机器学习

吴恩达机器学习笔记 | null (sanzo.top)

2.6 Jupyter notebooks_哔哩哔哩_bilibili #上课视频

Coursera | Online Courses & Credentials From Top Educators. Join for Free

sklearn.preprocessing.StandardScaler — scikit-learn 1.3.0 文档

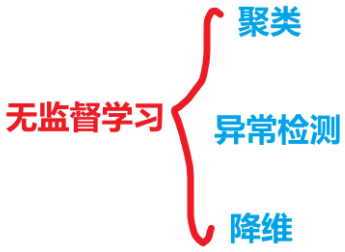
▼ CLASS1.0824:监督学习和无监督学习

监督学习：数据是标记好的，相当于知道了“答案”，那就可以根据答案不断地去纠正学习

回归算法：从无限多种可能的输出数字中预测数字，比如房价的预测

分类算法：结果类型可以是非数字的，预测输出结果是一组有限的可能输出类别，比如手写数字的识别

无监督学习：散养式，让它自己去探索哪些是有趣的（也就是算法），或者说某些数据中某种结构或模式，通过这样的方式，可以将数据分成不同的集群



聚类算法：能够自动去找到特征比如对每天新闻底部的相关推荐。找到和当前文章有相似词的并把他们归为一类，而人们事先并没有要求它根据什么什么来分，所以要求它在没有监督的情况下自行去找并分类。

线性回归模型：

根据训练集数据点的分布，尽可能地去拟合出一条直线。

输入为单一变量的称为单变量线性回归，多变量就是多元线性回归

▼ CLASS2.0829:数据预处理

1.数据标准化

Q1：我们为什么要对数据进行标准化？既它的作用是？最好是举例子说明，列点我很难直观感受

Q2：我们什么时候会选择使用数据标准化这个操作？

Q3：数据标准化大概的原理和流程，以及具体在代码应用中我们如何操作？即如何去实现它？

```
一些数据网站：UCI-----http://archive.ics.uci.edu/
SPSS一个数据统计软件
```

1.数据标准化

作用：将数据转换到0值附近，有负的，这一组数据变为均值为0，标准差为1的数据

实现原理：

标准差标准化也叫零均值标准化或分数标准化，是当前使用最广泛的数据标准化方法。经过该方法处理的数据均值为0，标准差为1，转化公式如下。

$$X^* = \frac{X - \bar{X}}{\delta}$$

其中$\bar{X}$为原始数据的均值，δ为原始数据的标准差。标准差标准化后的值区间不局限于[0,1]，并且存在负值。同时也不难发现，标准差标准化和离差标准化一样不会改变数据的分布情况。

代码实现：（利用sklearn库）

```
import numpy as np
import sklearn.preprocessing as sp

#   定义样本数据
raw_samples=np.array([[3.0,-1.0,2.0],
                      [0.0,4.0,-3.0],
                      [1.0,-4.0,2.0]]) #二维数据

std_samples = raw_samples.copy()
 #复制一个新数据，让原数据保持不变
std_samples = sp.scale (raw_samples)

输出结果：
==============================
[[ 1.33630621 -0.20203051  0.70710678]
 [-1.06904497  1.31319831 -1.41421356]
 [-0.26726124 -1.1111678   0.70710678]]
******************************
```

2 .范围缩放

作用：把每一列的数据缩放到0-1之间的数，并使得每一列当中最小的值缩放后为0，每一列当中最大的值缩放后变成1。结果是将原始数据的数值映射到[0,1]区间之间。

具体操作是：先减去最小值（变成0）,再对新的一列除以新的一列当中的最大值，完成缩放。

实现原理：

离差标准化是对原始数据的一种线性变换，结果是将原始数据的数值映射到[0,1]区间之间，转换公式为

$$X^* = \frac{X - min}{max - min}$$

其中max为样本数据的最大值，min为样本数据的最小值，max-min为极差。离差标准化保留了原始数据值之间的联系，是消除量纲和数据取值范围影响最简单的方法。

代码实现：（利用sklearn库）

MinMaxScaler详解（inverse_transform）_python_LuckyFucky-华为云开发者联盟 (csdn.net)

```
import numpy as np
import sklearn.preprocessing as sp

raw_samples = np.array([[1.0,4.0,9.0],
                        [10.0,16.0,29.0],
                        [6.0,19.0,67.0]])

#使用sklearn
#根据给定的范围创建一个范围缩放器对象
mms =sp.MinMaxScaler(feature_range=(0,1)) #定义对象（修改范围、观察现象）
#使用范围缩放器实现特征值范围缩放
mms_sapmles=mms.fit_transform(raw_samples)  #缩放
print(mms_sapmles)

输出结果：
==============================
[[0.         0.         0.        ]
 [1.         0.8        0.34482759]
 [0.55555556 1.         1.        ]]
******************************
```

2.数据范围缩放：

`3.归一化`

4.二值化——第五章，数据转化的哑变量处理

`np.round`

问题1：关于循环中只对循环

array，存放同一数据类型的数据，可以进行四则运算

CLASS2.0831:numpy基础用法

`1.array和列表异同`

1.数组相加是矩阵运算里的相加，列表相加是列表拼接

```
list=[1,3,4]
list2=list + list
```

```
list2=[1,3,4,1,3,4]
```

2.数组内的数据类型需要相同，而列表不用

数组和列表的区别是什么？数组中存储的数据元素类型必须是统一类型

优先级：字符串>浮点型>整数

2.array数组属性

1. 数组属性：

属性	说明
ndim	返回 int。表示数组的维数
shape	返回 tuple。表示数组的尺寸，对于 n 行 m 列的矩阵，形状为(n, m)
size	返回 int。表示数组的元素总数，等于数组形状的乘积
dtype	返回 data-type。描述数组中元素的类型
itemsize	返回 int。表示数组的每个元素的大小（以字节为单位）。

3.array生成特殊数组

```
#生成一个等差数组
arr1=np.arange(0,1,0.1).reshape(5,2)
arrange--->范围+步长
arr2=np.linspace(0,1,12)
输出结果：
==============================
array([[0. , 0.1],
       [0.2, 0.3],
       [0.4, 0.5],
       [0.6, 0.7],
       [0.8, 0.9]])
Out[41]: array([0.  , 0.25, 0.5 , 0.75, 1.  ])
==============================
#生成零阵
zero=np.zeros(4)
Out[43]: array([0., 0., 0., 0.])

zero=np.zeros((3,2))
```

```
array([[0., 0.],
       [0., 0.],
       [0., 0.]])

#生成全是1的阵
one = np.ones(4)
Out[47]: array([1., 1., 1., 1.])

one=np.ones((2,3))
array([[1., 1., 1.],
       [1., 1., 1.]])


#生成等比数列
arr3=np.logspace(0,2,10)
array([  1.  ,   1.67,   2.78, ...,  35.94,  59.95, 100.  ])
#0表示10的0次幂，2表示10的2次幂
#表示生成10^0~10^2之间10个数构成的等比数列

#生成单位对角阵
np.eye(3)
Out[52]:
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])

#生成对角阵
#列表里传入的是主元列表
np.diag([3,34,1])
Out[53]:
array([[ 3,  0,  0],
       [ 0, 34,  0],
       [ 0,  0,  1]])
```

4.随机数生成

```
data1=np.random.random(size=12)
Out[60]: array([0.83, 0.23, 0.34, ..., 0.55, 0.95, 0.59])
np.round(data1,2)         #保留两位小数
#size指定形状
#从[0,1）当中随机抽取12个小数出来，如果没有size就返回一个

np.random.rand(4,5)
#生成服从[0, 1)之间的均匀分布。
Out[62]:
array([[0.74, 0.19, 0.29, 0.45, 0.14],
       [0.44, 0.81, 0.05, 0.13, 0.55],
       [0.54, 0.3 , 0.29, 0.53, 0.58],
       [0.59, 0.68, 0.82, 0.75, 0.47]])
```

```
np.random.randn(4,5)
#生成服从标准正态分布的随机数
Out[70]:
array([[-2.32,  0.14, -1.94, -1.76, -0.61],
       [-1.1 ,  0.41,  1.08,  0.47,  0.85],
       [-1.91,  0.71,  1.11, -0.41,  0.21],
       [ 0.32,  1.82, -1.11,  0.34, -0.02]])

np.random.randint(1,100,(5,5))
#生成1-100之间5行5列的数组
Out[74]:
array([[40, 30,  9, 41, 42],
       [70, 32, 49, 67, 57],
       [41, 86,  7,  5,  5],
       [43, 15, 58, 99, 67],
       [62,  9, 69, 31, 37]])
```

5.数组的索引

一维数组的索引————按照列表索引的方法

```
#二维数据的索引
```

CLASS2.0905:numpy-文件读取

1. 保存一个numpy数组并读取

用np.save()保存成npy格式的文件，用np.load()来读取

```
import numpy as np
arr = np.arange(10).reshape((2,5))
np.save("test_1",arr)
load_arr=np.load("test_1.npy")
print(load_arr)
```

2.保存两个numpy数组并读取

用np.savez()保存成npz格式文件，用np.load()读取到的是存储地址，所以用“索引”的方式来读，先用arr.files查看两个小数组的名字，再“索引”读

```
import numpy as np
arr = np.arange(10).reshape((2,5))
np.save("test_1",arr)
```

```
load_arr=np.load("test_1.npy")
print(load_arr)

arr1= np.arange(20).reshape(4,5)
np.savez("test_2",arr,arr1)
load_new=np.load("test_2.npz")
print(load_new.files)
#['arr_0', 'arr_1']
print(load_new['arr_0'])
print(load_new['arr_1'])
```

3.实例–>复习访问数组（比如要取整组数据中的某些）

推荐收藏 | 最强(全) Matplotlib 可视化实操指南 (qq.com)

损失函数、误差函数：反映真实样本与模型计算的值之间的差异程度。

把损失函数优化到最小。

梯度（导数）：函数上升最快的方向。

通常采用梯度下降的方法来寻找损失函数的最小值。

参数更新法则。

符号保证调整方向永远是对的

0914课后作业：回去推导

CLASS2.0919:线性回归模型

梯度下降法：

详解梯度下降算法_梯度下降法-CSDN博客

疑问点一（时间：2023年9月20日16:21:14）:梯度，照理来说是和导数的概念是一致的，一元就是导数，二元或者多元就是偏导数，那么也就是某点切线的斜率，也就是该点变化最快的方向，那如果我建立一个三角形，就对不到了

重要：过程性考核

什么是线性回归？　利用线性模型做预测

什么情况下使用线性回归？　样本数据分布大概呈线性分布

如何来实现线性回归？

1.给出一个初始的线性模型　y=kx+b

2.计算损失函数

3.利用梯度下降法来优化线性模型的参数 k , b

4.利用优化好的线性模型做预测

非线性模型包括：多项式模型（多变量）

产生欠拟合的原因：模型复杂度不够，特征太少

产生过拟合的原因：模型过于复杂，特征太多，样本太少

正则化：有离群的样本，用正则化来弱化系数，考虑离群的样本但不完全考虑

超参数：训练之前提前设置好的数，比如学习率，相当于初始值呗

模型参数：模型里面的参数，通过学习得到的，y=wx+b中的w和b，通过学习率获得的。

CLASS2.1010:决策树

决策树:

分类：相同的输入相同的输出，同因同果，根据一系列的属性来判断，

回归：求均值，

如何选择特征：

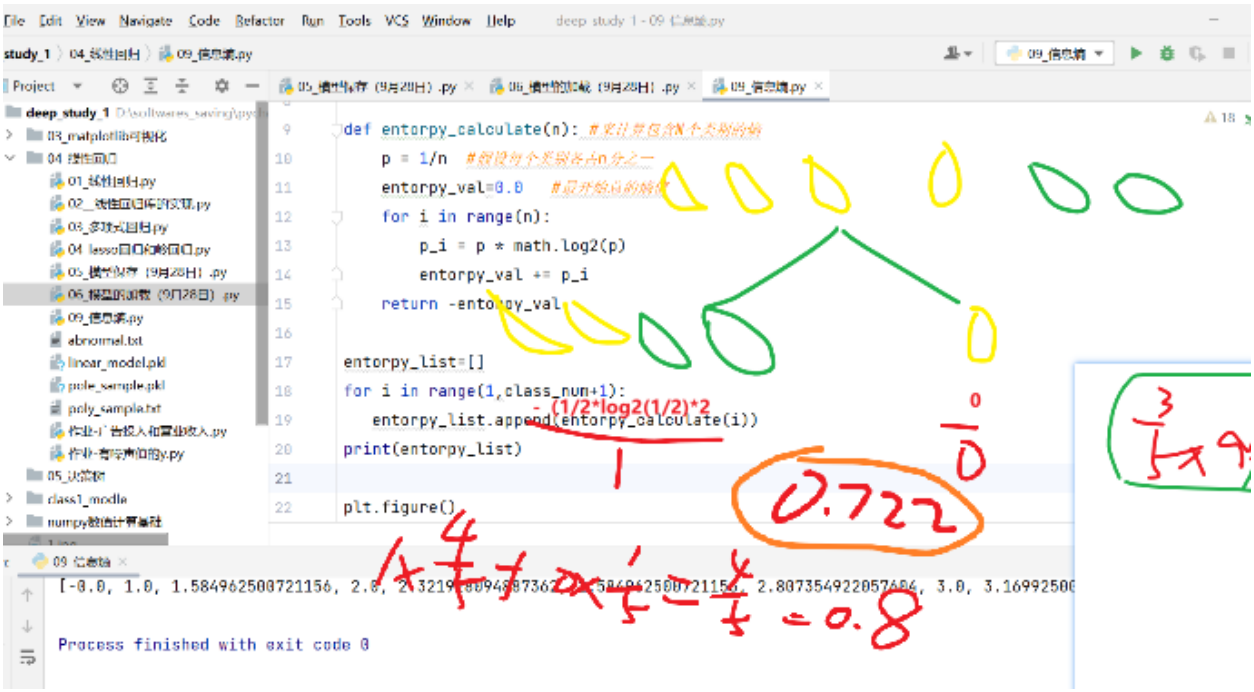
①信息熵，用来度量样本集合纯度的常用指标。若只有一个类别，熵为0，说明样本很纯

熵值越大说明它越混乱。

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x)$$

P（x）表示一个类别/概率

信息增益：



增益率：信息增益与熵值的比值

基尼系数：表示类别的确定性，值越大，越不确定

每个属性的排列

如何选取特征：决策树构建的每一步，都应该去挑选最优的特征，进行决策对数据集划分效果最好

如何停止分类子节点的构建：

1.当前节点所有样本都属于同一个类别，不需要再划分

2.当前属性集为空，或者所有样本取值相同，无法划分

3.当前节点样本数量少于指定数量（比如说2）

CLASS2.1012:波士顿房价

B401周四下节课波士顿房价的预测

深度学习入门——波士顿房价预测_BarbaraChow的博客-CSDN博客

随机森林

训练集：训练模型

测试集：评估模型

CLASS2.1024:决策树分类预测小汽车等级

机器学习-分类模型-决策树分类预测小汽车等级（7）_我欲乘风归去的博客-CSDN博客

数据集car -》 `low,low,4,2,med,med,unacc`

`low,low,4,2,med,high,unacc`

`low,low,4,2,big,low,unacc`

1.需要把字符串转换成数值型数据，这样才好塞到模型里面计算，使用便签编码

以列为一种类，转换成对应的便签编码

把它转换的规则给存下来（方便后续翻译还原回去） encoder1

先转置，变成行来操作。

```
low low low

low low low
4 4 4
2  2 2
med med big
med high low
unacc unacc unacc

编码
0,0,0
0,0,0
1,1,1
2,2,2
1,1,2
1,2,0
0,0,0

再转置回去
0,0,1,2,1,1,0
```

```
0,0,1,2,1,2,0
0,0,1,2,2,0,0
```

CLASS2.1026:支持向量机

二分类。SVM最优边界要求。

神经网络—（增加神经元）非线性模型

通过升维变换，从低纬度—》高纬度

线性可分和线性不可分。

将一维的数据空间升级为二维空间，来实现线性可分。`升维的本质就是增加特征`

2.核函数（用来实现升维）

高斯函数

支持向量机：（important，记住）

`1.支持向量机是一个二分类模型`

`2.寻找一个线性的分类边界（高纬度的话也是分割的超平面）`

`3.只考虑支持向量(考虑近的那些，离分类边界最近的样本）`

`4.要求支持向量到边界的距离要最大化。（距离最远）`

`5.对于线性不可分的问题，通过核函数进行升维变成线性可分的问题。`

`6.核函数种类：线性核函数，多项式核函数，径向基函数（rtf，高斯）`

对于少量的样本，效果会比较好

分类模型的评估，怎么来优化这个模型。调整超参数，网格搜索法，

CLASS2.1031:朴素贝叶斯

typora

应用：自然语言处理，垃圾邮件分类

1.一些概率的概念——联合概率，条件概率，事件的独立性，先验概率（事情未发生求发生的可能性）。后验概率　。

2.贝叶斯定理

3.朴素贝叶斯

朴素指的是事件之间互相独立没有影响。（adj）

+1是确保概率不能是0,+14是为了确保不大于1，加几都行，但是要确保一样

P（a　|　　sports)= (2+1) / 11 +14

来计算不太好统计的概率，比如词频统计

高斯朴素贝叶斯分类器：样本数据是连续的，呈正态分布，

多项式朴素贝叶斯分类器：样本数据是离散的

伯努利朴素贝叶斯分类器：适合特征为二元离散值或者是稀疏的多元离散值的数据集

分类模型的效果和性能（怎么说明模型的好坏）如何优化模型，性能指标

看样本分布情况

模型的评估

模型的优化

聚类

13

CLASS2.1102：模型评估

回归问题用R2评估模型好坏

分类问题：

错误率、精度

针对不同情况下有不同应用背景：

查准率：（查的准不准）

我认为是A的结果全是A（查准率）我认为是C的结果是A（召回率）

召回率：（查的全不全，有多少漏网之鱼）√ 比如疾病感染问题

查对了用T表示，真

查错了用F表示，假

查的结果：正—>P,反—>N

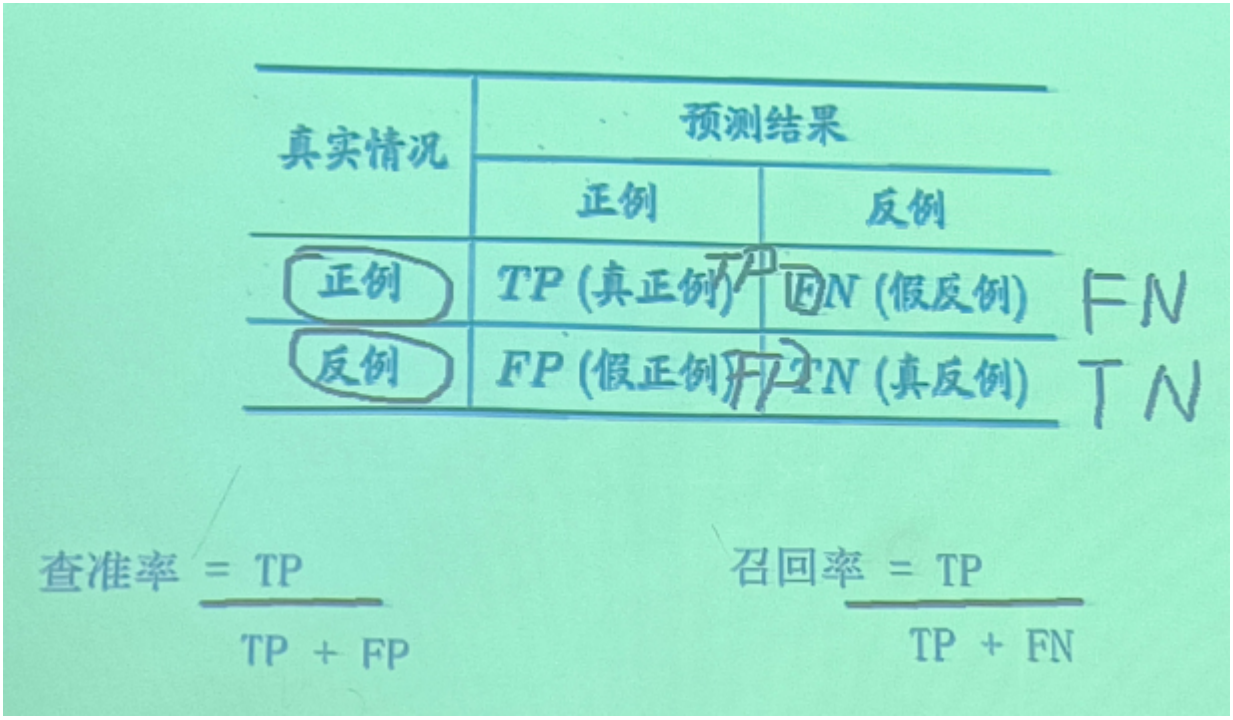
查准率： TP

两者有点矛盾。

推荐系统评测指标—准确率(Precision)、召回率(Recall)、F值(F-Measure)_召回率公式-CSDN博客

让查准率提高，也就是要提高标准，召回率会更低一点

想让召回率更高，也就是要降低标准，查准率会低



F1得分：

- F1得分：

$$f1 = \frac{2 * 查准率 * 召回率}{查准率 + 召回率} \quad (1)$$

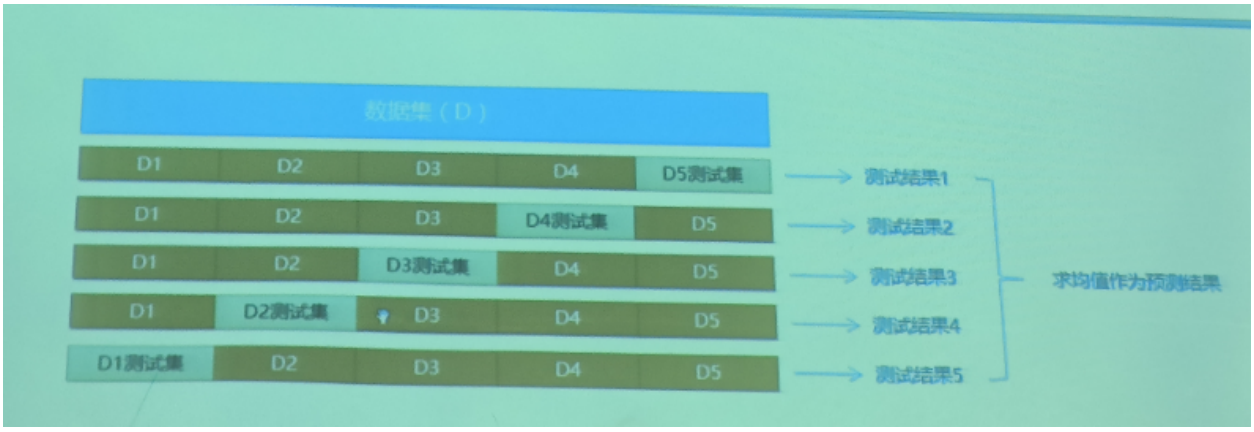
混淆矩阵：

2.训练集和测试集的划分

3.交叉验证法

当样本数量比较少的时候，采用（有点扩大数据集的意思）

把20个，分成五组，每组四个成员，每次选定一个小组作为测试集，其他四组作为训练集，依次轮流，直到每个小组都成为过测试集为止，得到五种结果，取平均值就是。



CLASS2.1107：模型优化

①验证曲线与学习曲线

验证曲线指的是根据不同的评估系数，来评估模型的优劣。

②学习曲线

学习曲线用来评估训练集和测试集如何划分对模型造成的影响

③超参数

在开始学习之前设置值的参数，而不是通过训练得到的参数数据。主要依据经验获得。

超参数选择的方法：随机搜索（参数不固定，随机的），网格搜索（模型参数的组合，相当于一下子确定多个参数的最优解，穷举法）

CLASS2.1109：聚类问题

聚类问题是无监督学习：没有标签，不知道答案

分类问题是有监督学习：有监督问题

①原型聚类，原型相当于一种规则，根据规则来聚类，典型模型：K-Means

②密度聚类，

③层次聚类，

噪声密度

不需要知道类别是多少，随机选一个点，然后做一个圆，把圈进来的归一类。（类似于传销）

支持向量机,每个类别