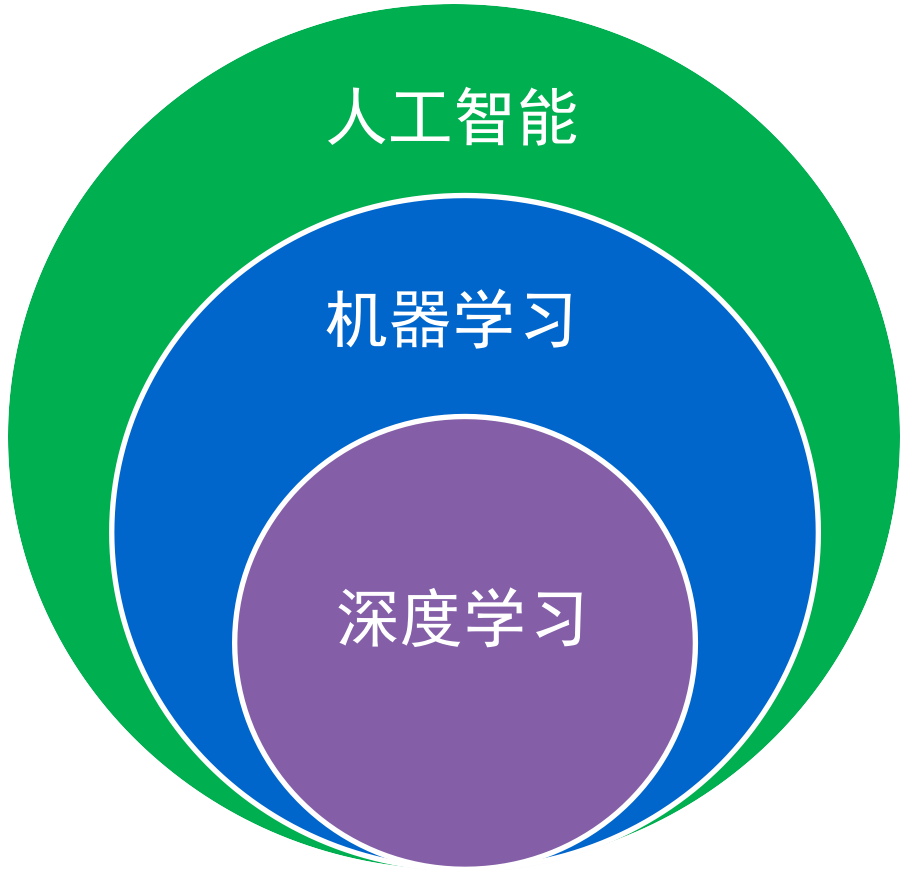
# 机器学习基础

2022.05.05

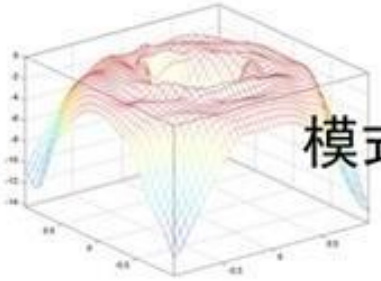
# 目录

# 1. 机器学习概述

- **人工智能：**机器展现的人类智能
- **机器学习：**计算机利用已有的数据(经验)，得出了某种模型，并利用此模型预测未来的一种方法
- **深度学习：**实现机器学习的一种技术



# 1. 机器学习概述—机器学习范围



模式识别

计算机视觉

数据挖掘

机器学习

语音识别

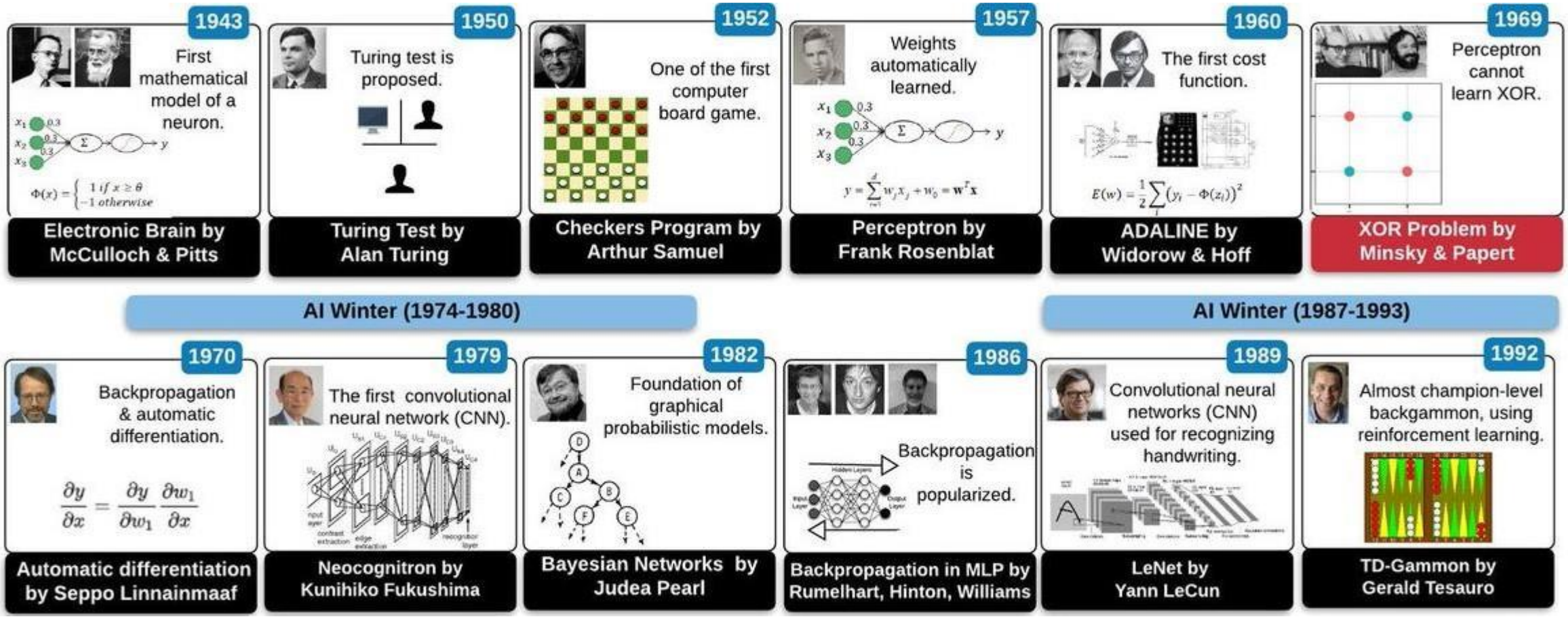
统计学习

自然语言处理

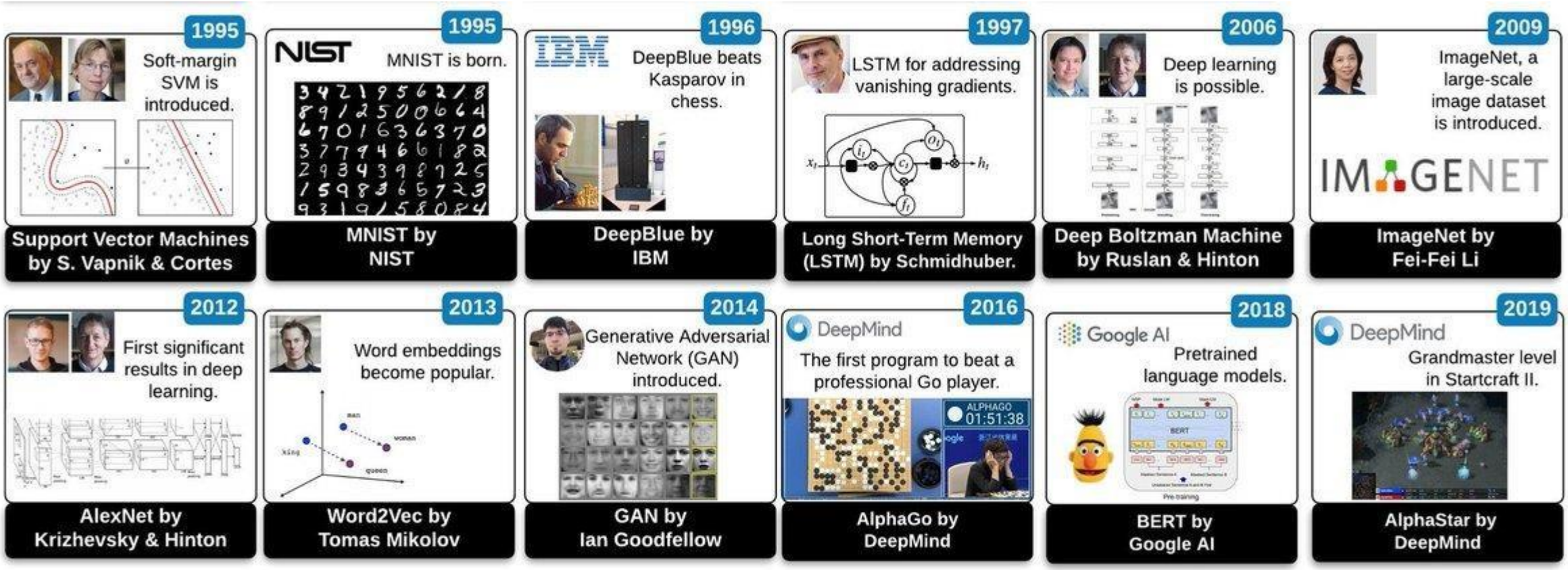
# 1. 机器学习概述—可以解决什么问题？

- 给定数据的预测问题
  - ✓ 数据清洗/特征选择
  - ✓ 确定算法模型/参数优化
  - ✓ 结果预测
- 不能解决什么
  - ✓ 大数据存储/并行计算
  - ✓ 做一个机器人

# 1. 机器学习概述—发展史


1943
First mathematical model of a neuron.
Electronic Brain by McCulloch & Pitts
1950
Turing test is proposed.
Turing Test by Alan Turing
1952
One of the first computer board game.
Checkers Program by Arthur Samuel
1957
Weights automatically learned.
Perceptron by Frank Rosenblat
1960
The first cost function.
ADALINE by Widorow & Hoff
1969
Perceptron cannot learn XOR.
XOR Problem by Minsky & Papert
AI Winter (1974-1980)
AI Winter (1987-1993)
1970
Backpropagation & automatic differentiation.
Automatic differentiation by Seppo Linnainmaaf
1979
The first convolutional neural network (CNN).
Neocognitron by Kunihiko Fukushima
1982
Foundation of graphical probabilistic models.
Bayesian Networks by Judea Pearl
1986
Backpropagation is popularized.
Backpropagation in MLP by Rumelhart, Hinton, Williams
1989
Convolutional neural networks (CNN) used for recognizing handwriting.
LeNet by Yann LeCun
1992
Almost champion-level backgammon, using reinforcement learning.
TD-Gammon by Gerald Tesauro

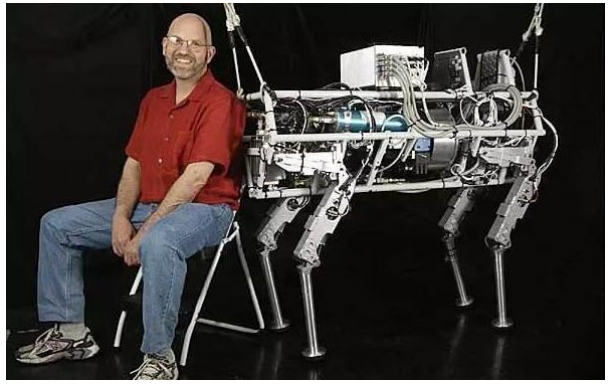
# 1. 机器学习概述—发展史




Parisa Rashidi, May 2020. CC BY 4.0

# 1. 机器学习概述—不同视角的机器学习

不同行业的人以为我做的事情



父母以为我做的事情

朋友以为我做的事情

$$\frac{\partial}{\partial w}L(w,b,\alpha)=w-\sum\alpha_i y_i x_i=0,\ w=\sum\alpha_i y_i x_i$$
$$\frac{\partial}{\partial b}L(w,b,\alpha)=\sum\alpha_i y_i=0$$

代入$L(w,b,\alpha)$

$$\min\ L(w,b,\alpha)=\frac{1}{2}||w||^2+\sum_{i=1}^m\alpha_i(-y_i(w^Tx_i+b)+1)$$
$$=\frac{1}{2}w^Tw-\sum_{i=1}^m\alpha_i y_i w^Tx_i-b\sum_{i=1}^m\alpha_i y_i+\sum_{i=1}^m\alpha_i$$
$$=\frac{1}{2}w^T\sum\alpha_i y_i x_i-\sum_{i=1}^m\alpha_i y_i w^Tx_i+\sum_{i=1}^m\alpha_i$$
$$=\sum_{i=1}^m\alpha_i-\frac{1}{2}\sum_{i=1}^m\alpha_i y_i w^Tx_i$$
$$=\sum_{i=1}^m\alpha_i-\frac{1}{2}\sum_{i,j=1}^m\alpha_i\alpha_j y_i y_j(x_i x_j)$$

再把 max 问题转成 min 问题：

$$\max\ \sum_{i=1}^m\alpha_i-\frac{1}{2}\sum_{i,j=1}^m\alpha_i\alpha_j y_i y_j(x_i x_j)=\min\frac{1}{2}\sum_{i,j=1}^m\alpha_i\alpha_j y_i y_j(x_i x_j)-\sum_{i=1}^m\alpha_i$$
$$s.t.\ \sum_{i=1}^m\alpha_i y_i=0,$$

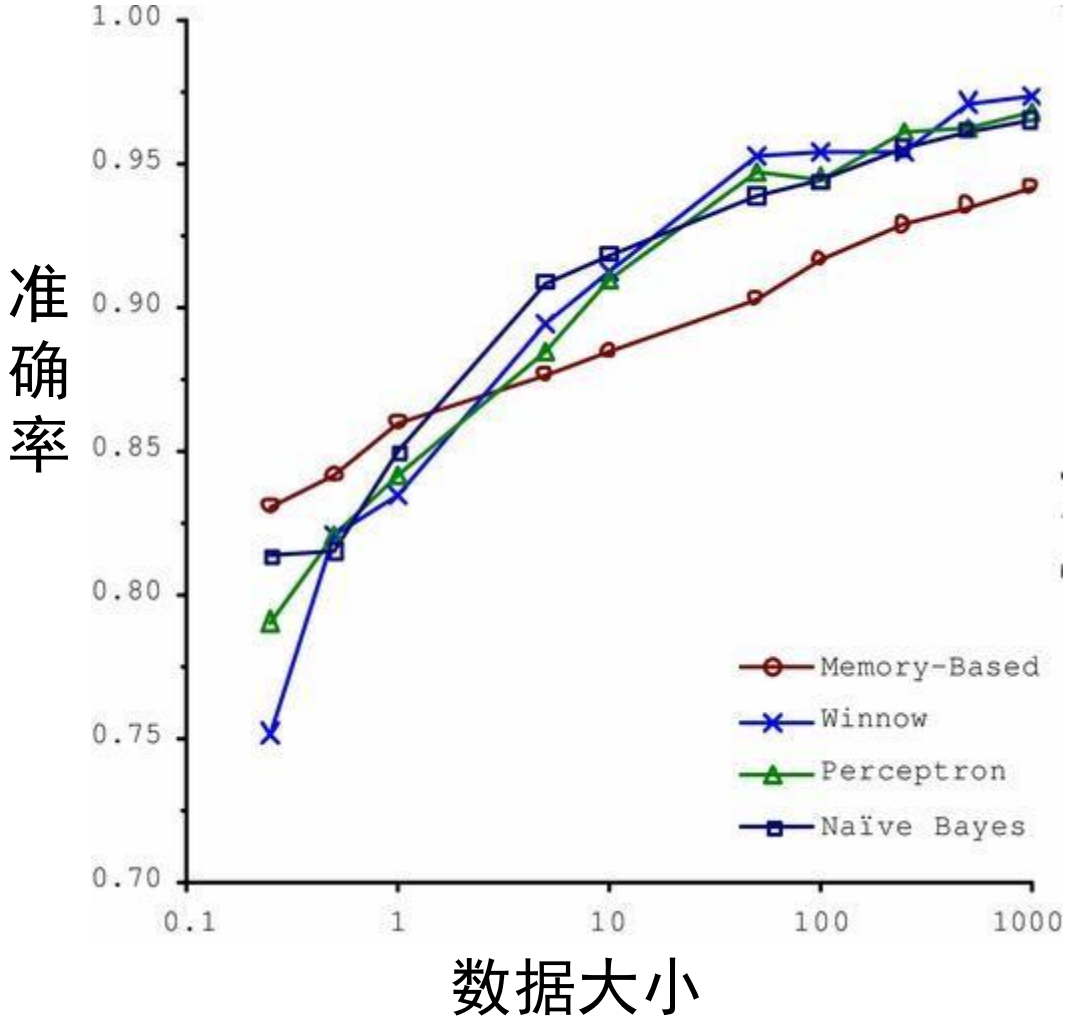
程序员以为我做的事情

我自己以为我做的事情

```
import xgboost as xgb
import numpy as np
```

实际上我做的事情

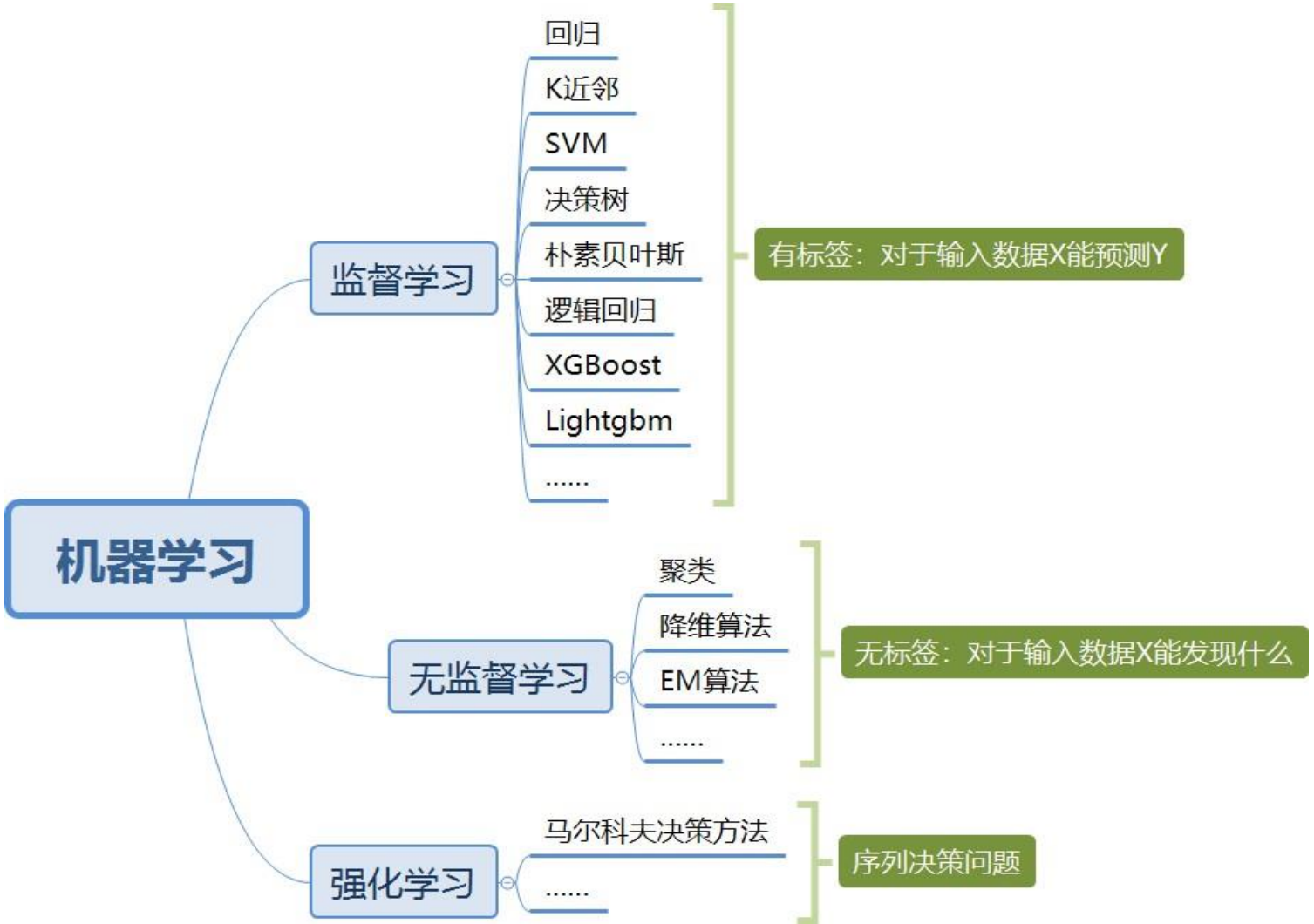
# 1. 机器学习概述—数据决定一切




通过这张图可以看出，各种不同算法在输入的数据量达到一定级数后，都有相近的高准确度。于是诞生了机器学习界的名言：

**成功的机器学习应用不是拥有最好的算法，而是拥有最多的数据！**

# 2. 机器学习的类型


机器学习
监督学习
回归
K近邻
SVM
决策树
朴素贝叶斯
逻辑回归
XGBoost
Lightgbm
.....
有标签：对于输入数据X能预测Y
无监督学习
聚类
降维算法
EM算法
.....
无标签：对于输入数据X能发现什么
强化学习
马尔科夫决策方法
.....
序列决策问题

# 2. 机器学习的类型—监督学习

- ✓分类（Classification）
  - ✓身高1.65m，体重100kg的男人肥胖吗？
  - ✓根据肿瘤的体积、患者的年龄来判断良性或恶性？
- ✓回归（Regression、Prediction）
  - ✓如何预测上海浦东的房价？
  - ✓未来的股票市场走向？

# 2. 机器学习的类型—无监督学习

- ✓聚类（Clustering）
  - ✓如何将教室里的学生按爱好、身高划分为5类？
- ✓降维（Dimensionality Reduction）
  - ✓如何将将原高维空间中的数据点映射到低维度的空间中？

# 2. 机器学习的类型—强化学习

- ✓强化学习（Reinforcement Learning）
  - ✓ 用于描述和解决智能体（agent）在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题 。

# 3. 机器学习背景知识

- 数学基础
  - ✓ 高等数学：导数、微分、泰勒公式…
  - ✓ 线性代数：向量、矩阵…
  - ✓ 概率论与数理统计：概率基本公式、常见分布、期望、协方差…
- Python基础
  - ✓ 环境安装：Anaconda、Jupyter Notebook、Pycharm、VSCode
  - ✓ Python基础
  - ✓ Python库：numpy、pandas、scipy、matplotlib、scikit-learn…

# 3. 机器学习背景知识

- Python基础—numpy

NumPy是一个用Python实现的科学计算的扩展程序库，包括：

1、一个强大的N维数组对象Array；

2、比较成熟的（广播）函数库；

3、用于整合C/C++和Fortran代码的工具包；

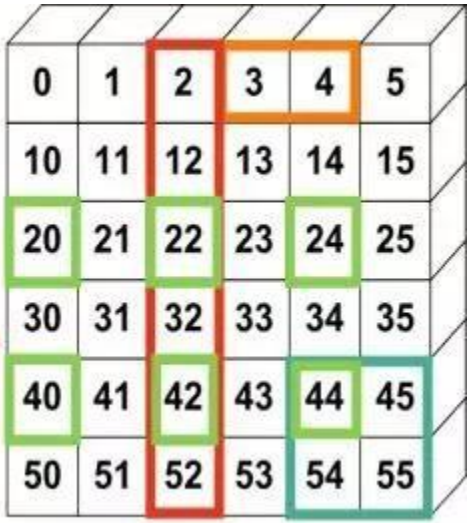
4、实用的线性代数、傅里叶变换和随机数生成函数。numpy和稀疏矩阵运算包scipy配合使用更加方便。

NumPy（Numeric Python）提供了许多高级的数值编程工具，如：矩阵数据类型、矢量处理，以及精密的运算库。专为进行严格的数字处理而产生。多为很多大型金融公司使用，以及核心的科学计算组织如：Lawrence Livermore，NASA用其处理一些本来使用C++，Fortran或Matlab等所做的任务。

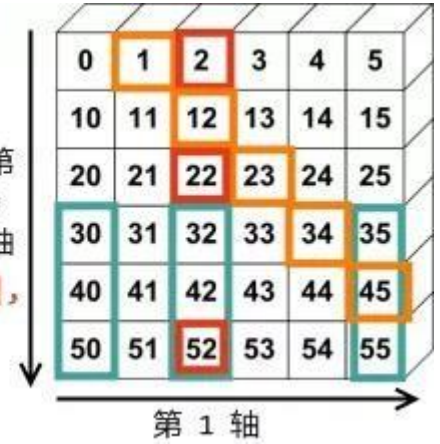
# 3. 机器学习背景知识

- Python基础—numpy 切片

```
>>> a[0,3:5]
array([3,4])
>>> a[4:,4:]
array([[44,45],[54,55]])
>>> a[:,2]
array([2,12,22,32,42,52])
>>> a[2::2,::2]
array([[20,22,24],
       [40,42,44]])
```

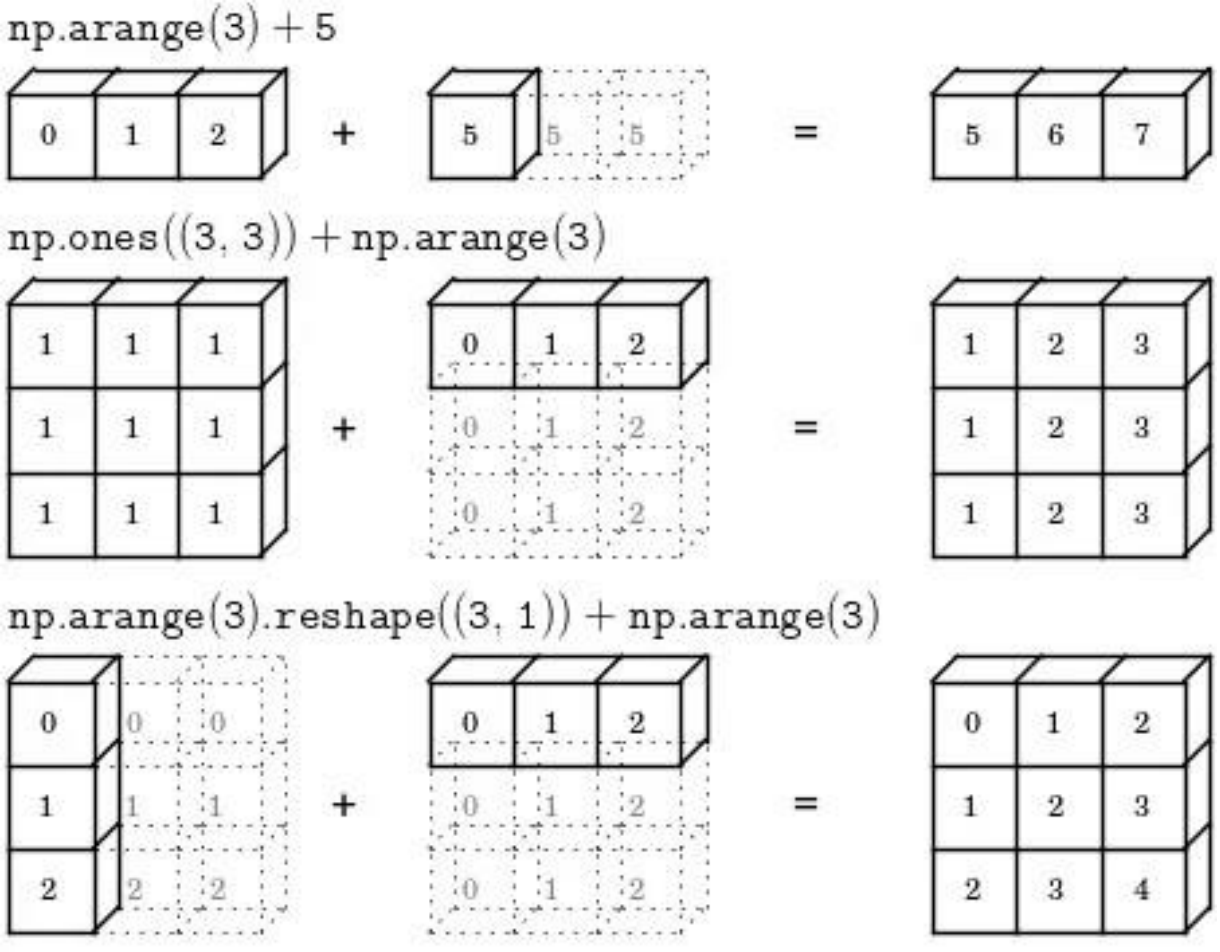



```
>>> a[(0,1,2,3,4),(1,2,3,4,5)]
array([1,12,23,34,45])
>>> a[3:,[0,2,5]]
array([[30,32,35],
       [40,42,45],
       [50,52,55]])
>>> mask=np.array([1,0,1,0,0,1],
                  dtype=np.bool)
>>> a[mask,2]
array([2,22,52])
```



# 3. 机器学习背景知识

- Python基础—numpy 广播



# 3. 机器学习背景知识

- Python基础—pandas

Pandas 是基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。

Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。Pandas提供了大量能使我们快速便捷地处理数据的函数和方法。你很快就会发现，它是使Python成为强大而高效的数据分析环境的重要因素之一。

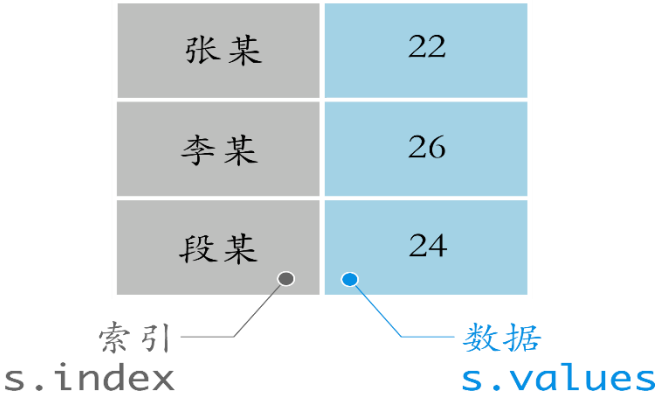
# 3. 机器学习背景知识

- Python基础—pandas

## ● 基本数据结构

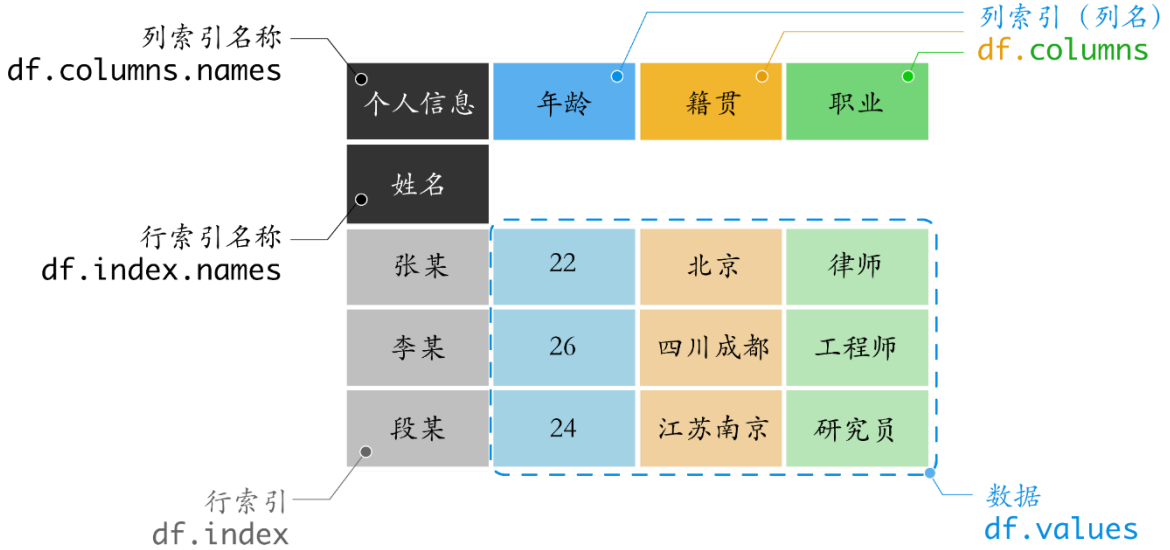
Series

一维数据结构，包含行索引和数据两个部分




DataFrame

二维数据结构，包含带索引的多列数据，各列的数据类型可能不同



# 3. 机器学习背景知识

- Python基础—pandas

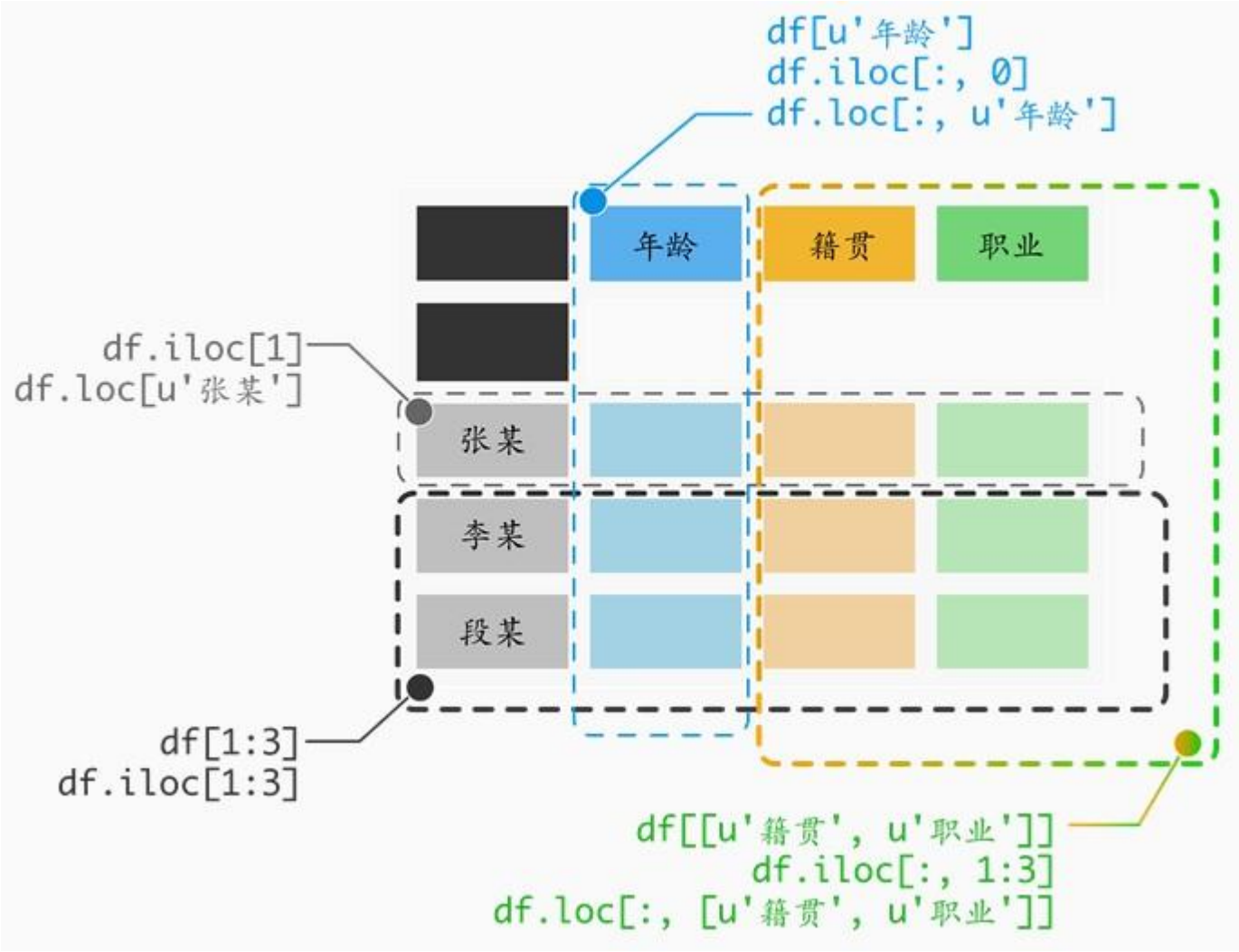
## ● 数据索引

df[5:10]
通过切片方式选取多行

df[col_label] or df.col_label
选取列

df.loc[row_label, col_label]
通过标签选取行/列

df.iloc[row_loc, col_loc]
通过位置（自然数）选取行/列



# 3. 机器学习背景知识

- Python基础—pandas

## ● 数据合并

pd.merge(left, right) 类数据库的数据融合操作.

参数：how，融合方式，包括左连接、右连接、内连接（默认）和外连接；on，连接键；left_on，左键；right_on，右键；left_index，是否将left行索引作　为左键；right_index，是否将right行索引作为右键.

| | 姓名 | 年龄 |
|---|---|---|
| 0 | 张某 | 22 |
| 1 | 李某 | 26 |
| 2 | 段某 | 24 |

left

| | 姓名 | 籍贯 |
|---|---|---|
| 7 | 张某 | 北京 |
| 8 | 李某 | 四川成都 |
| 9 | 钱某 | 江苏南京 |

right

inner

| | 姓名 | 年龄 | 籍贯 |
|---|---|---|---|
| 0 | 张某 | 22 | 北京 |
| 1 | 李某 | 26 | 四川成都 |

```
pd.merge(left, right,
         how='inner', on=u'姓名')
```

outer

| | 姓名 | 年龄 | 籍贯 |
|---|---|---|---|
| 0 | 张某 | 22.0 | 北京 |
| 1 | 李某 | 26.0 | 四川成都 |
| 2 | 段某 | 24.0 | NaN |
| 3 | 钱某 | NaN | 江苏南京 |

```
pd.merge(left, right,
         how='outer', on=u'姓名')
```

left

| | 姓名 | 年龄 | 籍贯 |
|---|---|---|---|
| 0 | 张某 | 22 | 北京 |
| 1 | 李某 | 26 | 四川成都 |
| 2 | 段某 | 24 | NaN |

```
pd.merge(left, right,
         how='left', on=u'姓名')
```

right

| | 姓名 | 年龄 | 籍贯 |
|---|---|---|---|
| 0 | 张某 | 22.0 | 北京 |
| 1 | 李某 | 26.0 | 四川成都 |
| 2 | 钱某 | NaN | 江苏南京 |

```
pd.merge(left, right,
         how='right', on=u'姓名')
```

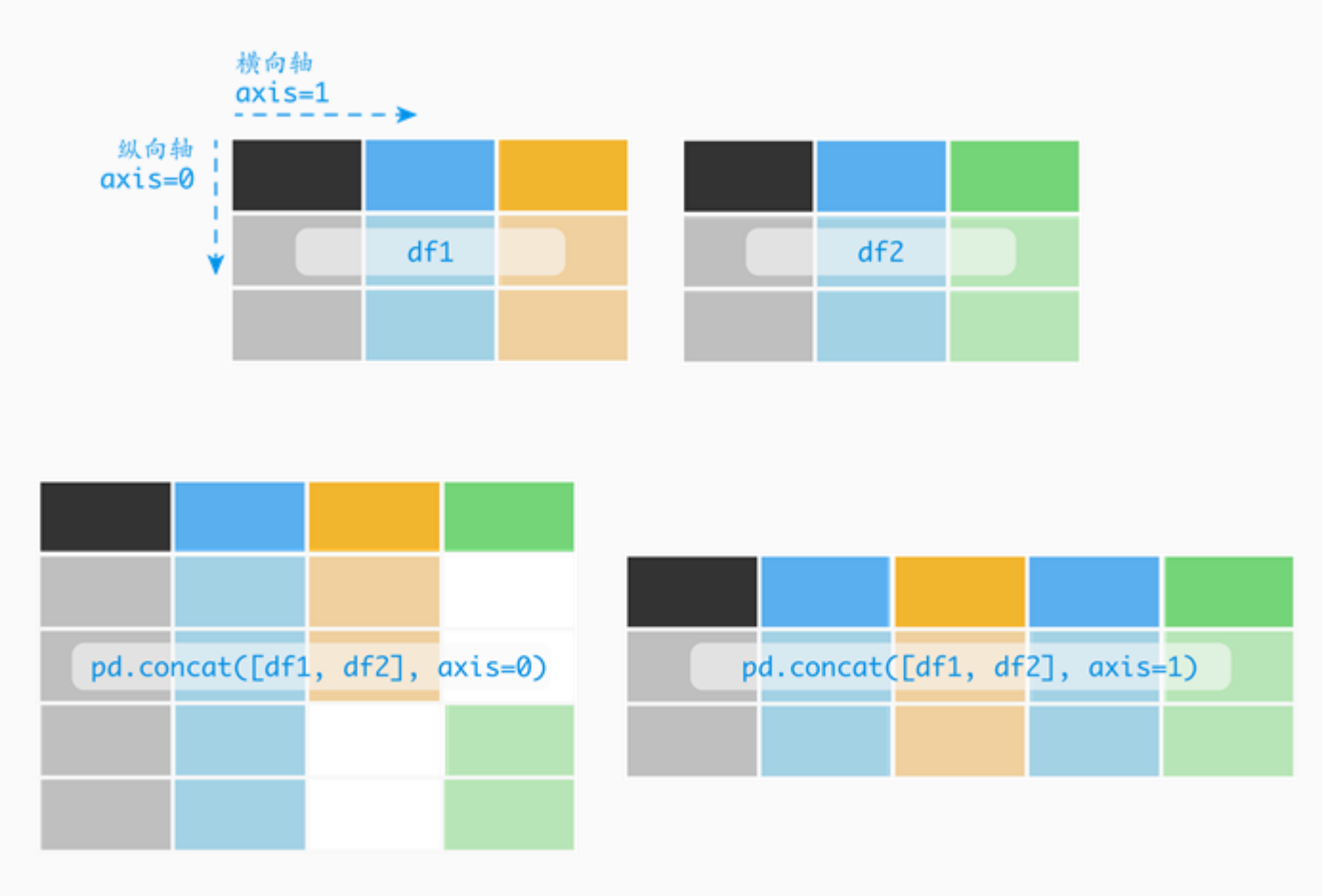
# 3. 机器学习背景知识

- Python基础—pandas

## ●数据融合

pd.concat([df1, df2])

轴向连接多个DataFrame.



# 3. 机器学习背景知识

- Python基础—pandas

## 文件读写

从文件中读取数据（DataFrame）

`pd.read_csv()` | 从CSV文件读取

`pd.read_table()` | 从制表符分隔文件读取，如TSV

`pd.read_excel()` | 从 Excel 文 件 读 取

`pd.read_sql()` | 从 SQL 表 或 数 据 库 读 取

`pd.read_json()` | 从JSON格式的URL或文件读取

`pd.read_clipboard()` | 从剪切板读取

将DataFrame写入文件 `df.to_csv()` | 写入CSV文件 `df.to_excel()` | 写入Excel文件 `df.to_sql()` | 写入SQL表或数据库

`df.to_json()` | 写入JSON格式的文件

`df.to_clipboard()` | 写入剪切板

# 3. 机器学习背景知识

- Python基础—scipy

SciPy是构建在NumPy的基础之上的，它提供了许多的操作NumPy的数组的函数。

SciPy是一款方便、易于使用、专为科学和工程设计的Python工具包，它包括了统计、优化、整合以及线性代数模块、傅里叶变换、信号和图像图例，常微分方差的求解等

| | |
|---|---|
| scipy.cluster | 向量量化 |
| scipy.constants | 数学常量 |
| scipy.fftpack | 快速傅里叶变换 |
| scipy.integrate | 积分 |
| scipy.interpolate | 插值 |
| scipy.io | 数据输入输出 |
| scipy.linalg | 线性代数 |
| scipy.ndimage | N维图像 |
| scipy.odr | 正交距离回归 |
| scipy.optimize | 优化算法 |
| scipy.signal | 信号处理 |
| scipy.sparse | 稀疏矩阵 |
| scipy.spatial | 空间数据结构和算法 |
| scipy.special | 特殊数学函数 |
| scipy.stats | 统计函数 |

# 3. 机器学习背景知识

- Python基础—matplotlib

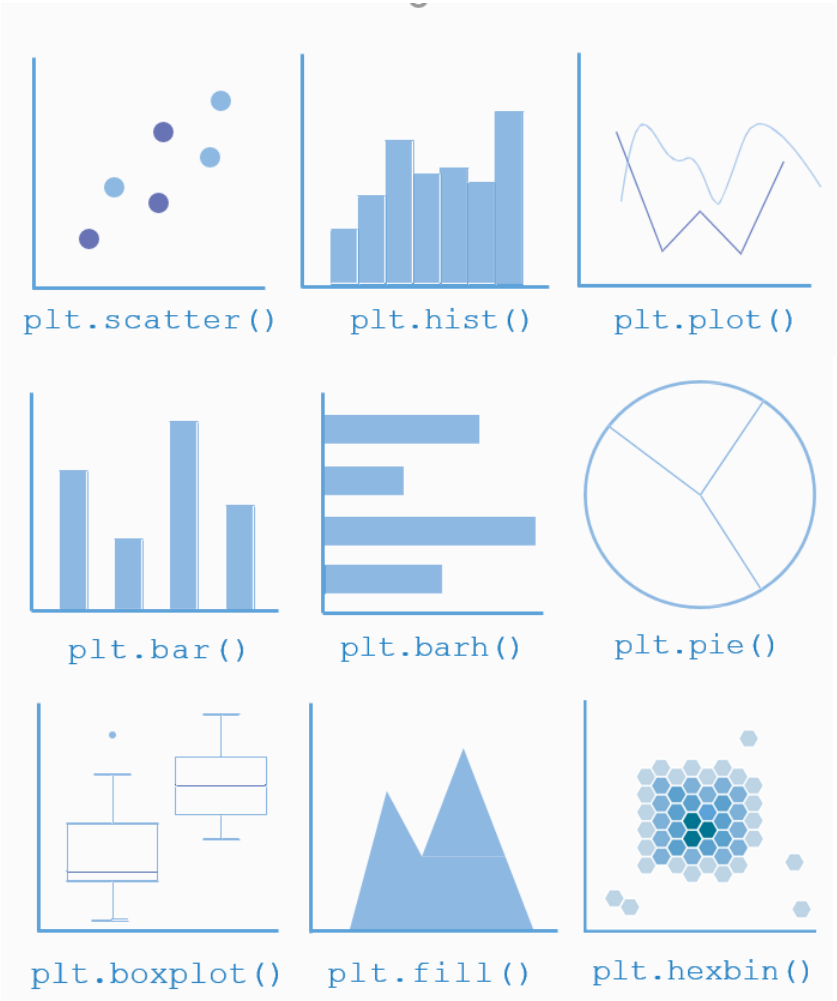
## ●Matplotlib

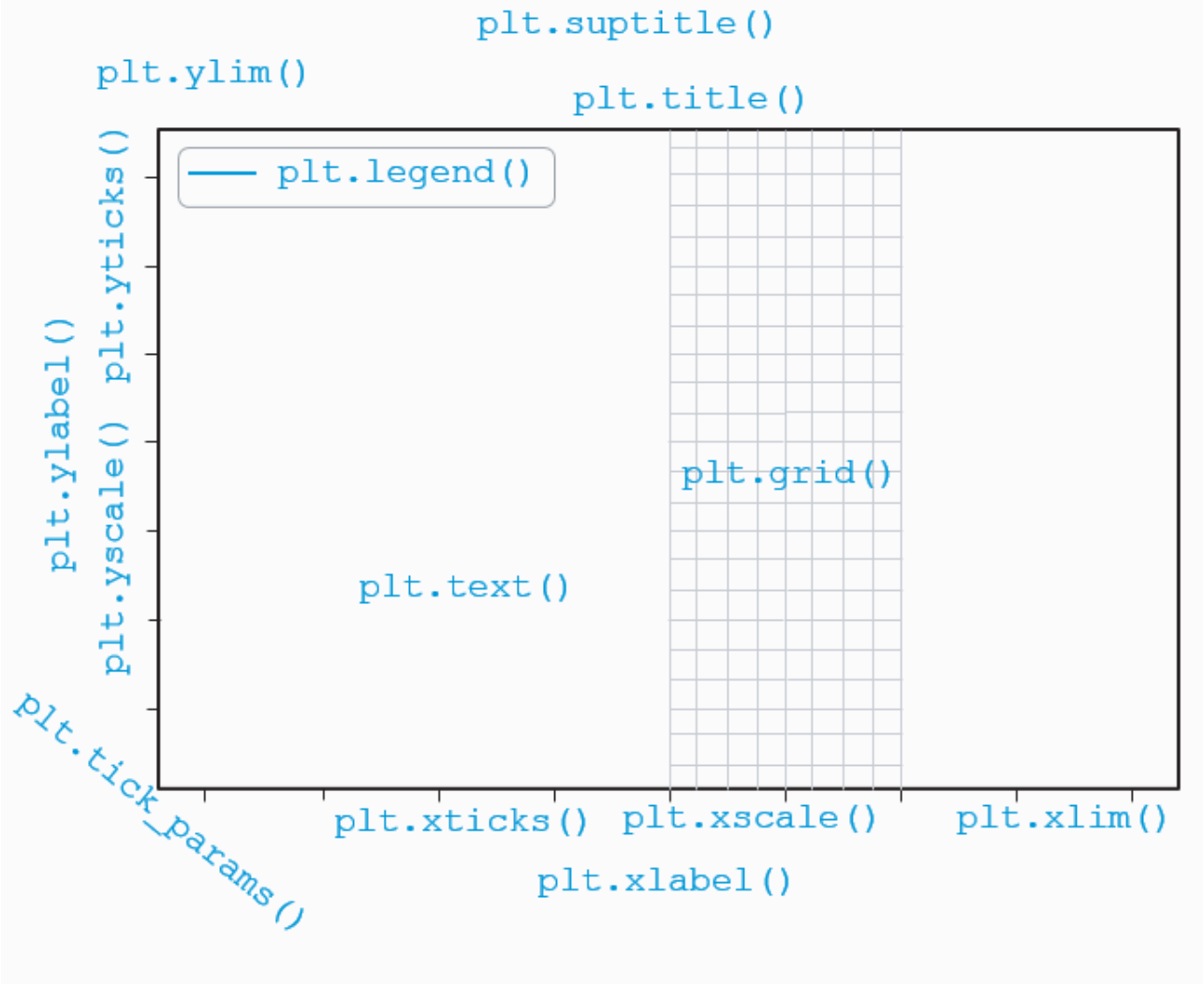
Matplotlib 是一个 Python 的2D绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形 。

通过 Matplotlib，开发者可以仅需要几行代码，便可以生成绘图,直方图，功率谱，条形图，错误图，散点图等。

# 3. 机器学习背景知识

- Python基础—matplotlib



# 4. 机器学习相关概念

- ✓机器学习方法
  - ✓模型
  - ✓损失函数
  - ✓优化算法
  - ✓模型评估指标

# 4. 机器学习相关概念—模型

机器学习首先要考虑使用什么样的模型。

模型的类别，大致有两种：一是概率模型(Probabilistic Model)和非概率模型(Non-Probabilistic Model)。

在监督学习中，概率模型可被表示为$P(y|x)$，非概率模型则为$y = f(x)$。其中，$x$是输入，$y$是输出。

在无监督学习中，概率模型可被表示为$P(z|x)$，非概率模型则为$z = f(x)$。其中，$x$是输入，$z$是输出。

# 4. 机器学习相关概念—模型

决策树、朴素贝叶斯、隐马尔科夫模型、高斯混合模型属于概率模型。

感知机、支持向量机、KNN、AdaBoost、K-means以及神经网络均属于非概率模型。

直观理解：拟合数据的分布函数

假设数据的特征是 $X=(x_1,x_2,...,x_n)$

我们需要找到一个函数能表示所有数据：$y=WX=w_1x_1+w_2x_2+...+w_nx_n$

例如，猫的特征是 X=(10，20，30)，$y=WX=w_1x_1+w_2x_2+w_3x_3=w_1\times10+w_2\times20+w_3\times30=\begin{cases}0\\1\end{cases}$

如何学习这些权重 $W=(w_1,w_2,...,w_n)$ ？

度量预测的 $y$ 和真实的 $\bar{y}$ 之间的差异，即计算loss值，通过使得loss最小化使得差异最小化

# 4. 机器学习相关概念—损失函数loss

## 1. 0-1损失函数(0-1 Loss Function)

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

## 2. 平方损失函数(Quadratic Loss Function)

$$L(Y, f(X)) = (Y - f(X))^2$$

## 3. 绝对损失函数(Absolute Loss Function)

$$L(Y, f(X)) = |Y - f(X)|$$

## 4. 对数损失函数(Logarithmic Loss Function)

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

# 4. 机器学习相关概念—损失函数

根据上述损失函数模型，我们可知，损失函数值越小，模型性能越好。给定一个数据集，我们将训练数据集的平均损失称为经验风险。基于经验风险最小化原则，可构建全局损失函数求解最优化问题：

$$\min_{f} \frac{1}{N} \sum_{n=1}^{N} \mathrm{L}\left(y_n, f(x_n)\right)$$

当样本数量足够大时，根据大数定理，经验风险会近似于模型的期望风险。此时，经验风险最小化能确保有好的学习性能。然而，当样本数量不足时，单单利用经验风险最小化可能会导致 “过拟合”的问题。为此，我们再原有基础上加上用于控制模型复杂度的正则项(Regularizer)，得到结构最小化准则。

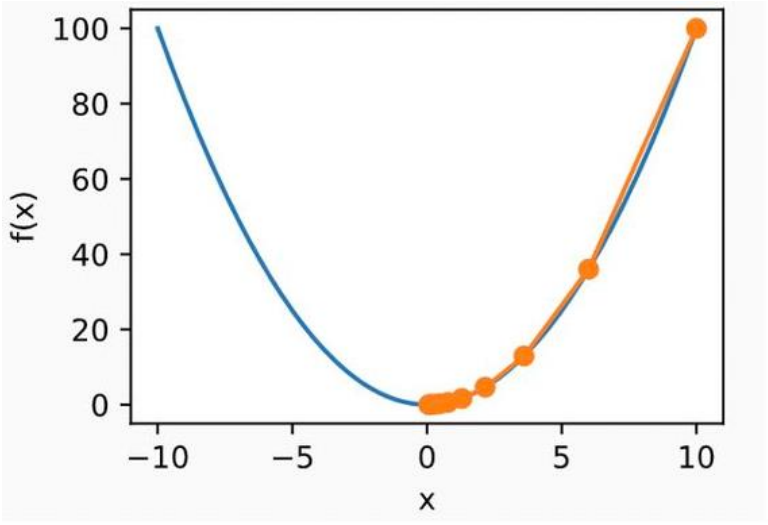
# 4. 机器学习相关概念—优化算法

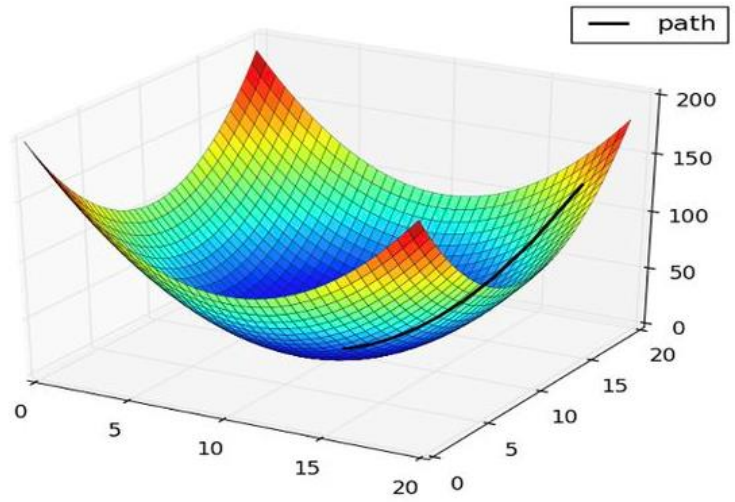
如何使得loss最小化？

算法指的是模型学习中的具体计算方法。一般来说，基于参数模型构建的统计学习问题都为最优化问题，它们都具有显式的解析解。

现有的优化方法主要有：梯度下降法、牛顿法、拟牛顿法、ADAM等等。

梯度下降法：一阶导数为0的点，不断下降寻找极小值



# 4. 机器学习相关概念—模型评估

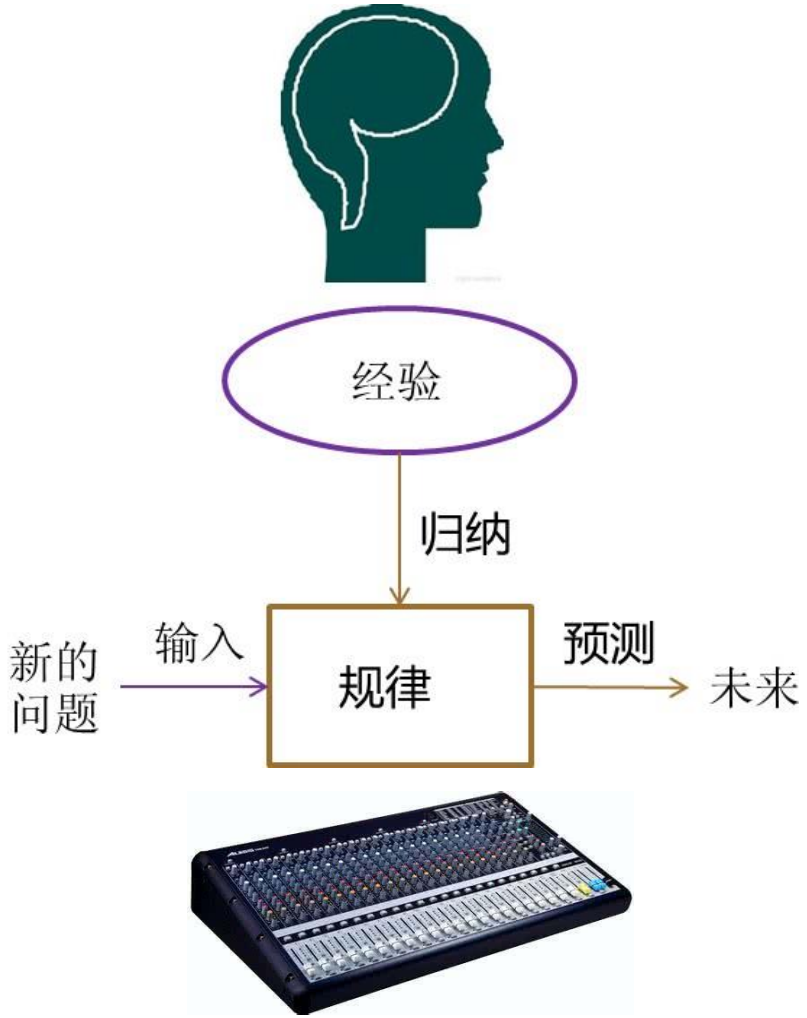
当损失函数给定时，我们将基于模型训练数据的误差(Training Error)和测试数据的误差(Testing Error)作为模型评估的标准。

测试误差的具体定义为：$E_{test} = \frac{1}{N'}\sum_{n=1}^{N'} \mathrm{L}\left(y_n, \hat{f}(x_n)\right)$

其中，$N'$为测试数据数量，$\mathrm{L}(y_n, \hat{f}(x_n))$是损失函数，$y_n$代表真实标签，$\hat{f}(x_n)$代表预测标签。

一般来说，若我们模型学习的效果好，则训练误差和测试误差接近一致。

# 5. 机器学习的开发流程

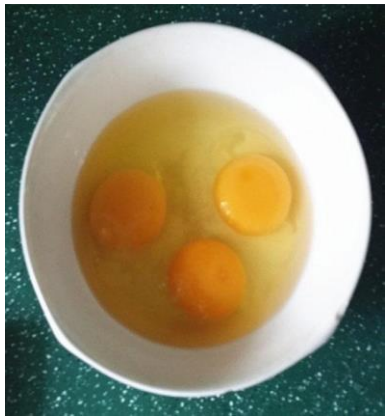
# 5. 机器学习的开发流程

数据搜集 → 数据清洗 → 特征工程 → 数据建模









# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法
  - 有监督学习
  - K-NN处理分类问题
  - K-NN实验参数设置
- 6.2 实验任务与要求

# 6. 机器学习算法--KNN

## • 6.1 K-近邻（KNN）算法—有监督学习

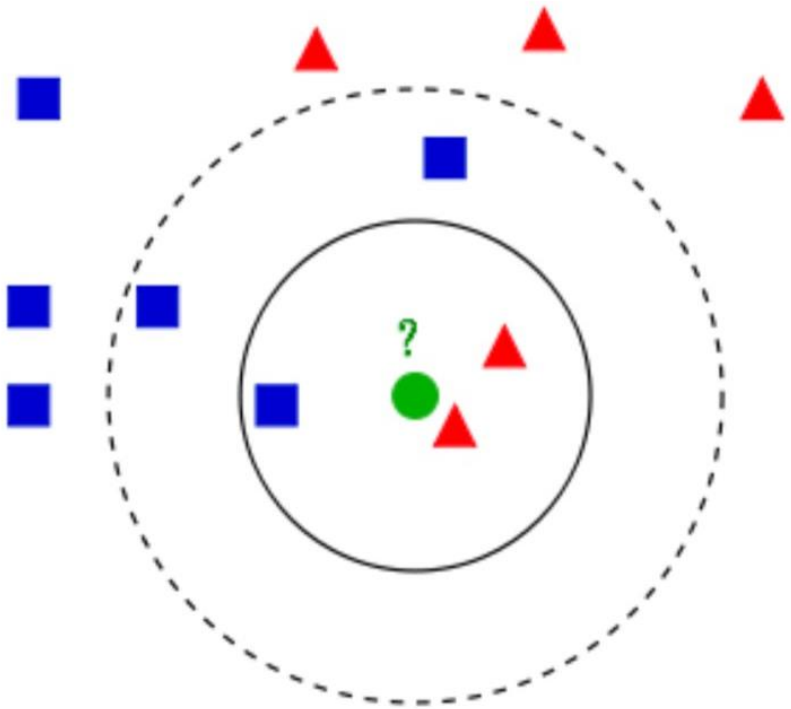
- $k$-NN是有监督的机器学习模型

- 有监督学习的基本步骤：上课—考试
  - 给出带标签的训练数据
  - 用训练数据训练模型至一定程度
  - 用训练好的模型预测不带标签的数据的标签

- 常见的有监督学习问题：
  - 分类问题：预测离散值的问题（如预测明天是否会下雨）
  - 回归问题：预测连续值的问题（如预测明天气温是多少度）

# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN处理分类问题




半径大小 表示 K值大小

k-nearest neighbours **classifier**:

$$f(q) = maj\left(g\left(\Phi_{X,k}(q)\right)\right)$$

其中:

$\Phi_{X,k}(q)$: 返回训练集$X$中距离$q$最近的k个样本

$g(\cdot)$: 返回（训练）样本的标签

$maj(\cdot)$: 返回众数

# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN处理分类问题：例子

给定文本的情感分类任务：

输入：文本

输出：类标签

分类：多数投票原则

| Document number | The sentence words | emotion |
|---|---|---|
| train 1 | I buy an apple phone | happy |
| train 2 | I eat the big apple | happy |
| train 3 | The apple products are too expensive | sadnesss |
| test 1 | My friend has an apple | ? |

# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN处理分类问题：步骤

| Document number | The sentence words | emotion |
|---|---|---|
| train 1 | I buy an apple phone | happy |
| train 2 | I eat the big apple | happy |
| train 3 | The apple products are too expensive | sadnesss |
| test 1 | My friend has an apple | ? |

1. 处理成one-hot矩阵

| Document number | I | buy | an | apple | ... | friend | has | emotion |
|---|---|---|---|---|---|---|---|---|
| train 1 | 1 | 1 | 1 | 1 | ... | 0 | 0 | happy |
| train 2 | 1 | 0 | 0 | 1 | ... | 0 | 0 | happy |
| train 3 | 0 | 0 | 0 | 1 | ... | 0 | 0 | sadness |
| test 1 | 0 | 0 | 1 | 1 | ... | 1 | 1 | ? |

# 6. 机器学习算法--KNN

## • 6.1 K-近邻（KNN）算法—KNN处理分类问题：步骤

2. 相似度计算：计算test1与每个train的距离

欧氏距离：

$$d(train1, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \cdots + (0-1)^2} = \sqrt{6};$$

$$d(train2, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \cdots + (0-1)^2} = \sqrt{8};$$

$$d(train3, test1) = \sqrt{(0-0)^2 + (0-0)^2 + \cdots + (0-1)^2} = \sqrt{9};$$

（也可以使用其他距离度量方式）

| Document number | I | buy | an | apple | ... | friend | has | emotion |
|---|---|---|---|---|---|---|---|---|
| train 1 | 1 | 1 | 1 | 1 | ... | 0 | 0 | happy |
| train 2 | 1 | 0 | 0 | 1 | ... | 0 | 0 | happy |
| train 3 | 0 | 0 | 0 | 1 | ... | 0 | 0 | sadness |
| test 1 | 0 | 0 | 1 | 1 | ... | 1 | 1 | ? |

3. 类别计算：最相似的k个样本之标签的众数

若k=1，test1的标签即为train1的标签happy；

若k=3，test1的标签为train1,train2,train3的标签中数量较多的，即为happy。

# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN参数设置
- 采用不同的距离度量方式（见下一页）
- 通过验证集对参数（k值）进行调优
  - 如果k值取的过大，学习的参考样本更多，会引入更多的噪音，所以可能存在欠拟合的情况；
  - 如果k值取的过小，参考样本少，容易出现过拟合的情况
  - 关于k的经验公式：一般取$k=\sqrt{N}$，N为训练集实例个数，大家可以尝试一下
- 权重归一化

| Name | Formula | Explain |
|---|---|---|
| Standard score | $X' = \frac{X-\mu}{\sigma}$ | $\mu$ is the mean and $\sigma$ is the standard deviation |
| Feature scaling | $X' = \frac{X-X_{min}}{X_{max}-X_{min}}$ | $X_{min}$ is the min value and $X_{max}$ is the max value |

# 6. 机器学习算法--KNN

## • 6.1 K-近邻（KNN）算法—不同的度量公式

距离公式：

$L_p$距离(所有距离的总公式)：

$$L_p(x_i, x_j)=\left\{\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^p\right\}^{\frac{1}{p}}$$

$p=1$：曼哈顿距离；

$p=2$：欧氏距离，最常见。

**例 3.1** 已知二维空间的 3 个点 $x_1=(1,1)^{\mathrm{T}}$, $x_2=(5,1)^{\mathrm{T}}$, $x_3=(4,4)^{\mathrm{T}}$，试求在 $p$ 取不同值时，$L_p$ 距离下 $x_1$ 的最近邻点。

**解** 因为 $x_1$ 和 $x_2$ 只有第一维的值不同，所以 $p$ 为任何值时，$L_p(x_1,x_2)=4$。而

$$L_1(x_1,x_3)=6,\quad L_2(x_1,x_3)=4.24,\quad L_3(x_1,x_3)=3.78,\quad L_4(x_1,x_3)=3.57$$

于是得到：$p$ 等于 1 或 2 时，$x_2$ 是 $x_1$ 的最近邻点；$p$ 大于等于 3 时，$x_3$ 是 $x_1$ 的最近邻点。■

余弦相似度：

$$\cos\left(\vec{A},\vec{B}\right)=\frac{\vec{A}\cdot\vec{B}}{\left|\vec{A}\right|\left|\vec{B}\right|}$$，其中$\vec{A}$和$\vec{B}$表示两个文本特征向量；

余弦值作为衡量两个个体间差异的大小的度量

为正且值越大，表示两个文本差距越小，为负代表差距越大，请大家自行脑补两个向量余弦值

# 6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN算法效率
- 假设训练集有N个样本，测试集有M个样本，每个样本是一个V维的向量。

- 如果使用线性搜索的话，那么*k*-NN的时间花销就是O(N*M*V)。

- 改善：KD树

# 6. 机器学习算法--KNN

- 6.2 实验任务与要求
- 在给定文本数据集完成文本情感分类训练，在测试集完成测试，计算准确率。
- 要求
  - 文本的特征可以使用TF或TF-IDF，对TF均使用拉普拉斯平滑技巧（可以使用sklearn库提取特征）
  - 利用KNN完成对测试集的分类，并计算准确率
  - 需要提交代码
  - 压缩包：学号_姓名_作业编号.zip，如 20331234_张三_实验7.zip
  - 截止日期：2022.5.25　23:59
  - 机器学习共有两次作业，本次作业和下周作业一起提交，不要单独提交

# Thanks!