Université Assane Seck de Ziguinchor
UFR Sciences et Technologies
Département d'Informatique
Master Génie logiciel

Année universitaire : 2021-2022[1]

Travaux Pratiques 2
K plus proches voisins

Contents

[1]Responsable du cours : Gorgoumack SAMBE

1 Objectifs du TP

1.1 Objectif général

A la fin de cette activité l'apprenant devra être capable de sélectionner, d'appliquer et d'évaluer une classification par les plus proches voisins sur python.

1.2 Objectifs spécifiques

Pour objectifs spécifiques, vous devrez être capable à la fin de cette activité

1. de sélectionner les valeurs de k et la métrique (de distance) adéquate pour l'application d'une classification par les plus proches voisins.
2. d'appliquer une classification par les plus proches voisins.
3. d'évaluer une classification par les plus proches voisins.

2 Création, entraînement et évaluation d'un classifier

Reportez vous à la documentation de scikit learning pour l'implémentation du classifier des k plus proches voisins :
https://scikit-learn.org/stable/modules/neighbors.html.

2.1 Exercice : Apprentissage sur Iris

Écrire un programme qui ouvre le jeux de données iris, entraîne un classifieur des k plus proches voisins (avec k = 10) sur ce jeu de données et affiche le score sur l'échantillon d'apprentissage.

2.2 Exercice : évaluation du classifier sur Iris

Améliorer votre programme pour qu'il répartisse les données en un ensemble d'apprentissage (70%) et un ensemble test, entraîne un classifieur des 10 plus proches voisins sur l'échantillon d'apprentissage et évalue son erreur sur l'échantillon test. Vous afficherez le score et la matrice de confusion.

2.3 Choix de k

Nous avons choisi au hasard 10 pour valeur de k. Pour le choix d'un k optimal nous allons prendre le k qui donne un score maximal. Améliorer votre code pour qu'il propose pour valeur de k (entre 1 et 25),celle qui offre un score maximal.

2.4 Choix de k avec une validation croisée

Le jeu de données iris étant de petite taille, le choix de k par une validation simple est très optimiste, on pourrait avoir un meilleur résultat en faisant une validation croisée.

Sklearn permet de faire une validation croisée. Testez le code suivant :
from sklearn import datasets

from sklearn.neighbors import KNeighborsClassifier
X, y = datasets.load_iris(return_X_y=True)
clf=KNeighborsClassifier(n_neighbors=15)
from sklearn.model_selection import cross_val_score
scores = cross_val_score(clf, X, y, cv=5)
print(scores)

cross_val_score renvoie ici les scores d'une validation croisée à 5 fold pour une classification par les 15 plus proches voisins sur iris.

Améliorer ce code pour qu'il propose pour valeur de k, celle qui donne le score maximal (somme des 10 folds pour chaque k) pour une validation croisée à 10 folds pour chaque valeur de k (entre 1 et 25).

3 Application

3.1 Le jeu de données Digits

Digits comprend 1797 images de résolution 8x8, représentant un chiffre manuscrit. Le champ image est un tableau 8×8 d'entiers représentant des niveaux de gris et variant de 0 (blanc) à 16 (noir). Le champ data est un vecteur de dimension 64 comprenant la même information. Le champ target,représentant la cible, est un chiffre compris entre 0 et 9.

Il est possible de visualiser chaque image au moyen du code suivant :
import pylab as pl
pl.gray()
pl.matshow(digits.images[20])
pl.show()

3.2 Apprentissage et évaluation

1. Proposer un code qui répartit les données de digits en un ensemble d'apprentissage et un ensemble test (30% pour le test) puis choisit par validation croisée sur l'ensemble d'apprentissage un nombre de voisins k_opt optimal.

2. Entraîner le classifieur des k_opt plus proches voisins sur l'échantillon d'apprentissage puis évaluer son erreur sur l'échantillon test.

3. Afficher la matrice de confusion.

4. Afficher quelques chiffres parmi ceux qui sont mal classés.

5. Accéder à la documentation des métriques de sklearn[2]. Afficher les valeurs des métriques vues en cours si elles sont implémentées (exactitude, taux d'erreur, rappel, spécificité, precision)[3].

[2] https://scikit-learn.org/stable/modules/model_evaluation.html
[3] sklearn.metrics.classification_report permet également d'afficher quelques indicateurs.