

# heart-attack

August 25, 2023

##Name:K.Saivinay ##Roll No:21X05A6731 ##Branch:Data Science Final Year  
##College:Narsimha Reddy Engineering College

#Project Title: ##To Predict the Heart attack disease for organization WHO (World Health Organization), using Machine Learning algorithm rate of heart attack disease will increasing manner or decreasing manner

#Project Statement: ##A world health organization eastimated 12 millions death data.One of them half of the death result is found in US. ##The research scholars point out the most relevent risk factor of heart attack.As a Data Science Engineer predict the overall risk using Machine Learning algorithm which ever them called as Logistic Regression.

#Task: ##1.Import the libraries which is required for prediction ##2.import the dataset your using workspace ##3.Use the apporipriate argument of sklearn library to train ,test and split the dataset ##4.Fit your values with arange function using FeatureScaling ##5.Check your model accuracy and precision using confusion matrix

```
[35]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

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

```
[40]:
```

	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]

```
[38]: data.shape
```

```
[38]: (4238, 16)
```

```
[39]: X=data[["age"]]
y=data[["currentSmoker"]]
X_test,X_train,y_test,y_train = train_test_split(X,y,test_size=0.
↪4,random_state=0)
```

```
[25]: 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]
```

```
[26]: 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]
```

```
[22]: print(X_test)
```

```
[[42]
 [60]
 [41]
 ...
 [39]
 [57]
 [40]]
```

```
[27]: 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]

```
[28]: from sklearn.preprocessing import StandardScaler
      sc = StandardScaler()
      X_train = sc.fit_transform(X_train)
      X_test = sc.transform(X_test)
```

```
[29]: print(X_train)
```

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

```
[30]: print(X_test)
```

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

```
[31]: 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)
```

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

```
[32]: y_pred = classifier.predict(X_test)
```

```
[33]: 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]]
```

```
[33]: 0.6018882769472856
```

###**Conclusion:**According to the model analysis the LogisticRegression algorithm works successfully with 0.60 Accuracy. ###The accuracy shows that building the model is successful.