**意大利红酒数据的标准化**

# 1简介

很多机器学习的算法都需要对变量进行标准化，以达到很好的效果。如：KNN（k-nearest neighbors）算法使用前进行变量标准化可以使每个特征对分类结果的影响相同；梯度下降算法，如果不进行变量标准化，一些权重会更新的较快；PCA和LDA降维，如果不进行变量标准化，降维后的结果会更偏向于更大取值的特征变量。

常用的标准化方法有：z-score标准化、0-1标准化、小数定标标准化和logistic标准化。其中，小数定标标准化即是通过移动数据的小数点位置来进行标准化，一般适用于数据系列分布比较离散的情况，实现较易。

logistic标准化是通过Sigmoid函数，将原始数据转化到[0,1]区间，且聚集于0或1附近，适用于数据分布较为集中在零点两侧的情况，简单易用。

本文不详细展示小数定标标准化和logistic标准化的方法。

# 2数据和环境准备

下面我们将通过意大利红酒的部分数据，调用scikit-learn包（sklearn）分别实现0-1标准化和z-score标准化，总结学习这两种标准化方法的特点。

本案例使用的环境为Anaconda + Jupyter notebook。

## 2.1数据说明

我们使用的是UCI提供的红酒数据集，该数据集显示意大利红酒化学分析的结果，共178个样本，每个样本有13个特征变量，一个类别变量。

为简化实验，更清晰地展现实验结果，仅选取两个特征变量进行展示。

Alcohol：酒精含量

Malic acid：羟基丁二酸

Class：红酒分类情况

请大家在自己的电脑上新建一个文件夹input，将实验数据集“wine.csv”放入此文件夹，以供实验使用。

新建一个文件夹output，作为实验保存数据的输出目录。

## 2.2导入依赖

导入我们要用到的工具包，注意，视自己电脑的情况可能需要安装依赖，请大家导入依赖时，逐个导入，从而快速定位要安装的工具包。

*# 导入依赖包*import pandas as pd  
import numpy as np  
from sklearn import preprocessing  
import seaborn as sns  
import matplotlib.pyplot as plt

## 2.3读取数据

实验数据集为较常见的csv格式，其格式全称为comma-separated value（逗号分隔值），一般每条记录占一行，记录之间用换行符分开；记录内的不同属性以逗号为分隔符。

时常用到的数据格式除了csv还有txt文件和xlsx（Excel）文件，txt文件默认同样一条记录占一行，记录之间用换行符分开，但记录不同属性之间默认使用制表符【TAB】为分隔符；xlsx文件则很简单，一条记录占一行，每个属性占一列。

大家可以分别尝试使用pandas的read\_table、read\_excel和read\_csv方法读取txt文件、xlsx文件和csv文件，尝试正确的对应关系后，使用其他两个方法打开文件，看看会发生什么。

使用Pandas的read\_csv()函数读取数据，并查看数据前五行。

*# 导入数据*wine = pd.read\_csv(**"D:/input/wine.csv"**)  
*# 查看前五行*wine.head(5)

由上表可见，两个特征Alcohol和Malic acid取值范围不同，Alcohol相较于Malic acid取值高一个数量级，故而进行标准化是有必要的。

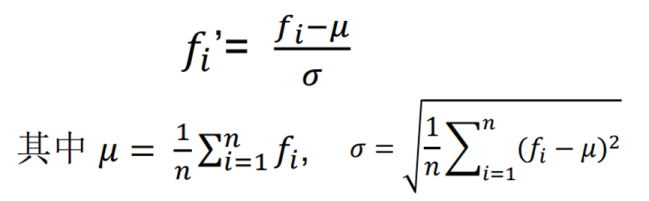
# 3数据标准化

## 3.1标准化简述

数据标准化（归一化）处理是数据挖掘的一项基础工作。不同评价指标往往具有不同的量纲，数值见的差别可能很大，不进行处理可能会影响到数据分析的结果；同时有很多机器学习算法需要其输入特征为标准化的形式。为了消除指标之间的量纲和取值范围差异的影响，需要进行标准化处理，将数据按照比例进行缩放，使之落入一个特定的区域，便于进行综合分析，这就是数据标准化。

Z-score标准化也称为标准差标准化，是最常用的标准化方法，该方法通过对数据集的每一个样本进行处理，使得处理后的数据具有固定的均值和标准差。

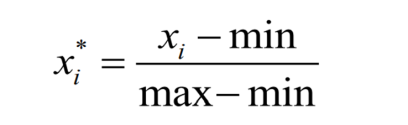
设原数据集中的元素为f，标准化后的取值为f’则



经过Z-Score标准化后的数据，能够直观反应每个数据点距离平均值点的标准差距离，从而理解整体数据的分布情况。均值将落在0附近，而每一个数据点离零点的距离可解释为其远离均值的标准差距离。

Z-Score的标准化方法适用于数据系列中最大值或最小值未知、数据分布非常离散的情况。

Min-Max标准化也称为离散标准化，该方法的核心即是对数据系列作线性变换，使得处理过后数据均落在［0，1］区间内。其公式为



0-1标准化适用于需要将数据简单地变换映射到某一区间中，保留了原来数据中存在的关系，是消除量纲和数据取值范围影响的最简单方法；但其不足之处在于当有新数据加入时，可能会导致数据系列中的最大值或最小值发生变化，此时便需要重新定义最大值、最小值。

## 3.2完成标准化

我们直接使用sklearn（scikit-learn）中的包，分别对这两个特征变量进行0-1标准化和z-score标准化。

*# 0-1标准化*minmax\_scale = preprocessing.MinMaxScaler().fit(wine[[**'Alcohol'**,**'Malic acid'**]])  
np\_minmax = minmax\_scale.transform(wine[[**'Alcohol'**,**'Malic acid'**]])  
*# z-score标准化*std\_scale = preprocessing.StandardScaler().fit(wine[[**'Alcohol'**,**'Malic acid'**]])  
np\_std = std\_scale.transform(wine[[**'Alcohol'**,**'Malic acid'**]])

下面，我们来看一下0-1标准化和z-score标准化后，两个特征变量取值的特点。

*# 0-1标准化后，两个特征变量的最小值*print(**"0-1标准化后两个指标的最小值：**\n**"**, **"Alcohol最小值为"**, np\_minmax[:, 0].min(), **"Malic acid最小值为"**, np\_minmax[:, 1].min())  
*# 0-1标准化后，两个特征变量的最大值*print(**"0-1标准化后两个指标的最大值：**\n**"**, **"Alcohol最大值为"**, np\_minmax[:, 0].max(), **"Malic acid最大值为"**, np\_minmax[:, 1].max())  
*# z-score标准化后，两个特征变量的最小值*print(**"z-score标准化后两个指标的最小值：**\n**"**, **"Alcohol最小值为"**, np\_std[:, 0].min(), **"Malic acid最小值为"**, np\_std[:, 1].min())  
*# z-score标准化后，两个特征变量的最大值*print(**"z-score标准化后两个指标的最大值：**\n**"**, **"Alcohol最大值为"**, np\_std[:, 0].max(), **"Malic acid最大值为"**, np\_std[:, 1].max())

看看标准化后的前五行数据。

## 3.3数据可视化

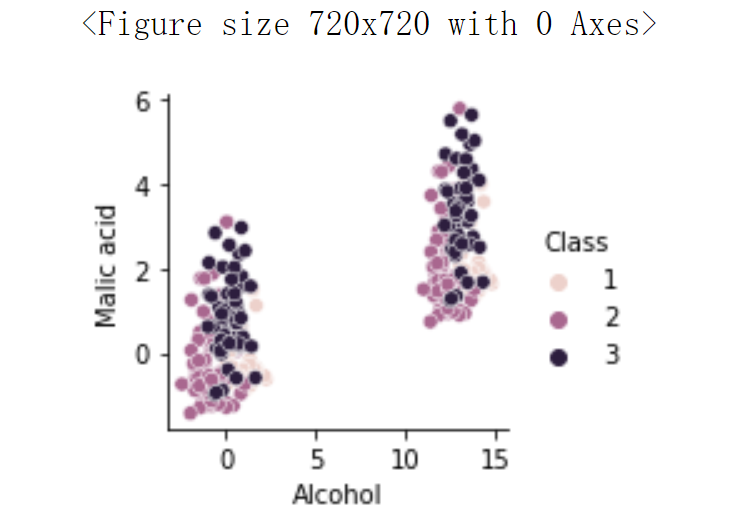
下面我们更直观的，通过seaborn画出相应散点图，展示0-1标准化和z-score标准化的不同。

我们可以将原数据、0-1标准化后的数据和z-score标准化后的数据合并为一个dataframe，用作将三个数据集画到同一散点图中。

*# 获取数据备份*df\_minmax = wine.copy()  
df\_std = wine.copy()  
*# 将前面标准化后的数据替换进备份*df\_minmax[[**'Alcohol'**,**'Malic acid'**]] = np\_minmax  
df\_std[[**'Alcohol'**,**'Malic acid'**]] = np\_std  
*# 合并数据集*wine\_contrast = pd.concat([wine, df\_minmax, df\_std], axis=0)

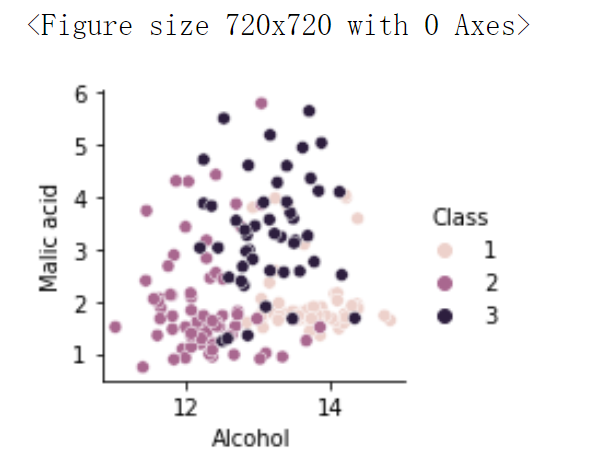
调用seaborn工具包的方法，绘制散点图。

*# 创建画布*plt.figure(figsize=(10, 10))  
*# 指定x轴和y轴的数据集和坐标轴标识*sns.pairplot(hue=**"Class"**, data=wine\_contrast, x\_vars=**"Alcohol"**, y\_vars=**"Malic acid"**)  
*# 显示图像*plt.show()

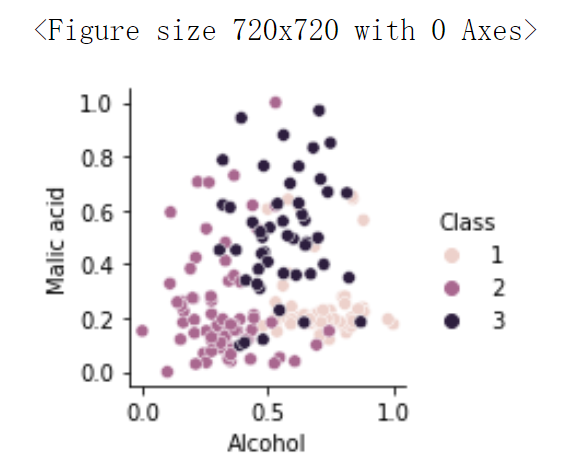


更进一步，我们分别用seaborn画出原数据、0-1标准化后数据、z-score标准化后数据，按照红酒种类不同进行分类的散点图。

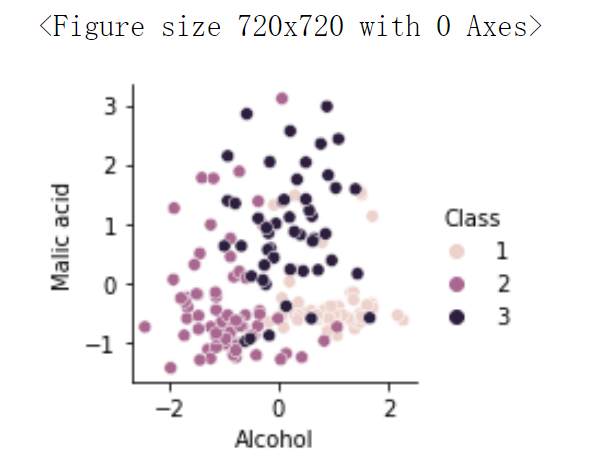
*# 画出原数据散点图  
# 创建画布*plt.figure(figsize=(10, 10))  
*# 指定x轴和y轴的数据集和坐标轴标识*sns.pairplot(hue=**"Class"**, data=wine, x\_vars=**"Alcohol"**, y\_vars=**"Malic acid"**)  
plt.show()



*# 画出0-1标准化后散点图  
# 创建画布*plt.figure(figsize=(10, 10))  
*# 指定x轴和y轴的数据集和坐标轴标识*sns.pairplot(hue=**"Class"**, data=df\_minmax, x\_vars=**"Alcohol"**, y\_vars=**"Malic acid"**)  
plt.show()



*# 画出z-score标准化后的散点图  
# 创建画布*plt.figure(figsize=(10, 10))  
*# 指定x轴和y轴的数据集和坐标轴标识*sns.pairplot(hue=**"Class"**, data=df\_std, x\_vars=**"Alcohol"**, y\_vars=**"Malic acid"**)  
plt.show()



上面的散点图很好地反映了0-1标准化和z-score标准化的特点，如0-1标准化后的数据取值范围为[0,1]，z-score标准化后的数据均值为0等。

# 4数据保存

在做完数据清洗后会对新的“好数据”进行存储，以方便之后的建模等工作时可以直接调用。

在Pandas中，可以直接对格式为DataFrame的数据进行文件的存储。使用Pandas中的to\_csv()函数可以进行csv文件的输出,因为不需要写入索引信息，所以此处对index参数设置为False。

*# 保存0-1标准化后的数据*df\_minmax.to\_csv(**"D:/output/minmax.csv"**,index = False)  
  
*# 保存z-score标准化后的数据*df\_std.to\_csv(**"D:/output/std.csv"**,index = False)

可见，数据保存完成，大家可以打开文件，验证数据集的正确性。

若想输出excel文件则可以使用to\_excel()函数，json文件可使用to\_json()函数等，大家可以自行尝试。