: Recursive Partitioning and Regression Trees [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=rpart>
11. Wickham H. plyr: Tools for Splitting, Applying and Combining Data [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=plyr>
12. Kuhn M, Quinlan R. C50: C5.0 Decision Trees and Rule-Based Models [Internet]. 2023. Disponible sur: <https://topepo.github.io/C5.0/>
13. Hothorn T, Hornik K, Strobl C, Zeileis A. party: A Laboratory for Recursive Partytioning [Internet]. 2022. Disponible sur: <http://party.R-forge.R-project.org>
14. Wright MN, Wager S, Probst P. ranger: A Fast Implementation of Random Forests [Internet]. 2022. Disponible sur: <https://github.com/imbs-hl/ranger>
15. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien [Internet]. 2023. Disponible sur: <https://CRAN.R-project.org/package=e1071>
16. Kursá MB. rFerns: Random Ferns Classifier [Internet]. 2021. Disponible sur: <https://gitlab.com/mbq/rFerns>

17. Seligman M. Rborist: Extensible, Parallelizable Implementation of the Random Forest Algorithm [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=Rborist>
18. Allaire J, Xie Y, McPherson J, Luraschi J, Ushey K, Atkins A, et al. rmarkdown: Dynamic Documents for R [Internet]. 2023. Disponible sur: <https://CRAN.R-project.org/package=rmarkdown>
19. Xie Y. knitr: A General-Purpose Package for Dynamic Report Generation in R [Internet]. 2022. Disponible sur: <https://yihui.org/knitr/>
20. Kassambara A. ggpubr: ggplot2 Based Publication Ready Plots [Internet]. 2023. Disponible sur: <https://rpkgs.datanovia.com/ggpubr/>
21. Franco J, Dupuy D, Roustant O, Kiener P, Damblin G, looss. B. DiceDesign: Designs of Computer Experiments [Internet]. 2021. Disponible sur: <http://dice.emse.fr>
22. Dupuy D, Helbert C. DiceEval: Construction and Evaluation of Metamodels [Internet]. 2022. Disponible sur: <https://CRAN.R-project.org/package=DiceEval>
23. Xie Y. bookdown: Authoring Books and Technical Documents with R Markdown [Internet]. 2023. Disponible sur: <https://CRAN.R-project.org/package=bookdown>
24. Courtecuisse R. Clé de détermination macroscopique des champignons supérieurs des régions du Nord de la France. Société mycologique du Nord de la France; 1986.
25. Courtecuisse R, Duhem B. Champignons de France et d'Europe. Delachaux et Niestlé; 2013. (Guides Delachaux).
26. Courtecuisse R, Moreau PA, Welti S. Initiation à la reconnaissance des champignons du Nord de la France - Clé pour la détermination des espèces les plus fréquentes. Département des Sciences Végétales et Fongiques, Faculté de Pharmacie de Lille; 2020.
27. Money NP. Insights on the mechanics of hyphal growth. Fungal Biology Reviews [Internet]. 2008 [cité 11 févr 2023];22(2):71-6. Disponible sur: <https://www.sciencedirect.com/science/article/pii/S1749461308000195>
28. Porter DL, Naleway SE. Hyphal systems and their effect on the mechanical properties of fungal sporocarps. Acta Biomaterialia [Internet]. juin 2022 [cité 11 févr 2023];145:272-82. Disponible sur: <https://www.sciencedirect.com/science/article/pii/S1742706122002161>
29. Johnson NL. Continuous univariate distributions, volume 2. 2nd ed. New York [etc: John Wiley & sons; 1995. (Wiley series in probability et mathematical statistics Applied probability et statistics; vol. 2).
30. R Core Team. R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2021. Disponible sur: <https://www.R-project.org/>
31. Brownlee J. What is the Difference Between Test and Validation Datasets? [Internet]. MachineLearningMastery.com. 2017 [cité 14 févr 2023]. Disponible sur: <https://machinelearningmastery.com/difference-test-validation-datasets/>

32. Joseph VR. Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal* [Internet]. 2022 [cité 15 févr 2023];15(4):531-8. Disponible sur: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11583>
33. Mak S, Joseph VR. Support points. *The Annals of Statistics* [Internet]. déc 2018 [cité 15 févr 2023];46(6A):2562-92. Disponible sur: <https://projecteuclid.org/journals/annals-of-statistics/volume-46/issue-6A/Support-points/10.1214/17-AOS1629.full>
34. Joseph VR, Vakayil A. SPlit: An Optimal Method for Data Splitting. *Technometrics* [Internet]. avr 2022 [cité 15 févr 2023];64(2):166-76. Disponible sur: <https://doi.org/10.1080/00401706.2021.1921037>
35. Santiago J, Claeys-Bruno M, Sergent M. Construction of space-filling designs using WSP algorithm for high dimensional spaces. *Chemometrics and Intelligent Laboratory Systems* [Internet]. avr 2012 [cité 4 mars 2023];113:26-31. Disponible sur: <https://linkinghub.elsevier.com/retrieve/pii/S0169743911001195>
36. Youden WJ. Index for rating diagnostic tests. *Cancer* [Internet]. 1950 [cité 4 mars 2023];3(1):32-5. Disponible sur: <https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0142%281950%293%3A1%3C32%3A%3AAID-CNCR2820030106%3E3.0.CO%3B2-3>
37. Rücker G, Schumacher M. Summary ROC curve based on a weighted Youden index for selecting an optimal cutpoint in meta-analysis of diagnostic accuracy. *Statistics in Medicine* [Internet]. 2010 [cité 4 mars 2023];29(30):3069-78. Disponible sur: <http://onlinelibrary.wiley.com/doi/abs/10.1002/sim.3937>

A Annexe : développement d'un algorithme de génération de lot synthétique

Plus de détails concrets sur la méthode utilisée en pratique, avec extraits de code. . .

Pour l'instant, l'algorithme est écrit et fonctionne, à partir des données primaires du Secondary Dataset de Dennis Wagner. C'est cet algorithme qui a servi à créer le lot de données de la classification multiclasse.

Pour avoir mes propres lots de données, il ne me reste plus qu'à avoir mes propres données primaires, c'est à dire entrer manuellement les caractéristiques clés de mes 400+ champignons, un par un. . .

La seule bibliothèque utilisée lors de la création du lot de données synthétique est le *tidyverse*³, collection de bibliothèques spécialisées dans le domaine de la *data science* et notamment dédiées au traitement, au nettoyage et à la visualisation de données.

```
library(tidyverse)
```

Nous chargeons ensuite le fichier zip, incluant le fichier csv contenant les caractéristiques typiques des macromycètes (type de sporophore, dimensions maximales du stipe, du chapeau, type de lames, couleur de sporée, etc.), qui est lu et attribué à un *dataframe*. Les lignes commentées correspondent à l'utilisation d'un fichier identique hébergé à distance sur un dépôt GitHub.

```
fichier_data <- tempfile()
#URL <- "https://github.com/EKRihani/mushrooms/raw/master/MushroomDataset.zip"
#download.file(URL, fichier_data)
fichier_data <- "~/projects/champis/MushroomDataset.zip"      # Fichier local
fichier_data <- unzip(fichier_data, "MushroomDataset/primary_data.csv")
data_champis <- read.csv(fichier_data,
                        header = TRUE,
                        sep = ";",
                        stringsAsFactors = TRUE)
```

etc etc.

B Annexe : outils d'analyse exploratoire des données (EDA)

Code EDA...

C Annexe : développement des algorithmes d'apprentissage machine

Le gros du code...

Nous détaillerons ici principalement les algorithmes utilisés pour le classifieur binaire. Les particularités d'intérêt des classifieurs multiples seront évoquées brièvement lors des développements de cette section.

C.1 Initialisation

Les bibliothèques utilisées lors des étapes d'apprentissage machine sont :

- tidyverse³, collection de bibliothèques spécialisées dans le domaine de la *data science*,
- DiceDesign²¹, bibliothèque spécialisée dans la création de plans d'expériences hypercubiques,
- DiceEval²², bibliothèque spécialisée dans la modélisation des résultats de plans d'expériences hypercubiques,
- caret⁷, collection d'outils dédiés à l'apprentissage machine.

```
library(tidyverse)
library(DiceDesign)
library(DiceEval)
library(caret)
```

Le chargement des données s'effectue de la même façon que lors des sections précédentes. Dans le cadre d'une classification binaire, nous attribuons arbitrairement, à l'aide de la fonction *relevel*, la valeur positive à la classe "toxique".

```
fichier_data <- tempfile()
fichier_data <- "~/projects/champis/MushroomDataset.zip"
fichier_data <- unzip(fichier_data, "MushroomDataset/secondary_data.csv")
dataset <- read.csv(fichier_data,
                    header = TRUE,
                    sep = ";",
                    stringsAsFactors = TRUE)
dataset$class <- relevel(dataset$class, ref = "toxique")
```

C.2 Création des jeux d'entraînement, optimisation et évaluation

La création du jeu d'évaluation s'effectue par dichotomie du jeu de données avec un ratio 92/8.

```
set.seed(7)
split1 <- 0.08
index1 <- createDataPartition(y = dataset$cap.diameter,
                               times = 1,
                               p = split1,
                               list = FALSE)
BI_lot_appr_opti <- dataset[-index1,]
BI_lot_evaluation <- dataset[index1,]
```

Université de Lille
FACULTE DE PHARMACIE DE LILLE
DIPLOME D'ETAT DE DOCTEUR EN PHARMACIE
Année Universitaire 2022/2023

Nom : RIHANI

Prénom : Emir Kaïs

Titre de la thèse : Application de modèles d'intelligence artificielle à la classification des macromycètes

Mots-clés : intelligence artificielle, apprentissage machine, *machine learning*, classification, mycologie

Résumé : L'IA c'est génial !

Membres du jury :

Président : Nom, Prenom, titre et lieu de fonction

Assesseur(s) : Nom1, Prenom1, titre et lieu de fonction

Nom2, Prenom2, titre et lieu de fonction

Nom3, Prenom3, titre et lieu de fonction

Membre(s) extérieur(s) : Nom1, Prenom1, titre et lieu de fonction

Nom2, Prenom2, titre et lieu de fonction

Nom3, Prenom3, titre et lieu de fonction