果汁饮料质量的研究

摘要

本文针对影响果汁饮料质量的多种因素，通过相关性检验，属性权重分析，logistic 回归分析，建立了合适的果汁饮料的质量预测模型，以达到调整原料比率，降低果汁生产成本，实现高质量生产的目的，并提出能更加优化模型。并

针对问题一，根据已知的数据样本，利用 Excel 软件，建立各变量与果汁饮料的数据拟合关系以及独立性检验的方式，分析图像趋势和相关系数等有效性质，判断和筛选出符合要求的变量，删除影响较小的变量 x2，x4，实现对模型的简化。

针对问题二，按照 x6 变量进行聚类分析，利用 JupyterNotebook 软件用 Python 语言编程，对 x6 不同水平下的变量进行权重分析，通过数据的可视化，比较不同水平下各变量对于果汁饮料质量的影响变化情况，从而分析出在不同水平下它们对于果汁饮料质量影响的规律没有发生很大变化，但是变量对于 y 的影响程度有明显改变，如 x1，x2 对于 y 的影响随着 x6 的增大而增大，对模型的进一步优化提供帮助。

针对问题三，综合问题一和问题二的所得出的结论，通过 python 实现采用逻辑回归的算法，加入不同的权重作为参数建立模型对附件 2 的 19 个果汁饮料样品，给出它们的质量等级预测，多次预测的平均准确率为 80.65%。

针对问题四，利用问题三中的模型预测的结果，通过 Excel 的筛选方法，对变量的成分和等级的改变，找到最接近的变量样本，根据分析，不能无限的缩小 x1 的分量，而修改 x2 变量会更加有利于提高果汁饮料的质量，并且在成本上达到相对较低，符合实际的果汁流水线生成方式。

关键词： Excel，python，逻辑回归，权重，相关性检验，预测分析

一、问题重述

1.1 问题的背景

随着生活水平的不断提高，人们开始追求更加健康、绿色的生活方式，人们消费观念的转变，致使食品质量成为人们最为关心的因素之一，也成为政府民生工程的一个主题。2016 年 12 月，国务院食品安全办会同发展改革委、财政部等部门研究起草了《“十三五”国家食品安全规划》，紧紧围绕统筹推进“五位一体”总体布局和协调推进“四个全面”战略布局，牢固树立和贯彻落实创新、协调、绿色、开放、共享的发展理念，坚持最严谨的标准、最严格的监管、最严厉的处罚、最严肃的问责，全面实施食品安全战略，着力推进监管体制机制改革创新和依法治理，着力解决人民群众反映强烈的突出问题，推动食品安全现代化治理体系建设，促进食品产业发展，推进健康中国建设。

1.2 问题的提出

针对上述的背景，为了对果汁饮料实行监管，现有关部门将其按照营养成分及口感等将质量划分为 2 个等级，一般和较好，0—表示较好，1—表示一般，为了考察影响果汁饮料质量有哪些因素，检验人员从市场的大量果汁饮料样品中测试其各种成分含量，如维生素 C、果肉、能量、糖分、矿物质、碳水化合物、各种食品添加剂等，经过一些数据处理后，筛选出 6 个主要指标。

1.3 问题的要求

问题一：利用题目给出的数据，分析果汁饮料的质量受哪几个指标的影响，如果有指标与果汁饮料的质量之间没有相关性，则将其删除。

问题二：分析在指标 x_6 的不同水平下，果汁饮料的质量与其它各指标之间的影响关系，并分析在 x_6 的不同水平下它们有什么异同。

问题三：根据问题（1）、（2）分析的结果，给出对附件 2 的 19 个果汁饮料样品的质量等级预测。

问题四：在配制生产过程中，改变指标 x_1, x_2, x_4, x_5 的单位含量的成本以及每改变指标 x_3, x_6 一个等级的成本见附件 3。问：对附件 2 中的等级一般的产品，如何调整各指标的含量或等级，使得产品质量等级尽可能达到较好，而成本尽可能低。

二、问题分析

本题针对果汁饮料的六种成分进行分析，查找出在一个或几个不同指标下的果汁饮料和质量之间的相关性，并进行预测果汁质量和最优的成本预测问题。我们认为对于果汁质量的综合评价应该从各种成分含量对质量的影响出发，从客观的实测数据出发，对六千多个果汁质量数据进行综合评价。给定各类物质各一个权值反映其对水质影响大小，对不同 x6 的果汁等级进行划分方式考虑，利用逻辑回归的原理进行评测。考虑到各个区间下 x2、x4 的权值很小，并且对果汁饮料质量之间没有相关性，我们将其删除处理。对于预测问题，可以考虑将现有的值划分为 80%的训练集和 20%的测试集，将数据拆分为测试和训练集的目的是避免过度拟合，将数据进行清洗删除影响性不大的数据和异常数据，再将特征进行具有高斯分布和不同平均值和标准偏差的属性转换为平均值为 0 和标准偏差为 1 的标准高斯分布的标准化数据。再利用逻辑回归的原理，把不同 x6 对应的其他

五个值的权重加入进行模型的计算，从而预测附件二中的样本。

模型假设

（1）假设除文中所给的变量，其余变量对果汁饮料质量的影响可以忽略

（2）假设各个变量之间的互相影响可以忽略。

（3）假设题中所给的样本均为真实可靠的数据

三、名词解释和符号说明

4.1 名词解释

4.2.3

过拟合：给定一个假设空间 H，一个假设 h 属于 H，如果存在其他的假设 h’属于 H,使得在训练样例上 h 的错误率比 h’小，但在整个实例分布上 h’比 h 的错误率小，那么就说假设 h 过度拟合训练数据。

正态分布：随机变量X服从一个位置参数为 μ 、尺度参数为 σ 的概率分布

标准化处理：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和量纲单位，这样的情况会影响到数据分析的结果，为了消除指标之间的量纲影响，需要进行数据标准化处理，以解决数据指标之间的可比性。原始数据经过数据标准化处理后，各指标处于同一数量级，适合进行综合对比评价。

逻辑回归： logistic 回归是一种广义线性回归（generalized linear model），因此与多重线性回归分析有很多相同之处。它们的模型形式基本上相同，都具有 w‘x+b，其中 w 和 b 是待求参数，其区别在于他们的因变量不同，多重线性回归直接将 w‘x+b 作为因变量，即 y =w‘x+b，而 logistic 回归则通过函数 L 将 w‘x+b 对应一个隐状态 p，p =L(w‘x+b),然后根据 p 与 1-p 的大小决定因变量的值。如果 L 是 logistic 函数，就是 logistic 回归，如果 L 是多项式函数就是多项式回归。

交叉验证：交叉验证，顾名思义，就是重复的使用数据，把得到的样本数据进行切分，组合为不同的训练集和测试集，用训练集来训练模型，用测试集来评估模型预测的好坏。

4.2 符号说明

4.2.1

变量	解释说明
y	果汁饮料的质量
R 的平方	相关系数
K 的平方	卡方统计量

4.2.2

变量	解释说明
A	决策矩阵
Xij	矩阵中的第 i 行第 j 列的元素
Pij	第 j 项的指标下第 i 个记录的所占比重
Ej	信息熵
Wj	对应元素的属性权重

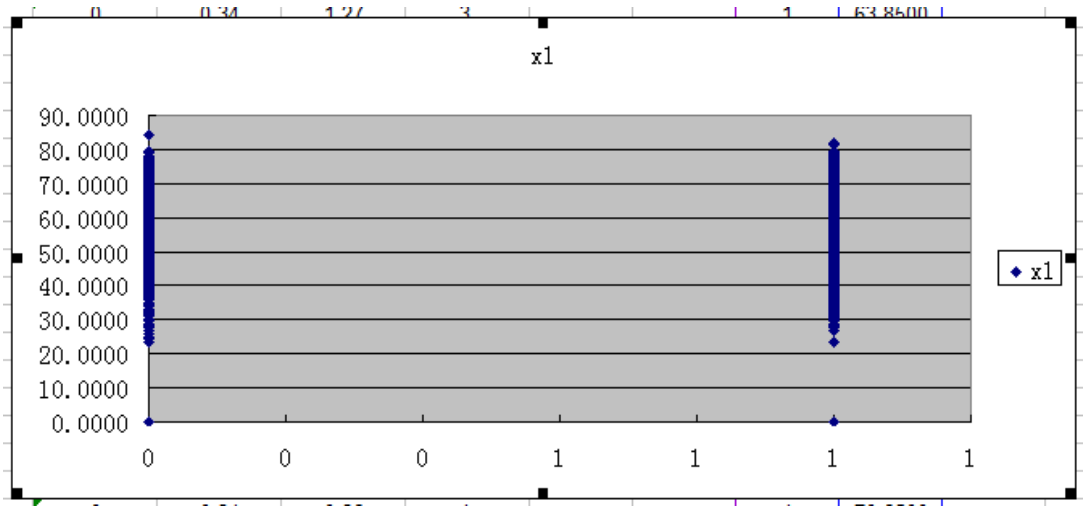
4.2.3

i	数字从 1-6
μ	x1-x6 中数据的平均数
δ	x1-x6 中数据的标准差
f(x)	各指标的激活函数
P	目标值为 0 或 1 的概率
θ	偏置，x1-x6 的权重比

四、模型建立与求解

5.1 模型建立

问题一：最初使用 EXCEL 数据散点图，无法得出规律，后来换为统计各变量的比例情况。

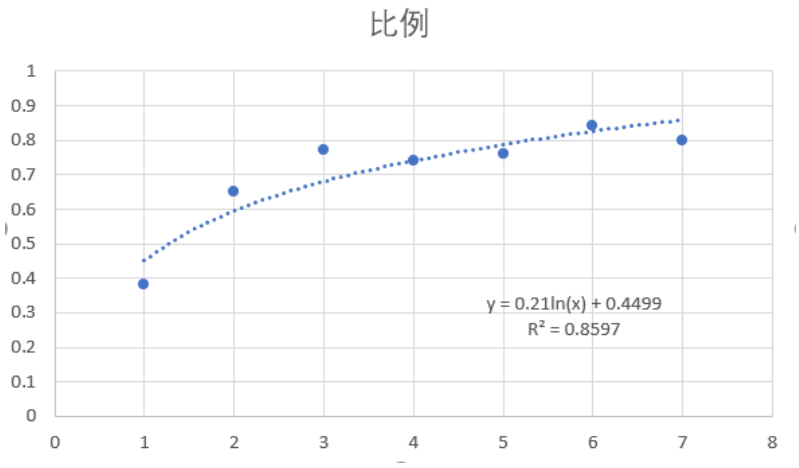


	A	B	C	D
1	x1	1的数量	总数	比例
2	20-30	13	34	0.38
3	30-40	262	401	0.65
4	40-50	653	841	0.77
5	50-60	1016	1356	0.74
6	60-70	1445	1886	0.76
7	70-80	1486	1765	0.84
8	80-85	8	10	0.8

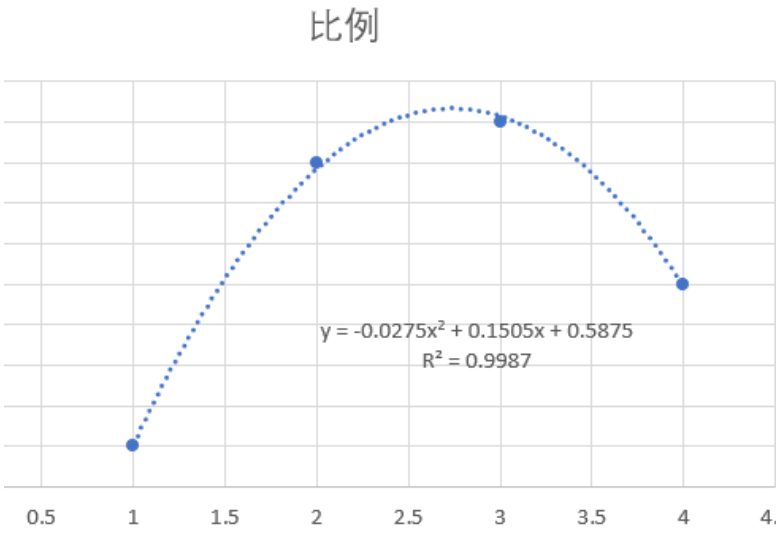
统计各项指标在产品质量为较好的情况下下所占的比例，可以看出比例基本不变，排除 x2、x4 与产品质量有相关性。

x2	1的数量	总数	比例
20-30	143	182	0.78
30-40	1351	1740	0.77
40-50	1525	1928	0.79
50-60	1241	1585	0.78
60-70	591	780	0.75
70-80	98	134	0.73
x4	1的数量	总数	比例
0.1-0.3	426	538	0.79
0.3-0.5	2397	3097	0.77
0.5-0.7	2124	2692	0.78

先画出 x1、x5 的散点图，再使用曲线拟合的方法计算 x1、 x5 指标与产品质量的相关系数，用 EXCEL 描绘出相关系数的图像，通过相关系数判断该指标和产品质量是否有相关性，因为 x1、x5 关系数接近 1，所以与质量等级有相关性。



x1 关系相关系数图像



x3 关系相关系数图像

使用卡方检测法列出 x3 的 2X2 列联表，可以发现 x3 的卡方统计量大于 8，所以 x3 与产品质量等级有相关性。

	x3=1	x3=0	
y=0	884	496	1340
y=1	3369	1580	4949
	4253	2076	6329
	卡方统计量=8.133		

X3 2x2 列联表

5.2

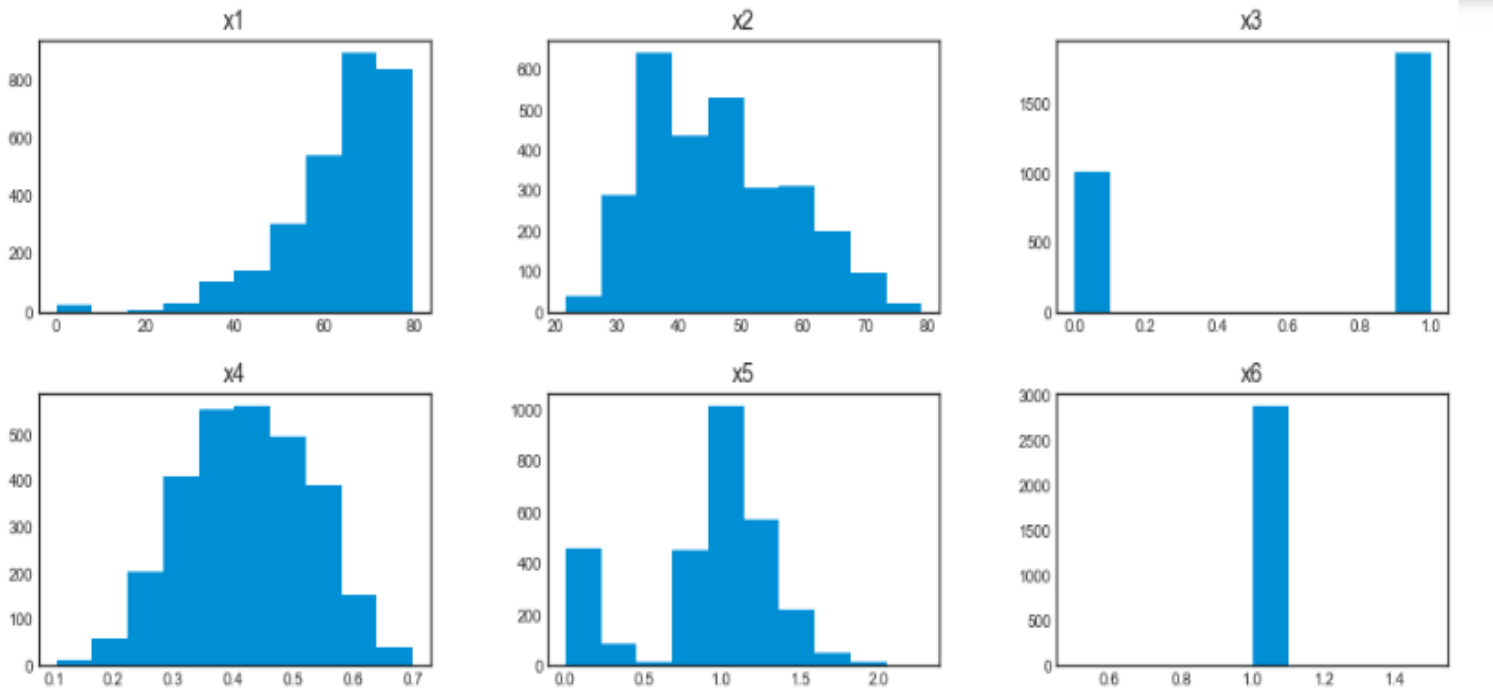
第二题的题目要求是分析指标 x6 在各个不同水平下，果汁饮料的质量与其它各指标之间的影响关系，并分析在不同水平下它们有什么异同。

首先利用 excel 按照 x6 的不同水平对其变量进行聚类分析，计算出 y=1 时的比例变化情况

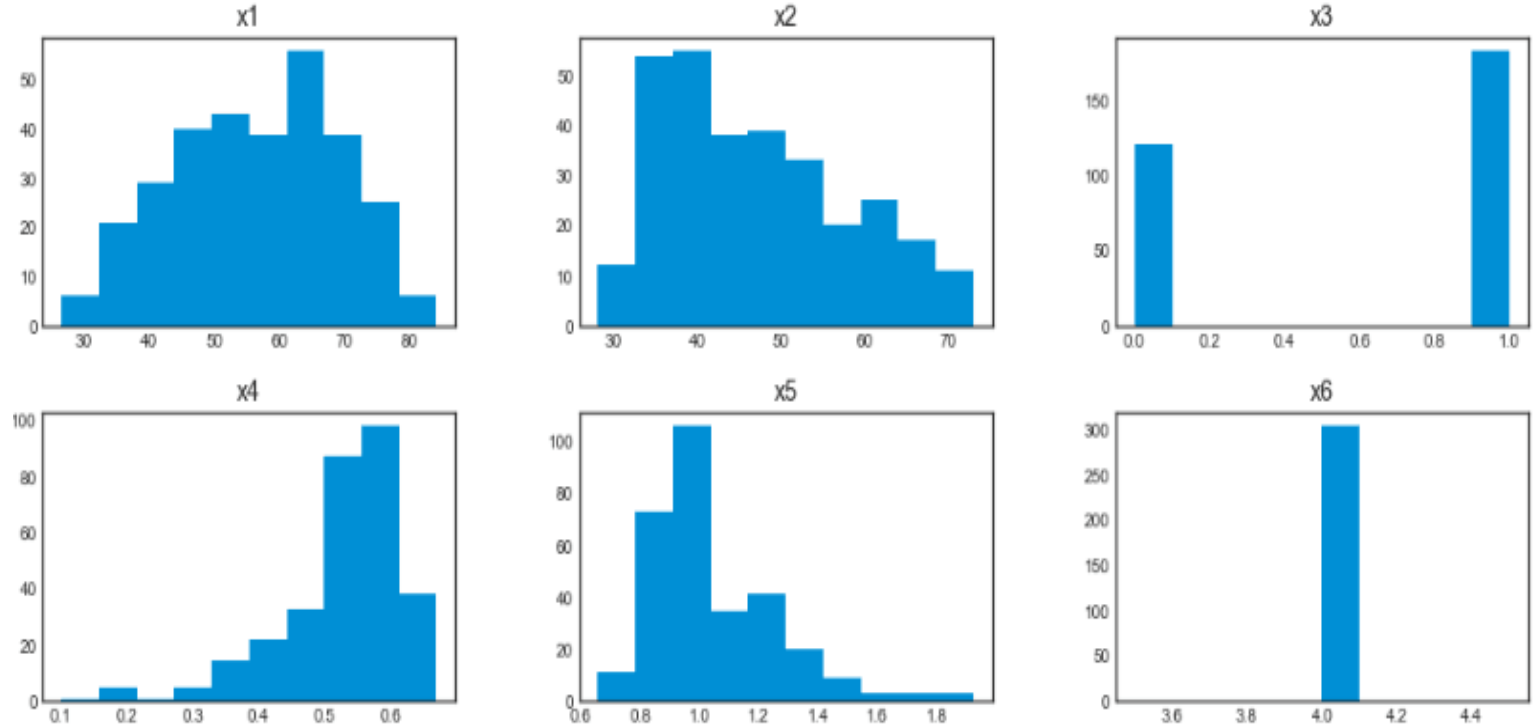
A	B	C	D	E	F	G	H	I	J	K	L	M
x6=1												
x1	y=1	总数	比例	x3	y=1	总数	比例	x5	y=1	总数	比例	
20-30	2	16	0.125	0	756	1003	0.75	0-0.5	388	540	0.71	
30-40	61	114	0.535	1	1865	2112	0.88	0.5-1.0	607	803	0.75	
40-50	141	206	0.684					1.0-1.5	1035	1401	0.73	
50-60	343	507	0.67					1.5-2.0	81	123	0.65	
60-70	602	866	0.69									
70-80	944	1136	0.83									
x6=2												
x1	y=1	总数	比例	x3	y=1	总数	比例	x5	y=1	总数	比例	
20-30	1	10	0.1	0	496	606	0.81	0.5-1.0	569	680	0.83	
30-40	81	103	0.78	1	1006	1214	0.82	1.0-1.5	881	1068	0.82	
40-50	253	318	0.79					1.5-2.0	54	74	0.72	
50-60	335	410	0.81									
60-70	462	559	0.82									
70-80	356	411	0.86									
x6=3												
x1	y=1	总数	比例	x3	y=1	总数	比例	x5	y=1	总数	比例	
20-30	4	7	0.57	1	816	989	0.82	0.5-1.0	446	536	0.83	
30-40	125	148	0.84	0	246	272	0.9	1.0-1.5	610	759	0.803	
40-50	212	259	0.81					1.5-2.0	32	40	0.8	
50-60	274	358	0.76									
60-70	313	377	0.83									
70-80	158	184	0.85									
80-90	2	2	1									
x6=4												
x1	y=1	总数	比例	x3	y=1	总数	比例	x5	y=1	总数	比例	
20-30	0	0	0	0	93	121	0.76	0.5-1.0	105	135	0.77	
30-40	32	36	0.88	1	152	182	0.83	1.0-1.5	131	158	0.82	
40-50	47	59	0.79					1.5-2.0	9	11	0.818	
50-60	63	81	0.77									
60-70	68	84	0.809									

然后利用 JupyterNotebook 软件，对不同情况下的数据分类，通过 Python 语言的可视化，可得出以下频数分布直方图：

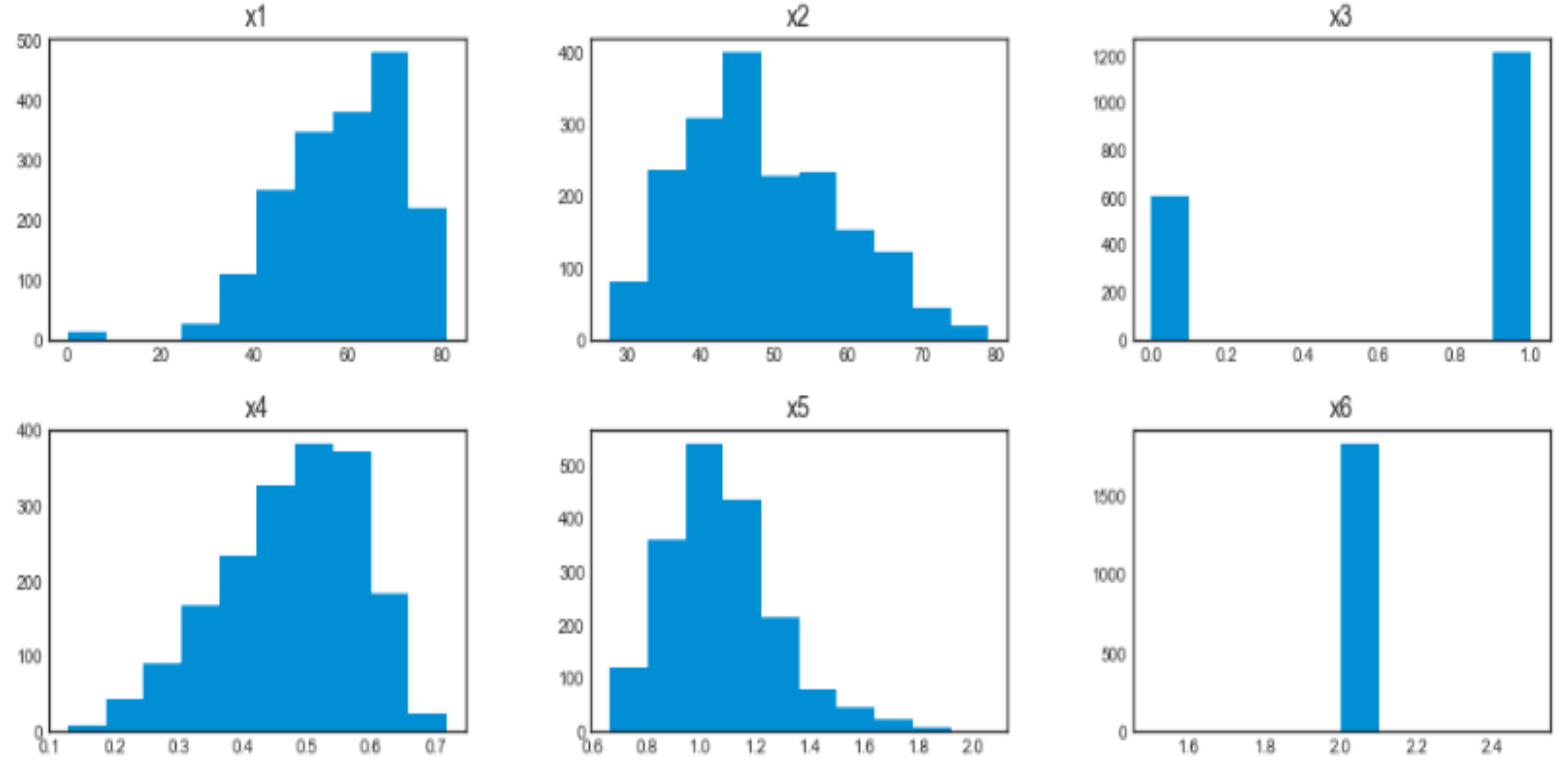
x6=1



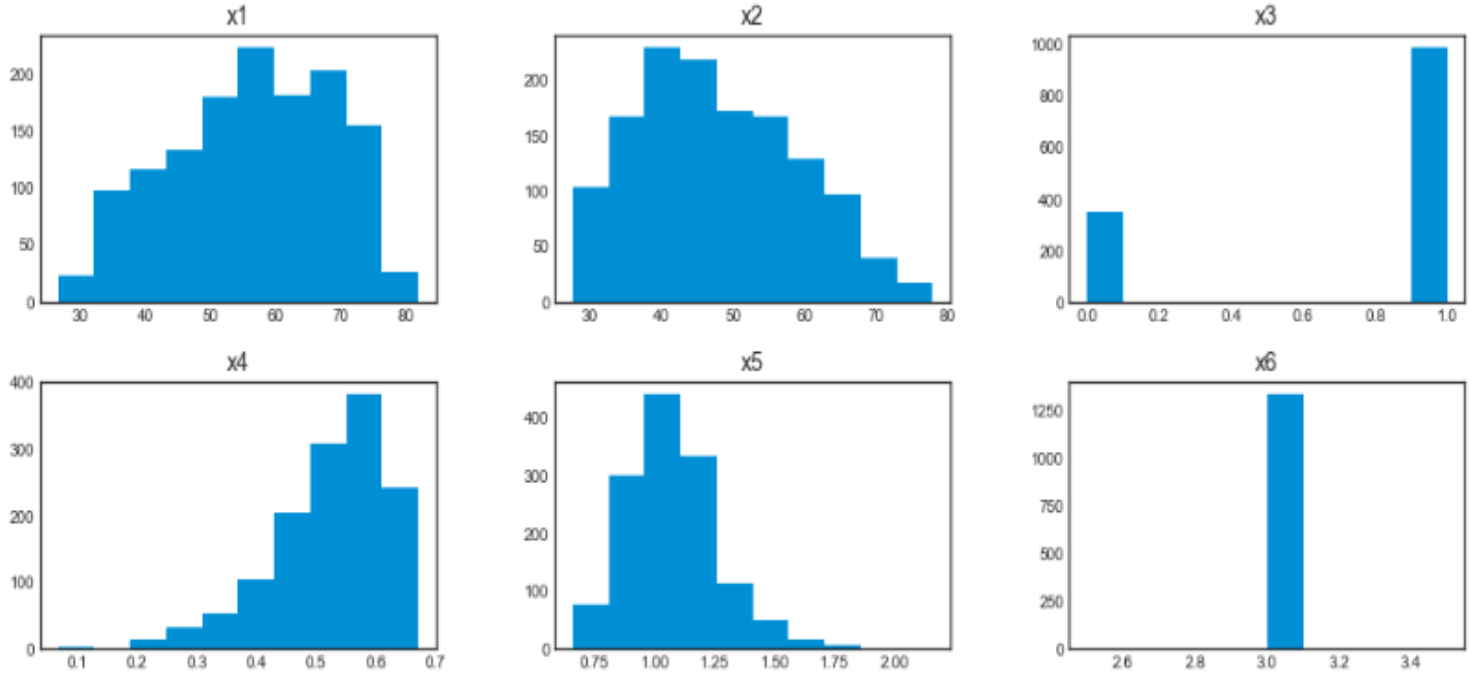
x6=2



x6=3



x6=4



通过以上图表和数据可知，上述变量在 x6 的不同水平下，变化趋势以及峰值的范围基本还是没有变化，对于 y 的影响规律依旧没有较大的改变，与第一问的相关性检验一致。

下面通过权重比较来寻找不同点，利用 JupyterNotebook 软件实现权重的算法，权重的比较更能看出各种饮料的影响变化，通过样本数据，以 y 为行，以变量属性为列，建立出决策矩阵 A。

1.假设数据有n行记录，m个变量，数据可以用一个n*m的矩阵A表示(n行m列，即n行记录数，m个特征列)

$$A = \begin{bmatrix} x_1 \ldots x_m \end{bmatrix}$$

2.数据的归一化处理

xij表示矩阵A的第i行j列元素。

$$x_{ij} = \frac{x_{ij} - min(x_j))}{max(x_j) - min(x_j)))};$$

计算第 j 项的指标下第 i 个记录的所占比重。

$$P_{ij} = \frac{x_{ij}}{\sum_1^n x_{ij}} (j = 1, 2, ..., m);$$

在信息论中熵是衡量不确定性的指标，一个信息量的分布越趋于一致，所提供的信息的不确定性越大，按照归一化处理中得到的比重（P1j，P2j，...Pnj）（j=1,2...n），按照 shannon 给出的数量指标——信息熵的定义，各个样本关于 Xj 的熵为

$$e_j = -k * \sum_1^n P_{ij} * log(P_{ij}), k = 1/ln(n);$$

计算第 j 项的指标差异系数
Fj=1-Ej，0≤Ej≤1;
计算权重

$$W_j = \frac{g_j}{\sum_1^m g_j};$$

最终得到结果如下：

当 x6=1 时

	weight
x1	0.031311
x2	0.147026
x3	0.559844
x4	0.077940
x5	0.183879

运行完成!

当 x6=2 时

	weight
x1	0.035522
x2	0.184933
x3	0.536135
x4	0.075098
x5	0.168311

运行完成!

当 x6=3 时

当 x6=4 时

	weight
x1	0.140657
x2	0.226460
x3	0.438336
x4	0.033004
x5	0.161544

运行完成!

	weight
x1	0.100150
x2	0.192424
x3	0.524048
x4	0.033732
x5	0.149646

运行完成!

经过分析可知，随着 x6 的增大，x1 和 x2 的权重在增加，而 x4 和 x5 的权重基本保持不变，起主导作用的依旧是 x3，与第一问中所给结论一致，故可以通过 x6 的不同取值，改变模型的变量数，以达到更加准确的预测果汁饮料的质量，在第三问中逻辑回归的算法中，加入各变量的权重，对数据进行预测，是结果更加符合实际情况。

5.1.3

对于预测问题，可以考虑将现有的值划分为 80%的训练集和 20%的测试集，将数据拆分为测试和训练集的目的是避免过度拟合，将数据进行清洗删除影响性不大的数据和异常数据，对数据进行标准化处理。这种方法给予原始数据的均值（mean）和标准差（standard deviation）进行数据的标准化。经过处理的数据符合标准正态分布，即均值为 0，标准差为 1，转化函数为：：

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

部分数据转化如下:

```
[[ 0.40481082 -1.0593379   0.6999958   0.45966008 -0.37422873  2.3314474 ]
 [ 0.24675388 -0.16794516  0.6999958  -1.28791872  0.03742728 -0.93718418]
 [-2.18747947  0.88901752  0.6999958  -0.17574211 -0.2017119  -0.93718418]
 [ 0.93923107  0.69749338  0.6999958  -1.02502441  0.03742728 -0.93718418]
 [-1.17497611  0.59009666 -1.42858001  1.42033827 -0.3412682   0.15235968]
 [-1.09125288  0.28401602  0.6999958   0.75909224 -0.41279957  0.15235968]
 [ 1.51199186 -1.76994617 -1.42858001 -0.35397554 -2.53349423 -0.93718418]
 [-1.69218223  0.88901752  0.6999958   0.09784621 -0.2017119  -0.93718418]
 [ 1.11293721 -1.46476051  0.6999958  -0.49923578  0.03742728 -0.93718418]
 [ 1.16066727  0.28222608 -1.42858001 -0.60439351 -2.88413819 -0.93718418]
```

再将特征进行具有高斯分布和不同平均值和标准偏差的属性转换为平均值为 0 和标准偏差为 1 的标准高斯分布的标准化数据。利用逻辑回归的算法采用 Sigmoid 的函数实现。

$$f(x) = \frac{1}{1+e^{-x}}$$

函数的定义域为全体实数，值域在[0,1]之间，x 轴在 0 点对应的结果为 0.5。当 x 取值足够大的时候，可以看成 0 或 1 两类问题，大于 0.5 可以认为是 1 类问题，反之是 0 类问题，而刚好是 0.5，则可以划分至 0 类或 1 类。

对于 0-1 型变量:
y=1 的概率分布公式定义如下：

$$P(y=1)=p$$

y=0 的概率分布公式定义如下：

$$P(y=0)=1-p$$

其离散型随机变量期望值公式如下：

$$E(y)=1*p+0*(1-p)=p$$

采用线性模型进行分析，其公式变换如下：
P(y=0 | X) = θ0+θ1X1+θ3X3+θ5X5+θ6X6
P(y=1 | X) = θ0+θ1X1+θ3X3+θ5X5+θ6X6

5.2 模型求解

5.2.3

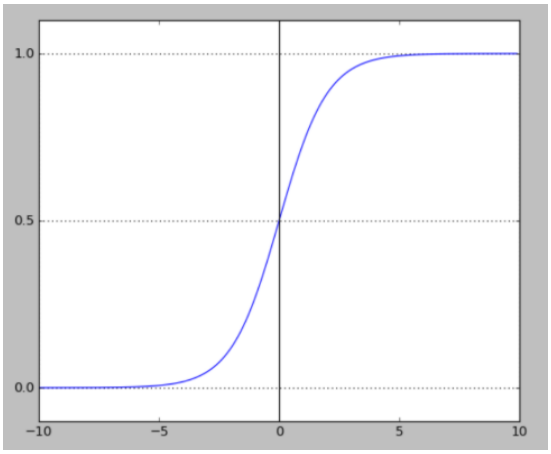
由于在这里概率 p 与因变量往往是非线性的，为了解决该类问题，我们引入了 logit 变换，使得 logit(p)与自变量之间存在线性相关的关系，逻辑回归模型定义如下：

$$\text{logit}(p)=In\left(\frac{p}{1-p}\right)=\theta_0+\theta_1x_1+\theta_2x_2+\ldots+\theta_nx_n$$

通过推导，概率 p 变换如下，这与 Sigmoid 函数相符，也体现了概率 p 与因变量之间的非线性关系。以 0.5 为界限，预测 p 大于 0.5 时，我们判断此时 y 更可能为 1，否则 y 为 0。

$$p=\frac{1}{1+e^{-(\theta_0+\theta_1x_1+\theta_2x_2+\ldots+\theta_nx_n)}}$$

其中 θ 是第二问中拟合的不同权重。
得到所需的 Sigmoid 函数后，接下来只需要和前面的线性回归一样，拟合出该式中 n 个参数 θ 即可。制 Sigmoid 曲线，输出如下图：



为了找到最优的参数，我们采取了交叉验证和网格搜索进行调优参数和误差分析，正如前面中所讨论的将数据拆分为测试和训练集对于避免过度拟合，这里进行交叉验证分类。 交叉验证（Cross Validation）是用来验证分类器的性能一种统计分析方法，基本思想是把在某种意义下将原始数据（dataset）进行分组，一部分作为训练集 （training set），另一部分作为验证集（validation set），首先用训练集对分类器进行训练，在利用验证集来测试训练得到的模型(model)，以此作为评价分类器的性能指标。找到最好的参数之后，保存训练好的模型，再将附件二的 19 个样本进行预测。

5.3 结果分析

5.3.3

在训练集和训练集的测试效果如下

```
[1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1
 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0
 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1
 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
[1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1
 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1
 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0]
```

结果真实度分析：

根据题目中所计算得到的预测值，可知有少部分的预测结果和实际不符合

其中准确率为：0.806477093206951

把附录 2 的数据带入模型中求得结果为：

```
[1   1 0   1   1   1   1   1 0   1   1   1   0   1   1   1   1   1   1]
```

5.4

针对问题四的要求，主要实现两个目的，一是提升果汁等级，二是降低果汁

成本，根据各成分在不同 x6 水平下的权重分析，我们可以更加抓住主要变量，更加科学有效的实现质量等级的提高，介于附件一所给样品的数量足够多，我们可以通过 Excel 的筛选方法，找到最接近的变量样本，然后修改主要变量，达到目的。

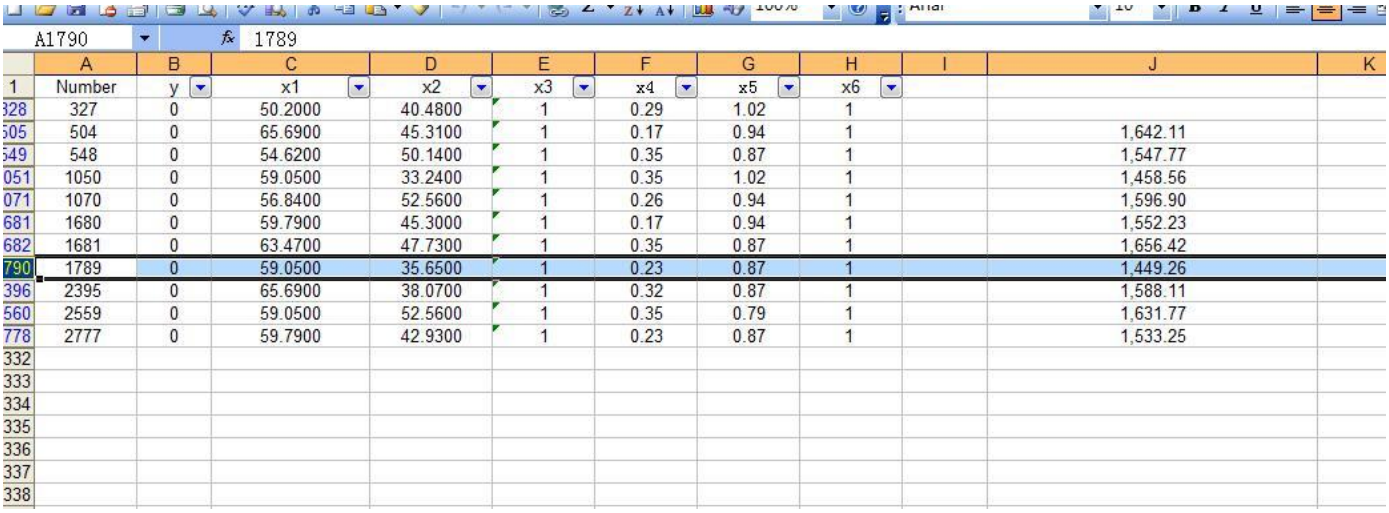
首先要说明的是，虽然把所有变量降到样本的最低标准一定能实现题中所给的要求，但是本着生产的时间效益和简便原则，这种改法是不适合高效率生产的方式，所以我们应该分析和抓住主要矛盾并找到成本最低的一组来实现。具体实现方法如下，举例说明：

利用 Excel 先求出 6329 组数据的 x1，x2，x4，x5 的平均值

μ1= 60.38　　μ2= 47.17，　μ4=1.00　　μ5= 0.46

A	B	C	D	E	F	G	H
Number	y	x1	x2	x3	x4	x5	x6
1	1	67.9000	54.9800	1	0.35	1.02	1
2	1	61.9000	50.0200	1	0.50	1.00	1
3	0	60.4700	41.2700	0	0.56	0.89	3
4	1	70.1100	57.3900	1	0.35	0.87	1
5	1	65.4900	54.1300	1	0.48	1.03	3
6	1	70.6100	37.4100	0	0.60	1.27	4
7	1	67.6500	33.0000	1	0.63	0.88	4
8	1	45.9300	54.9300	0	0.56	0.09	1
9	0	36.8600	36.9000	0	0.62	1.04	2
10	1	75.1500	52.6400	0	0.34	0.07	1
11	1	71.7800	44.3200	0	0.60	0.87	1
12	1	74.5400	33.2400	1	0.41	1.46	1
13	0	65.4900	63.5200	1	0.63	0.80	4
14	1	42.9100	52.3900	1	0.41	1.06	1
15	1	57.4300	69.1700	1	0.42	0.82	1
16	1	51.6200	39.0100	0	0.48	1.19	2
17	1	43.8900	44.7400	1	0.54	0.95	4
18	1	75.5900	31.2600	1	0.46	1.26	2
19	1	79.2200	35.6300	0	0.24	1.56	1

罗列出预测的效果，并对每一组数据，在附件一的样本中，筛选符合的方案，这是在测试某一组数据时的最优解，



A1790　　1789

	A	B	C	D	E	F	G	H	I	J	K
1	Number	y	x1	x2	x3	x4	x5	x6			
328	327	0	50.2000	40.4800	1	0.29	1.02	1			
505	504	0	65.6900	45.3100	1	0.17	0.94	1		1,642.11	
549	548	0	54.6200	50.1400	1	0.35	0.87	1		1,547.77	
051	1050	0	59.0500	33.2400	1	0.35	1.02	1		1,458.56	
071	1070	0	56.8400	52.5600	1	0.26	0.94	1		1,596.90	
681	1680	0	59.7900	45.3000	1	0.17	0.94	1		1,552.23	
682	1681	0	63.4700	47.7300	1	0.35	0.87	1		1,656.42	
790	1789	0	59.0500	35.6500	1	0.23	0.87	1		1,449.26	
396	2395	0	65.6900	38.0700	1	0.32	0.87	1		1,588.11	
560	2559	0	59.0500	52.5600	1	0.35	0.79	1		1,631.77	
778	2777	0	59.7900	42.9300	1	0.23	0.87	1		1,533.25	
332											
333											
334											
335											
336											
337											
338											

下面说明某些匹配最优解的不合适性，需要通过比较均值来寻找最适合的修

改途径。
这是需要修改的数据：

1	42.9100	52.3900	1	0.41	1.06	1
1	57.4300	69.1700	1	0.42	0.82	1
1	51.6200	39.0100	0	0.48	1.19	2
1	43.8900	44.7400	1	0.54	0.95	4

根据 excel 的拟合分析，筛选给出的方案只是无限制的缩小 x1 的分量，这样虽然达到了题目的要求，也使得成本降到最低，但并不适合生产时的具体调整，

1152	0	32.4300	48.4800	1	0.37	0.69	1		1,210.85		
1152	0	32.4300	48.4800	1	0.37	0.69	1		1,210.85		
1243	0	36.1500	31.7600	0	0.40	0.87	1		1,052.06		
1946	0	30.1600	36.9900	1	0.47	0.89	1		1,111.20		
745	0	64.0100	31.1100	1	0.41	0.87	1		1,620.35		

经过分析可知，此类变量中 x1 的值十分接近 μ1= 60.38，而 x2 却大大超出了 μ2= 47.17，所以根据分析，修改 x2 变量会更加有利于提高果汁饮料的质量，而且在实际的流水线上这也是最简便的方式。

因此我们最终得到了修改的数据，由于该算法主要是来自于对样本数据的拟合，因此其拟合效果将会比模型更加准确，也是得结果更加稳定。

六、模型改进、评价与推广

6.1.模型改进

我们已经用 python 代码实现了 logistic 回归，这里我们主要介绍如何来改进 logistic 回归算法的效率和性能。

使用随机向下升梯度算法：

实现 logistic 回归，随机梯度向下算法和梯度向下升算法的效果差不多，梯度向下法每次迭代的时候是使用所有的数据集进行迭代的，而随机梯度向下算法每次迭代的时候是使用一个数据进行迭代的。所以，从时间上来说，随机梯度算法要节省不少时间，一般对于数据集比较大的数据都是采用随机梯度算法（比如果汁的数据集就有六千多个样本）。

具体算法如下逻辑回归模型 f(θ):

$$\frac{\partial}{\partial\theta_j}(l(\theta))=\frac{\partial}{\partial\theta_j}(\sum_{i=1}^{m}y^{(i)}\log(h(x^{(i)}))+(1-y^{(i)})\log(1-h(x^{(i)})))$$

$$=(\frac{y^{(i)}}{h(x^{(i)})}-(1-y^{(i)})\frac{1}{1-h(x^{(i)})})\frac{\partial}{\partial\theta_j}(h(x^{(i)}))$$

$$=(\frac{y^{(i)}}{g(\theta^T x^{(i)})}-(1-y^{(i)})\frac{1}{1-g(\theta^T x^{(i)})})\frac{\partial}{\partial\theta_j}(g(\theta^T x^{(i)}))$$

$$=(\frac{y^{(i)}}{g(\theta^T x^{(i)})}-(1-y^{(i)})\frac{1}{1-g(\theta^T x^{(i)})})g(\theta^T x^{(i)})(1-g(\theta^T x^{(i)}))\frac{\partial\theta^T x^{(i)}}{\partial\theta_j}$$

$$=(y^{(i)}(1-g(\theta^T x^{(i)})-(1-y^{(i)})g(\theta^T x^{(i)})))x_j$$

$$=(y^{(i)}-h_\theta(x^{(i)}))x_j$$

故参数更新公式为：

$$\theta_j := \theta_j - a\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})\cdot x_j$$

加入正则项：

正则化有助于生成一个稀疏权值矩阵，进而可以用于特征选择。稀疏矩阵指的是很多元素为 0，只有少数元素是非零值的矩阵，即得到的线性回归模型的大部分系数都是 0. 在预测或分类时，那么 x3, x6 这样的离散行数据显然难以选择，但是如果代入这些特征得到的模型是一个稀疏模型，表示只有少数特征对这个模型有贡献，绝大部分特征是没有贡献的，或者贡献微小（因为它们前面的系数是 0 或者是很小的值，即使去掉对模型也没有什么影响），此时我们就可以只关注系数是非零值的特征。这就是稀疏模型与特征选择的关系。

假设有如下带 L1 正则化的损失函数：

$$J = J_0 + \alpha\sum_w |w|$$

其中 J0 是原始的损失函数，加号后面的一项是 L1 正则化项， α 是正则化系数。注意到 L1 正则化是权值的绝对值之和，J 是带有绝对值符号的函数，因此 J 是不完全可微的。要通过一些方法（比如梯度下降）求出损失函数的最小值。当我们在原始损失函数 J0 后添加 L1 正则化项时，相当于对 J0 做了一个约束。令 L=，则 J=J0+LJ，此时我们的任务变成在 L 约束下求出 J0 取最小值的解。考虑果汁的自变量是六维维的情况，即有六个权值 w1、w2、w3、w4、w5、w6，此时 L=|w1|+|w2|+|w3|+|w4|+|w5|+|w6|对于梯度下降法，求解 J0 的过程可以拟合出

等值线，从而求得关于成本的最优解。

6.2. 模型评价

模型的优点：

（1）该模型适用于需要得到一个分类概率的场景，预测结果的区间为 0 到 1；

（2）该模型简单易懂，容易被使用和理解；

（3）逻辑回归模型容易实现，并且具有在线算法实现，可以用较少的内存和时间处理大型的数据，在时间和内存需求上都非常高效；

模型的缺点：

（1）结果对数据变量的拟合程度较低，分类精度不高；

（2）若两个相关度较高的自变量同时放入回归模型，可能会造成较弱的自变量回归符号被逆转，导致结果出现误差。

建立模型时数据较多，拟合程度没有达到预期，在预测附件三给出的果汁质量时会出现一些偏差，但是可以通过模型改进来实现优化，尽可能地提高预测的准确度。

6.3 模型推广

逻辑回归模型运用广泛，对于目标值为 0 或 1 的二分类模型和多元分类模型更是好用。我们从果汁的质量好坏出发，随着生活水平的不断提高，人们开始追求更加健康、绿色的生活方式。此模型可以利用大量的样本数据进行预测产品质量的好坏。我们利用它进行了果汁质量的预测，也进行改进算法从而进行了成本的估计，是有效的分类模型。不仅如此，它还广泛运用于机器学习领域之中，是机器学习里的经典传统算法。

参考文献：

[1].姜启源,谢金星，叶俊等.数学模型（第五版）.北京：高等教育出版社.2018.P233-250

[2].孙逸敏.利用 SPSS 软件分析变量间的相关性.新疆：高等教育出版社.2018.P1-3

[3]. Logistic 回归原理及公式 https://blog.csdn.net/pql925/article/details/79021464

[4]. 机器学习--Logistic 回归计算过程的推导
https://blog.csdn.net/ligang_csdn/article/details/53838743

[5]. 信息熵计算权重
https://blog.csdn.net/wake_me_up123/article/details/73139770

[6]. 逻辑回归模型
https://blog.csdn.net/weixin_39910711/article/details/81607386#%C2%A02.2%20%E4%BC%BC%E7%84%B6%E5%87%BD%E6%95%B0%E7%9A%84%E6%B1%82%E8%A7%A3-%E6%A2%AF%E5%BA%A6%E4%B8%8B%E9%99%8D

附录：

Python 语言编程代码，JupyterNotebook 软件实现

```
##  权重的测量

import  pandas  as  pd
import  numpy  as  np
from  pandas  import  Series,  DataFrame


datafile  =  './data/'
data_train  =  pd.read_excel(datafile  +  'data.xls')

#  Dependents 变量缺失值比较少，直接删除，对总体模型不会造成太大影响。
对缺失值处理完之后，删除重复项
data_train  =  data_train.dropna()
data_train  =  data_train.drop_duplicates()
data_train.drop("Number",axis  =  1  ,inplace=True)
#data_train.drop(6329,axis  =  0  ,inplace=True)

import  math
from  numpy  import  array

#定义熵值法函数
def  cal_weight(x):
        '''熵值法计算变量的权重'''
        #  标准化
  x  =  x.apply(lambda  x:  ((x  -  np.min(x))  /  (np.max(x)  -  np.
min(x))))

        #  求 k
        rows  =  x.index.size    #  行
  cols  =  x.columns.size    #  列
  k  =  1.0  /  math.log(rows)

        lnf  =  [[None]  *  cols  for  i  in  range(rows)]

        #  矩阵计算--
  #  信息熵
  #  p=array(p)
        x  =  array(x)
        lnf  =  [[None]  *  cols  for  i  in  range(rows)]
        lnf  =  array(lnf)
        for  i  in  range(0,  rows):
                for  j  in  range(0,  cols):
```

```
                        if  x[i][j]  ==  0:
                                lnfij  =  0.0
                        else:
                                p  =  x[i][j]  /  x.sum(axis=0)[j]
                                lnfij  =  math.log(p)  *  p  *  (-k)
                        lnf[i][j]  =  lnfij
        lnf  =  pd.DataFrame(lnf)
        E  =  lnf

        #  计算冗余度
  d  =  1  -  E.sum(axis=0)
        #  计算各指标的权重
  w  =  [[None]  *  1  for  i  in  range(cols)]
        for  j  in  range(0,  cols):
                wj  =  d[j]  /  sum(d)
                w[j]  =  wj
                #  计算各样本的综合得分,用最原始的数据

  w  =  pd.DataFrame(w)
        return  w

#  计算 df 各字段的权重
w  =  cal_weight(data_train[['x1','x2','x3','x4','x5','x6']])    #
调用 cal_weight
w.index  =  data_train[['x1','x2','x3','x4','x5','x6']].columns
w.columns  =  ['weight']
print(w)
print('运行完成!')


##  逻辑回归的实现

import  pandas  as  pd
from  sklearn.preprocessing  import  StandardScaler        #标准化
from  sklearn.model_selection  import  train_test_split  ,      #分割
数据
from  sklearn.preprocessing  import  LabelEncoder
from  sklearn.linear_model  import  LogisticRegression#回归预测的 API
from  sklearn.metrics  import  accuracy_score

data  =  pd.read_excel("./data/data.xls",  index_col=False)

data.drop("Number",axis  =  1  ,inplace=True)
data.drop(6329,axis  =  0  ,inplace=True)
```

```
#data  =  data[data.x6==3]
X  =  data[['x1','x2','x3','x4','x5','x6']]
y  =  data.iloc[:,0]
#  '''#把目标值(M/B)转化为数字（0/1）
le  =  LabelEncoder()
y  =  le.fit_transform(y)

x_train,  x_test,  y_train,  y_test  =  train_test_split(X,  y,  test
_size=0.2)

scaler  =  StandardScaler()
x_train  =  scaler.fit_transform(x_train)
x_test  =  scaler.transform(x_test)

print(x_train[0:20])

new_data  =  pd.read_excel(r"C:\Users\77526\代码\data\附件
2.xlsx",  index_col=False)
new_x  =  new_data[['x1','x2','x3','x4','x5','x6']]

for  C  in  range(1,1000):
        classifier  =  LogisticRegression(solver='sag',C=1)
        classifier.fit(x_train,  y_train,sample_weight=[])

        y_pred  =  classifier.predict(x_test)
        print(y_pred[0:500])
        print(y_test[0:500])
        acc  =  accuracy_score(y_test,  y_pred)
        print(classifier.score(x_test,y_test))
        print(classifier.predict(new_x))
```