

```
In [1]: import pandas as p
import numpy as n
import seaborn as s
import matplotlib.pyplot as plt
df = p.read_csv("heart.csv")
df
```

```
Out[1]:
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldp
0	1	63	1	typical	145	233	1	2	150	0	
1	2	67	1	asymptomatic	160	286	0	2	108	1	
2	3	67	1	asymptomatic	120	229	0	2	129	1	
3	4	37	1	nonanginal	130	250	0	0	187	0	
4	5	41	0	nontypical	130	204	0	2	172	0	
...	
298	299	45	1	typical	110	264	0	0	132	0	
299	300	68	1	asymptomatic	144	193	1	0	141	0	
300	301	57	1	asymptomatic	130	131	0	0	115	1	
301	302	57	0	nontypical	130	236	0	2	174	0	
302	303	38	1	nonanginal	138	175	0	0	173	0	

303 rows × 15 columns



```
In [2]: df.shape
```

```
Out[2]: (303, 15)
```

```
In [3]: df.isnull().sum()
```

```
Out[3]: Unnamed: 0      0
Age      0
Sex      0
ChestPain  0
RestBP    0
Chol      0
Fbs      0
RestECG   0
MaxHR     0
ExAng     0
Oldpeak   0
Slope     0
Ca        4
Thal      2
AHD       0
dtype: int64
```

```
In [4]: df.dtypes
```

```
Out[4]: Unnamed: 0      int64
Age          int64
Sex          int64
ChestPain    object
RestBP       int64
Chol         int64
Fbs          int64
RestECG      int64
MaxHR        int64
ExAng        int64
Oldpeak      float64
Slope        int64
Ca           float64
Thal         object
AHD          object
dtype: object
```

```
In [5]: df[df==0].count()
```

```
Out[5]: Unnamed: 0      0
Age          0
Sex          97
ChestPain    0
RestBP       0
Chol         0
Fbs          258
RestECG      151
MaxHR        0
ExAng        204
Oldpeak      99
Slope        0
Ca           176
Thal         0
AHD          0
dtype: int64
```

```
In [6]: df["Age"].mean()
```

```
Out[6]: 54.43894389438944
```

```
In [7]: from sklearn.model_selection import train_test_split
```

```
In [8]: train , test = train_test_split(df,random_state=0,test_size=0.25)
train.shape
```

```
Out[8]: (227, 15)
```

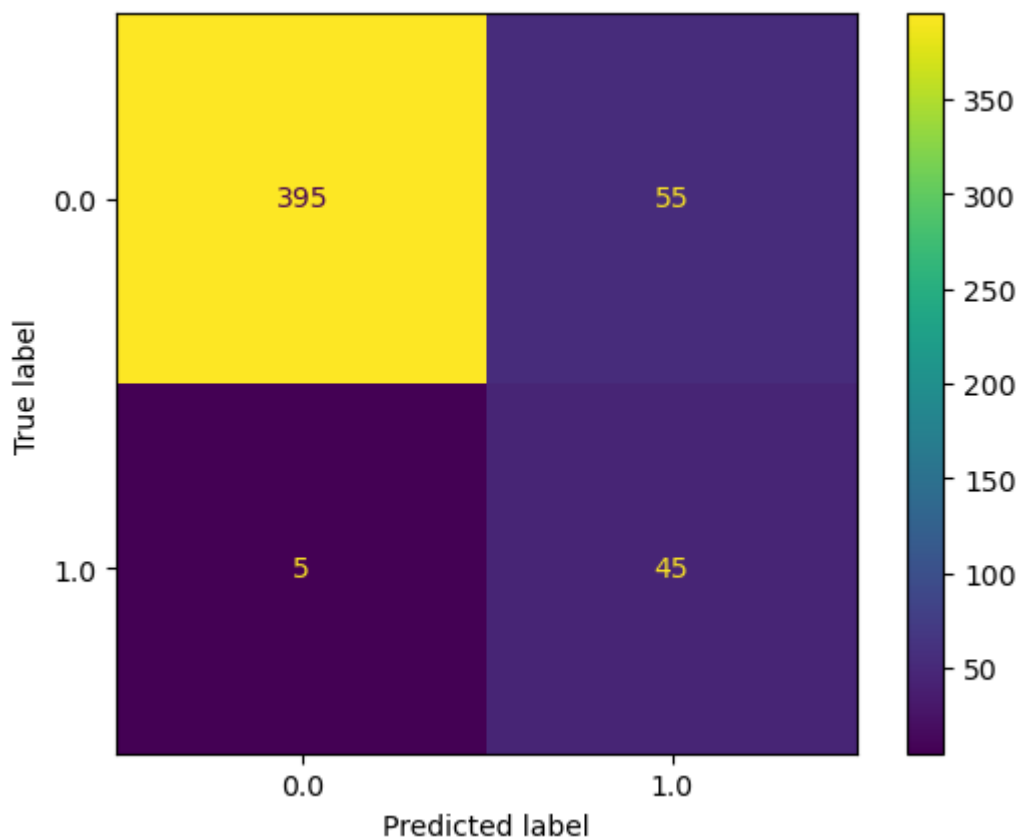
```
In [9]: test.shape
```

```
Out[9]: (76, 15)
```

```
In [22]: from sklearn.metrics import ConfusionMatrixDisplay,accuracy_score,classifica
```

```
In [20]: actual = list(n.ones(45)) + list(n.zeros(450))+list(n.ones(5))
predicted = list(n.ones(100)) + list(n.zeros(400))
ConfusionMatrixDisplay.from_predictions(actual,predicted)
```

```
Out[20]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x2540b8558d0>
```



```
In [21]: print(accuracy_score(actual,predicted))
```

```
0.88
```

```
In [23]: print(classification_report(actual,predicted))
```

	precision	recall	f1-score	support
0.0	0.99	0.88	0.93	450
1.0	0.45	0.90	0.60	50
accuracy			0.88	500
macro avg	0.72	0.89	0.76	500
weighted avg	0.93	0.88	0.90	500

```
In [ ]:
```