航空安全风险分析和飞行技术评估问题

摘要

飞行安全是民航运输业发展的基础，随着中国民航的快速发展，研究飞行安全问题变得越来越重要。目前，航空安全研究主要集中在飞行数据的分析与应用上，其中包括超限事件和非超限数据的研究。航空安全研究主要关注超限事件和非超限数据的分析与应用，以及通过全航段数据建模和分析来改进飞行安全管理和训练效果。此外，着陆瞬间 G 值也是评估着陆安全性的重要指标。

对于问题一，要求提取部分关键数据，并对其进行重要性分析，本题采用 **IQR**（**四分位距指标**）处理**异常值**、**随机森林模型**进行**特征重要性分析**，得到关于飞行安全重要程度高的数据项为下降率、航道偏向、下滑道偏向。其余数据项对于飞行安全的重要性程度不高，

对于问题二，要求对飞行操纵进行合理量化描述，本题先采用线性回归法对自变量（杆量、盘量）与因变量(着陆 G 值)进行**线性关系分析**，判断数据项之间有无线性关系。再使用**神经网络模型**，对数据进行拟合检验，从**非线性角度**去分析数据项之间的关系。得到的结果是，杆量和盘量对于着陆 G 值存在一定的线性关系，但是由于**拟合度**不高，故不宜使用线性回归模型，而通过神经网络模型分析，得到盘量对于 G 值的影响大于杆量的影响，可采用盘量和杆量作为飞行操作的量化标准量，通过观察 G 值的改变来判断飞行操纵的平稳程度。

对于问题三，要求研究分析不同超限的**基本特征**，本题采用**统计学分析**对数据进行筛查分析，研究不同超限的基本特征。得到的结果是 2 级超限频次对于 3 级超限、68 号机场发生超限警告的频率远远大于其它机场、在所有飞行阶段中着陆与起飞两个阶段发生概率性较高、对于某机场容易发生何种超限类型，在统计分析结果中没有明显体现。

关键词：随机森林模型；多元线性回归；神经网络模型; QAR 数据;

一、问题重述

飞行安全是民航运输业发展的基础。随着中国民航的快速发展，研究飞行安全问题变得越来越重要。严重的飞行事故不仅给航空公司带来巨大的经济损失，还对乘客的生命构成威胁。因此，需要聚焦飞行安全问题，强化航空安全研究，利用现有数据进行科学管理，提升从业人员素质，监测和预警风险，减少飞行事故的发生几率。

航空安全大数据主要包括快速存取记录器（QAR）数据，记录飞机飞行过程中的各项飞行参数，以及在飞行品质监控中超出设定限制值的数据。目前我国民航业内的研究主要集中在超限事件和非超限数据的研究与应用上。超限事件研究通过设置超限阈值并分析超限记录，以防范潜在飞行事故。然而，目前的分析往往缺乏对超限原因的深入分析，因为超限可能不仅是人为因素引发，还可能与特殊环境条件或飞机本身的因素相关。因此，仅依靠超限分析来管理飞行机组可能会导致误判。

为了提高飞行安全管理和飞行训练的效果，目前的趋势是通过全航段数据进行分析，并建立特定人员的飞行品质记录。这个方法通过对不同飞行机组、飞行航线、机场和特定飞行条件下的数据进行建模和分析，计算和评估风险倾向性，开展有针对性的安全管理，排查安全隐患，改进安全绩效。类似的研究主要依靠大规模读取飞行数据，并建立飞行品质服务平台，为风险评估和趋势分析提供数据支持。 此外，G 值是飞机 飞行过程中过载情况的直接反应，在着陆安全分析中，G 值通常是描述落地瞬间安全性的重要指标。着陆瞬间 G 值指的是飞机接地瞬间前 2 秒和后 5 秒数据的最大 G 值。通常用来评估着陆的安全性。

问题 1：有些 QAR 数据存在错误，需要对数据进行预处理，去伪存真， 以减少错误数据对研究分析带来的影响。请你们的队伍对附件 1 的数据质量开展可靠性研究，提取与飞行安全相关的部分关键数据项，并对其重要程度进行分析。

问题 2：飞机在从起飞到着陆的整个飞行过程中，通过一系列的飞行操纵确保飞行安全，这些操纵主要包括横滚操纵、俯仰操纵等。目前，国内航空公司通过超限监控飞行操纵动作，这种监控方法虽然能够快速分辨出飞机的状态偏差，但是只能告诉安全管理人员发生了什么，而不能立刻得出发生这种偏差的原因。为此，可以通过操纵杆的过程变化情况来分析产生这种偏差的原因。根据附件 1，请你们对飞行操纵进行合理量化描述。

问题 3：导致不同超限发生的原因各不相同，有时是特定机场容易出现特定的超限，有时是特定的天气容易出现特定的超限，有时是特定的飞行员容易出现特定的超限。请研究附件 2 的数据，对超限的不同情况进行分析，研究不同超限的基本特征，如分析飞机在哪些航线或者在哪些机场容易出现何种超限等。

二、问题分析

2.1 问题一的分析

本题要求对附件 1 的数据质量开展可靠性研究，提取与飞行安全相关的部分关键数据项，并对其重要程度进行分析。对附件 1 的数据质量进行可靠性研究，需要进行数据预处理，去除错误数据以减少对研究分析的影响。数据预处理操作在本题主要包括数据去重、缺失值处理、异常值处理。在数据质量可靠性研究的基础上，可以提取与分析安全相关的关键数据项，并对其重要程度进行分析。本题背景给出 G 值通常是描述落地瞬间安全性的重要指标。因此将影响 G 值的部分数据项设为关键数据项，对其进行重要程度评估。

本题采用随机森林模型，对各项关键数据项进行重要性评分，并排序，以达到题目要求的重要程度分析。

2.2 问题二的分析

本题要求请你们对飞行操纵进行合理量化描述。飞机操纵是由操纵杆来控制的，杆位的变化与 G 值相关，由问题 2 中的杆位随时间变化曲线可知 G 值的变动进一步说明了杆位的变动进而导致飞机操纵的改变，故对飞机操纵进行合理化描述需要找到导致 G 值发生改变的原因，分析附件 1 的数据可得出盘量（飞机角度的变化量）与横滚操纵相关，杆量（操纵杆的位移量）与俯仰操纵相关，且盘量和杆量都是由飞行员人为控制的进而说明了操纵杆的变化是与盘量和杆量变化相关，故将 G 值的变化和盘量和杆量的变化进行可视化数据分析，找出当 G 值发生改变时盘量和杆量的变化情况，用标准回归分析考虑是否存在线性关系，若不存在线性关系则利用 BP 神经网络模型对数据进行拟合检验分析其结果是否与实际数据一致。

2.3 问题三的分析

本题要求请研究附件 2 的数据，对超限的不同情况进行分析，研究不同超限的基本特征，如分析飞机在哪些航线或者在哪些机场容易出现何种超限等。由附件 2 数据可知产生超限的原因各不相同，为了分析超限发生的原因首先对附件 2 中的数据进行筛选预处理，对筛选后的数据按照超限名称进行数理分析绘制出频数频率表，根据超限不同级别分为 2 级超限和 3 级超限，依据频数频率表分析出在不同超限名称的情况下 ARN（机号）、ARR（目的机场）、DDATE（时间日期、DEP（起飞机场）、PHASE（飞行阶段）、月份排名前三的信息进行数理统计分析，从而量化超限事件，研究出不同超限的基本特征。

三、基本假设与符号说明

3.1 基本假设

1、假设数据项之间相互独立，即每个数据项不受其他数据项的影响。

2、假设收集到的数据是准确、完整和可靠的，并且遵循了相关的数据收集准则和标准。

3、假设数据量较小，以及文本类数据项不存在缺失值和异常值，即数据是准确，完整、可靠的。

4、假设采用随机抽样调查能体现实验要求。

3.2 符号说明

符号	说明
X	自变量
Y	因变量
IQR	四分位距指标
VIF	方差膨胀因子
n	样本数量
WLS	加权最小二乘法
GI	基尼指数
$\hat{p}_m$	概率估计值
VIM	评价因子
E	评估指标

四、数据预处理

4.1 描述性统计

数据名称	统计量	最大值	最小值	均值	标准差	方差	变异系数
海拔高度	38672	119	36027	30966.03	7896.652	62357108	0.25501015
下降率	38672	-3620	2465	-0.54	439.82	193441.6	814.45148
无线电高度	38672	-2042	2043	532.62	1138.3	1295728	2.13717096
计算空速	38672	30	324.5	283.8762	50.402056	2540.367	0.17754943
地速	38672	0	503.5	422.2657	89.82965	8069.367	0.21273252
着陆 G 值	38672	0.8047	1.1992	0.999065	0.016243	0.00	0.01625880
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
下滑道偏向（L）	38672	-5.4	5.02	-0.0607	0.87871	0.772	14.47627677
下滑道偏向（R）	38672	-3.94	4.96	-0.0565	0.89348	0.798	15.81380531
航道偏向(C)	38672	-5.16	5.16	-0.1873		2.524	8.482808329
航道偏向（L）	38672	-5.15	5.16	-0.1888	1.58799	2.522	8.410963983
航道偏向（R）	38672	-5.16	5.16	-0.1779	1.58769	2.521	8.924620573
俯仰角率	38672	-1.4375	2.8125	-0.02784	0.0802005	0.006	2.881282558
飞机重量	38672	256480	347680	298019	26593.326	0.000007	0.089233672

4.2 缺失值处理

通过使用 Excel 对附件 1 中的数据进行统计筛查，对缺失值进行统计处理。经过处理得到有且仅有一项数据（GEAR SELECT DOWN LGDN 起落架）项含有缺失值，且缺失率较大，故将该数据剔除，不用于问题研究分析。

设缺失率为 p，缺失值个数为 n，数据总数为 m，则有缺失值公式：

五、问题一的模型建立与求解

5.1 模型的建立

本题要求对附件一的数据质量开展可靠性研究，并提取与飞行安全相关的部分关键数据项，并对其重要程度进行分析，故本题运用 IQR（四分段距指标）对数据异常值进行处理，最后通过相关理论以及附件 2 对关键数据项进行提取，再采用随机森林模型进行特征重要性评分，反映重要程度。

5.1.1 异常值处理

本题采用 IQR（四分位距指标）对异常值进行处理，通过计算附 1 中数据的 IQR 值，来评估数据的离散程度。确定好离群值范围，将数据中超过范围的数据标记为离群值，即异常值。箱线图为 IQR 的可视化方法。

IQR 公式：

$$IQR = Q_3 - Q_1$$

$$[Q_1 - k \times IQR, Q_3 + k \times IQR]$$

$$[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$$

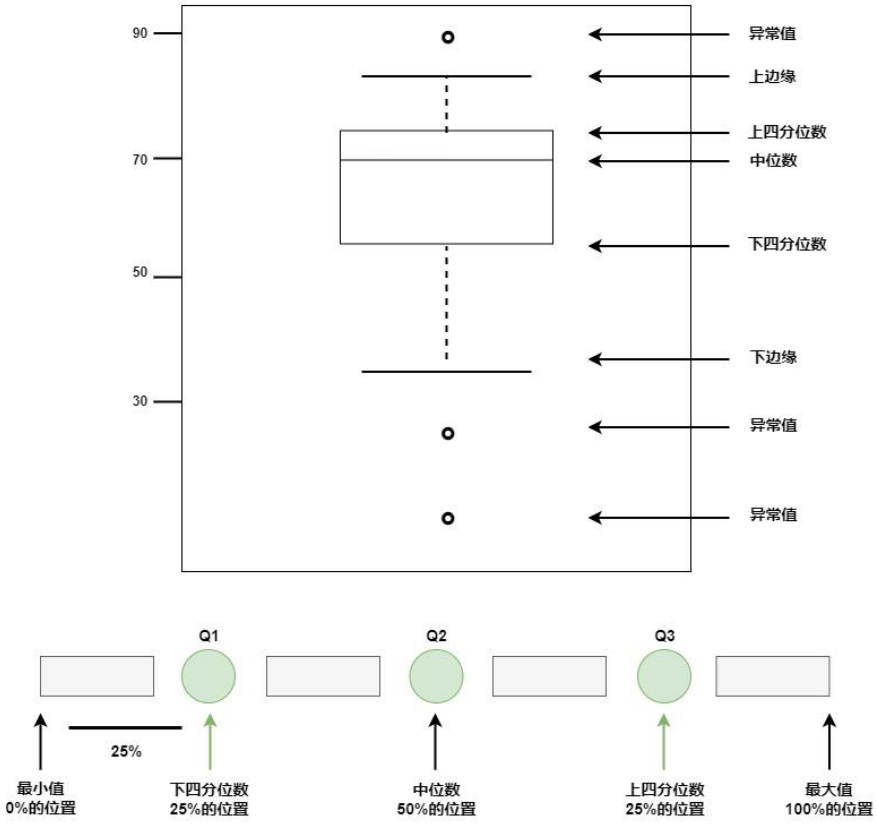


图 1 IQR 原理图

经过数据异常值处理并进行数据量统计，得到相关异常率与异常值数量统计表。

表一 异常类与异常值数量表

数据名称	异常率（%）	异常值数据量
海拔高度	10.030513	3879
下降率	10.635602	4113
无线电高度	0.000000	0
计算空速	9.208213	3561
地速	8.944456	3459
着陆 G 值	8.166115	3158
…	…	…
下滑道偏差（R）	21.118639	8167
航道偏差（C）	23.174390	8962
航道偏差（L）	23.306268	9013
航道偏差（R）	22.970108	8883
俯仰角率	4.543339	1757
飞机重量	0.000000	0

将所有异常值筛选并剔除。

5.1.2 关键项数据提取

G 值是飞机 飞行过程中过载情况的直接反应，在着陆安全分析中，G 值通常是描述落地瞬间安全性的重要指标。故将着陆 G 值设为因变量，影响着陆的数据项设为自变量。根据相关文献提示以及附件 1-2 的说明，得到以下数据变量表格。

表二 变量说明表

变量名	变量名称
因变量	着陆 G 值
自变量	海拔高度
	下降率
	计算空速
	地速
	下滑道偏差
	航向道偏差
	俯仰角率
	杆量
	盘量
	磁航向
	风向
	风速
	RUUD 位置

5.2 模型的求解

5.2.1 随机森林模型

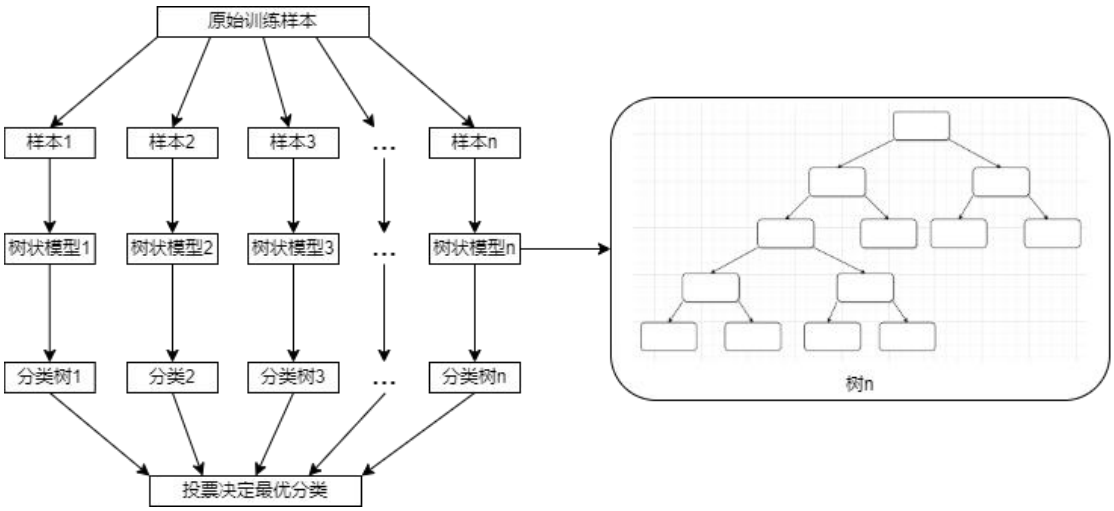


图 2 随机森林模型图

随机森林模型最终得到的结果取决于具体的应用场景和任务类型。本题中，随机森林模型可以提供特征重要性，随机森林模型可以计算每个特征对最终预测结果的贡献 程度，从而得到特征的重要性排名。

5.2.2 特征重要性评分

基尼指数：

基于基尼指数的变量重要性评分方法适用于分类问题，并且可以帮助我们理解每个特征对于预测结果的重要性程度。较高的变量重要性评分表示该特征对于预测结果的贡献更大。

Gini 指数的计算公式为：

$$GI_{\mathrm{m}} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$$

$\hat{p}_m$ 为样本在节点 m 属于任意一类的概率估计值

变量 X_j 在节点 m 的重要性，即节点 m 分枝前后 Gini 指数变化量为:

$$VIM_{jm}^{(Gini)} = GI_m - GI_l - GI_r$$

GI_l 和 GI_r 分别表示由节点 m 分裂的两新节点的 Gini 指数

$$VIM_{ij}^{(Gini)} = \sum_{m=1}^{M} VIM_{jm}^{(Gini)}$$

变量 X_j 在第 i 棵树出现 M 次,则变量 X_j 在第 i 棵树的重要性为：

$$VIM_{ij}^{(Gini)} = \frac{1}{n}\sum_{i=1}^{n} VIM_{ij}^{(Gini)}$$

变量 X_j 在 RF 中 Gini 重要性定义为：

$$VIM_j^{(Gini)} = \frac{1}{n}\sum_{i=1}^{n} VIM_{ij}^{(Gini)}$$

其中，n 为 RF 中分类数的数量

表三 特征值重要性评分表

排序	特征	重要性
1	下降率	0.072805
2	航道偏向	0.053915
3	下滑道偏向	0.051228
4	风速	0.049385
5	计算空速	0.044882
6	地速	0.039259
7	风向	0.039259
8	RUDD 位置	0.036795
9	海拔高度	0.035709
10	磁航向	0.035639
11	盘量	0.023351
12	俯仰角率	0.014630
13	杆量	0.000000

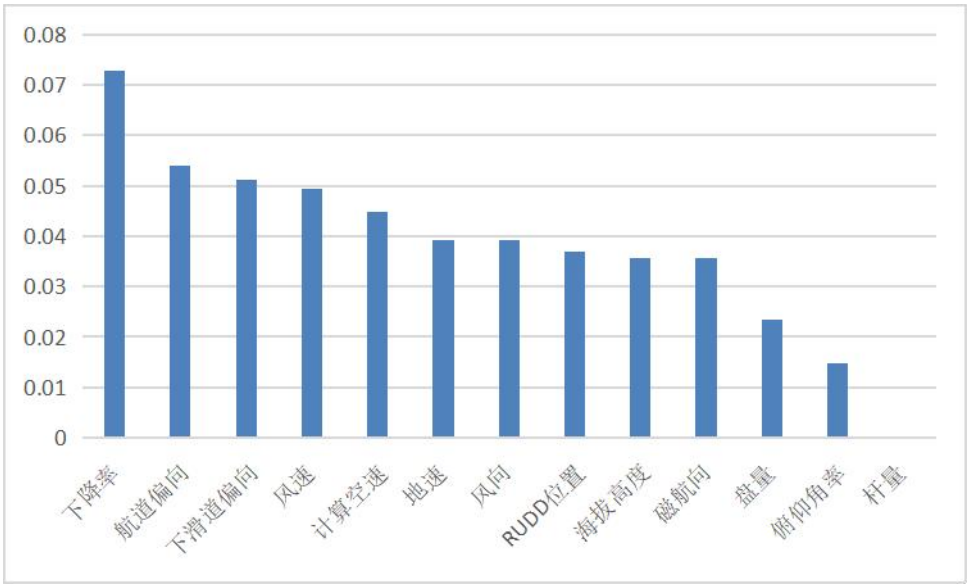


图 3 特征值重要性评分图

5.2.3 设定阈值

K 折验证交叉法：

在随机森林模型中，通过调整阈值可以改变模型预测结果的灵敏度和特异度。步骤如下：

（1）将数据集划分为训练集和测试集。

（2）对训练集进行 5 折交叉验证，将数据集划分为 5 个子集。

（3）对于每个阈值，循环进行以下步骤：

a、对于每个子集，训练随机森林模型并进行预测。

b、根据阈值将预测结果转化为分类标签。

c、计算每个子集的评估指标。

d、计算 5 个子集的平均交叉验证得分。

（4）选择得分最小的阈值作为最佳阈值。

需要注意的是，选择最佳阈值时，可以根据具体问题和需求来选择不同的评估指标

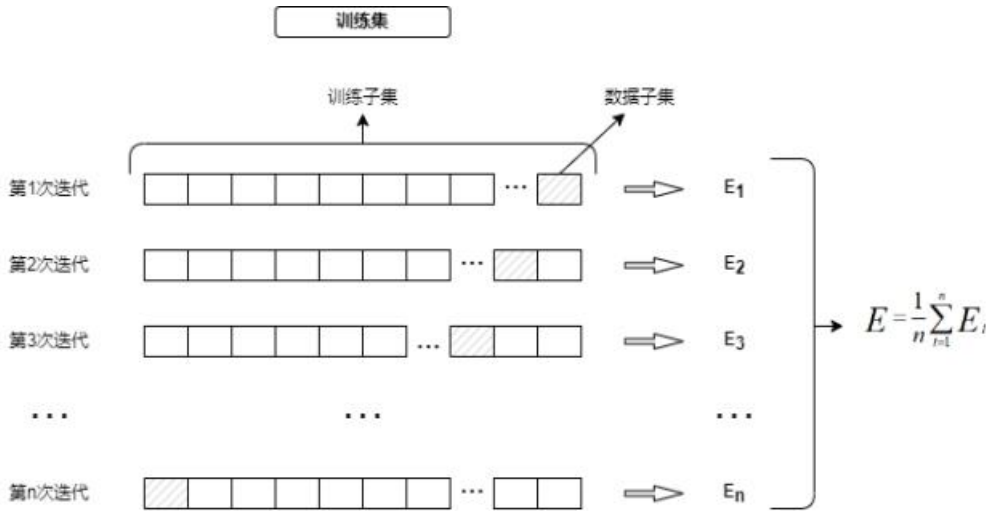


图 4　K 折验证交叉模型图

使用了 KFold 进行 5 折交叉验证，使用随机森林模型拟合数据，并计算每个阈值下的平均交叉验证得分。最后，找到得分最小的阈值，这被认为是最佳阈值。

最终所得最佳阈值：0.05，下降率、航道偏向、下滑道偏向都大于该阈值。

5.3 模型的结果

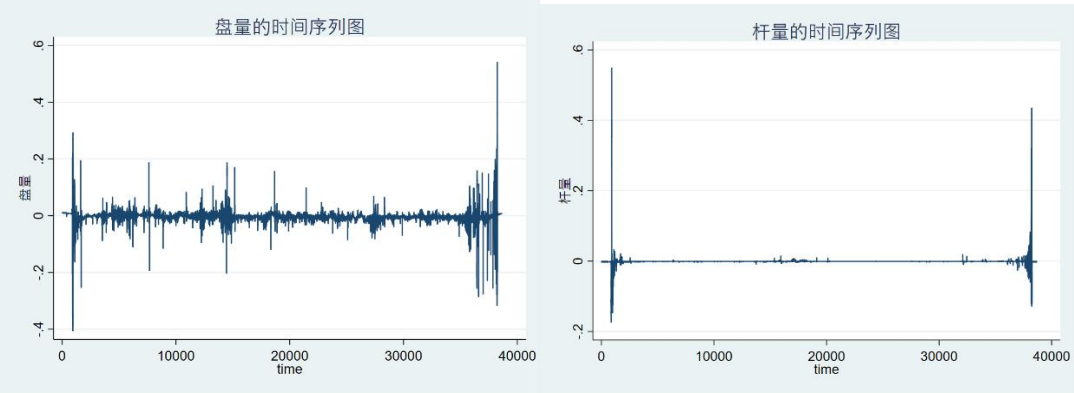
由特征重要性分析表可知，下降率、航道偏向、下滑道偏向对于飞行安全有良好的重要程度，风速、计算空速、地速、风向、RDUD 位置、海拔高度、磁航向对于飞行安全有一般的重要程度，俯仰角率、盘量、杆量对于飞行安全有较差的重要程度。

六、问题二的模型建立与求解

6.1 模型的建立

飞机操纵是由操纵杆来控制的，杆位的变化与 G 值相关，本题提取杆量与盘量作为自变量，设着陆 G 值为因变量。先进行数据可视化分析，在通过线性回归分析，分析它们之间是否存在线性关系，若存在线性，从而得到结果。若不存在线性关系，则利用 BP 神经网络模型对数据进行训练测试。

下图是分别对盘量、杆量以及 G 值对于时间所产生的变化。



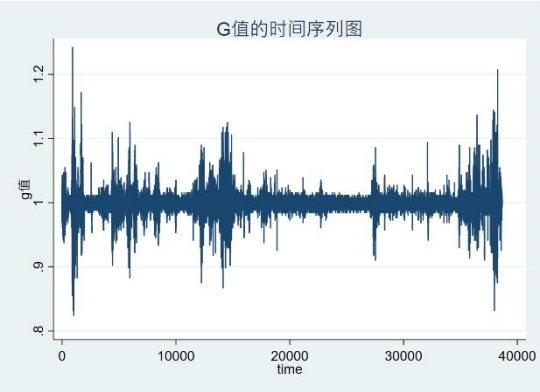


图 5 盘量、杆量和 G 值的时间序列图

由图可知当盘量与杆量发生明显变化时，G 值也相应产生变化。

6.2 模型的求解

6.2.1 线性关系检测

假设 G 值与杆量和盘量之间存在线性关系后对其进行标准化回归检验

标准化回归：

首先对自变量之间进行 vif 检验确保自变量之间不存在多重共线性

VIF 用于判断自变量之间的相关性，特别是检测是否有相关性很高的变量导致模型的结果不稳定，对于线性回归模型中的每个自变量，VIF 可以通过计算该自变量与其他所有自变量之间的关系的平方来获得。VIF 越大，表示自变量与其他自变量之间的共线性越强，可能存在多重共线性问题。通常情况下，如果某个自变量的 VIF 大于一个阈值（通常为 5 或 10），则可以认为该自变量与其他自变量存在高度相关性，可能会影响模型的稳定性和可解释性。这时可以考虑采取一些处理方法，如删除冗余的自变量、合并相关的自变量或者进行正交化等，以解决多重共线性问题。

$$VIF_i = \frac{1}{1 - R^2_{1-k/i}}$$

$R^2_{1-k/i}$ 是将第 i 个自变量作为因变量，剩下的 $k-1$ 个自变量回归得到的拟合

度。

$$VIF = \max\{VIF_i\}$$

若 VIF ＜ 10 则不存在严重的多重共线问题，反之则存在。

表 4 VIF 检验结果

变量名称	VIF	1/VIF
杆量	1.00	0.999
盘量	**1.00**	**0.999**
最大值	**1.00**	/

所含变量中，所有变量的 VIF 值都小于 10，拒绝原假设，认为自变量杆量和盘量之间不存在多重共线性

White 检验：

White 检验（White test）是一种针对线性回归模型中异方差性（heteroscedasticity）的检验方法，它是基于普通最小二乘法残差的方差与预测变量的关系。通过进行 White 检验，可以获得关于残差的异方差性信息，从而评估线性回归模型的合理性并采取相应的修正措施。

表 5 White 检验结果

源	卡方统计量	自由度	P 值
异方差性	382.29	5	0.0000
偏度	5.58	2	0.0000
峰度	44.31	1	0.0000
总体	432.17	8	0.0000

表中 p 值=0.000，强烈拒绝同方差的原假设，故存在异方差，利用加权最小二乘法（WLS）进行处理。

WLS 加权最小二乘法：

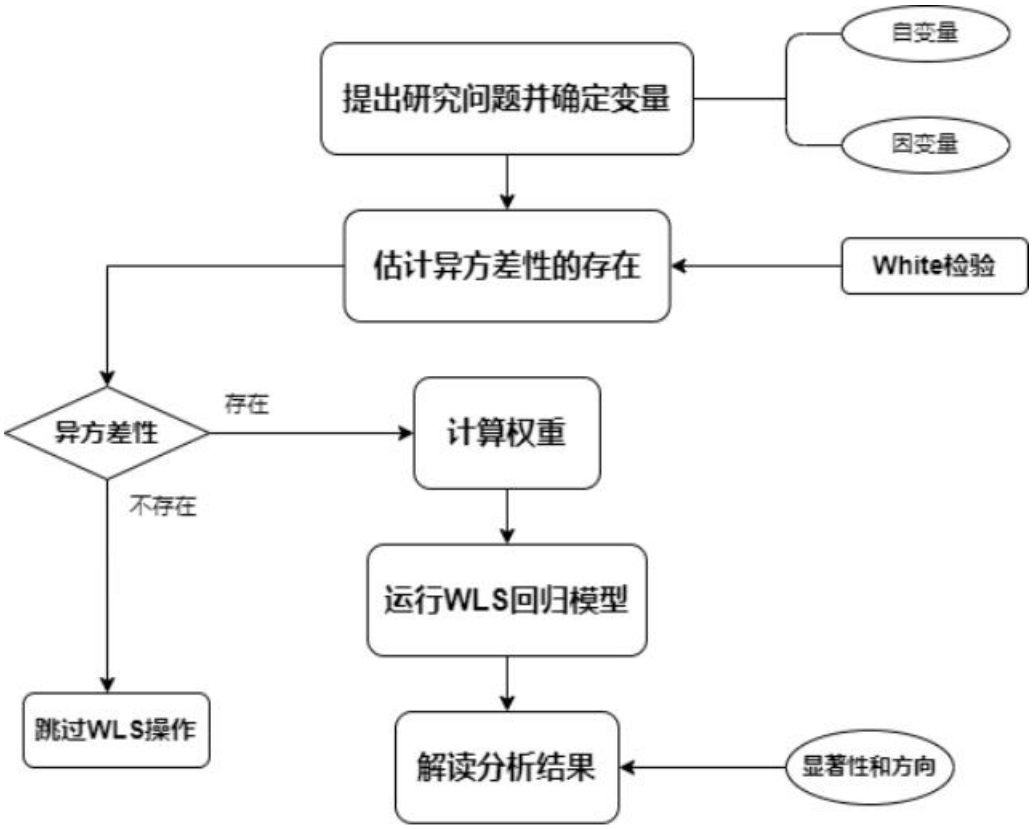


图 6 异方差检验流程图

WLS 方法通过对观测值进行加权来解决异方差性的问题。具体而言，WLS 赋予误差项更大的权重，使得方差较小的观测值在估计过程中有更大的影响力，而方差较大的观测值则有较小的影响力。这样可以减小方差较大观测值对估计结果的扰动。WLS 方法可以提高回归模型的效果，并更准确地估计参数，常被用于处理异方差性问题。

表 6 经 WLS 后方差分析表（二）

数据量	F 值	P 值	R^2 值	均方根误差	判定系数
5000	129.99	0.000	0.04951	0.0491	0.2368

表 7 经 WLS 回归后结果

评价量	系数	标准误差	T 值	P 值	Beta 值
杆量	0.257	0.159	16.9	0.000	0.222
盘量	0.009	0.013	0.72	0.473	0.009
常数量	0.998	0.003	2691.85	0.000	/

自变量的 P 值均小于 0.05，但是模型的拟合度为 0.0491 远小于 0.9，故拒绝原假设认为 G 值与杆量和盘量之间存在非线性关系，故对其进行 BP 神经网络模型的建立

6.2.2 BP 神经网络

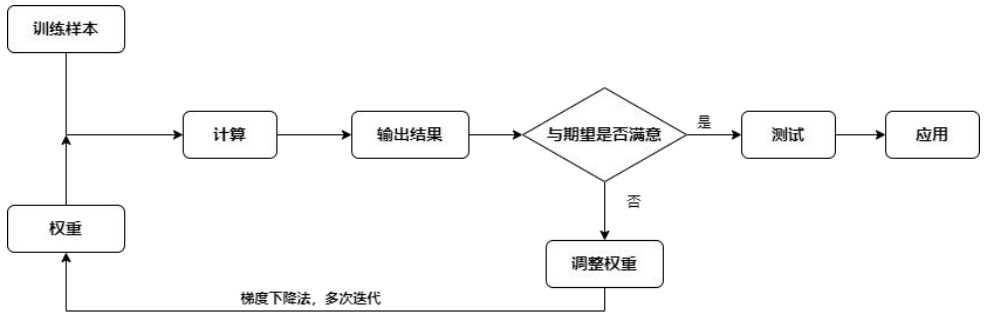


图 7 BP 神经网络流程图

BP 神经网络由输入层、隐藏层和输出层构成。每一层由多个神经元（或称为节点）组成，神经元之间以权重连接。输入层接收外部输入数据，输出层产生最终的预测结果，隐藏层在中间进行信息传递和处理。每层神经元之间实现全连接。BP 神经网络是一种基于误差反向传播算法的前馈神经网络模型，具有广泛的应用领域和能力。

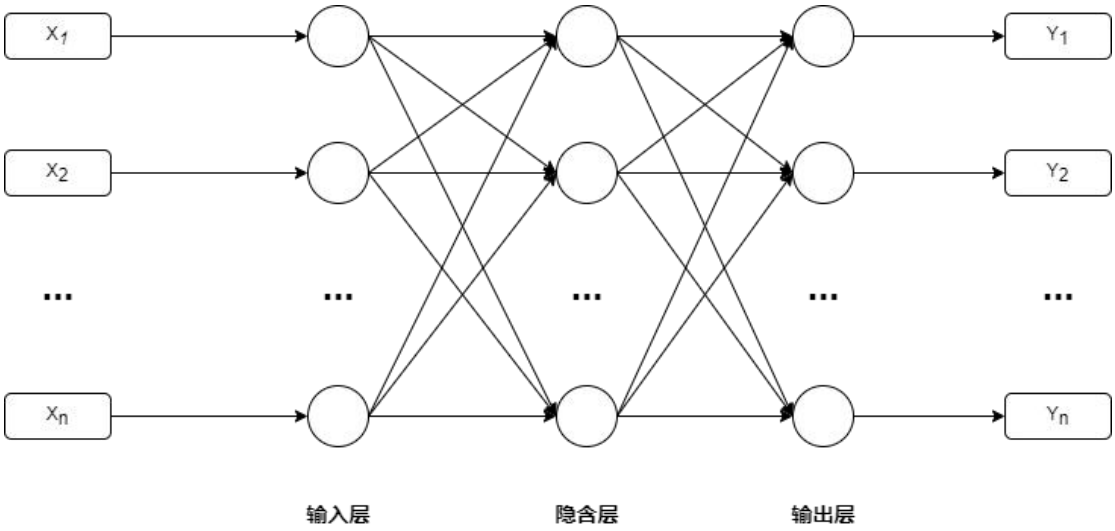


图 8 BP 神经网络原理图

本题设置自变量为杆量与盘量，因变量为 G 值，设置训练集和测试集。将训练好的模型对测试集进行预测，最终将预测结果与真实值进行比较。本题所训练的神经网络包含一个输入层，一个隐藏层和一个输出层。训练过程中使用的优化算法是随机梯度下降（SGD），学习率由参数 lr 确定。训练指定的 epochs 轮次，每轮次中进行一次前向传播和一次反向传播来更新权重和偏置。训练过程中会计算损失函数（均方误差）并打印出来，同时保存每个时期的损失值。predict 方法用于对新的输入数据进行预测，返回预测结果。

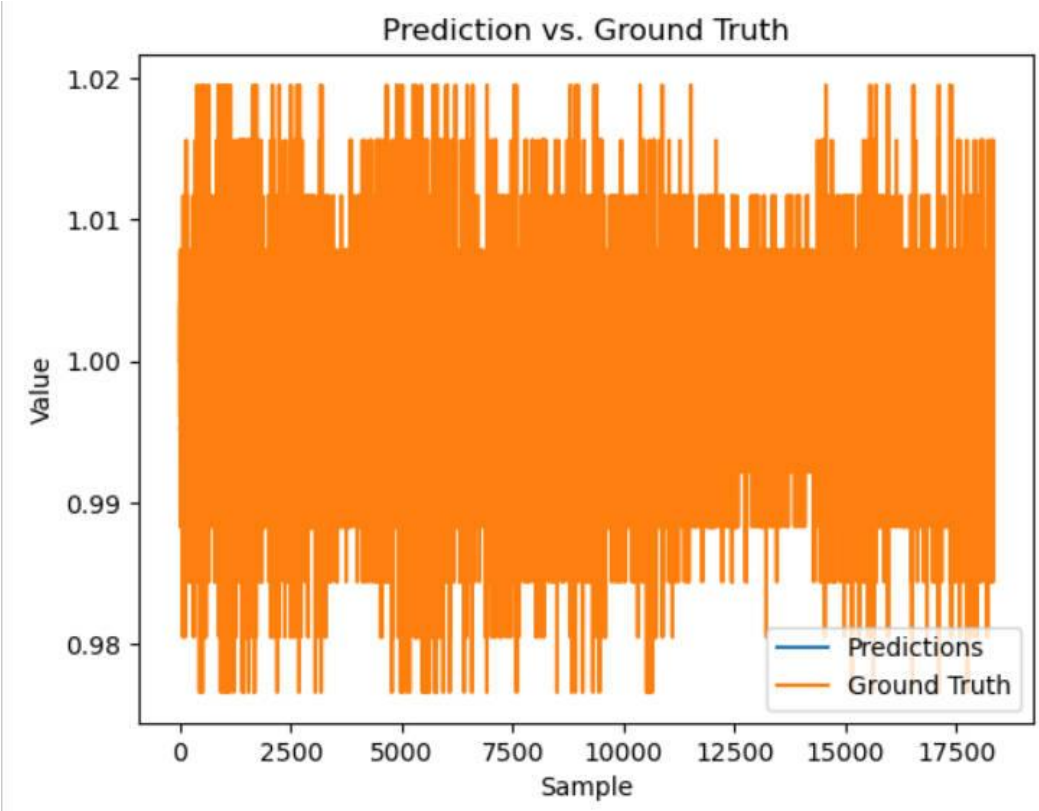


图 9　预测值与真实值对比图

Predictions 为预测值，Ground truth 为真实值，由图可以得出，真实值将预测值完整覆盖，即所得预测值在真实值范围之内，该模型的预测能力良好，满意度较高。

将所得结果绘制出损失值随着时间的变化曲线，当训练神经网络时，损失值（训练误差）应该随着时间的推移逐。渐减小，那么这是符合预期的行为，说明模型正在有效地学习。

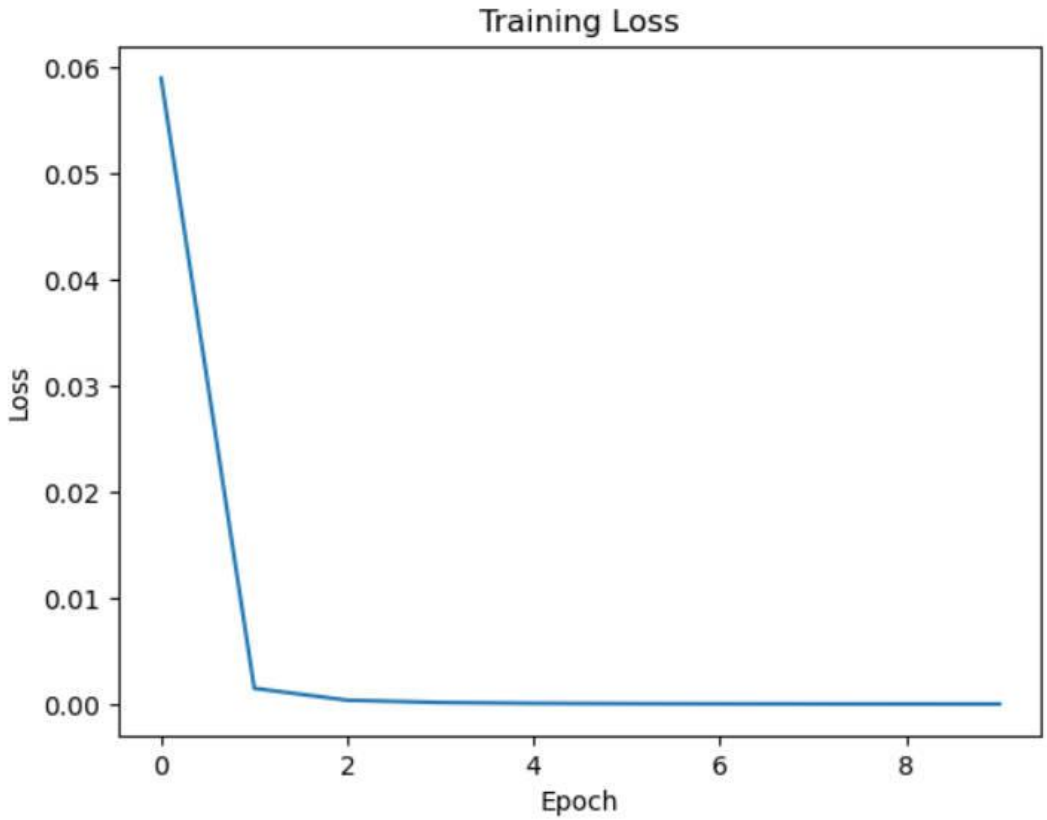


图 10 损失值随迭代变化图

图中 loss（损失值）随着时间的推移逐渐减小且下降趋势明显最终达到 0. 当训练神经网络时，损失值（训练误差）应该随着时间的推移逐渐减小。这是因为神经网络在每个训练周期中通过反向传播和权重更新来逐步调整模型以更好地拟合数据。故认为该模型训练程度很好，能够用于预测。

经过神经网络模型，计算所得出了隐藏层连接权重和输出层连接权重。

表八 权重值表

		权重值			
隐藏层	第一层	0.50141266	1.27371161	1.51221396	-1.44189848
	第二层	-1.08996276	0.77610609	-0.0274433	0.52930855
输出层		2.01286101	1.53618141	1.00325746	1.49395787

隐藏层连接权重：表示隐藏层有两个神经元，每个神经元对应一行权重。权重值反映了输入特征与隐藏层神经元之间的连接强度。通过观察权重值的大小，可以初步了解隐藏层神经元对应的特征对分类或预测的重要性。

输出层连接权重：这表示输出层有一个神经元，每个隐藏层神经元与该输出神经元的连接强度由权重值表示。权重值的大小决定了隐藏层神经元对最终输出结果的贡献程度。因此，权重值越大，表示相应的隐藏层神经元对最终输出结果的影响越大。

6.3 模型的结果

通过分析连接权重，了解到盘量和杆量与 G 值之间的的非线性关系，盘量对 G 值的影响较大，杆量对 G 值的影响相对较小，两个变量对 G 值都存在着正向影响且变化的趋势一致当盘量和杆量分别取极值时 G 值同样在此时发生巨大的变动（同样取极值），当 G 值发生巨大改变时操纵杆与之对应发生改变，因此最终确立的量化描述方法为：

采用盘量和杆量作为飞行操作量化的表征量，通过观察 G 值的改变来判断飞机操纵的平稳程度，当 G 值在短时间内发生大幅度上升或下降飞机操纵不平稳。

七、问题三的模型建立与求解

7.1 模型的建立

本题基于附件 2 的数据，对飞机在哪些航线或者在哪些机场容易出现何种超限，对附件 2 中的总数据，进行统计统计学分析，对数据进行筛查，建表筛查分析。

下图为所有超限事件作频数统计处理。

表九 总频数频率表

EVENT_NAME(超限名称)	频数	频率
50 英尺至接地距离远	16692	0.378477655
爬升速度大 35-1000ft	10748	0.243702242
接地速度小	9399	0.213114754
下降率大 400-50ft	1661	0.037661837
进近坡度大 50ft 以下	1553	0.035213024
进近速度大 50 ft 以下	764	0.017323085
下降率大 2000-1000（含）ft	577	0.01308301
着陆俯仰角小	401	0.009092352
低于下滑道	307	0.006960978
着陆垂直载荷大	281	0.006371449
下降率大 1000-400(含)ft	196	0.004444142
转弯滑行速度大	171	0.003877287
收反推晚	155	0.0035145
离地速度大	144	0.003265084
起飞坡度大 0-35（含）ft	131	0.002970319
高于下滑道	122	0.002766252
抬前轮速度大	105	0.00238079
空中过载	95	0.002154048
进近速度大 500-50（含）ft	76	0.001723239
爬升速度小 35-1000ft	69	0.001564519
进近坡度大 200-50（含）ft	58	0.001315103
未知	52	0.001179058
GPWS 警告(sink rate)	48	0.001088361
下滑道警告	43	0.00097499
抬前轮速度小	42	0.000952316
着陆襟翼到位晚	33	0.000748248

低空大速度 2500ft 以下	30	0.000680226
直线滑行速度大	27	0.000612203
抬头速率小	19	0.00043081
着陆俯仰角大	17	0.000385461
坡度大 400/1500ft 以上	12	0.00027209
GPWS 警告(windshear)	11	0.000249416
High normal accel with flap (in flight)	7	0.000158719
放起落架晚	7	0.000158719
进近坡度大 500-200（含）ft	6	0.000136045
着陆重、跳着陆	6	0.000136045
离地仰角大	6	0.000136045
50 英尺至接地距离近	5	0.000113371
起飞 EGT 超限	5	0.000113371
复飞	5	0.000113371
进近坡度大 1500-500（含）ft	3	6.80226E-05
超襟翼限制速度 Vfe	3	6.80226E-05
离地速度小	3	6.80226E-05
起飞滑跑方向不稳定	2	4.53484E-05
抬头速率大	1	2.26742E-05
起飞收起落架晚	1	2.26742E-05
Left of centreline below 1000ft	1	2.26742E-05
TCAS RA 警告	1	2.26742E-05
爬升坡度大 35-400(含)ft	1	2.26742E-05
进近速度小 500 ft 以下	1	2.26742E-05

在概率统计学中一般将频率低于 0.05 的事件视为小概率事件，本题将频率低于 0.01 的事件视为小概率事件（突发事件），不对其进行探讨，故对分析的数据进行筛选，结果如表 2 所示

表十 可分析数据表

EVENT_NAME(超限名称)	频数	频率
50 英尺至接地距离远	16692	0.378477655
爬升速度大 35-1000ft	10748	0.243702242
接地速度小	9399	0.213114754
下降率大 400-50ft	1661	0.037661837
进近坡度大 50ft 以下	1553	0.035213024
进近速度大 50 ft 以下	764	0.017323085
下降率大 2000-1000（含）ft	577	0.01308301

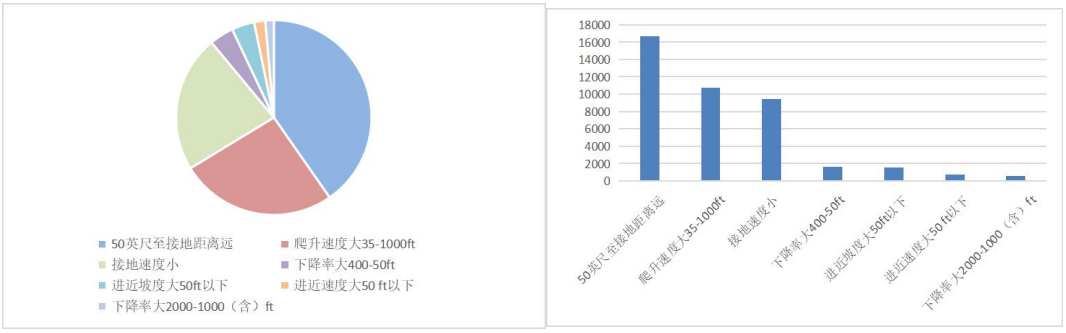


图 11 可分性数据直观图

可分析数据表中中 50 英尺至接地距离远、爬升速度大 35-1000ft、接地速度小超限事件发生的概率大，其中 50 英尺至接地距离远发生的频数远高于下降率大 2000-1000（含）ft 的情况，初步认为频数前三的超限名称数据是导致超限事件发生的主要原因，故对不同级别的超限事件分别做同表 2 一致的数理分析表

7.2 模型的求解

7.2.1 分类统计

表十一 警告级别 2 级超限数据

EVENT_NAME(超限名称)	频数	频率
50 英尺至接地距离远	14571	0.374104598
接地速度小	9205	0.236334694
爬升速度大 35-1000ft	8854	0.227322909
下降率大 400-50ft	1628	0.041798249
进近坡度大 50ft 以下	1500	0.0385119
进近速度大 50 ft 以下	707	0.018151942
下降率大 2000-1000	487	0.01250353

表十二 警告级别 3 级超限数据

EVENT_NAME(超限名称)	频数	频率
50 英尺至接地距离远	2121	0.411525029
爬升速度大 35-1000ft	1894	0.367481568
接地速度小	194	0.037640667
收反推晚	155	0.030073729
空中过载	95	0.018432286
下降率大 2000-1000（含）ft	90	0.017462165
进近速度大 50 ft 以下	57	0.011059371
爬升速度小 35-1000ft	57	0.011059371
着陆俯仰角小	57	0.011059371
进近坡度大 50ft 以下	53	0.010283275

表 3、4 的数据可知超限名称出现的频率大于 0.01 的情况相比表 2 中的数据都多，但都包含表 2 中所出现的情况，为了进一步提高准确性故对表 2、3、4 中都出现的情况进行一一分析，分别分析在不同超限名称发生的情况下 ARN（机号）、ARR（目的机场）、DDATE（时间日期、DEP（起飞机场）、EVENT_NAME(超限名称)、PHASE（飞行阶段）、月份的相关信息

表十三 50 英尺至接地距离远表

分析元素	2 级超限	3 级超限
超限名称	50 英尺至接地距离远	50 英尺至接地距离远
机号	26 号机 0.036922655	26 号机 0.044790193
	16 号机 0.035138288	14 号机 0.041489863
	23 号机 0.030471484	105 号机 0.040075436
目的机场	机场 68 0.453503534	机场 68 0.437057992
	机场 89 0.055315352	机场 89 0.069778406
	机场 27 0.034863771	机场 27 0.058462989
起飞机场	机场 68 0.422757532	机场 68 0.424799623
	机场 89 0.042824789	机场 89 0.058934465
	机场 16 0.027040011	机场 16 0.028760019
飞行阶段	着陆 0.997803857	着陆 1
	进近 0.002058884	
	未知 0.000137259	
年份	2015 0.461670441	2015 0.471004243
	2016 0.538329559	2016 0.528995757

50 英尺至接地距离远：该事件 99.8%在着陆阶段发生，发生 2 级超限的主要机号是 26、16、23 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号 26、14、105 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生超限事件的主要飞机是 26 号，机场是 68 号其占比远远高于其他机场，2015 年发生概率相当 2016 年偏低。

表十四 爬升速度大 35-1000ft 表

分析元素	2 级超限	3 级超限
超限名称	爬升速度大 35-1000ft	爬升速度大 35-1000ft
机号	71 号机 0.027332279	18 号机 0.032206969
	18 号机 0.026428733	106 号机 0.029039071
	23 号机 0.025864016	13 号机 0.025871172
目的机场	机场 68 0.515021459	机场 68 0.503167899
	机场 16 0.022814547	机场 21 0.03062302
	机场 89 0.021911001	机场 16 0.025871172
起飞机场	机场 68 0.361644454	机场 68 0.368532207
	机场 16 0.064603569	机场 16 0.103484688
	机场 88 0.022136887	机场 67 0.027983105
飞行阶段	空中 1	空中 1

	2015	0.468714705	2015	0.496832101
年份	2016	0.531285295	2016	0.503167899

爬升速度大 35-1000ft：该事件 100%在空中发生，发生 2 级超限的主要机号是 71、18、23 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号是 18、106、13 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生超限事件的主要飞机是 18 号，机场是 68 号其占比远远高于其他机场，2015 年发生相当 2016 年偏低。

表十五 接地速度小表

分析元素	2 级超限		3 级超限	
超限名称	接地速度小		接地速度小	
	26 号机	0.039869636	26 号机	0.067010309
机号	16 号机	0.035415535	20 号机	0.056701031
	105 号机	0.034655079	21 号机	0.046391753
	机场 68	0.493644758	机场 68	0.592783505
目的机场	机场 89	0.02563824	机场 89	0.030927835
	机场 61	0.021510049	机场 61	0.025773196
	机场 68	0.405975014	机场 68	0.298969072
起飞机场	机场 89	0.041933732	机场 89	0.051546392
	机场 16	0.030635524	机场 16	0.046391753
飞行阶段	着陆	1	着陆	1
	2015	0.505703422	2015	0.525773196
年份	2016	0.494296578	2016	0.474226804

接地速度小：该事件 100%在着陆阶段发生，发生 2 级超限的主要机号是 26、16、105 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号是 26、20、21 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生超限事件的主要飞机是 26 号，机场是 68 号其占比远远高于其他机场，2015 年与 2016 年发生事件的概率相差不大其中 2016 年略偏低。

表十六 进近坡度大 50ft 以下表

分析元素	2 级超限		3 级超限	
超限名称	进近坡度大 50ft 以下		进近坡度大 50ft 以下	
	105 号机	0.036	87 号机	0.075471698
机号	50 号机	0.026666667	14 号机	0.056603774
	20 号机	0.025333333	49 号机	0.037735849
	机场 68	0.526666667	机场 68	0.509433962
目的机场	机场 16	0.074666667	机场 16	0.113207547
	机场 89	0.039333333	机场 89	0.094339623
	机场 68	0.348	机场 68	0.339622642
起飞机场	机场 89	0.048666667	机场 55	0.075471698
	机场 16	0.037333333	机场 40	0.037735849
	进近	0.923333333	进近	0.905660377

飞行阶段	着陆	0.053333333	着陆	0.056603774
	地面	0.023333333	地面	0.037735849
	2015	0.496666667	2015	0.396226415
年份	2016	0.503333333	2016	0.603773585

进近坡度大 50ft 以下：该事件 92.5%在进近阶段发生，发生 2 级超限的主要机号是 20、50、105 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号是 87、14、49 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生 2 级超限事件的主要飞机是 105 号，发生 3 级超额事件的飞机主要是 87 号，机场是 68 号其占比远远高于其他机场，2 级事件在 2015 年和 2016 年发生的概率相差不大，3 级事件在 2015 年发生的概率低于 2016 年的概率。

表十七 进近速度大 50 ft 以下表

分析元素	2 级超限		3 级超限	
超限名称	进近速度大 50 ft 以下		进近速度大 50 ft 以下	
	69 号机	0.028288543	69 号机	0.070175439
机号	87 号机	0.026874116	20 号机	0.052631579
	61 号机	0.026874116	74 号机	0.052631579
	机场 68	0.561527581	机场 68	0.614035088
目的机场	机场 16	0.087694484	机场 16	0.070175439
	机场 27	0.028288543	机场 99	0.052631579
	机场 68	0.333804809	机场 68	0.228070175
起飞机场	机场 89	0.043847242	机场 16	0.087719298
	机场 16	0.025459689	机场 89	0.052631579
	进近	0.991513437	进近	1
飞行阶段	未知	0.008486563		
	2015	0.465346535	2015	0.456140351
年份	2016	0.534653465	2016	0.543859649

进近速度大 50 ft 以下：该事件 99.6%在进近阶段发生，发生 2 级超限的主要机号是 69、87、61 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号是 69、20、74 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生超限事件的主要飞机是 69 号，机场是 68 号其占比远远高于其他机场，2015 年发生概率相当 2016 年偏低。

表十八 下降率大 2000-1000（含）ft

分析元素	2 级超限	3 级超限
超限名称	下降率大 2000-1000（含）ft	下降率大 2000-1000（含）ft
机号	14 号机 0.036960986	18 号机 0.077777778
	60 号机 0.036960986	16 号机 0.055555556
	17 号机 0.030800821	51 号机 0.055555556
目的机场	机场 68 0.353182752	机场 68 0.144444444
		机场 81 0.111111111

	机场 104	0.051334702	机场 38	0.077777778
	机场 64	0.028747433	机场 68	0.555555556
起飞机场	机场 68	0.515400411	机场 89	0.122222222
	机场 89	0.063655031	机场 16	0.066666667
	机场 40	0.026694045	进近	0.966666667
飞行阶段	进近	0.965092402	空中	0.033333333
	空中	0.034907598	2015	0.511111111
年份	2015	0.470225873	2016	0.488888889
	2016	0.529774127		

下降率大 2000-1000（含）ft：该事件 96.5%在进近阶段发生，发生 2 级超限的主要机号是 14、60、17 号，主要目的机场是 68 号，主要起飞机场是 68 号；发生 3 级超限的主要机号是 18、16、51 号，主要目的机场是 68 号，主要起飞机场是 68 号，会发生 3 级超额事件的飞机主要是 18 号，机场是 68 号其占比远远高于其他机场，2 级超限 2015 年发生概率相当 2016 年偏低，3 级超限 2015 年发生概率相当 2016 年偏高。

7.2.2 占比统计

根据分类统计所得数据，对机场事故率进行，计算统计占比。

表十九 机场超限警告频数表

机场号	超限警告频数
机场 68	20993
机场 89	1777
机场 16	1315
机场 27	1141
机场 61	603
…	…
机场 9	2
机场 96	2
机场 66	1
机场 13	1
机场 24	1

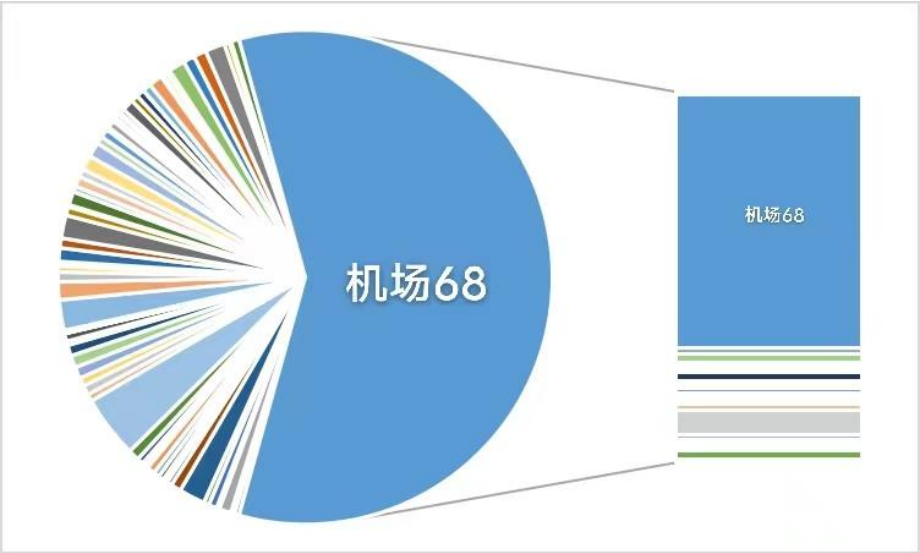


图 12 机场超限警告频数占比图

由图表可以知，在所有超限警告频数汇总中，频数排名第一的 68 号机场超限警告频数超 20000 次，而频数排名第二的 89 号机场，仅有 1777 次超限警告频数。相差值巨大。68 号机场的超限警告频数占总体的百分比数也超过 50%，远远大于其余机场。

表二十 飞行阶段超限警告频数表

飞行阶段	超限警告频数
着陆	26406
空中	11119
进近	5591
地面（着陆）	398
地面（起飞）	158
起飞	148
未知	231

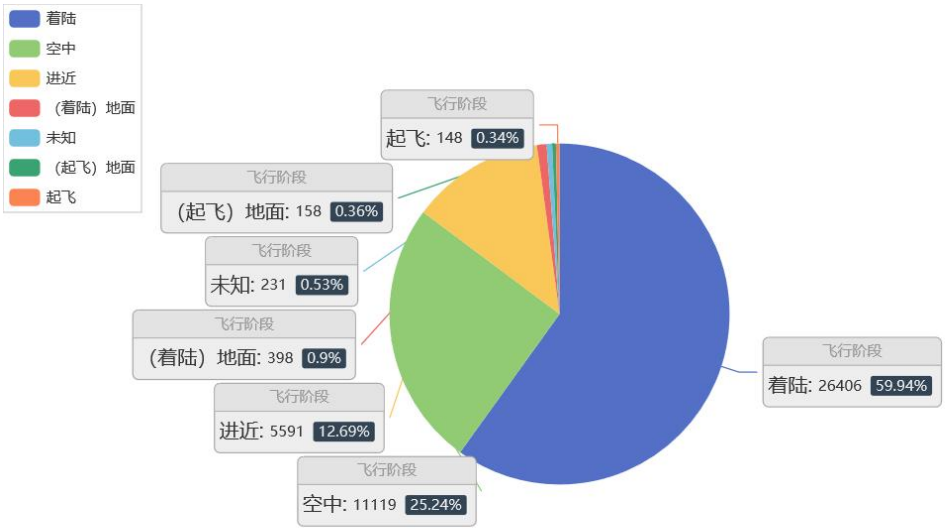


图 13 飞行阶段超限警告占比图

由图表可知，飞机在飞行阶段中，在着陆时的超限警告次数达到了 26106 次，在空中时的超限警告次数也达到了 11119 次，这两个阶段的超限警告频数是远大于其余几个阶段的。飞机在着陆阶段的超限警告频数占总超限警告频数的 59.94%，在空中阶段的超限警告频数占总超限警告频数的 25.24%，两者占比达到了 85.18%，远超其余飞行阶段。

表二十一 超限警告分类频数表

	2 级警告	3 级警告
超限警告频数	38949	5154

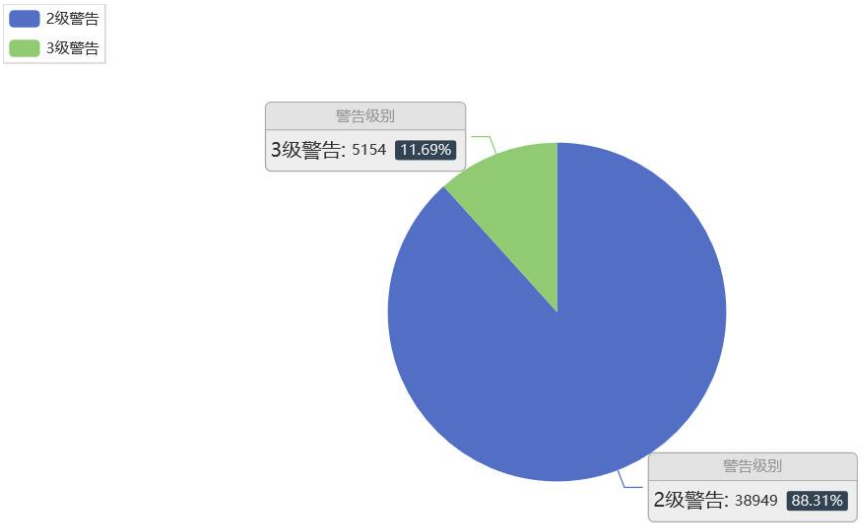


图 14 超限警告分类频数占比图

由图标可知在总超限警告频数统计中 2 级警告的次数达到了 38949 次，而 3

级警告的次数仅有 5154 次。2 级警告所占比到了 88.31%,远超 3 级警告的 11.69%占比。

7.3 模型结果

经过全局的统计分析处理后，可以得知，在飞机发生超限警告的情况中主要发生的是 2 级警告，相对于 3 级警告，2 级警告的发生频率是较高的。是否发生警告与飞机号没有明显的关联，对于机场有着较强的关联，在所有汇总统计的数据中，飞机在 68 号机场的超限警告频率是远远大于其余 100 号机场的，且在一般情况下是发生 2 级警告。对于在某机场容易发生何种超限类型，由于 68 号机场的超限警告率过于大，超限警告频数过于庞大，故统计结果中没有一个明显的体现。飞机在各飞行阶段与超限警告也有着较强的关联性，飞机一般在起飞与着陆过程中，较容易发生超限警告。

八、模型的评价与改进方向

九、参考文献

[1] 葛婷 胡占桥 QAR 飞行数据还原 不安全状态预测及影响因素分析（下）
[2] 余浪 易东 B-P 神经网络模型在缺血性心脏病住院费用影响因素分析中的应用

十、附录

10.1 模型代码

数据统计

```
import pandas as pd

# 读取 Excel 文件
df = pd.read_excel(r"C:\Users\30408\Desktop\附件 2 数据.xlsx",sheet_name="Sheet2")
# 获取指定列的数据
column_data = df['EVENT_NAME 超限名称']
# 统计每个类别的频数
category_counts = column_data.value_counts()
# 打印结果
print(category_counts)

import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings("ignore", category=UserWarning)
```

```
plt.rcParams['font.sans-serif'] = ['SimHei']
# 读取 Excel 文件的 Sheet2
df = pd.read_excel(r"C:\Users\30408\Desktop\附件 2 数据.xlsx",
sheet_name="Sheet2")

# 统计某一列的元素频数
column_counts = df["EVENT_NAME 超限名称"].value_counts()
# 创建柱状图
plt.bar(column_counts.index, column_counts.values)
# 设置图表标题和轴标签
plt.title("Frequency Count of Column")
plt.xlabel("Elements")
plt.ylabel("Frequency")
# 显示图表
plt.show()

import pandas as pd
# 读取 Excel 文件
df = pd.read_excel(r"C:\Users\30408\Desktop\附件 2 数
据.xlsx",sheet_name="Sheet3")
# 获取指定列的数据
column_data = df['DEP（起飞机场）']
# 统计每个类别的频数
category_counts = column_data.value_counts()
# 打印结果
print(category_counts)
```

数据处理

```
# coding=UTF-8
import pandas as pd
df = pd.read_excel(r"C:\Users\30408\Desktop\3 级.xlsx", sheet_name="爬
升速度大 35-1000ft")
column_data = df['月份']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数':
category_counts.values})
result_df['占比'] = result_df['频数'] * 100

# 指定要导入的 Excel 文件和 Sheet 名
excel_file = r"C:\Users\30408\Desktop\测试数据.xlsx"
sheet_name = 'Sheet3'

# 将 DataFrame 导入到 Excel 的指定 Sheet 中
```

```
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)
```

数据分析

```
# coding=UTF-8
import pandas as pd

df = pd.read_excel(r"C:\Users\30408\Desktop\2 级.xlsx", sheet_name="下降率大 2000-1000（含）ft")
column_data = df['ARN 机号']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数': category_counts.values})
result_df['占比'] = result_df['频数'] * 100
# 指定要导入的 Excel 文件和 Sheet 名
excel_file = r"C:\Users\30408\Desktop\2 级分析结果 7.xlsx"
sheet_name = 'Sheet6'
# 将 DataFrame 导入到 Excel 的指定 Sheet 中
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)


df = pd.read_excel(r"C:\Users\30408\Desktop\2 级.xlsx", sheet_name="下降率大 2000-1000（含）ft")
column_data = df['ARR 目的机场']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数': category_counts.values})
result_df['占比'] = result_df['频数'] * 100
# 指定要导入的 Excel 文件和 Sheet 名
excel_file = r"C:\Users\30408\Desktop\2 级分析结果 7.xlsx"
sheet_name = 'Sheet2'
# 将 DataFrame 导入到 Excel 的指定 Sheet 中
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)

df = pd.read_excel(r"C:\Users\30408\Desktop\2 级.xlsx", sheet_name="下降率大 2000-1000（含）ft")
column_data = df['DEP 起飞机场']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数': category_counts.values})
result_df['占比'] = result_df['频数'] * 100
# 指定要导入的 Excel 文件和 Sheet 名
```

```
excel_file = r"C:\Users\30408\Desktop\2级分析结果7.xlsx"
sheet_name = 'Sheet3'
# 将DataFrame导入到Excel的指定Sheet中
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)

df = pd.read_excel(r"C:\Users\30408\Desktop\2级.xlsx", sheet_name="下降率大2000-1000（含）ft")
column_data = df['PHASE_NAME_CN飞行阶段中文']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数': category_counts.values})
result_df['占比'] = result_df['频数'] * 100
# 指定要导入的Excel文件和Sheet名
excel_file = r"C:\Users\30408\Desktop\2级分析结果7.xlsx"
sheet_name = 'Sheet4'
# 将DataFrame导入到Excel的指定Sheet中
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)


df = pd.read_excel(r"C:\Users\30408\Desktop\2级.xlsx", sheet_name="下降率大2000-1000（含）ft")
column_data = df['月份']
category_counts = column_data.value_counts(normalize=True)
result_df = pd.DataFrame({'类别': category_counts.index, '频数': category_counts.values})
result_df['占比'] = result_df['频数'] * 100
# 指定要导入的Excel文件和Sheet名
excel_file = r"C:\Users\30408\Desktop\2级分析结果7.xlsx"
sheet_name = 'Sheet5'
# 将DataFrame导入到Excel的指定Sheet中
with pd.ExcelWriter(excel_file, mode='a') as writer:
    result_df.to_excel(writer, sheet_name=sheet_name, index=False)
```

数据可视化

```
# coding=UTF-8
import pyecharts.options as opts
from pyecharts.charts import Pie
from pyecharts.render import make_snapshot
from snapshot_selenium import snapshot as driver

# inner_x_data = ["26号机", "18号机", "105号机","16号机","23号机"]
# inner_y_data = [1419,1225,1211,1202,1194]
```

```
# inner_data_pair = [list(z) for z in zip(inner_x_data, inner_y_data)]

outer_x_data = ["2 级警告","3 级警告"]
outer_y_data = [38949,5154]
outer_data_pair = [list(z) for z in zip(outer_x_data, outer_y_data)]

(
    Pie()
    # .add(
    #     series_name="机号",
    #     data_pair=inner_data_pair,
    #     radius=[0, "35%"],
    #     label_opts=opts.LabelOpts(position="inner"),
    # )
    .add(
        series_name="警告级别",
        radius=["0%", "55%"],
        data_pair=outer_data_pair,
        label_opts=opts.LabelOpts(
            position="outside",
            formatter="{a|{a}}{abg|}\n{hr|}\n {b|{b}: }{c}  {per|{d}%}
",
            background_color="#eee",
            border_color="#aaa",
            border_width=1,
            border_radius=4,
            rich={
                "a": {"color": "#999", "lineHeight": 22, "align":
"center"},
                "abg": {
                    "backgroundColor": "#e3e3e3",
                    "width": "100%",
                    "align": "right",
                    "height": 22,
                    "borderRadius": [4, 4, 0, 0],
                },
                "hr": {
                    "borderColor": "#aaa",
                    "width": "100%",
                    "borderWidth": 0.5,
                    "height": 0,
                },
                "b": {"fontSize": 16, "lineHeight": 33},
                "per": {
```

```
                    "color": "#eee",
                    "backgroundColor": "#334455",
                    "padding": [2, 4],
                    "borderRadius": 2,
                },
            },
        ),
    )
    .set_global_opts(legend_opts=opts.LegendOpts(pos_left="left",
orient="vertical"))
    .set_series_opts(
        tooltip_opts=opts.TooltipOpts(
            trigger="item", formatter="{a} <br/>{b}: {c} ({d}%)"
        )
    )
    .render("扇形图.html")
)
```

问题一

用 IQR 处理异常值：

```
Import
pandas as pd
import matplotlib.pyplot as plt


# 读取表格数据

data = pd.read_excel(r'D:\桌面\副本筛选表.xlsx')

# 计算第一四分位数（Q1）和第三四分位数（Q3）

Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)


# 计算 IQR

IQR = Q3 - Q1


# 定义上边界和下边界

upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR
```

```
# 检测异常值
outliers = ((data > upper_bound) | (data < lower_bound))

# 计算每一列的异常率和异常值数量
outlier_rates = outliers.mean() * 100
outlier_counts = outliers.sum()

# 创建异常率和异常值数量表格
table = pd.DataFrame(
    {'Column': outlier_rates.index, 'Outlier Rate (%)':
outlier_rates.values, 'Outlier Count':
outlier_counts.values})

# 输出异常率和异常值数量表格

print("异常率和异常值数量统计：")
print(table)
print()

# 删除含有异常值的行并重置索引
filtered_data =
data[~outliers.any(axis=1)].reset_index(drop=True)
```

使用随机森林模型求特征重要性评分：

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor

# 读取数据

data = pd.read_excel(r'D:\桌面\纠正后的数据.xlsx')

# 分离特征和目标变量

X = data.drop('COG NORM ACCEL', axis=1)  # 替换为实际的目标变
量列名
y = data['COG NORM ACCEL']  # 替换为实际的目标变量列名
```

```
# 创建随机森林模型
rf = RandomForestRegressor()

# 拟合模型
rf.fit(X, y)

# 获取特征重要性
feature_importances = rf.feature_importances_

# 创建特征重要性表格
table = pd.DataFrame({'Feature': X.columns, 'Importance':
feature_importances})

# 按重要性降序排序
table = table.sort_values('Importance', ascending=False)

# 输出特征重要性表格

print("特征重要性评估结果：")
print(table)
```

交叉验证代码：

```
# 获取抽样后的特征矩阵 X 和目标向量 y

# 删除 'COG NORM ACCEL' 列并重置索引
X = sampled_data.drop('COG NORM ACCEL',
axis=1).reset_index(drop=True)
y = sampled_data['COG NORM ACCEL'].reset_index(drop=True)

# 获取原始特征名称
feature_names = X.columns

# 定义 K 折交叉验证
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

```
# 定义随机森林模型
rf = RandomForestRegressor(n_estimators=100,
random_state=42)

# 存储每个阈值对应的平均交叉验证得分
threshold_scores = {}

# 循环尝试不同的阈值
for threshold in np.linspace(0.05, 0.5, num=10):
    scores = []
    # 进行 K 折交叉验证
    for train_index, val_index in kfold.split(X):
        X_train, X_val = X.iloc[train_index],
X.iloc[val_index]
        y_train, y_val = y.iloc[train_index],
y.iloc[val_index]
        # 获取要选择的特征索引
        selected_feature_indices = [X.columns.get_loc(name)
for name in feature_names]

        # 定义特征选择器
        feature_selector = ColumnTransformer(
            transformers=[('selected_features',
'passthrough', selected_feature_indices)])

        # 构建训练流水线
        pipeline = Pipeline(steps=[('feature_selector',
feature_selector), ('rf', rf)])

        # 拟合随机森林模型
        pipeline.fit(X_train, y_train)

        # 使用选中的特征进行预测
        X_val_selected = X_val.iloc[:,
feature_selector.transformers_[0][2]]
```

```
        y_pred = pipeline.predict(X_val_selected)

        # 计算交叉验证得分
        score = np.mean((y_pred - y_val) ** 2)
        scores.append(score)

    # 计算平均交叉验证得分
    avg_score = np.mean(scores)

    # 存储到字典中
    threshold_scores[threshold] = avg_score

# 找到得分最小的阈值
best_threshold = min(threshold_scores,
key=threshold_scores.get)

print("最佳阈值:", best_threshold)
```

B - P 神经网络模型：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# 定义 Sigmoid 激活函数
def sigmoid(x):
    return 1 / (1 + np.exp(-x))


# 定义 B-P 神经网络类
class BPNeuralNetwork:
    def __init__(self, input_dim, hidden_dim, output_dim):
        self.input_dim = input_dim
        self.hidden_dim = hidden_dim
        self.output_dim = output_dim
```

```
        # 初始化权重和偏置
        self.W1 = np.random.randn(self.input_dim,
self.hidden_dim)
        self.b1 = np.zeros((1, self.hidden_dim))
        self.W2 = np.random.randn(self.hidden_dim,
self.output_dim)
        self.b2 = np.zeros((1, self.output_dim))
        self.loss_history = []  # 保存每个时期的损失值

    def forward(self, X):
        # 前向传播计算
        self.z1 = np.dot(X, self.W1) + self.b1
        self.a1 = sigmoid(self.z1)
        self.z2 = np.dot(self.a1, self.W2) + self.b2
        self.a2 = sigmoid(self.z2)
        return self.a2

    def backward(self, X, y, lr):
        m = X.shape[0]

        # 计算输出层的误差
        delta2 = self.a2 - y

        # 反向传播更新权重和偏置
        dW2 = np.dot(self.a1.T, delta2) / m
        db2 = np.sum(delta2, axis=0, keepdims=True) / m
        delta1 = np.dot(delta2, self.W2.T) * (self.a1 * (1 -
self.a1))
        dW1 = np.dot(X.T, delta1) / m
        db1 = np.sum(delta1, axis=0, keepdims=True) / m

        self.W2 -= lr * dW2
        self.b2 -= lr * db2
        self.W1 -= lr * dW1
        self.b1 -= lr * db1

    def train(self, X, y, epochs, lr):
        for epoch in range(epochs):
            # 前向传播
```

```
            output = self.forward(X)

            # 计算损失函数（均方误差）
            loss = np.mean((output - y) ** 2)

            # 反向传播更新权重和偏置
            self.backward(X, y, lr)

            # 打印训练过程
            if epoch % 100 == 0:
                print(f"Epoch {epoch}: Loss = {loss:.4f}")
                self.loss_history.append(loss)

    def predict(self, X):
        # 前向传播预测
        output = self.forward(X)
        return output


# 读取 Excel 文件
data = pd.read_excel(r'D:\桌面\纠正后的数据.xlsx')

# 提取自变量和因变量列的数据
X = data[['CAP CLM 1 POSN', 'CAP WHL 1 POSN']]
y = data['COG NORM ACCEL']

# 转换为 NumPy 数组
X = np.array(X)
y = np.array(y).reshape(-1, 1)

# 创建 B-P 神经网络对象
nn = BPNeuralNetwork(input_dim=X.shape[1], hidden_dim=4,
output_dim=1)

# 训练神经网络
```

```
nn.train(X, y, epochs=1000, lr=0.1)


# 读取新的数据文件

new_data = pd.read_excel(r'D:\桌面\盘量和杆量.xlsx')

new_X = new_data[['CAP CLM 1 POSN', 'CAP WHL 1 POSN']]
new_X = np.array(new_X)


# 输出预测结果和真实值的对比

# predictions = nn.predict(new_X)
# for i in range(len(new_X)):
#   print(f"Input: {new_X[i]} - Prediction:
{predictions[i][0]} - Ground Truth: {y[i][0]}")

# 绘制预测结果和真实值的对比图表


# 创建新的图表对象

plt.figure(1)
plt.plot(predictions, label="Predictions")
plt.plot(y, label="Ground Truth")
plt.xlabel("Sample")
plt.ylabel("Value")
plt.title("Prediction vs. Ground Truth")
plt.legend()
plt.show()

# plt.savefig('D:\桌面\chart1.png')


# 绘制损失值随时间的变化曲线

plt.figure(2)
plt.plot(nn.loss_history)
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("Training Loss")
plt.show()

# plt.savefig('D:\桌面\chart2.png')

hidden_layer_weights = nn.W1
output_layer_weights = nn.W2
```

```
print("Final Hidden Layer Weights:")
print(hidden_layer_weights)

print("\nFinal Output Layer Weights:")
print(output_layer_weights)
```