AI机器学习
回归算法精讲

日期：2021.01

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

目 录 / CONTENTS

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

PART 01

回归模型以及应用场景

简介 / INTRODUCTION

什么是回归模型

回归模型是一种预测性的建模技术，它研究的是因变量（Y）和自变量（X）之间的关系。这种技术通常用于预测分析，时间序列模型以及发现变量之间的因果关系。

常用的回归模型有线性回归、多项式回归、逻辑回归、岭回归等

回归算法的应用场景

输出标量，预测值

- 股票预测

f （  ） = 明日股价、指数

- 自动驾驶汽车

f （  ） = 方向盘角度

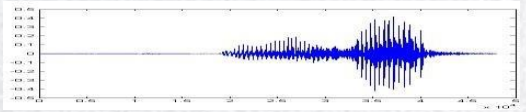
- 推荐算法

f（使用者A 商品B） = 购买可能性

AI算法原理

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

AI自己找函数

- 语音识别 $f($  $)=$ “你好啊，吃了吗”
- 图像识别 $f($  $)=$ “猫”
- 围棋 $f($  $)=$ “5-5”
(下一步)
- 对话机器人 $f($ “你吃了吗?” $)$ “你猜.”
=
(用户问)
(机器人自动回答
)

数据构成

A 日常

根据训练数据的已知X和Y，找到模型参数

✓ **训练集 Training Set Data**

根据训练好的模型，在测试集上做推断，验证

✓ **测试集 Testing Set Data**

B 用于竞赛等场景

根据训练数据的已知X和Y，找到模型参数

✓ **训练集 Training Set Data**

根据训练好的模型，在验证集上做推断，验证

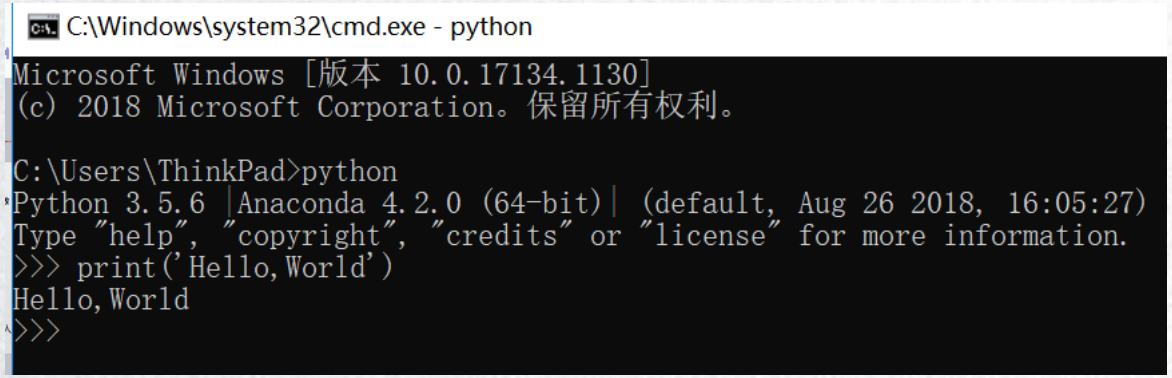
✓ **验证集　Validation Set　Data**

数据验证，打分

✓ **测试集 Testing Set Data**

人工智能开发IDE工具

A text IDE



B Jupyter notebook

C PyCharm

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

PART 02

线性回归

什么是线性回归

线性回归是利用数理统计中回归分析，来确定两种

或两种以上变量间相互依赖的定量关系的一种统计

分析方法，运用十分广泛，是回归算法的基础。

线性回归模型

$y = w \cdot x_i + b$

一组函数

模型

$f_1, f_2 \ldots\ldots$

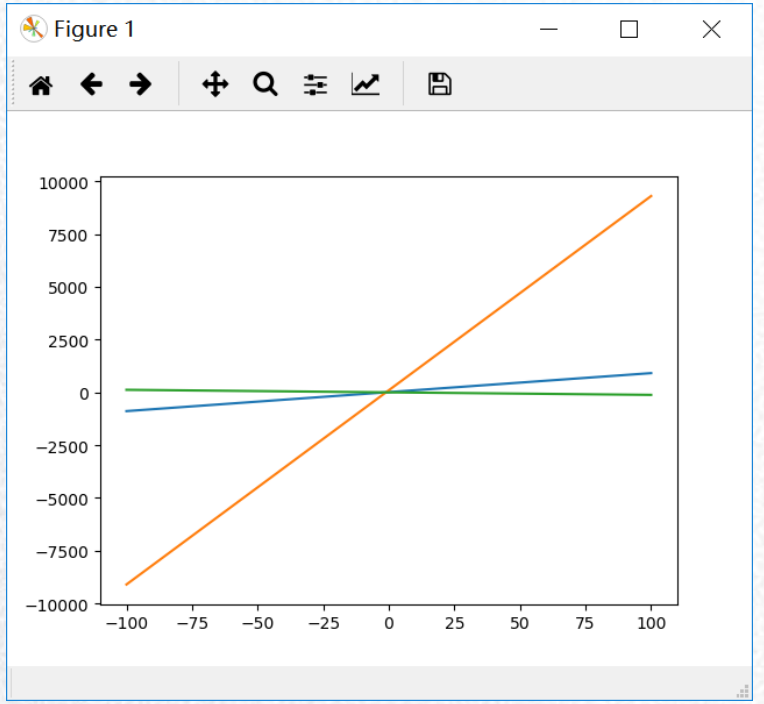
w 和 b 是参数
(可以是任何值)

$f_1: y = 10.0 + 9.0 \cdot x_i$

$f_2: y = 9.8 + 9.2 \cdot x$

$f_3: y = -0.8 - 1.2 \cdot x_i$

……



x_i: 特征

线性模型: $y = w_i x_i + b$

w_i: 权重, b: 偏置

疫情数据分析

E3 0

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	iso_code	continent	location	date	total_ca	new_case	new_case	total_de	new_deat	new_deat	total_ca	new_case	new_case
14	CHN	Asia	China	2019/12/31	27	27		0	0		0.019	0.019	
15	CHN	Asia	China	2020/1/1	27	0		0	0		0.019	0	
16	CHN	Asia	China	2020/1/2	27	0		0	0		0.019	0	
17	CHN	Asia	China	2020/1/3	44	17		0	0		0.031	0.012	
18	CHN	Asia	China	2020/1/4	44	0		0	0		0.031	0	
19	CHN	Asia	China	2020/1/5	59	15		0	0		0.041	0.01	
20	CHN	Asia	China	2020/1/6	59	0	8.429	0	0	0	0.041	0	0.006
21	CHN	Asia	China	2020/1/7	59	0	4.571	0	0	0	0.041	0	0.003
22	CHN	Asia	China	2020/1/8	59	0	4.571	0	0	0	0.041	0	0.003
23	CHN	Asia	China	2020/1/9	59	0	4.571	0	0	0	0.041	0	0.003
24	CHN	Asia	China	2020/1/10	59	0	2.143	0	0	0	0.041	0	0.001
25	CHN	Asia	China	2020/1/11	59	0	2.143	1	1	0.143	0.041	0	0.001
26	CHN	Asia	China	2020/1/12	59	0	0	1	0	0.143	0.041	0	0
27	CHN	Asia	China	2020/1/13	59	0	0	1	0	0.143	0.041	0	0
28	CHN	Asia	China	2020/1/14	59	0	0	1	0	0.143	0.041	0	0

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

PART 03

多项式回归

什么是多项式回归

多项式回归，回归函数是回归变量多项式的回归算法。多项式回归模型是线性回归模型的一种，此时回归函数关于回归系数是线性的。由于任一函数都可以用多项式逼近，因此多项式回归有着广泛应用。

多项式回归模型

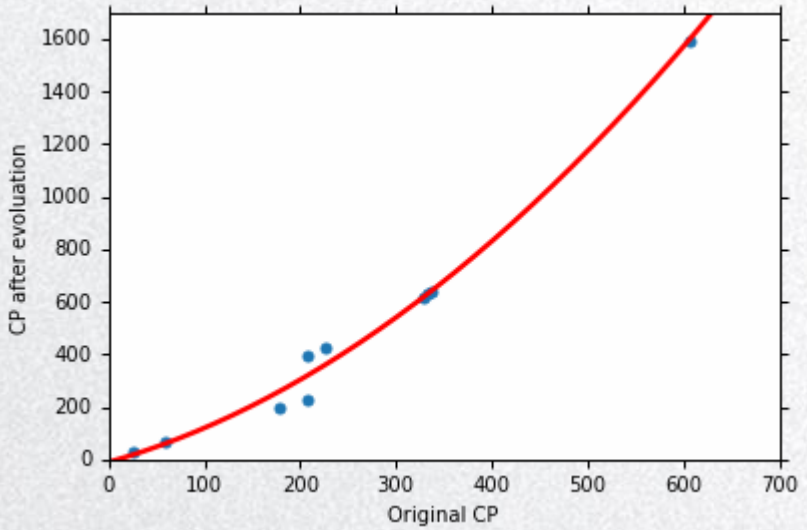
幂级数

一元N次方多项式模型表达式

$$y = a_0 \cdot x^0 + a_1 \cdot x^1 + a_2 \cdot x^2 + a_3 \cdot x^3$$

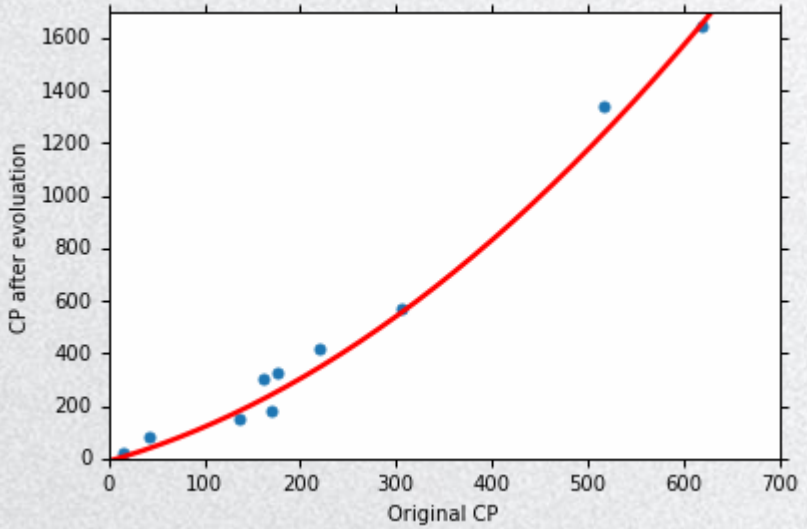
.....

$$+ a_n \cdot x^n$$



一元2次方多项式模型表达式

$$y = a_0 \cdot x^0 + a_1 \cdot x^1 + a_2 \cdot x^2$$



一元3次方多项式模型表达式

$$y = a_0 \cdot x^0 + a_1 \cdot x^1 + a_2 \cdot x^2 + a_3 \cdot x^3$$

PART 04

特征工程

什么是特征

【特征，标签】

【X，y】

单特征数据

	特征1
样本1	真实值1
样本2	真实值2
样本3	真实值3
样本4	真实值4
样本5	真实值5
样本6	真实值6
样本7	真实值7
样本8	真实值8
样本9	真实值9
样本10	真实值10
样本11	真实值11
样本12	真实值12
样本13	真实值13
样本14	真实值14
样本15	真实值15
样本16	真实值16

多特征数据

	特征1	特征2	特征3	特征4	特征5	特征6	特征7	特征8	真实值
样本1	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值1
样本2	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值2
样本3	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值3
样本4	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值4
样本5	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值5
样本6	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值6
样本7	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值7
样本8	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值8
样本9	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值9
样本10	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值10
样本11	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值11
样本12	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值12
样本13	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值13
样本14	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值14
样本15	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值15
样本16	特征值	特征值	特征值	特征值	特征值	特征值	特征值	特征值	真实值16

特征

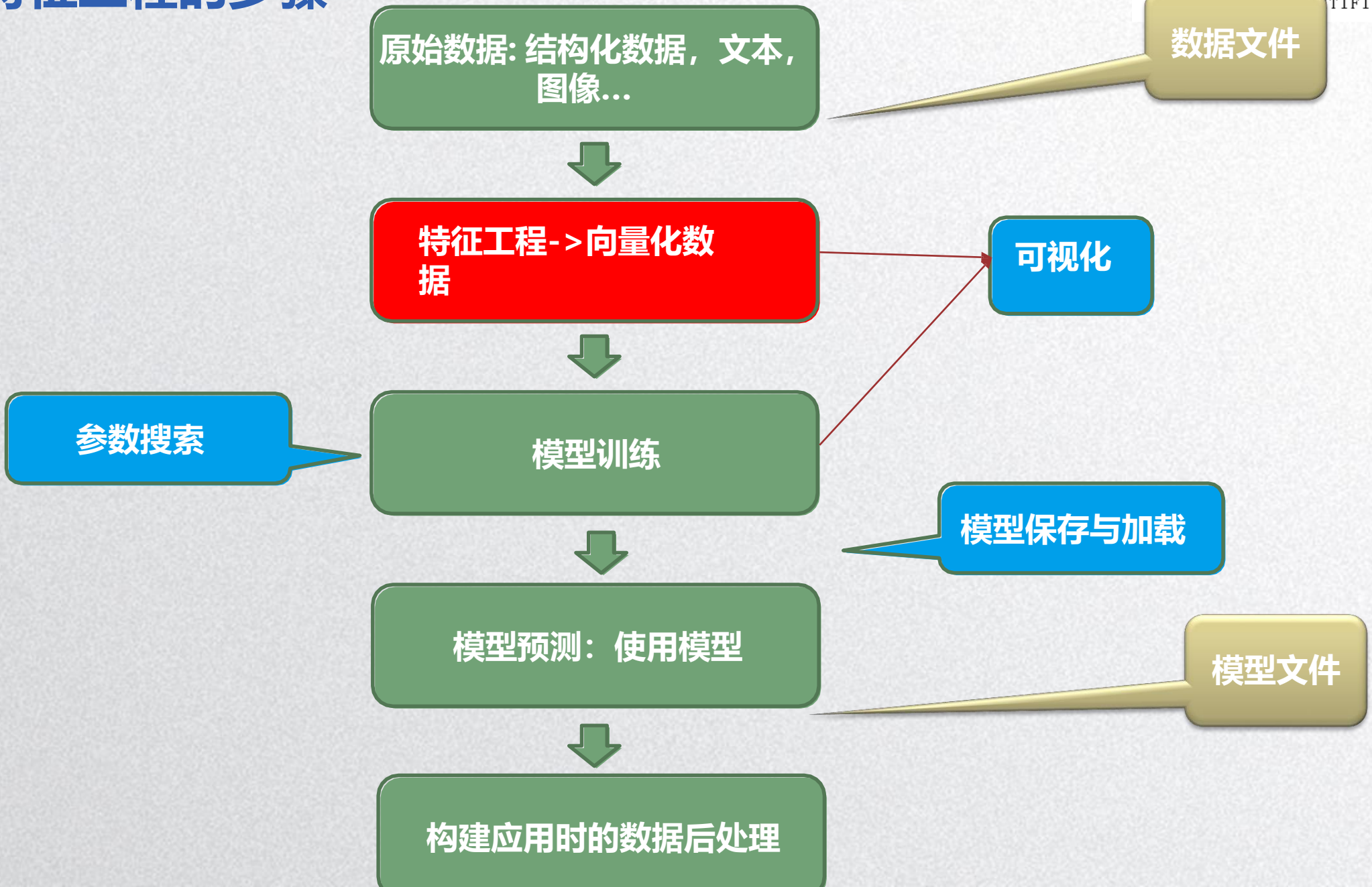
标签

特征工程的意义

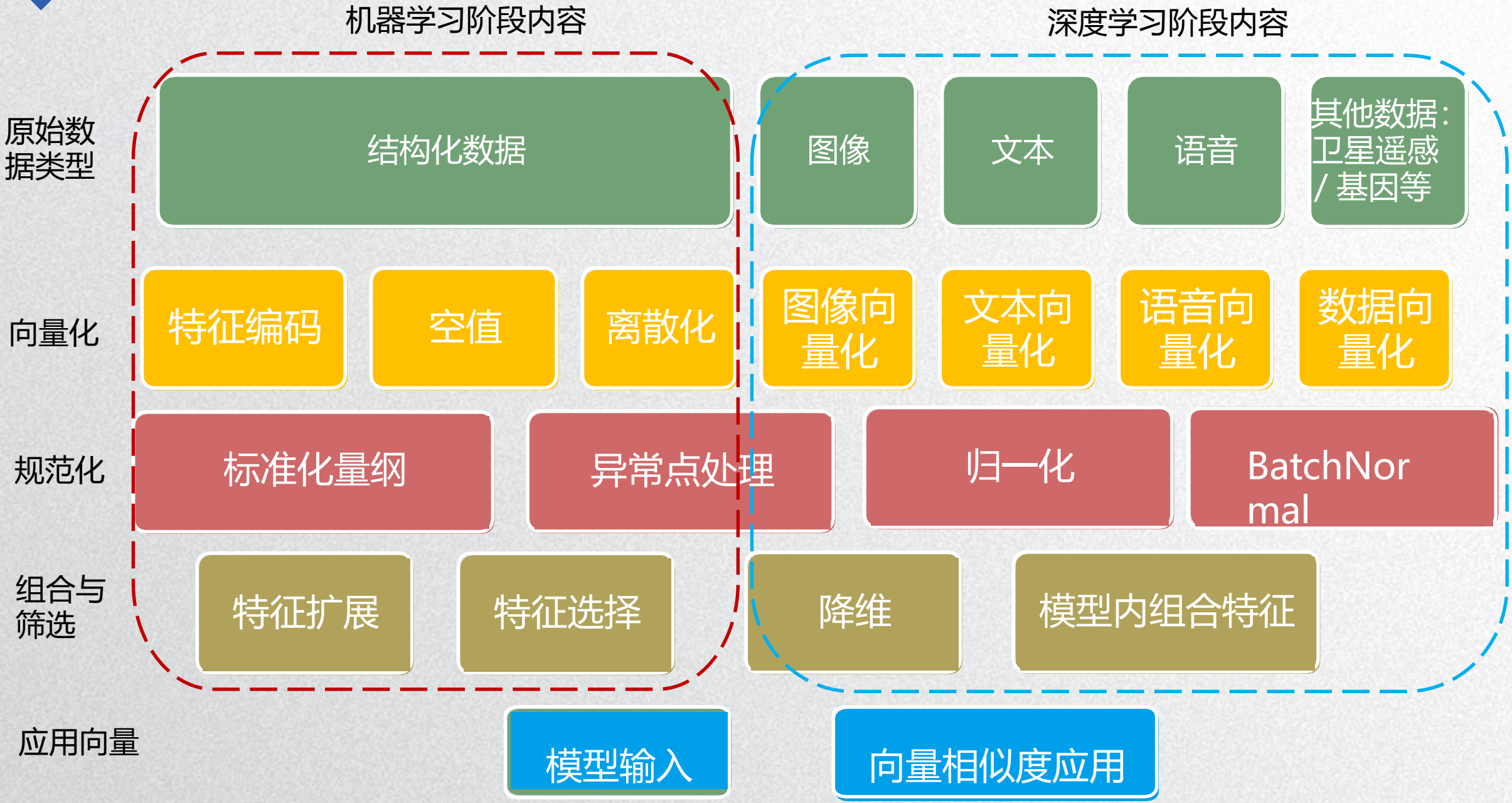
业界广为流传的一句话：
”数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已。”

其本质是一项工程活动，目的是最大限度地从原始数据中提取特征以供算法和模型使用。

特征工程的步骤



人工智能的数据处理



机器学习的使用场景

1、结构化数据、数据量小

数据并非图像、语音、文本，同时数据量不大

2、模型需要可解释性

模型需要可解释性，可以选择决策树等算法

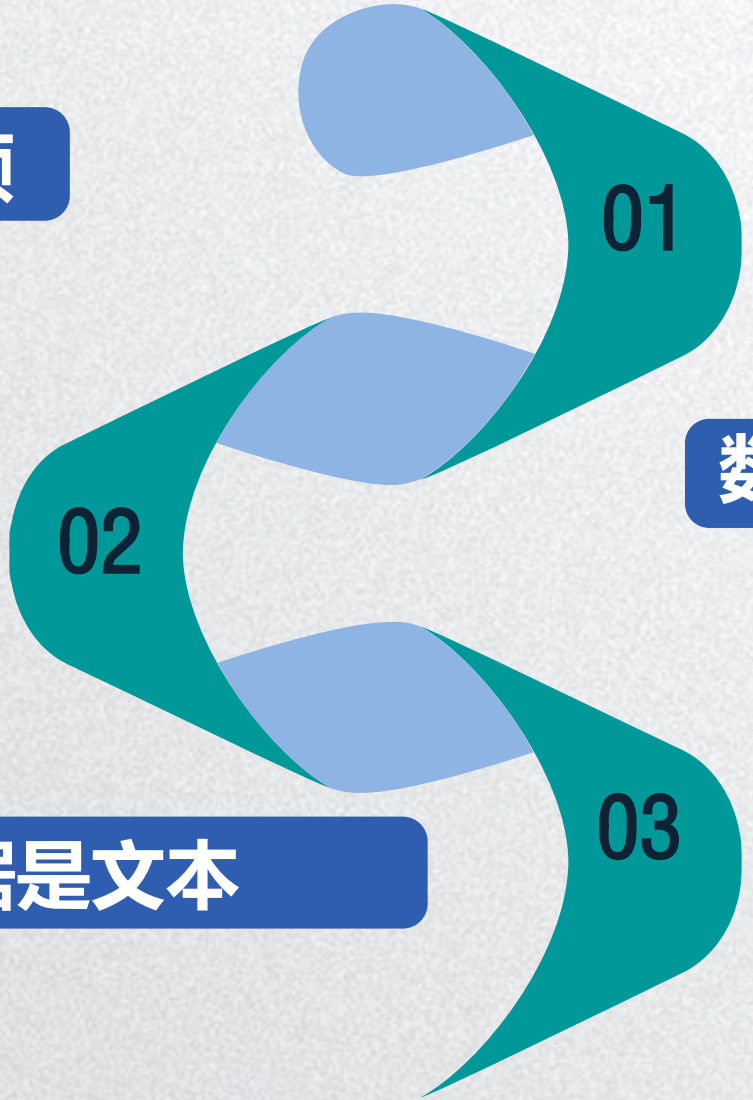
3、特征少

数据量非常小，特征也少，获取不到其他数据，可以使用SVM等算法

4、预测时间

对速度有要求，对于预测的速度要求很高

深度学习的使用场景



数据是图像、视频

数据是语音

数据是文本

一切特征向量化

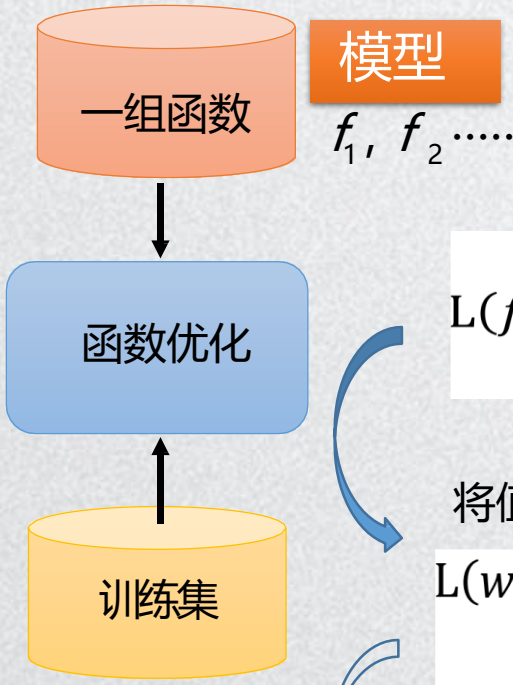
Area,Value,Room,Living,School,Year,Floor

PART 05

损失函数

回归模型的损失(Loss)函数

$y = w \cdot x_i + b$



Loss 函数 L:

输入: 函数,

输出: 多大误差

$$L(f) = \sum_{n=1}^{10} \left(\hat{y}^n - f(x_i^n)\right)^2$$

误差

预测值

将值代入函数

$$L(w,b) = \sum_{n=1}^{10} \left(\hat{y}^n - \left(b + w \cdot x_i^n\right)\right)^2$$

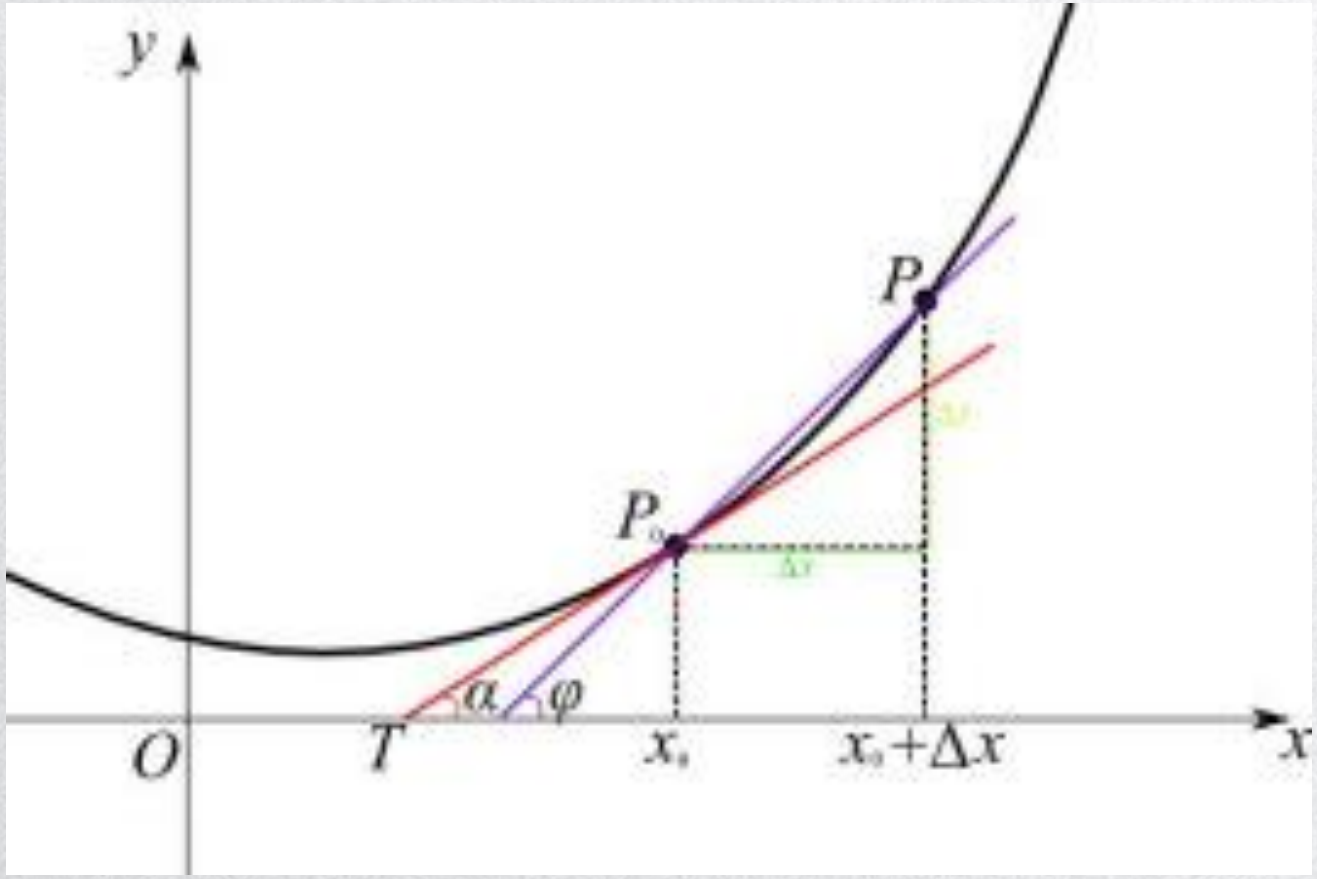
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h\left(\theta^T x_i\right) - y_i\right)^2$$

求Loss函数的均值

求损失函数的极小值

求极值

求极值，又叫求导数，表示在函数在某一点的斜率。
导数（Derivative），也叫导函数，是微积分中的重要基础概念。当函数y=f（x）的自变量x在一点x0上产生一个增量Δx时，函数输出值的增量Δy与自变量增量Δx的比值在Δx趋于0时的极限a如果存在，a即为在x0处的导数，记作f'（x0）或df（x0）/dx。



常见函数的导数

函数	原函数	导函数
常函数 (即常数)	$y = C$ （C为常数）	$y' = 0$
指数函数	$y = a^x$ $y = e^x$	$y' = a^x lna$ $y' = e^x$
幂函数	$y = x^n$	$y' = nx^{n-1}$
对数函数	$y = \log_a x$ $y = lnx$	$y' = \frac{1}{x}\log_a e$ $y' = \frac{1}{x}$
正弦函数	$y = \sin x$	$y' = \cos x$
余弦函数	$y = \cos x$	$y' = -\sin x$
正切函数	$y = \tan x$	$y' = \sec^2 x$
余切函数	$y = \cot x$	$y' = -\csc^2 x$

链式求导

链式法则是微积分中的求导法则，用于求一个复合函数的导数，是在微积分的求导运算中一种常用的方法。复合函数的导数将是构成复合这有限个函数在相应点的 导数的乘积，就像锁链一样一环套一环，故称链式法则。

y是复合函数，求x的导数，先对y求导，再求x的导数

$$\frac{du}{dx} = \frac{du}{dy} \bullet \frac{dy}{dx},$$

偏导数

当函数 z=f(x,y) 在 (x0,y0)的两个偏导数 f'x(x0,y0) 与 f'y(x0,y0) 都存在时，我们称 f(x,y) 在 (x0,y0)处可导。如果函数 f(x,y) 在域 D 的每一点均可导，那么称函数 f(x,y) 在域 D 可导。

求 $z = x^2 + 3xy + y^2$

解 把 y 看作常量，得

$$\frac{\partial z}{\partial x} = 2x + 3y$$

把 x 看作常量，得

$$\frac{\partial z}{\partial y} = 3x + 2y$$

最小二乘法公式推导（1）

$$L=\sum_{i=1}^{N}(y_i-\hat{y}_i)^2$$

$$=(y_1-\hat{y}_1)^2+(y_2-\hat{y}_2)^2+\ldots+(y_N-\hat{y}_N)^2$$

$$=(y_1-(ax_1+b))^2+(y_2-(ax_2+b))^2+\ldots+(y_N-(ax_2+b))^2$$

$$=y_1^2-2y_1(ax_1+b)+(ax_1+b)^2$$
$$+y_2^2-2y_2(ax_2+b)+(ax_2+b)^2$$
$$+\ldots$$
$$=y_1^2-2ax_1y_1-2by_1+a^2x_1^2+2abx_1+b^2$$
$$+y_2^2-2ax_2y_2-2by_2+a^2x_2^2+2abx_2+b^2$$
$$+\ldots$$
$$+y_N^2-2ax_Ny_N-2by_N+a^2x_N^2+2abx_N+b^2$$

$$=\sum_{i=1}^{N}y_i^2-2a\sum_{i=1}^{N}x_iy_i-2b\sum_{i=1}^{N}y_i+a^2\sum_{i=1}^{N}x_i^2+2ab\sum_{i=1}^{N}x_i+Nb^2$$

$$L=\sum_{i=1}^{N}y_i^2-2a\sum_{i=1}^{N}x_iy_i-2b\sum_{i=1}^{N}y_i+a^2\sum_{i=1}^{N}x_i^2+2ab\sum_{i=1}^{N}x_i+Nb^2$$

$$L=\sum_{i=1}^{N}y_i^2-2a\sum_{i=1}^{N}x_iy_i-2b\sum_{i=1}^{N}y_i+a^2\sum_{i=1}^{N}x_i^2+2ab\sum_{i=1}^{N}x_i+Nb^2$$

$$\frac{\partial L}{\partial a}=-2\sum_{i=1}^{N}x_iy_i+2a\sum_{i=1}^{N}x_i^2+2b\sum_{i=1}^{N}x_i=0$$

$$\frac{\partial L}{\partial b}=-2\sum_{i=1}^{N}y_i+2a\sum_{i=1}^{N}x_i+2Nb=0$$

$$\Rightarrow\begin{cases}-2\sum_{i=1}^{N}x_iy_i+2a\sum_{i=1}^{N}x_i^2+2b\sum_{i=1}^{N}x_i=0\\-2\sum_{i=1}^{N}y_i+2a\sum_{i=1}^{N}x_i+2Nb=0\end{cases}$$

$$\Rightarrow\begin{cases}a\sum_{i=1}^{N}x_i^2+b\sum_{i=1}^{N}x_i=\sum_{i=1}^{N}x_iy_i\\a\sum_{i=1}^{N}x_i+Nb=\sum_{i=1}^{N}y_i\end{cases}$$

$$\Rightarrow\begin{cases}a\sum_{i=1}^{N}x_i^2+b\sum_{i=1}^{N}x_i=\sum_{i=1}^{N}x_iy_i\\a\sum_{i=1}^{N}x_i+Nb=\sum_{i=1}^{N}y_i\end{cases}$$

最小二乘法公式推导（2）

$$\Rightarrow \begin{cases} a\overline{x^2} + b\bar{x} = \overline{xy} \\ a\bar{x} + b = \bar{y} \end{cases} \Rightarrow \begin{cases} a\overline{x^2} + (\bar{y} - a\bar{x})\bar{x} = \overline{xy} \\ b = \bar{y} - a\bar{x} \end{cases}$$

$$\Rightarrow \begin{cases} a\overline{x^2} + b\bar{x} = \overline{xy} \\ b = \bar{y} - a\bar{x} \end{cases} \Rightarrow \begin{cases} a\overline{x^2} + (\bar{y} - a\bar{x})\bar{x} = \overline{xy} \\ b = \bar{y} - a\bar{x} \end{cases}$$

$$\Rightarrow \begin{cases} a\overline{x^2} + (\bar{y} - a\bar{x})\bar{x} = \overline{xy} \\ b = \bar{y} - a\bar{x} \end{cases} \Rightarrow \begin{cases} a = \dfrac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}\bar{x}} \\ b = \bar{y} - a\bar{x} \end{cases}$$

光环人工智能
AURA ARTIFICIAL INTELLIGENCE

PART 06

模型选择

函数拟合程度

欠拟合（underfit）

- ✓ 训练集LOSS函数值大（误差大，效果不好），测试集LOSS也大
- ✓ 处理方法：增加模型复杂度，增加特征数量

过拟合（overifit）

- ✓ 训练集LOSS小，测试集LOSS大
- ✓ 处理方法：增加数据量，或者减小模型复杂度

模型收敛

- ✓ 训练集LOSS小，测试集LOSS也小

模型的复杂度

时间复杂度

- ✓ 在计算机科学中，时间复杂性，又称时间复杂度，算法的时间复杂度是一个函数，它定性描述该算法的运行时间。

空间复杂度

- ✓ 空间复杂度(Space Complexity)是对一个算法在运行过程中临时占用存储空间大小的量度，记做S(n)=O(f(n))。

谢谢