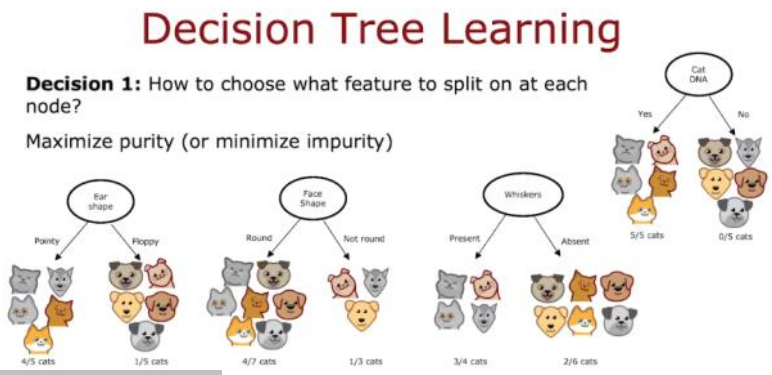
决策树

2023年5月27日　　11:04

决策树学习的任务是，在所有可能的决策树中，选出一个在训练集上表现较好的模型

决策树学习中的决策

1.如何决定每个节点该区分出哪个特征？👉使纯度最大化



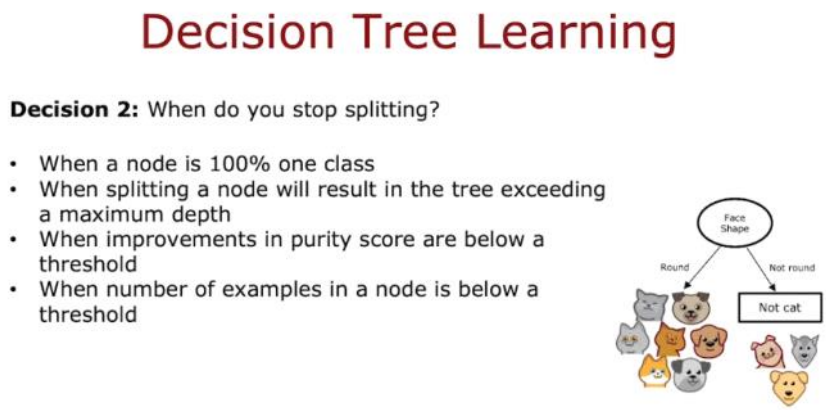
2.何时停止拆分特征？ (后三条主要防止过拟合)

当一个节点已经100%分出一类

继续拆分会导致超出树的深度

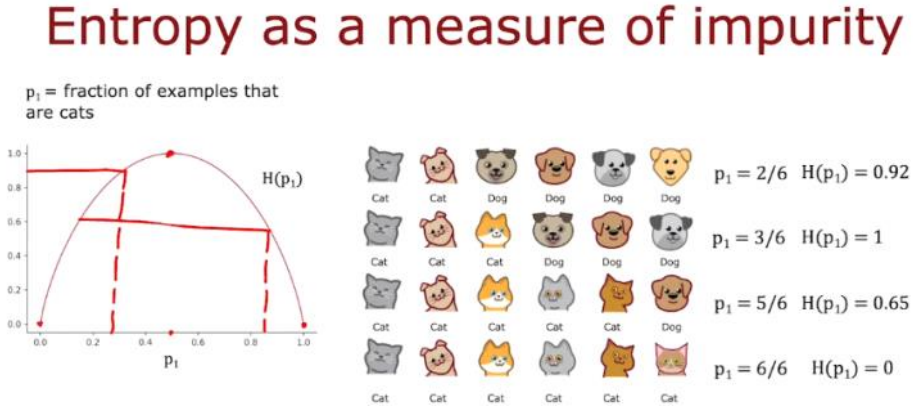
继续拆分对纯度改善较小

节点中案例的数量低于一定阈值

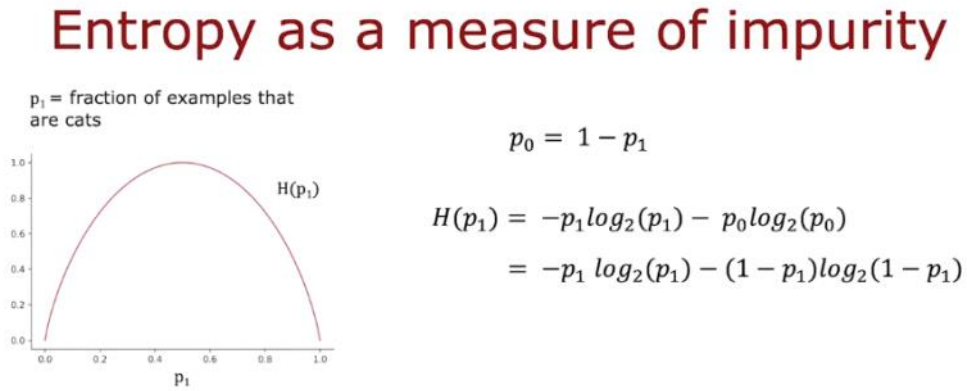


熵(entropy)：A measure of impurity

熵函数在纯度为0.5时达到最大值1.

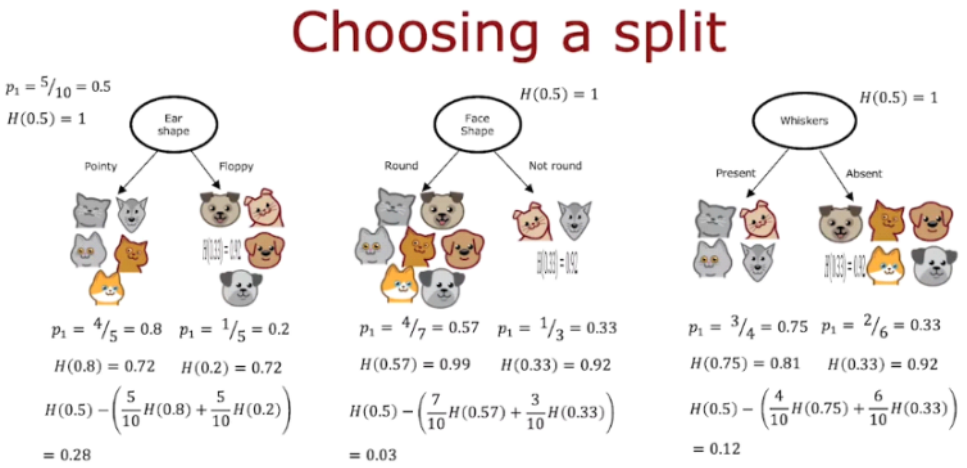


熵函数：



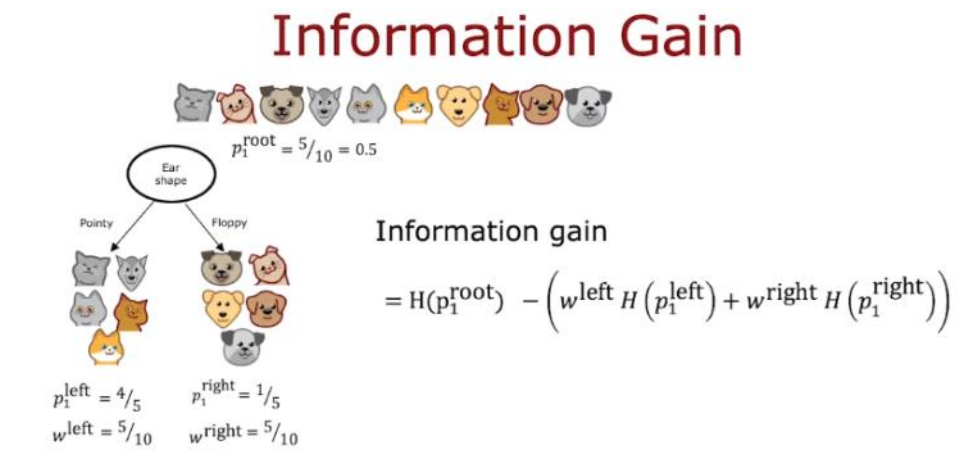
对数函数以2为底，是为了使这个函数在p1=0.5时的值为1

熵的减少也称为信息增益（Information Gain）



决策树中需要的数据并不是左右分支的熵，而是从节点到分支这个过程中产生的信息增益（熵的减少）

其中左右分支的熵是两个分支熵的加权平均，如下：



屏幕剪辑的捕获时间: 2023/5/29 9:25

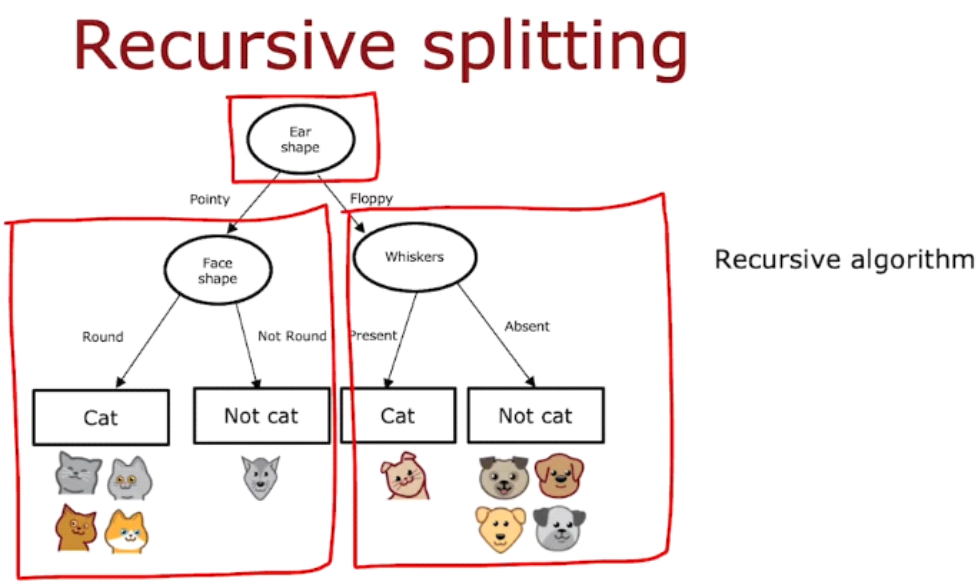
决策树学习

Decision Tree Learning

- Start with all examples at the root node
- Calculate information gain for all possible features, and pick the one with the highest information gain
- Split dataset according to selected feature, and create left and right branches of the tree
- Keep repeating splitting process until stopping criteria is met:
  - When a node is 100% one class
  - When splitting a node will result in the tree exceeding a maximum depth
  - Information gain from additional splits is less than threshold
  - When number of examples in a node is below a threshold

屏幕剪辑的捕获时间: 2023/5/29 9:29

决策树中的递归算法：由一个个小树组成大树



屏幕剪辑的捕获时间: 2023/5/29 9:35

One Hot encoding:

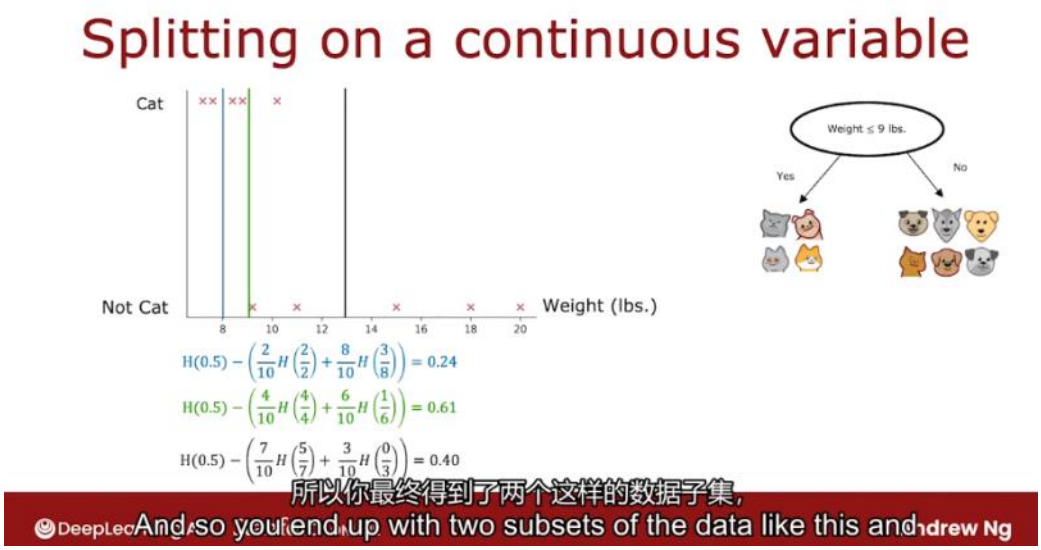
如果一个特征可以取k个值（k≥3），那么可以创建k个二元特征，这些特征只能取0和1值.

也可以让所有二元特征取0和1值，可用于神经网络

可以取连续值的特征

直觉是找一个阈值，将连续数据分为两类

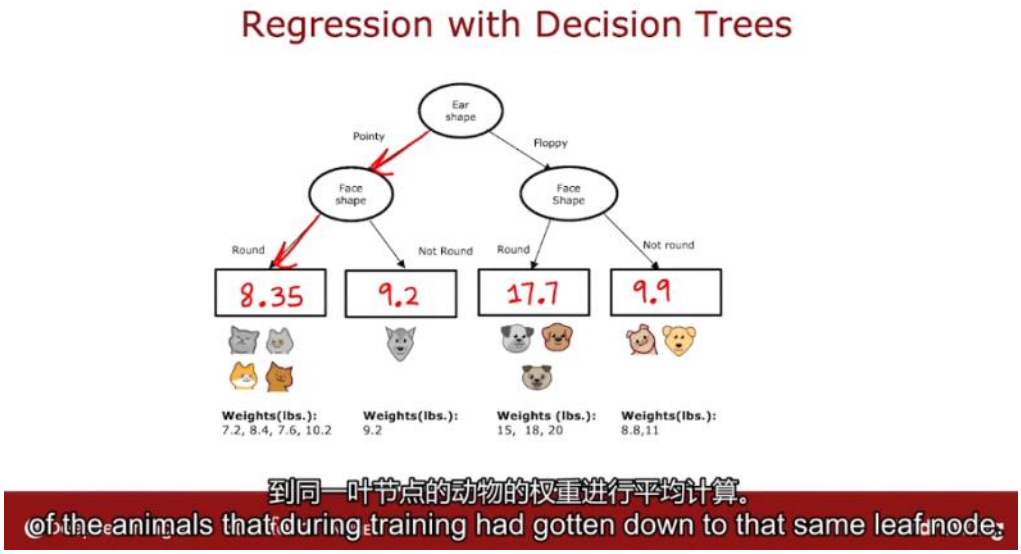
方法是把每两个相邻数据中间作为阈值，计算信息增益，找出这些增益中的最大值，作为划分连续数据集的阈值



屏幕剪辑的捕获时间: 2023/5/29 15:09

※回归树

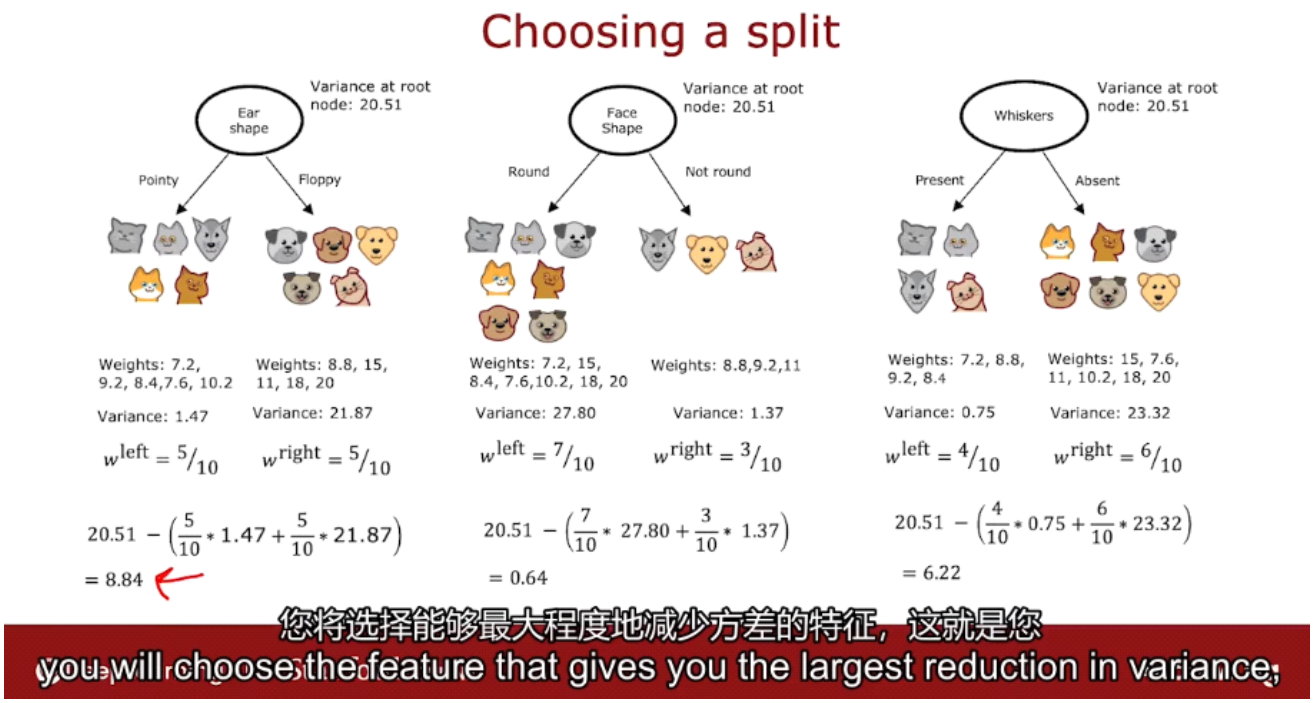
任务是根据特征预测一个连续值



屏幕剪辑的捕获时间: 2023/5/29 15:15

构建回归树时，决定如何划分节点的原则不再是信息增益最大原则，而是要使分类后的数据方差最小（Variance）

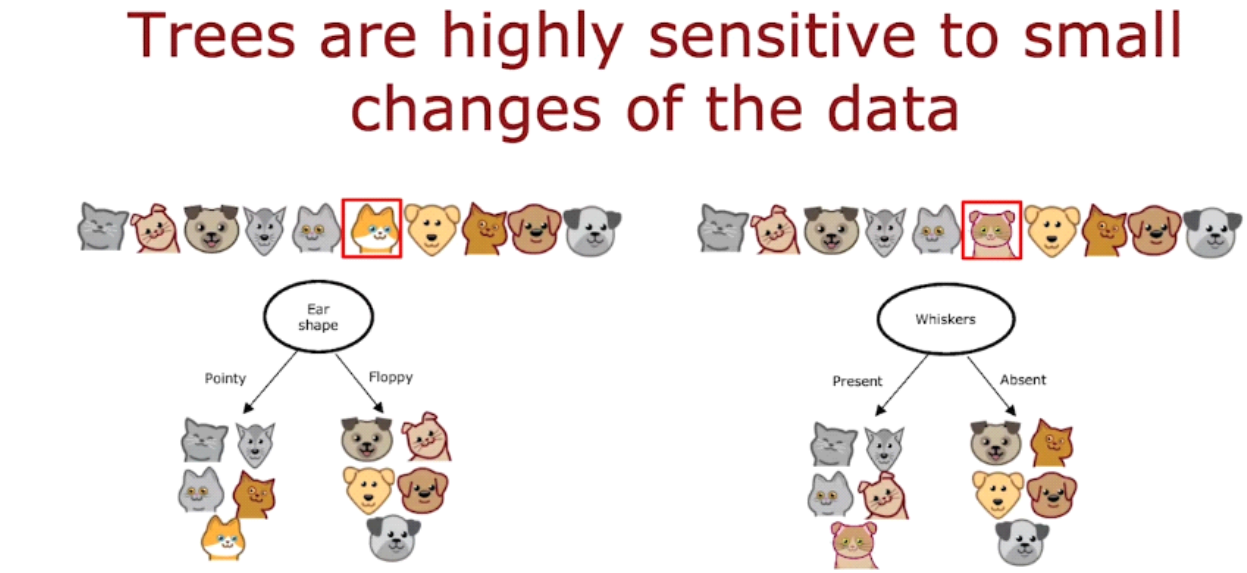
评价节点划分性能的公式仍然是由根部数据的方差减去左右两支方差的加权平均（方差减小最多的划分方法）

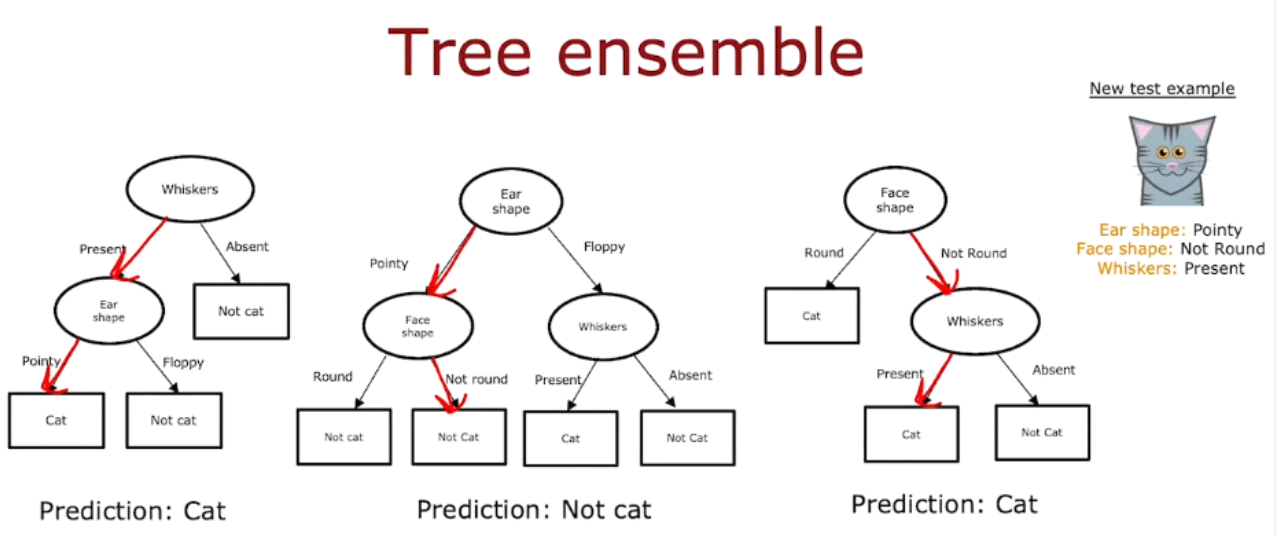


屏幕剪辑的捕获时间: 2023/5/29 15:46

使用多个决策树（树集合）

首先，决策树对于数据的微小变化十分敏感



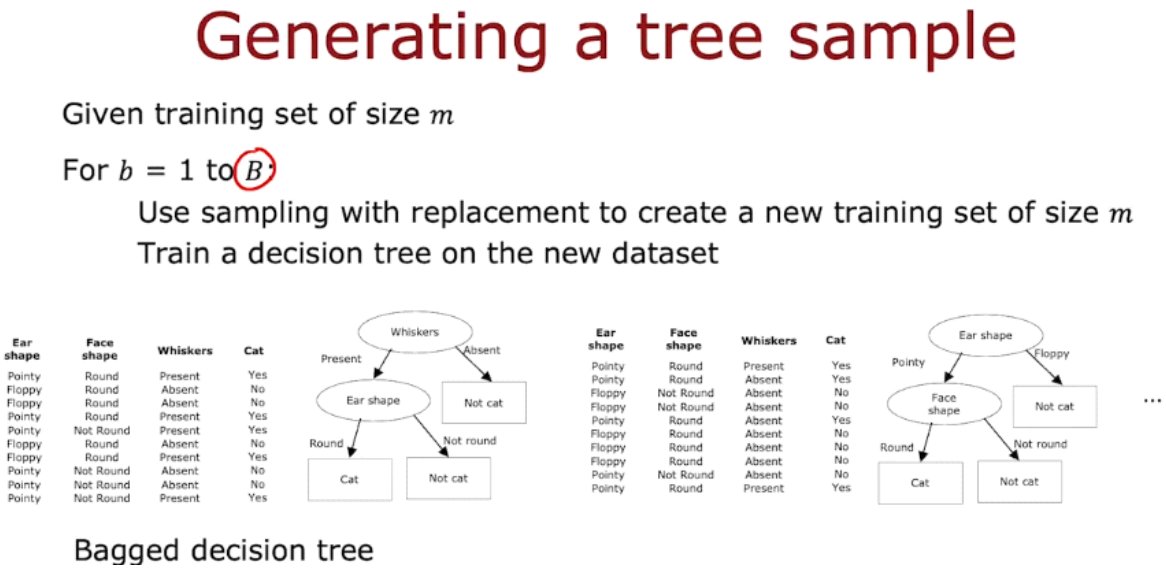


上面三个树，两个预测为🐱，一个预测不是🐱，从而认为这个例子是🐱

随机森林算法

1.生成树集合：对于一个大小为m的训练集，对其进行B次有放回的取样，从而训练出B个决策树

注意，B变大不会对算法的表现有不利影响，但是B过大会导致计算过程过长，收益下降很快

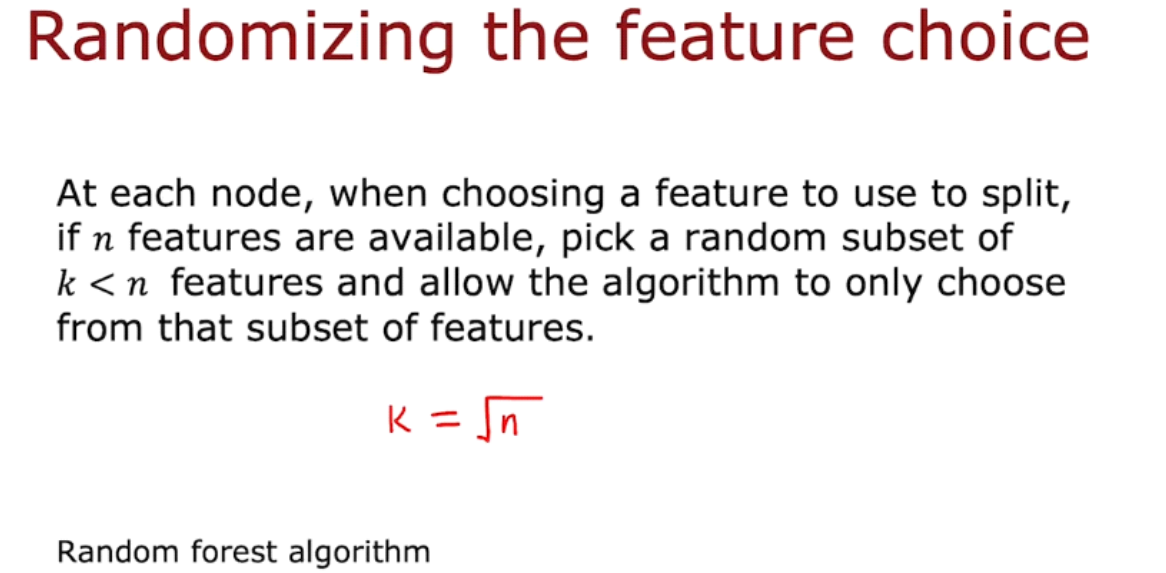


2.随机森林算法

有时，对于所有B个训练集，节点处选择的划分方法全都一样，所以需要对上面的算法进行改进

因此在每个分割节点处，随机选择特征，如果该处一共有n个特征，那么只在其中随机选出k个特征（k < n），从这k个特征中选出最合适的划分特征生成决策树。一般会取$k = \sqrt{n}$.

这样做会比单个决策树的鲁棒性更好，因为这样是在鼓励算法探索数据的小变化，并且对不同的决策树做平均，这也意味着数据集的微小改变不会让算法的整体表现变差。



当前应用最广的决策树集合算法：XGBoost

1.Boosted Trees：在先前生成树集合时，在B次中，每次有放回取样都是等概率取样，但现在不再这样取样，而是在第二次及以后取样时，倾向于在数据集中取出前面的决策树容易分类错误的数据，来生成下一个决策树，就像是针对性的学习

Boosted trees intuition

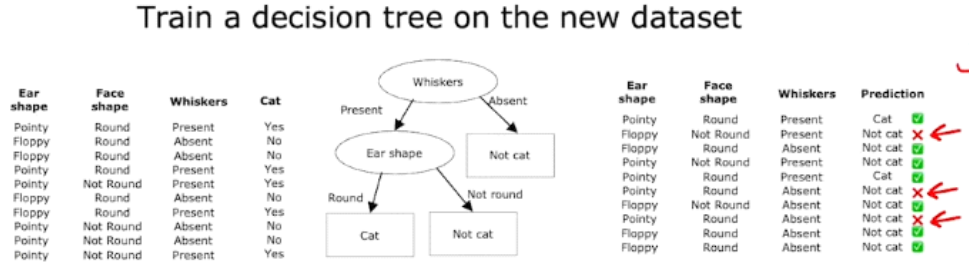
Given training set of size m

For $b = 1$ to B:

Use sampling with replacement to create a new training set of size m

But instead of picking from all examples with equal (1/m) probability, make it more likely to pick examples that the previously trained trees misclassify

Train a decision tree on the new dataset



先前样本树的集合仍然表现不佳。

the ensemble of the previously sample trees is still not yet doing well on g

2.XGBoost：（原理很复杂）

XGBoost (eXtreme Gradient Boosting)

- Open source implementation of boosted trees
- Fast efficient implementation
- Good choice of default splitting criteria and criteria for when to stop splitting
- Built in regularization to prevent overfitting
- Highly competitive algorithm for machine learning competitions (eg: Kaggle competitions)

何时用神经网络，何时用决策树?

Decision Trees vs Neural Networks

Decision Trees and Tree ensembles

- Works well on tabular (structured) data
- Not recommended for unstructured data (images, audio, text)
- Fast
- Small decision trees may be human interpretable

Neural Networks

- Works well on all types of data, including tabular (structured) and unstructured data
- May be slower than a decision tree
- Works with transfer learning
- When building a system of multiple models working together, it might be easier to string together multiple neural networks

而对于决策树，您一次只能训练一个决策树。
Whereas for decision trees you can only train

DeepLearning.AI Andrew Ng