TP2 : KNN(Régression)

Élément de module :	Machine learning	Année universitaire :	2019-2020
Semestre :	4		

L'algorithme des k-plus proches voisins est aussi utilisé pour des problèmes de régression. Dans cette seconde partie, nous allons étudier la base de données auto-mpg disponible ici(https://www-lisic.univ-littoral.fr/~teytaud/files/Cours/Apprentissage/data/auto-mpg.data). Il s'agit de prédire la consommation (miles per gallon- mpg) en fonction du nombre de cylindres, du déplacement, de la puissance, du poids, de l'accélération, de l'année et de l'origine. La dernière colonne correspond au nom de la voiture, il est unique et ne sera donc pas utile pour notre apprentissage, il nous faut donc ignorer cette colonne.

Comme pour la première partie, récupérez la base, analysez les données, et effectuez un apprentissage avec knn.

Là encore étudiez l'influence du paramètre k et la distance(manhattan, euclidean, minkowski, etc). Vous devez utiliser les bibliothèques suivantes :

- **from sklearn.neighbors import KNeighborsRegressor**
- **from sklearn.model_selection import cross_val_score**
- **from sklearn.model_selection import GridSearchCV**

Afin d'évaluer les performances de nos modèles, nous allons introduire deux nouvelles mesures pour les problèmes de régression : Mean Absolute Error

(MAE) et Mean Squared Error (MSE).

MAE est la moyenne des valeurs absolues des erreurs de prédiction, elle est donnée par la formule suivante :

$$MAE = \frac{1}{N}\sum_{i=0}^{N-1}|y_i - \hat{y}_i|$$

Avec $\hat{y}$ la valeur prédite pour le ième exemple, et yi la vraie valeur pour le ième exemple.

MSE est une mesure d'erreurs classiques, on la retrouve très souvent. Elle correspond à la moyenne des carrés des erreurs de prédiction. Elle est définie par la formule suivante :

$$MSE = \frac{1}{N}\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2$$

Quelle que soit l'erreur utilisée, il est important que dans le cadre d'une régression la mesure de performance ne tienne pas compte de la direction de l'erreur, car dans le cas contraire, une erreur "sur-prédite" et une "sous-prédite" s'annuleraient.

MAE et MSE tiennent comptent de ça en prenant la valeur absolue et le carré des différences. MSE donnent plus de poids aux erreurs sur les points aberrants (grosses erreurs). Pour résoudre ce problème, il est important de normaliser les données. Utilisez **from sklearn.preprocessing** import **StandardScaler** et en particulier la fonction **fit_transform** qui permet de normaliser les données.