CONTENU

RÉGION ACADÉMIQUE
PROVENCE-ALPES-
CÔTE D'AZUR

Liberté
Égalité
Fraternité

Cette activité est une ébauche, elle est en cours d'expérimentation et sera mise à jour et complétée

Le seul algorithme d'IA présent en NSI est l'algorithme des k-plus proches voisins. Malgré cette limitation, il est possible de sensibiliser aux enjeux propres à l'IA. Le but de cette activité est d'apporter des informations sur le prétraitement de données et le code présenté n'a pas vocation à être enseigné directement en NSI.

Code capytale de l'activité : d404-6776801

Déroulement de l'activité :

1. Présentation du problème et traitement des données :

On considère ici le problème suivant : étant donné des mesures médicales, peut-on déterminer si un patient est atteint du diabète ?
Le jeu de données est le Pima Indians Diabetes Database qui fournit 768 mesures associées à des femmes issues de la communauté nord-amérindienne des Pima dont la présence de diabète et d'obésité a été un sujet d'études.
Dans ce jeu de données, il y a 8 champs associés à des valeurs numériques et un champ Outcome valant 1 ou 0 selon que la personne soit atteinte de diabète ou non.
Pour traiter les données, on va

- Passer d'un tableau de tuples nommés à deux tableaux :
- carac composé de tableaux de valeurs numériques observées ;
- etiq composé des étiquettes de classification, ici 0 ou 1 selon le fait d'être atteint ou non du diabète ;
- Sélectionner une partie des données pour l'entrainement et une partie pour la vérification de l'efficacité, on parle de jeu d'entraînement et de jeu de tests.

2. Les classificateurs

Un classificateur (*classifier* en anglais) va partir d'un couple (carac, etiq) de données d'entrainement, s'initialiser pour s'aligner à ces données (on parle de *fit*) et permettre ensuite de prédire l'étiquette d'une valeur.
Certains classificateurs sont dits *sans modèle*, car ils se contentent d'utiliser le couple de données d'entrainement pour prédire. Pour avoir une ligne de comparaison, on va considérer deux classificateurs sans modèle *naïfs* :

- le classificateur Aléatoire qui renvoie une étiquette aléatoire parmi les étiquettes des données d'entrainement ;
- le classificateur Majoritaire qui renvoie l'étiquette la plus présente.

L'algorithme des **k-plus proches voisins, ou kNN,** est un classificateur sans modèle qui prédit l'étiquette en prenant l'étiquette majoritaire parmi les k données les plus proches. Il y a donc deux paramètres le nombre k, impair pour avoir une étiquette majoritaire dans le cas il y a deux classes, et la distance.
Pour estimer la précision d'un classificateur, on va prédire l'ensemble du jeu de tests et calculer le pourcentage de bonnes prédictions.

Les **réseaux de neurones** sont un classificateur avec modèle. `scikit-learn` propose un tel classificateur `MLPClassifier`, pour *multi-layer perceptron classifier`. Ici, la phase de `fit` va réaliser l'entrainement du réseau. Pour le faible nombre de données, on effectue un nombre conséquent d'itérations (epoch) pour garantir la convergence du modèle.

3. Nettoyage des données :

Si on affiche les données, on se rend compte que la notion de distance n'est pas forcément pertinente, car les échelles des distances sont différentes. Cela installe un biais dans l'expérimentation.
Une manière simple de résoudre ce problème est de redimensionner chaque donnée entre 0 et 1. Cela ne change pas la *forme* de la distribution, notamment cela ne l'étale pas, mais cela lui permet d'être comparée avec les autres.