```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
```

```
x = np.arange(10).reshape(-1, 1)
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

```
x
```

```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5],
       [6],
       [7],
       [8],
       [9]])
```

```
y
```

```
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

```
model = LogisticRegression(solver='liblinear', random_state=0)
```

```
model.fit(x, y)
```

```
▾                  LogisticRegression
LogisticRegression(random_state=0, solver='liblinear')
```

```
model = LogisticRegression(solver='liblinear', random_state=0).fit(x, y)
```

```
model.classes_
```

```
array([0, 1])
```

```
model.intercept_
```

```
array([-1.04608067])
```

```
model.coef_
```

```
array([[0.51491375]])
```

```
model.predict_proba(x)
```

```
array([[0.74002157, 0.25997843],
       [0.62975524, 0.37024476],
       [0.5040632 , 0.4959368 ],
       [0.37785549, 0.62214451],
       [0.26628093, 0.73371907],
       [0.17821501, 0.82178499],
       [0.11472079, 0.88527921],
       [0.07186982, 0.92813018],
       [0.04422513, 0.95577487],
       [0.02690569, 0.97309431]])
```

```
model.predict(x)
```

```
array([0, 0, 0, 1, 1, 1, 1, 1, 1, 1])
```

```
model.score(x, y)
```

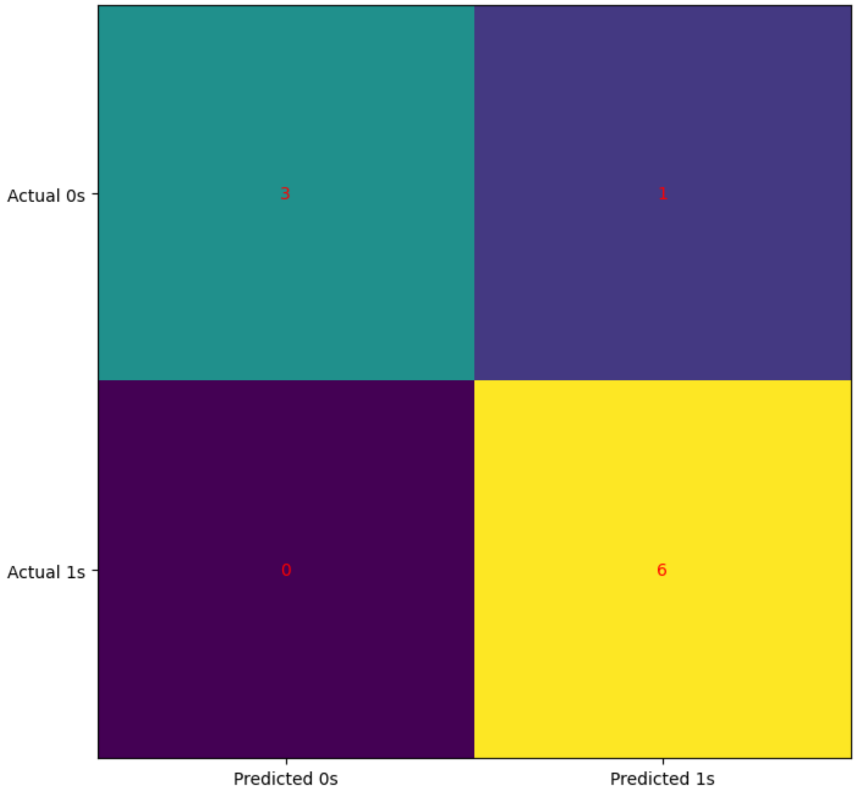
```
0.9
```

```
confusion_matrix(y, model.predict(x))
```

```
array([[3, 1],
       [0, 6]])
```

```
cm = confusion_matrix(y, model.predict(x))

fig, ax = plt.subplots(figsize=(8, 8))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
    for j in range(2):
        ax.text(j, i, cm[i, j], ha='center', va='center', color='red')
plt.show()
```



```
print(classification_report(y, model.predict(x)))
```

```
              precision    recall  f1-score   support

           0       1.00      0.75      0.86         4
           1       0.86      1.00      0.92         6

    accuracy                           0.90        10
   macro avg       0.93      0.88      0.89        10
weighted avg       0.91      0.90      0.90        10
```

```
model = LogisticRegression(solver='liblinear', C=10.0, random_state=0)
model.fit(x, y)
```

```
▾                  LogisticRegression
LogisticRegression(C=10.0, random_state=0, solver='liblinear')
```

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import load_digits
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
x, y = load_digits(return_X_y=True)
```

```
x_train, x_test, y_train, y_test =\
    train_test_split(x, y, test_size=0.2, random_state=0)
```

```
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
```

```
model = LogisticRegression(solver='liblinear', C=0.05, multi_class='ovr',
                           random_state=0)
model.fit(x_train, y_train)
```

```
▾                    LogisticRegression
LogisticRegression(C=0.05, multi_class='ovr', random_state=0,
                   solver='liblinear')
```

```
x_test = scaler.transform(x_test)
```

```
y_pred = model.predict(x_test)
```

```
model.score(x_train, y_train)
model.score(x_test, y_test)
```

```
0.9416666666666667
```

```
confusion_matrix(y_test, y_pred)
```

```
array([[27,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 32,  0,  0,  0,  0,  1,  0,  1,  1],
       [ 1,  1, 33,  1,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  1, 28,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 29,  0,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  0, 39,  0,  0,  0,  1],
       [ 0,  1,  0,  0,  0,  0, 43,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 39,  0,  0],
       [ 0,  2,  1,  2,  0,  0,  0,  1, 33,  0],
       [ 0,  0,  0,  1,  0,  1,  0,  2,  1, 36]])
```

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

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        27
           1       0.89      0.91      0.90        35
           2       0.94      0.92      0.93        36
           3       0.88      0.97      0.92        29
           4       1.00      0.97      0.98        30
           5       0.97      0.97      0.97        40
           6       0.98      0.98      0.98        44
           7       0.91      1.00      0.95        39
           8       0.94      0.85      0.89        39
           9       0.95      0.88      0.91        41

    accuracy                           0.94       360
   macro avg       0.94      0.94      0.94       360
weighted avg       0.94      0.94      0.94       360
```

✓ 0s completed at 2:15 PM ● ×