

Expt. No. 4

Page No. 16

Expt. Name Classification of cardiovascular disease using ML Classifiers Date : _____

Aim:-

To build a classification model with various machine learning classifiers using the Scikit-Learn library in Python.

PROGRAM:-

```
import pandas as pd
```

```
df = pd.read_csv('cardio-sample-dataset.csv')
```

```
df.head()
```

```
df.shape()
```

```
df.isnull().sum()
```

```
df.info()
```


OUTPUT:-

	Age	gender	height(cm)	Weight(kg)	SBP	DBP	Cholestrol	glucose	Smoking
0	48	1	156	56.0	100	60	1	1	0
1	60	1	151	67.0	120	80	2	2	0
2	61	1	157	93.0	130	80	8	1	0
3	48	1	158	71.0	110	70	1	1	0
4	54	1	164	68.0	110	60	1	1	0

(3000,12)

age	0
gender	0
height(cm)	0
weight(kg)	0
SBP	0
DBP	0
Cholestrol	0
glucose	0
Smoking	0
alcohol	0
Physical-active	0
Cardio	0

#	column	Non-Null count	Dtype
0	age	3000 non-null	int64
1	gender	3000 non-null	int64
2	height(cm)	3000 non-null	int64
3	weight(kg)	3000 non-null	float64
4	SBP	3000 non-null	int64
:			
10	Physical-active	3000 non-null	int64
11	Cardio	3000 non-null	int64

df.describe()

x = df.drop(['cardio'], axis = 'columns')
y = df.cardio

print(x)

print(y)

from sklearn.preprocessing import StandardScaler
scaler = ~~Scaler~~.fit_transform(x) StandardScaler()
X_scaled = scaler.fit_transform(x)
X_scaled

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=42)

	Age	gender	Weight(kg)	height(cm)	SBP
count	3000.00	3000.000	3000.0000	3000.0000	3000.000
mean	53.2310	1.500000	75.171600	165.714333	127.09200
std	6.877548	0.500083	14.890697	8.122934	26.97786
min	39.00000	1.000000	40.000000	76.00000	11.00000
25%	48.00000	1.000000	65.000000	160.00000	120.00000
50%	54.00000	1.500000	73.000000	166.00000	120.00000
75%	58.00000	2.000000	84.000000	170.00000	140.00000
max	65.00000	2.000000	200.000000	198.00000	196.00000

	Age	gender	height(cm)	Weight(kg)	SBP	DBP	cholesterol	glucose
0	48	1	156	56.0	100	60	1	1
1	60	1	151	64.0	120	80	2	2
2	61	1	157	93.0	130	80	3	1
:								
2998	52	2	175	94.0	170	110	3	3
2999	54	2	175	97.0	160	100	2	1

3000 rows X 11 columns

```

0      0
1      0
2      0
:      :
2998   1
2999   1

```

Name: Cardio, Length: 3000, dtype: float64

```

array([[ -0.75828873, -1. , -1.19611373, ..., -0.36635434,
        -0.2694026 , -1.99584286],
        ...,
        [ 0.11147468, 1. , 1.14333254, ..., -0.3663534,
        -0.2694026, 0.5010445]])

```


X-train.shape

X-test.shape

```
from sklearn.tree import DecisionTreeClassifier
```

```
DT = DecisionTreeClassifier()
```

```
DTC = DT.fit(X_train, Y_train)
```

```
DTC.score(X_test, Y_test)
```

```
from sklearn.model_selection import cross_val_score  
scores = cross_val_score(DecisionTreeClassifier(), X, Y, cv=5)
```

```
print(scores)
```

```
scores.mean()
```

```
from sklearn import svm
```

```
S = svm.SVC()
```

```
Support = S.fit(X_train, Y_train)
```

```
Support.score(X_test, Y_test)
```

```
scores = cross_val_score(svm.SVC(), X, Y, cv=5)
```

```
print(scores)
```

```
scores.mean()
```

```
from sklearn.naive-bayes import GaussianNB
```

```
GNB = GaussianNB()
```

```
nb = GNB.fit(X_train, Y_train)
```

```
nb.score(X_test, Y_test)
```


(2400, 11)

(600 11)

0.633333333333

[0.67666667 0.67666667 0.64 0.63166667 0.64]

0.647

0.7083333333333334

[0.72166667 0.705 0.69666667 0.69666667 0.715]

0.7070000000000001

0.601666666666667


```
scores = cross_val_score(GaussianNB(), x, y, cv=5)
print (scores)
scores.mean()
```

RESULT:-

Thus we have successfully built a classification model with various machine learning classifiers using the SciKit-learn library in python.

[0.615 0.585 0.57 0.60333333 0.565]
0.5876666666666667