2022 年北华大学数学建模竞赛

承　诺　书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题.

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出.

我们郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性. 如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理.

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）.

我们参赛选择的题号是（从 A/B/C 中选择一项填写）：　C　

我们的报名参赛队号为：　42　

所属学校（请填写完整的全名）：　北华大学　

参赛队员 (打印并签名) ：1.　卢佳　

2.　郑国正　

3.　陶李涛　

指导教师或指导教师组负责人　(打印并签名)：　曹名圆　

（论文纸质版与电子版中的以上信息必须一致，只是电子版中无需签名. 以上内容请仔细核对，提交后将不再允许做任何修改. 如填写错误，论文可能被取消评奖资格.）

日期：　2022　年　05　月　14　日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2022 年北华大学数学建模竞赛

编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

西班牙葡萄酒的价格预测

摘要

本文针对西班牙葡萄酒的价格数据集，以预处理后的数据集为基础，依据K-means++聚类分析将7500种葡萄酒分类，并且建立XGBoost预测模型，来确定数据集中编号为7001-7500葡萄酒的价格.

针对问题1，通过Python处理7500种葡萄酒的价格数据，从中查找出year、type、body和acidity这4个指标含有缺失值，并对缺失值进行处理；根据均值、方差、最值和四分位数进行统计性描述；利用Python进行数值型变量间的相关性分析，得出多数数值型变量在价格上没有太多的相关性，除了用户评分有一个弱到中等的正相关性.对于类别性变量，采用词云图的方式分析.

针对问题2，运用肯德尔系数确定其他指标对三种评分的影响，结合气泡图与箱图分析用户评分和各变量之间的关系；绘制气泡图反映评分与价格的相关性，结合SPSS建立回归方程确定评分与价格的关系，得出rating、acidity和body自变量的系数分别为721.08、12.34和18.27，所以rating对价格的影响更显著.

针对问题3，基于K-means聚类将7000种葡萄酒进行分类.观察不同k值对应的簇内误差平方和，绘制碎石图得出最优k值等于4；通过轮廓系数对k值进行评估检验，结合散点矩阵图得出主要变量之间具有一定的独立性；最终绘制饼图刻画分组聚类后各个组的占比情况，其中占比最大组为94.3%，占比最小组为0.1%.

针对问题4，构建基于Xgboost葡萄酒价格预测模型.选取mse作为评价指标，按照7：3将数据集进行，使用autogluon工具包，选择KNN、Catboost、Lightgbm等多个机器学习模型，结果显示Xgboost的效果最好，其平均绝对误差为-9.5133，根据奥卡姆剃刀原则，选用效果最好的极限梯度提升树来预测葡萄酒价格.

关键词：相关性分析, K-means聚类，轮廓系数, XGBoost预测模型

目录

1. 问题重述

1.1 问题背景

葡萄酒作为一种较为保值的产品越来越受到投资者的青睐,对其价格的预测也越来越引起人们的重视.所以我们将利用多种预测模型, 根据处理后的数据对葡萄酒的价格进行预测, 为葡萄酒爱好者的选择提供参考.

图 1 不同品种葡萄酒

1.2 问题的提出

在模型的建立过程中, 主要解决的问题有 4 个.

问题 1 是对数据集进行基础分析.其中包括缺失值处理、各变量的统计指标分析和变量间的相关性分析.

基于问题 1 中处理后的数据, 给出其他指标对三种评分（Acidity、body、rating）的影响并且阐述评分和价格之间的关系.

利用 K-means 算法对 7500 种葡萄酒进行分类.阐述分类的依据、分类方法以及通过轮廓系数与碎石图来评价与分析聚类的效果.

建立葡萄酒的价格预测模型, 根据可决系数 R^2、MSE、MAE 来衡量预测模型的准确性, 并利用模型确定数据集中编号为 7001-7500 葡萄酒的价格.

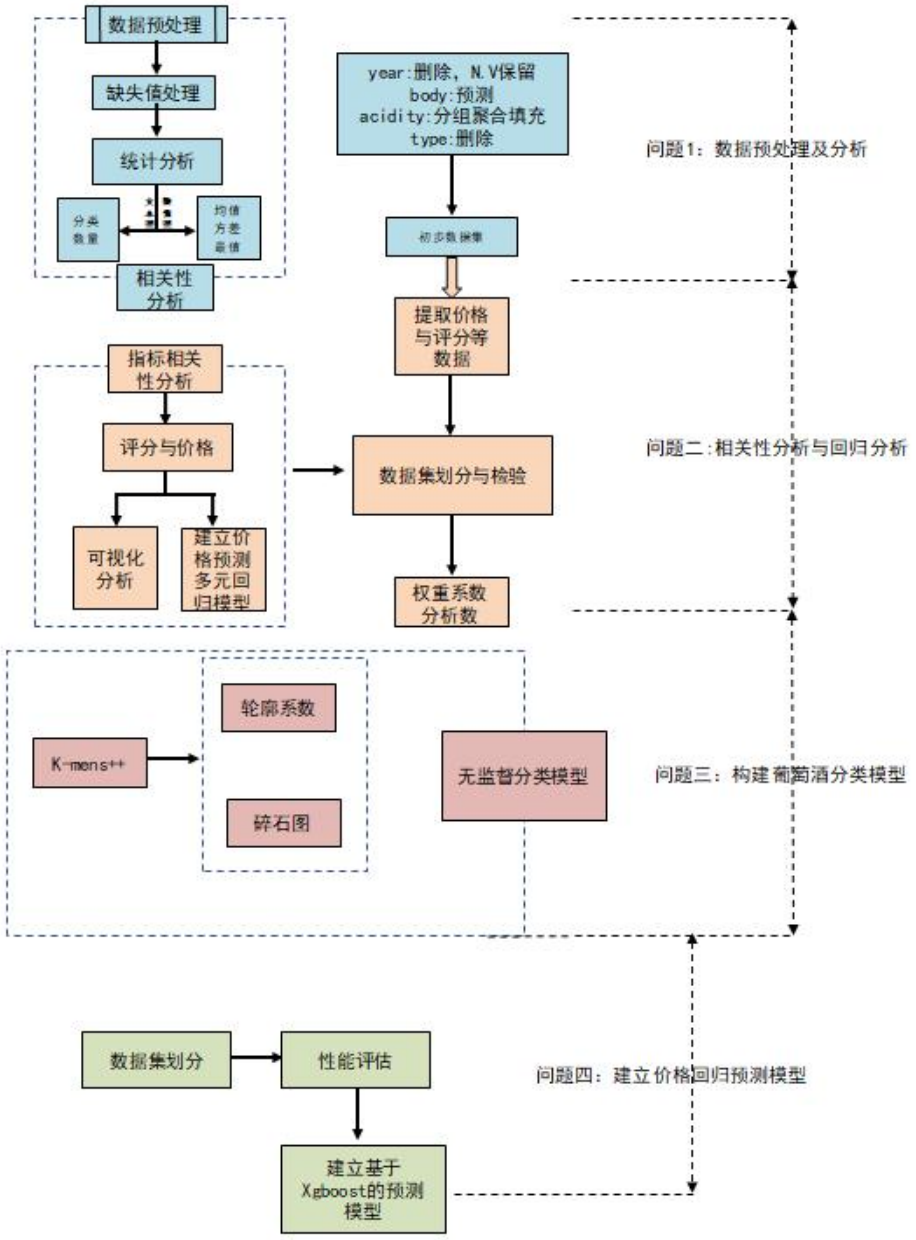

图 2 全文技术路线图

2. 问题分析

2.1 问题 1 的分析

对已知数据集进行缺失值处理.对于 year 类别中的缺失值，删除空白值并保留 N.V.（由不同年份的葡萄酒混合调配而成）作为一类；对于 type 类别，根据酒厂、产地确定分组聚合以众数填充；对于评分 Acidity、body 类别，分离缺失值与实际值，根据实际值预测出缺失值进行填充.对处理后的所有数值型变量进行

相关性分析.

对各变量统计指标分析.对于数值型变量, 根据均值、方差、最值和四分位数进行统计性描述；对于类别性变量, 采用词云图的方式分析.

2.2 问题 2 的分析

利用三维气泡图来讨论其他指标对三种评分指标的影响, 再按照肯德尔系数来进一步分析这种影响；通过四维气泡图初步探讨评分和价格之间的关系, 再使用 SPSS 软件建立回归方程, 比较权重系数的大小, 深入分析三种评分和价格相关的显著水平.

2.3 问题 3 的分析

基于处理后的葡萄酒数据集, 对 7500 种葡萄酒分类, 意味着把具有某些共性特征者予以整合在一起, 再分配到特定的群体, 最后形成许多不同集群的过程.

采用 K-means 算法直接对数据进行聚类分析；通过碎石图来判断聚类的类别数量；绘制饼图来展示分组后各组在 7500 种葡萄酒中所占的比例；最后绘制散点图矩阵及轮廓系数图来评价分类的效果.

2.4 问题 4 的分析

需要构建一个价格预测模型, 根据题目给出的数据, 需要考虑到数据类型和缺失值的相关处理等情况, 因此这里需要考虑的点比较多, 基于此可以考虑自动处理缺失值的树模型等来处理, 可以使用多个机器学习模型进行预测, 取结果较好的部分.

3. 模型假设

(1) 假设原始数据基本准确（个别缺失值数据可处理）；

理由：保证预测模型建立过程中的准确性.

(2) 假设 year 指标中 N.V.为一个时间类别；

理由：N.V.意味着该款酒并不是由某个单一年份所采摘的葡萄酿成的[6], 而是由不同年份的葡萄酒混合调配而成, 为了减少误差, 我们将其看作一类.

(3) 假设数据处理时, 对缺失值填充建立的模型对于完整的样本是正确的；

理由：此模型的建立依赖于剔除缺失值后的实际值, 将部分映射到整体.假设的原因在于能够减少模型带来的误差.

4. 符号说明

符号	说明
V_i	残余误差
S_e	组内离差平方和
S_A	组间离差平方和
ρ	Spearman 相关系数
x_i	表示第 i 个输入的向量
y_i	表示第 i 个输出的值
Y_actual	表示实验测量的值即真实值
$Y\_predict$	表示通过模型得到的预测值

5. 问题 1 模型的建立和求解

5.1 总体特征

对于文本特征，针对类别型需要做单独处理，且文本特征有些特征维度较高，需要进一步探索；对于日期特征，可以做时间特征提取.对于数值特征，可分为离散特征与连续特征，离散特征可以按照类别特征来探索，连续特征可以先查看其分布情况，对于数量级差别较大的数据，进行分箱处理，对于分布不满足正态的，需要进行 *log* 变换.

5.2 缺失值分析

首先进行了缺失值的查找，在已知数据集中的缺失值情况如下图：

图 3 各变量的缺失值

由表中可知，含有缺失值的变量分别是 year、type、body 和 acidity.对于 year 类别，将其空白值进行删除处理，但是在观察数据时发现 year 指标中含有 N.V. 无年份类别，这意味着该款酒并不是由某个单一年份所采摘的葡萄酿成的，而是由不同年份的葡萄酒混合调配而成，所以后续的分析我们将其看作一类.

对于 type 指标，根据 winery（酒厂的名字）、region（葡萄酒的产地）确定分组聚合以众数填充.根据价格绘制排名前 20 位的酒厂、产地和葡萄酒品种，结果如下图所示：

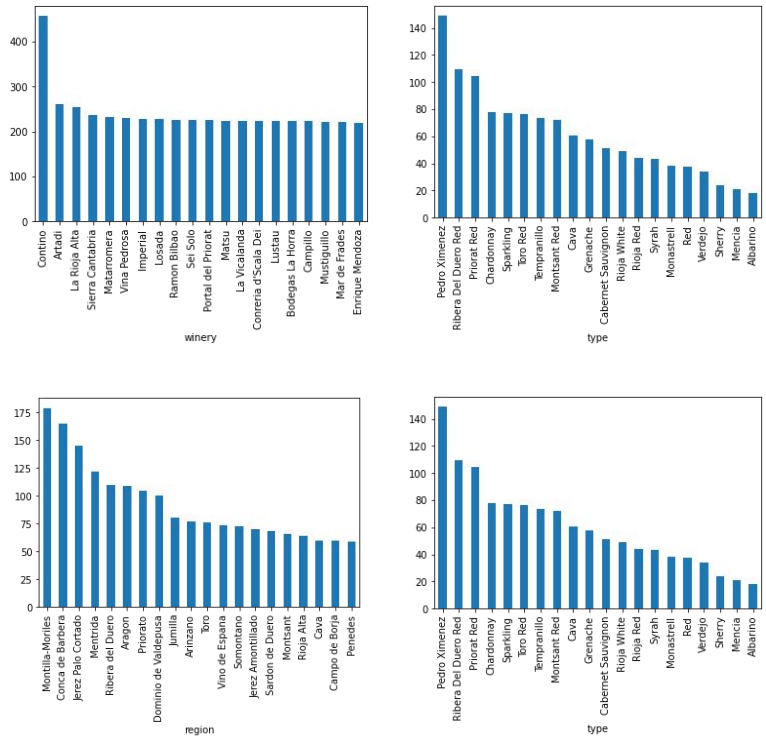

图 4 排名前 20 位的酒厂、产地和葡萄酒品种图

绘制饼图更直观清晰的展现出 type 指标中各个类别所占比例，如下图所示：

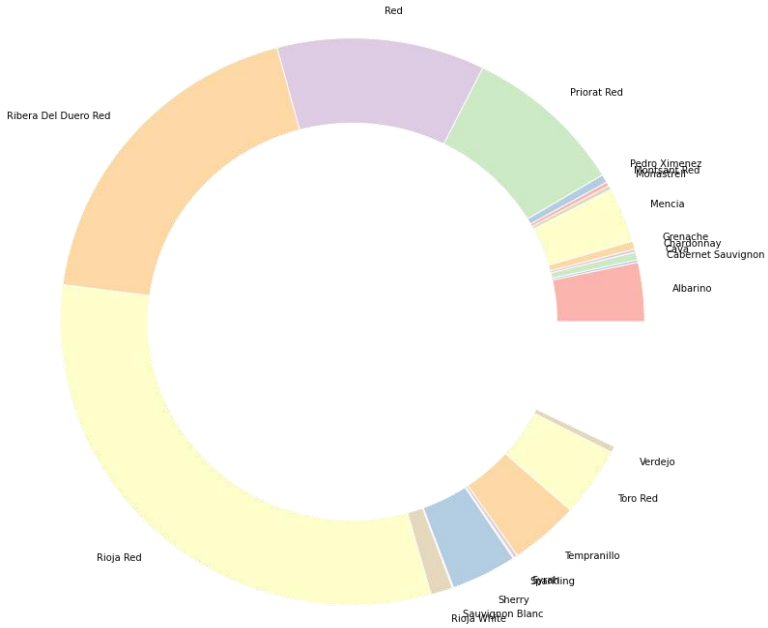

图 5 type 指标中各类别占比情况

对于评分（Acidity、body）类别，分离缺失值与实际值，根据极大似然估计和多重插补的方式，在缺失类型为随机缺失的条件下，假设模型对于完整的样本

是正确的，那么将通过观测数据的边际分布以及选取合适的插补值对缺失值进行填充.

5.2 各变量的统计指标分析

对处理后的所有数值型变量进行相关性分析.根据均值、方差、最值和四分位数进行统计性描述如下表所示：

表 1 各指标的统计性描述

	Rating	Num_eviews	Body	Acidity	Price
count	7500.00	7500.00	6331.00	6331.00	7000.00
mean	4.25	451.11	4.16	2.95	inf
std	0.12	723.00	0.58	0.25	inf
min	4.20	25.00	2.00	1.00	4.99
25%	4.20	389.00	4.00	3.00	19.22
50%	4.20	404.00	4.00	3.00	28.53
75%	4.20	415.00	5.00	3.00	55.00
max	4.90	32624.00	5.00	3.00	3120.00

从均值和四分位数角度，能了解数据数值上的集中平均情况；从方差和最值角度，能体现出各数值型变量在总体上的波动情况.

对所有处理后的类别型变量进行统计指标分析，绘制词云图如下所示：

图 6 类别型变量统计指标分析

在图 4 中，四个类别型指标下的各类别字体大小表示该类别在该指标下出现

的频率.不难发现, wine（葡萄酒的名称）中 Unico 和 Reserva 出现的频率最高；winery（酒厂的名字）中 Sicilia 和 Vega 出现的频率最高;type（葡萄酒品种）中 Red 和 Ribera 出现的频率最高；region（葡萄酒的产地）中 Ribera 和 Duero 出现的频率最高.

除此之外, 进一步分析 year 和 rating 类别型指标, 绘制直方图反映类别中各个指标出现的频率情况, 如下图所示：

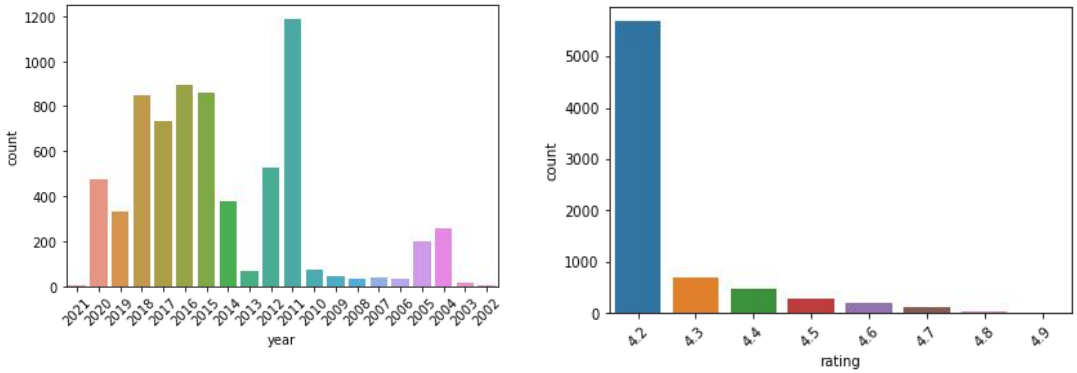

图 7 year 和 rating 指标下各类别出现的频率分布

在 year 指标中, 2011 年份出现的频率最大, 2002 年份出现的频率最小；在 rating 指标中, 用户对葡萄酒的平均评分集中在 4.2 分.

5.3 数值型变量的相关性分析

对数值型变量之间的相关性分析, 如下图所示：

图 8 各变量间的相关性分析

由上图得出大多数数值变量在价格栏上没有太多的相关性, 除了评级有一个弱到中等的正相关性.价格和评级呈正相关, 这意味着当评级很高时, 价格有可能也很高, 这是有意义的.

6 问题 2 模型的建立与求解

6.1 其他指标对三种评分的影响

针对其他指标对三种评分的影响，采用绘制气泡图的方式展示各个变量间的相关性.对 year、price 和 num_review 作气泡图的相关性分析，如下图所示：

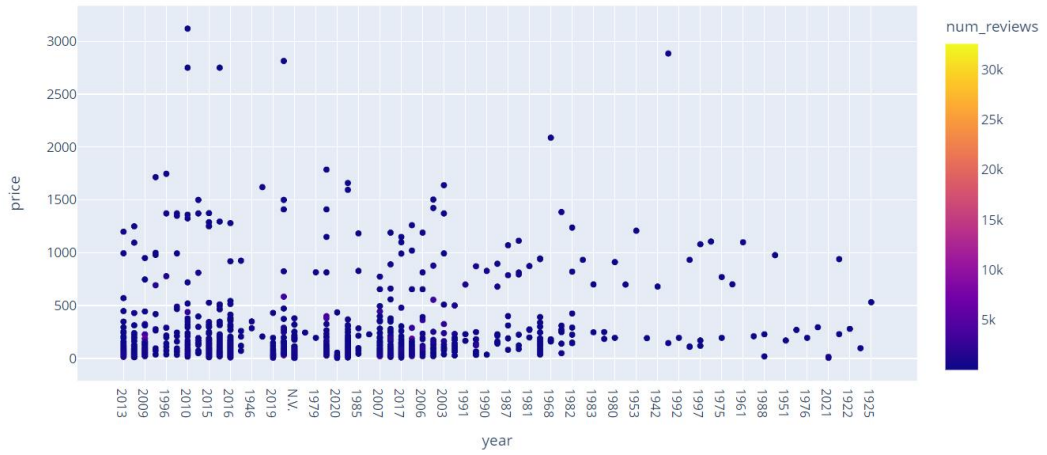

图 9 year、price 和 num_review 气泡图

上图是通过三个变量绘制气泡图，使用 num_review 的值赋予气泡不同的颜色. 由此得出，评论葡萄酒的用户数量与年份呈正相关而葡萄酒的用户数量与价格的相关性并不显著.

绘制箱图，进一步分析用户评分和各个变量之间的关系，具体如下图所示：

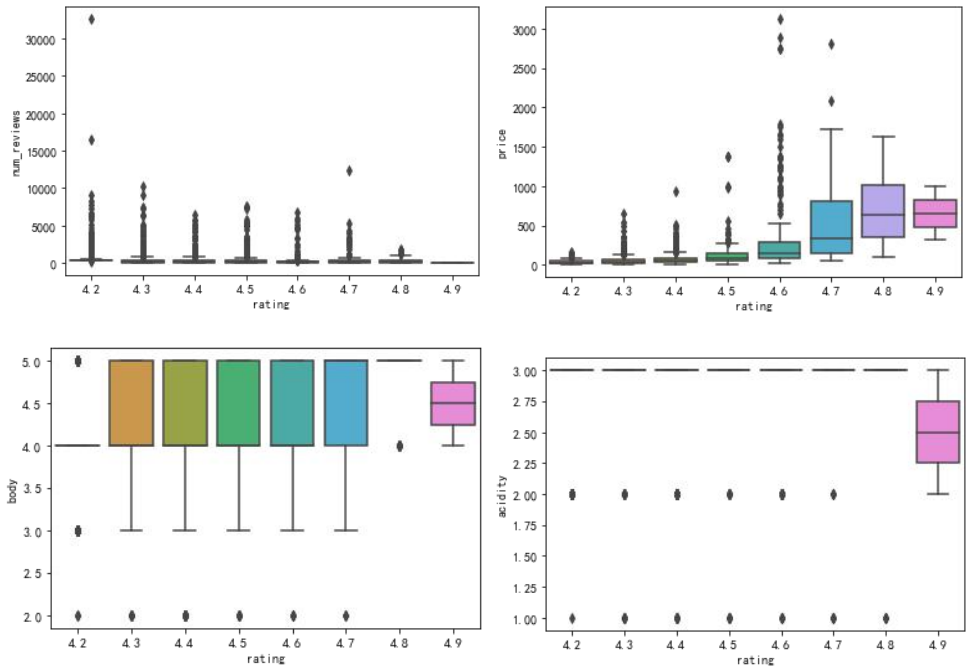

图 10 num_review、price、body、acidity 与 rating 的关系

由上图可以看出四组数据的分布特征，用户数量与用户评分基本无关；用户

评分越高，价格的离群点越少；酒体评分与价格评分的显著性不高；用户评分为4.9时，酸度评分没有离群点.

6.2 类别变量相关性分析

对于本题中出现的数据变量，考虑到文本类别型变量较多，基于此想要考虑变量间的相关性比较困难，因此考虑基于肯德尔相关系数来衡量，做出如下图所示的热力图，

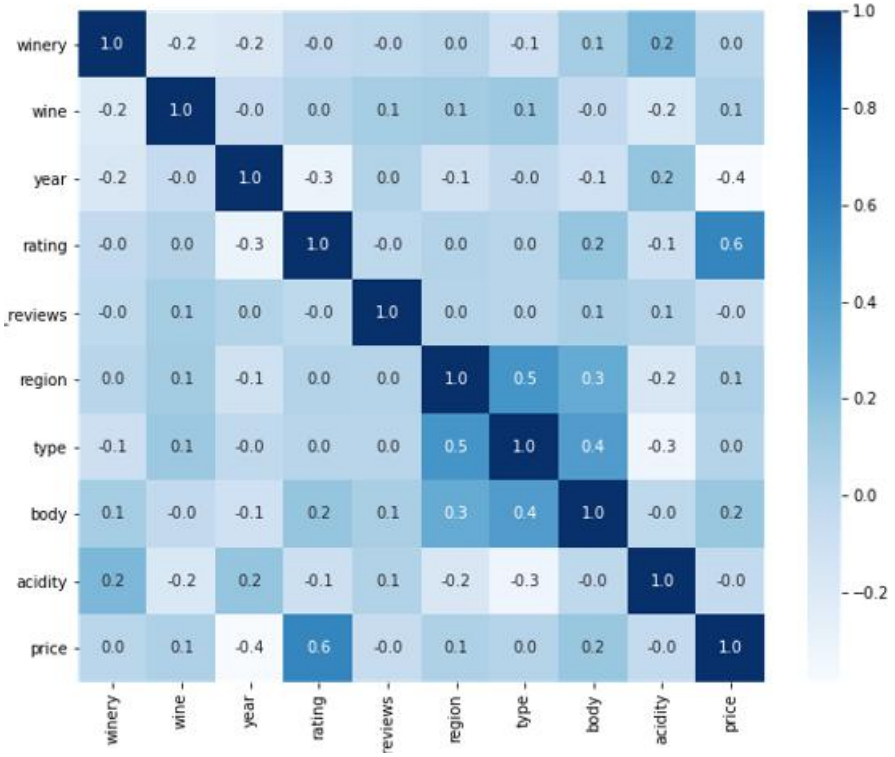

图 11 类别变量间的肯德尔相关系数热力图

从图中可以看出，对于rating和价格直接线性相关性较强，而winery与rating，reviews,region等变量数据线性相关性较低，考虑到后面的预测问题，可适当不考虑一些变化.

6.3 评分和价格之间的关系分析

针对评分和价格之间的关系，采用绘制气泡图的方式分析评分与价格之间的相关性.具体呈现如下图：

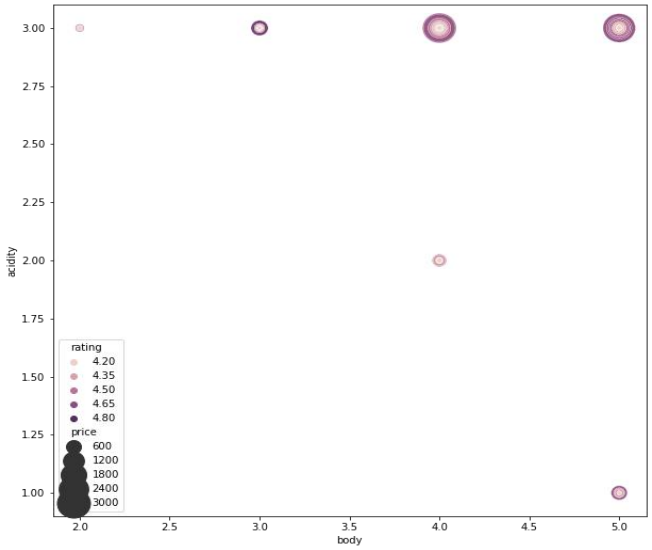

图 12 三种评分与价格的气泡图

气泡图是根据不同配色以及气泡的大小来反映各变量之间的相关性.上图中，使用 price 的值来反映气泡的大小；使用 rating 的值赋予气泡不同的颜色. 初步得出结论，酸度评分与酒体评分越高，价格越高，用户平均评分越高.

最后，利用 SPSS 软件对价格与评价之间的相关性进行进一步分析建立回归模型，观察各自变量的权重系数，结果如下表所示：

表 2 价格与评价之间的相关性分析

变量	B	标准错误	Beta	t	显著性	容差	VIF
常数项	-3117.67	69.077	0	-45.133	0.000	0	0
rating	721.08	15.01	0.542	48.043	0.000	0.996	1.036
acidity	12.34	7.76	0.018	1.590	0.112	0.992	1.008
body	18.27	3.24	0.063	5.642	0.000	0.973	1.028

其中自变量为 rating、acidity 和 body，因变量为 price.由上表可知，各自变量的系数分别为 721.08、12.34 和 18.27，所以 rating 对价格的影响更显著.

7. 问题 3 模型的建立与求解

7.1 对 7500 种葡萄酒的分类依据和方法

集群分析是一种将样本观察值进行分析，具有某些共性特征者予以整合在一起，再将之分配到特定的群体，最后形成许多不同集群的一种分析方法.针对问题三，利用集群分析中的 K-means 聚类分析将 7500 种葡萄酒进行分类[3].

在 K-means 算法前，需要给出划分组的数量（k 值），来达到更好的聚类效果.令 k 取值 2~11，做 k-means 聚类，看不同 k 值对应的簇内误差平方和，绘制碎石图抽取重要的公共因子来确定 k 值，从而得到 k 值最优值.

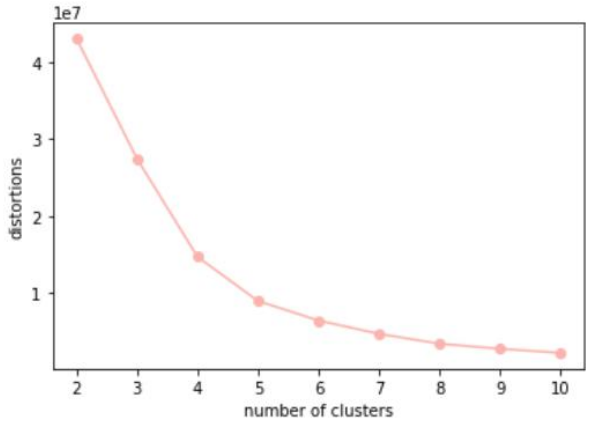

图 13 基于确定划分组数量的碎石图

碎石图主要看处在较大斜率位置处的点. 根据上图，4 处的节点为斜率出现较大变化的转折点，说明在 4 处的节点能更好的涵盖整个数据集. 基于以上的分析，确定划分组的数量为 4 组，并分别为其编号为 0，1，2，3. 部分分类结果如下表：

表 3 聚类后 price 结果统计分析

	类 0	类 1	类 2	类 3
count	5331	82	233	6
mean	inf	inf	inf	2734
std	26.98438	inf	inf	inf
min	6.261719	719.5	170	2088
25%	19.98438	899.5	205	2750
50%	28.95313	1097	259	2782
75%	51.34375	1343.75	350	2866.5
max	170.5	1786	701	3120

其中 Cluster 表示分组后的编号类别，鉴于篇幅问题详细见附件.

7.2 对 k 值选取的评估

用轮廓系数来评估分类结果的准确度或者称合适度. 通过求解轮廓系数值，绘制轮廓系数图来对 k 值等于 4 时的划分组数量进行评估. 在 k 取值为 2~11 时的轮廓系数值如下表所示：

表 4 轮廓系数值

类别个数	平均得分
2	0.9498065948973775
3	0.9459491024341974
4	0.8594991177926941
5	0.8456194864947186
6	0.6462914617532733
7	0.6401471801955745
8	0.6435398189031554
9	0.6493319636533448
10	0.6493474631703178

根据上表得出 k 值等于 4 时的聚类效果最好，对应的 k=4 时的轮廓系数图

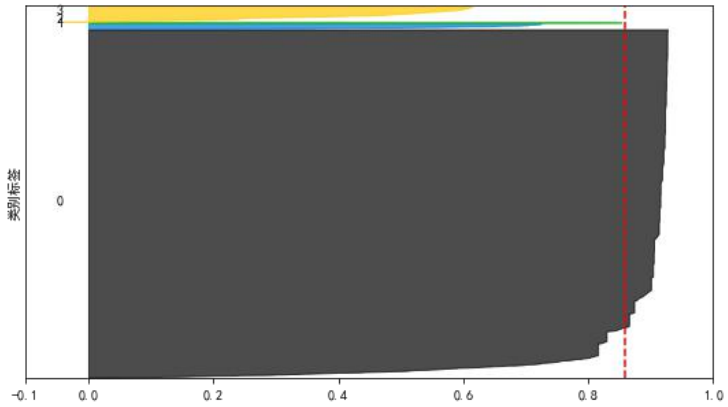

图 14 聚类数目取为 4 时轮廓系数图

其中的横坐标是一个衡量一个结点与它属聚类相较于其它聚类的相似程度. 取值范围是-0.1 到 1，它的值越大表明这个结点更匹配其属聚类而不与相邻的聚类匹配. 如果大多数结点都有很高的取值，那么聚类适当；若许多点都有低或者负的值，说明分类过多或者过少.

7.3 聚类效果的评价

选择若干个主要变量，绘制散点图矩阵.

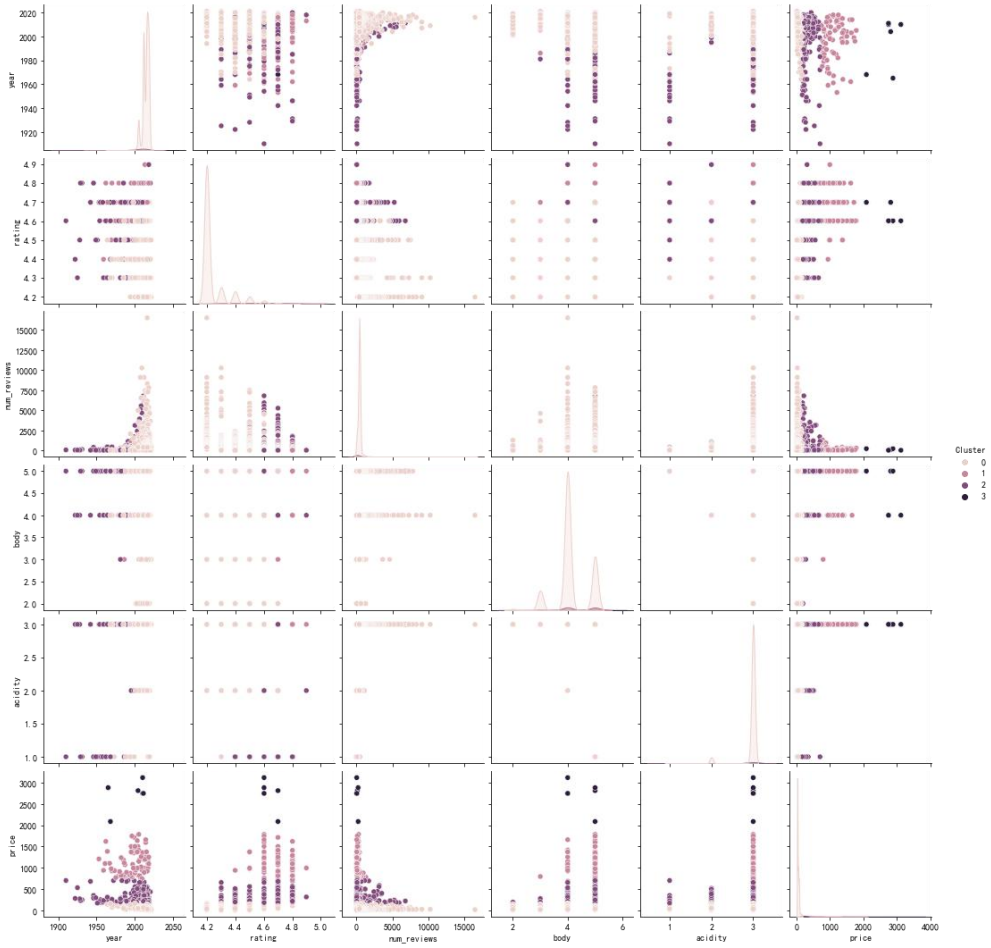

图 15 各变量间的散点图矩阵

由图可知，变量之间不存在当变量之间相关性较弱时，可认为主要变量之间具有一定的独立性.

根据 Cluster 类别制作散点图来体现聚类后的分布情况.为了有更好的可视化效果，利用 PCA（主成分分析）降维可视化来绘制散点图.如下图所示：

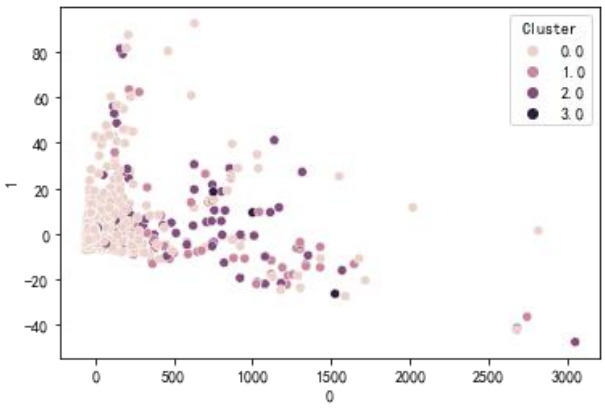

图 16 以 Cluster 为类别制作散点图

8. 问题 4 模型的建立与求解

8.1 数据集的划分

对于价格预测问题,是标准的回归问题, 针对此, 考虑主流的机器学习模型, 因此需要将数据集进行划分来处理, 即将 7000 个数据集划分为训练集和测试集, 采用训练集进行模型的构建, 测试集验证模型的准确性.

8.2 模型性能度量

定量预测模型本质上是一种回归模型, 我们采用平均绝对误差、均方绝对误差、R^2 值三种评价指标对以上四种模型进行评价.

平均绝对误差(MAE)是绝对误差的平均值, 能更好地反映预测值误差的实际情况, 同时对异常点的检测非常有效.

均方差误差(MSE)是指参数估计值与真实值之差平方的期望值, 反应自变量与因变量之间的相关程度. MSE 可以评价数据的变化程度, MSE 的值越小, 说明预测型描述实验数据具有更好的精确度.

R^2 的分母为原始数据的离散程度, 分子为预测数据和原始数据的误差, 两者相除可以消除原始数据离散程度的影响, R^2 越接近 1, 表明预测模型对数据拟合的越好.

三种评价指标的表达式为：

$$MSE=\sqrt{\frac{1}{n}\sum_{i-1}^{n}(Y\_actual-Y\_predict)^2}, \tag{1}$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}\left|Y\_actual-Y\_predict\right|, \tag{2}$$

$$R^2=1-\frac{\sum(Y\_actual-Y\_predict)^2}{\sum(Y\_actual-Y\_mean)^2}. \tag{3}$$

8.3 基于 Xgboost 的价格预测模型

XGBoost(eXtreme Gradient Boosting)极致梯度提升, 是基于 GBDT 的一种算法.XGBoost 的目标函数是由目标函数以及正则化两部分组成.

$$L^{(t)}=\sum_{j=1}^{T}[G_jw_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T \tag{4}$$

其中,$G_j=\sum_{i\in I_j} g_j$ 为叶子节点 j 所包含样本的一节偏导数累加之和；

$H_j=\sum_{i\in I_j} h_i$ 为叶子节点 j 所包含样本的二阶偏导数累加之和.其中求解的算法如下：

XGBoost 算法

Step1: 构造每个叶子节点 j 的目标函数：

$$f(w_j)=G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2 \tag{5}$$

Step2: 生成决策树

构建形如一元二次方程, 求最优解 $(H_j+\lambda)>0$, 则 $f(w_j)$ 在 $w_j=-\frac{G_j}{H_j+\lambda}$ 处取得最小值, 最小值为 $-\frac{1}{2}\frac{G_j^2}{H_j+\lambda}$；

Step3: 代入求得最优的 Obj 目标值：

$$Obj=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T \tag{6}$$

XGBoost 算法的原理如下图所示：

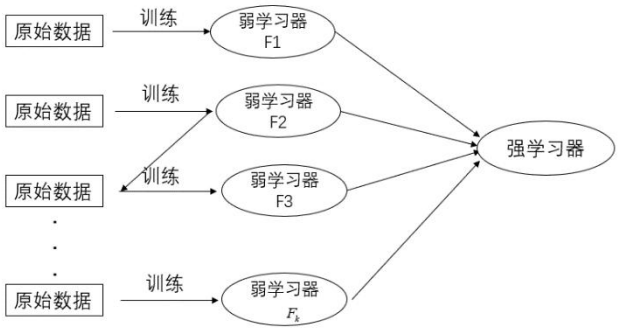

图 17 XGBoost 原理流程图

8.4 预测结果分析

求解过程使用 autogluon 程序包求解, AutoGluon 兼具易用和扩展性, 并专注于涵盖图像、文本或表格数据的深度学习和实际应用[7], 以 *MSE* 为评价标准, 得到如下的实验结果：

表 4 实验结果

model	score_test	score_val
XGBoost	-17.7934	-10.1904
NeuralNetFastAI	-17.8424	-12.613
NeuralNetTorch	-17.9003	-11.513
LightGBMLarge	-18.3767	-11.8541
LightGBMXT	-18.746	-11.8084
LightGBM	-18.9324	-12.0816
RandomForestMSE	-20.0122	-12.9472
CatBoost	-20.1193	-13.5414
ExtraTreesMSE	-20.3221	-13.3837
KNeighborsUnif	-54.9676	-48.1322
KNeighborsDist	-55.3286	-47.1297

从结果来看, Xgboost 模型的效果更好, 在测试集上平均绝对误差为-10.1904, 效果最好, 因此我们采用 Xgboost 模型作为预测模型.

接着抽取部分样本查看了预测值和真实值的差别, 如下图所示：

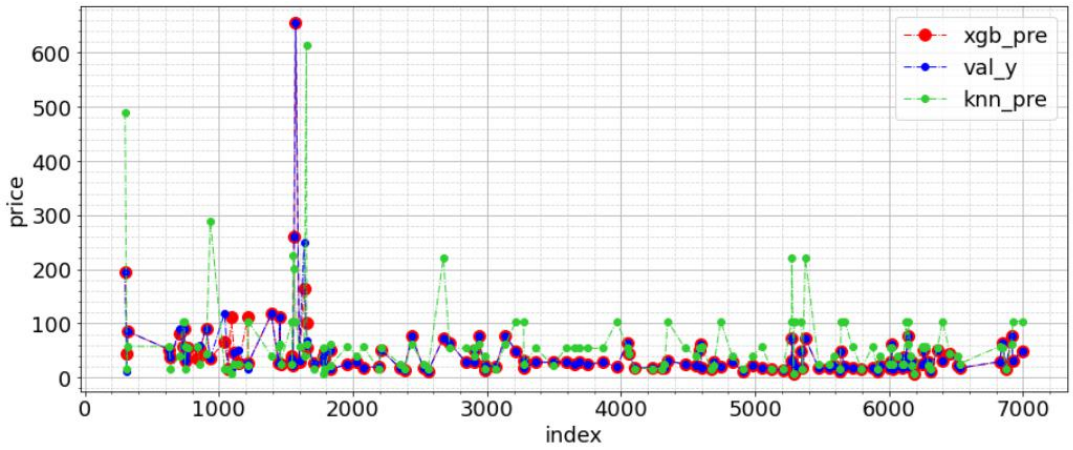

图 18 预测值与真实值的差值

可以看出 Xgboost 模型预测的葡萄酒价格相当贴近于真实值, 效果较好.

9. 总结

9.1 模型评价

9.1.1 模型的优点

(1) 决策树模型挖掘出来的分类规则准确性高, 具有很好的解释性, 便于理解.

(2) 建立的回归算法预测模型具有自适应能力、容错性强, 能够很好的处理非线性、非局域性的大型复杂系统.

(3) 通过分数的平均值的结果, 综合考量了各学习模型的预测准确度和稳定性. 选出来的模型更具有代表性, 泛化效果更好.

9.1.2 模型的缺点

Xgboos 算法可能有很多相似的决策树, 掩盖了真实的结果, 而且调参效率不高, 无法控制模型内部的运行, 只能在不同的参数和随机种子之间进行尝试.

9.2 模型的改进

(1)使用机器学习模型提取主要特征虽能加快训练速度, 但也会使样本信息提取不完全导致部分信息丢失, 后续可以尝试构造 Stacking 特征.

(2)对于问题 4 的处理,可以考虑构造多种特征来继续继续, 且为了进一步提高精度, 可以考虑使用多种模型来进行融合.

参考文献

[1] 刘拥民, 罗皓懿, 谢铁强. 基于 XGBoost-ARIMA 方法的 PM_(2.5)质量浓度预测模型的研究及应用[J/OL].安全与环境学报:1-13[2022-05-13].DOI:10.13637/j.issn.1009-6094.2022.1849.

[2] 孙林, 刘梦含, 徐久成. 基于优化初始聚类中心和轮廓系数的 K-means 聚类算法[J].模糊系统与数学,2022,36(01):47-65.

[3] 夏雪, 盖靖元. 基于 K-Means 聚类算法的城市轨道交通站点分类及客流特征分析[J].现代城市轨道交通,2021(04):112-118.

[4] 罗春芳, 张国华, 刘德华, 朱定欢.基于 Kmeans 聚类的 XGBoost 集成算法研究[J].计算机时代,2020(10):12-14.DOI:10.16644/j.cnki.cn33-1094/tp.2020.10.004.

[5] 王玉霞, 李果, 王芳, 陈世雄. 基于多元统计分析的葡萄酒及其理化指标评价研究[J].物流工程与管理,2014,36(01):160-164.

[6] 选酒大师. 哪一年的红酒好？葡萄酒年份解读[EB/OL]. 2022[02-22]. https://zhuanlan.zhihu.com/p/127194831.

[7] Fedesoriano. Spanish Wine Quality Dataset[EB/OL]. 2022[04-25]. https://www.kaggle.com/datasets/fedesoriano/spanish-wine-quality-dataset.

附 录

附录 1：附件及程序环境

表 1 支撑材料清单表

Jupyter 文件	spanish-wine.ipynb
提交结果文件表格	submit_results.csv
分析过程文件	type 特征数量分类描述.csv 数值类型变量描述表.csv 数值变量相关系数.csv 评分与价格的关系.spv Autogluon 保存的模型

表 2 运行环境

系统	windows10, 64 位操作系统
CPU	Intel i5-9300H 2.4GHz
GPU	NVIDIA GeForce GTX 1050 3G

表 3 python 相关包版本

包名	版本
LightGBM	3.2.1
Scikit-learn	0.3.2
seaborn	0.11.0
Autogluon	0.4.0

附录 2：问题 1 源程序代码

问题 1　python 程序	问题一数据处理相关程序

```
## 前期准备

### 导包

# In[1]:


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#from wordcloud import WordCloud
plt.rc('figure', figsize=(15, 10))
sns.set_context(font_scale=2)
import seaborn as sns
import plotly_express as px
get_ipython().run_line_magic('matplotlib', 'inline')
import datetime
import gc   #
import re
import warnings
warnings.filterwarnings('ignore')

### 导入数据

# In[2]:

df=pd.read_csv(r'D:\竞赛\2022 校赛\wines_SPA.csv',index_col=['index'])
df2=pd.read_csv(r'D:\竞赛\2022 校赛\wines_SPA.csv')
df.head()

### dataframe 压缩
# In[3]:

def reduce_mem(df):
    start_mem = df.memory_usage().sum() / 1024 ** 2
    for col in df.columns:
        col_type = df[col].dtypes
        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
```

```
                    if str(col_type)[:3] == 'int':
                        if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                            df[col] = df[col].astype(np.int8)
                        elif c_min > np.iinfo(np.int16).min and c_max <
np.iinfo(np.int16).max:
                            df[col] = df[col].astype(np.int16)
                        elif c_min > np.iinfo(np.int32).min and c_max <
np.iinfo(np.int32).max:
                            df[col] = df[col].astype(np.int32)
                        elif c_min > np.iinfo(np.int64).min and c_max <
np.iinfo(np.int64).max:
                            df[col] = df[col].astype(np.int64)
                    else:
                        if c_min > np.finfo(np.float16).min and c_max <
np.finfo(np.float16).max:
                            df[col] = df[col].astype(np.float16)
                        elif c_min > np.finfo(np.float32).min and c_max <
np.finfo(np.float32).max:
                            df[col] = df[col].astype(np.float32)
                        else:
                            df[col] = df[col].astype(np.float64)
            end_mem = df.memory_usage().sum() / 1024 ** 2
            print('{:.2f} Mb, {:.2f} Mb ({:.2f} %)'.format(start_mem, end_mem, 100 *
(start_mem - end_mem) / start_mem))
            gc.collect()
            return df
        # In[4]:

        #dataframe 压缩
        df=reduce_mem(df)
        # ##  第一问
        # ###  数值变量描述统计分析
        # In[5]:

        df.describe().to_csv(r'D:\竞赛\2022 校赛\work\数值类型变量描述表.csv')
        df.describe()

        # In[6]:

        df.info()
        # In[7]:
        ax = sns.heatmap(df.isna().sum().to_frame(), annot=True, fmt='d',cmap='coolwarm')
        plt.savefig(r"D:\竞赛\2022 校赛\work\缺失值分布.png")
        ax.set_xlabel("Missing Values")
```

```
# In[8]:
df.isnull().sum()
# ### type 类型数据比例

# In[9]:
#查看类型数量
plt.figure(figsize=(12, 12))
class_gp = df2.groupby('type')['index'].count()
class_gp=pd.concat([class_gp,class_gp/len(df2)],axis=1)
class_gp.columns={'数量','比例'}
from palettable.colorbrewer.qualitative import Pastel1_7
plt.pie(class_gp['数量'],labels=class_gp.index,
colors=Pastel1_7.hex_colors,wedgeprops=dict(width=0.3, edgecolor='w'))
#  设置等比例轴, x 和 y 轴等比例
plt.axis('equal')
plt.show();
class_gp.to_csv(r"D:/竞赛/2022 校赛/work/type 特征数量分类描述.csv")
plt.savefig(r"D:\竞赛\2022 校赛\work\type 特征数量分类描述图.png")
# ###    year 变量分布
#

# In[10]:
year = df[df['year'] != 'N.V.']
sns.set_palette(Pastel1_7.hex_colors)
sns.countplot(year.sort_values(by=['year'], ascending=False)['year'][:7000])
plt.xticks(rotation=45)
plt.show()
# ### rating 变量分布
# In[11]:
sns.countplot(df.sort_values(by=['rating'], ascending=False)['rating'])
plt.xticks(rotation=45)
plt.show()
# ###  年份和价格关系
# In[12]:
import plotly.express as px
fig = px.scatter(df, x="year", y="price", color="num_reviews", title='Prices and
Reviews')
fig.show()
# ###  纯文本变量词云图
# In[13]:
cols = ['wine', 'winery', 'type',    'region']
wc = WordCloud(height=1200, width =2000, random_state = 101,
background_color='white')
fig, axes = plt.subplots(2, 2, figsize=(20, 12))
```

```
axes = [ax for axes_row in axes for ax in axes_row]
for i, c in enumerate(cols):
        op = wc.generate(str(df[c]))
        x = axes[i].imshow(op)
        x = axes[i].set_title(c.upper(), fontsize=24)
        x = axes[i].axis('off')
# ###  相关性分析
# In[14]:
df.corr().to_csv(r'D:\竞赛\2022 校赛\work\数值变量相关系数.csv')
# In[15]:
cmap = sns.cubehelix_palette(start = 1.5, rot = 3, gamma=0.8, as_cmap = True)
sns.heatmap(df.corr(),annot=True, cmap=cmap, fmt='.1f')
```

附录 3：问题 2 源程序代码

python 程序	问题二处理相关程序

```
## 第二问
### 类别变量间肯德尔相关系数
# In[16]:
df.loc[:,{'winery','wine','year','region','type'}].corr('kendall')
# In[17]:
#将 year 中的 N.V 用 NaN 替换
df=df.replace('N.V.', 'NaN')
# In[18]:
#将 year 转为 datetime 格式
df['year']=pd.to_datetime(df['year'])
# In[19]:
#提取 year 中的年
df['year']=df['year'].dt.year
### 用户评分和各个变量间关系图
# In[20]:
sns.boxplot(x=df["rating"],y=df['num_reviews'])
# In[21]:
sns.boxplot(x=df["rating"],y=df['price'])
# In[22]:
sns.boxplot(x=df["rating"],y=df['body'])
# In[23]:
sns.boxplot(x=df["rating"],y=df['acidity'])
# In[24]:
plt.figure(figsize=(10, 10))
sns.scatterplot(data=df, x='body', y='acidity',hue='rating',size='price',sizes=(10,1000))
### spss 做的回归
# ![image.png](attachment:image.png)
```

附录 4：问题 3 源程序代码

python 程序	问题三处理相关程序

```
# In[26]:
df1=df.dropna()
# In[27]:
df1
# In[28]:
df2=df1.drop(['num_reviews','winery','wine','region','type'],axis=1)
# In[29]:
df2
# In[30]:
from sklearn.cluster import KMeans
kmeans_model = KMeans(n_clusters=4, random_state=10).fit(df2)
# In[31]:
labels = kmeans_model.labels_
labels
# In[32]:
color_codes = {0:'red', 1:'blue', 2:'yellow',3:'black'}
colors = [color_codes[x] for x in labels]
# In[33]:
pd.plotting.scatter_matrix(df1[df1.columns[0:]], figsize=(15,10), color=colors,
alpha=0.8, diagonal='kde')
plt.show()
# In[34]:
df1['Cluster']=labels
df1.head()
# In[35]:
df1.groupby('Cluster')['Cluster'].count().plot.pie(autopct="%1.1f%%")
# In[36]:
df2['Cluster']=labels
# In[37]:
df2.groupby('Cluster').describe().T
```

附录 5：问题 4 源程序代码

python 程序	问题四处理相关程序

```
# ###  数据集划分

# In[41]:
train_data=data.sample(frac=0.7)#按 0.7 比例随机采样
test_data=data[~data.index.isin(train_data.index)]
train_data
# In[42]:
test_data
# ### autogluon 选择合适模型

# In[43]:


from autogluon.tabular import TabularDataset, TabularPredictor
train_data = TabularDataset(train_data)
test_data = TabularDataset(test_data)


# In[44]:


label="price"
metric='mean_absolute_error'
predictor = TabularPredictor(label, eval_metric=metric).fit(train_data)


# ###  查看模型结果

# In[45]:


predictor.leaderboard(test_data, silent=True)


# ###  绘图显示误差

# In[64]:


###拿出部分数据画图
train_data1=data.sample(frac=0.98)
```

```
    test_data1=data[~data.index.isin(train_data1.index)]
    ##真实数据
    y=test_data1.price
    y


    # In[65]:
    ###预测数据
    xgb_pre_y=predictor.predict(test_data1.loc[:,"winery":"acidity"], model='XGBoost')
    knn_pre_y=predictor.predict(test_data1.loc[:,"winery":"acidity"],
model='KNeighborsUnif')
    xgb_pre_y


    # In[67]:


    plt.rcParams['font.sans-serif']=['SimHei']
    plt.rcParams['axes.unicode_minus']=False
    plt.figure(figsize=(15,6))
    plt.rcParams['font.size']=18
    plt.rcParams['font.family']='Time New Roman'

    plt.grid(visible=True,which='major',linestyle='-')
    plt.grid(visible=True,which='minor',linestyle='--',alpha=0.5)
    plt.minorticks_on()

    plt.plot(xgb_pre_y,'o-.',color='red',label='xgb_pre',linewidth=1,markersize=10)
    plt.plot(y, 'o-.', color='blue', label='val_y', linewidth=1)
    plt.plot(knn_pre_y, 'o-.', color='limegreen', label='knn_pre', linewidth=1)

    plt.xlabel("index")
    plt.ylabel("price")
    plt.legend()
    plt.show()


    # ### 导出结果

    # In[68]:
    pd.DataFrame(predictor.predict(df.loc[7000:,"winery":"acidity"],
model='XGBoost')).to_csv(r'D:/竞赛/2022 校赛/work/submit_results.csv')
```