

1.机器学习概述

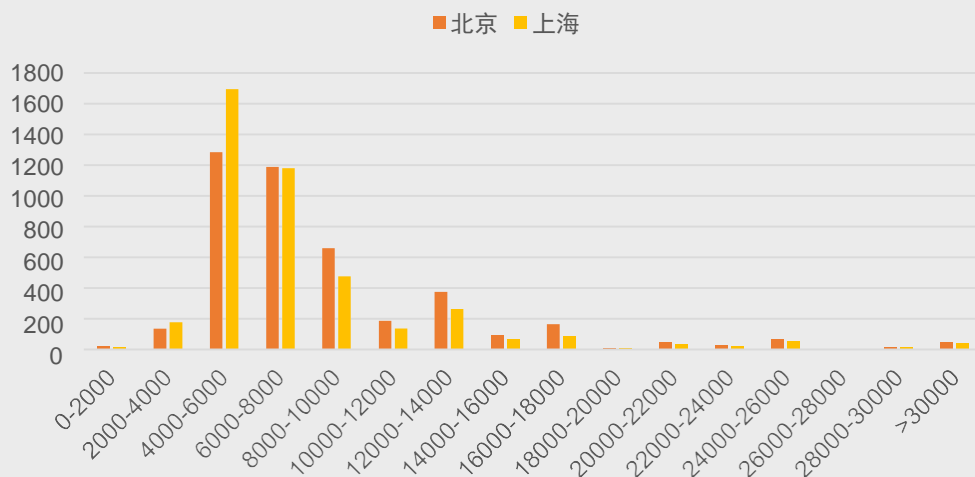
课程目录

Course catalogue

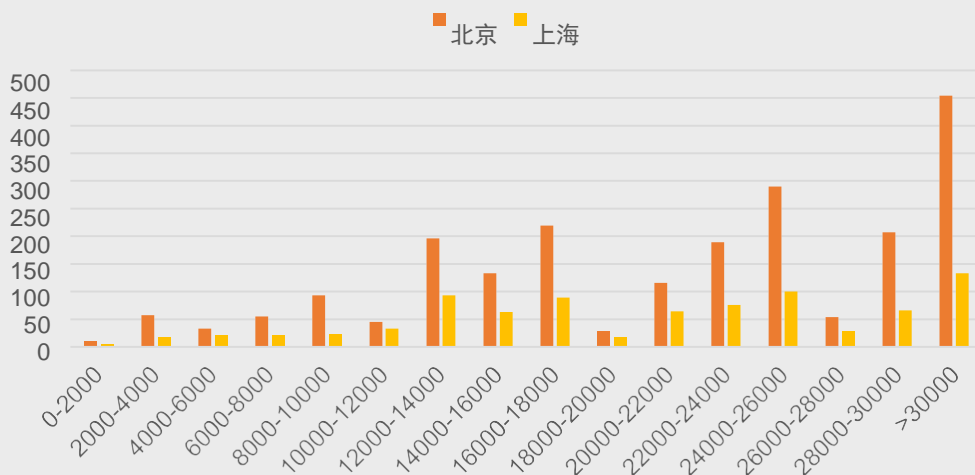
- 1/ 机器学习简介
- 2/ 机器学习、人工智能和数据挖掘
- 3/ 常用机器学习库
- 4/ Jupyter简介

机器学习概述

某招聘网站会计工资分布频数图



某招聘网站机器学习工资分布频数图



机器学习是什么

【问题】机器学习很高大上么？

图中数据采集自某招聘网站七月底的招聘数据。从工资上看，机器学习确实很高大上。那么机器学习是什么？想一想，能不能举个机器学习的例子？

生活中到处都是机器学习的应用，考虑以下场景：

买西瓜，怎么才能买到甜的？

平时努力学习的同学，期末成绩会怎么样？

台风“利奇马”要来，明天还去不去郊游？

通讯公司推荐我更改资费，他是怎么发现我的？

日常生活中的机器学习



什么是机器学习

机器学习是通过编程让计算机从数据中进行学习的科学（和艺术）。

➤ 广义的概念：

机器学习是让计算机具有学习的能力，无需进行明确编程。

—— 亚瑟·萨缪尔，1959

Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

什么是机器学习

◆机器学习的形式化的描述

对于某类任务 T 和性能度量 P ，如果一个计算机程序在 T 上以 P 衡量的性能随着经验 E 而自我完善，那么就称这个计算机程序在从经验 E 学习。

例如，你的垃圾邮件过滤器就是一个机器学习程序，它可以根据垃圾邮件（比如，用户标记的垃圾邮件）和普通邮件（非垃圾邮件，也称作 ham）学习标记垃圾邮件。用来进行学习的样例称作训练集。每个训练样例称作训练实例（或样本）。在这个例子中，**任务 T 就是标记新邮件是否是垃圾邮件，经验 E 是训练数据，性能 P 需要定义**：例如，可以使用正确分类的比例。这个性能指标称为准确率，通常用在分类任务中。

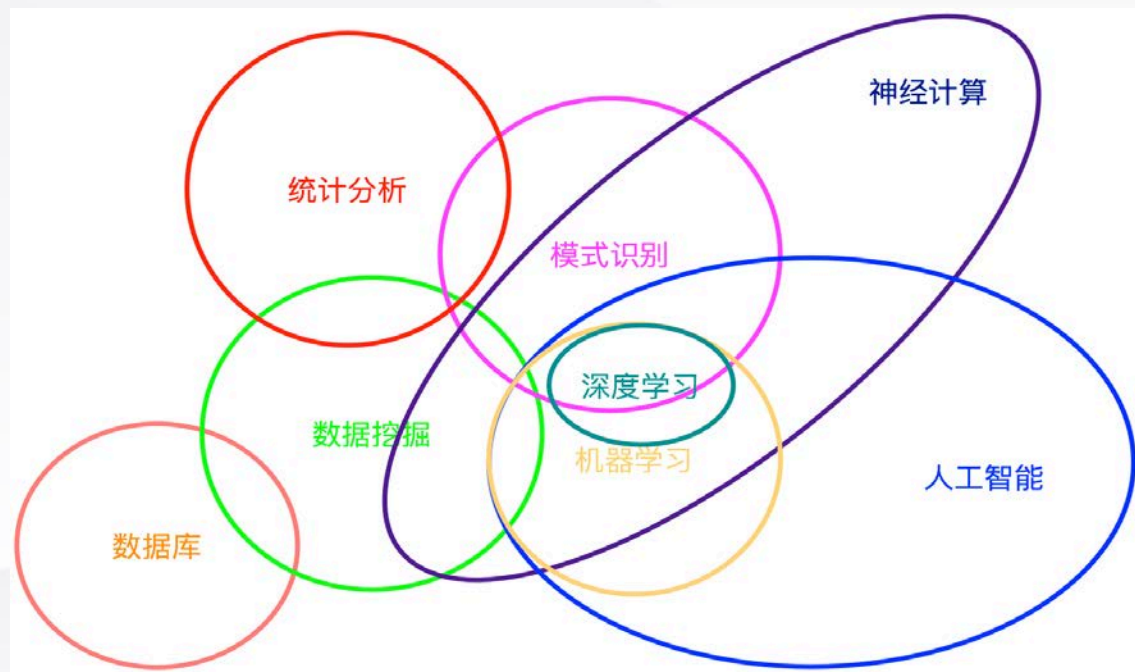
机器学习定义

机器学习简史

机器学习阶段	年份	主要成果	代表人物
人工智能起源	1936	自动机模型理论	Alan Turing
	1943	MP模型	Warren McCulloch、Walter Pitts
	1951	符号演算	John von Neumann
	1950	逻辑主义	Claude Shannon
	1956	人工智能	John McCarthy、Marvin Minsky、Claude Shannon
人工智能初期	1958	LISP	John McCarthy
	1962	感知器收敛理论	Frank Roseblatt
	1972	通用问题求解(GPS)	Allen Newell、Herbert Simon
	1975	框架知识表示	Marvin Minsky
进化计算	1965	进化策略	Ingo Rechenberg
	1975	遗传算法	John Henry Holland
	1992	基因计算	John Koza
专家系统和知识工程	1965	模糊逻辑、模糊集	Lotfi Zadeh
	1969	DENDRA、MYCIN	Feigenbaum、Buchanan、Lederberg
	1979	ROSPECTOR	Duda
神经网络	1982	Hopfield网络	Hopfield
	1982	自组织网络	Kohonen
	1986	BP算法	Rumelhart、McClelland
	1989	卷积神经网络	LeCun
	1998	LeNet	LeCun
	1997	循环神经网络RNN	Sepp Hochreiter、Jurgen Schmidhuber
分类算法	1986	决策树ID3算法	J. Ross Quinlan
	1988	Boosting算法	Freund、Michael Kearns
	1993	C4.5算法	J. Ross Quinlan
	1995	AdaBoost算法	Yoav Freund、Robert Schapire
	1995	支持向量机	Corinna Cortes、Vapnik
	2001	随机森林	Leo Breiman、Adele Cutler
深度学习	2006	深层神经网络训练方法	Geoffrey Hinton
	2012	谷歌大脑	Andrew Ng
	2014	生成对抗网络GAN	Ian Goodfellow

机器学习、人工智能和数据挖掘

- 数据科学的目标是理解事物
- 机器学习主要任务是用于预测事物
- 人工智能是生成行动



x_1

什么是人工智能

- 人工智能是要让机器的行为看起来像人所表现出的智能行为一样
- 人工智能包括计算智能、感知智能和认知智能等层次，目前人工智能还介于前两者之间
- 目前人工智能所处的阶段还在“弱人工智能”（Narrow AI）阶段，距离“强人工智能”（General AI）还有较长的路要走
- 人工智能的典型系统包括以下几个方面
 - 博弈游戏算法（如深蓝、Alpha Go、Alpha Zero等）
 - 机器人相关控制理论（运动规划、控制机器人行走等）
 - 优化（谷歌地图选择路线）
 - 自然语言处理（自动程序）
 - 强化学习

机器学习、人工智能与数据挖掘

- 机器学习是人工智能的一个分支，它是实现人工智能的一个核心技术，即以机器学习为手段解决人工智能中的问题。机器学习是通过一些让计算机可以自动“学习”的算法并从数据中分析获得规律，然后利用规律对新样本进行预测
- 机器学习和人工智能有很多交集，其中深度学习就是横跨机器学习和人工智能的一个典型例子。深度学习的典型应用是选择数据训练模型，然后用模型做出预测
- 数据挖掘是从大量的业务数据中挖掘隐藏、有用的、正确的知识促进决策的执行。数据挖掘的很多算法都来自于机器学习，并在实际应用中进行优化。机器学习最近几年也逐渐跳出实验室，解决从实际的数据中学习模式，解决实际问题。数据挖掘和机器学习的交集越来越大

典型机器学习应用领域

- 机器学习能够显著提高企业的智能水平，增强企业的竞争力，人工智能对于各行业的影响越来越大，机器学习应用的典型领域有网络安全、搜索引擎、产品推荐、自动驾驶、图像识别、识音识别、量化投资、自然语言处理等。随着海量数据的累积和硬件运算能力的不断提升，机器学习的应用领域还在快速地延展。

艺术创作

- 图像识别
- 照片分类
- 图像变形
- 图片生成
- 图片美化
- 图片修复
- 图片场景描述

艺术创作

- Neural Doodle项目
- 应用深度神经网络将图片生成艺术画



金融领域

- 信用评级
- 欺诈检测
- 股票市场预测
- 客户关系管理

医疗领域

- 预测患者的诊断结果
- 制订最佳疗程
- 评估风险等级
- 病理分析
- 个性化医疗
- 建立预测模型

自然语言处理

- 分词
- 词性标志
- 句法分析
- 自然语言生成
- 文本分类
- 信息检索
- 信息抽取
- 文字校对
- 问答系统
- 机器翻译
- 自动摘要

网络安全

- 反垃圾邮件
- 反网络钓鱼
- 上网内容过滤
- 反诈骗
- 防范攻击
- 活动监视
- 密码破解
- 无边界攻击模型 & 限制边界攻击模型

工业领域

- 质量管理
- 灾害预测
- 缺陷管理
- 工业分拣
- 故障感知
- 应用存在瓶颈
 - 数据质量
 - 工程师经验
 - 计算能力
 - 机器学习的不可解释性

娱乐行业

- 预测票房
- 视频识别
- 广告计划管理器

Machine Learning



what society thinks I
do



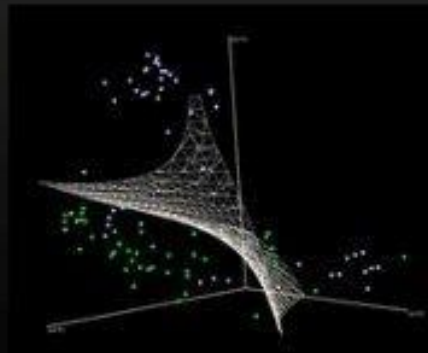
what my friends think
I do



what my parents think
I do

$$\begin{aligned} L_p &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^n \alpha_i \\ \alpha_i &\geq 0, \forall i \\ \mathbf{w} &= \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \sum_{i=1}^n \alpha_i y_i = 0 \\ \nabla \hat{g}(\theta_t) &= \frac{1}{n} \sum_{i=1}^n \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t) \\ \theta_{t+1} &= \theta_t - \eta_t \nabla \ell(x_{(t)}, y_{(t)}; \theta_t) - \eta_t \cdot \nabla r(\theta_t) \\ \mathbb{E}_{(t)}[\ell(x_{(t)}, y_{(t)}; \theta_t)] &= \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i; \theta_t) \end{aligned}$$

what other programmers
think I do



what I think I do

```
>>> from sklearn import svm
```

what I really do

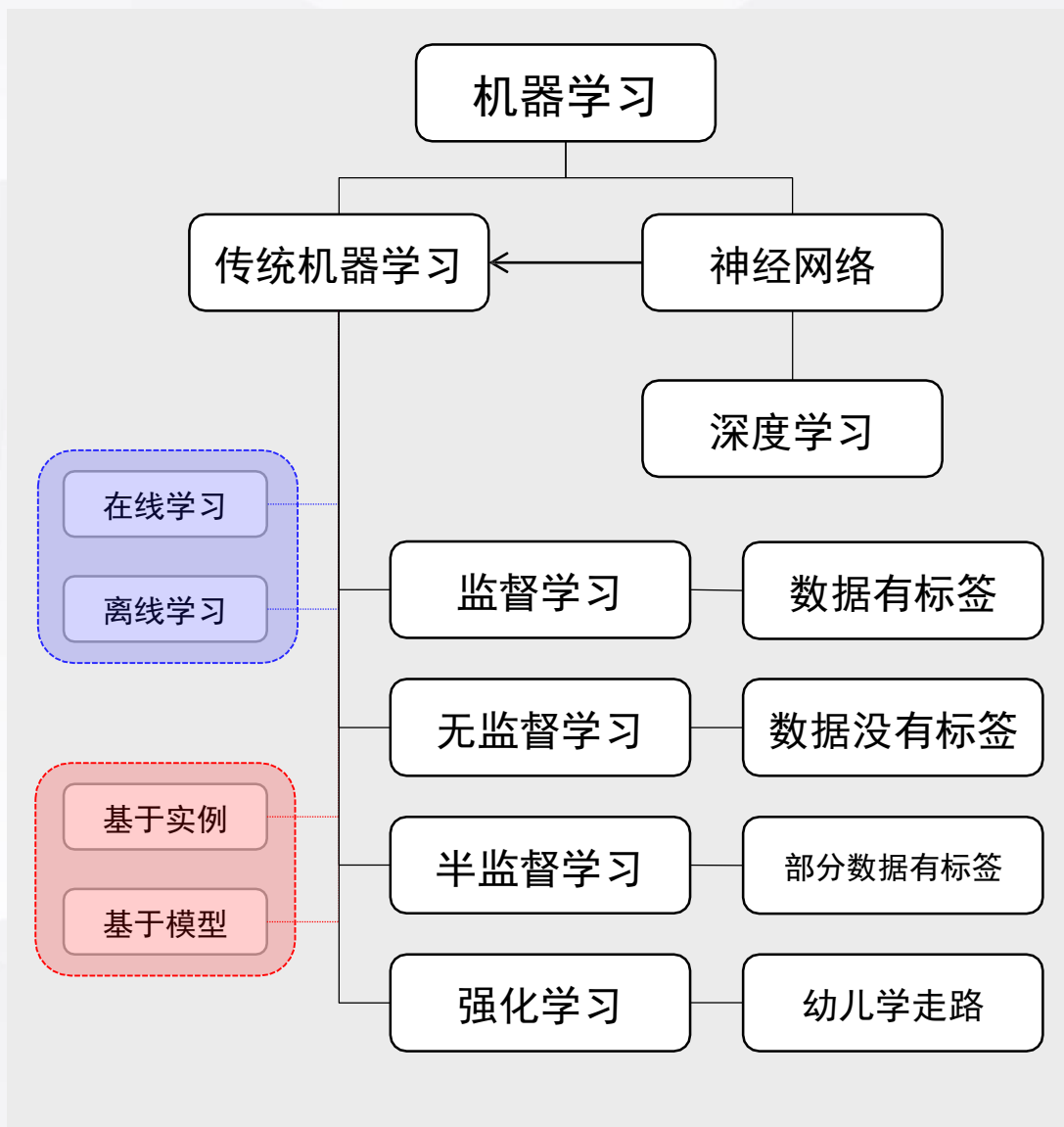
机器学习概述

机器学习框架

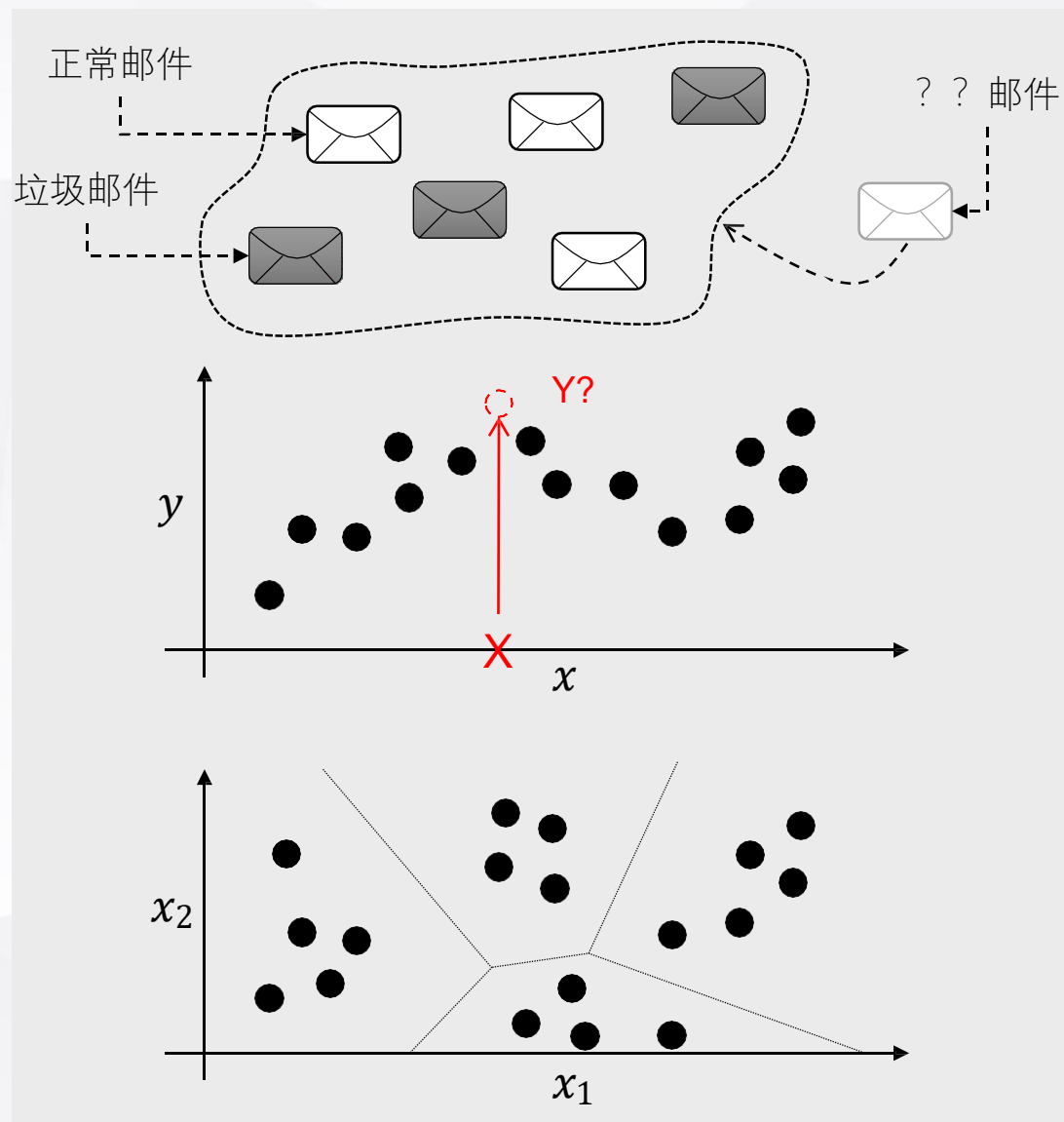
广义上的机器学习，包括传统机器学习和神经网络两部分。

传统机器学习划分方式很多，常用划分包括是否需要人类监督，能否在运行过程中增量学习，以及是否检测到某种预测模式。

按照是否需要人类监督，传统机器学习可以划分为监督学习、无监督学习、半监督学习、强化学习



机器学习概述



监督学习与无监督学习

监督学习分为分类和回归，区别就在于标签是连续变量还是离散变量。

例如根据用户标注，我们可以区分出哪些是垃圾邮件，哪些是正常邮件。分类问题要解决的是，根据历史数据，判断新接收的邮件是不是垃圾邮件？显然“是/否”是一个离散变量。

再比如，连续变量 xy 之间存在某种线性关系。回归需要解决的问题是，找到这种线性关系，从而预测新样本 X 的标签 y 值。

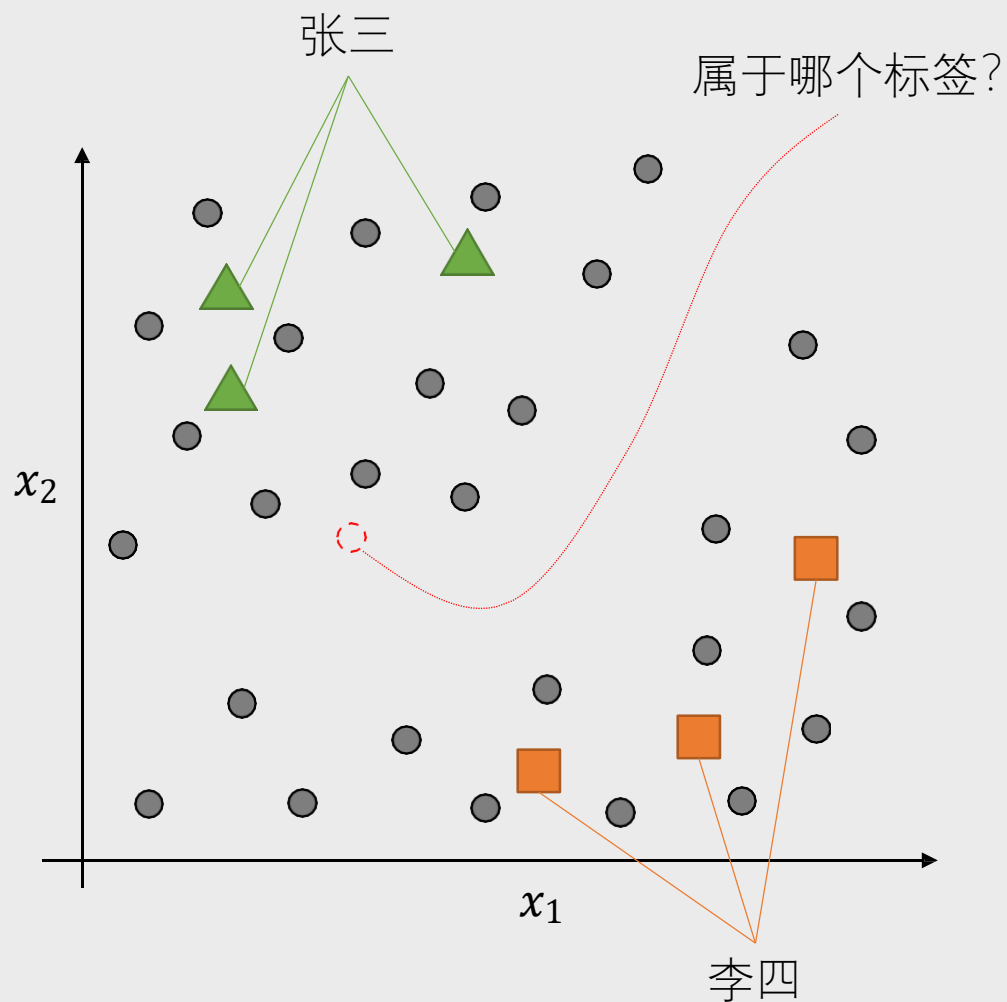
而非监督学习没有标签，只能分析特征值 x 的内在关系。因此非监督学习可能有多个答案。

机器学习概述

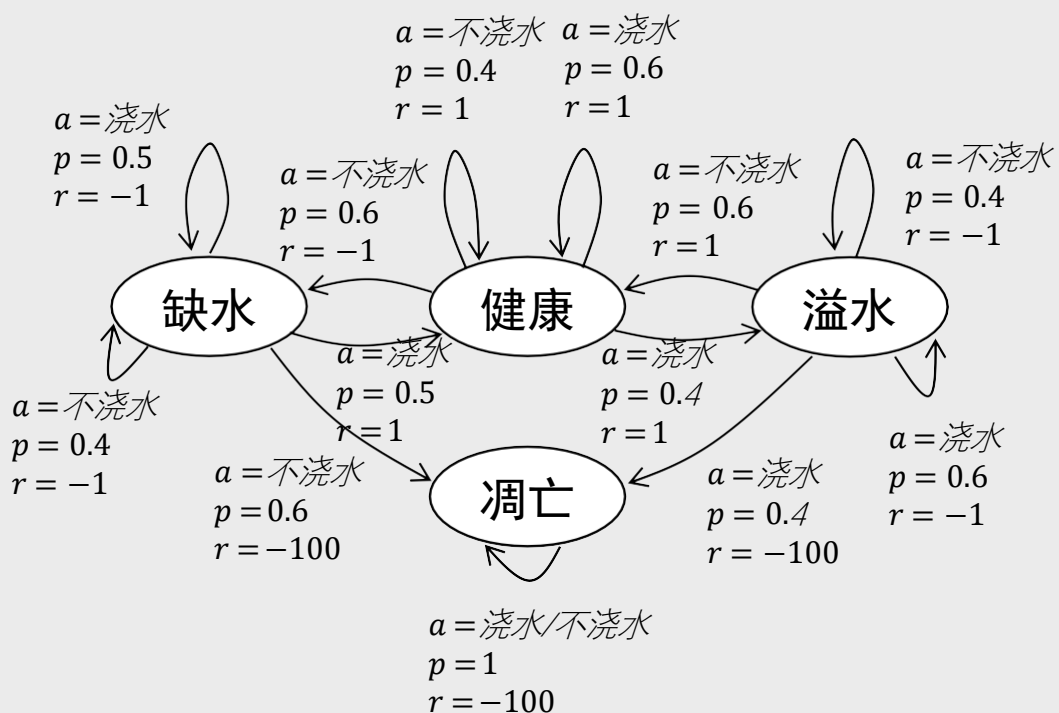
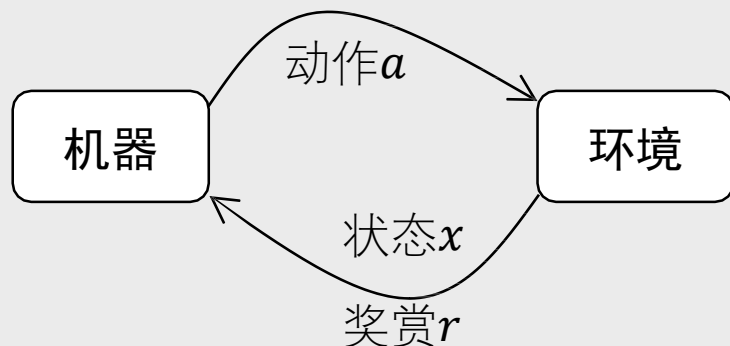
半监督学习

有些算法能够处理只有部分标签的训练集，而且通常来说是大部分数据都没有标签，只有少部分数据有标签，这种算法就叫做半监督学习。半监督学习示意图如图所示。

举例来说，智能手机越来越普及，智能程度越来越高，有的手机有智能相册，会自动按照照片中的人脸进行分类，如果你在拍照的时候，对某几个人进行了标注，那么智能相册也会对所有的照片进行标注，这就是半监督学习的典型应用。



机器学习概述

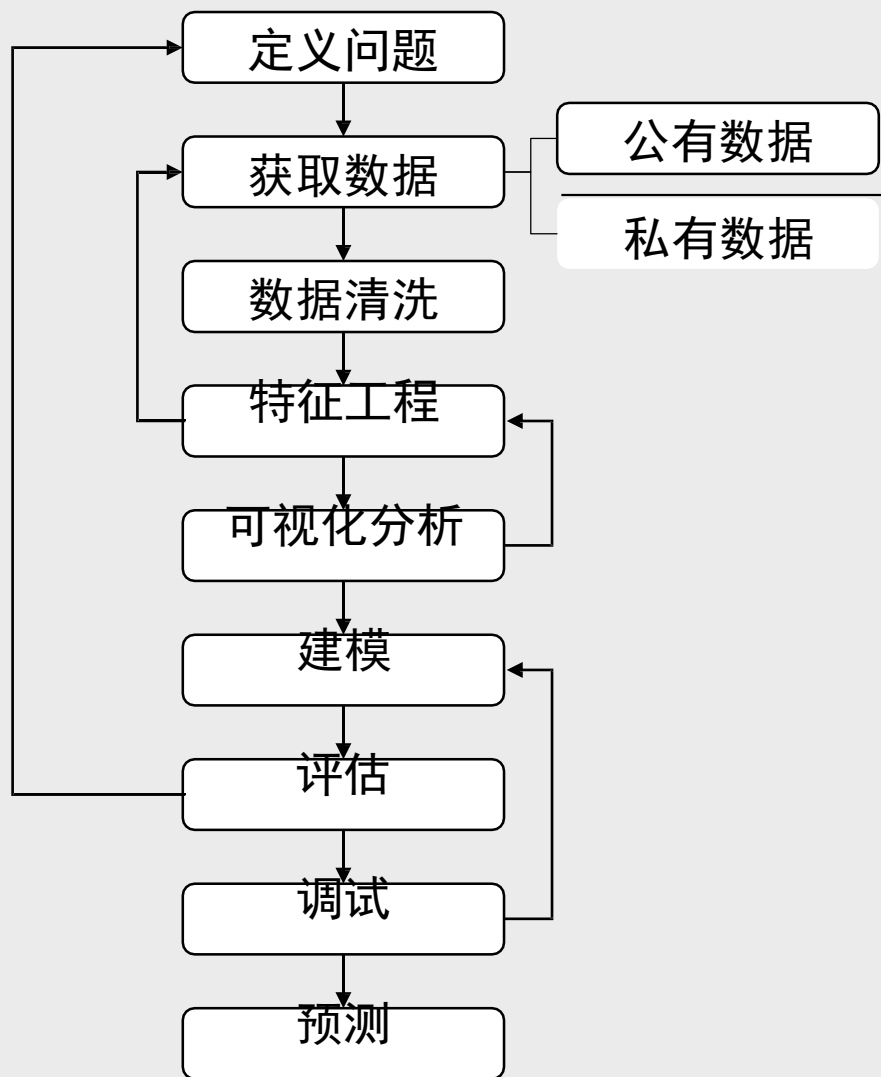


强化学习

将机器看作一个婴儿，强化学习就像婴儿学步，在学步的过程中，机器会观察周围的环境，然后选择下一步动作 a ，例如迈左腿，接下来返还一个状态 x 和对应的奖励 r ，如果没有摔倒，则返回状态正常，同时返回奖励+1，如果摔倒，则返回状态跌倒，同时返回奖励-100，这样不断摸索得到奖励最大的流程，机器就学会了走路。把这个过程抽象出来，就是强化学习。

如图，是一个简单的强化学习示意图。

一个更“复杂”的强化学习流程如图所示，该强化学习的目的是判断是否给西瓜浇水。



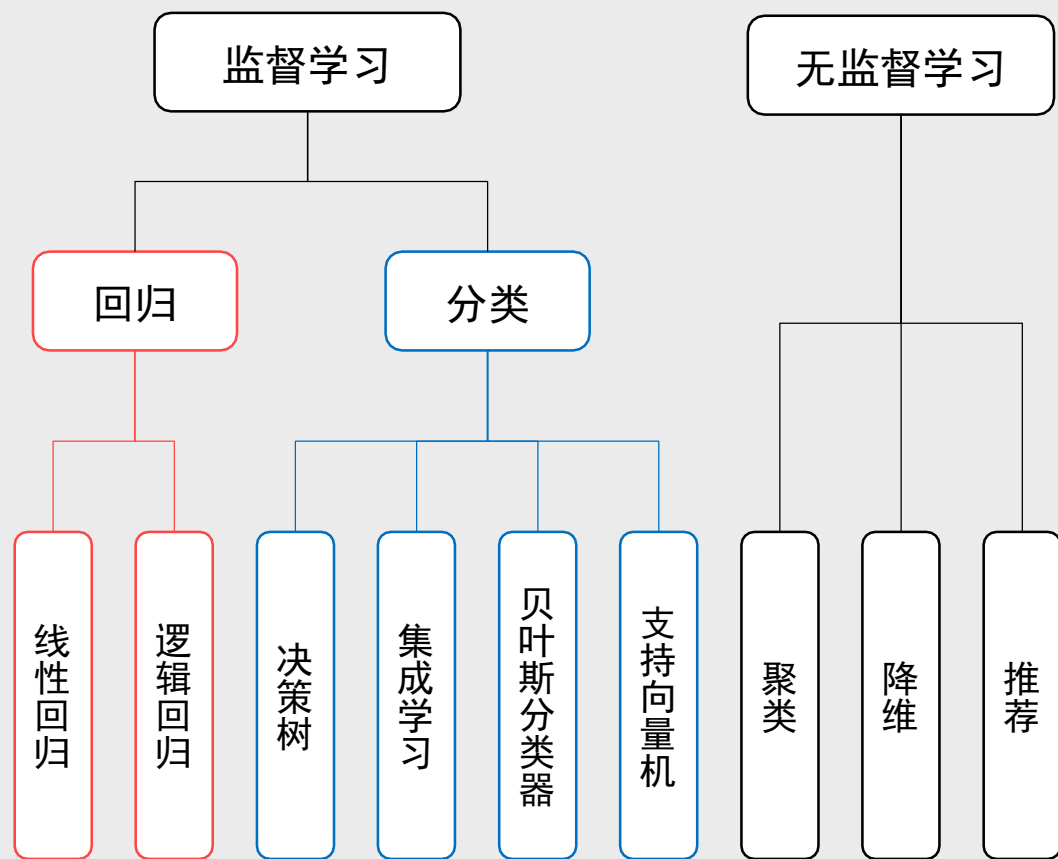
基本的建模流程

一个完整的基本建模流程如图所示

【思考】流程的核心是什么？

流程的核心是定义问题，是将业务问题转化为机器学习擅长解决的问题。举例来说，对于企业家，他提出的问题很有可能是“我怎样才能提高利润？”，显然这是一个业务问题，却不是机器学习擅长解决的问题，对于机器学习来说，问题应当是，我有这么一堆数据，其中哪些影响了我的利润？在多大程度上影响的？

课程目标与结构



课程目标与结构

课程目标是学会用机器学习解决实际问题

第一层目标：做个调包侠，学会调用python库实现机器学习算法

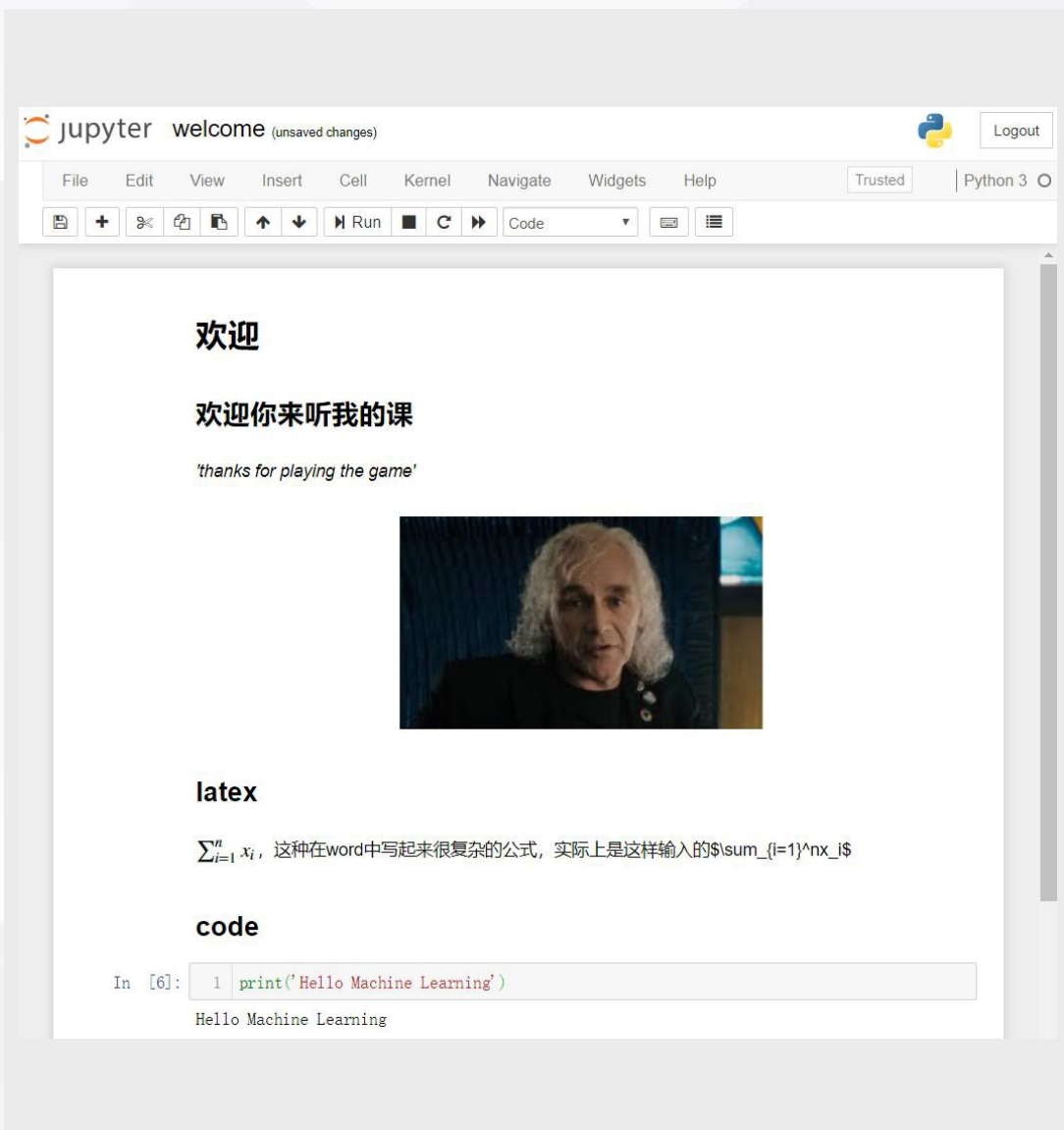
第二层目标：学习机器学习算法原理，知其然，知其所以然

第三层目标：能针对实际业务问题，有针对性的解决问题

课程结构包括传统机器学习中的监督学习与非监督学习。具体而言，包括线性回归、逻辑回归、决策树、集成学习、贝叶斯分类器、支持向量机、聚类、降维、推荐等九种算法。

每门算法包括理论和实验两部分组成。

Jupyter简介



Jupyter简介

通俗的说，jupyter就是在浏览器上运行的，可以跑代码的记事本。它具有以下优点：

支持代码种类繁多

轻量化的编辑器，安装简单

支持markdown语法

支持latex语法

逐代码块执行，完美搭配机器学习

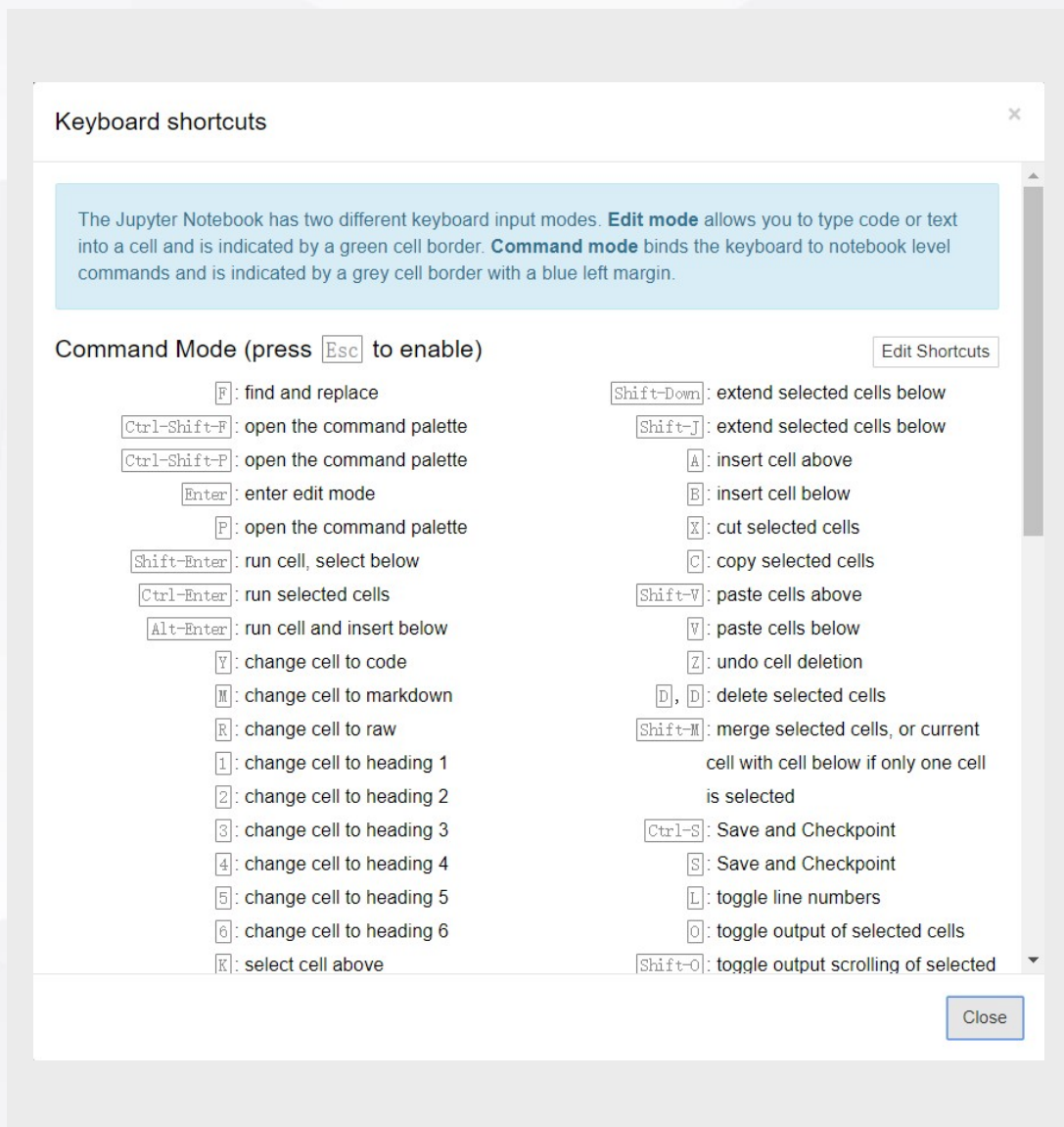
Jupyter的缺点之一是不方便转换为word格式，但是可以转换为html、pdf、md等格式。

Jupyter简介

Jupyter快捷键

Jupyter的大部分快捷键，都需要在非编辑模式下输入，按 h 可以查看快捷键帮助，常用快捷键如下：

- 编辑模式下ctrl+enter运行当前cell
- 编辑模式下shift+enter运行当前cell并移动到下个cell
- a，在当前cell前插入一个cell
- b，在当前cell后插入一个cell
- dd，删除当前cell
- m，将当前cell转化为markdown
- y，将当前cell转化为code（cell默认为code）
- z，撤销上一次对cell的操作



The screenshot shows the 'Keyboard shortcuts' dialog box in Jupyter. It has a title bar 'Keyboard shortcuts' with a close button. Below the title bar is a light blue informational box with text: 'The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code or text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level commands and is indicated by a grey cell border with a blue left margin.' Below this box, the text 'Command Mode (press `Esc` to enable)' is followed by an 'Edit Shortcuts' button. The main area contains two columns of keyboard shortcuts, each preceded by a key combination in a box. The first column includes shortcuts for find and replace, opening the command palette, entering edit mode, opening the command palette, running a cell, running selected cells, running a cell and inserting below, changing cell type (code, markdown, raw, heading 1-6), and selecting the cell above. The second column includes shortcuts for extending selected cells, inserting cells above/below, cutting/copying/pasting cells, undoing cell deletion, deleting selected cells, merging selected cells, saving and checkpointing, toggling line numbers/output, and toggling output scrolling. A 'Close' button is at the bottom right.

Keyboard shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code or text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level commands and is indicated by a grey cell border with a blue left margin.

Command Mode (press `Esc` to enable) Edit Shortcuts

<code>F</code> : find and replace	<code>Shift-Down</code> : extend selected cells below
<code>Ctrl-Shift-F</code> : open the command palette	<code>Shift-J</code> : extend selected cells below
<code>Ctrl-Shift-P</code> : open the command palette	<code>A</code> : insert cell above
<code>Enter</code> : enter edit mode	<code>B</code> : insert cell below
<code>P</code> : open the command palette	<code>X</code> : cut selected cells
<code>Shift-Enter</code> : run cell, select below	<code>C</code> : copy selected cells
<code>Ctrl-Enter</code> : run selected cells	<code>Shift-V</code> : paste cells above
<code>Alt-Enter</code> : run cell and insert below	<code>V</code> : paste cells below
<code>Y</code> : change cell to code	<code>Z</code> : undo cell deletion
<code>M</code> : change cell to markdown	<code>D, D</code> : delete selected cells
<code>R</code> : change cell to raw	<code>Shift-M</code> : merge selected cells, or current cell with cell below if only one cell is selected
<code>1</code> : change cell to heading 1	<code>Ctrl-S</code> : Save and Checkpoint
<code>2</code> : change cell to heading 2	<code>S</code> : Save and Checkpoint
<code>3</code> : change cell to heading 3	<code>L</code> : toggle line numbers
<code>4</code> : change cell to heading 4	<code>O</code> : toggle output of selected cells
<code>5</code> : change cell to heading 5	<code>Shift-O</code> : toggle output scrolling of selected
<code>6</code> : change cell to heading 6	
<code>K</code> : select cell above	

Close

Jupyter简介

一级标题

二级标题

三级标题

这是一段斜体文字

这是一段加粗文字

这是一段斜体加粗文字

~~这是删除线~~

```
import numpy as np
```

- 无序列表1
- 无序列表2

1. 有序列表
2. 有序列表

thanks for play the game

[这是某网站的超链接](#)

MarkDown

常用markdown语法如下：

一级标题

斜体 ****加粗**** ******撕佐功夫******

~~删除线~~

- 无序列表 1. 有序列表

> 引用

`code`

![图片注释](图片地址)

[超链接名](超链接地址)

注：markdown不支持直接输入换行、空格

Jupyter简介

x_1^2 : `x_1^2`

$\sum_{i=1}^n x_i^2$: `$\sum_{i=1}^n x_i^2$`

$x \geq y \leq z \neq s \times t$: `$x \geq y \leq z \neq s \times t$`

$x \in y \notin z \cap s \cup t$: `$x \in y \notin z \cap s \cup t$`

$\alpha\beta\gamma\sigma\delta\epsilon\Delta$: `$\alpha \beta \gamma \sigma \delta \epsilon \Delta$`

$\frac{\partial y}{\partial x}$: `$\frac{\partial y}{\partial x}$`

$Loss = (\hat{y} - y)^2 \times \sqrt{y}$: `$Loss=(\hat{y}-y)^2 \times \sqrt{y}$`

$\vec{a} \mathcal{XY} \boldsymbol{X}$: `$\vec{a} \mathcal{XY} \boldsymbol{X}$`

Latex

Latex是一种排版系统，尤其擅长公式排版，latex公式需要使用\$将公式内容包围起来，两个\$表示居中

x_1 表示下标 x_1 ， x^2 表示上标 x^2 ，

\neq 表示 \neq ， \geq 表示 \geq ， \leq 表示 \leq ， \times

\in 表示 \in ， \notin 表示 \notin ， \cap 表示 \cap ， \cup 表示 \cup

α 表示 α ， δ 表示 δ ， ϕ 表示 ϕ ， Δ 表示 Δ

∂ 表示 ∂ ， $\frac{a}{b}$ 表示 $\frac{a}{b}$

\hat{y} 表示 \hat{y} ， \bar{y} 表示 \bar{y} ， \sqrt{y} 表示 \sqrt{y}

\vec{a} 表示 \vec{a} ， \mathcal{X} 表示 \mathcal{X} ， \boldsymbol{X} 表示 \boldsymbol{X}

- Pandas
- Numpy
- Matplotlib
- SK-Learn