基于餐厅消费数据的隐形资助模型——XGBoost 预测模型

摘　要

本文探讨了一种新型的隐性资助模型——基于大学生餐厅消费数据的隐形资助模型。在当前社会中，精准资助的重要性日益凸显。特别是对于高校家庭经济困难学生，如何准确识别他们并提供有效的援助，是当前高等教育贫困援助工作的关键问题。我们利用大数据技术，通过统计和分析大学生的餐厅消费数据，建立隐性资助模型。该模型将能够根据学生的消费数据，评估其经济贫困程度，然后精准地给予援助。助力教育路上的精准扶贫。

在构建模型之前，我们首先对数据进行了必要的处理，并对两类数据集进行**特征提取**，总计提取到 20 多个特征，用于后续的模型构建。

在解决**问题 1** 时，我们采用了 **k-means 聚类算法**，通过肘部法则取 k 值为 3，对第一学年的学生进行聚类，**类别 1** 的学生是**消费最低但最稳定**的群体。**类别 3** 的学生是**消费最高但最不稳定**的群体，他们可能是经济状况较好或者更愿意消费的学生。**类别 2** 的学生在这两者之间，**消费水平和消费稳定性都中等**。

然后我们计算了这些群体的**三年的消费特征**的均值，绘制图表，来体现这些群体三年来的消费特征变化。三个群体的学生在三年的时间里，**其消费均价都在上升，而消费次数在下降**，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。

在解决**问题 2** 时，我们构建了 **XGBoost 模型**，同时使用**交叉验证、启发式算法**寻优来提高模型的精确度。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学年的贫困程度进行预测。模型**准确度达到 0.76**，效果很好。

在解决**问题 3** 时，我们将附件 4-7 中**提取到的新的特征**一起纳入模型的指标体系中，优化我们的 XGBoost 模型，使得**模型准确率达到了 0.84，提升了 10.53%**。更重要的是，新的模型**对贫困等级为 1、2 的学生预测准确度翻了几十倍**之多。

在解决**问题 4** 时，我们利用**熵权法**对第三学年学生的各个指标赋权重，并进行**综合评价**，计算其综合得分。得到每个学生的贫困程度得分后，我们将资助金额进行**线性插值**，得到贫困程度在前 80 位的同学的资助金额分配，最终贫困程度第一的同学分配资助金额 2987 元，排名第 80 的同学分配资助金额 500 元。

总体来看，这种基于大学生餐厅消费数据的隐形资助模型，**既体现了大数据时代的特点，也充分考虑了教育援助的精准性和公平性**。同时，它也**尊重了学生的隐私权**，真正做到了以人为本，关心每一个学生的需要。因此，这种模型具有广阔的应用前景，值得我们深入探讨和研究。

关键词：隐形资助　k-means 聚类　XGBoost　综合评价　线性插值

一、问题重述

1.1 问题背景

在当前社会中，精准资助的重要性日益凸显。特别是对于高校家庭经济困难学生，如何准确识别他们并提供有效的援助，是当前高等教育贫困援助工作的关键问题。这里，我们要探索的是一种新的、具有革命性的隐性资助模型——基于大学生餐厅消费数据的隐形资助模型。

近年来，大数据技术的发展为社会各行业带来了颠覆性的变化，教育领域也不例外。大数据能够为我们提供丰富的信息，从而在各种问题的处理中更加精确、深入。在这个背景下，我们意识到，大学生餐厅的消费数据，也许能够成为揭示学生经济状况的重要窗口。这些数据反映了学生的日常消费行为，而消费行为往往与家庭经济状况密切相关。

基于这种观察，我们决定**利用大数据技术，通过统计和分析大学生的餐厅消费数据，建立一个隐性资助模型**。该模型将能够根据学生的消费数据，评估其经济贫困程度，然后精准地给予援助。更重要的是，这种方式的资助，不需要公开学生的个人信息，也无需进行评比，既能保护家庭经济困难学生的隐私，又能够更公正、更公平地实施教育援助。

1.2 问题重述

1. 数据处理与群体特征分析： 附件 0 提供了学生的性别信息。附件 1-3 提供了学生不同学年的日三餐餐厅消费金额数据记录，附件 4-7 提供了部分同学的饮食种类信息。首先，需要对这些数据进行预处理（如删除不相关数据、缺失值处理、特征提取等）。然后，基于处理后的数据，建立模型来挖掘不同的代表性群体，定量分析这些群体在三个学年中的主要消费行为特征变化规律和饮食种类变化规律。

2. 贫困程度预测： 附件 8 给出了部分同学在第一学年结束后其他方式认定的贫困程度等级。这个等级是粗粒度的，等级 2 是准确的（可能不全），其他等级可能存在一些偏差。需要建立数学模型，根据学生的消费行为（附件 1-3 的数据）预测贫困程度，并补全附件 9。然后，结合第一问的研究结果，预测学生在

第二学年和第三学年的贫困程度，并分析相关变化。

3. 改进贫困程度预测模型： 在第二问的基础上，结合附件 4-7 的饮食种类数据，改进贫困程度预测模型，并比较预测结果的变化。

4. 构建差异化资助额度分配算法： 基于前面对贫困生本质特征的挖掘，构建一个差异化（细粒度）的资助额度分配算法。以第三学年为例，给出具体的分配结果。分配对象是附件 4-7 中涉及的同学，资助总金额为 10 万元，资助人员为 80 名。然后对资助结果的公平性和合理性进行评估。

二、问题分析与模型假设

2.1 问题分析

在开始解题之前，我们首先对数据进行了预处理，以及特征提取。从附近 1-3 中提取出全体学生三年来的消费特征，从附件 4-7 中提取出部分学生的消费金额特征及食物种类特征，用于后续的模型构建。

对问题一，我们采用 k-means 算法，对第一学年的学生进行聚类，然后计算这些群体的三年的消费特征的均值，绘制图表，来体现这些群体三年来的消费特征变化。

对问题二，我们构建 XGBoost 模型，同时使用交叉验证、启发式算法寻优来提高模型的精确度。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学年的贫困程度进行预测。

对问题三，我们将附件 4-7 中提取到的新的特征一起纳入模型的指标体系中，使得模型的预测更加准确。

对问题四，我们利用熵权法对第三学年学生的各个指标赋权重，并进行综合评价，计算其综合得分。得到每个学生的贫困程度得分后，我们将资助金额进行线性插值，得到贫困程度在前 80 位的同学的资助金额分配。

2.2 模型假设

- 假设每位同学的餐厅消费记录都是自己本人的真实消费记录，不考虑为同学带饭、餐卡丢失被盗刷等情况。
- 假设通过熵权法得出的权重能够有效地反映各个特征的重要性。

- 假设学生的贫困程度（综合评价得分）和他们应得的资助金额之间存在线性关系。即学生的贫困程度每增加一个单位，学生应得的资助金额也会增加一个固定的单位。
- 在应用熵权法时，假设所提起到的各个特征是独立的，即它们之间没有相关性。
- 我们假设所有学生得到资助的权利是公平的，即他们得到的资助金额应该基于他们的贫困程度，不考虑其他因素，如学习成绩、行为表现等。
- 假设所提供的数据是完整且准确的。

三、数据预处理及特征提取

3.1 数据预处理

在模型求解之前，为了是模型的预测效果更好，提高模型的准确性。我们首先对数据进行了剔除[1]。

1.在附件 1-3 中，包含了全体学生的三个学年的餐厅消费金额记录，共计 1098 列，包含每日三餐，即共计 366 天。但是我们发现在部分时间段（如寒暑假期间），有大量数据为 0，只有部分同学存在消费记录，我们认为这种情况可能是假期仅有部分同学留校，此时的数据 0 值过多，不具有参考价值。

因此，**倘若某一天消费为 0 的同学占比高达 90%以上，我们便剔除掉这一天的数据**。我们编写 matlab 代码来实现，最终剔除结果如下。

表 3-1 附件 1-3 的数据剔除

学年	剔除（天）	剩余（天）
第一学年	73	293
第二学年	195	171
第三学年	95	271

其中，我们发现第二学年剔除掉的数据过多，我们考虑这可能是由于疫情延迟返校导致的。

2.在附件 4-7 中，给除了部分同学的消费金额、消费食物种类，但是消费食物种类中存在部分缺失值。由于我们在附件 1-3 中已经有了学生的消费金额数据，显然这里消费的食物种类对我们的分析价值更大。因此我们**剔除掉了消费食物种**

类为空值的数据。

3.2 特征提取

附件 1-3 中的数据集过于庞大，都是 5415*1099 的数值矩阵，数据维度过高。相反，在附件 4-7 中，所给的信息又过少，只有学生序号、时间、金额、食物种类四种信息。

因此，对于两种数据，我们都要进行对应的处理，挖掘两个数据集中能够反映出来的学生消费特征，从而更好的构建相关模型。

3.2.1 附件 1-3 全体学生三年消费金额数据特征提取

我们利用 excel 与 matlab，对附件 1-3 中的数据提取到了如下图所示的特征。

图 3-1 附件 1-3 全体学生三年消费金额数据特征提取

单次消费均价，反映了学生每次在餐厅消费的平均金额。这可能反映出学生的经济状况和饮食习惯。我们通过 excel 的 averageif 函数实现。

全年消费次数，显示了学生在一年内在餐厅消费的次数。这可能反映出学生在校的时间以及他们选择在餐厅吃饭的频率。我们通过 excel 的 countif 函数实现。

早、中、晚餐消费次数，这些指标分别显示了学生在一年内吃早餐、中餐和晚餐的次数。这可以反映出学生的作息习惯和饮食选择。我们通过 excel 的 countif 函数实现。

早、中、晚餐消费均价，这些指标分别反映了学生平均在早餐、中餐和晚餐

上花费的金额。这可能反映出学生在每餐的食量以及他们选择的食物类型。我们通过 excel 的 averageif 函数实现。

全年消费波动性：表示学生的年度消费金额的变动程度。这可能反映出学生的消费稳定性，以及可能存在的特殊消费事件。我们通过计算全年的消费金额的**标准差**来反应。

日消费波动性：这个指标表示学生日常消费金额的变动程度。这可能反映出学生日常饮食习惯的稳定性，或者他们是否有不规则的消费行为。我们计算学生每天三餐的消费金额的**标准差，再求每天的标准差的均值**来反应整体的日消费波动性。

最大值：这个指标表示学生在一年内单次消费的最大金额。这可能反映出学生的最高消费能力，或者他们可能参加的一些特殊活动。

极差：表示学生在一年内单次消费的最大值和最小值之间的差距。这反映了学生消费金额的变动范围。

上下四分位点：这些指标表示学生单次消费金额分布的第 25 和第 75 百分位数。这些指标可以提供学生消费金额分布的更多信息，比如大部分消费金额在什么范围内等。

3.2.2 附件 4-7 部分学生消费金额及食物种类特征提取

在提取特征之前，为了得到与早中晚时间段相关的特征，我们首先在 excel 中利用 hour 函数和 if 函数，提取出每条数据的消费时间段。具体规则如下表。

表 3-2 附件 4-7 时间段的定义

时间点	时间段
10 点前	早
10 点-16 点	中
16 点之后	晚

另外，为了分析不同食物的价格差异，我们利用数据透视表统计得到了各个食物的消费均价如下表。

表 3-3 食物消费均价及价格区间

食物种类	均值（分）	食物价格区间
一层花样饼	120	低

一层五谷豆浆	207	低
一层豆豆奶	217	低
一层待定	248	低
层炸鸡饭	855	中
扯面	857	中
粒扒	864	中
致远一层	1265	高
干锅	1272	高
……	……	……（具体结果见附件）

在此之后，我们开始利用 excel 与 matlab 来提取附件 4-7 的数据特征，提取到的特征如下。

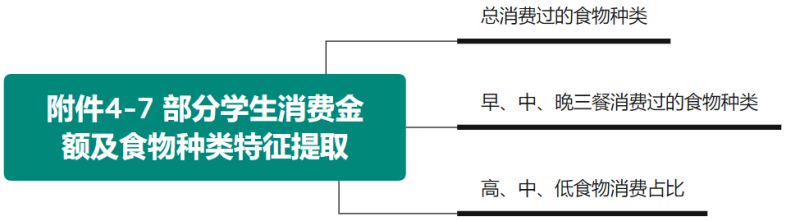


图 3-2 附件 4-7 特征提取

消费过的食物种类：这个特征展示了学生饮食的多样性，也可以反映他们的口味和饮食偏好。例如，如果一个学生消费过的食物种类很少，他可能是一个挑食的人或者他有一些特定的饮食需求或者习惯。如果一个学生消费过的食物种类很多，那他可能是一个愿意尝试新食物的人，或者他的饮食习惯较为均衡。这可以帮助学校了解学生的饮食需求，从而调整餐厅的菜单，提供更多学生喜欢的食物。

早、中、晚餐消费过的食物种类：这个特征可以展示学生在一天中不同时间段的饮食习惯。一些学生可能在早餐时候选择更健康的食物，而在晚餐时候选择更喜欢的食物。或者有些学生在晚餐时候消费的食物种类更多，这可能意味着他们在晚上有更多的社交活动。这可以帮助学校了解学生在一天中的饮食需求。

高、中、低价食物消费占比：这个特征可以反映学生的经济状况和消费习惯。如果一个学生主要消费高价食物，那他可能有较好的经济状况，或者他对食物的质量有较高的要求。如果一个学生主要消费低价食物，那他可能有一些经济压力，或者他比较节省。这可以帮助学校了解学生的经济状况，从而可以用于学校的经

济援助决策。

四、问题一 基于 k-means 聚类模型的学生群体挖掘

4.1 k-means 聚类算法介绍

K-means 是一种迭代的聚类算法，用于将数据点划分为 K 个组或聚类。每个聚类的中心是该聚类中所有点的均值。算法的目标是最小化所有聚类中的点到其聚类中心的距离之和[2]。

以下是 K-means 算法的基本步骤：

1. 随机选择 K 个点作为初始聚类中心。

2. 对于每个数据点，计算其到所有聚类中心的距离，并将其分配给最近的聚类中心。

3. 重新计算每个聚类的中心，新的聚类中心是该聚类中所有点的均值。

4. 重复步骤 2 和 3，直到聚类中心不再发生变化，或者达到预定的迭代次数。

K-means 聚类的目标函数可以表示为：

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \|\mathbf{x}-\boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

其中，C 是聚类的集合， μj 是聚类 Cj 的中心，x 是聚类 Cj 中的点。

这是 K-means 聚类算法的流程图：

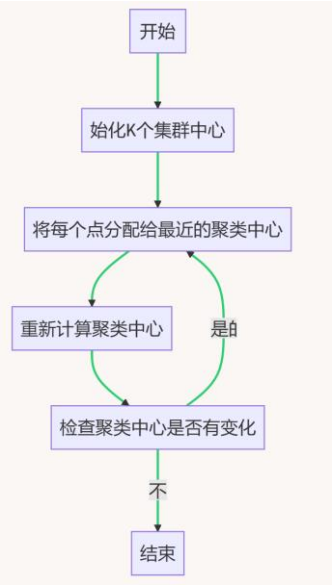


图 4-1 k-means 算法流程图

4.2 第一学年聚类结果

我们将第一学年提取到的全体学生的餐厅消费特征导入 spss 中，进行 k-means 聚类分析，得到距离平方和-聚类个数的折线图如下。

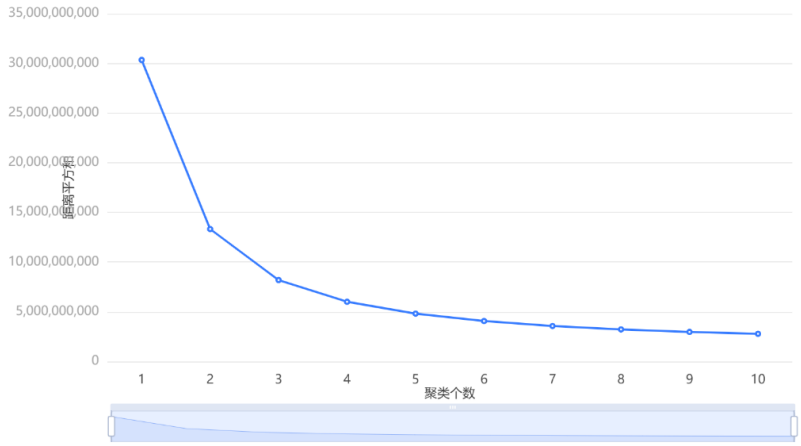


图 4-2 k-means 聚类数对比图（肘部法则）

从图中可以看出，当聚类数大于 3 时，折线图已经变得非常平缓，根据肘部法则，我们最终取聚类数为 3。各个聚类的数目占比如下：

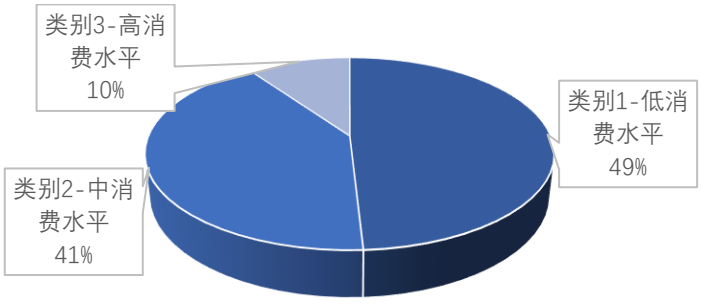


图 4-3 各个聚类占比

从图中可以看出，低消费水平、中消费水平、高消费水平占比分别为 49%、41%、10%。我们统计了三类群体的各个指标特征进行对比，结果如下表。、

可以看出，在所有的特征中，显著性 P 值都小于 0.001，水平上呈现显著性，拒绝原假设，说明各个变量在聚类分析划分的类别之间存在显著性差异。

我们绘制了各个聚类的特征如下图。

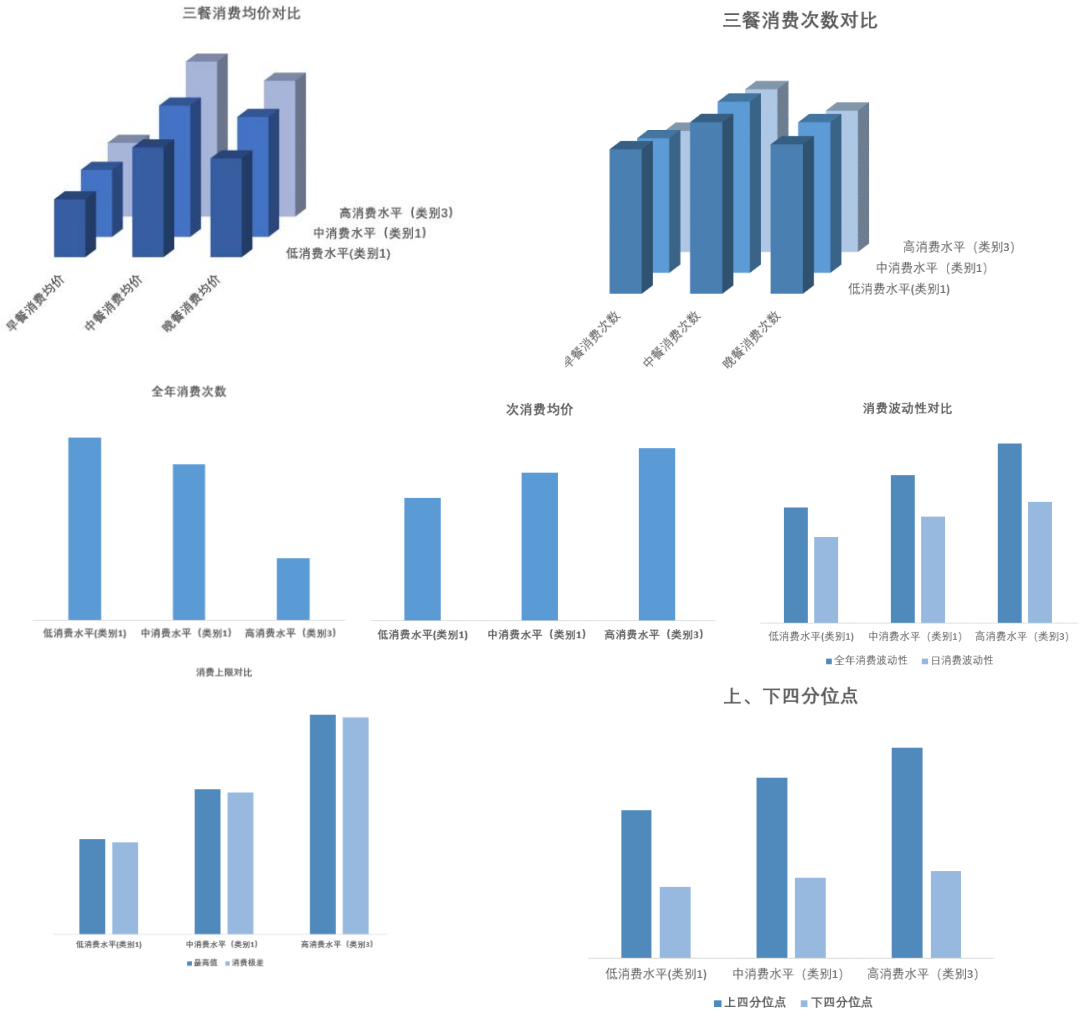


图 4-4　k-means 聚类结果对比

根据各个聚类的特征结果进行分析[3]，我们可以得到以下观察和结论：

1. 消费水平：类别 1 的学生有最低的消费水平，单次消费均价最低（约 918），而类别 3 的学生有最高的消费水平，单次消费均价最高（约 1289）。类别 2 的学生在这两者之间（约 1108）。这表明类别 3 的学生可能经济状况最好，或者他们更愿意在餐厅消费更多。

2. 消费次数：所有类别的学生全年消费次数相差不大，不过类别 1 和类别 2 的学生的消费次数略高于类别 3。这可能意味着类别 3 的学生虽然每次消费更多，但他们在餐厅的消费频率可能略低。

3. 早餐、中餐和晚餐消费次数：所有类别的学生在早餐、中餐和晚餐的消费次数都相似，但类别 1 的学生在早餐的消费次数最多，类别 3 的学生在所有时间

段的消费次数都略低于其他两个类别。这可能意味着类别 1 的学生更倾向于在早餐时间在餐厅用餐。

4. 早餐、中餐和晚餐消费均价：所有类别的学生在早餐、中餐和晚餐的消费均价都与他们的总消费水平一致，即类别 3 的学生的消费均价最高，类别 1 的学生的消费均价最低。

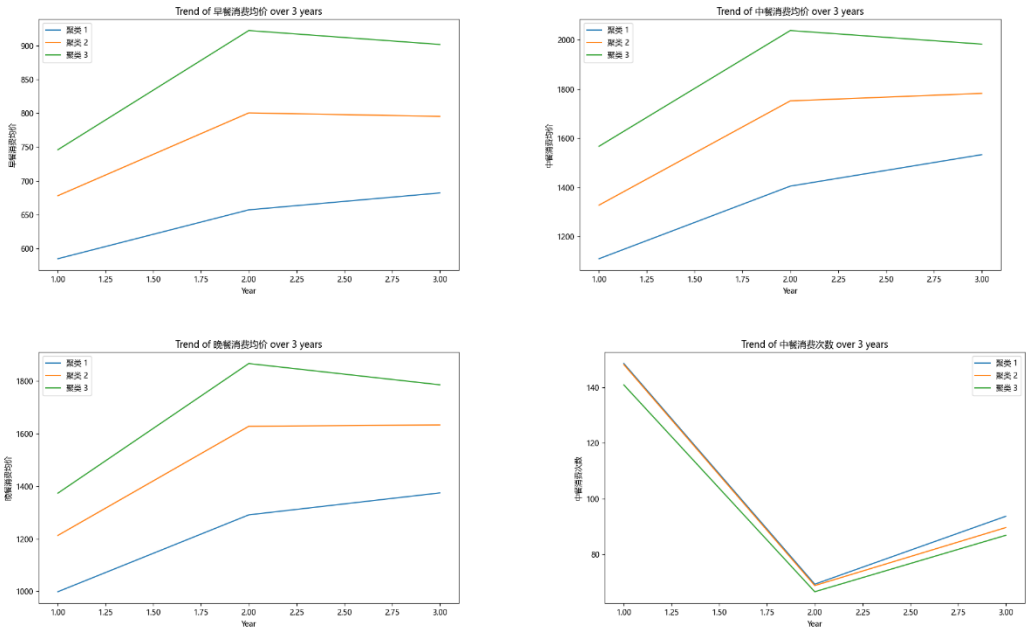
5. 消费波动性：类别 3 的学生全年和日常消费波动性都最大，说明他们的消费行为可能最不稳定，而类别 1 的学生消费波动性最小，消费行为最稳定。

6. 最高值和消费极差：类别 3 的学生有最大的消费最高值和消费极差，可能反映出他们有较大的单次消费能力，或者他们有时候会有特别高的消费行为。类别 1 的学生的消费最高值和消费极差最小，说明他们的消费行为更加均衡，没有特别高的消费峰值。

总的来说，**类别 1 的学生是消费最低但最稳定的群体，他们可能是经济状况较差或者更节省的学生。类别 3 的学生是消费最高但最不稳定的群体，他们可能是经济状况较好或者更愿意消费的学生。类别 2 的学生在这两者之间，消费水平和消费稳定性都中等。**

4. 3 三个学年消费特征变化描述

我们利用 python 计算了低消费水平(类别 1) 、中消费水平（类别 1）、高消费水平（类别 3）三个群体，三年的特征均值，绘制了下列折线图来体现其变化。



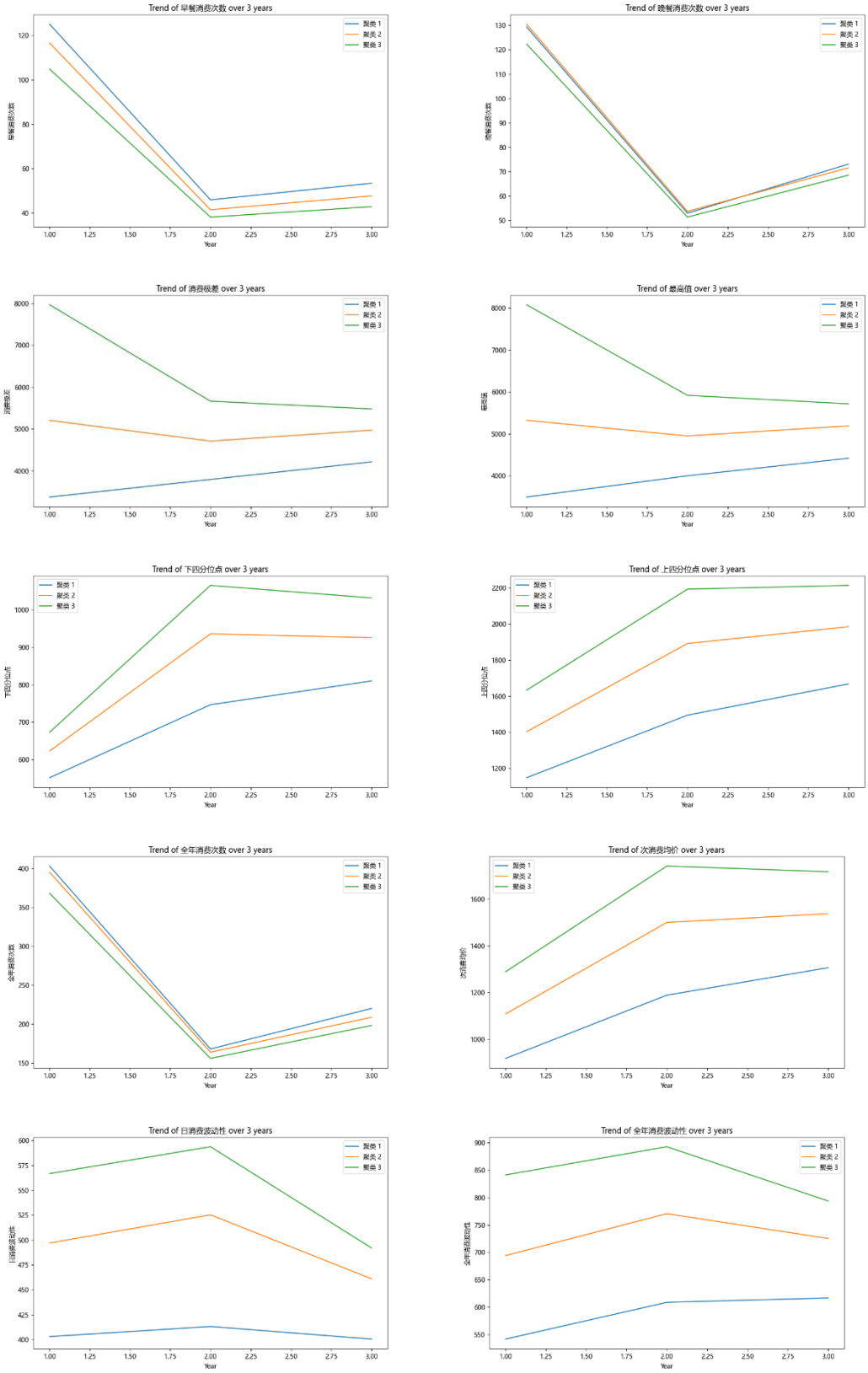


图 4-5　三个聚类三年的特征变化

1. 聚类 1: 这个群体的**学生平均消费价格在三年中有所提升**（从第一学年的 917.95 增长到第三学年的 1305.86）。**全年消费次数在减少**（从 403 次减少到 220

次）。其中，早餐、中餐、晚餐的消费次数均有所下降。就各餐次的消费均价来看，这三年来，早餐、中餐、晚餐的消费均价都有所增长。这可能意味着该群体学生的消费能力提高，但消费次数减少。此外，这个群体的**消费波动性和日消费波动性也在减少**，说明他们的消费习惯更加稳定。

2. 聚类 2: 这个群体的学生的消费均价也在三年中有所提升（从第一学年的 1108.07 增长到第三学年的 1537.12）。全年消费次数在减少（从 395 次减少到 208 次）。早餐、中餐、晚餐的消费次数也都有所下降。这三年来，早餐、中餐、晚餐的消费均价都有所增长。这可能意味着这个群体学生的消费能力提高，但消费次数减少。此外，这个群体的消费波动性和日消费波动性也在减少，说明他们的消费习惯更加稳定。

3. 聚类 3: 这个群体的学生的消费均价也在三年中有所提升（从第一学年的 1288.75 增长到第三学年的 1716.57）。全年消费次数在减少（从 368 次减少到 198 次）。早餐、中餐、晚餐的消费次数也都有所下降。这三年来，早餐、中餐、晚餐的消费均价都有所增长。这可能意味着这个群体学生的消费能力提高，但消费次数减少。此外，这个群体的消费波动性和日消费波动性也在减少，说明他们的消费习惯更加稳定。

总体来看，三个群体的学生在三年的时间里，**其消费均价都在上升，而消费次数在下降**，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。

五、问题二 XGBoost 模型对学生贫困程度的预测

5.1XGBoost 模型构建

5.1.1XGBoost 模型理论基础

XGBoost 采用 Boosting 策略逐步集成决策树。每次集成新树时，XGBoost 通过提高训练样本中高误判率样本的权值，加强新树对高误判率样本的关注和学习，不断修正模型的预测误差，将预测值向真实值推进[4]。其预测函数见式（1）

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

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 将预测值与真实值之间的误差作为模型损失，因而该模型在训练数据和测试数据上的损失差距可用于衡量其泛化能力。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）[32]）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。

$$obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right)$$

其中，$l(y_i, \hat{y}_i)$表示某样本 $\hat{y}_i$ ，与y_{i_2}之间的误差损失；$\sum_{i=1}^{n} l(y_i, \hat{y}_i)$是对 n 个样本的损失求和；$\text{obj} = \sum_{i=1}^{n} l\left(\text{y}_i \hat{y}_i^{\text{t-1}} + + \text{const} + 0(1.)\right.$是对组成模型的 K 棵树的复杂度求和。

XGBoost 对于目标函数的不断优化，其本质上包含了模型为自身获取泛化能力的过程。然而，由于模型无法在训练时预判自身在测试数据上的表现，XGBoost 很难通过优化目标函数最大程度缩减自身在训练数据和测试数据上的损失差距，从而自行获取最强泛化能力。但这为本文探究该模型中能够影响其泛化能力的针对性超参数提供了思路。 XGBoost 将目标函数（式（2））向最小值优化的具体过程如下

以模型集成第 t 棵树为例。根据式（1），此时 XGBoost 对样本的预测值为$\hat{y}_i^{t-1}$为

其中， $f_t(x_i)$为第 t-1棵树对样本的预测。

将式（3）代入并将模型中所有树的复杂度之和表示为前 t-1 棵树的复杂度之和加上第 t 棵树的复杂度，目标函数转化为

$$\text{obj} = \sum_{i=1}^{n} L\left(\text{y}_i \hat{y}_i^{\text{t-1}} + f_t(x_i) + \text{const} + \Omega(f_t)\right. \quad (4)$$

其中，前 t-1 棵树的复杂度之和已知，表示为常数项 $\Omega(f_{\text{t}})$ $const$ ；代表第 t 棵树的复杂度。

城（4）中$\text{y}_{\text{i}} \hat{\text{y}}_{\text{i}}^{\text{t-1}} + f_t(x_i)$项在处展开，并将前 t-1 棵树的预测值与真实值之间的误差 $l(y_i, \hat{y}_i{}^{t-1})$作为已知项移除，得

$$\text{obj}^{(t)} \approx \sum_{i=1}^{n} [\ \text{g}_i f_t + \qquad \sum_{\text{i=1}}^{\text{n}} \text{l}(\ \text{y}_{\text{i}} \hat{\text{y}}_{\text{i}}^{\text{t-1}} + f_t(x_i)$$

$$\frac{1}{2}h_i(f_t(x_i))^2]+\text{const}+\Omega(f_t) \quad (5)$$

其中， g_i和 h_i：分别为损失函数的一阶导数和二阶导数。 $\Omega(f_t)$通过限制树中叶节点数量及其权重控制第 t 棵树的复杂度，如式（6)

$$\Omega(f_t)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 \quad (6)$$

其中， T 代表树中叶节点数量；w，表示树中第 j个叶节点的预测值，即叶权重；和入分别是控制叶节点数和叶权重的超参数。将式（6）代入并移除常数项，目标函数转化为

$$\text{obj}^{(t)}\approx\sum_{i=1}^{n}[\ g_i w_{q(w_i)}+\frac{1}{2}h_i w_{q(w_i)}^2]+\gamma\text{T}+\frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2$$

$$=\sum_{j=1}^{T}[(\sum_{i\in I_j g_i}\quad)\omega_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)\omega_j^2]+\gamma T$$

$$\text{obj}^{(t)}=\sum_{j=1}^{T}[\ G_j\omega_j+\frac{1}{2}(H_j+\lambda)\omega_j^2]+\gamma T \quad (8)$$

其中， $\omega_{q(x_i)}$为样本经过第 t 棵树 的预测值；G_j，和 H_j ，分别是所有样本在 f_t的 j节点上g_i，与 h_i ，的和。

将目标函数向最小值优化，对式（8）求导使其得 0，得第 t 棵树的叶节点 j对应的最优预测值为

$$w_j=\frac{-G_j}{H_j+\lambda} \quad (9)$$

最终将式（9）代入目标函数得

$$\text{obj}=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T \quad (10)$$

5.1.2XGBoost 模型优化

（一）基于性能度量的模型交叉验证

模型交叉验证的基本思想是将数据样本切分为训练集和测试集, 交叉验证是用来观察模型的稳定性的一种方法，将数据划分为 n 份，依次使用其中一份作为测试集，其他 n-1 份作为训练集，通过训练集建立 **XGBoost** 分类模型,利用测试集评估 XGBoost 模型的泛化误差和拟合效果.多次计算模型的精确性来评估模型的平均准确程度。因此用交叉验证 n 次的结果求出的平均值，是对模型效果的一个更好的度量。

对二分类问题, XGBoost 模型分类预测结果与样本集真实分类结果比较, 有四种情况,即(模型预测为真, 真实结果为真)、 (模型预测为真, 真实结果为假) 、(模型预测为假, 真实结果为真) 、 (模型预测为假, 真实结果为假), 这四种情况分别定义为真正例 (TP) 、假正例 (FP) 、真反例 (TN) 、假反例 (FN), 如表 1 所示.准确率 P 和召回率 R 分别定义为:

$$P=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

表 5-2 XGBoost 分类结果情况

真实情况	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

F1 分数(F1 Score) 是统计学中用来衡量分类模型精确度的一种指标.它同时兼顾了分类模型的准确率和召回率, 模型的优劣可以根据 F1 分数大小来判别,而 F1 分数定义为:

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

(二)启发式算法优化模型参数

启发式算法（heuristic algorithm)是相对于最优化算法提出的。一个问题的最优算法求得该问题每个实例的最优解。启发式算法以仿自然体算法为主，主要有遗传算法，蚁群算法、模拟退火法、神经网络等。通过查阅资料发现，树的最大深度（max_depth），叶子节点含有的最少样本（min_samples_leaf），当前节点允许分裂的最小样本数（min_samples_split），基学习器数量（n_estimators），这四个参数对于缓解 XGBoost 模型的过拟合问题，提升模型在测试集上预测表现有显著的帮助（如表 ），因此我们**主要使用遗传算法寻**找这四个参数的最优设置。

5. 2 对附件 9 部分同学的贫困等级预测

我们利用 vlookup 函数，从我们已经提取过的附件 1-3 中的特征数据中，找到附件 8 中部分同学的第一学年的各个特征数据。同时，我们也在附件 9 中利用 vlookup 函数从附件 1-3 中的特征数据找到附件 9 部分同学的特征。

紧接着我们利用附件 8 的数据，已知各个指标的特征数据以及贫困等级，训练 XGBoost 模型，并且使用启发式算法进行寻优，得到如下最优参数：**学习率: 0.2，最大深度: 2，弱学习器的数量: 50。**

各个指标的特征重要性如下图所示。

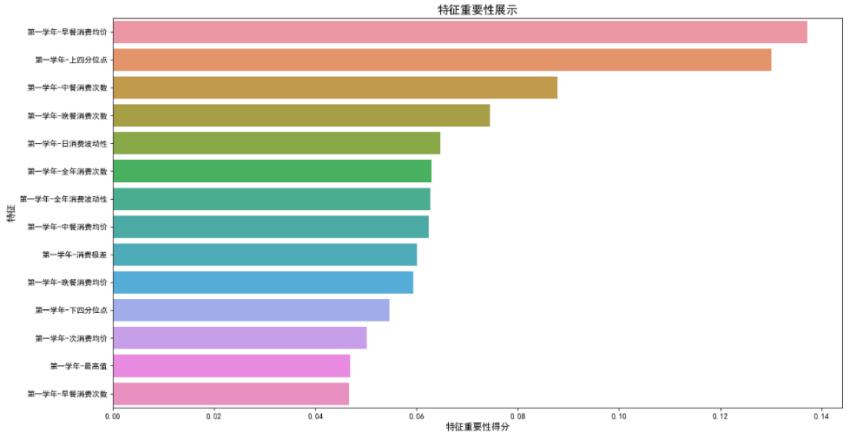


图 5-1 XGBoost 模型的特征重要性

模型的评价指标如下，数据中有三个类别，分别标记为 0，1 和 2。

精确率（Precision）：这是模型预测为某类的样本中，真正为该类的比例。例如，对于类别 0，模型预测为类别 0 的样本中有 76%真的是类别 0。

召回率（Recall）：这是模型预测出的某类样本占该类所有样本的比例。例如，所有真正为类别 0 的样本中，模型预测出了 100%。

F1 得分（F1 Score）：这是精确率和召回率的调和平均值，用于综合评价精确率和召回率。F1 得分越高，说明模型的精确率和召回率都比较高。

准确率（Accuracy）：这是模型预测正确的样本数占总样本数的比例，也就是模型的总体预测准确率。在这个模型中，总的准确率是 76%。

平均值（Mean）：这是每一列的平均值。这里的平均精确率是 77%，平均召回率是 76%，平均 F1 得分是 66%。

从这个结果看，这个**模型在类别 0 上的表现相对较好，精确率、召回率和 F1 得分都相对较高。**但在类别 1 和 2 上的表现却相对较差，尤其是召回率和 F1 得分非常低，这说明模型对于类别 1 和 2 的识别能力比较弱，有很大的提升空间。可能的原因有很多，可能是这两类的样本数量不足，也可能是这两类的特征不明显，导致模型无法很好地区分。这需要进一步分析数据和模型才能得出具体的结

论。

表 5-4 XGBoost 模型评价指标

	精确率	召回率	F1 得分
0	0.76	1.00	0.86
1	0.88	0.01	0.02
2	0.67	0.00	0.01
准确率			0.76
平均值	0.77	0.76	0.66

最终我们利用得到的模型，对附件 9 的数据进行预测，得到的结果如下表所示。

表 5-5 附件 9 预测结果

序号	贫困程度	次消费均价	全年消费次数	早餐消费次数	中餐消费次数	晚餐消费次数	早餐消费均价	中餐消费均价	晚餐消费均价	……
15	0	1327	172	12	86	74	1450	1213	1441	……
17	0	956	631	168	225	238	633	1042	1102	……
29	0	999	339	115	129	95	620	1171	1224	……
5006	2	550	519	152	192	175	247	817	519	
75	1	702	640	169	238	233	314	913	767	
1256	1	1386	527	73	221	233	460	1597	1476	
……	……	……	……	……	……	……	……	……	……	

5.3 对全体同学第二三学年的贫困等级预测

接下来我们使用训练好的 XGBoost 分类模型对全体学生第二三学年的贫困程度进行预测，得到如下结果。

表 5-6 全体学生三个学年贫困等级预测结果

序号	第一学年贫困程度	第二学年贫困程度	第三学年贫困程度
1	0	0	0
2	0	0	0
3	0	0	0

4	2	0	0
5	0	0	1
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	2	0	0
……	……	……	……

图 5-2 三个学年贫困等级数量

我们发现，模型对于贫困等级“1”、“2”的预测的预测并不准确，这可能是由于在已给出的附件 8 数据中，贫困等级为“1”、“2”数量过少，模型无法得到有效的训练。

接下来，我们在问题三中，增加新的特征，以增加模型的准确度。

六、问题三 XGBoost 模型的改进

在附件 4-7 中，给了我们 300 个学生的消费金额及食物种类的数据，我们已经事先对该数据集进行了特征提取。现在我们将这些特征与附近 1-3 提取到的特征合并，使我们的特征更加完善，从而改善模型，得到的结果如下。

序号	贫困程度	消费均价	早餐消费均价	中餐消费均价	晚餐消费均价	消费过的食物种类	早餐消费过的食物种类	中餐消费过的食物种类	晚餐消费过的食物种类	低价食物消费占比	中价食物消费占比	高价食物消费占比	次消费均价	全年消费次数	早餐消费次数	中餐消费次数	晚餐消费次数	早餐消费均价	中餐消费均价	晚餐消费均价	全年消费波动性	日消费波动性	最高值	消费极差	上四分位点	下四分位点
76	0	543	295	853	771	109	35	87	86	37%	57%	6%	900	777	267	258	252	634	1019	1061	611	457	6940	6820	1050	600
1275	0	529	265	805	621	61	23	56	35	48%	48%	5%	589	584	277	212	95	409	743	769	384	356	2050	1950	800	270
2738	0	821	329	1074	888	83	25	60	63	41%	52%	7%	1342	605	182	217	206	781	1675	1486	915	749	5400	5280	1800	750
2932	0	630	320	938	750	101	22	69	81	37%	52%	11%	1215	634	205	211	218	789	1511	1329	780	574	4400	4250	1600	700
3012	0	647	358	795	987	48	19	25	25	44%	50%	6%	1015	755	255	259	241	398	1320	1340	748	637	4400	4250	1600	350
3186	0	531	193	907	742	90	25	58	64	40%	53%	7%	796	706	240	247	219	447	988	964	550	472	5000	4900	1000	400
3239	0	562	350	853	768	72	35	38	52	40%	56%	4%	907	436	176	147	113	676	1087	1033	596	459	3250	3150	1100	550
3272	0	539	271	807	642	57	22	46	30	53%	44%	4%	986	309	109	117	83	596	1261	1113	612	394	3570	3540	1260	500
3287	0	825	531	1088	1059	106	42	77	84	42%	51%	8%	1414	576	210	208	158	959	1654	1704	951	779	6650	6500	1850	800
3321	0	970	545	1175	981	51	12	37	29	41%	53%	6%	1322	218	60	80	78	829	1549	1468	725	423	6600	6450	1740	800
3347	0	666	321	878	826	79	35	59	53	42%	51%	8%	1337	385	124	148	113	835	1725	1380	989	689	7800	7680	1610	600
3366	0	778	450	1104	589	93	25	76	68	41%	47%	12%	1010	466	101	200	165	647	1290	893	626	541	3900	3800	1200	650
3432	0	911	445	978	919	96	20	79	64	35%	54%	10%	1071	533	104	226	203	447	1232	1212	671	611	4100	4000	1300	800

图 6-1 附件 4-7 特征与附件 1-3 合并

在附件 4-7 的 301 位学生中，有 249 位的信息在附件 8 中，另外 51 位在附件 9 中。因此我们将合并后的数据分为两部分，我们从附件 8 中提取到这 249 位学生的贫困等级信息，训练 XGBoost 模型，对另外 51 同学的贫困等级进行预测。

我们得到的新的模型参数为：**学习率: 0.1，最大深度: 2，弱学习器的数量: 50。**

各个特征的重要性如下图。

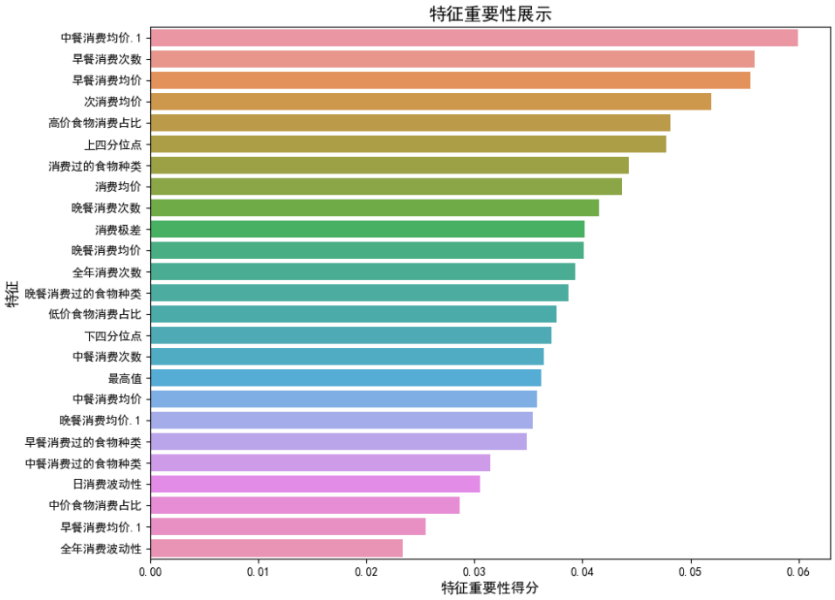


图 6-2 改进后的特征重要性

表 6-1 改进后的 XGBoost 模型评价

	精确率	召回率	F1 得分
0	0.82	1.00	0.90
1	1.00	0.22	**0.36**
2	1.00	0.56	**0.71**
准确率			0.84
平均值	0.94	0.84	0.80

从表中可以看出，在增加新的特征后，模型效果得到了极大的提升，新的模型准确率为 0.84，比之前的 0.76 提升了 10.53%，最重要的是，新的模型对于贫困程度为“1”、“2”的预测准确率得到了大大提升，原本模型对于“1”、“2”的预测 F1 得分仅有 0.02、0.01，改进后达到 0.36 与 0.71，翻了几十倍之多。

七、问题四 基于多指标综合评价的贫困等级综合评价模型

7.1 熵权法-综合评价理论介绍

熵权法（Entropy Method）是一种在多指标决策分析中常用的权重计算方法。它基于信息熵的概念，考虑到各个评价指标的主观和客观因素，将权重分配给各个评价指标[5]。熵权法是一个能够充分反映指标之间信息的科学评价方法，其基本步骤如下：

假设有 m 个评价对象，n 个评价指标，每个评价对象在各个评价指标上的得分构成一个 m×n 的矩阵。评价对象和评价指标的相关性，可以通过以下步骤计算：

1. 数据标准化：由于原始数据的量纲和数值大小可能不一，因此需要进行标准化处理。一般采用 0-1 标准化，公式如下：

Xij = (Xij - min(Xi)) / (max(Xi) - min(Xi))

其中，Xij 表示第 j 个评价对象在第 i 个评价指标上的原始得分，min(Xi)和 max(Xi)分别表示第 i 个评价指标所有评价对象得分的最小值和最大值。

2. 计算各项指标的信息熵：使用以下公式计算第 i 个指标的信息熵：

Ei = -k * Σj(Pij * ln(Pij))

其中，k 是常数，取值为 1/ln(m)；Pij 表示第 j 个评价对象在第 i 个评价指标上的比重，即 Pij = Xij / Σj(Xij)；ln 表示自然对数。

3. 计算各项指标的差异系数：使用以下公式计算第 i 个指标的差异系数：

Di = 1 - Ei

其中，Ei 是第 i 个指标的信息熵。

4. 计算各项指标的权重：使用以下公式计算第 i 个指标的权重：

Wi = Di / Σi(Di)

其中，Di 是第 i 个指标的差异系数。

5. 计算综合评价值：根据各项指标的权重，计算每个评价对象的综合评价值，公式如下：

Sj = Σi(Wi * Xij)

其中，Wi 是第 i 个指标的权重，Xij 是第 j 个评价对象在第 i 个评价指标上的标准化得分。

这就是熵权法的基本理论和步骤。这种方法的优点是，它不仅考虑到了各项指标的主观权重，而且还考虑到了各项指标的客观权重，从而得出了比较科学、公正的评价结果。

7.2 熵权法-综合评价对学生进行贫困程度评分

首先我们需要先判定各个指标对贫困程度影响的正负性。我们以附件 8 中已

有的学生数据为依据，计算了附件 8 中贫困程度为 0、1、2 的三个学生群体的特征如下表。

表 7-1 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）	0	1	2
次消费均价	1056	981	944
全年消费次数	383	427	440
早餐消费次数	116	129	131
中餐消费次数	142	159	167
晚餐消费次数	125	139	142
早餐消费均价	662	589	555
中餐消费均价	1266	1201	1151
晚餐消费均价	1149	1059	1029
全年消费波动性	638	623	611
日消费波动性	457	458	454
最高值	4689	4786	4559
消费极差	4570	4673	4450
上四分位点	1331	1232	1183
下四分位点	608	557	537

从表中我们可以看出，次消费均价、早中晚餐消费均价、全年消费波动性、日消费波动性、最高值、消费极差、上下四分位点等 10 个指标都是负向的，即贫困等级越高，这些指标的值越低，其余指标则都是正向的。最终我们得到的评价分数越高，说明学生贫困等级越高。

表 7-2 综合评价指标的正负向划分

	指标
正向	全年消费次数、早中晚餐消费次数
负向	次消费均价、早中晚餐消费均价、全年消费波动性、日消费波动性、最高值、消费极差、上下四分位点

紧接着我们对指标进行标准化处理，再利用熵权法计算各个指标权重。结果如下。

表 7-3 指标权重

熵权法			
项	信息熵值 e	信息效用值 d	权重(%)
全年消费次数	0.976	0.024	16.63
早餐消费次数	0.949	0.051	36
中餐消费次数	0.975	0.025	17.521
晚餐消费次数	0.969	0.031	21.626
次消费均价	0.998	0.002	1.074
早餐消费均价	1	0	0.272
中餐消费均价	0.999	0.001	1.024
晚餐消费均价	0.999	0.001	0.908
全年消费波动性	0.999	0.001	1.022
日消费波动性	0.998	0.002	1.526
最高值	0.999	0.001	0.51
消费极差	0.999	0.001	0.511
上四分位点	0.999	0.001	0.734
下四分位点	0.999	0.001	0.641

最终我们得到了全体学生第三学年的贫困程度得分，这里以得分降序排序，完整表格见附件。

表 7-4 学生贫困程度评分

序号	综合评价	排名
1597	0.9761	1
2914	0.9389	2
2852	0.9321	3
4410	0.9228	4
921	0.9222	5
2033	0.9218	6
4469	0.8916	7
5365	0.8844	8
……	……	

7.3 利用线性插值对 80 位同学进行金额分配

为了确保 80 位同学都能够获得一定的资助，我们首先对每位同学确定 500 元的资助额度，总计 500*80=40000 元，剩余 60000 元，紧接着我们采用线性插值的方法，根据学生贫困程度评分进行将剩余的金额进行线性插值，得到每位同学能够得到的剩余一部分资助金额。

线性插值是一种数值计算方法，它用于在两个已知值之间预测未知值。我们使用线性插值的方法能够确保**资助金额的分配是基于学生的贫困程度评分**的，评分越高的学生获得的资助金额越多。这个方法可以保证资助金额的分配是公平的，并且与学生的实际需要相符。

最终得到的结果如下表。贫困程度排名第一的同学资助额度为 3682 元，第 80 的同学资助额度为 200 元，且满足总资助金额为 100000 元。

表 7-5 资助金额分配表

序号	综合评价	排名	资助金额
1597	0.9761	1	2987
2914	0.9389	2	2645
2852	0.9321	3	2583
4410	0.9228	4	2497
……	……	……	……
3765	0.7061	79	506
4251	0.7055	80	500

7.4 资助方案合理性分析

我们计算出贫困程度评分前 80 位的各个指标特征的均值，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，结果如下。

可以看出，所选出的 80 位学生各个指标基本都基于贫困程度为 1、2 的群体之间。说明评价是合理的。

表 7-6 所选出 80 位同学与附件 8 的贫困群体特征对比

指标	贫困程度=0	贫困程度=1	贫困评分前 80 位同学	贫困程度=2
次消费均价	1056	981	969	944
全年消费次数	383	427	610	440
早餐消费次数	116	129	199	131
中餐消费次数	142	159	211	167
晚餐消费次数	125	139	200	142
早餐消费均价	662	589	545	555
中餐消费均价	1266	1201	1235	1151
晚餐消费均价	1149	1059	1108	1029
全年消费波动性	638	623	621	611
日消费波动性	457	458	503	454
最高值	4689	4786	4085	4559
消费极差	4570	4673	3955	4450
上四分位点	1331	1232	1259	1183
下四分位点	608	557	565	537

通过熵权法进行的综合评价，能够全面地考虑各个影响学生贫困程度的因素，为每位学生生成一个全面的贫困评分。这种评分方法对比只考虑单一因素的评分方法，能够更全面地反映学生的贫困情况，减少因为忽略某些因素导致的评价误差。线性插值算法则为这个评分系统提供了一种灵活的分配策略。它会根据每个学生的贫困程度评分，为每位学生分配不同的资助金额。评分较高的学生（贫困程度较高）将获得更多的资助，而评分较低的学生（贫困程度较低）将获得较少的资助。此外，我们在分配资助金额时也引入了最小资助额度的设定，确保每个获得资助的学生都能获得一定的资助金额。这种设计旨在避免资助金额过低而无法对学生产生实质性帮助的情况。综上，这种资助金额分配方法充分考虑了学生的贫困程度，并且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，符合教育资助的目标。

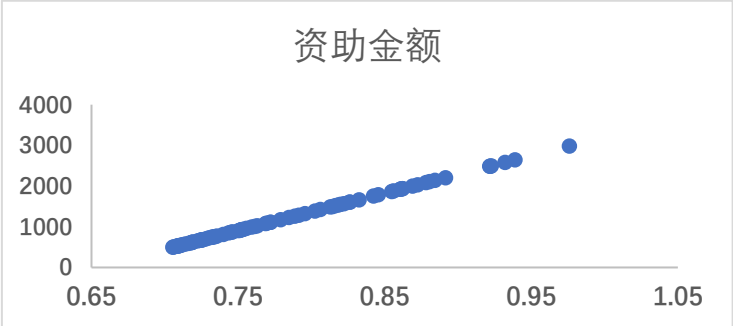


图 7-1 资助金额-贫困程度评分

八、模型的评价与改进

8.1 模型优点

- ✓ 在进行特征提取时，我们尽可能充分的挖掘数据特征，对附件 1-3、4-7 两个数据集提取出共计 20 多个指标，使得我们的模型准确率很高。
- ✓ 使用聚类算法
- ✓ 在进行 XGBoost 模型构建时，我们采用了交叉验证、启发式算法寻优等多种方法，提高模型的准确度。
- ✓ 使用熵权法综合评价+线性插值的资助金额分配方法充分考虑了学生的贫困程度，并且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，符合教育资助的目标。

8.2 模型缺点

- 附件 8 中关于贫困等级为 1、2 的数据较少，导致 XGBoost 模型对于这两类贫困等级识别不准确；
- 由于附件 4-7 只给出了 300 位同学的消费食物种类，并没有给出全体的，因此，即使利用这 300 位同学的数据训练出来的模型大大提升，但是由于没有其他同学的这些指标数据，导致无法利用这个优化后的模型进行预测。
- 尽管学生的餐厅消费数据能够很大程度上反映学生的贫困程度，但是仍然不能全面的反映，例如还可以通过学生的购物水平、所使用的电子产品价格等等来反映。因此我们通过餐厅消费数据所构建出来的预测贫困等级的模型，仍然有提升的空间。

8.3 模型改进

- 收集更多的贫困等级为 1、2 的学生数据：更多的数据可以帮助模型更准确地理解和识别这两类学生的特征。
- 拓宽特征来源：我们现在的模型主要基于学生的餐厅消费数据，而这并不能全面反映学生的贫困程度。在未来，我们可以考虑引入更多其他类型的数据，如学生的购物数据、电子产品使用数据等，以便更全面地评估学生的贫困程度。
- 对分配资助金额的算法进行优化：当前我们使用线性插值的方法进行资助金额分配，但实际情况可能更复杂，可能需要更复杂的算法来确保资助金额的公平分配。例如，可以考虑引入非线性插值方法，或者基于优化理论的方法，以确保更公平的分配结果。
- 在使用熵权法综合评价时，可能还需要考虑如何更好地确定各项指标的权重。目前，权重是根据数据的信息熵来确定的，这并不一定能完全反映各项指标对评价结果的真实影响。未来，我们可以考虑使用基于专家经验或者基于机器学习的权重确定方法。

参考文献

[1]高建平.高校一卡通系统数据分析的设计与实现[J].信息科学,2020:35-36.

[2]殷雨晨.我国东部地区居民消费结构聚类分析[J].经济与管理科学,2022:81-83.

[3]纪松江,陈豪,唐博浩, 等.基于大学生校园卡消费数据特征分析学生消费能力[J].2021:25-26.

[4]张 晟,刘长江,苗凯尧, 等.基于优化 XGBoost 的运动效果量化评估与分析研究[J].社会科学Ⅱ辑,2022:85-88.

[5]黄莲琴,明玥,梁 晨.基于熵权 TOPSIS 法的公司绿色治理观测指标与评价研究[J].工程科技Ⅱ辑 2023:98-105.

附录

```
% 假设 data 是一个 5415x1098 的数组，保存了所有的数据
studentCount = 5415; % 学生数量
dayCount = 366; % 天数数量

% 初始化一个空的逻辑向量来保存需要删除的列的索引
deleteColumns = false(1, dayCount * 3);

% 遍历每一天
for i = 1:3:(dayCount * 3)
    % 计算每一天三餐消费总额为 0 的学生数量
    zeroCounter = sum(sum(data(:, i:i+2), 2) == 0);

    % 如果三餐消费总额为 0 的学生数量超过总学生数量的 90%，标记这一天的数据列需要被删除
    if zeroCounter / studentCount > 0.9
        deleteColumns(i:i+2) = true;
    end
end

% 删除需要被删除的列
data(:, deleteColumns) = [];

% 计算并显示已删除的列
deletedColumns = find(deleteColumns);
disp('Deleted columns: ');
disp(deletedColumns


% 提取消费数据
consumptionData = data(:, 1:879); % 提取第 1 列到第 879 列的数据

% 计算每位学生每天三餐的标准差
dailyStd = zeros(size(consumptionData, 1), 293); % 存储每位学生每天三餐的标准差
for i = 1:size(consumptionData, 1)
    studentData = reshape(consumptionData(i, :), 3, 293); % 将每位学生的消费数据重新排列为 3 行 293 列
    dailyStd(i, :) = std(studentData); % 计算每天三餐的标准差
end

% 计算每位学生每天标准差的平均值
```

```
studentAvgStd = mean(dailyStd, 2); % 沿第二维计算平均值，得到每位学生每天标准差的平均值

% 显示结果
disp(dailyStd); % 显示每位学生每天三餐的标准差矩阵
disp(studentAvgStd); % 显示每位学生每天标准差的平均值

k=7.86*10^-4
b=-0.063
co60=k.*co60ljz-b

cs137=k.*cs137ljz-b



% 导入数据
% 假设数据已经在名为 'myData' 的二维矩阵中
% myData = importdata('your_file_name');

% 初始化结果矩阵
maxValues = zeros(size(myData,1), 1);
rangeValues = zeros(size(myData,1), 1);
q1Values = zeros(size(myData,1), 1);
q3Values = zeros(size(myData,1), 1);

% 对于 myData 中的每一行
for i = 1:size(myData,1)
    row = myData(i, :);

    % 计算最大值
    maxValues(i) = max(row);

    % 计算除去 0 之外的极差
    rowWithoutZeros = row(row ~= 0);
    if isempty(rowWithoutZeros)
        rangeValues(i) = 0;
    else
        rangeValues(i) = range(rowWithoutZeros);
    end

    % 计算除去 0 之外的上四分位点和下四分位点
    if isempty(rowWithoutZeros)
        q1Values(i) = 0;
        q3Values(i) = 0;
    else
```

```
        q1Values(i) = quantile(rowWithoutZeros, 0.25);
        q3Values(i) = quantile(rowWithoutZeros, 0.75);
    end
end

% 输出结果
disp('Max values:')
disp(maxValues)
disp('Range values:')
disp(rangeValues)
disp('Q1 values:')
disp(q1Values)
disp('Q3 values:')
disp(q3Values)



import pandas as pd
import numpy as np

# 读取 Excel 文件
df = pd.read_excel("D:\\6.xlsx")

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

# 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]

    # 计算每个指标的均值
    for year in ['第一学年-', '第二学年-', '第三学年-']:
        for column in df.columns:
            if year in column:
                mean_value = group_df[column].mean()
                results.loc[group, f"{year}{column.replace(year, '')}"] =
mean_value

# 将结果导出到新的 Excel 文件
results.to_excel("D:\\results.xlsx")

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

# 加载数据
data_train = pd.read_excel("D:\\附件 8 已知贫困标签.xlsx")
data_predict = pd.read_excel("D:\\附件 9 问题 2 待补全标签数据 - 副本.xlsx")

# 分割特征和标签
X_train = data_train.iloc[:, 2:]
y_train = data_train.iloc[:, 1]
X_test = data_predict.iloc[:, 2:]

# 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')

# 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
y_pred = grid_search.predict(X_test)

# 保存预测结果到原数据中
data_predict.iloc[:, 1] = y_pred
data_predict.to_excel("D:\\附件 9 问题 2 待补全标签数据 - 副本.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_
sns.barplot(x=feature_importances, y=X_train.columns)
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Visualizing Important Features")
plt.show()
```