

```
In [1]: #Experiment no.8
```

```
In [ ]: #Aim:To perform and analysis of Logistic Regression Algorithm
```

```
In [ ]: #Name:Sakshi Rambhau Wankhade
        #Roll No.:72
        #Sec:A
        #Subject:ET-1
        #Date:30-09-2025
```

# To Perform and Analysis of Logistic Regression Algorithm

## Importing the Libraries

```
In [2]: import pandas as pd
        import numpy as np
```

## Data acquisitionuing Pandas

```
In [3]: import os
```

```
In [4]: os.getcwd()
```

```
Out[4]: 'C:\\Users\\ADMIN\\DSS_practical'
```

```
In [5]: os.chdir('C:\\Users\\ADMIN\\DSS_practical')
```

```
In [6]: data=pd.read_csv("heart.csv")
```

```
In [7]: data.head()
```


```
Out[7]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	ti
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	

In [8]: `data.tail()`

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
<b>1020</b>	59	1	1	140	221	0	1	164	1	0.0	2	0	2
<b>1021</b>	60	1	0	125	258	0	0	141	1	2.8	1	1	3
<b>1022</b>	47	1	0	110	275	0	0	118	1	1.0	1	1	2
<b>1023</b>	50	0	0	110	254	0	0	159	0	0.0	2	0	2
<b>1024</b>	54	1	0	120	188	0	1	113	0	1.4	1	1	3



In [9]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   age         1025 non-null   int64
 1   sex         1025 non-null   int64
 2   cp          1025 non-null   int64
 3   trestbps    1025 non-null   int64
 4   chol        1025 non-null   int64
 5   fbs         1025 non-null   int64
 6   restecg     1025 non-null   int64
 7   thalach     1025 non-null   int64
 8   exang       1025 non-null   int64
 9   oldpeak     1025 non-null   float64
10   slope       1025 non-null   int64
11   ca          1025 non-null   int64
12   thal        1025 non-null   int64
13   target      1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [10]: `data.describe()`

Out[10]:

	age	sex	cp	trestbps	chol	fbs	restecg
<b>count</b>	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
<b>mean</b>	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.500000
<b>std</b>	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.500000
<b>min</b>	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
<b>25%</b>	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
<b>50%</b>	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
<b>75%</b>	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000
<b>max</b>	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

In [11]: `data.shape`

Out[11]: (1025, 14)

In [12]: `data.size`

Out[12]: 14350

In [13]: `data.ndim`

Out[13]: 2

## Data preprocessing \_ data cleaning \_ missing value treatment

In [14]: `# check Missing Value by record`  
`data.isna()`

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	False	False	False	False	False	False	False	False	False	False	False	False
1021	False	False	False	False	False	False	False	False	False	False	False	False
1022	False	False	False	False	False	False	False	False	False	False	False	False
1023	False	False	False	False	False	False	False	False	False	False	False	False
1024	False	False	False	False	False	False	False	False	False	False	False	False

1025 rows × 14 columns

In [15]: `data.isna().any()`

```
Out[15]: age      False
sex      False
cp       False
trestbps False
chol     False
fbs      False
restecg  False
thalach  False
exang    False
oldpeak  False
slope    False
ca       False
thal     False
target   False
dtype: bool
```

In [16]: `data.isna().sum()`

```
Out[16]: age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

## Independent and Dependent Variables

```
In [24]: x=data.drop("target", axis=1)
y=data["target"]
```

## Splitting of DataSet into train and Test

```
In [25]: #splitting the data into training and testing data sets
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=42)
```

## Logistic Regression

```
In [26]: import warnings
warnings.filterwarnings("ignore")
```

```
In [27]: from sklearn.linear_model import LogisticRegression
```

```
In [28]: log = LogisticRegression()
log.fit(x_train, y_train)
```

```
Out[28]: ▼ LogisticRegression ⓘ ?
LogisticRegression()
```

```
In [29]: y_pred1 = log.predict(x_test)
```

```
In [30]: from sklearn.metrics import accuracy_score
```

```
In [31]: accuracy_score (y_test,y_pred1)
```

Out[31]: 0.7902439024390244

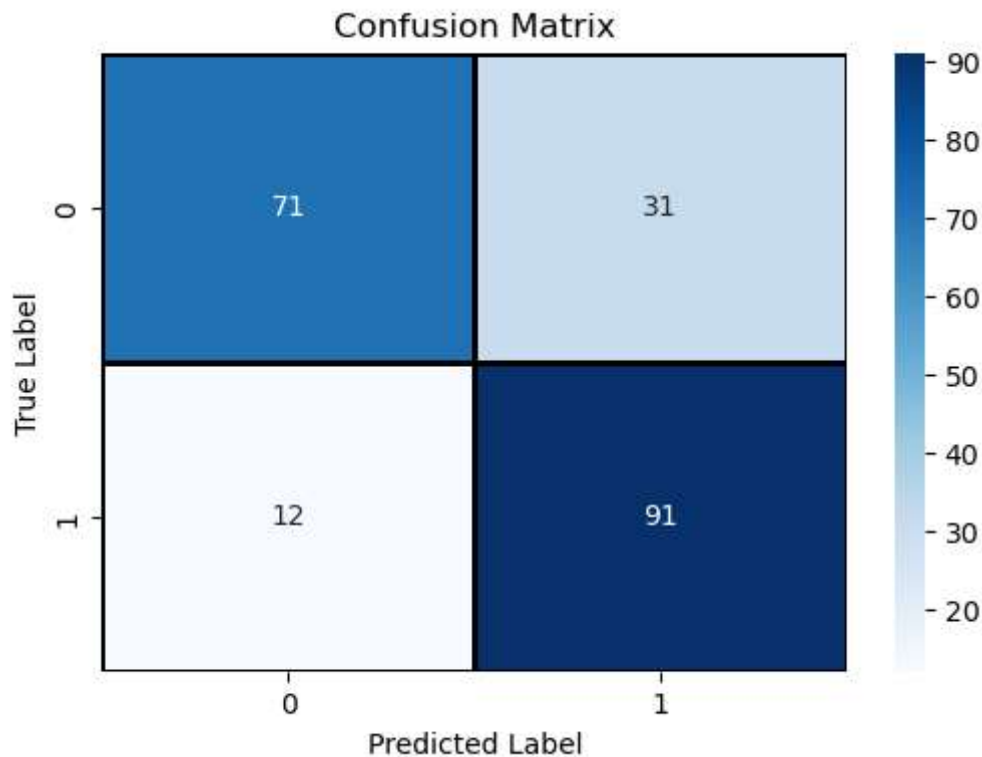
```
In [32]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
```

```
In [33]: cm = confusion_matrix(y_test,y_pred1)
```

```
In [35]: labels = np.unique(y_test) # Get unique class labels
cm_df = pd.DataFrame(cm, index=labels, columns=labels)
```

```
In [36]: # Plot confusion matrix using seaborn
plt.figure(figsize=(6, 4))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', linewidths=1, linecolor='black')

plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```



In [ ]: