seance1

April 8, 2023

Mariem Ben Hmida 2GT1

[1]:
```
import pandas as pd #manipulation des tableaux des données
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns # pour tracer le boxplot
import warnings # ignorer les alertes
warnings.filterwarnings
```

[1]:
```
<function warnings.filterwarnings(action, message='', category=<class
'Warning'>, module='', lineno=0, append=False)>
```

[2]:
```
df = pd.read_csv("stroke_data.csv", sep=";")
```

[3]:
```
df
```

[3]:
```
         id  gender   age  hypertension  heart_disease ever_married  \
0      9046    Male  67.0             0              1          Yes
1     51676  Female  61.0             0              0          Yes
2     31112    Male  80.0             0              1          Yes
3     60182  Female  49.0             0              0          Yes
4      1665  Female  79.0             1              0          Yes
…     …     …   …           …            …          …
5105  18234  Female  80.0             1              0          Yes
5106  44873  Female  81.0             0              0          Yes
5107  19723  Female  35.0             0              0          Yes
5108  37544    Male  51.0             0              0          Yes
5109  44679  Female  44.0             0              0          Yes

          work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
0           Private          Urban             228.69  36.6  formerly smoked
1     Self-employed          Rural             202.21   NaN     never smoked
2           Private          Rural             105.92  32.5     never smoked
3           Private          Urban             171.23  34.4           smokes
4     Self-employed          Rural             174.12  24.0     never smoked
…             …            …                …   …              …
5105        Private          Urban              83.75   NaN     never smoked
5106  Self-employed          Urban             125.20  40.0     never smoked
```

```
5107  Self-employed          Rural              82.99  30.6     never smoked
5108        Private          Rural             166.29  25.6  formerly smoked
5109       Govt_job          Urban              85.28  26.2          Unknown

      stroke
0          1
1          1
2          1
3          1
4          1
…      …
5105       0
5106       0
5107       0
5108       0
5109       0

[5110 rows x 12 columns]
```

[4]:
```
df.shape
```

[4]:
```
(5110, 12)
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

[6]:
```
df.head()#affiche les 5 premieres lignes
```

[6]:

```
      id  gender   age  hypertension  heart_disease ever_married  \
0   9046    Male  67.0             0              1          Yes
1  51676  Female  61.0             0              0          Yes
2  31112    Male  80.0             0              1          Yes
3  60182  Female  49.0             0              0          Yes
4   1665  Female  79.0             1              0          Yes

       work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
0        Private          Urban             228.69  36.6  formerly smoked
1  Self-employed          Rural             202.21   NaN     never smoked
2        Private          Rural             105.92  32.5     never smoked
3        Private          Urban             171.23  34.4           smokes
4  Self-employed          Rural             174.12  24.0     never smoked

   stroke
0       1
1       1
2       1
3       1
4       1
```

[7]:

```python
df.head(7) #affiche les 7 premieres lignes
```

[7]:

```
      id  gender   age  hypertension  heart_disease ever_married  \
0   9046    Male  67.0             0              1          Yes
1  51676  Female  61.0             0              0          Yes
2  31112    Male  80.0             0              1          Yes
3  60182  Female  49.0             0              0          Yes
4   1665  Female  79.0             1              0          Yes
5  56669    Male  81.0             0              0          Yes
6  53882    Male  74.0             1              1          Yes

       work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
0        Private          Urban             228.69  36.6  formerly smoked
1  Self-employed          Rural             202.21   NaN     never smoked
2        Private          Rural             105.92  32.5     never smoked
3        Private          Urban             171.23  34.4           smokes
4  Self-employed          Rural             174.12  24.0     never smoked
5        Private          Urban             186.21  29.0  formerly smoked
6        Private          Rural              70.09  27.4     never smoked

   stroke
0       1
1       1
2       1
3       1
4       1
```

```
5       1
6       1
```

[8]:
```
df.describe() # les statistiques numeriques
```

[8]:
```
                 id          age  hypertension  heart_disease  \
count   5110.000000  5110.000000   5110.000000    5110.000000
mean   36517.829354    43.226614      0.097456       0.054012
std    21161.721625    22.612647      0.296607       0.226063
min       67.000000     0.080000      0.000000       0.000000
25%    17741.250000    25.000000      0.000000       0.000000
50%    36932.000000    45.000000      0.000000       0.000000
75%    54682.000000    61.000000      0.000000       0.000000
max    72940.000000    82.000000      1.000000       1.000000

       avg_glucose_level          bmi       stroke
count        5110.000000  4909.000000  5110.000000
mean          106.147677    28.893237     0.048728
std            45.283560     7.854067     0.215320
min            55.120000    10.300000     0.000000
25%            77.245000    23.500000     0.000000
50%            91.885000    28.100000     0.000000
75%           114.090000    33.100000     0.000000
max           271.740000    97.600000     1.000000
```

[9]:
```
df.describe(include='object')  #decription des objects ( catégorique) o all both
```

[9]:
```
        gender ever_married work_type Residence_type smoking_status
count     5110         5110      5110           5110           5110
unique       3            2         5              2              4
top     Female          Yes   Private          Urban   never smoked
freq      2994         3353      2925           2596           1892
```

[10]:
```
df.isnull().sum() #verification des valeurs manquantes
```

[10]:
```
id                     0
gender                 0
age                    0
hypertension           0
heart_disease          0
ever_married           0
work_type              0
Residence_type         0
avg_glucose_level      0
bmi                  201
smoking_status         0
stroke                 0
```

```
dtype: int64
```

[11]:
```
df=df.drop(["id"], axis=1) # supprimer la colonne id
```

[12]:
```
df.bmi=df.bmi.fillna(df.bmi.mean()) #remplir les valeurs manquantes en␣
 ↪utilisant la valeur moyenne
```

[13]:
```
df.shape
```

[13]:
```
(5110, 11)
```

[14]:
```
df.isnull().sum()
```

[14]:
```
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64
```

[15]:
```
df.stroke[df.stroke==1].count() #calculer le nombre de stroke==1
```

[15]:
```
249
```

[16]:
```
df.hypertension[df.hypertension==1].count()
```

[16]:
```
498
```

[17]:
```
246/5110 # pourcentage de stroke par rapport total
```

[17]:
```
0.04814090019569472
```

[18]:
```
df.bmi.quantile(0.25) #quantile d'ordre 0.25
# ou bien dataset['bmi'].quantile(0.25)
```

[18]:
```
23.8
```

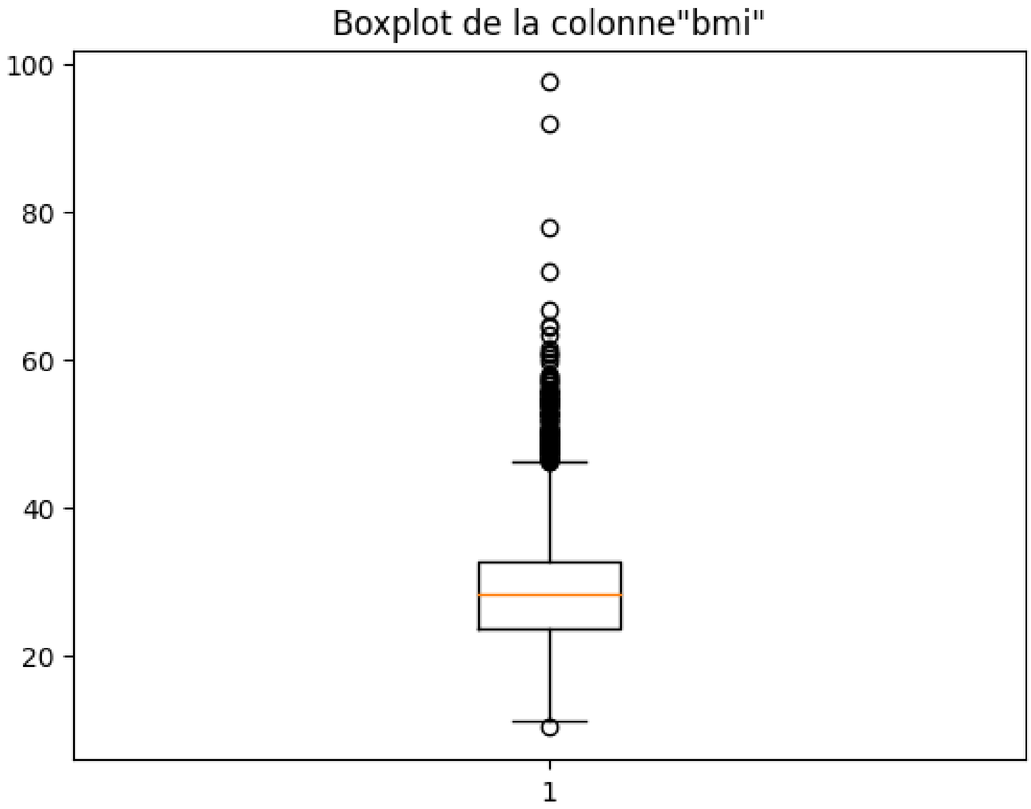
[19]:
```
df.bmi.quantile(0.5)
```

[19]:
```
28.4
```

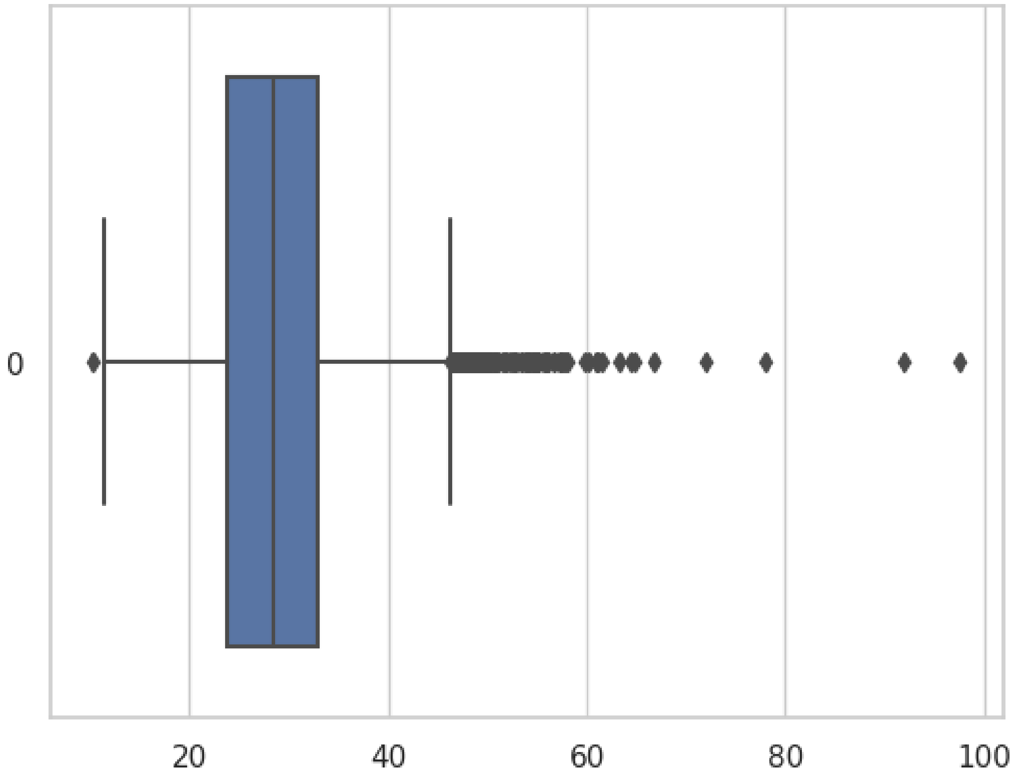
```
[20]: df.bmi.quantile(0.75)
```

```
[20]: 32.8
```

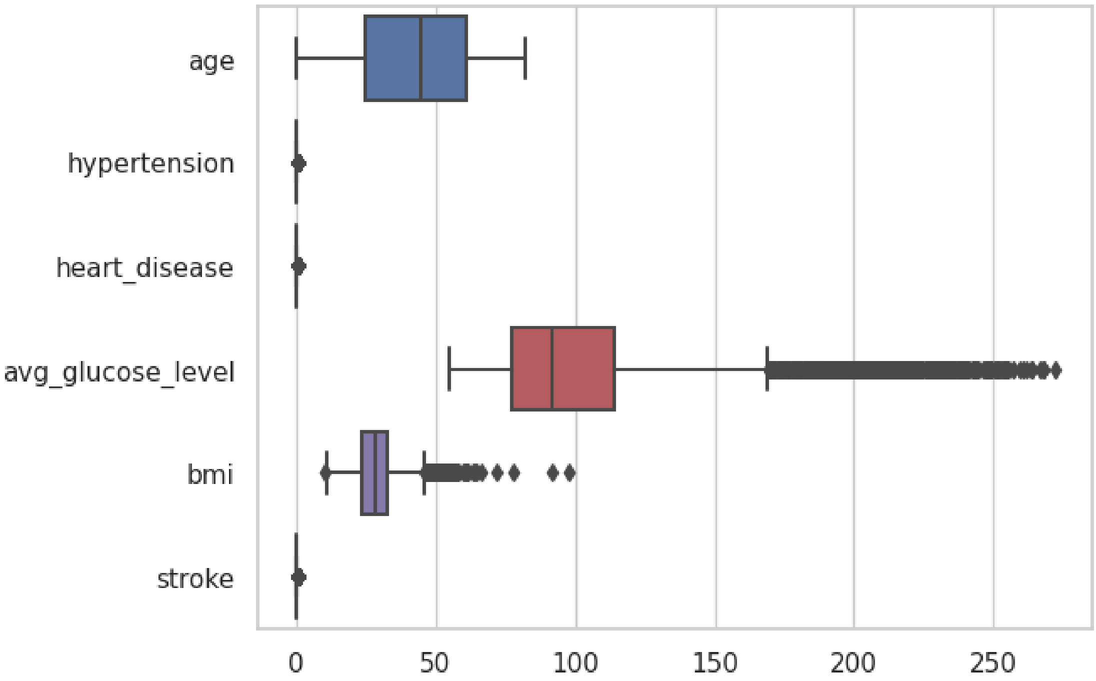
```
[21]: plt.boxplot(df['bmi'])
      plt.title('Boxplot de la colonne"bmi"')
      plt.show()
```



```
[22]: sns.set (style ='whitegrid')
      sns.boxplot(df.bmi, orient='h')
      plt.show()
```

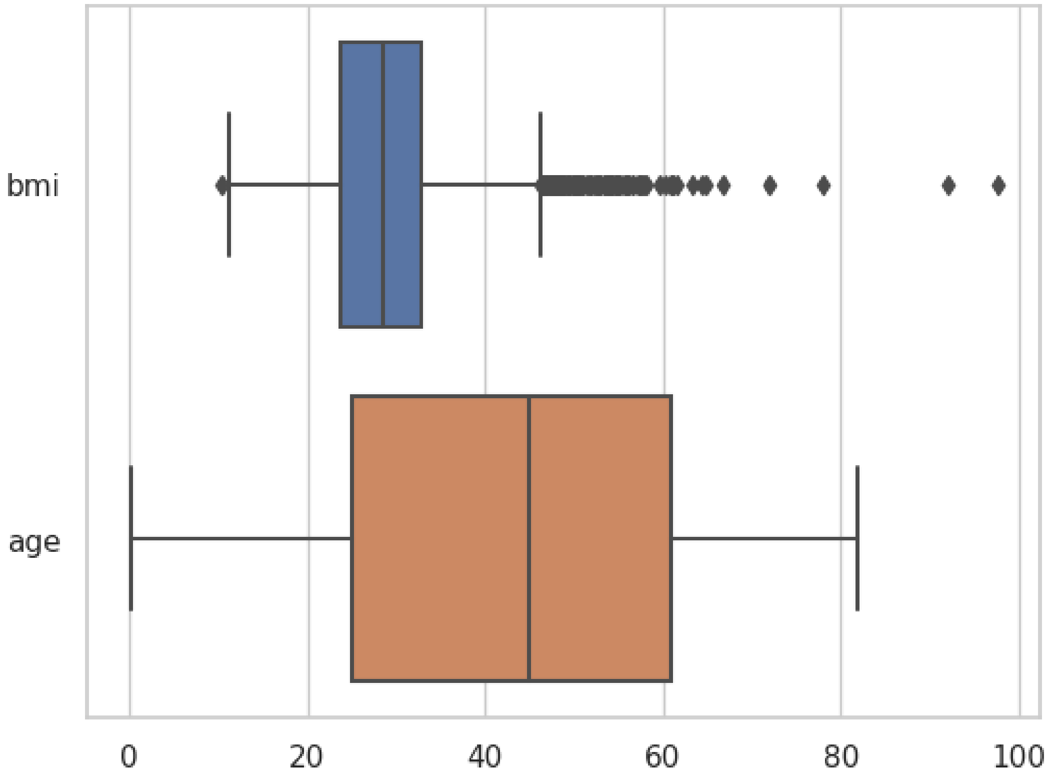


```
[23]: sns.set (style ='whitegrid')
      sns.boxplot(df, orient='h')
      plt.show()
```



[24]:

```
my_list=['bmi', 'age']
data = df[my_list]
sns.set (style ='whitegrid')
sns.boxplot(data, orient='h')
plt.show()
```



```
[25]: q1=df['bmi'].quantile(0.25)
      q3=df['bmi'].quantile(0.75)
      IQR = q3-q1
      BM=q3+1.5*IQR
      BI=q1-1.5*IQR
      df=df[df['bmi']<= BM]
      df= df[df['bmi']>= BI]
```

```
[26]: df.shape
```

```
[26]: (4984, 11)
```

```
[27]: q1=df['avg_glucose_level'].quantile(0.25)
      q3=df['avg_glucose_level'].quantile(0.75)
      IQR = q3-q1
      BM=q3+1.5*IQR
      BI=q1-1.5*IQR
      df=df[df['avg_glucose_level']<= BM]
      df= df[df['avg_glucose_level']>= BI]
```

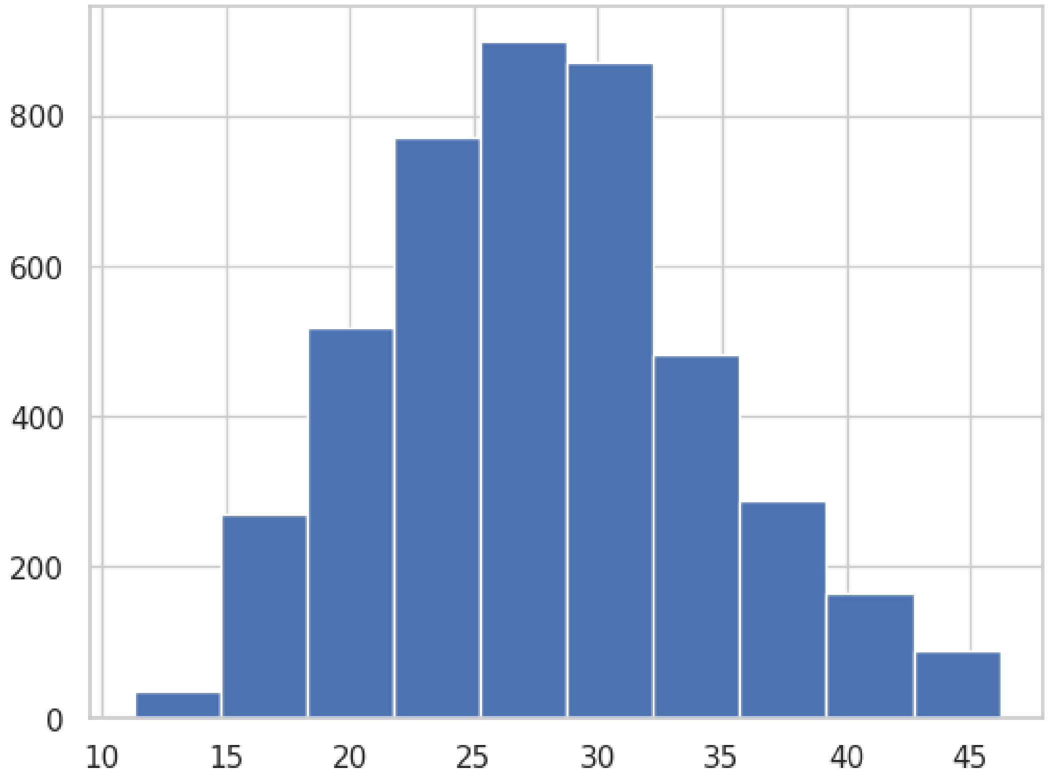
```
[28]: df.shape
```

[28]:

```
(4390, 11)
```

[29]:

```
plt.hist(df.bmi)
plt.show()
```



[30]:

```
#x=df.bmi.min()
#y=df.bmi.max()
#df.bmi=(df.bmi-x)/(y-x)
```

[31]:

```
df.head()
```

[31]:

```
   gender   age  hypertension  heart_disease ever_married work_type  \
2    Male  80.0             0              1          Yes   Private
6    Male  74.0             1              1          Yes   Private
7  Female  69.0             0              0           No   Private
8  Female  59.0             0              0          Yes   Private
9  Female  78.0             0              0          Yes   Private

  Residence_type  avg_glucose_level        bmi smoking_status  stroke
2          Rural             105.92  32.500000   never smoked       1
6          Rural              70.09  27.400000   never smoked       1
7          Urban              94.39  22.800000   never smoked       1
```

```
8          Rural             76.15  28.893237        Unknown       1
9          Urban             58.57  24.200000        Unknown       1
```

[32]:
```
#x=df.age.min()
#y=df.age.max()
#df.age=(df.age-x)/(y-x)
```

[33]:
```
df.head()
```

[33]:
```
   gender   age  hypertension  heart_disease ever_married work_type  \
2    Male  80.0             0              1          Yes   Private
6    Male  74.0             1              1          Yes   Private
7  Female  69.0             0              0           No   Private
8  Female  59.0             0              0          Yes   Private
9  Female  78.0             0              0          Yes   Private

  Residence_type  avg_glucose_level        bmi smoking_status  stroke
2          Rural             105.92  32.500000   never smoked       1
6          Rural              70.09  27.400000   never smoked       1
7          Urban              94.39  22.800000   never smoked       1
8          Rural              76.15  28.893237        Unknown       1
9          Urban              58.57  24.200000        Unknown       1
```

[34]:
```
#x=df.avg_glucose_level.min()
#y=df.avg_glucose_level.max()
#df.avg_glucose_level=(df.avg_glucose_level-x)/(y-x)
```

[35]:
```
df.head()
```

[35]:
```
   gender   age  hypertension  heart_disease ever_married work_type  \
2    Male  80.0             0              1          Yes   Private
6    Male  74.0             1              1          Yes   Private
7  Female  69.0             0              0           No   Private
8  Female  59.0             0              0          Yes   Private
9  Female  78.0             0              0          Yes   Private

  Residence_type  avg_glucose_level        bmi smoking_status  stroke
2          Rural             105.92  32.500000   never smoked       1
6          Rural              70.09  27.400000   never smoked       1
7          Urban              94.39  22.800000   never smoked       1
8          Rural              76.15  28.893237        Unknown       1
9          Urban              58.57  24.200000        Unknown       1
```

[36]:
```
df.corr(method='pearson')
```

```
/tmp/ipykernel_6708/4020201063.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
```

```
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  df.corr(method='pearson')
```

[36]:

```
                        age  hypertension  heart_disease  avg_glucose_level  \
age                1.000000      0.252344       0.239817          -0.026012
hypertension       0.252344      1.000000       0.090573           0.002433
heart_disease      0.239817      0.090573       1.000000          -0.000535
avg_glucose_level -0.026012      0.002433      -0.000535           1.000000
bmi                0.377089      0.150011       0.055670           0.002780
stroke             0.227138      0.113044       0.089726           0.003014

                        bmi    stroke
age                0.377089  0.227138
hypertension       0.150011  0.113044
heart_disease      0.055670  0.089726
avg_glucose_level  0.002780  0.003014
bmi                1.000000  0.034124
stroke             0.034124  1.000000
```

[37]:

```
from sklearn.preprocessing import LabelEncoder #apprentissage automatique
```

[38]:

```
label = LabelEncoder()
x= df.iloc[:,0:10]
y=df.iloc[:,10]
x.head()
```

[38]:

```
   gender   age  hypertension  heart_disease ever_married work_type  \
2    Male  80.0             0              1          Yes   Private
6    Male  74.0             1              1          Yes   Private
7  Female  69.0             0              0           No   Private
8  Female  59.0             0              0          Yes   Private
9  Female  78.0             0              0          Yes   Private

  Residence_type  avg_glucose_level        bmi smoking_status
2          Rural             105.92  32.500000   never smoked
6          Rural              70.09  27.400000   never smoked
7          Urban              94.39  22.800000   never smoked
8          Rural              76.15  28.893237        Unknown
9          Urban              58.57  24.200000        Unknown
```

[39]:

```
X= x.values #matrice
Y=y.values #vecteur
X[0:5,:]
```

[39]:

```
array([['Male', 80.0, 0, 1, 'Yes', 'Private', 'Rural', 105.92, 32.5,
        'never smoked'],
       ['Male', 74.0, 1, 1, 'Yes', 'Private', 'Rural', 70.09, 27.4,
        'never smoked'],
       ['Female', 69.0, 0, 0, 'No', 'Private', 'Urban', 94.39, 22.8,
        'never smoked'],
       ['Female', 59.0, 0, 0, 'Yes', 'Private', 'Rural', 76.15,
        28.893236911794666, 'Unknown'],
       ['Female', 78.0, 0, 0, 'Yes', 'Private', 'Urban', 58.57, 24.2,
        'Unknown']], dtype=object)
```

[40]:
```
X[:,0]= label.fit_transform(X[:,0])
```

[41]:
```
X[:,4]= label.fit_transform(X[:,4])
```

[42]:
```
X[:,5]= label.fit_transform(X[:,5])
```

[43]:
```
X[:,6]= label.fit_transform(X[:,6])
```

[44]:
```
X[:,9]= label.fit_transform(X[:,9])
```

[45]:
```
X[0:5,:]
```

[45]:
```
array([[1, 80.0, 0, 1, 1, 2, 0, 105.92, 32.5, 2],
       [1, 74.0, 1, 1, 1, 2, 0, 70.09, 27.4, 2],
       [0, 69.0, 0, 0, 0, 2, 1, 94.39, 22.8, 2],
       [0, 59.0, 0, 0, 1, 2, 0, 76.15, 28.893236911794666, 0],
       [0, 78.0, 0, 0, 1, 2, 1, 58.57, 24.2, 0]], dtype=object)
```

[46]:
```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
ct =ColumnTransformer([('gender', OneHotEncoder(),[0])],␣
 ↪remainder='passthrough')
```

[47]:
```
X = ct.fit_transform(X) #appliquer la transformation d'une maniere séléctive␣
 ↪sur la colonne specifique
X= X[:,1:] # supprimmer la 1 ere colonne
```

[48]:
```
X.shape
```

[48]:
```
(4390, 11)
```

[49]:
```
ct =ColumnTransformer([('ever_married', OneHotEncoder(),[5])],␣
 ↪remainder='passthrough')
```

[50]:
```
X = ct.fit_transform(X)
X= X[:,1:]
```

[51]:
```
X.shape
```

[51]: (4390, 11)

[52]:
```
ct =ColumnTransformer([('work_type', OneHotEncoder(),[6])],␣
 ↪remainder='passthrough')
X = ct.fit_transform(X)
X= X[:,1:]
```

[53]:
```
X.shape
```

[53]: (4390, 14)

[54]:
```
ct =ColumnTransformer([('Residence_type', OneHotEncoder(),[10])],␣
 ↪remainder='passthrough')
X = ct.fit_transform(X)
X= X[:,1:]
```

[55]:
```
X.shape
```

[55]: (4390, 14)

[56]:
```
ct =ColumnTransformer([('smoking_status', OneHotEncoder(),[13])],␣
 ↪remainder='passthrough')
X = ct.fit_transform(X)
X= X[:,1:]
```

[57]:
```
X.shape
```

[57]: (4390, 16)

[58]:
```
x1=df.iloc[:,0:10]
y1=df.iloc[:,10]
```

[59]:
```
x1=pd.get_dummies(data=x1,drop_first=True)
```

[60]:
```
X.shape
```

[60]: (4390, 16)

[61]:
```
x1.head()
```

[61]:
```
    age  hypertension  heart_disease  avg_glucose_level        bmi  \
2  80.0             0              1             105.92  32.500000
6  74.0             1              1              70.09  27.400000
7  69.0             0              0              94.39  22.800000
8  59.0             0              0              76.15  28.893237
9  78.0             0              0              58.57  24.200000
```

```
   gender_Male  gender_Other  ever_married_Yes  work_type_Never_worked  \
2            1             0                 1                       0
6            1             0                 1                       0
7            0             0                 0                       0
8            0             0                 1                       0
9            0             0                 1                       0

   work_type_Private  work_type_Self-employed  work_type_children  \
2                  1                        0                   0
6                  1                        0                   0
7                  1                        0                   0
8                  1                        0                   0
9                  1                        0                   0

   Residence_type_Urban  smoking_status_formerly smoked  \
2                     0                               0
6                     0                               0
7                     1                               0
8                     0                               0
9                     1                               0

   smoking_status_never smoked  smoking_status_smokes
2                            1                      0
6                            1                      0
7                            1                      0
8                            0                      0
9                            0                      0
```

[62]:
```
from sklearn.model_selection import train_test_split
X_train ,X_test , y_train , y_test=train_test_split(X,y,test_size=0.
 ↪2,random_state=0)
```

[63]:
```
from sklearn.preprocessing import MinMaxScaler #standardScaler
#scaler = standardscaler()
scaler = MinMaxScaler()
X_train_sc = scaler.fit_transform(X_train) #appliquer la transformation␣
 ↪s'adapter sur les données de x_tain ( min , max ) et enregistrer les valeurs␣
 ↪min et max
X_Test_sc = scaler.transform (X_test)
```

[64]:
```
X_train_sc [0:5,:]
```

[64]:
```
array([[0.        , 0.        , 1.        , 1.        , 0.        ,
        1.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.30419922, 0.        , 0.        , 0.1150702 ,
        0.20916905],
       [0.        , 0.        , 0.        , 1.        , 0.        ,
        0.        , 0.        , 1.        , 0.        , 1.        ,
        0.        , 0.08447266, 0.        , 0.        , 0.19113204,
        0.17191977],
       [1.        , 0.        , 0.        , 1.        , 0.        ,
        1.        , 0.        , 0.        , 1.        , 1.        ,
        0.        , 0.76806641, 0.        , 0.        , 0.33010485,
        0.68481375],
       [0.        , 0.        , 1.        , 0.        , 0.        ,
        1.        , 0.        , 0.        , 1.        , 0.        ,
        0.        , 0.59716797, 0.        , 0.        , 0.24249156,
        0.97707736],
       [0.        , 1.        , 0.        , 0.        , 0.        ,
        0.        , 1.        , 0.        , 1.        , 0.        ,
        0.        , 0.97558594, 0.        , 1.        , 0.42598187,
        0.50143266]])
```

[ ]: