

homework_05_solutions_Marco_Sciorilli

April 21, 2021

```
[27]: from sklearn import datasets
      from sklearn.model_selection import train_test_split

      dataset = datasets.load_wine()

      X = dataset.data
      y = dataset.target

      print("\nX.shape =", X.shape)
      print("\ny.shape =", y.shape)
      print("\nwine categories:\n", dataset['target'])
      print("\nfeatures names:\n", dataset['feature_names'])
```

```
X.shape = (178, 13)
```

```
y.shape = (178,)
```

```
wine categories:  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

```
features names:
```

```
['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',  
'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',  
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
```

0.1 1.

```
[38]: y_train=[]
      y_test=[]

      while 0 not in y_train or 1 not in y_train or 2 not in y_train \
            or 0 not in y_test or 1 not in y_test or 2 not in y_test:
```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
↪random_state=0)
print("\nX_train.shape =", X_train.shape)
print("\ny_train.shape =", y_train.shape)
print("\nX_test.shape =", X_test.shape)
print("\ny_test.shape =", y_test.shape)
print("\nwine categories train:\n", y_train)
print("\nwine categories test:\n", y_test)

```

X_train.shape = (124, 13)

y_train.shape = (124,)

X_test.shape = (54, 13)

y_test.shape = (54,)

wine categories train:

```

[0 1 2 2 1 2 2 0 1 2 1 1 2 1 0 1 1 1 0 0 1 1 2 2 1 1 1 2 2 0 2 1 1 1 1 1 0
 0 1 0 2 0 2 0 0 1 0 1 1 0 2 1 0 1 1 1 2 0 2 2 0 0 1 2 0 0 1 1 0 0 0 2 1 0
 1 2 0 1 2 2 0 0 2 0 2 1 0 0 1 0 2 1 2 0 2 2 2 0 1 1 1 1 2 2 1 1 0 1 2 2 0
 0 2 1 1 1 0 0 0 1 1 1 0 2]

```

wine categories test:

```

[0 2 1 0 1 1 0 2 1 1 2 2 0 1 2 1 0 0 1 0 1 0 0 1 1 1 1 1 1 2 0 0 1 0 0 0 2
 1 1 2 0 0 1 1 1 0 2 1 2 0 2 2 0 2]

```

0.2 2.

```

[50]: from sklearn.linear_model import LogisticRegression
      from sklearn.metrics import classification_report

model = LogisticRegression(random_state=0,max_iter=10000).fit(X_train, y_train)
predictions = model.predict(X_test)
print(classification_report(y_test, predictions))

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	0.95	0.98	22
2	0.93	1.00	0.96	13
accuracy			0.98	54
macro avg	0.98	0.98	0.98	54
weighted avg	0.98	0.98	0.98	54

0.3 3.

```
[51]: from sklearn.naive_bayes import GaussianNB
gauss = GaussianNB()
modelNB = gauss.fit(X_train, y_train)
predictionsNB= modelNB.predict(X_test)
print(classification_report(y_test, predictionsNB))
```

	precision	recall	f1-score	support
0	0.90	1.00	0.95	19
1	1.00	0.86	0.93	22
2	0.93	1.00	0.96	13
accuracy			0.94	54
macro avg	0.94	0.95	0.95	54
weighted avg	0.95	0.94	0.94	54