In [19]:
```python
# Logistic Regression on SUV dataset
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
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, accuracy_score
```

In [20]:
```python
data = pd.read_csv(r"C:\Users\Admin\Downloads\archive\suv_data.csv")
data.head()
data.shape
data['Age'].unique()
```

Out[20]:
```
array([19, 35, 26, 27, 32, 25, 20, 18, 29, 47, 45, 46, 48, 49, 31, 21, 28,
       33, 30, 23, 24, 22, 59, 34, 39, 38, 37, 42, 40, 36, 41, 58, 55, 52,
       60, 56, 53, 50, 51, 57, 44, 43, 54])
```

In [21]:
```python
data.shape
```

Out[21]:
```
(400, 5)
```

In [22]:
```python
data['Age'].unique()
```

Out[22]:
```
array([19, 35, 26, 27, 32, 25, 20, 18, 29, 47, 45, 46, 48, 49, 31, 21, 28,
       33, 30, 23, 24, 22, 59, 34, 39, 38, 37, 42, 40, 36, 41, 58, 55, 52,
       60, 56, 53, 50, 51, 57, 44, 43, 54])
```

In [23]:
```python
for col in data.columns:
    print(f"Unique values in '{col}': {data[col].unique()}")
```

```
Unique values in 'User ID': [15624510 15810944 15668575 15603246 15804002 15728773 15598044 15
694829
 15600575 15727311 15570769 15606274 15746139 15704987 15628972 15697686
 15733883 15617482 15704583 15621083 15649487 15736760 15714658 15599081
 15705113 15631159 15792818 15633531 15744529 15669656 15581198 15729054
 15573452 15776733 15724858 15713144 15690188 15689425 15671766 15782806
 15764419 15591915 15772798 15792008 15715541 15639277 15798850 15776348
 15727696 15793813 15694395 15764195 15744919 15671655 15654901 15649136
 15775562 15807481 15642885 15789109 15814004 15673619 15595135 15583681
 15605000 15718071 15679760 15654574 15577178 15595324 15756932 15726358
 15595228 15782530 15592877 15651983 15746737 15774179 15667265 15655123
 15595917 15668385 15709476 15711218 15798659 15663939 15694946 15631912
 15768816 15682268 15684801 15636428 15809823 15699284 15786993 15709441
 15710257 15582492 15575694 15756820 15766289 15593014 15584545 15675949
 15672091 15801658 15706185 15789863 15720943 15697997 15665416 15660200
 15619653 15773447 15739160 15689237 15679297 15591433 15642725 15701962
 15811613 15741049 15724423 15574305 15678168 15697020 15610801 15745232
 15722758 15792102 15675185 15801247 15725660 15638963 15800061 15578006
 15668504 15687491 15610403 15741094 15807909 15666141 15617134 15783029
 15622833 15746422 15750839 15749130 15779862 15767871 15679651 15576219
 15699247 15619087 15605327 15610140 15791174 15602373 15762605 15598840
 15744279 15670619 15599533 15757837 15697574 15578738 15762228 15614827
 15789815 15579781 15587013 15570932 15794661 15581654 15644296 15614420
 15609653 15594577 15584114 15673367 15685576 15774727 15694288 15603319
 15759066 15814816 15724402 15571059 15674206 15715160 15730448 15662067
 15779581 15662901 15689751 15667742 15738448 15680243 15745083 15708228
 15628523 15708196 15735549 15809347 15660866 15766609 15654230 15794566
 15800890 15697424 15724536 15735878 15707596 15657163 15622478 15779529
 15636023 15582066 15666675 15732987 15789432 15663161 15694879 15593715
 15575002 15622171 15795224 15685346 15691808 15721007 15794253 15694453
 15813113 15614187 15619407 15646227 15660541 15753874 15617877 15772073
 15701537 15736228 15780572 15769596 15586996 15722061 15638003 15775590
 15730688 15753102 15810075 15723373 15795298 15584320 15724161 15750056
 15609637 15794493 15569641 15815236 15811177 15680587 15672821 15767681
 15600379 15801336 15721592 15581282 15746203 15583137 15680752 15688172
 15791373 15589449 15692819 15727467 15734312 15764604 15613014 15759684
 15609669 15685536 15750447 15663249 15638646 15734161 15631070 15761950
 15649668 15713912 15586757 15596522 15625395 15760570 15566689 15725794
 15673539 15705298 15675791 15747043 15736397 15678201 15720745 15637593
 15598070 15787550 15603942 15733973 15596761 15652400 15717893 15622585
 15733964 15753861 15747097 15594762 15667417 15684861 15742204 15623502
 15774872 15611191 15674331 15619465 15575247 15695679 15713463 15785170
 15796351 15639576 15693264 15589715 15769902 15587177 15814553 15601550
 15664907 15612465 15810800 15665760 15588080 15776844 15717560 15629739
 15729908 15716781 15646936 15768151 15579212 15721835 15800515 15591279
 15587419 15750335 15699619 15606472 15778368 15671387 15573926 15709183
 15577514 15778830 15768072 15768293 15654456 15807525 15574372 15671249
 15779744 15624755 15611430 15774744 15629885 15708791 15793890 15646091
 15596984 15800215 15577806 15749381 15683758 15670615 15715622 15707634
 15806901 15775335 15724150 15627220 15672330 15668521 15807837 15592570
 15748589 15635893 15757632 15691863 15706071 15654296 15755018 15594041]
Unique values in 'Gender': ['Male' 'Female']
Unique values in 'Age': [19 35 26 27 32 25 20 18 29 47 45 46 48 49 31 21 28 33 30 23 24 22 59
34
 39 38 37 42 40 36 41 58 55 52 60 56 53 50 51 57 44 43 54]
Unique values in 'EstimatedSalary': [ 19000  20000  43000  57000  76000  58000  84000 150000
33000  65000
  80000  52000  86000  18000  82000  25000  26000  28000  29000  22000
  49000  41000  23000  30000  74000 137000  16000  44000  90000  27000
  72000  31000  17000  51000 108000  15000  79000  54000 135000  89000
  32000  83000  55000  48000 117000  87000  66000 120000  63000  68000
 113000 112000  42000  88000  62000 118000  85000  81000  50000 116000
 123000  73000  37000  59000 149000  21000  35000  71000  61000  75000
```

```
  53000 107000  96000  45000  47000 100000  38000  69000 148000 115000
  34000  60000  70000  36000  39000 134000 101000 130000 114000 142000
  78000 143000  91000 144000 102000 126000 133000 147000 104000 146000
 122000  97000  95000 131000  77000 125000 106000 141000  93000 138000
 119000 105000  99000 129000  46000  64000 139000]
Unique values in 'Purchased': [0 1]
```

In [24]:
```
data.isnull().sum()
```

Out[24]:
```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [25]:
```
data.duplicated().sum()
```

Out[25]:
```
np.int64(0)
```

In [26]:
```
data.describe()
```

Out[26]:

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

In [27]:
```
# ---- STEP 2: Select Features and Target ----
X = data[['Age', 'EstimatedSalary']]  # Features
y = data['Purchased']                   # Target (0 = No, 1 = Yes)
```

In [28]:
```
# ---- STEP 3: Split Dataset ----
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)
```

In [29]:
```
# ---- STEP 4: Feature Scaling ----
# -3 to +3
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [30]:
```
X_train_scaled
```

```
Out[30]: array([[ 1.8925893 ,  1.52189404],
                [ 0.1250379 ,  0.03213212],
                [ 0.9106163 , -1.31157471],
                [-1.34792161, -1.48684082],
                [-0.169554  , -0.58129926],
                [-0.56234321,  2.33980255],
                [ 1.0088136 , -1.19473064],
                [-0.75873781,  1.08372877],
                [ 2.1871812 , -1.04867555],
                [ 0.0268406 , -0.25997806],
                [-0.46414591, -1.1363086 ],
                [ 0.1250379 ,  0.03213212],
                [ 1.6961947 , -0.90262046],
                [ 1.1070109 , -0.90262046],
                [ 0.5178271 ,  1.22978386],
                [-1.05332971, -1.4576298 ],
                [-1.15152701, -1.54526286],
                [-0.0713567 ,  0.67477452],
                [ 0.4196298 , -0.46445519],
                [-0.2677513 , -0.25997806],
                [-0.85693511,  0.14897619],
                [ 0.0268406 ,  0.29503128],
                [ 0.7142217 , -1.28236369],
                [ 1.5979974 ,  1.11293979],
                [ 0.812419  , -1.36999675],
                [-1.44611891, -1.22394166],
                [-0.0713567 ,  0.14897619],
                [ 0.4196298 , -0.14313399],
                [-0.2677513 ,  0.03213212],
                [ 1.3034055 ,  2.22295848],
                [ 0.1250379 ,  0.76240757],
                [-1.34792161,  0.55793045],
                [ 1.9907866 ,  0.73319655],
                [-1.24972431, -1.39920777],
                [ 0.3214325 , -0.3184001 ],
                [-0.95513241,  0.55793045],
                [ 0.4196298 ,  0.29503128],
                [ 0.4196298 ,  1.11293979],
                [ 0.812419  ,  0.76240757],
                [ 0.9106163 ,  1.25899488],
                [-0.46414591, -1.22394166],
                [-1.83890811, -1.31157471],
                [ 1.1070109 ,  0.55793045],
                [-0.66054051, -1.60368489],
                [-0.75873781,  0.26582026],
                [ 1.0088136 ,  2.07690339],
                [-0.56234321,  1.37583895],
                [-0.0713567 ,  0.03213212],
                [-1.93710541,  0.47029739],
                [ 0.4196298 ,  0.26582026],
                [-1.05332971,  0.41187535],
                [ 0.2232352 , -0.14313399],
                [ 1.8925893 ,  0.11976517],
                [-1.15152701, -1.60368489],
                [-1.15152701,  0.29503128],
                [-0.85693511, -0.78577639],
                [-0.46414591,  2.31059153],
                [ 0.1250379 , -0.8149874 ],
                [ 1.5979974 ,  0.99609572],
                [-0.169554  , -1.07788657],
                [ 0.812419  , -1.10709759],
                [ 0.2232352 ,  2.1061144 ],
                [-0.0713567 , -0.23076704],
```

```
       [-0.85693511,  2.28138051],
       [-0.0713567 , -0.37682213],
       [-0.2677513 , -0.58129926],
       [ 0.4196298 , -0.49366621],
       [-0.2677513 , -0.93183148],
       [ 0.3214325 , -1.16551962],
       [ 0.2232352 ,  0.06134314],
       [-1.15152701, -1.60368489],
       [-0.66054051, -0.05550093],
       [-0.2677513 , -0.49366621],
       [-0.2677513 , -1.31157471],
       [-0.75873781,  0.55793045],
       [ 0.3214325 ,  0.06134314],
       [-0.95513241,  1.55110506],
       [ 0.812419  ,  0.35345332],
       [-1.54431621, -0.20155602],
       [ 0.7142217 , -1.39920777],
       [-0.75873781, -0.61051028],
       [-0.36594861, -1.31157471],
       [ 0.2232352 ,  0.14897619],
       [-0.56234321,  1.37583895],
       [-1.44611891,  0.35345332],
       [-1.15152701,  0.29503128],
       [ 1.0088136 ,  1.7847932 ],
       [ 2.0889839 ,  2.13532542],
       [-0.2677513 , -0.43524417],
       [-0.36594861, -0.78577639],
       [ 0.1250379 , -0.25997806],
       [-1.05332971,  0.76240757],
       [ 2.1871812 ,  0.38266434],
       [-1.34792161, -0.43524417],
       [ 1.9907866 ,  2.16453644],
       [ 1.4998001 ,  0.99609572],
       [-0.2677513 ,  0.26582026],
       [-0.169554  ,  0.85004063],
       [ 1.8925893 , -0.28918908],
       [-0.0713567 ,  1.96005931],
       [-0.46414591, -0.78577639],
       [ 0.3214325 ,  0.03213212],
       [ 1.4016028 , -1.42841878],
       [ 1.4016028 ,  2.33980255],
       [-0.0713567 ,  0.0029211 ],
       [-1.15152701,  0.41187535],
       [-1.15152701,  0.06134314],
       [-1.15152701, -0.52287722],
       [ 0.3214325 , -0.28918908],
       [-0.66054051, -0.11392297],
       [-0.0713567 ,  2.16453644],
       [ 0.0268406 , -0.25997806],
       [-0.66054051, -1.04867555],
       [ 0.4196298 ,  0.14897619],
       [ 0.812419  ,  1.37583895],
       [-0.169554  , -0.52287722],
       [ 0.0268406 ,  0.03213212],
       [ 1.1070109 ,  0.52871943],
       [ 1.8925893 , -1.07788657],
       [ 1.0088136 ,  1.98927033],
       [ 0.9106163 , -0.58129926],
       [-0.46414591, -0.02628992],
       [-0.0713567 ,  2.22295848],
       [-1.74071081,  0.35345332],
       [ 0.2232352 , -0.66893231],
       [-0.2677513 , -1.39920777],
```

```
[-1.74071081, -0.99025351],
[ 0.7142217 , -0.72735435],
[-1.15152701, -0.78577639],
[ 1.9907866 ,  0.90846266],
[ 0.2232352 , -0.37682213],
[ 0.0268406 ,  1.22978386],
[-0.0713567 ,  0.29503128],
[ 1.1070109 , -1.22394166],
[-0.169554  ,  0.14897619],
[ 0.3214325 ,  0.06134314],
[-0.2677513 , -0.28918908],
[ 0.5178271 ,  1.72637117],
[ 0.3214325 ,  0.49950841],
[ 0.0268406 ,  1.25899488],
[ 1.9907866 , -1.36999675],
[-1.15152701, -1.10709759],
[-0.56234321, -1.51605184],
[ 0.3214325 , -0.52287722],
[-0.66054051, -1.51605184],
[-0.2677513 ,  0.52871943],
[ 1.0088136 , -1.01946453],
[-1.05332971,  0.55793045],
[-0.2677513 ,  0.79161859],
[ 0.4196298 ,  0.09055416],
[-1.64251351, -0.05550093],
[ 0.3214325 ,  0.06134314],
[-0.46414591, -0.28918908],
[ 0.1250379 ,  1.87242626],
[-0.95513241,  0.41187535],
[ 0.812419  ,  0.52871943],
[-1.34792161, -0.34761112],
[-0.66054051,  1.40504997],
[ 1.2052082 ,  0.52871943],
[-1.74071081,  0.35345332],
[-0.56234321,  1.90163728],
[-1.74071081,  0.47029739],
[-0.2677513 , -1.25315268],
[-0.85693511, -0.78577639],
[-1.64251351,  0.52871943],
[-0.2677513 , -0.3184001 ],
[ 0.7142217 , -1.10709759],
[ 2.1871812 , -0.8149874 ],
[-0.2677513 ,  0.61635248],
[-0.2677513 , -0.75656537],
[-1.93710541, -0.75656537],
[ 0.4196298 ,  2.31059153],
[ 0.7142217 ,  0.26582026],
[ 0.2232352 , -0.28918908],
[-1.34792161, -1.36999675],
[-0.46414591, -0.55208824],
[ 0.4196298 ,  0.0029211 ],
[ 1.6961947 ,  1.75558219],
[ 1.2052082 , -0.75656537],
[ 1.0088136 ,  1.43426099],
[-1.83890811,  0.17818721],
[ 0.3214325 ,  0.06134314],
[ 0.2232352 ,  0.03213212],
[-1.05332971,  0.52871943],
[-0.169554  ,  1.63873811],
[-0.2677513 ,  0.14897619],
[-0.2677513 , -0.34761112],
[-0.36594861,  1.31741692],
[-0.2677513 ,  0.09055416],
```

```
[ 2.1871812 ,  1.11293979],
[-1.24972431,  0.49950841],
[-1.05332971, -0.46445519],
[-1.64251351, -1.57447387],
[-0.0713567 ,  0.11976517],
[-0.2677513 , -0.90262046],
[ 1.5979974 ,  0.0029211 ],
[ 0.9106163 , -1.16551962],
[ 0.0268406 , -0.58129926],
[-0.2677513 ,  2.25216949],
[-0.2677513 ,  0.20739823],
[ 0.3214325 ,  0.26582026],
[-0.0713567 , -1.07788657],
[ 2.1871812 , -0.69814333],
[-0.95513241, -0.3184001 ],
[-1.44611891, -0.11392297],
[-1.44611891, -0.20155602],
[-0.75873781, -1.54526286],
[-1.24972431, -1.07788657],
[ 2.0889839 ,  0.38266434],
[ 1.9907866 , -0.93183148],
[-1.93710541,  0.35345332],
[ 0.812419  , -0.3184001 ],
[-1.05332971, -1.54526286],
[ 1.9907866 , -0.66893231],
[ 1.0088136 , -1.16551962],
[ 1.1070109 , -0.14313399],
[ 1.2052082 , -0.99025351],
[ 1.4998001 ,  0.06134314],
[ 0.2232352 , -0.37682213],
[ 1.4016028 ,  1.2882059 ],
[ 2.0889839 , -0.8149874 ],
[-0.169554  , -0.20155602],
[ 0.4196298 ,  0.99609572],
[-0.0713567 , -0.52287722],
[ 1.0088136 , -1.07788657],
[ 2.0889839 , -1.19473064],
[-0.0713567 ,  0.26582026],
[ 0.2232352 , -0.25997806],
[ 1.1070109 ,  0.11976517],
[-1.24972431,  0.58714146],
[-0.75873781, -1.60368489],
[ 0.2232352 ,  0.23660925],
[-1.34792161,  0.41187535],
[ 0.0268406 , -0.58129926],
[ 0.7142217 ,  1.7847932 ],
[-1.64251351,  0.06134314],
[-0.2677513 , -1.36999675],
[-0.2677513 , -1.4576298 ],
[-0.66054051,  0.55793045],
[-0.75873781,  0.29503128],
[ 0.9106163 , -0.66893231],
[-1.05332971,  0.58714146],
[-0.2677513 ,  0.06134314],
[-0.75873781,  1.34662793],
[ 0.1250379 ,  1.52189404],
[-0.85693511,  0.38266434],
[ 0.3214325 , -0.20155602],
[ 1.0088136 ,  0.58714146],
[ 0.0268406 , -0.3184001 ],
[-0.56234321, -1.51605184],
[ 0.1250379 ,  0.14897619],
[-1.15152701,  0.3242423 ],
```

```
       [ 0.1250379 ,  1.05451775],
       [-1.54431621, -0.43524417],
       [-0.169554  ,  1.40504997],
       [ 2.1871812 , -0.8149874 ],
       [-0.95513241, -0.43524417],
       [ 1.3034055 ,  1.87242626],
       [ 1.1070109 , -1.22394166],
       [-0.169554  , -0.28918908],
       [ 1.794392  ,  0.99609572],
       [-1.05332971, -0.34761112],
       [-1.34792161, -1.10709759],
       [-0.36594861,  0.06134314],
       [-0.95513241, -1.10709759],
       [ 1.2052082 , -1.4576298 ],
       [-0.46414591, -0.84419842],
       [-0.85693511, -0.66893231],
       [-1.54431621, -1.51605184],
       [-0.75873781,  1.90163728],
       [ 0.9106163 ,  1.02530673],
       [ 0.812419  ,  0.26582026],
       [ 1.0088136 ,  1.87242626],
       [ 0.9106163 , -0.61051028],
       [ 1.1070109 ,  2.07690339],
       [-0.56234321,  0.87925164],
       [ 0.0268406 ,  0.03213212],
       [-1.83890811, -1.28236369],
       [-0.0713567 ,  0.20739823],
       [ 0.9106163 , -0.55208824],
       [ 0.2232352 , -0.37682213],
       [-0.169554  ,  1.6095271 ],
       [-1.74071081,  0.11976517],
       [-0.66054051, -0.34761112],
       [ 0.3214325 , -0.72735435],
       [ 0.4196298 , -0.46445519],
       [-0.95513241, -0.96104249],
       [ 0.1250379 ,  0.09055416],
       [-0.95513241,  0.44108637],
       [ 0.0268406 , -0.55208824],
       [ 0.9106163 , -0.78577639],
       [-0.0713567 ,  0.06134314],
       [ 1.1070109 , -0.99025351],
       [ 0.7142217 , -1.39920777],
       [-0.2677513 ,  0.06134314],
       [-1.34792161, -1.25315268],
       [-1.15152701, -1.01946453],
       [ 0.5178271 ,  1.84321524],
       [ 0.1250379 ,  0.20739823],
       [-0.56234321,  0.47029739]])
```

In [31]:
```
# ---- STEP 5: Train Logistic Regression Model ----
model = LogisticRegression()
model.fit(X_train_scaled, y_train)
```

Out[31]:
```
▾ LogisticRegression
LogisticRegression()
```

In [32]:
```
# ---- STEP 6: Predictions ----
y_pred = model.predict(X_test_scaled)
```

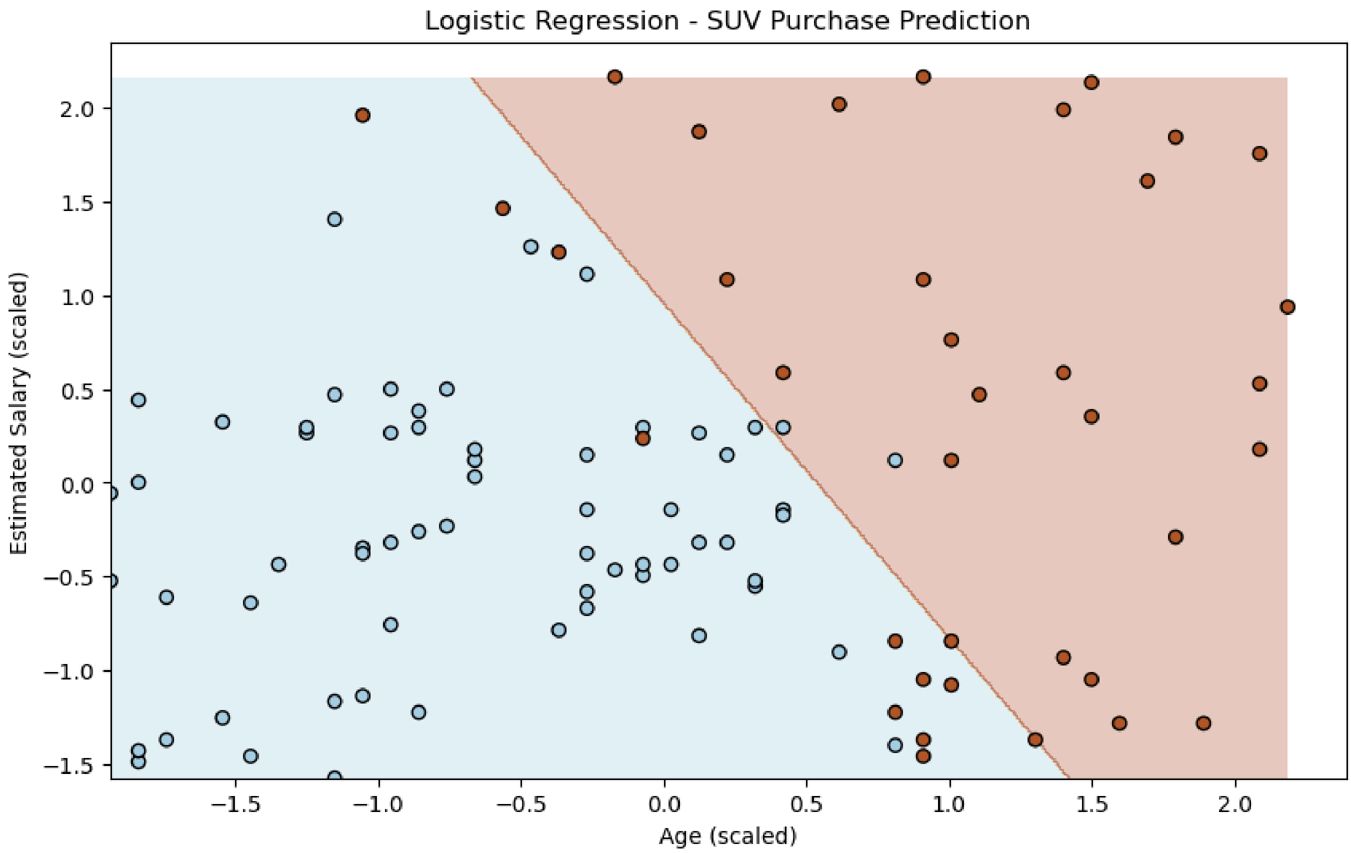
In [33]:
```
# ---- STEP 7: Evaluation ----
cm = confusion_matrix(y_test, y_pred)
```

```python
print("Confusion Matrix:\n", cm)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
accuracy = accuracy_score(y_test, y_pred)
print(f"\nAccuracy: {accuracy:.2f}")
```

```
Confusion Matrix:
 [[61  2]
 [12 25]]

Classification Report:
               precision    recall  f1-score   support

           0       0.84      0.97      0.90        63
           1       0.93      0.68      0.78        37

    accuracy                           0.86       100
   macro avg       0.88      0.82      0.84       100
weighted avg       0.87      0.86      0.85       100


Accuracy: 0.86
```

In [34]:

```python
import matplotlib.pyplot as plt
import numpy as np

# Use test set features
X_set, y_set = X_test_scaled, y_test

# Create mesh grid directly from feature values
X1, X2 = np.meshgrid(
    np.arange(start=X_set[:, 0].min(), stop=X_set[:, 0].max(), step=0.01),
    np.arange(start=X_set[:, 1].min(), stop=X_set[:, 1].max(), step=0.01)
)

# Predict on the grid
Z = model.predict(np.array([X1.ravel(), X2.ravel()]).T)
Z = Z.reshape(X1.shape)

# Plot decision boundary
plt.figure(figsize=(10, 6))
plt.contourf(X1, X2, Z, alpha=0.3, cmap=plt.cm.Paired)

# Plot test points
plt.scatter(X_set[:, 0], X_set[:, 1], c=y_set, cmap=plt.cm.Paired, edgecolors='k')
plt.title("Logistic Regression - SUV Purchase Prediction")
plt.xlabel("Age (scaled)")
plt.ylabel("Estimated Salary (scaled)")
plt.show()
```



In [35]:
```
test_data = pd.DataFrame({
    "Age": [25, 40, 50],
    "EstimatedSalary": [30000, 70000, 90000]
})
```

In [36]:
```
# Scale and predict
test_scaled = scaler.transform(test_data)
predictions = model.predict(test_scaled)
probs = model.predict_proba(test_scaled)
```

In [37]:
```
print("\n--- Test Data Predictions ---")
for i, row in test_data.iterrows():
    print(f"Age: {row['Age']}, Salary: {row['EstimatedSalary']} -> "
          f"{'Buys SUV' if predictions[i] else 'No Purchase'} "
          f"(Prob: {probs[i][1]*100:.1f}%)")
```

```
--- Test Data Predictions ---
Age: 25, Salary: 30000 -> No Purchase (Prob: 0.9%)
Age: 40, Salary: 70000 -> No Purchase (Prob: 35.5%)
Age: 50, Salary: 90000 -> Buys SUV (Prob: 87.2%)
```

In [38]:
```
# Simple plot with legend + probability annotations (on top)
plt.figure(figsize=(8, 5))

# Decision boundary
plt.contourf(
    X1, X2,
    model.predict(np.c_[X1.ravel(), X2.ravel()]).reshape(X1.shape),
    alpha=0.3, cmap=plt.cm.Paired
)

# Original test set points
plt.scatter(X_set[:, 0], X_set[:, 1], c=y_set, cmap=plt.cm.Paired, alpha=0.5, label="Test set

# Test points (new data)
plt.scatter(
```

```
    test_scaled[:, 0], test_scaled[:, 1], c=predictions,
    cmap=plt.cm.coolwarm, marker='X', s=200, edgecolors='black', label="New test data"
)



plt.title("Logistic Regression - Test Data with Probabilities")
plt.xlabel("Age (scaled)")
plt.ylabel("Salary (scaled)")
plt.legend()
plt.show()
```