Forest Cover Type Prediction

Kevin Zagalo
kevin.zagalo@etu.upmc.fr

Ismail Benkirane
ismail.benkirane@etu.upmc.fr

Projet pour le cours *Apprentissage Statistique* du LIP6 [0], Sorbonne Université — Janvier 2019

Résumé

Ce projet a pour but de proposer et tester des modèles pour l'étude de la base de données *Covertype* [1]. Il s'agit d'un problème de classification multi-classe avec 7 classes, et 581 012 instances de 54 attributs sans données manquantes.

Table des matières

0. Laboratoire d'Informatique de Paris 6 : : https://www.lip6.fr
1. https://archive.ics.uci.edu/ml/datasets/Covertype

1 Chargement des données

On choisit d'utiliser la bibliothèque `pandas` pour charger les données, surtout pour l'analyse préliminaire. `pandas` fournit une panoplie de fonctions pour visualiser les données. `groupby`, `boxplot` et `hist` nous seront forts utiles pour choisir les données que nous exploiterons. Le *notebook* contenant le code joint au rapport nécessite aussi les bibliothèques `matplotlib`, `numpy`, `sklearn` et `plotly`.

Les attributs sont les suivants :

Nom	Unité	Description
`Elevation`	mètres	Altitude
`Aspect`	degrés	Orientation
`Slope`	degrés	Pente
`Horizontal_Distance_To_Hydrology`	mètres	Distance horizontale au point d'eau le plus proche
`Vertical_Distance_To_Hydrology`	mètres	Distance verticale au point d'eau le plus proche
`Horizontal_Distance_To_Roadways`	mètres	Distance horizontale à la route la plus proche
`Hilshade_9am`	entier entre 0 et 255	Ombrage à 9h au solstice d'été
`Hilshade_Noon`	entier entre 0 et 255	Ombrage à 12h au solstice d'été
`Hilshade_3pm`	entier entre 0 et 255	Ombrage à 15h au solstice d'été
`Horizontal_Distance_To_Fire_Points`	mètres	Distance horizontale au départ de feu le plus proche
`Wilderness_Area`	4 colonnes binaires	Wilderness area designation
`Soil_Type`	40 colonnes binaires	Type de sol
`Cover_Type`	entier entre 1 et 7	Classe

Le problème de ce chargement est qu'il stocke les données en type `str`, il nous faut donc convertir le type des données. C'est ce que font les fonctions `convert_to_listofbool`, `convert_to_int` et `convert_to_float`.

On préfèrera garder dans le DataFrame `df_covtype` des entiers plutôt que des vecteurs binaires, quitte à les y remettre dans les données de train et de test ensuite. Cela facilitera grandement l'analyse préliminaire. On utilisera donc `wilderness` et `soil` uniquement pour la partie test des modèles.

Nos données seront donc :
- `df_covtype` : *DataFrame* de toutes les données non traitées triées par attributs.
- `labels` :*np.array* des étiquettes des types de forêts.
- `dict_attributs` : *dictionnaire* qui associe les attributs aux bons index.
- `qualitative` : *liste* des attributs des données qualitatives.
  - `Wilderness_Area` :
    - 1 : Rawah Wilderness Area
    - 2 : Neota Wilderness Area
    - 3 : Comanche Peak Wilderness Area
    - 4 : Cache la Poudre Wilderness Area
  - `Soil_Type` :

    - 1 to 40 : based on the USFS Ecological Landtype Units for this study area
  - `forest_cover_types` :
    - 1 : Spruce/Firze
    - 2 : Lodgepole Pine
    - 3 : Ponderosa Pine
    - 4 : Cottonwood/Willow
    - 5 : Aspen
    - 6 : Douglas-fir
    - 7 : Krummholz
- `wilderness` et `soil` : *listes* gardant en mémoire les vecteurs binaires pour les remplacer par des entiers. On garde donc les 44 paramètres de ces attributs dans notre modèle.

2 Analyse préliminaire et pré-traitement des données

Tout d'abord on constate sur la figure 1 que les données sont inégalement réparties selon les classes. Cela peut vouloir dire plusieurs choses : soit nos données sont mal échantillonnées, soit les types `1` et `2` sont effectivement largement plus répandues.

C'est quelque chose dont nous n'avons pas la maitrise, une discussion avec un expert sur le sujet serait préférable. Nous continuerons l'étude sans experts et en supposant que les données sont raisonnablement échantillonnées.

La suite consistera globalement à faire la même chose sur le reste des données grâce aux méthodes de la bibliothèque `pandas`.

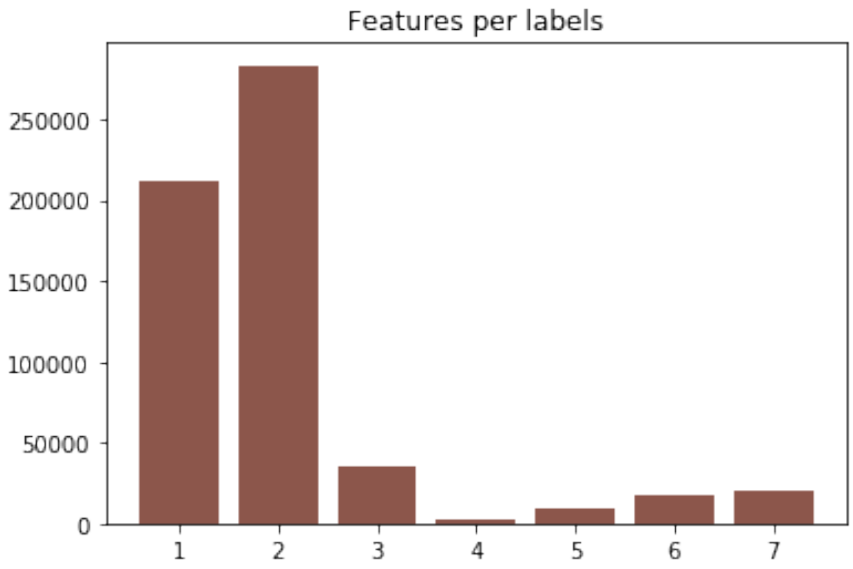
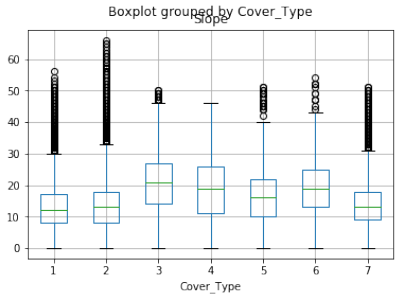


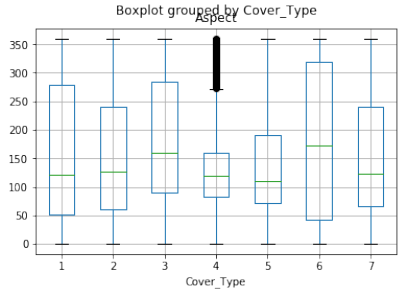
Figure 1 – Histogramme des données par types de forêts

On observe dans la figure 2 l'importance des attributs `Elevation`, `Slope`et `Aspect`. On voit aussi la très faible variation de `Vertical_Distance_To_Hydrology`.
Pour ce qui est des attributs concernant l'ombrage au solstice, on pourra chercher un lien de corrélation pour réduire à une seule variable le triplet de données `Hilshade_9am`, `Hilshade_Noon`, `Hilshade_3pm`.

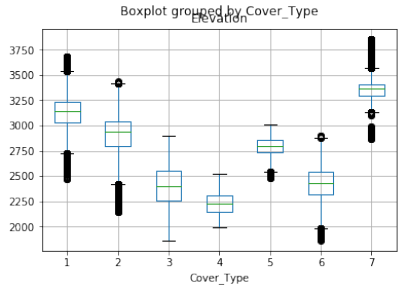
La première partie consistera donc à voir si on peut réduire le nombre de paramètres, la deuxième à modifier les données pour une meilleure analyse, et enfin mettre les données de train, de validation et de test. On trouve déjà quelques idées dans [BD99].



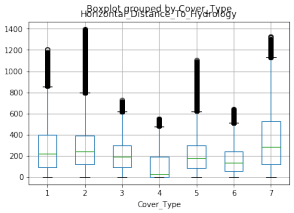
(a) Slope



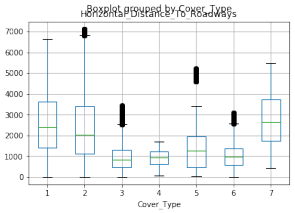
(b) Aspect



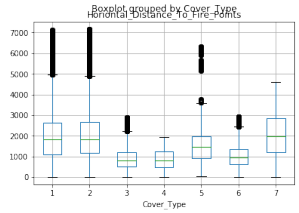
(c) Elevation



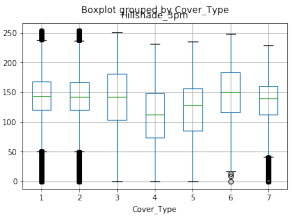
(d) Horizontal distance to hydrology



(e) Distance to roadways



(f) Distance to fire points



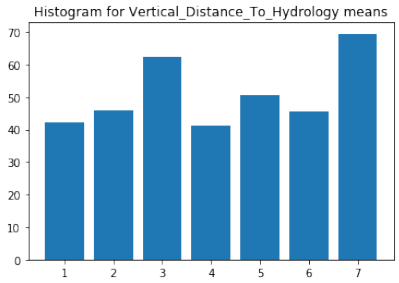
(g) Hillshade at 3pm

FIGURE 2 – Boxplot des données numériques

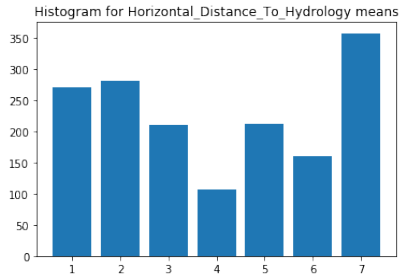
2.1 RÉDUCTION DES PARAMÈTRES

Tout d'abord, nous nous contenterons de la distance au point d'eau le plus proche, c'est à dire du nouvel attribut `Distance_To_Hydrology` définie par la distance euclidienne entre la forêt et le point d'eau, c'est-à-dire la quantité

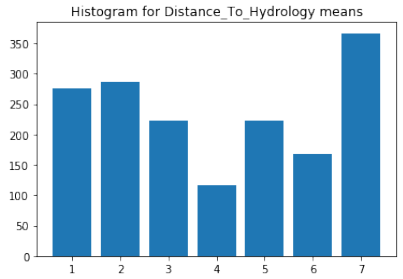
$$\sqrt{\texttt{Vertical\_Distance\_To\_Hydrology}^2 + \texttt{Horizontal\_Distance\_To\_Hydrology}^2}$$



(a) Vertical distance to hydrology



(b) Horizontal distance to hydrology



(c) Distance to hydrology

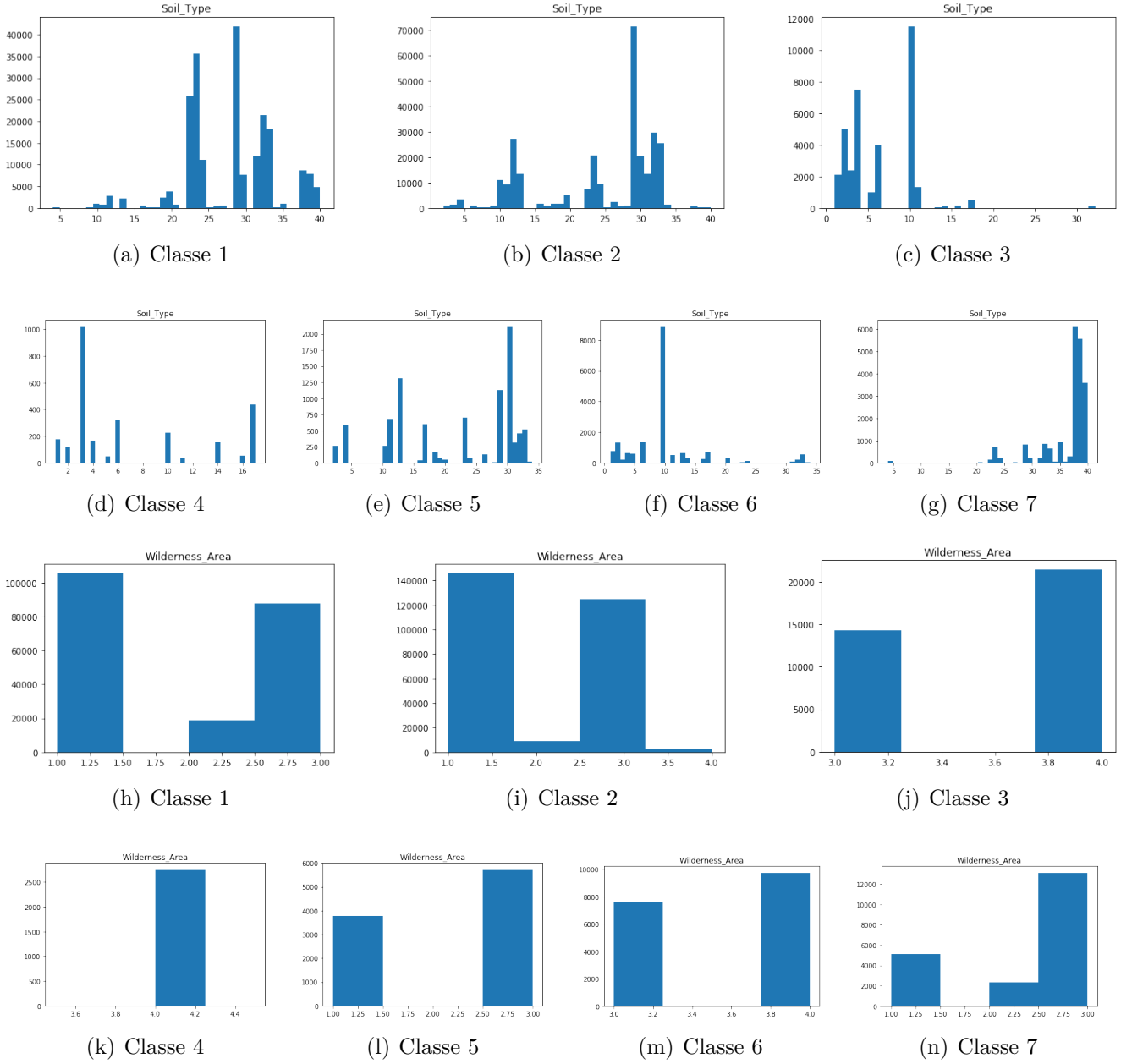
FIGURE 3 – Histogrammes des distances moyennes au point d'eau le plus proches par types de forêts

En effet, lorsqu'on regarde la figure 3, on constate que ce nouvel attribut reste similaire à `Horizontal_Distance_To_Hydrology`tout en étant sensible aux variations de `Vertical_Distance_To_Hydrology`.

Ensuite,

2.2 Données qualitatives

Premièrement, au vu du nombre de paramètres que comportent les attributs `Soil_Type`et `Wilderness_Area`, on peut se demander si ces données sont vraiment utiles au problème,



(a) Classe 1 (b) Classe 2 (c) Classe 3

(d) Classe 4 (e) Classe 5 (f) Classe 6 (g) Classe 7

(h) Classe 1 (i) Classe 2 (j) Classe 3

(k) Classe 4 (l) Classe 5 (m) Classe 6 (n) Classe 7

Figure 4 – Distributions de `Soil_Type`(a)–(g) et `Wilderness_Area`(h)–(n) par classes de forêts

et effectivement les données qualitatives sont importantes : on voit sur la figure 4 qu'elles varient beaucoup selon les classes, et dans [CD14] qu'elles changent considérablement le score des modèles *K-Means* et *SVM*.

3 Test des méthodes

3.1 k-plus proches voisins

L'apprentissage prend un temps fou...

3.2 Random Forest

On fait varier les paramètres de notre premier modèle. On commence par les paramètres `n_estimators`, qui est le nombre d'arbres, et `max_depth` qui la profondeur des arbres utilisés dans le modèle.
On constate par différents test [2] qu'aller au delà de 15 et 50 respectivement ne change pas grand chose.

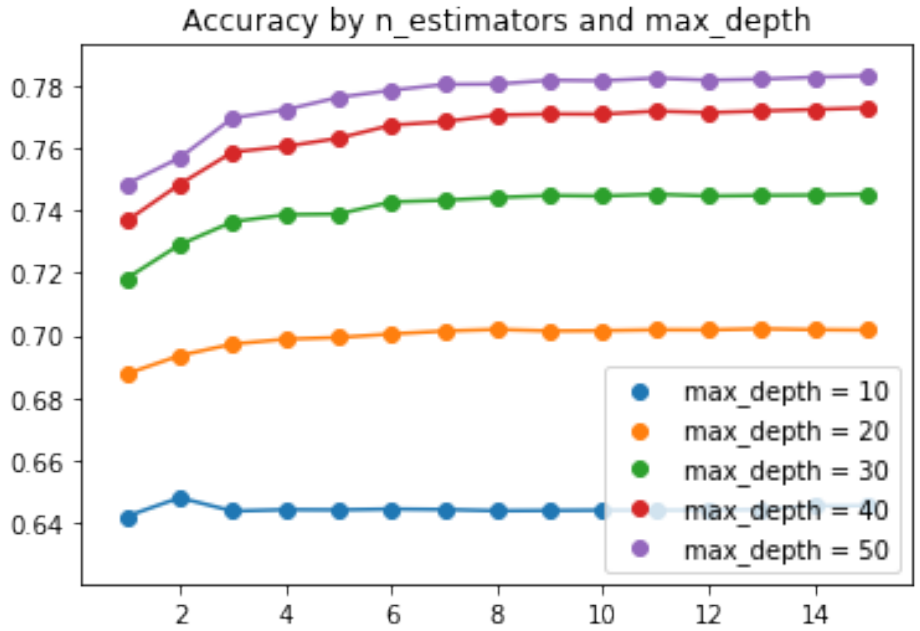


Figure 5 – Accuracy en fonction des paramètrse `n_variables` et `max_depth`

Références

[BD99] Blackard, Jock A. ; Dean, Denis J. : Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. In : *Computers and Electronics in Agriculture* (1999)

[CD14] Crain, Kevin ; David, Graham : Classifying Forest Cover Type using Cartographic Features. (2014)

2. cf. le notebook joint au rapport