Assignment No. 5

Aim :

1. Logistic Regression
2. Differentiate between Linear and Logistic Regression
3. Sigmoid Function
4. Types of LogisticRegression
5. Confusion Matrix Evaluation Metrics

Code:

In [43]:
```
import numpy as np
import pandas as pd
import matplotlib as plt
```

In [44]:
```
data = pd.read_csv("diabetes.csv")
data
```

Out[44]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 9 columns

In [45]:
```
data.head()
```

Out[45]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	

In [46]:
```
data.tail()
```

Out[46]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

In [47]:
```
print("The shape of the data is: ")
data.shape
```

```
The shape of the data is:
```

Out[47]:
```
(768, 9)
```

In [48]:
```
print(data.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [49]:
```
X = data.iloc[:,0:13]
y = data.iloc[:,-1]
X
y
```

Out[49]:
```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

In [50]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,random_s
```

In [51]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(614, 9)
(154, 9)
(614,)
(154,)
```

In [68]:
```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

X, y = make_classification(random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
pipe = make_pipeline(StandardScaler(), LogisticRegression())
pipe.fit(X_train, y_train)
```

Out[68]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('logisticregression', LogisticRegression())])
```

In [53]:
```
from sklearn.linear_model import LinearRegression
```

In [54]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
model.fit(X_train, y_train)
```

Out[54]:
```
Pipeline(steps=[('standardscaler', StandardScaler(with_mean=False)),
                ('linearregression', LinearRegression())])
```

In [55]:
```
model.score(X_test,y_test)
```

Out[55]:
```
1.0
```

In [56]:
```
X_test
```

Out[56]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
668	6	98	58	33	190	34.0	0.430
324	2	112	75	32	0	35.7	0.148
624	2	108	64	0	0	30.8	0.158
690	8	107	80	0	0	24.6	0.856
473	7	136	90	0	0	29.9	0.210
...	...	...	...	...	...	...	...
355	9	165	88	0	0	30.4	0.302
534	1	77	56	30	56	33.3	1.251
344	8	95	72	0	0	36.8	0.485
296	2	146	70	38	360	28.0	0.337
462	8	74	70	40	49	35.3	0.705

154 rows × 9 columns

In [57]:
```
y_test
```

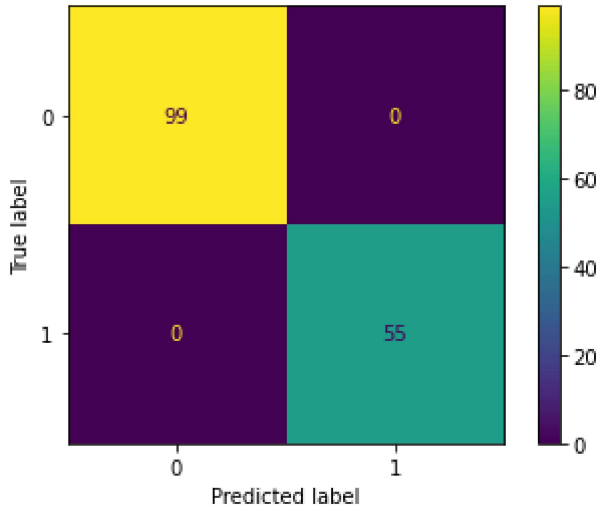
Out[57]:
```
668    0
324    0
624    0
690    0
473    0
      ..
355    1
534    0
344    0
296    1
462    0
Name: Outcome, Length: 154, dtype: int64
```

In [63]:
```
from sklearn.metrics import precision_score,ConfusionMatrixDisplayconfusion_matri
cm= confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix = cm)
print("Confusion matrix :")
print(cm)
```

```
Confusion matrix :
[[99  0]
 [ 0 55]]
```

In [65]:
```
disp.plot()
```

Out[65]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x29b493eabb0
>
```



Name : Mohan Kadambande

Roll No. : 13212 (TECO-b1)