scikit-learn基础课程

目录

scikit-learn简介

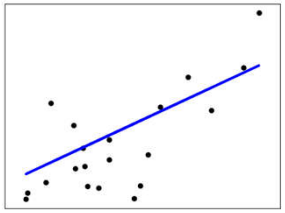
scikit-learn是python实现的机器学习算法库，它具有各种分类，回归和聚类算法，包括支持向量机，随机森林，梯度提升，k均值和DBSCAN，并且旨在与Python数值科学图书馆NumPy和SciPy。

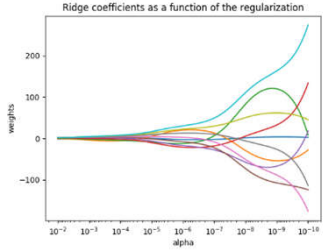
David Cournapeau

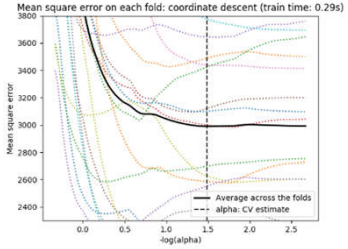
scikit-learn的优点有哪些？

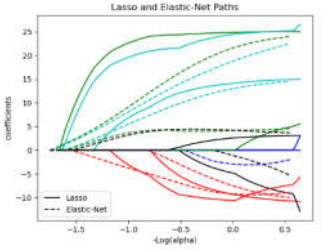
- 与机器学习模型的一致界面
- 提供许多调整参数，但具有合理的默认值
- 含特殊的文件
- 丰富的配套任务功能。

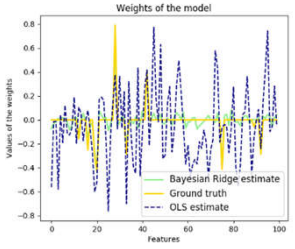
https://www.scikitlearn.com.cn/











目录

机器学习简单流程

- 自带的小数据集
- 可在线下载的数据集
- 算机生成的数据集

数据集使用 1

- 降维模型
- 模型训练

数据降维 2

- 留出法
- 交叉验证

数据集划分 3

- 调用模型
- 训练模型
- 模型预测

模型调用 4

- 分类结果评价
- 聚类结果评价
- 回归结果评价

结果评价 5

- 过程可视化
- 结果可视化

可视化 6

scikit-learn自带数据集

scikit-learn 的数据集有多个种类，其中常用的是它自带的小数据集

数据集	函数	适用范围
波士顿房价数据集	datasets.load_boston([return_X_y])	经典的用于回归任务的数据集
乳腺癌数据集	datasets.load_breast_cancer([return_X_y])	简单经典的用于二分类任务的数据集
糖尿病数据集	datasets.load_diabetes([return_X_y])	经典的用于回归的数据集，
手写数字数据集	datasets.load_digits([n_class, return_X_y])	用于分类任务或者降维任务的数据集
体能训练数据集	datasets.load_linnerud([return_X_y])	经典的用于多变量回归任务的数据集。
鸢尾花数据集	datasets.load_iris([return_X_y])	用于分类任务的数据集

鸢尾花数据调用示例

```
from sklearn.datasets import load_iris
data=load_iris( )
```

其他数据集示例

```
from  sklearn.datasets  import  fetch_olivetti_faces          #下载大型数据集
from  sklearn.datasets  import  make_swiss_roll               #生成数据集
```

scikit-learn数据降维

sk-learn的降维算法都被包括在模块decomposition中，在此模块中可以实现主成分分析、因子分析、独立成分分析、字典学习、高级矩阵分解以及其他矩阵分解

PCA降维示例

model=sklearn.decomposition.PCA(*n_components=None, copy=True, whiten=False*)

n_components：赋值为int，比如n_components=1，将把原始数据降到一个维度。

copy:表示是否在运行算法时，将原始训练数据复制一份。

whiten:白化，使得每个特征具有相同的方差。

X_LessDim=model.fit_transform(X)

scikit-learn数据预处理

sk-learn的Preprocessing包含了简单的预处理和标准化函数，在此模块中可以实现简单的数据预处理，如：独热编码转换、数据二值化等操作，还可以实现0-1标准化、最大最小标准化等数据标准化转换操作。

标准化转换示例

```
from sklearn. preprocessing import Normalizer
model=Normalizer()
X_stand=model.transform(X)
```

scikit-learn数据集划分

scikit-learn支持多种种数据集划分的方法，这里以留出法和交叉验证法为例。

留出法

```
from sklearn.model_selection import train_test_split
X_tr,X_te,Y_tr,Y_te=train_test_split(X,Y,test_size=0.2)
```

test_size：控制测试集的比例

交叉验证

```
from sklearn.model_selection import KFold
kf = KFold(n_splits=4)
for train, test in kf.split(X, Y):
    print(train, test)
```

scikit-learn模型调用

scikit-learn的常规模型调用一般分三步

1. 调用模型

```
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier()
```

2. 代入数据训练

```
model.fit(X_tr，Y_tr)
```

3. 模型预测

```
Y_pre=model.predict(X_te)/model. predict_proba(X_te)
```

***对于部分模型可以直接输出评价结果**

```
model.score(X_te，  Y_te)
```

scikit-learn结果评价

分类结果评价函数

from sklearn.metrics import

错误率

accuracy_score

混淆矩阵

recall_score

precision_score

f1_score

综合性方法

classification_report

回归结果评价函数

from sklearn.metrics import

MAE

mean_absolute_error

MSE

mean_squared_error

R^2

r2_score

聚类结果评价函数

SSE

使用inertia属性获得

km.intertia_

互信息（NMI）

normalized_mutual_info_score

scikit-learn过程可视化（以决策树为例）

决策树的可视化需要用到graphviz和pydotplus，然而graphviz除了在python里pip安装模块，还需要在电脑中安装graphviz软件。

首先载入所需的模块：

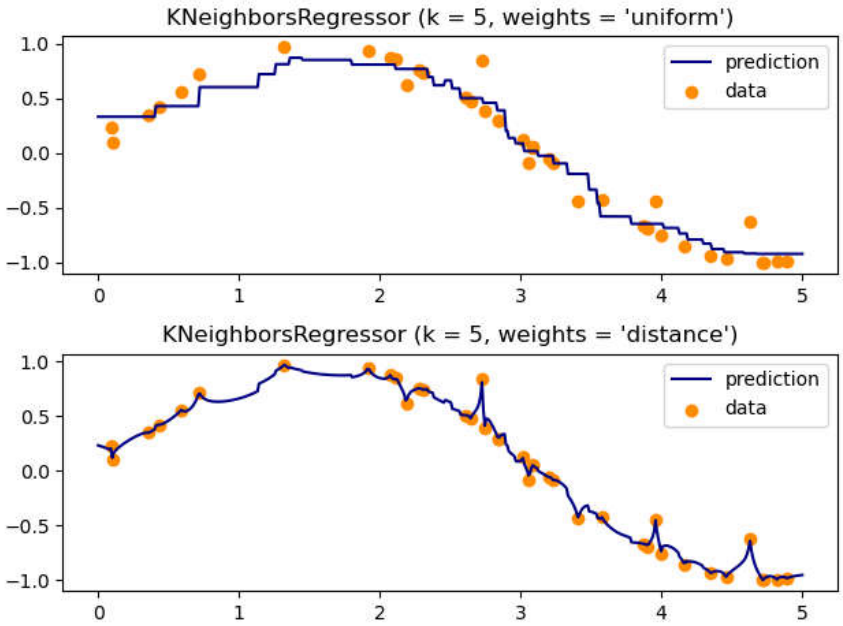
```
from sklearn.tree import export_graphviz
import pydotplus
```

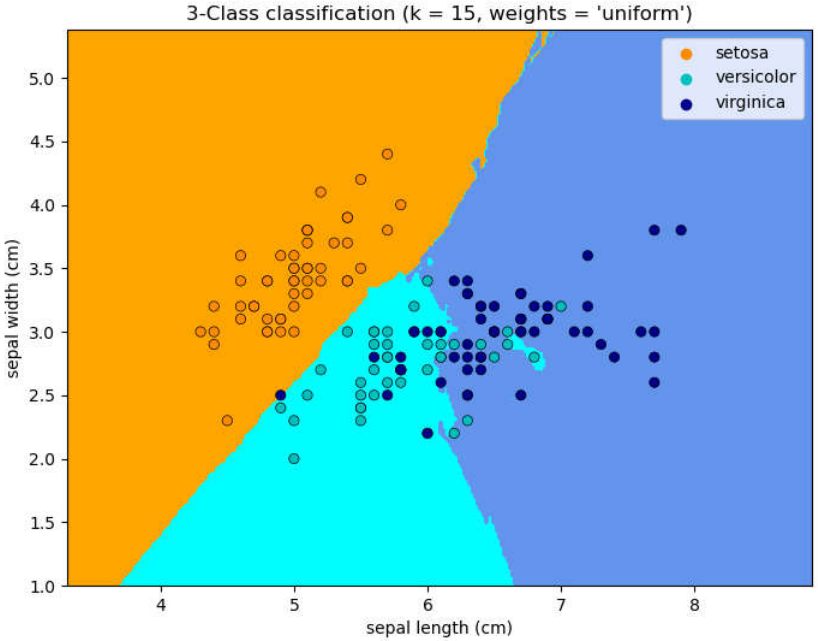
生成模型后即可使用上述模块产生需要格式的决策树图片

```
dot_data = tree.export_graphviz(决策树模型, out_file=None)
graph = pydotplus.graph_from_dot_data(dot_data)
graph.write_png("tree.png")     # 生成png文件
graph.write_jpg("tree.jpg")     # 生成jpg文件
graph.write_pdf("tree.pdf")     # 生成pdf文件
```

scikit-learn结果可视化

机器学习处理后的结果可以通过可视化的手段将其展现，以方便结果的直观呈现。





Thank you!