In [17]:

```
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 confusion_matrix, classification_report
```

In [18]:

```
dataset = pd.read_csv('../Data/Social_Network_Ads.csv')
print(dataset.head())
```

```
   Age  EstimatedSalary  Purchased
0   19            19000          0
1   35            20000          0
2   26            43000          0
3   27            57000          0
4   19            76000          0
```

In [19]:

```
X = dataset[['Age', 'EstimatedSalary']].values
y = dataset['Purchased'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random
```

In [20]:

```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [21]:

```
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

Out[21]:

```
LogisticRegression(random_state=0)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [22]:

```
y_pred = classifier.predict(X_test)
```

In [23]:

```
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Confusion Matrix:
 [[65  3]
 [ 8 24]]
```

In [24]:

```
cl_report = classification_report(y_test, y_pred)
print("Classification Report:\n", cl_report)
```

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

           0       0.89      0.96      0.92        68
           1       0.89      0.75      0.81        32

    accuracy                           0.89       100
   macro avg       0.89      0.85      0.87       100
weighted avg       0.89      0.89      0.89       100
```