智能化生产中的故障预测和识别与人员排班配置优化

摘 要

随着信息技术的飞速发展和广泛应用，智能化、自动化的工业生产线已成为制造业的新常态。这些生产线能够独立完成物品搬运填充、产品包装及检测等多个环节，无需人工干预，极大地提高了生产效率和产品质量，同时降低了生产成本。通过引入故障智能报警系统，实时发现并上报故障问题，有效预防设备故障引发的生产停滞和经济损失。再结合合理的人员配置，进一步优化了资源利用，确保生产过程的高效稳定。

对于问题1、2，本文以某十条生产线的全年机器运行数据为基础，对各种故障的出现频次、持续时长进行了深入分析，并考察了故障发生时生产线的具体状况，为了强化故障数据的特征，以便模型能够更有效地进行学习，先进行样本数据均衡，再采用了过采样方法对数据进行处理。通过多种机器学习算法进行综合评估，决定采用XGBoost模型来对故障的发生概率及其持续时间进行预测。同时使用遗传算法GA对模型的超参数选择进行优化。经过一系列的训练和优化，最终成功构建了一个故障自动报警模型。该模型已针对M201和M202两条生产线的数据进行了故障预测的应用求解。

对于问题3，基于数据计算绩效指标，采用多独立样本Kruskal-Wallis检验方法检验不同生产线以及不同工龄对各种绩效指标是否存在差异性。发现不同生产线的机器总运行时间、综合故障率均无显著差异，但在其他因素上存在显著差异，推断差异性是基于工龄的影响，后续证实不同工龄与部分指标的确存在显著差异。结合多种机器学习模型来评估工龄对各种指标的影响程度，选择决策树模型进行SHAP分析。最后证实，工龄越大的操作人员，拥有更丰富的生产经验和技能，让生产线中各种工作效率的提升进而带动产量和合格率的提升。

对于问题4，针对扩大后的生产规模，开发了一个最大化生产效率的规划模型，考虑多个约束条件，使用模拟退火算法求解，得到一个较优的班次计划。

最后，对本文涉及到的模型进行总体评价以及合理的领域推广。

关键词：故障预测 人员排班优化 差异性分析 GA-XGBoost**模型** SHAP分析

目录

一、问题重述.....3
二、问题分析.....4
三、模型假设.....7
四、故障预测与识别模型的建立和应用求解.....8
4.1 数据分析与数据清洗聚合.....8
4.2 定义相关评估指标对模型的预测能力进行评估分析.....10
4.3 基于GA-XGBoost模型的建立和评估.....11
4.4 基于GA-XGBoost模型的应用求解.....16
五、产品生产绩效的相关性分析.....18
5.1 数据分析以及相关绩效指标计算.....18
5.2 差异性分析.....20
5.3 机器学习模型求解影响程度.....26
六、人员排班配置优化规划模型分析建立与求解.....32
6.1 人员排班配置优化规划模型的建立.....32
6.2 基于模拟退火算法的人员出勤排班规划模型优化求解.....34
七、模型整体评价.....37
7.1 模型优势.....38
7.2 模型不足.....38
7.3 模型推广.....39
八、参考文献.....40
附录.....41

一、问题重述

随着信息技术的快速发展，智能化控制生产技术在产品生产领域的应用正变得日益成熟。工厂生产线能够独立完成物品的搬运、物料的填充、产品的包装以及产品质量的检测等多个环节，无需人工干预，极大地提高了生产效率和产品质量，同时降低了生产成本。除了技术层面的智能化，故障智能报警技术的应用也是至关重要的。通过在设备和生产线上安装传感器，实时监测设备运行状态，及时发现异常并报警，可以最大程度地减少因设备故障导致的生产中断。这种预防性维护不仅可以降低维修成本，还能提高设备的可靠性和稳定性，确保生产中心的持续高效运转。

在人力资源管理方面，合理配置人员同样至关重要。分拨中心的运营往往需要多个岗位的协同配合，因此需要根据不同生产线的需求，合理安排员工的岗位和工作时间。特别是在多班次、长时间运行的情况下，合理的轮班制度不仅可以保证员工的健康和工作积极性，还能最大限度地利用人力资源，提高分拨中心的运作效率。

因此，本文利用历史数据进行深度分析并以此构建预测模型来预测生产线的对故障的发生概率及其持续时间，同时分析数据从而制定高效的人员排班策略，确保智能化生产线的生产效率以及优化智能化生产线的管理。本文需要解决的问题大致如下：

1. 构建故障报警模型：根据某工厂十条生产线的全年机器运行数据记录，分析生产线中各装置故障的数据特征和影响因素，构建故障报警模型，实现故障的自动即时报警。

2. 应用模型预测和识别故障：根据建立好的预测模型，对题目中给出的新的数据集（两条生产线一年的运行记录数据，但不包含各装置的故障信息）进行模型测试，根据已有的自变量即生产线各装置的状态，进行故障的自动识别预测并报警故障，同时记录故障的具体时间和持续时长，并汇总相关信息。

3. 分析影响产品产量和合格率的因素：根据题目中的给出的10 条生产线一年的运行记录数据以及各生产线操作人员的信息，分析产品的产量、合格率与

生产线、操作人员等各种因素的关系，

4. 人员排班配置优化：由于需要扩大生产规模，现在将生产线每天的运行时间从8小时增加到24小时不间断生产。基于问题3的10条生产线和人员比例，结合问题 3 的分析结果，考虑生产线与操作人员的搭配，制定最佳的操作人员排班方案即求解一个目标规划函数问题。

二、问题分析

（1）问题1、2：历史机器运行数据预处理和构建故障预测模型应用求解

为了确保智能化制造产业线的高效运作，本文旨在建立一个能够基于当前机器各方面状态准确预测故障并自动识别故障类型的预测模型。该模型的预测准确性对于指导资源的合理分配和调度至关重要。在模型建立过程中，需要综合考虑推送装置、检测装置等各种装置自身的因素的影响。

鉴于数据量的庞大，故障数据在常规运行数据中占比较小，而模型需具备捕捉分析各因素与不同故障状态的相关性的能力，对此我对数据进行了过采样SMOTE处理，使得模型训练时可以更加公平地学习到正常运行与故障状态之间的差异，同时减少模型过拟合的风险。采取了多种机器学习模型来尝试捕捉数据变量之间的长期依赖关系，经过一系列评估指标的对比后，最后选择XGBoost（极度梯度提升树）模型作为本次问题的解决模型，在模型训练和验证之后，将其应用于M201和M202两条生产线的新数据集，进行故障的预测。这包括预测故障的发生概率、类型及其可能的持续时间。根据预测结果，实时生成故障报警，以便生产线管理人员可以及时采取相应的预防或修复措施，为智能化制造产业线的高效运作提供有力的支持。

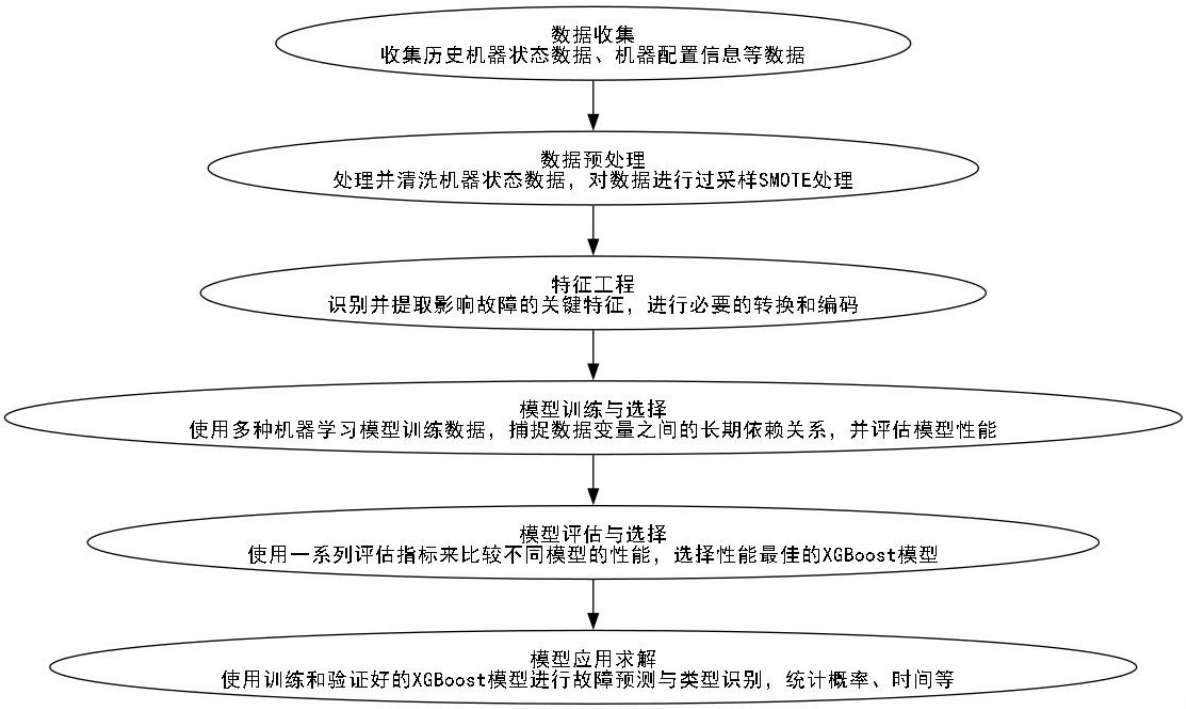

图1：问题1、2的分析求解流程

（2）问题3：分析生产线以及操作人员对产品产量和合格率的影响

在这个问题中，题目要求根据另一组十条生产线1年内的运行数据，来分析不同生产线以及操作人员即工龄对产品产量以及合格率的影响。

首先需要对附件3中提供的生产线运行记录进行详细的数据清洗和预处理，由于初始数据过于庞大且类别繁多，为了能够精确的分析各种因素之间的影响，我引入了一些基于原有数据计算出的绩效指标例如总运行时间、总产量、平均生产效率、填装效率、加盖效率等，基于每一个独立的日期进行数据处理，再聚合10条生产线的所有数据，构造出了一个新的数据集。预处理后的数据还将包括操作人员的详细信息，如班次、工龄等，这些都可能对生产效率和产品质量有直接或间接的影响。针对不同生产线，使用多独立样本Kruskal-Wallis检验方法进行差异性分析，来分析不同生产线下各种绩效指标是否存在差异性。由于在分类处理好所给的数据后，操作人员的有效相关因素有且仅有工龄，使用多独立样本Kruskal-Wallis检验方法分析不同工龄的工作人员产生的不同的绩效指标是否存在差异性。最后结合多种机器学习算法模型来评估工龄对各种绩效指标的影响程度。经过综合评估模型性能，选择性能较好决策树模型进行整体的分析。

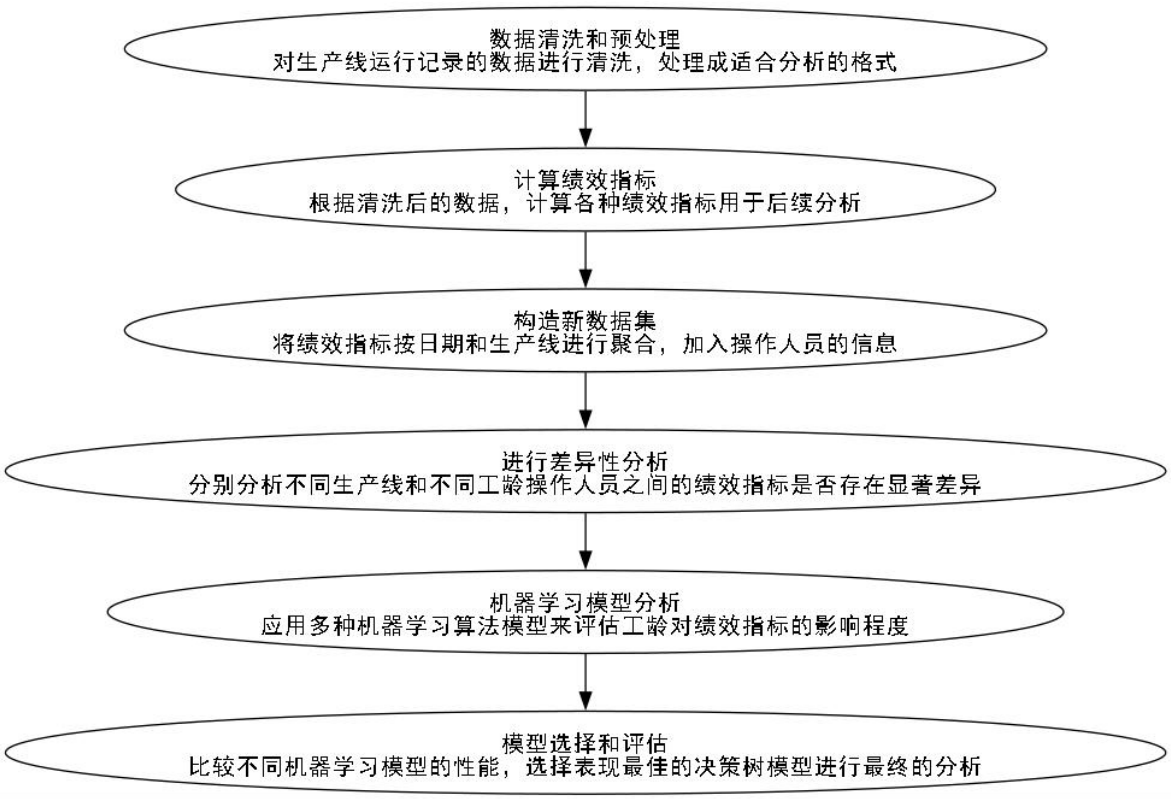

图2：问题3分析求解流程

（3）问题4：最大化生产效率的人员排班配置优化问题求解

问题4旨在针对扩大的生产规模即生产线运行时间从每天8小时增加到24小时不间断生产的情况后的生产线，为其设计最优的操作人员排班方案来精细管理人力资源以确保高效率的生产和员工的工作生活平衡。

首先需要分析现有的操作人员配置情况，包括每个员工的工龄、技能等级、历史表现和工作偏好。在问题3中，由于对数据进行了绩效指标的计算和数据聚合，基于此，我专门提取计算并单独保存了不同工龄所管理的生产线的产品合格率和总产量，这些信息是制定合理排班计划的基础。根据生产规模的扩大，确定需要的总操作人员数，经过计算，每周需要配置42名操作人员以覆盖三班制的全天候运行。定义目标函数为最大化生产效益同时考虑员工自身的各种因素。同时确立多种约束条件，包括员工的工作天数限制、每个班次的人数限制、休息需求以及生产需求的波动等。使用线性规划模型来制定排班计划，应用求解器来找到满足所有约束条件的最优解。

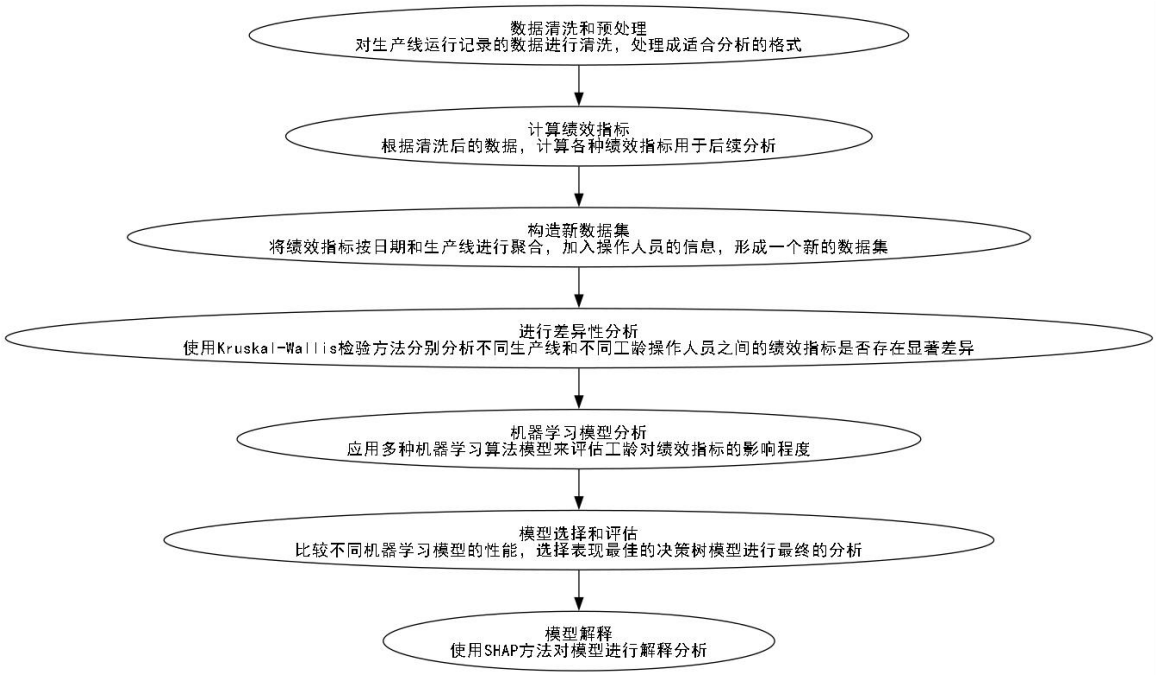

图3：问题4求解和分析过程

三、模型假设

在构建模型时，首先需要明确一些关键假设，这些假设将指导模型的设计和应用，同时防止模型建立求解或评估时被其他外界因素干扰，在此做出以下假设：

假设1: 数据中记录的故障信息是完整且准确的，可以直接用于进一步的分析和模型训练。

假设2: 用来训练模型的数据足够代表未来的运行状况，即经过处理后的训练数据与未来数据在统计特征上保持一致。

假设3: 故障的发生是随机性的，且故障数据在数据集中的分布是均匀的，不存在时间序列上的偏差。

假设4: 故障的定义和分类在各条生产线上是统一的，不存在因技术升级或操作差异而改变的可能性。

假设5: 影响产量和合格率的主要因素包括设备性能、操作人员的技能和工龄等，其他外部因素（如原材料质量、机器品牌）的影响均忽略不计。

假设6: 所有生产线的操作条件（环境温度、湿度等）保持一致，不会对产

量和合格率产生差异性影响。

假设7: 数据中记录的每个操作人员的信息是准确和完整的，能够用于分析其对生产效率和产品质量的影响。

假设8: 扩大生产规模并转为24小时不间断生产后，每个班次的工作强度和人员需求是稳定的。

假设9: 操作人员都愿意接受新的排班方案，不存在因个人偏好或外部约束影响排班的情况。

假设10: 延长工作时间不会导致操作人员的健康和效率显著下降。

假设11：大部分数据的一年数据为362天，在此以30天作为1个月记录，最后两天归于最后一个月即12月。

四、故障预测与识别模型的建立和应用求解

4.1 数据分析与数据清洗聚合

首先进行数据的可视化，由于包含多条生产线的数据，在本小节中的选取M101这条生产线作为示例进行说明，如下图所示：

	日期	时间	生产线编号	物料推送气缸推送状态	物料推送气缸收回状态	物料推送数	物料待抓取数	放置容器数	容器上传检测数	填装检测数	...	不合格数	物料推送装置故障1001	物料检测装置故障2001	填装装置检测故障4001	填装装置定位故障4002	填装装置填装故障4003	加盖装置定位故障5001	加盖装置加盖故障5002	拧盖装置定位故障6001	拧盖装置拧盖故障6002
0	1	0	M101	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	1	M101	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	2	M101	0	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1	3	M101	1	0	2	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	1	4	M101	0	1	2	1	1	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7528788	362	28866	M101	0	1	4332	4332	1446	1445	1444	...	0	0	0	0	0	0	0	0	0	0
7528789	362	28867	M101	0	1	4332	4332	1446	1445	1444	...	0	0	0	0	0	0	0	0	0	0
7528790	362	28868	M101	0	1	4332	4332	1446	1445	1444	...	0	0	0	0	0	0	0	0	0	0
7528791	362	28869	M101	0	1	4332	4332	1446	1445	1444	...	0	0	0	0	0	0	0	0	0	0
7528792	362	28870	M101	0	1	4332	4332	1446	1445	1444	...	0	0	0	0	0	0	0	0	0	0

7528793 rows × 37 columns

图4：M101生产线运行数据部分展示

数据记录的是一年内每一个工作日期的对应的工作时间内的每秒机器状态，其中最后9列表示的是9类不同的故障在对应的时间节点是否发生，若为0则

为故障，为1则表示发生这类故障。

在对这10份数据集的观察分析中，不难发现故障数据和非故障数据的数据量之间差异过大，如在M101生产线运行数据中，所有类型的故障的非故障数据量基本在区间[7523000,7525000]之间波动，但是故障数据只有几千条，仅在区间[2600,7700]内波动，数据量之间的差异过大，达到了几千倍的规模。当数据集中一个类别的样本远多于另一个类别时，模型可能会偏向于样本数量多的类别，导致在预测少数类别的样本时性能不佳。

为了后续模型的训练和评估，我将10条生产线的数据聚合为一个新的数据集，同时考虑到数据量的差异。在数据聚合的过程中，首先针对每条生产线的每一种故障类型，在选取所有正样本即该类型故障数据的同时，随机选择相同数量的负样本即该类型非故障数据进行聚合。在每一条生产线的数据处理提取完成后，再次采用SMOTE过采样方法（Synthetic Minority Over-sampling Technique）[1]平衡数据，增加少数类别的样本数量。下面是SMOTE方法的简要解释：

SMOTE是一种更为复杂的上采样方法，它通过在少数类样本之间插值来合成新的样本。主要步骤如下：

1. 选择样本：对于每一个少数类样本，从其最近邻中随机选择一个样本。
2. 合成新样本：对于选择的两个样本，计算它们的特征差，并乘以一个随机数 （0到1之间），然后加到原始样本的特征上。

合成新样本的公式为：

$$x_{new} = x_i + \lambda \cdot (x_z - x_i)$$

其中：x_i 是原始少数类样本，x_z 是从最近邻中选出的样本，x_{new}是生成的新样本。λ是介于0和1之间的随机数，用于控制新样本与原始样本之间的相似度。

最后聚合10组经过随机筛选和SMOTE过采样后的生产线数据构成本次模型训练和评估的数据集。下表为处理后的故障数据分布情况展示，为了精简显示，不具体说明故障信息，仅以故障类型的编号作为故障的分类标准，如物料推送装置故障1001记为故障1001。

表1：聚合数据集的故障数据分布情况

故障类型	故障数据量	非故障数据量
故障1001	37008	475518
故障2001	23254	489272
故障4001	28062	484464
故障4002	26371	486155
故障4003	27020	485506
故障5001	25892	486634
故障5002	34125	478401
故障6001	25962	486564
故障6002	28998	483528

4.2 定义相关评估指标对模型的预测能力进行评估分析

在模型训练完成后，需要对模型的预测效果进行评估。为了衡量模型的性能和预测情况，我们通常会使用一些统计指标来对比预测值和实际观测值。在本文中，选用了准确率（ACC，Accuracy）、召回率（REC, Recall）和F1（F1_Score）作为评估指标。它们的原理解释和基本公式如下：

1. 准确率（ACC）:准确率是指模型正确预测的所有样本（包括正类和负类）占所有预测样本的比例。它是分类问题中最直观的评价指标，但并不总是最佳指标，尤其是在数据集不平衡的情况下。公式：

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

其中，TP（True Positive）表示正确预测的正类样本数量，TN（True Negative）表示正确预测的负类样本数量，FP（False Positive）表示错误预测为正类的负类样本数量，FN（False Negative）表示错误预测为负类的正类样本数量，下同。

2. 召回率（REC）：召回率（也称为真正类率或灵敏度）是指模型正确预测的正类样本占所有实际正类样本的比例。召回率关注的是模型能否找到所有的正类样本。公式：

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

3. F1_Score （F1）：F1分数是准确率和召回率的调和平均数，它同时考虑了模型预测的准确性和完整性。F1分数在准确率和召回率之间取得了平衡，当两者都很高时，F1分数也会很高。公式：

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

其中，Precision（精确率）是指模型正确预测的正类样本占所有预测为正类样本的比例，公式为：

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

4.3 基于GA-XGBoost模型的建立和评估

在本次问题中，为了选择性能相对较好的回归模型，通过多种机器学习模型，包括逻辑回归模型、XGBoost极度梯度提升树回归模型、决策树回归模型、随机森林回归模型、Adaboost模型、GBDT梯度提升树回归模型[2-7]，在结合4.2中的指标进行总体评估后，在此不做过多展示，最后选择效果较好的XGBoost极度梯度提升树回归模型作为本次的问题求解模型。

另外，为了节省计算时间，本次模型训练针对每一种故障类型的数据分别进行模型的训练测试，以便后期针对不同的故障进行模型调用预测

4.3.1 XGBoost模型建立

XGBoost是一个优化的分布式梯度增强库，旨在提供高效、灵活且可扩展的梯度提升框架。它是由 Tianqi Chen 在2014年开发的，受到他在卡内基梅隆大学（CMU）的实习经历和他在华盛顿大学的博士研究的影响,在研究机器学习算法的效率和可扩展性时，为了解决 Gradient Boosting 实现中的计算瓶颈和扩展性问题而创建的[8]。XGBoost的核心算法是基于决策树的梯度提升框架。它通过不断地添加新的决策树来拟合上一次迭代的残差，最终将所有决策树的结果加权求和得到最终的预测结果。每棵决策树都是在一个新的子空间上学习，这样可以捕捉到数据中的复杂关系。XGBoost通过在每次迭代时优化目标函数，

不断添加新的决策树来构建一个强学习器。它的优势在于能够自动处理缺失值，支持并行计算，提供了多种防止过拟合的技术，并且可以通过交叉验证来调整模型参数。下面是其计算流程以及相关公式的简单描述：

1. **初始化**：设定初始预测值 $F_0(x)$。在回归问题中，通常设为训练集标签的均值。
2. **迭代构建弱学习器**：（对于 $m = 1,2,\ldots,M$）：
   - a. 计算残差：对于每个样本 ii，计算残差r_{mi}：
     $$r_{mi} = y_i - F_{m-1}(x_i)$$
     其中 y_i是真实标签，$F_{m-1}(x_i)$ 是上一轮的预测值。
   - b. 训练弱学习器：使用残差r_{mi}作为标签，训练一个新的决策树 $h_m(x)$，用于拟合残差。
   - c. 更新预测：$F_m(x) = F_{m-1}(x) + \eta h_m(x)$，其中 η是学习率。
3. **计算损失函数**：模型使用的是损失函数的泰勒二阶展开来优化模型。对于回归问题，常用的损失函数是平方损失
   $$L\big(y, F(x)\big) = \big(y - F(x)\big)^2$$
   其泰勒展开为：
   $$L\big(y, F(x)\big) = \big(y - F(x)\big)^2 = \big(y - F(x) - h(x)\big)^2 + 2\big(y - F(x)\big)h(x) + h(x)^2$$
   其中，第一项是常数项，第二项是关于$h(x)$的一次项，第三项是关于$h(x)$的二次项。
4. **定义目标函数**：XGBoost 的目标函数由损失函数和正则化项组成：
   $$\text{Obj} = \sum_i L\big(y_i, F(x_i)\big) + \sum_m \Omega(h_m)$$
   其中，$\Omega(h_m)$是正则化项，用于控制模型的复杂度。对于决策树，正则化项通常包括树的叶子节点数量和叶子节点的分数的平方和。
5. **优化目标函数**：对于每个决策树 $h_m(x)$，XGBoost 通过贪心算法选择最优的分割点，以最小化目标函数。具体的优化过程涉及到数值优化和梯度计算。
6. **输出最终模型**：最终的模型函数是所有弱学习器的加权和
   $$F(x) = \sum_{m=1}^{M} \eta h_m(x)$$
   下面是模型的基本结构图：

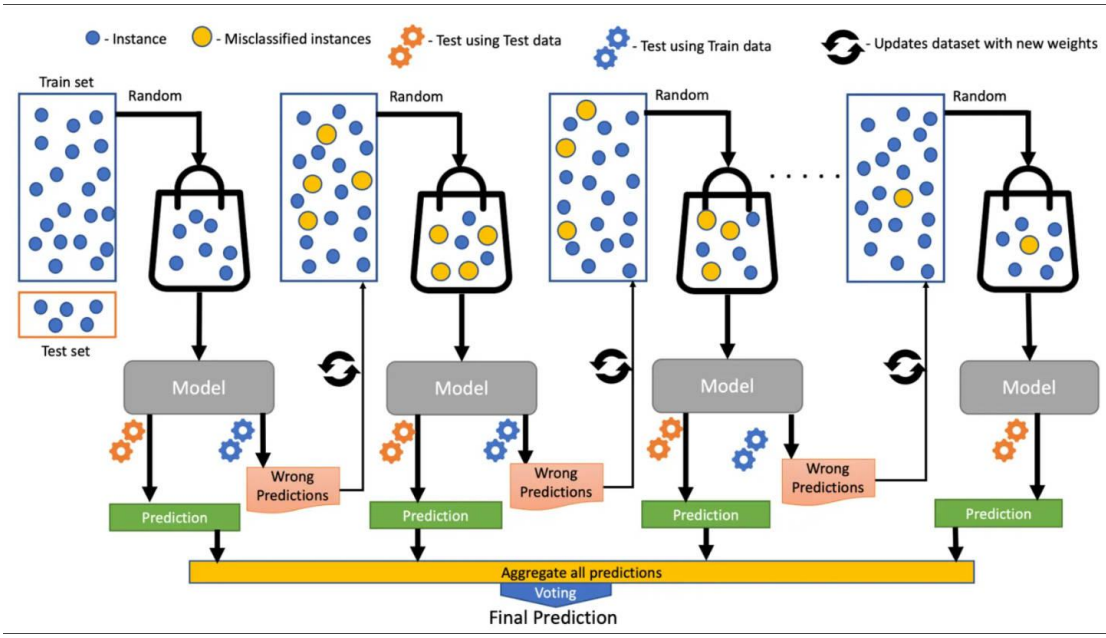

图5：XGBoost模型结构图

4.3.2 基于遗传算法进行超参数模型调优

高效的超参数优化算法对于模型的效果提升具有重要的意义。本文选择了时下较为常用的模型超参数优化启发式算法遗传算法[9, 10]对 XGBoost 模型进行优化。

遗传算法（Genetic Algorithm，GA）是一种启发式搜索算法，它受到生物学的自然选择和遗传机制的启发。遗传算法模拟了自然界中生物的遗传、变异和自然选择过程，以此来搜索问题的最优解或满意解。

遗传算法的基本步骤如下：

1. **初始化：**随机生成一个初始解集合，这个集合被称为种群(population)。
2. **适应度评估：**计算每个解的适应度值（fitness value），适应度值反映了解的优劣程度。
3. **选择（Selection）：**根据解的适应度值，从种群中选择一些解（通常是适应度较高的解）进入下一代。
4. **交叉（Crossover）：**对选中的解进行交叉操作，产生新的解。交叉操作模拟了生物的遗传过程。

5. **变异（Mutation）：**对某些新解进行小的随机扰动，以增加种群的多样性。变异操作模拟了生物的变异过程。

6. **生成下一代：**将交叉和变异产生的新解与上一代中未被选择的解合并，形成新的种群。

7. **迭代：**重复上述步骤，直到满足某个终止条件（如达到预设的迭代次数或适应度满足要求）。

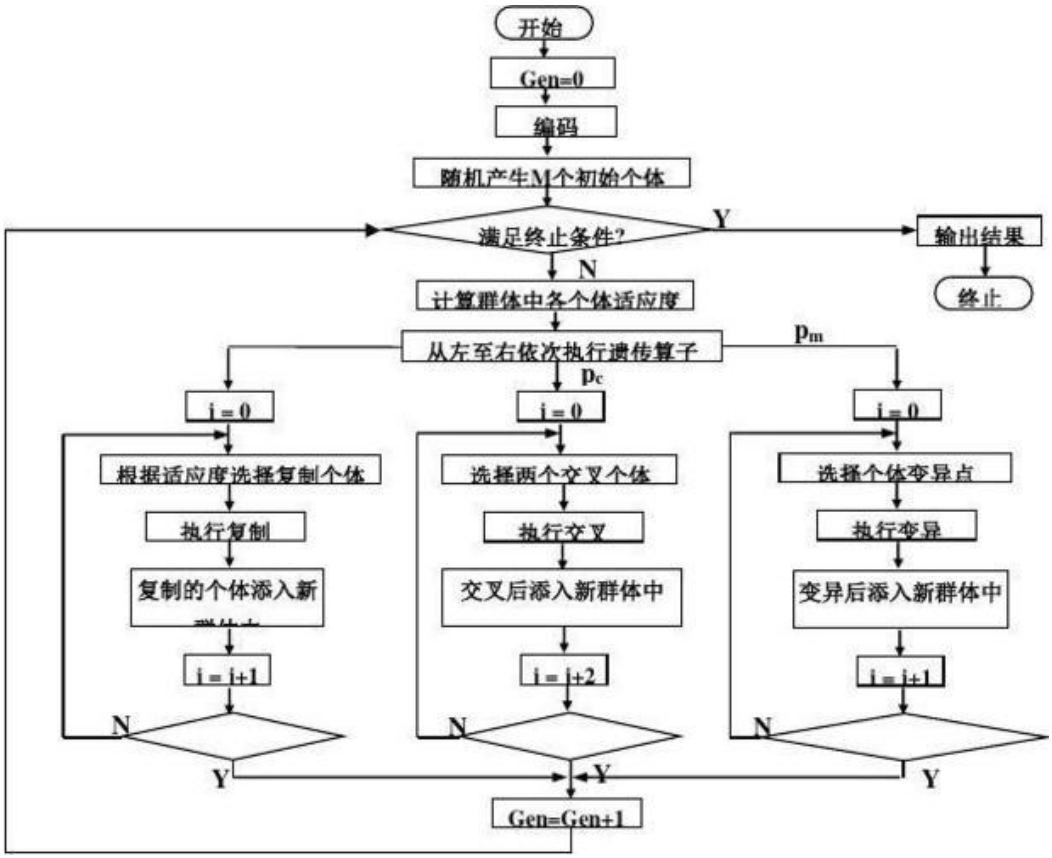

图6：遗传算法流程图

使用遗传（GA）算法对XGBoost模型进行超参数调优的具体步骤如下：

1. **定义XGBoost的参数空间：**首先，需要定义XGBoost的参数空间，包括树的数量、树的深度、学习率、正则化参数等。

2. **编码参数：**将XGBoost的参数转换为遗传算法可以处理的染色体形式。这通常涉及到参数的编码，例如使用二进制编码、实数编码或整数编码。

3. **初始化种群：**生成一个初始种群，每个个体代表一组XGBoost参数的配置。

4. **评估种群**：使用XGBoost模型评估每个个体的性能，通常是通过交叉验证来评估模型在测试集上的性能。

5. **选择、交叉和变异**：遗传算法的核心操作。选择是指根据个体的性能选择哪些个体应该用于下一代；交叉是指将两个或多个个体的染色体组合在一起，形成新的个体；变异是指随机改变某些个体的染色体，以增加种群的多样性。

6. **迭代**：重复选择、交叉和变异步骤，直到满足终止条件，例如达到最大迭代次数或找到满足性能要求的个体。

7. **解码并应用最佳参数**：从最后一代种群中选择表现最佳的个体，将其解码为XGBoost的参数配置，并使用这些参数训练最终的XGBoost模型。

对比评估指标，经过GA遗传算法优化后，在由故障数据1001中，模型的F1_Score评分由0.862提升至0.994，其他类型故障数据训练出的模型也均有提升，可见改进后的模型预测的准确性和稳定性得到显著提升。可以得出结论，GA-XGBoost模型表现更佳，其通过 GA算法对XGBoost模型参数的优化，有效地提高了故障预测的准确性和可靠性。

为了强化该模型的泛化能力。各条生产线经过重新筛选和过采样的一整年的数据作为训练集，不断训练上文提到的GA-XGBoost模型，用M101的数据重新对训练好后模型的准确率进行验证，发现其F1_Score下降到了0.987，故障监测模型的性能指标稍有下降但尚可接受，同时模型的泛化能力得到大幅提升。

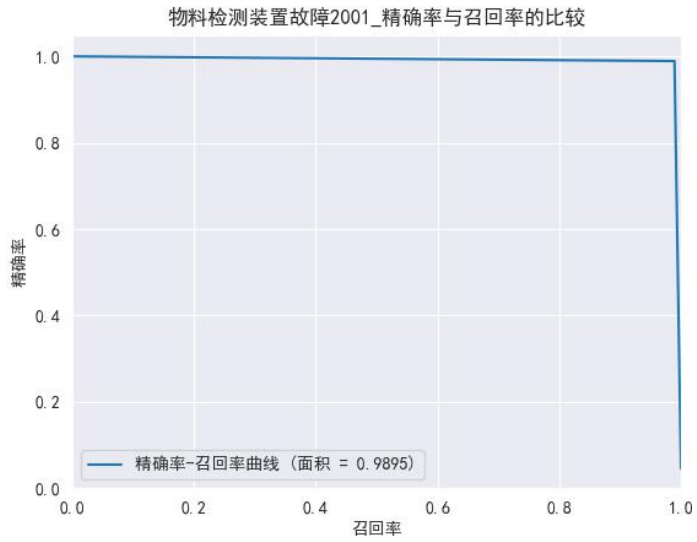

图7：故障数据2001模型评估

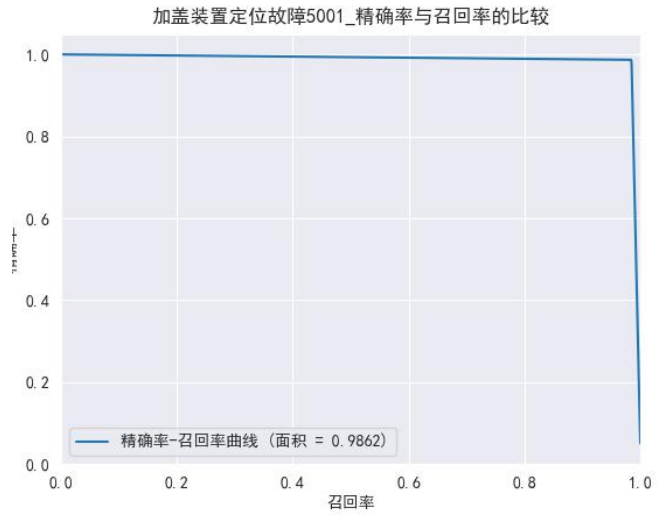

图8：故障5001模型评估

上面两图是基于故障数据2001和故障数据5001训练出的预测模型的精确率与召回率比较图，精确率-召回率曲线围绕的面积可以被视为F1_Score的一个度量，一个为0.9895一个为0.9862，不难发现模型对于各组故障数据的预测的表现都很良好。

4.4 基于GA-XGBoost模型的应用求解

基于4.3.2中训练好的9组针对不同故障类型的GA-XGBoost模型，对问题提供的新的两条生产线数据M201与M202（无故障数据）进行不同类型的故障预测，即根据模型计算预测，对两条生产线的故障数据缺失进行填充。基于预测

后的生产线数据，提取为故障状态的数据进行计算，通过以下方式统计计算各生产线各故障的日期、开始时间、持续时长：

1. 将故障数据以天为单位进行分割
2. 当前天已排好序的数据进行遍历，计算故障事件的开始和结束时间以及持续时长。
3. 遍历过程中，如果开始时间为空，则开始一个新的事件；如果当前时间与前一个时间相差1，则继续当前的事件；否则，结束当前事件并开始一个新的事件。

由此，对一系列时间数据进行处理，将其划分为不同的事件，并计算每个事件的开始时间、结束时间和持续时长，得到各类故障的故障报警的日期、开始时间与持续时长，部分数据展示如下：

故障编号	1001			2001			4001			4002		
序号	日期	开始时间	持续时长/秒	日期	开始时间	持续时长/秒	日期	开始时间	持续时长/秒	日期	开始时间	持续时长/秒
1	3	16826	193	18	7016	3	5	26746	201	1	12089	188
2	5	2035	201	71	7023	1	5	26957	2	1	17209	1
3	8	7483	8	102	7164	3	25	18385	204	1	20688	1
4	18	19012	11	151	7120	1	25	18599	1	1	20728	1
5	32	8829	210	191	20972	9	43	10665	191	1	23248	1
6	32	18214	14	191	20995	9	43	10866	1	1	24416	1

图9：故障报警的日期、开始时间与持续时长的部分数据

基于上述数据的计算结果，根据题目要求，继续计算每条生产线中各装置每月的故障总次数及最长与最短的持续时长，按照以下方式即可计算：

1. 根据装置类别进行分类，根据故障编号的第一个数字进行故障分类。
2. 将日期换算为月份进行月份的分类，依据公式：月份 $= \lfloor 日期/30 \rfloor + 1$
3. 计算同一月份内故障总次数及最长与最短的持续时长。

最后得到每条生产线中各装置每月的故障相关数据，下面展示一下生产线M201中物料推送装置的故障数据汇总，具体所有数据可见附录：

表2：M201中物料推送装置的故障数据汇总

月份	故障总次数	最长持续时长	最短持续时长
1	4	201	8
2	5	210	2
3	5	191	3
4	3	10	8
5	5	195	8
6	10	212	2
7	6	210	1
8	7	209	2
9	3	189	3
10	5	210	2
11	5	212	4
12	2	173	172

五、产品生产绩效的相关性分析

5.1 数据分析以及相关绩效指标计算

经过对数据的观察，不难发现除了故障数据和生产线编号外，一共有26种数据类别即26种的生产影响因素存在，取M301生产线的部分数据展示如下图：

	日期	时间	生产线编号	物料推送气缸推送状态	物料推送气缸收回状态	物料推送数	物料待抓取数	放置容器数	容器上传检测数	填装检测数	...	不合格数	物料推送装置故障1001	物料检测装置故障2001	填装装置检测故障4001	填装装置定位故障4002	填装装置填装故障4003	加盖装置定位故障5001	加盖装置加盖故障5002	拧盖装置定位故障6001	拧盖装置拧盖故障6002
0	1	0	M301	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	1	M301	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	2	M301	0	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	1	3	M301	1	0	2	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	1	4	M301	0	1	2	1	1	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7528804	362	29069	M301	0	1	4362	4362	1456	1455	1454	...	0	0	0	0	0	0	0	0	0	0
7528805	362	29070	M301	0	1	4362	4362	1456	1455	1454	...	0	0	0	0	0	0	0	0	0	0
7528806	362	29071	M301	0	1	4362	4362	1456	1455	1454	...	0	0	0	0	0	0	0	0	0	0
7528807	362	29072	M301	0	1	4362	4362	1456	1455	1454	...	0	0	0	0	0	0	0	0	0	0
7528808	362	29073	M301	1	0	4363	4362	1456	1455	1454	...	0	0	0	0	0	0	0	0	0	0

7528809 rows × 37 columns

图10：M301生产线的部分数据

对此，为了减少计算量以及更清晰的表达产品的产量、合格率与生产线、操作人员等因素的关系，基于原始数据，为了将生产线的运行记录数据汇总成一条数据，我们需要设计一些汇总变量，这些变量能够代表生产线在一年内的整体运行情况。以下是一些可能的汇总变量及其计算公式和原理：

1. **总运行时间**：计算生产线在一年内运行的总时间。 累加每个生产周期的运行时间。公式：

$$\text{总运行时间} = \sum(\text{结束时间} - \text{开始时间})$$

2. **总合格数量**：计算一年内生产线生产的合格产品总数量。计算公式：

$$\text{总合格数量} = \sum \text{合格数}$$

3. **总不合格数量**：计算一年内生产线生产的不合格产品总数量。计算公式：

$$\text{总不合格数量} = \sum \text{不合格数}$$

4. **总生产数量**：计算一年内生产线生产的总产品数量。累加每天记录的合格产品数量和不合格产品总数量。计算公式：

$$\text{总生产数量} = \sum \text{合格数} + \sum \text{不合格数}$$

5. **平均生产效率**：计算生产线的平均生产效率，即每小时生产的合格产品数量。 计算公式：

$$\text{平均生产效率} = \text{总生产数量} / \text{总运行时间}$$

6. **设备综合故障率**：计算所有设备故障的综合故障率。 累加所有设备故障的发生次数，然后除以一年的总天数，得到设备综合故障率。计算公式：

$$\text{设备综合故障率} = \text{所有设备故障的发生次数} / \text{总天数} * 100\%$$

7. **物料推送效率**：计算物料推送气缸的平均推送效率。计算公式：

$$\text{物料推送效率} = \left(\text{物料推送数} / \left(\text{物料推送数} + \text{物料待抓取数}\right)\right) * 100\%$$

8. **填装效率**：计算填装过程的效率。计算公式：

$$\text{填装效率} = \left(\text{填装数} / \text{物料推送数}\right) * 100\%$$

9. **加盖效率**：计算加盖过程的效率。计算公式：

$$\text{加盖效率} = (\text{加盖数}/\text{填装数}) * 100\%$$

10. **拧盖效率**：计算拧盖过程的效率。计算公式：

$$\text{拧盖效率} = (\text{拧盖数}/\text{加盖数}) * 100\%$$

11. **生产总周期数**：计算生产线一年内的总生产周期数。计算公式：

$$\text{生产总周期数} = \sum \text{填装数}$$

12. **合格率**：计算生产线的合格率。计算公式：

$$\text{合格率} = (\text{总合格数量}/\text{总生产数量}) * 100\%$$

基于上述12条汇总变量即绩效指标的计算汇总，舍弃原来繁杂的数据。然后，再将另一份记录着每条生产线相关操作人员的信息的数据根据各自对应的生产线与上述指标进行聚合，以此作为本问题的分析数据标准，其中前七项汇总变量可视为生产线的基本绩效指标，后五项视为进阶绩效指标。将10条生产线的数据聚合后，记录着每一个工作日期下每条生产线的相关绩效指标和操作人员信息，大致展示如下图：

	日期	总运行时间	总合格数量	总不合格数量	总产量	平均生产效率	设备综合故障率	物料推送效率	填装效率	加盖效率	拧盖效率	生产总周期数	合格率	操作人员编号	工龄	生产线编号
0	1	28942	20652487	0	20652487	713.581888	100.0	50.013148	33.298663	99.867257	99.880825	20726488	100.000000	A001	1	M301
1	2	28845	20733120	0	20733120	718.776911	0.0	50.005198	33.307938	99.868350	99.882108	20806612	100.000000	A001	1	M301
2	3	28999	20627556	0	20627556	711.319563	100.0	50.005152	33.307841	99.866839	99.880462	20701308	100.000000	A001	1	M301
3	4	28924	20847028	0	20847028	720.751902	0.0	50.005184	33.308008	99.868709	99.882426	20920722	100.000000	A001	1	M301
4	5	29100	19784942	20893	19805835	680.612887	100.0	50.005133	33.322980	99.968679	99.875100	19858892	99.894511	A001	1	M301
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2595	358	28998	20954008	0	20954008	722.601835	0.0	50.005171	33.308077	99.869045	99.882721	21027894	100.000000	A010	5	M310
2596	359	28985	20751522	101671	20853193	719.447749	100.0	50.005175	33.307787	100.357811	99.882480	20824963	99.512444	A010	5	M310
2597	360	28959	20896354	0	20896354	721.584102	100.0	50.005968	33.307209	99.868847	99.882513	20970169	100.000000	A010	5	M310
2598	361	28978	20410855	0	20410855	704.356926	100.0	50.039086	33.271940	99.865433	99.878988	20485036	100.000000	A010	5	M310
2599	362	28881	20567724	212148	20779872	719.499740	100.0	50.005189	33.307476	100.896608	99.882737	20640803	98.979070	A010	5	M310

2600 rows × 16 columns

图11：生产线的相关绩效指标和操作人员信息部分展示

5.2 差异性分析

基于5.1中创建的数据，为了探究产品的产量、合格率与生产线、操作人员等因素的关系，采用多独立样本Kruskal-Wallis检验方法[11]检验不同生产线以及不同工龄的操作人员的各种绩效指标是否存在差异性。

多独立样本Kruskal-Wallis检验是一种非参数检验方法，用于判断多个独立样本的分布是否存在显著差异。当数据不满足正态分布或者方差不齐时，如果想要比较多个样本的分布是否存在差异，就可以使用Kruskal-Wallis检验。Kruskal-Wallis检验是独立样本的秩和检验，是Wilcoxon秩和检验在三个及以上独立样本情况下的推广。其基本思想是将多个样本的数据混合在一起，通过计算各样本的秩和，并比较其差异，可以判断多个样本的分布是否存在显著差异。Kruskal-Wallis检验的步骤与相关公式大致如下：

1. **数据排序**：将所有样本的数据放在一起，按大小顺序排序，并赋予相应的秩次。
2. **计算秩和**：对于有相同数值的数据，取其平均秩次。
3. **计算 Kruskal-Wallis H 统计量**：使用以下公式计算 Kruskal-Wallis H 统计量：

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)$$

其中，N是所有样本观测值的总和，k是样本组数，R_i是第i组的秩和，n_i是第i组的样本量。

4. **确定显著性水平**：选择一个显著性水平（例如，$\alpha = 0.05$）。
5. **查找临界值**：根据自由度$(df = k - 1)$和卡方分布表查找对应的临界值。
6. **做出决策**：如果计算出的H统计量大于临界值，则拒绝原假设，认为组间存在显著差异。如果H统计量小于或等于临界值，则不能拒绝原假设，认为没有足够的证据表明组间存在显著差异。
7. Kruskal-Wallis 检验的假设为：
    - 零假设（H_0 ）：所有组的分布相同。
    - 备择假设（H_1 ）：至少有一个组的分布与其他组不同。

   如果 p 值小于显著性水平，则拒绝零假设，接受备择假设，认为组间存在显著差异。如果 p 值大于或等于显著性水平，则不能拒绝零假设，认为没有足够的证据表明组间存在显著差异。

5.2.1 不同生产线与绩效指标的差异性分析

首先，针对不同的生产线，先对基本绩效指标进行差异性分析，其中由于

总产量的数值来源于总合格数量和总不合格数量的相加，在此只需考虑总产量。样本量均为2600，而且经过正态分布检验，所有数据均不符合正态分布，将检验结果汇总记录在下表中：

表3：基本绩效指标的差异性分析结果

分析项	统计量	P	Cohen's f值
总运行时间	5.715	0.768	0.003
设备综合故障率	9.529	0.390	0.004
物料推送效率	7.879	0.606	0.006
总产量	1734.691	0.000***	0.052
平均生产效率	1746.759	0.000***	0.053

注：*、**、*分别代表1%、5%、10%的显著性水平，下同**

由上表可得知，基于总运行时间，检验结果P值为0.768>0.05，因此统计结果不显著，说明不同生产线编号在总运行时间上不存在显著差异；其差异幅度Cohen's f值为：0.003，极小程度差异。

基于设备综合故障率，检验结果P值为0.390>0.05，因此统计结果不显著，说明不同生产线编号在设备综合故障率上不存在显著差异；其差异幅度Cohen's f值为：0.004，极小程度差异。

基于物料推送效率，检验结果P值为0.606>0.05，因此统计结果不显著，说明不同生产线编号在物料推送效率上不存在显著差异；其差异幅度Cohen's f值为：0.006，极小程度差异。

基于总产量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在总产量上存在显著差异；其差异幅度Cohen's f值为：0.052，极小程度差异。

基于平均生产效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在平均生产效率上存在显著差异；其差异幅度Cohen's f值为：0.053，极小程度差异。

因此可以得到结论：不同生产线之间的机器总运行时间和设备综合故障率以及物料的推送效率可视为没有差异性，也就是所有机器本身无论在哪一条生

产线运行不存在差异性即不存在品牌、质量、以及产品在生产线中运输等问题。但是可以观察到不同生产线之间产品总产量、以及产品平均生产效率是存在显著差异的，由此推测可能与不同生产线所负责的操作人员的工龄有关，由此对不同生产线进行进阶绩效指标以及上述分析未涉及的总合格数量和总不合格数量的差异性分析，分析结果如下表：

表4：进阶绩效指标以及部分指标的差异性分析结果

分析项	统计量	P	Cohen's f值
总合格数量	1728.354	0.000***	0.051
总不合格数量	24.793	0.003***	0.053
填装效率	1145.494	0.000***	0.008
加盖效率	975.707	0.000***	0.007
拧盖效率	1978.152	0.000***	0.062
生产总周期数	1746.618	0.000***	0.052
合格率	43.02	0.000***	0.007
工龄	2599	0.000***	0.062

基于上表而得知，基于变量总合格数量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在总合格数量上存在显著差异；其差异幅度Cohen's f值为：0.051，极小程度差异。

基于总不合格数量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在总不合格数量上存在显著差异；其差异幅度Cohen's f值为：0.053，极小程度差异。

基于变量填装效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在填装效率上存在显著差异；其差异幅度Cohen's f值为：0.008，极小程度差异。

基于变量加盖效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在加盖效率上存在显著差异；其差异幅度Cohen's f值为：0.007，极小程度差异。

基于变量拧盖效率，检验结果P值为0.000***<0.05，因此统计结果显著，

说明不同生产线编号在拧盖效率上存在显著差异；其差异幅度Cohen's f值为：0.062，极小程度差异。

基于变量生产总周期数，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在生产总周期数上存在显著差异；其差异幅度Cohen's f值为：0.052，极小程度差异。

基于变量合格率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在合格率上存在显著差异；其差异幅度Cohen's f值为：0.007，极小程度差异。

基于变量工龄，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同生产线编号在工龄上存在显著差异；其差异幅度Cohen's f值为：0.062，极小程度差异。

由此基本验证了不同生产线的进阶绩效指标存在显著差异，这一原因大概率是操作人员的工龄对各生产线的作用。下面小杰进一步验证工龄对绩效指标的影响。

5.2.2 不同工龄操作人员与绩效指标的差异性分析

在本小节中进一步验证工龄对绩效指标的影响，在此只需要考虑5.2.1存在显著关系的指标，基于多独立样本Kruskal-Wallis检验分析后的相关结果汇总如下表：

表5：基于工龄的进阶绩效指标以及部分指标的差异性分析结果

分析项	统计量	P	Cohen's f值
总合格数量	669.774	0.000***	0.027
总不合格数量	19.362	0.000***	0.053
总产量	650.412	0.000***	0.027
平均生产效率	651.854	0.000***	0.028
填装效率	400.057	0.000***	0.002
加盖效率	460.468	0.000***	0.005
拧盖效率	840.336	0.000***	0.034

生产总周期数	677.18	0.000***	0.028
合格率	12.495	0.029**	0.003

由上表可得知，基于变量总合格数量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在总合格数量上存在显著差异；其差异幅度Cohen's f值为：0.027，极小程度差异。

基于变量总不合格数量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在总合格数量上存在显著差异；其差异幅度Cohen's f值为：0.027，极小程度差异。

基于总产量，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在总产量上存在显著差异；其差异幅度Cohen's f值为：0.027，极小程度差异。

基于平均生产效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在平均生产效率上存在显著差异；其差异幅度Cohen's f值为：0.028，极小程度差异。

基于物料推送效率，检验结果P值为0.885>0.05，因此统计结果不显著，说明不同工龄在物料推送效率上不存在显著差异；其差异幅度Cohen's f值为：0.003，极小程度差异。

基于填装效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在填装效率上存在显著差异；其差异幅度Cohen's f值为：0.002，极小程度差异。

基于变量加盖效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在加盖效率上存在显著差异；其差异幅度Cohen's f值为：0.005，极小程度差异。

基于变量拧盖效率，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在拧盖效率上存在显著差异；其差异幅度Cohen's f值为：0.034，极小程度差异。

基于变量生产总周期数，检验结果P值为0.000***<0.05，因此统计结果显著，说明不同工龄在生产总周期数上存在显著差异；其差异幅度Cohen's f值

为：0.028，极小程度差异。

基于合格率，检验结果P值为0.029**<0.05，因此统计结果显著，说明不同工龄在合格率上存在显著差异；其差异幅度Cohen's f值为：0.003，极小程度差异。

最后验证结论，不同工龄的操作人员的确对生产线的产品合格率和产量有一定的影响。对此，基于所有数据都不符合正态分布，我先进行了工龄与部分指标的斯皮尔曼相关性分析[12]，具体原理不展开阐述。结果汇总如下图：

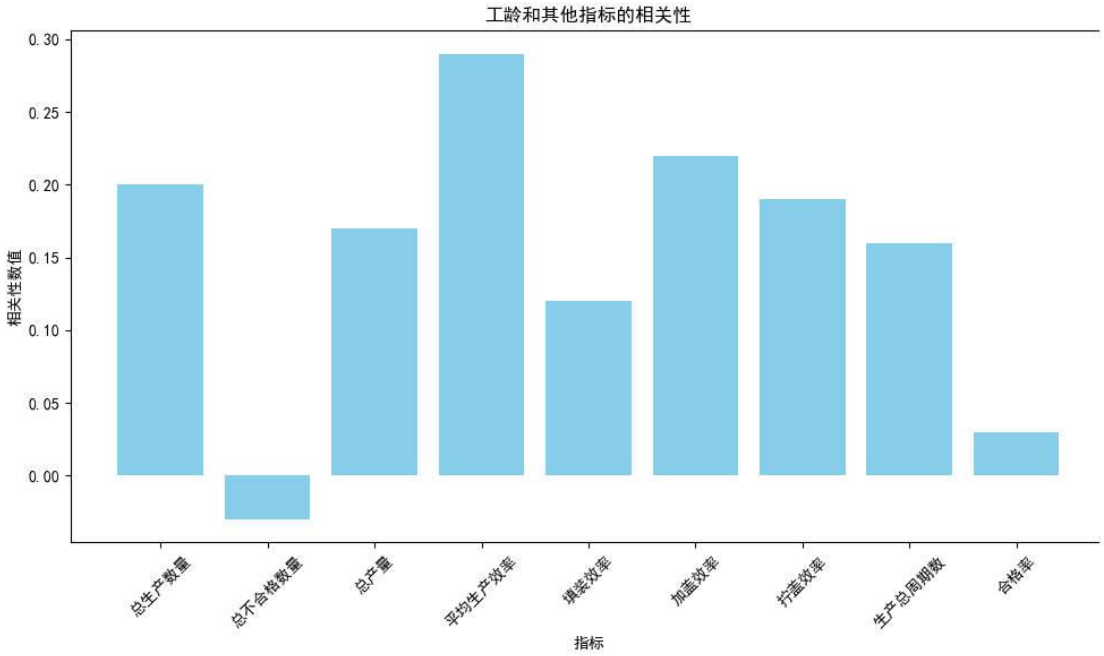

图12：工龄与部分指标的斯皮尔曼相关性分析

可以看出工龄越大的操作人员，能够带来各种工作效率的提升进而带动平均生产效率，且产品的不合格数量减少，生产产品的合格率变高，产量也有所增加。

5.3 机器学习模型求解影响程度

在本节中，为了进一步确定不同工龄的操作人员对生产线的产品合格率和产量的影响。我采用多种机器学习算法模型来进行一个模拟回归的预测，以此来判断工龄对各绩效指标的影响程度。其中包括线性回归模型、支持向量机回归模型、决策树回归模型、随机森林回归模型、神经网络模型、GBDT梯度提

升树回归模型[2, 4, 5, 7, 13, 14]，引入均方误差（Mean Squared Error, MSE）和决定系数（R^2，R Squared）作为评估指标，相关原理和公式简述如下：

1. MSE衡量的是模型预测值与实际值之间差异的平方的平均值。MSE越小，说明模型的预测结果与真实值的偏差越小，即模型性能越好。MSE的公式如下：

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

其中：$\boldsymbol{n}$ 是数据点的数量，Y_i是第 $\boldsymbol{i}$ 个实际观测值，$\widehat{Y}_i$是第 $\boldsymbol{i}$ 个预测值。

2. R^2，也称为解释度，表示模型对总变异的解释程度。其值介于0和1之间，R^2越接近1，说明模型解释变异的能力越强。R^2的计算通常基于SSR（回归平方和）和SST（总平方和）：

$$\mathrm{R}^2 = 1 - \frac{SSR}{SST}$$

其中：SSR（Sum of Squares of the Regression）是回归模型解释的变异，SST（Total Sum of Squares）是总变异，包括模型解释的和未解释的。

R^2也可以通过相关系数（R）的平方来计算：

$$\mathrm{R}^2 = r^2$$

其中r是预测值和实际值之间的皮尔逊相关系数。

MSE和R^2通常结合使用来全面评估回归模型的性能。MSE关注的是模型预测的准确性，而R^2关注的是模型对数据变异的解释能力。一个小的MSE值和一个接近1的R^2值通常意味着一个性能良好的模型。然而，需要注意的是，R^2存在一个局限性，即随着模型复杂度的增加，R^2可能会误导性地增加，即使模型并没有真正改进。因此，单独依赖R^2作为模型评估指标是不够的，需要结合其他指标，如MSE进行综合判断。

5.3.1 选择最优模型求解

在经过多个模型训练评估后，得到的评估指标情况如下表和下图所示：

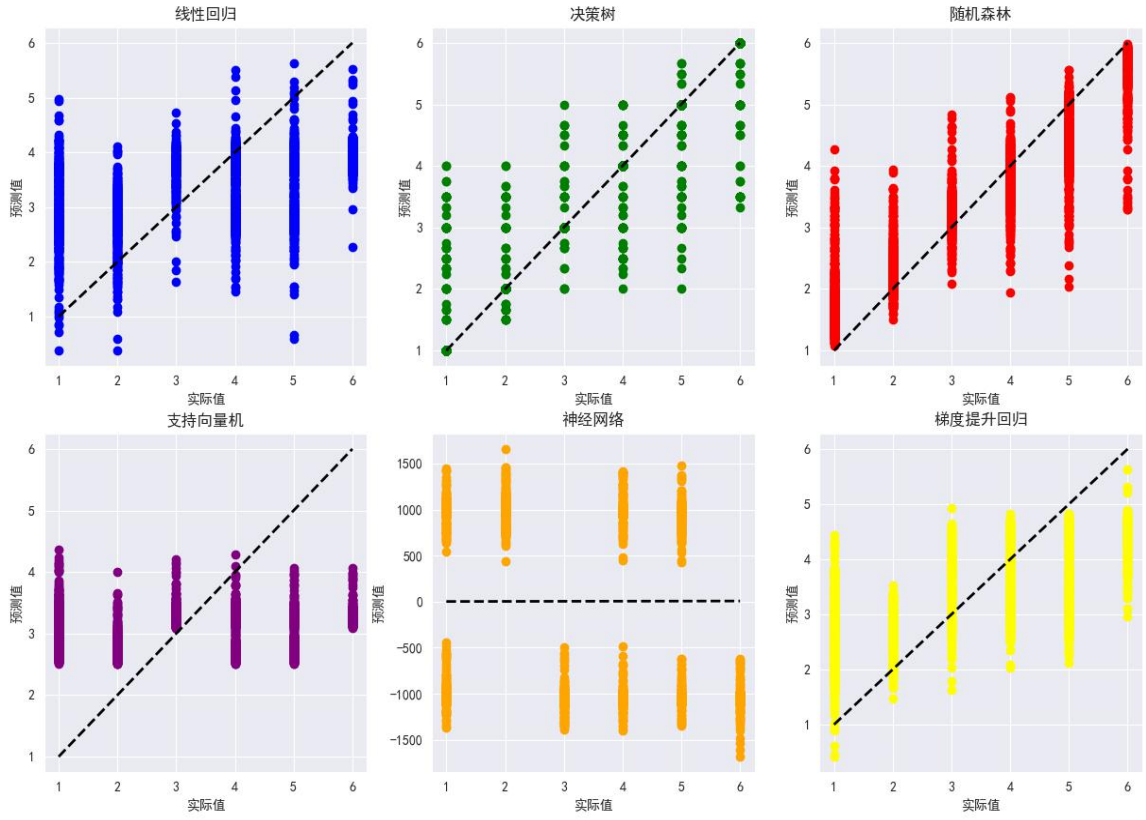

图13：模型效果对比图

表6：模型评估指标结果

模型	MSE	R^2
线性回归	2.35867	0.1606
决策树	0.26906	0.9042
随机森林	0.52534	0.8130
支持向量机	2.77630	0.1189
神经网络	10.48334	-3.8244
梯度提升回归	1.81010	0.3588

由上述数据和图表，可以明确决策树模型的MSE值最小且R^2值最接近于1，为最优表现模型因此我选择决策树模型进行工龄对各因素影响程度的分析。决策树是一种常用的机器学习算法，主要用于分类和回归任务。它是一种树形结构，其中每个内部节点代表一个特征，每个分支代表一个特征值的选择，每个叶节点代表一个类别标签（在分类问题中）或一个预测值（在回归问题中）。决策树模型的原理基于一系列的决策规则，这些规则通过递归地划分数据集来

生成树。决策树的优势在于它们的模型简单、易于理解，并且不需要进行特征缩放或归一化。它们可以处理不同类型的特征（数值型和类别型），并且对异常值具有一定的鲁棒性。下面简述一下决策树模型的原理和过程：

1. **选择最优特征：** 在构建决策树时，首先要选择一个最优的特征来分割数据集。这通常是通过计算信息增益（如ID3算法）、基尼不纯度（如CART算法）或增益率（如C4.5算法）来完成的。这些指标衡量了特征对分类结果的纯度提高程度。
2. **分割数据集：** 根据最优特征的值，将数据集分割成子集。每个子集包含具有相同特征值的样本。这个过程会递归地应用于每个子集，直到满足停止条件。
3. **停止条件：** 递归构建决策树的过程会在满足某些停止条件时停止，例如：当所有样本都属于同一类别时，创建一个叶节点。
4. **树剪枝：** 为了避免过拟合，决策树可能会进行剪枝。预剪枝是在树构建过程中提前停止生长，而后剪枝是在树完全生长后删除不必要的节点。剪枝可以通过设置阈值（如树的深度、叶节点的最小样本数）或使用交叉验证来确定。
5. **预测：** 对于新的样本，决策树通过从根节点开始，根据样本的特征值沿着树向下遍历，直到达到一个叶节点。叶节点的类别标签或预测值就是模型的输出。

5.3.2 SHAP方法对模型进行解释分析

SHAP（SHapley Additive exPlanations）[15]是一种解释机器学习模型预测的统一方法。它基于博弈论中的Shapley值，为每个特征赋予一个影响得分，该得分表示该特征对模型预测的贡献。SHAP可以用于分类和回归任务，它为每个预测提供了局部解释，同时保持了全局解释的一致性。HAP模型检验通常不是指对模型本身进行检验，而是指使用SHAP值来理解和评估模型的行为。以下是使用SHAP进行模型检验的一些关键步骤：

1. **计算SHAP值：** 对于给定的数据点，SHAP值表示每个特征对模型预测的

具体贡献。对于回归任务，SHAP值是特征的边际贡献，使得模型的输出接近实际值。对于分类任务，SHAP值表示特征如何推动模型输出向某个类别倾斜。

2. **可视化SHAP值：** SHAP值可以通过多种方式进行可视化，例如使用SHAP摘要图、依赖图或力图。这些可视化有助于理解模型是如何做出特定预测的，以及哪些特征对预测最重要。
3. **全局解释：** 通过汇总所有数据点的SHAP值，可以获得特征的重要性排序。这有助于理解模型的整体行为，以及哪些特征在模型中起主导作用。
4. **局部解释：** 对于单个预测，SHAP值可以解释为什么模型会做出特定的预测。这有助于模型的可解释性和信任度。
5. **一致性检验：** 通过比较不同预测的SHAP值，可以检验模型的一致性。如果模型对类似的输入给出相似的SHAP值，则说明模型是一致的。
6. **敏感性分析：** 通过分析特征SHAP值的变化，可以进行敏感性分析，了解模型对特征变化的反应。
7. **模型诊断：** 如果发现某些特征具有异常高的SHAP值，这可能表明模型存在问题，如过拟合或数据泄漏。
8. **公平性分析：** SHAP值还可以用于检查模型是否对某些受保护的特征（如性别、种族）给予了不公平的权重，从而进行公平性分析。

应用SHAP方法及进行处理检验后，得到如下两图的结果：

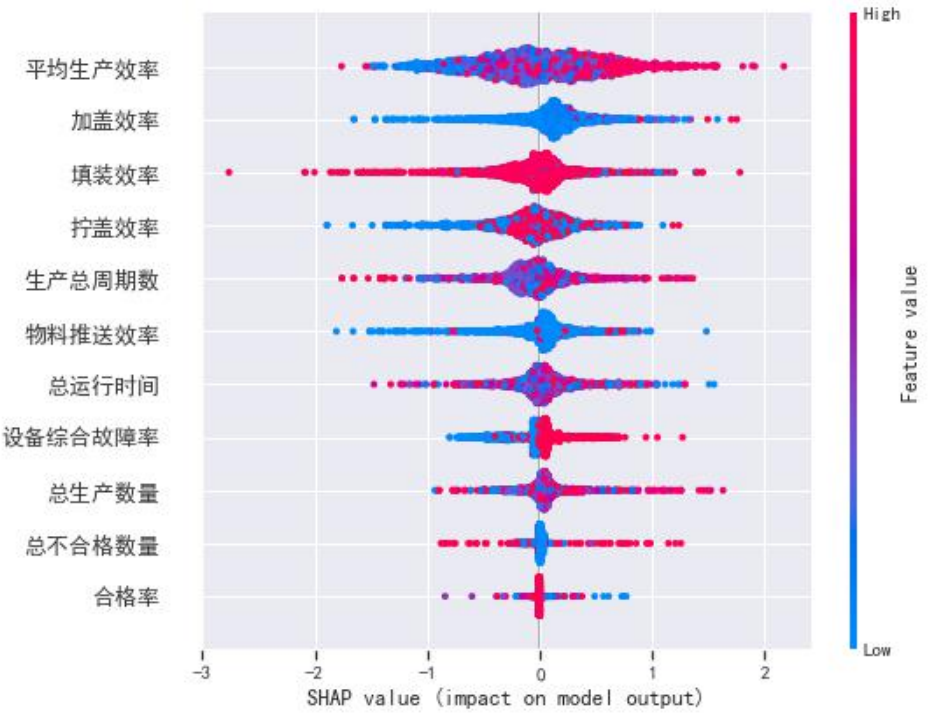

图14：shap值汇总

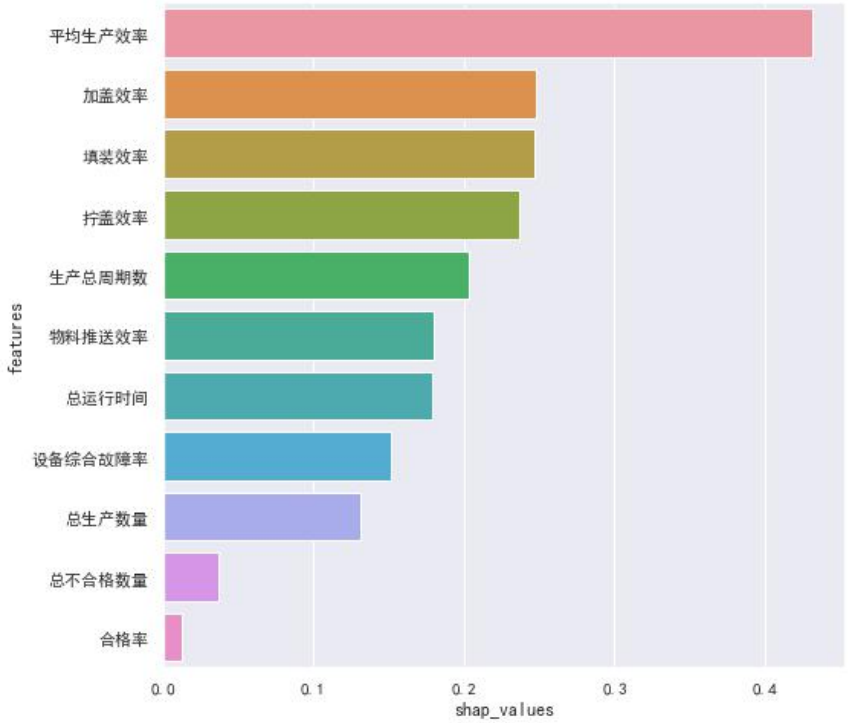

图15：工龄与各因素之间的shap值相关度

可以看出操作人员工龄和涉及到的众多生产指标都有关系，其中工龄对平均生产效率影响最大，可得知工龄带来的生产经验有利于生产线生产效率的提高，由此也能验证5.2中的说法。

总的来说，工龄越大的操作人员，拥有更丰富的生产经验和技能，能够让生产线中各种工作效率的提升进而带动平均生产效率，且产品的不合格数量减少，生产产品的合格率变高，产量也有所增加。

六、人员排班配置优化规划模型分析建立与求解

6.1 人员排班配置优化规划模型的建立

在问题4中，需要针对扩大生产规模后的生产线进行人员的规划排班，目标是在未来一年内，合理安排不同操作人员在每一工作日内三个班次的出勤规划，考虑人员的技术、身体条件等因素，以此实现人员安排的最优化从而最大化生产效率。此问题需要建立一个目标规划模型来规划生产线M301-M310这十条生产线在未来一年内每个工作日内三个班次的最优人员配置。

在生产线扩大规模之后，需要对操作人员的排班方案进行调整。首先考虑的是操作人员的分布情况，已知操作人员的比例是确定的，同M301-M310上的操作人员工龄分布保持一致。因此，结合生成线扩大规模后的人手情况，可以计算出需要的操作人员数以及对应的工龄分布情况。得出下面的公式：

$$N^w = 7 * \lambda * n^l / D$$

其中N^w表示需要的操作人员数量，λ表示每天生成线需要安排的班次，在本文中将分成早、中、晚三个班次，n^l表示已有的生产线数即10，D表示的是操作人员每周的工作天数，最大值为5。

容易计算出，在扩大规模后最少需要42名操作人员即可以满足生产线每天24小时全年午休的工作需求。本次问题就取最小限度的42名操作人员，尽量拟合问题3中不同工龄操作人员的比例，得到本次问题求解的操作人员工龄的分布情况，如下表所示。为操作人员附上编号，从B001-B042，按照下表的顺序依次编号。

表7：操作人员工龄的分布情况表

工龄	1	2	3	4	5	6

数量	8	8	4	9	9	4

在问题3的求解中，为了问题4求解，计算汇总了不同工龄操作人员每日的效率信息，其中总产量取得是平均值，展示如下：

表8：不同工龄操作人员每日的工作效率信息表

工龄(年)	生产效率(%)	总产量(件)
1	99.90789	81063
2	99.90778	82753
3	99.87249	79820
4	99.92405	81329
5	99.89709	81465
6	99.91535	79955

6.1.1 定义与构建问题相关目标函数

为了确定未来一年每个工作日期内每条生产线在各个班次的最佳人员配置，需要建立一个目标优化模型，该模型将结合不同工龄的人员的技术效率，均衡操作人员工作时间，确保生产线生产效率最大化。由此我建立了一个目标函数，取具体工人每日生产数量和产品合格率的乘积作为生产效率的度量公式如下：

$$Max(Z) = \sum_{i=1}^{N} \sum_{j=1}^{D} \sum_{k \in S} x_{ijk} \cdot E_{\mathrm{r},i} \cdot E_{\mathrm{q},i}$$

其中:

- N 是工人总数。
- D 是一周的天数 。
- S 是班次集合 ，包括早、中、晚班。
- x_{ijk} 是二进制决策变量，表示工人 i 在第 j 天的班次 k 是否工作（1表示工作， 0 表示不工作）。
- $E_{\mathrm{r},i}$ 是工人 i 的合格率。

- $E_{\mathrm{q},i}$ 是工人 i 的每日生产数量。

6.1.2 定义与构建问题相关约束条件

根据题目要求，基于6.1.1提出的目标函数，总共需要满足以下3个约束条件：

1. 工作天数约束：每个工人每周工作 5 天，休息 2 天。

$$\sum_{j=1}^{D}\sum_{k\in S} x_{ij} = 5,\ \forall i = 1,2,\ldots,N$$

2. 班次人数约束: 每个班次每天需要10个不同的人。

$$\sum_{i=1}^{N} x_{ijk} = 10,\ \forall j = 1,2,\ldots,D,\ \forall k \in S$$

3. 工龄分布比例约束：确保每个工龄等级的工人数量符合给定的比例。

$$\sum_{i=1}^{N}\sum_{j=1}^{D}\sum_{k\in S} x_{ijk} = W_e, \forall e \in working\ year$$

其中 W_e是工龄等级 e的工人在一周内工作的总人数，根据工龄等级的比例和工人总数计算取整得出。

6.2 基于模拟退火算法的人员出勤排班规划模型优化求解

模拟退火算法[16, 17]（Simulated Annealing，SA）是一种通用概率算法，用来在一个大的搜寻空间内找寻问题的近似最优解，它是一种启发式算法模拟退火算法最早的思想是由N.Metropolis等于1953年提出。1983年，S.Kirkpatrick等成功地将退火思想引入组合优化领域。它是基于Monte-Carlo迭代求解策略的一种随机寻优算法，其出发点基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。模拟退火算法从某一较高初温出发，伴随温度参数的不断下降，结合概率突跳特性，在解空间中随机寻找目标函数的全局最优解，即局部最优解能概率性地跳出，并最终趋于全局最优。该算法具有概率的全局优化

性能，目前已在工程中得到了广泛应用，如VLSI（超大规模集成电路）、生产调度、控制工程、机器学习、神经网络、信号处理等领域。[18]

模拟退火算法是通过赋予搜索过程一种时变且最终趋于零的概率突跳性，从而可有效避免陷入局部极小，并最终趋于全局最优的串行结构的优化算法。该算法还具有较强的鲁棒性、全局收敛性、隐含并行性及广泛的适应性，并且能处理不同类型的优化设计变量（离散的、连续的和混合型的），无须任何辅助信息，对目标函数和约束函数没有任何要求。它利用Metropolis算法，并适当地控制温度下降过程，在优化问题中具有很强的竞争力。

下面是结合模拟退火算法对本问题进行求解的详细步骤与分析：

1. **初始化**：随机创建一个初始解集，每个解代表一种排班方案，包括每个班次的不同生产线的操作人员分配。定义初始能量即初始的生产效率、标准偏差和方差、初始温度和退火速率。
2. **模拟退火过程（邻域搜索）**：在当前温度下，在每一步迭代中随机选择一个或多个班次，并尝试改变操作人员的分配。这个过程称为邻域搜索，生成的解称为邻域解。生成当前解的一个邻域解，即对当前解进行微小的随机扰动。
3. **接受准则**：计算当前解和邻域解的目标函数值。计算目标函数值的差值，并根据接受概率公式决定是否接受邻域解。如果邻域解的目标函数值优于当前解则接受这个新解作为当前解。如果邻域解的目标函数值劣于当前解,以一定的概率接受这个劣解，这个概率由当前温度和解的差异决定，遵循玻尔兹曼分布:

$$P(accept) = exp\left(-\frac{\Delta E}{k_B T}\right)$$

其中，ΔE是新解和当前解的能量差即目标函数数值差，k_B是玻尔兹曼常数，T是当前温度。重复上述步骤，直到满足停止条件，如达到预定的迭代次数或目标函数值收敛。

4. **温度下降**：每次迭代后，根据退火速率降低温度。随着温度的降低，本质上劣解接受概率逐渐减小，算法逐渐收敛到最优解。

5. **结果分析**：输出最优解，即目标函数值最小的解，作为人员出勤排班规划的优化结果。分析结果，评估其在实际操作中的可行性和效果。
6. **调整参数**：根据实验结果调整模拟退火算法的参数，如初始温度、退火速率等，以获得更好的优化效果。

针对本次问题，设置迭代次数为1000次，求解过程中适应度迭代图如下图所示：

从迭代图中得出，图中的纵坐标通常代表目标函数的值，而横坐标代表迭代次数或时间，可以看到，算法在第40次左右迭代时，已经达到了个相对稳定的状态，几乎没有进一步的变化，这表明算法已经接近收敛，找到了最优解或者是一个非常接近的解。说明模拟算法成功地对问题进存了优化，并且很可能找到了一个质量较高的解。这个平稳的收效表明了算法的稳定性和有效性。

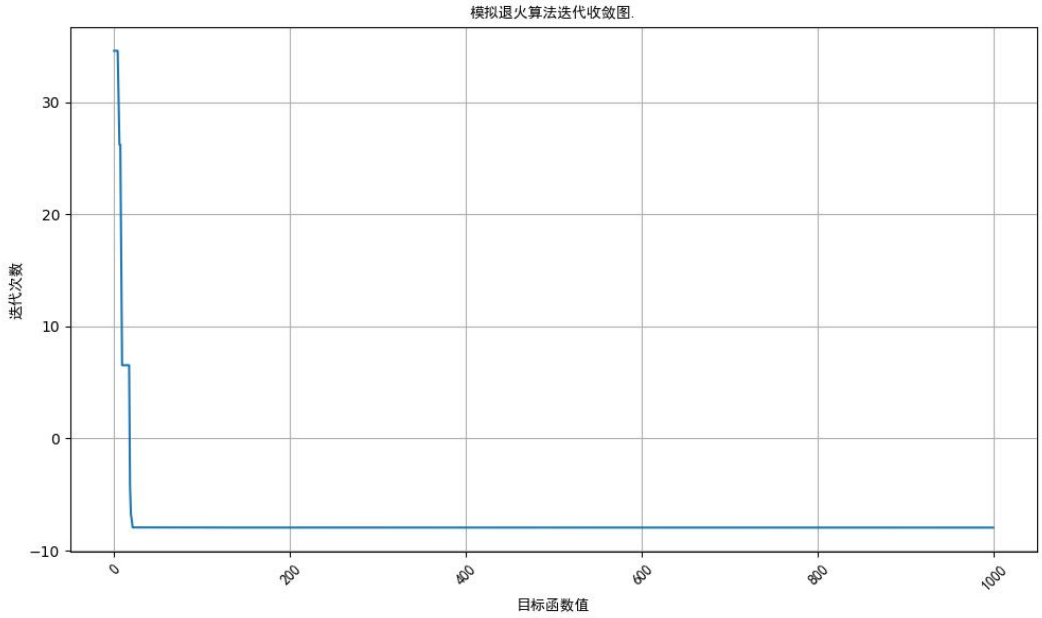

图16: 模拟退火算法迭代收敛图

最后，我利用该算法模型对问题进行求解，分别求解得出一年内每一个操作人员出勤班次整体规划方案以及对于每条生产线不同班次的操作人员分配方案，下面展示部分求解后得到的数据，详细数据请查阅附件中的结果表result4-1和result4-2。

	日期	B001	B002	B003	B004	B005	B006	B007	B008	B009	...	B033	B034	B035	B036	B037	B038	B039	B040	B041	B042
0	1	晚	晚	早	早	中	早	晚	中	晚	...	休	休	早	休	中	早	早	晚	休	早
1	2	休	晚	休	中	中	休	休	中	晚	...	早	早	晚	早	晚	晚	中	晚	中	中
2	3	晚	休	中	休	休	早	早	晚	休	...	早	中	休	早	晚	休	晚	休	中	休
3	4	休	晚	早	早	晚	休	休	休	早	...	早	晚	中	晚	休	晚	休	晚	中	早
4	5	早	中	休	早	休	早	早	休	晚	...	休	晚	休	晚	休	晚	休	晚	晚	中
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
359	360	晚	晚	早	早	中	早	晚	中	晚	...	休	休	早	休	中	早	早	晚	休	早
360	361	休	晚	休	中	中	休	休	中	晚	...	早	早	晚	早	晚	晚	中	晚	中	中
361	362	早	休	中	休	休	早	早	晚	休	...	早	中	休	早	晚	休	晚	休	中	休
362	363	中	晚	早	早	晚	休	休	休	早	...	早	晚	中	晚	休	晚	休	晚	中	早
363	364	早	晚	早	早	中	早	晚	中	晚	...	休	休	早	休	中	早	早	晚	休	早

图17：每一个操作人员出勤班次整体规划方案

	日期	班次	M301	M302	M303	M304	M305	M306	M307	M308	M309	M310
0	1	早	B003	B004	B006	B011	B028	B031	B035	B038	B039	B042
1	1	午	B005	B008	B010	B019	B020	B027	B029	B030	B032	B037
2	1	晚	B001	B002	B007	B009	B017	B021	B022	B023	B025	B040
3	2	早	B012	B013	B014	B016	B018	B024	B026	B033	B034	B036
4	2	午	B004	B005	B008	B010	B011	B015	B017	B039	B041	B042
...	...	...	...	...	...	...	...	...	...	...	...	...
1087	363	午	B001	B002	B003	B005	B008	B017	B025	B028	B035	B039
1088	363	晚	B009	B011	B018	B019	B021	B023	B027	B031	B034	B038
1089	364	早	B001	B004	B008	B012	B014	B017	B020	B030	B039	B040
1090	364	午	B006	B010	B016	B022	B024	B026	B036	B037	B041	B042
1091	364	晚	B003	B005	B007	B013	B015	B025	B029	B032	B033	B035

图18：每条生产线不同班次的操作人员分配方案

七、模型整体评价

本文针对智能化生产线中的故障预测和识别及人员安排规划排班问题,提出了一套综合的解决方案。同时探究了操作人员的工龄对产品产量与合格率的影响。通过深入分析历史机器运行数据，并结合差异性分析、机器学习算法和启发式优化算法以及SHAP方法解释，本文建立了一系列较为准确的故障预测模型和人员安排规划排班模型，能够在一定程度上有效提升了智能化生产线的生产效率和降低了运作成本。

7.1 模型优势

首先，在本次的问题解决中，我发现了数据类别上的波动性可能会对模型的效果有影响，于是经过多重分析，对数据进行清洗和再处理和生成，样本数据均衡，再采用了过采样方法对数据进行处理，构建了一个可信服的数据进行分析。

其次，我尝试引入了先进的有效的机器学习算法以及启发式优化算法，例如XGBoost（XGBoost极度梯度提升树回归模型）来捕捉故障数据和其他数据之间的长期依赖关系，又利用遗传算法GA对模型的超参数进行全局寻优，让模型有更好的性能和泛化效果。

然后，在考虑人员的具体排版优化规划时，模型不仅考虑到了生产线的的产品产量和合格率需求，还细致地结合了操作人员的工作效率、时间成本及工作时间限制等多重因素，最后通过启发式优化算法模拟退火算法，实现了在满足基本工作需求的同时，最大化生产效率，并能够在一定程度上保持工作效率的均衡。

最后，在模型的求解中，针对目标函数和约束条件，我充分考虑了各种实际的可能的约束条件，例如人员的出勤率限制、班次安排的唯一性等，保证了解的可行性。

7.2 模型不足

第一，模型对于数据还是存在着一定程度上不可避免的过分依赖，机器学习模型的预测准确性很大一部分程度上取决于训练数据的质量和完整性，这关乎到模型是否能很好的从数据中提取相应的具体代表性的特征，如果历史货量数据存在缺失或异常值或者明显的波动性，很容易影响到模型的预测效果和模型的泛化性能。

第二，由于本次问题数据量的庞大和选择算法的复杂度，模型求解和计算的复杂度很高，虽然XGBoost模型和GA、模拟退火等优化算法在预测表现上都呈现了不俗的效果，但是还是需要足够多的计算资源和时间，这可能在一定程

度上会限制模型在实时实地预测或需要大规模处理数据时的应用体验和效果。

第三，由于模型的复杂，无论是XGBoost模型和GA、模拟退火等优化算法等，都需要调整多个超参数的设定，这些参数的选择对算法和模型的性能有极大影响，但由于缺乏足够的理论指导和实践经验，需要通过大量的验证对比实验来确定超参数设置，增加了模型求解和应用的难度，泛化性不够好。

同时，尽管模型在特定数据集上表现良好，但其算法稳定性仍需进一步验证。模型的性能可能受到特定数据分布和特征的影响，因此需要进一步测试以评估其在不同环境下的鲁棒性。

最后，模型可能缺乏直观的用户界面，这使得非技术用户难以理解和使用模型。用户界面设计的重要性不应被忽视，因为它直接影响用户体验和模型的实际应用

7.3 模型推广

跨行业应用：该模型不仅适用于智能化制造业行业，还可以推广到其他需要进行故障预测和人员排班的行业，如快递行业等。具有较强通用性和扩展性，意味着它可以适应不同行业的特定需求和规则。

长期规划支持：通过该模型的预测结果，企业可以进行更长远的资源规划和人力配置。这有助于提高整体运营效率，降低成本，因为企业可以提前规划资源使用和人员分配，以适应未来的需求变化。

决策支持系统：模型可以作为企业决策支持系统的一部分。为管理层提供数据支持和策略建议，帮助企业更好地应对市场变化和不确定性。决策支持系统通常包括数据分析、预测模型和报告工具，以帮助管理层做出基于数据的决策。

供应链管理优化：模型可以与供应链管理系统集成，优化物流和库存管理。

通过精确预测货量，企业可以减少库存成本，提高供应链效率。

自动化与智能化：模型可以与自动化和智能化技术集成，实现排班的自动化和智能化。这可以减少人工干预，提高排班的准确性和效率。

与其他系统的集成：模型的输出可以与其他企业资源规划（ERP）系统、供应链管理系统等进行集成。实现数据共享和业务流程自动化，提高企业的整体信息水平。集成多个系统可以减少重复工作，提高数据一致性和业务流程的效率。

这些应用和优势强调了模型在支持企业决策和提高运营效率方面的潜力，以及它如何通过集成到企业现有的IT基础设施中，为企业带来额外的价值。

八、参考文献

[1] A. Fernández, S. Garcia, F. Herrera, and N. V. J. J. o. a. i. r. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," vol. 61, pp. 863-905, 2018.

[2] G. Biau and E. J. T. Scornet, "A random forest guided tour," vol. 25, pp. 197-227, 2016.

[3] T. Hastie, S. Rosset, J. Zhu, H. J. S. Zou, and i. Interface, "Multi-class adaboost," vol. 2, no. 3, pp. 349-360, 2009.

[4] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[5] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. J. J. o. C. A. J. o. t. C. S. Brown, "An introduction to decision tree modeling," vol. 18, no. 6, pp. 275-285, 2004.

[6] A. Ogunleye, Q.-G. J. I. A. t. o. c. b. Wang, and bioinformatics, "XGBoost model for chronic kidney disease diagnosis," vol. 17, no. 6, pp. 2131-2140, 2019.

[7] J. Ye, J.-H. Chow, J. Chen, and Z. Zheng, "Stochastic gradient boosted distributed decision trees," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 2061-2064.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.

[9] J. H. J. S. a. Holland, "Genetic algorithms," vol. 267, no. 1, pp. 66-73, 1992.

[10]葛继科, 邱玉辉, 吴春明, and 蒲. J. 计算机应用研究, "遗传算法研究综述," vol. 25, no. 10, pp. 2911-2916, 2008.

[11]N. J. B. Breslow, "A generalized Kruskal-Wallis test for comparing K samples subject to unequal patterns of censorship," vol. 57, no. 3, pp. 579-594, 1970.

[12]L. Myers and M. J. J. E. o. s. s. Sirois, "Spearman correlation coefficients, differences between," vol. 12, 2004.

[13]G. A. J. N. n. Carpenter, "Neural network models for pattern recognition and associative memory," vol. 2, no. 4, pp. 243-257, 1989.

[14]M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B. J. I. I. S. Scholkopf, and t. applications, "Support vector machines," vol. 13, no. 4, pp. 18-28, 1998.
[15]S. Mangalathu, S.-H. Hwang, and J.-S. J. E. S. Jeon, "Failure mode and effects analysis of RC members based on machine-learning-based SHapley Additive exPlanations (SHAP) approach," vol. 219, p. 110927, 2020.
[16]P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts, *Simulated annealing*. Springer, 1987.
[17]陈华根, 吴健生, 王家林, and 陈. J. 同. 自然科学版, "模拟退火算法机理研究," vol. 32, no. 6, pp. 802-805, 2004.
[18]岳琪 and 沈. J. 信息技术, "模拟退火算法在单目标规划问题中的应用," vol. 30, no. 5, pp. 27-28, 2006.

附录

问题2中每条生产线中各装置每月的故障总次数及最长与最短的持续时长记录如下：

表：M201 物料检测装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	1	3	3
2	0	0	0
3	1	1	1
4	1	3	3
5	0	0	0
6	1	1	1
7	5	9	9
8	2	1	1
9	1	1	1
10	3	1	1
11	6	9	8
12	0	0	0

表：M201 填装装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	296	204	1
2	240	212	4
3	188	194	7
4	244	205	1
5	185	217	2
6	264	205	1
7	247	214	3
8	221	210	1
9	234	199	1
10	267	199	2
11	204	200	1
12	249	201	5

表：M201 加盖装置故障

1	802	166	1
2	888	147	1
3	824	11	2
4	964	45	1
5	905	187	1
6	1099	166	6
7	1040	21	7
8	1055	190	1
9	1076	11	2
10	1193	11	1
11	1016	130	5
12	821	203	4

表：M201 拧盖装置故障

1	323	142	1
2	392	74	7

3	310	11	1
4	381	64	9
5	350	11	1
6	475	82	5
7	420	111	1
8	467	176	1
9	455	26	4
10	541	11	3
11	367	17	1
12	334	213	6

表：M202 物料推送装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	7	93	3
2	3	70	66
3	6	98	5
4	4	99	9
5	5	94	4
6	9	99	9
7	6	96	7
8	6	90	10
9	5	96	4
10	3	95	65
11	4	91	8
12	5	97	78

表：M202 物料检测装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	156	7	3
2	191	7	1
3	149	8	4

4	457	7	1
5	161	7	1
6	233	7	1
7	409	58	1
8	230	7	1
9	85	7	2
10	170	9	3
11	176	7	1
12	281	15	2

表：M202 填装装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	612	104	1
2	712	98	2
3	518	91	7
4	750	96	1
5	560	87	6
6	691	89	3
7	610	102	1
8	478	78	4
9	640	96	2
10	554	87	1
11	372	90	8
12	493	100	1

表：M202 拧盖装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	1163	11	2
2	1415	62	7
3	1101	80	5
4	1483	23	1

5	1194	11	1
6	1380	11	6
7	1162	11	1
8	975	82	5
9	1369	11	2
10	1152	11	4
11	745	11	1
12	913	11	1

表：M202 加盖装置故障

月份	故障总次数	最长持续时长	最短持续时长
1	528	77	3
2	677	11	1
3	425	8	5
4	698	11	2
5	561	11	2
6	658	11	1
7	548	93	7
8	413	11	9
9	583	73	10
10	493	66	1
11	330	11	1
12	528	77	1