

# 暑期机器学习课程上机实践指南

作者：赵 帅\*

上机指导老师：蔡 登

2019.06

---

\*有疑问请联系21721043@zju.edu.cn

# 目录

<b>1 简介</b>	<b>3</b>
<b>2 实践任务</b>	<b>3</b>
2.1 概述	3
2.2 关于第4个任务 $knn$ 的补充说明	4
<b>3 编程语言和实验环境</b>	<b>4</b>
3.1 编程语言	4
3.1.1 Python	4
3.1.2 Matlab	5
3.2 Jupyter Notebook	6
3.2.1 什么是Jupyter Notebook	6
3.2.2 Jupyter Notebook的使用	6
<b>4 实验结果评定</b>	<b>7</b>
4.1 助教评定	7
4.2 自我评定	8
<b>A Numpy</b>	<b>9</b>
A.1 np.array()	9
A.2 np.sum()	9
A.3 np.min()	10
A.4 np.matmul()	10
A.5 np.multiply()	11
A.6 np.log()	11
A.7 np.tile()	11
A.8 np.argsort()	12
A.9 np.expand_dims()	13
<b>B Scipy</b>	<b>13</b>
B.1 scipy.stats.mode()	13
<b>C Matplotlib</b>	<b>14</b>

## 1 简介

本文档将简单介绍，在2019年浙江大学暑期机器学习课程上机实践中，要完成的实践任务，使用的编程语言和相关实验环境及其对应的实验步骤，最后将介绍如何对实验结果进行评估。

## 2 实践任务

### 2.1 概述

下载本次上机实践的资料并解压后，文件目录结构如图 1：

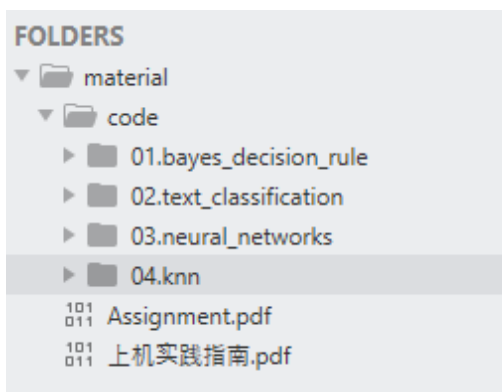


图 1: 上机实践材料目录结构，`code`目录中是四个编程任务的代码文件，*Assignment.pdf*中包含要完成的实践题目，*上机实践指南.pdf*是本次上机的实践指南(你现在正在看的)。

本次上机实践，包含4个具体的任务。

- *bayes\_decision\_rule*: 理解并应用贝叶斯定理，利用朴素贝叶斯定理实现简单的二分类任务；
- *text\_classification*: 利用贝叶斯定理设计垃圾邮件分类器；
- *neural\_networks*: 实现多层全连接神经网络的前向传播和梯度反向传播算法；
- *knn*: 实现简单的最近邻搜索算法并利用其破解浙江大学现代教务管理系统网站的数字验证码。

在每个代码文件夹中，都有一个`run.m`或者`run.ipynb`文件(*knn*中是`knn_exp`)，大家在每个编程任务中，要完成的主要工作都在这个文件里面。但是请注意，你的工作并不仅限于这个文件，你可能还需要在一些其他的文件中填上相应的代码，请仔细阅读*Assignment.pdf*的内容和代码文件中的注释(需要你填写代码的地方一般都有*TODO*标志)。代码文件中的注释都是极为有用的，能帮助你大大加快任务完成的速度。

选择使用Python则完成`run.ipynb`及相关`.py`文件的内容，选择使用Matlab则完成`run.m`及相关`.m`文件的内容。二者选择其一就可，因为他们只是同一内容的不同编程语言实现而已。

## 2.2 关于第4个任务knn的补充说明

第四个任务中，会要求大家破解浙江大学现代教务管理系统网站的数字验证码。这个任务中，你首先要去这个网站上爬取充足数量的验证码，接着对这些验证码的数字进行标注，然后你才可以利用你实现的knn算法和你标注好的数据对教务管理系统网站的验证码进行破解。

这整个的流程和常见的机器学习算法的学习流程是大致相同的，收集数据—清洗、标注数据—训练机器学习模型—应用模型。这个任务目的也是让大家体会这个过程。为了加快大家的实践完成速度，我们已经提前收集好了999张验证码图片，位于knn/CAPTCHA.zip中，但这些图片并没有被标注，大家可以分工合作进行标注。不一定要全部标注，但训练数据的多少会对你最后算法的准确度有影响。

假如觉得图片不够，可以利用knn/spider文件夹中的简单爬虫脚本爬取更多图片。

## 3 编程语言和实验环境

### 3.1 编程语言

本次上机实践，支持Python和Matlab。

Python是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python由Guido van Rossum于1989年底发明，第一个公开发布版发行于1991年。由于其开源、简单易用、拓展性强等特点，在当前人工智能和深度学习的浪潮之下，Python成为了使用最为广泛的语言。熟练掌握Python对于后续的学习大有益处。因此，**Python是被鼓励的选择**，本指南中将主要介绍Python语言及相关工具的使用。

下面将简单介绍如何配置实验环境及确认实验环境是否配置成功。

#### 3.1.1 Python

Python的版本应当大于等于3.0。可以在命令行窗口运行python -version检查python版本(双短线)。

```
$ python --version
Python 3.6.5
```

假如机器上面没有安装python, 可前往官方网站<https://www.python.org/>下载安装。

Python的一大特点在于其拥有众多功能丰富的第三方模块，本次实验主要使用到的有交互计算环境Jupyter Notebook，科学计算包numpy，Python算法库和数学工具包scipy，图形用户界面工具包matplotlib。Python的工具包一般情况下由一个叫做pip的包管理工具管理着，可以在命令行窗口运行pip3 list查看已经安装的包

```
$ pip3 list
Package            Version
-----
absl-py            0.2.0
astor              0.6.2
backcall           0.1.0
bleach             1.5.0
brewer2mpl         1.4.1
```

```

$ pip3 install -i https://mirrors.zju.edu.cn/pypi/web/simple/ jupyter numpy scipy matplotlib
Looking in indexes: https://mirrors.zju.edu.cn/pypi/web/simple/
Requirement already satisfied: jupyter in c:\python36\lib\site-packages (1.0.0)
Requirement already satisfied: numpy in c:\python36\lib\site-packages (1.14.3)
Requirement already satisfied: scipy in c:\python36\lib\site-packages (1.1.0)
Requirement already satisfied: matplotlib in c:\python36\lib\site-packages (2.2.2)
Requirement already satisfied: jupyter-console in c:\python36\lib\site-packages (from jupyter) (5.2.0)
Requirement already satisfied: notebook in c:\python36\lib\site-packages (from jupyter) (5.5.0)
Requirement already satisfied: nbconvert in c:\python36\lib\site-packages (from jupyter) (5.3.1)
Requirement already satisfied: ipywidgets in c:\python36\lib\site-packages (from jupyter) (7.2.1)
Requirement already satisfied: ipykernel in c:\python36\lib\site-packages (from jupyter) (4.8.2)
Requirement already satisfied: qtconsole in c:\python36\lib\site-packages (from jupyter) (4.3.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\python36\lib\site-packages (from matplotlib) (2.7.3)
Requirement already satisfied: pytz in c:\python36\lib\site-packages (from matplotlib) (2018.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\python36\lib\site-packages (from matplotlib) (2.2.0)
Requirement already satisfied: cyclar>=0.10 in c:\python36\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\python36\lib\site-packages (from matplotlib) (1.0.1)
Requirement already satisfied: six>=1.10 in c:\python36\lib\site-packages (from matplotlib) (1.11.0)
Requirement already satisfied: pygments in c:\python36\lib\site-packages (from jupyter-console->jupyter) (2.2.0)
Requirement already satisfied: ipython in c:\python36\lib\site-packages (from jupyter-console->jupyter) (6.4.0)
Requirement already satisfied: jupyter-client in c:\python36\lib\site-packages (from jupyter-console->jupyter) (5.2.3)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.0 in c:\python36\lib\site-packages (from jupyter-console->jupyter) (1.0.15)

```

图 2: 安装Python第三方包。

```

cloudpickle      0.5.3
colorama         0.3.9
cyclar           0.10.0
.....

```

你需要在上面已经安装的包中确认有没有本次实践需要的包，否则在运行时会引发找不到对应包的错误。

假如没有对应的包，你可以运行pip3 install命令来安装对应的包。

```

pip3 install -i https://mirrors.zju.edu.cn/pypi/web/simple/
jupyter==1.0.0
numpy==1.15.2
scipy==1.1.0
matplotlib==3.0.0

```

这里-i选项指定了安装源，==符号指定了安装的版本。图 2中我已经安装好了，再运行就提示我需求已满足。

我们会在后续提供一些关于这些包的说明，在最后的附录中我们也会提供一些可能会用到的这些包中的函数接口的说明。假如你对于Python完全不熟悉，推荐前往菜鸟教程<https://www.runoob.com/python3/python3-tutorial.html> 阅读关于Python的介绍以及一些基础知识<sup>1</sup>。

### 3.1.2 Matlab

本次实践要求Matlab的版本大于2014a。假如机器上没有安装Matlab，可以前往浙江大学信息技术中心 <http://itc.zju.edu.cn/2017/1110/c7943a689905/pagem.htm> 查看安装说明并下载安装。通常情况下，Matlab已经是被安装好的。Matlab没有被安装的情况下，从上面的地址中下载完Matlab软件后可能还需要作一些额外的操作才能正常使用，这可能需要各位自行解决。

<sup>1</sup>实际上，Python自身就提供有教程<https://docs.python.org/3/tutorial/>，中文版<http://www.pythondoc.com/python3/python3-tutorial.html>。更加深入的学习推荐Mark Lutz著的 *Learning Python, 4ed.*

## 3.2 Jupyter Notebook

本次实践中，Python代码的编程主要是在Jupyter Notebook中完成的，因此我们将先介绍什么是Jupyter Notebook，以及如何使用它来完成本次实践。

### 3.2.1 什么是Jupyter Notebook

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。Jupyter Notebook是有两部分组成：

1. 网页应用：网页应用即基于网页形式的、结合了编写说明文档、数学公式、交互计算和其他富媒体形式的工具。简言之，网页应用是可以实现各种功能的工具。
2. 文档：即Jupyter Notebook中所有交互计算、编写说明文档、数学公式、图片以及其他富媒体形式的输入和输出，都是以文档的形式体现的。这些文档是保存为后缀名为`.ipynb`的json格式文件，不仅便于版本控制，也方便与他人共享。此外，文档还可以导出为：HTML、LaTeX、PDF等格式。

——Jupyter Notebook官方介绍<sup>2</sup>

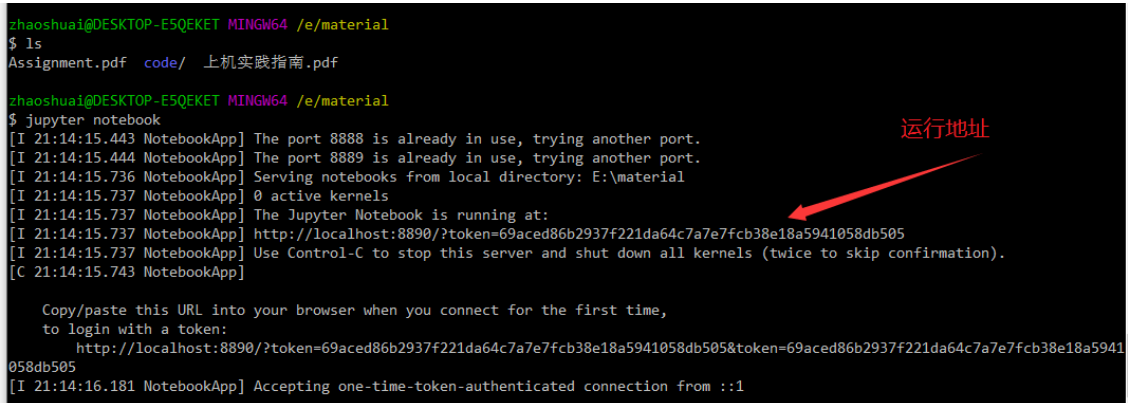
简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中直接编写代码和运行代码，代码的运行结果也会直接在代码块下显示。如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释。

### 3.2.2 Jupyter Notebook的使用

同样的，所有的操作都是在命令行窗口中进行的。

首先，切换来到本次编程实践的材料目录下，然后运行`jupyter notebook`命令即可，结果见图3。

```
jupyter notebook
```



```
zhaoshuai@DESKTOP-E5QKET MINGW64 /e/material
$ ls
Assignment.pdf  code/  上机实践指南.pdf

zhaoshuai@DESKTOP-E5QKET MINGW64 /e/material
$ jupyter notebook
[I 21:14:15.443 NotebookApp] The port 8888 is already in use, trying another port.
[I 21:14:15.444 NotebookApp] The port 8889 is already in use, trying another port.
[I 21:14:15.736 NotebookApp] Serving notebooks from local directory: E:\material
[I 21:14:15.737 NotebookApp] 0 active kernels
[I 21:14:15.737 NotebookApp] The Jupyter Notebook is running at:
[I 21:14:15.737 NotebookApp] http://localhost:8890/?token=69aced86b2937f221da64c7a7e7fcb38e18a5941058db505
[I 21:14:15.737 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:14:15.743 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8890/?token=69aced86b2937f221da64c7a7e7fcb38e18a5941058db505&token=69aced86b2937f221da64c7a7e7fcb38e18a5941058db505
[I 21:14:16.181 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

图 3: 运行`jupyter notebook`命令，正常情况下浏览器会自动弹出。没有的话复制这个地址到浏览器中打开即可。

当Jupyter Notebook成功运行之后，浏览器中视图如图4。

<sup>2</sup><https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>



图 4: 运行中的Jupyter Notebook。

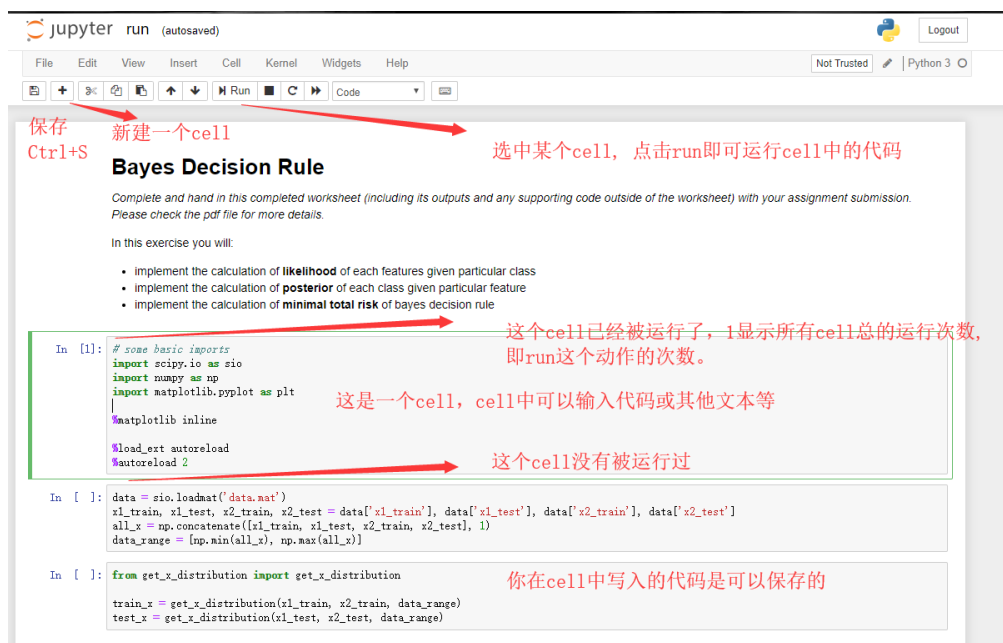


图 5: 打开一个.ipynb后缀文件后的视图。

点击进入code文件夹，此处以第一个任务为例，双击打开01.bayes\_decision\_rule文件夹下的run.ipynb文件的视图如图 5。

你在前面cell创建的变量，在后面的cell中是可以被访问的，见图 6。

好了，到现在为止，你就可以开始探索和享受你的实践工作了。

## 4 实验结果评定

这一章将介绍如何对你的实验结果进行评定。

### 4.1 助教评定

你可以将你的所有实验结果汇集写入到一个pdf文件中，然后命名为 **2019机器学习实践报告-姓名.pdf** 发送到邮箱 21721043@zju.edu.cn。邮件主题名请务必和pdf名称保持一致。助教会及时批改你的报告并回复。这个评定方式的有效日期为北京时间**2019.07.12晚23:59:59**之前，以初次收到邮件的时间为准。

```
In [1]: # some basic imports
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
%load_ext autoreload
%autoreload 2

In [2]: data = sio.loadmat('data.mat')
x1_train, x1_test, x2_train, x2_test = data['x1_train'], data['x1_test'], data['x2_train'], data['x2_test']
all_x = np.concatenate([x1_train, x1_test, x2_train, x2_test], 1)
data_range = (np.min(all_x), np.max(all_x))

In [4]: from get_x_distribution import get_x_distribution

train_x = get_x_distribution(x1_train, x2_train, data_range)
test_x = get_x_distribution(x1_test, x2_test, data_range)

print('Hello')
print('The shape of the x1_train is {}'.format(x1_train.shape))

Hello
The shape of the x1_train is (1, 400)
```

图 6: 在一个cell中添加代码并运行。这里，在依次运行了这三个cell后，然后添加输出代码再运行第四次，所以显示运行次数为4。

超过时间，邮件主题名不符，pdf名称不符，你可能将不会得到回复。助教超过3天没有及时回复，可以发送邮件催促。

## 4.2 自我评定

实际上，这些任务你是可以进行自我评定的。下面将给出每个任务的评定标准。

### 1. *bayes\_decision\_rule*。

- (a) 你可以从<https://www.bilibili.com/video/av25648623>和[https://en.wikipedia.org/wiki/Monty\\_Hall\\_problem](https://en.wikipedia.org/wiki/Monty_Hall_problem)中得到解答。或者你可以直接写一个程序来模拟换或者不换两种情况下，中奖的概率，从而得出结论。
- (b) (i) test error 小于30%; (ii) test error 小于20%; (iii) minimal risk 小于100。

### 2. *text\_classification*。 accuracy、precision、recall 均大于95%。

### 3. *neural\_networks*。数字的识别正确率(accuracy)在90%以上。

### 4. *knn*: K 越小，decision boundary 会越贴近数据的边界。验证码的识别正确率自然越高越好。



## 附录

附录中将介绍一些可能会用到的Python函数接口。

# A Numpy

附上一份Numpy官方入门教程的中文译本<https://juejin.im/post/5a76d2c56fb9a063557d8357>。

导入numpy模块并使用np的别名来使用他

```
import numpy as np
```

## A.1 np.array()

```
numpy.array(object, dtype=None, copy=True, order='K', subok=False, ndmin=0)
```

Create an array.

Parameters:

**object** : array\_like

An array, **any object** exposing the array interface, an **object** whose `__array__` method returns an array, or **any** (nested) sequence.

## A.2 np.sum()

```
numpy.sum(a, axis=None, dtype=None, out=None, keepdims=<no value>)
```

Sum of array elements over a given axis.

Parameters:

**a** : array\_like

Elements to **sum**.

**axis** : None or int or tuple of ints, optional

Axis or axes along which a **sum** is performed.

**keepdims** : bool, optional. If this is set to True, the axes which are reduced are left in the result as dimensions with size one. With this option, the result will broadcast correctly against the **input** array.

Examples:

```
>>> np.sum([[0,1], [0, 5]])
6
>>> np.sum([[0, 1], [0, 5]], axis=0)
array([0, 6])
>>> np.sum([[0, 1], [0, 5]], axis=1)
array([1, 5]), its shape is (2, ).
>>> np.sum([[0,1], [0, 5]], axis=1, keepdims=True)
array([[1], [5]]), its shape is (2, 1).
```

### A.3 np.min()

```
numpy.min(a, axis=None, out=None, keepdims=<no value>)
```

Return the minimum of an array or minimum along an axis.

Examples:

```
>>> a = np.arange(4).reshape((2,2))
>>> a
array([[0, 1],
       [2, 3]])
>>> np.min(a)           # Minimum of the flattened array
0
>>> np.min(a, axis=0)   # Minima along the first axis
array([0, 1])
>>> np.min(a, axis=1)   # Minima along the second axis
array([0, 2])
```

### A.4 np.matmul()

```
numpy.matmul(x1, x2)
```

Matrix product of two arrays.

Examples:

```
>>> a = np.array([[1, 0],
...               [0, 1]])
>>> b = np.array([[4, 1],
...               [2, 2]])
>>> np.matmul(a, b)
array([[4, 1],
```

```
[2, 2]])
```

## A.5 np.multiply()

```
numpy.multiply(x1, x2)
```

Multiply arguments element-wise..

Examples:

```
>>> x1 = np.arange(9.0).reshape((3, 3))
>>> x2 = np.arange(3.0)
>>> np.multiply(x1, x2) (This is equal to x1 * x2)
array([[ 0.,  1.,  4.],
       [ 0.,  4., 10.],
       [ 0.,  7., 16.]])
>>> x1 * x2
array([[ 0.,  1.,  4.],
       [ 0.,  4., 10.],
       [ 0.,  7., 16.]])
```

## A.6 np.log()

```
numpy.log(x)
```

Natural logarithm, element-wise.

The natural logarithm `log` is the inverse of the exponential function, so that  $\log(\exp(x)) = x$ . The natural logarithm is logarithm in base  $e$ .

Examples:

```
>>> np.log([1, np.e, np.e**2, 0])
array([ 0.,  1.,  2., -Inf])
```

## A.7 np.tile()

```
numpy.tile(A, reps)
```

Construct an array by repeating A the number of times given by reps.

If reps has length d, the result will have dimension of `max(d, A.ndim)`.

Examples:

```
>>> a = np.array([0, 1, 2])
>>> np.tile(a, 2)
array([0, 1, 2, 0, 1, 2])
>>> np.tile(a, (2, 2))
array([[0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2]])
>>> np.tile(a, (2, 1, 2))
array([[[0, 1, 2, 0, 1, 2]],
       [[0, 1, 2, 0, 1, 2]]])

>>> c = np.array([1,2,3,4])
>>> np.tile(c, (4,1))
array([[1, 2, 3, 4],
       [1, 2, 3, 4],
       [1, 2, 3, 4],
       [1, 2, 3, 4]])
```

## A.8 np.argsort()

```
numpy.argsort(a, axis=-1, kind='quicksort', order=None)
```

Returns the indices that would sort an array.

Perform an indirect sort along the given axis using the algorithm specified by the kind keyword.

It returns an array of indices of the same shape as a that index data along the given axis in sorted order.

Examples:

```
>>> x = np.array([3, 1, 2])
>>> np.argsort(x)
array([1, 2, 0])

>>> x = np.array([[0, 3], [2, 2]])
>>> x
array([[0, 3],
       [2, 2]])
>>> np.argsort(x, axis=0)  # sorts along first axis (down)
```

```
array([[0, 1],
       [1, 0]])
>>> np.argsort(x, axis=1)  # sorts along last axis (across)
array([[0, 1],
       [0, 1]])
```

## A.9 np.expand\_dims()

```
numpy.expand_dims(a, axis)
```

Expand the shape of an array.

Insert a new axis that will appear at the axis position  
in the expanded array shape.

Examples:

```
>>> x = np.array([1,2])
>>> x.shape
(2,)
>>> y = np.expand_dims(x, axis=0)
>>> y
array([[1, 2]])
>>> y.shape
(1, 2)
>>> y = np.expand_dims(x, axis=1)  # Equivalent to x[:,np.newaxis]
>>> y
array([[1],
       [2]])
>>> y.shape
(2, 1)
```

## B Scipy

### B.1 scipy.stats.mode()

```
scipy.stats.mode(a, axis=0, nan_policy='propagate')
```

Return an array of the modal (most common) value in the passed array.

If there is more than one such value, only the smallest is returned.

The `bin-count` for the modal bins is also returned.

Examples:

```
>>> a = np.array([[6, 8, 3, 0],
...               [3, 2, 1, 7],
...               [8, 1, 8, 4],
...               [5, 3, 0, 5],
...               [4, 7, 5, 9]])
>>> from scipy import stats
>>> stats.mode(a)
(array([[3, 1, 0, 0]]), array([[1, 1, 1, 1]]))
```

## C Matplotlib

实践中基本不会有要求写关于这个包的代码，所以此处没有关于其api的介绍。在此给出一个教程链接<https://www.runoob.com/w3cnote/matplotlib-tutorial.html>。实际上本次实践课程用到的Python工具都是较为常见的工具，网络上有很多教程。