NAME:GORLA PRASHANTHREDDY

ROLL NO:21X05A6719

BRANCH:CSE-DATASCIENCE

NRCM

PROJECT TITLE: PREDICT THE HEART ATTACK DISEASE FOR ORGANISATION "WHO" USING MACHINE LEARNING ALGORITHM RATE OF HEART DISEASE WILL BE INCREASING ORDER OR DECREASING MANNER

PROBLEM STATEMENT

A WORLD HEALTH ORGANISATION ESTIMATED 12 MILLIONS DEATH RECORDS.ONE OF THEM HALF OF THERE RESULT IS FOUND IN US. THE REASEARCH SCHOLAR POINT OUT THE MOST RELEVANT RISK FACTOR OF HEART ATTACK,AS A DATA SCIENCE ENGINEER PREDICT THE OVERALL RISK USING MACHINE LEARNING ALGORITHM CALLED AS LOGISTIC REGRESSION

TASK

1.IMPORT THE LIBRARIES,WHICH IS REQUIRED FOR PREDICTION.

2.IMPORT THE DATASET, YOUR USING WORKSPACE.

3.USE THE APPROPRIATE ARGUMENT OF SKLEARN LIBRARY TO TRAIN,TEST AND SPLIT THE DATASET.

4.FIT YOUR VALUES WITH ARRANGE FUNCTION USING FEATURE SCALING.

5.CHECK YOUR MODEL ACCURACY AND PRECISION USING CONFUSION MATRIX.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

ds=pd.read_csv("framingham.csv")
ds
```

```
      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]
```

```python
x=ds[["age"]]
y=ds[["currentSmoker"]]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,rando
m_state=0)

print(x_train)
```

```
      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_train)
```

```
      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]
```

```
print(x_test)
```

```
      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_test)
```

```
      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]
```

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

```python
print(x_train)
```

```
[[-0.89361628]
 [ 1.21446304]
 [-1.0107318 ]
 ...
 [-1.24496283]
 [ 0.86311649]
 [-1.12784731]]
```

```python
print(x_test)
```

```
[[-0.30803869]
 [ 0.980232  ]
 [ 1.33157856]
 ...
 [ 1.21446304]
 [-1.36207835]
 [-0.19092317]]
```

```python
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)

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

```
[[503 371]
 [303 519]]
```

```
0.6025943396226415
```

CONCLUSION

ACCORDING TO THE MODEL ANALYSIS THE LOGISTIC REGRESSION ALGORITHM WORK SUCCESFULLY WITH 0.6 ACCURACY SHOWS THAT THE BUILDING THE MODEL SUCCESSFUL