**scikit-learn——特征选择**

**任务目标**

1. 了解特征选择的方法

2.熟练掌握sklearn.feature中特征选择的函数使用

**相关知识**

sklearn.feature\_selection

特征选择的原因如下：

(1)降低复杂度

(2)降低噪音

(3)增加模型可读性

* VarianceThreshold： 删除特征值的方差达不到最低标准的特征
* SelectKBest： 返回k个最佳特征
* SelectPercentile： 返回表现最佳的前r%个特征

单个特征和某一类别之间相关性的计算方法有很多。最常用的有卡方检验（χ2）。其他方法还有互信息和信息熵。

* chi2： 卡方检验（χ2）

[](https://www.ipieuvre.com/doc/exper/36d18743-91ad-11e9-beeb-00215ec892f4/img/02.png)

**任务内容**

练习scikit\_learn中特征选择的函数或方法。

**系统环境**

Python

Jupyter

**任务步骤**

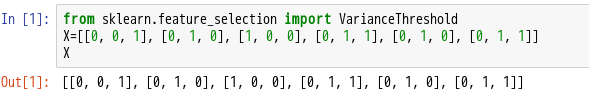
1.打开终端模拟器，在命令行输入ipython notebook

2.新建一个ipynb文件，用于编写并执行代码。

去掉取值变化小的特征

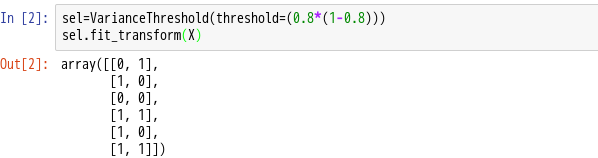
3.导入sklearn库中feature\_select模块的VarianceThreshold类，创建一个二维列表X。

1. from sklearn.feature\_selection **import** VarianceThreshold
2. X=[[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
3. X



4.使用VarianceThreshold函数创建一个方差选择对象sel，设置阀值参数threshold=(0.8\*(1-0.8))，使用sel对象的fit\_transform方法，删除X中每个特征值的方差小于阀值的特征。

1. sel=VarianceThreshold(threshold=(0.8\*(1-0.8)))
2. sel.fit\_transform(X)



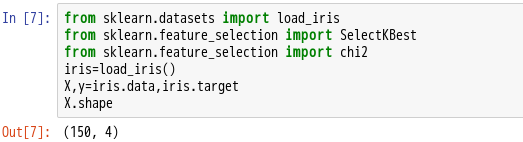
单变量特征选择

5.导入sklearn库中feature\_select模块的SelectKBest、chi2，导入sklearn库中datasets模块的load\_iris，

1. from sklearn.datasets **import** load\_iris
2. from sklearn.feature\_selection **import** SelectKBest
3. from sklearn.feature\_selection **import** chi2

6.使用load\_iris类创建一个iris数据的实例化对象iris，将iris对象的数据与标签分别赋值给X,y，并查看X的形状。

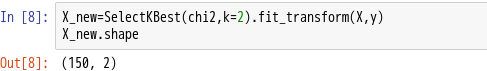
1. iris=load\_iris()
2. X,y=iris.data,iris.target
3. X.shape



单变量特征选择

7.使用SelectKBase函数创建一个自定义特征评分选择对象，设置评分参数score\_func=chi2，选择评分最高特征数量参数k=2，然后使用fir\_transform方法，通过卡方检验选择法选择X,y数据中评分最高的2的特征，并检查返回数据的形状。

1. X\_new=SelectKBest(chi2,k=2).fit\_transform(X,y)
2. X\_new.shape



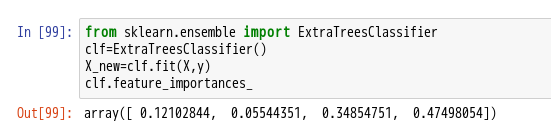
基于树的特征选择(Tree-based Feature Selection)

8.导入sklearn库中ensemble模块的ExtraTreesClassifier

1. from sklearn.ensemble **import** ExtraTreesClassifier

9.使用ExtraTreesClassifier函数创建一个基于树的特征选择对象clf，使用clf对象的fit方法通过X,y数据创建基于树的特征选择模型，查看每个特征的重要性.

1. clf=ExtraTreesClassifier()
2. X\_new=clf.fit(X,y)
3. clf.feature\_importances\_



递归特征消除

10.导入sklearn库中feature\_selection模块的RFE，导入sklearn库中datasets模块的 make\_friedman1，导入sklearn库中svm模块的 SVR，

1. from sklearn.datasets **import** make\_friedman1
2. from sklearn.feature\_selection **import** RFE
3. from sklearn.svm **import** SVR

11.使用make\_friendman1函数创建训练数据X，标签数据y，设置样本个数参数n\_samples=50，特征个数参数n\_features=10，随机状态参数random\_state=0。

1. X, y = make\_friedman1(n\_samples=50, n\_features=10, random\_state=0)

12.使用SVR函数创建一个支持向量机回归模型实例对象，设置核函数参数为kernel=linear，

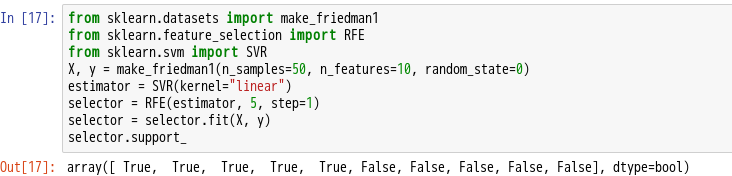
1. estimator=SVR(kernel='linear')

13.使用RFE函数创建一个递归消除对象selector，设置 估计量参数estimator=estimator，n\_features\_to\_select=5，使用对象的fit方法通过X,y数据创建递归特征消除模型。

1. selector = RFE(estimator, 5, step=1)
2. selector=selector.fit(X,y)

14.使用support\_获取训练数据X中特征是否被选择的布尔值。

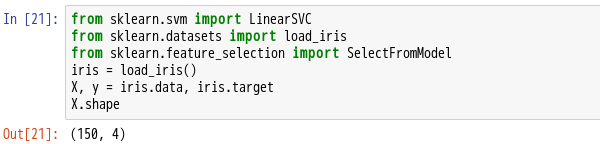
1. selector.support\_



自定义模型训练选择法

15.导入sklearn库中feature\_selection模块的SelectFromModel，导入sklearn库中datasets模块的 load\_iris，导入sklearn库中svm模块的LinearSVC，使用load\_iris类创建一个iris数据的实例化对象iris，将iris对象的数据与标签分别赋值给X,y，并查看X的形状。

1. from sklearn.svm **import** LinearSVC
2. from sklearn.datasets **import** load\_iris
3. from sklearn.feature\_selection **import** SelectFromModel
4. iris = load\_iris()
5. X, y = iris.data, iris.target
6. X.shape



16.使用LinearSVC函数创建一个实例对象，设置误差项的惩罚参数C=0.01，正则化参数penalty='l1'，对偶参数dual=False，然后实验fit方法通过训练数据X，y创建线性支持向量机模型lsvc，然后使用SelectFromModel函数，设置自定义函数参数为lsvc，预适应模型是否会直接传递到构造函数中的参数prefit=True，创建自定义选择模型model，然后使用transform方法将X数据进行特征选择，返回结果X\_new，并查看X\_new的形状。

1. lsvc = LinearSVC(C=0.01, penalty="l1", dual=False).fit(X, y)
2. model = SelectFromModel(lsvc, prefit=True)
3. X\_new = model.transform(X)
4. X\_new.shape

