Alibaba Cloud 阿里巴巴基础设施 Alibaba Infrastructure Service

3RD ALIBABA CLOUD AIOPS COMPETITION

第 三 届 阿 里 云 磐 久 智 维 算 法 大 赛

“基于大规模日志的故障诊断”总决赛

基于树模型的AIOps故障根因分析与定位方法

悦智AI实验室

CONTENTS

赛题分析

1、赛题理解

基于故障工单+系统日志数据，构建多分类模型，快速、高效的定位出故障类型。

2、评价指标

采用加权的F1-score，类别0和1赋予的权重更大。

初赛评价指标

- 综合四类，即「两种CPU故障」(0-1)、「内存故障」(2)和「其他故障」(3)，有
  - 对应该四类的权重向量

$$w = \{\frac{3}{7}, \frac{2}{7}, \frac{1}{7}, \frac{1}{7}\}$$

  - Macro F1-score

$$\text{Macro F1-score} = \sum_{i \in \{0,1,2,3\}} w_i \times \text{F1-score}_i$$

复赛评价指标

四个类别对应的权重向量为：

$$w = \{\frac{5}{11}, \frac{4}{11}, \frac{1}{11}, \frac{1}{11}\}$$

其余与初赛相同。

数据分析

1、数据样本

故障工单数据

sn	fault_time	label
SERVER_10001	2020-05-01 10:04:00	1
SERVER_10003	2020-03-28 09:48:00	2
SERVER_10008	2020-02-25 16:12:00	1
SERVER_10008	2020-03-11 18:04:00	2
SERVER_10009	2020-05-08 16:37:00	3
SERVER_10012	2020-07-13 03:32:00	3
SERVER_10017	2020-06-11 15:52:00	3
SERVER_10017	2020-06-11 15:52:00	3
SERVER_10018	2020-05-31 03:33:00	3
SERVER_10019	2020-01-29 22:38:00	3

故障日志

sn	time	msg	server_model
000d33b21436	2020-09-02 11:38:40	System Boot Initiated BIOS_Boot_Up | Initiated by warm reset | Asserted	SM40
000d33b21436	2020-09-02 15:46:23	System Boot Initiated BIOS_Boot_Up | Initiated by power up | Asserted	SM40
004d5a7954e7	2020-08-24 08:07:29	Memory CPU1D0_DIMM_Stat | Uncorrectable ECC | Asserted	SM35
004d5a7954e7	2020-08-24 08:07:30	Processor CPU0_Status | Configuration Error | Asserted	SM35
004d5a7954e7	2020-08-24 08:07:54	Processor CPU0_Status | Configuration Error | Asserted	SM35
004d5a7954e7	2020-08-24 08:07:54	Memory CPU1D0_DIMM_Stat | Uncorrectable ECC | Asserted	SM35
004d5a7954e7	2020-08-24 08:13:11	Memory CPU1D0_DIMM_Stat | Correctable ECC | Asserted	SM35
007bdf23b62f	2020-06-16 17:04:19	Memory #0xe2 | Correctable ECC | Asserted	SM93
007bdf23b62f	2020-06-16 17:10:54	Memory #0xe2 | Correctable ECC | Asserted	SM93
007bdf23b62f	2020-06-16 17:11:17	Memory #0xe2 | Correctable ECC | Asserted	SM93

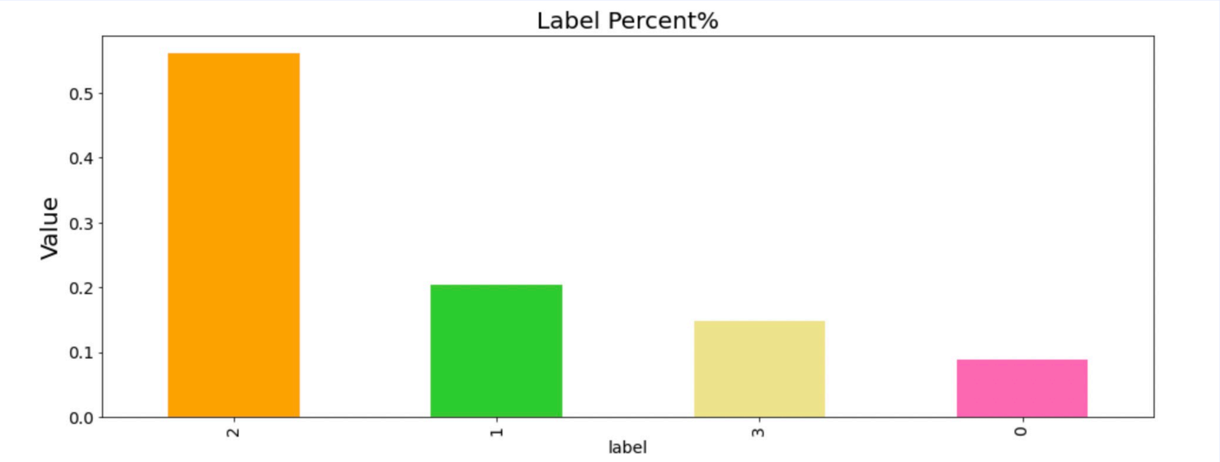
msg_0	msg_1	msg_2
[' System Boot Initiated BIOS_Boot_Up']	['Initiated by warm reset']	['Asserted']
[' System Boot Initiated BIOS_Boot_Up']	['Initiated by power up']	['Asserted']
[' Memory CPU1D0_DIMM_Stat']	['Uncorrectable ECC']	['Asserted']
[' Processor CPU0_Status']	['Configuration Error']	['Asserted']
[' Processor CPU0_Status']	['Configuration Error']	['Asserted']
[' Memory CPU1D0_DIMM_Stat']	['Uncorrectable ECC']	['Asserted']
[' Memory CPU1D0_DIMM_Stat']	['Correctable ECC']	['Asserted']
[' Memory CPU1D0_DIMM_Stat']	['Uncorrectable ECC']	['Asserted']
[' Processor CPU0_Status']	['Configuration Error']	['Asserted']
[' Processor CPU0_Status']	['Configuration Error']	['Asserted']

- 本赛题提供了故障工单与故障日志数据，具体数据如上图所示。通过初步分析日志结构，可以将其分解为右图数据。
- 初步认识：从服务器出现故障的业务场景出发，在故障发生的前后5/10/15/30分钟或更久，所产生的日志信息，都会与此故障有关。

数据分析

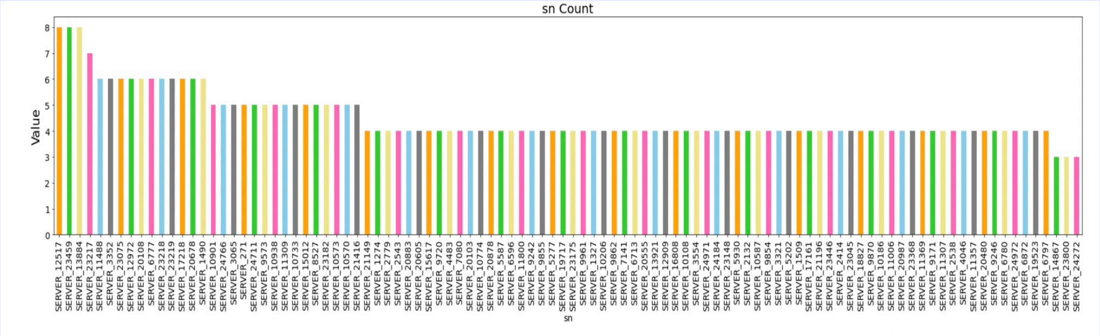
2、标签分布

- 4类标签：0类和1类表示CPU相关故障，2类表示内存相关故障，3类表示其他类型故障
- label_2占比最多，达到56%
- label_0占比最少，达到9%



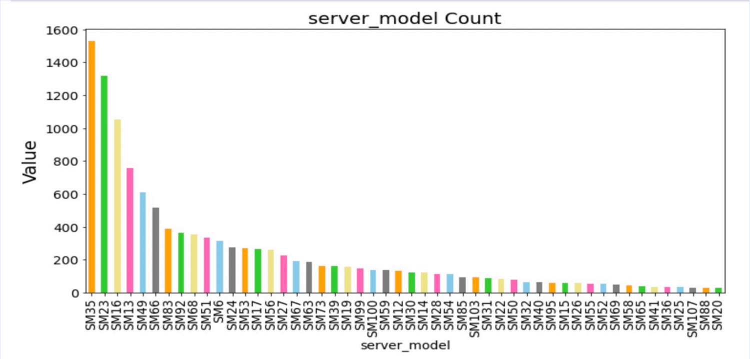
3、sn分布

- 唯一值达到13000+。



4、 server_model分布

- 服务器型号server_model和服务器序列号sn是一对多的关系。



数据分析

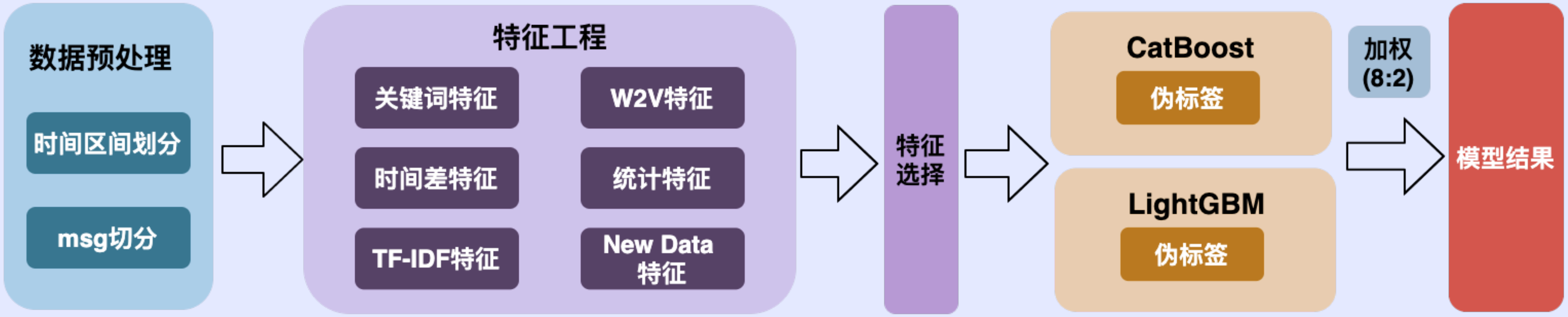
5、 msg分析

- 将msg经过TF-IDF编码，输入到线性模型中，使用**eli5**得出每个类别下，msg单词的贡献程度，权重越高表示区分该类别的贡献越大；
- 0类和1类表示CPU相关故障，processor的权重都是最高的；
- 2类表示内存相关故障，权重较高的是memory、mem、ecc;
- 3类表示其他类型故障，权重较高的是hdd、fpga、bus。
- 推断出：
  a. 0类和1类是cpu相关故障，且区分度不是很高
  b. 3类可能是和硬件相关的故障

类别	权重高的单词	解释
label_0/label_1	processor	处理器
label_2	memory/mem	内存
	ecc	纠错码
label_3	hdd	硬盘驱动器
	fpga	可编程门阵列器件
	bus	计算机系统总线

y=0 top features		y=1 top features		y=2 top features		y=3 top features	
Weight?	Feature	Weight?	Feature	Weight?	Feature	Weight?	Feature
+3.787	processor	+4.773	processor	+6.943	memory	+3.634	hdd
+3.167	asserted	+4.261	mcerr	+6.734	uncorrectable	+3.156	failure
+3.035	ps0	+3.720	0xfa	+4.929	mem	+2.595	fpga
+3.025	ierr	+2.917	statususe	+4.872	ecc	+2.336	bus
+2.643	0x7d	+2.419	0xf5	+2.837	limit	+2.247	0x8c
+2.546	caterr	+2.403	0xbe	+2.827	reached	+2.229	0x10
+2.427	0x82	+2.367	0xf9	+2.544	0x8b	+2.150	0xe9
+2.380	0x3b	+2.352	pressed	+2.493	disabled	+2.071	fatal
+2.367	watchdog2	+2.227	deasserted	+2.370	hardware	+2.068	vr
+2.247	correctable	+2.158	0x7c	+2.336	che0	+2.037	log
+2.246	nvmessd	+2.112	0x74	+2.267	limiting	+1.999	fault
+2.112	dimm101	+2.090	0xe4	+2.177	dimm	+1.996	restart
+2.072	0x13	+2.029	statusus	+2.092	device	+1.984	mngmnt
+2.044	mcerr	+1.976	cpu0	+2.087	recoverable	+1.964	sel
+1.999	thermal	+1.914	0xff	+2.052	dimm050	+1.960	dimm030
+1.910	inlet	+1.799	watchdog2	+2.031	0x84	+1.912	0x87
+1.896	0x0f	+1.736	0x3b	+1.999	0x90	+1.903	cpu1e1
+1.830	c3d1	+1.735	trip	+1.946	105	+1.830	dimm001
+1.822	c2d1	+1.735	dimm110	+1.940	undetermined	+1.811	dimm3a
+1.784	sta	+1.678	core	+1.922	dimm140	+1.800	detected
+1.732	cpu0	+1.673	ierr	+1.903	shutdown	+1.737	0xe2
+1.600	record	+1.673	running	+1.839	change	+1.706	cpu0d0
+1.598	000010003511	+1.625	correctable	+1.819	presence	+1.654	unit
+1.597	max	+1.624	dimm120	+1.789	watchdog	+1.638	controller
+1.597	tsb	+1.621	chg0	+1.743	0x75	+1.595	d0
+1.563	deasserted	+1.551	cpu2	+1.741	slot	+1.588	0x16
+1.549	c2	+1.435	dimm060	+1.714	0x96	+1.519	0x17
+1.546	temperature	+1.425	asserted	+1.610	0x98	+1.502	125
+1.545	0xfa	+1.391	lt	+1.596	bay	+1.472	powerlimiting
+1.467	ef	+1.387	prochot	+1.595	0x80	+1.464	0xef
+1.388	cpu3	+1.367	tota	+1.514	0x92	+1.439	bmc
+1.386	0xfe	+1.364	ipmi	+1.505	port2	+1.436	0x30
+1.380	dimm170	+1.339	present	+1.487	link	+1.389	12
+1.321	0x70	+1.293	0x71	+1.461	0x10	+1.366	specific
+1.317	d1	+1.277	ab	+1.439	0x8a	+1.340	100
+1.294	lan	+1.274	cpu0a1	+1.382	c5d0	+1.304	absent
+1.277	low	+1.271	critical	+1.369	dimm070	+1.298	interrupt

方案介绍



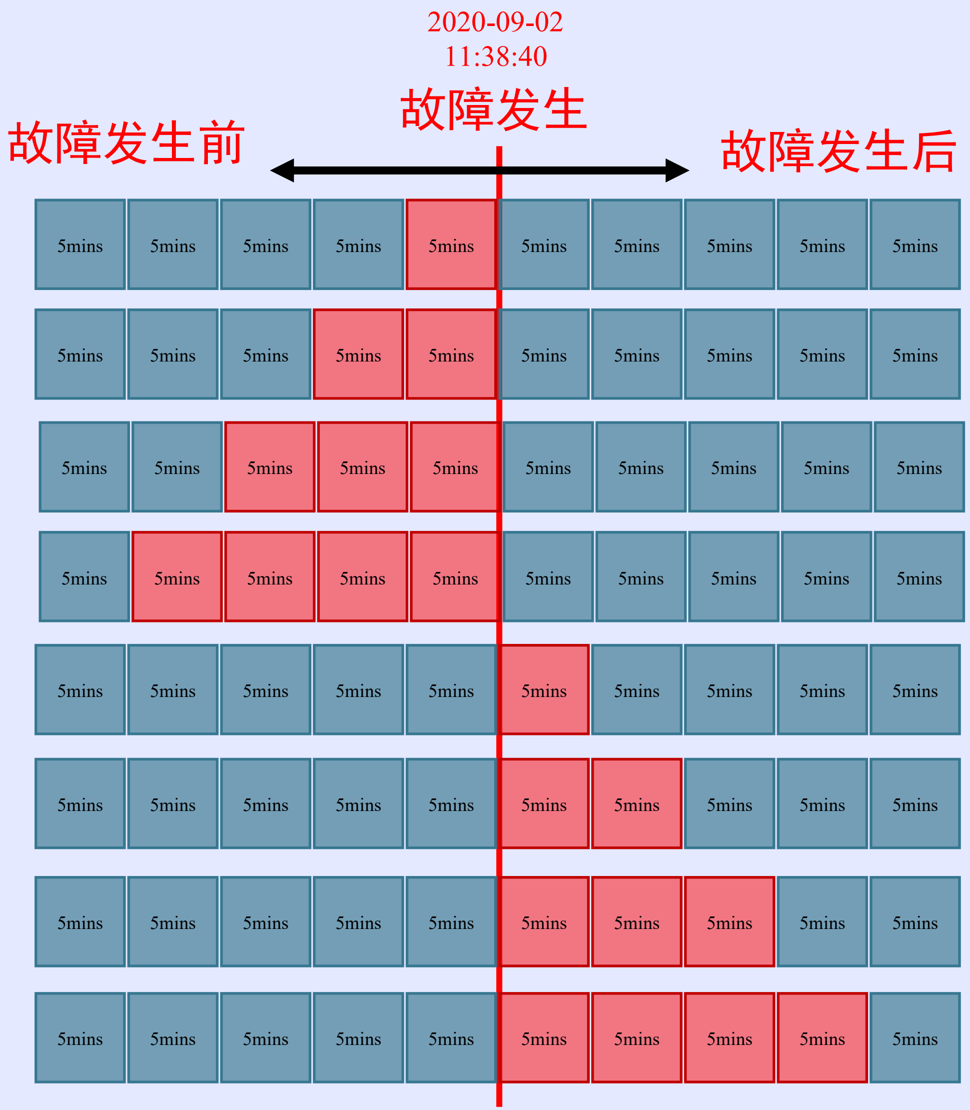
方案思想：基于智能运维场景，争取用更少的时间及资源达到最优效果，主要分为以下几个部分：

1. 数据预处理：按照距离故障发生时间的间隔，将日志分成不同的时间区间；msg根据特殊符号进行标准化。
2. 特征工程：主要构建关键词特征、时间差特征、TF-IDF词频统计特征、W2V特征、统计特征、New Data特征。
3. 特征选择：对抗验证进行特征选择，保证训练和测试集的一致性，提高模型在测试集的泛化能力。
4. 模型训练：CatBoost与LightGBM使用伪标签技术进行模型训练。
5. 模型融合：CatBoost与LightGBM的预测结果以8:2进行加权融合得到最终的模型预测结果。

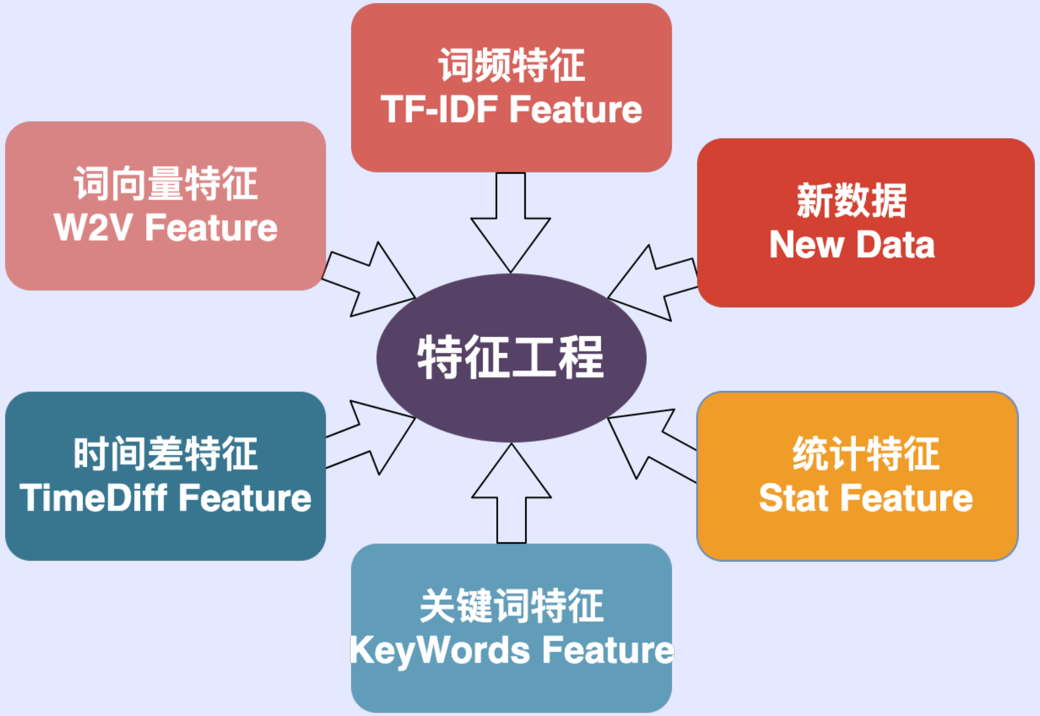
方案介绍—数据预处理

针对每一条故障工单，构造日志数据，用于后续的特征工程：

- 故障发生时间前5/10/15/30/60/120等分钟的日志数据
- 故障发生时间后5/10/15/30/60/120等分钟的日志数据
- 构造的每一份日志数据，都可以用于聚合构造统计特征



方案介绍—特征工程

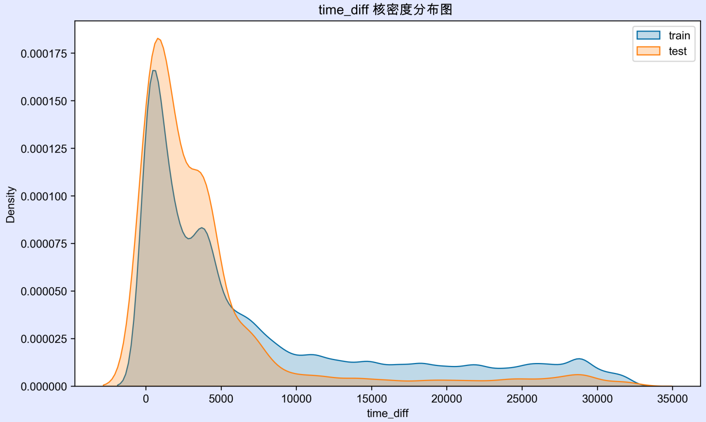


本方案从以下几个层面构建特征：

1. 时间差特征：反映故障日志与正常日志发生的间隔。
2. TF-IDF词频特征：反映每个词对每个类别的重要性。
3. W2V特征：反映msg的语义信息。
4. 统计特征：根据类别特征分组构造统计特征，使类别特征隐藏的信息充分暴露出来。
5. 关键词特征：反映关键词对各个类别的影响。
6. 新数据特征：新数据建模，预测结果作为新的特征。

方案介绍—特征工程

1、时间差特征



- 获取日志时间和故障发生时间的时间差，并且结合sn, server_model, 进行分组特征衍生。
- 时间差的统计特征：[max, min, median, std, var, skw, sum, mode]
- 时间差的分位数特征：[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
- 时间差差分: [max, min, median, std, var, skw, sum]

2、关键词特征

y=0 top features		y=1 top features		y=2 top features		y=3 top features	
Weight?	Feature	Weight?	Feature	Weight?	Feature	Weight?	Feature
+3.787	processor	+4.773	processor	+6.943	memory	+3.634	hdd
+3.167	asserted	+4.261	mcerr	+6.734	uncorrectable	+3.156	failure
+3.035	ps0	+3.720	0xfa	+4.929	mem	+2.595	fpga
+3.025	ierr	+2.917	statususe	+4.872	ecc	+2.336	bus
+2.643	0x7d	+2.419	0xf5	+2.837	limit	+2.247	0x8c
+2.546	caterr	+2.403	0xbe	+2.827	reached	+2.229	0x10
+2.427	0x82	+2.367	0xf9	+2.544	0x8b	+2.150	0xe9
+2.380	0x3b	+2.352	pressed	+2.493	disabled	+2.071	fatal
+2.367	watchdog2	+2.227	deasserted	+2.370	hardware	+2.068	vr
+2.247	correctable	+2.158	0x7c	+2.336	che0	+2.037	log
+2.246	nvmessd	+2.112	0x74	+2.267	limiting	+1.999	fault
+2.112	dimm101	+2.090	0xe4	+2.177	dimm	+1.996	restart
+2.072	0x13	+2.029	statusus	+2.092	device	+1.984	mngmnt
+2.044	mcerr	+1.976	cpu0	+2.087	recoverable	+1.964	sel
+1.999	thermal	+1.914	0xff	+2.052	dimm050	+1.960	dimm030
+1.910	inlet	+1.799	watchdog2	+2.031	0x84	+1.912	0x87
+1.896	0x0f	+1.736	0x3b	+1.999	0x90	+1.903	cpu1e1
+1.830	c3d1	+1.735	trip	+1.946	105	+1.830	dimm001
+1.822	c2d1	+1.735	dimm110	+1.940	undetermined	+1.811	dimm3a
+1.784	sta	+1.678	core	+1.922	dimm140	+1.800	detected
+1.732	cpu0	+1.673	ierr	+1.903	shutdown	+1.737	0xe2
+1.600	record	+1.673	running	+1.839	change	+1.706	cpu0d0
+1.598	000010003511	+1.625	correctable	+1.819	presence	+1.654	unit

方法1：基于TF-IDF模型

fea	fea_cnt_0	fea_cnt_1	fea_cnt_2	fea_cnt_3	fea_cnt_ratio_std	fea_max
Reading 87 > Threshold 85 degrees C	25	1	1	1	0.428571	0
OEM record c2	367	26	13	8	0.424711	0
Processor CPU_Core_Error	369	37	15	11	0.403646	0
Reading 71 > Threshold 85 degrees C	10	1	1	1	0.346154	0
Reading 86 > Threshold 85 degrees C	30	3	3	5	0.321960	0
OEM CPU0 CATERR	133	348	1	3	0.336693	1
OEM CPU0 MCERR	133	349	1	3	0.336953	1
Reading 0 < Threshold 2 degrees C	112	270	2	1	0.329890	1
Microcontroller #0xff	15	117	1	7	0.392619	1
S5/G2: soft-off	2	89	22	6	0.339782	1
Memory	1	14	9401	15	0.497880	2
Correctable ECC logging limit reached	9	152	7914	20	0.485160	2
Memory MEM_CHE0_Status	2	4	302	1	0.484914	2
Memory #0x87	266	147	17141	661	0.460850	2
Memory Memory_Status	169	74	4007	14	0.460062	2
Power Supply FPGA_PG	1	3	1	58	0.447340	3
Drive Fault	36	17	142	5425	0.476969	3
NMI/Diag Interrupt	4	20	2	510	0.467903	3
Failure detected	34	110	61	3034	0.457909	3
Drive Slot / Bay HDD_L_2_Status	4	1	1	62	0.441666	3

方法2: 基于词袋模型(统计词频数)

- 将msg经过TF-IDF编码，输入到线性模型中，计算每个类别下关键词的权重，取每个类别的TOP20。
- 根据' | '对msg进行分词，统计每个类别的词频占比，取每个类别的TOP20。
- 方法1与方法2取并集，得到最终的关键词。
- 每个关键词在msg中出现与否作为特征。

方案介绍—特征工程

3、统计特征

- 根据sn分组， server_mode统计特征:[count,nunique,freq,rank]
- 根据sn分组，日志统计特征：msg, msg_0, msg_1, msg_2:[count,nunique,freq,rank]

4、W2V特征

- 根据sn分组，按照时间对msg进行排序，对于每一个sn，将排序好的msg作为一个序列，提取embedding特征。

sn	msg_list
SERVER_10001	['Processor CPU1_Status', 'IERR', 'Asserted', 'Processor CPU0_Status', 'IERR', 'Asserted', 'Management Subsys Health System_Health', 'Sensor acce...
SERVER_10003	['Memory CPU1D0_DIMM_Stat', 'Correctable ECC', 'Asserted', 'Memory CPU1D0_DIMM_Stat', 'Correctable ECC', 'Asserted', 'Memory CPU1D0_DIM...
SERVER_10008	['Memory DIMM050_Stat', 'Uncorrectable ECC', 'Asserted', 'Processor CPU0_Status', 'Configuration Error', 'Asserted', 'Processor CPU0_Status', 'Co...
SERVER_10009	['Drive Slot HDD_L_14_Status', 'Drive Fault', 'Asserted', 'Drive Slot / Bay HDD_L_14_Status', 'Drive Fault', 'Asserted', 'Drive Slot HDD_L_14_Status', '...

5、TF-IDF特征

- 根据fault_id分组，对于每一个fault_id，将msg拼接作为一个序列，提取TF-IDF特征。

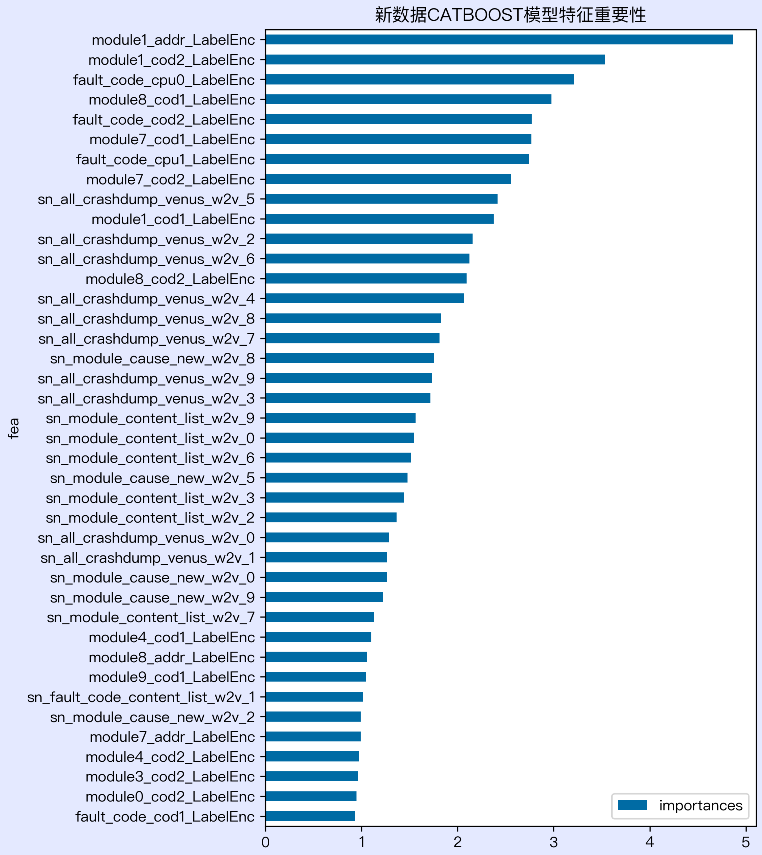
方案介绍—特征工程

6、venus、crashdump日志新数据特征构造

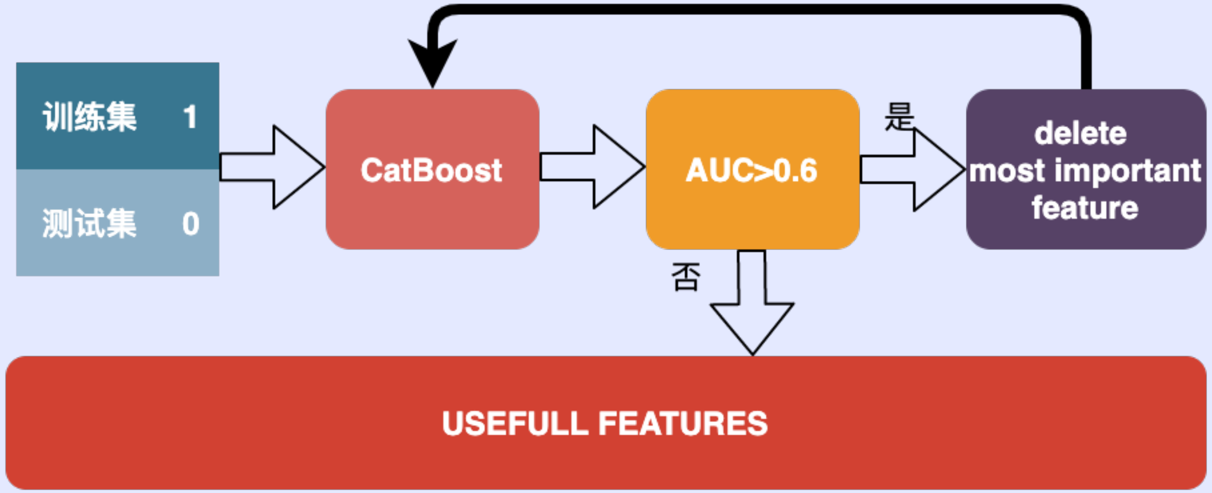
	fault_code_list	fault_code_cod1	fault_code_cod2	fault_code_cpu0	fault_code_cpu1
0	[cpu0, cha14, cod2, 0xc]	[' ']	['0xc']	['cha14']	[' ']
1	[cpu0, cha21, cod2, 0xc]	[' ']	['0xc']	['cha21']	[' ']
2	[cpu1, core0, cod1, 0x00800400]	['0x00800400']	[' ']	[' ']	['core0']
3	[cpu1, cha21, cod2, 0xc]	[' ']	['0xc']	[' ']	['cha21']
4	[cpu0, cha8, cod2, 0xc]	[' ']	['0xc']	['cha8']	[' ']
...	...	...	...	...	...
101	[cpu1, core25, cod1, 0x00000114]	['0x00000114']	[' ']	[' ']	['core25']
102	[cpu0, m2m0, cod2, 0x1]	[' ']	['0x1']	['m2m0']	[' ']
103	[cpu0, cha8, cod2, 0xc]	[' ']	['0xc']	['cha8']	[' ']
104	[cpu1, m2m0, cod2, 0x1]	[' ']	['0x1']	[' ']	['m2m0']
105	[cpu0, m2m0, cod2, 0x4]	[' ']	['0x4']	['m2m0']	[' ']

	module_cause_list	module3_cod1	module3_cod2	module3_addr	module4_cod1	module4_cod2	module4_addr
0	[module10, cod1:0x110a, cod2:0xc, addr:0x104bf...	['0x400']	['0x80']	['0xffffffff810779ab']	['0x400']	['0x80']	['0x7fbbe95ddc4a']
1	[module9, cod1:0x110a, cod2:0xc, addr:0xee501000]	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
2	[module3, cod1:0x400, cod2:0x80, addr:0x7f9455...	['0x400']	['0x80']	['0x7f945563bfdf']	['0x400']	['0x80']	['0x7f945563bfdf']
3	[module9, cod1:0x110a, cod2:0xc, addr:0xee501000]	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
4	[module11, cod1:0x110a, cod2:0xc, addr:0xc5d01...	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
...	...	...	...	...	...	...	...
101	[module1, cod1:0x114, cod2:0x0, addr:0x5e55c53...	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
102	[module7, cod1:0x92, cod2:0x101, addr:0xbd6f45...	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
103	[module11, cod1:0x110a, cod2:0xc, addr:0x40, m...	['0x400']	['0x80']	['0x564447b2f197']	['0x400']	['0x80']	['0x564447b2f197']
104	[module1, cod1:0x134, cod2:0x10, addr:0xb72c2a...	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']
105	[module1, cod1:0x134, cod2:0x10, addr:0x6ff4a000]	[' ']	[' ']	[' ']	[' ']	[' ']	[' ']

- 对新数据的数据格式进行标准化，关键词提取，生成类别特征。
- 通过新数据提取的特征构造模型，预测结果作为特征，与旧数据特征一起训练，新数据模型特征重要性如下：



方案介绍—特征选择

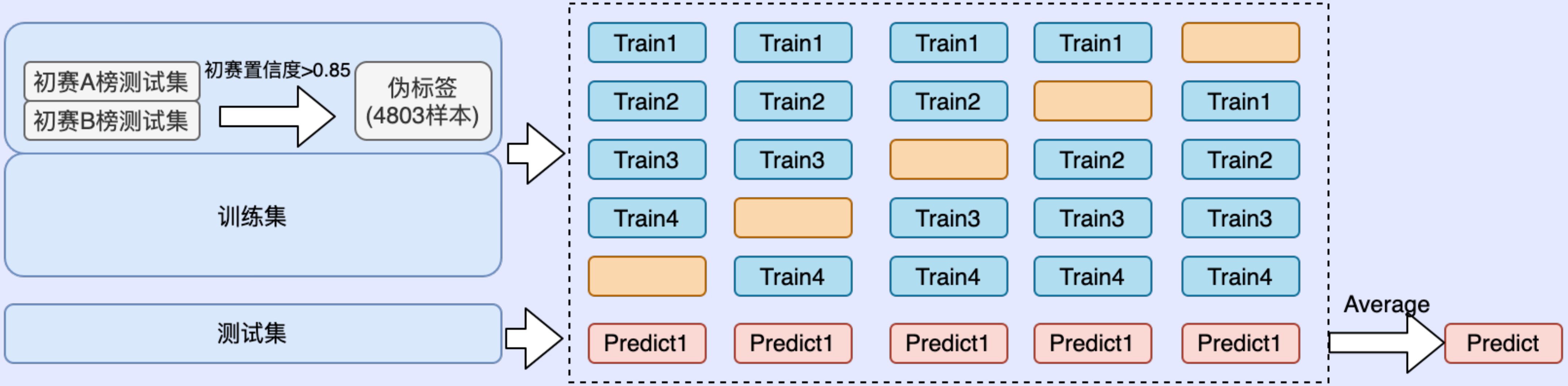


对抗验证特征选择

1. 准备好已完成特征构造的训练集和测试集。
2. 删除label，重新打标，训练集为1，测试集为0。
3. 数据集合并，训练 CatBoost 模型，计算AUC。
4. 设置一定的阈值(默认阈值为0.6，阈值必须大于0.5)。
5. 若AUC大于阈值，删除特征重要性最高的特征，重新执行第3步。
6. 所有第5步中被删除的特征，就是最终获得的训练集和测试集分布不一致的特征，不推荐在后续过程中使用。

方案介绍—模型训练

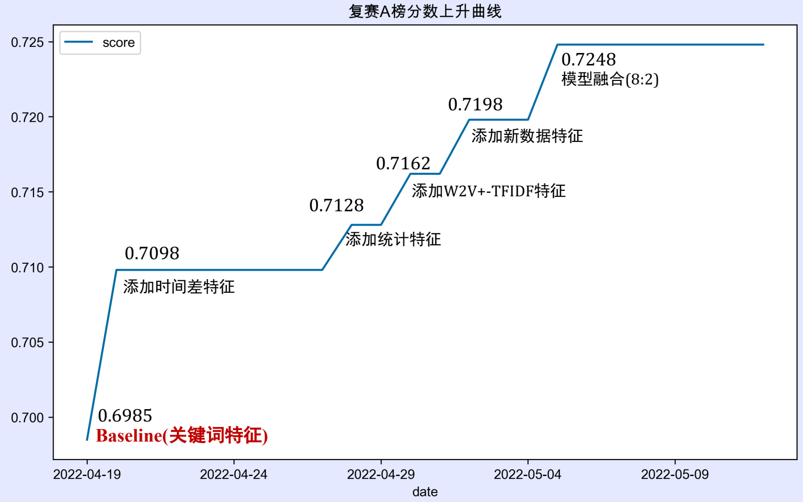
对于CatBoost与LightGBM均进行五折交叉构造出5个模型，将预测出来的结果进行平均作为单模型最终结果，以保证模型的稳定性。最终 CatBoost 与 LightGBM 的预测结果以8:2进行加权。

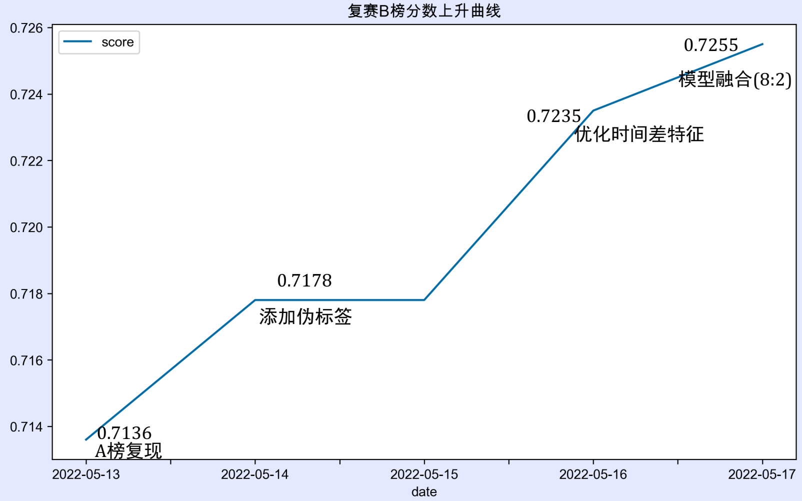


伪标签技术：

- 将A、B榜测试集的数据，选取模型预测结果置信度>0.85的样本作为可信样本，加入到训练集中，增大样本量。

方案介绍—模型效果





本方案代码运行环境基于8核32G服务器。主要使用两套特征模型，CatBoost 模型主要依赖统计与关键词特征，复赛B榜单模得分 0.7235，预测总耗时**仅需00:01:45（约0.04秒/条）**；LightGBM 模型主要依赖TF-IDF特征与W2V特征，复赛B榜单模得分 0.7167，预测耗时约**仅需00:02:07（约0.04秒/条）**。两者融合模型得分0.7255。

	CatBoost 模型		LightGBM 模型	
	训练耗时	**预测耗时**	**训练耗时**	**预测耗时**
特征工程	0:18:19	**0:01:44**	0:22:38	0:02:05
模型层	0:06:31	**0:00:01**	0:07:20	0:00:02
总耗时	52:39:01	**0:01:45**	66:25:06	0:02:07

总结展望

1. 模型方案要考虑到落地情况。如智能运维场景下，计算资源一般不会太多，或没有GPU资源，部分场景要求高及时性，越快找到问题并解决，就能越快的恢复对应的业务线。
2. 问题分析一定要考虑业务层面。想到可能有用的稍微有一点业务含义的特征就添加，哪怕不太确定，或者觉得和已有特征关联较大。
3. 使用对抗验证进行特征选择，保证训练集测试集的一致性，是提高模型准确性的关键。
4. 对msg信息的充分挖掘，也使得模型在AB榜测试相对稳定。

THANKS!