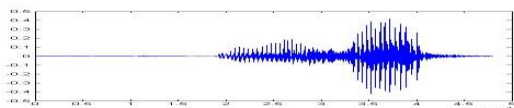


机器学习概述

机器学习算法之一

机器学习 \approx 构建一个映射函数


- 语音识别

$$f(\text{  }) = \text{“你好”}$$

- 图像识别

$$f(\text{  }) = \text{“猫”}$$

- 围棋

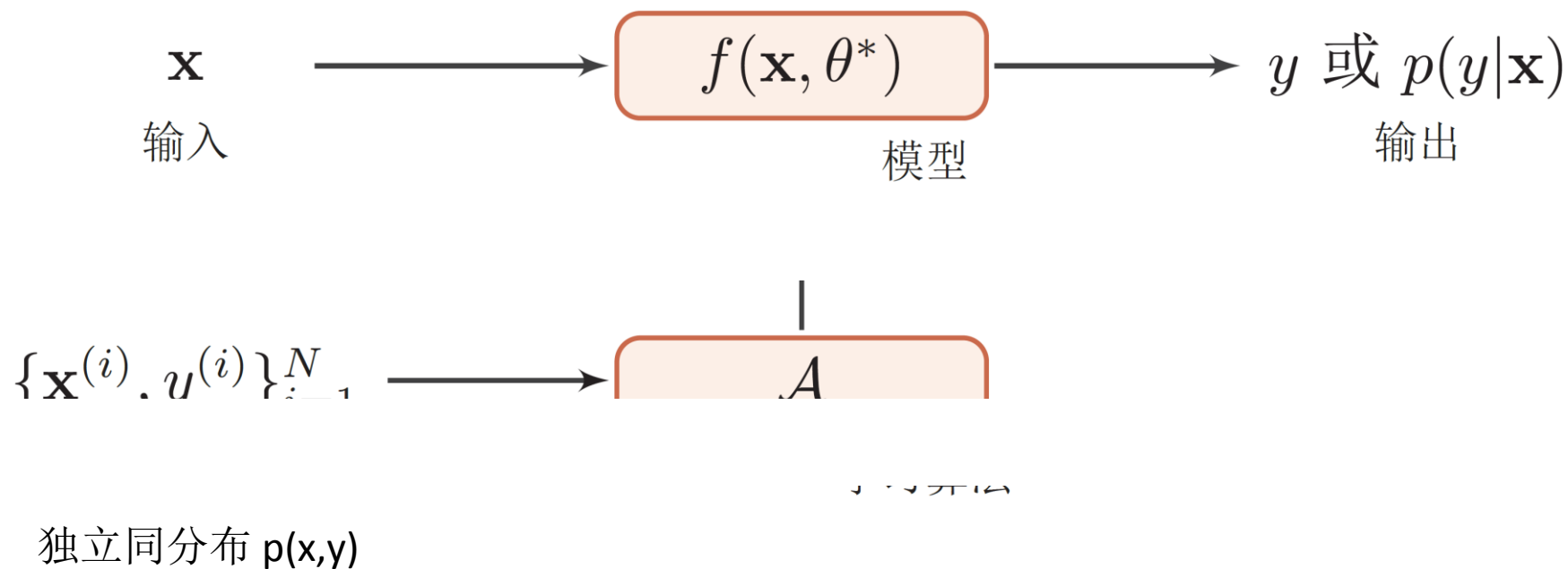
$$f(\text{  }) = \text{“5-5”} \quad (\text{落子位置})$$

- 对话系统

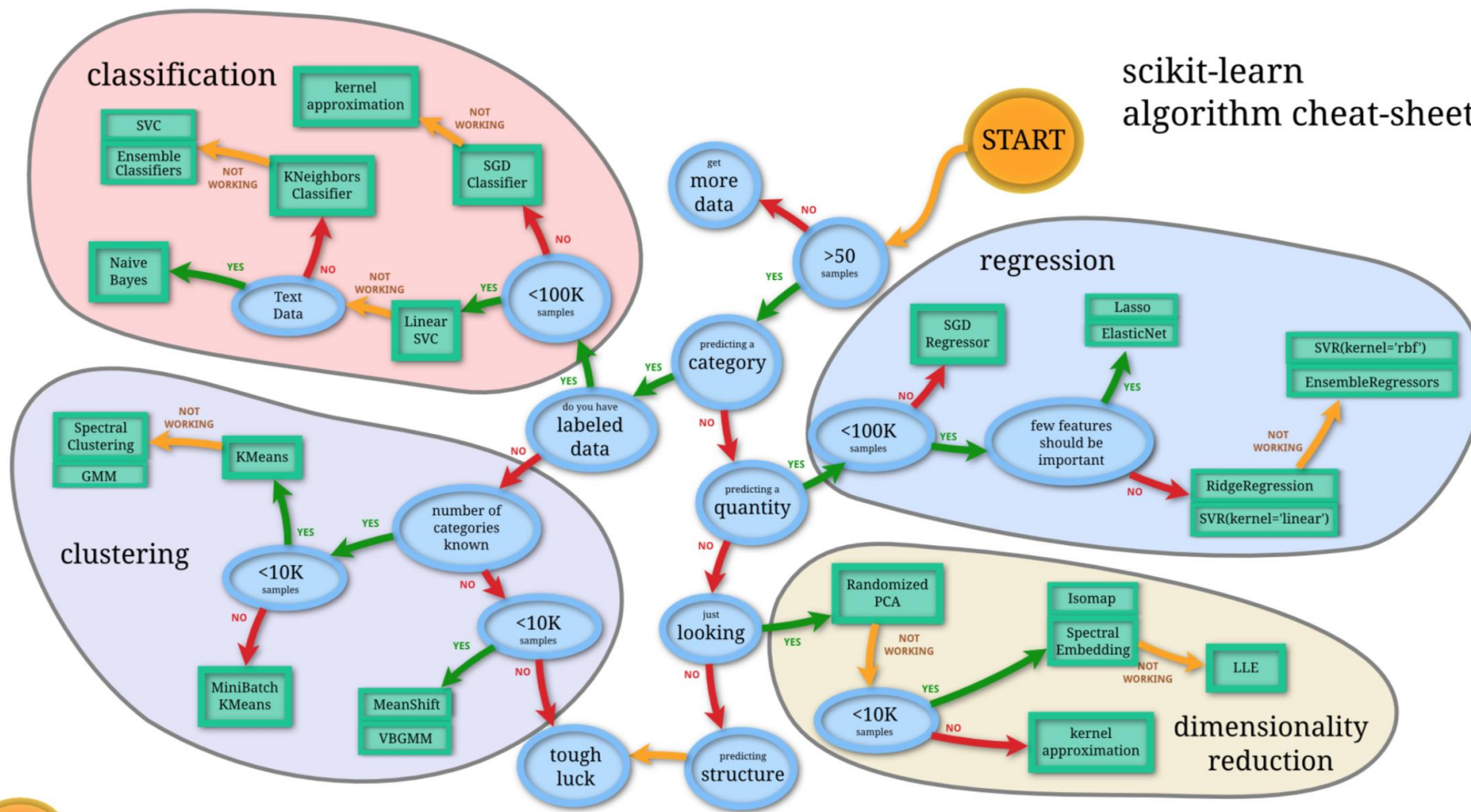
$$f(\text{ “你好” 用户输入 }) = \text{ “今天天气真不错” 机器 }$$

什么是机器学习？

机器学习：从数据中获得决策（预测）函数使得机器可以根据数据进行自动学习，通过算法使得机器能从大量历史数据中学习规律从而对新的样本做决策。



scikit-learn
algorithm cheat-sheet



机器学习的三要素

• 模型

- 线性方法: $f(\mathbf{x}, \theta) = \mathbf{w}^T \mathbf{x} + b$
- 广义线性方法: $f(\mathbf{x}, \theta) = \mathbf{w}^T \phi(\mathbf{x}) + b$
 - 如果 $\phi(\mathbf{x})$ 为可学习的非线性基函数, $f(\mathbf{x}, \theta)$ 就等价于神经网络。

• 学习准则

- 期望风险

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)} [\mathcal{L}(f(\mathbf{x}), y)],$$

• 优化

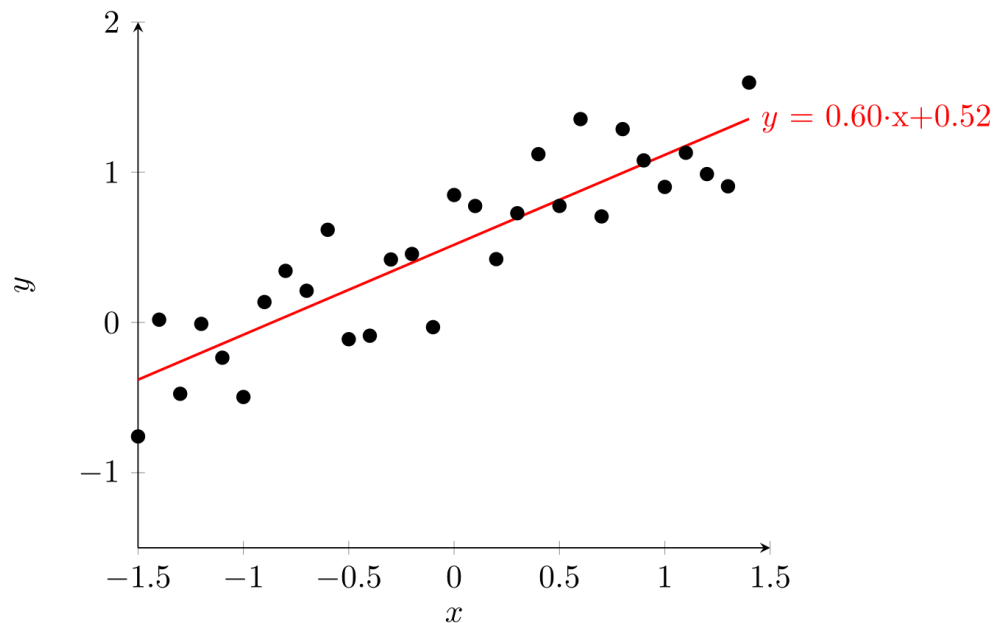
- 梯度下降

线性回归 (Linear Regression)

- 模型:

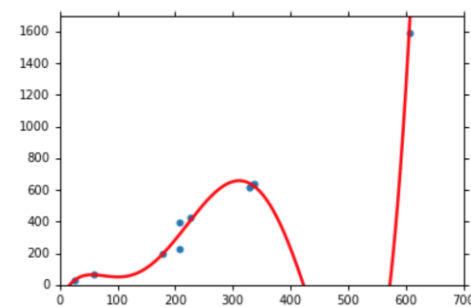
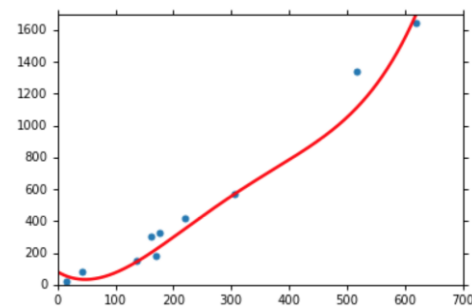
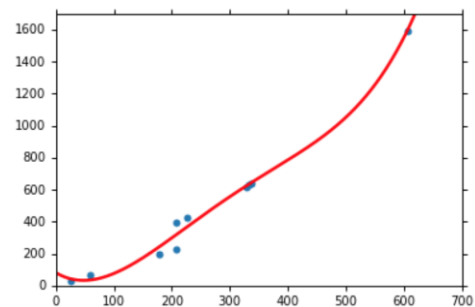
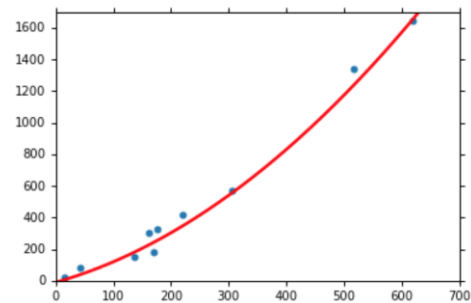
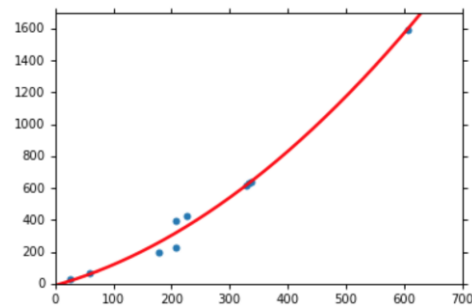
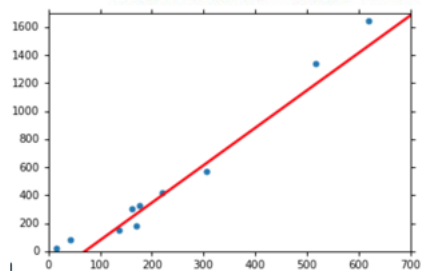
$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T \mathbf{x} + b$$

期望风险:



$$\begin{aligned} \mathcal{R}(\mathbf{w}) &= \sum_{n=1}^N \mathcal{L}(y^{(n)}, f(\mathbf{x}^{(n)}, \mathbf{w})) \\ &= \frac{1}{2} \sum_{n=1}^N \left(y^{(n)} - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 \\ &= \frac{1}{2} \|\mathbf{y} - X^T \mathbf{w}\|^2, \end{aligned}$$

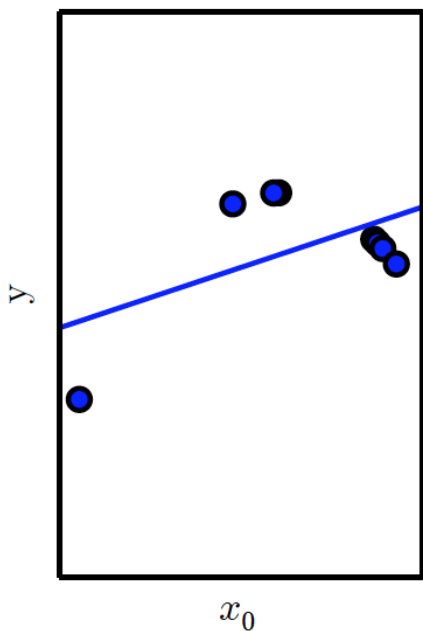
期望误差越小越好？



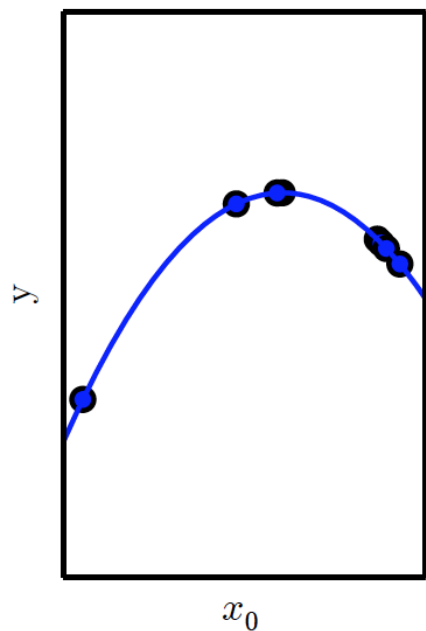
过拟合与欠拟合

- 欠拟合是指模型不能在训练集上获得足够小的误差（训练误差太大），
- 过拟合是指训练误差和泛化误差的差距太大。

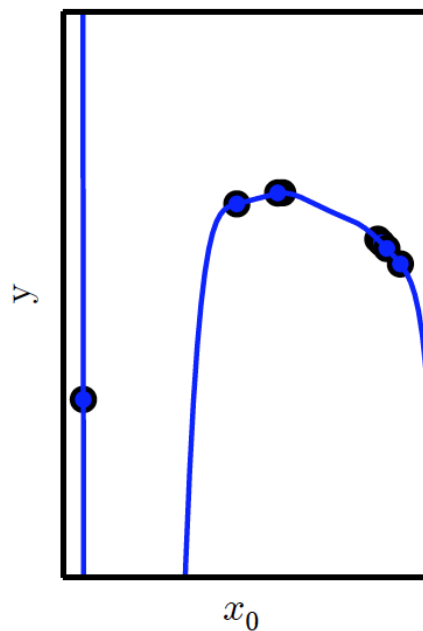
Underfitting



Appropriate capacity



Overfitting



泛化错误可以衡量一个机器学习模型是否可以很好地泛化到未知数据。机器学习的目标是减少泛化错误。

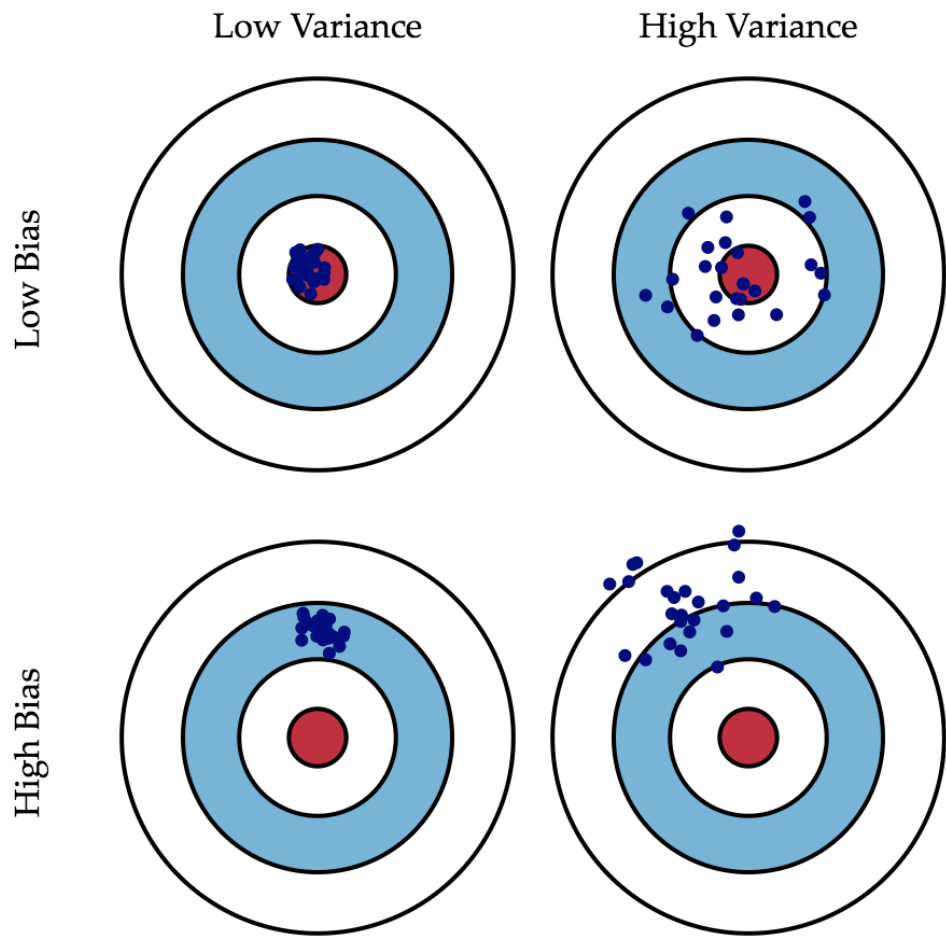
泛化错误一般表现为一个模型在训练集和测试集上错误率的。

对于单个样本 \mathbf{x} ，不同训练集 \mathcal{D} 得到模型 $f_{\mathcal{D}}(\mathbf{x})$ 和最优模型 $f^*(\mathbf{x})$ 的上的期望差距为

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}} \left[\left(f_{\mathcal{D}}(\mathbf{x}) - f^*(\mathbf{x}) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[\left(f_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})] + \mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})] - f^*(\mathbf{x}) \right)^2 \right] \\ &= \underbrace{\left(\mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})] - f^*(\mathbf{x}) \right)^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}} \left[\left(f_{\mathcal{D}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(\mathbf{x})] \right)^2 \right]}_{\text{variance}}. \end{aligned}$$

$$\mathcal{R}(f) = (\text{bias})^2 + \text{variance} + \varepsilon.$$

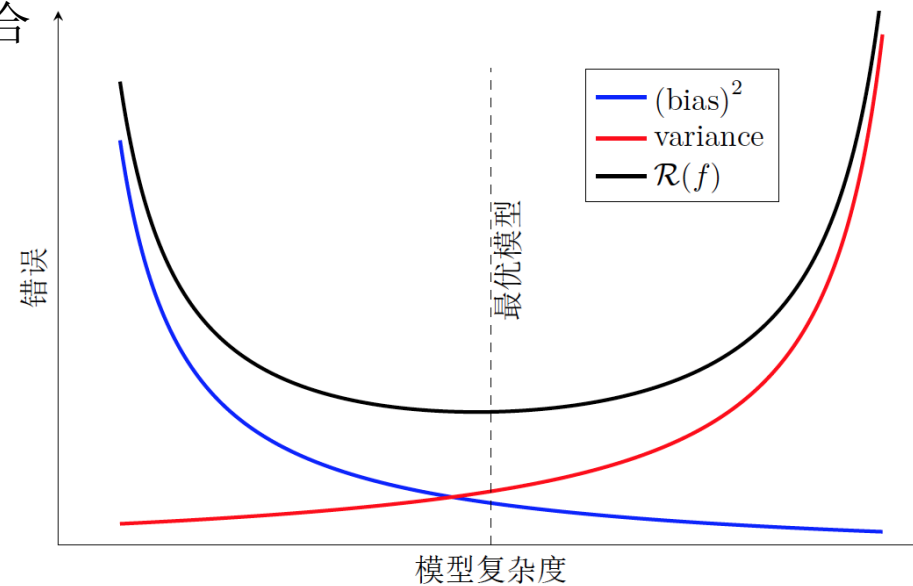
过拟合与欠拟合



高偏差对应欠拟合

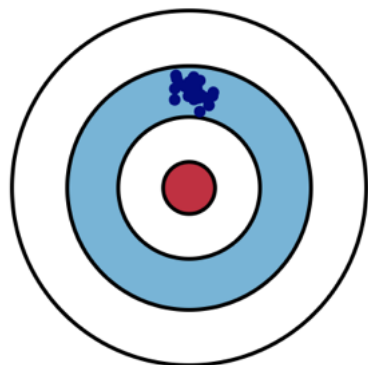
高方差对应过拟合

模型的容量是指其拟合各种函数的能力（可以表示多少种曲线）。容量低的模型可能很难拟合训练集，比如上例中的用线性函数去拟合二次函数。

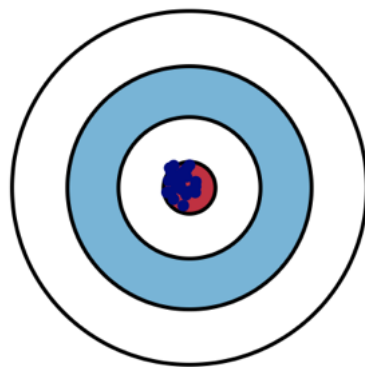
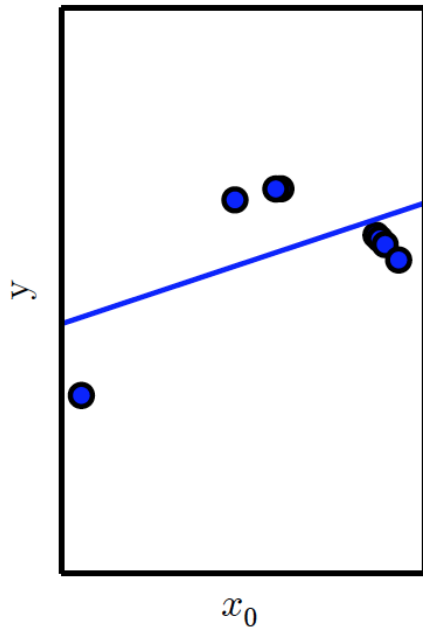


过拟合与欠拟合

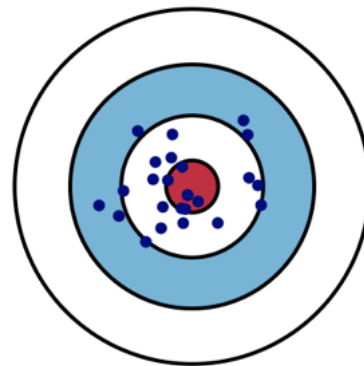
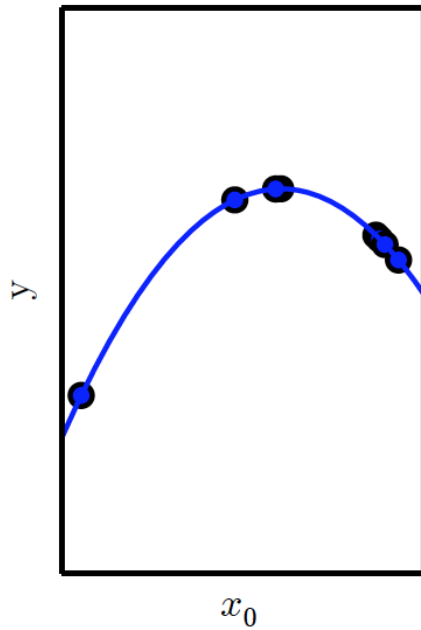
多次进行
采样建模



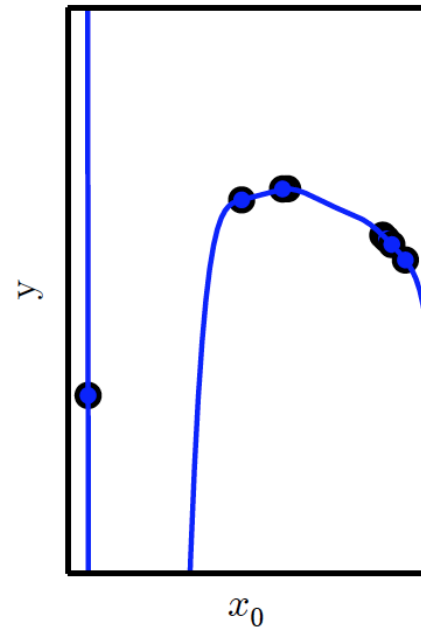
Underfitting



Appropriate capacity

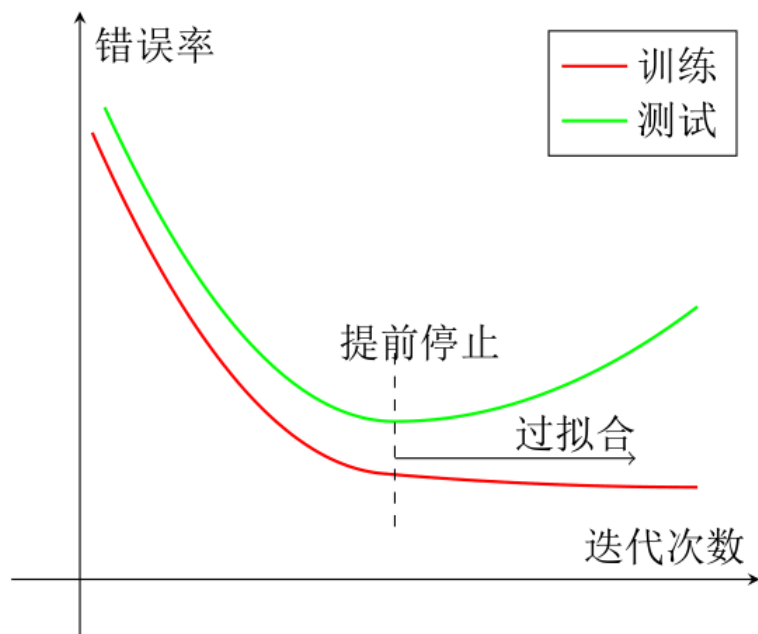


Overfitting



如何判断是否过拟合？

我们使用一个测试集（Test Dataset）来测试每一次迭代的参数在测试集上是否最优。

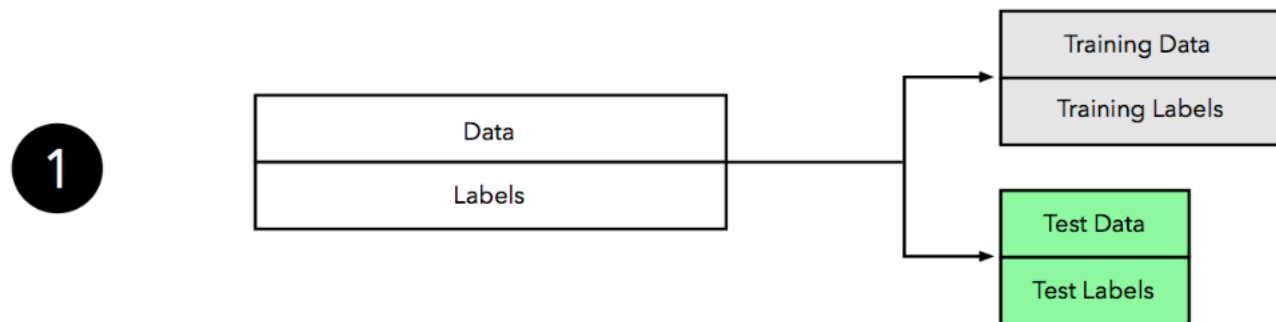


模型选择与评估

- 模型选择方法：
 - Hold Out
 - Cross Validation
 - Bootstrap
- 模型评估方法
 - Confusion Matrix
 - ROC曲线 与 AUC
 - Lift(提升)和Gain(增益)

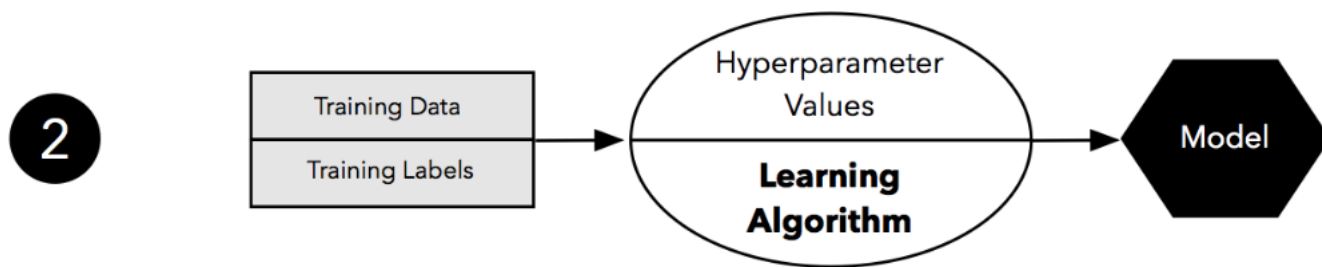
- holdout cross-validation 以及 k-fold cross-validation 都是交叉验证的方法。
- 评价模型在新的数据集上的性能。
- 平衡过拟合与欠拟合。

- holdout cross-validation 过程



划分训练集，一般训练集大概在 $\frac{2}{3}$ 左右。

- holdout 过程

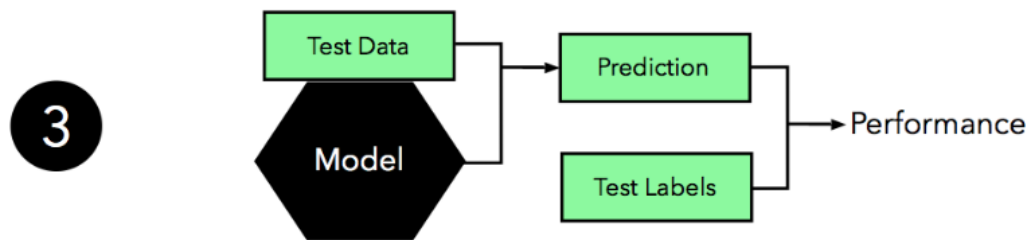


调节超参数。

超参数：机器学习模型里面的框架参数，如聚类个数，正则化系数，KNN中的K的选取，一般人工手动调节。

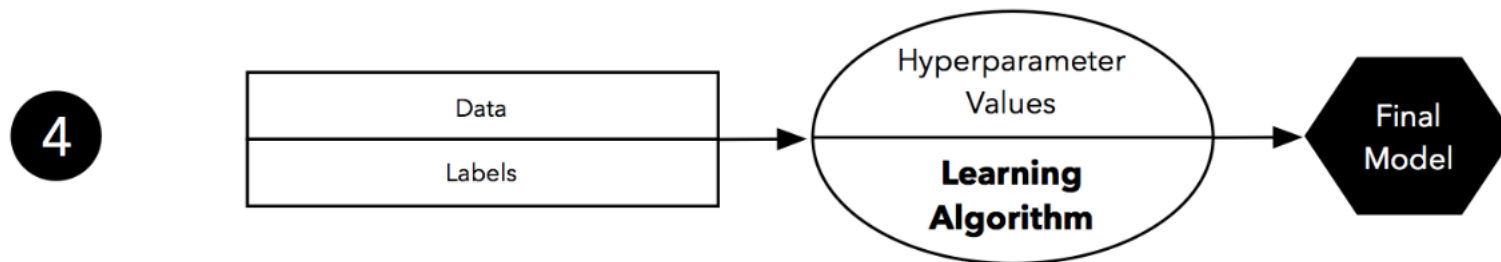
学习参数：由数据学习到的权重，不需要手动调节，由数据决定。

- holdout cross-validation 过程



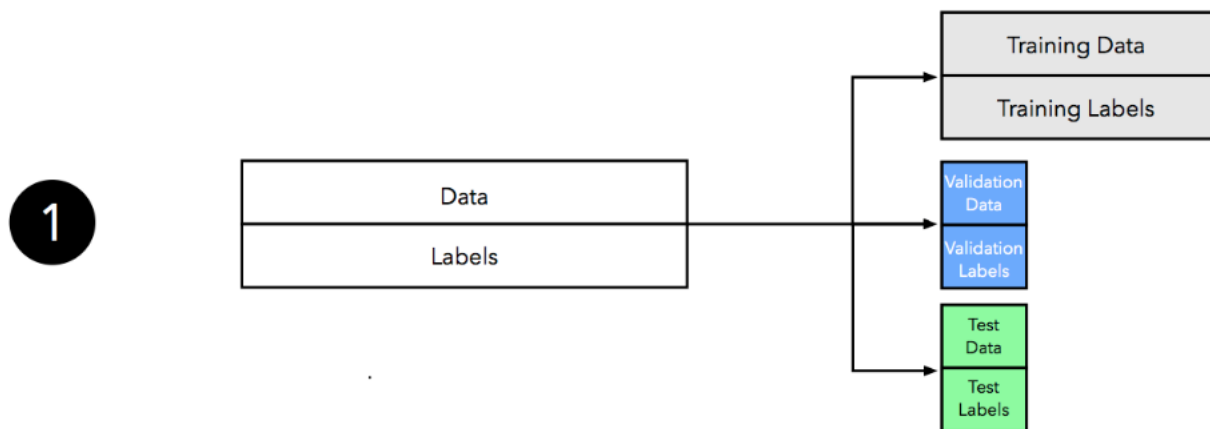
训练好的模型需要用测试集做验证。通常情况下，还需要对训练集进行评估。

- holdout cross-validation 过程



超参数确定了，还有一部分没有参与训练，利用全量数据集训练模型。得到最终的结果。

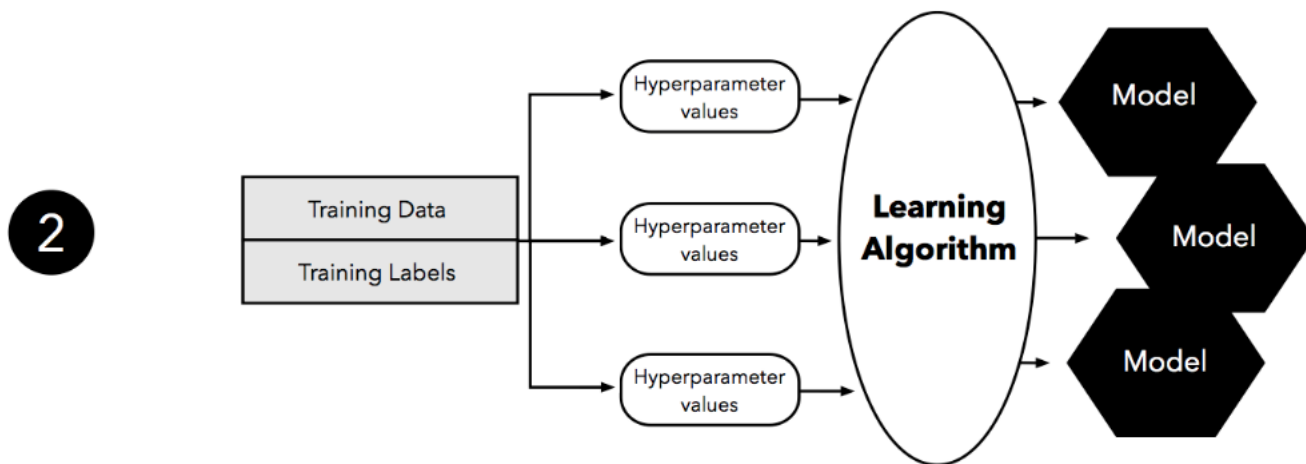
- 上述方法存在一定的偶然性，并且不能监督过程。
- 引出 three-way holdout method



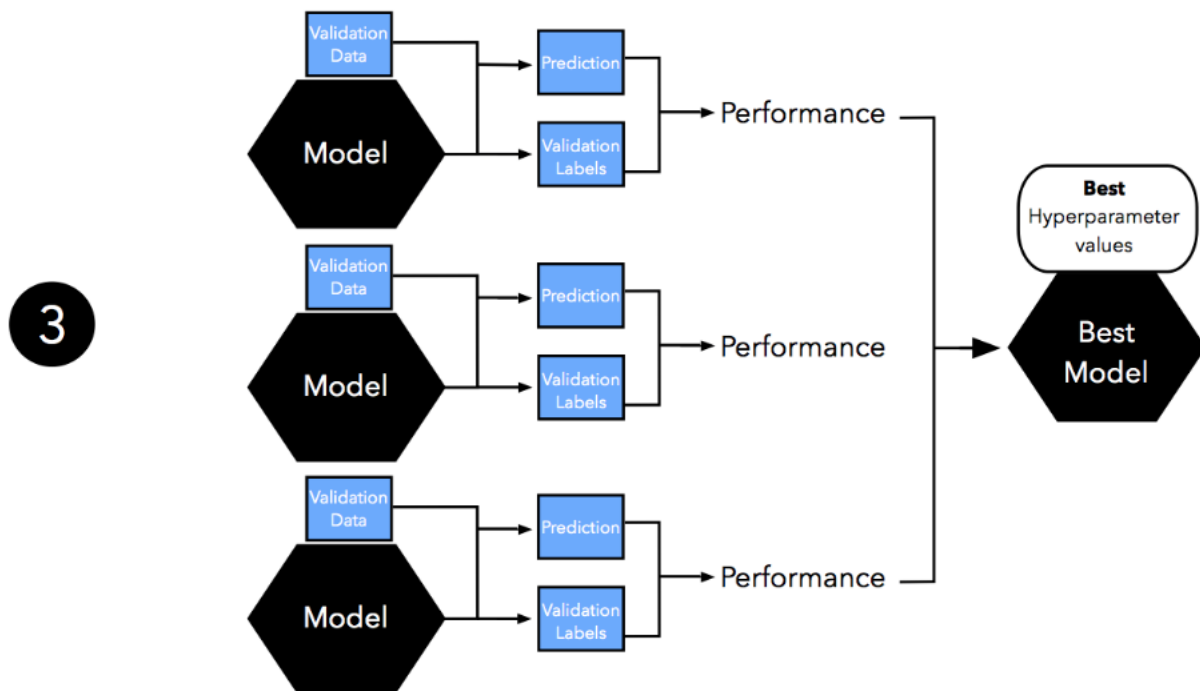
分为训练集、验证集、测试集。

测试集的作用：参与监督过程和调参指导。

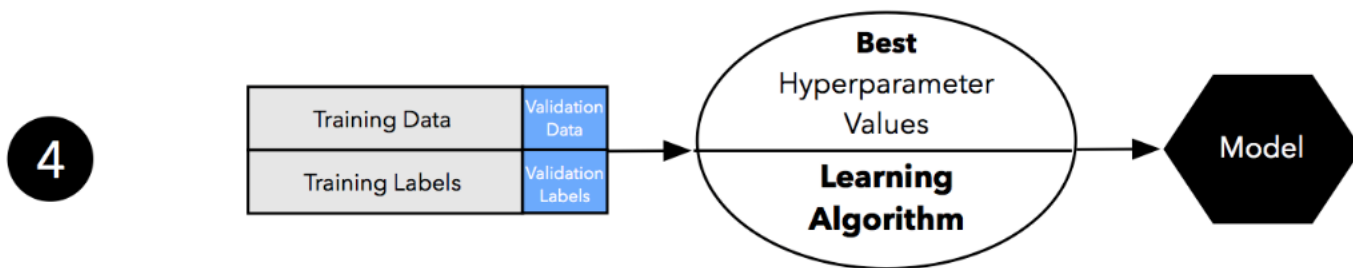
- 采用不同的超参数在训练集上训练同一模型。



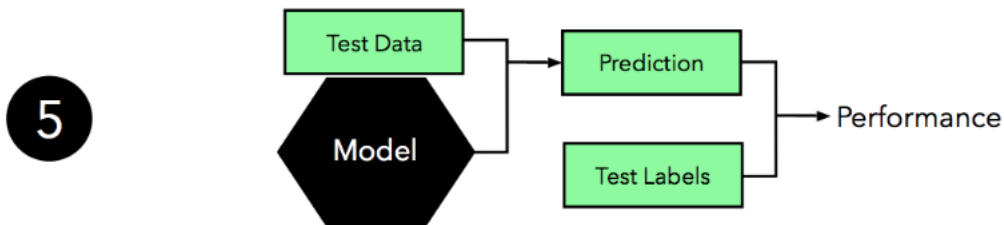
- 训练过程中利用验证集挑选最优超参数，选择最好的一个模型。



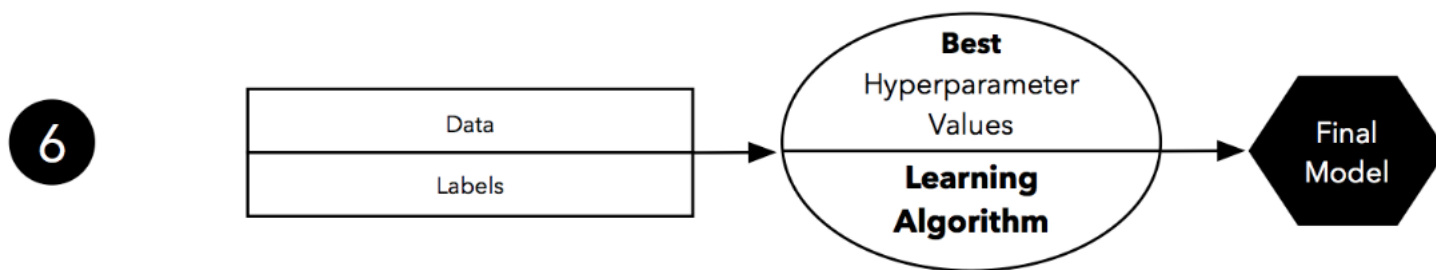
- 挑选出了最优的超参数，那么将训练集和验证集合在一起训练。



- 在测试集上的表现结果。



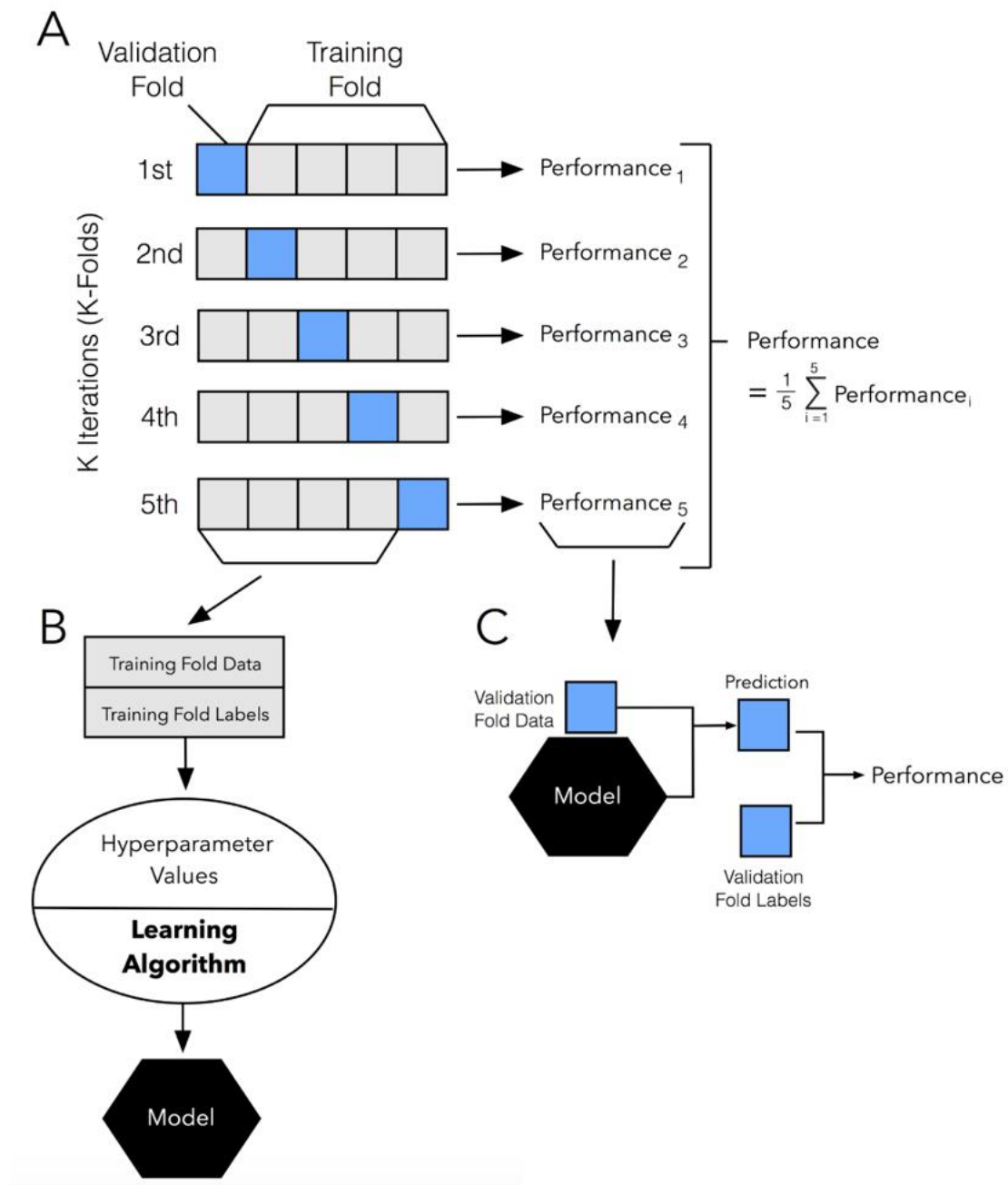
- 所有的训练数据放在一起，拿到模型中训练。



- cross-validation

K-fold 交叉验证

优点：利用全部数据集
更加鲁棒



- Bootstrap

- 自助法
- 一种重采样（Resampling）技术
- 集成学习中会用到它的思想



有放回的抽样;

Original Dataset

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------

Bootstrap 1

x_8	x_6	x_2	x_9	x_5	x_8	x_1	x_4	x_8	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_{10}
-------	-------	----------

Bootstrap 2

x_{10}	x_1	x_3	x_5	x_1	x_7	x_4	x_2	x_1	x_8
----------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_6	x_9
-------	-------

Bootstrap 3

x_6	x_5	x_4	x_1	x_2	x_4	x_2	x_6	x_9	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_8	x_{10}
-------	-------	-------	----------

Training Sets

Test Sets

Training Sets

Test Sets

是否有的样本压根没有采样到

- 这个概率是可以计算的:
- 有n个样本，每个样本的取到的概率是1/n

$$P(\text{not chosen}) = \left(1 - \frac{1}{n}\right)^n$$

如果n取 ∞ ，则极限为1/e, ~0.368

Random Forest

通过常用极限推导而来 $\lim_{x \rightarrow 0} (1+x)^{\frac{1}{x}} = e$

- 天然的划分训练集和测试集。
- 实际上，Random Forest 模型验证就是采用这种方法。

模型评估

- Confusion Matrix
 - 对于二分类问题，可将样例根据其真实类别与学习期预测类别的组合划分为真正例(True Positive),假正例(False Positive)，真反例(True Negative)，假反例(False Negative)四种情形，四种情形组成的混淆矩阵如下：

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

模型评估

准确率(Accuracy)

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{TN} + \text{FN})$$

精准率(Precision)

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

召回率(Recall)

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

F值: F-score是Precision和Recall加权调和平均数, 并假设两者一样重要

$$\text{F1 Score} = (2\text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision})$$

模型评估

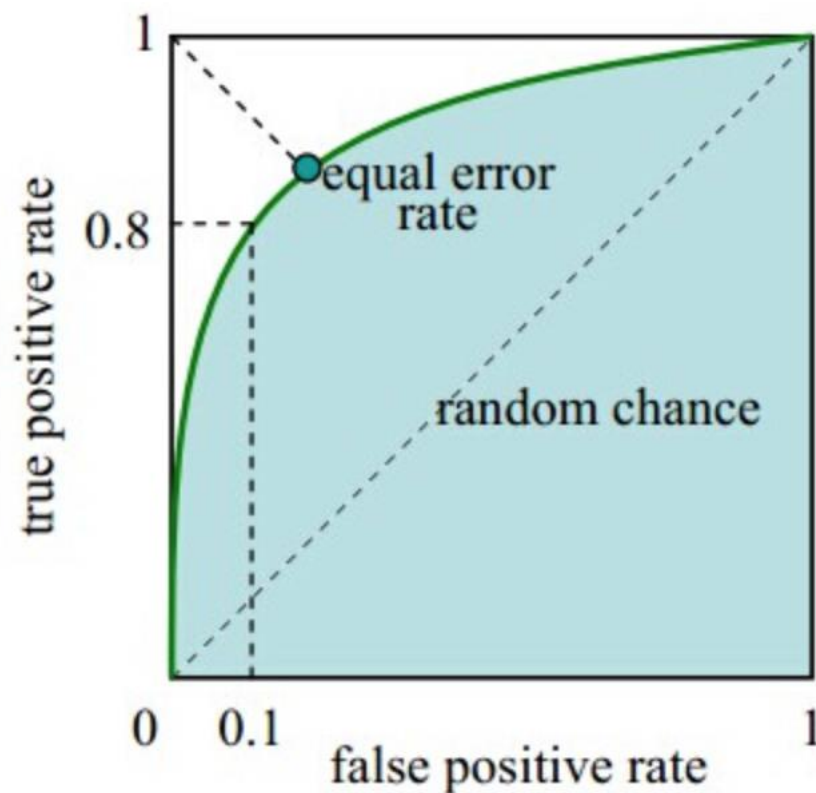
- 某池塘有1400条鲤鱼，300只虾，300只鳖。现在以捕鲤鱼为目的。撒一大网，逮着了700条鲤鱼，200只虾，100只鳖。那么，精确率和召回率分别为多少？
- 精确率 = $700 / (700 + 200 + 100)$
- 召回率 = $700 / 1400$

模型评估

- ROC 与AUC

- ROC

- AUC



ROC全称是“受试者工作特征”（Receiver Operating Characteristic）。ROC曲线的面积就是AUC（Area Under the Curve）。AUC用于衡量“二分类问题”机器学习算法性能（泛化能力）。

模型评估

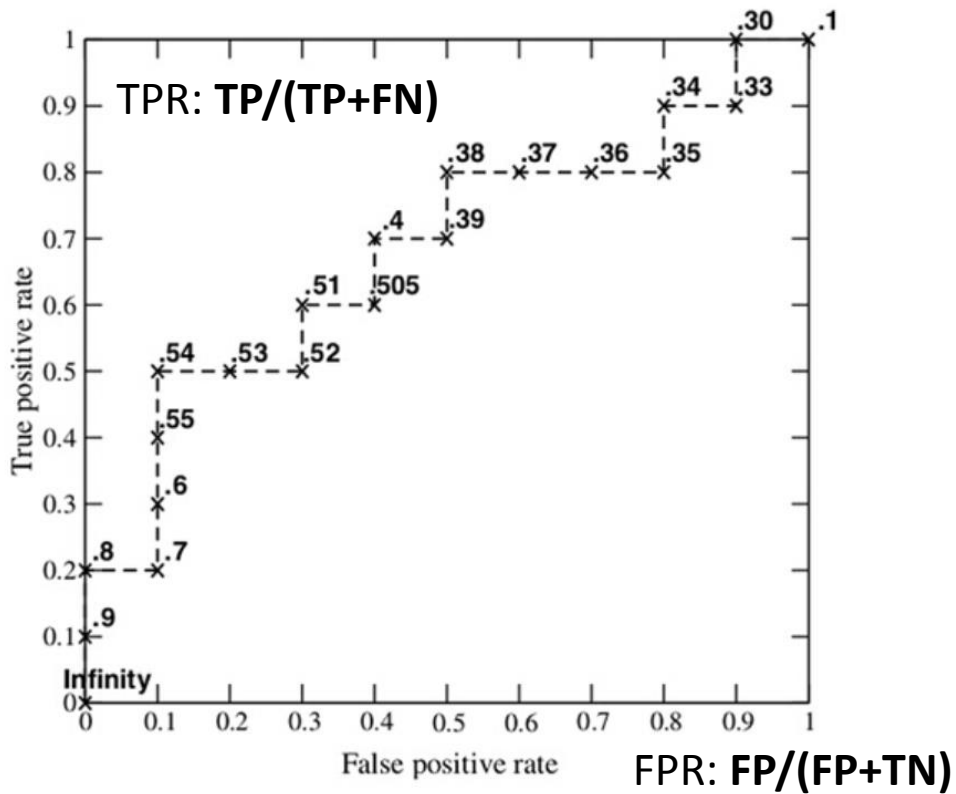
- 如何画Roc
- 真正类率(True Postive Rate)
- TPR: $TP/(TP+FN)$
- 负正类率(False Postive Rate)
- FPR: $FP/(FP+TN)$

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

模型评估

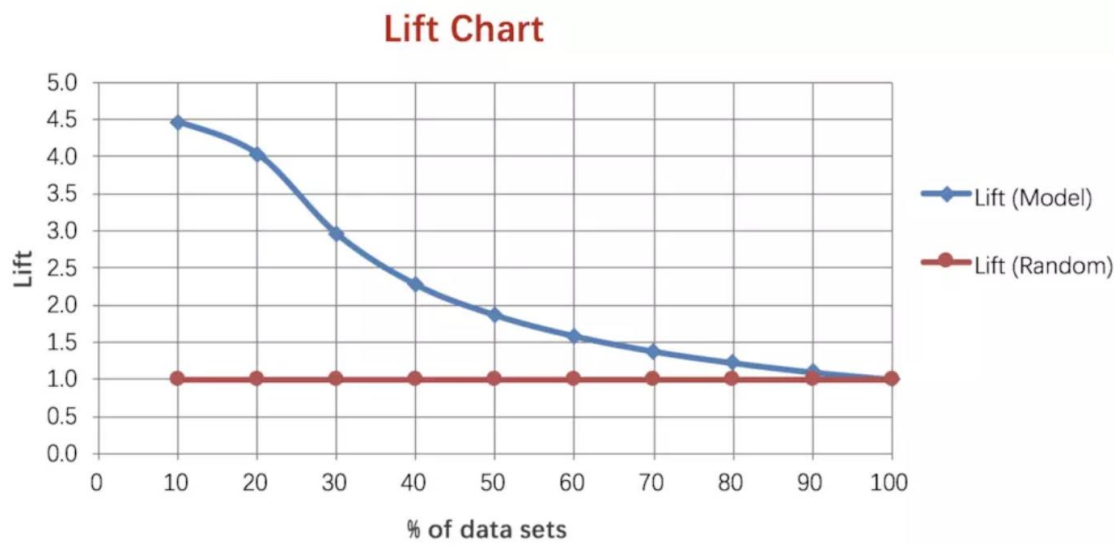
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN

- Lift(提升)和Gain(增益)
 - $\text{Lift} = [\text{TP}/(\text{TP}+\text{FP})] / [(\text{TP}+\text{FN})/(\text{TP}+\text{FP}+\text{FN}+\text{TN})]$

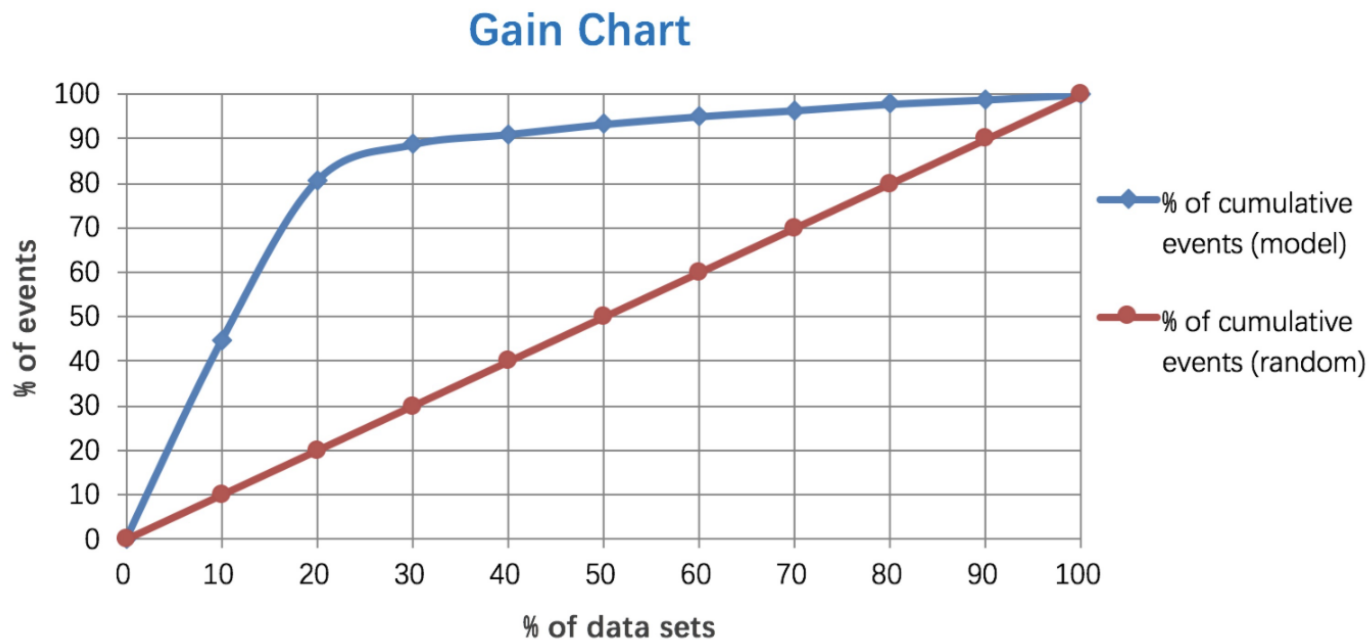
真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN



纵坐标是lift，横坐标是正例集百分比。

- Lift(提升)和Gain(增益)
- **Gain**= $TP / (TP + FP)$

真实情况	预测结果	
	正例	反例
正例	TP	FN
反例	FP	TN



纵坐标Gain，横坐标是正例集百分比。

- 多分类
 - 宏平均（macro-average）和微平均（micro-average）

$$Macro_P = \frac{1}{n} \sum_{i=1}^n P_i$$

$$Micro_P = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i}$$

$$Macro_R = \frac{1}{n} \sum_{i=1}^n R_i$$

$$Micro_P = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i}$$

$$Macro_F = \frac{1}{n} \sum_{i=1}^n F_i$$

$$Macro_F = \frac{2 * Macro_P * Macro_R}{Macro_P + Macro_R}$$

- 多分类
 - 宏平均（macro-average）和微平均（micro-average）

类别	TP	FP	P
1	3	5	0.6
2	2	3	0.67
3	1	2	0.5
total	6	10	

- $\text{Macro} = (0.6+0.67+0.5)/3=0.59$
- $\text{Micro} = (3+2+1)/(5+3+2)=0.6$