

# 法律声明

■ 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，北风网和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

■ 课程详情请咨询

◆ 微信公众号：北风教育

◆ 官方网址：<http://www.ibeifeng.com/>



# 人工智能之机器学习

## 机器学习概述

主讲人：Gerry

上海育创网络科技有限公司

# 课程要求

## ■ 课上课下 “九字” 真言

- ◆ 认真听，善摘录，勤思考
- ◆ **多温故，乐实践**，再发散

## ■ 四不原则

- ◆ **不懒散惰性，不迟到早退**
- ◆ **不请假旷课，不拖延作业**

## ■ 一点注意事项

- ◆ 违反 “四不原则”，不包就业和推荐就业

# ▶ 课程介绍

## 一 课程体系

# 1

### 机器学习入门

介绍什么是机器学习，机器学习的内容以及应用场景

# 2

### 模型原理讲解

针对具体的算法，进行算法原理的讲解，包括：回归、决策树、SVM、聚类等算法

# 3

### 实战演练

大量的实例来应用具体的算法，掌握具体的算法的应用

# ▶ 机器学习概述



- 01 ▶ 机器学习的定义
- 02 ▶ 机器学习、人工智能和深度学习的关系
- 03 ▶ 机器学习基本概念和常用的应用场景
- 04 ▶ 机器学习、数据分析、数据挖掘的区别与联系
- 05 ▶ 机器学习分类
- 06 ▶ 机器学习数据处理流程

# Machine Learning



What society thinks I do



What my friend thinks I do



What my parents thinks I do

SVM:

$$\min_{\mathbf{w}, \xi, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right\}$$

subject to (for any  $i = 1, \dots, n$ )

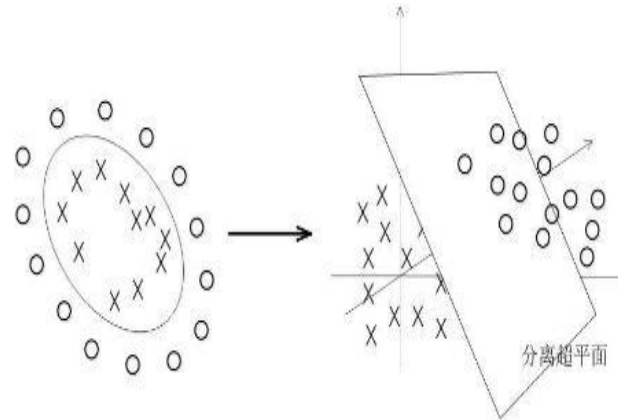
$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

LR:

$$\min_{\theta} \sum_{i=1}^M -\log p(y^{(i)} | \mathbf{x}^{(i)}; \theta) + \beta \|\theta\|_1.$$

$$P_w(y|x) = \frac{\exp w^T \Phi(x, y)}{\sum_{y' \in \text{GEN}(x)} \exp w^T \Phi(x, y')}.$$

What other programmers  
thinks I do



What I thinks I do

```
In [1]: from sklearn import svm

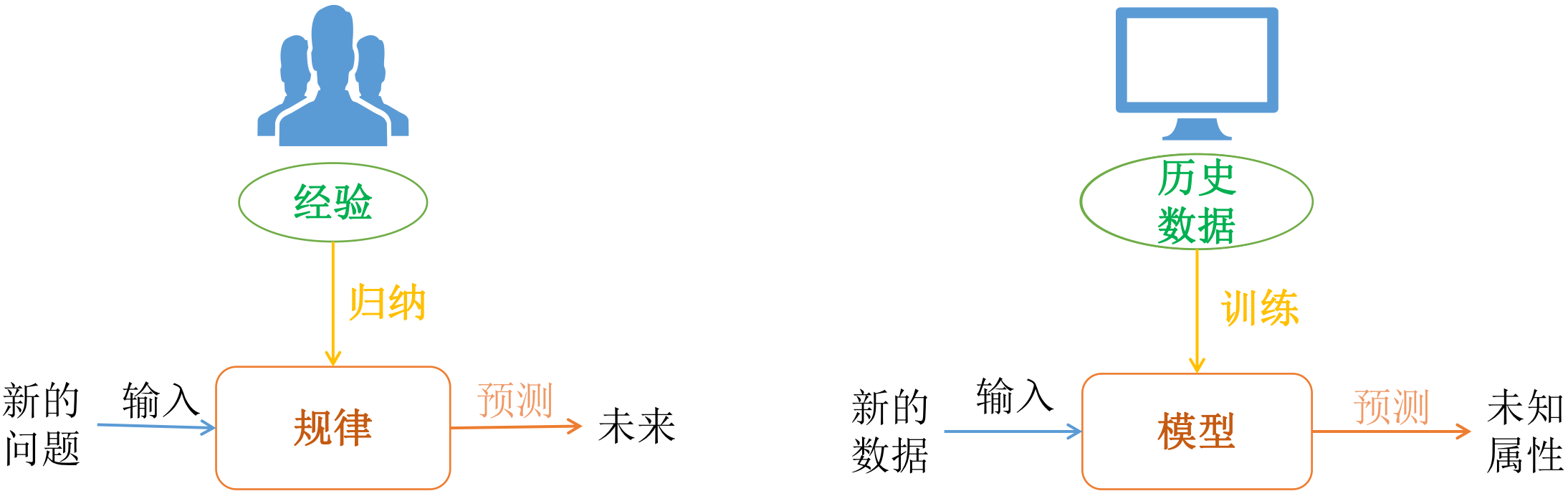
In [ ]:
```

What I really do

# 机器学习定义

- **Machine Learning**(ML) is a scientific discipline that deals with the construction and study of algorithms that can learn from data.
- 机器学习是一门从数据中研究算法的科学学科。
- 机器学习直白来讲，是根据已有的数据，进行算法选择，并基于算法和数据构建模型，最终对未来进行预测

# 机器学习理性认识





# 机器学习理性认识

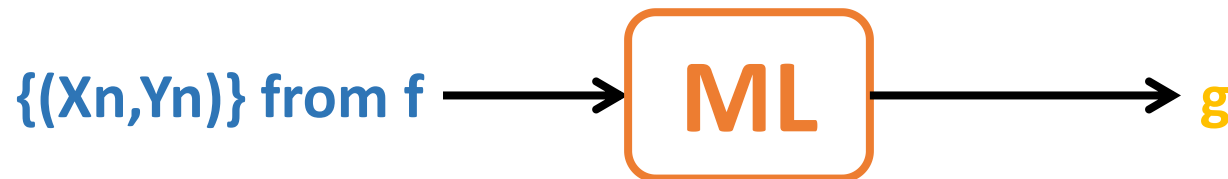
## ■ 基本概念

- ◆ 输入:  $x \in \mathcal{X}$  (属性值)
- ◆ 输出:  $y \in Y$  (目标值)
- ◆ 获得一个目标函数(target function):

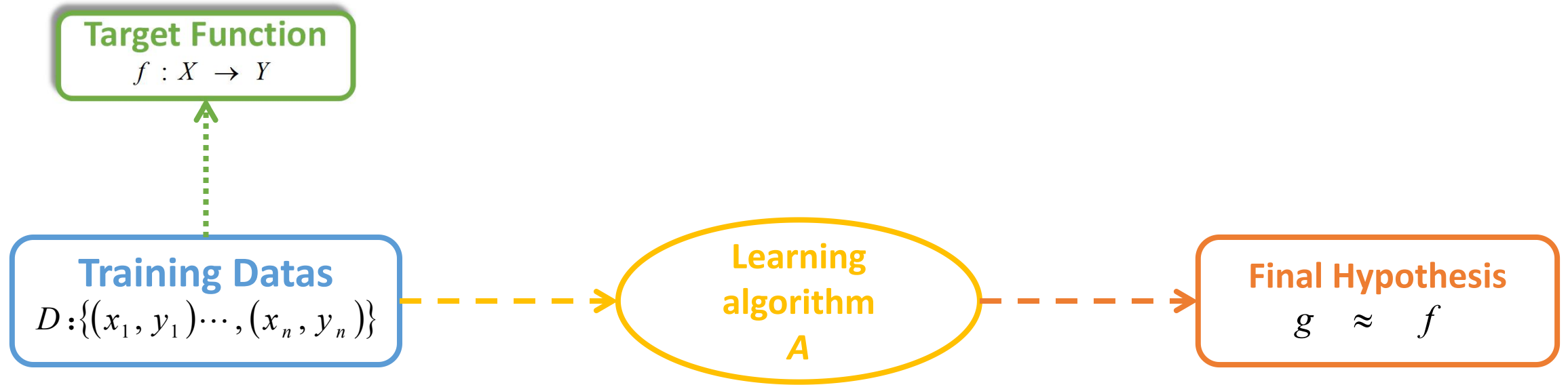
$$f : X \rightarrow Y \text{ (理想的公式)}$$

- ◆ 输入数据:  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  (历史记录信息)
- ◆ 最终具有最优性能的假设公式:

$$g : X \rightarrow Y \text{ (学习得到的最终公式)}$$



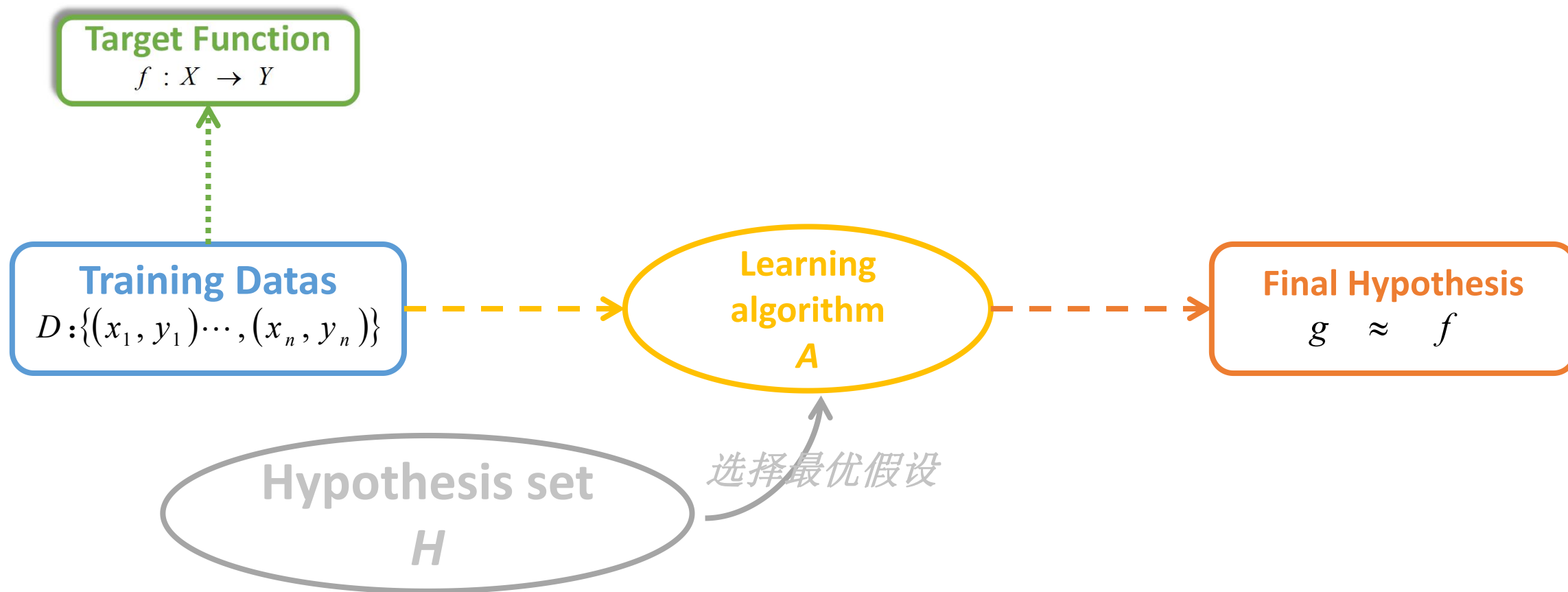
# 机器学习理性认识



- 目标函数 $f$ 未知（无法得到）
- 假设函数 $g$ 类似函数 $f$ ，但是可能和函数 $f$ 不同

机器学习中是无法找到一个完美的函数 $f$

# 机器学习理性认识



## ■ 机器学习

从数据中获得一个假设的函数 $g$ ，使其非常接近目标函数 $f$ 的效果。

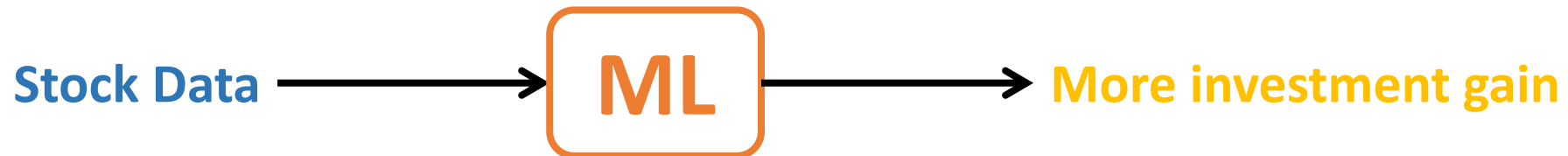
# 机器学习概念

- A program can be said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$  , If its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .
- 对于某给定的任务 $T$ ，在合理的性能度量方案 $P$ 的前提下，某计算机程序可以自主学习任务 $T$ 的经验 $E$ ；随着提供合适、优质、大量的经验 $E$ ，该程序对于任务 $T$ 的性能逐步提高。
- 其中重要的机器学习对象：
- 任务Task  $T$ ，一个或多个、经验Experience  $E$ 、度量性能Performance  $P$
- 即：随着任务的不断执行，经验的累积会带来计算机性能的提升。
- 美国卡内基梅隆大学（Carnegie Mellon University）机器学习研究领域的著名教授Tom Mitchell对机器学习的经典定义

# 机器学习概念



- 算法(T)：根据业务需要和数据特征选择的相关算法，也就是一个数学公式
- 模型(E)：基于数据和算法构建出来的模型
- 评估/测试(P)：对模型进行评估的策略



# 机器学习概念性含义

- 机器学习是人工智能的一个分支。我们使用计算机设计一个系统，使它能够根据提供的训练数据按照一定的方式来学习；随着训练次数的增加，该系统可以在性能上不断学习和改进；通过参数优化的学习模型，能够用于预测相关问题的输出。

# 机器学习概念

■ 向量： 1.2   2.1   3.2   4.2   1.2   3.2

维度

■ 矩阵：

|     |     |     |     |      |
|-----|-----|-----|-----|------|
| 1.2 | 2.1 | 3.2 | 4.2 | 1.1  |
| 0   | -1  | 2.2 | 0.2 | -2.3 |
| 23  | 12  | 10  | 15  | 18   |

■ 标量：

|     |     |      |     |      |   |
|-----|-----|------|-----|------|---|
| 1.2 | 2.1 | 3.2  | 4.5 | 1.2  | 1 |
| 0   | 0.1 | -0.2 | 3   | -1.2 | 0 |
| 23  | 21  | 20   | 15  | 19   | 1 |

向量

标量

# 机器学习概念

- 拟合：构建的算法符合给定数据的特征
- $x^{(i)}$ ：表示第*i*个样本的x向量
- $x_i$ ：x向量的第*i*维度的值
- 鲁棒性：也就是健壮性、稳健性、强健性,是系统的健壮性；当存在异常数据的时候，算法也会拟合数据
- 过拟合：算法太符合样本数据的特征，对于实际生产中的数据特征无法拟合
- 欠拟合：算法不太符合样本的数据特征



# 机器学习之常见应用框架

## ■ scikit-learn(Python)

◆ <http://scikit-learn.org/stable/>



## ■ Mahout(Hadoop生态圈基于MapReduce)

◆ <http://mahout.apache.org/>



## ■ Spark MLlib

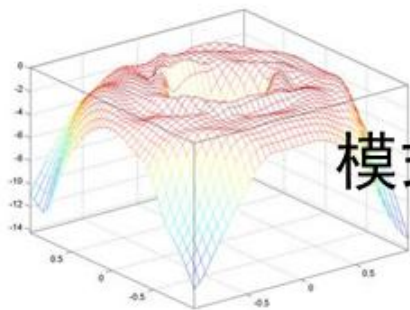
◆ <http://spark.apache.org/>





- **个性化推荐**：个性化指的是根据各种因素来改变用户体验和呈现给用户内容，这些因素可能包含用户的行为数据和外部因素；推荐常指系统向用户呈现一个用户可能感兴趣的物品列表。
- **精准营销**：从用户群众中找出特定的要求的营销对象。
- **客户细分**：试图将用户群体分为不同的组，根据给定的用户特征进行客户分组。
- **预测建模及分析**：根据已有的数据进行建模，并使用得到的模型预测未来。

# 机器学习之商业场景



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



# 机器学习、数据分析、数据挖掘区别与联系

- **数据分析**：数据分析是指用适当的统计分析方法对收集的大量数据进行分析，并提取有用的信息，以及形成结论，从而对数据进行详细的研究和概括过程。在实际工作中，数据分析可帮助人们做出判断；数据分析一般而言可以分为统计分析、探索性数据分析和验证性数据分析三大类。
- **数据挖掘**：一般指从大量的数据中通过算法搜索隐藏于其中的信息的过程。通常通过统计、检索、机器学习、模式匹配等诸多方法来实现这个过程。
- **机器学习**：是数据分析和数据挖掘的一种比较常用、比较好的手段。

## ■ 有监督学习

- ◆ 用已知某种或某些特性的样本作为训练集，以建立一个数学模型，再用已建立的模型来预测未知样本，此种方法被称为有监督学习，是最常用的一种机器学习方法。是从**标签化**训练数据集中推断出模型的机器学习任务。

## ■ 无监督学习

- ◆ 与监督学习相比，无监督学习的训练集中没有人为的标注的结果，在非监督的学习过程中，数据并不被特别标识，学习模型是为了推断出数据的一些内在结构。

## ■ 半监督学习

- ◆ 考虑如何利用少量的标注样本和大量的未标注样本进行训练和分类的问题，是有监督学习和无监督学习的结合



# 有监督学习

- **判别式模型**(Discriminative Model)：直接对条件概率 $p(y|x)$ 进行建模，常见判别模型有：线性回归、决策树、支持向量机SVM、k近邻、神经网络等；
- **生成式模型**(Generative Model)：对联合分布概率 $p(x,y)$ 进行建模，常见生成式模型有：隐马尔可夫模型HMM、朴素贝叶斯模型、高斯混合模型GMM、LDA等；
- 生成式模型更普适；判别式模型更直接，目标性更强
- 生成式模型关注数据是如何产生的，寻找的是数据分布模型；判别式模型关注的数据的差异性，寻找的是分类面
- 由生成式模型可以产生判别式模型，但是由判别式模式没法形成生成式模型

# 无监督学习

- 无监督学习试图学习或者提取数据背后的数据特征，或者从数据中抽取重要的特征信息，常见的算法有聚类、降维、文本处理(特征抽取)等。
- 无监督学习一般是作为有监督学习的前期数据处理，功能是从原始数据中抽取必要的标签信息。



# 半监督学习(SSL)

- 主要考虑如何利用少量的标注样本和大量的未标注样本进行训练和分类的问题。  
半监督学习对于减少标注代价，提高学习机器性能具有非常重大的实际意义。
- SSL的成立依赖于模型假设，主要分为三大类：平滑假设、聚类假设、流行假设；其中流行假设更具有普片性。
- SSL类型的算法主要分为四大类：半监督分类、半监督回归、半监督聚类、半监督降维。
- 缺点：抗干扰能力弱，仅适合于实验室环境，其现实意义还没有体现出来；未来的发展主要是聚焦于新模型假设的产生。

# 机器学习分类2

## ■ 分类

- ◆ 通过分类模型，将样本数据集中的样本映射到某个给定的类别中

## ■ 聚类

- ◆ 通过聚类模型，将样本数据集中的样本分为几个类别，属于同一类别的样本相似性比较大

## ■ 回归

- ◆ 反映了样本数据集中样本的属性值的特性，通过函数表达样本映射的关系来发现属性值之间的依赖关系

## ■ 关联规则

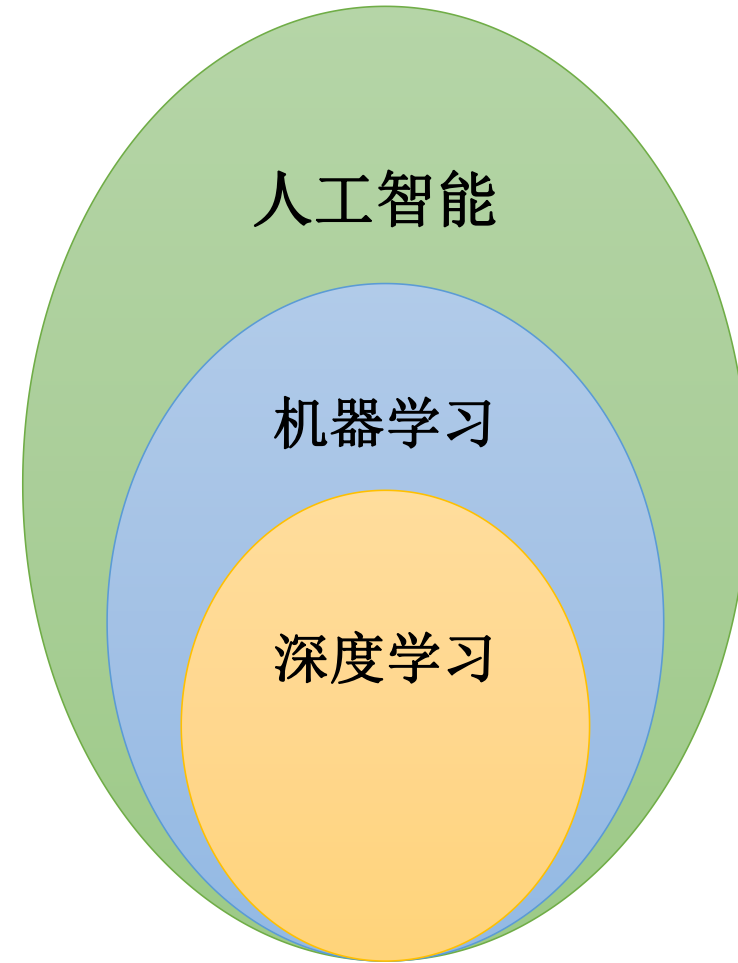
- ◆ 获取隐藏在数据项之间的关联或相互关系，即可以根据一个数据项的出现推导出其他数据项的出现频率。

# 机器学习算法Top10

| 算法名称       | 算法描述  |
|------------|---|
| C4.5       | 分类决策树算法，决策树的核心算法，ID3算法的改进算法。  |
| CART       | 分类与回归树(Classification and Regression Trees)                                     |
| kNN        | K近邻分类算法；如果一个样本在特征空间中的k个最相似的样本中大多数属于某一个类别，那么该样本也属于该类别                            |
| NaiveBayes | 贝叶斯分类模型；该模型比较适合属性相关性比较小的时候，如果属性相关性比较大的时候，决策树模型比贝叶斯分类模型效果好(原因：贝叶斯模型假设属性之间是互不影响的) |
| SVM        | 支持向量机，一种有监督学习的统计学习方法，广泛应用于统计分类和回归分析中。   |
| EM         | 最大期望算法，常用于机器学习和计算机视觉中的数据集聚领域  |
| Apriori    | 关联规则挖掘算法  |
| K-Means    | 聚类算法，功能是将n个对象根据属性特征分为k个分割( $k < n$ ); 属于无监督学习                                   |
| PageRank   | Google搜索重要算法之一  |
| AdaBoost   | 迭代算法；利用多个分类器进行数据分类  |

# 机器学习、人工智能和深度学习的关系

- 深度学习是机器学习的子类；  
深度学习是基于传统的神经网络算法发展到多隐层的一种算法体现。
- 机器学习是人工智能的一个子类；



计算：云计算

推理：专家系统

智慧：机器学习

灵敏：事件驱动

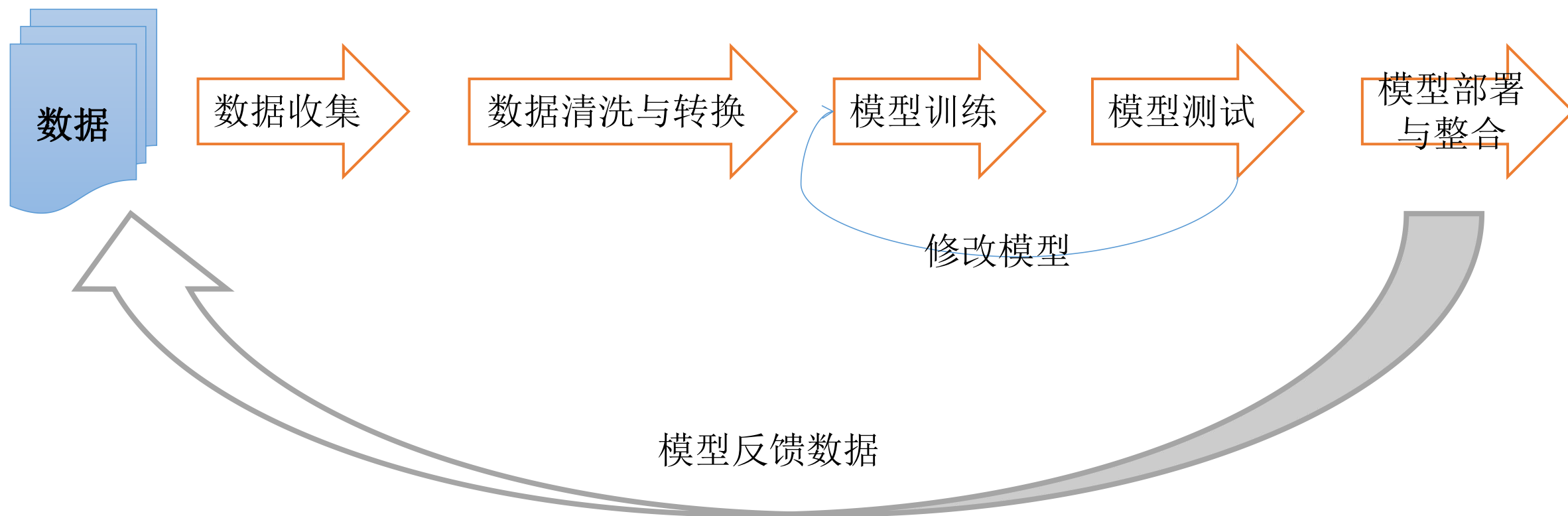
知识：数据仓库

检索：搜索引擎

# 机器学习开发流程

- 数据收集
- 数据预处理
- 特征提取
- 模型构建
- 模型测试评估
- 投入使用(模型部署与整合)
- 迭代优化

# 机器学习开发流程



# 机器学习一般流程 ——生活案例

数据收集



数据清洗



特征工程



数据建模





# 数据收集与存储

## ■ 数据来源：

- ◆ 用户访问行为数据
- ◆ 业务数据
- ◆ 外部第三方数据

## ■ 数据存储：

- ◆ 需要存储的数据：原始数据、预处理后数据、模型结果
- ◆ 存储设施：mysql、HDFS、HBase、Solr、Elasticsearch、Kafka、Redis等

## ■ 数据收集方式：

- ◆ Flume & Kafka

# 机器学习可用公开数据集

- 在实际工作中，我们可以使用业务数据进行机器学习开发，但是在学习过程中，没有业务数据，此时可以使用公开的数据集进行开发，常用数据集如下：

- ◆ <http://archive.ics.uci.edu/ml/datasets.html>
- ◆ <https://aws.amazon.com/cn/public-datasets/>
- ◆ <https://www.kaggle.com/competitions>
- ◆ <http://www.kdnuggets.com/datasets/index.html>
- ◆ [http://www.sogou.com/labs/resource/list\\_pingce.php](http://www.sogou.com/labs/resource/list_pingce.php)
- ◆ <https://tianchi.aliyun.com/datalab/index.htm>
- ◆ <http://www.pkbigdata.com/common/cmptIndex.html>

# 数据清洗和转换

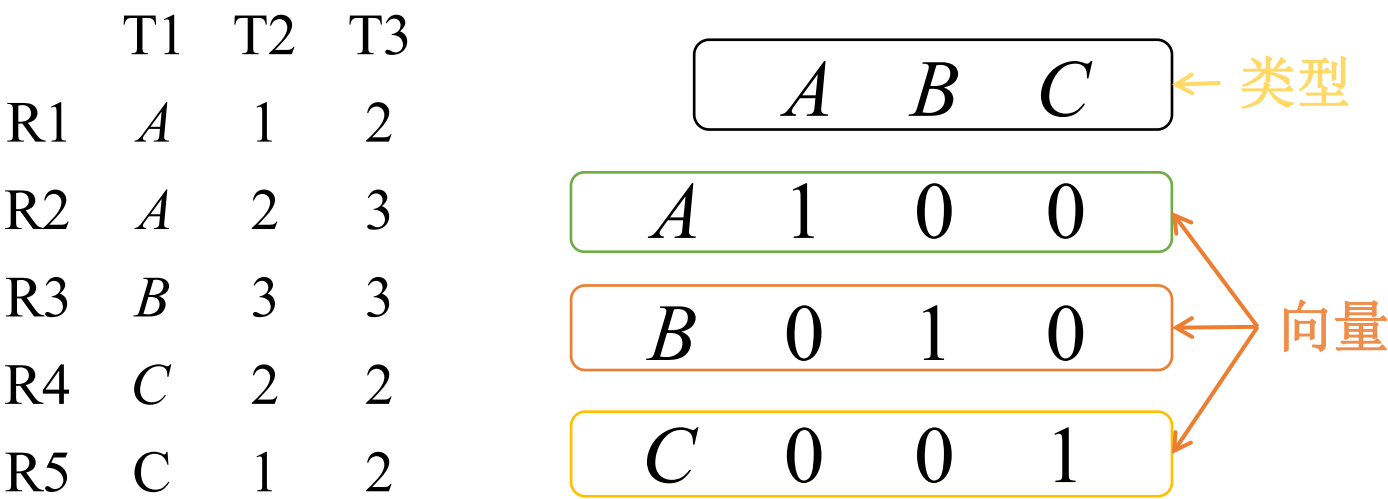
- 实际生产环境中机器学习比较耗时的一部分
- 大部分的机器学习模型所处理的都是特征，特征通常是输入变量所对应的可用于模型的数值表示
- 大部分情况下，收集得到的数据需要经过预处理后才能够为算法所使用，预处理的操作主要包括以下几个部分：
  - ◆ 数据过滤
  - ◆ 处理数据缺失
  - ◆ 处理可能的异常、错误或者异常值
  - ◆ 合并多个数据源数据
  - ◆ 数据汇总

# 数据清洗和转换

- 对数据进行初步的预处理，需要将其转换为一种适合机器学习模型的表示形式，对许多模型类型来说，这种表示就是包含数值数据的向量或者矩阵
  - ◆ 将类别数据编码成为对应的数值表示(一般使用1-of-k方法)-dummy
  - ◆ 从文本数据中提取有用的数据(一般使用词袋法或者TF-IDF)
  - ◆ 处理图像或者音频数据(像素、声波、音频、振幅等<傅里叶变换>)
  - ◆ 数值数据转换为类别数据以减少变量的值，比如年龄分段
  - ◆ 对数值数据进行转换，比如对数转换
  - ◆ 对特征进行正则化、标准化，以保证同一模型的不同输入变量的值域相同
  - ◆ 对现有变量进行组合或转换以生成新特征，比如平均数 (做虚拟变量)不断尝试

# 类型特征转换之1-of-k

- 功能：将非数值型的特征值转换为数值型的数据
- 描述：假设变量的取值有 k 个，如果对这些值用 1 到 k 编序，则可用维度为 k 的向量来表示一个变量的值。在这样的向量里，该取值所对应的序号所在的元素为1，其他元素均为0.



|    |      |      |      |    |    |
|----|------|------|------|----|----|
|    | T1特征 |      |      |    |    |
|    | T1-A | T1-B | T1-C | T2 | T3 |
| R1 | 1    | 0    | 0    | 1  | 2  |
| R2 | 1    | 0    | 0    | 2  | 3  |
| R3 | 0    | 1    | 0    | 3  | 3  |
| R4 | 0    | 0    | 1    | 2  | 2  |
| R5 | 0    | 0    | 1    | 1  | 2  |

- **词袋法**：将文本当作一个无序的数据集合，文本特征可以采用文本中的词条T进行体现，那么文本中出现的所有词条及其出现的次数就可以体现文档的特征
- **TF-IDF**：词条的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降；也就是说词条在文本中出现的次数越多，表示该词条对该文本的重要性越高，词条在所有文本中出现的次数越少，说明这个词条对文本的重要性越高。TF(词频)指某个词条在文本中出现的次数，一般会将其进行归一化处理(该词条数量/该文档中所有词条数量)；IDF(逆向文件频率)指一个词条重要性的度量，一般计算方式为总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。TF-IDF实际上是： **$TF * IDF$**

# 文本数据抽取

- 文档1内容：A(2)、B(1)、C(3)、D(9)、E(1)
- 文档2内容：A(1)、B(5)、C(2)、D(10)

|     | <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> |
|-----|----------|----------|----------|----------|----------|
| 文档1 | 2        | 1        | 3        | 9        | 1        |
| 文档2 | 1        | 5        | 2        | 10       | 0        |

$$TF(A | \text{文档1}) = 1/8$$

$$TF(B | \text{文档1}) = 1/16$$

.....

$$TF(E | \text{文档2}) = 0/18 = 0$$

$$IDF(A) = IDF(B) = IDF(C) = IDF(D) = 2 / 2 = 1$$

$$IDF(E) = 2/1 = 2$$

|     | <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> |
|-----|----------|----------|----------|----------|----------|
| 文档1 | 1/8      | 1/16     | 3/16     | 9/16     | 1/8      |
| 文档2 | 1/18     | 5/18     | 1/9      | 5/9      | 0        |

# 模型训练及测试

- 模型选择：对特定任务最优建模方法的选择或者对特定模型最佳参数的选择。
- 在训练数据集上运行模型(算法)并在测试数据集中测试效果，迭代进行数据模型的修改，这种方式被称为**交叉验证**(将数据分为**训练集**和**测试集**，使用训练集构建模型，并使用测试集评估模型提供修改建议)
- 模型的选择会尽可能多的选择算法进行执行，并比较执行结果
- 模型的测试一般以下几个方面来进行比较，分别是**准确率/召回率/精准率/F值**
  - ◆ 准确率(Accuracy)=提取出的正确样本数/总样本数
  - ◆ 召回率(Recall)=正确的正例样本数/样本中的正例样本数——覆盖率
  - ◆ 精准率(Precision)=正确的正例样本数/预测为**正例**的样本数
  - ◆  $F值 = \frac{Precision * Recall * 2}{Precision + Recall}$  (即F值为准确率和召回率的调和平均值)



# 模型训练及测试

|     |    | 预测值    |        |
|-----|----|--------|--------|
|     |    | 正例     | 负例     |
| 真实值 | 正例 | 真正例(A) | 假负例(B) |
|     | 负例 | 假正例(C) | 真负例(D) |

A和D预测正确，B和C预测错误，测试计算结果为：

$$Accuracy = \frac{\#(A) + \#(D)}{\#(A) + \#(B) + \#(C) + \#(D)}$$

$$Recall = \frac{\#(A)}{\#(A) + \#(B)} \quad Precision = \frac{\#(A)}{\#(A) + \#(C)} \quad F = \frac{2 * Recall * Precision}{Recall + Precision}$$

# 混淆矩阵

|  |                    | predicted condition   |  |   |  |
|--|--------------------|---|--|---|--|
| total population   |                    | prediction positive   | prediction negative  | Prevalence = $\frac{\Sigma \text{condition positive}}{\Sigma \text{total population}}$  |  |
| true condition   | condition positive | True Positive (TP)  | False Negative (FN)<br>(type II error)   | True Positive Rate (TPR),<br>Sensitivity, Recall,<br>Probability of Detection<br>$= \frac{\Sigma TP}{\Sigma \text{condition positive}}$ | False Negative Rate (FNR),<br>Miss Rate<br>$= \frac{\Sigma FN}{\Sigma \text{condition positive}}$        |
|  | condition negative | False Positive (FP)<br>(Type I error)   | True Negative (TN)   | False Positive Rate (FPR),<br>Fall-out,<br>Probability of False Alarm<br>$= \frac{\Sigma FP}{\Sigma \text{condition negative}}$         | True Negative Rate (TNR),<br>Specificity (SPC)<br>$= \frac{\Sigma TN}{\Sigma \text{condition negative}}$ |
| Accuracy<br>$= \frac{\Sigma TP + \Sigma TN}{\Sigma \text{total population}}$ |                    | Positive Predictive Value (PPV),<br>Precision = $\frac{\Sigma TP}{\Sigma \text{prediction positive}}$ | False Omission Rate (FOR)<br>$= \frac{\Sigma FN}{\Sigma \text{prediction negative}}$       | Positive Likelihood Ratio (LR+)<br>$= \frac{TPR}{FPR}$  | Diagnostic Odds Ratio (DOR)<br>$= \frac{LR+}{LR-}$   |
|  |                    | False Discovery Rate (FDR)<br>$= \frac{\Sigma FP}{\Sigma \text{prediction positive}}$                 | Negative Predictive Value (NPV)<br>$= \frac{\Sigma TN}{\Sigma \text{prediction negative}}$ | Negative Likelihood Ratio (LR-)<br>$= \frac{FNR}{TNR}$  |  |

[http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html)

[https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)

# 模型评估

- 准确率 Accuracy

- $$\text{Accuracy} = \frac{\#(\text{True positive}) + \#(\text{True negative})}{\#(\text{True positive}) + \#(\text{True negative}) + \#(\text{False positive}) + \#(\text{False negative})}$$

- 召回率 Recall

- $$\text{Recall} = \frac{\#(\text{True positive})}{\#(\text{True positive}) + \#(\text{False negative})}$$

- 精确率 Precision

- $$\text{Precision} = \frac{\#(\text{True positive})}{\#(\text{True positive}) + \#(\text{False positive})}$$

- F1指标 F1 measure

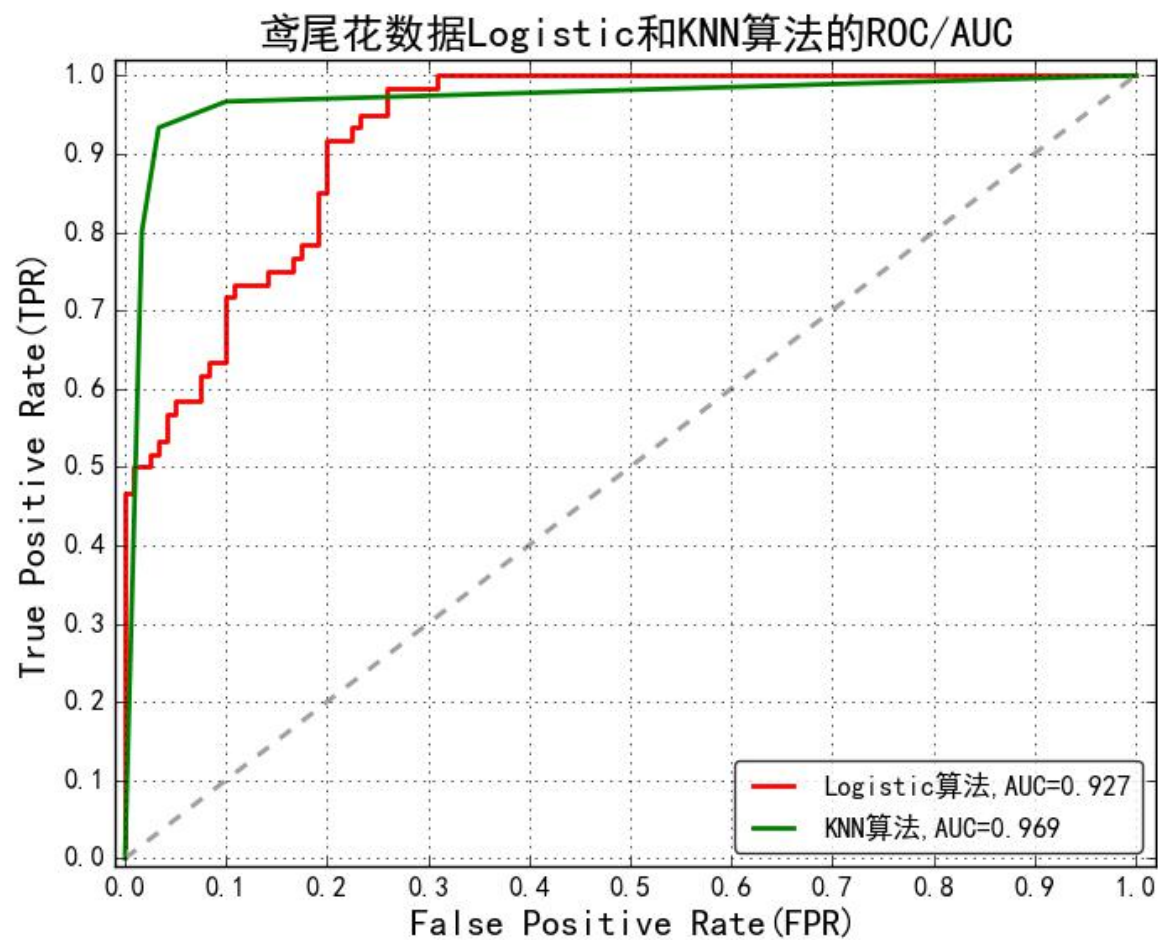
- $$F1\ measure = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

ROC ( Receiver Operating Characteristic ) 最初源于20世纪70年代的信号检测理论，描述的是分类混淆矩阵中FPR-TPR两个量之间的相对变化情况，ROC曲线的纵轴是“真正例率” ( True Positive Rate 简称TPR )，横轴是“假正例率” ( False Positive Rate 简称FPR )。

如果二元分类器输出的是对正样本的一个分类概率值，当取不同阈值时会得到不同的混淆矩阵，对应于ROC曲线上的一个点。那么ROC曲线就反映了FPR与TPR之间权衡的情况，通俗地说，即在TPR随着FPR递增的情况下，谁增长得更快，快多少的问题。TPR增长得越快，曲线越往上屈，AUC就越大，反映了模型的分类性能就越好。当正负样本不平衡时，这种模型评价方式比起一般的精确度评价方式的好处尤其显著。

# ROC

## ROC曲线



# AUC

## ■ AUC的值越大表达模型越好

AUC ( Area Under Curve ) 被定义为ROC曲线下的面积，显然这个面积的数值不会大于1。又由于ROC曲线一般都处于 $y=x$ 这条直线的上方，所以AUC的取值范围在0.5和1之间。使用AUC值作为评价标准是因为很多时候ROC曲线并不能清晰的说明哪个分类器的效果更好，而AUC作为数值可以直观的评价分类器的好坏，值越大越好。

$AUC = 1$ ，是完美分类器，采用这个预测模型时，不管设定什么阈值都能得出完美预测。绝大多数预测的场合，不存在完美分类器。

$0.5 < AUC < 1$ ，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。

$AUC = 0.5$ ，跟随机猜测一样（例：丢铜板），模型没有预测价值。

$AUC < 0.5$ ，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。

# 模型评估

## ■ 回归结果度量

- ◆ `explained_varicance_score` : 可解释方差的回归评分函数
- ◆ `mean_absolute_error` : 平均绝对误差
- ◆ `mean_squared_error` : 平均平方误差

# 模型评估总结\_分类算法评估方式

| 指标               | 描述        | scikit-learn函数  |
|------------------|-----------|---|
| Precision        | 精确度       | <code>from sklearn.metrics import precision_score</code>  |
| Recall           | 召回率       | <code>from sklearn.metrics import recall_score</code>     |
| F1               | F1指标      | <code>from sklearn.metrics import f1_score</code>         |
| Confusion Matrix | 混淆矩阵      | <code>from sklearn.metrics import confusion_matrix</code> |
| ROC              | ROC曲线     | <code>from sklearn.metrics import roc</code>              |
| AUC              | ROC曲线下的面积 | <code>from sklearn.metrics import auc</code>              |



# 模型评估总结\_回归算法评估方式

| 指标                            | 描述   | scikit-learn函数  |
|-------------------------------|------|---|
| Mean Square Error (MSE, RMSE) | 平均方差 | <code>from sklearn.metrics import mean_squared_error</code>                         |
| Absolute Error (MAE, RAE)     | 绝对误差 | <code>from sklearn.metrics import mean_absolute_error, median_absolute_error</code> |
| R-Squared                     | R平方值 | <code>from sklearn.metrics import r2_score</code>                                   |

# 模型部署和整合

- 当模型构建好后，将训练好的模型存储到数据库中，方便其它使用模型的应用加载(构建好的模型一般为一个矩阵)
- 模型需要周期性
- 一个月、一周

# 模型的监控与反馈

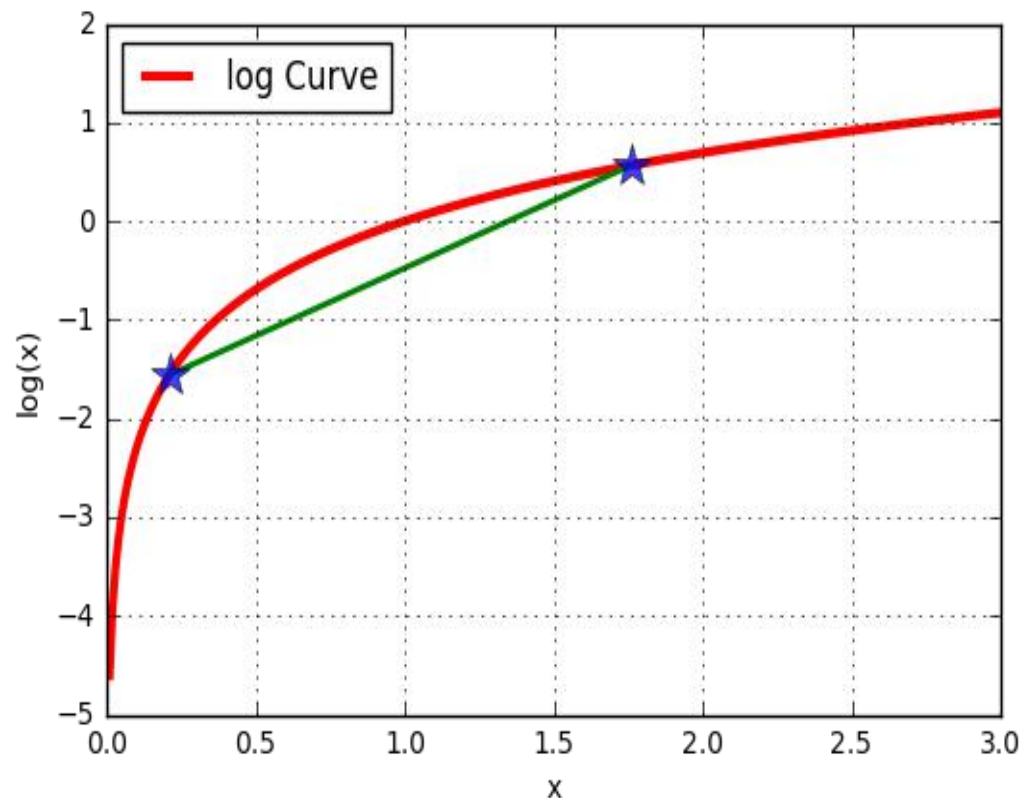
- 当模型一旦投入到实际生产环境中，模型的效果监控是非常重要的，往往需要关注业务效果和用户体验，所以有时候会进行A/B测试
- 模型需要对用户的反馈进行响应操作，即进行模型修改，但是要注意异常反馈信息对模型的影响，故需要进行必要的预处理操作

# 本课程软件环境

- Anaconda5
- Python 3.6

# Python code 示例1

```
import math
import matplotlib.pyplot as plt
if __name__=="__main__":
    x=[float(i)/100.0 for i in range(1,300)]
    y=[math.log(i) for i in x]
    plt.plot(x,y,'r-',linewidth=3,label='log Curve')
    a=[x[20],x[175]]
    b=[y[20],y[175]]
    plt.plot(a,b,'g-',linewidth=2)
    plt.plot(a,b,'b*',markersize=15,alpha=0.75)
    plt.legend(loc='upper left')
    plt.grid(True)
    plt.xlabel('x')
    plt.ylabel('log(x)')
    plt.show()
```

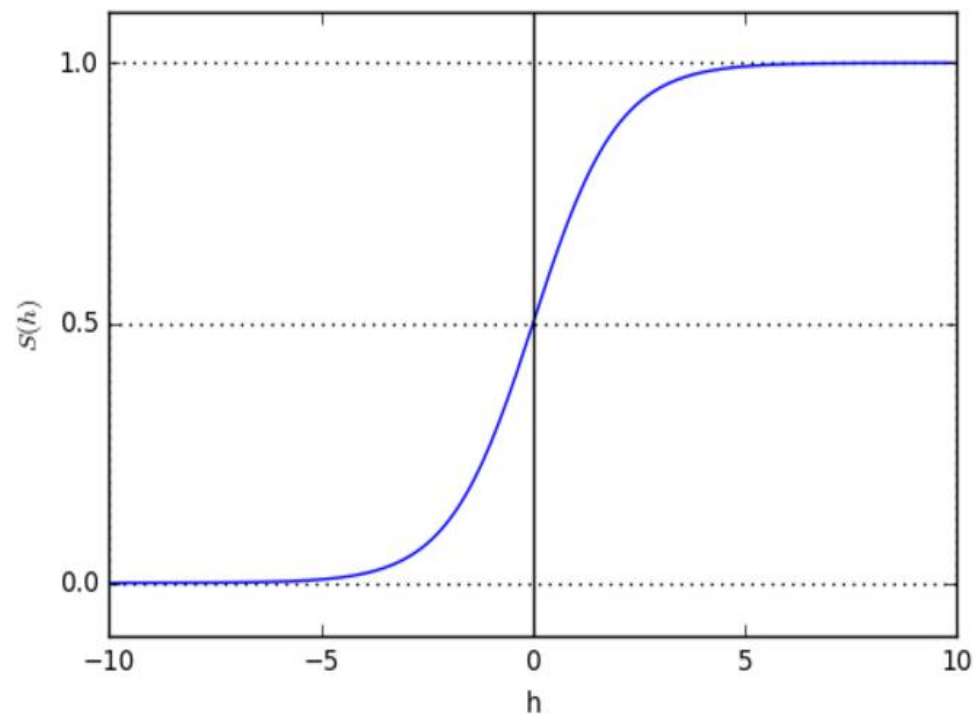


# Python code 示例2

```
import matplotlib.pyplot as plt
import numpy as np

def sigmoid(h):
    return 1.0 / (1.0 + np.exp(-h))

h = np.arange(-10, 10, 0.1) # 定义x的范围, 像素为0.1
s_h = sigmoid(h) # sigmoid为上面定义的函数
plt.plot(h, s_h)
plt.axvline(0.0, color='k') # 在坐标轴上加一条竖直的线, 0.0为竖直线在坐标轴上的位置
plt.axhspan(0.0, 1.0, facecolor='1.0', alpha=1.0, ls='dotted') # 加水平间距通过坐标轴
plt.axhline(y=0.5, ls='dotted', color='k') # 加水线通过坐标轴
plt.yticks([0.0, 0.5, 1.0]) # 加y轴刻度
plt.ylim(-0.1, 1.1) # 加y轴范围
plt.xlabel('h')
plt.ylabel('$S(h)$')
plt.show()
```





# THANK YOU

上海育创网络科技有限公司