

```
In [19]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
```

```
In [8]: data = pd.read_csv('Social_Network_Ads.csv')
data.head(5)
```

```
Out[8]:
```

	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

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 2 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                   27 non-null    int64
1   brought_insurance     27 non-null    int64
dtypes: int64(2)
memory usage: 560.0 bytes
```

```
In [9]: x = data.iloc[:,2:4]
y = data.iloc[:,4]
```

```
In [12]: #splitting data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_
```

```
In [13]: scale = StandardScaler()
x_train = scale.fit_transform(x_train)
x_test = scale.transform(x_test)
lr = LogisticRegression(random_state = 0,solver = 'lbfgs')
lr.fit(x_train,y_train)
pred = lr.predict(x_test)
print
(x_test[:10])
print('-'*15)
print(pred[:10])
```

```
-----
[0 1 0 1 0 0 1 0 0 0]
```

```
In [14]: print(pred[:20])
print('-'*15)
print(y_test[:20])
```

```
[0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 1]
```

```
-----
```

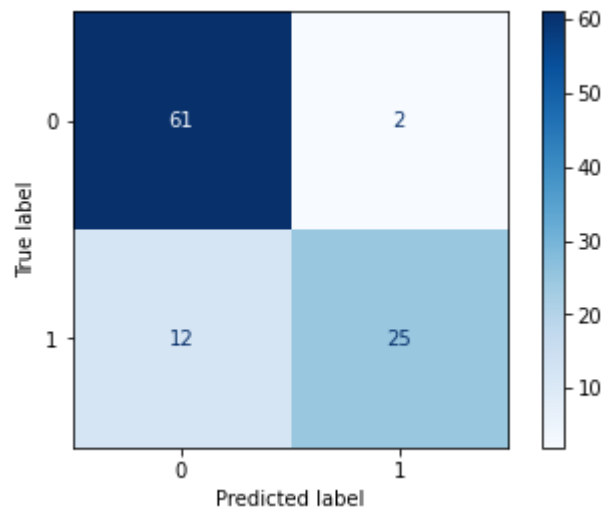
```
209    0
280    1
33     0
210    1
93     0
84     0
329    1
94     0
266    0
126    0
9      0
361    1
56     0
72     0
132    0
42     0
278    1
376    0
231    0
385    1
```

```
Name: Purchased, dtype: int64
```

```
In [15]: matrix = confusion_matrix(y_test,pred,labels = lr.classes_)
print(matrix)
tp, fn, fp, tn = confusion_matrix(y_test,pred,labels=[1,0]).reshape(-1)
```

```
[[61  2]
 [12 25]]
```

```
In [16]: conf_matrix = ConfusionMatrixDisplay(confusion_matrix=matrix,display_labels=lr.c[
conf_matrix.plot(cmap=plt.cm.Blues)
plt.show()
```



```
In [17]: print(classification_report(y_test,pred))
```

	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

```
In [20]: print('\nAccuracy: {:.2f}'.format(accuracy_score(y_test,pred)))
```

Accuracy: 0.86

```
In [ ]:
```