Sklearn机器学习算法包的框架

1. 如何封装一个算法包？
2. 写个Python包，打so文件，不再需要复制base\_operator？

通过pypi打包，和其他pandas、numpy类似

一、背景

       现有算法文件打成so包时，会将base\_operator整个代码打进so包，若基类需要扩展或者修改，将会导致所有的so包都会随着改变，并重新打包，这不符合高内聚低耦合的设计思想，

所以需要将base\_operator从so包中隔离处理。

二、思路

       将base\_operator作为单独的一个包，单独的安装到执行算法的python环境中，就可以很轻松的实现低耦合的思想，以后还可以在该包扩展算子的公共模块代码。

三、打包流程

**1、首先需要将base\_operator整合成可安装模块，目录结构如下所示：**

     其中base\_operator.py即base\_operator模块代码

**2、Python 库打包的关键在于编写 setup.py 文件。setup.py 文件编写的规则是从 setuptools 或者 distuils 模块导入 setup 函数，并传入各类参数进行调用**

|  |
| --- |
| from setuptools import setup, find\_packages    setup(      name='base\_operator',   # 包名      version='1.2.0',        # 包版本      py\_modules=['base\_operator'],  # 需要打成包的模块列表      url='',                        # 程序的官网地址      license='',                    # 程序的授权信息      author='jianping.tang',        # 程序作者      author\_email='jianping.tang@tongdun.cn',  # 作者email      description='operator\_test base util',    # 包的简单描述      packages=find\_packages('src'),            # 需要处理的包目录(通常为包含 \_\_init\_\_.py 的文件夹)      package\_dir={'':'src'},                   # 指定哪些文件需要映射到src      include\_package\_data = True,              # 自动包含包内所有受版本控制(cvs/svn/git)的数据文件  ) |

1. 接口一致，底下调用不同的语言，Python、Scala、r等？