Les Pima sont un groupe d'Amérindiens vivant en Arizona. Une prédisposition génétique a permis à ce groupe de survivre normalement à un régime pauvre en glucides pendant des années. Au cours des dernières années, en raison d'un passage soudain des cultures agricoles traditionnelles aux aliments transformés, ainsi que d'un déclin de l'activité physique, ils ont développé la prévalence la plus élevée de diabète de type 2 et pour cette raison ils ont fait l'objet de nombreuses études.

L'ensemble de données comprend des données provenant de 768 femmes présentant 8 caractéristiques, en particulier :

- Nombre de grossesses
- Concentration de glucose plasmatique a 2 heures dans un test de tolérance au glucose par voie orale
- Tension artérielle diastolique (mm Hg)
- Epaisseur du pli cutané du triceps (mm)
- 2 heures d'insuline sérique (mu U/ml)
- Indice de masse corporelle (poids en kg/(taille en m)^2)
- Le diabète pedigree fonction
- Âge (années) La dernière colonne de l'ensemble de données indique si la personne a reçu un diagnostic de diabète (1) ou non (0).

L'objectif est de déterminer quelles sont les caractéristiques (features) pour identifier les personnes qui ont un diabète de type 2

```
1 Récupérer le fichier pima-indians-diabetes.csv et le mettre dans un
  dataframe. Attention les premières lignes correspondent à la description des
  données. Il est possible de ne pas les lire en mettant skiprows=9 dans la
  fonction read_csv.
```

In [ ]:

```
1
```

Afficher le nombre de ligne et de colonnes du dataframe ainsi que les 5 premières lignes

In [ ]:

```
1
```

Afficher la matrice de corrélation. Rappel il faut utiliser la fonction corr().

In [ ]:

```
1
```

Afficher, à l'aide de seaborn, la matrice de correlation

In [ ]:

```
1
```

Il est important d'analyser les histogrammes de chaque variable pour mieux comprendre comment les données sont réparties.
A l'aide du code suivant, afficher les différents histogrammes.
import matplotlib.pyplot as plt
df.hist(bins=50, figsize=(20, 15))
plt.show()

In [ ]:

```
1
```

In [ ]:

```
1
```

Existe-t'il des valeurs nulles ? Existe-til des valeurs manquantes ? Rappel vous pouvez le voir avec des histogrammes mais aussi avec une heatmap.

In [ ]:

```
1
```

En fait on peut constater qu'il n'y a pas de valeurs manquantes avec le heatmap mais par contre il y a des valeurs nulles. Il faut toujours faire attention à la manière dont sont codées les valeurs manquantes. Ici nous voyons dans les histogrammes que pour BMI, BloodP, PlGlcConc, SkinThick, TwoHourSerIns il existe des valeurs manquantes. Le nombre de grossesses n'est pas considéré comme une valeur manquante bien sûr.

Transformer les valeurs nulles par la médiane de la série.

In [ ]:

```
1
```

Les données sont, à présent, transformées et nous allons pouvoir créer un jeu de données de test et d'apprentissage. Faire une copie du dataframe en df2. Sur df appliquer un scaling pour normaliser les valeurs par rapport à la moyenne et l'écart type (utilisation de StandardScaler (). Nous conservons la copie df2 sans transformation.
L'objectif à présent est d'appliquer différents classifieurs pour voir celui qui est le plus performant. Pour le ou les meilleurs rechercher les hyperparamètres et créer un pipeline à sauvegarder. Il faut ensuite pouvoir traiter de nouvelles données pour prédire si il y a diabète ou pas.

Tester les résultats sur df et sur df2.

In [ ]:

```
1
```