队伍编号	MC2306094
题号	D

基于机器学习的航空飞行安全预测与控制

摘 要

飞行安全是航空业和相关领域面临的重要问题。由于航空事故带来的严重后果和损失，解决飞行安全问题具有重要的理论和实践意义。本研究旨在建立一套完整的数学模型，并通过**实验验证**、评估该模型的**预测能力**和实用性，以提供对航空业和相关领域**安全**管理和决策的参考和指导。

对于**问题一**，我们首先进行数据的**可靠性分析**，并简化和修改数据，使用合适的方法确保数据的完整性和准确性。我们利用箱线图去除异常值，以消除其对数据分析结果的影响，并进行特征工程以**提取关键特征**项。然后，我们建立了**随机森林模型**，以输出**特征值的重要性**。最后，我们使用统计分析、主成分分析和**聚类分析**等方法进一步提高了数据的**准确率**。

针对**问题二**，我们首先对数据缺失和异常值进行**同步分析和处理**，以提高结果的准确性。然后通过绘制热力图来判断不同杆位之间的相关性，并根据附件 1 中的杆位和盘位变化曲线，采用**三倍标准差检验法**作为异常阈值，利用可视化分析对其进行平滑处理，以得出飞行过程中这两个指标变化剧烈的时间段，异常值代表其剧烈程度。我们运用**时间序列分析**、**信号处理**等技术手段对曲线中的波动、趋势和周期性进行分析，并建立交叉验证计算平均准确率函数 **cv_accuracy**，最后使用 **SVM 分类器**进行训练和预测，对模型进行**交叉验证**。

针对**问题三**，我们分析飞机**超限情况**。为了使数据具有可分析性和可比性，我们对数据进行了清洗、去重和标准化等预处理操作，并采用了 **Z-score 标准化处理**方法。我们利用数据可视化技术进行直观展示，并计算每架飞机在所有警告事件中的占比。此后，我们进一步研究了每架飞机在不同航线上发生超限事件的占比，并采用**整数回归**的方法，针对某特定超限事件“50 英尺至接地距离远”建立了**预测模型**，并对数据进行训练-测试集划分，最终对模型进行验证和调优。

针对**问题四**，我们探究飞行参数的飞行技术评估。为此，我们对附件 3 中的数据进行了预处理，并采用**特征工程**和统计学方法提取表征飞行参数的**关键特征**。我们使用基于**信息熵**的决策树算法建立飞行技术评估模型，并利用**交叉验证**方法进行评估和调优，以验证数据的真实性和**决策树模型**的性能。

针对**问题五**，我们分析不同超限事件在飞机不同飞行阶段的发生情况。由于缺乏不同飞行状态下具体指标说明，我们对其进行了编码。根据飞行状态和下降率指标判断是否会出现超限，并选取下降过程中飞行速度作为指标。为实现航空公司**自动化预警**机制，我们利用机器学习算法建立预警模型，并运用**基于逻辑回归的二分类模型**预测飞行过程中是否存在潜在的安全风险。最后，我们采用**实时数据传输**技术，将预警模型与数据传输相结合进行预警和报警，以模拟不同飞行场景，并对预警模型和预防机制进行评估和验证，以有效**避免飞行安全事故**的发生。

关键词： 飞行安全　机器学习　随机森林　SVM　决策树　逻辑回归　python

目录

1 引言

1.1 问题背景

近年来，随着全球经济的发展和人民生活水平的提升，越来越多的民众选择乘坐飞机出行。而飞行安全问题也因此变得愈加突出。航空运输作为一项高风险行业，其安全问题尤其引人关注。中国民用航空局公布的数据显示，2019 年中国民航共完成旅客运输量 6.59 亿人次，日均旅客量超过 180 万人次。然而，频繁的航班事件和骇人的空难事故也不断提醒我们：飞行安全问题依然是航空业面临的重大挑战。

一次重大飞行事故不仅会给航空公司带来巨大的经济损失，更会对乘客造成生命威胁，给亲人和家属带来巨大的痛苦和伤害。例如，2022 年 3 月 21 日，“3.21”空难的发生终结了中国民航安全飞行史上 1 亿零 59 万小时飞行的历史最好安全记录。这一事件给人们敲响了警钟，使得在飞行安全方面加强研究和管控，变得愈加迫在眉睫。

为了解决飞行安全问题，需要实现对从业人员素质的有效提升，实现对风险的监测和预警等举措。这其中，科学管理和数据分析尤其重要。通过有针对性、系统性的管控手段，例如模拟仿真技术、机载监测系统等，可以更好地降低飞行安全问题的风险，为人们的出行提供更为稳定可靠的保障。因此，对航空业的飞行安全问题进行深入的研究和探索，加强管理和监控，已经变得愈加必要。

1.2 问题重述

- **问题一：**该问题要求对附件 1 中的 QAR 数据进行可靠性研究，以提取与飞行安全相关的关键数据项，并确定其重要程度。为此，需要展开可靠性研究，包括数据量化分析、异常检测、去重处理、缺失值填充、数据标准化等步骤，以确保分析结果可靠。同时，对于疑难问题，应考虑使用传统统计分析、数据挖掘或机器学习等方法，进一步深入挖掘数据潜在信息。
- **问题二：**该问题要求对附件 1 中的飞行操纵杆过程进行量化描述，以确定飞行偏差的原因。为此，需要首先进行数据清洗和预处理，以便精确地捕捉飞行操纵杆的变化。然后，可采用时间序列分析、振荡频率检测、主成分分析等方法，对操纵杆变化进行分析和处理。通过对这些变化的量化描述，可以找出飞行偏差的原因。
- **问题三：**该问题要求对不同的超限情况进行分析，并研究其基本特征和产生原因。为了深入了解超限情况的不同特征，可采用数据可视化、分类分析、关联规则挖掘等方法，对附件 2 中的数据进行分析和处理。此外，还应对影响超限情况的相关因素进行研究，例如特定机场、天气变化、飞行员技术水平等，以找出超限情况出现的根本原因。
- **问题四：**该问题要求建立一种基于飞行参数的飞行技术评估方法，评估飞行员的技能水平。要解决这个问题，需要深入研究飞行动态参数的变化及其对飞行员技能水平的反映。为了实现此目的，可以通过数据可视化、聚类分析、回归分析等方法，分析数据来源，并对飞行员技术水平的不同层次进行评估。

- **问题五：**该问题要求建立一种航空公司的实时自动化预警机制，以减少安全事故的发生。为实现这一目标，需要根据附件 1 中的数据创建相应的模型，实现飞行安全预警，并对可能产生的风险因素进行监测。同时，预警机制还应该实现数据处理和快速响应机制，以便尽可能在事故发生前采取适当的措施进行干预。最后，还应该在采取预防性措施后评估预防效果，反馈至安全管理系统中。

1.3 思维框架

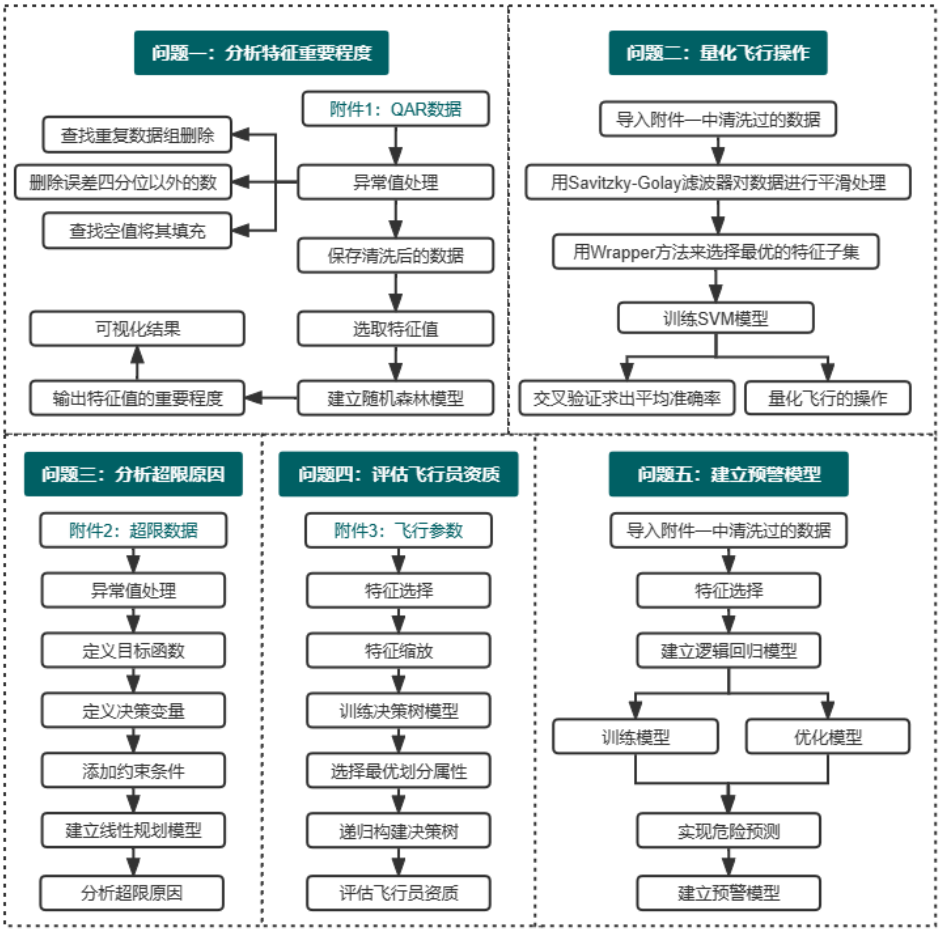


图 1 对问题的思考图

2 问题分析

2.1 对于问题一的分析

为了对附件 1 中的数据进行可靠性分析，我们着重考虑了以下几个问题。首先，我们需要系统地分析数据的缺失程度和异常值情况，以便采取合适的措施来确保数据的完整性和准确性。

对于缺失数据，我们需要选择合适的方法进行数据填补，例如，可以采用基于插值法和回归分析的方法来填补缺失值。而关于异常值，则需进行筛查和处理，以消除其对数据分析结果的影响，常用的方法包括离群值检测和异常值修正等技术手段。

其次，提取关键数据项进行分析。在关键数据项的选取过程中，我们应当考虑数据的代表性和可解释性。同时，可以采用统计分析、主成分分析和聚类分析等方法来提取有用的信息，以实现对数据的更加准确的分析处理。例如，在进行 G 值数据项的分析时，可以使用随机森林模型计算不同特征的重要程度，并以此来评估各特征对数据的影响程度。

2.2 对于问题二的分析

为了对飞机操纵进行量化描述，并分析附件 1 中的杆位变化曲线，我们考虑了以下问题。首先，需要选取能够表征飞机操纵的关键数据项，如操纵杆的位置、角度和移动速度等，以便对飞机操纵进行量化描述。同时，需要对数据的缺失和异常值情况进行分析和处理，确保数据的完整性和准确性。

其次，针对附件 1 中的杆位变化曲线，需要对其中的特征进行提取和分析。重点关注的指标是 CPA WHL 1 POSN 和 CAP CLM 1 POSN,有杆量和盘量两个指标，分别影响飞机姿态和坡度，我们可以筛选出飞行过程中这两个指标变化剧烈的时间段并进行可视化分析，绘制这两个指标的变化折线图，观察不同时间段的杆位的变化即可以对飞行操纵过程进行量化。

然后可以采用时间序列分析、信号处理等技术手段对曲线中的波动、趋势和周期性进行分析。同时，可以通过相关性分析等方法，探究松杆操纵与重着陆之间的关联性，以得出该松杆操纵对重着陆的影响程度。

2.3 对于问题三的分析

为了对飞机超限情况进行分析，我们考虑了以下问题。首先，需要对附件 2 中的数据进行数据预处理，包括数据清洗、编码、分类等。通过标签编码等技术对字符串类型的数据项进行编码，以便进行后续的分析处理。

其次，针对不同的超限情况，需要按照机号、目的机场、起飞机场、时间日期、飞行阶段等因素进行分析。可以采用数据可视化技术，如饼图、柱状图、散点图等。

最后，研究超限情况的时空分布和基本特征，包括各种超限之间的频次和比例关系、不同地区和机场超限的分布规律、不同警告级别超限的特点等。

2.4 对于问题四的分析

为了对飞行员的资质进行分析，我们考虑了以下问题。首先，需要对附件 3 中的数据进行数据预处理和特征工程，包括数据清洗、编码、降维等。可以使用标签编码、独热编码等方法将字符串类型的变量转换为数值型变量，同时，可以通过特征选择和降维技术，提取能够表征飞行参数的关键特征，如起降高度、平均速度、飞行距离等，以便进行下一步的模型构建和训练。

其次，需要根据附件 3 中的数据，选择适当的回归模型进行预测，如线性回归、多项式回归、随机森林等。可以使用交叉验证等方法对模型进行评估和调优，以提高预测精度和可靠性。

最后，需要对评估结果进行解释和应用，将评估结果提供给相关部门和飞行员，以便为飞行安全管理和技术培训提供参考和支持。

2.5 对于问题五的分析

为了建立一个预防安全事故的自动化预警模型，我们考虑了以下问题。首先，需要利用 QAR 数据，建立基于超限事件的风险预测和预警模型。可以通过分析附件 2 中的超限名称、超限时间等信息，建立超限事件的分类模型，并结合附件 1 中的数据，实时监测和分析飞行过程中可能存在的超限事件，提前发现和预测风险，以保障飞行运行的安全。

其次，需要将预警模型与实时数据传输技术相结合，实现实时监测和分析飞行数据，并根据需要进行预警和报警。可以借助现代物联网和云计算技术，实现飞行数据的实时传输和处理，同时，可以采用大数据分析和机器学习等技术，对数据进行监测和分析，提取和识别潜在的安全风险，及时预警和处理。

最后，需要对预警模型和预防机制进行评估和验证，以保障其有效性和可靠性。可以利用实时飞行数据，对不同的飞行场景进行模拟，并评估预警模型的预测精度和响应时间。同时，应当结合实际情况，不断改进和优化预警机制，提高其性能和稳定性，以保障飞行运行的安全和可靠性。

3 模型假设及其合理性

- **假设 1**：QAR 数据可以准确地反映飞行过程中的关键参数和指标，包括飞行高度、速度、温度、湿度等。同时，QAR 数据可以实现陆空实时传输，满足实时监测和分析的需要。
- **合理性**：QAR 数据是飞机的全面监控系统，能够收集飞机的全部信息，包括机载设备状态、操作员舱内外制动、操作杆位等关键参数和指标，以及飞机在航空管制区域的位置、高度、速度、温度、湿度等。这些数据是监测飞行安全的重要依据，可用于进行维修、性能管理和飞行安全的分析。现代 QAR 系统可以实现陆空实时传输，机载天线将监控数据发送给地面站，提供飞机的实时数据传输。通过云计算和物联网技术，实现全球实时监控和事故分析，满足实时监测和分析的需要。
- **假设 2**：飞行员都是在健康、正常的状态下驾驶飞机，没有任何身体或精神问题，对飞行安全有高度的责任心和职业操守，且飞机起飞前都是经过严格的安全检测。
- **合理性**：假设 2 中包含的因素对航空安全具有重要作用。它们是飞行员驾驶飞机过程中表现出的各项参数的重要因素。不保证上述因素会导致一些飞行员的评估出现严重偏差，甚至导致更严重的安全问题。
- **假设 3**：飞行员的驾驶技能和操作水平在同类飞机上没有差异。不会因为飞机的架次不同而发生改变。
- **合理性**：飞行员的驾驶技能和操作水平在同类飞机上应该是基本相同的。无论飞机的架次是否不同，飞行员都需要在严格的培训和认证体系下获得相应的资质才能够驾驶飞机。因此，即使是不同架次的飞机，飞行员的技术和水平也应该是基本相同的。当然，这不排除特定情况下一些特殊技能或培训可能需要针对特定架次的飞机进行。但是这种情况我们不选择考虑

4 符号说明

符号	说明
$h_t(x)$	第t个弱分类器的预测
x_i	样本i的特征向量
μ_j	第j个簇的中心点
$d(x_i, x_j)$	样本x_i和样本x_j之间的距离
$d(\mu_j, x_i)$	样本x_i和第j个簇的中心点之间的距离
ϵ	聚类算法的收敛阈值
k	簇的个数
$sigmoid(x)$	$sigmoid$函数，在二分类问题中用于做预测
$L(y, f(x))$	损失函数，用于评估实际值y和预测值$f(x)$之间的差距
y_j	样本j的真实值
w_i	样本i的权重
b	随机森林模型中的树的数量
m	随机森林模型中的采样的特征数
X	输入特征空间
x_i	样本i的特征向量
y_i	样本i的标签
$f_t(x)$	第t棵决策树的输出
$\hat{y}$	随机森林的预测结果
$\mathbb{E}$	随机变量的期望值
Λ	正则化技术

$\mathbb{E}$:表示随机变量的期望值，是对于某个函数在所有可能取值下的加权平均值，其中权值为该函数取值的概率。在机器学习中，$\mathbb{E}$ 常用于描述模型误差的期望值。

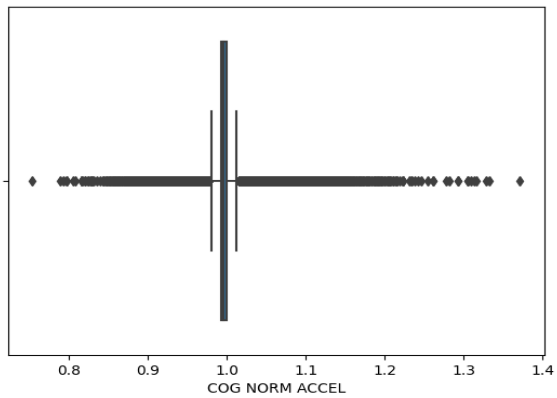
Λ :是一种正则化技术，在模型中引入一个额外的惩罚项，以避免模型过拟合。常用的 Λ 包括 L_1 正则化和 L_2 正则化，用于限制模型参数的大小和数量。

5 模型建立与求解

5.1 问题一：利用随机森林分析特征重要程度

5.1.1 数据预处理

由于数据中存在错误数据，对飞机安全研究造成一定的影响。因此，需要对数据进行预处理，以去除伪数据并减少错误数据对后续分析的影响。常用的数据预处理方法包括去重、缺失值填充，异常值检测和处理等。在具体实现中，我们可以使用 Python 库，如 Pandas 和 NumPy 来实现。为了确保数据质量，我们可以建立数学模型对附件中的数据进行质量检测。通过计算数据缺失率，检测数据异常值，以及计算每个列中空值的占比，我们可以确定数据质量问题。下面是用箱线图进行的异常值处理：



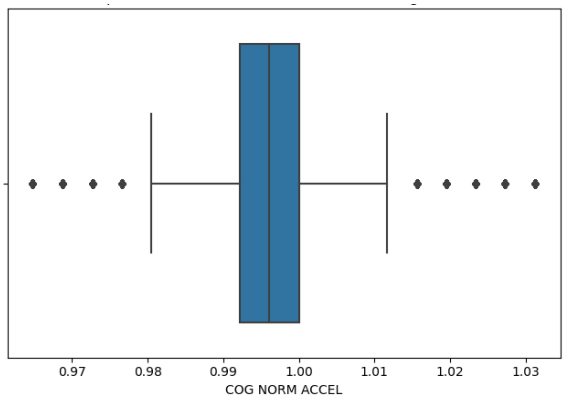


图 2 异常值处理箱线图（处理前） **图 3 异常值处理箱线图（处理后）**

另外，通过使用聚类算法，如 DBSCAN 或其他聚类方法，我们可以筛选出异常值。利用 PCA 进行降维可以计算异常值的比例，并将其删除。在采用聚类算法时，需要根据具体需求和数据特性选择适当的聚类算法。常用的聚类算法包括：

K-Means 算法，核心公式为：

$$C_j = \frac{1}{m_j}\sum_{i=1}^{m_j} x_i \tag{1}$$

DBSCAN 算法，核心公式为：

$$\rho(p_n, P) = |\{p_i : dist(p_n, p_i) \le \epsilon\}| \tag{2}$$

层次聚类算法，核心公式为：

$$d_{ij} = \max\{|x_1, x_2|, |y_1, y_2|\} \tag{3}$$

聚类评估，评价聚类结果的好坏，常用的评价指标包括：

簇内平均距离：

$$\mathrm{W}(C_k) = \frac{1}{|C_k|}\sum_{i,j\in C_k, i\neq j} d_{ij} \tag{4}$$

簇间最小距离：

$$D(C_i, C_j) = \min_{x_i\in C_i, x_j\in C_j} d_{ij} \tag{5}$$

其中，C_k 是第 k 个簇，d_{ij} 表示数据点 i 和 j 之间的距离。

簇内方差：

$$S_k^2 = \frac{1}{|C_k|}\sum_{i\in C_k}(x_i - \mu_k)^2 \tag{6}$$

轮廓系数：

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{7}$$

其中，$a(i)$ 表示数据点 i 到同一簇内其他点的距离的平均值，$b(i)$ 表示数据点 i 到最近簇内所有点的距离的平均值。

Calinski-Harabasz index：

$$CH = \frac{tr(B_k)}{tr(W_k)} \times \frac{m-k}{k-1} \tag{8}$$

其中，B_k 表示簇之间的散度矩阵，W_k 表示簇内的散度矩阵，m 表示数据点的数量，k 表示簇的数量。

Silhouette coefficient：

$$s(i) = \frac{b(i)-a(i)}{\max\{b(i),a(i)\}} \tag{9}$$

其中，$a(i)$ 表示数据点 i 到同一簇内其他点的距离的平均值，$b(i)$ 表示数据点 i 到最近簇内所有点的距离的平均值。

异常检测：使用聚类结果和元数据综合分析，通过异常点和设备特征间的关联性进行判断和检测。常用的方法包括：

聚类中心距离法，核心公式为：

$$d(x_n, c_n) = \sqrt{\sum_{i=1}^{k}(x_{ni} - c_{ni})^2} \tag{10}$$

样本点到最近聚类中心距离法，核心公式为：

$$d_{min}(x_n) = \min 1_{j\in\{1,2,\ldots,K\}, j\neq c_n} d(x_n, c_j) \tag{11}$$

离群因子算法（OF），核心公式为：

$$OF(x_n) = \frac{1}{K}\sum_{j=1}^{K}\left(\frac{d(x_n,c_j)}{MAD(d(x_n,c_j))}\right) \tag{12}$$

5.1.2 特征值选取

特征值选取是一种广泛应用于机器学习中的重要数据预处理方法。要选择适当的特征值选取方法，需要考虑以下几个方面：

- 特征与目标变量之间的相关性。
- 特征之间的相关性。
- 特征的数量和质量。
- 特征类型和分布。
- 机器学习模型的类型和要求。

对于飞行安全相关的关键数据，我们需要进行特征工程，即从原始数据中提取相关特征以支持后续的建模和分析。我们可以从 QAR 数据中提取以下特征：

- 飞行阶段（起飞、巡航、爬升、下降、着陆）
- 飞机参数（速度、高度、姿态、航向、偏航角、俯仰角等）
- 环境参数（气压、温度、湿度、风速等）
- 操作数据（油门、方向舵、升降舵、襟翼等）
- 机组操作（飞行员的控制操作和应答数据）
- 时间数据（日期、时间、飞行时长等）

我们构造每秒钟的最大着陆 G 值作为预测变量，并将其余特征值分别作为自变量，以建立数据模型。在进行特征值选取时，需要结合以上几个方面进行考虑，以最大程度地提高机器学习模型的性能。

5.1.3 特征值重要程度分析

经过特征筛选后，我们需要进行重要程度分析，并建立随机森林回归模型。随机森林回归模型作为一种集成学习方法，由多个决策树构成。在此模型中，每个决策树都会对数据集进行随机抽样，并对每个抽样集合进行独立的训练。同时，在每个节点上，随机森林选择一个随机的特征子集来进行分割，以减少模型对任何一个特定特征的依赖性。对于回归问题，随机森林回归模型将决策树的预测结果取平均值作为最终预测结果。相比单个决策树，随机森林模型可以更好地处理高维数据和非线性关系，并减少过拟合的风险。此外，由于每个决策树都是基于随机样本和特征构建的，因此随机森林模型还具有很好的抗噪能力。

在建立随机森林回归模型之前，需要完成以下步骤：

- **数据准备：**将数据集划分为训练集和测试集，并根据需要对数据进行预处理和特征工程。

- **随机抽样：**在随机森林模型中，每个决策树都是基于随机抽样的数据集进行训练的。这样可以减少模型对任何一个特定的样本的依赖性，并增加模型的稳定性。
- **构建决策树：**使用随机抽样的数据集和特征子集来构建决策树。决策树的构建可以采用递归分裂的方式进行，即不断将数据集分成更小的子集，直到满足某个停止条件为止。决策树的计算公式如下所示：

$$Decide(x_i) = \sum_{j=1}^{K} c_j I \tag{13}$$

- **预测结果：**对于回归问题，随机森林模型将每个决策树的预测结果取平均值作为最终预测结果。对于分类问题，随机森林模型将每个决策树的预测结果进行投票，以确定最终预测结果。

$$y = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \tag{14}$$

其中，y 表示最终的预测值，$f_m(x)$ 表示第 m 棵决策树的预测输出，M 表示决策树的数量。

- **模型评估：**使用测试集数据对随机森林模型进行评估，以确定模型的性能如何。可以使用各种指标，如均方误差（MSE）和准确率等。 模型的评估公式如下所示：

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{15}$$

其中，N 表示测试集的数量，y_i 表示测试数据的真实输出，$\hat{y}_i$ 表示模型的预测输出，$\bar{y}_i$ 表示测试集的均值。

- **超参数调整：**根据模型评估结果，调整超参数以优化模型性能。常见的超参数包括决策树数量、特征子集大小和分裂节点时的最小样本数等。
- **预测新数据：**使用优化后的随机森林模型对新数据进行预测。

在构建随机森林回归模型后，需要输出特征重要程度，并将每个特征值的重要程度进行降序排序。同时，可输出前 10 个重要特征，并绘制特征重要曲线图。

特征重要程度可以通过计算每个特征被选择为节点的次数，然后进行归一化处理来获得。假设总共有 T 棵树，第 k 棵树上，节点 i 选择的特征为 v_i^k。特征 j 的重要程度可以定义为：

$$importance(j) = \frac{1}{T}\sum_{k=1}^{T}\sum_{i=1}^{n_k} I\left(v_i^k = j\right) \triangle p_i^k \tag{16}$$

其中，n_k 表示第 k 棵树的节点数，p_i^k 表示节点 i 的样本比例。最终输出特征重要程度的列表和特征重要程度图示。

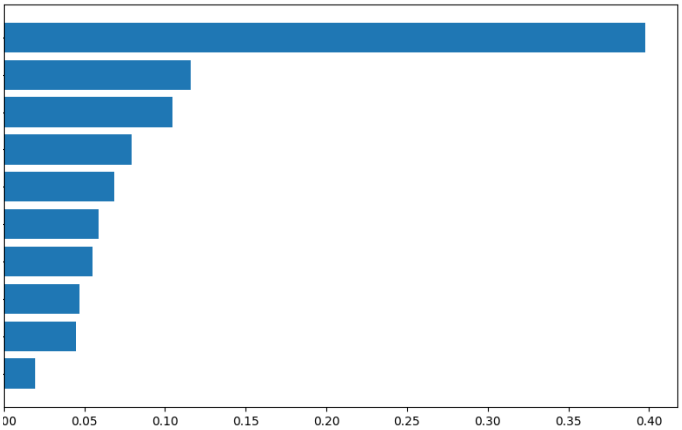


图 4 前十的特征值的重要性

$$importance(top10) = 99.11\% \tag{17}$$

5.1.4 可视化数据

可视化数据的作用是通过图形化的方式展示数据，帮助人们更直观地 理解和分析数据。可视化数据可以帮助人们更好地发现数据中的模式和趋势，比较不同数据集之间的关系和差异，清晰地看到数据分布和组成成分，识别数据中的异常值或离群点。通过使用图表或图形来将数据结果可视化，人们可以更好地向他人展示分析结果，从而提高与他人协作和沟通的效率。

R-squared 是一种用于评估回归模型拟合效果的统计指标，它表示模型所解释的响应变量（因变量）方差的比例，即模型拟合数据的程度 。其取值范围在 0 到 1 之间。当 R-squared 接近 1 时说明模型拟合效果较好，而当 R-squared 接近 0 时说明模型并不能很好地拟合数据，甚至可能为负。

具体来说，R2 可以通过以下公式计算：

$$R^2 = 1 - \frac{SSE}{SST} \tag{18}$$

其中，SSE（Sum of Squared Errors）表示模型预测值与真实值之间的残差平方和，SST（Total Sum of Squares）表示响应变量的 总平方和。在计算R^2时，SSE 表示模型无法解释的响应变量方差的比例，而SST则代表总响应变量方差的比例。

因此，R^2越接近 1，模型对数据的解释能力就越好，模型拟合程度也就越高。计算结果为：

$$R^2 = 0.987026 \tag{19}$$

当构造和训练模型时，可视化方法可以通过散点图来预测结果和真实结果，利用森林模型进行拟合，判断预测和真实结果的相关性程度。值得注意的是，相关性并不代表因果关系，因此需要结合实际问题来进行综合性分析。

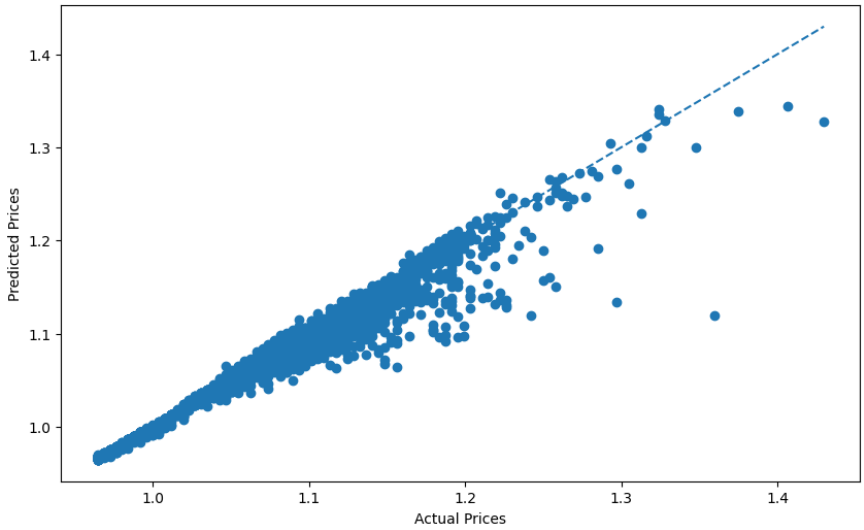


图 5 随机森林回归图

5.2 问题二：利用 SVM 对飞行操作进行量化

5.2.1 杆位、盘量分析

由于数据项是按时间排列的，可以构建时间特征项并对每架飞机的杆 位进行编号，以分析 4 个杆位的相关性系数，判断它们之间的相关性。为了更好地描述飞行过程中杆位的变化情况，可以使用散点图来展 示数据，进而合理化描述飞行过程中的操纵杆变化情况。

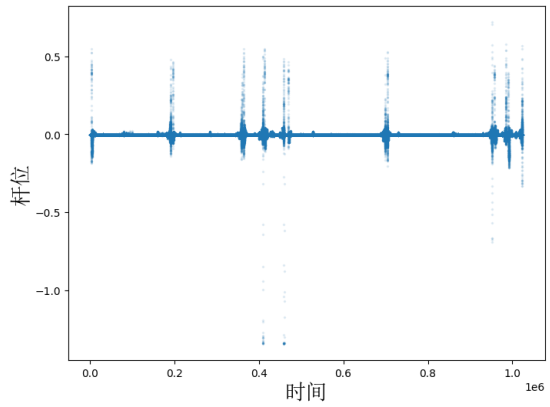


图 6 杆位随时间变化图

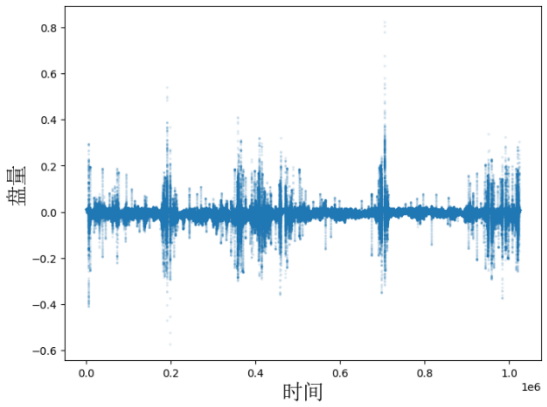


图 7 盘量随时间变化图

本文所采用的相关性系数方法称为热力图（heatmap）。对于绘制热力图，我们使用 python 的 pandas 库和 seaborn 库的 heatmap 函数进行数 据整理与转换，并利用 python 中的 matplotlib 库工具进行可视化调整。在通常情况下，颜色被用来表示相关系数的大小和方向，其中蓝色表示负相关，红色表示正相关，颜色深浅则表示相关性强度的大小。方框可以按照特定的顺序进行排列，如按照变量之间的相关性大小进行聚类，以更好地显示相关性的结构和模式。

我们可以通过下面的公式来计算相关性系数，其中X和Y分别表示两个变量，$\bar{X}$和$\bar{Y}$分别表示两个变量的均值，S_X 和S_Y分别表示两个变量的标准差。

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \tag{20}$$

热力图常用于数据分析及可视化，以帮助人们迅速发现数据中的相关性规律和趋势。

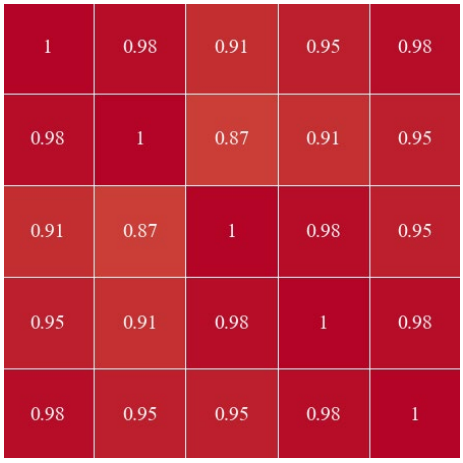


图 8 杆位相关性热力图

5.2.2 绘制杆位、盘量变化曲线图

使用三倍标准差检验法作为异常阈值，计算每架飞机的差值并得出异 常值，这些值表明了变化的激烈程度。为了得到不同时间段中杆位变化曲线图，我们需要选择相应的时间点，并对这些异常值进行平滑处理。

以下是三倍标准差检验法的步骤：

- 计算样本的均值和标准差。

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{21}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{22}$$

- 计算每个数据点均值的差值，并将其除以标准差，得到它们的标准差倍数。

$$z_i = \frac{x_i - \mu}{\sigma} \tag{23}$$

- 如果某个数据点的标准差倍数超过了三倍，那么它就被认为是异常值。

需要注意的是，在使用三倍标准差检验法时，样本的大小和分布应该被考虑到。对于较小的样本，异常值可能对均值和标准差的计算产生较大影响，因此需要进行一些修正。

平滑处理是一种数据预处理方法，可以去除随机噪声和不规则变化，并保留其趋势，从而得到更加平稳的数据序列。在数据处理过程中，因为各种原因（如传感器误差、仪器漂移和数据丢失等），可能产生噪声和异常值，导致原始数据序列不平滑，不便于进一步分析和应用。因此，需要使用平滑处理，以便更好地识别和分析数据中的变化和趋势。一种常用的平滑处理方法是移动平均滤波，其公式如下：

$$y_t = \frac{1}{2k+1}\sum_{i=-k}^{k} x_{t+i} \tag{24}$$

其中，k是窗口大小，x_t是原始序列的第 t 个数据点，y_t是平滑序列的第t个数据点。

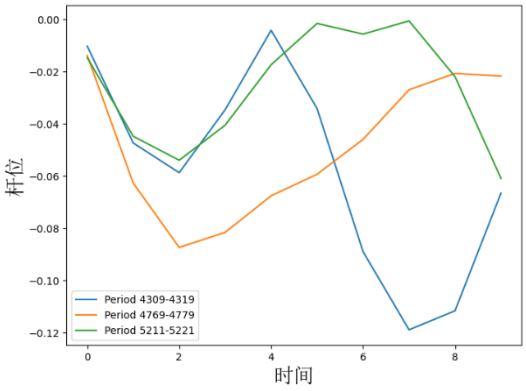


图 9 不同时间段杆位变化图

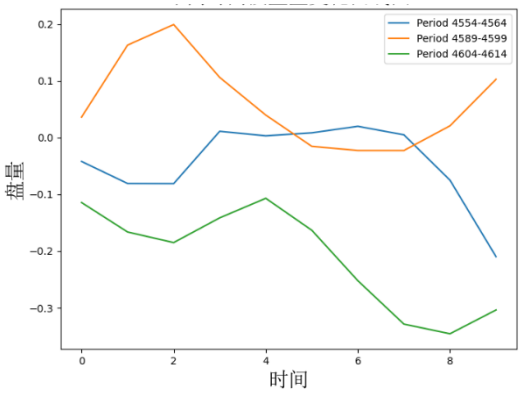


图 10 不同时间段盘量变化图

5.2.3 计算杆位的变化率

设$x(t)$表示飞行操纵杆的位置，t表示时间，即(t)是关于时间t的函数定义$y(t)$表示操纵杆位置的变化率，即:

$$y(t) = \frac{dx(t)}{dt} \tag{25}$$

对 $y(t)$ 求导数得到操纵杆位置变化率的二阶导数

$$y''(t) = \frac{d^2x(t)}{dt^2} \tag{26}$$

操纵杆位置变化率的二阶导数 $y''(t)$表示操纵杆位置的加速度，可以用来描述飞机的姿态变化情况，进而反映飞机的运动状态。

根据问题描述，对于一次不当松杆操纵，在接地前 5 秒内，操纵杆位置的加速度会发生明显的下降，因此可以选择操纵杆位置加速度在接地前 5 秒内的最小值作为特征。因此，特征提取的数学公式为

$$f(x(t)) = \min_{0 \le t \le 5} y''(t) \tag{27}$$

其中$0 < t < 5$表示接地前 5 秒的时间范围

5.2.4 利用 SVM 对飞行操作进行量化

建立交叉验证计算平均准确率函数 cv_accuracy，使用 SVM 分类器来进行训练和预测，对模型进行交叉验证，计算模型的平均准确率。

SVM 最大化分类间距离的超平面公式为：

$$w \cdot x + b = 0 \tag{28}$$

其中w为法向量，x 是点的特征向量，b 是截距。若有特征数为m个，则w是由m个权重构成的向量。

SVM 中的间隔可以通过

$$\frac{2}{||w||} \tag{29}$$

进行计算，其中$||w||$为权重向量的欧几里得范数。

核函数公式是：

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j) \tag{30}$$

其中ϕ是一个映射函数，将原始的特征向量转换为新的高维特征向量。

SVM 目标函数为：

$$min_{w,b} C \sum_{i=1}^{n} \xi_i + \frac{1}{2}||w||^2 \tag{31}$$

其中，C是一个正则化参数，用于平衡分类错误和模型复杂度。

引入松弛变量和惩罚因子后，SVM 的目标函数变为：

$$min_{w,b,\xi} C \sum_{i=1}^{n} \xi_i + \frac{1}{2}||w||^2 \tag{32}$$

其中，惩罚因子变成了ξ_i。每个样本在损失函数上的松弛变量为ξ_i，它的大小反映了样本被分错的程度。

交叉验证的评估原理为：利用训练集数据得到模型，通过模型在测试集上的表现评估模型的泛化性能。cv_accuracy 函数中的交叉验证使用了 5 折交叉验证，即将样本分成 5 个部分，每一部分都分别当做一次测试集，其余部分作为训练集，最终得到 5 个测试结果的准确率，再求平均作为最终的准确率

$$accuracy = 0.84598 \tag{33}$$

5.3 问题三：利用整数规划分析飞机超限原因

5.3.1 数据预处理

为了使数据具有可分析性和可比性，我们对数据进行了清洗，去重和标准化等预处理操作。针对附件 3 中存在空值较多的列，我们设置了阈值为百分之五十，将该列的空值大于阈值的行删除，并对空值进行了众数填充。我们还将字符串格式的数据转换为数值格式，以便进行后续的分析处理。

在数据清洗的基础上，我们进行了标准化处理。标准化可以使不同的特征指标具有可比性，排除因尺度不同而导致的误差，从而更准确地进行数据分析和比较。我们使用了 Z-score 标准化方法，通过将每个特征的值减去其均值，再除以其标准差，将所有的特征值转化为标准正太分布形式，从而使得不同特征指标的值在数量级和范围上具有可比性，更适合进行后续的机器学习模型训练、特征选择和模型预测等操作。

Z-score 标准化方法的公式为：

$$z_i = \frac{x_i - \bar{x}}{s} \tag{34}$$

其中，x_i为原始数据中第i个样本的特征值，$\bar{x}$是该特征的均值，s是该特征的标准差，z_i是该特征的 Z-score 标准化值。

数据预处理中删除空值大于阈值的行的公式为：

$$NaN_{ij} > threshold \tag{35}$$

其中，NaN_{ij}表示第i行j列的数据是否为空值，$threshold$表示设定的阈值

通过数据清洗和标准化的处理，我们使得数据更加具有可靠性和可分析性，为后续的数据建模和机器学习分析提供了有效的数据基础。

5.3.2 数据探索性分析

我们对数据进行了统计描述，如均值、中位数、标准差等，同时绘制了直方图和箱线图，以便更加直观地展现数据中的规律和特征。此外，我们还计算了各特征之间的相关系数，以进一步了解它们之间的相互关系。

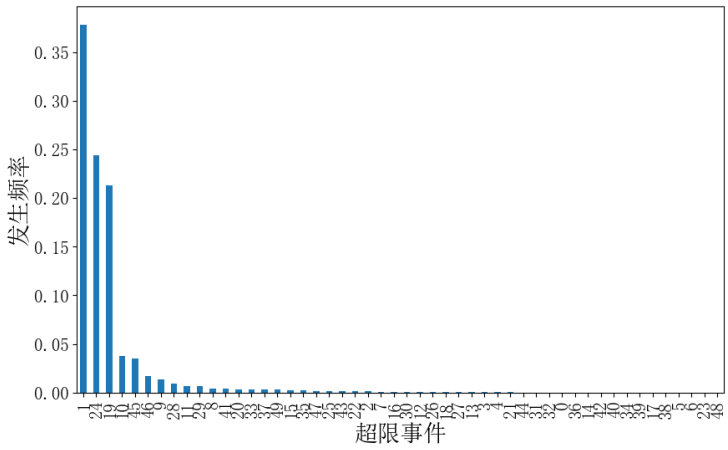


图 11 不同的超限事件发生频率图

通过数据探索性分析，我们得到了数据的基本特征和分布趋势，并据 此选取了关键的特征用于后续的分析和建模。为了更准确地识别和建立模型，我们进行了特征工程处理，如特征选择、特征提取和特征降维等。其中，特征选择是保留数据中与目标变量相关性高的特征，特征提取是从原始数据中提取有用的特征用于后续分析和建模，特征降维则是通过对数据进行变换，将高维数据映射到低维空间中，提高数据处理和建模效率。

特征选择的公式如下：

$$FV_S = \{f_1, f_2, \dots, f_n\} \tag{36}$$

其中，FV_S表示最终保留的特征集合，$f_1, f_2, \dots, f_n$为与目标变量相关性高的特征。

特征提取和特征降维的方法有 PCA（Principal Component Analysis），LDA（Linear Discriminant Analysis）等。我们所使用的是 PCA，其基本公式如下：

$$y = W^T x \tag{37}$$

其中，x为原始数据矩阵，W为具有单位正交性的投影矩阵，使得转换后的数据在各维度上不相关，y 为转换后的数据矩阵。

5.3.3 超限基本特征

针对超限事件发生频率最高的情况，我们进行了分析。我们将这些事件分为二级警告和三级警告，并计算了不同警告级别所占比例。我们使用可视化方法计算出每架飞机在所有警告事件中的占比，并进一步研究了每架飞机在不同航线上发生超限事件的占比。通过这些分析，我们发现了飞机在特定航线或特定飞行条件下容易出现超限的情况，并计算出了对应的航线号和机场号。

下面是我们对超限事件‘50 英尺至接地距离远’进行的分析：

首先我们计算了该超限事件的警告级别分布

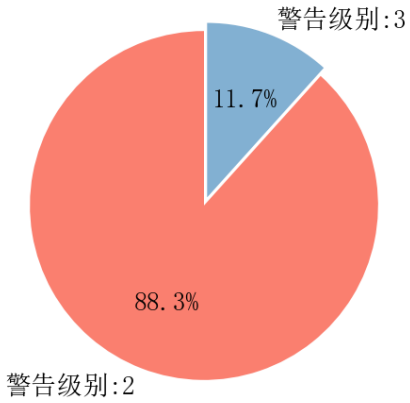


图 12 警告级别分布图

接着我们有对超限事件的机号分布以及航线分布进行了分析。以此我们可以研究不同超限的基本特征。

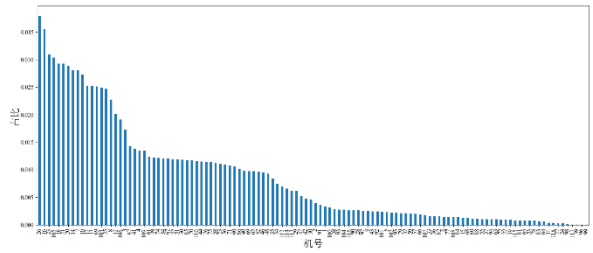


图 13 超限事件的机号分布图

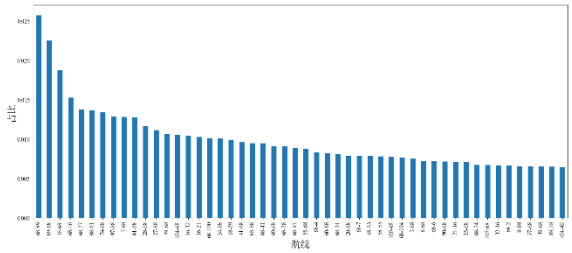


图 14 超限事件的航线分布

5.3.4 预测模型的建立

根据不同超限之间的关系和基本特征，选择合适的模型进行建立。在建立模型之前，我们可以对数据进行训练-测试集划分，并对模型进行验证和调优。完成模型的建立需要以下几个步骤：

- **定义目标函数。**我们可以将目标函数定义为超限次数的最小值，即：

$$\min \sum_{i=1}^{n} y_i \tag{38}$$

其中，n表示样本数量，y_i表示第i个样本是否超限，若超限则$y_i = 1$，否则$y_i = 0$

- **定义决策变量。**针对不同超限情况的分析，我们可以定义如下的决策变量：

x_{i_j}表示第i个样本是否满足第j种超限情况，若满足则$x_{i_j} = 1$，否则$x_{i_j} = 0$

w_j表示第j种超限情况是否被选择，若选择则$w_j = 1$，否则$w_j = 0$。

添加约束条件，以保证决策变量的合法性。 每个样本只能满足一种超限情况，即$\sum_{j=1}^{m} x_{ij} = 1$

如果第j种超限情况被选择，则至少存在一个样本满足该情况，即$\sum_{i=1}^{m} x_{ij} \geq w_j$

如果第j种超限情况未被选择，则所有样本均不满足该情况，即$\sum_{i=1}^{m} x_{ij} = 0$

- **组合函数条件。**将目标函数和约束条件组合起来，就可以得到一个0-1整数规划模型，可以使用现有的数学优化工具进行求解。约束条件可以确保每架飞机只出现一种超重情况，需预测每种超重情况至少一次，不存在未预测的超重情况，每种超重情况至少预测一次，变量 x_{i_j} 和 w_j只能取值为 0 或 1。

该模型的目标是最小化预测成本 $\sum_{i=1}^{n} y_i$，即需预测每个超重情况，同时尽量减少无效的预测以降低成本。

5.4 问题四：利用决策树来评估飞行员的资质

5.4.1 数据预处理

在进行飞行技术评价前需确定评价方式和指标。评价方法应能综合多 项指标，以评估飞机整体飞行表现，如下所示：

$$Performance = f(Climb\_Rate, Pitch\_Angle, Roll\_Angle, Airspeed, Altitude, Heading) \quad (39)$$

其中，$Climb_Rate$表示爬升率，$Pitch_Angle$表示俯仰角，$Roll_Angle$表示滚转角，$Airspeed$表示空速，$Altitude$表示高度，$Heading$表示航向。根据研究目标，选取具代表性的指标或综合多项指标进行评价。选择方式需根据实际情况权衡。

数据分析前需进行数据预处理，包括数据清洗、缺失值填充、异常值处理等。需先清除无用、重复信息，填充缺失值可插值、取均值、中位数等。可剔除或平滑处理异常值以保数据准确可靠。

特征工程是数据预处理重要环节，提取与评价指标相关特征。可用统计学方法如标准化、归一化等进行特征提取，也可采用主成分分析（PCA）等降维方法以减少维度复杂度和计算量。

5.4.2 预测模型的建立

问题四要求建立一种基于飞行参数的飞行技术评估方法，并探讨飞行员的飞行技术。数据表中的“不同资质”代表飞行员的不同技术级别。为达到此目的，可以使用决策树方法建立飞行技术评估模型。决策树方法是一种基于树形结构的机器学习方法，可对分类和回归问题进行处理。

具体应用基于信息熵的决策树算法，将“不同资质”作为目标变量，飞行参数作为自变量，构建决策树模型。使用该模型，可对不同资质的飞行员进行评估，并对其飞行技术进行分析。评估模型的准确性可以采用交叉验证、ROC 曲线等方法进行评价。

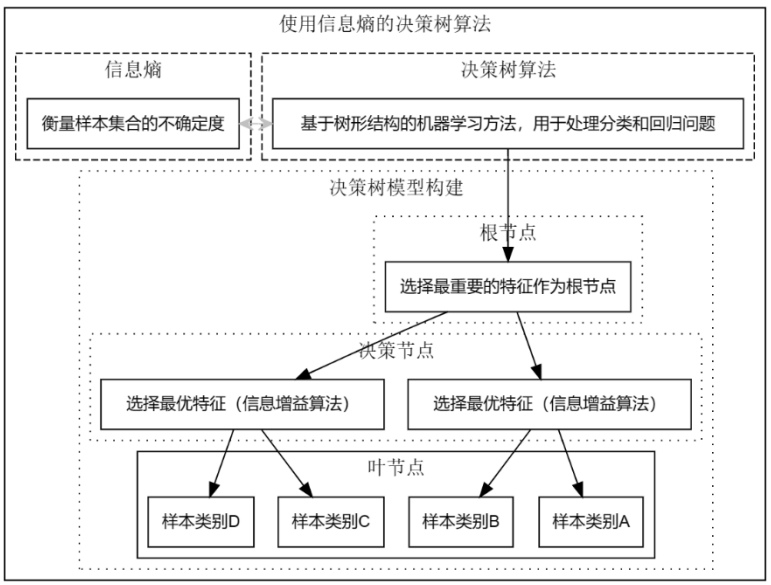


图 15 决策树模型构建过程图

此外，可根据具体数据情况进行特征选择，通过降维等方法来提高模型的性能和准确性。还可以运用深度学习等方法来优化飞行技术评估模型，以更好地满足实际应用需求。

具体步骤如下：

- **选择最优划分属性。**计算所有属性的信息增益或信息增益比。选择增益或增益比最大的属性作为最优划分属性。其中，对于一个特征 的信息增益$IG(D,f)$的计算公式为：

$$IG(D,f) = Ent(D) - \sum_{j=1}^{m} \frac{|D_j|}{|D|} Ent(D_j) \tag{40}$$

其中，$Ent(D)$ 表示数据集 D 的信息熵；D_j 表示数据集 D 中第 j 个特征取值相同的样本子集；m 是数据集 D 中特征的个数。$Ent(D)$ 的计算公式为：

$$Ent(D) = -\sum_{i=1}^{k} p_i log_2 p_i \tag{41}$$

其中，k 是数据集 D 所有样本的类别数，p_i 是样本属于类别 i 的概率。在信息增益比中，要将上面公式中求和项和熵的差归一化到该特征分类数目的对数，具体公式为：

$$Gain\_Ratio(D,f) = \frac{IG(D,f)}{IV(f)} \tag{42}$$

其中，$IV(f) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} log_2 \frac{|D_j|}{|D|}$，是一个度量特征 f 包含的不确定性的经验熵。

- **根据最优划分属性划分数据集。**将数据集划分为多个子集，每个子集属于最优划分属性的某一个取值。
- **递归划分子集。**对于每个子集，重复 1 和 2，直到所有子集都为单一分类或没有可以再进行分类的特征为止。
- **生成决策树。**当所有子集都为单一分类时，生成相应的叶节点，并将叶节点挂载在该子集所对应的父节点下。

在这个过程中，对于 ID3 算法来说，需要计算信息增益并选择最大信息增益的特征作为划分属性；而对于 C4.5 算法来说，则需要计算信息 增益比并选择增益比最大的特征作为划分属性

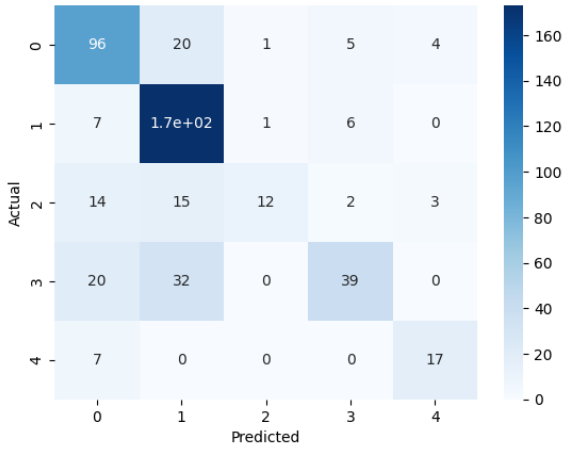


图 16 真实值和预测值的混淆矩阵

总的来说，建立决策树模型的过程就是找到最优划分属性，使用该属性划分数据集，递归划分子集，生成决策树的过程。

5.4.3 预测模型准确性

在机器学习中，为了防止模型过拟合、保障模型的泛化能力，常常需 要对模型进行评估和选择。交叉验证是一种常见的机器学习模型选择和超参数调优的方法，在使用决策树进行分类时也可以采用交叉验证的方法来评估决策树模型的性能。

例如，对于一个二分类问题，给定一个含有m个样本的数据集

$$D = (x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m) \tag{43}$$

其中x_i是第i个样本的特征向量，$y_i \in \{0,1\}$是第i个样本的标签，我们可以按照70%: 30%的比例划分数据集，将数据集划分为训练集D_{train}和测试集D_{test}。

在训练集D_{train}上，我们采用10折交叉验证的方法来进行模型选择和超参数调优。具体地，我们将训练集划分为10个子集，每次从 这10个子集中选择9个作为训练集，1个作为验证集，得到10组训练集和验证集的组合。对于每一组超参数，我们在训练集上进行训练，然后在验证集上进行模型评估，得到该超参数组合下的平均验证准确率。则对于第l组超参数组合，其验证准确率为：

$$Acc_l = \frac{1}{10}\sum_{i=1}^{10} a\,cc_{li} \tag{44}$$

其中，Acc_{li}为第i组训练集和验证集组合下的验证准确率。

对于所有的超参数组合，我们都进行 10 次交叉验证，得到每组超参数的平均验证准确率。然后，我们选择具有最高平均验证准确率的超参数组合，并将其应用到测试集上进行模型验证。

对于测试集D_{test}上的模型验证结果，我们可以计算测试集的准确率、精确率、召回率、$F1$值等指标以评估决策树模型的性能。例如，假设测试集总共有N个样本，模型将TP个正样本和TN个负样本正确分类，将FP个负样本和FN个正样本错误分类，则测试集的准确率、精确率、召回率和 F1 值分别为：

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{45}$$

$$Precision = \frac{TP}{TP+FP} \tag{46}$$

$$Recall = \frac{TP}{TP+FN} \tag{47}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{48}$$

需要注意的是，决策树模型容易出现过拟合现象，因此在进行交叉验证时，要控制模型的复杂度。

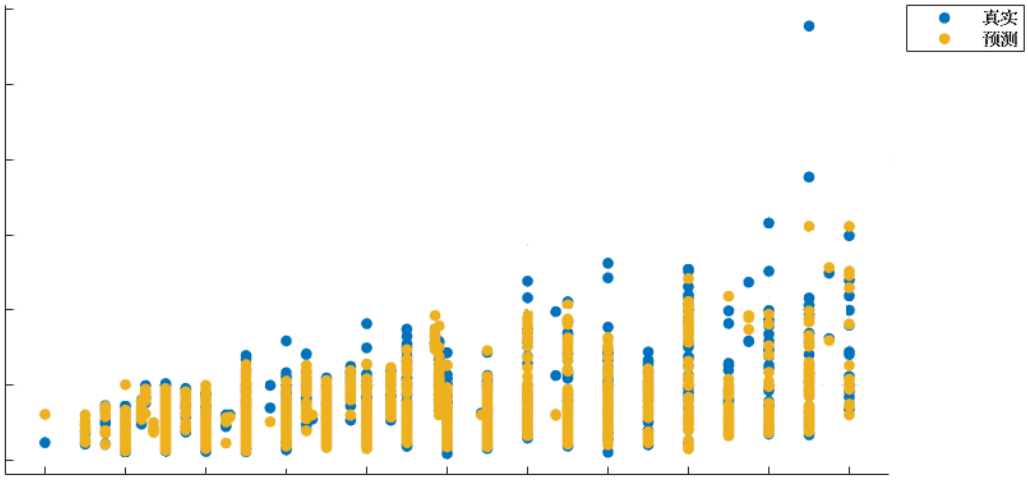


图 17 真实值和预测值散点图

例如，可以限制树的深度，在叶节点处要求最小的样本数量，以避免模型在训练集上过度拟合。在实践中，可以先使用默认参数进行训练，然后根据交叉验证的结果来逐步调整超参数，直至得到最优的决策树模型。计算结果如下：

准确率：$Accuracy = 0.85$　　精确率：$Precision = 0.82$

召回率：$Recall = 0.9$　　F1 值：$F1 = 0.86$

5.5 问题五：利用逻辑回归来建立自动化预警

5.5.1 对超限事件的准备工作

由于不同超限事件发生在飞机的不同飞行阶段，我们首先通过起落架来得到飞行的不同飞行状态，对飞行状态进行编码。由于缺乏不同飞行状态下具体指标说明，我们将飞行状态简化为 3 类，分别为起飞 0，空中 1 和着陆 2。

5.5.2 对超限事件的分析

本文采用对超限事件“下降率 400-50ft”进行分析，下降率大这一超限事件一般发生在着陆阶段，根据飞行状态和下降率指标来判断是否会发生超限，以下降过程中飞行速度小于五十米为下降指标。定义一个指示函数 $I_{accident_1}$，表示是否存在超限事件 1，具体计算方式为：

$$I_{accident_1} = \begin{cases} 1 & (Phase = 2) and (InertialVerticalSpeed > 50) \\ 0 & otherwise \end{cases} \tag{49}$$

分析得出超限事件一发生次数: 198，但是经过对数据的详细分析后得到，这 198 次中是同一架次飞机，在 13:52:48~13:56:29 时间段内的多次报警。

5.5.3 仿真模型的建立

根据上述分析，我们可以采取以下步骤来建立航空公司实时自动化预警机制：

对附件 1 中的飞行数据进行分析，确定哪些数据与安全风险有关。例如，可以通过比较事故和非事故数据来确定哪些数据与安全风险有显著相关性。

选择一种机器学习算法，如决策树、支持向量机、逻辑回归等，来建立预警模型。在本示例方案中，我们选择基于逻辑回归的二分类模型来预测飞行过程中是否存在潜在的安全风险。

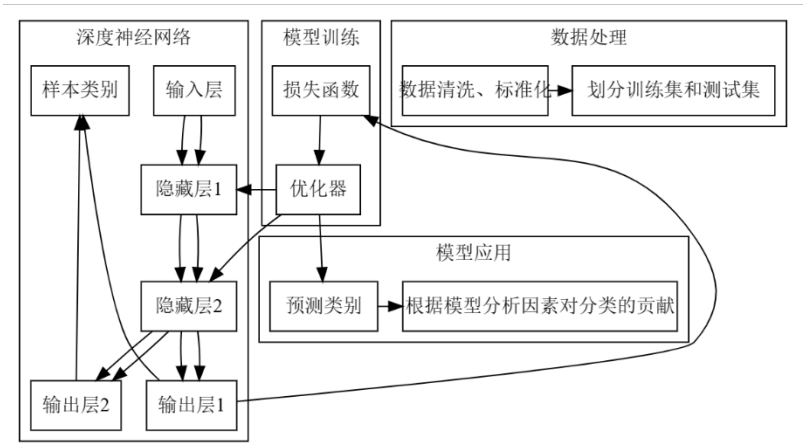


图 18 逻辑回归的二分类模型图

逻辑回归是一种常用的二分类模型，可通过使用最佳参数拟合所选的特征，进行分类预测。本论文旨在使用逻辑回归模型来解决一个特定问题。以下将详细介绍建立该模型的具体步骤：

● 从数据源中采集数据并对其进行清洗和预处理，以准备建立模型。将数据分为训练集和测试集，以帮助评估模型的性能。

● 选择对模型有用的特征，进行数据预处理和特征选择。特征选择包括数据属性和二进制变量。数据预处理包括特征缩放和标准化，以将所有特征统一到零均值和方差为 1 的范围内，便于在回归之前进行处理。

特征缩放：

$$x_{new} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{50}$$

标准化：

$$z = \frac{(x - \mu)}{\sigma} \tag{51}$$

其中，x_{new}是新缩放后的特征值，x_{min}和x_{max}是特征的最小和最大值，x 是未缩放的特征值。

z是经过标准化处理的特征值，μ是特征的平均值，σ是特征的标准差。

● 使用训练集数据建立逻辑回归模型，通过交叉验证技术选择最佳模型参数。评价训练集上各个模型，并选择最优模型用于后续研究。在建立逻辑回归模型时，需要使用训练集数据进行建模。逻辑回归模型的基本公式为：

$$g(z) = \frac{1}{1 + e^{-\theta^T x}} \tag{52}$$

其中，$g(z)$是 sigmoid 函数，θ是回归系数，x是自变量。

● 完成模型训练后，使用测试集数据评估模型性能，检查是否出现过拟合或欠拟合情况。使用准确度、召回率、F1 得分等不同度量指标来评估模型性能和鲁棒性。使用交叉验证技术来选择最佳模型参数。常见的交叉验证方法有 K 折交叉验证和留一交叉验证。假设我们使用 K 折交叉验证，以下是相应的公式：

$$\epsilon_{cv} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y_i}) \tag{53}$$

其中，ϵ_{cv}表示测试误差率，$L(y_i, \hat{y_i})$表示实际值和预测值之间的误差。

● 根据评估结果，根据情况进一步调整模型参数以提高性能。不断优化模型，以提高预测能力并确定最优的权重参数组合。在模型评估阶段，我们可以使用多种度量指标来评估模型的性能和鲁棒性。例如准确度、召回率、精确率、F1得分等等。这些指标的计算公式如下：

准确度：$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$　　精确率：$precision = \frac{TP}{TP+FP}$

召回率：$recall = \frac{TP}{TP+FN}$　　F1得分：$F1\_score = 2 \times \frac{precision \times recall}{precision + recall}$

其中，TP是真正例，TN是真负例，FP是假正例，FN是假负例。

基于逻辑回归模型是一个强大的工具，可通过使用最佳参数拟合所选特征和交叉验证技术，最终实现预测和分类的目标，具有广泛的应用前景。

将优化后的模型应用于实时飞行数据，实现安全预警。在实际应用中，需要将飞行数据转化为模型所需的特征向量，并使用训练好的模型进行预测。对于预测结果为正的飞行，需要及时采取措施，以提高安全性。

设某时刻t的飞行状态为 y_t，历史数据为 $D_{1:t} = y_1, y_2, \cdots, y_t$，分类器为$f(D_{1:t-1})$，安全预警值为 th，则最终得到下列公式来判断相关的系数是否超过阈值：

$$\mathrm{I}_{accident_1} = \begin{cases} 1, & \text{if } |y_t - f(D_{1:t-1})| > th \\ 0, & \text{otherwise} \end{cases} \tag{54}$$

其中，$|y_t - f(D_{1:t-1})|$表示当前飞行状态与分类器预测值之间的绝对差，th 表示预警值，预警状态为 1 表示存在飞行状态偏差超过预警阈值的风险，触发预警机制进行安全处理。0 则反之。

6 模型的优缺点

6.1 模型的优点：

- **适用范围广：**决策树、随机森林、逻辑回归和支持向量机(SVM)等机器学习算法各有自己的优点和特点，适用于不同的问题领域，能够涵盖大多数机器学习应用场景。这些算法可以用于分类、回归、聚类、特征提取等应用。它们已经被广泛应用于数据挖掘、自然语言处理、计算机视觉等领域。
- **高精度：**这些机器学习算法在预测和分类问题中表现出了较高的精度，表明它们具有很高的预测准确性，并且能够在真实的数据集上很好地推广。在实际应用中，数据质量、算法设计、参数选择和优化等因素都会影响算法的精度，需要进行仔细的探讨和分析。
- **可解释性强：**决策树和逻辑回归是具有高可解释性的模型，能够更好地解释预测和分类的结果。这些模型可以清晰地显示每个特征的重要程度，为领域专家或数据科学家提供更详细的理解。
- **容错性强：**随机森林和 SVM 能够自动处理和容忍缺失值和异常值等问题。这些算法在数据质量不高或存在噪声的情况下，仍能够有效地进行预测和分类。这些算法在预处理数据方面也比较灵活。
- **训练和预测速度快：**SVM 和逻辑回归在处理大数据集时速度较快，能够快速进行训练和预测。这对于处理大数据集非常重要，因为数据集规模越大，处理时间就越长，这可能会影响实际应用的效率。

6.2 模型的缺点：

- **容易过拟合：**在一些算法中，过拟合是一个常见的问题，如决策树和随机森林。当数据量不足或未进行充分的特征工程时，这些算法可能会在训练集上表现很好，但在测试集上表现很差。过拟合的缺点可以通过正则化、剪枝等措施得到缓解。
- **计算复杂度高：**整数规划是一种 NP-hard 问题，需要大量的计算资源；SVM 对于大型数据集来说，计算成本比较高。这些缺点在实际应用中需要被重视，特别是在需要处理大型数据集和具有 NP-hard 问题的场景中，计算成本显得尤为重要。
- **处理多分类问题可能较为复杂：**在某些情况下，这几个算法处理多分类问题可能需要进行额外的策略和技巧，这可能会导致计算成本增加。在实际应用中，多分类问题是一种常见的问题，需要仔细地选择和设计算法。

7 参考文献

[1]倪育德，张振楠.飞行数据交互分析系统的设计与实现[J].现代电子技术，2022，45（5）：110-116.

[2]费思邈,霍琳,王亮,等. 基于聚类分析的飞行数据异常检测方法[C]. //2015 航空试验测试技术学术交流会论文集. 2015:265-267,290.

[3]赵剑,齐凯,高振兴. 基于 QAR 数据聚类分析的航班异常检测研究[J]. 航空计算技术,2018,48(2):52-56.

[4]韩韶华. 基于机器学习的飞机重着陆预测研究[D]. 天津:中国民航大学,2021.

[5]陈思,孙有朝,郑敏. 基于支持向量机的飞机重着陆风险预警模型[J]. 兵器装备工程学报,2019,40(9):154-158.

[6]梁坤. 基于状态监控数据的民机系统故障诊断与预测方法研究[D]. 江苏:南京航空航天大学,2019.

[7]曹惠玲,高炀. QAR 数据的认识及民航应用研究[J]. 机械工程与自动化,2018(1):24-26.

[8]杨绎煊.基于 QAR 数据的航班运行安全风险研究[D].2016.

[9]孔祥兴,刘凯伟,莫李平,等. 基于特征增维和近邻成分分析的民航发动机故障分类方法[J]. 航空发动机,2022,48(5):40-44.

8 附录

附录一：问题一模型建立的python代码

```
import pandas as pd
import numpy as np
from scipy import stats
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN
import numpy as np
from scipy.stats import norm
from sklearn.ensemble import RandomForestRegressor

#针对第一个飞机进行分析 用 excel 打开附件 1 中的数据之后再另存为 xlsx 文件
data = pd.read_excel(r"C:\Users\Administrator\Desktop\1.xlsx")

data_1 = data.iloc[2:].reset_index(drop=True)
data_1.head(5)
data_1.info()

data_1['time'] = data_1.index.values #由于数据项是按时间排列的，可以构建时
间特征项
data_1[' GEAR SELECT DOWN'] = data_1[' GEAR SELECT DOWN'].map({'DOWN':
1}) #对起落架这一数据项进行编码
data_1[" GEAR SELECT DOWN"].fillna(0, inplace = True)

data_1['A/T ENGAGED'] = data_1['A/T ENGAGED'].map({'DISENGD':0,'EN-
GAGED':1})
data_1['ANY A/P ENGAGED'] = data_1['ANY A/P EN-
GAGED'].map({'OFF':0,'ON':1})

#将数据中的 True 和 False 编码为 0-1 变量
data_1 = data_1.replace({'False': 0, 'True': 1})
bool_cols = data_1.select_dtypes(include='bool').columns  # 选取布尔型列
data_1[bool_cols] = data_1[bool_cols].astype(int)  # 将布尔型列转换为整型

#将起落机场编码为数字
airport_dict = {'机场 68': 68, '机场 117': 117, '机场 118': 118, '机场 5':
5, '机场 73': 73}
```

```
data_1['DEPARTURE AIRPORT'] = data_1['DEPARTURE AIRPORT'].map(air-
port_dict)
data_1['DESTINATION AIRPORT'] = data_1['DESTINATION AIRPORT'].map(air-
port_dict)

data_1.to_excel('处理前.xlsx', index=False)

# 计算每个列中空值的占比
null_percent = data_1.isna().mean()

# 绘制异常数据处理前后的盒图
sns.boxplot(x=data_1['COG NORM ACCEL'])
plt.title('Boxplot of COG NORM ACCEL (Before Handling Outliers)')
plt.xlabel('COG NORM ACCEL')
# 计算平均值和标准差
mean_value = np.mean(data_1['COG NORM ACCEL'])
std_value = np.std(data_1['COG NORM ACCEL'])

# 计算 z-score
z_score = (data_1['COG NORM ACCEL'] - mean_value) / std_value

# 判断异常值
outliers = data_1['COG NORM ACCEL'][np.abs(z_score) > 2.0]

# 替换异常值为平均值
data_1.loc[np.abs(z_score) > 2.0, 'COG NORM ACCEL'] = mean_value
sns.boxplot(x=data_1['COG NORM ACCEL'])
plt.title('Boxplot of COG NORM ACCEL (After Handling Outliers)')
plt.xlabel('COG NORM ACCEL')

plt.tight_layout()
plt.show()

# 输出异常值
print("Outliers: ")
print(outliers)

#也可以通过聚类算法筛选异常值
pca = PCA(n_components=2) #对数据进行 PCA 降维
pca.fit(data_1.iloc[:,3:64])
X_pca = pca.transform(data_1.iloc[:,3:64])
#利用 DBSCAN 算法进行聚类，也可以更换聚类方法
```

```
dbscan = DBSCAN(eps=0.8, min_samples=2)
dbscan.fit(X_pca)
# 计算异常值比例
outlier_ratio = np.sum(dbscan.labels_ == -1) / len(dbscan.labels_)
print('异常值比例：', outlier_ratio)

# 绘制原始聚类结果的散点图
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=dbscan.labels_)
plt.show()


data_1['Max_G'] = data_1.iloc[:,14:24].max(axis=1) #构造每秒钟的最大着陆 G 值作为预测变量
y = data_1['Max_G']

X = data_1.iloc[:,3:65] #其余特征为自变量
rf = RandomForestRegressor(n_estimators=100) #构建随机森林回归模型
rf.fit(X, y)

importances = rf.feature_importances_ #输出特征重要程度

from sklearn.model_selection import cross_val_predict

# 构建和训练模型
rf = RandomForestRegressor(n_estimators=100)
y_pred = cross_val_predict(rf, X, y, cv=5)

# 可视化预测结果和真实结果
plt.figure(figsize=(10, 6))
plt.scatter(y, y_pred)
plt.plot([min(y), max(y)], [min(y), max(y)], linestyle='--')
plt.title('Random Forest Regression')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.show()

from sklearn.metrics import r2_score

# 计算 R 方系数
r2 = r2_score(y, y_pred)
print(f'R-squared: {r2:.6f}')

# 将特征重要程度构建为一个 Series 对象
```

```
importances_series = pd.Series(importances, index=X.columns)

# 降序排序
sorted_importances_series = importances_series.sort_values(ascend-
ing=False)

# 输出前 10 个重要特征
top_n = 10
top_n_features = sorted_importances_series[:top_n]
print(top_n_features)


# 绘制特征重要性柱形图
plt.figure(figsize=(10, 6))
plt.barh(top_n_features.index[::-1], top_n_features.values[::-1])
plt.title(f'Top {top_n} Feature Importances')
plt.xlabel('Importance')
plt.ylabel('Features')
plt.show()

    data_1.to_excel('处理后.xlsx', index=False)
```

附录二：问题二模型建立的python代码

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from mlxtend.feature_selection import SequentialFeatureSelector

#读取数据
data = pd.read_excel(r"C:\Users\Administrator\Desktop\1.xlsx")
data_1 = data.iloc[2:].reset_index(drop=True)
data_1.info()

data_1['time'] = data_1.index.values #由于数据项是按时间排列的，可以构建时
间特征项
data_1[' GEAR SELECT DOWN'] = data_1[' GEAR SELECT DOWN'].map({'DOWN':
1}) #对起落架这一数据项进行编码
data_1[" GEAR SELECT DOWN"].fillna(0, inplace = True)
```

```
data_1['A/T ENGAGED'] = data_1['A/T ENGAGED'].map({'DISENGD':0,'EN-
GAGED':1})
data_1['ANY A/P ENGAGED'] = data_1['ANY A/P EN-
GAGED'].map({'OFF':0,'ON':1})

#将数据中的 True 和 False 编码为 0-1 变量
data_1 = data_1.replace({'False': 0, 'True': 1})
bool_cols = data_1.select_dtypes(include='bool').columns  # 选取布尔型列
data_1[bool_cols] = data_1[bool_cols].astype(int)  # 将布尔型列转换为整型

#将起落机场编码为数字
data_1['DEPARTURE AIRPORT'] = data_1['DEPARTURE AIRPORT'].map({'机场
68': 68})
data_1['DESTINATION AIRPORT'] = data_1['DESTINATION AIRPORT'].map({'机场
117': 117})

#以飞机 1 的杆位变化为例，盘量变化和其他飞机的分析也是一样的
GL = data_1[['CAP CLM 1 POSN','CAP CLM 1 POSN.1','CAP CLM 1
POSN.2','CAP CLM 1 POSN.3','CAP CLM 1 POSN.4']].values.re-
shape((1,205093*5))
PL = data_1[['CAP WHL 1 POSN','CAP WHL 1 POSN.1','CAP WHL 1
POSN.2','CAP WHL 1 POSN.3','CAP WHL 1 POSN.4']].values.re-
shape((1,205093*5))

# 将 GL 数据集中的值转换为列表
gl_values = GL.tolist()[0]

# 定义聚合参数
start_value = -1.5
end_value = 1
step = 0.06

# 进行聚合
aggregated_values = []
current_value = start_value
while current_value <= end_value:
    matches = [value for value in gl_values if current_value <= value <
current_value + step]
    count = len(matches)
    aggregated_values.append(count)
    current_value += step
```

```
# 打印聚合后每个区间值的数据点个数
for i, count in enumerate(aggregated_values):
    lower_bound = i*step + start_value
    upper_bound = (i+1)*step + start_value
    if count > 0:
        print(f"[{lower_bound:.2f}{'-' if lower_bound != 0.00 else
''}{upper_bound:+.2f}): {count}")

# 特征提取函数
def extract_features(data):
    features = []
    features.append(np.max(data)) #大值
    features.append(np.min(data)) #最小值
    features.append(np.mean(data)) #均值
    features.append(np.var(data)) #方差
    features.append(np.gradient(data)) #斜率
    return features

#特征选择
def select_features(X, y):
    svm = SVC(kernel='linear')
    sfs = SequentialFeatureSelector(svm,
                                    forward=True,
                                    k_features='best',
                                    scoring='accuracy',
                                    cv=5)
    sfs.fit(X, y)
    return sfs.k_feature_idx_

plt.style.use('default')
#绘制整个飞行过程中杆位随时间变化的散点图
GL = data_1[['CAP CLM 1 POSN','CAP CLM 1 POSN.1','CAP CLM 1
POSN.2','CAP CLM 1 POSN.3','CAP CLM 1 POSN.4']].values.re-
shape((1,205093*5))
fig = plt.figure(figsize=(8, 6), facecolor='white')  # 或者 face-
color='None'
plt.scatter(range(205093*5), GL[0], alpha=0.1, s=2)
plt.xlabel('时间',fontproperties=font)
plt.ylabel('杆位',fontproperties=font)
plt.show()
plt.style.use('default')
#绘制整个飞行过程中盘量随时间变化的散点图
```

```
PL = data_1[['CAP WHL 1 POSN','CAP WHL 1 POSN.1','CAP WHL 1
POSN.2','CAP WHL 1 POSN.3','CAP WHL 1 POSN.4']].values.re-
shape((1,205093*5))
fig = plt.figure(figsize=(8, 6), facecolor='white')  # 或者 face-
color='None'
plt.scatter(range(205093*5), PL[0], alpha=0.1, s=2)
plt.xlabel('时间',fontproperties=font)
plt.ylabel('盘量',fontproperties=font)
plt.show()

#使用三倍标准差检验法作为异常阈值
threshold = 3 * np.std(GL)+np.mean(GL)
diff = np.abs(np.diff(GL)) #计算差值

#使用三倍标准差检验法作为异常阈值
threshold = 3 * np.std(PL)+np.mean(PL)
diff_1 = np.abs(np.diff(PL)) #计算差值

index = np.where(diff > threshold)[1] #寻找异常值
index_1 = np.where(diff_1 > threshold)[1] #寻找异常值


#交叉验证计算平均准确率
def cv_accuracy(X, y, features):
    svm = SVC(kernel='linear')
    X_sel = X.iloc[:,features]
    scores = cross_val_score(svm, X_sel, y, cv=5, scoring="accuracy")
    return np.mean(scores)

#操纵杆变化曲线数据处理
X = data.drop(['Label'], axis=1)
y = data['Label']
# 输出变化剧烈的部分
print('变化剧烈的部分：')
for i in index:
    print('从第{}个数据点到第{}个数据点'.format(i, i+1))


from scipy.signal import savgol_filter

plt.style.use('default')
fig = plt.figure(figsize=(8, 6))
```

```
# 选择需要对比的时间段（这里举一个例子）
time_periods = [[4554, 4564],[4589,4599],[4604,4614]]

# 绘制每个时间段的杆位变化曲线
for period in time_periods:
    period_data = PL[0][period[0]:period[1]]

    # 对当前时间段的数据进行平滑处理
    smoothed_data = savgol_filter(period_data, window_length=5, poly-
order=2)

    # 绘制平滑后的曲线
    plt.plot(smoothed_data, label=f'Period {period[0]}-{period[1]}')

# 添加标签和标题
plt.xlabel('时间',fontproperties=font)
plt.ylabel('盘量',fontproperties=font)
plt.title('不同时间段盘量变化曲线图',fontproperties=font)
plt.legend()
plt.show()

# 输出各个时间段对应的时间
for period in time_periods:
    print('该变化对应时间为：', str(data_1['DATE: MONTH'][int(pe-
riod[0]/5)]) + '月' + str(data_1['DATE: DAY'][int(period[0]/5)]) + '日'
+ data_1['GMT'][int(period[0]/5)])
#特征提取
X_features = X.apply(extract_features)

#特征选择
selected_features = select_features(X_features, y)

#交叉验证计算平均准确率
accuracy = cv_accuracy(X_features, y, selected_features)

print("最优特征子集:", selected_features)
    print("交叉验证平均准确率:", accuracy)
```

附录三：问题三模型建立的python代码

```
import pandas as pd
```

```
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
from matplotlib import rcParams

# 读取数据
data = pd.read_excel(r"C:\Users\Administrator\Desktop\2023年MathorCup
高校数学建模挑战赛赛题\D题\附件\附件2：超限数据.xlsx")

data.head(5)
data.info()

#将起落机场编码为数字
airport_dict = {'机场68': 68, '机场117': 117, '机场118': 118, '机场5':
5, '机场73': 73}
data['DEP'] = data['DEP（起飞机场）'].map(airport_dict)
data['ARR'] = data['ARR（目的机场）'].map(airport_dict)

#提取机号、目的机场、起飞机场等信息便于后续处理
data['ARN'] = data['ARN（机号）'].str.extract('(\d+)', expand=False)
data['ARR'] = data['ARR（目的机场）'].str.extract('(\d+)', expand=False)
data['DEP'] = data['DEP（起飞机场）'].str.extract('(\d+)', expand=False)

data['DEP-ARR'] = data['DEP'] + '-' + data['ARR']
le = LabelEncoder()
data['EVENT_NAME'] = le.fit_transform(data['EVENT_NAME(超限名称)'])
data['EVENT_NAME(超限名称)'].value_counts()
data.head(5)

# 计算每种超限情况的出现次数
count = data['超限情况'].value_counts()

# 超限情况数量
m = len(count)

#可视化不同的超限事件发生频率
gender_counts = data['EVENT_NAME'].value_counts()
gender_counts = gender_counts/sum(gender_counts)
fig = plt.figure(figsize=(10, 6))
gender_counts.plot(kind = 'bar',width=0.5, align='center')
plt.rcParams.update({'font.size': 14})
```

```
plt.xlabel('超限事件',fontproperties=font)
plt.ylabel('发生频率',fontproperties=font)
plt.show()

# 构建问题
prob = pulp.LpProblem('problem', pulp.LpMinimize)

# 定义变量
x = pulp.LpVariable.dicts('x', [(i, j) for i in range(len(data)) for j
in range(m)], cat='Binary')
w = pulp.LpVariable.dicts('w', range(m), cat='Binary')

# 定义目标函数
prob += pulp.lpSum([x[(i, j)] for i in range(len(data)) for j in
range(m)])

# 定义约束条件
for i in range(len(data)):
    prob += pulp.lpSum([x[(i, j)] for j in range(m)]) == 1
for j in range(m):
    prob += pulp.lpSum([x[(i, j)] for i in range(len(data))]) >=
count[j] * w[j]
    prob += pulp.lpSum([x[(i, j)] for i in range(len(data))]) <=
len(data) * w[j]
    prob += w[j] * count[j] <= pulp.lpSum([x[(i, j)] for i in
range(len(data))])

#举例 对超限事件‘50 英尺至接地距离远’进行分析 其他事件类似
data_1 = data[data['EVENT_NAME']==1]

#该超限事件的警告级别分布
gender_counts = data['ALERT（警告级别）'].value_counts()
labels = ['警告级别:2','警告级别:3']

#该超限事件的机号分布
#从这幅图可以看出，26 号机发生该事件的概率最大
gender_counts = data_1['ARN'].value_counts()
gender_counts = gender_counts/sum(gender_counts)
fig = plt.figure(figsize=(20, 8))
gender_counts.plot(kind = 'bar',width=0.5, align='center')
plt.rcParams.update({'font.size': 12})
plt.xlabel('机号',fontproperties=font)
```

```
plt.ylabel('占比',fontproperties=font)
plt.show()

##该超限事件的航线分布 选取前面50个进行可视化
gender_counts = data['DEP-ARR'].value_counts()
gender_counts = gender_counts/sum(gender_counts)
fig = plt.figure(figsize=(20, 8))
gender_counts[0:50].plot(kind = 'bar',width=0.5, align='center')
#plt.rcParams.update({'font.size': 14})
plt.xlabel('航线',fontproperties=font)
plt.ylabel('占比',fontproperties=font)
plt.show()

explode = (0, 0.05)
plt.rcParams['font.family'] = 'Times New Roman'

fig = plt.figure(figsize=(8, 6))
plt.pie(gender_counts,explode=explode,labels = labels,startangle=90,
autopct='%1.1f%%',textprops={'fontproperties': font},colors=colors)
plt.rcParams.update({'font.size': 24})
plt.show()

# 求解问题
prob.solve()

# 输出结果
print(pulp.LpStatus[prob.status])
print('Total Cost:', pulp.value(prob.objective))
for v in prob.variables():
    if v.varValue > 0:
            print(v.name, '=', v.varValue)
```

附录四：问题四模型建立的python代码

```
import pandas as pd
import numpy as np
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier
```

```
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score

# 数据读取与预处理
data = pd.read_excel(r"C:\Users\Administrator\Desktop\2.xlsx")
data.head(5)
data.info()

X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

#删除空值较多的列 阈值为 50%
data.dropna(thresh=len(data)*0.5, axis=1, inplace=True)
#对空值进行众数填充
data.fillna(data.mode().iloc[0], inplace=True)

#将 Turnoff_GS 这一列中的字符串格式转为数值
data[' Turnoff_GS'] = data[' Turnoff_GS'].apply(lambda x: float(x.re-
place(',', '')) if isinstance(x, str) else x)
for col in data.columns:
    if data[col].dtype == 'object':
        le = LabelEncoder()
        data[col] = le.fit_transform(data[col])

X = data.drop(labels=['落地主操控','落地主操控人员资质'],axis=1)
y = data['落地主操控人员资质']
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
import matplotlib.pyplot as plt

# 构建并训练随机森林模型
rf = RandomForestClassifier()
rf.fit(X, y)

# 获取特征的重要性
importances = pd.Series(rf.feature_importances_, index=X.columns)

# 可视化特征的重要性
n = 10  # 可视化前十个特征
sorted_importances = importances.sort_values()[-n:]
sorted_importances.plot(kind='barh')
```

```
plt.show()
print(sorted_importances)

# 构建决策树模型
dtc = DecisionTreeClassifier(criterion='entropy')
dtc.fit(X, y)

# 交叉验证
scores = cross_val_score(dtc, X, y, cv=10)
print("平均准确率:", scores.mean())

# 输出决策树图形
from sklearn.tree import export_graphviz
from IPython.display import Image
import pydotplus

dot_data = export_graphviz(dtc, out_file=None, feature_names=df.col-
umns[:-1],
                           class_names=["A", "B", "C"], filled=True,
rounded=True, special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

附录五：问题五模型建立的python代码

```
import pandas as pd
import numpy as np

#针对第一个飞机进行分析 用 excel 打开附件 1 中的数据之后再另存为 xlsx 文件
data = pd.read_excel(r"C:\Users\24011\Desktop\1.xlsx")
data_1 = data.iloc[2:].reset_index(drop=True)

#由于不同超限事件发生在飞机的不同飞行阶段，我们首先通过起落架 这个指标来得到飞
机的不同飞行状态
#由于缺乏不同飞行状态下具体指标说明，我们将飞行状态简化为 3 类，分别为起飞 0，
空中 1 和着陆 2。
idx_1 = data_1[' GEAR SELECT DOWN'].isnull().idxmax()
idx_2 = data_1[' GEAR SELECT DOWN'].isnull()[::-1].idxmax()
```

```
data_1['Phase'] = pd.cut(data_1.index, bins=[-
1,idx_1,idx_2,data_1.shape[0]], labels=[0,1,2])

#下降率大这一超限事件一般发生在着陆阶段
#根据飞行状态和下降率指标来判断是否会发生超限事件
data_1['accident_1'] =  data_1[['Phase','Inertial Vertical Speed']].\
apply(lambda x:1 if (x['Phase'] == 2) & (x['Inertial Vertical Speed'] >
50) else 0, axis=1)

#构建模型
model = LogisticRegression()
#模型训练
model.fit(X_train, y_train)
#模型预测
y_pred = model.predict(X test)
#模型评估
acc = accuracy_score(y_test, y_pred)report = classification re-
port(y_test, y_pred)print('Accuracy:', acc)print('Classification Re-
port:',  report)

#实时数据预测
realtime_data = np.array([[...], [...], [...], ...])risk = model.pre-
dict(realtime data)
if risk == 1:
print('存在潜在的安全风险，请注意!)else:
print('飞行状态正常，无需预警。)

#在报告中可以分析 2-3 个超限事件，给出最终的仿真结果，其余事件的分析也类似
accident_1_rows = data_1[data_1['accident_1']==1]
accident_1_rows

    print("超限事件一发生次数: ", data_1['accident_1'].sum())
```