

# heart-attack3

August 25, 2023

NAME : SUMANTH

ROLL NO : 21X05A6704

BRANCH : CSE(DS)

NRCM

## PROJECT TITLE

Predict the heart attack disease for organization whO using machine learning algorithm rate of heart attack disease will increasing manner or decreasing manner

## ##PROBLEM STATEMENT

A 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

## ##Task

1. Import the libraries which is the required for prediction.
2. Import the data set using work space.
3. Use a appropriate argument of sklearn libraries to train, test, and Split the data sets.
4. Fit your value with arrange function using future scaling.
5. Check your model accuracy and precision using confuser matrix.

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

```
[ ]: data=pd.read_csv("/content/framingham.csv")
data
```

```
[ ]:
   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]

```
[35]: x=data[["age"]]
      y=data[["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.
      ↪2,random_state=0)
```

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

	age
4203	63
3534	52
1736	38

```

541    49
2526   40
...    ...
1033   44
3264   51
1653   39
2607   57
2732   40

```

```
[3390 rows x 1 columns]
```

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

```

      currentSmoker
4203                1
3534                1
1736                1
541                 0
2526                0
...                ...
1033                0
3264                1
1653                1
2607                0
2732                1

```

```
[3390 rows x 1 columns]
```

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

```

      age
1669    47
156     58
87      61
685     45
666     57
...     ...
245     46
4215    63
548     50
4225    45
2996    50

```

```
[848 rows x 1 columns]
```

```
[39]: print(y_test)
```

```

      currentSmoker

```

```

1669          0
156          0
87           1
685          0
666          0
...         ...
245          1
4215         0
548          1
4225         1
2996         0

```

[848 rows x 1 columns]

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

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

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

```
[43]: 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)
```

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

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

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

```

[[268 185]
 [155 240]]

```

```
[47]: 0.5990566037735849
```

###Conclusion

according to the model analysis the logisticregression algorithm work suucessfully with 0.6 accuracy.  
the accuracy shows that building the model is succesful.

[ ]: