Dans la première et deuxième partie de cet atelier, on va utiliser la base de données **Advertising**.Alors, l'objectif principal est de créer un modèle de régression pour prédire la variable Sales. La troisième partie traite le problème de classification en appliquant la régression logistique sur la base de données **diabetescsv.csv**. Pour cela, on commence à savoir comment

- Récupérer des données à partir d'un fichier csv et découvrir ses principales caractéristiques.
- Visualiser les données sous forme de graphe.

Partie 1 : Régression linéaire simple

1. Récupérer des données à partir de fichier advertising.scv et découvrir ses principales caractéristiques.
2. Initialiser X (predictors, variable indépendante) par la variable TV et y (cible, variable dépendante).
3. Pourquoi il s'agit d'un problème de régression
4. Visualiser la base de données avec matplotlib.
5. Soit le modèle de régression linéaire simple définit par

$$y = w_0 + w_1 x$$

   (a) à l'aide de train_test_split de sklearn.model_selection, diviser la base de données en base d'apprentissage et base de test (80% pour la base d'apprentissage et 20% pour la base de test)

   (b) Définir une fonction MSE(X,y,W) qui retourne la moyenne des erreurs entre la valeur théorique et la valeur réelle.

   (c) Définir la fonction Gradient(X,y,W).

   (d) Etablir une fonction de mise à jour des paramètres W MiseJour(grad,W,alpha).

   (e) Ecrire la fonction Batch_Gradient_Descent(X,y,W,N_max,alpha,eps), qui retourne les meilleurs paramètres estimés et de plus affiche l'évolution de MSE après chaque itération.

   (f) Tracer l'MSE.

   (g) Donner l'erreur de ce modèle.

Partie 2 : Régression linéaire multiple

Soit le modèle de régression linéaire multiple définit par

$$y = w_0 + \sum_{j=1}^{m} w_j x_j$$

1. Adapter les étapes de la partie précédente pour estimer les paramètres de modèle de régression multiple.
2. Donner l'erreur de ce modèle.

Problème 2 : Régression logistique

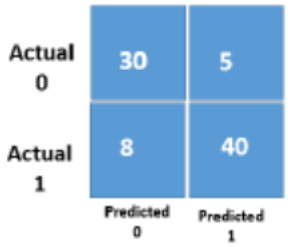
Dans ce problème, on va utiliser le Dataset https ://www.kaggle.com/saurabh00007/diabetescsv

- Ouvrir un data set à l'aide de pandas et récupérer un dataframe
- Connaitre les dimensions du dataframe
- Explorer la liste des colonnes
- Récupérer une colonne, un ensemble de colonnes
- Le max, le min, la moyenne d'une colonne

Pour faire la classification, on se base sur la variable indépendante Glucose et la variable dépendante outcome de diabetescsv.csv

1. Quelles sont les principales étapes à suivre pour établir un modèle de Single Variate Logistic Regression basé sur sklearn ?

2. Data
   (a) Définir le dataset (X,y) et les afficher
   (b) Tracer les données à l'aide de matplotlib
   (c) Conclusions ?
   (d) à l'aide de train_test_split de sklearn.model_selection, diviser le dataset en training dataset et test dataset (80% pour training_data et 20% pour test_data)
   (e) Pourquoi il faut séparer les données en données d'apprentissage et en données de test ?
   (f) Afficher le nombre d'enregistrement pour le training et le nombre d'enregistrement pour le test
3. Modèle
   (a) Créer le modèle
   (b) Faire l'apprentissage
   (c) Afficher les coefficients
   (d) Tracer la fonction Logit (matplotlib)
   (e) Tracer la fonction Sigmoid (matplotlib)
   (f) En utilisant les coefficients trouvés par l'algorithme :
      i. Écrire l'expression mathématique pour faire la prédiction pour une personne ayant le glucose 197
      ii. Calculer la sortie de cette valeur (à l'aide de python) et comment l'interpréter
      iii. À quelle classe appartient cette personne ?
   (g) En utilisant les coefficients trouvés par l'algorithme :
      i. Écrire l'expression mathématique pour faire la prédiction pour un ensemble de personnes ayant le glucose respectivement [110,139,100,84,44]
      ii. Calculer la sortie pour l'ensemble des valeurs (à l'aide de python) et interpréter le résultat obtenu
      iii. À quelle classe appartient chaque personne ?
4. Évaluation
   (a) Soit la matrice de confusion suivante :



   Quelles sont les valeurs de FP, TP, FN, TN

   (b) Soit les deux vecteurs suivants :

y_real	y_predicted
0	0
0	0
0	0
0	1
1	1
1	0
1	1
1	1
1	0
1	1

      i. Trouver la matrice de confusion équivalente aux vecteurs ci-dessus.

ii. Écrire une fonction en python qui retourne une matrice de confusion. Tester cette fonction en utilisant les deux vecteurs de la question précédente.

(c) Matrice de confusion du modèle

i. À l'aide de sklearn.metrics, afficher la matrice de confusion du modèle trouvé en 3

ii. Retourner la même matrice en utilisant votre fonction

(d) Que signifie accuracy metric ?

La formule de accuracy : $\boldsymbol{accuracy} = \dfrac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN}}$ pour une classification binaire

i. Pour la matrice de confusion en 4.c, calculer accuracy du modèle trouvé en 3

ii. À l'aide de sklearn.metrics.accuracy_score, calculer accuracy du modèle trouvé en 3