

**NORTHEASTERN**

**UNIVERSITY**

课程设计报告

课程名称：人工智能

设计题目：葡萄酒质量预测（分类）

指导教师：张岩峰

学生姓名：谭雅云 学号：20205926

班 级：计算机2004

专 业：计算机科学与技术

设计日期：2022年11月

# 课程设计任务书

|  |
| --- |
| **设计题目：**葡萄酒质量预测（分类） |
| **一、实验简介**   1. 预测每个葡萄酒样品是红葡萄酒(red)还是白葡萄酒(white)，测试集与训练集放入类型文件夹，比例为9：1 2. 预测每个葡萄酒样品的质量，可以是低、中或高，注意quality值小于等于5为低品质葡萄酒，六到七之间为中品质酒，7以上为高品质酒，测试集与训练集放入质量文件夹，比例为9：1   要求使用决策树或神经网络作为分类模型预测以上两个分类问题。预测葡萄酒样品的种类可以使用深度神经网络模型进行预测，预测葡萄酒样品的质量可以使用决策树分类器。本次实验的目的在于掌握使用决策树模型和神经网络模型来分析数据集不同特征，对样本进行分类。  二、数据简介  该数据集与“Vinho Verde”种类葡萄酒的红色和白色种类有关，该品牌是葡萄牙地区的独特产品。数据集有几个特征会被用来对葡萄酒的质量进行分类，其中很多都是化学相关的，所以我们需要对这些化学物质有一个基本的了解。可用的变量如下：  输入变量：  1 - fixed acidity 固定酸度，葡萄酒中的主要酸，包括酒石酸，琥珀酸，柠檬酸等  2 - volatile acidity 挥发酸度，葡萄酒中存在的气态酸  3 - citric acid 柠檬酸，弱有机酸，天然存在于柑橘类水果中  4 - residual sugar 残糖，发酵后剩余的糖量  5 - chlorides 氯化物，葡萄酒中存在的盐量  6 - free sulfur dioxide 游离二氧化硫，用于放置葡萄酒氧化和微生物腐败  7 - total sulfur dioxide 总二氧化硫，二氧化硫总量  8 - density 密度  9 - pH ph值  10 - sulphates 硫酸盐，添加的硫酸盐可以保持新鲜度并保护葡萄酒免受氧化和细菌的侵害。  11 - alcohol Output variable (based on sensory data) 酒精，酒精百分比  输出变量：  12 - quality (score between 0 and 10) 质量    图(1)  我们可以看到数据集包含的一些信息如图(1)所示。 |

目录

[课程设计任务书 2](#_Toc120634118)

[1. 葡萄酒质量预测 5](#_Toc120634119)

[1.1直接尝试：建模 5](#_Toc120634120)

[1.2探索数据 10](#_Toc120634121)

[1.3特征工程 28](#_Toc120634122)

[1.4再次建模 35](#_Toc120634123)

[1.5尝试其他方法 36](#_Toc120634124)

[1.6交叉验证 39](#_Toc120634125)

[2.葡萄酒类型分类 40](#_Toc120634126)

[2.1划分x、y 41](#_Toc120634127)

[2.2划分训练集和测试集 41](#_Toc120634128)

[2.3特征编码 42](#_Toc120634129)

[2.4建模 42](#_Toc120634130)

[2.5可视化训练结果 44](#_Toc120634131)

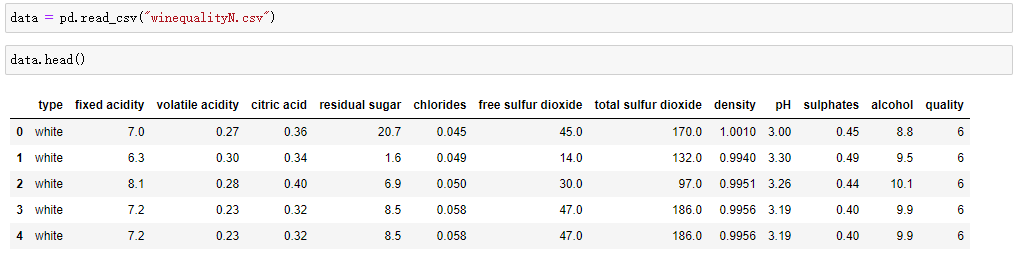
[2.6评估性能 46](#_Toc120634132)

[3.心得体会 46](#_Toc120634133)

## 1. 葡萄酒质量预测

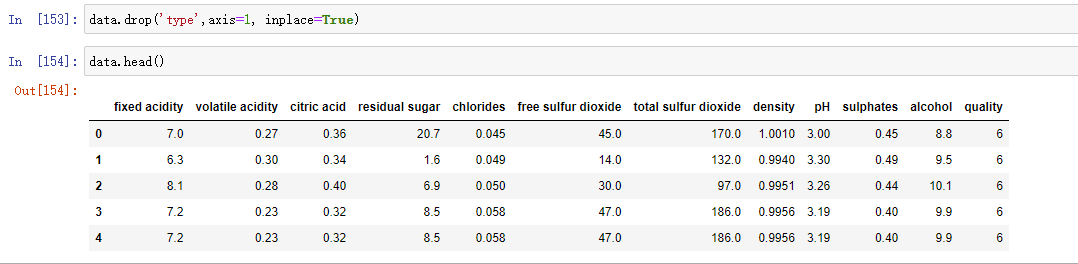
### 1.1直接尝试：建模

首先导入数据

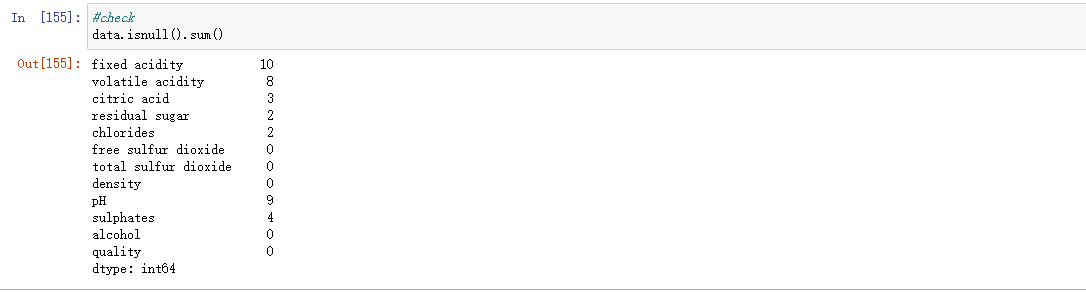


#### 1.1.1数据预处理：处理null

因为做的题目是预测质量，所以不需要type这一项，因为type列的值不是数，之后建模部分会报错。那这里就先丢弃type这一列。需要注意的是，一定要添加inplace=True，否则没有真正改变data。



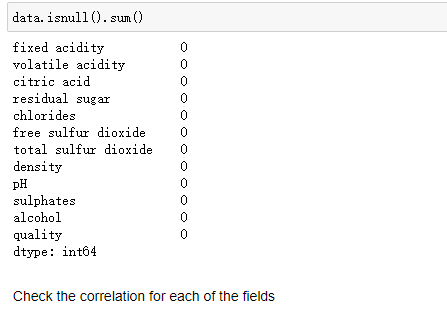
检查是否存在缺失值：



发现有缺失值，但是基数相对于总数来说可以不计入，所以我选择丢弃缺失值。



再检查是否还有缺失值：

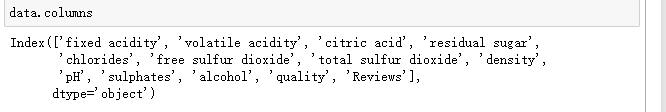


此时可以看出缺失值已经处理完毕。

接下来创造一个新列，存放质量高低评价值：低中高分别为1、2、3.



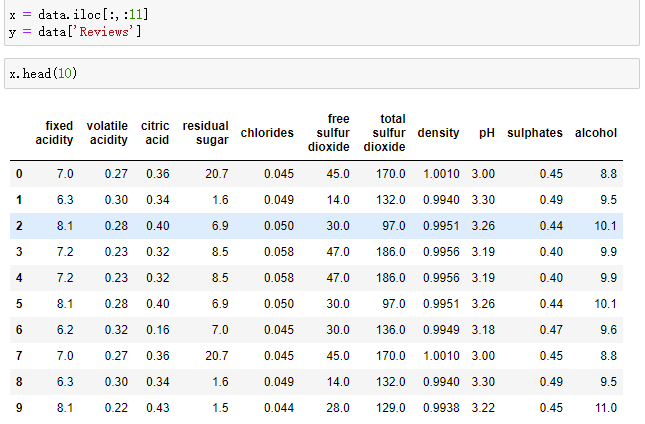
可以看到已经添加了新列

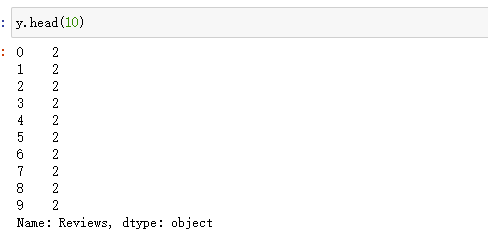


查看一下各评级的葡萄酒数量，差别有些大，优质葡萄酒数量太少了。



#### 1.1.2划分train/test





先尝试一些不做任何特征工程，直接使用数据建模出来的结果会是什么样呢？

将数据拆分为训练集和测试集。

补充说明一下：我尝试过用助教预先给出的训练集和测试集来进行建模训练，发现准确率只能达到59%（未调参、未处理偏移、未降维情况下），而如果使用sklearn.model\_selection.train\_test\_split随机划分训练集和测试集，同样设置比例为9：1，同等数据未处理的条件下，准确率几乎都在70%左右，因此在询问助教以后我的训练集测试集都是通过将train\_test\_split()函数将原始数据打乱得到的。

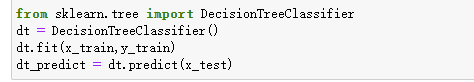


#### 1.1.3建模算法

决策树、随机森林

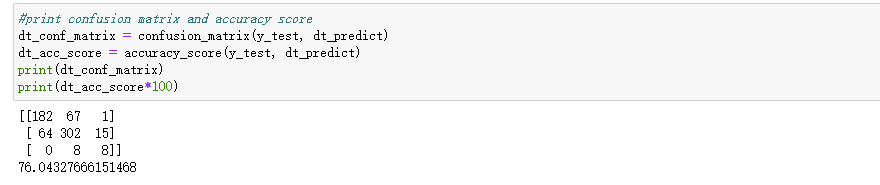
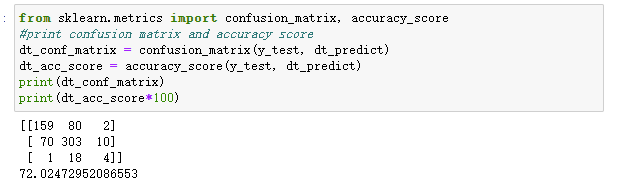
1.决策树

这里使用sklearn进行决策树建模并且实例化对象dt，再导入训练集



结果:

导入测试集，查看混淆矩阵和准确率结果。

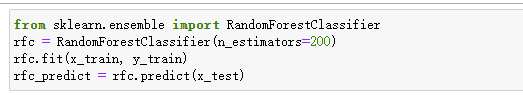


分类准确率只有72%，不是很理想。

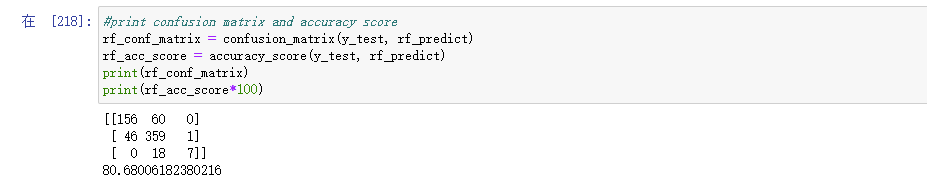
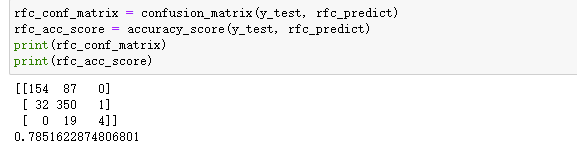
2.随机森林

在查阅决策树资料的时候我了解到有一种模型叫做随机森林，是一种融合了若干决策树的模型，即，一种集成方法。

这里使用sklearn进行随机森林建模并且实例化对象，再导入训练集

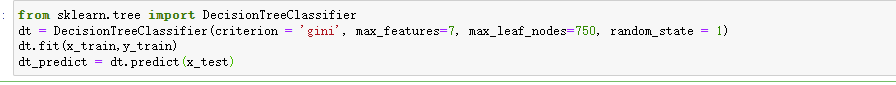


导入测试集，查看混淆矩阵和准确率结果。



结果较决策树模型有明显提升。

那么尝试一下调整决策树模型的参数呢：



针对参数，我查阅了相关文档

criterion：gini或者entropy,前者是基尼系数，后者是信息熵。两种算法差异不大对准确率无影响，信息墒运算效率低一点，因为它有对数运算.一般说使用默认的基尼系数”gini”就可以了，即CART算法。除非你更喜欢类似ID3, C4.5的最优特征选择方法。

splitter： best or random 前者是在所有特征中找最好的切分点 后者是在部分特征中，默认的”best”适合样本量不大的时候，而如果样本数据量非常大，此时决策树构建推荐”random” 。

max\_features：None（所有），log2，sqrt，N 特征小于50的时候一般使用所有的

max\_depth： int or None, optional (default=None) 一般来说，数据少或者特征少的时候可以不管这个值。如果模型样本量多，特征也多的情况下，推荐限制这个最大深度，具体的取值取决于数据的分布。常用的可以取值10-100之间。常用来解决过拟合

min\_samples\_split： 如果某节点的样本数少于min\_samples\_split，则不会继续再尝试选择最优特征来进行划分，如果样本量不大，不需要管这个值。如果样本量数量级非常大，则推荐增大这个值。

min\_samples\_leaf： 这个值限制了叶子节点最少的样本数，如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝，如果样本量不大，不需要管这个值，大些如10W可是尝试下5

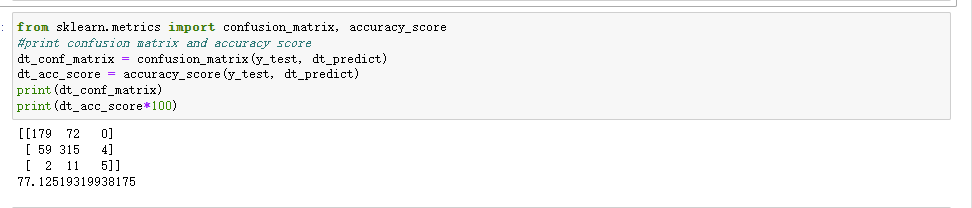
min\_weight\_fraction\_leaf： 这个值限制了叶子节点所有样本权重和的最小值，如果小于这个值，则会和兄弟节点一起被剪枝默认是0，就是不考虑权重问题。一般来说，如果我们有较多样本有缺失值，或者分类树样本的分布类别偏差很大，就会引入样本权重，这时我们就要注意这个值了。

max\_leaf\_nodes： 通过限制最大叶子节点数，可以防止过拟合，默认是"None”，即不限制最大的叶子节点数。如果加了限制，算法会建立在最大叶子节点数内最优的决策树。如果特征不多，可以不考虑这个值，但是如果特征分成多的话，可以加以限制具体的值可以通过交叉验证得到。

class\_weight： 指定样本各类别的的权重，主要是为了防止训练集某些类别的样本过多导致训练的决策树过于偏向这些类别。这里可以自己指定各个样本的权重，如果使用“balanced”，则算法会自己计算权重，样本量少的类别所对应的样本权重会高。

min\_impurity\_split： 这个值限制了决策树的增长，如果某节点的不纯度(基尼系数，信息增益，均方差，绝对差)小于这个阈值则该节点不再生成子节点。即为叶子节点 。

一步一步尝试：首先是random\_state，我看见很多人把这个值设置的都比较高，我试过random\_state=33或者random\_state=44，或者random\_state=10，无一例外都比不指定random\_state使用默认值的效果差。而同等条件下，gini又比 entropy的准确率要高几个百分点，max\_features，max\_leaf\_nodes这两个参数就是通过手动一步一步二分尝试出来的，最后准确率能达到77%

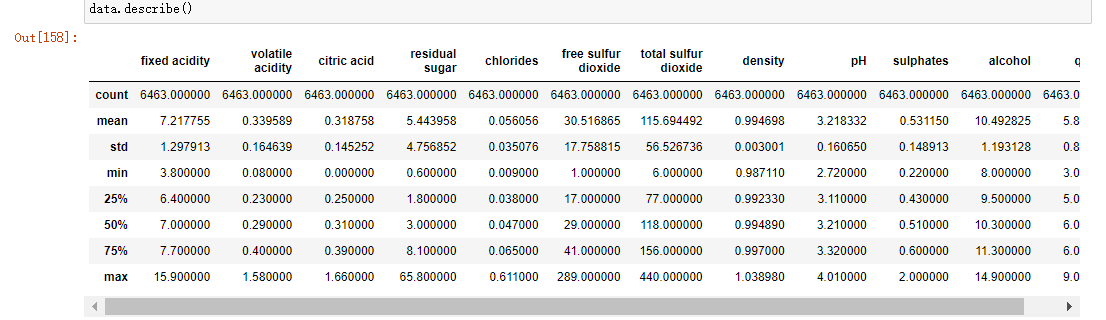


### 1.2探索数据

综上可见未经过特征工程的准确率始终差强人意，那让我们来做一些数据处理的工作。

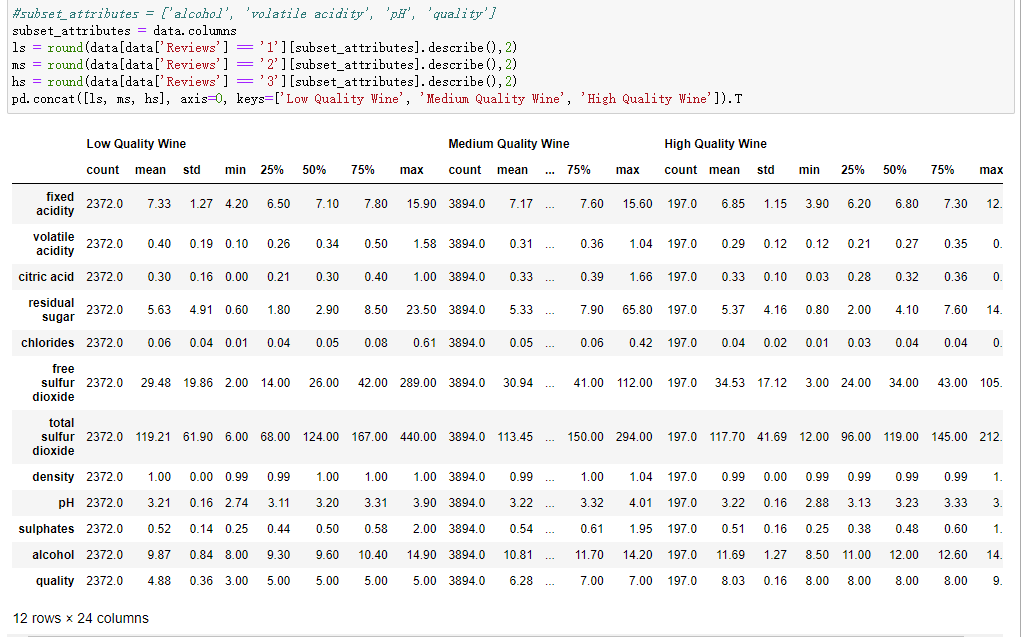
#### 1.2.1表格观察

首先我们来观察一下数据：



可以看到fixed acidity - 25% - 7.1 和 50% - 7.9，变化不大，citric acid似乎是均匀分布，而residual sugar - min - 0.9, max – 15，区别很大。

如果让描述表能按葡萄酒品质的分类来看：



肉眼可以看出来以下几个结论：

1.平均alcohol浓度越高，质量越高。

2. chlorides和volatile acidity在更高质量的葡萄酒中呈现较小的标准差。

3. free sulfur dioxide越高，质量越高，但标准差随质量的提高而减小。

4. fixed acidity越少，质量越高。

表格不是很方便观察，那么接下来用可视化进一步探索。

#### 1.2.2数据可视化

数据可视化有很多种方法：

##### 1.2.2.1 Matplotlib Pyplot

import matplotlib.pyplot as plt

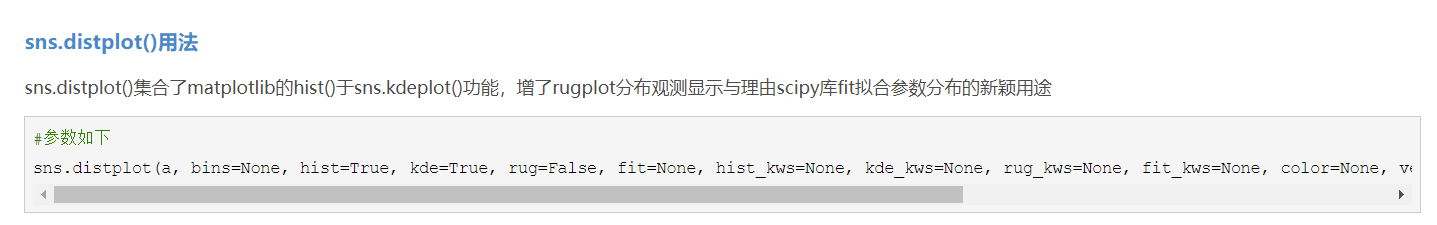
import seaborn as sns

先用plt.figure(figsize=(20, 4)) figsize:指定figure的宽和高，单位为英寸

再用plt.subplot(nrows, ncols, index)划分子区域（参数分别为行、列和索引）

再引入各种视图

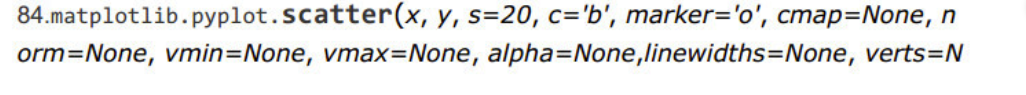
1.1直方图



sns.histplot(df[variable], bins=30,color = 'r')

这里就划分了30个区间

1.2散点图

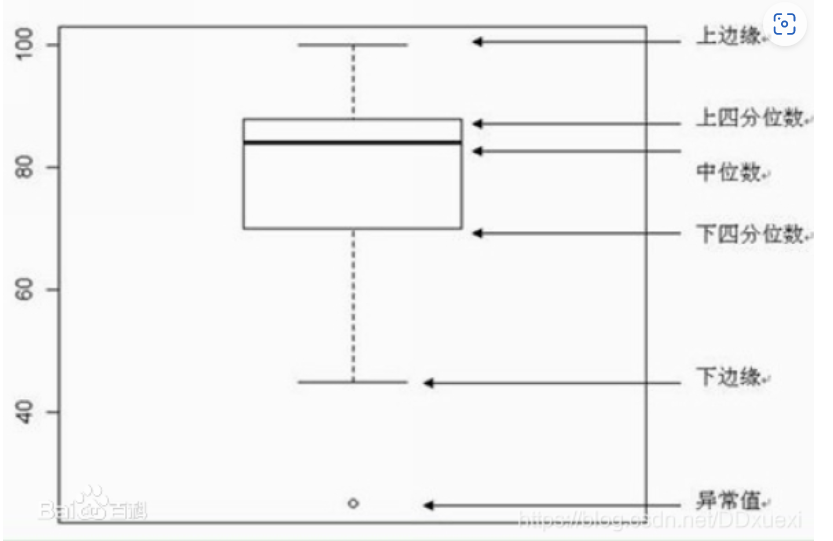




plt.scatter(df[variable],df[target],color = 'g')这里输入的x、y变量就是data和‘quality’

1.3boxplot

箱图它能显示出一组数据的最大值、最小值、中位数及上下四分位数，可用于查看离群值。

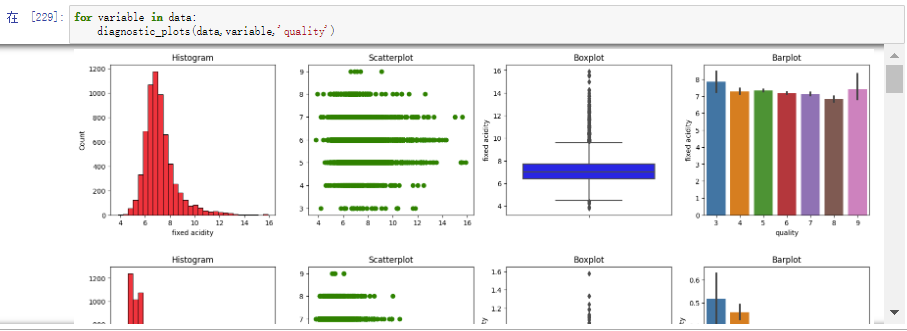


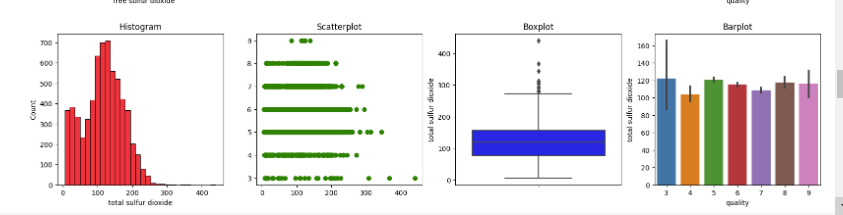
sns.boxplot(y=df[variable],color = 'b')

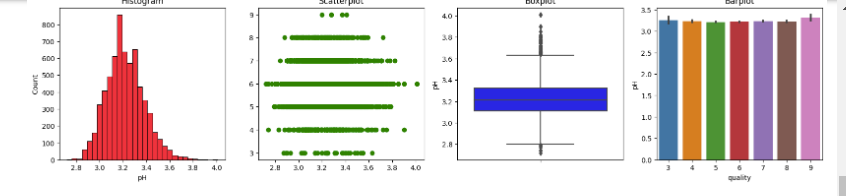
1.4柱状图

sns.barplot(x = target, y = variable, data = df)

那么汇总这四种图形以后，就如下图所示：（仅截取fixed acidity、total sulfur dioxide、alcohol作为示例）





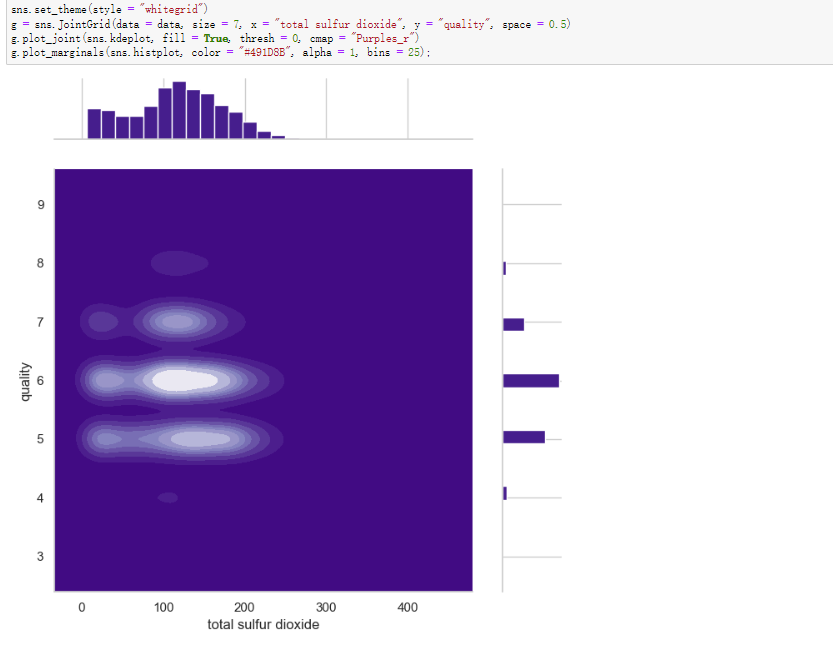


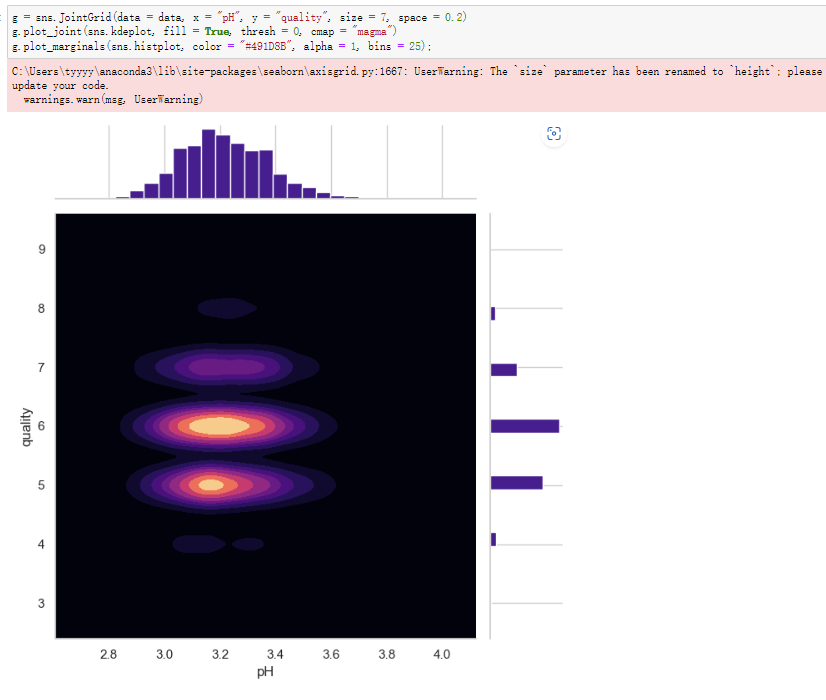
观察结论:只有pH变量分布接近正态，其他变量都是右偏/正偏；除了“酒精”，所有其他特征列显示存在离群值。

边际直方图：

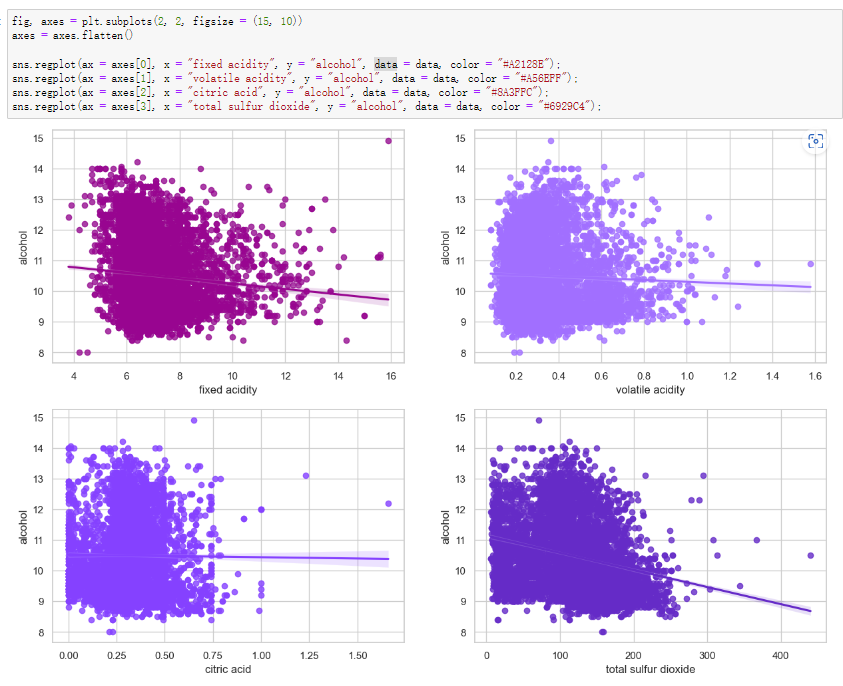
是使用其他图探索横纵坐标的基础上，还使用直方图对横坐标和纵坐标分别进分布探索的图像。

这里探索横纵坐标的时候采用了kdeplot()核密度估计图



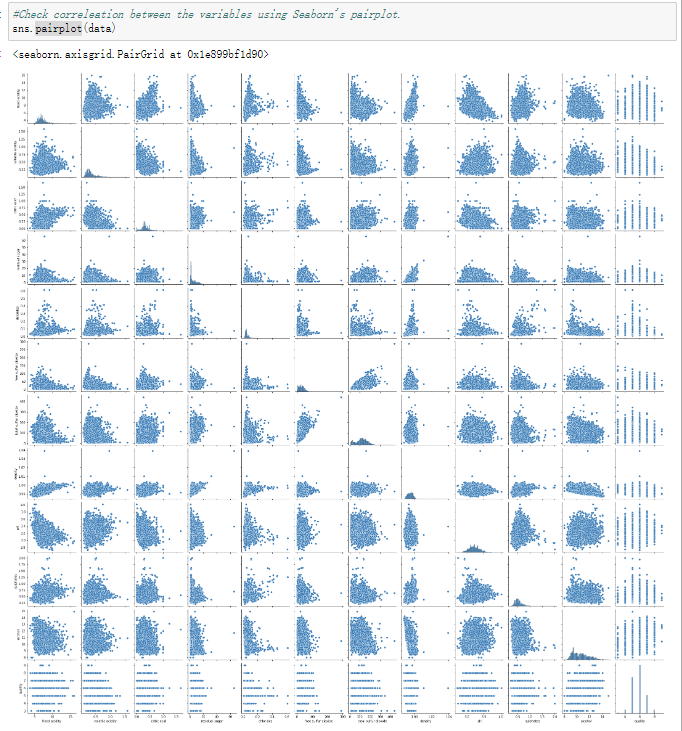


Regplot：两个一维数据关联性的观察利器

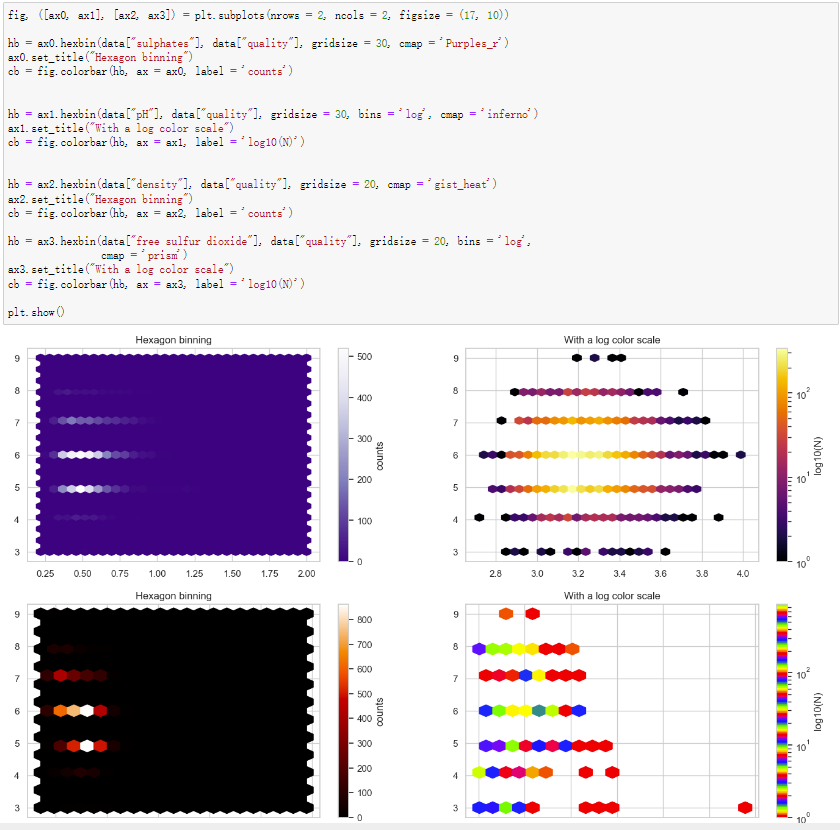


注意“alcohol”-“fixed acidity”和“alcohol”-“total\_sulfur\_dioxide”变量之间的负线性关系。

Pairplot:展现变量两两之间的关系



而箱图还有更美观的做法：六边形箱图



如果想要观察多个变量中的相关联系，那么就要运用热力图。

hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot\_kws={'size': 10}, yticklabels=cols.values, xticklabels=cols.values,cmap="Blues")

annot：数据注释

square：sqaure为True时，每个单元格都是方形的

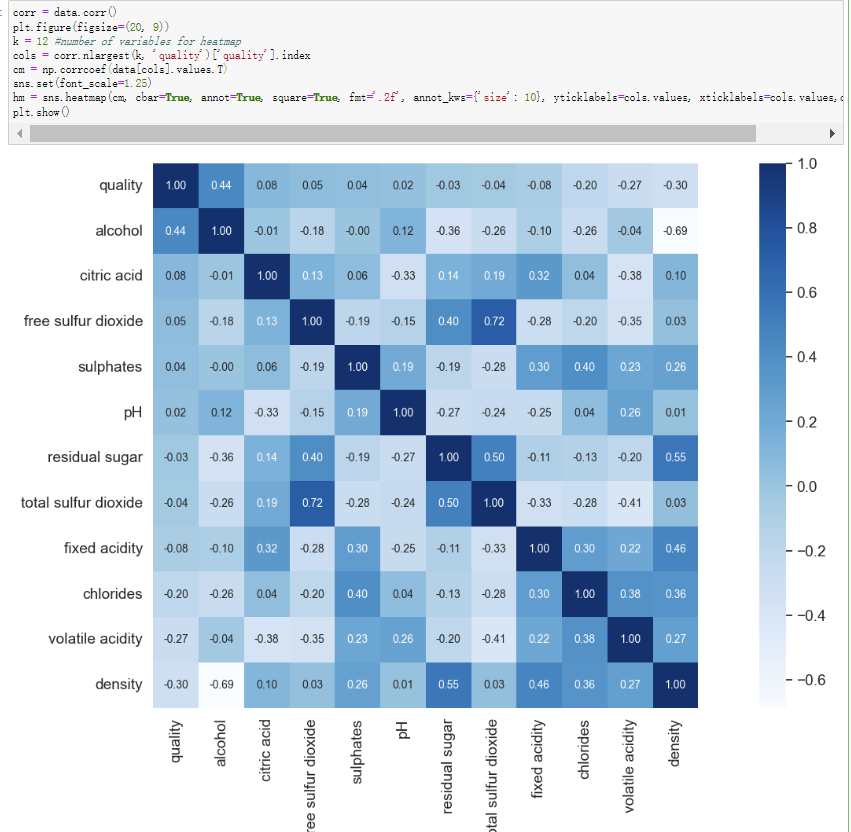
fmt：设置数值的格式化形式

annot\_kws：annot为True时，可设置各个参数，包括大小，颜色，加粗，斜体字

xticklabels, yticklabels：绘制数据框的列名称

cmap：设置颜色带的色系

展示截图：



深色调表示正相关，而浅色调表示负相关。

不难看出，density与residual sugar有很强的正相关关系，而它与alcohol有很强的负相关关系。

free sulphur dioxide与quality，densityr与pH几乎没有相关性。

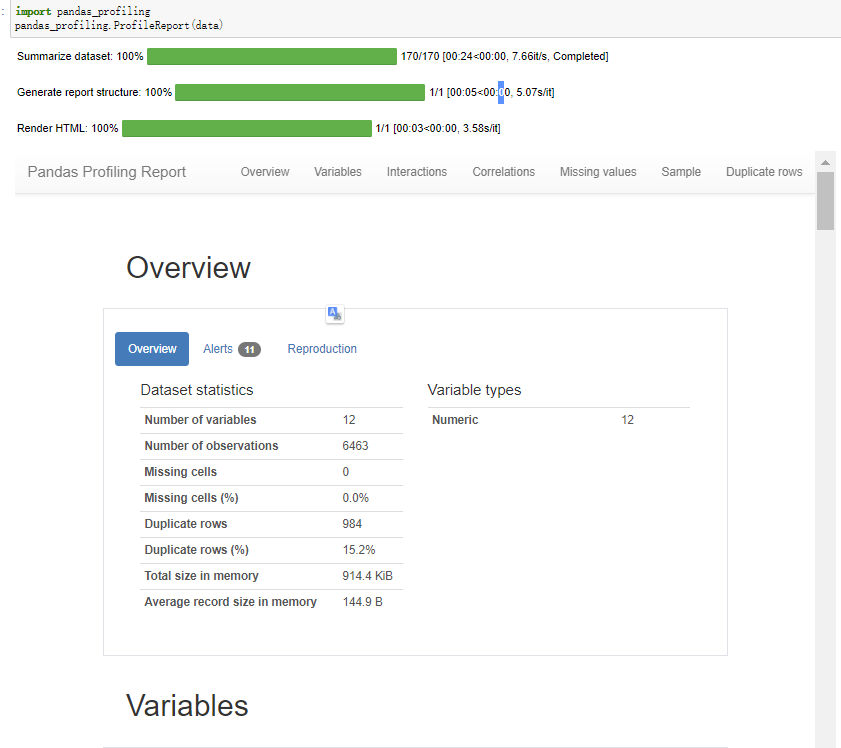
由于相关性为零，我们可以推断这两个预测因子之间没有线性关系。

针对quality，我们得出和describe图表里一致的结论：alcohol与quality的正相关关系最高。

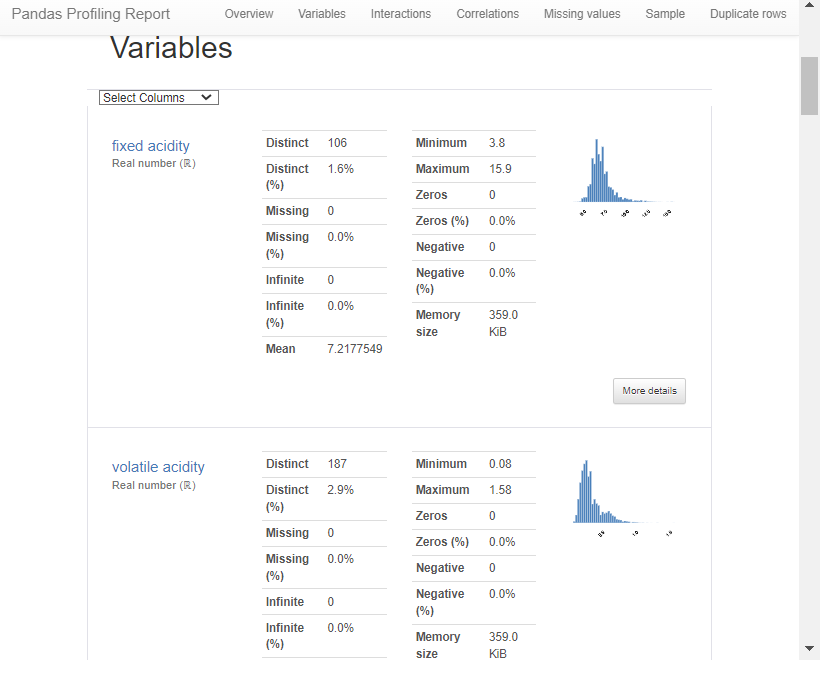
##### 1.2.2.2EDA插件

在查阅资料的时候，我发现有一些pandas EDA 插件可以轻松实现以上describe、直方图散点图、热力图等功能。

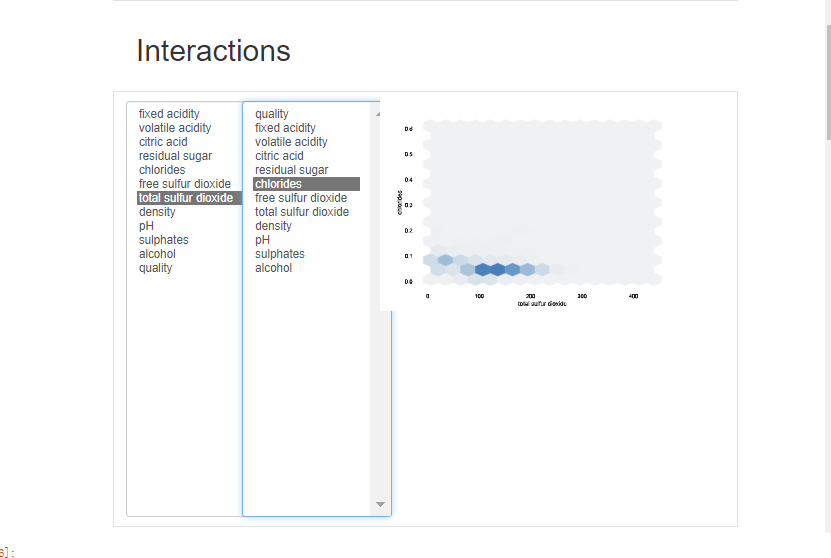
2.1 pandas\_profiling



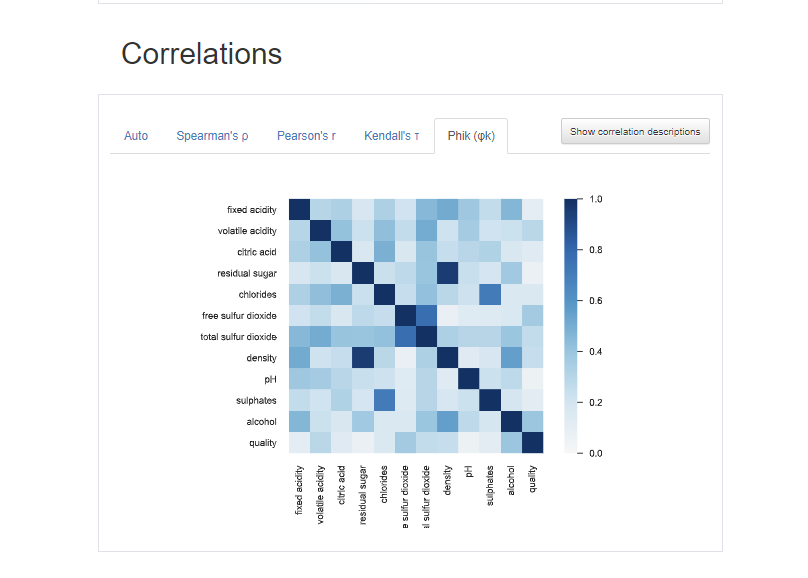
Pandas Profiling使用滑动的用户界面（UI）生成类似的报告，它会在输出框产生一些交互式窗口：



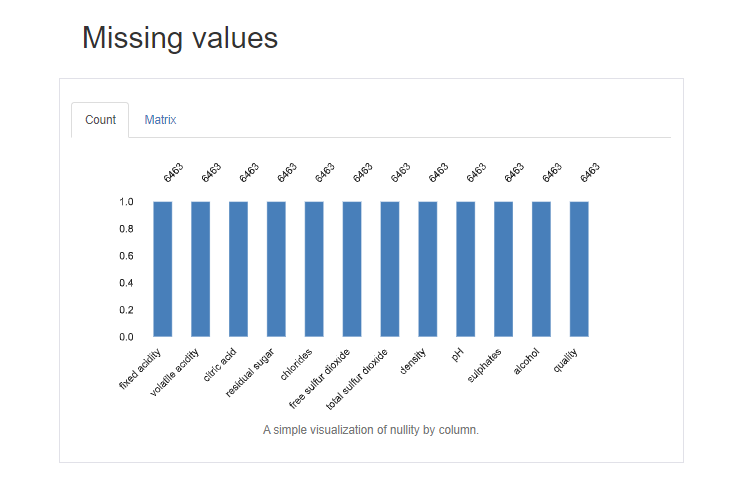
还可以选择两个变量查看它们的六边形箱图，寻找相关性



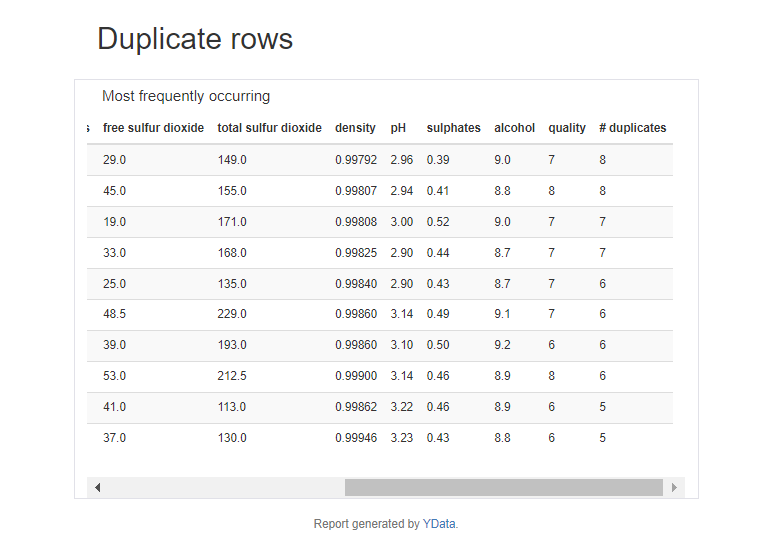
以及针对多个变量查看热力图，寻找相关性;



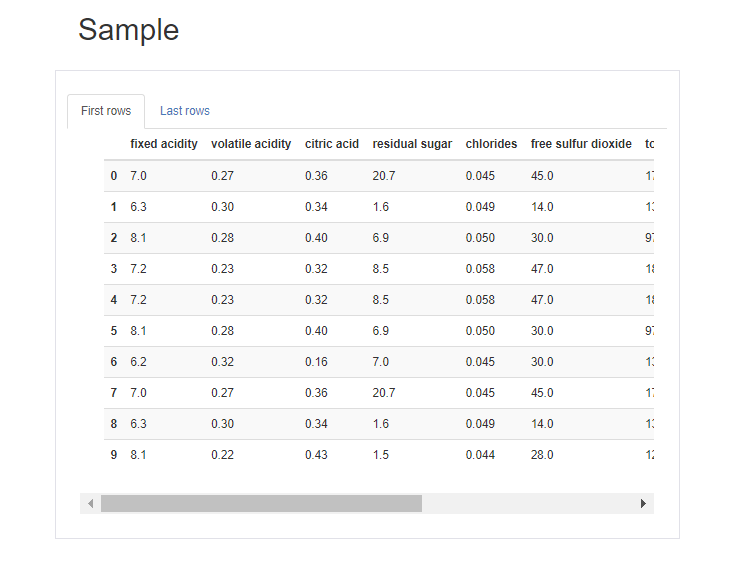
查看缺失值：



查看重复行：

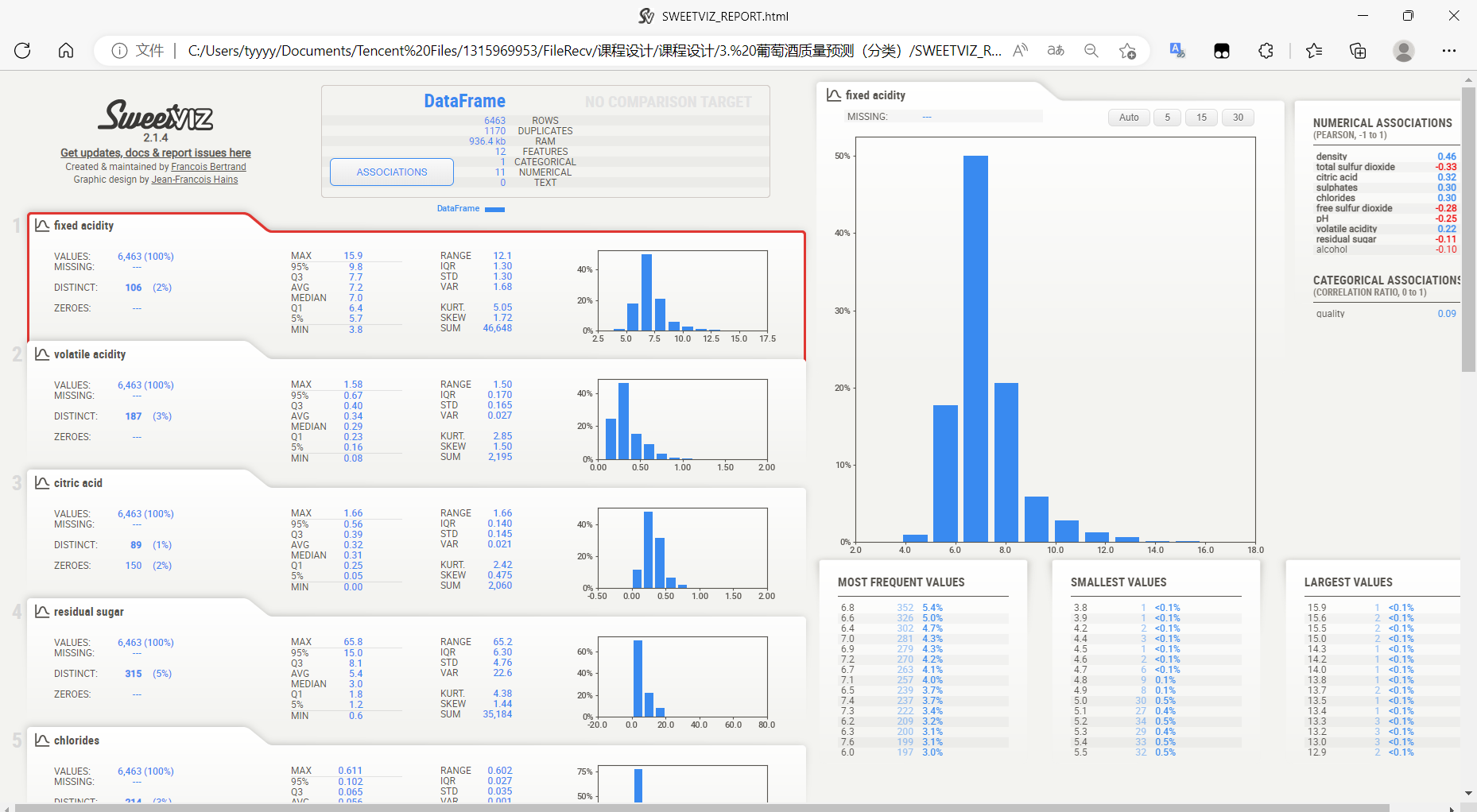


查看样本：

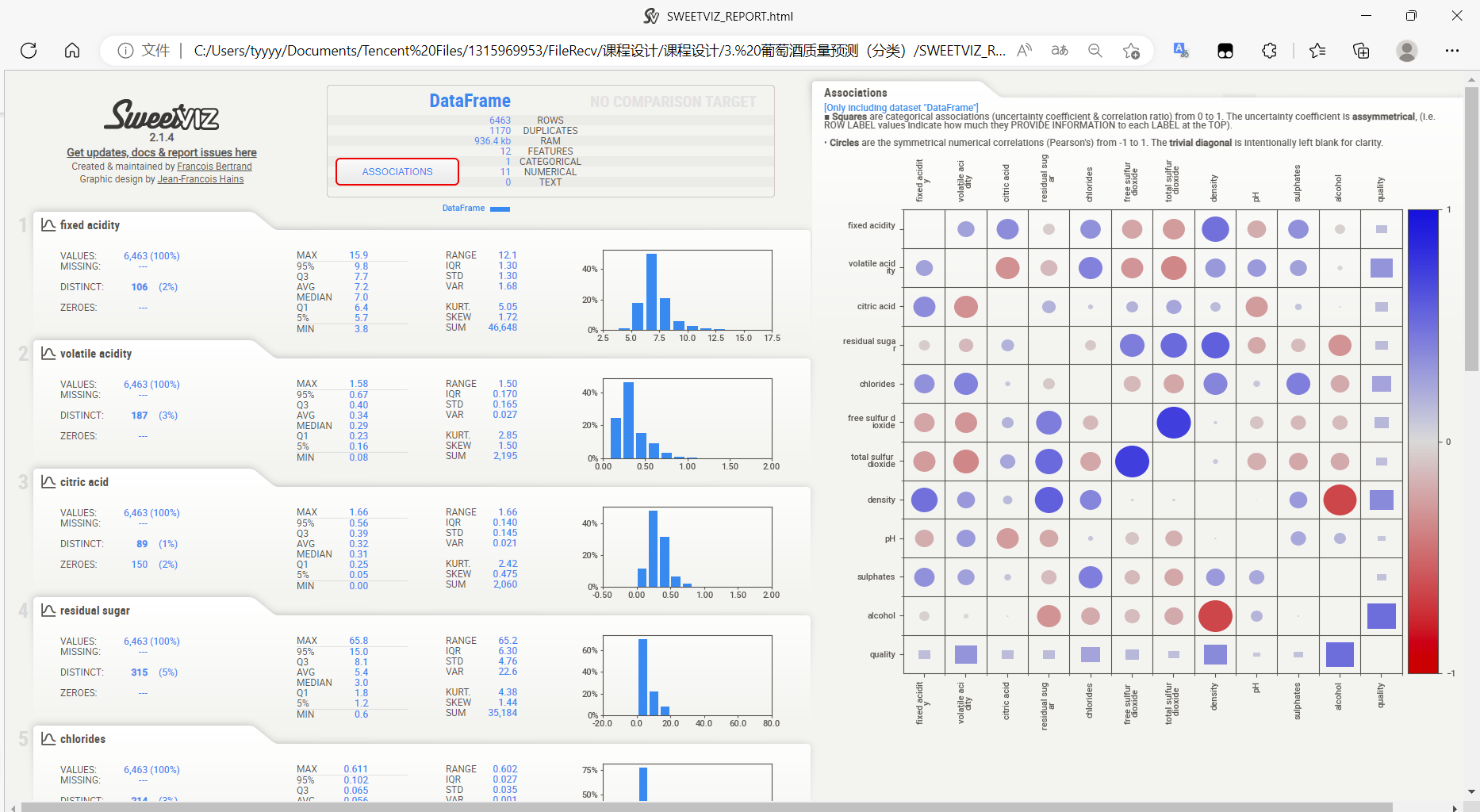


2.2 sweetviz

它的输出是一个完全独立的HTML应用程序。可以很直观地看到每个变量特征的统计数据和直方图。

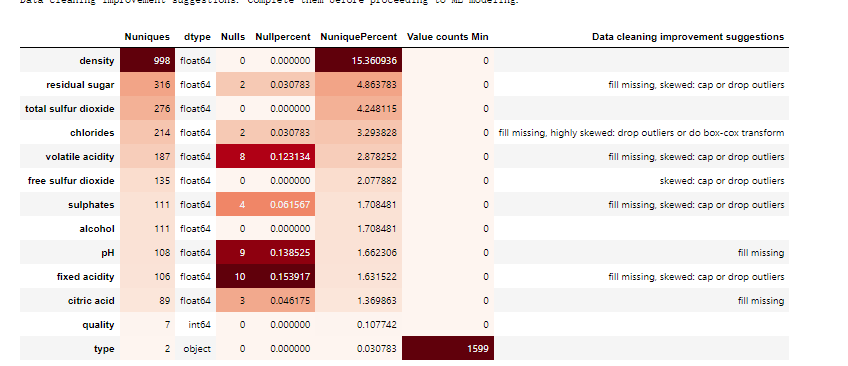


同时也方便查看联系：



2.3autoviz

和pandas\_profiling一样，都是在下方输出框中显示报告结果。

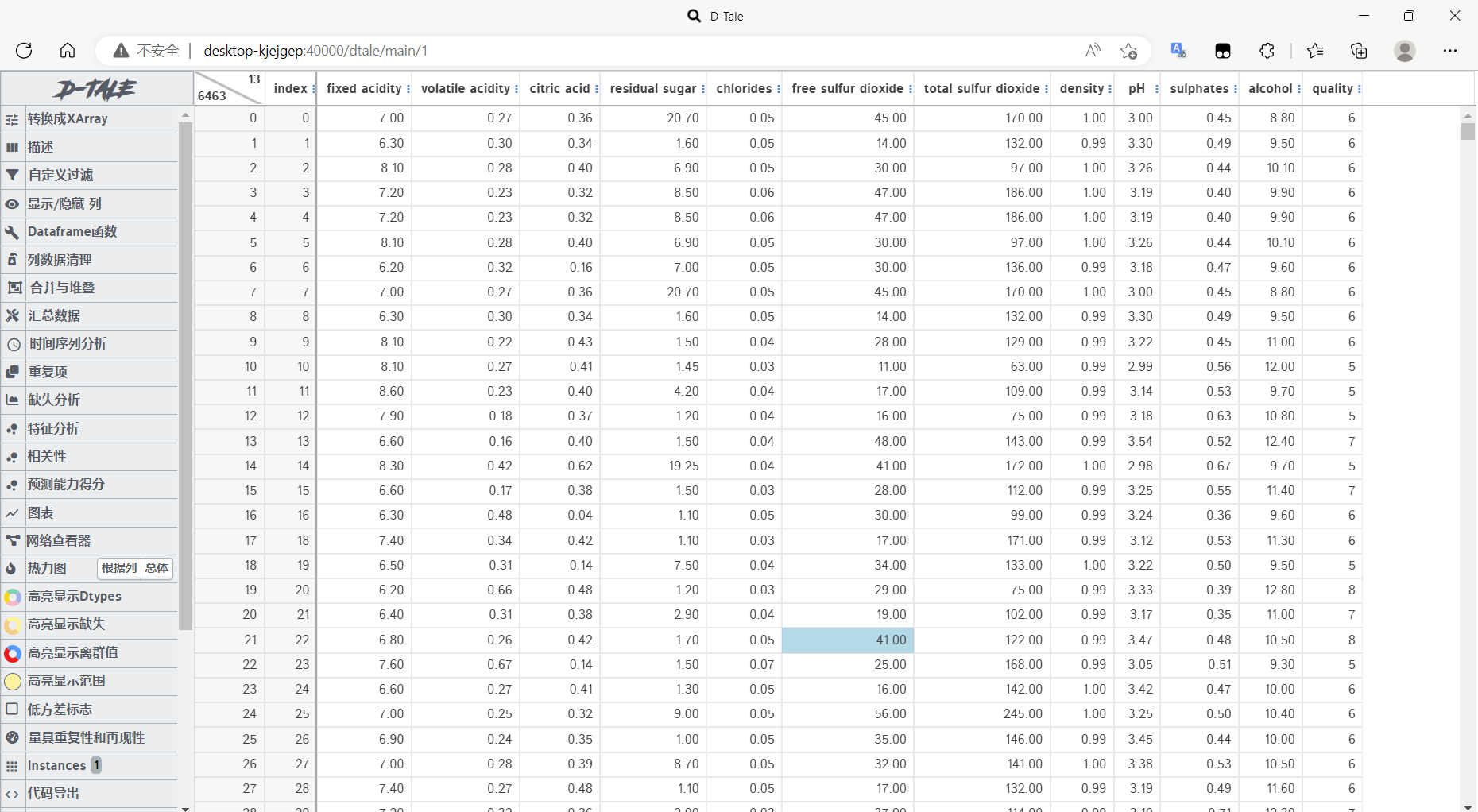


2.4DTale

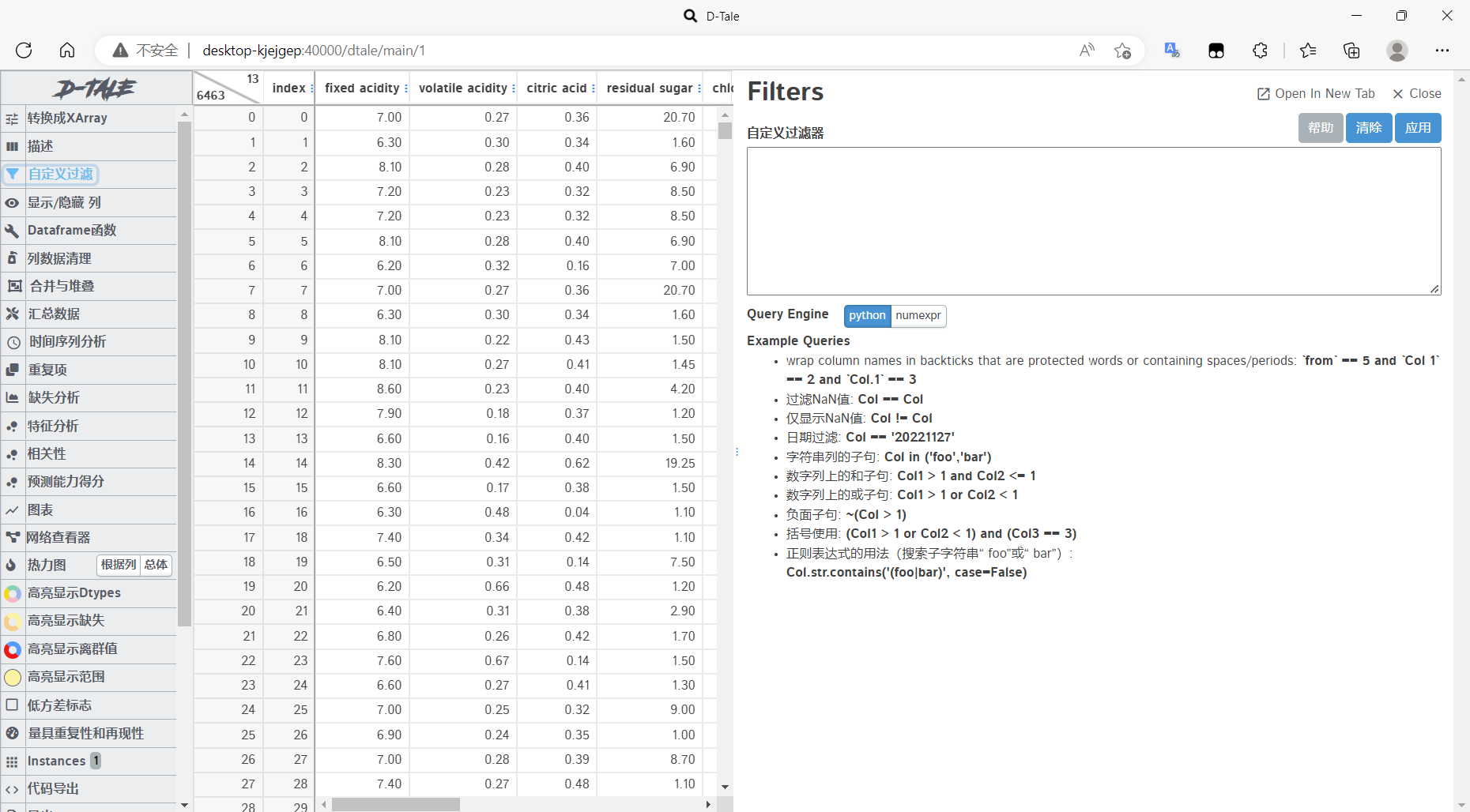
DTale是我目前发现的功能最齐全的EDA插件，涵盖describe, filter, Dataframe Functions, heatmap等多种功能，并且还能调整界面设置，更改语言选项，改变配色环境等。

可以选择输出形式，这里我把它用网页打开。

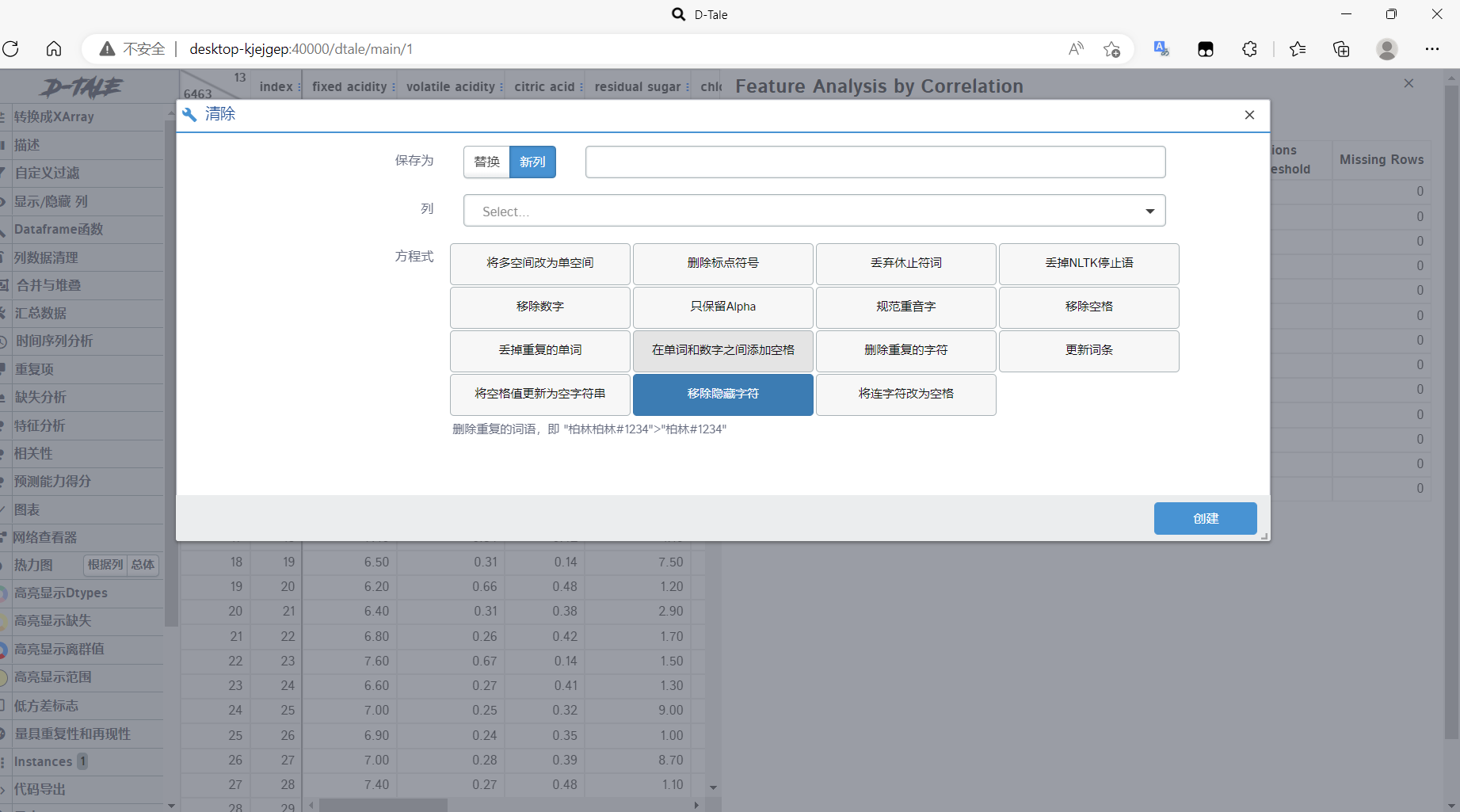




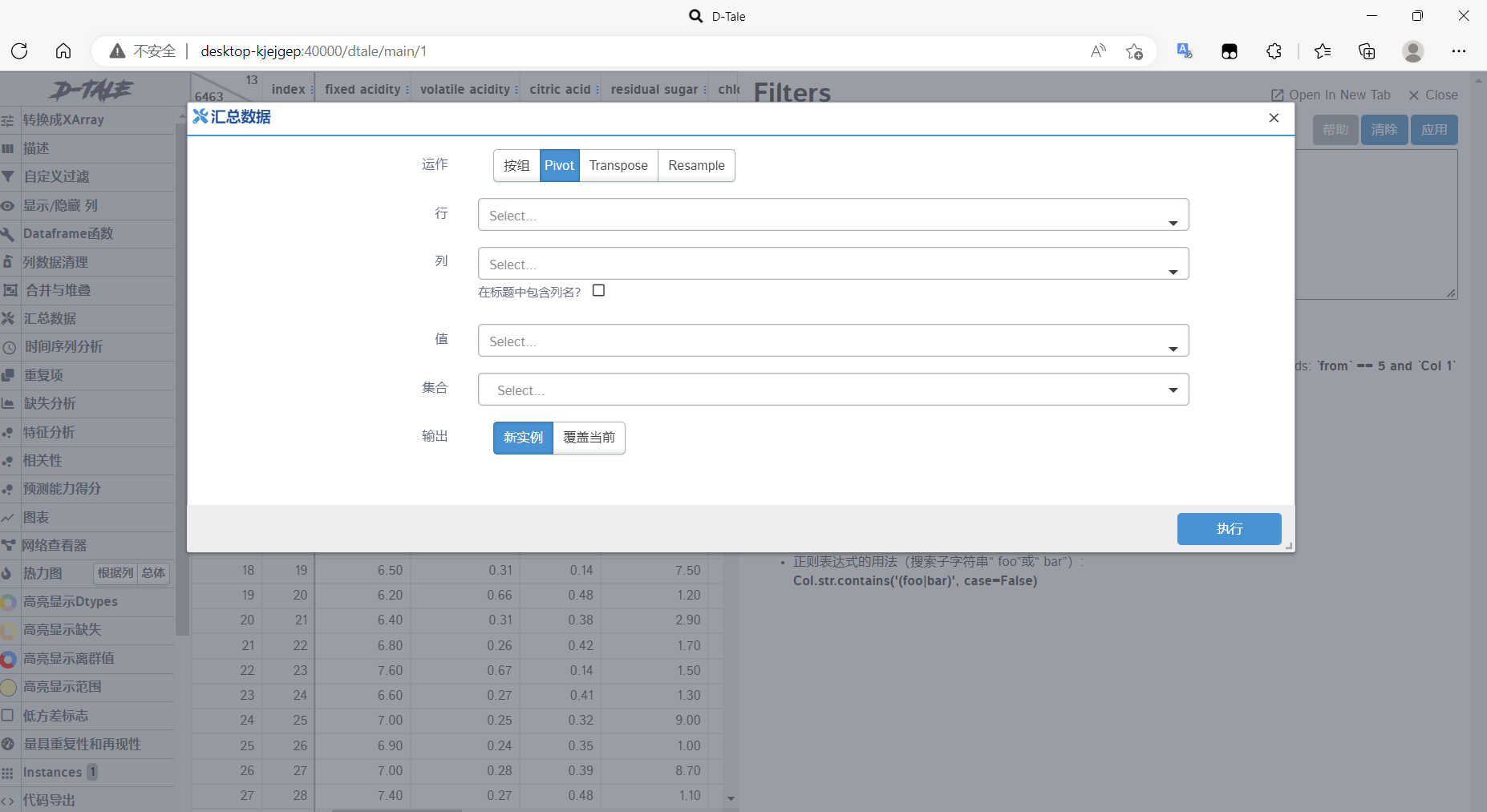
能实现自定义过滤;



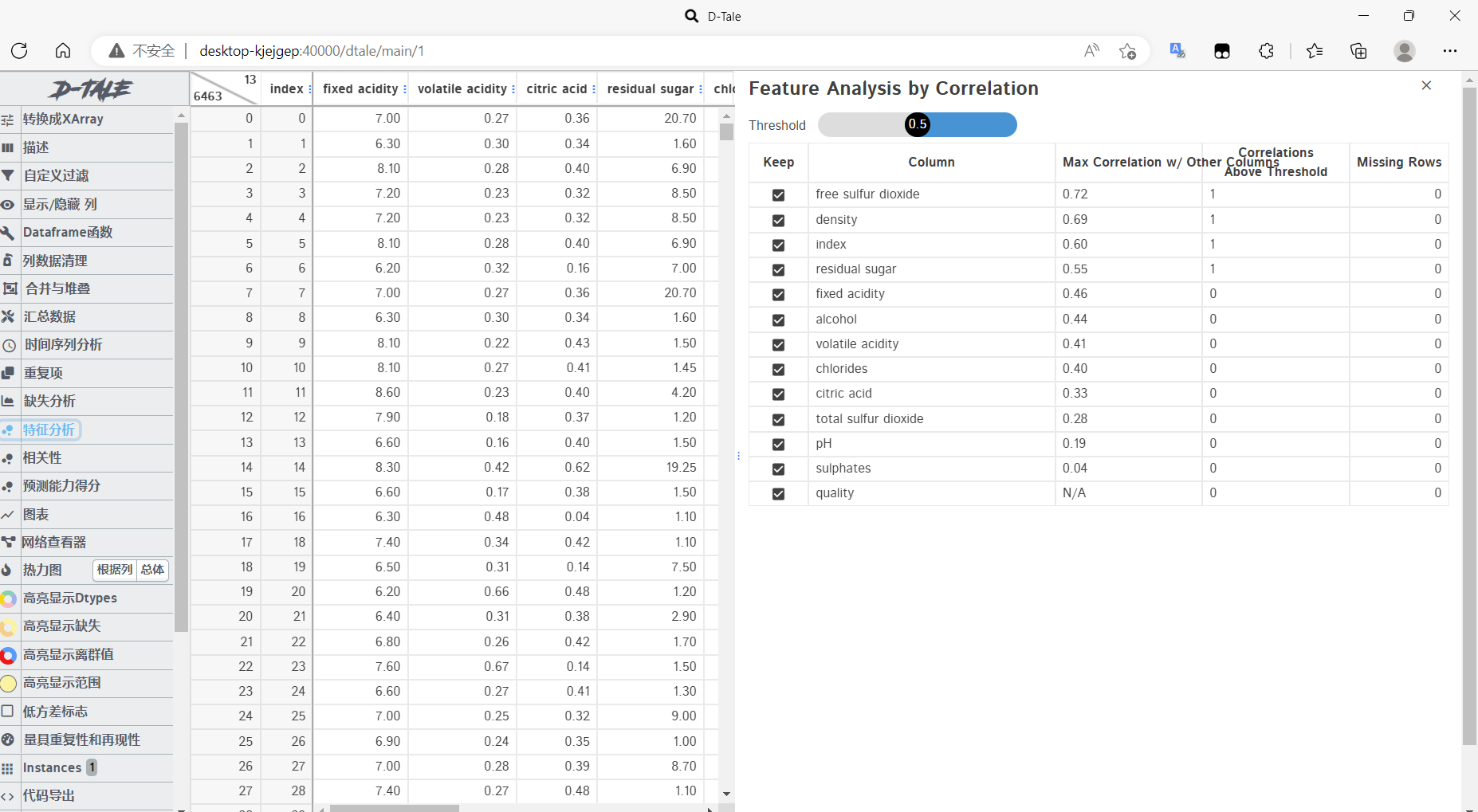
能做数据清理：



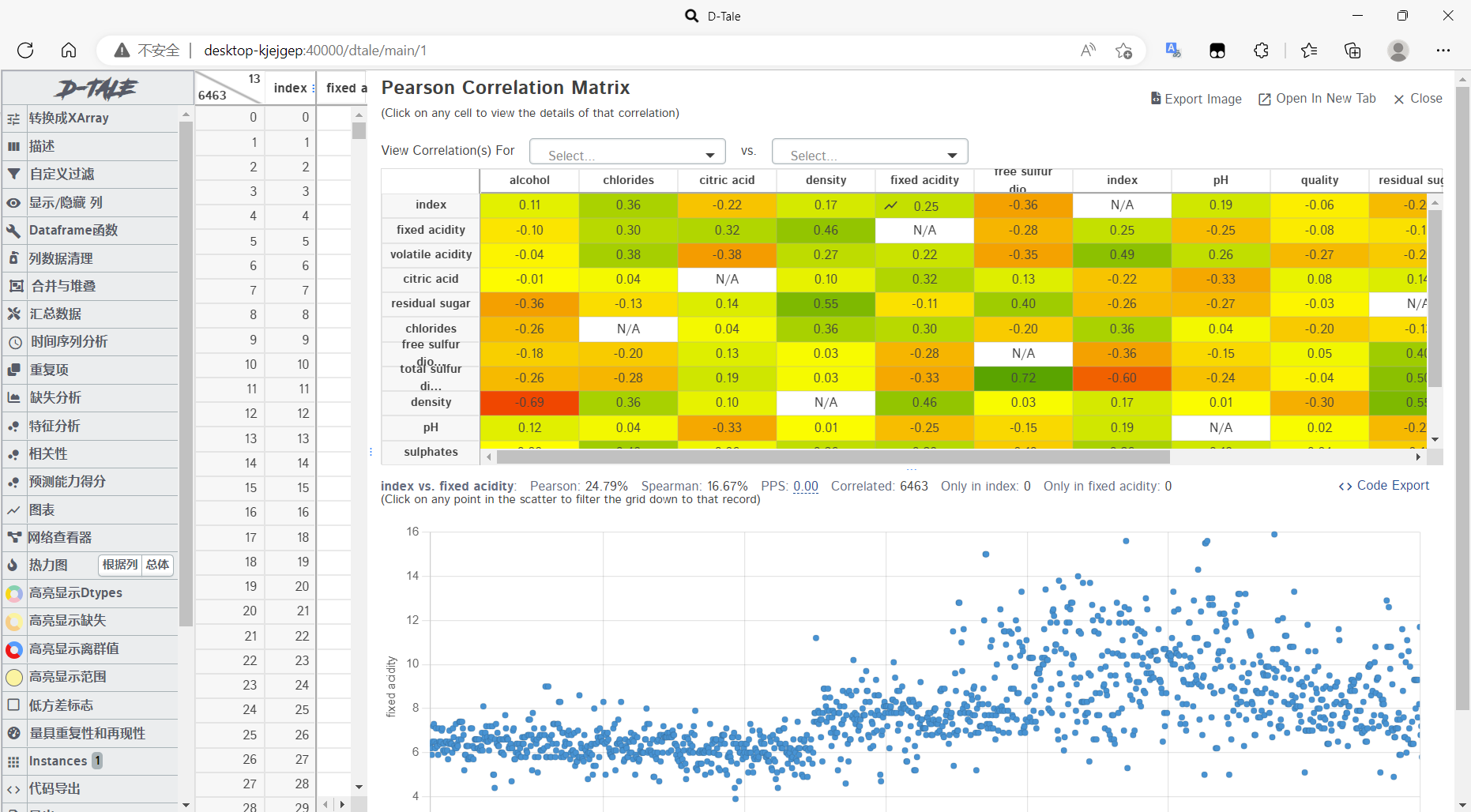
能汇总数据：



能特征分析：



能探索变量间相关性：



能预测能力得分：

能高亮显示离群值：（因为数据量过大，过程有些缓慢，我没有加载出来）

还有很多我没有摸索完全的功能，真的很有作用，帮助人类解放双手了。

需要注意的是我安装DTale的时候不能仅仅在Anaconda Prompt里pip安装，会有权限不够运行不了的情况，应该在Anaconda Powershell Prompt里安装。

### 1.3特征工程

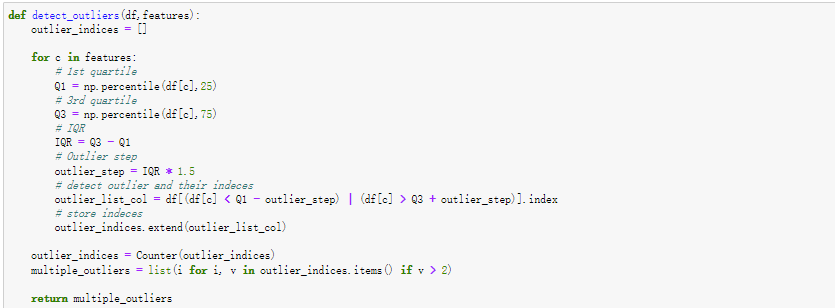
数据可视化结束后，我们来进行一下特征工程：

#### 1.3.1处理异常值

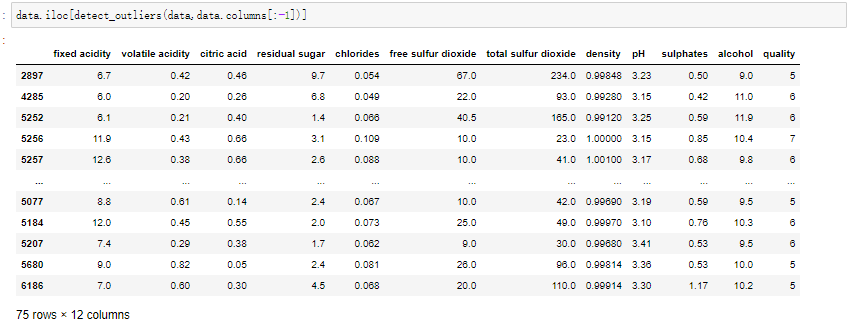
在第一轮的尝试中，我已经处理了缺失值，但在可视化过程中我发现明显存在一些异常值。接下来的工作就是检测和处理异常值：

检测异常值：

利用箱型图的四分位距（IQR）对异常值进行检测，四分位距(IQR)就是上四分位与下四分位的差值。而我们通过IQR的1.5倍为标准，规定：超过上四分位+1.5倍IQR距离，或者下四分位-1.5倍IQR距离的点为异常值。



检测到有75行异常值。

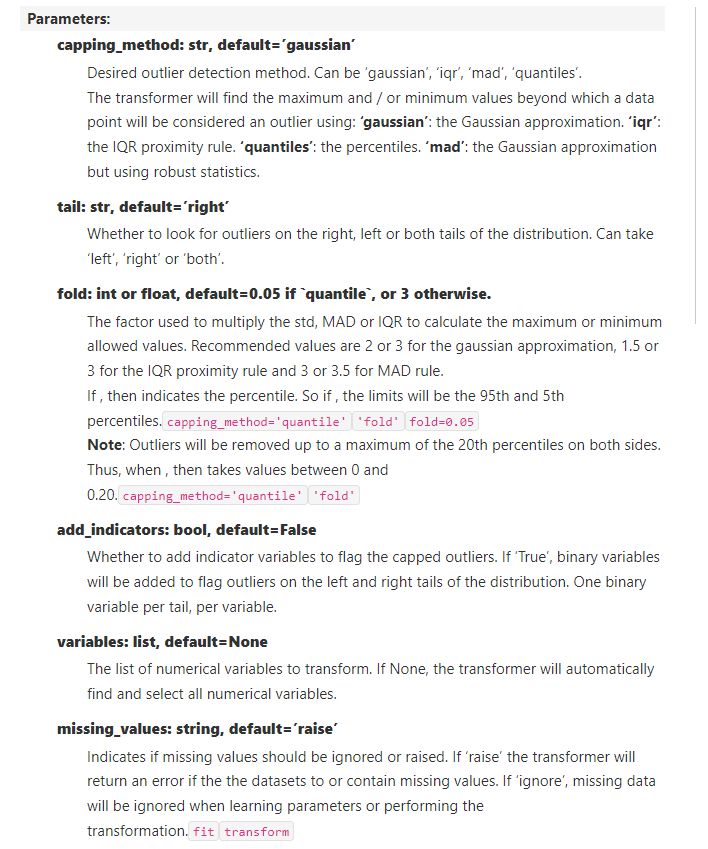


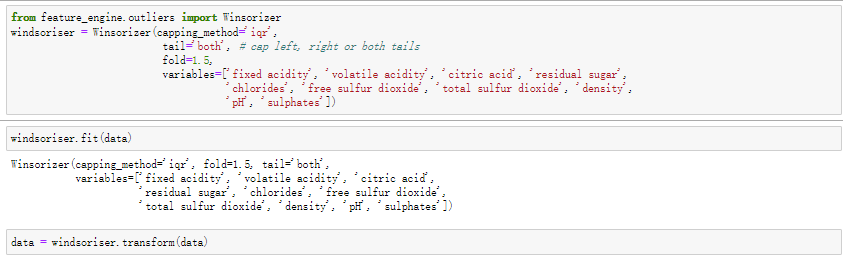
处理异常值：

通过模块feature\_engine里的Winsorizer()来处理异常值。

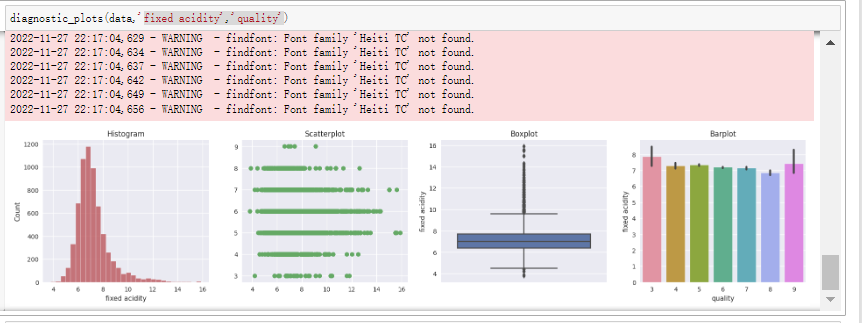
Class feature\_engine.outliers.Winsorizer(capping\_method='gaussian', tail='right', fold=3, add\_indicators=False, variables=None, missing\_values='raise')[source]

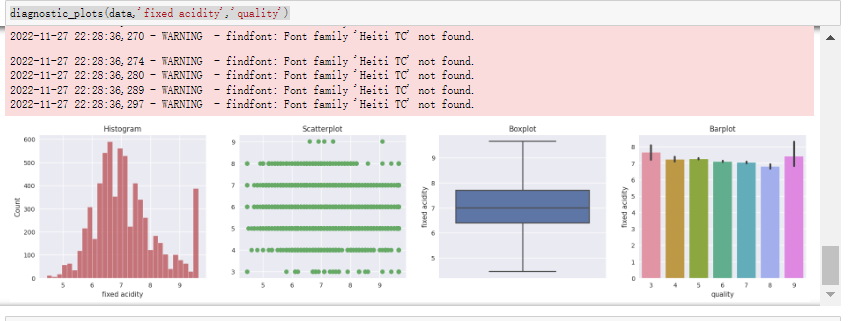
具体参数说明见下图：





进行缩尾处理后fixed acidity 和quality视图的对比：





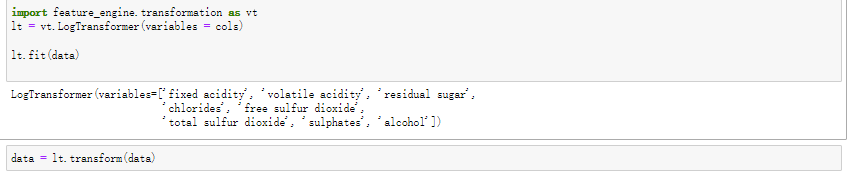
可以看出异常值明显减少。

#### 1.3.2处理偏移量

处理左偏、右偏：

因为除了pH值其他变量都有一定程度的左偏或者右偏，所以取这些有偏度的列进cols。





同样用到了模块feature\_engine里的方法，对数变换。

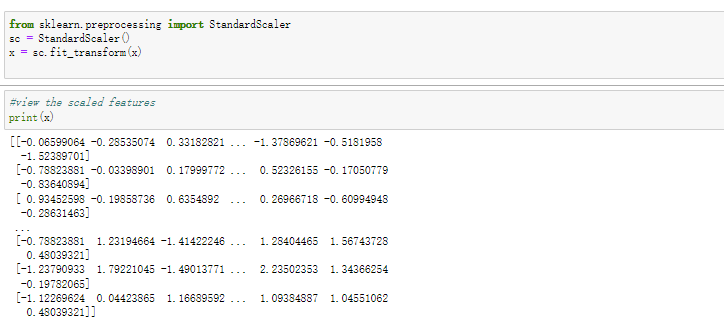
class feature\_engine.transformation.LogTransformer(variables=None, base='e')



#### 1.3.3数据标准化

使用标准差标准化 StandardScaler来进行：

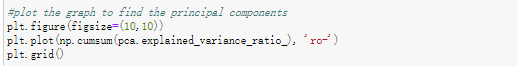
StandardScaler处理方法：标准化数据减去均值，然后除以标准差，经过处理后数据符合标准正态分布，即均值为0，标准差为1。基本可用于有outlier的情况，但在计算方差和均值时outliers仍然会影响计算。

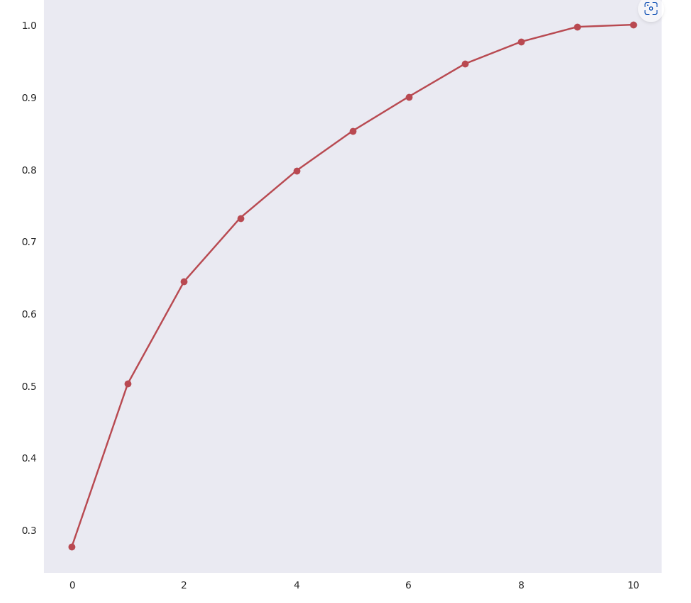


#### 1.3.4特征降维

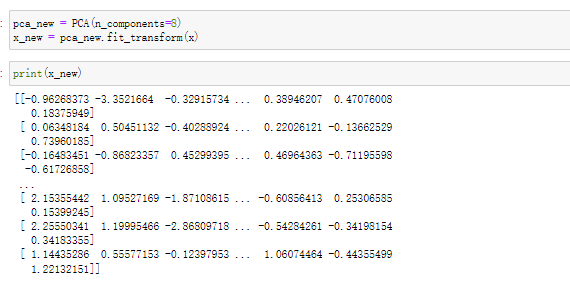
使用PCA降维。PCA降维的本质是对原矩阵的每一维特征向K个基向量投影，使得投影的结果在每个基向量上最离散，从而保留最多的信息。







在不超过1.0的情况下principal components 为8 的时候效果最好，所以参数设置为8。



除了PDA降维，我了解到，还有一种降维方法叫LDA降维：

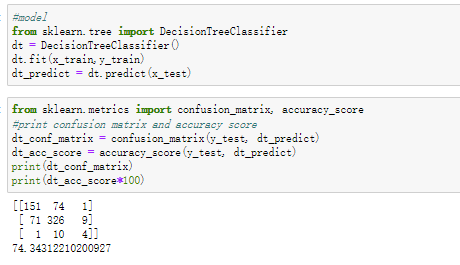


因为PCA效果已经不错了，所以这里就不采用LDA了。

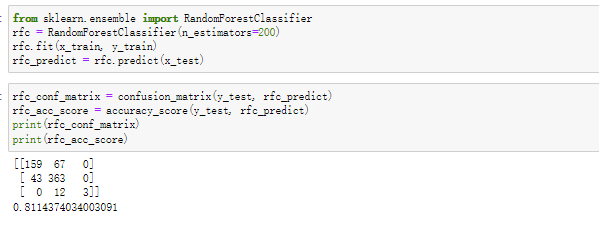
### 1.4再次建模

同第一次尝试一样, 模型采用决策树、随机森林。

#### 1.4.1决策树



#### 1.4.2随机森林



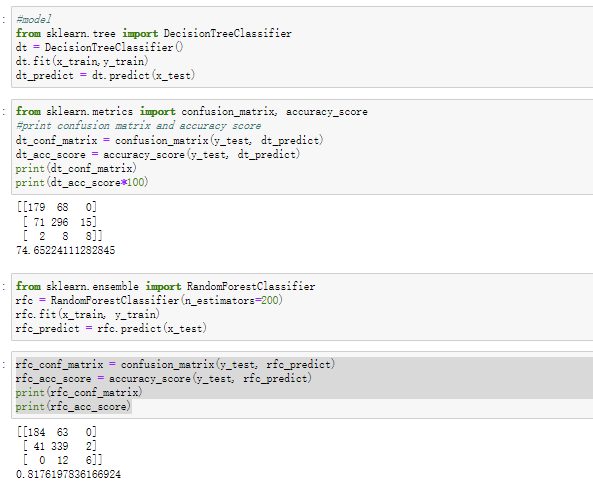
### 1.5尝试其他方法

#### 1.5.1处理偏移值BoxCoxTransformer

之前采用的是LogTransformer，尝试BoxCoxTransformer。

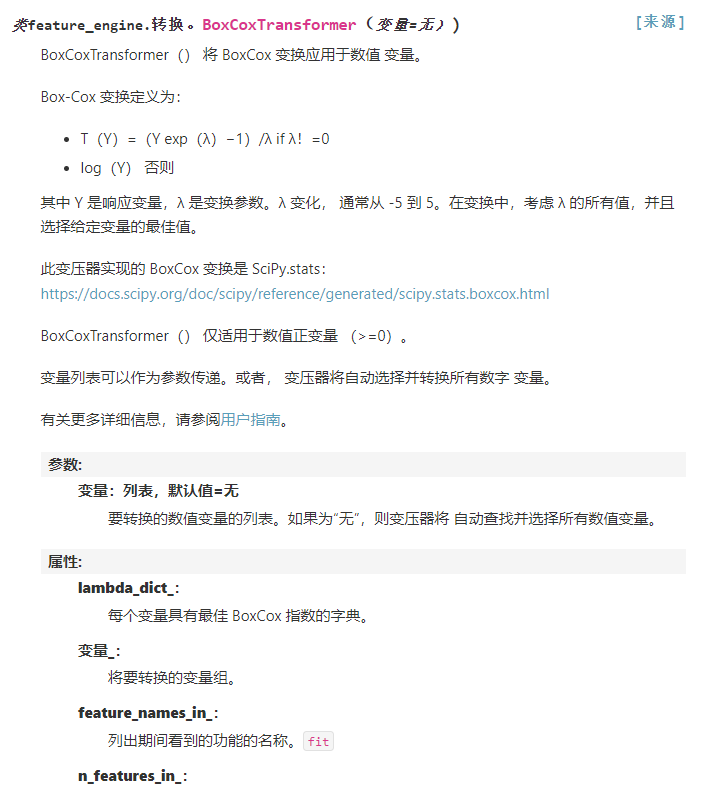


建模评估结果：

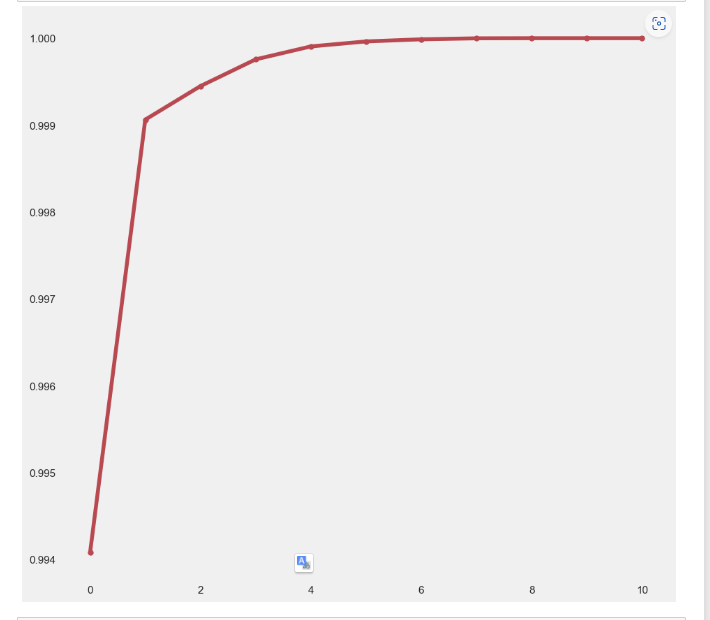


有些微提升。

Box Cox 变换的核心是一个指数，λ （λ），它从 -5 到 5 不等。考虑 λ 的所有值，并选择数据的最佳值;“最优值”是导致正态分布曲线的最佳近似值的值。

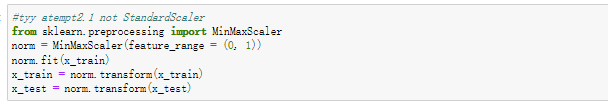


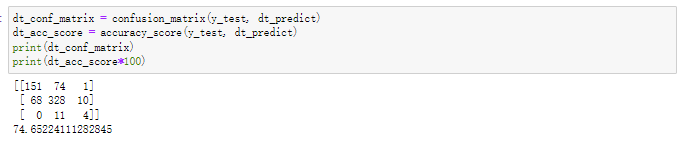
注意用它变换后，PCA的最佳component变为1了，不能依旧按照原来的8给参数，否则会降低准确率。



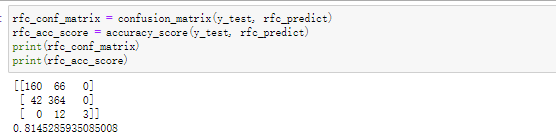
#### 1.5.2 数据标准化MinMaxScaler

如果在数据标准化时不采用StandardScaler而是MinMaxScaler结果会更好吗？





决策树有0.2%的提升。



随机森林也有些微提升。

可以看出这种数据标准化处理方法比之前更好。

我找了一下它们的区别资料：

StandardScaler：

适用性：适用于本身服从正态分布的数据；

Outlier 的影响：基本可用于有outlier的情况，但在计算方差和均值时outliers仍然会影响计算。

MinMaxScaler：

适用性：适用于分布范围较稳定的数据，当新数据的加入导致max/min变化，则需重新定义；

Outlier 的影响：因为outlier会影响最大值或最小值，因此对outlier非常敏感。

### 1.6交叉验证

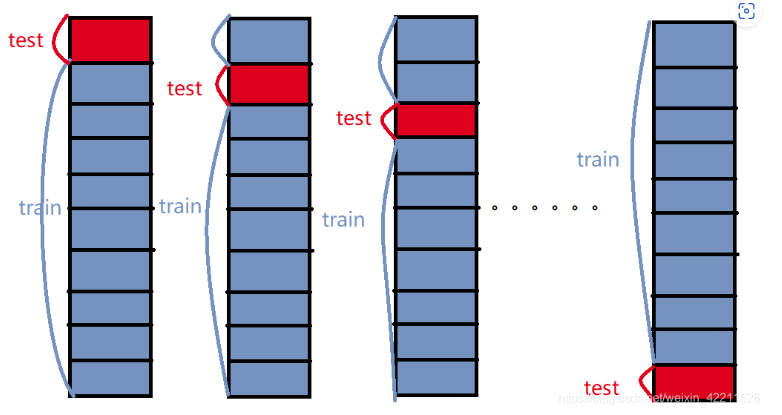
做完这一切以后我们尝试一下交叉验证检验模型。交叉验证用于评估模型的预测性能，尤其是训练好的模型在新数据上的表现，可以在一定程度上减小过拟合；还可以从有限的数据中获取尽可能多的有效信息。

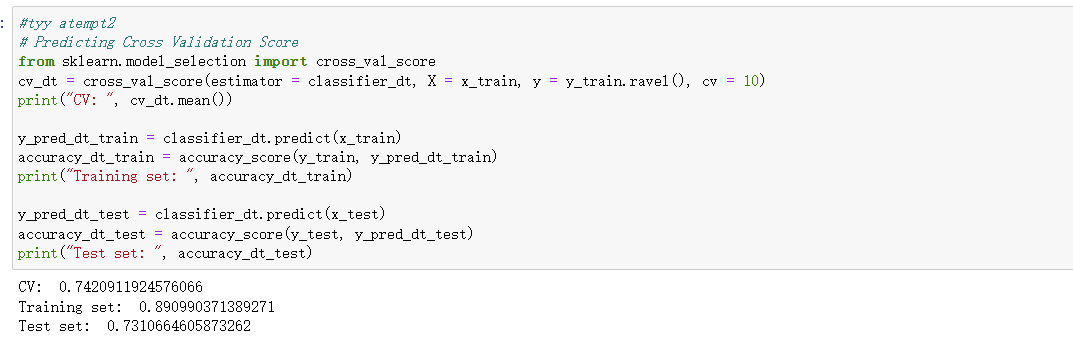
estimator:估计方法对象(分类器)

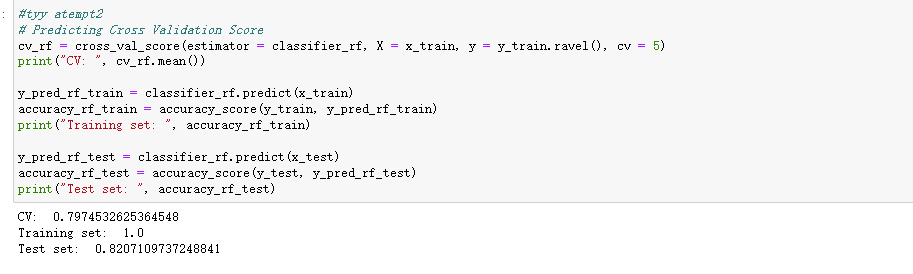
X：数据特征(Features)

y：数据标签(Labels)

cv：几折交叉验证





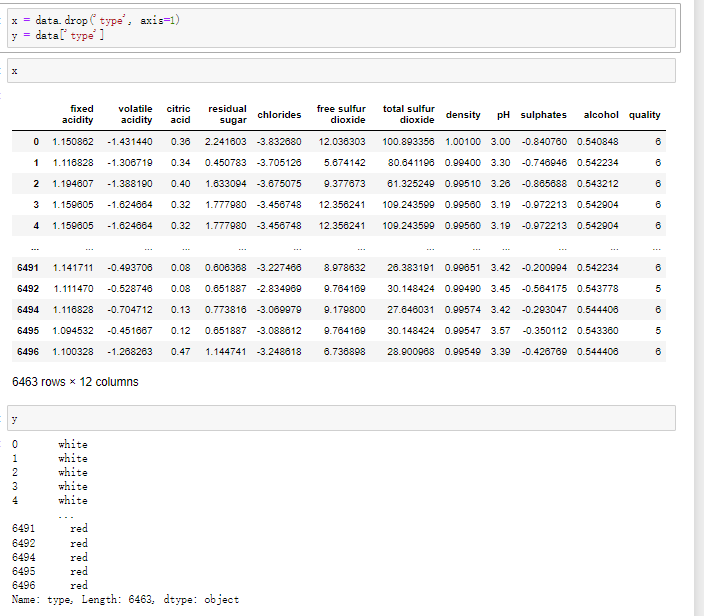


结果看起来都还不错，尤其是随机森林。

## 2.葡萄酒类型分类

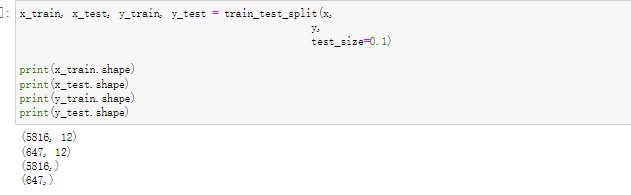
对于葡萄酒类型分类问题，由于预处理和数据可视化部分与葡萄酒质量分类问题类似，这里就不再过多阐述。

### 2.1划分x、y



### 2.2划分训练集和测试集

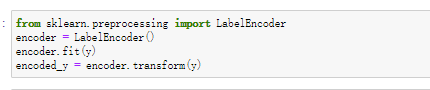
初步划分训练集和测试集：



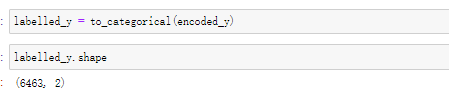
因为y的元素不是数值，而且还没有label，所以列项会给后面带来麻烦，需要转换为编码，这里使用独热码编码。

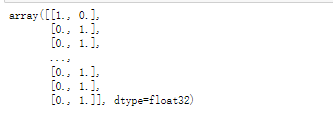
### 2.3特征编码

先使用sklearn里的标签编码：



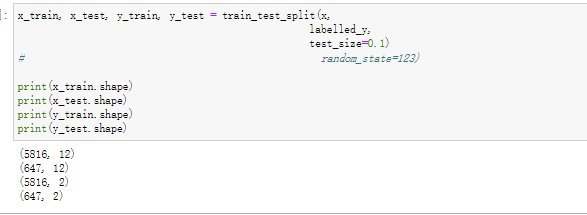
再把标签编码转换为独热码：（否则之后的model.fit会报错）





这个时候看y的元素，就符合传参条件了。

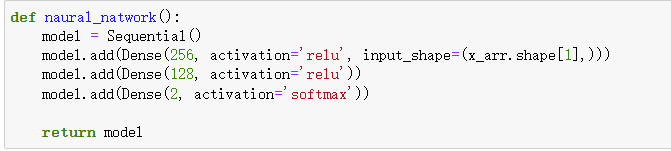
再划分训练集和测试集：



### 2.4建模

采用深度神经网络模型来做：

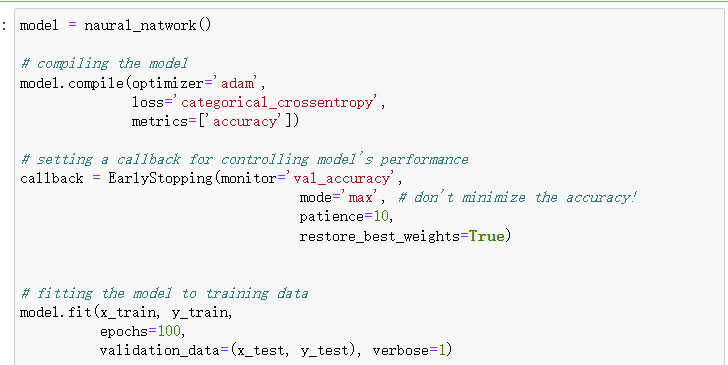
使用 Sequential() 搭建模型，再使用model.add一层一层往上加。这样做的简便之处是，除第一层输入数据的 shape 要指定外，其他层的数据的 shape 框架会自动推导。



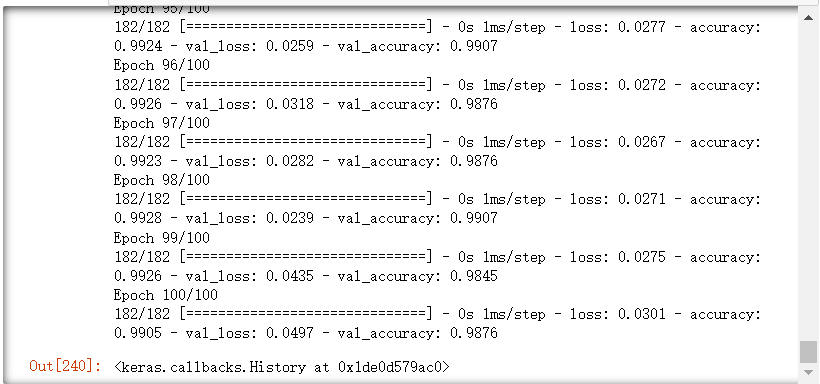
在模型使用前必须编译，否则在调用fit或evaluate时会抛出异常。

model.compile()方法用于在配置训练方法时，告知训练时用的优化器、损失函数和准确率评测标准。这里使用Adam优化器，损失函数用了categorical\_crossentropy（ps因为之前没有进行独热码编码的时候我尝试过用数值编码，损失函数也可以使用sparse\_categorical\_crossentropy）评估函数使用的是accuracy。

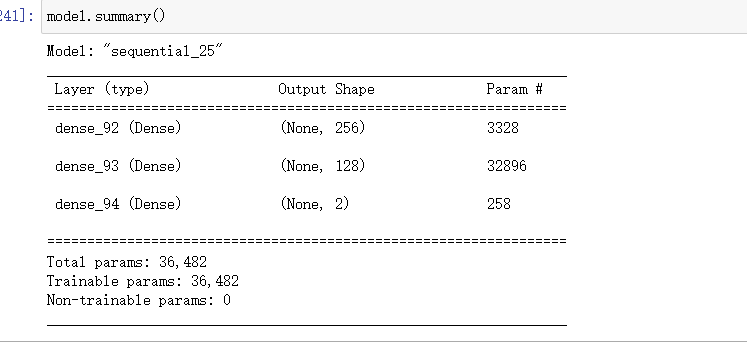
因为epochs 越大，最后训练的损失值会越小，但是迭代次数过大，会导致过拟合的现象。所以我们往往希望当loss值，或准确率达到一定值后，就停止训练。所以定义一个Callback函数，它调用了tensorflow里的Early stopping函数。其中monitor是监控的数据接口，因为有验证集，所以就用val\_acc给它赋参，而patience是指能够容忍多少个epoch内都没有improvement，这里我尝试了几次，效果都挺好的。因为我们的monitor是val\_acc，所以mode就选择max，而不是auto或者min。restore\_best\_weights：当发生EarlyStopping时，模型的参数未必是最优的，即monitor的指标未必处于最优状态。如果restore\_best\_weights设置为True，则自动查找最优的monitor指标时的模型参数。



结果：



最后，使用keras构建深度学习模型，我们会通过model.summary()输出模型各层的参数状况。



### 2.5可视化训练结果

用history使得训练结果可视化。注意一个keras版本问题，老版关键字是 ‘acc’，而新版是‘accuracy’。

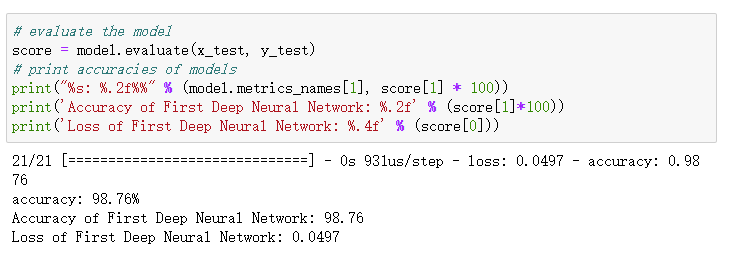




可以看到本身精度就还不错了，epoch次数不高的时候就能接近100%了。

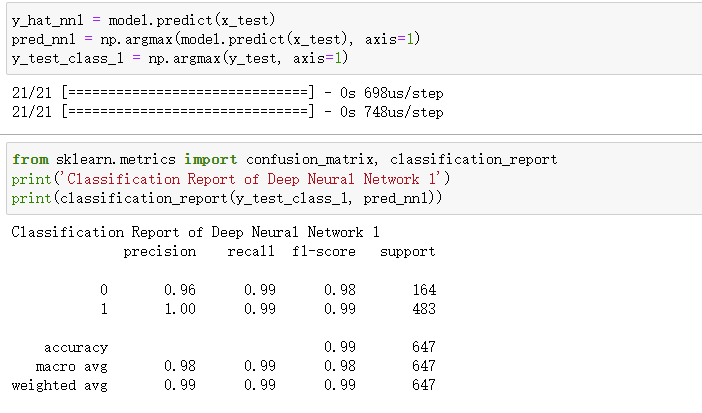
### 2.6评估性能

用evaluate 模型评估，返回一个测试误差的标量值。



可以看到，效果还是相当不错的，准确率达到98.76%，而损失率只有0.0497。

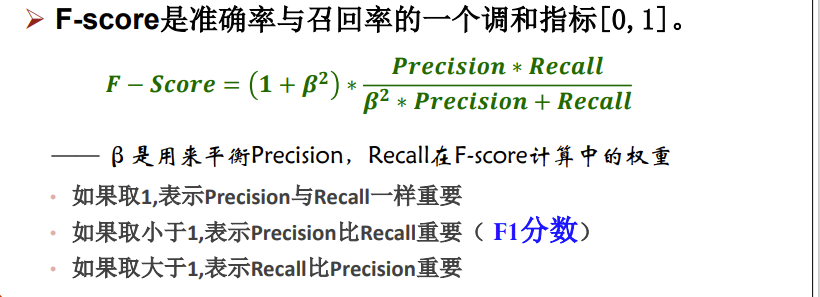
如果用model\_ predict模型评估, 用classification\_report()函数输出模型评估报告：



可以看到准确率分别问0.96和1.0，召回率都为0.99，f1-score分别为0.98和0.99，再次证实这个模型确实很不错。

这里要提到f1-score：

它比其他的评估信息从名称上来看不那么直接易懂，刚好今天上课的数据科学导论也提到它了。它就是是准确率与召回率的一个调和指标：



## 3.心得体会

这次大作业耗费了我比较长的时间来完成，因为对于机器学习我是没有太多基础的，只是大二在接触大创的时候粗略了解了一下相关的知识，看了一点点的pytorch语法，但是都只停留在理论层面。我觉得理论层面的机器学习还是比较枯燥的，因为它比起计算机知识更像是数学理论，看着一堆公式的变换我只觉得头疼，我感觉如果不是热爱肯定很少有人能坚持学下去，但是实际动手来做一个项目就有趣多了。感谢老师的安排，比起背书背公式我更喜欢去尝试实践，就像动手搭积木一样，一点儿一点儿地堆上去就可以了，看着成型的积木是很有成就感的。调参数虽然我不是很会，但我觉得很有趣，就像是有人爱抽盲盒一样，这种不确定性带给我很多乐趣哈哈，因为老师并不会把模型的得分计入分数，所以做这个事情没有压力，只是试探着摸索，而探索这件事情本身就很快乐。

通过这次动手实践的大项目，我确实学到了很多东西，导入数据，数据探索，数据可视化，在可视化这方面我做了比较多的工作，因为看着输出的图片形状很有意思，雷格图箱型图提琴图热力图等等怎么输出怎么看，我都学会了，而最后了解到的EDA插件，更是帮助人类解放双手，我感觉要是熟练用会了这些插件，不仅在科研项目中，在工作中也是一个大杀器。而数据处理部分，我了解到了很多种方法来处理，并且因为顺藤摸瓜挖到了feature-engine这个官方网站（https://feature-engine.readthedocs.io/en/latest/user\_guide/index.html），配合翻译器我了解到它提供了很多数据处理的方法，举几个例子，在处理异常值的时候我采用的是Winsorizer的功能，其实还有ArbitraryOutlierCapper、OutlierTrimmer的办法来处理，而处理扭曲、偏移，使之正态分布时，除了LogTransformer、BoxCoxTransformer，我看见官方还提供了LogCpTransformer、ReciprocalTransformer、ArcsinTransformer、PowerTransformer、YeoJohnsonTransformer这些转换方法，还有降维操作，虽然我们这个数据集特征数量本身不大，降维的效果不是很明显，不过因为想学习更多知识所以也去学习了，除了PCA降维，还有LDA降维。而后面的建模，在了解各个参数怎么用怎么传的时候我也学到了不少，尤其是深度神经网络模型，以及模型的评估，包括一知半解的交叉验证、混淆矩阵，准确率召回率的概念，在动手实践的过程中我都清晰了不少。

这一次的实践项目，我觉得不光提升了我对机器学习本身知识的理解，更培养了我获取知识，查阅英文文献的能力，因为我在准备留学，所以目前英语水平是比过去而言提升了的。以前我对英文网站就抱有一种恐惧，老师经常让我们去官网查阅帮助手册，可是我看着陌生的字符就头疼，但现在因为学习的需要，以及自身水平的提升，我慢慢地去接触这些从前我害怕的东西，我发现也并没有很困难，相反，很多信息比国内的百度问答、CSDN等网站简明易懂地多。包括一些问答网站，国外用户的问题问得太全面了。几乎解决了我80%的bug问题。我真的觉得这次的实践项目很不错！可以流传下去！方便学弟学妹们深入了解机器学习。