Le génie pour l'industrie

GTI770
Systèmes intelligents et apprentissage machine

TP02 — Arbres de décision, Bayes naïf et *KNN*

1. Contexte

Ce deuxième laboratoire porte sur l'utilisation de trois algorithmes de classification, soit les arbres de décision tels que vus dans le premier laboratoire, Bayes naïf et les *K* plus proches voisins (*KNN*). Dans ce laboratoire, vous serez amené à résoudre deux problèmes de classification :

- *Galaxy Zoo* (https://www.galaxyzoo.org/);
- Un problème de classification de courriels indésirables (*pourriels*).

Dans le premier cas, le but est de classer les images de galaxies dans les trois catégories explorées au premier laboratoire («*spirales*», «*smooth*» et «*artifact*») avec les deux nouveaux algorithmes que sont Bayes naïf et *KNN*. Dans le deuxième cas, il s'agit de classer les courriels en deux classes distinctes, soit dans la catégorie pourriel ou dans la catégorie non-pourriel, afin de concevoir, par exemple, un filtre antipourriel classique.

En vous basant sur les concepts et algorithmes vus en classe, vous devez comparer les performances de classification non seulement entre les différents algorithmes, mais également entre des données de nature différentes. Dans le cas de *Galaxy Zoo*, vous devrez utiliser les primitives trouvées lors du premier laboratoire afin d'en former un vecteur. Le total des échantillons est divisé en trois classes :

- 8132 images figurent dans la catégorie «*smooth*» (48,09 %);
- 8776 images figurent dans la catégorie «*spirals*» (51,91 %);

Dans le cas des pourriels, on suppose que l'extraction des primitives a déjà été effectuée. De ce fait, un vecteur de primitives comportant 57 dimensions vous est fourni. Cet ensemble de données portant sur les pourriels comporte :

- 1 105 vecteurs portant la catégorie pourriel (40,03 %) identifiée par le nombre entier «*1*»;
- 1 655 vecteurs associés à la catégorie non-pourriel (59,97 %) identifiée par le nombre entier «*0*».

Tout comme le premier travail pratique, vous réaliserez ce deuxième laboratoire avec la technologie Python3 conjointement avec la librairie d'apprentissage machine *scikit-learn*. Vous êtes invité à reprendre le code développé lors du laboratoire 1 afin de continuer son développement.

2. Durée du laboratoire

Vous disposez d'un total de trois (3) séances de laboratoire afin d'effectuer ce travail. En fonction de votre groupe, vous disposez des séances suivantes :

Tableau 2.1 : Date des séances et date de remise

Date des séances	GTI770-01	GTI770-02
Séance 1	11 octobre 2018	11 octobre 2018
Séance 2	18 octobre 2018	18 octobre 2018
Séance 3	25 octobre 2018	25 octobre 2018
Remise	30 octobre 2018	30 octobre 2018

3. Objectifs

Les principaux objectifs de ce laboratoire sont :

1. Produire un code source utilisant les technologies Python et des techniques d'apprentissage machine permettant de classer des échantillons de donnés automatiquement;
2. Se familiariser avec les étapes de prétraitement sur un ensemble de données;
3. Comparer les performances de classification au sein de données de même nature et de nature différente;
4. Analyser l'impact des hyperparamètres des différents algorithmes utilisés;
5. Se familiariser avec le langage de programmation *Python* et les outils et librairies scientifiques d'apprentissage machine;

4. Matériel fourni

Vous disposez du matériel suivant :

- Gabarit du rapport de laboratoire sous format Jupyter Notebook;
- Ensemble de données d'entrainement comportant 16 908 images de galaxies et un fichier CSV contenant l'étiquette associée à chaque image;
- Ensemble de données d'entrainement de 2 760 échantillons de courriels;
- Code source de base permettant de lire les fichiers images et de lire le fichier contenant le vecteur de primitives des courriels (optionnel);

5. Manipulations

- Téléchargez l'ensemble de données des pourriels ainsi que le fichier contenant l'ensemble des primitives préextraites des galaxies à partir de la plateforme *Moodle*.
- Fusionnez vos primitives à celles existantes (fiez-vous à l'annexe de cet énoncé pour ne pas répéter les mêmes primitives).

Astuces :

1. Pensez à exécuter votre code du laboratoire 1 une seule fois afin d'extraire vos primitives d'images de galaxies et de sauvegarder les valeurs numériques dans un fichier au format standard «*.csv*» avec une librairie Python gérant ce type de fichier. Cela vous évitera de réextraire vos primitives à chaque manipulation de votre ensemble de données.
2. Vous devez créer un ensemble de validation parmi vos primitives de galaxies. Créez un script permettant de choisir aléatoirement un ensemble de validation représentant 20 % de votre ensemble de données principal des galaxies (celui contenant 100 % des primitives). Vous devrez faire la même opération avec l'ensemble de données de pourriels.
3. Faites en sorte d'avoir le nombre exacte d'occurrences de chaque classe dans chaque sous ensemble de données. Vous aurez besoin de cette information ultérieurement.

- Pour les modèles entraînés avec les deux ensembles de données, plusieurs variations d'hyperparamètres sont demandées :

Tableau 5.1 : Matrice des hyperparamètres - Arbre de décision

Paramètre «*x*»	Sans contrôle de profondeur (*max depth = None*)	Avec contrôle de profondeur (*max depth = x*)
max_depth = None	X	
max_depth = 3		X
max_depth = 5		X
max_depth = 10		X

Tableau 5.2 : Matrice des hyperparamètres - KNN

Paramètre «*k*»	Poids='uniform'	Poids='distance'
K = 3	X	X
K = 5	X	X
K = 10	X	X

Tableau 5.3 : Matrice des hyperparamètres - Bayes naïf

Paramètre	MinMaxScaler	Discrétisation non-supervisée (K-Bins discretization)
Bayes naïf gaussien		
Bayes naïf multinomial	X	X

- À l'aide de la librairie *scikit-learn* et du vecteur de primitives fournis, produisez un code source permettant d'entraîner un modèle afin de classifier les pourriels avec l'aide des trois algorithmes vus en classe, soit :
  - un arbre de décision;
  - Bayes naïf;
  - *KNN*.
- Refaites le même exercice pour les galaxies avec la fusion des primitives fournis avec celles que vous avez développées au laboratoire 1. Puisque votre code implémentant un

arbre de décision a déjà été fait au laboratoire 1, vous devez produire le code afin d'entraîner un classificateur à l'aide des algorithmes :

- Bayes naïf;
- *KNN*.

- Entraînez les classificateurs à l'aide des ensembles de données portant un «X» rouge du tableau 5.1 ci-dessus. Pour chacun de ces modèles, vous devez prendre en note la précision (*accuracy*) ainsi que le score F1 (*F1-Score*) du modèle et tracer des graphiques montrant l'évolution de ces deux mesures en fonction des hyperparamètres utilisés. Le score F1 est important puisqu'il prend en considération la précision et le rappel (*recall*) des classes qui ne sont pas balancées. Le score F1 est un indice qui fournit un résultat en balançant les classes. Utilisez la classe *sklearn.metrics.f1_score()* afin de produire le score. Assurez-vous d'avoir le paramètre `average='weighted'` afin de pondérer correctement le score en fonction du nombre d'instances de chaque classe.
- Vous devez également appliquer le concept de validation croisée (*cross-validation*) au problème que vous tentez de résoudre. Pour chacun des algorithmes utilisés, sélectionnez les hyperparamètres ayant obtenu les meilleurs résultats au niveau de la précision (*accuracy*) et du *F1-Score*. Réentraînez ces modèles avec la technique *K-Fold Cross-validation*. Choisissez la valeur ***K*=10**. Prenez en note les résultats des deux mesures, soit la précision (*accuracy*) et le score F1 et produisez un tableau comparatif avec votre autre méthode de validation.

Bonus (+5 points)

À titre de question bonus, vous devez effectuer les mêmes manipulations présentées précédemment, mais avec un nouveau modèle, soit les forêts aléatoires (*random forest*). Vous devrez comparer vos résultats au même titre que ce que vous avez effectué dans la section 5 de l'énoncé, à savoir effectuer une comparaison sur la nature des données ainsi que les hyperparamètres. Vous devrez également en discuter dans le rapport et appuyez votre discussion avec la théorie relative aux forêts aléatoires.

6. Rapport

Votre rapport devra contenir les réponses aux questions suivantes ainsi que la structure présentée dans un *Jupyter Notebook* en suivant le gabarit du rapport fournit sur la plateforme *Moodle*. Il devra notamment avoir une analyse détaillée des résultats de classification obtenus par les différents modèles et leurs variations d'hyperparamètres.

6.1 Questions du rapport

1. Quelle a été votre approche de manipulation de données? Comment avez-vous créé vos ensembles de données? Quels en ont été les résultats? Détaillez les ensembles produits.
2. Parmi les méthodes de validation (*Leave-one-out cross-validation*, *Leave-p-out cross-validation, k-fold cross-validation, holdout*), présentez les approches de validation que vous avez utilisées. Quels ont été les résultats de votre comparaison des méthodes?
3. Pour chacun des modèles d'apprentissage élaborés, expliquez l'impact des hyperparamètres sur les performances des modèles.
4. Faites une discussion mettant en parallèle la nature des données. Est-ce qu'un ensemble de données se démarque par rapport à un algorithme de classification?
5. Quel type de classificateur recommanderiez-vous pour l'une et l'autre des ensembles de données et dans quelles conditions (par exemple, le nombre de données privilégié)?
6. Formulez quelques pistes d'amélioration des classificateurs.

Annexe 1 : Sources d'information pertinentes

LIENS DIVERS

Documentation de la librairie *scikit-learn* : http://scikit-learn.org/stable/documentation.html

LIVRES

Aurélien Géron. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow—Concept, Tools, and Techniques to Build Intelligent Systems.* 566 p. ISBN-13: 978–1491962299

Annexe 2 : Description des primitives des pourriels

Les courriels sont représentés par 57 attributs numériques et 1 attribut nominal. La dernière colonne indique si le courriel a été considéré comme du « spam (1) » ou « non-spam (0) ». La plupart des primitives indiquent si un mot ou un caractère particulier a été souvent produit dans le courriel. Les attributs (55-57) mesurent la longueur des séquences de lettres majuscules consécutives. Voici les définitions détaillées des attributs :

- 48 attributs continus réels [0,100] du type word_freq_WORD = Pourcentage de mots dans le courriel qui correspondent à Word, soit 100 * (nombre de fois où le mot apparaît dans le courriel)/nombre total de mots dans le courriel. Un « mot » dans ce cas est une chaîne de caractères alphanumériques délimités par des caractères non alphanumériques ou de fin de chaîne.
- 6 attributs continus réels [0,100] de type char_freq_CHAR] = Pourcentage de caractères dans le courriel qui correspondent à CHAR, soit 100 * (nombre d'occurrences de char)/ nombre total de caractères dans le courriel.
- 1 attribut continue réelle [1,...] de type capital_run_length_average = Longueur moyenne des séquences ininterrompues de lettres majuscules.
- 1 attribut entier continu [1,...] de type capital_run_length_longest = Longueur de la plus longue séquence ininterrompue de lettres majuscules.
- 1 attribut entier continu [1,...] de type capital_run_length_total = Somme de longueur des séquences ininterrompues de lettres majuscules = Nombre total de lettres majuscules dans le courriel.
- 1 concept cible nominal {0,1} de type pourriel = Indique si le courriel a été considéré comme du «pourriel (1)» ou «non-pourriel (0)».

Annexe 3 : Description des primitives des galaxies

#	Nom	Description
0	ID	ID de l'image de la galaxie.
1	Couleur moyenne du centre [0]	Sur les pixels différents de zéro de l'image originale, la couleur moyenne du centre est extraite.
2	Couleur moyenne du centre [1]	Sur les pixels différents de zéro de l'image originale, la couleur moyenne du centre est extraite.
3	Couleur moyenne du centre [2]	Sur les pixels différents de zéro de l'image originale, la couleur moyenne du centre est extraite.
4	Couleur moyenne [0]	Sur les pixels différents de zéro de l'image originale, une moyenne de la couleur *bleue* est extraite.
5	Couleur moyenne [1]	Sur les pixels différents de zéro de l'image originale, une moyenne de la couleur *vert* est extraite.
6	Couleur moyenne [2]	Sur les pixels différents de zéro de l'image originale, une moyenne de la couleur *rouge* est extraite.
7	Standard Deviation [0]	Sur les pixels différents de zéro de l'image originale, la déviation standard est extraite.
8	Standard Deviation [1]	Sur les pixels différents de zéro de l'image originale, la déviation standard est extraite.
9	Standard Deviation [2]	Sur les pixels différents de zéro de l'image originale, la déviation standard est extraite.
10	Distribution Kurtosis [0]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels différents de zéro.
11	Distribution Kurtosis [1]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels différents de zéro.
12	Distribution Kurtosis [2]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels différents de zéro.
13	Distribution normale asymétrique [0]	Sur les pixels différents de zéro, la distribution normale asymétrique est appliquée.
14	Distribution normale asymétrique [1]	Sur les pixels différents de zéro, la distribution normale asymétrique est appliquée.

#	Nom	Description
15	Distribution normale asymétrique [2]	Sur les pixels différents de zéro, la distribution normale asymétrique est appliquée.
16	Coefficient Gini [0]	Mesure de la dispersion des pixels différents de zéro.
17	Coefficient Gini [1]	Mesure de la dispersion des pixels différents de zéro.
18	Coefficient Gini [2]	Mesure de la dispersion des pixels différents de zéro.
19	Excentricité	Caractéristique de la courbure de la galaxie.
20	Largeur	Mesure de la largeur de la galaxie.
21	Hauteur	Mesure de la hauteur de la galaxie.
22	Somme	Somme des pixels de l'image *threasholded.*
23	Entropie	Mesure de l'entropie (du hasard) dans l'image.
24	Chiralité	Mesure de l'asymétrie de l'image.
25	Aire de l'ellipse	Mesure de l'aire de l'ellipse.
26	Aire *box-to-image*	Mesure de l'aire de la galaxie adaptée à une boîte.
27	Décalage du centre (*offset*)	Le décalage du centre de la galaxie par rapport à l'image.
28	Rayon de la lumière [0]	La mesure du rayon de la lumière (rayon de la galaxie).
29	Rayon de la lumière [1]	La mesure du rayon de la lumière (rayon de la galaxie).
30	Nombre de *labels*	Le nombre de caractéristiques (f*eatures*) repérées dans l'image.
31	Distribution Kurtosis [0]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels de l'image couleur.
32	Distribution Kurtosis [1]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels de l'image couleur.
33	Distribution Kurtosis [2]	Distribution kurtosis (mesure de l'aplatissement) sur les pixels de l'image couleur.
34	Distribution normale asymétrique [0]	Sur les pixels de l'image couleur, la distribution normale asymétrique est appliquée.

#	Nom	Description
35	Distribution normale asymétrique [1]	Sur les pixels de l'image couleur, la distribution normale asymétrique est appliquée.
36	Distribution normale asymétrique [2]	Sur les pixels de l'image couleur, la distribution normale asymétrique est appliquée.
37	Coefficient Gini [0]	Mesure de la dispersion des pixels de l'image en couleur.
38	Coefficient Gini [1]	Mesure de la dispersion des pixels de l'image en couleur.
39	Coefficient Gini [2]	Mesure de la dispersion des pixels de l'image en couleur.
40	Distribution Kurtosis image noir et blanc	Mesure de la dispersion des pixels de l'image en nuances de gris.
41	Distribution normale asymétrique image noir et blanc	Sur les pixels de l'image en nuances de gris, la distribution normale asymétrique est appliquée.
42	Coefficient Gini image noir et blanc	Mesure de la dispersion des pixels de l'image en nuances de gris.
43	Couleur du centre [0]	Couleur du canal *bleu* du pixel du centre.
44	Couleur du centre [1]	Couleur du canal *vert* du pixel du centre.
45	Couleur du centre [2]	Couleur du canal *rouge* du pixel du centre.
46	Couleur moyenne [0]	Sur les pixels de l'image couleur, une moyenne de la couleur *bleue* est extraite.
47	Couleur moyenne [1]	Sur les pixels de l'image couleur, une moyenne de la couleur *vert* est extraite.
48	Couleur moyenne [2]	Sur les pixels de l'image couleur, une moyenne de la couleur *rouge* est extraite.
49	Couleur moyenne du centre [0]	Sur les pixels de l'image couleur, la couleur moyenne en *bleu* du centre est extraite.
50	Couleur moyenne du centre [1]	Sur les pixels de l'image couleur la couleur moyenne en *vert* du centre est extraite.
51	Couleur moyenne du centre [2]	Sur les pixels de l'image couleur, la couleur moyenne en *rouge* du centre est extraite.
52	Couleur du centre	Couleur du pixel du centre de l'image en nuances de gris.
53	Rapport couleur du centre / moyenne de gris	Rapport.

#	Nom	Description
53 à 74	Moments de l'image	Moyenne pondérée de l'intensité des pixels des images *thresholded*, en nuances de gris et de l'image représentant la magnitude des gradients de l'image originale.
75	Label	La classe encodée associée à la galaxie.