

jasonfreak

一个懒惰的人，总是想设计更智能的程序来避免做重复性工作

导航

博客园

首页

联系

订阅 XML

管理

统计信息

随笔 - 12

文章 - 0

评论 - 67

Trackbacks - 0

NEWS

昵称：jasonfreak

园龄：1年9个月

粉丝：191

关注：0

+加关注

搜索

找找我

谷歌搜索

我的标签

数据挖掘(8)

sklearn(5)

Python(3)

线性模型(3)

特征工程(2)

线性代数(2)

机器学习(2)

集成学习(2)

数据分析(2)

SciPy(1)

更多

随笔分类

代码发布(1)

环境搭建(1)

机器学习(2)

数据分析(2)

数据挖掘(8)

特征工程(2)

随笔档案

2016年11月 (1)

2016年7月 (3)

2016年6月 (4)

2016年5月 (2)

2016年4月 (2)

最新评论

1. Re:使用sklearn做单机特征工程
标准化是依照特征矩阵的列处理数据，归一化是依照特征矩阵的行处理数据，这个不太理解，博主可以解释下吗？

—Stone1111

2. Re:使用sklearn进行集成学习——理论
你好博主！我想问一下“在bagging和boosting框架中，通过计算基模型的期望和方差，我们可以得到模型整体的期望和方差。为了简化模型，我们假设基模型的权重、方差及两两间的相关系数相等。”这里接.....

使用sklearn做单机特征工程

目录

- 1 特征工程是什么？
- 2 数据预处理
 - 2.1 无量纲化
 - 2.1.1 标准化
 - 2.1.2 区间缩放法
 - 2.1.3 标准化与归一化的区别
 - 2.2 对定量特征二值化
 - 2.3 对定性特征哑编码
 - 2.4 缺失值计算
 - 2.5 数据变换
 - 2.6 回顾
- 3 特征选择
 - 3.1 Filter
 - 3.1.1 方差选择法
 - 3.1.2 相关系数法
 - 3.1.3 卡方检验
 - 3.1.4 互信息法
 - 3.2 Wrapper
 - 3.2.1 递归特征消除法
 - 3.3 Embedded
 - 3.3.1 基于惩罚项的特征选择法
 - 3.3.2 基于树模型的特征选择法
 - 3.4 回顾
- 4 降维
 - 4.1 主成分分析法（PCA）
 - 4.2 线性判别分析法（LDA）
 - 4.3 回顾
- 5 总结
- 6 参考资料

1 特征工程是什么？

有这么一句话在业界广泛流传：数据和特征决定了机器学习上限，而模型和算法只是逼近这个上限而已。那特征工程到底是什么呢？顾名思义，其本质是一项工程活动，目的是最大限度地从原始数据中提取特征以供算法和模型使用。通过总结和归纳，人们认为特征工程包括以下方面：



3. Re:使用Python进行描述性统计
很好，很清晰，赞！

4. Re:使用sklearn优雅地进行数据挖掘
@会飞的蜗牛引用@魔灵幽亭在你的数据集DataFrame上加一句df = df.fillna(0)...

5. Re:谁动了我的特征？——sklearn
特征转换行为全记录
@hnxsm这个调的哪里的...

--implus

--iuwai

--liuer2009

--hustenn

阅读排行榜

1. 使用sklearn做单机特征工程(47866)
2. 使用sklearn优雅地进行数据挖掘(42921)
3. 使用Python进行描述性统计(34743)
4. 使用sklearn进行集成学习——理论(22104)
5. 使用sklearn进行集成学习——实践(22084)

评论排行榜

1. 使用sklearn优雅地进行数据挖掘(21)
2. 使用sklearn做单机特征工程(18)
3. 使用sklearn进行集成学习——理论(9)
4. 虎扑论坛装备区到底有没有李宁水军？——论坛水军发现实践(6)
5. 使用sklearn进行集成学习——实践(3)

推荐排行榜

1. 使用sklearn做单机特征工程(20)
2. 使用sklearn优雅地进行数据挖掘(17)
3. 使用sklearn进行集成学习——理论(8)
4. 使用Python进行描述性统计(6)
5. easyconf——基于AngularJS的配置管理系统开发框架(5)

特征处理是特征工程的核心部分，sklearn提供了较为完整的特征处理方法，包括数据预处理，特征选择，降维等。首次接触到sklearn，通常会被其丰富且方便的算法模型库吸引，但是这里介绍的特征处理库也十分强大！

本文中使用了sklearn中的IRIS（鸢尾花）数据集来对特征处理功能进行说明。IRIS数据集由Fisher在1936年整理，包含4个特征（Sepal.Length（花萼长度）、Sepal.Width（花萼宽度）、Petal.Length（花瓣长度）、Petal.Width（花瓣宽度）），特征值都为正浮点数，单位为厘米。目标值为鸢尾花的分类（Iris Setosa（山鸢尾）、Iris Versicolour（杂色鸢尾）、Iris Virginica（维吉尼亚鸢尾））。导入IRIS数据集的代码如下：

```
1 from sklearn.datasets import load_iris
2
3 #导入IRIS数据集
4 iris = load_iris()
5
6 #特征矩阵
7 iris.data
8
9 #目标向量
10 iris.target
```

2 数据预处理

通过特征提取，我们能得到未经处理的特征，这时的特征可能有以下问题：

- 不属于同一量纲：即特征的规格不一样，不能够放在一起比较。无量纲化可以解决这一问题。
- 信息冗余：对于某些定量特征，其包含的有效信息为区间划分，例如学习成绩，假若只关心“及格”或“不及格”，那么需要将量化的得分，转换成“1”和“0”表示及格和不及格。二值化可以解决这一问题。
- 定性特征不能直接使用：某些机器学习算法和模型只能接受定量特征的输入，那么需要将定性特征转换为定量特征。最简单的方式是为每一种定性值指定一个定量值，但是这种方式过于灵活，增加了调参的工作。通常使用哑编码的方式将定性特征转换为定量特征：假设有N种定性值，则将这一个特征扩展为N种特征，当原始特征值为第i种定性值时，第i个扩展特征赋值为1，其他扩展特征赋值为0。哑编码的方式相比直接指定的方式，不用增加调参的工作，对于线性模型来说，使用哑编码后的特征可达到非线性的效果。
- 存在缺失值：缺失值需要补充。
- 信息利用率低：不同的机器学习算法和模型对数据中信息的利用是不同的，之前提到在线性模型中，使用对定性特征哑编码可以达到非线性的效果。类似地，对定量变量多项式化，或者进行其他的转换，都能达到非线性的效果。

我们使用sklearn中的preprocessing库来进行数据预处理，可以覆盖以上问题的解决方案。

2.1 无量纲化

无量纲化使不同规格的数据转换到同一规格。常见的无量纲化方法有标准化和区间缩放法。标准化的前提是特征值服从正态分布，标准化后，其转换成标准正态分布。区间缩放法利用了边界值信息，将特征的取值区间缩放到某个特点的范围，例如[0, 1]等。

2.1.1 标准化

标准化需要计算特征的均值和标准差，公式表达为：

$$x' = \frac{x - \bar{x}}{s}$$

第2页 共10页

2018/1/22 上午9:07

使用preprocessing库的StandardScaler类对数据进行标准化的代码如下：

```
1 from sklearn.preprocessing import StandardScaler
2
3 #标准化，返回值为标准化后的数据
4 StandardScaler().fit_transform(iris.data)
```

2.1.2 区间缩放法

区间缩放法的思路有多种，常见的一种为利用两个最值进行缩放，公式表达为：

$$x' = \frac{x - Min}{Max - Min}$$

使用preprocessing库的MinMaxScaler类对数据进行区间缩放的代码如下：

```
1 from sklearn.preprocessing import MinMaxScaler
2
3 #区间缩放，返回值为缩放到[0, 1]区间的数据
4 MinMaxScaler().fit_transform(iris.data)
```

2.1.3 标准化与归一化的区别

简单来说，标准化是依照特征矩阵的列处理数据，其通过求z-score的方法，将样本的特征值转换到同一量纲下。归一化是依照特征矩阵的行处理数据，其目的在于样本向量在点乘运算或其他核函数计算相似性时，拥有统一的标准，也就是说都转化为“单位向量”。规则为l2的归一化公式如下：

$$x' = \frac{x}{\sqrt{\sum_j x[j]^2}}$$

使用preprocessing库的Normalizer类对数据进行归一化的代码如下：

```
1 from sklearn.preprocessing import Normalizer
2
3 #归一化，返回值为归一化后的数据
4 Normalizer().fit_transform(iris.data)
```

2.2 对定量特征二值化

定量特征二值化的核心在于设定一个阈值，大于阈值的赋值为1，小于等于阈值的赋值为0，公式表达如下：

$$x' = \begin{cases} 1, x > threshold \\ 0, x \leq threshold \end{cases}$$

使用preprocessing库的Binarizer类对数据进行二值化的代码如下：

```
1 from sklearn.preprocessing import Binarizer
2
3 #二值化，阈值设置为3，返回值为二值化后的数据
4 Binarizer(threshold=3).fit_transform(iris.data)
```


2.3 对定性特征哑编码

由于IRIS数据集的特征皆为定量特征，故使用其目标值进行哑编码（实际上是不需要的）。使用preprocessing库的OneHotEncoder类对数据进行哑编码的代码如下：

```
1 from sklearn.preprocessing import OneHotEncoder
2
3 #哑编码，对IRIS数据集的目标值，返回值为哑编码后的数据
4 OneHotEncoder().fit_transform(iris.target.reshape((-1,1)))
```

2.4 缺失值计算

由于IRIS数据集没有缺失值，故对数据集新增一个样本，4个特征均赋值为NaN，表示数据缺失。使用preprocessing库的Imputer类对数据进行缺失值计算的代码如下：

```

1 from numpy import vstack, array, nan
2 from sklearn.preprocessing import Imputer
3
4 #缺失值计算，返回值为计算缺失值后的数据
5 #参数missing_value为缺失值的表示形式，默认为NaN
6 #参数strategy为缺失值填充方式，默认为mean（均值）
7 Imputer().fit_transform(vstack((array([nan, nan, nan, nan]), iris.data)))
```



2.5 数据变换

常见的数据变换有基于多项式的、基于指数函数的、基于对数函数的。4个特征，度为2的多项式转换公式如下：

$$\begin{aligned} &(x_1', x_2', x_3', x_4', x_5', x_6', x_7', x_8', x_{10}', x_{11}', x_{12}', x_{13}', x_{14}', x_{15}') \\ &= (1, x_1, x_2, x_3, x_4, x_1^2, x_1 * x_2, x_1 * x_3, x_1 * x_4, x_2^2, x_2 * x_3, x_2 * x_4, x_3^2, x_3 * x_4, x_4^2) \end{aligned}$$

使用preprocessing库的PolynomialFeatures类对数据进行多项式转换的代码如下：

```
1 from sklearn.preprocessing import PolynomialFeatures
2
3 #多项式转换
4 #参数degree为度，默认值为2
5 PolynomialFeatures().fit_transform(iris.data)
```

基于单变元函数的数据变换可以使用一个统一的方式完成，使用preprocessing库的FunctionTransformer对数据进行对数函数转换的代码如下：

```
1 from numpy import log1p
2 from sklearn.preprocessing import FunctionTransformer
3
4 #自定义转换函数为对数函数的数据变换
5 #第一个参数是单变元函数
6 FunctionTransformer(log1p).fit_transform(iris.data)
```

2.6 回顾

类	功能	说明
StandardScaler	无量纲化	标准化，基于特征矩阵的列，将特征值转换至服从标准正态分布
MinMaxScaler	无量纲化	区间缩放，基于最大最小值，将特征值转换到[0, 1]区间上
Normalizer	归一化	基于特征矩阵的行，将样本向量转换为“单位向量”
Binarizer	二值化	基于给定阈值，将定量特征按阈值划分
OneHotEncoder	哑编码	将定性数据编码为定量数据
Imputer	缺失值计算	计算缺失值，缺失值可填充为均值等
PolynomialFeatures	多项式数据转换	多项式数据转换
FunctionTransformer	自定义单元数据转换	使用单变元的函数来转换数据

3 特征选择

当数据预处理完成后，我们需要选择有意义的特征输入机器学习的算法和模型进行训练。通常来说，从两个方面考虑来选择特征：

- 特征是否发散：如果一个特征不发散，例如方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。
- 特征与目标的相关性：这点比较显见，与目标相关性高的特征，应当优先选择。除方差法外，本文介绍的其他方法均从相关性考虑。

根据特征选择的形式又可以将特征选择方法分为3种：

- Filter：过滤法，按照发散性或者相关性对各个特征进行评分，设定阈值或者待选择阈值的个数，选择特征。
- Wrapper：包装法，根据目标函数（通常是预测效果评分），每次选择若干特征，或者排除若干特征。
- Embedded：嵌入法，先使用某些机器学习的算法和模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。类似于Filter方法，但是是通过训练来确定特征的优劣。

我们使用sklearn中的feature_selection库来进行特征选择。

3.1 Filter

3.1.1 方差选择法

使用方差选择法，先要计算各个特征的方差，然后根据阈值，选择方差大于阈值的特征。使用feature_selection库的VarianceThreshold类来选择特征的代码如下：

```
1 from sklearn.feature_selection import VarianceThreshold
```

```
2
3 #方差选择法，返回值为特征选择后的数据
4 #参数threshold为方差的阈值
5 VarianceThreshold(threshold=3).fit_transform(iris.data)
```

3.1.2 相关系数法

使用相关系数法，先要计算各个特征对目标值的相关系数以及相关系数的P值。用feature_selection库的SelectKBest类结合相关系数来选择特征的代码如下：

```

1 from sklearn.feature_selection import SelectKBest
2 from scipy.stats import pearsonr
3
4 #选择K个最好的特征，返回选择特征后的数据
5 #第一个参数为计算评估特征是否好的函数，该函数输入特征矩阵和目标向量，输出二元组（评分，P值）的数组，数组第i项为第i个特征的评分和P值。在此定义为计算相关系数
6 #参数k为选择的特征个数
7 SelectKBest(lambda X, Y: array(map(lambda x: pearsonr(x, Y), X.T)).T, k=2).fit_transform(iris.data, iris.target)

```

3.1.3 卡方检验

经典的卡方检验是检验定性自变量对定性因变量的相关性。假设自变量有N种取值，因变量有M种取值，考虑自变量等于i且因变量等于j的样本频数的观察值与期望的差距，构建统计量：

$$\chi^2 = \sum \frac{(A - E)^2}{E}$$

这个统计量的含义简而言之就是自变量对因变量的相关性。用feature_selection库的SelectKBest类结合卡方检验来选择特征的代码如下：



```
1 from sklearn.feature_selection import SelectKBest
2 from sklearn.feature_selection import chi2
3
4 #选择K个最好的特征，返回选择特征后的数据
5 SelectKBest(chi2, k=2).fit_transform(iris.data, iris.target)
```

3.1.4 互信息法

经典的互信息也是评价定性自变量对定性因变量的相关性的，互信息计算公式如下：

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$


为了处理定量数据，最大信息系数法被提出，使用feature_selection库的SelectKBest类结合最大信息系数法来选择特征的代码如下：

```

1 from sklearn.feature_selection import SelectKBest
2 from minepy import MINE
3
4 #由于MINE的设计不是函数式的，定义mic方法将其为函数式的，返回一个二元组，二元组的第2项设置成固定的P值0.5
5 def mic(x, y):
6     m = MINE()
7     m.compute_score(x, y)
8     return (m.mic(), 0.5)
9
10 #选择K个最好的特征，返回特征选择后的数据
11 SelectKBest(lambda X, Y: array(map(lambda x: mic(x, Y), X.T)).T, k=2).fit_transform(iris.data, iris.target)

```

3.2 Wrapper

3.2.1 递归特征消除法

递归消除特征法使用一个基模型来进行多轮训练，每轮训练后，消除若干权值系数的特征，再基于新的特征集进行下一轮训练。使用feature_selection库的RFE类来选择特征的代码如下：

```

1 from sklearn.feature_selection import RFE
2 from sklearn.linear_model import LogisticRegression
3
4 #递归特征消除法，返回特征选择后的数据
```

```
5 #参数estimator为基模型
6 #参数n_features_to_select为选择的特征个数
7 RFE(estimator=LogisticRegression(), n_features_to_select=2).fit_transform(iris.data, iris.target)
```

3.3 Embedded

3.3.1 基于惩罚项的特征选择法

使用带惩罚项的基模型，除了筛选出特征外，同时也进行了降维。使用feature_selection库的SelectFromModel类结合带L1惩罚项的逻辑回归模型，来选择特征的代码如下：

```
1 from sklearn.feature_selection import SelectFromModel
2 from sklearn.linear_model import LogisticRegression
3
4 #带L1惩罚项的逻辑回归作为基模型的特征选择
5 SelectFromModel(LogisticRegression(penalty="l1", C=0.1)).fit_transform(iris.data, iris.target)
```

L1惩罚项降维的原理在于保留多个对目标值具有同等相关性的特征中的一个，所以没选到的特征不代表不重要。故，可结合L2惩罚项来优化。具体操作为：若一个特征在L1中的权值为1，选择在L2中权值差别不大且在L1中权值为0的特征构成同类集合，将这一集合中的特征平分L1中的权值，故需要构建一个新的逻辑回归模型：

View Code

使用feature_selection库的SelectFromModel类结合带L1以及L2惩罚项的逻辑回归模型，来选择特征的代码如下：

```
1 from sklearn.feature_selection import SelectFromModel
2
3 #带L1和L2惩罚项的逻辑回归作为基模型的特征选择
4 #参数threshold为权值系数之差的阈值
5 SelectFromModel(LR(threshold=0.5, C=0.1)).fit_transform(iris.data, iris.target)
```

3.3.2 基于树模型的特征选择法

树模型中GBDT也可用来作为基模型进行特征选择，使用feature_selection库的SelectFromModel类结合GBDT模型，来选择特征的代码如下：

```
1 from sklearn.feature_selection import SelectFromModel
2 from sklearn.ensemble import GradientBoostingClassifier
3
4 #GBDT作为基模型的特征选择
5 SelectFromModel(GradientBoostingClassifier()).fit_transform(iris.data, iris.target)
```

3.4 回顾

类	所属方式	说明
VarianceThreshold	Filter	方差选择法
SelectKBest	Filter	可选关联系数、卡方校验、最大信息系数作为得分计算的方法
RFE	Wrapper	递归地训练基模型，将权值系数较小的特征从特征集合中消除
SelectFromModel	Embedded	训练基模型，选择权值系数较高的特征

4 降维

当特征选择完成后，可以直接训练模型了，但是可能由于特征矩阵过大，导致计算量大，训练时间长的问题，因此降低特征矩阵维度也是必不可少的。常见的降维方法除了以上提到的基于L1惩罚项的模型以外，另外还有主成分分析法（PCA）和线性判别分析（LDA），线性判别分析本身也是一个分类模型。PCA和LDA有很多的相似点，其本质是要将原始的样本映射到维度更低的样本空间中，但是PCA和LDA的映射目标不一样：PCA是为了让映射后的样本具有最大的发散性；而LDA是为了让映射后的样本有最好的分类性能。所以说PCA是一种无监督的降维方法，而LDA是一种有监督的降维方法。

4.1 主成分分析法（PCA）

使用decomposition库的PCA类选择特征的代码如下：

```
1 from sklearn.decomposition import PCA
2
3 #主成分分析法，返回降维后的数据
4 #参数n_components为主成分数目
5 PCA(n_components=2).fit_transform(iris.data)
```

4.2 线性判别分析法（LDA）

使用lda库的LDA类选择特征的代码如下：

```
1 from sklearn lda import LDA
2
3 #线性判别分析法，返回降维后的数据
4 #参数n_components为降维后的维数
5 LDA(n_components=2).fit_transform(iris.data, iris.target)
```

4.3 回顾

库	类	说明
decomposition	PCA	主成分分析法
lda	LDA	线性判别分析法

5 总结

再让我们回归一下本文开始的特征工程的思维导图，我们可以使用sklearn完成几乎所有特征处理的工作，而且不管是数据预处理，还是特征选择，抑或降维，它们都是通过某个类的方法fit_transform完成的，fit_transform要不只带一个参数：特征矩阵，要不带两个参数：特征矩阵加目标向量。这些难道都是巧合吗？还是故意设计成这样？方法fit_transform中有fit这一单词，它和训练模型的fit方法有关联吗？接下来，我将在《使用sklearn优雅地进行数据挖掘》中阐述其中的奥妙！

6 参考资料

- 1. FAQ: What is dummy coding?
- 2. IRIS（鸢尾花）数据集
- 3. 卡方检验
- 4. 干货：结合Scikit-learn介绍几种常用的特征选择方法
- 5. 机器学习中，有哪些特征选择的工程方法？
- 6. 机器学习中的数学(4)-线性判别分析（LDA），主成分分析(PCA)

分类: 特征工程

标签: Python, sklearn, 特征工程

好文要顶

关注我

收藏该文

jasonfreak

关注 - 0

粉丝 - 191

+加关注

20

2

« 上一篇：使用Python进行描述性统计
» 下一篇：使用sklearn优雅地进行数据挖掘

posted on 2016-05-02 17:41 jasonfreak 阅读(47867) 评论(18) 编辑 收藏

Feedback

#1楼 2016-05-03 20:16 Charlotte77

哈哈，我就知道是iris 数据

支持(0) 反对(0)

#2楼 2016-06-22 07:49 罗兵

好文！拜读！感谢！

支持(0) 反对(0)

#3楼 2016-06-24 16:04 Lydon

期待楼主大作。谢谢分享！

支持(0) 反对(0)

#4楼 2016-06-25 07:28 罗兵

博主，感谢写出这么好的文章！

另，我能否转载这篇博文？

支持(0) 反对(0)

#5楼[楼主] 2016-06-25 07:32 jasonfreak

@ 罗兵

可以的，互相学习！

支持(0) 反对(0)

#6楼 2016-08-13 12:15 丁磊-ml

大神，你好！！！

发现，虽然你特征工程的方法很全，但没有每个处理方法适用于哪种问题的介绍？？？

是不是在实战中根据自己的经验来尝试？？？从而得到自己的提升

支持(0) 反对(0)

#7楼[楼主] 2016-08-13 12:30 jasonfreak

@ 丁磊-ml

您好，本文只介绍了常用的特征处理方法及其sklearn实现，以及同类方法的简单的比较，例如：“PCA是为了让映射后的样本具有最大的发散性；而LDA是为了让映射后的样本有最好的分类性能”。当然，还有更多的特征处理办法和技巧，这就说来话长啦，哈哈哈。

我也觉得需要通过实践来加深对理论的理解，试错是少不了的过程。期待您在学习和实践过程中分享关于特征工程方面更深入的经验 and 知识！

支持(0) 反对(0)

#8楼 2016-08-13 14:32 丁磊-ml

@ jasonfreak

大神知道有哪些书是讲解 特征工程，数据预处理的吗？？？

很想读读那些书！！！！

支持(0) 反对(0)

#9楼 2016-12-27 22:37 魔灵幽亭

同样，感谢博主的文章！我在运行代码的时候，也遇到一个问题。用皮尔森来挑选变量处的代码，我这里需要改成：

1 | SelectKBest(lambda X, Y: tuple(map(tuple,array(list(map(lambda x:pearsonr(x, Y), X.T))).T)), k=2).fit_transform(iri:

需要将其转化成tuple才行。不知博主看后有没有建议。

至于，上面的map之后加个list是因为我用的Python3。

谢谢博主的文章。

支持(0) 反对(0)

#10楼 2017-01-10 21:56 会飞的蜗牛

请问博主，你这个博客的主题是哪里来的啊，好简洁，特别欣赏，能推荐一下吗

支持(0) 反对(0)

#11楼 2017-04-21 17:38 状语从句

拜读了，多谢

支持(0) 反对(0)

#12楼 2017-05-16 15:53 詹晴天

楼主，文章中给的有些链接打不开，尤其是idre.ucla的链接，不针对外界开放，请问有没有什么方法获得这些链接里的内容

支持(0) 反对(0)

#13楼 2017-09-24 17:26 哈士奇说喵

我觉得楼主的无量纲化这个描述不对，我个人的理解是，不同的特征之间做无量纲化，比如说身高1.7m和体重160斤这个两个特征做无量纲化，而楼主所说的max-min和z-score应该是归一化的两种形式，是对同一特征下，数值进行缩放

支持(0) 反对(0)

#14楼 2017-09-26 09:51 司徒道

@ 魔灵幽亭
同python3,这样就行

```
1 | SelectKBest(lambda X, Y: list(array([pearsonr(x, Y) for x in X.T]).T), k=2).fit_transform(iris.data, iris.target)
```

支持(0) 反对(0)

#15楼 2017-10-22 20:55 骑着蜗牛逛世界

good post

支持(0) 反对(0)

#16楼 2017-11-09 20:27 paris008

发散是啥？

支持(0) 反对(0)

#17楼 2017-12-01 10:35 xddexiaobaicai

我是python2 我改成这样可以执行了，我看了一下源码，好像有score就行了，SelectKBest(lambda X,Y:array(map(lambda x: pearsonr(x, Y), X.T)).T[0], k=3).fit_transform(iris.data, iris.target)
还请各位指教。

支持(0) 反对(0)

#18楼 2018-01-19 10:38 Stone1111

标准化是依照特征矩阵的列处理数据，归一化是依照特征矩阵的行处理数据，这个不太理解，博主可以解释下吗？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型工控、组态\仿真、建模CAD源码2018！
【推荐】腾讯云如何购买服务器更划算？



最新IT新闻:
· 高晓松：阿里与腾讯万达们讨论组建“好莱坞中国俱乐部”
· Rocket Lab成功发射第一颗卫星
· 淘宝卧榻之侧，岂容拼多多安睡？
· Docker日志的10大陷阱
· OpenSSL改变开发策略：转用GitHub issue讨论补丁
» 更多新闻...



最新知识库文章:

- 领域驱动设计在互联网业务开发中的实践
- 步入云计算
- 以操作系统的角度述说线程与进程
- 软件测试转型之路
- 门内门外看招聘
- » 更多知识库文章...

Powered by:

博客园

Copyright © jasonfreak