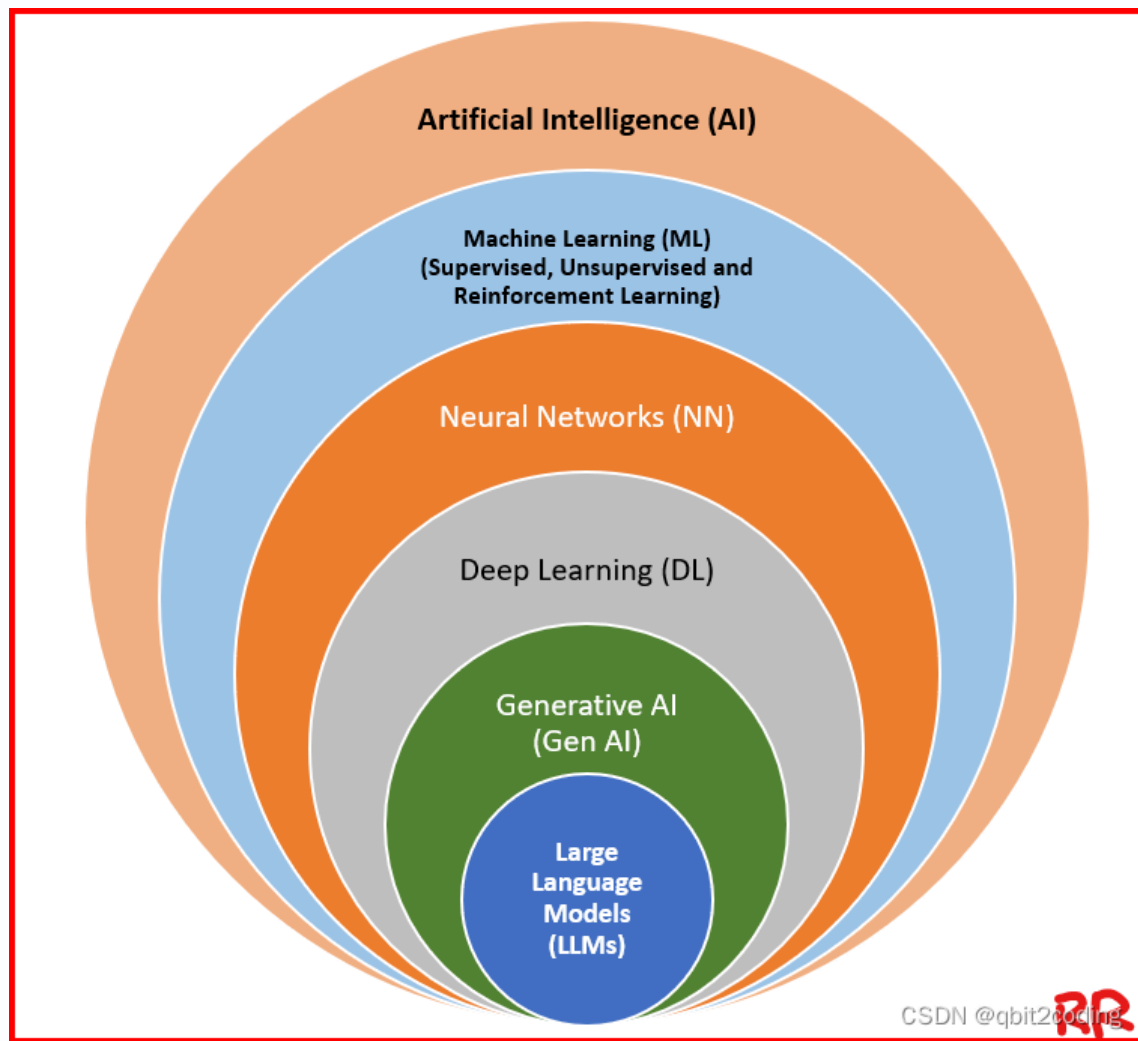


Market Research and Analysis

Lecture 2: Introduction to Machine Learning

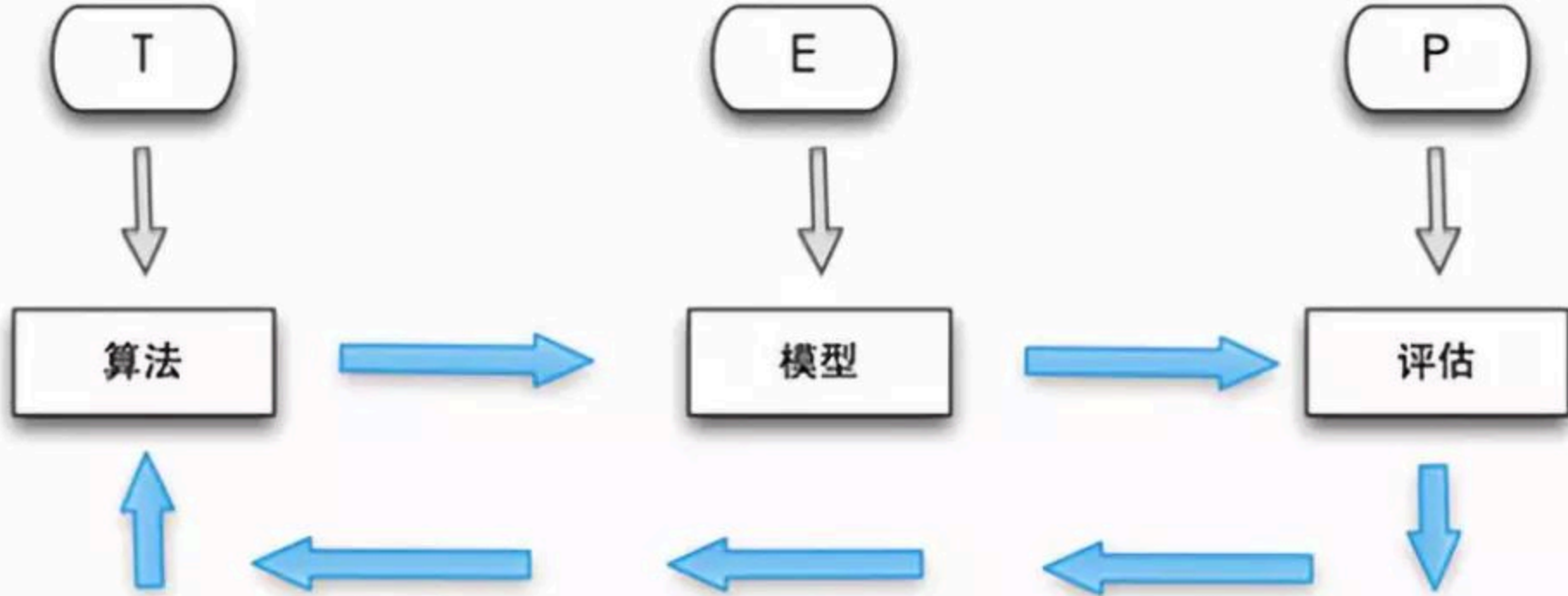
- Zhenyu Zhao
- Nankai Institute of International Economics
- Feishu: 2120253538
- Email: zzynankai@outlook.com
- Website: xishanyu2.github.io



01

机器学习导论

Machine learning is a subfield of AI that trains algorithms to learn from data in order to make predictions without being explicitly programmed.



A computer program is 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**.
假设用**P**衡量计算机程序在任务**T**上的性能，如果计算机程序在任务**T**上的性能随着经验**E**而提高，则称计算机程序从经验**E**中学习。——*Tom Mitchell*



奥卡姆剃刀原理
(Ockham's Razor)

如无必要，勿增实体

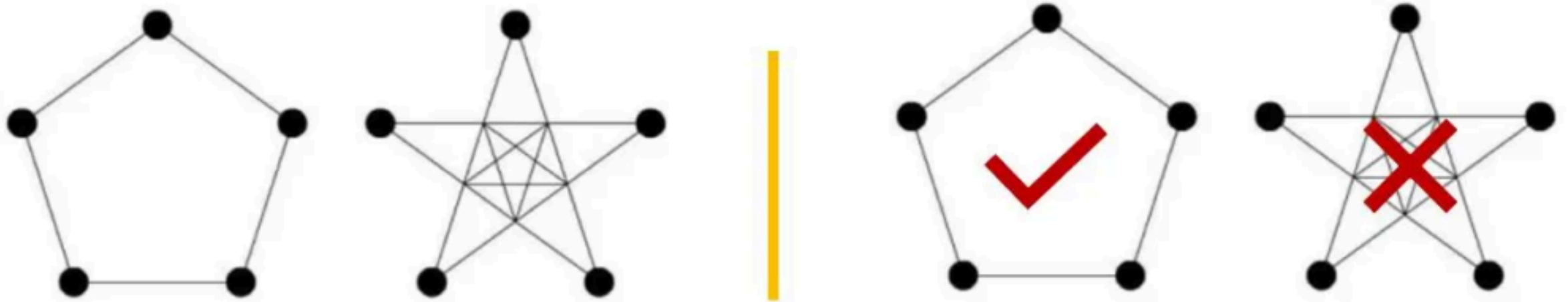
一切勿浪费较多东西去做用较少的东西同样可以做好的事情。



1

奥卡姆剃刀

简单有效原理：避重就轻，避繁逐简，以简御繁，避虚就实



一个事实如果同时被两个理论解释，选择简单的那一个

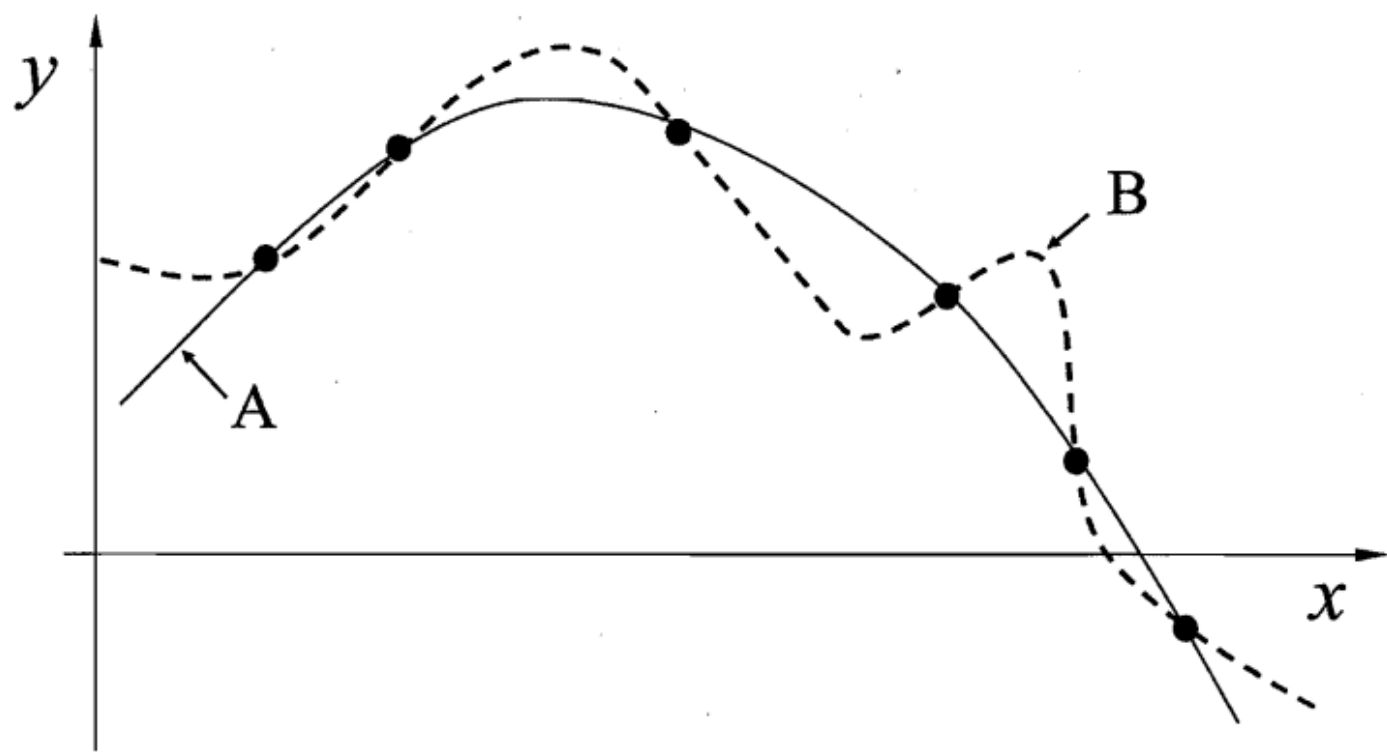
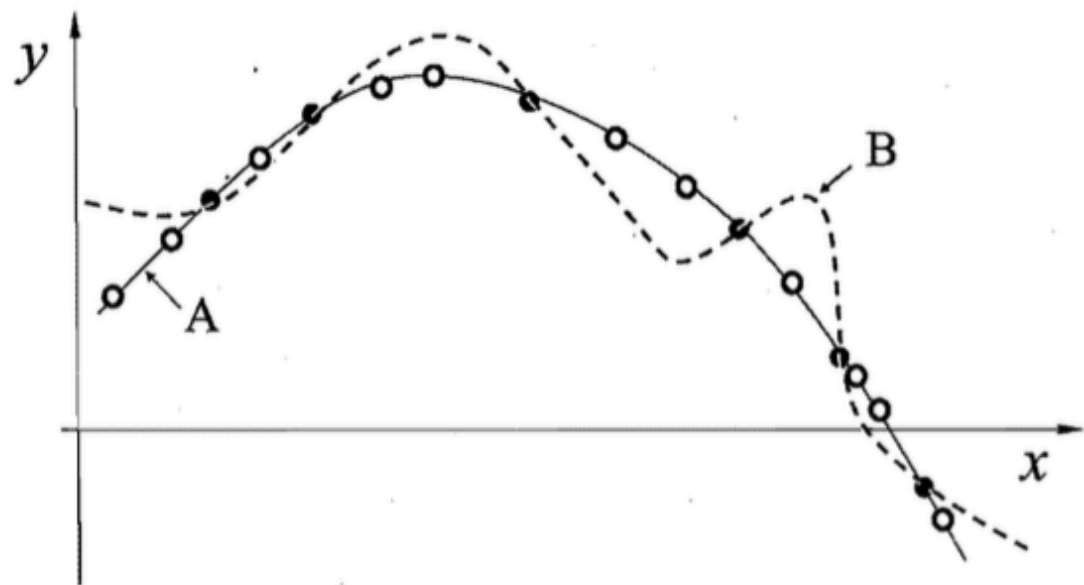
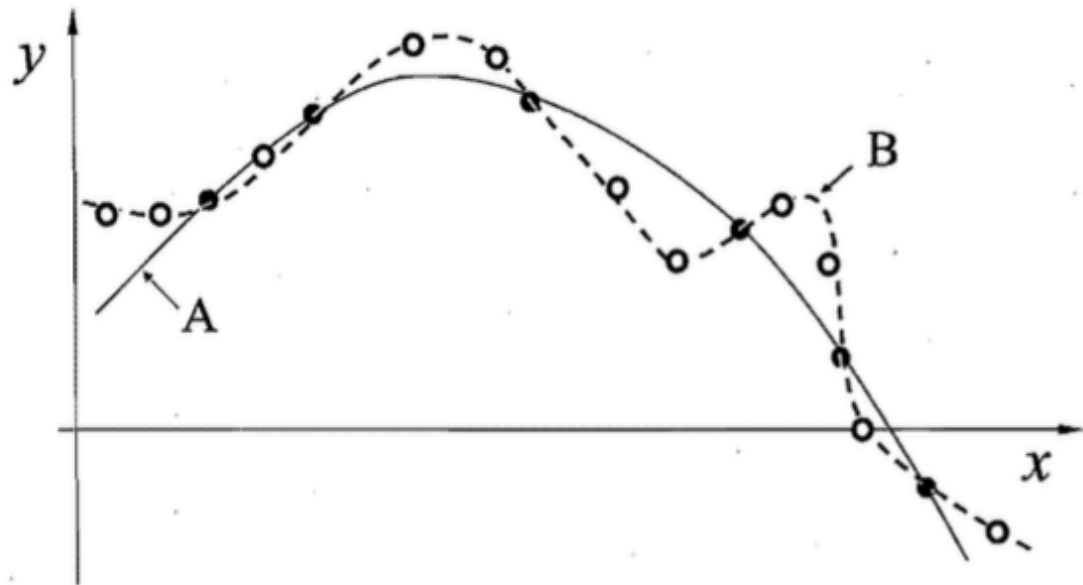


图 1.3 存在多条曲线与有限样本训练集一致



(a) A 优于 B



(b) B 优于 A

图 1.4 没有免费的午餐. (黑点: 训练样本; 白点: 测试样本)

NFL定理

定理内容：一个算法A若在某些问题上比另一个算法B好，必存在另一些问题，B比A好。

重要前提：所有“问题”出现的机会相同、或所有问题同等重要；实际情形并非如此，我们通常只关注自己正在试图解决的问题；脱离具体问题，空泛地谈论“什么学习算法更好”毫无意义！

具体问题，具体分析！

最优方案往往来自：按需设计、度身定制。

Traditional classification

- supervised (labelled data)
 - regression: predict quantity
 - classification: predict index (categorical variable)
- unsupervised (no labels)
 - dimension reduction
 - clustering
- semi-supervised / self-supervised
- reinforcement learning

02

回归问题与模型拟合

如果要预测的值是连续变量，如房价，那么就属于回归问题。

线性回归 (Linear Regression)

线性回归通过回归分析确定线性相关变量间的定量关系。给定含有响应和n个特征变量的数据，若特征变量与响应之间线性相关，可采用线性回归模型学得二者之间的线性关系，即：

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon_i$$

用矩阵形式写成：

$$Y = X\beta + \varepsilon$$

算法从数据中学得 β 之后，模型得以确定，此后可以用于预测新样本的标签值。

为何使用线性模型？难道世界不是非线性的吗？因为线性模型有以下优点：

1. 线性模型十分简单，且易于解释。在线性模型的假设下，每个特征变量 x 对响应变量 y 的边际效应为常数。
2. 即使 $f(x, \beta)$ 为非线性函数，在足够小的局部，一般也可用线性函数来近似（一阶泰勒展开）。
3. 线性模型虽然简单，但可作为复杂模型的组成部分。从线性模型入手有助于理解机器学习的思想与方法。

线性回归是一切回归问题的基础，也是机器学习的重要模型之一。线性回归模型虽形式简单，却蕴含着机器学习中一些关键的基本思想，许多复杂的机器学习模型都是从简单的线性回归模型演化而来。因其易于建模、可解释性强等优点，线性回归模型至今仍被广泛使用。

传统回归:

- 全样本: $X, Y = (x_i, y_i)_{i=1:n}$
- OLS: $\min_{\beta} \sum_{i=1}^n (\beta_0 + \beta_i x_i - y_i)^2$
- 闭式解: $\beta = (X'X)^{-1} X'Y$
- 当 X 非常大时, 计算困难 (**Curse of Dimensionality**)

增量学习(Incremental learning):

- 给定初始值 β_j
- 随机选取 m 个观测值 (作为一个批次(**batch**))
 - 进行回归, 得到新的估计值 β
 - 使误差平方和最小化
- 使用 (可能变化的) **学习率** α 进行更新:
 - $\beta_{j+1} \leftarrow \beta_j(1 - \alpha) + \alpha\beta$
- 只要 α 随时间下降得足够快, 该过程就是无偏的 (即 β 会收敛到真实值)

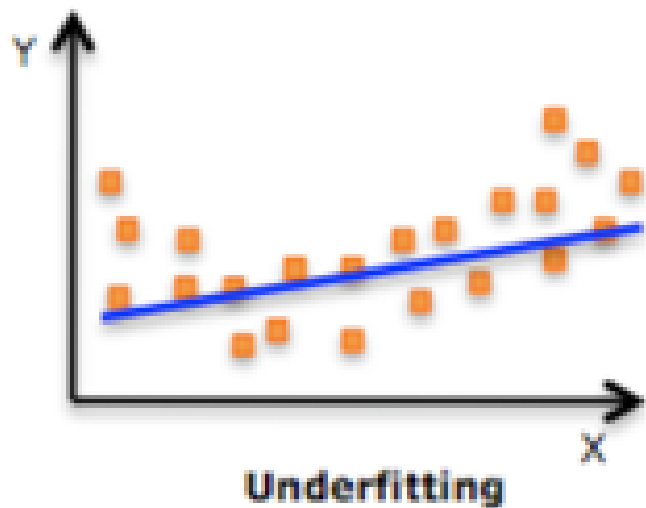
一切为了泛化!

不同于经典统计方法在全样本数据上拟合模型，机器学习方法训练模型时只用到了样本数据的一个子集，即**训练集**；另一个子集则用于评估模型的**泛化能力**（Generalization Ability），称为**测试集**。

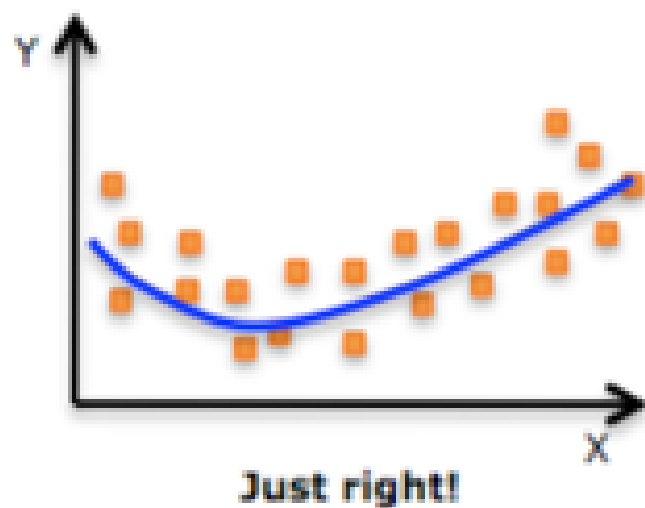
模型泛化能力是机器学习训练模型追求的目标，它是指算法在一个数据集上学习到的模型，运用到新数据集上的表现。在新数据集上的表现越好，说明模型泛化能力越强，模型就越优。为更准确度量模型泛化能力，测试集应尽可能与训练集数据不重合。如果测试样本被用于训练，得到的将会是过于乐观的估计结果。

以高考为例：高考前练习的所有题目相当于训练集，高考考题相当于测试集，每年的高考考题都会避免与此前市面已有题目重合，正是为了更加准确地考察考生对知识的泛化能力。

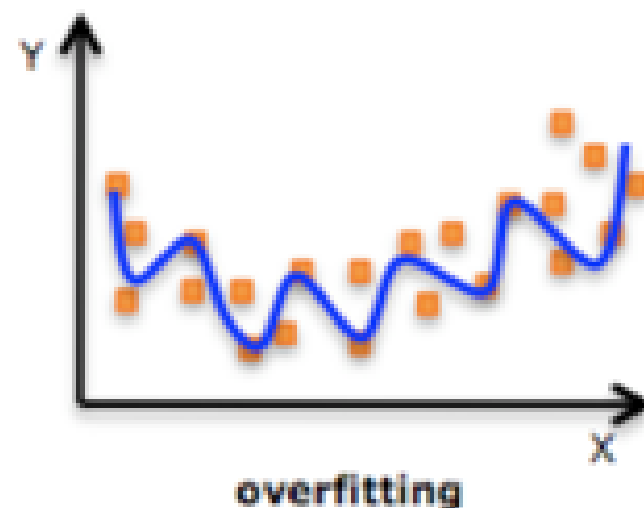
在很多现实场景中数据往往存在非线性关系，面对数据关系更加复杂的场景，线性回归模型是否还能应用？模型在拟合训练集数据时又会产生什么问题？



$$\theta_0 + \theta_1 x$$



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

模型复杂度与泛化能力之间的关系：

线性回归模型拟合效果并不好，这种“模型过于简单而不能很好地拟合数据”的现象，称为**欠拟合 (Underfitting)**。更准确地说，欠拟合是指模型尚未学好训练样本的一般规律，不能完整地表述数据关系。

在加入二次项后，模型拟合效果得到了显著提升：拟合曲线很好地捕捉到了数据的U型关系。这种现象被称为**恰拟合 (Rightfitting)**，即模型从训练样本中尽可能地学到了适用于所有潜在新样本的普遍规律，这是机器学习训练模型追求的目标。

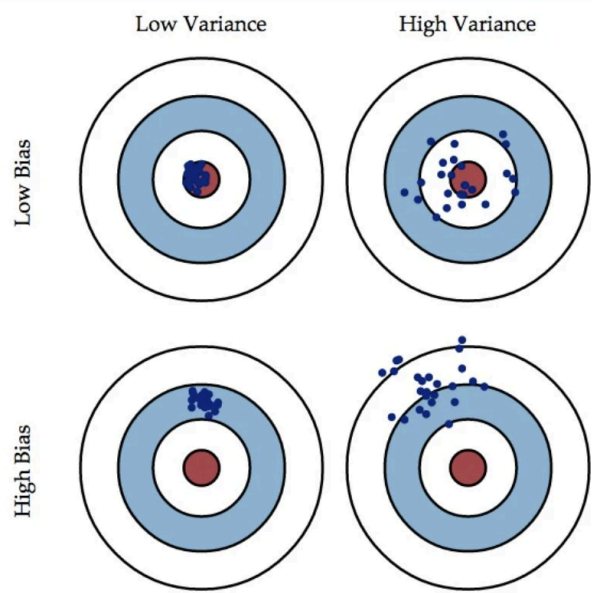
训练集拟合效果进一步提升，但测试集预测能力却下降了，这被称为**过拟合 (Overfitting)**。更准确地说，过拟合是指训练的模型过多地表达了样本数据的特有关系，把训练样本自身独特规律当作所有潜在样本都会具有的一般规律，从而导致泛化能力下降。

偏差 (Bias) 和方差 (Variance)

偏差：采用不同数据集进行算法训练所获得模型的预测平均值对真实值的**偏离程度**，通常采用各预测值的期望与真实值之差来衡量。偏差度量了学习算法在给定训练数据集下的预测准确度，即刻画算法本身的拟合能力。

方差：采用不同数据集进行算法训练所获得模型预测值的**离散程度**，即各预测值相对其期望值的波动幅度。方差度量了训练数据集的变动所导致学习性能的变化，即刻画了数据样本扰动所造成的影响。

偏差和方差分别从两个角度描述学习到的模型与真实模型之间的差距。



泛化误差 (Generalization Error)

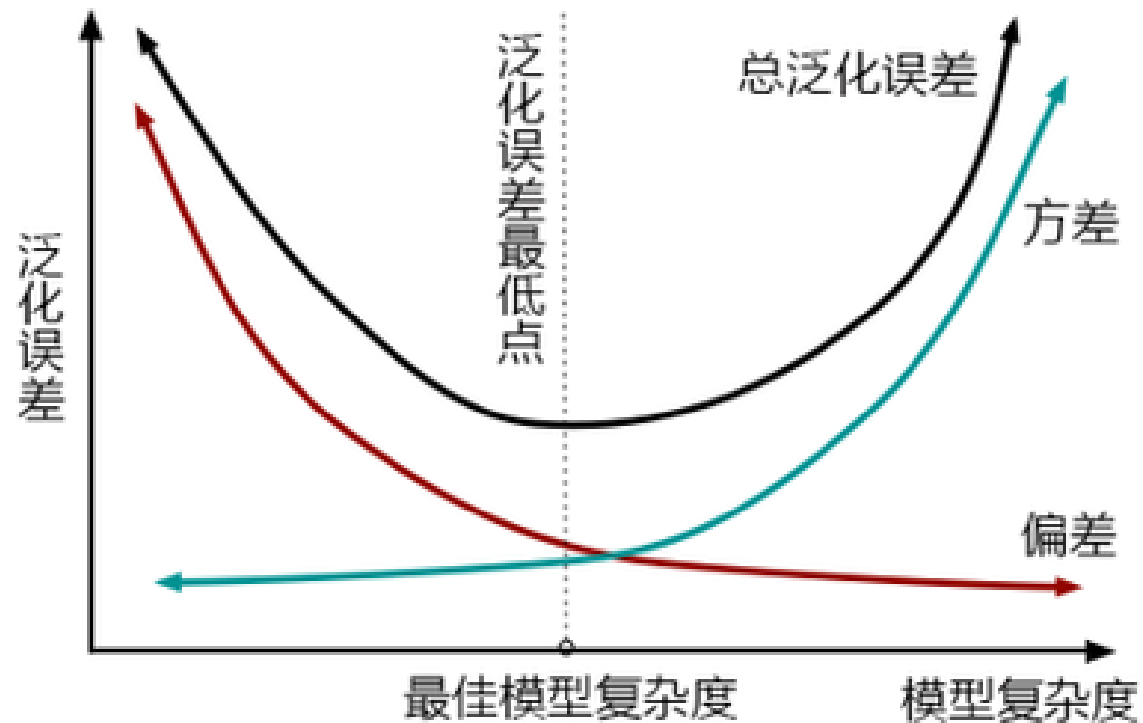
模型泛化能力是机器学习算法训练的核心目标，通常使用**泛化误差**来度量。所谓泛化误差即算法训练所得模型对未知数据预测的平均误差。泛化误差越小，这说明模型泛化能力越强。

模型泛化误差可分解为**偏差、方差和噪声**三个部分。偏差和方差分别刻画了算法本身拟合能力和数据样本扰动所造成的影响。噪声则表达了学习算法所能达到的期望泛化误差的下界，刻画了机器学习问题本身难度。

$$Err(x) = bias^2(x) + var(x) + \epsilon^2$$

$$\begin{aligned}
E(f; D) &= \mathbb{E}_D [(f(\mathbf{x}; D) - y_D)^2] \\
&= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}) + \bar{f}(\mathbf{x}) - y_D)^2] \\
&= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] + \mathbb{E}_D [2(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))(\bar{f}(\mathbf{x}) - y_D)] \\
&= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] \\
&= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y + y - y_D)^2] \\
&= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)^2] + \mathbb{E}_D [(y - y_D)^2] + 2\mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)(y - y_D)] \\
&= \underbrace{\mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2]}_{\text{variance}} + \underbrace{(\bar{f}(\mathbf{x}) - y)^2}_{\text{bias}^2} + \underbrace{\mathbb{E}_D [(y - y_D)^2]}_{\text{noise}}
\end{aligned}$$

随着模型复杂度变化，偏差和方差在模型预测误差中所占比重也会发生变化，并进一步影响模型拟合能力。模型复杂度较低时，模型拟合能力较差，此时**偏差主导**泛化误差，模型处于**欠拟合**阶段。随着模型复杂度提升，模型拟合能力增强，模型逐步捕捉到训练数据中蕴含的各种细节规律甚至噪声，此时**方差主导**泛化误差，模型处于**过拟合**阶段。模型训练最理想结果是得到低偏差且低方差的模型，然而，偏差与方差的冲突导致现实中几乎无法得到这种理想模型，只能无限近似。因此，更为现实的选择是通过**权衡偏差与方差**，达到**泛化误差最小**的平衡点。这个权衡偏差与方差的过程，也即探索追求**恰拟合**状态的过程。



正则化 (Regularization)

当模型处于欠拟合状态时，其在训练集上表现较差，因此相对容易被检测和识别。欠拟合问题比较容易克服，通过增加模型复杂度就可以使模型更加“细腻”地学习样本中蕴含的特征。相比于模型过于简单导致的欠拟合，模型过于复杂导致的过拟合更值得关注。过拟合模型在训练集上表现非常好，仅在测试集上表现很差，这使其容易被误解为性能较好。**正则化**是解决过拟合问题的经典方法。

正则化是一种纠正策略：通过约束参数得到简化的模型，从而降低过拟合风险。通过在损失函数中添加特定**惩罚项** (Penalty Term)，在算法学习过程中抑制模型系数（使某些系数趋向于零）控制模型复杂度，进而缓解模型过拟合现象。根据添加惩罚项的不同形式，可分为**L1正则化**和**L2正则化**。

正则化一般具有如下形式：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$

其中，第1项是经验风险，第2项是正则化项， $\lambda \geq 0$ 为调节两者之间关系的系数。

套索回归 (LASSO Regression)

L1 正则化又称为**套索回归**，其损失函数为：

$$L(w) = \frac{1}{N} \sum_{i=1}^N (f(x_i; w) - y_i)^2 + \lambda \|w\|_1$$

其中，L1范数是各系数的绝对值之和。

在L1范数的作用下，如果系数 ω 变大，那么损失函数 $L(\omega)$ 也会变大。因此在算法学习过程中，为保证损失函数尽可能小，系数 ω 也应尽可能小。

L1范数前的系数 λ 代表正则化强度， λ 越大，正则化程度就越强。极端情况下，当 λ 趋向无穷大时，所有系数都会变为0，此时多项式模型的拟合曲线将变成一条水平直线。

岭回归 (Ridge Regression)

L2 正则化又称为**岭回归**，其损失函数为：

$$L(w) = \frac{1}{N} \sum_{i=1}^N (f(x_i; w) - y_i)^2 + \frac{\lambda}{2} \|w\|^2$$

L2范数为模型系数平方和再求平方根（欧氏距离）。

在L2范数的作用下，如果系数 ω 变大，那么损失函数 $L(\omega)$ 也会变大。因此在算法学习过程中，为保证损失函数尽可能小，系数 ω 也应尽可能小。

λ 仍然代表正则化强度， λ 越大则惩罚强度越大。极端情况下，当 λ 趋向无穷大时，所有系数都会变为0，此时多项式模型的拟合曲线将变成一条水平直线。

总结

在机器学习算法训练过程中，有两种不同机制共同作用于损失函数最小化：

- 高复杂度模型（模型系数远离0）可以更好拟合训练数据，有利于降低损失函数值。
- 正则化通过引入惩罚项增加了“反向牵引”力量，此时低复杂度模型（模型系数趋向0）更有利于损失函数最小化，从而避免用过于复杂的模型拟合训练数据。

如果选择的正则化参数 λ 过大，则会把所有参数都最小化，导致模型变成 $f(x, w) = w_0$ ，造成欠拟合。

若 λ 的值太小了，那么相当于退回为普通的最小二乘回归，造成过拟合。

两种不同正则化方法的差异：L1正则化使模型系数分布更为稀疏，而L2正则化使模型系数分布更为平滑。L1正则化采用绝对值惩罚项，这意味着绝对值较小的系数被惩罚后很容易变成0；L2正则化采用平方惩罚项，这意味着当系数绝对值趋近于0时，惩罚将非常轻微。因此，岭回归结果倾向于**保留更多特征项**，而随着海量数据处理兴起，对于模型稀疏化的要求也随之出现，利用L1正则化的“特征选择”功能可**删除非必要特征变量**。

数学附录

岭回归最优化问题

目标函数：

$$\min L(\beta, \lambda) = \sum_{i=1}^n (y_i - X_i' \beta)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

即：

$$\min_{\beta, \lambda} (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \quad s.t. \quad \|\beta\|_2^2 \leq t$$

拉格朗日乘数法：

$$\min_{\beta, \lambda} \tilde{L}(\beta) = (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda(\|\beta\|_2^2 - t)$$

解得： $\hat{\beta} = (X'X + \lambda I)^{-1} X'y$

在X为正交矩阵的特殊情况下， $\hat{\beta}^{Ridge} = \frac{\hat{\beta}^{OLS}}{1+\lambda}$ 。

岭回归是对最小二乘回归的一种补充，在方差和误差的权衡中，Ridge以有偏的代价换取更小的方差。

偏差：

$$Bias_{Ridge} = \frac{\beta}{1 + \lambda} - \beta \neq 0$$

方差：

$$Var_{Ridge} = \frac{\sigma^2}{(1 + \lambda)^2} \leq \sigma^2$$

存在性定理：总是存在一个 λ 使得 $\hat{\beta}^{Ridge}$ 的MSE小于 $\hat{\beta}^{OLS}$ 的MSE.

Lasso回归最优化问题

目标函数：

$$\min L(\beta, \lambda) = \sum_{i=1}^n (y_i - X_i' \beta)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

即：

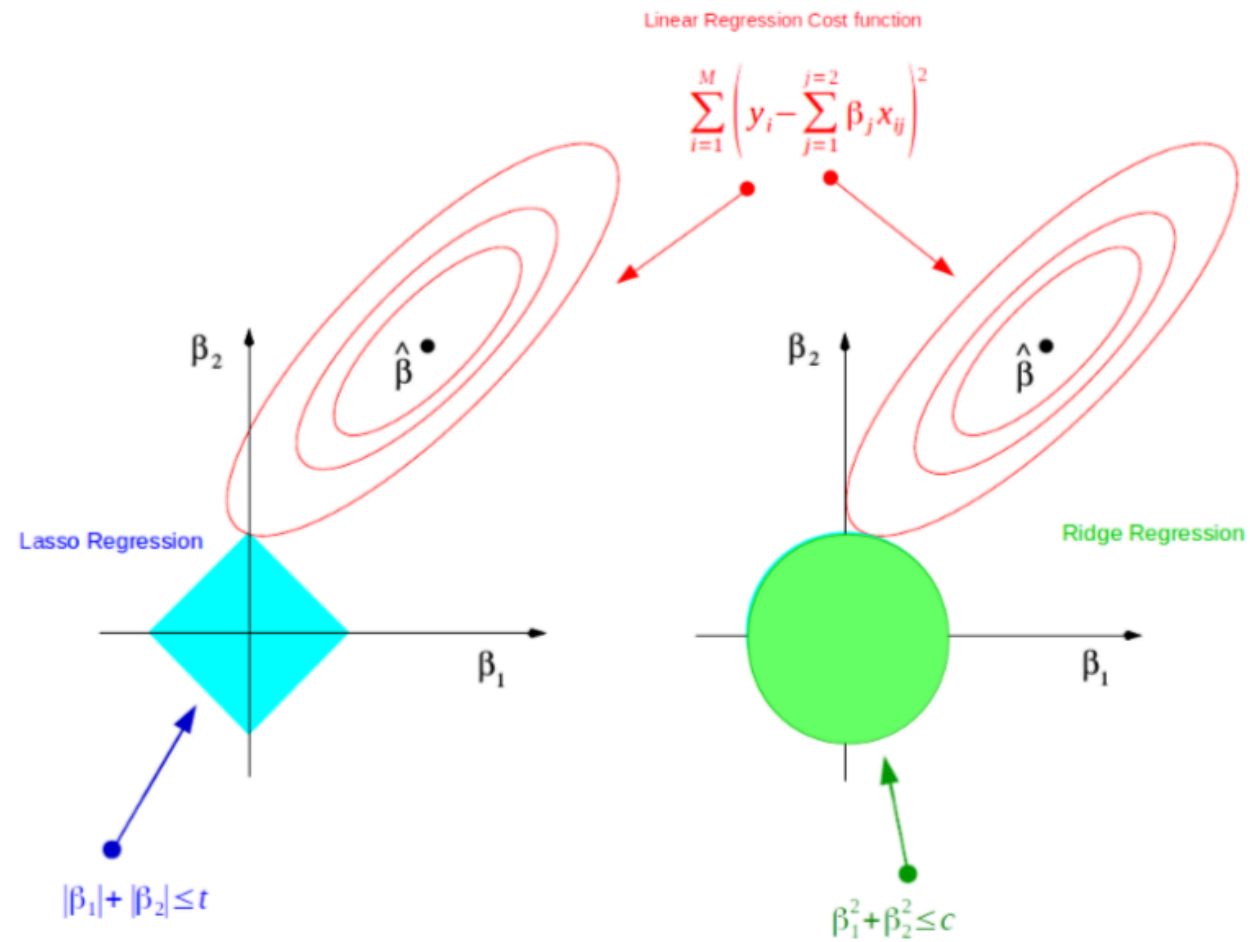
$$\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) \quad s.t. \quad \|\beta\|_1 \leq t$$

此约束极值问题的约束集不再是**圆形**，而是**菱形**或高维的菱状体。

为了在可行集中最小化残差平方和，Lasso的最优解必然发生于等值线与菱形线相切的位置，而菱形的顶点恰好在坐标轴上，故等值线较易与约束集**相切于坐标轴**的位置，导致Lasso估计量的某些回归系数严格**等于0**，从而得到“稀疏解”(sparse solution)。

由于Lasso为“绝对值收缩”(absolute shrinkage)，故合称为“最小绝对值收缩与筛选算子”(Least Absolute Shrinkage and Selection Operator)，简记LASSO。

Dimension Reduction of Feature Space with LASSO



弹性网 (Elasti Net)

将Lasso与岭回归相结合，得到**弹性网**估计量，即在弹性网估计量的损失函数中，同时包含L1与L2惩罚项：

$$\min_{\beta, \lambda} (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$$

弹性网为岭回归与Lasso之间的折衷，兼具两者特征，岭回归与Lasso都是弹性网的特例。

- 与Lasso的菱形约束集类似，弹性网的约束集在坐标轴上有四个尖角，故弹性网也具有**筛选变量**的功能；
- 与岭回归的圆形约束集类似，弹性网的约束集在四个象限也呈弧形，故弹性网也具有**收缩参数**的功能。

03

分类问题与模型评估

如果要预测的值是离散变量，即一个个标签，那么就属于分类问题。

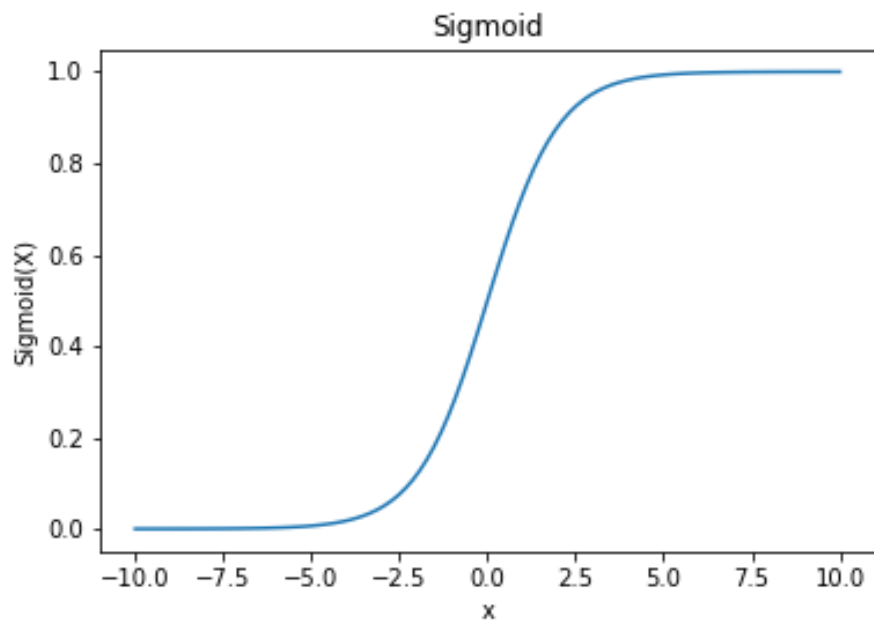
Common classification algorithms

- Logistic Regression
- Nearest Distance
- Decision Trees
- Support Vector Machines
- Naive Bayes Classifier
- Neural Networks

以过滤垃圾邮件为例，假设响应变量只有两种可能取值，即 $y=1$ （垃圾邮件）或 $y=0$ （正常邮件），这种0-1变量称为虚拟变量（dummy variable）或哑变量。最简单的建模方法为**线性概率模型**（Linear Probability Model），LPM的优点在于计算方便（就是OLS估计），且容易得到边际效应（即回归系数）；但LPM一般并不适合作预测，因为 y 的取值非0即1，但根据LPM所作的预测值却可能出现大于1或小于0的不现实情形。

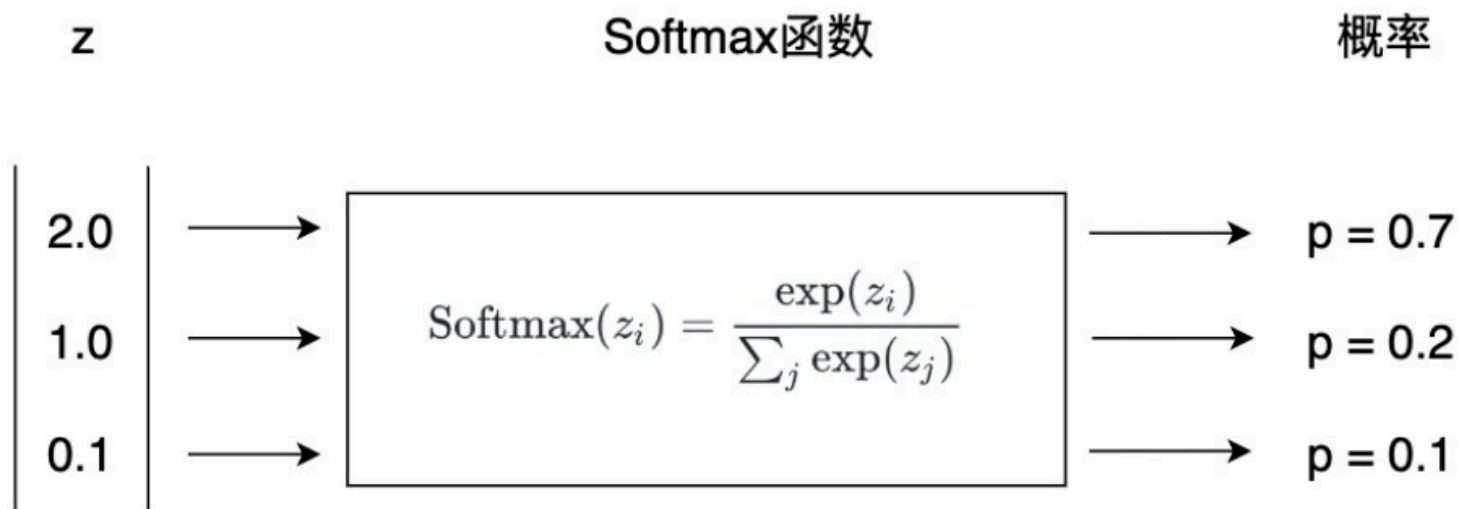
可以通过一个**映射函数**将LPM预测值范围转换至(0,1)区间，再通过设定阈值将其输出为0/1二分类值。

逻辑函数（Logistic Function）/sigmoid函数： $y = \frac{1}{1+e^{-x}}$



多元逻辑回归 (Multinomial Logit Regression)

二元逻辑回归使用**sigmoid函数**将线性预测结果转化为概率，并通过设定阈值来划分类别。多元逻辑回归则使用**softmax函数**将线性预测结果转化为多个类别的概率分布，每个类别的概率表示样本属于该类别的可能性，最终选择概率最大的类别作为分类结果。



相较于多元逻辑回归，**One-vs-all多元分类**算法原理更为简单，其本质是将多元分类视为多个二元分类。

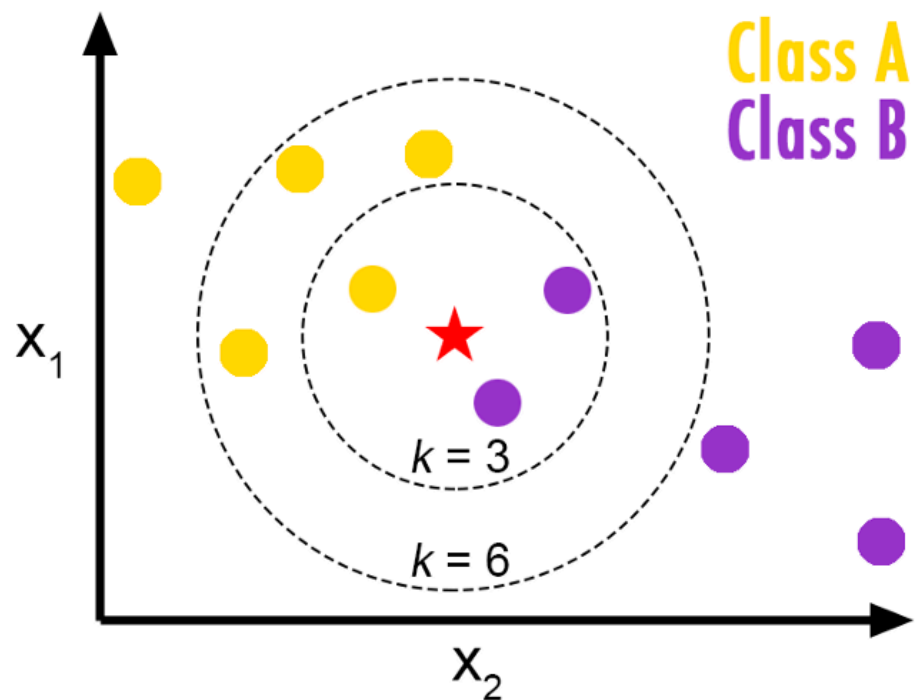
在进行模型训练时，对于具有 n 个类别的多元分类问题，One-vs-all算法会为每个类别构建一个二元分类模型。在每个二元分类模型中，将一个类别作为正例，其余 $n-1$ 个类别作为反例，然后使用二元分类算法进行训练。按照这种方法，总共可以构建 n 个二元分类模型。

如果样本 x_i 只在第 k 个二元分类模型中被预测为正例，那么类别标签的预测值就为 k 。若样本 x_i 同时在多个二元分类模型中被预测为正例，则通常考虑多个二元分类模型的预测置信度，并且选择置信度最高的类别标签作为最终预测结果。

k近邻算法 (k-Nearest Neighbor, KNN)

如果一个样本在特征空间中最邻近的k个样本的大多数属于某一个类别，则该样本也属于这个类别。

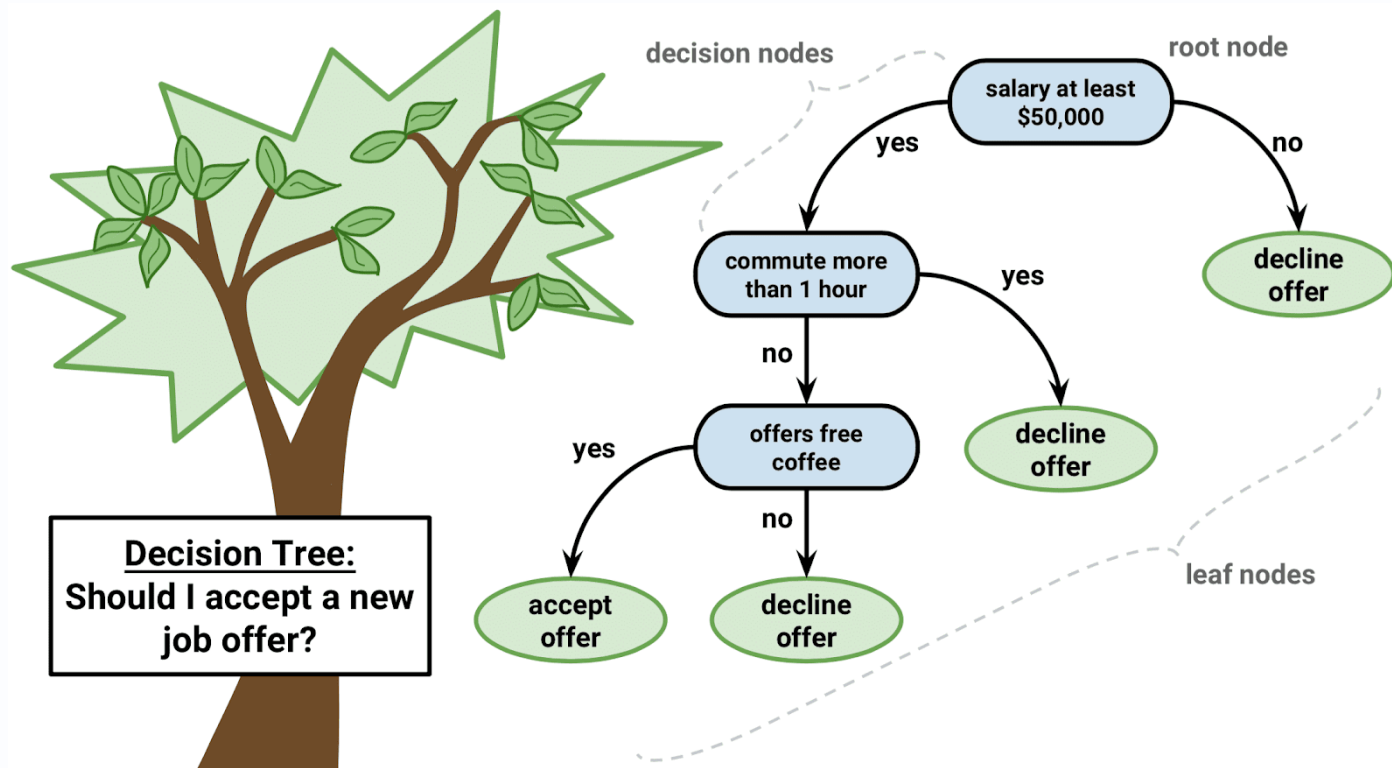
物以类聚，人以群分。



K如何取值？如何度量距离？优缺点？

决策树 (Decision Tree)

通过树状结构表示决策过程，每个内部节点代表一个特征的测试，每个分支代表测试结果，每个叶节点代表一个类别或值。决策树的核心思想是递归地将数据集划分为更小的子集，直至满足停止条件。



分类标准 (信息增益/信息增益率/基尼指数)、算法 (ID3/C4.5/CART)、剪枝 (预剪枝和后剪枝)

常见的集成学习有三种：Bagging，Boosting 和 Stacking。

树的横向集成——随机森林（Random Forests）

基于**Bagging**与特征随机性，从不同数据子集和特征子集训练每棵树，通过构建多棵**相互独立的决策树**并将它们的预测结果进行投票（分类）或取平均（回归），来提升模型的准确率与泛化能力。

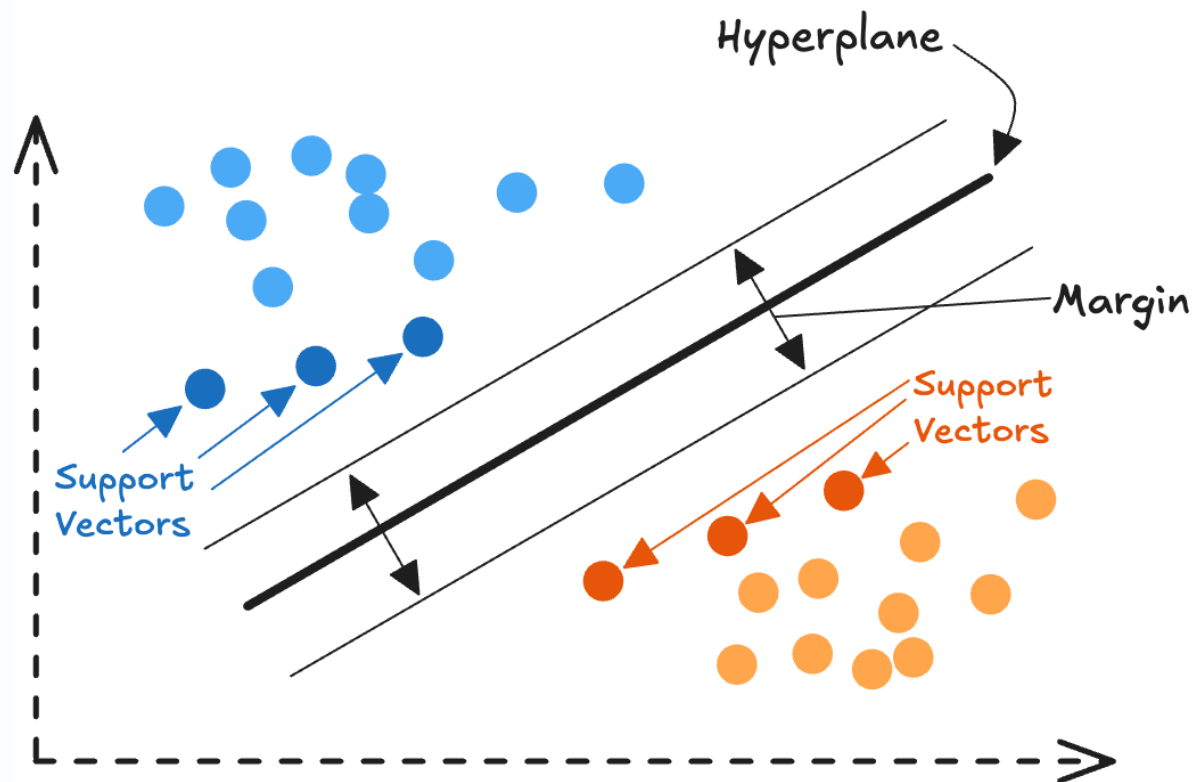
树的纵向集成——梯度提升树（Gradient Boosting Decision Tree, GBDT）

基于**Boosting**，通过多轮迭代将多个弱学习器（通常是浅层决策树）组合成一个强学习器，**逐步拟合损失函数的负梯度**，不断修正前一轮模型的误差，从而提升整体预测性能。

AdaBoost、XGBoost.....

支持向量机 (SVM, Support Vector Machine)

在特征空间中寻找一个**最优超平面**，不仅能正确分开不同类别，还能**最大化分类间隔**，以提升泛化能力。



非线性可分怎么办? **升维!**

模型选择

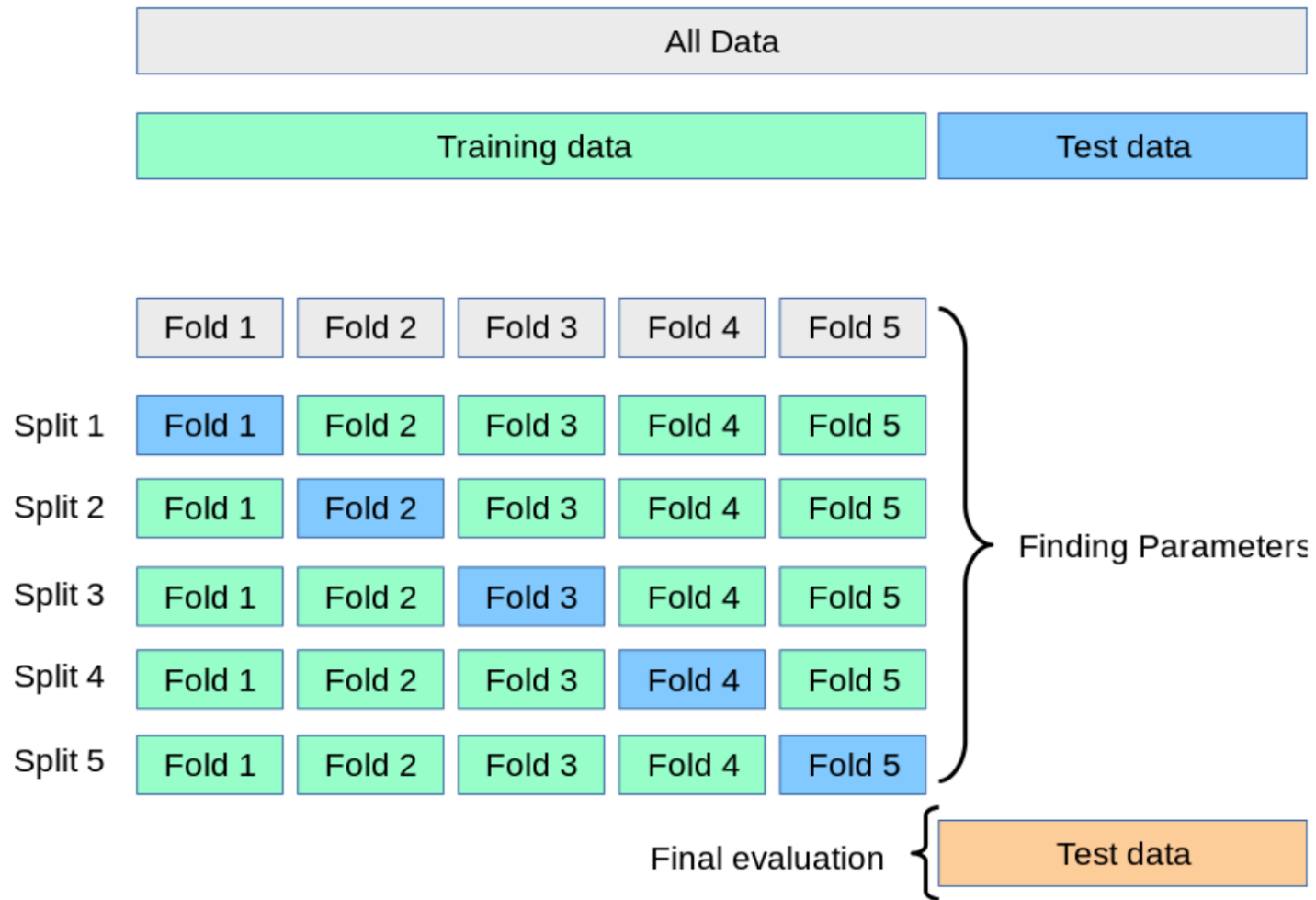
1. 如何获得测试结果? → 评估方法
2. 如何评估性能优劣? → 性能度量
3. 如何判断实质差别? → 比较检验

评估方法

关键在于怎么获得“测试集”：**测试集应该与训练集“互斥”**。

常见方法：

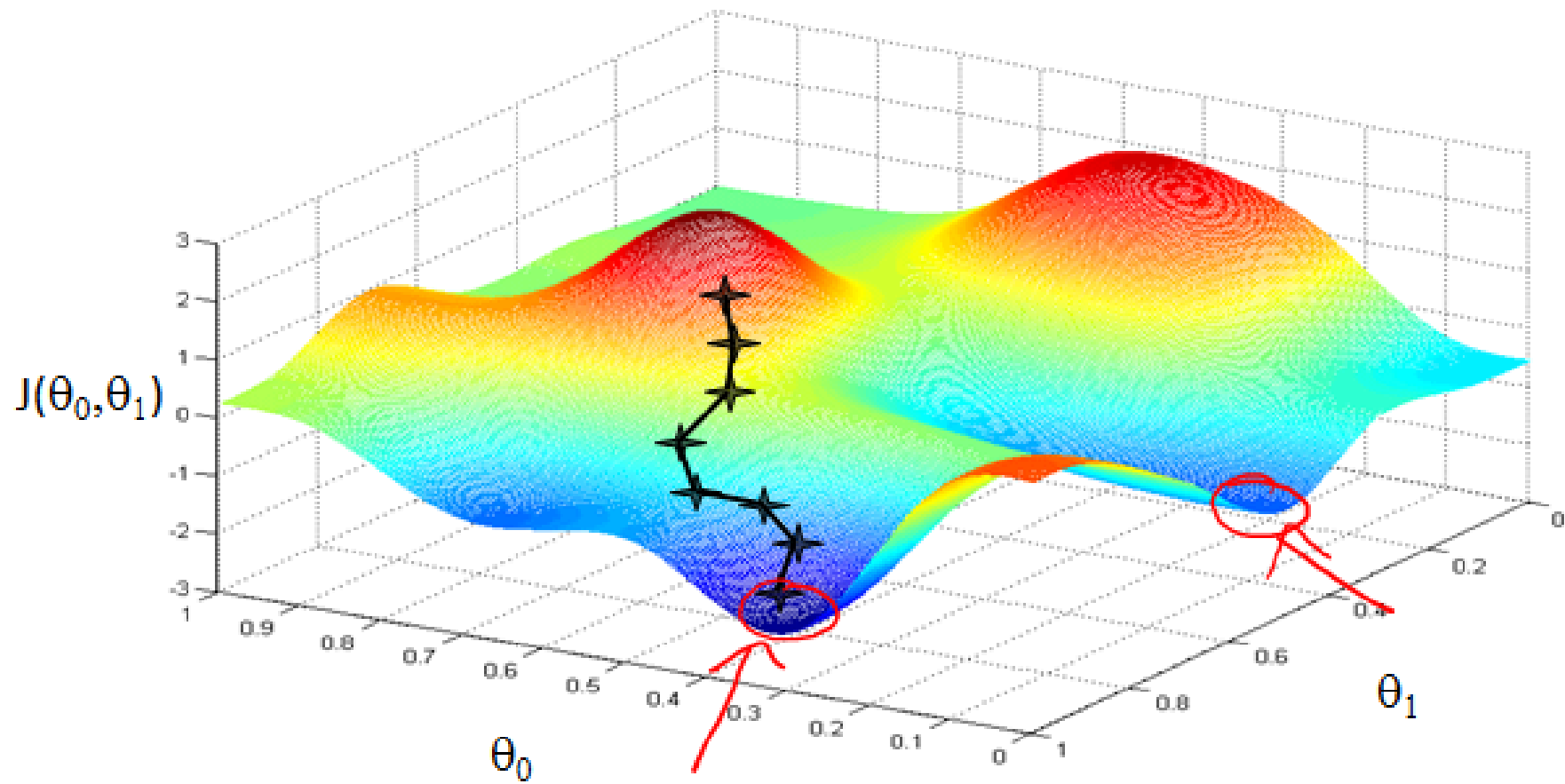
- 留出法 (hold-out)：将数据集划分为训练集和测试集两个互斥的子集，采用测试集评估模型训练结果。
- 交叉验证法 (cross validation)：将数据集划分为n个规模相同且互斥的子集，每次以其中一个子集作为测试集，其余全部作为训练集。通过重复n次训练和测试，最终使用n个评估指标的均值衡量模型泛化能力。
- 自助法 (bootstrap)：从原始数据集中对样本进行有放回抽样，得到含有m个样本的训练集，剩余未抽取的样本则作为测试集。



算法调参

- 算法的参数：一般由人工设定，亦称“超参数”
- 模型的参数：一般由学习确定
- 调参过程相似：先产生若干模型，然后基于某种评估方法进行选择
- 参数调得好不好对性能往往对最终性能有关键影响

可以采用**随机搜索**和**网格搜索**模型选择器在超参数取值空间进行遍历搜寻，并通过交叉验证方法比较各种参数组合下的模型得分，从而获得最优超参数组合。



性能度量

准确率 (Accuracy) = $(TP + TN) / (TP + TN + FP + FN)$

精确率/查准率 (Precision) = $TP / (TP + FP)$

召回率/查全率/灵敏度 (Recall/Sensitivity) = $TP / (TP + FN)$

特异度 (Specificity) = $TN / (FP + TN)$

F1-score:

$$F_1 = \frac{2}{\left(\frac{1}{precision} + \frac{1}{recall}\right)} = 2 \times \frac{precision \times recall}{precision + recall}$$

ROC曲线和AUC

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

举例：

根据是否含关键词与是否能反应相关能力可将“句子-标签”对分成四类：

1. 含有关键词且句子可以反应相关能力
2. 含有关键词但句子不能反应相关能力
3. 不含关键词且句子没有反应相关能力
4. 不含关键词但句子能够反应相关能力

实际操作中还可能面临含有一种能力的关键词但句子实际反应的是另一种能力的情况。

查全率和查准率从不同角度衡量了预测效果，最为理想的情况是同时提升查全率和查准率两个指标，然而这两个指标之间存在此消彼长关系，现实中常常面临“鱼和熊掌不可兼得”的困境。采用F1-score可兼顾两个指标的影响，数学上其实是查准率与查全率的**调和平均数**。

真阳率（True Positive Rate, TPR）即查全率，表示针对所有真实阳性样本，模型有多大概率对它们的预测结果也是阳性；**假阳率**（False Positive Rate, FPR）即1-特异度，表示针对所有真实阴性样本，模型有多大概率对它们的预测结果却是阳性。

显然，对于一份预测结果，其真阳率越高越好，而假阳率越低越好。与查准率与查全率往往呈反向变动不一样，真伪阳性率一般同方向变动。换言之，**提高阳性预测的正确程度必须以提高阴性预测的错误程度为代价**；反之亦然。

