Les DataFrames et l'apprentissage

Evaluation des targets

Exercice 1 : Émission de gaz à effet de serre

- *Lire les données du fichier* Gaz.csv. *Comme la dernière fois vous indiquerez que le type de la feature* INSEE *et une* 'string' *et que cette* feature *représente l'index de votre* DataFrame.
- *Réaliser une copie de la base de données, en ne conserver que les features de type float64. Vous supprimerez ensuite les données de localisation (longitude et latitude).*
- *Dans la suite* Data *sera la base d'origine et* co2 *la copie ne contenant que les émissions de co2.*
- *Réalisez une analyse statistique et interprétez les résultats.*
- *Effectuer une découpe en 30 bacs de la feature* 'Déchets', *vous enregistrerez les résultats en tant que nouvelle feature :* 'bacs' *dans le* DataFrame Data.
- *Affichez les résultats triés en ordre croissant. Que constatez-vous.*
- *Utilisez maintenant la fonction* histplot *de* seaborn *pour afficher le nombre d'individus la feature* 'Déchets' *dans une découpe en 30 bacs.*
- *Enregistrez maintenant dans la feature* 'bacs' *les découpes de la* feature 'Autres transports' *et affichez via la fonction* scatterplot *de* seaborn *toutes les communes en fonction de leur longitude et leur latitude, en utilisant la feature* 'bacs' *pour le paramètre* hue. *Vous limiterez la taille des points à 1 grâce à l'option* 's=1' *dans* scatterplot.
- *Affichez maintenant les corrélations existantes entre les différentes émissions de co2. Vous utilisez pour cela la fonction* heatmap *de* seaborn *en lui fournissant les corrélations. L'option* annot *de* heatmap *permet d'ajouter les valeurs dans l'affichage de la map. Si votre affichage n'est pas assez grand vous pourrez préciser la taille de l'affichage via la fonction* figure *de* matplotlib.pyplot *en lui donnant la taille de la figure avec l'option* figsize=(taille en x, taille en y).

Exercice 2 : Apprentissage

- *Afin de pouvoir effectuer un apprentissage sur les données de co2, remplacez toutes les valeurs manquantes par la valeur 0, dans le DataFrame co2.*
- *On souhaite estimer la target 'CO2 biomasse hors-total' à partir des autres émissions de co2. Utilisez un modèle de régression linéaire pour calculer les prédictions et évaluer le score.*
- *Afficher les résultats via la fonction scatterplot de seaborn entre les prédictions et les valeurs réelle de la target.*
- *Effectuer le même travail en considérant cette fois la feature 'Résidentiel' comme target. Le modèle de régression linéaire estime-t-il mieux cette target. Comparer les deux affichages scatterplot.*
- *On souhaite maintenant réduire l'influence des outliers. Pour cela on appliquera la fonction log au DataFrame co2 afin de réduire les distances entre les valeurs. Mais avant cela, afin que le log ne retourne pas des valeurs négatives très grandes lorsque les émissions de co2 sont proches 0, vous devez modifier via la fonction apply toutes valeurs inférieures à 1.5.*
- *Pour cela vous utiliserez une boucle sur toutes les features et vous appliquerez à chacune de ces features la fonction apply avec une lambda fonction qui conserve les valeurs supérieures à 1.5 et remplace les autres par cette valeur.*
- *Appliquez la fonction log de numpy à la base co2, afin d'avoir des données plus recentrées et comparables entre elles.*
- *Réaffichez via histplot la feature 'Déchets', avec une découpe de 30 bacs, et comparez les résultats obtenus précédemment avec la même découpe.*
- *Effectuer un nouvel apprentissage avec un modèle de régression linéaire sur la target 'Résidentiel'. Afficher la comparaison entre les valeurs prédites et celles attendues. Commentez les résultats obtenus.*
- *Pourquoi les résultats obtenus sont-ils meilleurs.*