



温州大學  
WENZHOU UNIVERSITY

# 机器学习-机器学习库Scikit-learn

黄海广 副教授

2021年06月

**01 Scikit-learn概述**

**02 Scikit-learn主要用法**

**03 Scikit-learn案例**

# 1.Scikit-learn概述

3

**01 Scikit-learn概述**

**02 Scikit-learn主要用法**

**03 Scikit-learn案例**

# 1.Scikit-learn概述

4

Scikit-learn是基于NumPy、 SciPy和 Matplotlib的开源Python机器学习包,它封装了一系列数据预处理、机器学习算法、模型选择等工具,是数据分析师首选的机器学习工具包。

自2007年发布以来, scikit-learn已经成为Python重要的机器学习库了, scikit-learn简称sklearn, 支持包括分类, 回归, 降维和聚类四大机器学习算法。还包括了特征提取, 数据处理和模型评估三大模块。





## 2.Scikit-learn主要用法

6

**01 Scikit-learn概述**

**02 Scikit-learn主要用法**

**03 Scikit-learn案例**

## 2.Scikit-learn主要用法

7

### 符号标记

`X_train` | 训练数据.

`X_test` | 测试数据.

`X` | 完整数据.

`y_train` | 训练集标签.

`y_test` | 测试集标签.

`y` | 数据标签.

# 2.Scikit-learn主要用法

8

## 基本建模流程

### 导入工具包

```
from sklearn import datasets, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```



## 2.Scikit-learn主要用法

9

### 加载数据

✓ Scikit-learn支持以NumPy的arrays对象、Pandas对象、SciPy的稀疏矩阵及其他可转换为数值型arrays的数据结构作为其输入，前提是数据必须是数值型的

✓ sklearn.datasets模块提供了一系列加载和获取著名数据集如鸢尾花、波士顿房价、Olivetti人脸、MNIST数据集等的工具，也包括了一些toy data如S型数据等的生成工具

```
from sklearn.datasets import load_iris  
  
iris = load_iris()  
X = iris.data  
y = iris.target
```

## 2.Scikit-learn主要用法

10

### 数据划分

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y,  
random_state=12, stratify=y, test_size=0.3)
```

数据集

训练集

测试集

将完整数据集的70%作为训练集，30%作为测试集，并使得测试集和训练集中各类别数据的比例与原始数据集比例一致（stratify分层策略），另外可通过设置 `shuffle=True` 提前打乱数据

## 2.Scikit-learn主要用法

11

### 数据预处理

使用Scikit-learn进行数据标准化

```
from sklearn.preprocessing import StandardScaler
```

构建转换器实例

```
scaler = StandardScaler()
```

拟合及转换

```
scaler.fit_transform(X_train)
```

### Z-Score标准化

$$x^* = \frac{x - \mu}{\sigma}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$$
$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

处理后的数据均值为0，方差为1

# 2.Scikit-learn主要用法

12

## 数据预处理

使用Scikit-learn进行数据变换

最小最大标准化

One-Hot编码

归一化

二值化 (单个特征转换)

标签编码

缺失值填补

多项式特征生成

MinMaxScaler

OneHotEncoder

Normalizer

Binarizer

LabelEncoder

Imputer

PolynomialFeatures



**归一化 (最大-最小规范化)**

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

将数据映射到[0,1]区间

## 2.Scikit-learn主要用法

13

### 特征选择

```
from sklearn import feature_selection as fs  
fs.SelectKBest(score_func, k)
```

过滤式 (Filter) , 保留得分排名前k的特征 (top k方式)

```
fs.RFECV(estimator, scoring="r2")
```

封装式 (Wrapper) , 结合交叉验证的递归特征消除法, 自动选择最优特征个数

```
fs.SelectFromModel(estimator)
```

嵌入式 (Embedded) , 从 模型中自动选择特征, 任何具有coef\_或者  
feature\_importances\_的 基模型都可以作为estimator参数传入

## 2.Scikit-learn主要用法

14

### 监督学习算法-回归

```
from sklearn.linear_model import LinearRegression
```

#### 构建模型实例

```
lr = LinearRegression(normalize=True)
```

#### 训练模型

```
lr.fit(X_train, y_train)
```

#### 作出预测

```
y_pred = lr.predict(X_test)
```

LASSO    `linear_model.Lasso`

Ridge    `linear_model.Ridge`

ElasticNet    `linear_model.ElasticNet`

回归树        `tree.DecisionTreeRegressor`



## 2.Scikit-learn主要用法

15

### 监督学习算法-分类

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=5)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)
```

使用决策树分类算法解决二分类问题, `y_prob` 为每个样本预测为  
“0” 和 “1” 类的概率

# 1.Scikit-learn概述

16

## 监督学习算法-分类

逻辑回归	<code>linear_model.LogisticRegression</code>
支持向量机	<code>svm.SVC</code>
朴素贝叶斯	<code>naive_bayes.GaussianNB</code>
K近邻	<code>neighbors.NearestNeighbors</code>

# 2.Scikit-learn主要用法

17

## 监督学习算法-集成学习

sklearn.ensemble模块包含了一系列基于集成思想的分类、回归和离群值检测方法.

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=20)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)
```

AdaBoost

`ensemble.AdaBoostClassifier`

`ensemble.AdaBoostRegressor`

基于梯度提升

`ensemble.GradientBoostingClassifier`

`ensemble.GradientBoostingRegressor`

## 2.Scikit-learn主要用法

18

### 无监督学习算法

sklearn.cluster模块包含了一系列无监督聚类算法.

```
from sklearn.cluster import KMeans
```

### 构建聚类实例

```
kmeans = KMeans(n_clusters=3, random_state=0)
```

### 拟合

```
kmeans.fit(X_train)
```

### 预测

```
kmeans.predict(X_test)
```

## 2.Scikit-learn主要用法

19

### 无监督学习算法

DBSCAN            `cluster.DBSCAN`

层次聚类          `cluster.AgglomerativeClustering`

谱聚类            `cluster.SpectralClustering`

## 2.Scikit-learn主要用法

20

### 评价指标

sklearn.metrics模块包含了一系列用于评价模型的评分函数、损失函数以及成对数据的距离度量函数.

```
from sklearn.metrics import accuracy_score  
accuracy_score(y_true, y_pred)
```

对于测试集而言, `y_test`即是`y_true`, 大部分函数都必须包含真实值`y_true`和预测值`y_pred`.



## 2.Scikit-learn主要用法

21

### 评价指标

#### 回归模型评价

<code>metrics.mean_absolute_error()</code>		平均绝对误差MAE
<code>metrics.mean_squared_error()</code>		均方误差MSE
<code>metrics.r2_score()</code>		决定系数 $R^2$ .

## 2.Scikit-learn主要用法

22

### 评价指标

#### 分类模型评价

`metrics.accuracy_score()` | 正确率

`metrics.precision_score()` | 各类精确率

`metrics.f1_score()` | F1 值

`metrics.log_loss()` | 对数损失或交叉熵损失

`metrics.confusion_matrix` | 混淆矩阵

`metrics.classification_report` | 含多种评价的分类报告

## 2.Scikit-learn主要用法

23

评价指标

分类模型评价

`metrics.accuracy_score()` | 正确率 .

`metrics.precision_score()` | 各类精确率.

`metrics.f1_score()` | F1 值 .

`metrics.log_loss()` | 对数损失或交叉熵损失.

`metrics.confusion_matrix` | 混淆矩阵.

`metrics.classification_report` | 含多种评价的分类报告.

## 2.Scikit-learn主要用法

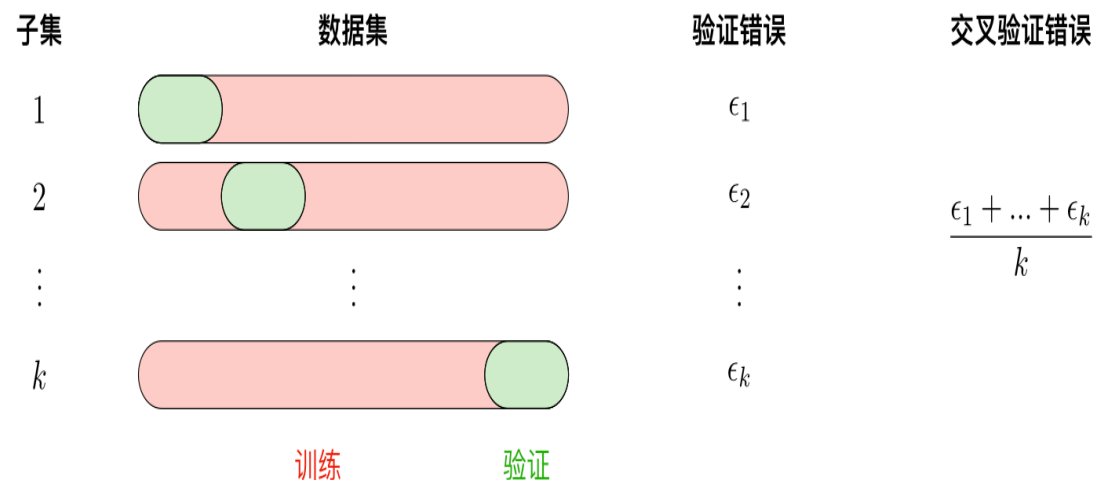
24

### 交叉验证及超参数调优

```
from sklearn.model_selection import cross_val_score  
  
clf = DecisionTreeClassifier(max_depth=5)  
scores = cross_val_score(clf, X_train, y_train,  
                          cv=5, scoring='f1_weighted')
```

使用5折交叉验证对决策树模型进行评估,  
使用的评分函数为F1值

sklearn提供了部分带交叉验证功能的模型  
类如LassoCV、LogisticRegressionCV等,  
这些类包含cv参数



## 2.Scikit-learn主要用法

25

### 交叉验证及超参数调优

#### 超参数调优——网格搜索

```
from sklearn.model_selection import GridSearchCV
from sklearn import svm
svc = svm.SVC()
params = {'kernel':['linear', 'rbf'], 'C':[1, 10]} grid_search =
GridSearchCV(svc, params, cv=5) grid_search.fit(X_train, y_train)
grid_search.best_params_
```

在参数网格上进行穷举搜索，方法简单但是**搜索速度慢**（超参数较多时），且不容易找到参数空间中的局部最优

## 2.Scikit-learn主要用法

26

### 交叉验证及超参数调优

#### 超参数调优——随机搜索

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
svc = svm.SVC()
param_dist = {'kernel': ['linear', 'rbf'], 'C': randint(1, 20)}
random_search = RandomizedSearchCV(svc, param_dist, n_iter=10)
random_search.fit(X_train, y_train)
random_search.best_params_
```

在参数子空间中进行随机搜索，选取空间中的100个点进行建模（可从scipy.stats常见分布如正态分布norm、均匀分布uniform中随机采样得到），时间耗费较少，更容易找到局部最优



# 3.Scikit-learn案例

27

**01 Scikit-learn概述**

**02 Scikit-learn主要用法**

**03 Scikit-learn案例**

# 3.Scikit-learn案例

28

见Jupyter notebook 代码

1. <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>  
， scikit-learn (sklearn) 官方文档
2. <https://sklearn.apachecn.org/> ， scikit-learn (sklearn) 官方  
文档中文版

谢谢!

