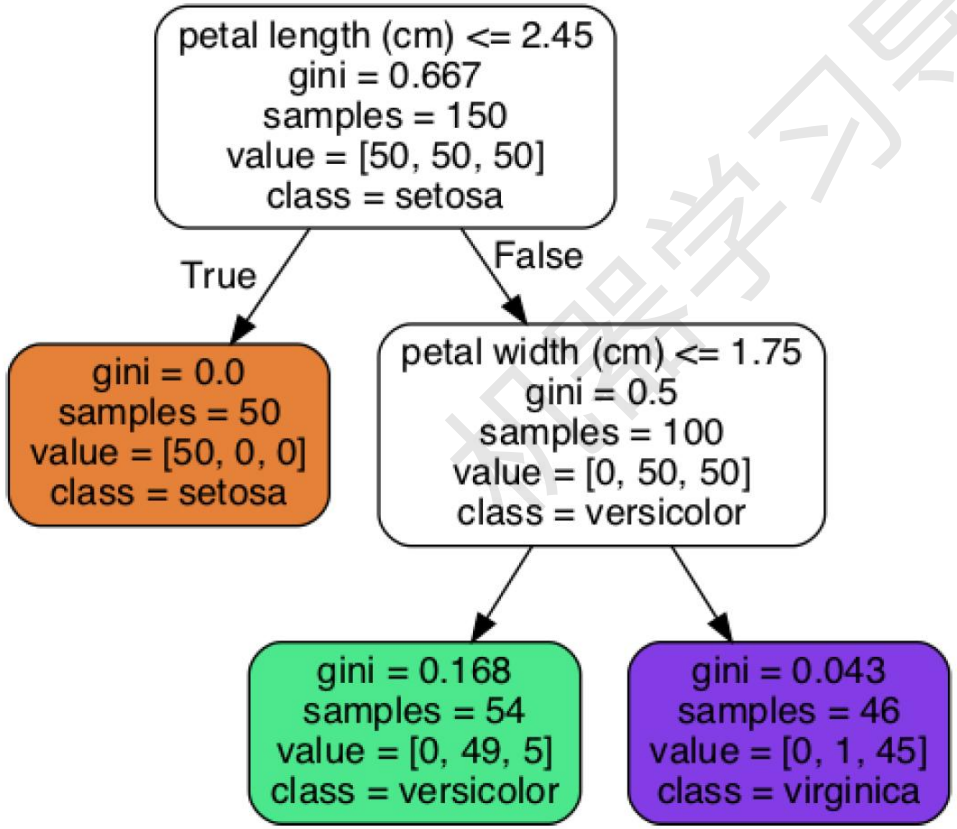
# 机器学习导论
# 第六章

王小航

# 决策树

- 用petal length和petal width两个属性，在鸢尾花数据集上训练一个决策树模型（Scikit-Learn使用的是CART算法）



# 做出预测

- 假设你找到一朵鸢尾花，要对其进行分类。
- 你从**根节点**开始（深度为0，在顶部）：该节点询问花的花瓣长度是否小于2.45cm。如果是，则向下移动到根的左子节点（深度1，左）。
- 在这种情况下，它是一片**叶子节点**（即它没有任何子节点），因此它不会提出任何问题：只需查看该节点的预测类，然后决策树就可以预测花朵是山鸢尾花（class=setosa）。

# 做出预测

- 现在假设你发现了另一朵花，这次花瓣的长度大于2.45cm
- 你必须向下移动到根的右子节点（深度1，右），该子节点不是叶子节点，因此该节点会问另一个问题：花瓣宽度是否小于1.75cm？
- 如果是，则你的花朵很可能是变色鸢尾花（深度2，左）。如果不是，则可能是维吉尼亚鸢尾花（深度2，右）

# 做出预测

- 节点的samples属性统计它应用的训练实例数量。
- 例如，有100个训练实例的花瓣长度大于2.45cm（深度1，右），其中54个花瓣宽度小于1.75cm（深度2，左）。
- 节点的value属性说明了该节点上每个类别的训练实例数量。例如，右下节点应用在0个山鸢尾、1个变色鸢尾和45个维吉尼亚鸢尾实例上。

# 特征选择

| ID | 年龄 | 有工作 | 有自己的房子 | 信贷情况 | 类别 |
|---|---|---|---|---|---|
| 1 | 青年 | 否 | 否 | 一般 | 否 |
| 2 | 青年 | 否 | 否 | 好 | 否 |
| 3 | 青年 | 是 | 否 | 好 | 是 |
| 4 | 青年 | 是 | 是 | 一般 | 是 |
| 5 | 青年 | 否 | 否 | 一般 | 否 |
| 6 | 中年 | 否 | 否 | 一般 | 否 |
| 7 | 中年 | 否 | 否 | 好 | 否 |
| 8 | 中年 | 是 | 是 | 好 | 是 |
| 9 | 中年 | 否 | 是 | 非常好 | 是 |
| 10 | 中年 | 否 | 是 | 非常好 | 是 |
| 11 | 老年 | 否 | 是 | 非常好 | 是 |
| 12 | 老年 | 否 | 是 | 好 | 是 |
| 13 | 老年 | 是 | 否 | 好 | 是 |
| 14 | 老年 | 是 | 否 | 非常好 | 是 |
| 15 | 老年 | 否 | 否 | 一般 | 否 |

# 特征选择

- 上表是一个由15 个样本组成的贷款申请训练数据，数据包括贷款申请人的4 个特征（属性）
- 第1 个特征是年龄，有3 个可能值：青年，中年，老年
- 第2 个特征是有工作，有2 个可能值：是，否
- 第3 个特征是有自己的房子，有2 个可能值：是，否
- 第4 个特征是信贷情况，有3 个可能值：非常好，好，一般
- 表的最后一列是类别，是否同意贷款，取2 个值：是，否

# 熵

- 在信息论与概率统计中，熵（entropy）是表示随机变量不确定性的度量。设X 是一个取有限个值的离散随机变量，其概率分布为

$$p(X = x_i) = p_i, \quad i = 1,2, \ldots, n$$

- 则随机变量X 的熵定义为

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$

若$p_i = 0$，则定义$0\log 0 = 0$

- 熵只依赖于X 的分布，而与X 的取值无关，所以也可将X的熵记作$H(\mathrm{p})$

# 信息增益

- 特征 A对训练数据集D的信息增益g(D, A)，定义为集合D的经验熵H(D)与特征 A给定条件下D的经验条件熵H(D|A)之差，即

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

- 对训练数据集（或子集）D，计算其每个特征的信息增益，并比较它们的大小，选择信息增益最大的特征。

# 例子

- 对前表所给的训练数据集D，根据信息增益准则选择最优特征
- 首先计算经验熵H(D)

$$H(D) = -\frac{9}{15}\log_2\frac{9}{15} - \frac{6}{15}\log_2\frac{6}{15} = 0.971$$

- 然后计算各特征对数据集D的信息增益、分别以$A_1, A_2, A_3, A_4$表示年龄、有工作、有自己的房子和信贷情况4个特征

# 例子

$$g(D,A_1)=H(D)-\left[\frac{5}{15}H(D_1)+\frac{5}{15}H(D_2)+\frac{5}{15}H(D_3)\right]$$

$$=0.971-\left[\frac{5}{15}\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)\right.$$

$$\left.+\frac{5}{15}\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)+\frac{5}{15}\left(-\frac{4}{5}\log_2\frac{4}{5}-\frac{1}{5}\log_2\frac{1}{5}\right)\right]$$

$$=0.971-0.888=0.083$$

这里$D_1, D_2, D_3$ 分别是D 中$A_1$（年龄）取值为青年、中年和老年的样本子集

# 例子

$$g(D,A_2)=H(D)-\left[\frac{5}{15}H(D_1)+\frac{10}{15}H(D_2)\right]$$

$$=0.971-\left[\frac{5}{15}\times 0+\frac{10}{15}\left(-\frac{4}{10}\log_2\frac{4}{10}-\frac{6}{10}\log_2\frac{6}{10}\right)\right]=0.324$$

$$g(D,A_3)=0.971-\left[\frac{6}{15}\times 0+\frac{9}{15}\left(-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}\right)\right]$$

$$=0.971-0.551=0.420$$

$$g(D,A_4)=0.971-0.608=0.363$$

# 例子

- 比较各特征的信息增益值，由于特征$A_3$（有自己的房子）的信息增益值最大，所以选择特征$A_3$作为最优特征

# 决策树的生成

- ID3 算法
- 从根结点（root node）开始，对结点计算所有可能的特征的信息增益，选择信息增益最大的特征作为结点的特征，由该特征的不同取值建立子结点
- 再对子结点递归地调用以上方法，构建决策树
- 直到所有特征的信息增益均很小或没有特征可以选择为止，最后得到一个决策树

# 例子

- 由于特征$A_3$（有自己的房子）的信息增益值最大，所以选择特征$A_3$作为根结点的特征、它将训练数据集D 划分为两个子集$D_1$（$A_3$ 取值为“是”）和$D_2$（$A_3$ 取值为“否”），由于$D_1$只有同一类的样本点，所以它成为一个叶结点，结点的类标记为“是”

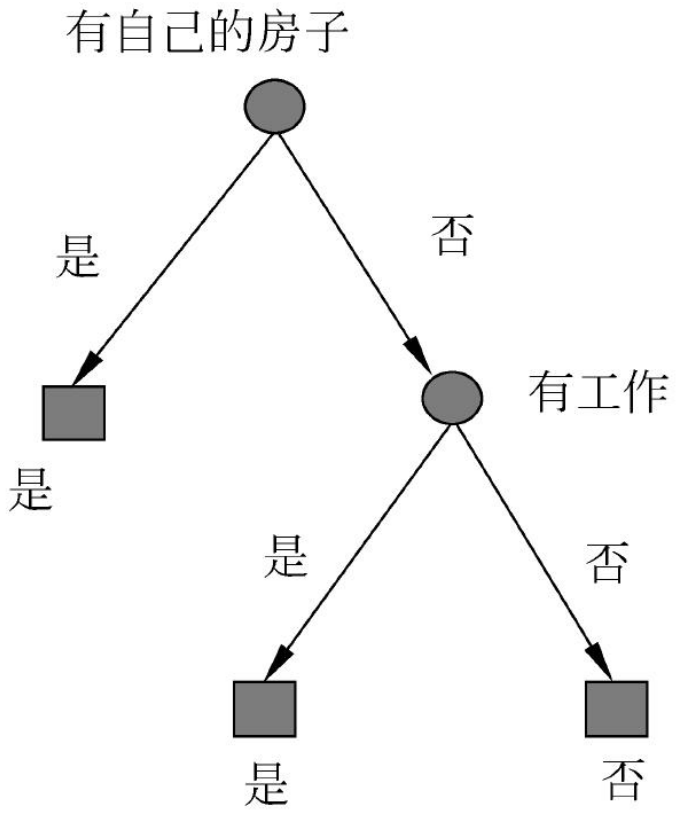
# 例子

- 对$D_2$则需从特征$A_1$（年龄），$A_2$（有工作）和$A_4$（信贷情况）中选择新的特征、计算各个特征的信息增益：

$$g(D_2, A_1) = H(D_2) - H(D_2 \mid A_1) = 0.918 - 0.667 = 0.251$$
$$g(D_2, A_2) = H(D_2) - H(D_2 \mid A_2) = 0.918$$
$$g(D_2, A_4) = H(D_2) - H(D_2 \mid A_4) = 0.474$$

- 选择信息增益最大的特征$A_2$（有工作）作为结点的特征

# 例子




- 由于$A_2$有两个可能取值，从这一结点引出两个子结点：一个对应“是”（有工作）的子结点，包含3个样本，它们属于同一类，所以这是一个叶结点，类标记为“是”
- 另一个是对应“否”（无工作）的子结点，包含6 个样本，它们也属于同一类，所以这也是一个叶结点，类标记为“否”

# 决策树的生成

- CART 算法
- 分类与回归树（classification and regression tree，CART）模型由Breiman 等人在1984 年提出，是应用广泛的决策树学习方法
- CART 同样由特征选择、树的生成及剪枝组成，既可以用于分类也可以用于回归
- CART 假设决策树是二叉树，内部结点特征的取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支

# 基尼指数

- 分类问题中，假设有K 个类，样本点属于第k 类的概率为$p_k$，则概率分布的基尼指数定义为

$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

- 对于二类分类问题，若样本点属于第1 个类的概率是p，则概率分布的基尼指数为

$$Gini(p) = 2p(1-p)$$

# 例子

- 首先计算各特征的基尼指数，选择最优特征以及其最优切分点
- 仍采用之前的记号，分别以$A_1, A_2, A_3, A_4$表示年龄、有工作、有自己的房子和信贷情况4 个特征
- 并以1，2，3 表示年龄的值为青年、中年和老年，以1，2 表示有工作和有自己的房子的值为是和否，以1，2，3 表示信贷情况的值为非常好、好和一般

# 例子

- 求特征$A_1$的基尼指数：

$$\text{Gini}(D, A_1 = 1) = \frac{5}{15}\left(2 \times \frac{2}{5} \times \left(1 - \frac{2}{5}\right)\right) + \frac{10}{15}\left(2 \times \frac{7}{10} \times \left(1 - \frac{7}{10}\right)\right) = 0.44$$

$$\text{Gini}(D, A_1 = 2) = 0.48$$

$$\text{Gini}(D, A_1 = 3) = 0.44$$

- 由于Gini(D, A=1)和 1 Gini(D, A=3)相等，且最小，所以$A_1 = 1$和$A_1 = 3$都可以选作$A_1$的最优切分点

# 例子

- 求特征$A_2$和$A_3$的基尼指数：

$$\text{Gini}(D, A_2 = 1) = 0.32$$

$$\text{Gini}(D, A_3 = 1) = 0.27$$

- 由于$A_2$和$A_3$只有一个切分点，所以它们就是最优切分点

# 例子

- 求特征$A_4$的基尼指数：

$$\text{Gini}(D, A_4 = 1) = 0.36$$
$$\text{Gini}(D, A_4 = 2) = 0.47$$
$$\text{Gini}(D, A_4 = 3) = 0.32$$

- Gini(D, A =3)最小，所以$A_4 = 3$为$A_4$的最优切分点

# 例子

- 在$A_1, A_2, A_3, A_4$几个特征中， Gini(D, A=1)=0.27最小，所以选择特征$A_3$为最优特征，$A_3 = 1$为其最优切分点
- 于是根结点生成两个子结点，一个是叶结点．对另一个结点继续使用以上方法在$A_1, A_2, A_4$中选择最优特征及其最优切分点，结果是$A_2 = 1$
- 依此计算得知，所得结点都是叶结点
- 对于本问题，按照CART 算法所生成的决策树与按照ID3 算法所生成的决策树完全一致．

# Scikit-Learn实现

- 以下代码在鸢尾花数据集上训练了一个DecisionTreeClassifier

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # petal length and width
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(X, y)
```

# Scikit-Learn实现

- 要将决策树可视化，首先，使用export_graphviz（）方法输出一个图形定义文件，命名为iris_tree.dot：

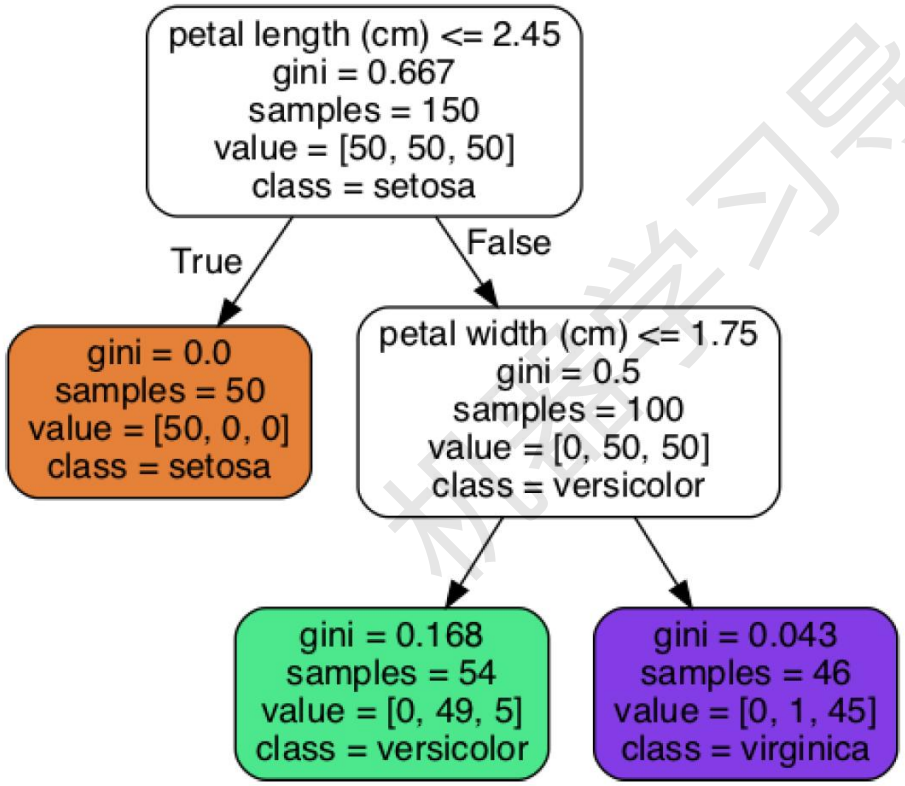
```
from sklearn.tree import export_graphviz

export_graphviz(
        tree_clf,
        out_file=image_path("iris_tree.dot"),
        feature_names=iris.feature_names[2:],
        class_names=iris.target_names,
        rounded=True,
        filled=True
    )
```

# Scikit-Learn实现

- 可以使用Graphviz软件包中的dot命令行工具将此.dot文件转换为多种格式，例如PDF或PNG。此命令行将.dot文件转换为.png图像文件：

```
$ dot -Tpng iris_tree.dot -o iris_tree.png
```

# Scikit-Learn实现



# Scikit-Learn实现

- 决策树同样可以估算某个实例属于特定类k的概率：首先，跟随决策树找到该实例的叶节点，然后返回该节点中类k的训练实例占比。
- 例如，假设你发现一朵花，其花瓣长5cm，宽1.5cm。相应的叶节点为深度2左侧节点，因此决策树输出如下概率：山鸢尾花，0%（0/54）；变色鸢尾花，90.7%（49/54）；维吉尼亚鸢尾花，9.3%（5/54）

```
>>> tree_clf.predict_proba([[5, 1.5]])
array([[0.        , 0.90740741, 0.09259259]])
>>> tree_clf.predict([[5, 1.5]])
array([1])
```

# 其他问题

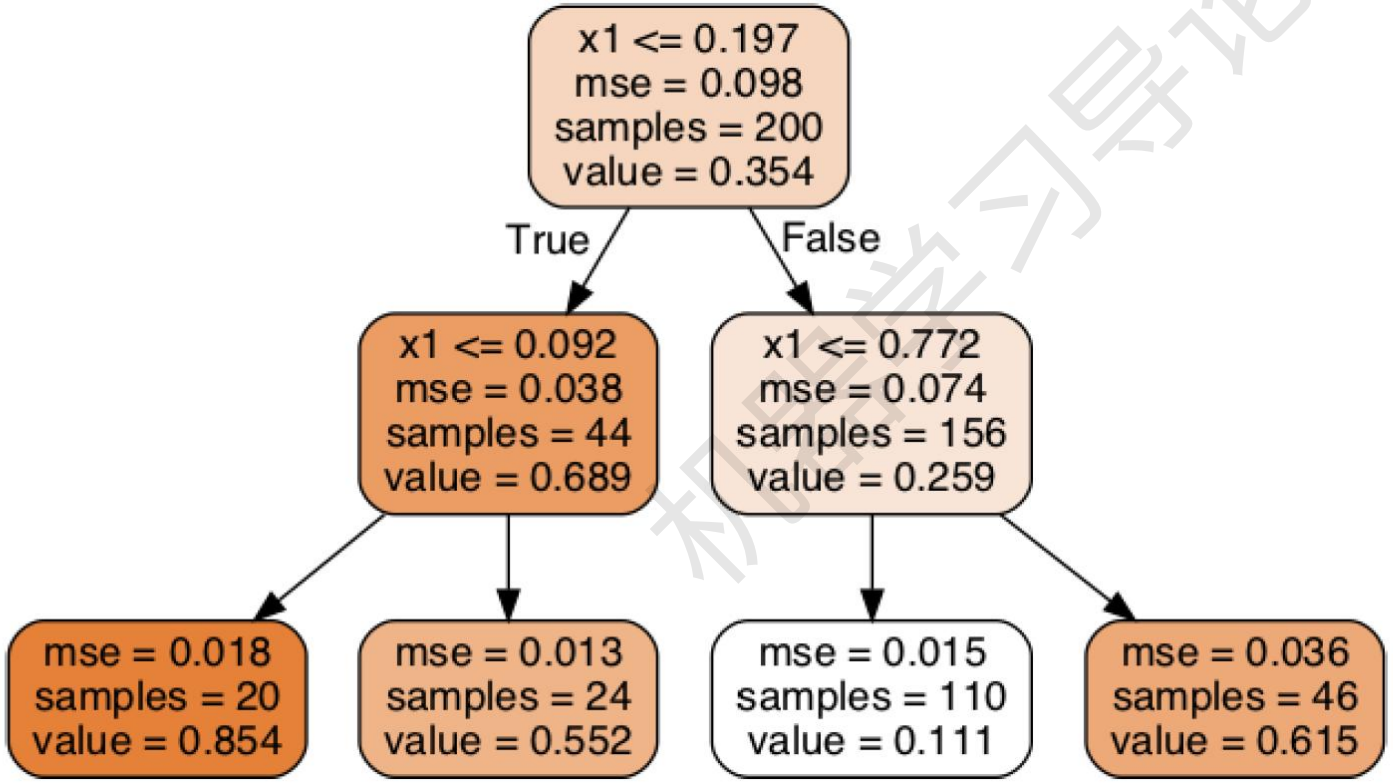
- 决策树的许多特质之一就是它们几乎不需要数据准备。实际上，它们根本不需要特征缩放或居中。
- 决策树极少对训练数据做出假设
- 避免过拟合至少可以限制决策树的最大深度：减小max_depth可使模型正则化，从而降低过拟合的风险
- 还可以先不加约束地训练模型，然后再对不必要的节点进行剪枝（删除）
- 决策树的主要问题是它们对训练数据中的小变化非常敏感，随机森林可以通过对许多树进行平均预测来限制这种不稳定性

# 回归

- 决策树还能够执行回归任务
- 使用Scikit-Learn的DecisionTreeRegressor类构建一个回归树，并用max_depth=2在一个有噪声的二次数据集上对其进行训练：

```
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(max_depth=2)
tree_reg.fit(X, y)
```

# 回归


x1 <= 0.197
mse = 0.098
samples = 200
value = 0.354
True
False
x1 <= 0.092
mse = 0.038
samples = 44
value = 0.689
x1 <= 0.772
mse = 0.074
samples = 156
value = 0.259
mse = 0.018
samples = 20
value = 0.854
mse = 0.013
samples = 24
value = 0.552
mse = 0.015
samples = 110
value = 0.111
mse = 0.036
samples = 46
value = 0.615

# 回归

- 这棵树看起来与之前建立的分类树很相似
- 主要差别在于，每个节点上不再预测一个类别而是预测一个值
- 例如，如果你想要对一个$x_1 = 0.6$的新实例进行预测，那么从根节点开始遍历，最后到达预测value=0.111的叶节点。这个预测结果其实就是与这个叶节点关联的110个实例的平均目标值