Activité 1 : Lire la dataset + préparation des données

Dans cette activité, on se basera sur la base "pima-indians-diabetes.csv".
Pima est un groupe d'indien de l'Amérique (Native Americans) vivant à Arizona. Ils sont sujets de plusieurs études car le changement dans leurs habitude alimentaire a favorisé qu'ils soient atteints de diabète de type 2.
La base inclus 767 femmes décrites par les 8 attributs suivants :
1. Nombre de grossesses (NumTimesPrg)
2. Concentration du plasma en glucose (PlGlcConc)
3. Tension artérielle (BloodP)
4. Épaisseur du pli cutané des triceps (SkinThick)
5. Taux d'insuline (TwoHourSerIns)
6. Indice de masse corporelle (BMI)
7. Fonction pédigrée du diabète : hérédité (DiPedFunc)
8. Age (age)

La dernière colonne "HasDiabetes" de la base (9ème colonne) indique si la personne est diagnostiqué (1) de diabète ou pas (0).

On souhaite d'abord lire et inspecter les données dans la base. La base "pima indians-diabetes.data.csv" est fournie avec le format CSV (Comma Separated Values, valeurs séparées par des virgules). Il s'agit du format le plus commun dans l'import et l'export de feuilles de calculs et de bases de données.

Exercice 1

1. Lire le contenu du fichier "**pima-indians-diabetes.csv**" à l'aide de la fonction "**read_csv**" du paquetage "pandas".

2. Attribuer à chaque colonne le nom de l'attribut correspondant.

3. Afficher la structure de la base à l'aide de l'attribut "**shape**" appliqué sur un objet de type "**dataframe**".

4. Afficher les 10 premières lignes de la base à l'aide de la fonction" **head(nbre)**" appliquée sur un objet de type "**dataframe**".

5. Afficher seulement les valeurs de la tension artérielle (attribut "BloodP") pour la totalité des individus.

6. Obtenir des statistiques sur les individus dans la base à l'aide de la fonction" **describe()**" appliquée sur un objet de type "**dataframe**".

7. Créer une matrice qui ne contient que les données des attributs (séparer les valeurs des attributs de leurs classes d'appartenances).

II- Préparation des données

a. Data Cleaning

Certaines instances peuvent avoir des attributs ayant des valeurs nulles. Ce qui peut dégrader les performances d'un algorithme d'apprentissage. La solution la plus simple est de supprimer les individus dont les valeurs de certains attributs sont manquantes. Mais, ceci peut causer la perte de données importantes. Un seconde alternative et de déterminer, la valeur médiane de chaque attribut, puis de remplacer les valeurs nulle par la médiane.

Exercice 2

1. Filtrer les valeurs de l'attribut '**SkinThick'** pour déterminer les valeurs non nulles.
2. Calculer la médiane de ces valeurs en utilisant la fonction" **median ()**" appliquée sur un objet de type "**dataframe**".
3. Remplacer les valeurs nulles de l'attribut "**SkinThick"** par la médiane**. 4.**

Pourquoi on ne peut pas faire la même chose avec l'attribut **" NumTimesPrg"** ?

b. Data Rescaling

Le changement d'échelle est une sorte de normalisation. L'intervalle dans lequel vari les variables numériques peut être différent selon l'attribut. Ceci peut influencer les performances de certains algorithmes d'apprentissage automatique, surtout ceux qui se basent sur le calcul de distances. Pour cela, le but de cette étape est de ramener toutes les valeurs dans l'intervalle [0,1]. Une des techniques utilisées pour la normalisation est la suivante :

Exercice 3

1. Utiliser les fonctions "**MinMaxScaler**" du pacquage "**sklearn.preprocessing**" pour créer un normalisateur approprié "**scaler**".
2. Utiliser la fonction "**scaler.fit_transform(…)**" pour ramener les valeurs des attributs dans un intervalle [0-1].
3. Réafficher les données.

c. Data standardization

La standardisation des données est aussi une sorte de normalisation. Ramener les valeurs des attributs à l'intervalle [0-1] est parfois insuffisant surtouts dans le cas des bases qui contiennent beaucoup de zéros ou des algorithmes qui multiplient ces valeurs par une certaine pondération. Les valeurs d'un attribut sont transformées pour suivre une loi gaussienne ayant une moyenne $\mu=0$ et un écart type $\sigma=1$.

Exercice 4

1. Utiliser la fonction **"StandardScaler().fit(…))"** du pacquage "**sklearn.preprocessing**" pour créer un normalisateur approprié "**scaler**".
2. Utiliser la fonction "**scaler.fit_transform(…)**" pour standardiser les données.
3. Réafficher les données.