

Projet Noté

Algorithmes Avancés

José Ferro Pinto Fabrice Ceresa

HEPIA

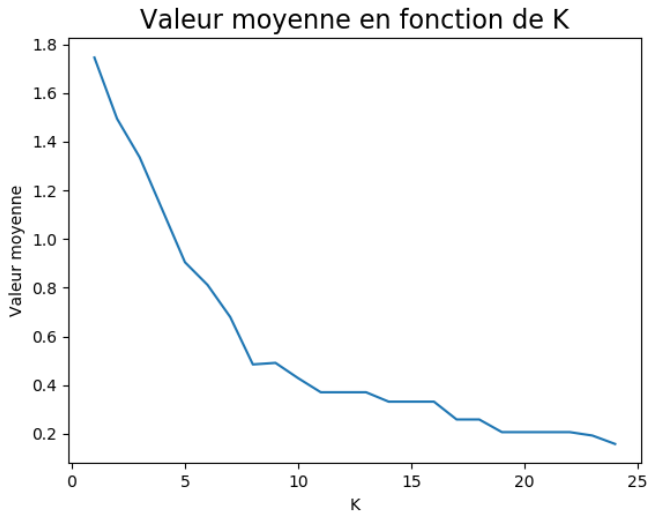
8 juin 2018

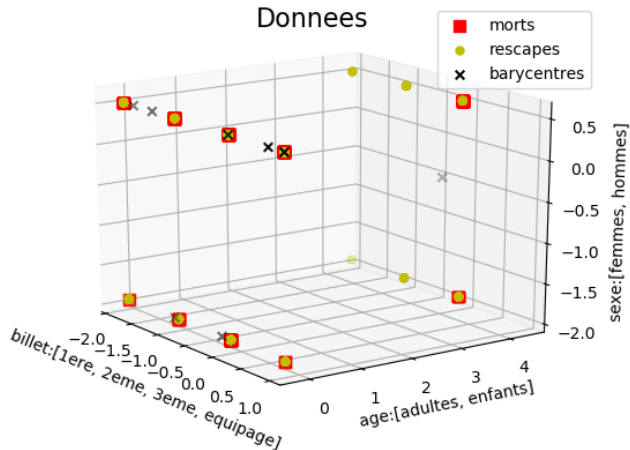


Titanic :

- Nombre de données : 2201
- Nombre de classes : 2 (Survived or Died)
- Dimensionnalité : 3 (billet, âge, sexe)

- KMeans
- Paramètres fixes : `init='random'`, `random_state=0`
- Paramètre variable : `n_clusters` nombre de clusters [1,25] avec un pas de 1

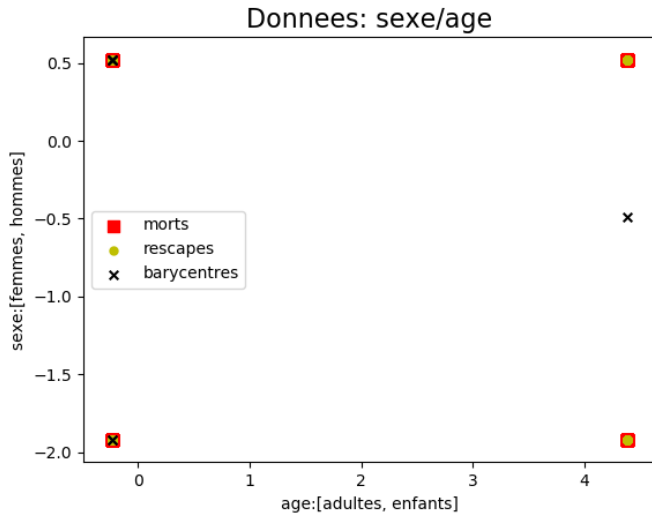


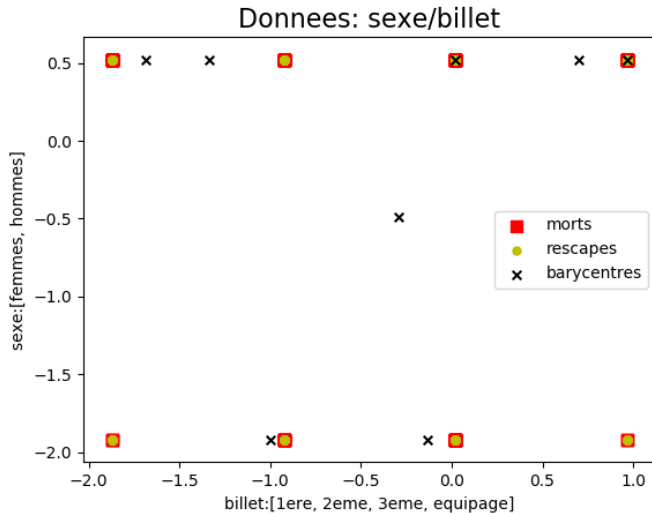


Partie 1

Résultats







Cancer du sein du Wisconsin :

- Nombre de données : 569
- Nombre de classes : 2 (Malignant or Benign)
- Dimensionnalité : 30

Vin :

- Nombre de données : 178
- Nombre de classes : 3
- Dimensionnalité : 13

Le programme se passe en 4 boucles intriquées :

- ➊ Pour chacun des datasets
- ➋ Pour chacun des classificateurs
- ➌ Pour chacun des paramètres
- ➍ Pour chacune des étapes de la validation croisée

Partie 2

Classificateur : K-plus proches voisins

- `KNeighborsClassifier`
- Paramètre variable : `n_neighbors` nombre de voisins pris en compte : $[1,51]$ avec un pas de 5

Partie 2

Classificateur : arbres de décisions

- DecisionTreeClassifier
- Paramètre variable : `min_samples_leaf` Nombre d'objets à partir duquel on comptabilise une feuille [2, 52] avec un pas de 5

Partie 2

Classificateur : Perceptron multi-couche avec une couche utilisant "stochastic gradient descent"

- MLPClassifier
- Paramètres fixes : `solver='sgd'`, `activation='logistic'`,
`max_iter=1000`, `verbose=False`,
`learning_rate_init=0.1`, `tol=0.`, `early_stopping=True`
- Paramètre variable : `hidden_layer_sizes=(nodes,)` donc une couche et nodes varie entre 2 et 20 par pas de 3.

Partie 2

Classificateur : Perceptron multi-couche avec une couche utilisant “a stochastic gradient-based optimizer”

- MLPClassifier
- Paramètres fixes : `solver='adam',`
`activation='logistic', max_iter=1000, verbose=False,`
`learning_rate_init=0.1, tol=0., early_stopping=True`
- Paramètre variable : `hidden_layer_sizes=(nodes,)` donc une couche et nodes varie entre 2 et 20 par pas de 3.

Partie 2

Classificateur : Perceptron multi-couche avec deux couches utilisant “a stochastic gradient-based optimizer”

- MLPClassifier
- Paramètres fixes : `solver='adam'`,
`activation='logistic'`, `max_iter=1000`, `verbose=False`,
`learning_rate_init=0.1`, `tol=0.`, `early_stopping=True`
- Paramètre variable : `hidden_layer_sizes=(nodes, 5)` donc deux couches et nodes varie entre 2 et 20 par pas de 3.

Partie 2

Validation : Validation croisée



Utilisation de

`RepeatedKfold(n_splits=5, n_repeats=10, random_state=None)`

Partie 2

Resultats : Cancer du sein

Breast Cancer data of size 569 :

KNeighborsClassifier

mean is 0.9156607528475251 with 0.005261918472447842 as standard deviation.

Worked for 58.37147793300028 seconds

DecisionTreeClassifier

mean is 0.918515193857532 with 0.007526398539306361 as standard deviation.

Worked for 1.9232283050005208 seconds

MLPClassifier one layer sgd_solver

mean is 0.727365574703721 with 0.012868801670619141 as standard deviation.

Worked for 29.68843971000024 seconds

MLPClassifier one layer adam_solver

mean is 0.8277927857993066 with 0.01944967972932608 as standard deviation.

Worked for 6.480671232000532 seconds

MLPClassifier two layers, second layer 5 nodes, sgd_solver

mean is 0.7151894115820524 with 0.017601386032980122 as standard deviation.

Worked for 204.33885359600026 seconds

MLPClassifier two layers, second layer 5 nodes, adam_solver

mean is 0.8806233504114266 with 0.007611351937622535 as standard deviation.

Worked for 19.824140925999927 seconds

Partie 2

Resultats : Vin

Wine data of size 178 :

KNeighborsClassifier

mean is 0.7270447330447332 with 0.05395925687595904 as standard deviation.

Worked for 58.24855370300065 seconds

DecisionTreeClassifier

mean is 0.8258658008658007 with 0.06566382345753549 as standard deviation.

Worked for 0.2911996069997258 seconds

MLPClassifier one layer SGD_solver

mean is 0.511431216931217 with 0.013653796940041013 as standard deviation.

Worked for 52.15133309700013 seconds

MLPClassifier one layer adam_solver

mean is 0.46363756613756607 with 0.04200778922986875 as standard deviation.

Worked for 4.492833305000204 seconds

MLPClassifier two layers, second layer 5 nodes, SGD_solver

mean is 0.37066137566137564 with 0.013903555873703817 as standard deviation.

Worked for 113.28683143299986 seconds

MLPClassifier two layers, second layer 5 nodes, adam_solver

mean is 0.49454232804232806 with 0.011857834905633243 as standard deviation.

Worked for 7.685992364000413 seconds

- Echantillonnage plus grand (569 vs 178) \Rightarrow meilleurs résultats
 - score moyen général : 0.830857845 vs 0.565530503
 - écart-type moyen général : 0.011719923 vs 0.033507676
- Moins échantillon \Rightarrow plus rapide
- Plus lent \nRightarrow meilleurs résultats

Letter Image Recognition Data¹

- Nombre de données : 20000
- Nombre de classes : 26 (1 par lettre)
- Dimensionnalité : 16

- Transformation de lettres vers numéro
- import des données avec `numpy.genfromtxt()`
- Reprise du travail de la partie 2

Partie 3

Resultats

```
KNeighborsClassifier
    mean is 0.9203122727272728 with 0.020939017447170083 as standard deviation.
    Worked for 360.20789452599956 seconds
DecisionTreeClassifier
    mean is 0.733379090909091 with 0.04240275735128243 as standard deviation.
    Worked for 174.38097832700032 seconds
MLPClassifier one layer sgd_solver
    mean is 0.5157916666666665 with 0.17533825530163752 as standard deviation.
    Worked for 700.9990525189996 seconds
MLPClassifier one layer adam_solver
    mean is 0.6683058333333333 with 0.16247980283245123 as standard deviation.
    Worked for 254.53706921899993 seconds
MLPClassifier two layers, second layer 5 nodes, sgd_solver
    mean is 0.039465 with 0.0011532309684823187 as standard deviation.
    Worked for 102.64895842199985 seconds
MLPClassifier two layers, second layer 5 nodes, adam_solver
    mean is 0.319565 with 0.07064751617478612 as standard deviation.
    Worked for 232.7630711769998 seconds
```

- Résultats beaucoup plus variables
- Perceptron travaille longtemps pour un résultat pas très probant (~ 0.66)
- Arbre de décision (~ 0.73)
- K-plus proches voisins (~ 0.92)