天池工业AI大赛-智能制造质量预测

文档编号：TC-GYAI-ZNZZYC-004

发布日期：1/15/2018 9:57:06 PM

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本号 | 修订说明 | 修订人 | 备注 |
| 2018.01.04 | 001 | 目的、解题思路、每天思路 | 许鹏 |  |
| 2018.01.06 | 002 | 补齐2.2.5特征剔除 | 许鹏 |  |
| 2018.01.13 | 003 | 补齐2.4.4固定顺序划分 | 许鹏 |  |
| 2018.01.14 | 004 | 基于官网给定的testA结果，进一步分析结果 | 许鹏 |  |

目录

[1 赛题简介 1](#_Toc503715114)

[1.1 问题描述 1](#_Toc503715115)

[1.2 评测指标 1](#_Toc503715116)

[1.3 赛题修正 1](#_Toc503715117)

[2 结题思路 2](#_Toc503715118)

[2.1 简述 2](#_Toc503715119)

[2.2 数据预处理 3](#_Toc503715120)

[2.2.1 简述 3](#_Toc503715121)

[2.2.2 修改非数字类字符 3](#_Toc503715122)

[2.2.3 数据非线性修正 3](#_Toc503715123)

[2.2.4 数据补齐 4](#_Toc503715124)

[2.2.5 特征剔除（最重要） 4](#_Toc503715125)

[2.2.6 数据归一化 6](#_Toc503715126)

[2.2.7 数据保存 7](#_Toc503715127)

[2.3 降维（可选） 8](#_Toc503715128)

[2.4 线性回归（+正则化） 9](#_Toc503715129)

[2.4.1 线性回归方法概述 9](#_Toc503715130)

[2.4.2 数据随机打乱后第一种划分 9](#_Toc503715131)

[2.4.3 数据随机打乱后第二种划分 9](#_Toc503715132)

[2.4.4 数据随机打乱后第三种划分 9](#_Toc503715133)

[2.4.5 数据固定顺序划分 9](#_Toc503715134)

[3 每日思路 10](#_Toc503715135)

[3.1 2017.12.31 [0.43674] 10](#_Toc503715136)

[3.2 2017.01.01 [0.23662]： 10](#_Toc503715137)

[3.3 2017.01.02 [1.08823]： 10](#_Toc503715138)

[3.4 2017.01.03 [0.08080]： 10](#_Toc503715139)

[3.5 2017.01.04 [0.08800]： 11](#_Toc503715140)

[3.6 2017.01.05 [0.06664]： 11](#_Toc503715141)

[3.7 2017.01.06 [0.06387]： 12](#_Toc503715142)

[3.8 2017.01.07 [0.08303]： 13](#_Toc503715143)

[3.9 2017.01.08 [0.06414]： 13](#_Toc503715144)

[3.10 2017.01.09 [0.05916]： 13](#_Toc503715145)

[3.11 2017.01.10 [0.06924]： 13](#_Toc503715146)

[3.12 2017.01.11 [0.06087]： 13](#_Toc503715147)

[3.13 2017.01.12 [0.05852]： 13](#_Toc503715148)

[3.14 2017.01.13[0.13068]： 14](#_Toc503715149)

[3.15 2017.01.14[]： 15](#_Toc503715150)

[3.16 2017.01.15 []： 15](#_Toc503715151)

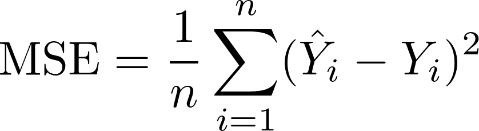
# 赛题简介

## 问题描述

比赛提供了生产线上的数据，反应机台的温度，气体，液体流量，功率，制成时间等因子。通过这些因子，需要选手设计出模型，准确的预测与之相对应的特性数值。这是一个典型的回归预测问题。因为数据中可能存在异常等现象，比赛鼓励选手发挥想象力，创造力，去设计出智能的算法。初赛阶段将提供3个星期左右的数据，复赛阶段提供数据待定，决赛阶段需要选手给出因子的重要性排序或者分类，供评委参考并作为最终评判指标的一个考量。

## ****评测指标****

MSE越小越好



本竞赛排行榜分A/B榜，A榜数据100条，B榜数据121条。最终线上成绩与排名以B榜成绩与排名为主（B榜重新排名，需要再提交答案，B榜成绩只在初赛最后两天公布）。

## 赛题修正

“训练.xlsx”两个ID466，将第二个改为ID4662，后来发现，这组数据的分布和其它差异太大，不适合做后续预测，所以直接删除了（如果不删除，要将前面很多的0置换为nan，否则相当不合理）

# 结题思路

## 简述

本题是一个典型的样板特征数（**8027**）远大于样本数（**500**）的问题。一般有三种方法：

1. 数据预处理（直接只抽取部分特征） + 线性回归 + 正则化（消除过拟合）；
2. 数据预处理 + 降维（PCA） + 线性回归；
3. 结合方法1和方法2：数据预处理 + 降维 + 线性回归 + 正则化；

对于方法1，需要对正则化的系数进行调整训练；对于方法2，需要对最终维度进行调整训练。

下面对这几个模块分别进行分析。

## 数据预处理

### 简述

观察数据发现以下问题：

1. 数据集：训练集总数500个（有标签），测试A为100个（无标签），测试B为120个（无标签）；
2. 每组数据均有多个数据是空缺的，需要补齐；
3. 每组数据中有些类别不是数字，是字母（例如：TOOL\_ID）；
4. 有些数据是时间日期（非线性变化），可能需要转换为时间戳（线性变化）
5. 每组数据之间的数量级相差很大；

基于这样的情况，需要针对所有数据（500+100+120）进行数据预处理，主要包括以下步骤：修改非数字类字符、非线性修正、数据补齐、特征剔除、数据归一化

### 修改非数字类字符

将字母转为ASCII码

### 数据非线性修正

1. 有些数据是时间（例如：2017078419130209.8），不能线性分析：

|  |  |
| --- | --- |
| 210X204 | 20170721052115 |
| 210X205 | 20170721051559 |
| 210X213 | 20170721052114 |
| 210X215 | 20170721051626 |
| 220X67 | 2017072119261900 |
| 220X71 | 2017072119245810 |
| 220X87 | 2017072119245390 |
| 220X91 | 2017072119260120 |
| 220X95 | 2017072119251050 |
| 220X225 | 20170721194037 |
| 220X226 | 20170721194037 |
| 220X234 | 20170713110723 |
| 220X235 | 20170705163823 |
| 220X236 | 20170705163823 |
| 330X623 | 20170804 |
| 330X640 | 20170804 |
| 。。。 |  |

所有时间数值对应的列的index（共计105个）：

24 204 205 212 214 282 284 286 288 290 292 294 296 404 405 406 407 408 409 410 411 412 413 414 415 753 754 755 757 758 760 761 764 765 771 824 826 828 830 832 834 836 838 971 972 973 974 1040 1041 1127 1128 1312 1313 1497 1498 1682 1683 3004 3021 3546 3549 3550 4859 4878 4879 5455 5459 5460 5461 5626 5627 5640 5642 5669 5670 5673 5674 5692 5693 5760 5761 5797 5798 5808 5810 5823 5825 5941 6338 6341 6357 6359 6417 6419 6504 6506 6512 6514 7265 7284 7285 7861 7865 7866 7867

注：统计后发现，训练、测试A、测试B中的时间分布确实差异较大，所以这里的矫正应该很有必要，否则会造成数据集的差异导致预测结果不准确。

1. 不明数据：

|  |  |
| --- | --- |
| 210X206 | 1797544036 |
| 210X207 | 2622970737 |
| 210X216 | 2731696826 |
| 210X217 | 2622970791 |
| 330X647 | 300000000000000 |
| 330X648 | 300000000000000 |
| 360X717 | 17000000000000 |
| 750X717 | 2400000000000 |
| 。。。 |  |

1. 很多9999、999、1E-10、100、4000、25、0、3000数据，是不是量程阈值

### 数据补齐

随机性数据，采用

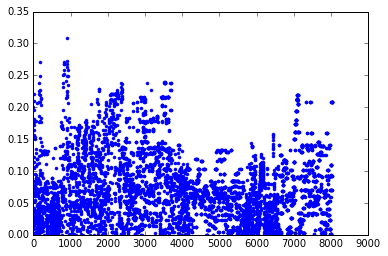
1. 均值补全
2. 中值补全

### 特征剔除（最重要）

特征太多，冗余太大，计算量太大。

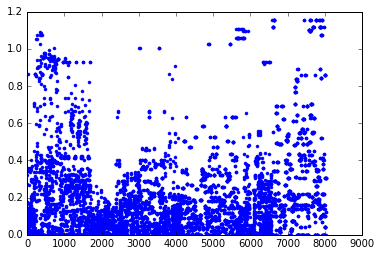
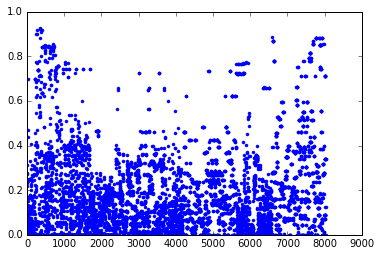
1. 全部为nan、np.nanstd = 0的可以删除（检测有1080列）
2. 如果所有样本中，对于某个特征值，如果和预测Y的相关性极低（理论上包含上面一条的所有特征数据），可以直接删除，使用协方差计算相关系数判断

参考：<http://mp.weixin.qq.com/s/oejfQS-705PI5DhmC4AAug>

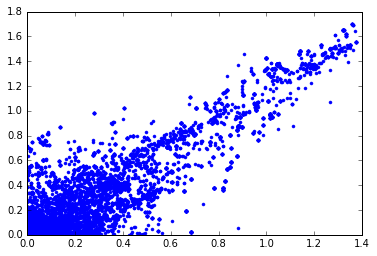


每个特征的相关系数分布情况

1. 当前cv值和系统的test差异很大，说明样本偏差很大，所以可以选择train和test中数值差异很大的直接去除掉。



每个特征的（训练、测试A的均值差别）、（训练、测试B的均值差别）分布情况

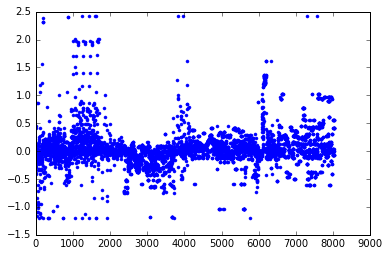
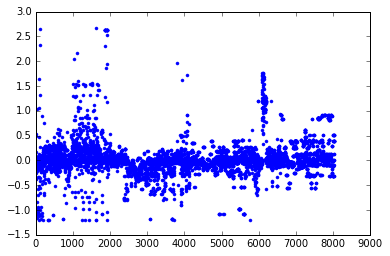


（训练、测试A的均值差别）~（训练、测试B的均值差别）

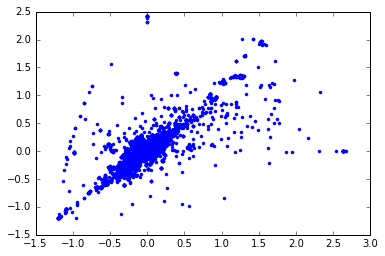
可以考虑将均值差异大于1、0.5、0.2、0.1、0.05的分别剔除出来，然后只对剩余的进行分析，如果训练集和CV的误差可以到0.02，那么就可以了。

理论上：抽掉特征越多，train和cv偏差越大，cv和test偏差越小；

1. 均值比较接近，但是测试A测试B的方差远大于训练集的，也是无法准确预测的，理论上也是需要删除的



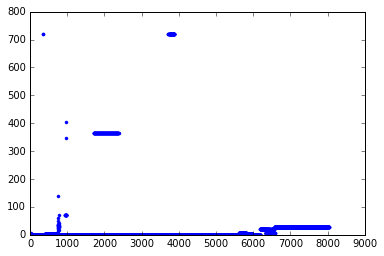
每个特征的（测试A标准差-训练标准差）、（测试B标准差-训练标准差）分布情况



（测试A标准差-训练标准差）~（测试B标准差-训练标准差）

1. 数据补齐个数很多的直接去除

由于补齐的方法，对数据本身进行了修正，这个里面存在极大的不确定性，可以考虑剔除



720组数据中，每个特征的补齐数据个数分布情况

### 数据归一化

其中：是平均值，是标准差。

### 数据保存

保存为excel文件

## 降维（可选）

针对训练集（开放式）还是所有数据（比赛式）进行

如果选择降维计算，这里针对所有数据进行降维处理。

## 线性回归（+正则化）

### 线性回归方法概述

参考：<http://mp.weixin.qq.com/s/ZbT5ASvnTODkodTJjBGhnw>

这里采用：

1. scipy.optimize. minimize函数实现 data\_linear\_regression v1, v2, v3, v5
2. sklearn.linear\_model. LinearRegression函数实现 data\_linear\_regression v4, v6

### 数据随机打乱后第一种划分

Train：300

CV：100

Test：100

### 数据随机打乱后第二种划分

Train：400

CV：95

Test：5

### 数据随机打乱后第三种划分

Train：400

CV：100

### 数据固定顺序划分

通过观察数据可以发现，训练集的数据分布还有有些规律了，例如时间上，按序列号是不断增加的，所以为了进一步预测将来的数据，可以考虑使用较旧的数据作为训练集，较新的数据作为验证集，具体如下：

Train：400 （前400）

CV：100 （后100）

# 每日思路

## 2017.12.31 [0.43674]

初步建立模型，提交第一次答案；

## 2017.01.01 [0.23662]：

优化data\_preprocess步骤；

## 2017.01.02 [1.08823]：

将数据中的所有时间变量修正为时间戳；

## 2017.01.03 [0.08080]：

将所有均值差异（训练和测试A、测试B比较）较大的特征值剔除出去；

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→) | | | | | |
| PCA阈值  （数据越来越少↓） |  | 1  (6811) | 0.5  (6197) | 0.2  (3502) | 0.1  (1463) | 0.05  (597) |
| 100 |  |  |  | 0.005/0.022  **(0.08080)** | 0.012/0.032 |
| 99 |  | 0.006/0.026 | 0.008/0.023 | 0.011/0.022 | 0.022/0.036 |
| 95 | 0.011/0.021 | 0.009/0.049 | 0.011/0.028 | 0.018/0.031 | 0.027/0.030 |
| 92 |  |  |  | 0.021/0.034 |  |
| 90 | 0.014/0.020 | 0.013/0.030 | 0.014/0.046 | 0.021/0.031 | 0.032/0.031 |
| 88 |  |  |  | 0.023/0.031 |  |
| 85 | 0.018/0.025  **(0.10383)** |  | 0.019/0.024 | 0.026/0.039 | 0.036/0.039  **(0.10032)** |
| 80 |  | 0.023/0.037 |  | 0.038/0.038 |  |

## 2017.01.04 [0.08800]：

计算训练数据中和Y相关度，将相关度较低、nan很多的特征直接剔除（correlation\_threshold = 0.05，nan\_num\_use > 100）；

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→) | | | | | |
| PCA阈值  （数据越来越少↓） |  | 1  (4200) | 0.5  (3795) | 0.2  (2025) | 0.1  (838) | 0.05  (304) |

（correlation\_threshold = 0.02，nan\_num\_use > 100）；

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | | | |
| PCA阈值  （数据越来越少↓） |  | 1  (5630) | 0.5  (5065) | 0.2  (2833) | 0.1  (1155) | 0.05  (447) |
| 100 |  |  |  | 0.007/0.025 | 0.015/0.021  **(0.06664)** |
| 99 | 0.007/0.015  **(0.08800)** | 0.007/0.018 | 0.010/0.025 | 0.012/0.023 | 0.018/0.031 |
| 95 | 0.014/0.016 |  |  | 0.017/0.026 | 0.028/0.033 |
| 92 | 0.014/0.017 | 0.014/0.028 | 0.016/0.023 | 0.021/0.033 | 0.032/0.041 |
| 90 | 0.015/0.027 |  |  | 0.026/0.035 | 0.043/0.038 |
| 88 | 0.019/0.018 | 0.017/0.029 | 0.020/0.030 | 0.028/0.032 | 0.039/0.049 |
| 85 | 0.018/0.022 |  |  | 0.035/0.040 | 0.043/0.056 |
| 80 | 0.023/0.033 | 0.030/0.023 | 0.027/0.038 | 0.038/0.056 | 0.084/0.104 |

## 2017.01.05 [0.06664]：

提交上图中0.05/100的结果

## 2017.01.06 [0.06387]：

去除训练集和测试集方差差别大的数据

（correlation\_threshold = 0.05，nan\_num\_use > 20，std\_diff\_threshold = 0.1）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| 次数 |  | 0.2  (1397) | 0.1  (690) | 0.05  (**241**) |
| 1 | 0.010/0.019 | 0.009/0.021 | 0.011/0.021  **(0.08303)** |

（correlation\_threshold = 0.05，nan\_num\_use > 100，std\_diff\_threshold = 0.05）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| 次数 |  | 0.2  (1315) | 0.1  (635) | 0.05  (246) |
| 1 | 0.010/0.024 | 0.010/0.023 | 0.010/0.027 |

（correlation\_threshold = 0.02，nan\_num\_use > 100，std\_diff\_threshold = 0.2）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| 次数 |  | 0.2  (2635) | 0.1  (1103) | 0.05  (428) |
| 1 |  | 0.008/0.027 | 0.016/0.024 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.05）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| 次数 |  | 0.2  (1629) | 0.1  (831) | 0.05  (364) |
| 1 | 0.008/0.029 | 0.010/0.026 | 0.015/0.022 |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| PCA阈值  （数据越来越少↓） |  | 0.8  (2545) | 0.6  (2513) | 0.4  (2445) |
| 99 | 0.008/0.017 | 0.007/0.022 | 0.011/0.024 |
| 95 | 0.012/0.025 | 0.014/0.020 | 0.015/0.017  **(0.06414)** |
| 90 | 0.018/0.021 | 0.022/0.024 | 0.017/0.033 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.1）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| 次数 |  | 0.2  (1994) | 0.1  (967) | 0.05  (378) |
| 1 | 0.007/0.021  **(0.05852)** | 0.009/0.023 | 0.014/0.024 |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| PCA阈值  （数据越来越少↓） |  | 0.8  (3110) | 0.6  (3073) | 0.4  (2976) |
| 99 | 0.006/0.016 | 0.008/0.024 | 0.009/0.023  **(0.06087)** |
| 95 | 0.013/0.019 | 0.012/0.024 | 0.012/0.022 |
| 90 | 0.015/0.026 | 0.019/0.016 | 0.019/0.021 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.2）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
| PCA阈值  （数据越来越少↓） |  | 0.8  (3701) | 0.6  (3660) | 0.4  (3505) |
| 99 | 0.008/0.020 | 0.008/0.021 | 0.009/0.017  **(0.06387)**  **(0.05916)**  **(0.06924)** |
| 95 | 0.013/0.020 | 0.013/0.023 | 0.012/0.027 |
| 90 | 0.017/0.017 | 0.018/0.024 | 0.019/0.017 |

## 2017.01.07 [0.08303]：

使用2017.01.06数据；

## 2017.01.08 [0.06414]：

使用2017.01.06数据；

## 2017.01.09 [0.05916]：

使用2017.01.06数据；

## 2017.01.10 [0.06924]：

使用2017.01.06数据；

## 2017.01.11 [0.06087]：

使用2017.01.06数据；

## 2017.01.12 [0.05852]：

使用2017.01.06数据；

## 2017.01.13[0.13068]：

将训练数据中划分train、dev更加合理化，保证dev的特征（均值分布）和测试A相近（直接顺序划分训练集和测试集），再训练计算；

（correlation\_threshold = 0.05，nan\_num\_use > 20，std\_diff\_threshold = 0.1）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  |  | 0.2  (1397) | 0.1  (690) | 0.05  (241) |
|  | 0.009/0.049 | 0.012/0.031 | 0.011/0.026 |

（correlation\_threshold = 0.05，nan\_num\_use > 100，std\_diff\_threshold = 0.05）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  |  | 0.2  (1315) | 0.1  (635) | 0.05  (246) |
|  | 0.010/0.046 | 0.008/0.029 | 0.013/0.024 |

（correlation\_threshold = 0.02，nan\_num\_use > 100，std\_diff\_threshold = 0.2）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  |  | 0.2  (2635) | 0.1  (1103) | 0.05  (428) |
|  |  | 0.008/0.028 | 0.017/0.033 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.05）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  | 0.4  (2445) | 0.2  (1629) | 0.1  (831) | 0.05  (364) |
|  | 0.008/0.031 | 0.010/0.028 | 0.016/0.030 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.1）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  | 0.4  (2976) | 0.2  (1994) | 0.1  (967) | 0.05  (378) |
|  | 0.007/0.040 | 0.009/0.025 | 0.015/0.029 |

（correlation\_threshold = 0.02，nan\_num\_use > 20，std\_diff\_threshold = 0.2）；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/dev | Diff阈值(数据越来越少→，与测试AB越接近) | | | |
|  | 0.4  (3505) | 0.4\_pca\_99  (302) |  |  |
|  | 0.006/0.036 |  |  |

从上面结果来看，预测的结果都较差，其实本质反应的是，这个题目不是很好，就算训练很好，测试组A或B预测结果很准确，但是针对将来的数据，预测性也不会很好，因为这不是一个随机分布的数据。

## 2017.01.14[0.05756]：

（工作一）：增加testA数据进行训练【temp\_data\_v6\_v5】

|  |  |  |
| --- | --- | --- |
| **数据剔除** | **Pca处理** | **结果** |
| (0.02\_20\_0.1) | mean\_diff\_0.2 | 0.008/0.020  **(0.05756)** |
| 0.009/0.021 |
| 0.008/0.023 |
| (0.02\_20\_0.2) | mean\_diff\_0.4\_pca\_99 | 0.010/0.020 |
| 0.010/0.023 |
| 0.010/0.020 |

（工作二）：新的回归模型【temp\_data\_v6\_v6】

参考：

<https://github.com/demonicCode/Intelligent-manufacturing?spm=5176.9876270.0.0.7ff8121fBXcvP1>

使用函数sklearn.linear\_model. LinearRegression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0.02\_20\_0.1** | **0.02\_20\_0.2** | **0.02\_20\_0.05** | **0.02\_100\_0.2** | **0.05\_20\_0.1** | **0.05\_100\_0.05** |
| 0.075 | 0.054 | 0.068 |  | 0.040 | 0.045 |
|  | 0.063 | 0.086 |  | 0.045 | 0.053 |
|  | 0.098 |  |  | 0.050 | 0.056 |
|  | 0.118 |  |  | 0.056 | 0.061 |
|  | 0.206 |  |  | 0.063 | 0.071 |

**生成的预测“测试B”数据里面，有很多奇异值，明显不正确。**

## 2017.01.15 []：

使用2017.01.14数据；