

heart-attack-disease

August 25, 2023

1 Name: P.Ganesh

2 Rollno: 21X05A6743

3 Branch : DS

4 NRCM

5 project title : predict the heart attack disease for organisation WHO. using machine learning algorithm rate of heart attack disease will increasing manner or decreasing manner

6 Problem Statement:

World Health Organization estimated 12 millions Dead records. One of them half of the dead result is found in US. The research scholar point out the most relevant risk factor of heart attack as a data science engineer predict the overall risk using machine learning algorithm is called logistic regression

7 TASK:

1.import the libraries which is the 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 range function using feature scaling

5.check your model accuracy and precision using confusion matrix

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

```
[3]: dataset=pd.read_csv("framingham.csv")
dataset
```

```
[3]:
```

	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]

```
[19]: x=dataset[["age"]]
y=dataset[["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,random_state=0)
```

```
[20]: 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]
```

```
[21]: 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]
```

```
[22]: 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]

```
[23]: 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]

```
[24]: from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
      x_train = sc.fit_transform(x_train)
      x_test = sc.transform(x_test)
```

```
[25]: print(x_train)
```

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

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

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

```
[27]: 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)
```

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

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

```
[30]: y_pred
```

```
[30]: array([1, 0, 0, ..., 0, 1, 1])
```

```
[31]: 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]]
```

```
[31]: 0.6025943396226415
```

###**conclusion:** According to the model analysis the logisticregression algorithm work successfully with 0.6 accuracy shows that building the model is successful