In [ ]:

```
                ASSIGNMENT NO:5
```

In [ ]:

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

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib as plt
import warnings
import seaborn as sns
warnings.filterwarnings("ignore")
```

In [5]:

```
data=pd.read_csv("C:\\Users\\shrey\\Downloads\\diabetes.csv")
```

In [7]:

```
data
```

Out[7]:

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

768 rows × 9 columns

In [9]:

```
from sklearn.model_selection import train_test_split
```

In [13]:

```
X = data.drop('Outcome', axis=1)
Y = data['Outcome']
```

In [15]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_sta
```

In [17]:

```
print(f"Training data shape (X_train): {X_train.shape}")
print(f"Testing data shape (X_test): {X_test.shape}")
print(f"Training data shape (Y_train): {Y_train.shape}")
print(f"Testing data shape (Y_test): {Y_test.shape}")
```

```
Training data shape (X_train): (614, 8)
Testing data shape (X_test): (154, 8)
Training data shape (Y_train): (614,)
Testing data shape (Y_test): (154,)
```

In [19]:

```
 from sklearn.linear_model import LogisticRegression
 logreg = LogisticRegression()
```

In [21]:

```
logreg.fit(X_train,Y_train)
```

Out[21]:

```
▾  LogisticRegression ⓘ ?

LogisticRegression()
```

In [25]:

```
y_testpred=logreg.predict(X_test)
y_trainpred = logreg.predict(X_train)
```

In [27]:

```
 from sklearn.metrics import precision_score, confusion_matrix, accuracy_score, rec
```

In [31]:

```
train_accuracy = accuracy_score(Y_train, y_trainpred)
train_precision = precision_score(Y_train, y_trainpred)
train_recall = recall_score(Y_train, y_trainpred)
train_cm = confusion_matrix(Y_train, y_trainpred)
test_accuracy = accuracy_score(Y_test, y_testpred)
test_precision = precision_score(Y_test, y_testpred)
test_recall = recall_score(Y_test, y_testpred)
test_cm = confusion_matrix(Y_test, y_testpred)
print("Training Accuracy: ", train_accuracy)
print("Training Precision: ", train_precision)
print("Training Recall: ", train_recall)
print("Training Confusion Matrix:\n", train_cm)
print("\nTesting Accuracy: ", test_accuracy)
print("Testing Precision: ", test_precision)
print("Testing Recall: ", test_recall)
print("Testing Confusion Matrix:\n", test_cm)
```

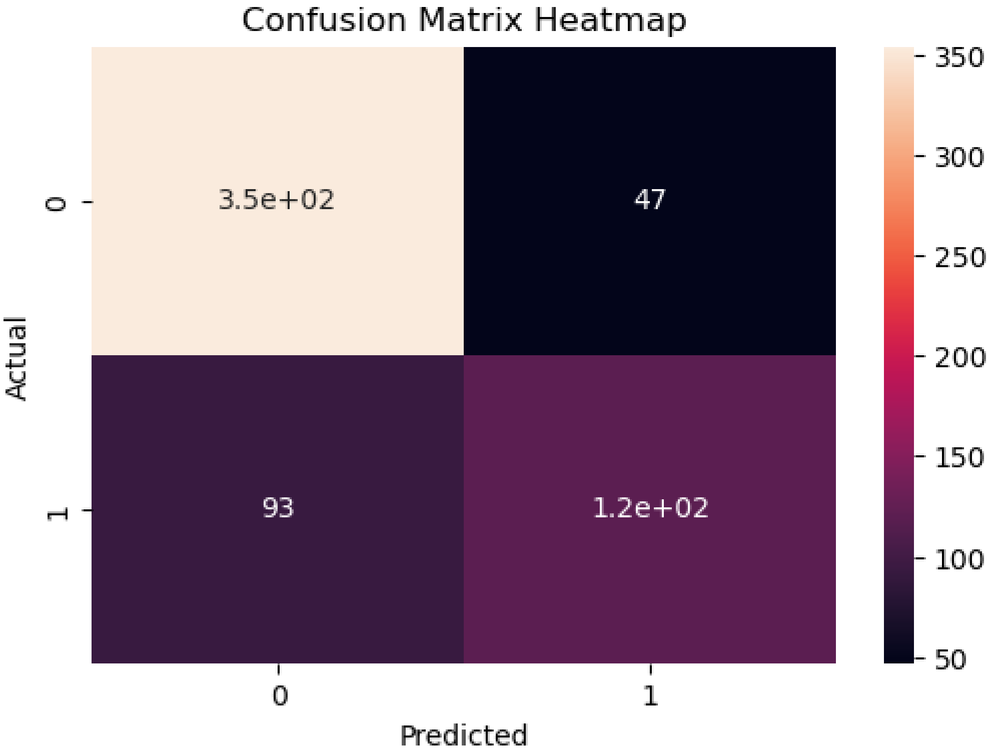
```
Training Accuracy:  0.7719869706840391
Training Precision:  0.718562874251497
Training Recall:  0.5633802816901409
Training Confusion Matrix:
 [[354  47]
 [ 93 120]]

Testing Accuracy:  0.7467532467532467
Testing Precision:  0.6379310344827587
Testing Recall:  0.6727272727272727
Testing Confusion Matrix:
 [[78 21]
 [18 37]]
```

In [33]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [35]:

```
plt.figure(figsize=(6,4))
sns.heatmap(train_cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.show()
```



In [ ]:

```
NAME:Karan More
ROLLNO:13234
```