pip install -U scikit-learn 安装并更新scikit-learn

pip list 显示所有安装包及版本

# Scikit-learn

Scikit-learn是一个用于Python编程语言的免费软件机器学习库。 它具有各种分类，回归和聚类算法，包括支持向量机，随机森林，梯度增强，k-means和DBSCAN，旨在与Python数值和科学库NumPy和SciPy互操作。

# pandas

Python Data Analysis Library

pandas是BSD许可的开放源代码库，为Python编程语言提供了高性能，易于使用的数据结构和数据分析工具。pandas是NumFOCUS赞助的项目。 这将有助于确保将pandas开发成功作为世界一流的开源项目，并有可能向该项目捐款。

# Scipy

Scipy是一个用于数学、科学、工程领域的常用软件包，可以处理插值、积分、优化、图像处理、常微分方程数值解的求解、信号处理等问题。它用于有效计算Numpy矩阵，使Numpy和Scipy协同工作，高效解决问题。

# NumPy

NumPy（Numeric Python）提供了许多高级的数值编程工具，如：矩阵数据类型、矢量处理，以及精密的运算库。专为进行严格的数字处理而产生。多为很多大型金融公司使用，以及核心的科学计算组织如：Lawrence Livermore，NASA用其处理一些本来使用C++，Fortran或[Matlab](https://baike.baidu.com/item/Matlab)等所做的任务。  
编程问题：

我们使用处理二进制文件时，需要用如下方法

binfile=open(filepath,’rb’) 读二进制文件

binfile=open(filepath,’wb’) 写二进制文件

那么和binfile=open(filepath,’r’)的结果到底有何不同呢？

不同之处有两个地方：

第一，使用’r’的时候如果碰到’0x1A’，就会视为文件结束，这就是EOF。使用’rb’则不存在这个问题。即，如果你用二进制写入再用文本读出的话，如果其中存在’0X1A’，就只会读出文件的一部分。使用’rb’的时候会一直读到文件末尾。

第二，对于字符串x=’abc\ndef’，我们可用len(x)得到它的长度为7，\n我们称之为换行符，实际上是’0X0A’。当我们用’w’即文本方式写的时候，在windows平台上会自动将’0X0A’变成两个字符’0X0D’，’0X0A’，即文件长度实际上变成8.。当用’r’文本方式读取时，又自动的转换成原来的换行符。如果换成’wb’二进制方式来写的话，则会保持一个字符不变，读取时也是原样读取。所以如果用文本方式写入，用二进制方式读取的话，就要考虑这多出的一个字节了。’0X0D’又称回车符。linux下不会变。因为linux只使用’0X0A’来表示换行。

# python中strip(),rstrip(),lstrip()用法详解

Python中有三个去除头尾字符、空白符的函数，它们依次为:

Strip： 用来去除头尾字符、空白符(包括\n、\r、\t、' '，即：换行、回车、制表符、空格)  
lstrip：用来去除开头字符、空白符(包括\n、\r、\t、' '，即：换行、回车、制表符、空格)  
rstrip：用来去除结尾字符、空白符(包括\n、\r、\t、' '，即：换行、回车、制表符、空格)

注意：这些函数都只会删除头和尾的字符，中间的不会删除。

用法分别为：  
string.strip([chars])  
string.lstrip([chars])  
string.rstrip([chars])

参数chars是可选的，当chars为空，默认删除string头尾的空白符(包括\n、\r、\t、' ')  
当chars不为空时，函数会被chars解成一个个的字符，然后将这些字符去掉。

它返回的是去除头尾字符(或空白符)的string副本，string本身不会发生改变。

举例说明如下：  
1. 当chars为空时，默认删除空白符(包括'\n', '\r', '\t', ' ')

>>> str = ' ab cd '

>>> str

' ab cd '

>>> str.strip() #删除头尾空格

'ab cd'

>>> str.lstrip() #删除开头空格

'ab cd '

>>> str.rstrip() #删除结尾空格

' ab cd'

## 2.当chars不为空时，函数会被chars解成一个个的字符，然后将这些字符去掉。

>>> str2 = '1a2b12c21'

>>> str2.strip('12') #删除头尾的1和2

'a2b12c'

>>> str2.lstrip('12') #删除开头的1和2

'a2b12c21'

>>> str2.rstrip('12') #删除结尾的1和2

'1a2b12c'

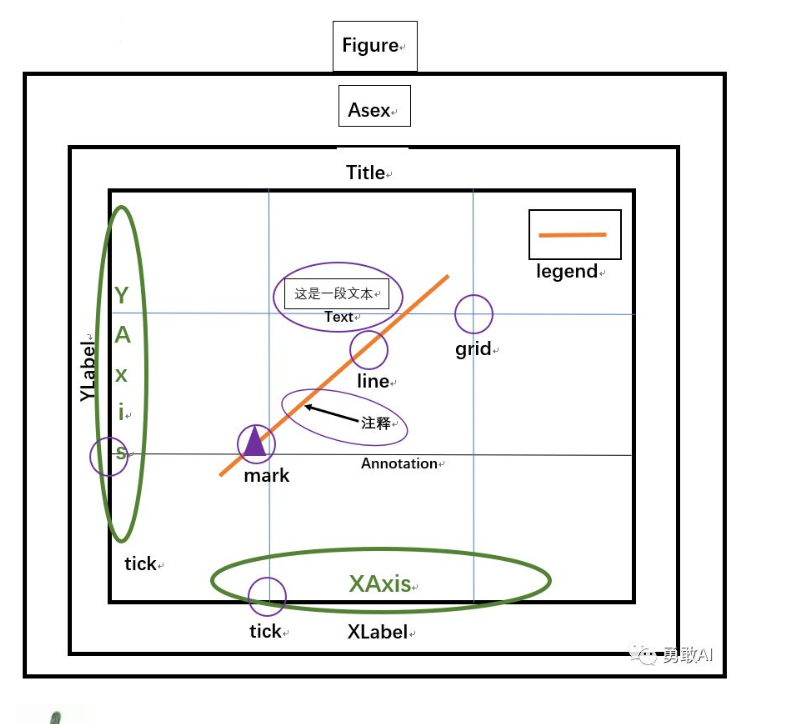
# Matplotlib作图编程问题总结

## 1 .画板理解

Figure就是画板，画纸的载体，在pyplot上，画纸的概念就是axes/subplot

第一已经回答得很清楚了，我也被这个恶心过，说一种更好理解的方式。

可以把fig想象成windows的桌面，你可以有好几个桌面。然后axes就是桌面上的图标，subplot也是图标，他们的区别在：axes是自由摆放的图标，甚至可以相互重叠，而subplot是“自动对齐到网格”。

但他们本质上都是图标，也就是说subplot内部其实也是调用的axes，只不过规范了各个axes的排列罢了。

## 2. 画图函数理解

subgraph(G, nbunch) - 由图G的分支nbunch生成子图

union(G1,G2) - 将图G1，G2结合构成一个图

disjoint\_union(G1,G2) - 假设所有的节点都不相同构成图

cartesian\_product(G1,G2) - 由笛卡尔积生成图

compose(G1,G2) - 结合两个图中共有的点构成图

complement(G) - 图的补

create\_empty\_copy(G) - 返回相同图类的空副本

convert\_to\_undirected(G) - 返回G的无向图表示

convert\_to\_directed(G) - 返回G的有向图表示