logistic-regression-hd

August 25, 2023

[ ]:

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

##Name:

##Project title:To predict the heart attack disease for orgnization WHO(World Health Organization),using machine learning Algorithm rate of heart_Attack disease will increasing or decreasing manner

1 Problem Statement:A world health organization estimated 12 millions death records.One of them half of the daeath results is found in US.The Research Scholars point out the most relevant risk factor of Heart Attack AS a Data Science Engineer predict the overall risks using machine learning algorithm whichever them called as logistic Regression

##tasks

##import the libraries which is required for prediction ##import the dataset your using workspace ##use the approprite arguement of sklearn librry to train,test,split the dataset ##use the approprite arguement of sklearn librry to train,test,split the dataset ##Chech ur model accuracy and precison using confusion matrix

[ ]:

```python
from google.colab import files
files=files.upload()
```

```
<IPython.core.display.HTML object>

Saving framingham.csv to framingham.csv
```

[ ]:

```python
df=pd.read_csv('framingham.csv')
df
```

[ ]:

```
      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]
```

[ ]:

```
x=df[['age']]
y=df[['currentSmoker']]
```

[ ]:

```

```

[ ]:

```
x_test,x_train,y_test,y_train=train_test_split(x,y,test_size=0.4,random_state=0)
print(x_train)
```

```
      age
```

```
1669   47
156    58
87     61
685    45
666    57
…    …
2790   53
1855   66
700    60
2060   38
2348   48

[1696 rows x 1 columns]
```

```
[ ]: print(y_train)
```

```
      currentSmoker
1669              0
156               0
87                1
685               0
666               0
…               …
2790              0
1855              0
700               0
2060              0
2348              1

[1696 rows x 1 columns]
```

```
[ ]: print(x_test)
```

```
      age
3218   42
590    60
3880   41
1548   59
2601   55
…    …
1033   44
3264   51
1653   39
2607   57
2732   40

[2542 rows x 1 columns]
```

[ ]:

```
print(y_test)
```

```
      currentSmoker
3218              1
590               1
3880              0
1548              0
2601              1
…               …
1033              0
3264              1
1653              1
2607              0
2732              1

[2542 rows x 1 columns]
```

[ ]:

```
from sklearn.preprocessing import StandardScaler
sc =StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

[ ]:

```
print(x_train)
```

```
[[-0.2920309 ]
 [ 0.9841763 ]
 [ 1.33223281]
 …
 [ 1.21621397]
 [-1.33620043]
 [-0.17601207]]
```

[ ]:

```
print(x_test)
```

```
[[-0.87212509]
 [ 1.21621397]
 [-0.98814392]
 …
 [-1.2201816 ]
 [ 0.86815746]
 [-1.10416276]]
```

[ ]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
```

```
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

```
[ ]: LogisticRegression(random_state=0)
```

```
[ ]: y_pred = classifier.predict(x_test)
```

```
[ ]: y_pred
```

```
[ ]: array([1, 0, 1, …, 1, 0, 1])
```

```
[ ]: from sklearn.metrics import confusion_matrix, accuracy_score
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
     accuracy_score(y_test, y_pred)
```

```
[[775 495]
 [517 755]]
```

```
[ ]: 0.6018882769472856
```

##CONCLUSIONS ##ACCORDING TO THE MODEL ANALYSIS THE LOGISTIC REGRESION ALGORITHM WORKS SUCCESSFULLY WITH 0.6 ACCURACY. ##THE ACCURACY SHOWS THAT BUILDING THE MODEL IS SUCCEEFUL

```
[ ]:
```

```
[ ]:
```