

Jupyter Notebook的使用技巧

Jupyter Notebook的使用技巧

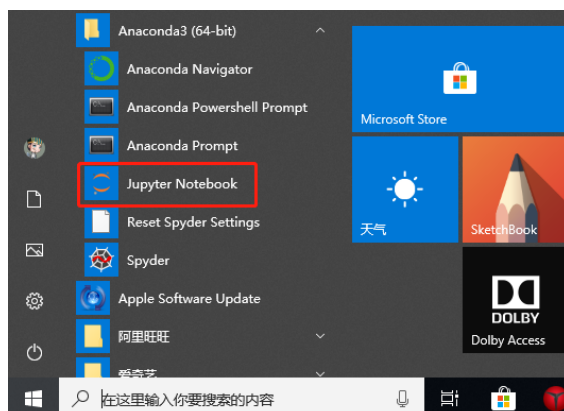
by 华能信托 王宇韬

Jupyter Notebook是Anaconda自带的一款非常不错的代码编辑软件，其的优点在于：
1、可以非常方便的进行代码分块运行；2、运行的结果可以自动保存，不需要之后重复运行代码；3、可以直接在这个在单个模块中打印数据进行查看，非常方便代码调试，所以它在机器学习这种常和数据打交道的过程中是很有帮助的。

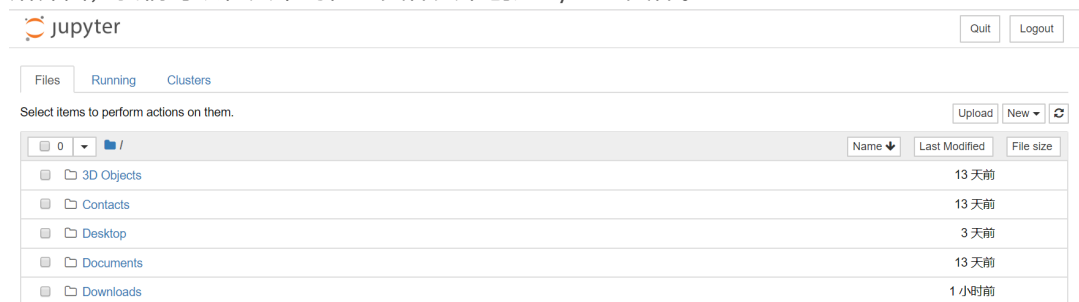
笔者常用Jupyter Notebook进行机器学习的代码调试与整理，最终在Pycharm中运行完整的项目。下面讲解下Jupyter Notebook的使用技巧：

1.打开Jupyter Notebook（重点在打开任一文件夹中的Notebook文件）

Jupyter Notebook 打开方法如下：电脑左下角打开 Anaconda，点击 Jupyter Notebook。



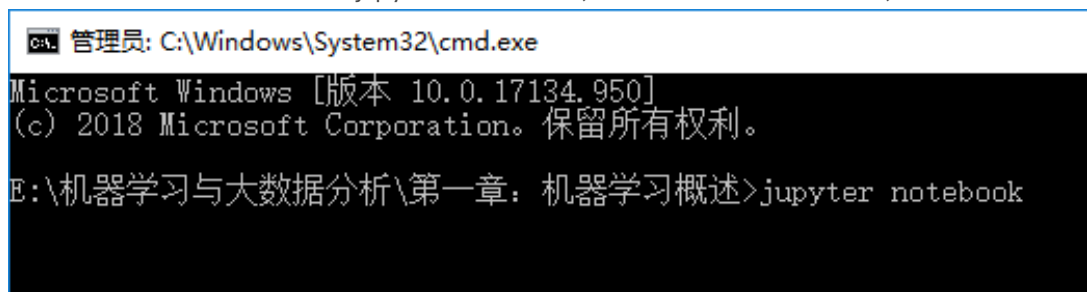
此时会在浏览器中Jupyter Notebook，此时浏览器只是个工具载体，如下图所示是其初始界面，我们可以在其中的任一文件夹下创建Python文件。



这里默认打开的是C盘里的文件夹，**如果想要打开指定文件夹里的Jupyter Notebook文件**，可以采用如下办法，在该文件夹的路径框内输入“cmd”，然后按Enter回车键，如下图所示：



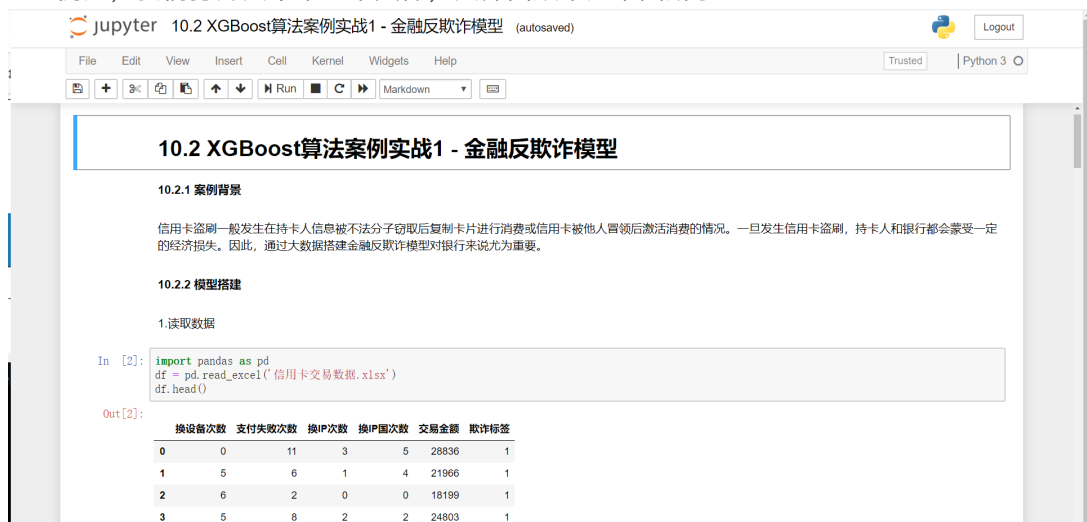
然后在弹出的界面中输入“jupyter notebook”，然后按Enter回车键即可，如下图所示：



然后便能在默认浏览器中看到如下内容，单击相关Python文件即可将其打开并进行查看。



例如，我们打开其中第二个文件，其界面效果如下图所示：



此外，因为Jupyter Notebook是通过浏览器打开的，所以如果觉得界面的字体较小，可以通过Ctrl + 鼠标滚轮键来调节界面大小。

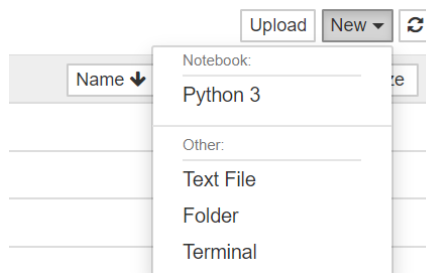
此外，除了弹出浏览器界面外，其实它还会弹出Jupyter Notebook的管理窗口，这个管理窗口正常情况下用户用不着，但是不可以关闭它，一旦关闭，则浏览器中的Jupyter Notebook则会显示连接断开。此外，如果浏览器中没有自动弹出Jupyter Notebook相关界面，也可以复制下图中红框中那行链接至浏览器搜索栏中即可。

```
Jupyter Notebook
[I 15:46:43.398 NotebookApp] The port 8888 is already in use, trying another port.
[I 15:46:43.758 NotebookApp] JupyterLab extension loaded from D:\Anaconda\Anaconda\lib\site-packages\jupyterlab
[I 15:46:43.759 NotebookApp] JupyterLab application directory is D:\Anaconda\Anaconda\share\jupyter\lab
[I 15:46:43.790 NotebookApp] Serving notebooks from local directory: C:\Users\85079
[I 15:46:43.791 NotebookApp] The Jupyter Notebook is running at:
[I 15:46:43.791 NotebookApp] http://localhost:8889/?token=0cd336b2a08fa2d95d12b8754e1873122e7d4741534f8f2a
[I 15:46:43.791 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 15:46:49.070 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8889/?token=0cd336b2a08fa2d95d12b8754e1873122e7d4741534f8f2a
[I 15:46:49.781 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

2.创建Python文件

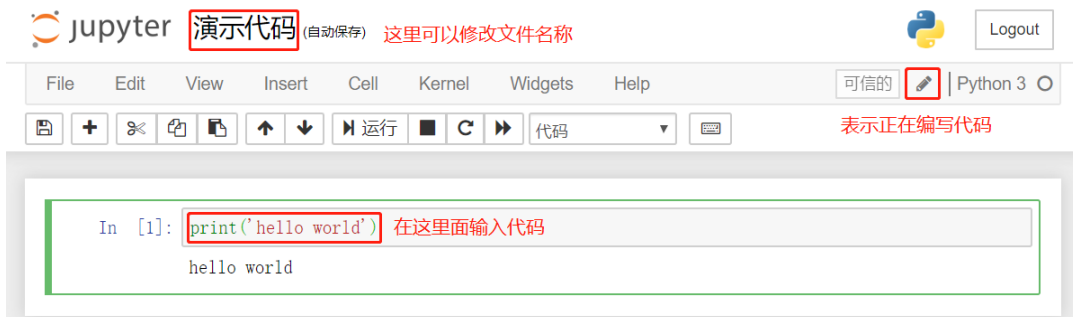
如下图所示，在右上角的New按钮，选择Python3，可以创建Python文件，如果需要创建新文件夹，你们选择其中的Folder即可。



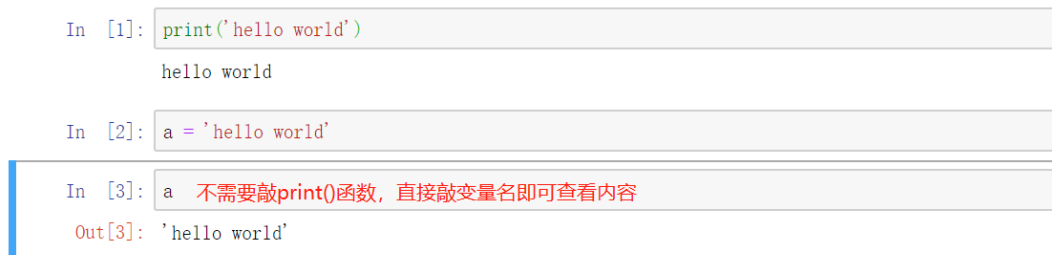
Jupyter Notebook创建的Python文件后缀名为.ipynb，而常规的Python文件后缀则为.py。因此在Jupyter Notebook中我们创建和打开的都是后缀名为.ipynb的文件。

3.编写代码

如下图所示，在区块中即可编写代码，编写完毕后，按住Ctrl + Enter键即可运行当前区块，或者按上方菜单栏中的运行按钮运行代码，在编写代码的时候区块边框显示为绿色。



Jupyter Notebook的一个好处就是在对于变量，它不需要输入print()函数，也能快速打印内容，方便编程者查看，如下图所示。



对于一些类型的数据，如下一章将要讲到的DataFrame表格类型数据，直接敲变量名比利用print()函数打印呈现的效果更好。

4.菜单栏介绍

这里再介绍下菜单栏，通常情况下，我们不会经常使用菜单栏，不过其中有些功能还是需要关注一下的，我们稍后会重点讲一些Cell和Kernel中的一些特色功能，下图为菜单栏：

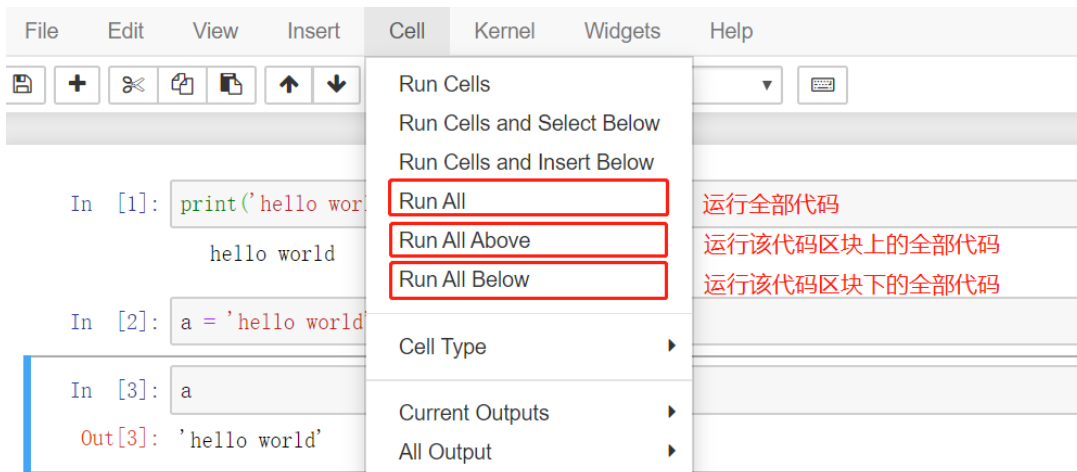


其中File按钮可以用来打开和存储文件，File按钮中的Download As可以把Jupyter Notebook创建的后缀名为.ipynb的Python文件另存为后缀名为.py的常规Python文件。

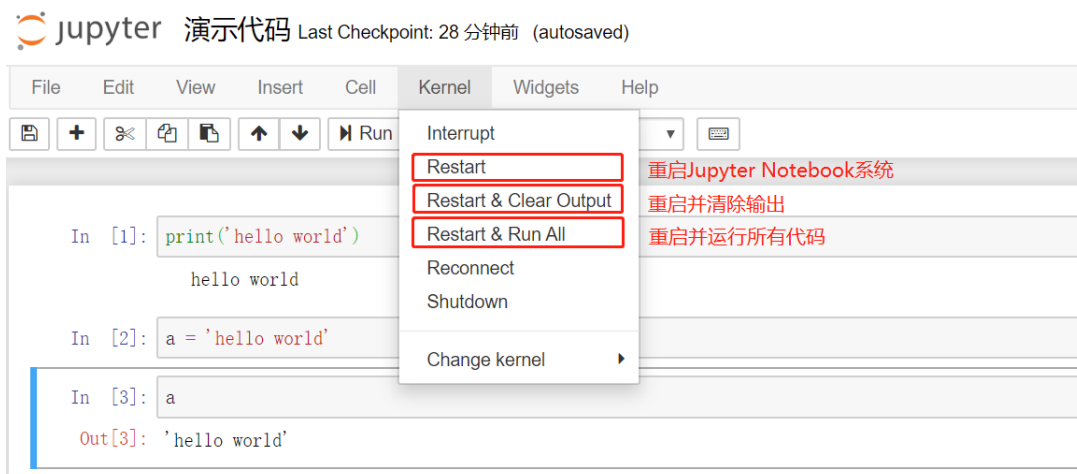
Edit按钮中则是一些编辑区块的内容，如剪切、复制、删除区块等，其中有些功能也可以下图中的快捷方式来实现，当把鼠标悬停在快捷按钮上，可以看到相关解释。

Insert按钮可以插入区块，这个一般用下面要讲的快捷键完成；Cell按钮可以选择运行当前区块、运行当前区块之前或之后等内容；Kernel按钮中可以中断或重启程序；Help按钮中的Keyboard Shortcuts可以查看快捷键。

Cell菜单中的一些比较有意义的功能如下图所示：



Kernel菜单中的一些比较有意义的功能如下图所示：



之所以要特地强调下“Kernel”菜单中的“Restart”（重启系统）选项，是因为有的时候Jupyter Notebook运行过程中程序因为某些问题一直卡着不动（如代码陷入死循环），这时候通过终止按钮，或者通过上图中的“Interrupt”（中断系统）选项是终止不了程序的，而通过Restart（重启系统）则能非常快速的终止程序。

5.快捷按钮介绍

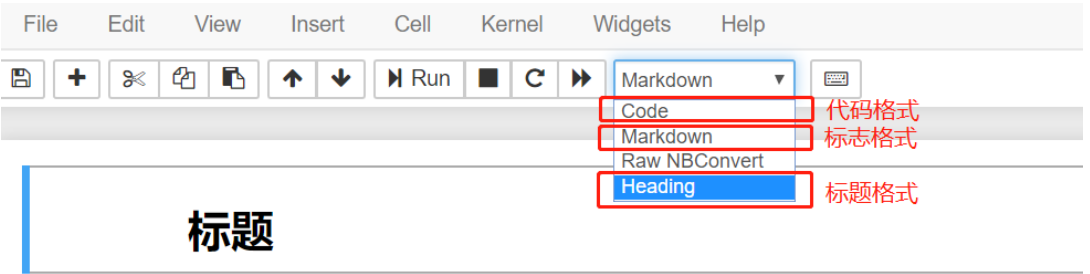
除了使用菜单栏外，Jupyter Notebook还有不错的快捷按钮，如下图所示：



依次作用为：保存；在下面插入代码块；剪切代码块、复制代码块、粘贴到下面；将选中代码块上移、将选中代码块下移；运行当前代码块、中断系统（如果中断不了，则推荐选择重

启系统)、重启系统(就是上面讲的Kernel中的Restart)、重启并运行所有代码;代码以及标题框;打开命令配置。

这里单独讲解一下“代码及标题框”按钮,它可以设置区块为代码(Code)、标题(Heading)或标志(Markdown)(类似于笔记或者注释,Markdown是一种专门的笔记语言,更多Markdown的使用技巧可以自行搜索),如下图所示,通过它我们可以在代码里设置标题和标志,方便阅读代码。注意设置后要按Ctrl+Enter键运行该区块才可以完成设置。



这是一个标志 (Markdown)

```
In [1]: print('hello world')
hello world
```

此外,通过快捷键“m”可以将代码格式的内容快速切换为标志格式的内容,通过快捷键“y”则可以切换为代码格式的内容。

6.常用快捷键

在实际操作中更多会使用快捷键来进行操作, Jupyter Notebook常用快捷键如下所示:

快捷键	作用
Ctrl + Enter	运行当前区块代码
Shift + Enter	运行当前区块, 并转到下一区块
a	在当前区块上方新建一个区块
b	在当前区块下方新建一个区块
连续按两次d键	删除当前区块

注意a、b、连续按两次d键等快捷键方式需要选中区块的情况下才会生效,当区块被选中时,其边框颜色为蓝色,如下图所示:



此外, Jupyter Notebook是默认不显示代码行号的, 如果需要显示代码行号, 可以在编程界面上通过快捷键Shift + L来显示行号, 显示行号效果如下图所示:

```
1 # 1. 读取数据
2 import pandas as pd
3 df = pd.read_excel('股票客户流失.xlsx')
4 df.head()
```

7.在Jupyter Notebook中安装库

如果想安装第三方库, 可以在Jupyter Notebook中非常方便的安装, 只需要在代码框中输入如下类似代码即可, 然后通过Ctrl+Enter或者Shift+Enter键运行该代码框即可。

```
!pip install 库名
```

例如安装爬虫相关库requests库, 即可采用如下代码, 然后运行该代码框即可:

```
1 !pip install requests
```