附件 2：

湖南省研究生第九届数学建模竞赛承诺书

我们仔细阅读了湖南省高校研究生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的赛题中选择一项填写）：A

我们的参赛编号（请填写完整参赛编号）：202418001010

所属学校（请填写完整的全名）：国防科技大学

参赛队员（打印后签名）：1. 刘威

2. 江波

3. 刘洋

指导教师或指导教师组负责人（打印后签名）：李文梓

日期： 2024 年 7 月 12 日

（请勿改动此页内容和格式。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

面向人体活动状态判别问题的聚类及集成学习分类方法

摘要

随着智能手机技术的快速发展，准确记录与分析人体活动状态已成为智能健康领域的重要研究方向。针对人体活动状态的识别问题，本文通过数据挖掘与机器学习技术，构建了高精度的分类与判别模型，实现了对人体活动状态数据的分类判别。具体方法包括：构建约束 K-means 聚类模型、集成学习模型以及深度卷积神经网络模型。此外，本文结合方差分析与多元回归分析探究了生理特征与活动状态间的关联性。

在问题正式求解前，首先对数据进行预处理，包括基于 3σ 准则的异常值剔除、关键特征提取、数据标准化及特征降维。通过多次实验，筛选出 20 个关键特征，并通过对比不同降维方法（主成分分析、ReliefF、自动编码器），最终选定主成分分析作为最优降维策略。

针对问题一：对于**无标签**（活动状态）数据的分类问题，采用约束 K-means 聚类方法进行求解。为避免传统 K-means 算法的局部最优问题，引入了 K-means++方法优化聚类中心的初始化。通过多维尺度分析（MDS）技术，绘制了三位实验人员数据的聚类结果图，清晰地展示了不同实验中每种活动状态的一致性和稳定性。针对聚类结果，进一步引入了三类有代表性的聚类评价指标：DB 指数、归一化互信息以及轮廓系数，验证了聚类算法的准确性和可靠性。

针对问题二：对于**有标签**（活动状态）数据的分类问题，针对问题需求，本文：**1）构建集成学习判别模型**：该模型集成了神经网络、决策树、随机森林、朴素贝叶斯和 K 近邻这 5 个先进机器学习子模型，在一致的训练集上通过离线独立训练子模型，之后在在线求解过程中通过加权投票机制综合五个子模型的判别结果形成最终结果；**2）进行不同模型对比**：首先建立模型评价指标，然后分别使用问题一建立的约束 K-means 聚类模型和问题二建立的集成学习模型对附件 2 的数据进行分类，并比较这二者的分类准确度。两者都显示出了较高精度，其中集成学习模型的准确度（98%）高于聚类模型精度（93%）；**3）将集成学习模型应用于附件 3 的验证数据**，完成活动状态的判别。

针对问题三：本问题进一步已知实验人员的年龄、身高及体重数据，针对问题需求，本文：**1）进行方差分析**：首先按活动状态对数据进行分类整理，然后对相同人员的数据进行合并，并应用方差分析进行差异性判断。结果表明在 95%的置信度上不同人员的同一活动状态存在差异，其差异主要体现在“向前走”、“跳跃”等较为剧烈的活动上；**2）进行多元回归分析**：将人员活动特征结合年龄、身高、体重信息，开展多元回归分析，以 P 值评估活动状态与生理特征间的相关性。结果同样表明在 95%的置信度上较为剧烈的活动与生理特征具有相关性。此外，多元回归分析可以使用活动传感器数据进行人员画像，基于拟合得到的系数可以进一步估计人员的年龄、身高、体重数值；**3）构建深度卷积神经网络人员类型判别模型**：对已知标签数据的 10 名人员的活动状态数据进行数据拆分与扩充，构建上万容量的训练数据，同时构建并训练深度卷积神经网络模型，模型精度达到 97%，最终实现对附件 5 中数据的人员编号判别。

本文的模型贴合实际, 对人体活动状态的识别准确率达到了较高水平（>98%），具有求解问题高效、准确的特点，具有较好的泛化性，适用于多个领域。为使得模型进一步优化，可从优化模型架构和超参数选择等角度，加以深化和完善。

关键词：集成学习，聚类分析，深度卷积神经网络，特征提取，多分类

目录

1 问题综述

1.1 问题背景

智能手机技术的进步极大地推动了智能健康领域的一个关键分支——利用智能手机记录人体活动状态。尽管该领域具有巨大的发展潜力，但同时也面临着一系列挑战。市场竞争激烈、功能多样性以及价格的低廉化等因素，共同导致用户在众多健康监测应用中难以做出选择。相关研究资料显示，智能设备的普及率极高，但用户对于能够精确记录和分析人体活动状态的智能手机应用的需求持续增长。传统的记录方法可能存在数据解读的不准确性和个性化服务的不足，这些问题严重削弱了用户对智能手机记录人体活动状态的信任和满意度。随着人们对智能手机记录人体活动状态的需求不断增加，以及专业医疗资源的相对匮乏，传统的记录方法已难以满足市场和用户的高标准需求。因此，探索新的方法并提出合理的解决方案显得尤为迫切。

目前，许多学者已对使用智能手机记录人体活动状态的问题进行了深入研究，并提出了多种算法以提高记录的准确性和提供个性化服务。传统算法如 K-means 聚类、决策树、逻辑回归等已被广泛采用。近年来，随着高性能计算设备的快速发展，深度学习算法和卷积神经网络等先进算法也应运而生。这些算法虽然能够迅速提供初步的活动状态监测结果，但它们通常基于通用性框架，仅能在大多数情况下保证合理性。在特殊情况下，如针对特殊人群或特定活动模式，可能会遇到数据解读的偏差或个性化服务的不精准问题。鉴于此，必须采用更为创新的方法，结合智能手机内置传感器数据特性，建立更为精准和个性化的人体活动状态监测模型。本文将基于智能手机传感器数据，综合采用多种数据预处理及特征提取技术，面向人体活动状态判别问题，提出高准确度的聚类及集成学习分类方法。该方法旨在提供更准确、更个性化的监测方案，以满足用户对智能手机记录人体活动状态的高标准要求。

1.2 问题提出

智能手机内置的加速度计与陀螺仪是监测用户活动状态的关键组件。加速度计负责测量设备在 X、Y、Z 三个坐标轴上的线性加速度变化，而陀螺仪则负责捕捉设备绕这些轴的旋转角加速度。这些传感器收集的数据随后被传输至手机的处理器，后者能够利用这些数据来识别设备的空间姿态、角度以及方向变化，进而对用户的活动模式进行分析和追踪。

本题的研究背景为 10 余名实验人员携带运动状态传感器进行活动，收集他们的日常活动状态的数据。规定他们需要完成“向前走，向 左走，向右走、步行上楼、步行下楼、向前跑、跳跃、坐下、站立、 躺下、乘坐电梯向上移动、乘坐电梯向下移动”12 种活动（如图 1），每种活动记录了 5 组实验数据，每组数据记录其数秒的线加速度和角加速度数据。

图 1 12 种日常活动状态

基于所给数据，需要通过数据处理、特征提取、聚类分析、监督学习等方法建立问题求解模型，以解决以下三个问题：

问题一：附件 1 提供了 3 名实验者的运动数据，其中包含每名实验者在 12 种活动状态下的 5 组加速度计和陀螺仪数据，总计每人 60 组数据。任务是将每位实验者的 60 组未标记数据进行分类，并在表 1 中记录分类结果。

问题二：附件 2 提供了 10 名实验人员的已标记活动状态的运动数据，任务是：

1) 从这些数据中提取 12 种活动状态的特征，构建一个精确的判别模型，以识别实验者的活动状态；

2) 运用问题一中的分类模型处理问题二的数据，比较分类模型与判别模型的结果，并分析分类模型在不同活动类型上的准确度；

3) 使用判别模型对附件 3 提供的某实验人员的活动状态数据进行判别，给出其对应的活动状态，并将结果填入表 2。

问题三：附件 4 提供了问题 1 和问题 2 中参与实验的 13 位实验人员的年龄、身高、体重等数据，根据该数据解决以下问题：

1) 分析不同个体在执行同一活动时是否存在显著差异？

2) 分析活动状态数据与个体生理特征（年龄、身高、体重）之间的相关性，以及是否能够通过这些数据推断个体的生理特征；

3) 附件 5 提供了问题 2 的 10 位实验人员中的 5 位的某次活动数据。任务是建立判别模型，用以识别未知活动数据最可能属于的实验者，并记录结果填入表 3。

附件数据：

【附件 1】提供了 3 名实验人员的未标记活动状态的运动数据

【附件 2】提供了 10 名实验人员的已标记活动状态的运动数据

【附件 3】提供了某实验人员的未标记活动状态的运动数据

【附件 4】提供了 13 名实验人员的年龄、身高、体重等相关数据

【附件 5】提供了【附件 2】中某 5 名未知实验人员的已标记活动状态的运动数据

2 模型假设与符号说明

2.1 模型基本假设

(1) 假设不同的活动状态在加速度计和陀螺仪的数据表现上会有明显的差异，例如跑步和走路在加速度和角速度的变化模式上会有所不同。
(2) 假设相同的活动状态在不同人员身上进行时，其加速度计和陀螺仪工作正常，数据采集正常。
(3) 假设同一人员在不同时间段内进行相同的活动状态时，其加速度计和陀螺仪的数据也会展现出相似的稳定性和一致性。
(4) 虽然假设了活动状态在不同人员间的相似性，但也需要考虑个体之间可能存在的差异，如体型、步态、速度等因素可能对数据的表现产生影响。

2.2 符号说明

本文定义了 12 个使用次数较多的符号，如表 1 所示, 其余符号在使用时注明。

表 1 符号说明

符号	含义	单位
K	聚类数	个
H	硬约束集	无
D	数据集	无
C	聚类中心	无
d_{ij}	样本点到中心点的距离	无
$X_1, X_2, ..., X_n$	未带标签的数据组	无
$Y_1, Y_2, ..., Y_n$	带标签的数据组	无
F 值	组间变异性/组内变异性	无
P 值	统计显著性的概率值	无
σ	标准差	无
ε	误差项	无
FFT	快速傅里叶变换的函数	无

3 数据预处理

数据预处理是数据科学的核心环节，其步骤包括数据清洗、异常值处理、特征提取和数据标准化。如图 2 所示，本研究首先基于 3σ 原则对初始数据的异常值进行清洗，以避免分析结果失真；随后，通过特征提取技术，从原始数据中提取其时域及频率关键信息，以支持分析和建模；此后，再进行数据标准化确保了不同特征间的可比性；最后通过主成分分析进行特征降维，进一步去除数据冗余，优化数据结构。

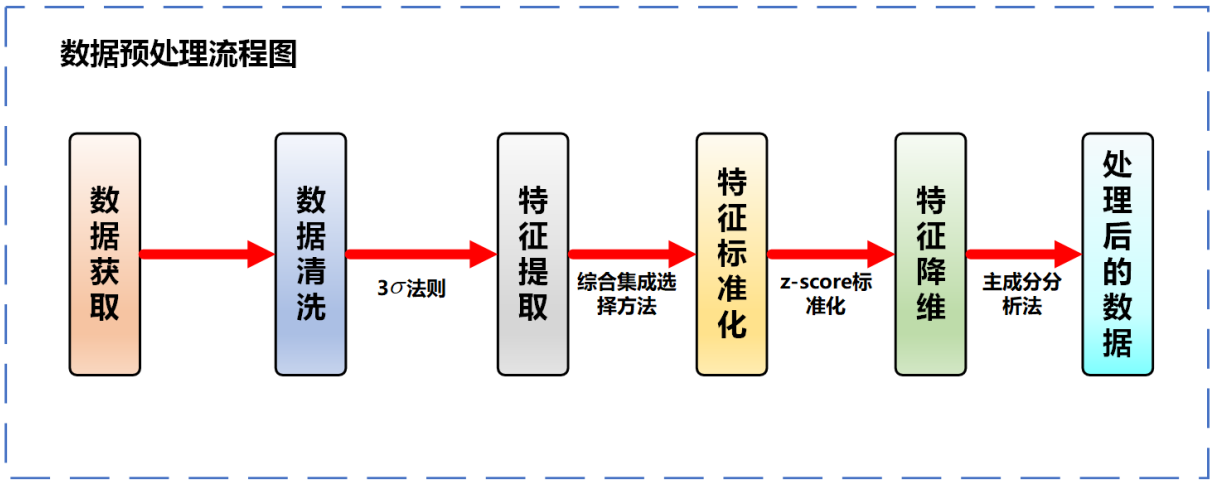


图 2 数据预处理流程图

3.1 数据清洗

通过数据可视化分析，发现原始数据中疑似存在异常值，如图 3 所示的附件 1 中 Person1 的 SY3、SY4、SY6、SY9 四组数据。为此需要对原始数据进行数据清洗，包括识别和处理缺失值、异常值、重复数据以及格式不一致问题，确保数据的准确性和一致性。

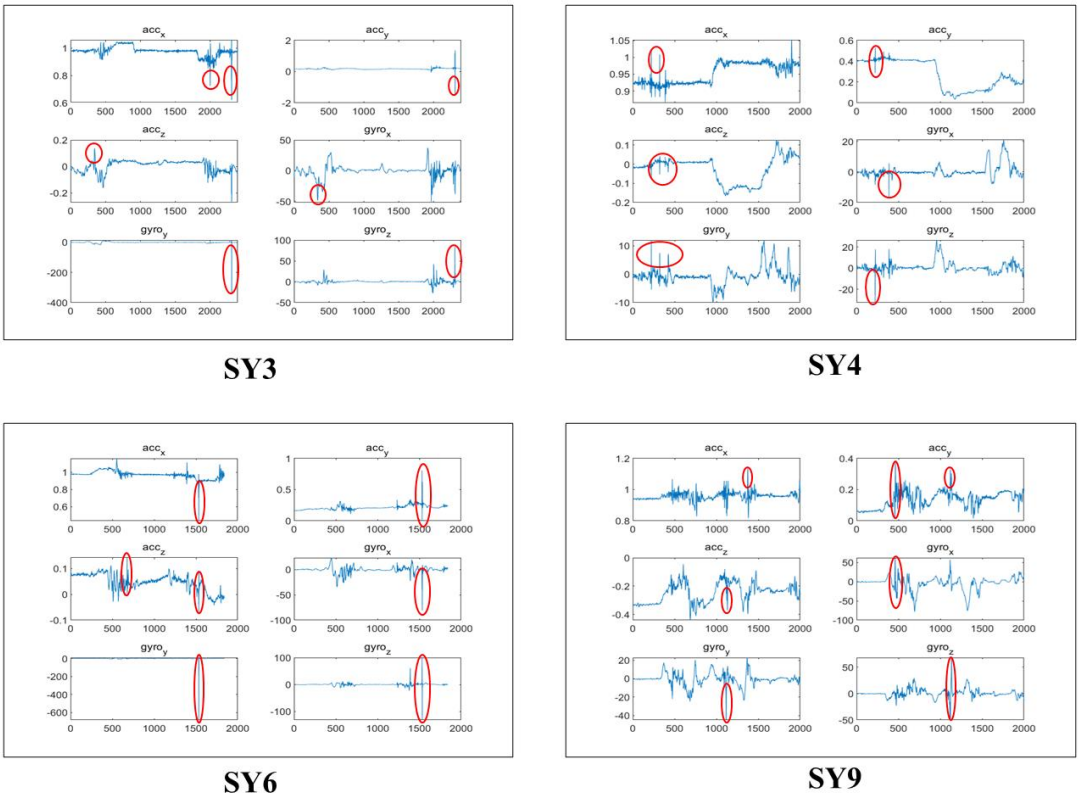


图 3 原始数据可视化

在处理数据中的异常值时，采用了 3σ（3 倍标准差）原则，这是一种统计学中普遍应用的方法。该方法通过计算数据的均值和标准差来确定异常值的阈值，这种方法有助于识别数据集中显著偏离正常范围的数值。正常值为 $x \pm 3\sigma$ 内的数值，超出此范围的

数值被视为异常值，并从数据集中移除。此处理策略旨在避免异常值对分析和建模结果造成的失真。

3.2 特征提取

由于本题数据特征空间庞大，记录下的实验人员活动状态数据维度介于 1000×6 至 5000×6 之间，直接使用高维数据进行分析和建模不仅极大增加计算负担，也会影响数据分析精度。为提高问题求解精度同时统一数据维度，本文对数据进行综合特征提取，计算多种时间序列的时域和频域特征，用来共同表征原始数据，在尽量保留数据原始信息的前提下降低数据维度。本文提取了平均值、最大最小值、偏度、峰度、主频、频率能量等时域、频域特征，综合表征原始数据，以精简数据集，去除冗余信息，提高数据学习效率。

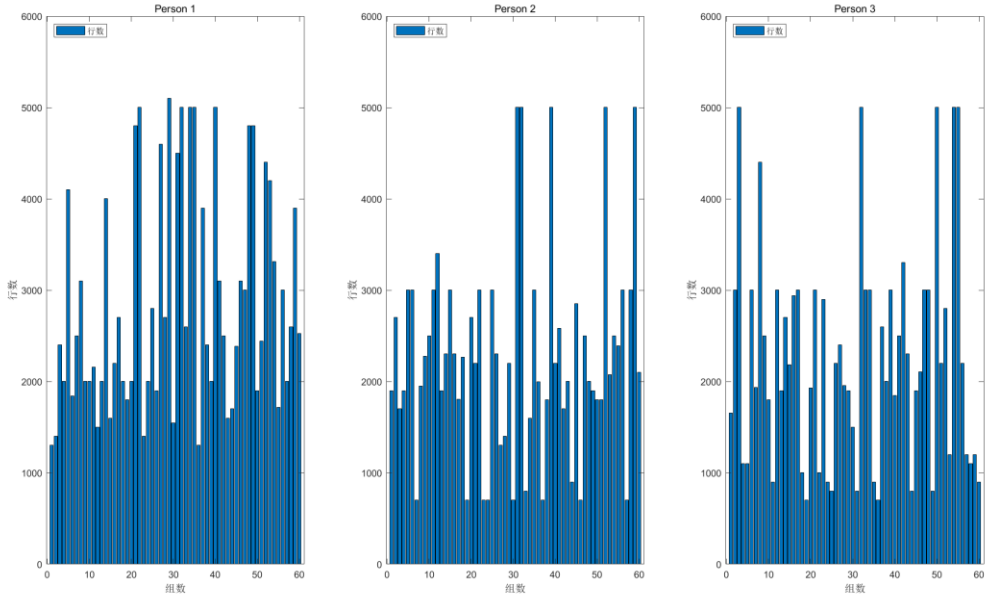


图 4 原始数据维度可视化

具体来说，通过多次实验分析，本文最终提取了 20 个时域及频域关键特征，如表 2 所示。时域特征直接从时间轴上观察信号获得，包括平均值、方差和波形指标等，它们反映了信号的基本特性。频域特征通过傅里叶变换从时间序列中提取，揭示了数据的频率成分，包括主频和频域能量等。主频指示了数据中的主要频率成分，频域能量衡量了各频率成分的贡献。此外，考虑到不同活动状态对时长的影响，还选取了时间序列的长度作为特征之一。

表 2 特征提取的 20 种特征

特征	公式	特征	公式
平均值	$F_1=\frac{1}{n}\sum_{i=1}^{n}x_i$	均方根值	$F_{11}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}$
标准差	$F_2=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$	方根幅值	$F_{12}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}$

特征	公式	特征	公式
最小值	$F_3=\min(x_1,x_2,\ldots,x_n)$	裕度指标	$F_{13}=\frac{1}{n}\sum_{i=1}^{n}\frac{x_i^2}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}$
最大值	$F_4=\max(x_1,x_2,\ldots,x_n)$	波形指标	$F_{14}=\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert}$
中位数	$F_5=\mathrm{Median}(x_1,x_2,\ldots,x_n)$	脉冲指标	$F_{15}=\frac{\max(\lvert x_1\rvert,\lvert x_2\rvert,\ldots,\lvert x_n\rvert)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}$
范围	$F_6=\max(x_1,x_2,\ldots,x_n)-\min(x_1,x_2,\ldots,x_n)$	峰值指标	$F_{16}=\frac{\max(\lvert x_1\rvert,\lvert x_2\rvert,\ldots,\lvert x_n\rvert)}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert}$
偏度	$F_7=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{3/2}}$	峭度指标	$F_{17}=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{2}}$
峰度	$F_8=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right)^{2}}-3$	主频	$F_{18}=\arg\max_f\lvert\mathrm{FFT}(x)\rvert$
峰值	$F_9=\max(\lvert x_1\rvert,\lvert x_2\rvert,\ldots,\lvert x_n\rvert)$	频域能量	$F_{19}=\sum_f\lvert\mathrm{FFT}(x,f)\rvert^2$
绝对平均值	$F_{10}=\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert$	时间序列长度	$F_{20}=n$

3.3 特征标准化

考虑不同组数据特征数值范围和分布各异，进一步对特征数据进行标准化处理能有效增强数据表征和学习能力。本文选择常用的 Z-Score 方法进行数据标准化。该方法通过将特征值转换为符合标准正态分布的形式消除不同特征尺度差异带来的影响。标准化后，各特征的均值调整为 0，标准差调整为 1。Z-Score 方法公式如下所示：

$$z=\frac{x-\mu}{\sigma}(z\sim N(0,1)) \tag{1}$$

3.4 特征降维

在特征提取阶段，本文对加速度计和陀螺仪在三个轴向（X，Y，Z）共 6 类数据中的每一类数据中均提取出了 20 个特征，总计 120 种特征。特征之间存在较大相关性以及冗余性，进一步对这 120 种特征进行特征降维是必要的。因此，本文依次引入了主成分分析（PCA）、ReliefF、自动编码器三种经典的有效降维方法，并通过实验对比选择最佳降维方法。

（1）PCA

PCA 是一种数据降维和特征提取的统计方法，它能够从多维数据中提取关键特征，即主成分。这些主成分不仅捕捉了数据的主要变异性，而且相互独立，避免了多重共线性问题。PCA 的主要步骤包括：1) 数据预处理——对原始数据进行标准化处理，确保各特征具有零均值和单位方差，以消除量纲和尺度的影响；2) 协方差矩阵构建——计算数据的协方差矩阵或相关系数矩阵，以量化特征间的线性关系；3) 特征值分解——对协方差矩阵进行特征值分解，得到特征值和相应的特征向量；4) 主成分选择——根据特征值的大小，选择前 k 个主成分，这些成分解释了数据集中的大部分方差。

在对数据进行标准化后，应用 PCA 方法，如图 5 所示，从 120 个特征中提取了前 20 个主成分，这 20 个主成分的解释方差比达到了 98%，说明降维是合理的。这些成分既包含了数据的主要信息，又通过降维简化了分析过程。

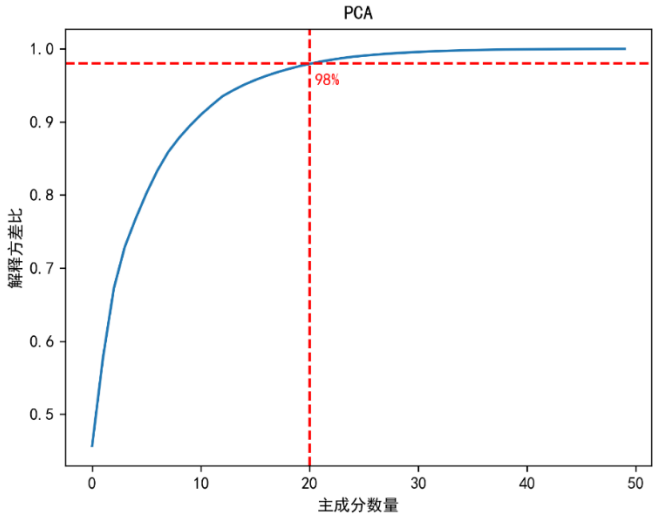


图 5 主成分分析结果

（2）ReliefF 算法

ReliefF 是一种基于实例和邻近样本比较的特征选择方法，通过评估每个特征在区分不同类别样本中的重要性来实现降维。该算法通过评估每个特征在区分不同类别样本中的重要性来进行特征选择。其具体步骤和原理为：

1) 准备数据集：首先，需要准备包含多个特征变量和一个目标变量的数据集。ReliefF 需要有标签的数据，于是本文使用附件 2 的数据进行训练。

2）计算特征权重：使用 ReliefF 算法计算每个特征对所有样本的重要性权重。这通常通过迭代每个样本，并计算其与最近邻和次近邻之间的差异来实现。

3）特征排序：根据计算出的特征权重，对特征进行排序，权重越高的特征被认为越重要。

4）选择重要特征：根据特征的重要性排序，选择出排名靠前的特征作为重要特征。.

（3）自动编码器方法

自动编码器（Autoencoder）是一种无监督的神经网络模型，广泛用于数据的压缩、特征学习及降维。它通过编码器和解码器的组合，学习输入数据的低维表示（编码），并尝试从该低维表示中重构原始输入数据（解码）。自动编码器在降维方面的应用尤为突出，因为它能够在保留数据关键信息的同时，显著降低数据的维度。其原理如下所示：

1) 编码器：输入数据首先被编码器转换为一个低维的内部表示，这个过程称为编码。编码器通常是一个前馈神经网络，其隐藏层的节点数少于输入层和输出层，从而实现降维。

2) 瓶颈层：编码器的输出是一个低维的内部表示，这一层被称为瓶颈层或编码层。

3) 解码器：瓶颈层的输出随后被解码器用来重建原始输入数据。解码器通常与编码器结构相反，逐步增加节点数，直到达到原始数据的维度。

4) 重建损失：自动编码器的训练目标是最小化输入数据和重建数据之间的差异。这通常通过一个损失函数来实现，如均方误差（MSE）或交叉熵损失。

5) 训练过程：通过反向传播算法调整网络的权重，以最小化重建损失。

（4）对比结果

先后利用 PCA、ReliefF、自动编码器三种方法对附件 2 提供的数据集进行降维，并利用问题 2 建立的判别模型对相应的降维数据进行判别分类，得到分类准确度结果如表 3 所示。通过对比下表中的数据发现 PCA 在降维过程中表现出了最优的性能。因此，本文选择 PCA 作为特征降维方法。

表 3 三种降维方法效果对比表

降维方法	PCA	ReliefF	自动编码器
分类准确度	**98%**	85%	77%

4 问题一分析与模型建立

4.1 问题分析

问题一要求对附件 1 中 3 名实验人员的运动数据进行分类，目标是识别数据间的相似性和差异性，以区分不同活动类型，形成 12 个类别的数据集。考虑该问题不存在标签数据，本文拟采用聚类分析方法对问题进行求解，具体思路流程如图 6 所示：本文基于预处理的数据提出约束 K-means 方法对特征数据进行聚类，实现人员类别的无监督分类。特别地，在初始化过程中，本文通过引入 K-means++方法来改善聚类中心的初始化效果。聚类结果采用 MDS 方法进行可视化分析。

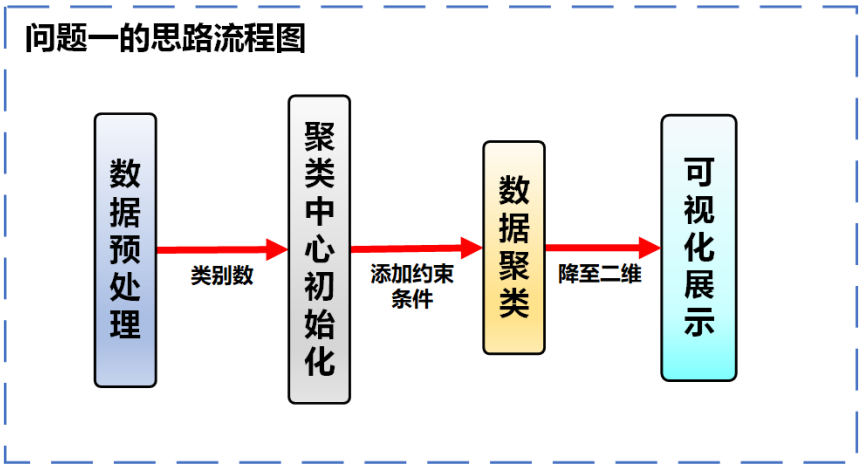


图 6 问题一求解思路示意图

4.2 模型建立

在模型建立阶段，本文构建了约束 K-means 聚类模型以及 K-means++簇心初始化模型，方法流程图如图 7 所示。其中，约束 K-means 模型是 K-means 算法的扩展，通过增加约束条件，强制数据点归属于或排除于特定聚类中心。由于簇中心点的初始化对聚类结果具有决定性影响，本文采用了 K-means++方法来启发式地选择初始中心点。K-means++通过概率机制选择与现有中心点距离较远的数据点作为新中心，以此提高中心点的多样性和代表性，克服了传统 K-means 算法对初始点选择敏感的问题。

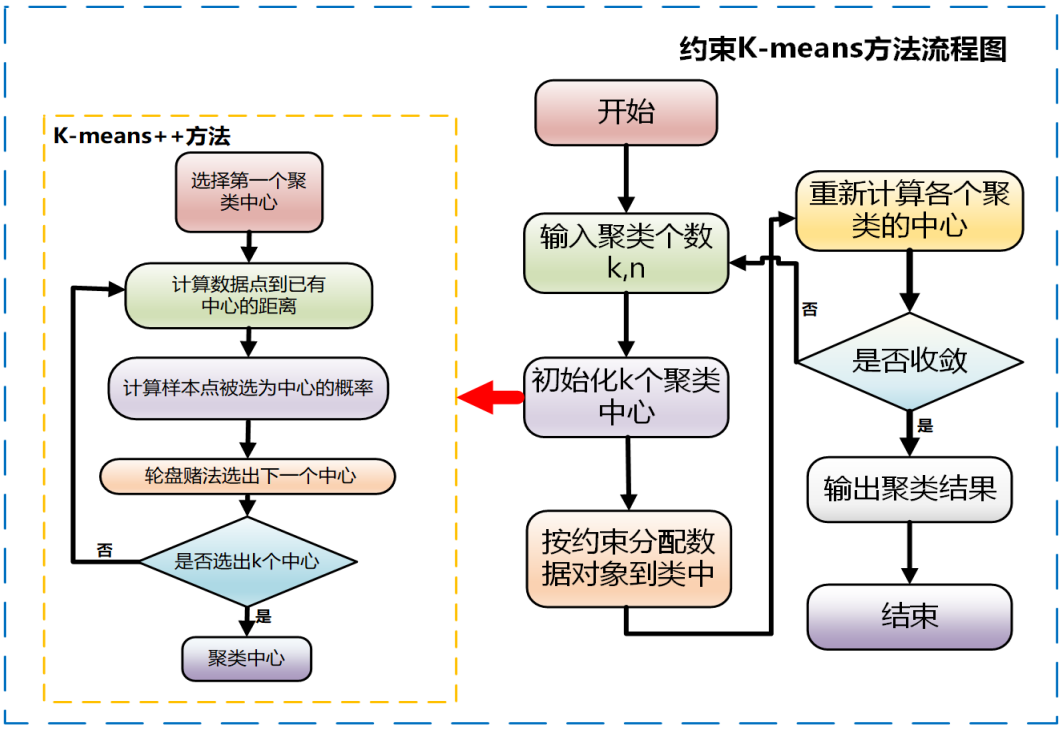


图 7 约束 K-means 方法流程图

4.2.1 约束 K-means 模型

约束 K-means（Constrained K-means）是标准 K-means 聚类算法的改进版[1]，它通过引入硬性（严格限制每个聚类的大小）或软性（倾向于满足但不强制）约束条件来满足特定的聚类需求，如控制簇的大小或定义数据点间的关系。与传统 K-means 不同，约束 K-means 在执行中考虑这些约束，确保聚类结果满足既定条件。

对于某一实验者，设其不同人体活动状态有 K 种，每种活动状态有 N 组数据，则共需对这 $M=N\times K$ 组数据进行聚类分析。具体实现过程如算法 1 所示，首先需要输入预处理后的数据集 D、聚类数 K，硬约束集 H。然后使用 K-means++方法初始化 K 个簇中心点，并设定每个聚类的大小约束为 N，迭代次数 $\lambda=0$。随后计算每个数据点到各聚类中心的距离，并根据最近原则分配数据点至相应聚类，同时检查是否满足大小约束。如果分配违反约束，则选择下一个最近的聚类中心，直至满足条件。此后，重新计算每个聚类的中心，即该聚类所有数据点的平均值。重复上述步骤，直至聚类中心稳定或达到最大迭代次数，输出聚类结果。

算法 1 约束 K-means 算法伪代码

算法 1：约束 K-means 算法

输入： 数据集 $D=\{x_1,x_2,...,x_M\}$，聚类数 K，硬约束集 $H=\{h_1,h_2,...,h_K\}$

输出： 每个数据点的所属聚类，每个聚类的中心点 $C=\{c_1,c_2,...,c_K\}$

使用 K-means++初始化聚类中心 C_0，设置迭代次数 $\lambda=0$

for $x_i\in D$ **do**

计算其到每个聚类中心 c_j 的距离 d_{ij}；

最小化 d_{ij}，将 x_i 分配给 c_j；

根据约束 h_j，调整数据点的分配情况，确保满足约束条件；

更新每个聚类的中心为其成员点的平均值；

迭代次数 $\lambda=\lambda+1$

if $\lambda\geq\lambda_{\max}$ or 聚类中心的变化 $C_i-C_{i-1}\leq$ 阈值

算法终止

end if

end for

4.2.2 K-means++ 方法

为优化聚类中心的初始化，本文在约束 K-means 模型中应用了 K-means++方法[2]，以选取更具代表性的初始聚类中心，提升算法的收敛速度和效果。K-means++是对 K-means 算法的改进，旨在克服传统的 K-means 算法随机初始化可能导致的局部最优问题。K-means++的初始化策略如算法 2 所示，首先从数据集中随机选取一个点作为第一个聚类中心。然后计算每个样本到现有聚类中心的最短距离。根据距离，使用概率比例和轮盘赌方法选择下一个聚类中心，以此增加中心点的分布均匀性。重复步骤，直至选出 K 个聚类中心。

算法 2 K-means++算法伪代码

算法 1：K-means++算法

输入： 数据集 $D=\{x_1,x_2,...,x_M\}$，聚类数 K

输出： 聚类中心 $C_0=\{c_1,c_2,...,c_K\}$

随机选择 $x_i(x_i\in D)$ 作为第一个聚类中心 c_1

算法 1：K-means++算法

for $l = 2,...,K$ **do**

初始化数组 Q，将所有值设为 0

for $x_i \in D$ **do**

计算 x_i 到已选择的最近聚类中心的距离 q_i

更新 $Q(x_i) = q_i$

计算 Q 中所有距离的综合 Q_{sum}

计算 x_i 被选为下一个聚类中心的概率 $P(x_i)$， $P(x_i)=Q(x_i)/Q_{sum}$

使用概率 $P(x_i)$ 结合轮盘赌方法选择下一个聚类中心 c_l

end for

end for

4.2.3 聚类效果评估指标

为了全面评估聚类效果，本文采用聚类分析中常用的三种效果评估指标：DB（Davies-Bouldin）指数，归一化互信息（Normalized Mutual Information, NMI），以及轮廓系数（Silhouette Coefficient）来进行评价，这三个指标的概念如下：

1) **DB 指数：**DB 指数是一种用于评估聚类结果的有效性的指标。它通过计算簇内的紧密度和簇间的分离度来衡量聚类的质量。较低的 DB 指数表示簇内紧密度高、簇间分离度好，即聚类效果较好。其计算公式如下：

$$\mathrm{DBI} = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{2}$$

其中：

k 是簇的数量；

c_i 是第 i 个簇的中心点；

σ_i 是第 i 个簇中所有样本到中心点 c_i 的平均距离；

$d\left(c_i, c_j\right)$ 是簇中心 c_i 和 c_j 之间的距离。

2) **归一化互信息：**归一化互信息是一种用于评估聚类结果与真实类标之间的相似度的指标。它考虑了聚类结果和真实类标之间的一致性和完整性，数值范围在 0 到 1 之间，越接近 1 表示聚类结果与真实类标的一致性越高。其计算公式如下：

$$\mathrm{NMI} = \frac{I(C,K)}{\sqrt{H(C)\cdot H(K)}} \tag{3}$$

其中：

$I\left(C,K\right)$ 是聚类结果 K 和真实类标 C 之间的互信息；

$H(C)$ 和 $H(K)$ 分别是真实类标和聚类结果的熵。

3) **轮廓系数：**轮廓系数是一种用于评估聚类结果紧密度和分离度的指标。它结合了簇内的距离和簇间的距离，通过计算每个样本的轮廓系数来衡量聚类的紧凑性和分离度。

轮廓系数的取值范围在-1 到 1 之间，较高的轮廓系数表示聚类结果较好，簇内距离紧凑且簇间距离较远。其计算公式如下：

$$s(i)=\frac{b(i)-a(i)}{\max(a(i),b(i))} \tag{4}$$

其中：

$a(i)$ 是样本 i 到同簇其他样本的平均距离（簇内平均距离）；

$b(i)$ 是样本 i 到最近其他簇的所有样本的平均距离（簇间平均距离）。

4.3 模型求解

根据本题数据要求，需要求解 12 个簇，每个簇代表一种人体活动状态，设定 $K=12$。由于每种活动状态对应 5 组数据，约束 K-means 模型的约束条件设为每个簇内数据点数小于等于 5。求解步骤包括：数据预处理、K-means++方法初始化聚类中心、约束 K-means 模型聚类、利用可视化手段对聚类结果进行分析。

4.3.1 聚类中心初始化

首先用 K-means++方法对聚类中心初始化，结果如图 8 所示。

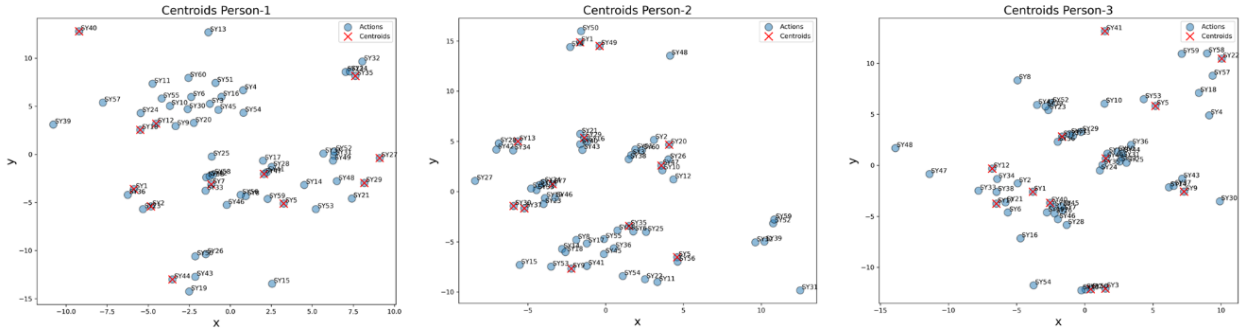


图 8 聚类中心初始化的结果图

在图 8 中，红色“×”代表使用 K-means++方法初始化的 12 个簇中心。浅蓝色圆点代表待聚类的数据集，每人包含 60 个数据点。聚类过程中，将依据数据点到这 12 个簇中心的最短距离进行分类，并用相同颜色表示同一类别的数据点，从而将数据集划分为 12 个类别。

为进一步验证 K-means++簇心初始化方法的有效性，本节对比了 K-means++初始化方法和随机初始化方法对附件 2 中 10 名人员数据的聚类效果的影响，结果如图 9 所示。从图中可以看出，使用 K-means++进行初始化几乎在所有数据样本中得到的分类准确率都更高。

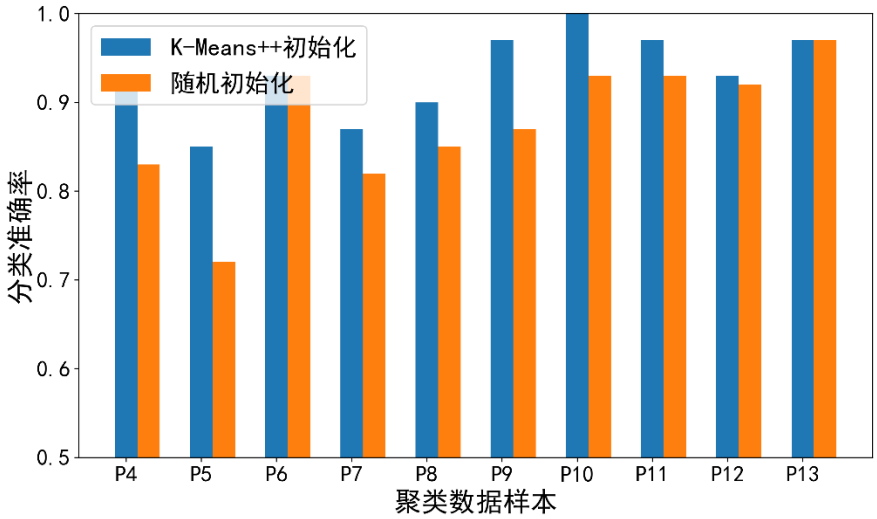


图 9 不同初始化方法对比图

4.3.2 聚类分析

基于得到的初始聚类中心，本文紧接着采用约束 K-means 模型对数据进行聚类分析，将参与者分为 Person1、Person2 和 Person3 三部分，并对每部分数据应用同一模型。

为优化模型展示效果，本文采用多维尺度分析（MDS）技术对数据进行降维[3]，以便可视化。MDS 技术通过保持数据点间的距离关系，将高维数据嵌入到低维空间（如二维或三维），便于数据可视化和模式识别。其主要原理是通过最小化目标函数，促使算法不断寻找满足条件的低维空间下的坐标矩阵，最终得到 2 维空间中的坐标矩阵，用于可视化分析聚类效果。

图 10 呈现了 Person1、Person2 和 Person3 的人体活动状态聚类结果。每个子图中包含 60 个数据点，每个点代表一类活动状态的单组人体活动。数据点按 12 种颜色区分，每种颜色代表一种活动状态，每种状态有 5 个数据点。该图清晰地显示了不同实验中每种活动状态的一致性和稳定性，证明了聚类算法的有效性。数据点的紧密聚集也显示了它们在特征空间内的高相似性。并且我们用建立的聚类效果评价指标对结果进行评价，如表 4 所示，Person1、Person2 和 Person3 的 DB 指数均较低，归一化互信息的值很接近 1，轮廓系数也较高，说明三名实验人员的聚类效果均较好，进一步确认了聚类结果的准确性和可靠性。

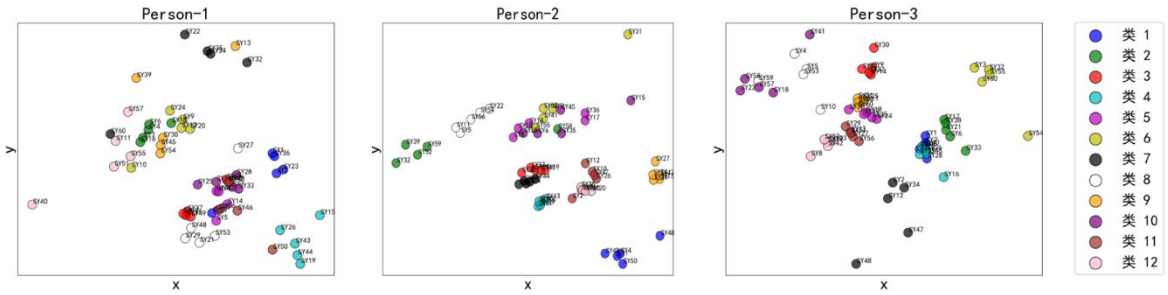


图 10 聚类结果图

表 4 聚类分析指标结果

	DB 指数（Min）	归一化互信息（Max）	轮廓系数（Max）
取值范围	**[0,∞)**	**[0, 1]**	**[-1, 1]**
Person1	0.25	0.75	0.48
Person2	0.14	0.67	0.59
Person3	0.13	0.69	0.24

表 5 具体展示了问题一的聚类结果，表中的每个分类标识符SY*i*代表了某一实验人员第*i*次实验的特定的活动状态。

表 5 问题一结果

分类	Person1	Person2	Person3
第一类	SY1, SY2, SY23, SY36, SY59	SY1, SY4, SY48, SY49, SY50	SY1, SY7, SY20, SY28, SY40
第二类	SY3, SY4, SY6, SY16, SY18	SY32, SY39, SY52, SY58, SY59	SY6, SY17, SY21, SY33, SY38
第三类	SY31, SY37, SY49, SY52, SY58	SY19, SY23, SY30, SY37, SY46	SY9, SY14, SY30, SY37, SY43
第四类	SY15, SY19, SY26, SY43, SY44	SY16, SY21, SY29, SY40, SY43	SY15, SY16, SY26, SY45, SY46
第五类	SY5, SY8, SY14, SY41, SY47	SY8, SY14, SY17, SY18, SY36	SY19, SY24, SY35, SY36, SY49
第六类	SY9, SY10, SY12, SY20, SY24	SY9, SY31, SY41, SY53, SY55	SY3, SY32, SY50, SY54, SY55
第七类	SY22, SY32, SY34, SY35, SY60	SY7, SY24, SY33, SY44, SY57	SY2, SY12, SY34, SY47, SY48
第八类	SY21, SY27, SY29, SY48, SY53	SY5, SY11, SY22, SY54, SY56	SY4, SY5, SY10, SY53, SY59
第九类	SY13, SY30, SY39, SY45, SY54	SY13, SY27, SY28, SY34, SY42	SY11, SY25, SY31, SY44, SY60
第十类	SY7, SY17, SY25, SY28, SY33	SY6, SY15, SY25, SY35, SY45	SY18, SY22, SY41, SY57, SY58
第十一类	SY38, SY42, SY46, SY50, SY56	SY2, SY10, SY12, SY26, SY47	SY13, SY27, SY29, SY51, SY56
第十二类	SY11, SY40, SY51, SY55, SY57	SY3, SY20, SY38, SY51, SY60	SY8, SY23, SY39, SY42, SY52

5 问题二分析与模型建立

5.1 问题分析

问题二由 3 个子问题组成，分别是：

1) 从附件 2 的数据中提取 12 种活动状态的特征，构建一个精确的判别模型，以识别实验者的活动状态；

2) 运用问题一中的分类模型处理问题二的数据，比较分类模型与判别模型的结果，并分析分类模型在不同活动类型上的准确度；

3) 使用判别模型对附件 3 提供的某实验人员的活动状态数据进行判别，给出其对应的活动状态，并将结果填入表格。

对于第一个子问题，本文构建了一个集成学习模型，其中包含神经网络、决策树、随机森林、朴素贝叶斯以及 K 近邻机器学习子模型，子模型基于统一训练集进行离线训练，在问题求解中采用加权投票机制综合形成最终的判别结果。验证集用于评估并调整模型；

对于第二个子问题，首先建立模型评价指标，分别使用问题一建立的约束 K-means 聚类模型以及本节提出的集成学习模型对附件 2 的数据进行分类，进而比较两个模型的分类准确度；

对于第三个子问题，将集成学习模型应用于附件 3 的验证数据，进行判别并得出结果。

问题二的求解思路如下：

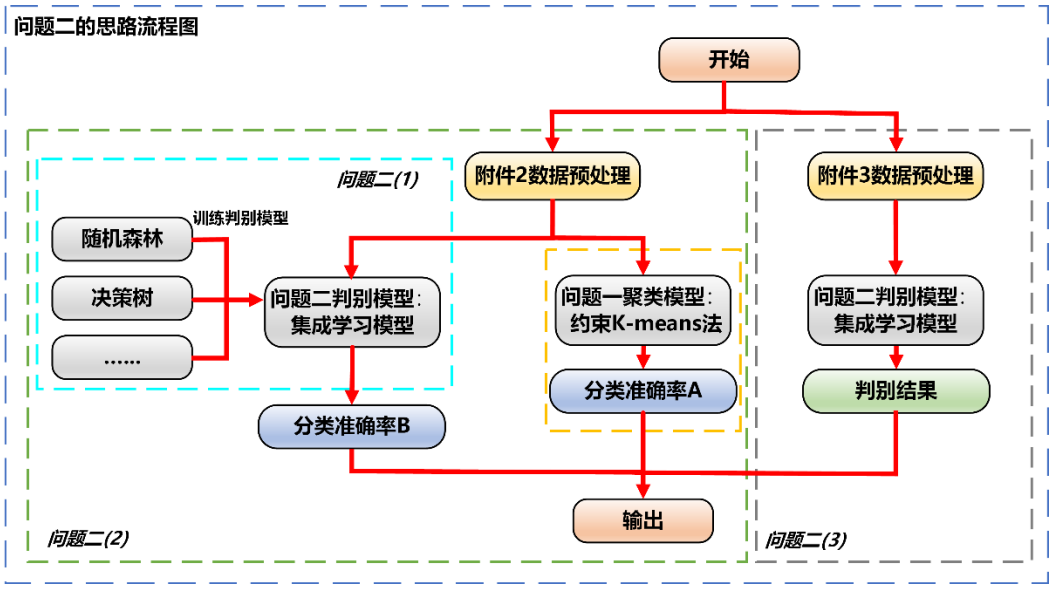


图 11 问题二求解思路示意图

5.2 模型建立

5.2.1 模型准备：评价指标

评估分类模型性能时，准确率（Accuracy）是一个关键指标，定义为模型正确分类样本数占总样本数的比例，其数学表达式为：

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{5}$$

N_{correct} 是模型正确分类的样本数量，在本题中为被正确分类的数据组数。N_{total} 是模型总共分类的样本数量，本题中为实验数据的总组数。

5.2.2 模型构建：集成学习模型

集成学习模型是一种机器学习方法[4]，它通过结合多个基本模型的结果来做出最终的决策。这种方法能够利用各个模型的优势，提升整体模型的性能和稳定性，常见的集成策略包括投票（Voting）、Bagging 和 Boosting 等。集成学习广泛应用于各种领域，特别是在需要处理复杂数据关系和提高预测准确度的应用场景中表现出色。

在解决问题二时，本文构建了一个集成学习模型（见图 12），该模型结合了神经网络、决策树、随机森林、朴素贝叶斯和 K-最近邻（KNN）五种算法。在各模型得出结果后通过投票机制进行综合，形成最终的决策。各模型独立预测后，通过投票机制确定最

终决策。在所采用的机器学习子模型中，神经网络能够处理复杂的非线性模式，决策树提供清晰的决策逻辑，随机森林通过集成方法提升模型的泛化性，朴素贝叶斯适用于概率预测，而 KNN 作为一种非参数方法，适用于分类非线性问题。这种融合多种模型的方法可以显著提升预测的准确性和鲁棒性。

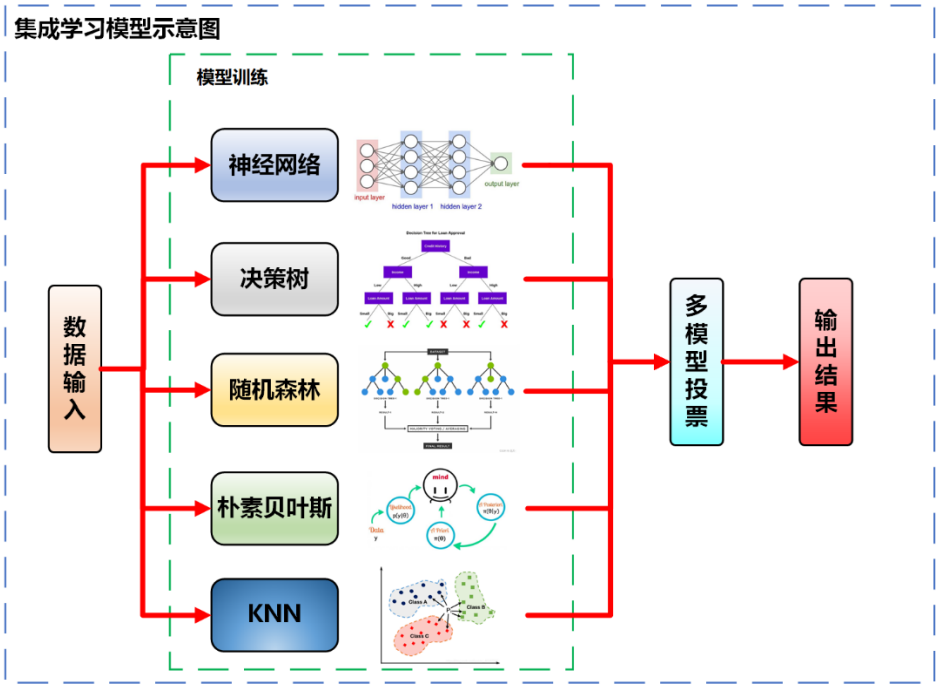


图 12 集成学习模型示意图

（1）神经网络

神经网络是一种包含输入层、多层隐藏层和输出层的复杂计算模型，用于模拟大量神经元的互联。在本问题的处理中，本文采用了卷积神经网络模型，该模型首先通过一维卷积层提取特征，随后引入 ReLU 激活函数以增加非线性，并通过 Flatten 层将数据拉平，进而利用全连接层进行深入分析。为避免过拟合，模型集成了 Dropout 技术，最终输出层应用 softmax 激活函数以实现多类别的预测。此结构适合解决具有多特征和多类别的分类问题，通过不断优化权重和偏置来学习输入数据的模式和特征。

（2）决策树

决策树是一种用于分类和回归的机器学习算法[5]，通过基于特征值的逐层分割构建树状决策结构。它从根节点开始，选择最佳特征以最大化信息增益或最小化不纯度，如使用基尼指数或信息增益等标准，递归地分裂数据直至叶节点，得到预测结果。决策树模型直观易懂，易于解释，但可能面临过拟合的风险，可以通过剪枝等技术进行优化以提高模型的泛化能力。

（3）随机森林

随机森林是一种集成学习方法[6]，用于解决分类和回归问题。它基于多个决策树构建而成，每棵树通过随机选择数据子集和特征子集进行训练。在预测时，随机森林将每棵树的结果进行整合，通常采用投票方式（分类问题）或平均方式（回归问题）来确定最终输出。这种方法能够有效地减少过拟合风险，提高模型的泛化能力，适用于处理大规模数据和复杂特征的机器学习任务。

（4）朴素贝叶斯

朴素贝叶斯是一种基于贝叶斯定理的分类算法[7]，其核心假设是特征之间相互独立。它通过计算每个类别的先验概率和在给定类别下各特征的条件概率来进行分类。在训练过程中，朴素贝叶斯模型利用训练数据估计这些概率，并在预测时根据贝叶斯定理计算每个类别的后验概率，从而选择具有最高后验概率的类别作为预测结果。尽管特征独立的假设在现实中通常不成立，但朴素贝叶斯算法在实际应用中仍然表现出较好的效果，特别是对于文本分类等任务。

（5）KNN

KNN 算法是一种直观的分类和回归方法[8]，其核心思想是在特征空间中找到一个新的数据点的 K 个最近邻居，并基于这些邻居来预测新数据点的类别或数值。KNN 的工作原理简单易懂：首先确定参数 K，然后计算待分类点与其他所有点之间的距离，选择距离最近的 K 个点，最后根据这些点的已知类别或数值进行决策。KNN 的优势在于它的简单性、无需训练数据、对数据分布无假设，以及对非线性问题的良好适应性。

5.2.3 模型训练：有监督学习

将输入数据 x 通过网络进行前向传播，计算每一层的输出，最终得到模型的预测输出 $\hat{y}$。对于一个简单的全连接网络，这个过程可以用如下公式表示：

其中：$\hat{y} = g\left(W^{(l)}x + b^{(l)}\right)$

- 是第输入数据；
- $W^{(l)}$ 是第 l 层的权重矩阵；
- $b^{(l)}$ 是第 l 层的偏置向量；
- g 是激活函数（如 ReLU、Sigmoid 等）。

最终输出层的激活值 $\hat{y}$ 即为模型的预测结果。将模型的预测输出 $\hat{y}$ 与真实标签 y 进行比较，计算损失函数的值 $L\left(\hat{y}, y\right)$。常见的损失函数包括均方误差（MSE）和交叉熵损失等。对于分类问题，交叉熵损失可以表示为：

$$L\left(\hat{y}, y\right) = -\frac{1}{N}\sum_{i=1}^{N}\left[y^{(i)}\log\left(\hat{y}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\hat{y}^{(i)}\right)\right] \tag{6}$$

通过反向传播算法，计算损失函数相对于模型参数 θ 的梯度 $\nabla_{\theta}L$。反向传播过程利用链式法则，从输出层开始，逐层向后计算每一层的梯度。具体公式例如：

$$\delta^{(l)} = \frac{\partial L}{\partial a^{(l)}} \cdot g'\left(z^{(l)}\right) \tag{7}$$

其中：

- $\delta^{(l)}$ 是第 l 层的误差项；
- $z^{(l)}$ 是第 l 层的线性组合 $W^{(l)}a^{(l-1)} + b^{(l)}$。

权重和偏置的梯度可以表示为：

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)}\left(a^{(l-1)}\right)^{T} \tag{8}$$

$$\frac{\partial L}{\partial b^{(l)}} = \delta^{(l)} \tag{9}$$

利用梯度下降算法，根据计算得到的梯度调整模型的参数，以最小化损失函数。标准梯度下降法的参数更新公式为：

$$\theta := \theta - \eta \nabla_\theta L \tag{10}$$

其中η为学习率，$\nabla_\theta L$是损失函数关于参数的梯度。上述过程（前向传播、损失计算、反向传播、参数更新）在整个训练数据集上反复迭代多次（称为训练轮次或 epoch），直至模型在训练集上的损失函数收敛到一个较低的值，或者达到预设的训练轮次。

5.3 模型求解和分析

根据问题二的分析，首要任务是构建并训练一个判别模型，这一模型的建立是解决子问题二和子问题三的基础。

5.3.1 子问题一：集成学习模型训练

将处理好的数据集划分为 70%的训练集和 30%的验证集，基于训练集对集成模型中的五个子模型分别进行训练，其训练过程和分类准确率如图 13 和图 14 所示。基于五个子模型的分类结果，结合其训练过程中的分类准确率，采用加权投票机制综合产生最终分类结果。

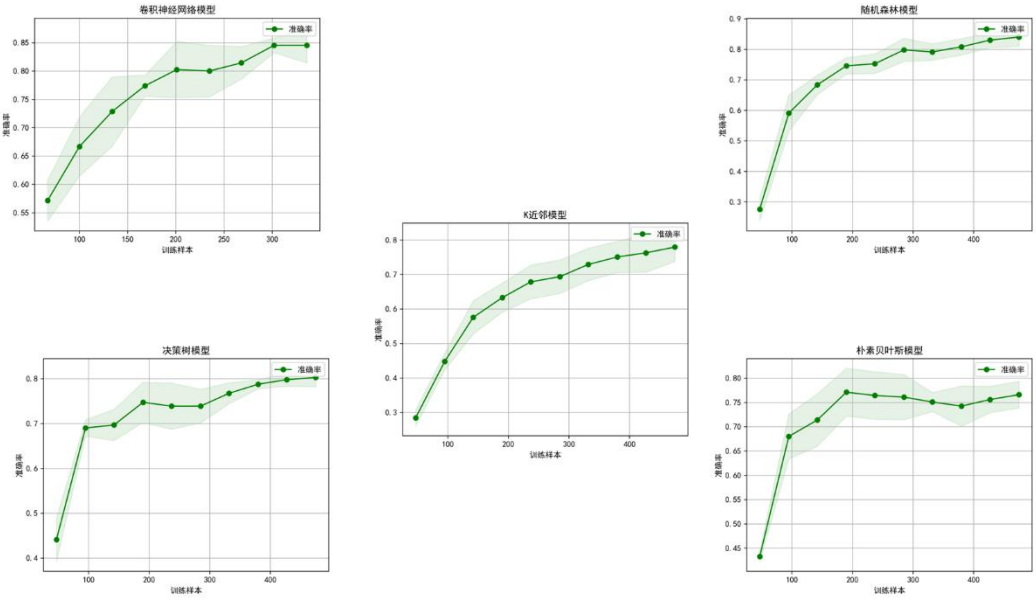


图 13 基本模型训练过程

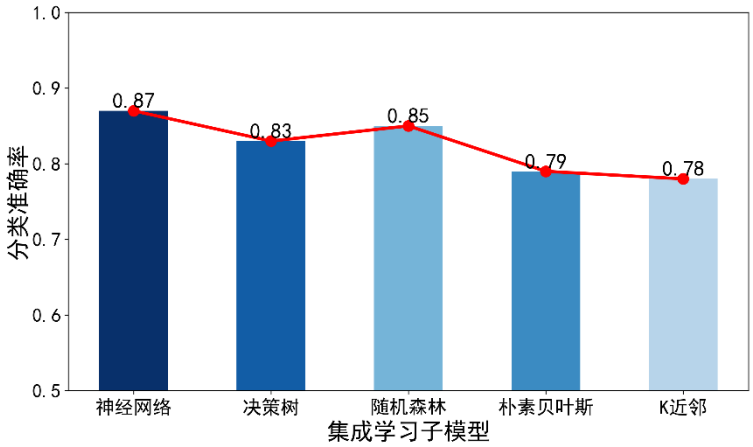


图 14 集成模型子模型预测准确率

5.3.2 子问题二：模型比较

（1）使用问题一聚类模型（约束 K-means 模型）进行分类

子问题二要求使用问题一中的约束 K-means 聚类模型对 10 名实验人员的数据进行分类，但该模型仅能分组相似数据，无法直接标记特定活动状态标签。为解决这一问题，我们结合活动状态标签，采用匈牙利算法对 K-means 模型的聚类结果进行标签分配。分配标签后的具体分类结果详见【附录 A】。

匈牙利算法常用于解决二分图匹配问题[9]。如图 15 所示，其通过定义 X 集合中的数据 $X_1, X_2, ..., X_n$ 和 Y 集合中的数据 $Y_1, Y_2, ..., Y_n$ 的匹配原则，旨在一个二分图中找到一个权重最大（或最小）的完美匹配。这个算法在机器学习和数据分析中常用于解决线性分配问题，例如在聚类任务中计算聚类准确性。

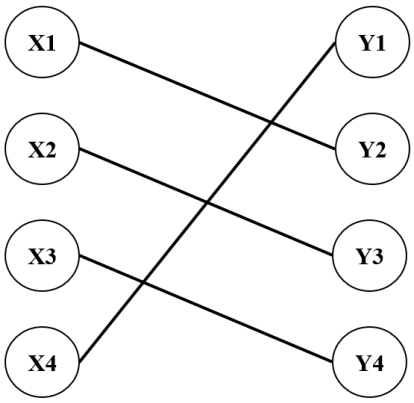


图 15 匈牙利算法解决匹配问题示意图

为得到使用问题一聚类模型（约束 K-means 模型）进行分类的准确率，我们采用混淆矩阵进行准确率计算和可视化展示。混淆矩阵常用于评估分类模型性能[10]，其通过表格形式对比预测结果与实际类别，其原理如表 6 所示，其含义为：

- 真阳性（True Positives，TP）：正确地分类为正类的样本数。
- 假阳性（False Positives，FP）：错误地分类为正类的样本数（即实际为负类，但被分类为正类）。
- 真阴性（True Negatives，TN）：正确地分类为负类的样本数。

- 假阴性（False Negatives，FN）：错误地分类为负类的样本数（即实际为正类，但被分类为负类）。

表 6 混淆矩阵

	预测为正类	预测为负类
实际为正类	真阳性 (TP)	假阴性 (FN)
实际为负类	假阳性 (FP)	真阴性 (TN)

图 16 以可视化方式展示了使用混淆矩阵对问题一中的约束 K-means 聚类模型在附件 2 数据集上分类的结果。考虑到问题二包含 10 名实验人员，每人完成 12 种活动状态，每种状态 5 组数据，形成了 12 个类别，每类别 50 组数据。图 16 中，横坐标列出了模型预测的类别，纵坐标对应实际类别，矩阵中的数值表示正确分类的数据组数。例如，第一行第一列的“50”表示第一类状态的所有数据均被正确分类，准确度达到 100%。整体分类准确度为 93%，如图表标题所示，显示了模型的可靠性。【附录 A】详细记录了这些分类结果。

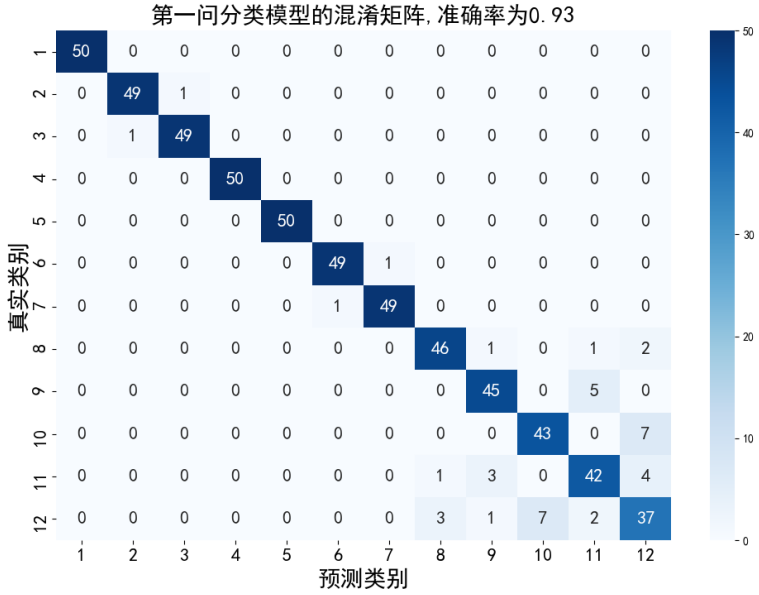


图 16 问题一分类模型的混淆矩阵

（2）使用问题一判别模型（集成学习模型）进行分类

为得到分类准确率 B，接下来使用训练好的集成学习模型对附件 2 数据集进行分类，得到的最终结果如【附录 B】所示，将其用混淆矩阵进行可视化后的结果图如图 17 所示

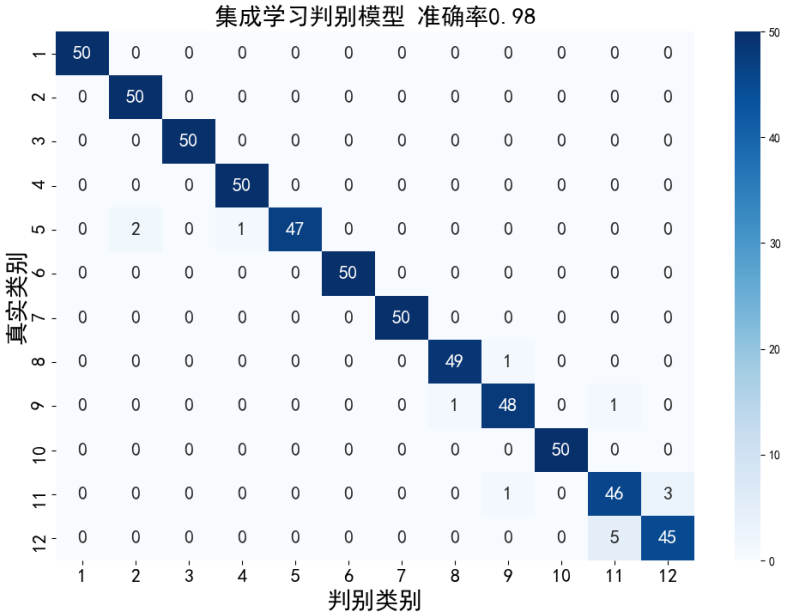


图 17 集成学习判别模型的混淆矩阵

分析该结果可知，集成学习模型的准确率达到 95%，混淆矩阵的主对角线元素集中在 50 附近，分类效果非常显著。

3) 比较分析两种分类模型

采用问题一中的约束 K-means 模型结合匈牙利最大匹配算法得到的分类准确度为 93%，而采用集成学习模型得到的分类准确度可以达到 95%，由此可见，采用集成学习模型对不同活动类型分类时的准确度更高，效果更好。

5.3.3 子问题三：状态判别

对于问题二的子问题三，需要分类的数据为附件 3 中收集的某实验人员 30 次活动的状态数据，用所建立的集成学习模型进行分类，得到的结果如表 7 所示

表 7 问题二第二小问的结果

活动类型	判别状态	活动类型	判别状态	活动类型	判别状态
SY1	3-向右走	**SY11**	9-站立	**SY21**	6-向前跑
SY2	1-向前走	**SY12**	7-跳跃	**SY22**	2-向左走
SY3	7-跳跃	**SY13**	4-步行上楼	**SY23**	5-步行下楼
SY4	7-跳跃	**SY14**	3-向右走	**SY24**	2-向左走
SY5	7-跳跃	**SY15**	4-步行上楼	**SY25**	2-向左走
SY6	10-躺下	**SY16**	1-向前走	**SY26**	9-站立
SY7	2-向左走	**SY17**	4-步行上楼	**SY27**	10-躺下
SY8	6-向前跑	**SY18**	5-步行下楼	**SY28**	2-向左走
SY9	7-跳跃	**SY19**	10-躺下	**SY29**	6-向前跑

SY10	10-躺下	**SY20**	9-站立	**SY30**	3-向右走

6 问题三分析与模型建立

6.1 问题分析

问题三由 3 个子问题组成，分别是：

1) 根据附件 4 给出的问题一和问题二中参与实验的 13 位实验人员的年龄、身高、体重等数据，分析不同人员的同一活动状态是否存在差异；

2) 根据附件 4 的人员数据，分析活动状态数据与实验人员的年龄、身高、体重有无关系，并探究能否使用活动传感器数据进行人员画像；

3) 结合附件 5 给出的问题二的 10 位实验人员中的 5 位的某次活动数据，建立一个判别模型，判断他们分别最可能来源于问题二中哪一名实验人员，并将结果填入表 3。

相应求解思路如下：

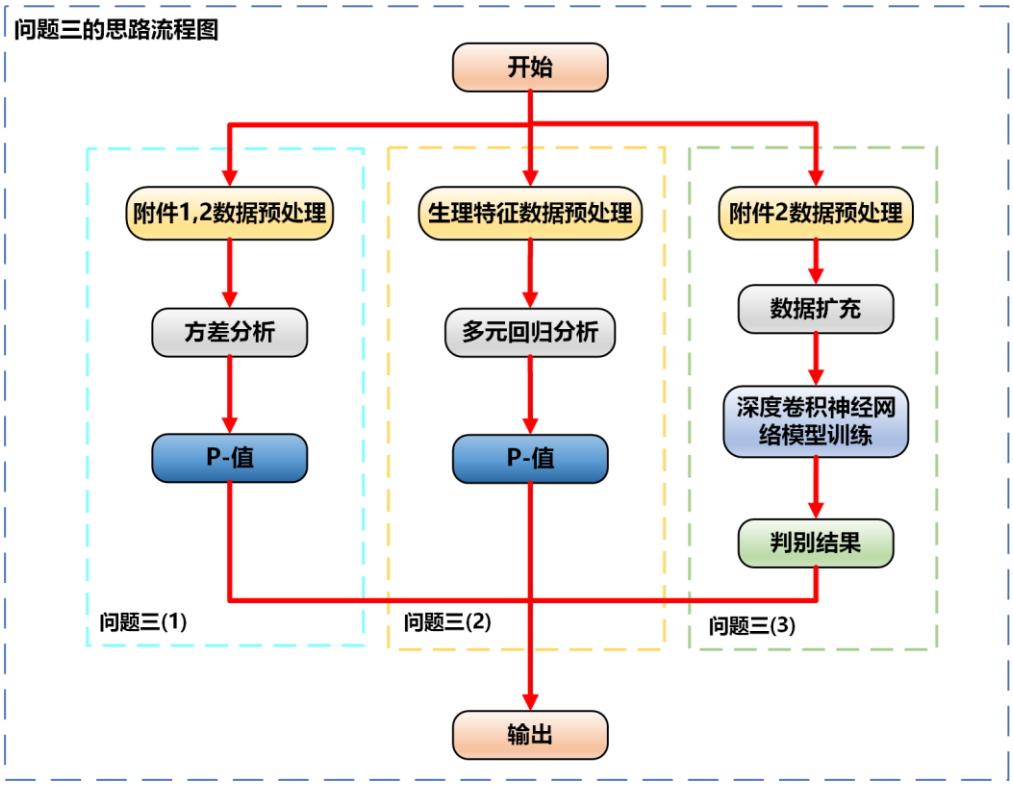


图 18 问题三求解思路示意图

1) 对于子问题一，需要分别判断 12 种活动状态下的人员数据是否存在差异。首先按活动状态对数据进行分类整理，确保每种状态下涵盖 13 名人员的数据，每人 5 组数据。然后合并每人的 5 组数据，并应用方差分析。计算 F 值和 P 值。最后根据 P 值进行差异性判断，若 P 值低于显著性水平，则认为不同活动状态下的数据存在显著差异。

2) 对于子问题二，需要分别探究 12 种活动状态下的人员数据和年龄、身高、体重的关系。将人员活动特征结合年龄、身高、体重信息，开展多元回归分析，以 P 值评估活动状态与生理特征间的相关性，若 P 值低于显著性水平，则认为活动状态与个体的生理特征存在显著相关性。

3) 对于子问题三，该问题的目标是基于运动状态数据识别测试数据的实验人员类别。本文对已知标签数据的 10 名人员的活动状态数据进行数据拆分与扩充，构建上万容量的训练数据，将数据标签转换为人员类别，同时构建并训练深度卷积神经网络模型，实现对附件 5 中数据的人员类别判别。

6.2 模型建立

6.2.1 子问题一模型：方差分析模型

方差分析（ANOVA，Analysis of Variance）是一种常用的多变量相关性统计方法，用于比较多个样本均值之间的差异，判断这些差异是否具有统计显著性。它通过分析数据的方差来推断不同组之间均值是否相同。ANOVA 的基本思想是将总方差分解为组间方差和组内方差，然后通过比较这两部分方差来判断组间差异是否显著。其主要步骤包括：

1. 建立假设：零假设（H_0）通常表示所有组的均值相等，备择假设（H_1）表示至少有两个组的均值不等。

2. 计算 F 值和 P 值：F 值是组间变异性与组内变异性的比率，用于评估组间差异的显著性；P 值是用于测试零假设（所有组的均值相等）的统计显著性的概率值。

3. 确定显著性水平：通常使用 0.05 或 0.01 作为显著性水平。

4. 做出结论：如果 F 值超过临界值或 p 值小于显著性水平，则拒绝零假设，认为至少有两个组之间存在显著差异。

6.2.2 子问题二模型：多元回归分析模型

多元回归分析是一种用于评估多个自变量对一个连续因变量影响的统计方法。通过构建数学模型并进行参数估计和检验，可以判断自变量和因变量之间是否存在统计学上的关系。如果模型的统计检验结果显著，并且自变量的系数在个体层面上也显著，则可以认为自变量与因变量之间存在关系。其主要步骤包括：

1. 建立回归模型：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{11}$$

其中，Y 是因变量，X_1，X_2，…，X_P 是自变量，β_0 是截距项，β_1，β_2，…，β_P 是各自变量的回归系数，ϵ 是误差项。

2. 拟合模型：

通过最小二乘法（Least Squares Method）拟合模型，即通过最小化预测值与实际值之间的均方误差（MSE）来确定回归系数 β_i。

3.检验显著性：

（1）使用 F 检验（F-test）来检验整个回归模型的显著性，即是否至少有一个自变量对因变量有显著影响。

（2）查看 p 值（p-value），p 值越小，表明对应的回归系数越显著。通常，如果 p 值小于 0.05，则认为该自变量对因变量有显著影响

6.2.3 子问题三模型：深度卷积神经网络模型

本节构建了一个深度卷积神经网络模型用以对活动数据的人员类别进行分类。相比于一般神经网络模型，得益于其更深的网络结构和更丰富的连接关系，深度卷积神经网络展现出更强大的特征学习能力和信息处理能力，能够捕捉数据中的复杂模式，被广泛

应用于各种高精度分类任务[11]。如图 19 所示，本文构建的深度神经卷积网络由多层神经元组成，首先是三个一维卷积层 (Conv1D)，分别具有 64、64 和 32 个滤波器，每个滤波器大小为 3。在卷积层之后是一个展平层 (Flatten)，将三维的卷积输出转换为一维向量，以便连接到全连接层。后面连续添加了几个全连接层 (Dense)，神经元数量分别为 128、64、32、64 和 128。每个全连接层都使用 ReLU 激活函数，其中间穿插使用了 Dropout 层，以减少过拟合的风险。最后一个全连接层具有 10 个神经元，使用了 softmax 激活函数。

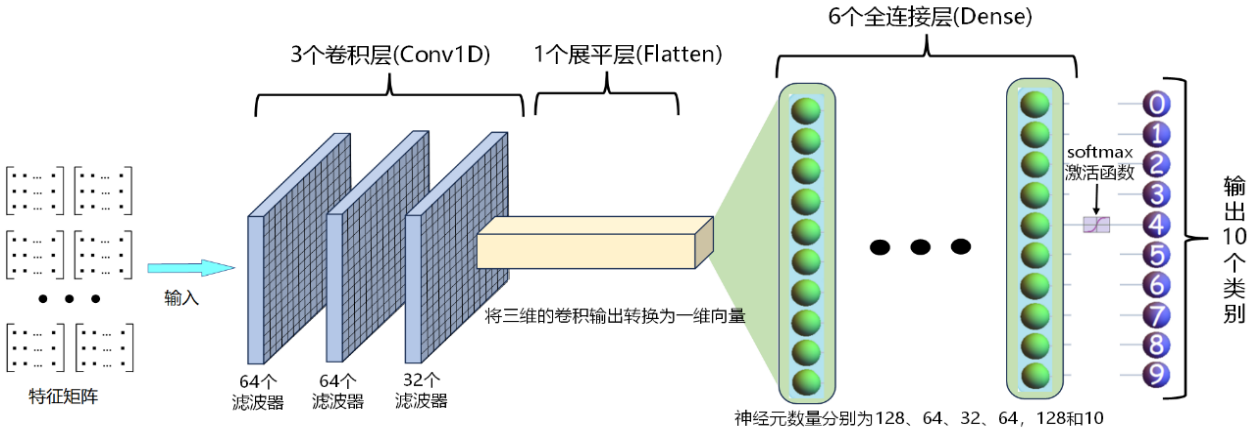


图 19 深度神经网络模型

6.3 模型求解和分析

6.3.1 子问题一：人员活动状态差异性分析

表 8 展示了方差分析的结果，从中可以看出，除“坐下”、“站立”、“躺下”、“乘坐电梯向上移动”和“乘坐电梯向下移动”这 5 种活动状态外，其他活动状态下的不同人员的数据都存在显著差异，且显著性水平均为 0.01。这表明不同人员的同一活动状态存在差异，但差异仅体现在“向前走”、“跳跃”等较为剧烈的活动上。

表 8 方差分析的结果

活动状态	F 值	P 值	显著性水平
向前走	7.32	2.21E-10	0.01
向左走	3.14	9.64E-04	0.01
向右走	2.95	1.83E-03	0.01
步行上楼	2.67	4.60E-03	0.01
步行下楼	3.48	3.05E-04	0.01
向前跑	4.55	3.17e-07	0.01
跳跃	3.21	7.63E-04	0.01
坐下	1.02	4.23E-01	>0.05
站立	1.45	1.64E-01	>0.05
躺下	0.56	8.33E-01	>0.05
乘坐电梯向上移动	0.54	8.46E-01	>0.05

乘坐电梯向下移动	0.58	8.10E-01	>0.05

6.3.2 子问题二：活动状态数据与人员生理特征相关性分析

【附录 C】中的 12 张图以视觉化的方式呈现了在 12 种不同活动状态下，通过多元回归分析得到的 P 值结果。以“向前走”活动状态为例，它详细描绘了该状态与个体生理特征（年龄、身高、体重）之间的关联。图中包含 60 个方块，每个方块对应一个状态特征（数据预处理中选取的主成分）与生理特征的关系，横轴表示状态特征的序号，而纵轴则代表生理特征。方块的颜色深浅直接反映了状态特征与生理特征之间关系的显著性，颜色越深，表明关系越显著。特别地，标记数值的方块表明 P 值小于 0.05，即该方块对应的状态特征与生理特征显著相关。例如，在该图中，三个数值 0.50e-3、0.97e-2 和 0.93e-2 均小于 0.05，进一步证实了显著性。

观察【附录 C】中的图可以发现，某一行中标记数值的方块数量越多，颜色越深，则表明该活动状态与相应的生理特征之间的关联性越强。通过综合分析这 12 张图，我们能够得出活动状态与生理特征之间的相关性总结，具体结果详见表 9。在表 9 中，使用“√”标记表示在 0.05 的显著性水平下，活动状态与生理特征之间存在显著相关性；而“×”则表示二者之间没有显著的相关性。从表中可以看出，较为剧烈的活动与生理特征相关性更大，如“跳跃”、“向前跑”、“向前走”等；而相对静止的活动则与生理特征不显著相关，如“站立”、“乘坐电梯向上移动”和“乘坐电梯向下移动”等。因此，可以使用活动传感器数据进行人员画像。

表 9 活动状态与生理特征之间的相关性总结表

活动状态	年龄	身高	体重
向前走	√	√	√
向左走	√	√	√
向右走	√	×	×
步行上楼	×	×	√
步行下楼	√	×	√
向前跑	×	√	√
跳跃	√	√	√
坐下	×	×	×
站立	×	×	×
躺下	×	×	×
乘坐电梯向上移动	×	×	×
乘坐电梯向下移动	×	×	×

6.3.3 子问题三：人员编号判别

子问题三要求完成识别 5 位特定实验人员的任务。训练此前构建的深度卷积神经网络需要大量带标签的数据样本，而附件 2 仅提供了 10 位实验人员的 12 种活动状态的 5 组时序数据，这意味着最多仅有 600 组带标签样本供模型训练，对于深度神经网络的训

练是远远不够的。考虑到该问题特性，同一人员的同一动作数据具有较大相似性，可假定其服从同一数据分布。因此，本文提出一种数据扩充方法，即对同一人员的 12 种活动状态的 5 组时序数据进行打乱重组，针对每种活动状态各从其 5 组时序数据中进行随机抽样，这样最多可以组合出 5^{12} 种不同的带标签数据，其标签为对应的人员编号。在训练过程中，随机抽取 10000 条上述样本数据，按 70%和 30%的比例划分为训练集和测试集，对深度卷积神经网络展开训练和测试。训练中迭代次数设置为 3000，批次大小设置为 512，监督学习优化器采用 Adam，学习率设置为 0.0005。具体训练过程如下图所示：

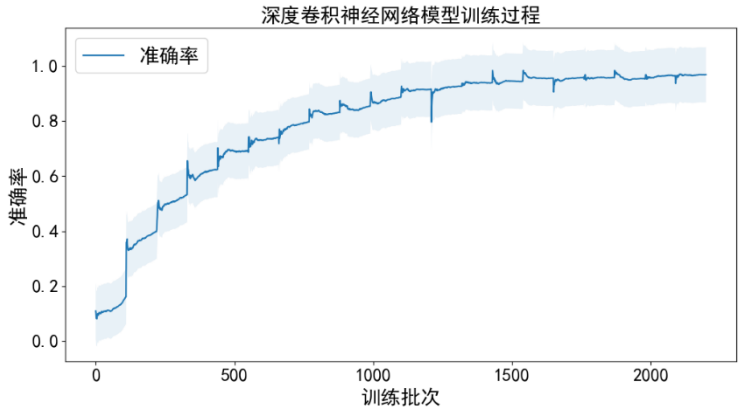


图 20 深度卷积神经网络模型训练过程

将训练好的模型用于子问题三的求解，将 5 个未知编号的活动人员实验数据输入，得到该问题的求解的结果如表 10 所示。

表 10 问题三的结果

活动类型	判别结果
Unkonw1	Person10
Unkonw2	Person7
Unkonw3	Person9
Unkonw4	Person4
Unkonw5	Person5

7 模型评价与改进

7.1 模型的优点

(1) 模型充分结合实际，通过合理的假设，这些假设充分考虑了活动状态的多样性和个体差异，为模型提供了实际场景中的可行性和准确性基础。

(2) 通过特征提取的方法，有效地从原始数据中提取关键特征，通过降维处理使得数据在维度上得以统一，从而简化了模型的复杂度并提升了处理效率。

(3) 模型大量运用可视化方式显示结果，有助于直观展示数据分析和模型输出，提升了结果的可理解性和决策的可靠性。

(4) 本文使用的集成学习模型和深度神经网络模型，具有求解问题高效、准确的特点，有效提升了分类问题的准确性和解决效率。

7.2 模型的不足

(1) 模型的结构较为复杂，涉及多个参数和变量的调整，这增加了模型训练和调试的难度。同时，对于一些非专业用户而言，模型的使用和理解可能存在困难。
(2) 模型对输入数据的质量和完整性依赖性较强，若输入数据存在较大缺失或质量问题，模型的预测性能将显著下降。
(3) 模型在训练过程中准确度较高，这不排除模型可能存在的过拟合问题。这种问题会导致模型在实际应用中表现不如预期，无法有效泛化到新数据或者未能捕捉到数据中的复杂关系。

7.3 模型的改进

针对模型存在的不足，我们将考虑以下改进方案以提升其性能和效率。

首先，针对精度不足、参数过多以及问题解空间优化不足的问题，采用以下策略：优化模型架构和超参数选择，例如使用交叉验证来调整参数以提高模型的泛化能力；同时，考虑使用模型剪枝技术来减少不必要的参数和复杂度，以提高处理大规模数据和复杂任务时的效率。

其次，针对训练过程中可能存在的过拟合问题，建议引入正则化技术，如 L1 或 L2 正则化，以提升网络学习能力和泛化能力。

值得说明的是，我们所提出的集成学习模型以及深度卷积神经网络模型具有很强的可扩展性，随着数据量的增加，模型的性能会进一步提升。在求解问题上，除了本文所研究的人体活动状态判别问题，本文所提模型是一个分类问题的通用求解框架，对于其他类型的多分类任务同样适用。

综上所述，下一步我们将通过这些改进措施的综合应用，显著提升模型在实际应用中的表现和处理能力。

参考文献

[1] 唐海波, 林煜明, 李优. 一种基于 K-Means 的平衡约束聚类算法 [J]. 华东师范大学学报(自然科学版), 2018, 2018(5).

[2] 张玉琴, 张建亮, 冯向东. 基于改进 K-Means++和 DBSCAN 的大数据聚类方法 [J]. 国外电子测量技术, 2022.

[3] 张美霞, 李丽, 杨秀, 等. 基于高斯混合模型聚类和多维尺度分析的负荷分类方法 [J]. 电网技术, 2020, 44(11): 4283–4296.

[4] 逄岩, 许枫, 刘佳, 等. 联合特征选择与改进集成学习模型的海底底质分类 [J]. 声学学报, 2023, 48(01): 83–92.

[5] 谢兆贤, 邹兴敏, 张文静. 面向大型数据集的高效决策树参数剪枝算法 [J]. 计算机工程, 2024, 50(01): 156–165.

[6] 张锟滨, 陈玉明, 吴克寿, 等. 粒向量驱动的随机森林分类算法研究 [J]. 计算机工程与应用, 2024, 60(03): 148–156.

[7] 侯敏, 张仕斌, 黄曦. 量子模糊朴素贝叶斯分类算法 [J]. 电子科技大学学报, 2024, 53(01): 149–154.

[8] Chen Z, Song Z, Zhang T, et al. Retraction Note: IoT devices and data availability optimization by ANN and KNN [J]. EURASIP Journal on Information Security, 2024, 2024(1).

[9] 王欢, 李峰, 宋思盛, 等. 基于匈牙利算法的多分量微动信号提取方法 [J]. ., 2022, 51(04): 32–37.

[10] 张开放, 苏华友, 窦勇. 一种基于混淆矩阵的多分类任务准确率评估新方法 [J]. 计算机工程与科学, 2021, 43(11): 1910–1919.

[11] 吴欢欢, 谢瑞麟, 乔垭心, 等. 基于可解释性分析的深度神经网络优化方法 [J]. 计算机研究与发展, 2024, 61(01): 209–220.

附　录

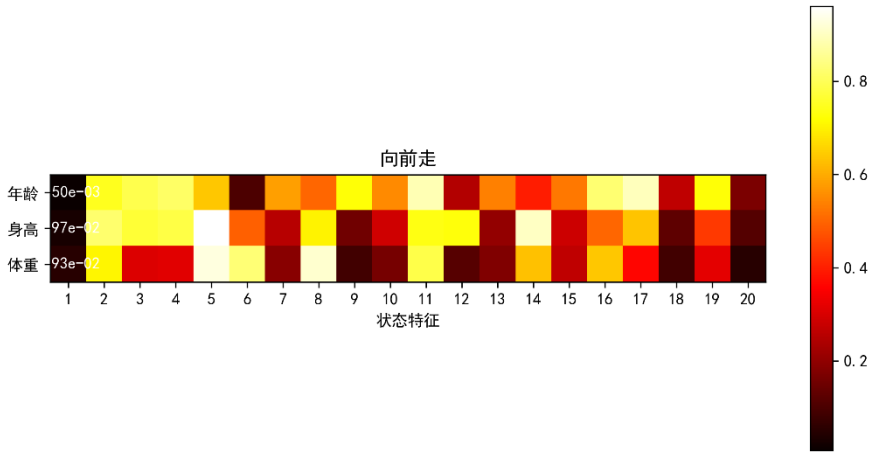
附录 A：问题二（1）中用分类模型得到的结果

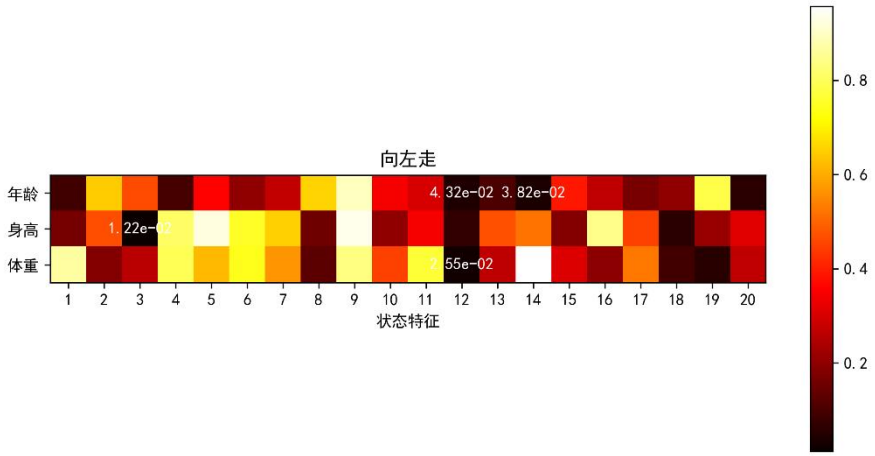
	Person1	Person2	Person3	Person4	Person5	Person6	Person7	Person8	Person9	Person10
动作1	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,
动作2	a2t2,a2t3,a2t4,a2t5,a3t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,
动作3	a2t1,a3t1,a3t2,a3t3,a3t4,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,
动作4	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,
动作5	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,
动作6	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t2,a6t3,a6t4,a6t5,a7t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,
动作7	a7t1,a7t2,a7t3,a7t4,a7t5,	a6t1,a7t1,a7t2,a7t3,a7t4,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,
动作8	a8t1,a8t3,a8t4,a8t5,a12t3,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t4,a8t5,a11t3,	a8t1,a8t2,a8t3,a8t4,a12t4,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a12t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,
动作9	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t3,a9t4,a9t5,a11t5,a12t2,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t4,a11t1,a11t2,	a8t5,a9t1,a9t2,a9t3,a9t4,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,
动作10	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t3,a10t4,a10t5,a12t5,	a10t3,a10t4,a10t5,a12t4,a12t5,	a10t2,a10t3,a10t4,a10t5,a12t5,	a10t1,a10t3,a10t4,a10t5,a12t5,	a10t1,a10t2,a10t3,a10t5,a12t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t4,a10t5,a12t5,
动作11	a11t1,a11t2,a11t3,a11t4,a11t5,	a9t1,a9t2,a11t2,a11t3,a11t4,	a11t1,a11t2,a11t3,a11t4,a11t5,	a8t3,a9t3,a9t5,a11t4,a11t5,	a9t5,a11t1,a11t2,a11t3,a11t4,	a11t1,a11t2,a11t3,a11t4,a11t5,	a11t1,a11t2,a11t3,a11t4,a11t5,	a11t1,a11t2,a11t3,a11t4,a11t5,	a11t1,a11t4,a11t5,a12t1,a12t2,	a11t1,a11t2,a11t3,a11t4,a11t5,
动作12	a8t2,a12t1,a12t2,a12t4,a12t5,	a10t2,a11t1,a12t1,a12t3,a12t4,	a10t1,a10t2,a12t1,a12t2,a12t3,	a10t1,a12t1,a12t2,a12t3,a12t4,	a10t2,a11t5,a12t1,a12t2,a12t3,	a10t4,a12t1,a12t2,a12t3,a12t4,	a12t1,a12t2,a12t3,a12t4,a12t5,	a8t5,a12t1,a12t2,a12t3,a12t4,	a11t2,a11t3,a12t3,a12t4,a12t5,	a10t3,a12t1,a12t2,a12t3,a12t4,

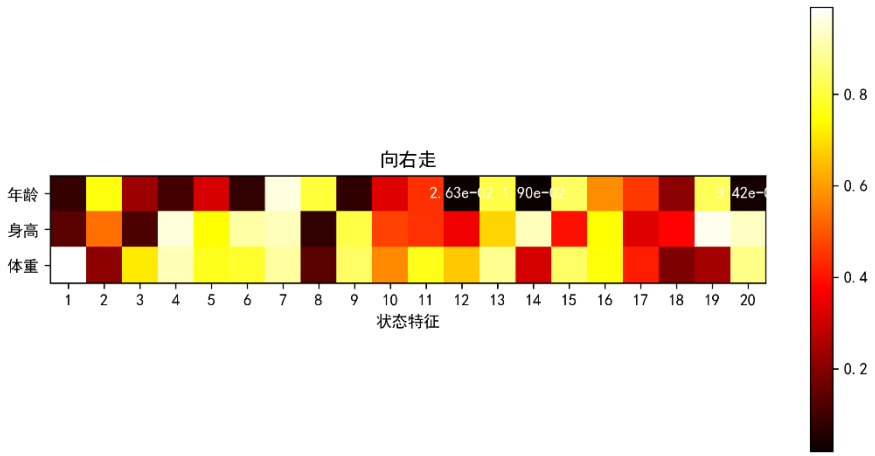
附录 B：问题二（1）中用判别模型得到的结果

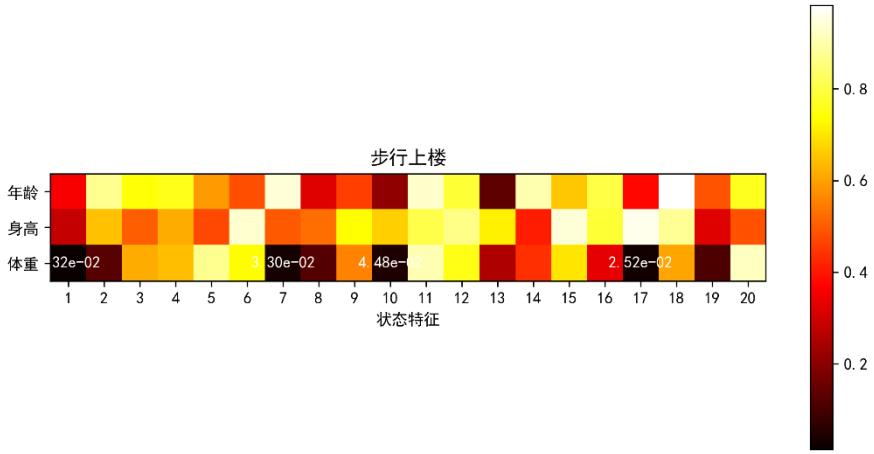
	Person1	Person2	Person3	Person4	Person5	Person6	Person7	Person8	Person9	Person10
动作1	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,	a1t1,a1t2,a1t3,a1t4,a1t5,
动作2	a2t1,a2t2,a2t3,a2t4,a2t5,a5t4,a5t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,	a2t1,a2t2,a2t3,a2t4,a2t5,
动作3	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,	a3t1,a3t2,a3t3,a3t4,a3t5,
动作4	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,a5t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,	a4t1,a4t2,a4t3,a4t4,a4t5,
动作5	a5t1,a5t2,a5t3,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,	a5t1,a5t2,a5t3,a5t4,a5t5,
动作6	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,	a6t1,a6t2,a6t3,a6t4,a6t5,
动作7	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,	a7t1,a7t2,a7t3,a7t4,a7t5,
动作8	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,a9t4,	a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,	a8t1,a8t2,a8t3,a8t4,a8t5,
动作9	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,a11t3,	a9t1,a9t3,a9t5,	a8t1,a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,	a9t1,a9t2,a9t3,a9t4,a9t5,
动作10	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,	a10t1,a10t2,a10t3,a10t4,a10t5,
动作11	a11t1,a11t2,a11t3,a11t4,a11t5,a12t1,	a11t1,a11t2,a11t3,a11t4,a11t5,a12t2,	a11t1,a11t2,a11t5,	a9t2,a11t1,a11t2,a11t3,a11t4,a11t5,a12t1,	a11t1,a11t2,a11t3,a11t4,	a11t1,a11t2,a11t3,a11t4,a11t5,	a11t1,a11t2,a11t3,a11t4,a11t5,a12t4,	a11t1,a11t2,a11t3,a11t4,a11t5,a12t5,	a11t1,a11t2,a11t3,a11t4,a11t5,	a11t1,a11t2,a11t3,a11t4,a11t5,
动作12	a12t2,a12t3,a12t4,a12t5,	a11t3,a12t1,a12t3,a12t4,a12t5,	a11t4,a12t1,a12t2,a12t3,a12t4,a12t5,	a12t2,a12t3,a12t4,a12t5,	a11t5,a12t1,a12t2,a12t3,a12t4,a12t5,	a12t1,a12t2,a12t3,a12t4,a12t5,	a12t1,a12t2,a12t3,a12t5,	a12t1,a12t2,a12t3,a12t4,	a12t1,a12t2,a12t3,a12t4,a12t5,	a12t1,a12t2,a12t3,a12t4,a12t5,

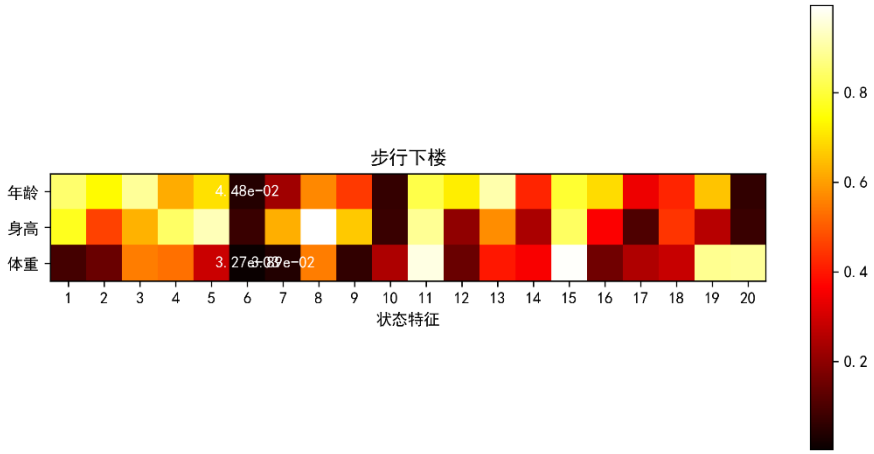
附录 C：问题三（2）中多元回归分析结果

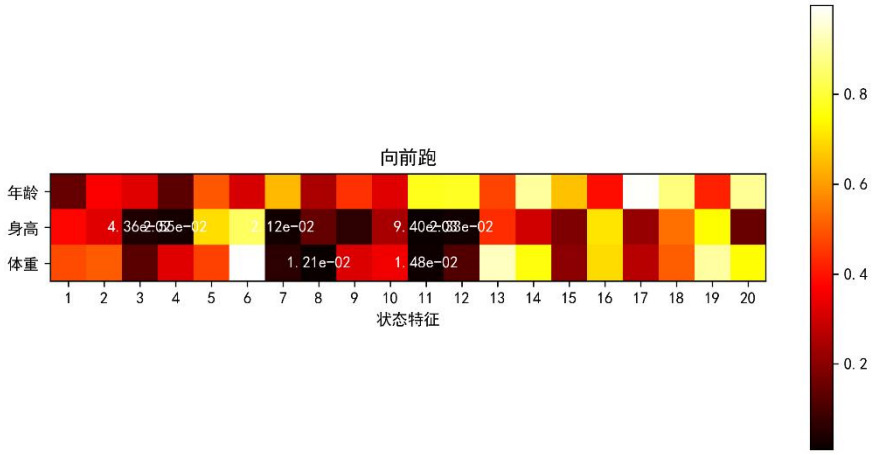


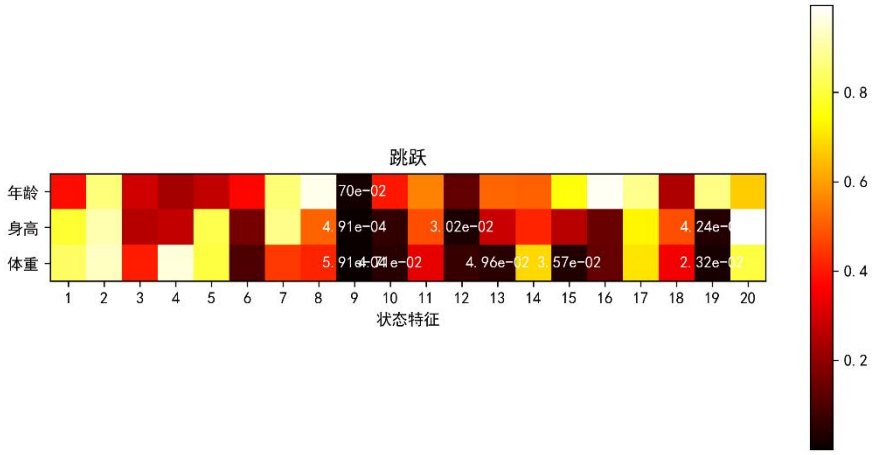


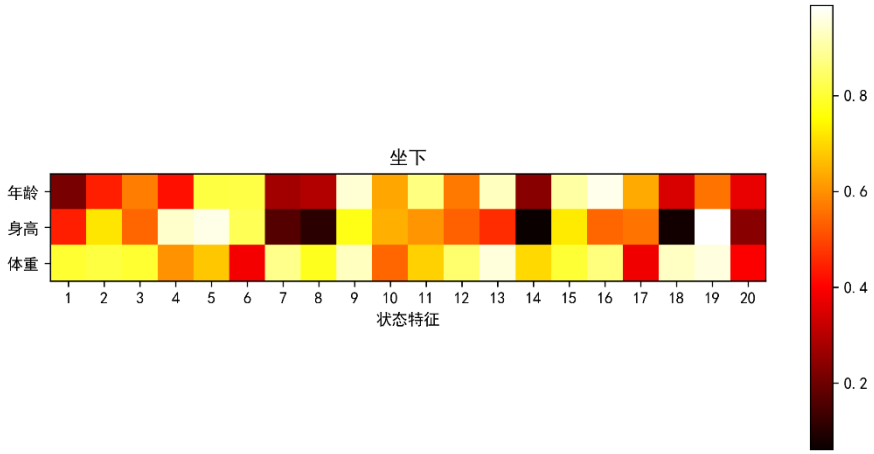


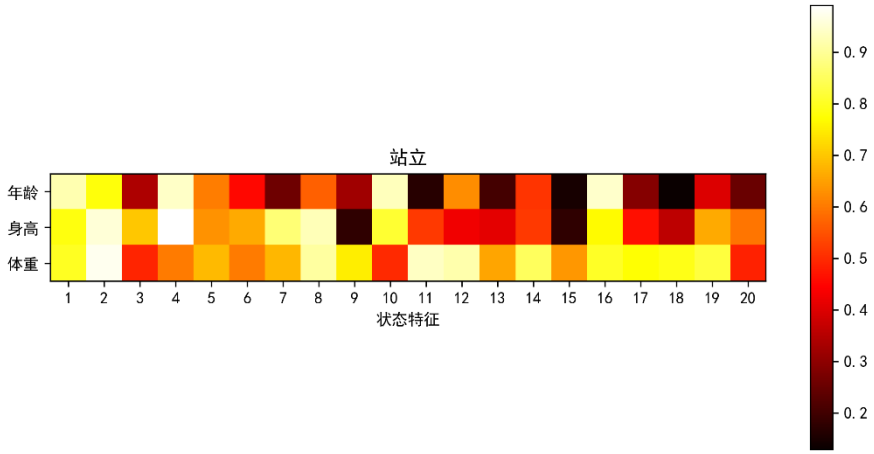

步行上楼
年龄
身高
体重
32e-02
3.30e-02
4.48e-02
2.52e-02
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2

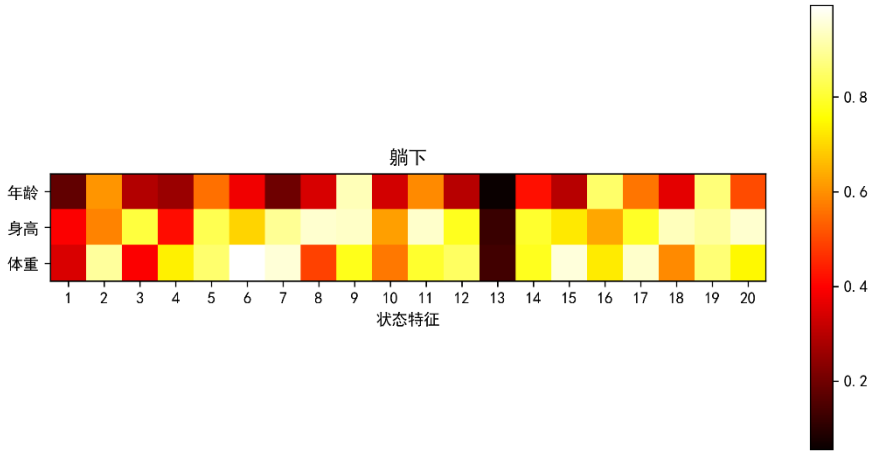

步行下楼
年龄
身高
体重
4.48e-02
3.27e-02
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2

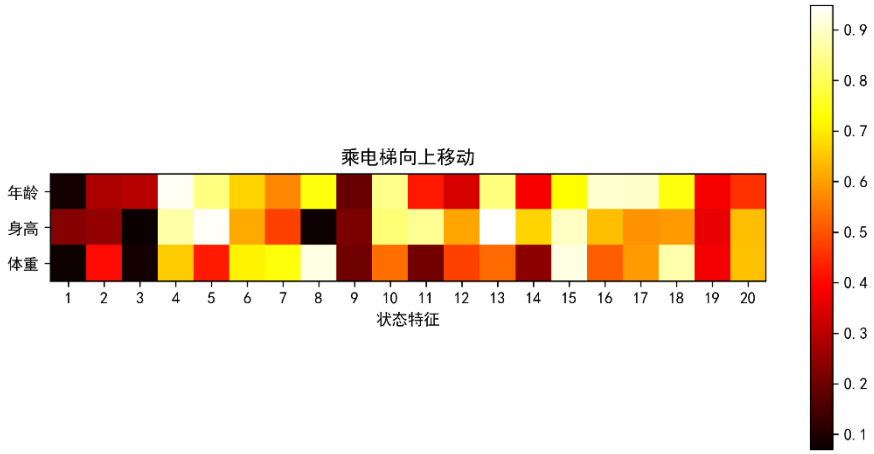

向前跑
年龄
身高
体重
2.12e-02
1.21e-02
1.48e-02
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2

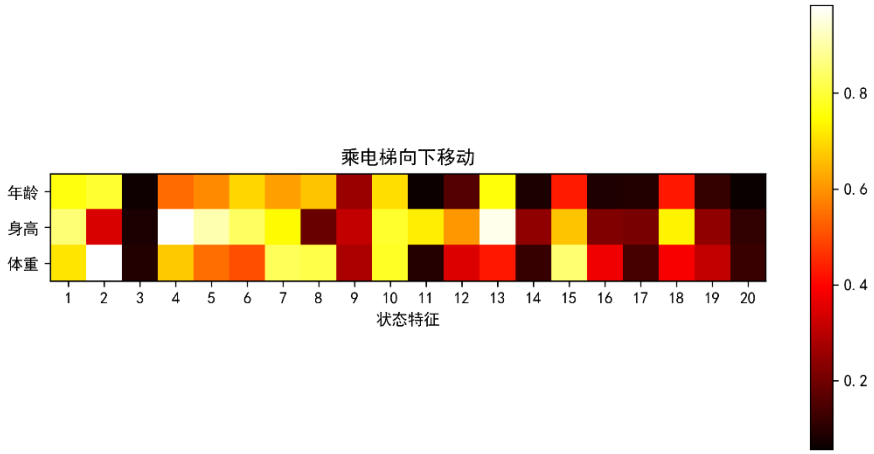

跳跃
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2


坐下
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2


站立
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2


躺下
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2


乘电梯向上移动
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


乘电梯向下移动
年龄
身高
体重
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
状态特征
0.8
0.6
0.4
0.2

附录 D：支撑材料列表

表 11 支撑材料列表

序号	文件名	材料说明
1	Q1_cluster.py	第一问：聚类模型
2	Q2_cluster-2.py	第二问：用聚类模型对附件 2 中数据进行分类
3	Q2_model.py	第二问：集成学习模型
4	Q2_test.py	第二问：用集成学习模型对附件 2 中数据进行分类
5	Q2_train.py	第二问：训练集成学习模型
6	Q3-1_pearson.py	第三问：方差分析及多元回归分析
7	Q3-3_train.py	第三问：训练深度卷积神经网络模型
8	Q3-test.py	第三问：用深度卷积神经网络模型对附件 5 中数据进行人员判别
9	read_excel.py	原始数据表格读取

附录 E：主要程序/关键代码

代码环境

操作系统: Win10
编程语言: Python 3.8
编辑器: PyCharm 2022.3.2 (Professional Edition)
代码详见：支撑材料

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.model_selection import train_test_split
import joblib
from tensorflow.keras.utils import to_categorical
import matplotlib
matplotlib.use('Agg')
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

# 第一问主程序，包含数据预处理和约束K-Means聚类
def cluster(deal_data, constrained_kmeans):
    # 输入分别为原始数据和约束K-Means聚类函数
    # 读取变量data_all
    data_all = np.load('data/data_file1.npy', allow_pickle=True)
    labels_list = []
    for data_index in range(3):
```

```
        # 取前60个数据
        data_list = data_all[data_index*60:(data_index+1)*60]
        # 数据预处理，包含数据清洗，特征提取，数据标准化，特征降维
        features = deal_data(data_list)
        # 使用约束K-Means 算法进行聚类，使用K-Means++初始化
        labels, centroids, index_list = constrained_kmeans(features, 12,
[5, 5])
        labels_list.append(labels)
    return labels_list

# 第二问主程序，训练集成学习模型
def Q2_train_model(deal_data, generate_true_labels, NN, DT, RF, NB,
KNN):
    # 输入分别为特征数据，标签数据，五个模型的训练函数
    # 读取数据
    data_list = np.load('data/data_file2.npy', allow_pickle=True)
    features= deal_data(data_list)
    # 生成标签数据，利用one-hot编码
    labels = generate_true_labels(data_list)
    # 划分训练集和测试集
    X_train, X_test, y_train, y_test = train_test_split(features, la-
bels, test_size=0.01, random_state=2,
                                                        shuffle=True)
    # 训练集成模型中的子模型
    # 神经网络
    accuracy_1 = NN(features, X_train, X_test, y_train, y_test)  # 神经网
络模型必须要特征提取，其他的不需要
    # 决策树
    accuracy_2 = DT(features, X_train, X_test, y_train, y_test)
    # 随机森林
    accuracy_3 = RF(features, X_train, X_test, y_train, y_test)
    # 朴素贝叶斯
    accuracy_4 = NB(features, X_train, X_test, y_train, y_test)
    # KNN
    accuracy_5 = KNN(features, X_train, X_test, y_train, y_test)
    return accuracy_1, accuracy_2, accuracy_3, accuracy_4, accuracy_5


# 集成学习模型进行分类
def Q2_test_model(w1, w2, w3, w4, w5, features):
    # 输入分别为五个模型的权重和特征数据
    # 读取模型
    model_NN = tf.keras.models.load_model('models/models_train1.0/model-
NN.h5')
    model_DT = joblib.load('models/models_train1.0/model-DT.pkl')
    model_RF = joblib.load('models/models_train1.0/model-RF.pkl')
    model_NB = joblib.load('models/models_train1.0/model-NB.pkl')
    model_KNN = joblib.load('models/models_train1.0/model-KNN.pkl')
    test_data_ori = features
    for times in range(3):  # 将数据划分为三份
        test_data = test_data_ori[times * 60:(times + 1) * 60]
        result_1 = model_NN.predict(test_data)
        result_2 = model_DT.predict(test_data)
        result_3 = model_RF.predict(test_data)
        result_4 = model_NB.predict(test_data)
        result_4 = to_categorical(result_4, num_classes=12)
        result_5 = model_KNN.predict(test_data)
        # 将四个矩阵进行加权平均
```

```
        result = w1 * result_1 + w2 * result_2 + w3 * result_3 + w4 *
result_4 + w5 * result_5
        result_argmax = np.argmax(result, axis=1)
    return result_argmax

# 第三问第一小问主程序，方差分析
def Q3_ANOVA(data, labels):
    from scipy.stats import f_oneway
    f_statistic_list = []
    p_value_list = []
    for i in range(12):
        print(f"Cluster {i}: {np.where(labels[:, i] == 1)[0] + 1}")
        data_person_action = data[i]
        data_person_action = data_person_action.reshape(
            (data_person_action.shape[0] // 5, 5 * data_person_ac-
tion.shape[1]))
        # 对所有行的数据进行方差分析，判定是否有显著差异
        # 准备数据，将每一行视为一个独立的组
        groups = [data_person_action[i, :] for i in range(data_per-
son_action.shape[0])]
        # 执行单因素方差分析
        f_statistic, p_value = f_oneway(*groups)
        f_statistic_list.append(f_statistic)
        p_value_list.append(p_value)
    return f_statistic_list, p_value_list

# 第三问第二小问主程序，多元回归分析
def Q3_OLS(data_person_action, age, height, weight):
    # 输入为每个人的活动特征数据，年龄、身高、体重
    import statsmodels.api as sm
    p_value_list_all = []
    for i in range(12):
        # 构建包含数据的DataFrame
        df1 = pd.DataFrame(data_person_action, columns=[f'activity_sta-
tus_{j}' for j in range(1, 21)])
        # 自变量
        df2 = pd.DataFrame({'age': age, 'height': height, 'weight':
weight})
        # 合并两个DataFrame
        df = pd.concat([df1, df2], axis=1)
        X = df[['age', 'height', 'weight']]
        X = sm.add_constant(X)  # 添加常量项
        p_value_list = []
        for J in range(1, 21):  # 假定有从 1 到 20 的活动状态
            y = df[f'activity_status_{J}']
            model = sm.OLS(y, X).fit()
            temp = model.pvalues
            age_p_value = temp['age']
            height_p_value = temp['height']
            weight_p_value = temp['weight']
            p_value_list.append(np.array([age_p_value, height_p_value,
weight_p_value]))
        p_value_list_all.append(p_value_list)
    return p_value_list_all

# 第三问第三小问主程序，深度卷积神经网络训练
def Q3_train(model, features):
    from tensorflow.keras.layers import Conv1D, Flatten, Dense
    # 输入为模型和特征数据
```

```
    # 设置随机种子
    np.random.seed(1234)
    samp_num = 10000
    sample_features = np.zeros((10, samp_num, 12, 20))
    for person in range(10):
        sample_feature_3 = np.zeros((samp_num, 12, 20))
        for times in range(samp_num):
            sample_feature = np.zeros((12, 20))
            for i in range(12):
                # 随机选择j 为0-4
                j = np.random.randint(0, 5)
                sample_feature[i, :] = features[person, i, j, :]
            sample_feature_3[times, :, :] = sample_feature
        sample_features[person, :, :, :] = sample_feature_3
    sample_features = sample_features.reshape(10 * samp_num, 12, 20)
    # 建立标签数据
    sample_labels = np.zeros((10, samp_num))
    # 按照person进行标签
    for i in range(10):
        sample_labels[i, :] = i
    sample_labels = sample_labels.reshape(10 * samp_num)
    # 划分训练集和测试集
    X_train, X_test, y_train, y_test = train_test_split(sample_features,
sample_labels, test_size=0.3, random_state=42,
                                                        shuffle=True)
    model.compile(optimizer='adam', loss='sparse_categorical_crossentro-
py', metrics=['accuracy'])
    # 训练模型
    model.fit(X_train, y_train, epochs=20, batch_size=512, valida-
tion_data=(X_test, y_test))
    model.evaluate(X_test, y_test)
    # 保存模型
    model.save('results/Q3-3/model-Q3.h5')

# 第三问第三小问主程序，深度卷积神经网络求解分类问题
def Q3_test(features):
    # 读取模型
    model = tf.keras.models.load_model('results/Q3-3/model-Q3.h5')
    test_data = features.reshape((5, 12, 20))
    result = model.predict(test_data)
    result = np.argmax(result, axis=1)
    return result

# main函数
if __name__ == "__main__":
    # 第一问主程序运行
    cluster_labels = cluster(deal_data, constrained_kmeans)
    # 第二问模型训练
    accuracy_1, accuracy_2, accuracy_3, accuracy_4, accuracy_5 =
Q2_train_model(deal_data, generate_true_labels, NN, DT, RF, NB, KNN)
    # 第二问模型测试
    result = Q2_test_model(w1, w2, w3, w4, w5, features)
    # 第三问第一小问主程序运行
    f_statistic_list, p_value_list = Q3_ANOVA(data, labels)
    # 第三问第二小问主程序运行
    p_value_list_all = Q3_OLS(data_person_action, age, height, weight)
    # 第三问第三小问主程序运行
    Q3_train(model, features)
    result = Q3_test(features)
```