第3章

机器学习库 Scikit-learn

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。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。

3.1 背景知识

机器学习的编程语言没有限制，读者可选用自己熟悉的语言对算法进行实现。本书的代码基于 Python 编写。回顾一下本书需要掌握的 Python 相关知识，限于篇幅，本书不再对 Python 的使用进行详细讲解，仅列出使用 Python 进行机器学习算法所需要掌握的知识点。

（1）Python 环境的安装：包括安装 Anaconda、Jupyter 和 PyCharm。

（2）Python 数据结构：列表、元组、集合、字典。

① 列表是用来存储一连串元素的容器，用[ ]来表示，其他元素类型可不相同。

② 元组与列表类似，元组中的元素也可进行索引计算，用()来表示。二者的区别是列表里面的元素值可以被修改，而元组中的元素值不可以被修改，只能读取。

③ 集合有两个功能：一是进行集合操作；二是消除重复元素。集合的格式是 set()，其中()内可以是列表、字典或字符串。

④ 字典(dict)也称作关联数组，用{ }表示，使用键-值存储。

（3）Python 控制流：顺序结构、分支结构、循环结构。

（4）Python 函数：定义函数、调用函数、高阶函数。

（5）Python 主要模块：NumPy、Pandas、SciPy、Matplotlib、Scikit-learn。

（6）NumPy：是一个用 Python 实现的科学计算扩展程序库。

（7）Pandas：是基于 NumPy 的一种工具，为解决数据分析任务而创建。

（8）SciPy：是一款为科学和工程设计的工具包，包括统计、优化、线性代数、

傅里叶变化等。

(9) Matplotlib：是一款 2D 绘图库，以各种硬拷贝格式和跨平台的交互式环境生成绘图、直方图、功率谱、条形图等。

(10) Scikit-learn：Python 重要的机器学习库。

Scikit-learn 概述

3.2 Scikit-learn 概述

Scikit-learn 是基于 Python 语言的机器学习工具。它建立在 NumPy、SciPy、Pandas 和 Matplotlib 之上，里面的应用程序编程接口(application programming interface，API)的设计非常好，所有对象的接口简单，很适合新手。

Scikit-learn 库的算法主要有四类：分类、回归、聚类、降维。

(1) 常用的回归：线性回归、决策树回归、SVM 回归、KNN 回归。集成回归：随机森林、Adaboost、GradientBoosting、Bagging、ExtraTrees。

(2) 常用的分类：线性分类、决策树、SVM、KNN，朴素贝叶斯。集成分类：随机森林、Adaboost、GradientBoosting、Bagging、ExtraTrees。

(3) 常用聚类：K 均值(K-means)、层次聚类(hierarchical clustering)、DBSCAN。

(4) 常用降维：线性判别分析(linear discriminant analysis，LDA)、PCA。

图 3-1 代表了 Scikit-learn 算法选择的一个简单路径，这个路径图代表：蓝色圆圈(见彩插)是判断条件，绿色方框是可以选择的算法，可以根据自己的数据特征和任务目标去找一条自己的操作路线。

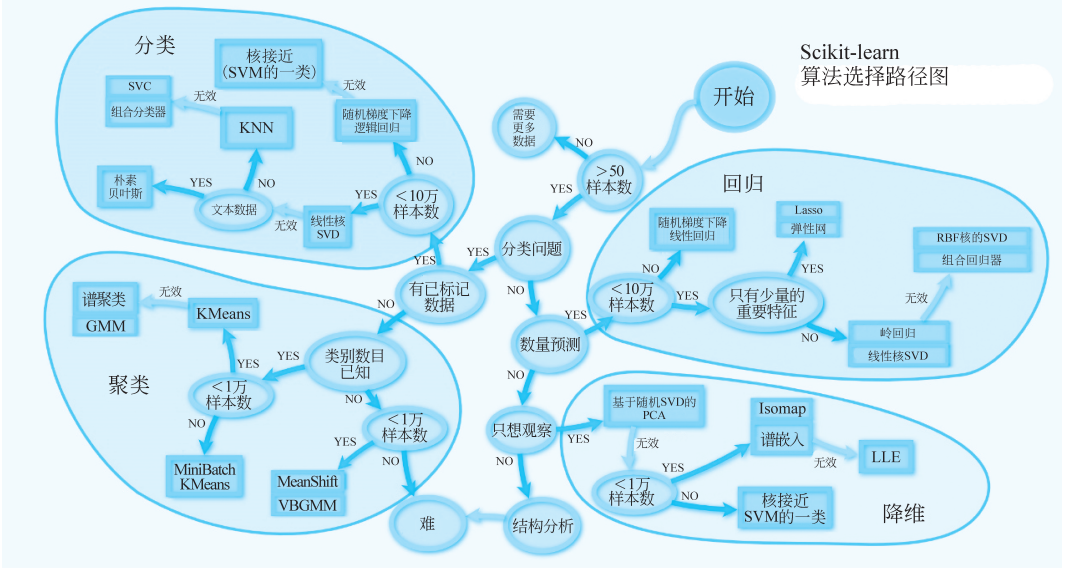

图 3-1 Scikit-learn 算法选择路径图(见彩插)

Scikit-learn 中包含众多与数据预处理和特征工程相关的模块，但其实 Scikit-learn 六大板块中有两块都是关于数据预处理和特征工程的，两个板块互相交互，为建模之前的全

部工程打下基础。


Scikit-learn
主要用法 1

3.3　Scikit-learn 主要用法

3.3.1　基本建模流程

基本建模的符号标记如表 3-1 所示。

表 3-1　符号标记

符　　号	代 表 含 义
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
y_pred	预测标签

1. 导入工具包

导入工具包的方法如下(这里使用伪代码)。

```
from sklearn import 包名称
from sklearn.库名称 import 包名称
```

代码示例如下。

```
from sklearn import datasets, preprocessing
#导入数据集,数据预处理库
from sklearn.model_selection import train_test_split
#从模型选择库导入数据切分包
from sklearn.linear_model import LinearRegression
#从线性模型库导入线性回归包
from sklearn.metrics import r2_score
#从评价指标库导入 R2 评价指标
```

2. 导入数据

导入数据的方法如下。

```
from sklearn.datasets import 数据名称
```

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。

Scikit-learn 内置了很多可以用于机器学习的数据,用两行代码就可以使用这些数据。内置数据分为可以直接使用的自带数据集、需要下载的自带数据集及生成数据集。

1) 可以直接使用的自带数据集

此类数据集可以直接导入使用数据,数据集和描述如表 3-2 所示。

表 3-2 可以直接使用的自带数据集

数据集名称	描 述	类 型	维 度
load_boston	波士顿房屋价格	回归	506×13
fetch_california_housing	加州住房	回归	20640×9
load_diabetes	糖尿病	回归	442×10
load_digits	手写字	分类	1797×64
load_breast_cancer	乳腺癌	分类、聚类	(357+212)×30
load_iris	鸢尾花	分类、聚类	(50×3)×4
load_wine	葡萄酒	分类	(59+71+48)×13
load_linnerud	体能训练	多分类	20

2) 需要下载的自带数据集

此类数据集第一次使用,需要联网下载数据,数据集和描述如表 3-3 所示。

表 3-3 需要下载的自带数据集

数据集名称	描 述
fetch_20newsgroups	用于文本分类、文本挖掘和信息检索研究的国际标准数据集之一,数据集收集了大约 20 000 个新闻组文档,均匀分为 20 个不同主题的新闻组集合,返回一个可以提取文本特征的提取器
fetch_20newsgroups_vectorized	这是上面这个文本数据向量化后的数据,返回一个已提取特征的文本序列,即不需要使用特征提取器
fetch_california_housing	加利福尼亚的房价数据,总计 20 640 个样本,每个样本由 8 个属性表示,房价作为 target,所有属性值均为 number,详情可调用 fetch_california_housing()['DESCR'],了解每个属性的具体含义
fetch_covtype	森林植被类型,总计 581 012 个样本,每个样本由 54 个维度表示(12 个属性,其中 2 个分别是 onehot4 维和 onehot40 维),target 表示植被类型 1~7,所有属性值均为 number,详情可调用 fetch_covtype()['DESCR']了解每个属性的具体含义
fetch_kddcup99	KDD 竞赛在 1999 年举行时采用的数据集,KDD99 数据集仍然是网络入侵检测领域的事实 Benckmark,为基于计算智能的网络入侵检测研究奠定基础,包含 41 项特征

续表

数据集名称	描　　述
fetch_lfw_pairs	该任务称为人脸验证：给定一对两张图片，二分类器必须预测这两张图片是否来自同一个人
fetch_lfw_people	打好标签的人脸数据集
fetch_mldata	从 mldata.org 中下载数据集
fetch_olivetti_faces	Olivetti 脸部图片数据集
fetch_rcv1	路透社新闻语聊数据集
fetch_species_distributions	物种分布数据集

3）生成数据集

此类数据集可以用于分类任务、回归任务、聚类任务、流形学习、因子分解任务等。这些函数产生样本特征向量矩阵及对应的类别标签集合，数据集和描述如表 3-4 所示。

表 3-4　生成数据集

数据集名称	描　　述
make_blobs	多类单标签数据集，为每个类分配一个或多个正态分布的点集
make_classification	多类单标签数据集，为每个类分配一个或多个正态分布的点集，提供了为数据添加噪声的方式，包括维度相关性、无效特征及冗余特征等
make_gaussian-quantiles	将一个单高斯分布的点集划分为两个数量均等的点集，作为两类
make_hastie-10-2	产生一个相似的二元分类数据集，有 10 个维度
make_circle 和 make_moons	产生二维二元分类数据集来测试某些算法的性能，可以为数据集添加噪声，可以为二元分类器产生一些球形判决界面的数据

代码示例如下。

```
1.  #导入内置的鸢尾花数据
2.  from sklearn.datasets import load_iris
3.  iris = load_iris( )
4.  #定义数据、标签
5.  X = iris.data
6.  y = iris.target
```

3.3.2　数据预处理

Scikit-learn 主要用法 2

1. 数据划分

机器学习的数据，可以划分为训练集、验证集和测试集，也可以划分为训练集和测试集（见图 3-2）。

代码示例如下。

图 3-2 数据集划分

```
1. from sklearn.model_selection import train_test_split
2. X_train, X_test, y_train, y_test =  train_test_split(X, y, random_state=
      12, stratify=y,  test_size=0.3)
3. #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. 数据变换操作

sklearn.preprocessing 模块包含了数据变换的主要操作(见表 3-5),数据变换的方法如下。

```
from sklearn.preprocessing import 库名称
```

表 3-5 使用 Scikit-learn 进行数据变换

预处理操作	库 名 称
标准化	StandardScaler
最小最大标准化	MinMaxScaler
One-Hot 编码	OneHotEncoder
归一化	Normalizer
二值化(单个特征转换)	Binarizer
标签编码	LabelEncoder
缺失值填补	Imputer
多项式特征生成	PolynomialFeatures

代码示例如下。

```
1. #使用 Scikit-learn 进行数据标准化
2. from sklearn.preprocessing import StandardScaler
3. #构建转换器实例
4. scaler = StandardScaler( )
5. #拟合及转换
6. scaler.fit_transform(X_train)
```

3. 特征选择

特征选择的方法如下。

```
#导入特征选择库
from sklearn import feature_selection as fs
```

- 过滤式(filter)。

```
#保留得分排名前 k 的特征(top k 方式)
fs.SelectKBest(score_func, k)
#交叉验证特征选择
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 feature_importances_的基模型都可以作为 estimator 参数传入。

3.3.3　监督学习算法


Scikit-learn 主要用法 3

1. 监督学习算法——回归

常见的回归模型如表 3-6 所示。

表 3-6　常见的回归模型

回归模型名称	库　名　称
线性回归	LinearRegression
岭回归	Ridge
LASSO 回归	Lasso
ElasticNet 回归	ElasticNet
决策树回归	tree.DecisionTreeRegressor

代码示例如下。

```
#从线性模型库导入线性回归模型
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)
#做出预测
y_pred = lr.predict(X_test)
```

2. 监督学习算法——分类

常见的分类模型如表 3-7 所示。

表 3-7 常见的分类模型

模 型 名 称	库 名 称
逻辑回归	linear_model.LogisticRearession
支持向量机	svm.SVC
朴素贝叶斯	naive_bayes.GaussianNB
KNN	neighbors.NearestNeighbors
随机森林	ensemble.RandomForestClassifier
GBDT	ensemble.GradientBoostingClassifier

代码示例如下。

```
#从树模型库导入决策树
from sklearn.tree import DecisionTreeClassifier
#定义模型
clf = DecisionTreeClassifier(max_depth=5)
#训练模型
clf.fit(X_train, y_train)
#使用决策树分类算法解决二分类问题,得到的是类别
y_pred = clf.predict(X_test)
#y_prob 为每个样本预测为 0 和 1 类的概率
y_prob = clf.predict_proba(X_test)
```

3.3.4 无监督学习算法

1. 聚类算法

sklearn.cluster 模块包含了一系列无监督聚类算法,聚类使用的方法如下。

`from sklearn.cluster import 库名称`

常见的聚类模型如表 3-8 所示。

表 3-8 常见的聚类模型

模 型 名 称	库 名 称
K-means	KMeans
DBSCAN	DBSCAN
层次聚类	AgglomerativeClustering
谱聚类	SpectralClustering

代码示例如下。

```
#从聚类模型库导入 KMeans
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. 降维算法

Scikit-learn 中的降维算法都被包括在模块 decomposition 中，sklearn.decomposition 模块本质上是一个矩阵分解模块。最常见的降维方法是 PCA。

降维使用的方法如下。

```
from sklearn.decomposition import 库名称
```

代码示例如下。

```
#导入 PCA 库
from sklearn.decomposition import PCA
#设置主成分数量为 3,n_components 代表主成分数量
pca = PCA(n_components=3)
#训练模型
pca.fit(X)
#投影后各个特征维度的方差比例(这里是 3 个主成分)
print(pca.explained_variance_ratio_)
#投影后的特征维度的方差
print(pca.explained_variance_)
```

3.3.5　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、损失函数及成对数据的距离度量函数。评价指标主要分为分类评价指标、回归评价指标等，表 3-9 列举了常见的几种评价指标。

评价指标使用的方法如下。

```
from sklearn.metrics import 库名称
```

表 3-9 常见评价指标

评价指标	库名称	使用范围
正确率	accuracy_score	分类
精确率	precision_score	分类
F1 值	f1_score	分类
对数损失	log_loss	分类
混淆矩阵	confusion_matrix	分类
含多种评价的分类报告	classification_report	分类
均方误差 MSE	mean_squared_error	回归
平均绝对误差 MAE	mean_absolute_error	回归
决定系数 R2	r2_score	回归

代码示例如下。

```
1. #从评价指标库导入准确率
2. from sklearn.metrics import accuracy_score
3. #计算样本的准确率
4. accuracy_score(y_test, y_pred)
5. #对于测试集而言,大部分函数都必须包含真实值 y_test 和预测值 y_pred
```

3.3.6 交叉验证及超参数调优

1. 交叉验证

交叉验证的方法如图 3-3 所示,具体原理将在第 7 章讲解,本章仅讲解使用方法。

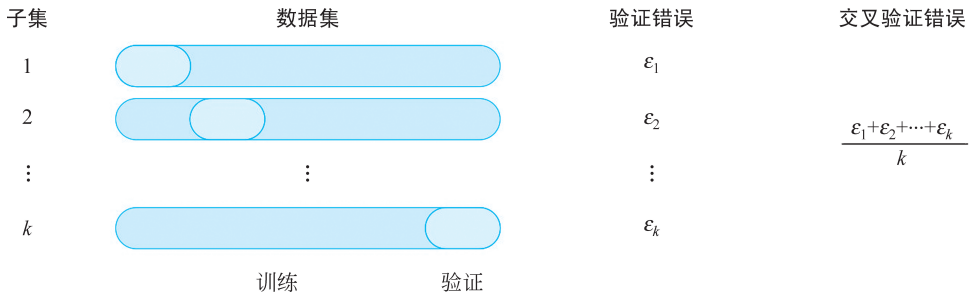

图 3-3 交叉验证示意图

代码示例如下。

```
1. #从模型选择库导入交叉验证分数
2. from sklearn.model_selection import cross_val_score
```

```
clf = DecisionTreeClassifier(max_depth=5)
#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
scores = cross_val_score(clf, X_train, y_train, cv=5, scoring='f1_
    weighted')
```

此外,Scikit-learn 提供了部分带交叉验证功能的模型类,如 LogisticRegressionCV、LassoCV 等,这些类包含 CV 参数。

2. 超参数调优

在机器学习中,超参数指无法从数据中学习而需要在训练前提供的参数。机器学习模型的性能在很大程度上依赖于寻找最佳超参数集。

超参数调整一般指调整模型的超参数,这是一个非常耗时的过程。目前主要有 3 种最流行的超参数调整技术:网格搜索、随机搜索和贝叶斯搜索。其中,Scikit-learn 内置了网格搜索、随机搜索,本章进行简单讲解,其余调参方法如贝叶斯搜索,本章不进行讨论。

1) 网格搜索

代码示例如下。

```
#从模型选择库导入网格搜索
from sklearn.model_selection import GridSearchCV
from sklearn import svm
svc = svm.SVC()
#把超参数集合作为字典
params = {'kernel':['linear', 'rbf'], 'C':[1, 10]}
#进行网格搜索,使用了支持向量机分类器,并进行 5 折交叉验证
grid_search = GridSearchCV(svc, params, cv=5)
#模型训练
grid_search.fit(X_train, y_train)
#获取模型最优超参数组合
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) 随机搜索

代码示例如下。

```
#从模型选择库导入随机搜索
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
svc = svm.SVC()
#把超参数组合作为字典
param_dist = {'kernel':['linear', 'rbf'], 'C':randint(1, 20)}
```

```
#进行随机搜索
random_search = RandomizedSearchCV(svc, param_dist, n_iter=10)
#模型训练
random_search.fit(X_train, y_train)
#获取最优超参数组合
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。

3.4 Scikit-learn 总结

Scikit-learn 是基于 Python 语言的机器学习工具,它建立在 NumPy、SciPy、Pandas 和 Matplotlib 之上,被广泛地用于统计分析和机器学习建模等数据科学领域,其主要优点如下。

(1) 建模方便:用户通过 Scikit-learn 能够实现各种监督学习和非监督学习的模型,仅仅需要几行代码就可以实现。

(2) 功能多样:使用 Scikit-learn 还能够进行数据的预处理、特征工程、数据集切分、模型评估等工作。

(3) 数据丰富:内置丰富的数据集,如泰坦尼克、鸢尾花等,还可以生成数据,非常方便。


Scikit-learn 案例 1


Scikit-learn 案例 2


Scikit-learn 案例 3


Scikit-learn 案例 4

习题

一、回归模型练习(一)

1. 导入预置的波士顿房价数据集,设置房价为 y,特征为 $\boldsymbol{X}$。

```
from sklearn import datasets
X, y = datasets.load_boston(return_X_y=True)
```

2. 设置 30%的数据为测试集(2 行代码)。

3. 导入线性回归模型(1 行代码)。

4. 用线性回归模型拟合波士顿房价数据集(1 行代码)。

5. 用训练完的模型进行预测(1 行代码)。

6. 输出线性回归模型的斜率和截距(2 行代码)。

二、回归模型练习(二)

1. 导入预置的鸢尾花数据集,设置类别为 y,特征为 $\boldsymbol{X}$。

```
1.  X,y = datasets.load_iris(return_X_y=True)
```

2. 设置 30%的数据为测试集(2 行代码)。

3. 导入逻辑回归模型(1 行代码)。

4. 用逻辑回归模型拟合鸢尾花数据集(1 行代码)。

5. 用训练完的模型进行预测(1 行代码)。

6. 输出分类报告(2 行代码)。

参考文献

[1] PEDREGOSA F, VAROQUAUX G, GRAMFORT A, et al. Scikit-learn: machine learning in Python[J]. Journal of Machine Learning Research, 2011, 12: 2825-2830.