**Tabpy环境搭建**

在Tableau公司的Github主页下，有一个名为Tabpy的项目，提供了Tableau和Python集成的套件:Tabpy framework，该套件包含两个组件：

1. Tabpy Server, 一个远程服务器，用于运行从tableau传递过来的python代码；

2. Tabpy client, 用于将用户建立的数据分析或数据挖掘的模型发布到Tabpy Server， 增强代码可重用性。

**Tabpy Server**

官方提供了一个整体打包的方案，在github页面下载整个项目到本地，点击setup.bat或setup.sh文件就可以自动下载Anaconda环境，创建一个名为Tableau-Python-Server的环境，并且在这个环境中安装好Tabpy及相关的各种库。

如果你是老司机喜欢用手动档，那我们一起来看一下手动安装的步骤(以Windows平台为例)：

1. 本机下载安装Anaconda，把Anaconda的Python环境加入到环境变量；

2. 运行以下命令在Anaconda上建立一个名为Tableau-Python-Server的环境：

conda create --name Tableau-Python-Server python=2.7 anaconda

3. 运行以下命令切换到新创建的环境：

activate Tableau-Python-Server

然后运行命令安装Tabpy

pip install tabpy-server

如果命令运行一切正常，你会在该目录Users\yourname\anaconda\envs\Tableau-Python-Server\Lib\site-packages下看到一个名为tabpy\_server的文件夹，里面包含了一个用于启动Tabpy server的startup.bat文件，把它创建一个快捷方式到你的桌面，以后你会需要经常点击它启动Tabpy Server。

截至目前，如果不考虑代码可重用性的话，你现在就可以启动Tabpy Server，打开tableau然后在外部服务器中配置本地IP和端口9004，然后在计算字段中写python代码了。

不过机器学习模型包含大量代码，把它们都放在tableau小的可怜的计算字段编辑框里面显然不是一种友好的方式，或者你不想每次使用都要重新写一遍函数逻辑的话，最好还是使用Tabpy client, 在本地编辑好你要使用的函数或模型，然后发布到Tabpy Server，这样每个连接到Tabpy Server的tableau工作簿，只要输入你写好的函数名就可以直接使用它，是不是很方便？

**Tabpy Client**

和前面一样，如果你运行bat或sh脚本的话，client应该已经安装上了，不过你也可以选择手动安装它：

pip install tabpy-client

下面给出一个简单的例子说明如何使用tabpy client发布python代码，本地创建一个py文件写入一下代码：

import tabpy\_client

client = tabpy\_client.Client('http://localhost:9004/')

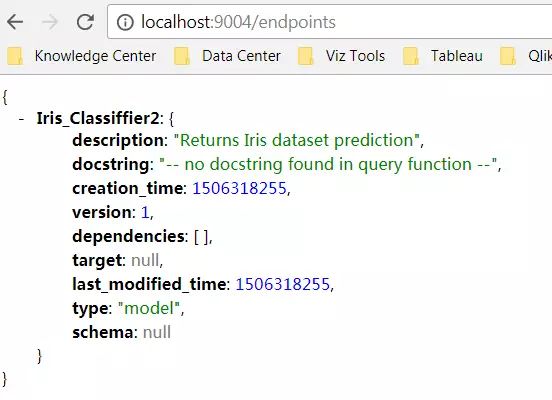
def add(x,y):

import numpy as np

return np.add(x, y).tolist()

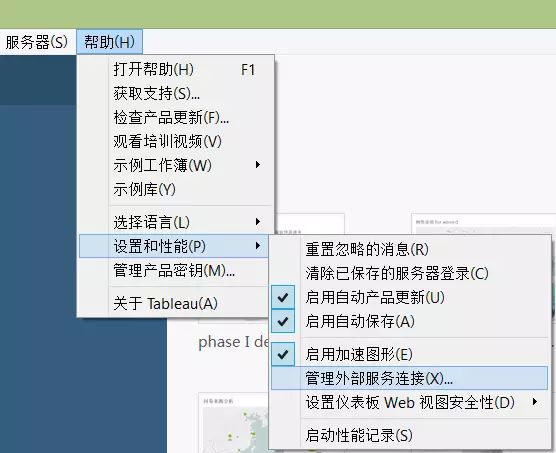
client.deploy('add', add, 'Adds two numbers x and y')

先运行startup.bat文件,再在本地python环境中运行上面的代码，没有任何问题的话接着浏览器中打开 http://localhost:9004/endpoints 看一下，如果返回一堆包含JSON格式键值对的话说明你的函数已经加入到tabpy server了。



**连接Tableau到Tabpy**

接下来打开Tableau，依次点击“帮助-设置和性能-管理外部服务连接”



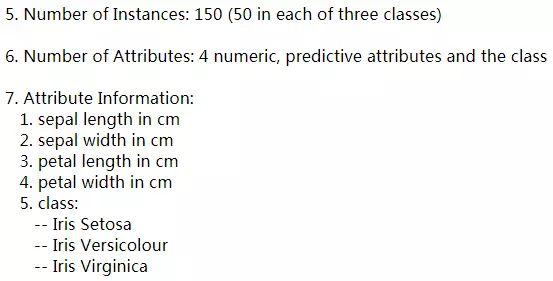
弹出的对话框输入地址和端口号：“localhost” "9004"，测试一下，bingo！成功连接到server。

**基于Iris数据集的机器学习实践**

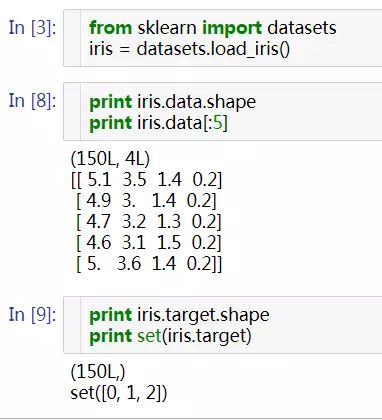
**Iris数据集简介**

iris数据集由Fisher, 1936收集整理，Iris也称***安德森鸢尾花卉数据集***，英文全称是***Anderson’s Iris data set***。Iris数据集是常用的分类实验数据集，是一类多重变量分析的数据集。数据集包含150个数据集，分为3类，每类50个数据，每个数据包含4个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于Setosa，Versicolour，Virginica三个种类中的哪一类。

用户可通过链接[http://aima.cs.berkeley.edu/data/iris.txt](https://link.zhihu.com/?target=http%3A//aima.cs.berkeley.edu/data/iris.txt) 查看iris数据集的详细介绍了解更多信息。

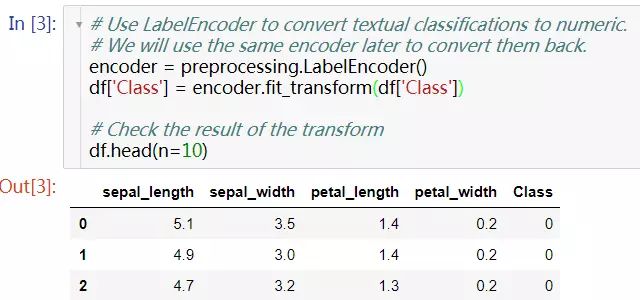


用户可以从 [http://aima.cs.berkeley.edu/data/iris.csv](https://link.zhihu.com/?target=http%3A//aima.cs.berkeley.edu/data/iris.csv)下载该数据集，另外，python的数据挖掘/机器学习库scikit已经内置了iris数据集。

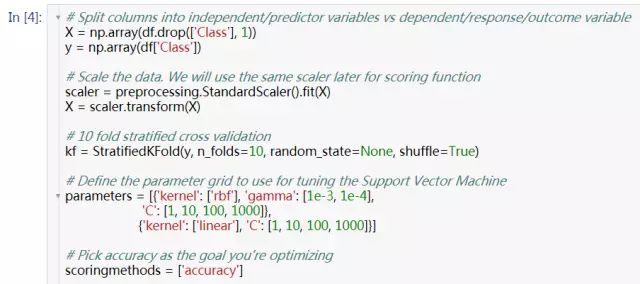


**分类算法**

1. 我们选用jupyter notebook作为实验平台，数据集采用从上面链接下载的csv文件，添加列名以便于操作。csv文件的Class列有三类值Setosa，Versicolour，Virginica，因此这是一个多分类的问题。首先我们要将Class列转换为数值型字符。

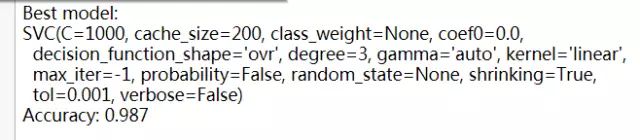


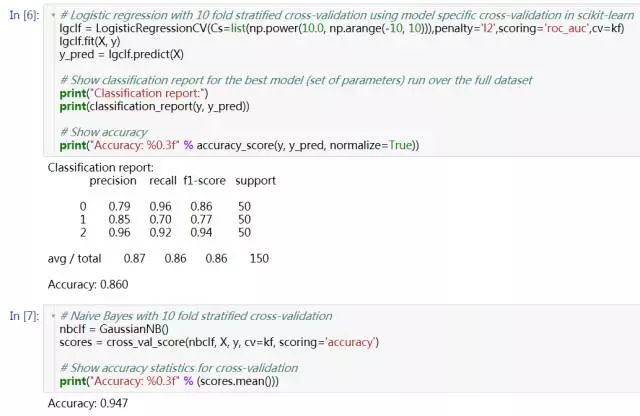
2. 对数据进行标准化预处理，数据交叉验证采用sklearn 包中的 StratifiedKFold 方法，StratifiedKFold 是一种将数据集中每一类样本的数据成分，按均等方式拆分的方法，使用准确率作为评价模型好坏的标准。



3. 接下来分别使用SVM， 逻辑回归分类器，朴素贝叶斯三种分类算法进行验证



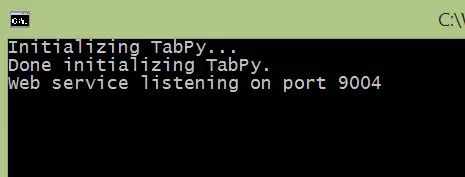




得到准确率分别是：SVM 0.987， 逻辑回归 0.860，朴素贝叶斯 0.947，就是你了**SVM**！

**发布函数**

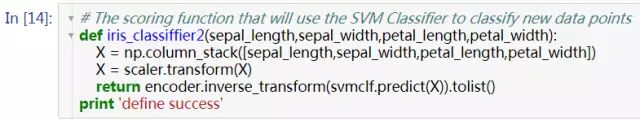
首先运行startup.bat（Mac下为startup.sh）启动tabpy\_server 服务器



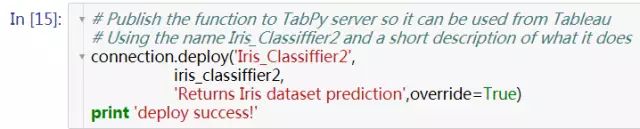
接下来需要创建一个到tabpy server的连接对象



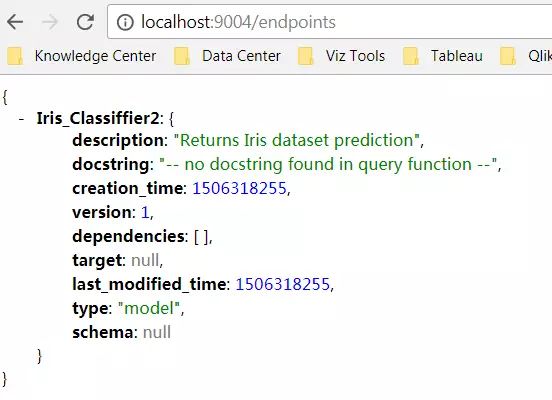
定义我们要发布的函数名称，参数，以及返回对象



然后将方法发布到tabpy server上



最好再去 localhost:9004/endpoints 下确认一下方法已经发布到了tabpy server上。如果没有问题，我们就可以进入到下一阶段，去创建Tableau dashboard了，**Cheers**!





到这里，我们已经成功搭建好了Tabpy环境，**Cheers**!

**Tableau报表创建**

**数据准备**

我们还是引用iris数据集的csv数据格式作为tableau的数据源，里面包含了class这个维度字段和sepal\_length, sepal\_width, petal\_length, petal\_width等4个度量字段。接下来，依次创建同名的4个参数，作为手动预测输入值。同时我们还要对批量的数据做预测，在这里还是使用iris数据集，把预测值和原来的值做一个对比，也算是做验证，看一下准确率如何。因此，我们还需要创建下面几个字段。

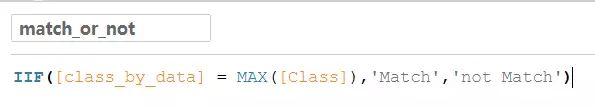
**class\_by\_para**：基于手动调整的参数进行预测



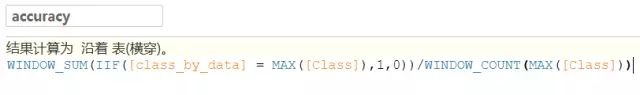
**class\_by\_data**：基于表格中的数据进行预测



**match\_or\_not**： 用于标记预测准确与否



**accuracy**: 用于计算预测准确率



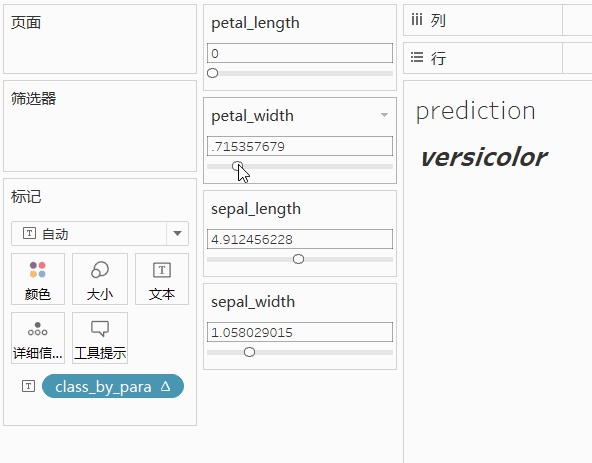
**可视化分析**

先来做一个聚类分析图看看预测的效果，看起来petal\_width这个特征上分类效果显著。setosa这个类别和其它两个类差别明显。

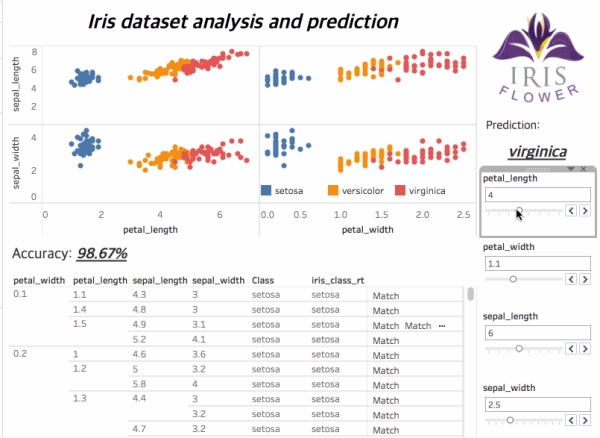


接下来看一下明细的数据，真实分类和预测分类数据不match的用形状来标识一下，效果很直观。

基于参数的预测，最好指定一下参数的取值范围和格式。



最后把worksheets放在dashboard里，增加筛选器，文字和图片装饰，就大功告成啦！



**总结**

通过上面的实践，有以下几点值得注意(**敲黑板，要划重点了**)：

1. Tableau目前只能在SCRIPT\_XXX等4个函数中写Python脚本，对于本身比较复杂的函数，将代码部署至Tabpy服务器不仅使代码更简洁，而且也使得函数可复用；

2. 在Tableau中引用算法模型可实现基于机器学习模型的实时分析，增强了Tableau的可视化分析能力；

3. 在Tableau中引入的函数需要进行实时表计算，因此对于大规模的数据会比较耗时。一种折中的做法是在数据预处理阶段先使用算法模型计算好要求的值，然后Tableau中进行可视化分析。

**参考资料**

[Tableau Tabpy Github page](https://link.zhihu.com/?target=https%3A//github.com/tableau/TabPy)

[Building advanced analytics applications with TabPy](https://link.zhihu.com/?target=https%3A//www.tableau.com/about/blog/2017/1/building-advanced-analytics-applications-tabpy-64916)

本文所有数据，算法模型以及tableau文件均已上传至[**这里**](https://link.zhihu.com/?target=https%3A//github.com/swucim/iris_tableau_python)，欢迎交流。