

# 机器学习模型服务

## 1. 作业简述

仿照 DaaS 平台，提供模型部署上线一站式解决方案。

## 2. 模型服务平台 DaaS

2.1. 导入模型。项目创建成功后，进入项目主页，切换到模型标签页，点击命令导入模型。

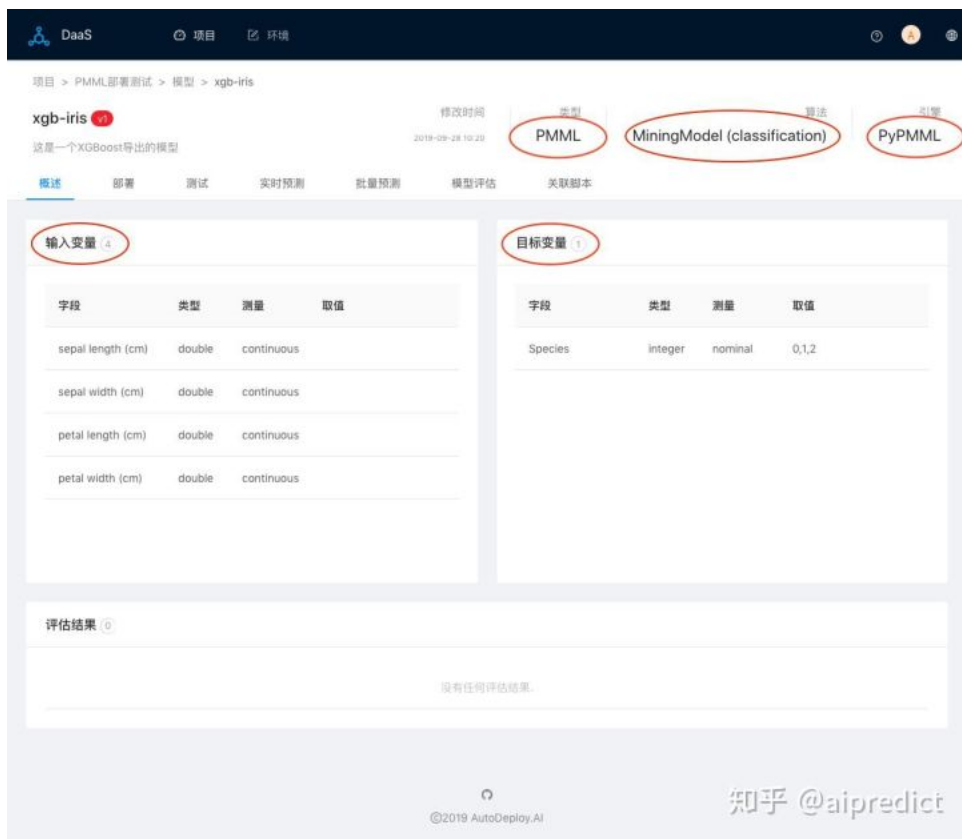


2.2. 选择要部署的 PMML 模型文件，点击此处可下载当前使用的 PMML 模型 xgb-iris.pmml。在该流程中，首先会对模型进行验证，如果模型不是一个有效的 PMML，会导致添加失败，DaaS 将返回错误信息。

导入模型

The screenshot shows the 'Import Model' form. It has a title '导入模型'. The form contains several fields: '名称' (Name) with the value 'xgb-iris', '描述' (Description) with the value '这是一个XGBoost导出的模型', '类型' (Type) with a dropdown menu showing 'PMML' (circled in red), and '文件' (File) with a file upload area showing a folder icon and the filename 'xgb-iris.pmml'.

**2.3. 模型概述。**PMML 导入成功后，进入模型主页（概述），显示了模型的基本信息，比如输入和目标变量、模型类型、使用算法、运行引擎等。

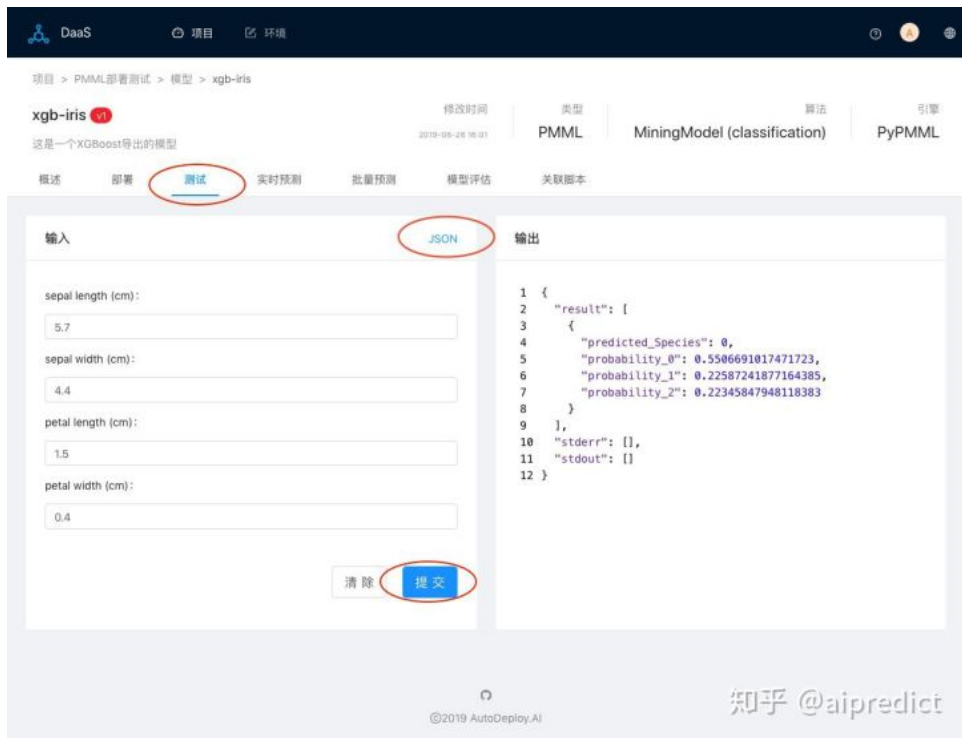


The screenshot shows the 'Overview' page for a PMML model named 'xgb-iris'. The page header includes navigation links for '项目' (Project) and '环境' (Environment). The model details section shows the model name, a red 'v1' version indicator, the modification time '2019-09-28 10:29', and the model type 'PMML', algorithm 'MiningModel (classification)', and engine 'PyPMML'. Below this, there are tabs for '概述' (Overview), '部署' (Deploy), '测试' (Test), '实时预测' (Real-time Prediction), '批量预测' (Batch Prediction), '模型评估' (Model Evaluation), and '关联脚本' (Associated Scripts). The 'Overview' tab is active, showing two tables: '输入变量' (Input Variables) with 4 variables and '目标变量' (Target Variable) with 1 variable. The '评估结果' (Evaluation Results) section shows '没有任何评估结果' (No evaluation results). The footer includes the copyright '©2019 AutoDeploy.AI' and the text '知乎 @aipredict'.

字段	类型	测量	取值
sepal length (cm)	double	continuous	
sepal width (cm)	double	continuous	
petal length (cm)	double	continuous	
petal width (cm)	double	continuous	

字段	类型	测量	取值
Species	integer	nominal	0,1,2

**2.4. 测试模型。**切换到测试标签页，通过表单输入数据或者点击 JSON 命令直接输入 JSON 格式的数据，然后点击提交命令，等待预测结果的返回



The screenshot shows the 'Test' page for the 'xgb-iris' model. The 'Test' tab is selected, showing input fields for 'sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', and 'petal width (cm)'. A 'JSON' button is visible next to the input fields. At the bottom, there are '清除' (Clear) and '提交' (Submit) buttons. The output section shows a JSON response with predicted species and probabilities. The footer includes the copyright '©2019 AutoDeploy.AI' and the text '知乎 @aipredict'.

```
1 {
2   "result": [
3     {
4       "predicted_Species": 0,
5       "probability_0": 0.5506691017471723,
6       "probability_1": 0.22587241877164385,
7       "probability_2": 0.22345847948118383
8     }
9   ],
10  "stderr": [],
11  "stdout": []
12 }
```

## 2.5. 添加 Web 服务。当模型测试成功后，切换到部署标签页，点击命令添加服务



## 2.6. 测试 Web 服务。部署创建成功后，进入部署页面（概述），可以看到一个副本，状态是启动中，等待状态变成运行中后，该部署才算创建完成。这时候就可以接受预测请求。切换到测试标签页，在请求正文中输入测试数据，测试该服务。



**2.7.** DaaS 为部署服务提供标准的 REST API，可以通过任意 REST 客户端来调用，方便生产环境的集成。点击生成代码命令，会生成通过 curl 调用 REST API 的命令参数。

```
bash-3.2$ curl -k -X POST \
> https://192.168.64.3:30931/api/v1/svc/pmm1/xgb-iris-svc/predict \
> -H 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ1aWQ0IjEwMDA5InVzZSxJYWI1IjoieWRtaW4iLCJyb2x1IjoieWRtaW4iLCJwcm9qZWN0TmFtZSI6IjBNTUxdTkZlZW50IiwiaWF0IjoxNTYyOTc3NDYifQ.u2FK82T4ea2sQQz40M6bVxcZhBdMqIEB98Y6vNtjOJTUPPV_eX-QG5V0oV3uoexufHcc2swLc0IGAackptoyBUqKs3SYJDPys-tJU37_0KmtEDHIlimDI0jmKY1LV3pUP2ij9alMDAaoofrdyswf82e8F4fR4XXFKvwUZMvBBK3lm60tyrFy8qhrF_UmwjK_ZA7w0qPLVsloTV6A_LsI4P3Z-R4GeDziwtL3LKARcnM2EowTeGt1tIlZMKrzEYBRgdi0M-K-WVPvQa2aBENlUWpnHqbG914x11VA5Lw2KQms5oEwnbkuzshrv-VIAcwBpb6mw-vPq-TdjqqKE90w' \
> -H 'Cache-Control: no-cache' \
> -H 'Content-Type: application/json' \
> -d '{"args":{"X":{"sepal length (cm)":5.7,"sepal width (cm)":4.4,"petal length (cm)":1.5,"petal width (cm)":0.4}}}'
{"result":{"predicted_Species":0,"probability_0":0.5506691017471723,"probability_1":0.22587241877164385,"probability_2":0.22345847948118383}}
```

**2.8.** 返回 DaaS，切换到部署模型标签页。指标页面显示 Web 服务性能指标：执行次数、平均响应时间、最大最小时间等。可以看到，我们执行了二次调用：第一次通过 DaaS 部署测试界面，第二次通过在 Shell 中执行 curl 命令。一般来说，第一次调用是要慢一些，后面就会快很多。

The screenshot displays the 'xgb-iris-svc' service page in the AutoDeploy AI console. The breadcrumb navigation at the top reads '项目 > PMML部署测试 > 部署 > xgb-iris-svc'. The service name 'xgb-iris-svc' is prominently displayed. Below it, the endpoint is listed as '端点: POST https://192.168.64.3:30931/api/v1/svc/pmml/xgb-iris-svc/predict' and the deployment command is '部署令牌: [icon]'. On the right, a table shows the service's configuration: '类别' (Category) is '网络服务' (Network Service), '类型' (Type) is '默认实时预测' (Default Real-time Prediction), '对象' (Object) is 'xgb-iris v1', and '创建时间' (Creation Time) is '2019-09-28 17:27'. Below this, another table shows resource usage: 'CPU核数' (CPU Cores) is '1' and '内存(GB)' (Memory) is '1'. The '概述' (Overview) tab is selected, showing a table of metrics. The '指标' (Metrics) section is circled in red. The metrics table has columns for '函数名' (Function Name), '访问次数' (Access Count), '平均响应时间(ms)' (Average Response Time), '中间响应时间(ms)' (Median Response Time), '最小响应时间(ms)' (Minimum Response Time), '最大响应时间(ms)' (Maximum Response Time), '首次访问时间' (First Access Time), and '最新访问时间' (Latest Access Time). The data row shows 'predict' with 2 accesses, an average response time of 205.0 ms, and a latest access time of 2019-09-28 17:31:40. The '副本' (Replicas) section shows 1 replica with the name 'd-pmml-xgb-iris-svc-5954487d5b-zbsjn' in a '运行中' (Running) state.

项目 > PMML部署测试 > 部署 > xgb-iris-svc

## xgb-iris-svc

端点: **POST** <https://192.168.64.3:30931/api/v1/svc/pmml/xgb-iris-svc/predict>

部署令牌: [icon]

类别	类型	对象	创建时间
网络服务	默认实时预测	xgb-iris v1	2019-09-28 17:27

CPU核数	内存(GB)
1	1

**概述** 测试

**指标**

函数名	访问次数	平均响应时间(ms)	中间响应时间(ms)	最小响应时间(ms)	最大响应时间(ms)	首次访问时间	最新访问时间
predict	2	205.0	205.0	12.0	398.0	2019-09-28 17:30:59	2019-09-28 17:31:40

**副本** ①

名称	状态	操作
d-pmml-xgb-iris-svc-5954487d5b-zbsjn	运行中	[icon]

©2019 AutoDeploy.AI

知乎 @aipredict

### 3. 技术难点

本任务的主要难点在于模型的部署，支持模型服务的暂停、启动、删除，并且对这些部署后的模型进行统一的管理，这其中常用的技术栈包括 docker+k8s。

## 4. 评分说明

以下功能若有不理解的可结合最后两个参考链接里面的示例思考，若仍旧不理解可联系助教处理。

- 基本功能（100 分）
  - 上传模型（至少包括 pmml 与 onnx 格式），并且上传后能查看模型信息：类似 DaaS 平台的信息，包括输入（字段名、类型、取值[如有]、维数[如有]）和目标变量（15 分）
  - 能够测试模型，通过表单输入数据（表单项可能是文本也可能是文件，类似 postman）或者使用 JSON 命令直接输入 JSON 格式的数据，提交后预测结果会显示在界面中（10 分）
  - 部署模型，对外提供 restful api 接口进行调用，支持模型服务的暂停，启动，删除，并且显示当前服务的状态（25 分）
  - 部署模型后可在前端测试部署的接口，类似于测试模型的界面（使用快速返回接口测试，解释见下）（25 分）
  - 对外提供的 api 接口分两种：快速返回与等待返回，快速返回接口接受 json 格式的输入，一次处理一条数据（比如一张图片、一段视频、一段文本或者一些字段与取值），直接返回预测结果；等待返回接受批量的数据，比如 zip 文件包（里面是图片集、文本文件）、或者 csv 格式的文件，先返回任务 id（立即返回），之后可通过任务 id 查找结果。（25 分）
- 加分项
  - 验证模型文件的有效性。（5 分）
  - 预留 CPU 和预留内存：为了降低系统的不稳定风险，用户可以选择为部署分配指定的 CPU 核数和内存量。（10 分）
  - 创建自定义实施预测脚本，预处理输入；可进行 API 测试并最终可部署。（20 分）
  - 扩展模型格式，不仅仅包括 pmml 与 onnx。（酌情加分）
  - 能够监控模型服务，比如显示 Web 服务性能指标：执行次数、平均响应时间、最大最小时间等。（5 分）
  - 任务管理查看界面。（5 分）
  - 模型服务的伸缩（控制负载均衡，支持高并发请求）。（20 分）
- 其他要求：
  - 禁止上传不符合国家相关法律法规、攻击侮辱他人或其他对他人造成困扰的内容；上传富文本内容不得包含可执行js 代码来进行xss 攻击；
  - 每位同学应妥善保管自己的密码，不得交予他人使用；发现上述行为，本次综合实验计 0 分。

## 5. 提交说明

小组提交的作业文件夹需包含下面 2 个目录：

- - src：代码目录，目录下包括所有实现的代码文件。
- - doc：项目文档目录，目录下包括所有项目文档。

其他目录可根据需要自行添加。提交时请将作业文件夹压缩后提交压缩包到网络学堂。

## 6. 参考链接

<https://docs.docker.com/> docker 文档

<https://kubernetes.io/zh-cn/docs/home/> kubernetes 中文文档

<http://nginx.org/> nginx 文档

<https://www.autodeploy.ai/cn/> DaaS 官网

<https://www.jianshu.com/p/552fa06415a5> 自动部署 PMML 模型生成 REST API

<https://www.jianshu.com/p/c1e0efe6482f> 使用 ONNX 部署深度学习和传统机器学习模型