参赛编号: YRDMCM2021

选题: B 参赛赛道: 研究生

**第一届长三角高校数学建模竞赛**

题 目 基于多元线性回归模型的锅炉水冷壁温度曲线研究

摘 要

在燃煤发电过程中，锅炉是一种重要的热能动力设备。锅炉的主要受热部分是水冷壁，其内部为流动的水用于吸收炉膛中高温燃烧产生的辐射热量，水受热蒸发产生高压蒸汽。影响水冷壁温度的因素有很多，本文旨在通过建立多元线性回归模型对水冷壁温度变化曲线进行分析和优化研究。

**问题一中**：基于各个水冷壁管道的共5000组温度数据，从集中趋势、离中趋势及分布形态三个方面对数据进行描述。我们使用均值、标准差、最小值、最大值、2阶中心距、波峰、趋势斜率等统计特征对10个水冷壁管道的温度时间序列数据进行分析，并辅以各管道温度变化的时序图、直方图与核密度图和箱型图等综合分析管道温度的变化情况。

**问题二中**：在问题一对各管道温度时间序列数据分析的基础上，主要从题目中给出的温度变化的平稳性和水冷壁温度不宜过高两个方面来对水冷壁管道的工作状态进行评价。根据超温警戒线阈值445℃，以及通过ADF检验判断时间序列的平稳性和各管道的标准差等多个维度，从10个水冷壁管道的温度曲线中选取最差工作曲线和最优工作曲线。

**问题三中**：基于题目给出的影响水冷壁温度的153个输入变量，首先对高维数据进行降维以便计算和可视化，采用线性拟合从中选取23个重要性变量；其次，在多个常规机器学习方法：线性回归、岭回归、套索回归、随机森林、XGBoost、AdaBoost、神经网络中，通过试验，选择效果最优的线性回归模型对10个水冷壁管道温度变化规律进行建模，并求解出各管道温度变化的线性回归方程。通过训练集和测试集拟合情况和模型拟合残差的分析，发现该模型拟合效果良好。

**问题四中**：本文共给出了影响水冷壁温度的153个变量，由于控制变量操作人员能够根据实际生产的需要而进行调节，而状态变量操作人员无法直接调节，因此本文通过皮尔逊相关系数分析管道10温度与111个操作变量之间的线性关系，并基于给定的样本点和温度值数据分析和定位引发超温现象的主要操作变量。

**问题五中**：首先在问题四的基础上，依据引发超温现象的主要操作变量，通过多元回归拟合得出相应回归方程。其次，通过最小化样本方差建立模型，从而优化第10个水冷壁管道超温段温度曲线，并基于遗传算法求解目标优化模型。

**关键词**：多元线性回归、水冷壁温度曲线、遗传算法

**目录**

[**一、问题提出** 1](#_Toc72693223)

[**1.1问题背景** 1](#_Toc72693224)

[**1.2问题重述** 1](#_Toc72693225)

[**二、问题分析** 2](#_Toc72693226)

[**2.1问题的整体分析** 2](#_Toc72693227)

[**2.2问题一分析** 3](#_Toc72693228)

[**2.3问题二分析** 3](#_Toc72693229)

[**2.4问题三分析** 3](#_Toc72693230)

[**2.5问题四分析** 3](#_Toc72693231)

[**2.6问题五分析** 4](#_Toc72693232)

[**三、模型假设** 4](#_Toc72693233)

[**四、符号说明** 4](#_Toc72693234)

[**五、模型的建立与求解** 4](#_Toc72693235)

[**5.1问题一求解** 4](#_Toc72693236)

[**5.1.1问题分析** 4](#_Toc72693237)

[**5.1.2 温度值统计特征分析** 4](#_Toc72693238)

[**5.1.3 温度值图例分析** 5](#_Toc72693239)

[**5.2问题二求解** 7](#_Toc72693240)

[**5.2.1问题分析** 7](#_Toc72693241)

[**5.2.2管道温度数据曲线分析** 8](#_Toc72693242)

[**5.3问题三求解** 9](#_Toc72693243)

[**5.3.1问题分析** 9](#_Toc72693244)

[**5.3.2数据降维** 10](#_Toc72693245)

[**5.3.3模型分析** 11](#_Toc72693246)

[**5.3.4模型构建** 11](#_Toc72693247)

[**5.3.5模型求解** 12](#_Toc72693248)

[**5.4问题四求解** 12](#_Toc72693249)

[**5.4.1 问题分析** 12](#_Toc72693250)

[**5.4.2皮尔逊相关系数分析** 13](#_Toc72693251)

[**5.5问题五求解** 14](#_Toc72693252)

[**5.5.1 模型建立** 14](#_Toc72693253)

[**5.5.2 遗传算法求解目标优化模型** 15](#_Toc72693254)

[**5.5.3 模型的求解** 16](#_Toc72693255)

[**六、模型优缺点** 17](#_Toc72693256)

[**6.1模型的优点** 17](#_Toc72693257)

[**6.2模型的缺点** 17](#_Toc72693258)

[**6.2模型的改进** 17](#_Toc72693259)

[**参考文献** 18](#_Toc72693260)

[附录1.问题三的相关图表 19](#_Toc72693261)

[附录2.Python代码 22](#_Toc72693262)

**基于多元线性回归模型的锅炉水冷壁温度曲线研究**

**一、问题提出**

**1.1问题背景**

在燃煤发电过程中，锅炉是一种重要的热能动力设备。它通过在炉膛中燃烧煤粉释放热量，将水加热成一定温度（或压力）的蒸汽，蒸汽再推动汽轮机旋转并驱动发电机发电。锅炉的主要受热部分是水冷壁，通常由数排钢管组成，分布于锅炉炉膛的四周，其内部为流动的水，用于吸收炉膛中高温燃烧产生的辐射热量，水受热蒸发产生高压蒸汽。水冷壁的结构如图1.1.1所示。

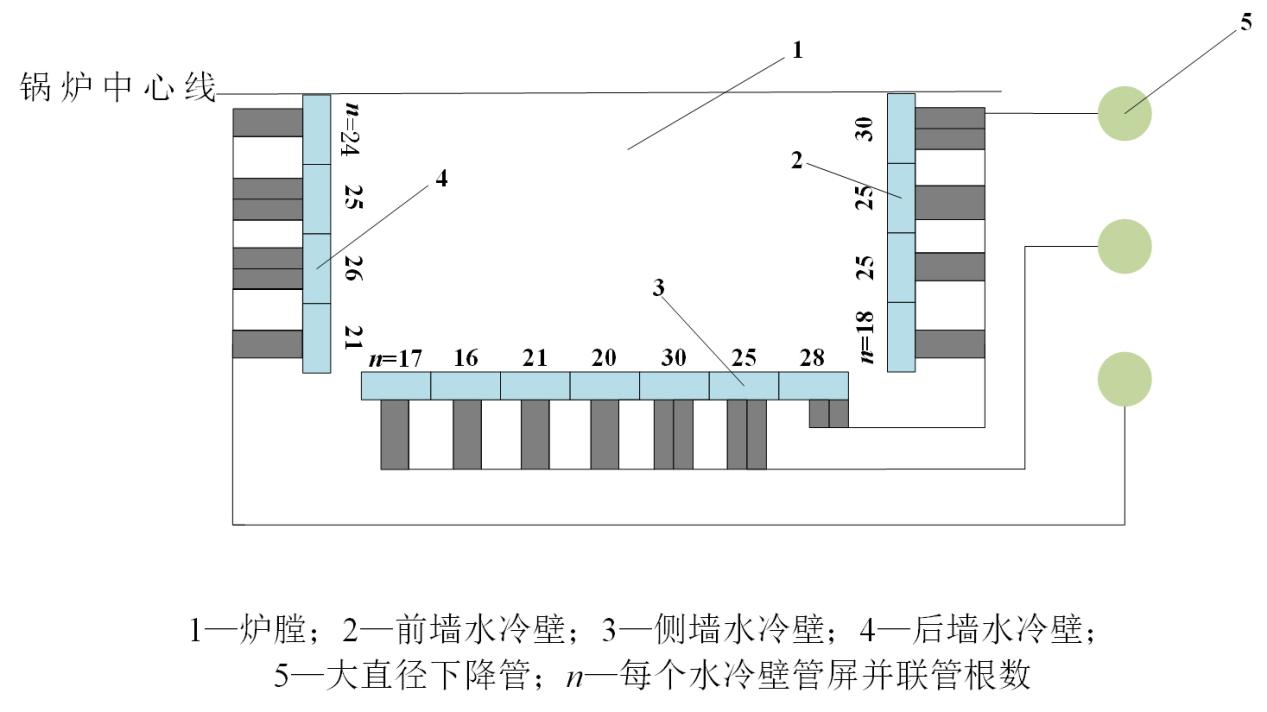


图1.1.1 锅炉水冷壁截面示意图

在实际生产过程中，希望水冷壁的温度变化尽可能平稳，同时为保证安全，水冷壁温度不宜过高，否则有烧坏的风险。按照实际经验，操作人员给出的水冷壁温度超温报警线为445℃。影响水冷壁温度的因素有很多，包括锅炉负荷、蒸汽温度、蒸汽压力、燃料量、水煤比等。

由于影响水冷壁温度的因素众多，因此需要建立一定的数学模型对其温度变化曲线进行分析和优化研究。

**1.2问题重述**

本题中给出了10个具有代表性水冷壁管道的温度值，采样周期为15s，共5000组数据；同时给出了影响水冷壁温度的153个输入变量的5000组数据，其中包括111个操作变量和42个状态变量（注1）。请利用这些信息和数据，建立数学模型解决以下问题。

1. 统计分析各个水冷壁管道的温度数据，并给出刻画这些温度时间序列数据变化情况的特征。
2. 请对附件1中10个水冷壁管道的温度数据曲线进行评价，确定其中的最优工作曲线和最差工作曲线。
3. 请利用附件1和附件2中的数据，分别建立10个水冷壁管道温度变化规律的数学模型，并对模型效果进行评价。
4. 第10个水冷壁管道温度曲线如图1.2.1所示，从图中可以看出，在第3172个样本点后水冷壁出现明显的超温现象，请基于给出的数据，分析并定位引发超温现象的主要操作变量。
5. 请针对第10个水冷壁管道温度曲线超温段建立优化模型，给出该超温段从第3172个样本开始的最优调节策略，满足操控的变量数尽量少、操作变量的调控量尽量小、优化调节后的工作曲线与问题2中的最优工作曲线的特征尽量吻合。

|  |
| --- |
|  |
| 图1.2.1 第10根水冷壁管道温度曲线时间序列图 |

**注1：**这里的操作变量，也常称作控制变量，是指在锅炉燃烧过程中，操作人员能够根据实际生产的需要而进行调节的量；而状态变量是用来描述锅炉燃烧系统运行状态的量，它的取值由相关检测设备采集得到，操作人员无法进行直接调节。

**二、问题分析**

**2.1问题的整体分析**

从整篇文章来看，是对水冷壁管道温度变化曲线进行评价和优化的问题。第一问，是基于各个水冷壁管道的温度时间序列数据，利用统计特征描述各管道温度变化特征和情况。第二问，是在第一问的基础上，对水冷壁管道的工作状态进行评价，从10个管道的温度变化曲线中选取最差工作曲线和最优工作曲线。第三问，从影响温度变化的多个因素中，选取主要变量分别建立10个水冷壁管道温度变化的多元线性回归模型。第四问，基于给定的样本点和温度值数据分析和定位引发超温现象的主要操作变量。第五问，是在第四问的基础上，依据引发超温现象的主要操作变量建立多元线性回归方程，通过最小化样本方差建立模型，并使用遗传算法求解。

|  |
| --- |
| 水冷壁管道的温度时间序列数据特征分析  水冷壁管道超温段建立优化模型  水冷壁管道温度变化规律模型建立  水冷壁管道温度变化曲线分析与评价 |
| 图2.1.1 整体分析图 |

从上图中可以看出，给出的问题是层层递进的，问题与问题之间有着紧密的相关性。

**2.2问题一分析**

数据分布特征可以从集中趋势、离中趋势及分布形态三个方面进行描述。平均指标是在反映总体的一般水平或分布的集中趋势的指标，包括位置平均数和数值平均数，常用的指标有众数、中位数、平均值等；变异指标是用来刻画总体分布的变异状况或离散程度的指标，测定离中趋势的指标有极差、平均差、四分位差、方差和标准差、以及离散系数等；矩、偏度和峰度是反映总体分布形态的指标。

因此，针对各个水冷壁管道的温度数据分析，我们使用均值、最大值、最小值、标准差、2阶中心距、波峰、趋势斜率等统计特征指标来分析，并辅以各管道温度变化的时序图、直方图与核密度图和箱型图等进行综合分析温度时间序列数据的变化情况。

**2.3问题二分析**

在问题一对各管道温度时间序列数据分析的基础上，我们从题目中温度变化的平稳性和水冷壁温度不宜过高两个方面来对水冷壁管道的工作状态进行评价，根据超温警戒线阈值445℃，以及通过ADF检验判断时间序列的平稳性和各管道的标准差等多个维度，从中选取最差工作曲线和最优工作曲线。

**2.4问题三分析**

本题给出的影响水冷壁温度的输入变量多达153个，首先对高维数据进行降维以便计算和可视化，我们采用线性拟合从中选取23个重要性变量；其次，在多个常规机器学习方法：线性回归（Linear Regression）、岭回归（Ridge Regression）、套索回归（Lasso Regression）、随机森林（Random Forest）、XGBoost、AdaBoost、神经网络（MLP）中，通过试验，我们选择了得到效果最优的线性回归模型对10个水冷壁管道温度变化规律进行建模，并求解出各管道温度变化的线性回归方程。

**2.5问题四分析**

本文共给出了影响水冷壁温度的153个变量，由于控制变量操作人员能够根据实际生产的需要而进行调节，而状态变量操作人员无法直接调节，因此本文通过皮尔逊相关系数分析管道10温度与111个操作变量之间的线性关系，并基于给定的样本点和温度值数据分析和定位引发超温现象的主要操作变量。

**2.6问题五分析**

在问题四分析的基础上，首先依据引发超温现象的主要操作变量，通过多元回归拟合得出相应关系式。其次，通过最小化样本方差建立模型，从而优化第10个水冷壁管道超温段温度曲线，并基于遗传算法求解目标优化模型。

**三、模型假设**

1. 假设建模所用数据真实可靠。

2. 假设水冷壁温度变化仅受附件2所给的因素影响。

3. 假设所选指标能够代表所要研究对象的整体情况。

4. 假设变量间的相关性不会对回归造成影响。

5. 假设相对重要性可量化。

**四、符号说明**

|  |  |
| --- | --- |
| **符号** | **含义** |
| MV | 操作变量 |
| SV | 状态变量 |
| ρ | 相关系数 |
| y | 管道温度 |

**注**：本文未在符号说明处声明的符号会在文中声明。

**五、模型的建立与求解**

**5.1问题一求解**

**5.1.1问题分析**

通过附件一10个水冷壁管道在采样周期为15s下的温度值数据，基于水冷壁管道温度的时间序列统计特征研究，由此分析温度值的均值、标准差、2阶中心距、波峰、趋势斜率，并根据其10个管道温度值的时序图、直方图、核密度图和箱型图等，分析得出各个水冷壁管道的温度值分布及变化情况。

**5.1.2 温度值统计特征分析**

通过对10个水冷壁管道5000组数据的分析，得出各个管道温度值的相关统计特征，包括均值、标准差、最小值、最大值、2阶中心距、波峰、趋势斜率等特征，具体数据如表5.1所示：

表5.1 温度值统计特征

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **管道** | **均值** | **标准差** | **最小值** | **最大值** | **2阶中心距** | **波峰** | **趋势斜率** |
| 管道1 | 374.31 | 21.45 | 338.6 | 420.2 | 460.09 | 157 | -0.00651 |
| 管道2 | 397.90 | 16.85 | 356.8 | 426.6 | 283.94 | 106 | -0.00276 |
| 管道3 | 396.70 | 10.67 | 364.2 | 420.2 | 113.95 | 139 | -0.00094 |
| 管道4 | 387.60 | 16.37 | 350.6 | 420.9 | 268.11 | 255 | -0.00489 |
| 管道5 | 391.36 | 12.97 | 360 | 419.9 | 168.38 | 229 | -0.00248 |
| 管道6 | 376.84 | 16.65 | 338 | 409.5 | 277.39 | 117 | -0.00464 |
| 管道7 | 386.18 | 16.02 | 350.6 | 416.5 | 256.64 | 121 | -0.00328 |
| 管道8 | 378.68 | 17.21 | 351.2 | 421.3 | 296.14 | 778 | -0.00428 |
| 管道9 | 408.03 | 17.70 | 368.7 | 454.3 | 313.45 | 241 | 0.005998 |
| 管道10 | 424.90 | 18.69 | 378.8 | 465.4 | 349.39 | 192 | 0.007902 |

由此可以分析出，各个管道的温度在338℃～466℃的范围内波动。管道1～管道8的温度均值均在400℃以下，而管道9和管道10的温度均值在400℃以上，且最高温度值均高于超温报警线445℃，温度过高，有烧坏风险。从标准差来看，管道1和管道10的标准差明显高于其他管道，温度变化不够平稳；而管道3和管道5的标准差相比于其他管道较小，说明温度变化相对较平稳。从2阶中心距也可以得出上述结果。

**5.1.3 温度值图例分析**

**1.时序图分析**

图5.1.1展示了10个水冷壁管道温度的时序图。

|  |
| --- |
|  |
| 图5.1.1 10个水冷壁管道温度的时序图 |

由时序图可以得出，10个水冷壁管道的温度变化范围基本在338℃～466℃以内。其温度变化均有一定的趋势，在取样点1000附近时，各个管道的温度都在平缓地上升，并达到最高值。随后各管道的温度在小范围内波动并以较快的速度下降到较低值，在取样点3500左右时又缓慢回升，最后趋于稳定温度。

管道1～管道8的总体变化趋势较为相似，且其温度变化范围都在超温警戒线以内。而管道9和管道10均有超过警戒线445℃的取样时间段，且管道10在取样点3200后，多个时段温度超出445℃，出现明显的超温现象。

**2.直方图与核密度图分析**

|  |
| --- |
|  |
| 图5.1.2 10个水冷壁管道温度的频率直方图与核密度图 |
|  |
| 图5.1.3 10个水冷壁管道温度的频数直方图 |

为了研究这些取样点的温度分布情况，我们采用直方图的方法来进行直观的观察。从频率直方图中可以得出，水冷壁管道的温度集中分布在某些温度附近，且有多个集中分布的温度点，这一点从核密度图中也可以清晰地观察到。从10个管道的频数直方图中可以发现，各管道的温度集中于370℃～410℃。

核密度估计完全利用数据本身信息，根据数据自身的特点、性质来拟合分布，避免人为主观带入得先验知识，对样本数据进行最大程度的近似。因此，除了使用直方图来表示水冷壁管道温度的分布情况外，我们还利用核密度估计来进行进一步分析。

以管道1为例进行具体分析，如图5.1.4为管道1的温度直方图与核密度图，从中可以得到，管道1的温度集中在350℃、370℃和410℃附近，其中温度集中在370℃附近的取样点最多。温度在360℃、400℃和420℃附近的取样点较少。

|  |
| --- |
|  |
| 图5.1.4 管道1的直方图与核密度图 |

**3.箱型图分析**

为了更好的描述各水冷壁管道温度数据的离散情况，我们使用箱型图进行进一步分析。箱型图最大的优点就是不受异常值的影响，能够以一种相对稳定的方式描述数据的离散分布情况。

|  |
| --- |
|  |
| 图5.1.5 10个水冷壁管道温度的箱型图 |

由图5.1.5可以分析得出，管道1的平均温度最低，管道10的平均温度最高。管道6的最高温度相较于其他管道最低，管道10的最高温度最大，且已超过警戒值445℃。管道2、3、9的温度值出现了离群点，存在异常值，而其他管道温度无异常值。

**5.2问题二求解**

**5.2.1问题分析**

通过问题一对10个水冷壁管道温度数据的分析，我们从温度变化的平稳性和水冷壁温度不宜过高两个方面来对水冷壁管道的工作状态进行评价，从中选取最差工作曲线和最优工作曲线。

**5.2.2管道温度数据曲线分析**

**1.** **最差工作曲线分析**

首先，从水冷壁温度不宜过高的角度考虑，超温报警线为445℃。从以下时序图中可以发现，在10个管道中，仅有管道9和管道10中存在超过445℃的采样点。因此，管道9和管道10的工作状态相较于其他管道是较差的，且管道10超温的时间段明显多于管道9。其次，从标准差的角度来看，管道10的标准差大于管道9，因此管道10的温度变化相较于管道9更不平稳。

|  |
| --- |
|  |
| 图5.2.1 10个水冷壁管道温度的时序总图 |

|  |
| --- |
|  |
| 图5.2.2 管道10温度的时间序列图 |

由图5.1.7管道10温度的时间序列图也可以发现，在样本序号3000以后，有多段时间段水冷壁温度超过警戒值445℃。

所以，从综合角度来看，在10个水冷壁管道中，管道10的温度曲线为最差工作曲线。

**2.** **最优工作曲线分析**

在最高温度未超过警戒线的剩余8个管道中，我们对这8个管道的温度变化曲线进行进一步分析，从中选择最优工作曲线。

我们使用ADF检验来评判8个管道温度变化曲线的平稳性。ADF检验是指检验序列中是否存在单位根，若存在单位根即为非平稳时间序列。ADF检验的原假设是存在单位根，只要此统计值是小于1%水平下的数字就可以极显著的拒绝原假设，认为数据平稳。

因此，ADF检验的结果主要依据以下两个方面：

（1）Test Statistic的值若小于Critical Value的值(5%)，则满足稳定性需求；

（2）p-value越低（理论上需要低于0.05）证明序列越稳定。

管道1～管道8的温度值ADF检验结果如下表所示：

表5.2 10个水冷壁管道温度值ADF检验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **管道** | **Test Statistic** | **p-value** | **Lags** | **Number** | **stationarity** |
| 管道1 | -2.13492 | 0.230707 | 19 | 4980 | × |
| 管道2 | -3.99027 | 0.00146235 | 18 | 4981 | √ |
| 管道3 | -3.88804 | 0.00212468 | 16 | 4983 | √ |
| 管道4 | -2.33704 | 0.160299 | 17 | 4982 | × |
| 管道5 | -2.64218 | 0.0845889 | 16 | 4983 | × |
| 管道6 | -2.64559 | 0.0839362 | 16 | 4983 | × |
| 管道7 | -2.41457 | 0.137667 | 15 | 4984 | × |
| 管道8 | -2.09904 | 0.2449 | 14 | 4985 | × |

因此，根据8个管道的ADF检验结果可以得出，管道2和管道3的温度序列通过平稳性检验。

|  |
| --- |
|  |
| 图5.2.3管道2和管道3温度的时间序列图 |

而对于管道2和管道3，管道3的标准差明显小于管道2。从图5.1.8的时序图也可以发现，管道3的温度变化更平稳。因此，管道3的温度曲线为最优工作曲线。

**5.3问题三求解**

**5.3.1问题分析**

附件2中影响水冷壁温度的变量涉及153个，包括111个操作变量和42个状态变量，对于涉及到的高纬度数据首先应当对数据降维，以便计算和可视化。常用的数据降维方法有主成分分析、树模型、相关系数等，问题三我们采用线性拟合从中选取重要性变量，并以此为基础分别构建10个水冷壁管道温度变化规律的线性回归模型[1]。

**5.3.2数据降维**

附件2给出了153个影响水冷壁温度的输入变量，在原始的高维空间中，包含冗余信息和噪声信息，会影响准确率。因此，我们采用线性拟合对高维数据进行降维，以便更为有效地综合提取有用数据，摒弃无效信息。

根据线性拟合，得到以下23个主要变量：

|  |
| --- |
|  |
| 图5.3.1 线性拟合得到的23个主要变量 |

|  |
| --- |
|  |
| 图5.3.2 主要变量的可视化 |

**5.3.3模型分析**

为了构建10个水冷壁管道温度变化规律的数学模型，我们尝试使用了常规的机器学习方法：线性回归（Linear Regression）、岭回归（Ridge Regression）、套索回归（Lasso Regression）、随机森林（Random Forest）、XGBoost、AdaBoost、神经网络（MLP）等，各算法得到的结果如下图所示（以管道1为例）：

|  |
| --- |
|  |
| 图5.3.3 各算法的箱型图  通过分析发现，线性回归模型所得出的效果最好，因此我们选择线性回归方法来构建10个水冷壁管道温度变化规律的数学模型。 |

**5.3.4模型构建**

**1.线性回归模型结构**

线性回归试图学得一个线性模型以尽可能准确地预测实值输出标记[2]。假设有数据集，假设所有样本只有一个属性值，线性回归即对数据样本进行训练学习得出假设函数，使得。

**2.多元线性回归模型构建**

多元线性回归模型通常用来描述变量和之间的随机线性关系，即：

|  |  |
| --- | --- |
|  | （5.3-1） |

式（5.3-1）中，是非随机的变量；是随机的因变量；是回归系数；是随机误差项。我们将影响温度值变化的23个主要变量作为自变量,各管道变化的温度值作为因变量，构建多元回归模型。

**3.回归模型的评价指标**

训练模型的过程就是确定参数和的过程,目的是使训练学习得到的函数和样本的输出标签尽可能的相近[3]。为了衡量和之间的相近程度，有以下定义：

|  |  |
| --- | --- |
|  | （5.3-2） |

为代价函数或者均方误差，即要求得和,使得最小。

**5.3.5模型求解**

我们通过构建线性回归模型，来求得10个水冷壁管道温度变化规律的数学模型。我们按照时间顺序划分训练数据与测试数据，训练集与测试集数量比例为7：3。通过求解，各管道温度变化模型的线性回归方差及模型拟合情况如下所示（以管道1为例，其余在附录中展示）：

管道1温度变化的线性回归方程：

|  |
| --- |
|  |
| 图 5.3.4 管道1训练集与测试集拟合情况 |

|  |
| --- |
|  |
| 图 5.3.5 管道1线性回归模型拟合的残差图 |

从管道1～管道10训练集与测试集的时间序列图可以看出，测试样本拟合程度很好。测试集拟合误差损失小于0.02，由此得出线性回归模型效果良好。

**5.4问题四求解**

**5.4.1 问题分析**

问题4中，绘制第10个水冷壁管道温度曲线图，从下图中可以看出，在第3172个样本点后水冷壁出现明显的超温现象。本文基于给出的数据，分析并定位引发超温现象的主要操作变量。

|  |
| --- |
|  |
| 图5.4.1第10根水冷壁管道温度曲线时间序列图 |

从图中容易看出水冷壁的超温现象主要是从取样点3172以后开始的，因此需要对在第3172个样本点后1829个样本点进行分析。附件2提供了111个操作变量,42个状态变量。由于这里的操作变量，是在锅炉燃烧过程中，操作人员能够根据实际生产的需要而进行调节的量；而状态变量是用来描述锅炉燃烧系统运行状态的量，它的取值由相关检测设备采集得到，操作人员无法进行直接调节。因此需要找出引发超温现象主要操作变量。

**5.4.2皮尔逊相关系数分析**

皮尔逊相关系数广泛用于度量两个变量之间的线性相关程度。 本文通过皮尔逊相关系数分析管道10温度与111个操作变量之间的线性关系。由于操作变量过多，下表展示前10个相关系数较大的因素。

表5.3 前10个相关系数大的操作变量

|  |  |
| --- | --- |
| **操作变量(MV)** | **相关系数(ρ)** |
| D磨入口冷一次风电调挡板位置 | -0.70 |
| #3角DD层二次风调节挡板位置反馈 | 0.53 |
| E磨入口冷一次风电调挡板位置 | -0.53 |
| #3角DE层二次风调节挡板位置反馈 | 0.51 |
| #2角FF层二次风调节挡板位置反馈 | 0.50 |
| #3角CC层二次风调节挡板位置反馈 | 0.49 |
| #1角摆角燃烧器调节门反馈 | 0.48 |
| #4角AB层二次风调节挡板位置反馈 | 0.48 |
| #1角B层二次风调节挡板位置反馈 | 0.48 |
| #4角B层二次风调节挡板位置反馈 | 0.47 |

|  |
| --- |
|  |
| 图5.4.2 主要操作变量与管道10温度相关系数热力图 |

由上表可以看出，前5个操作变量与管道10温度存在显著的线性相关关系（）。引发超温现象主要操作变量有：D磨入口冷一次风电调挡板位置、#3角DD层二次风调节挡板位置反馈、E磨入口冷一次风电调挡板位置、#3角DE层二次风调节挡板位置反馈、#2角FF层二次风调节挡板位置反馈。

**5.5问题五求解**

**5.5.1 模型建立**

从第四问分析可以得出，引发超温现象的主要操作变量，通过多元回归拟合得出相应关系式：

|  |  |
| --- | --- |
|  | (5.5-1) |

(5.5-1)为拟合出的多元回归方程。

|  |  |
| --- | --- |
|  | (5.5-2) |

式5.5-2中， 为样本数的集合， 为样本数的索引，表示第j个样本的管道温度值。b为多元回归截距， 为变量的集合，为变量的索引。 为第 个变量的系数。为第 个变量的第j个样本值。

|  |  |
| --- | --- |
|  | (5.5-3) |

(5.5-3)中，为样本管道温度值的平均值，N 为样本数。

|  |  |
| --- | --- |
|  | (5.5-4) |

(5.5-4)表示管道温度的样本方差，其中为样本均值。通过问题二分析可知，时间序列平稳性是衡量最优工作曲线的关键指标，通过最小化样本方差从而优化第10个水冷壁管道超温段温度曲线。

|  |  |
| --- | --- |
|  | (5.5-5) |

(5.5-5)中，为警戒温度，表示每个样本温度不能超过警戒温度。

|  |  |
| --- | --- |
|  | (5.5-6) |

(5.5-6)为最优工作曲线的样本方差。此约束为变量的样本方差小于最优工作曲线的样本方差。

基于以上分析，以（5.5-4）为目标，以（5.5-5）、（5.5-6）为约束条件，建立模型如下：

|  |  |
| --- | --- |
|  | （5.7-8) |

其中，为决策优化变量。

**5.5.2 遗传算法求解目标优化模型**

遗传算法是模拟达尔文遗传选择和自然淘汰的生物进化过程的计算模型[4]，其主要步骤如下：

Step1：本文采用实数编码方法编码，每个染色体为一个实数向量。

Step2：搜索待优化样本点邻域得到初始种群。

Step3：确定适应函数。可用目标函数作为代替种群的适应度计算函数：

|  |  |
| --- | --- |
|  | (5.5-9) |

Step4：依据轮盘赌法筛选优秀个体，个体适应度越高，被选择进行遗传的概率越大；个体适应度越低，被选择的概率就小。这样计算的结果和目的是：最优秀的染色体被复制成多分遗传给下一代，中等的染色体维持原有水平，较差的染色体则不被选择最终淘汰。个体被选中的概率为

|  |  |
| --- | --- |
|  | （5.5-10） |

式中：为个体的适应度值；为种群个体的数目。

Step5：遗传算子的设计：

①交叉操作：由于本文中采取实数编码。所以交叉操作采用实数交叉法，第个染色体和第个染色体在位的交叉操作方法为

|  |  |
| --- | --- |
|  | （5.5-11） |

其中，是区间的随机数。

②变异操作：其主要目的是维持种群多样性。第个体的第个基因进行变异的操作方法为

|  |  |
| --- | --- |
|  | （5.5-12） |

式中：是基因的上界；是基因的下界；为随机数，是当前迭代次数，是最大进化次数，为区间的随机数。

Step6：设定遗传算法相关参数：本文种群大小因为群体规模越大，越容易找到最优解。交叉概率为0.6能够保证种群的充分进化。一般而言，变异发生的可能性较小，变异概率为 0.01更加符合自然规律。考虑到时间复杂度，设定最大遗传代数为1000[5]。

**5.5.3 模型的求解**

利用上述遗传算法的步骤求解目标函数，下图反映了目标函数随着迭代代数的平均寻优情况。

|  |
| --- |
|  |
| 图 5.5.1 遗传算法函数值曲线 |

从图中可以看出，函数值在前300代下降速度很快，在600代以后逐渐趋于平稳。寻找道的全局最优值为100.76，与管道3最优曲线的方差113.98，下降了13.22，曲线更加平稳。从而达到了较好的优化效果。

|  |
| --- |
|  |
| 图5.5.2 优化后的管道10超温度工作曲线 |

从图中容易看出，优化的超温段工作曲线，时间序列表现的更为平稳，同时远低于警戒温度值。表明优化模型达到了效果。

**六、模型优缺点**

**6.1模型的优点**

（1）时间序列平稳性检验可以很好地评价管道温度工作曲线的稳定性，使分析结果更具有合理性。

（2）问题三中综合比较线性回归，岭回归，随机森林等机器学习模型，由于线性回归在10个管道中表现较好，因此采用线性回归模型进行结果，结果更加准确。

（3）问题五优化采用了最小化管道温度时间序列方差，同时考虑调整变量幅度尽可能小，并采用遗传算法求解，模型收敛速度快。

**6.2模型的缺点**

（1）回归模型较为简单，预测效果有限。

（2）问题四进行皮尔逊相关性分析，应进一步做因子分析进行验证。

（3）问题五应考虑双目标建模分析，最小化时间序列方差以及调整操作变量的数量，从而使模型更加适用。

**6.2模型的改进**

（1）应考虑其他非线性模型，例如进行多元非线性拟合出最优方程或者建立LSTM神经网络建立水冷壁管道温度变化规律的数学模型，从而得到的优化模型更为准确。

（2）问题四变量与目标函数的线性相关不是很强，因此可以采用灰色关系分析筛选出主要操作变量。

（3）问题二也可采用层次分析法，通过分析权重，得出加权分析最优曲线，使分析结果更具有合理性。

**参考文献**

[1]仙树祥. 大容量“W”火焰超临界锅炉优化运行技术研究[D].华北电力大学,2015.

[2]陈世杰,唐秋华.优化神经网络用电量预测性能的多元线性回归方法[J].机械设计与制造，2019(06):17-21.

[3]Neeraj Kumar, Ganesh Upadhyay and Pankaj Kumar. Comparative Performance of Multiple Linear Regression and Artificial Neural Network Based Models in Estimation of Evaporation [J]Advances in Research, 2017,11

[4]李雷,赵柏森.基于人工神经网络和遗传算法的封头成形工艺参数多目标优化[J/OL].锻压技术,2021(05):39-45[2021-05-23].

[5] 陈士华,马益平,方健美.遗传算法在智能配电网故障定位应用分析[J].能源与环保，2017，39(12):219-222

**附录**

附录1.问题三的相关图表

问题三中管道2～管道10温度变化的线性回归方程、训练集与测试集拟合情况以及模型拟合的残差图如下所示：

**管道2：**

|  |  |
| --- | --- |
|  |  |
| 管道2训练集与测试集拟合情况 | 管道2线性回归模型拟合的残差图 |

**管道3：**

|  |  |
| --- | --- |
|  |  |
| 管道3训练集与测试集拟合情况 | 管道3线性回归模型拟合的残差图 |

**管道4：**

|  |  |
| --- | --- |
|  |  |
| 管道4训练集与测试集拟合情况 | 管道4线性回归模型拟合的残差图 |

**管道5：**

|  |  |
| --- | --- |
|  |  |
| 管道5训练集与测试集拟合情况 | 管道5线性回归模型拟合的残差图 |

**管道6：**

|  |  |
| --- | --- |
|  |  |
| 管道6训练集与测试集拟合情况 | 管道6线性回归模型拟合的残差图 |

**管道7：**

|  |  |
| --- | --- |
|  |  |
| 管道7训练集与测试集拟合情况 | 管道7线性回归模型拟合的残差图 |

**管道8：**

|  |  |
| --- | --- |
|  |  |
| 管道8训练集与测试集拟合情况 | 管道8线性回归模型拟合的残差图 |

**管道9：**

|  |  |
| --- | --- |
|  |  |
| 管道9训练集与测试集拟合情况 | 管道9线性回归模型拟合的残差图 |

**管道10：**

|  |  |
| --- | --- |
|  |  |
| 管道10训练集与测试集拟合情况 | 管道10线性回归模型拟合的残差图 |

附录2.Python代码

# 均值、标准差、四分位数

desc\_df = rawdata1.describe().T

# 斜率趋势

def trend(seq):

idx = np.array(range(len(seq)))

slope = linregress(idx, seq.squeeze())[0]

return slope

trend\_df = pd.DataFrame(rawdata1.apply(trend)).rename(columns={0:"趋势斜率"})

# k阶中心矩(2阶)

def moment\_k(seq, k=2):

mean = seq.mean()

moment\_k = np.mean((seq-mean)\*\*k)

return moment\_k

moment\_k = pd.DataFrame(rawdata1.apply(moment\_k)).rename(columns={0:"2阶中心距"})

# 波峰、波谷

peaks = dict()

troughs = dict()

for i in range(len(rawdata1.columns)):

peaks[rawdata1.columns[i]] = len(signal.argrelextrema(np.array(rawdata1.iloc[:,i]),np.greater)[0])

troughs[rawdata1.columns[i]] = len(signal.argrelextrema(np.array(rawdata1.iloc[:,i]),np.less)[0])

peaks\_df = pd.DataFrame([troughs]).T

troughs\_df = pd.DataFrame([troughs]).T

peak\_df = peaks\_df.rename(columns={0:"波峰"})

trough\_df = troughs\_df.rename(columns={0:"波谷"})

#时序图

plt.figure(figsize=(30,15))

for col in rawdata1.columns:

plt.plot(rawdata1.loc[:,col], label=col)

plt.axhline(445, c="red",ls="--")

plt.axvline(3171,ls="-",c="gray", linewidth=0.3)

plt.legend(loc="lower right")

plt.xticks(fontsize=14)

plt.yticks(fontsize=14)

plt.xlabel("取样点",fontsize = 16)

plt.ylabel("各个管道温度",fontsize = 16)

plt.savefig("Time\_Series\_plot1.png",dpi=500, bbox\_inches = "tight")

plt.show()

import pandas as pd

import matplotlib.pyplot as plt

plt.style.use('ggplot')

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']

plt.rcParams['axes.unicode\_minus']=False

temp\_data['管道1温度'].plot(kind = 'hist', bins = 70, color = 'steelblue', edgecolor = 'black', density = True, label = '直方图')

temp\_data['管道1温度'].plot(kind = 'kde', color = 'red', label = '核密度图')

plt.xlabel('时刻')

plt.ylabel('温度')

# 添加标题

plt.title('管道1温度分布图')

plt.legend(fontsize = 6)

plt.show()

plt.figure(figsize=(12,8))

for i in range(1,11):

plt.subplot(2,5,i)

plt.style.use('ggplot')

temp\_data.iloc[:,i-1].plot(kind = 'hist', bins = 70, color = 'cyan', density = True, label = 'HIST',fontsize = 3)

temp\_data.iloc[:,i-1].plot(kind = 'kde', color = 'red', label = 'KDE',fontsize = 3)

plt.xlabel('温度',fontsize = 6)

plt.ylabel('频率',fontsize = 6)

plt.title('管道%s温度分布图'%i,fontsize = 6)

plt.legend(fontsize = 6)

plt.subplots\_adjust(wspace =0.5, hspace =0.5)

plt.show()

col= ['管道1', '管道2', '管道3', '管道4', '管道5', '管道6', '管道7', '管道8',

'管道9', '管道10']

temp\_data\_plot = temp\_data

temp\_data\_plot.columns = col

temp\_data\_plot.iplot(kind='box',barmode='stack',yTitle='温度',boxpoints= 'outliers')

temp\_data\_plot.iplot(kind='histogram',barmode='stack', xTitle='温度',yTitle='频数')

# 原始数据平稳性检验

#数据平稳性检测

# def judge\_stationarity(data\_sanya\_one):

# dftest = sm.tsa.stattools.adfuller(data\_sanya\_one)

# dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])

# stationarity = "是"

# for key, value in dftest[4].items():

# dfoutput['Critical Value (%s)'%key] = value

# if dftest[0] > value:

# stationarity = "否"

# print(dfoutput)

# print("是否平稳(是/否): %s" %(stationarity))

# return stationarity

# rawdata1.apply(judge\_stationarity)

# 用Dickey-Fuller test检验序列是否平稳

dftest = rawdata1.apply(sm.tsa.stattools.adfuller).T.drop(5,axis=1).rename(columns={0:"Test Statistic",

1:'P-Value',

2:'Lags',

3:'Number',

4:"Critical Value"})

dfoutput = dftest.drop("Critical Value", axis=1)

dfoutput["stationarity"] = dfoutput["P-Value"].apply(lambda x : x < 0.05)

dfoutput.to\_csv("./rawdata\_stationarity\_test\_result.csv",encoding="utf\_8\_sig")

dfoutput

# 1阶差分后数据平稳性检验

# 用Dickey-Fuller test检验序列是否平稳

dftest2 = diff\_data1.apply(sm.tsa.stattools.adfuller).T.drop(5,axis=1).rename(columns={0:"Test Statistic",

1:'P-Value',

2:'Lags',

3:'Number',

4:"Critical Value"})

dfoutput2 = dftest2.drop("Critical Value", axis=1)

dfoutput2["stationarity"] = dfoutput2["P-Value"].apply(lambda x : x < 0.05)

dfoutput2.to\_csv("./diffdata\_stationarity\_test\_result.csv",encoding="utf\_8\_sig")

dfoutput2

fig=plt.figure(figsize=(30,25))

colors=list(mcolors.TABLEAU\_COLORS.keys()) #颜色

cols = diff\_data1.columns

for i in range(len(cols)):

ax=fig.add\_subplot(10,1,i+1)

ax.plot(diff\_data1.iloc[:,i], label=cols[i],color=mcolors.TABLEAU\_COLORS[colors[i]])

ax.legend(loc="lower right")

plt.xticks(fontsize=14)

plt.yticks(fontsize=14)

plt.xlabel("取样点",fontsize = 16)

plt.savefig("Time\_Diff\_Series.png",dpi=500, bbox\_inches = "tight")

plt.show()

# 合并数据集,划分训练集

data\_all = rawdata1.join(rawdata2)

X\_data = data\_all.iloc[:,10:].fillna(method="ffill")

Y\_data = data\_all.iloc[:,:10]

# 线性拟合系数确定重要性

model = LinearRegression()

linear\_coefs = dict()

for i in range(len(Y\_data.columns)):

model.fit(X\_data, Y\_data.iloc[:,i])

linear\_coefs[Y\_data.columns[i]] = model.coef\_

importance\_df = pd.DataFrame(linear\_coefs, index=X\_data.columns)

# importance\_df.to\_csv("./importance\_linear\_coef\_allfeature.csv",encoding="utf\_8\_sig")

# 变量选择

importance\_df["mean"] = importance\_df.mean(axis=1)

select\_feature = importance\_df.loc[np.abs(importance\_df["mean"]) > 0.5,:].drop("mean",axis=1)

X\_newdata = X\_data[select\_feature.index.to\_list()]

X\_newdata

# 变量重要性的可视化

fig=plt.figure(figsize=(30,25))

colors=list(mcolors.TABLEAU\_COLORS.keys()) #颜色

cols = importance\_df.drop("mean",axis=1).columns

for i in range(len(cols)):

ax=fig.add\_subplot(10,1,i+1)

ax.bar(range(len(importance\_df)),importance\_df.iloc[:,i],label=cols[i],color=mcolors.TABLEAU\_COLORS[colors[i]])

ax.legend(loc="lower right")

plt.xticks(fontsize=14)

plt.yticks(fontsize=14)

plt.savefig("importance\_bar\_allfeature.png",dpi=500, bbox\_inches = "tight")

plt.show()

# 建立模型

# 划分数据集(可能需要按照时间顺序划分Train、Test)

# X\_train,X\_test,y\_train,y\_test = train\_test\_split(X\_newdata,Y\_data.iloc[:,1],random\_state=200,test\_size=0.7)

#划分训练集(按照时间划分)

for i in range(len(Y\_data.columns)): # 循环10个管道

index = X\_newdata.index[int(len(X\_newdata)\*0.7)]

X\_train = X\_newdata[:int(len(X\_newdata)\*0.7)+1]

X\_test = X\_newdata[int(len(X\_newdata)\*0.7):]

y\_train = Y\_data.iloc[:,i][:int(len(Y\_data.iloc[:,i])\*0.7)+1]

y\_test = Y\_data.iloc[:,i][int(len(Y\_data.iloc[:,i])\*0.7):]

# 各个模型的交叉验证

models = {

"Linear Regression":LinearRegression(),

"Ridge Regression":Ridge(),

"Lasso Regression":Lasso(),

"Random Forest":RandomForestRegressor(),

"XGBoost":XGBRegressor(),

"AdaBoost":AdaBoostRegressor(),

"MLP":MLPRegressor() }

scores\_dict = {}

for name,model in models.items():

cv = ShuffleSplit(n\_splits=10, test\_size=.3, random\_state=0)

scores\_CV = cross\_val\_score(model, X\_newdata, Y\_data.iloc[:,i], cv=cv)

print(name,":",scores\_CV)

scores\_dict[name] = scores\_CV # 保存交叉验证评分结果

df\_scoresCV = pd.DataFrame(scores\_dict)

df\_scoresCV.to\_csv("./df\_scoresCV\_{}.csv".format(Y\_data.columns[i]),index=False,header=True)

# 可视化绘制箱线图

score\_plotdata = pd.DataFrame(scores\_dict).melt()

plt.figure(figsize=(15,10))

sns.set(style="whitegrid", color\_codes=True)

sns.boxplot(x="variable", y="value",data=score\_plotdata)

sns.swarmplot(x="variable", y="value", data=score\_plotdata, color=".25")

plt.xticks(rotation=90)

plt.xlabel("Machine Learning Models")

plt.ylabel("Score")

plt.savefig("./score\_plot\_cv\_machine\_learning\_{}.png".format(Y\_data.columns[i]),dpi=500, bbox\_inches = "tight")

plt.show()

"""线性回归"""

lr = LinearRegression(fit\_intercept=False,normalize=True,n\_jobs=10).fit(X\_train,y\_train)

lr\_coef = np.round(lr.coef\_,3)

print("线性回归系数：\n",lr\_coef)

print('Training set score:{:.10f}\n'.format(lr.score(X\_train,y\_train)))

print('Test set score:{:.10f}\n'.format(lr.score(X\_test,y\_test)))

print("Training set MSE:{:.2f}\n".format(mean\_squared\_error(y\_train,lr.predict(X\_train))))

print("Test set MSE:{:.2f}\n".format(mean\_squared\_error(y\_test,lr.predict(X\_test))))

lr\_predict = lr.predict(X\_newdata)

predict\_data = pd.concat([pd.DataFrame(X\_data),pd.DataFrame(Y\_data,columns=["Y"]),pd.DataFrame(lr\_predict,columns=["predict"])],axis=1)

predict\_data.to\_csv("./predictdata\_{}.csv".format(Y\_data.columns[i]),index=False,header=True)

# 残差图

plt.plot((lr.predict(X\_newdata)-Y\_data.iloc[:,1])\*\*2)

plt.savefig("./线性回归模型拟合的残差图\_{}.png".format(Y\_data.columns[i]), dpi = 500, format = "png", bbox\_inches = "tight")

plt.show()

#可视化绘制线性模型的预测效果时序图

predict\_train = lr.predict(X\_train)

predict\_test = lr.predict(X\_test)

fig = plt.figure(figsize=(15,8))

plt.plot(y\_train,"black",label="Train Data")

plt.plot(y\_test,"black",label="Test Data")

plt.plot(predict\_train,"red",label = "Predict Data")

plt.plot(range(index-1,5000),predict\_test,"blue",label = "Predict Test")

plt.xticks(fontsize = 14)

plt.yticks(fontsize = 14)

plt.xlabel("DateTime", fontsize = 16)

plt.ylabel("Temperature", fontsize = 16)

plt.axvline(x=index,ls="-",c="black")#添加垂直直线

plt.legend(loc="best")

plt.savefig("./PredictResultAll{}.png".format(Y\_data.columns[i]), dpi = 500, format = "png", bbox\_inches = "tight")

print('Train RMSE: %.4f'% np.sqrt(np.sum((predict\_train-y\_train)\*\*2)/y\_train.size))

print('Test RMSE: %.4f'% np.sqrt(np.sum((predict\_test-y\_test)\*\*2)/y\_test.size))

plt.show()

all\_data = pd.read\_excel(r'D:\Download\数学建模\第一届长三角高校数学建模竞赛赛题\竞赛题目\竞赛B题\all\_data.xlsx')

col= ['pipe10','MV1', 'MV2', 'MV3','MV4', 'MV5','MV6', 'MV7','MV8','MV9','MV10']

all\_data2.columns = col

#绘制图形

#显示中文

plt.rcParams['font.sans-serif']=['SimHei']

plt.rcParams['axes.unicode\_minus'] =False

ax = plt.subplots(figsize=(12,8 ))#调整画布大小

cmap = sns.cubehelix\_palette(start = 1.5, rot = 3, gamma=0.8, as\_cmap = True)

# ax = sns.heatmap(mv\_corr2, linewidths = 0.05, vmax=900, vmin=0,annot=True,cmap=cmap)

ax = sns.heatmap(all\_data3, linewidths =.5, annot=True )

# 设置刻度字体大小

plt.title('Pipe\_Temp MV correlation coefficient')

plt.xticks(fontsize=10,rotation = 30)

plt.yticks(fontsize=10,rotation = 30)

# 建立模型

from sklearn.model\_selection import train\_test\_split #这里是引用了交叉验证

from sklearn.linear\_model import LinearRegression #线性回归

def build\_lr():

X = X\_data

y = Y\_data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=532)#选择20%为测试集

print('训练集测试及参数:')

print('X\_train.shape={}\n y\_train.shape ={}\n X\_test.shape={}\n, y\_test.shape={}'.format(X\_train.shape,

y\_train.shape,

X\_test.shape,

y\_test.shape))

linreg = LinearRegression()

#训练

model = linreg.fit(X\_train, y\_train)

print('模型参数:')

print(model)

# 训练后模型截距

print('模型截距:')

print (linreg.intercept\_)

# 训练后模型权重（特征个数无变化）

print('参数权重:')

print (linreg.coef\_)

y\_pred = linreg.predict(X\_test)

sum\_mean = 0

for i in range(len(y\_pred)):

sum\_mean += (y\_pred[i] - y\_test.values[i]) \*\* 2

sum\_erro = np.sqrt(sum\_mean /len(y\_pred)) # 测试级的数量

# calculate RMSE

print ("RMSE by hand:", sum\_erro)

# 做ROC曲线

plt.figure()

plt.plot(range(len(y\_pred)), y\_pred, 'b', label="predict")

plt.plot(range(len(y\_pred)), y\_test, 'r', label="test")

plt.legend(loc="upper right") # 显示图中的标签

plt.xlabel("the number of sales")

plt.ylabel('value of sales')

plt.show()