

In [25]:

```
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

In [2]:

```
data = load_iris()
```

In [5]:

```
type(data)
```

Out[5]:

```
sklearn.utils.Bunch
```

In [6]:

```
data.keys()
```

Out[6]:

```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names',
'filename'])
```

In [8]:

```
feature = data['data']
feature.shape
```

Out[8]:

```
(150, 4)
```

In [9]:

```
target = data['target']
target.shape
```

Out[9]:

```
(150,)
```

In [10]:

```
type(feature)
```

Out[10]:

```
numpy.ndarray
```

```
print(feature)
```

In [12]:

```
print(target)
```

In [13]:

```
print(data[ 'target_names' ])
```

```
['setosa' 'versicolor' 'virginica']
```

In [14]:

```
print(data['DESCR'])
```

```
.. _iris_dataset:
```

```
Iris plants dataset
```

```
-----
```

```
**Data Set Characteristics:**
```

```
:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive attributes and the class
```

```
:Attribute Information:
```

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm
- class:
  - Iris-Setosa
  - Iris-Versicolour
  - Iris-Virginica

```
:Summary Statistics:
```

|               | Min | Max | Mean | SD   | Class Correlation |
|---------------|-----|-----|------|------|-------------------|
| sepal length: | 4.3 | 7.9 | 5.84 | 0.83 | 0.7826            |
| sepal width:  | 2.0 | 4.4 | 3.05 | 0.43 | -0.4194           |
| petal length: | 1.0 | 6.9 | 3.76 | 1.76 | 0.9490 (high!)    |
| petal width:  | 0.1 | 2.5 | 1.20 | 0.76 | 0.9565 (high!)    |

```
:Missing Attribute Values: None
```

```
:Class Distribution: 33.3% for each of 3 classes.
```

```
:Creator: R.A. Fisher
```

```
:Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
```

```
:Date: July, 1988
```

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken from Fisher's paper. Note that it's the same as in R, but not as in the UCI Machine Learning Repository, which has two wrong data points.

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

```
.. topic:: References
```

- Fisher, R.A. "The use of multiple measurements in taxonomic problems"

ems"

Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis. (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al's AUTOCLASS II conceptual clustering system finds 3 classes in the data.
- Many, many more ...

In [15]:

```
print(data['feature_names'])
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [16]:

```
print(data['filename'])
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/datasets/data/iris.csv
```

In [21]:

```
featurePd = pd.DataFrame(feature)
featurePd.isnull().sum()
```

Out[21]:

```
0    0
1    0
2    0
3    0
dtype: int64
```

In [22]:

```
targetPd = pd.DataFrame(target)
targetPd.isnull().sum()
```

Out[22]:

```
0    0
dtype: int64
```

In [26]:

```
X_train, X_test, y_train, y_test = train_test_split(feature, target, test_size=0.3,
```

In [27]:

```
stdsc = StandardScaler()#实例化
X_train_conti_std = stdsc.fit_transform(X_train)
X_test_conti_std = stdsc.fit_transform(X_test)
```

In [29]:

```
print(X_test_conti_std)
```

```
[[ 0.10652036 -0.6802299  0.90797473  1.77175914]
 [ 0.39703042 -2.10396689  0.24507283 -0.18650096]
 [-0.32924474  2.64182309 -1.32178623 -1.30550673]
 [ 2.28534586 -0.4429404  1.63114045  0.93250481]
 [-1.05551991  0.7435071 -1.26152242 -1.30550673]
 [ 0.83279553  0.5062176  1.45034902  1.91163486]
 [-1.05551991  0.9807966 -1.38205004 -1.16563101]
 [ 1.41381566  0.0316386  0.66691949  0.51287764]
 [ 1.55907069 -0.6802299  0.7271833  0.37300192]
 [ 0.54228546 -0.6802299  0.24507283  0.2331262 ]
 [ 0.54228546 -1.15480889  1.20929378  0.37300192]
 [ 0.97805056  0.2689281  0.54639188  0.51287764]
 [ 0.54228546 -0.6802299  0.66691949  0.09325048]
 [ 1.12330559 -0.6802299  0.60665568  0.51287764]
 [ 0.54228546 -0.4429404  0.66691949  0.37300192]
 [-1.20077494  1.21808609 -1.32178623 -1.44538245]
 [ 0.39703042 -0.4429404  0.54639188  0.51287764]
 [-0.32924474 -1.15480889  0.48612807  0.09325048]
 [-1.34602998 -0.2056509 -1.32178623 -1.16563101]
 [-0.47449978  1.92995459 -1.38205004 -1.02575529]
 [-0.18398971 -0.6802299  0.78744711  1.21225625]
 [-0.18398971 -0.2056509  0.54639188  0.51287764]
 [-1.34602998  0.7435071 -1.02046718 -1.30550673]
 [-1.92705011 -0.4429404 -1.32178623 -1.30550673]
 [ 0.68754049 -0.6802299  0.7271833  0.93250481]
 [-1.63654004  1.21808609 -1.56284147 -1.30550673]
 [-0.91026488  1.69266509 -1.02046718 -1.02575529]
 [ 0.68754049 -0.4429404  0.42586426  0.2331262 ]
 [-1.05551991 -1.86667739 -0.17677384 -0.18650096]
 [-1.05551991  0.7435071 -1.20125861 -1.02575529]
 [ 0.97805056  0.0316386  1.14902997  0.93250481]
 [-0.47449978 -0.2056509  0.54639188  0.51287764]
 [-0.76500984  0.9807966 -1.26152242 -1.30550673]
 [ 0.54228546 -0.2056509  0.78744711  0.93250481]
 [ 0.97805056 -0.6802299  1.20929378  1.49200769]
 [-0.76500984 -0.9175194  0.18480902  0.37300192]
 [-0.03873468  1.69266509 -1.1409948 -1.16563101]
 [ 0.39703042 -0.9175194  0.90797473  0.65275337]
 [ 0.25177539 -0.2056509  0.36560045  0.51287764]
 [ 0.10652036 -1.15480889  0.24507283  0.09325048]
 [ 1.55907069 -0.2056509  1.14902997  1.35213197]
 [-1.49128501  0.2689281 -1.38205004 -1.30550673]
 [ 1.70432573  0.0316386  0.90797473  1.63188341]
 [-1.05551991  0.9807966 -1.20125861 -0.74600385]
 [-0.47449978  1.45537559 -1.26152242 -1.30550673]]
```

In [28]:

```
print(X_train_conti_std)
```

```
[[-1.02366372 -2.37846268 -0.18295039 -0.29318114]
 [ 0.69517462 -0.10190314  0.93066067  0.7372463 ]
 [ 0.92435306  0.58106472  1.04202177  1.63887031]
 [ 0.1222285  -1.92315077  0.6522579   0.35083601]
 [ 0.92435306 -1.24018291  1.09770233  0.7372463 ]
 [-0.33612839 -1.24018291  0.03977182 -0.16437771]
 [ 2.07024529 -0.10190314  1.26474398  1.38126345]
 [ 0.46599617  0.58106472  0.48521625  0.47963944]
 [-0.45071761 -1.46783886 -0.01590873 -0.16437771]
 [ 0.46599617 -0.784871    0.59657735  0.7372463 ]
 [ 0.46599617 -0.55721505  0.70793846  0.35083601]
 [-1.13825295 -1.24018291  0.37385514  0.60844287]
 [ 0.46599617 -1.24018291  0.6522579   0.86604973]
 [ 1.26812073  0.35340877  0.48521625  0.22203258]
 [ 0.69517462 -0.10190314  0.76361901  0.99485316]
 [ 0.1222285   0.80872067  0.37385514  0.47963944]
 [-1.25284217  0.12575281 -1.24088089 -1.32360858]
 [-0.10694994 -0.784871    0.70793846  0.86604973]
 [-0.33612839 -0.784871    0.20681348  0.09322915]
 [-0.33612839 -0.32955909 -0.12726983  0.09322915]
 [-0.45071761 -1.24018291  0.09545238  0.09322915]
 [ 0.23681773 -0.10190314  0.42953569  0.22203258]
 [ 1.49729918  0.35340877  1.20906343  0.7372463 ]
 [-0.67989605  1.49168853 -1.29656144 -1.32360858]
 [-1.82578828 -0.10190314 -1.51928365 -1.45241201]
 [ 0.5805854  -0.784871    0.81929956  0.86604973]
 [-0.22153916 -0.10190314  0.20681348 -0.03557428]
 [-0.56530683  0.80872067 -1.18520034 -1.32360858]
 [-0.22153916  3.08528021 -1.29656144 -1.06600172]
 [ 1.15353151  0.12575281  0.59657735  0.35083601]
 [-1.48202061  0.12575281 -1.29656144 -1.32360858]
 [ 0.00763928 -0.10190314  0.70793846  0.7372463 ]
 [-0.9090745  -1.24018291 -0.46135315 -0.16437771]
 [-1.48202061  0.80872067 -1.35224199 -1.19480515]
 [ 0.35140695 -1.92315077  0.37385514  0.35083601]
 [ 1.49729918  1.26403258  1.26474398  1.63887031]
 [-0.22153916 -0.32955909  0.20681348  0.09322915]
 [-1.25284217 -0.10190314 -1.35224199 -1.45241201]
 [ 1.38270995 -0.10190314  1.15338288  1.12365659]
 [ 1.15353151  0.35340877  1.04202177  1.38126345]
 [ 0.69517462 -0.10190314  1.09770233  1.25246002]
 [ 0.5805854  -0.55721505  0.98634122  1.12365659]
 [-0.9090745   1.71934449 -1.24088089 -1.32360858]
 [-1.25284217  0.80872067 -1.24088089 -1.32360858]
 [ 0.69517462  0.35340877  0.70793846  0.99485316]
 [ 0.92435306  0.58106472  1.04202177  1.12365659]
 [-1.59660984 -1.69549482 -1.40792255 -1.19480515]
 [ 0.35140695  0.80872067  0.87498011  1.38126345]
 [-1.13825295 -0.10190314 -1.35224199 -1.32360858]
 [-0.22153916 -1.24018291  0.6522579   0.99485316]
 [ 1.15353151  0.12575281  0.87498011  1.12365659]
 [-1.71119906  0.35340877 -1.40792255 -1.32360858]
 [-1.02366372  1.26403258 -1.35224199 -1.32360858]
 [ 1.49729918 -0.10190314  1.09770233  0.47963944]
 [-0.9090745   1.03637663 -1.35224199 -1.19480515]
 [-1.71119906 -0.10190314 -1.40792255 -1.32360858]
 [-0.56530683  1.94700044 -1.18520034 -1.06600172]
```

```
[ -0.45071761 -1.69549482  0.09545238  0.09322915 ]
[  1.03894229  0.35340877  1.15338288  1.38126345 ]
[  1.95565607 -0.10190314  1.54314675  1.12365659 ]
[ -0.9090745   1.03637663 -1.35224199 -1.32360858 ]
[ -1.13825295  0.12575281 -1.29656144 -1.32360858 ]
[ -0.79448528  0.80872067 -1.35224199 -1.32360858 ]
[ -0.22153916 -0.55721505  0.37385514  0.09322915 ]
[  0.80976384 -0.10190314  0.31817459  0.22203258 ]
[ -1.02366372  0.35340877 -1.4636031  -1.32360858 ]
[ -0.9090745   0.58106472 -1.18520034 -0.93719829 ]
[  0.5805854  -0.32955909  0.26249403  0.09322915 ]
[ -0.56530683  0.80872067 -1.29656144 -1.06600172 ]
[  2.07024529 -1.01252695  1.71018841  1.38126345 ]
[ -1.13825295 -1.46783886 -0.29431149 -0.29318114 ]
[  2.29942374  1.71934449  1.43178564  0.99485316 ]
[  0.92435306  0.12575281  0.31817459  0.22203258 ]
[ -0.79448528  2.40231235 -1.29656144 -1.45241201 ]
[  0.1222285  -0.10190314  0.5408968   0.7372463  ]
[ -0.10694994  2.17465639 -1.4636031  -1.32360858 ]
[  2.07024529 -0.55721505  1.5988273   0.99485316 ]
[ -0.9090745   1.71934449 -1.29656144 -1.19480515 ]
[ -1.36743139  0.35340877 -1.24088089 -1.32360858 ]
[  1.72647762 -0.55721505  1.26474398  0.86604973 ]
[ -1.02366372  0.58106472 -1.35224199 -1.32360858 ]
[  0.46599617  0.80872067  0.98634122  1.51006688 ]
[ -0.22153916 -0.55721505  0.15113293  0.09322915 ]
[ -0.10694994 -0.784871    0.03977182 -0.03557428 ]
[ -0.22153916 -1.01252695 -0.18295039 -0.29318114 ]
[  0.5805854   0.35340877  0.81929956  1.38126345 ]
[  0.92435306 -0.10190314  0.76361901  1.38126345 ]
[  0.46599617 -1.24018291  0.59657735  0.35083601 ]
[  0.92435306 -0.10190314  0.6522579   0.60844287 ]
[ -1.02366372 -0.10190314 -1.24088089 -1.32360858 ]
[ -0.45071761 -1.46783886 -0.07158928 -0.29318114 ]
[  0.92435306  0.12575281  0.98634122  1.51006688 ]
[ -0.10694994 -0.784871    0.70793846  0.86604973 ]
[ -0.9090745   0.80872067 -1.29656144 -1.32360858 ]
[  0.80976384 -0.32955909  0.42953569  0.09322915 ]
[ -0.33612839 -0.10190314  0.15113293  0.09322915 ]
[  0.00763928  0.35340877  0.5408968   0.7372463  ]
[  0.46599617 -1.69549482  0.31817459  0.09322915 ]
[ -0.45071761  1.03637663 -1.40792255 -1.32360858 ]
[ -0.9090745   1.49168853 -1.29656144 -1.06600172 ]
[ -1.13825295  0.12575281 -1.29656144 -1.45241201 ]
[  0.46599617 -0.32955909  0.98634122  0.7372463  ]
[ -0.10694994 -0.784871    0.15113293 -0.29318114 ]
[  2.07024529  1.71934449  1.5988273   1.25246002 ]
[ -1.48202061  0.35340877 -1.35224199 -1.32360858 ]]
```

In [30]:

```
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
confusion_matrix = confusion_matrix(y_test, y_pred)#打印混淆矩阵, 是很多评分函数的标准来源
print(confusion_matrix)#打印混淆矩阵
```

```
[[16  0  0]
 [ 0 13  5]
 [ 0  0 11]]
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

In [31]:

```
classifier.score(X_test, y_test)
```

Out[31]:

```
0.8888888888888888
```

In [34]:

```
#基于训练集使用逻辑回归建模
classifierStd = LogisticRegression(random_state=0)#实例化算法
classifierStd.fit(X_train_conti_std, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifierStd.predict(X_test_conti_std)#预测
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

In [35]:

```
classifier.score(X_test_conti_std, y_test)
```

Out[35]:

```
0.7777777777777778
```

In [36]:

```
featureTwo = feature[:, :2]
```



In [37]:

```
print(featureTwo)
```

```
[7.2 3. ]
[7.4 2.8]
[7.9 3.8]
[6.4 2.8]
[6.3 2.8]
[6.1 2.6]
[7.7 3. ]
[6.3 3.4]
[6.4 3.1]
[6.  3. ]
[6.9 3.1]
[6.7 3.1]
[6.9 3.1]
[5.8 2.7]
[6.8 3.2]
[6.7 3.3]
[6.7 3. ]
[6.3 2.5]
[6.5 3. ]
[6.2 3.4]
[5.  2. ]
```

In [38]:

```
X_train, X_test, y_train, y_test = train_test_split(featureTwo, target, test_size=0.2)
```

In [40]:

```
#基于训练集使用逻辑回归建模
```

```
classifier = LogisticRegression(random_state=0)#实例化算法
```

```
classifier.fit(X_train, y_train)#模型训练
```

```
# 将模型应用于测试集并查看混淆矩阵
```

```
y_pred = classifier.predict(X_test)#预测
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

In [41]:

```
classifier.score(X_test, y_test)
```

Out[41]:

0.6

In [42]:

```
featureTmp = feature[:, [0,2]]
```

In [43]:

```
print(featureTmp)
```

```
[[5.1 1.4]
 [4.9 1.4]
 [4.7 1.3]
 [4.6 1.5]
 [5. 1.4]
 [5.4 1.7]
 [4.6 1.4]
 [5. 1.5]
 [4.4 1.4]
 [4.9 1.5]
 [5.4 1.5]
 [4.8 1.6]
 [4.8 1.4]
 [4.3 1.1]
 [5.8 1.2]
 [5.7 1.5]
 [5.4 1.3]
 [5.1 1.4]
 [5.7 1.7]
 - - - -]
```

In [44]:

```
X_train, X_test, y_train, y_test = train_test_split(featureTmp, target, test_size=0.2,
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

In [45]:

```
classifier.score(X_test, y_test)
```

Out[45]:

```
0.8333333333333334
```

In [46]:

```
featureTmp2 = feature[:, [2,3]]
```

In [47]:

```
X_train, X_test, y_train, y_test = train_test_split(featureTmp2, target, test_size=0.3)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[47]:

0.7

In [48]:

```
featureTmp3 = feature[:, [0,3]]
X_train, X_test, y_train, y_test = train_test_split(featureTmp3, target, test_size=0.3)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[48]:

0.7555555555555555

In [49]:

```
X_train, X_test, y_train, y_test = train_test_split(featureTmp2, target, test_size=0.2)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[49]:

0.6888888888888889

In [51]:

```
stdsc = StandardScaler()#实例化
featureStd = stdsc.fit_transform(feature)
```

In [52]:

```
X_train, X_test, y_train, y_test = train_test_split(featureStd, target, test_size=0.2)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(random_state=0)#实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[52]:

0.8666666666666667

In [53]:

```
featureTmp3 = feature[:, [0,2]]
X_train, X_test, y_train, y_test = train_test_split(featureTmp2, target, test_size=0.2)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(C=1e5) #实例化算法
classifier.fit(X_train, y_train) #模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test) #预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[53]:

0.9666666666666667

In [54]:

```
X_train, X_test, y_train, y_test = train_test_split(featureTmp3, target, test_size=0.2)
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(C=1e5) #实例化算法
classifier.fit(X_train, y_train) #模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test) #预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[54]:

0.9666666666666667

In [55]:

```
X_train, X_test, y_train, y_test = train_test_split(feature, target, test_size=0.2,
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(C=1e5) #实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[55]:

1.0

In [56]:

```
X_train, X_test, y_train, y_test = train_test_split(feature, target, test_size=0.3,
#基于训练集使用逻辑回归建模
classifier = LogisticRegression(C=1e5) #实例化算法
classifier.fit(X_train, y_train)#模型训练
# 将模型应用于测试集并查看混淆矩阵
y_pred = classifier.predict(X_test)#预测
classifier.score(X_test, y_test)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
```

```
FutureWarning)
```

```
/Users/didi/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

Out[56]:

0.9777777777777777

In [ ]: