基于餐厅消费数据的隐形资助研究-XGBoost 模型

摘要

隐形资助是通过大数据挖掘的形式，找准家庭经济困难学生的行为或经济状况特征，隐形认定（识别）经济相对困难学生群体，并通过隐形实施的方式给予适度的资助补偿，助力教育公平的实现。随着大数据技术的发展，我们可以通过学生在餐厅的海量消费数据来对学生的贫困程度进行预测，进而确定对贫困群体的资助方案。

针对问题 1，我们采用 **k-means 聚类算法**，将学生群体划分为 **k** 类，通过比较不同 **k** 值下误差平方和以及轮廓系数的取值，最终确定 **k** 取 3，其中**类别 1** 是消费水平低但消费最稳定的群体，**类别 3** 是消费水平高但消费最不稳定的群体，**类别 2** 的消费水平和消费稳定度都居中。为了反映三个群体的消费行为特征变化规律和饮食种类变化规律，我们从附件中提取出单次消费均价，早、中、晚消费均价，全年消费次数等十余种特征量，我们分别计算了三个群体三年统计周期内其消费特征的均值，导入 matlab 中绘制了变化图表体现其变化。总体而言，三个群体三年的消费水平和消费稳定性都有所提高，消费食品种类变多，饮食结构变的更加合理。

针对问题二，我们构建 **XGBoost** 模型并使用**启发式优化算法**来确定模型最佳参数，以此来预测附件 9 中学生的贫困度。首先将附件 1-3 提取到的特征量合并到附件 8 和 9 中，再通过**启发式优化算法**确定参数绘制网格参数，然后以附件 8 的数据为训练集来训练 **XGBoost** 模型，最后对附件 9 中的预测集进行预测。最终可以确定附件 9 中学生在三学年内的贫困度。

针对问题三：我们以具有多个预测特征量的嵌套 **XGBoost** 模型为基础，首先将附件 4-7 中提取到的特征量结合 vlookup 函数与附件 8 和 9 合并，以附件 8 内 250 个学生为训练集，使 $train\_test\_split$ 函数划分数据为训练集和测试集，比例为 8:2，随机种子为 42, 再训练 **XGboost** 模型并且使用 5 折交叉验证和 r2 分数作为评估指标，返回训练分数，确定最佳参数后预测并补全附件 8 本身，将误差控制在合理范围内并且增加特征量，然后同理以附件 8 为训练集训练 **XGboost** 模型并预测附件 9 中三学年内学生的贫困度。

针对问题 4，我们采用**熵权法**计算第三学年附件 4-7 中学生各项指标所占的权重，并计算附件 4-7 中每个同学的综合评价分数，综合评价分数越高，其贫困度越大，对 301 名学生进行排序，取前 80 名即为最终的资助对象。为了确定资金分配方案，我们假设所获资助金与贫困度之间存在线性关系，基于此假设，我们将资助金额进行**线性插值**，得到资助金额与我们综合评价分数之间的函数关系，由此可确定最终的资助金额分配方案。

关键字：**隐形资助　k-means 聚类　XGBoost　交叉验证　综合评价　线性插值**

一、 问题重述

1.1 题目背景知识

在高校资助工作中，判断并精准资助家庭经济困难的学生是极为关键的。而通过大数据挖掘的方式，我们可以实现隐形识别和资助困难学生的目标，并且可以保护学生的隐私，有助于实现教育公平的目标。同时随着数据存储与管理技术的日益完善，学生的部分消费数据，例如在餐厅的三餐消费，可以被记录并保存，而学生的经济状况可以通过餐厅的消费金额、消费品类和消费次数等信息来间接反映，所以我们通过大数据挖掘的方式挖掘有效数据，分析学生的消费特征，建立相关模型，最后实现隐形精准资助，在保护学生隐私、维护教育公平的前提下达成对困难学生的精准资助，促进教育公平的实现。

1.2 问题重述

问题 1：针对附件 0-7 提供的数据建立模型，挖掘不同代表性群体，并定量分析该群体三学年的主要消费行为特征变化规律、饮食种类变化规律等。注意建模前需对数据进行必要预处理 (如删除不相关数据、缺失补全、特征提取等)。

问题 2：除以上信息外，附件 8 给出部分同学第一学年后经其它方式认定的贫困程度等级 (粗粒度)，其中等级 2 准确 (可能不全)、其它等级认定可能有少量偏差。请建立数学模型依据消费行为 (附件 1-3) 预测贫困程度，补全附件 9 (不要改动附件 9 的已有数据及顺序) 并作为附件提交；进一步结合第 1 问研究结论预测该组同学第二、第三学年的贫困程度隐形认定等级，分析相关变化。

问题 3: 在第 2 问基础上，结合附件 4-7 饮食种类数据，改进你们的预测模型，比较分析相关同学的预测结果变化情况。

问题 4：通过以上贫困生本质特征挖掘，构建差异化 (细粒度) 资助额度分配算法，并以第三学年为例给出具体结果：对象为附件 4-7 中涉及的同学、资助总金额 10 万、资助人员 80 名，并对资助结果的公平合理性进行评估。。

二、 问题分析模型假设

2.1 问题分析

针对问题 1，我们采用 k-means 聚类算法，根据附件 1 第一学年的数据我们进行聚类分析，分析最佳的聚类个数，然后计算这几个聚类的消费特征和饮食种类的均值，绘制图表，体现若干群体三年来的变化规律；

针对问题 2，我们选择构建 XGBoost 模型，同时使用 K-fold 交叉验证法和遗传算法优化模型精度，利用附件 8 的数据进行模型训练，对附件 9 中的同学进行预测，并且对全体同学第二、三年的贫困程度进行预测。

针对问题 3，我们新加入了附件 4-7 的饮食种类数据，完善了 XGBoost 模型，并重新用模型进行了预测，得到了更为准确的预测数据。

针对问题 4，我们通过熵权法求得各指标所占的权重，对附件 4-7 中学生的贫困程度进行综合评价，并进行排序，算出综合评价得分与所获资助金之间的线性关系，合理分配了资助金。

2.2 模型假设

· 学生在食堂消费的数据真实可信，不考虑同学带饭，代刷卡等极少数特殊情况；

· 由于市场波动等因素的影响，食品价格不可能保持不变，因此我们以食品在统计周期内的平均价格作为食品价格的衡量值；

· 由其它方式认定的贫困程度等级得到的附件 8 中数据真实可信，不存在严重的评估误差 (如将不困难生列为困难生或将困难生列为不困难生)；

· 假设熵权法得到的权重是客观合理的，可以真实反映各个指标在贫困程度评价中的重要性；

· 假设资助完全遵照学生经济条件，遵从客观规律，按照贫困指标分配资助金额，不考虑其他干扰因素 (如学生的学习成绩，社会关系等)

· 在进行机器学习迭代时，不考虑数据采样偏差、标签偏差等系统偏差和随机误差，认为最终得到的 XGBoost 模型具有较强的可信度。

三、 数据预处理

由于题目给出的附件数据量大，数据冗余度大，无效信息多，还存在部分数据缺失，为此，我们首先进行数据预处理：我们发现附件 1-3 三年消费记录中存在大量的消费记录为 0，推测这些天很有可能是周末或假期或是收到了新冠疫情的影响，因此，我们规定：若某一天消费为 0 的同学占比超过 85%，我们就将这一天的数据删除，得到初步的数据表。

附件 4-7 对于部分同学的消费记录进行分析，我们发现存在一些数据没有记录相应的食物种类，由于食物种类对我们分析饮食规律价值较大，故我们删除了这部分为空的数据。

四、 问题一-基于 k-means 聚类算法模型的群体挖掘

4.1 k-means 聚类算法模型的简介

K-means 算法是一种典型的基于划分的聚类算法，也是一种无监督学习算法。他的基本思路是对给定的样本集，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。预先指定初始聚类数以及初始聚类中心，按照样本之间的距离大小，将样本集划分为若干个簇，根据数据对象与聚类中心之间的相似度，不断更新聚类中心的位置，不断降低类簇的误差平方和（Sum of Squared Error, SSE），当 SSE 不再变化时，聚类结束，从而得到最终结果。而空间中数据对象与聚类中心间的欧氏距离计算公式为：

$$d(X, C_i) = \sqrt{\sum_{j=1}^{m}(X_j - C_{ij})^2} \tag{1}$$

其中，X 为数据对象；C_i 为第 i 个聚类中心；m 为数据对象的维度；X_j，C_{ij} 为 X 和 C_i 的第 j 个属性值。

整个数据集的误差平方和 SSE 计算公式为：

$$SSE = \sum_{i=1}^{k}\sum_{X \in C_i}|d(X - C_i)|^2 \tag{2}$$

其中，SSE 的大小表示聚类结果的好坏；k 为簇的个数。

4.2 k-means 聚类算法挖掘不同代表性群体

要使用 k-means 算法，首先需要确定衡量学生消费的特征量，以下是经数据处理得到的几个消费特征的特征量。我们将其分为两类：一类反映学生的消费水平，称为消费水平特征量，该特征值可为后面分析贫困度作参考；另一类反映学生的饮食规律，称为规律特征值。

1. 单次消费均价：将学生三年的消费总费用除以天数得到单次消费单价，该特征值可大致衡量学生的消费水平；

2. 早、中、晚餐消费均价：分别将每位学生 3 年早、中、晚的消费总额除以相应天数，得到单餐消费均价，该指标不仅可以衡量对应早中晚的消费水平，亦可一定程度上反映学生早中晚的食物选择偏好，进而反映其饮食规律；

3. 全年消费次数：一定程度上反映学生的用餐消费频率；

4. 早、中、晚餐年消费次数：该指标反映了学生在统计周期内的吃早、午、晚餐的次数，反映学生的饮食规率；

5. 全年消费波动性：计算全年的消费金额的标准差来表示，该指标反映学生年度消费金额的波动情况，衡量其消费稳定性；

6. 日均消费极差：分别得到每日三餐消费的最大值和最小值，将两者作差，取年平均，该指标可反映日均消费波动幅度；

由附录 1-3 提取每个学生三年内消费特征的有效数据，导入 matlab 中进行 k-means 聚类分析，分别将聚类个数设为 1-10，分别计算其 SSE, 得到 SSE 随聚类个数变化的统计图：

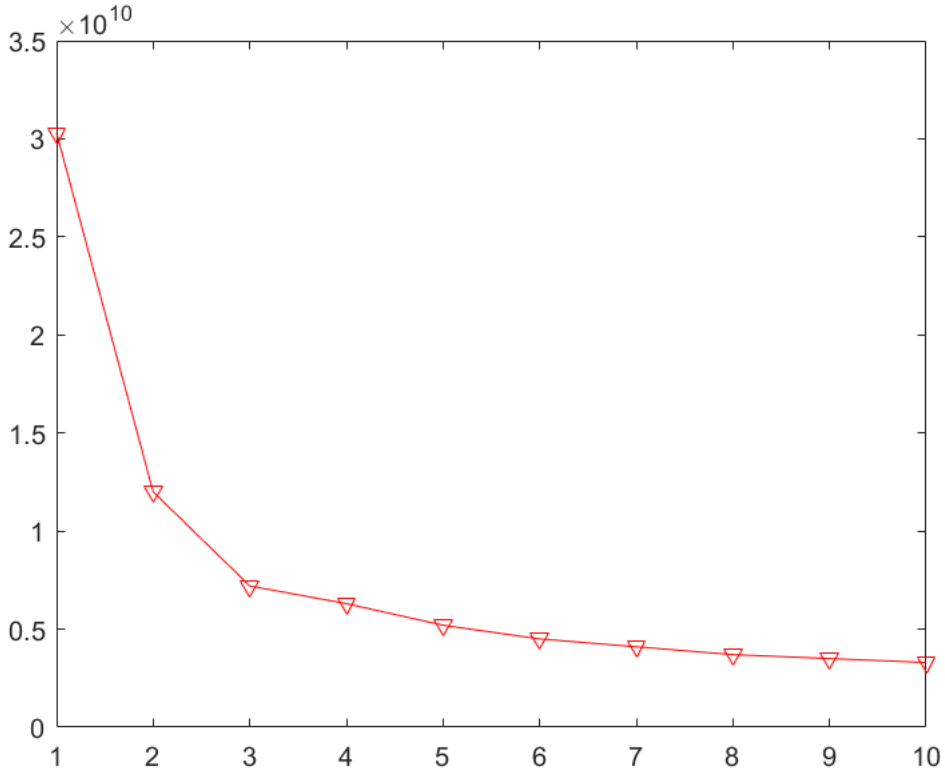


图 1　k-means 聚类数对比图

在此处我们运用肘部法则可以得到取聚类等于 3 时最为合适。但为求数据更加精确，即为了使我们得到的簇中，簇内尽量紧密，簇间尽量远离，我们引入轮廓系数。

其公式表达如下：

$$s = b - \frac{a}{max(a,b)} \tag{3}$$

其中 a 代表同簇样本到彼此间距离的均值，b 代表样本到除自身所在簇外的最近簇的样本的均值，s 取值在 [-1, 1] 之间。

判断：轮廓系数范围在 [-1,1] 之间。该值越大，越合理。s_i 接近 1，则说明样本 i 聚类合理；s_i 接近-1，则说明样本 i 更应该分类到另外的簇；若 s_i 近似为 0，则说明样本 i 在两个簇的边界上。所有样本的 s_i 的均值称为聚类结果的轮廓系数，是该聚类是否合理、有效的度量。使用轮廓系数 (silhouette coefficient) 来确定，选择使系数较大所对应的 k 值

根据对轮廓系数计算的相关算法，我们绘制了不同 k 值时的轮廓系数变化曲线。

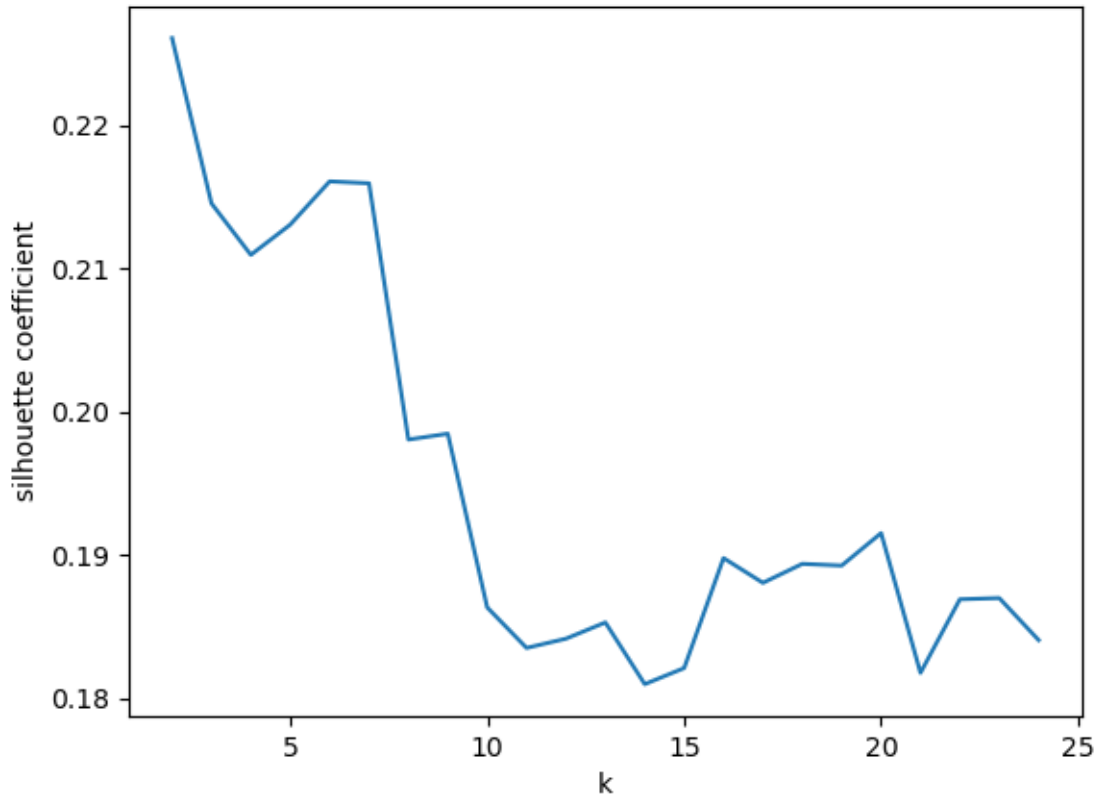


图 2　轮廓系数对比图

此时将轮廓系数对比图与 k-means 聚类数对比图进行比较可最终确定 k 值为 3。此时进行聚类分析。对学生群体进行划分，通过 python 中的 k-means 算法进行 10000 次迭代将学生分为三类群体：低消费、中等消费、高消费。

4.3 三个学年消费特征变化描述

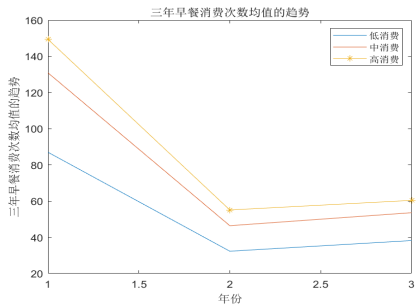


图 3　三年早餐消费次数均值的趋势

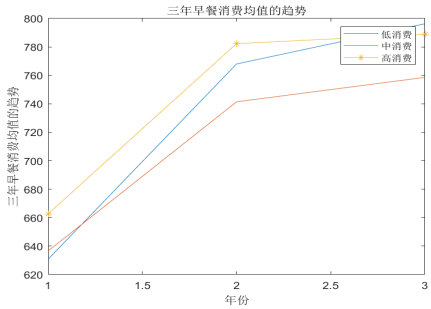


图 4　三年早餐消费均值的趋势

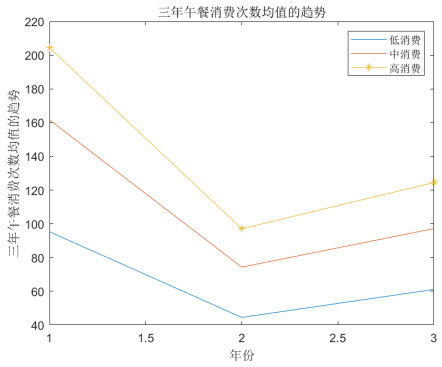


图 5　三年午餐消费次数均值的趋势

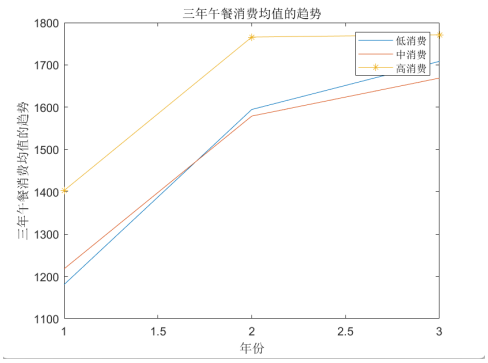


图 6　三年午餐消费均值的趋势

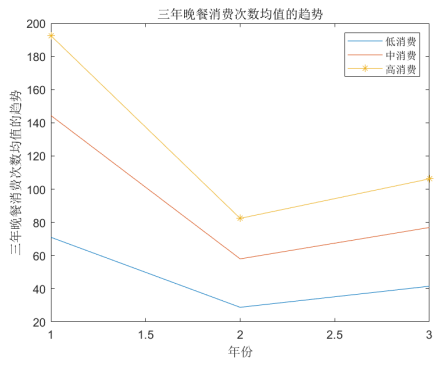


图 7　三年晚餐消费次数均值的趋势

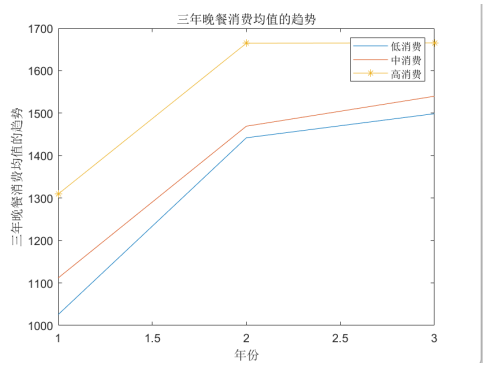


图 8　三年晚餐消费均值的趋势

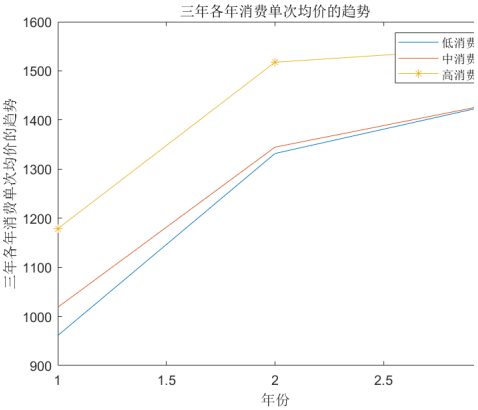


图 9　三年各年消费单次均价的趋势

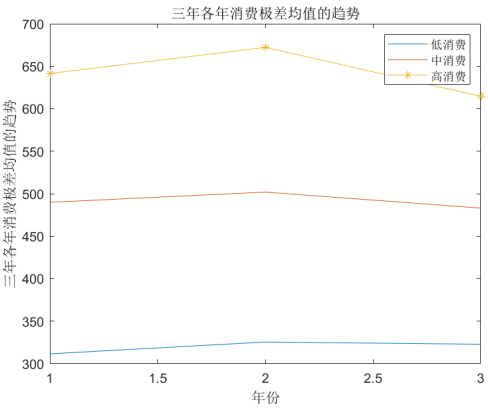
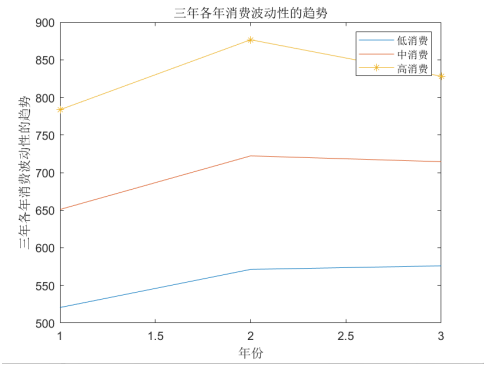


图 10　三年各年消费极差均值的趋势



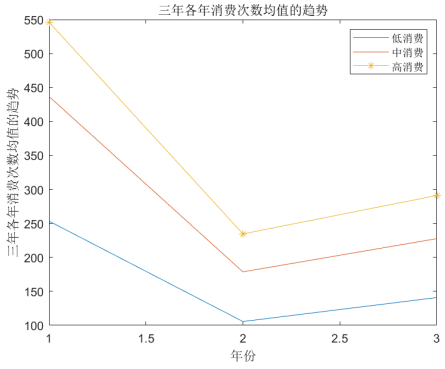


图 11　三年各年消费波动性的趋势　**图 12　三年各年消费次数均值的趋势**

低消费水平，低消费群体三年内早餐消费次数均值先下降后缓慢上升，三年早餐消费均值一直上升午餐和晚餐相应的变化趋势相同，低消费群体三年各年消费单次均价不断上升，三年各年消费极差均值基本保持稳定，但有小幅上升。三年各年消费波动性有缓慢上升，三年各年消费次数均值先下降后上升。

中消费水平，中消费群体三年内早餐消费次数均值先下降后缓慢上升，早餐消费均值一直稳步上升，午餐和晚餐同上，其中，中消费人群三年各年消费单次均价不断上升，各年消费极差均值基本保持稳定，三年消费波动性先上升后不变，三年各年消费次数均值先下降后上升

高消费水平，高消费群体三年内早餐消费次数均值先下降后上升，早餐消费均值不断上升或过于平缓和晚餐同升，其中，高消费群体三年内各年消费单次均价先上升后趋于平缓，三年消费极差均值先上升后下降有较明显的变化，明显各年消费波动性先上升后下降，三年各年消费次数均值先下降后上升

总体来看，三个群体的学生在三年中，其消费均价都在上升，而消费次数一般先下降后上升，这可能表明他们的消费能力在提高，同时他们的消费习惯也变得更加稳定。

4.4 三个学年饮食种类变化描述

我们合并附件 4-7，在提取特征量以前，我们规定每天 10 点以前为早餐，10 点-17 点为午餐，17 点之后为晚餐；设定 2 个消费划分值：400 分和 1000 分，低于 400 分的食物称为低价食物，介于 400 和 1000 分之间的食物称为中价食物，高于 1000 分的称为高价食物，

同衡量消费特征的几个特征量一样，我们引入几个衡量饮食结构（种类）的几个特征量：

1. 年消费种类数：我们认为附件 4-7 中每出现一种食物就记为一类，统计每个学生每年内三餐消费的总种类，该特征表征学生饮食的总体广泛度，若值较大，则代表学生

饮食较为均衡，没有挑食等习惯；若值偏小或异常偏小，则该学生可能存在挑食、饮食结构不均衡等问题。通过年消费种类的变化可大致反映饮食变化规律；

2. 低价、中价、高价食物消费占比：通过统计分析，将每个学生统计周期内相应食物的消费次数与总次数做商，即可得到相应食物消费占比，该特征值可反映学生的饮食结构（价格可一定程度上反映饮食种类）；

然后分析数据，得到 3 年统计周期内附件 4-7 学生的特征值数据。

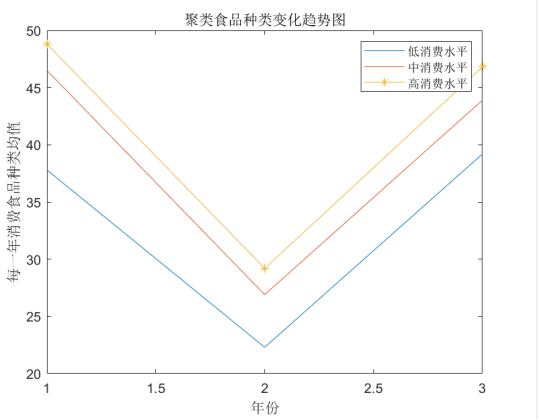


图 13　聚类食品种类变化趋势图

由图表分析，三个类群都在第二年消费食品种类数产生了明显的下降，我们分析可能是受到了新冠疫情影响，使得学生进入食堂的次数变少，从而食品种类数出现明显的下降，总体来讲，年消费食品种类数与消费水平成正相关，且高消费类群 3 年平均消费种类数在 40 附近，中等消费人群 3 年平均消费种类数在 35 附近，低消费水平类群 3 年平均消费种类数在 30 附近。

除此之外，我们还得到了各个类群的三种食物占比，该指标为后文改进 XGBoost 模型提供依据。

五、 问题二-XGBoost 模型对学生贫困程度的预测

5.1 XGBoost 理论基础：

XGBoost 模型是由 k 个基模型组成的一个加法模型，XGBoost 是在训练出一棵树的基础上，再训练下一棵树，预测它与真实分布的差距，通过不断训练来弥补差距的树，最终用树的组合实现对真实分布的模拟，其独特之处在于其目标函数包含损失函数和正则项两部分，损失函数代表着模型拟合数据的程度，我们通常用它的一阶导数指出梯度下降的方向，XGBoost 还计算了它的二阶导数，进一步考虑了梯度变化的趋势，拟和更快，精度更高；正则项用来控制模型的复杂程度，其正则项是一个惩罚机制，叶子结点的数量越多，惩罚力度越大，从而限制他们的数量；由于 XGBoost 采用贪心算法生成决

策树，为防止过拟合，采用了最大深度限制和后剪枝策略；其允许在每一轮的 boosting 中使用交叉验证，因此可以方便的得到最优的 boosting 轮数；由于 XGBoost 模型善于捕捉复杂数据之间的依赖关系，能从大规模数据集中获取有效的模型，因此我们选择该模型来预测学生的贫困程度。建模的基本步骤为：

1. 数据处理：将附件 8 所包含的数据进行采集，处理缺失值；

2. 模型训练：用 XGBoost 在训练集上训练模型，调整参数，如学习率、树的深度等；

3. 模型验证/调参；

4. 模型预测：利用最优参数的模型在测试集上进行预测，根据附件 1-3 预测其贫困度；

5. 结果评估：使用适合的评价指标召回率、精确率、F1 分数等指标评价模型的预测效果。

XGBoost 模型其预测函数：

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{4}$$

其中，F 表示模型中所有树的集合，K 表示模型中树的棵数，f 表示模型中某棵树，x_i 为样本 i 的特征向量。

XGBoost 在训练时对误判样本尤为关注，这种逼近式的拟合最终将导致模型在训练数据上的损失远小于在测试数据上的损失，预测能力较差。为此，在每次集成新树时，XGBoost 都通过将目标函数 obj(式（2) 向最小值优化，在降低自身在训练数据上损失的同时，缩减自身在训练数据和测试数据上的损失差距。

$$obj = \sum_{i=1}^{l} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{5}$$

$$L(y_i, f_t(x_i) + \widehat{y_i}^{t-1}) \tag{6}$$

对于第 t 棵树：

$$\begin{aligned} L &= \sum_{i=1}^{N} L(y_i, f_t(x_i) + \widehat{y_i}^{t-1}) \\ &= \sum_{i=1}^{N} [l(y_i, \widehat{y_i}^{t-1}) + \frac{\partial f_t(x_i) + \widehat{y_i}^{t-1}}{\partial \widehat{y_i}^{t-1}} * f_t(x_i) + \frac{1}{2} \frac{\partial^2 (f_t(x_i) + \widehat{y_i}^{t-1})}{\partial \widehat{y_i}^{t-1}{}_2} * f_t^2(x_i)] \\ &\approx \sum_{i=1}^{N} [l + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \\ &= \sum_{i=1}^{N} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \end{aligned} \tag{7}$$

加入正则项，假设 T 个叶子节点，W_j 使第 j 个节点预测值

$$\begin{aligned}\widehat{L} &= \sum_{j=1}^{T}[(\sum_{i\in L} g_i)w_j + \frac{1}{2}(\sum_{i\in L_j} h_i + l)w_j^2] + \gamma T \\ &= \sum_{j=1}^{T}[G_i W_i + \frac{1}{2}(H_J + l)W_j^2] + \gamma T \\ &= -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + l} + \gamma T\end{aligned} \tag{8}$$

最终可以算出目标函数为：

$$obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_i + \lambda} + \gamma T \tag{9}$$

在对该模型进行检验时，我们首先认为贫困的预测属于二分类问题，那么引入二分问题的 4 种情况：

1. 真正例（True Positive，TP）：模型正确地预测样本为正例（Positive）。实际上是正例，模型预测也为正例。

2. 假正例（False Positive，FP）：模型错误地预测样本为正例。实际上是负例，但模型预测为正例。

3. 假反例（False Negative，FN）：模型错误地预测样本为负例。实际上是正例，但模型预测为负例。

4. 真反例（True Negative，TN）：模型正确地预测样本为负例。实际上是负例，模型预测也为负例。

对于此问题，数据中有三个类别（贫困程度等级）：0（不困难），1（一般困难），2（特别困难），其中 2 准确但可能不全，0 和 1 可能有少量偏差。

以下进行模型交叉验证评测：为了检验模型，需要以下几个参数：

1. 召回率（Recall）：它是在二分类问题中衡量模型对正例样本的识别能力的指标。召回率表示模型能够正确预测的正例样本数量与真实正例样本数量之间的比例。

召回率的计算公式：

$$R = \frac{TP}{TP + FN} \tag{10}$$

XGBoost 模型的精确率：

2. 精确率（Precision）是二分类问题中衡量模型对预测为正例的样本中真正例数量的指标。计算精确率时，我们可以按照以下公式计算：

$$P = \frac{TP}{TP + FP} \tag{11}$$

3.F1 分数：F1 分数是综合考虑精确率（Precision）和召回率（Recall）的一种评估指标，用于衡量二分类问题中模型的性能。

F1 分数的计算公式：

$$F1 = \frac{2 \times P \times R}{P + R} \tag{12}$$

F1 分数综合了精确率和召回率，旨在找到一个平衡点，以综合评估模型对正例和负例的预测能力。F1 分数的取值范围为 0 到 1，其中 1 表示最佳性能，0 表示最差性能。F1 分数越高，说明模型在预测正例和负例方面的性能越好。

5. 准确率（Accuracy): 模型预测正确的样本数占总样本数的比例，是模型的总体预测准确率。

5.2 模型的交叉验证

模型交叉验证（Model cross-validation）是一种常用的验证机制，用于评估机器学习模型的性能和泛化能力。它通过将数据集划分为多个互斥的子集，然后将模型在这些子集上进行多次训练和测试，从而评估模型在不同子集上的表现。我们采用 K-fold 交叉验证进行，将原始数据集划分为 K 个互斥的子集，称为折（folds）。选取一个折作为验证集，其余 K-1 个折作为训练集。在训练集上建立 XGBoost 分类模型，并在验证集上进行测试。记录模型在验证集上的性能指标，比如准确率、损失函数等。重复以上步骤，每次选择一个不同的验证集，直到所有的折都被用作验证集。对于 K 次训练和验证，计算性能指标的平均值和标准差，作为模型的最终性能评估。

5.3 建立 XGBoost 模型预测贫困度

首先我们运用 vlookup 函数，以从附件 2、3 以及 4 中提取到的第一学年的特征向量为参照，总计十一个特征量，对附件 8、9 中的数据进行补全，实现附件 8、9 中的数据从而完善 XGBoost 模型中的训练集。

然后我们导入附件 8、9 至 python 程序中，以附件 8 的数据为训练集，通过已知的贫困度以及其对应的第一学年各项指标的特征量，建立 XGBoost 模型，并且通过启发式优化算法确定了相对最优参数，从而进行预测对附件 9 中缺失的贫困度指标进行补全，此时训练所得各项参数重要性表如下：

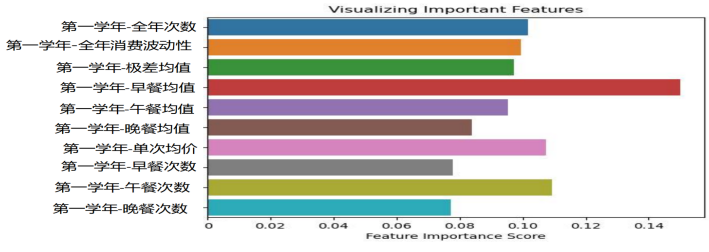


图 14　XGBboost 模型各项参数重要性

XGBoost 模型评价指标

	精确度	召回率	F1分数	支持度
0	0.78	1.00	0.88	3350
1	0.97	0.10	0.18	578
2	0.94	0.10	0.18	486
精确度			0.78	4414
宏平均	0.90	0.40	0.41	4414
加权平均	0.82	0.78	0.71	4414

图 15　XGBoost 模型评价指标表

数据中贫困度总共为三个指标，分别为 0、1 和 2，以下是对建立的 XGBoost 模型进行的相关检验。对于贫困度为 0 的学生群体，该模型对应的精确度，召回率，F1 分数以及支持度均较高，但对于贫困度 1 和 2 的学生群体，该模型只有精确度较高，召回率、F1 分数和支持度较低。同时整体而言精确度、宏平均以及加权平均值均较高。

综上，对于该模型而言，贫困度指标为 0 时均能较好进行预测，而指标为 1 和 2 时预测相对较差，这表明该模型在贫困度指标为 1 或 2 时仍有较大的改进空间。

对模型进行基本评估后，我们利用该模型对附件 9 中数据进行预测，所得结果在附件中。

对预测结果进行分析的过程中，我们发现该模型对贫困度为 0 的预测准确性较高，能较好地完成预测任务，但是对贫困度为 1 或 2 的学生群体预测结果并不理想，这可能是因为我们训练集中第一学年的贫困度为 1 或 2 的学生数过少有关系。模型无法得到较好的训练，以致无法进行良好的预测。

以下在解决问题三的过程中我们将引入部分同学新的特征量来对题目二中建立的模型进行优化。

六、 问题三-基于具有多个预测特征量的嵌套 XGBoost 模型对贫困度的预测

附件 4-7 中，我们得到了 300 个学生的消费时间、金额以及种类的详细数据，对这些数据进行不相关数据删除，数据补全，特征提取后得到新的特征量，并将这部分特征量与附件 1-3 特征提取所得的特征量进行合并，得到全新的训练集。

以下我们在单次训练 XGBoost 模型时对第二问中训练模型的方法进行改进，首先不再对特征量进行简单划分，而是借助划分函数来将原本的训练集划分为训练集和测试集，有利于更合理的评估我们建立的 XGBoost 模型的准确性，对于修改参数更加有利。同时因为虽然附件 4-7 的引入增加了特征向量的个数，但是总体数据量不够，我们需要嵌套 XGBoost 模型预测多个特征量来扩大训练集，所以我们将原本单变量预测的 XGBoost 模型改为多变量预测的 XGBoost 模型，同时使用 5 折交叉验证和 r2 分数作为评估指标，返回训练分数，并进行合理的网格参数修改。

首先附件 4-7 中提取得到的有效数据对应的学生数量只有 250 个，所以我们利用这部分学生对应的特征量对附件 8 和 9 中学生相对缺失的特征量进行预测。最后补全这部分特征量后以附件 8 整体为训练集，附件 9 为预测集建立新的 XGboost 模型，最后对附件 9 中对应学生三学年的贫困度进行预测。

以下是特征重要性表以及优化后的 XGBoost 模型的评估：

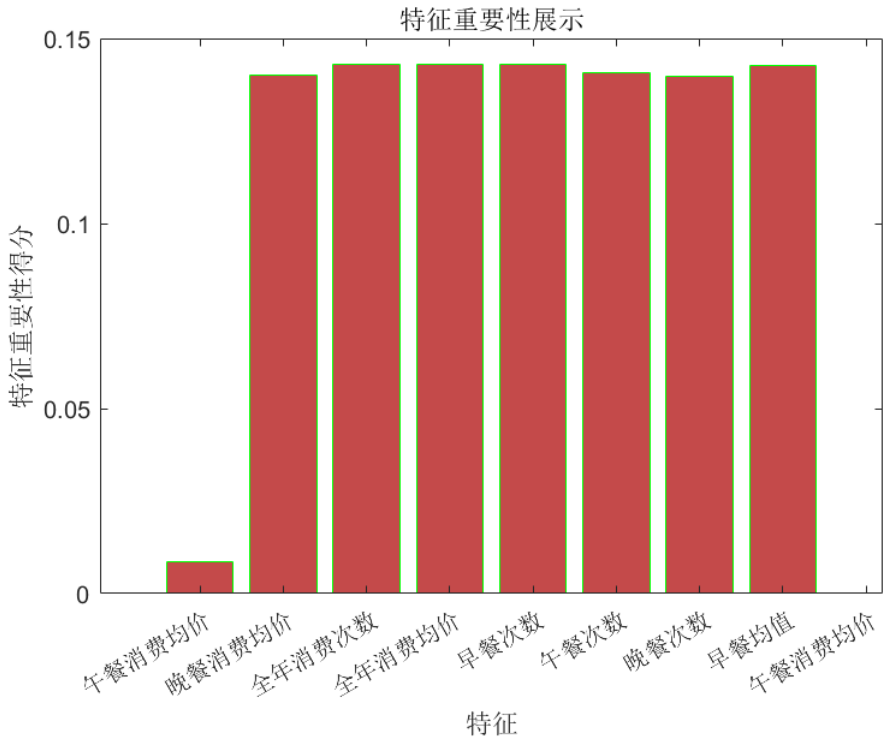


图 16　优化模型后的特征重要性

	精确率	召回率	F1得分
0	0.84	1.00	0.91
1	0.90	0.34	0.49
2	0.88	0.64	0.74
准确率			0.90
平均值	0.87	0.66	0.71

图 17　优化后的 XGboost 模型的评估

从上表可以看出，对贫困度为 0 的学生群体预测的各项指标仍旧良好，同时对贫困度为 1 或 2 的学生预测准确性得到极大提升。对贫困度为 1 的学生的 F1-score 提升为两倍，对贫困度为 2 的学生的 F1-score 提升为三倍，最终的准确率提升到 0.90，优化后的 XGboost 模型相对第二问中的模型取得明显提升，对附件 9 中学生的贫困度预测准确性得到加强。

七、 问题四-基于多指标综合评价的贫困资助问题

7.1 理论基础：熵权法

理论基础：熵权法（Entropy Weight Method）是一种多属性决策分析方法，用于确定各个属性在决策中的权重。它基于信息论中的熵概念，并通过计算属性的信息熵来衡量属性的重要性和对决策结果的贡献。熵是信息论中衡量不确定性的度量，用来衡量一组数据的随机性和不确定性程度。在熵权法中，通过计算属性的熵值来度量属性的不确定性，进而确定属性的重要性和权重。

假设有 n 个评价对象，m 个评价指标，第 i 个对象关于第 j 个指标变量的取值 aij(i=1,2,...,n;j=1,2,...,m), 构造数据矩阵 A=$(a_{ij})_{nxm}$.

基于熵权法的评价方法如下：

1. 利用原始数据矩阵 A=$(a_{ij})_{n\times m}$ 计算 p_{ij}, 即第 i 个评价对象关于第 j 个指标值的比重：

$$p_{ij} = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}} i = 1, 2\cdots, n, j = 1, 2, \cdots, n \tag{13}$$

2. 计算第 j 项指标的熵值：

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \ln p_{ij} \tag{14}$$

3. 计算第 j 项指标的熵值:

$$g_j = 1 - e_j, j = 1, 2, \cdots, m \tag{15}$$

4、计算第 j 项指标的权重:

$$w_j = \frac{g_j}{\sum_{j=1}^{m} g_j}, j = 1, 2, \cdots, n \tag{16}$$

5. 计算第 i 个评价对象的综合评价值:

$$s_i = \sum_{j=1}^{m} w_j P_{ij} \tag{17}$$

评价方法主要依赖综合评价值，评价值越高越好

7.2 熵权法-综合评价对学生的贫困程度进行评分

首先，我们以第三学年的数据为依据，对附件 4-7 的学生进行数据统计，分别计算贫困程度为 0,1,2 学生的特征量均值，得到如下数据表:

三个贫困程度学生群体特征

贫困程度(2是特别困难，1是一般困难，0是不困难)	全年次数	单次均价	早餐次数	午餐次数	晚餐次数	早餐均值	午餐均值	晚餐均值	全年消费波动性	极差均值
0	380.5	1039.1	121.4	139.5	119.6	665.2	1250.6	1143.8	645.5	457.9
1	390.3	992.5	118.6	141.8	129.8	620.2	1195.7	1066.3	620.8	448.1
2	426.4	902.4	138.3	157.1	130.9	542.9	1119.5	990.9	600.8	439.2

图 18　三个贫困程度学生群体特征

在 10 个指标中，全年次数，早餐次数，午餐次数，晚餐次数与贫困度成正相关，单次均价，早、中、晚餐均价与贫困度成负相关。最终我们得到的评价分数越高，说明学生贫困等级越高。按照熵权法计算公式，得到以下权重表:

然后，我们将附件 4-7 所有学生第三年的特征量绘成表格，计算其每个人的综合得分

7.3 利用线性插值对 80 位同学进行资助金额分配

我们假设资助金额与贫困程度（综合评价得分）成线性关系，即满足 y=ax+b 的形式，首先，我们给每一位同学一个基础资助额度 400 元，总计 32000 元，剩余 68000 元，紧接着将剩下的 68000 元按照线性关系分配给贫困程度前 80 名的学生，将两者相加即的每个学生得到的资助金。

通过 matlab 计算，得到以下资助金额表，全部人员的见附件:

1	综合评价	序号	排名	分配金额
2	0.009049154	5160	1	1763
3	0.008751144	4950	2	1718
4	0.008453133	4901	3	1673
5	0.007574572	3186	4	1541
6	0.009062873	76	5	1765
7	0.008424725	5041	6	1669
8	0.006780292	4907	7	1421
9	0.007637051	5101	8	1551
10	0.006647964	5086	9	1402
11	0.007003343	5042	10	1455
12	0.006920323	5023	11	1443
13	0.006605838	4914	12	1395
14	0.007142427	5121	13	1476
15	0.005689093	4964	14	1257
16	0.006343021	5069	15	1356
17	0.007428742	4957	16	1519
18	0.006732824	5098	17	1414
19	0.008132702	5096	18	1625
20	0.00650441	5118	19	1380
21	0.006117924	5001	20	1322
22	0.006195272	5107	21	1333
23	0.006310986	4969	22	1351

图 19　部分人员资助金额表

7.4 资助方案合理性分析

我们计算出需要资助的 80 人各个特征指标的均值，与附件 8 中给出的贫困程度分别为 0、1、2 的同学进行对比，对比图如下：

所选出 80 位同学与附件的贫困群体特征对比

贫困程度(2是特别困难，1是一般困难，0是不困难)	全年次数	单次均价	早餐次数	午餐次数	晚餐次数	早餐均值	午餐均值	晚餐均值	全年消费波动性	极差均值
0	380.5	1039.1	121.4	139.5	119.6	665.2	1250.6	1143.8	645.5	457.9
1	390.3	992.5	118.6	141.8	129.8	620.2	1195.7	1066.3	620.8	448.1
2	426.4	902.4	138.3	157.1	130.9	542.9	1119.5	990.9	600.8	439.2
贫困评分前八十名同学	432.5	912.5	128.6	149.5	128.6	550.6	1132.6	1022.2	605.4	441.3

图 20　所选出 80 位同学与附件的贫困群体特征对比

由比较图可以看出，贫困程度评分前八十名同学的各项评价指标都介于贫困程度 1 和 2 类群的各项指标之间，可以看出我们对于前 80 位的贫困生的资助方案是合理的。基于熵权法得到的资助方案综合分析了可以反映学生贫困程度的各个指标，为每位学生生成一个全面的贫困程度评分。这种方案因为参考特征量多而减少了评判的误差，根据学生综合得分分配金额，评分较高则得到的资助金额较高，评分较低则得到的资助金额较少，实现了尽可能公平地对资助金额进行分配，体现了公平性和差异化的原则，符合教育援助的基本宗旨。

八、 模型的评价与改进

8.1 模型的优点

1. 在提取数据时，我们按照数据的特征对数据进行了预处理，分类和整合，降低了数据的冗余度和复杂度，使大数据数据价值密度相对提高；

2. 我们假设了大量的消费特征量和规律特征量，从海量数据中尽可能的提取到有用信息，尽可能全面的反映学生的消费特征，为评价学生贫困程度提供了重要依据，有力支撑了隐形资助的进行；

3. 我们使用 k-means 聚类分析，采用肘部法则和轮廓系数来判断聚类数，得到按照消费水平划分的三个聚类，其具有典型的代表性；

4. 对于贫困程度的预测我们使用了 XGBoost 模型，利用机器学习研究特征量与贫困程度的内在联系，同时，在搭建完模型后，我们使用 K-fold 交叉验证法和遗传算法优化模型参数，使模型的预测能力更强；

5. 在资助金额分配时，我们采用熵权法 + 线性插值综合分析法，充分考虑了体现贫困度的特征量的占比。

8.2 模型不足

1. 在实际饮食中，男女的进食量存在一定的差异，较多情况下在同样的消费金额下男生的平均消费水平要低于女生，而我们在建模时选择忽略性别差异的因素，因为性别差异造成的直接饮食量与价格之间的关系时未知的，但是可能会对模型的准确性有一定影响。

2. 在构建 XGBoost 模型时，由于建模时需要训练集和预测集的特征向量一一对应，所以在最初就先对多个预测集中缺失的特征量进行预测，而后建立含多个预测值的 XGBoost 模型来再进一步预测贫困度，有一定的可能性会引起一定误差。

3. 仅仅通过学生在食堂三餐的消费情况不能全面反映学生的饮食情况，在外卖业发达的今天，许多学生选择在校园点外卖来改善伙食，而这部分数据是缺失的，难以进行分析并归入训练集来优化模型。

参考文献

[1] 李映铮, 李志斌, 金磊等.XGBoost 机器学习在光电编码器误差补偿中的应用 [J]. 光学仪器,2023,45(01):32-37.

[2] 聂雷, 杨拓, 张俊杰等. 基于多属性决策和 k-means 聚类的车载安全消息中继选择方法 [J/OL]. 武汉大学学报 (理学版):1-8[2023-07-02].DOI:10.14188/j.1671-8836.2022.0198.

[3] 浩杰, 马超, 李东东等. 基于熵权法-TOPSIS 对桂林市各县级物流产业分析与障碍诊断研究 [J]. 上海商业,2023(03):214-216.

[4] 文斌, 廖晶, 夏国恩. 基于商务智能的广西高校大学生餐饮消费行为研究 [J]. 经贸实践,2018(24):35-36.

附录 A　绘图和提取特征值–matlab 源程序

```
%求均值
mem=readmatrix("附件4-7特征处理 - 4-7采用第一年数据新版.xlsx");
count=0;
sum=0;
for i=1:301
if (mem(i,2)==2)
count=count+1;
sum=sum+mem(i,19);
end
end
t=sum/count

%绘制折线图
x=[1,2,3];
y1=[37.8,22.3,39.2];
y2=[46.5,26.9,43.9];
y3=[48.8,29.2,46.8];
plot(x,y1,x,y2,x,y3,"*-");
xlabel("年份");
ylabel("每一年消费食品种类均值")
legend("低消费水平","中消费水平","高消费水平")
title("聚类食品种类变化趋势图")

%得出贫困数据
a=readmatrix("C:\Users\阴子昂\Desktop\附件4-7特征处理 - 4-7采用第三年数据新版 -.xlsx");
temp=a(:,1);
a(:,1)=[];
[n,m]=size(a);
p=a./sum(a);
e=-sum(p.*log(p))/log(n);
g=1-e;
w=g/sum(g);
s=w*p';
[ss,ind1]=sort(s,"descend");
xuhao=zeros(1,301);
for i=1:301
xuhao(1,i)=temp(ind1(1,i),1);
end
writematrix([1:n;ss;xuhao],"C:\Users\阴子昂\Desktop\学生的综合评价值 - 副本 - 副本.xlsx");


%绘制柱状图
x = [0,1,2,3,4,5,6,7];
y = [ 0.008446,0.139950,0.142862,0.142872,0.142937,0.140615,0.139718,0.142588];
```

```
GO = bar(x,y,'edgeColor','green'); %使用bar函数绘制柱状图
set(gca,"xTick",0:9);
set(gca,"XTickLabel",{"午餐消费均价","晚餐消费均价","全年消费次数","全年消费均价","早餐次数","午餐次数","晚餐次数"
GO.FaceColor = [196/255,74/255,74/255]; %通过GO句柄自定义柱形的颜色
xlabel("特征");
ylabel("特征重要性得分");
title("特征重要性展示");

%特征提取
newdata1=readmatrix("附件123的数据特征提取.xlsx");
flag=0;
t=zeros(5415,1);
for i=1:5415
if(newdata1(i,3)==2&&~isnan(newdata1(i,33)))
flag=flag+1;
t(i,1)=newdata1(i,33);
end
end
he=0;
for i=1:5415
he=he+t(i,1);
end
k=he/flag
```

附录 B　keans 聚类算法-python

```
# 导入库
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.impute import SimpleImputer

# 读取Excel文件
df = pd.read_excel(r"C:\Users\92412\Desktop\数据\附件123的数据特征提取.xlsx")

# 初始化一个新的DataFrame来保存结果
results = pd.DataFrame()

# 创建KMeans对象
kmeans = KMeans(n_clusters=3, max_iter=50) # 指定分成三类，并且最大迭代次数为50



# 循环处理每个群体
for group in [1, 2, 3]:
    # 选择该群体的数据
```

```
    group_df = df[df['聚类种类'] == group]

    # 处理缺失值
    group_df = group_df.dropna()
    group_df = group_df.fillna(group_df.mean())
    imputer = SimpleImputer(strategy="mean")
    group_df = imputer.fit_transform(group_df)

    # 进行聚类
    kmeans.fit(group_df)
    labels = kmeans.labels_

    # 把labels变量存入results中，并添加群体标签
    labels_df = pd.DataFrame(labels, columns=["簇标签"])
    labels_df["群体"] = group
    results = pd.concat([results, labels_df], ignore_index=True)

# 将结果导出到新的Excel文件
results.to_excel(r"C:\Users\92412\Desktop\数据\result (2).xlsx")
```

附录 C　单特征值预测的 XGBoost 算法

```
import pandas as pd
import xgboost as xgb
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# 设置Matplotlib的默认字体为SimHei
plt.rcParams['font.sans-serif'] = ['SimHei']

# 加载数据
data_train = pd.read_excel(r"C:\Users\92412\Desktop\2023年校赛命题\C题\附件8 已知贫困标签1.xlsx")
data_predict = pd.read_excel(r"C:\Users\92412\Desktop\2023年校赛命题\C题\附件9
    问题2待补全标签数据1.xlsx")
# 输出训练集的基本信息
data_train.info()
# 输出测试集的基本信息
data_predict.info()

# 删除训练集和测试集中空白行
data_train.dropna(axis=0, how='all', inplace=True)
data_predict.dropna(axis=0, how='all', inplace=True)
```

```
# 给训练集中每一列命名
data_train.columns =
    ['序号','贫困程度','四年消费总种类','价格低消费占比','价格中消费占比','价格高消费占比','早餐消费均价','午餐消费
# 给测试集中每一列命名
data_predict.columns =
    ['序号','贫困程度','四年消费总种类','价格低消费占比','价格中消费占比','价格高消费占比','早餐消费均价','午餐消费

# 删除训练集和测试集中空白列
data_train = data_train.dropna(axis=1, how='all')
data_predict = data_predict.dropna(axis=1, how='all')



# 在训练之前，把训练数据转换成numpy数组
X_train = data_train.iloc[:, 2:18].values
y_train = data_train.iloc[:, 1].values
# 定义模型
model = xgb.XGBClassifier()

# 定义参数网格
param_grid = {
    'max_depth': [2, 4, 6],
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
}

# 创建网格搜索对象
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='accuracy',
    return_train_score=True)

# 训练模型
grid_search.fit(X_train,y_train)

# 输出最优参数
print("Best parameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)

# 在预测之前，把预测数据转换成numpy数组
X_test = data_predict.iloc[:, 1:17].values

# 使用最优模型预测
y_pred = grid_search.predict(X_test)

# 保存预测结果到原数据中
data_predict.iloc[:, 0] = y_pred
data_predict.to_excel(r"C:\Users\92412\Desktop\2023年校赛命题\C题\test.xlsx", index=False)

# 输出模型报告
```

```
print("Classification report:\n", classification_report(y_train, grid_search.predict(X_train)))

# 输出特征重要性
feature_importances = grid_search.best_estimator_.feature_importances_
# 创建一个新的数据框，包含特征名和特征重要性
feat_imp = pd.DataFrame({'feature': X_train.columns, 'importance': feature_importances})
# 按照特征重要性降序排序
feat_imp = feat_imp.sort_values(by='importance', ascending=False)
# 绘制柱状图
sns.countplot(data=feat_imp, x="reputation", order=feat_imp.mean().index)
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Visualizing Important Features")
# 调整柱状图的横轴范围
plt.xlim(0, 1)
# 给柱状图底下加上变量名
plt.xticks(range(len(data_train.columns)), data_train.columns)
plt.show()
```

九、 多特征值预测的 XGBoost 算法

```
   import pandas as pd
import xgboost as xgb
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.multioutput import MultiOutputRegressor
from sklearn.multioutput import MultiOutputClassifier
import numpy as np # 导入numpy模块
# 设置Matplotlib的默认字体为SimHei
plt.rcParams['font.sans-serif'] = ['SimHei']
# 加载数据
data_train = pd.read_excel(r"C:\Users\92412\Desktop\2023年校赛命题\C题\附件8 已知贫困标签1.xlsx")
data_predict = pd.read_excel(r"C:\Users\92412\Desktop\数据\附件4-7特征处理 -
    4-7采用的三年总新版数据1.xlsx")
# 输出训练集的基本信息
data_train.info()
# 输出测试集的基本信息
data_predict.info()
data_train.fillna(0, inplace=True)
data_predict.fillna(0,inplace=True)

# 给训练集中每一列命名
data_train.columns =
```

```
    ['序号','贫困程度','四年消费总种类','价格低消费占比','价格中消费占比','价格高消费占比','早餐消费均价','午餐消费

# 给测试集中每一列命名
data_predict.columns =
    ['序号','贫困程度','四年消费总种类','价格低消费占比','价格中消费占比','价格高消费占比','早餐消费均价','午餐消费

# 删除训练集和测试集中空白列
data_train = data_train.dropna(axis=1, how='all')
data_predict = data_predict.dropna(axis=1, how='all')

# 选择第2到9列的特征作为X
X = data_train.iloc[:, 2:17]

# 选择第10到17列的标签作为y
y = data_train.iloc[:, 10:17]
y = y.squeeze()

# 使用train_test_split函数划分数据
from sklearn.model_selection import train_test_split

# 划分数据为训练集和测试集，比例为8:2，随机种子为42
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# 输出划分后的数据形状
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

# 定义模型
model = MultiOutputRegressor(xgb.XGBRegressor(base_score=0.4))



# 定义参数网格
param_grid = {
    'estimator__max_depth': [2, 4, 6],
    'estimator__n_estimators': [50, 100, 200],
    'estimator__learning_rate': [0.01, 0.1, 0.2],
}

# 创建网格搜索对象
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='r2', return_train_score=True)

# 训练模型
grid_search.fit(X_train,y_train)

# 输出最优参数
print("Best parameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)
```

```
# 使用最优模型预测训练集和测试集
y_pred_train = grid_search.predict(X_train)
y_pred_test = grid_search.predict(X_test)

# 使用最优模型预测待补全标签数据
X_predict = data_predict.iloc[:, 2:17]
y_pred_predict = grid_search.predict(X_predict)

# 把预测结果转换成一个数据框，列名为y1，y2，...，y7
y_pred_df = pd.DataFrame(y_pred_predict, columns=['y'+str(i) for i in range(1, 8)])
# 和data_predict合并，按照行索引对齐
data_predict = pd.concat([data_predict, y_pred_df], axis=1)
# 保存到excel中
data_predict.to_excel(r"C:\Users\92412\Desktop\2023年校赛命题\C题\test.xlsx", index=False)

threshold = 1000
y_train_label = np.where(y_train > threshold, 1, 0).ravel()
y_pred_train_label = np.where(y_pred_train > threshold, 1, 0).ravel()
# 然后用classification_report函数来评估分类效果
print("Classification report on train set:\n", classification_report(y_train_label,
    y_pred_train_label))
threshold = 1000
y_test_label = np.where(y_test > threshold, 1, 0).ravel()
y_pred_test_label = np.where(y_pred_test > threshold, 1, 0).ravel()
# 然后用classification_report函数来评估分类效果
print("Classification report on test set:\n", classification_report(y_test_label,
    y_pred_test_label))
print("Classification report on test set:\n", classification_report(y_test_label,
    y_pred_test_label, labels=[0, 1, 2]))

feature_importances = []
for estimator in grid_search.best_estimator_.estimators_:
   feature_importances.append(estimator.feature_importances_)
# 然后你可以用一个数据框来展示每个回归器的特征重要性

feature_importances_df = pd.DataFrame(feature_importances, index=y_train.columns)
print(feature_importances_df.mean())
print(feature_importances_df)
sns.countplot(data=feature_importances_df, x="reputation",
    order=feature_importances_df.mean().index)
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Visualizing Important Features")
# 调整柱状图的横轴范围
plt.xlim(0, 1)
# 给柱状图底下加上变量名
```