logistic-regression

February 28, 2024

[1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
df = pd.read_csv('framingham.csv')
df
```

[2]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	\
0	1	39	4.0	0	0.0	0.0	
1	0	46	2.0	0	0.0	0.0	
2	1	48	1.0	1	20.0	0.0	
3	0	61	3.0	1	30.0	0.0	
4	0	46	3.0	1	23.0	0.0	
…	…	…	…	…	…	…	
4233	1	50	1.0	1	1.0	0.0	
4234	1	51	3.0	1	43.0	0.0	
4235	0	48	2.0	1	20.0	NaN	
4236	0	44	1.0	1	15.0	0.0	
4237	0	52	2.0	0	0.0	0.0	

	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	\
0	0	0	0	195.0	106.0	70.0	26.97	
1	0	0	0	250.0	121.0	81.0	28.73	
2	0	0	0	245.0	127.5	80.0	25.34	
3	0	1	0	225.0	150.0	95.0	28.58	
4	0	0	0	285.0	130.0	84.0	23.10	
…	…	…	…	…	…	…	…	
4233	0	1	0	313.0	179.0	92.0	25.97	
4234	0	0	0	207.0	126.5	80.0	19.71	
4235	0	0	0	248.0	131.0	72.0	22.00	
4236	0	0	0	210.0	126.5	87.0	19.16	
4237	0	0	0	269.0	133.5	83.0	21.47	

	heartRate	glucose	TenYearCHD
0	80.0	77.0	0
1	95.0	76.0	0

```
2          75.0     70.0           0
3          65.0    103.0           1
4          85.0     85.0           0
…           …       …              …
4233       66.0     86.0           1
4234       65.0     68.0           0
4235       84.0     86.0           0
4236       86.0      NaN           0
4237       80.0    107.0           0

[4238 rows x 16 columns]
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

[4]:
```
df.isna().sum()
```

[4]:
```
male               0
age                0
education        105
currentSmoker      0
cigsPerDay        29
BPMeds            53
prevalentStroke    0
```

```
prevalentHyp          0
diabetes              0
totChol              50
sysBP                 0
diaBP                 0
BMI                  19
heartRate             1
glucose             388
TenYearCHD            0
dtype: int64
```

[5]:
```
median_col =␣
 ↪['education','cigsPerDay','BMI','BPMeds','totChol','glucose','heartRate']
```

[6]:
```
for i in median_col:
    med_value = df[i].median()
    df[i] = df[i].fillna(med_value)
```

[7]:
```
df.isna().sum()
```

[7]:
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

[8]:
```
df.duplicated().sum()
```

[8]:
```
0
```

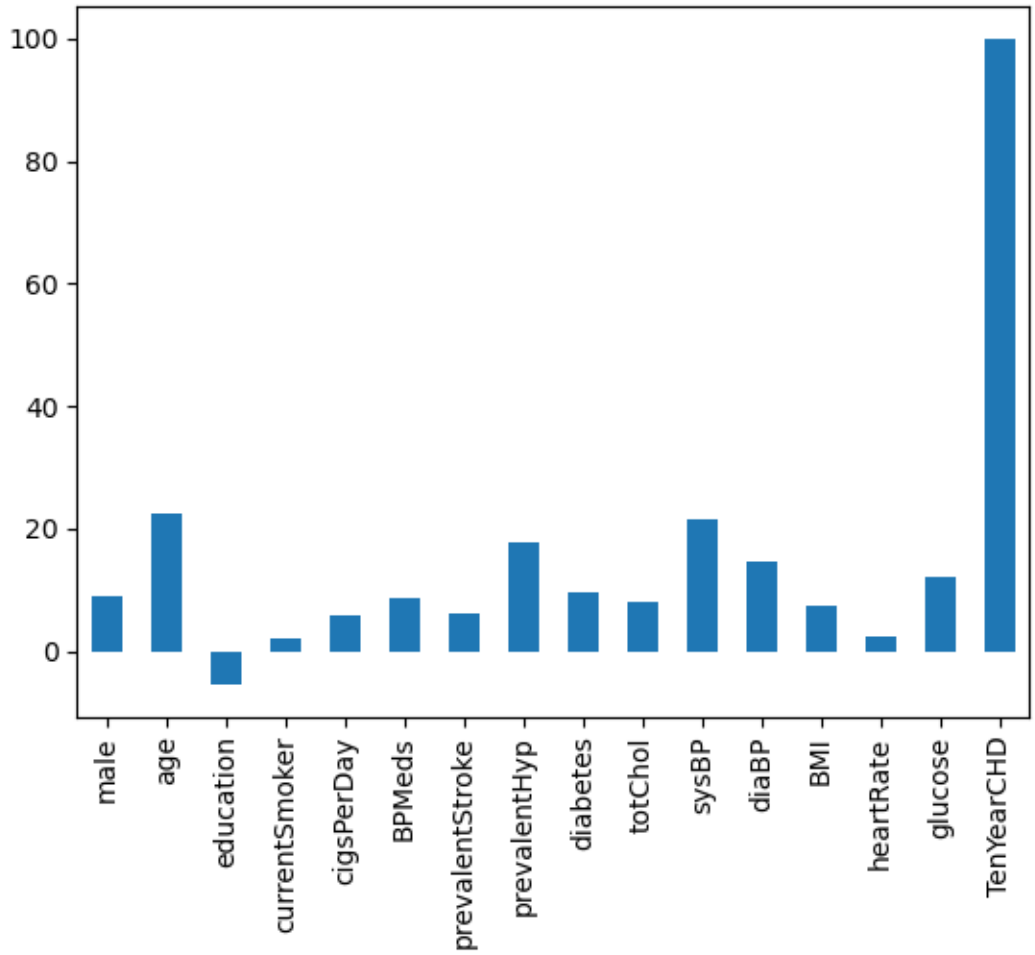
[9]:
```
df.corrwith(df['TenYearCHD'])*100
```

[9]:
```
male                 8.842757
age                 22.525610
education           -5.338264
```

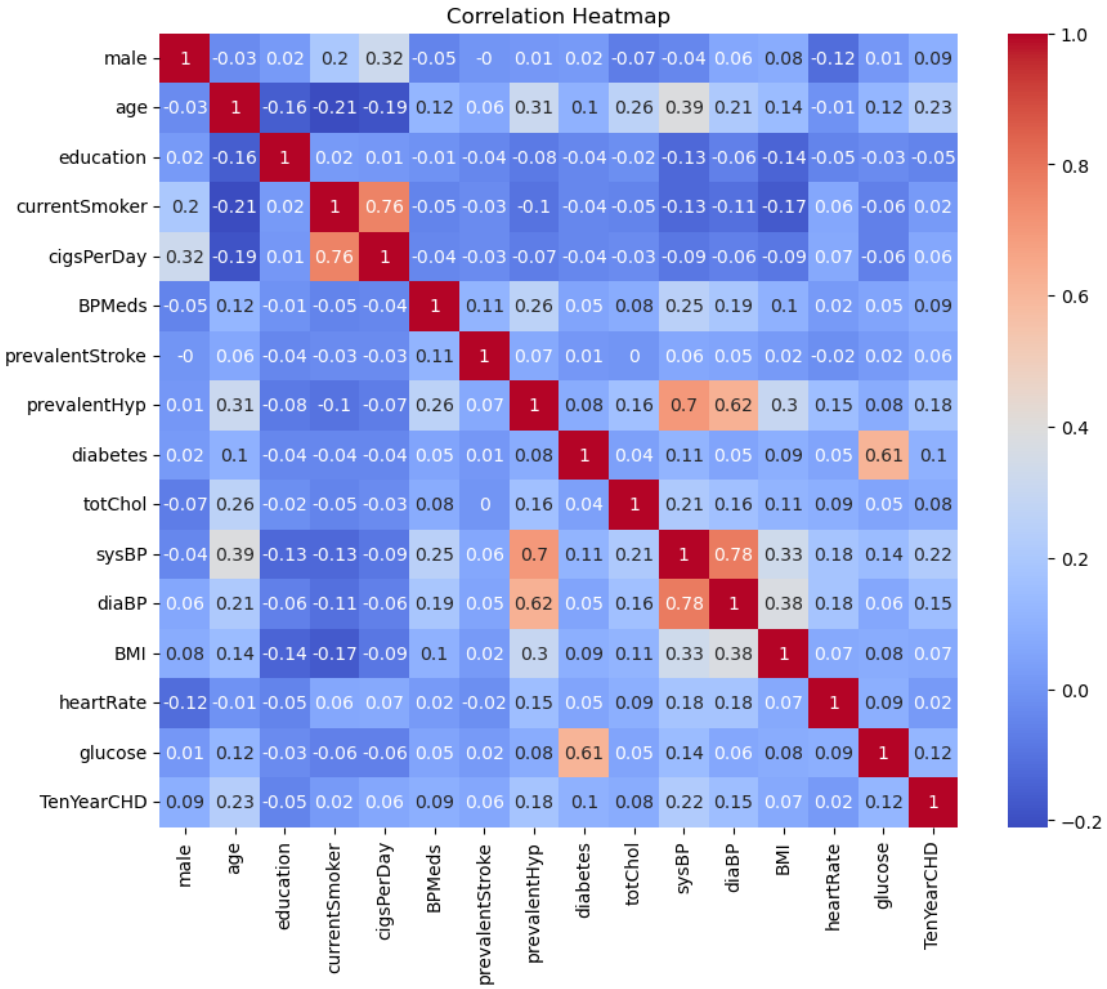
```
currentSmoker         1.945627
cigsPerDay            5.885914
BPMeds                8.641714
prevalentStroke       6.180995
prevalentHyp         17.760273
diabetes              9.731651
totChol               8.156572
sysBP                21.642904
diaBP                14.529910
BMI                   7.421662
heartRate             2.285676
glucose              12.127740
TenYearCHD          100.000000
dtype: float64
```

[19]:
```
(df.corrwith(df['TenYearCHD'])*100).plot(kind = 'bar')
```

[19]: <Axes: >



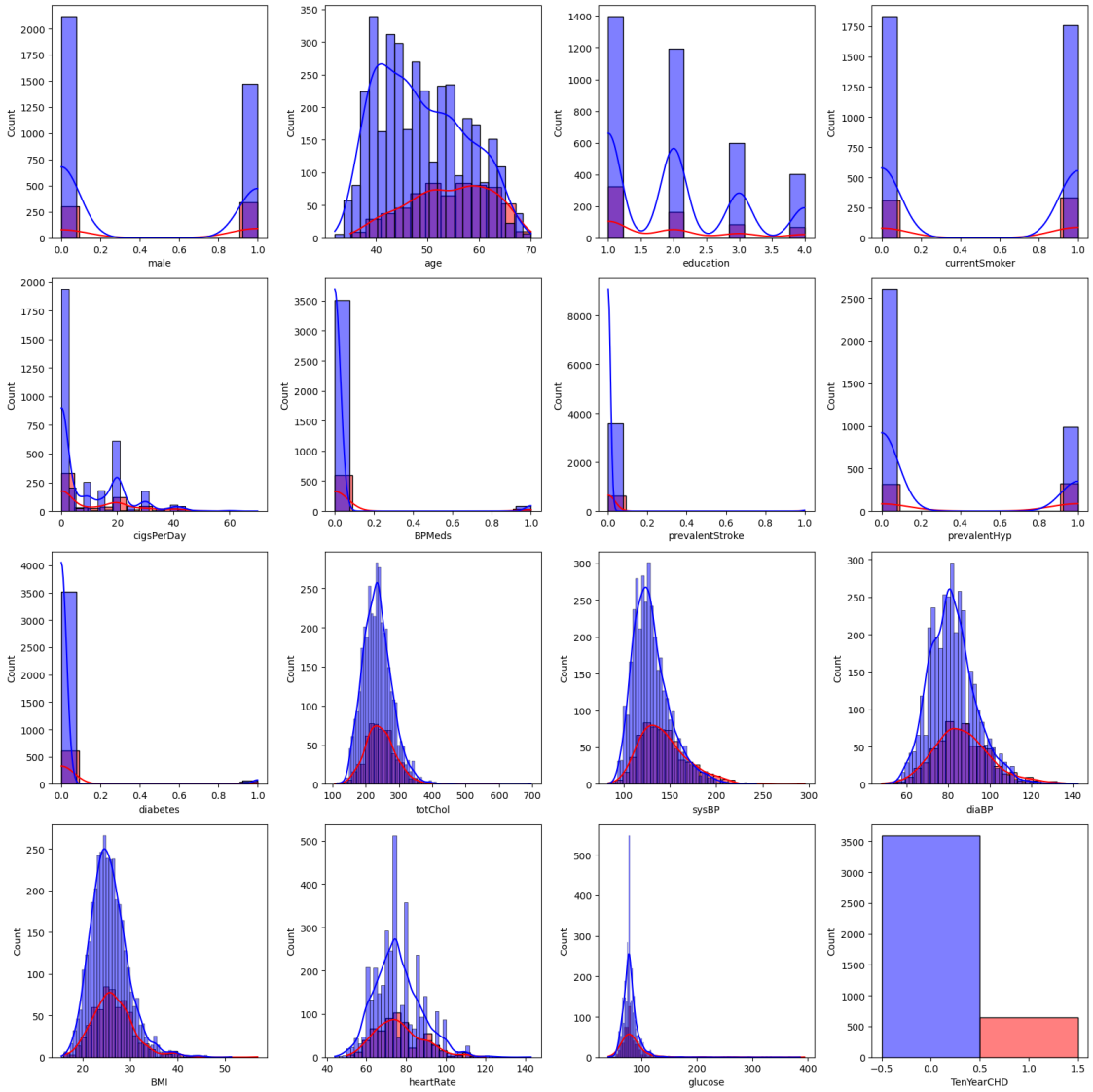
[10]:
```
correlation_matrix = df.corr().apply(lambda x:round(x,2))

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[11]:
```
plt.figure(figsize = (16,16))
for i in range(1,17):
    plt.subplot(4,4,i)
    sns.histplot(df[df['TenYearCHD'] == 1][df.columns.to_list()[i-1]], kde =␣
 ↪True, color = 'red')
    sns.histplot(df[df['TenYearCHD'] == 0][df.columns.to_list()[i-1]], kde =␣
 ↪True, color = 'blue')
```

```
    plt.tight_layout()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\user\AppData\Local\Temp\ipykernel_12068\1053655601.py:6: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
```

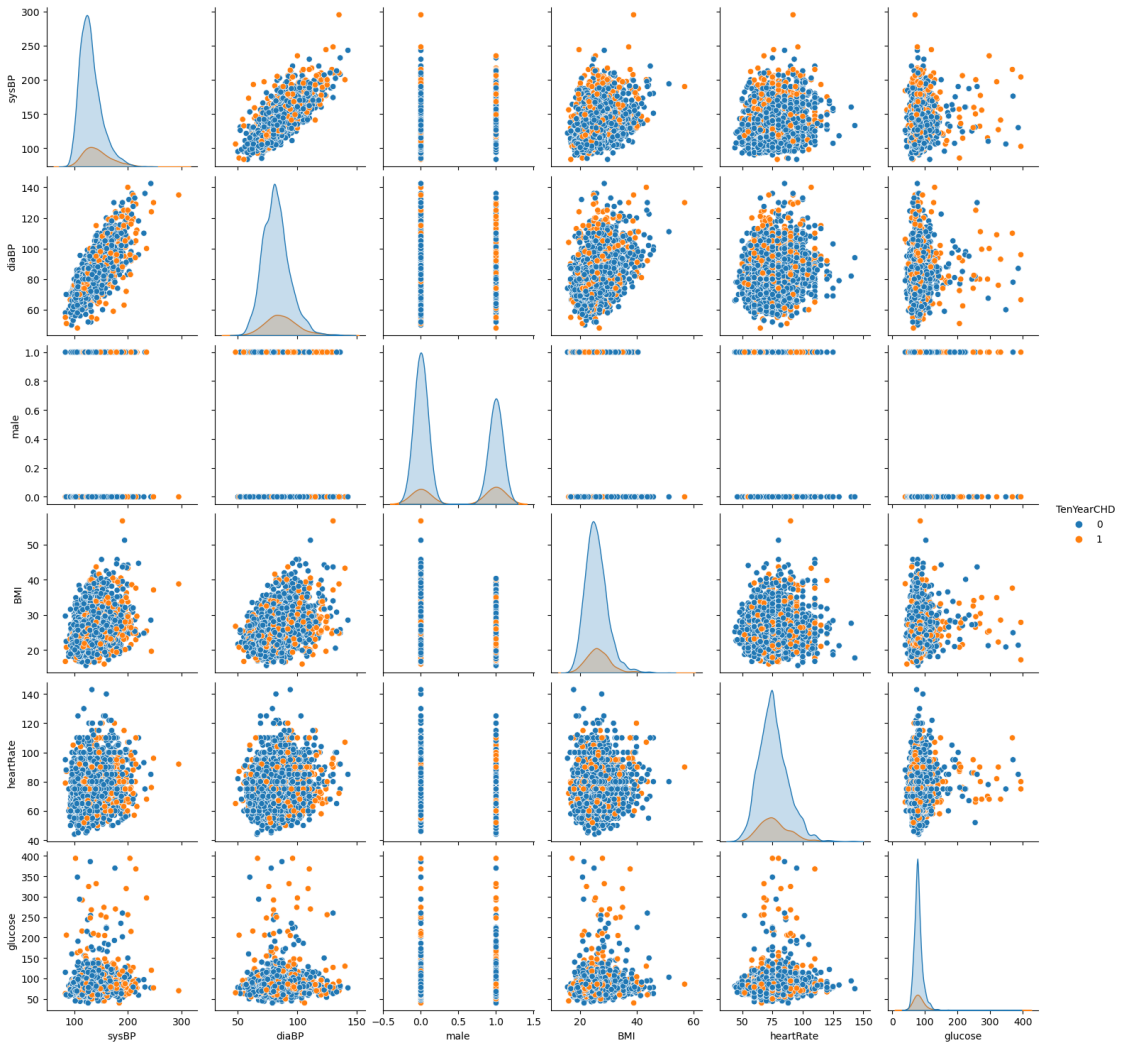


[12]:
```
pair_col= df[['sysBP','diaBP', 'male','BMI', 'heartRate',␣
  ↪'glucose','TenYearCHD']]
plt.figure(figsize=(20,20))
sns.pairplot(pair_col,hue='TenYearCHD')
```

```
C:\Users\user\AppData\Local\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[12]:
```
<seaborn.axisgrid.PairGrid at 0x2e1b297d010>
```

```
<Figure size 2000x2000 with 0 Axes>
```



[13]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report
```

[14]:
```
X = df.iloc[:,:-1]
y = df.iloc[:,-1]

X
```

[14]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	\
0	1	39	4.0	0	0.0	0.0	
1	0	46	2.0	0	0.0	0.0	
2	1	48	1.0	1	20.0	0.0	

```
3          0   61        3.0               1        30.0     0.0
4          0   46        3.0               1        23.0     0.0
…    …   …         …              …         …      …
4233       1   50        1.0               1         1.0     0.0
4234       1   51        3.0               1        43.0     0.0
4235       0   48        2.0               1        20.0     0.0
4236       0   44        1.0               1        15.0     0.0
4237       0   52        2.0               0         0.0     0.0

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
0                   0             0         0    195.0  106.0   70.0  26.97
1                   0             0         0    250.0  121.0   81.0  28.73
2                   0             0         0    245.0  127.5   80.0  25.34
3                   0             1         0    225.0  150.0   95.0  28.58
4                   0             0         0    285.0  130.0   84.0  23.10
…                 …           …       …       …    …     …    …
4233                0             1         0    313.0  179.0   92.0  25.97
4234                0             0         0    207.0  126.5   80.0  19.71
4235                0             0         0    248.0  131.0   72.0  22.00
4236                0             0         0    210.0  126.5   87.0  19.16
4237                0             0         0    269.0  133.5   83.0  21.47

      heartRate  glucose
0          80.0     77.0
1          95.0     76.0
2          75.0     70.0
3          65.0    103.0
4          85.0     85.0
…         …      …
4233       66.0     86.0
4234       65.0     68.0
4235       84.0     86.0
4236       86.0     78.0
4237       80.0    107.0

[4238 rows x 15 columns]
```

[15]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size= 0.3,␣
 ↪random_state=42)

lr = LogisticRegression()
```

[16]:
```
l1 = {'penalty' : ['l1','l2','elasticnet','none'],
     'C': [1,2,4,5, 40,50],
    'max_iter' : [100, 1000,2500, 5000]}
```

[17]:
```
clf = GridSearchCV(lr, param_grid= l1, scoring = 'accuracy', cv = 5)
```

[18]:
```
clf
```

[18]:
```
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [1, 2, 4, 5, 40, 50],
                         'max_iter': [100, 1000, 2500, 5000],
                         'penalty': ['l1', 'l2', 'elasticnet', 'none']},
             scoring='accuracy')
```

[19]:
```
import warnings
warnings.filterwarnings('ignore')
best_clf = clf.fit(X_train, y_train)
```

[20]:
```
print(best_clf.best_params_)
```

```
{'C': 2, 'max_iter': 1000, 'penalty': 'l2'}
```

[21]:
```
print(best_clf.best_score_)
```

```
0.8513147211292236
```

[22]:
```
y_pred = best_clf.predict(X_test)
```

[23]:
```
score = accuracy_score(y_pred, y_test)
score
```

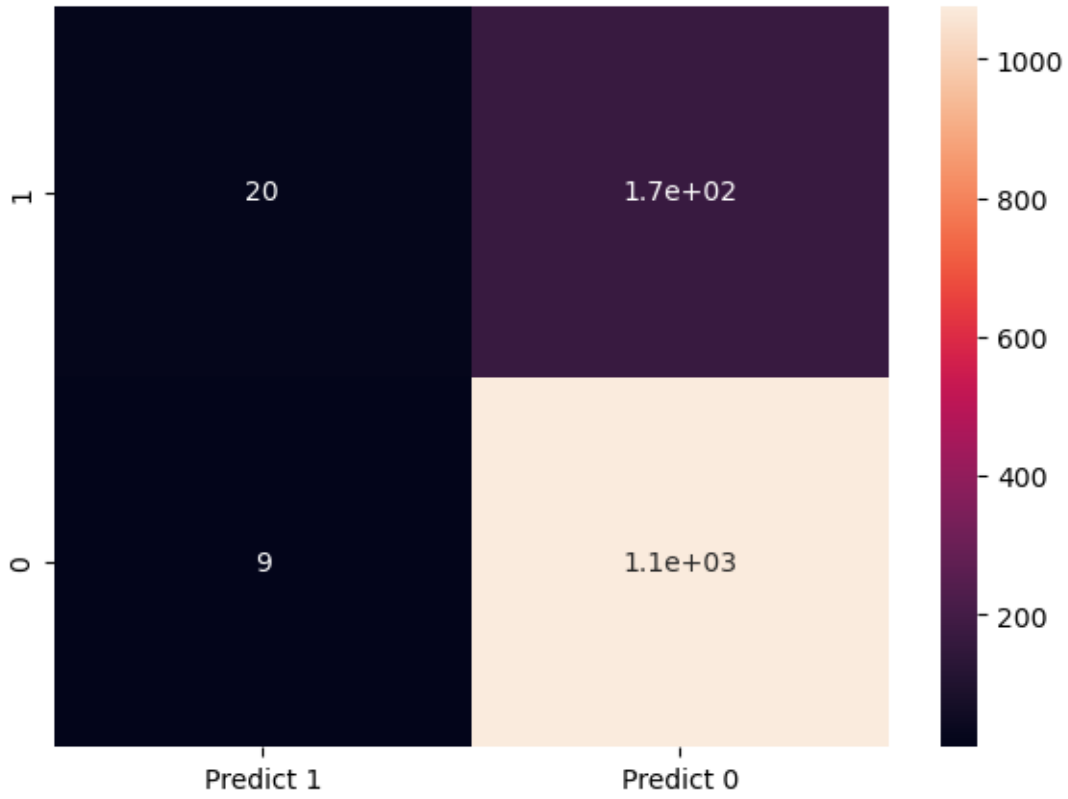
[23]:
```
0.8608490566037735
```

[70]:
```
print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.99      0.86      0.92      1245
           1       0.10      0.70      0.18        27

    accuracy                           0.86      1272
   macro avg       0.55      0.78      0.55      1272
weighted avg       0.97      0.86      0.91      1272
```

[24]:
```
from sklearn import metrics
cm=metrics.confusion_matrix(y_test, y_pred, labels=[1, 0])

df_cm = pd.DataFrame(cm, index = [i for i in ["1","0"]],
                  columns = [i for i in ["Predict 1","Predict 0"]])
plt.figure(figsize = (7,5))
sns.heatmap(df_cm, annot=True)
```

[24]:
```
<Axes: >
```



[25]:
```
cm
```

[25]:
```
array([[  20,  168],
       [   9, 1075]], dtype=int64)
```

[ ]:
```

```