

项目：图像识别，情感分析，金融风控，用户群体分析，广告点击率预测，新闻推荐。
chatbot中的意图，股价预测。

限制领域 (narrow):特定场景特定问题

通用AI (general):让AI做任何事

特征工程：将输入数据转化成向量/矩阵/张量的形式

一.KNN：K邻近算法

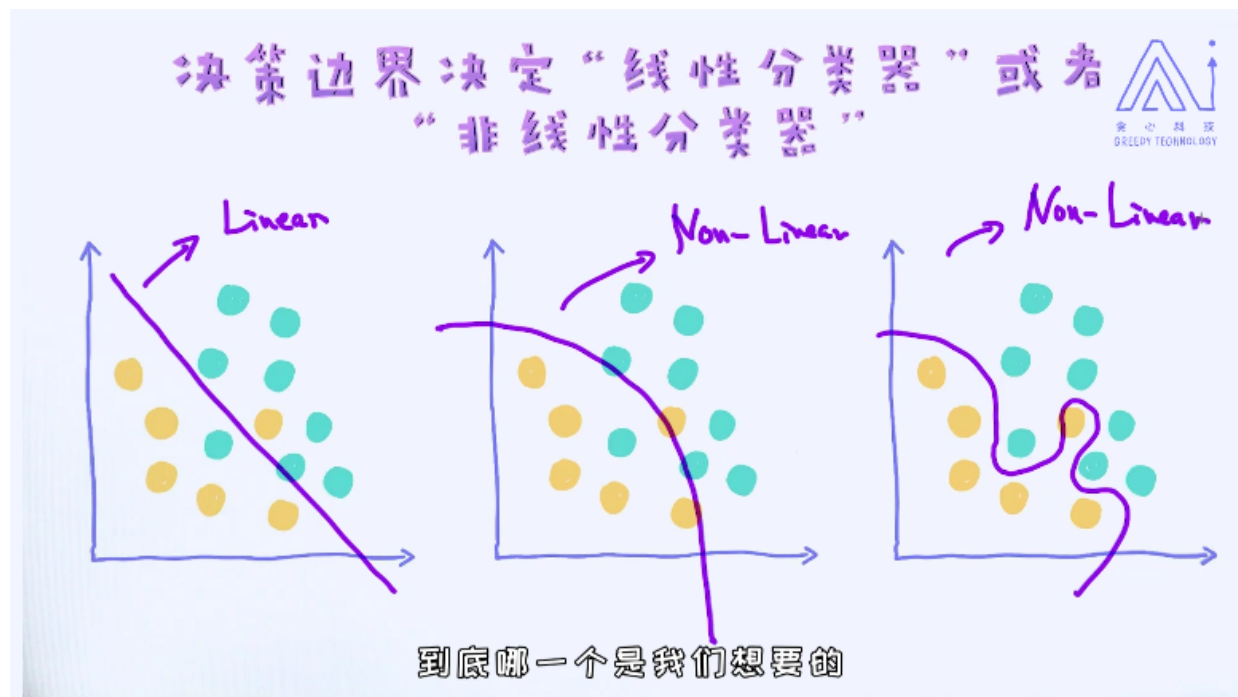
- 1.将数据特征化
- 2.数据需要提前标注好的样本
- 3.计算两个样本之间的距离或者相似度，才能选出最相近的样本

欧式距离=

$$X=(x_1, x_2, x_3, x_4), Y=(y_1, y_2, y_3, y_4)$$
$$d_E(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2}$$

- 4.需要知道如何选择最合适的K值

决策边界：线性决策边界（线性模型）和非线性决策边界（非线性模型）



模型的泛化能力，可以简单理解成“它在新环境中的适应能力”。

数据表现最好，测试不一定最好，图3为过拟合现象。

随着K值的增加，决策边界确实会变得更加平滑。决策边界的平滑也意味着模型的稳定性。

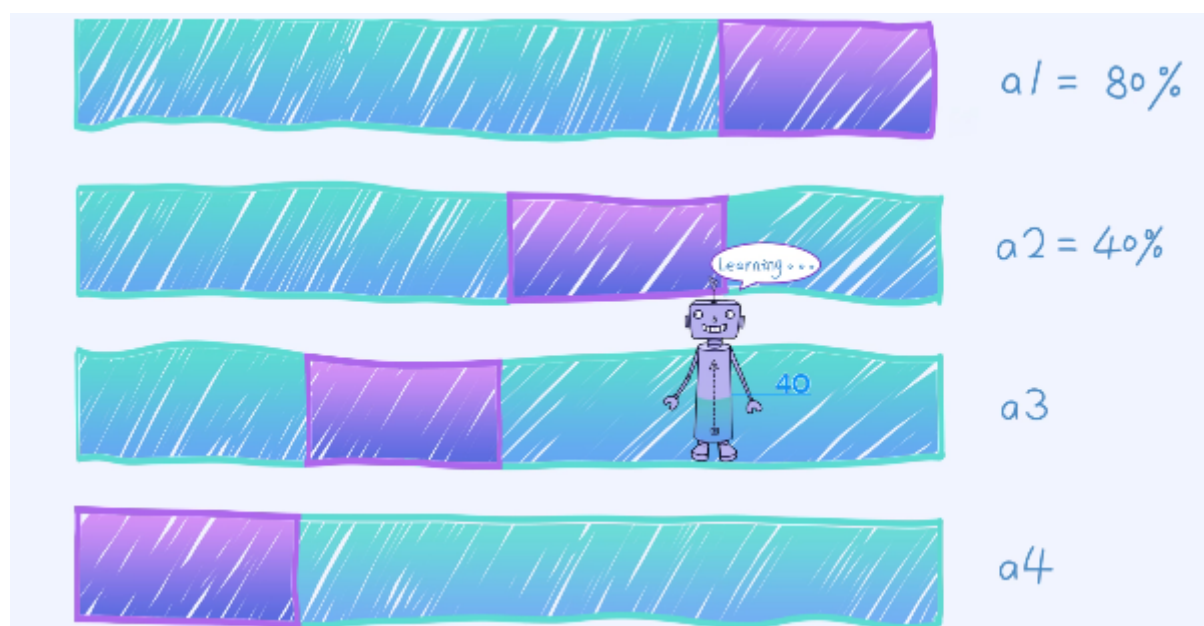
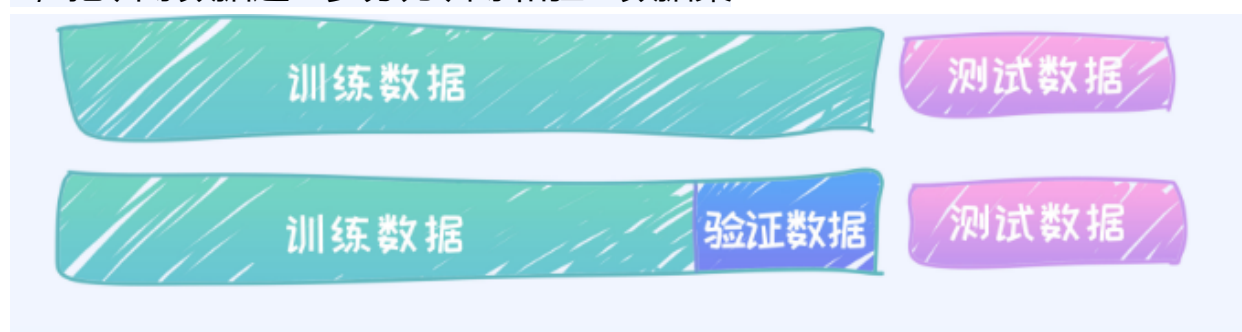
高斯分布

二.交叉验证(cross validation)

KNN决策边界 风险收益平衡

步骤：

1, 把训练数据进一步分为训练和验证数据集



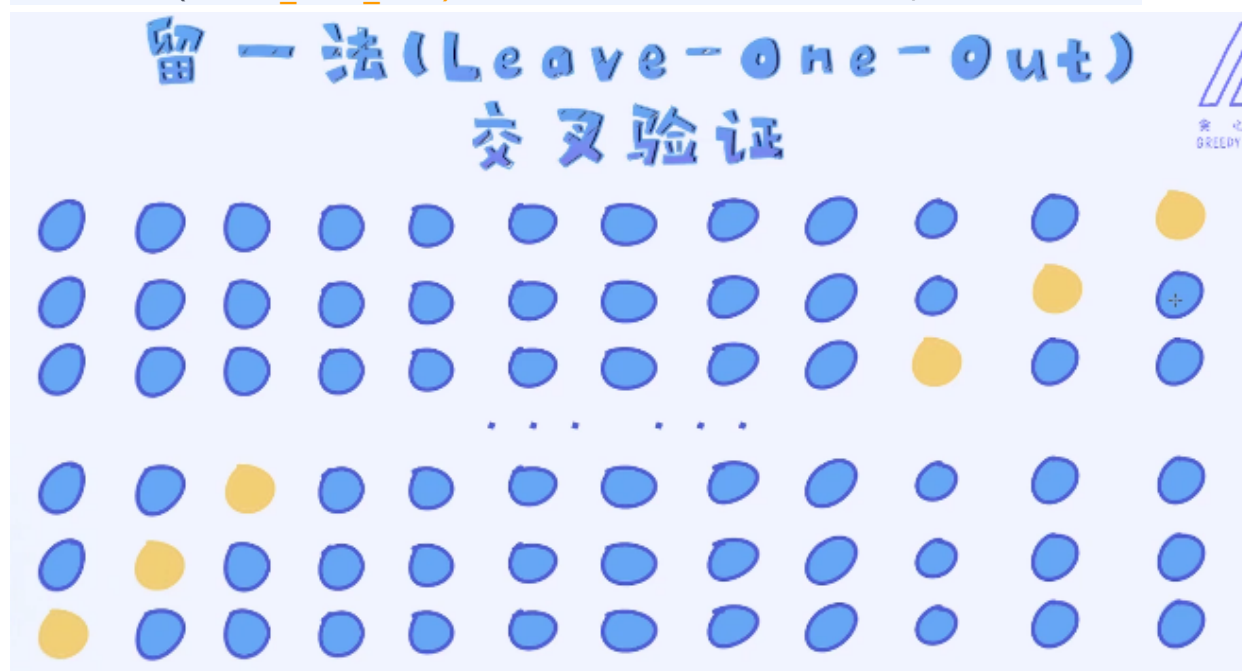
训练集训练模型，验证集评估模型的准确率。

※K折交叉验证(K-fold Cross Validation)：将训练集分成K块，评估k个参数a (k组) 的准确率，然后再取平均值。K称为超参数：不同K值对结果有影响，不取一次是为了避免偶然性。

数据量较少的时候我们取的K值会更大：因为数据量较少的时候如果每次留出比较多的验证数据，对于训练模型本身来说是比较吃亏的，所以这时候我们尽可能使用更多的数据来训练模型。由于每次选择的验证数据量较少，这时候K折中的

K值也会随之而增大，但到最后可以发现，无论K值如何选择，用来验证的样本个数都是等于总样本个数（这句是废话）。

※留一法 (leave_one_out) 交叉验证：K折交叉验证的特例--> $K = N$



对于KNN，K值一般从 $K=1$ 开始尝试，也不会选择太大的值（耗费时间）

提升交叉验证的方法：并行化，分布式处理

※※※**KNN不能用测试数据来引导模型的训练，必须是训练数据集，测试数据是最后一步验证数据准确度使用！！！！**

三、特征缩放

特征标准化的方法：

- 线性归一化：把特征值的范围映射到 $[0,1]$ 区间

特征缩放 - 线性归一化 (Min-max Normalization)

助教经验	助教经验
1.5	0
2	0.2
3	0.4
4	1.0
1.5	0
2.5	0.4
2	0.2
2.3	0.33
3	0.6
1.5	0

$$X_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

\downarrow x的最小值
 \uparrow x的最大值

$$X_{new} = \frac{2.5 - 1.5}{2.5} = \frac{1.0}{2.5} = 0.4$$

- 标准差归一化：把特征值映射到均值为0，标准差为1的正态分布

特征缩放 - 标准差标准化 (Z-score Normalization) *Standardization*

	助教经验	助教经验
x_1	1.5	-1.07
x_2	2	-0.42
	3	0.86
	4	2.15
	1.5	-1.07
	2.5	0.21
	2	-0.42
	2.3	-0.03
	3	0.86
x_{10}	1.5	-1.07

$$X_{new} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

\uparrow x的平均值
 \downarrow x的标准差

$$\bar{x} = \text{mean}(x) = \frac{x_1 + x_2 + \dots + x_{10}}{10} = 2.33$$

$$\text{std}(x) = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_{10} - \bar{x})^2}{10}}$$

数值范围 = [1.5, 4] 数值范围 = N [0, 1]

$$X_{new} = \frac{X - \bar{X}}{u}$$

四、KNN总结

kNN的总结

1. kNN是一个极其简单的算法
2. 算法比较适合应用在低维空间
3. kNN在训练过程中实质上不需要做任何事情，所以训练本身不产生任何时间上的消耗
4. 然而，kNN在预测过程中需要循环所有的样本数据，复杂度线性依赖于样本个数，这成为kNN应用在大数据上时的瓶颈

第二点扩展：当特征数量非常庞大时，要采用特征选择降低维度剔除相关性不大的特征，这样算法复杂度会降低，因此KNN不适合应用在大数据上，以及高维度的特征空间里

第三点扩展：KNN在训练时没有训练参数，核心阶段主要在测试阶段，只需选择测试阶段准确率最高的临近的K个样本即可

第四点扩展：算法时间复杂度跟样本个数N线性相关：减少样本数量，类似KD-Tree方案，LSH近似算法--降低算法时间复杂度