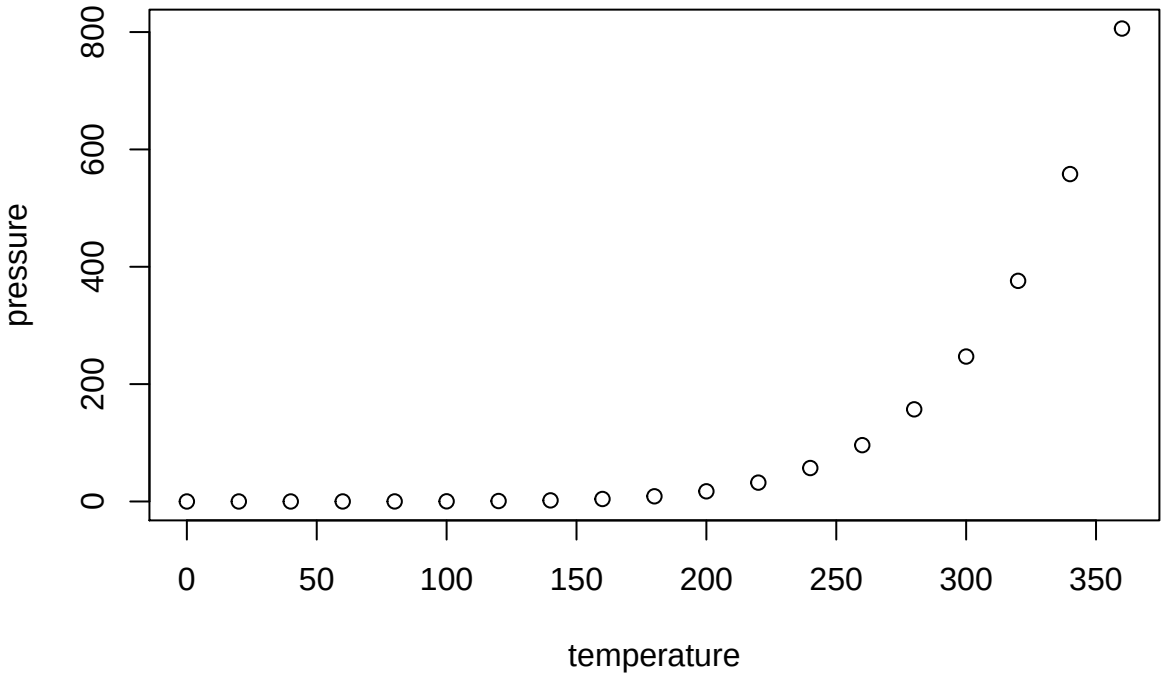
Visite des Algorithmes d'apprentissage automatique

L'apprentissage automatique

L'apprentissage automatique (en anglais : machine learning), apprentissage artificiel1 ou apprentissage statistique est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, il concerne la conception, l'analyse, l'optimisation, le développement et l'implémentation de telles méthodes.

L'apprentissage automatique comporte généralement deux phases. La première consiste à estimer un modèle à partir de données, appelées observations, qui sont disponibles et en nombre fini, lors de la phase de conception du système. L'estimation du modèle consiste à résoudre une tâche pratique, telle que traduire un discours, estimer une densité de probabilité, reconnaître la présence d'un chat dans une photographie ou participer à la conduite d'un véhicule autonome. Cette phase dite « d'apprentissage » ou « d'entraînement » est généralement réalisée préalablement à l'utilisation pratique du modèle. La seconde phase correspond à la mise en production : le modèle étant déterminé, de nouvelles données peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée. En pratique, certains systèmes peuvent poursuivre leur apprentissage une fois en production, pour peu qu'ils aient un moyen d'obtenir un retour sur la qualité des résultats produits.

Un exemple sympa d'un ensemble de lignes de meilleur ajustement.



Algorithmes regroupés par style d'apprentissage

Un algorithme peut modéliser un problème de différentes manières en fonction de son interaction avec l'expérience ou l'environnement ou selon ce que nous voulons appeler les données d'entrée.

Il est courant dans les manuels d'apprentissage automatique et d'intelligence artificielle de considérer d'abord les styles d'apprentissage qu'un algorithme peut adopter.

Un algorithme ne peut avoir que quelques styles d'apprentissage ou modèles d'apprentissage principaux et nous les passerons en revue ici avec quelques exemples d'algorithmes et de types de problèmes qui leur conviennent.

Cette taxonomie ou manière d'organiser les algorithmes d'apprentissage automatique est utile car elle vous oblige à réfléchir aux rôles des données d'entrée et au processus de préparation du modèle et à sélectionner celui qui est le plus approprié pour votre problème afin d'obtenir le meilleur résultat.

Jetons un coup d'œil à trois styles d'apprentissage différents dans les algorithmes d'apprentissage automatique:

1. Apprentissage supervisé

Algorithmes d'apprentissage supervisé lces données d'entrée sont appelées données de formation et ont une étiquette ou un résultat connu tel que spam / non-spam ou un cours boursier à la fois.

Un modèle est préparé à travers un processus de formation dans lequel il est nécessaire de faire des prédictions et est corrigé lorsque ces prédictions sont fausses. Le processus d'apprentissage se poursuit jusqu'à ce que le modèle atteigne le niveau de précision souhaité sur les données d'entraînement.

Des exemples de problèmes sont la classification et la régression.

Des exemples d'algorithmes comprennent: la régression logistique et le réseau neuronal de rétro-propagation.

2. Apprentissage non supervisé

Algorithmes d'apprentissage non supervisés les données d'entrée ne sont pas étiquetées et n'ont pas de résultat connu.

Un modèle est préparé en déduisant les structures présentes dans les données d'entrée. Cela peut être pour extraire des règles générales. Il peut s'agir d'un processus mathématique pour réduire systématiquement la redondance ou d'organiser les données par similitude.

Des exemples de problèmes sont le regroupement, la réduction de la dimensionnalité et l'apprentissage des règles d'association.

Les exemples d'algorithmes incluent: l'algorithme Apriori et K-Means.

3. Apprentissage semi-supervisé

Algorithmes d'apprentissage semi-supervisé les données d'entrée sont un mélange d'exemples étiquetés et non étiquetés.

Il existe un problème de prédiction souhaité, mais le modèle doit apprendre les structures pour organiser les données et faire des prédictions.

Des exemples de problèmes sont la classification et la régression.

Les exemples d'algorithmes sont des extensions d'autres méthodes flexibles qui émettent des hypothèses sur la façon de modéliser les données non étiquetées.

Choix d'une approche

Quelle approche est la mieux adaptée à vos besoins? Choisir un algorithme d'apprentissage supervisé ou non supervisé dépend habituellement de facteurs liés à la structure et au volume de vos données, et le cas d'utilisation auquel vous voulez l'appliquer. L'apprentissage automatique a pris de l'importance dans de nombreux secteurs, en soutenant différents objectifs et cas d'utilisation opérationnels dont :

- la valeur à vie des clients;
- la détection d'anomalies;
- la tarification dynamique;
- la maintenance prédictive;
- la classification des images;
- les moteurs de recommandation.

Présentation des algorithmes d'apprentissage automatique

Lorsque vous analysez des données pour modéliser des décisions commerciales, vous utilisez généralement des méthodes d'apprentissage supervisées et non supervisées.

Un sujet brûlant pour le moment est celui des méthodes d'apprentissage semi-supervisé dans des domaines tels que la classification d'images où il existe de grands ensembles de données avec très peu d'exemples étiquetés.

Algorithmes de régression

Algorithmes de régressionLa régression concerne la modélisation de la relation entre les variables qui est affinée de manière itérative à l'aide d'une mesure d'erreur dans les prédictions faites par le modèle.

Les méthodes de régression sont une bête de somme des statistiques et ont été cooptées dans l'apprentissage automatique statistique. Cela peut prêter à confusion car nous pouvons utiliser la régression pour désigner la classe du problème et la classe de l'algorithme. Vraiment, la régression est un processus.

Les algorithmes de régression les plus populaires sont:

- Régression des moindres carrés ordinaires (OLSR)
- Régression linéaire
- Régression logistique
- Régression pas à pas -Splines de régression adaptative multivariée (MARS)
- Lissage du nuage de points estimé localement (LOESS)

Algorithmes d'apprentissage des règles d'association

Algorithmes d'apprentissage des règles d'associationLes méthodes d'apprentissage des règles d'association extraient les règles qui expliquent le mieux les relations observées entre les variables dans les données.

Ces règles peuvent découvrir des associations importantes et commercialement utiles dans de grands ensembles de données multidimensionnels qui peuvent être exploités par une organisation.

Les algorithmes d'apprentissage des règles d'association les plus populaires sont:

- Algorithme Apriori
- Algorithme Eclat

lgorithmes d'apprentissage profond

Algorithmes d'apprentissage profondLes méthodes d' apprentissage en profondeur sont une mise à jour moderne des réseaux de neurones artificiels qui exploitent de nombreux calculs bon marché.

Ils sont concernés par la construction de réseaux de neurones beaucoup plus grands et plus complexes et, comme indiqué ci-dessus, de nombreuses méthodes concernent de très grands ensembles de données de données analogiques étiquetées, telles que des images, du texte. audio et vidéo.

Les algorithmes d'apprentissage profond les plus populaires sont:

- Réseau neuronal convolutif (CNN)
- Réseaux de neurones récurrents (RNN)
- Réseaux de mémoire à long terme (LSTM)
- Encodeurs automatiques empilés
- Machine Deep Boltzmann (DBM)
- Réseaux de croyances profondes (DBN)

Descriptions des algorithmes

Un algorithme est un îlot de recherche et dans toute la réalité, il peut être difficile de cerner la définition canonique. Par exemple, est-ce la version décrite dans la source principale ou est-ce la version qui inclut tous les correctifs et améliorations qui sont des «meilleures pratiques».

Une solution consiste à considérer un algorithme donné sous plusieurs angles, chacun pouvant avoir un objectif différent. Par exemple, la description abstraite du traitement des informations de l'algorithme pourrait être réalisée par une variété de différentes implémentations de calcul spécifiques.

J'aime cette approche car elle défend la nécessité de se télescoper sur un cas spécifique de l'algorithme à partir de nombreux cas possibles à chaque étape de la description tout en laissant également la possibilité de décrire les variations.

Il existe de nombreuses descriptions que vous pouvez utiliser de spécificité variable en fonction de vos besoins. Certains que j'aime utiliser incluent: l'inspiration pour l'algorithme, la métaphore ou l'analogie pour la stratégie, les objectifs de traitement de l'information, le pseudocode et le code.

Concevoir un modèle de description d'algorithme

Un modèle de description d'algorithme vous offre un moyen structuré de vous familiariser avec un algorithme d'apprentissage automatique.

Vous pouvez commencer avec un document vierge et lister les en-têtes de section pour les types de descriptions dont vous avez besoin de l'algorithme, par exemple appliqué, mise en œuvre ou votre propre feuille de triche de référence personnelle.

Pour déterminer les sections à inclure dans votre modèle, répertoriez les questions auxquelles vous souhaitez répondre sur l'algorithme ou les algorithmes si vous souhaitez créer une référence. Voici quelques questions que vous pourriez utiliser:

- Quelle est la norme et les abréviations utilisées pour l'algorithme?
- Quelle est la stratégie de traitement de l'information de l'algorithme?
- Quel est l'objectif ou le but de l'algorithme?
- Quelles métaphores ou analogies sont couramment utilisées pour décrire le comportement de l'algorithme?
- Quel est le pseudocode ou la description de l'organigramme de l'algorithme?
- Quelles sont les heuristiques ou les règles empiriques pour utiliser l'algorithme?
- À quelles classes de problème l'algorithme est-il bien adapté?
- Quels sont les ensembles de données de référence ou d'exemple courants utilisés pour démontrer l'algorithme?
- Quelles sont les ressources utiles pour en savoir plus sur l'algorithme?
- Quelles sont les principales références ou ressources dans lesquelles l'algorithme a été décrit pour la première fois?

Description de l'échantillon, du devis et des variables

Les données ont été recueillies initialement auprès de n = 6773 d'adolescents provenant de 12 écoles où le taux de décrochage est particulièrement élevé, autour de 36%, afin de mesurer un ensemble de facteurs de risque du décrochage scolaire. Au total, 10 des 12 écoles étaient situées dans des quartiers défavorisés. Un sous-échantillon a par la suite été invité à une entrevue afin d'établir les stresseurs auxquels les adolescents étaient exposés. L'objectif était d'interviewer 45 adolescents par école (pour un total de n = 545), suivant un devis avec cas témoins appariés. D'abord, 15 élèves qui venaient de décrocher de l'école ont été interviewés. Ensuite, 15 élèves appariés ayant un profil initial de risques similaire, mais qui persévéraient ont également été interviewés. Finalement, 15 autres élèves « normatifs », également persévérants, qui avaient un niveau moyen de risque ont été interviewés.

Variables

La variable dépendante est une variable dichotomique (codée 0 = non/1 = oui) représentant le fait qu'un élève a décroché de l'école ou non. Un élève est considéré comme décrocheur s'il remplit au moins une des trois conditions suivantes : 1) avoir avisé officiellement de la cessation de ses études avant d'obtenir son diplôme d'études secondaires ou DES, 2) avoir été transféré au système d'éducation aux adultes, 3) être absent pendant plus d'un mois de l'école sans avoir avisé la direction des motivations sous-jacentes. Pour plus de détails sur les variables et le devis, voir l'article original de Dupéré et al. (2018)Une particularité de la structure des données est que la grande proportion des variables sont ordinales et recodées en variables factices (dummy) dichotomiques. En général, les effets de la régulation sont plus marqués : 1) avec des variables intervalles/ratio puisqu'elles ont une plus grande variance et 2) en présence de multicollinéarité. Une autre caractéristique des données est que nous ne sommes pas en contexte de haute dimensionnalité puisque nous avons 25 variables pour 1000 participants, donc p « n. En dernier lieu, l'étude originale est de nature confirmatoire (hypothético-déductive) et non exploratoire (inductive), ce qui favorisera les algorithmes les moins flexibles, comme la régression logistique « classique ».

Description des variables indépendantes introduites dans le modèle de régression logistique régularisée provenant de l'étude de Dupéré et al. (2018)

Nom des variables - Types de variables - Nom des variables dans le fichier de données

1. Sexe - Dichotomique - MALE
2. Âge - Intervalle - AGE
3. Parent immigré - Dichotomique - PAR_IMM

4. Ethnicité - Dichotomique - MINORITY
5. Niveau de scolarité parents - Intervalle - SCOLMAX
6. Mère en emploi - Dichotomique - TRAVAILM
7. Père en emploi - Dichotomique - TRAVAILP
8. Parents séparés - Dichotomique - PAR_SEP
9. Adaptation scolaire - Intervalle - ADAPT
10. Risque décrochage scolaire - Intervalle - SRDQ
11. Difficultés chroniques sévères - Intervalle - CHRONSEVACT
12. Stresseurs sévères 0-3 mois - Dichotomique - SEVER03DICO
13. Stresseurs sévères 3-6 mois - Dichotomique - SEVER36DICO
14. Stresseurs sévères 6-9 mois - Dichotomique - SEVER69DICO
15. Stresseurs sévères 9-12 mois - Dichotomique - SEVER912DICO
16. Stresseurs modérés 0-3 mois - Dichotomique - MODER203DICO
17. Stresseurs modérés 3-6 mois - Dichotomique - MODER236DICO
18. Stresseurs modérés 6-9 mois - Dichotomique - MODER269DICO
19. Stresseurs modérés 9-12 mois - Dichotomique - MODER2912DICO
20. Stresseurs faibles 0-3 mois - Dichotomique - LOW203DICO
21. Stresseurs faibles 3-6 mois - Dichotomique - LOW236DICO
22. Stresseurs faibles 6-9 mois - Dichotomique - LOW269DICO
23. Stresseurs faibles 9-12 mois - Dichotomique - LOW2912DICO
24. Stresseurs distaux sévères - Intervalle/ratio - EVDISTSEV
25. Stresseurs distaux modérés - Intervalle/ratio - EVDISTMOD

Objectifs de l'analyse

L'objectif principal de cette analyse est de sélectionner un modèle de régression logistique, de manière exploratoire/inductive, en utilisant des techniques qui sont particulières à l'apprentissage automatique afin de potentiellement prédire le décrochage scolaire avec la plus grande justesse prédictive possible à partir des 25 variables indépendantes. Nous utilisons des données simulées à partir de l'échantillon initial.En résumé, la tâche de classification consiste à trouver à la fois le nombre optimal de prédicteurs du décrochagescolaire et l'algorithme de régularisation qui permettra le meilleur ajustement du modèle aux données, compte tenu de la spécificité des variables introduites dans le modèle.

Régression logistique régularisée : procédures et modèles + interprétations et taleaux de résultats

La suite du document montre le code utilisé (originalement dans le logiciel RStudio) pour la sélection du modèle conformément aux procédures décrites dans la partie 2.3 et aux résultats montrés dans la partie 2.4 du chapitre

```
# Téléchargement des packages nécessaire à l'analyse (si vous installez ces packages pour la première f
#install.packages("CUFF") #Package CUFF (Charles's Utility Function using Formula) pour affichage des v
#install.packages ("dplyr") #Package dplyr pour manipulation flexible des données
#install.packages("ggplot2") #Package ggplot2 pour création de graphiques
#install.packages("haven") #Package haven pour importer des données d'autres formats dans R
#install.packages("knitr") #Package knitr pour production de tableau
#install.packages("xtable") #Package xtable pour production de tableau
#install.packages("pairwise") #Package xtable pour production de tableau
require(dplyr, quietly = TRUE, warn.conflicts = FALSE)
require(ggplot2, quietly = TRUE, warn.conflicts = FALSE)
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'ggplot2'
```

```
require(CUFF, quietly = TRUE, warn.conflicts = FALSE)
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'CUFF'
```

```
require(haven, quietly = TRUE, warn.conflicts = FALSE)
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'haven'
```

```
require(knitr, quietly = TRUE, warn.conflicts = FALSE)
require(xtable, quietly = TRUE, warn.conflicts = FALSE)
require(pairwise, quietly = TRUE, warn.conflicts = FALSE)
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'pairwise'
```

```
opts_chunk$set(echo = TRUE, prompt = TRUE, comment = "", cache = TRUE)
options(xtable.comment = FALSE)
```