第一章：.pyc文件封装流程

软件准备：VS code ,建立 “调用pyc示范”文件夹

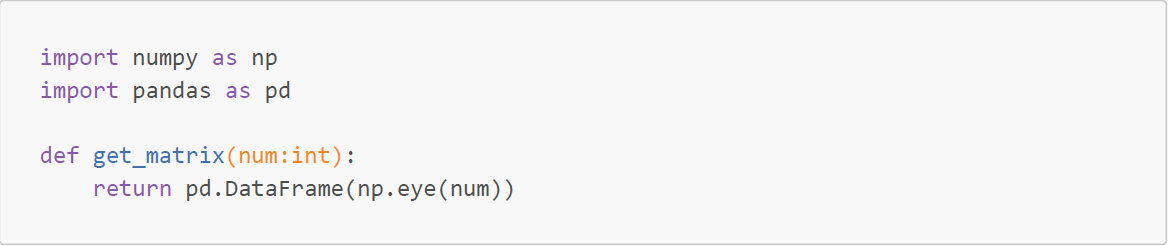
1、编写源代码

1.1 有class时，按照下图模式进行编写：



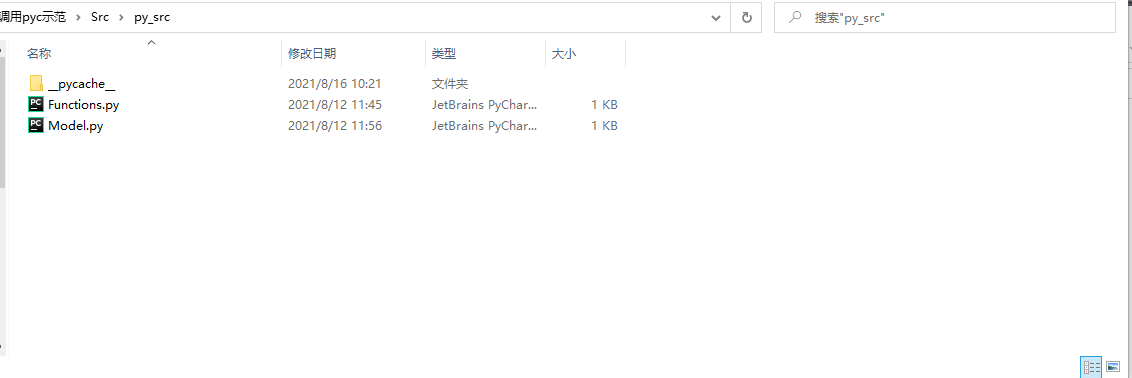
图 有class模式

1.2 无class时，按照图2模式进行编写：



图无class模式

2、.py源码存放在…\调用pyc示范\Src\py\_src”文件夹下



3、将‘py\_src’文件夹下的.py源码编译为.pyc文件，有两种方式：（1）使用代码python-m src.py直接进行编译，该方法只能进行单个.py文件编译；（2）使用compile模型批量进行编译。



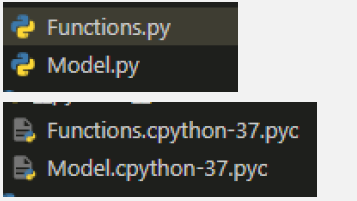
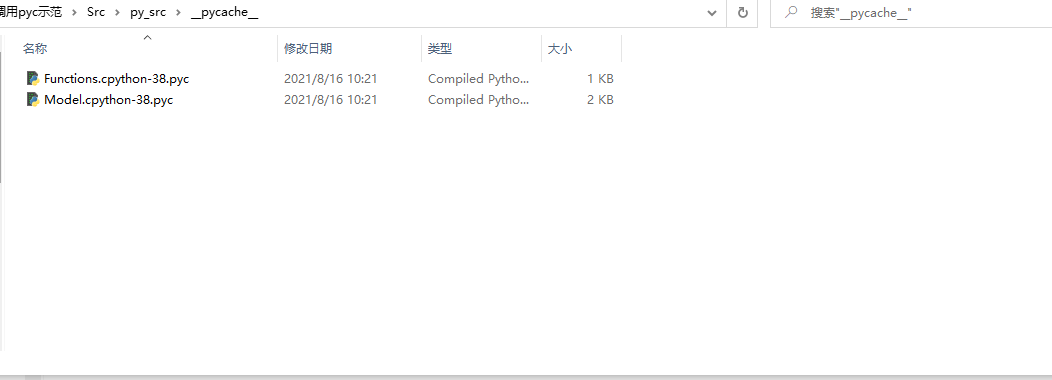
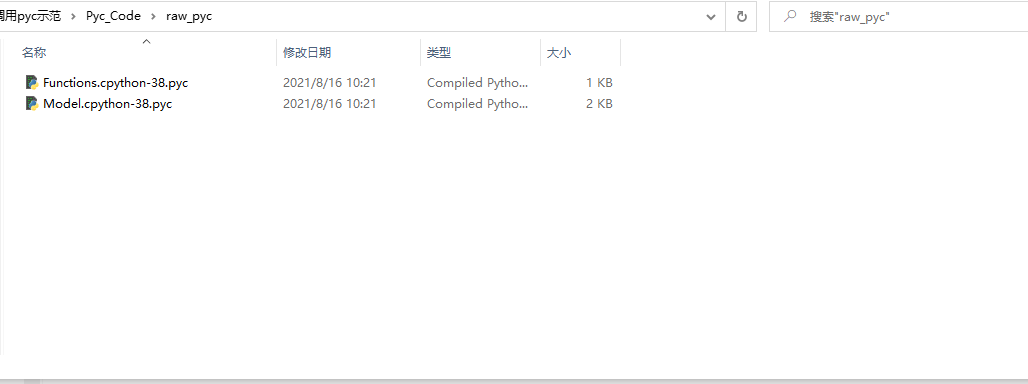


图 批量编译py文件

4、编译后的.pyc文件存放在“…\调用pyc示范\Src\py\_src\\_\_pycache\_\_”文件夹下：



5、将编译后的.pyc文件复制到“…\调用pyc示范\Pyc\_Code\raw\_pyc”文件夹下



6、将.pyc文件的默认名字修改为与源码文件名相同，有两种方式：（1）直接将.pyc文件的默认名字修改为与源码文件名一致；（2）使用compile模块批量修改默认的.pyc文件的名字，将修改名字之后的.pyc文件存放在“…\调用pyc示范\Pyc\_Code”文件夹下：

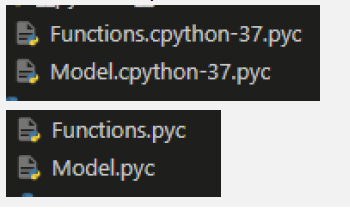
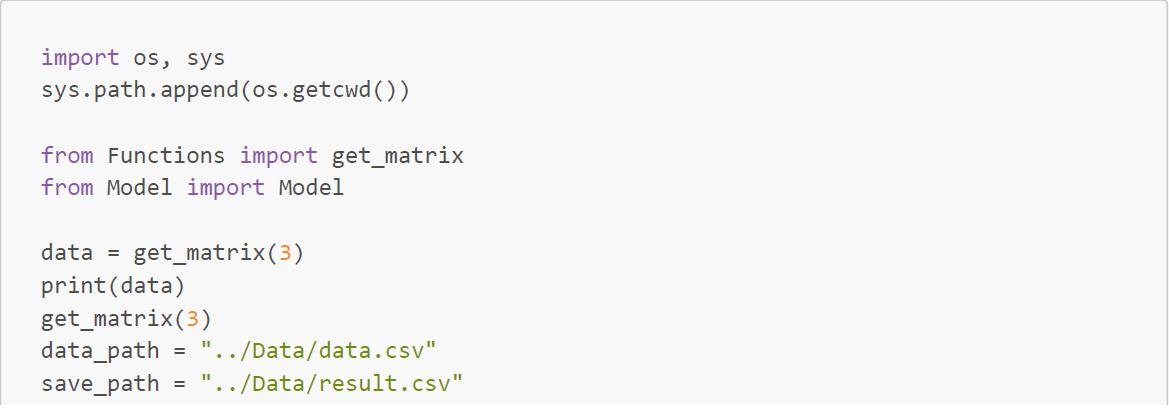




图4 修改.pyc文件名

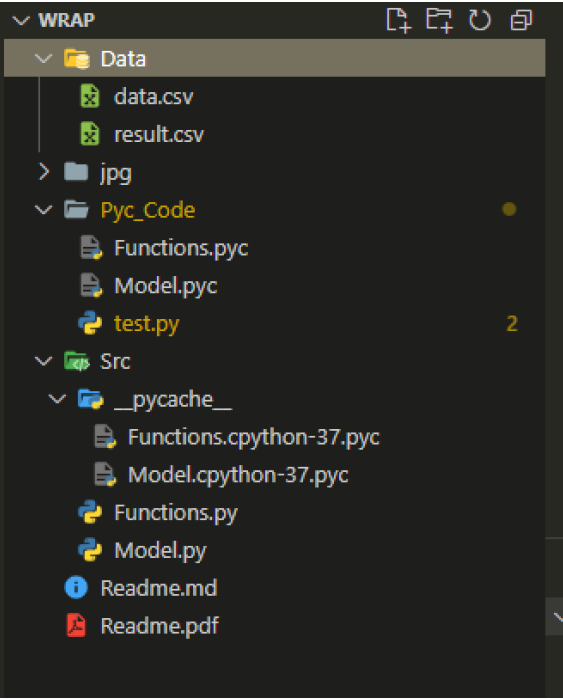
7、调用模块：在最终的test.py文件中对.pyc文件模块进行调用，与调用第三方py文件无异



图调用.pyc文件

8、flask封装： flask相关代码只与最终的调用代码test.py进行交互，与原来.py文件封装规范无异，具体封装规范见第二章。

9、总体代码结构



第二章 .py文件封装规范

* 准备工作：

1. 需要封装的算法源码，以封装某项目某产线质量评估代码为例；
2. flask微服务框架，主要包括公共脚本文件api.py、config.py、errors.py、main.py、utils.py配置文件logging.conf，以及服务调用框架脚本文件；
3. 本机必须安装封装算法源码所使用的第三方包，例如pandas、sklearn等，本机必须安装flask微服务框架所需拓展包，flask、flask\_restful；

* 将源码放入flask微服务调用框架脚本文件：

以封装某项目某产线质量评估代码为例，flask微服务调用框架脚本文件为JXXQualityEvaluation.py和QualityEvaluation.py，其中QualityEvaluation.py文件主要包括四部分，第一部分为第三方、拓展包导入以及配置文件配置信息读入，必须将源码所需第三方包导入，形式以 import +拓展包名称即可，脚本文件自带flask\_restful、logging拓展包导入以及配置文件配置信息读入。

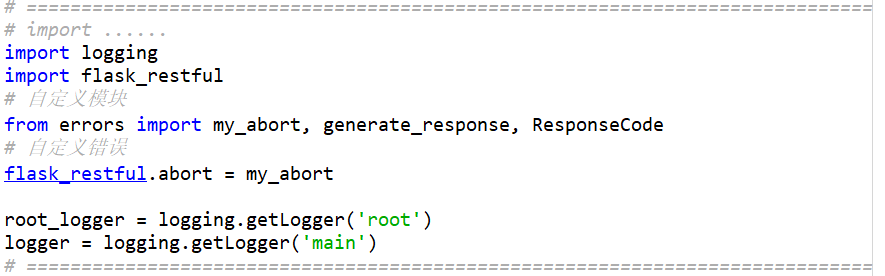


图 QualityEvaluation 第一部分

第二部分为封装算法源码，算法源码可以以函数、对象形式放置，以QualityEvaluation.py为例，部分源码如下，该代码以函数放置在flask微服务调用框架脚本文件QualityEvaluation代码最后即可。注意封装源码文件中不允许出现相对或绝对路径信息，整体代码使用由main.py进行操作，参数传入以算法路由操作。

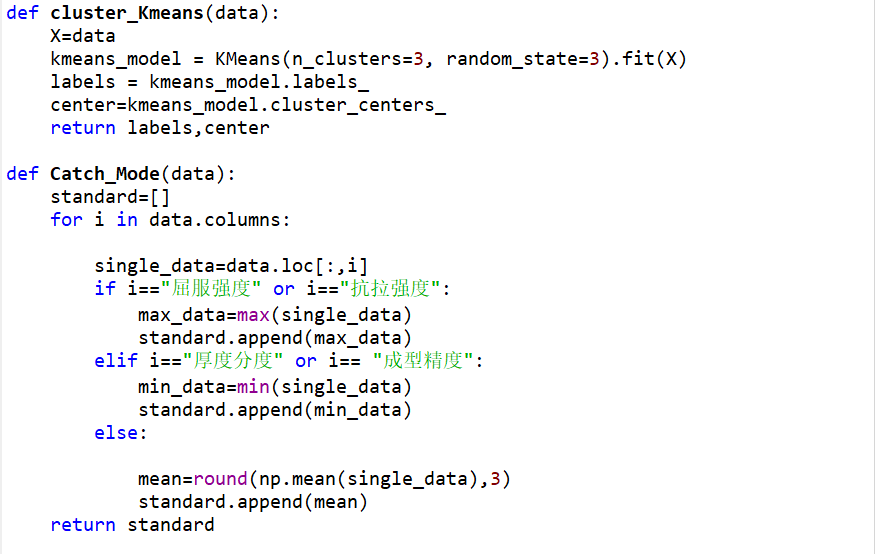


图 QualityEvaluation 第二部分

第三部分主要调用所封装算法源码，以函数形式进行，代码如下，函数形参可以是处理文件路径、也可以为python基本变量，如int，float类型，本例为路径。该函数返回值return内容将反应在web端界面，格式为json。

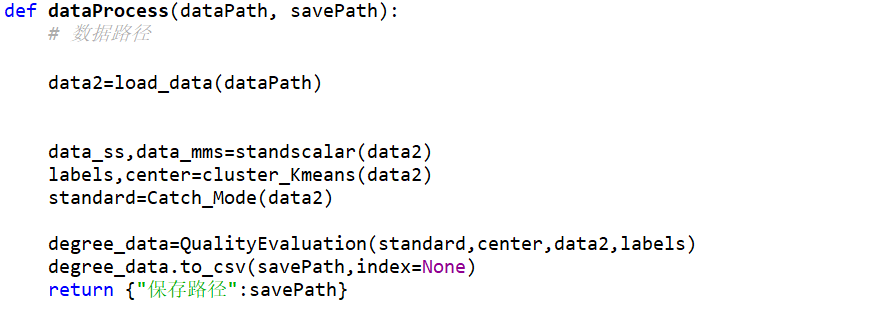


图 QualityEvaluation 第三部分

第四部分为两个函数function和function\_qs,两函数为框架内容，已写完整，函数形参与第三部分形参一致，同时注意两函数采用异常捕获方式编码，在try内容中需要对应形参内容，保持一直，logger信息为日志信息，根据所封装内容调整。



图 QualityEvaluation 第四部分

* 定义从路由中需要获取的参数及其类型

flask微服务调用框架脚本文件为JXXQualityEvaluation.py主要内容分为两部分，第一部分为第三方、拓展包导入，文件中需要的包基本写好，仅需将上文QualityEvaluation文件中function函数导入，导入格式 from 相对路径文件名 import 函数名。相对路径以main.py文件存放为起点。

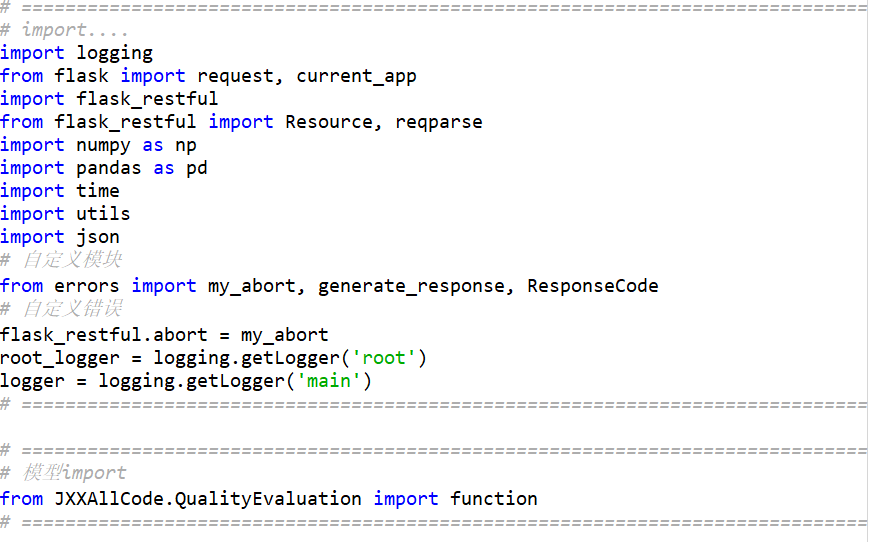


图 JXXQualityEvaluation第一部分

第二部分整体为两个类，第一个类，继承flask\_resful中内容，其中该对象名称需要与该脚本文件名称一致，同时对象中定义的初始化参数与QualityEvaluation中dataProcess的形参一致，函数get中参数也必须与QualityEvaluation中dataProcess的形参一致。

第二个类继承json.JSONEncoder，主要对一些数据类型进行转换，不需要改动，若项目需要，自行添加。



图 JXXQualityEvaluation第二部分

* api编程接口文件修改

该部分内容主要修改编程接口文件api.py,该文件内容主要为两部分，一是框架拓展包、路由地址调用，主要格式为from 相对路径文件名 import 类名，其中文件名和类名一致。相对路径以main.py文件存放为起点。

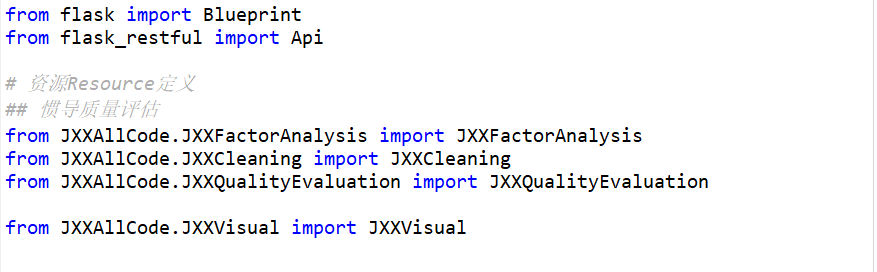


图 api第一部分

继续为api添加资源，内容为文件名以及相对路径文件名

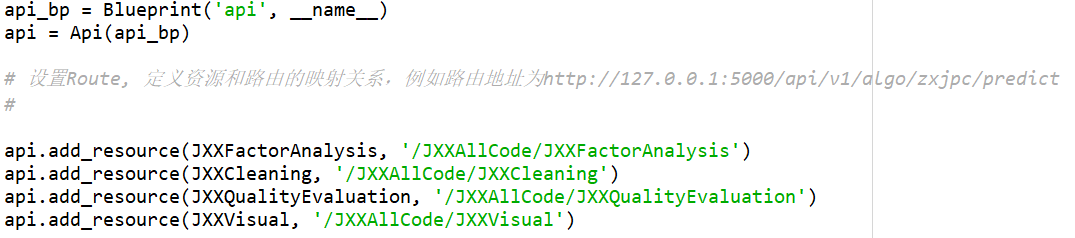


图 api第二部分

* 服务启动

运行公共脚本文件main.py,显示结果如下说明启动成功

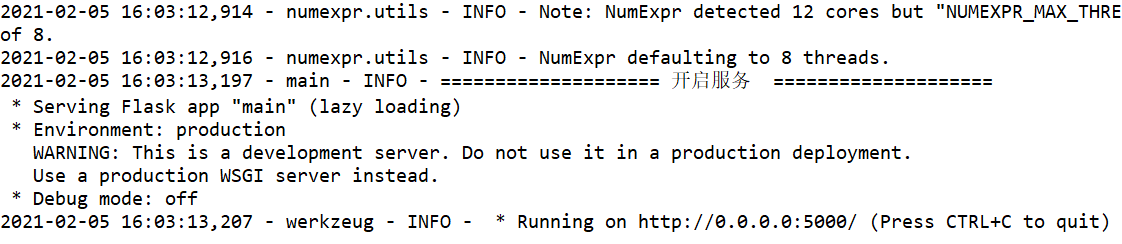


图 服务启动成功

测试封装代码，需要使用浏览器，建议谷歌、edge浏览器，网址中输入算法路由，格式如下：

<http://ip:port/api/algo/v1/文件路径/>文件名称?输入路径（输入参数）=???&输出路径（输出参数）=???

算法与平台的通信协议，总是使用 HTTP 协议，以确保交互数据的传输安全。

ip为本机IPV4地址，例如192.168.1.102；

port 默认为5000；

输入路径和输出路径名称与第二部dataProcess的一致，多个路径依次续写，以以某项目某产线质量评估代码为例，在网址中输入内容

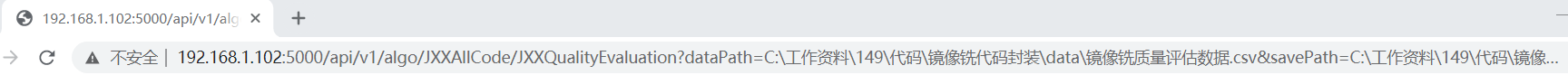


图 网址内容

页面反馈内容则说明成功



图 成功反馈

若失败，则可以根据页面反馈失败内容进行debug，如



图 失败反馈

* 封装debug

该部分为封装人员调试代码、以及修改flask服务进行说明。

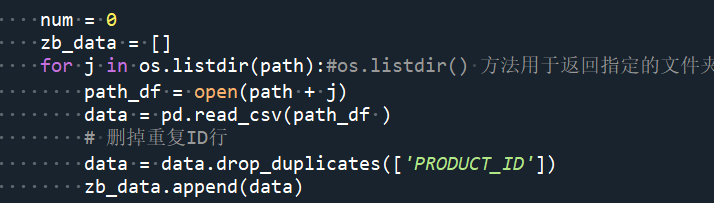
如发生错误，需要调试，flask微服务框架带有development | production | testing模式，其中development为开发者模式，将开启debug，开启服务后修改代码，服务会刷新并做出反馈，修改方式为main.py文件如下图



修改为如下图



* 注意事项
* 代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。正确的英文拼写和语法可以让算法调用者便于理解guanlian/julei/fenlei 等中心熟悉算法名，可视为英文，英文算法名统一使,lowerCamelCase 风格，需遵从驼峰形式，如 localValue/nputUserId。
* 结果输出文件应该以csv数据文件格式进行输出，不能以图片格式进行输出。
* 若结果输出文件较多，为避免浏览器输入地址有字节长度的限制而无法完整输入的情况，应选择采用批量读取当前文件夹下的内容的格式进行读取。如图：



* 封装完成后代码文件应该包括以下内容：



其中图片样式应给出输出的数据文件要以何种图形（折线图，柱状图等等）来展示。