

IBM SPSS Neural Networks 28



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section [«Remarques»](#), à la page 19.

Notice d'édition

La présente édition s'applique à la version 28.0.0 d'IBM® SPSS Statistics et à toutes les éditions et modifications ultérieures sauf mention contraire dans les nouvelles éditions.

© Copyright International Business Machines Corporation .

Table des matières

Chapitre 1. Réseaux neuronaux.....	1
Introduction aux réseaux neuronaux.....	1
Qu'est-ce qu'un réseau neuronal ?.....	1
Structure d'un réseau neuronal.....	2
Perceptron multicouche.....	3
Partitions	5
Architecture	6
Entraînement	7
Sortie	9
Enregistrer	10
Exportation	11
Options	11
fonction de base radiale.....	12
Partitions	14
Architecture	14
Sortie	15
Enregistrer	16
Exportation	17
Options	17
Remarques.....	19
Marques.....	20
Index.....	23

Chapitre 1. Réseaux neuronaux

Les fonctions de réseau neuronal sont incluses dans SPSS Statistics Premium Edition ou l'option Réseaux neuronaux.

Introduction aux réseaux neuronaux

Les réseaux neuronaux constituent l'outil de prédilection dans de nombreuses applications prédictives d'exploration de données en raison de leurs puissance, souplesse et convivialité. Les réseaux neuronaux prédictifs sont particulièrement utiles dans les applications où le processus sous-jacent est complexe, comme dans les cas suivants :

- Préviation de la demande des consommateurs pour rationaliser les coûts de production et de livraison.
- Préviation de la probabilité de réponse à un publipostage pour déterminer les ménages d'une liste de mailing auxquels envoyer une offre.
- Evaluation d'un demandeur pour déterminer le risque de prolonger son crédit.
- Détection des opérations frauduleuses dans une base de données de déclarations de sinistre.

Les réseaux neuronaux utilisés dans les applications prédictives, tels que les réseaux de type **perceptron multicouche (MLP, Multilayer Perceptron)** et **fonction à base radiale (RBF, Radial Basis Function)**, sont supervisés, en ce sens que les résultats prévus par le modèle peuvent être comparés aux valeurs connues des variables cible. L'option Réseaux neuronaux vous permet d'ajuster les réseaux MLP et RBF et d'enregistrer les modèles obtenus à des fins d'évaluation.

Qu'est-ce qu'un réseau neuronal ?

Le terme **réseau neuronal** s'applique à une famille de modèles vaguement apparentée, caractérisée par un grand espace de paramètres et une structure flexible, inspirée des études sur le fonctionnement du cerveau. Au fur et à mesure que la famille s'est agrandie, la plupart des nouveaux modèles ont été conçus pour des applications non biologiques, bien que la majeure partie de la terminologie associée reflète leur origine.

Les définitions spécifiques des réseaux neuronaux sont aussi variées que les domaines dans lesquels elles sont employées. Alors qu'aucune définition ne couvre exactement l'ensemble de la famille de modèles, pour l'instant, examinons la description suivante ¹ :

Un réseau neuronal est un processeur massivement distribué en parallèle qui a une propension naturelle à stocker des connaissances empiriques et à les rendre disponibles en vue d'une utilisation. Il ressemble au cerveau sur deux aspects :

- La connaissance est acquise par le réseau à travers un processus d'apprentissage.
- Les connexions entre les neurones, connues sous le nom de pondérations synaptiques, servent à stocker les connaissances.

Pour obtenir des détails sur la raison pour laquelle cette définition est peut-être trop restrictive, voir ².

Pour différencier les réseaux neuronaux des méthodes statistiques traditionnelles à l'aide de cette définition, ce qui n'est *pas* explicite est tout aussi important que le texte de la définition lui-même. Par exemple, le modèle de régression linéaire traditionnel peut acquérir des connaissances via la méthode des moindres carrés et stocker cette connaissance dans les coefficients de régression. En ce sens, il s'agit d'un réseau neuronal. En réalité, vous pouvez avancer que la régression linéaire est un cas particulier de certains réseaux neuronaux. Toutefois, la régression linéaire a une structure de modèle stricte et un ensemble d'hypothèses qui sont imposés avant d'acquérir des connaissances de ces données.

¹ Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation*, 2nd ed. New York: Macmillan College Publishing.

² Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

En revanche, la définition ci-dessus crée des contraintes minimales pour la structure du modèle et les hypothèses. Par conséquent, un réseau neuronal peut se rapprocher d'un grand nombre de modèles statistiques sans que vous deviez imaginer à l'avance certaines relations entre les variables dépendantes et indépendantes. En fait, la forme de la relation est déterminée pendant le processus d'apprentissage. Si une relation linéaire entre les variables dépendantes et indépendantes est adaptée, les résultats du réseau neuronal devraient se rapprocher étroitement de ceux du modèle de régression linéaire. Si une relation non linéaire est plus adaptée, le réseau neuronal se rapproche automatiquement de la structure du modèle "correcte".

L'inconvénient de cette flexibilité est que les pondérations synaptiques d'un réseau neuronal ne sont pas facilement interprétables. Par conséquent, si vous essayez d'expliquer un processus sous-jacent qui crée des relations entre les variables dépendantes et indépendantes, il serait préférable d'utiliser un modèle statistique plus traditionnel. Toutefois, s'il n'est pas important de pouvoir interpréter le modèle, vous pouvez souvent obtenir plus rapidement des résultats du modèle satisfaisants en utilisant un réseau neuronal.

Structure d'un réseau neuronal

Bien que les réseaux neuronaux imposent des contraintes minimales à la structure du modèle et aux hypothèses, il est utile de comprendre leur **architecture** générale. Le réseau MLP ou RBF est une fonction de prédicteurs (également appelées entrées) qui minimise l'erreur de prévision des variables cible (également appelées sorties).

Le produit comporte le jeu de données *bankloan.sav*, qui peut s'avérer utile pour l'identification des personnes susceptibles d'être débiteur en défaut parmi un groupe de demandeurs de prêt. Un réseau MLP ou RBF appliqué à ce problème est une fonction des mesures qui minimise l'erreur dans la prévision du manquement. La figure suivante permet de saisir la forme de cette fonction.

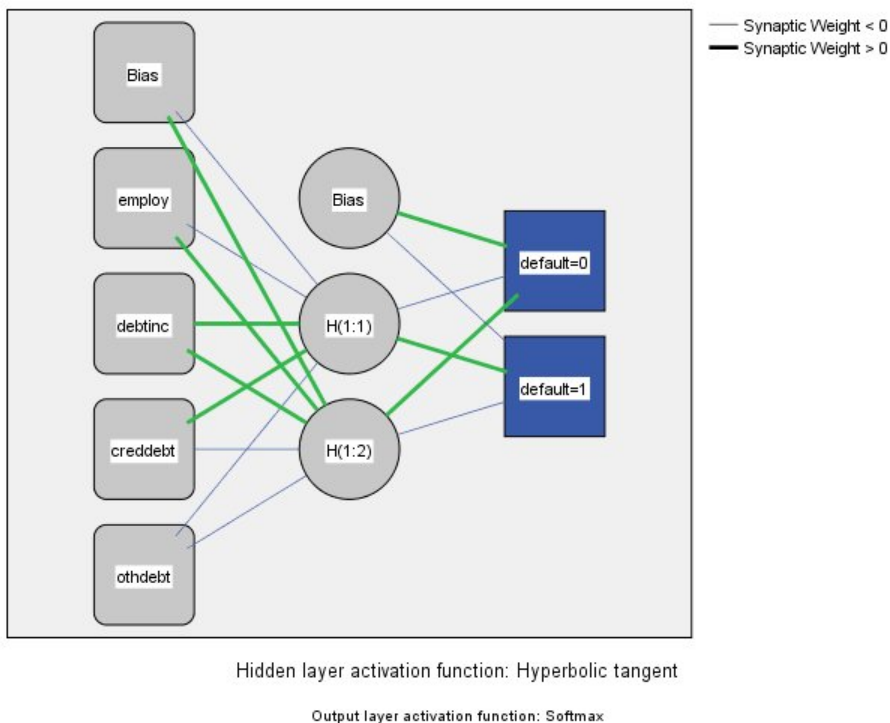


Figure 1. Architecture d'anticipation avec une couche masquée

Cette structure est appelée architecture d'**anticipation** car le flux des connexions du réseau progresse de la couche d'entrée vers la couche de sortie sans former de boucles de réaction. Dans cette figure :

- La **couche d'entrée** contient les prédicteurs.

- La **couche masquée** contient les unités ou noeuds non observables. La valeur de chaque unité masquée constitue une fonction des prédicteurs ; la forme exacte de la fonction dépend, pour une partie, du type de réseau et, pour une autre partie, des spécifications contrôlables par l'utilisateur.
- La **couche de sortie** contient les réponses. Dans la mesure où l'historique du manquement est une variable catégorielle comprenant deux catégories, celle-ci est recodée en deux variables indicateur. Chaque unité de sortie constitue une fonction des unités masquées. La forme exacte de la fonction dépend, pour une partie, du type de réseau et, pour une autre partie, des spécifications contrôlables par l'utilisateur.

Le réseau MLP autorise une seconde couche masquée ; dans ce cas, chaque unité de la seconde couche masquée est une fonction des unités dans la première couche masquée et chaque réponse est une fonction des unités dans la seconde couche masquée.

Perceptron multicouche

La procédure Perceptron multicouche produit un modèle de prévision pour une ou plusieurs variables (cible) dépendantes en fonction de valeurs de variables de prédicteur.

Exemples : Voici deux scénarios utilisant la procédure MLP :

Un responsable des prêts dans une banque souhaite pouvoir identifier les caractéristiques qui indiquent les personnes susceptibles de manquer à leurs engagements et d'utiliser ces caractéristiques pour identifier les bons et les mauvais risques de crédit. A l'aide d'un échantillon d'anciens clients, elle peut entraîner un modèle Perceptron multicouche, valider l'analyse grâce à un échantillon restant d'anciens clients, puis utiliser le réseau pour classer les clients éventuels entre bons et mauvais risques de crédit.

Un hôpital souhaite effectuer un suivi des coûts et des durées de séjour des patients admis pour soigner un infarctus du myocarde (crise cardiaque). Des estimations précises de ces mesures permettent à l'administration de gérer correctement le nombre de lits disponibles lors du traitement des patients. Grâce à l'utilisation des archives sur les traitements d'un échantillon de patients ayant été soignés pour un infarctus du myocarde, l'administration peut entraîner un réseau pour prévoir le coût et la durée du séjour à l'hôpital.












Analyse des données

Variables dépendantes : Les variables dépendantes peuvent être :

- *Nominal*. Une variable peut être traitée comme étant nominale si ses valeurs représentent des catégories sans classement intrinsèque (par exemple, le service de la société dans lequel travaille un employé). La région, le code postal ou l'appartenance religieuse sont des exemples de variables nominales.
- *Ordinal*. Une variable peut être traitée comme étant ordinale si ses valeurs représentent des catégories associées à un classement intrinsèque (par exemple, des niveaux de satisfaction allant de Très mécontent à Très satisfait). Exemples de variable ordinale : des scores d'attitude représentant le degré de satisfaction ou de confiance, et des scores de classement des préférences.
- *Echelle*. Une variable peut être traitée comme une variable d'échelle (continue) si ses valeurs représentent des catégories ordonnées avec une mesure significative, de sorte que les comparaisons de distance entre les valeurs soient adéquates. L'âge en années et le revenu en milliers de dollars sont des exemples de variable d'échelle.

La procédure considère que le niveau de mesure approprié a été assigné à toutes les variables dépendantes, bien que vous puissiez changer provisoirement le niveau de mesure d'une variable en cliquant avec le bouton droit de la souris sur la variable dans la liste des variables source, puis en sélectionnant un niveau de mesure dans le menu contextuel.

Dans la liste des variables, une icône indique le niveau de mesure et le type de données :

	Numérique	Chaîne	Date	Heure
Echelle (continue).		n/a		
Ordinal				
Champs nominaux				

Variables de prédicteur : Les variables de prédicteur peuvent être spécifiées en tant que facteurs (qualitatifs) ou covariables (d'échelle).

Codification des variables indicatrices : La procédure recode provisoirement les variables de prédicteur catégorielles et les variables dépendantes via la codification un-de-c pour la durée de la procédure. S'il existe des catégories c d'une variable, la variable est stockée comme vecteurs c, la première catégorie étant identifiée par (1,0,...,0), la suivante par (0,1,0,...,0), ... et la dernière par (0,0,...,0,1).

Ce système de codification augmente le nombre de pondérations synaptiques et peut résulter en un entraînement plus lent, mais les méthodes de codification plus compactes aboutissent généralement à des réseaux neuronaux mal ajustés. Si l'entraînement de votre réseau s'effectue très lentement, essayez de réduire le nombre de catégories dans vos prédicteurs catégoriels en combinant des catégories similaires ou en supprimant les observations comportant des catégories extrêmement rares.

Toute codification un-de-c repose sur les données d'apprentissage, même si un échantillon de test ou restant est défini (voir «Partitions», à la page 5). Ainsi, si les échantillons de test ou restants contiennent des observations avec des catégories de prédicteurs ne figurant pas dans les données d'apprentissage, ces observations ne sont pas utilisées par la procédure ou dans l'évaluation. Si les échantillons de test ou restants contiennent des observations avec des catégories de variables dépendantes ne figurant pas dans les données d'apprentissage, ces observations ne sont pas utilisées par la procédure mais peuvent être notées.

Rééchelonnement : Les variables d'échelle dépendantes et les covariables sont rééchelonnées par défaut pour améliorer l'entraînement du réseau. Le rééchelonnement repose entièrement sur les données d'apprentissage, même si un échantillon de test ou restant est défini (voir «Partitions», à la page 5). En d'autres termes, en fonction du type de rééchelonnement, la moyenne, l'écart type, la valeur minimale ou la valeur maximale d'une covariable ou d'une variable dépendante ne sont calculées qu'à l'aide des données d'apprentissage. Si vous spécifiez une variable pour définir des partitions, il est important que ces covariables ou variables dépendantes présentent des distributions similaires à travers les échantillons d'apprentissage, de test et restants.

Pondérations de fréquence : Cette procédure ignore les pondérations de fréquence.

Réplication de résultats : Si vous souhaitez répliquer exactement vos résultats, outre les mêmes paramètres de procédure, utilisez la même valeur d'initialisation pour le générateur de nombres aléatoires, le même ordre de données et le même ordre de variables. Vous trouverez ci-après plus de détails sur cet aspect.

- **Génération de nombres aléatoires :** La procédure utilise la génération de nombres aléatoires pendant l'affectation aléatoire de partitions, le sous-échantillonnage aléatoire pour l'initialisation des pondérations synaptiques, le sous-échantillonnage aléatoire pour la sélection automatique de l'architecture et l'algorithme recuit simulé utilisé dans l'initialisation de la pondération et la sélection automatique de l'architecture. Pour reproduire les mêmes résultats aléatoires à l'avenir, utilisez la même valeur d'initialisation pour le générateur de nombres aléatoires avant chaque exécution de la procédure Perceptron multicouche.
- **Tri par observation :** Les méthodes d'entraînement en ligne et par mini-lot (voir «Entraînement», à la page 7) dépendent explicitement de l'ordre des observations. Toutefois, même l'entraînement par lot

dépend de l'ordre des observations car l'initialisation des pondérations synaptiques implique le sous-échantillonnage du jeu de données.

Pour réduire les effets de tri, classez les observations de manière aléatoire. Pour vérifier la stabilité d'une solution donnée, vous pouvez obtenir différentes solutions dans lesquelles les observations sont triées de différentes manières aléatoires. Si les fichiers sont très volumineux, vous pouvez effectuer plusieurs fois l'opération sur un échantillon des observations triées de différentes manières aléatoires.

- **Ordre des variables** : Les résultats peuvent être influencés par l'ordre des variables dans les listes des facteurs et des covariables en raison du motif différent de valeurs initiales affectées lorsque l'on change l'ordre des variables. Comme avec les effets d'ordre des observations, vous pouvez essayer différents ordres de variables (il suffit d'utiliser la fonction glisser-déplacer dans les listes de facteurs et de covariables) pour évaluer la stabilité d'une solution donnée.

Création d'un réseau Perceptron multicouche

A partir des menus, sélectionnez :

Analyse > Réseaux neuronaux > Perceptron multicouche...

1. Sélectionnez au moins une variable dépendante.
2. Sélectionnez au moins un facteur ou une covariable.

Dans l'onglet Variables, vous pouvez aussi changer la méthode de rééchantillonnage des covariables. Les choix sont les suivants :

- **Standardisé** : Soustrayez la moyenne et divisez le résultat par l'écart type, $(x - \text{moyenne})/s$.
- **Normalisé** : Soustrayez le minimum et divisez le résultat par la plage, $(x - \text{min})/(\text{max} - \text{min})$. Les valeurs normalisées sont comprises entre 0 et 1.
- **Normalisé ajusté** : Version ajustée de la soustraction du minimum et de la division du résultat par la plage, $[2 * (x - \text{min}) / (\text{max} - \text{min})] - 1$. Les valeurs normalisées ajustées sont comprises entre -1 et 1.
- **Aucun** : Aucun rééchantillonnage des covariables.

Champs avec un niveau de mesure inconnu

L'alerte du niveau de mesure apparaît lorsque le niveau de mesure d'une ou plusieurs variables (champs) du jeu de données est inconnu. Le niveau de mesure ayant une incidence sur le calcul des résultats de cette procédure, toutes les variables doivent avoir un niveau de mesure défini.

Analyser les données : Lit les données dans le jeu de données actifs et attribue le niveau de mesure par défaut à tous les champs ayant un niveau de mesure inconnu. Si le jeu de données est important, cette action peut prendre un certain temps.

Affecter manuellement : Ouvre une boîte de dialogue qui répertorie tous les champs ayant un niveau de mesure inconnu. Vous pouvez utiliser cette boîte de dialogue pour attribuer un niveau de mesure à ces champs. Vous pouvez également attribuer un niveau de mesure dans la vue de variable de l'éditeur de données.

Le niveau de mesure étant important pour cette procédure, vous ne pouvez pas accéder à la boîte de dialogue d'exécution de cette procédure avant que tous les champs n'aient des niveaux de mesure définis.

Partitions

Jeu de données de partition : Ce groupe indique la méthode de partitionnement du jeu de données actif en échantillons d'apprentissage, de test et restants. L'**échantillon d'apprentissage** comprend les enregistrements de données utilisés pour entraîner le réseau neuronal. Un certain pourcentage d'observations contenues dans le jeu de données doit être affecté à l'échantillon d'apprentissage pour l'obtention d'un modèle. L'**échantillon de test** est un ensemble indépendant d'enregistrements de données utilisé pour identifier des erreurs au cours de l'entraînement afin d'empêcher un entraînement excessif. Nous vous conseillons fortement de créer un échantillon de test. Un entraînement du réseau sera en général plus efficace si l'échantillon de test est plus petit que l'échantillon d'apprentissage. L'**échantillon restant** est un autre ensemble indépendant d'enregistrements de données utilisé pour

évaluer le réseau neuronal final ; l'erreur pour l'échantillon restant donne une estimation "honnête" de la capacité de prévision du modèle parce que les observations restantes n'ont pas été utilisées pour construire le modèle.

- **Affecter des observations de façon aléatoire en fonction du nombre d'observations relatif** : Indiquez le nombre d'observations relatif (rapport) affecté de façon aléatoire à chaque échantillon (d'apprentissage, de test et restant). La colonne % rapporte le pourcentage d'observations qui seront affectées à chaque échantillon en fonction des nombres relatifs que vous avez spécifiés.

Par exemple, si vous spécifiez 7, 3 et 0 comme nombres relatifs pour les échantillons d'apprentissage, de test et restants, ces valeurs correspondent à 70 %, 30 % et 0 %. Si vous spécifiez 2, 1, 1 comme nombres relatifs, ces valeurs correspondent à 50 %, 25 % et 25 %. 1, 1 et 1 correspond à la division du jeu de données en tiers égaux entre l'apprentissage, le test et les données restantes.

- **Utiliser une variable de partitionnement pour affecter des observations** : Indiquez une variable numérique qui affecte chaque observation du jeu de données actif à l'échantillon d'apprentissage, de test et restant. Les observations contenant une valeur positive sur la variable sont affectées à l'échantillon d'apprentissage, celles contenant une valeur égale à 0 sont affectées à l'échantillon de test, et celles contenant une valeur négative sont affectées à l'échantillon restant. Les observations contenant des valeurs système manquantes sont exclues de l'analyse. Les valeurs manquantes de l'utilisateur pour la variable de partitionnement sont toujours considérées comme étant valides.

Remarque : L'utilisation d'une variable de partitionnement ne garantira pas des résultats identiques dans les exécutions successives de la procédure. Reportez-vous à la section "Réplication des résultats" de la rubrique principale [Perceptron multicouche](#).

Architecture

L'onglet Architecture permet de spécifier la structure du réseau. La procédure peut sélectionner la "meilleure" architecture automatiquement ou vous pouvez spécifier une architecture personnalisée.

La sélection automatique de l'architecture construit un réseau avec une couche masquée. Spécifiez le nombre minimal et maximal d'unités autorisé dans la couche masquée pour que la sélection automatique de l'architecture calcule le "meilleur" nombre d'unités figurant dans la couche masquée. La sélection automatique de l'architecture utilise les fonctions d'activation par défaut pour les couches masquées et de sortie.

La sélection personnalisée de l'architecture vous permet de contrôler très précisément les couches masquées et de sortie et peut être très utile lorsque vous savez à l'avance quelle architecture vous souhaitez ou lorsque vous devez modéliser les résultats de la sélection automatique d'architecture.

Couches masquées

La couche masquée contient des nœuds (unités) de réseau non observables. Chaque unité masquée est une fonction de la somme pondérée des entrées. La fonction de la fonction d'activation et les valeurs des pondérations sont déterminées par l'algorithme d'estimation. Si le réseau contient une seconde couche masquée, chaque unité masquée de la seconde couche est une fonction de la somme pondérée des unités de la première couche masquée. La même fonction d'activation est utilisée dans les deux couches.

Nombre de couches masquées. Le perceptron multicouche peut avoir une ou deux couches masquées.

Fonction d'activation. La fonction d'activation "lie" les sommes pondérées des unités dans une couche aux valeurs des unités dans la couche réussie.

- **Tangente hyperbolique** : La syntaxe de cette fonction est la suivante : $\gamma(c) = \tanh(c) = (e^c - e^{-c}) / (e^c + e^{-c})$. Elle extrait les arguments de valeurs réelles et les transforme en plage (-1, 1). Lorsque la sélection automatique de l'architecture est utilisée, il s'agit de la fonction d'activation de toutes les unités dans les couches masquées.
- **Ogive de Galton** : La syntaxe de cette fonction est la suivante : $\gamma(c) = 1 / (1 + e^{-c})$. Elle extrait les arguments de valeurs réelles et les transforme en plage (0, 1).

Nombre d'unités. Le nombre d'unités dans chaque couche masquée peut être défini explicitement ou déterminé automatiquement par l'algorithme d'estimation.

Couche de sortie

La couche de sortie contient les variables (dépendantes) cible.

Fonction d'activation. La fonction d'activation "lie" les sommes pondérées des unités dans une couche aux valeurs des unités dans la couche réussie.

- **Identité** : La syntaxe de cette fonction est la suivante : $\gamma(c) = c$. Elle extrait les arguments de valeurs réelles et les renvoie inchangés. Lorsque la sélection automatique de l'architecture est utilisée, il s'agit de la fonction d'activation des unités de la couche de sortie s'il existe des variables d'échelle dépendantes.
- **Softmax** : La syntaxe de cette fonction est la suivante : $\gamma(c_k) = \exp(c_k) / \sum_j \exp(c_j)$. Cette fonction extrait un vecteur des arguments de valeur réels et le transforme en un vecteur dont les éléments sont compris dans la plage (0, 1) et ont pour somme 1. La fonction Softmax n'est disponible que si toutes les variables dépendantes sont des variables catégorielles. Lorsque la sélection automatique de l'architecture est utilisée, il s'agit de la fonction d'activation des unités de la couche de sortie si toutes les variables dépendantes sont des variables catégorielles.
- **Tangente hyperbolique** : La syntaxe de cette fonction est la suivante : $\gamma(c) = \tanh(c) = (e^c - e^{-c}) / (e^c + e^{-c})$. Elle extrait les arguments de valeurs réelles et les transforme en plage (-1, 1).
- **Ogive de Galton** : La syntaxe de cette fonction est la suivante : $\gamma(c) = 1 / (1 + e^{-c})$. Elle extrait les arguments de valeurs réelles et les transforme en plage (0, 1).

Rééchelonnement des variables d'échelle dépendantes : Ces contrôles ne sont disponibles que si une variable d'échelle dépendante au moins a été sélectionnée.

- **Standardisé** : Soustrayez la moyenne et divisez le résultat par l'écart type, $(x - \text{moyenne}) / s$.
- **Normalisé** : Soustrayez le minimum et divisez le résultat par la plage, $(x - \text{min}) / (\text{max} - \text{min})$. Les valeurs normalisées sont comprises entre 0 et 1. Il s'agit de la méthode de rééchelonnement requise pour les variables d'échelle dépendantes si la couche de sortie utilise la fonction d'activation d'ogive de Galton. L'option de correction spécifie un nombre ϵ qui est appliqué en tant que correction à la formule de rééchelonnement ; avec cette correction, toutes les valeurs de variable dépendante rééchelonnée se situeront dans la plage de la fonction d'activation. En particulier, les valeurs 0 et 1, qui sont présentes dans la formule non corrigée quand x prend sa valeur minimum et maximum, définissent les limites de la plage de la fonction d'ogive de Galton, mais ne sont pas comprises dans cette plage. La formule corrigée est $[x - (\text{min} - \epsilon)] / [(\text{max} + \epsilon) - (\text{min} - \epsilon)]$. Spécifiez une valeur supérieure ou égale à 0.
- **Normalisé ajusté** : Version ajustée de la soustraction du minimum et de la division du résultat par la plage, $[2 * (x - \text{min}) / (\text{max} - \text{min})] - 1$. Les valeurs normalisées ajustées sont comprises entre -1 et 1. Il s'agit de la méthode de rééchelonnement requise pour les variables d'échelle dépendantes si la couche de sortie utilise la fonction d'activation de tangente hyperbolique. L'option de correction spécifie un nombre ϵ qui est appliqué en tant que correction à la formule de rééchelonnement ; avec cette correction, toutes les valeurs de variable dépendante rééchelonnée se situeront dans la plage de la fonction d'activation. En particulier, les valeurs -1 et 1, qui sont présentes dans la formule non corrigée quand x prend sa valeur minimum et maximum, définissent les limites de la plage de la fonction de tangente hyperbolique, mais ne sont pas comprises dans cette plage. La formule corrigée est $\{2 * [(x - (\text{min} - \epsilon)) / ((\text{max} + \epsilon) - (\text{min} - \epsilon))] - 1\}$. Spécifiez une valeur supérieure ou égale à 0.
- **Aucun** : Pas de rééchelonnement des variables d'échelle dépendantes.

Entraînement

L'onglet Entraînement permet de spécifier la manière dont le réseau doit être entraîné. Le type d'entraînement et l'algorithme d'optimisation déterminent les options d'entraînement disponibles.

Type d'entraînement : Le type d'entraînement détermine la manière dont le réseau traite les enregistrements. Choisissez l'une des options d'entraînement suivantes :

- **Par lot** : Met à jour les pondérations synaptiques uniquement après avoir lu tous les enregistrements de données d'apprentissage ; en d'autres termes, l'entraînement par lot utilise les informations issues de tous les enregistrements du jeu de données d'apprentissage. L'entraînement par lot est souvent préféré car il minimise directement le nombre total d'erreurs ; cependant, il peut nécessiter de mettre à jour les

pondérations de nombreuses fois jusqu'à ce que l'une des règles d'arrêt soit observée, ce qui peut nécessiter de nombreux passages de données. Elle est très utile pour les petits jeux de données.

- **En ligne :** Met à jour les pondérations synaptiques après chaque enregistrement de données d'apprentissage ; en d'autres termes, l'entraînement en ligne utilise les informations issues d'un enregistrement à la fois. L'entraînement en ligne extrait en permanence un enregistrement et met à jour les pondérations jusqu'à ce que l'une des règles d'arrêt soit observée. Si tous les enregistrements sont utilisés une fois et qu'aucune des règles d'arrêt n'est observée, le processus continue en recyclant les enregistrements de données. L'entraînement en ligne est supérieur à l'entraînement par lot pour les jeux de données "volumineux" associés à des prédicteurs ; en d'autres termes, s'il existe de nombreux enregistrements et de nombreuses entrées et que leurs valeurs ne sont pas indépendantes les unes des autres, l'entraînement en ligne peut obtenir plus rapidement une réponse raisonnable que l'entraînement par lot.
- **Par mini-lot :** Divise les enregistrements de données d'apprentissage en groupes de taille approximativement égale, puis met à jour les pondérations synaptiques après lecture d'un groupe ; en d'autres termes, l'entraînement par mini-lot utilise les informations issues d'un groupe d'enregistrements. Le processus recycle ensuite le groupe de données si nécessaire. L'entraînement par mini-lot offre un compromis entre les entraînements par lot et en ligne et peut être préférable pour les jeux de données de taille moyenne. La procédure peut automatiquement déterminer le nombre d'enregistrements d'apprentissage par mini-lot ou vous pouvez spécifier un entier supérieur à 1 ou inférieur ou égal au nombre maximal d'observations à stocker en mémoire. Vous pouvez définir le nombre maximum d'observations à stocker en mémoire dans l'onglet [Options](#).

Algorithme d'optimisation : Il s'agit de la méthode utilisée pour estimer les pondérations synaptiques.

- **Gradient conjugué échelonné :** Les hypothèses qui justifient l'utilisation des méthodes de gradient conjugué ne s'appliquant qu'aux types d'entraînement par lot, cette méthode n'est pas disponible pour l'entraînement en ligne ou par mini-lot.
- **Descente de gradient :** Cette méthode doit être utilisée avec les entraînements en ligne et par mini-lot; elle peut également être utilisée avec l'entraînement par lot.

Options d'entraînement : Les options d'entraînement vous permettent d'affiner l'algorithme d'optimisation. Vous n'aurez normalement pas besoin de modifier ces paramètres, sauf si le réseau rencontre des problèmes d'estimation.

Les options d'entraînement de l'algorithme gradient conjugué redimensionné sont les suivantes :

- **Lambda initial :** Valeur initiale du paramètre lambda de l'algorithme gradient conjugué redimensionné. Spécifiez un nombre supérieur à 0 et inférieur à 0,000001.
- **Sigma initial :** Valeur initiale du paramètre sigma de l'algorithme gradient conjugué redimensionné. Spécifiez un nombre supérieur à 0 et inférieur à 0,0001.
- **Centre d'intervalle et décalage d'intervalle :** Le centre d'intervalle (a_0) et le décalage d'intervalle (a) définissent l'intervalle $[a_0 - a, a_0 + a]$, dans lequel les vecteurs de pondération sont générés de manière aléatoire lorsque l'algorithme recuit simulé est utilisé. L'algorithme recuit simulé est utilisé pour décomposer un minimum local, dans le but d'identifier le minimum global, pendant l'application de l'algorithme d'optimisation. Cette approche est utilisée dans l'initialisation de pondération et la sélection automatique de l'architecture. Spécifiez un nombre pour le centre d'intervalle et un nombre supérieur à 0 pour le décalage d'intervalle.

Les options d'entraînement de l'algorithme de descente de gradient sont les suivantes :

- **Taux d'apprentissage initial :** Valeur initiale du taux d'apprentissage de l'algorithme de descente de gradient. Un taux d'apprentissage plus élevé signifie que le réseau se formera plus rapidement, au risque de devenir instable. Spécifiez un nombre supérieur à 0.
- **Limite inférieure du taux d'apprentissage :** Limite inférieure du taux d'apprentissage de l'algorithme de descente de gradient. Ce paramètre ne s'applique qu'aux formations en ligne et par mini-lot. Spécifiez un nombre supérieur à 0 et inférieur au taux d'apprentissage initial.
- **Vitesse :** Paramètre de vitesse initial de l'algorithme de descente de gradient. Ce paramètre permet d'empêcher les instabilités causées par un taux d'apprentissage trop élevé. Spécifiez un nombre supérieur à 0.

- **Réduction du taux d'apprentissage, par période :** Le nombre de périodes (p), ou passages de données de l'échantillon d'apprentissage, pour réduire le taux d'apprentissage initial à la limite inférieure du taux d'apprentissage lorsque l'algorithme de descente de gradient est utilisé avec l'entraînement en ligne ou par mini-lot. Vous pouvez ainsi contrôler le facteur de diminution du taux d'apprentissage $\beta = (1/p \cdot K) \cdot \ln(\eta_0/\eta_{low})$, où η_0 représente le taux d'apprentissage initial, η_{low} la borne inférieure du taux d'apprentissage et K le nombre total de mini-lots (ou le nombre d'enregistrements d'apprentissage pour l'entraînement en ligne) dans le jeu de données d'apprentissage. Entrez un entier supérieur à 0.

Sortie

Structure de réseau : Affiche des informations récapitulatives sur le réseau neuronal.

- **Description :** Affiche des informations sur le réseau neuronal, y compris les variables dépendantes, le nombre d'unités d'entrée et de sortie, le nombre de couches et d'unités masquées, ainsi que les fonctions d'activation.
- **Diagramme :** Affiche le diagramme réseau sous forme de graphique non modifiable. A mesure que le nombre de covariables et de niveaux de facteur augmente, le diagramme devient plus difficile à interpréter.
- **Pondérations synaptiques :** Affiche les estimations de coefficients qui indiquent la relation existant entre les unités d'une couche donnée et celles de la couche suivante. Les pondérations synaptiques sont basées sur l'échantillon d'apprentissage même si le jeu de données actif est partitionné en données d'apprentissage, de test et restantes. Le nombre de pondérations synaptiques peut être élevé et ces pondérations ne sont généralement pas utilisées pour interpréter les résultats du réseau.

Performances réseau : Affiche les résultats utilisés pour déterminer si le modèle est correct. *Remarque :* Les graphiques figurant dans ce groupe sont basés sur les échantillons d'apprentissage et de test combinés, ou uniquement sur l'échantillon d'apprentissage s'il n'existe aucun échantillon de test.

- **Récapitulatif du modèle :** Affiche un récapitulatif des résultats du réseau neuronal par partition et globalement, y compris les erreurs, les erreurs ou les pourcentages relatifs de prévisions incorrectes, la règle d'arrêt utilisée pour arrêter l'entraînement et le temps d'entraînement.

L'erreur est l'erreur de somme des carrés lorsque la fonction d'activation d'identité, d'ogive de Galton ou de tangente hyperbolique est appliquée à la couche de sortie. Il s'agit de l'erreur d'entropie croisée lorsque la fonction d'activation de Softmax est appliquée à la couche de sortie.

Les erreurs ou les pourcentages relatifs de prévisions incorrectes sont affichés en fonction des niveaux de mesure de variable dépendante. Si une variable dépendante comporte un niveau de mesure d'échelle, l'erreur relative globale moyenne (par rapport au modèle moyen) est affichée. Si toutes les variables dépendantes sont des variables catégorielles, le pourcentage moyen de prévisions incorrectes est affiché. Les erreurs ou les pourcentages relatifs de prévisions incorrectes sont également affichés pour les variables dépendantes individuelles.

- **Résultats du classement :** Affiche une table de classification pour chaque variable dépendante catégorielle par partition et globalement. Chaque tableau indique le nombre d'observations classées correctement et incorrectement pour chaque catégorie de variable dépendante. Le pourcentage d'observations totales ayant été correctement classées est également indiqué.
- **Courbe ROC :** Affiche une courbe ROC (Receiver Operating Characteristic) pour chaque variable dépendante catégorielle. Affiche également un tableau indiquant la zone au-dessous de chaque courbe. Pour une variable dépendante donnée, le graphique ROC affiche une courbe pour chaque catégorie. Si la variable dépendante comporte deux catégories, chaque courbe traite la catégorie en question comme étant l'état positif par rapport à l'autre catégorie. Si la variable dépendante comporte plus de deux catégories, chaque courbe traite la catégorie en question comme étant l'état positif par rapport à la somme de toutes les autres catégories.
- **Graphique de gains cumulés :** Affiche un graphique de gains cumulés pour chaque variable dépendante catégorielle. L'affichage d'une courbe pour chaque catégorie de variable dépendante est identique à celui des courbes ROC.

- **Graphique de lift** : Affiche un graphique de lift pour chaque variable dépendante catégorielle. L'affichage d'une courbe pour chaque catégorie de variable dépendante est identique à celui des courbes ROC.
- **Graphique estimé/observé** : Affiche un graphique estimé/observé pour chaque variable dépendante. Pour les variables dépendantes catégorielles, des boîtes à moustaches juxtaposées des pseudo-probabilités prévues sont affichées pour chaque catégorie de réponse, avec la catégorie de réponse observée comme variable de cluster. Pour les variables d'échelle dépendantes, un nuage de points est affiché.
- **Graphique résiduel/estimé** : Affiche un graphique résiduel/estimé pour chaque variable d'échelle dépendante. Il ne doit exister aucun motif visible entre les résidus et les prévisions. Ce graphique n'est généré que pour les variables d'échelle dépendantes.

Récapitulatif du traitement des observations : Affiche la table récapitulative de traitement des observations, qui récapitule le nombre d'observations incluses et exclues dans l'analyse, au total et par échantillon d'apprentissage, de test et restant.

Analyse de l'importance des variables indépendantes : Effectue une analyse de sensibilité, qui calcule l'importance de chaque prédicteur dans la détermination du réseau neuronal. L'analyse est basée sur les échantillons d'apprentissage et de test combinés, ou uniquement sur l'échantillon d'apprentissage s'il n'existe aucun échantillon de test. Ceci produit un tableau et un graphique qui indiquent l'importance et l'importance normalisée de chaque prédicteur. L'analyse de sensibilité nécessite beaucoup de calculs et de temps si les prédicteurs ou les observations sont nombreuses.

Enregistrer

L'onglet Enregistrer permet d'enregistrer les prévisions en tant que variables dans le jeu de données.

- **Enregistrer la valeur ou la catégorie prévue pour chaque variable dépendante** : Cette option enregistre la valeur prévue pour les variables d'échelle dépendantes et la catégorie prévue pour les variables dépendantes catégorielles.
- **Enregistrer la pseudo-probabilité prévue ou la catégorie pour chaque variable dépendante** : Cette option enregistre les pseudo-probabilités prévues pour les variables dépendantes catégorielles. Une variable distincte est enregistrée pour chacune des n premières catégories, n étant spécifié dans la colonne **Catégories à enregistrer**.

Noms des variables enregistrées : Grâce à la génération automatique de nom, vous conservez l'ensemble de votre travail. Les noms personnalisés vous permettent de supprimer/remplacer les résultats d'exécutions précédentes sans supprimer d'abord les variables enregistrées dans l'éditeur de données.

Probabilités et pseudo-probabilités

Les variables dépendantes catégorielles présentant une erreur d'activation softmax et d'entropie croisée comporteront une valeur prévue pour chaque catégorie, chaque valeur prévue étant la probabilité que l'observation appartienne à la catégorie.

Les variables dépendantes catégorielles présentant une erreur de somme des carrés comporteront une valeur prévue pour chaque catégorie, mais les valeurs prévues ne peuvent pas être interprétées comme probabilités. La procédure enregistre ces pseudo-probabilités prévues même si certaines d'entre elles sont inférieures à zéro ou supérieures à 1, ou si la somme d'une variable dépendante donnée n'est pas égale à 1.

Les graphiques de ROC, de gains cumulés et de lift (voir «[Sortie](#)», à la page 9) sont créés en fonction des pseudo-probabilités. Si des pseudo-probabilités sont inférieures à 0 ou supérieures à 1 ou que la somme d'une variable donnée n'est pas égale à 1, elles sont d'abord rééchelonnées pour se situer entre 0 et 1, et avoir pour somme 1. Les pseudo-probabilités sont rééchelonnées en étant divisées par leur somme. Par exemple, si une observation comporte des pseudo-probabilités de 0,50, 0,60 et 0,40 pour une variable dépendante à trois catégories, chaque pseudo-probabilité est alors divisée par la somme 1,50 afin d'obtenir 0,33, 0,40 et 0,27.

Si des pseudo-probabilités sont négatives, la valeur absolue de la plus faible est ajoutée à toutes les pseudo-probabilités avant le rééchelonnement ci-dessus. Par exemple, si les pseudo-probabilités sont -0,30, 0,50 et 1,30, ajoutez d'abord 0,30 à chaque valeur pour obtenir 0,00, 0,80 et 1,60. Divisez ensuite chaque nouvelle valeur par la somme 2,40 pour obtenir 0,00, 0,33 et 0,67.

Exportation

L'onglet Exporter permet d'enregistrer les estimations des pondérations synaptiques de chaque variable dépendante dans un fichier XML (PMML). Vous pouvez utiliser ce fichier de modèle pour appliquer les informations du modèle aux autres fichiers de données à des fins d'évaluation. Cette option n'est pas disponible si des fichiers scindés ont été définis.

Options

Valeurs manquantes de l'utilisateur : Les facteurs doivent avoir des valeurs valides pour qu'une observation puisse être incluse dans l'analyse. Ces contrôles vous permettent d'indiquer si les valeurs manquantes de l'utilisateur sont considérées comme valides parmi les facteurs et les variables dépendantes catégorielles.

Règles d'arrêt : Ces règles déterminent le moment où l'entraînement du réseau neuronal doit être arrêté. L'entraînement se poursuit avec au moins un passage de données. L'entraînement peut être arrêté en fonction des critères suivants, qui sont sélectionnés dans l'ordre indiqué. Dans les définitions de règle d'arrêt qui suivent, une étape correspond à un passage de données pour les méthodes en ligne et par mini-lot, ainsi qu'à une itération pour la méthode par lot.

- **Nombre maximal d'étapes sans réduire le nombre d'erreurs :** Nombre d'étapes à autoriser avant de vérifier une baisse du nombre d'erreurs. Si le nombre d'erreurs ne diminue pas après le nombre spécifié d'étapes, l'entraînement s'arrête. Spécifiez un entier supérieur à 0. Vous pouvez également spécifier l'échantillon de données utilisé pour calculer les erreurs. L'option **Sélectionner automatiquement** utilise l'échantillon de test s'il existe et, dans le cas contraire, l'échantillon d'apprentissage. L'entraînement par lot garantit la réduction du nombre d'erreurs d'échantillon d'apprentissage après chaque passage de données. Par conséquent, cette option ne s'applique qu'à l'entraînement par lot s'il existe un échantillon de test. L'option **Données d'apprentissage et de test** vérifie les erreurs pour chacun de ces échantillons ; cette option ne s'applique que s'il existe un échantillon de test.

Remarque : Après chaque passage complet de données, les entraînements en ligne et par mini-lot nécessitent un passage supplémentaire de données pour calculer l'erreur d'entraînement. Ce passage supplémentaire de données pouvant ralentir considérablement l'entraînement, il est généralement recommandé de fournir un échantillon de test et de sélectionner **Sélectionner automatiquement** dans tous les cas.

- **Durée maximale d'entraînement :** Choisissez de spécifier ou non un nombre maximal de minutes pour l'exécution de l'algorithme. Spécifiez un nombre supérieur à 0.
- **Nombre maximal de périodes d'entraînement :** Nombre maximal de périodes (passages de données) autorisé. Si le nombre maximal de périodes est dépassé, l'entraînement s'arrête. Entrez un entier supérieur à 0.
- **Modification relative minimale de l'erreur d'entraînement :** L'entraînement s'arrête si le changement relatif dans les erreurs d'entraînement par rapport à l'étape précédente est inférieur à la valeur de critère. Spécifiez un nombre supérieur à 0. Pour les entraînements en ligne et par mini-lot, ce critère est ignoré si les données de test sont les seules à être utilisées pour calculer les erreurs.
- **Modification relative minimale du rapport d'erreur d'entraînement :** L'entraînement s'arrête si le rapport entre erreurs d'entraînement et erreurs du modèle nul est inférieur à la valeur de critère. Le modèle nul prévoit la valeur moyenne de toutes les variables dépendantes. Spécifiez un nombre supérieur à 0. Pour les entraînements en ligne et par mini-lot, ce critère est ignoré si les données de test sont les seules à être utilisées pour calculer les erreurs.

Nombre maximal d'observations à stocker en mémoire : Cette option contrôle les paramètres suivants dans les algorithmes de perceptron multicouche. Entrez un entier supérieur à 1.

- Lors de la sélection automatique de l'architecture, la taille de l'échantillon permet de déterminer si la taille de l'architecture réseau est $\min(1000, memsize)$, $memsize$ représentant le nombre maximal d'observations à stocker en mémoire.
- Lors de l'entraînement par mini-lot avec calcul automatique du nombre de mini-lot, le nombre de mini-lot est $\min(\max(M/10, 2), memsize)$, M représentant le nombre d'observations de l'échantillon d'apprentissage.

fonction de base radiale

La procédure de fonction à base radiale (RBF) produit un modèle de prévision pour une ou plusieurs variables dépendantes (cibles) en fonction des valeurs des variables de prédicteur.

Exemple : Un fournisseur de services de télécommunication a segmenté sa base de clients par motif d'utilisation des services en catégorisant les clients en quatre groupes. Un réseau RBF utilisant des données démographiques pour prévoir les groupes d'affectations permet à l'entreprise de personnaliser les offres pour chaque client éventuel.












Analyse des données

Variables dépendantes : Les variables dépendantes peuvent être :

- *Nominal.* Une variable peut être traitée comme étant nominale si ses valeurs représentent des catégories sans classement intrinsèque (par exemple, le service de la société dans lequel travaille un employé). La région, le code postal ou l'appartenance religieuse sont des exemples de variables nominales.
- *Ordinal.* Une variable peut être traitée comme étant ordinale si ses valeurs représentent des catégories associées à un classement intrinsèque (par exemple, des niveaux de satisfaction allant de Très mécontent à Très satisfait). Exemples de variable ordinale : des scores d'attitude représentant le degré de satisfaction ou de confiance, et des scores de classement des préférences.
- *Echelle.* Une variable peut être traitée comme une variable d'échelle (continue) si ses valeurs représentent des catégories ordonnées avec une mesure significative, de sorte que les comparaisons de distance entre les valeurs soient adéquates. L'âge en années et le revenu en milliers de dollars sont des exemples de variable d'échelle.

La procédure considère que le niveau de mesure approprié a été assigné à toutes les variables dépendantes, bien que vous puissiez changer provisoirement le niveau de mesure d'une variable en cliquant avec le bouton droit de la souris sur la variable dans la liste des variables source, puis en sélectionnant un niveau de mesure dans le menu contextuel.

Dans la liste des variables, une icône indique le niveau de mesure et le type de données :

	Numérique	Chaîne	Date	Heure
Echelle (continue).		n/a		
Ordinal				
Champs nominaux				

Variables de prédicteur : Les variables de prédicteur peuvent être spécifiées en tant que facteurs (qualitatifs) ou covariables (d'échelle).

Codification des variables indicatrices : La procédure recode provisoirement les variables de prédicteur catégorielles et les variables dépendantes via la codification un-de-c pour la durée de la procédure. S'il

existe des catégories c d'une variable, la variable est stockée comme vecteurs c , la première catégorie étant identifiée par $(1,0,\dots,0)$, la suivante par $(0,1,0,\dots,0)$, ... et la dernière par $(0,0,\dots,0,1)$.

Ce système de codification augmente le nombre de pondérations synaptiques et peut résulter en un entraînement plus lent, mais les méthodes de codification plus compactes aboutissent généralement à des réseaux neuronaux mal ajustés. Si l'entraînement de votre réseau s'effectue très lentement, essayez de réduire le nombre de catégories dans vos prédicteurs catégoriels en combinant des catégories similaires ou en supprimant les observations comportant des catégories extrêmement rares.

Toute codification un-de-c repose sur les données d'apprentissage, même si un échantillon de test ou restant est défini (voir «Partitions », à la page 14). Ainsi, si les échantillons de test ou restants contiennent des observations avec des catégories de prédicteurs ne figurant pas dans les données d'apprentissage, ces observations ne sont pas utilisées par la procédure ou dans l'évaluation. Si les échantillons de test ou restants contiennent des observations avec des catégories de variables dépendantes ne figurant pas dans les données d'apprentissage, ces observations ne sont pas utilisées par la procédure mais peuvent être notées.

Rééchelonnement : Les variables d'échelle dépendantes et les covariables sont rééchelonnées par défaut pour améliorer l'entraînement du réseau. Le rééchelonnement repose entièrement sur les données d'apprentissage, même si un échantillon de test ou restant est défini (voir «Partitions », à la page 14). En d'autres termes, en fonction du type de rééchelonnement, la moyenne, l'écart type, la valeur minimale ou la valeur maximale d'une covariable ou d'une variable dépendante ne sont calculées qu'à l'aide des données d'apprentissage. Si vous spécifiez une variable pour définir des partitions, il est important que ces covariables ou variables dépendantes présentent des distributions similaires à travers les échantillons d'apprentissage, de test et restants.

Pondérations de fréquence : Cette procédure ignore les pondérations de fréquence.

Réplication de résultats : Si vous souhaitez répliquer vos résultats exactement, outre les mêmes paramètres de procédure, utilisez la même valeur d'initialisation pour le générateur de nombres aléatoires et le même ordre de données. Vous trouverez ci-après plus de détails sur cet aspect.

- **Génération de nombres aléatoires :** La procédure utilise la génération de nombres aléatoires pendant l'affectation aléatoire des partitions. Pour reproduire les mêmes résultats aléatoires à l'avenir, utilisez la même valeur d'initialisation pour le générateur de nombres aléatoires avant chaque exécution de la procédure de fonction à base radiale. Reportez-vous à pour des instructions détaillées.
- **Tri par observation :** Les résultats dépendent également de l'ordre des données, car l'algorithme de classification en deux étapes intervient dans la détermination des fonctions à base radiale.

Pour réduire les effets de tri, classez les observations de manière aléatoire. Pour vérifier la stabilité d'une solution donnée, vous pouvez obtenir différentes solutions dans lesquelles les observations sont triées de différentes manières aléatoires. Si les fichiers sont très volumineux, vous pouvez effectuer plusieurs fois l'opération sur un échantillon des observations triées de différentes manières aléatoires.

Création d'un réseau de fonction à base radiale

A partir des menus, sélectionnez :

Analyse > Réseaux neuronaux > Fonction de base radiale...

1. Sélectionnez au moins une variable dépendante.
2. Sélectionnez au moins un facteur ou une covariable.

Dans l'onglet Variables, vous pouvez aussi changer la méthode de rééchelonnement des covariables. Les choix sont les suivants :

- **Standardisé :** Soustrayez la moyenne et divisez le résultat par l'écart type, $(x - \text{moyenne})/s$.
- **Normalisé :** Soustrayez le minimum et divisez le résultat par la plage, $(x - \text{min})/(\text{max} - \text{min})$. Les valeurs normalisées sont comprises entre 0 et 1.
- **Normalisé ajusté :** Version ajustée de la soustraction du minimum et de la division du résultat par la plage, $[2 * (x - \text{min}) / (\text{max} - \text{min})] - 1$. Les valeurs normalisées ajustées sont comprises entre -1 et 1.
- **Aucun :** Aucun rééchelonnement des covariables.

Champs avec un niveau de mesure inconnu

L'alerte du niveau de mesure apparaît lorsque le niveau de mesure d'une ou plusieurs variables (champs) du jeu de données est inconnu. Le niveau de mesure ayant une incidence sur le calcul des résultats de cette procédure, toutes les variables doivent avoir un niveau de mesure défini.

Analyser les données : Lit les données dans le jeu de données actifs et attribue le niveau de mesure par défaut à tous les champs ayant un niveau de mesure inconnu. Si le jeu de données est important, cette action peut prendre un certain temps.

Affecter manuellement : Ouvre une boîte de dialogue qui répertorie tous les champs ayant un niveau de mesure inconnu. Vous pouvez utiliser cette boîte de dialogue pour attribuer un niveau de mesure à ces champs. Vous pouvez également attribuer un niveau de mesure dans la vue de variable de l'éditeur de données.

Le niveau de mesure étant important pour cette procédure, vous ne pouvez pas accéder à la boîte de dialogue d'exécution de cette procédure avant que tous les champs n'aient des niveaux de mesure définis.

Partitions

Jeu de données de partition : Ce groupe indique la méthode de partitionnement du jeu de données actif en échantillons d'apprentissage, de test et restants. L'**échantillon d'apprentissage** comprend les enregistrements de données utilisés pour entraîner le réseau neuronal. Un certain pourcentage d'observations contenues dans le jeu de données doit être affecté à l'échantillon d'apprentissage pour l'obtention d'un modèle. L'**échantillon de test** est un ensemble indépendant d'enregistrements de données utilisé pour identifier des erreurs au cours de l'entraînement afin d'empêcher un entraînement excessif. Nous vous conseillons fortement de créer un échantillon de test. Un entraînement du réseau sera en général plus efficace si l'échantillon de test est plus petit que l'échantillon d'apprentissage. L'**échantillon restant** est un autre ensemble indépendant d'enregistrements de données utilisé pour évaluer le réseau neuronal final ; l'erreur pour l'échantillon restant donne une estimation "honnête" de la capacité de prévision du modèle parce que les observations restantes n'ont pas été utilisées pour construire le modèle.

- **Affecter des observations de façon aléatoire en fonction du nombre d'observations relatif :** Indiquez le nombre d'observations relatif (rapport) affecté de façon aléatoire à chaque échantillon (d'apprentissage, de test et restant). La colonne **%** rapporte le pourcentage d'observations qui seront affectées à chaque échantillon en fonction des nombres relatifs que vous avez spécifiés.

Par exemple, si vous spécifiez 7, 3 et 0 comme nombres relatifs pour les échantillons d'apprentissage, de test et restants, ces valeurs correspondent à 70 %, 30 % et 0 %. Si vous spécifiez 2, 1, 1 comme nombres relatifs, ces valeurs correspondent à 50 %, 25 % et 25 %. 1, 1 et 1 correspond à la division du jeu de données en tiers égaux entre l'apprentissage, le test et les données restantes.

- **Utiliser une variable de partitionnement pour affecter des observations :** Indiquez une variable numérique qui affecte chaque observation du jeu de données actif à l'échantillon d'apprentissage, de test et restant. Les observations contenant une valeur positive sur la variable sont affectées à l'échantillon d'apprentissage, celles contenant une valeur égale à 0 sont affectées à l'échantillon de test, et celles contenant une valeur négative sont affectées à l'échantillon restant. Les observations contenant des valeurs système manquantes sont exclues de l'analyse. Les valeurs manquantes de l'utilisateur pour la variable de partitionnement sont toujours considérées comme étant valides.

Architecture

L'onglet Architecture permet de spécifier la structure du réseau. La procédure crée un réseau neuronal avec une couche de "fonction à base radiale" masquée ; en général, il n'est pas nécessaire de modifier ces paramètres.

Nombre d'unités de la couche masquée : Vous pouvez choisir le nombre d'unités masquées de trois façons.

1. **Rechercher le nombre optimal d'unités dans une plage calculée automatiquement** : La procédure calcule automatiquement les valeurs minimale et maximale de la plage et recherche le nombre optimal d'unités masquées à l'intérieur de la plage.
Si un échantillon de test est défini, la procédure utilise le critère de données de test : le "meilleur" nombre d'unités masquées est celui qui renvoie le plus petit nombre d'erreurs dans les données de test. Si aucun échantillon de test n'est défini, la procédure utilise le critère d'information bayésien (BIC) : le "meilleur" nombre d'unités masquées est celui qui renvoie le plus petit BIC dans les données d'apprentissage.
2. **Rechercher le nombre optimal d'unités dans une plage spécifique** : Vous pouvez indiquer la plage de votre choix afin que la procédure y recherche le nombre "optimal" d'unités masquées. Comme dans la méthode précédente, le nombre optimal d'unités masquées dans la plage est déterminé à l'aide du critère de données de test ou du critère d'information bayésien.
3. **Utiliser le nombre d'unités indiqué** : Vous pouvez passer outre à l'utilisation d'une plage et indiquer directement un nombre d'unités spécifique.

Fonction d'activation pour la couche masquée : La fonction d'activation pour la couche masquée est la fonction à base radiale, qui "lie" les unités d'une couche aux valeurs des unités de la suivante. Pour la couche de sortie, la fonction d'activation est la fonction d'identité ; les unités de sortie sont donc simplement les sommes pondérées des unités masquées.

- **Fonction à base radiale normalisée** : Utilise la fonction d'activation softmax afin que les activations de toutes les unités masquées soient normalisées pour être égales à un.
- **Fonction à base radiale ordinaire** : Utilise la fonction d'activation exponentielle afin que l'activation de l'unité masquée soit une "bosse" gaussienne en guise de fonction des entrées.

Chevaucher les unités masquées : Le facteur de chevauchement est un multiplicateur appliqué à la largeur des fonctions à base radiale. Valeur automatiquement calculée du facteur de chevauchement $1+0,1d$, où d représente le nombre d'unités d'entrée (somme du nombre de catégories dans tous les facteurs et du nombre de covariables).

Sortie

Structure de réseau : Affiche des informations récapitulatives sur le réseau neuronal.

- **Description** : Affiche des informations sur le réseau neuronal, y compris les variables dépendantes, le nombre d'unités d'entrée et de sortie, le nombre de couches et d'unités masquées, ainsi que les fonctions d'activation.
- **Diagramme** : Affiche le diagramme réseau sous forme de graphique non modifiable. A mesure que le nombre de covariables et de niveaux de facteur augmente, le diagramme devient plus difficile à interpréter.
- **Pondérations synaptiques** : Affiche les estimations de coefficients qui indiquent la relation existant entre les unités d'une couche donnée et celles de la couche suivante. Les pondérations synaptiques sont basées sur l'échantillon d'apprentissage même si le jeu de données actif est partitionné en données d'apprentissage, de test et restantes. Le nombre de pondérations synaptiques peut être élevé et ces pondérations ne sont généralement pas utilisées pour interpréter les résultats du réseau.

Performances réseau : Affiche les résultats utilisés pour déterminer si le modèle est correct. *Remarque* : Les graphiques figurant dans ce groupe sont basés sur les échantillons d'apprentissage et de test combinés, ou uniquement sur l'échantillon d'apprentissage s'il n'existe aucun échantillon de test.

- **Récapitulatif du modèle** : Affiche un récapitulatif des résultats du réseau neuronal par partition et globalement, y compris l'erreur, l'erreur relative ou le pourcentage de prévisions incorrectes, ainsi que la durée d'entraînement.

L'erreur est l'erreur de la somme des carrés. Apparaissent également les erreurs relatives ou les pourcentages de prévisions incorrectes, suivant les niveaux de mesure des variables dépendantes. Si une variable dépendante comporte un niveau de mesure d'échelle, l'erreur relative globale moyenne (par rapport au modèle moyen) est affichée. Si toutes les variables dépendantes sont des variables catégorielles, le pourcentage moyen de prévisions incorrectes est affiché. Les erreurs ou les

pourcentages relatifs de prévisions incorrectes sont également affichés pour les variables dépendantes individuelles.

- **Résultats du classement** : Affiche une table de classification pour chaque variable dépendante catégorielle. Chaque tableau indique le nombre d'observations classées correctement et incorrectement pour chaque catégorie de variable dépendante. Le pourcentage d'observations totales ayant été correctement classées est également indiqué.
- **Courbe ROC** : Affiche une courbe ROC (Receiver Operating Characteristic) pour chaque variable dépendante catégorielle. Affiche également un tableau indiquant la zone au-dessous de chaque courbe. Pour une variable dépendante donnée, le graphique ROC affiche une courbe pour chaque catégorie. Si la variable dépendante comporte deux catégories, chaque courbe traite la catégorie en question comme étant l'état positif par rapport à l'autre catégorie. Si la variable dépendante comporte plus de deux catégories, chaque courbe traite la catégorie en question comme étant l'état positif par rapport à la somme de toutes les autres catégories.
- **Graphique de gains cumulés** : Affiche un graphique de gains cumulés pour chaque variable dépendante catégorielle. L'affichage d'une courbe pour chaque catégorie de variable dépendante est identique à celui des courbes ROC.
- **Graphique de lift** : Affiche un graphique de lift pour chaque variable dépendante catégorielle. L'affichage d'une courbe pour chaque catégorie de variable dépendante est identique à celui des courbes ROC.
- **Graphique estimé/observé** : Affiche un graphique estimé/observé pour chaque variable dépendante. Pour les variables dépendantes catégorielles, des boîtes à moustaches juxtaposées des pseudo-probabilités prévues sont affichées pour chaque catégorie de réponse, avec la catégorie de réponse observée comme variable de cluster. Pour les variables d'échelle dépendantes, un nuage de points est affiché.
- **Graphique résiduel/estimé** : Affiche un graphique résiduel/estimé pour chaque variable d'échelle dépendante. Il ne doit exister aucun motif visible entre les résidus et les prévisions. Ce graphique n'est généré que pour les variables d'échelle dépendantes.

Récapitulatif du traitement des observations : Affiche la table récapitulative de traitement des observations, qui récapitule le nombre d'observations incluses et exclues dans l'analyse, au total et par échantillon d'apprentissage, de test et restant.

Analyse de l'importance des variables indépendantes : Effectue une analyse de sensibilité, qui calcule l'importance de chaque prédicteur dans la détermination du réseau neuronal. L'analyse est basée sur les échantillons d'apprentissage et de test combinés, ou uniquement sur l'échantillon d'apprentissage s'il n'existe aucun échantillon de test. Ceci produit un tableau et un graphique qui indiquent l'importance et l'importance normalisée de chaque prédicteur. L'analyse de sensibilité nécessite beaucoup de calculs et de temps si les prédicteurs ou les observations sont nombreuses.

Enregistrer

L'onglet Enregistrer permet d'enregistrer les prévisions en tant que variables dans le jeu de données.

- **Enregistrer la valeur ou la catégorie prévue pour chaque variable dépendante** : Cette option enregistre la valeur prévue pour les variables d'échelle dépendantes et la catégorie prévue pour les variables dépendantes catégorielles.
- **Enregistrer la pseudo-probabilité prévue pour chaque variable dépendante** : Cette option enregistre les pseudo-probabilités prévues pour les variables dépendantes catégorielles. Une variable distincte est enregistrée pour chacune des n premières catégories, n étant spécifié dans la colonne *Catégories à enregistrer*.

Noms des variables enregistrées : Grâce à la génération automatique de nom, vous conservez l'ensemble de votre travail. Les noms personnalisés vous permettent de supprimer ou de remplacer les résultats d'exécutions précédentes sans supprimer d'abord les variables enregistrées dans l'éditeur de données.

Probabilités et pseudo-probabilités

Les pseudo-probabilités prévues ne peuvent pas être interprétées comme des probabilités, car la procédure de fonction à base radiale utilise l'erreur de la somme des carrés et la fonction d'activation d'identité pour la couche de sortie. La procédure enregistre ces pseudo-probabilités prévues même si certaines d'entre elles sont inférieures à 0 ou supérieures à 1, ou si la somme d'une variable dépendante donnée n'est pas égale à 1.

Les graphiques de ROC, de gains cumulés et de lift (voir «Sortie », à la page 15) sont créés en fonction des pseudo-probabilités. Si des pseudo-probabilités sont inférieures à 0 ou supérieures à 1 ou que la somme d'une variable donnée n'est pas égale à 1, elles sont d'abord rééchelonnées pour se situer entre 0 et 1, et avoir pour somme 1. Les pseudo-probabilités sont rééchelonnées en étant divisées par leur somme. Par exemple, si une observation comporte des pseudo-probabilités de 0,50, 0,60 et 0,40 pour une variable dépendante à trois catégories, chaque pseudo-probabilité est alors divisée par la somme 1,50 afin d'obtenir 0,33, 0,40 et 0,27.

Si des pseudo-probabilités sont négatives, la valeur absolue de la plus faible est ajoutée à toutes les pseudo-probabilités avant le rééchelonnement ci-dessus. Par exemple, si les pseudo-probabilités sont -0,30, 0,50, et 1,30, ajoutez d'abord 0,30 à chaque valeur pour obtenir 0,00, 0,80 et 1,60. Divisez ensuite chaque nouvelle valeur par la somme 2,40 pour obtenir 0,00, 0,33 et 0,67.

Exportation

L'onglet Exporter permet d'enregistrer les estimations des pondérations synaptiques de chaque variable dépendante dans un fichier XML (PMML). Vous pouvez utiliser ce fichier de modèle pour appliquer les informations du modèle aux autres fichiers de données à des fins d'évaluation. Cette option n'est pas disponible si des fichiers scindés ont été définis.

Options

Valeurs manquantes de l'utilisateur : Les facteurs doivent avoir des valeurs valides pour qu'une observation puisse être incluse dans l'analyse. Ces contrôles vous permettent d'indiquer si les valeurs manquantes de l'utilisateur sont considérées comme valides parmi les facteurs et les variables dépendantes catégorielles.

Remarques

Le présent document a été développé pour des produits et des services proposés aux Etats-Unis et peut être mis à disposition par IBM dans d'autres langues. Toutefois, il peut être nécessaire de posséder une copie du produit ou de la version du produit dans cette langue pour pouvoir y accéder.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, programme ou service IBM n'implique pas que seul ce produit, programme ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A*

Pour toute demande d'informations relatives au jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle IBM ou envoyez vos questions par courrier à l'adresse suivante :

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

*IBM Director of Licensing
IBM Corporation*

North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret Contractuel IBM, des Conditions Internationales d'Utilisation de Logiciels IBM, des Conditions d'Utilisation du Code Machine ou de tout autre contrat équivalent.

Les données de performances et les exemples de clients sont fournis à titre d'exemple uniquement. Les performances réelles peuvent varier en fonction des configurations et des conditions d'exploitation spécifiques.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces programmes exemples sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces programmes exemples n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les programmes exemples sont fournis "EN L'ETAT", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation des programmes exemples.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© Copyright IBM Corp. 2021. Des segments de code sont dérivés des Programmes exemples d'IBM Corp.

© Copyright IBM Corp. 1989 - 2021. All rights reserved.

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark information" à l'adresse www.ibm.com/legal/copytrade.shtml.

Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium et Pentium sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que toutes les marques et tous les logos incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

Index

A

architecture
réseaux neuronaux [2](#)
architecture du réseau
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [6](#)

C

couche de sortie
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [6](#)
couche masquée
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [6](#)
Courbe ROC
dans la fonction à base radiale [15](#)
dans Perceptron multicouche [9](#)

D

diagramme réseau
dans la fonction à base radiale [15](#)
dans Perceptron multicouche [9](#)

E

échantillon d'apprentissage
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [5](#)
échantillon de test
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [5](#)
échantillon restant
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [5](#)
entraînement du réseau
dans Perceptron multicouche [7](#)
entraînement en ligne
dans Perceptron multicouche [7](#)
entraînement par lot
dans Perceptron multicouche [7](#)
entraînement par mini-lot
dans Perceptron multicouche [7](#)

F

fonction d'activation
dans la fonction à base radiale [14](#)
dans Perceptron multicouche [6](#)
fonction de base radiale
architecture du réseau [14](#)
enregistrement des variables dans le jeu de données
actif [16](#)
export de modèle [17](#)

fonction de base radiale (*suite*)
options [17](#)
partitions [14](#)
sortie [15](#)

G

graphique de lift
dans la fonction à base radiale [15](#)
dans Perceptron multicouche [9](#)
graphique des gains
dans la fonction à base radiale [15](#)
dans Perceptron multicouche [9](#)

P

Perceptron multicouche
architecture du réseau [6](#)
enregistrement des variables dans le jeu de données
actif [10](#)
entraînement [7](#)
export de modèle [11](#)
options [11](#)
partitions [5](#)
sortie [9](#)

R

règles d'arrêt
dans Perceptron multicouche [11](#)
réseaux neuronaux
architecture [2](#)

V

valeurs manquantes
dans Perceptron multicouche [11](#)

