

scikit-learn 是一个开源的基于 **python** 语言的机器学习工具包。它通过 NumPy、SciPy 和 Matplotlib 等 python 数值计算的库实现高效的算法应用，并且涵盖了几乎所有主流机器学习算法。sklearn 包含分析采集到的数据、根据数据特征选择适合的算法、在工具包中调用算法、调整算法的参数、获取需要的信息等机器学习算法应用全流程。

Scikit-learn 示例

```
from sklearn import neighbors, datasets, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
iris = datasets.load_iris()
X, y = iris.data[:, :2], iris.target
X_train, X_test, y_train, y_test = train_test_split( \
    X, y, random_state=33)
scaler = preprocessing.StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
knn = neighbors.KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy_score(y_test, y_pred)
0.631578947368421 # 输出accuracy指标得分
```



1. 加载数据

Scikit-learn 处理的数据是存储为 NumPy 数组或 SciPy 稀疏矩阵的数字，还支持 Pandas 数据框等可转换为数字数组的其它数据类型。

```
import numpy as np
X = np.random.random((10, 5))
y = np.array(['M', 'M', 'F', 'F', 'M', 'F', 'M', 'M', 'F', 'F'])
x[X<0.7] = 0
```

2. 训练 / 测试集切分

```
from sklearn.model_selection import train_test_split
x_train, X_test, y_train, y_test = train_test_split( \
    X, y, random_state=0)
```

3. 数据预处理

3.1 标准化

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(X_train) # 拟合
standardized_X = scaler.transform(X_train) # 训练集变换
standardized_X_test = scaler.transform(X_test) # 测试集变换
```

3.2 归一化

```
from sklearn.preprocessing import Normalizer
scaler = Normalizer().fit(X_train) # 拟合
normalized_X = scaler.transform(X_train) # 训练集变换
normalized_X_test = scaler.transform(X_test) # 测试集变换
```

3.3 二值化

```
from sklearn.preprocessing import Binarizer
binarizer = Binarizer(threshold=0.0).fit(X) # 拟合
binary_X = binarizer.transform(X) # 变换
```

3.4 编码分类特征

```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
y = enc.fit_transform(y)
```

3.5 缺失值处理

```
from sklearn.impute import SimpleImputer
imp = Imputer(missing_values=0, strategy='mean') # 均值填充器
imp.fit_transform(X_train) # 对数据进行缺失值均值填充变换
```

3.6 生成多项式特征

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(5)
poly.fit_transform(X)
```

4. 创建模型

4.1 有监督学习评估器

线性回归

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression(normalize=True)
```

支持向量机 (SVM)

```
from sklearn.svm import SVC
svc = SVC(kernel='linear')
```

朴素贝叶斯

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
```

KNN

```
from sklearn import neighbors
knn = neighbors.KNeighborsClassifier(n_neighbors=5)
```

4.2 无监督学习评估器

主成分分析 (PCA)

```
from sklearn.decomposition import PCA
pca = PCA(n_components=0.95)
```

K-Means 聚类

```
from sklearn.cluster import KMeans
k_means = KMeans(n_clusters=3, random_state=0)
```

5. 模型拟合

有监督学习

```
lr.fit(X, y) # 拟合数据与模型
knn.fit(X_train, y_train)
svc.fit(X_train, y_train)
```

无监督学习

```
k_means.fit(X_train) # 拟合数据与模型
# 拟合并转换数据
pca_model = pca.fit_transform(X_train)
```

6. 预测

有监督评估器

```
y_pred = svc.predict(np.random.random((2,5))) # 预测标签
y_pred = lr.predict(X_test) # 预测标签
y_pred = knn.predict_proba(X_test) # 评估标签概率
```

无监督评估器

```
y_pred = k_means.predict(X_test) # 预测聚类算法里的标签
```

7. 评估模型性能

7.1 分类评价指标

准确率

```
svc.fit(X_train, y_train)
# 评估器评分法
svc.score(X_test, y_test)
# 指标评分函数
from sklearn.metrics import accuracy_score
y_pred = svc.predict(X_test)
# 评估 accuracy
accuracy_score(y_test, y_pred)
```

分类预估评价函数

```
# 精确度、召回率、F1 分数及支持率
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

混淆矩阵

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test, y_pred))
```



7.2 回归评价指标

平均绝对误差

```
from sklearn.metrics import mean_absolute_error
house_price = datasets.load_boston()
X, y = house_price.data, house_price.target
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor().fit(X_train, y_train)
y_pred = dt.predict(X_test)
mean_absolute_error(y_test, y_pred)
```

均方误差

```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred)
R^2 评分
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

7.3 聚类评价指标

调整兰德系数

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

同质性

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

V-measure

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

7.4 交叉验证

```
from sklearn.model_selection import cross_val_score
print(cross_val_score(knn, X_train, y_train, cv=4))
print(cross_val_score(lr, X, y, cv=2))
```

8. 模型调整

8.1 网格搜索超参优化

```
from sklearn.model_selection import GridSearchCV
params = {"n_neighbors": np.arange(1, 3), \
          "metric": ["euclidean", "cityblock"]}
grid = GridSearchCV(estimator=knn, \
                    param_grid=params)
grid.fit(X_train, y_train)
print(grid.best_score_)
print(grid.best_estimator_.n_neighbors)
```

8.2 随机搜索超参优化

```
from sklearn.model_selection import RandomizedSearchCV
params = {"n_neighbors": range(1,5), \
          "weights": ["uniform", "distance"]}
rsearch = RandomizedSearchCV(estimator=knn, \
                             param_distributions=params, \
                             cv=4, n_iter=8, random_state=5)
rsearch.fit(X_train, y_train)
print(rsearch.best_score_)
```



扫码回复“工具库”

下载最新全套速查表

Scikit-Learn 速查表

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

参考 | DataCamp Cheatsheet

数据科学工具库速查表



NumPy 是 Python 数据科学计算的核心库，提供了高性能多维数组对象及处理数组的工具。使用以下语句导入 NumPy 库：

```
import numpy as np
```



SciPy 是基于 NumPy 创建的 Python 科学计算核心库，提供了众多数学算法与函数。



Pandas 是基于 NumPy 创建的 Python 库，为 Python 提供了易于使用的的数据结构和数据分析工具。使用以下语句导入：

```
import pandas as pd
```



Matplotlib 是 Python 的二维绘图库，用于生成符合出版质量或跨平台交互环境的各类图形。

```
import matplotlib.pyplot as plt
```



Seaborn 是基于 matplotlib 开发的高阶 Python 数据可视图库，用于绘制优雅、美观的统计图形。使用下列别名导入该库：

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```



Bokeh 是 Python 的交互式可视图库，用于生成在浏览器里显示的大规模数据集高性能可视图。Bokeh 的中间层通用 **bokeh.plotting** 界面主要为两个组件：数据与图示例。

```
from bokeh.plotting import figure
```

```
from bokeh.io import output_file, show
```



PySpark 是 Spark 的 Python API，允许 Python 调用 Spark 编程模型。Spark SQL 是 Apache Spark 处理结构化数据模块。

AI 垂直领域工具库速查表



Scikit-learn 是开源的 Python 库，通过统一的界面实现机器学习、预处理、交叉验证及可视化算法。



Keras 是强大、易用的深度学习库，基于 Theano 和 TensorFlow 提供了高阶神经网络 API，用于开发和评估深度学习模型。



“TensorFlow™ is an open source software library for numerical computation using data flow graphs.” **TensorFlow** 是 Google 公司开发的机器学习架构，兼顾灵活性和扩展性，既适合用于工业生产也适合用于科学研究。



PyTorch 是 Facebook 团队 2017 年初发布的深度学习框架，有利于研究人员、爱好者、小规模项目等快速搞出原型。**PyTorch** 也是 Python 程序员最容易上手的深度学习框架。



Hugging Face 以开源的 NLP 预训练模型库 **Transformers** 而广为人知，目前 GitHub Star 已超过 54000+。**Transformers** 提供 100+ 种语言的 32 种预训练语言模型，简单，强大，高性能，是新手入门的不二选择。



OpenCV 是一个跨平台计算机视觉库，由 C 函数 /C++ 类构成，提供了 Python、MATLAB 等语言的接口。**OpenCV** 实现了图像处理和计算机视觉领域的很多通用算法。

编程语言速查表



SQL 是管理关系数据库的结构化查询语言，包括数据的增删查改等。作为数据分析的必备技能、岗位 JD 的重要关键词，SQL 是技术及相关岗位同学一定要掌握的语言。



Python 编程语言简洁快速、入门简单且功能强大，拥有丰富的第三方库，已经成为大数据和人工智能领域的主流编程语言。

More...

AI 知识技能速查表



Jupyter Notebook 交互式计算环境，支持运行 40+ 种编程语言，可以用来编写漂亮的交互式文档。这个教程把常用的基础功能讲解得很清楚，对新手非常友好。



正则表达式 非常强大，能匹配很多规则的文本，常用于文本提取和爬虫处理。这也是一门令人难以捉摸的语言，字母、数字和符号堆在一起，像极了“火星文”。

More...



ShowMeAI 速查表 (©2021)

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

数据科学工具库速查表

扫码回复“数据科学”

获取最新全套速查表

AI 垂直领域工具库速查表

扫码回复“工具库”

获取最新全套速查表

编程语言速查表

扫码回复“编程语言”

获取最新全套速查表

AI 知识技能速查表

扫码回复“知识技能”

获取最新全套速查表