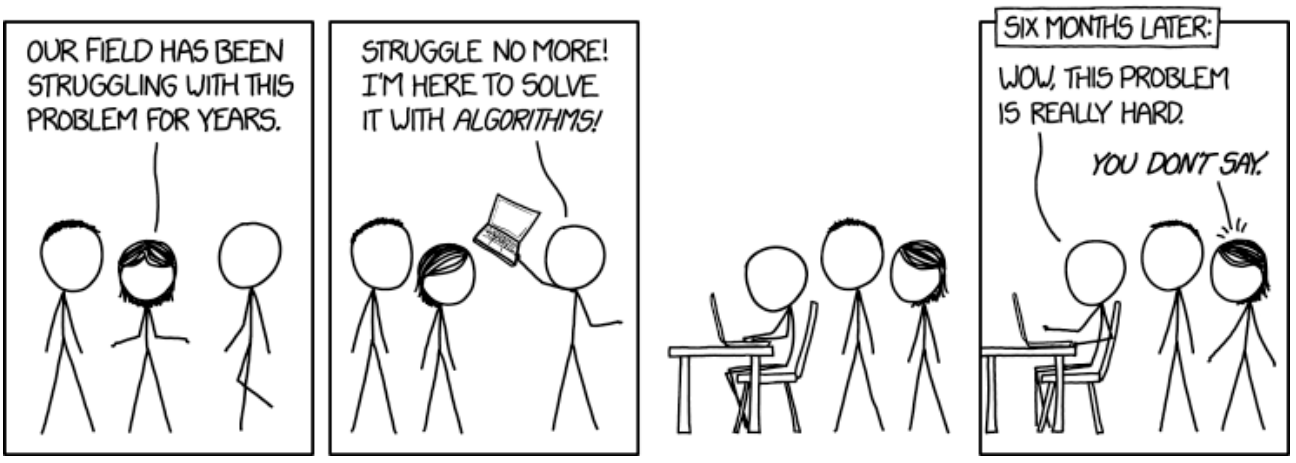
Algorithmes de Machine Learning

Clustering et autres

promo **Data+** - le 03/06/2022



Algorithms are hard

Source: xkcd

Objectifs du module

Approfondir vos connaissances et votre compréhension de certains algorithmes de machine learning

- En savoir plus sur les hyperparamètres permettant de “réguler” les algorithmes d'arbres de décision (decision trees), des forêts aléatoires (random forests) et les machines à vecteurs de support (support-vector machines)
- Utilisez les courbes d'apprentissage pour aider à évaluer les performances du modèle et suggérer des améliorations
- Comprendre le fonctionnement de l'algorithme de clustering k-means en l'implémentant vous-même

Modalités

- Durée du projet : 3 jours
- Travailler en équipes de deux
- Produire vos propres scripts et mémos individuels pour terminer le projet

Contexte

Vous avez maintenant déjà implémenté de nombreux algorithmes à l'aide de scikit-learn. Dans ce module, nous prenons le temps de mieux comprendre le fonctionnement de ces algorithmes d'apprentissage automatique. Nous prendrons également un certain temps pour mieux comprendre la manière dont les choix de modèles (et de leurs hyperparamètres) affectent le biais et la variance que nous voyons dans leur pouvoir de prédiction. Les «courbes d’apprentissage» vous permettront de facilement déceler les problèmes de biais et de variance et vous donneront donc des pistes d’amélioration de vos modèles.

Nous allons également approfondir la compréhension de l'algorithme de clustering k-means en l'implémentant nous-mêmes à partir de zéro en Python! C'est une bonne occasion de mettre en pratique vos compétences en algorithmie et en rédaction de scripts. Il est également important d'écrire du code reproductible, nous y réfléchirons également brièvement.

Etape 1

Algorithmes ML et courbes d'apprentissage

Objectifs de l'activité

- Être capable d'expliquer le fonctionnement des algorithmes suivants: l'arbre de décision,la forêt aléatoire, les SVM
- Connaître et comprendre les hyperparamètres qui existent pour "régler" ces modèles
- Tracer des courbes d'apprentissage pour ces différents algorithmes de ML et les interpréter pour expliquer comment le choix du type de modèle et des hyperparamètres affecte le **sur-ajustement** et le **sous-ajustement**

Compétences

- Expliquer ce qu'est un arbre de décision, une forêt aléatoire, un SVM
- Tracer et interpréter les courbes d'apprentissage utilisant ces différents modèles

Consignes

- Si les concepts n'étaient pas encore très clairs, prenez le temps de relire les sections du livre pour l'arbre de décision (pages 177-182), la forêt aléatoire (ch. Random Forests) et les algorithmes SVM (pages 155-158).
- Importez ces algorithmes à partir de scikit-learn et implémentez le code donné dans le livre ou les ressources sur le dataset iris.
- Lisez les Ressources 1 et 2 sur les courbes d'apprentissage.
- Mettez en œuvre le code du livre sur les courbes d'apprentissage (en utilisant les mêmes données). Assurez-vous de bien comprendre ce que fait le code. Parlez-en à vos voisins! Essayez d'adapter le code dans la Ressource 1 pour l'appliquer au même dataset que dans le livre.
- Choisissez un algorithme pour lequel vous allez ensuite produire les courbes d'apprentissage. Vous pouvez essayer le jeu de données des arbres de Grenoble depuis 'Intro au ML' ou le dataset sur les maladies cardiaques que vous avez vu en 'ML2.1'
- Créez un modèle ML qui montre des signes de sous-ajustement et un autre qui montre des signes de sur-ajustement. Essayez de modifier les hyperparamètres du modèle et voyez comment cela affecte la courbe d'apprentissage. Expliquez comment le choix du type de modèle et des hyperparamètres affecte le sur-ajustement et le sous-ajustement. Parlez-en à vos voisins!

Livrables

- Script / notebook python (ou mémo) qui contient:
    - Un exemple de courbe d'apprentissage d'un modèle qui est « underfitted »
    - Un exemple de courbe d'apprentissage d'un modèle qui est « overfitted »

Pour aller plus loin

- Essayez d'implémenter des courbes d'apprentissage sur un problème de classification que vous avez déjà rencontré dans les modules précédents.
- Essayez de produire les courbes d'apprentissage pour les autres algorithmes de ML
- Essayes de produire des courbes de validation comme montré dans la ressource Courbes de validation 1.

Ressources

General :

1. Hands on Machine Learning with scikit-learn and tensorflow, chapter 5, 6, 7

Ressources supplémentaires :

- Arbres de décision :
    1. https://scikit-learn.org/stable/modules/tree.html
    2. https://www.lovelyanalytics.com/2016/08/16/decision-tree-comment-ca-marche/
- Random forest :
    1. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random%20forest#sklearn.ensemble.RandomForestClassifier
    2. https://towardsdatascience.com/random-forest-in-python-24d0893d51c0
- SVM :
    1. https://scikit-learn.org/stable/modules/svm.html

Courbes d'apprentissage :

1. https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html#sphxglr-auto-examples-model-selection-plot-learning-curve-py
2. https://towardsdatascience.com/why-you-should-be-plotting-learning-curves-in-your-nextmachine-learning-project-221bae60c53

Courbes de validation :

1. https://scikit-learn.org/stable/modules/learning_curve.html