In [9]:
```
import numpy as np
```

In [10]:
```
import pandas as pd
```

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
from sklearn.linear_model import LogisticRegression
```

In [16]:
```
from sklearn.metrics import confusion_matrix
```

In [17]:
```

```

In [18]:
```
from sklearn.metrics import accuracy_score
```

In [36]:
```
from sklearn.metrics import precision_score, recall_score
```

In [37]:
```
df = pd.read_csv("/home/comp/Desktop/Social_Network_Ads.csv")
```

In [38]:
```
df
```

Out[38]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
...	...	...	...
395	46	41000	1
396	51	23000	1
397	50	20000	1
398	36	33000	0
399	49	36000	1

400 rows × 3 columns

In [40]:
```
df.columns
```

Out[40]:
```
Index(['Age', 'EstimatedSalary', 'Purchased'], dtype='object')
```

In [42]:
```
x = df[['Age', 'EstimatedSalary']]
```

In [44]:
```
x
```

Out[44]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [46]:
```
y = df[['Purchased']]
```

In [48]:
```
y
```

Out[48]:

	Purchased
0	0
1	0
2	0
3	0
4	0
...	...
395	1
396	1
397	1
398	0
399	1

400 rows × 1 columns

In [50]:
```
x_train,x_test, y_train, y_test = train_test_split(x,y,test_size = 0.
```

In [58]:
```
model = LogisticRegression()
model.fit(x_train, y_train.values.ravel())
```

Out[58]:
```
LogisticRegression()
```

In [60]:
```
y_pred = model.predict(x_test)
```

In [62]:
```
y_pred
```

Out[62]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [64]:
```
model.score(x_train, y_train)
```

Out[64]:
```
0.6266666666666667
```

In [66]:
```
model.score(x,y)
```

Out[66]:
```
0.6425
```

In [69]:
```
cm = confusion_matrix(y_test, y_pred)
```

In [71]:
```
cm
```

Out[71]:
```
array([[69,  0],
       [31,  0]])
```

In [74]:
```
tn,fp,fn,tp = confusion_matrix(y_test,y_pred).ravel()
```

In [76]:
```
print(tn,fp,fn,tp)
```

```
69 0 31 0
```

In [78]:
```
a = accuracy_score(y_test, y_pred)
```

In [80]:
```
a
```

Out[80]:
```
0.69
```

In [82]:
```
e = 1-a
```

In [84]:
```
e
```

Out[84]:
```
0.31000000000000005
```

In [87]:
```
precision_score(y_test, y_pred, zero_division=1)
```

Out[87]:
```
1.0
```

In [89]:
```
recall_score(y_test,y_pred)
```

Out[89]:
```
0.0
```