blog.csdn.net

(26条消息) dataframe数据标准化处理_数据预处理——标准化/归一化（实例）_weixin_39987138的博客-CSDN博客

4-5 minutes

这次我们来说说关于数据预处理中的数据标准化及归一化的问题。主要以理论+实例的方式为大家展示。

本次实验也将会展示部分数据以及代码，有兴趣的小伙伴可以自己动手试试~

在本次实例过程中，我们使用的数据是：2010-2018年间广州市经济与环境的时间序列资料，数据来源为《广州市统计年鉴》及《国民经济和社会发展统计公报》，感兴趣的同学也可利用其它数据进行实例操作。(本次实验的Excel数据附在文后)

一、归一化(Normalization)

描述：

将数据映射到指定的范围，如：把数据映射到0～1或-1~1的范围之内处理。

作用：1、数据映射到指定的范围内进行处理，更加便捷快速。

2、把有量纲表达式变成无量纲表达式，便于不同单位或量级的指标能够进行比较和加权。经过归一化后，将有量纲的数据集变成纯量，还可以达到简化计算的作用。

常见做法：Min-Max归一化

python实现：

(1)导入数据并删除我们不要的列：

```
import numpy as np

import pandas as pd

df=pd.read_excel('C://Users/Administrator/Desktop/data_py.xlsx',sheet_name='广州',encoding='utf-8')

df.drop(columns="时间",axis=1,inplace=True)

df.set_index([[2010,2011,2012,2013,2014,2015,2016,2017,2018]],inplace=True)

df.drop(columns=['第二产业产值占比','第三产业产值占比','一般工业固体废物综合利用
率'],axis=1,inplace=True)
```

(2)查看数据：

可以看到，数据前三列特征的数量级明显大于后面的特征很多，如果这个时候我们想要利用这个数据表来衡量广州市的发展情况时，地区生产总值、公共财政收入、人均生产总值这三项特征就会起到主导作用从而遮盖住其他的特征，这样的模型效果是很差的，因此我们可以通过归一化来解决这个问题。

(3)Min-Max归一化：

for i in list(df.columns):

获取各个指标的最大值和最小值

Max = np.max(df[i])

Min = np.min(df[i])

df[i] = (df[i] - Min)/(Max - Min)

(4)查看归一化结果：

二、标准化(Normalization)

注：在英文翻译中，归一化和标准化的翻译是一致的，而在实际使用中，我们需要根据实际的公式(或用途)去理解~

数据标准化方法有多种，如：直线型方法(如极值法、标准差法)、折线型方法(如三折线法)、曲线型方法(如半正态性分布)。不同的标准化方法，对系统的评价结果会产生不同的影响。其中，最常用的是Z-Score 标准化。

Z-Score 标准化

其中，

为数据均值(mean)，

为标准差(std)。

描述：

将原数据转换为符合均值为0，标准差为1的标准正态分布的新数据。

作用：1、提升模型的收敛速度(加快梯度下降的求解速度)

2、提升模型的精度(消除量级和量纲的影响)

3、简化计算(与归一化的简化原理相同)

python实现：

(1)(这里我们重置一下数据表df，避免实验的偶然性)

```
from sklearn import preprocessing

df=pd.read_excel('C://Users/Administrator/Desktop/data_py.xlsx',sheet_name='广州',encoding='utf-8')

df.drop(columns="时间",axis=1,inplace=True)

df.set_index([[2010,2011,2012,2013,2014,2015,2016,2017,2018]],inplace=True)

df.drop(columns=['第二产业产值占比','第三产业产值占比','一般工业固体废物综合利用
率'],axis=1,inplace=True)
```

(2)Z-Score 标准化，最简便、也是L推荐的方法是用：sklearn库里的StandardScaler()。

实例化：

```
zscore = preprocessing.StandardScaler()

# zscore标准化

zscore = zscore.fit_transform(df)
```

查看标准化后的数据：

```
df_zscore = pd.DataFrame(zscore,index=df.index,columns=df.columns)

df_zscore
```

使用归一化/标准化会改变数据原来的规律吗?

归一化/标准化实质是一种线性变换，线性变换有很多良好的性质，这些性质决定了对数据改变后不会造成“失效”，反而能提高数据的表现，这些性质是归一化/标准化的前提。比如有一个很重要的性质：线性变换不会改变原始数据的数值排序。

如果是单纯想实现消除量级和量纲的影响，用Min-Max还是用Z-Score?

1、数据的分布本身就服从正态分布，使用Z-Score。

2、有离群值的情况：使用Z-Score。

这里不是说有离群值时使用Z-Score不受影响，而是，Min-Max对于离群值十分敏感，因为离群值的出现，会影响数据中max或min值，从而使Min-Max的效果很差。相比之下，虽然使用Z-Score计算方差和均值的时候仍然会受到离群值的影响，但是相比于Min-Max法，影响会小一点。

当数据出现离群点时，用什么方法?

当数据中有离群点时，我们可以使用Z-Score进行标准化，但是标准化后的数据并不理想，因为异常点的特征往往在标准化后容易失去离群特征，此时就可以用RobustScaler 针对离群点做标准化处理。

三、Robust标准化(RobustScaler)

很多时候我们在机器学习中，或是其他模型都会经常见到一个词：鲁棒性。也就是Robust的音译。

计算机科学中，健壮性(英语：Robustness)是指一个计算机系统在执行过程中处理错误，以及算法在遭遇输入、运算等异常时继续正常运行的能力。 诸如模糊测试之类的形式化方法中，必须通过制造错误的或不可预期的输入来验证程序的健壮性。很多商业产品都可用来测试软件系统的健壮性。健壮性也是失效评定分析中的一个方面。

关于Robust,是这么描述的：

This Scaler removes the median(中位数) and scales the data according to the quantile range(四分位距离，也就是说排除了outliers).

Huber从稳健统计的角度系统地给出了鲁棒性3个层面的概念：

一是模型具有较高的精度或有效性，这也是对于机器学习中所有学习模型的基本要求；

二是对于模型假设出现的较小偏差，只能对算法性能产生较小的影响；

主要是：噪声(noise)

三是对于模型假设出现的较大偏差，不可对算法性能产生“灾难性”的影响。

主要是：离群点(outlier)

在机器学习，训练模型时，工程师可能会向算法内添加噪声(如对抗训练)，以便测试算法的「鲁棒性」。可以将此处的鲁棒性理解述算法对数据变化的容忍度有多高。鲁棒性并不同于稳定性，稳定性通常意味着「特性随时间不变化的能力」，鲁棒性则常被用来描述可以面对复杂适应系统的能力，需要更全面的对系统进行考虑。

使用方法

(1)和Z-Score一样，进行实例化：

```
robust = preprocessing.RobustScaler()

# robust标准化处理

df_robust = robust.fit_transform(df)
```

(2)查看标准化后的数据：

df_robust = pd.DataFrame(df_robust,index=df.index,columns=df.columns)

df_robust

(在这里我们仅仅是做一个示范，并不是说当前这个数据表必须用Robust进行标准化)

实验数据表：

链接：https://pan.baidu.com/s/1MOmda_0kDbwRNp9jJ0XOgwpan.baidu.com

提取码：5ca2

由于时间关系，剩下的内容我会在下次更新中一并补充~

以上便是的内容，感谢大家的细心阅读，同时欢迎感兴趣的小伙伴一起讨论、学习，想要了解更多内容的可以看我的其他文章，同时可以持续关注我的动态~