

```
In [13]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

```
In [14]: data = pd.read_csv('heart.csv')
data.head()
```

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
In [16]: Y = data['target']
X = data.drop('target',axis = 1)
print(X.head())
print(Y.head())
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
0	63	1	3	145	233	1	0	150	0	2.3
1	37	1	2	130	250	0	1	187	0	3.5
2	41	0	1	130	204	0	0	172	0	1.4
3	56	1	1	120	236	0	1	178	0	0.8
4	57	0	0	120	354	0	1	163	1	0.6

	ca	thal
0	0	1
1	0	2
2	0	2
3	0	2
4	0	2

	slope
0	1
1	1
2	1
3	1
4	1

Name: target, dtype: int64

```
In [18]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2
, random_state = 42)
#Use standard scaler to scale the features for preprocessing
scaler = StandardScaler()
scale = scaler.fit(X_train)
X_train = scale.transform(X_train)
X_test = scale.transform(X_test)
```

```
In [19]: # creating logistic Regresion model
model = LogisticRegression()
model.fit(X_train, Y_train)
pred = model.predict(X_test)
```

```
In [20]: score = accuracy_score(Y_test,pred)
score
```

```
Out[20]: 0.8524590163934426
```

```
In [21]: #creating a confusion Matrix\n",
confusion_matrix(Y_test,pred)
```

```
Out[21]: array([[25,  4],
               [ 5, 27]])
```

```
In [22]: #extracting TrueNegative(TN), TruePositive(TP),FalseNegative (FN),FalseP
ositive(FP)
tn, fp, fn, tp = confusion_matrix(Y_test, pred).ravel()
(tn, fp, fn, tp)
```

```
Out[22]: (25, 4, 5, 27)
```

```
In [23]: #confusion Matrix metrics,
matrix = classification_report(Y_test,pred)
print('Classification report: \n', matrix)
```

```
Classification report:
              precision    recall  f1-score   support

         0       0.83      0.86      0.85         29
         1       0.87      0.84      0.86         32

 accuracy          0.85
 macro avg          0.85
weighted avg          0.85
```

```
In [ ]:
```