基于膳食质量评价和决策树算法的居民健康影响因素分析

摘要

本文研究影响城市居民身体健康的因素分析，问题一建立基于 DBI 和营养曲线的膳食质量评价模型来评价饮食合理性，问题二建立生活饮食习惯相关性分析模型分析生活饮食习惯与各因素间的关联度，问题三建立基于决策树慢性疾病关联性分析模型研究了慢性疾病与各因素之间的相关性，问题四建立基于 K-Means 聚类的健康改善模型对各类人群提出改进性意见。

针对问题一，本文对不同人群的饮食规律和膳食平衡进行分析，并建立基于 DBI 和营养曲线的膳食质量评价模型。首先对附件二中不规范的数据进行缺失值填补、异常值删除等操作。然后主要从两部分评估饮食合理性。第一部分统计分析**不同年龄段和性别**人群的饮食规律和膳食平衡情况；第二部分，建立**基于 DBI 和引入 Sigmoid 函数的营养曲线**的膳食质量评价模型求取综合评分，求得居民的**膳食质量平均分为 26.7608**。再根据得分和**单因素分析**分析合理性和存在的问题。

针对问题二，本文建立了生活饮食习惯相关性分析模型来研究生活、饮食习惯与各因素之间的相关性。首先用**主成分分析**法对饮食情况数据进行降维，综合主成分求取反映饮食习惯的三餐和食物摄取情况的综合得分，然后根据因素属性选取 **Kendall 检验**或**卡方检验**方法，求得不同因素与生活、饮食的相关性，并从**宏观和微观**层面进行分析。

针对问题三，本文建立了基于决策树的慢性疾病关联性分析模型，从**宏观和微观**层面分别研究**糖尿病**、**高血压**、**代谢综合征**、**低高密度脂蛋白血症四种慢性病**与各因素之间的关联度。本文选取吸烟量、饮酒量、运动量、膳食质量评分、活动量、BMI、年龄、工作性质、性别、文化程度作为决策树的特征变量，根据慢性疾病诊断结果建立**基于决策树的慢性疾病关联性分析模型**，分析结果得**年龄**、**BMI**、**膳食质量评分**与四个慢性疾病的关联程度最大。最终决策树的平均准确率达到 **91.61%**，说明模型效果良好。

针对问题四，本文建立了**基于 K-Means 聚类的健康改善模型**，对各类人群和患病人群的健康情况改善进行分析。首先根据年龄、BMI、膳食质量评分进行聚类，利用**肘部法确定 K 值为 3**。然后，先根据问题二的初步结果，对患病居民和不患病居民提出合适的建议，再利用**单因素分析**的方法对每类人群进行情况分析，分别命名为 **“年轻工作党”**、**“年轻健康党”**、**“老年风险党”**，最后为每类群体提出合适的建议。

本文的亮点有：1、采用 DBI 和营养曲线综合评分对膳食质量进行评价；2、在进行影响身体健康情况的因素分析时从宏观和微观层面多角度考虑.

关键词：营养曲线　DBI　膳食质量评价　影响因素分析　决策树　K-Means 聚类

目录

1 问题重述 **1**

1.1 问题背景 1

1.2 问题提出 1

2 模型假设 **2**

3 符号说明 **2**

4 问题一模型的建立及求解 **3**

4.1 问题一分析 3

4.2 数据预处理 3

4.3 不同类别人群的饮食规律和膳食平衡分析 4

4.3.1 人群分类 4

4.3.2 饮食规律情况 5

4.3.3 膳食平衡情况 5

4.3.4 饮食习惯存在的问题和合理性分析 6

4.4 基于 DBI 和营养曲线的膳食质量评价模型 6

4.4.1 评价指标 6

4.4.2 基于 Sigmoid 函数的营养曲线评分 7

4.4.3 DBI 评分 9

4.4.4 综合评价 9

4.4.5 饮食习惯存在的问题和合理性分析 10

5 问题二模型的建立与求解 **11**

5.1 问题二分析 11

5.2 数据预处理 11

5.3 主成分分析降维 11

5.4 生活饮食习惯相关性分析模型 12

5.4.1 饮食习惯与生活习惯指标 12

5.4.2 因素属性 13

5.4.3 相关性检验方法 13

5.5 结果分析 14

5.5.1 饮食习惯与各因素间的相关性结果分析 14
5.5.2 生活习惯与各因素间的相关性结果分析 16
6 问题三模型的建立与求解 16
6.1 问题三分析 16
6.2 数据预处理 16
6.3 基于决策树慢性疾病关联性分析模型 17
6.3.1 慢性疾病的诊断与标签说明 17
6.3.2 特征变量说明 17
6.3.3 决策树算法求解 18
6.4 结果分析 19
6.4.1 常见慢性病与各因素间的相关性分析 19
7 问题四模型的建立与求解 22
7.1 问题四分析 22
7.2 基于 K—Means 聚类的健康改善模型 22
7.2.1 K—Means 聚类算法求解 22
7.3 结果分析 24
7.3.1 患病人群的健康情况改善分析 24
7.3.2 各类别人群健康情况改善分析 25
8 模型总结与评价 26
8.1 模型总结 26
8.1.1 模型优点 26
8.1.2 模型缺点 26
8.2 模型改进 26
参考文献 27
附录 A 部分居民的膳食质量评分 28
附录 B 居民对照组与病例组的特征统计表 34
附录 C 居民分类组的特征统计表 38
附录 D 问题一源代码 41
4.1 数据预处理代码 41
4.2 绘制营养曲线代码 47

附录 E　问题二源代码 **48**
5.1 相关性分析代码 48
5.2 绘制箱线图代码 52
附录 F　问题三决策树源代码 **53**
附录 G　问题四 K-Means 聚类源代码 **55**

1 问题重述

1.1 问题背景

“十三五规划”作出实施健康中国战略的决策部署，中共中央国务院印发《“健康中国 2030”规划纲要》,[1] [2]作为促进健康中国建设的行动纲领。由此反映出，居民的健康需求与健康意识已经达到了一个新的高度。但如今慢性非传染性疾病致死是我国第一大死亡病因,[3]已经成为威胁我国居民身体健康的重要问题，得到国家社会广泛关注。经研究发现，通过健康合理的饮食和生活习惯，可以有效避免慢性非传染性疾病的发生。

图 1　问题背景图

1.2 问题提出

附件一是某个城市的卫生健康研究部门对部分居民所做的“慢性非传染性疾病及其相关影响因素流行病学”调查问卷表，附件二是附件一问卷的调查数据结果，附件三是最新由中国营养学会修订的《中国居民膳食指南》中的为平衡居民膳食提出的八条准则。

问题一：基于附件三，分析附件二居民饮食习惯的合理性、并说明附件二居民饮食习惯存在的问题。

问题二：对居民的饮食习惯和生活习惯与性别、年龄、婚姻状况、职业、文化程度等因素进行相关性分析。

问题三：基于附件二，对常见疾病与饮酒、吸烟、饮食习惯、运动、工作性质、生活习惯等因素进行相关性分析。

问题四：基于附件二，对居民进行分类，并对各个类别人群提出在健康饮食、运动等方面针对性的建议。

2 模型假设

(1) 假设每个被调查者在填写个人基本信息和判断型数据时都是如实填写，若空则判断为缺失值处理;

(2) 假设附件二居民调查结果数据可反映整个社会群体的情况;

（3）假设食物的摄入量仅受食物规定的单位量影响，与具体食物种类无关;

（4）假设问题三中，被社区或以上医院的医生诊断为患病的居民现在仍患病。

3 符号说明

符号	说明
n	附件二居民总人数
m	指标食物种类数
a_j	第 j 种指标的最小适宜摄入量
b_j	第 j 种指标的最大适宜摄入量
P_i	居民 i 的营养曲线评分
Q_i	居民 i 的 DBI 评分
G_i	居民 i 的膳食质量综合评分
V	饮食习惯综合得分
a_{1i}	吸烟量
a_{2i}	饮酒量
a_{3i}	活动量

注：表中未列出及重复的符号均以首次出现处为准。

4 问题一模型的建立及求解

4.1 问题一分析

问题一要求根据附件三信息对附件二居民的饮食习惯的合理性做评价，并说明存在的问题。本文主要从两方面进行评价，一方面从饮食规律和膳食平衡分析不同类别人群的饮食习惯存在的问题，另一方面从摄入量方面评价饮食习惯的合理性。首先，由于不同类别人群体质不同，我们将居民分为青年、中年、老年，分别统计不同类别群体的三餐饮食规律和膳食平衡情况，从年龄和性别两个角度出发分析不同人群的饮食存在不同的问题。然后，建立 DBI 和营养曲线的膳食质量评价模型[4] [5]，选出九项食物指标，引入 Sigmoid 函数求取营养曲线评分，再基于 DBI 评价体系求取评分，综合两个评分结果，在摄入量方面计算饮食习惯的合理程度。问题一思路图如下所示。

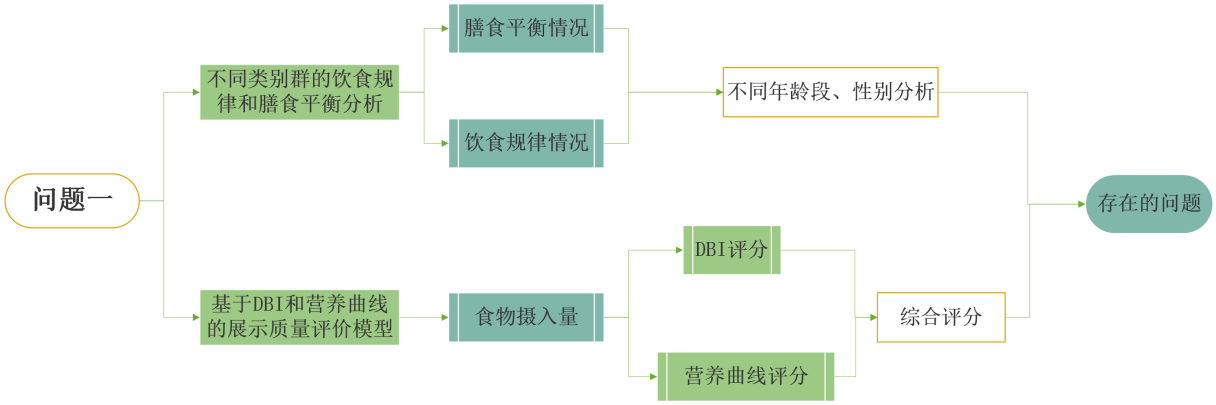

图 2　问题一思路图

4.2 数据预处理

基于附件一调查问卷信息，首先对附件二调查结果数据进行分析处理。经过统计，本文发现有缺失值和异常值存在。

为了模型的简洁，本文假设所有数据均真实可靠，当出现逻辑不符时，以不吃早、午、晚餐天数问题、是否食用 $D4-D30$ 类食物判别问题及平均每次食用量和次数问题结果为准。

由于 $D1-D3$、$D4-D30$、$D31-D37$ 三类问题独立不相关，为了避免剔除一类残缺异常但另一类完整无误的数据，在后续数据处理时将三者分开处理。

附件二的数据缺失与处理情况如下：

(1) 存在部分 D 类问题结果都缺失的情况和整行数据存在 90% 以上的缺失的情况，经统计该部分数据占总体比例较小，对整体影响不大，本文采取剔除处理；

(2) 部分数据存在不吃三餐天数结果小于 7，但后面就餐地点空缺的情况，但占比极小，本文采取剔除处理；

(3) 部分数据存在是否食用该食物列空缺的情况，但是用食用频率和平均每次食用量不全为空的情况，本文通过人为判断补全是否食用列的数据。

(4) 部分数据存在前面是否食用该食物结果为“是”，但是用食用频率和平均每次食用量缺失的情况，本文的食用频率采用三列中数据量最大的一列的众数进行填充，平均每次食用量直接用该列数据的众数填充。

附件二的数据异常与处理情况如下：

(1)$D1-D3$ 问题数据存在一周内不吃三餐天数与在不同地点就餐天数和不为 7 的情况，不符实际，但占比极小，本文采取剔除处理；

(2)$D4-D30$ 问题数据存在未食用该食物，但食用该食物频数不为空的情况，占比极小，本文采取剔除处理；

4.3 不同类别人群的饮食规律和膳食平衡分析

本文首先主要从从宏观层面对不同类别人群的三餐规律情况、食物多样方面进行分析，在饮食规律和膳食平衡两个方面评价居民饮食习惯的合理性.

4.3.1 人群分类

经查找资料发现不同年龄不同性别的居民体质情况差异较大，且不同类型的人群饮食习惯不同，因此需要分类研究。本文基于附件二，参考世界卫生组织经过全球人体素质和平均寿命标准进行测定而规定的 2023 年“年龄划分标准”，根据性别和年龄对居民进行分类，统计结果如下图所示，可以看出附件二居民年龄分布情况较符合整个社会年龄分布结构，可见数据实用性良好。

表 1　性别年龄分类结果

年龄划分范围	阶段	性别	人群数量
18—44 岁	青年	男	832
		女	1066
45-59 岁	中年	男	1794
		女	2008
60—123 岁	老年	男	788
		女	1221

4.3.2 饮食规律情况

针对三餐规律情况，基于附件二 $D1-D3$ 问题数据，计算每天坚持吃早餐、午餐、晚餐的人占比，统计结果如下

表 2　不同类别人群的三餐情况

年龄阶段	性别	吃早餐占比	吃午餐占比	吃晚餐占比
青年	男	71.39%	97.36%	98.92%
	女	79.92%	98.50%	99.25%
中年	男	73.91%	98.66%	99.50%
	女	84.66%	98.8%	99.1%
老年	男	86.93%	99.24%	99.11%
	女	93.94%	99.34%	99.59%

4.3.3 膳食平衡情况

针对膳食平衡情况，根据附件三的膳食指南，人们应坚持谷类为主的平衡膳食模式，每日的膳食应包括谷薯类、蔬菜水果、畜禽鱼蛋奶和豆类食物。本文采用单因素分析的方法，将附件二 $D4-D30$ 类食物分为以上四类，计算每天坚持吃四类食物的人在各自人群中的占比，统计结果如下所示。从表中可以看出，居民基本符合每天坚持吃四种不同类别食物的要求，绝大多数人能够达到膳食多样性要求。

表 3　不同类别人群每天坚持吃四类食物的占比

年龄阶段	性别	占比
青年	男	100%
	女	100%
中年	男	100%
	女	100%
老年	男	99.87%
	女	100%

4.3.4 饮食习惯存在的问题和合理性分析

基于上述统计结果，本文得到各类人群的饮食结构情况如下所示。在年龄纵向上，从图中可以看出，随着年龄增长，早餐、中餐、晚餐习惯逐渐规律，突出体现在早餐习惯上。总体上看，老年相对于青年、中年群体饮食更加合理的。青年群体的早餐习惯较差，这可能与青年群体工作原因和生活作息有关。各类人群的膳食多样与合理搭配搭配均良好，绝大多数在膳食上能够做到食物多样、合理搭配，只有极小部分的老年群体不能满足每天坚持吃四类食物。在性别横向上，女性无论是在三餐习惯规律上还是膳食多样搭配上几乎普遍比男性表现好，女性饮食更加合理，这可能与男性的社会压力普遍比女性大有关。

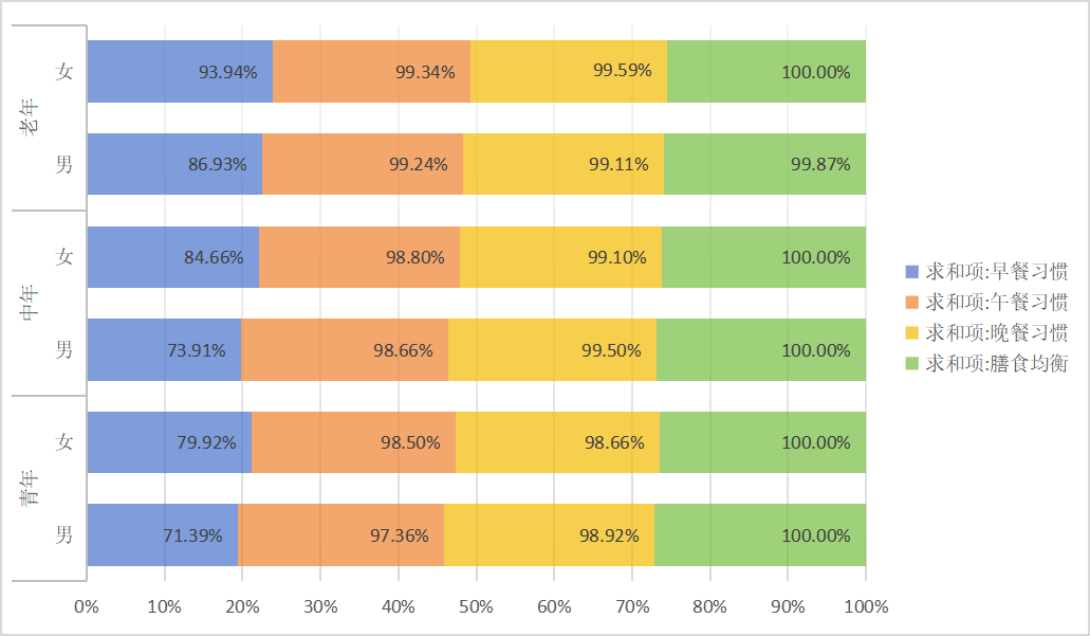

图 3　不同类别人群的饮食规律与膳食平衡情况图

4.4 基于 DBI 和营养曲线的膳食质量评价模型

本文建立膳食评价模型，基于附件二的 $D4-D37$ 问题数据，利用 DBI 和营养曲线进行综合评分，在食物摄入量上评价居民饮食的合理性。

4.4.1 评价指标

基于附件二、三信息，并结合中国居民膳食指南和膳食宝塔的核心内容，本文对食物种类进行筛选，将筛选出的指标作为膳食质量评价模型的评价指标。将 $D4-D30$ 类食物依据膳食指南进行合并，得到谷类、薯类、蔬菜、水果、畜禽鱼蛋、奶类、豆类七类指标。为了模型的简洁性，将动、植物油合并为食用油作为一项指标食物，并在调味品

中选取不同摄取量对人体健康影响最大的盐作为指标。并基于附件一重量折算参照表，将食物摄入量用摄入重量来衡量。

最终将谷类、薯类、蔬菜、水果、畜禽鱼蛋、奶类、豆类、食用油、盐作为膳食质量评价模型的九项指标食物。

4.4.2 基于 Sigmoid 函数的营养曲线评分

传统上，常利用经典数学方法对膳食摄入进行评价，一般以是否达到膳食营养成分摄入推荐值作为评价标准，采用布尔变量来反映饮食是否合理，但由于变量取值只能为 0 或 1，最终只有合理与不合理两种评判结果。本文希望能够在 $[0,1]$ 上连续取值利用模糊数学表示食物不同摄入量的合理程度。因此，本文决定首先利用营养曲线来对膳食质量进行评分。

营养曲线指当其他营养成分处于适宜水平时，个体健康情况随某种食物摄入量变化而变化的曲线，反映了某种食物摄入量与健康之间的关系，从而我们可以利用健康状况作为指标来进一步反映饮食的合理性。

本文先为每个评价指标设置最小适宜摄取量与最大适宜摄取量两个参数，在两个参数范围内，假定健康情况最佳，处在范围外，健康情况就会下降。经查阅相关文献，将两个参数设定如下：

表 4　各指标的最小、大适宜摄取量

指标	最小适宜摄入量/g	最大适宜摄入量/g
蔬菜	300	500
水果	200	350
奶类	300	500
谷类	200	300
薯类	50	100
豆类	25	35
畜禽鱼蛋	120	200
食用油	25	30
盐	3	6

首先，本文利用两个 Sigmoid 函数来求取每个指标的营养曲线，两个函数分别反映了摄取量由过低到适量和葱适量到过量健康情况的变化，具体公式如下：

$$f_{ij} = \frac{1}{1+e^{-k_1(x_{ij}-a_j)}} - \frac{1}{1+e^{-k_2(x_{ij}-b_j)}} \tag{1}$$

其中 f_{ij} 表示居民 i 的第 j 种指标的营养曲线健康状况得分，a_j 表示第 j 种指标的最小适宜摄入量，b_j 表示第 j 种指标的最大适宜摄入量，x_{ij} 表示居民 i 的第 j 种指标的摄入量，k_1 和 k_2 表示控制曲线陡度的参数，本文都设置为 0.1.

当 $x_{ij} < a_j$ 或 $x_{ij} > b_j$ 时，f_{ij} 都接近 0，表示该指标摄入量不足或过量时，都对健康情况不利，饮食不合理；当 $a_j \leq x_{ij} \leq b_j$ 时，f_{ij} 都趋向于 1，表示指标摄入量处于上下限之间时，饮食较合理。然后，基于膳食金字塔并综合文献与专家评分，本文将各个指标影响个体健康情况的权重设置如下：

表 5　各指标影响个体健康情况的权重

指标	谷类	薯类	蔬菜	水果	畜禽鱼蛋	奶类	豆类	食用油	盐
权重	0.2	0.2	0.125	0.125	0.1	0.075	0.075	0.05	0.05

之后，绘制每项指标的营养曲线，综合各指标对个体健康的影响情况，利用如下公式计算各居民的膳食营养曲线得分。

$$\boldsymbol{P_i} = \sum_{j=1}^{m} \boldsymbol{f_{ij}} \times \boldsymbol{\omega_j} \tag{2}$$

其中 P_i 表示居民 i 的营养曲线得分，ω_j 表示第 j 项指标的权重，m 表示指标个数。

最后，对各指标的营养曲线进行分析，根据健康程度将摄入量划分为不足区、安全区、最佳去、过量区。本文以水果为例，其摄入量营养曲线如下所示。由图可以看出水果最合理摄入量为 250——300g，摄入量处于此范围内最有助于身体健康。

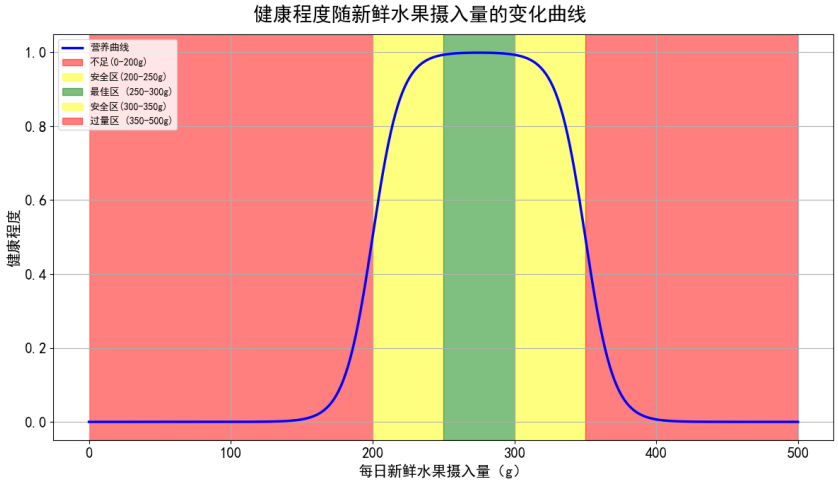

图 4　水果摄入量营养曲线

4.4.3 DBI 评分

膳食平衡指数（DBI）是依据中国居民膳食指南及平衡宝塔建立的，可较好反映膳食平衡状况。因此，本文决定再用 DBI 对附件二居民的饮食质量进行评分。

首先，本文根据 DBI 指标体系的要求，按照膳食指标数据大小与选定的评分规则进行匹配，得到各指标的分值, 匹配规则为

$$\text{If } x_{ij} \in \left[c_j^{(k)}, c_j^{(k+1)}\right), \text{Then } \mathrm{S}_{ij} = t^{(k)}. \tag{3}$$

其中 $c_j^{(k)}$ 表示指标 j 的评分规则中的第 k 个取值区间的端点阈值，S_{ij} 表示居民 i 第 j 项指标的得分，$t^{(k)}$ 表示第 k 取值区间对应的分值。

由于 DBI 采用的是正负双向分值，当 $S_{ij} > 0$ 时表示指标 j 对应的食物摄入过量，当 $S_{ij} < 0$ 时表示指标 j 对应的食物摄入不足。因此，$|\mathrm{S}_{ij}|$ 越大则说明该事物摄入量越不合理。基于此，本文设置评分规则如下 然后，利用 DBI 中的膳食质量距来综合反映

表 6　评分规则

$\lvert\mathrm{S}_{ij}\rvert$ /分	摄入状况
0	合理
3	不足或过量

一个特定膳食中的营养均衡问题，其计算公式为

$$D_{\mathrm{i}} = \sum_{j=1}^{m} S_{ij} \quad (i = 1, 2...n) \tag{4}$$

其中，D_{i} 表示居民 i 的膳食质量距，n 表示附件二居民总人数。

最后，为便于评分，本文利用下面式子将膳食质量转换到 $[0, 100]$ 区间内，并将其转化成极大型指标，从而可以通过最终得分情况来反映膳食平衡。

$$Q_i = \max\{D_i\} - \frac{\max\{D_i\}}{\min\{D_i\}} D_i. \tag{5}$$

其中 Q_i 表示居民 i 的 DBI 评分。

4.4.4 综合评价

营养曲线主要反映了食物摄入量与健康状况的关系，DBI 主要反映了食物摄入量与膳食均衡的关系。综合两者，将营养曲线和 DBI 评分线性加权，得到综合评价分值，本文用其评价在食物摄入量方面居民饮食习惯的合理性。计算公式为

$$G_i = \alpha P_i \times 100 + (1 - \alpha) Q_i \tag{6}$$

其中 α 表示营养曲线评分权值，$1-\alpha$ 表示 DBI 评分权值，本文 α 取值为 0.25.

4.4.5 饮食习惯存在的问题和合理性分析

本文以前一百名居民的膳食质量评分为例 (详细数据请见附录)，评价结果如下图所示。附件二居民的的平均膳食质量评分为 **26.7608**. 从结果数据可以看出，居民的评分都普遍较低。

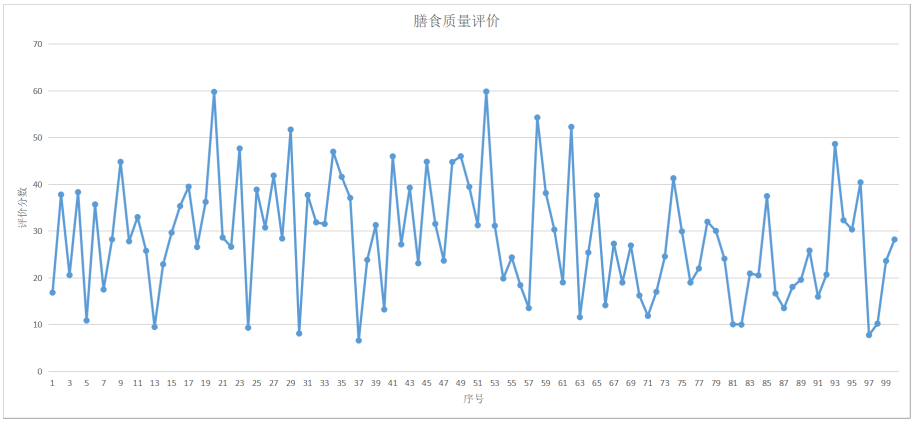

图 5　膳食质量评价结果

对能满足各项食物膳食指南摄入量条件的居民占比进行统计，发现能够满足各类食物摄入量条件的居民占比都较小，满足豆类、奶类、薯类的摄入量的居民甚至小于 10%，说明只有极少数居民达到了饮食摄入量条件，大部分居民的饮食都或多或少存在问题，需要改进。也印证了上述评分的正确性。根据上述统计结果，建议居民多吃蔬果、奶类、谷类、豆类食物，适量吃畜禽鱼蛋类食物，控制盐油的摄取。居民可以参考膳食指南和膳食金字塔，根据各类食物最佳摄入量，合理规划自己饮食，改进饮食习惯。问题二思路图如下所示。

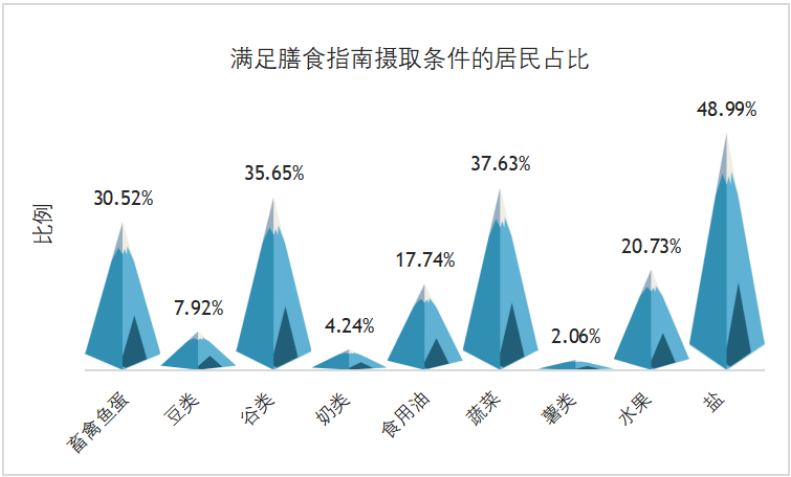

图 6　满足膳食指南摄入量条件的居民占比直方图

5 问题二模型的建立与求解

5.1 问题二分析

问题二要求分析居民的生活习惯和饮食习惯是否与年龄、性别、婚姻状况、文化程度、职业等因素相关。首先对附件二其余数据进行数据清洗。然后通过主成分分析法对附件二数据进行降维处理，并求得饮食习惯指标数据，将做休闲、家务活动强度作为生活习惯指标。之后根据因素属性选择合适的相关性检验方法，对饮食生活习惯和各因素之间进行相关性分析。最后分别对饮食、生活习惯与各因素间的相关性结果进行分析。

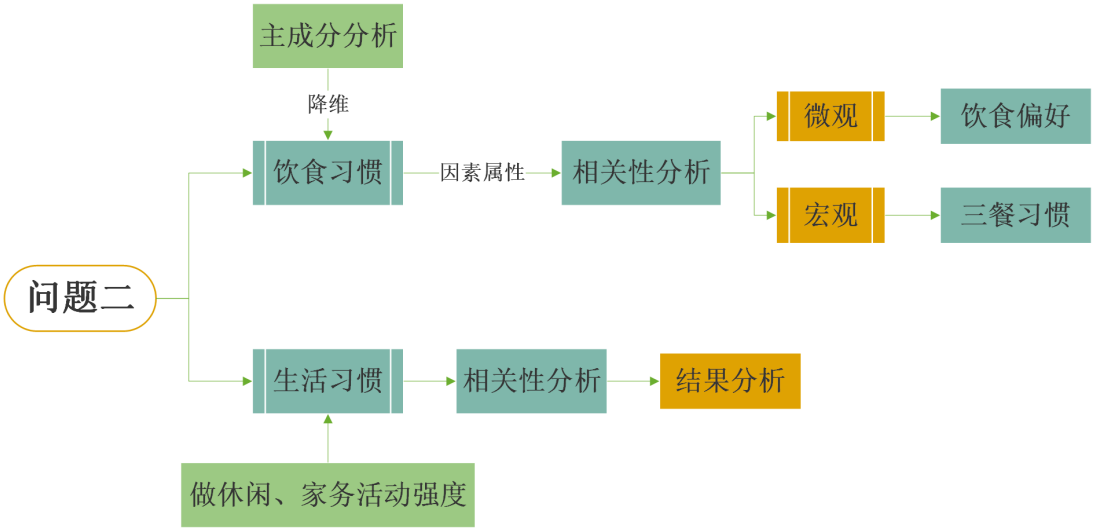

图 7　问题二思路图

5.2 数据预处理

依照问题一饮食情况数据的预处理规则，对附件二中其他情况数据的缺失值与异常值也做相同的的检验和处理。

5.3 主成分分析降维

附件二饮食情况数据维度过高，为方便研究饮食习惯与各因素之间的关系，本文首先采用主成分分析[7]分别对 $D1-D3$、$D4-D37$ 数据进行降维，将得到的两个得分作为三餐情况和食物食用情况的综合反映指标。主成分分析具体步骤如下。

首先，对数据进行标准化得到数据集

$$X=[X_1,X_{2,\ldots,}X_d]=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} \tag{7}$$

其中 x_{ij} 表示附件二数据第 i 个居民的问题 j 的调查结果,d 表示调查结果列数即指标数。

然后，利用如下公式计算得到协方差矩阵为 R.

$$r_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ki} - \overline{X}_i)(X_{kj} - \overline{X_j}) = \frac{1}{n-1}\sum_{k=1}^{n} X_{ki}X_{kj}. \tag{8}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1d} \\ r_{21} & r_{22} & \cdots & r_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ r_{d1} & r_{d2} & \cdots & r_{dd} \end{bmatrix} \tag{9}$$

之后，计算 R 的特征值与特征向量，并用其计算累计贡献率

$$\beta_j = \frac{\sum_{k=1}^{j}\lambda_{\mathrm{k}}}{\sum_{k=1}^{d}\lambda_k}\,(j = 1, 2, \cdots, d) \tag{10}$$

其中 β_j 表示第 j 个指标的累计贡献率，λ_k 表示第 k 个指标的特征值。

最后，根据累计贡献率取值得到主成分，一般取累计贡献率超过 80% 的特征值对应的指标作为主成分。第 i 个主成分可表示为

$$F_i = a_{1i}X_1 + a_{2i}X_2 + \cdots + a_{di}X_d \quad (i = 1, 2, \cdots, r) \tag{11}$$

其中，F_i 表示数据在第 i 主成分上的投影，$a_i = (a_{1i}, a_{2i}..., a_{di})$ 表示第 i 个指标的特征向量,r 表示主成分个数。

本文选取累计贡献率在 80% 附近的对应的成分作为主成分。通过上述计算，$D4 - D37$ 的前五个主成分累计贡献率达到了 79.11%.

5.4 生活饮食习惯相关性分析模型

5.4.1 饮食习惯与生活习惯指标

本文用经降维得到的主成分来为每个居民计算一个综合评价得分，将综合的三餐得分和食物摄取得分作为衡量饮食习惯的指标。得分计算公式如下:

$$V = \sum_{i=1}^{r}\omega_i \times F_i \tag{12}$$

其中 ω_i 是主成分 i 的贡献率，将其作为该主成分计算综合得分时的权重。

经分析题目，本文用身体活动状况中 $E2$ 做休闲、家务活动强度来作为衡量生活习惯的指标。

5.4.2 因素属性

为方便后续研究饮食习惯、生活习惯与各因素之间的相关性，本文首先对各因素属性进行归类。将各因素分为有序分类变量、无序分类变量。有序分类变量是指其取值的各类别之间存在着程度上的差别。无序分类变量是指所分类别或属性之间无程度和顺序的差别。因此可以将年龄、性别、婚姻状况、文化程度、职业划分如下。

表 7　因素属性

属性	因素
有序分类变量	年龄、文化程度
无序分类变量	职业、婚姻状况、性别

5.4.3 相关性检验方法

本文针对不同属性的因素，在研究与饮食、生活习惯时采用不同的相关性分析方法。

本问在研究饮食习惯和有序分类变量即年龄、文化程度的相关性时，采用 Mann—Kendall（曼—肯德尔）检验法。**Kendall 检验**是一个无参数假设检验，使用计算得的相关系数去检验两个随机变量的统计依赖性。Kendall 相关系数的取值范围为 $[-1,1]$，当系数为 1 时，表示两个随机变量 X、Y 正相关；当系数为 -1 时，表示两个随机变量拥有完全负相关；当系数为 0 时，表示两个变量是相互独立无关。其计算公式如下：

$$\tau_b = \frac{C-E}{\sqrt{D-T_e}\sqrt{D-T_c}} \tag{13}$$

其中 C 表示两两中具有一致性的元素对数,E 表示两两中具有不一致性的元素对数，D 表示两两比较的对数，即 $\frac{n(n-1)}{2}$，T_c 表示不变对中 X 值不变的个数，T_e 表示不变对中 Y 值不变的个数.

本问在研究饮食习惯和无序分类变量即年龄、婚姻状况、文化程度、性别的相关性时，采用卡方检验。**卡方检验**常用于检验两个变量之间是否存在相关性或独立性。其基本公式如下。检验结果选取置信度 p 来作为判断变量相关性的依据。将 p 值的阈值设置为 0.05，置信度小于 0.05，则认为有显著影响。

5.5 结果分析

5.5.1 饮食习惯与各因素间的相关性结果分析

本文分别从宏观和微观两个层面解释饮食习惯与各因素间的相关性。在宏观上，主要从各类人群的三餐得分与多个因素进行相关性分析；在微观上，主要通过分析不同人群的饮食偏好来反映饮食习惯与与各因素间的相关性。

宏观层面

在宏观层面，经分析相关性结果，并绘制不同职业、婚姻状况的三餐指标得分情况的箱线图，本文发现三餐习惯与年龄、文化程度有关，性别上也有一定的差异，验证了问题一三餐习惯在年龄、性别上存在差异的结果；选择用餐地点的习惯与职业、婚姻关系有着密不可分的关系；饮食习惯跟职业有一定的关系，其中工人、农民和其他职业会存在较大的差异，从图中也可看出工人、农民的得分偏低，这可能与工作安排有关；饮食习惯跟婚姻状况也有一定的关系，其中未婚和已婚与其他婚姻状况会存在较大差异。经学者研究发现，与单身和婚配时相比，在离婚或丧偶后饮食习惯会明显倒退，印证了本文结果。

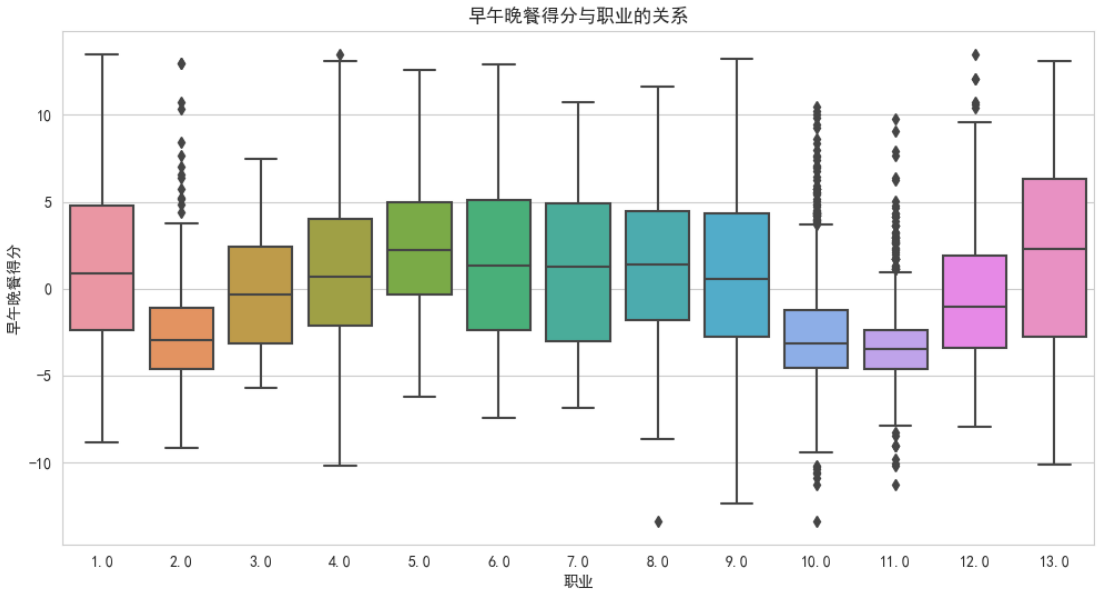

图 8　不同职业的三餐得分情况

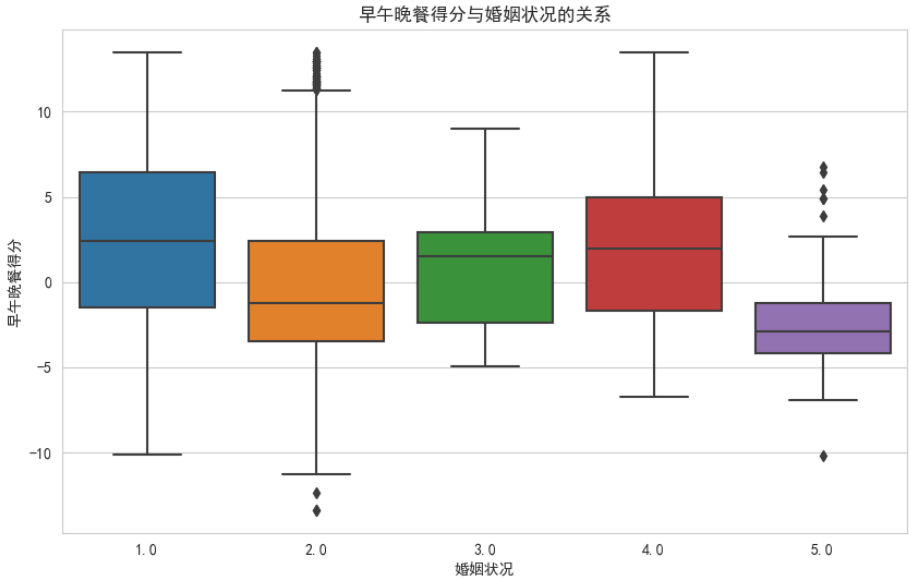

图 9　不同婚姻状况的三餐得分情况

微观层面

经统计不同年龄、性别、职业食用每种食物在各自人群中的占比情况，发现饮食习惯也存在性别、年龄差异，且更多的表现在饮食偏好差异上。以老年群体的饮食偏好为例，根据下图可以看出食用鲜奶酸奶的老人占比较小，食用老年人奶粉的老人占比增大。这可能是由于身体年龄原因不易对鲜奶酸奶进行吸收从而用奶粉代替。根据总体统计结果分析，男性更偏好于喝饮料解渴，女性更偏好吃糕点，肉类、奶类也同样发现有性别的偏好差异；随着年龄的增加，人们对油炸食品、饮料的摄取量减少。

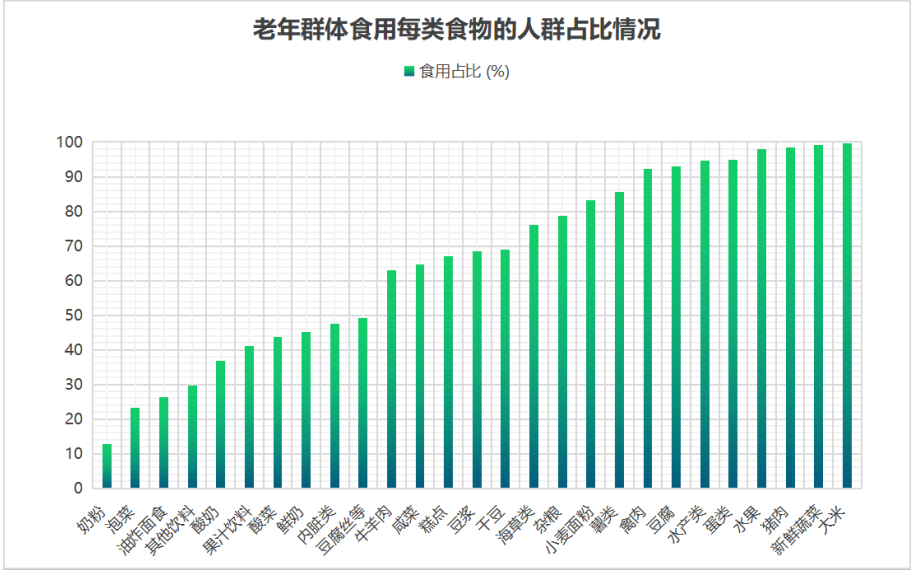

图 10　老年群体食用每类食物的人群占比情况

5.5.2 生活习惯与各因素间的相关性结果分析

通过统计分析，生活习惯与年龄、性别、文化程度、职业、婚姻状况都有着一些密不可分的关系。以性别来看，女性进行休闲、家务活动的强度往往比男性大；以年龄来看，对休闲、家务活动的强度会有一定的提升；以职业来看，家庭主妇往往进行强度较大的休闲、家务活动，强度在 2 及以上的占 **80.27%**，学生则相反，强度在 2 及以上的只占 **25.38%**，其他职业相对平稳，强度在 1 和 2 的占比相对平衡; 以文化程度来看，休闲、家务活动的强度与文化程度呈负相关，这或许与文化更高的人会进行更多的工作活动有关; 以婚姻状况来看，往往已婚的人休闲、家务活动的强度较高，而离婚、未婚、再婚的人强度较低。

6 问题三模型的建立与求解

6.1 问题三分析

问题三要求研究慢性疾病与吸烟、饮酒、饮食习惯、生活习惯、工作性质、运动等因素的关系以及相关程度。由于数据信息有限，本文仅研究糖尿病、高血压代谢综合征、低高密度脂蛋白血症四种慢性疾病。然后本文基于对数据和问题的分析，决定采用决策树算法[8]求解与各因素之间的关联性分析[6]，选取吸烟量、饮酒量、运动量、膳食质量评分、活动量、BMI、年龄、工作性质、性别、文化程度、是否被诊断过糖尿病或高血压作为特征变量，建立两棵决策树，树的结构可以直观反映各因素与两种疾病的关联性强弱。问题三思路图如下所示。

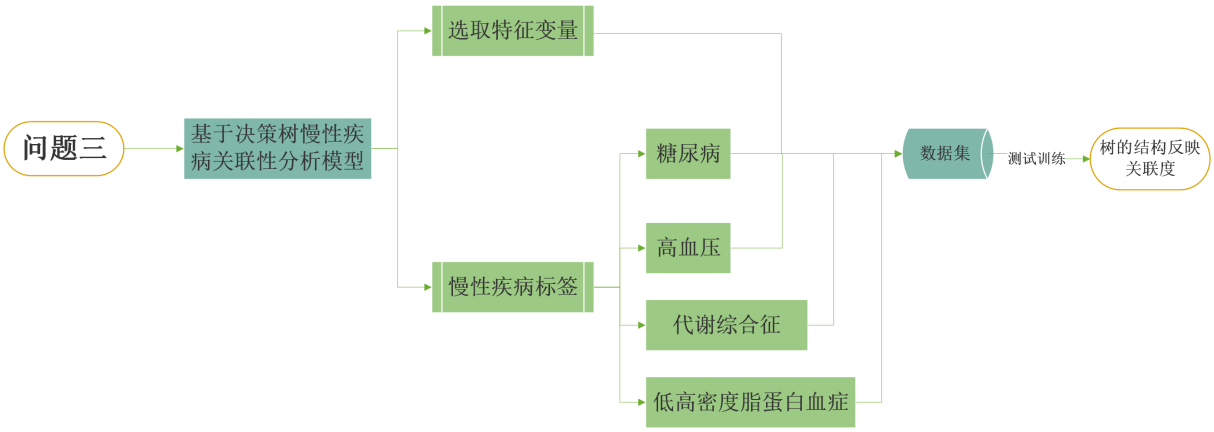

图 11　问题三思路图

6.2 数据预处理

除少部分缺失数据需采用与问题一相同的处理方法外，问题三涉及由于问卷问题设置造成的大量数据缺少的情况，数据缺失量占总数居的 80% 以上，但是这类数据主要

对吸烟、饮酒、工作性质、运动这四类因素的统计与计算有重要影响，不能直接删去，故采用赋 0 的方式进行标记，且经后续统计和计算后，也能与原有数据进行区分，不影响对这些因素重要性和关联性的评估。

6.3 基于决策树慢性疾病关联性分析模型

决策树是一种非参数的有监督方法，可以从一系列有特征和标签的数据中总结出规则，并用树形结构呈现。决策树具有可以生成理解的规则、不需要背景知识就可以进行分类、可从大量的数据中选取最具有识别能力的特征数据、可处理高维特征空间等优点。

本文根据附件数据，可以知道每个居民高血压和糖尿病的诊断记录，假设诊断数据真实可靠，因此存在标签数据，属于有监督问题。而且本题目的是分析慢性疾病与多个因素之间的相关性，这需要在高维特征空间中研究。并经过问题二研究，发现因素饮食生活习惯与其他因素之间还存在相关性，而决策树不会受变量共性影响。综上，本文决定用决策树算法来建立慢性疾病关联性分析模型。

6.3.1 慢性疾病的诊断与标签说明

本题要求研究慢性疾病与各因素之间的关联度，但由于附件二所给数据信息有限，本文选取糖尿病、代谢综合征、低高密度脂蛋白血症、高血压四种常见疾病进行研究。在求解过程中，对于高血压和糖尿病标签的设定，根据是否被社区或以上医院的医生诊断过患有高血压或糖尿病数据，将填写 1 的居民标记为患病，填写 2 的居民标记为不患病；对于低高密度脂蛋白血症、代谢综合征根据《中国成人血异常防治指南》的判定标准和据《中国 2 型糖尿病防治指南 (2013 年版) 》得到以下诊断标准，满足以下诊断标准标记为患病，否则为不患病。以上四种慢性疾病患病居民均用标签“1”表示，不患病居民均用标签“2”表示。

6.3.2 特征变量说明

综合考虑附件二数据信息与相关慢性疾病的研究结果，本文选取吸烟量、饮酒量、运动量、膳食质量评分、活动量、BMI、年龄、工作性质、性别、文化程度作为决策树的特征变量。部分变量的计算公式如下：

(1) 吸烟量 a_{1i}: 在该模型中本文仅考虑现在仍吸烟的居民，对过去吸、现在已戒烟的居民后面单独分析。由于被动吸烟占比很小且很难估计被动吸入量的多少，为了模型简洁，不考虑该部分数据。计算公式如下：

$$a_{1i} = (Y_i - B_{2i}) \times B_{3i} \times B_{4i} \tag{14}$$

表 8　低高密度脂蛋白血症和代谢综合征诊断标准

慢性疾病	评价标准
低高密度脂蛋白血症	低密度脂蛋白或高密度脂蛋白 $<$ 1. 04mmol /L
	男性腰围 ≥90 cm，女性腰围 ≥85 cm
	甘油三酯 ≥1. 7 mmol /L
代谢综合征	高密度脂蛋白胆固醇 $<$ 1. 04 mmol /L
	收缩压 ≥130 mm Hg 或舒张压 ≥85 mm Hg
	或已被诊断为高血压而接受治疗
	空腹血糖 ≥6. 1 mmol /L 或已被诊断为糖尿病

其中 a_{1i} 表示居民 i 的吸烟量；Y_i 表示的年龄，根据出生日期计算；B_{2i} 表示开始吸烟的年龄，B_{3i} 表示平均每周吸烟的天数,B_{4i} 表示每天吸烟的支数。

(2) 饮酒量 a_{2i}：根据附附件二饮酒情况数据计算，公式如下:

$$a_{2i} = C_{2i} \times Z_i \times L_i \tag{15}$$

其中 C_{2i} 表示居民 i 的饮酒年数，z_i 表示每周的饮用频次，L_i 表示平均每次饮入量。

(3) 活动量 a_{3i}：根据附附件二身体活动情况数据并结合一些外部查找数据计算，公式如下:

$$a_{3i} = 20 \times E_{1i} + 10 \times E_{2i} + E_{4i} \times E_{5i} \tag{16}$$

其中 a_{3i} 表示居民 i 的活动量,E_{1i} 表示工作强度，E_{2i} 表示休闲、家务活动强度，E_{4i} 表示体育锻炼强度，E_{2i} 表示平均每天体育锻练时间。

6.3.3 决策树算法求解

本文选用的决策树自根节点开始从上而下将特定空间佳划分为一系列子空间，每个树节点代表对某个特征的划分，每个叶子节点代表标签划分类别。为将附件二信息转化成一棵树，我们需要找到最佳节点和最佳的分枝方法，即寻找最低“不纯度”，本文用信息熵作为 Criterion 参数决定不纯度的计算方法。

$$Entropy\,(t) = -\sum_{i=0}^{c-1} p\left(\frac{i}{t}\right) log_2 p\left(\frac{i}{t}\right) \tag{17}$$

其中 t 代表给定的节点，i 代表标签的任意分类，代表标签分类 i 在节点 t 上所占的比例，c 表示标签数目。

决策树的主要步骤如下所示。

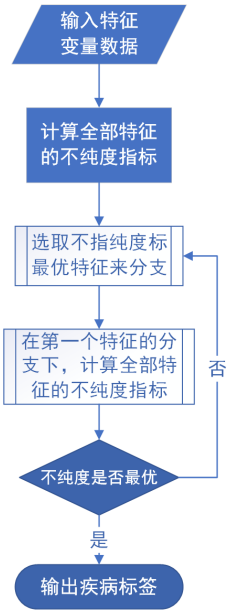

图 12　决策树求解流程

将数据集按三七比分为测试集和训练集，分别建立糖尿病和高血压的决策树，将吸烟量、饮酒量、运动量、膳食质量评分、活动量、BMI、年龄、工作性质、性别、文化程度、是否被诊断过糖尿病或高血压作为特征变量输入，经过不断迭代，最终决策树输出是否患糖尿病和高血压的标签，0 代表不患病，1 代表患病。通过决最终决策树结构，我们可以判断各特征变量与该类疾病的关联程度。

6.4 结果分析

6.4.1 常见慢性病与各因素间的相关性分析

本文分别从宏观和微观两个层面分析常见慢性病与各因素间的相关性。在宏观上，对人群的慢性病得病情况与其他因素之间的关系进行分析，并判断各因素对于慢性病的相关程度；在微观上，根据得病情况将人群分为对照组和病例组，并对各因素进行单因素分析，得到慢性病与各因素之间的具体关系。

宏观层面

在宏观层面，经决策树算法求解得到相关程度的分析结果，在问题研究中，我们考虑了糖尿病、高血压、代谢综合征、低高密度脂蛋白血症四种慢性病，以高血压为例，描述高血压与各因素之间相关程度的树状图如下所示。

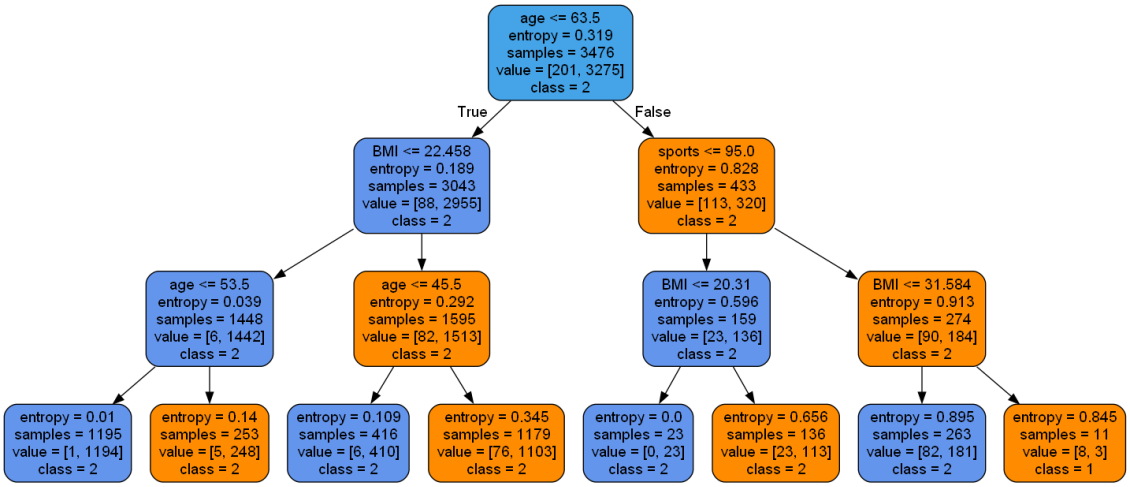

图 13　高血压重要特征树状图

四类慢性病与各因素间的相关程度表如下所示。决策树的平均准确率达到了**91.61%**，具有可信性。根据结果可分析，年龄、BMI、膳食质量评价与常见慢性病相关程度较大，往往年龄增大、BMI 的不合理，膳食质量评价降低，都会提高患慢性病的风险。

表 9　四类慢性病的特征相关程度分析表

指标\慢性病	糖尿病	高血压	代谢综合征	低高密度脂蛋白血症
年龄	0.284136	0.253394	0.105090	0.148619
性别	0.017936	0.012008	0.000000	0.059739
工作性质	0.011432	0.016008	0.021893	0.020138
文化程度	0.037717	0.056467	0.047075	0.061851
BMI	0.284136	0.258265	0.328685	0.280234
膳食质量评价	0.221419	0.247791	0.316000	0.242194
吸烟量	0.049686	0.038300	0.067905	0.044741
饮酒量	0.031517	0.023095	0.048425	0.040336
运动量	0.065634	0.094670	0.064930	0.102146

微观层面

本文在微观层面对调查结果较完整的 5000 名对象进行统计并不同群体，分别为对照组、糖尿病病例组、高血压病例组，再根据各因素进行单因素分析，统计各因素下的

成分占比，部分因素的统计结果如下表所示。

表 10　居民对照组与病例组的特征统计表（部分）

指标	类别	对照组（4608）	高血压病例组（293）	糖尿病病例组（95）
年龄阶段				
	青年	29.98%	3.07%	5.26%
	中年	55.34%	32.42%	36.84%
	老年	14.68%	64.51%	57.89%
吸烟情况				
	吸烟	20.56%	19.80%	29.47%
	已戒烟	4.15%	10.24%	6.32%
	不吸烟	75.30%	69.97%	64.21%
饮酒情况				
	饮酒	24.61%	25.60%	24.21%
	已戒酒	2.13%	13.65%	13.68%
	不饮酒	73.26%	60.75%	62.11%
锻炼情况				
	不参加	50.74%	40.96%	43.16%
	1-2/天每周	29.06%	24.91%	23.16%
	3-5/天每周	10.70%	12.29%	9.47%
	>5 天/周	9.51%	21.84%	24.21%

由表可以分析得出，性别、年龄、吸烟、饮酒、生活习惯、饮食习惯、工作性质与常见慢性病都有着密不可分的关系。随着年龄增长，慢性病的风险会增大，且得病率往往男性高于女性。经结果也可以发现，患有糖尿病、高血压的患者往往经受过诊断并根据医疗建议进行调整，例如吸烟、饮酒、锻炼情况的分析中，更多的患者为了健康会采取戒烟、戒酒的手段来进行控制，并且提高自己的锻炼频率，以保证自己健康的生活作息，这些都是因为慢性病造成的反馈结果。

7 问题四模型的建立与求解

7.1 问题四分析

问题四要求对附件二居民进行分类，并提出健康改善意见。本文首先用 K-Means 聚类算法，依据问题三得出的与慢性疾病关联程度最大的年龄、BMI、膳食质量评分三项指标作聚类，通过肘部法求出最佳聚类类别数为 3，从而将附件二居民分为三类。接着，本文基于问题三结果对患病和不患病居民进行健康情况改善分析。最后对每个类别人群健康情况进行分析，并提出合理建议。问题四思路图如下所示。

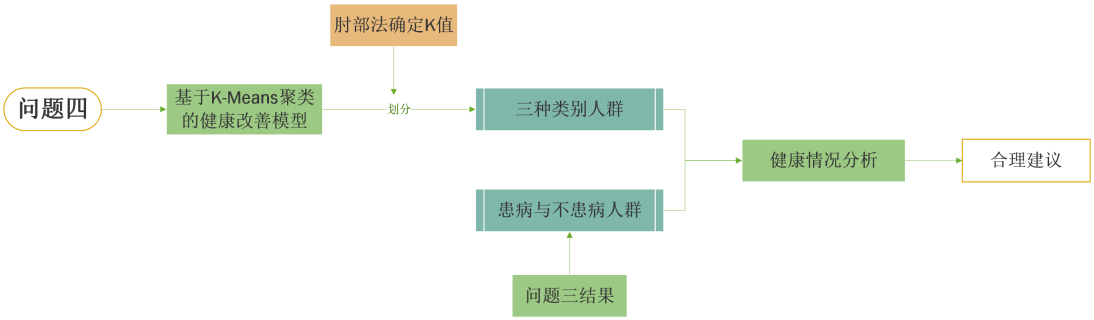

图 14　问题四思路图

7.2 基于 K—Means 聚类的健康改善模型

本问题首先需要对附件二居民进行合理分类。根据问题三结果，年龄、BMI、膳食质量评分与慢性疾病的关联程度最大，因此，本文将三者作为指标，采用 K—Means 聚类算法对附件二居民进行聚类。

7.2.1 K—Means 聚类算法求解

K—Means 聚类算法[9] [10]是一种无监督学习，同时也是基于划分的聚类算法，一般用欧式距离作为衡量数据对象间相似度的指标，相似度与数据对象间的距离成反比，相似度越大，距离越小该算法的主要作用是将相似的样本自动归到一个类别中，划分为若干个通常是不相交的子集，每个子集称为一个“簇（cluster）”，聚类既能作为一个单独过程，用于找寻数据内在的分布结构，也可作为分类等其他学习任务的前去过程，一般聚类结果较好。

该算法主要步骤如下图所示：

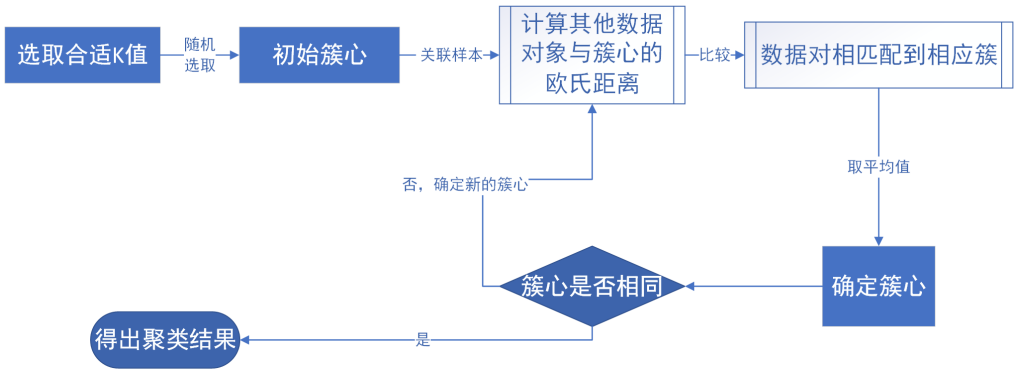

图 15　K-Means 聚类算法求解

在上述步骤中，定义聚类类别数即选取合适的 K 值是 K-Means 聚类算法的关键步骤。本文采用肘部法进行确定 K 值，K 值的最优解是以成本函数最小化为目标，成本函数为各个类畸变程度之和，每个类的畸变程度等于该类重心与其内部成员位置距离的平方和，在这个平方和随 K 值的变化过程中，会出现一个拐点也即“肘”点，下降率突然变小时即认为是最优的 K 值。

本文对附件二居民进行聚类，绘制肘部图如下图所示，其横坐标为聚类别数 K，纵坐标为畸变程度。

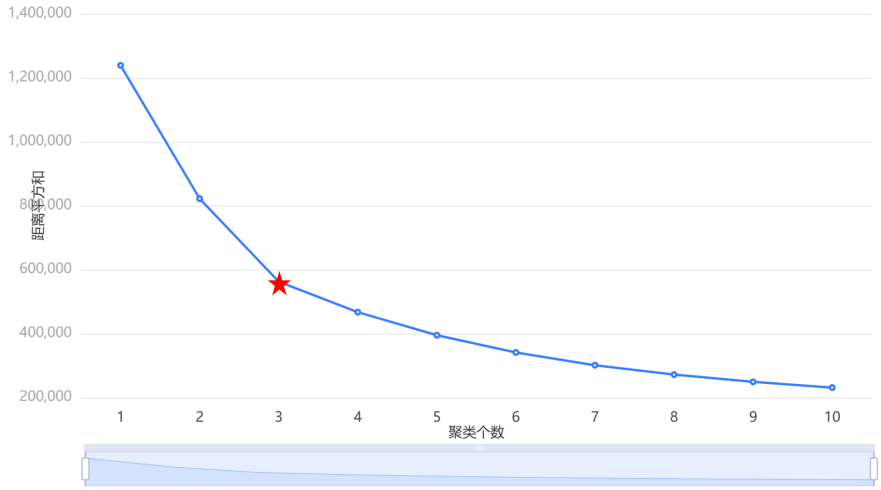

图 16　肘部图

从图中可以看出当类数从 1 增加到 3 时，总畸变程度下降较快；但当类别数超过 3 时，总畸变程度变化趋势变缓。对此，本文确定 K= 3 为最佳聚类数。

基于上述步骤，根据膳食质量评分、BMI、年龄三项指标，将附件二居民聚类成三个类别，得到每个类别具体情况和居民数占比如下。每个类别的 p 值均小于 1%, 显著水平较好，轮廓系数为 0.395，聚类效果良好。

表 11　聚类结果

指标	聚类类别（平均值 ± 标准差）		
	类别一 (n=2186)	类别二 (n=1724)	类别三 (n=1057)
膳食质量评分	18.8±6.3	38.3±6.8	24.3±8.7
BMI	22.8±3.5	22.8±3.4	24.3±3.4
年龄	46±7	48±7	66±8

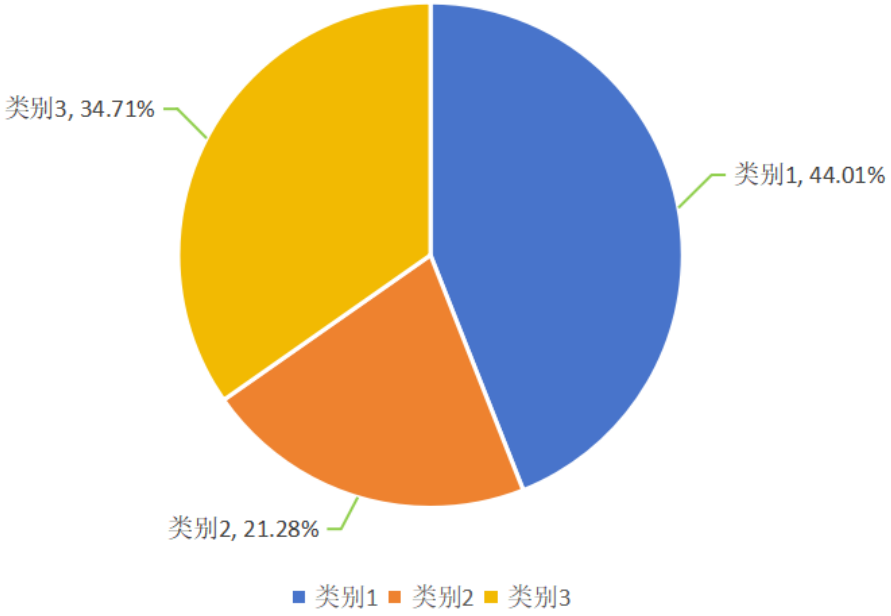

图 17　各聚类类别居民人数占比情况

7.3 结果分析

7.3.1 患病人群的健康情况改善分析

根据问题二的分析，我们也可以针对患病人群提出有利于身体健康的膳食、运动等方面的合理建议。

不患病居民

合理建议：加强身体锻炼，养成良好的饮食生活习惯，尽量少甚至不抽烟、饮酒，了解家庭病史，做好慢性病的预防工作。

患病居民

合理建议：加强身体锻炼，控制饮食，尽量少甚至不抽烟、饮酒，积极进行药物治疗，定期去社区或以上医院进行检查，观察自己的身体情况，保持健康的生活习惯。

7.3.2 各类别人群健康情况改善分析

从各类别三个指标情况上看，类别一群体的膳食评分最低，类别二群体的膳食质量评分最高，类别三的人群大部分年龄阶段处于老年阶段。根据类别将调查对象分成 3 个群体，再根据各因素进行单因素分析，统计各因素下成分占比，部分因素的统计结果如下所示。

表 12　居民分类组的特征统计表

指标	类别	1 类（696）	2 类（551）	3 类（376）
高血压				
	患	1.15%	1.63%	15.16%
	不患	98.85%	98.37%	84.84%
糖尿病				
	患	1.01%	0.54%	4.26%
	不患	98.99%	99.46%	95.74%

由表分析可以分析不同类别的人群各因素的状况，为此我们针对处在每个类别的群体提出有利于身体健康的膳食、运动等方面的合理建议。

类别一

情况分析：处于该类别的居民的膳食质量较低，BMI 较低，主要以青年、中年群体为主，文化程度往往较高，工作强度较大，属于**“年轻工作党”**。

合理建议：该类别的居民应加大对营养食物的摄取量，根据《国家居民膳食指南》进行控制，并在工作之余加强日常锻炼以提高身体素质，尽量少甚至不抽烟、饮酒，

类别二

情况分析：处于该类别的居民的膳食质量较高，BMI 大部分维持在正常水平，主要以青年、中年为主，且患慢性病的风险更低，属于**“年轻健康党”**。

合理建议：该类别的居民应继续保持自己的膳食质量，并进一步改进自己的饮食习惯做到更好，多锻炼运动，保持身材，尽量少甚至不抽烟、饮酒。

类别三

情况分析：处于该类别的居民膳食质量较低，BMI 较高，主要以中年、老年群体为主，文化程度往往较低，得慢性病的比例也比其他两类更高，属于**“老年风险党”**。

合理建议：该类别的居民要认清自己的位置，老年人应是重点关注慢性病预防的对象，须加强对饮食习惯、身体健康、慢性病预防相关知识的学习，控制自己的饮食，减少对油炸食品等限量食物的摄取，尽量少甚至不抽烟、饮酒。

8 模型总结与评价

8.1 模型总结

8.1.1 模型优点

1) 问题一使用基于营养曲线和 DBI 的膳食质量评价模型将多维数据有效整合在一起，全面衡量膳食质量，结果具备客观性及合理性的优点。

2) 问题二采用的主成分分析法，通过分析特征间相关性从而去除特征间冗余，将多维特征映射到一维空间，实现高效降维，简化数据分析复杂度，从而提高算法效率。

3) 问题三的基于决策树慢性疾病关联性分析模型对于噪声和离群点都具有较好的鲁棒性，且能够同时处理数值型数据和分类数据，结果的准确率较高且可解释性和可靠性强。

8.1.2 模型缺点

1) 决策树算法耗时不稳定，参数的设置会大大影响决策树的效果以及算法的耗时。

2) 决策树学习算法不能保证返回全局最优决策树，这个问题可以通过集成学习来训练多颗决策树来缓解。

8.2 模型改进

1) 由于原数据较小且存在患病人数远小于不患病人数造成的决策树模型产生偏差的问题，可以采用采样方法进行数据均衡化处理。

2) 为避免产生过于复杂的决策树，可以考虑模型剪枝算法，提高模型泛化能力和算法效率。

参考文献

[1] 侯琬娇, 杨子刚. 我国城市居民健康信息搜寻行为影响因素分析 [J]. 现代情报,2019.

[2] 陈树林, 黄鑫. 城市居民生活质量及其影响因素调查分析 [J]. 中国行为医学科学, 2001, 10(4):3.DOI:CNKI:SUN:ZGXX.0.2001-04-043.

[3] 陈辰. 中小城市慢性非传染病死亡病例时空分布模式分析及影响因子探究 [D]. 成都理工大学,2023.

[4] 罗森林, 陈燕颖, 潘丽敏, 等. 基于膳食平衡指数和营养曲线的个体膳食评价方法 [J]. 北京理工大学学报, 2013, 33(008):876-880.DOI:10.3969/j.issn.1001-0645.2013.08.021.

[5] 何宇纳, 王惠君, 房月晖, 等. 中国精简膳食质量评分 [J]. 卫生研究, 2021,50(2):7.DOI:10.19813/j.cnki.weishengyanjiu.2021.02.006.

[6] 张汇君. 考虑标记相关性的多标记决策树算法研究 [D]. 华北电力大学 (北京),2023.DOI:10.27140/d.cnki.ghbbu.2022.000427.

[7] Wang Y , Gao H , Jiang S ,et al.Principal component analysis of routine blood test results with Parkinson's disease: A case-control study[J].Experimental Gerontology, 2021,144(114):111188.DOI:10.1016/j.exger.2020.111188.

[8] Su Y , Shen J , Qian H ,et al.Diagnosis of gastric cancer using decision tree classification of mass spectral data[J].Cancer Science, 2010, 98(1):37-43.DOI:10.1111/j.1349-7006.2006.00339.x.

[9] Saroj,Kavita.Review: Study on Simple k-Mean and Modified K Mean Clustering Technique[J].[2023-08-09].

[10] Bari S , Love D J , Rispoli J V ,et al.Erratum to: Global and peak local specific absorption rate control on parallel transmit systems using k-means SAR compression model (Magn Reson Med. 2021;85:1093-1103)[J].Magnetic Resonance in Medicine, 2021, 86(6):3391-3396.

附录 A　部分居民的膳食质量评分

ID	膳食评价	ID	膳食评价	ID	膳食评价
10001	16.84456	10296	35.08724	10545	66.71795
10002	37.81012	10297	41.06008	10546	33.02417
10003	20.58826	10298	32.3984	10548	35.26368
10005	10.86914	10299	30.96429	10549	27.83016
10006	35.7078	10300	47.73549	10550	14.76878
10011	32.99376	10301	31.93726	10551	8.327846
10012	25.77535	10302	59.63542	10552	21.12303
10013	9.473995	10303	15.12881	10554	46.80866
10015	29.66317	10304	45.2545	10555	24.47216
10018	26.58589	10305	32.70751	10557	19.80671
10020	59.78475	10306	27.77613	10558	20.82069
10021	28.61328	10307	47.5135	10559	28.54228
10022	26.63226	10308	13.35112	10560	25.27971
10023	47.67003	10309	26.87136	10564	25.43246
10024	9.318078	10310	22.00344	10565	48.57886
10025	38.87368	10311	25.11356	10566	14.05307
10026	30.76563	10312	47.34043	10572	34.93315
10027	41.88245	10313	37.17158	10573	12.93564
10028	28.4084	10314	32.24034	10574	17.6025
10029	51.71217	10315	40.22371	10575	25.91099
10030	8.089163	10316	52.64639	10576	13.81736
10032	31.85689	10317	45.03943	10578	24.56643
10033	31.53133	10318	39.58193	10579	34.87043
10038	23.84061	10319	55.58923	10580	17.52791
10039	31.2866	10320	25.40963	10581	17.61765
10041	45.98057	10322	27.22571	10584	6.524649
10042	27.14188	10324	27.95118	10586	22.08953
10046	31.54601	10326	7.165883	10587	19.69985
10047	23.66771	10327	31.02153	10588	28.09665
10048	44.77752	10328	26.78871	10589	9.159146
10049	46.00494	10330	4.006351	10590	10.51958

10050	39.45598	10332	4.95594	10591	35.6736
10051	31.25168	10333	27.2387	10592	26.70416
10053	31.14096	10334	29.70651	10595	22.31745
10054	19.84946	10336	30.29813	10596	17.44591
10055	24.37218	10337	13.72411	10599	22.49939
10056	18.41842	10338	10.44986	10600	24.18662
10057	13.5251	10340	15.69679	10604	19.8281
10058	54.29468	10341	28.16303	10605	11.92317
10060	30.30542	10347	9.990199	10606	17.72445
10061	19.01576	10349	6.915084	10607	15.17103
10062	52.29223	10350	21.07721	10608	16.75686
10063	11.58378	10352	33.74088	10609	20.93195
10064	25.41493	10353	13.66167	10610	23.30631
10065	37.63549	10355	28.98917	10611	20.32894
10067	27.30802	10356	29.14495	10613	24.19384
10068	18.99336	10358	14.88596	10614	21.13314
10069	26.91918	10359	28.68857	10615	32.70966
10070	16.22003	10360	17.40964	10616	15.23726
10071	11.85638	10361	10.26332	10617	20.8713
10072	17.01609	10362	14.45405	10622	52.34734
10073	24.58062	10363	24.00129	10624	28.00019
10074	41.31011	10364	13.31997	10626	25.29704
10075	29.94298	10365	23.74421	10627	13.33945
10076	18.98868	10366	6.762586	10628	11.03239
10077	21.98902	10367	18.24022	10629	20.94083
10078	32.00013	10368	20.52492	10630	27.56512
10079	30.03288	10369	28.51357	10631	31.49377
10080	24.08655	10372	9.235679	10634	38.04308
10085	37.47685	10373	29.33584	10635	18.53088
10088	18.05362	10374	21.62295	10636	27.59952
10092	20.68948	10376	25.24109	10637	13.32299
10098	10.203	10377	32.77907	10638	25.03852
10099	23.62066	10379	20.98143	10641	13.2837
10103	26.18435	10380	30.11368	10642	25.5993

10104	13.83082	10382	25.07761	10644	9.519658
10107	28.64786	10386	29.93825	10646	23.81486
10108	15.69542	10387	40.32383	10647	27.98314
10111	27.38694	10388	28.12269	10648	18.56699
10114	39.4737	10390	24.1598	10649	18.67312
10118	24.12397	10392	34.34969	10650	17.97204
10121	30.54572	10393	28.4856	10651	22.68116
10122	26.77891	10395	13.88511	10652	12.29216
10123	35.31269	10400	8.320963	10653	29.25727
10124	11.44873	10401	16.63006	10655	4.348192
10126	33.29604	10402	34.94508	10656	12.61616
10127	35.11328	10403	33.17884	10657	33.2437
10128	12.09989	10404	17.25325	10658	21.97358
10131	21.07569	10405	19.98609	10659	38.00153
10135	29.10154	10407	36.55853	10662	21.19712
10136	22.96088	10408	23.31667	10665	22.86001
10137	34.15272	10409	24.14037	10667	28.68833
10138	46.60329	10410	22.45102	10668	22.07429
10139	18.54808	10411	25.42332	10670	45.24758
10141	21.97685	10412	32.16017	10671	32.84037
10142	17.05136	10413	11.1651	10672	22.31137
10145	33.7522	10415	16.84607	10673	32.20602
10146	41.02174	10416	36.08361	10674	6.142813
10147	7.650892	10417	10.84138	10677	29.09761
10148	26.05349	10419	45.95487	10680	24.88758
10150	52.08273	10420	22.01117	10687	13.02262
10154	38.35176	10421	35.59277	10695	36.95982
10155	17.17543	10422	12.07526	10698	38.46512
10156	28.52229	10424	32.1982	10701	46.66371
10161	33.79701	10425	18.51994	10702	35.91636
10162	17.45188	10426	17.6677	10708	33.90063
10163	14.00186	10427	20.87469	10714	40.50451
10164	27.61551	10428	9.642285	10731	18.52819
10165	26.92926	10429	20.43448	10736	22.04667

10169	23.1753	10430	33.28193	10739	20.29523
10171	14.04667	10433	15.71273	10748	36.85356
10173	31.15092	10434	10.57084	10754	44.51779
10175	26.84307	10435	20.42158	10759	29.48754
10176	17.21209	10436	27.38715	10760	35.50195
10177	7.492686	10438	44.55282	10762	33.76662
10182	13.34777	10439	18.94929	10768	11.13635
10188	9.839739	10440	35.58591	10769	12.01192
10190	29.07417	10442	5.560606	10771	19.21384
10199	37.54968	10443	16.22476	10773	38.32396
10201	12.49583	10444	27.03056	10777	29.39016
10202	22.40022	10445	10.80835	10780	18.98482
10203	8.720912	10446	22.19261	10781	25.32652
10204	36.86183	10447	36.9803	10793	21.91176
10205	26.29823	10449	36.22644	10795	36.16503
10206	36.23242	10450	26.06752	10798	50.09411
10207	27.24694	10451	27.34305	10800	11.91013
10208	35.29136	10453	39.15915	10806	36.04356
10209	14.67881	10454	30.96833	10809	27.10364
10210	13.74911	10455	13.83303	10811	40.90317
10211	11.70106	10456	7.818537	10814	20.40981
10212	17.98903	10457	23.48907	10815	41.13746
10214	32.47166	10458	30.92781	10816	20.61874
10215	15.84497	10459	41.29331	10818	19.56277
10216	10.51965	10460	17.334	10819	13.2821
10217	7.235665	10461	19.81283	10820	29.91259
10219	10.37873	10462	10.6431	10821	33.64484
10220	7.948976	10463	21.88763	10822	20.90195
10221	18.11531	10464	38.19875	10823	13.46728
10223	32.94216	10465	26.69029	10825	23.0797
10224	20.22332	10466	18.39093	10826	21.24167
10225	6.830448	10467	19.39472	10827	25.08694
10226	21.34256	10468	36.71247	10828	15.00708
10227	26.50709	10469	24.76313	10829	6.894357

10228	16.87008	10470	47.60496	10830	22.94079
10229	14.85287	10471	9.485058	10831	36.06314
10230	13.58904	10472	27.06491	10832	33.06125
10231	14.53362	10473	41.53212	10833	8.788928
10232	19.22697	10474	26.24397	10834	18.59506
10233	37.09775	10475	33.54673	10835	47.04757
10234	8.422846	10476	22.91403	10837	19.23383
10235	17.51826	10477	43.85841	10838	19.43948
10236	29.75942	10478	9.499204	10841	41.23795
10237	35.9292	10479	29.77129	10842	37.47446
10239	21.25198	10480	16.05417	10843	37.5841
10240	18.1932	10481	34.6073	10844	20.53554
10241	23.74275	10482	15.37748	10845	48.6602
10242	10.44775	10483	42.02638	10849	18.61123
10243	9.895539	10484	14.13984	10850	35.02148
10244	33.85996	10486	33.16447	10853	48.59809
10245	47.7864	10487	44.15901	10855	45.21858
10246	32.46094	10488	21.06557	10856	32.90457
10247	15.15835	10489	33.75295	10858	8.626917
10248	29.47433	10490	5.674148	10859	19.93994
10249	14.17198	10491	16.49426	10860	31.29476
10250	25.61541	10492	7.226695	10861	40.3494
10251	24.61082	10493	17.45937	10862	15.27322
10252	41.80942	10494	36.69391	10863	40.74877
10253	17.02081	10495	9.449436	10864	26.37829
10254	27.32263	10497	39.53454	10865	7.967028
10255	31.24743	10498	22.61006	10866	39.05583
10256	22.34316	10499	33.80602	10867	15.28365
10257	31.36425	10501	27.16025	10868	20.35631
10258	45.87074	10502	26.8956	10869	28.64565
10259	18.84442	10503	20.9475	10870	9.7564
10260	25.60266	10505	36.61826	10872	34.57469
10261	23.59511	10507	41.72089	10873	24.14644
10262	48.92909	10508	7.039592	10874	25.13302

10263	20.81869	10509	4.787566	10875	23.46877
10264	22.56391	10510	28.43462	10877	25.38044
10265	27.98506	10511	28.21569	10878	27.22327
10266	36.36973	10512	48.61368	10883	27.28691
10267	45.35293	10513	32.96386	10884	21.98765
10268	19.37356	10514	18.3267	10885	35.59807
10269	21.76761	10516	25.41078	10886	46.54106
10270	27.77104	10517	22.70629	10888	36.07184
10271	29.77044	10518	11.83688	10890	48.81172
10272	30.73457	10519	10.38268	10891	43.42874
10273	38.18336	10520	17.34907	10895	8.141889
10274	40.84421	10521	28.50013	10896	35.08114
10275	11.55512	10522	27.59821	10897	28.33474
10276	22.48754	10523	11.45502	10898	27.67789
10277	32.46508	10524	12.95083	10900	58.20766
10278	33.1114	10525	41.66873	10902	9.140829
10279	19.44289	10526	19.2174	10903	36.77016
10280	30.94055	10527	22.75255	10906	7.051395
10281	51.10177	10528	28.97143	10908	33.80964
10282	21.25885	10529	27.19665	10909	39.9016
10283	41.0727	10530	39.52349	10910	23.18526
10284	27.14742	10531	11.58275	10911	39.18445
10285	22.45709	10532	22.215	10912	27.28136
10286	14.90241	10533	29.96076	10913	15.35937
10287	26.34538	10534	34.86417	10917	35.96269
10288	22.83665	10535	40.3264	10918	21.37229
10289	51.40601	10536	31.79688	10922	10.85178
10290	29.74993	10537	22.06886	10923	19.64483
10291	27.24159	10540	30.01912	10927	16.96219
10292	39.91398	10541	13.78226	10928	20.02348
10293	41.6219	10542	19.01952	10929	20.57701
10294	36.38635	10543	25.28499	10934	26.87393
10295	35.6405	10544	28.62302	10939	29.52335

附录 B　居民对照组与病例组的特征统计表

指标	类别	对照组（4608）	高血压人群（293）	糖尿病人群（95）
性别				
	男	46.18%	54.61%	57.89%
	女	53.82%	45.39%	42.11%
年龄				
	青年	29.98%	3.07%	5.26%
	中年	55.34%	32.42%	36.84%
	老年	14.68%	64.51%	57.89%
吸烟情况				
	吸烟	20.56%	19.80%	29.47%
	已戒烟	4.15%	10.24%	6.32%
	不吸烟	75.30%	69.97%	64.21%
饮酒情况				
	饮酒	24.61%	25.60%	24.21%
	已戒酒	2.13%	13.65%	13.68%
	不饮酒	73.26%	60.75%	62.11%
休闲、家务活动				
	轻度	45.72%	48.46%	47.37%
	中度	53.58%	50.17%	51.58%
	重度	0.69%	1.37%	1.05%
锻炼情况				
	不参加	50.74%	40.96%	43.16%
	1-2/天每周	29.06%	24.91%	23.16%
	3-5/天每周	10.70%	12.29%	9.47%
	>5 天/周	9.51%	21.84%	24.21%
工作性质				
	轻度	76.30%	77.13%	80.00%
	中度	22.92%	22.18%	17.89%
	重度	0.78%	0.68%	2.11%
文化程度				
	文盲	0.85%	2.73%	2.11%
	小学	7.51%	20.48%	14.74%
	初中	29.75%	23.89%	30.53%

	高中/中专	33.42%	32.42%	28.42%
	大本/大专	27.73%	19.11%	23.16%
	研究生及以上	0.74%	1.37%	1.05%
大米				
	吃	99.89%	100.00%	100.00%
	不吃	0.11%	0.00%	0.00%
小麦				
	吃	82.38%	81.57%	82.11%
	不吃	17.62%	18.43%	17.89%
杂粮				
	吃	77.89%	73.72%	77.89%
	不吃	22.11%	26.28%	22.11%
薯类				
	吃	84.24%	81.91%	82.11%
	不吃	15.76%	18.09%	17.89%
油炸面食				
	吃	63.39%	66.21%	58.95%
	不吃	36.61%	33.79%	41.05%
猪肉				
	吃	98.78%	97.61%	98.95%
	不吃	1.22%	2.39%	1.05%
牛羊肉				
	吃	70.75%	60.07%	64.21%
	不吃	29.25%	39.93%	35.79%
禽肉				
	吃	94.73%	91.47%	93.68%
	不吃	5.27%	8.53%	6.32%
内脏				
	吃	58.81%	45.05%	54.74%
	不吃	41.19%	54.95%	45.26%
水产品				
	吃	95.20%	91.13%	87.37%
	不吃	4.80%	8.87%	12.63%
鲜奶				

	吃	55.06%	40.96%	47.37%
	不吃	44.94%	59.04%	52.63%
奶粉				
	吃	92.08%	91.47%	89.47%
	不吃	7.92%	8.53%	10.53%
酸奶				
	吃	52.63%	30.72%	30.53%
	不吃	47.37%	69.28%	69.47%
蛋类				
	吃	96.59%	92.15%	93.68%
	不吃	3.41%	7.85%	6.32%
豆腐				
	吃	92.80%	93.86%	92.63%
	不吃	7.20%	6.14%	7.37%
豆腐丝				
	吃	52.63%	53.92%	52.63%
	不吃	47.37%	46.08%	47.37%
豆浆				
	吃	76.15%	65.53%	55.79%
	不吃	23.85%	34.47%	44.21%
干豆				
	吃	65.63%	69.62%	71.58%
	不吃	34.38%	30.38%	28.42%
新鲜蔬菜				
	吃	99.33%	99.66%	97.89%
	不吃	0.67%	0.34%	2.11%
海草类				
	吃	78.15%	74.06%	69.47%
	不吃	21.85%	25.94%	30.53%
咸菜				
	吃	64.28%	66.55%	73.68%
	不吃	35.72%	33.45%	26.32%
泡菜				
	吃	75.13%	79.86%	70.53%

	不吃	24.87%	20.14%	29.47%
酸菜				
	吃	52.50%	59.39%	53.68%
	不吃	47.50%	40.61%	46.32%
糕点				
	吃	75.09%	62.80%	46.32%
	不吃	24.91%	37.20%	53.68%
新鲜水果				
	吃	98.50%	98.63%	97.89%
	不吃	1.50%	1.37%	2.11%
果汁饮料				
	吃	60.70%	42.32%	22.11%
	不吃	39.30%	57.68%	77.89%
其他饮料				
	吃	53.39%	29.35%	27.37%
	不吃	46.61%	70.65%	72.63%

附录 C　居民分类组的特征统计表

指标	类别	1 类 (696)	2 类 (551)	3 类 (376)
性别				
	男	47.99%	42.47%	48.14%
	女	52.01%	57.53%	51.86%
年龄				
		31.75%	27.22%	0.00%
	中年	68.25%	72.78%	23.94%
	老年	0.00%	0.00%	76.06%
吸烟情况				
	吸烟	22.84%	17.64%	25.53%
	已戒烟	3.59%	4.73%	7.45%
	不吸烟	73.56%	77.64%	67.02%
饮酒情况				
	饮酒	24.71%	25.05%	22.87%
	已戒酒	2.44%	1.27%	6.38%
	不饮酒	72.84%	73.68%	70.74%
休闲、家务活动				
	轻度	45.98%	44.83%	42.29%
	中度	53.74%	54.99%	57.45%
	重度	0.29%	0.18%	0.27%
锻炼情况				
	不参加	52.01%	50.64%	48.94%
	1-2/天每周	33.33%	32.85%	22.61%
	3-5/天每周	9.63%	10.53%	9.31%
	>5 天/周	5.03%	5.99%	19.15%
工作性质				
	轻度	76.87%	74.41%	74.73%
	中度	22.56%	25.23%	23.67%
	重度	0.57%	0.36%	1.60%
文化程度				
	文盲	0.57%	0.18%	2.66%
	小学	3.02%	3.27%	17.29%

	初中	27.01%	27.77%	34.84%
	高中/中专	33.91%	36.48%	27.66%
	大本/大专	34.34%	31.58%	16.49%
	研究生及以上	1.15%	0.73%	1.06%
高血压				
	患	1.15%	1.63%	15.16%
	不患	98.85%	98.37%	84.84%
糖尿病				
	患	1.01%	0.54%	4.26%
	不患	98.99%	99.46%	95.74%
大米				
	吃	100.00%	100.00%	99.73%
	不吃	0.00%	0.00%	0.27%
小麦				
	吃	84.48%	82.40%	79.52%
	不吃	15.52%	17.60%	20.48%
杂粮				
	吃	80.75%	78.77%	74.20%
	不吃	19.25%	21.23%	25.80%
薯类				
	吃	84.20%	86.57%	82.45%
	不吃	15.80%	13.43%	17.55%
油炸面食				
	吃	61.06%	62.61%	72.87%
	不吃	38.94%	37.39%	27.13%
猪肉				
	吃	98.13%	99.64%	99.73%
	不吃	1.87%	0.36%	0.27%
牛羊肉				
	吃	71.55%	73.68%	62.77%
	不吃	28.45%	26.32%	37.23%
禽肉				
	吃	94.68%	96.19%	91.49%
	不吃	5.32%	3.81%	8.51%

内脏				
	吃	60.20%	60.44%	51.86%
	不吃	39.80%	39.56%	48.14%
水产品				
	吃	95.11%	95.28%	94.68%
	不吃	4.89%	4.72%	5.32%
鲜奶				
	吃	53.74%	59.89%	43.88%
	不吃	46.26%	40.11%	56.12%
奶粉				
	吃	94.25%	93.28%	91.49%
	不吃	5.75%	6.72%	8.51%
酸奶				
	吃	57.90%	55.90%	32.18%
	不吃	42.10%	44.10%	67.82%
蛋类				
	吃	97.70%	95.83%	94.95%
	不吃	2.30%	4.17%	5.05%
豆腐				
	吃	92.96%	94.01%	94.15%
	不吃	7.04%	5.99%	5.85%
豆腐丝				
	吃	51.72%	51.54%	54.52%
	不吃	48.28%	48.46%	45.48%
豆浆				
	吃	80.17%	77.86%	67.82%
	不吃	19.83%	22.14%	32.18%
干豆				
	吃	66.67%	60.44%	69.68%
	不吃	33.33%	39.56%	30.32%
新鲜蔬菜				
	吃	99.43%	99.46%	99.47%
	不吃	0.57%	0.54%	0.53%
海草类				

	吃	78.16%	78.95%	70.21%
	不吃	21.84%	21.05%	29.79%
咸菜				
	吃	64.51%	60.44%	67.02%
	不吃	35.49%	39.56%	32.98%
泡菜				
	吃	74.57%	75.14%	80.32%
	不吃	25.43%	24.86%	19.68%
酸菜				
	吃	50.14%	49.73%	43.88%
	不吃	49.86%	50.27%	56.12%
糕点				
	吃	74.43%	77.13%	64.89%
	不吃	25.57%	22.87%	35.11%
新鲜水果				
	吃	98.56%	99.64%	97.34%
	不吃	1.44%	0.36%	2.66%
果汁饮料				
	吃	66.09%	60.44%	42.29%
	不吃	33.91%	39.56%	57.71%
其他饮料				
	吃	59.48%	57.53%	35.37%
	不吃	40.52%	42.47%	64.63%

附录 D　问题一源代码

4.1　数据预处理代码

```
#载入训练集和测试集
data = pd.read_csv("E:/Desktop/data/eating.csv",encoding='gbk')
data.describe()

#数据量化与预处理
data["食用大米的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用大米的频率周"].fillna(0, inplace=True)
data["食用大米的频率月"].fillna(0, inplace=True)
```

```
data["平均每次食用量大米"].fillna(data["平均每次食用量大米"].mean(), inplace=True)
data["大米摄入量"] = (data["食用大米的频率天"]*365 + data["食用大米的频率周"]*52 +
    data["食用大米的频率月"]*12)*data["平均每次食用量大米"]/365
data["大米摄入量"].describe()

data["食用杂粮的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用杂粮的频率周"].fillna(0, inplace=True)
data["食用杂粮的频率月"].fillna(0, inplace=True)
data["平均每次食用量食用杂粮"].fillna(data["平均每次食用量食用杂粮"].mean(), inplace=True)
data["食用杂粮摄入量"] = (data["食用杂粮的频率天"]*365 + data["食用杂粮的频率周"]*52 +
    data["食用杂粮的频率月"]*12)*data["平均每次食用量食用杂粮"]/365
data["食用杂粮摄入量"] .describe()

data["食用薯类的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用薯类的频率周"].fillna(0, inplace=True)
data["食用薯类的频率月"].fillna(0, inplace=True)
data["平均每次食用量食用薯类"].fillna(data["平均每次食用量食用薯类"].mean(), inplace=True)
data["食用薯类摄入量"] = (data["食用薯类的频率天"]*365 + data["食用薯类的频率周"]*52 +
    data["食用薯类的频率月"]*12)*data["平均每次食用量食用薯类"]/365
data["食用薯类摄入量"] .describe()

data["食用油炸面食的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用油炸面食的频率周"].fillna(0, inplace=True)
data["食用油炸面食的频率月"].fillna(0, inplace=True)
data["平均每次食用量油炸面食"].fillna(data["平均每次食用量油炸面食"].mean(), inplace=True)
data["食用油炸面食摄入量"] = (data["食用油炸面食的频率天"]*365 + data["食用油炸面食的频率周"]*52
    + data["食用油炸面食的频率月"]*12)*data["平均每次食用量油炸面食"]/365
data["食用油炸面食摄入量"] .describe()

data["食用油炸面食的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用油炸面食的频率周"].fillna(0, inplace=True)
data["食用油炸面食的频率月"].fillna(0, inplace=True)
data["平均每次食用量油炸面食"].fillna(data["平均每次食用量油炸面食"].mean(), inplace=True)
data["食用油炸面食摄入量"] = (data["食用油炸面食的频率天"]*365 + data["食用油炸面食的频率周"]*52
    + data["食用油炸面食的频率月"]*12)*data["平均每次食用量油炸面食"]/365
data["食用油炸面食摄入量"] .describe()

data["食用猪肉的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用猪肉的频率周"].fillna(0, inplace=True)
data["食用猪肉的频率月"].fillna(0, inplace=True)
data["平均每次食用量猪肉"].fillna(data["平均每次食用量猪肉"].mean(), inplace=True)
data["食用猪肉的摄入量"] = (data["食用猪肉的频率天"]*365 + data["食用猪肉的频率周"]*52 +
    data["食用猪肉的频率月"]*12)*data["平均每次食用量猪肉"]/365
data["食用猪肉的摄入量"] .describe()

data["食用牛羊肉的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用牛羊肉的频率周"].fillna(0, inplace=True)
```

```
data["食用牛羊肉的频率月"].fillna(0, inplace=True)
data["平均每次食用量牛羊肉"].fillna(data["平均每次食用量牛羊肉"].mean(), inplace=True)
data["食用牛羊肉的摄入量"] = (data["食用牛羊肉的频率天"]*365 + data["食用牛羊肉的频率周"]*52 +
    data["食用牛羊肉的频率月"]*12)*data["平均每次食用量牛羊肉"]/365
data["食用牛羊肉的摄入量"] .describe()

data["食用禽肉的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用禽肉的频率周"].fillna(0, inplace=True)
data["食用禽肉的频率月"].fillna(0, inplace=True)
data["平均每次食用量禽肉"].fillna(data["平均每次食用量禽肉"].mean(), inplace=True)
data["食用禽肉的摄入量"] = (data["食用禽肉的频率天"]*365 + data["食用禽肉的频率周"]*52 +
    data["食用禽肉的频率月"]*12)*data["平均每次食用量禽肉"]/365
data["食用禽肉的摄入量"] .describe()

data["食用内脏的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用内脏的频率周"].fillna(0, inplace=True)
data["食用内脏的频率月"].fillna(0, inplace=True)
data["平均每次食用量内脏类"].fillna(data["平均每次食用量内脏类"].mean(), inplace=True)
data["食用内脏的摄入量"] = (data["食用内脏的频率天"]*365 + data["食用内脏的频率周"]*52 +
    data["食用内脏的频率月"]*12)*data["平均每次食用量内脏类"]/365
data["食用内脏的摄入量"] .describe()

data["食用水产的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用水产的频率周"].fillna(0, inplace=True)
data["食用水产的频率月"].fillna(0, inplace=True)
data["平均每次食用量水产类"].fillna(data["平均每次食用量水产类"].mean(), inplace=True)
data["食用水产的摄入量"] = (data["食用水产的频率天"]*365 + data["食用水产的频率周"]*52 +
    data["食用水产的频率月"]*12)*data["平均每次食用量水产类"]/365
data["食用水产的摄入量"] .describe()

data["食用鲜奶的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用鲜奶的频率周"].fillna(0, inplace=True)
data["食用鲜奶的频率月"].fillna(0, inplace=True)
data["平均每次食用量鲜奶"].fillna(data["平均每次食用量鲜奶"].mean(),inplace=True)
data["食用鲜奶的摄入量"] = (data["食用鲜奶的频率天"]*365 + data["食用鲜奶的频率周"]*52 +
    data["食用鲜奶的频率月"]*12)*data["平均每次食用量鲜奶"]/365
data["食用鲜奶的摄入量"] .describe()

data["食用奶粉的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用奶粉的频率周"].fillna(0, inplace=True)
data["食用奶粉的频率月"].fillna(0, inplace=True)
data["平均每次食用量奶粉"].fillna(data["平均每次食用量奶粉"].mean(), inplace=True)
data["食用奶粉的摄入量"] = (data["食用奶粉的频率天"]*365 + data["食用奶粉的频率周"]*52 +
    data["食用奶粉的频率月"]*12)*data["平均每次食用量奶粉"]/365
data["食用奶粉的摄入量"] .describe()

data["食用酸奶的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
```

```
data["食用酸奶的频率周"].fillna(0, inplace=True)
data["食用酸奶的频率月"].fillna(0, inplace=True)
data["平均每次食用量酸奶"].fillna(data["平均每次食用量酸奶"].mean(), inplace=True)
data["食用酸奶的摄入量"] = (data["食用酸奶的频率天"]*365 + data["食用酸奶的频率周"]*52 +
    data["食用酸奶的频率月"]*12)*data["平均每次食用量酸奶"]/365
data["食用酸奶的摄入量"] .describe()

data["食用蛋类的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用蛋类的频率周"].fillna(0, inplace=True)
data["食用蛋类的频率月"].fillna(0, inplace=True)
data["平均每次食用量蛋类"].fillna(data["平均每次食用量蛋类"].mean(), inplace=True)
data["食用蛋类的摄入量"] = (data["食用蛋类的频率天"]*365 + data["食用蛋类的频率周"]*52 +
    data["食用蛋类的频率月"]*12)*data["平均每次食用量蛋类"]/365
data["食用蛋类的摄入量"] .describe()

data["食用豆腐的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用豆腐的频率周"].fillna(0, inplace=True)
data["食用豆腐的频率月"].fillna(0, inplace=True)
data["平均每次食用量豆腐"].fillna(data["平均每次食用量豆腐"].mean(), inplace=True)
data["食用豆腐的摄入量"] = (data["食用豆腐的频率天"]*365 + data["食用豆腐的频率周"]*52 +
    data["食用豆腐的频率月"]*12)*data["平均每次食用量豆腐"]/365
data["食用豆腐的摄入量"] .describe()

data["食用豆腐丝等的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用豆腐丝等的频率周"].fillna(0, inplace=True)
data["食用豆腐丝等的频率月"].fillna(0, inplace=True)
data["平均每次食用量豆腐丝等"].fillna(data["平均每次食用量豆腐丝等"].mean(), inplace=True)
data["食用豆腐丝等的摄入量"] = (data["食用豆腐丝等的频率天"]*365 +
    data["食用豆腐丝等的频率周"]*52 +
    data["食用豆腐丝等的频率月"]*12)*data["平均每次食用量豆腐丝等"]/365
data["食用豆腐丝等的摄入量"] .describe()

data["食用豆浆的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用豆浆的频率周"].fillna(0, inplace=True)
data["食用豆浆的频率月"].fillna(0, inplace=True)
data["平均每次食用量豆浆"].fillna(data["平均每次食用量豆浆"].mean(), inplace=True)
data["食用豆浆的摄入量"] = (data["食用豆浆的频率天"]*365 + data["食用豆浆的频率周"]*52 +
    data["食用豆浆的频率月"]*12)*data["平均每次食用量豆浆"]/365
data["食用豆浆的摄入量"] .describe()

data["食用干豆的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用干豆的频率周"].fillna(0, inplace=True)
data["食用干豆的频率月"].fillna(0, inplace=True)
data["平均每次食用量干豆"].fillna(data["平均每次食用量干豆"].mean(), inplace=True)
data["食用干豆的摄入量"] = (data["食用干豆的频率天"]*365 + data["食用干豆的频率周"]*52 +
    data["食用干豆的频率月"]*12)*data["平均每次食用量干豆"]/365
data["食用干豆的摄入量"] .describe()
```

```
data["食用海草的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用海草的频率周"].fillna(0, inplace=True)
data["食用海草的频率月"].fillna(0, inplace=True)
data["平均每次食用量海草类"].fillna(data["平均每次食用量海草类"].mean(), inplace=True)
data["食用海草的摄入量"] = (data["食用海草的频率天"]*365 + data["食用海草的频率周"]*52 +
    data["食用海草的频率月"]*12)*data["平均每次食用量海草类"]/365
data["食用海草的摄入量"] .describe()

data["食用新鲜蔬菜的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用新鲜蔬菜的频率周"].fillna(0, inplace=True)
data["食用新鲜蔬菜的频率月"].fillna(0, inplace=True)
data["平均每次食用量新鲜蔬菜"].fillna(data["平均每次食用量新鲜蔬菜"].mean(), inplace=True)
data["食用新鲜蔬菜的摄入量"] = (data["食用新鲜蔬菜的频率天"]*365 +
    data["食用新鲜蔬菜的频率周"]*52 +
    data["食用新鲜蔬菜的频率月"]*12)*data["平均每次食用量新鲜蔬菜"]/365
data["食用新鲜蔬菜的摄入量"] .describe()

data["食用咸菜的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用咸菜的频率周"].fillna(0, inplace=True)
data["食用咸菜的频率月"].fillna(0, inplace=True)
data["平均每次食用量咸菜"].fillna(data["平均每次食用量咸菜"].mean(), inplace=True)
data["食用咸菜的摄入量"] = (data["食用咸菜的频率天"]*365 + data["食用咸菜的频率周"]*52 +
    data["食用咸菜的频率月"]*12)*data["平均每次食用量咸菜"]/365
data["食用咸菜的摄入量"] .describe()

data["食用泡菜的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用泡菜的频率周"].fillna(0, inplace=True)
data["食用泡菜的频率月"].fillna(0, inplace=True)
data["平均每次食用量泡菜"].fillna(data["平均每次食用量泡菜"].mean(), inplace=True)
data["食用泡菜的摄入量"] = (data["食用泡菜的频率天"]*365 + data["食用泡菜的频率周"]*52 +
    data["食用泡菜的频率月"]*12)*data["平均每次食用量泡菜"]/365
data["食用泡菜的摄入量"] .describe()

data["食用酸菜的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用酸菜的频率周"].fillna(0, inplace=True)
data["食用酸菜的频率月"].fillna(0, inplace=True)
data["平均每次食用量酸菜"].fillna(data["平均每次食用量酸菜"].mean(), inplace=True)
data["食用酸菜的摄入量"] = (data["食用酸菜的频率天"]*365 + data["食用酸菜的频率周"]*52 +
    data["食用酸菜的频率月"]*12)*data["平均每次食用量酸菜"]/365
data["食用酸菜的摄入量"] .describe()

data["食用糕点的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用糕点的频率周"].fillna(0, inplace=True)
data["食用糕点的频率月"].fillna(0, inplace=True)
data["平均每次食用量糕点"].fillna(data["平均每次食用量糕点"].mean(), inplace=True)
data["食用糕点的摄入量"] = (data["食用糕点的频率天"]*365 + data["食用糕点的频率周"]*52 +
```

```
    data["食用糕点的频率月"]*12)*data["平均每次食用量糕点"]/365
data["食用糕点的摄入量"] .describe()

data["食用水果的频率天"].fillna(0, inplace=True) #缺失值为空值用0填充
data["食用水果的频率周"].fillna(0, inplace=True)
data["食用水果的频率月"].fillna(0, inplace=True)
data["平均每次食用量水果"].fillna(data["平均每次食用量水果"].mean(), inplace=True)
data["食用水果的摄入量"] = (data["食用水果的频率天"]*365 + data["食用水果的频率周"]*52 +
    data["食用水果的频率月"]*12)*data["平均每次食用量水果"]/365
data["食用水果的摄入量"] .describe()

data.to_csv("E:/Desktop/data/摄入量（不含食用油调味品）.csv",encoding="gbk")
data1 = pd.read_csv("E:/Desktop/data/eating.csv",encoding='gbk')

# 假设数据是一个二维数组，每个子数组表示一行数据
data1["工作日在家吃中餐人数"].fillna(0, inplace=True)
data1["周末在家吃中餐人数"].fillna(0, inplace=True)
data1["工作日在家吃早餐人数"].fillna(0, inplace=True)
data1["周末在家吃早餐人数"].fillna(0, inplace=True)
data1["工作日在家吃晚餐人数"].fillna(0, inplace=True)
data1["周末在家吃晚餐人数"].fillna(0, inplace=True)


data_member = data1[["工作日在家吃中餐人数",
                     "周末在家吃中餐人数",
                     "工作日在家吃早餐人数",
                     "周末在家吃早餐人数",
                     "工作日在家吃晚餐人数",
                     "周末在家吃晚餐人数"]]


data_member

data_member['member']= data_member.max(axis=1) #取出该最大值
data_member['member'].fillna(1, inplace=True)

data_member['member'].describe()

data["植物油摄入量"]=data["植物油"]/(data_member['member']*30)
# data["植物油"].describe()
data["动物油摄入量"]=data["动物油"]/(data_member['member']*30)
data["盐摄入量"]=data["盐"]/(data_member['member']*30)
data["酱油摄入量"]=data["酱油"]/(data_member['member']*30)
data["醋摄入量"]=data["醋"]/(data_member['member']*30)
data["酱类摄入量"]=data["酱类"]/(data_member['member']*30)
data["味精摄入量"]=data["味精"]/(data_member['member']*30)
data["member"] = data_member['member']
```

```
data.to_csv("E:/Desktop/data/摄入量.csv",encoding="gbk")
```

4.2 绘制营养曲线代码

```
    #画健康程度随食物摄入量的变化曲线
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号

# Define the nutritional curve model
def nutritional_curve(x, k1=0.1, k2=0.1):
    a = 200
    b = 350
    return 1 / (1 + np.exp(-k1 * (x - a))) - 1 / (1 + np.exp(-k2 * (x - b)))

# Generate x values
x_values = np.linspace(0, 500, 1000)

# Compute y values
y_values = nutritional_curve(x_values)

# Define the regions
regions = {
    "不足(0-200g)": {"color": "red", "range": (0, 200)},
    "安全区(200-250g)": {"color": "yellow", "range": (200, 250)},
    "最佳区 (250-300g)": {"color": "green", "range": (250, 300)},
    "安全区(300-350g)": {"color": "yellow", "range": (300, 350)},
    "过量区 (350-500g)": {"color": "red", "range": (350, 500)}
}



# Plot the curve with regions
plt.figure(figsize=(12, 7))
plt.plot(x_values, y_values, label="营养曲线", color="blue", linewidth=2.5)

# Highlight the regions
for region, props in regions.items():
    plt.axvspan(props["range"][0], props["range"][1], color=props["color"], alpha=0.5,
        label=region)


ax = plt.subplot(111)                # 设置刻度字体大小
plt.xticks(fontsize=15)
```

```
plt.yticks(fontsize=15)              # 设置坐标标签字体大小

ax.set_xlabel(..., fontsize=15)
ax.set_ylabel(..., fontsize=15)

plt.xlabel("每日新鲜水果摄入量（g）")
plt.ylabel("健康程度")
plt.suptitle('健康程度随新鲜水果摄入量的变化曲线', fontsize=20)
# plt.title("新鲜水果摄入量营养曲线")
plt.legend(loc="upper left")
plt.grid(True)
plt.tight_layout()

# Save the enhanced plot to a file
enhanced_file_path = "E:/Desktop/nutrition_curve/新鲜水果摄入量营养曲线.png"
plt.savefig(enhanced_file_path)
plt.show()
```

附录 E　问题二源代码

5.1　相关性分析代码

```
    import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# # 设置字体和正常显示负号
# plt.rcParams["font.sans-serif"] = ["SimHei"]
# plt.rcParams["axes.unicode_minus"] = False

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")
merged_data_with_diet

# 将年龄划分为不同的类别
def categorize_age(age):
    if 18 <= age <= 44:
        return '青年'
    elif 45 <= age <= 59:
        return '中年'
    else:
        return '老年'

merged_data_with_diet['年龄段'] = merged_data_with_diet['年龄'].apply(categorize_age)

# data_class={'青年':0,'中年':1,'老年':2}
```

```
# merged_data_with_diet['年龄段']=merged_data_with_diet['年龄段'].map(data_class)
data_age_1 = merged_data_with_diet[['年龄段','平均每天体育锻炼时间']]
data_age_1
data_age_1.groupby('年龄段').agg('mean')
sns.barplot(data=data_age_1,x="年龄段",y="平均每天体育锻炼时间",ci=None,palette="Blues_d")
plt.title("年龄段与每天体育锻炼时间")
plt.xlabel("年龄段")
plt.ylabel("每天体育锻炼时间")
plt.show()

from sklearn import metrics

data_age_1 = data_age_1.dropna(axis=0,how='any')
label=data_age_1['年龄段']
sample=data_age_1['平均每天体育锻炼时间']
print(metrics.mutual_info_score(label,sample))
data_age_1 = merged_data_with_diet[['年龄段','体育锻炼的强度总']]
data_age_1.groupby('年龄段').agg('mean')
sns.barplot(data=data_age_1,x="年龄段",y="体育锻炼的强度总",ci=None,palette="Blues")
plt.title("年龄段与体育锻炼的强度总")
plt.xlabel("年龄段")
plt.ylabel("体育锻炼的强度总")

enhanced_file_path = "E:/Desktop/nutrition_curve/年龄段与体育锻炼的强度总.png"
plt.savefig(enhanced_file_path)
plt.show()

data_age_1 = data_age_1.dropna(axis=0,how='any')

label=data_age_1['年龄段']
sample=data_age_1['体育锻炼的强度总']
print(metrics.mutual_info_score(label,sample))
job_titles = {1: '工人', 2: '农民', 3: '军人', 4: '行政干部', 5: '科技人员', 6: '医务人员', 7:
    '教师', 8: '金融财务', 9: '商业服务人员', 10: '家庭妇女', 11: '离、退休人员', 12: '待业', 13:
    '学生'}
merged_data_with_diet['职业']=merged_data_with_diet['职业'].replace(job_titles)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['职业','平均每天体育锻炼时间']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('职业').agg('mean')

plt.figure(figsize=(12, 8))
sns.barplot(data=data_job_1,x="职业",y="平均每天体育锻炼时间",ci=None,palette="Blues")
plt.title("职业与平均每天体育锻炼时间相关性")
```

```
plt.xlabel("职业")
plt.ylabel("平均每天体育锻炼时间")

enhanced_file_path = "E:/Desktop/nutrition_curve/职业与平均每天体育锻炼时间相关性.png"
plt.savefig(enhanced_file_path)
plt.show()

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")

job_titles = {1: '工人', 2: '农民', 3: '军人', 4: '行政干部', 5: '科技人员', 6: '医务人员', 7:
    '教师', 8: '金融财务', 9: '商业服务人员', 10: '家庭妇女', 11: '离、退休人员', 12: '待业', 13:
    '学生'}
merged_data_with_diet['职业']=merged_data_with_diet['职业'].replace(job_titles)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['职业','体育锻炼的强度总']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('职业').agg('mean')

plt.figure(figsize=(12, 8))
sns.barplot(data=data_job_1,x="职业",y="体育锻炼的强度总",ci=None,palette="Blues")
plt.title("职业与体育锻炼的强度总相关性")
plt.xlabel("职业")
plt.ylabel("体育锻炼的强度总")

enhanced_file_path = "E:/Desktop/nutrition_curve/职业与体育锻炼的强度总相关性.png"
plt.savefig(enhanced_file_path)
plt.show()

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")

sex = {1: '男', 2: '女'}
merged_data_with_diet['性别']=merged_data_with_diet['性别'].replace(sex)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['性别','体育锻炼的强度总']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('性别').agg('mean')

plt.figure(figsize=(4, 4))
sns.barplot(data=data_job_1,x="性别",y="体育锻炼的强度总",ci=None,palette="Blues")
plt.title("性别与体育锻炼的强度总相关性")
plt.xlabel("性别")
```

```
plt.ylabel("体育锻炼的强度总")

enhanced_file_path = "E:/Desktop/nutrition_curve/性别与体育锻炼的强度总相关性.png"
plt.savefig(enhanced_file_path)
plt.show()

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")

sex = {1: '男', 2: '女'}
merged_data_with_diet['性别']=merged_data_with_diet['性别'].replace(sex)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['性别','平均每天体育锻炼时间']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('性别').agg('mean')

plt.figure(figsize=(4, 4))
sns.barplot(data=data_job_1,x="性别",y="平均每天体育锻炼时间",ci=None,palette="Blues")
plt.title("性别与平均每天体育锻炼时间相关性")
plt.xlabel("性别")
plt.ylabel("平均每天体育锻炼时间")

enhanced_file_path = "E:/Desktop/nutrition_curve/性别与平均每天体育锻炼时间相关性.png"
plt.savefig(enhanced_file_path)
plt.show()

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")
marriage = {1: '未婚', 2: '已婚', 3: '再婚', 4: '离婚', 5: '丧偶'}
merged_data_with_diet['婚姻状况']=merged_data_with_diet['婚姻状况'].replace(marriage)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['婚姻状况','平均每天体育锻炼时间']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('婚姻状况').agg('mean')

plt.figure(figsize=(8, 4))
sns.barplot(data=data_job_1,x="婚姻状况",y="平均每天体育锻炼时间",ci=None,palette="Blues")
plt.title("婚姻状况与平均每天体育锻炼时间相关性")
plt.xlabel("婚姻状况")
plt.ylabel("平均每天体育锻炼时间")

enhanced_file_path = "E:/Desktop/nutrition_curve/婚姻状况与平均每天体育锻炼时间相关性.png"
plt.savefig(enhanced_file_path)
```

```
plt.show()

merged_data_with_diet = pd.read_csv("E:/Desktop/data/003.csv",encoding="gbk")
marriage = {1: '未婚', 2: '已婚', 3: '再婚', 4: '离婚', 5: '丧偶'}
merged_data_with_diet['婚姻状况']=merged_data_with_diet['婚姻状况'].replace(marriage)
# merged_data_with_diet['职业']


data_job_1 = merged_data_with_diet[['婚姻状况','体育锻炼的强度总']]
# data_job_1
data_job_1 = data_job_1.dropna(axis=0,how='any')
data_job_1.groupby('婚姻状况').agg('mean')

plt.figure(figsize=(8, 4))
sns.barplot(data=data_job_1,x="婚姻状况",y="体育锻炼的强度总",ci=None,palette="Blues")
plt.title("婚姻状况与体育锻炼的强度总")
plt.xlabel("婚姻状况")
plt.ylabel("体育锻炼的强度总")

enhanced_file_path = "E:/Desktop/nutrition_curve/婚姻状况与体育锻炼的强度总.png"
plt.savefig(enhanced_file_path)
plt.show()
```

5.2 绘制箱线图代码

```
#画箱线图
# 导入所需库
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats


# 读取Excel文件
data = pd.read_csv("E:/Desktop/data/基本情况.csv",encoding="gbk")

# 删除包含缺失值的行
cleaned_data = data.dropna(subset=['年龄', '早午晚餐得分', '婚姻状况', '职业'])

# 排除职业类别3的数据
# filtered_data = cleaned_data[cleaned_data['职业'] != 3]
filtered_data = cleaned_data

# 设置绘图风格
```

```
sns.set_style("whitegrid")

# 绘制婚姻状况与早午晚餐得分的箱线图
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
plt.figure(figsize=(10, 6))
sns.boxplot(data=filtered_data, x="婚姻状况", y="早午晚餐得分")
plt.title('早午晚餐得分与婚姻状况的关系')
plt.xlabel('婚姻状况')
plt.ylabel('早午晚餐得分')
enhanced_file_path = "E:/Desktop/nutrition_curve/早午晚餐得分与婚姻状况的关系(不排除职业3).png"
plt.savefig(enhanced_file_path)
plt.show()


# 绘制职业与早午晚餐得分的箱线图
plt.figure(figsize=(12, 6))
sns.boxplot(data=filtered_data, x="职业", y="早午晚餐得分")
plt.title('早午晚餐得分与职业的关系')
plt.xlabel('职业')
plt.ylabel('早午晚餐得分')
enhanced_file_path = "E:/Desktop/nutrition_curve/早午晚餐得分与职业的关系（不排除职业3）.png"
plt.savefig(enhanced_file_path)
plt.show()
```

附录 F　问题三决策树源代码

```
    from sklearn import tree
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np

data = pd.read_csv("E:/Desktop/data/merged_data.csv",encoding="gbk")
data

# 2.划分特征变量与目标变量

df = data[['吸烟量','饮酒量','运动量','膳食质量评价','BMI',
         '年龄','工作性质','患高血压','性别','文化程度'
         ]]
X = df.drop(columns='患高血压')
y = df['患高血压']

#df = data[['吸烟量','饮酒量','运动量','膳食质量评价','BMI',
         '年龄','工作性质','低高密度脂蛋白血症','性别','文化程度'
```

```
          ]]
#X = df.drop(columns='低高密度脂蛋白血症')
#y = df['低高密度脂蛋白血症']


# 3.划分数据集与测试集
from sklearn.model_selection import train_test_split
Xtrain,Xtest,Ytrain,Ytest = train_test_split(X,y,test_size=0.2)

clf = tree.DecisionTreeClassifier(criterion="entropy")
clf = clf.fit(Xtrain,Ytrain)
score = clf.score(Xtest,Ytest)#返回预测的准确accuracy
score

feature_name = ['吸烟量','饮酒量','运动量','膳食质量评价','BMI',
          '年龄','工作性质','性别','文化程度']

import graphviz
from pydotplus.graphviz import graph_from_dot_data
import pydot #pip install pydot
import pydotplus

rf_small = tree.DecisionTreeClassifier(criterion="entropy", max_depth = 3)
rf_small = rf_small.fit(Xtrain,Ytrain)
# # rf_small = RandomForestRegressor(n_estimators=10, max_depth = 3, random_state=42)
# # rf_small.fit(train_features, train_labels)
# # 拿到其中的一棵树
# tree_small = rf_small
# # 将图像导出为 dot 文件
# # dot_data = tree.export_graphviz(clf,feature_names=
    feature_name,class_names=['1','2'],filled=True,rounded=True )
# tree.export_graphviz(tree_small, out_file = 'small_tree.dot', feature_names = feature_name,
    rounded = True, precision = 1)
# # 绘图
# (graph, ) = pydot.graph_from_dot_file('small_tree.dot')
# # 展示
# graph.write_png('E:/Desktop/nutrition_curvesmall_tree.png');
# 绘图
graph = pydotplus.graph_from_dot_file('small_tree.dot')

# 展示
graph.write_png('tree.png')

#特征重要性
clf.feature_importances_
[*zip(feature_name,clf.feature_importances_)]
```

附录 G　问题四 K-Means 聚类源代码

```
    import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import f1_score, accuracy_score, normalized_mutual_info_score, rand_score
from sklearn.preprocessing import LabelEncoder
from sklearn.decomposition import PCA


df = pd.read_csv("E:/Desktop/data/merged_data.csv",encoding="gbk")
# print(df)

columns = list(df.columns) # 获取数据集的第一行，第一行通常为特征名，所以先取出
features = columns[:len(columns) - 1] #
    数据集的特征名（去除了最后一列，因为最后一列存放的是标签，不是数据）
dataset = df[features] #
    预处理之后的数据，去除掉了第一行的数据（因为其为特征名，如果数据第一行不是特征名，可跳过这一步）
attributes = len(df.columns) - 1 # 属性数量（数据集维度）
class_labels = list(df[columns[-1]]) # 原始标签

k = 3

# 这里已经知道了分3类，其他分类这里的参数需要调试
model = KMeans(n_clusters=k)
# 训练模型
model.fit(dataset)
# 预测全部数据
label = model.predict(dataset)
print(label)


def clustering_indicators(labels_true, labels_pred):
    if type(labels_true[0]) != int:
        labels_true = LabelEncoder().fit_transform(df[columns[len(columns) - 1]]) #
            如果数据集的标签为文本类型，把文本标签转换为数字标签
    f_measure = f1_score(labels_true, labels_pred, average='macro') # F值
    accuracy = accuracy_score(labels_true, labels_pred) # ACC
    normalized_mutual_information = normalized_mutual_info_score(labels_true, labels_pred) # NMI
    rand_index = rand_score(labels_true, labels_pred) # RI
    return f_measure, accuracy, normalized_mutual_information, rand_index


F_measure, ACC, NMI, RI = clustering_indicators(class_labels, label)
print("F_measure:", F_measure, "ACC:", ACC, "NMI", NMI, "RI", RI)
```

```
if attributes > 2:
    dataset = PCA(n_components=2).fit_transform(dataset) # 如果属性数量大于2，降维
# 打印出聚类散点图
plt.scatter(dataset[:, 0], dataset[:, 1], marker='o', c='black', s=7) # 原图
plt.show()
colors = np.array(["red", "blue", "green", "orange", "purple", "cyan", "magenta", "beige",
    "hotpink", "#88c999"])
# 循换打印k个簇，每个簇使用不同的颜色
for i in range(k):
    plt.scatter(dataset[np.nonzero(label == i), 0], dataset[np.nonzero(label == i), 1],
        c=colors[i], s=7)
plt.show()
```