首发于**[数学建模](https://www.zhihu.com/column/c_1434248425172418560)**

写文章





**【2024泰迪杯】A 题：生产线的故障自动识别与人员配置 Python代码实现**

[](https://www.zhihu.com/people/823316627)

[**BetterBench**](https://www.zhihu.com/people/823316627)

CS博士/类脑计算/大数据/数学建模/竞赛博主/国奖/网球

​关注

2 人赞同了该文章

**【2024泰迪杯】A 题：生产线的故障自动识别与人员配置 Python代码实现**



**1 问题**

一、问题背景

随着新兴信息技术的大规模应用，工业生产线的智能化控制技术日益成熟。自动生产线

可以自动完成物品传送、物料填装、产品包装和质量检测等过程，极大地提高了生产效率和产品质量，减少了生产成本。自动生产线融入故障智能报警技术，能避免因故障带来的生产中断和经济损失；同时合理的人员配置，能够减少资源浪费、提高生产效率。

二、解决问题

问题 1： 根据附件 1 中的数据，分析生产线中各装置故障的数据特征，构建故障报警模型，实现故障的自动即时报警。

问题 2： 应用问题 1 所建立的模型，对附件 2 中的数据进行分析判断，实现生产线中各装置故障的自动即时报警，给出故障报警的日期、开始时间与持续时长，将结果存放到result2.xlsx 中（格式见表 1，模板文件在附件 2 中），并在论文中给出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。

表 1 result2.xlsx 的格式

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 故障编号 | 1001 | 2001 | … | 6002 |  |  |  |  |  |  |
| 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 |
| 1 | 3 | 579 | 67 | 1 | 4792 | 98 | … | 1 | 2047 | 51 |
| 2 | 4 | 1648 | 101 | 1 | 18379 | 72 | … | 2 | 9482 | 48 |
| 3 | 7 | 20947 | 96 | 3 | 1971 | 122 | … | 3 | 1903 | 58 |
| … | … | … | … | … | … | … | … | … | … | … |

问题 3 ： 根据附件 3 中的数据，分析产品的产量、合格率与生产线、操作人员等因素的关系。

问题 4： 根据实际情况，现需要扩大生产规模，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。针对问题 3 的 10 条生产线，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，将结果存放到 result4-1.xlsx 和result4-2.xlsx 中（格式见表 2 和表 3，模板文件在附件 4 中），并在论文中给出最佳的排班方案及相关结果。

要求排班满足如下条件：

(1) 各操作人员做五休二，尽量连休 2 天；

(2) 各操作人员每班连续工作 8 小时；

(3) 班次时间：早班（8:00-16:00）、中班（16:00-24:00）、晚班（0:00-8:00）；

(4) 各工龄操作人员的人数比例与问题 3 中的比例相同；各操作人员的班次安排尽量均衡。

**2 问题分析**

**2.1 问题一**

（1）分析故障数据特征，包括故障发生的时间分布、故障发生的频率、故障类型的分布等。

（2）构建故障报警模型，是多标签的分类问题，或者是针对每个故障编码建模，就是一个分类问题。分类算法包括决策树、随机森林、支持向量机、逻辑回归和神经网络等。也可以建立成一个时间序列问题，注意数据集给的日期和时间，格式不是标准的datetime格式，是1、2、3、4这样的格式。需要转换为datetime再去建模。

**2.2 问题二**

（1）故障识别

* 如果建立成一个分类问题，针对每个生产线、每个故障编码的序列，分别训练一个分类模型。
* 如果建立时间序列回归问题，可以采用ARIMA、LSTM等构建模型进行故障预测。

题目要求以附件一的数据作为训练集，预测附件2，然后生成result2.xlsx文件。但是目前还没有提供附件二的数据，就可以以附件一为训练集和测试集。跑通baseline模型。

（2）故障时长计算

* 单次故障，不连续，持续时间为1
* 连续两次故障，持续时间为2
* 连续多次故障，持续时间为，开始时刻和结束时刻之间的时间差加1。

（3）每条生产线中各装置的每月统计数据

注意需要计算每个生产线的每个月统计数据，则需要使用分组操作来分析每条生产线中各装置的每月统计数据（故障次数、最长持续时长和最短持续时长）。

（4）输出结果

将每种类型的故障分析结果，横向排列，纵向是按故障发生的顺序排列，整理为dataframe格式，输出到result2.xlsx中。

**2.3 问题三**

（1） 产量分析

使用时间序列分析方法（如ARIMA模型）来预测产量随时间的变化趋势。

回归分析来探索产量与其他因素（如生产线编号、推送状态、装置故障等）的关系。

（2）合格率分析

逻辑回归分析来探索合格率与其他因素之间的关系；

使用决策树或随机森林等分类算法，寻找影响合格率的关键因素

（3） 故障分析

关联规则挖掘算法查找故障与其他因素（如操作人员、生产线编号）之间的关联性。

**2.4 问题四**

（1）在问题三的基础上，计算出每个生产线的生产效率和合格率，找出与操作人员、班次等因素的关系；

（2）根据生产效率和合格率的关系，制定最佳的操作人员排班方案；制定最佳的操作人员排班方案涉及到多个因素，包括但不限于：工作时间要求、人员能力匹配、轮班规则、休息时间安排、法定假期等。

（3）根据实际数据进行生产效率和合格率的计算，然后根据操作人员的工作时间和班次对其进行排班。这可能涉及到安排不同技能水平的操作人员，以确保生产线能够在高效率和高合格率下运行。实时调整排班计划，随着生产线的运行，实时监控生产效率和合格率的变化，及时调整操作人员的排班计划以适应变化的生产需求。结合其他因素进行综合考虑，如在排班计划中、操作人员的休息时间、加班安排、交接班等，以确保生产线的持续稳定运行。

**3 Python代码实现**

**3.1 问题一**

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** warnings

**import** seaborn **as** sns

warnings**.**filterwarnings('ignore')

plt**.**rcParams['font.sans-serif'] **=** 'SimSun' *# 换成自己环境下的中文字体，比如'SimHei'*

df **=** pd**.**read\_csv('A题-示例数据/附件1/M101.csv')

*# 查看数据前几行*

**print**(df**.**head())

*# 故障数据分析*

fault\_columns **=** [col **for** col **in** df**.**columns **if** '故障' **in** col]

fault\_counts **=** df[fault\_columns]**.**sum()

**print**(fault\_counts)

*# 可视化故障数据*

plt**.**figure(figsize**=**(10, 6))

colors **=** sns**.**color\_palette("hls", len(fault\_counts))

fault\_counts**.**plot(kind**=**'bar', color**=**colors)

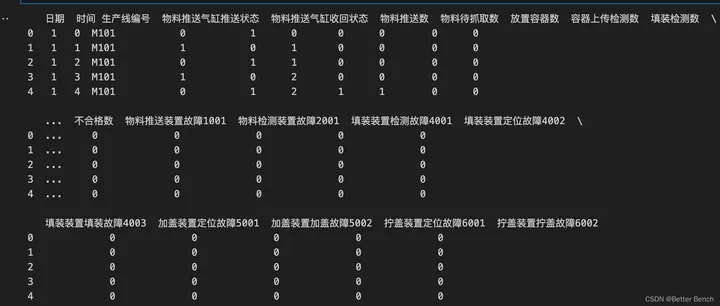
plt**.**title('Device Fault Counts')

plt**.**xlabel('Device')

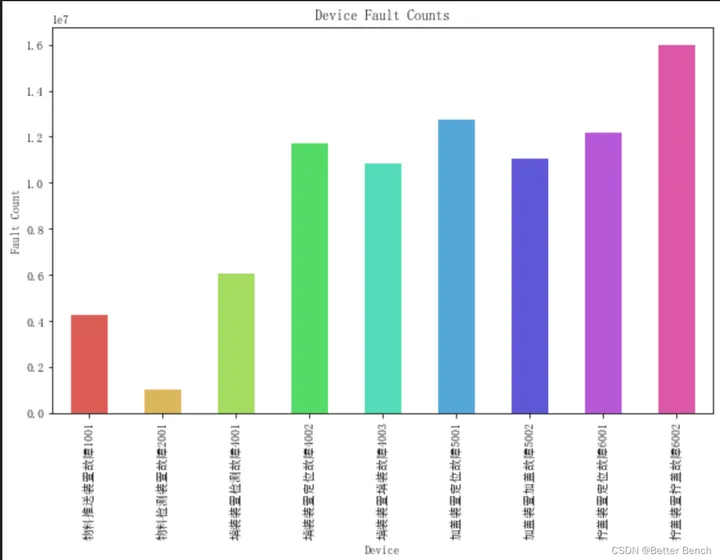
plt**.**ylabel('Fault Count')

plt**.**savefig('img/1.png',dpi**=**100)

plt**.**show()







*# 生产线编号分析*

line\_counts **=** df['生产线编号']**.**value\_counts()

**print**(line\_counts)

*# 可视化生产线编号分布*

plt**.**figure(figsize**=**(10, 6))

colors **=** sns**.**color\_palette("hls", len(line\_counts))

line\_counts**.**plot(kind**=**'bar',color **=** colors)

plt**.**title('Production Line Distribution')

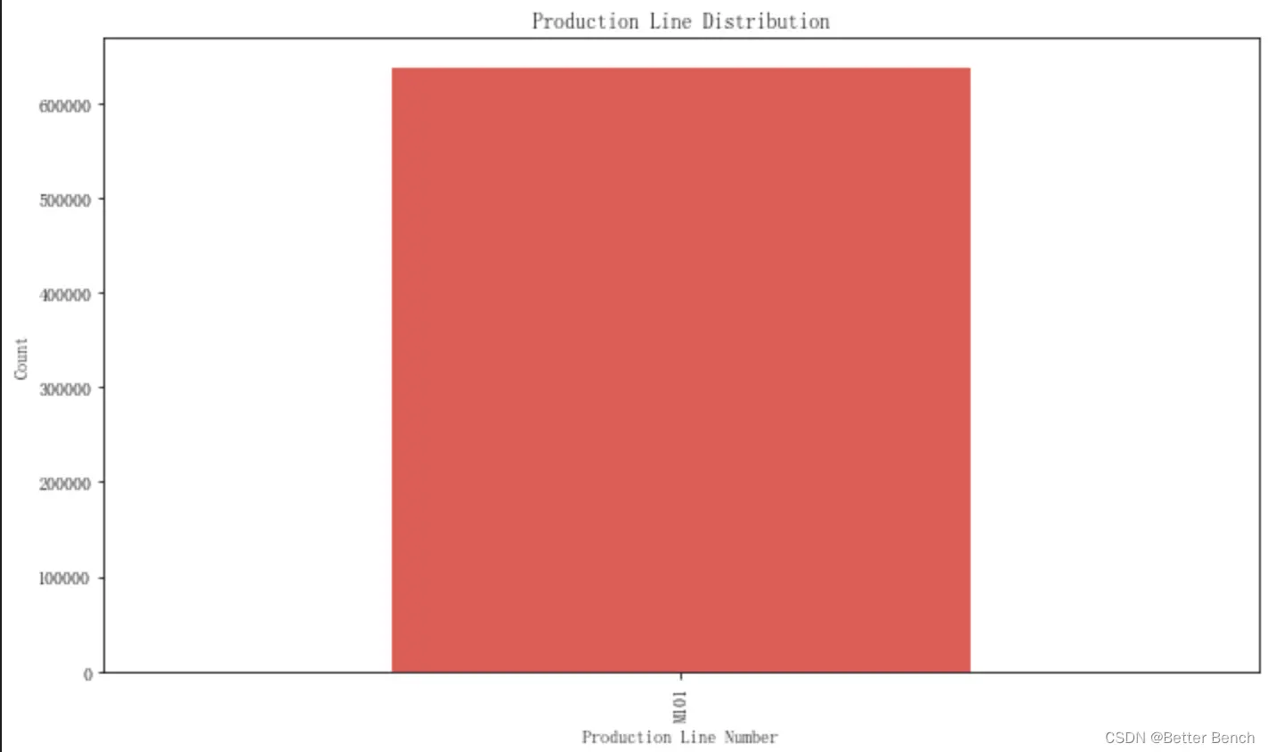
plt**.**xlabel('Production Line Number')

plt**.**ylabel('Count')

plt**.**tight\_layout()

plt**.**savefig('img/2.png',dpi**=**100)

plt**.**show()



**import** matplotlib.pyplot **as** plt

**import** warnings

**import** seaborn **as** sns

warnings**.**filterwarnings('ignore')

df **=** pd**.**read\_csv('A题-示例数据/附件1/M101.csv')

*# 物料推送气缸状态分析*

push\_counts **=** df['物料推送气缸推送状态']**.**value\_counts()

pull\_counts **=** df['物料推送气缸收回状态']**.**value\_counts()

push\_pull\_counts **=** pd**.**DataFrame({'Push': push\_counts, 'Pull': pull\_counts})

**print**(push\_pull\_counts)

plt**.**figure(figsize**=**(10, 6))

*# 可视化物料推送气缸状态*

colors **=** sns**.**color\_palette("hls", len(push\_pull\_counts))

push\_pull\_counts**.**plot(kind**=**'bar', stacked**=**True,color **=** colors)

plt**.**title('Material Push Cylinder State')

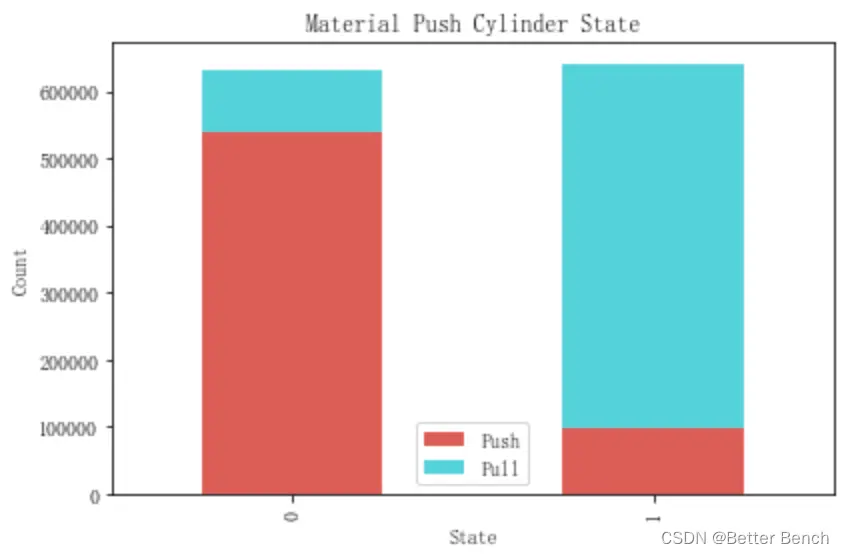
plt**.**xlabel('State')

plt**.**ylabel('Count')

plt**.**tight\_layout()

plt**.**savefig('img/3.png',dpi**=**100)

plt**.**show()



**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.metrics **import** confusion\_matrix, classification\_report

*# 统计各装置故障的情况*

fault\_columns **=** ['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']

*# 加载数据*

data **=** pd**.**read\_csv('A题-示例数据/附件1/M101.csv')

*# 构建故障报警模型*

*# 首先，选取特征和目标变量*

feature\_columns **=** ['时间', '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',

'填装数', '加盖数', '拧盖数', '合格数', '不合格数']

target\_columns **=** fault\_columns

data\_len **=** 10000

X **=** data[feature\_columns][0:data\_len]

*# 多标签问题*

y **=** data[target\_columns][0:data\_len]

*# 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)

*# 使用随机森林模型进行训练*

model **=** RandomForestClassifier()

model**.**fit(X\_train, y\_train)

*# 在测试集上进行预测*

y\_pred **=** model**.**predict(X\_test)

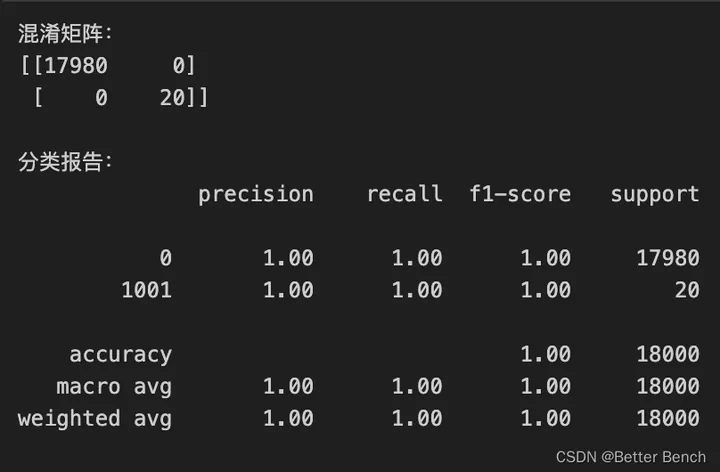
*# 评估模型性能*

**print**('混淆矩阵：')

**print**(confusion\_matrix(y\_test**.**values**.**flatten(), y\_pred**.**flatten()))

**print**('\n分类报告：')

**print**(classification\_report(y\_test**.**values**.**flatten(), y\_pred**.**flatten()))



**3.2 问题二**

**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.metrics **import** accuracy\_score

**from** datetime **import** datetime

**from** collections **import** defaultdict

*# 读取数据*

data **=** pd**.**read\_csv('A题-示例数据/附件1/M101.csv')

*# 定义故障字段*

fault\_columns **=** ['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']

data**.**replace({1001: 1, 2001: 1, 4001: 1, 4002: 1, 4003: 1, 5001: 1, 5002: 1, 6001: 1, 6002: 1}, inplace**=**True)

*# 故障报警日期、开始时间与持续时长*

fault\_info **=** defaultdict(list)

*# line\_number = data.groupby(['生产线编号'])*

**for** line\_number,line\_data **in** data**.**groupby(['生产线编号']):

**for** label **in** fault\_columns:

*# 训练数据集长度*

data\_length **=** 10000 *#只取部分数据，用于跑通baseline，应该为len(line\_data)*

*# 构建故障预警模型*

X **=** line\_data[line\_data**.**columns**.**difference(['日期', '时间','生产线编号'])][0:data\_length]

y **=** line\_data[label][0:data\_length]

*# 80%用于训练，20%用于测试*

*# train\_len = int(0.8\*len(X))*

*# X\_train, X\_test, y\_train, y\_test = X[0:train\_len],X[train\_len:],y[0:train\_len],y[train\_len:]*

*# 假设100%用于训练，100%用于测试*

X\_train, X\_test, y\_train, y\_test **=** X,X,y,y

**print**(f'生产线编号:{line\_number},故障编码:{label},训练集样本数:{len(X\_train)}，测试集样本数:{len(X\_test)}')

。。。略，请下载完整资料

acc **=** accuracy\_score(y\_test, y\_pred)

**print**("模型准确率：", acc)

*# 预测故障报警,假设全部数据集为附件2数据*

file2\_df **=** line\_data[line\_data**.**columns**.**difference(['日期', '时间','生产线编号'])]

predictions **=** rf**.**predict(file2\_df)

*# 用生产线编号和故障编号，组成唯一的键，来唯一表示某个生产线上的特定故障数据*

key **=** (line\_number,label)

*# 统计故障发生的日期、开始时间、持续时长*

**for** i **in** range(len(predictions) **-** 1):

**if** predictions[i] **==** 1 **and** predictions[i**+**1] **==** 0:

*# 只发生一次，持续时长为1*

**...**略

fault\_info[key]**.**append((happend\_date,happend\_time,happend\_duration))

**if** predictions[i] **==** 1 **and** predictions[i**+**1] **==** 1:

**if** len(fault\_info[key])**==**0:

*# 连续两次发生，持续时长为2*

**...**略

fault\_info[key]**.**append((happend\_date,happend\_time,happend\_duration))

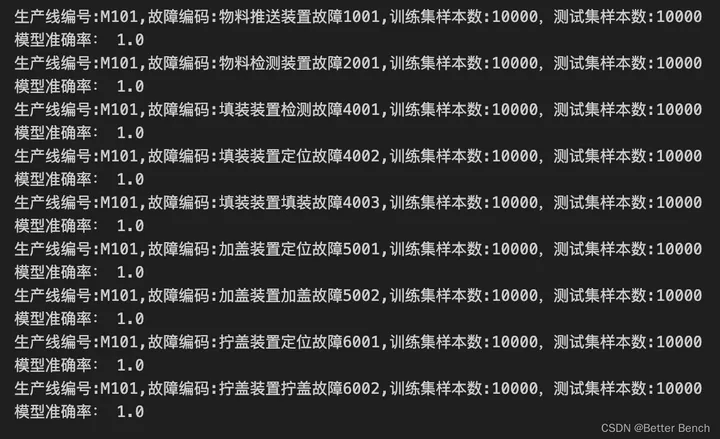
**else**:

*# 连续多次发生，持续时长为，当前时间-上一次发生时间*

**...**略

fault\_info[key][**-**1] **=** ((happend\_date,happend\_time,happend\_duration))

fault\_info



*# 将每条生产线，故障发生的日期、开始时间、持续时长数据，转为result2.xlsx格式*

fault\_info\_new **=** defaultdict(list)

**for** key **in** fault\_info**.**keys():

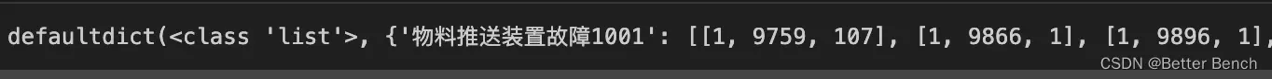
**if** len(fault\_info\_new[key[1]])**==**0:

fault\_info\_new[key[1]] **=** fault\_info[key]

**else**:

fault\_info\_new[key[1]] **+=** fault\_info[key]

**print**(fault\_info\_new)



result\_df\_list **=** []

**for** df\_cols **in** fault\_info\_new**.**keys():

**print**(df\_cols)

*# 将特定故障编号的数据，保存为单个表格*

temp\_error\_df **=** pd**.**DataFrame(fault\_info\_new[df\_cols],columns**=**['日期', '开始时间', '持续时长/秒'])

result\_df\_list**.**append(temp\_error\_df)

*# 横向合并所有故障编号的数据*

result2\_df **=** pd**.**concat(result\_df\_list,axis**=**1)

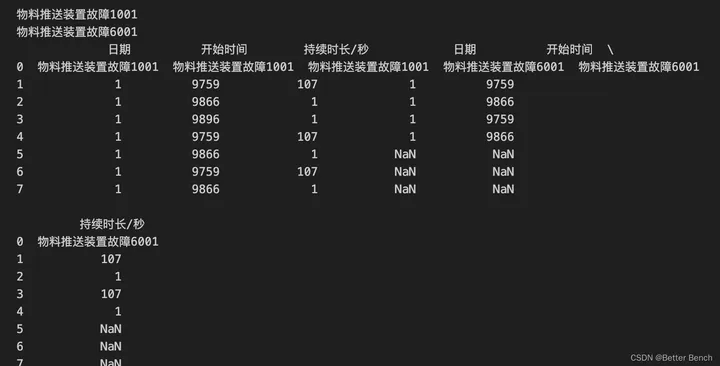
result2\_df**.**loc[**-**1] **=** [x **for** x **in** fault\_info\_new**.**keys() **for** \_ **in** range(3)] *# 添加故障编码到首行*

result2\_df**.**index **=** result2\_df**.**index **+** 1 *# 将索引整体后移*

result2\_df **=** result2\_df**.**sort\_index() *# 重新按索引排序*

result2\_df**.**to\_excel('./提交数据/result2.xlsx',index**=**False)

**print**(result2\_df)



*# 统计每条生产线的数据，故障发生的日期、开始时间、持续时长数据*

fault\_info\_new **=** defaultdict(list)

**for** key **in** fault\_info**.**keys():

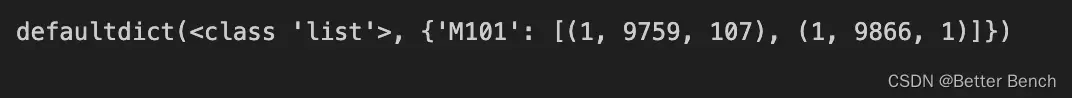
**if** len(fault\_info\_new[key[0]])**==**0:

fault\_info\_new[key[0]] **=** fault\_info[key]

**else**:

fault\_info\_new[key[0]] **+=** fault\_info[key]

**print**(fault\_info\_new)



**for** line\_n **in** fault\_info\_new**.**keys():

line\_fault\_info **=** fault\_info\_new[line\_n]

third\_elements **=** [x[2] **for** x **in** line\_fault\_info]

monthly\_fault **=** sum(third\_elements)

max\_duration **=** max(third\_elements)

min\_duration **=** min(third\_elements)

**print**(f"生产线:{line\_n},每月故障总次数:{monthly\_fault},最长时间:{max\_duration},最短时间:{min\_duration}")



**3.3 问题三、四**

请下载完整资料

**4 完整资料**



发布于 2024-03-11 17:25・IP 属地中国澳门

[生产线](https://www.zhihu.com/topic/19653365" \t "_blank)

[Python](https://www.zhihu.com/topic/19552832" \t "_blank)

[泰迪杯数据挖掘竞赛](https://www.zhihu.com/topic/27339634" \t "_blank)

​赞同 2​​4 条评论

​分享

​喜欢​收藏​申请转载

​



欢迎参与讨论

**4 条评论**

**默认**

**最新**

[](https://www.zhihu.com/people/5c018643a04c7a76d2f4e662b04dbd09)

[**Big Risk**](https://www.zhihu.com/people/5c018643a04c7a76d2f4e662b04dbd09)

目前分享出来的数据中，没有发现有重合故障的（同一时间两个故障）请问是用单标签模型还是多标签的？

03-16 · IP 属地四川

​回复​喜欢

[](https://www.zhihu.com/people/5c018643a04c7a76d2f4e662b04dbd09)

[**Big Risk**](https://www.zhihu.com/people/5c018643a04c7a76d2f4e662b04dbd09)

[**BetterBench**](https://www.zhihu.com/people/ac039ca45ec54f9fe17b3e6efa94e97d)

ok

昨天 10:19 · IP 属地四川

​回复​喜欢

[](https://www.zhihu.com/people/ac039ca45ec54f9fe17b3e6efa94e97d)

[**BetterBench**](https://www.zhihu.com/people/ac039ca45ec54f9fe17b3e6efa94e97d)

作者

我实现的是多单标签模型

03-16 · IP 属地中国澳门

​回复​喜欢

展开其他 1 条回复​

**文章被以下专栏收录**

[[](https://www.zhihu.com/column/c_1434248425172418560)](https://www.zhihu.com/column/c_1434248425172418560)

**[数学建模](https://www.zhihu.com/column/c_1434248425172418560)**

BetterBench全年各类数学建模竞赛笔记

**推荐阅读**

[[](https://zhuanlan.zhihu.com/p/109438532)](https://zhuanlan.zhihu.com/p/109438532)

**[乐高史上最难拼装的科技车模42110路虎评测](https://zhuanlan.zhihu.com/p/109438532)**

[积木控](https://zhuanlan.zhihu.com/p/109438532)

**[浅谈BB小玩具——误系列1](https://zhuanlan.zhihu.com/p/190180792)**

[不单单80后90后的那些人小时候玩的BB弹枪及BB弹很废柴，00后的也是如此。今天给大家展示一些我小时候用过的BB弹。 说实话，把这堆东西翻出来，实属不易。这一箱子满载着破烂和乐高，还有一…](https://zhuanlan.zhihu.com/p/190180792)

[逆転](https://zhuanlan.zhihu.com/p/190180792)

[[](https://zhuanlan.zhihu.com/p/24098458)](https://zhuanlan.zhihu.com/p/24098458)

**[当乐高遇见Arduino：乐高PM2.5 探测仪](https://zhuanlan.zhihu.com/p/24098458)**

[Ryan老...发表于科技学堂](https://zhuanlan.zhihu.com/p/24098458)

[[](https://zhuanlan.zhihu.com/p/77570597)](https://zhuanlan.zhihu.com/p/77570597)

**[乐高新科技——42099 4×4大脚车](https://zhuanlan.zhihu.com/p/77570597)**

[乐高Jac...发表于乐高创意与...](https://zhuanlan.zhihu.com/p/77570597)