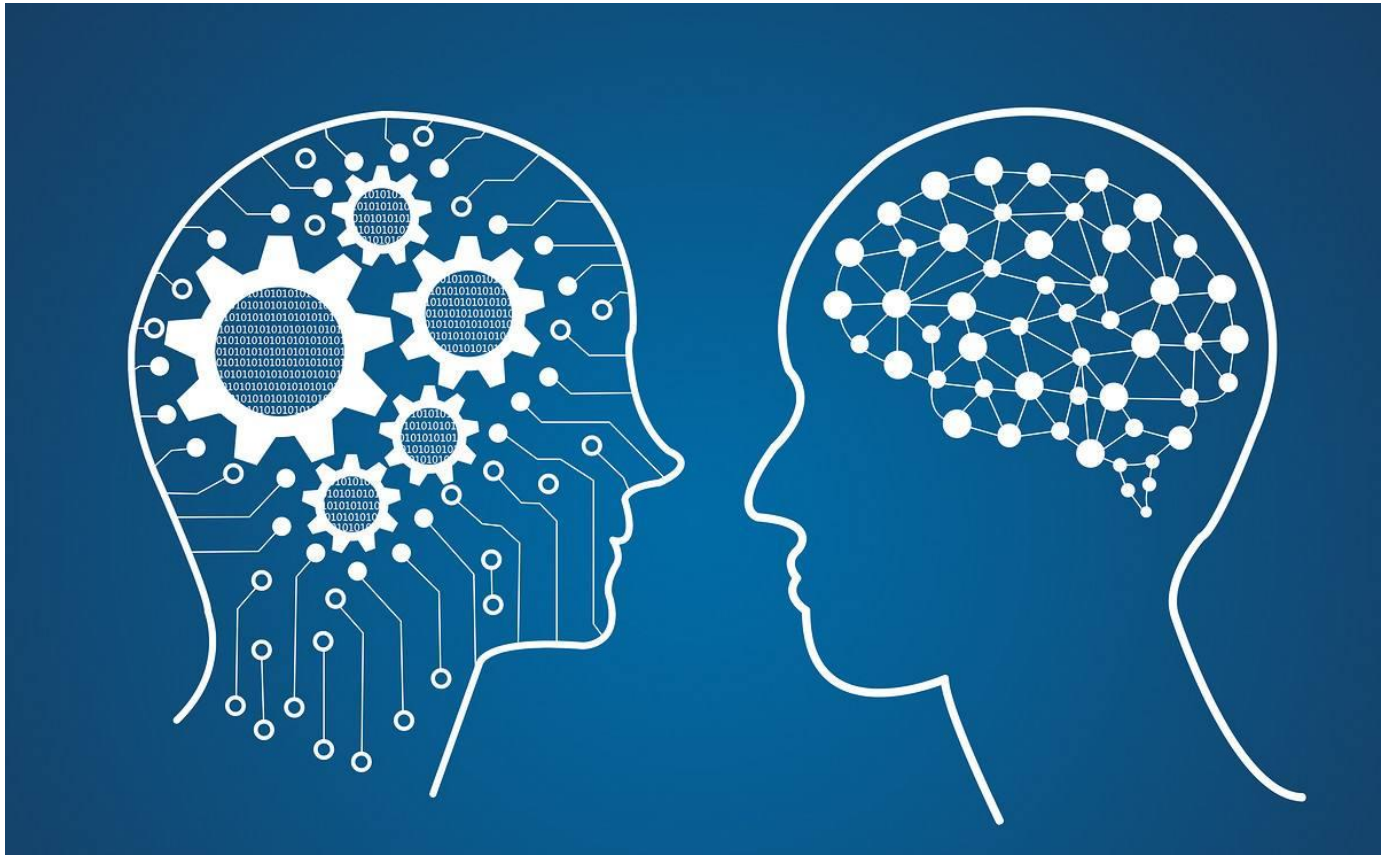
人工智能基础

机器学习——K近邻算法

北京三十五中　　张文轩

知识回顾

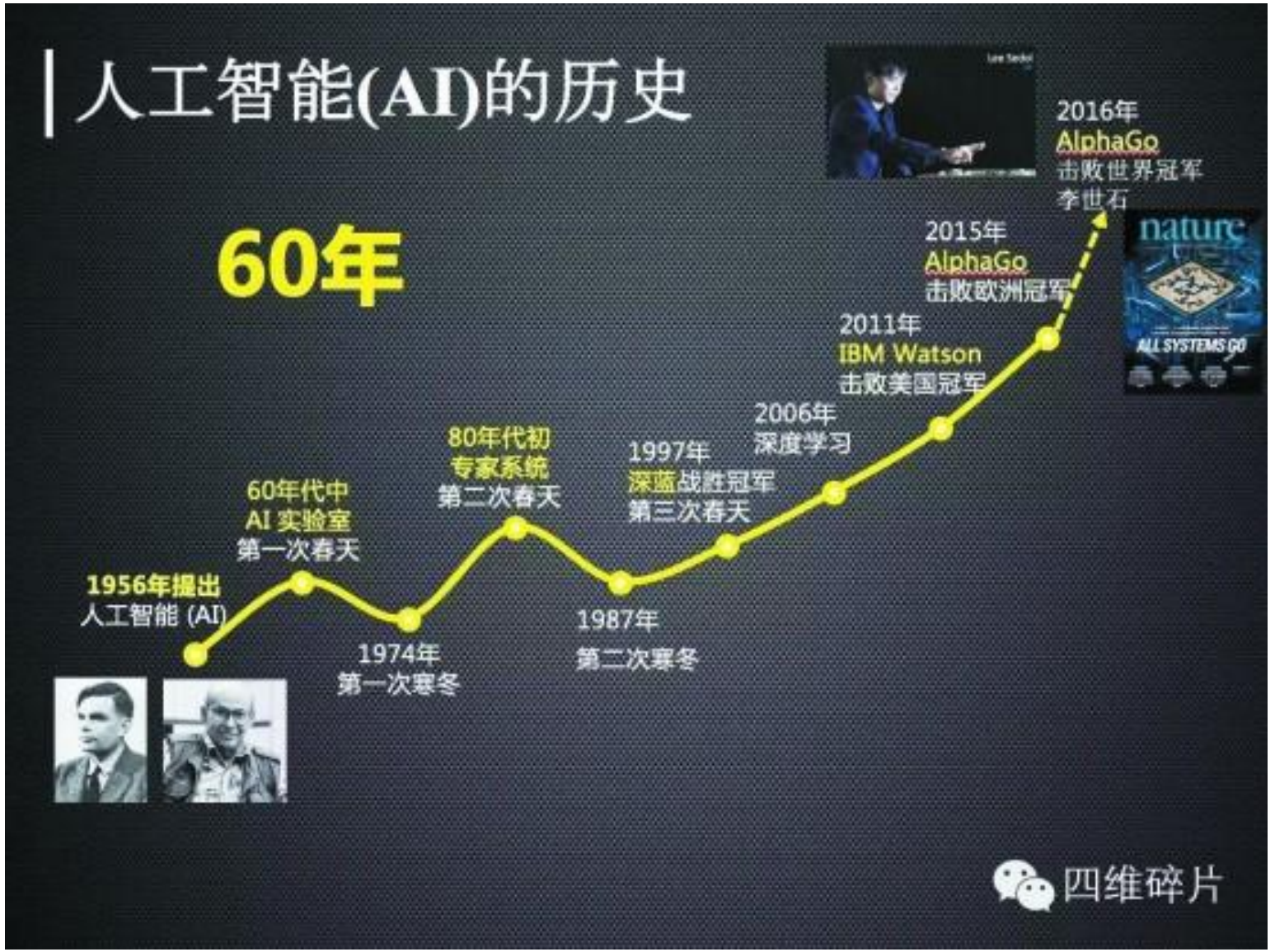
- 1、人工智能的概念：通过机器来模拟人类认知技能的技术。



知识回顾

2、人工智能的历史：

- 1956（诞生）
- 1980（专家系统）
- 21世纪（大数据时代）

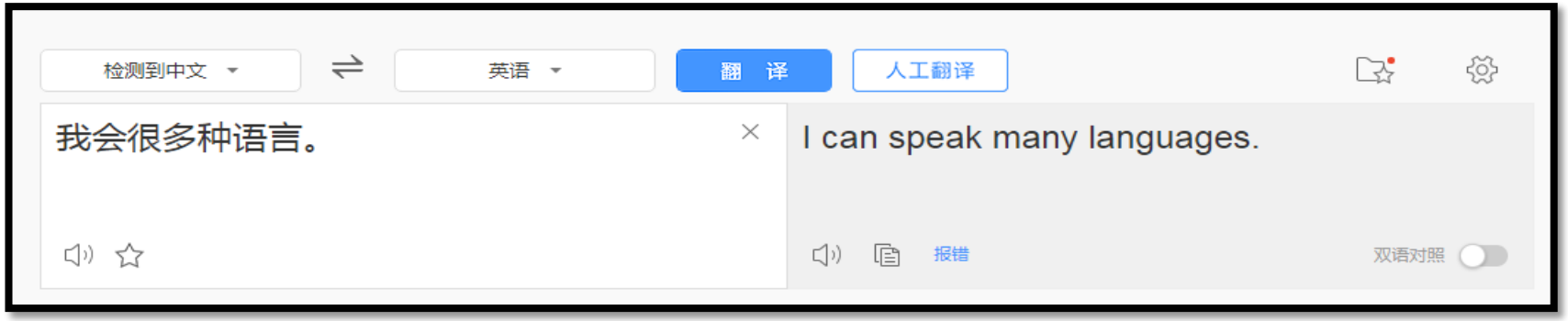


知识回顾

- 3、人工智能的分类：
  - 强人工智能
  - 弱人工智能

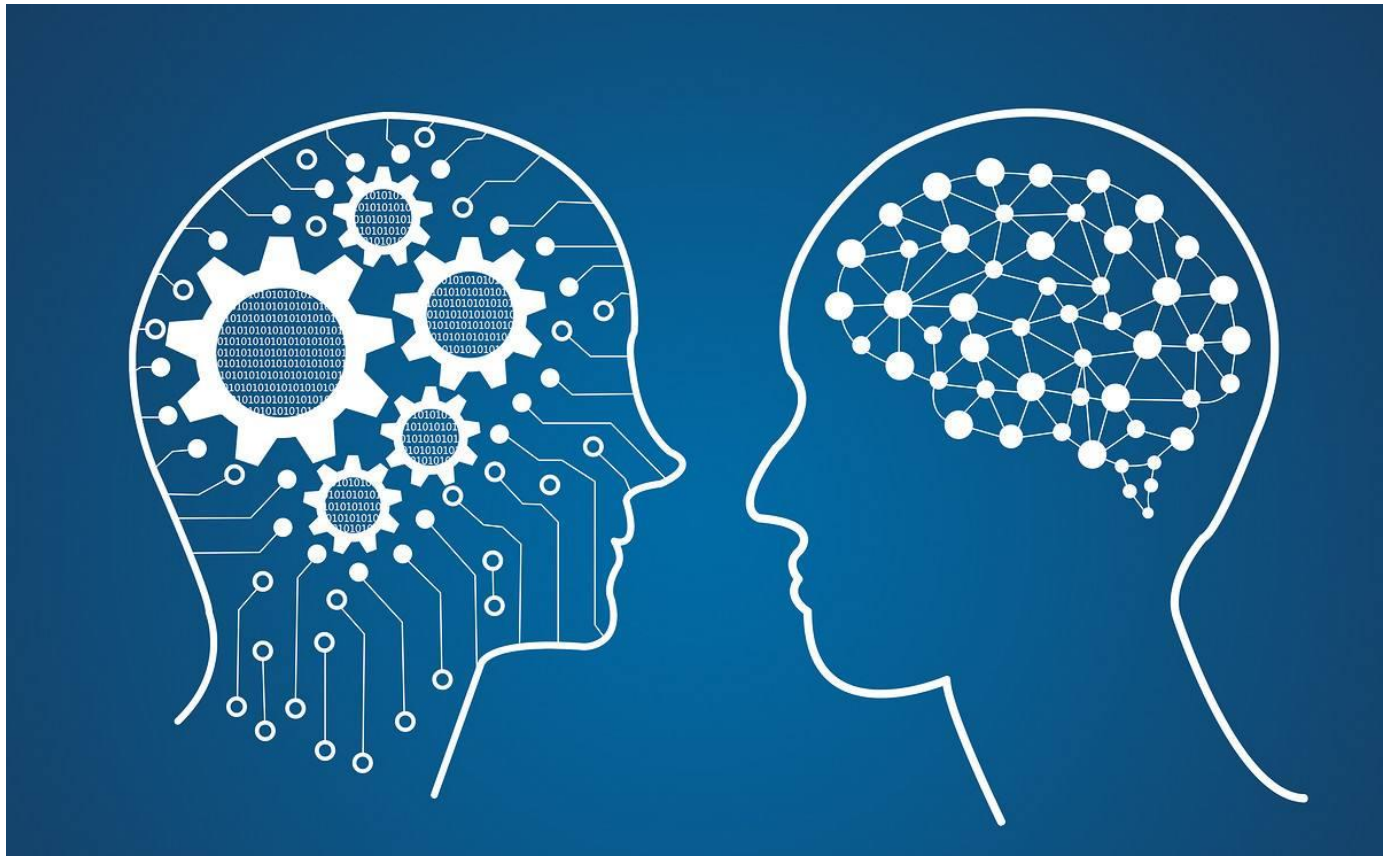
感受人工智能-百度智慧课堂

- 1、百度智慧课堂：人脸识别、文字识别
- 2、百度翻译



思考：机器如何获得智能？

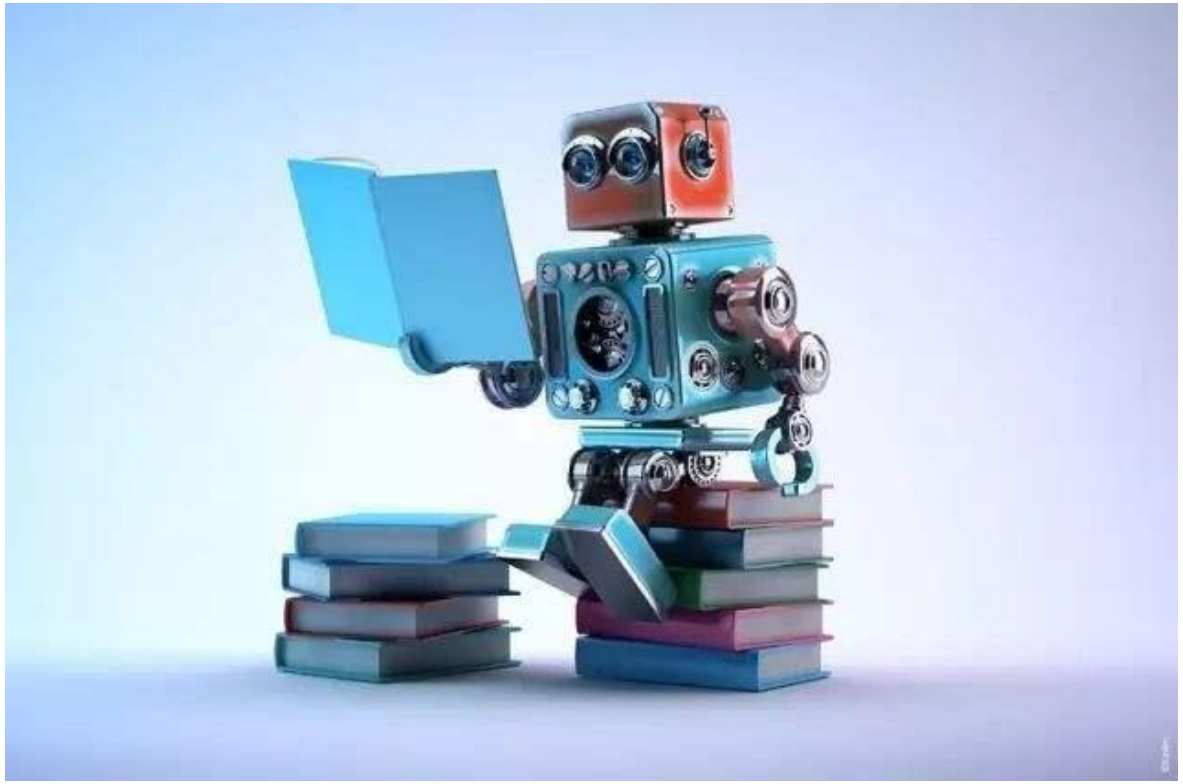
- 人工智能：通过机器来模拟人类认知技能的技术。



我们人类怎样获得智能？

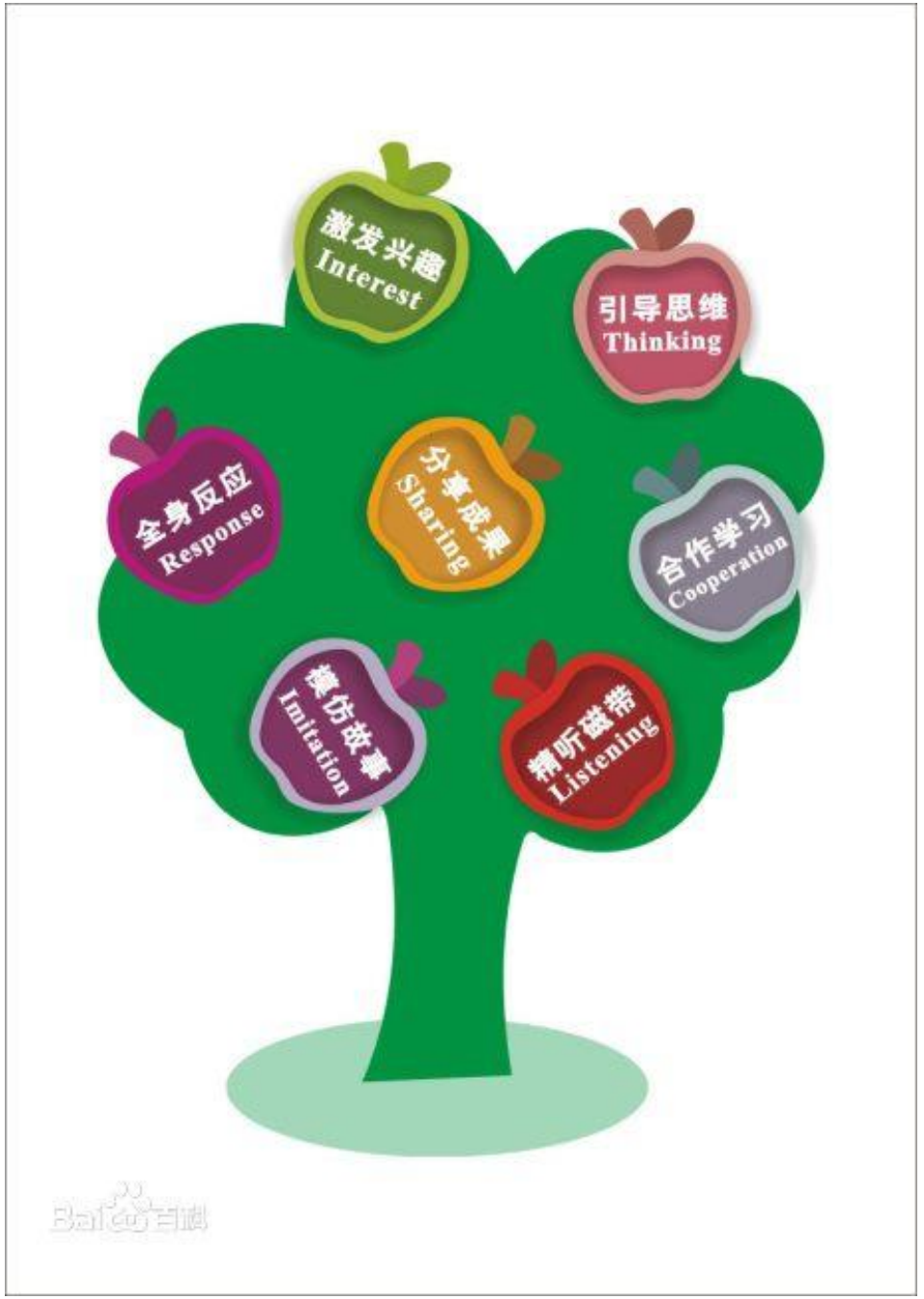
机器如何获得智能？

机器学习

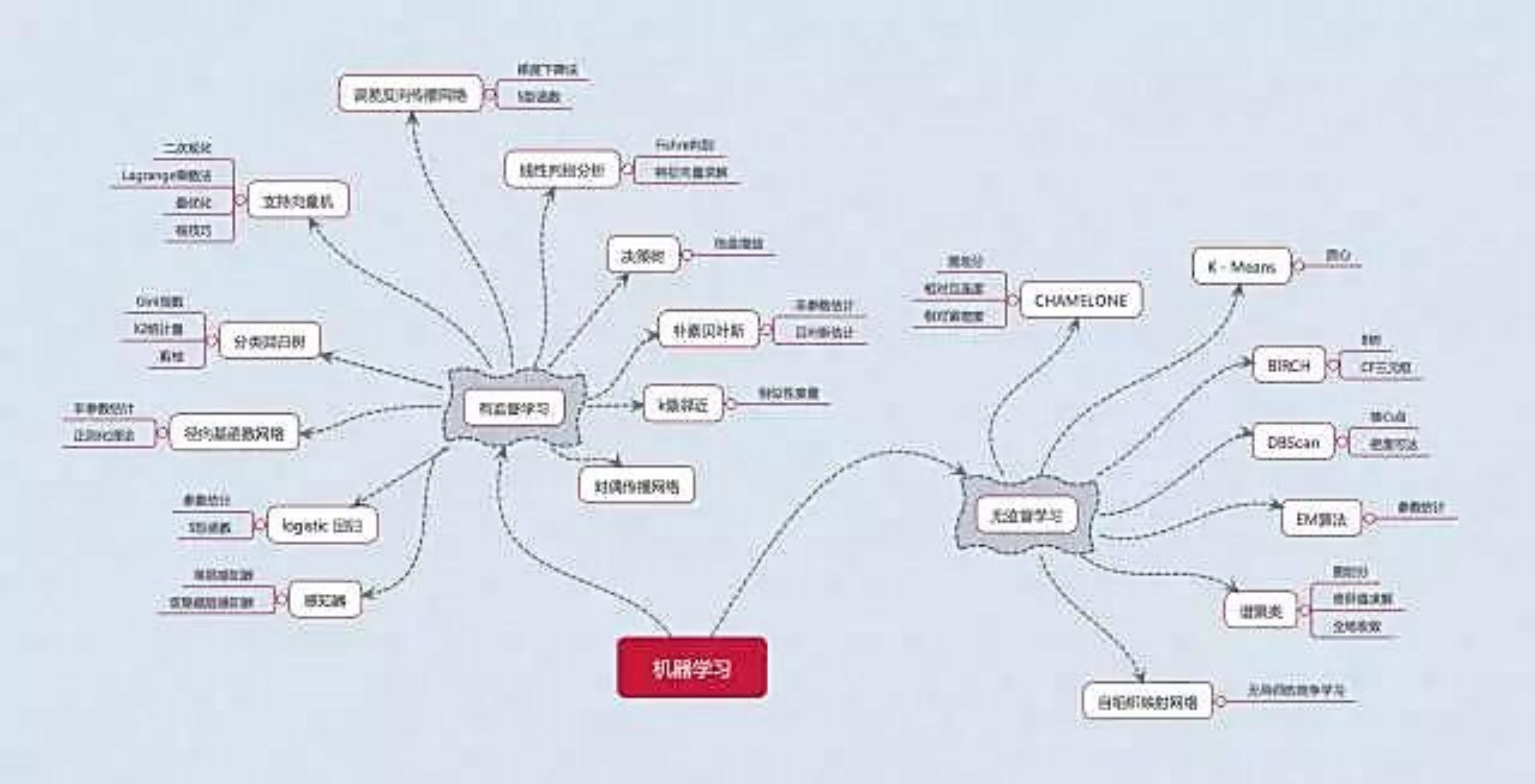


我们怎样学习？


学习方法很重要


激发兴趣
Interest
引导思维
Thinking
全身反应
Response
分享成果
Sharing
合作学习
Cooperation
模仿故事
Imitation
精听磁带
Listening

机器怎样学习？----算法（许多）



机器学习
有监督学习
支持向量机
二次规化
Lagrange乘数法
分类回归树
线性判别分析
朴素贝叶斯
决策树
k最邻近
logistic 回归
无监督学习
CHAMELONE
K - Means
BIRCH
CF三元组
DBScan
EM算法
参数估计
谱聚类
自组织映射网络

各种算法的用途

- 决策树算法：信用评估、赛马结果预测
- 支持向量机：新闻分类、手写识别
- 回归算法：路面交通流量分析、邮件过滤
- 朴素贝叶斯分类：情感分析、消费者分类
- 隐马尔可夫模型：面部表情分析、气象预测
- 循环神经网络：图像分类、政治情感分析
- 长短期记忆：自然语言处理、翻译
- 卷积神经网络：图像识别、文本转语音
- KNN算法：分类预测、手写识别
- 。。。。。。。。。。。。。。

机器学习

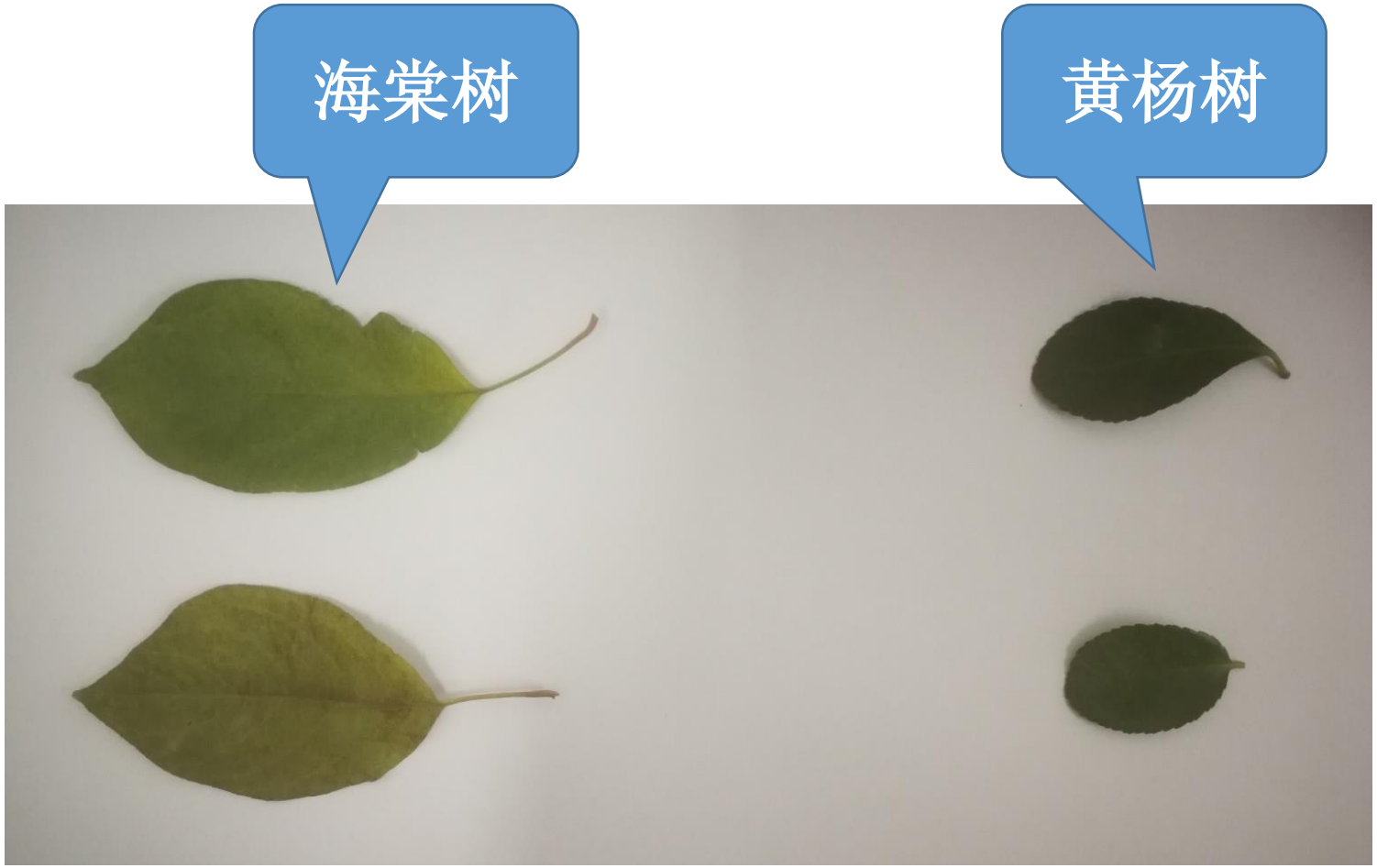
- 观看百度智慧课堂视频：机器学习

第一个机器学习算法

K近邻算法（KNN）

最简单的分类的算法

- 这是什么树的树叶？


海棠树
黄杨树

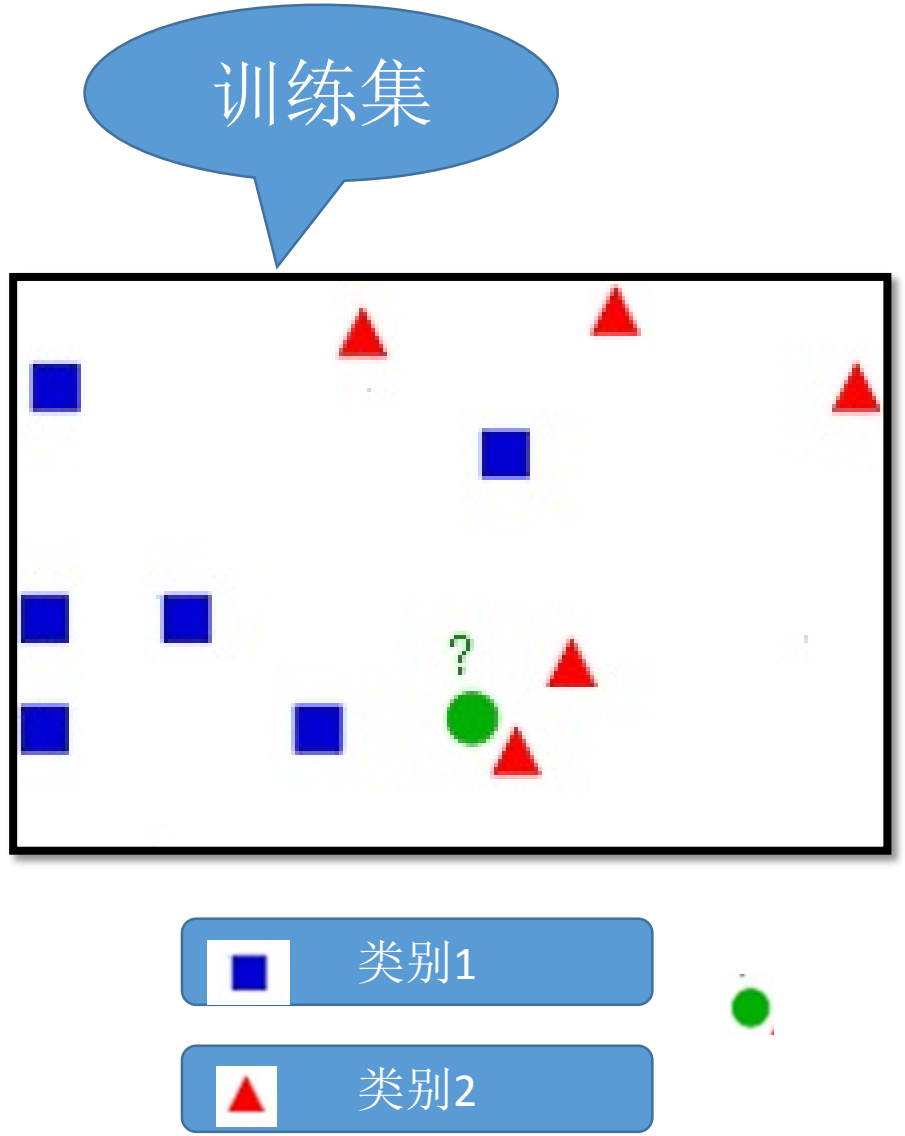
这是哪一种树的树叶？？？？

海棠树

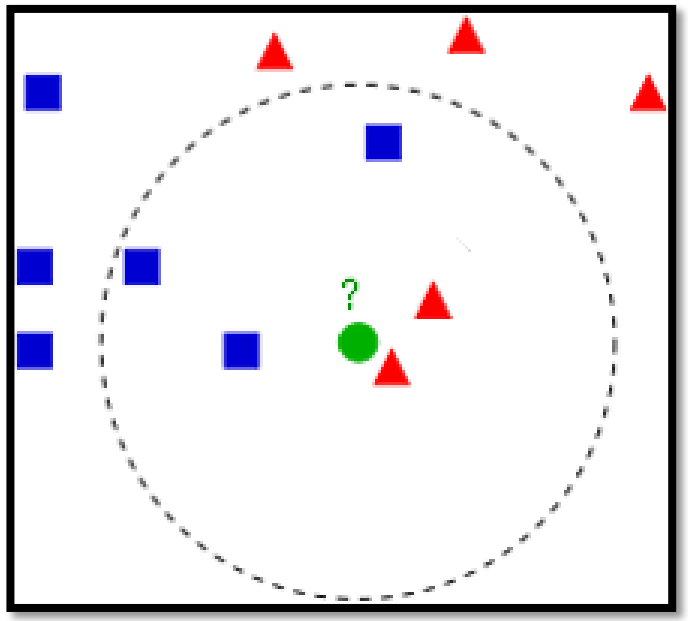
黄杨树

K近邻算法（KNN）

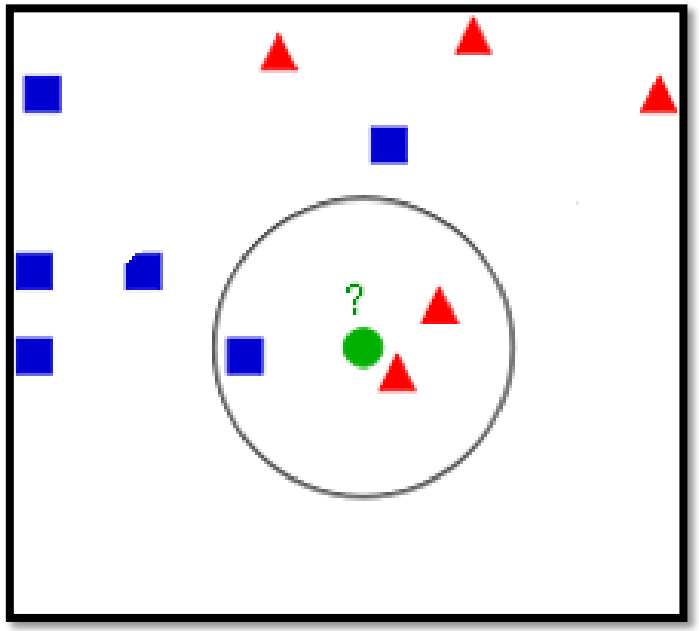
- K近邻算法，即K-Nearest Neighbor algorithm，简称KNN算法。
- K近邻算法是一种简单的用于实现分类的算法。
- 所谓K近邻算法，即是给定已知类别的训练数据集，对于新的未知实例，在训练数据集中找到与该实例最邻近的K个实例（也就是K个邻居），这K个实例的多数属于哪个类，该未知实例就归属于这个类。



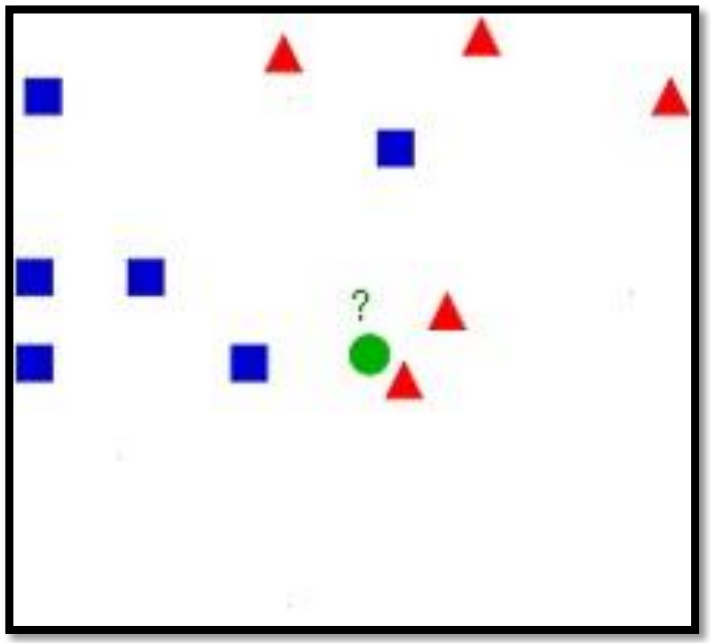
找到几个邻居最合适？--最佳K值



K=5



K=3



如何确定未知实例与邻居的距离？

- **欧氏距离**，最常见的两点之间或多点之间的距离表示法，又称之为欧几里得度量，它定义于欧几里得空间中，如点 x = (x1,...,xn) 和 y = (y1,...,yn) 之间的距离为：

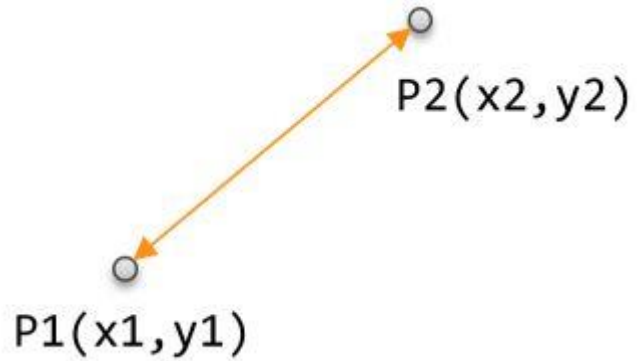
$$d(x,y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- 三维空间两点a(x1,y1,z1)与b(x2,y2,z2)间的欧氏距离：

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- 二维平面上两点a(x1,y1)与b(x2,y2)间的欧氏距离：

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

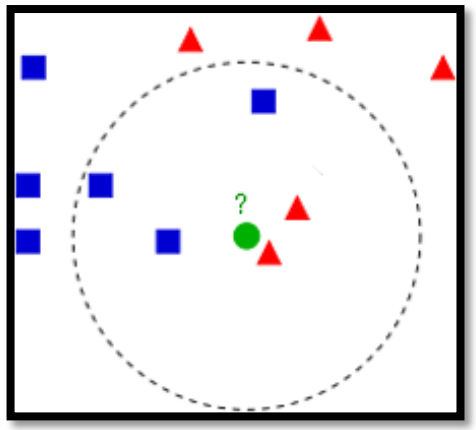


- 一维空间两点a(x1)与b(x2)间的欧氏距离：

  d = |x1-x2|

KNN算法的基本步骤

- 1、确定并获取训练集中的样本的特征值；
- 2、计算训练集中的样本与未知数据之间的距离；
- 3、按距离降序排序，选取与当前点距离最小的k个点；
- 4、计算前k个点所在类别出现的频率；
- 5、以前k个点中出现频率最高的类别作为预测类别；

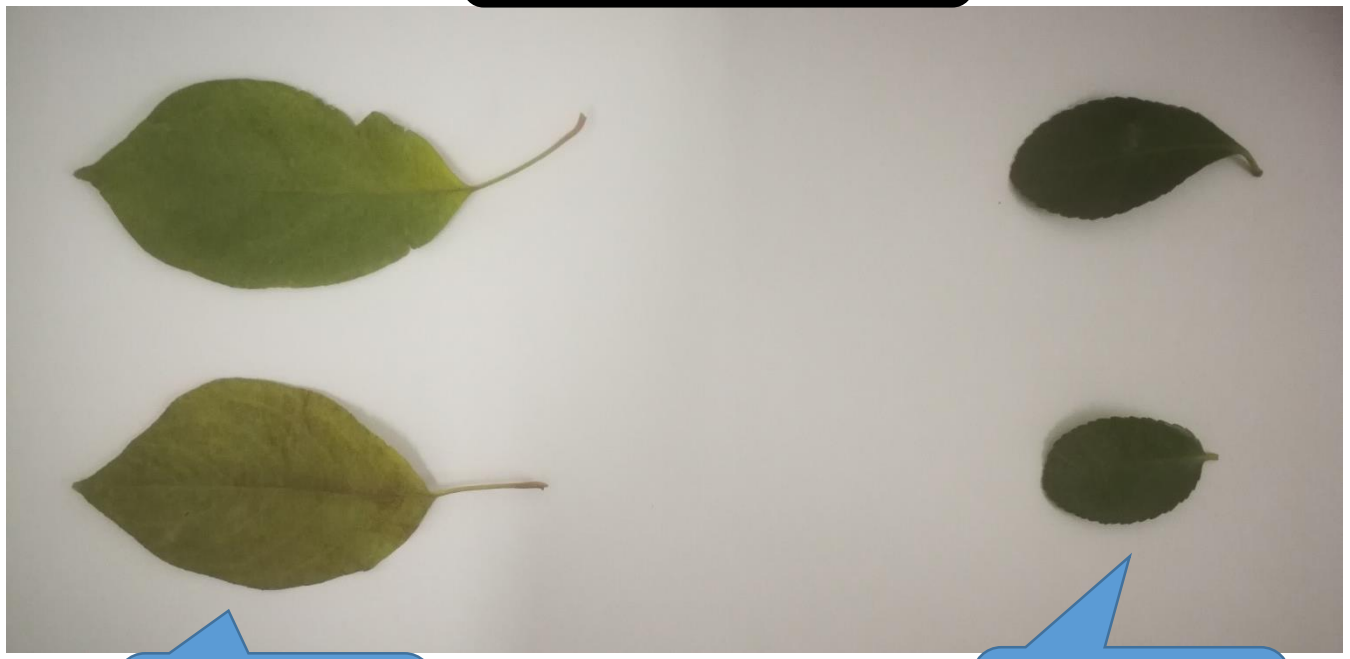


判断树叶的类别

未知数据

？？？

训练集

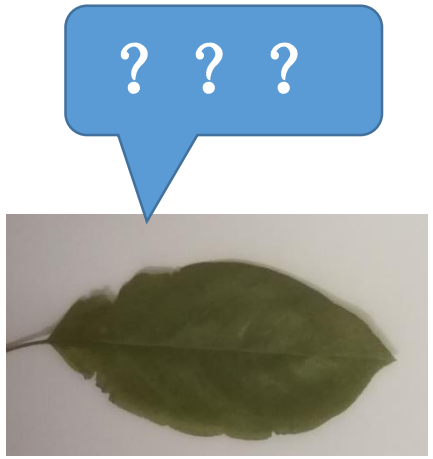


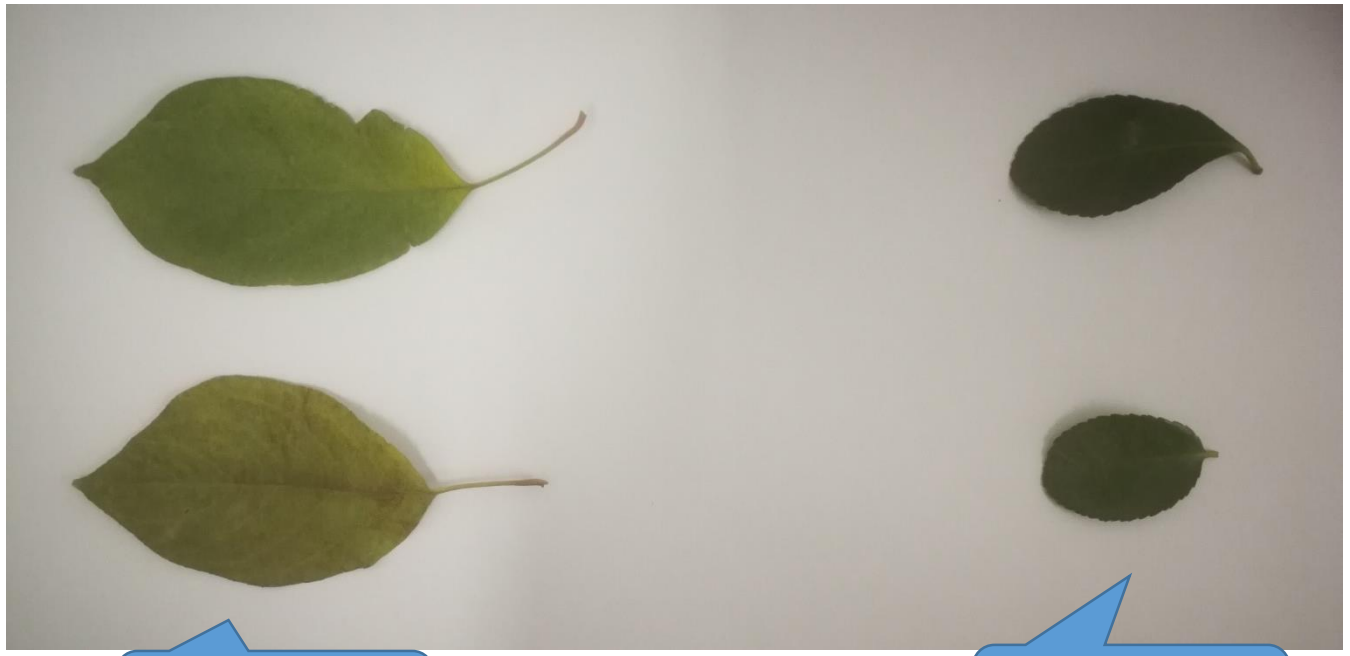
海棠树

黄杨树

KNN算法参数确定

- 特征值：一个特征值（树叶的长度）
- 距离：欧氏距离（长度的差）
- 邻居个数（K值）：K = 1





详细步骤演示

- 1、取得所有的已知树叶的长度值和类型
- 获取未知树叶的长度值

未知数据

？？？

训练集

海棠树

黄杨树

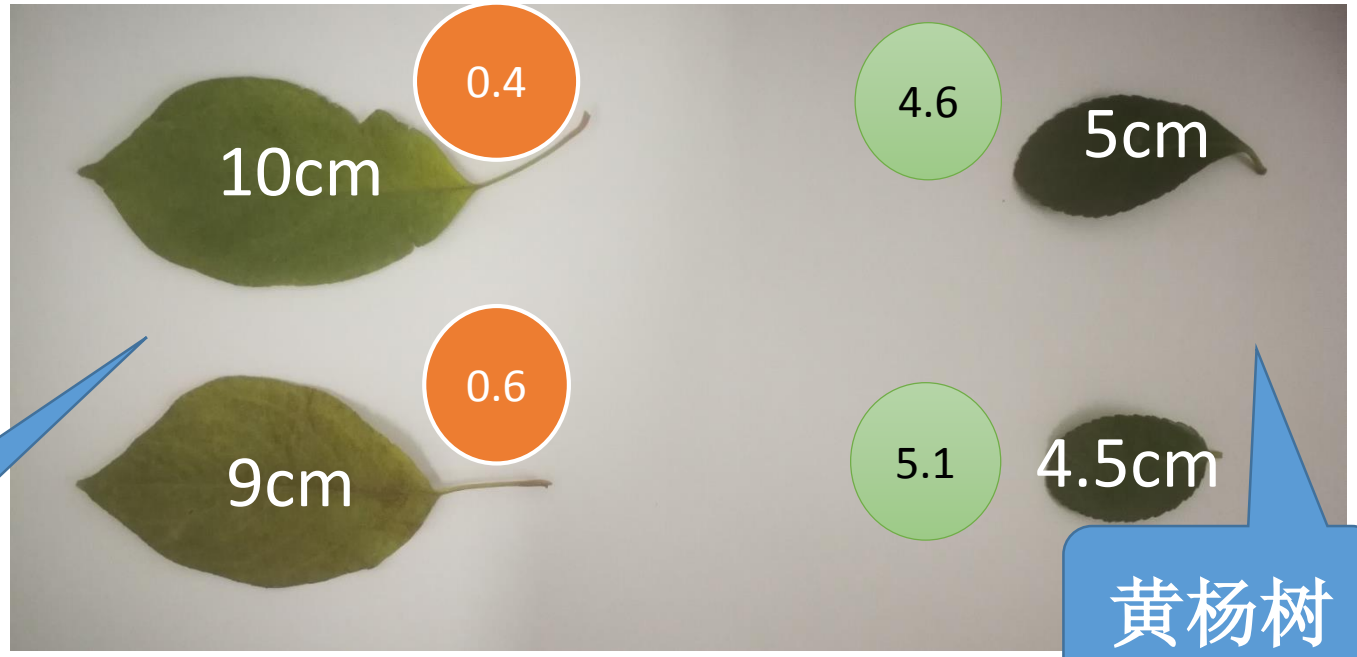
详细步骤演示

- 2、计算获得未知种类的树叶和每一个已知种类的树叶的长度的差（距离）

未知数据

？？？

训练集



海棠树

黄杨树

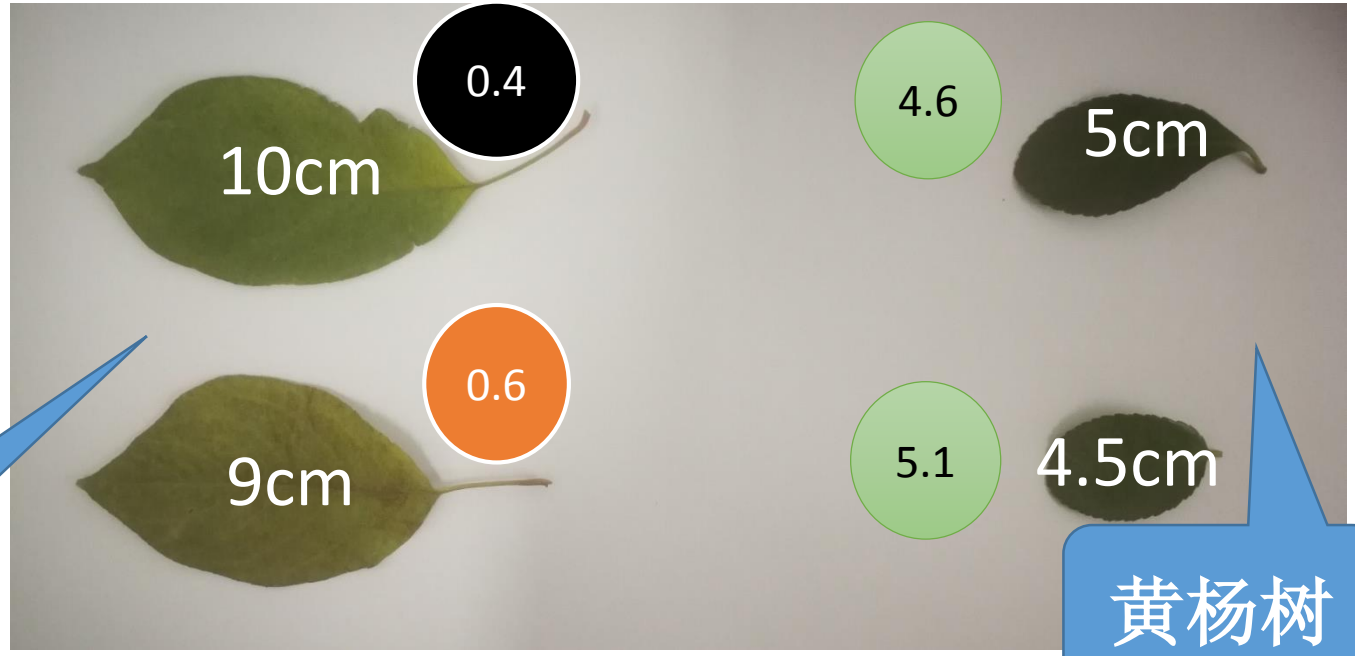
判断树叶的类别

- 3、找到差值中的最小值（K=1）

未知数据

？？？

训练集



海棠树

黄杨树

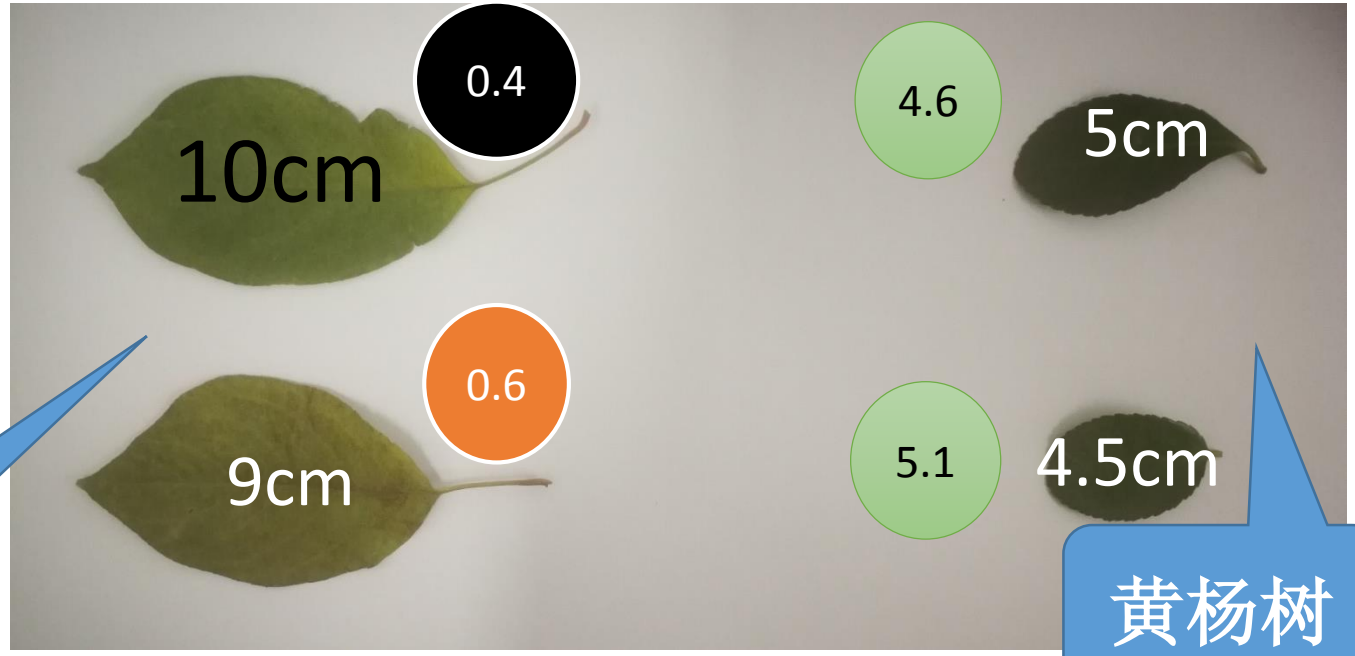
判断树叶的类别

- 4、找到最小差值对应的树叶长度值

未知数据

？？？

训练集



海棠树

黄杨树

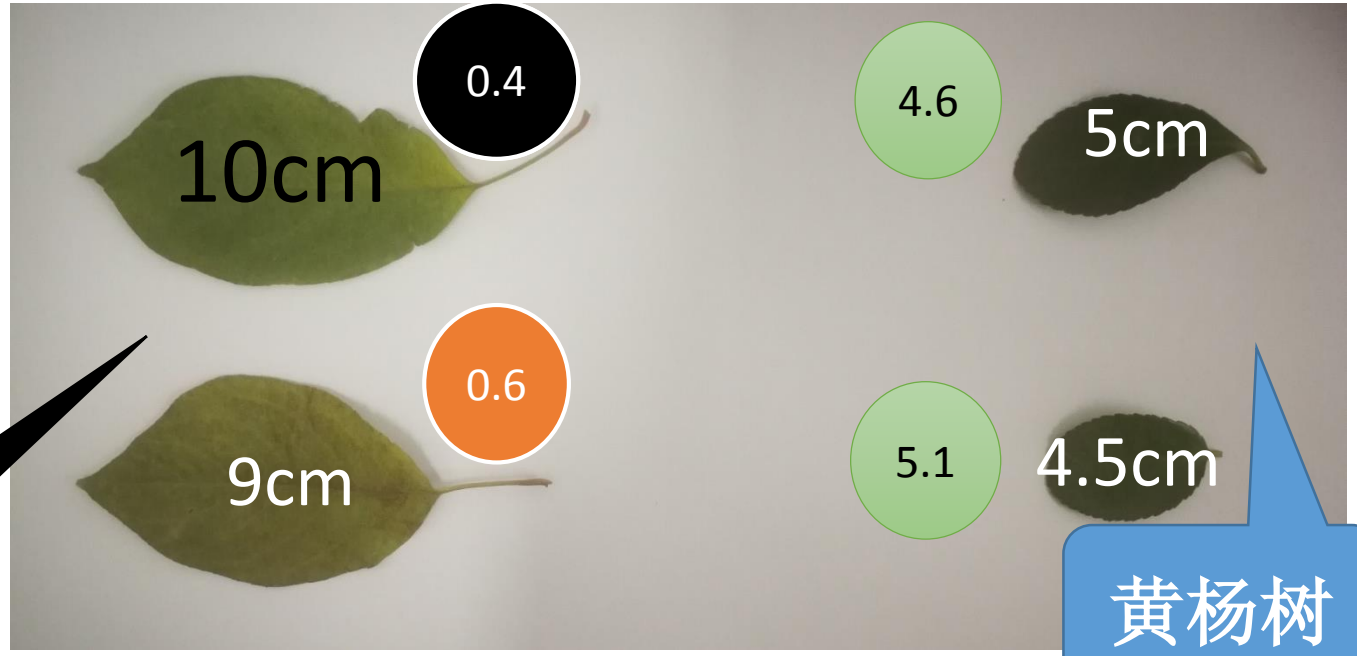
判断树叶的类别

- 5、该长度值对应的种类既为未知树叶的种类。

未知数据

海棠树

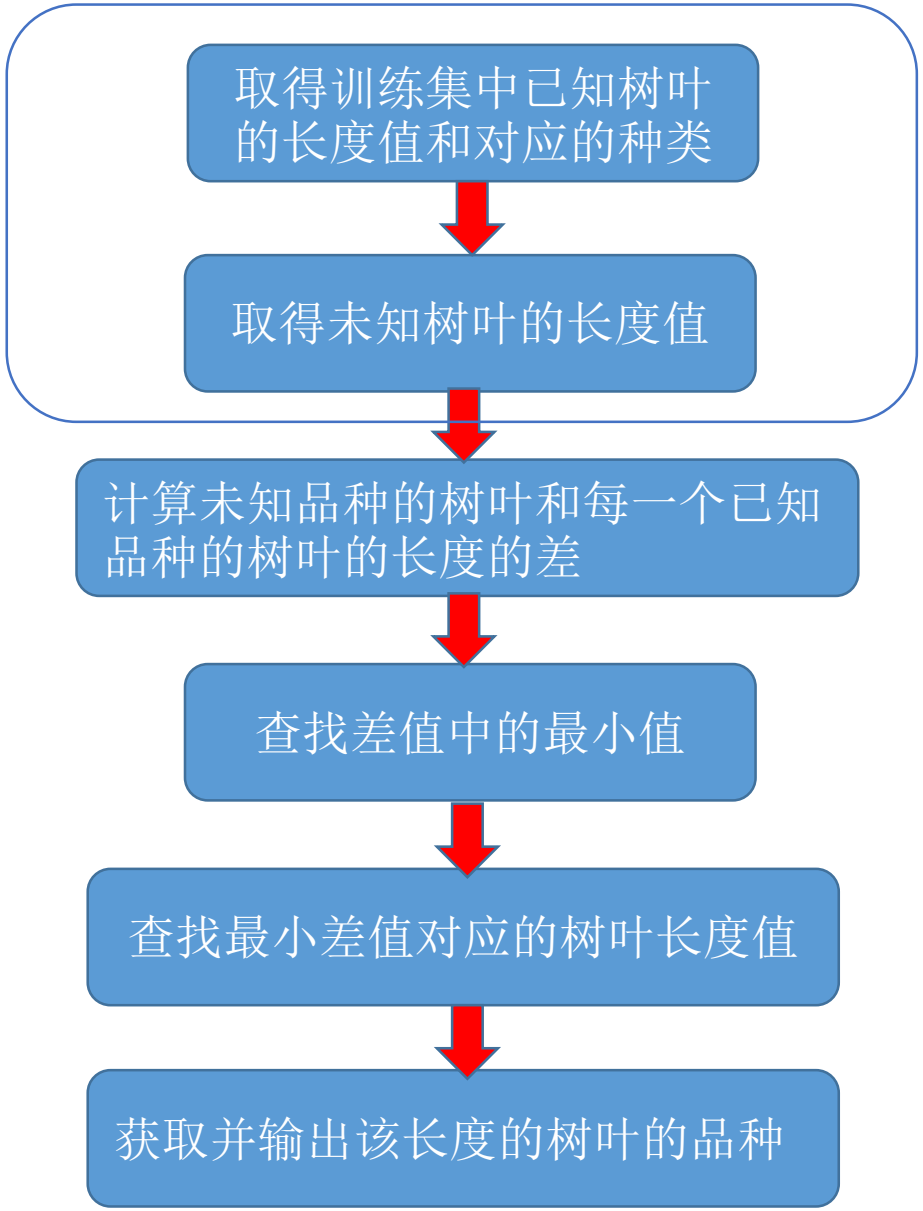
训练集



海棠树

黄杨树

算法流程图


取得训练集中已知树叶的长度值和对应的种类
取得未知树叶的长度值
计算未知品种的树叶和每一个已知品种的树叶的长度的差
查找差值中的最小值
查找最小差值对应的树叶长度值
获取并输出该长度的树叶的品种

程序实现

knn01.py - D:\用户\Desktop\机器学习KNN\机器学习-KNN算法\教师\程序\knn01.py (3.6.3)

File Edit Format Run Options Window Help

```
group = [10.0,9.0,5.0,4.5]
labels = ['海棠','海棠','黄杨','黄杨']
inX = float(input("请输入树叶的长度（厘米）："))
distense = []
for i in range(len(group)):
    distense.append(abs(group[i]-inX))
x = min(distense)
y = distense.index(x)
print("这个树的品种为：",labels[y])
```

Ln: 10 Col: 0

knn01-1.py - D:\用户\Desktop\机器学习KNN\机器学习-KNN算法\教师\程序\knn01-1.py (3.6.3)

File Edit Format Run Options Window Help

```
#1、创建两个列表，保存已知数据特征值集与结果集
group = [10.0,9.0,5.0,4.5]
labels = ['海棠','海棠','黄杨','黄杨']

#2、用户输入未知数据的特征值
inX = float(input("请输入树叶的长度（厘米）："))

#3、创建一个空列表，用来存放特征值的差值
distense = []
#循环求得被判断的数据与已知数据集内的每个数据的距离
for i in range(len(group)):
    distense.append(abs(group[i]-inX))

#4、求距离集合中的最小值
x = min(distense)

#5、求最小值在其集合列表中的序号，该序号即为其对应的已知数据的结果的序号
y = distense.index(x)

#6、打印最小距离需要对应的结果
print("这个树的品种为：",labels[y])
```

Ln: 21 Col: 27

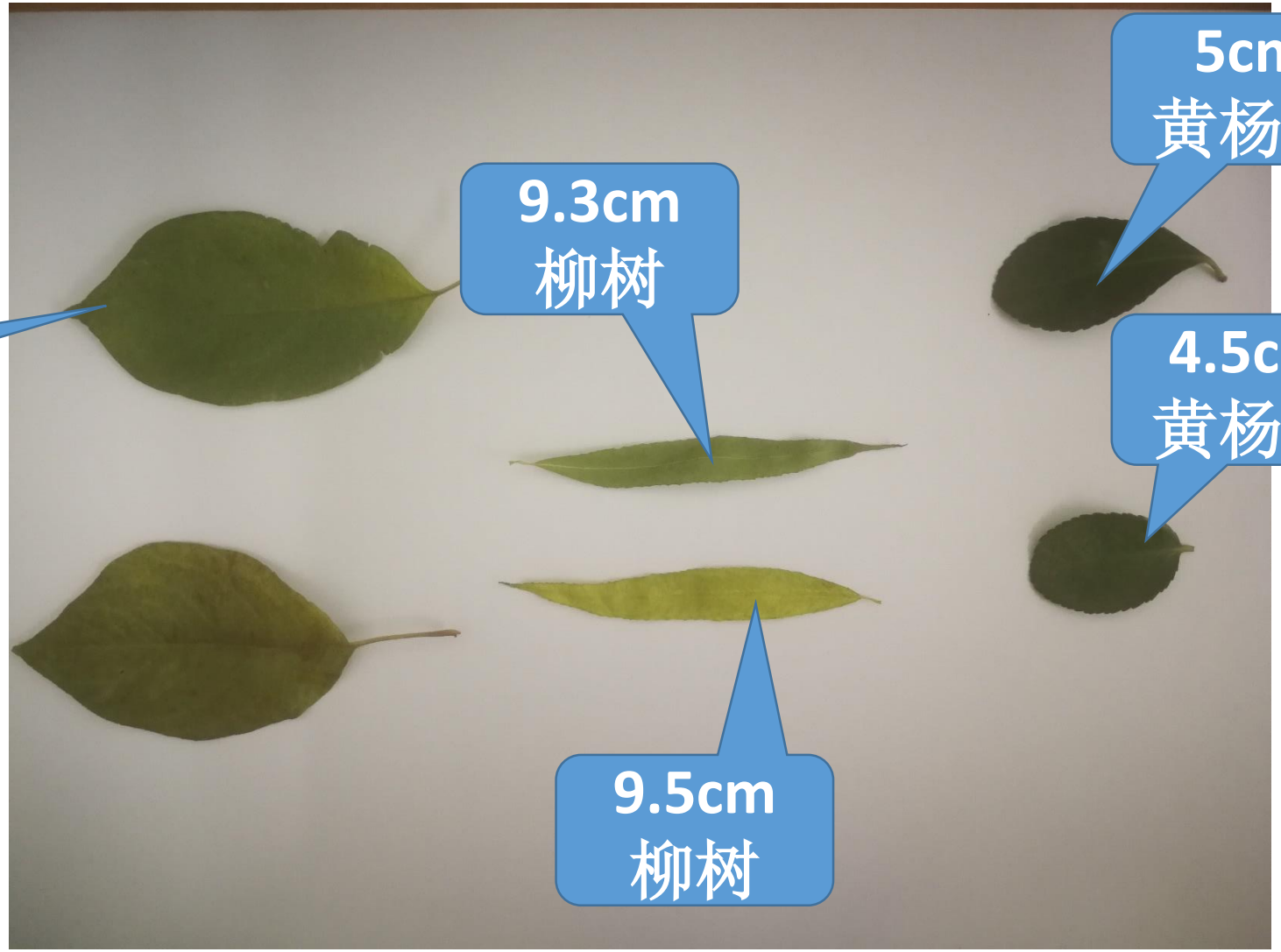
实践题目1：

- 1、选取不同种类的树叶各2个，测量长度值，填写在表格1中。
- 2、打开idel程序环境，打开程序文档knn01.py。
- 3、运行该程序，输入前面测量的长度值，输入结果填写在表格1中。

表格 1			
序号	树叶长度（厘米）	**预测结果（种类）**	是否正确
1（示例）	11.2	海棠	是
2			
3			
4			

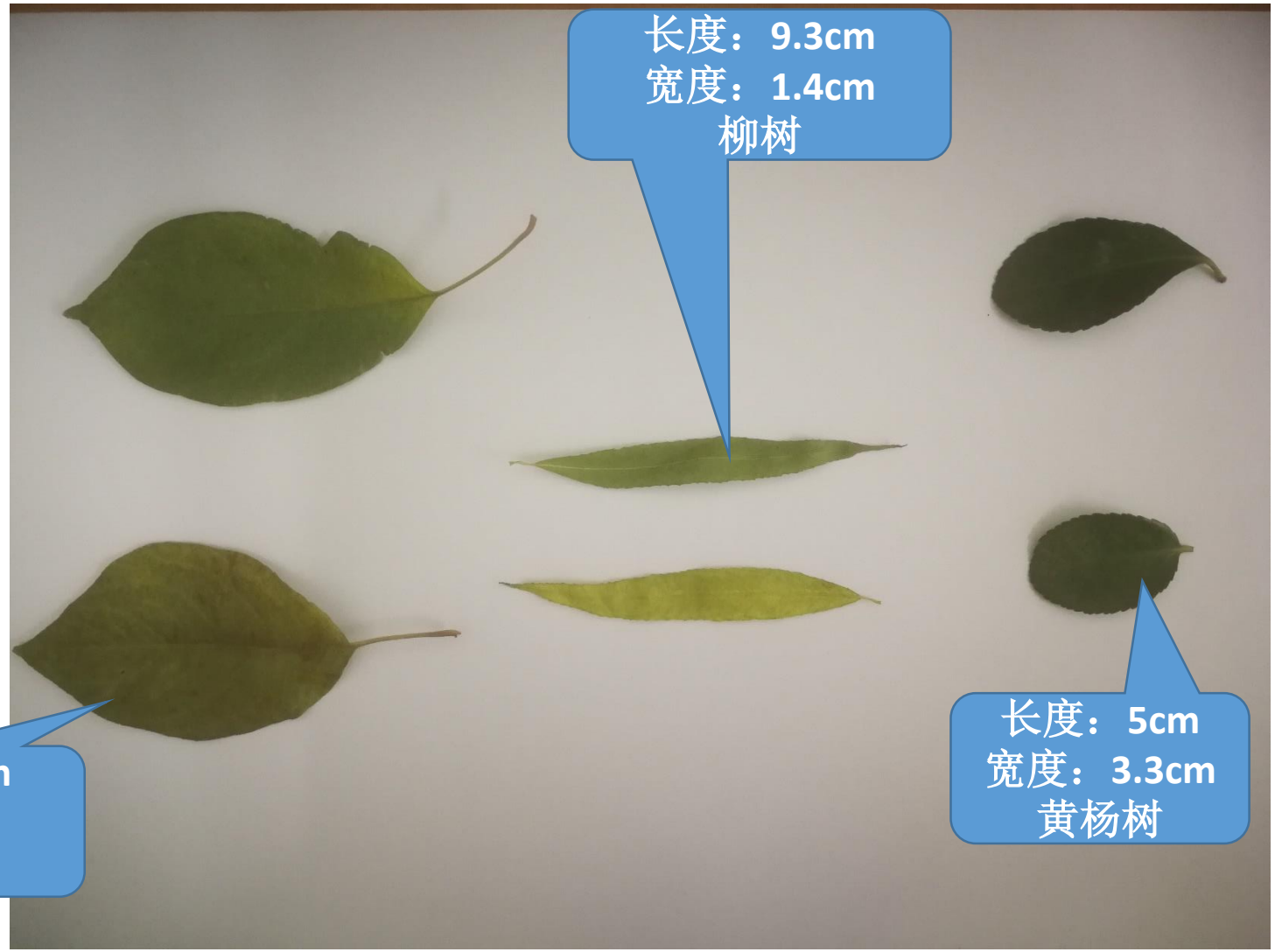
又来了一个叶子

- 问题来了：柳树叶与海棠树叶子的长度基本相同。



解决办法

- 1、使用两个特征向量（长度，宽度）
- 2、增加训练集中样本的数量
- 3、测试寻找最佳的K值



实践题目2：

- 1、每种类型的树叶选取至少4个样本添加在表格2训练集中。

表格 2：训练集							
序号	长度	宽度	种类	序号	长度	宽度	种类
1（示例）	9.5	1.5	柳树	7			
2				8			
3				9			
4				10			
5				11			
6				12			

实践题目2：

- 2、每种类型的树叶选取1个样本添加在表格3测试集的左半部中。

表格 3：测试集

序号	长度	宽度	K 值	预测结果（种类）	是否正确
1（示例）	12.3	1.8	3	柳树	正确
			2		
			4		
2					
3					

- 3、打开程序knn02.py，把表格3的样本数据添加到程序代码中。

knn02-1.py - D:/用户/Desktop/机器学习KNN/二维KNN/knn02-1.py (3.6.3)

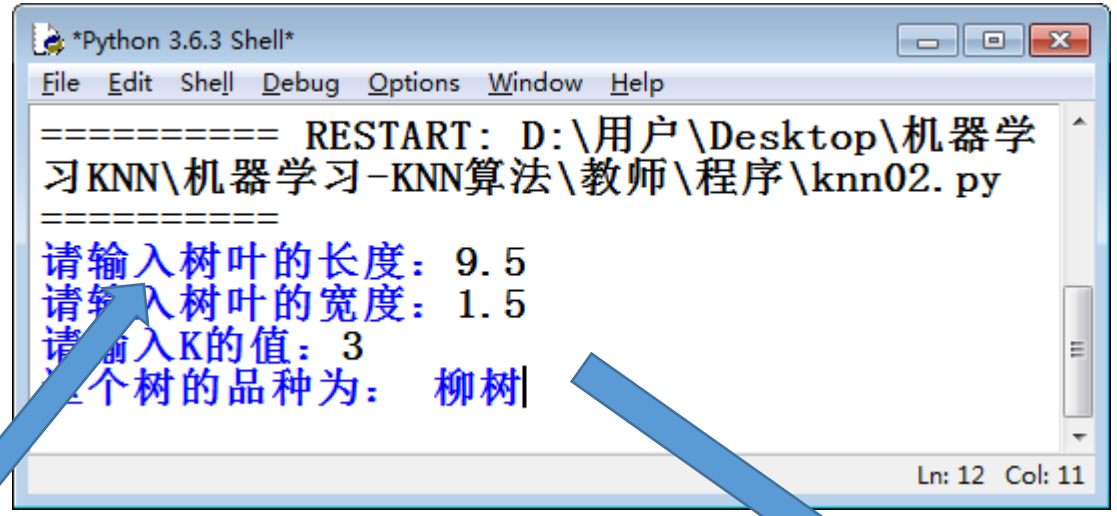
File Edit Format Run Options Window Help

```
import numpy as np
import operator
import matplotlib
import matplotlib.pyplot as plt
from os import listdir
def Create_DataSet():
    group = np.array([[5.0,3.1],[5.5,3.0],[9.8,5.8],[10.5,6.4],[9.8,2.4],[10.3,1.8]])
    labels = ['黄杨','黄杨','海棠','海棠','柳树','柳树']
    return group,labels
def classify0(inX, dataSet, labels, n_estimators=3):
    dataSetSize = dataSet.shape[0]
    diffMat = np.tile(inX, (dataSetSize,1)) - dataSet
```

表格 1：训练集

序号	长度	宽度	标签	序号	长度	宽度	标签
1（示例）	9.5	1.5	柳树	7			
2				8			
3				9			
4				10			
5				11			
6				12			

- 4、运行程序，输入表格3训练集中的样本数据，选择不同的k值进行测试，把测试结果添加到表格3测试集的右半部分。



表格 3：测试集

序号	长度	宽度	K 值	预测结果（种类）	是否正确
1（示例）	12.3	1.8	3	柳树	正确
			2		
			4		
2					
3					

谢谢！

作者：张文轩
单位：北京市第三十五中学
电话：13401082207
邮箱：housemanzwx@sina.com
分享：可以现场分享