机器学习云服务平台的业务架构及业务应用

首先非常高兴可以和各位gitchat的同学做一次分享，这次分享希望给大家介绍一些关于机器学习云服务相关的理念，也欢迎大家来阿里云机器学习PAI上面做客，为我们的下一步产品建设提供更多的输入。产品地址：https://data.aliyun.com/product/learn

机器学习平台PAI特点

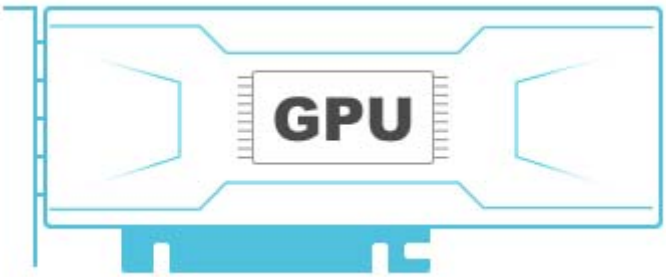
阿里云机器学习PAI（Platform of Artificial Intelligence）是一款一站式的机器学习平台，包含数据预处理、特征工程、常规机器学习算法、深度学习框架、模型的评估以及预测这一整套机器学习相关服务。得益于底层的飞天计算平台的CPU集群以及GPU集群，PAI可以为用户提供PB级别数据的高效计算保证。另外，PAI还将算法组件进行封装，并且增添了大量的可视化工具，让用户可以低门槛上手，真正实现人工智能触手可及。目前无论是国内还是国际上，有许多互联网公司都推出了类似PAI这样的PAAS层机器学习服务，PAI目前在国内的竞争中处于领先集团。

PAI特点1

目前机器学习PAI平台已经涵盖了100余种算法组件，包含聚类、分类、回归、文本分析、关系网络等种类的算法。

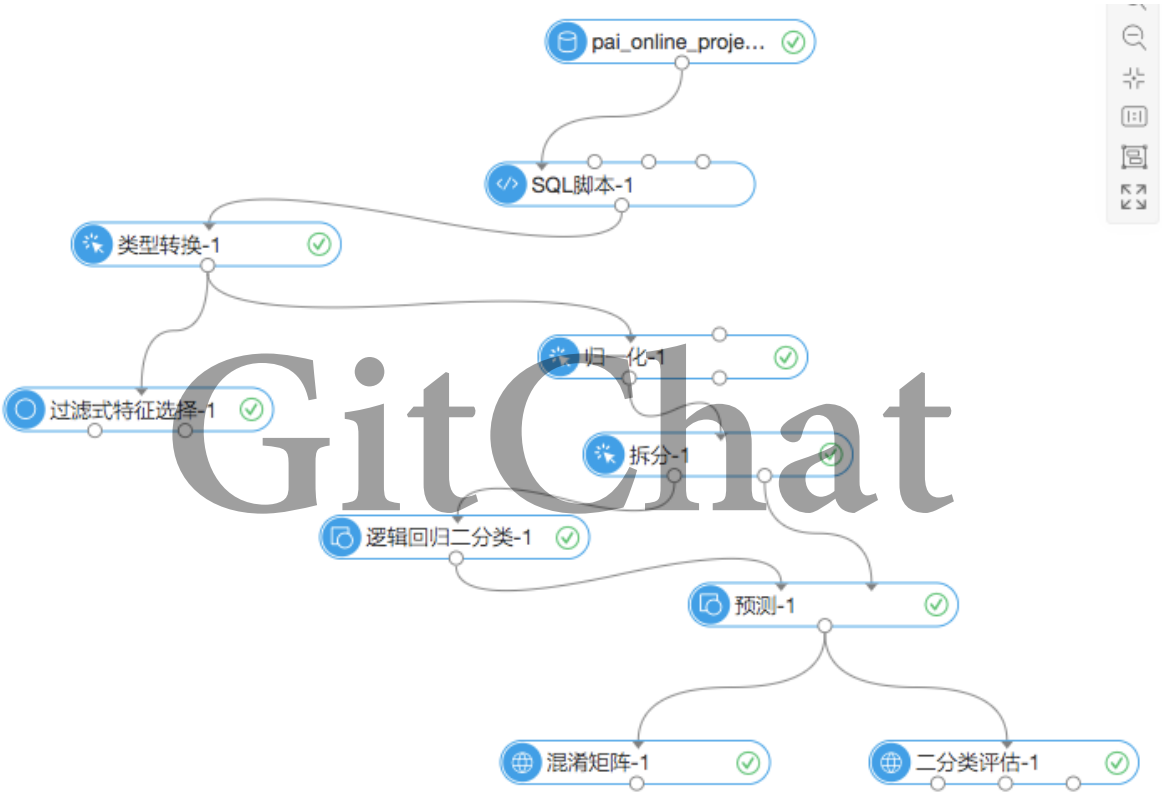
PAI特点2

PAI平台底层支持CPU以及GPU的集群，在GPU集群之上支持了业内主流的三款深度学习框架TensorFlow、Caffe、MXNet。



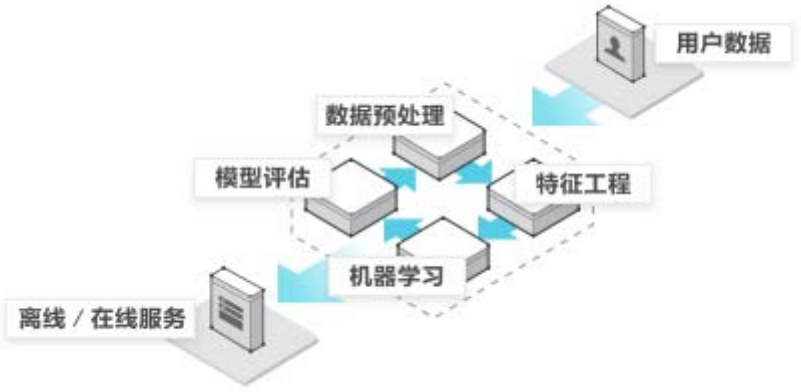
PAI特点3

提供拖拉拽的组件操作方式，让搭建机器学习实验像搭建积木一样简单。



PAI特点4

提供一站式的机器学习企业级服务覆盖机器学习的的整个流程，包含数据的预处理、特征工程、机器学习算法、深度学习框架、评估和预测。



机器学习云平台与自建基于开源的机器学习框架的区别

其实如果要搭建一套企业级机器学习服务架构，大体的架构是一致的，作为参考，首先我们来看下PAI的架构。

从下向上看：

1. 首先在底层需要有计算基础设施，无论是CPU、GPU或是FPGA，需要有调度系统来统一调度底层的计算资源。

2. 向上需要有支持分布式计算架构的框架，常见的有ParameterServer或者Mapreduce、Tensorflow等，这一层的作用是将算法计算任务通过分布式框架分发到底层的基础设施。

3. 再向上一层就是各种算法以及数据预处理或者统计相关的工具，常见的lr、svm等算法都在这一层实现。

4. 最上一层就是基于算法搭建的各种业务场景下的服务。另外，看架构图右边，一个成熟的机器学习服务，还需要支持模型的离线和在线预测，用以支持各种类型的业务需求。

云服务与开源的框架的区别表现在那几个方面呢？

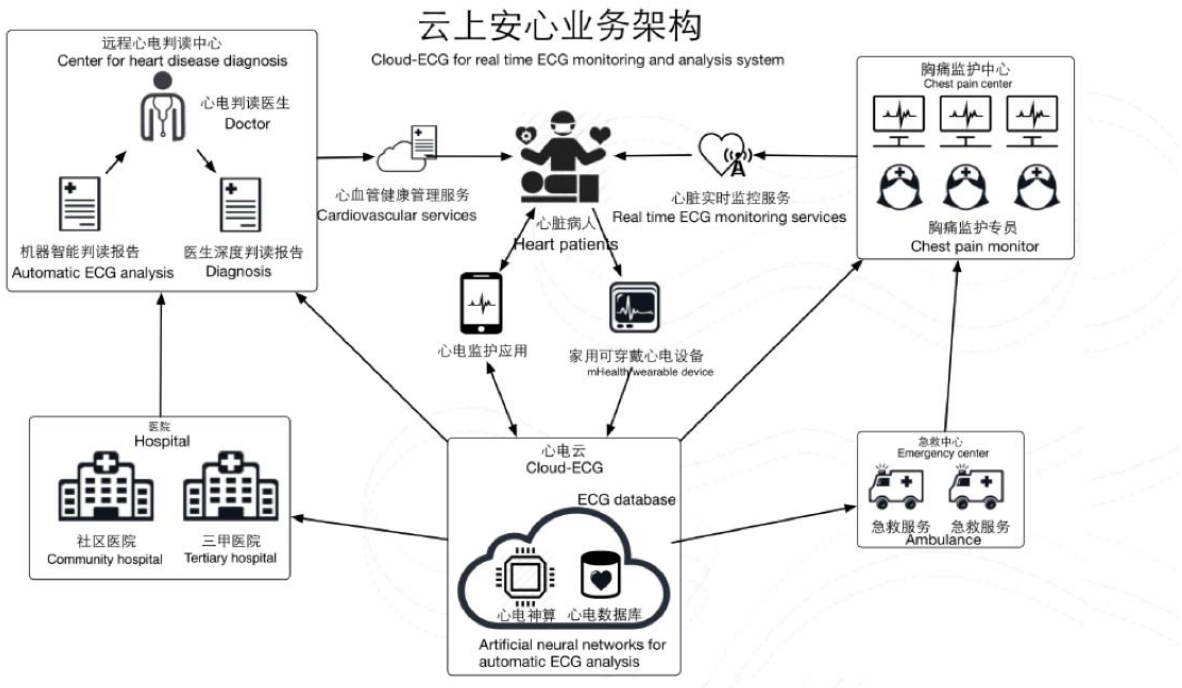
1. 首先是如何保证机器学习链路的连贯性，机器学习云平台的底层框架可以看作为黑盒，里面做了大量的数据以及模型应用的优化，比如为了方便算法平台生成的模型部署为在线服务，云服务平台可以自定义模型的表现形式。而开源框架想要实现整套的机器学习链路，可能需要采用多种开源服务，这些服务之间的关联可能会有性能奉献。但是基于开源架构自建，因为所有的底层服务都是自己实现，所以在功能延展性方面会有优势。

2. 在成本方面，需要考虑云服务价格以及自建集群运维的成本。自建集群的成本主要体现在集群硬件配置，扩容成本以及整个服务体系的软硬件开发人员成本。我认为自建服务最大的挑战可能是无法动态的伸缩计算资源来满足业务需求，举个例子，如果一家公司日均作业计算量需要50个节点，但是如果这家公司搞一个活动，可能突然需要80个节点的资源，那么就要考虑为了一次活动而对集群扩容是否划算。然而云服务通常是动态扩容的，就不会出现这样的顾虑。

3. 服务稳定性方面，自建环境如何达到云端服务的高可用性也是非常有挑战的，从计算集群的任务分配、周期性计算任务调度稳定性、以及在线服务的QPS这些方面都非常考验自建环境的性能。

如何在机器学习云服务上搭建业务

在本文为大家介绍如何通过机器学习云服务搭建一套云端的心脏病监控系统。首先，通过PAI生成心脏病预测模型，然后将模型部署为在线预测API，通过调用API实时的对用户的健康状况进行诊断。



下面看下整个具体流程：

数据集介绍

数据源： UCI开源数据集heart_disease

针对美国某区域的心脏病检查患者的体测数据，共303条数据。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：

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、填充缺失值、类型变换等操作。本次实验的输入数据包括14个特征和1个目标队列。需要解决的场景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，是分类问题。因为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型的数据。

输入数据展示：

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

我们看到有很多数据是文字描述的，在数据预处理的过程中我们需要根据每个字段的含义将字符型转为数值。

二值类的数据

二值类的比较容易转换，如sex字段有两种表现形式female和male，我们可以将female表示成0，把male表示成1。

多值类的数据

比如cp字段，表示胸部的疼痛感，我们可以通过疼痛的由轻到重映射成0~3的数值。

数据的预处理通过sql脚本来实现，具体请参考SQL脚本-1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};
```

特征工程

特征工程主要是包括特征的衍生、尺度变化等。本例中有两个组件负责特征工程的部分。

过滤式特征选择

主要是通过这个组件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，可以通过查看评估报告来显示最终的结果。

归一化

因为本次实验选择的是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。归一化的作用是将每个特征的数值范围变为0到1之间。归一化的公式为result=(val-min)/(max-min)。

归一化结果：

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...

模型训练和预测

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

拆分

首先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，在这里通过计算结果的阈值实现分类。具体的算法详情推荐大家在网上或者书籍中自行了解。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。

评估

通过混淆矩阵组件可以评估模型的准确率等参数，

通过此组件可以方便的通过预测的准确性来评估模型。

通过以上数据探索的流程我们可以得到以下的结论。

特征权重

我们可以通过过滤式特征选择得到每个特征对于结果的权重。

featname ▲	weight ▲
thalach	0.16569171224597157
oldpeak	0.14640697618779352
thal	0.13769166559906015
ca	0.11467097546217575
chol	0.10267709576600859
age	0.07876430484527841
trestbps	0.0772599125640569
slop	0.07702762609078306
restecg	0.015246832497405105
cp	0.0037507283721422424
exang	0
fbs	0
sex	0

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

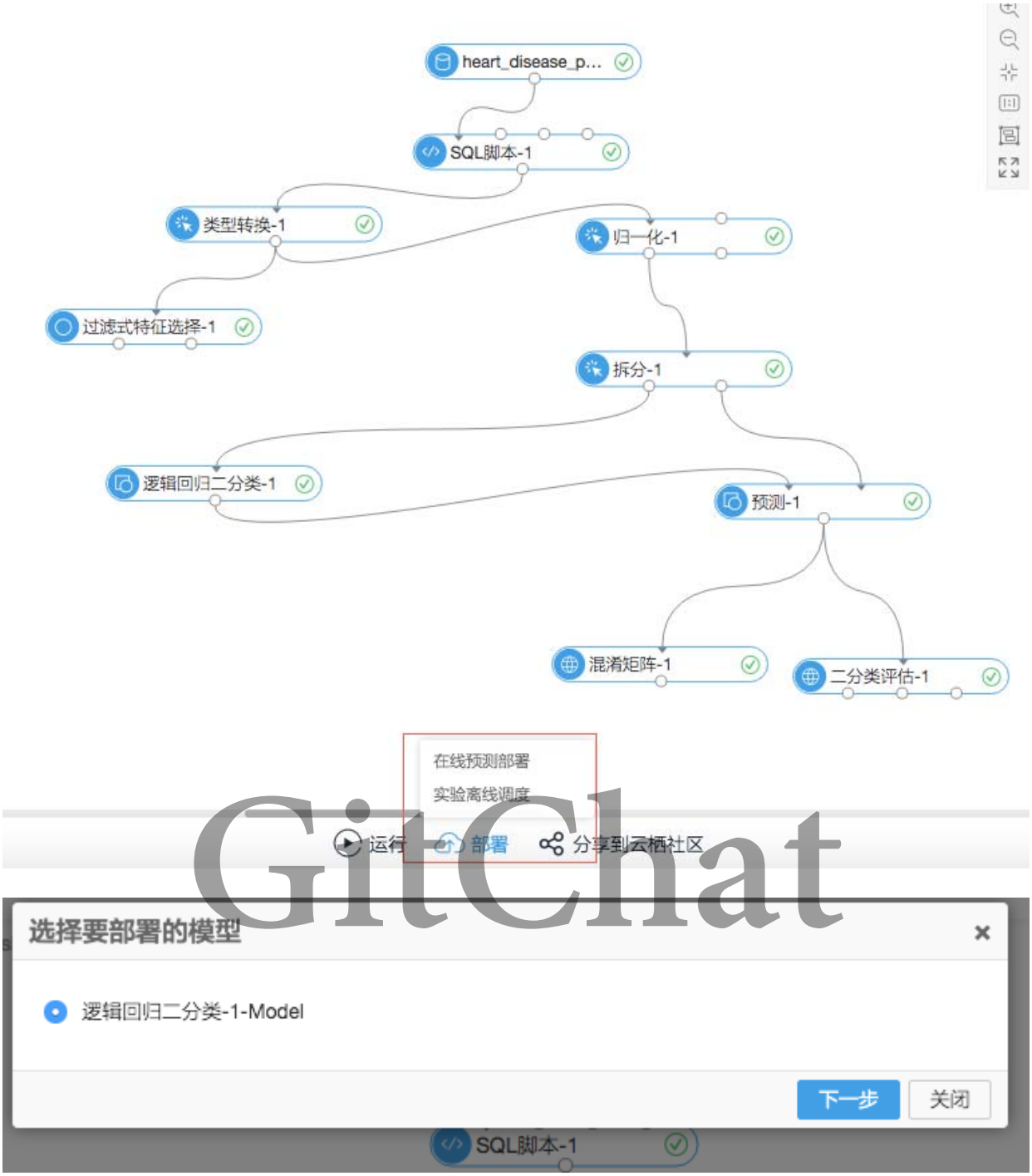
模型效果

通过上文提供的14个特征，可以达到百分之八十多的心脏病预测准确率。模型可以用来做预测，辅助医生预防和治疗心脏病。

模型部署为在线预测API

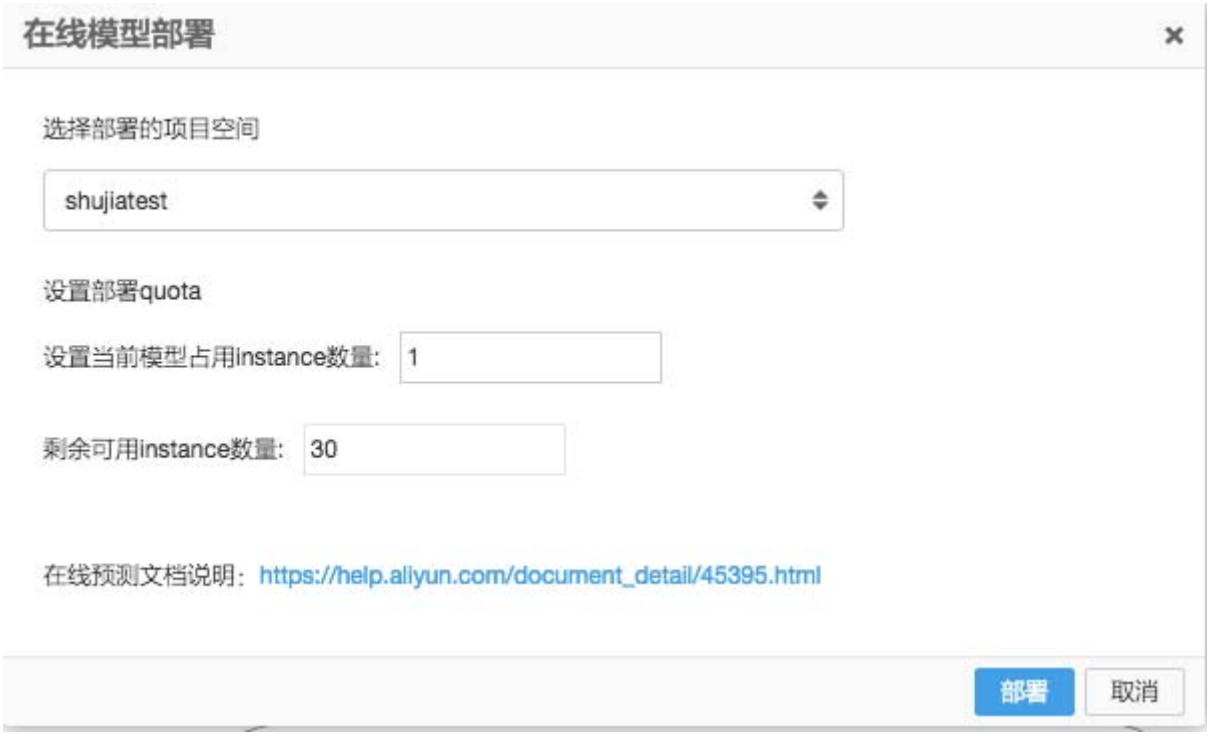
选择部署模型

我们以心脏病预测案例为例，实验生成一个逻辑回归模型，是用在线预测可以在当前实验点击“部署”按钮，选择“在线预测部署”。



配置模型部署信息

进入模型配置页：

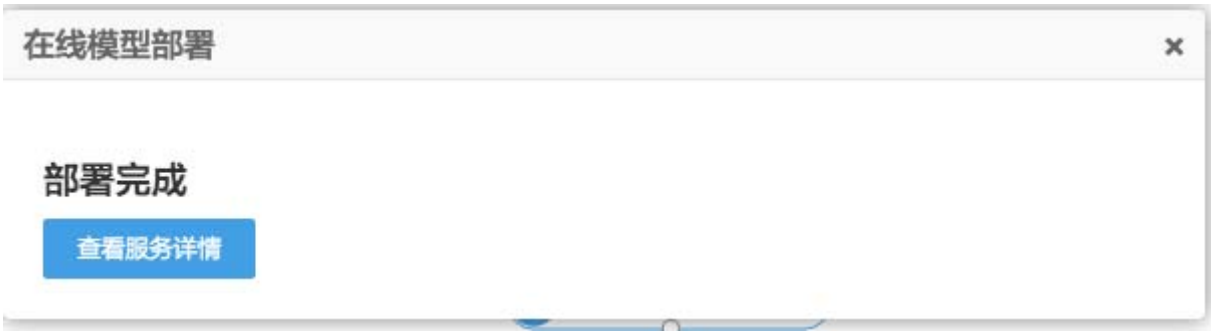


选择对应的项目空间，如果是第一次使用需要开通在线预测权限，权限申请是实时开通。下面详细解释instance的定义：

- 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2G内存。
- 单个模型的instance部署限制是[1,15]。

模型管控

模型部署完成可以进入如下界面进行管理，新部署模型可以在“查看模型详情”进行查看。

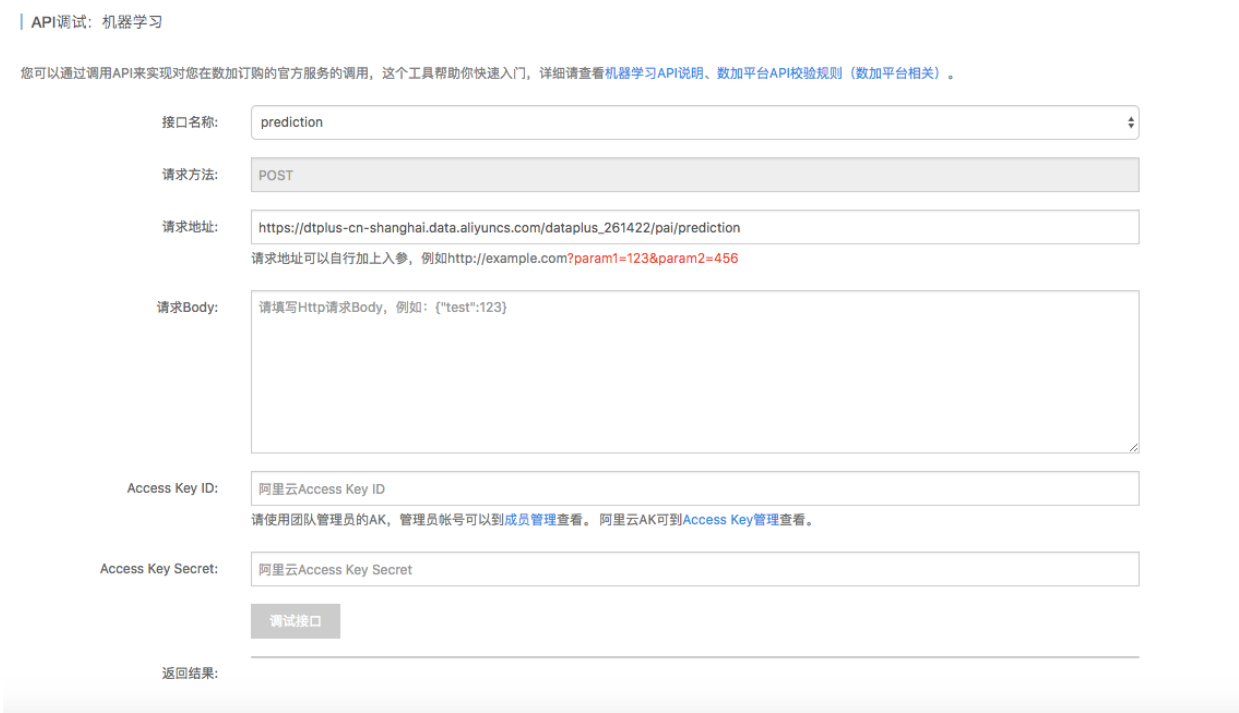


已经部署的模型可以在“已部署在线模型”里进行管理，

模型管理界面，版本表示的是同一模型多次部署的区分，通过下图红框可以拿到模型所在的项目和模型名称：

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，进入模型调试页面。



请求地址：https://dtplus-cn-shanghai.data.aliyuncs.com/dataplus_261422/pai/prediction/projects/

*project*名称/*onlinemodels*/模型名称

请求body为json串，以本文逻辑回归算法为例，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，常数列不用写。dataValue表示预测集对应特征的取值。dataType表示数值类型，dataType定义如下：

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

预测结果

现在我们已经配置好了服务，接下来只要编辑服务的body部分并且发送请求即可获得预测结果。我们假设用户的实时性别、血压、心跳波动等参数都是1，推送以下数据。

本案例body范例：

```
{
    "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
                }
            }
        ]
    }
```


可以获得返回，返回结果显示label为1（1表示用户患病，0表示健康），并且患病概率为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 - - - - -
```

应用案例

下面介绍在阿里内部基于机器学习云服务的三个应用场景。

应用一：推荐系统

第一个应用是推荐系统，主要是参数服务器在推荐系统内的应用。当在淘宝购物时，经查询显示的商品一般都是非常个性化的推荐，它是基于商品的信息和用户的个人信息以及行为信息三者的特征提取。这个过程中形成的特征一般都是很大，在没有参数服务器时，采用的是MPI实现方法，MPI中所有的模型都存在于一个节点上，受限于自身物理内存上限，它只能处理2000万个特征；通过使用参数服务器，我们可以把更大模型（比如说百亿个特征的模型），分散到数十个乃至于上百个参数服务器上，打破了规模的瓶颈，实现了模型性能上的提升。

应用二：芝麻信用分

第二个应用是芝麻信用分。芝麻信用分是通过个人的数据来评估你个人信用。做芝麻信用分时，我们将个人信息分成了五大纬度：身份特质、履约能力、信用历史、朋友圈状况和个人行为进行评估信用等级。在去年，我们利用DNN深度学习模型，尝试做芝麻信用分评分模型。输入是用户原始的特征，基于专家知识将上千维的特征分为五部分，每部分对应评分的维度。我们通过一个本地结构化的深度学习网络，来捕捉相应方面的评分。由于业务对解释性的需求，我们改变了模型的结构，在最上面的隐层，一共有五个神经原，每个神经原的输出都对应着它五个维度上面值的变化；再往下一层，是改变维度分数的因子层；用这种本地结构化的方式，维持模型的可解释性。

应用三：光学文字识别

最后一个应用是图象上面的光学文字的识别（OCR）。我们主要做强模板类、证件类的文字识别，以及自然场景下文字的识别。强模板服务（身份证识别）在数加平台上也提供了相应的入口，目前可以达到身份证单字准确率99.6%以上、整体的准确率93%。在识别中用到的是CNN模型，但其实整个流程特别长，不是深度学习一个建模就能解决的问题，包括版面分析、文字行的检测、切割等等技术。在CNN训练中，我们采用了多机多卡分布式算法产品，之前利用一千万个图像训练CNN模型，大约需要耗时70个小时，迭代速度非常缓慢；采用分布式8卡产品之后，不到十个小时就可以完成模型训练。目前OCR的服务已经成为了一个受欢迎的阿里云市场上的API，尤其是证件类的识别，准确率高，种类齐全，成为了一种可以在商业场景中广泛使用的数据服务。