基于餐厅消费数据的隐形资助研究

摘要

本文探讨了一种**基于大学生餐厅消费数据的隐性资助模型**，以解决当前高等教育贫困援助工作中如何准确识别和援助高校家庭经济困难学生的问题。利用大数据技术，通过统计和分析学生的餐厅消费数据，我们建立了 K-Means 和随机森林 (Random Forest) 数学模型, 并基于 Topsis 算法，可以根据学生的消费数据评估他们的经济贫困程度，并提供精准的援助，为教育精准扶贫事业提供助力。

针对数据处理，我们针对题目所给的附录 1-3 中部分学生不同学年的日三餐餐厅消费金额数据与附录 4-7 部分同学的饮食种类信息进行坏值剔除，特征提取与归一化处理, 便于后续模型拟合及预测。

针对问题一，我们选择使用 K-means 聚类方法进行模型构建，并利用附件 1-3 中部分学生不同学年的日三餐餐厅消费金额数据（取第一学年）进行模型训练。基于肘部法则初步确定聚类数 k 为 3，进一步将该部分学生分类。根据他们的每次消费均值将三个群体区分为低消费群体（类 1）、高消费群体（类 2）以及中等消费群体（类 3），并进一步结合他们的特征异同、三年消费行为的变化进行分析。

针对问题二，我们基于附件 1-3 数据以及附件 8 的贫困程度标签，选用随机森林模型进行监督学习模型的训练，以此来对附件 9 中的学生以及第二三年的全体同学进行贫困程度预测，模型在准确率（accuracy）、查准率（precision）和召回率（recall）方面都有较好表现。

针对问题三，我们利用附件 4-7 中的数据提高了样本数据的维度，进一步优化了问题二构建的分类模型，使得分类器对一般贫困、特别贫困的学生分类方面具有更好的表现。

针对问题四，我们使用问题一种所得特征重要性作为权值、以全体学生第三学年为数据集、结合 TOPSIS 法，进一步增强了评价模型的客观性与科学性，最后采用线性插值的方法分配对贫困程度不同学生的资助金额，实现了公平性的目的。

关键字： K-Means 聚类算法　决策树　 Random Forest 随机森林　TOPSIS 算法

一、 问题的背景和重述

1.1 问题背景

伴随国家精准扶贫政策的不断推进，针对大学贫困生的精确判定被提上重要日程。基于当下大数据互联时代的特性，进行隐形资助是一个重要的研究方向，对切实完善精准资助手段，协助高校开展贫困资助工作有着极其深远的意义。因此，我们将通过探索基于餐厅消费数据的隐形资助研究，以寻求建立富有创新性和实用性的隐形资助模型。

1.2 问题重述

考虑到学生通过餐厅消费相关信息，如消费金额、消费品类与消费次数等具有高信息熵的特性，我们可以通过这些信息建立模型间接推断学生的经济状况。

问题一：附件 0 为性别信息，附件 1-3 为该管理部门仿真的学生在不同学年每日在餐厅三餐的消费数据，而附件 4-7 为其中部分学生的饮食种类信息。题目的要求是要对这部分数据进行**预处理**，然后针对这些信息建立合适的模型**挖掘不同代表性群体**，并从**定量**的角度分析该群体三个学年的主要**消费行为特征变化**规律和**饮食种类变化**规律。

问题二：除了第一问中用到的附件 0-7 的信息，附件 8 包含了部分同学的贫困程度认定等级，不过由于某些原因只有等级 2 的评定是准确的，而且可能不全，其他等级的认定存在少量偏差。题目的要求是，**依据消费行为建立数学模型预测贫困程度，以此补全附件 9**。此外，还需要我们结合第一题的研究结果来**预测往后两学年的贫困程度隐形认定等级，分析相关变化**。

问题三：要求在第二问基础上，结合饮食种类数据改进预测模型，**比较分析相关同学的预测结果变化情况**。

问题四：基于以上内容对贫困生本质特征进行挖掘，**构建差异化资助额度分配算法**，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

本文主要解决四个问题。以下是对每个问题的分析：

针对问题一：在这个问题中，我们选择使用 K-means 聚类方法进行模型构建。为了训练该模型，我们使用了附件 1-3 中部分学生在不同学年的日三餐餐厅消费金额数据

（只选择了第一学年的数据）。通过应用肘部法则，初步确定聚类数 k，并将这部分学生进一步分类。通过计算每个学生的平均消费金额，他们将这些群体划分为不同的类。便于进一步分析这些群体的特征差异以及三年内的消费行为变化。

针对问题二：在这个问题中，我们使用了附件 1-3 的数据以及附件 8 中的贫困程度标签。我们选择了随机森林模型进行监督学习模型的训练，以此来预测附件 9 中学生以及第二三年的全体同学的贫困程度。并通过准确率（accuracy）、查准率（precision）和召回率（recall）等方面对模型表现进行评估。

针对问题三：为了解决问题三，我们决定利用附件 4-7 中的数据扩展了样本数据的维度，并进一步优化问题二中构建的分类模型。以期分类器在对贫困学生的识别上表现更优秀。

针对问题四：对于问题四，我们使用问题一中获得的特征重要性作为权值，并以全体学生第三学年的数据集为基础。结合 TOPSIS 法，我们进一步增强了评价模型的客观性和科学性。最后采用线性插值的方法为不同贫困程度的学生分配资助金额，以实现公平性的目标。

三、 模型的假设与约定

- 假设附录中所给出的数据全部真实有效, 并且关系上具有完整性，如附件 1-3 与附件 4-7 数据之间的依赖关系。
- 假设食堂消费可以反映出学生真实的经济状况, 不考虑个人因素导致的特例。
- 假设在随机森林算法中，每一棵决策树最终的结果被采纳的可能性对等。
- 假设在问题四中每个学生收到资助可能性平权，模型具有公平性。且不考虑放弃接受资助等特例。

四、 数据预处理

为了便于后续的分析和建模工作，我们需要通过预处理来进行数据的清洗和准备。这里我们使用了 **EXCEL** 工具来对数据集中低信息熵的部分进行去除, 降低噪声干扰。并从中提取原始数据的特征及将数据标准化，提高后续分析和建模的准确性和可靠性，为后续的任务奠定良好的数据基础。

4.1 针对” 该组学生不同学年的日三餐餐厅消费金额数据记录”(附件 1-3) 中的数据。

我们可以发现附件 1-3 的数据集较大，也是我们进行机器学习的主要数据来源，需要进行妥善且充足的处理。

下面我们从**坏值剔除、特征提取以及归一化处理**三个方面来对” 该组学生不同学年的日三餐餐厅消费金额数据记录”(附件 1-3) 中的数据进行处理。

4.1.1 坏值剔除

结合现实情况可知，学生在寒暑假离校时，数据集中将出现大量’0’ 值，这些数据对于建模和分析没有意义, 予以去除。**结合我校相关情况以及网络上资料查询，规定当 95% 的数据点为‘0’时，该天数据记录为无效**。

表 1　坏值剔除结果

年份	有效样本天数	去除坏值天数
NO.1	298	68
NO.2	172	194
NO.3	280	86

需要注意的是，在数据处理过程中我们发现了一些具有不合法特征的**异常点**，如无早餐使用记录，标准差无法计算等等，这里进行部分列举：

不合理人员名单

- 561
- 691
- …
- 4999

针对这部分数据，我们进行删除处理。

4.1.2 特征提取

经过上述数据处理，我们下一步对附件 1-3 的内容进行特征提取。

利用 **EXCEL 和 MATLAB** 工具，获得包括**三餐及每餐的消费均价**特征, **三餐和每餐消费次数**特征，**全年消费标准差**。

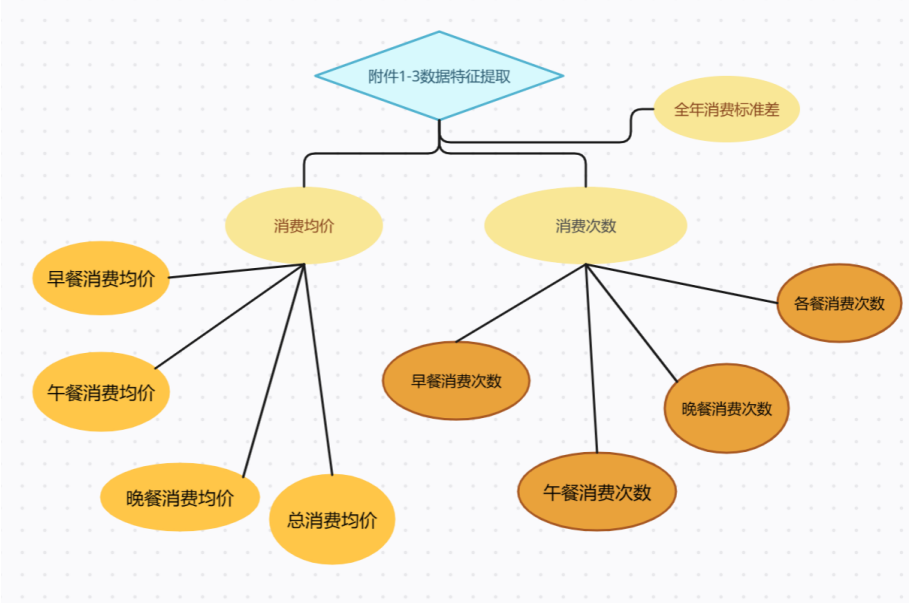

图 1　不同学年的日三餐餐厅消费金额数据记录（附件 1-3）数据特征提取

4.2 针对“部分同学的饮食种类信息”(附件 4-7) 中的数据

“部分同学的饮食种类信息”(附件 4-7) 中的数据具有缺失较多的情况，可能影响到建模的准确性，但同时也存在一些无意义的点的情况。因此我们采取挖除的方法进行处理。通过对**特征时间段**和**食物价格**进行分层，依据食物均价划分种类，得到三餐及每天消费的不同食物种类次数的数据特征。

4.2.1 坏点剔除

针对表格中部分数据存在**仅有消费金额无食物种类**的情况，我们予以剔除，然后再进行特征处理。此外，我们推测这部分校园卡消费可能是水房打水消费，澡堂刷卡消费等其他不属于食堂消费的部分，对模型建立没有意义，也不需要进行统计。

4.2.2 特征提取

类似于附件 1-3 的处理，我们同样利用 **EXCEL 和 MATLAB** 对附件 4-7 进行数据特征提取。

4.3 归一化处理 (Normalization)

我们这里对数据进行线性归一化处理, 也称 min-max 标准化。对原始数据的线性变换，将结果值映射到 [0,1] 之间。转换函数如下：

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

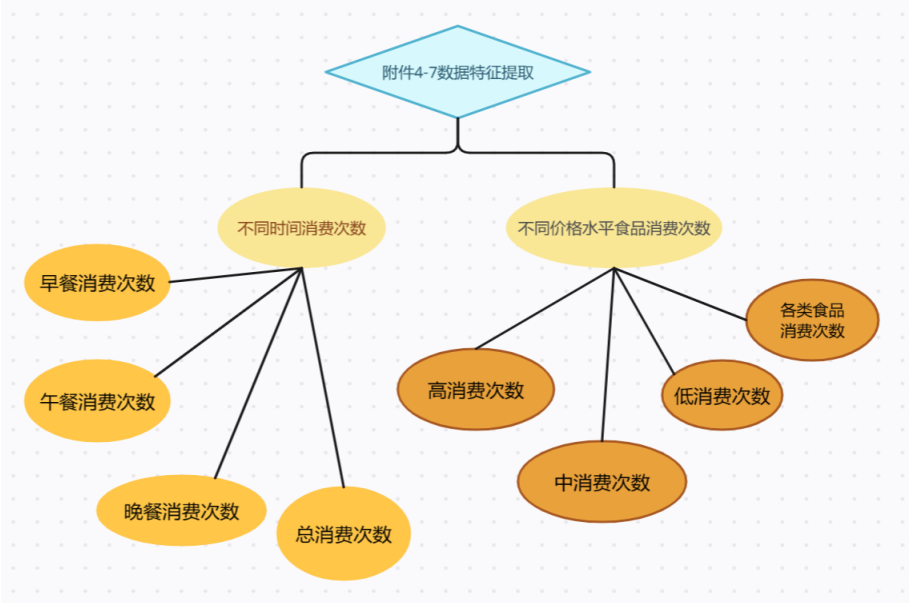

图 2　部分同学的饮食种类信息（附件 4-7）数据特征提取

机器学习中的关键一步就在于数据特征处理，其中对特征数据进行归一化处理至关重要。它可以确保数据点没有因为特征的基本性质而产生较大差异，即**确保数据处于同一数量级（同一量纲）**，提高不同特征数据的可比性。

五、 基于 K-Means 与决策树复合挖掘代表性群体及变化规律

5.1 K-Means 聚类算法

K-Means Clustering 是一种无监督学习算法，用于解决机器学习或数据科学中的**聚类问题** [1]，将未标记的数据集分组为不同的簇。

k-means 聚类算法主要执行两个任务：

通过迭代过程确定 K 个中心点或质心的最佳值。

将每个数据点分配给其最近的 k 中心。那些靠近特定 k 中心的数据点创建一个簇。因此，每个簇都有具有某些共性的数据点，并且远离其他簇。

下面是 K-means 算法的步骤：

- 选择数字 K 来决定簇的数量。
- 选择随机 K 点或质心。（它可以是输入数据集中的其他数据）。
- 将每个数据点分配给它们最近的质心，这将形成预定义的 K 个簇。
- 计算方差并为每个簇放置一个新的质心。
- 重复第三步，这意味着将每个数据点重新分配给每个簇的新的最近的质心。
- 如果发生任何重新分配，则转至步骤 4，否则转至完成。

- 模型准备就绪。

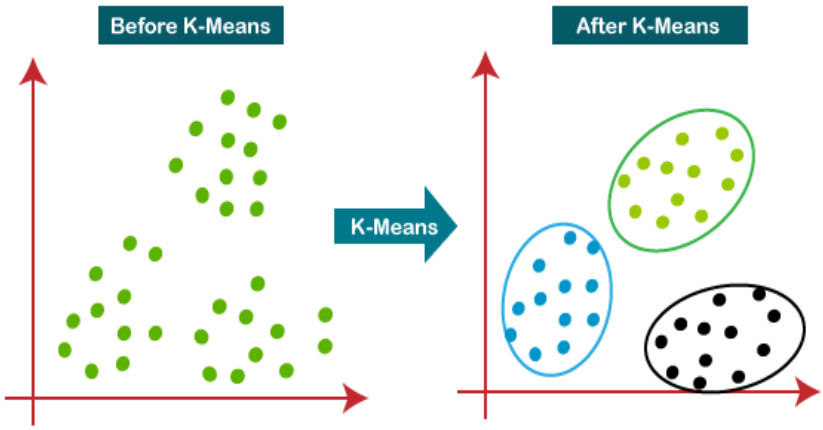

图 3　K-means 聚类算法的工作原理

Elbow 方法是查找最佳簇数最流行的方法之一。
WCSS 值的概念。其代表**簇内平方和**，它定义簇内的总变化。
由针对三个簇的 WCSS 计算方法可得 K-means 的目标函数为：

$$WCSS = \sum_{P_i \, in \, Cluster1} distance(P_i \, C_1)^2 + \sum_{P_i \, in \, Cluster2} distance(P_i \, C_2)^2 + \sum_{P_i \, in \, Cluster3} distance(P_i \, C_3)^2$$

通过求解 WCSS 值，可以求得聚类系数。通过绘制计算出的聚类系数与聚类数量 K 之间的曲线，可以求得最佳簇数 K。

5.2 决策树

决策树是一种监督学习技术，可用于分类和回归问题，但大多数情况下它更适合解决分类问题。它是一个树形结构的分类器，有两个节点，即**决策节点和叶节点**。决策节点用于做出任何决策并具有多个分支，而叶节点是这些决策的输出并且不包含任何进一步的分支。
以下为决策树的步骤：

- 从根节点（S）开始创建树，其中包含完整的数据集。
- 使用属性选择度量 (ASM) 在数据集中查找最佳属性。
- 将 S 划分为包含最佳属性的可能值的子集。
- 生成决策树节点，其中包含最佳属性。

- 使用步骤 -3 中创建的数据集的子集递归地创建新的决策树。继续此过程，直到达到无法进一步对节点进行分类的阶段，并将最终节点称为叶节点。

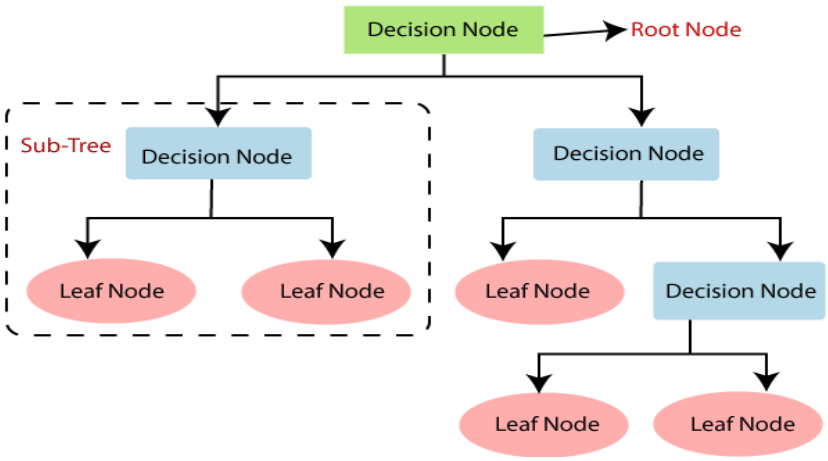

图 4　Decision Tree 工作原理

在实现决策树时，主要的问题是如何为根节点和子节点选择最佳属性。因此，为了解决此类问题，有一种技术称为属性选择措施或 ASM。通过这种测量，我们可以轻松地为树的节点选择最佳属性。ASM 有两种流行的技术——信息增益 (Information Gain) 和基尼系数 (Gini Index), 在第一问中我们使用**基尼系数 (Gini Index)** 基尼指数是在 CART（分类和回归树）算法中创建决策树时使用的杂质或纯度的度量。有下列公式：

$$\text{GINI}\quad \text{INDEX}=1-\sum_j P_j^2$$

与高基尼指数相比，应**优先选择具有低基尼指数**的属性。

5.3 基于 K-Means 得到的第一学年不同代表性群体聚类

这里借助 SPSS 将处理后的三餐餐厅消费金额数据记录 (附件 1-3)，使用 K-Means 算法进行聚类分析，Elbow 法得到聚类系数和聚类个数 K 的图像如下：

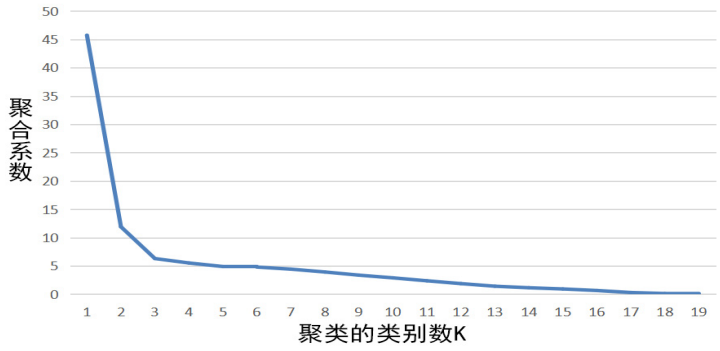

图 5　手肘法进行聚类个数 K 分析

通过分析，可知 K=3 处弯曲的尖点或图中的看起来像手肘的点，因此 **K=3 被认为是 K 的最佳值**。

5.4 通过决策树判断不同特征重要性

特征变量的重要性决定了它们在决策树中的影响力。

基尼不纯度是衡量随机变量的分类不确定性的度量。具体来说，特征变量越能够清晰地将数据划分为不同类别，其重要性就越高。决策树模型会选择具有最高基尼不纯度的特征变量作为第一个分裂点，并递归地对剩余的数据进行分裂。这个过程重复进行，直到所有的叶子节点都包含相似的样本或者达到了预先设定的最大深度。

这里我们利用 SPSS 并采用求解基尼系数的方法来对特征重要性进行定量分析。

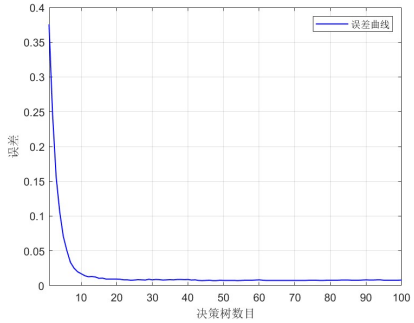

图 6　误差曲线

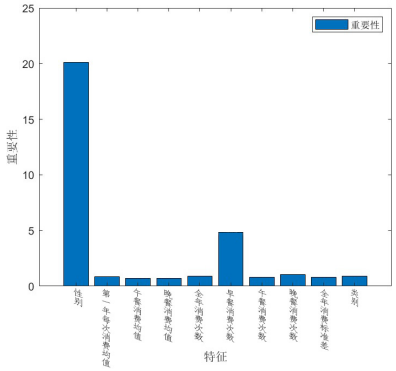

图 7　各特征的重要性柱状图

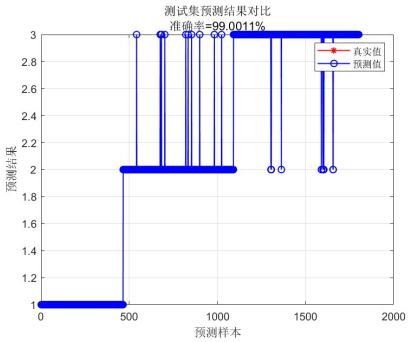

图 8　误差曲线

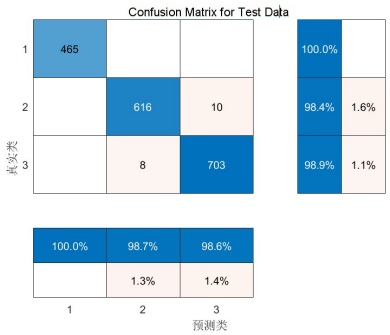

图 9　各特征的重要性柱状图

通过借助 SPSS 的决策树分析，我们可以发现：

1. 当决策树数目达到较大程度时，根据误差曲线，模型相对偏差达到较小水平，且曲线平缓，**基本达到稳定状态**。

2. 根据各特征的重要性柱状图，我们可以发现初性别外，重要性占比前三的特征为**全年消费次数**、**第一年消费均值**、**午餐消费次数**，注意这里**性别重要性**虽然占比较大，但是由于其数据的离散型较强，不予以考虑。

3. 根据**测试集结果准确性对比**，我们可以了解到通过 K-Means 算法拟合的三个聚类准确度已经达到较高程度（99% 以上）。

4. 根据**测试集的混淆矩阵 (Confusion Martrix for Test Data)**, 对聚类 1 的预测准确度达到 100%，而对聚类 2 和聚类 3 也都分别达到了 98% 以上，充分证明了 **K-Means 算法进行聚类分析对学生日三餐餐厅消费金额数据记录进行聚类分析良好的符合性**

综上所述，通过借助于 SPSS 的决策树算法，我们可以充分验证 K-Means 算法对附件 1-3 该组学生不同学年的日三餐餐厅消费金额数据进行聚类分析的**合理性和正确性**。

5.5 定量分析主要消费行为特征变化规律

通过上面对 K-Means 算法的验证，我们可以得到：

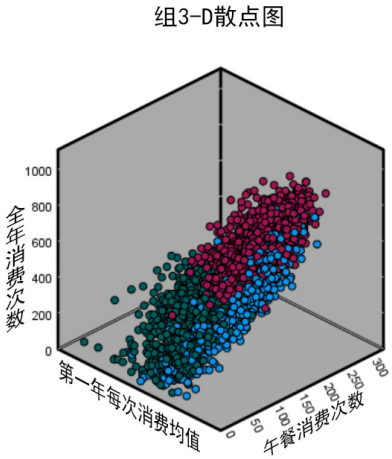

图 10　特征散点图

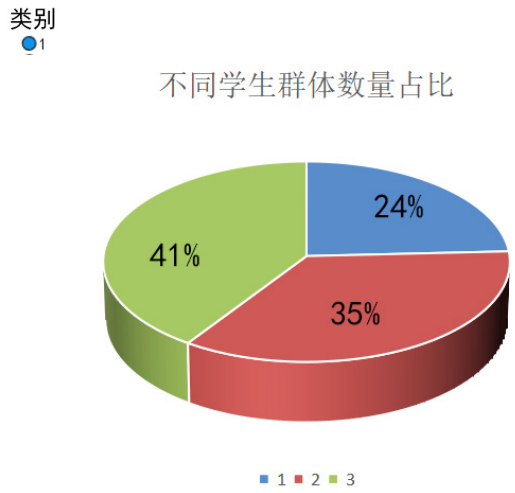

图 11　聚类占比饼状图

从而我们可以得知，**类 1 为高消费水平，类 2 为中消费水平，类 3 为低消费水平**，其中高消费水平占比 24%，中消费水平占比 35%，低消费水平占比 41%。

从而我们可以对高、中、低三个类别的学生群体进行消费行为特征变化规律的预测和总结，以下为相关特征的变化趋势：

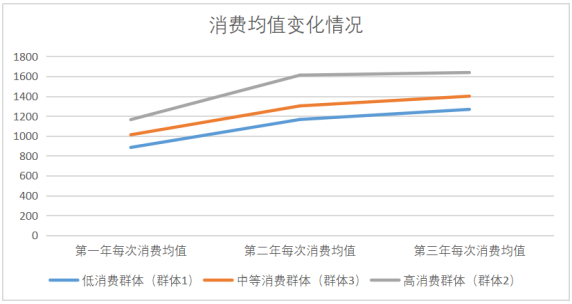

图 12　消费均值变化情况

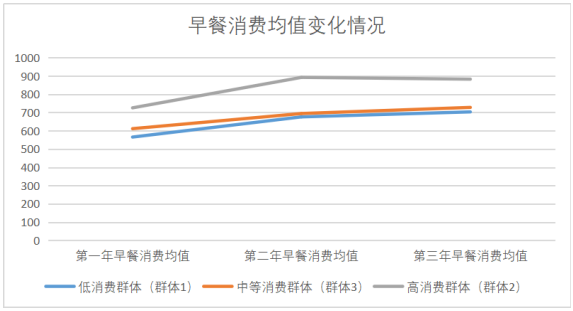

图 13　早餐消费均值变化情况

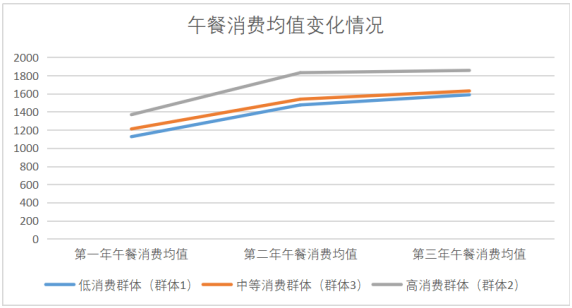

图 14　午餐消费均值变化情况

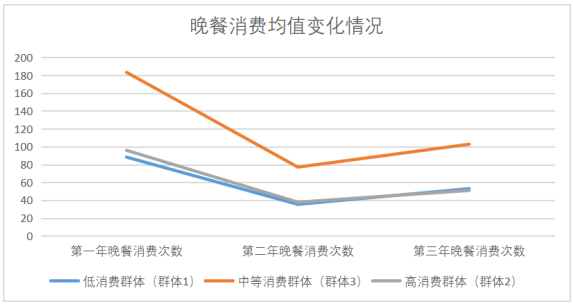

图 15　晚餐消费均值变化情况

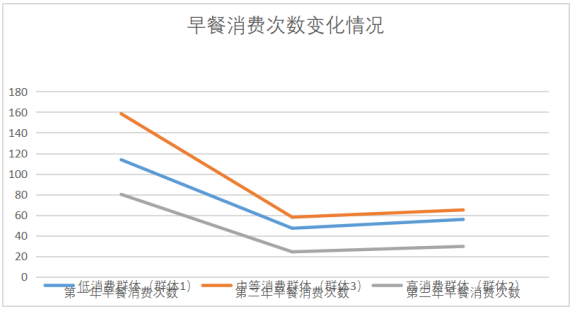

图 16　早餐消费次数变化情况

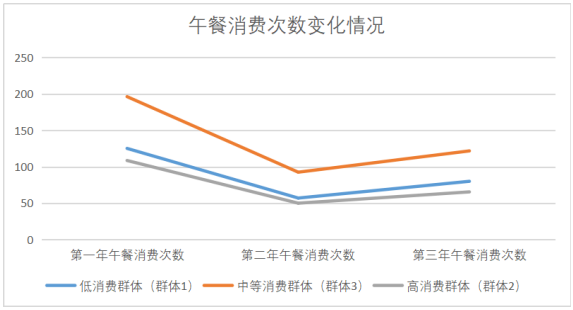

图 17　午餐消费次数变化情况

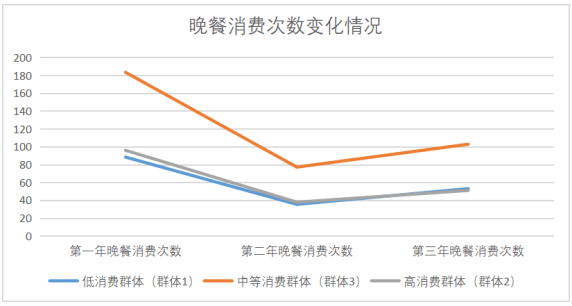

图 18　晚餐消费次数变化情况

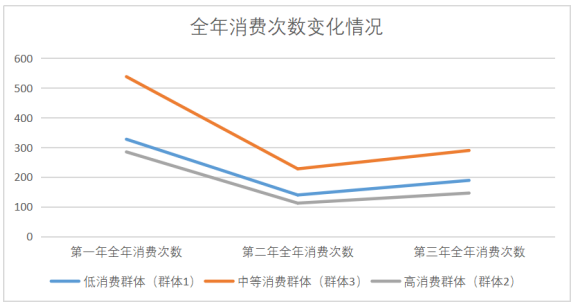

图 19　全年消费次数变化情况

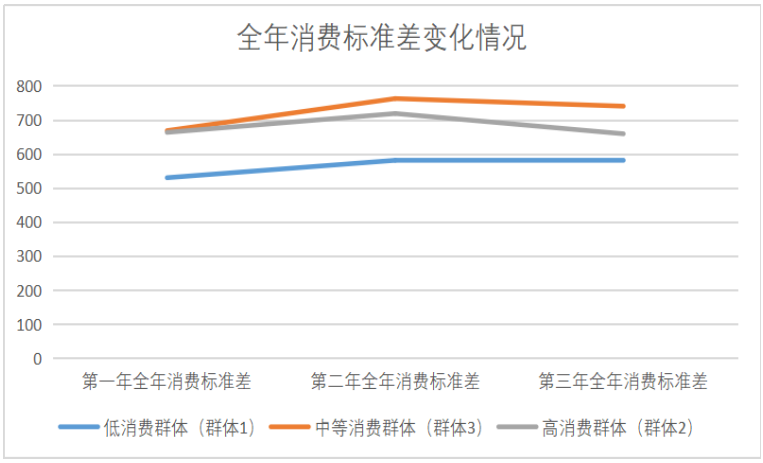

图 20　全年消费标准差变化情况

根据以上不同聚类数据特征，我们可以发现高、中、低收入人群的消费习惯和消费水平**在趋势上呈现一致性**。

在早餐和午餐以及各餐均价上呈现上升趋势，而在晚餐上呈现下降趋势。在早中晚三餐以及各餐的平均消费次数上都呈现下降趋势。

但值得一提的是，**低收入人群在全年消费标准差上呈现持续上升趋势**。即使在第三学年，可能由于疫情的影响，中高消费水平学生群体标准差普遍下降时，低消费水平学生群体的全年消费标准差保持稳定。我们猜测是因为部分贫困同学得到资助，经济状况

得到改良，从而使得消费的离散程度变大。

不仅如此，我们还可以发现**低消费水平学生群体的消费水平和消费习惯逐步向中高消费水平学生群体靠拢**。

5.6 定量分析饮食种类变化规律

类似于上面定量分析主要消费行为特征变化规律的方法，我们同样对数据中饮食种类变化规律进行分析，得到以下结果：

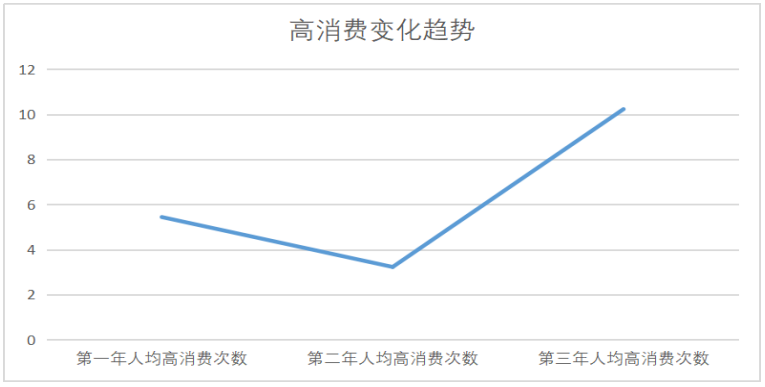

图 21　高消费变化趋势

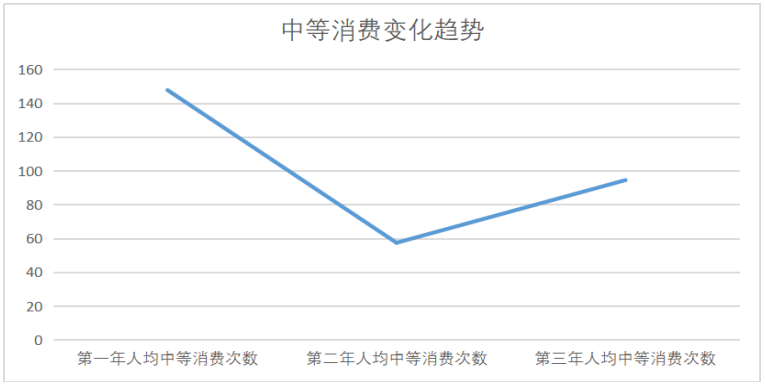

图 22　中消费变化趋势

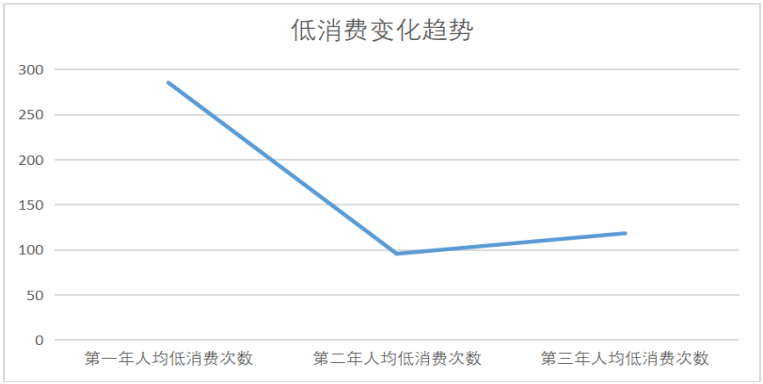

图 23　低消费变化趋势

基于上述变化趋势，我们可以得出**高消费的频次在逐步上升，而中低消费的频次在逐步下降**。可以猜测，这可能也是因为部分贫困同学得到资助所产生的结果。

六、 问题二的模型建立和求解——Random Forest

6.1 随机森林

随机森林 [2] 是一种流行的机器学习算法，属于监督学习技术。它可用于机器学习中的分类和回归问题。它基于集成学习的概念，集成学习是结合多个分类器来解决复杂问题并提高模型性能的过程。

首先，我们在解决问题一建立模型时已经引入了**决策树**的概念，实际上我们可以说决策树是随机森林的最小组成单元。顾名思义，“随机森林是一个分类器，它包含给定数据集的各个子集上的许多决策树，并取平均值以提高该数据集的预测准确性。”随机森林不依赖一棵决策树，而是从每棵树中获取预测，**基于多数的预测结果**，并预测最终输出。随机森林中**决策树的数量越多，准确率就越高**，并可以防止过度拟合的问题。

6.1.1 随机森林的假设

由于随机森林结合了多棵树来预测数据集的类别，我们在上面已经提到了随机森林是给予多数的预测结果。因此有可能某些决策树可以预测正确的输出，而其他决策树则可能不能。但所有树一起预测正确的输出。

因此，以下是更好的随机森林分类器的两个假设：

1. 数据集的特征变量中应该有一些**实际值**，以便分类器可以预测准确的结果而不是猜测的结果。

2. 每棵树的预测必须具有**低相关性**。

6.1.2 随机森林的工作流程

随机森林分两个阶段工作，首先是通过组合 N 个决策树来创建随机森林，其次是对第一阶段创建的每棵树进行预测。

下面是随机森林工作流程:

- 从训练集中选择随机 K 个数据点。
- 构建与所选数据点（子集）相关的决策树。
- 选择要构建的决策树的数字 N。
- 重复步骤 1 和 2。
- 对于新的数据点，找到每个决策树的预测，并将新的数据点分配给赢得多数票的类别。

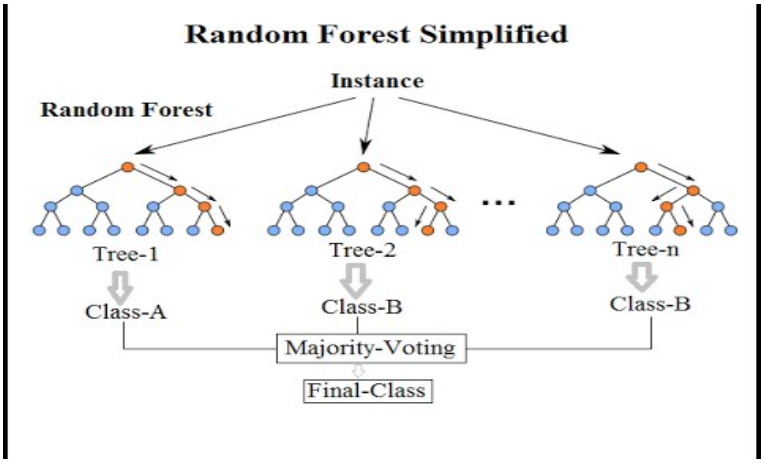

图 24　Random Forest

6.1.3 随机森林的泛化误差

理论证明，当决策树的数目足够大时，随机森林的泛化误差 (bias) 的上界收敛于下面的表达式

$$bias \leq \frac{\bar{\rho}(1-s^2)}{s^2}$$

一组分类器的强度是指分类器的平均性能，而性能以分类器的余量 (M) 用概率算法度量:

$$M(X,Y) = P(\hat{Y_\theta} = Y) - \underset{Z \neq Y}{max} P(\hat{Y_\theta} = Z)$$

余量越大，分类器正确预测给定的样本 X 的可能性就越大。由泛化误差上界的定义公式可知，随着树的相关性增加或组合分类器的强度降低，泛化误差的上界趋于增加。因此，随机化有助于减少决策树之间的相关性，从而改善组合分类器的泛化误差。

6.2 依据 Random Forest 的贫困预测模型评估

6.2.1 基于样本重用的 K 折交叉验证法（K-Fold cross validation）

在机器学习建模过程中，我们常常使用样本重用的方法，它的基本思想就是将**原始数据（Dataset）**进行分组，一部分做为训练集来训练模型，另一部分做为测试集来评价模型。通常的做法通常是将数据分为**训练集 (Training Set) 和测试集 (Test Set)**。测试集是与训练独立的数据，完全不参与训练，用于最终模型的评估。

在训练过程中，经常会出现过拟合的问题，就是模型可以很好的匹配训练数据，却不能很好在预测训练集外的数据。如果此时就使用测试数据来调整模型参数，就相当于在训练时已知部分测试数据的信息，会影响最终评估结果的准确性。因此我们使用 **K 折交叉验证法 (K-Fold cross validation)**，通过对 k 个不同分组训练的结果进行平均来减少方差，下面是 **K 折交叉验证法**的步骤：

- 不重复抽样将原始数据随机分为 k 份。
- 每一次挑选其中 1 份作为测试集，剩余 k-1 份作为训练集用于模型训练。
- 重复第二步 k 次，这样每个子集都有一次机会作为测试集，其余机会作为训练集。
- 在每个训练集上训练后得到一个模型，用这个模型在相应的测试集上测试，计算并保存模型的评估指标，
- 计算 k 组测试结果的平均值作为模型精度的估计，并作为当前 k 折交叉验证下模型的性能指标。

本题中我们采用这种方法来对依据随机森林建立的消费行为和贫困程度的数学模型来进行检验，通过多次交叉验证评估该模型的泛化误差和拟合效果，以期验证依据随机森林建立模型的合理性可靠性，并更好地进行预测。

6.2.2 贫困预测模型性能评估指标

准确率（Accuracy），表示的是分类准确的样本数占该类样本数的比例。计算公式为：

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

精确率（Precision），表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了，一种就是把正类预测为真正 (TP)，另一种就是把负类预测为假正 (FP)，也就是

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

召回率（Recall)，表示的是样本中的正例有多少被预测正确了。那也有两种可能，一种是把原来的正类预测成真正 (TP)，另一种就是把原来的正类预测为假负 (FN)，也就是

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

此外，还有 **F1 值**，是**精确率和召回率的调和均值**.

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

精确率和准确率都高的情况下，F1 值也会高。

6.3 对附录 9 中同学进行贫困预测

利用 **MATLAB** 依据随机森林进行该贫困预测模型进行建构，在附件 1-3 中组学生不同学年的日三餐餐厅消费金额数据记录的特征数据和附件 8 贫困标签的训练下，**对附件 9 中的同学进行贫困程度预测, 得到如下曲线**:

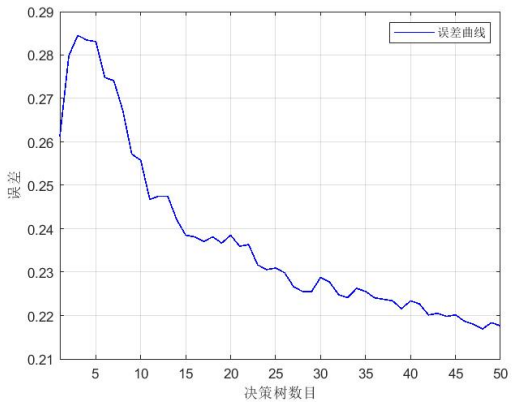

图 25　误差曲线

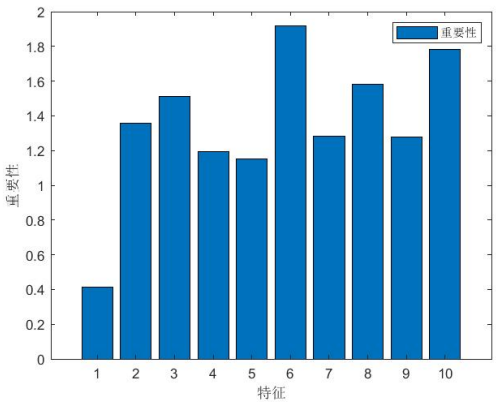

图 26　特征重要性

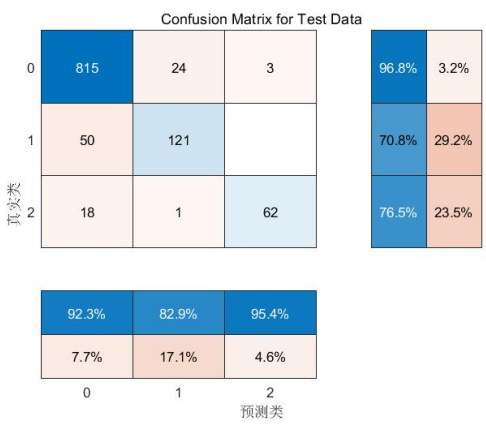

图 27　混淆矩阵

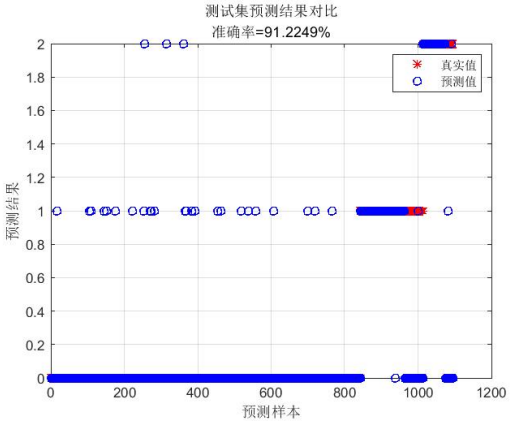

图 28　测试集预测结果对比

模型中有三个值 0/1/2，分别对应**不贫困标签，一般贫困标签和特别贫困标签**。

1. 从误差曲线中可以看出，**当决策树数量达到 50 时，误差已经减小到较低水平**（0.22 以下），可以较好的与实际情况拟合，进行贫困程度预测的进一步精确化。

2. 特征重要性柱状图中编号分别对应性别、每早中晚消费金额、每早中晚消费次数、标准差。其中特征重要性占比前列的包括午餐的消费次数。同时我们可以发现**性别特征的重要性占比较低**，恰恰验证了第一问中通过决策树判断 K-Means 合理性时舍弃性别因素的合理性。

3. 通过测试集的混淆矩阵，我们可以得到各个标签的精确率和召回率。

表 2　模型评价指标

标签	精确率	召回率	F_1 得分
0（不贫困标签）	0.923	0.968	0.945
1（一般贫困标签）	0.829	0.708	0.764
2（特别贫困标签）	0.954	0.765	0.849

比对发现，该模型**对不贫困标签的预测较优**，三项指标都属于较高水平。而**对于一般贫困标签的预测不佳**，三项指标都属于较低水平。而**对于特别贫困标签的预测，精确率较高，但召回率较低，导致 F1 得分水平仅为一般水平**。

4. 通过测试集预测结果对比，发现**真实值与预测值贴合较好，准确率达到 91% 以上**。

以下是利用模型对附件 9 进行预测，得到相关数据的结果（部分）：

表 3　对附件 9 的部分预测结果

序号	**...**	***1564***	***1579***	***1585***	***1591***	**...**	***2225***	***2243***	***2244***	**...**
贫困程度认定标签	**...**	*0*	***0***	***1***	***0***	**...**	***0***	***2***	***0***	**...**

6.4 预测第二、三学年贫困程度隐形认定等级

下面我们借助先前构建的 Random Forest 分类模型进行对该组同学第二、第三学年的贫困程度隐形分析，得到如下结果：

表 4　第二、第三学年贫困等级预测

序号	第一学年贫困等级	第二学年贫困等级	第三学年贫困等级
1	0	0	0
2	0	0	0
3	0	0	0
4	2	1	0
5	0	0	0
6	0	0	0
...	...	...	...

我们可以发现该模型对于大部分同学贫困等级认定结果为 0，即不贫困标签。这与实际情况并不符合，**说明该模型对于严重贫困等级和一般贫困等级的预测并不准确**。此外，我们还可以发现某些特殊的样本例如四号样本，它的贫困认定等级逐年下降，最终归为不贫困认定。

综合上述模型预测结果，我们可以给出如下猜想：随着精准资助的开展，判定高校家庭经济贫困学生逐步精准化，困难学生得到适当的资助，经济状况得到改善，从而脱离贫困情况。

此外，也不排除是由于数据集中部分样本缺失和由于初始贫困程度判断偏差所导致的噪声干扰。

七、 问题三的模型建立和求解——Random Forest 的进一步深入发掘

7.1 质量优化的复合数据集建立

基于先前建立的随机森林贫困程度预测模型，我们这里**引入附录 4-7 的部分同学的三学年饮食种类信息**，进一步对模型进行完善。

但是由于附录 4-7 中的内容存在部分缺失，因此我们将附录 4-7 中的内容和附录 1-3 “该组学生不同学年的日三餐餐厅消费金额数据” 和附录 8 “部分同学第一学年后经其它方式认定的贫困程度等级” **建立内部链接**，从而**获得更高维度的数据集**。

7.2 优化 Random Forest 模型进行深度预测

通过 K-Fold 交叉检验的方法进一步预测相关同学的贫困程度认定。

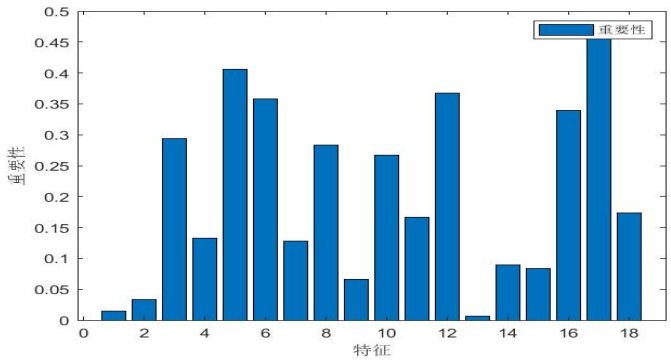

图 29　特征重要性

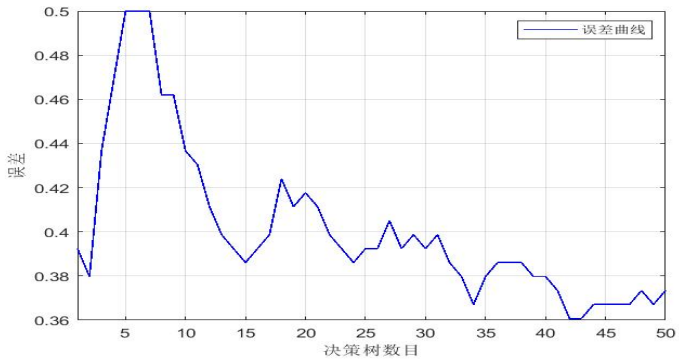

图 30　误差曲线

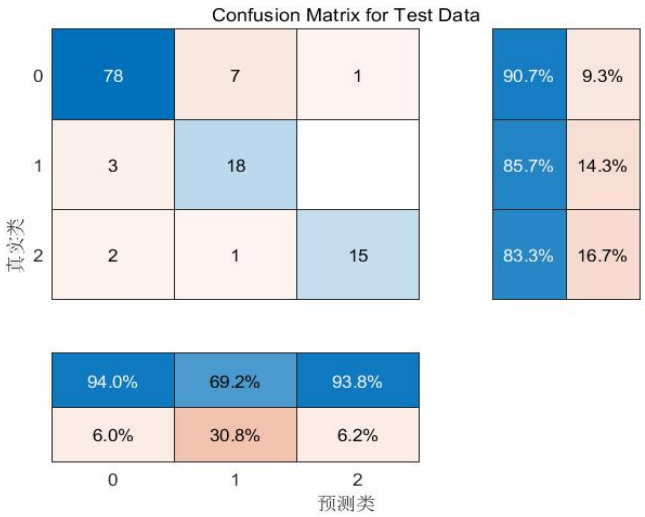

图 31　混淆矩阵

1. 对模型进行数据特征重要性分析，序号分别为性别、每早中晚消费金额、每早中晚消费次数、标准差、每早中晚食物种类、低中高及总消费次数。，我们可以发现特征重要性中引入的**高消费次数特征占据最重要地位**，达到 0.45 以上。更高维度的数据特征重要性评判可以使得模型更加贴近真实情况。

2. 分析误差曲线，我们可以发现，伴随决策树数目的增加，**误差总体呈现下降趋势**。但出现曲线波动性增强，这是由于**训练样本较少，使得模型无法得到充分训练**，给预测带来不利影响，误差波动较大。

3. 通过混淆矩阵，我们可以得到改进后模型进行预测的精确率、召回率以及 F1 得分：

表 5　优化模型评价指标

标签	精确率	召回率	F_1 得分
0（不贫困标签）	0.940	0.907	0.923
1（一般贫困标签）	0.692	0.857	0.766
2（特别贫困标签）	0.938	0.833	0.882

该优化模型与原模型类似该模型**对不贫困标签的预测较优**，三项指标都属于较高水平。但在**对于一般贫困和严重贫困等级划分上有一定提升**，F_1 评分分别达到了 **0.766** 和 **0.882**，相比原模型 F_1 值都达到更高水平。其中尤其**特别贫困等级标签的认定有更大幅度的提升**。

八、 基于特征重要性结合 TOPSIS 法的贫困程度综合评价模型

8.1 TOPSIS 法的方法原理

设多属性评价对象集为 $D=\{d_1,d_2,\ldots,d_m\}$, 衡量对象优劣的属性变量为 $x_1,x_2,\ldots,x_n$, 这时对象集 D 中的每个评价对象 $d_i(i=1,2,\ldots,m)$ 的 n 个属性值构成的向量是 $[a_{i1},a_{i2},\ldots,a_{in}]$, 它作为 n 维空间中的一个点, 能唯一地表征评价对象 d_i。

正理想解 C^* 是一个对象集 D 中并不存在的虚拟的最佳对象, 它的每个属性值都是决策矩阵中该属性的最优值; 而负理想解 C^0 则是虚拟的最差对象, 它的每个属性值都是决策矩阵中该属性的最差值。在 n 维空间中, 将对象集 D 中的各评价对象 d_i 与正理想解 C^* 和负理想解 C^0 的距离进行比较, 既靠近正理想解又远离负理想解的对象就是对象集 D 中的最优对象; 并可以据此排定对象集 D 中各评价对象的优先次序。

TOPSIS 法所用的是欧几里得距离。有时会出现某两个评价对象与正理想解的距离相同的情况, 为了区分这两个对象的优劣, 引入负理想解并计算这两个对象与负理想解的距离, 与正理想解的距离相同的对象离负理想解远者为优。

下面我们来介绍算法过程:

8.1.1 正向化处理

首先，在处理数据时，有些指标的数据越大越好，有些则是越小越好，有些又是中间某个值或者某段区间最好。我们可以对其进行“正向化处理”，使指标都越大越好。

对于极小型指标的正向化处理 对于任何情况，我们可以使用

$$x_i'=max-x_i$$

来实现正向化。特别地，如果元素全是正数，还可以采用

$$x_i'=\frac{1}{x_i}$$

来实现正向化的目的

对于中间型指标的正向化处理 假设其最佳数值是 x_{best}，我们可以取

$$M=|x_i-x_{best}|,$$

再取

$$x_i'=1-\frac{x_i-x_{best}}{M}$$

也能实现正向化处理

对于区间型指标的正向化处理　如果一个区间型指标, 其最佳区间是 $[a,b]$, 我们取 $M = maxa - minx_i, max(x_i) - b$, 之后再按照

$$x_i' = \begin{cases} 1 - \frac{a-x_i}{M}, & x_i < a \\ 1, & a < x_i < b \\ 1 - \frac{x_i-b}{M}, & x_i > b \end{cases}$$

8.1.2 标准化处理

使用向量规划化方法求得规范决策矩阵。设多属性决策问题的决策矩阵 $A = (a_{ij})_{m\times n}$, 规范化决策矩阵 $B = (b_{ij})_{m\times n}$, 其中

$$b_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{m} a_{ij}^2}}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n.$$

8.1.3 构造规范矩阵

构造加权规范阵 $C = (c_{ij})_{m\times n}$。设由决策人给定各属性的权重向量为 $w = [w_1, w_2, \ldots, w_n]^T$, 则

$$c_{ij} = w_{ij} \cdot b_{ij}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n.$$

8.1.4 确定正理想解与负理想解

设正理想解 C^* 的第 j 个属性值为 c_j^*, 负理想解 C^0 第 j 个属性值为 c_j^0, 则

$$\text{正理想解}: c_j^* = \begin{cases} \max\limits_i c_{ij}, j\text{是效益型属性}, \\ \min\limits_i c_{ij}, j\text{是成本型属性}, \end{cases} \quad j = 1, 2, \ldots, n,$$

$$\text{负理想解}: c_j^0 = \begin{cases} \min\limits_i c_{ij}, j\text{是效益型属性}, \\ \max\limits_i c_{ij}, j\text{是成本型属性}, \end{cases} \quad j = 1, 2, \ldots, n,$$

8.1.5 计算欧氏距离

评价对象 d_i 到正理想解的距离 s_i^* 为

$$s_i^* = \sqrt{\sum_{j=1}^{n} (c_{ij} - c_j^*)^2}, i = 1, 2, \ldots, m;$$

评价对象 d_i 到负理想解的距离 s_i^0 为

$$s_i^0 = \sqrt{\sum_{j=1}^{n} (c_{ij} - c_j^0)^2}, i = 1, 2, \ldots, m.$$

8.1.6 排序

设评价对象 d_i 的综合评价指数为 f_i,

$$f_i = \frac{s_i^0}{s_i^0 + s_i^*}, i = 1, 2, \ldots, m.$$

然后将 f_i 从大到小排列即可得到各评价对象的优劣次序
以上就是 TOPSIS 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，从而得出了较为科学、公正、客观的评价结果。

8.2 TOPSIS 法结合特征重要性参数对学生进行贫困程度打分

由上述对 TOPSIS 法的介绍可知，我们采用问题一得到的赋权向量对经过正向化、标准化的数据矩阵进行处理。注意，该过程抛弃掉了性别变量对于评分的影响，因为性别变量无论是 0 还是 1，均属于正常值，不应参与正理想解与负理想解的计算中。于是我们采用了除去这一特征变量的另九个特征按重要性得分对矩阵进行赋权，得到规范矩阵。

8.3 利用线性函数对排出的 80 位同学进行资助

此分配方法的基本精神是：依据贫困程度评分高低来进行线性的金额分配，对于此线性函数的求解，使用待定系数法。

首先，为保证这些同学都能得到一定的补助，我们设定补助下界，即第八十名分到 500 元。而后可列出二元一次方程组

$$\begin{cases} 0.0296572488441 \times k + 80 \times b = 100000 \\ 0.000337 \times k + b = 500 \end{cases}$$

其中 k 是线性函数的斜率，b 为其纵截距，0.0296572488 为前八十位同学贫困评分归一化后数据之和，0.000337 是第八十位同学贫困评分归一化后所得分数

表 6　资助额度分配

序号	贫困程度评价（经归一化后的得分）	排名	资助金额
1597	0.000454605	01	3116.10
2914	0.000439928	02	2789.62
921	0.00042967	03	2561.44
...	...	...	...
4125	0.00033727	79	506.00
4114	0.000337053	80	501.18

该贫困程度评分对资助金额的函数为

$$y = 22244888.6020 \times x - 6996.5275$$

其中 x 是贫困程度评分，y 是资助金额

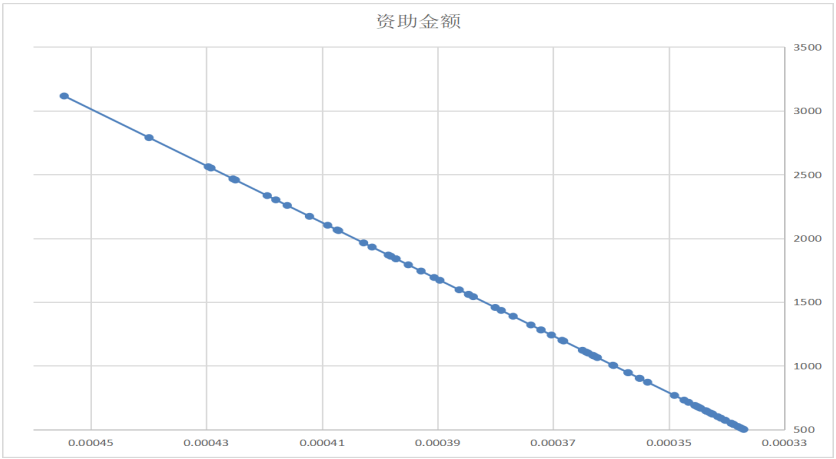

图 32　函数图像

最终资助前八十位的总金额共计 99999.99 元，于是我们可以认定这是一个合理的方案。

九、 模型的评价与改进

9.1 模型的优缺点

9.1.1 优点

- 在问题一中使用的 K-means 算法的**时间复杂度相对较低，计算效率高**，适用于附件 0、1、2、3 这种大规模数据集；而且它**可扩展性强**，可以通过各种改进和优化应用于不同类型的数据和问题。
- 在问题一中利用**系统聚类分析结合肘部法则**验证了选择 k=3 进行 K-means 对样本进行分类的合理性与正确性
- 问题一中使用的**决策树算法，时间复杂度较小**，为用于训练决策树的数据点的对数，**对缺失值不敏感，可以处理不相关特征数据**。
- 问题二和三中使用的随机森林算法，表现性能好，与其他算法相比有着很大优势；**能合适处理高维度数据，并且避免繁琐特征选择**；在训练完之后，随机森林能给出不同特征的重要性；训练速度快，基于决策树之间的独立性，易优化为并行方案。
- 问题四中按照各贫困特征的**重要性数据加权后的得分线性分配资助金额**，这样可以确保资助金额的分配是基于学生的贫困程度评分来完成的。

9.1.2 缺点

- **模型样本固定:** 经过数据预处理后，附件 4-7 中仅涉及了不到 300 名同学的饮食种类，而该组学生中其余部分不含此维度数据，使得难以继续通过训练优化模型。
- **特征样本少导致模型偏差:** 附件 8 中学生样本偏少，且建立模型的主要依据严重贫困标签样本更是极度缺失，这会导致模型训练后精准度不高而产生误差。
- **数据集存在片面性：** 单一的餐厅消费数据并不能完整的反映学生真正经济状况。例如，学生饮食支出在其日常开销中占比，所用电子产品的数量及价格，日常穿戴衣物饰品价格等等。

9.2 模型的改进

- **脏标记数据的处理:** 附件 8 中给出的学生的贫困程度，在值为 0、1 时有数据不准确的情况，导致训练数据混入了脏数据。要基于这个方面对模型进行改进，可以软化标签 [3]，使用 label distribution，即将标签看成在不同类别之间的离散分布，由此能更好地表示边界模糊的标签之间的联系
- **深入挖掘有效数据来源:** 当前模型主要基于学生食堂消费数据，最后的分析结果难免会存在片面的情况，未来再次分析时，可以考虑引入学生饮食支出在其日常开销中占比，所用电子产品的数量及价格，日常穿戴衣物饰品价格等数据，并以此为指

导，构建更客观、正确、全面的模型。

- **精确信息收集体系广泛化:** 由于附件 8 中的样本数据过小，导致形成的随机森林模型准确率和召回率偏低，应当收集更多精确数据保证随机森林模型的精确度。
- **设置立体化、高维度的扶贫金额取值机制**在此模型中，我们主要采用了线性插值的方法来为 80 位学生分配资助金额，但现实生活中的真实情况可能更加复杂，导致线性插值的方法失灵。对此，我们可以考虑按某些特征重要性次序来引入非线性插值方法进行分配
- **加权系数合理化：**在问题四的评价中，使用权重的数据为问题一中所得到的特征重要性数据，可能会因数据以及统计方法而与真实、科学的情况产生偏差，可以通过专家经验来给出更加系统、更加客观加权系数，以此达到更好的效果

9.3 模型的推广

- 该模型是基于餐厅消费数据的隐形资助模型，但是同样的，有许多场景下的数据与餐厅消费数据具有相似性，我们可以将该模型推广到相关场景，以便响应国家打赢全面脱贫攻坚战的号召。

参考文献

[1] 吴海丽. 大数据挖掘中的 K-means 无监督聚类算法的改进 [J]. 现代电子技术,2020,43(19):118-121.DOI:10.16652/j.issn.1004-373x.2020.19.028.

[2] 段嫦慧. 基于随机森林算法的跌倒风险预测系统设计与研究 [D]. 华南理工大学,2022.DOI:10.27151/d.cnki.ghnlu.2022.000178.

[3] Gao B B , Xing C , Xie C W ,et al.Deep Label Distribution Learning with Label Ambiguity[J].IEEE Transactions on Image Processing, 2016, PP(99):1-1.DOI:10.1109/TIP.2017.2689998.

附录 A　附录

1.1　问题一

1.1.1　数据的预处理：除去 95% 以上的人未吃饭的天数

```
clear;clc;
data1 = readtable('E:\校赛\数模\2023 年校赛命题\2023 年校赛命题\C 题\附件 1 第一年三餐消费数据.xlsx','ReadVariableNames',true);
% data2 = readtable('E:\校赛\数模\2023 年校赛命题\2023 年校赛命题\C 题\附件 2 第二年三餐消费数据.xlsx','ReadVariableNames',true);
% data3 = readtable('E:\校赛\数模\2023 年校赛命题\2023 年校赛命题\C 题\附件 3 第三年三餐消费数据.xlsx','ReadVariableNames',true);

% data2(:,1) = [ ];
% data3(:,1) = [ ];
% data = horzcat(data1,data2,data3);
data = table2cell(data1);
% 假设 data 是一个 5415x1098 的数组，保存了所有的数据
studentNum = 5415;  % 学生数量
dayNum = 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, dayNum * 3 + 1);

% 遍历每一天
for i = 2:3:(dayNum * 3) + 1
% 计算每一天三餐消费总额为 0 的学生数量
zeroNum = 0;
for j = 1:5415
if dataj,i + dataj,i+1 + dataj,i+2 == 0
zeroNum = zeroNum + 1;
end
end

% 如果三餐消费总额为 0 的学生数量超过总学生数量的 95%，标记这一天的数据列需要被删除
```

```
if zeroNum / studentNum > 0.95
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
year = find(deleteColumns(2:1099));

% fid = fopen('E:\校赛\数模\数据处理\不合理天数.txt','w');
% fprintf(fid,' 第一年删除天数： %d\n',length(year)/3);
% fprintf(fid,' 第二年删除天数： %d\n',length(year)/3);
% fprintf(fid,' 第三年删除天数： %d\n',length(year)/3);
deletedColumns = find(deleteColumns);
disp('Deleted columns: ');
disp(deletedColumns);
xlswrite('E:\校赛\数模\数据处理\附件 1 第一年食堂消费删除不合理天数.xlsx', data)
%xlswrite('E:\校赛\数模\数据处理\附件 1 第二年食堂消费删除不合理天数.xlsx', data)
%xlswrite('E:\校赛\数模\数据处理\附件 1 第三年食堂消费删除不合理天数.xlsx', data)
```

1.1.2 决策树的 MATLAB 代码

```
% 清空环境变量
warning off % 关闭报警信息
close all % 关闭开启的图窗
clear % 清空变量
clc % 清空命令行

% 导入数据
res = xlsread('C:\Users\17519\Desktop\2023年校赛真题\第一问\问题一 最后数据 - 副本.xlsx');

% 划分训练集和测试集
temp = randperm(5404);
```

```
P_train = res(temp(1: 3602), 1: 10)';
T_train = res(temp(1: 3602), 20)';
M = size(P_train, 2);

P_test = res(temp(3603: end), 1: 10)';
T_test = res(temp(3603: end), 20)';
N = size(P_test, 2);

% 数据归一化
[p_train, ps_input] = mapminmax(P_train, 0, 1);
p_test = mapminmax('apply', P_test, ps_input );
t_train = T_train;
t_test = T_test ;

% 转置以适应模型
p_train = p_train'; p_test = p_test';
t_train = t_train'; t_test = t_test';

% 训练模型
trees = 100; % 决策树数目
leaf = 1; % 最小叶子数
OOBPrediction = 'on'; % 打开误差图
OOBPredictorImportance = 'on'; % 计算特征重要性
Method = 'classification'; % 分类还是回归
net = TreeBagger(trees, p_train, t_train, 'OOBPredictorImportance', OOBPredictorImportance,
...
'Method', Method, 'OOBPrediction', OOBPrediction, 'minleaf', leaf);
importance = net.OOBPermutedPredictorDeltaError; % 重要性

% 仿真测试
t_sim1 = predict(net, p_train);
t_sim2 = predict(net, p_test );

% 格式转换
T_sim1 = str2double(t_sim1);
```

```
T_sim2 = str2double(t_sim2);

% 性能评价
error1 = sum((T_sim1' == T_train)) / M * 100 ;
error2 = sum((T_sim2' == T_test )) / N * 100 ;

% 绘制误差曲线
figure
plot(1: trees, oobError(net), 'b-', 'LineWidth', 1)
legend(' 误差曲线')
xlabel(' 决策树数目')
ylabel(' 误差')
xlim([1, trees])
grid

% 绘制特征重要性 figure
bar(importance)
legend(' 重要性')
xlabel(' 特征')
ylabel(' 重要性')

% 数据排序
[T_train, index_1] = sort(T_train);
[T_test, index_2] = sort(T_test );

T_sim1 = T_sim1(index_1);
T_sim2 = T_sim2(index_2);

% 绘图
figure
plot(1: M, T_train, 'r', 1: M, T_sim1, 'b-o', 'LineWidth', 1)
legend(' 真实值', ' 预测值')
xlabel(' 预测样本')
ylabel(' 预测结果')
string = ' 训练集预测结果对比'; [' 准确率 =' num2str(error1) '%'];
```

```
title(string)
grid

figure
plot(1: N, T_test, 'r-*', 1: N, T_sim2, 'b-o', 'LineWidth', 1)
legend(' 真实值', ' 预测值')
xlabel(' 预测样本')
ylabel(' 预测结果')
string = ' 测试集预测结果对比'; [' 准确率 =' num2str(error2) '%'];
title(string)
grid

% 混淆矩阵
figure
cm = confusionchart(T_train, T_sim1);
cm.Title = 'Confusion Matrix for Train Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';

figure
cm = confusionchart(T_test, T_sim2);
cm.Title = 'Confusion Matrix for Test Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';
```

1.2 问题二

```
% 清空环境变量
clear    % 清空变量
clc    % 清空命令行

    % 导入数据
res = xlsread('E:\校赛\数模\数据处理\第二题训练集.xlsx');
tosolve = xlsread('E:\校赛\数模\数据处理\求解集.xlsx');
```

```
% 划分训练集和测试集
temp = randperm(4406);
order0 = [ ];
order1 = [ ];
order2 = [ ];

P_train = res(temp(1:3304), 3: 12)';
T_train = res(temp(1:3304), 2)';

M = size(P_train, 2);

P_test = res(temp(3305:4406), 3: 12)';
T_test = res(temp(3305:4406), 2)';

N = size(P_test, 2);
% 数据归一化
[p_train, ps_input] = mapminmax(P_train, 0, 1);
p_test = mapminmax('apply', P_test, ps_input );
t_train = T_train;
t_test = T_test ;

% 转置以适应模型
p_train = p_train';
p_test = p_test';
t_train = t_train';
t_test = t_test';

% 训练模型
trees = 50;    % 决策树数目
leaf = 1;    % 最小叶子数
OOBPrediction = 'on';    % 打开误差图
OOBPredictorImportance = 'on';    % 计算特征重要性
Method = 'classification';    % 分类还是回归
net = TreeBagger(trees, p_train, t_train, 'OOBPredictorImportance', OOBPredictorImportance,
... 'Method', Method, 'OOBPrediction', OOBPrediction, 'minleaf', leaf);
```

```
importance = net.OOBPermutedPredictorDeltaError;  % 重要性

% 仿真测试
t_sim1 = predict(net, p_train);
t_sim2 = predict(net, p_test );

% 格式转换
T_sim1 = str2double(t_sim1);
T_sim2 = str2double(t_sim2);

% 性能评价
error1 = sum((T_sim1' == T_train)) / M * 100 ;
error2 = sum((T_sim2' == T_test )) / N * 100 ;
% 绘制误差曲线
figure
plot(1: trees, oobError(net), 'b-', 'LineWidth', 1)
legend(' 误差曲线')
xlabel(' 决策树数目')
ylabel(' 误差')
xlim([1, trees])
grid

% 绘制特征重要性
figure
bar(importance)
legend(' 重要性')
xlabel(' 特征')
ylabel(' 重要性')

% 数据排序
[T_train, index_1] = sort(T_train);
[T_test, index_2] = sort(T_test );

T_sim1 = T_sim1(index_1);
T_sim2 = T_sim2(index_2);
```

```
% 绘图

figure
plot(1: N, T_test, 'r*', 1: N, T_sim2, 'bo', 'LineWidth', 1)
legend(' 真实值', ' 预测值')
xlabel(' 预测样本')
ylabel(' 预测结果')
string = ' 测试集预测结果对比'; [' 准确率 =' num2str(error2) '%'];
title(string)
grid

% 混淆矩阵

figure
cm = confusionchart(T_test, T_sim2);
cm.Title = 'Confusion Matrix for Test Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';
% 结果输出
[ansLabel, score] = predict(net, tosolve(:,2:11));
ansLabel = cell2mat(ansLabel);
ansLabel = str2num(ansLabel);
xls = horzcat(tosolve(:,1),ansLabel);
xlswrite('E:\校赛\数模\数据处理\第二题预测.xlsx',xls);
data = xlsread('E:\校赛\数模\数据处理\第二年.xlsx');
[ansLab, fen] = predict(net, data(:,2:11));
ansLab = cell2mat(ansLab);
ansLab = str2num(ansLab);
xls = horzcat(data(:,1),ansLab);
xlswrite('E:\校赛\数模\数据处理\第二年预测.xlsx',xls);
data = xlsread('E:\校赛\数模\数据处理\第三年.xlsx');
[ansLab, fen] = predict(net, data(:,2:11));
ansLab = cell2mat(ansLab);
ansLab = str2num(ansLab);
```

```
xls = horzcat(data(:,1),ansLab);
xlswrite('E:\校赛\数模\数据处理\第三年预测.xlsx',xls);
```

1.3 问题三

```
% 清空环境变量
clear    % 清空变量
clc    % 清空命令行

% 导入数据
res = xlsread('E:\校赛\数模\数据处理\第二题训练集.xlsx');
tosolve = xlsread('E:\校赛\数模\数据处理\求解集.xlsx');

% 划分训练集和测试集
temp = randperm(4406);
order0 = [ ];
order1 = [ ];
order2 = [ ];

P_train = res(temp(1:3304), 3: 12)';
T_train = res(temp(1:3304), 2)';

M = size(P_train, 2);

P_test = res(temp(3305:4406), 3: 12)';
T_test = res(temp(3305:4406), 2)';

N = size(P_test, 2);
% 数据归一化
[p_train, ps_input] = mapminmax(P_train, 0, 1);
p_test = mapminmax('apply', P_test, ps_input );
t_train = T_train;
t_test = T_test ;

% 转置以适应模型
p_train = p_train';
```

```
p_test = p_test';
t_train = t_train';
t_test = t_test';

% 训练模型
trees = 50;    % 决策树数目
leaf = 1;    % 最小叶子数
OOBPrediction = 'on';    % 打开误差图
OOBPredictorImportance = 'on';    % 计算特征重要性
Method = 'classification';    % 分类还是回归
net = TreeBagger(trees, p_train, t_train, 'OOBPredictorImportance', OOBPredictorImportance,
... 'Method', Method, 'OOBPrediction', OOBPrediction, 'minleaf', leaf);
importance = net.OOBPermutedPredictorDeltaError;    % 重要性

% 仿真测试
t_sim1 = predict(net, p_train);
t_sim2 = predict(net, p_test );

% 格式转换
T_sim1 = str2double(t_sim1);
T_sim2 = str2double(t_sim2);

% 性能评价
error1 = sum((T_sim1' == T_train)) / M * 100 ;
error2 = sum((T_sim2' == T_test )) / N * 100 ;
% 绘制误差曲线
figure
plot(1: trees, oobError(net), 'b-', 'LineWidth', 1)
legend(' 误差曲线')
xlabel(' 决策树数目')
ylabel(' 误差')
xlim([1, trees])
grid

% 绘制特征重要性
```

```
figure
bar(importance)
legend(' 重要性')
xlabel(' 特征')
ylabel(' 重要性')

% 数据排序
[T_train, index_1] = sort(T_train);
[T_test, index_2] = sort(T_test );

T_sim1 = T_sim1(index_1);
T_sim2 = T_sim2(index_2);

% 绘图

figure
plot(1: N, T_test, 'r*', 1: N, T_sim2, 'bo', 'LineWidth', 1)
legend(' 真实值', ' 预测值')
xlabel(' 预测样本')
ylabel(' 预测结果')
string = ' 测试集预测结果对比'; [' 准确率 =' num2str(error2) '%'];
title(string)
grid

% 混淆矩阵

figure
cm = confusionchart(T_test, T_sim2);
cm.Title = 'Confusion Matrix for Test Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';
% 结果输出
[ansLabel, score] = predict(net, tosolve(:,2:11));
ansLabel = cell2mat(ansLabel);
ansLabel = str2num(ansLabel);
```

xls = horzcat(tosolve(:,1),ansLabel);
xlswrite('E:\校赛\数模\数据处理\第二题预测.xlsx',xls);
data = xlsread('E:\校赛\数模\数据处理\第二年.xlsx');
[$ansLab, fen$] = predict(net, data(:,2:11));
ansLab = cell2mat(ansLab);
ansLab = str2num(ansLab);
xls = horzcat(data(:,1),ansLab);
xlswrite('E:\校赛\数模\数据处理\第二年预测.xlsx',xls);
data = xlsread('E:\校赛\数模\数据处理\第三年.xlsx');
[$ansLab, fen$] = predict(net, data(:,2:11));
ansLab = cell2mat(ansLab);
ansLab = str2num(ansLab);
xls = horzcat(data(:,1),ansLab);
xlswrite('E:\校赛\数模\数据处理\第三年预测.xlsx',xls);

1.4 问题四 TOPSIS 法的 MATLAB 代码

% 导入矩阵，记作 Y（使用原数据，不是标准化数据，只使用 9 列）
[n, m] = size(Y)

% 加入赋权向量
%1: 20.2596(因为在区间内都合理，所以不计入加权)
%2: 0.786873
%3: 0.642723
%4: 0.639639
%5: 0.854186
%6: 4.73251
%7: 0.799405
%8: 0.933516
%9: 0.81919
%10: 0.91923

W=diag([$0.786873, 0.642723, 0.639639, 0.854186, 4.73251, 0.799405, 0.933516, 0.81919, 0.91923$]);
X=Y*W;

% 正向化（由于是正数，第 1,2,3,4,9 每列最大值减去每一个值）

```
X(:,1) = max(X(:,1))-X(:,1);
X(:,2) = max(X(:,2))-X(:,2);
X(:,3) = max(X(:,3))-X(:,3);
X(:,4) = max(X(:,4))-X(:,4);
X(:,9) = max(X(:,9))-X(:,9);

disp([' 共有'num2str(n)' 个评价对象, 'num2str(m)' 个评价指标'])

% 得到标准化矩阵 Z
Z = X ./ repmat(sum(X.*X) .^0.5, n, 1);
disp(' 标准化矩阵 Z = ')
disp(Z)

% 计算距离
D_P = sum([(Z - repmat(max(Z),n,1)) .^ 2 ],2) .^ 0.5;     % D+ 与最大值的距离向量
D_N = sum([(Z - repmat(min(Z),n,1)) .^ 2 ],2) .^ 0.5;     % D- 与最小值的距离向量
S = D_N ./ (D_P+D_N); % 未归一化的得分
disp(' 最后的得分为：')
stand_S = S / sum(S) % 归一化后的得分
%[sorted_S,index] = sort(stand_S ,'descend')

% 导入 Excel 中
xlswrite('C:\Users\17519\Desktop\2023年校赛命题\数据处理\附件 1-3 合并删除不合理天数不合理人员
% 在 Excel 中排序
```