```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import metrics
data = pd.read_csv('iris.csv')

data.head(10)
data.shape
```

```
(150, 5)
```

```
X = data.drop('Name', axis=1)
y = data['Name']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.20)
logr = linear_model.LogisticRegression()
logr.fit(X_train,y_train)

#predict if tumor is cancerous where the size is 3.46mm:
predicted = logr.predict([[5.0,3.6,1.4,0.2]])

print(predicted)
y_pred = logr.predict(X_test)
print(X_test,y_pred )
```

```
['Iris-setosa']
     SepalLength  SepalWidth  PetalLength  PetalWidth
10           5.4         3.7          1.5         0.2
67           5.8         2.7          4.1         1.0
111          6.4         2.7          5.3         1.9
66           5.6         3.0          4.5         1.5
15           5.7         4.4          1.5         0.4
99           5.7         2.8          4.1         1.3
88           5.6         3.0          4.1         1.3
112          6.8         3.0          5.5         2.1
138          6.0         3.0          4.8         1.8
97           6.2         2.9          4.3         1.3
64           5.6         2.9          3.6         1.3
3            4.6         3.1          1.5         0.2
35           5.0         3.2          1.2         0.2
13           4.3         3.0          1.1         0.1
45           4.8         3.0          1.4         0.3
100          6.3         3.3          6.0         2.5
20           5.4         3.4          1.7         0.2
84           5.4         3.0          4.5         1.5
117          7.7         3.8          6.7         2.2
93           5.0         2.3          3.3         1.0
39           5.1         3.4          1.5         0.2
```

```
58           6.6          2.9           4.6          1.3
130          7.4          2.8           6.1          1.9
14           5.8          4.0           1.2          0.2
125          7.2          3.2           6.0          1.8
7            5.0          3.4           1.5          0.2
135          7.7          3.0           6.1          2.3
26           5.0          3.4           1.6          0.4
79           5.7          2.6           3.5          1.0
1            4.9          3.0           1.4          0.2 ['Iris-setosa'
'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica' 'Iris-
setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-setosa']

C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[12  0  0]
 [ 0 10  0]
 [ 0  0  8]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        12
Iris-versicolor       1.00      1.00      1.00        10
 Iris-virginica       1.00      1.00      1.00         8

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

```
X = data.drop('Name', axis=1)
y = data['Name']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.30)
logr = linear_model.LogisticRegression()
logr.fit(X_train,y_train)
```

```
#predict if tumor is cancerous where the size is 3.46mm:
predicted = logr.predict([[5.0,3.6,1.4,0.2]])

print(predicted)
y_pred = logr.predict(X_test)
print(X_test,y_pred )
```

```
['Iris-setosa']
     SepalLength  SepalWidth  PetalLength  PetalWidth
10           5.4         3.7          1.5         0.2
34           4.9         3.1          1.5         0.1
51           6.4         3.2          4.5         1.5
8            4.4         2.9          1.4         0.2
31           5.4         3.4          1.5         0.4
118          7.7         2.6          6.9         2.3
48           5.3         3.7          1.5         0.2
90           5.5         2.6          4.4         1.2
1            4.9         3.0          1.4         0.2
37           4.9         3.1          1.5         0.1
21           5.1         3.7          1.5         0.4
30           4.8         3.1          1.6         0.2
107          7.3         2.9          6.3         1.8
91           6.1         3.0          4.6         1.4
15           5.7         4.4          1.5         0.4
54           6.5         2.8          4.6         1.5
74           6.4         2.9          4.3         1.3
109          7.2         3.6          6.1         2.5
16           5.4         3.9          1.3         0.4
43           5.0         3.5          1.6         0.6
58           6.6         2.9          4.6         1.3
126          6.2         2.8          4.8         1.8
67           5.8         2.7          4.1         1.0
116          6.5         3.0          5.5         1.8
52           6.9         3.1          4.9         1.5
131          7.9         3.8          6.4         2.0
25           5.0         3.0          1.6         0.2
57           4.9         2.4          3.3         1.0
80           5.5         2.4          3.8         1.1
99           5.7         2.8          4.1         1.3
73           6.1         2.8          4.7         1.2
32           5.2         4.1          1.5         0.1
115          6.4         3.2          5.3         2.3
84           5.4         3.0          4.5         1.5
138          6.0         3.0          4.8         1.8
77           6.7         3.0          5.0         1.7
20           5.4         3.4          1.7         0.2
23           5.1         3.3          1.7         0.5
50           7.0         3.2          4.7         1.4
128          6.4         2.8          5.6         2.1
124          6.7         3.3          5.7         2.1
```

```
46           5.1         3.8          1.6         0.2
135          7.7         3.0          6.1         2.3
62           6.0         2.2          4.0         1.0
122          7.7         2.8          6.7         2.0 ['Iris-setosa'
'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica']

C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[17  0  0]
 [ 0 15  1]
 [ 0  0 12]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        17
Iris-versicolor       1.00      0.94      0.97        16
 Iris-virginica       0.92      1.00      0.96        12

       accuracy                           0.98        45
      macro avg       0.97      0.98      0.98        45
   weighted avg       0.98      0.98      0.98        45
```

```
X = data.drop('Name', axis=1)
y = data['Name']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.40)
logr = linear_model.LogisticRegression()
logr.fit(X_train,y_train)

#predict if tumor is cancerous where the size is 3.46mm:
predicted = logr.predict([[5.0,3.6,1.4,0.2]])
```

```
print(predicted)
y_pred = logr.predict(X_test)
print(X_test,y_pred )
```

```
['Iris-setosa']
     SepalLength  SepalWidth  PetalLength  PetalWidth
38           4.4         3.0          1.3         0.2
122          7.7         2.8          6.7         2.0
127          6.1         3.0          4.9         1.8
95           5.7         3.0          4.2         1.2
149          5.9         3.0          5.1         1.8
34           4.9         3.1          1.5         0.1
7            5.0         3.4          1.5         0.2
67           5.8         2.7          4.1         1.0
2            4.7         3.2          1.3         0.2
14           5.8         4.0          1.2         0.2
28           5.2         3.4          1.4         0.2
97           6.2         2.9          4.3         1.3
31           5.4         3.4          1.5         0.4
30           4.8         3.1          1.6         0.2
73           6.1         2.8          4.7         1.2
146          6.3         2.5          5.0         1.9
17           5.1         3.5          1.4         0.3
72           6.3         2.5          4.9         1.5
62           6.0         2.2          4.0         1.0
45           4.8         3.0          1.4         0.3
142          5.8         2.7          5.1         1.9
78           6.0         2.9          4.5         1.5
52           6.9         3.1          4.9         1.5
91           6.1         3.0          4.6         1.4
43           5.0         3.5          1.6         0.6
1            4.9         3.0          1.4         0.2
22           4.6         3.6          1.0         0.2
9            4.9         3.1          1.5         0.1
133          6.3         2.8          5.1         1.5
36           5.5         3.5          1.3         0.2
141          6.9         3.1          5.1         2.3
48           5.3         3.7          1.5         0.2
39           5.1         3.4          1.5         0.2
80           5.5         2.4          3.8         1.1
53           5.5         2.3          4.0         1.3
18           5.7         3.8          1.7         0.3
145          6.7         3.0          5.2         2.3
88           5.6         3.0          4.1         1.3
94           5.6         2.7          4.2         1.3
47           4.6         3.2          1.4         0.2
40           5.0         3.5          1.3         0.3
59           5.2         2.7          3.9         1.4
54           6.5         2.8          4.6         1.5
13           4.3         3.0          1.1         0.1
```

```
83           6.0         2.7          5.1         1.6
49           5.0         3.3          1.4         0.2
111          6.4         2.7          5.3         1.9
35           5.0         3.2          1.2         0.2
5            5.4         3.9          1.7         0.4
118          7.7         2.6          6.9         2.3
12           4.8         3.0          1.4         0.1
32           5.2         4.1          1.5         0.1
10           5.4         3.7          1.5         0.2
102          7.1         3.0          5.9         2.1
113          5.7         2.5          5.0         2.0
147          6.5         3.0          5.2         2.0
140          6.7         3.1          5.6         2.4
79           5.7         2.6          3.5         1.0
114          5.8         2.8          5.1         2.4
63           6.1         2.9          4.7         1.4 ['Iris-setosa'
'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-
setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica' 'Iris-
setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor']

C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[27  0  0]
 [ 0 17  1]
 [ 0  0 15]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        27
```

```
Iris-versicolor       1.00      0.94      0.97        18
 Iris-virginica       0.94      1.00      0.97        15

       accuracy                           0.98        60
      macro avg       0.98      0.98      0.98        60
   weighted avg       0.98      0.98      0.98        60
```

```
X = data.drop('Name', axis=1)
y = data['Name']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.50)
logr = linear_model.LogisticRegression()
logr.fit(X_train,y_train)

#predict if tumor is cancerous where the size is 3.46mm:
predicted = logr.predict([[5.0,3.6,1.4,0.2]])

print(predicted)
y_pred = logr.predict(X_test)
print(X_test,y_pred )
```

```
['Iris-setosa']
     SepalLength  SepalWidth  PetalLength  PetalWidth
72           6.3         2.5          4.9         1.5
124          6.7         3.3          5.7         2.1
11           4.8         3.4          1.6         0.2
18           5.7         3.8          1.7         0.3
130          7.4         2.8          6.1         1.9
..           ...         ...          ...         ...
39           5.1         3.4          1.5         0.2
96           5.7         2.9          4.2         1.3
78           6.0         2.9          4.5         1.5
82           5.8         2.7          3.9         1.2
38           4.4         3.0          1.3         0.2

[75 rows x 4 columns] ['Iris-versicolor' 'Iris-virginica' 'Iris-
setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
```

```
 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa']

C:\Users\MGM\anaconda3\Lib\site-packages\sklearn\base.py:493:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
```

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[27  0  0]
 [ 0 21  3]
 [ 0  0 24]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        27
Iris-versicolor       1.00      0.88      0.93        24
 Iris-virginica       0.89      1.00      0.94        24

       accuracy                           0.96        75
      macro avg       0.96      0.96      0.96        75
   weighted avg       0.96      0.96      0.96        75
```