你好，欢迎来到数据挖掘的第三课，今天要讲的内容是 Python 中与数据挖掘有关的扩展包，以及其环境。

在 02 课时，我简单介绍了 Python 的数据结构和基本语法，让你对其有了一个基本的认识。本课时，我将从 Python 标准库和第三方扩展库，以及如何搭建 Python 环境入手，让你可以更加顺畅地学好数据挖掘。

这部分内容有很多知识点可以讲，但详细介绍这些并不是本专栏的目的，我希望你学会的是建设数据挖掘 Python 环境的基本思路，这样当你遇到对应的问题时，就不必冥思苦想，而是能快速找到适合自己项目的工具。

标准库

Python 的标准库是其核心的扩展，其中包括了操作系统接口、文件操作、输入输出流、文本处理等功能。

这里先推荐两个能帮助我们学习常用包的方法：

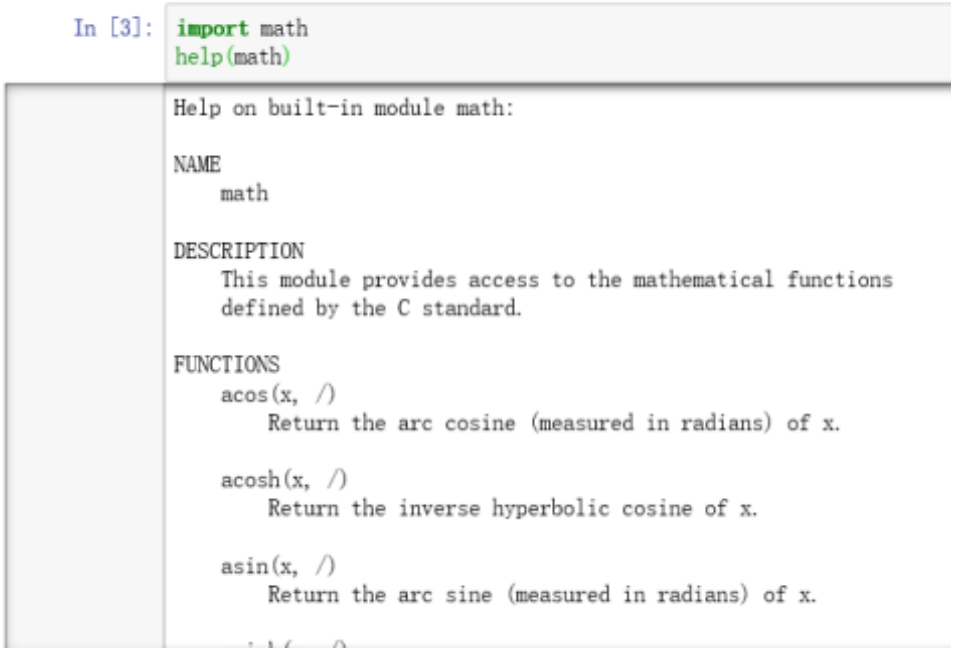
比如，对 math 模块使用 dir(math)，可以看到 math 模块里所有方法的名称，结果如下：

```
In [4]: import math
        dir(math)

Out[4]: ['__doc__',
         '__loader__',
         '__name__',
         '__package__',
         '__spec__',
         'acos',
         'acosh',
         'asin',
         'asinh',
         'atan',
         'atan2',
         'atanh',
         'ceil',
         'copysign',
         'cos',
         'cosh',
         'degrees',
         'e',
         'erf',
         'erfc',
         'exp',
         'expm1',
         'fabs',
         'factorial',
         'floor',
         'fmod',
         'frexp',
         'fsum',
         'gamma',
         'gcd',
```

math 模块里面所有方法的名称图

使用 help(math)，可以看到 math 模块的描述，以及各个方法的介绍， 结果如下：



math 模块的描述图

在写代码的时候如果忘记某个功能的名字或者不知道有哪些功能，就可以通过上面两个方法去查看。

接下来，我要介绍一下数据挖掘中常用的 6 个模块，及其应用场景，请见下表：

模块名称	模块简介	应用场景
数学模块（math）	包含很多科学计算方法，如平方根、对数计算、三角函数，等等	在数据挖掘中，经常要对**数据进行标准化**、求统计值等处理，math 模块基本上包含了所用的基本操作
日期时间模块（datetime）	主要用于处理时间类型的数据，如时间数据格式化、时间的获取、时间数据与字符串的转换，等等	数据通常都会带有时间戳，有时，时间也是一种重要的特征。如新闻中，有新闻的发生时间、发布时间等，就会用到该模块
随机模块（random）	主要可以进行随机数的生成，随机选取	在进行**数据采样**、**数据生成**时经常会用到这些随机方法
文件操作模块（file）	主要提供了文件操作，包括文件的读取和写入等，在处理本地数据时，通常都会用到这些操作	数据挖掘的样本通常都会被存放在文件中，所以文件操作也是基本技能之一
正则匹配模块（re）	可以使用正则表达式来进行字符串的匹配、检测等，其编写方法可以在网上搜索一下	在处理文本数据时，经常需要用到正则匹配来进行**文本的检索**
系统接口模块（sys）	主要实现了与操作系统交互的一些功能，如获取当前操作系统的情况、设置编码格式等，编写完整的程序通常都会用到	系统接口模块主要是为了**获取**系统的各种数据

第三方库

除了有应用广泛的标准库，Python 的魅力之一就是拥有庞大的第三方库，代码之丰富，大大简化了大家开发的过程。这里我列出了在数据挖掘、机器学习项目中一些常用的项目库，如果你对某一个库感兴趣，想要深入学习，可以先从官网入手了解。当然，第三方库的内容远远不止这么一点，随着工作的深入，你将会接触到各种各样的第三方库，感受到什么是众人拾柴火焰高。

- 基础模块

我将给你推荐 4 款常用且功能强大的科学运算基础工具包，请见下表：

名称	含义
NumPy	Python 语言扩展程序库，支持大量的维度数组与矩阵运算。
SciPy	集成了数学、科学和工程的计算包， 它用于有效计算 Numpy 矩阵，使 Numpy 和 Scipy 协同工作。
Matplotlib	专门用来绘图的工具包，可以使用它来进行数据可视化。
pandas	数据分析工具包，它基于 NumPy 构建，纳入了大量的库和标准数据模型。

- 机器学习

机器学习常用的库也有 4 个，包含了基础数据挖掘、图像处理与自然语言处理常用算法。它们可以支撑日常工程中的常见算法处理方案，所以非常推荐你使用，请见下表：

名称	含义
scikit-learn	基于 SciPy 进行延伸的机器学习工具包，包含大量的机器学习算法模型，有 6 大基本功能：分类、回归、聚类、数据降维、模型选择和数据预处理。
OpenCV	非常庞大的图像处理库，实现了非常多的图像和视频处理方法，如图像视频加载、基础特征获取、边缘检测等，处理图像通常都需要其支持。
NLTK	比较传统的自然语言处理模块，自带很多语料，以及全面的传统自然语言处理算法，比如字符串处理、卡方检验等，非常适合**自然语言入门**使用。
Gensim	包含了浅层词嵌入的文本处理模块，以及常用的自然语言处理相关方法，如 TF-IDF、word2vec 等模型。

- 深度学习平台

这里我再介绍 3 个深度学习的平台，你可以根据自己的需求进行了解，请见下表：

平台名称	开发平台	优点
TensorFlow	谷歌	相对成熟、应用广泛、服务全面、提供学习视频和其认证计划。
PyTorch	Facebook	支持更加快速地构建项目。
PaddlePaddle	百度	中文文档全面，对于汉语的相关模型比较丰富。

使用任何一个框架都可以构建深度学习项目，在实际的应用中，根据自己的需要进行选择即可。

除了上面介绍的模块，还有很多相关的模块，在这里我就不一一介绍了，等到具体应用时我会针对相应的算法再讲一些其他的模块。接下来，我们介绍一种模块的安装方法。

使用 pip 安装扩展包

pip 是一个特殊的模块，可以用它来安装扩展包。使用 pip 可以对 Python 扩展包进行查找、下载、安装、卸载等。在 Python 3.6 中，pip 已经成为一个自带的模块，如果你不确定你的 Python 中是否有该模块，可以执行以下命令：

升级 pip 到最新版，命令如下：

用 pip 安装扩展包，以安装 TensorFlow 为例，命令如下：

```
pip install tensorflow


pip install tensorflow==1.14


pip install tensorflow>=1.14
```

用 pip 卸载某个模块，命令如下：

在 pip 库中搜索某个模块，命令如下：

用 pip 显示已安装的包，命令如下：

由于某些原因，使用 pip 自带的镜像源可能会出现让你抓狂的下载中断问题，我们可以自己配置成国内的镜像源。在安装某个模块时，如需临时切换镜像源，命令如下：

```
pip install tensorflow -i https://pypi.tuna.tsinghua.edu.cn/simple
```

用 pip 更新配置文件，修改默认源，命令如下：

```
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

常用镜像源

镜像源在数据挖掘中也比较常用，我列举了一些常用的，就不一一介绍了，如果你感兴趣可以去了解一下，请见下表：

镜像名称	网站地址
阿里云	https://mirrors.aliyun.com/pypi/simple/
中国科技大学	https://pypi.mirrors.ustc.edu.cn/simple/
清华大学	https://pypi.tuna.tsinghua.edu.cn/simple/
豆瓣	http://pypi.douban.com/simple/
华中理工大学	http://pypi.hustunique.com/simple/
山东理工大学	http://pypi.sdutlinux.org/simple/

由于 Python 方便、好用，所以在机器学习、数据挖掘领域受到了广泛的追捧。但是各种资源代码包层出不穷，更新的包多、频率快，就会出现缺少对应代码包、版本不兼容等问题，因此选择一款好用的编辑器就变得非常重要了。

为了避免不必要的问题，我为你推荐一款叫作 Anaconda 的软件。

什么是 Anaconda

Anaconda 是包管理器，也是环境管理器，更是 Python 的编辑器。其致力于为用户提供最便捷的方式来使用 Python，进行数据科学计算和机器学习。这个免费的软件安装起来非常方便，涵盖的源码包、工具包之多，以及适用的平台之广，使得该软件在安装、运行和升级等复杂的科学数据运算和机器学习环境方面变得极其简单。

当前流行的三个开源软件库 sklearn、TensorFlow 和 sciPy 都支持 Anaconda，不仅如此，你还可以在网上找到该软件的 免费交流论坛，随时进行讨论学习。

因为这是一个开源的工具，所以它拥有众多用户。许多数据科学运算工程都在使用 Anaconda，其中不乏一些大公司的项目，例如 Amazon Web Services' Machine Learning AMIs、Anaconda for Microsoft on Azure and Windows，等等。

为什么要用 Anaconda

当我第一次接触到 Anaconda 的时候，就被它深深地吸引了，它很大程度上解决了我之前提到的 Python 资源更新多且快的痛点。

- **依赖包安装方便**
  预装 150+ 依赖包，提供 250+ 可选开源依赖包，可以直接使用命令 conda install，也可以使用 pip install 命令安装，非常方便。甚至可以使用 conda build 来构建你自己的依赖包，之后把它上传到 anaconda cloud、PyPi 或者其他的资源站上面，分享给大家使用。
- **多平台支持**
  在日常使用的 Linux 系统、Windows 系统和 MacOS 系统上都有对应的 Anaconda 版本，不管是 32 位还是 64 位都是可以的。不光如此，还有图形界面版本，对使用图形桌面系统的同学很友好
- **多环境切换**
  使用 Anaconda 可以依据不同的项目依赖构建多套互不干扰的环境，随时切换，而不用担心各个环境之间的冲突。

不仅如此，使用 Anaconda 配置好的环境还可以进行打包储存，迁移项目到其他机器上的时候，只需要把打包的环境一并移到新的服务器上，就可以一键安装整个已经配置好的环境，不需要再重新建设了，非常方便。

你现在一定迫不及待地想要使用 Anaconda，下面我就来介绍一下它的常用功能。

如何使用 Anaconda

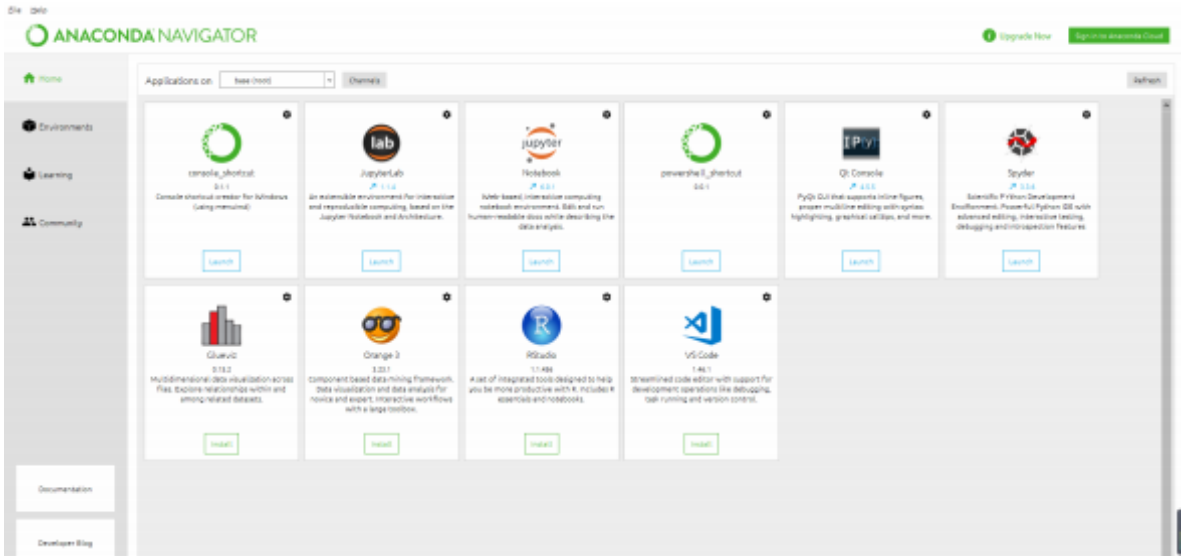
进入 Anaconda 的官网，根据自己的需求下载对应的 Anaconda 版本。

Anaconda 官网图

本课时我就以 64 位的 Windows 版本为例，给大家演示如何安装 Anaconda。

安装的过程是傻瓜式的，基本上是一路 “Next” 和“I agree”就可以结束了。如果你想使用命令行安装模式，那么可以选择 shell 命令来进行控制，也是十分简单的。

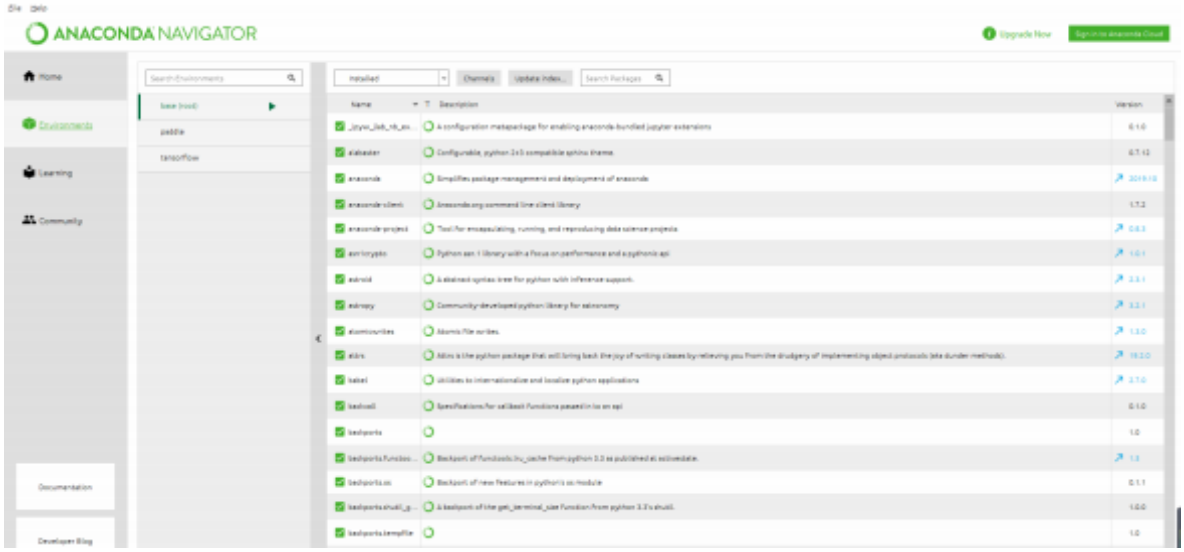
安装完成后，我们打开 Anaconda 的 Navigator，可以看到如下的界面。



Anaconda 的界面图

这里会有一个初始化的环境 base。除了环境配置以外，内部还搭载了很多实用的模块，包括 Jupyter Notebook、Spyder、Qt Console 等，对这些内容我就不再做详细介绍了，感兴趣的同学可以去官网了解学习。

如果我们想使用这个界面工具去创建环境，可以切换到 Environments 的 tab 页：



Environments 的 tab 页图

你可以使用页面下方的 Create 按钮去创建一个新的环境，也可以用命令行的方式创建，具体操作如下：

（1）在命令行中配置 Anaconda 环境；

（2）在 Windows 中打开命令行工具 CMD；

（3）使用 conda 创建一个环境，执行命令如下。

```
conda create --name python-my python=3.7
```

接下来就是自动安装环节，静静等待安装完成，然后来激活这个环境，命令如下：

激活之后，这些配置环境变量的内容就生效了，命令也都可以直接使用。如果要安装一些包，可以直接使用 Anaconda 的命令代替 pip，比如：

如果我们想要切换到其他的环境，可以使用下面的命令来注销当前环境：

完成注销之后再使用激活方法来激活新的环境。

如果你想看一下电脑上有没有其他的环境，可以使用以下命令：

在环境已激活的情况下使用 conda 导出已有环境，导出命令如下：

```
conda env export > environment.yaml
```

在创建环境的时候，直接使用移到其他服务器上的 yaml 文件即可：

```
conda env create -f environment.yaml
```

激活环境，然后打开 Spyder 试试前面是否已经安装成功了，执行命令如下：

```
import numpy as np

import matplotlib.pyplot as plt

labels = ['G1', 'G2', 'G3', 'G4', 'G5']

men_means = [20, 35, 30, 35, 27]

women_means = [25, 32, 34, 20, 25]

men_std = [2, 3, 4, 1, 2]

women_std = [3, 5, 2, 3, 3]

width = 0.35

fig, ax = plt.subplots()

ax.bar(labels, men_means, width, yerr=men_std, label='Men')

ax.bar(labels, women_means, width, yerr=women_std, bottom=men_means,

       label='Women')
```

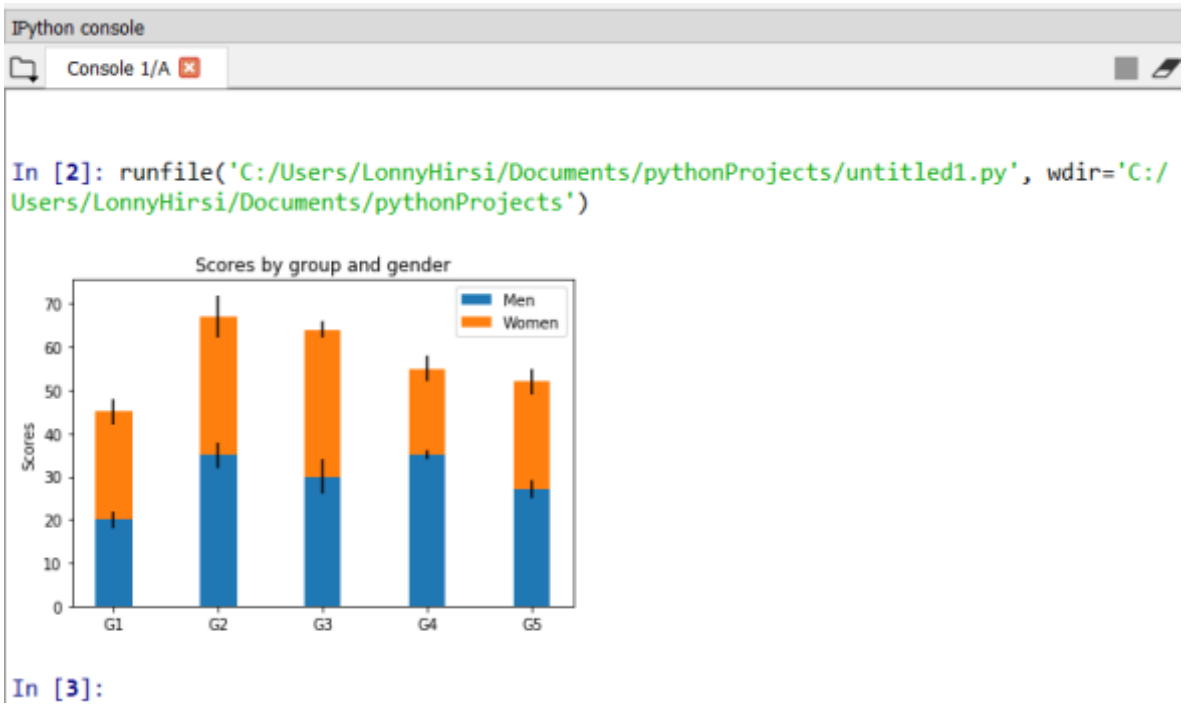
```
ax.set_ylabel('Scores')

ax.set_title('Scores by group and gender')

ax.legend()

plt.show()
```

如果成功运行，将在界面右侧的输出框显示一个柱状图：



柱状图

总结

这一课时，我主要介绍了 Python 的标准库和扩展库中常用于数据挖掘的功能模块，然后讲了使用 Anaconda 来搭建环境的方法。

整个内容涉及的东西比较多，对每一个知识点的介绍也都比较粗略，但是希望大家能够在脑海中有一个印象，就是我们使用 Python 可以做什么，这样就不会在遇到问题的时候没有想法了。如果你在日后的工作中遇到了问题，知道去哪里寻求解决问题的方法，我觉得本课的目的就达到了。

如果你对里面的某些模块很有兴趣，可以去他们的首页上进行详细的了解，也可以在网上找一些学习资料，如够买相关书和课程，也欢迎你在留言区写下你的问题，与我和其他同学一起讨论。