# 程序部署运行说明

### 程序一共包含以下代码：

|  |  |
| --- | --- |
| 程序名 | 作用 |
| dataset\_provider.py | 提供数据的程序。从平台或者当前所在文件夹提取原始数据，通过api把数据逐一发送给middle.py |
| middle.py | 对数据库进行操作的程序。与一个mongodb数据库连接，对外开放api，根据api传入的内容对数据库进行增删查操作 |
| feature\_get.py | 特征提取程序。从middle.py处请求得到清洗后的数据，提取其特征，然后向middle.py返回特征，数据存入数据库 |
| aggregate.py | 特征数据合并程序。从middle.py处请求得到所有特征数据，整合成一个单一表格返回给middle.py，数据存入数据库 |
| test.py | 用模型对数据进行分类判断的程序。从同目录下的model文件夹中提取训练好的模型cwru.model，用该模型对从middle.py处请求得到的特征数据总表进行分类，将分类结果返回middle.py，并本地生成一个result\_日期.csv文件 |
| api\_forward.py | 向前端微信小程序提供接口的程序。监听来自微信小程序的请求，根据请求从middle.py处调用所需的数据并进行整合，再返回给微信小程序 |

### 这些代码分别放在code文件夹内的几个文件夹下，整个code文件夹可以作为一个pycharm project打开

### 运行说明：

1. **所需环境：**

# packages in environment at /root/anaconda3/envs/env1:

#

# Name Version Build Channel

\_libgcc\_mutex 0.1 main

asn1crypto 1.3.0 py37\_0

attrs 19.3.0 py\_0

blas 1.0 openblas

ca-certificates 2020.1.1 0

certifi 2020.4.5.1 py37\_0

cffi 1.14.0 py37h2e261b9\_0

chardet 3.0.4 py37\_1003

click 7.1.1 py\_0

cryptography 2.8 py37h1ba5d50\_0

flask 1.1.2 py\_0

idna 2.9 py\_1

imbalanced-learn 0.3.1 pyh2cb239c\_0 glemaitre

importlib\_metadata 1.5.0 py37\_0

intel-openmp 2020.0 166

itsdangerous 1.1.0 py37\_0

jinja2 2.11.1 py\_0

jsonschema 3.2.0 py37\_0

ld\_impl\_linux-64 2.33.1 h53a641e\_7

libedit 3.1.20181209 hc058e9b\_0

libffi 3.2.1 hd88cf55\_4

libgcc-ng 9.1.0 hdf63c60\_0

libgfortran-ng 7.3.0 hdf63c60\_0

libopenblas 0.2.20 h9ac9557\_7

libstdcxx-ng 9.1.0 hdf63c60\_0

markupsafe 1.1.1 py37h7b6447c\_0

mkl 2020.0 166

mkl-service 2.3.0 py37he904b0f\_0

ncurses 6.2 he6710b0\_0

numpy 1.15.0 py37h2aefc1b\_0

numpy-base 1.15.0 py37h7cdd4dd\_0

openssl 1.1.1f h7b6447c\_0

pandas 1.0.3 py37h0573a6f\_0

pip 20.0.2 py37\_1

pycparser 2.20 py\_0

pymongo 3.9.0 py37he6710b0\_0

pyopenssl 19.1.0 py37\_0

pyrsistent 0.16.0 py37h7b6447c\_0

pysocks 1.7.1 py37\_0

python 3.7.7 hcf32534\_0\_cpython

python-dateutil 2.8.1 py\_0

python\_abi 3.7 1\_cp37m conda-forge

pytz 2019.3 py\_0

pywavelets 1.1.1 py37h03ebfcd\_1 conda-forge

readline 8.0 h7b6447c\_0

requests 2.23.0 py37\_0

scikit-learn 0.19.1 py37\_nomklh6cfcb94\_0 [nomkl]

scipy 1.1.0 py37\_nomklh9c1e066\_0 [nomkl]

setuptools 46.1.3 py37\_0

six 1.14.0 py37\_0

sqlite 3.31.1 h7b6447c\_0

tk 8.6.8 hbc83047\_0

urllib3 1.25.8 py37\_0

werkzeug 1.0.1 py\_0

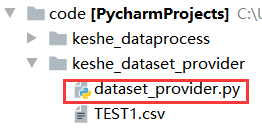
wheel 0.34.2 py37\_0

xz 5.2.5 h7b6447c\_0

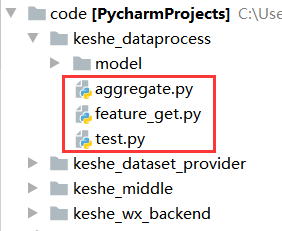
zipp 2.2.0 py\_0

zlib 1.2.11 h7b6447c\_3

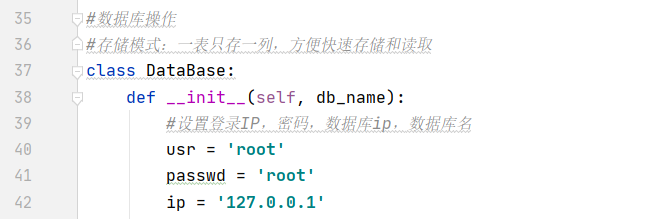
1. **不建库本地运行：**
2. 将code文件夹作为一个pycharm项目打开
3. 进入keshe\_dataset\_provider文件夹下，运行dataset\_provider.py



1. 然后进入keshe\_dataprocess文件夹下，依次运行feature\_get.py，aggregate.py，test.py

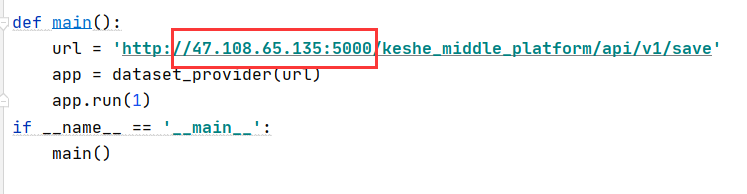


1. **完全本地运行：**
2. 安装mongodb，进入mongodb，在admin表下设置用户root，密码root，赋予root权限
3. 运行middle.py（如果mongodb与middle.py不在同一台计算机上，或者用户名密码不同，则还需修改middle.py的以下部分代码）



middle.py启动后，将在本机的5000端口进行监听

1. 修改其余所有python代码中设置的ip。每个程序的ip都在程序末尾的main函数中定义，如下所示：



1. 运行其余所有python代码，代码可以不放在同一台机器上，如果是第一次运行可以不用考虑运行顺序，但如果不是第一次运行则需要按照以下顺序执行程序：

运行顺序为：

dataset\_provider.py

feature\_get.py

aggregate.py

test.py

程序默认从本地读取csv文件作为原始数据。如果想从phm平台调取数据，则要修改dataset\_provider.py中main函数内app.run的参数，参数为0则从本地调取csv文件，参数为1则从phm平台上调取数据。修改位置如下图所示：



启动所有程序可能会需要一两分钟（主要是数据清洗有点慢）），可以看屏幕输出或是查看.py文件下的日志观察程序是否运行正常

完全启动之后，可以通过api\_forward.py中设置的api对数据进行请求。

在test.py所在文件夹下，会生成一个result\_(时间).py的文件，这就是运行的结果了

1. 运行微信小程序

由于微信小程序中调用的ip被写死了，不方便修改，所以这里就不介绍如何运行了

### 一些想说的：

我是这次课设的后端编写组员。在构思这次课设程序该如何编写的时候，我想试着把每个数据处理步骤之间相互解耦，让所有数据之间的交互通过http请求和数据库来进行（单纯只是想尝试一下不一样的编程思路，其实对这次课设来说并没有必要...只要直接读csv文件就可以做得很好），结果就是这次的程序做出来每个数据处理小模块都带有单独的api，整个程序的逻辑有点复杂，操作也不是很方便，还请老师见谅！