In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df=pd.read_csv("diabetes.csv")
df
```

Out[2]:

	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 [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [4]:
```
x=df.iloc[:,0:-1]
x
```

Out[4]:

	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 × 8 columns

In [5]:
```
y=df.Outcome
y
```

Out[5]:
```
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 [6]:
```
def sigmoid(x):
    import math
    y=1/(1+math.exp(-x))
    return y
```

In [7]:
```
sigmoid(2.5)
```

Out[7]:
```
0.9241418199787566
```

In [8]:
```
import math
```

In [9]:
```
print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh',
'asin', 'asinh', 'atan', 'atan2', 'atanh', 'cbrt', 'ceil', 'comb', 'copysign', 'co
s', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'exp2', 'expm1', 'fabs',
'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'is
close', 'isfinite', 'isinf', 'isnan', 'isqrt', 'lcm', 'ldexp', 'lgamma', 'log', 'l
og10', 'log1p', 'log2', 'modf', 'nan', 'nextafter', 'perm', 'pi', 'pow', 'prod',
'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc', 'ul
p']
```

In [10]:
```
# train_test_split

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)
```

In [11]:
```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(614, 8)
(614,)
(154, 8)
(154,)
```

In [12]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
```

Out[12]:
```
▾ LogisticRegression

LogisticRegression()
```

In [13]:
```
model.score(x_test,y_test)
```

Out[13]:
```
0.7467532467532467
```

In [14]:
```
y_predicted=model.predict(x_test)
y_predicted
```

Out[14]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      dtype=int64)
```

In [15]:
```
model.coef_
```

Out[15]:
```
array([[ 0.0579772 ,  0.03406027, -0.01410581,  0.00428063, -0.00187732,
         0.09908123,  0.61205411,  0.03731179]])
```

In [16]:
```
model.intercept_
```

Out[16]:
```
array([-8.85497785])
```

In [17]:
```
x_test
```

Out[17]:

	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 × 8 columns

In [18]:
```
x_test["actual"]=y_test
x_test
```

Out[18]:

	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 [19]:
```
x_test["predicted"]=y_predicted
x_test
```

Out[19]:

	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 × 10 columns

In [20]:
```
k=0.06576265*2+0.03383761*146-0.01357809*70+0.00418336*38-0.00180533*360+0.10095649
k
```

Out[20]:
```
-1.203754904490002
```

In [21]:
```
def sigmoid(k):
    import math
    y=1/(1+math.exp(-k))
    return y
sigmoid(k)
```

Out[21]:
```
0.23080791348266091
```

In [22]:
```
if sigmoid(k)<0.5:
    print("the person has no diabetes")
else:
    print("person have diabetes")
```

```
the person has no diabetes
```

In [ ]: