阿里云机器学习

最佳实践

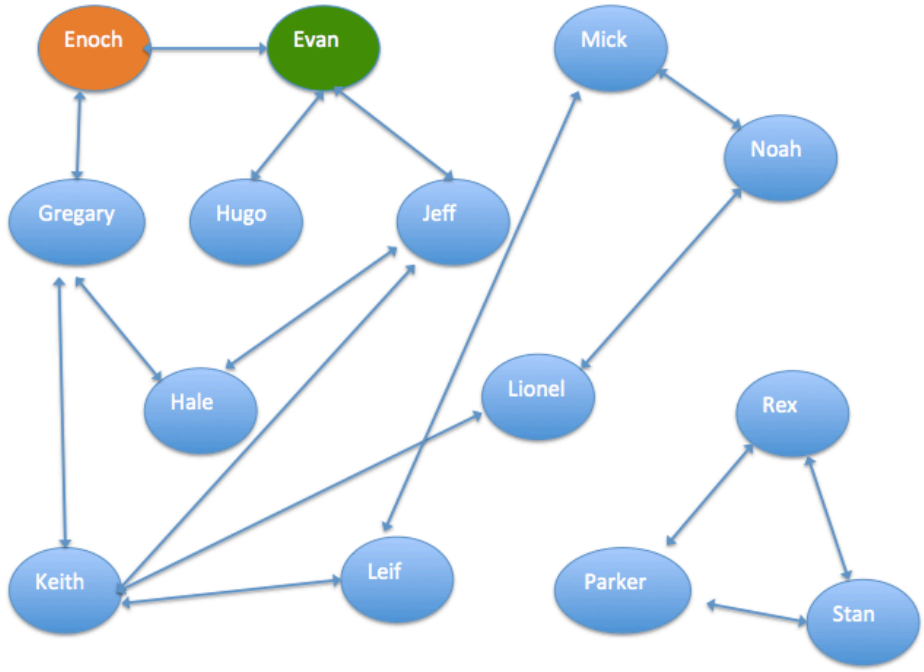
最佳实践

【图算法】金融风控实验

本文数据为虚构，仅供实验。

背景

图算法一般用来解决关系网状的业务场景。与常规的结构化数据不同，图算法需要把数据整理成首尾相连的关系图谱，更多考虑的是边和点的概念。阿里云机器学习平台上提供了丰富的图算法组件，包括K-Core、最大联通子图、标签传播聚类等。
本文档针对阿里云机器学习平台上图算法模块来进行实验，业务场景如下。
下图是已知的一份人物通联关系图，每两个人之间的连线表示两人有一定关系，可以是同事或者亲人关系等。已知“Enoch”是信用用户，“Evan”是欺诈用户。需要通过图算法，计算出其它人的信用指数，即得到图中每个人是欺诈用户的概率。这个数据可以方便相关机构做风控。



数据集介绍

具体字段如下表所示。

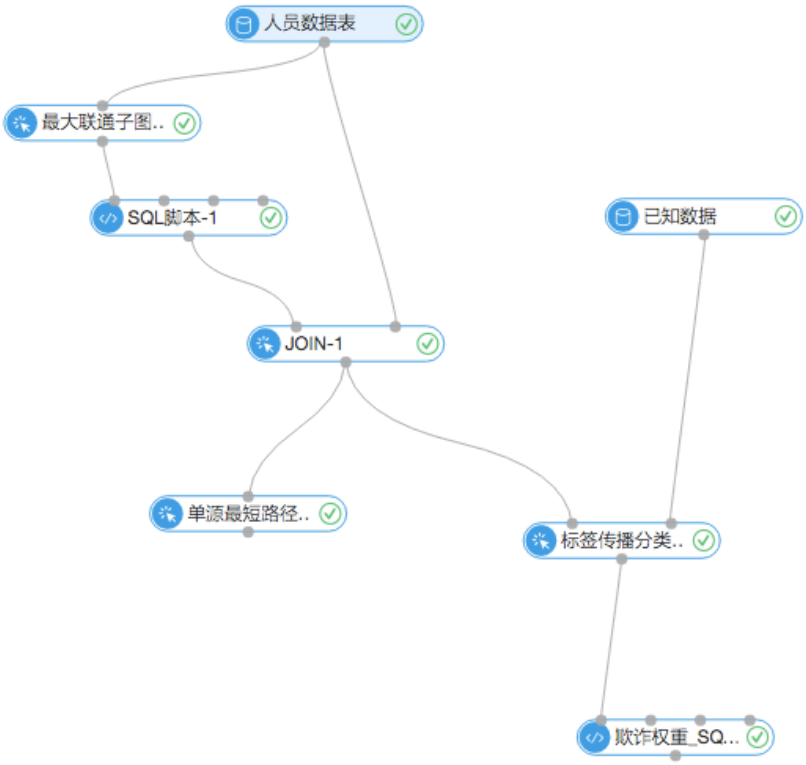
字段名	含义	类型	描述
start_point	边的起始节点	string	人
end_point	边结束节点	string	人
count	关系紧密度	double	数值越大，两人的关系越紧密

数据截图如下。

start_point ▲	end_point ▲	count ▲
Enoch	Evan	10
Enoch	Gregary	2
Gregary	Hale	6
Evan	Hugo	2
Evan	Jeff	4
Gregary	Keith	7
Jeff	Keith	5
Hale	Jeff	11
Keith	Leif	3
Keith	Lionel	1
Leif	Mick	4

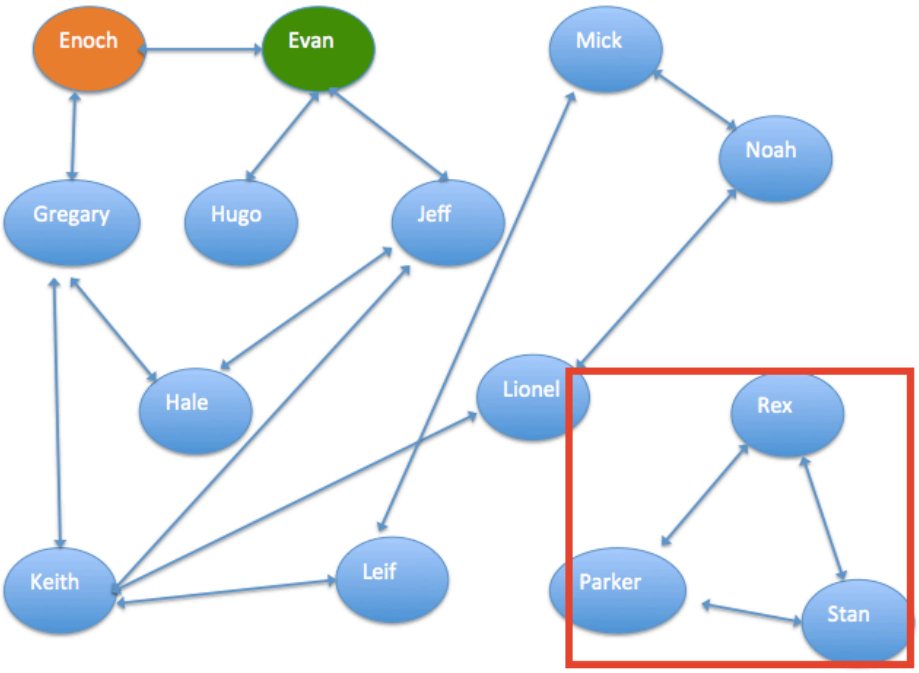
数据探索流程

实验流程图如下。



1. 最大联通子图

最大联通子图的功能：图算法的输入数据是关系图谱结构的，最大联通子图可以找到有通联关系的最大集合，在团伙发现的场景中可以排除掉一些与风控场景无关的人。
本次实验通过**最大联通子图**组件将数据中的群体分为两部分，并赋予group_id。通过**SQL脚本**组件和**JOIN**组件去除下图中的无关联人员。



2. 单源最短路径

通过**单源最短路径**组件探查出每个人的一度人脉、二度人脉等关系。“distance”表示“Enoch”通过几个人可以联络到目标人，如下图所示。

start_node ▲	dest_node ▲	distance ▲	distance_cnt ▲
Enoch	Hale	2	1
Enoch	Leif	3	1
Enoch	Hugo	2	1
Enoch	Keith	2	1
Enoch	Jeff	2	1
Enoch	Evan	1	1
Enoch	Lionel	3	1
Enoch	Mick	4	1
Enoch	Gregary	1	1
Enoch	Noah	4	1
Enoch	Enoch	0	0

3. 标签传播分类

标签传播分类算法为半监督的分类算法，原理是用已标记节点的标签信息去预测未标记节点的标签信息。在算

法执行过程中，每个节点的标签按相似度传播给相邻节点。
使用**标签传播分类**组件除了需要所有人员的通联图数据以外，还要有人员打标数据。本实验通过**已知数据（读数据表）**组件导入打标数据（ “weight”表示目标是欺诈用户的概率），如下图所示。

point	point_type	weight
Enoch	信用用户	1
Evan	欺诈用户	0.8

4. 结论

通过**SQL脚本**组件对结果进行筛选，最终展现的是每个人涉嫌欺诈的概率，数值越大表示是欺诈用户的概率越大，如下图所示。

node	tag	weight
Hugo	欺诈用户	1
Evan	欺诈用户	0.8
Noah	欺诈用户	0.42059743476528927
Jeff	欺诈用户	0.34784053907648443
Mick	欺诈用户	0.3113287445872401
Lionel	欺诈用户	0.2938277295951075
Leif	欺诈用户	0.24091136964145973
Keith	欺诈用户	0.2264783897173419

回归算法做农业贷款发放预测

本文数据为虚构，仅供实验。

背景

农业贷款发放问题是一个典型的数据挖掘问题。贷款发放人通过往年的数据，包括贷款人的年收入、种植的作物种类、历史借贷信息等特征来构建经验模型，通过这个模型来预测受贷人的还款能力。
本文档根据真实的农业贷款业务场景，利用线性回归算法解决贷款发放业务。线性回归是利用数理统计中的回归分析方法，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，运用十分广泛。本文通过农业贷款的历史发放情况，预测是否给预测集的用户发放他们所需金额的贷款。

数据集介绍

具体字段如下表所示。

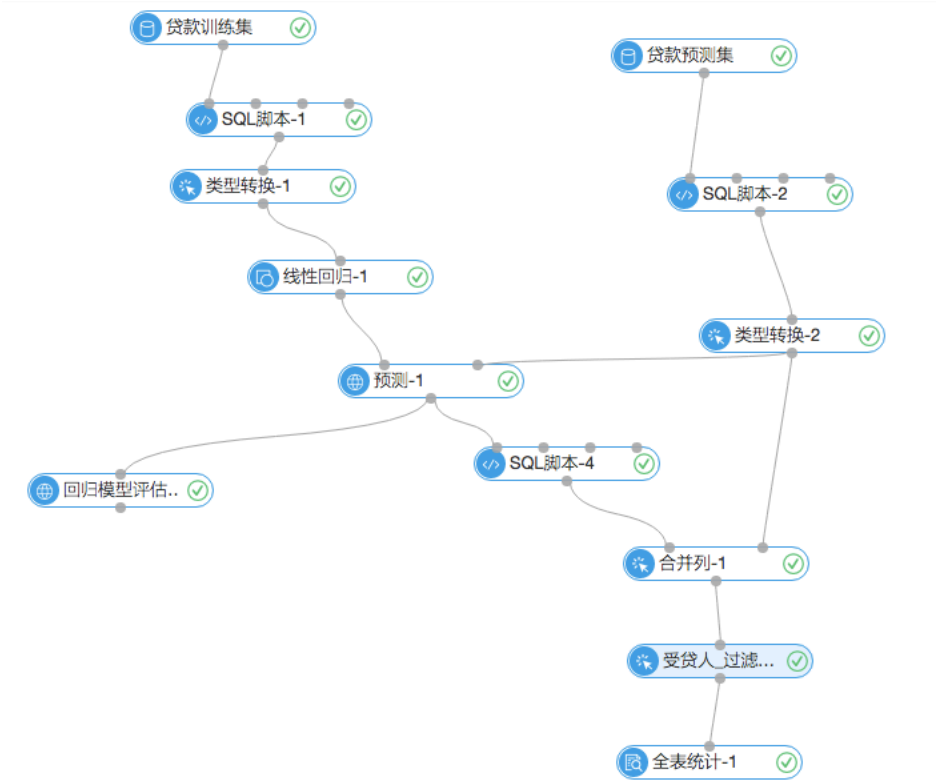
字段名	含义	类型	描述
id	数据唯一标识符	string	人
name	用户名	string	人
region	用户所属地区	string	从北到南排列
farmsize	拥有土地大小	double	土地面积
rainfall	降雨量	double	降雨量
landquality	土地质量	double	土地质量数值越大越好
farmincome	收入	double	年收入
maincrop	种植作物	string	种植作物的种类
claimtype	贷款类型	string	两种
claimvalue	贷款金额	double	贷款金额

数据截图如下。

id	name	region	farmsize	rainfall	landquality	farmincome	maincrop	claimtype	claimvalue
"id...	"name...	"midland...	1480	30	8	330729	"wheat"	"decommiss...	74703.1
"id...	"name...	"north"	1780	42	9	734118	"maize"	"arable_dev"	245354
"id...	"name...	"midland...	500	69	7	231965	"rapeseed"	"decommiss...	84213
"id...	"name...	"southw...	1860	103	3	625251	"potatoes"	"decommiss...	281082
"id...	"name...	"north"	1700	46	8	621148	"wheat"	"decommiss...	122006
"id...	"name...	"southea...	1580	42	7	445785	"maize"	"arable_dev"	122135
"id...	"name...	"southea...	1820	29	6	211605	"maize"	"arable_dev"	68969.2
"id...	"name...	"southea...	1640	108	7	1167040	"maize"	"arable_dev"	485011
"id...	"name...	"southw...	1600	101	5	756755	"wheat"	"decommiss...	160904
"id...	"name...	"southea...	600	80	6	267928	"wheat"	"arable_dev"	90350.6

数据探索流程

实验流程图如下。



1. 数据源准备

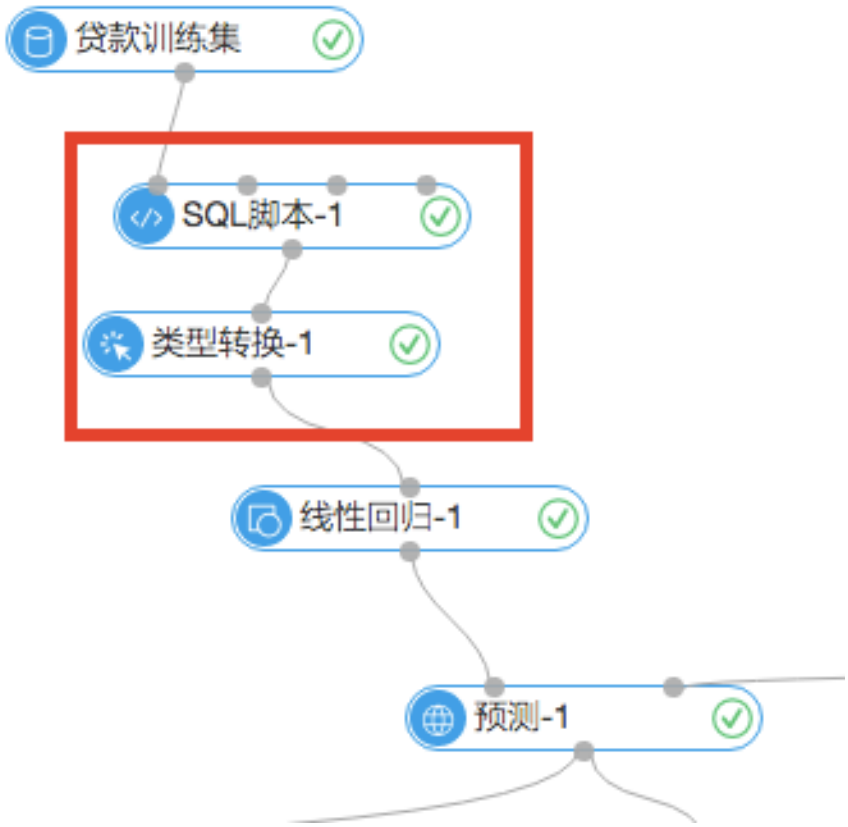
输入数据分为两部分：

- 贷款训练集：共二百余条历史贷款数据，用来训练回归模型。包括“farmsize”、“rainfall”等特征，“claimvalue”是贷款收回的金额。
- 贷款预测集：共七十一人，是今年申请贷款者，“claimvalue”是农民申请的贷款金额。

通过已有的二百余条历史数据，预测给七十一人中的哪些申请人发放贷款。

2. 数据预处理

根据含义将字符串类型的数据映射成数字。例如“region”字段，将其中的north、middle、south按照从北到南的顺序分别映射为0、1、2，再通过类型转换组件将字段转换成double类型，如下图所示。完成后即可进行模型训练。



3. 模型训练及预测

使用线性回归组件对历史数据进行训练并生成回归模型，在预测组件中利用回归模型对于预测集数据进行了预测。通过合并列组件将用户ID、预测值、申请的贷款值合并，结果如下图所示。
预测值表示的是用户的还贷能力（预期可以归还的金额）。

claimvalue ▲	prediction_score ▲	id ▲
172753	164424.3413395547	1
93415.4	146370.52166158534	2
46800.2	41879.999271195346	3
131728	192648.19077439874	4
89040.8	76369.8134277192	5
135493	103695.67105783387	6
88906.8	136845.30246967232	7
147159	144156.81362150217	8
277397	466728.8170899566	9
67547.3	131340.40980772747	10
345394	402192.7992950041	11

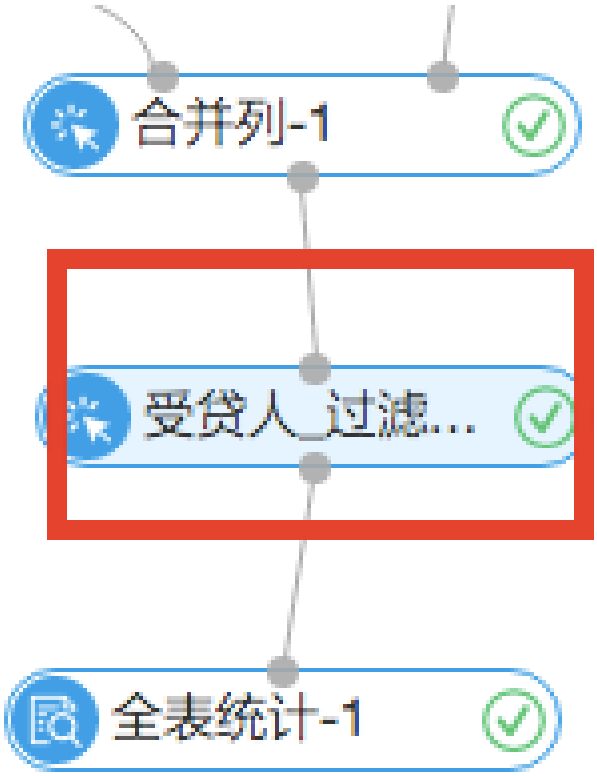
4. 回归模型评估

通过回归模型评估组件对模型进行评估，评估结果如下图所示。

字段名称	描述
SST	总平方和
SSE	误差平方和
SSR	回归平方和
R2	判定系数
R	多重相关系数
MSE	均方误差
RMSE	均方根误差
MAE	平均绝对误差
MAD	平均误差
MAPE	平均绝对百分误差
count	行数
yMean	原始因变量的均值
predictionMean	预测结果的均值

5. 贷款发放

通过过滤与映射组件筛选出可以获得贷款的人。实验的原理是针对每个客户，如果贷款人被预测得到的还款能力大于他申请贷款的金额，就给他发放贷款。



其它

请进入阿里云数加机器学习平台体验阿里云机器学习产品，并通过云栖社区公众号参与讨论。

评分卡信用评分

机器学习算法基于信用卡消费记录做信用评分

背景

评分卡是**信用风险评估**和**互联网金融**领域常用的建模方法，并不简单对应于某一种机器学习算法，而是一种通用的建模框架。它将原始数据通过分箱后进行特征工程变换，继而应用于线性模型进行建模。

评分卡建模理论常被用于各种信用评估领域，比如信用卡风险评估、贷款发放等业务。另外，在其它领域评分卡常被用来作为分数评估，比如常见的客服质量打分、芝麻信用分打分等。本文档通过一个案例讲解如何通过机器学习平台的金融板块组件，搭建出一套评分卡建模方案。

单击**加载更多**，可以直接从模板创建评分卡实验，如下图所示。该模板包含了整个实验的流程和数据。

数据集介绍

源表字段信息

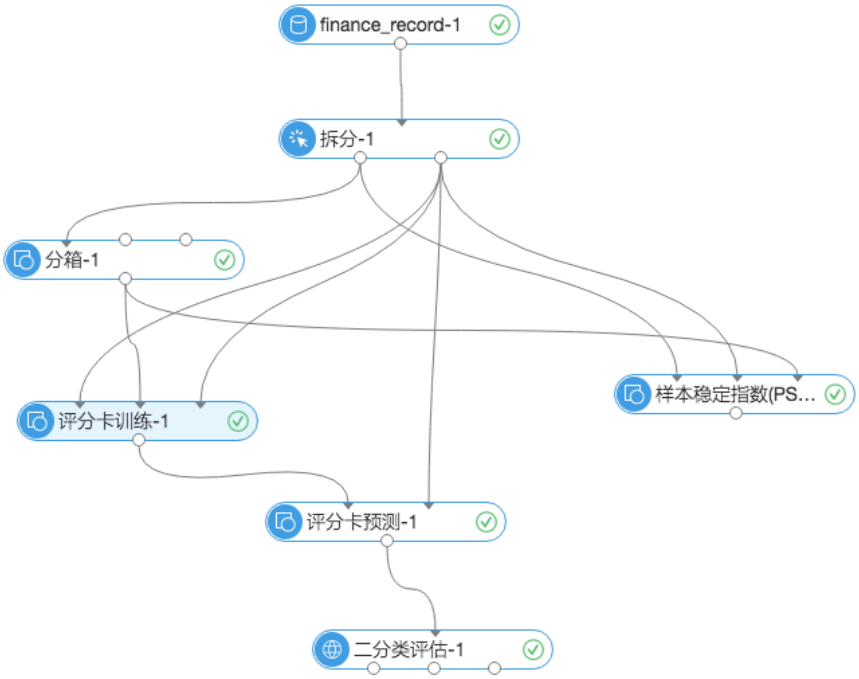
字段	类型	前 100 条记录
id	STRING	1,2,3,4,5
limit_bal	BIGINT	20000,50000,
sex	STRING	女,男
education	STRING	本科
marriage	STRING	已婚,未婚
age	BIGINT	24,26,34,37,5
pay_0	BIGINT	-1,0,2
pay_2	BIGINT	0,2
pay_3	BIGINT	-1,0
pay_4	BIGINT	-1,0
pay_5	BIGINT	-2,0
pay_6	BIGINT	-2,0,2
bill_amt1	DOUBLE	2682.0,3913.(
bill_amt2	DOUBLE	1725.0,3102.(
bill_amt3	DOUBLE	689.0,2682.0,
bill_amt4	DOUBLE	0.0,3272.0,14
bill_amt5	DOUBLE	0.0,3455.0,14
bill_amt6	DOUBLE	0.0,3261.0,15
pay_amt1	DOUBLE	0.0,1518.0,20
pay_amt2	DOUBLE	689.0,1000.0,
pay_amt3	DOUBLE	0.0,1000.0,12
pay_amt4	DOUBLE	0.0,1000.0,11
pay_amt5	DOUBLE	0.0,689.0,100

上图中是一份国外某机构开源的数据集，共30000条。包含了每个用户的性别、教育、婚姻、年龄等属性，及用户过去一段时间的信用卡消费情况和账单情况。payment_next_month是目标队列，表示用户是否偿还信用卡账单，1表示偿还，0表示没有偿还。

数据集下载地址：https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset

实验流程

实验流程图如下。



拆分
将输入数据集分为两部分，一部分用来训练模型，另一部分用来预测评估。

分箱
分箱组件类似于onehot编码，可以将数据按照分布映射成更高维度的特征。以age字段为例，分箱组件可以按照数据在不同区间的分布进行分箱操作，分箱结果如图所示。

	Index	Label	Constraint		WoE		Number			Rate		
			Operator	Value	WoE	Chart	Total	Positive	Negative	Total	Positive	Negative
	0	(-inf,25]			0.249		3082	822	2260	12.84%	15.5%	12.09%
	1	[25,27]			-0.12		2184	439	1745	9.1%	8.28%	9.33%
	2	[27,29]			-0.137		2421	480	1941	10.09%	9.05%	10.38%
	3	[29,31]			-0.196		2084	394	1690	8.68%	7.43%	9.04%
	4	[31,34]			-0.2		2791	526	2265	11.63%	9.92%	12.11%
	5	[34,37]			-0.016		2622	572	2050	10.93%	10.79%	10.96%
	6	[37,40]			-0.025		2224	482	1742	9.27%	9.09%	9.32%
	7	[40,43]			0.026		1823	411	1412	7.6%	7.75%	7.55%
	8	[43,49]			0.083		2628	619	2009	10.95%	11.67%	10.74%
	9	[49,+inf)			0.215		2141	557	1584	8.92%	10.51%	8.47%
	-2	ELSE					-	-	-	-	-	-

最终分箱组件的输出如下图所示，每个字段都被分箱到多个区间上。

序号	feature	json
1	limit_bal	{ "bin": {"norm": [{ "iv": 0.076602, "n": 2104, "p": 1187, "prate": 0.360681, "total": 3291, "value": "(-inf,30000]", "woe": 0.687921}, { "iv": 0.009549999999999999, "n": 2095...
2	age	{ "bin": {"norm": [{ "iv": 0.008506, "n": 2260, "p": 822, "prate": 0.26671, "total": 3082, "value": "(-inf,25]", "woe": 0.248953}, { "iv": 0.00126, "n": 1745, "p": 439, "prate": 0.2...
3	pay_0	{ "bin": {"norm": [{ "iv": 0.047172, "n": 5735, "p": 1052, "prate": 0.155002, "total": 6787, "value": "(-inf,-1]", "woe": -0.435562}, { "iv": 0.170225, "n": 10252, "p": 1518, "prat...
4	pay_2	{ "bin": {"norm": [{ "iv": 0.007479, "n": 2483, "p": 547, "prate": 0.180526, "total": 3030, "value": "(-inf,-2]", "woe": -0.252442}, { "iv": 0.028735, "n": 4094, "p": 779, "prate": ...
5	pay_3	{ "bin": {"norm": [{ "iv": 0.006939, "n": 2676, "p": 601, "prate": 0.183399, "total": 3277, "value": "(-inf,-2]", "woe": -0.233151}, { "iv": 0.032692, "n": 4040, "p": 744, "prate": ...
6	pay_4	{ "bin": {"norm": [{ "iv": 0.004796, "n": 2826, "p": 665, "prate": 0.19049, "total": 3491, "value": "(-inf,-2]", "woe": -0.186498}, { "iv": 0.02676, "n": 3858, "p": 736, "prate": 0.1...
7	pay_5	{ "bin": {"norm": [{ "iv": 0.003088, "n": 2925, "p": 717, "prate": 0.19687, "total": 3642, "value": "(-inf,-2]", "woe": -0.145641}, { "iv": 0.023437, "n": 3740, "p": 729, "prate": 0....
8	pay_6	{ "bin": {"norm": [{ "iv": 0.002296, "n": 3135, "p": 788, "prate": 0.200867, "total": 3923, "value": "(-inf,-2]", "woe": -0.120554}, { "iv": 0.019253, "n": 3847, "p": 783, "prate": ...
9	bill_amt1	{ "bin": {"norm": [{ "iv": 0.001611, "n": 1818, "p": 584, "prate": 0.243131, "total": 2402, "value": "(-inf,282]", "woe": 0.124741}, { "iv": 3e-06, "n": 1866, "p": 532, "prate": 0.2...
10	bill_amt2	{ "bin": {"norm": [{ "iv": 0.000701, "n": 1929, "p": 593, "prate": 0.235131, "total": 2522, "value": "(-inf,0]", "woe": 0.08076999999999999}, { "iv": 0, "n": 1789, "p": 508, "prat...
11	bill_amt3	{ "bin": {"norm": [{ "iv": 0.000503, "n": 2158, "p": 653, "prate": 0.232302, "total": 2811, "value": "(-inf,0]", "woe": 0.064972}, { "iv": 5.2e-05, "n": 1541, "p": 448, "prate": 0.2...
12	bill_amt4	{ "bin": {"norm": [{ "iv": 0.000712, "n": 2362, "p": 721, "prate": 0.233863, "total": 3083, "value": "(-inf,0]", "woe": 0.073708}, { "iv": 0.000344, "n": 1317, "p": 400, "prate": 0....
13	bill_amt5	{ "bin": {"norm": [{ "iv": 0.001599, "n": 2535, "p": 799, "prate": 0.239652, "total": 3334, "value": "(-inf,0]", "woe": 0.105744}, { "iv": 2.4e-05, "n": 1141, "p": 330, "prate": 0.2...
14	bill_amt6	{ "bin": {"norm": [{ "iv": 0.0002, "n": 2917, "p": 857, "prate": 0.22708, "total": 3774, "value": "(-inf,0]", "woe": 0.035459}, { "iv": 0.000112, "n": 791, "p": 236, "prate": 0.2297...
15	pay_amt1	{ "bin": {"norm": [{ "iv": 0.098387, "n": 2681, "p": 1516, "prate": 0.36121, "total": 4197, "value": "(-inf,0]", "woe": 0.690218}, { "iv": 0.000189, "n": 463, "p": 143, "prate": 0.2...
16	pay_amt2	{ "bin": {"norm": [{ "iv": 0.068019, "n": 2864, "p": 1441, "prate": 0.334727, "total": 4305, "value": "(-inf,0]", "woe": 0.573451}, { "iv": 0.002296, "n": 356, "p": 139, "prate": 0....
17	pay_amt3	{ "bin": {"norm": [{ "iv": 0.061212, "n": 3232, "p": 1541, "prate": 0.322858, "total": 4773, "value": "(-inf,0]", "woe": 0.519663}, { "iv": 7.7e-05, "n": 31, "p": 7, "prate": 0.1842...

样本稳定指数PSI

样本稳定指数是衡量样本变化所产生的偏移量的一种重要指标，通常用来衡量样本的稳定程度。比如样本在两个月份之间的变化是否稳定。通常变量的PSI值在0.1以下表示变化不太显著，在0.1到0.25之间表示变化比较显著，大于0.25表示变量变化比较剧烈，需要特殊关注。

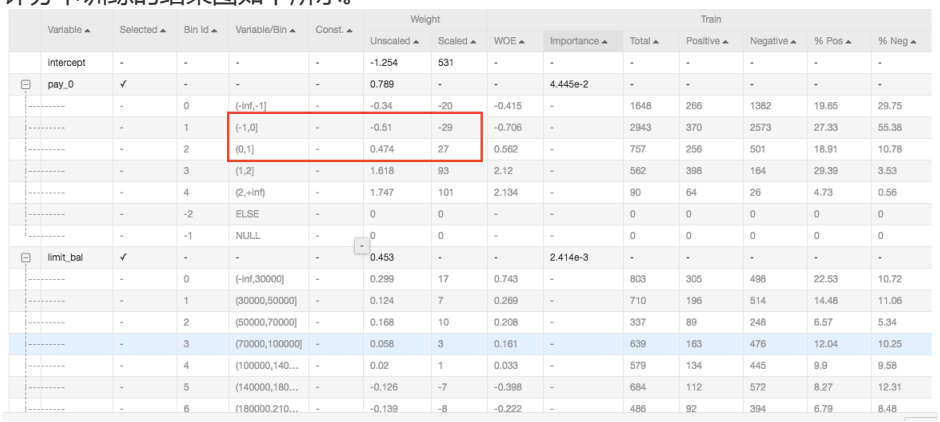
本案例中，综合比较拆分前后以及分箱结果的样本稳定程度，返回每个特征的PSI数值，如下图所示

收起 展开

Feature	Bin	Test %	Base %	Test - Base	ln(Test/Base)	PSI
limit_bal	-	-	-	-	-	0.0019
age	-	-	-	-	-	0.0005
pay_0	-	-	-	-	-	0.0002
pay_2	-	-	-	-	-	0.0006
pay_3	-	-	-	-	-	0.0005
pay_4	-	-	-	-	-	0.0016
pay_5	-	-	-	-	-	0.0015
pay_6	-	-	-	-	-	0.0019
bill_amt1	-	-	-	-	-	0.001
bill_amt2	-	-	-	-	-	0.0025
bill_amt3	-	-	-	-	-	0.0022
bill_amt4	-	-	-	-	-	0.0014
bill_amt5	-	-	-	-	-	0.0011
bill_amt6	-	-	-	-	-	0.0009
pay_amt1	-	-	-	-	-	0.0032
pay_amt2	-	-	-	-	-	0.0009

评分卡训练

评分卡训练的结果图如下所示。



	Variable	Selected	Bin Id	Variable/Bin	Const.	Weight		WOE	Importance	Train				
						Unscaled	Scaled			Total	Positive	Negative	% Pos	% Neg
	intercept	-	-	-	-	-1.254	531	-	-	-	-	-	-	-
⊟	pay_0	✓	-	-	-	0.789	-	-	4.445e-2	-	-	-	-	-
	---------	-	0	(-inf,-1]	-	-0.34	-20	-0.415	-	1648	266	1382	19.65	28.75
	---------	-	1	(-1,0]	-	-0.51	-29	-0.706	-	2943	370	2573	27.33	55.38
	---------	-	2	(0,1]	-	0.474	27	0.562	-	757	256	501	18.91	10.78
	---------	-	3	(1,2]	-	1.618	93	2.12	-	562	398	164	29.39	3.53
	---------	-	4	(2,+inf)	-	1.747	101	2.134	-	90	64	26	4.73	0.56
	---------	-	-2	ELSE	-	0	0	-	-	0	0	0	0	0
	---------	-	-1	NULL	-	0	0	-	-	0	0	0	0	0
⊟	limit_bal	✓	-	-	-	0.453	-	-	2.414e-3	-	-	-	-	-
	---------	-	0	(-inf,30000]	-	0.299	17	0.743	-	803	305	498	22.53	10.72
	---------	-	1	(30000,50000]	-	0.124	7	0.269	-	710	196	514	14.48	11.06
	---------	-	2	(50000,70000]	-	0.168	10	0.208	-	337	89	248	6.57	5.34
	---------	-	3	(70000,100000]	-	0.058	3	0.161	-	639	163	476	12.04	10.25
	---------	-	4	(100000,140...	-	0.02	1	0.033	-	579	134	445	9.9	9.58
	---------	-	5	(140000,180...	-	-0.126	-7	-0.398	-	684	112	572	8.27	12.31
	---------	-	6	(180000,210...	-	-0.139	-8	-0.222	-	486	92	394	6.79	8.48

评分卡的精髓是将复杂的模型权重用符合业务标准的分数表示。

- intercepy：截距。
- Unscaled：原始的权重值。
- Scaled：分数更改指标，比如对于pay_0这个特征，如果特征落在(-1,0]之间分数就减29，如果特征落在（0，1]之间分数就加上27。
- importance：每个特征对于结果的影响大小，数值越大表示影响越大。

评分卡预测

每个预测结果的最终评分，本案例中表示的是每个用户的信用评分。

序号	payment_next_month	prediction_score	prediction_prob	prediction_detail
1	0	499	0.14314626458020613	{"0":0.8568537354,"1":0.1431462646}
2	0	564	0.3367775480162267	{"0":0.6632224520,"1":0.3367775480}
3	0	555	0.3035873747480541	{"0":0.6964126253,"1":0.3035873747}
4	1	519	0.18818103244164777	{"0":0.8118189676,"1":0.1881810324}
5	1	651	0.7013570482913543	{"0":0.2986429517,"1":0.7013570483}
6	0	502	0.1474992646536902	{"0":0.8525007353,"1":0.1474992647}
7	1	560	0.3199046397072833	{"0":0.6800953603,"1":0.3199046397}
8	0	435	0.05207880036730361	{"0":0.9479211996,"1":0.0520788004}
9	0	491	0.12535852489673346	{"0":0.8746414751,"1":0.1253585249}

结论

基于用户的信用卡消费记录，通过评分卡模型训练及评分卡预测得到了每个用户的最终信用评分，这个评分可以应用到各种贷款或者金融相关的征信领域中。

心脏病预测案例

背景

心脏病是人类健康的头号杀手。全世界1/3的人口死亡是心脏病引起的。而我国，每年有几十万人死于心脏病。

如果可以通过提取人体相关的体测指标，通过数据挖掘方式来分析不同特征对于心脏病的影响，将对预防心脏病起到至关重要的作用。本文提供真实的数据，并通过阿里云机器学习平台搭建心脏病预测案例。

数据集介绍

数据源为UCI开源数据集heart_disease。包含了303条美国某区域的心脏病检查患者的体测数据。具体字段如下表。

字段名	含义	类型	描述
age	年龄	string	对象的年龄，数字表示
sex	性别	string	对象的性别，female和male
cp	胸部疼痛类型	string	痛感由重到无typical、atypical、non-anginal、asymptomatic
trestbps	血压	string	血压数值
chol	胆固醇	string	胆固醇数值
fbs	空腹血糖	string	血糖含量大于120mg/dl为true，否则为false
restecg	心电图结果	string	是否有T波，由轻到重为norm、hyp
thalach	最大心跳数	string	最大心跳数
exang	运动时是否心绞痛	string	是否有心绞痛，true为是，false为否
oldpeak	运动相对于休息的ST depression	string	st段压数值
slop	心电图ST segment的倾斜度	string	ST segment的slope，程度分为down、flat、up
ca	透视检查看到的血管数	string	透视检查看到的血管数
thal	缺陷种类	string	并发种类，由轻到重norm、fix、rev
status	是否患病	string	是否患病，buff是健康、sick是患病

数据探索流程

数据挖掘流程如下：

整体实验流程：

1. 数据预处理

数据预处理也叫作数据清洗，主要在数据进入算法流程前对数据进行去噪、缺失值填充、类型变换等操作。本次实验的输入数据包括14个特征列和1个目标列。需要解决的问题是根据用户的体检指标预测是否会患有心脏病，每个样本只有患病或不患病两种情况，是分类问题。
本次分类实验选用的是线性模型逻辑回归，要求输入的特征都是double类型的数据，如下图所示。

数据探查 - heart_disease_prediction - (仅显示前一百条)

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop	ca	thal	status	style
63.0	male	ang...	145.0	233.0	true	hyp	150.0	fal	2.3	down	0.0	fix	buff	H
67.0	male	asy...	160.0	286.0	fal	hyp	108.0	true	1.5	flat	3.0	norm	sick	S2
67.0	male	asy...	120.0	229.0	fal	hyp	129.0	true	2.6	flat	2.0	rev	sick	S1
37.0	male	not...	130.0	250.0	fal	norm	187.0	fal	3.5	down	0.0	norm	buff	H
41.0	fem	abn...	130.0	204.0	fal	hyp	172.0	fal	1.4	up	0.0	norm	buff	H
56.0	male	abn...	120.0	236.0	fal	norm	178.0	fal	0.8	up	0.0	norm	buff	H
62.0	fem	asy...	140.0	268.0	fal	hyp	160.0	fal	3.6	down	2.0	norm	sick	S3
57.0	fem	asy...	120.0	354.0	fal	norm	163.0	true	0.6	up	0.0	norm	buff	H
63.0	male	asy...	130.0	254.0	fal	hyp	147.0	fal	1.4	flat	1.0	rev	sick	S2
53.0	male	asy...	140.0	203.0	true	hy	155.0	true	3.1	down	0.0	rev	sick	S1

上图中很多数据都是文字描述的，在数据预处理的过程中需要根据每个字段的含义将字符转为数值。

二值类的数据
比如sex字段有female和male两种形式，可以将female表示成0，male表示成1。

多值类的数据
比如cp字段，表示胸部的疼痛感，可以将疼痛感由轻到重映射成0～3的数值。

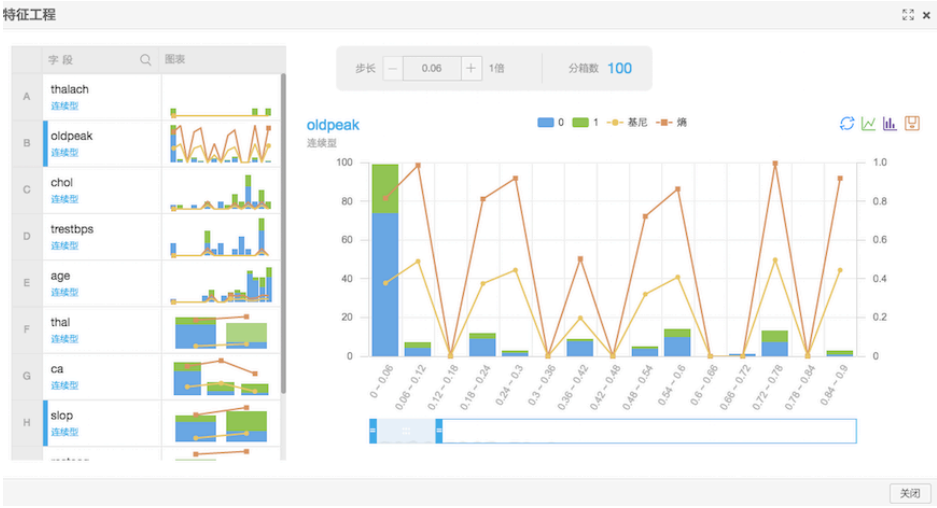
数据预处理通过sql脚本来实现，具体请参考SQL脚本组件。

```
select age,
(case sex when 'male' then 1 else 0 end) as sex,
(case cp when 'angina' then 0 when 'notang' then 1 else 2 end) as cp,
trestbps,
chol,
(case fbs when 'true' then 1 else 0 end) as fbs,
(case restecg when 'norm' then 0 when 'abn' then 1 else 2 end) as restecg,
thalach,
(case exang when 'true' then 1 else 0 end) as exang,
oldpeak,
(case slop when 'up' then 0 when 'flat' then 1 else 2 end) as slop,
ca,
(case thal when 'norm' then 0 when 'fix' then 1 else 2 end) as thal,
(case status when 'sick' then 1 else 0 end) as ifHealth
from ${t1};
```

2. 特征工程

特征工程主要包括特征的衍生、尺度变化等功能。本案例中有两个组件负责特征工程部分。

过滤式特征选择
判断每个特征对于结果的影响，通过信息熵和基尼系数来表示。右键单击组件，选择**查看评估报告**显示最终结果，如下图所示。



归一化

将每个特征的数值范围变为0到1之间，可以去除量纲对结果的影响，公式为result=(val-min)/(max-min)。本次实验通过逻辑回归二分类来进行模型训练，需要每个特征去除量纲的影响。归一化结果如下图所示。

数据探查 - pai_temp_2954_36756_1 - (仅显示前一百条)

sex	cp	fbs	restecg	exang	slop	thal	ifhealth	age	trestbps	chol	thalach	oldpeak
1	0	1	1	0	1	0.5	0	0.70...	0.4811320...	0.244...	0.603053...	0.370967...
1	1	0	1	1	0.5	0	1	0.79...	0.6226415...	0.365...	0.282442...	0.241935...
1	1	0	1	1	0.5	1	1	0.79...	0.2452830...	0.235...	0.442748...	0.419354...
1	0.5	0	0	0	1	0	0	0.16...	0.3396226...	0.283...	0.885496...	0.564516...
0	1	0	1	0	0	0	0	0.25	0.3396226...	0.178...	0.770992...	0.225806...
1	1	0	0	0	0	0	0	0.5625	0.2452830...	0.251...	0.816793...	0.129032...
0	1	0	1	0	1	0	1	0.6875	0.4339622...	0.324...	0.679389...	0.580645...
0	1	0	0	1	0	0	0	0.58...	0.2452830...	0.520...	0.702290...	0.096774...
1	1	0	1	0	0.5	1	1	0.70...	0.3396226...	0.292...	0.580152...	0.225806...
1	1	1	1	1	1	1	1	0.5	0.4339622...	0.175...	0.641221...	0.5
1	1	0	0	0	0.5	0	0	0.58...	0.4339622...	0.150...	0.587786...	0.064516...

3. 模型训练和预测

监督学习就是已知结果来训练模型。因为已经知道每个样本是否患有心脏病，因此本次实验是监督学习。解决的问题是预测一组用户是否患有心脏病。

拆分

通过拆分组件将数据分为两部分，本次实验按照训练集和预测集7：3的比例拆分。训练集数据流入逻辑回归二分类组件用来训练模型，预测集数据进入预测组件。

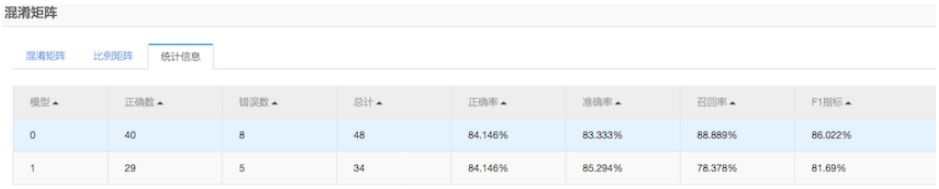
逻辑回归二分类

逻辑回归是一个线性模型，通过计算结果的阈值实现分类（具体的算法详情请自行查阅资料）。逻辑回归训练好的模型可以在模型页签中查看。

预测
预测组件的两个输入桩分别是模型和预测集。预测结果展示的是预测数据、真实数据、每组数据不同结果的概率。

4. 评估

通过混淆矩阵组件可以查看模型的准确率等参数。



通过此组件可以方便地根据预测的准确性来评估模型。

总结

通过以上数据探索流程可以得到以下结论。

特征权重

- 通过过滤式特征选择组件得到每个特征对于结果的权重。

featname ▲	weight ▲
thalach	0.1656917122459715 7
oldpeak	0.1464069761877935 2
thal	0.1376916655990601 5
ca	0.1146709754621757 5
chol	0.1026770957660085 9
age	0.0787643048452784 1
trestbps	0.0772599125640569
slop	0.0770276260907830 6
restecg	0.0152468324974051 05
cp	0.0037507283721422424
exang	0
fbs	0
sex	0

- thalach（心跳数）对于是否发生心脏病影响最大。
- 性别对于是否发生心脏病没有影响。

模型效果
通过本文档提供的14个特征，心脏病预测准确率可以达到百分之八十以上。模型可以用来做预测，辅助医生预防和治疗心脏病。

新闻分类案例

本文数据为虚构，仅供实验。

本实验拟在介绍文本类组件。如果您有相关的需求，想要提高最终的效果，请联系我们。我们为您提供完整的解决方案和商业合作。

背景

新闻分类是文本挖掘领域较为常见的场景。目前很多媒体或是内容生产商对于新闻这种文本的分类常常采用人肉打标的方式，消耗了大量的人力资源。本文通过智能的文本挖掘算法对新闻文本进行分类。无需任何人肉打标，完全由机器智能化实现。

本文通过PLDA算法挖掘文章的主题，通过主题权重的聚类，实现新闻自动分类。包括了分词、词型转换、停用词过滤、主题挖掘、聚类等流程。

数据集介绍

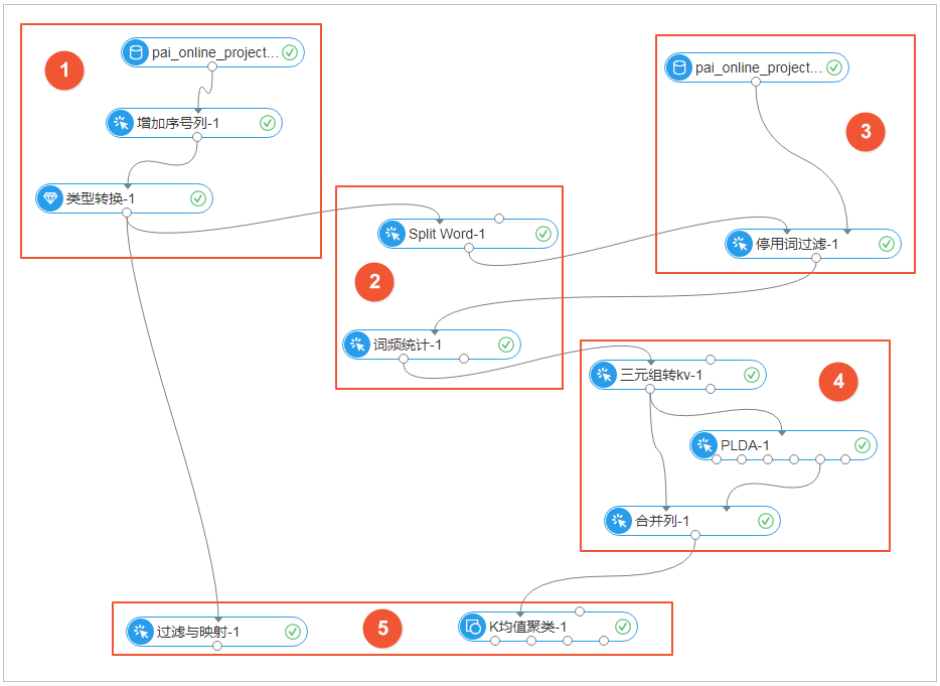
数据截图如下图所示。

具体字段如下：

字段名	含义	类型	描述
category	新闻类型	string	体育、女性、社会、军事、科技等
title	标题	string	新闻标题
content	内容	string	新闻内容

数据探索流程

实验流程图如下：



实验大致分为以下五个步骤：

- 1：增加序号列
- 2：停用词过滤
- 3：分词及词频统计
- 4：文本主题挖掘
- 5：结果分析和评估

1. 增加序号列

本实验的数据源是以单个新闻为单元，需要增加ID列来作为每篇新闻的唯一标识，方便下面算法的计算。

2. 分词及词频统计

这两步都是文本挖掘领域最常规的做法。

首先使用分词组件对**content**字段（新闻内容）进行分词。**去除过滤词**之后（过滤词一般是标点符号及助语），再对词频进行统计。结果如下图所示。

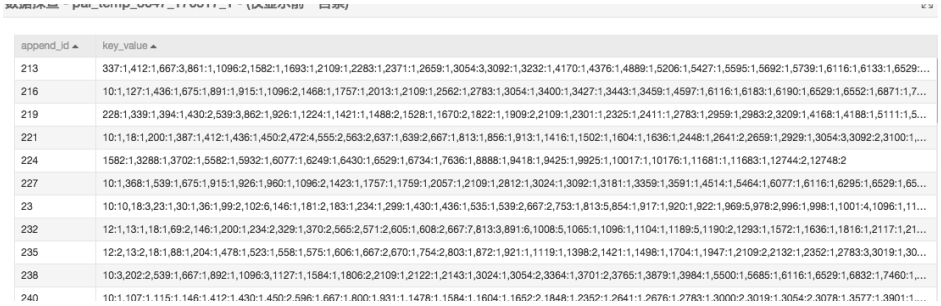
append_id ▲	word ▲	count ▲
0	分分	1
0	别墅	1
0	勇敢	1
0	包装	1
0	博爱	1
0	却	1
0	又	2
0	发	1

3. 停用词过滤

停用词过滤组件用于过滤输入的停用词词库，一般过滤标点符号以及对文章影响较小的助语等。

4. 文本主题挖掘

使用PLDA文本挖掘组件需要先将文本转换成三元形式（文本转数字），结果如下图所示。



append_id ▲	key_value ▲
213	337:1,412:1,667:3,861:1,1096:2,1582:1,1693:1,2109:1,2283:1,2371:1,2659:1,3054:3,3092:1,3232:1,4170:1,4376:1,4889:1,5206:1,5427:1,5595:1,5692:1,5739:1,6116:1,6133:1,6529:...
216	10:1,127:1,436:1,675:1,891:1,915:1,1096:2,1468:1,1757:1,2013:1,2109:1,2562:1,2783:1,3054:1,3400:1,3427:1,3443:1,3459:1,4597:1,6116:1,6183:1,6190:1,6529:1,6552:1,6871:1,7...
219	228:1,339:1,394:1,430:2,539:3,862:1,926:1,1224:1,1421:1,1488:2,1528:1,1670:2,1822:1,1909:2,2109:1,2301:1,2325:1,2411:1,2783:1,2959:1,2983:2,3209:1,4168:1,4188:1,5111:1,5...
221	10:1,18:1,200:1,387:1,412:1,436:1,450:2,472:4,555:2,563:2,637:1,639:2,667:1,813:1,856:1,913:1,1416:1,1502:1,1604:1,1636:1,2448:1,2641:2,2659:1,2929:1,3054:3,3092:2,3100:1,...
224	1582:1,3288:1,3702:1,5582:1,5932:1,6077:1,6249:1,6430:1,6529:1,6734:1,7636:1,8888:1,9418:1,9425:1,9925:1,10017:1,10176:1,11681:1,11683:1,12744:2,12748:2
227	10:1,368:1,539:1,675:1,915:1,926:1,960:1,1096:2,1423:1,1757:1,1759:1,2057:1,2109:1,2812:1,3024:1,3092:1,3181:1,3359:1,3591:1,4514:1,5464:1,6077:1,6116:1,6295:1,6529:1,65...
23	10:10,18:3,23:1,30:1,36:1,99:2,102:6,146:1,181:2,183:1,234:1,299:1,430:1,436:1,535:1,539:2,667:2,753:1,813:5,854:1,917:1,920:1,922:1,969:5,978:2,996:1,998:1,1001:4,1096:1,11...
232	12:1,13:1,18:1,69:2,146:1,200:1,234:2,329:1,370:2,565:2,571:2,605:1,608:2,667:7,813:3,891:6,1008:5,1065:1,1096:1,1104:1,1189:5,1190:2,1293:1,1572:1,1636:1,1816:1,2117:1,21...
235	12:2,13:2,18:1,88:1,204:1,478:1,523:1,558:1,575:1,606:1,667:2,670:1,754:2,803:1,872:1,921:1,1119:1,1398:2,1421:1,1498:1,1704:1,1947:1,2109:2,2132:1,2352:1,2783:3,3019:1,30...
238	10:3,202:2,539:1,667:1,892:1,1096:3,1127:1,1584:1,1806:2,2109:1,2122:1,2143:1,3024:1,3054:2,3364:1,3701:2,3765:1,3879:1,3984:1,5500:1,5685:1,6116:1,6529:1,6832:1,7460:1,...
240	10:1,107:1,115:1,146:1,412:1,430:1,450:2,596:1,667:1,800:1,931:1,1478:1,1584:1,1604:1,1652:2,1848:1,2352:1,2641:1,2676:1,2783:1,3000:2,3019:1,3054:2,3078:1,3577:1,3901:1,...

- **append_id** 是每篇新闻的唯一标识。
- **key_value** 字段中冒号前面的数字表示的是单词抽象成的数字标识，冒号后面是对应的单词出现的频率。

数据进入PLDA算法。

PLDA算法又叫主题模型，算法可以定位代表每篇文章的主题的词语。本次试验设置了50个主题，PLDA有六个输出桩，第五个输出桩输出结果展示的是每篇文章对应的每个主题的概率，如下图所

示。

docid	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9	topic_10	topic_11	topi
0	0.0015625	0.0015625	0.0015625	0.0171875	0.0015625	0.0484375	0.0015625	0.0015625	0.0015625	0.0015625	0.0015625	0.0328125	0.0(
1	0.001298...	0.014285...	0.001298...	0.014285...	0.001298...	0.001298...	0.014285...	0.001298...	0.001298...	0.014285...	0.1831168...	0.0012987...	0.0(
2	0.011224...	0.021428...	0.001020...	0.011224...	0.011224...	0.001020...	0.001020...	0.001020...	0.001020...	0.011224...	0.0010204...	0.0214285...	0.0(
3	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.0716814...	0.0008849...	0.0(
4	0.039285...	0.003571...	0.003571...	0.289285...	0.003571...	0.003571...	0.003571...	0.003571...	0.003571...	0.039285...	0.0035714...	0.075	0.0
5	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.043661...	0.0295774...	0.0014084...	0.1(
6	0.002736...	0.010199...	0.010199...	0.000248...	0.000248...	0.040049...	0.000248...	0.000248...	0.000248...	0.000248...	0.0201492...	0.0002487...	0.0(
7	0.000543	0.000543	0.000543	0.000543	0.000543	0.027717	0.000543	0.000543	0.000543	0.000543	0.0548913	0.0059782	0.0(

5. 结果分析和评估

上面的步骤将文章从主题的维度表示成了一个向量。

下面就可以通过向量的距离实现聚类，从而实现文章分类。K均值聚类组件的分类结果如下图所示。

docid	cluster_index
115	0
292	0
248	0
166	0
116	2
210	3
8	4
15	4

- **cluster_index** 表示的是每一类的名称。
- 找到第0类，一共有 **docid** 为115，292，248，166四篇文章。

通过过滤与映射组件查询115，292，248，166四篇文章。结果如下图所示。

append_id	category	title	content
115	体育	“欧洲通...	来源：重庆晚报“欧洲通行证”考验门将每次大赛，新推出的用球都会成为球员和市场关注的焦点，此次欧洲杯的用球“欧洲通行证”估计也会让门将们大伤脑...
166	财经	新旗舰...	机构：周四上证指数快速击穿新低进一步摧毁了市场在３０００点一带进行抵抗的信心，大盘如同自由落体，直至２９００点附近才出现抵抗，最终当天再...
248	体育	图文：...	来源：体育体育讯　北京时间６月１５号凌晨，０８欧洲杯D组第二轮开战，在奥地利因斯布鲁克的蒂沃利球场，西班牙２比１险胜瑞典，斗牛士军团以６...
292	科技	ＬＧ第...	赛迪网讯６月３０日消息，据台湾媒体报道，随着第二季度摩托罗拉在全球的手机市场的表现持续低迷，ＬＧ电子第二季度手机出货量有望突破３，０００...

实验效果并不十分理想，上图中将一篇财经、一篇科技的新闻跟两个体育类新闻分到了一起。

主要原因如下：

- 没有进行细节的调优。
- 没有对数据进行特征工程处理。
- 数据量太小。

离线调度说明

背景

本文实现的是广告CTR预测的场景。广告CTR预测是广告行业的典型应用，通过历史数据训练预测模型，对于每天的增量数据进行预测，找出广告的CTR符合标准的样本进行投放。
整套实验使用了阿里云机器学习进行数据挖掘，通过大数据开发套件进行调度和推送。具体的业务场景是：

- 通过历史数据在阿里云机器学习平台上进行模型训练。
- 通过大数据开发套件对模型进行调度。
- 每天凌晨对广告投放进行CTR预测，甄选出符合标准的广告进行推送。

数据集介绍

具体字段如下表。

字段名	含义	类型	描述
id	ID	string	广告的唯一标识
age	年龄	double	广告投放人群的年龄
sex	性别	double	广告投放人群的性别，1代表男，0代表女
duration	时长	double	广告在界面的停留时长，以秒为单位
place	位置	double	广告投放位置，0～4，按照投放位置从上到下的顺序排列
ctr	广告CTR	double	广告点击量除以展现量，大于0.03是1，其它

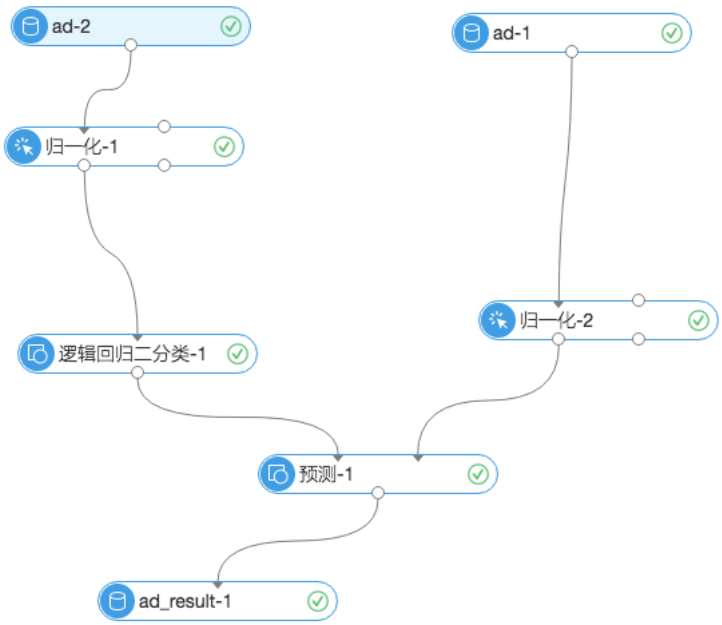
			是0
dt	partition	string	年月日，格式为yyyyMMdd

如下图所示，数据是通过random算法随机生成，所以本次实验不针对结果进行评估，主要介绍实验搭建以及和大数据开发套件的调度使用。数据包含20160919、20160920的历史数据，需要针对20160921的数据预测。使用的是MaxCompute的分区表。

id ▲	age ▲	sex ▲	duration ▲	place ▲	ctr ▲	dt ▲
0	49	1	9	0	0	20160919
1	17	1	3	1	1	20160919
2	44	0	4	0	0	20160919
3	14	1	9	1	0	20160919
4	44	1	5	4	0	20160919
5	10	1	9	3	1	20160919
6	42	1	7	3	0	20160919
7	51	1	3	1	1	20160919
8	18	0	3	3	0	20160919
9	39	0	8	4	1	20160919
10	45	1	3	2	0	20160919
11	57	0	8	2	0	20160919
12	14	0	7	2	1	20160919

实验搭建

实验流程图如下。



实验可以大致分为四个模块，数据源导入（ad），数据预处理（归一化），模型训练（逻辑回归二分类），预测（预测）。

1. 数据源导入

- “ad-2”是训练数据源。
- “ad-1”是预测数据源。
- 通过配置分区表的partition dt=@@{yyyyMMdd}，确定预测数据是每日的增量数据，如下图所示。（分区使用详情请参见 https://help.aliyun.com/document_detail/30281.html?spm=5176.doc30276.6.126.3kX7OU ）

2. 中间过程

中间过程包括数据归一化和模型训练两个步骤。模型训练是通过历史数据训练生成的预测模型。（详细原理可以参考心脏病预测案例）

3. 预测

预测生成的结果表为“ad_result-1”，数据如下图所示。

id	prediction_result	prediction_score	prediction_detail
400	0	0.5090281750932395	{"0": 0.5090281750932395, "1": 0.4909718249067604}
401	0	0.5185830406571692	{"0": 0.5185830406571692, "1": 0.4814169593428308}
402	0	0.5037390968394624	{"0": 0.5037390968394624, "1": 0.4962609031605377}
403	1	0.5136006398483877	{"0": 0.4863993601516123, "1": 0.5136006398483877}
404	0	0.5032116074286588	{"0": 0.5032116074286588, "1": 0.4967883925713412}
405	0	0.5170683273721821	{"0": 0.5170683273721821, "1": 0.4829316726278179}
406	1	0.5561919238468677	{"0": 0.4438080761531323, "1": 0.5561919238468677}
407	0	0.51090881729545	{"0": 0.51090881729545, "1": 0.48909118270455}

- prediction_result：每个广告id是否被点击。1表示被点击，0表示不被点击。
- prediction_score：对应被点击概率。

模块调度

1. 进入大数据开发套件工作空间

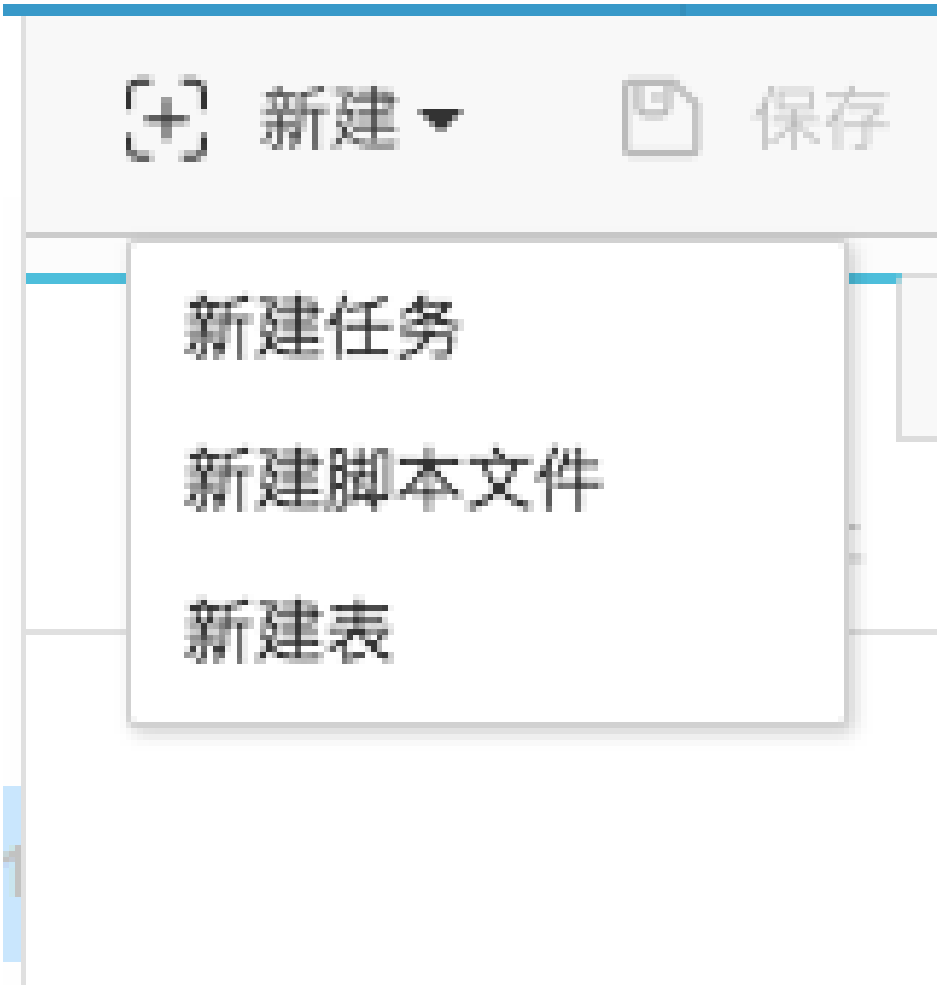
进入控制台首页，单击**DataWorks**，进入大数据开发工作空间。


大数据（数加）
数加控制台概览
DataWorks
Quick BI
机器学习
推荐引擎
公众趋势分析
DataV数据可视化
大数据计算服务

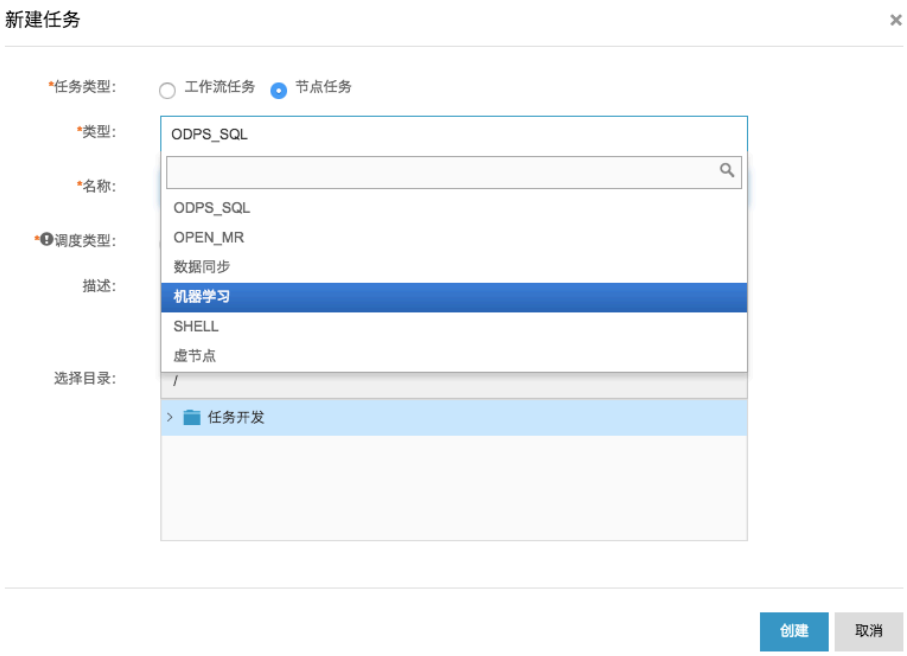
大数据开发套件与机器学习平台共用一套项目，选择需要调度的实验所在的项目，单击**进入数据开发**。

2. 新建节点调度任务

单击**新建**并选择**新建任务**。



在新建任务的配置中，**任务类型**选择**节点任务**，**类型**选择**机器学习**。



3. 配置调度任务

建立了节点任务之后，选择需要调度的机器学习实验，并在右边的配置栏选择需要调度的时间，本实验选择每日的凌晨0点进行训练和推送信息。

单击**提交**。提交的作业从**第二天**开始生效。

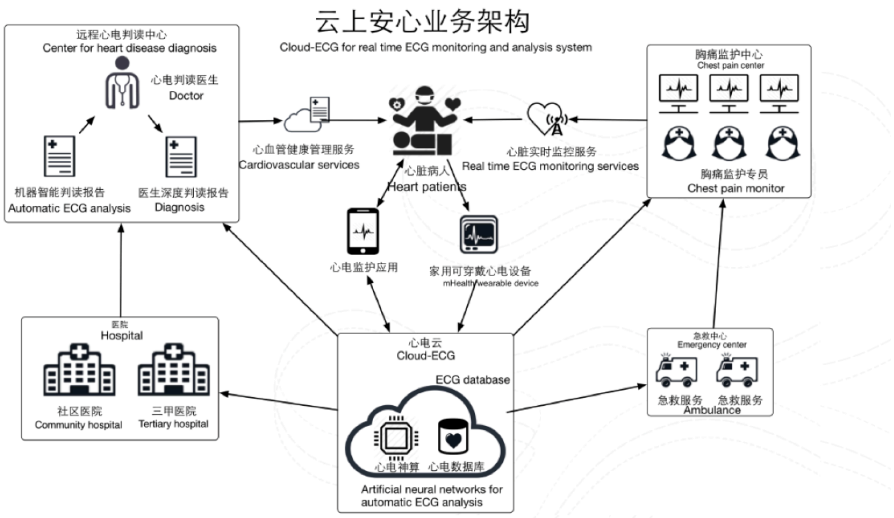
4. 查询任务日志

提交调度任务之后，单击**前往运维**查看日志。

模型在线预测

背景

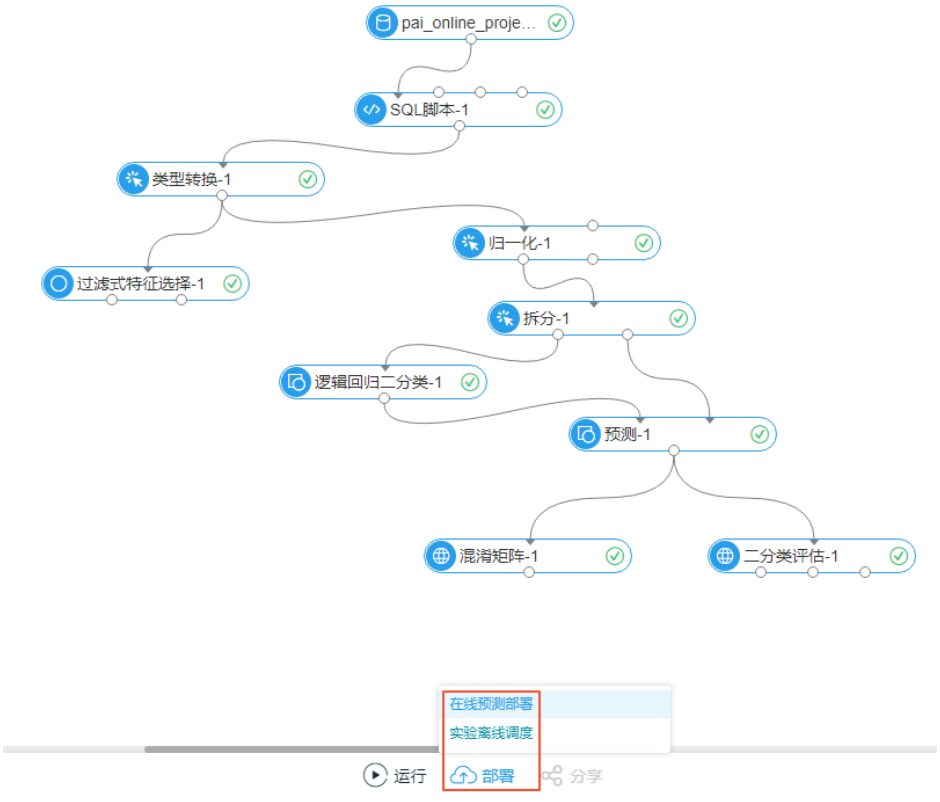
机器学习平台生成的模型可以通过在线部署方式生成API供其它业务调用。本文档基于心脏病预测案例，介绍通过机器学习平台的在线预测部署功能，实时监测用户健康情况的方法。



步骤

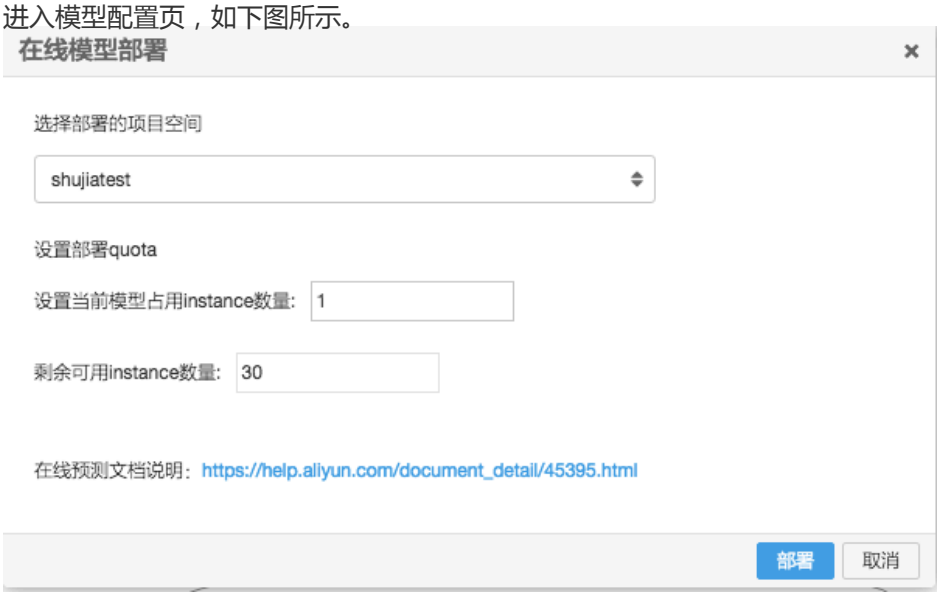
1. 模型部署

在当前实验界面下方单击**部署**，选择**在线预测部署**。并选择心脏病预测案例中生成的逻辑回归模型，如下图所示。



2. 模型部署信息配置

进入模型配置页，如下图所示。

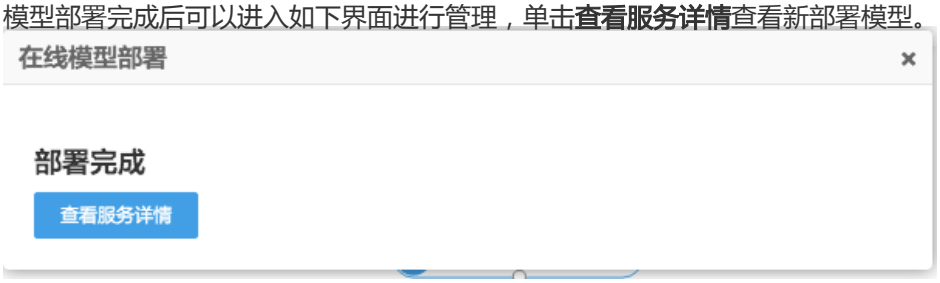


选择对应的项目空间，如果是第一次使用需要开通在线预测权限，权限申请是实时开通。设置当前模型占用的instance数量，instance定义如下。

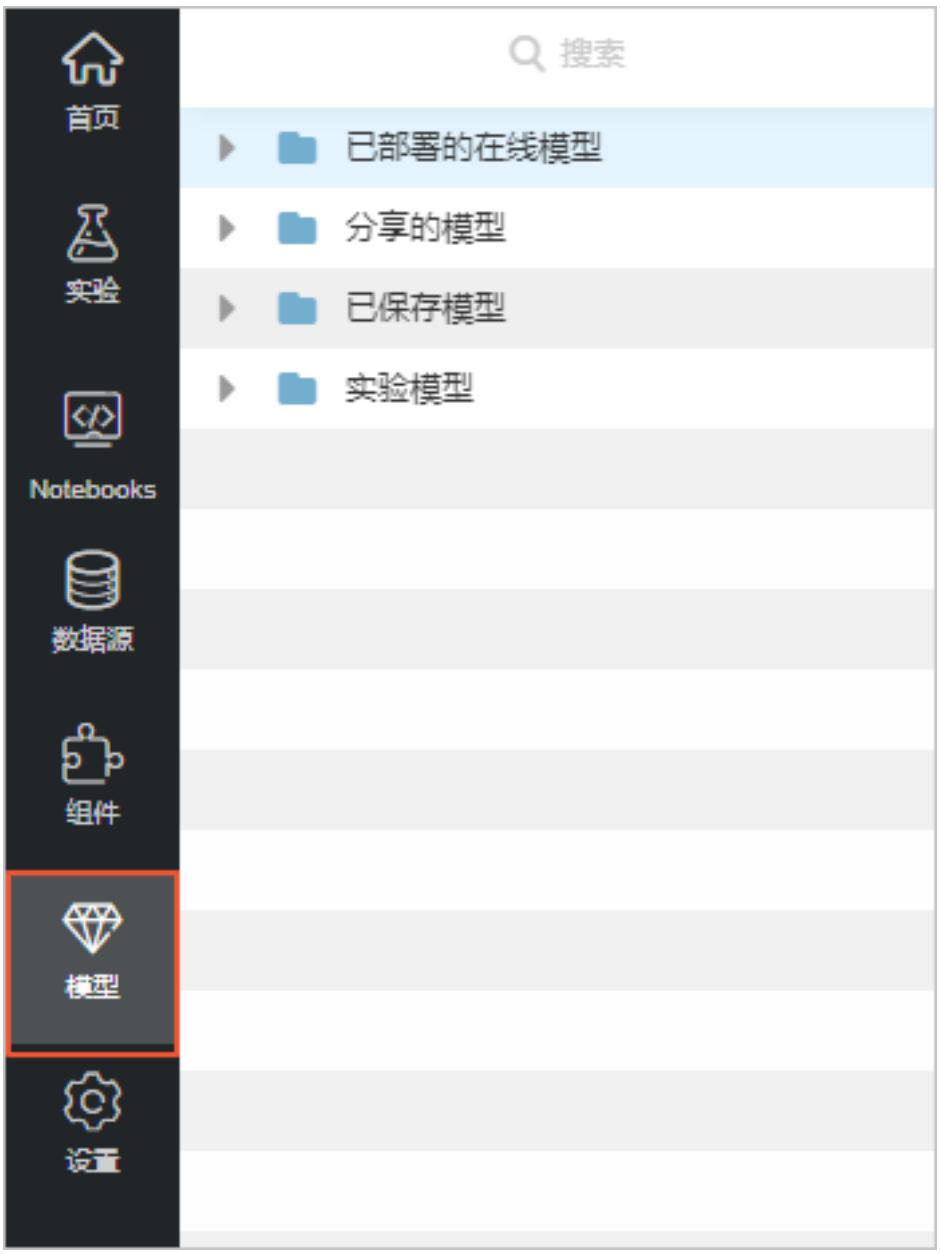
- 每个项目默认包含30个instance，可提工单扩容。删除已部署模型会释放当前模型的instance。
- instance决定模型的QPS，每个instance为1核2GB内存。
- 单个模型的instance部署限制是[1,15]。

3. 模型管控

模型部署完成后可以进入如下界面进行管理，单击**查看服务详情**查看新部署模型。



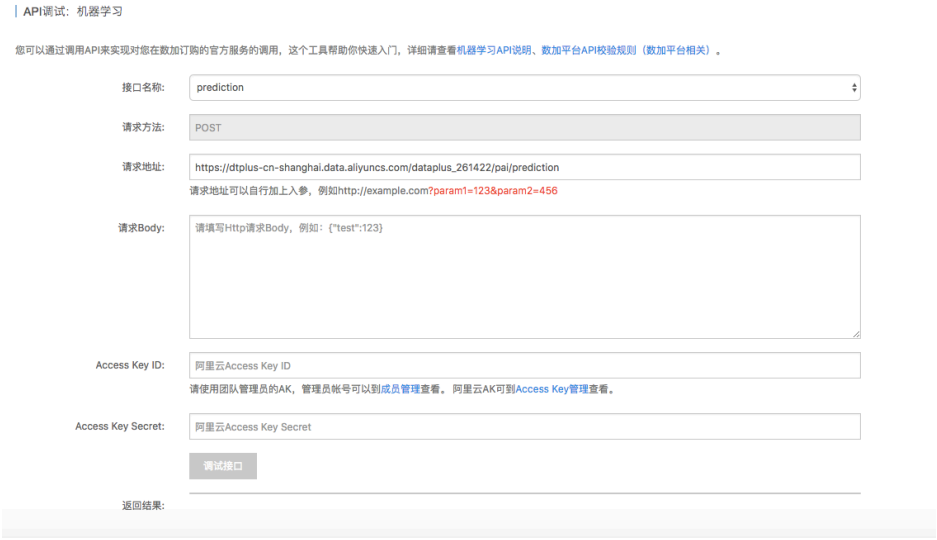
已经部署的模型可以在**已部署的在线模型**中进行管理，如下图所示。



模型管理界面，版本表示的是同一模型多次部署的区分，通过下图可以得到模型所在的项目和模型名称。

4. 模型调试

模型调试页面可以帮助用户了解在线预测请求参数的书写规范，单击上图中的**模型调试**，进入模型调试页面（浏览器可能会阻止页面该页面，请设置浏览器允许访问该链接）。



- **请求地址**的格式为“https://dtplus-cn-shanghai.data.aliyuncs.com/dataplus_261422/pai/prediction/projects/$project名称/onlinemodels/$模型名称 ”。
- **请求body**的格式为json串，以逻辑回归算法为例，需要填写每个特征的信息，特征名字需要与模型表

特征名对应，常数列不填。

- dataValue：预测集对应特征的取值。
- dataType：数值类型，定义如下图所示。

数据类型	dataType
bool	1
int32	10
int64	20
float	30
double	40
string	50

5. 结果预测

完成模型调试配置后，编辑**请求body**部分并发送请求即可获得预测结果。假设用户的性别、实时血压、实时心跳波动等参数都是1，向服务器推送以下数据。
本案例body范例如下。

```
{
"inputs": [
{
"sex": {
"dataType": 40,
"dataValue": 1
},
"cp": {
"dataType": 40,
"dataValue": 1
},
"fbs": {
"dataType": 40,
"dataValue": 1
},
"restecg": {
"dataType": 40,
"dataValue": 1
},
"exang": {
"dataType": 40,
"dataValue": 1
},
"slop": {
"dataType": 40,
"dataValue": 1
},
"thal": {
"dataType": 40,
"dataValue": 1
},
"age": {
```

```
"dataType": 40,
"dataValue": 1
},
"trestbps": {
"dataType": 40,
"dataValue": 1
},
"chol": {
"dataType": 40,
"dataValue": 1
},
"thalach": {
"dataType": 40,
"dataValue": 1
}
}
]
}
```

发送请求后可以获得返回结果，显示label为1（1表示患病，0表示健康），患病概率为0.98649974。

```
- - - - - 请求 - - - - -
- - - - - 返回 - - - - -
状态码: 200
返回Body: {
  "outputs": [
    {
      "outputLabel": "1",
      "outputMulti": {
        "0": 0.01351125016100008,
        "1": 0.9864887498389999
      },
      "outputValue": {
        "dataType": 40,
        "dataValue": 0.9864887498389999
      }
    }
  ]
}
- - - - - 返回 - - - - -
```

API调用方法请参考https://help.aliyun.com/document_detail/30245.html。

协同过滤做商品推荐

本文数据为虚构，仅供实验。

背景

数据挖掘的一个经典案例就是尿布与啤酒的例子。尿布与啤酒看似毫不相关的两种产品，但是当超市将两种产品放到相邻货架销售的时候，会大大提高两者销量。很多时候看似不相关的两种产品，却会存在这某种神秘的隐含关系，获取这种关系将会对提高销售额起到推动作用，然而有时这种关联是很难通过理性的分析得到的。这时候我们需要借助数据挖掘中的常见算法-协同过滤来实现。这种算法可以帮助我们挖掘人与人以及商品与商品的关联关系。

协同过滤算法是一种基于关联规则的算法，以购物行为为例。假设有甲和乙两名用户，有a、b、c三款产品。如果甲和乙都购买了a和b这两种产品，我们可以假定甲和乙有近似的购物品味。当甲购买了产品c而乙还没有购买c的时候，我们就可以把c也推荐给乙。这是一种典型的user-based情况，就是以user的特性做为一种关联。

本文的业务场景如下：
通过一份7月份前的用户购物行为数据，获取商品的关联关系，对用户7月份之后的购买形成推荐，并评估结果。比如用户甲某在7月份之前买了商品A，商品A与B强相关，我们就在7月份之后推荐了商品B，并探查这次推荐是否命中。

数据集介绍

本文档数据源为天池大赛提供数据，数据按时间分为两份，分别是7月份之前和7月份之后的购买行为数据。

具体字段如下表。

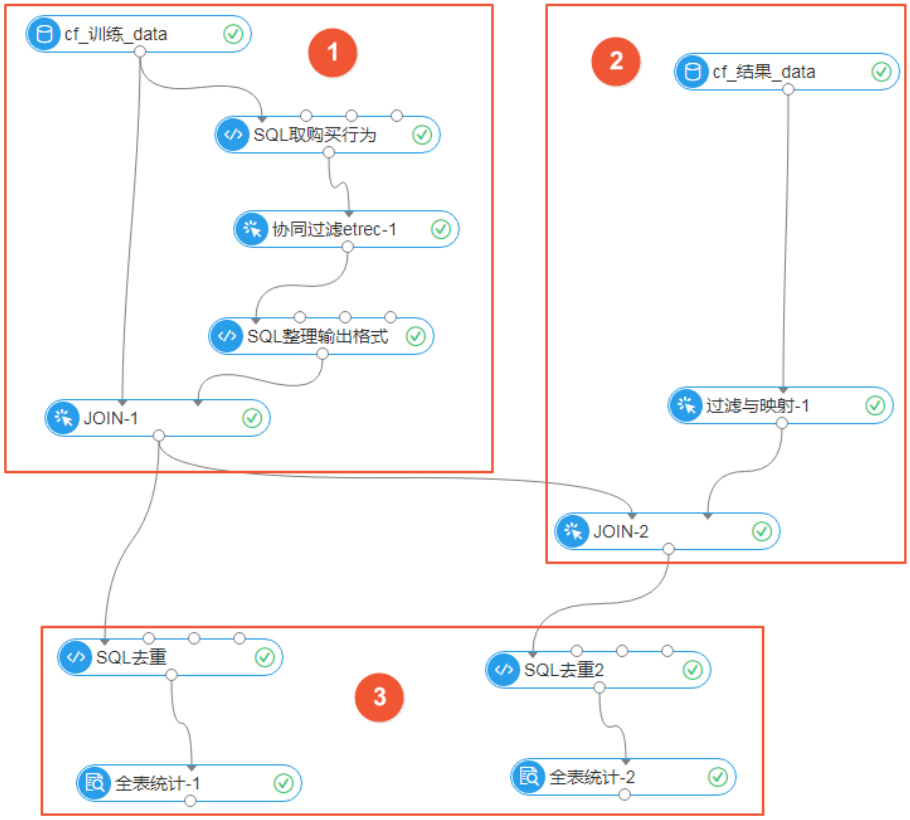
字段名	含义	类型	描述
user_id	用户编号	string	购物的用户ID
item_id	物品编号	string	被购买物品的编号
active_type	购物行为	string	0表示点击，1表示购买，2表示收藏，3表示购物车
active_date	购物时间	string	购物发生的时间

数据截图如下。

10944750	8689	2	5月2日
10944750	25687	2	5月8日
10944750	7150	1	6月7日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日

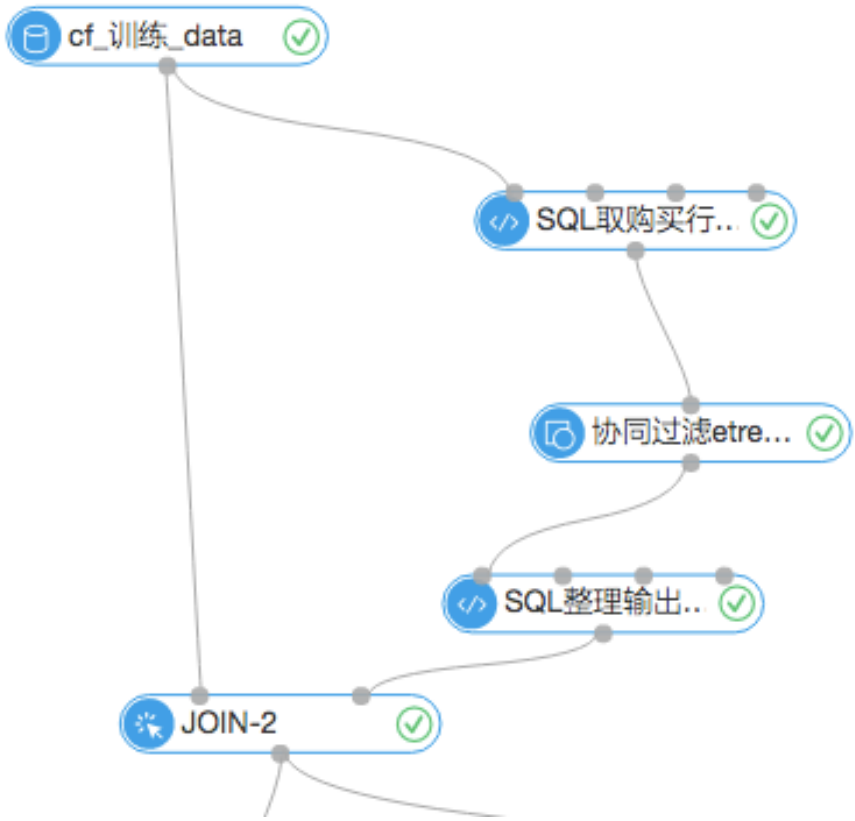
数据探索流程

实验流程图如下。



- 1：根据关联规则生成推荐列表
- 2：七月之后的真实购物行为
- 3：推荐数和命中数的统计

1. 生成推荐列表



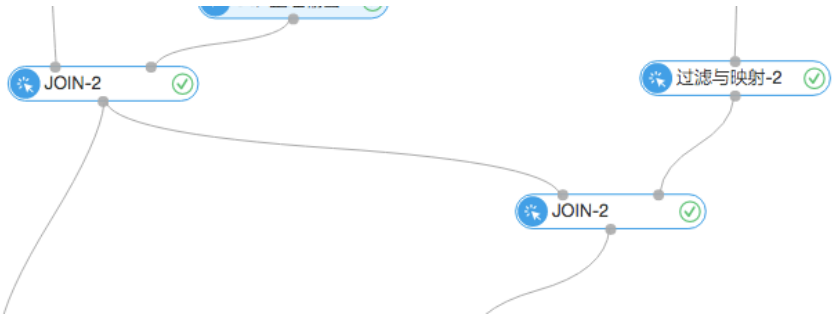
输入的数据源是7月份之前的购物行为数据，通过SQL脚本取出用户的购买行为数据，进入协同过滤组件。协同过滤组件设置中把TopN设置成1，表示每个item返回最相近的item和它的权重。通过购买行为，分析出哪些商品被同一个user购买的可能性最大，如下图所示。

协同过滤结果表示的是商品的关联性，“itemid”表示目标商品，“similarity”字段冒号的左侧表示与目标关联性高的商品，右侧表示两个商品的关联性概率。

itemid ▲	similarity ▲
1000	15584:0.2747133918
10014	18712:0.05229603127
10066	3228:0.2650900672
1008	24507:1
10082	18024:0.1781525919
1010	18024:0.2104947227
10133	14020:0.2070609237
1015	18024:0.2104947227
10151	26288:0.4366713611
10171	11080:0.2401992435

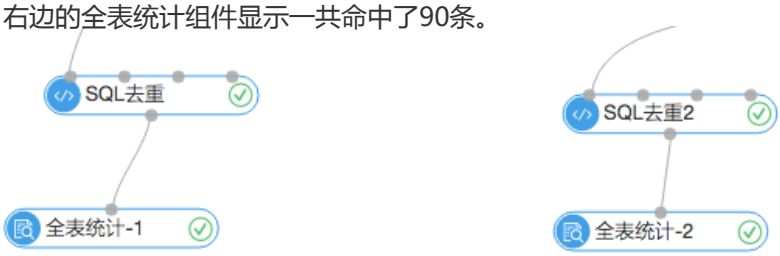
2. 推荐

步骤一介绍了如何生成强关联商品的对应列表。这里使用了比较简单的推荐规则，比如用户甲在7月份之前买了商品A，商品A与B强相关，我们就在7月份之后推荐了商品B，并探查这次推荐是否命中，实验流程如下图所示。



3. 结果统计

下图是统计模块，左边的全表统计组件展示的是根据7月份之前的购物行为生成的推荐列表，去重后共18065条。右边的全表统计组件显示一共命中了90条。



推荐系统反思

根据上文的统计结果可以看出，本次试验的推荐效果并不理想，原因如下。

- 本文档只是针对了业务场景大致介绍了协同过滤推荐的用法。很多针对于购物行为推荐的关键点都没有处理，比如说时间序列。购物行为一定要注意时效性的分析，跨度达到几个月的推荐不会有好的效果。
- 本文档只考虑了商品的关联性，没有考虑推荐商品的属性，例如是高频还是低频商品。比如用户A上个月买了个手机，那下个月就不大会继续购买手机，因为手机是低频消费品。
- 基于关联规则的推荐方法最好是作为补充，真正想提高准确率还是要依靠机器学习算法训练模型的方式。

其它

请进入阿里云数加机器学习平台体验阿里云机器学习产品，并通过云栖社区公众号参与讨论。

人口普查统计案例

背景

本文档场景如下：
通过一份人口普查数据，根据人物的年龄、工作类型、教育程度等属性，统计学历对收入的影响。主要目的是帮助用户学习阿里云机器学习实验的搭建流程和组件的使用方式。

数据集介绍

数据源： UCI开源数据集Adult 是针对美国某区域的一次人口普查结果，共32561条数据。具体字段如下表。

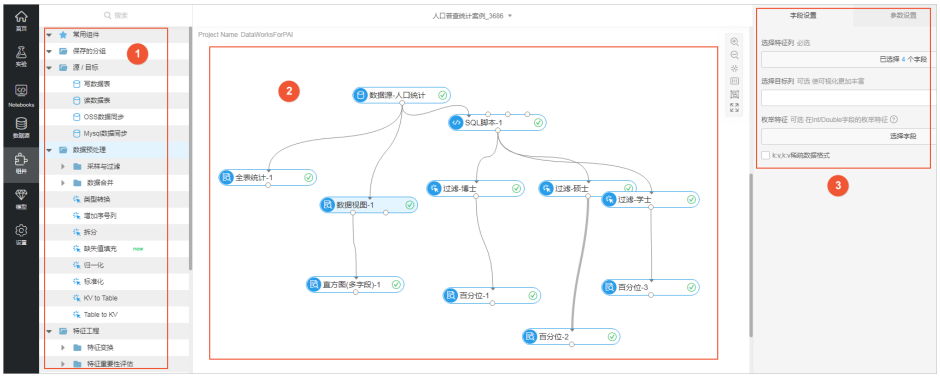
字段名	含义	类型

age	年龄	double
workclass	工作类型	string
fnlwgt	序号	string
education	教育程度	string
education_num	受教育时间	double
maritial_status	婚姻状况	string
occupation	职业	string
relationship	关系	string
race	种族	string
sex	性别	string
capital_gain	资本收益	string
capital_loss	资本损失	string
hours_per_week	每周工作小时数	double
native_country	原籍	string
income	收入	string

数据探索流程

在机器学习控制台首页，选择人口普查统计案例，单击**从模板创建**，如下图所示。

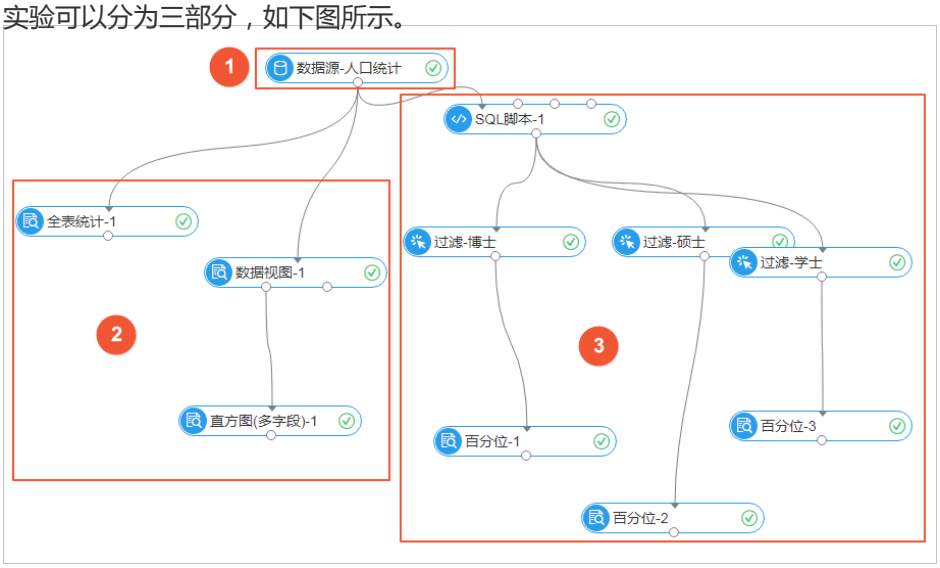
实验界面如下图所示。



- 图中第一部分为组件区域。用户可以将其拖拽到中间的空白区域搭建实验。

- 图中第二部分为实验区域。用户可以在此区域搭建实验。
- 图中第三部分为组件配置区域。用户可以在此区域配置组件参数。

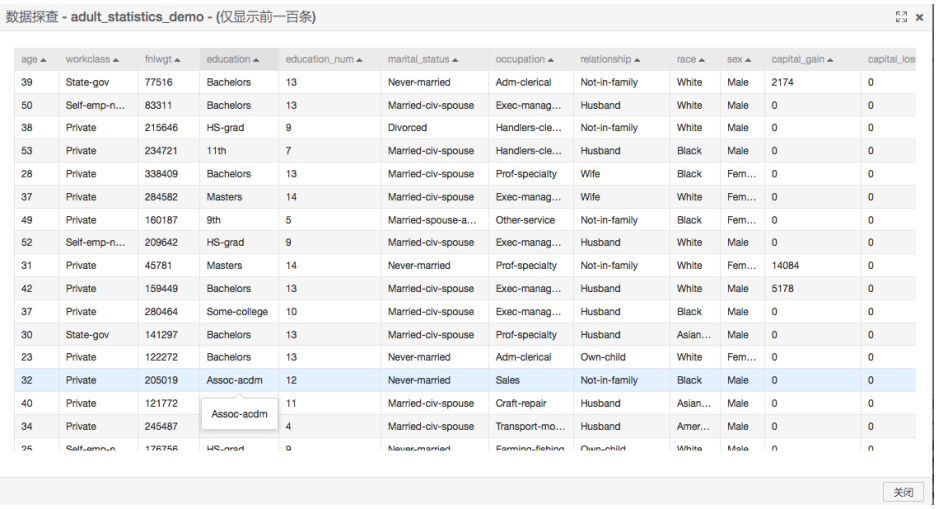
实验可以分为三部分，如下图所示。



第一部分完成数据源准备，第二部分完成数据统计，第三部分完成学历对收入的影响统计。

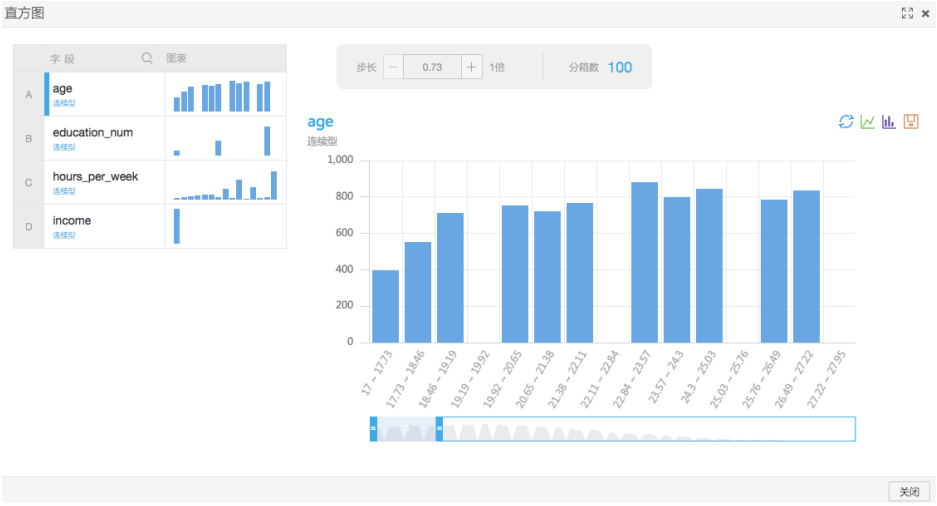
1. 数据源准备

通过机器学习IDE或者tunnel命令行工具，将数据上传到MaxCompute上。通过**读数据表**组件（图中的数据源-人口统计）读取数据。完成后右键单击组件查看数据，如下图所示。



2. 数据统计

通过全表统计和数值分布统计结果（实验中的数据视图和直方图组件）可以判断一份数据是符合泊松分布还是高斯分布、是连续还是离散。
阿里云机器学习的每个组件都提供了可视化显示结果的功能，下图是数值统计的直方图组件的输出结果，可以清楚地看到每个输入数据的分布情况。

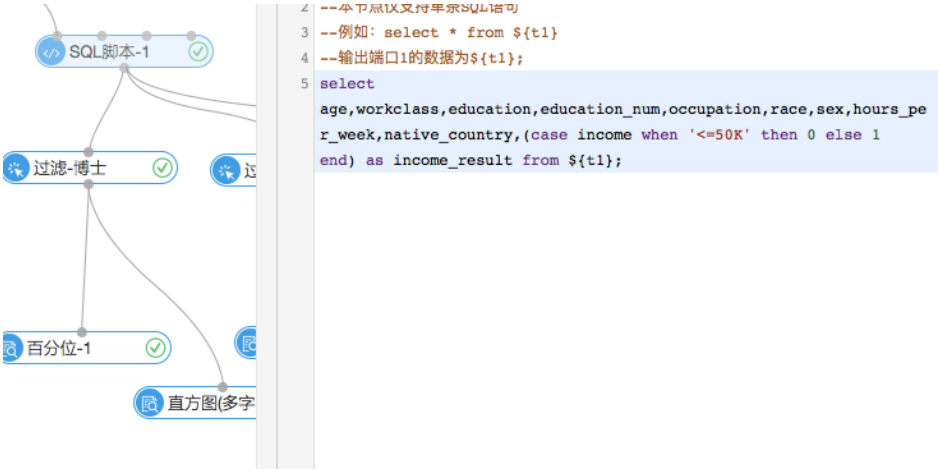


3. 学历对收入的影响统计

通过特征提取，使用机器学习算法计算得到哪些因素对收入的影响最大。本文档仅简单地针对不同学历人员的收入做统计，主要目的是介绍机器学习平台的使用方法。

数据预处理

如下图所示，数据流入的第一个组件是**SQL脚本**组件，实现数据预处理的功能。本实验是将string类型的“income”字段转换成二值型的0和1的形式。0表示年收入在50K以下，1表示年收入在50K以上（这种将文本数据数值化是机器学习特征处理的常用方式）。

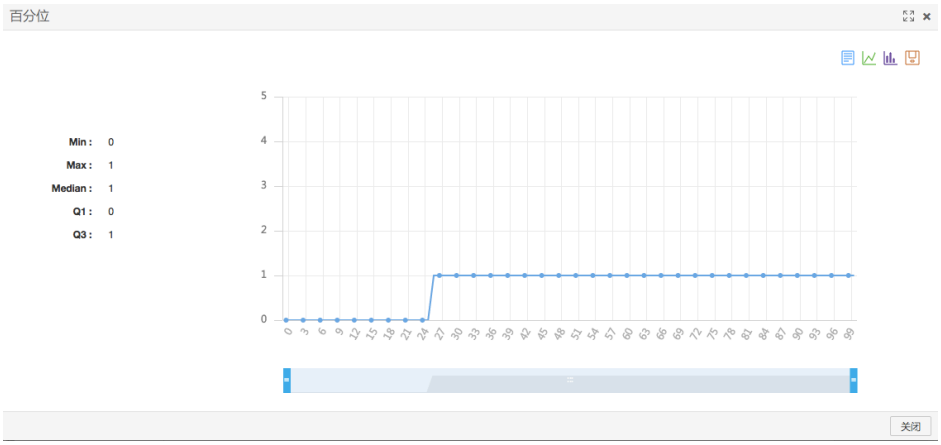


过滤与映射

通过**过滤与映射**组件将数据按照学历分为三部分，分别是博士、硕士和学士，如下图所示。
过滤与映射组件支持SQL语句，需要用户在右侧的配置栏填写where过滤条件。

结果统计

通过**百分位**组件可以得到每个分类下的收入比例。下图是折线图的展示效果，可以看到年收入在50K以下（结果中为0的点）的人群占总人数的百分之25左右。



结合三个百分位组件就可以得到如下图所示的结果。

学历	年收入大于50K的比例
博士	75%
硕士	57%
学士	42%

其它

请进入**阿里云数加机器学习平台**体验阿里云机器学习产品，并通过**云栖社区公众号**参与讨论。

学生考试成绩预测

本文数据为虚构，仅供实验。

背景

本文档通过真实的中学生数据和机器挖掘算法得到影响中学生学业的关键因素。比如父母的职业、父母的学历、家庭能否上网等。
本文档的数据采集于某中学在校生的家庭背景以及在校行为。通过逻辑回归算法生成离线模型和学业指标评估报告，对学生的期末成绩进行预测。同时生成在线预测API，通过API把训练好的离线模型应用到在线的业务场景中。

数据集介绍

数据集由25个特征列和一个目标列构成，具体字段如下表。

字段名	含义	类型	描述
sex	性别	string	F表示女，M表示男
address	住址	string	U表示城市，R表示乡村
famsize	家庭成员数	string	LE3表示少于三人，GT3表示多于三人
pstatus	是否与父母住在一起	string	T表示住在一起，A表示分开
medu	母亲的文化水平	string	从0～4逐步增高
fedu	父亲的文化水平	string	从0～4逐步增高
mjob	母亲的工作	string	分为教师相关、健康相关、服务业
fjob	父亲的工作	string	分为教师相关、健康相关、服务业
guardian	学生的监管人	string	mother、father、other
traveltime	从家到学校需要的时间	double	以分钟为单位
studytime	每周学习时间	double	以小时为单位
failures	挂科数	double	挂科次数
schoolsup	是否有额外的学习辅助	string	yes、no
fumsup	是否有家教	string	yes、no
paid	是否有相关考试学科的辅助	string	yes、no
activities	是否有课外兴趣班	string	yes、no

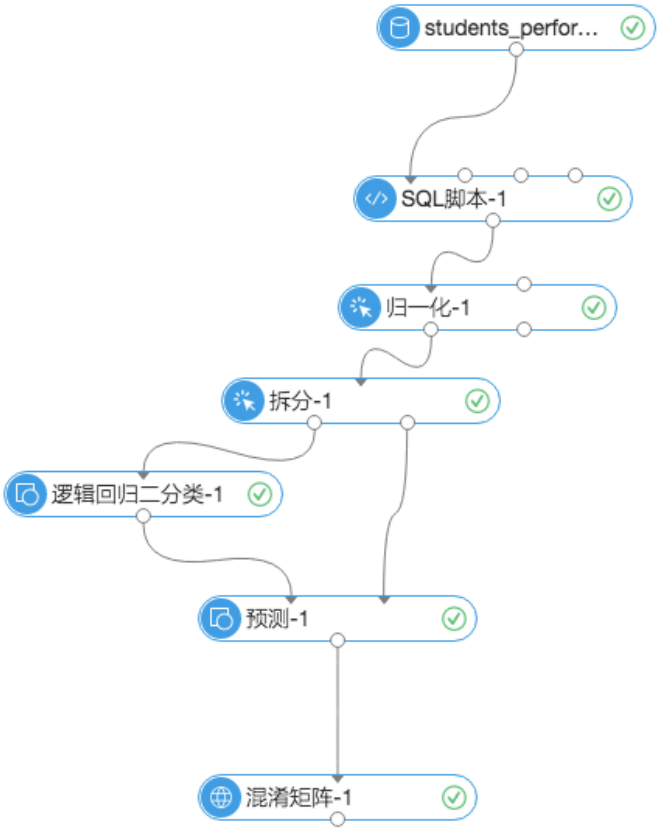
higher	是否有向上求学意愿	string	yes、no
internet	家里是否联网	string	yes、no
famrel	家庭关系	double	从1～5表示关系从差到好
freetime	课余时间量	double	从1～5从少到多
goout	跟朋友出去玩的频率	double	从1～5从少到多
dalc	日饮酒量	double	从1～5从少到多
walc	周饮酒量	double	从1～5从少到多
health	健康状况	double	从1～5表示状态从差到好
absences	出勤量	double	0到93次
g3	期末成绩	double	20分制

数据截图如下。

sex	address	famsize	pstatus	medu	fedu	mjob	fjob	guardian	traveltime	studytime	failures	schoolsup	famsup
F	U	GT3	A	4	4	at_ho...	teacher	mother	2	2	0	yes	no
F	U	GT3	T	1	1	at_ho...	other	father	1	2	0	no	yes
F	U	LE3	T	1	1	at_ho...	other	mother	1	2	3	yes	no
F	U	GT3	T	4	2	health	services	mother	1	3	0	no	yes
F	U	GT3	T	3	3	other	other	father	1	2	0	no	yes
M	U	LE3	T	4	3	services	other	mother	1	2	0	no	yes
M	U	LE3	T	2	2	other	other	mother	1	2	0	no	no
F	U	GT3	A	4	4	other	teacher	mother	2	2	0	yes	yes
M	U	LE3	A	3	2	services	other	mother	1	2	0	no	yes
M	U	GT3	T	3	4	other	other	mother	1	2	0	no	yes
F	U	GT3	T	4	4	teacher	health	mother	1	2	0	no	yes

离线训练

实验流程图如下。



数据自上到下流入实验，依次完成了数据预处理、拆分、训练、预测与评估。

1. 数据预处理

SQL脚本如下。

```
select (case sex when 'F' then 1 else 0 end) as sex,
(case address when 'U' then 1 else 0 end) as address,
(case famsize when 'LE3' then 1 else 0 end) as famsize,
(case Pstatus when 'T' then 1 else 0 end) as Pstatus,
Medu,
Fedu,
(case Mjob when 'teacher' then 1 else 0 end) as Mjob,
(case Fjob when 'teacher' then 1 else 0 end) as Fjob,
(case guardian when 'mother' then 0 when 'father' then 1 else 2 end) as guardian,
traveltime,
studytime,
failures,
(case schoolsup when 'yes' then 1 else 0 end) as schoolsup,
(case fumsup when 'yes' then 1 else 0 end) as fumsup,
(case paid when 'yes' then 1 else 0 end) as paid,
(case activities when 'yes' then 1 else 0 end) as activities,
(case higher when 'yes' then 1 else 0 end) as higher,
(case internet when 'yes' then 1 else 0 end) as internet,
famrel,
freetime,
```

```
goout,
Dalc,
Walc,
health,
absences,
(case when G3>14 then 1 else 0 end) as finalScore
from ${t1};
```

使用**SQL脚本**组件将文本数据结构化。

- 比如源数据分别有yes和no的情况，可以通过0表示yes，1表示no，将文本数据量化。
- 对于一些多种类的文本型字段，可以结合业务场景将数据抽象化。比如“Mjob”字段，是teacher表示为1，不是teacher表示为0。抽象后这个特征的意义就是表示工作是否与教育相关。
- 对于目标列，按照大于18分设为1，其它为0的方式进行量化。目的是通过训练，找出可以预测分数的模型。

2. 归一化

归一化组件的作用是去除量纲，将所有的字段都变换到0～1之间，去除字段间大小不均衡带来的影响，结果如下图所示。

sex	address	famsize	pstatus	medu	fedu	mjob	fjob	guardian	traveltime	studytime	failures	schoolsup	famsup
1	1	0	0	1	1	0	1	0	0.333333333...	0.33333333...	0	1	0
1	1	0	1	0.25	0.25	0	0	0.5	0	0.33333333...	0	0	1
1	1	1	1	0.25	0.25	0	0	0	0	0.33333333...	1	1	0
1	1	0	1	1	0.5	0	0	0	0	0.66666666...	0	0	1
1	1	0	1	0.75	0.75	0	0	0.5	0	0.33333333...	0	0	1
0	1	1	1	1	0.75	0	0	0	0	0.33333333...	0	0	1
0	1	1	1	0.5	0.5	0	0	0	0	0.33333333...	0	0	0
1	1	0	0	1	1	0	1	0	0.333333333...	0.33333333...	0	1	1
0	1	1	0	0.75	0.5	0	0	0	0	0.33333333...	0	0	1
0	1	0	1	0.75	1	0	0	0	0	0.33333333...	0	0	1
1	1	0	1	1	1	1	0	0	0	0.33333333...	0	0	1
1	1	0	1	0.5	0.25	0	0	0.5	0.666666666...	0.66666666...	0	0	1
0	1	1	1	1	1	0	0	0.5	0	0	0	0	1
0	1	0	1	1	0.75	1	0	0	0.333333333...	0.33333333...	0	0	1

3. 拆分

将数据集按照8：2的比例拆分，百分之八十用来训练模型，百分之二十用来预测。

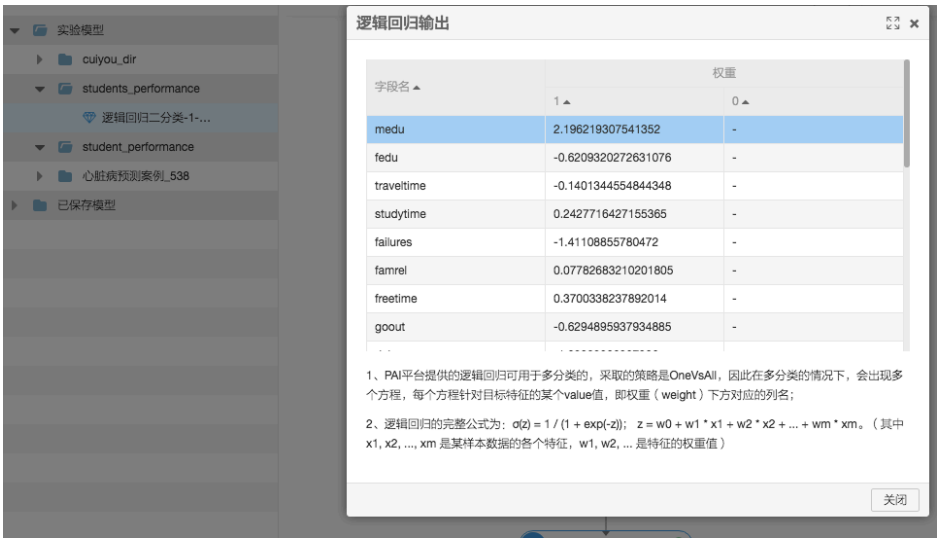
4. 逻辑回归

通过逻辑回归算法训练生成离线模型。算法详情请参见wiki。

5. 结果分析与评估

通过混淆矩阵查看模型预测的准确率。从下图中可以看到本实验的预测准确率为82.911%。

根据逻辑回归算法的特性，可以通过模型系数挖掘出一些有价值的信息。右键单击**逻辑回归二分类**组件查看模型，结果如下图所示。



根据逻辑回归算法的算法特性，权重越大表示特征对于结果的影响越大。权重为正数表示对结果1（期末高分）正相关，权重负数表示负相关。下表对几个权重较大的特征进行了分析。

字段名	含义	权重	分析
mjob	母亲的工作	-0.7998341777833717	母亲是老师对于孩子考高分是不利的。
fjob	父亲工作	1.422595764037065	如果父亲是老师，对于孩子取得好的成绩是非常有利的。
internet	家里是否联网	1.070938672974736	家里联网不但不会影响成绩，还会促进孩子的学习。
medu	母亲的文化水平	2.196219307541352	母亲的文化水平高低对于孩子的影响是最大的，母亲文化越高孩子学习越好。

由于本次实验的数据集较小，以上分析结果不一定准确，仅供参考。

在线预测部署

生成离线模型后，可以将离线模型部署到线上，通过调用**restful-api**实现在线预测功能。详细步骤请参考**在线预测功能介绍**。

其它

请进入**阿里云数加机器学习平台**体验阿里云机器学习产品，并通过**云栖社区公众号**参与讨论。

相似标签自动归类

背景

本文档使用机器学习平台的文本分析功能，实现一版简单的商品标签自动归类系统。具体场景如下：

双十一购物狂欢节马上又要到来了，各种关于双十一的爆品购物列表在网上层出不穷。对于经常网购的用户来说，一定清楚通常一件商品会有很多维度的标签来展示。比如一个鞋子，它的商品描述可能是“韩都少女英伦风系带马丁靴女磨砂真皮厚底休闲短靴”。如果是一个包，那么它的商品描述可能是“天天特价包包2016新款秋冬斜挎包韩版手提包流苏贝壳包女包单肩包”。

每个产品的描述都包含非常多的维度，可能是时间、产地、款式等，如何按照特定的维度将数以万计的产品进行归类，往往是电商平台最头痛的问题。其中最大的挑战是如何判断每种商品的维度由哪些标签组成。如果可以通过算法自动学习标签词语，例如“日本”、“福建”、“韩国”等与地点相关的标签，那么就可以快速地构建标签归类体系。

数据说明

数据是在网上直接下载并且整理的一份2016年双十一购物清单，一共有两千多条商品描述，每一行代表一款商品的标签聚合，如下图所示。

```
【天天特价】韩都少女英伦风系带马丁靴女磨砂真皮厚底休闲短靴
拉杆箱万向轮24寸铝框皮箱旅行箱女行李箱26寸包硬箱复古登机箱20
韩国孔孝真欧尼oni同款2016新款mini三角小包包手机包单肩斜挎包
迷失麋鹿2016秋冬独家定制款女圆头粗跟靴子沙漠靴子真皮细带短靴
MASOOMAKE秋季中跟厚底马丁靴女英伦风 2016新款系带粗跟短靴子女
羊皮毛一体雪地靴代购天天特价冬保暖运动套筒松糕防滑短筒雪地鞋
潮2016冬季新款欧美加绒高跟女靴尖头粗跟短靴马丁靴短筒棉靴女鞋
2016冬韩国内增高女棉鞋保暖加绒羊羔毛高帮系带休闲鞋韩版雪地靴
天天特价英伦学院风粗跟小皮鞋漆皮系带复古牛津鞋尖头加绒女单鞋
包包2016新款女包韩版简约复古百搭链条包单肩小包包小方包斜挎包
天天特价包包2016新款秋冬斜挎包韩版手提包流苏贝壳包女包单肩包
铝框拉杆箱万向轮行李箱男旅行箱女登机箱20/24/26寸密码箱皮箱子
艾迪猫2016秋冬新款潮单肩斜跨小包包韩版时尚女包斜挎女士小方包
天天特价欧美2016秋季新款平底单鞋懒人鞋马衔扣一脚蹬乐福鞋女鞋
韩国自制翻盖信封包圆环链条单肩包休闲气质百搭简约款女包斜挎包
```

将数据导入机器学习平台进行处理，数据上传方式请参考数据准备。

实验说明

数据上传完成后，通过拖拽机器学习组件，生成如下实验逻辑图，每一步的具体功能如下图所示。

各步骤的详细说明如下。

1. 上传数据并分词

参考**数据准备**上传shopping_data数据，代表底层数据存储。
通过**分词**组件对数据分词，分词是NLP的基础操作，本文不做介绍。

2. 增加序号列

由于上传的数据只有一个字段，需要通过增加序号列为每个数据增加主键，处理后的数据如下图所示。

content ▲	append_id ▲
【 天天 特价 】韩 都 少女 英伦风 系带 马丁 靴 女 磨砂 真皮 ...	0
拉杆箱 万向轮 24 寸 铝框 皮箱 旅行箱 女 行李箱 26 寸 包 硬...	1
韩国 孔孝真 欧尼 oni 同款 2016 新款 mini 三角 小 包包 手机...	2
迷失 麋鹿 2016 秋冬 独家 定制 款 女 圆头 粗跟靴子 沙漠 靴...	3
MASOOMAKE 秋季 中跟 厚底 马丁 靴 女 英伦风 2016 新款 ...	4
羊 皮毛一体雪地靴 代购 天天 特价 冬 保暖 运动 套筒 松糕 防...	5
潮 2016 冬季 新款 欧美 加绒 高跟女靴 尖头 粗跟短靴 马丁 靴...	6
2016 冬 韩国 内增高 女 棉鞋 保暖 加绒 羊羔毛 高帮系带休闲...	7
天天 特价 英伦 学院风 粗跟 小 皮鞋 漆皮 系带 复古 牛津 鞋 ...	8
包包 2016 新款女包 韩版 简约 复古 百搭 链条包 单肩小包包 ...	9
天天 特价包包 2016 新款 秋冬 斜挎包 韩版手提包 流苏 贝壳...	10
铝框 拉杆箱 万向轮 行李箱 男 旅行箱 女 登机箱 20 / 24 / 26 ...	11
艾迪 猫 2016 秋冬新款 潮 单肩 斜跨小包包 韩版 时尚女包 斜...	12
天天 特价 欧美 2016 秋季 新款平底单鞋 懒人 鞋 马 衔 扣 一 ...	13
韩国 自制 翻盖 信封包 圆环 链条单肩包 休闲 气质 百搭 简约...	14

3. 统计词频

展示了每个商品中出现的各种词语的个数。

4. 生成词向量

使用word2vector算法，将每个词按照意义在向量维度展开，词向量有两层含义。

- 向量距离近的两个词的真实含义比较相近，比如数据中的“新加坡”和“日本”都表示产品的产地，那么这两个词的向量距离就比较近。
- 不同词之间的距离差值也具有一定的意义，比如“北京”是“中国”的首都，“巴黎”是“法国”的首都，在训练量足够的情况下，可以得到“|中国|-|北京|=|法国|-|巴黎|”。

经过word2vector算法，将每个词被映射到百维空间上，结果如下图所示。

序号	word	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13
7	加厚	0.1177	0.009646	0.07124	-0.009802	0.008854	-0.1568	-0.2333	0.1643	0.0...	-0....	-0.2...	0.07...	-0.0...	0.02...
8	2016	0.1488	-0.1518	0.1813	0.02331	-0.03854	-0.06455	-0.001774	0.1854	0.1...	-0....	-0.2...	0.04...	0.1299	-0.0...
9	韩版	0.101	-0.02068	0.04436	0.02251	-0.1528	-0.2823	-0.2211	0.2521	0.0...	-0....	-0.2...	0.1006	0.05...	-0.0...
10	/	-0.02318	0.07028	0.189	-0.1704	0.01743	0.1096	0.1458	-0.2436	-0....	0.0...	0.1876	0.08...	-0.0...	0.1625
11	新款	0.1374	-0.05232	0.08965	0.09086	-0.09875	-0.2254	-0.1866	0.2333	0.0...	-0....	-0.189	0.07...	-0.0...	0.0253
12	6	-0.131	0.08679	0.009914	-0.3171	-0.1743	-0.1615	0.005242	-0.102	-0....	0.1...	-0.0...	0.1186	0.08...	0.1017
13	包邮	0.06004	0.04959	0.1578	0.1021	0.04368	0.1318	-0.05841	-0.01082	-0....	-0....	0.05...	0.1499	0.03...	0.0249
14	简约	-0.065	0.01107	0.02025	-0.1287	-0.09461	-0.1241	-0.05828	0.1292	-0....	0.0...	-0.0...	0.1496	0.08...	-0.1...
15	冬季	0.1803	-0.04212	0.1512	0.06145	-0.02388	-0.1422	-0.1718	0.1897	0.1...	-0....	-0.1...	0.08...	-0.0...	0.02...
16	秋冬	0.1078	-0.07883	0.1803	0.02858	-0.08247	-0.1859	-0.1708	0.2181	0.0...	-0....	-0.1...	0.1327	0.07...	-0.0...
17	-	0.04343	0.1467	0.1142	-0.2973	0.05655	0.1708	0.01833	-0.09293	-0....	0.1...	0.00...	0.1291	-0.0...	0.1162
18	纯棉	0.06417	-0.08088	0.07554	0.04668	-0.07626	-0.2355	-0.1062	0.1727	0.0...	-0....	-0.1...	0.08...	0.09...	-0.0...
19	韩国	0.0284	-0.03408	0.1062	-0.02404	-0.04606	-0.0249	-0.01154	0.05106	0.0...	0.0...	-0.0...	0.06...	0.1056	0.00...
20	家用	0.09303	0.004674	0.151	-0.08795	0.03799	0.1286	0.1244	-0.1209	0.0...	0.1...	0.07...	0.1694	0.2598	0.1493
21	g	0.06279	0.01393	0.2534	-0.01994	0.03998	0.3231	0.07817	-0.07714	-0....	0.1...	0.06...	0.1422	0.1651	0.03...
22	男	0.06893	0.009893	0.1051	0.0005736	-0.02107	-0.1202	-0.1323	0.1462	-0....	-0....	-0.1...	0.09...	-0.0...	0.0156

5. 词向量聚类

使用kmeans算法，在已经产生的词向量的基础上，计算出哪些词的向量距离比较近，并按照意义将标签词自动归类。结果展示的是每个词属于哪个聚类簇，如下图所示。

word	cluster_index
家用	83
g	83
男	79
套装	94
保暖	98
加绒	98
儿童	79
潮	90
正品	87

结果验证

通过SQL组件，在聚类簇中随意挑选一个类别，判断是否将同一类别的标签进行了自动归类，本实验选用第

10组聚类簇。

```
6 select * from ${t1} where
  cluster_index=10
```

结果如下图所示。

word ▲	cluster_index ▲
日本进口	10
俄罗斯	10
雨	10
坚果	10
台湾	10
韩国进口	10
男士内裤	10
记	10
云南	10
螺	10
油	10
新疆特产	10

通过结果中的“日本”、“俄罗斯”、“韩国”、“云南”、“新疆”、“台湾”等词可以发现系统自动将一些跟地理相关的标签进行了归类，但是里面混入了“男士内裤”、“坚果”等明显与类别不符合的标签。可能是训练样本数量不足造成的，如果训练样本足够大，那么标签聚类结果会非常准确。

TensorFlow实现图像分类

背景

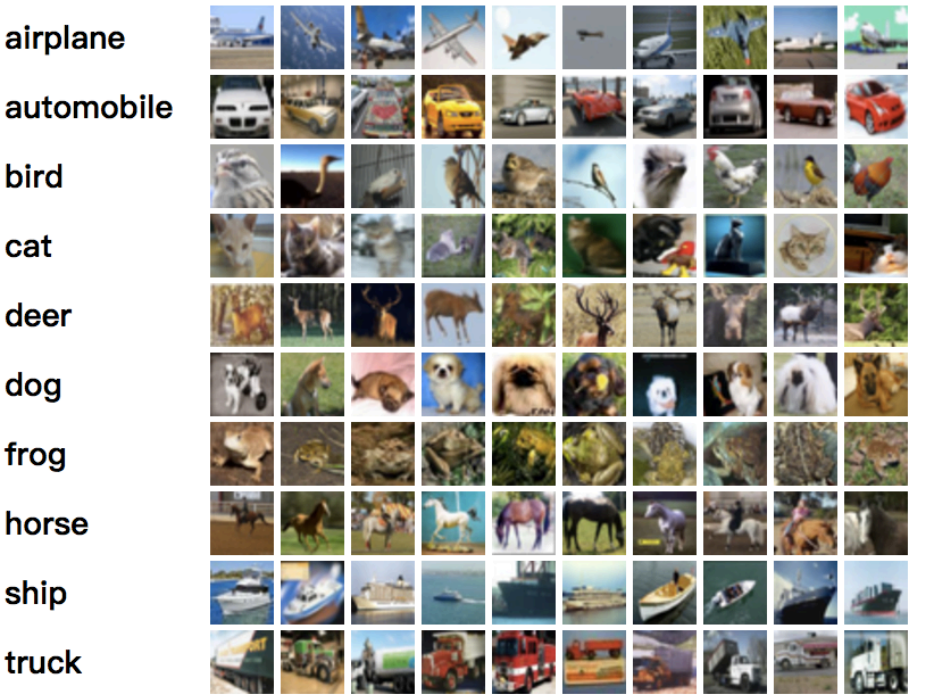
随着互联网的发展，产生了大量的图片以及语音数据，如何将这部分非结构化数据有效地利用起来，一直是困扰数据挖掘工程师的难题。首先，解决非结构化数据问题通常要使用深度学习算法，上手门槛高。其次，对于这部分数据的处理，往往需要依赖GPU计算引擎，计算资源代价大。
本文档通过阿里云机器学习产品，使用Tensorflow深度学习框架，快速搭建了图像识别的预测模型。整个流程只需要半小时，就可以实现对下面这幅图片的识别，系统会返回“鸟”。这种使用深度学习实现图片识别的案例也可以用在图片检黄、人脸识别、物体检测等各个领域。

数据集介绍

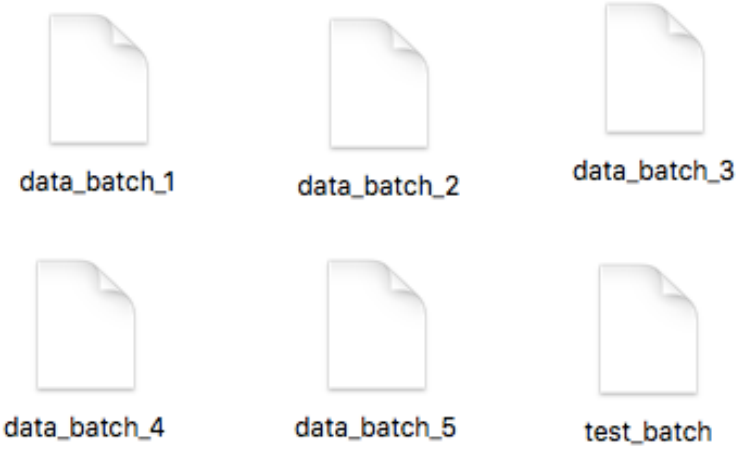
本案例数据集及相关代码下载地址：
https://help.aliyun.com/document_detail/51800.html?spm=5176.doc50654.6.564.mS4bn9。

CIFAR-10数据集：包含6万张像素为32*32的彩色图片，分成10个类别，分别是飞机、汽车、鸟、毛、鹿、狗

、青蛙、马、船、卡车。数据集截图如下。



数据源在使用过程中被拆分成两部分，其中5万张用于训练，1万张用于测试。5万张训练数据又被拆分成5个data_batch，1万张测试数据组成test_batch。最终数据源如下图所示。



数据探索流程

搭建实验前，需要开通阿里云机器学习产品的GPU使用权限，并且开通OSS，用于存储数据。

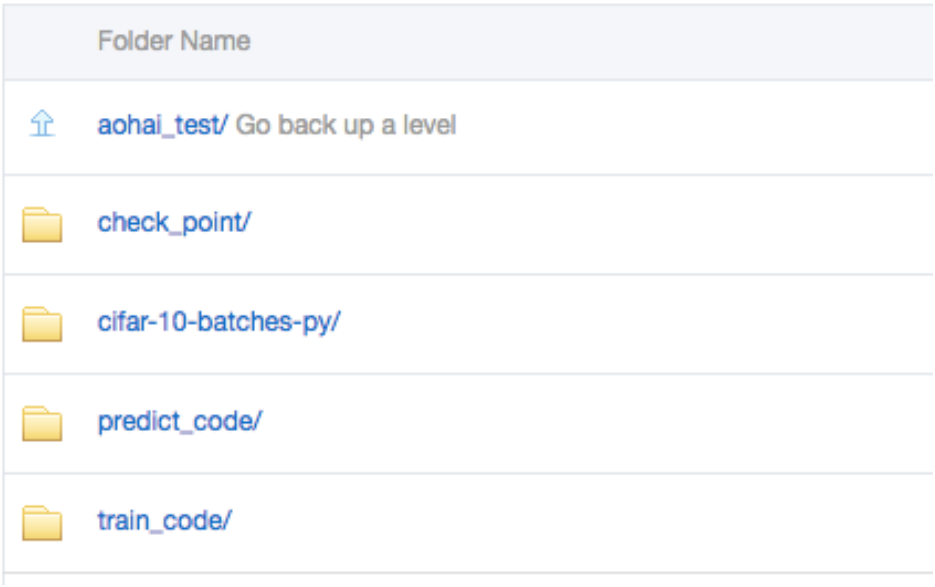
机器学习产品控制台：https://data.aliyun.com/product/learn?spm=a21gt.99266.416540.112.IOG7OU
OSS控制台：https://www.aliyun.com/product/oss?spm=a2c0j.103967.416540.50.KkZyBu

1. 数据源准备

下载本案例提供的相关数据和代码，并解压缩。

进入OSS对象存储，创建OSS存储空间，即OSS Bucket（详细请参考OSS产品文档）。

在OSS Bucket中新建目录，本案例创建了“aohai_test”文件夹，并在这个目录下建立了如下4个文件夹目录。



每个文件夹的作用如下：

- check_point：用来存放实验生成的模型。
- cifar-10-batches-py：用来存放训练数据和预测集数据，对应数据源的cifar-10-batcher-py文件和bird_mount_bluebird.jpg文件。
- predict_code：用来存放预测代码文件，对应数据源的cifar_predict_pai.py文件。
- train_code：用来存放训练代码文件，对应数据源的cifar_pai.py文件。

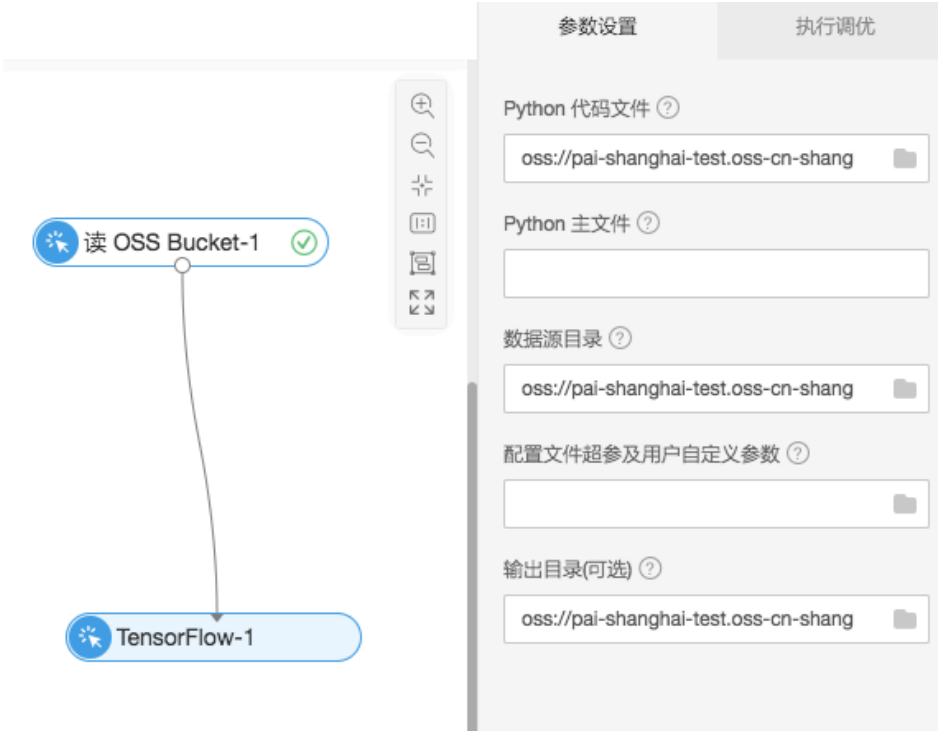
将已经下载好的数据和代码文件上传到OSS Bucket的对应目录下。

2. OSS访问授权配置

进入阿里云机器学习平台，单击**设置**，配置OSS的访问授权，如下图所示，详细请参见深度学习文档的“读OSS Bucket”章节。

3. 模型训练

在控制台左侧的组件区域中拖拽“读OSS Bucket”和“Tensorflow”组件并链接，完成后配置“Tensorflow”组件的参数，如下图所示。

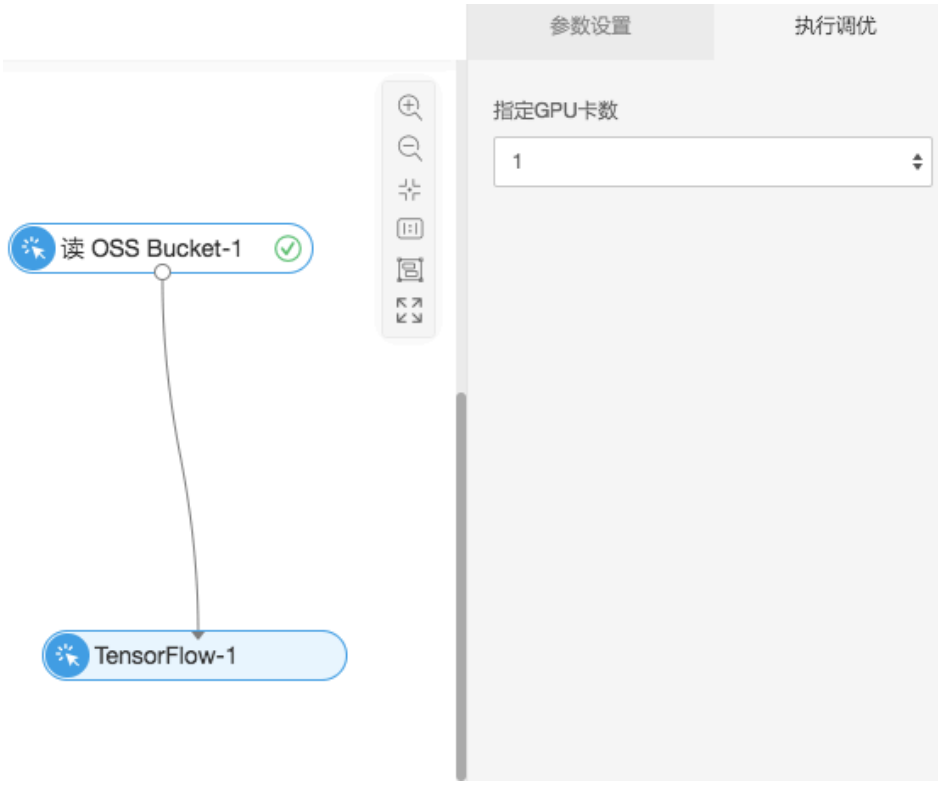


参数说明如下：

- Python代码文件：OSS中的cifar_pai.py文件。
- 数据源目录：OSS中的cifar-10-batches-py文件夹。

- 输出目录：OSS中的check_point文件夹。

单击**运行**，实验开始训练。
可以灵活调节底层的GPU资源。支持在界面端进行设置，如下图所示。也支持通过代码文件进行设置，代码编写需要符合Tensorflow的多卡规范。



4. 模型训练代码解析

“cifar_pai.py”文件中的关键代码说明如下：

- 构建CNN图片训练模型

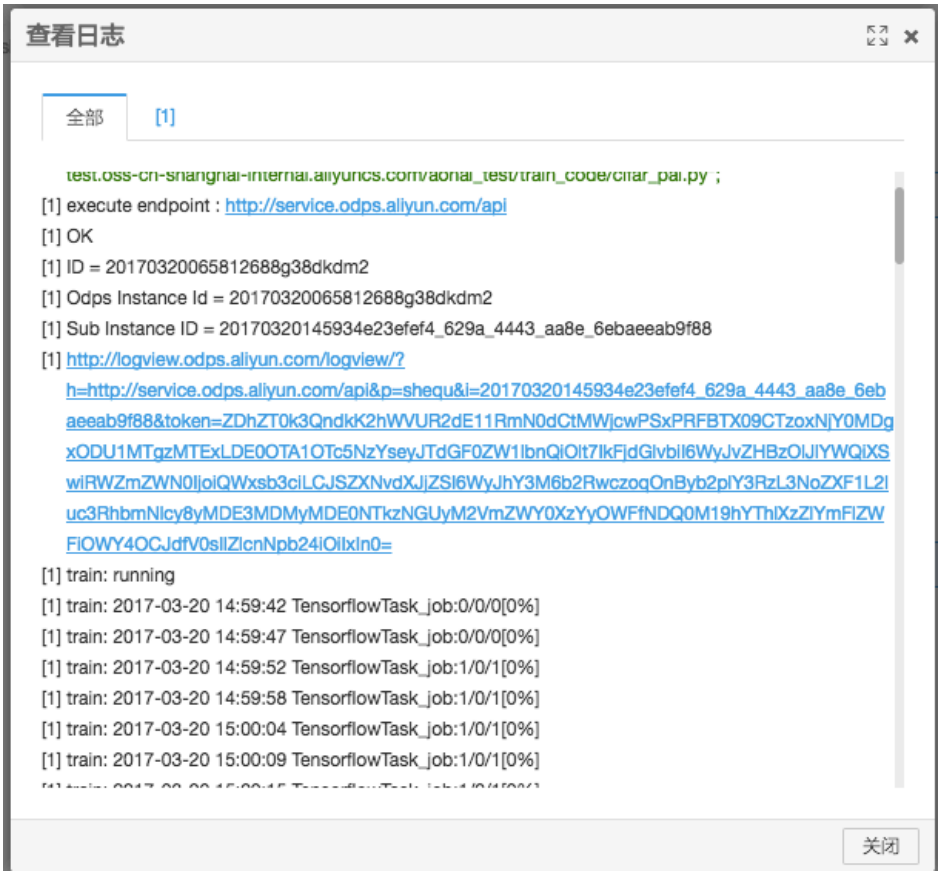
```
network = input_data(shape=[None, 32, 32, 3],
data_preprocessing=img_prep,
data_augmentation=img_aug)
network = conv_2d(network, 32, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 64, 3, activation='relu')
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
network = fully_connected(network, 512, activation='relu')
network = dropout(network, 0.5)
network = fully_connected(network, 10, activation='softmax')
network = regression(network, optimizer='adam',
loss='categorical_crossentropy',
learning_rate=0.001)
```

- 训练生成模型名为model的一系列文件，这些文件组成了TF的预测模型

```
    model = tflearn.DNN(network, tensorboard_verbose=0)
model.fit(X, Y, n_epoch=100, shuffle=True, validation_set=(X_test, Y_test),
show_metric=True, batch_size=96, run_id='cifar10_cnn')
model_path = os.path.join(FLAGS.checkpointDir, "model.tfl")
print(model_path)
model.save(model_path)
```

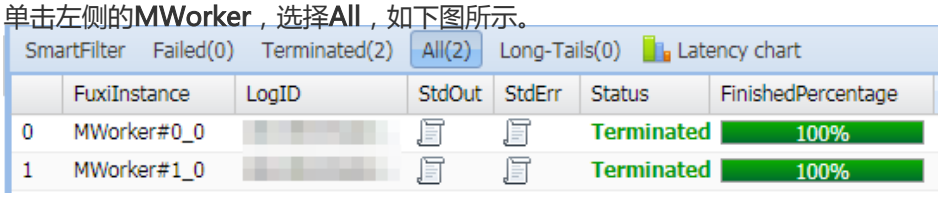
5. 日志查看

实验运行过程中，右键单击“Tensorflow”组件，选择**查看日志**，结果如下图所示。



单击上图中的蓝色logview链接，执行以下步骤查看日志。

i. 打开**ODPS Tasks**下面的**Algo Task**。
ii. 双击**Tensorflow Task**。
iii. 单击左侧的**MWorker**，选择**All**，如下图所示。

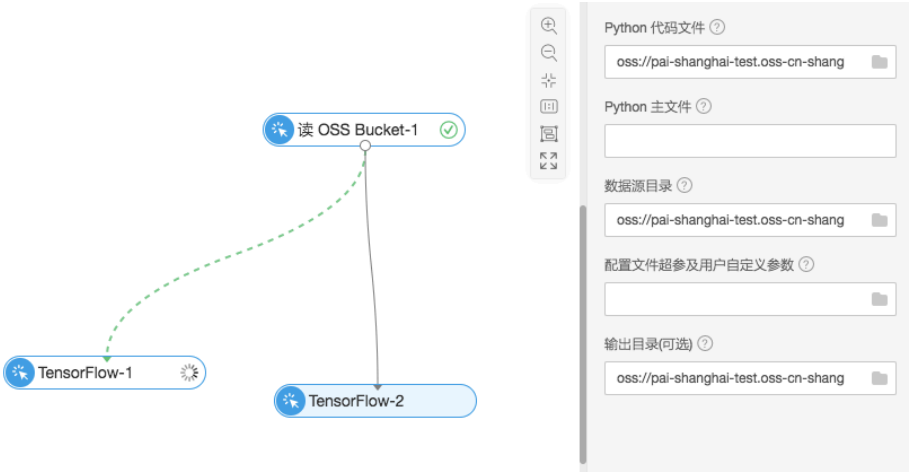


iv. 单击StdOut下方的小图标，模型训练日志被实时地打印出来，如下图所示。

实验运行过程中会不断地打印日志，也可以使用print函数在代码中打印关键信息，结果会显示在日志中。可以通过日志中的“acc”参数查看本案例模型训练的准确度。

6. 结果预测

在控制台左侧的组件区域中拖拽一个“Tensorflow”组件用于预测，并配置参数，如下图所示。



参数说明如下：

- Python代码文件：OSS中的cifar_predict_pai.py文件。
- 数据源目录：OSS中的cifar-10-batches-py文件夹，用来读取bird_mount_bluebird.jpg文件。
- 输出目录：读取OSS中的check_point文件夹下模型训练生成的model.tfl文件。

预测的图片是存储在“checkpoint”文件夹下的图。

可通过日志查看结果，如下图所示。

7. 预测代码解析

部分预测代码说明如下：

```
  predict_pic = os.path.join(FLAGS.buckets, "bird_bullocks_oriole.jpg")
img_obj = file_io.read_file_to_string(predict_pic)
file_io.write_string_to_file("bird_bullocks_oriole.jpg", img_obj)

img = scipy.ndimage.imread("bird_bullocks_oriole.jpg", mode="RGB")

# Scale it to 32x32
img = scipy.misc.imresize(img, (32, 32), interp="bicubic").astype(np.float32, casting='unsafe')

# Predict
prediction = model.predict([img])
print (prediction[0])
print (prediction[0])
#print (prediction[0].index(max(prediction[0])))
num=['airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck']
print ("This is a %s"%(num[prediction[0].index(max(prediction[0]))]))
```

1. 读入图片“bird_bullocks_oriole.jpg”，将图片像素调整为32*32。

2. 将该图片作为参数，传入model.predict预测函数中进行评分。
3. 最终返回这张图片对应的十种分类
   [‘airplane’,’ automobile’,’ bird’,’ cat’,’ deer’,’ dog’,’ frog’,’ horse’,’ ship’,’ truck’]的权重，选择权重最高的一项作为预测结果返回。

注意：因为模型训练存在随机性，所以不保证每次训练出的模型对于预测图片都可以返回准确结果，需要不断调试对应参数才能达到稳定效果，本案例比较简单，仅供参考。

雾霾天气预测

背景

如果要人们评选当今最受关注话题的top10榜单，雾霾一定能够入选。如今走在北京街头，随处可见带着厚厚口罩的人在埋头前行，雾霾天气不光影响了人们的出行和娱乐，对于人们的健康也有很大危害。本文通过分析北京一年来的真实天气数据，挖掘出二氧化氮是跟雾霾天气（指PM2.5）相关性最强的污染物，从而为您揭秘形成雾霾的罪魁祸首。

登录阿里云机器学习平台，通过模板创建雾霾天气预测实验。

数据集介绍

数据源：2016全年的北京天气指标。

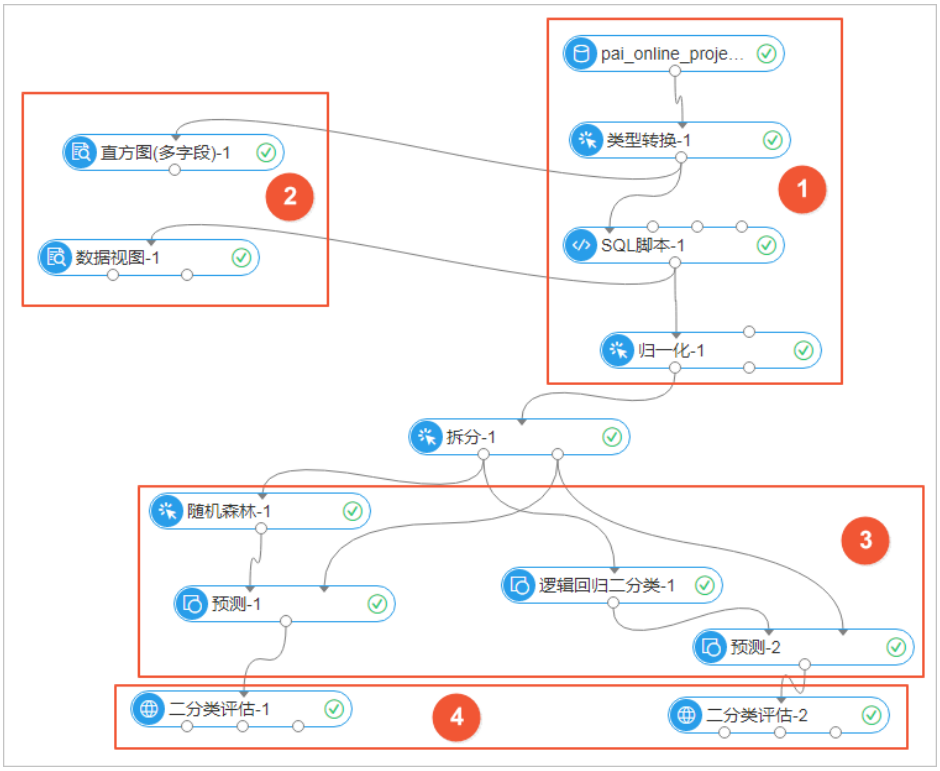
采集的是从2016年1月1号以来每个小时的空气指标数据，具体字段如下表。

字段名	含义	类型

time	日期，精确到天	string
hour	表示的是时间，第几小时的数据	string
pm2	pm2.5的指标	string
pm10	pm10的指标	string
so2	二氧化硫的指标	string
co	一氧化碳的指标	string
no2	二氧化氮的指标	string

数据探索流程

实验流程如下。

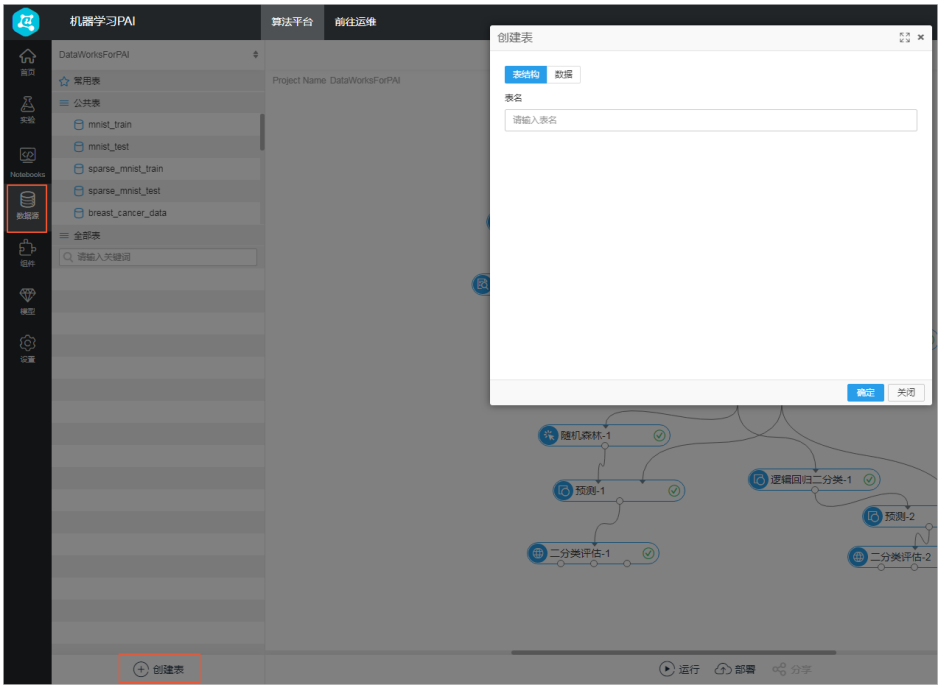


整个实验分为四部分，分别是数据导入及预处理（上图的1）、统计分析（上图的2）、模型训练及预测（上图的3）、模型评估分析（上图的4），详细介绍如下。

1. 数据导入及预处理

数据导入

单击**数据源**，选择**创建表**，上传的数据支持.txt和.csv文件。



数据导入后，右键单击组件，选择**查看数据**，结果如下。

time	hour	pm2	pm10	so2	co	no2
2016...	2	85	123	18	1.8	72
2016...	8	114	127	25	2.3	81
2016...	11	123	140	27	2.5	83
2016...	14	134	150	30	2.6	86
2016...	17	150	168	32	2.8	92
2016...	20	166	191	34	3	97
2016...	23	179	207	35	3.2	101
2016...	1	190	222	37	3.4	104
2016...	10	225	249	39	3.8	107
2016...	19	244	287	41	4	113

数据预处理

通过“类型转换”组件把string类型的数据转换成double类型。

通过“SQL脚本”组件，将目标列转换成0和1的二值类型。本实验中“pm2”列为目标列，数值超过200的作为重度雾霾天气打标为1，低于200为0，实现的SQL语句如下。

```
select time,hour,(case when pm2>200 then 1 else 0 end),pm10,so2,co,no2 from ${t1};
```

归一化

归一化的主要作用是去除量纲，即把不同指标的污染物的单位进行统一。

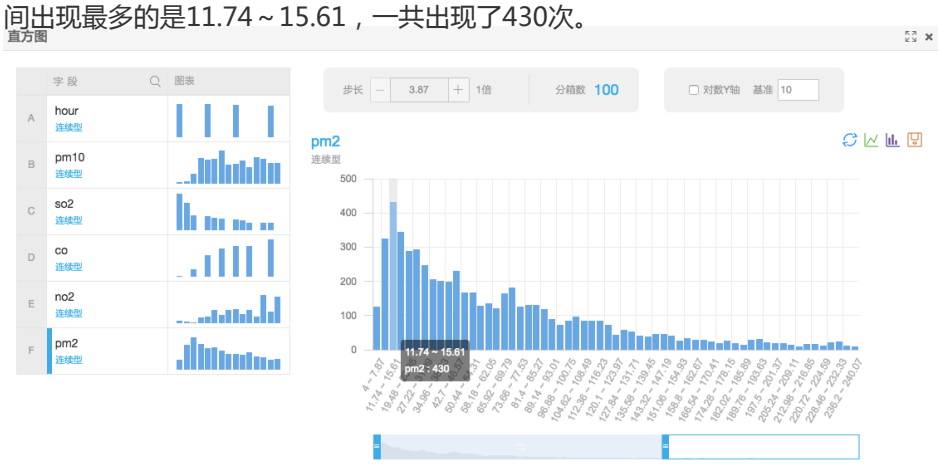
time	hour	_c2	pm10	so2	co	no2
20160101	2	0	0.24532224...	0.21917808219...	0.36956521739130427	0.43312101910828027
20160101	8	0	0.25363825...	0.31506849315...	0.4782608695652173	0.49044585987261147
20160101	11	0	0.28066528...	0.34246575342...	0.5217391304347825	0.5031847133757962
20160101	14	0	0.30145530...	0.38356164383...	0.5434782608695652	0.5222929936305732
20160101	17	0	0.33887733...	0.41095890410...	0.5869565217391303	0.5605095541401274
20160101	20	0	0.38669438...	0.43835616438...	0.6304347826086956	0.5923566878980892
20160101	23	0	0.41995841...	0.45205479452...	0.6739130434782609	0.6178343949044586
20160102	1	0	0.45114345...	0.47945205479...	0.7173913043478259	0.6369426751592356
20160102	10	1	0.50727650...	0.50684931506...	0.8043478260869563	0.6560509554140127
20160102	19	1	0.58627858...	0.53424657534...	0.8478260869565216	0.6942675159235668
20160102	22	1	0.68191268...	0.53424657534...	0.8913043478260869	0.7197452229299363
20160103	0	1	0.74428274...	0.53424657534...	0.8913043478260869	0.732484076433121
20160105	16	0	0.06860706...	0.02739726027...	0.06521739130434782	0.16560509554140126

2. 统计分析

直方图

通过“直方图”组件可以可视化地查看不同数据在不同区间下的分布。

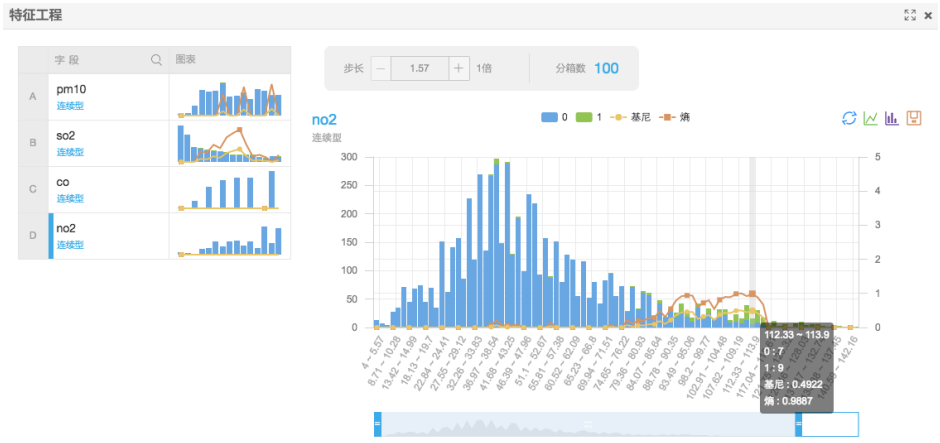
本实验通过可视化的展现，直观地看到了每个字段数据的分布情况。如下图，以PM2.5为例，数值区间出现最多的是11.74～15.61，一共出现了430次。



数据视图

通过数据视图可以查看不同指标的不同区间对于结果的影响。

如下图，以no2为例，在112.33～113.9区间产生了7个目标列为0的目标，产生了9个目标列为1的目标。即当no2在112.33～113.9区间的情况下，出现重度雾霾的天气的概率是非常大的。熵和基尼系数表示这个特征区间对于目标值的影响（信息量层面的影响），数值越大影响越大。

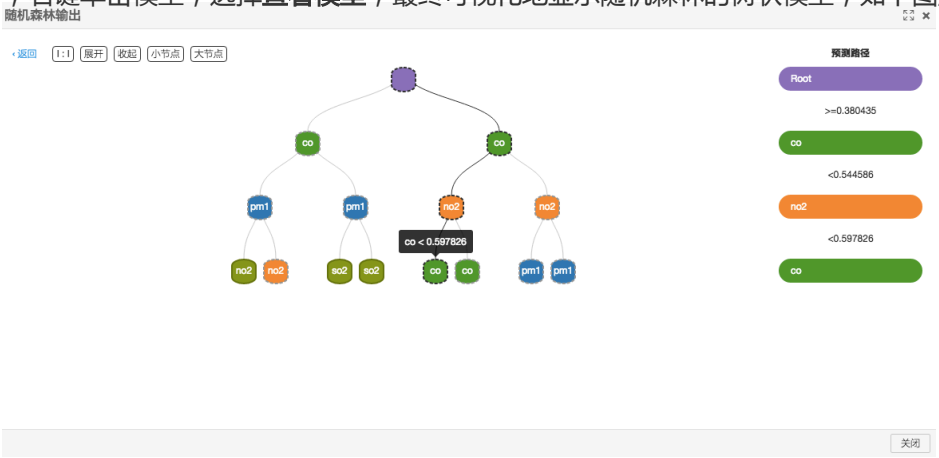


3. 模型训练及预测

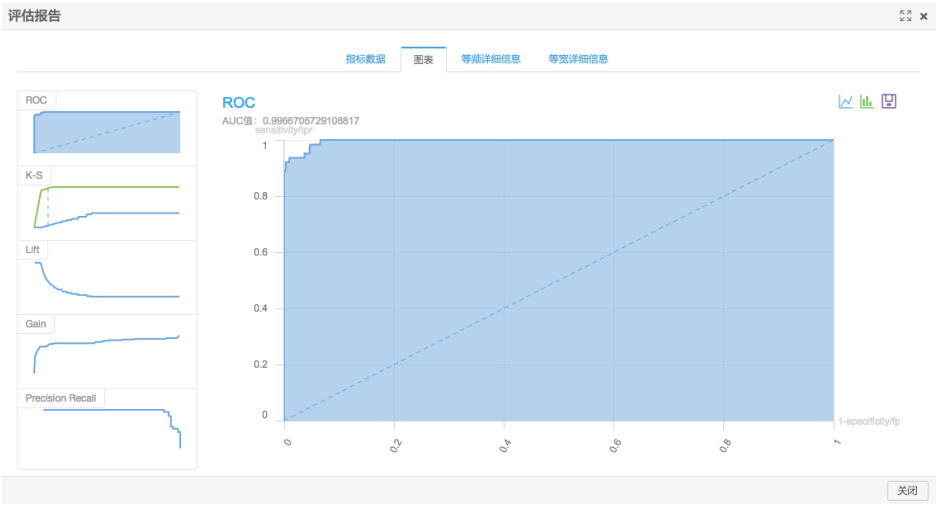
本案例采用了两种不同的算法对结果进行预测和分析，分别是随机森林和逻辑回归。

随机森林

将数据集拆分，百分之八十用来训练模型，百分之二十用来预测。单击控制台左边的**模型**，选择**已保存模型**，右键单击模型，选择**查看模型**，最终可视化地显示随机森林的树状模型，如下图所示。



预测结果如下图。



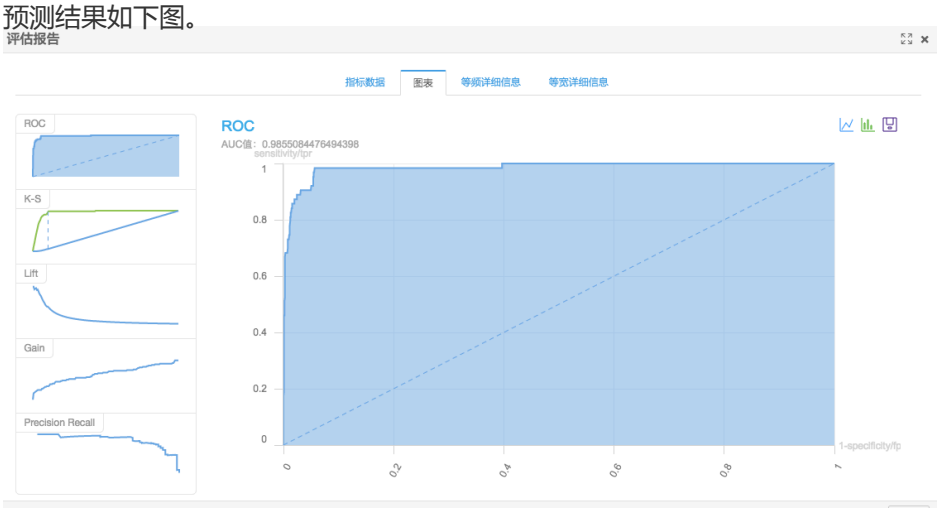
上图中的**AUC**为0.99，说明当有了本文档用到的天气指标数据，就可以预测天气是否雾霾，而且准确率可以达到百分之九十以上。

逻辑回归

使用逻辑回归算法训练得到的是一个线性模型，如下图所示。

字段名	权重	
	1	0
pm10	18.32146628653672	-
so2	1.767062094833547	-
co	-0.2519492790928399	-
no2	10.95221282178011	-
常量	-16.66654139199668	0

预测结果如下图。

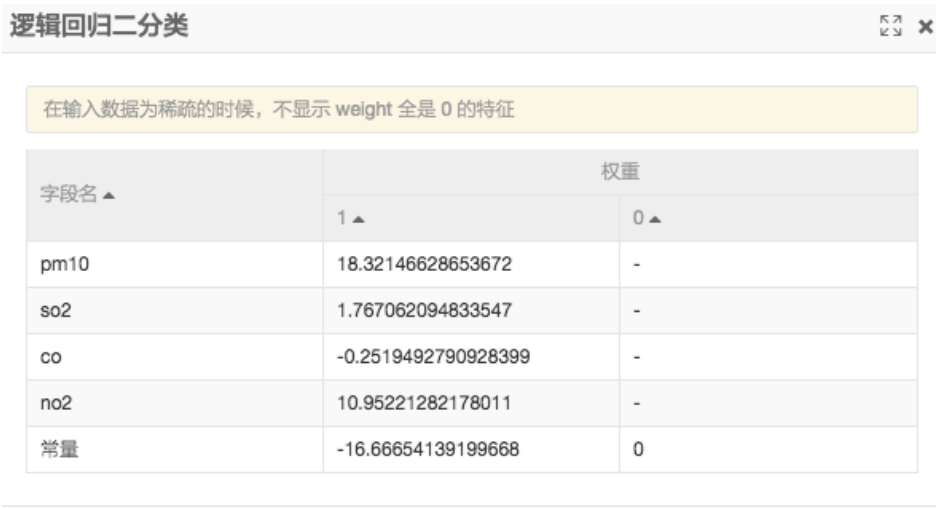


上图中的AUC为0.98，比用随机森林计算得到的结果略低一点。如果排除调参对于结果的影响，可以说明针对这个数据集，随机森林的训练效果会更好一些。

模型评估分析

根据上文中的模型和预测结果来分析哪种空气指标对于PM2.5影响最大。

逻辑回归生成的模型如下图所示。



逻辑回归二分类

在输入数据为稀疏的时候，不显示 weight 全是 0 的特征

字段名	权重	
	1	0
pm10	18.32146628653672	-
so2	1.767062094833547	-
co	-0.2519492790928399	-
no2	10.95221282178011	-
常量	-16.66654139199668	0

经过归一化计算的逻辑回归算法的模型系数越大，对于结果的影响越大。系数符号为正表示正相关，为负表示负相关。上图中正号系数里pm10和no2最大。

- pm10和pm2只是颗粒尺寸大小不同，是一个包含关系，可以不考虑。
- no2（二氧化氮）对于pm2.5的影响最大。查阅相关文档，了解哪些因素会造成no2的大量排放，即可找出影响pm2.5的主要因素。
  通过来自互联网的no2来源文章，说明了no2主要来自汽车尾气。

其它

请进入阿里云数加机器学习平台体验阿里云机器学习产品，并通过云栖社区公众号参与讨论。

Caffe实现图片分类

背景

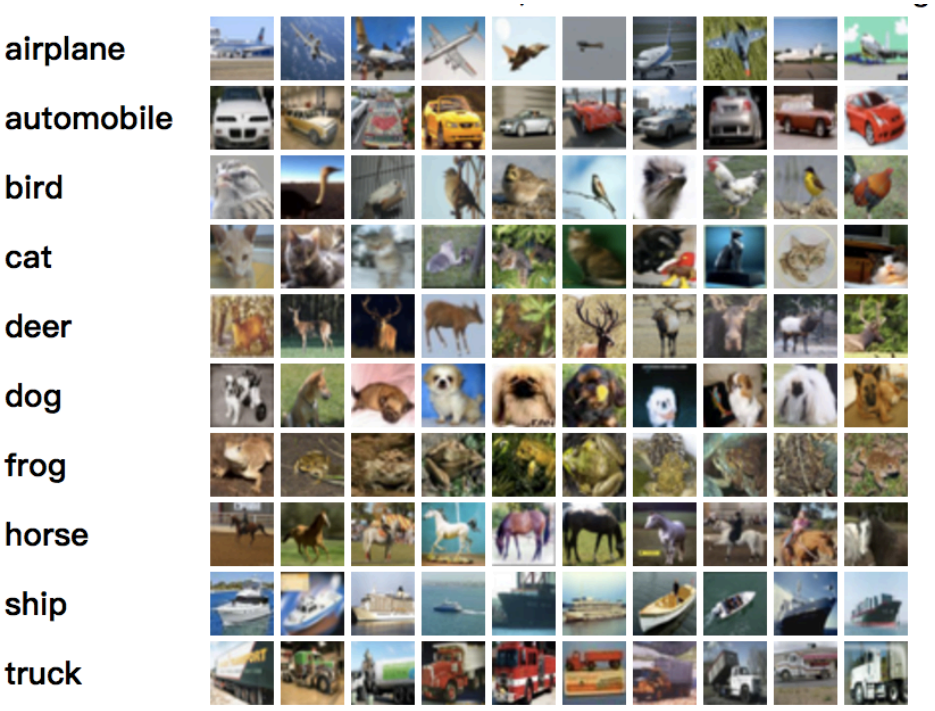
TensorFlow实现图像分类文档介绍了如何通过深度学习的TensorFlow框架，实现对Cifar10图像的分类。

本文档介绍另一个深度学习框架Caffe，通过Caffe只需要填写一些配置文件就可以实现图像分类的模型训练。

请提前阅读**深度学习**文档，在机器学习平台上开通深度学习功能，文末提供了相关下载链接。

数据介绍

本文使用的是cifar10开源数据集，包含6万张像素为32*32的彩色图片，这6万张图片被分成10个类别，分别是飞机、汽车、鸟、毛、鹿、狗、青蛙、马、船、卡车，数据集截图如下。



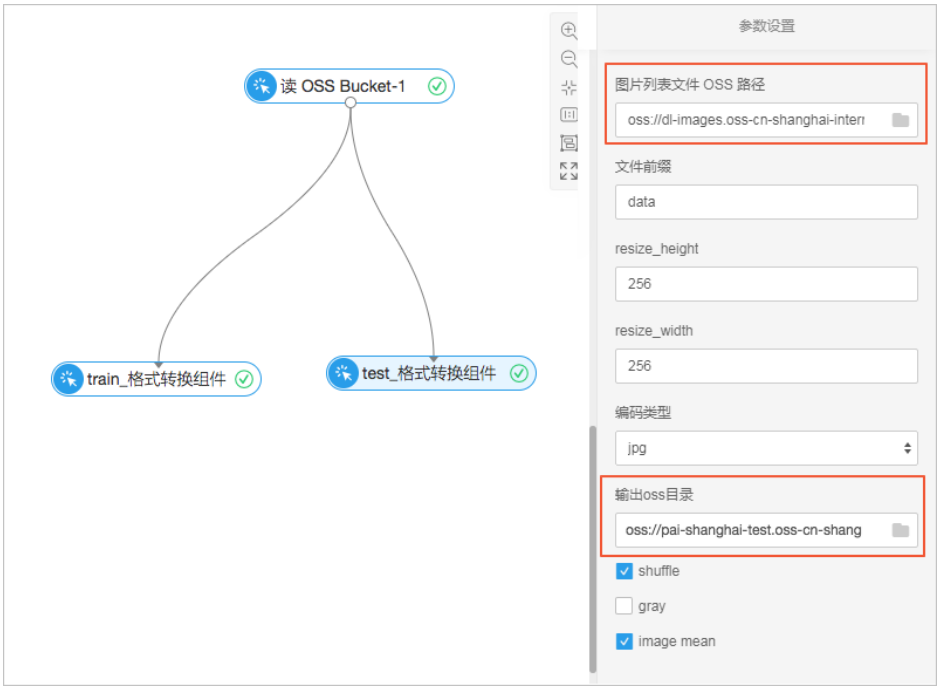
这份数据已经内置在机器学习平台的公共数据集中，以jpg格式存储。任何机器学习用户都可以在深度学习组件的**数据源目录**中直接输入以下路径：

- 测试数据： oss://dl-images.oss-cn-shanghai-internal.aliyuncs.com/cifar10/caffe/images/cifar10_test_image_list.txt
- 训练数据：oss://dl-images.oss-cn-shanghai-internal.aliyuncs.com/cifar10/caffe/images/cifar10_train_image_list.txt

如下图所示。

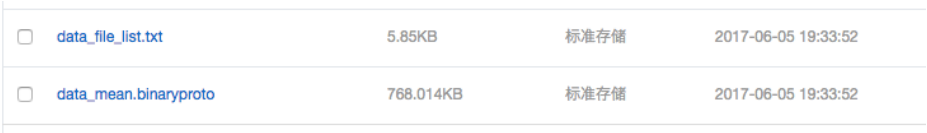
格式转换

目前深度学习的Caffe框架只支持特定的格式，所以首先需要使用“格式转换”组件，对jpg格式的图片进行转换。



- 图片列表文件OSS路径：上文提到的机器学习内置的公共数据集。
- 输出oss目录：用户自定义的OSS目录。

经过格式转换，在输出的OSS目录下生成如下文件，训练数据和测试数据各一份。



需要记录对应的OSS路径用于Net文件的填写，假设路径名分别是：

训练数据data_file_list.txt：bucket/cifar/train/data_file_list.txt

训练数据data_mean.binaryproto:bucket/cifar/train/data_mean.binaryproto

测试数据data_file_list.txt：bucket/cifar/test/data_file_list.txt

测试数据data_mean.binaryproto:bucket/cifar/test/data_mean.binaryproto

Caffe配置文件

Net文件编写，对应上文格式转换生成的路径：

```
  transform_param {
    mean_file: "bucket/cifar/train/data_mean.binaryproto"
    crop_size: 31
  }
  binary_data_param {
    source: "bucket/cifar/train/data_file_list.txt"
    batch_size: 100
  }
}
layer {
  name: "cifar"
  type: "BinaryData"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    mean_file: "bucket/cifar/test/data_mean.binaryproto"
    crop_size: 31
  }
  binary_data_param {
    source: "bucket/cifar/test/data_file_list.txt"
    batch_size: 100
  }
```

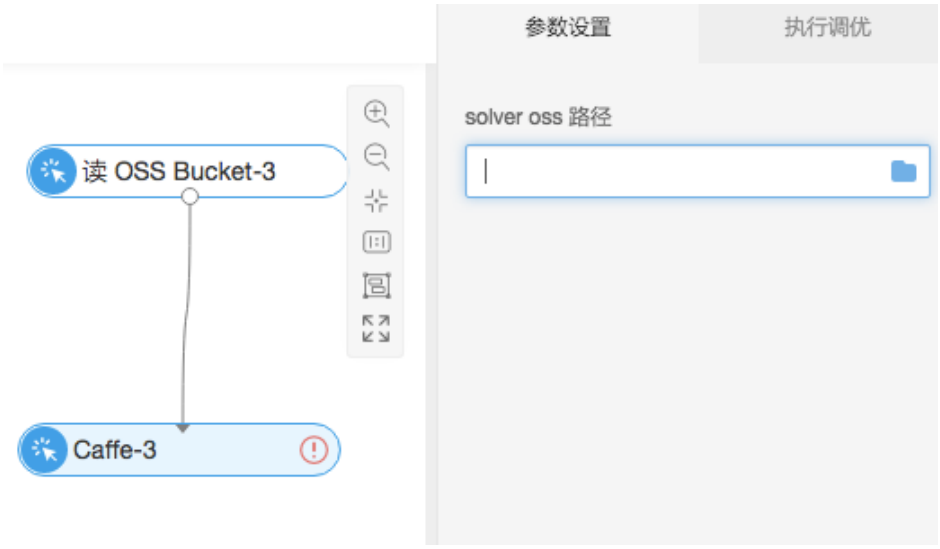
Solver文件编写：

```
# The train/test net protocol buffer definition
net: "填写net文件的OSS路径"
# test_iter specifies how many forward passes the test should carry out.
# In the case of MNIST, we have test batch size 100 and 100 test iterations,
# covering the full 10,000 testing images.
test_iter: 100
# Carry out testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
base_lr: 0.001
momentum: 0.9
weight_decay: 0.004
# The learning rate policy
lr_policy: "fixed"
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 5000
# snapshot intermediate results
snapshot_after_train: true
# snapshot: 10000
# snapshot_format: HDF5
snapshot_prefix: "生成model的存储路径"
# solver mode: CPU or GPU
solver_mode: GPU
data_distribute_mode: MANUALLY
model_average_iter_interval: 1
```

运行

将编辑好的Solver文件和Net文件上传到OSS上，拖拽Caffe组件到画布中，并与数据源链接。

配置Caffe组件参数，如下图所示，**sovler oss路径**选择已经上传到OSS上的Solver文件，单击**运行**。



在OSS的模型路径下查看生成的图片分类模型文件，结果如下，可以用以下模型进行图片分类。

参考TensorFlow实现图像分类的“日志查看”章节，查看日志。

相关下载

Tensorflow相关下载

TensorFlow_mnist

需要把以下三个文件都上传到OSS同一目录下。

执行代码下载

训练数据下载

测试数据下载

Tensorflow_cifar10案例

请结合云栖社区相关图片识别案例使用。

训练数据

训练代码

预测代码

预测图片

TensorBoard

TensorBoard代码下载

Tensorflow写歌案例

代码及数据

Tensorflow多机多卡案例

详细使用情况请参见深度学习。

多机多卡案例代码下载

MXNet相关下载

执行代码包下载

超参配置文件下载

训练数据集下载

测试数据集下载

Caffe相关下载

Caffe Mnist下载

caffe_mnist（包含solver文件以及训练数据）

Caffe Cifar下载

caffe_cifar10