

## 1.机器学习概述

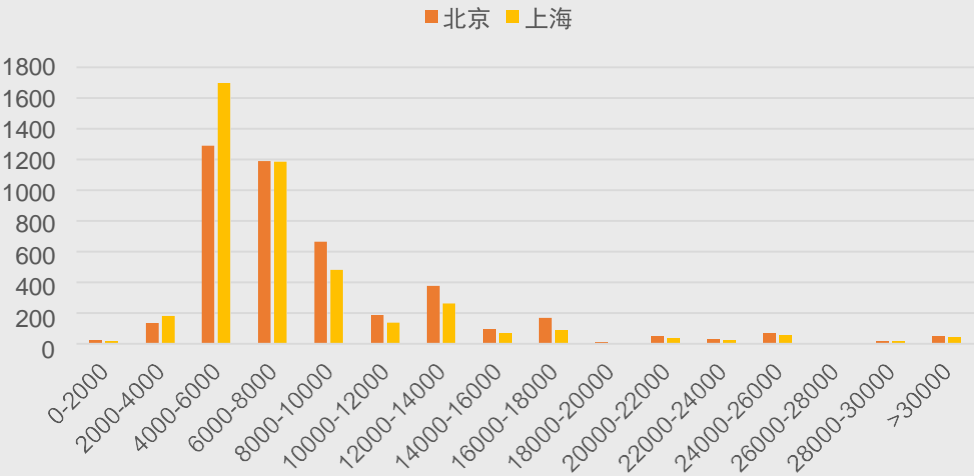
# 课程目录

Course catalogue

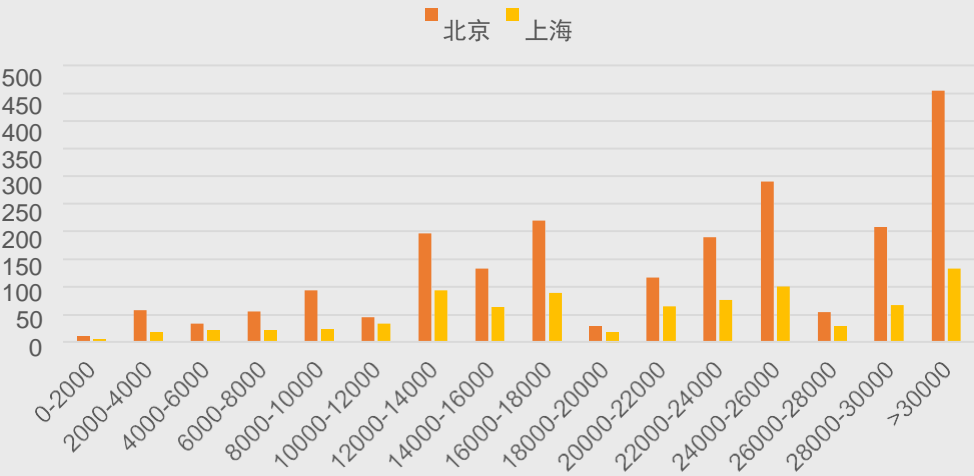
- 1/ 机器学习简介
- 2/ 机器学习、人工智能和数据挖掘
- 3/ 常用机器学习库
- 4/ Jupyter简介

# 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？



# 机器学习定义

## 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：

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

在不直接针对问题进行编程的情况下，赋予计算机学习能力的  
一个研究领域。

## 什么是机器学习

### ◆ 机器学习的形式化描述-Tom Mitchell (1998)

A computer program is said to *learn* from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$ .

“假设用 $P$ 来评估计算机程序在某任务类 $T$ 上的性能，若一个程序通过利用经验 $E$ 在 $T$ 中任务上获得了性能改善，则我们就说关于 $T$ 和 $P$ ，该程序对 $E$ 进行了学习”

# 机器学习定义

## 典型的机器学习过程



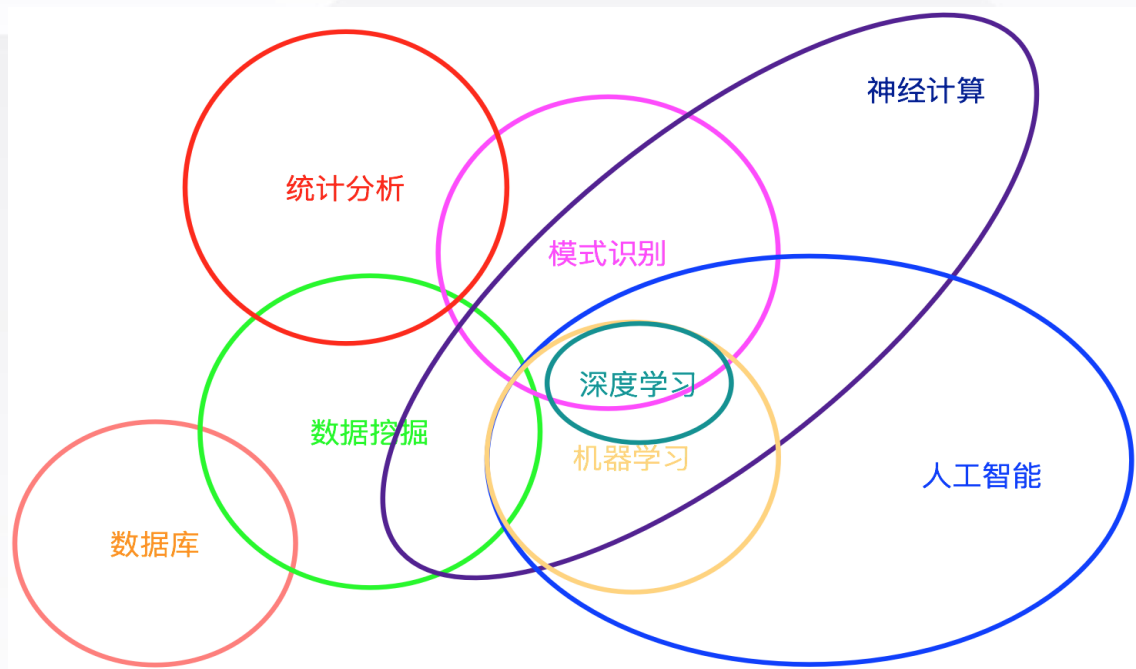
# 机器学习定义

## 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

# 人工智能、机器学习、深度学习

---



$x_1$



# 机器学习定义

## 机器学习应用领域

### 1. 计算机视觉

典型的应用包括：人脸识别、车牌识别、扫描文字识别、图片内容识别、图片搜索等等。

### 2. 自然语言处理

典型的应用包括：搜索引擎智能匹配、文本内容理解、文本情绪判断，语音识别、输入法、机器翻译等等。

### 3. 社会网络分析

典型的应用包括：用户画像、网络关联分析、欺诈作弊发现、热点发现等等。

### 4. 推荐

典型的应用包括：音乐的“歌曲推荐”，某宝的“猜你喜欢”等

### 5. 艺术创作

图片的识别、分类、生成、美化...

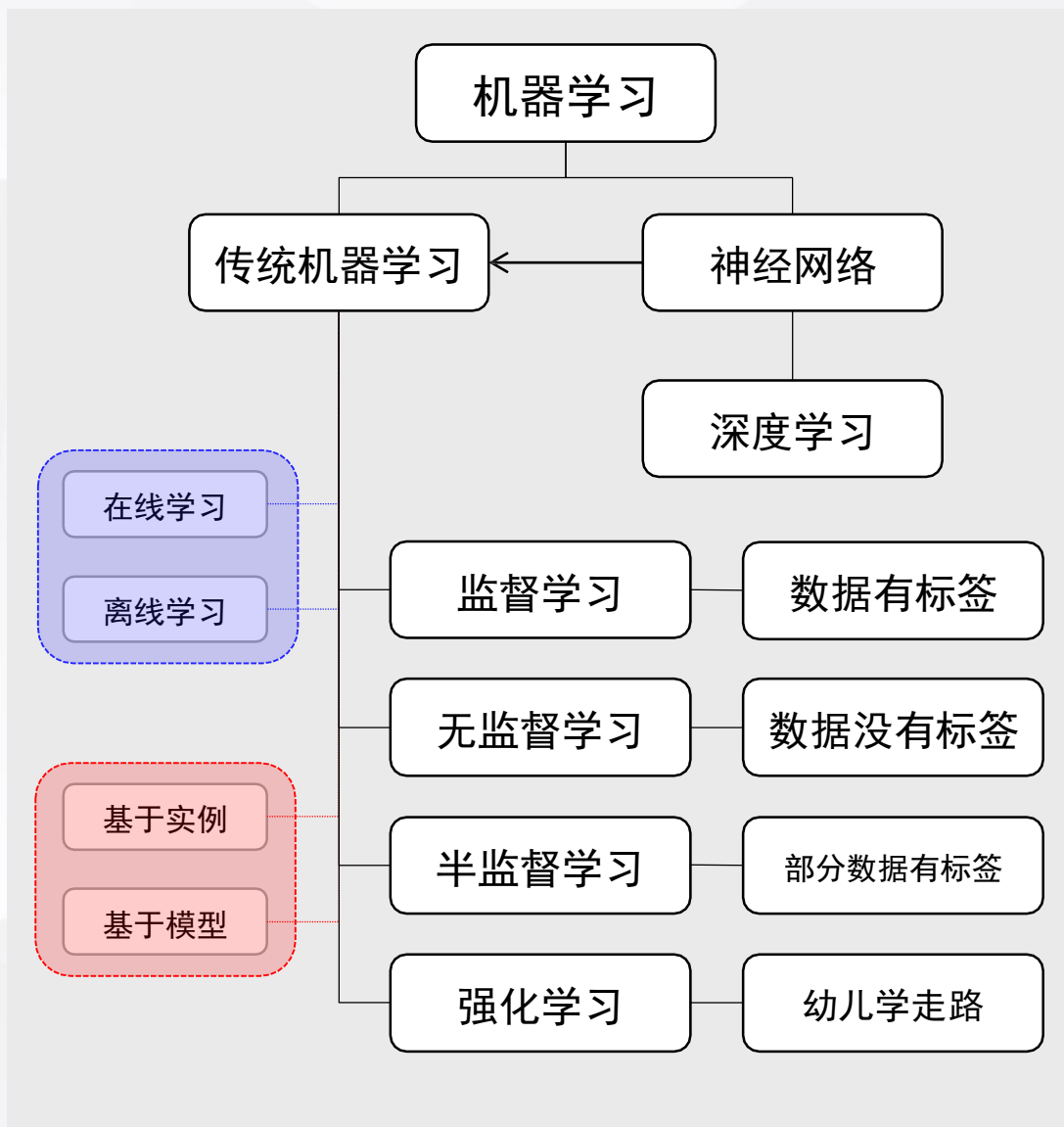
# 机器学习概述

## 机器学习框架

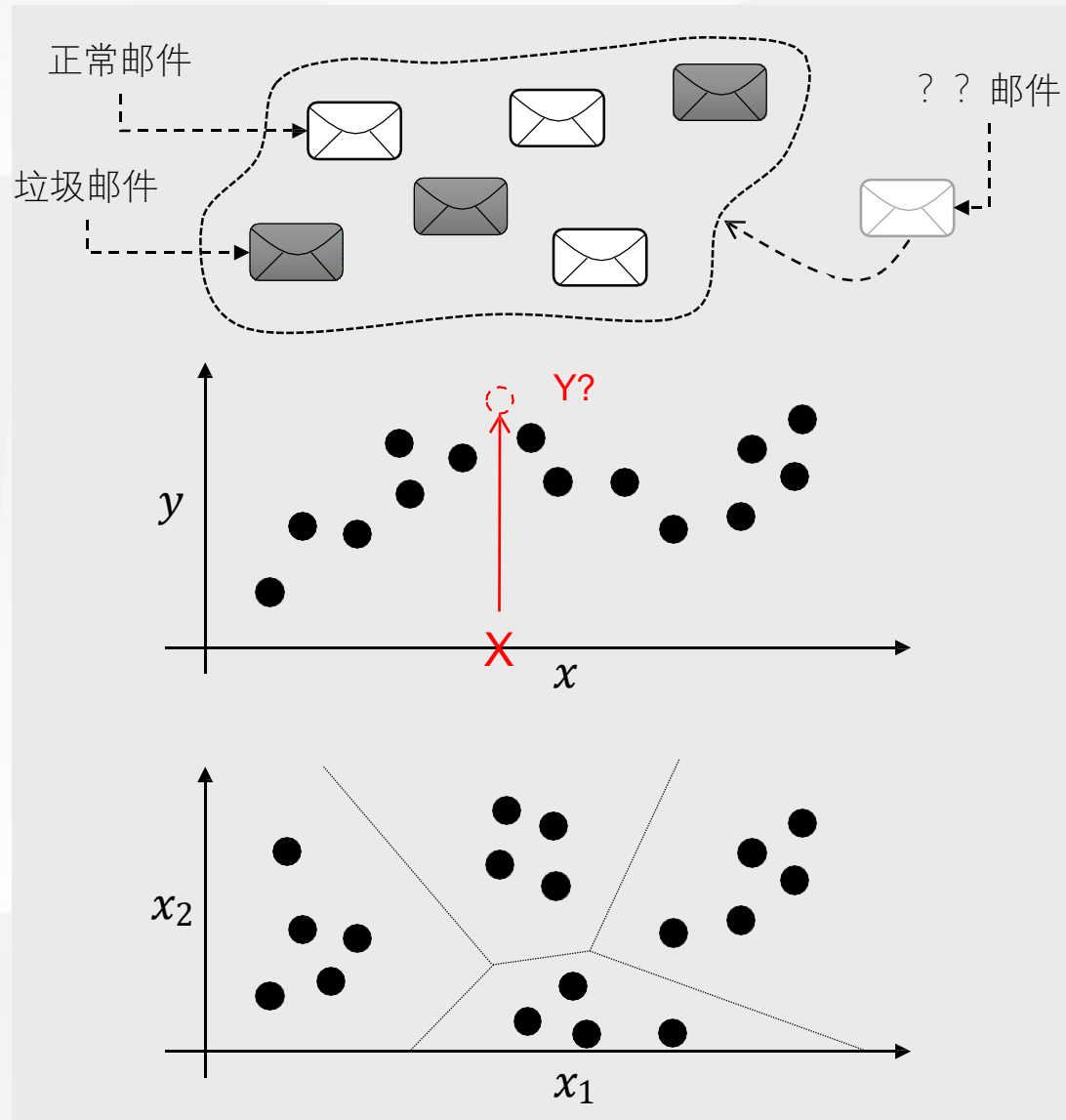
广义上的机器学习，包括传统机器学习和神经网络两部分。

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。

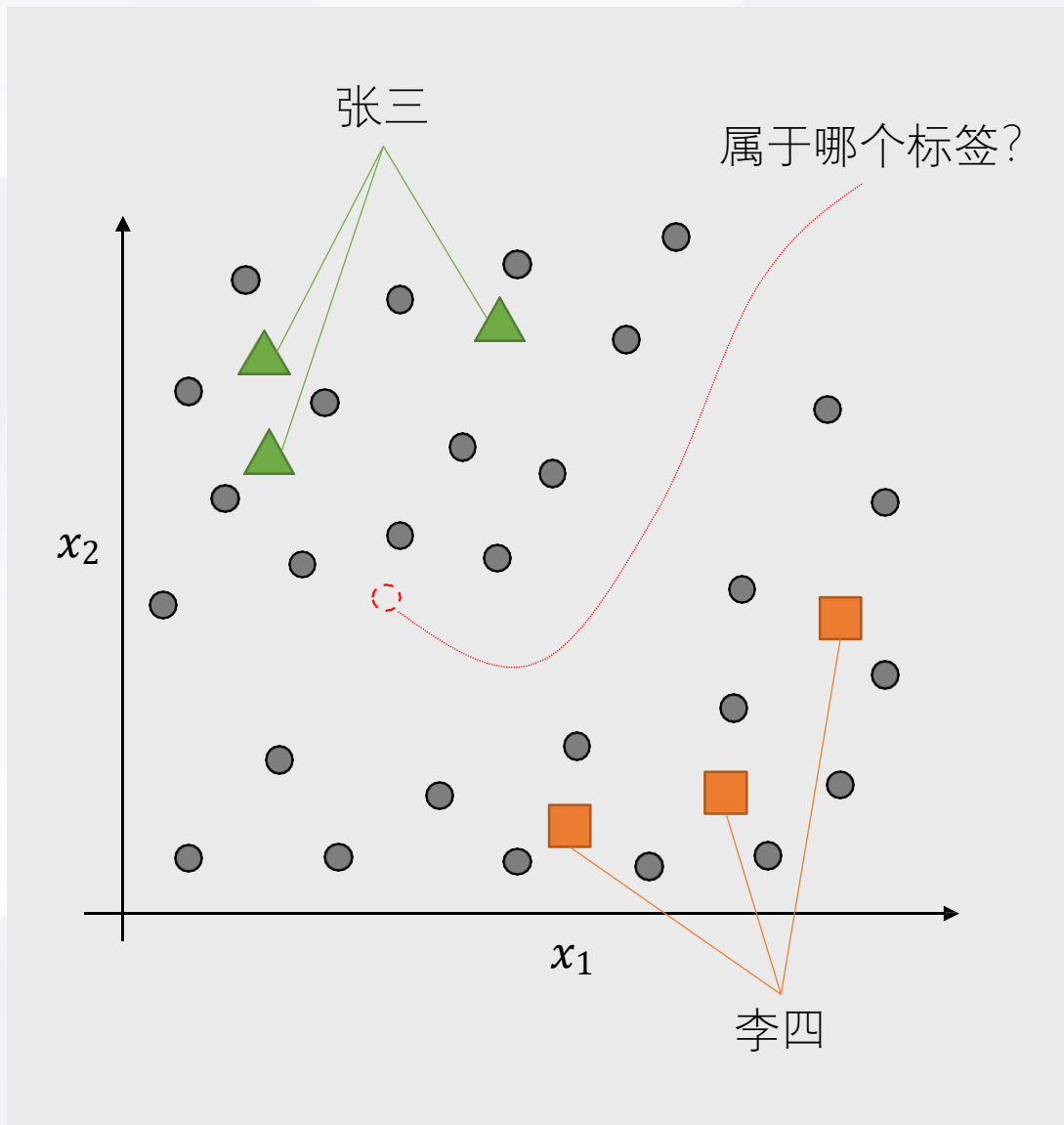
再比如，连续变量 $xy$ 之间存在某种线性关系。回归需要解决的问题是，找到这种线性关系，从而预测新样本 $X$ 的标签 $y$ 值。

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。

## 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。

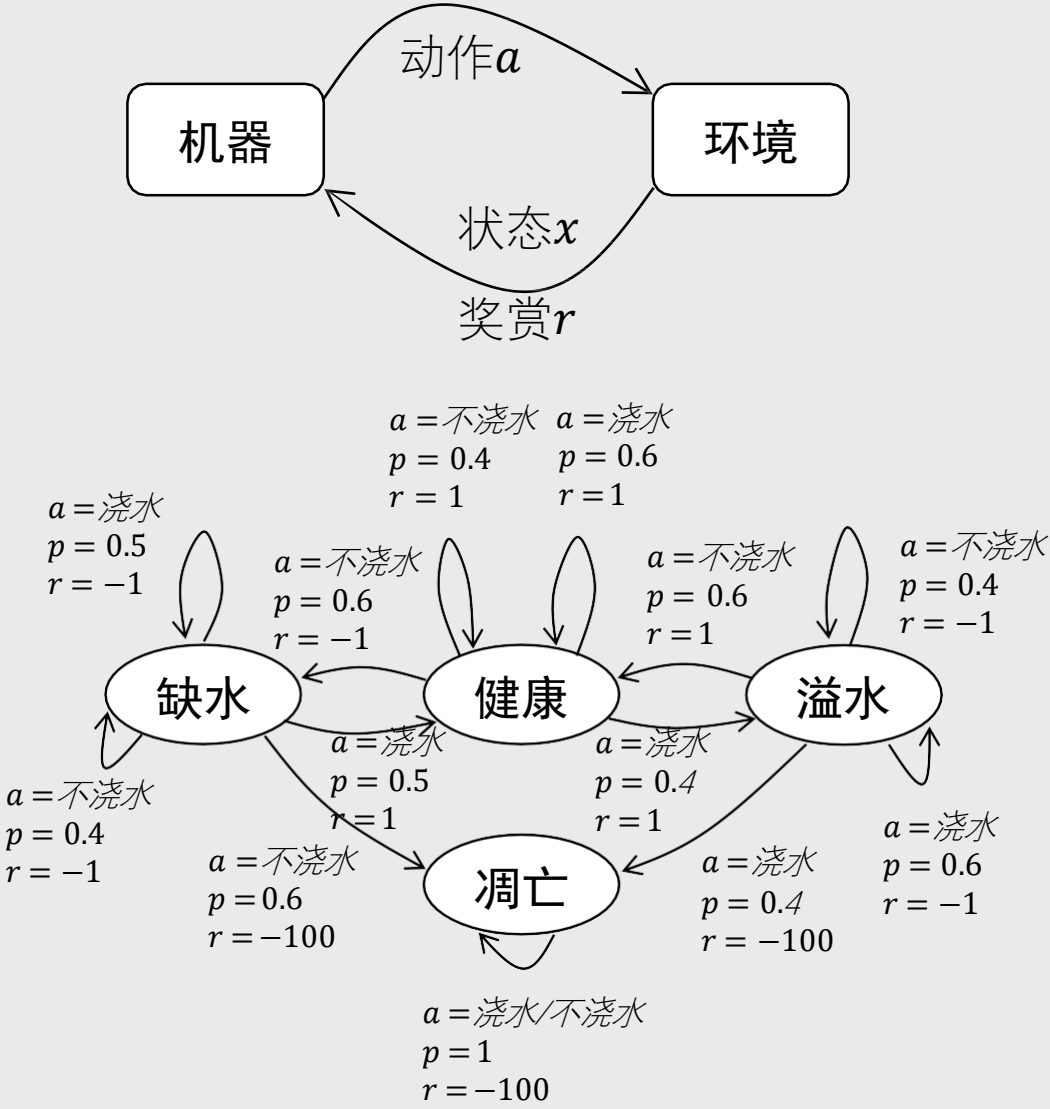


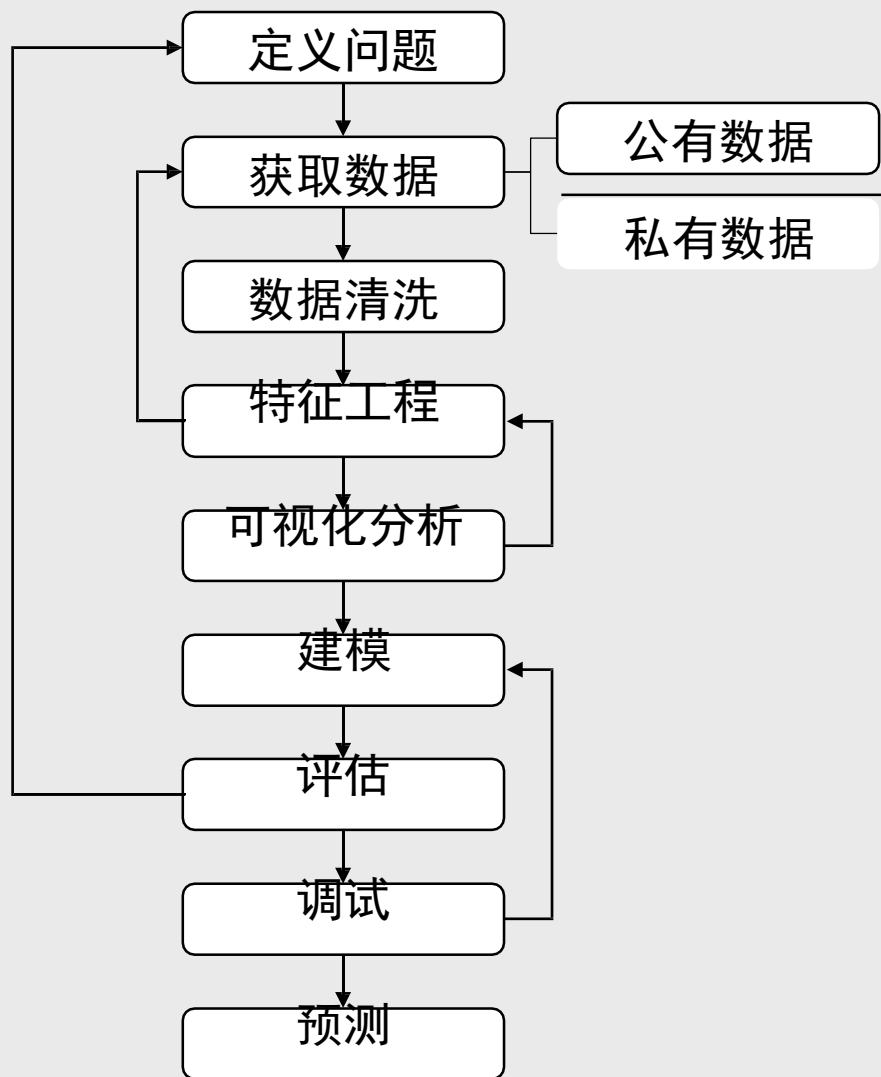
## 强化学习

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。





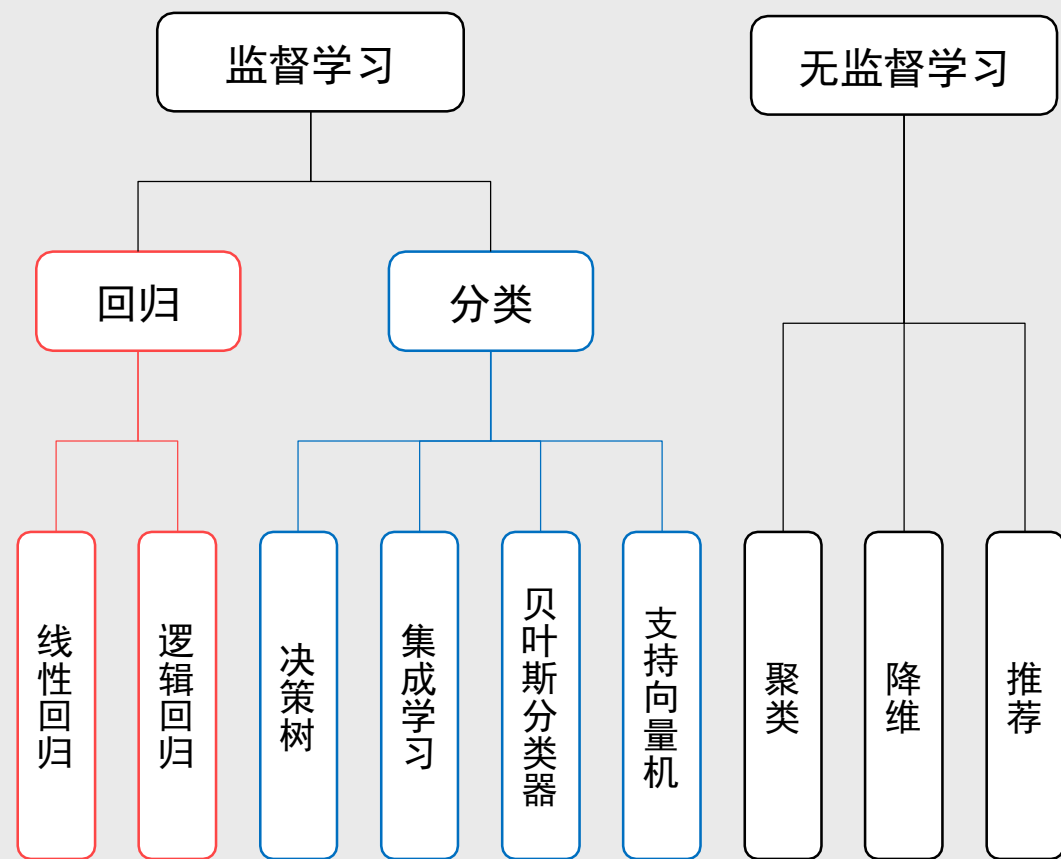
## 基本的建模流程

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。



# Machine Learning



what society thinks I  
do



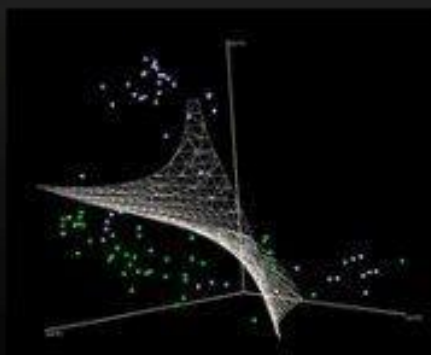
what my friends think  
I do



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



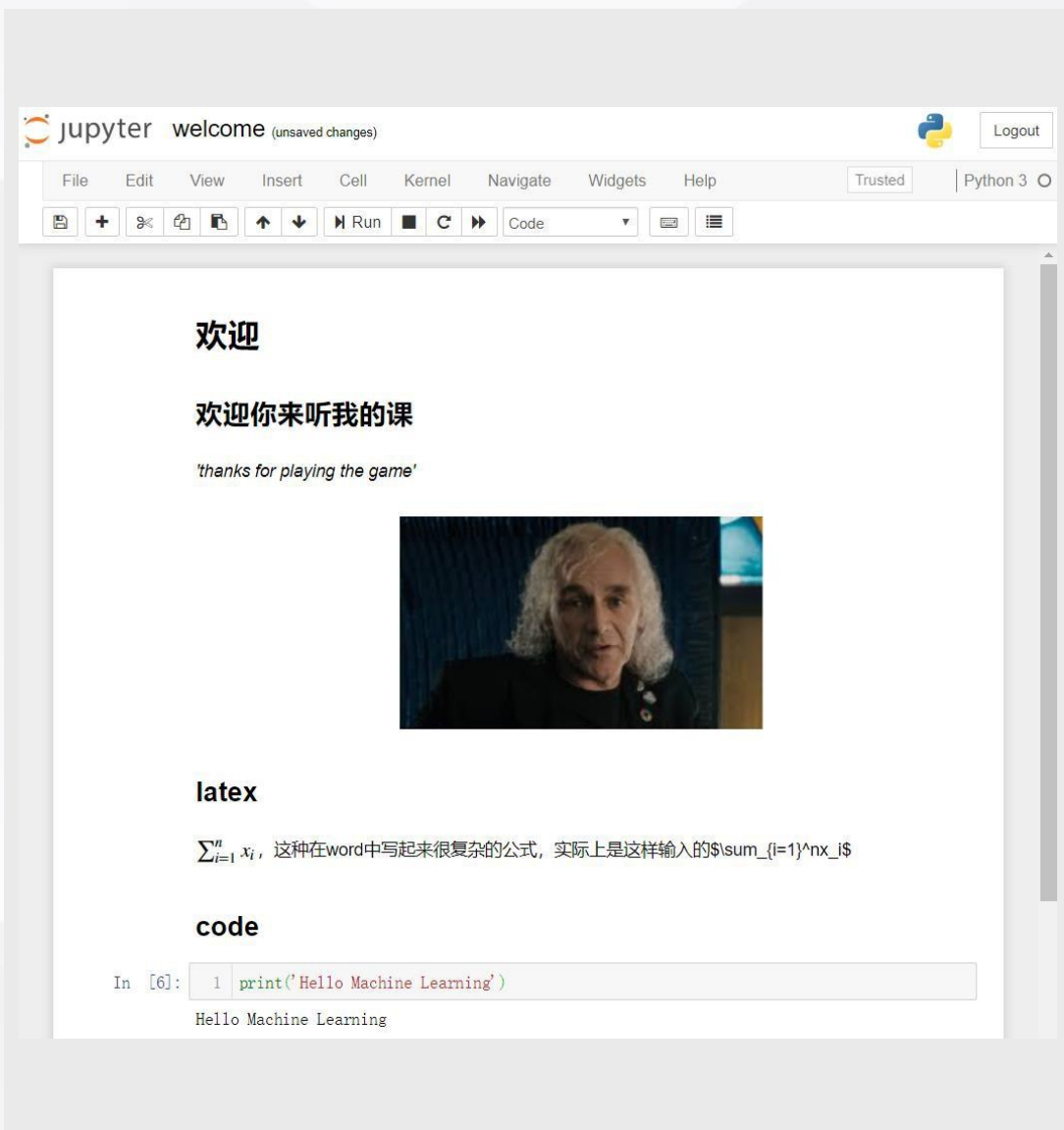
what I think I do

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

what I really do



# 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

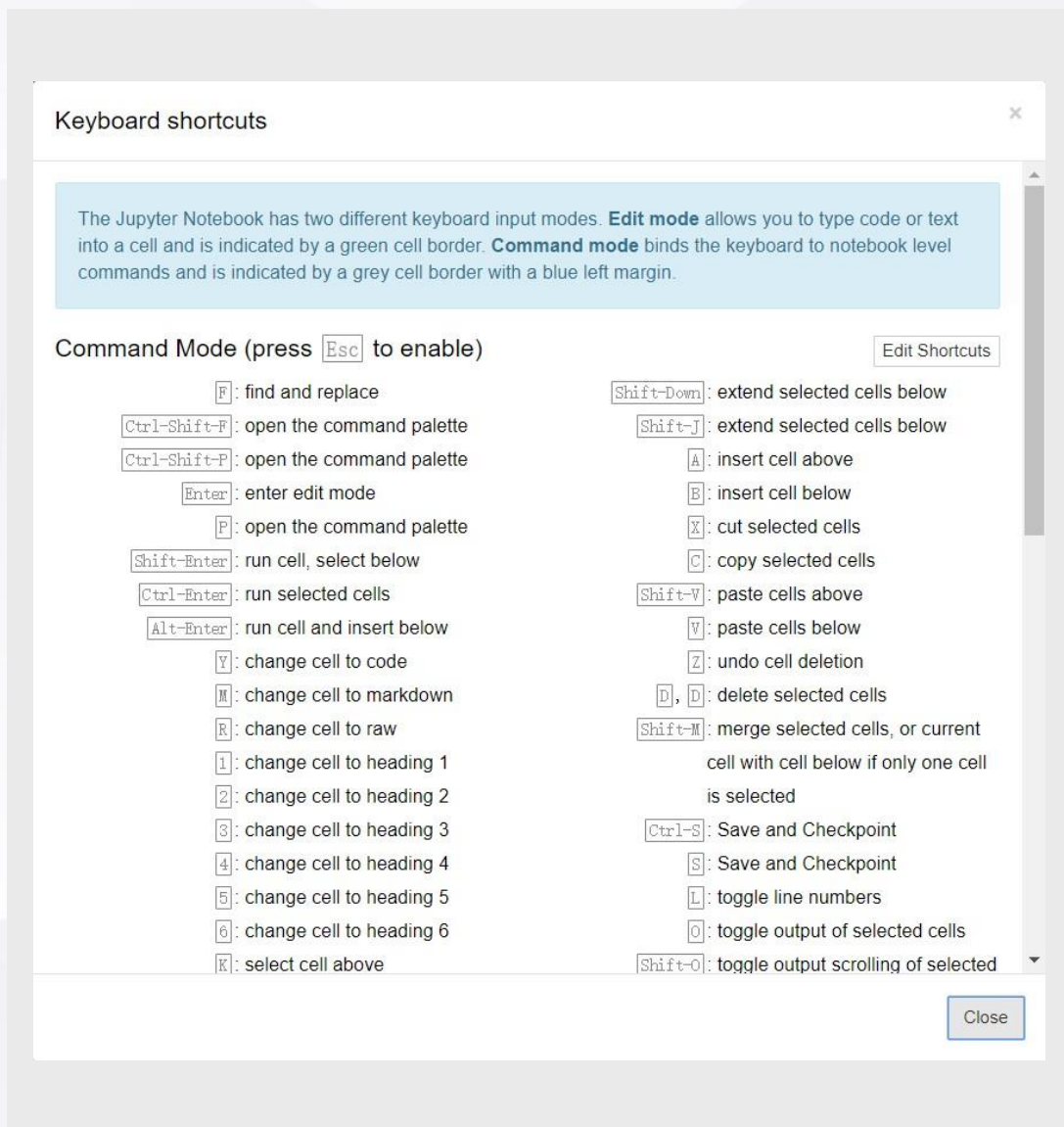
Jupyter的缺点之一是不方便转换为word格式，但是可以转换为html、pdf、md等格式。

# 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 with a button icon and a description. The first column includes shortcuts for finding and replacing, opening the command palette, entering edit mode, and changing cell types. The second column includes shortcuts for extending selections, inserting cells, cutting and copying, pasting, undoing deletions, deleting cells, merging cells, saving and checkpointing, toggling line numbers and output, and selecting cells. 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\*\***

***\*\*\*斜体加粗\*\*\****

~~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`表示 $\times$

`\in`表示 $\in$ ，`\notin`表示 $\notin$ ，`\cap`表示 $\cap$ ，`\cup`表示 $\cup$

`\alpha`表示 $\alpha$ ，`\delta`表示 $\delta$ ，`\phi`表示 $\phi$ ，`\Delta`表示 $\Delta$

`\partial`表示 $\partial$ ，`\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