```
import numpy as np
import pandas as pd

test1 = pd.read_csv("C:\\Users\\HP\\Desktop\\phy\\health care
diabetes.csv")
test1.head()
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin
BMI  \
0            6      148             72             35        0  33.6

1            1       85             66             29        0  26.6

2            8      183             64              0        0  23.3

3            1       89             66             23       94  28.1

4            0      137             40             35      168  43.1


   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

```
df = test1.replace(0,'NAN')
df.head()
```

```
  Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI  \
0           6     148            72            35     NAN  33.6
1           1      85            66            29     NAN  26.6
2           8     183            64           NAN     NAN  23.3
3           1      89            66            23      94  28.1
4         NAN     137            40            35     168  43.1

   DiabetesPedigreeFunction  Age Outcome
0                     0.627   50       1
1                     0.351   31     NAN
2                     0.672   32       1
3                     0.167   21     NAN
4                     2.288   33       1
```

```
df1 = df.replace('NAN',np.nan)
df1.head()
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin
BMI  \
0          6.0    148.0           72.0           35.0      NaN  33.6
```

```
1          1.0     85.0           66.0           29.0      NaN  26.6

2          8.0    183.0           64.0            NaN      NaN  23.3

3          1.0     89.0           66.0           23.0     94.0  28.1

4          NaN    137.0           40.0           35.0    168.0  43.1


   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50      1.0
1                     0.351   31      NaN
2                     0.672   32      1.0
3                     0.167   21      NaN
4                     2.288   33      1.0
```

```
df2= df1.interpolate()
df2.head()
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin
BMI  \
0          6.0    148.0           72.0           35.0      NaN  33.6

1          1.0     85.0           66.0           29.0      NaN  26.6

2          8.0    183.0           64.0           26.0      NaN  23.3

3          1.0     89.0           66.0           23.0     94.0  28.1

4          3.0    137.0           40.0           35.0    168.0  43.1


   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50      1.0
1                     0.351   31      1.0
2                     0.672   32      1.0
3                     0.167   21      1.0
4                     2.288   33      1.0
```

```
x = df2['Insulin'].mean()
df2['Insulin'].fillna(x,inplace = True)
print(df2)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin
BMI  \
0            6.0    148.0           72.0           35.0  159.045098
33.6
1            1.0     85.0           66.0           29.0  159.045098
26.6
2            8.0    183.0           64.0           26.0  159.045098
```

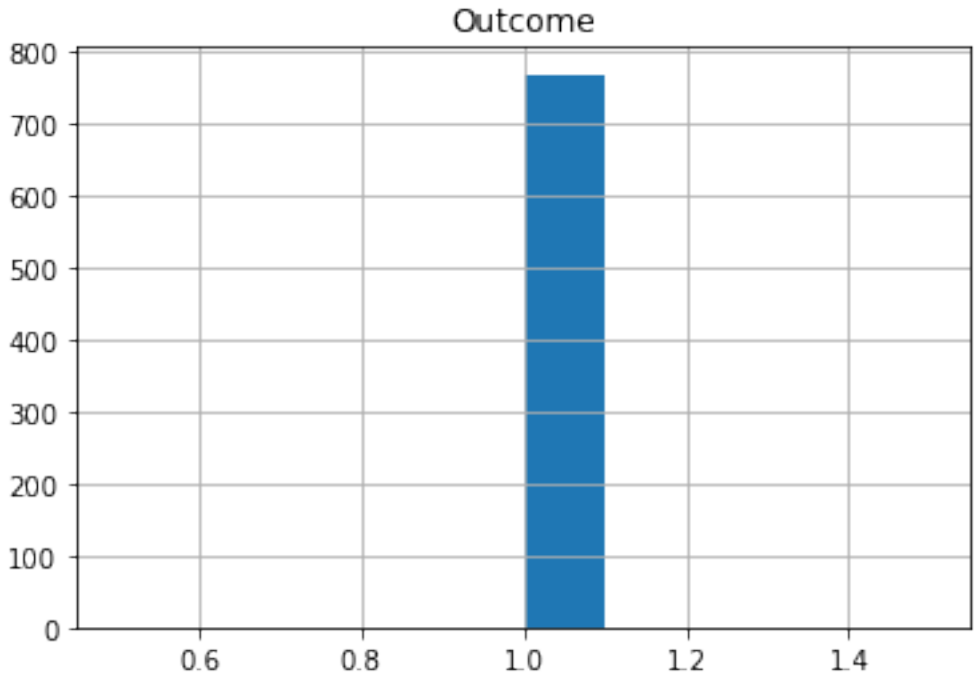
```
23.3
3            1.0     89.0           66.0           23.0   94.000000
28.1
4            3.0    137.0           40.0           35.0  168.000000
43.1
..           ...      ...            ...            ...         ...
...
763         10.0    101.0           76.0           48.0  180.000000
32.9
764          2.0    122.0           70.0           27.0  146.000000
36.8
765          5.0    121.0           72.0           23.0  112.000000
26.2
766          1.0    126.0           60.0           27.0  112.000000
30.1
767          1.0     93.0           70.0           31.0  112.000000
30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50      1.0
1                       0.351   31      1.0
2                       0.672   32      1.0
3                       0.167   21      1.0
4                       2.288   33      1.0
..                        ...  ...      ...
763                     0.171   63      1.0
764                     0.340   27      1.0
765                     0.245   30      1.0
766                     0.349   47      1.0
767                     0.315   23      1.0

[768 rows x 9 columns]

df2.head()

   Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin
BMI  \
0          6.0    148.0           72.0           35.0  159.045098
33.6
1          1.0     85.0           66.0           29.0  159.045098
26.6
2          8.0    183.0           64.0           26.0  159.045098
23.3
3          1.0     89.0           66.0           23.0   94.000000
28.1
4          3.0    137.0           40.0           35.0  168.000000
43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50      1.0
```

```
1                       0.351   31      1.0
2                       0.672   32      1.0
3                       0.167   21      1.0
4                       2.288   33      1.0
```

```
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
```

```
df2.hist(column = 'Outcome')
```

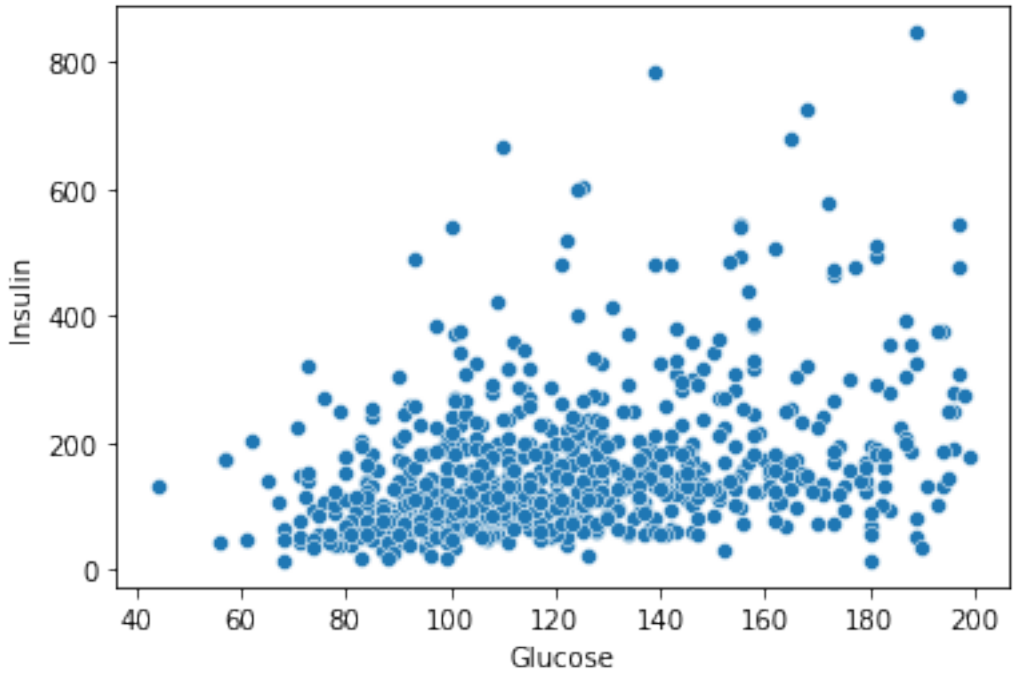
```
array([[<AxesSubplot:title={'center':'Outcome'}>]], dtype=object)
```




```
import seaborn as sns
```

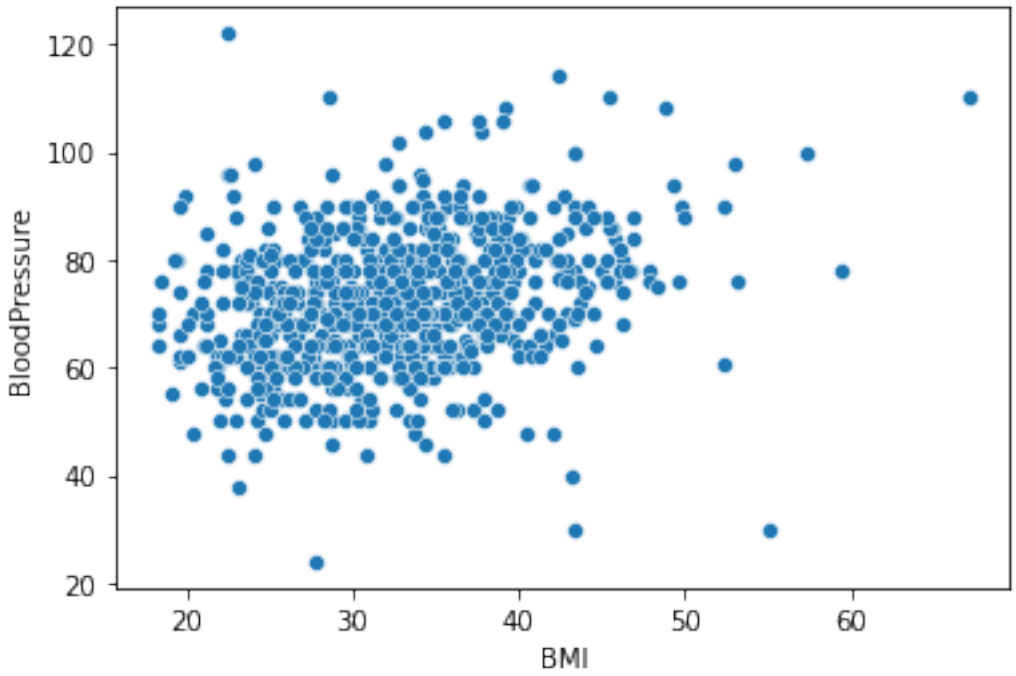
```
sns.scatterplot(df2.Glucose, df2.Insulin);
```

```
C:\ANACONDA\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```



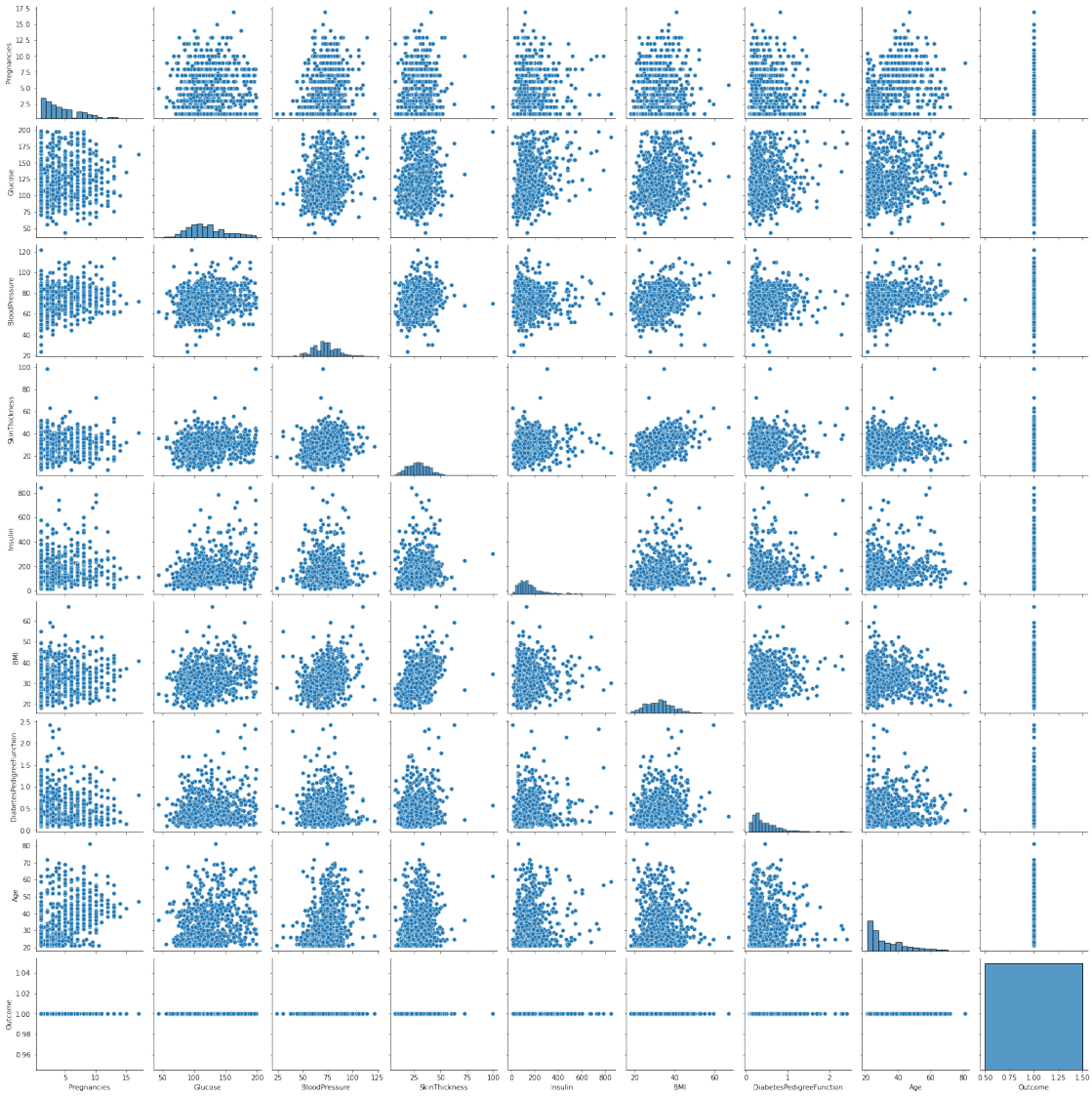

```
sns.scatterplot(df2.BMI, df2.BloodPressure);
```

```
C:\ANACONDA\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```



```
sns.pairplot(df2)
```

```
<seaborn.axisgrid.PairGrid at 0x19db391b340>
```



##CORRELATION ANALYSIS

```
np.corrcoef(df2['Glucose'],df2['Insulin'])
```

```
array([[1.        , 0.36233601],
       [0.36233601, 1.        ]])
```

```
np.corrcoef(df2['BloodPressure'],df2['BMI'])
```

```
array([[1.        , 0.27397764],
       [0.27397764, 1.        ]])
```

```
df2.corr()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness
Pregnancies	1.000000	0.158430	0.233739	0.110458
Glucose	0.158430	1.000000	0.216927	0.183719
BloodPressure	0.233739	0.216927	1.000000	0.175081
SkinThickness	0.110458	0.183719	0.175081	1.000000
Insulin	0.130491	0.362336	0.050028	0.139029
BMI	0.090296	0.231684	0.273978	0.494883
DiabetesPedigreeFunction	-0.023819	0.138814	-0.005157	0.103014
Age	0.491399	0.268057	0.318358	0.130788
Outcome	NaN	NaN	NaN	NaN

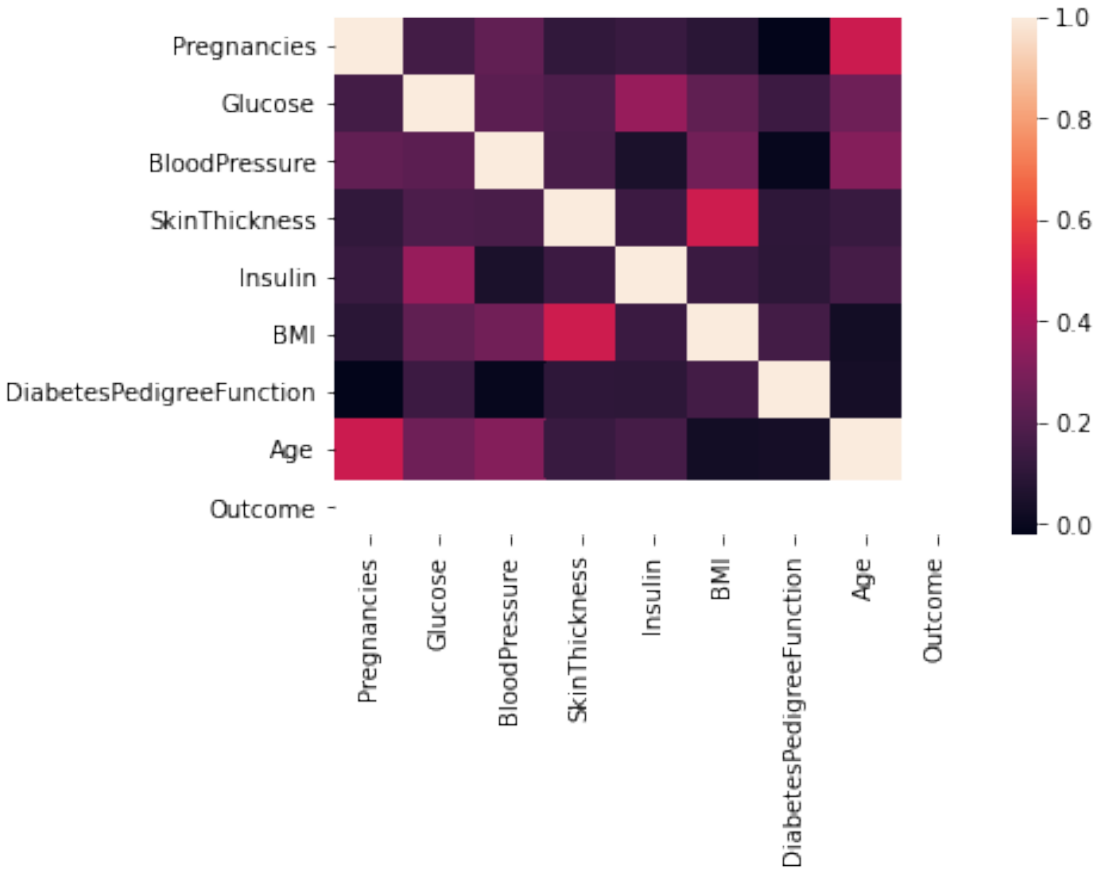
	Insulin	BMI	DiabetesPedigreeFunction
Pregnancies	0.130491	0.090296	-0.023819
Glucose	0.362336	0.231684	0.138814
BloodPressure	0.050028	0.273978	-0.005157
SkinThickness	0.139029	0.494883	0.103014
Insulin	1.000000	0.136063	0.100111
BMI	0.136063	1.000000	0.157147
DiabetesPedigreeFunction	0.100111	0.157147	1.000000
Age	0.161294	0.025975	0.033561
Outcome	NaN	NaN	NaN

	Age	Outcome
Pregnancies	0.491399	NaN
Glucose	0.268057	NaN
BloodPressure	0.318358	NaN
SkinThickness	0.130788	NaN
Insulin	0.161294	NaN
BMI	0.025975	NaN

```
DiabetesPedigreeFunction  0.033561      NaN
Age                       1.000000      NaN
Outcome                        NaN      NaN
```

```
sns.heatmap(df2.corr())
```

```
<AxesSubplot:>
```




```
train = pd.read_csv("C:\\Users\\HP\\Desktop\\phy\\train.csv")
train.head()
```

```
   Age Gender AppointmentRegistration        ApointmentData
DayOfTheWeek  \
0   38      F    2015-10-20T08:33:56Z  2015-10-23T00:00:00Z
Friday
1   56      F    2014-02-03T10:05:26Z  2014-02-20T00:00:00Z
Thursday
2   27      F    2014-04-29T07:57:32Z  2014-05-20T00:00:00Z
Tuesday
3   24      M    2014-04-02T13:53:37Z  2014-05-06T00:00:00Z
Tuesday
4   48      F    2014-01-07T10:07:17Z  2014-01-30T00:00:00Z
Thursday
```

```
    Status  Diabetes  Alcoolism  HiperTension  Handcap  Smokes
Scholarship  \
0  No-Show         0          0             0        0       1
0
1  No-Show         1          0             1        0       0
0
2  Show-Up         0          0             0        0       0
0
3  Show-Up         0          0             0        0       0
0
4  Show-Up         0          0             0        0       0
0

   Tuberculosis  Sms_Reminder  AwaitingTime
0             0             0            -3
1             0             1           -17
2             0             0           -21
3             0             0           -34
4             0             1           -23
```

```
test = pd.read_csv("C:\\Users\\HP\\Desktop\\phy\\healthcare
appointment data.csv")
test.head()
```

```
   Age Gender AppointmentRegistration        ApointmentData
DayOfTheWeek  \
0   19      M    2014-12-16T14:46:25Z  2015-01-14T00:00:00Z
Wednesday
1   24      F    2015-08-18T07:01:26Z  2015-08-19T00:00:00Z
Wednesday
2    4      F    2014-02-17T12:53:46Z  2014-02-18T00:00:00Z
Tuesday
3    5      M    2014-07-23T17:02:11Z  2014-08-07T00:00:00Z
Thursday
4   38      M    2015-10-21T15:20:09Z  2015-10-27T00:00:00Z
Tuesday

    Status  Diabetes  Alcoolism  HiperTension  Handcap  Smokes
Scholarship  \
0  Show-Up         0          0             0        0       0
0
1  Show-Up         0          0             0        0       0
0
2  Show-Up         0          0             0        0       0
0
3  Show-Up         0          0             0        0       0
0
4  Show-Up         0          0             0        0       0
0
```

```
   Tuberculosis  Sms_Reminder  AwaitingTime
0             0             0           -29
1             0             0            -1
2             0             0            -1
3             0             1           -15
4             0             1            -6
```

```
train.describe()
```

```
                 Age       Diabetes      Alcoolism   HiperTension  \
count  210000.000000  210000.000000  210000.000000  210000.000000
mean       37.761824       0.077290       0.024676       0.214862
std        22.794334       0.267052       0.155137       0.410727
min        -1.000000       0.000000       0.000000       0.000000
25%        19.000000       0.000000       0.000000       0.000000
50%        38.000000       0.000000       0.000000       0.000000
75%        56.000000       0.000000       0.000000       0.000000
max       113.000000       1.000000       1.000000       1.000000

             Handcap         Smokes    Scholarship   Tuberculosis  \
count  210000.000000  210000.000000  210000.000000  210000.000000
mean        0.020471       0.052033       0.097738       0.000429
std         0.155854       0.222095       0.296961       0.020698
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.000000       0.000000
50%         0.000000       0.000000       0.000000       0.000000
75%         0.000000       0.000000       0.000000       0.000000
max         4.000000       1.000000       1.000000       1.000000

        Sms_Reminder   AwaitingTime
count  210000.000000  210000.000000
mean        0.574238     -13.833538
std         0.499776      15.685272
min         0.000000    -398.000000
25%         0.000000     -20.000000
50%         1.000000      -8.000000
75%         1.000000      -4.000000
max         2.000000      -1.000000
```

```
test.describe()
```

```
                 Age       Diabetes      Alcoolism   HiperTension  \
count  300000.000000  300000.000000  300000.000000  300000.000000
mean       37.808017       0.077967       0.025010       0.215890
std        22.809014       0.268120       0.156156       0.411439
min        -2.000000       0.000000       0.000000       0.000000
25%        19.000000       0.000000       0.000000       0.000000
50%        38.000000       0.000000       0.000000       0.000000
75%        56.000000       0.000000       0.000000       0.000000
max       113.000000       1.000000       1.000000       1.000000
```

```
             Handcap         Smokes    Scholarship   Tuberculosis  \
count  300000.000000  300000.000000  300000.000000  300000.000000
mean        0.020523       0.052370       0.096897       0.000450
std         0.155934       0.222772       0.295818       0.021208
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       0.000000       0.000000
50%         0.000000       0.000000       0.000000       0.000000
75%         0.000000       0.000000       0.000000       0.000000
max         4.000000       1.000000       1.000000       1.000000

        Sms_Reminder   AwaitingTime
count  300000.000000  300000.000000
mean        0.574173     -13.841813
std         0.499826      15.687697
min         0.000000    -398.000000
25%         0.000000     -20.000000
50%         1.000000      -8.000000
75%         1.000000      -4.000000
max         2.000000      -1.000000
```

```
train.shape
```

```
(210000, 15)
```

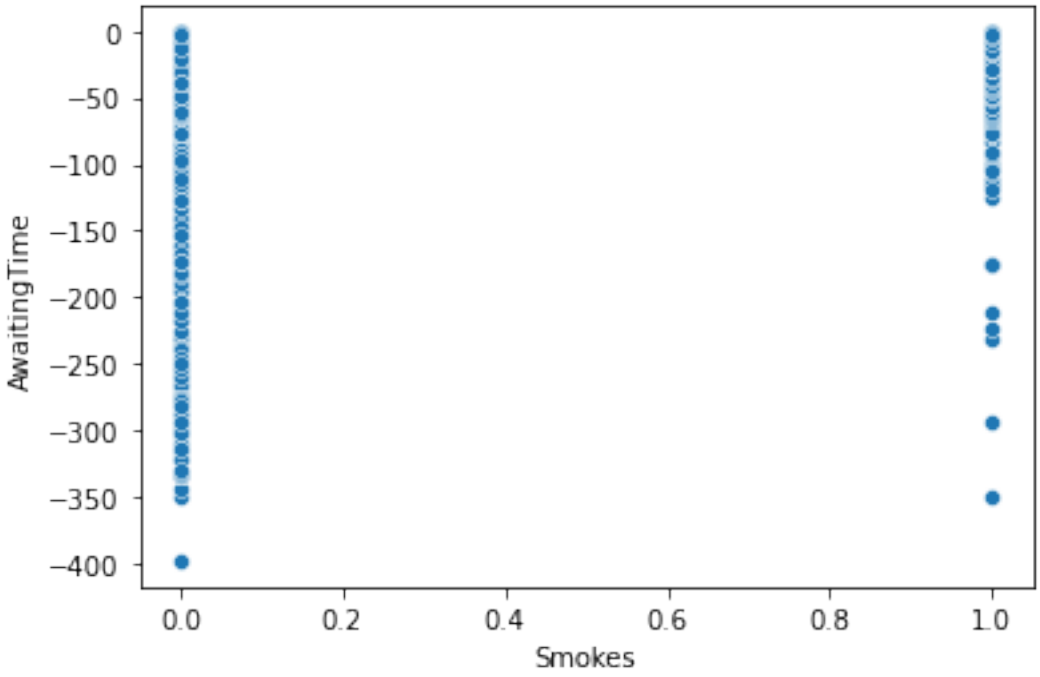
```
test.shape
```

```
(300000, 15)
```

```
sns.scatterplot(train.Smokes,train.AwaitingTime)
```

```
C:\ANACONDA\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Smokes', ylabel='AwaitingTime'>
```

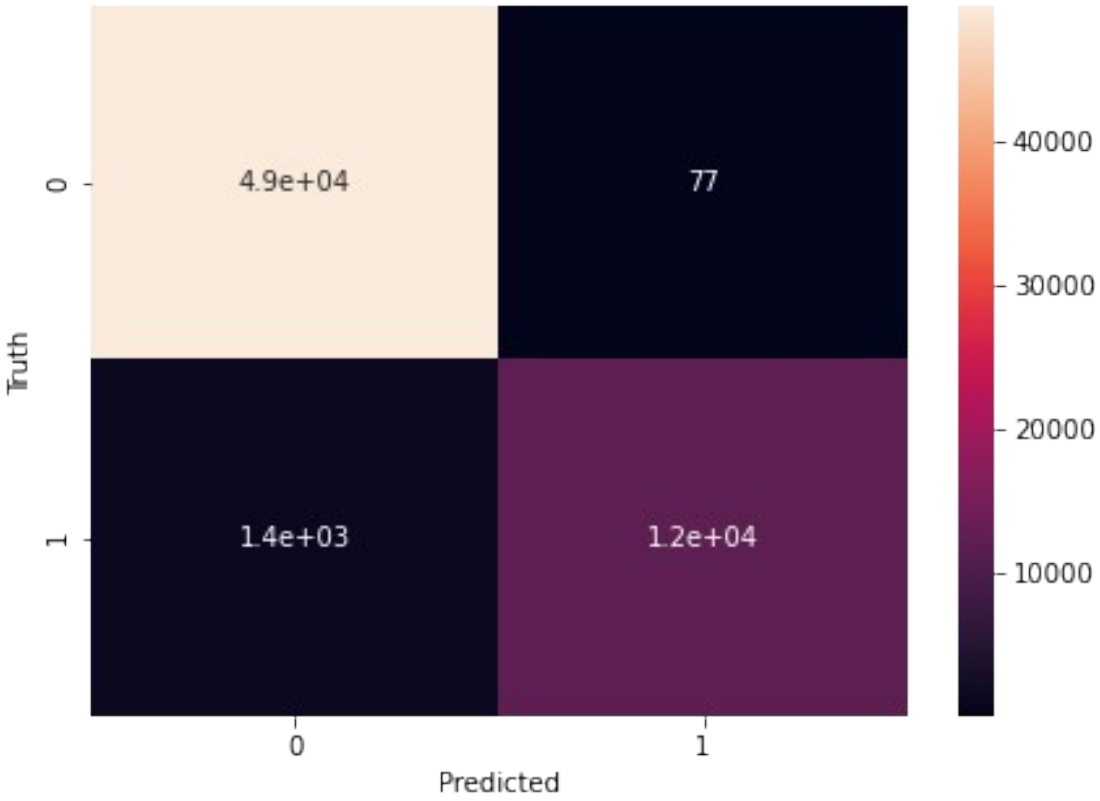



```
from sklearn.model_selection import train_test_split

x =
train.drop(['Gender','AppointmentRegistration','ApointmentData','DayOf
TheWeek','Status'], axis = 'columns')
y = train.HiperTension

x_train,x_test,y_train,y_test =
train_test_split(x,y,random_state=42,test_size=0.3)

x_train.shape,x_test.shape,y_train.shape,y_test.shape

((147000, 10), (63000, 10), (147000,), (63000,))

##KNN CLASSIFIER

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 10)
knn.fit(x_train, y_train)

KNeighborsClassifier(n_neighbors=10)

knn.score(x_test,y_test)

0.9763333333333334

from sklearn.metrics import confusion_matrix
y_pred = knn.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[49366,    77],
       [ 1414, 12143]], dtype=int64)
```

```
plt.figure(figsize=(7,5))
sns.heatmap(cm, annot = True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(42.0, 0.5, 'Truth')
```




```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99     49443
           1       0.99      0.90      0.94     13557

    accuracy                           0.98     63000
   macro avg       0.98      0.95      0.96     63000
weighted avg       0.98      0.98      0.98     63000
```