Modèle de classification supervisée avec copules

par

Amel Mansouri

mémoire présenté au Département de mathématiques en vue de l'obtention
du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, septembre 2025

Le 19 Août 2025

Membres du jury

Professeur Taoufik Bouezmarni

Directeur de recherche

Département de mathématiques

Professeur Mohamed Chaouch

Co-Directeur de recherche

Qatar University

Professeur Éric Marchand

Membre interne

Département de mathématiques

Professeur Félix Camirand Lemyre

Président-rapporteur

Département de mathématiques

SOMMAIRE

Dans ce mémoire, nous étudions diverses méthodes de classification statistique, et nous focalisons sur la comparaison entre des techniques classiques de classification supervisée, telles que la régression logistique, (SVM) et des méthodes fondées sur l'utilisation des copules. Pour ce faire, nous générons des données simulées à partir de différentes familles de copules (Gaussienne, t-Student, Clayton, Gumbel, Frank et Joe), ainsi que selon les modèles logistique et probit. Ces jeux de données sont ensuite soumis à plusieurs algorithmes de classification afin d'évaluer leurs performances respectives au moyen de métriques standards telles que le score F1, la Précision, la Sensibilité et l'exactitude (Accuracy). L'objectif principal de ce travail est d'identifier les conditions sous lesquelles les méthodes à base de copules permettent une amélioration significative des performances de classification, et de mieux cerner leur potentiel dans un cadre supervisé. Enfin, nous discutons les limites de notre approche et proposons quelques pistes d'amélioration, notamment dans le contexte de données réelles ou de grande dimension.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce travail. À ma famille et particulièrement à mon mari, dont le soutien indéfectible et les encouragements constants ont été mon moteur tout au long de ce parcours exigeant.

Mes remerciements les plus sincères vont à mon directeur de recherche, le Professeur Taoufik Bouezmarni, pour sa patience, et la rigueur scientifique qu'il m'a transmise. Son accompagnement attentif et ses conseils précieux ont été essentiels à chaque étape de ce mémoire, et je lui suis profondément reconnaissante pour la confiance qu'il m'a accordée. Je suis également reconnaissante envers mon co-directeur, le Professeur Mohamed Chaouch, pour ses conseils avisés et son regard critique qui ont grandement enrichi cette recherche.

Je n'oublie pas de remercier l'administration de l'Université de Sherbrooke pour l'accompagnement administratif et les facilités accordées qui ont rendu possible l'aboutissement de ce projet. Enfin, à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire, je dis un sincère merci.

Amel Mansouri

Sherbrooke, 19/08/2025

TABLE DES MATIÈRES

LISTE DES FIGURES

INTRODUCTION

Dans l'analyse statistique contemporaine, comprendre la relation entre une variable cible et des variables explicatives est un enjeu central, notamment dans les problématiques de classification binaire. Ces situations sont fréquentes, par exemple lorsqu'on cherche à prédire une issue qualitative (succès/échec, maladie/santé, etc.) à partir de données observées. Les approches paramétriques classiques comme la régression logistique ou probit ont longtemps constitué des outils de référence. Cependant, elles reposent sur des hypothèses structurelles pouvant s'avérer restrictives, notamment lorsqu'elles ne capturent pas fidèlement la structure de dépendance présente dans les données.

Dans ce contexte, les *copules* offrent une alternative prometteuse. Elles permettent de modéliser de manière flexible et distincte les dépendances entre variables, indépendamment de leurs distributions marginales. Grâce à cela, les copules se révèlent particulièrement utiles pour simuler des données réalistes et complexes, et pour tester la robustesse et l'efficacité de différentes méthodes de classification.

Ce mémoire propose une étude de simulation comparative entre plusieurs classifieurs populaires, en s'appuyant sur des données générées à l'aide de différentes familles de copules (Gaussienne, t-Student, Clayton, Gumbel, Frank et Joe), ainsi que deux modèles standards : la régression logistique et la régression probit. L'objectif principal est d'évaluer la performance de ces méthodes dans des contextes de dépendance contrôlée, selon divers

critères d'évaluation (F1 Score, Accuracy, Précision, Sensibilité).

Le contenu de ce mémoire est organisé comme suit :

— **Le premier chapitre** est consacré à la présentation des principales méthodes de classification supervisée utilisées dans la littérature. Il introduit notamment la régression logistique, le SVM, les réseaux de neurones, les k-plus proches voisins (k-NN), l'analyse discriminante linéaire (LDA), les arbres de décision (CART), ainsi que les méthodes d'ensemble telles que les forêts aléatoires, le bagging et le boosting. Ce chapitre fournit également les notions essentielles liées aux mesures de performance utilisées tout au long de l'étude.

— **Le deuxième chapitre** présente la théorie des copules, en commençant par leur définition, les familles les plus utilisées, et leurs propriétés mathématiques. Il est également question des méthodes de simulation de données à partir de copules, ainsi que de leur utilité pour modéliser la dépendance entre variables aléatoires dans un cadre de classification.

— **Le troisième chapitre** introduit les processus générateurs de données (DGP) utilisés dans l'étude. Il expose les paramètres choisis (tailles d'échantillons, niveaux de dépendance en utilisant le tau de Kendall et le nombre de répétitions), ainsi que les résultats obtenus pour chaque classifieur et chaque modèle. Une analyse comparative est proposée sur la base des moyennes, médianes et écarts-types des scores F1, de la précision, de la sensibilité et de l'exactitude. Des tableaux synthétiques permettent d'identifier les méthodes les plus performantes selon les contextes.

Ce travail vise à éclairer le choix des méthodes de classification en fonction de la structure de dépendance sous-jacente aux données. Il met en évidence les situations dans lesquelles certaines approches surpassent les autres, et offre une base solide pour des travaux futurs en apprentissage supervisé sous des formes de dépendance complexes.

CHAPITRE 1

Modèles de classification : Revue de la littérature

Dans ce chapitre, nous introduisons une diversité de méthodes et de modèles de classification, en distinguant les approches paramétriques (telles que les modèles logistiques et l'analyse discriminante, le modèle de mélange), les approches semi-paramétriques (comme le modèle à indice logistique), ainsi que les méthodes non-paramétriques (par exemple, les k-plus-proches-voisins et les arbres de décision de type CART). La section 1 est consacrée à l'analyse discriminante (AFD), suivie de la section 2, qui traite des modèles logistiques. La section 3 explore les modèles de mélanges, tandis que la section 4 examine la méthode des k-plus-proches-voisins. La section 5 est dédiée aux méthodes de segmentation, en particulier l'algorithme CART. La section 6 introduit les méthodes d'ensemble, notamment le boosting, le bagging et les forêts aléatoires. La section 7 est consacrée aux machines à vecteurs de support (SVM). Enfin, la section 8 est dédiée aux réseaux de neurones, qui sont présentés avec son algorithme.

1.1 Analyse factorielle discriminante

Introduite par Ronald A. Fisher en 1936 [Fis36], l'analyse factorielle discriminante (AFD) constitue une approche intermédiaire entre une méthode descriptive et une méthode décisionnelle. Elle s'applique aux données quantitatives où n observations sont réparties à priori en k catégories ; chaque observation étant décrite par p variables. Deux approches principales sont fréquemment utilisées.

D'une part, l'AFD peut être exploitée comme une technique descriptive, visant à représenter graphiquement les k groupes à l'aide de nouvelles variables latentes dérivées des variables explicatives. Dans cette perspective, l'analyse cherche à réduire la dimensionnalité tout en maximisant la séparation entre groupes. D'autre part, l'AFD représente une extension de l'analyse en composantes principales (ACP) appliquée aux centres de gravité des catégories mais en intégrant une métrique spécifique adaptée à la discrimination inter-groupes. Dans la littérature anglo-saxonne, cette méthode est également désignée sous le terme d'analyse canonique discriminante.

Dans une seconde approche, l'AFD vise à établir une fonction de classification fondée sur les valeurs des variables prédictives, permettant ainsi de prédire l'appartenance de chaque observation à un groupe donné. Cette approche s'apparente aux méthodes supervisées d'apprentissage automatique (telles que : les arbres de décision et les réseaux de neurones) qui s'appuient généralement sur des hypothèses probabilistes. Parmi celles-ci, l'hypothèse de distribution multinormale des données et l'homoscédasticité (égalité des matrices de covariance entre classes), conduisant à la modélisation par l'analyse discriminante linéaire. L'AFD est largement utilisée en raison de sa simplicité d'interprétation, la fonction de classification étant exprimée sous forme d'une combinaison linéaire des variables prédictives, facilitant ainsi l'analyse des contributions de chaque variable à la séparation des groupes. Dans le cas des matrices de covariances différentes selon les

classes, on parle de l'analyse discriminante quadratique.

1.1.1 Notations

Nous cherchons à modéliser une variable qualitative comportant K classes à partir d'un ensemble de p variables explicatives quantitatives. Considérons un échantillon de n observations de la variable à expliquer, notée Y, ainsi des variables explicatives, notées $X_1, \ldots, X_p$.

Nous représentons alors les valeurs observées de Y sur l'ensemble des individus par le vecteur $\mathbf{y} = (y_1, \ldots, y_n)^t$ et nous notons $\mathbf{X}$ la matrice des valeurs des variables explicatives ; chaque élément $(\mathbf{X})_{i,j} = \boldsymbol{x}_{ij}$ correspond à la valeur de la j-ème variable pour le i-ème individu. Nous définissons comme variables explicatives de l'individu i, le vecteur $x_i = (x_{i1}, \ldots, x_{ip})$ correspond à la i-ème ligne de la matrice $\mathbf{X}$, regroupant l'ensemble des observations de cet individu. Par ailleurs, nous supposons que le nombre de variables explicatives p est inférieur au nombre d'observation n, soit $p < n$, une hypothèse couramment utilisée en pratique.

(a) Variances globale, inter et intra-classe. Nous introduisons ici trois conceptes fondamentaux de l'AFD. Pour commencer, introduisons quelques notations essentielles. Soit $\mathbf{g} := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ le center de gravité du nuage de points, et $\mathbf{g}_k := \frac{1}{n_k}\sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i$ le centre de gravité de la classe C_k, pour $k = 1, \ldots, K$.

— La matrice de covariance globale, de dimension $p \times p$, est définie par :

$$V_T := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^t - \mathbf{g}\mathbf{g}^t.$$

— La matrice de covariance de la classe C_k est définie par :

$$V_k := \frac{1}{n_k}\sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i \boldsymbol{x}_i^t - \mathbf{g}_k \mathbf{g}_k^t.$$

— Les matrices de covariance intra-classe et inter-classes, notées respectivement V_W et V_B, sont définies par :

$$V_W = \frac{1}{n}\sum_{k=1}^{K} n_k V_k \quad \text{et} \quad V_B = \frac{1}{n}\sum_{k=1}^{K} n_k \mathbf{g}_k \mathbf{g}_k^t - \mathbf{g}\mathbf{g}^t.$$

Une relation fondamentale lie ces trois matrices est donnée par :

$$V_T = V_B + V_W.$$

Remarque 1. *Dans de nombreuses analyses, notamment en analyse en composantes principales, on suppose que les données sont centrées ; c'est-à-dire que le centre de gravité du nuage de points,* $\mathbf{g}$*, est situé à l'origine. Lorsque cette condition n'est pas satisfaite, une translation du nuage de points permet de recentrer les données sans altérer leur structure intrinsèque. Cette transformation, purement affine, préserve les relations entre observations tout en simplifiant considérablement les calculs. Dans ce cadre, les expressions des matrices de covariance globale et inter-classes s'écrivent alors comme suit :*

$$V_T = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^t, \quad V_B = \frac{1}{n}\sum_{k=1}^{K} n_k \mathbf{g}_k \mathbf{g}_k^t.$$

Jusqu'à présent, nous avons supposé que tous les individus étaient affectés d'un poids uniforme égal à $1/n$*. Toutefois, il est possible de généraliser ces résultats en introduisant des poids spécifiques* p_i *associés à chaque individu* i*. Dans ce cadre, les poids des classes sont définis par* $q_k = \sum_{\boldsymbol{x}_i \in C_k} p_i$*. Ainsi les quantités introduisant des précédemment définies s'expriment sous la forme suivante :*

$$\mathbf{g}_k = \frac{1}{q_k}\sum_{\boldsymbol{x}_i \in C_k} p_i \boldsymbol{x}_i, \quad \mathbf{g} = \sum_{i=1}^{n} p_i \boldsymbol{x}_i, \quad V_k = \frac{1}{q_k}\sum_{\boldsymbol{x}_i \in C_k} p_i \boldsymbol{x}_i \boldsymbol{x}_i^t - \mathbf{g}_k \mathbf{g}_k^t,$$

$$V_B = \sum_{k=1}^{K} q_k \mathbf{g}_k \mathbf{g}_k^t - \mathbf{g}\mathbf{g}^t \quad \textit{et} \quad V_W = \sum_{k=1}^{K} q_k V_k.$$

Dans ce cas, on parle d'une variance pondérée ou d'une matrice d'inertie plutôt que de la variance simple.

(b) Choix de la distance. Il est essentiel de définir une notion de distance entre les individus de l'échantillon afin d'établir des comparaisons quantitatives. À cette fin, nous introduisons la forme quadratique suivante, construite à partir d'une matrice symétrique M, définie positive, et de dimension $p \times p$:

$$d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^t M (\boldsymbol{x}_i - \boldsymbol{x}_j).$$

Le choix de la matrice M est à la discrétion de l'utilisateur, offrant ainsi la possibilité d'attribuer des pondérations différenciées aux différentes dimensions. Cette flexibilité est particulièrement utile lorsque les variables sont exprimées dans des unités distinctes, garantissant ainsi une comparabilité adéquate des mesures.

(c) Inertie. L'inertie totale, notée I_T, du nuage de points est définie comme la somme des carrés des distances entre chaque point et le centre de gravité du nuage. Formellement, elle s'exprime comme suit :

$$I_T = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \mathbf{g})^t M (\boldsymbol{x}_i - \mathbf{g}).$$

Il est possible de démontrer que $I_T = \text{Trace}(V_T M)$.

1.1.2 Principes généraux

Selon [LPM06], l'analyse factorielle discriminante vise à identifier un nouvel ensemble de variables, appelées variables discriminantes, permettant d'optimiser la séparation des K groupes après projection sur des axes appropries. La figure 1.1 illustre cette méthode. En particulier, la projection des données sur l'axe en pointillés met en évidence la manière dont une séparation optimale entre les groupes peut-être obtenue.

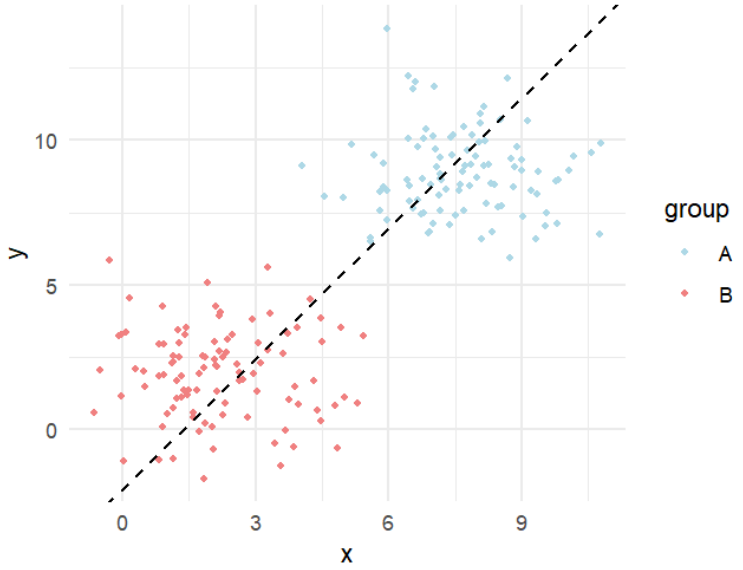

FIGURE 1.1 – Discrimination des deux groupes : la meilleure séparation est obtenue après projection sur l'axe en pointillés.

Considérons un axe défini par un vecteur directeur $\mathbf{u}$. L'inertie projetée sur cet axe $\mathbf{u}$ avec la métrique M, peut être décomposée de la manière suivante :

$$\mathbf{u}^t M V_T M \mathbf{u} = \mathbf{u}^t M V_B M \mathbf{u} + \mathbf{u}^t M V_W M \mathbf{u}.$$

Un axe possède un fort pouvoir discriminant si la projection des points sur cet axe génère des groupes bien séparés et homogènes. Autrement dit, il s'agit de maximiser l'inertie inter-classes définie par $\mathbf{u}^t M V_B M \mathbf{u}$ et de minimiser l'inertie intra-classes donnée par $\mathbf{u}^t M V_W M \mathbf{u}$. D'un point de vue pratique, cet objectif revient à maximiser le critère discriminant suivant :

$$g(\mathbf{u}) = \frac{\mathbf{u}^t M V_B M \mathbf{u}}{\mathbf{u}^t M V_T M \mathbf{u}}.$$

Théorème 1.1. *Si la matrice V_T est inversible, alors le critère discriminant g est maximisé pour le vecteur propre, $\mathbf{v_1}$, de $V_T^{-1} V_B$, correspondant à la plus grande valeur propre λ_1. Autrement dit $\lambda_1 = \max_{\mathbf{u}} g(\mathbf{u})$.*

Preuve du théorème 1.1 Considérons la fonction à maximiser :

$$g(\mathbf{u}) = \frac{\mathbf{u}^t M V_B M \mathbf{u}}{\mathbf{u}^t M V_T M \mathbf{u}},$$

où M est une matrice définie positive.

Pour déterminer le maximum de $g(\mathbf{u})$, nous calculons son gradient, $\nabla g(\mathbf{u})$, et posons :

$$\nabla g(\mathbf{u}) = 0.$$

Ce gradient s'écrit explicitement sous la forme :

$$\nabla g(\mathbf{u}) = \frac{(2\mathbf{u}^t M V_T M \mathbf{u})(M V_B M \mathbf{u}) - (2\mathbf{u}^t M V_B M \mathbf{u})(M V_T M \mathbf{u})}{\left(\mathbf{u}^t M V_T M \mathbf{u}\right)^2}.$$

Pour que le gradient s'annule, il faut satisfaire l'égalité suivante :

$$\left(\mathbf{u}^t M V_T M \mathbf{u}\right) M V_B M \mathbf{u} = \left(\mathbf{u}^t M V_B M \mathbf{u}\right) M V_T M \mathbf{u}.$$

En multipliant des deux côtés par M^{-1} (matrice inversible), nous obtenons :

$$\left(\mathbf{u}^t M V_T M \mathbf{u}\right) V_B M \mathbf{u} = \left(\mathbf{u}^t M V_B M \mathbf{u}\right) V_T M \mathbf{u}.$$

En divisons par $\mathbf{u}^t M V_T M \mathbf{u}$ (strictement positif, puisque $M V_T M$ est définie positive), il vient que :

$$V_B M \mathbf{u} = g(\mathbf{u}) V_T M \mathbf{u}.$$

En posons $\mathbf{v} = M\mathbf{u}$, l'équation se réécrit sous la forme :

$$V_B \mathbf{v} = g(\mathbf{u}) V_T \mathbf{v}.$$

On en déduit que $\mathbf{v}$ est un vecteur propre de la matrice $V_T^{-1} V_B$, et que $g(\mathbf{u})$ correspond à la valeur propre associée.

Remarque 2. *Le choix de la métrique M n'affecte pas la détermination de l'axe le plus discriminant. Afin de simplifier les calculs, nous supposerons donc que $M = I$ dans la suite du document.*

Définition 3. *Le premier facteur discriminant, noté $\mathbf{u}_1$, est défini comme le vecteur propre associé à la plus grande valeur propre de la matrice $V_T^{-1}V_B$. La première variable discriminante est alors donnée par $\mathbf{v}_1 = \mathbf{X}\mathbf{u}_1$, où le pouvoir discriminant associé à l'axe $\mathbf{u}_1$ est représenté par λ_1.*

1.1.3 Nombre d'axes discriminants

Après identification du premier axe discriminant, les axes discriminants supplémentaires peuvent être déterminés en cherchant les autres valeurs propres et vecteurs propres de la matrice $V_T^{-1}V_B$. Par ailleurs, il est établi que le rang de $V_T^{-1}V_B$ est au plus égal au minimum des rangs de matrices V_T et V_B, soit $\text{rang}(V_T^{-1}V_B) \leq \min(\text{rang}(V_T), \text{rang}(V_B))$. En pratique, sous l'hypothèse que la matrice des données $\mathbf{X}$ est de plein rang p, cette relation implique que $\text{rang}(V_T^{-1}V_B) \leq \min(p, K-1)$. Ainsi, lorsque le nombre de classes est inférieur au nombre de variables, c'est-à-dire si $K - 1 < p$, le nombre maximal d'axes discriminants est donné par $K - 1$.

Proposition 4. *Voici quelques propriétés fondamentales des vecteurs propres associés à l'analyse factorielle discriminante :*

- *Comme $0 \leq \mathbf{u}^t V_B \mathbf{u} \leq \mathbf{u}^t V_T \mathbf{u}$, soit $0 \leq g(\mathbf{u}) \leq 1$, on en déduit que la plus grande valeur propre vérifie $\lambda_1 \in [0, 1]$.*
- *Si $\lambda_1 = 0$, alors $\mathbf{u}_1^t V_B \mathbf{u}_1 = 0$, cela signifie que l'inertie inter-classes après projection sur $\mathbf{u}_1$ est nulle. Sous l'hypothèse où les $\boldsymbol{x}_i$ sont tous distincts, cela implique que l'axe discriminant optimal ne permet pas de séparer les centres de gravité des classes, ces derniers se confondent. Toutefois, une séparation non linéaire peut éxis-*

ter, par exemple lorsque les classes sont disposées en cercles concentriques, partageant un même centre geometrique (barycentre).

— *Si* $\lambda_1 = 1$*, alors* $\mathbf{u}_1^t V_B \mathbf{u}_1 = \mathbf{u}_1^t V_T \mathbf{u}_1$*, cela signifie que l'inertie intra-classes après projection sur* $\mathbf{u}_1$ *est nulle. Dans ce cas, les projections des points* $\boldsymbol{x}_i$ *sur l'axe discriminant* $\mathbf{u}_1$ *coïncident exactement avec les centres de gravité des classes* $\mathbf{g}_k$*. La discrimination est alors parfaite si les projections des centres de gravité sont distinctes.*

— *La valeur propre* λ_1 *représente une mesure conservatrice du pouvoir discriminant. En effet,* λ_1 *est inférieure à* 1 *même lorsqu'une discrimination parfaite est théoriquement possible.*

1.1.4 Lien avec l'analyse en composantes principales

L'analyse en composantes principales (ACP) a pour objectif d'identifier un sous-espace de faible dimension capturant la variabilité observée dans un ensemble de données multivariées. Lorsqu'un jeu de données est décrit par p variables, l'objectif est de déterminer q combinaisons linéaires de ces variables, avec $q < p$, permettant de préserver l'essentiel de la variabilité du nuage des points.

En parallèle, l'analyse factorielle discriminante (AFD) vise à identifier des combinaisons linéaires des p variables initiales qui optimisent la séparation entre les groupes présents dans le nuage de points. Si K groupes sont définis, avec $K < p$, il est possible d'extraire au maximum $K - 1$ variables discriminantes.

Il convient de noter que l'AFD peut être interprétée comme une ACP appliquée aux centres de gravité des classes, en adoptant la métrique V_T^{-1}. Les variables discriminantes ainsi obtenues sont mutuellement orthogonales, garantissant une indépendance entre les axes discriminants. De plus, ces variables peuvent être analysées et interprétées à l'aide du

cercle des corrélations, facilitant leur interprétation dans l'éspace des variables initiales.

1.2 Modèles Logistique

La régression logistique, introduite dans de nombreux travaux classiques tels que [MN83], [DS66] et [Dob90], est une méthode statistique utilisée pour modéliser la relation entre une variable de réponse binaire ($Y \in \{0, 1\}$) et un ensemble de variables explicatives $\mathbf{X} = (X_1, \ldots, X_p)$, qui peuvent être catégorielles (par exemple, le sexe) ou continues (par exemple, l'âge). Ce modèle, également connu sous le nom de modèle logit, est spécifiquement conçu pour des tâches de classification et d'analyse prédictive. Son objectif principal est d'estimer la probabilité d'occurrence d'un événement ($P(Y = 1|\mathbf{X})$) en fonction des prédicteurs, grâce à la fonction logistique :

$$P(Y = 1|\mathbf{X}) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i X_i)}}.$$

Contrairement à la régression linéaire, la régression logistique est adaptée aux variables quantitatives continue, variables nominales, telles que "oui/non" ou "1/0". Ce modèle repose sur la fonction logit, qui permet de transformer la relation non linéaire entre les variables explicatives et la probabilité de l'évènement étudié, en la linéarisant dans l'espace des logits. Ce modèle trouve des applications dans des domaines variés, comme la médecine (prédiction de guérison), la finance (évaluation des risques), et le marketing (analyse d'achat). On distingue trois types de modèles de régression logistique, qui sont définis en fonction de la réponse nominale Y.

Régression logistique binaire : Ce type de régression est utilisé lorsque la variable dépendante est dichotomique, c'est-à-dire qu'elle prend uniquement deux valeurs possibles, comme 0 ou 1. Ce modèle est couramment utilisé dans des applications telles que

la détection de spam dans les e-mails, diagnostique médical (présence ou absence d'une maladie) ou prédiction de l'acceptation d'un prêt bancaire (accordé ou refusé).

Régression logistique multinomiale : Lorsque la variable dépendante comporte plus de deux catégories sans ordre hiérarchique, on utilise un modèle de régression logistique multinomiale. Par exemple, un studio de cinéma peut prédire le genre de film qu'un individu est susceptible de préférer (comédie, action, drame, etc.) en fonction de variables explicatives comme l'âge, le sexe ou le statut marital. Ce type de modèle permet d'orienter les campagnes marketing vers des groupes spécifiques.

Régression logistique ordinale : Ce modèle s'applique lorsque la variable dépendante présente plusieurs catégories ordonnées. Contrairement au cas multinomial, les modalités possèdent une relation d'ordre. Par exemple, des notes académiques (A, B, C, etc.) ou des échelles de satisfaction (de 1 à 5) constituent des données ordinales.

1.2.1 Estimation du modèle logistique avec la méthode du maximum de vraisemblance

Considérons un ensemble de données $(\mathbf{X}_i, Y_i)_{i=1,\dots,n}$, où $\mathbf{X}_i \in \mathbb{R}^p$ représente les variables explicatives et $Y_i \in \{0, 1\}$ est la variable binaire de réponse. La probabilité conditionnelle que $Y_i = 1$, était donné les prédicteurs $\mathbf{X}_i = (X_{i1}, \dots, X_{ip})$, est définie par la fonction logistique suivante :

$$P(Y_i = 1 \mid \mathbf{X}_i; \boldsymbol{\beta}) = P(\mathbf{X}_i; \boldsymbol{\beta}) = \frac{\exp(\mathbf{X}_i^T \boldsymbol{\beta})}{1 + \exp(\mathbf{X}_i^T \boldsymbol{\beta})},$$

où $\boldsymbol{\beta} \in \mathbb{R}^p$ est le vecteur des coefficients à estimer.

Remarque 5. *Dans le modèle logistique classique, le lien entre la probabilité conditionnelle de succès et le prédicteur $\mathbf{X}_i^T \boldsymbol{\beta}$ est pré-déterminé. Le modèle logistique semi-*

paramétrique, également appelé index model, est défini par :

$$P(Y_i = 1 \mid \mathbf{X}_i; \boldsymbol{\beta}) = g(\mathbf{X}_i^T \boldsymbol{\beta})$$

où g est une fonction inconnue. Ce type de modèle constitue une approche semi-paramétrique couramment utilisée pour modéliser des réponses discrètes.

La fonction de vraisemblance associée au modèle est donnée par :

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} P(\mathbf{X}_i; \boldsymbol{\beta})^{Y_i} \left[1 - P(\mathbf{X}_i; \boldsymbol{\beta})\right]^{1-Y_i}.$$

En prenant le logarithme de la vraisemblance, nous obtenons la log-vraisemblance :

$$\ell(\beta) = \sum_{i=1}^{n} \left[Y_i \log P(\mathbf{X}_i; \boldsymbol{\beta}) + (1 - Y_i) \log \left(1 - P(\mathbf{X}_i; \boldsymbol{\beta})\right)\right].$$

L'estimation des paramètres $\boldsymbol{\beta}$ s'effectue en maximisant la log-vraisemblance. Cette maximisation est généralement réalisée par des méthodes numériques, telles que l'algorithme de Newton-Raphson, en résolvant l'équation suivante :

$$\nabla \ell(\boldsymbol{\beta}) = 0.$$

La solution $\hat{\boldsymbol{\beta}}$ correspond à l'estimateur du maximum de vraisemblancede de $\boldsymbol{\beta}$.

1.2.2 Algorithme de Newton-Raphson pour l'estimation des paramètres

L'algorithme de Newton-Raphson est une méthode itérative utilisée pour maximiser la log-vraisemblance $\log L(\boldsymbol{\beta})$. Il repose sur une approximation locale de la fonction à maximiser à l'aide d'un développement de Taylor au second ordre. À chaque itération, les coefficients $\boldsymbol{\beta}$ sont mis à jour en utilisant la relation suivante :

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - H^{-1}(\boldsymbol{\beta}^{(t)}) \times g(\boldsymbol{\beta}^{(t)}),$$

où

— $\boldsymbol{\beta}^{(t)}$ désigne le vecteur des coefficients à l'itération t.

— $g(\boldsymbol{\beta}) = \frac{\partial \ln L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$ est le gradient (vecteur des dérivées premières de la log-vraisemblance).

— $H(\boldsymbol{\beta}) = \frac{\partial^2 \ln L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}$ est la matrice Hessienne (matrice des dérivées secondes de la log-vraisemblance).

— $H^{-1}(\boldsymbol{\beta})$ est l'inverse de la matrice Hessienne.

Dans le cas de la régression logistique, ces quantities s'expriment comme suit :

$$g(\boldsymbol{\beta}) = \sum_{i=1}^{n} (Y_i - P(\mathbf{X}_i; \boldsymbol{\beta}))\mathbf{X}_i \quad \text{et} \quad H(\boldsymbol{\beta}) = -\sum_{i=1}^{n} P(\mathbf{X}_i; \boldsymbol{\beta})(1 - P(\mathbf{X}_i; \boldsymbol{\beta}))\mathbf{X}_i \mathbf{X}_i^T,$$

avec $P(\mathbf{X}_i; \boldsymbol{\beta}) = \frac{1}{1+\exp(-\mathbf{X}_i^T \boldsymbol{\beta})}$.

Remarque 6. *En régression probit, les expressions du gradient et de la Hessienne diffèrent en raison de l'utilisation de la fonction de répartition de la loi normale standard Φ et de sa densité ϕ :*

$$g(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{Y_i - \Phi(\mathbf{X}_i^T \boldsymbol{\beta})}{\phi(\mathbf{X}_i^T \boldsymbol{\beta})} \mathbf{X}_i \quad \textit{et} \quad H(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \frac{\Phi(\mathbf{X}_i^T \boldsymbol{\beta})(1 - \Phi(\mathbf{X}_i^T \boldsymbol{\beta}))}{\phi(\mathbf{X}_i^T \boldsymbol{\beta})^2} \mathbf{X}_i^T.$$

Étapes de l'algorithme

1. Initialiser $\boldsymbol{\beta}^{(0)}$.
2. Calculer $g(\boldsymbol{\beta}^{(t)})$ et $H(\boldsymbol{\beta}^{(t)})$.
3. Mettre à jour : $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - H^{-1}(\boldsymbol{\beta}^{(t)}) \cdot g(\boldsymbol{\beta}^{(t)})$.
4. Répéter jusqu'à convergence ($\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\| < \epsilon$), où $\|.\|$ est une norme choisie

1.3 Modèle de mélanges

1.3.1 Notations

Avant d'aborder les développements de cette section, nous introduisons les notations fondamentales qui seront utilisées tout au long de cette section. Soit $\mathbf{X} \in \mathbb{R}^p$ le vecteur

aléatoire représentant les observations associées à un individu donné et soit $Y \in \{1, \ldots, K\}$ la variable aléatoire indiquant la classe à laquelle appartient cet individu. La densité conditionnelle de $\mathbf{X}$, sachant que l'individu appartient à la classe k, est notée f_k et s'écrit formellement :

$$\mathbf{X} \mid Y = k \sim f_k.$$

On suppose que chaque individu appartient exclusivement à une seule classe. Nous définissons également p_k comme la probabilité a priori qu'un individu appartienne à la classe k, soit $\mathbb{P}(Y = k) = p_k$, avec la contrainte $\sum_k p_k = 1$. La loi jointe du couple $(\mathbf{X}, Y)$ sachant que l'individu est dans la classe k est donnée par $p_k f_k$, tandis que la loi marginale de $\mathbf{X}$, notée $f_{\mathbf{X}}$, s'écrit :

$$f_{\mathbf{X}}(\boldsymbol{x}) = \sum_{k=1}^{K} p_k f_k(\boldsymbol{x}).$$

Nous définissons ensuite la probabilité conditionnelle qu'un individu appartienne à la classe C_k, étant donné ses caractéristiques $\mathbf{X} = \boldsymbol{x}$, de la manière suivante :

$$t_k(\boldsymbol{x}) = \mathbb{P}(Y = k \mid \mathbf{X} = \boldsymbol{x}) = \frac{p_k f_k(\boldsymbol{x})}{f_{\mathbf{X}}(\boldsymbol{x})}, \ \mathbf{x} = (x_1, \ldots, x_p). \tag{1.1}$$

Dans la suite, nous supposons que la densité f_k suit une distribution normale multivariée de dimension p, notée $N_p(\mu_k, \Sigma_k)$, soit :

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \mu_k)^t \Sigma_k^{-1}(\boldsymbol{x} - \mu_k)\right).$$

Chaque classe est ainsi caractérisée par une moyenne μ_k et une matrice de covariance Σ_k. Nous notons par $\boldsymbol{\theta} = (p_1, \ldots, p_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$ l'ensemble des paramètres à estimer.

Définition 7. *On parle d'homoscédasticité lorsque* $\Sigma_1 = \ldots = \Sigma_K$. *Dans le cas contraire, on parle d'hétéroscédasticité du modèle.*

1.3.2 Règle de classement

Nous considérons dans un premier temps le cas où le vecteur des paramètres, $\boldsymbol{\theta}$, est supposé connu. Dans ce cadre, nous pouvons définir une règle théorique de classement permettant d'attribuer chaque observation à l'une des classes considérées.

Afin de simplifier les notations, nous nous intéressons d'abord au cas particulier où $K = 2$; deux classes ($Y = 1, 2$), sachant que les résultats obtenus peuvent être généralisés sans difficulté au cas général $K > 2$;

Lorsque l'on distingue deux groupes, la règle optimale de classement de Bayes, notée r^*, est définie par la relation suivante :

$r^*(\boldsymbol{x}) = 1$ (l'individu est affecté au groupe des Y=1) si et seulement si $t_1(\boldsymbol{x}) > t_2(\boldsymbol{x})$,

ce qui est équivalent à

$$\ln\left(\frac{t_1(\boldsymbol{x})}{t_2(\boldsymbol{x})}\right) > 0.$$

En posons $G_{\boldsymbol{\theta}}(\boldsymbol{x}) := \ln\left(\frac{t_1(\boldsymbol{x})}{t_2(\boldsymbol{x})}\right)$, nous pouvons alors définir l'équation de la surface discriminante par la condition : $G_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$. En utilisant la définition de t_k donnée par (1.1), nous obtenons la formulation suivante :

$$G_{\boldsymbol{\theta}}(\boldsymbol{x}) = \ln\frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} + \ln\frac{p_1}{p_2}.$$

En substituant les expressions exactes des densités conditionnelles, f_1 et f_2, et en notant que le second terme de l'équation ne dépend pas de $\boldsymbol{x}$, l'équation de la surface discriminante s'écrit, avec $\boldsymbol{\theta} = (\Sigma_1, \Sigma_2, \mu_1, \mu_2, p_1, p_2)$:

$$G_{\boldsymbol{\theta}}(x) = 0 \iff \ln\frac{|\Sigma_2|}{|\Sigma_1|} - (\boldsymbol{x} - \mu_1)^t\Sigma_1^{-1}(\boldsymbol{x} - \mu_1) + (\boldsymbol{x} - \mu_2)^t\Sigma_2^{-1}(\boldsymbol{x} - \mu_2) + 2\ln\frac{p_1}{p_2} = 0.$$

Lorsque les matrices de covariance des classes sont égales, soit $\Sigma_1 = \Sigma_2 = \Sigma$, la fonction discriminante se simplifie et prend une forme linéaire,

$$G_\theta(x) = (\mu_1 - \mu_2)^T\Sigma^{-1}\left(\boldsymbol{x} - \frac{\mu_1 + \mu_2}{2}\right) + \ln\frac{p_1}{p_2} = 0.$$

Cette équation indique que la frontière de décision entre deux classes est un hyperplan dans l'espace des variables explicatives. La fonction G_θ est également connue sous le nom de fonction score, car elle quantifie l'éloignement d'une observation par rapport à la frontière de classification. La figure 1.2 La figure illustre un modèle de mélange Gaussien pour deux groupes.

La probabilité a posteriori d'appartenance à une classe donnée peut être calculée à partir de la fonction score. Pour une observation donnée $\boldsymbol{x}$, la probabilité d'appartenir à la classe 1 est donnée par la relation suivante :

$$t_1(\boldsymbol{x}) = \mathbb{P}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{\exp(G_\theta(\boldsymbol{x}))}{1 + \exp(G_\theta(\boldsymbol{x}))}.$$

Taux d'erreur théorique et règle de classification Dans le cadre d'un modèle homoscédastique à deux classes, la règle de décision en classification repose directement sur la valeur de la fonction score G_θ. La règle de classement est définie comme suit :

$$r_\theta(\boldsymbol{x}) = \begin{cases} 1 & \text{si } G_\theta(\boldsymbol{x}) > 0, \\ 2 & \text{si } G_\theta(\boldsymbol{x}) < 0. \end{cases}$$

Lorsque les matrices de covariance des classes sont identiques, la frontière de classification prend une forme linéaire, simplifiant ainsi le modèle discriminant quadratique en un modèle discriminant linéaire. Cette simplification facilite l'interprétation des résultats et réduit la complexité des calculs. La fonction score G_θ joue un rôle crucial dans l'évaluation de cette frontière. En particulier, dans les problèmes de classification multi-classes, la généralisation à travers les log-ratios des fonctions de densité permet de préserver la cohérence du modèle discriminant tout en élargissant son domaine d'application à un nombre quelconque de groupes. Par ailleurs, la probabilité a posteriori, dérivée de la fonction score G_θ, constitue un critère probabiliste permettant de déterminer l'appartenance d'un individu à une classe donnée. Enfin, dans le cadre d'une classification binaire le taux d'erreur théorique est rigoureusement défini, facilitant ainsi la prise de décision.

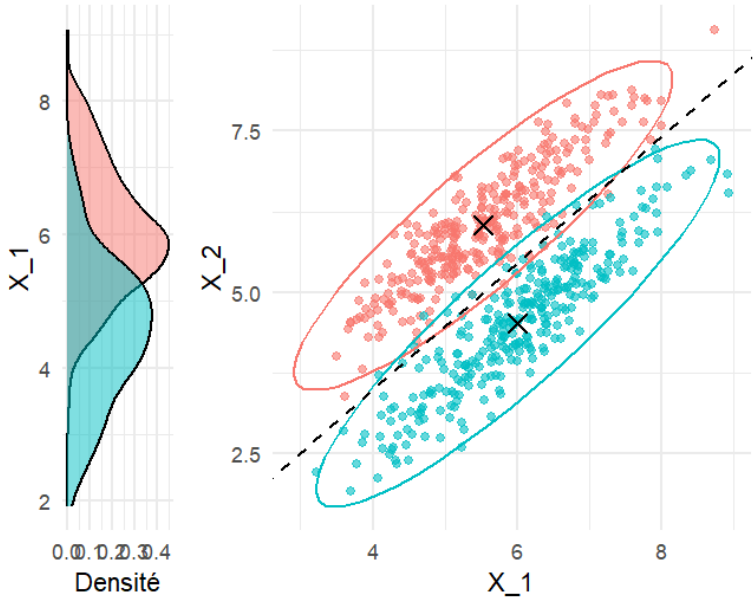

FIGURE 1.2 – La figure illustre un modèle de mélange Gaussien pour deux groupes. L'axe optimal minimise la variance intra-groupe et maximise la variance inter-groupe projetée, mais les distributions se recouvrent après projection. La frontière de classement, représentée par des tirets noirs, réduit ce recouvrement pour minimiser les erreurs de classification.

Remarque 8. *Dans un problème de classification avec plus de deux groupes, il est possible d'étendre cette approche en calculant les log-ratios des densités de probabilité des classes. La fonction score généralisée est alors définie comme :*

$$G_{k\ell}(\boldsymbol{x}) = \ln\left(\frac{t_k(\boldsymbol{x})}{t_\ell(\boldsymbol{x})}\right).$$

1.4 Méthodes des k-plus proches voisins

1.4.1 Principe général

La méthode des k-plus proches voisins *(k-Nearest Neighbors, k-NN)* est une technique non paramétrique utilisée pour estimer, pour chaque individu $\boldsymbol{x}$, les probabilités conditionnelles d'appartenance aux différents classe $C_1, \ldots, C_K$, notées $t_1(\boldsymbol{x}), \ldots, t_K(\boldsymbol{x})$.

Le principe fondamental de cette méthode repose sur l'analyse des k-plus proches voisins de $\boldsymbol{x}$ dans l'espace des variables explicatives (sous-espace de $\mathbb{R}^p$). Pour estimer la probabilité conditionnelle $t_\ell(\boldsymbol{x})$, on détermine la proportion des k voisins appartenant à la classe C_ℓ. Considérons un échantillon de n paires d'observations $(\mathbf{X}_i, Y_i)$, où $i = 1, \ldots, n$, avec $Y_i \in \{1, \ldots, K\}$ représente la classe de l'individu i et $\mathbf{X}_i \in \mathbb{R}^p$ est son vecteur de caractéristiques. Pour un nouveau point $\mathbf{X}$, dont nous observons la réalisation $\mathbf{X}$, l'estimation de la probabilité conditionnelle $t_\ell(\mathbf{X})$ est alors donnée par :

$$\hat{t}_\ell(\boldsymbol{x}) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}_{\{Y_i = \ell\}},$$

où Y_i est l'étiquette du i-ème voisin de $\boldsymbol{x}$. D'après la règle de décision de Bayes, on attribue à $\boldsymbol{x}$ la classe C_ℓ pour laquelle $\hat{t}_\ell(\boldsymbol{x})$ est maximale. Cette règle correspond à un vote majoritaire, où l'individu $\boldsymbol{x}$ est classé en fonction de la majorité des voisins dans son voisinage.

La règle de décision de Bayes est une stratégie de classification qui vise à minimiser la probabilité d'erreur moyenne en attribuant à une observation $\boldsymbol{x}$ la classe ω_i ayant la plus grande probabilité a posteriori $P(\omega_i|\boldsymbol{x})$. Mathématiquement, cette règle s'écrit :

$$\omega^*(\boldsymbol{x}) = \arg\max_{\omega_i} P(\omega_i|\boldsymbol{x}),$$

où $\omega^*(\boldsymbol{x})$ désigne la classe prédite pour l'observation $\boldsymbol{x}$. Cette règle repose sur le principe que l'on doit affecter chaque individu à la classe la plus probable compte tenu des données

observées. Elle constitue la référence optimale en termes de minimisation du taux d'erreur de classification, tel que décrit dans le chapitre 2 de [DHS01].

1.4.2 Choix des k plus proches voisins

L'un des aspects fondamentaux de l'application de la méthode des k-plus proches voisins concerne la détermination des points voisins et la sélection optimale du paramètre k. Pour cela, il est nécessaire de définir une mesure de distance dans l'espace des variables $\mathbb{R}^p$. Plusieurs métriques sont couramment utilisées pour évaluer la proximité entre deux points $\boldsymbol{x} = (x_1, \ldots, x_p)$ et $\boldsymbol{x}^* = (x_1^*, \ldots, x_p^*)$ dans un espace à p dimensions :

— **Distance Euclidienne** :

$$d(\boldsymbol{x}, \boldsymbol{x}^*) = \sqrt{(x_1^* - x_1)^2 + \cdots + (x_p^* - x_p)^2} = \left(\sum_{i=1}^{p} (x_i^* - x_i)^2 \right)^{1/2}.$$

— **Distance de Minkowski (paramètre q)** :

$$d(\boldsymbol{x}, \boldsymbol{x}^*) = \left(\sum_{i=1}^{p} |x_i^* - x_i|^q \right)^{1/q}.$$

— **Distance Chebyshev** :

$$d(\boldsymbol{x}, \boldsymbol{x}^*) = \max_{i \in \{1,\ldots,p\}} |x_i - x_i^*|.$$

— **Distance de Manhattan** :

$$d(\boldsymbol{x}, \boldsymbol{x}^*) = \sum_{i=1}^{p} |x_i - x_i^*|.$$

— **Distance de Canberra** :

$$d(\boldsymbol{x}, \boldsymbol{x}^*) = \sum_{i=1}^{p} \frac{|x_i - x_i^*|}{|x_i| + |x_i^*|}.$$

Ces différentes distances permettent de quantifier la similarité ou la dissimilarité entre les points de données dans l'espace $\mathbb{R}^p$, influençant ainsi directement l'identification des k voisins les plus proches d'un point donné $\boldsymbol{x}$.

1.4.3 Voisinage et choix de la distance

Pour définir le voisinage d'un point $\boldsymbol{x}$ après avoir muni l'espace d'une distance, on considère l'ensemble des points situés à une distance inférieure ou égale à une valeur seuil $d_{\max}$ de $\boldsymbol{x}$. Si l'on fixe un nombre de voisins k, alors la distance $d_{\max}$ correspond à la distance entre $\boldsymbol{x}$ et son k-ème plus proche. Une fois ces k-plus proches voisins identifiés, le point $\boldsymbol{x}$ est assigné à la classe majoritaire parmi eux, conformément à la règle du vote majoritaire utilisée dans l'algorithme des k-plus proches voisins. Une illustration de cet algorithme est présentée dans les figures 1.3 et 1.4 mettant en évidence la manière dont le voisinage est déterminé et comment la classification est réalisée à partir des données disponibles.

Le choix de la distance est crucial car il peut fortement influencer les résultats, en particulier lorsque les variables explicatives possèdent des unités de mesure différentes. Prenons l'exemple de deux variables : la fréquence cardiaque (exprimée en battements par minute) et le taux de cholestérol (mesure en g/L). Sans normalisation, la fréquence cardiaque, typiquement autour de 70 (bat/min), dominerait la distance par rapport au cholestérol, qui varie entre 1.5 et 2.5 (g/L). Pour éviter ce déséquilibre et garantir une influence équitable de chaque variable sur la mesure de la distance, il est recommandé de standardiser les données en divisant chaque variable par son écart-type.

1.5 Méthode de segmentation CART

La méthode de segmentation **CART** *(Classification and Regression Trees)* est une approche puissante pour la classification supervisée ainsi que pour la régression multiple. Introduite par Breiman et al. en 1984 [BFOS84], cette technique repose sur la construction d'arbres de décision, établi à partir de critères de division optimaux et des règles d'arrêt spécifiques.

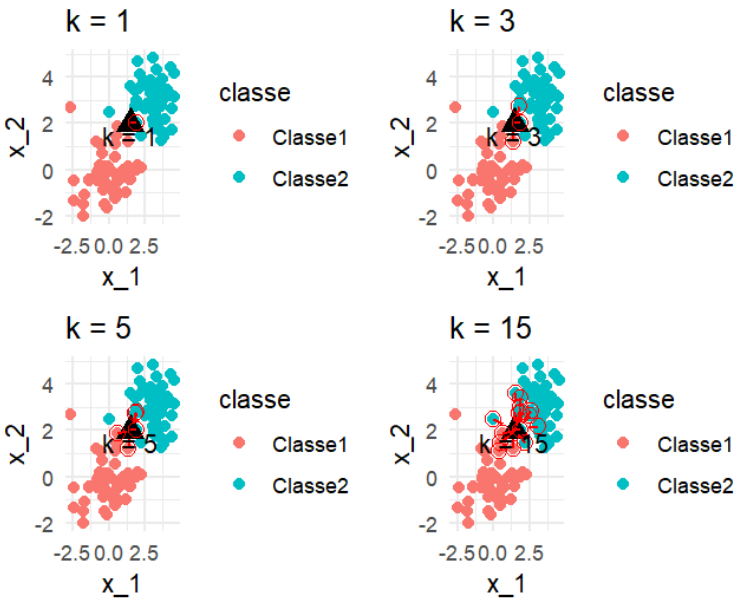

FIGURE 1.3 – Illustration de l'algorithme pour classifier un nouveau point (en noir) selon le nombre de voisins sélectionnés.

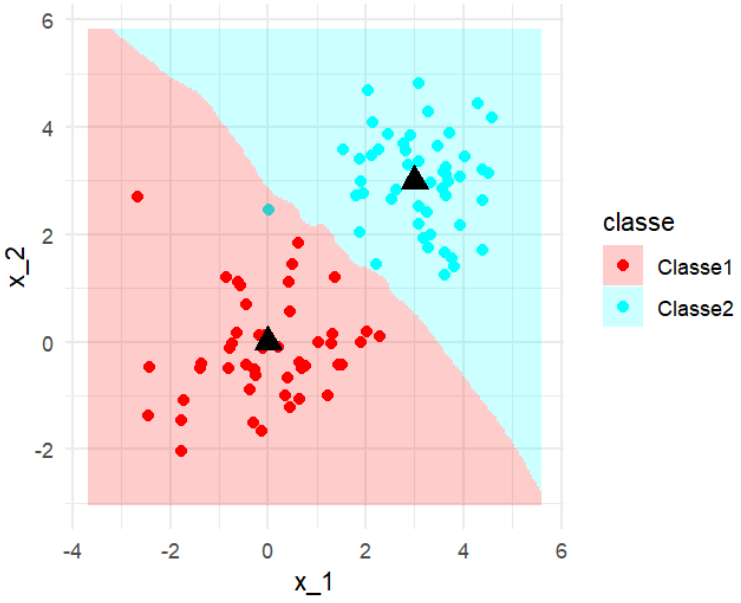

FIGURE 1.4 – Frontière de décision pour $k = 5$.

1.5.1 Principes généraux

La méthode des arbres de décision binaires repose sur une partition récursive de l'espace des variables explicatives, aboutissant à une segmentation en sous-espaces, où la variable cible Y est supposée homogène. La figure 1.5 illustre le processus de classification d'un nouvel individu dont les valeurs des variables explicatives sont $(x_1, x_2) = (0.50, 0.75)$. Pour cela, on procède de manière itérative en descendant progressivement le long des branches de l'arbre, mettant en évidence le chemin de décision emprunté. La trajectoire suivie est représentée par un trait plus épais dans l'arbre. Le processus commence au premier nœud, qui concerne la variable X_2, ayant ici la valeur 0.75. Étant donné que cette valeur est supérieure à 0.35, la règle de décision impose de suivre la branche droite de l'arbre. À l'étape suivante, sur cette branche droite, l'exploration de l'arbre se poursuit avec un deuxième nœud, qui concerne la variable X_1, dont la valeur est 0.50. Étant donné que cette valeur est inférieure ou égale à 0.65, la règle de décision impose de suivre la branche gauche. Enfin, un dernier nœud est rencontré, portant de nouveau sur la variable X_2. Sa valeur observée, 0.75, étant elle est supérieure à 0.62, le chemin suivi conduit vers la branche droite. Ainsi après avoir parcouru l'ensemble des divisions successives, le nouvel individu est alors classé dans le groupe 1.

Définition 9. *Nous définissons quelques notions utiles pour la méthode CART :*

- *Un arbre est un graphe orienté sans cycle, composé d'un ensemble de nœuds N et d'arêtes E.*
- *Un arbre enraciné est un arbre possédant une racine unique, et chaque nœud (sauf la racine) a un parent unique.*
- *Une feuille est un nœud sans successeurs. L'ensemble des feuilles est noté F.*
- *Une feuille est dite pure si tout ses éléments appartiennent à la même classe.*

Pour expliquer l'algorithme en détail, nous commençons par quelques définitions :

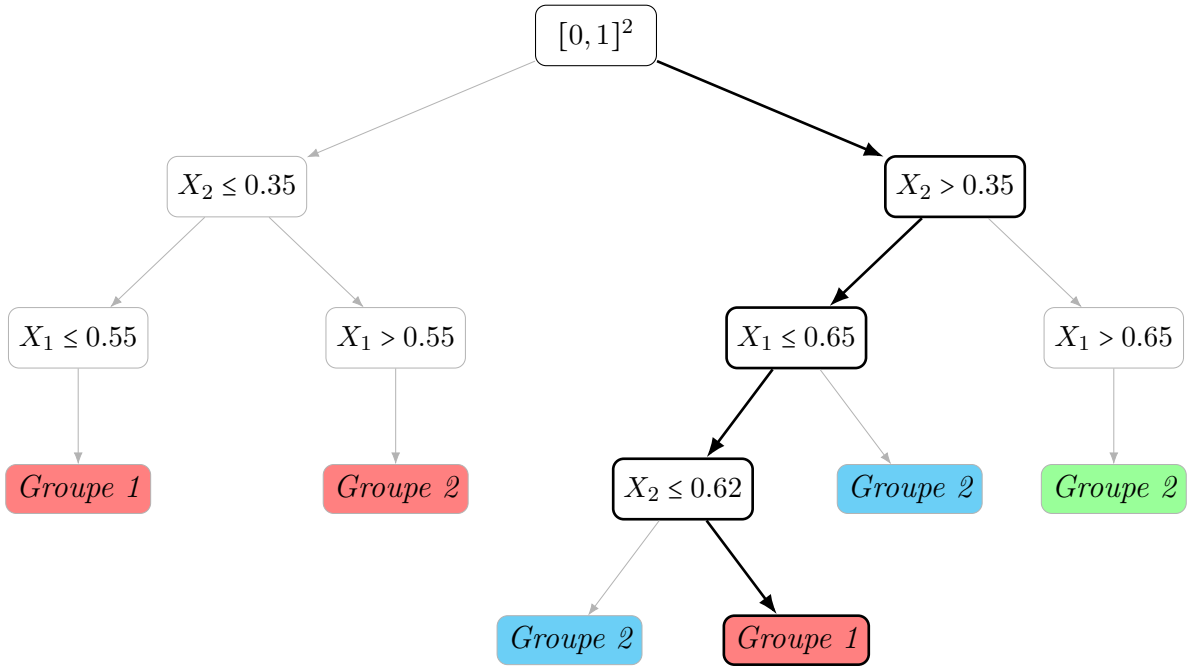

FIGURE 1.5 – Illustration d'arbre de décision.

1.5.2 Construction de l'arbre maximal

Nous nous intéressons dans cette section à la construction de l'arbre maximal, qui repose sur les trois étapes suivantes :

1. divisions successives des nœuds,
2. choix d'un critère d'arrêt de la division,
3. affectation des individus de chaque feuille à un groupe.

1.5.3 Division des nœuds

Comme l'illustre l'exemple simple présenté dans la figure 1.5 , il existe plusieurs manières de diviser un nœud à chaque itération de l'algorithme. Une question essentielle se pose

alors : quelle est la meilleure division à effectuer, en fonction de l'objectif final de classification ou de régression ? Afin d'y répondre, il convient d'introduire quelques notions fondamentales.

L'algorithme CART se déroule en deux phases distinctes. Dans la première phase, l'algorithme construit un arbre maximal, note $T_{\max}$, dans lequel toutes les feuilles sont pures. Cependant, lorsque les données sont très fragmentées, les feuilles obtenues peuvent être de taille très petite, ou même réduites à un seul élément, ce qui augmente le risque de sur apprentissage. La deuxième phase de l'algorithme consiste à *élaguer* l'arbre, c'est-à-dire à supprimer certaines branches en regroupant les feuilles (ou nœuds terminaux).

Définition 10. *Une division est dite admissible si aucun des nœuds successeurs n'est vide.*

En pratique, le nombre de divisions admissibles pour une variable explicative X dépend de sa nature :

- Si X est une variable qualitative ordinale comportant m modalités, il est possible de définir $m-1$ points de séparation, correspondant aux différentes frontières entre les modalités ordonnées.
- Lorsque X est une variable qualitative nominale comportant m modalités distinctes, le nombre de partitions admissibles est donné par $2^{m-1}-1$.
- Si X est une variable quantitative, le nombre théorique de divisions possibles est infini. Cependant, en pratique, on observe un nombre fini de réalisations de X et on assimile cette variable à une variable qualitative ordinale.

En pratique, les algorithmes de segmentation ont tendance à favoriser les variables avec un grand nombre de modalités, car celle-ci permettent un plus grand nombre de divisions possibles et augmentent ainsi leur probabilité d'être sélectionnées au cours du processus de partitionnement.

Dans ce contexte, on définit la probabilité conditionnelle $p_k(a) = \mathbf{P}(Y = k \mid a)$; la probabilité d'être dans la classe C_k sachant que l'on se trouve dans le nœud a. Cette probabilité est estimée empiriquement, pour tout $k \in \{1, \ldots, K\}$, par :

$$\hat{p}_k(a) = \frac{n_{k,a}}{n_a}$$

où $n_{k,a}$ représente le nombre d'observations du nœud a appartenant à la classe C_k et n_a correspond au nombre total d'observations contenues dans le nœud a.

1.5.4 Impureté d'un nœud

Nous introduisons à présent la notion de l'impureté d'un nœud, qui permet de mesurer l'hétérogénéité d'un nœud en fonction de la distribution de la variable Y. Cette mesure sera utile dans le critère de division d'un nœud qu'on introduira dans la sous section 1.5.5.

Définition 11. *L'impureté d'un nœud, a, est déterminée par une fonction i, définie sur l'ensemble des nœuds N de l'arbre de décision et à valeurs dans $\mathbb{R}$, telle que :*

$$i(a) = f(p_1(a), \ldots, p_K(a)),$$

où f est une fonction respectant les conditions suivantes :

- *f est une fonction symétrique par rapport aux probabilités conditionnelles $p_k(a)$ (i.e., invariante par permutation des variables $p_1(a), \ldots, p_K(a)$).*
- *f atteint son maximum lorsque $p_k(a) = \frac{1}{K}$, pour $k \in \{1, \ldots, K\}$.*
- *f atteint son minimum s'il existe un indice k tel que $p_k(a) = 1$.*

Les deux fonctions d'impureté les plus utilisées sont (voir figure 1.6) :

— **L'entropie de Shannon** :

$$i(a) = n_a \sum_{k=1}^{K} p_k(a) \ln p_k(a).$$

— **L'indice de Gini** :

$$i(a) = n_a \sum_{k=1}^{K} p_k(a) \left(1 - p_k(a)\right).$$

1.5.5 Critère de division d'un nœud

En s'appuyant sur les définitions précédentes, nous pouvons établir le critère de division suivant : parmi toutes les divisions admissibles du nœud a, nous sélectionnons celle qui maximise la réduction de l'impureté. Soit d une division du nœud a en deux nœuds descendants, $a_{G,d}$ (nœud de gauche) et $a_{D,d}$ (nœud de droite). La réduction de l'impureté du nœud a par la division d est définie par :

$$\Delta i(d, a) = i(a) - \left(i(a_{G,d}) + i(a_{D,d})\right).$$

Nous retenons alors, parmi toutes les divisions admissibles D du nœud a, celle qui maximise la réduction de l'impureté $\Delta i(d, a)$. Dans la pratique, comme les $p_k(a)$ sont souvent inconnus, nous utilisons une estimation de l'impureté :

$$d^* = \arg\max_{d \in D} \Delta\hat{i}(d, a) \quad \text{où} \quad \hat{i}(d, a) = f(\hat{p}_1(a), \ldots, \hat{p}_K(a)).$$

1.5.6 Critère d'arrêt

L'algorithme de la segmentation s'arrête lorsque tous les nœuds sont purs ou lorsqu'aucune division admissible ne peut être réalisée pour un nœud donné. Ce dernier cas se produit lorsque les observations de ce nœud présentent des valeurs identiques pour toutes

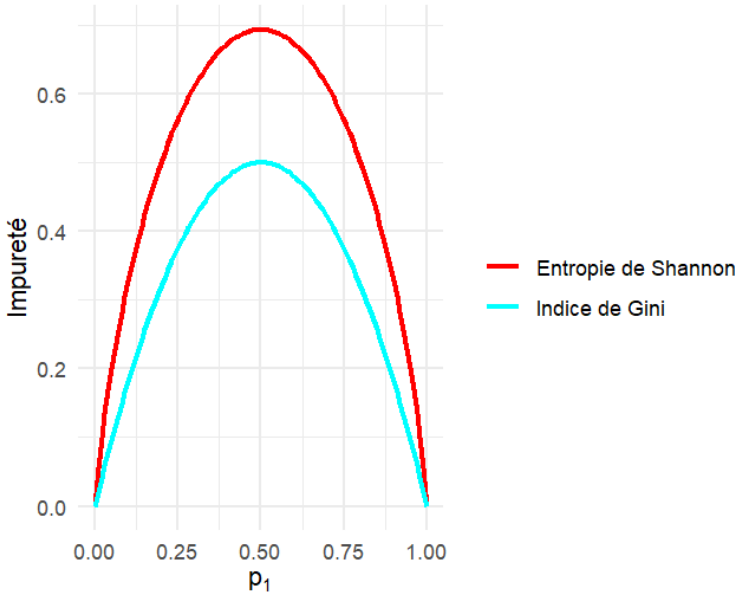

FIGURE 1.6 – Les courbes de l'entropie de Shannon et de l'indice de Gini en fonction de p_1

les variables explicatives. En pratique, il est également possible d'arrêter la division d'un nœud lorsque le nombre d'observations qu'il contient est trop faible (par exemple, inférieur à 5).

1.5.7 Affectation des individus

L'arbre maximal permet de segmenter l'espace des variables explicatives en sous-ensembles distincts, définis par les nœuds de l'arbre. Pour un individu donné, caractérisé par un vecteur de variables explicative $\boldsymbol{x}$, le nœud auquel il appartient est noté $a(\boldsymbol{x})$.

Une fois le nœud correspondant identifié, l'individu est affecté à une des classes $C_1, \ldots, C_K$. Si le nœud a est pur, l'individu est assigné à la classe unique représentée dans ce nœud. Si le nœud a n'est pas pur, on applique la règle de Bayes, décrite en section 1.4. Dans ce

cas l'individu est assigné à la classe majoritaire C_k dans le nœud. Cette règle de décision est formalisée par :

$$r(\boldsymbol{x}) = k \iff k = \arg\min_{\ell} p_\ell(a(\boldsymbol{x})).$$

En d'autres termes, l'algorithme CART génère une règle de décision constante par morceaux.

1.6 Méthodes d'ensemble : boosting, bagging et forêts aléatoires

Les méthodes d'ensemble constituent une famille d'approches puissantes en apprentissage supervisé, visant à améliorer la performance des modèles de classification et de régression. Elles se divisent principalement en deux grandes catégories :

Le *boosting*, qui repose sur une approche récursive et adaptative cherchant à renforcer une série de classificateurs faibles en accordant un poids plus important aux observations mal classées à chaque itération.

Le *bagging* (Bootstrap Aggregating), qui emploie plusieurs échantillons bootstrap pour entraîner une famille de classificateurs indépendants, puis agrège leurs prédictions afin de réduire la variance et d'améliorer la robustesse du modèle. Un cas particulier de *bagging* est celui des forêts aléatoires, qui appliquent cette stratégie aux arbres de décision CART.

Ces méthodes d'ensemble sont reconnues pour leur efficacité. En effet elles permettent d'obtenir de résultats supérieurs à ceux des classificateurs faibles pris individuellement et qui sont souvent instables et présentant une forte variance. En combinant les résultats de différents classificateurs, ces méthodes parviennent à réduire la variance des prédictions. En théorie, toute famille de classificateurs faibles peut être utilisée dans le cadre des méthodes d'ensemble. Toutefois, en pratique, les arbres de décision sont les plus couramment employés. Leur forte sensibilité aux variations des données d'apprentissage

fait d'elles des modèles instables, ce qui les rend particulièrement adaptés aux approches d'ensemble telles que le bagging et le boosting.

1.6.1 Méthodes de bagging

Les méthodes de *bagging* (*Bootstrap Aggregation*), introduites par Leo Breiman en 1996 [Bre96], constituent une approche de classification supervisée reposant sur l'agrégation d'un ensemble de B estimateurs $\hat{b}_1(\boldsymbol{x}), \ldots, \hat{b}_B(\boldsymbol{x})$. Chaque $\hat{b}_k(\boldsymbol{x})$ est la prédiction de l'observation $\boldsymbol{x}$ produite par le k-ème modèle, lequel est formé à partir d'un échantillon bootstrap de l'ensemble de données d'entraînement. Ces prédictions individuelles sont ensuite agrégées selon la formule suivante :

$$\hat{b}(\boldsymbol{x}) = \frac{1}{B} \sum_{k=1}^{B} \hat{b}_k(\boldsymbol{x}).$$

Ainsi, $\hat{b}_k(\boldsymbol{x})$ est la prédiction fournie par le k-ème classifieur du modèle de bagging sur l'observation $\boldsymbol{x}$.

Dans le cas d'un problème de classification, la prédiction finale est obtenue par vote majoritaire entre les différentes prédictions $\hat{b}_k(\boldsymbol{x})$, en retenant la classe ayant obtenu le plus de votes.

Pour illustrer l'efficacité de la méthode de bagging, nous l'appliquons sur un modèle de régression. Soit $(\mathbf{X}, Y) \in (\mathbb{R}^p, \mathbb{R})$ un vecteur aléatoire, et soit $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ un échantillon i.i.d. suivant la même loi que $(\mathbf{X}, Y)$. La fonction de régression est définie par :

$$m(\boldsymbol{x}) = \mathrm{E}[Y|\mathbf{X} = \boldsymbol{x}].$$

Pour un point fixe $\boldsymbol{x} \in \mathbb{R}^p$, on considère l'erreur quadratique moyenne (EQM) d'un estimateur $\hat{b}$, qui s'exprime à travers la décomposition biais-variance :

$$\mathrm{E}[(\hat{b}(\boldsymbol{x}) - m(\boldsymbol{x}))^2] = (\mathrm{E}[\hat{b}(\boldsymbol{x})] - m(\boldsymbol{x}))^2 + \mathrm{Var}(\hat{b}(\boldsymbol{x})).$$

Sous l'hypothèse que les estimateurs $\hat{b}_1(\boldsymbol{x}), \ldots, \hat{b}_B(\boldsymbol{x})$ sont indépendants et identiquement distribués (i.i.d.), on obtient les relations suivantes :

$$\mathrm{E}[\hat{b}(\boldsymbol{x})] = \mathrm{E}[\hat{b}_1(\boldsymbol{x})] \quad \text{et} \quad \mathrm{Var}(\hat{b}(\boldsymbol{x})) = \frac{1}{B}\mathrm{Var}(\hat{b}_1(\boldsymbol{x})).$$

Ainsi, le biais de l'estimateur agrégé reste identique à celui des $\hat{b}_k$, tandis que la variance diminue d'un facteur $\frac{1}{B}$. Cependant, en pratique, il est difficile de garantir l'indépendance parfaite des estimateurs $\hat{b}_k$, puisque ces derniers sont tous construits à partir du même échantillon D_n.

L'algorithme

L'algorithme de bagging repose sur la construction de B estimateurs, $\hat{b}_k$, indépendants en utilisant des échantillons bootstrap issus des données d'origine, suivie de leur agrégation. À chaque itération k, un échantillon bootstrap, noté $\theta_k = \theta_k(D_n)$, est généré en réalisant des tirages aléatoires avec remise de l'échantillon initial D_n.

À partir de chaque échantillon θ_k, un estimateur $\hat{b}(., \theta_k)$ est construit. L'estimateur final est ensuite obtenu en prenant la moyenne des B estimateurs individuels, selon la formule suivante :

$$\hat{b}_B(\boldsymbol{x}) = \frac{1}{B}\sum_{k=1}^{B} \hat{b}(\boldsymbol{x}, \theta_k).$$

Les étapes suivantes décrivent la méthode de bagging.

Algorithm 1 Bagging

Entrées :

- $\boldsymbol{x}$: l'observation pour laquelle on souhaite effectuer une prédiction.
- Un modèle de base (ex. arbre CART, k-plus proches voisins).
- D_n : l'échantillon de données d'apprentissage.
- B : le nombre de modèles à entraîner et à agréger.

Étapes :

1. Pour $k = 1, \ldots, B$:
   (a) Tirer un échantillon bootstrap θ_k à partir de l'échantillon d'origine D_n.
   (b) Ajuster le modèle de base sur cet échantillon bootstrap pour obtenir un estimateur $\hat{b}_k(\boldsymbol{x})$.

Sortie : L'estimateur final est calculé en agrégeant les B modèles entraînés :

$$\hat{b}_B(\boldsymbol{x}) = \frac{1}{B} \sum_{k=1}^{B} \hat{b}_k(\boldsymbol{x}).$$

Tirage de l'échantillon bootstrap

Dans l'algorithme de *bagging*, la génération des échantillons bootstrap consiste à produire des sous-échantillons aléatoires à partir des données d'origine D_n. Deux méthodes principales sont employées :

- **Technique A :** Tirer n observations avec remise à partir de l'échantillon D_n, chaque observation ayant une probabilité $\frac{1}{n}$ d'être sélectionnée.
- **Technique B :** Tirer $\ell < n$ observations, avec ou sans remise, dans D_n.

Ces échantillons bootstrap sont utilisés pour construire des estimateurs individuels sur des sous-ensembles de données, puis les agréger. Le *bagging* améliore ainsi les performances des règles de décision "faibles", comme la règle du $k = 1$ plus proche voisin, en réduisant la variance de l'estimateur tout en maintenant son biais.

Théorème 1.2. *[BD10] Si la taille de l'échantillon bootstrap ℓ satisfait les conditions*

suivantes :

$\ell = \ell_n$ *tel que* $\lim_{n\to\infty} \ell_n = +\infty$ *et* $\lim_{n\to\infty} \frac{\ell_n}{n} = 0$*, alors l'estimateur agrègé* $\hat{b}(\boldsymbol{x})$ *est uniformément convergent.*

1.6.2 Les forêts aléatoires

Les forêts aléatoires constituent une méthode de *bagging* appliquée aux arbres de décision CART, et utilisée aussi bien pour la régression que pour la classification. Le concept principal est similaire à celui du *bagging* mentionné précédemment, mais avec une dimension aléatoire supplémentaire. En effet, pour chaque échantillon bootstrap, un sous-ensemble de taille m des variables explicatives sélectionné de manière aléatoire est utilisé pour la classification. L'algorithme détaillé des forets aléatoires est présenté dans l'algorithme 2 ci-dessous.

Algorithm 2 L'algorithme des forêts aléatoires

Entrées : Déterminer le nombre d'échantillons bootstrap B ainsi que le nombre de variables explicatives m à sélectionner à chaque itération.

Étapes :

1. Pour $b = 1, \ldots, B$:
   (a) Tirer un échantillon bootstrap D_b de taille n à partir de l'échantillon d'apprentissage initial D_n.
   (b) Sélectionner aléatoirement m variables explicatives parmi les p variables disponibles.
   (c) Construire un arbre de décision (de type CART), $b_{\max}$, en utilisant uniquement les m variables sélectionnées et les données de l'échantillon bootstrap D_b.

Sortie : Agréger les B arbres, $b_{\max}$, pour former la forêt F, où la prédiction finale est obtenue par un vote majoritaire pour les tâches de classification ou par la moyenne des prédictions pour les tâches de régression.

Erreur "out-of-bag"

L'un des principaux avantages des méthodes de *bagging* réside dans leur capacité à estimer l'erreur de classification sans nécessiter la séparation explicites des données en ensembles d'apprentissage et de test. En effet, lors de chaque tirage bootstrap b, un sous-ensemble d'observations est sélectionné parmi les données d'apprentissage. Considérons l'exemple des forêts aléatoires comme exemple (ceci est également valable pour toutes les méthodes de type *bagging*). Notons par F_b^i la forêt aléatoire composée des arbres construits à partir des échantillons bootstrap qui n'incluent pas l'observation i. Dans ce contexte, il est possible d'estimer la prédiction $\hat{y}_i^{\text{OOB}}$ de l'observation i en appliquant la règle du vote majoritaire sur les arbres de la foret F_b^i. Autrement dit, seuls les arbres qui n'ont pas vu l'observation i lors de l'apprentissage sont utilisés pour prédire sa classe.

L'erreur *out-of-bag* est alors définie par la formule suivante :

$$\hat{e}_{\text{OOB}}(F) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}(\hat{y_i}^{\text{OOB}} \neq y_i).$$

1.6.3 Méthodes de boosting

Le *boosting* a émergé à la fin des années 80 et au début des années 90, répondant à la question de savoir s'il était possible de créer un classificateur "puissant" à partir d'un ensemble de classificateurs faibles. Le premier algorithme de *boosting*, nommé *AdaBoost* (pour *Adaptive Boosting*), a été introduit quelques années plus tard par Freund et Schapire (1997) [FS97]. Il s'est rapidement devenu comme une référence importante en classification supervisée. C'est un champ de recherche extrêmement dynamique, avec de nombreuses variantes et nouveaux algorithmes régulièrement proposés.

L'algorithme AdaBoost

L'algorithme *AdaBoost*, conçu à l'origine pour la classification binaire, sera considéré dans ce contexte pour simplifier la présentation de l'algorithme. Considérons un échantillon d'apprentissage composé de n vecteurs aléatoires indépendants et identiquement distribués, $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$, avec $\mathbf{X}_i \in \mathbb{R}^p$ et $Y_i \in \{-1, 1\}$.

Dans une première étape, une règle de décision initiale est établie en attribuant des poids égaux à chaque observation. L'erreur de classification associée à cette règle est ensuite évaluée, et les poids des observations mal classées sont augmentés afin d'orienter la règle de décision suivante vers les cas les plus difficiles à prédire. Ainsi, au fil des itérations, les modèles successifs se concentrent davantage sur les observations mal classées. Le classificateur final résulte d'un vote majoritaire pondéré de l'ensemble des classificateurs générés au cours des itérations. L'algorithme 3 présente les étapes du processus AdaBoost. Les poids w_i des observations sont initialisés à $1/n$ lors de la première itération, puis ajustés dynamiquement à chaque étape. Si une observation est correctement classée, son poids reste inchangé. Si une observation est mal classée, son poids est augmenté proportionnellement à la performance du modèle, mesurée par un coefficient notée α_b. L'estimateur final est alors construit sous la forme d'une combinaison pondérée des règles successives $g_1, \ldots, g_B$, où chaque règle contribue au modèle global en fonction de sa capacité d'ajustement aux données.

1.7 Machines à vecteurs supports

La machine à vecteurs du support (appelée en anglais *Support Vector Machine, SVM*) est un algorithme d'apprentissage supervisé principalement utilisé pour la classification, bien qu'il puisse également être appliqué à des problèmes de régression. Développé dans

les années 90, il a rapidement gagné en popularité en raison de son efficacité et à sa capacité de produire des résultats précis avec des ressources de calcul limitées. Aujourd'hui encore, le SVM reste un outil incontournable, notamment dans des domaines tels que la reconnaissance de motifs, où il est apprécié pour sa capacité à traiter des données complexes avec une grande précision.

Algorithm 3 AdaBoost

Entrées :

- $\boldsymbol{x}$: l'observation pour laquelle on souhaite effectuer une prédiction.
- $D_n = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$: l'échantillon d'apprentissage, $y_i \in \{-1, 1\}$.
- Une règle faible (ex. 1-plus proches voisins).
- B : le nombre d'itérations.

Étapes :

1. Initialiser les poids : $w_i = \frac{1}{n}$, $i = 1, \ldots, n$.
2. Pour $b = 1, \ldots, B$:
   (a) Ajuster la règle faible sur l'échantillon pondéré par les poids w_i pour obtenir une règle $g_b(\boldsymbol{x})$.
   (b) Calculer le taux d'erreur :
   $$e_b = \frac{\sum_{i=1}^{n} w_i \, \mathbf{1}_{y_i \neq g_b(\boldsymbol{x}_i)}}{\sum_{i=1}^{n} w_i}.$$
   (c) Calculer la qualité d'ajustement :
   $$\alpha_b = \log\left(\frac{1 - e_b}{e_b}\right).$$
   (d) Mettre à jour les poids :
   $$w_i = w_i \exp\left(\alpha_b \, \mathbf{1}_{\{y_i \neq g_b(\boldsymbol{x}_i)\}}\right), \quad i = 1, \ldots, n.$$

Sortie : L'estimateur final est une combinaison pondérée des B règles faibles :

$$\hat{g}_B(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{b=1}^{B} \alpha_b g_b(\boldsymbol{x})\right), \text{ avec } \operatorname{sign}(z) = \begin{cases} +1 & \text{si } z \geq 0 \\ -1 & \text{si } z < 0. \end{cases}$$

1.7.1 Fonctionnement de l'algorithme SVM

L'algorithme SVM repose sur la construction d'un hyperplan permettant de séparer les données en plusieurs classes tout en maximisant la marge entre les points les plus proches

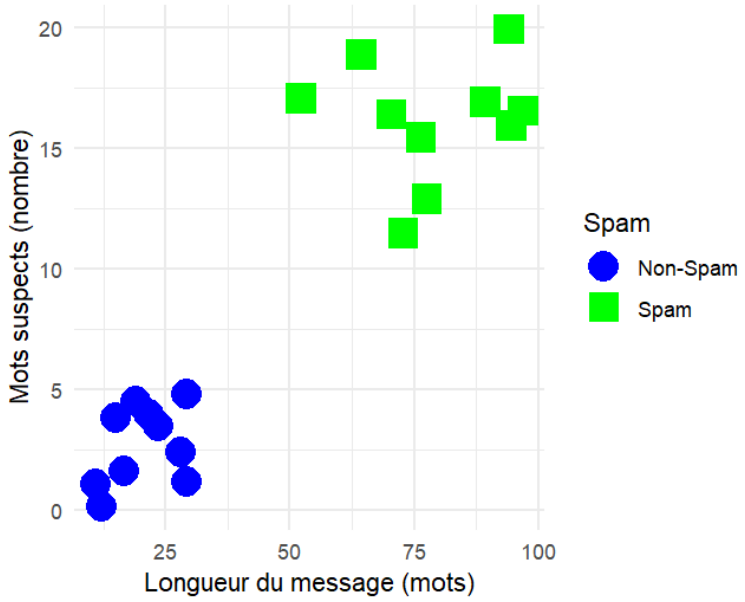

FIGURE 1.7 – Diagramme de dispersion de l'échantillon

de chaque classe. Prenons un exemple : on souhaite classer des emails comme "spam" ou "non spam" en utilisant des caractéristiques, par exemple, le nombre de mots clés suspects et la longueur du message. L'algorithme SVM établit une frontière de décision optimale permettant de séparer ces deux catégories, tout en minimisant les erreurs de classification. On suppose que les deux facteurs de différenciation identifiés sont : la longueur du message et le nombre de mots suspects. La figure 1.7 représente un diagramme de dispersion des observations. La figure 1.8 montre une classification binaire entre des emails de type Spam (en vert) et Non-Spam (en bleu). Les deux variables sur les axes sont la longueur du message (en mots) et le nombre de mots suspects.

- Non-Spam (en bleu) : Ces points correspondent à des emails dont la longueur est relativement courte et contiennent peu de mots suspects. Ils sont regroupés dans la partie inférieure à gauche du graphique.
- Spam (en vert) : Ces points correspondent à des emails ayant une longueur plus impor-

tante et un nombre élevé de mots suspects, indiquant des différences significatives par rapport aux non-spams.

Supposons qu'on a un email avec une longueur de 70 mots et contenant 15 mots suspects. Intuitivement, en observant ces caractéristiques, on l'associerait à la catégorie Spam, car ses caractéristiques sont proches des email classes comme spams dans l'échantillon.

Comment peut-on amener une machine à comprendre et appliquer ces principes sur de nouvelles données ? Grâce à un classificateur SVM, cela est possible à travers un hyperplan bien défini. Chaque donnée, $\boldsymbol{x}_i \in \mathbb{R}^p$ où p représente le nombre de caractéristiques, est projetée dans un espace de caractéristiques. L'hyperplan est défini par l'équation suivante :

$$\mathbf{w}^T \boldsymbol{x} + b = 0.$$

où $\mathbf{w}$ est le vecteur des poids et b est le biais. L'objectif est de maximiser la marge de séparation ce qui revient à minimiser la fonction suivante :

$$\min \frac{1}{2} \|\mathbf{w}\|^2.$$

sous la contrainte :

$$y_i(\mathbf{w}^T \boldsymbol{x}_i + b) \geq 1, \quad \text{pour} \quad i = 1, \ldots, n$$

où y_i est la classe associée à chaque donnée, avec $y_i = 1$ pour le spam et $y_i = -1$ pour le non-spam, $\boldsymbol{x}_i$ est le vecteur de caractéristiques pour chaque donnée d'entraînement i et n est le nombre total des données.

Ensuite, la classification est effectuée en trouvant l'hyperplan optimal qui sépare au mieux les deux classes. Dans cet exemple, plusieurs hyperplans sont envisageables, mais l'objectif est de sélectionner celui qui maximise la marge entre les points de chaque classe, garantissant ainsi une séparation optimale.

La figure 1.8 illustre cette approche. La frontière rouge située proche des points bleus

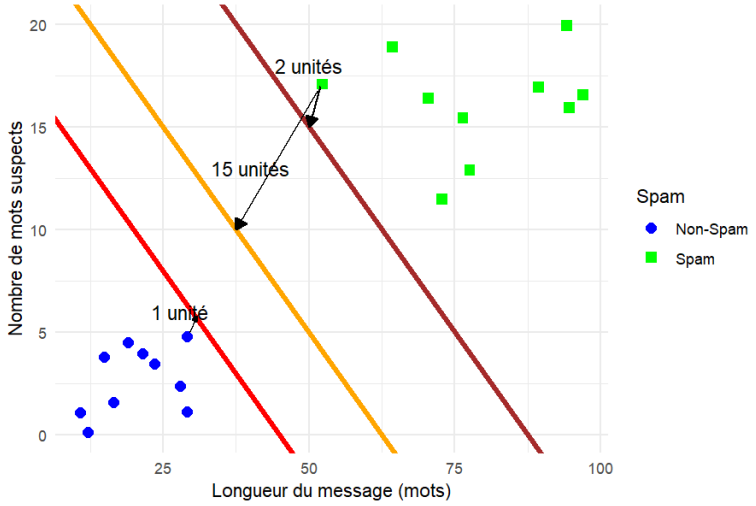

FIGURE 1.8 – Classification Spam/Non-Spam avec hyperplans ajustés et annotations

(non-spam) avec une distance minimale d'une seule unité. La frontière orange, maximise la marge en maintenant une distance minimale de 15 unités entre l'hyperplan et les points les plus proche. L'algorithme SVM sélectionne donc l'hyperplan orange, garantissant une séparation optimale entre les classes.

Cas particuliers

L'exemple précédent illustrait un cas où les données étaient facilement séparables à l'aide d'un hyperplan linéaire. Cependant, il arrive que les données soient dispersées de manière complexe, rendant leur séparation difficile, aussi bien visuellement que linéairement. Que faire lorsqu'il n'existe pas de frontière linéaire claire pour séparer les classes ?

Plusieurs scénarios peuvent se présenter. Il est important d'envisager différentes méthodes adaptées à des données plus complexes qui ne se traitent pas à une séparation linéaire évidente.

Première situation : considérons l'exemple illustré par la figure 1.9. Dans cette si-

tuation, une séparation linéaire stricte est impossible, car une observation aberrante, représentée par une étoile, se trouve dans la région correspondant à la classe des ronds. Une telle observation peut perturber la construction de l'hyperplan, rendant l'application d'un classifieur linéaire sous-optimale.

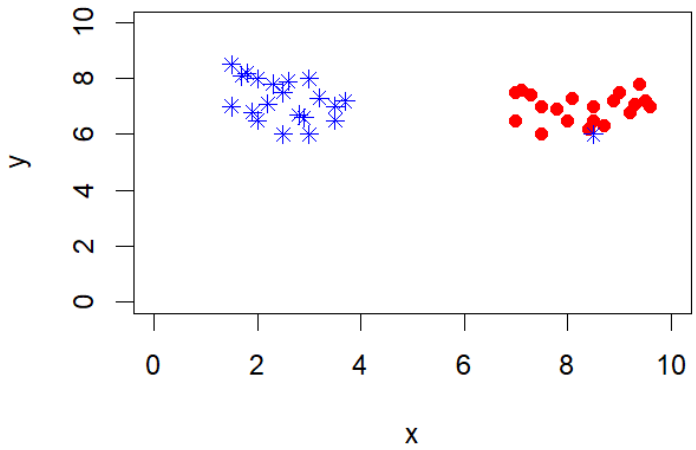

FIGURE 1.9 – Cas non-séparable 1.

Deuxième situation : Dans le cas présenté par la figure 1.10, il n'est pas possible de tracer un hyperplan linéaire entre les deux classes. Le SVM peut résoudre ce problème en introduisant une troisième dimension. Ici, nous allons ajouter une nouvelle dimension à partir des deux précédentes : $z = x^2 + y^2$. Ainsi, nous avons pris un espace initial de faible dimension et l'avons transformé en un espace de dimension supérieure où une séparation linéaire devient possible. À présent, projetons les points de données sur les axes x et z comme illustré dans figure 1.11.

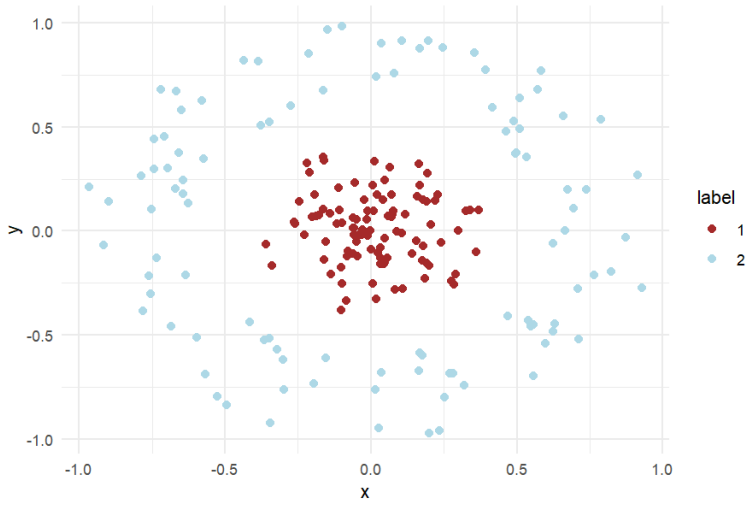

FIGURE 1.10 – Cas non-séparable 2.

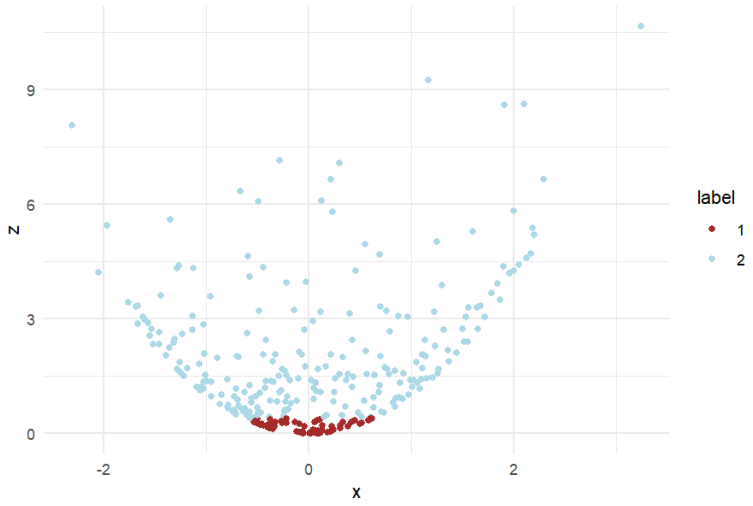

FIGURE 1.11 – Transformation des données non-linéaires dans la figure 1.10.

1.8 Réseaux de Neurones

Les réseaux de neurones artificiels (RNA) représentent une méthode puissante et flexible pour la classification supervisée. Inspirés du fonctionnement des neurones biologiques, ces modèles apprennent à partir de données en identifiant des motifs complexes. Cette section explore les caractéristiques fondamentales des réseaux de neurones, leur architecture, leur fonctionnement et leur application dans le cadre de la classification supervisée.

1.8.1 Historique des réseaux de neurones

L'idée des réseaux de neurones remonte aux années 40 avec les travaux de McCulloch et Pitts [MP43], qui ont proposé le premier modèle de neurone formel. En 1959, Rosenblatt [Ros58] a développé le perceptron, un réseau de neurones à une couche capable de résoudre des problèmes de classification simple. Cependant, les limites technologiques et théoriques de cette époque ont freiné leur adoption.

L'intérêt pour les réseaux de neurones a connu un regain dans les années 1980, grâce à des avancées comme l'algorithme de rétro-propagation du gradient, permettant d'entraîner des réseaux multicouches. Au début des années 2010, l'essor des techniques d'apprentissage profond (deep learning) a propulsé les réseaux de neurones au premier plan de la recherche en intelligence artificielle.

1.8.2 Structure et fonctionnement des réseaux de neurones

Comme le montre la figure 1.12, le réseau de neurones est composé de différentes couches :

— **Couche d'entrée** : Reçoit les données d'entrée.

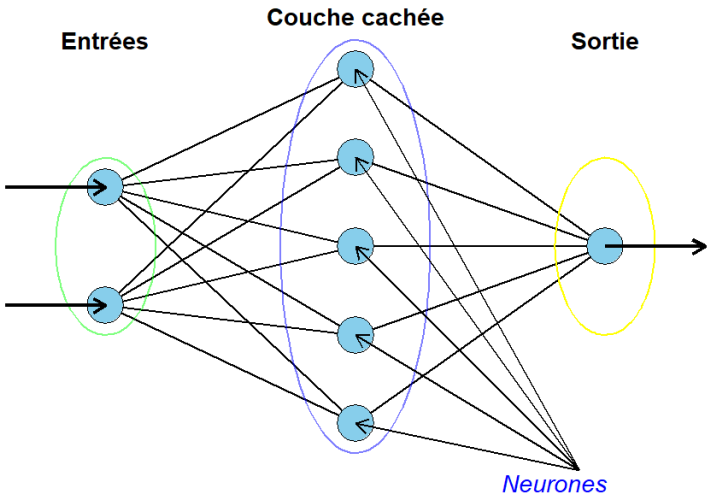

FIGURE 1.12 – Architecture d'un réseau de neurones simple

— **Couches cachées** : Effectuent des transformations non linéaires sur les données. Dans la figure 1.12, on a une seule couche cachée, mais on peut avoir plusieurs couches.

— **Couche de sortie** : Fournit la prédiction finale.

Chaque neurone effectue une combinaison linéaire des entrées suivie d'une fonction d'activation :

$$s(\boldsymbol{x}) = h(x_1, x_2, \ldots, x_p, \alpha_0, \ldots, \alpha_p) = g\left(\alpha_0 + \sum_{j=1}^{p} \alpha_j x_j\right),$$

où :

— s est l'état interne du neurone.

— $\boldsymbol{x} = (x_1, \ldots, x_p)$ est le vecteur des entrées.

— α_j sont les poids associés aux entrées.

— g est la fonction d'activation, qui peut être une de ces trois fonctions :

1. Sigmoïde
$$g(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{x}}},$$
2. ReLU
$$g(\boldsymbol{x}) = \max(0, \boldsymbol{x}).$$
3. softmax
$$g(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i}}{\sum_{j=1}^{K} e^{\boldsymbol{x}_j}}.$$

1.8.3 Algorithme d'apprentissage

L'apprentissage dans un réseau de neurones implique l'ajustement des poids pour minimiser une fonction de perte. Pour un problème de régression, la fonction de perte quadratique est souvent utilisée :

$$Q(\alpha) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i;\alpha))^2,$$

où :

— y_i est la sortie réelle.

— $f(\boldsymbol{x}_i;\alpha)$ est la prédiction du modèle.

L'algorithme de rétro-propagation est utilisé pour calculer les gradients de la fonction de perte par rapport aux poids, permettant ainsi d'ajuster les poids $\alpha_j^{(r+1)}$ à la $(r+1)$−ème itération, par la règle :

$$\alpha_j^{(r+1)} = \alpha_j^{(r)} - \tau\frac{\partial Q}{\partial \alpha_j^{(r)}},$$

où τ est le taux d'apprentissage.

Pour les problèmes de classification, la fonction de perte par entropie croisée est souvent utilisée pour calculer la différence entre les probabilités prédites par le modèle et les vraies valeurs. Elle est définie en fonction du type de classification, qu'il s'agisse d'un problème

binaire ou multi-classes. Dans le cas de la classification binaire, la fonction de perte par entropie croisée est donnée par :

$$L = -\frac{1}{n}\sum_{i=1}^{n}\Big[y_i \log(f(\boldsymbol{x}_i;\alpha)) + (1-y_i)\log(1-f(\boldsymbol{x}_i;\alpha))\Big],$$

où :

— L est la fonction de perte moyenne,

— $y_i \in \{0,1\}$ est la véritable valeur associée à la i ème observation,

— $f(\boldsymbol{x}_i,\alpha) \in [0,1]$ est la probabilité prédite par le modèle pour que la i ème observation appartienne à la classe positive,

— n est le nombre total d'observations.

Cette fonction pénalise fortement les prédictions incorrectes avec une grande confiance (probabilités proches de 0 ou 1) et encourage les prédictions confiantes lorsqu'elles sont correctes.

Remarque 12. *L'entropie croisée est détaillée dans [GBC16], cet ouvrage explique son utilisation dans les réseaux de neurones pour les problèmes de classification.*

1.8.4 Applications des réseaux de neurones dans la classification supervisée

Les réseaux de neurones sont utilisés dans divers domaines pour des tâches de classification. Par exemple :

— **Reconnaissance d'images** : Les réseaux convolutifs (CNN) sont particulièrement efficaces pour extraire des caractéristiques d'images, voir [GBC16].

— **Analyse de sentiments** : Les réseaux de neurones récurrents (RNN) peuvent modéliser des séquences de texte pour classer les sentiments, voir [MKB+10].

— **Prévision de défaillance d'entreprises** : Les réseaux de neurones permettent de classer les entreprises en fonction de leur risque de défaillance, voir [HZHA16].

Pour illustrer la capacité des réseaux de neurones à classifier des données complexes, nous avons créé un jeu de données synthétique de 250 points ayant une forme "exotique". Ces données sont générées de manière à présenter des motifs non linéaires, rendant leur séparation difficile pour les modèles de classification linéaires classiques. En utilisant un réseau de neurones, nous chercherons à apprendre cette relation complexe et à déterminer des frontières de décision qui permettent de bien classer les points dans leurs classes respectives. La figure 1.13 illustre les données initiales, présentant des motifs complexes et non linéaires qui rendent difficile la séparation des classes. La figure 1.14 montre l'application d'un réseau de neurones sur ces mêmes données. Le réseau a appris à tracer une frontière de décision qui sépare les différentes classes en fonction des relations complexes entre les caractéristiques. Cela démontre l'efficacité des réseaux de neurones à gérer la non-linéarité des données et à modéliser des relations complexes, en construisant une séparation approximative mais adaptée aux motifs observés, même dans des jeux de données difficiles.

1.9 La courbe ROC : Principe et applications

La courbe ROC est une représentation graphique de la performance d'un modèle de classification dont la variable cible est binaire. Elle illustre le compromis entre la sensibilité, qui est le taux de vrais positifs (True Positifs Rate : TPR), et la spécificité, qui est le taux de vrais négatifs (False Positifs Rate : FPR), à des différents seuils de précision. Ces deux indicateurs permettent d'apprécier la capacité d'un modèle à distinguer correctement les classes.

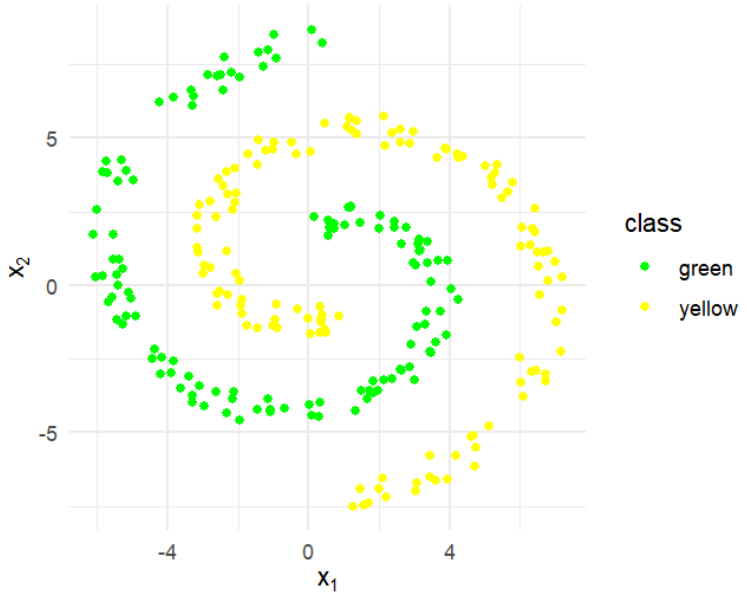

FIGURE 1.13 – Les données spirales exotiques

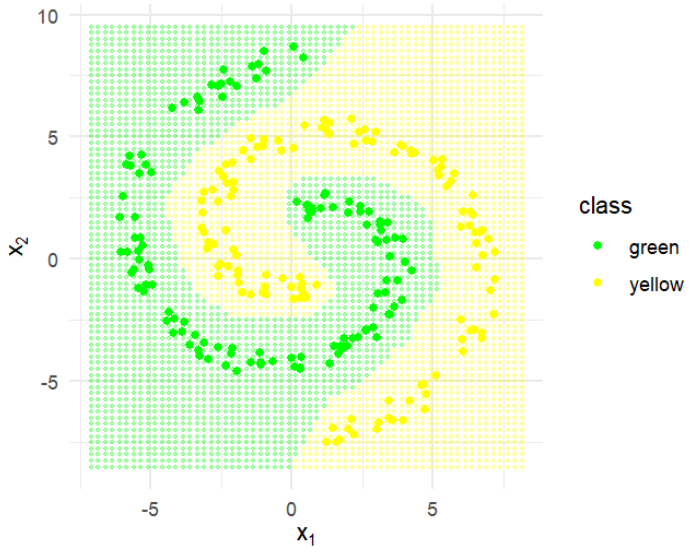

FIGURE 1.14 – Classification des spirales exotiques avec un réseau de neurones

Historiquement, la courbe ROC a été introduite pendant la Seconde Guerre mondiale pour évaluer la performance des systèmes de détection radar. Aujourd'hui, elle est utilisée dans divers domaines, notamment : la médecine pour évaluer les tests diagnostiques, la finance pour la prédiction des risques, et l'apprentissage automatique pour comparer les modèles de classification. Dans la partie suivante on va utiliser comme référence ce site : https ://psychometrie.espaceweb.usherbrooke.ca/la-sensibilite-et-la-specificiteSensibilité et spécificité (site de psychométrie, Université de Sherbrooke) pour expliquer la sensibilité et la spécificité.

1.9.1 La sensibilité et de la spécificité

Le tableau croisé 1.1 représente les différentes possibilités de classification des cas (les instances positives) et des non-cas (les instances négatives) en fonction des résultats fournis par l'instrument d'évaluation. Les colonnes reflètent le statut réel des individus, indiquant s'ils possèdent effectivement (cas) ou non (non-cas) la caractéristique étudiée. Les lignes, quant à elles, correspondent aux deux catégories de résultats que l'instrument peut produire. Sur la base des réponses aux items, l'instrument détermine si un individu appartient au groupe des cas ou des non-cas, le classant soit comme porteur de la caractéristique étudiée (positif), soit comme ne la possédant pas (négatif).

Dans une situation idéale, la majorité des observations devraient se situer sur la diagonale du tableau croisé, représentant les prédictions correctes effectuées par l'instrument. En revanche, les cellules situées hors de cette diagonale, correspondent aux erreurs de classification commises par l'instrument, qu'il s'agisse de fausses prédictions positives ou négatives.

Nous allons définir les deux termes principaux de la courbe ROC :

Résultat	**Statut réel**		**Total**
	Cas	**Non-cas**	
Positif	VP (vrais positifs)	FP (faux positifs)	VP + FP
Négatif	FN (faux négatifs)	VN (vrais négatifs)	FN + VN
Total	VP + FN	FP + VN	

TABLEAU 1.1 – Tableau croisé des résultats et du statut réel

— **La sensibilité (ou** *recall en anglais)* : Il s'agit du taux de détection des vrais positifs, calculé par la formule $\frac{VP}{VP+FN}$.

— (1– **la spécificité)** : Il s'agit du taux de faux positifs, calculé par la formule $\frac{FP}{FP+VN}$.

Pour tracer la courbe ROC, dans la figure 1.15, on calcule la sensibilité et (1 – spécificité) pour une série de seuils variant entre 0 et 1. Ces valeurs sont ensuite représentées dans un graphique où l'axe y correspond à la sensibilité, et l'axe x représente 1 – spécificité.

1.9.2 L'aire sous la courbe ROC (*Area Under the Curve, AUC)*

La courbe ROC permet de visualiser la capacité discriminante d'un modèle, tandis que l'aire sous la courbe (AUC), dans la figure 1.15, mesure l'aire située sous l'ensemble de la courbe ROC. AUC est un indicateur numérique simple et utile qui varie entre 0 et 1. Par exemple une valeur de 0,84 signifie qu'un individu du groupe positif à 84% de chance d'être mieux classé qu'un individu du groupe négatif, par contre une AUC de 0,5 indique une absence de discrimination entre les groupes, tandis qu'une AUC de 1 reflète une séparation parfaite. La figure 1.16 illustre cette interprétation. La courbe en escalier orange représente les différents couples (taux de faux positifs, taux de vrais positifs) obtenus pour chaque seuil de classification. Les points rouges indiquent quelques seuils spécifiques utilisés dans le calcul. La ligne bleu diagonale correspond à un classificateur

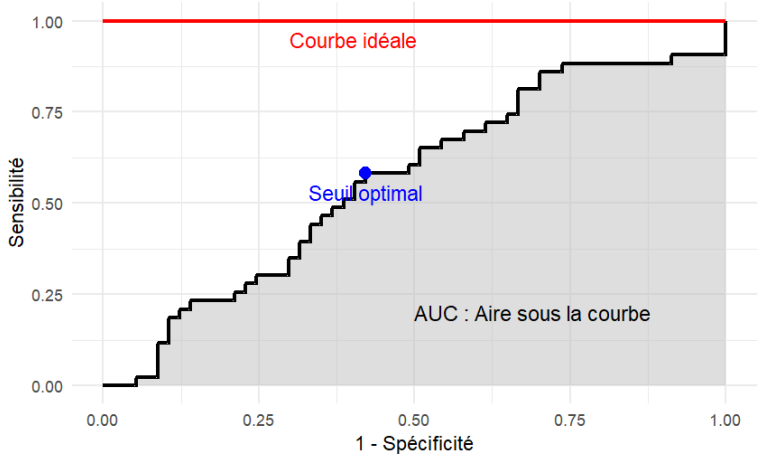

FIGURE 1.15 – Exemple de courbe ROC.

aléatoire, qui ne possède aucune capacité de discrimination. Enfin, la surface grise entre la courbe ROC et cette diagonale représente l'aire sous la courbe (AUC).

1.10 Conclusion

Dans ce chapitre, nous avons présenté deux grandes familles de méthodes de classification : les méthodes paramétriques et non paramétriques. Parmi les méthodes paramétriques, nous avons étudié l'Analyse Discriminante Linéaire (LDA), la régression logistique et les modèles de mélange. Ces techniques sont simples à implémenter, rapides à exécuter sur de petits jeux de données et facilement interprétables. Cependant, elles manquent de flexibilité lorsque les hypothèses sur la distribution des données sont incorrectes. Elles sont sensibles à une mauvaise spécification du modèle, ce qui peut affecter sur la performance du classificateur.

Les méthodes de classification non-paramétriques, telles que la méthode des k plus proches voisins (KNN), la méthode de segmentation CART, les méthodes d'ensembles

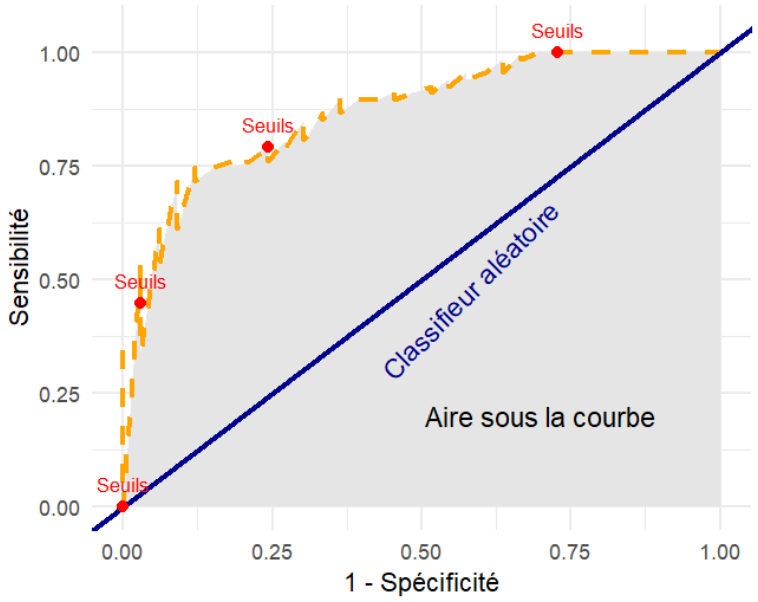

FIGURE 1.16 – Schéma de courbe ROC en orange et AUC en gris.

(boosting, bagging, forêt aléatoire), les machines à vecteurs de support (SVM) et les réseaux de neurones, sont flexibles et puissants pour modéliser des relations complexes et peu sensibles aux hypothèses sur les données. Cependant, on risque du surapprentissage (overfitting) avec ces méthodes si la régularisation n'est pas correctement utilisée. De plus, elles nécessitent un grand nombre d'observations pour assurer une bonne capacité de classification et éviter les biais d'échantillonnage.

Remarque 13. *Dans la littérature, on trouve d'autres critères pour mesurer la performance des méthodes de classification :*

— ***Exactitude (Accuracy)*** *: proportion des prédictions correctes (positives et négatives) parmi toutes les observations.*

$$Exactitude = \frac{VP + VN}{VP + VN + FP + FN}$$

— ***Précision (Precision)*** *: proportion des prédictions positives qui sont réellement*

positives.

$$Précision = \frac{VP}{VP + FP}$$

— ***Score F1*** *: moyenne harmonique entre la précision et le rappel.*

$$F_1 = 2 \cdot \frac{Précision \cdot Rappel}{Précision + Rappel} = \frac{2VP}{2VP + FP + FN}$$

CHAPITRE 2

Modèles de copules et estimation

L'une des méthodes modernes pour construire une distribution multivariée à partir de marginales spécifiées repose sur l'utilisation des copules. [Joe14] et [SH07] ont brièvement introduit ces techniques dans leurs ouvrages respectifs. Une copule permet d'élaborer une distribution multivariée tout en conservant des marginales univariées suivant une loi uniforme sur l'intervalle $[0, 1]$.

Le principe fondamental sous-jacent à la construction d'une distribution multivariée en utilisant une copule repose sur la *transformation intégrale de probabilité*. En effet, étant donné une variable aléatoire continue, X, qui suit une fonction de répartition cumulative (CDF), F, la transformation $F(X)$ suit une distribution uniforme sur l'intervalle $[0, 1]$. Cette propriété constitue le fondement de la définition des copules, introduite formellement ci-après.

Définition 14. *Une copule de dimension $d = 2$ est une fonction, C, de $[0, 1]^2$ vers $[0, 1]$ satisfaisant les propriétés suivantes :*

1. *Pour tout $u, v \in [0, 1]$, on a : $C(u, 0) = C(0, v) = 0$,*
2. *Pour tout $u, v \in [0, 1]$, on a : $C(u, 1) = u$ et $C(1, v) = v$,*

3. *Pour tout* $(u_1, u_2), (v_1, v_2) \in [0,1]^2$ *tels que* $u_1 \leq v_1$ *et* $u_2 \leq v_2$*on a :*

$$C(v_1, v_2) - C(v_1, u_2) - C(u_1, v_2) + C(u_1, u_2) \geq 0.$$

Exemple 15. *Considérons la fonction* $C(u, v) = u \times v$*, Afin de* $u, v \in [0,1]$*. Pour vérifier que cette fonction constitue bien une copule, nous devons établir qu'elle satisfait les trois propriétés définissant une copule :*

1. *Pour tout* $u, v \in [0,1]$*,* $C(u, 0) = 0 = C(0, v)$*, ce qui confirme la première condition.*
2. *Pour tout* $u, v \in [0,1]$*, on a* $C(u, 1) = u \times 1 = u$ *et* $C(1, v) = 1 \times v = v$*, établissant ainsi la validité de la deuxième condition.*
3. *Pour tout* $(u_1, u_2), (v_1, v_2) \in [0,1]^2$ *tels que* $u_1 \leq v_1$ *et* $u_2 \leq v_2$*, on a :*

$$\begin{aligned} C(v_1, v_2) - C(v_1, u_2) - C(u_1, v_2) + C(u_1, u_2) &= v_1 v_2 - v_1 u_2 - u_1 v_2 + u_1 u_2. \\ &= (v_1 - u_1)(v_2 - u_2) \geq 0. \end{aligned}$$

Comme $u_1 \leq v_1$ *et* $u_2 \leq v_2$*, le produit* $(v_1 - u_1)(v_2 - u_2)$ *est toujours positif ou nul, validant ainsi la troisième condition.*

Ainsi, la fonction $C(u, v) = uv$ *satisfait les trois propriétés de la définition d'une copule.*

On peut généraliser la définition 14 pour un vecteur dans $\mathbb{R}^d$. Les copules ont été introduite dans le cadre du théorème de Sklar 2.1, qui établi un lien fondamental entre une distribution multidimensionnelle et une copule. Ce théorème met en évidence la décomposition d'une fonction de répartition cumulative en deux composantes distinctes : les lois marginales et une copule.

Théorème 2.1 (Sklar (1958))**.** *Soit* $\mathbf{X} = (X_1, \ldots, X_d)$ *un vecteur aléatoire, et soit* F *la fonction de répartition conjointe de* $\mathbf{X}$*, dont les lois marginales sont* $F_1, \ldots, F_d$*. Alors, il existe une fonction de copule* C *telle que :*

$$\forall (x_1, \ldots, x_d) \in \mathbb{R}^d, \quad F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)).$$

La densité de copule est définie par :

$$c(u_1, \ldots, u_d) = \frac{\partial^d}{\partial u_1 \ldots \partial u_d} C(u_1, \ldots, u_d),$$

où $C(u_1, \ldots, u_d)$ est la copule.

2.1 Modèles de copules

Dans la présente section, nous présentons les principaux modèles de copules paramétriques couramment utilisés en pratique. Ces modèles jouent un rôle fondamental dans la modélisation des dépendances entre variables et offrent une grande flexibilité pour diverses applications en statistique, en finance et dans d'autres domaines.
Nous introduisons tout d'abord les copules elliptiques, qui reposent sur des distributions elliptiques et incluent certaines des copules les plus largement utilisées en raison de leur simplicité et de leurs propriétés mathématiques élégantes. Ensuite, nous examinerons les copules archimédiennes, qui se distinguent par leur structure analytique et leur capacités à capturer des dépendances asymétriques, ce qui les rend particulièrement adaptées à de nombreuses applications empiriques. Chaque famille de copule sera illustré par des exemples concrets afin d'en clarifier les propriétés et les usages.

2.1.1 Copules Elliptiques

Les copules elliptiques sont issues des distributions elliptiques et constituent une classe essentielle pour la modélisation des dépendances. Elles offrent une flexibilité et elles sont largement utilisées dans divers contextes appliqués.
Dans ce qui suit, nous définissons brièvement le concept de distribution elliptique, avant de présenter deux exemples de copules elliptiques : la copule gaussienne et la copule t de Student.

Définition 16. *Un vecteur aléatoire* $\mathbf{X} = (X_1, \ldots, X_d)$ *suit une distribution elliptique s'il peut être représenté par l'expression suivante :*

$$\mathbf{X} = \boldsymbol{\mu} + \mathbf{RAU},$$

où :

- $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d) \in \mathbb{R}^d$ *est le vecteur de moyennes,*
- $\mathbf{U}$ *est un vecteur aléatoire uniformément distribué sur la sphère unité dans* $\mathbb{R}^d$*,*
- $\mathbf{R}$ *est un vecteur aléatoire indépendant de* $\mathbf{U}$*,*
- $\mathbf{A}$ *est une matrice de dimension* $d \times d$ *telle que* $\Sigma = \mathbf{AA}^T$ *est non singulière.*

Définition 17. *Si elle existe, la densité d'une distribution elliptique est donnée par l'expression suivante :*

$$f(x) = |\Sigma|^{-1/2} \, g\left((\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right),$$

où g *est une fonction définie sur* $\mathbb{R}^+$*, appelée fonction génératrice de densité. Cette fonction* g *détermine la forme spécifique de la distribution elliptique.*

Les copules elliptiques incluent principalement la copule gaussienne et la copule t de Student, qui sont largement utilisées en pratique.

Définition 18. *La copule gaussienne d'un vecteur,* (X_1, X_2)*, également appelée copule normale, est définie par :*

$$C(u_1, u_2; \rho) = \Phi_\rho\left(\Phi^{-1}(u_1), \Phi^{-1}(u_2)\right),$$

où, ρ *représente le coefficient de corrélation entre les deux variables* X_1 *et* X_2*,* Φ^{-1} *désigne la fonction quantile (ou fonction réciproque) de la loi normale standard univariée, et* Φ_ρ *est la fonction de répartition de la loi normale bidimensionnelle centrée, avec une matrice de corrélation donnée par :*

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

La copule gaussienne permet de modéliser les dépendances linéaires entre deux variables en fonction de leur coefficient de corrélation ρ.

L'intérêt de la copule gaussienne réside dans son lien direct avec la distribution normale multidimensionnelle. Elle permet de modéliser la dépendance entre variables à travers le coefficient de corrélation linéaire, offrant ainsi une approche naturelle dans le cadre de relations linéaires. Toutefois, lorsque l'objectif est de capturer une dépendance non linéaire ou de modéliser des co-movements dans les queues de distribution tels que les événements extrêmes d'autres familles de copules s'avèrent plus appropriées. Celles-ci seront introduites dans les sections ultérieures.
En procèdent à la dérivation de la formule de la copule gaussienne, on obtient l'expression de sa densité en dimensions deux, donnée par :

$$c(u_1,u_2;\rho) = \frac{\exp\left(-\frac{\left(\Phi^{-1}(u_1)-\Phi^{-1}(u_2)\right)^2}{2(1-\rho^2)}\right)}{\sqrt{2(1-\rho^2)}}\left(\frac{1}{\sqrt{1-\rho^2}}\right) = \frac{1}{\sqrt{1-\rho^2}}\exp\left(\frac{x_1^2+x_2^2-2\rho x_1x_2}{2(1-\rho^2)}-\frac{x_1^2+x_2^2}{2}\right),$$

où $x_1 = \Phi^{-1}(u_1)$ et $x_2 = \Phi^{-1}(u_2)$. Lorsque $\rho = 0$, la densité de la copule gaussienne se réduit à la forme suivante :

$$c(u_1,u_2;\rho=0) = 1.$$

La figure 2.1 illustre la densité de la copule gaussienne bidimensionnelle pour une valeur de $\rho = 0.5$.

Définition 19. *La copule de Student bidimensionnelle est définie par l'expression suivante :*

$$C(u_1,u_2;\rho,\nu) = T_{\nu,\rho}(t_\nu^{-1}(u_1),t_\nu^{-1}(u_2)),$$

où ρ désigne le coefficient de corrélation et $T_{\nu,\rho}$ est la fonction de répartition de la loi de Student bidimensionnelle standard, caractérisée par une matrice de corrélation dépendant de ρ et un paramètre ν correspondant au nombre de degrés de liberté.

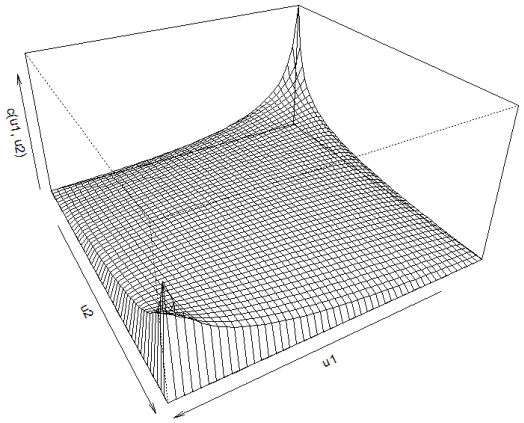

FIGURE 2.1 – Densité de la copule gaussienne bidimensionnelle avec $\rho = 0.5$

À la différence de la copule gaussienne, la copule de Student est en mesure de capturer des dépendances marquées dans les queues de la distribution, aussi bien à gauche qu'a droite. Cette propriété lui confère une aptitude particulière à modéliser des événements extrêmes dans les deux directions, grâce à la présence de queues plus épaisses. Une telle caractéristique est essentielle dans de nombreuses applications pratiques où la prise en compte des extrêmes est primordiale. La dépendance extrême est présente pour $\rho \neq -1$, comme le montre le tableau 2.1.

Copule elliptique	**Paramètres**	**Tau de Kendall**	**Dépendance extrême**
Gaussienne	$\rho \in (-1, 1)$	$\frac{2}{\pi} \arcsin(\rho)$	0
Student	$\rho \in (-1, 1)$, $\nu > 2$	$\frac{2}{\pi} \arcsin(\rho)$	$2t_{\nu+1}\left(-\sqrt{\nu} + 1\sqrt{\frac{1-\rho}{1+\rho}}\right)$

TABLEAU 2.1 – Caractéristiques des copules elliptiques.

Pour la copule de Student, une mesure de dépendance dans la queue supérieure est donnée par l'expression suivante :

$$2\, t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right),$$

où $t_{\nu+1}(\cdot)$ désigne la fonction de répartition de la loi de Student univariée à $\nu+1$ degrés de liberté.

Définition 20. *La densité de la copule de Student à deux dimensions s'écrit comme suit :*

$$c(u_1, u_2; \rho, \nu) = \frac{\kappa}{\sqrt{1-\rho^2}} \frac{\Gamma\left(\frac{\nu+2}{2}\right)}{\Gamma\left(\frac{\nu+1}{2}\right)^2} \left(1 + \frac{x_1^2 + x_2^2 - 2\rho x_1 x_2}{\nu(1-\rho^2)}\right)^{-\frac{\nu+2}{2}},$$

où $x_1 = t_\nu^{-1}(u_1)$, $x_2 = t_\nu^{-1}(u_2)$, $\kappa > 0$, $\rho \in (-1, 1)$ *et* ν *désigne le nombre de degrés de liberté. La copule de Student (également appelée t-copule) est construite selon un principe analogue à celui de la copule gaussienne ; la distribution normale multidimensionnelle est remplacée par la distribution de Student multidimensionnelle.*

La figure 2.2 représente la densité de la copule de Student en dimension deux pour un coefficient de corrélation $\rho = 0.5$ et $\nu = 14$ degrés de liberté.

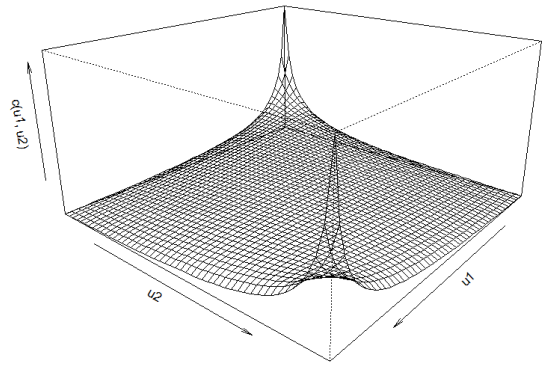

FIGURE 2.2 – Densité de la copule de Student avec $\rho = 0.5$ et $\nu = 14$.

À la différence de la copule gaussienne, qui ne présente aucune dépendance dans les queues, la copule de Student est capable de représenter une dépendance significative entre événements extrêmes. Cette propriété, en fait, est un outil particulièrement utile pour la modélisation des co-mouvements sous des conditions extrêmes.

Nous allons à présent démontrer que le tau de Kendall associé aux copules gaussienne et Student s'exprime selon la relation suivante :

$$\tau = \frac{2}{\pi} \arcsin(\rho).$$

Pour ce faire, considérons la définition de $\tau(X, Y)$, donnée par :

$$\tau = \mathbb{P}(X_1 \geq X_2, Y_1 \geq Y_2) - \mathbb{P}(X_1 \geq X_2, Y_1 \leq Y_2) = 4\mathbb{P}(X_1 \geq X_2, Y_1 \geq Y_2) - 1,$$

où (X_1, Y_1) et (X_2, Y_2) sont deux vecteurs aléatoires indépendants suivant la même loi conjointe que le vecteur (X, Y). Notons par F (resp. G) la distribution de X (resp. Y), $U_j = F(X_j), V_j = G(Y_j), Z_j = \Phi^{-1}(U_j)$ et $W_j = \Phi^{-1}(V_j), j = 1, 2$. Les couples (Z_j, W_j) sont indépendants et de même distribution gaussienne :

$$f_{Z,W}(z, w) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(z^2 + w^2 - 2\rho zw\right)\right),$$

où ρ est le coefficient de corrélation entre X et Y.

Par conséquent,

$$\begin{aligned}
\mathbb{P}(X_1 \geq X_2, Y_1 \geq Y_2) &= \mathbb{P}(U_1 \geq U_2, V \geq V_2) \\
&= \mathbb{P}(Z_1 \geq Z_2, W_1 \geq W_2) \\
&= \mathbb{P}\left(\frac{Z_1 - Z_2}{\sqrt{2}} \geq 0, \frac{W_1 - W_2}{\sqrt{2}} \geq 0\right)
\end{aligned}$$

Or, le couple $\left(\frac{Z_1-Z_2}{\sqrt{2}} \geq 0, \frac{W_1-W_2}{\sqrt{2}} \geq 0\right)$ suit la même loi $f_{(Z,W)}$. Donc,

$$\begin{aligned}
\mathbb{P}(X_1 \geq X_2, Y_1 \geq Y_2) &= \int_0^\infty \int_0^\infty f_{Z,W}(z, w)\, dzdw \\
&= \int_0^\infty \int_0^\infty \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(z^2 + w^2 - 2\rho zw\right)\right) dzdw.
\end{aligned}$$

Pour simplifier cette intégrale, effectuons le changement de variables suivant :

$$s = \frac{z - \rho w}{\sqrt{1-\rho^2}}, \quad t = w.$$

Ainsi, les différentielles se transforment selon :

$$dzdw = \sqrt{1-\rho^2}\, dsdt.$$

En substituant ces expressions dans l'intégrale de la probabilite conjointe, on obtient :

$$\mathbb{P}(X_1 \geq X_2, Y_1 \geq Y_2) = \frac{1}{2\pi} \int_0^{\infty} \int_{-\frac{\rho t}{\sqrt{1-\rho^2}}}^{\infty} \exp\left(-\frac{1}{2}(s^2+t^2)\right) dsdt.$$

En passant ensuite aux coordonnées polaires, à l'aide des substitutions $s = r\cos\theta$ et $t = r\sin\theta$, (ce qui implique que $dsdt = rdrd\theta$), l'intégrale devient :

$$\tau = \frac{1}{\pi} \int_0^{\pi/2+\arctan(\rho/\sqrt{1-\rho^2})} \int_0^{\infty} r \exp\left(-\frac{r^2}{2}\right) drd\theta - 1.$$

Après intégration, nous obtenons :

$$\tau = \frac{2}{\pi} \arctan\left(\frac{\rho}{\sqrt{1-\rho^2}}\right).$$

En simplifiant cette dernière expression à l'aide d'identités trigonométriques, nous arrivons à la formule finale :

$$\tau = \frac{2}{\pi} \arcsin(\rho).$$

2.1.2 Copules Archimédiennes

Les copules archimédiennes, introduites initialement par Schweizer [SS81], permettent de générer une grande variété de familles de copules, offrant des structures de dépendance flexibles, y compris des dépendances asymétriques entre les queues inférieure et supérieure. Grâce à leur forme analytique explicite, les copules archimédiennes sont aisément manipulables et simulables, ceci les rend particulièrement adaptées à la modélisation de relations complexes de dépendance. Dans le domaine de l'assurance et de la réassurance, les copules de Clayton ou Gumbel, sont largement utilisées pour modéliser les interdépendances entre différents types de risques, tels que les risques de mortalité, de morbidité

ou les risques matériels. Elles sont également appliquées en finance, notamment pour l'analyse du risque de crédit et l'étude des événements extrêmes.

Définition 21. *Une copule C, sur le carré unité $[0,1]^2$, est dite archimédienne s'il existe une fonction génératrice $\varphi : [0,1] \to [0,\infty]$, continue, strictement décroissante, convexe, et $\varphi(1) = 0$, vérifiant :*

$$C(u,v) = \varphi^{-1}(\varphi(u) + \varphi(v)) \quad \textit{pour tout} \quad (u,v) \in [0,1]^2.$$

La fonction φ est alors appelée la génératrice archimédienne de C. Lorsque $\varphi(0) = \infty$, la copule est dite strictement archimédienne, et φ est qualifiée de génératrice archimédienne stricte.

Exemple 22. *La copule indépendante correspond à la fonction génératrice $\varphi(t) = -ln(t)$.*

Nous présenterons dans la suite quelques familles de copules archimédiennes, à savoir : la copule de Clayton, la copule de Gumbel, la copule de Frank et la copule de Joe.

La famille de copules de Clayton

La Copule de Clayton [Cla78], appartenant à la famille des copules archimédiennes, est couramment utilisée pour modéliser une dépendance asymétrique, en particulier une forte dépendance dans la queue inférieure. Elle est définie à l'aide de sa fonction génératrice $\varphi_\theta(u)$, donnée par :

$$\varphi_\theta(u) = \frac{1}{\theta}(u^{-\theta} - 1),$$

où le paramètre $\theta > 0$ contrôle l'intensité de la dépendance : plus sa valeur est élevée, plus la dépendance dans la queue inférieure est forte. Quand θ tends vers zéro, $\varphi_\theta(u)$ converge vers $-ln(u)$; ce qui correspond à la copule d'indépendence.
La fonction de copule de Clayton s'exprime alors sous la forme :

$$C(u,v) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-1/\theta}, \quad \text{pour} \quad u,v \in [0,1] \quad \text{et} \quad \theta > 0.$$

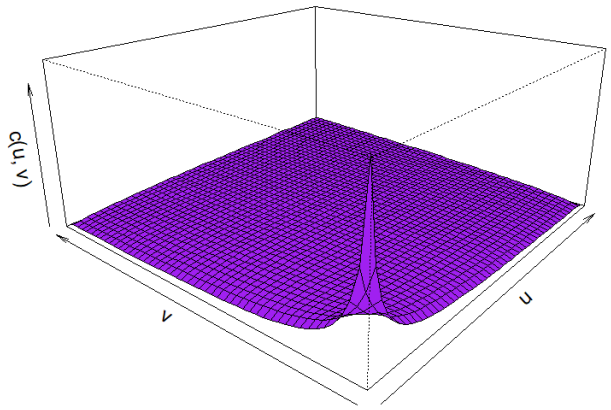

FIGURE 2.3 – Densité de la copule de Clayton pour $\theta = 0.5$.

Cette copule est différentiable et sa densité est donnée par :

$$c(u, v; \theta) = (\theta + 1)(uv)^{-\theta-1}(u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}-2}.$$

La figure 2.3 illustre la densité bidimensionnelle de la copule de Clayton avec un paramètre $\theta = 0.5$.

La famille de copules de Gumbel

La copule de Gumbel, nommée en l'honneur d'Emil Julius Gumbel (1960) [Gum60], appartient également à la famille des copules archimédiennes. Elle se distingue par sa capacité à modéliser une forte dépendance dans la queue supérieure des distributions marginales. Elle est définie par une fonction génératrice de la forme :

$$\varphi_\theta(t) = (-\ln t)^\theta, \theta \geq 1.$$

Pour $\theta = 1$, la copule de Gumbel correspond à la copule d'indépendance. La copule de Gumbel s'écrit alors :

$$C_\theta(u, v) = \exp\left\{-\left[(-\ln u)^\theta + (-\ln v)^\theta\right]^{1/\theta}\right\}, \quad \theta \geq 1.$$

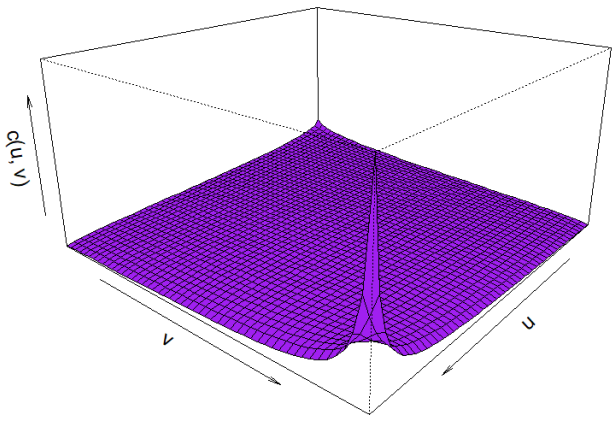

FIGURE 2.4 – Densité de la copule de Gumbel pour $\theta = 1.2$.

Sa densité peut être exprimée sous la forme suivante :

$$c_\theta(u,v) = -\frac{\frac{\theta}{C_\theta(u,v)^2}\left[(-\ln u)^\theta + (-\ln v)^\theta\right]^{\frac{\theta-2}{\theta}}\left[\theta - 1 + \left((-\ln u)^\theta + (-\ln v)^\theta\right)^{\frac{1}{\theta}}\right]\frac{\theta}{u}(-\ln u)^{\theta-1}\frac{\theta}{v}(-\ln v)^{\theta-1}}{\frac{\theta^3}{C_\theta(u,v)^3}\left[(-\ln u)^\theta + (-\ln v)^\theta\right]^{\frac{3(\theta-1)}{\theta}}}.$$

Après simplifications, la densité de la copule s'écrit comme suit :

$$c_\theta(u,v) = C_\theta(u,v)\left[\varphi_\theta(u) + \varphi_\theta(v)\right]^{\frac{1}{\theta}-2}\left[\theta - 1 + \left(\varphi_\theta(u) + \varphi_\theta(v)\right)^{\frac{1}{\theta}}\right]\frac{\varphi_{\theta-1}(u)\varphi_{\theta-1}(v)}{uv}.$$

La figure 2.4 présente la densité bidimensionnelle de la copule de Gumbel pour un paramètre $\theta = 1.2$.

Cette structure permet de capturer des formes de dépendance asymétrique, en mettant particulièrement l'accent sur la dépendance dans la queue supérieure. Elle s'avère ainsi précieuse pour l'étude de phénomènes rares mais significatifs, notamment dans les domaines de gestion des risques et de la modélisation des extrêmes.

La famille de copules de Frank

Une autre copule appartenant à la famille archimédienne est la copule de Frank, introduite par William Frank en 1979, [Fra79]. Elle se caractérise par une fonction génératrice exponentielle, qui lui confère la capacité de modéliser une structure de dépendance symétrique entre variables aléatoires. Contrairement à d'autres copules archimédiennes telles que celles de Clayton ou de Gambel, la copule de Frank ne présente pas de dépendance dans les queues de distribution. Cette propriété la rend particulièrement adaptée aux situations où l'on souhaite représenter une dépendance globale sans accentuer les corrélations dans les valeurs extrêmes.

La copule de Frank bidimensionnelle est définie, pour tout $u, v \in [0,1]$, par :

$$C_\theta(u,v) = \log_\theta\left[1 + \frac{(\theta^u - 1)(\theta^v - 1)}{\theta - 1}\right], \quad \theta > 0, \theta \neq 1.$$

La fonction génératrice associée à cette copule est donnée par :

$$\varphi_\theta(t) = -\ln\left(\frac{\exp(-\theta t) - 1}{\exp(-\theta) - 1}\right), \quad t \in [0,1].$$

La figure 2.5 illustre la densité de la copule de Frank bidimensionnelle pour un paramètre de dépendance $\theta = 1.2$. Si θ est proche de zéro, la fonction génératrice, $\varphi_\theta(t)$, tend vers $-ln(t)$; qui est égale à fonction génératrice de la copule de l'indépendance.

La copule de Joe

Une autre copule Archimédienne largement utilisée est la Copule de Joe [Joe97]. Elle se distingue par sa capacité à modéliser une forte dépendance dans la queue supérieure des distribution. Cette copule est définie par sa fonction génératrice :

$$\varphi(t) = -\ln\left(1 - (1-t)^\theta\right), \quad \text{avec } \theta \geq 1.$$

L'expression de la copule de Joe est donnée par :

$$C_\theta(u,v) = 1 - \left[\left(1 - (1-u)^\theta\right)^{\frac{1}{\theta}} + \left(1 - (1-v)^\theta\right)^{\frac{1}{\theta}} - 1\right]^\theta.$$

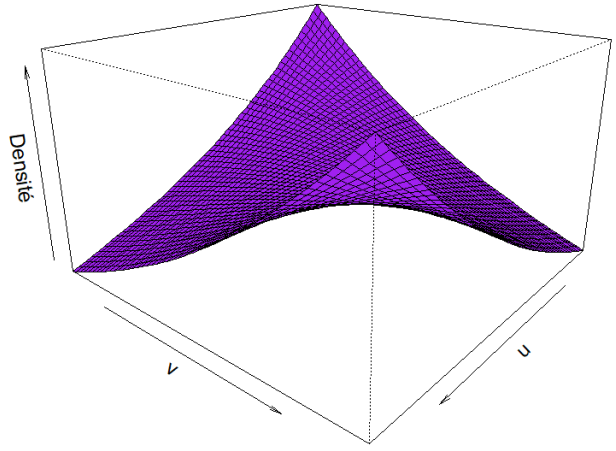

FIGURE 2.5 – Densité de la copule de Frank pour $\theta = 1.2$.

La densité correspondante s'écrit :

$$c_\theta(u,v) = -\frac{\theta(1-u)^{\theta-1}(1-v)^{\theta-1}}{\left[1-\left((1-u)^\theta+(1-v)^\theta-(1-u)^\theta(1-v)^\theta\right)^{\frac{1}{\theta}}\right]^{\theta+1}}.$$

Le paramètre $\theta \geq 1$ contrôle le degré de dépendance entre les variables. Si $\theta = 1$, on retrouve la copule de l'indépendance. Une valeur élevée de θ implique une forte dépendance dans les queues de distribution, rendant cette copule particulièrement utile dans des applications liées aux phénomènes extrêmes comme, en finance ou en gestion des risques. Le paramètre θ est relié au tau de Kendall à travers la fonction Debye

$$D_n(x) = \frac{n}{x^n}\int_0^x \frac{t^n}{e^t-1}\,dt, \quad x > 0, n \geq 1.$$

La figure 2.6 montre la densité de la copule de Joe bidimensionnelle pour un paramètre de dépendance $\theta = 2$. Le tableau 2.2 résume certaines propriétés caractéristiques des copules archimédiennes bidimensionnelles.

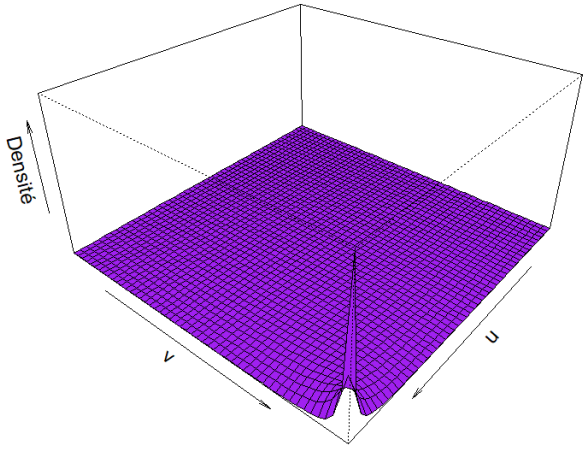

FIGURE 2.6 – Densité de la copule de Joe pour $\theta = 2$.

Copule archimédienne	Générateur	Copule bidimensionnelle	Tau de Kendall	Dépendance extrême
Clayton ($\theta > 0$)	$u^{-\theta} - 1$	$(u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$	$\frac{\theta}{\theta+2}$	$(2^{-1/\theta}, 0)$
Gumbel ($\theta \geq 1$)	$(-\ln(u))^{\theta}$	$\exp\left(-(\tilde{u}_1^{\theta} + \tilde{u}_2^{\theta})^{1/\theta}\right)$	$1 - \frac{1}{\theta}$	$(0, 2 - 2^{1/\theta})$
Frank ($\theta \neq 0$)	$-\ln\left(\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right)$	$-\frac{1}{\theta}\ln\left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}\right)$	$\left(1 - \frac{4}{\theta} + \frac{4D_1(\theta)}{\theta}\right)$	$(0, 0)$
Joe ($\theta \geq 1$)	$-\ln[1 - (1-u)^{\theta}]$	$1 - [(1-u_1)^{\theta} + (1-u_2)^{\theta} - (1-u_1)^{\theta}(1-u_2)^{\theta}]^{1/\theta}$	$1 - \frac{4}{\theta} + \frac{4}{\theta}\sum_{k=1}^{\infty}\left(\frac{1}{k(1-2^{-k})}\right)$	$(0, 2 - 2^{1/\theta})$

TABLEAU 2.2 – Caractéristiques des copules archimédiennes

2.2 Estimation d'une copule

L'estimation d'une copule joue un rôle fondamental en modélisation statistique multivariée. Elle permet de caractériser la dépendance entre plusieurs variables aléatoires. Avant de présenter la méthode de vraisemblance maximale fondée sur la copule (Full Maximum Likelihood, FLM), il convient de rappeler la représentation canonique de la densité conjointe d'un vecteur aléatoire $(X_1, X_2, \ldots, X_d)$, en s'appuyant sur une copule C et les fonctions de distribution marginales F_j. Cette densité conjointe peut s'écrire comme suit :

$$f(x_1, x_2, \ldots, x_d) = c(F_1(x_1), F_2(x_2), \ldots, F_d(x_d)) \times \prod_{j=1}^{d} f_j(x_j),$$

où f_j désigne la densité marginale de la variable X_j et c est la densité de copule, définie par :

$$c(u_1, u_2, \ldots, u_d) = \frac{\partial^n C(u_1, u_2, \ldots, u_d)}{\partial u_1 \partial u_2 \cdots \partial u_d}.$$

Cette représentation canonique permet de décomposer le problème de modélisation statistique des copules en deux étapes distinctes :

— L'identification et l'estimation des distributions marginales ;

— La spécification et l'estimation de la fonction copule adaptée à la structure de dépendance observée.

2.2.1 Méthode FML (Full Maximum Likelihood)

L'estimation des paramètres d'une copule repose fréquemment sur la méthode du maximum de vraisemblance. Soit $\mathbf{X} = \{x_{1i}, x_{2i}, \ldots, x_{di}\}_{i=1}^{n}$ la matrice des observations. La fonction du log-vraisemblance associée s'écrit comme suit :

$$\ell(\theta) = \sum_{i=1}^{n} \ln(c(F_1(x_{1i}; \theta_1), \ldots, F_d(x_{di}; \theta_d)); \alpha) + \sum_{i=1}^{n} \sum_{j=1}^{d} \ln f_j(x_{ji}; \theta_j),$$

où $\boldsymbol{\theta} = \{\alpha, \theta_1, \cdots, \theta_d\}$ désigne l'ensemble des paramètres des distributions marginales F_j ainsi que la copule c. L'estimateur du maximum de vraisemblance, noté FML, est obtenu en maximisant la fonction de log-vraisemblance :

$$\hat{\theta}_{FML} = \arg\max_{\theta \epsilon \boldsymbol{\theta}} \ell(\theta).$$

Sous des conditions de régularité (comme discuté par Serfling [Ser80] et Shao en 1999 [Sha99]), l'estimateur du maximum de vraisemblance est à la fois convergeant et asymptotiquement efficace. De plus, il satisfait la propriété de normalité asymptotique :

$$\sqrt{T}(\hat{\theta}_{MLE} - \theta_0) \xrightarrow{d} N(0, J^{-1}(\theta_0)),$$

où $J(\theta_0)$ désigne la matrice d'information de Fisher évaluée en θ_0, et θ_0 la vraie valeur du vecteur de paramètres ayant généré les données. Pour plus de détails sur ces propriétés asymptotiques, on pourra se référer à [Rao73].

2.2.2 Méthode IFM (Inference Functions for Margins)

La méthode du maximum de vraisemblance (FML) peut s'avérer particulièrement coûteuse en terme de calcul, notamment lorsque le modèle comporte un grand nombre de paramètres. En effet, cette approche nécessite l'estimation simultanée des paramètres des distributions marginales ainsi que ceux décrivant la structure de dépendance, représentée par la copule. Afin de remédier à cette complexité, Joe et Xu (1996) [JX96] ont proposé une alternative qui consiste à découper le processus d'estimation en deux étapes successives, simplifiant ainsi considérablement le calcul.

Dans cette approche, la fonction de log-vraisemblance totale est décomposée en deux composantes : une partie relative aux densités marginales univariantes, et une autre associée à la densité de la copule. La méthode IFM consiste alors à procéder en deux étapes distinctes :

1. **Estimation des paramètres marginaux** : Dans une première phase, on estime les paramètres des marginales, notés $\theta_1^* = (\theta_1, \ldots, \theta_d)$, en maximisant séparément la log-vraisemblance de chaque distribution marginale univariante. Cette étape est formalisée comme suit :

$$\hat{\theta}_j = \arg\max_{\theta_j} \sum_{i=1}^{n} \ln f_j(x_{ji}; \theta_j), j = 1, \ldots, d.$$

   Dans cette première étape, chaque distribution marginale est estimée indépendamment des autres. Ceci permet de réduire considérablement la complexité computationnelle de l'estimation globale.

2. **Estimation des paramètres de la copule** : Dans la seconde étape, le vecteur des paramètres de la copule, notés α, est estimé en maximisant la log-vraisemblance

conditionnelle de la copule. Cette estimation est effectuée en utilisant les valeurs des paramètres marginaux obtenues lors de la première étape, notées $(\hat{\theta}_j; j = 1, \ldots, d)$. Cette étape s'écrit formellement comme suit :

$$\hat{\alpha} = \arg\max_{\alpha} \sum_{i=1}^{n} \ln c(F_1(x_{1i}, \hat{\theta}_1), F_2(x_{2i}, \hat{\theta}_2), \ldots, F_d(x_{di}, \hat{\theta}_d); \alpha).$$

Le résultat final de la méthode IFM est l'estimateur $\hat{\theta}_{\text{IFM}} = (\hat{\alpha}, \hat{\theta}_1, \cdots, \hat{\theta}_d)$, correspondant respectivement aux estimations des paramètres des distributions marginales et de la copule.

2.2.3 Méthode CML (Canonical Maximum Likelihood)

La méthode de vraisemblance maximale canonique (Canonical Maximum Likelihhod, CML) constitue une approche semi-paramétrique d'estimation des copules. Elle permet d'ajuster un modèle de dépendance sans imposer une forme paramétrique particulière aux distributions marginales. En effet, contrairement à la méthode IFM, qui repose sur une estimation paramétrique séparé des marges, la méthode CML s'appuie sur l'utilisation de distributions empiriques pour estimer les paramètres de dépendance de la copule. Cette approche a été introduite par Genest et al. (1995) [GGR95]. Elle a été détaillé aussi par [SL95] dans le but de simplifier le processus d'estimation tout en réduisant la sensibilité aux erreurs de spécification des distribution marginales.
Le processus d'estimation selon la méthode CML peut être résumé en deux étapes principales :

1. **Transformation des données en variables uniformes** : On considère un échantillon $\{x_{1i}, x_{2i}, \ldots, x_{di}\}_{i=1}^{n}$. Chaque variable est transformée en une variable uniforme à l'aide de sa fonction de répartition empirique. Pour la variable X_j, la fonction de répartition empirique s'écrit :

$$u_{ji} = F_{j,n}(x_{ji}) = \frac{1}{n+1} \sum_{k=1}^{n} \mathbf{1}_{\{x_{jk} \leq x_{ji}\}}, \tag{2.1}$$

où **1** est la fonction indicatrice. La normalisation par $(n+1)$ au lieu de n permet d'éviter les problèmes avec certaines densités de copules (qui ne sont pas bornées au bornes 0 et/ou 1). Cette transformation en variables uniformes permet de construire un nouvel échantillon sans supposer une forme spécifique pour les distributions marginales. Ainsi, on évite le besoin de spécifier des modèles paramétriques pour les fonctions cumulatives de chaque marge.

2. **Estimation des paramètres de la copule** : Une fois les données transformées, les paramètres de la copule peuvent être estimés en maximisant la pseudo log-vraisemblance conditionnelle suivante :

$$\alpha = \arg\max_{\alpha} \sum_{i=1}^{n} \ln[c(F_{1,n}(x_{1i}), F_{2,n}(x_{2i}), \ldots, F_{d,n}(x_{di}); \alpha)]$$

où c représente la densité de la copule, et les $F_{j,n}(x_{ji})$ est la version empirique de distribution marginale de X_j définie par (2.1). Cette étape optimise la pseudo vraisemblance pour les paramètres de la copule indépendamment des spécifications paramétriques des distributions marges.

Dans leur étude de 2007, Kim, J. Silvapulle et P. Silvapulle [KSS07] ont comparé diverses méthodes paramétriques et semi-paramétriques pour l'estimation des copules. Ils ont constaté que les méthodes IFM et FML manquent de robustesse lorsqu'il y a des erreurs dans la spécification des distributions marginales, ce qui peut affecter la qualité des estimations. A l'inverse, la méthode CML s'est révélée nettement plus robuste. Bien qu'elle soit conceptuellement similaire à la méthode IFM, elle se distingue par une résistance aux erreurs de spécification des distributions marginales, tout en étant tout aussi simple a mettre en œuvre.

2.3 Distribution conditionnelle pour les copules

Selon le théorème de Sklar, pour une variable de réponse notée Y et un vecteur de variables explicatives, $X = (X_1, X_2, \ldots, X_d)$, il existe une fonction copule C qui relie les distributions marginales et la distribution conjointe. Ceci permet de modéliser la probabilité conditionnelle en fonction de la copule.

On considère le vecteur de distributions marginales défini par :

$$\mathbf{F}(X) = (F_1(x_1), F_2(x_2), \ldots, F_d(x_d)), \quad \mathbf{X} \in \mathbb{R}^d,$$

et la copule C, définie sur :

$$\prod_{i=1}^{d} \mathrm{Ran}(F_i),$$

où $\mathrm{Ran}(F_i)$ désigne l'image de F_i.

Proposition 23. *La fonction de répartition conditionnelle de Y sachant que $\mathbf{X} = \mathbf{x}$ peut être exprimée à partir de la copule C comme suit :*

$$P(Y \leq y \mid \mathbf{X} = \mathbf{x}) = C^c\big(F_0(y) \mid \mathbf{F}(\mathbf{X})\big) \quad (y, \mathbf{x}) \in [0,1] \times \mathbb{R}^d,$$

où la fonction F_0 désigne la fonction de répartition marginale de la variable aléatoire Y, définie par :

$$F_0(y) = P(Y \leq y), \quad \textit{pour tout } y \in \mathbb{R}$$

et

$$C^c(u \mid \mathbf{v}) = \frac{\partial v_1 \cdots \partial v_d C(u, v_1, \ldots, v_d)}{\partial v_1 \cdots \partial v_d C(1, v_1, \ldots, v_d)}. \tag{2.2}$$

Démonstration

D'après le théorème de Sklar, la fonction de répartition conjointe de $(Y, \mathbf{X})$, avec $\mathbf{X} = (X_1, \ldots, X_d)$, peut s'exprimer à l'aide d'une copule C telle que :

$$F(y, \mathbf{x}) = C\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big)$$

où F_0 est la fonction de répartition marginale de Y, et F_j celle de X_j, pour $j = 1, \ldots, d$. En dérivant cette expression, on obtient la densité conjointe :

$$f(y, \mathbf{x}) = c\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big) f_0(y) f_1(x_1) \ldots f_d(x_d).$$

où c désigne la densité de la copule C, et $f_0, f_1, \ldots, f_d$ sont les densités marginales respectives.

Par définition, la fonction de répartition conditionnelle de Y sachant $\mathbf{X} = \mathbf{x}$ est donnée par :

$$P(Y \leq y \mid \mathbf{X} = \mathbf{x}) = \int_{-\infty}^{y} f(t \mid \mathbf{x})\, dt.$$

En utilisant l'expression de la densité conditionnelle :

$$P(Y \leq y \mid \mathbf{X} = \mathbf{x}) = \frac{1}{f(x)} \int_{-\infty}^{y} f(t, \mathbf{x})\, dt.$$

Or, la densité jointe de $(X_1, \ldots, X_d)$ peut s'écrire sous la forme :

$$f(x_1, \ldots, x_d) = c^*\big(F_1(x_1), \ldots, F_d(x_d)\big) f_1(x_1) \ldots f_d(x_d),$$

où c^* est la densité de copule de $(X_1, \ldots, X_d)$.

En remplaçant dans l'expression précédente, on obtient :

$$P(Y \leq y \mid X = x) = \frac{\int_{-\infty}^{y} c\big(F_0(t), F_1(x_1), \ldots, F_d(x_d)\big) f_0(t) f_1(x_1) \ldots f_d(x_d)\, dt}{c^*\big(F_1(x_1), \ldots, F_d(x_d)\big) f_1(x_1) \ldots f_d(x_d)}.$$

En considérant le changement de variables suivant :

$$u = F_0(t), \quad \text{(d'où} \quad du = f_0(t)\, dt),$$

on obtient :

$$P(Y \leq y \mid X = x) = \frac{\int_0^{F_0(y)} c\big(u, F_1(x_1), \ldots, F_d(x_d)\big)\, du}{c_{\mathbf{X}}\big(F_1(x_1), \ldots, F_d(x_d)\big)}.$$

Par définition de la densité d'une copule, on a :

$$c(u, v_1, \ldots, v_d) = \frac{\partial^{d+1} C(u, v_1, \ldots, v_d)}{\partial u\, \partial v_1 \ldots \partial v_d}.$$

En intégrant cette expression, il vient :

$$\begin{aligned}\int_0^{F_0(y)} C\big(u, F_1(x_1), \ldots, F_d(x_d)\big)du &= \int_0^{F_0(y)} \frac{\partial^{d+1} C\big(u, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial u\, \partial F_1(x_1) \ldots \partial F_d(x_d)}\, du \\ &= \frac{\partial^d C\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)} - \frac{\partial^d C\big(0, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)} \\ &= \frac{\partial^d C\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)}.\end{aligned}$$

Or, par définition d'une copule :

$$\frac{\partial^d C\big(0, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)} = 0.$$

D'autre part la densité de la copule de $\mathbf{X}$ s'écrit :

$$\begin{aligned}c_{\mathbf{X}}(F_1(x_1), \ldots, F_d(x_d)) &= \int_0^1 c\big(u, F_1(x_1), \ldots, F_d(x_d)\big) \\ &= \int_0^1 \frac{\partial^{d+1} C^*\big(u, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial u\ \partial F_1(x_1) \ldots \partial F_d(x_d)}\, du \\ &= \int_0^1 \frac{\partial^d C^*\big(u, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)}\, du \\ &= \frac{\partial^d C^*\big(1, F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)}.\end{aligned}$$

En combinant ces deux résultats, on obtient :

$$\begin{aligned}P(Y \leq y \mid \mathbf{X} = \mathbf{x}) &= \frac{\partial^d C\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big)}{\partial F_1(x_1) \ldots \partial F_d(x_d)} \frac{\partial F_1(x_1) \ldots \partial F_d(x_d)}{\partial^d C^*\big(1, F_1(x_1), \ldots, F_d(x_d)\big)} \\ &= \frac{\partial^d C\big(F_0(y), F_1(x_1), \ldots, F_d(x_d)\big)}{\partial^d C^*\big(1, F_1(x_1), \ldots, F_d(x_d)\big)}.\end{aligned}$$

La proposition 2.2 établi le lien entre la fonction de répartition conditionnelle de Y sachant $\mathbf{X} = \mathbf{x}$ et la copule du vecteur $(Y, \mathbf{X})$

2.3.1 Classification supervisée basée sur les copules

Dans le travail de [MBB23], les auteurs ont étudié un modèle de régression où la variable réponse Y est binaire ($\in \{0, 1\}$), en fonction de d covariables explicatives. Pour ce faire,

ils ont proposé une fonction de liaison paramétrique fondée sur une copule, permettant de modéliser la probabilité prédictive de succès (*Predictive Probability of Success*), définie par :

$$\Pi(\boldsymbol{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d.$$

Cette approche repose sur la représentation suivante de la fonction de répartition conditionnelle :

$$\mathbb{P}(Y \leq y \mid \mathbf{X} = \mathbf{x}) = C^c \left\{ F_0(y) \mid F(\boldsymbol{x}) \right\}, \quad (y, \boldsymbol{x}) \in \{0, 1\} \times \mathbb{R}^d.$$

La probabilité prédictive de succès peut alors s'exprimer de la manière suivante :

$$\Pi(\boldsymbol{x}) = 1 - \mathbb{P}(Y \leq 0 \mid \boldsymbol{x} = \boldsymbol{x}) = 1 - C^c \left\{ 1 - p \mid F(\boldsymbol{x}) \right\}. \tag{2.3}$$

En utilisant (2.3), une nouvelle classe de fonctions a été développée par [MBB23], en s'appuyant sur les familles de copules les plus couramment utilisées, notamment les copules archimédiennes et elliptiques.

Modèle de fonction de lien fondé sur les copules archimédiennes

La famille des copules archimédiennes constitue une classe importante de copules, et elle est définit par l'expression suivante :

$$C_\theta(u, \boldsymbol{v}) = \varphi_\theta^{-1} \left(\varphi_\theta(u) + \sum_{i=1}^{d} \varphi_\theta(v_i) \right),$$

où $\theta \in \Theta$ est un vecteur de paramètres de dépendance, et φ_θ est le générateur associé à la copule.

La distribution conditionnelle associée, C^c, de cette copule a la forme suivante :

$$C^c(u \mid \boldsymbol{v}) = \frac{\psi_\theta^{(d)} \left[\varphi_\theta(u) + \varphi_\theta(v_1) + \ldots + \varphi_\theta(v_d) \right]}{\psi_\theta^{(d)} \left[\varphi_\theta(v_1) + \ldots + \varphi_\theta(v_d) \right]},$$

où ψ_θ, l'inverse de φ_θ, est différentiable jusqu'à l'ordre d.

En effet, en utilisant l'équation (2.2).

$$C^c(u \mid \boldsymbol{v}) = \frac{\partial_{v_1}\partial_{v_2}\ldots\partial_{v_d}C(u,v_1,\ldots,v_d)}{\partial_{v_1}\partial_{v_2}\ldots\partial_{v_d}C(1,v_1,\ldots,v_d)}.$$

D'une part on a :

$$\begin{aligned}\frac{\partial v_1\ldots\partial v_d C(u,v_1,\ldots,v_d)}{\partial v_1\ldots\partial v_d} &= \frac{\partial v_1\ldots\partial v_d}{\partial v_1\ldots\partial v_d}\left[\varphi_\theta^{-1}\left(\varphi_\theta(u)+\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\right]\\ &= \left(\varphi_\theta^{-1}\right)^d\left(\varphi_\theta(u)+\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\times\prod_{i=1}^{d}\varphi_\theta'(v_i)\\ &= \psi_\theta^{(d)}\left(\varphi_\theta(u)+\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\times\prod_{i=1}^{d}\varphi_\theta'(v_i). \qquad (2.4)\end{aligned}$$

D'autre part on a :

$$\begin{aligned}\frac{\partial_{v_1}\ldots\partial_{v_d}}{\partial_{v_1}\ldots\partial_{v_d}}\left(C(1,v_1,\ldots,v_d)\right) &= \frac{\partial_{v_1}\ldots\partial_{v_d}}{\partial_{v_1}\ldots\partial_{v_d}}\left(\varphi_\theta^{-1}\left(\varphi_\theta(1)+\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\right)\\ &= \frac{\partial_{v_1}\ldots\partial_{v_d}}{\partial_{v_1}\ldots\partial_{v_d}}\left(\varphi_\theta^{-1}\left(\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\right),\ \varphi_\theta(1)=0\\ &= \psi_\theta^{(d)}\left(\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\times\prod_{i=1}^{d}\varphi_\theta'(v_i). \qquad (2.5)\end{aligned}$$

En utilisant (2.4) et (2.5), on obtient :

$$\begin{aligned}C^c(u \mid \boldsymbol{v}) &= \frac{\psi_\theta^{(d)}\left(\varphi_\theta(u)+\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\times\prod_{i=1}^{d}\varphi_\theta'(v_i)}{\psi_\theta^{(d)}\left(\sum_{i=1}^{d}\varphi_\theta(v_i)\right)\times\prod_{i=1}^{d}\varphi_\theta'(v_i)}\\ &= \frac{\psi_\theta^{(d)}\left(\varphi_\theta(u)+\varphi_\theta(v_1)+\cdots+\varphi_\theta(v_d)\right)}{\psi_\theta^{(d)}\left(\varphi_\theta(v_1)+\cdots+\varphi_\theta(v_d)\right)}.\end{aligned}$$

La probabilité prédictive de succès est donnée par :

$$\Pi(\boldsymbol{x}) = 1 - \frac{\psi_\theta^{(d)}\left[\varphi_\theta(1-p)+\sum_{i=1}^{d}\varphi_\theta(F_i(x_i))\right]}{\psi_\theta^{(d)}\left[\sum_{i=1}^{d}\varphi_\theta(F_i(x_i))\right]}.$$

Pour des illustrations concrètes de cette probabilité prédictive appliquée à différents copules archimédiennes, on pourra se référer aux travaux menés par [MBB23]. Plusieurs exemples détaillés mettant en œuvre ces fonctions de liaison dans divers contextes de modélisation ont été illustrés dans [MBB23].

Modèle de fonction de lien fondé sur les copules elliptiques

Les copules elliptiques constituent une autre classe importante de copules dérivées de distributions elliptiques telles que la distribution normale multivariée et la distribution de Student à plusieurs dimensions. Parmi les copules les plus représentatives de cette famille, on retrouve notamment la copule gaussienne et la copule de student (ou t-copule). Soit Φ la fonction de répartition normale multivariée standard de dimension $d+1$, avec une matrice de corrélation donnée par :

$$\rho = \begin{pmatrix} 1 & \rho_1 \\ \rho_1^\top & R \end{pmatrix},$$

où $\rho_1 = (\rho_{0,1}, \ldots, \rho_{0,d})$ désigne le vecteur des corrélations entre Y et $\mathbf{X}$, et $\rho = (\rho_{ij})$ pour $i, j = 1, \ldots, d$, est la matrice de corrélation des covariables.

La copule gaussienne est définie comme suit :

$$C(u, \boldsymbol{v}; \rho) = \Phi_\rho \left(\Phi^{-1}(u), \Phi^{-1}(\boldsymbol{v}); \rho \right),$$

où

$$\Phi^{-1}(\boldsymbol{v}) = \left(\Phi^{-1}(v_1), \ldots, \Phi^{-1}(v_d) \right)^\top.$$

Ainsi, pour tout $(u, \boldsymbol{v}) \in [0,1]^{d+1}$, la distribution conditionnelle est donnée par :

$$C^c(u \mid \boldsymbol{v}; \rho) = \Phi \left(\frac{\Phi^{-1}(u) - \rho_1 R^{-1} \Phi^{-1}(\boldsymbol{v})}{\sqrt{1 - \rho_1 R^{-1} \rho_1^\top}} \right). \tag{2.6}$$

Lorsque Y et $\mathbf{X}$ sont liés par une copule gaussienne, la fonction de lien associée à cette copule peut être directement déduite des équations (2.3) et (2.6) de la manière suivante :

$$\pi(\boldsymbol{x}) = \Phi \left(\frac{\rho_1 R^{-1} \Phi^{-1}(F(\boldsymbol{x})) - \Phi^{-1}(1-p)}{\sqrt{1 - \rho_1 R^{-1} \rho_1^\top}} \right). \tag{2.7}$$

Dans le cas particulier où $d = 1$, l'équation (2.7) devient :

$$\pi(x_1) = \Phi \left(\frac{1}{\sqrt{1-\rho^2}} \left[\rho \Phi^{-1}(F_1(x_1)) - \Phi^{-1}(1-p) \right] \right), \tag{2.8}$$

où ρ représente le coefficient de corrélation entre Y et X_1.

CHAPITRE 3

Simulations

3.1 Introduction

Dans ce chapitre, nous présentons une étude de simulation ayant pour objectif d'évaluer les performances de différents classifieurs appliqués à des données simulées, en utilisant déférents scénarios de génération de données (*Data Generating Process*, DGP).
D'une part, les données ont été générées selon six familles de copules, chacune représentant une structure de dépendance particulière. D'autre part, deux modèles classiques (logistique et probit) ont également été utilisés pour générer des données, servant de références comparatives.

3.2 Modèles de génération de données

Les différents scénarios de génération de données sont décrits comme suit :

1. **DGP1 :** la copule gaussienne avec $Y \sim \text{Bernoulli}(p)$ et $X \sim N(0,1)$.
2. **DGP2 :** la copule t-Student avec $df = 5$ degré de liberté. La variable réponse $Y \sim \text{Bernoulli}(p)$ et la variable explicative $X \sim N(0,1)$.

3. **DGP3 :** la copule de Clayton avec $Y \sim \text{Bernoulli}(p)$ et $X \sim N(0,1)$.
4. **DGP4 :** la copule de Gumbel avec $Y \sim \text{Bernoulli}(p)$ et $X \sim N(0,1)$.
5. **DGP5 :** la copule de Frank avec $Y \sim \text{Bernoulli}(p)$ et $X \sim N(0,1)$.
6. **DGP6 :** la copule de Joe avec $Y \sim \text{Bernoulli}(p)$ et $X \sim N(0,1)$.
7. **DGP7 :** la variable cible $Y \sim \text{Bernoulli}(p)$ est générée par une régression logistique, avec $X \sim N(0,1)$, selon la relation suivante :

$$\mathbb{P}(Y = 1 \mid X = x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x))}.$$

   On considère un modèle équilibré avec $p = 0{,}5$, ce qui implique que $\beta_0 = \log\left(\frac{p}{1-p}\right) = 0$. Par ailleurs, on choisit deux valeurs pour β_1, à savoir 0,5 et 3, afin de représenter respectivement une faible et une forte dépendance entre X et Y.

8. **DGP8 :** La variable cible $Y \sim \text{Bernoulli}(p)$ est générée selon un modèle Probit, avec $X \sim N(0,1)$, selon la relation suivante :

$$\mathbb{P}(Y = 1 \mid X = x) = \Phi(\beta_0 + \beta_1 x),$$

   où Φ désigne la fonction de répartition de la loi normale standard. On considère un modèle équilibré avec $p = 0{,}5$, ce qui implique que $\beta_0 = \Phi^{-1}(p) = 0$. On choisit deux valeurs pour β_1, à savoir $0,1$ et 3, afin de représenter respectivement une faible et une forte dépendance entre X et Y.

Pour chaque scénario de simulation (copule ou modèle logistique) des jeux de données ont été générés avec une taille d'échantillon $n = 100$. On utilise $n_1 = 80$ observations pour entraîner le modèle et $n_2 = 20$ pour mesurer la performance de chaque méthode de classification. Dans le cas des copules, on considère deux niveaux de dépendance mesurés par le coefficient de Kendall ($\tau = 0.25$ et $\tau = 0.5$) et la probabilité de succès est fixée à $p = 0.6$. L'évaluation porte sur plusieurs classifieurs présentés au chapitre 1, tels que la régression logistique, les machines à vecteurs de support (SVM), les réseaux de

neurones, les k plus proches voisins (k-NN), l'analyse discriminante linéaire (LDA), les forêts aléatoires, le bagging et le boosting, ainsi que sur les copules abordées au chapitre 2. Pour garantir la robustesse des résultats, chaque configuration a été répétée $R = 500$ fois. À partir de ces simulations, quatre mesures de performance ont été considérées : le score F_1, l'exactitude (accuracy), la précision positive (precision) et la sensibilité (recall), voir le chapitre 1 pour les définitions de ces coefficients. Pour chaque combinaison de DGP et de classifieur, nous avons calculé la moyenne, la médiane et l'écart-type de ces indicateurs sur les $R = 500$ répétitions, et les résultats sont présentés sous forme de tableaux comparatifs.

3.3 Résultats pour les modèles DGP1 à DGP6

Dans cette section, nous étudions la performance des classifieurs lorsque les données sont générées à partir d'un modèle de copule. Les résultats obtenus sont présentés sous forme de tableaux.

Les deux tableaux 3.1 et 3.2 présentent une comparaison des performances moyennes des différentes méthodes pour les DGPs 1 à 6 avec un coefficient de dépendance $\tau = 0.25$. Globalement, la copule est le classifieur le plus performant pour tous les DGPs, atteignant des F1-Scores et des précisions significativement plus élevés que les autres classifieurs. Par exemple, pour le DGP1, la copule atteint un F1-Score de 0.8328 et une précision de 0.8799, ce qui la place nettement au-dessus des autres méthodes. Cependant, d'autres modèles comme Boosting montrent des performances intéressantes, surtout pour les DGPs 4 et 5 où le F1-Score atteint jusqu'à 0.5262 et la précision monte à 0.6299. Les réseaux de neurones et le SVM présentent des performances plus faibles en terme de sensibilité, bien qu'ils soient compétitifs en précision. Les réseaux de neurones, en particulier, ont des résultats moins robustes pour la sensibilité (par exemple : 0.3767 pour le DGP1). La régression logistique et les KNN restent des clasifieurs forts mais ne rivalisent pas avec les copules en terme de F1-Score et sensibilité. Les modèles CART et forêts aléatoires et le modèle de mélange montrent des performances globales relativement faibles, en

particulier en sensibilité, indiquant leurs difficultés à capturer les dépendances complexes dans les données générées par les copules. En résumé, la copule est le modèle le plus efficace pour $\tau = 0.25$, mais des méthodes comme Boosting, régression logistique, et les réseaux de neurones offrent des alternatives intéressantes.

Les résultats présentés dans les tableaux 3.3 et 3.4 comparent les performances médianes des différents classifieurs pour un échantillon de taille $n = 100$ et un coefficient de dépendance $\tau = 0.25$, selon six générateurs de données (DGP1 à DGP6). On observe que la méthode fondée sur les copules se distingue systématiquement par ses performances élevées. Elle affiche des F1-scores supérieurs à 0.82, une précision et une sensibilité remarquables, et une exactitude proche de 0.89. À l'inverse, les classifieurs classiques tels que la régression logistique, SVM, KNN ou encore les réseaux de neurones montrent des performances plus limitées, avec des F1 scores souvent inférieurs à 0.55 et une sensibilité généralement en dessous de 0.45. Bien que les méthodes comme le modèle de mélange, LDA, boosting ou bagging offrent des résultats relativement bonnes, elles demeurent globalement moins performantes que l'approche copule. Il convient également de souligner que la précision est souvent supérieure à la sensibilité pour plusieurs méthodes, traduisant une certaine difficulté à identifier correctement les classes minoritaires. En résumé, la méthode basée sur les copules s'impose comme la meilleurs dans ce cadre de simulation.

Les tableaux 3.5 et 3.6 montrent les écarts-types des mesures de performance (F1 Score, accuracy, précision et sensibilité). Les résultats dans ces tableaux révèlent que la méthode basée sur les copules est la plus stable, avec les plus faibles variations observées sur l'ensemble des modèles DGP1 à DGP6. À l'opposé, les classifieurs classiques ; tels que le réseau de neurones, le SVM et la régression logistique, présentent des écarts-types plus élevés, traduisant une variabilité importante de leurs performances d'une simulation à l'autre. Cette instabilité est particulièrement marquée pour la précision et la sensibilité, surtout dans les modèles DGP4 à DGP6. De plus, les méthodes d'ensemble comme le bagging, le boosting et la forêt aléatoire affichent une variabilité intermédiaire, plus pro-

noncée que la copule mais généralement inférieure aux méthodes linéaires. En résumé, ces résultats confirment que la méthode copule est meilleure grâce à sa stabilité.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP 1	Copule	**0.8327646**	**0.8799263**	**0.8482833**	**0.8491793**
	Réseau de neurones	0.491147	0.6335789	0.570176	0.3767154
	SVM	0.4562296	0.6466158	0.5823589	0.3134949
	Régression logistique	0.485704	0.5930579	0.5995153	0.3620529
	KNN	0.4888514	0.6398947	0.501216	0.4350536
	Modèle de mélange	0.4882752	0.6474474	0.5839426	0.3688132
	LDA	0.4848764	0.5715947	0.6027599	0.3584934
	Forêt aléatoire	0.4798832	0.6051684	0.4743376	0.4616077
	CART	0.4881958	0.5723211	0.5216141	0.4355481
	Bagging	0.4798313	0.6300842	0.4733055	0.4621260
	Boosting	0.5148633	0.5100241	0.5547819	0.4420021
DGP 2	Copule	**0.8351978**	**0.8811526**	**0.8516121**	**0.8470695**
	Réseau de neurones	0.4965328	0.6427947	0.5812491	0.3791982
	SVM	0.4698593	0.6475474	0.6159358	0.3230609
	Régression logistique	0.5005889	0.6553632	0.6097866	0.3645744
	KNN	0.4781854	0.5890474	0.4926707	0.4271693
	Modèle de mélange	0.4967485	0.6474895	0.5995485	0.3633712
	LDA	0.4964432	0.6547579	0.6119325	0.3577331
	Forêt aléatoire	0.4811585	0.5697263	0.4727706	0.4643071
	CART	0.4998492	0.6135368	0.5390357	0.4374410
	Bagging	0.4820633	0.5695421	0.4736239	0.4637992
	Boosting	0.5181956	0.6357842	0.5679451	0.4454297
DGP 3	Copule	**0.8225163**	**0.8706737**	**0.8431566**	**0.8288475**
	Réseau de neurones	0.5154339	0.6617895	0.6374598	0.4110144
	SVM	0.4818536	0.6627368	0.6617488	0.3515137
	Régression logistique	0.5214394	0.6681474	0.6458513	0.4073605
	KNN	0.5125942	0.6155421	0.5346957	0.4696111
	Modèle de mélange	0.5126713	0.6668842	0.6445363	0.4042082
	LDA	0.5188929	0.6686737	0.6496671	0.4014136
	Forêt aléatoire	0.4861605	0.5795368	0.4766805	0.4717193
	CART	0.5150443	0.6287684	0.5607592	0.4674280
	Bagging	0.4857803	0.5792105	0.4752026	0.4716764
	Boosting	0.5344278	0.6541684	0.6158396	0.4558048

TABLEAU 3.1 – Comparaison des performances moyennes des méthodes pour les DGP 1 à 3 avec $n = 100, \tau = 0.25$.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0.8463929**	**0.8861842**	**0.8544038**	**0.8609087**
	Réseau de neurones	0.4885889	0.6178421	0.5487675	0.3815333
	SVM	0.4610685	0.6250895	0.5671870	0.3143397
	Régression logistique	0.4751475	0.6382316	0.5691953	0.3477595
	KNN	0.4867190	0.5938895	0.4941440	0.4524183
	Modèle de mélange	0.4857158	0.6295368	0.5564284	0.3628833
	LDA	0.4669912	0.6359263	0.5659419	0.3382437
	Forêt aléatoire	0.4675399	0.5650632	0.4611398	0.4532460
	CART	0.4915881	0.5974789	0.5105934	0.4435556
	Bagging	0.4675380	0.5655474	0.4602331	0.4536984
	Boosting	0.5121179	0.6187579	0.5432837	0.4445159
DGP5	Copule	**0.8321185**	**0.8768000**	**0.8481636**	**0.8450659**
	Réseau de neurones	0.5146209	0.6403368	0.5658508	0.4189214
	SVM	0.4906172	0.6409579	0.5849293	0.3615897
	Régression logistique	0.4987932	0.6482632	0.5911868	0.3683825
	KNN	0.4768325	0.5925000	0.4981634	0.4360690
	Modèle de mélange	0.5113397	0.6455211	0.5801045	0.3849381
	LDA	0.4958790	0.6478737	0.5890794	0.3631365
	Forêt aléatoire	0.4630999	0.5664895	0.4692122	0.4399778
	CART	0.4981459	0.6062579	0.5195075	0.4559698
	Bagging	0.4633771	0.5673368	0.4690671	0.4405532
	Boosting	0.5261925	0.6299474	0.5557172	0.4665063
DGP6	Copule	**0.8506192**	**0.8923526**	**0.8506680**	**0.8736753**
	Réseau de neurones	0.5119575	0.6167947	0.5393494	0.3841236
	SVM	0.4827983	0.6217211	0.5612360	0.3131561
	Régression logistique	0.4904246	0.6349053	0.5711263	0.3306411
	KNN	0.4902237	0.5858263	0.4876752	0.4327639
	Modèle de mélange	0.5094007	0.6270000	0.5430060	0.3637291
	LDA	0.4855119	0.6344737	0.5700005	0.3259704
	Forêt aléatoire	0.4849924	0.5673895	0.4743486	0.4528963
	CART	0.4925308	0.5887105	0.4980591	0.4320711
	Bagging	0.4849505	0.5669789	0.4745520	0.4508924
	Boosting	0.5261291	0.6075158	0.5272857	0.4431760

TABLEAU 3.2 – Comparaison des performances moyennes des méthodes pour les DGP 4 à 6 avec $n = 100, \tau = 0.25$.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP1	Copule	**0,8333333**	**0,8947368**	**0,8571429**	**0,8571429**
	Réseau de neurones	0,5	0,6315789	0,5714286	0,375
	SVM	0,4615385	0,6315789	0,5916667	0,2857143
	Régression Logistique	0,5	0,65	0,6	0,375
	KNN	0,5	0,6	0,5	0,4285714
	Modèle de Mélange	0,5	0,6315789	0,5714286	0,375
	LDA	0,5	0,65	0,6	0,375
	Forêt Aléatoire	0,5	0,5789474	0,5	0,5
	CART	0,5	0,6	0,5	0,4285714
	Bagging	0,5	0,5789474	0,5	0,5
	Boosting	0,5263158	0,6315789	0,5454545	0,4285714
DGP2	Copule	**0,8421053**	**0,8947368**	0**,8571429**	**0,8571429**
	Réseau de neurones	0,5	0,6315789	0,5714286	0,375
	SVM	0,4615385	0,65	0,6	0,2857143
	Régression Logistique	0,5	0,65	0,6153846	0,375
	KNN	0,4705882	0,5789474	0,5	0,4285714
	Modèle Mélangé	0,5	0,6315789	0,6	0,375
	LDA	0,5	0,65	0,625	0,375
	Forêt Aléatoire	0,5	0,5789474	0,5	0,4772727
	CART	0,5	0,6315789	0,5	0,4285714
	Bagging	0,5	0,5789474	0,5	0,5
	Boosting	0,5333333	0,6315789	0,5625	0,4444444
DGP3	Copule	**0,8235294**	**0,8947368**	**0,8571429**	**0,8571429**
	Réseau de neurones	0,5333333	0,6842105	0,625	0,4285714
	SVM	0,5	0,6842105	0,6666667	0,375
	Régression Logistique	0,5454545	0,6842105	0,6666667	0,4285714
	KNN	0,5263158	0,6315789	0,5	0,4494949
	Modèle de Mélange	0,5333333	0,6842105	0,6666667	0,4285714
	LDA	0,5454545	0,6842105	0,6666667	0,4285714
	Forêt Aléatoire	0,5	0,5789474	0,5	0,5
	CART	0,5333333	0,6315789	0,5555556	0,5
	Bagging	0,5	0,5789474	0,5	0,5
	Boosting	0,5454545	0,6671053	0,6	0,4444444

TABLEAU 3.3 – Comparaison des performances médianes des différentes méthodes pour $n = 100, \tau = 0.25$ et les DGP 1-3.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0,8571429**	**0,8947368**	**0,8571429**	**0,875**
	Réseau de neurones	0,5	0,6315789	0,5454545	0,375
	SVM	0,4615385	0,6315789	0,5555556	0,2857143
	Régression logistique	0,4705882	0,6315789	0,6	0,3541667
	KNN	0,5	0,5789474	0,5	0,4444444
	Modèle de Mélange	0,4705882	0,6315789	0,5555556	0,375
	LDA	0,4615385	0,6315789	0,5714286	0,3333333
	Foret aléatoire	0,4705882	0,5789474	0,4444444	0,4444444
	CART	0,5	0,6315789	0,5	0,4365079
	Bagging	0,4705882	0,5789474	0,4444444	0,4444444
	Boosting	0,5333333	0,6315789	0,5454545	0,4444444
DGP5	Copule	**0,8421053**	**0,8947368**	**0,8571429**	**0,8660714**
	Réseau de neurones	0,5263158	0,6315789	0,5714286	0,4285714
	SVM	0,5	0,6315789	0,6	0,375
	Régression logistique	0,5	0,65	0,6	0,375
	kNN	0,4705882	0,5789474	0,5	0,4285714
	Modèle de Mélange	0,5333333	0,6315789	0,577381	0,4285714
	LDA	0,5	0,6315789	0,6	0,375
	Foret aléatoire	0,4705882	0,5789474	0,4833333	0,4365079
	CART	0,5	0,6	0,5	0,4722222
	Bagging	0,4705882	0,5789474	0,4833333	0,4444444
	Boosting	0,5333333	0,6315789	0,5555556	0,5
DGP6	Copule	**0,8571429**	**0,8947368**	**0,8571429**	**0,875**
	Réseau de neurones	0,5263158	0,6315789	0,5555556	0,375
	SVM	0,5	0,6315789	0,5555556	0,2857143
	Régression logistique	0,5	0,6315789	0,5714286	0,3333333
	kNN	0,5	0,5789474	0,5	0,4285714
	Modèle de Mélange	0,5263158	0,6315789	0,5454545	0,375
	LDA	0,5	0,6315789	0,5714286	0,3333333
	Forêt Aléatoire	0,5	0,5789474	0,5	0,4444444
	CART	0,5	0,5789474	0,5	0,4285714
	Bagging	0,5	0,5789474	0,5	0,4444444
	Boosting	0,5333333	0,6315789	0,5384615	0,4444444

TABLEAU 3.4 – Comparaison des performances médianes des différentes méthodes pour $n = 100, \tau = 0.25$ et les DGP 4-6.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP1	Copule	**0,08825155**	**0,06833467**	**0,105035**	**0,116158**
	Réseau de neurones	0,15924554	0,10604637	0,2302397	0,2366641
	SVM	0,17137902	0,09468405	0,2652417	0,2541282
	Régression logistique	0,14577	0,09280822	0,1540243	0,1955851
	KNN	0,1508568	0,10004451	0,1601843	0,1942063
	Modèle de Mélange	0,1437471	0,09209573	0,1525788	0,1968533
	LDA	0,1490159	0,09218348	0,1567453	0,1978749
	Forêt Aléatoire	0,1503316	0,10544309	0,15944	0,1878818
	CART	0,1493513	0,09959517	0,1680651	0,2127736
	Bagging	0,1496211	0,10638929	0,1605486	0,1863048
	Boosting	0,1471314	0,09862178	0,1689809	0,2062174
DGP2	Copule	**0,1040537**	**0,07715546**	**0,120582**	**0,1290707**
	Réseau de neurones	0,1450549	0,09249193	0,1658449	0,2074528
	SVM	0,1503566	0,09218649	0,1730162	0,2113428
	Régression logistique	0,1324253	0,09157385	0,15773	0,1884614
	KNN	0,1464044	0,10828316	0,1655341	0,2039663
	Modèle de Mélange	0,1368391	0,09203781	0,1536918	0,1909236
	LDA	0,1326785	0,09101158	0,158566	0,1867612
	Forêt Aléatoire	0,1439322	0,11087486	0,1628964	0,1953051
	CART	0,1469497	0,10138208	0,1629844	0,2115049
	Bagging	0,1424074	0,10981352	0,1620194	0,1946604
	Boosting	0,1363357	0,09231107	0,1539942	0,2020369
DGP3	Copule	**0,09900176**	**0,07113861**	**0,1220622**	**0,1297627**
	Réseau de neurones	0,1402004	0,08774651	0,1511514	0,186109
	SVM	0,14510764	0,08605307	0,1498522	0,1894994
	Régression logistique	0,13343917	0,0888632	0,1387687	0,1754764
	KNN	0,14333038	0,0994984	0,1569045	0,1825769
	Modèle de Mélange	0,13609225	0,08874744	0,1426155	0,1783317
	LDA	0,13611606	0,08944484	0,1425564	0,1775545
	Forêt Aléatoire	0,14399279	0,10682165	0,1598207	0,1775047
	CART	0,14540414	0,0935232	0,161534	0,1892051
	Bagging	0,14368054	0,10687638	0,1613635	0,1762549
	Boosting	0,13524226	0,08713268	0,1519963	0,1833782

TABLEAU 3.5 – Comparaison des performances des Écart-type des différentes méthodes pour $n = 100, \tau = 0.25$ et les DGP 1-3.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0,0876553**	**0,068833467**	**0,1025454**	**0,1165184**
	Réseau de neurones	0,1517191	0,10321885	0,2052381	0,2347069
	SVM	0,1642176	0,0923269	0,2517462	0,2572854
	Régression logistique	0,1604399	0,09635643	0,2251881	0,2203893
	KNN	0,1490015	0,10797215	0,1643215	0,1911507
	Modèle de Mélange	0,1593496	0,09902551	0,2096995	0,2430007
	LDA	0,1612971	0,09498873	0,2297578	0,2206889
	Forêt Aléatoire	0,1467244	0,10916052	0,161913	0,1855896
	CART	0,1552884	0,12016212	0,1968792	0,2149786
	Bagging	0,1483316	0,11019381	0,1633022	0,1874704
	Boosting	0,153069	0,11076061	0,1916346	0,2294312
DGP5	Copule	**0,1040537**	**0,07715546**	**0,120582**	**0,1290707**
	Réseau de neurones	0,1450549	0,09249193	0,1658449	0,2074528
	SVM	0,1503566	0,09218649	0,1730162	0,2113428
	Régression logistique	0,1324253	0,09157385	0,15773	0,1884614
	KNN	0,1464044	0,10828316	0,1655341	0,2039663
	Modèle de Mélange	0,1368391	0,09203781	0,1536918	0,1909236
	LDA	0,1326785	0,09101158	0,158566	0,1867612
	Forêt Aléatoire	0,1439322	0,11087486	0,1628964	0,1953051
	CART	0,1469497	0,10138208	0,1629844	0,2115049
	Bagging	0,1424074	0,10981352	0,1620194	0,1946604
	Boosting	0,1363357	0,09231107	0,1539942	0,2020369
DGP6	Copule	**0,09900176**	**0,07113861**	**0,1220622**	**0,1297627**
	Réseau de neurones	0,1402004	0,08774651	0,1511514	0,186109
	SVM	0,14510764	0,08605307	0,1498522	0,1894994
	Régression logistique	0,13343917	0,0888632	0,1387687	0,1754764
	KNN	0,14333038	0,0994984	0,1569045	0,1825769
	Modèle de Mélange	0,13609225	0,08874744	0,1426155	0,1783317
	LDA	0,13611606	0,08944484	0,1425564	0,1775545
	Forêt Aléatoire	0,14399279	0,10682165	0,1598207	0,1775047
	CART	0,14540414	0,0935232	0,161534	0,1892051
	Bagging	0,14368054	0,10687638	0,1613635	0,1762549
	Boosting	0,13524226	0,08713268	0,1519963	0,1833782

TABLEAU 3.6 – Comparaison des performances Écart-type des différentes méthodes pour $n = 100, \tau = 0.25$ et les DGP 4-6.

Les tableaux 3.7 et 3.8 présentent les performances moyennes des différents classifieurs sur les modèles générateurs de données DGP1 à DGP6 avec une taille d'échantillon $n = 100$ et un coefficient de dépendance $\tau = 0.5$.

De manière générale, on constate que les méthodes classiques de classification supervisée telles que la régression logistique, le SVM, le LDA ainsi que les méthodes d'ensemble (notamment le boosting) affichent des bonnes performances sur l'ensemble des scénarios. Par exemple, la régression logistique et le LDA atteignent des scores F1-score supérieurs à (0.72) pour le DGP3, avec des exactitude(accuracy) avoisinant les (0.80). Le modèle copule, en revanche, obtient un comportement contrasté, il produit des bons résultats dans certains cas (par exemple DGP2 avec un F1-score de 0.6855). En ce qui concerne les méthodes d'ensemble, le boosting apparaît comme un bon compromis, affichant de bonnes performances tant en terme de précision que de sensibilité. À l'inverse, le bagging et les forêts aléatoires sont moins performants globalement.

Les tableaux 3.9 et 3.10 exposent les résultats des performances médianes des classifieurs pour $n = 100$ et $\tau = 0.5$ à travers les modèles générateurs de données DGP1 à DGP6. Dans l'ensemble, on observe des bonnes performances pour plusieurs classifieurs, notamment la régression logistique, le SVM, le LDA et les modèles de mélange, qui maintiennent des scores F1, d'accuracy, de précision et de sensibilité relativement élevés et équilibrés. En particulier, pour les DGP3, DGP5 et DGP6, ces méthodes affichent systématiquement des médianes proches de (0.75) en F1-score, indiquant leur robustesse dans des contextes variés de dépendance. Le modèle copule, bien qu'il montre des performances plus modestes dans les DGPs initiaux, parvient à se distinguer dans certains cas, notamment dans le DGP4 où il obtient le F1-score le plus élevé (0.7059). Les classifieurs tels que les forêts aléatoires, le bagging et parfois le CART, se positionnent en retrait, avec des F1-score et des précisions globales (accuracy) régulièrement en dessous de (0.70). Enfin, le boosting maintient des performances solides et constantes, en particulier dans le DGP5.

Les résultats des tableaux 3.11 et 3.12, montrent que les classifieurs classiques comme la régression logistique, le SVM, et le LDA présentent une stabilité relativement satisfaisante avec des écarts-types modérés. En revanche, les méthodes basées sur les copules affichent les plus faibles écarts-types, ce qui indique une certaine régularité dans leurs performances, bien que celles-ci soient généralement plus faibles en moyenne. Le bagging et les forêts aléatoires tendent à avoir des écarts-types légèrement plus élevés.

3.4 Résultats pour le modèle DGP7 pour deux valeurs de β_1

Les deux tableaux 3.13 et 3.14 illustrent les performances de différents classifieurs pour le modèle logistique, en utilisant deux valeurs du paramètre β_1 ; β_1 = 0,1 (faible corrélation) et β_1 = 3 (forte corrélation).

Pour β_1 = 0.1, la variable explicative exerce une faible influence sur la variable à prédire, rendant le problème de classification particulièrement difficile. On observe ainsi que la majorité des classifieurs affichent des performances de classification non-informative, avec des F1-scores, d'accuracy, de sensibilité et de précision proche de 0.5. Cette faible performance est également accompagnée de forts écarts-types, en particulier sur la sensibilité, ce qui reflète une instabilité importante de ces modèles.

Une exception notable est toutefois observée pour le classifieur basé sur les copules. Ce dernier se distingue nettement des autres en affichant des performances très élevées, avec un F1-score moyen de 0.95, une accuracy de 0.96, une sensibilité de 0.96 et une précision de 0.95. Les médianes confirment cette tendance, s'approchant de 1 pour la sensibilité et la précision, tandis que les écarts-types restent faibles. Cela indique que le modèle à copule est capable de capter des dépendances fines entre les variables.

Pour β_1 = 3, le lien entre la variable explicative et la réponse devient plus marqué, rendant le problème de classification plus simple. Dans ce cas, la plupart des classifieurs classiques

(régression logistique, SVM, LDA, réseau de neurones, etc.) voient leurs performances croître de manière significative. Les F1-scores moyens, d'accuracy, de sensibilité et de précision dépassent généralement 0.83 pour les meilleurs modèles, avec des écarts-types plus faibles, témoignant d'une plus grande robustesse. La régression logistique, en particulier, atteint un F1-score moyen de 0.84 et une précision de 0.85, ce qui la positionne parmi les meilleurs modèles dans ce contexte.

En revanche, le classifieur en utilsant les copules, très performant dans le cas précédent, montre ici des résultats en retrait. Ses F1-scores moyens sont nettement plus faibles (≈ 0.65), et ses écarts-types sont plus élevés. Cela suggère que le modèle copule perd de son avantage lorsque la relation entre les variables devient fortement corrélées et que d'autres modèles plus standards peuvent facilement capturer cette structure. Une mauvaise spécification de la copule dans ce cas peut expliquer les résultats de cette méthode de classification.

3.5 Résultats pour le modèle DGP8 pour deux valeurs de β_1

Les tableaux 3.15 et 3.16 présentent les performances des différents classifieurs évalués selon quatre métriques classiques : F1-score, accuracy, sensibilité et précision, pour deux valeurs de β_1 dans le modèle probit (0.1 et 3).

Pour $\beta_1 = 0.1$, la majorité des classifieurs affichent des scores proches de 0.5 pour l'ensemble des métriques. Cela indique que leurs performances sont équivalentes à celles d'un tirage aléatoire. En effet, à l'exception du classifieur basé sur les copules, qui obtient des performances nettement supérieures (F1-score moyen de 0.95), accuracy de 0.96, sensibilité et précision également élevées), tous les autres classifieurs, y compris les réseaux de neurones, SVM, régression logistique ou forêts aléatoires, présentent des résultats médiocres, avec des écarts-types parfois élevés.

Lorsque la valeur de $\beta_1 = 3$, les performances de l'ensemble des classifieurs s'améliorent. Tous les modèles, à l'exception de celui basé sur les copules, affichent des F1-scores, accuracy, sensibilité et précision supérieurs à 0.85, avec une dispersion réduite des résultats. La copule, en revanche, affiche des performances en diminution (F1-score moyen de 0.61, ce qui suggère qu'elle pourrait être moins adaptée dans des situations où la relation entre les variables devient plus marquée.

3.6 Conclusion

Ce chapitre a été consacré à l'étude comparative des performances de plusieurs méthodes de classification dans un cadre de données simulées. La structure de dépendance entre les variables explicatives et la variable d'intérêt est modélisée à l'aide de différentes familles de copules, ainsi que par des modèles classiques de régression logistique et probit. À travers une approche rigoureuse, nous avons examiné comment la nature du générateur de données (copule ou modèle paramétrique), la force de dépendance (via le coefficient de Kendall), et la taille d'échantillon influencent l'efficacité des classifieurs.

Les résultats ont mis en évidence que les méthodes traditionnelles telles que la régression logistique ou les SVM affichent de bonnes performances en terme de F1 Score et d'accuracy, notamment sous des modèles bien spécifiés. Cependant, les méthodes d'ensemble comme le boosting se sont montrées robustes dans plusieurs contextes, y compris lorsque la structure de dépendance est plus complexe (copules non gaussiennes). La méthode basée sur les copules représente une alternative intéressante à explorer.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP1	Copule	**0.6757416**	0.7492842	0.6984028	**0.6687143**
	Réseau de neurones	0.6634909	0.7548263	0.7292072	0.6386268
	SVM	0.6538155	0.7549632	**0.7441954**	0.6169206
	Régression logistique	0.6744451	**0.7624842**	0.7317565	0.6447563
	KNN	0.6178683	0.7119789	0.6520948	0.6091683
	Modèle de Mélange	0.6747677	**0.7602263**	0.7265458	0.6512706
	LDA	0.6731086	**0.7627789**	0.7340804	0.6407421
	Foret aléatoire	0.5871212	0.6811053	0.6053555	0.5910000
	CART	0.6430330	0.7315263	0.6858011	0.6340698
	Bagging	0.5874139	0.6804526	0.6042259	0.5915456
	Boosting	0.6627531	0.7478053	0.7126531	0.6478270
DGP2	Copule	**0.6855258**	0.7568789	0.7027915	**0.6813602**
	Réseau de neurones	0.6726338	0.7570316	0.7326498	0.6479792
	SVM	0.6639505	0.7572789	0.7433979	0.6247991
	Régression logistique	**0.6823959**	**0.7624789**	0.7404134	0.6503223
	KNN	0.6358793	0.7160684	0.6579277	0.6265952
	Modèle de Mélange	0.6777016	0.7578737	0.7290369	0.6545580
	LDA	**0.6832069**	**0.7638211**	**0.7443603**	0.6489014
	Foret aléatoire	0.5958542	0.6784526	0.6000938	0.5962387
	CART	0.6519433	0.7325211	0.6882981	0.6448388
	Bagging	0.5975470	0.6794158	0.6024052	0.5985221
	Boosting	0.6720205	0.7495263	0.7111623	0.6601991
DGP3	Copule	0.6447381	0.7315474	0.684051	0.6268528
	Réseau de neurones	0.7088316	**0.7953421**	0.8049874	0.6597587
	SVM	0.7045822	**0.7982789**	**0.8226904**	0.6441055
	Régression logistique	**0.7285688**	**0.7995316**	0.7788968	**0.7058904**
	KNN	0.679451	0.7614105	0.7286197	0.6599664
	Modèle de Mélange	0.7239646	**0.8000632**	0.7927997	0.6905142
	LDA	**0.7283547**	**0.8006211**	0.7833505	**0.7024294**
	Foret aléatoire	0.6404454	0.7183632	0.6551483	0.6478633
	CART	0.6937231	0.7777	0.7691419	0.6662579
	Bagging	0.6411985	0.7187	0.6560562	0.6486371
	Boosting	0.7094914	**0.7931474**	0.8004514	0.6683761

TABLEAU 3.7 – Comparaison des performances moyennes des différentes méthodes pour $n = 100, tau = 0.5$ et les DGP 1-3.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0,6924756**	**0,7589158**	**0,7092155**	**0,6906083**
	Réseau de neurones	0,6546840	0,7359579	0,6954246	0,6486607
	SVM	0,6475476	0,7366316	**0,7038127**	0,6300443
	Régression logistique	0,6505157	0,7437895	**0,7147419**	0,6203815
	KNN	0,6100363	0,6971737	0,6354337	0,6096893
	Modèle de Mélange	0,6590363	0,7421947	0,7021457	0,6447776
	LDA	0,6496118	0,7434842	**0,7154188**	0,6171879
	Foret aléatoire	0,5690364	0,6614842	0,5856711	0,5747382
	CART	0,6287127	0,7088053	0,6591802	0,6333217
	Bagging	0,5683368	0,6609474	0,5848888	0,5743136
	Boosting	0,6519014	0,7291421	0,6833024	0,6606212
DGP5	Copule	0,6636711	0,7402526	0,6830391	0,6592892
	Réseau de neurones	**0,6943140**	0,7693263	0,7326586	**0,6903918**
	SVM	**0,6914428**	**0,7714158**	0,7443719	0,6747219
	Régression logistique	**0,6933004**	**0,7771632**	**0,7532522**	0,6700149
	KNN	0,6476493	0,7305263	0,6753358	0,6484465
	Modèle de Mélange	**0,6913514**	**0,7723789**	0,7431481	0,6753680
	LDA	**0,6900189**	**0,7755053**	**0,7512176**	0,6655998
	Foret aléatoire	0,5980945	0,6880895	0,6156942	0,6063178
	CART	0,6774559	0,7519579	0,7069897	0,6831227
	Bagging	0,5982008	0,6891368	0,6171087	0,6052543
	Boosting	**0,6981526**	0,7684842	0,7268875	**0,700964**
DGP6	Copule	**0.7063889**	**0.7703526**	**0.716682**	**0.7113164**
	Réseau de neurones	0.6615946	0.7313263	0.6617276	0.6784070
	SVM	0.6569266	0.7322947	0.6694261	0.6636777
	Régression logistique	0.6326577	0.7330053	0.6978547	0.5901823
	KNN	0.6054953	0.6906421	0.6206628	0.6029982
	Modèle de Mélange	0.6468401	0.7283579	0.6690006	0.6395522
	LDA	0.6306031	0.7327000	0.6981145	0.5866657
	Foret aléatoire	0.5813256	0.6683368	0.5902561	0.5813577
	CART	0.6344207	0.7093526	0.6393682	0.6542927
	Bagging	0.5806027	0.6668789	0.5897153	0.5812109
	Boosting	0.6695226	0.7306632	0.6590824	0.6980196

TABLEAU 3.8 – Comparaison des performances moyennes des différentes méthodes pour $n = 100, tau = 0.5$ et les DGP 4-6.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP1	Copule	**0.6666667**	0.7368421	0.7	**0.6666667**
	Réseau de neurones	**0.6666667**	**0.75**	0.7142857	0.625
	SVM	**0.6666667**	0.7368421	**0.75**	0.625
	Régression logistique	**0.6666667**	**0.75**	**0.7142857**	**0.6666667**
	KNN	0.6315789	0.7368421	0.6666667	0.625
	Modèle de Mélange	**0.6666667**	**0.75**	0.7142857	**0.6666667**
	LDA	**0.6666667**	**0.75**	**0.7272727**	**0.6666667**
	Foret aléatoire	0.5882353	0.6842105	0.6	0.625
	CART	0.6666667	**0.7368421**	0.6666667	0.625
	Bagging	0.5882353	0.6842105	0.6	0.625
	Boosting	0.6666667	**0.7368421**	0.7	0.6666667
DGP2	Copule	**0.7**	0.7368421	0.7	0.6666667
	Réseau de neurones	**0.7**	0.75	0.7142857	0.6666667
	SVM	0.6666667	0.75	**0.75**	0.625
	Régression logistique	**0.7058824**	**0.7894737**	**0.75**	0.6666667
	KNN	0.6666667	0.7368421	0.6666667	0.6666667
	Modèle de Mélange	**0.7**	0.75	0.7272727	0.6666667
	LDA	**0.7058824**	**0.7894737**	0.75	0.6666667
	Foret aléatoire	0.6	0.6842105	0.6	0.5714286
	CART	0.6666667	0.7368421	0.6666667	0.6666667
	Bagging	0.6	0.6842105	0.6	0.6
	Boosting	0.6666667	0.7368421	0.7142857	0.6666667
DGP3	Copule	0.6666667	0.7368421	0.6666667	0.625
	Réseau de neurones	0.7142857	**0.7894737**	0.8	0.6666667
	SVM	0.7142857	**0.7894737**	**0.8333333**	0.6666667
	Régression logistique	**0.75**	**0.7894737**	0.7777778	**0.7142857**
	KNN	0.6978261	0.75	0.7142857	0.6666667
	Modèle de Mélange	0.7272727	**0.7894737**	0.8	**0.7142857**
	LDA	**0.75**	**0.7894737**	0.7777778	**0.7142857**
	Foret aléatoire	0.6666667	0.7368421	0.6666667	0.6666667
	CART	0.7142857	**0.7894737**	0.7777778	0.6666667
	Bagging	0.6666667	0.7368421	0.6666667	0.6666667
	Boosting	0.7142857	**0.7894737**	0.8	0.6666667

TABLEAU 3.9 – Comparaison des performances médianes des différentes méthodes pour $n = 100, \tau = 0.5$ et les DGP 1-3.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0.7058824**	**0.75**	0.7142857	**0.7142857**
	Réseau de neurones	0.6666667	0.7368421	0.6666667	0.6666667
	SVM	0.6666667	0.7368421	0.6666667	0.625
	Régression logistique	0.6666667	0.7368421	0.7142857	0.625
	KNN	0.625	0.6842105	0.625	0.625
	Modèle de Mélange	0.6666667	**0.75**	0.7142857	0.6666667
	LDA	0.6666667	0.7368421	**0.7272727**	0.6666667
	Foret aléatoire	0.5882353	0.6842105	0.6	0.5714286
	CART	0.6315789	0.7	0.6666667	0.6306818
	Bagging	0.5882353	0.6842105	0.5833333	0.5714286
	Boosting	0.6666667	0.7368421	0.6666667	**0.7142857**
DGP5	Copule	0.6666667	0.7368421	0.6666667	0.6666667
	Réseau de neurones	**0.7142857**	**0.7894737**	**0.75**	**0.7142857**
	SVM	**0.7142857**	**0.7894737**	**0.75**	**0.7142857**
	Régression logistique	**0.7142857**	**0.7894737**	**0.75**	0.7
	KNN	0.6666667	0.7368421	0.6666667	0.6666667
	Modèle de Mélange	0.7058824	**0.7894737**	**0.75**	**0.7142857**
	LDA	**0.7142857**	**0.7894737**	**0.75**	0.6666667
	Foret aléatoire	0.6153846	0.6842105	0.6	0.625
	CART	0.7	0.75	0.7	**0.7142857**
	Bagging	0.6153846	0.6842105	0.6153846	0.625
	Boosting	**0.7142857**	**0.7894737**	0.7142857	**0.7142857**
DGP6	Copule	**0.7142857**	**0.7894737**	**0.7142857**	**0.7142857**
	Réseau de neurones	0.6666667	0.7368421	0.6666667	**0.7142857**
	SVM	0.6666667	0.7368421	0.6666667	**0.7142857**
	Régression logistique	0.6666667	0.7368421	0.7	0.6
	KNN	0.625	0.6842105	0.625	0.625
	Modèle de Mélange	0.6666667	0.7368421	0.6666667	0.6458333
	LDA	0.6666667	0.7368421	0.7	0.5714286
	Foret aléatoire	0.5882353	0.6842105	0.6	0.5714286
	CART	0.6363636	0.7368421	0.6363636	0.6666667
	Bagging	0.5882353	0.6842105	0.6	0.5714286
	Boosting	0.7	0.7368421	0.6666667	**0.7142857**

TABLEAU 3.10 – Comparaison des performances médianes des différentes méthodes pour $n = 100, \tau = 0.5$ et les DGP 4-6.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP1	Copule	**0.10223**	**0.07995689**	**0.1218781**	**0.1265888**
	Réseau de neurones	0.1517923	0.09563318	0.165292	0.2069391
	SVM	0.1584898	0.09280772	0.1713584	0.2186562
	Régression logistique	0.14577	0.09280822	0.1540243	0.1955851
	KNN	0.1508568	0.10004451	0.1601843	0.1942063
	Modèle de Mélange	0.1437471	0.09209573	0.1525788	0.1968533
	LDA	0.1490159	0.09218348	0.1567453	0.1978749
	Foret aléatoire	0.1503316	0.10544309	0.15944	0.1878818
	CART	0.1493513	0.09959517	0.1680651	0.2127736
	Bagging	0.1496211	0.10638929	0.1605486	0.1863048
	Boosting	0.1471314	0.09862178	0.1689809	0.2062174
DGP2	Copule	**0.1040537**	**0.07715546**	**0.120582**	**0.1290707**
	Réseau de neurones	0.1450549	0.09249193	0.1658449	0.2074528
	SVM	0.1503566	0.09218649	0.1730162	0.2113428
	Régression logistique	0.1324253	0.09157385	0.15773	0.1884614
	KNN	0.1464044	0.10828316	0.1655341	0.2039663
	Modèle de Mélange	0.1368391	0.09203781	0.1536918	0.1909236
	LDA	0.1326785	0.09101158	0.158566	0.1867612
	Foret aléatoire	0.1439322	0.11087486	0.1628964	0.1953051
	CART	0.1469497	0.10138208	0.1629844	0.2115049
	Bagging	0.1424074	0.10981352	0.1620194	0.1946604
	Boosting	0.1363357	0.09231107	0.1539942	0.2020369
DGP3	Copule	**0.09900176**	**0.07113861**	**0.1220622**	**0.1297627**
	Réseau de neurones	0.1402004	0.08774651	0.1511514	0.186109
	SVM	0.14510764	0.08605307	0.1498522	0.1894994
	Régression logistique	0.13343917	0.0888632	0.1387687	0.1754764
	KNN	0.14333038	0.0994984	0.1569045	0.1825769
	Modèle de Mélange	0.13609225	0.08874744	0.1426155	0.1783317
	LDA	0.13611606	0.08944484	0.1425564	0.1775545
	Foret aléatoire	0.14399279	0.10682165	0.1598207	0.1775047
	CART	0.14540414	0.0935232	0.161534	0.1892051
	Bagging	0.14368054	0.10687638	0.1613635	0.1762549
	Boosting	0.13524226	0.08713268	0.1519963	0.1833782

TABLEAU 3.11 – Comparaison des performances et des écarts-types pour les méthodes avec $n = 100, \tau = 0.5$ et les DJP 1-3.

Modèles	Classifieurs	F1 Score	Accuracy	Précision	Sensibilité
DGP4	Copule	**0.103274**	**0.07870993**	**0.1184702**	**0.1326921**
	Réseau de neurones	0.1427374	0.09698701	0.1583039	0.1983986
	SVM	0.1537337	0.09964094	0.171574	0.2081118
	Régression logistique	0.1499372	0.09672793	0.1688343	0.1943091
	KNN	0.1503867	0.10864915	0.1634281	0.1872285
	Modèle de Mélange	0.1463263	0.09826421	0.1615546	0.1922693
	LDA	0.1525073	0.09844752	0.1703745	0.195064
	Forêt Aléatoire	0.1503744	0.10808742	0.1562533	0.188822
	CART	0.1423792	0.10521892	0.1659151	0.1935873
	Bagging	0.1501892	0.10804796	0.1554123	0.1894767
	Boosting	0.1454753	0.10184803	0.1623999	0.1990836
DGP5	Copule	**0.1075327**	**0.07986825**	**0.119599**	**0.1353605**
	Réseau de neurones	0.1381974	0.08929908	0.1529273	0.1902283
	SVM	0.1385133	0.08893862	0.1544557	0.1926684
	Régression logistique	0.1445732	0.08720699	0.1508421	0.1924088
	KNN	0.1449053	0.09809922	0.1568601	0.1903711
	Modèle de Mélange	0.1415992	0.0882638	0.1519749	0.1900719
	LDA	0.1456384	0.08754304	0.1502738	0.1935964
	Forêt Aléatoire	0.1491166	0.10262723	0.161245	0.185289
	CART	0.1413892	0.09788432	0.1596576	0.197617
	Bagging	0.1511567	0.10317797	0.1626794	0.1871676
	Boosting	0.1388863	0.09342038	0.1556024	0.1900723
DGP6	Copule	**0.1007616**	**0.07808548**	**0.1120845**	**0.1293982**
	Réseau de neurones	0.149304	0.10406924	0.158706	0.2085729
	SVM	0.1553987	0.10048532	0.1592664	0.2222239
	Régression logistique	0.1573125	0.09983959	0.1762621	0.1993549
	KNN	0.1379568	0.10159899	0.1597042	0.1863157
	Modèle de Mélange	0.1562605	0.10637785	0.1617565	0.2017715
	LDA	0.1584114	0.09897331	0.1757016	0.2014623
	Forêt Aléatoire	0.1450543	0.1088223	0.1682477	0.1856273
	CART	0.1491105	0.10489954	0.1548556	0.2032843
	Bagging	0.1434348	0.10881833	0.1690083	0.1836707
	Boosting	0.1449732	0.10478847	0.1573199	0.2054712

TABLEAU 3.12 – Comparaison des performances et des écarts-types pour les méthodes avec $n = 100, \tau = 0.5$ et les DGP 4-6.

Classifieur	F1-score	Accuracy	Sensitivité	Précision
Moyennes				
Copule	0.9536622	0.9562737	0.9597428	0.9545826
Réseau de neurones	0.5369195	0.5245895	0.4955320	0.5187385
SVM	0.5475374	0.5287053	0.5071968	0.5216457
Régression logistique	0.5431528	0.5258421	0.4853770	0.5099792
KNN	0.5073275	0.5023105	0.4899058	0.5035470
Modèle de Mélange	0.5357819	0.5225368	0.4933255	0.5136002
LDA	0.5433944	0.5264526	0.4853437	0.5094034
Forêt Aléatoire	0.5007827	0.4997737	0.4867650	0.5038327
CART	0.5180343	0.5143737	0.5012934	0.5138777
Bagging	0.5007693	0.4998684	0.4863132	0.5034198
Boosting	0.5121529	0.5105368	0.4838958	0.5078286
Médianes				
Copule	0.9523810	0.9473684	1.0000000	1.0000000
Réseau de neurones	0.5600000	0.5263158	0.5000000	0.5333333
SVM	0.5714286	0.5263158	0.5000000	0.5294118
Régression logistique	0.5833333	0.5263158	0.5000000	0.5263158
KNN	0.5217391	0.5000000	0.5000000	0.5000000
Modèle de Mélange	0.5600000	0.5263158	0.5000000	0.5000000
LDA	0.5833333	0.5263158	0.5000000	0.5263158
Forêt Aléatoire	0.5000000	0.5000000	0.5000000	0.5000000
CART	0.5263158	0.5263158	0.5000000	0.5000000
Bagging	0.5000000	0.5000000	0.5000000	0.5000000
Boosting	0.5217391	0.5263158	0.4444444	0.5000000
Écarts-types				
Copule	0.04611841	0.0457056	0.06110331	0.05952557
Réseau de neurones	0.15759699	0.1044189	0.27732424	0.14870848
SVM	0.16787768	0.1050308	0.32206843	0.17099421
Régression logistique	0.17115521	0.1005200	0.34294245	0.18031296
KNN	0.13781654	0.1138741	0.18382955	0.13303609
Modèle de Mélange	0.17139767	0.1060618	0.31664724	0.15905624
LDA	0.17213704	0.1001001	0.34470329	0.18667007
Forêt Aléatoire	0.13622054	0.1135069	0.17708844	0.13157502
CART	0.14399685	0.1116313	0.20823486	0.14278016
Bagging	0.13703654	0.1141221	0.17652184	0.13232284
Boosting	0.15530201	0.1075281	0.23640883	0.14379121

TABLEAU 3.13 – Les performances (moyennes, médianes et écarts-types) pour chaque classifieur pour le modèle logistique avec $\beta_1 = 0{,}1$.

Classifieur	F1-score	Accuracy	Sensitivité	Précision
Moyennes				
Copule	0.6528520	0.6587316	0.6536397	0.6590894
Réseau de neurones	0.8318680	0.8367211	0.8333326	0.8501443
SVM	0.8320796	0.8376474	0.8315905	0.8517091
Régression logistique	0.8373739	0.8421211	0.8382937	0.8530610
KNN	0.7985896	0.8055579	0.7985414	0.8182671
Modèle de Mélange	0.8352444	0.8404789	0.8359438	0.8525607
LDA	0.8353843	0.8406684	0.8355328	0.8522405
Forêt Aléatoire	0.7658962	0.7732474	0.7685108	0.7843583
CART	0.8156286	0.8218000	0.8194190	0.8362468
Bagging	0.7652705	0.7726316	0.7677542	0.7842099
Boosting	0.8256290	0.8307263	0.8263091	0.8460760
Médianes				
Copule	0.6666667	0.6500000	0.6666667	0.6666667
Réseau de neurones	0.8421053	0.8421053	0.8750000	0.8571429
SVM	0.8421053	0.8421053	0.8750000	0.8571429
Régression logistique	0.8571429	0.8421053	0.8750000	0.8571429
KNN	0.8181818	0.8000000	0.8181818	0.8181818
Modèle de Mélange	0.8571429	0.8421053	0.8750000	0.8571429
LDA	0.8441296	0.8421053	0.8750000	0.8571429
Forêt Aléatoire	0.7777778	0.7894737	0.8000000	0.7777778
CART	0.8235294	0.8421053	0.8571429	0.8333333
Bagging	0.7777778	0.7894737	0.8000000	0.7777778
Boosting	0.8333333	0.8421053	0.8750000	0.8571429
Écarts-types				
Copule	0.09252652	0.07842676	0.1134503	0.09534935
Réseau de neurones	0.09697337	0.08684713	0.1408039	0.10923837
SVM	0.09649466	0.08563281	0.1413190	0.10788513
Régression logistique	0.09505153	0.08450345	0.1342520	0.10602360
KNN	0.11022464	0.09573938	0.1509821	0.11565596
Modèle de Mélange	0.09824241	0.08684634	0.1398440	0.10714180
LDA	0.09676874	0.08500508	0.1366943	0.10588316
Forêt Aléatoire	0.11478168	0.09964250	0.1569166	0.12081534
CART	0.10779450	0.09319743	0.1553482	0.11815096
Bagging	0.11509450	0.10003344	0.1571409	0.12233065
Boosting	0.09888306	0.08969827	0.1432627	0.11316580

TABLEAU 3.14 – Les performances (moyennes, médianes et écarts-types) pour chaque classifieur pour le modèle logistique avec $\beta_1 = 3$.

Classifieur	F1-score	Accuracy	Sensitivité	Précision
Moyennes				
Copule	0.9526148	0.9555105	0.9583633	0.9539690
Réseau de neurones	0.5440102	0.5238895	0.5099412	0.5173149
SVM	0.5454833	0.5335684	0.5087980	0.5290235
Régression logistique	0.5614345	0.5440053	0.5117330	0.5311646
KNN	0.5209751	0.5137053	0.4982602	0.5143812
Modèle de Mélange	0.5463807	0.5357526	0.5030507	0.5257155
LDA	0.5599838	0.5432737	0.5101967	0.5299459
Forêt Aléatoire	0.5202574	0.5141737	0.5020347	0.5217031
CART	0.5161863	0.5065316	0.5029676	0.5096370
Bagging	0.5201951	0.5141632	0.5018322	0.5220561
Boosting	0.5341865	0.5207211	0.5086977	0.5213981
Médianes				
Copule	0.9523810	0.9473684	1.0000000	1.0000000
Réseau de neurones	0.5555556	0.5263158	0.5000000	0.5333333
SVM	0.5714286	0.5263158	0.5000000	0.5384615
Régression logistique	0.6086957	0.5500000	0.5000000	0.5500000
KNN	0.5263158	0.5263158	0.5000000	0.5000000
Modèle de Mélange	0.5714286	0.5263158	0.5000000	0.5358974
LDA	0.6000000	0.5500000	0.5000000	0.5500000
Forêt Aléatoire	0.5263158	0.5263158	0.5000000	0.5384615
CART	0.5263158	0.5263158	0.5000000	0.5000000
Bagging	0.5263158	0.5263158	0.5000000	0.5384615
Boosting	0.5454545	0.5263158	0.5000000	0.5333333
Écarts-types				
Copule	0.0486944	0.0484880	0.0628439	0.0625926
Réseau de neurones	0.1391005	0.0995793	0.2667719	0.1354307
SVM	0.1625722	0.0925645	0.3238106	0.1565284
Régression logistique	0.1619193	0.0924309	0.3417343	0.1513099
KNN	0.1273555	0.1093102	0.1798768	0.1356553
Modèle de Mélange	0.1545489	0.0943713	0.3045047	0.1417462
LDA	0.1635391	0.0924730	0.3444824	0.1550338
Forêt Aléatoire	0.1319609	0.1104477	0.1700993	0.1300815
CART	0.1401046	0.1131931	0.2014472	0.1358631
Bagging	0.1326537	0.1109005	0.1700789	0.1315457
Boosting	0.1423880	0.1073476	0.2305088	0.1335392

TABLEAU 3.15 – Les performances (moyennes, médianes et écarts-types) pour chaque classifieur pour le modèle Probit avec $\beta_1 = 0{,}1$.

Classifieur	F1-score	Accuracy	Sensitivité	Précision
Moyennes				
Copule	0.613	0.622	0.614	0.619
Réseau de neurones	0.892	0.895	0.892	0.902
SVM	0.891	0.894	0.890	0.903
Régression logistique	0.895	0.897	0.896	0.904
KNN	0.877	0.880	0.876	0.890
Modèle de Mélange	0.892	0.895	0.891	0.904
LDA	0.893	0.897	0.890	0.907
Forêt Aléatoire	0.853	0.856	0.858	0.861
CART	0.881	0.885	0.881	0.897
BAGGING	0.854	0.857	0.859	0.862
BOOSTING	0.891	0.894	0.891	0.904
Médianes				
Copule	0.632	0.632	0.625	0.625
Réseau de neurones	0.900	0.895	0.900	0.900
SVM	0.900	0.895	0.900	0.900
Régression logistique	0.900	0.895	0.900	0.900
KNN	0.880	0.895	0.889	0.889
Modèle de Mélange	0.900	0.895	0.900	0.900
LDA	0.900	0.895	0.900	0.909
Forêt Aléatoire	0.857	0.850	0.889	0.875
CART	0.889	0.895	0.900	0.900
Bagging	0.857	0.850	0.889	0.875
Boosting	0.900	0.895	0.900	0.900
Écarts-types				
Copule	0.090	0.073	0.112	0.087
Réseau de neurones	0.078	0.074	0.106	0.096
SVM	0.078	0.073	0.106	0.095
Régression logistique	0.074	0.071	0.101	0.093
KNN	0.081	0.076	0.108	0.100
Modèle de Mélange	0.078	0.073	0.105	0.096
LDA	0.078	0.072	0.107	0.094
Forêt Aléatoire	0.086	0.081	0.115	0.106
CART	0.082	0.076	0.119	0.100
Bagging	0.087	0.081	0.116	0.107
Boosting	0.076	0.072	0.107	0.096

TABLEAU 3.16 – Les performances (moyennes, médianes et écarts-types) pour chaque classifieurs pour le modèle Probit avec $\beta_1 = 3$.

CONCLUSION

Ce mémoire a exploré différentes approches de classification, en comparant les méthodes classiques avec celle basée sur les copules. Le premier chapitre a détaillé les techniques classiques de classification, tandis que le second a mis en avant les copules, offrant une meilleure modélisation des dépendances complexes. Dans le troisième chapitre, nous avons simulé des données pour comparer les performances des méthodes. Les résultats ont montré que les copules surpassent dans plusieurs scénarios les méthodes classiques. Enfin, les pistes de recherche future incluent l'exploration d'autres familles de copules, l'application de ces méthodes à des échantillons plus grands, ainsi que l'adaptation des modèles à des données avec des structures plus complexes.

ANNEXE A

Bibliographie

[BD10] G. Biau and L. Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101 :2499–2518, 2010.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 26(2) :123–140, 1996.

[Cla78] D. G. Clayton. A model for association in bivariate data. *Biometrika*, 65(1) :141–151, 1978.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[Dob90] A. J. Dobson. *An Introduction to Generalized Linear Models*. Chapman and Hall/CRC, 1990.

[DS66] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. Wiley, 1966.

[Fis36] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2) :179–188, 1936.

[Fra79] William Frank. On the class of archimedean copulas. *Journal of Statistical Planning and Inference*, 2(3) :233–243, 1979.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1) :119–139, 1997.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 1st edition, 2016. Chapter 6 : Deep Feedforward Networks.

[GGR95] Christian Genest, Karim Ghoudi, and Louis-Paul Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3) :543–552, 1995.

[Gum60] Emil Julius Gumbel. *Statistics of Extremes*. Columbia University Press, New York, 1960.

[HZHA16] Gazi Hossain, Mohammad Zubair, Md Rashed Harun, and Sanzida Akter. Neural network models for predicting business failures. *Journal of Business Research*, 69(8) :3011–3016, 2016.

[Joe97] Harry Joe. *Multivariate Models and Dependence Concepts*. Chapman and Hall/CRC, 1997.

[Joe14] Harry Joe. *Dependence Modeling with Copulas*. Monographs on Statistics & Applied Probability. Chapman & Hall/CRC, 2014.

[JX96] Harry Joe and J. J. Xu. The estimation method of inference functions for margins for multivariate models. *Journal of the American Statistical Association*, 91(433) :507–514, 1996.

[KSS07] Minjoo Kim, Mervyn J. Silvapulle, and Paramsothy Silvapulle. Comparison of semiparametric and parametric methods for estimating copulas. *Computational Statistics Data Analysis*, 51(6) :2836–2850, 2007.

[LPM06] Ludovic Lebart, Marie Piron, and Alain Morineau. *Statistique exploratoire multidimensionnelle*. Dunod, Paris, 4ème édition edition, 2006. Visualisation et inférence en fouille de données.

[MBB23] M. Mesfioui, T. Bouezmarni, and M. Belalia. Copula-based link functions in binary regression models. *Statistical Papers*, 64(2) :557–585, 2023.

[MKB+10] Tomas Mikolov, Martin Karafiát, Lukáš Burget, Jiri Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. *Interspeech 2010*, 2010. pp. 1045-1048.

[MN83] Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall/CRC, 1983.

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4) :115–133, 1943.

[Rao73] C. R. Rao. *Linear Statistical Inference and Its Applications*. John Wiley & Sons, New York, 2nd edition, 1973. Voir pages 327–329 pour l'information de Fisher.

[Ros58] F. Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6) :386–408, 1958.

[Ser80] Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, New York, 1980.

[SH07] Xi C. Song and T. M. Huang. Nonparametric estimation of copula functions for dependence modelling. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 35(2) :265–282, 2007.

[Sha99] Jun Shao. *Mathematical Statistics*. Springer, New York, 1999.

[SL95] Joanne H. Shih and Thomas A. Louis. Inferences on the association parameter in copula models for bivariate survival data. *Biometrika*, 82(4) :841–850, 1995.

[SS81] B. Schweizer and A. Sklar. Associative copulas. *Annals of Probability*, 9(3) :459–480, 1981.