In [1]:	Load data
<pre>In [70]: Out[70]:</pre>	Age Salary Status 0 18 82000 0 1 29 80000 0
	2 47 25000 1 3 45 26000 1 4 46 28000 1 395 35 65000 0
	396 26 80000 0 397 26 52000 0 398 20 86000 0 399 32 18000 0 400 rows × 3 columns
In [33]:	Summarizing data set print(dataset.shape) print(dataset.head(5)) (400, 3)
	Age Salary Status 0 18 82000
In [34]:	<pre>X = dataset.iloc[:, :-1].values</pre>
	<pre>[45, 26000], [46, 28000], [48, 29000], [45, 22000], [47, 49000], [48, 41000], [45, 22000], [46, 23000],</pre>
	<pre>[47, 20000], [49, 28000], [47, 30000], [29, 43000], [31, 18000], [31, 74000], [27, 137000], [21, 16000], [28, 44000],</pre>
	<pre>[27, 90000], [35, 27000], [33, 28000], [30, 49000], [26, 72000], [27, 31000], [27, 17000], [33, 51000],</pre>
	[35, 108000], [30, 15000], [28, 84000], [23, 20000], [27, 54000], [30, 135000], [31, 89000], [24, 32000],
	<pre>[18, 44000], [29, 83000], [35, 23000], [27, 58000], [24, 55000], [23, 48000], [28, 79000], [22, 18000],</pre>
	<pre>[32, 117000], [27, 20000], [25, 87000], [23, 66000], [32, 120000], [59, 83000], [24, 58000], [24, 58000],</pre>
	<pre>[23, 82000], [22, 63000], [31, 68000], [25, 80000], [24, 27000], [20, 23000], [33, 113000], [32, 18000], [34, 112000],</pre>
	<pre>[18, 52000], [22, 27000], [28, 87000], [26, 17000], [30, 80000], [39, 42000], [20, 49000], [35, 88000],</pre>
	[30, 62000], [31, 118000], [24, 55000], [28, 85000], [26, 81000], [32, 81000], [32, 81000], [32, 81000], [32, 81000],
	<pre>[29, 28000], [29, 83000], [35, 44000], [35, 25000], [28, 123000], [35, 73000], [28, 37000], [27, 88000],</pre>
	<pre>[28, 59000], [32, 86000], [33, 149000], [19, 21000], [21, 72000], [26, 35000], [27, 89000], [26, 86000], [38, 80000],</pre>
	[39, 71000], [37, 71000], [38, 61000], [37, 55000], [42, 80000], [40, 57000], [35, 75000], [36, 52000],
	<pre>[40, 5900], [41, 59000], [36, 75000], [37, 72000], [40, 75000], [41, 51000], [41, 51000], [39, 61000], [42, 65000],</pre>
	<pre>[26, 32000], [30, 17000], [26, 84000], [31, 58000], [33, 31000], [30, 87000], [21, 68000], [22, 63000],</pre>
	<pre>[20, 82000], [30, 107000], [28, 59000], [19, 25000], [19, 85000], [18, 68000], [35, 59000], [30, 89000],</pre>
	[34, 25000], [24, 89000], [27, 96000], [41, 30000], [29, 61000], [20, 74000], [26, 15000], [41, 45000], [31, 76000],
	<pre>[36, 50000], [40, 47000], [31, 15000], [46, 59000], [29, 75000], [26, 30000], [32, 135000], [32, 100000],</pre>
	[25, 90000], [37, 33000], [35, 38000], [38, 69000], [18, 86000], [22, 55000], [29, 148000], [29, 47000],
	<pre>[21, 88000], [34, 115000], [26, 118000], [34, 43000], [34, 72000], [23, 28000], [35, 47000], [25, 22000],</pre>
	<pre>[24, 23000], [31, 34000], [26, 16000], [31, 71000], [32, 117000], [33, 43000], [33, 60000], [20, 82000],</pre>
	<pre>[33, 41000], [35, 72000], [28, 32000], [24, 84000], [19, 26000], [29, 43000], [19, 70000], [28, 89000], [34, 43000],</pre>
	<pre>[30, 79000], [20, 36000], [26, 80000], [35, 22000], [35, 39000], [49, 74000], [39, 134000], [41, 71000],</pre>
	[58, 101000], [47, 47000], [55, 130000], [52, 114000], [40, 142000], [48, 96000], [52, 150000], [59, 42000],
	[35, 58000], [47, 43000], [60, 108000], [49, 65000], [40, 78000], [46, 96000], [59, 143000], [41, 80000],
	[35, 91000], [37, 144000], [60, 102000], [35, 60000], [37, 53000], [36, 126000], [56, 133000], [40, 72000],
	<pre>[42, 80000], [35, 147000], [39, 42000], [40, 107000], [49, 86000], [38, 112000], [46, 79000], [40, 57000], [37, 80000],</pre>
	<pre>[46, 82000], [53, 143000], [42, 149000], [38, 59000], [50, 88000], [56, 104000], [41, 72000], [51, 146000], [35, 50000],</pre>
	<pre>[57, 122000], [41, 52000], [35, 97000], [44, 39000], [37, 52000], [48, 134000], [37, 146000], [50, 44000],</pre>
	<pre>[52, 90000], [41, 72000], [40, 57000], [58, 95000], [45, 131000], [35, 77000], [36, 144000], [35, 72000],</pre>
	<pre>[48, 90000], [42, 108000], [40, 75000], [37, 74000], [47, 144000], [49, 61000], [43, 133000], [59, 76000], [60, 42000],</pre>
	<pre>[39, 106000], [57, 26000], [57, 74000], [38, 71000], [49, 88000], [52, 38000], [50, 36000], [59, 88000],</pre>
	[35, 61000], [37, 70000], [52, 21000], [48, 141000], [37, 93000], [37, 62000], [48, 138000], [41, 79000], [37, 78000],
	<pre>[39, 134000], [49, 89000], [55, 39000], [37, 77000], [36, 63000], [42, 73000], [43, 112000], [45, 79000],</pre>
	[43, 75000], [46, 117000], [48, 74000], [48, 74000], [37, 137000], [40, 60000], [42, 54000], [51, 134000],
	<pre>[47, 113000], [36, 125000], [38, 50000], [42, 70000], [39, 96000], [38, 50000], [39, 79000], [39, 75000],</pre>
	[54, 104000], [35, 55000], [45, 32000], [36, 60000], [52, 138000], [53, 82000], [41, 52000], [48, 30000], [48, 131000],
	<pre>[41, 60000], [41, 72000], [42, 75000], [36, 118000], [47, 107000], [38, 51000], [48, 119000], [42, 65000],</pre>
	<pre>[40, 65000], [57, 60000], [36, 54000], [58, 144000], [35, 79000], [38, 55000], [39, 122000], [53, 104000], [53, 75000],</pre>
	[38, 65000], [47, 51000], [47, 105000], [41, 63000], [53, 72000], [54, 108000], [39, 77000], [38, 61000], [38, 61000],
	<pre>[37, 75000], [42, 90000], [37, 57000], [36, 99000], [60, 34000], [54, 70000], [41, 72000], [40, 71000],</pre>
	<pre>[42, 54000], [43, 129000], [53, 34000], [47, 50000], [42, 79000], [42, 104000], [59, 29000], [58, 47000], [46, 88000],</pre>
	[38, 71000], [54, 26000], [60, 46000], [60, 83000], [39, 730000], [59, 130000], [37, 80000], [46, 32000], [46, 74000],
	<pre>[42, 53000], [41, 87000], [58, 23000], [42, 64000], [48, 33000], [44, 139000], [49, 28000], [57, 33000],</pre>
	<pre>[56, 60000], [49, 39000], [39, 71000], [47, 34000], [48, 35000], [48, 33000], [47, 23000], [45, 45000], [60, 42000],</pre>
	<pre>[39, 59000], [46, 41000], [51, 23000], [50, 20000], [36, 33000], [49, 36000], [19, 19000], [35, 20000], [26, 43000],</pre>
	<pre>[27, 57000], [19, 76000], [27, 58000], [27, 84000], [32, 150000], [25, 33000], [35, 65000], [26, 80000],</pre>
In [37]:	
ouc[3/].	array([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
	0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
In [38]:	
In [39]: In [40]: In [41]:	Feature scalling from sklearn.preprocessing import StandardScaler
In [41]: In [42]: In [43]:	<pre>X_train=sc.fit_transform(X_train)</pre>
In [44]: In [45]: In [46]:	from sklearn.linear_model import LogisticRegression model=LogisticRegression(random_state=0)
Out[46]:	Predicting , wheather new customer with age and salry will buy on not Y_pred=model.predict(X_test)
<pre>In [63]: Out[63]:</pre>	Y_pred array([0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
In [65]:	<pre>Predicting , weather new customer age and salry will buy or not age = int(input("Enter the age:")) sal = int(input("Enter the salry:")) newCust = [[age,sal]] result = model.predict(sc.transform(newCust))</pre>
	<pre>print(result) if result ==1: print("Customer will but") else: print("Customer won't but") Enter the age:45 Enter the salry:200000 [1]</pre>
	Evaluation MOdel - Confusion Matrix from sklearn.metrics import confusion_matrix , accuracy_score
In [67]: In [68]:	[[61 0] [20 19]]