机器学习

第三讲: 决策树

2025 春

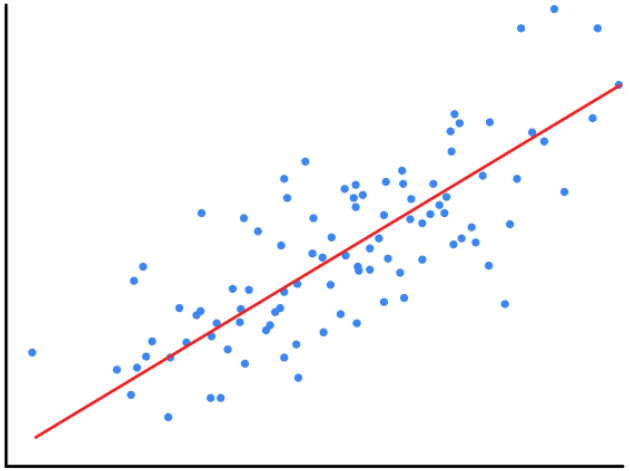
授课老师:顾小东

回忆一下目前为止学了什么？

线性回归模型

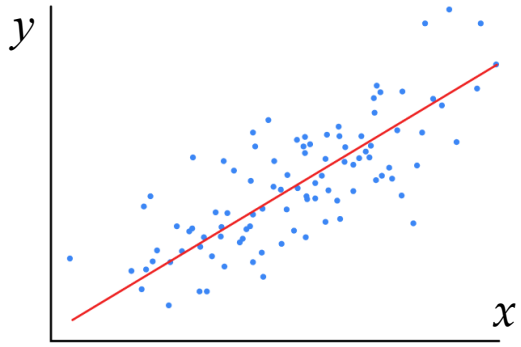
$$y = f(x) = \boldsymbol{w}^T x + w_0$$

characterize the relationship between one or more independent variables and a target variable



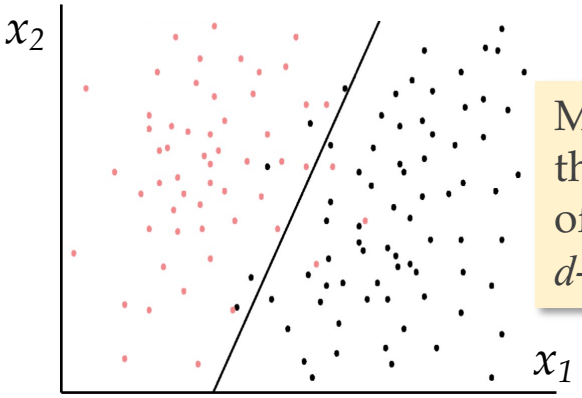
回归 vs. 分类

- 机器学习
  - 监督学习
    - 回归
    - 分类 (✓)
    - …
  - 无监督学习
  - 强化学习



Model the data points spread in (d+1)-dim space.

回归: (predicts real-valued labels)

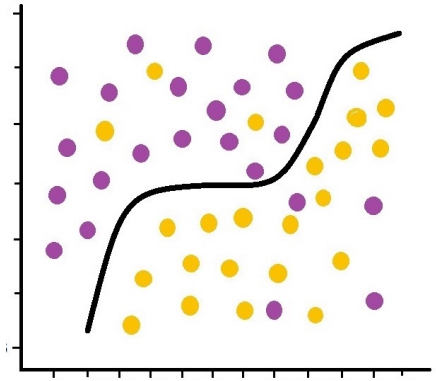


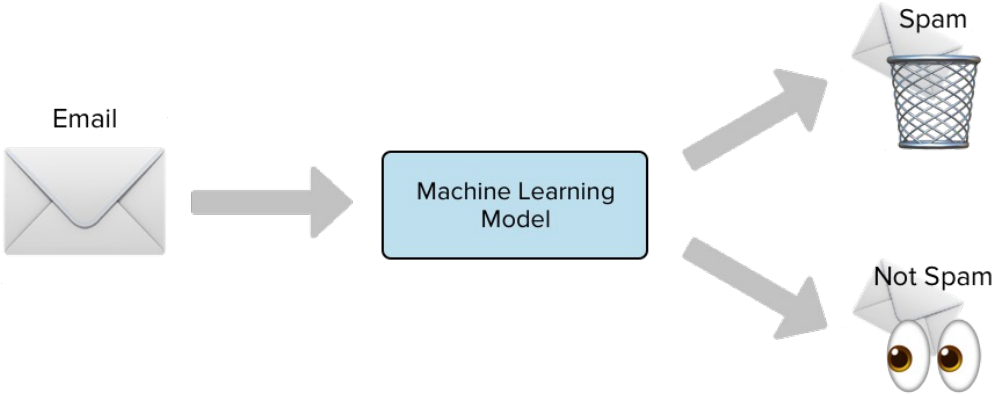
Model the boundaries that separate the data of different labels in *d*-dim space

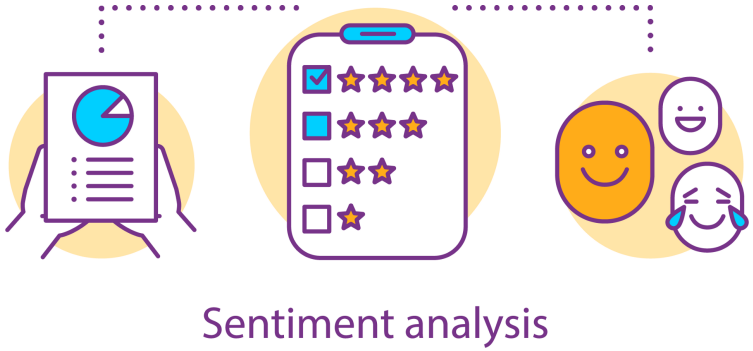
分类: (predicts categorical labels)

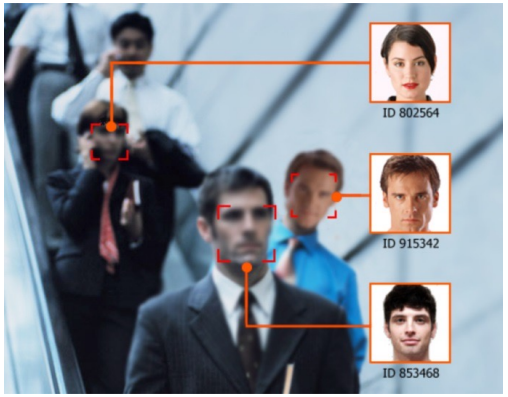
分类问题

机器学习处理的主要任务





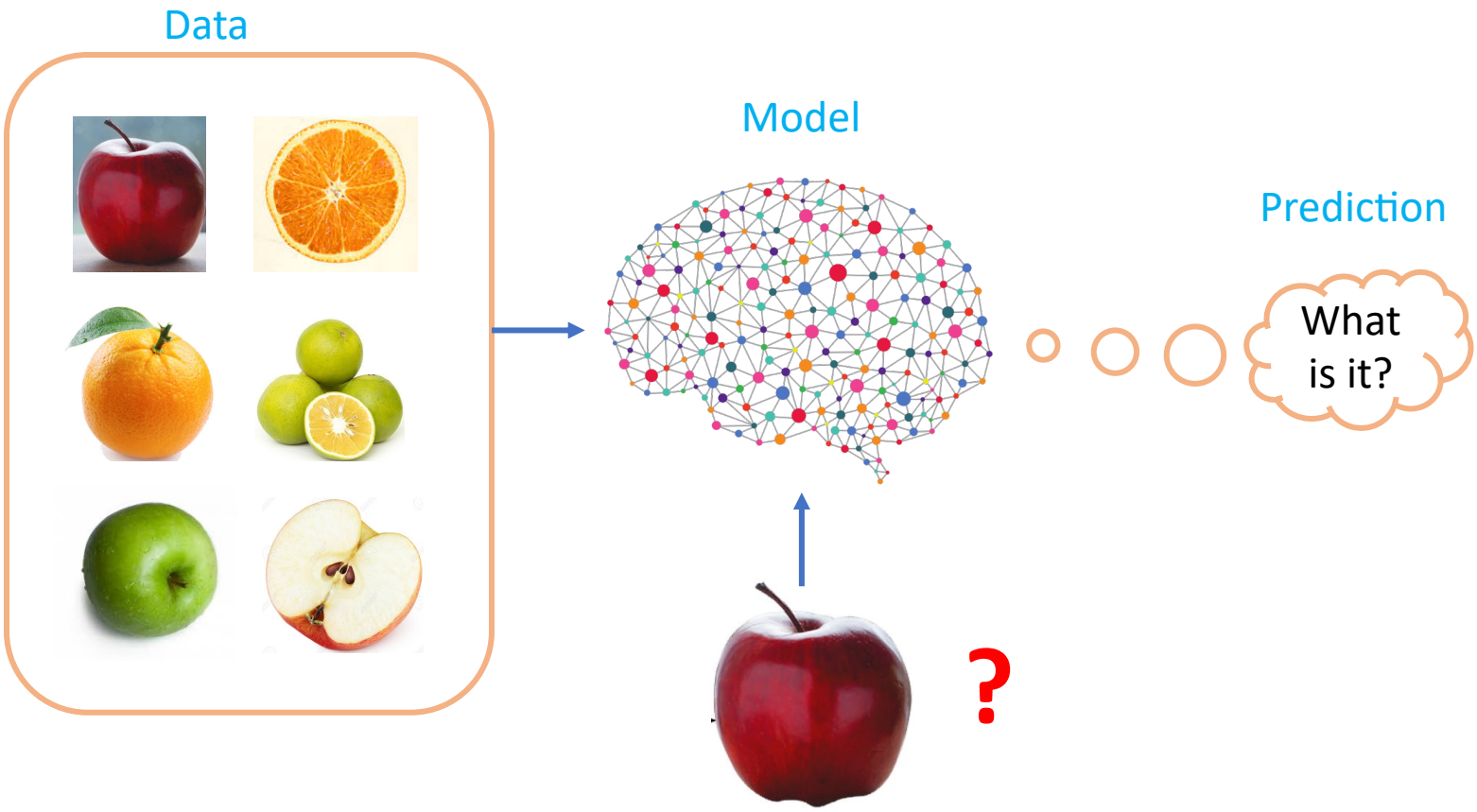




我们从最简单的分类算法开始讲起

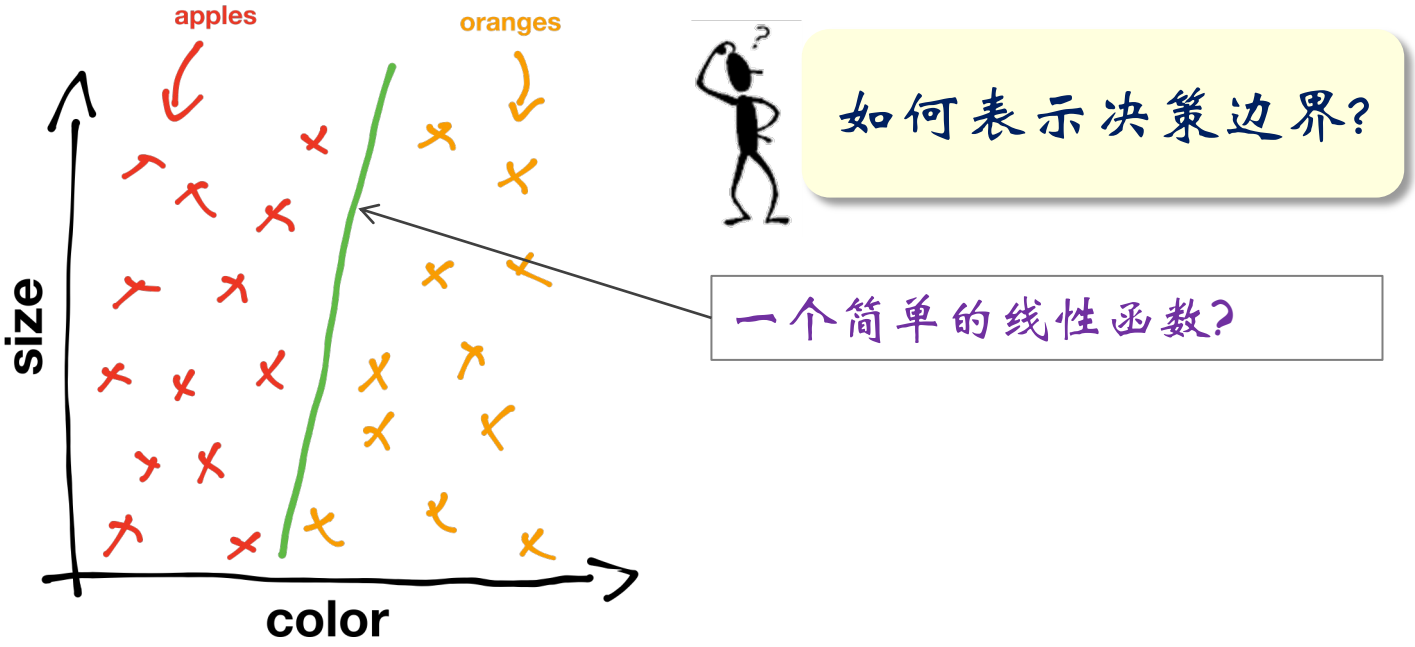
- 决策树
- 随机森林

回顾: 水果分类Hello World

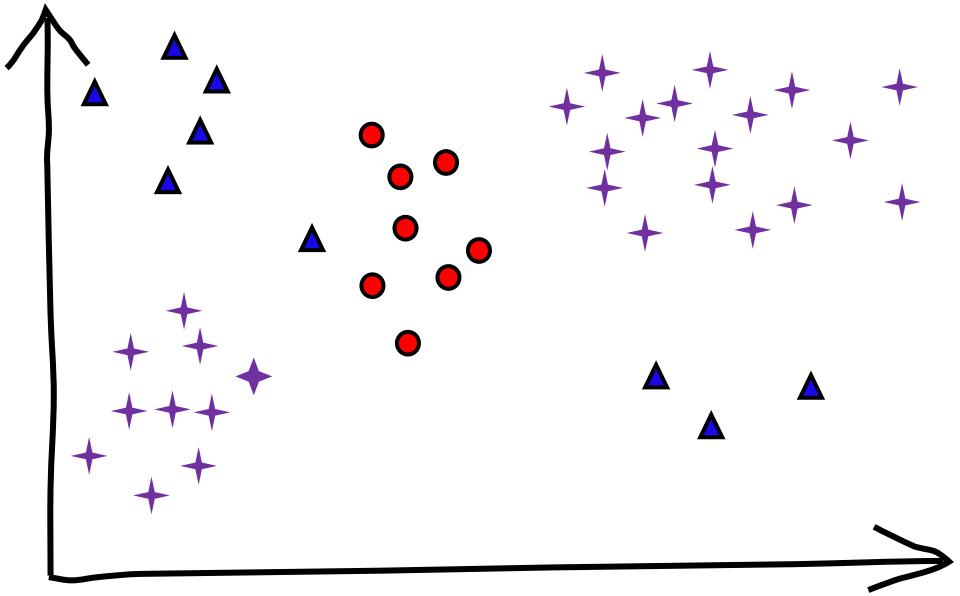


回顾: 水果分类Hello World

分类 = 寻找一个**决策边界**，将特征空间**划分**成2个部分（每部分代表一个水果类型）



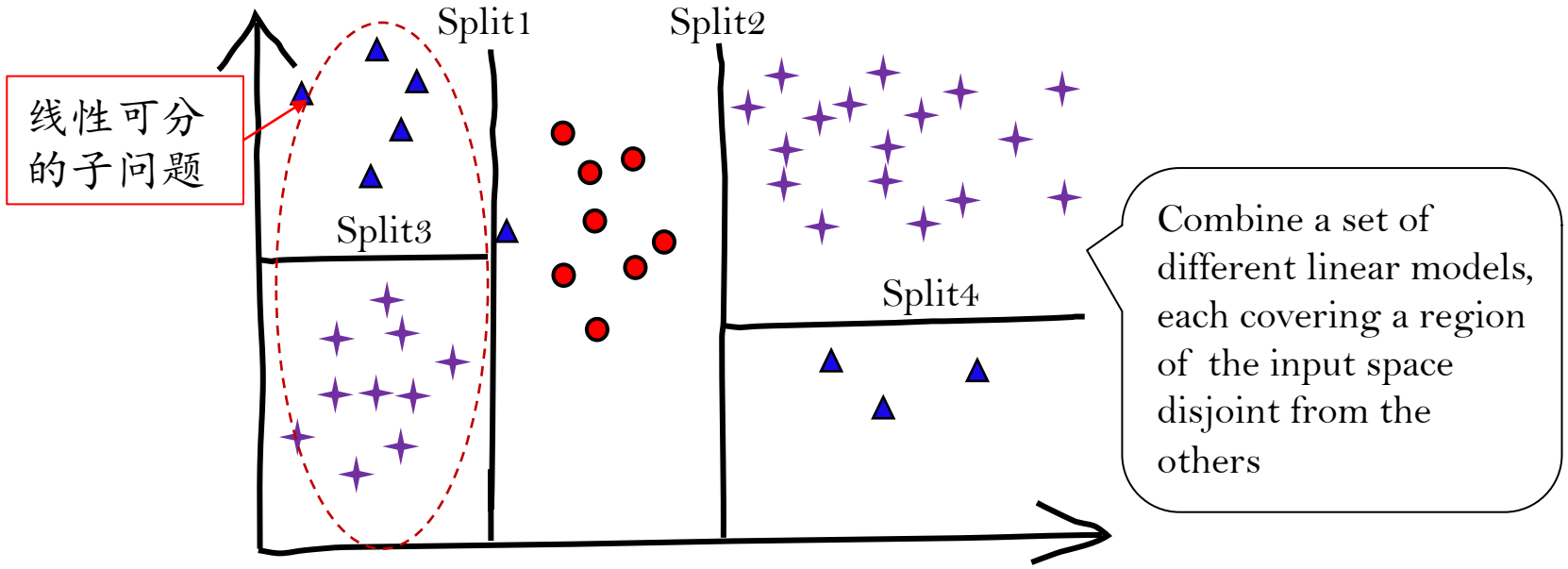
如果数据是这样分布的呢？



- 非线性 (can not be linearly separated)
- 可能非数值 (e.g., “male”, “female”)

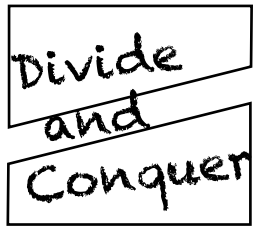
从线性到分段线性

如果数据是这样分布的呢？

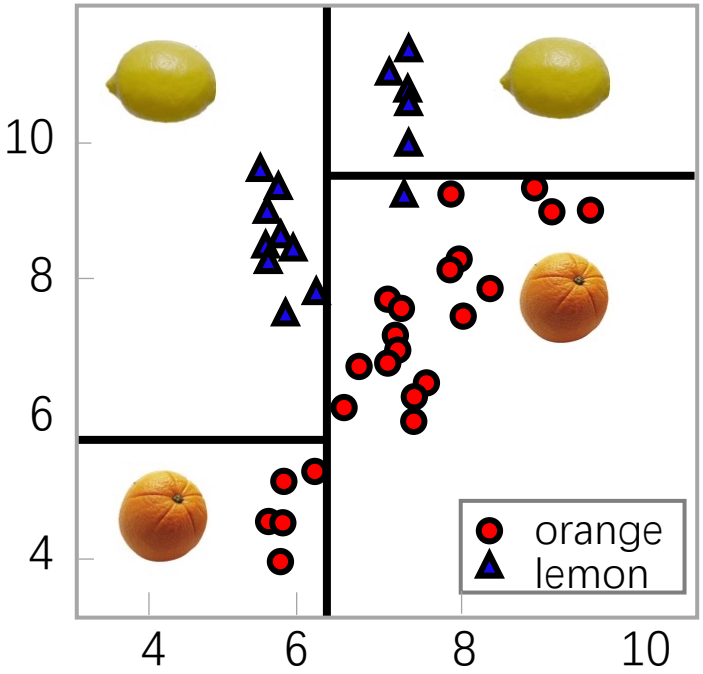


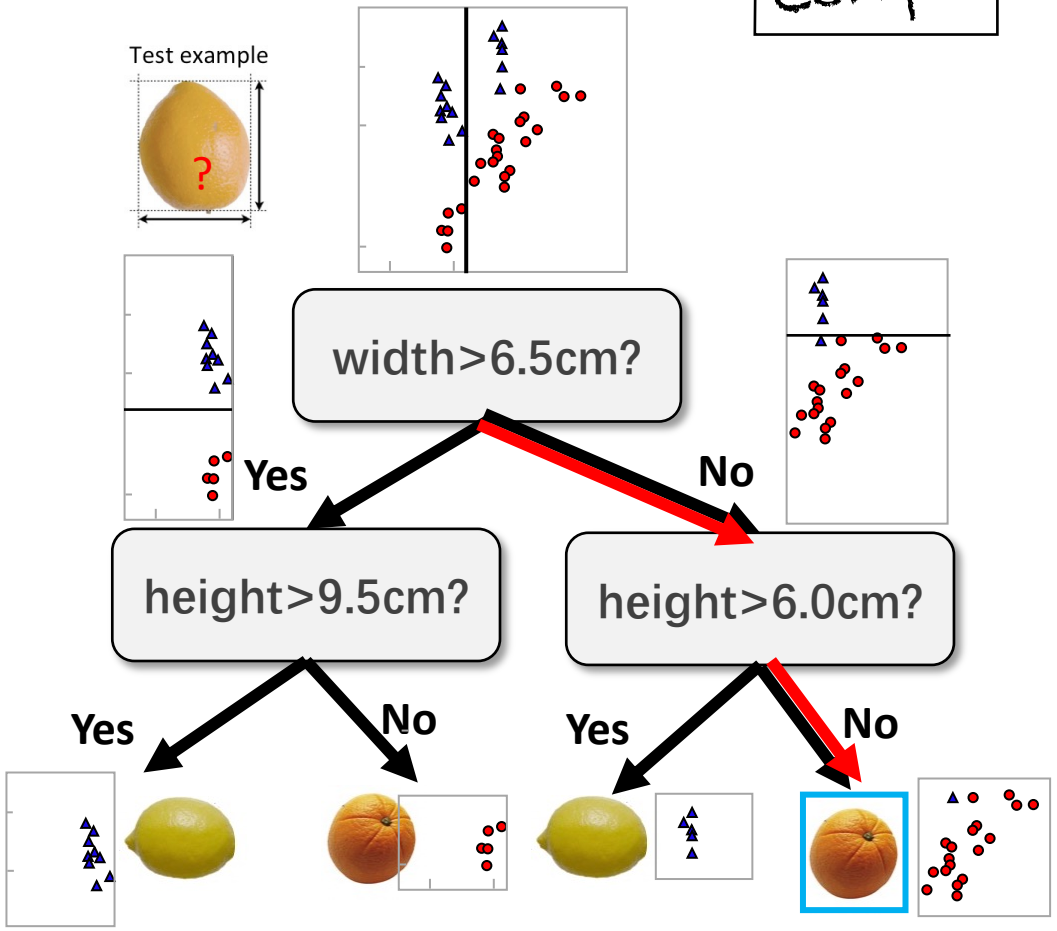
- 非线性 (can not be linearly separated)
- 可能非数值 (e.g., “male”, “female”)
- **但是，沿着某些维度局部线性可分**

决策树的基本思想



- **分而治之**
  - 划分数据属性.
  - 创建if-then规则



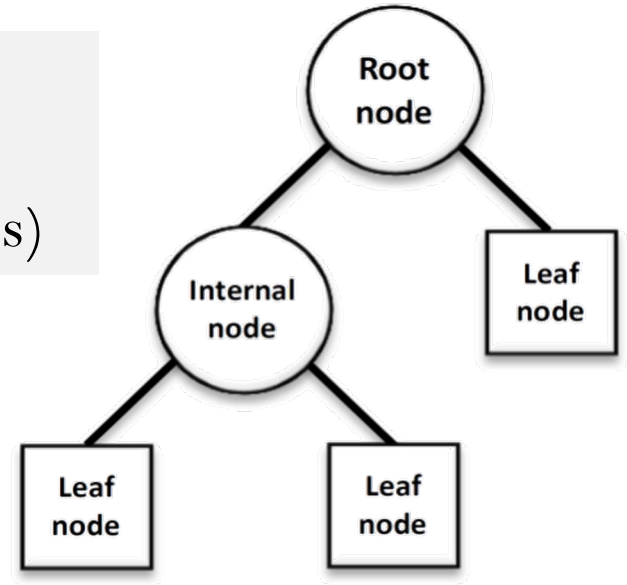


决策树: 通过一系列决策规则对数据进行分类。

模型结构

- 决策树包含三种类型的节点:

1. 根节点(**root node**)
2. 中间节点(**Internal nodes**)
3. 叶子节点(**Leaf nodes**，terminal nodes)



- 每个叶子节点对应一个类别标签
- 每个非叶子节点包含属性测试条件，用来根据属性不同特征划分数据.

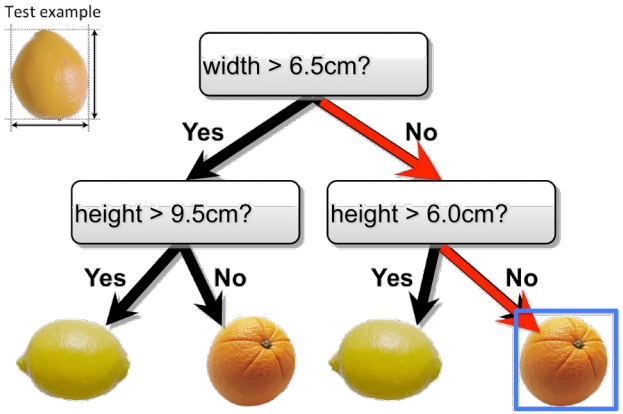
如何构建(训练)一棵决策树?

模型训练 – (构建决策树)

自顶向下分治学习

1. 构造一个根节点，包含整个数据集.
2. 选择一个最合适的属性
3. 根据选择属性的不同取值，将当前节点的样本划分成若干子集；
4. 对每个划分后的子集创建一个孩子节点，并将子集的数据传给该孩子节点；
5. 递归重复2~4直到满足停止条件.

ID	width	height	Type
1	5.2	8.0	lemons
2	6.7	9.8	lemons
3	7.2	7.5	orang
4	6.1	5.3	orang
5	4.1	6.5	lemons

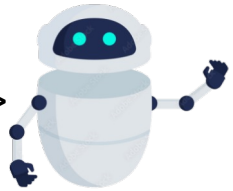


面临问题

- 按照不同划分顺序，同样的数据可以构建多种不同的树！

Q: 如何高效率的构建决策树?

>> >>



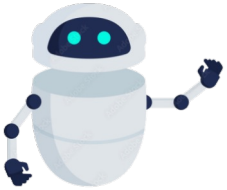
>>

A: 选择每个属性X的时候，尽可能最大化标签Y的纯度

Any decision tree will successively split the data into smaller and smaller subsets. It would be ideal if all the samples associated with a leaf node were from the same class. Such a subset, or node, is considered **pure** in this case.

Q:如何选择属性以最大化标签的纯度?

>> >>



>>

A: 选择X以最大化信息增益、Gini系数等

每一种指标对应一种决策树构造算法

训练算法

ID3 (Information Gain)

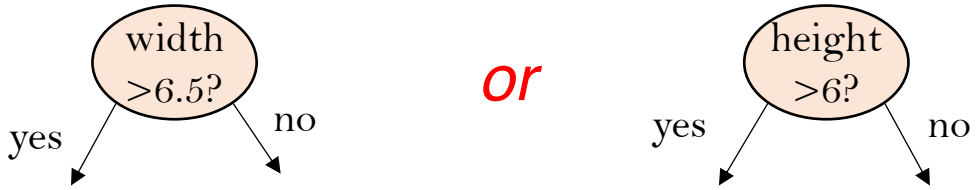
CART (Gini Index)

C4.5 (Gain Ratio)

…

ID3算法

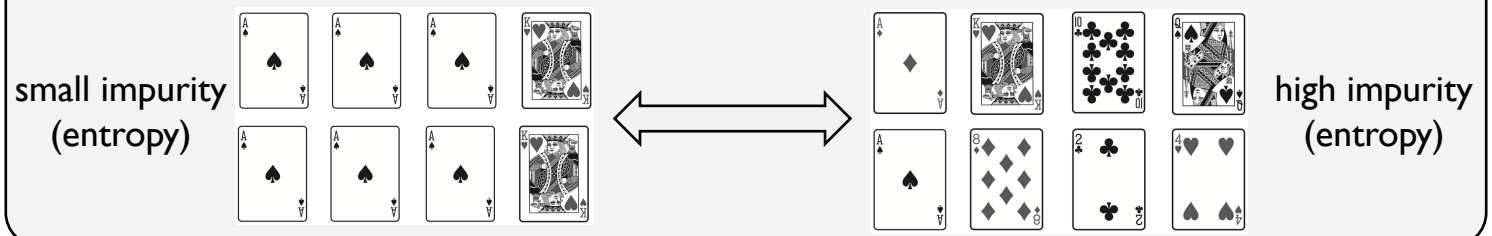
- 划分属性使最大化信息增益 (Information Gain)
  - 试着对每个属性进行划分，看看哪个划分效果“最好”



  - 评估样本的整体信息熵的降低程度

回顾: 熵 – 刻画一个数据集合的不纯净度.

$$H(D) = -\sum_{k=1}^{K} p_k \log p_k$$



ID3算法

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log \frac{|C_k|}{|D|}$$

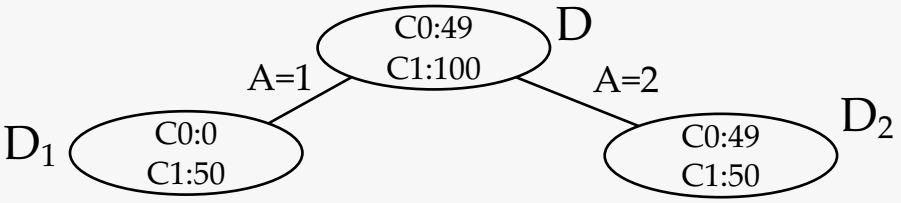
($|C_k|$: # samples of class C_k in dataset D)

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \left(\sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log \frac{|D_{ik}|}{|D_i|} \right)$$

($|D_i|$: # samples whose attribute A is set to the i-th value in D; $|D_{ik}|$: #samples of class C_k in D_i)

Gain $(D,A) = H(D) - H(D|A)$

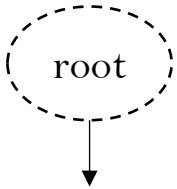
Example: What is the information gain of this split?



- root entropy: $H(D) = -{}^{49}/_{149} \log({}^{49}/_{149}) - {}^{100}/_{149} \log({}^{100}/_{149}) \approx 0.91$
- leaves entropy: $H(D|A=1) = 0,\ H(D|A=2) \approx 1$
- IG(D|A) $\approx 0.91 - (\frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1) \approx 0.24 > 0$

举例

ID	width	height	Type
1	5.2	8.0	lemons
2	6.7	9.8	lemons
3	7.2	7.5	orang
4	6.1	5.3	orang
5	4.1	6.5	lemons



$\{1,2,3,4,5\}$

Step1 – 创建根节点

Step2 – 计算每个(子)数据集的熵.

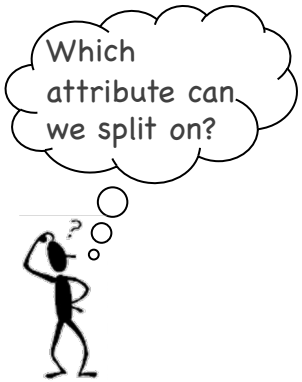
lemon	orange
3	2

$$H(D) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.97$$

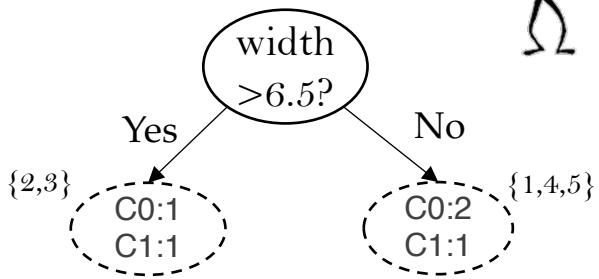
举例

Step 3 – 对每个属性计算信息增益IG, 选择具有最大IG的属性

ID	width	Type
1	5.2	lemons
2	6.7	lemons
3	7.2	orang
4	6.1	orang
5	4.1	lemons

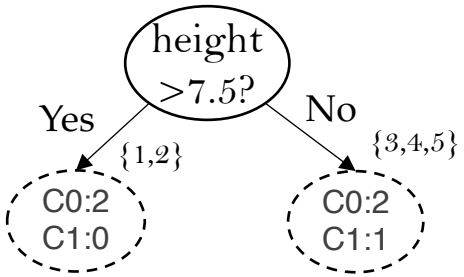


ID	height	Type
1	8.0	lemons
2	9.8	lemons
3	7.5	orang
4	5.3	orang
5	6.5	lemons



H=-½log½-½log½ = 1

H=-⅔log⅔-⅓log⅓ =0.92



H=-1log1-0log0 = 0

H=-⅔log⅔-⅓log⅓ =0.92

H(D | width) = ⅖ H(D | width>6.5) + ⅗H(D | width<6.5)
= ⅖ x 1 + ⅗ x 0.92 = 0.95
Gain(D | width) = H(D) – H(D | width) = 0.97-0.95=0.02

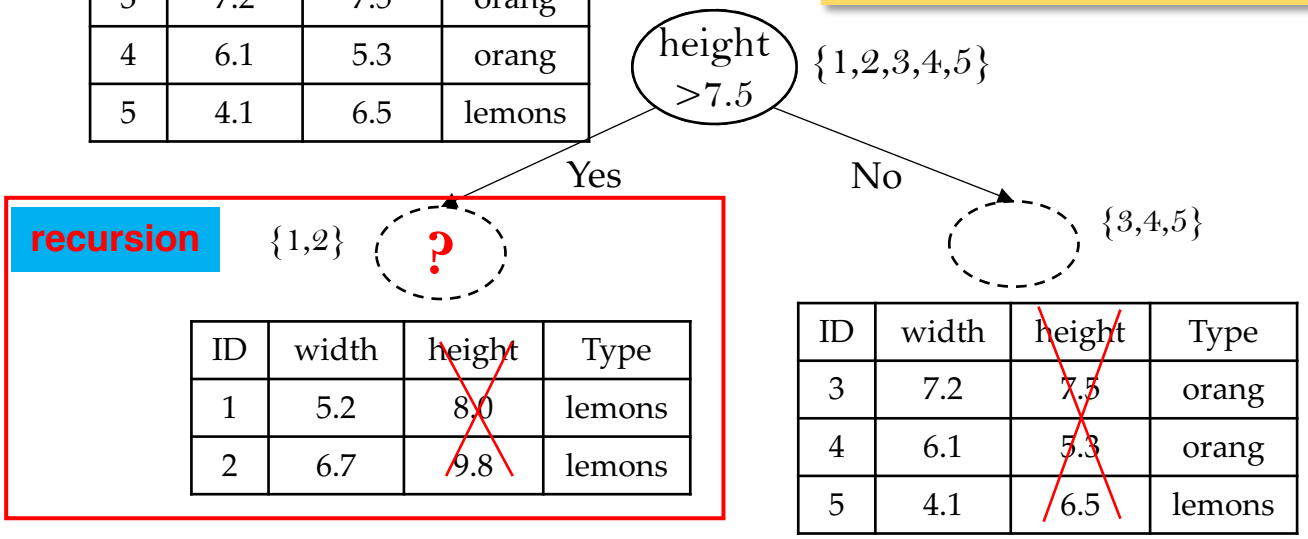
H(D | height) = ⅖ H(D | height>7.5) + ⅗ H(D |height<7.5)
= ⅖ x 0 + ⅗ x 0.92 = 0.55
Gain(D | height) = H(D) – H(D | height) = 0.97-0.55=0.42

举例

Step 4 - Assign the current (root) node with the attribute that has the maximum IG. For each attribute value, expand an outgoing branch to a newly-created node.

ID	width	height	Type
1	5.2	8.0	lemons
2	6.7	9.8	lemons
3	7.2	7.5	orang
4	6.1	5.3	orang
5	4.1	6.5	lemons

Step 5 – Split the dataset along the values of the maxIG attribute and remove this feature from the dataset.



ID	width	~~height~~	Type
1	5.2	~~8.0~~	lemons
2	6.7	~~9.8~~	lemons

ID	width	~~height~~	Type
3	7.2	~~7.5~~	orang
4	6.1	~~5.3~~	orang
5	4.1	~~6.5~~	lemons

Step 6 – For each subset, repeat steps 2-5 until a stopping criteria is satisfied → here the recursion kicks in.

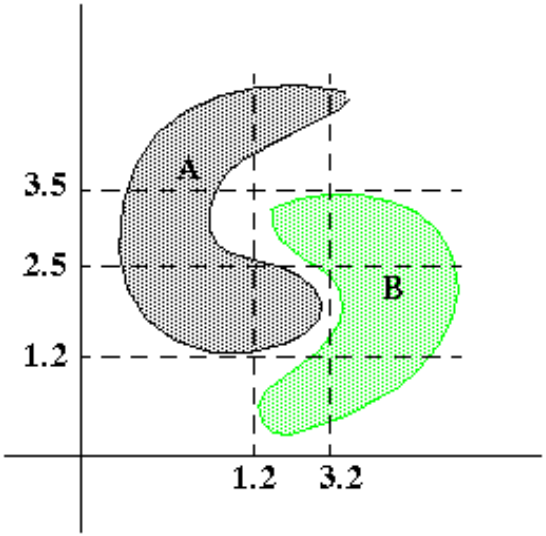
停止划分的条件

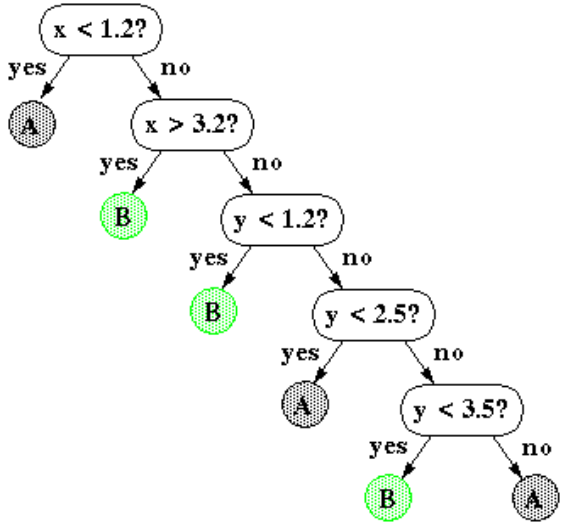
- 纯度
  - 叶子节点包含的样本均属于同一类
- 数据量过小
  - 叶子节点包含的样本数小于一个阈值
- 没有属性可分(ID3)

算法优缺点

优点：

- 易理解，可解释性，便于可视化分析，
- 数据预处理要求低，
- 可以轻易处理非线性边界，
- 数据驱动，可以以任意精度拟合训练集.

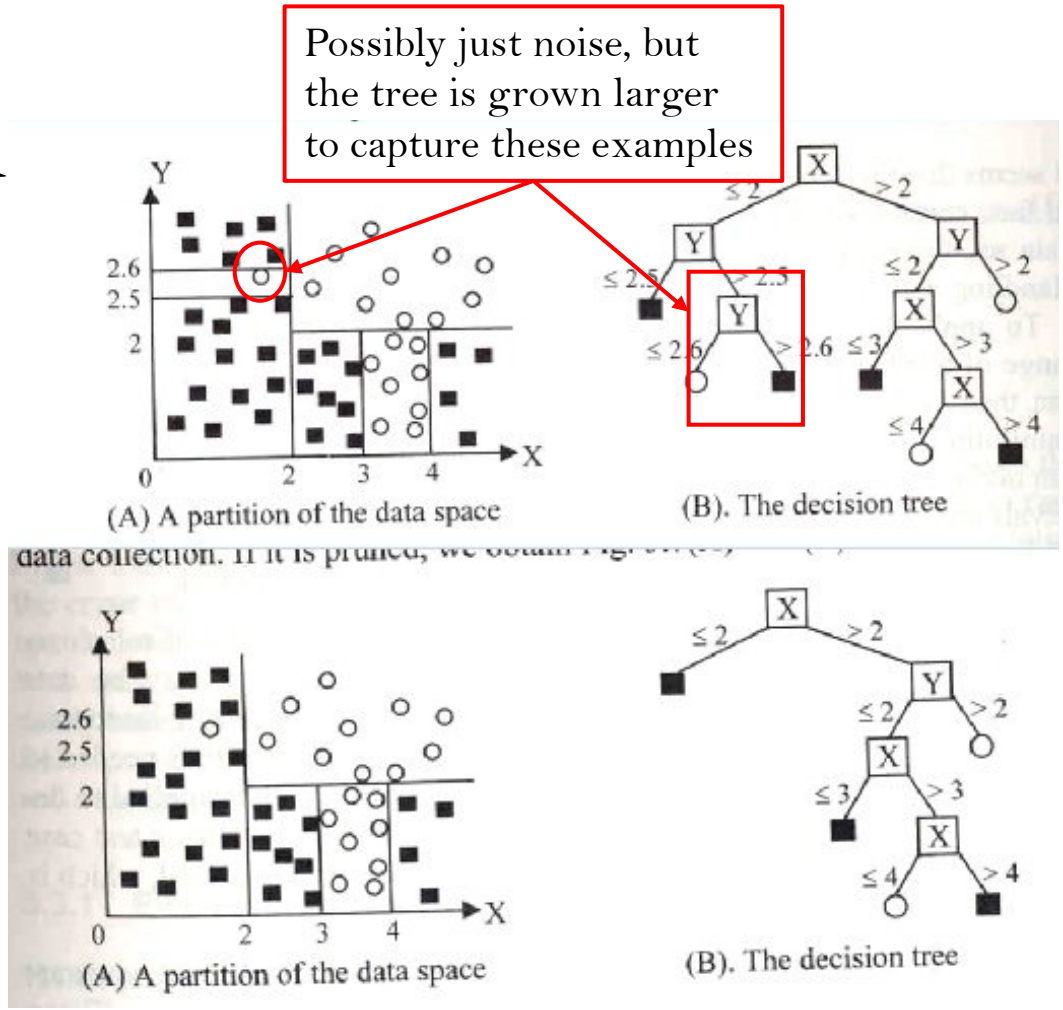




算法优缺点

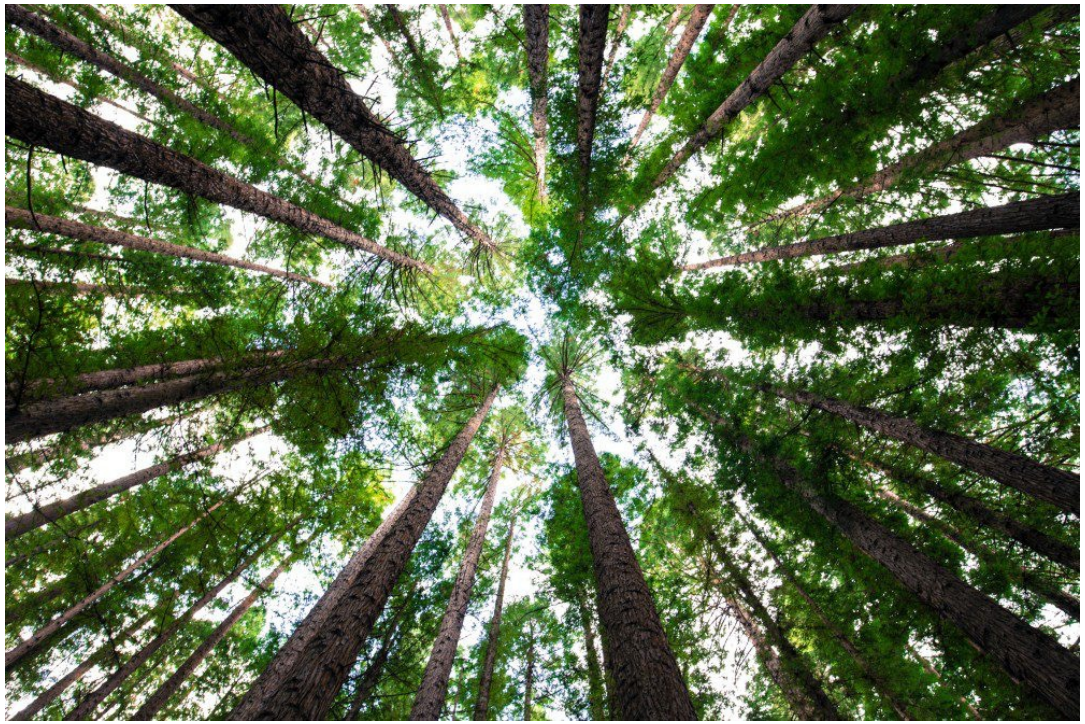
缺点：

- 容易过拟合

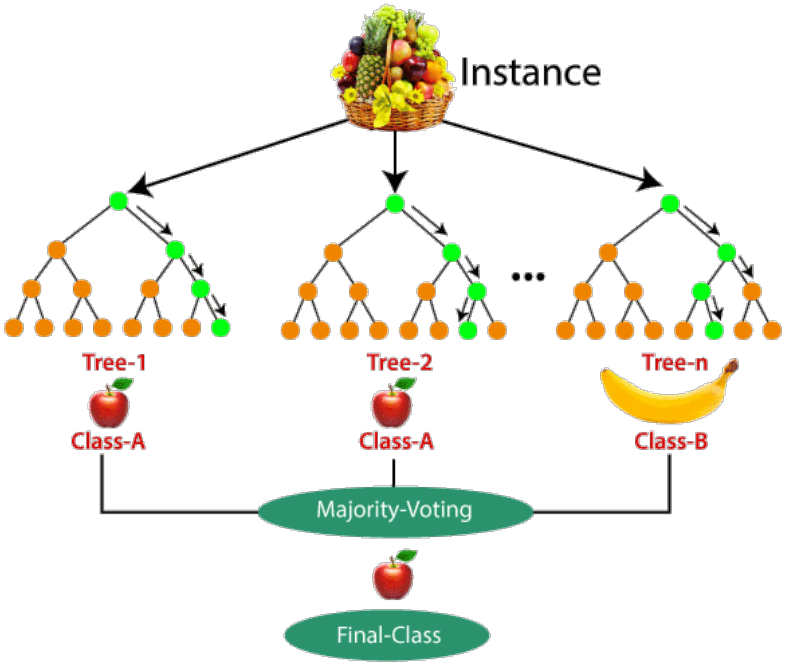


Ensemble of Multiple Decision Trees

- Build multiple decision trees on random subsets of data and attributes, and combine their results. **(三个臭皮匠，顶一个诸葛亮)**



Random Forest



1. Sampling a set of N samples from the original dataset, **putting them back after sampling.**
2. Train a decision tree using the random dataset. For each node:
   a. Randomly sample d attributes
   b. Split nodes based on the selected attributes (e.g., information gain)
3. Repeat 1~2 for k times
4. Aggregate predictions by all decision trees, produce the final results by majority voting.

Random Forest

Advantages:

- More accurate than decision tree.
- Fast, and easy for parallel (independence between trees).
- Can deal with high-dimensional data (i.e., with many attributes) without feature selection (because the subset of attributes for training are randomly selected).
- Can deal with missing attributes.
- Can explain the results (i.e., tell which attributes are more important after training).
- Can identify the correlations between attributes during training.
- Balance errors on unbalanced datasets.
- ...

TIME for Coding

Tutorial: Decision Trees from Scratch with Python

Manually setup using the Jupyter lab online [no sklearn]:
../tutorials/ch3_decision_tree.ipynb
../tutorials/data/zoo.csv

A runnable online tutorial using sklearn:
https://www.kaggle.com/code/prashant111/decision-tree-classifier-tutorial

What's Next?

Bayes Classifier

- A **new perspective** of machine learning and classification.
- Classify data by estimating posterior probability.
  - Bayes Decision
  - Classification using Bayes Net
  - Naïve Bayes