队伍编号	MC2312927
题号	D

航空安全风险分析和飞行技术评估问题

摘 要

本文通过 QAR 数据分析，对航空安全风险分析和飞行技术评估问题进行研究。综合运用了数据归一化、K-means 算法、聚类分析法等方法，机器学习等方法的优势，根据题意构建相关模型，并借助 Jupyter 软件、Matlab 软件求解。

针对问题一，基于标准差、均方根、独热编码、最小缩放最大值的数据归一化算法和 K-means 算法进行提取相关程度较大的数据项，利用 Jupyter 软件求解。将得到的带有权重的部分关键数据项再运用 K-means 算法进行聚类，文中聚类参数取k=3，随机状态参数为 42，结果选择并获取前 10 个其中互信息值最大的特征名称列表，作为重要数据项。结果为：[FMF GROSS WEIGHT，MAGNETIC HEADING，LTITUDE (1013)，TRA-R.1，TRA-R，TRA-L，TRA-L.1，COMPUTED AIR SPD，PITCH ATT_MEAN，N1 SELTED-L]

针对问题二，基于多项式拟合并累加系数对预处理后的 QAR 数据进行对杠量的趋势分析，利用 Matlab 软件求解。利用操纵杆位置变化幅度、操纵杆位置变化速度和操纵杆位置稳定性对飞机操纵进行合理量化描述。其中将多项式拟合系数设为 1。滑动窗口趋势分析，数据可视化，绘制趋势线，对趋势线进行杆量的趋势分析。结论为：相关杠量曲线斜率为正且较大的趋势线表示操纵杆位置在随时间推移增加，可能对应着操作员对某个杆量的增大操作等。

针对问题三，基于岭回归模型和 K-Fold 交叉验证算法数据进行建模和验证，利用 Jupyter 软件求解。根据附件 2 中的预处理数据，运用岭回归模型进行建模，文中岭回归参数 alpha=1。再运用 K-Fold 交叉验证进行检验，文中测试数据集为总数据集的 0.2，随机状态参数为 42，参数 cv=5。最终得出 50 组相关性较大的数据项。

针对问题四，采用基于邻域粗糙集的属性约简方法。，提出了邻域半径的调整值，探讨了邻域半径对约简效果的影响，对原始飞行参数、不同邻域方法和因子分析三个方面对变量进行比较，通过其分析结果可采用随机森林模型预测。

针对问题五，基于数据的异常性，利用 Jupyter 软件建模对数据进行分区处理，使用函数判断数据分区，分析数据的风险性，对于不同分区实施多态化管理，以便于预警能够给予安组人员针对性的帮助。

关键词：独热编码，数据归一化，K-means 算法，岭回归模型，交叉验证，邻域粗糙集理论

目录

一、问题重述

1.1 问题的基本背景

近几十年来，中国民航领域快速发展，影响飞行安全的各方面隐患也随之产生。如何有力保障飞行安全成为阻碍民航业发展的一大难题。虽然中国民航航空事故发生率总体呈不断下降的趋势，但是导致事故的不安全事件却经常发生。为了及时发现飞行中的安全隐患，防止不安全事件的累积和严重化，因而需要聚焦飞行安全问题，强化航空安全研究，进行航空安全风险分析和研究飞行技术评估问题，通过有针对性、系统性的管控手段有效提升从业人员的技术和监测、预警风险，进而降低飞行事故的发生几率。

1.2 要求解决的问题

问题一：为减少错误数据在研究分析中带来的影响，对附件 1 中的 QAR 数据进行预处理，并提取部分关键数据项和分析其重要程度。

问题二：通过操纵杆的过程变化情况来分析飞机产生状态偏差的原因，根据附件 1，对飞行操纵进行合理量化描述。

问题三：根据附件 2 的数据，对超限的不同情况进行分析，研究和分析不同超限的基本特征。

问题四：根据附件 3 探讨和建立一种基于飞行参数的飞行技术评估方法数学模型，并分析飞行员的飞行技术。

问题五：假设飞行数据已能实现陆空实时传输，建立航空公司实时自动化预警机制模型，预防可能的安全事故发生，结合附件 1 的数据，并给出仿真结果。

二、问题分析

2.1 问题一的分析

针对问题一，在给定的附件 1 中的 QAR 数据进行去伪存真等预处理，并提取飞行安全相关的部分关键数据项并分析其重要程度。

首先对数据进行预处理。根据附件 1，对 QAR 数据进行预处理，填充删除后对结果的影响差异很小。

由于数据可以看成欧氏距离，所以运用标准差、均方根、独热编码和最小缩放最大值的数据归一化算法进行提取相关关键数据项。再将相关关键数据项运用 K-means 算法进行聚类，选择获取靠前部分其中互信息值较大的特征名称列表，并作为程度重要数据项。

2.2 问题二的分析

针对问题二，在给定的附件 1 中的 QAR 数据，通过操纵杆的过程变化情况来分析飞机产生状态偏差的原因，对飞行操纵进行合理量化描述。

首先对数据进行预处理。根据附件 1，对 QAR 数据进行预处理。

前面的线性回归是一种多元回归问题，每个样本对应于多个特征，在前面的例子中，特征之前的值相差不大，所以采用初始化趋势线系数矩阵和多项式拟合并进行累加系数处理 QAR 数据，依据操纵杆位置变化幅度、操纵杆位置变化速度和操纵杆位置稳定性，对结果的趋势线进行杆量的趋势分析，即对飞行操纵的合理化描述。

2.3 问题三的分析

针对问题三，在给定附件 2 的相关 QAR 数据进行预处理，对超限的不同情况进行分析，研究和分析不同超限的基本特征。

数据中处理特征数多，随后运用岭回归模型和线性回归模型对数据进行建模，再利用 K-Fold 交叉验证对两个模型进行检验，根据误差小的结果，挑选出更合适的数据模型并使用，对处理结果不同情况进行分析，研究不同超限的基本特征。

2.4 问题四的分析

针对问题四，在给定的附件 3 中的对 QAR 数据，通过飞行参数对飞行技术进行评估,，对数据进行预处理，收集并通过数值数据分类，对飞行资质进行合理化描述。

首先对数据进行预处理，对缺失值和异常值进行线性插值的方法，根据 QAR 数据的特点，去除不相关和重复的数据，保留与飞行参数相关的一系列数据，并对处理后的数据进行分析，提取与飞行参数相关的数据项，然后进行邻域粗糙集的属性约简方法，对数据进行分类，对数据进行归一化处理，确定各个属性的领域和相似阈值，通过对原始飞行参数、不同邻域方法和因子分析三个方面对变量进行比较，计算每个决策论的上近似和下近似，根据上.下近似，可以计算出规约，即属性集合的最小子集，找出与飞行资质相关的关键数据项，并对其结果利用随机森林的方法进行趋势分析，对飞行资质进行合理化描述。

2.5 问题五的分析

针对问题五，在给定附件 1。假设飞行数据已能实现陆空实时传输，建立航空公司实时自动化预警机制，预防可能的安全事故发生，计算仿真结果

首先，由于数据是无标签数据，处理的问题便为无监督问题，在此考虑从数据异常性着手，导入问题一已经预处理完善，对与飞行安全相关性最大的前 10 的特征，划分数据合理范围，当实时传入数据超出预定范围时，发出预警。

三、基本假设

1.数据的采集满足统计学要求，例如随机性、正态性、独立性、连续性等。
2.样本数据来自相同的地区：如果两个样本数据的差异可能由于地理位置的影响而产生，可以限制样本数据来自相同的地区，以减少对结果的影响。
3. 样本数据不受外来因素影响：如果样本数据可能受到外来因素的影响，例如天气、季节性变化等，需要采取相应的措施进行控制和调整。
4. 样本数据的质量和完整性：如果样本数据的质量存在问题，例如存在缺失、错误或异常值等情况，需要进行相应的数据处理和调整，以确保样本数据的完整性和准确性。
5.非题目要求外，分析相关数据时不考虑数据项以外的因素，如飞行员最近生活心理状态、机器老化、环境因素等。
6. 统计分析软件的使用：为了能够有效地分析样本数据并得到可靠的结果，需要确保使用的统计分析软件符合要求，且进行相应的参数和设置。

四、符号说明

符号	符号说明
S	样本标准差
n	第 i 个数据
$\bar{x}$	样本数据平均值
max	样本数据最大值
min	样本数据最小值
$\triangle P(t)$	操纵位置数据
$\triangle t$	单位时间间隔
$V(t)$	操纵位置变化速度
σ	操纵位置稳定系数
N	时间段内采样点数
w	岭回归估计

五、问题一的模型建立和求解

5.1 数据预处理

5.1.1 基本处理

首先对附件 1 现有 QAR 数据进行整合并部分读取显示，直观认识整体的数据信息，如图 1 示。

	日期：月	日期：天	格林威治标准时间	海拔（1013）	惯性垂直速度	无线电替代	计算空气SPD	地速	档位选择下降	哇指示空中	...	滑坡开发点（L）	滑坡开发点（R）	本地化开发点（C）	本地化开发点（L）	本地化器开发点（R）	俯仰率	出发机场	目的地机场	俯仰电阻率.1	FMF毛重
0	达克蒙	日期 CLKDAY	格林威治标准时间	备选项（ALT）	齐夫	GPRA	中国科学院	接地速度	LGDN	哇哇航空	...	GSDEVL	GSDEVR	洛克蒂夫克	洛克蒂夫尔	洛克蒂弗	PTCHATTRAT	德普塔普特	滕肛	PTCHATTRAT	FMFGROSSWT
1	月份	日	时	英尺	英尺	呜呜"	昆特斯	昆特斯	下	真	...	点	点	点	点	点	度/秒			度/秒	幼雅园
2	4	7	5:32:48	135	-5	-8	30	8	下	假	...	-0.58	-0.29	-3.17	-3.04	-3.28	0.0625	机场68	机场117	0.0625	347680
3	4	7	5:32:49	136	-9	-8	30	8.25	下	假	...	-0.62	-0.36	-2.78	-2.78	-2.95	-0.0625	机场68	机场117	-0.0625	347680
4	4	7	5:32:50	136	-12	-8	30	8.25	下	假	...	0.26	-0.1	-3.06	-2.96	-3.18	-0.0625	机场68	机场117	-0.0625	347680

5 行 × 64 列

图 1 附件一数据整合前 6 行显示图

其次对此 QAR 数据中的缺失值进行填充，并进行去除重复行和删除无关列，采用描述性统计处理异常值，再进行数据调整列，结果显示如图 2 示。

]:

	日期：天	海拔（1013）	惯性垂直速度	无线电替代	计算空气SPD	地速	档位选择下降	哇指示空中	哇指示空气.1	哇指示空气.2	...	滑坡开发点（L）	滑坡开发点（R）	本地化开发点（C）	本地化开发点（L）	本地化器开发点（R）	俯仰率	出发机场	目的地机场	俯仰电阻率.1	FMF毛重
0	7	135	-5	-8	30.0	8.00	0	0	0	0	...	-0.58	-0.29	-3.17	-3.04	-3.28	0.0625	68	117	0.0625	347680
1	7	136	-9	-8	30.0	8.25	0	0	0	0	...	-0.62	-0.36	-2.78	-2.78	-2.95	-0.0625	68	117	-0.0625	347680
2	7	136	-12	-8	30.0	8.25	0	0	0	0	...	0.26	-0.10	-3.06	-2.96	-3.18	-0.0625	68	117	-0.0625	347680
3	7	135	-4	-8	30.0	8.25	0	0	0	0	...	-0.21	0.13	-2.78	-2.78	-2.92	-0.0625	68	117	-0.0625	347680
4	7	136	-3	-8	30.0	8.50	0	0	0	0	...	0.00	-0.48	-2.94	-2.88	-3.09	0.0000	68	117	0.0000	347680
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
205088	10	-107	0	-8	30.0	0.75	0	0	0	0	...	1.38	1.03	2.67	2.34	2.76	-0.0625	73	68	-0.0625	237920
205089	10	-107	0	-8	30.0	0.75	0	0	0	0	...	1.66	0.40	2.48	2.60	2.56	0.0000	73	68	0.0000	237920
205090	10	-108	8	-8	30.0	0.75	0	0	0	0	...	1.82	1.03	2.22	2.70	2.29	-0.0625	73	68	-0.0625	237920
205091	10	-108	14	-8	30.0	0.75	0	0	0	0	...	2.00	0.55	2.33	2.69	2.39	0.0000	73	68	0.0000	237920
205092	10	-108	15	-8	30.0	0.75	0	0	0	0	...	2.02	0.67	2.41	2.72	2.43	0.0000	73	68	0.0000	237920

205093 行 × 62 列

图 2 附件一预处理结果显示图

5.1.2 标准差和均方根

标准差[1]是一组数据平均值分散程度的一种度量，标准差是反映一组数据离散程度最常用的一种量化形式，是表示精确度的重要指标。我们使用方法去检测它，但检测方法总是有误差的，所以检测值并不是其真实值。检测值与真实值之间的差距就是评价检测方法最有决定性的指标。所以我们先采用标准差进行剔除异常数据。

$$S=\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \tag{1}$$

（1）式中 S 指样本标准差，n 指样本总数，$\bar{x}$指样本数据平均值，x_i 指第 i 个数据。

均方根[2]误差，它是观测值与真值偏差的平方和观测次数 n 比值的平方根，在实际测量中，观测次数 n 总是有限的，真值只能用最可信赖（最佳）值来代替。方根误差对一组测量中的特大或特小误差反映非常敏感，所以，均方根误差能够很好地反映出测量的精密度。所以运用均方根方法。

$$X_{rms}=\sqrt{\frac{\sum_{i=1}^{N}X_i^2}{N}}=\sqrt{\frac{X_1^2+X_2^2+\cdots+X_N^2}{N}} \tag{2}$$

（2）式中 X_{rms} 指样本均方根值，n 指样本总数，$\bar{x}$ 指样本数据平均值，x_i 指第 i 个数据。

本文分别进行多次标准差方法和多次均方根方法处理，对 QAR 数据中异常数据进行剔除和一般的降维，结果如图 3 示。

	日期：天	海拔(1013)	惯性垂直速度	无线电替代	计算空气SPD	地速	档位选择下降	滚动阿特	卷1	N1 SELTED-L	...	齿轮规范 ACCEL_STD	齿轮规范 ACCEL_MEAN	RMS_LOCALIZER	RMS_GLIDESLOPE	音高 ATTL_STD	音高 ATT_MEAN	帽 CLM 1 POSN_STD
0	7	135	-5	-8	30.0	8.00	0	0.0879	0.0879	21.4	...	0.010244	1.00117	3.164854	0.469255	0.0	-0.3516	0.000492
1	7	136	-9	-8	30.0	8.25	0	0.0879	0.0879	21.4	...	0.015708	1.00000	2.837798	0.494941	0.0	-0.3516	0.000602
2	7	136	-12	-8	30.0	8.25	0	0.0879	0.0879	21.4	...	0.012664	0.99961	3.067985	0.167332	0.0	-0.3516	0.000000
3	7	135	-4	-8	30.0	8.25	0	0.0879	0.0879	21.4	...	0.017806	0.99688	2.827437	0.173109	0.0	-0.3516	0.000492
4	7	136	-3	-8	30.0	8.50	0	0.0879	0.0879	21.4	...	0.011303	1.00078	2.971313	0.278628	0.0	-0.3516	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
205088	10	-107	0	-8	30.0	0.75	0	0.0879	0.0879	21.1	...	0.005265	0.99766	2.596286	1.231923	0.0	-0.7031	0.000000
205089	10	-107	0	-8	30.0	0.75	0	0.0879	0.0879	21.0	...	0.006153	0.99844	2.547155	1.235543	0.0	-0.7031	0.000000
205090	10	-108	8	-8	30.0	0.75	0	0.0879	0.0879	21.1	...	0.005652	0.99961	2.412640	1.615849	0.0	-0.7031	0.000000
205091	10	-108	14	-8	30.0	0.75	0	0.0879	0.0879	18.7	...	0.005872	0.99844	2.475015	1.461164	0.0	-0.7031	0.000000
205092	10	-108	15	-8	30.0	0.75	0	0.0879	0.0879	16.9	...	0.003768	0.99766	2.523978	1.576219	0.0	-0.7031	0.000000

205093 行 × 38 列

图 3 一般的降维处理结果显示图

5.1.3 独热编码

独热编码[3]即 One-Hot 编码，其方法是使用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由他独立的寄存器位。One-Hot 编码，将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点，会让特征之间的距离计算更加合理。One-Hot 编码后的特征，每一维度的特征都可以看做是连续的特征。本文对接下来的数据重复使用三次独热编码进行降维，结果如图 4 示。

	海拔(1013)	惯性垂直速度	无线电替代	计算空气SPD	地速	档位选择下降	滚动阿特	卷1	N1 SELTED-L	N1 SELTED-R	...	音高 ATTL_STD	音高 ATT_MEAN	帽 CLM 1 POSN_STD	帽 CLM 1 POSN_MEAN	帽 WHL 1 POSN_STD	帽 WHL 1 POSN_MEAN	日期：DAY_7	日期：DAY_8	期 DAY
0	135	-5	-8	30.0	8.00	0	0.0879	0.0879	21.4	21.4	...	0.0	-0.3516	0.000492	-0.00122	0.000000	0.01150	1	0	
1	136	-9	-8	30.0	8.25	0	0.0879	0.0879	21.4	21.5	...	0.0	-0.3516	0.000602	-0.00144	0.000602	0.01084	1	0	
2	136	-12	-8	30.0	8.25	0	0.0879	0.0879	21.4	21.5	...	0.0	-0.3516	0.000000	-0.00100	0.000000	0.01040	1	0	
3	135	-4	-8	30.0	8.25	0	0.0879	0.0879	21.4	21.5	...	0.0	-0.3516	0.000492	-0.00122	0.000000	0.01040	1	0	
4	136	-3	-8	30.0	8.50	0	0.0879	0.0879	21.4	21.5	...	0.0	-0.3516	0.000000	-0.00100	0.000492	0.01062	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
205088	-107	0	-8	30.0	0.75	0	0.0879	0.0879	21.1	21.2	...	0.0	-0.7031	0.000000	-0.00210	0.000000	0.00840	0	0	
205089	-107	0	-8	30.0	0.75	0	0.0879	0.0879	21.0	20.8	...	0.0	-0.7031	0.000000	-0.00210	0.000447	0.00860	0	0	
205090	-108	8	-8	30.0	0.75	0	0.0879	0.0879	21.1	18.9	...	0.0	-0.7031	0.000000	-0.00210	0.000000	0.00840	0	0	
205091	-108	14	-8	30.0	0.75	0	0.0879	0.0879	18.7	17.3	...	0.0	-0.7031	0.000000	-0.00210	0.000000	0.00840	0	0	
205092	-108	15	-8	30.0	0.75	0	0.0879	0.0879	16.9	15.8	...	0.0	-0.7031	0.000000	-0.00210	0.000000	0.00840	0	0	

205093 行 × 41 列

图 4 独热编码处理结果显示图

5.1.4 最小缩放最大值的数据归一化算法

在建模之前，都需要对数据进行标准化处理，以消除量纲的影响。如果对未标准化的数据直接进行建模，可能会导致模型对数值大的变量学习过多，而对数值小的变量训练不够充分，往往模型效果会不好。但是最大最小归一化[4]容易受极端值的影响，所以在此之前已经运用多种方法剔除了异常值，现在运用最大最小归一化对数据进行处理标准处理。

$$x' = \frac{x - \min}{\max - \min} \tag{3}$$

（3）式中 x’ 指修改后数据，*max* 指数据中最大值，*min* 指数据中最小值，*x* 指当前使用的数据

本文针对独热编码筛选后的数据进行了一次最小缩放最大值的数据归一化算法，结果如图 5 示。

归一化后的数据：

	海拔(1013)	惯性垂直速度	无线电替代	计算空气SPD	地速	档位选择下降	滚动阿特	卷1	N1 SELTED-L	N1 SELTED-R	...	DEPARTURE_AIRPORT_5	DEPARTURE_AIRPORT_68	DEPARTURE_AIRPORT_73
0	0.009025	0.569260	0.49829	0.0	0.014085	0.0	0.471557	0.472264	0.067553	0.064444	...	0.0	1.0	0.0
1	0.009052	0.568810	0.49829	0.0	0.014525	0.0	0.471557	0.472264	0.067553	0.065556	...	0.0	1.0	0.0
2	0.009052	0.568473	0.49829	0.0	0.014525	0.0	0.471557	0.472264	0.067553	0.065556	...	0.0	1.0	0.0
3	0.009025	0.569373	0.49829	0.0	0.014525	0.0	0.471557	0.472264	0.067553	0.065556	...	0.0	1.0	0.0
4	0.009052	0.569485	0.49829	0.0	0.014965	0.0	0.471557	0.472264	0.067553	0.065556	...	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
205088	0.002544	0.569822	0.49829	0.0	0.001320	0.0	0.471557	0.472264	0.064230	0.062222	...	0.0	0.0	1.0
205089	0.002544	0.569822	0.49829	0.0	0.001320	0.0	0.471557	0.472264	0.063123	0.057778	...	0.0	0.0	1.0
205090	0.002517	0.570722	0.49829	0.0	0.001320	0.0	0.471557	0.472264	0.064230	0.036667	...	0.0	0.0	1.0
205091	0.002517	0.571396	0.49829	0.0	0.001320	0.0	0.471557	0.472264	0.037652	0.018889	...	0.0	0.0	1.0
205092	0.002517	0.571509	0.49829	0.0	0.001320	0.0	0.471557	0.472264	0.017719	0.002222	...	0.0	0.0	1.0

205093 行 × 49 列

图 5 最小缩放最大值的归一化算法处理结果显示图

5.2 K-means 算法模型建立

K-means[5]的本质是基于欧式距离的数据划分算法。容易理解，聚类效果不错，虽然是局部最优，但往往局部最优就够了；处理大量数据集的时候，该算法可以保证较好的伸缩性。

K-means 的算法步骤为：

步骤一：随机初始化 3 个点作为簇质心；

步骤二：将样本集中的每个点分配到一个簇中；

步骤三：计算每个点与质心之间的距离（常用欧式距离），并将其分配给距离最近的质心所对应的簇中；

步骤四：更新簇的质心。每个簇的质心更新为该簇所有点的平均值；

步骤五：反复迭代步骤三和步骤四，直到达到某个终止条件；

K-means 算法示意图如图 6 示。

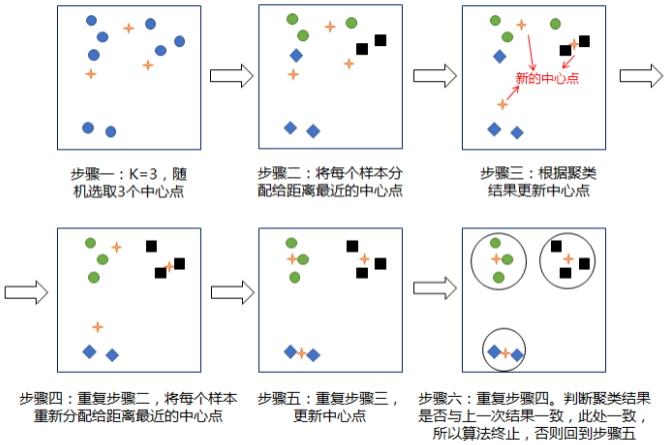

图 6 K-means 算法示意图

本文使用聚类参数取k=3，随机状态参数为 42，确定特征的重要性，得到每个数据项及其程度比例，现在将其排列输出，结果部分如图 7 示。

```
for i in range(df6.shape[1]):
    print("Feature {} Mutual Information: {:.4f}".format(i, mutual_info_scores[i]))
```

```
Feature 0 Mutual Information: 0.6162
Feature 1 Mutual Information: 0.0917
Feature 2 Mutual Information: 0.1129
Feature 3 Mutual Information: 0.3990
Feature 4 Mutual Information: 0.2289
Feature 5 Mutual Information: 0.0479
Feature 6 Mutual Information: 0.0536
Feature 7 Mutual Information: 0.0581
Feature 8 Mutual Information: 0.3063
Feature 9 Mutual Information: 0.3056
Feature 10 Mutual Information: 0.7034
Feature 11 Mutual Information: 0.1503
Feature 12 Mutual Information: 0.0957
Feature 13 Mutual Information: 0.0423
Feature 14 Mutual Information: 0.0530
Feature 15 Mutual Information: 0.4664
Feature 16 Mutual Information: 0.4658
Feature 17 Mutual Information: 0.4695
Feature 18 Mutual Information: 0.4703
Feature 19 Mutual Information: 0.0338
Feature 20 Mutual Information: 0.0153
```

图 7 K-means 算法部分结果显示图

5.3 模型结果

根据 K-means 算法模型求解出的结果，再根据其相关程度比例排序，选择前 10 个数据项，并作为重要程度较大的数据项，[FMF GROSS WEIGHT，MAGNETIC HEADING，LTITUDE (1013)，TRA-R.1，TRA-R，TRA-L，TRA-L.1，COMPUTED AIR SPD，PITCH ATT_MEAN，N1 SELTED-L]，结果如图 8 示。

```
{'FMF GROSS WEIGHT': 1.0772040072225644,
 'MAGNETIC HEADING': 0.7033938576718883,
 'ALTITUDE (1013)': 0.6161810329581932,
 'TRA-R.1': 0.4702851787321869,
 'TRA-R': 0.46954438749493677,
 'TRA-L': 0.4664298446926787,
 'TRA-L.1': 0.4658418133645956,
 'COMPUTED AIR SPD': 0.398997384815331,
 'PITCH ATT_MEAN': 0.3588688236763906,
 'N1 SELTED-L': 0.30631984567708614}
```

图 8 结果前 10 个重要数据项显示图

六、问题二的模型建立和求解

6.1 数据的预处理

本题和问题一使用附件 1 中同一 QAR 数据，在此依然使用问题一中数据预处理后的数据结果。

6.2 模型建立

6.2.1 操纵杆位置原理依据

操纵杆[6]位置变化幅度：可以计算操纵杆在单位时间内的位置变化幅度，例如每秒钟的变化速率或者在特定时间段内的累积变化量。这可以用来描述操纵杆在飞行过程中的操纵幅度大小，包括杆位变化的快慢和幅度的大小。

$$\triangle \mathrm{p(t)} = \mathrm{p(t)} - \mathrm{p(t} - \triangle \mathrm{t)} \tag{4}$$

公式（4）中$\triangle P(t)$指操纵位置数据，$\triangle t$ 指单位时间间隔。

操纵杆位置变化速度：可以计算操纵杆在单位时间内的位置变化速度，例如每秒钟的速度或者在特定时间段内的平均速度。这可以用来描述操纵杆的动态变化情况，包括操纵杆在单位时间内的运动速度，如是否存在急剧的加速或减速。

$$\mathrm{v(t)} = \triangle \mathrm{p(t)} / \triangle \mathrm{t} \tag{5}$$

公式（5）中 $V(t)$指操纵杆位置变化速度，$\triangle P(t)$指操纵位置数据，$\triangle t$ 指单位时间间隔。

操纵杆位置稳定性：可以计算操纵杆在一段时间内的波动情况，例如标准差或方差等指标来描述操纵杆位置的稳定性。这可以用来描述操纵杆的稳定性和平稳性，包括操纵杆位置的波动情况，如是否存在剧烈的波动或者频繁的变化。

$$\sigma = \sqrt[2]{\frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left(\mathrm{p}\left(\mathrm{t_i}\right) - \overline{\mathrm{p}} \right)^2} \tag{6}$$

公式（5）中 σ 指操纵杆位置稳定性系数，N 指时间段内的采样点数，t_i 指 i 个采样点的时间，$P(t_i)$指操纵位置数据，P 指单位时间间隔，$\bar{p}$指操纵平均位置数据。

操纵杆位置变化趋势：可以通过分析操纵杆位置的变化趋势，例如斜率或趋势线的拟合情况来描述操纵杆的变化趋势。这可以用来描述操纵杆在飞行过程中的趋势性变化，包括操纵杆位置的变化趋势，如是否持续上升、下降或者持平。

6.2.2 多项式拟合算法[7]

最小二乘法拟合曲线的原理并且找出这种拟合方法的不足,针对这种不足提出另外一种新的拟合方法正交多项式拟合。这种方法能弥补最小二乘拟合中的 x 拟合 y 和 y 拟合 x 出现的曲线不一样的现象。在实际测量工作中所测得的一系列坐标点 x,y 必然

都存在随机误差,如果只用普通的多项式最小二乘法来拟合显然就有较大误差了。由此本文采用多项式拟合算法。

(1).设拟合多项式为：

$$y = a_0 + a_1 x + \ldots + a_k x^k, \tag{7}$$

(2).各点到这条曲线的距离之和，即偏差平方和如下：

$$R^2 \equiv \sum_{i=1}^{n} \left[y_i - \left(a_0 + a_1 x_i + \ldots + a_k x_i^k \right) \right]^2. \tag{8}$$

(3).为了求得符合条件的 a 值，对等式右边求 ai 偏导数，因而我们得到了：

$$-2 \sum_{i=1}^{n} \left[y - \left(a_0 + a_1 x + \ldots + a_k x^k \right) \right] x^k = 0. \tag{9}$$

(4).将等式左边进行一下化简，然后应该可以得到下面的等式：

$$a_0 \sum_{i=1}^{n} x_i^k + a_1 \sum_{i=1}^{n} x_i^{k+1} + \ldots + a_k \sum_{i=1}^{n} x_i^{2k} \tag{10}$$

(5).把这些等式表示成矩阵的形式，就可以得到下面的矩阵：

$$\begin{cases} na_0 + (\sum_{i=1}^{k} x_i)a_1 + \ldots + (\sum_{i=1}^{k} x_i^k)a_k = \sum_{i=1}^{k} y_i \\ (\sum_{i=1}^{k} x_i)a_1 + (\sum_{i=1}^{k} x_i^2)a_2 \ldots + (\sum_{i=1} x^{k+1})a_k = \sum_{i=1}^{k} x_i y_i \\ (\sum_{i=1}^{k} x_i^k)a_1 + (\sum_{i=1}^{k} x_i^{k+1})a_2 \ldots + (\sum_{i=1} x^{2k})a_k = \sum_{i=1}^{k} x_i^k y_i \end{cases} \tag{11}$$

(6).将这个范德蒙得矩阵化简后可得到:

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^k \\ 1 & x_2 & \cdots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \tag{12}$$

(7).也就是说 X*A=Y，那么 A = (X'*X)-1*X'*Y，便得到了系数矩阵 A，同时，我们也就得到了拟合曲线。

6.3 模型结果

首先提取杆量数据，初始化趋势线系数矩阵，多项式拟合的阶数，这里选择 1 表示线性拟合的阶数，并进行滑动窗口趋势分析，之后绘制趋势线，对趋势线进行分析得到模型结果。

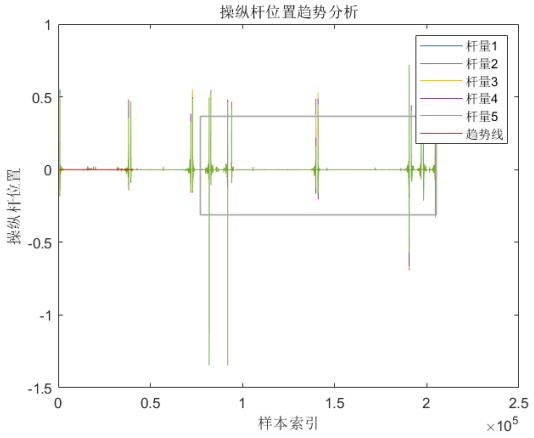

图 9 操纵杆位置趋势分析图

图形的 x 轴表示样本索引，y 轴表示操纵杆位置的数值。每个杆量（1-5）都会在图中以不同的颜色进行显示，并且每个杆量对应的趋势线也会以红色进行绘制。
趋势线表示了操纵杆位置数据在时间上的变化趋势。通过多项式拟合，我们可以得到趋势线的斜率和截距，从而了解操纵杆位置随着时间的变化趋势。趋势线的斜率表示了操纵杆位置的变化速率，截距表示了初始位置。趋势曲线结果如图 9 示。

通过观察趋势线的斜率和截距，可以得出以下结论：

斜率为正且较大的趋势线表示操纵杆位置在随时间推移增加，可能对应着操作员对某个杆量的增大操作。

斜率为负且较大的趋势线表示操纵杆位置在随时间推移减小，可能对应着操作员对某个杆量的减小操作。

斜率接近零的趋势线表示操纵杆位置在时间上没有明显的变化趋势，可能对应着操作员对某个杆量的保持稳定的操作。
截距表示了操纵杆位置的初始位置，可以用于了解操作员在开始记录数据时的操纵杆位置状态。

七、问题三的模型建立和求解

7.1 数据预处理

7.1.1 基本处理

首先对附件 2 中现有 QAR 数据缺失值进行填充，并进行去除重复行和删除无关列，然后分别取不相同的数转列表，直观认识整体的数据信息，如图 10 示。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44103 entries, 0 to 44102
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ALERT（警告级别）       44103 non-null  int64
 1   ARN（机号）           44103 non-null  object
 2   ARR（目的机场）         44103 non-null  int64
 3   DEP（起飞机场）         44103 non-null  int64
 4   EVENT_NAME(超限名称)  44103 non-null  object
 5   PHASE（飞行阶段）       44103 non-null  int64
 6   月份              44103 non-null  int64
dtypes: int64(5), object(2)
memory usage: 2.4+ MB
```

图 10 附件二基本处理结果显示图

7.1.2 独热编码

独热编码[3]即 One-Hot 编码，其方法是使用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由他独立的寄存器位。One-Hot 编码，将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点，会让特征之间的距离计算更加合理。One-Hot 编码后的特征，每一维度的特征都可以看做是连续的特征。本文对接下来的数据中每一列运用独热编码进行降维，并得到更稳定和可靠的分析结果。结果如图 11 示。

	EVENT_NAME(超限名称)	ARN（机号）_100号机	ARN（机号）_101号机	ARN（机号）_102号机	ARN（机号）_103号机	ARN（机号）_104号机	ARN（机号）_105号机	ARN（机号）_106号机	ARN（机号）_107号机	ARN（机号）_108号机	...	DEP（起飞机场）_109	DEP（起飞机场）_110	DEP（起飞机场）_111	DEP（起飞机场）_112	DEP（起飞机场）_113	DEP（起飞机场）_114	DEP（起飞机场）_115	DEP（起飞机场）_116	ALERT（警告级别）_2	ALERT（警告级别）_3
0	50英尺至接地距离远	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
1	50英尺至接地距离远	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
2	GPWS警告(sink rate)	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
3	下降率大400-50ft	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
4	50英尺至接地距离远	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
44098	爬升速度大35-1000ft	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	1	0
44099	爬升速度大35-1000ft	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	1
44100	爬升速度大35-1000ft	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	1	0
44101	50英尺至接地距离远	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	1	0
44102	接地速度小	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	1	0

44103 rows × 380 columns

图 11 每列独热编码处理结果显示图

7.2 模型建立

7.2.1 岭回归模型

岭回归[8]实质上是一种改良的最小二乘估计法，通过放弃最小二乘法的无偏性，以损失部分信息、降低精度为代价获得回归系数更为符合实际、更可靠的回归方法，对病态数据的拟合要强于最小二乘法。

$$w = (X^T X + kI)^{-1} X^T y \tag{13}$$

公式（）w 为岭回归估计，k 为岭参数，当 k=0 时，岭回归估计就是普通的最小二乘估计

对运用独热编码后的数据进行多元线性回归模型建立，找出对应的前三列影响最大的因子并输出，现部分结果如图 12 示。

```
for label, cols in top3_cols.items():
    print(f'{label}\t\t{", ".join(cols)}')
```

```
标签              前三列名称
50英尺至接地距离近              DEP（起飞机场）_46，ARN（机号）_33号机，ARR（目的机场）_109
50英尺至接地距离远              ARN（机号）_90号机，ARN（机号）_24号机，PHASE（飞行阶段）_6
GPWS警告(sink rate)             ARN（机号）_9号机，ARR（目的机场）_54，ARR（目的机场）_65
GPWS警告(windshear)             ARR（目的机场）_74，DEP（起飞机场）_106，DEP（起飞机场）_22
High normal accel with flap (in flight)         ARR（目的机场）_4，DEP（起飞机场）_19，ARN（机号）_6号机
Left of centreline below 1000ft         月份_201606，ARR（目的机场）_16，ARN（机号）_3号机
TCAS RA 警告            月份_201608，ARR（目的机场）_78，ARN（机号）_82号机
下滑道警告              ARN（机号）_68号机，ARN（机号）_91号机，ARR（目的机场）_95
下降率大1000-400(含)ft          ARR（目的机场）_69，ARR（目的机场）_65，ARR（目的机场）_36
下降率大2000-1000（含）ft               ARR（目的机场）_64，PHASE（飞行阶段）_5，ARR（目的机场）_84
下降率大400-50ft                ARN（机号）_11号机，ARN（机号）_31号机，PHASE（飞行阶段）_5
低于下滑道              DEP（起飞机场）_72，ARR（目的机场）_80，PHASE（飞行阶段）_5
低空大速度2500ft以下            ARN（机号）_90号机，ARR（目的机场）_67，ARN（机号）_94号机
坡度大400/1500ft以上            ARN（机号）_55号机，ARN（机号）_36号机，ARR（目的机场）_49
复飞            ARR（目的机场）_93，ARN（机号）_95号机，ARN（机号）_107号机
抬前轮速度大            ARN（机号）_28号机，ARR（目的机场）_101，PHASE（飞行阶段）_2
抬前轮速度小            ARR（目的机场）_98，ARR（目的机场）_44，PHASE（飞行阶段）_2
抬头速率大              DEP（起飞机场）_16，ARN（机号）_69号机，PHASE（飞行阶段）_2
抬头速率小              ARN（机号）_84号机，ARN（机号）_77号机，PHASE（飞行阶段）_0
接地速度小              ARR（目的机场）_9，DEP（起飞机场）_24，PHASE（飞行阶段）_6
收反推晚                ARR（目的机场）_107，ARR（目的机场）_65，PHASE（飞行阶段）_7
放起落架晚              ARR（目的机场）_4，DEP（起飞机场）_6，ARR（目的机场）_22
未知            ARN（机号）_97号机，ARN（机号）_22号机，PHASE（飞行阶段）_0
爬升坡度大35-400(含)ft          月份_201507，ARN（机号）_18号机，ARR（目的机场）_99
爬升速度大35-1000ft             ARR（目的机场）_66，DEP（起飞机场）_115，PHASE（飞行阶段）_4
爬升速度小35-1000ft             ARR（目的机场）_30，ARR（目的机场）_98，ARR（目的机场）_91
直线滑行速度大          DEP（起飞机场）_49，ARN（机号）_95号机，PHASE（飞行阶段）_1
着陆俯仰角大            ARR（目的机场）_88，DEP（起飞机场）_83，PHASE（飞行阶段）_7
着陆俯仰角小            ARR（目的机场）_31，ARR（目的机场）_91，ARR（目的机场）_25
着陆垂直载荷大          ARR（目的机场）_75，ARR（目的机场）_82，PHASE（飞行阶段）_7
着陆襟翼到位晚          ARN（机号）_68号机，ARR（目的机场）_101，ARR（目的机场）_96
着陆重、跳着陆          DEP（起飞机场）_45，PHASE（飞行阶段）_7，ARN（机号）_108号机
离地仰角大              ARN（机号）_82号机，ARN（机号）_97号机，ARN（机号）_57号机
离地速度大              ARR（目的机场）_101，ARR（目的机场）_63，PHASE（飞行阶段）_3
离地速度小              ARR（目的机场）_99，ARN（机号）_84号机，PHASE（飞行阶段）_3
空中过载                DEP（起飞机场）_84，DEP（起飞机场）_48，ARR（目的机场）_9
起飞EGT超限             ARN（机号）_49号机，DEP（起飞机场）_50，ARR（目的机场）_4
起飞坡度大0-35（含）ft          ARR（目的机场）_46，ARN（机号）_84号机，ARN（机号）_111号机
起飞收起落架晚          DEP（起飞机场）_43，PHASE（飞行阶段）_0，ARN（机号）_93号机
起飞滑跑方向不稳定              ARN（机号）_60号机，PHASE（飞行阶段）_2，DEP（起飞机场）_108
超襟翼限制速度Vfe               ARR（目的机场）_23，ARN（机号）_66号机，ARN（机号）_9号机
```

图 12 岭回归处理结果部分输出显示图

7.2.2 多元线性回归模型

多元线性回归[9]通过获取数据集中的线性关系，就是假设输出 y 和输入 x 之间存在线性关系即为样本 x 的每一个特征与对应权重相乘就和就是 y 的值，θ_0 代表线性模型中的截距项，在矩阵乘法中其对应的权重是 1。

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \varepsilon, \tag{14}$$

公式（13）其中 ε 为随机误差，服从均值为 0 的正态分布。

对运用独热编码后的数据进行多元线性回归模型建立，找出对应的前三列影响最大的因子并输出，现部分结果如图 13 示。

```python
# 遍历每个标签，找出对应的前三列影响最大的因子
for i, label in enumerate(labels_encoded.columns):
    idx = coefs[i].argsort()[-3:]  # 找出最大的三个系数的索引
    top5_cols[label] = features.columns[idx]  # 存储对应的列名称

# 输出结果
print('标签\t\t前三列名称')
for label, cols in top5_cols.items():
    print(f'{label}\t\t{", ".join(cols)}')
```

```
标签            前三列名称
50英尺至接地距离近              DEP（起飞机场）_46, ARR（目的机场）_76, ARR（目的机场）_24
50英尺至接地距离远              ARR（目的机场）_107, ARR（目的机场）_53, ARR（目的机场）_13
GPWS警告(sink rate)             DEP（起飞机场）_1, ARR（目的机场）_76, ARR（目的机场）_24
GPWS警告(windshear)             DEP（起飞机场）_22, ARR（目的机场）_24, ARR（目的机场）_76
High normal accel with flap (in flight)         ARN（机号）_48号机, ARN（机号）_6号机, DEP（起飞机场）_113
Left of centreline below 1000ft         月份_201506, 月份_201507, 月份_201606
TCAS RA 警告            DEP（起飞机场）_93, ARR（目的机场）_76, ARR（目的机场）_24
下滑道警告              ARN（机号）_68号机, ARN（机号）_91号机, DEP（起飞机场）_113
下降率大1000-400(含)ft          ARR（目的机场）_65, ARR（目的机场）_36, DEP（起飞机场）_113
下降率大2000-1000（含）ft               DEP（起飞机场）_23, ARR（目的机场）_76, ARR（目的机场）_24
下降率大400-50ft                DEP（起飞机场）_60, ARR（目的机场）_76, ARR（目的机场）_24
低于下滑道              ARN（机号）_32号机, ARN（机号）_47号机, ARN（机号）_98号机
低空大速度2500ft以下            月份_201505, 月份_201502, 月份_201602
坡度大400/1500ft以上            ARR（目的机场）_95, ARR（目的机场）_49, DEP（起飞机场）_113
复飞            月份_201609, 月份_201608, 月份_201611
抬前轮速度大            DEP（起飞机场）_83, ARR（目的机场）_76, ARR（目的机场）_24
抬前轮速度小            月份_201507, 月份_201604, 月份_201508
抬头速率大              月份_201607, 月份_201602, DEP（起飞机场）_113
抬头速率小              DEP（起飞机场）_114, ARR（目的机场）_24, ARR（目的机场）_76
接地速度小              月份_201507, 月份_201501, 月份_201505
收反推晚                ARR（目的机场）_107, ARR（目的机场）_65, DEP（起飞机场）_113
放起落架晚              DEP（起飞机场）_6, ARR（目的机场）_24, ARR（目的机场）_76
未知            ARR（目的机场）_20, ARR（目的机场）_76, ARR（目的机场）_24
爬升坡度大35-400(含)ft          月份_201609, 月份_201608, 月份_201507
爬升速度大35-1000ft             ARN（机号）_65号机, ARN（机号）_64号机, DEP（起飞机场）_113
爬升速度小35-1000ft             ARR（目的机场）_98, ARR（目的机场）_91, DEP（起飞机场）_113
直线滑行速度大          ARR（目的机场）_17, ARR（目的机场）_76, ARR（目的机场）_24
着陆俯仰角大            ARR（目的机场）_103, ARR（目的机场）_88, DEP（起飞机场）_113
着陆俯仰角小            PHASE（飞行阶段）_6, PHASE（飞行阶段）_5, PHASE（飞行阶段）_0
着陆垂直载荷大          DEP（起飞机场）_49, ARR（目的机场）_76, ARR（目的机场）_24
着陆襟翼到位晚          DEP（起飞机场）_91, ARR（目的机场）_24, ARR（目的机场）_76
着陆重、跳着陆          DEP（起飞机场）_45, ARR（目的机场）_24, ARR（目的机场）_76
离地仰角大              PHASE（飞行阶段） 2, PHASE（飞行阶段） 4, PHASE（飞行阶段） 3
```

图 13 多元线性回归处理结果部分输出显示图

7.2.3 K-Fold 交叉验证

交叉验证[10]经常用于数据的验证，原理是将数据分为 n 组，每组数据都要作为一次验证集进行一次验证，而其余的 n-1 组数据作为训练集。这样一共要循环 n 次，验证 n 次，得到 n 个模型，这 n 个模型得到的 n 个误差计算均值，得到交叉验证误差。

K 折交叉验证（K-Fold）会将数据集划分为 k 个分组，成为折叠（fold）。如果 k 的值等于数据集实例的个数，那么每次的测试集就只有一个，这种处理方式称为“留一”。

现将 K-Fold 交叉验证的参数 cv=5，随机状态参数为 42。对岭回归模型进行检测，计算出平均均方误差。如图 14 示。

```python
print("Mean Squared Error (MSE):", mse_scores.mean())
```

```
岭回归模型的每折交叉验证的均方误差,平均均方误差
Mean Squared Error for each fold: [0.00798278 0.00807174 0.00807823 0.00803226 0.00817225]
Mean Squared Error (MSE): 0.008067451951656 86
```

图 14 岭回归模型平均均方误差示意图

现对多元线性回归模型进行检测，计算出平均均方误差。如图 15 示。

```
print("Mean Squared Error (MSE):", mse_mean)
```

```
线性回归模型的每折交叉验证的均方误差,平均均方误差
Mean Squared Error for each fold: [1.85519468e+20 8.12605321e+19 1.27686951e+16 3.69909206e+18
 2.24166621e+19]
Mean Squared Error (MSE): 5.8581704657836474e+19
```

图 15 多元线性回归模型平均均方误差示意图

7.2.4 模型选择

岭回归通过引入正则化项，可以有效处理数据中存在的多重共线性问题，避免模型过于依赖某个特征，提高模型的稳定性和可靠性。

控制模型复杂度：岭回归通过正则化项控制模型的复杂度，可以防止过拟合，提高模型的泛化能力。

对异常值具有鲁棒性：岭回归对异常值的影响较小，能够在存在少量异常值的情况下依然获得较好的拟合效果。

可调节的超参数：岭回归通过调节超参数（正则化强度）来控制模型的拟合效果，可以根据实际情况进行调优。

通过以上实现思路和使用岭回归的方法，可以对附件 2 中的数据进行分析，研究不同超限的基本特征，如航线、机场等，并得出结论。同时，独热编码和岭回归方法可以在处理非数值列和解决多重共线性问题时提供有效的解决方案，从而得到更稳定和可靠的分析结果。

根据给出的结果，线性回归的均方误差（MSE）明显较大，为 5.8581704657836474e+19，而岭回归的均方误差（MSE）较小，为 0.00806745195165686。因此，从评估指标 MSE 的角度来看，岭回归模型在这个数据集上的表现更好，因为其均方误差较小。

MSE 是评估模型预测效果的一种常用指标，其值越小表示模型预测效果越好。在这里，线性回归模型的 MSE 值较大，说明模型在训练集上的预测效果较差，可能存在过拟合的情况。而岭回归模型的 MSE 值较小，说明模型在训练集上的预测效果较好，相对来说更具有泛化能力，可以更好地应对新数据的预测。因此，从 MSE 指标来看，岭回归模型优于线性回归模型。

7.3 模型结果

将岭回归算法输出前三列数据相关分别作为发生超限原因的主要三项相关项。结果如下：

（1）50 英尺至接地距离近　　DEP（起飞机场）_46，ARN（机号）_33 号机，ARR（目的机场）_109

(2）50 英尺至接地距离远 ARN（机号）_90 号机，ARN（机号）_24 号机，PHASE（飞行阶段）_6

(3)GPWS 警告(sink rate)　　ARN（机号）_9 号机，ARR（目的机场）_54，ARR（目的机场）_65

(4)GPWS 警告(windshear)　　ARR（目的机场）_74，DEP（起飞机场）_106，DEP（起飞机场）_22

(5)High normal accel with flap (in flight)　　ARR（目的机场）_4，DEP（起飞机场）_19，ARN（机号）_6 号机

(6)Left of centreline below 1000ft 月份_201606，ARR（目的机场）_16，ARN（机号）_3 号机
(7)TCAS RA 警告 月份_201608，ARR（目的机场）_78，ARN（机号）_82 号机
(8)下滑道警告 ARN（机号）_68 号机，ARN（机号）_91 号机，ARR（目的机场）_95
(9)下降率大 1000-400(含)ft ARR（目的机场）_69，ARR（目的机场）_65，ARR（目的机场）_36
(10)下降率大 2000-1000（含）ft ARR（目的机场）_64，PHASE（飞行阶段）_5，ARR（目的机场）_84
(11)下降率大 400-50ft ARN（机号）_11 号机，ARN（机号）_31 号机，PHASE（飞行阶段）_5
(12)低于下滑道 DEP（起飞机场）_72，ARR（目的机场）_80，PHASE（飞行阶段）_5
(13)低空大速度 2500ft 以下 ARN（机号）_90 号机，ARR（目的机场）_67，ARN（机号）_94 号机
(14)坡度大 400/1500ft 以上 ARN（机号）_55 号机，ARN（机号）_36 号机，ARR（目的机场）_49
(15)复飞 ARR（目的机场）_93，ARN（机号）_95 号机，ARN（机号）_107 号机
(16)抬前轮速度大 ARN（机号）_28 号机，ARR（目的机场）_101，PHASE（飞行阶段）_2
(17)抬前轮速度小 ARR（目的机场）_98，ARR（目的机场）_44，PHASE（飞行阶段）_2
(18)抬头速率大 DEP（起飞机场）_16，ARN（机号）_69 号机，PHASE（飞行阶段）_2
(19)抬头速率小 ARN（机号）_84 号机，ARN（机号）_77 号机，PHASE（飞行阶段）_0
(20)接地速度小 ARR（目的机场）_9，DEP（起飞机场）_24，PHASE（飞行阶段）_6
(21)收反推晚 ARR（目的机场）_107，ARR（目的机场）_65，PHASE（飞行阶段）_7
(22)放起落架晚 ARR（目的机场）_4，DEP（起飞机场）_6，ARR（目的机场）_22
(23)未知 ARN（机号）_97 号机，ARN（机号）_22 号机，PHASE（飞行阶段）_0
(24)爬升坡度大 35-400(含)ft 月份_201507，ARN（机号）_18 号机，ARR（目的机场）_99
(25)爬升速度大 35-1000ft ARR（目的机场）_66，DEP（起飞机场）_115，PHASE（飞行阶段）_4
(26)爬升速度小 35-1000ft ARR（目的机场）_30，ARR（目的机场）_98，ARR（目的机场）_91

(27)直线滑行速度大 DEP（起飞机场）_49，ARN（机号）_95 号机，PHASE（飞行阶段）_1
(28)着陆俯仰角大 ARR（目的机场）_88，DEP（起飞机场）_83，PHASE（飞行阶段）_7
(29)着陆俯仰角小 ARR（目的机场）_31，ARR（目的机场）_91，ARR（目的机场）_25
(30)着陆垂直载荷大 ARR（目的机场）_75，ARR（目的机场）_82，PHASE（飞行阶段）_7
(31)着陆襟翼到位晚 ARN（机号）_68 号机，ARR（目的机场）_101，ARR（目的机场）_96
(32)着陆重、跳着陆 DEP（起飞机场）_45，PHASE（飞行阶段）_7，ARN（机号）_108 号机
(33)离地仰角大 ARN（机号）_82 号机，ARN（机号）_97 号机，ARN（机号）_57 号机
(34)离地速度大 ARR（目的机场）_101，ARR（目的机场）_63，PHASE（飞行阶段）_3
(35)离地速度小 ARR（目的机场）_99，ARN（机号）_84 号机，PHASE（飞行阶段）_3
(36)空中过载 DEP（起飞机场）_84，DEP（起飞机场）_48，ARR（目的机场）_9
(37)起飞 EGT 超限 ARN（机号）_49 号机，DEP（起飞机场）_50，ARR（目的机场）_4
(38)起飞坡度大 0-35（含）ft ARR（目的机场）_46，ARN（机号）_84 号机，ARN（机号）_111 号机
(39)起飞收起落架晚 DEP（起飞机场）_43，PHASE（飞行阶段）_0，ARN（机号）_93 号机
(40)起飞滑跑方向不稳定 ARN（机号）_60 号机，PHASE（飞行阶段）_2，DEP（起飞机场）_108
(41)超襟翼限制速度 Vfe ARR（目的机场）_23，ARN（机号）_66 号机，ARN（机号）_9 号机
(42)转弯滑行速度大 DEP（起飞机场）_1，ARR（目的机场）_83，PHASE（飞行阶段）_1
(43)进近坡度大 1500-500（含）ft ARN（机号）_85 号机，ARR（目的机场）_104，ARR（目的机场）_21
(44)进近坡度大 200-50（含）ft ARR（目的机场）_67，ARN（机号）_104 号机，ARR（目的机场）_31
(45)进近坡度大 500-200（含）ft ARR（目的机场）_69，ARR（目的机场）_54，ARR（目的机场）_56
(46）进近坡度大 50ft 以下 ARN（机号）_79 号机，ARR（目的机场）_25，PHASE（飞行阶段）_5
(47）进近速度大 50 ft 以下 ARN（机号）_83 号机，ARN（机号）_63 号机，PHASE（飞行阶段）_5

(48）进近速度大 500-50（含）ft　　DEP（起飞机场）_71，ARR（目的机场）_109，ARN（机号）_118 号机
(49）进近速度小 500 ft 以下　　月份_201505，ARN（机号）_42 号机，DEP（起飞机场）_86
（50）高于下滑道　　PHASE（飞行阶段）_0，ARN（机号）_27 号机，ARR（目的机场）_87

八、问题四的模型建立和求解

8.1 数据预处理

8.1.1 数据解读

根据附件 3 飞行参数测量数据可提取与飞行技术相关的特征。这些特征可以包括飞行速度、飞行高度、俯仰角、G 值等，以及其他可能对飞行技术评估有用的参数。可根据飞行参数评估飞行员的技术水平，可以通过飞行参数的特征选择，找出与飞行安全最相关的一些数据，，进而进行飞行技术评估。

8.1.2 数据处理

首先删除重复项和无关项，识别和处理缺失值，分析数据中的缺失值，用 matlab 软件进行处理，如果缺失值较少，可以考虑线性插值方法，如相邻时间点的平均值或中值；如果缺失值较多，可以考虑删除这个飞行参数。

8.2 领域粗糙集模型

领域粗糙集方法[11]是一种基于粗糙集理论的数据挖掘技术，适用于处理领域专业性强、数据不完整、不确定性较高的问题，在处理模糊、不确定性等信息方面具有优势，适用于数据挖掘、知识发现等领域，可以实现数值数据分类，分类效果非常好。提出了邻域半径的调整值，探讨了邻域半径对约简效果的影响。本文利用邻域粗糙集和前向贪婪算法简化了正域的计算。建立了随机森林登陆质量预测模型。然后从原始飞行参数、不同邻域方法和因子分析三个方面对变量进行比较。符号 Φ 表示完整集 U 的度量函数，N=⟨U，Φ⟩
表示邻域关系。f(xi，ak)表示样本元素 xi 的第 i 个属性值 n 表示域字段集中样本元素的总数。在二维空间中，上述三个距离函数的表达式图如图 16 示。

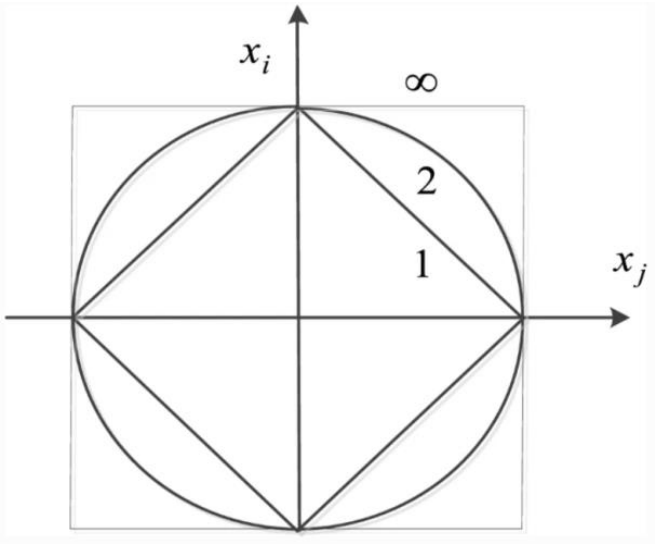

图 16 二维空间二分类问题示意图

图中展示了一个二维空间的二分类问题。每个样本对象都有其相应的邻域。假设有三个样本对象，分别表示为 x1、x2 和 x3。从图 2 中，我们可以看到 x1 的邻域位于类别 1（用“*”表示）的范围内，而 x3 的邻域位于类别 2（用“+”表示）的范围内。接着，确定 x1 和 x3 分别为类别 1 和类别 2 的下近似。然而，对于 x2，其对象邻域同时包含在类别 1 和类别 2 中，因此样本 x2 被定义为类别 1 和类别 2 之间的边界。图 3 对经典粗糙集模型进行了集合解释。通过一些分类属性将样本空间划分为信息粒子集合，每个网格表示具有相同特征值的对象的信息粒子。深色框区域代表正域，这意味着 X 的下近似值。深色框区域内的等价类完全属于 X。灯箱区域的一些信息粒子属于 X，而其他部分则不属于 X。浅色区域是 X 的边界区域。如图 17 示。

。

图 17 领域粗糙集示意图

邻域粗糙集属性简化在许多领域中得到了广泛应用。然而，计算每个样本的邻居依然效率较低。这是一种普遍存在但不可避免的邻域粗糙集模型简化方法。简化算法中的关键度量可能包括正域、不一致性和熵等。基于度量函数的候选属性子集构建策略

包括贪心搜索和启发式搜索。在所有策略方法中，贪心算法因其效率而被广泛采用。其中，前向贪心算法能够产生简化属性，由简化结果构建的规约器满足最小偏差原则。因此，贪心算法在机器学习领域得到了广泛应用。

本研究主要关注 47 个着陆阶段的 19 个飞行参数。首先，对飞行参数的缺失值和异常值进行了初步处理。删除了着陆后的数据。飞行高度范围为 9-2 米，每个着陆阶段的飞行参数以 0.5 米的间隔进行逐个切片。共访问了 15 个数据切片，并在本文中对这些数据进行了整合。最终，本研究获得了 47 个着陆阶段的 705 个数据和 15 个高度值。结果见表 1。需要注意的是，“飞行高度”数据源于无线电高度，因此无线电高度不再作为飞行属性进行计算。

表 2 飞行相关属性处理表

Order	Height	Distance	Vacuum speed	……	Side offset
1	9	228.8853	31.3176	……	- 0.3052
1	8.5	219.7299	31.2825	……	- 0.3052
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
1	2	21.3626	27.1504	……	7.9248
2	9	198.3672	31.6991	……	7.6296
2	8.5	186.1601	31.6640	……	7.6296
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
2	2	- 64.0879	26.9078	……	0
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
47	2	21.3626	28.3665	……	2.4415

本文采用邻域粗糙集理论来消除冗余。对飞行参数的原始数据进行归一化处理。标准化数据如表 2 所示

。

表 2 标准化数据处理结果表

Order	Height	C11	C22	……	C1919
1	9	0.7415	0.6217	……	0.7983

1	8.5	0.7330	0.6183	……	0.8067
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
1	2	0.1676	0.2162	……	0.9160
2	9	0.7417	0.6588	……	0.7275
2	8.5	0.7330	0.6554	……	0.7395
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
2	2	0.1364	0.1926	……	0.6975
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
47	2	0.3811	0.3480	……	0.3613

接下来是因子分析。首先计算了 19 个飞行参数的相关系数，发现部分变量之间存在较强的线性相关。例如，大气高度与真实航向的相关系数为-0.928，真空速度与俯仰角的相关系数为-0.669。变量之间的强相关性表明变量之间存在重叠和冗余。输入模型变量的冗余必然不利于数据分析，甚至会降低模型的输出。因此，有必要对 19 个飞行参数进行降维。因子分析是在产生较少飞行因子的情况下，尽量减少原始变量信息的损失，消除输入模型变量之间的相关性。对原 19 个飞行参数进行了 KMO 试验。KMO 测试值为 0.618，大于 0.6，说明原飞行参数适合进行因子分析。飞行参数因子分析结果如表 3 所示。

表 3 飞行参数因子分析结果表

Flight parameters	Original feature value		Extract the square load			
	Total	Variance ratio (%)	Cumulative ratio (%)	All	Variance ratio (%)	Cumulative ratio (%)
1	4.153	22.913	20.913	4.153	22.913	22.913
2	2.754	16.572	35.484	2.754	16.572	39.485
3	1.623	9.127	46.612	1.623	9.127	48.612
4	1.619	8.743	52.336	1.619	8.743	57.335
5	1.334	7.093	60.869	1.334	7.093	64.448
6	1.007	5.289	65.645	1.007	5.289	69.737

8.3 模型结果

根据上述分析结果，利用随机森林的模型进行预测，了解他们在不同技术级别之间的相对表现，观测其大致变化，然后，根据这些值计算总技术评分。如图 18 示。

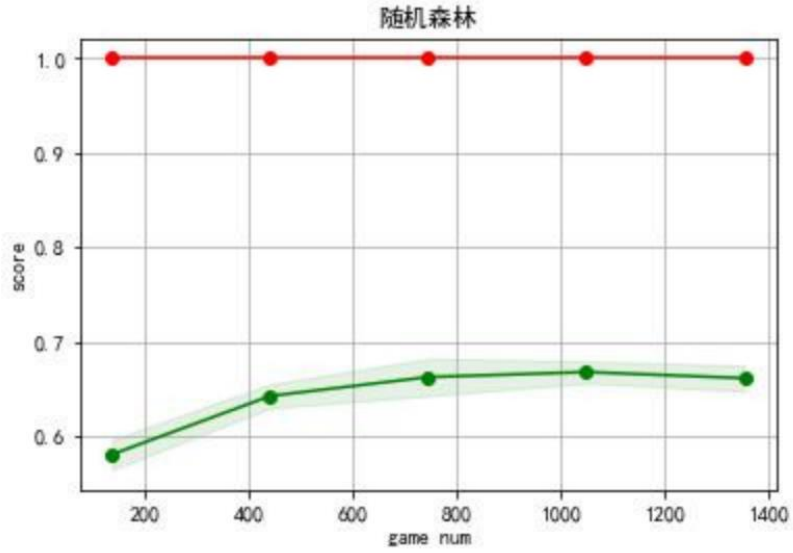

图 18 随机森林示意图

九、问题五的模型建立和求解

9.1 数据预处理

本题和问题二使用附件 3 中同一 QAR 数据，在此依然使用问题一中数据预处理后的数据结果。并根据处理后数据进行高互信息值数据表生成，如图 19 示。

```
# 提取关键列形成新的数据
data_keys = data[['FMF GROSS WEIGHT',
 'MAGNETIC HEADING',
 'ALTITUDE (1013)',
 'TRA-R.1',
 'TRA-R',
 'TRA-L',
 'TRA-L.1',
 'COMPUTED AIR SPD',
 'PITCH ATT_MEAN',
 'N1 SELTED-L']] # 列名1、列名2、列名3为您需要提取的关键列名称
# 查看新数据
data_keys.head()
```

	FMF GROSS WEIGHT	MAGNETIC HEADING	ALTITUDE (1013)	TRA-R.1	TRA-R	TRA-L	TRA-L.1	COMPUTED AIR SPD	PITCH ATT_MEAN	N1 SELTED-L
0	347680	178.2	135	34.0576	34.0576	34.0137	34.0137	30.0	-0.3516	21.4
1	347680	178.4	136	34.0576	34.0576	34.0137	34.0137	30.0	-0.3516	21.4
2	347680	178.6	136	34.0576	34.0576	33.9697	33.9697	30.0	-0.3516	21.4
3	347680	178.7	135	34.0576	34.0576	33.9697	33.9697	30.0	-0.3516	21.4
4	347680	178.7	136	34.0576	34.0576	33.9697	33.9697	30.0	-0.3516	21.4

图 19 高互信息值数据图

9.2 模型建立

这段代码通过计算了各列数据分位数，并定义了一个函数 check_value_in_range 用于判断数值是否属于区间。为了实现警告程度的区分，模型应用了不同的分位数划定了不同的区间以便细化警告程度，高等次的警告更不易触发。代码如图 20 示。

```
# 定义函数判断数值是否属于区间
def check_value_in_range(column, value, lower_percentile_1, upper_percentile_1,lower_percentile_2, upper_percentile_2):
    if value < lower_percentile_1 or value > upper_percentile_1:
        print("初级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内".format(column, value, lower_percentile_1, upper_percentile_1))
    if value < lower_percentile_1 or value > upper_percentile_2:
        print("高级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内".format(column, value, lower_percentile_2, upper_percentile_2))
```

图 20 模型代码图

9.3 模型结果

这段代码通过计算了各列数据分位数，并定义了一个函数 check_value_in_range 用于判断数值是否属于区间。为了实现警告程度的区分，模型应用了不同的分位数划定了不同的区间以便细化警告程度，高等次的警告更不易触发。代码如图 21 示。

```
# 定义函数判断数值是否属于区间
def check_value_in_range(column, value, lower_percentile_1, upper_percentile_1,lower_percentile_2, upper_percentile_2):
    if value < lower_percentile_1 or value > upper_percentile_1:
        print("初级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内".format(column, value, lower_percentile_1, upper_percentile_1))
    if value < lower_percentile_1 or value > upper_percentile_2:
        print("高级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内".format(column, value, lower_percentile_2, upper_percentile_2))
```

图 21 警告代码图

参考文献

[1]许杨鹏,喻亚宇,付文杰,陶圆,吴君怡,何倩,张超.Meta 分析中缺失标准差换算与标准化均数差估计方法简介[J].中国循证心血管医学杂志,2016,8(12):1412-1415.

[2]D.Freedman ,漠草.回归的均方根误差[J].数理统计与管理,1983(04):25-30.DOI:10.13860/j.cnki.sltj.1983.04.007.

[3]梁杰,陈嘉豪,张雪芹,周悦,林家骏.基于独热编码和卷积神经网络的异常检测[J].清华大学学报(自然科学版),2019,59(07):523-529.DOI:10.16511/j.cnki.qhdxxb.2018.25.061.

[4]杨寒雨,赵晓永,王磊.数据归一化方法综述[J].计算机工程与应用,2023,59(03):13-22.

[5]李凤英,许洪光,周方,李培.基于数据挖掘和 K-Means 算法的高校学情数据集成研究[J].黑龙江工程学院学报,2022,36(04):31-36.DOI:10.19352/j.cnki.issn1671-4679.2022.04.006.

[6]杜官明,孙瑾,高敬博.基于视觉的飞机操纵系统输入端位置测量技术[J].测控技术,2023,42(02):81-86+128.DOI:10.19708/j.ckjs.2022.02.218.

[7]朱晓东,鲁铁定,陈西江.正交多项式曲线拟合[J].东华理工大学学报(自然科学版),2010,33(04):398-400.

[8]李莉莉,靳士檑,周楷贺.基于岭回归模型大数据最优子抽样算法研究[J].系统科学与数学,2022,42(01):50-63.

[9]汪奇生. 线性回归模型的总体最小二乘平差算法及其应用研究[D].昆明理工大学,2014.

[10]王伟. 基于深度学习的纽扣表面缺陷视觉检测系统研究[D].南京航空航天大学,2021.DOI:10.27239/d.cnki.gnhhu.2021.000267.

[11]孙希进. 语义 Web 粗糙模型本体及其应用研究[D].大连海事大学,2012.

附录

附录 1

#Jupyter 代码

```
# 文件 1
import pandas as pd

# 读取 Excel 文件
df = pd.read_excel('D:\mathercup\OneData.xlsx')

# 显示前几行数据+-
df.head()

df.info()

# 缺失项填充并删除
df[' GEAR SELECT DOWN'].fillna('unknow', inplace=True)
df.to_excel('D:\mathercup\OneData2.xlsx', index=False)
df.drop_duplicates(inplace=True)#把重复行删掉覆盖原数据
df.info()

# 设置最大列宽为 None，使得所有列都完整显示
pd.set_option('display.max_columns', 60)
df.describe().T

# 删除名为 'DATE: MONTH' 的列
df.drop('DATE: MONTH', axis=1, inplace=True)
df.info()

df.index=range(df.shape[0])#恢复索引
df.head()

# 设置要保存的文件名和文件路径
file_path = "D:\mathercup\OneData3.xlsx"

# 使用 to_excel() 方法将数据保存到 Excel 文件中
df.to_excel(file_path, index=False)

# 文件 2

import pandas as pd

# 读取 Excel 文件
```

```
df = pd.read_excel('D:\mathercup\OneData3.xlsx')

# 显示前几行数据+-
df.head()

df3=df
df3

df3[' GEAR SELECT DOWN'] = df3[' GEAR SELECT DOWN'].replace({'DOWN':0, 'True': 1})
df3

df3['DEPARTURE AIRPORT'] = df3['DEPARTURE AIRPORT'].replace({'机场 68':68})
df3

df2=df
df2

df2['WOW INDICATE INAIR'] = df2['WOW INDICATE INAIR'].astype(int)
labels=df2['WOW INDICATE INAIR'].unique().tolist()#取不相同的数，转列表
labels
df2

df2['WOW INDICATE INAIR.1'] = df2['WOW INDICATE INAIR.1'].astype(int)
df2['WOW INDICATE INAIR.2'] = df2['WOW INDICATE INAIR.2'].astype(int)
df2['WOW INDICATE INAIR.3'] = df2['WOW INDICATE INAIR.3'].astype(int)
df2['WOW INDICATE INAIR.4'] = df2['WOW INDICATE INAIR.4'].astype(int)
df2

df2[' GEAR SELECT DOWN'] = df2[' GEAR SELECT DOWN'].replace({'DOWN':0, 'unknow': 1})
labels=df2[' GEAR SELECT DOWN'].unique().tolist()#取不相同的数，转列表
labels
df2

labels2=df2['DEPARTURE AIRPORT'].unique().tolist()#取不相同的数，转列表
labels2

df2['DEPARTURE AIRPORT'] = df2['DEPARTURE AIRPORT'].replace({'机场 68':68, '机场 117':117,
'机场 5':5, '机场 118':118, '机场 73':73})
labels3=df2['DEPARTURE AIRPORT'].unique().tolist()#取不相同的数，转列表
labels3
df2
```

```
df2['DESTINATION AIRPORT'] = df2['DESTINATION AIRPORT'].replace({'机场 68':68, '机场
117':117, '机场 5':5, '机场 118':118, '机场 73':73})
labels3=df2['DESTINATION AIRPORT'].unique().tolist()#取不相同的数，转列表
labels3
df2

# 设置要保存的文件名和文件路径
file_path = "D:\mathercup\OneData4.xlsx"

# 使用 to_excel() 方法将数据保存到 Excel 文件中
df2.to_excel(file_path, index=False)
#3.2.3 描述性统计处理异常值
df2.describe([0.01,0.1,0.25,.5,.75,.9,.99]).T

df4 = df2.drop('GMT', axis=1, inplace=False), #详细时间信息并非关键，舍去
df4

df2.describe([0.01,0.1,0.25,.5,.75,.9,.99]).T

# 指定需要计算标准差的五列数据
Cols_WOW=cols_WOW_INDICATE_INAIR = ['WOW INDICATE INAIR','WOW INDICATE
INAIR.1','WOW INDICATE INAIR.2','WOW INDICATE INAIR.3','WOW INDICATE
INAIR.4']
# 计算指定五列数据的标准差
df4['WOW INDICATE INAIR_STD'] = df4[Cols_WOW].std(axis=1)
df4['WOW INDICATE INAIR_MEAN'] =df[Cols_WOW].mean(axis=1)
df4.drop(Cols_WOW, axis=1, inplace=True)
# 显示结果
print(df4)

df4

Cols_COG= ['COG NORM ACCEL','COG NORM ACCEL.1','COG NORM ACCEL.2','COG NORM ACCEL.3','COG
NORM ACCEL.4',
           'COG NORM ACCEL.5','COG NORM ACCEL.6','COG NORM ACCEL.7','COG NORM
ACCEL.8','COG NORM ACCEL.9']
# 计算指定列数据的标准差
df4['COG NORM ACCEL_STD'] = df4[Cols_COG].std(axis=1)
df4['COG NORM ACCEL_MEAN'] =df4[Cols_COG].mean(axis=1)
df4.drop(Cols_COG, axis=1, inplace=True)
df4

# 指定要计算均方根的列
```

```
# 指定要计算均方根的列
import numpy as np
Cols_LOCALIZER= ['LOCALIZER DEV Dots (C)', 'LOCALIZER DEV Dots (L)', 'LOCALIZER DEV Dots
(R)']

# 计算每一行指定列的平方和的均值
df4['Mean_Squared_LOCALIZER'] = np.mean(df4[Cols_LOCALIZER]**2, axis=1)

# 计算均方根
df4['RMS_LOCALIZER'] = np.sqrt(df4['Mean_Squared_LOCALIZER'])
df4.drop(Cols_LOCALIZER, axis=1, inplace=True)
df4

df4.drop('Mean_Squared_LOCALIZER', axis=1, inplace=True)
df4

# 指定要计算均方根的列
import numpy as np
Cols_GLIDESLOPE= ['GLIDESLOPE DEV Dots (C)', 'GLIDESLOPE DEV Dots (L)', 'GLIDESLOPE DEV
Dots (R)']

# 计算每一行指定列的平方和的均值
df4['Mean_Squared_GLIDESLOPE'] = np.mean(df4[Cols_GLIDESLOPE]**2, axis=1)

# 计算均方根
df4['RMS_GLIDESLOPE'] = np.sqrt(df4['Mean_Squared_GLIDESLOPE'])
df4.drop(Cols_GLIDESLOPE, axis=1, inplace=True)
df4.drop('Mean_Squared_GLIDESLOPE', axis=1, inplace=True)
df4

Cols_PITCH= ['PITCH ATT','PITCH ATT.1','PITCH ATT.2','PITCH ATT.3','PITCH ATT.4']
# 计算指定列数据的标准差
df4['PITCH ATTL_STD'] = df4[Cols_PITCH].std(axis=1)
df4['PITCH ATT_MEAN'] =df4[Cols_PITCH].mean(axis=1)
df4.drop(Cols_PITCH, axis=1, inplace=True)
df4

Cols_CAP_CLM = ['CAP CLM 1 POSN','CAP CLM 1 POSN.1','CAP CLM 1 POSN.2','CAP CLM 1
POSN.3','CAP CLM 1 POSN.4']
# 计算指定列数据的标准差
df4['CAP CLM 1 POSN_STD'] = df4[Cols_CAP_CLM].std(axis=1)
```

```
df4['CAP CLM 1 POSN_MEAN'] =df4[Cols_CAP_CLM].mean(axis=1)
df4.drop(Cols_CAP_CLM, axis=1, inplace=True)
df4

Cols_CAP_WHL = ['CAP WHL 1 POSN','CAP WHL 1 POSN.1','CAP WHL 1 POSN.2','CAP WHL 1
POSN.3','CAP WHL 1 POSN.4']
# 计算指定列数据的标准差
df4['CAP WHL 1 POSN_STD'] = df4[Cols_CAP_WHL].std(axis=1)
df4['CAP WHL 1 POSN_MEAN'] =df4[Cols_CAP_WHL].mean(axis=1)
df4.drop(Cols_CAP_WHL, axis=1, inplace=True)
df4

# 设置要保存的文件名和文件路径
file_path = "D:\mathercup\OneData5.xlsx"

# 使用 to_excel() 方法将数据保存到 Excel 文件中
df4.to_excel(file_path, index=False)

df5=df4
df5

# 使用 get_dummies 方法进行独热编码
Month_Onehot = pd.get_dummies(df['DATE: DAY'], prefix='DATE: DAY')

# 将独热编码的结果添加到原数据中
df5 = pd.concat([df5, Month_Onehot], axis=1)

# 删除原始的月份列
df5 = df5.drop('DATE: DAY', axis=1)
df5

df5.describe([0.01,0.1,0.25,.5,.75,.9,.99]).T

# 使用 get_dummies 方法进行独热编码
DEPARTURE_AIRPORT_Onehot = pd.get_dummies(df['DEPARTURE AIRPORT'],
prefix='DEPARTURE_AIRPORT')

# 将独热编码的结果添加到原数据中
df5 = pd.concat([df5, DEPARTURE_AIRPORT_Onehot], axis=1)

# 删除原始的列
```

```
df5 = df5.drop('DEPARTURE AIRPORT', axis=1)

df5

# 使用 get_dummies 方法进行独热编码
DESTINATION_AIRPORT_Onehot = pd.get_dummies(df['DESTINATION AIRPORT'],
prefix='DESTINATION_AIRPORTT')

# 将独热编码的结果添加到原数据中
df5 = pd.concat([df5, DESTINATION_AIRPORT_Onehot], axis=1)

# 删除原始的列
df5 = df5.drop('DESTINATION AIRPORT', axis=1)
df5

# 检查数据类型
print(df5.dtypes)

# 将字符串类型的列转换为数值类型
#data['A'] = data['A'].astype(float)

df5['A/T ENGAGED'] = df5['A/T ENGAGED'].replace({'DISENGD':0, 'ENGAGED': 1})
labels=df5['A/T ENGAGED'].unique().tolist()#取不相同的数，转列表
labels

df5['ANY A/P ENGAGED'] = df5['ANY A/P ENGAGED'].replace({'OFF':0, 'ON': 1})
labels=df5['ANY A/P ENGAGED'].unique().tolist()#取不相同的数，转列表
labels

print(df5.dtypes)

# 使用最小-最大缩放对数据进行归一化
min_val = df5.min()
max_val = df5.max()
normalized_df5 = (df5 - min_val) / (max_val - min_val)

# 打印归一化后的数据
print("归一化后的数据：")
normalized_df5

# 假设你的处理后的数据存储在名为 processed_data 的 DataFrame 对象中
# 设置要保存的文件名和文件路径
file_path = "D:\mathercup\OneData6.xlsx"
```

```
# 使用 to_excel() 方法将数据保存到 Excel 文件中

df5.to_excel(file_path, index=False)

# 文件3
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import chi2
import numpy as np
import pandas as pd

# 读取 Excel 文件
df6 = pd.read_excel('D:\mathercup\OneData6.xlsx')

# 显示前几行数据+-
df6.head()

kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(df6)

labels = kmeans.labels_
labels

mutual_info_scores = mutual_info_classif(df6, labels)

for i in range(df6.shape[1]):
    print("Feature {} Mutual Information: {:.4f}".format(i, mutual_info_scores[i]))

# 排序并选择前 10 个互信息值最大的特征
selected_features_indices = sorted(range(df6.shape[1]), key=lambda i:
mutual_info_scores[i], reverse=True)[:10]

# 输出选择的特征索引
print("Selected Features Indices: ", selected_features_indices)

# 获取特征名称列表
feature_names = df6.columns.tolist()
print("Feature names from file: ", feature_names)

mutual_info_dict = {feature_names[i]: mutual_info_scores[i] for i in
selected_features_indices}
```

mutual_info_dict

附件 2

```
% 提取杆量数据
stick_position = OneData(:, 1:5); % 操纵杆位置数据，假设在第 1 列到第 5 列

% 滑动窗口参数
window_size = 5; % 滑动窗口大小，表示每秒记录的次数
stride = 1; % 滑动窗口的步长

% 初始化趋势线系数矩阵
order = 1; % 多项式拟合的阶数，这里选择 1 表示线性拟合
coeffs = zeros(order + 1, size(stick_position, 2)); % 初始化系数矩阵

% 滑动窗口趋势分析
for i = 1:size(stick_position, 2)
    for j = 1:stride:(size(stick_position, 1) - window_size + 1)
        window_data = stick_position(j:(j + window_size - 1), i); % 提取滑动窗口内的数据
        time = (j:(j + window_size - 1)) / window_size; % 时间，假设每秒记录 5 次
        coeffs(:, i) = coeffs(:, i) + polyfit(time, window_data, order)'; % 对滑动窗口内的数据进行多项式拟合并累加系数
    end
    coeffs(:, i) = coeffs(:, i) / ((size(stick_position, 1) - window_size + 1) / stride); % 求平均系数
end

% 数据可视化
figure;
plot(stick_position);
xlabel('样本索引');
ylabel('操纵杆位置');
title('操纵杆位置趋势分析');

% 绘制趋势线
hold on;
for i = 1:size(stick_position, 2)
    time = (1:size(stick_position, 1)) / window_size; % 计算时间
    y_fit = polyval(coeffs(:, i), time); % 计算拟合后的值
    plot(time, y_fit, 'r'); % 绘制趋势线
end
legend('杆量 1', '杆量 2', '杆量 3', '杆量 4', '杆量 5', '趋势线');
```

附件 3

```
#文件一
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']  # 设置使用 SimHei 字体，可以根据实际安装的字体进行设置

import pandas as pd

# 读取 Excel 文件
df = pd.read_excel('D:\mathercup\TwoData.xlsx')

# 显示前几行数据+-
df.head()

labels=df['EVENT_NAME(超限名称)'].unique().tolist()#取不相同的数，转列表
labels

labels=df['ARN（机号）'].unique().tolist()#取不相同的数，转列表
labels

labels=df['DEP（起飞机场）'].unique().tolist()#取不相同的数，转列表
labels

df.info()

import pandas as pd

# 将指定的一列数据转换为独热编码
column_name = 'ARN（机号）'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列

df

column_name = 'PHASE（飞行阶段）'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列
df
```

```
column_name = '月份'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列

column_name = 'ARR（目的机场）'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列

column_name = 'DEP（起飞机场）'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列

column_name = 'ALERT（警告级别）'  # 需要转换的列名
df_one_hot = pd.get_dummies(df[column_name], prefix=column_name)  # 进行独热编码
df = pd.concat([df, df_one_hot], axis=1)  # 将转换后的独热编码数据合并到原始数据框中
df.drop(column_name, axis=1, inplace=True)  # 删除原始数据列

df

# 设置要保存的文件名和文件路径
file_path = "D:\mathercup\TwoData2.xlsx"

# 使用 to_excel() 方法将数据保存到 Excel 文件中
df.to_excel(file_path, index=False)
```

#文件二

```
import pandas as pd

# 读取 Excel 文件
df = pd.read_excel('D:\mathercup\TwoData2.xlsx')

# 显示前几行数据+-
df.head()

data=df

data=data.drop(columns=['ALERT（警告级别）_2', 'ALERT（警告级别）_3'])
data

import pandas as pd
```

```
from sklearn.linear_model import LinearRegression


# 对标签列进行独热编码
labels = data.iloc[:, 0]
labels_encoded = pd.get_dummies(labels)

# 提取其他 n 列数据作为特征
n = data.shape[1] - 1  # 计算特征列数
features = data.iloc[:, 1:]  # 提取特征数据

# 执行多元线性回归
mdl = LinearRegression()
mdl.fit(features, labels_encoded)  # 使用 LinearRegression 进行多元线性回归

# 提取回归模型的回归系数
coefs = mdl.coef_

# 初始化存储结果的字典
top5_cols = {}

# 遍历每个标签，找出对应的前三列影响最大的因子
for i, label in enumerate(labels_encoded.columns):
    idx = coefs[i].argsort()[-3:]  # 找出最大的三个系数的索引
    top5_cols[label] = features.columns[idx]  # 存储对应的列名称

# 输出结果
print('标签\t\t前三列名称')
for label, cols in top5_cols.items():
    print(f'{label}\t\t{", ".join(cols)}')

import pandas as pd
from sklearn.linear_model import Ridge

# 对标签列进行独热编码
labels = data.iloc[:, 0]
labels_encoded = pd.get_dummies(labels)

# 提取其他 n 列数据作为特征
n = data.shape[1] - 1  # 计算特征列数
features = data.iloc[:, 1:]  # 提取特征数据

# 执行岭回归
```

```
alpha = 1.0  # 设置岭回归的超参数 alpha
mdl = Ridge(alpha=alpha)
mdl.fit(features, labels_encoded)  # 使用 Ridge 进行岭回归

# 提取回归模型的回归系数
coefs = mdl.coef_

# 初始化存储结果的字典
top5_cols = {}

# 遍历每个标签，找出对应的前三列影响最大的因子
for i, label in enumerate(labels_encoded.columns):
    idx = coefs[i].argsort()[-3:]  # 找出最大的三个系数的索引
    top5_cols[label] = features.columns[idx]  # 存储对应的列名称

# 输出结果
print('标签\t\t前三列名称')
for label, cols in top5_cols.items():
    print(f'{label}\t\t{", ".join(cols)}')

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import Ridge

X=features
y=labels_encoded
# 将数据集划分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建岭回归模型
ridge_model = Ridge(alpha=1.0)

# 训练岭回归模型
ridge_model.fit(X_train, y_train)

# 使用交叉验证对模型进行评估
cv_scores = cross_val_score(ridge_model, X_train, y_train, cv=5,
scoring='neg_mean_squared_error')

# 将负均方误差转换为正值
mse_scores = -cv_scores
print("岭回归模型的每折交叉验证的均方误差,平均均方误差")
# 输出每折交叉验证的均方误差
print("Mean Squared Error for each fold:", mse_scores)
```

```
# 输出平均均方误差
print("Mean Squared Error (MSE):", mse_scores.mean())

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error


X=features
y=labels_encoded

# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建线性回归模型
lr_model = LinearRegression()

# 在训练集上拟合线性回归模型
lr_model.fit(X_train, y_train)

# 进行5折交叉验证，计算均方误差
mse_scores = -cross_val_score(lr_model, X_train, y_train, cv=5,
scoring='neg_mean_squared_error')
print("线性回归模型的每折交叉验证的均方误差,平均均方误差")
# 输出每折交叉验证的均方误差
print("Mean Squared Error for each fold:", mse_scores)

# 计算平均均方误差
mse_mean = mse_scores.mean()
print("Mean Squared Error (MSE):", mse_mean)
```

附件 4

使用 matlab 软件编程：

```
clc
clear
close all
D=xlsread('1.xlsx','Sheet2','B2:G7');%原始数据
k=6;
x0=D(:,k)';%代表 xk 的已有数据
n=length(x0);
%累加数据
x1=zeros(1,n);
x1(1)=x0(1);
for i= 2:n
x1(i)=x0(i)+x1(i-1);
end
%进行光滑，权值取 0-1
af=8;
z1=zeros(1,n);
z1(1)=0;
for i=2:n
z1(i)=x1(i)*af+(1-af)*x1(i-1);
end
%构造方程矩阵，X 为 Z1(K)与 1 组成的 2 列矩阵，Y 为一阶列矩阵，B 为 a,b 的解矩
阵。
%有最小二乘法可以知道，最佳解围 B=（X(XT)）^(-1)(XT)Y
Y=zeros(n-1,1);X=zeros(n-1,2);
for i=2:n
Y(i-1)=x0(i);
X(i-1)=-z1(i);
X(i-1,2)=1;
end
%求解参数矩阵，inv 为求逆运算
B=inv(X'*X)*X'*Y;
a=B(1);
b=B(2);
%求解 n+1 年的式子为 x0(n+1)=(x0(1)-b/a)*exp(-a*n)*(1-e^a);
pred_n_1=(x0(1)-b/a)*exp(-a*n)*(1-exp(a));
```

附件 5

```
import pandas as pd
# 加载原始数据 df6 = pd.read_excel('D:\mathercup\OneData6.xlsx')
data=df6
data_full=datadata_full
# 提取关键列形成新的数据 data_keys = data[['FMF GROSS WEIGHT',
 'MAGNETIC HEADING',
 'ALTITUDE (1013)',
 'TRA-R.1',
 'TRA-R',
 'TRA-L',
 'TRA-L.1',
 'COMPUTED AIR SPD',
 'PITCH ATT_MEAN',
 'N1 SELTED-L']]
# 列名 1、列名 2、列名 3 为您需要提取的关键列名称# 查看新数据
data_keys.head()

# 通过往常数据确定数据正常区间，通过判断数据是否属于正常区间，实现实时自动化预警机制
# 根据偏离区间不同，来发送等级不同的预警
import pandas as pd
import numpy as np


# 计算各列数据的 0.01%和 99.99%分位数
lower_percentiles_1 = data_keys.quantile(0.0001)
upper_percentiles_1 = data_keys.quantile(0.9999)

lower_percentiles_2 = data_keys.quantile(0.00001)
upper_percentiles_2 = data_keys.quantile(0.99999)

# 定义函数判断数值是否属于区间
def check_value_in_range(column, value, lower_percentile_1,
upper_percentile_1,lower_percentile_2, upper_percentile_2):
    if value < lower_percentile_1 or value > upper_percentile_1:
        print("初级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内
".format(column, value, lower_percentile_1, upper_percentile_1))
    if value < lower_percentile_1 or value > upper_percentile_2:
        print("高级警告：列名 {} 中的值 {} 不在区间 [{}, {}] 内
".format(column, value, lower_percentile_2, upper_percentile_2))
```

```
# 遍历各列数据并判断数值是否属于区间
for column in data_keys.columns:
    for value in data_keys[column]:
        lower_1 = lower_percentiles_1[column]
        upper_1 = upper_percentiles_1[column]
        lower_2 = lower_percentiles_2[column]
        upper_2 = upper_percentiles_2[column]
        check_value_in_range(column,  value,  lower_1,  upper_1,lower_2,
upper_2)

# 将实时数据导入判断函数，即可实现实时自动化预警机制，预防可能的安全事故发
生
```