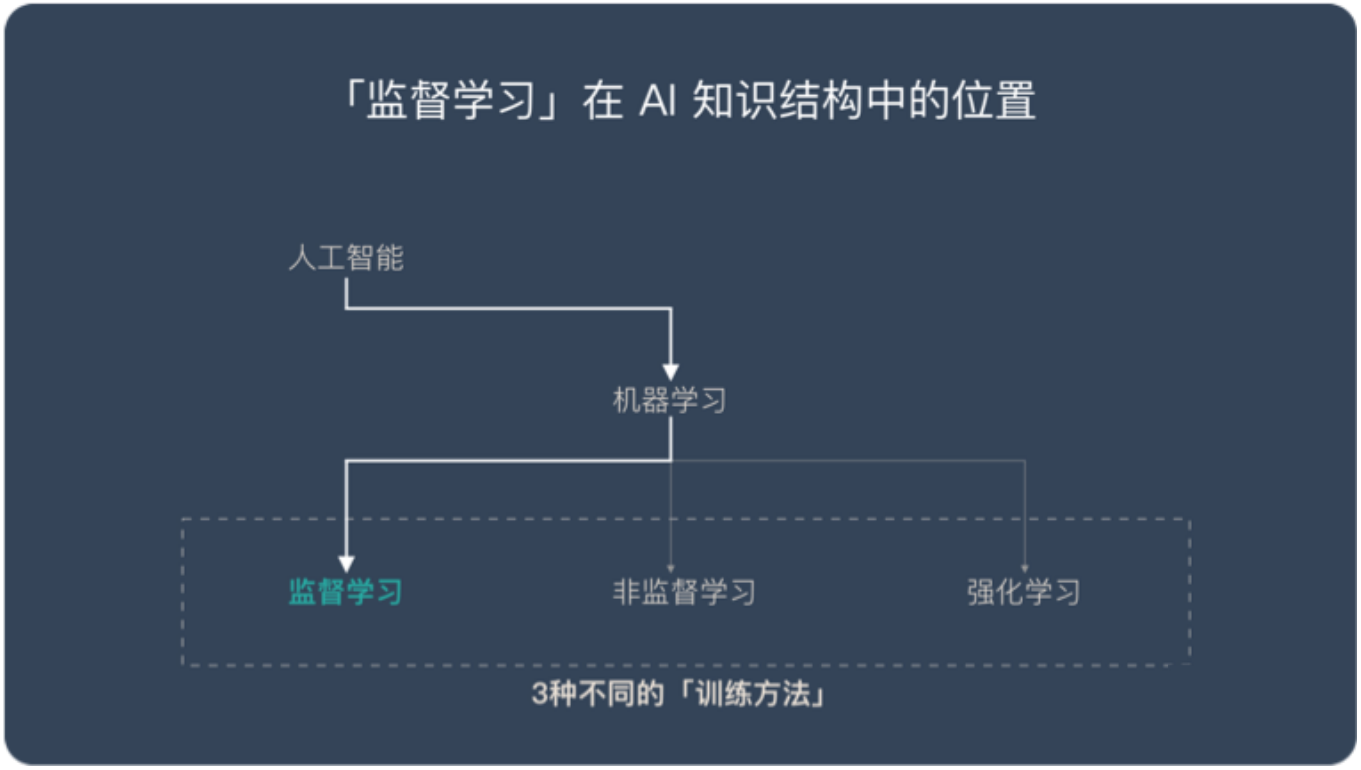
基本分类器之
最近邻与决策树

有监督学习

- 监督学习是机器学习中的一种**训练方式/学习方式**：



有监督学习流程

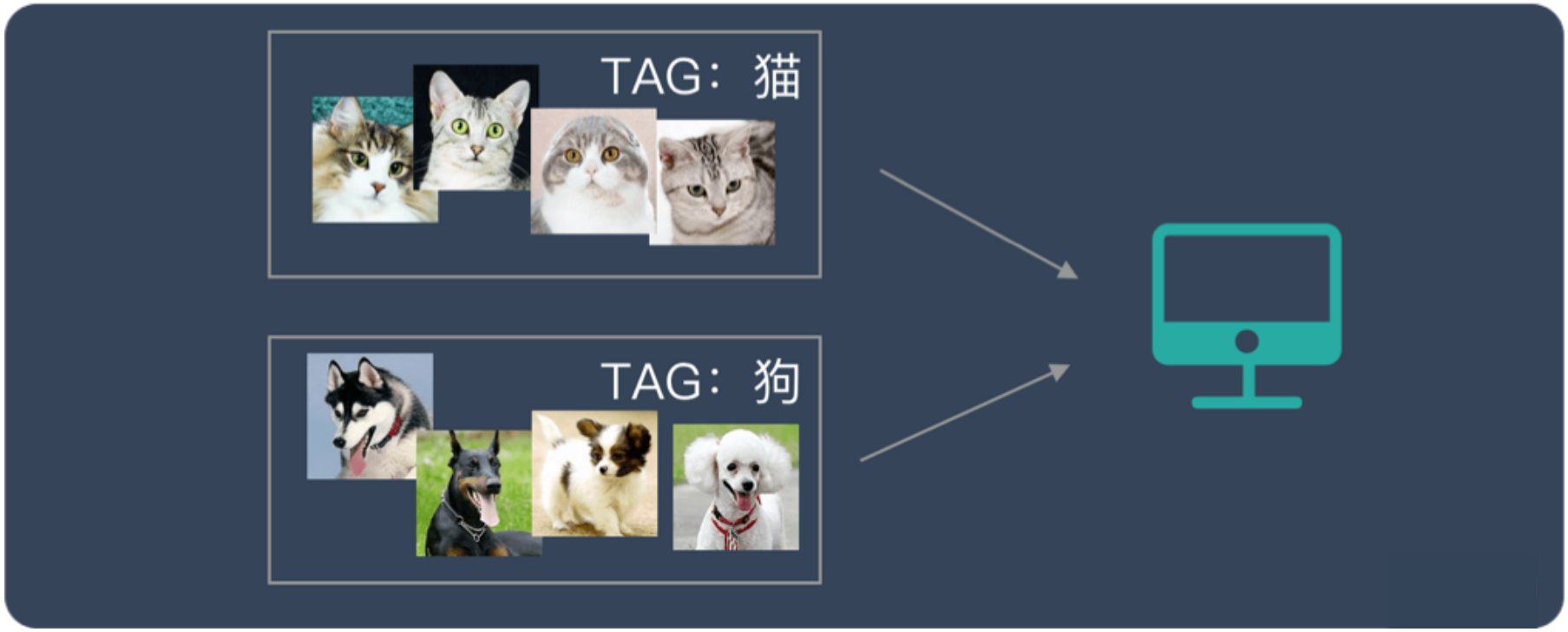
有监督学习流程例子（文章分类）

1. 选择一个合适的数学模型
2. 把一堆已经分好类的文章和他们的分类给机器
3. 机器学会了分类的“方法论”
4. 机器学会后，再丢给他一些新的文章（不带分类），让机器预测这些文章的分类

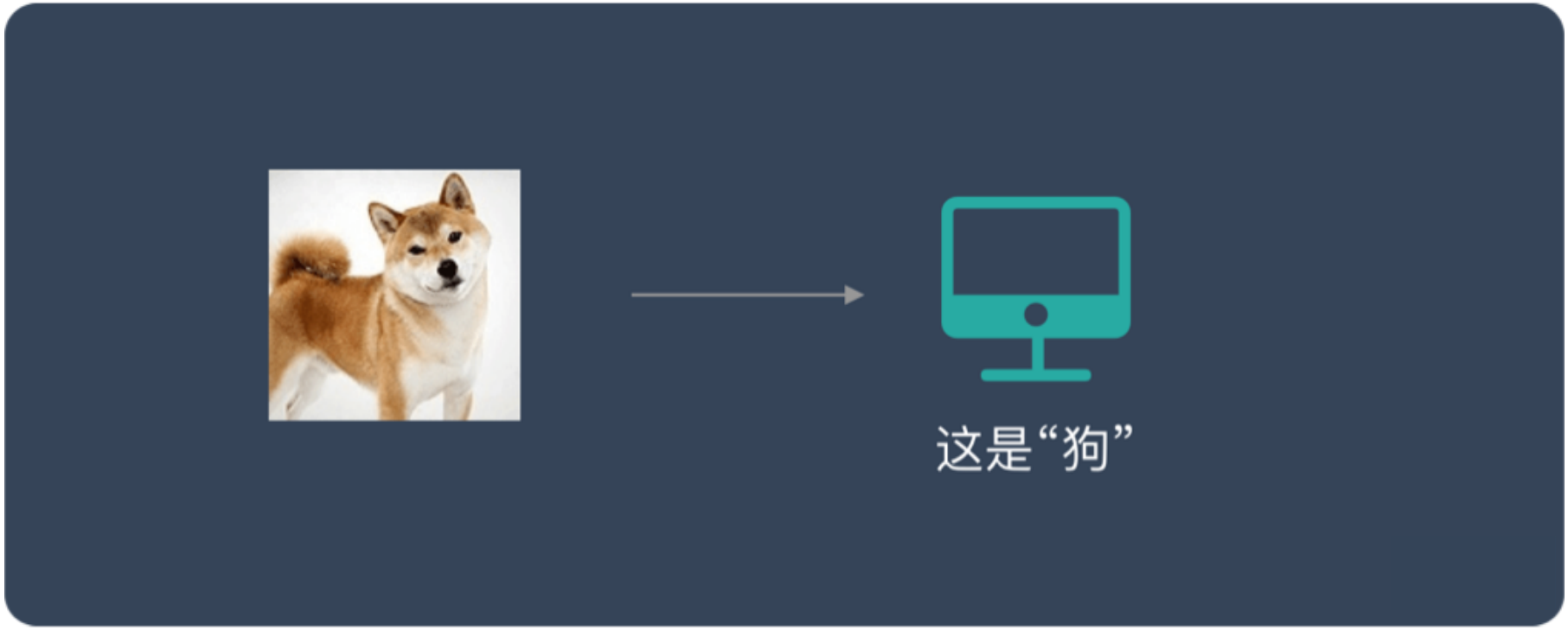
监督学习的2个任务：回归、分类

- 监督学习有2个主要的任务：
  - 回归：**预测连续的、具体的数值。**比如：支付宝里的芝麻信用分数
  - 分类：**对各种事物分门别类，用于离散型预测**

有监督学习（猫狗分类）



对未知照片进行分类



机器学习实操的7个步骤

分类案例

案例：机器学习如何区分啤酒和红酒

步骤1：收集数据

颜色	酒精度	种类
610	5	啤酒
599	13	红酒
693	14	红酒
…	…	…

步骤2：数据准备


将数据分为训练集、验证集、测试集
训练集
60%
验证集
20%
测试集
20%
用来训练模型
确保模型没有过拟合
用来评估模型效果

步骤3：选择一个模型


构建 FICO 的模型
Y = f (A、B、C、D、E)
Y：个人信用评分
A：付款记录
B：账户总金额
C：信用记录跨度
D：新账户
E：信用类别

步骤4：训练

- 大部分人都认为这个是最重要的部分，其实并非如此。数据数量和质量、还有模型的选择比训练本身重要更多（训练知识台上的3分钟，更重要的是台下的10年功）。
- 这个过程就不需要人来参与的，机器独立就可以完成，整个过程就好像是在做算术题。因为机器学习的本质就是**将问题转化为数学问题，然后解答数学题的过程**。

步骤5：评估

- 一旦训练完成，就可以评估模型是否有用。这是我们之前预留的验证集和测试集发挥作用的地方。评估的指标主要有 准确率、召回率、F值。
- 这个过程可以让我们看到模型如何对尚未看到的数是如何做预测的。这意味着代表模型在现实世界中的表现。

步骤6：参数调整

- 完成评估后，您可能希望了解是否可以以任何方式进一步改进训练。我们可以通过调整参数来做到这一点。当我们进行训练时，我们隐含地假设了一些参数，我们可以通过认为的调整这些参数让模型表现的更出色。

步骤7：预测

- 我们上面的6个步骤都是为了这一步来服务的。这也是机器学习的价值。这个时候，当我们买来一瓶新的酒，只要告诉机器他的颜色和酒精度，他就会告诉你，这时啤酒还是红酒了。

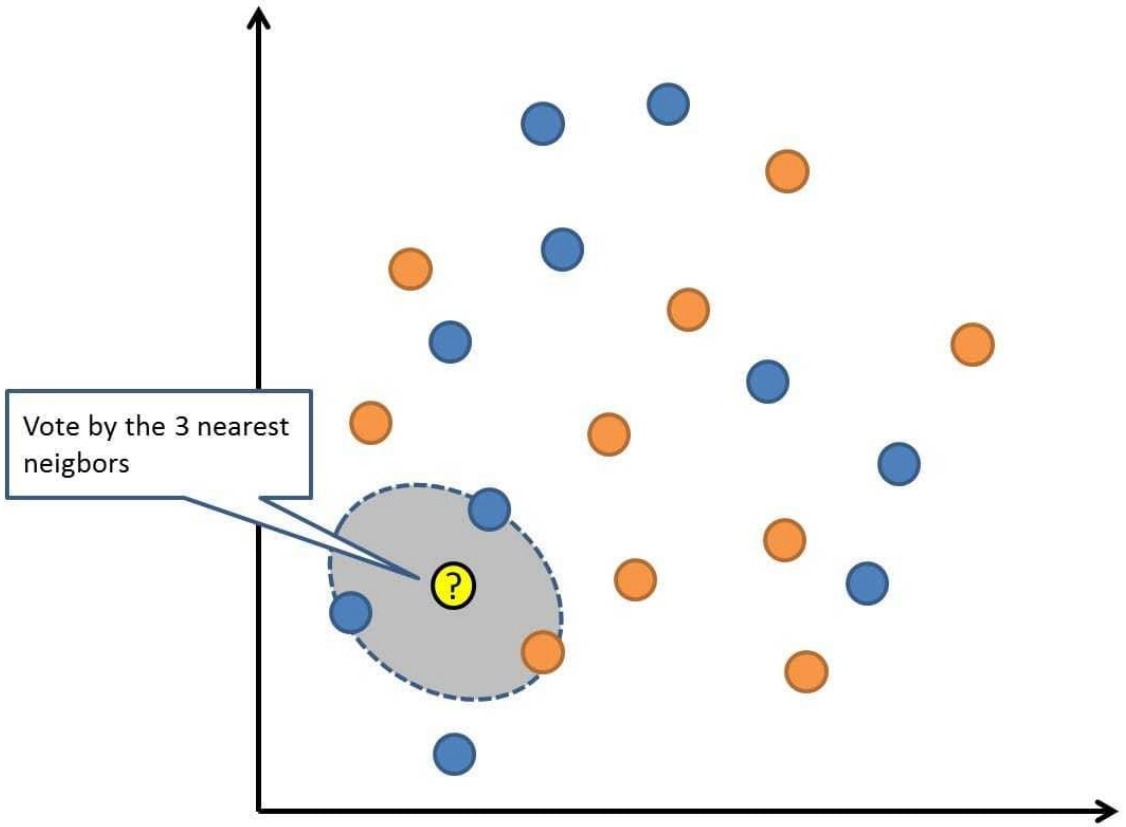
经典分类算法

算法	类型	简介
朴素贝叶斯	分类	贝叶斯分类法是基于贝叶斯定定理的统计学分类方法。它通过预测一个给定的元组属于一个特定类的概率，来进行分类。朴素贝叶斯分类法假定一个属性值在给定类的影响独立于其他属性的 —— 类条件独立性。
决策树	分类	决策树是一种简单但广泛使用的分类器，它通过训练数据构建决策树，对未知的数据进行分类。
SVM	分类	支持向量机把分类问题转化为寻找分类平面的问题，并通过最大化分类边界点距离分类平面的距离来实现分类。
逻辑回归	分类	逻辑回归是用于处理因变量为分类变量的回归问题，常见的是二分类或二项分布问题，也可以处理多分类问题，它实际上是属于一种分类方法。
K邻近	分类+回归	通过搜索K个最相似的实例（邻居）的整个训练集并总结那些K个实例的输出变量，对新数据点进行预测。
Adaboosting	分类+回归	Adaboost目的就是从训练数据中学习一系列的弱分类器或基本分类器，然后将这些弱分类器组合成一个强分类器。
神经网络	分类+回归	它从信息处理角度对人脑神经元网络进行抽象， 建立某种简单模型，按不同的连接方式组成不同的网络。

什么是 K邻近算法？

- 通过搜索K个最相似的实例（邻居）的整个训练集并总结那些K个实例的输出变量，对新数据点进行预测。对于回归问题，这可能是平均输出变量，对于分类问题，这可能是模式（或最常见）类值。
- 关键在于如何确定数据实例之间的相似性。

什么是 K邻近算法？



K邻近算法的优缺点

优点

- 理论成熟，思想简单，既可以用来做分类也可以用来做回归；
- 可用于非线性分类；
- 训练时间复杂度为O(n)；
- 对数据没有假设，准确度高，对outlier不敏感；
- KNN是一种在线技术，新数据可以直接加入数据集而不必进行重新训练；
- KNN理论简单，容易实现；

K邻近算法的优缺点

缺点

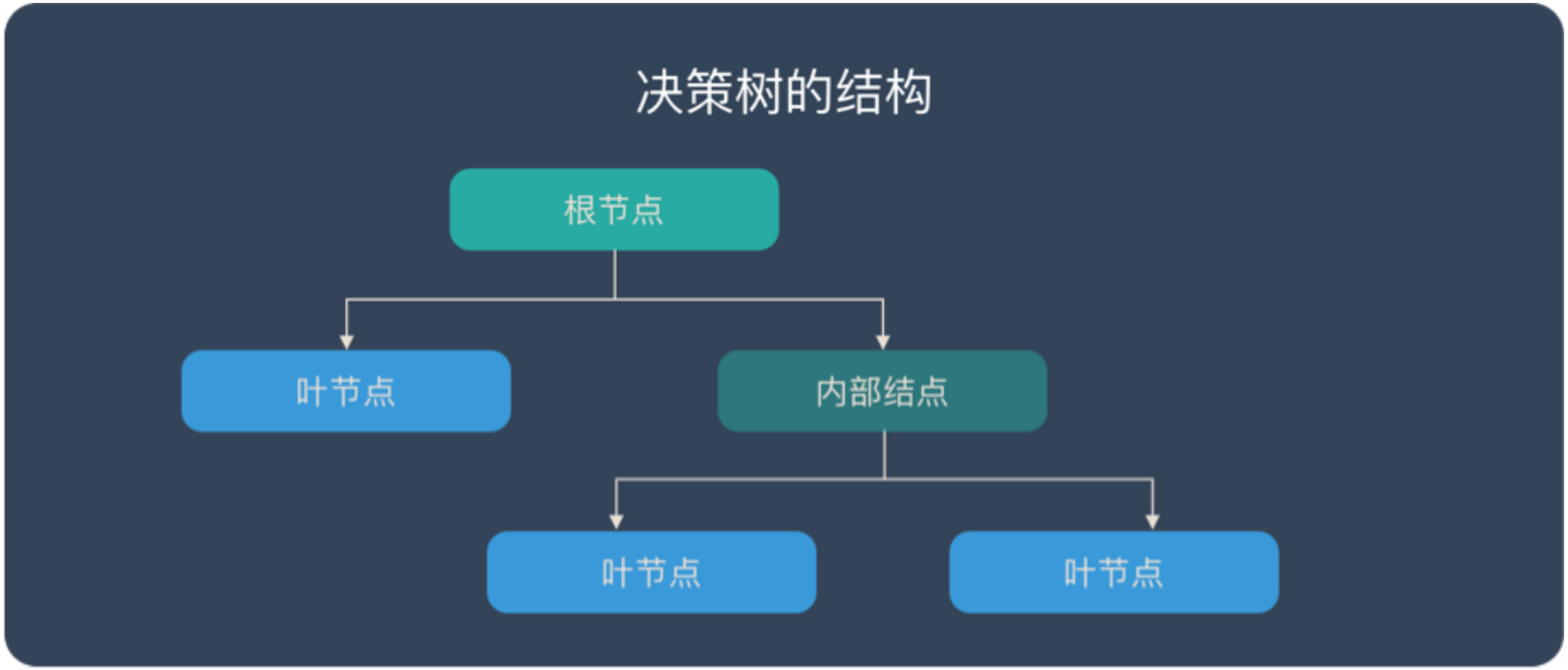
- 样本不平衡问题（即有些类别的样本数量很多，而其它样本的数量很少）效果差；
- 需要大量内存；
- 对于样本容量大的数据集计算量比较大（体现在距离计算上）；
- 样本不平衡时，预测偏差比较大。如：某一类的样本比较少，而其它类样本比较多；
- KNN每一次分类都会重新进行一次全局运算；
- k值大小的选择没有理论选择最优，往往是结合K-折交叉验证得到最优k值选择；

什么是决策树？

决策树算法采用树形结构，使用层层推理来实现最终的分类。决策树由下面几种元素构成：

- 根节点：包含样本的全集
- 内部节点：对应特征属性测试
- 叶节点：代表决策的结果

决策树的结构



决策树特点

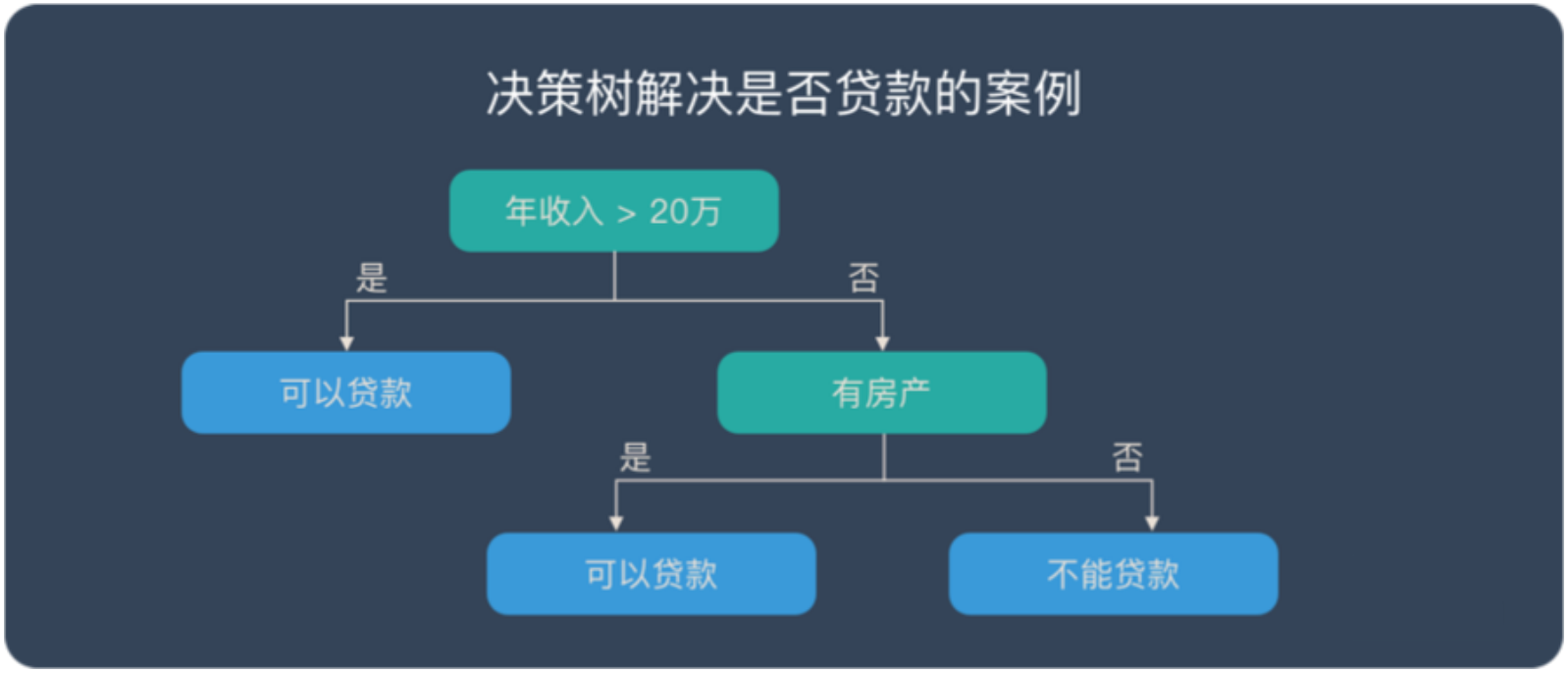
- 预测时，在树的内部节点处用某一属性值进行判断，根据判断结果决定进入哪个分支节点，直到到达叶节点处，得到分类结果。
- 这是一种基于 if-then-else 规则的有监督学习算法，决策树的这些规则通过训练得到，而不是人工制定的。
- 决策树是最简单的机器学习算法，它易于实现，可解释性强，完全符合人类的直观思维，有着广泛的应用。

决策树例子

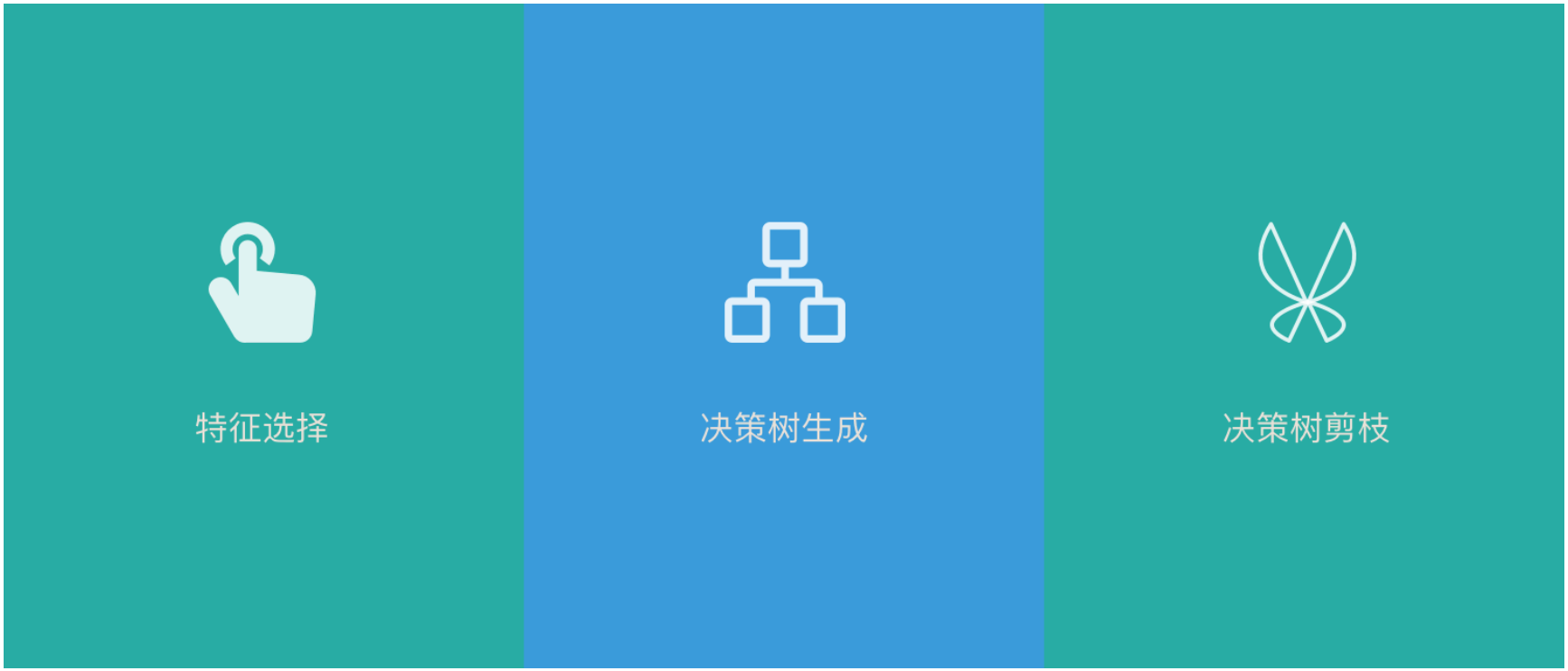
- 需求：
  - 银行要用机器学习算法来确定是否给客户发放贷款，为此需要考察客户的年收入，是否有房产这两个指标。领导安排你实现这个算法，你想到了最简单的线性模型，很快就完成了这个任务。

决策树例子

- 首先判断客户的年收入指标。如果大于20万，可以贷款；否则继续判断。然后判断客户是否有房产。如果有房产，可以贷款；否则不能贷款。



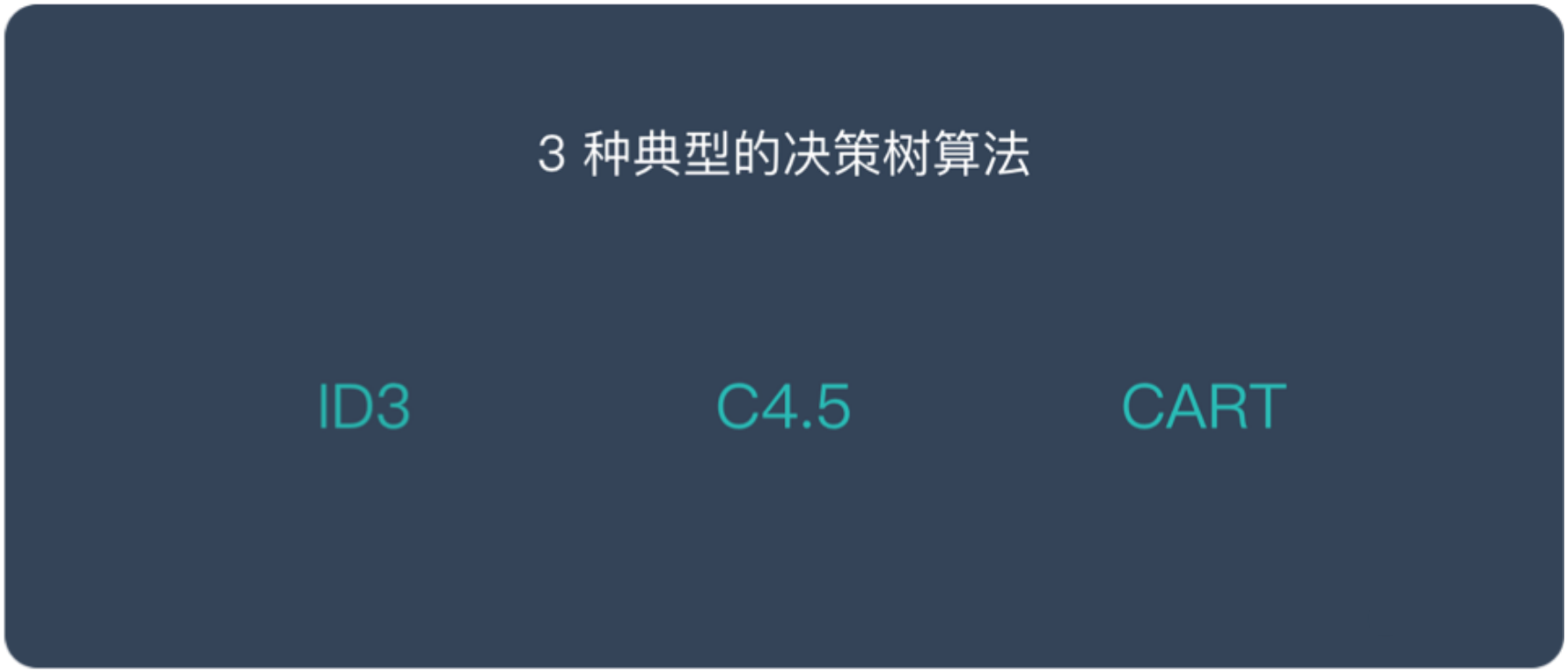
决策树学习的 3 个步骤



决策树学习的 3 个步骤

- **特征选择**
  - 特征选择决定了使用哪些特征来做判断。在训练数据集中，每个样本的属性可能有很多个，不同属性的作用有大有小。因而特征选择的作用就是筛选出跟分类结果相关性较高的特征，也就是分类能力较强的特征。
- **决策树生成**
  - 选择好特征后，就从根节点触发，对节点计算所有特征的信息增益，选择信息增益最大的特征作为节点特征，根据该特征的不同取值建立子节点；对每个子节点使用相同的方式生成新的子节点，直到信息增益很小或者没有特征可以选择为止。
- **决策树剪枝**
  - 剪枝的主要目的是对抗「过拟合」，通过主动去掉部分分支来降低过拟合的风险。

3 种典型的决策树算法



决策树的优缺点

优点

- 决策树易于理解和解释，可以可视化分析，容易提取出规则；
- 可以同时处理标称型和数值型数据；
- 比较适合处理有缺失属性的样本；
- 能够处理不相关的特征；
- 测试数据集时，运行速度比较快；
- 在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

决策树的优缺点

缺点

- 容易发生过拟合（随机森林可以很大程度上减少过拟合）；
- 容易忽略数据集中属性的相互关联；
- 对于那些各类别样本数量不一致的数据，在决策树中，进行属性划分时，不同的判定准则会带来不同的属性选择倾向；信息增益准则对可取数目较多的属性有所偏好（典型代表ID3算法），而增益率准则（CART）则对可取数目较少的属性有所偏好，但CART进行属性划分时候不再简单地直接利用增益率尽心划分，而是采用一种启发式规则）（只要是使用了信息增益，都有这个缺点，如RF）。
- ID3算法计算信息增益时结果偏向数值比较多的特征。