Travaux pratique N° 1

Régression linéaire multiple et les réseaux de neurones artificiels

1. Contexte

Une compagnie d'assurance souhaite développer un outil permettant d'estimer les frais (charges) à payer par un client en tenant compte de certaines variables. Pour cela, la compagnie a collecté les données de plus de 1330 anciens clients. Les données collectées sont :

- age: âge du client
- sex (sexe): sexe du client (2 pour femme, 1 pour homme)
- bmi (Body mass index): indice de masse corporelle, fournissant une compréhension du corps, des poids relativement élevés ou faibles par rapport à la taille,
- children (enfants) : nombre d'enfants couverts à charge
- smoker (fumeur): fumeur (1 pour oui, 2 pour non)
- region (région): la zone résidentielle du bénéficiaire aux États-Unis, nord-est (4), sud-est(2), sud-ouest(1), nord-ouest (3).
- charges: frais médicaux individuels facturés par l'assurance maladie

Le fichier csv contenant les données peut être téléchargé à partir de ce lien :
http://sabeur.elkosantini.me/courses/ML/labs/insurance.csv

2. Objectif

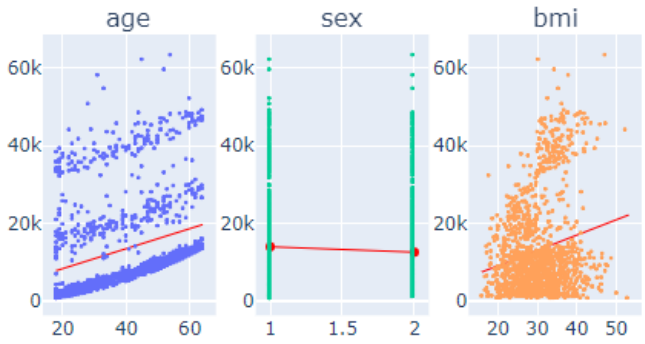
L'objectif de ce type est d'identifier le modèle ML le plus approprié pour développer l'outil souhaité. Dans ce TP, on souhaite explorer les modèles de régression linéaire et les réseaux de neurones artificiels.

3. Travail demandé

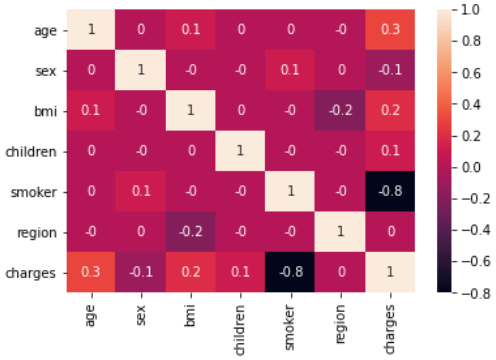
- Charger le fichier csv dans votre drive

3.1. Régression linéaire multiple

- Créer un fichier colab (appelé RLM-votrenom-insurance)
- En utilisant ce fichier, développer le code python qui permet de :
  - Affiche le nom des attributs (features ou clés)
  - Affiche les 5 premiers lignes du fichier
  - Vérifier s'il y a des attributs sans valeurs (ayant la valeur null)
  - Tracer les courbes (voir l'exemple ci-dessous) montrant l'évolution de chaque attribut par rapport à l'attribut target (charges).



- Afficher le heatmap (voir la figure ci-dessous)



- Commenter les courbes et le heatmap obtenues
- Créer :
  - Le vecteur Y qui contient la colonne « charges »
  - La matrice qui contient les colonnes 1-6.
- Diviser X et Y (splitting) en 70% pour l'apprentissage et 30% pour le test (créer x_train, y_train, x_test et y_test)
- Afficher la taille des 4 vecteurs.
- Créer le modèle de régression linéaire et lancer l'apprentissage
- Afficher les coefficients du modèle
- Calculer R^2 du modèle appliqué aux données d'apprentissage et aux données de test
- En utilisant le modèle construit, lancer la prédiction de toutes les données et afficher un tableau à deux colonnes montrant la charge (ou les frais) estimée et la charge réelle (voir l'exemple ci-dessous).

	charges	charge_REG
0	16884.92400	24798.335792
1	1725.55230	3778.852977
2	4449.46200	7049.447254
3	21984.47061	3784.094149
4	3866.85520	5559.238949

- Effectuer l'analyse des résidus et commenter le diagramme obtenu.

- Conclure quant au choix du modèle de régression linéaire

3.2. Réseau de neurones artificiel

- Créer un fichier colab (appelé MLP-votrenom-insurance)
- En utilisant ce fichier, développer le code python qui permet de :
  - Affiche le nom des attributs (features ou clés)
  - Affiche les 5 premiers lignes du fichier
  - Vérifier s'il y a des attributs sans valeurs (ayant la valeur null)
  - Créer un réseau de neurones caractérisé par :
    - Une couche cachée à 10 neurones fortement connecté (dense) et la fonction d'activation est ReLU.
    - Une couche de sortie avec une seule neurone avec la fonction d'activation linéaire (linear).
  - Lancer l'apprentissage pour 50 epoches et tracer la courbe d'erreur
  - Lancer l'apprentissage pour 200 epoches et tracer la courbe d'erreur
  - Comparer et commenter les deux résultats
  - Afin d'améliorer le résultat, changer le RNA par :
    - Une première couche cachée à 256 neurones fortement connecté (dense) et la fonction d'activation est ReLU.
    - Une deuxième couche cachée à 256 neurones fortement connecté (dense) et la fonction d'activation est ReLU.
    - Une troisème couche cachée à 256 neurones fortement connecté (dense) et la fonction d'activation est ReLU.
    - Une couche de sortie avec une seule neurone avec la fonction d'activation linéaire.
  - Lancer l'apprentissage pour 50 epoches et tracer la courbe d'erreur
  - Lancer l'apprentissage pour 200 epoches et tracer la courbe d'erreur
  - Comparer et commenter les deux résultats
  - Afin d'améliorer la performance du réseau de neurones, faut-il augmenter le nombre d'epoches en justifiant la réponse ?
  - Afficher le « MSE » (mean square error) pour le dernier réseau.
  - Essayer de changer les hyperparamètres pour améliorer la qualité du réseau.
  - En utilisant le RNA construit, lancer la prédiction de toutes les données et afficher un tableau à deux colonnes montrant la charge (ou les frais) estimée et la charge réelle (voir l'exemple ci-dessous).

	charges	charge_REG
0	16884.92400	24798.335792
1	1725.55230	3778.852977
2	4449.46200	7049.447254
3	21984.47061	3784.094149
4	3866.85520	5559.238949

3.3. Comparaison de modèles

- Comparer les deux modèles (régression linaire et le réseau de neurones)

4. Livrables

Les livrables demandés sont :

- Un rapport synthétisant le travail réalisé (sans code)
- Le fichier colab (extension : ipynb)

Ces documents doivent être envoyés par mail à : sabeur.elkosantini@yahoo.fr