In [50]:
```
import pandas as pd
```

In [51]:
```
df = pd.read_csv("Social_Network_Ads.csv")
```

In [52]:
```
df
```

Out[52]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [53]:
```
x=df[['Age','EstimatedSalary']]
```

In [54]:
```
y=df['Purchased']
```

In [55]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)
```

In [56]:
```
from sklearn.model_selection import train_test_split
```

In [117…
```
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,random_state=0, te
```

In [118…
```
x_train
```

Out[118…
```
array([[0.61904762, 0.17777778],
       [0.33333333, 0.77777778],
       [0.47619048, 0.25925926],
       [0.33333333, 0.88888889],
       [0.80952381, 0.04444444],
       [0.83333333, 0.65925926],
       [0.5       , 0.2       ],
       [0.47619048, 0.34074074],
       [0.42857143, 0.25925926],
       [0.42857143, 0.35555556],
       [0.4047619 , 0.07407407],
       [0.4047619 , 0.25925926],
       [0.57142857, 0.42962963],
       [0.69047619, 0.25185185],
       [0.97619048, 0.1037037 ],
       [0.73809524, 0.37037037],
       [0.64285714, 0.85925926],
       [0.30952381, 0.54814815],
       [0.66666667, 0.4962963 ],
       [0.69047619, 0.26666667],
       [0.19047619, 0.        ],
       [1.        , 0.64444444],
       [0.47619048, 0.71851852],
       [0.52380952, 0.68148148],
       [0.57142857, 0.28148148],
       [0.4047619 , 0.32592593],
       [0.71428571, 0.19259259],
       [0.71428571, 0.88148148],
       [0.47619048, 0.72592593],
       [0.26190476, 0.98518519],
       [0.19047619, 0.        ],
       [1.        , 0.2       ],
       [0.14285714, 0.02962963],
       [0.57142857, 0.99259259],
       [0.66666667, 0.6       ],
       [0.23809524, 0.32592593],
       [0.5       , 0.6       ],
       [0.23809524, 0.54814815],
       [0.54761905, 0.42222222],
       [0.64285714, 0.08148148],
       [0.35714286, 0.4       ],
       [0.04761905, 0.4962963 ],
       [0.30952381, 0.43703704],
       [0.57142857, 0.48148148],
       [0.4047619 , 0.42222222],
       [0.35714286, 0.99259259],
       [0.52380952, 0.41481481],
       [0.78571429, 0.97037037],
       [0.66666667, 0.47407407],
       [0.4047619 , 0.44444444],
       [0.47619048, 0.26666667],
       [0.42857143, 0.44444444],
       [0.45238095, 0.46666667],
       [0.47619048, 0.34074074],
       [1.        , 0.68888889],
       [0.04761905, 0.4962963 ],
```

```
       [0.73809524, 0.17777778],
       [0.21428571, 0.11851852],
       [0.02380952, 0.40740741],
       [0.5       , 0.47407407],
       [0.19047619, 0.48888889],
       [0.16666667, 0.48148148],
       [0.23809524, 0.51851852],
       [0.88095238, 0.17777778],
       [0.76190476, 0.54074074],
       [0.73809524, 0.54074074],
       [0.80952381, 1.        ],
       [0.4047619 , 0.37037037],
       [0.57142857, 0.28888889],
       [0.38095238, 0.20740741],
       [0.45238095, 0.27407407],
       [0.71428571, 0.11111111],
       [0.26190476, 0.20740741],
       [0.42857143, 0.27407407],
       [0.21428571, 0.28888889],
       [0.19047619, 0.76296296]])
```

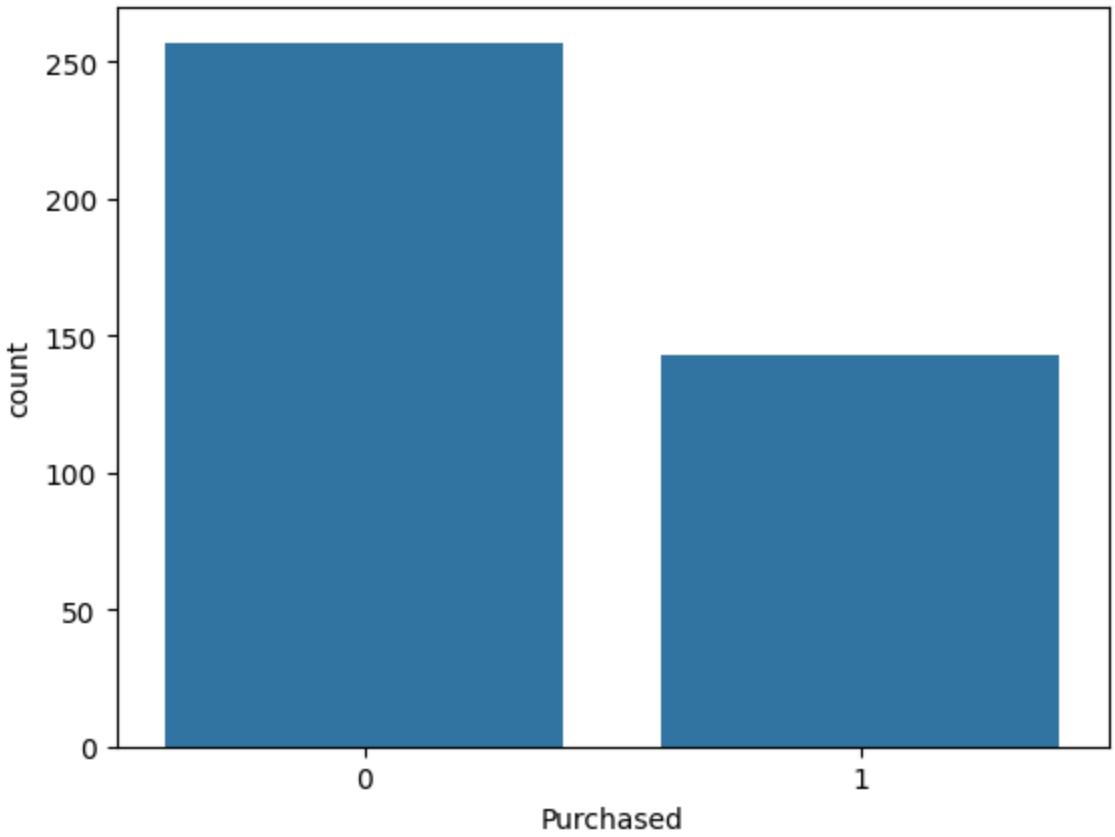
In [119…
```
y_train
```

Out[119…
```
250    0
63     1
312    0
159    1
283    1
      ..
323    1
192    0
117    0
47     0
172    0
Name: Purchased, Length: 300, dtype: int64
```

In [120…
```
from sklearn.linear_model import LogisticRegression
```

In [121…
```
import seaborn as sns
sns.countplot(x=y)
```

Out[121…
```
<Axes: xlabel='Purchased', ylabel='count'>
```



In [122…
```
y.value_counts()
```

Out[122…
```
Purchased
0    257
1    143
Name: count, dtype: int64
```

In [123…
```
classifier = LogisticRegression()
```

In [124…
```
classifier.fit(x_train,y_train)
```

Out[124…
```
▾  LogisticRegression i ?

LogisticRegression()
```

In [125…
```
y_pred = classifier.predict(x_test)
```

In [126…
```
y_train.shape
```

Out[126…
```
(300,)
```

In [127…
```
x_train.shape
```

Out[127…
```
(300, 2)
```

In [128…
```
y_pred
```

Out[128…
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

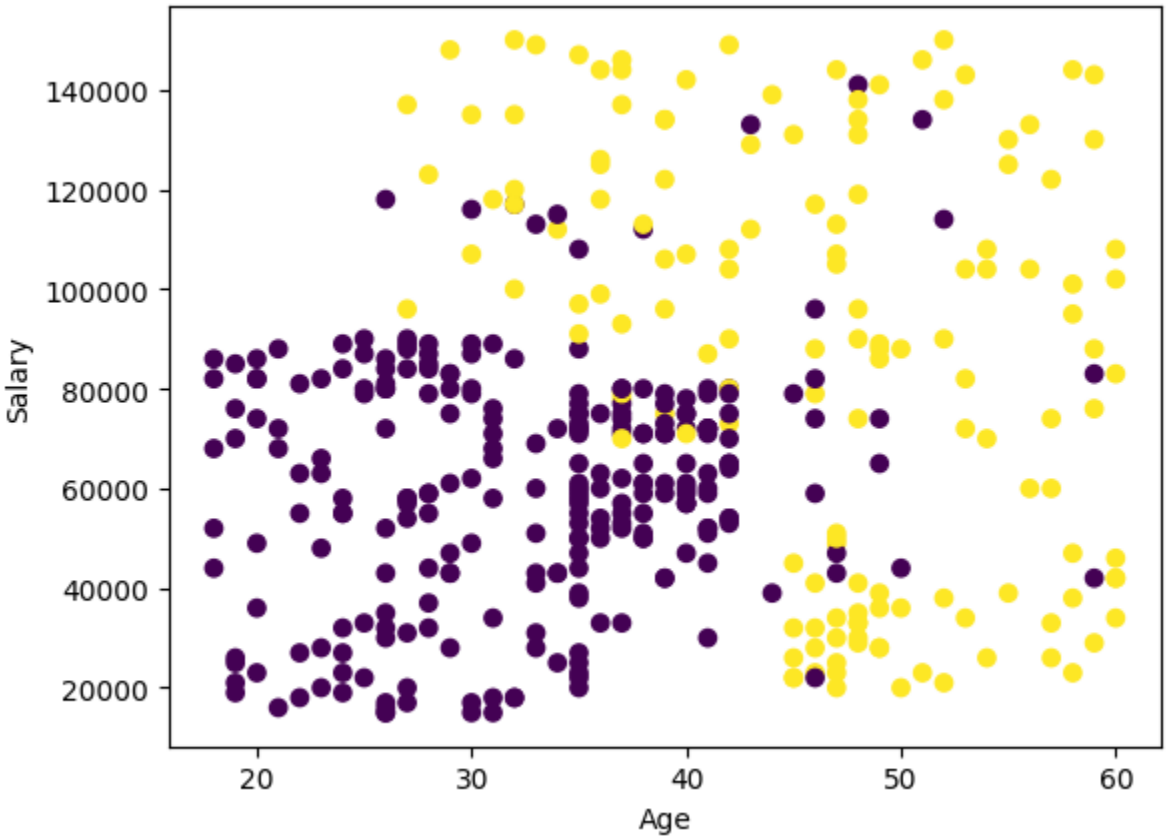
In [129…
```
y_test
```

Out[129…
```
132    0
309    0
341    0
196    0
246    0
      ..
146    1
135    0
390    1
264    1
364    1
Name: Purchased, Length: 100, dtype: int64
```

In [130…
```
import matplotlib.pyplot as plt
```

In [131…
```
plt.xlabel('Age')
plt.ylabel('Salary')
plt.scatter(x['Age'],x['EstimatedSalary'],c=y)
```

Out[131…
```
<matplotlib.collections.PathCollection at 0x7fe9d4927fd0>
```



In [132…
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)
```

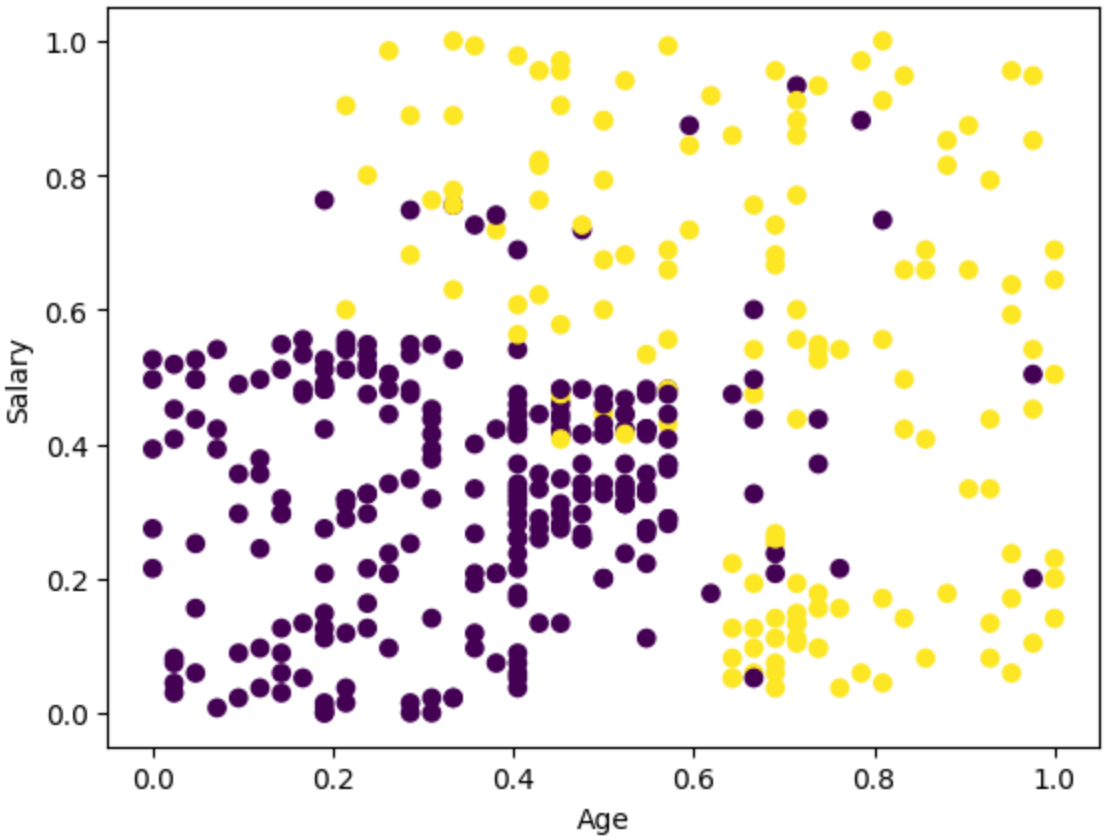
In [133…
```
pd.DataFrame(x_scaled).describe()
```

Out[133…

	0	1
count	400.000000	400.000000
mean	0.467976	0.405500
std	0.249592	0.252570
min	0.000000	0.000000
25%	0.279762	0.207407
50%	0.452381	0.407407
75%	0.666667	0.540741
max	1.000000	1.000000

In [134…
```
plt.xlabel('Age')
plt.ylabel('Salary')
plt.scatter(x_scaled[:,0],x_scaled[:,1],c=y)
```

Out[134… `<matplotlib.collections.PathCollection at 0x7fe9d4999e50>`



In [135…
```
from sklearn.metrics import confusion_matrix
```

In [136…
```
confusion_matrix(y_test,y_pred)
```

Out[136…
```
array([[67,  1],
       [10, 22]])
```

In [137…
```
y_pred
```

Out[137…
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

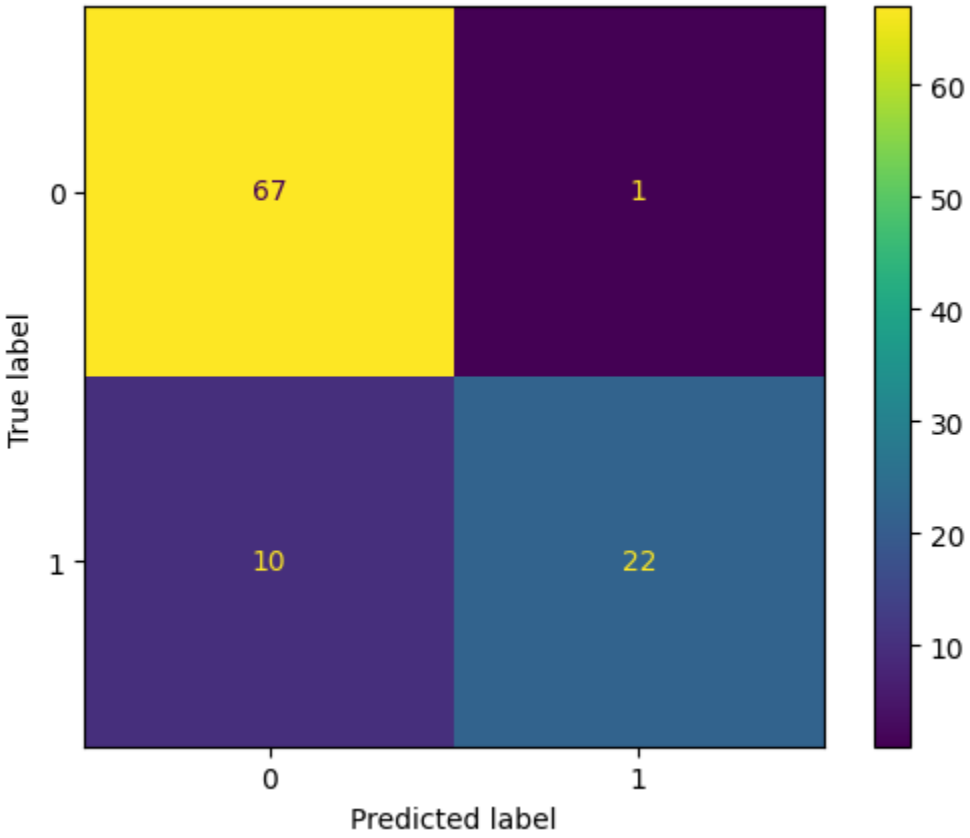
In [138…
```
y_test.value_counts()
```

Out[138…
```
Purchased
0    68
1    32
Name: count, dtype: int64
```

In [139…
```
from sklearn.metrics import ConfusionMatrixDisplay
```

In [140…
```
ConfusionMatrixDisplay.from_estimator(classifier, x_test, y_test)
```

Out[140…
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fe9d4b
0aa50>
```



In [141…
```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,y_pred)
```

Out[141… 0.89

In [142…
```
from sklearn.metrics import classification_report
```

In [143…
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.99      0.92        68
           1       0.96      0.69      0.80        32

    accuracy                           0.89       100
   macro avg       0.91      0.84      0.86       100
weighted avg       0.90      0.89      0.88       100
```