



《机器学习的编程设计》课程简介

曹相湧

电信学部计算机学院



自我介绍

曹相湧

电信学部计算机学院

Email: caoxiangyong@xjtu.edu.cn

研究方向

机器学习、图像处理

2008.09-2012.07	西安交通大学	信息与计算科学	理学学士
2012.09-2018.06	西安交通大学	应用数学(硕博连读)	理学博士
2016.09-2017.09	哥伦比亚大学	数据科学中心	访问学者
2018.07-2021.12	西安交通大学	数学与统计学院	助理教授
2021.12-2022.04	西安交通大学	计算机学院	助理教授
2022.04-至今	西安交通大学	计算机学院	副教授

团队主页: <https://gr.xjtu.edu.cn/web/dymeng/1>

个人主页: <https://gr.xjtu.edu.cn/en/web/caoxiangyong>



重要事项

- 上课时间：9-17周 周四3-4，5-6节
- 9-12周：曹相湧 13-17周：孙凯
- 助教：任鹏飞
- 关于成绩：
 - 平时作业(60分)
 - 期末大作业(40分)
 - 完全抄袭其它同学代码，经发现，**双方均0分！**
 - 可以参考课件、网上资源等



四门机器学习课程

- 贝叶斯统计，赵谦（大四）
- 机器学习，孟德宇（大三、研一上）
- 人工智能与深度学习，谢琦（研一上）
- **机器学习的编程设计**，曹相湧 孙凯（研一下）



课程内容简介

- 授课方式：课堂理论讲解
- 教学目的：培养应用机器学习方法的能力
- 本课程主要讨论机器学习方法，如支持向量机、随机森林、梯度增强和神经网络在真实世界数据集上的应用，包括数据准备、模型选择和模型评估等
- 参考教材：

[1] Python机器学习基础教程. 安德里亚斯穆勒, 莎拉吉多著, 人民邮电出版社.

[2] Müller, Andreas C., and Sarah Guido. Introduction to machine learning with Python: a guide for data scientists.

教学大纲



小节↵	主要内容↵	学时↵	任课教师↵
第一节↵	有监督学习↵	2 学时↵	曹相湧↵
第二节↵	数据预处理↵	2 学时↵	曹相湧↵
第三节↵	线性回归↵	2 学时↵	曹相湧↵
第四节↵	线性分类、支撑 <u>向量机</u> ↵	2 学时↵	曹相湧↵
第五节↵	随机森林↵	2 学时↵	曹相湧↵
第六节↵	集成方法↵	2 学时↵	曹相湧↵
第七节↵	梯度增强↵	2 学时↵	曹相湧↵
第八节↵	模型评估↵	2 学时↵	曹相湧↵
第九节↵	不平衡数据的学习↵	2 学时↵	孙凯↵
第十节↵	模型解释和特征选择↵	2 学时↵	孙凯↵
第十一节↵	参数调节和自动机器学习↵	2 学时↵	孙凯↵
第十二节↵	维数减少↵	2 学时↵	孙凯↵
第十三节↵	聚类和混合模型↵	2 学时↵	孙凯↵
第十四节↵	神经网络↵	2 学时↵	孙凯↵
第十五节↵	神经网络↵	2 学时↵	孙凯↵
第十六节↵	神经网络↵	2 学时↵	孙凯↵

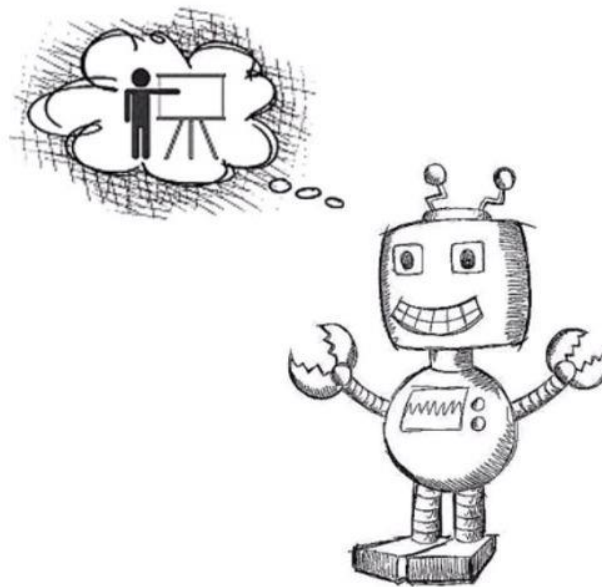


什么是机器学习

➤ 人类从过去的经验学习，机器遵循人类指示



**Humans learn from
past experiences**

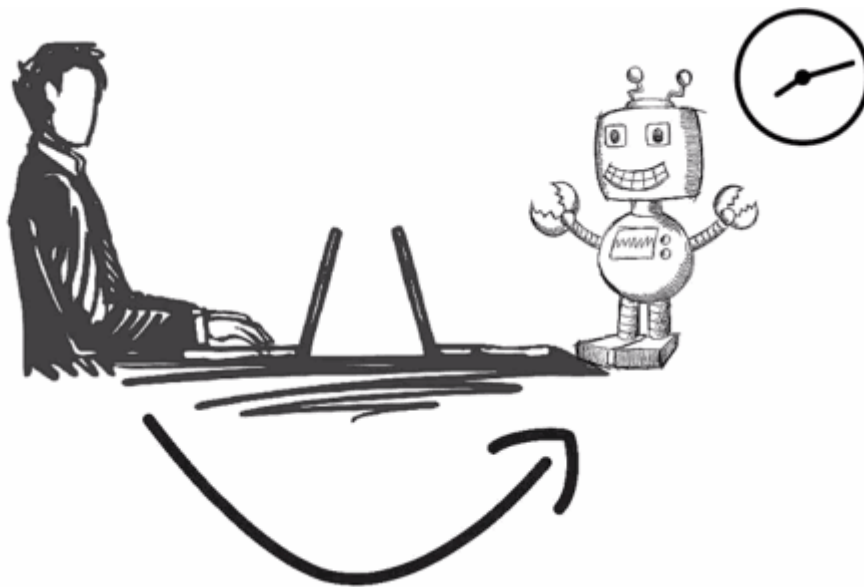


**Machine follows instruction
given by human**



什么是机器学习

➤ 机器学习：人类训练机器从过去的经验学习

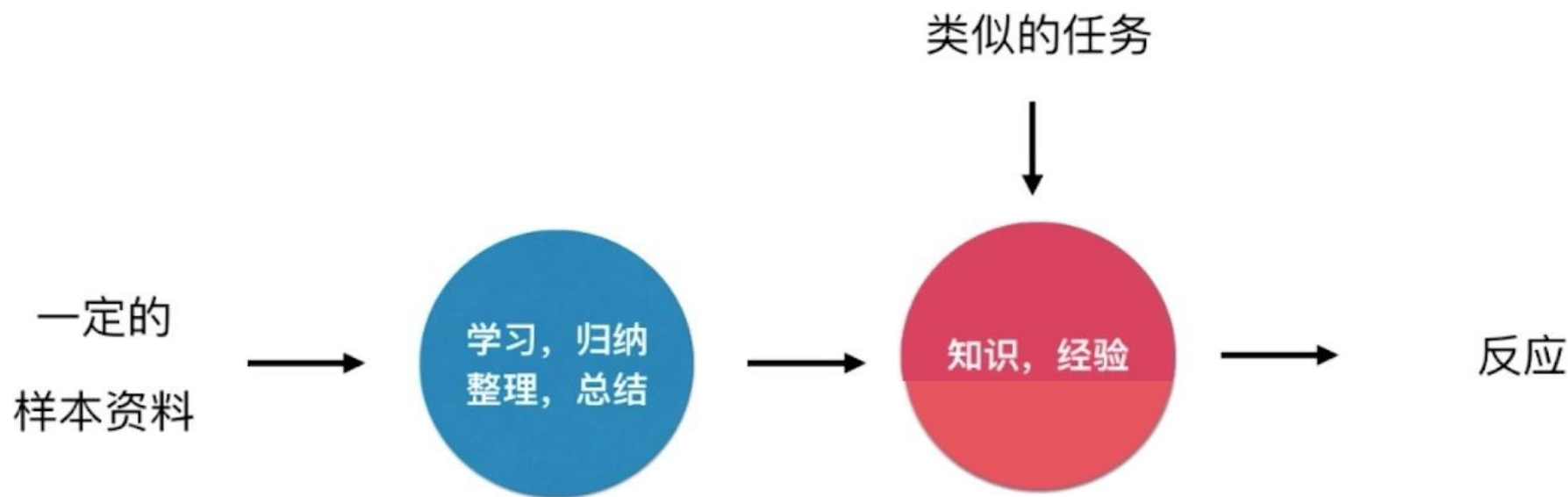


Humans train the machine to learn from past experiences



人类怎么学习

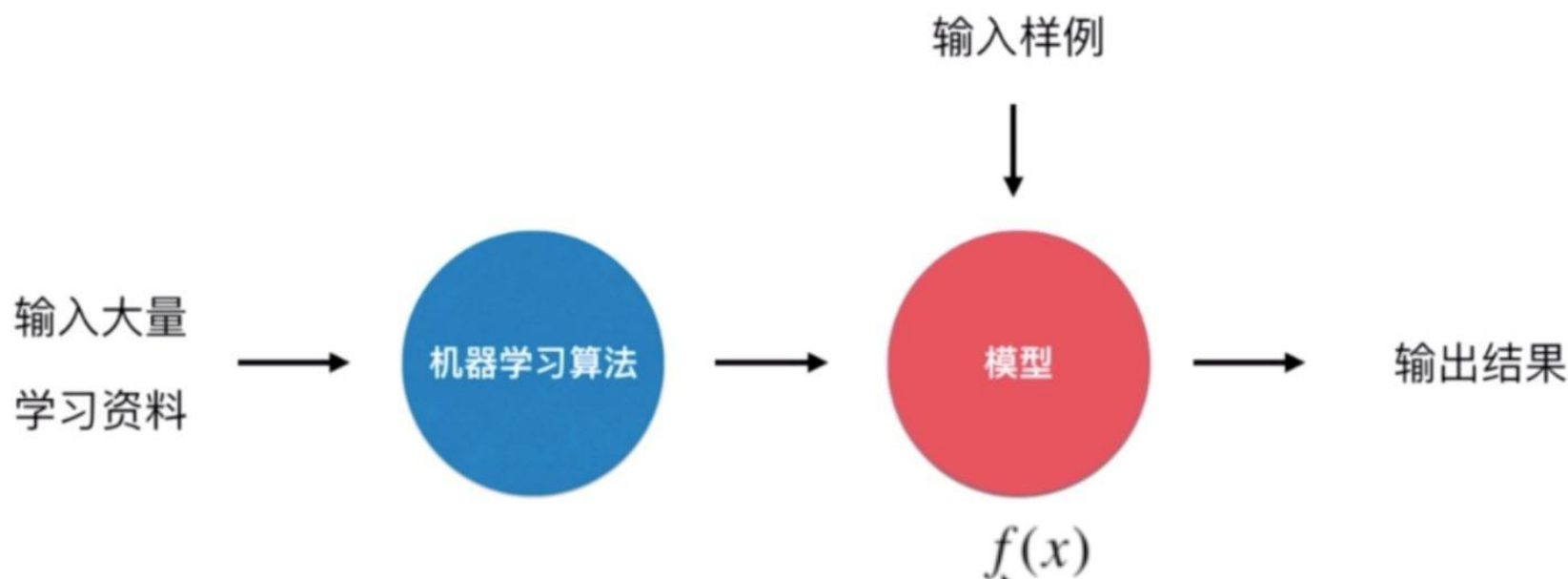
➤ 通过给大脑输入一定的资料，经过学习总结得到知识和经验，有当类似的任务时可以根据已有的经验做出决定或行动





什么是机器学习

➤ 机器学习算法目的：从数据中获得预测模型 $f(x)$



➤ 输入新样本 x^* ,

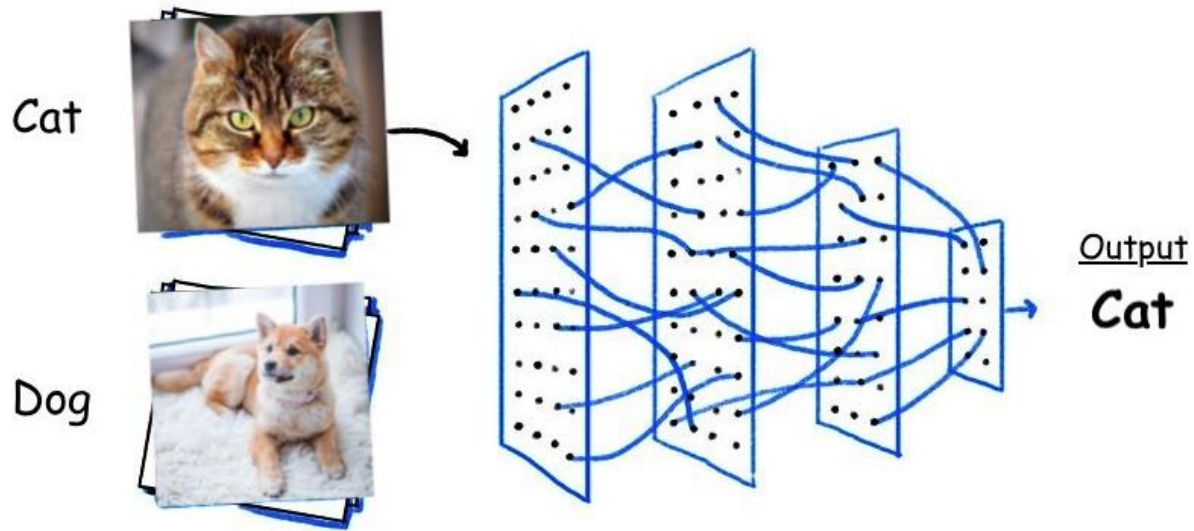
➤ 如果 $f(x^*)$ 是离散数值，解决的是分类问题

➤ 如果 $f(x^*)$ 是连续数值，解决的是回归问题



分类问题示例

图像分类



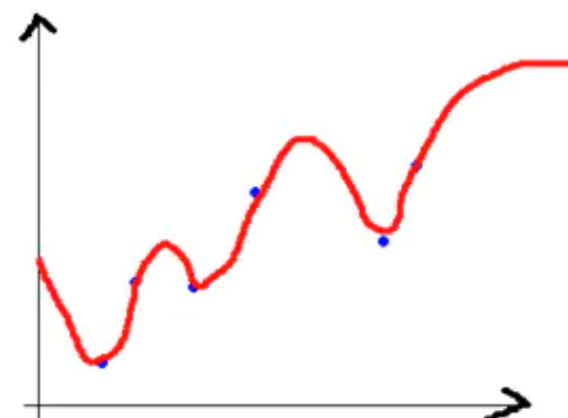
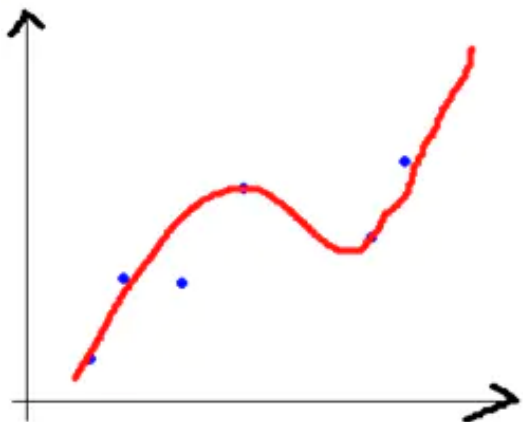
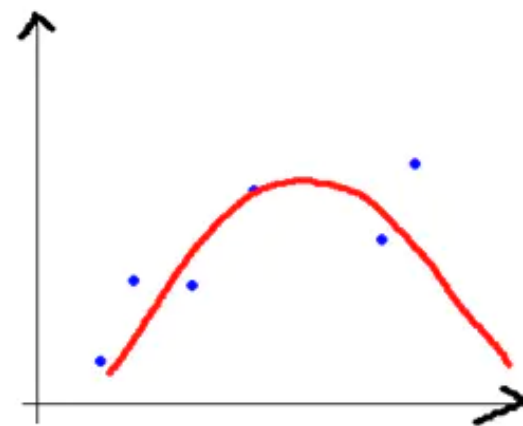
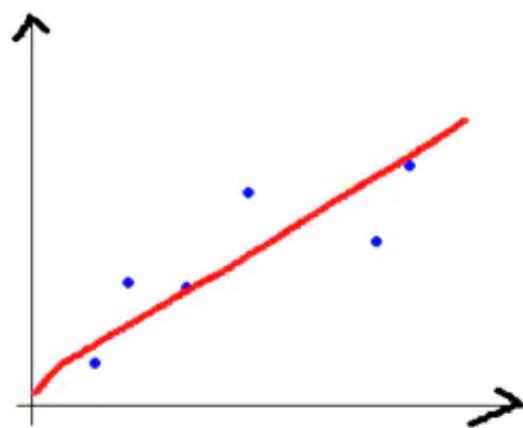
垃圾邮件分类



回归问题示例



回归问题

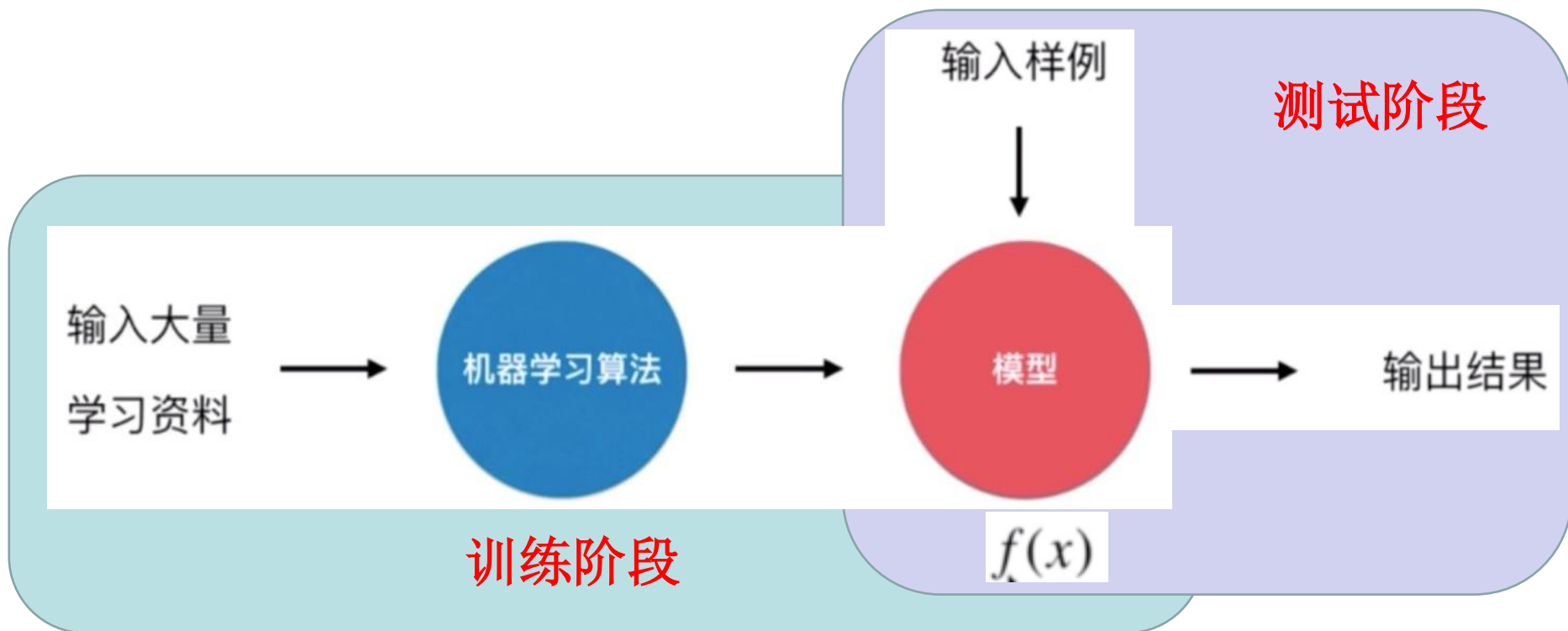




机器学习的两个阶段

➤ **训练阶段：**通过某种机器学习算法，从大量数据中获得预测模型 $f(x)$

➤ **测试阶段：**给定新样本 x^* ，输入预测模型 $f(x)$ ，得到预测结果 $f(x^*)$





机器学习的分类

- 监督学习 (supervised learning)
- 无监督学习 (unsupervised learning)
- 强化学习 (reinforcement learning)



监督学习

训练数据(对) $(x_i, y_i) \propto p(x, y)$ i.i.d. $i=1,2,\dots,n$

$x_i \in \mathbb{R}^p$ 数据

$y_i \in \mathbb{R}$ 标签

$$f(x_i) \approx y_i$$

➤ 泛化性(Generalization)

Not only $f(x_i) \approx y_i$,
also for new data: $f(x) \approx y$



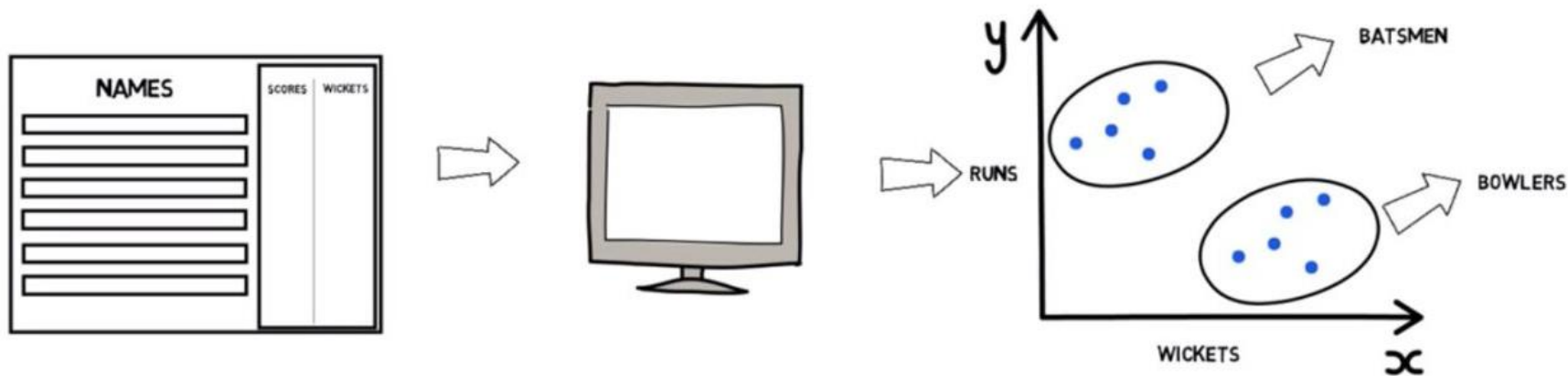
无监督学习

$$x_i \propto p(x) \text{ i.i.d. } i=1,2,\dots,n$$

只有数据，无标签

无监督学习是一种数据探索方式

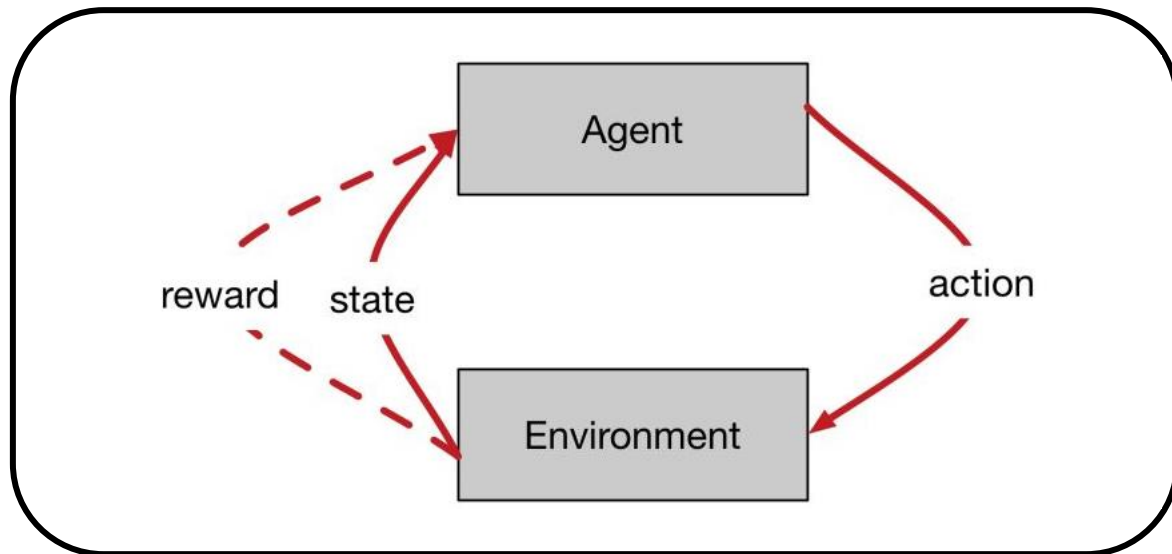
目的：从数据 x_i 中学习数据的某种模式





强化学习

➤ 强化学习：通过与环境间的交互和反馈来学习





其它类型的学习

- 半监督学习 (semi-supervised learning)
- 元学习 (meta learning)
- 自监督学习 (self-supervised learning)
-



简单练习

确定下列场景使用监督还是非监督学习：

- 场景 1: Facebook 从一张标签照片相册中识别出你的朋友
- 场景 2: Netflix 根据某人过去的电影选择推荐新电影
- 场景 3: 分析可疑交易的银行数据并标记欺诈交易



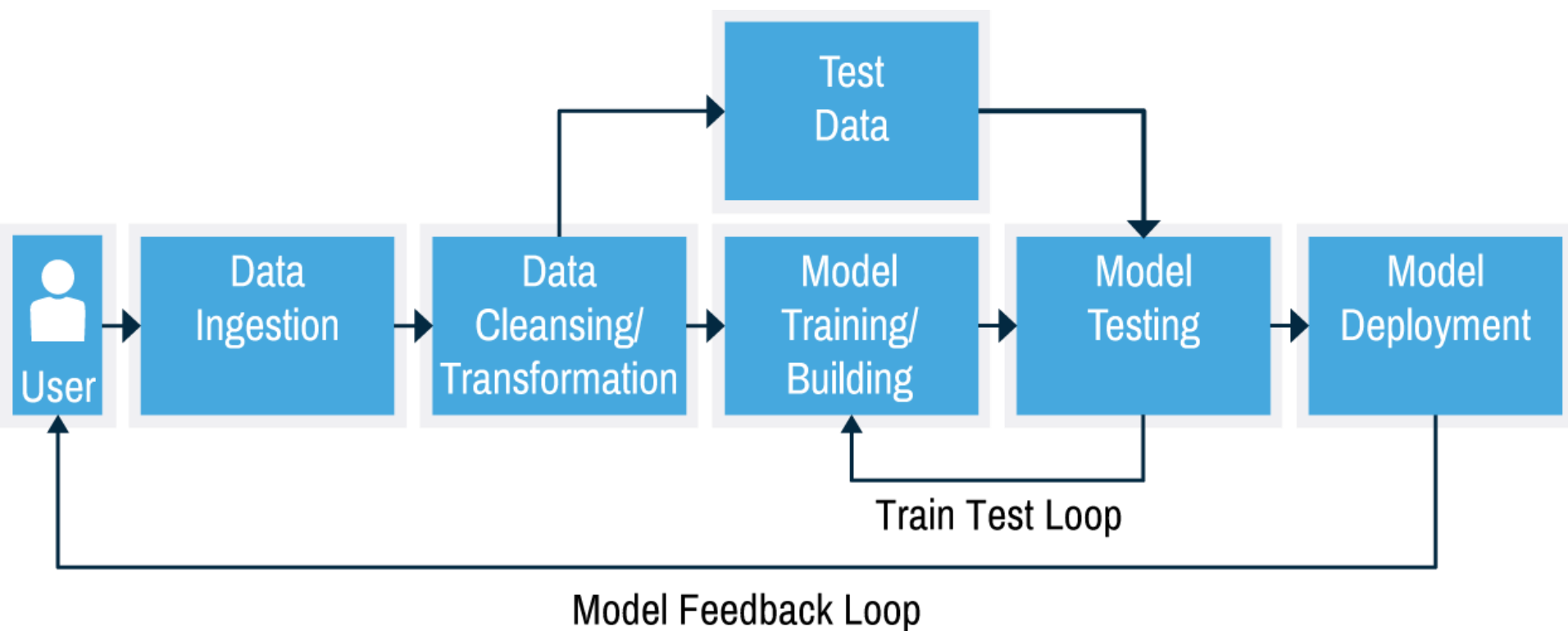
简单练习

确定下列场景使用监督还是非监督学习：

- 场景 1: Facebook 从一张标签照片相册中识别出你的朋友（有监督学习）
- 场景 2: Netflix 根据某人过去的电影选择推荐新电影（有监督学习）
- 场景 3: 分析可疑交易的银行数据并标记欺诈交易（无监督学习）



机器学习的工作流程



Taken from MAPR <https://www.mapr.com/ebooks/spark/08-recommendation-engine-spark.html>



熟悉任务和数据

构建机器学习方案时，给出下列问题答案：

- 想要回答什么问题？已收集数据能够回答吗？
- 将问题表达为机器学习问题，用哪种方法？
- 收集的数据是否足够表达想解决的问题？
- 提取了数据的哪些特征？特征能否实现正确预测？
- 如何衡量应用是否成功？
- 机器学习解决方案与我的研究或商业产品中的其它部分如何相互影响？



为什么选择Python

- 数据科学应用的通用语言
- 具有通用编程语言的强大功能，且具有易用性
- 提供数据加载、可视化、统计、自然语言处理、图像处理等各种功能的库
- 机器学习和数据分析本质上都是迭代过程，利用终端或其它类似Jupyter notebook的工具能够直接与代码进行交互



scikit-learn

- scikit-learn是最著名的Python机器学习库
- 包含许多目前最先进的机器学习算法，每个算法都有详细的文档 (<https://scikit-learn.org/stable/>)
- 广泛应用于学术界和工业界





Anaconda

- 用于大规模数据处理、预测分析和科学计算的python发行版
- Anaconda已预先安装Numpy、Scipy、matplotlib、pandas、Jupyter Notebook和scikit-learn等库和交互平台



ANACONDA

Products ▼

Pricing

Solutions ▼

Resources ▼

Pa

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform



安装Anaconda

1. Anaconda下载

历史版本地址: <https://repo.anaconda.com/archive/>

Anaconda2-5.2.0-Windows-x86_64.exe	564.0M	2018-05-30 13:04:18	695e427e4b625b6eab92b23a28dc4e21
Anaconda3-5.2.0-Linux-ppc64le.sh	288.3M	2018-05-30 13:05:40	cbd1d5435ead2b0b97dba5b3cf45d694
Anaconda3-5.2.0-Linux-x86.sh	507.3M	2018-05-30 13:05:46	81d5a1648e3aca4843f88ca3769c0830
Anaconda3-5.2.0-Linux-x86_64.sh	621.6M	2018-05-30 13:05:43	3e58f494ab9fbe12db4460dc152377b5
Anaconda3-5.2.0-MacOSX-x86_64.pkg	613.1M	2018-05-30 13:07:00	9c35bf27e9986701f7d80241616c665f
Anaconda3-5.2.0-MacOSX-x86_64.sh	523.3M	2018-05-30 13:07:03	b5b789c01e1992de55ee911754c310d4
Anaconda3-5.2.0-Windows-x86.exe	506.3M	2018-05-30 13:04:19	285387e7b6ea81edba98c011922e235a
Anaconda3-5.2.0-Windows-x86_64.exe	631.3M	2018-05-30 13:04:18	62244c0382b8142743622fdc3526eda7
Anaconda2-5.1.0-Linux-ppc64le.sh	267.3M	2018-02-15 09:08:49	e894dcc547a1c7d67deb04f6bba7223a
Anaconda2-5.1.0-Linux-x86.sh	431.3M	2018-02-15 09:08:51	e26fb9d3e53049f6e32212270af6b987
Anaconda2-5.1.0-Linux-x86_64.sh	533.0M	2018-02-15 09:08:50	5b1b5784cae93cf696e11e66983d8756

安装Anaconda



2. Anaconda 安装

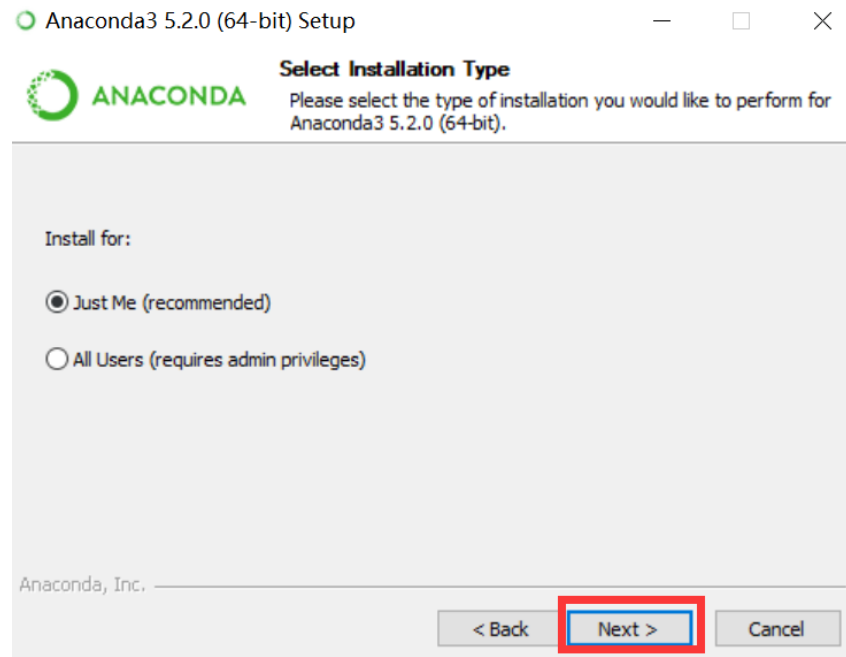
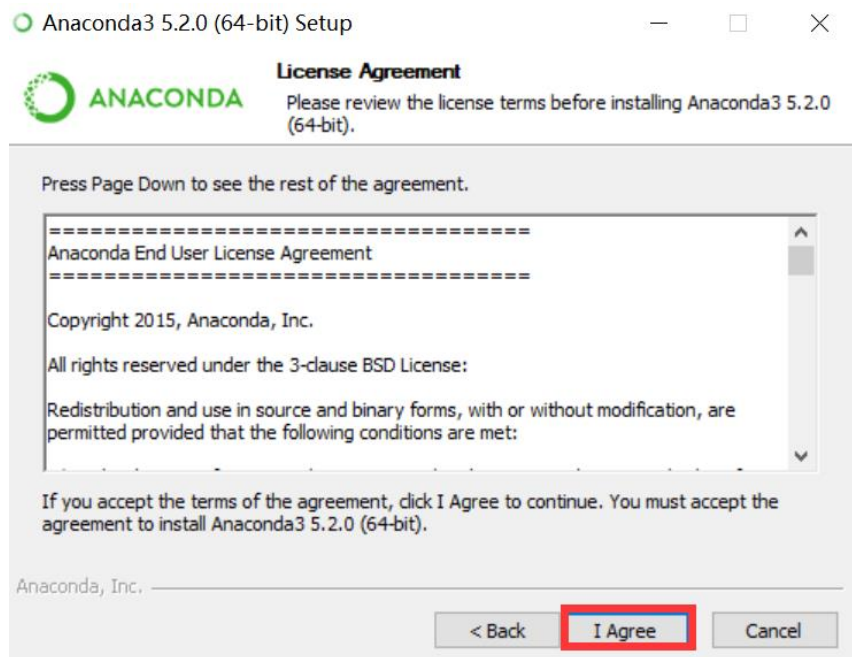
<input type="checkbox"/> 名称	修改日期	类型	大小
<input checked="" type="radio"/> Anaconda3-5.2.0-Windows-x86_64.exe	2022/4/16 20:51	应用程序	646,472 KB



安装Anaconda



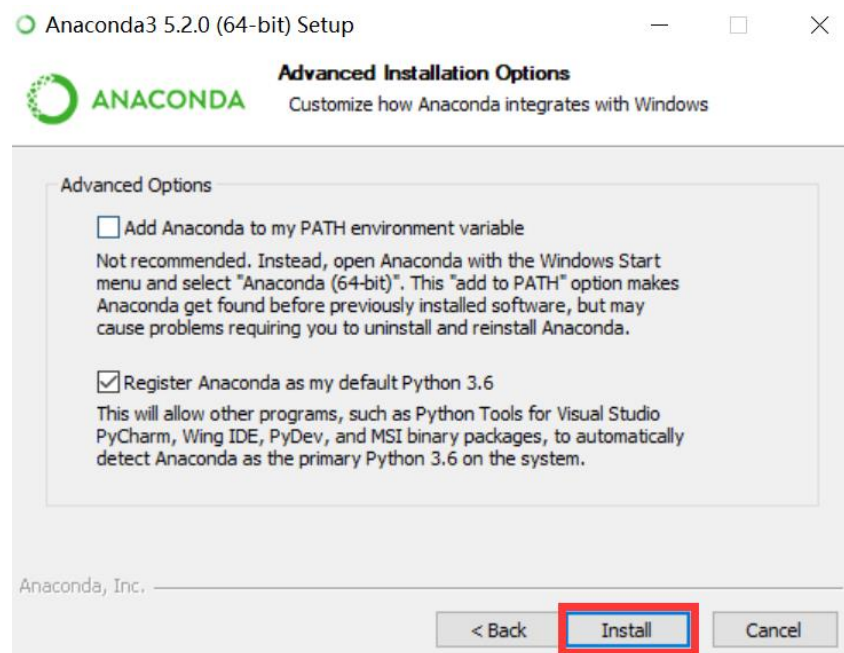
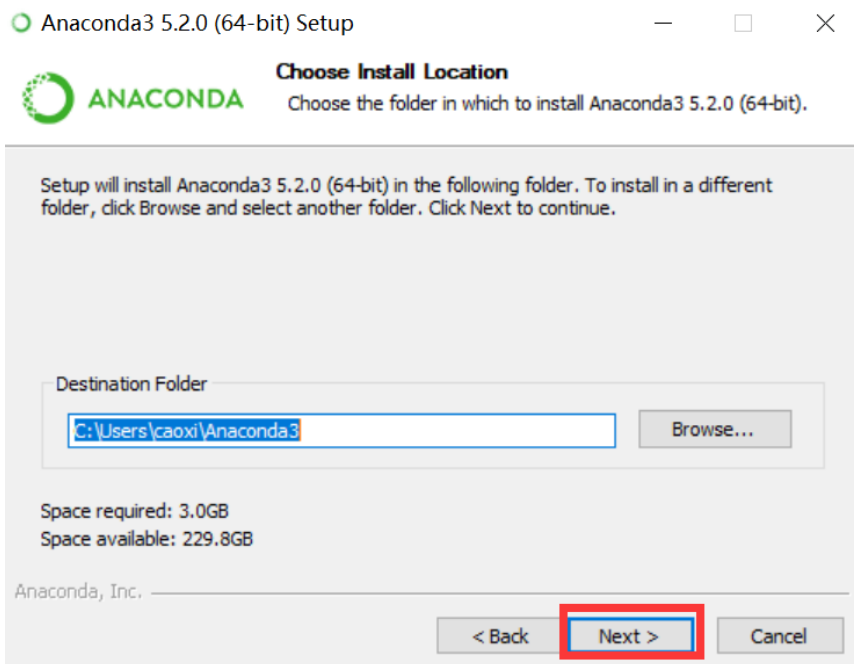
2. Anaconda 安装



安装Anaconda



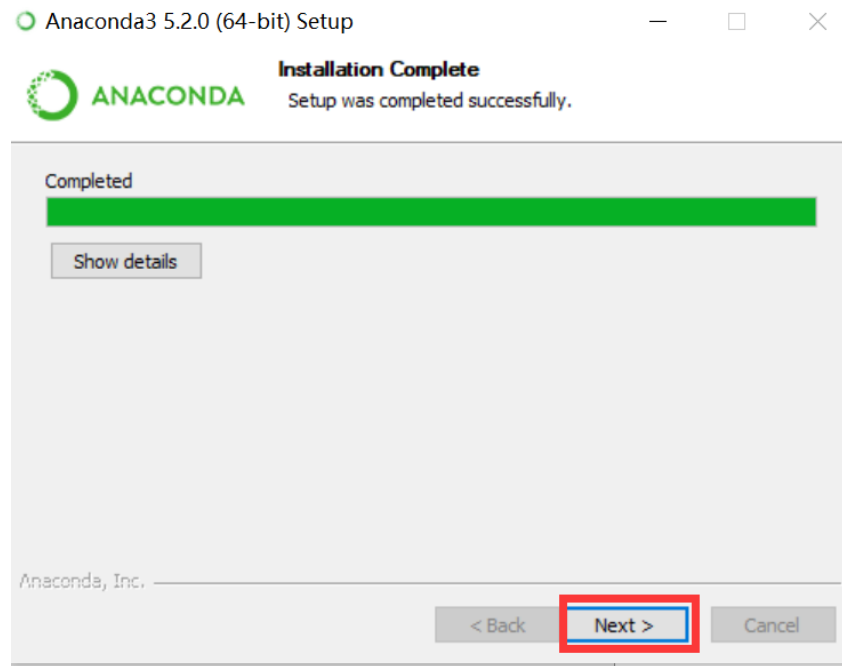
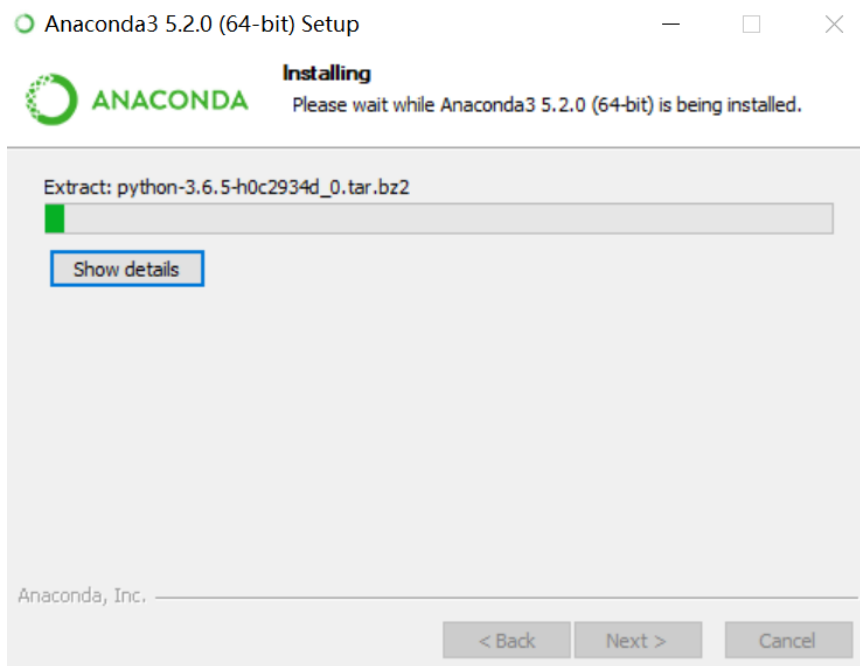
2. Anaconda 安装



安装Anaconda



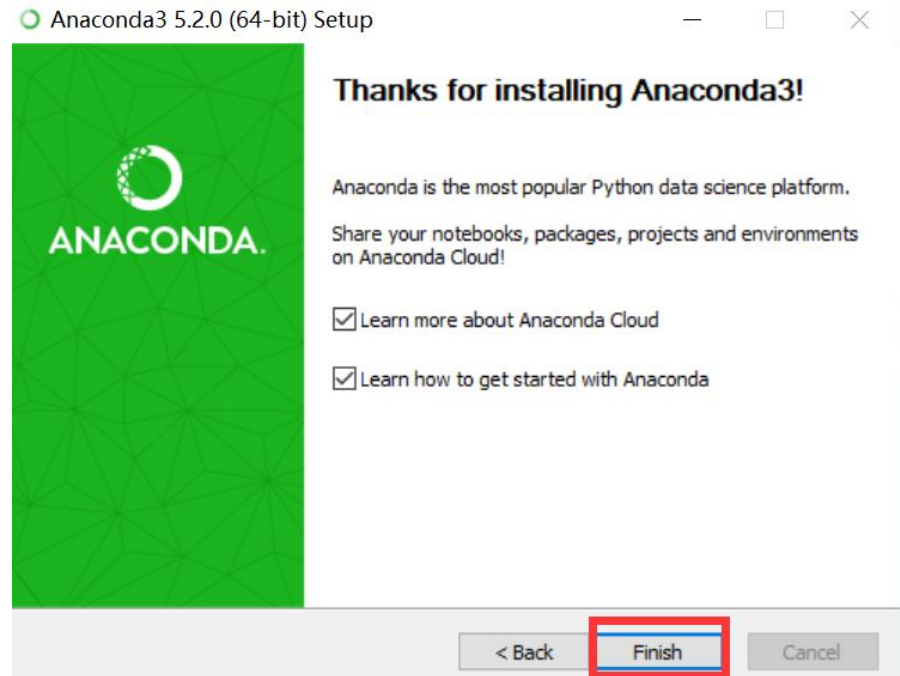
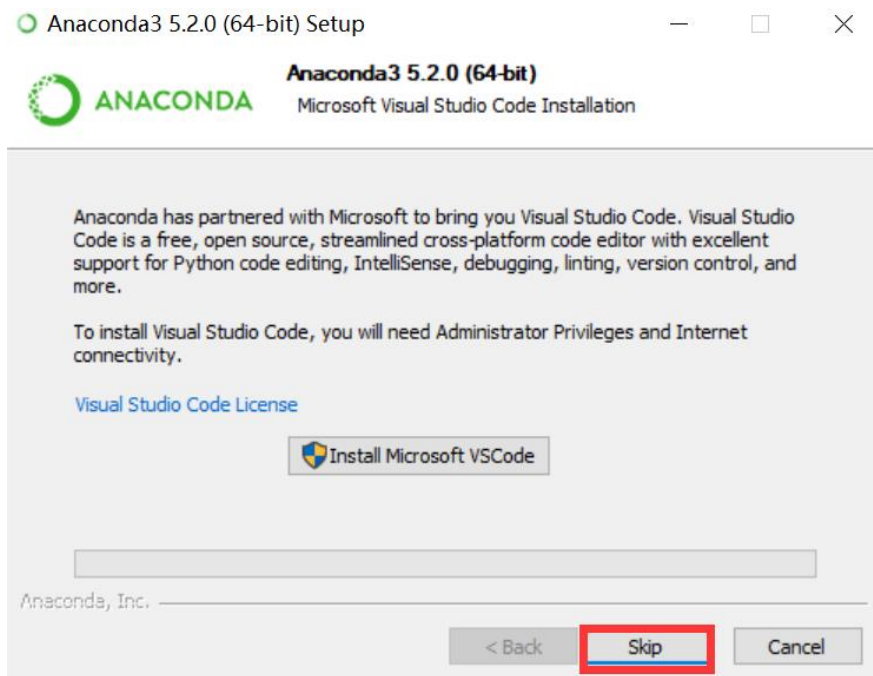
2. Anaconda 安装



安装Anaconda



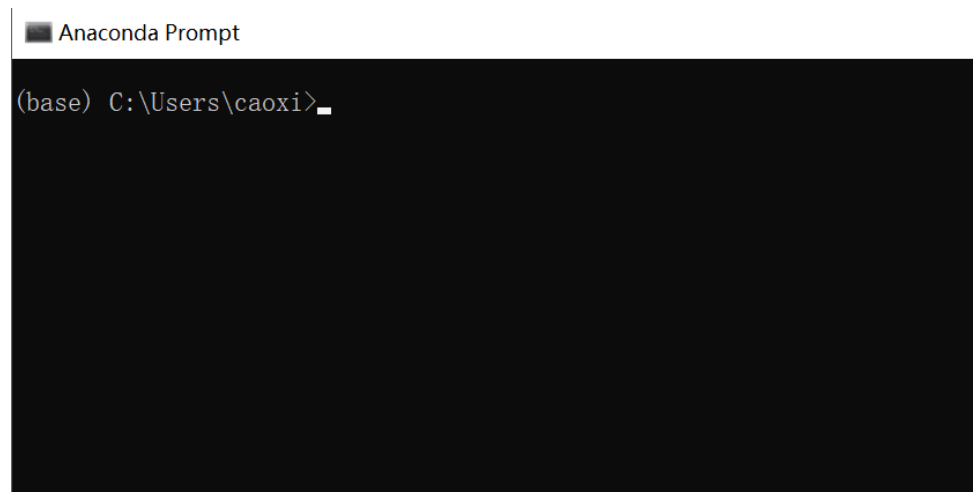
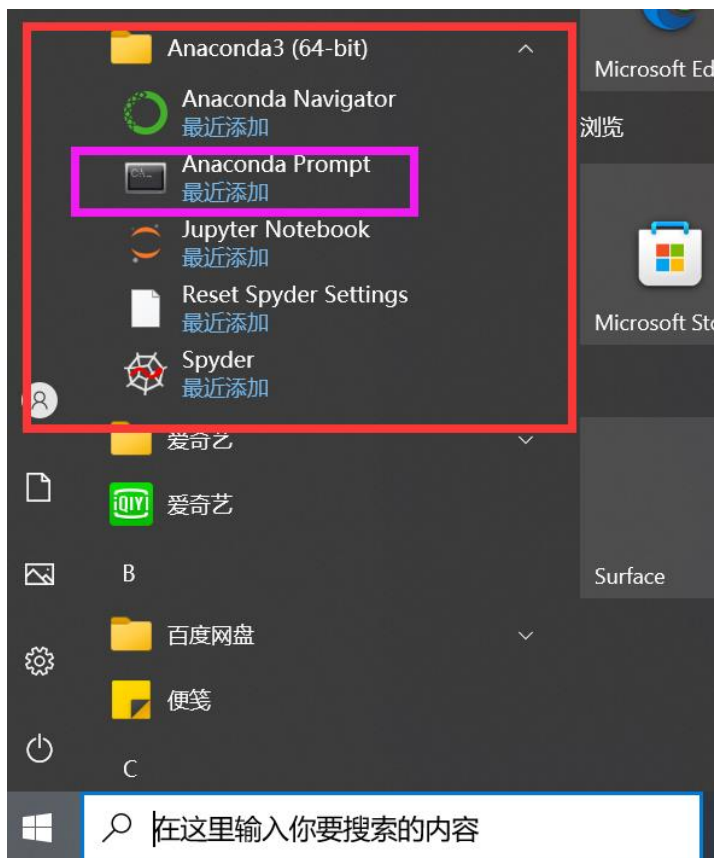
2. Anaconda 安装





启动Anaconda

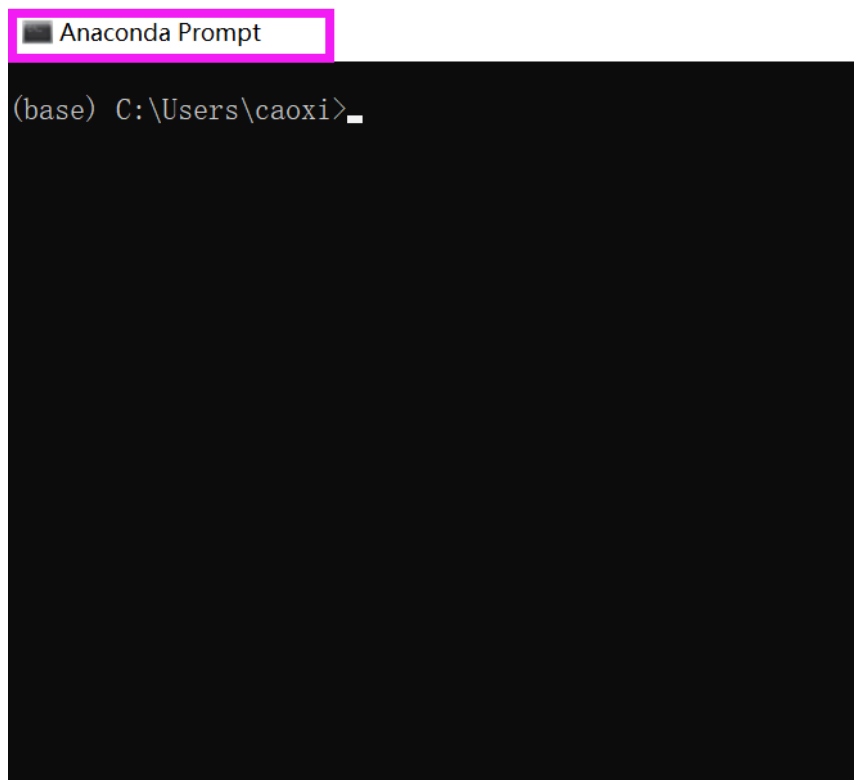
Anaconda Prompt: anaconda版的cmd 命令提示窗





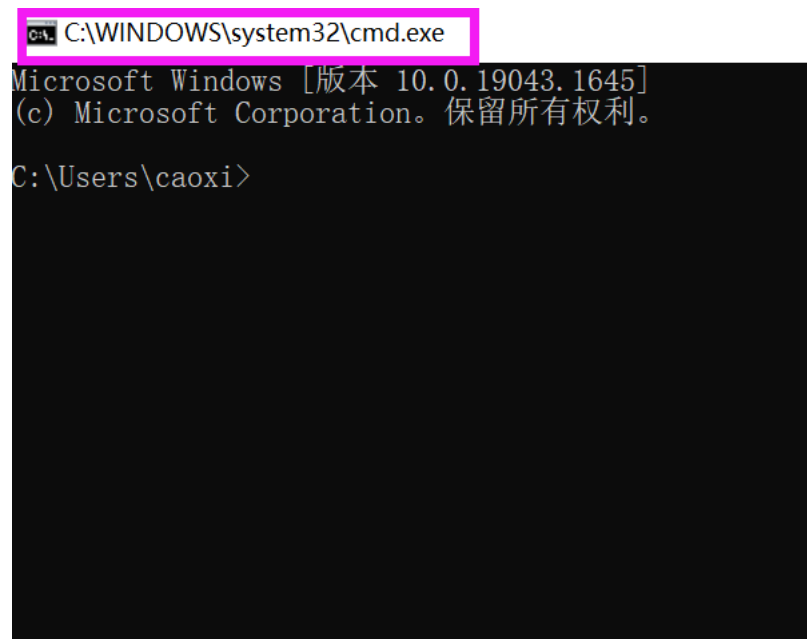
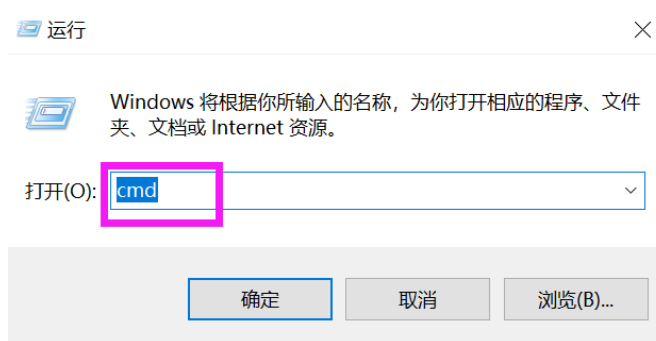
命令提示窗的区别

Anaconda Prompt:
anaconda版的cmd 命令
提示窗



The screenshot shows the Anaconda Prompt terminal window. The title bar is highlighted with a pink box and contains the text "Anaconda Prompt". The terminal content shows the prompt "(base) C:\Users\caoxi>_" on a black background.

Windows系统自带的
cmd 命令提示窗

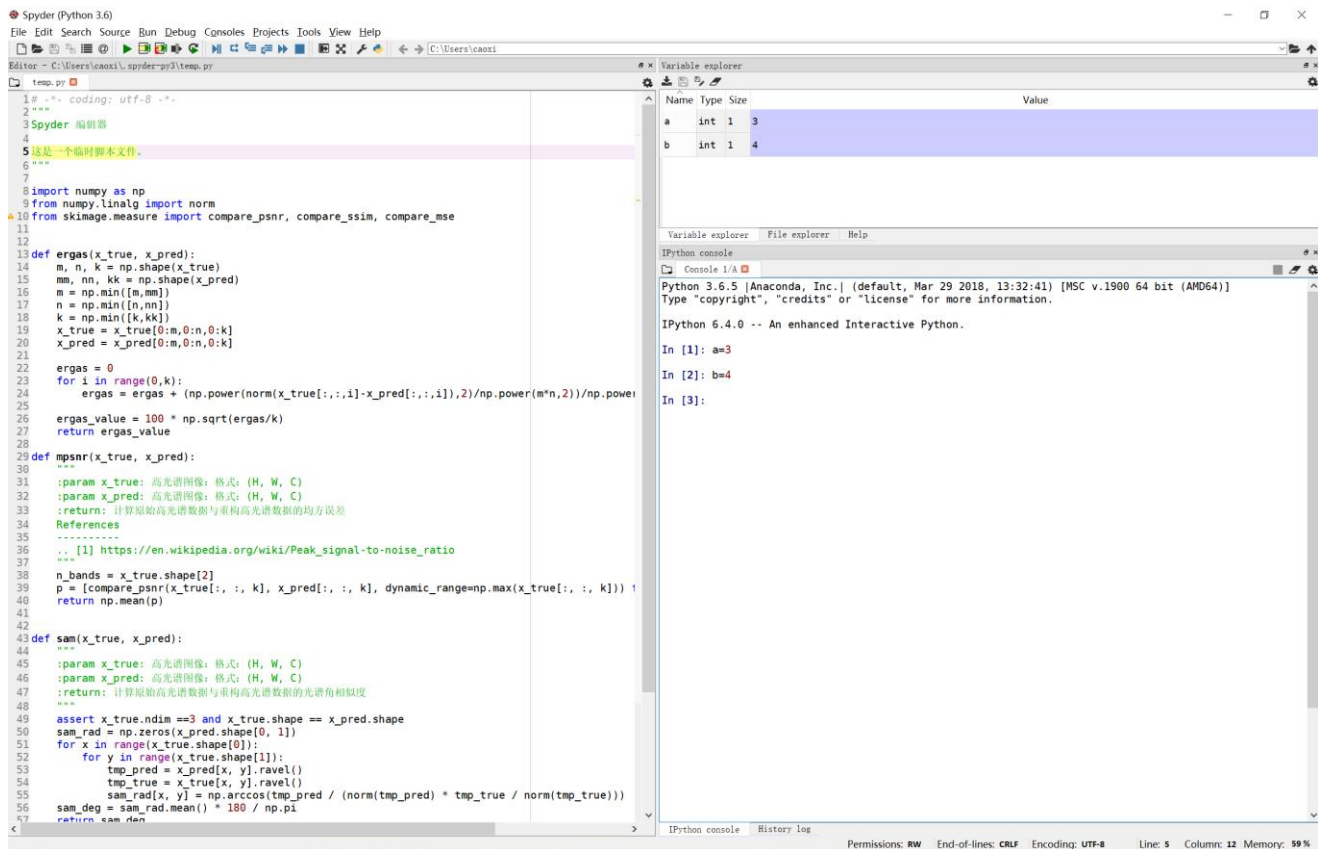
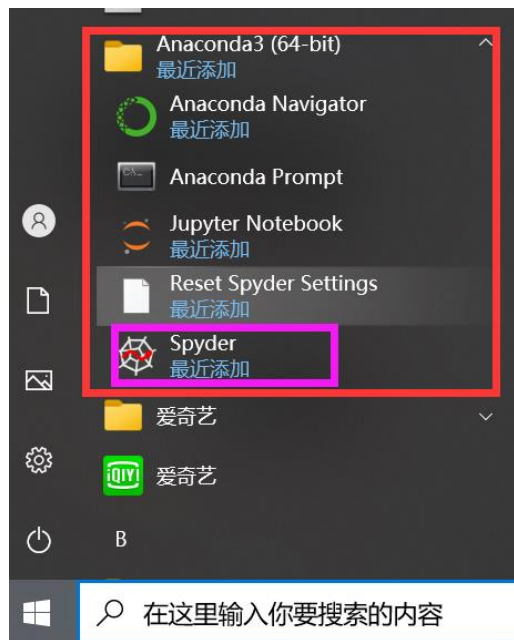


The screenshot shows the Windows Command Prompt terminal window. The title bar is highlighted with a pink box and contains the text "C:\WINDOWS\system32\cmd.exe". The terminal content shows the Microsoft Windows version and copyright information, followed by the prompt "C:\Users\caoxi>" on a black background.

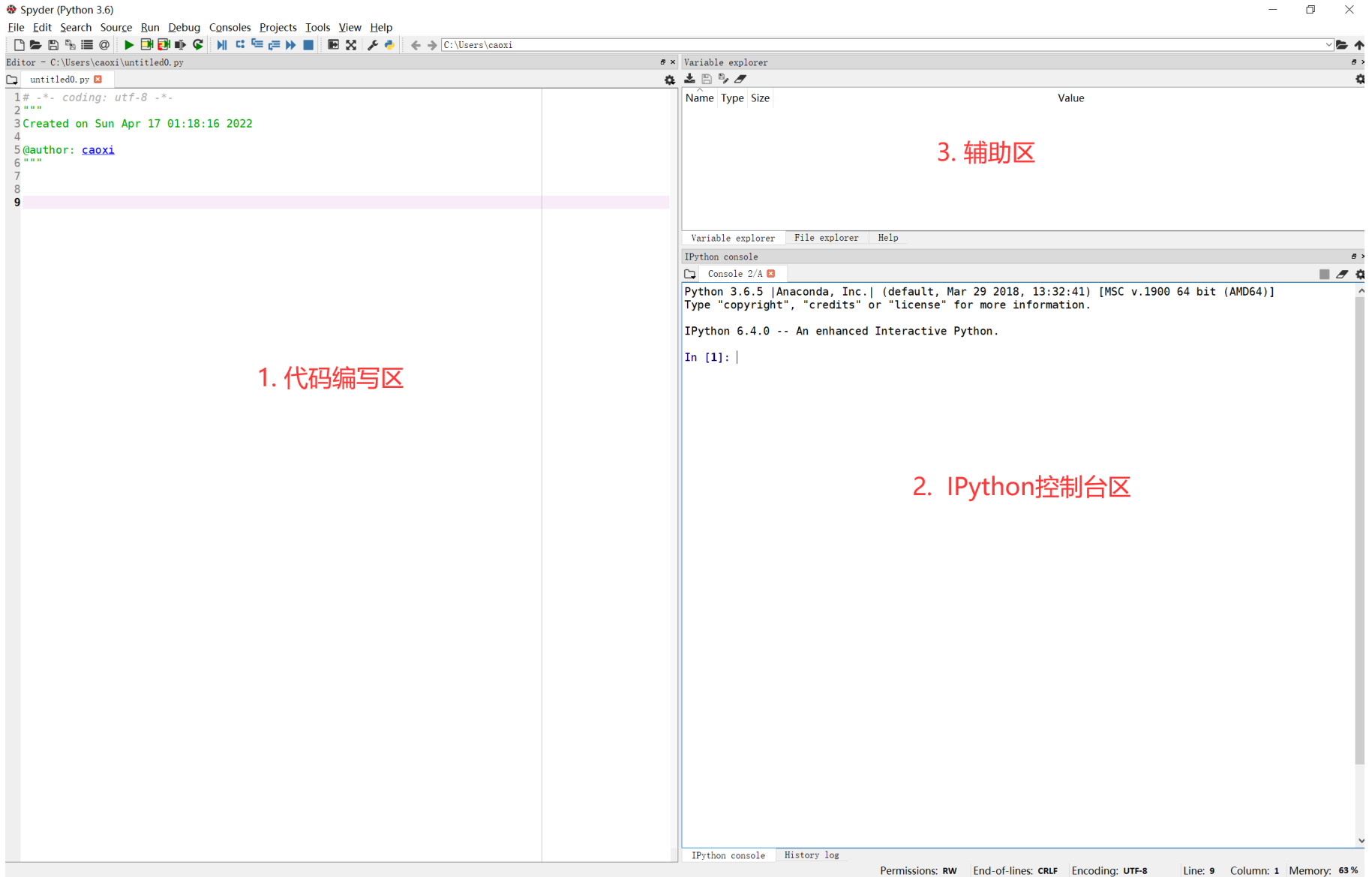


启动Anaconda

Spyder: Python的一个简单的集成开发环境



Spyder





Spyder

初步编程使用: 输入python代码, 编写完毕后按下 F5 键, 将在 IPython控制台显示结果

```
1# -*- coding: utf-8 -*-
2"""
3Created on Sun Apr 17 01:18:16 2022
4
5@author: caoxi
6"""
7
8
9sentence = "Hello World"
10print(sentence)
```

Name	Type	Size
sentence	str	1

Hello World

```
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29
2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more
information.

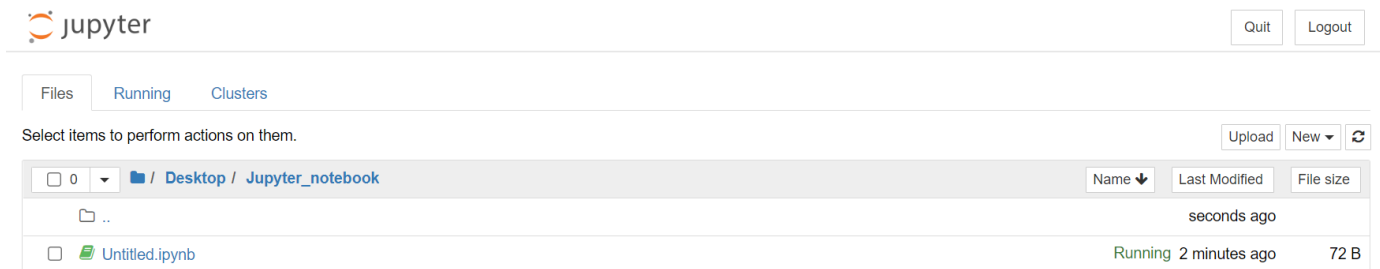
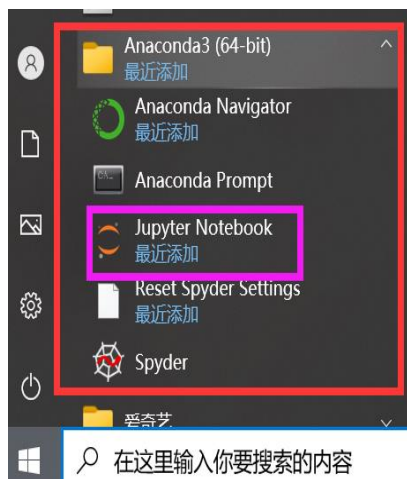
IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/caoxi/Desktop/机器学习编程实践课程/机器学习编程设计-曹相湧孙凯/first.py',
wdir='C:/Users/caoxi/Desktop/机器学习编程实践课程/机器学习编程设计-曹相湧孙凯')
Hello World
```



启动Anaconda

Jupyter Notebook: 基于浏览器的交互编程环境，可用于全过程计算：开发、文档编写、运行代码和展示结果，在数据科学家中广泛使用，用来整合代码、文本和图像非常方便



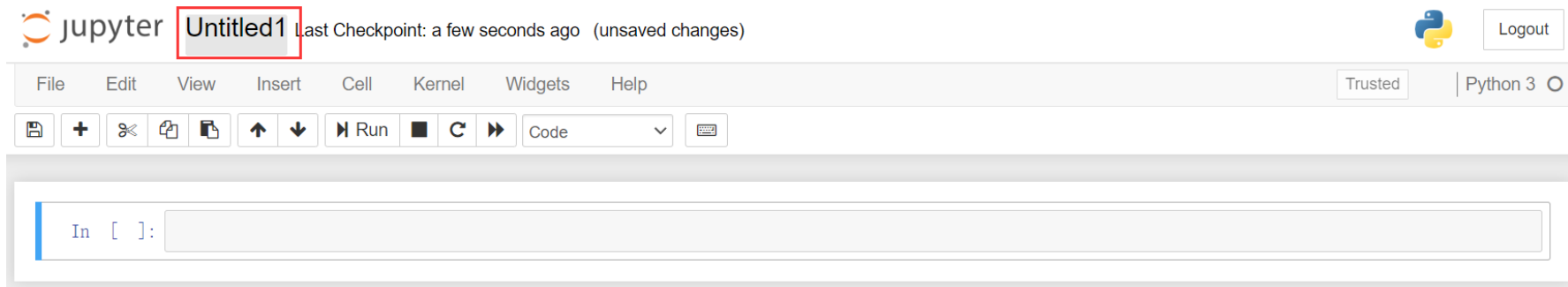
如何使用Jupyter Notebook



第一步（创建文件）： 点击jupyter右上角的new->python3，创建一个基于python3的交互式文档



第二步（修改文件名）： 点击左上角文件名称，显示输入框后可以给文件重命名



如何使用Jupyter Notebook



第三步（往文档里插入并运行代码）：新建的文件里会自动有一个输入代码的cell，可在里面输入并运行自己的python代码

The screenshot displays the Jupyter Notebook interface. At the top, the header shows the Jupyter logo, the text "介绍Jupyter", and "Last Checkpoint: 9 minutes ago (autosaved)". On the right, there is a Python logo and a "Logout" button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Under the "Cell" menu, the "Run" button is highlighted with a red box and a red arrow pointing to it, with the label "2.运行代码" next to it. The main area contains a code cell with the following Python code:

```
In [1]: a = 1
        b = 2
        c = a + b
        print(c)
```

Below the code cell, the output "3" is displayed, with a red arrow pointing to it and the label "3. 代码运行结果".

如何使用Jupyter Notebook



第四步（插入markdown文档）

jupyter 介绍Jupyter Last Checkpoint: 16 minutes ago (unsaved changes)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted



Python 3



1. 增加cell

2. 将cell设置为文本模式(markdown)

```
In [1]: a = 1
        b = 2
        c = a + b
        print(c)
```

3

Jupyter使用教程

****jupyter****可以很方便的将代码和文本结合在一起

3. 编辑文本

jupyter 介绍Jupyter Last Checkpoint: 22 minutes ago (unsaved changes)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted



Python 3



```
In [1]: a = 1
        b = 2
        c = a + b
        print(c)
```

3

Jupyter使用教程

jupyter可以很方便的将代码和文本结合在一起

Markdown语法介绍: <https://blog.csdn.net/afei/article/details/80717153>



Scikit-learn必要的库

Scikit-learn是基于以下科学计算库开发的：

- **numpy**
- **scipy**
- **pandas**
- **matplotlib**



numpy

- **numpy**: Python科学计算基础包之一，它的功能包括多维数组、高级数学函数，以及伪随机数生成器
- 在scikit-learn中， numpy数组是基本数据结构。numpy核心功能是ndarray类，即多维数组，数组元素必须同一类型

In[2]:

```
import numpy as np

x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

Out[2]:

```
x:
[[1 2 3]
 [4 5 6]]
```



scipy

- **scipy**: Python中用于科学计算的函数集合，具有线性代数高级程序、数学函数优化、信号处理、统计分布等多项功能
- scikit-learn利用scipy中的函数集合实现算法，对scikit-learn来说，scipy中最重要的是scipy.sparse：给出稀疏矩阵

In[3]:

```
from scipy import sparse
```

```
# Create a 2D NumPy array with a diagonal of ones, and zeros everywhere else  
eye = np.eye(4)  
print("NumPy array:\n{}".format(eye))
```

Out[3]:

```
NumPy array:  
[[ 1.  0.  0.  0.]  
 [ 0.  1.  0.  0.]  
 [ 0.  0.  1.  0.]  
 [ 0.  0.  0.  1.]]
```



scipy

- **scipy**: Python中用于科学计算的函数集合，具有线性代数高级程序、数学函数优化、信号处理、统计分布等多项功能
- scikit-learn利用scipy中的函数集合实现算法，对scikit-learn来说，scipy中最重要的是scipy.sparse：给出稀疏矩阵

In[4]:

```
# Convert the NumPy array to a SciPy sparse matrix in CSR format  
# Only the nonzero entries are stored  
sparse_matrix = sparse.csr_matrix(eye)  
print("\nSciPy sparse CSR matrix:\n{}".format(sparse_matrix))
```

Out[4]:

```
SciPy sparse CSR matrix:  
(0, 0)    1.0  
(1, 1)    1.0  
(2, 2)    1.0  
(3, 3)    1.0
```



scipy

- **scipy**: Python中用于科学计算的函数集合，具有线性代数高级程序、数学函数优化、信号处理、统计分布等多项功能
- scikit-learn利用scipy中的函数集合实现算法，对scikit-learn来说，scipy中最重要的是scipy.sparse：给出稀疏矩阵

In[5]:

```
data = np.ones(4)
row_indices = np.arange(4)
col_indices = np.arange(4)
eye_coo = sparse.coo_matrix((data, (row_indices, col_indices)))
print("C00 representation:\n{}".format(eye_coo))
```

Out[5]:

C00 representation:

(0, 0)	1.0
(1, 1)	1.0
(2, 2)	1.0
(3, 3)	1.0



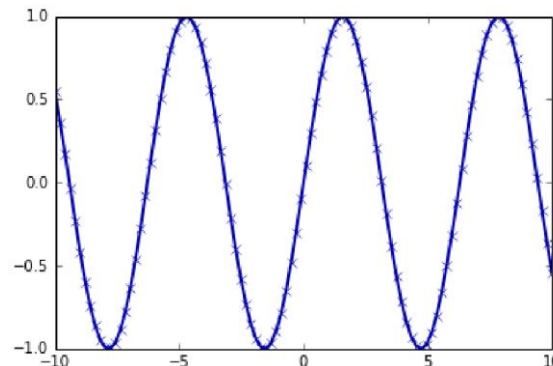
matplotlib

- **matplotlib:** Python主要的科学绘图库，完成绘图和数据可视化
- 在Jupyter Notebook中，可使用`%matplotlib notebook`或`%matplotlib inline`命令，将图像直接显示在浏览器中

In[6]:

```
%matplotlib inline
import matplotlib.pyplot as plt

# Generate a sequence of numbers from -10 to 10 with 100 steps in between
x = np.linspace(-10, 10, 100)
# Create a second array using sine
y = np.sin(x)
# The plot function makes a line chart of one array against another
plt.plot(x, y, marker="x")
```





pandas

- **pandas:** 用于处理和分析数据的Python库，它基于一种叫做 DataFrame 的数据结构，一个pandas DataFrame就是一张表格
- pandas中包含大量用于修改表格和操作表格的方法
- pandas中每列数据类型可互不相同(如整型、字符串、浮点等)
- 可从许多文件格式(Excel文件、CSV文件)和数据库中读取数据

In[7]:

```
import pandas as pd
```

```
# create a simple dataset of people
```

```
data = {'Name': ["John", "Anna", "Peter", "Linda"],  
        'Location': ["New York", "Paris", "Berlin", "London"],  
        'Age' : [24, 13, 53, 33]  
}
```

```
data_pandas = pd.DataFrame(data)
```

```
# IPython.display allows "pretty printing" of dataframes
```

```
# in the Jupyter notebook
```

```
display(data_pandas)
```

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda



pandas

- **pandas:** 用于处理和分析数据的Python库，它基于一种叫做 DataFrame 的数据结构，一个pandas DataFrame就是一张表格
- pandas中包含大量用于修改表格和操作表格的方法
- pandas中每列数据类型可互不相同(如整型、字符串、浮点等)
- 可从许多文件格式(Excel文件、CSV文件)和数据库中读取数据

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

In[8]:

```
# Select all rows that have an age column greater than 30  
display(data_pandas[data_pandas.Age > 30])
```

	Age	Location	Name
2	53	Berlin	Peter
3	33	London	Linda



mglearn

- **mglearn**: 快速美化绘图，或获取一些有趣数据
- 本课程内容频繁使用numpy、matplotlib和pandas等库，所有代码都默认导入了以下这些库：

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import mglearn
```

- 在Jupyter notebook中运行代码，使用%matplotlib notebook或%matplotlib inline命令显示图像
- 如果没使用Jupyter notebook或以上显示命令，调用**plt.show**来显示图像

Python 2与Python 3的对比



- Python两大版本广为使用: Python 2 (Python 2.7)和Python 3 (Python 3.5、3.6、3.7、3.8、3.9)
- Python 2已经停止开发, Python 2代码通常无法在Python 3中运行, Python 3代码可在Python 2中运行
- 建议大家使用Python 3
- 本课程提供的代码在Python 2和Python 3中都能运行

本书用到的各个库的版本



In[9]:

```
import sys
print("Python version: {}".format(sys.version))

import pandas as pd
print("pandas version: {}".format(pd.__version__))

import matplotlib
print("matplotlib version: {}".format(matplotlib.__version__))

import numpy as np
print("NumPy version: {}".format(np.__version__))

import scipy as sp
print("SciPy version: {}".format(sp.__version__))

import IPython
print("IPython version: {}".format(IPython.__version__))

import sklearn
print("scikit-learn version: {}".format(sklearn.__version__))
```

Out[9]:

```
Python version: 3.5.2 |Anaconda 4.1.1 (64-bit)| (default, Jul  2 2016, 17:53:06)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]
pandas version: 0.18.1
matplotlib version: 1.5.1
NumPy version: 1.11.1
SciPy version: 0.17.1
IPython version: 5.1.0
scikit-learn version: 0.18
```

使用本书代码，scikit-learn的版本不应低于这个版本



第一个应用：鸢尾花分类

- **问题描述**: 一名植物学爱好者对鸢尾花品种感兴趣，她搜集了一些鸢尾花测量数据，每朵鸢尾花测量数据包括：**花瓣的长度和宽度、花萼的长度和宽度** (单位:cm)，这些花之前已被植物学家鉴定为属于：**setosa、versicolor或virginica**三个品种之一。因此，对于这些测量数据，她可以确定每朵鸢尾花所属的品种，假设这位植物学爱好者在野外只会遇到这三种鸢尾花。
- **目标**: 构建一个机器学习模型，从已知品种的鸢尾花测量数据中进行学习，并对新鸢尾花数据进行品种预测

$$\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$$

\mathbf{x}_i : 鸢尾花测量数据（数据、样本）

y_i : 鸢尾花的品种（标签）

n : 鸢尾花的个数（样本总数）

监督学习问题、分类问题



鸢尾花分类-初识数据



In[10]: Bunch对象, 包含key和value

```
from sklearn.datasets import load_iris  
iris_dataset = load_iris()
```

In[11]:

```
print("Keys of iris_dataset: \n{}".format(iris_dataset.keys()))
```

Out[11]:

```
Keys of iris_dataset:  
dict_keys(['target_names', 'feature_names', 'DESCR', 'data', 'target'])
```

In[12]:

```
print(iris_dataset['DESCR'][:193] + "\n...")
```

Out[12]:

```
Iris Plants Database  
=====
```

```
Notes  
----
```

```
Data Set Characteristics:
```

```
:Number of Instances: 150 (50 in each of three classes)  
:Number of Attributes: 4 numeric, predictive att
```

```
...  
----
```

鸢尾花分类-初识数据



In[13]:

```
print("Target names: {}".format(iris_dataset['target_names']))
```

Out[13]:

```
Target names: ['setosa' 'versicolor' 'virginica']
```

In[14]:

```
print("Feature names: \n{}".format(iris_dataset['feature_names']))
```

Out[14]:

```
Feature names:
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
 'petal width (cm)']
```

In[15]:

```
print("Type of data: {}".format(type(iris_dataset['data'])))
```

Out[15]:

```
Type of data: <class 'numpy.ndarray'>
```

In[16]:

```
print("Shape of data: {}".format(iris_dataset['data'].shape))
```

Out[16]:

```
Shape of data: (150, 4)
```

鸢尾花分类-初识数据



In[17]:

```
print("First five columns of data:\n{}".format(iris_dataset['data'][:5]))
```

Out[17]:

```
First five columns of data:  
[[ 5.1  3.5  1.4  0.2]  
 [ 4.9  3.   1.4  0.2]  
 [ 4.7  3.2  1.3  0.2]  
 [ 4.6  3.1  1.5  0.2]  
 [ 5.   3.6  1.4  0.2]]
```

In[18]:

```
print("Type of target: {}".format(type(iris_dataset['target'])))
```

Out[18]:

```
Type of target: <class 'numpy.ndarray'>
```

In[19]:

```
print("Shape of target: {}".format(iris_dataset['target'].shape))
```

Out[19]:

```
Shape of target: (150,)
```



```
print("Target:\n{}".format(iris_dataset['target']))
```

Out[20]:[illegible]

0: setosa 1: versicolor 2: virginica



衡量模型是否成功：训练数据和测试数据

- 模型应用于新测试数据之前，需要判断模型是否有效
- 不能用训练模型的数据进行评估测试，模型会记住训练集，对训练集中的数据总会给出正确标签，导致无法判断模型的泛化能力（即新数据上能否正确预测）
- 采用新数据来评估模型，通常做法：将收集好的带标签数据集分成两部分，一部分数据用于构建机器学习模型，叫做训练数据(training data)或训练集(training set); 剩余的数据用于评估模型性能，叫做测试数据(test data)或测试集(test set)
- Scikit-learn中的train_test_split函数可打乱数据集并进行数据划分，默认是75%训练，剩下25%测试，比例也可调

训练数据和测试数据构建



In[21]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], random_state=0)
```

In[22]:

```
print("X_train shape: {}".format(X_train.shape))
print("y_train shape: {}".format(y_train.shape))
```

Out[22]:

```
X_train shape: (112, 4)
y_train shape: (112,)
```

In[23]:

```
print("X_test shape: {}".format(X_test.shape))
print("y_test shape: {}".format(y_test.shape))
```

Out[23]:

```
X_test shape: (38, 4)
y_test shape: (38,)
```



观察数据

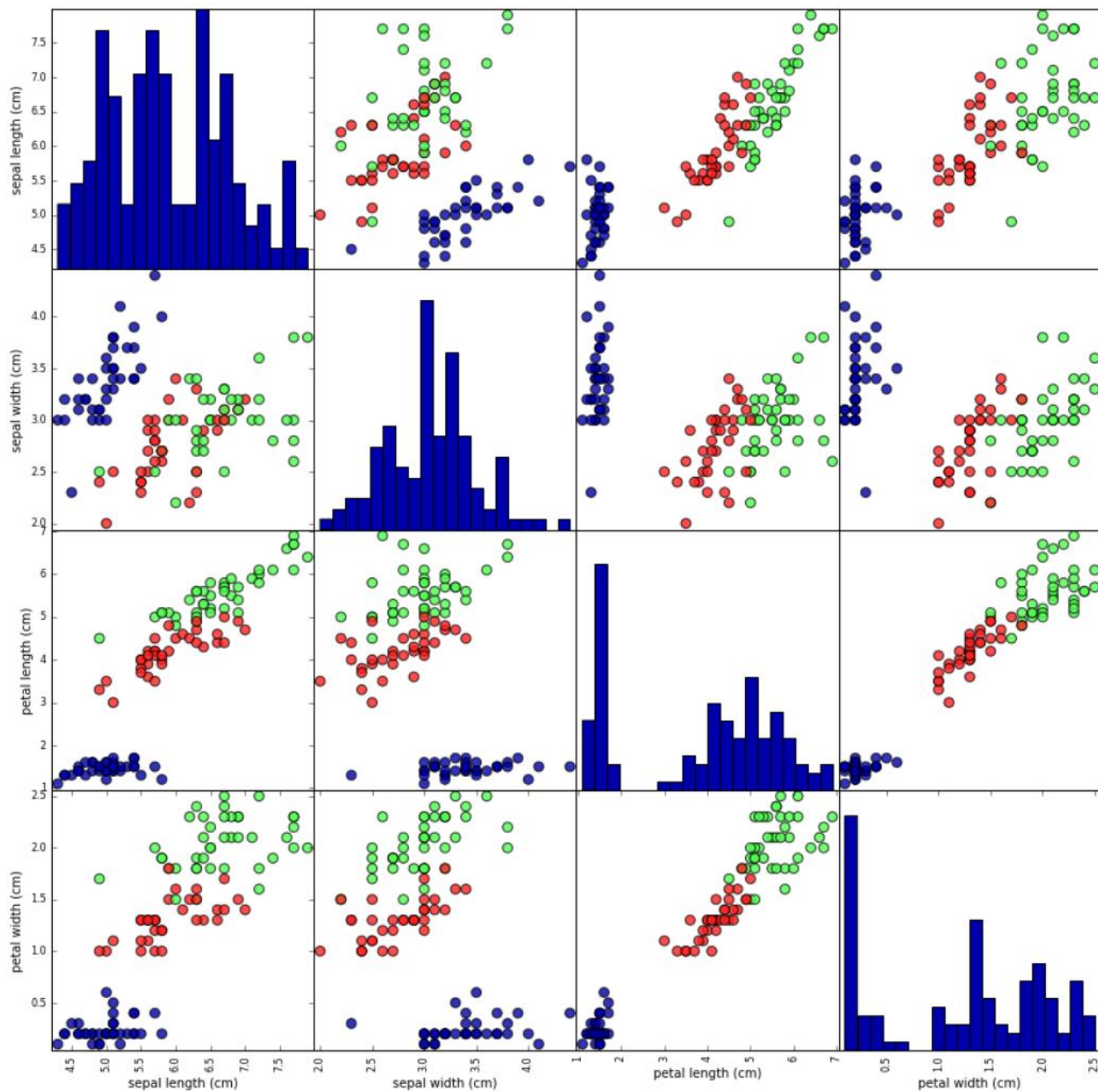
- 构建模型之前，观察一下数据，判断机器学习是否能完成
- 观察数据也能发现异常值和特殊值
- 观察数据的一种方法是数据可视化：不超过3个特征的数据，可采用绘制散点图的方法；特征维数大于3的情况，可采用绘制散点图矩阵的方法
- 散点图矩阵：查看两两特征间的散点图，比如特征维数是4，散点图矩阵就是一个4x4，它的局限性是：没法显示所有特征之间的关系

观察数据



In[24]:

```
# create dataframe from data in X_train  
# label the columns using the strings in iris_dataset.feature_names  
iris_dataframe = pd.DataFrame(X_train, columns=iris_dataset.feature_names)  
# create a scatter matrix from the dataframe, color by y_train  
grr = pd.scatter_matrix(iris_dataframe, c=y_train, figsize=(15, 15), marker='o',  
                        hist_kwds={'bins': 20}, s=60, alpha=.8, cmap=mglearn.cm3)
```





构建第一个模型：k近邻算法

- scikit-learn中有许多可用的分类算法
- k近邻算法: 对新数据点做预测时，算法会在训练集中寻找与这个新数据点距离最近的k个数据点，然后将这k个数据点中数量最多的类别标签赋给这个新数据点
- scikit-learn中所有机器学习模型都在各自的类中实现，k近邻分类算法在neighbors模块的KNeighborsClassifier类中实现，需要将这个类实例化一个对象，然后才能使用这个模型

构建第一个模型：k近邻算法



In[25]:

```
from sklearn.neighbors import KNeighborsClassifier  
knn = KNeighborsClassifier(n_neighbors=1)
```

In[26]:

```
knn.fit(X_train, y_train)
```

Out[26]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',  
                     metric_params=None, n_jobs=1, n_neighbors=1, p=2,  
                     weights='uniform')
```



评估模型

- 用测试集评估模型，通过计算精度(accuracy)衡量模型优劣
- 精度 = 品种预测正确的花/测试集中花的个数

In[29]:

```
y_pred = knn.predict(X_test)
print("Test set predictions:\n {}".format(y_pred))
```

Out[29]:

```
Test set predictions:
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0 2]
```

In[30]:

```
print("Test set score: {:.2f}".format(np.mean(y_pred == y_test)))
```

Out[30]:

```
Test set score: 0.97
```

In[31]:

```
print("Test set score: {:.2f}".format(knn.score(X_test, y_test)))
```

Out[31]:

```
Test set score: 0.97
```

模型在测试集的精度为**97%**，**模型足够可信**



做出预测

- 对新数据样本，利用评估后的高精度模型做出预测

In[27]:

```
X_new = np.array([[5, 2.9, 1, 0.2]])  
print("X_new.shape: {}".format(X_new.shape))
```

Out[27]:

```
X_new.shape: (1, 4)
```

In[28]:

```
prediction = knn.predict(X_new)  
print("Prediction: {}".format(prediction))  
print("Predicted target name: {}".format(  
    iris_dataset['target_names'][prediction]))
```

Out[28]:

```
Prediction: [0]  
Predicted target name: ['setosa']
```

本节课学到了什么



- 了解机器学习的概念、简要介绍了监督学习、半监督学习、强化学习等范式，以及机器学习的基本工作流程
- 简要介绍Python, scikit-learn机器学习库，如何安装scikit-learn库，通过安装anaconda来实现，介绍anaconda中的编译器spyder, 基于浏览器的交互编程环境Jupyter notebook, Markdown的基本用法，介绍了scikit-learn依赖的一些常用库，包括numpy, scipy, matplotlib, pandas等
- 利用scikit-learn构建了一个简单的机器学习应用-鸢尾花分类，通过这个应用了解了scikit-learn中任何机器学习算法的核心代码，如train_test_split、fit、predict、score等方法时scikit-learn监督学习模型中的最常用接口
- 下一章：深入介绍scikit-learn中各种类型的监督学习模型及其正确使用方法

未完待续！