工业 4.0 大数据竞赛－制造业质量控制

技术方案总结

一、回顾

我是 wepon，参加这个比赛大概是在暑期八月中旬时，因为当时写完了一篇论文，有两三周的空闲时间，恰好又看到天池的众智平台上线了这个比赛，所以就报名参加了。这个比赛不同于我以往参加过的比赛，不能组队，单人参赛，所以很多工作都自己独立完成，不过好在赛题本身并不复杂，是典型的回归问题。最终有幸取得了第一名的成绩，这篇文章总结一下我的解题思路，代码也会整理发布在我的 github 上（https://github.com/wepe），欢迎交流。

二、赛题

在工业 4.0 的战略背景下，大数据是提升制造业智能化水平的关键技术之一。本次赛题针对制造业的产品质量控制问题，提供了生产过程中积累下来的海量数据，包括：选材环节记录下的原材料供应商、等级等属性数据，加工环节记录下的核心监控指标随着时间变化的值，质量检测环节记录下的关键质量指标值。根据这些数据，完成以下两个任务：

任务一：建立关键质量指标的预测模型，要求预测生产流程中指定进度的良品率

任务二：对生产过程中的工艺可调参数推荐三组最优的预设值，以取得较好的关键质量指标

（详细的赛题说明请看：天池众智-工业 4.0 大数据竞赛）

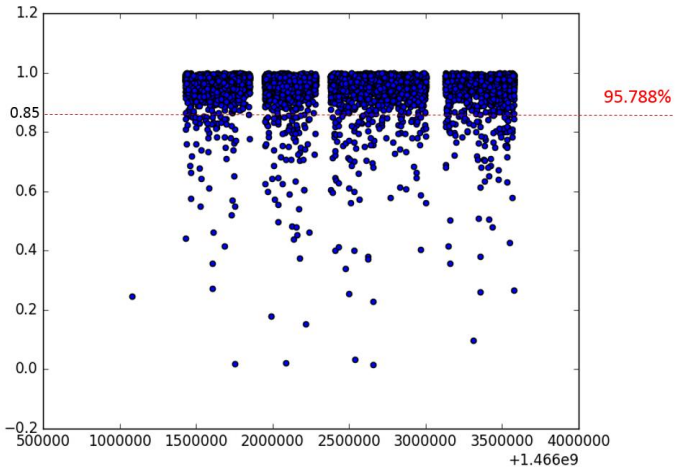
三、任务一，关键质量指标预测

3.1 解决方案概述

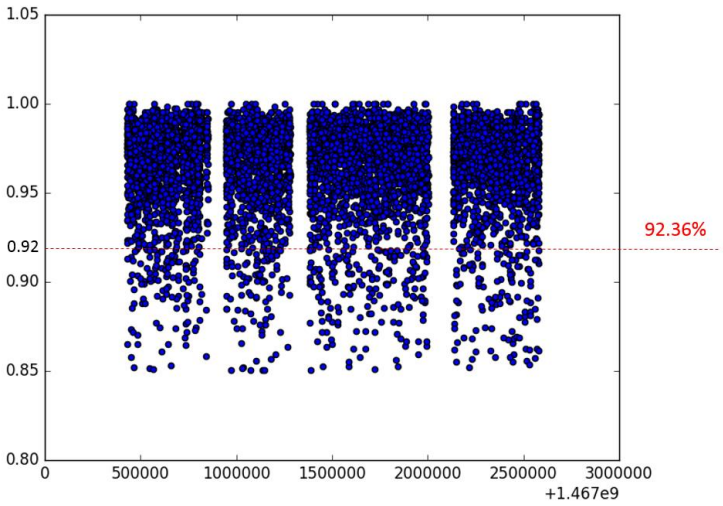
关键质量指标（良品率）是界于[0,1]之间的连续值，可建立回归模型对其进行预测。基于所提供的工艺可调参数表（draft_data）、工艺不可调参数表 （param_data）、时序状态监控指标表（param_data_timevarying），提取了丰富而有判别性的特征，建立 XGBoost、Dart、Random Forest 三种模型对关键质量指标进行预测，并进一步地进行模型融合。此外，由于数据分布不平衡，训练得到的模型的预测值会集中在[0.92,0.98]之间，所以在回归的基础上，又设计了分类模型，检测出良品率低于 0.92、高于 0.98 的产品批次。总体来说，这是一个回归和分类相结合的方案，回归为主，分类为辅。

3.2 数据分析

以时间(unix 时间戳)为横坐标，良品率为纵坐标，每个产品批次为一个散点，画出如下的散点图，并统计每个区间产品批次的百分比：



从图中可以得出，良品率在 0.85 以上的产品批次占了 95.788%，只有极少数产品批次良品率分布在[0,0.85]之间，这种极度不平衡的数据分布会导致模型的预测值集中在 0.85 以上的部分（当然，造成这种模型偏差的原因不仅仅是数据分布不平衡，数据量小、特征判别性不足等也是可能的客观因素）。因此，用给定的数据建模，以 RMSE 为评测指标，测试集中真实良品率低于 0.85 的那些产品批次占据了大部分的误差，这也是主办方在切换数据后剔除良品率低于 0.85 的产品批次的原因。那么对于良品率高于 0.85 的产品批次，同样画出散点图：



从图中看出，92.36%的产品批次良品率高于 92%，仍然存在着数据分布不平衡的现象。解决数据分布不平衡是机器学习中的一个研究课题，常用的方法有上采样/下采样、代价敏感学习。由于本赛题数据量本身比较小，不考虑下采样。上采样我在比赛初期进行了尝试，线上线下效果都不佳。由于是回归问题，代价敏感学习比较难操作，有待进一步探索。

最终我采用的方法是不做预处理，但是做后处理。具体来说，先用回归模型进行预测，由于预测值集中分布在[0.92,0.98]之间，所以又建立分类模型，找出低于 0.92、高于 0.98 的产品批次，以此对预测结果做后处理，将被分类模型判定为低于 0.92 的产品的预测值限定为 0.92，将被分类模型判定为高于 0.98 的产品的预测值限定为 0.98。这个方案最终取得了 RMSE 0.02558 的线上成绩，排名第一。

3.3 特征工程

3.3.1 类别型变量的处理

draft_data 和 param_data 两张表中都含有类别型变量，在输入模型前需要做编码处理，采

用常用的 one-hot encode。

3.3.2 数值型变量的处理

param_data 数据表中的 param5 和 param9 虽是数值型变量，但是其取值只含有限的几种，所以也当成类别型进行了 one-hot encode，但在训练时保留了原始的数值。其它数值变量不做处理。对于缺失值，用中值填充。

3.3.3 时序监控指标特征提取

param_data_timevarying 提供了加工环节过程中监控的温度、湿度、流量、转速等指标，从这部分数据，分时间阶段提取了各个参数的统计量，包括均值、中值、众值、最大/小值、方差。具体来说，根据 add_time，将每个 product_no 的加工过程均匀划分成 10 个阶段，统计每个时间段内的各个参数的各种统计量。另外，也提取了整个加工过程、加工进度 50%时的参数统计量。这些统计量特征虽然简单，却具有一定的判别性，比如某个时间段内，温度方差太大，对于产品的质量可能会产生比较大的影响。

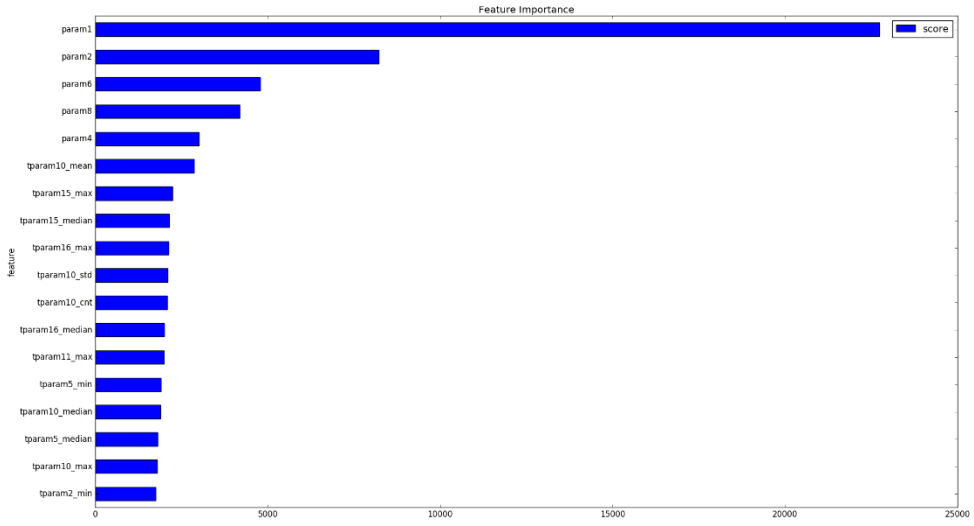
3.3.4 其它特征

除了上面提到的特征，还构建了两个特征：

* 近似的加工总时长，用最大 add_time 减去最小 add_time
* param1-param2，因为观察到这两个参数的值非常接近，它们的差或许有意义

3.3.5 特征选择

从 param_data_timevarying 数据表中提取的特征有几百维，这些特征很多都是冗余的，容易引起过拟合，有必要进行特征选择。特征选择方法主要有嵌入式(Embedded)、过滤式(Filter)和封装式(Wrapper)式三种。本文采用了嵌入式的方法，因为模型部分采用了树模型 XGBoost 和随机森林，这两种模型在训练完成后可以直接得到特征的重要性，可以方便地对特征进行排序，剔除重要性低的特征。下图是 XGBoost 输出的特征重要性排序（只显示了 top20）：

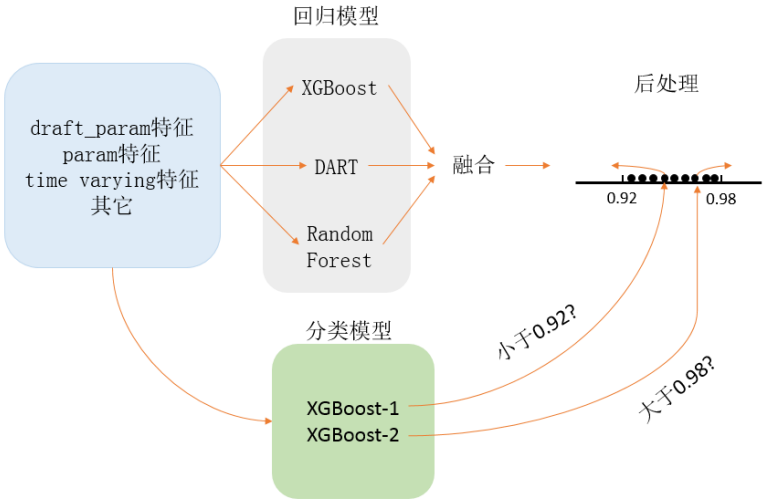


最终只保留了不到 100 维特征，从线上线下得结果来看，特征选择起到了很好的防止过拟合的作用。

3.4 模型设计与模型融合

3.4.1 总体框架

下图显示了本方案的总体框架，基于所构建的特征，先建立回归模型进行预测和融合，然后再用分类模型找出置信度比较高的那些低于 0.92、高于 0.98 的产品批次，对回归模型的预测值进行后处理。



3.4.2 三种回归模型：XGBoost、Dart、Random Forest

模型方面，采用了数据挖掘竞赛领域广泛使用的 XGBoost、最近在 JMLR 上刚发表的 DART（XGBoost 的改进版，引入了深度学习中常用的 Dropout）、以及经典的随机森林。

值得一提的是，这三种树模型有一个共同并且重要的参数：min_child_weight，对于回归问题来说，这个参数对应的就是每个叶子节点上最小的样本个数。这个参数值设置得越小，越容易过拟合。在本赛题中，min_child_weight 设置为 5，相比设置为 1，rmse 会有不小的提升。

3.4.3 回归模型融合

XGBoost、Dart、Random Forest 这三种模型都是基于树的模型。XGBoost 和 DART 是 boosting 算法，侧重于降低模型偏差。随机森林是 bagging 算法，偏重于降低模型方差。将这几种模型进行融合，可以进一步地提高模型的性能。常用的且效果比较好的融合方法是 stacking 或 blending，但是由于本赛题数据量相对比较小，做多层（multi level）的 stacking/blending learning 容易过拟合，所以最终只采用了简单的加权平均的方法：0.4*XGB + 0.4*DART + 0.3*RF，权重是根据模型的线下表现进行调节的。

3.4.4 分类模型

回归模型的预测值集中分布在[0.92,0.98]之间，所以又建立了两个二分类模型，其中一个预测样本的 key_index 是否低于 0.92，另一个预测样本的 key_index 是否高于 0.98。这两个分类模型所采用的特征跟回归模型所用的特征是一样的，模型也同样采用了 XGBoost，但是模型以“binary:logistic”作为目标函数，以 AUC 作为评估指标。

模型训练完成后，对测试集进行预测，得到每个样本 key_index 低于 0.92 或者高于 0.98 的概率，将那些置信度高（概率大）的样本的预测值，限定为 0.92 或者 0.98。使用这个后处理方案，线上线下都有不小的提升。

四、任务二，工艺可调整参数推荐

4.1 任务描述

根据给定的工艺上可调整的参数列表（draft_data）、工艺上不可调整的参数列表（param_data）和时序状态监控指标表（param_data_timevarying），对生产过程中的工艺可调整参数（draft_data）推荐三组最优的预设值，以取得较好的关键质量指标。

4.2 解决方案概述

param_data_timevarying 数据表是在 param_data 和 draft_data 已经确定的条件下，在加工环节被记录的，有明显的先后顺序，所以 param_data_timevarying 数据对于最优 draft_param 的推荐可能没有本质的影响。基于这个假设，本文在对 draft_param 进行推荐时，只考虑 param_data 数据的影响。下文给出两个解决方案，一个是整体的工艺可调整参数推荐，另一个是针对特定的工艺不可调整参数，对工艺可调参数进行推荐。

4.3 整体的工艺可调参数推荐

draft_data_train 数据表含有 draft_param1~draft_param11 的参数组合，也给出了相应的关键质量指标值 key_index，基于此，可挖掘出使得 key_index 最大的最佳参数组合。

（draft_param 简写为 dp）

参数	dp1	dp2	dp3	dp4	dp5	dp6	dp7	dp9	dp10	dp11
类型	类别	类别	类别	数值	数值	数值	数值	数值	类别	类别
取值	0~5	0,1	0,1	342,343	0.06,0.065,0.075, 0.08,0.085,0.27, 0.28,0.285,0.295	0.4,1	1,1.04, 1.05,1.3	0,0.34, 0.35,0.04	0~2	0~4

上面的表是由 draft_data_train 数据得到的，可以看到，类别型变量的取值个数很少，而数值型（double）变量的取值个数也非常有限。对 draft_param1~draft_param11 的参数组合进行分组，可以得到 train 数据中所有出现的参数组合，并计算每种组合的 key_index 的均值，中值，最大值，最小值，以及每种组合出现的次数。得到 draft_param_statistics 统计表，该表中总共只有 91 种 draft_param 的组合，按照 key_index 均值从大到小排序，下图显示了 Top 40 种组合的统计信息：

	draft_param1	draft_par	draft_par	draft_par	draft_par	draft_par	draft_par	draft_par	draft_par	draft_param11	keyindex_mean	keyindex_median	keyindex_max	keyindex_min	count
2	5	0	1	343	0.075	0.4	1.05	0.34	1	3	0.98918	0.98918	0.99595	0.98241	2
3	3	0	0	343	0.065	1	1	0.34	0	1	0.9862825	0.989445	0.99017	0.97607	4
4	5	0	1	343	0.075	1	1.3	0.35	0	4	0.983195	0.983195	0.99562	0.97077	2
5	5	0	1	343	0.065	1	1.3	0.34	1	3	0.98298	0.98291	0.98901	0.97709	4
6	3	0	0	343	0.065	1	1	0.34	2	1	0.98225	0.98338	0.9878	0.97557	3
7	1	0	0	342	0.065	1	1	0.34	1	0	0.9819625	0.98032	0.99534	0.97187	4
8	3	0	0	343	0.075	1	1.05	0.35	2	1	0.980566667	0.98275	0.99202	0.96693	3
9	0	0	0	343	0.065	1	1.05	0.34	1	4	0.9798325	0.981695	0.98679	0.96915	4
10	5	0	1	343	0.065	1	1.05	0.34	2	3	0.976726667	0.982115	0.99526	0.95322	6
11	3	0	0	343	0.065	1	1.04	0.34	1	1	0.976670714	0.97789	0.99216	0.95459	14
12	3	0	0	343	0.065	1	1.04	0.34	0	1	0.9749275	0.97635	0.98397	0.96304	4
13	3	0	0	343	0.065	1	1.05	0.34	0	4	0.971166	0.98283	0.99765	0.73938	65
14	3	0	0	343	0.06	1	1.05	0.34	1	0	0.968736667	0.97171	0.97266	0.96184	3
15	5	0	1	343	0.285	0.4	1.05	0	1	3	0.968496667	0.98279	0.98422	0.93848	3
16	4	0	1	343	0.28	0.4	1.05	0	0	3	0.968286667	0.965	0.99056	0.94952	12
17	5	0	1	343	0.285	0.4	1.05	0	0	3	0.96805	0.975465	0.97832	0.94295	4
18	3	0	0	343	0.065	1	1.04	0.34	1	0	0.967869375	0.97243	0.98648	0.91962	16
19	5	0	1	343	0.085	0.4	1.05	0.35	1	3	0.967505	0.967855	0.98837	0.94594	4
20	4	1	1	343	0.075	0.4	1.05	0.35	1	3	0.965615385	0.969	0.99097	0.92345	13
21	5	0	1	343	0.075	0.4	1.05	0.35	0	3	0.9655985	0.9746	0.99095	0.896	20
22	3	0	0	343	0.065	1	1	0.34	1	1	0.964548649	0.97101	0.99543	0.85794	37
23	5	0	1	343	0.075	1	1.3	0.34	0	3	0.964335	0.964335	0.96918	0.95949	2
24	5	0	1	343	0.28	0.4	1.05	0	1	3	0.963945766	0.97169	1	0.57816	751
25	4	1	1	343	0.075	0.4	1.05	0.35	2	3	0.963435	0.965445	0.98253	0.94032	4
26	3	0	0	342	0.065	1	1.05	0.34	1	0	0.961358854	0.9756	1	0.63087	157
27	4	1	1	343	0.28	0.4	1.05	0	1	3	0.960654412	0.97264	0.9948	0.84848	68
28	3	0	0	343	0.075	1	1.3	0.35	0	1	0.96006	0.95622	0.98095	0.94685	4
29	3	0	0	343	0.075	1	1.3	0.35	1	0	0.959861935	0.97494	0.99209	0.7756	31
30	5	0	1	343	0.28	0.4	1.05	0	0	3	0.959758553	0.971455	1	0.63411	380
31	3	0	0	343	0.065	1	1.05	0.34	1	4	0.959428269	0.968335	0.99371	0.84984	104
32	5	0	1	343	0.075	0.4	1.05	0.35	1	3	0.958972157	0.96935	1	0.8524	51
33	3	0	0	343	0.065	1	1.05	0.35	1	1	0.9588675	0.96873	0.976	0.92201	4
34	4	1	1	343	0.075	1	1.3	0.35	2	3	0.958202895	0.96734	0.993	0.84897	38
35	3	0	0	343	0.065	1	1.05	0.35	1	0	0.958121429	0.97333	0.99216	0.86666	7
36	4	1	1	343	0.065	1	1.05	0.34	1	3	0.957957143	0.9625	0.98491	0.91001	7
37	1	0	0	342	0.065	1	1.05	0.34	1	0	0.957943966	0.97504	0.99598	0.77073	58
38	2	0	0	343	0.065	1	1.05	0.34	1	4	0.95775386	0.97176	0.9954	0.68934	57
39	3	0	0	343	0.065	1	1.04	0.34	2	1	0.9571625	0.96473	0.99198	0.87469	8
40	5	0	1	343	0.075	1	1.3	0.35	1	3	0.956916534	0.968935	1	0.11414	2458
41	5	0	1	343	0.075	1	1.3	0.35	2	3	0.956434017	0.96733	0.99555	0.63488	229

对 Top20/30/40 参数组合中的每个 draft_param 进行 count 累加，找出每个 draft_param 的众数值作为推荐。对于 double 型的参数，也可以取中值或者均值作为推荐，但是从给定的数据来看，这些参数的取值个数也只是有限的几个，并且参数具体的含义无从得知，不好做给定值以外的其它值的推荐。另外，在推荐时也考虑了一些固定的参数组合，比如 draft_param1,draft_param2,draft_param3 这三个类别型变量，在 trainset 中出现得比较多的组合是"3,0,0"和“5,0,1”，而“3,0,1”，“5,0,0”之类的组合并未出现过。

最终，只根据 draft_data_train 数据表，以 key_index 均值为评估标准，推荐了以下三种参数组合：

（draft_param 简写为 dp）

参数	dp1	dp2	dp3	dp4	dp5	dp6	dp7	dp9	dp10	dp11
推荐值 1	3	0	0	343	0.065	1.0	1.045	0.34	1	1
推荐值 2	5	0	1	343	0.07	1.0	1.3	0.345	1	3
推荐值 3	5	0	1	343	0.075	0.4	1.05	0.34	1	3

4.4 针对特定的工艺不可调整参数，对工艺可调参数进行推荐

上面的方法给出了整体最佳的工艺可调参数组合，没有考虑不同产品批次的工艺不可调参数的差异性，这部分将更加细致地针对特定的工艺不可调参数，对工艺可调参数进行推荐。

所采取的方法思路很简单，首先从训练数据里筛选出 key_index 大于一定阈值（阈值可根据需求调节，本方案中设置为 0.97）的产品批次，以这些产品批次的 draft_param 作为候选值。对于新的 product_no（即测试样本），以工艺不可调参数（param_data 表）作为特征，从候选样本里找出与之最相似（或 Top k 个最相似）的样本，取其 draft_param 作为推荐。问题转化为一个相似性度量（距离度量）的问题

我们熟知的 KNN、Kmeans 算法中采用了欧式距离，只适用于数值型变量的情况，而本赛题中同时含有类别型变量和数值型变量，欧式距离无法直接处理这种具有混合属性的情况。为了度量混合属性的相似性，一般采用结合不同距离度量方式的方法，比如 MinkovDM，结合了闵可夫斯基距离和 VDM（Value Difference Metric），其中闵可夫斯基距离用于处理数值型变量，VDM 用于处理类别型变量：

$$\mathrm{MinkovDM}_p(x_i,x_j)=\Big(\sum_{u\in numeric}|x_{iu}-x_{ju}|^p+\sum_{u\in category}VDM_p(x_{iu},x_{ju})\Big)^{\frac{1}{p}}$$

由于本赛题中类别型变量只有 param3、param4、param7、param8 共四个，且取值只有有限的几种，所以测试样本总是能从候选样本中找到（param3、param4、param7、param8）取值完全一样的样本（而且不少），即类别型变量部分产生的距离为 0，起作用的只有数值型变量产生的距离。

重点考察数值型变量的距离度量部分，取 p = 2，闵可夫斯基距离即欧氏距离，由于欧氏距离对数值大小敏感，所以先对每个数值型 param 进行了归一化。另外，在任务一中已经得到了特征的重要性排序（见 3.3.5 部分图），工艺不可调参数中的数值型参数的重要性为：param1>param2>param6>param5>param9，所以可以进一步地赋予不同的权重，即加权欧氏距离，权重如下表所示：

参数	param1	param2	param5	param6	param9
权重	0.24	0.22	0.18	0.2	0.16

根据这个方法，给 testset 中的每个 product_no 都推荐了一组 draft_param，结果文件为 **recommend_every_product.csv**