**决策树**

06

顾名思义，决策树模型为树状结构，可以实现分类或回归。决策树模型可视为一组“if … then …”规则，可解释性强。决策树是一种非线性模型，表示能力强。随着树结构的变化，决策树模型可以很简单，也可以很复杂，所以模型复杂度控制是决策树模型的重要问题。

本章我们讨论决策树的构建过程、包括树的构建和剪枝，并通过案例学习Scikit-Learn中决策树模型用于回归和分类的API。

## **标题2**  6.1 决策树基本原理

决策树模型由结点和有向边组成，其中结点表示特征空间子集，有向边代表一个划分规则，从根结点到叶子结点的有向边代表了一条决策路径。决策树的路径是互斥并且完备的，因此决策树将特征空间划分为互不相交的单元。图6-1给出了鸢尾花分类数据上一个3层决策树及其对应的2维空间上的空间划分。

|  |
| --- |
|  |
| （a）决策树 |
|  |
| （b）特征空间划分 |

图6-1 决策树示意

用决策树进行预测时，从根结点开始，对样本的相应特征进行测试，根据测试结果将样本分配到相应的子结点，所以决策树模型可以认为是if-then规则的集合。递归地对样本测试，直到样本被划分某个叶子结点。最后根据该叶子结点的分数对测试样本进行预测：

|  |  |
| --- | --- |
| ， | （6-1） |

其中表示第个叶子结点代表的特征空间，指示性函数，括号中的条件满足时取1，否则值为0。为第个叶子结点对应的参数，包括从根结点到第个叶子结点的路径上，每个结点选择的特征及划分阈值。是第个叶子结点的预测值。对回归任务，通常为第个叶子结点所有样本的的均值，这时预测的结果L2损失最小；对分类任务，通常为第个叶子结点所有样本的的分布，分类结果可以取分布中概率最大的类别。

例如，对图6-1（a）中的决策树，输入一个样本，其特征值为（4.90， 3.10， 1.50， 0.10），决策时从树的根结点出发，取该样本的第4维特征花瓣宽度（pedal width）值为0.10，由于，走左侧分支，到底叶子结点，对应特征区间划分。该叶子结点对应的鸢尾花的类别为山鸢尾（setosa），所以模型将该样本的类别判定为山鸢尾。

同其他机器学习模型一样，决策树模型的目标函数与包含两部分：训练集上的损失函数之和以及正则项。决策树模型的损失函数与结点的不均净度有关，不同的决策树算法中不纯净度度量稍有不同，下面我们会详细讨论。正则项可取L1正则（叶子结点的数目）或/和L2正则（叶子结点的分数）。

## **标题2**  6.2 建树

决策树的训练就是决策树建树过程，对应特征空间的划分。选择最优决策树的问题是个NP完全问题，所以我们一般采用启发式方法近似求解。决策树的学习算法通常递归地选择最优特征及划分阈值，并根据该特征对训练数据进行划分，使得划分后各个数据子集越纯净越好。

**算法**6-1**：决策树生成算法**

1. 构建根结点：将所有训练数据放在根结点，并将该结点加入叶子结点列表；

2. 若叶子结点列表为空，算法结束；

否则从叶子结点列表中挑选1个叶子结点，

（1）若该叶子结点的样本集合已足够纯净，计算该叶子结点对应的预测分数，并将其从叶子结点列表中删除（最终叶子结点，不再划分）；

（2）否则：

对每个特征的每个可能的划分方式，尝试将该结点的样本集合进行划分，计算划分后的纯净度；

从第步所有的划分中，选择一个最优划分（划分后不纯净度下降最多），将训练数据划分成若干子集，每个子集为当前结点的子结点，并将这些子结点加入叶子结点列表。

历史上出现过3种比较流行的决策树模型：ID3、C4.5、分类回归树（Classification And Regression Tree，CART）。其中ID3和C4.5建树建树过程几乎一样，只是算法中计算不纯净度的方式不同，ID3采用的准则是信息增益，而C4.5是信息增益比。ID3和C4.5可以是多叉树。CART是二叉树，既可以做分类，也可以做回归，特征选择准则是Gini指数（回归任务中为L2损失）。

### **6.2.1** **ID3**和**C4.5**

ID3和C4.5选择特征的准则和信息熵有关。

熵表示不确定程度。对于随机变量，熵定义为

|  |  |
| --- | --- |
| 。 | （6-2） |

数据集的经验熵定义为：

|  |  |
| --- | --- |
|  | （6-3） |

其中为数据集中的样本数目，样本的类别可取值为，为第个类别数据子集中的样本数目。式（6-3）相当于我们用每个类别样本出现的频率去估计概率，所以是熵的估计，刻画了数据集中样本的类别分布的散布程度，亦可视为数据集的不纯净程度度量。熵越大，表示该数据集中不同类别样本分布越均衡，数据集越不纯净。

对于某个特征，定义其条件熵为：

|  |  |
| --- | --- |
|  | （6-4） |

其中为特征可取值的数目（此处只能取离散值，特征可取值为）。条件熵表示知道后，还有多不确定。可以证明条件熵和联合熵之间的关系为：。

定义数据集关于特征的经验条件熵为：

|  |  |
| --- | --- |
|  | （6-5） |

其中为数据集中的样本数目，特征可取值为，为数据集中特征的值为的样本数目，为为数据集中特征的值为且类别为的样本数目。为条件熵的估计，刻画了数据集中属性中的那些样本的类别的分布情况。是熵的估计，刻画将数据集按照特征取值分为个子集后，各子集中样本的类别分布的平均散布程度，亦可视为将数据集分成多个子集后的平均不纯净程度。

#### 1．信息增益

特征对训练数据集的信息增益定义为：集合的经验熵与关于特征经验条件熵之差，即：

|  |  |
| --- | --- |
|  | （6-6） |

综上所述，

• 经验熵刻画了数据集的不纯净度（不确定性）。

• 经验条件熵刻画了在给定特征的条件下，将样本根据特征分成个子集，各子集的平均不纯净度。

• 信息增益刻画了根据特征的值，将数据集的划分为个子集后，各子集的平均不确定性减少量。

ID3决策树算法每次选择信息增益最大的特征对数据集进行划分。信息增益越大，表示根据这个特征取值对数据集进行划分后，每个子集的数据集越纯净，从而对应的分类器的分类能力更强。

#### 2．信息增益比

以信息增益作为划分训练集的特征选取方案，会偏向于选取值较多的特征。因为特征取值较多，会将数据集划分为更多个子集，从而每个子集相对较小，数据偏向于更纯净。极端情况下，特征在每个样本上的取值都不同（如每个样本的编号），此时根据特征对数据集进行划分，每个样本会被划分到不同的子集。即，此时条件熵取最小值：。这意味着信息增益达到了最大，然而很显然这个特征不是最佳选择，因为它并不具有分类能力。

C4.5通过信息增益比来解决该问题。特征对训练集的信息增益比定义为信息增益与关于特征的熵之比：

|  |  |
| --- | --- |
|  | （6-7） |

其中，表征了特征对训练集的拆分能力，其中为集合中的样本数，表示集合中特征取值为的样本数。因为只考虑样本在特征上的取值，而不考虑样本的标签，所以这种拆分并不是对样本的分类。

信息增益比本质上是对信息增益乘以一个加权系数，希望增加信息不要以分割太细为代价：

• 当特征的取值集合较大时，加权系数较小，表示抑制该特征。

• 当特征的取值集合较小时，加权系数较大，表示鼓励该特征。

**例6-1：采**用ID3和C4.5，根据用户属性（日志密度、好友密度、是否使用真实头像），判断该用户账号的SNS帐号是否真实。训练数据集如表6-1所示。

表**6-1** 账号真实性判断案例

|  |  |  |  |
| --- | --- | --- | --- |
| 日志密度 | 好友密度 | 是否使用真实头像 | 账号是否真实 |
| s | s | no | no |
| s | l | yes | yes |
| l | m | yes | yes |
| m | m | yes | yes |
| l | m | yes | yes |
| m | l | no | yes |
| m | s | no | no |
| l | m | no | yes |
| m | s | no | yes |
| s | s | yes | no |

在=10个样本中，账户真实的账号有个，不真实的账号有个，所以：

日志密度有s，m，l共3种取值，每种取值的样本数为：。特征取值为s的3个样本中，真实帐号的样本数为，不真实帐号的样本数为；特征取值为m的4个样本中，真实帐号的样本数为，不真实账号的样本数，；特征取值为l的3个样本中，全部为账号真实的样本，。所以：

同理，得到：

，

。

所以ID3选择信息增益最大的特征。

为了计算信息增益比，先计算熵：

则信息增益比为：

C4.5选择信息增益率最大的特征，也是。

根据特征的3个取值，上述10个样本被分为3个子集，对应的树结构如6-2所示：

• 特征取值为的2个样本，对应的标签均为，无需再细分，叶子结点；

• 特征取值为的4个样本，对应的标签均为，无需再细分，叶子结点；

• 特征取值为的4个样本，对应的标签两种取值都有，生成中间结点，再细分。

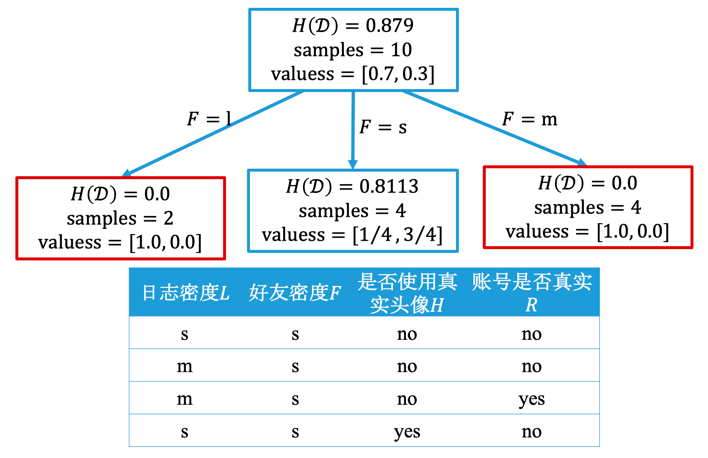


图6-2 SNS帐号真实性判断的ID3/C4.5决策树。

### **6.2.2** **CART**

CART的树是二叉树，递归地二分特征，将输入空间划分为有限个单元。与ID3和C4.5根据某个特征的多个取值，将数据集分为多个子集不同，CART每次只能将数据集分为左、右两个集合：、，在每次增加结点时，不仅要选择最佳特征，还要选择该特征的最佳划分阈值。我们设特征值小于等于划分阈值为左侧分支，特征值大于划分阈值为右侧分支。选择特征及其划分阈值的准则也是分割后两个分支的样本越纯净越好。

令结点的样本集合为，对候选划分，选择特征，分裂阈值为，将样本分裂成左右两个分支和，定义数据集关于特征和划分阈值的不均净度度量为

|  |  |
| --- | --- |
|  | （6-8） |

其中为某种不纯净度度量。可以为：

• 回归：集合中样本响应与响应均值的残差平方和，也是集合内的样本方差：

|  |  |
| --- | --- |
|  | （6-9） |

集合内的样本方差越小，表示该集合样本的响应变化越小，即数据越纯净。越小，也表示我们对该特征区域用响应均值预测时，预测残差平方（L2损失）最小。

• 分类：基尼指数（Gini Index）：

|  |  |
| --- | --- |
|  | （6-10） |

基尼指数表示在样本集合中，随机选中一个样本，该样本被分错的概率。基尼指数越小，表示越不容易分错。相比于熵，基尼指数随分布的变化趋势同熵一致，但基尼指数计算无需运算，速度更快。当时，不同对应的熵和基尼指数的对比如图6-3所示。

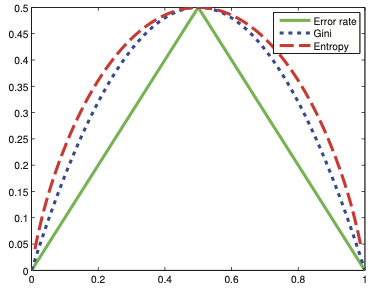


图6-3 两类分类的不纯度度量（错误率、熵和Gini指数）的比较，横坐标表示类别1的概率。

**例6-2：采**用CART，根据用户属性（日志密度、好友密度、是否使用真实头像），判断该用户账号的SNS帐号是否真实。训练数据集如表6-1所示。

1. 根据日志密度，候选划分方式有3种：

* 左侧分支，右侧分支：
* 左侧分支，右侧分支：
* 左侧分支，右侧分支：

（2）类似地，根据好友密度，候选划分方式有3种：

* 左侧分支，右侧分支：
* 左侧分支，右侧分支：
* 左侧分支，右侧分支：

（3）类似地，根据是否使用真实头像，候选划分方式有1种：

* 左侧分支，右侧分支：

在这些划分方式中，最小的为，对应的划分为根据日志密度，左侧分支为左侧分支，右侧分支，树结构如6-4所示。

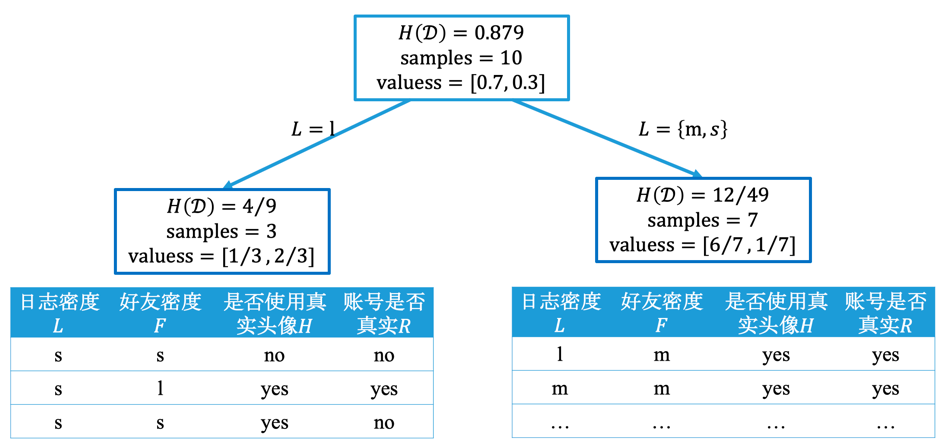


图6-4 SNS帐号真实性判断的CART。

在上述决策树的构建过程中，需要计算经验熵，原则上只支持离散型特征。对连续型特征，通常采用二分法将其离散化。假设特征在中出现了个不同的取值，将这些值从小到大进行排列，记作，则共有个候选划分点依次为：。对大数据集，训练数据中特征的取值数目可能会很多，要考虑所有个候选点开销太大，此时可考虑将特征分成多个区间（等间隔划分或根据百分位数划分），采用直方图方式快速寻找近似的最佳划分点。第7章介绍的XGBoost和LightGBM都支持这种快速建树方式。

## **标题2**  6.3 剪枝

如6.2节中所述的决策树建树过程只考虑了决策树对训练样本的拟合程度。和训练数据集拟合得很好，未必意味着对未知的测试数据也有好的性能，即可能发生过拟合的现象。一种解决方案是对生成的决策树进行剪枝，简化树结构，使其具有更好的泛化能力。剪枝是去掉过于细分的叶子结点，使得该叶子结点中的子集回退到其父结点并让其父结点成为叶子结点。

设树的叶子结点个数为，叶子结点的索引为，定义树的分数为：

|  |  |
| --- | --- |
| ， | （6-11） |

其中表示集合中包含的训练样本的数目，为正则参数。这样上述表达式的形式同机器学习模型的目标函数

的形式一致。注意这里对所有叶子结点求和等价于对所有样本求和（一个样本总会落入某个叶子结点），叶子结点的不纯净度表示模型预测值和真实值的损失，正则项为叶子结点的数目。

• 叶子结点数目越大，决策树越复杂。

• 叶子结点的不纯净度越大，表示叶子结点的样本类别分布越分散，损失函数越大。

• 叶子结点的不纯净度还需要加权，权重为叶结点大小，即叶结点越大，其预测错误的影响越大。

通常决策树划分得越细致，叶子结点内的样本越纯净，但此时叶子结点越多。所以最佳模型是这两方面的折中。

建树的过程是从根结点到叶子结点，剪枝则从树的叶子结点开始，递归地向上回退：设某个叶子结点回退到父结点之前与之后的整棵树分别为和 ，对应的分数分别为和。若，则进行剪枝，将父结点变成新的叶子结点。

## **标题2**  6.4 提前终止

6.3节中的剪枝亦被称为后剪枝，因为决策树构造完成后再进行剪枝。另一种控制决策树复杂度的方式是预剪枝，即在构造决策树的同时进行剪枝。在决策树构建时，当树达到某些条件时，停止树增长，亦被称为提前终止（Early Stopping）。

常用的提前终止条件包括：

• 达到最大树深度；

• 叶子结点数目达到最大值；

• 叶子结点的纯净度已达到一定精度；

• 结点中的样本数量少于某个阈值；

• 最优划分带来的增益（不纯净度的减少）小于某个阈值。

## **标题2**  6.5 案例分析——蘑菇分类

本节我们对蘑菇分类数据集采用决策树进行分类。数据中每个样本包含模型的22个属性，比如形状、气味、颜色等。我们数据集中分离出80%样本（6,513个）为训练样本，其余20%（1,611个）为测试样本。蘑菇的属性均为字符表示的离散值，我们通过Scikit-Learn的LabelEncoder类将其编码为数字（标签编码）。由于决策树中特征取值不参与算术运算，标签编码结果的不同数字表示不同特征取值，即使数字有序关系问题不大，所以不一定要对离散值特征进行独热编码。

Scikit-Learn中决策树分类器接口为，建树过程中穷举搜索所有特征的所有可能的分裂点。Scikit-Learn实现了CART算法的改进版本，没有后剪枝操作，采用预剪枝控制树的复杂度。模型的超参数如表6-2所示，这些超参数需可通过验证集上的性能进行调优。由于超参数较多，为避免多个参数一起调优产生组合爆炸问题，我们通常每次对1～2个参数一起调优。如有必要，亦可进行多轮超参数调优，一般不超过两轮。我们对最重要的两个超参数（和）进行了调优，其他参数的调优类似。越大模型越复杂，越小，模型越复杂。不同超参数对应的模型性能如图6-5（a）所示。在本案例中，树的最大深度对模型性能很大（最佳值为7），叶子结点包含的最小样本数对模型性能的影响较小。

表6-2 DecisionTreeClassifier的超参数

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 备注 |
| max\_depth | 树的最大深度。默认为None，表示在建树时不限制树的深度，直到每个叶子结点都是纯净的或叶子节点的样本数目小于min\_samples\_split。 | 数据少或者特征少的时候可以默认值。如果模型样本量多，特征也多的情况下，推荐限制这个最大深度。通常可取值10-100之间。 |
| max\_leaf\_nodes | 最大叶子结点数目。以最好优先（best-first）的方式生成树时，用该参数限制叶子节点数目。默认None:不限制叶子节点数目。 | 如果不为None，则忽略max\_depth。max\_depth和max\_leaf\_nodes约束选其中一个。 |
| min\_samples\_split | 对中间结点进行分裂的最小样本数。整数：样本绝对数目；浮点数：样本百分比，默认值为2。 | 如果样本量数量级非常大，则推荐增大该参数。如10万样本，minsamplessplit=10。 |
| min\_samples\_leaf | 叶子结点包含的最小样本数。整数：样本绝对数目浮点数：样本百分比默认值为缺1。 | 若某叶子结点数目小于该参数，则会和兄弟节点一起被剪枝。 |
| 续表 | | |
| 参数 | 说明 | 备注 |
| min\_weight\_fraction\_leaf | 叶子结点所有样本权重和的最小值。默认值为0，表示不考虑权重约束。 | 如果叶子节点样本权重和小于该参数，会和兄弟节点一起被剪枝。 |
| minim\_purity\_decrease | 结点分裂最小不纯净度。如果结点节点的不纯度（基尼系数，信息增益，均方差，绝对差）小于该阈值，则该节点不再分裂，为叶子节点。 |  |

决策树模型的一个好处是可以得到特征的重要性，并且可以自动实现特征选择。没有在树中出现过的特征不被模型选中，其重要性为0。决策树模型中的特征重要性表示该特征带来的度量指标（基尼指数或L2损失）的下降程度。蘑菇数据集上的特征重要性如图6-5（b）所示，可见只有13个特征对模型有贡献，其余9个特征的重要性为0。

如果必要，还可以根据特征重要性进一步进行特征选择。根据特征的重要性对特征继续排序，每次去掉最不重要的一部分特征，重新训练模型并在验证集上评估模型的性能，如果去掉部分特征后的模型性能更好，我们就可以去掉这些特征。在蘑菇数据集上，我们发现最终只需7维特征即可完美对训练数据和测试数据进行分类。

|  |  |
| --- | --- |
|  |  |
| （a）交叉验证性能 | （b）最佳模型的特征重要性 |

图6-5 Mushroom数据集上决策树模型的特征重要性。

在这个数据集上，决策树可以做到极致，AUC的值为1。得到决策树如图6-6所示。从图中，我们可以很清楚地看到模型做决策的过程，因此决策树模型的可解释性很好，这也是在很多领域决策树模型受欢迎的原因之一。

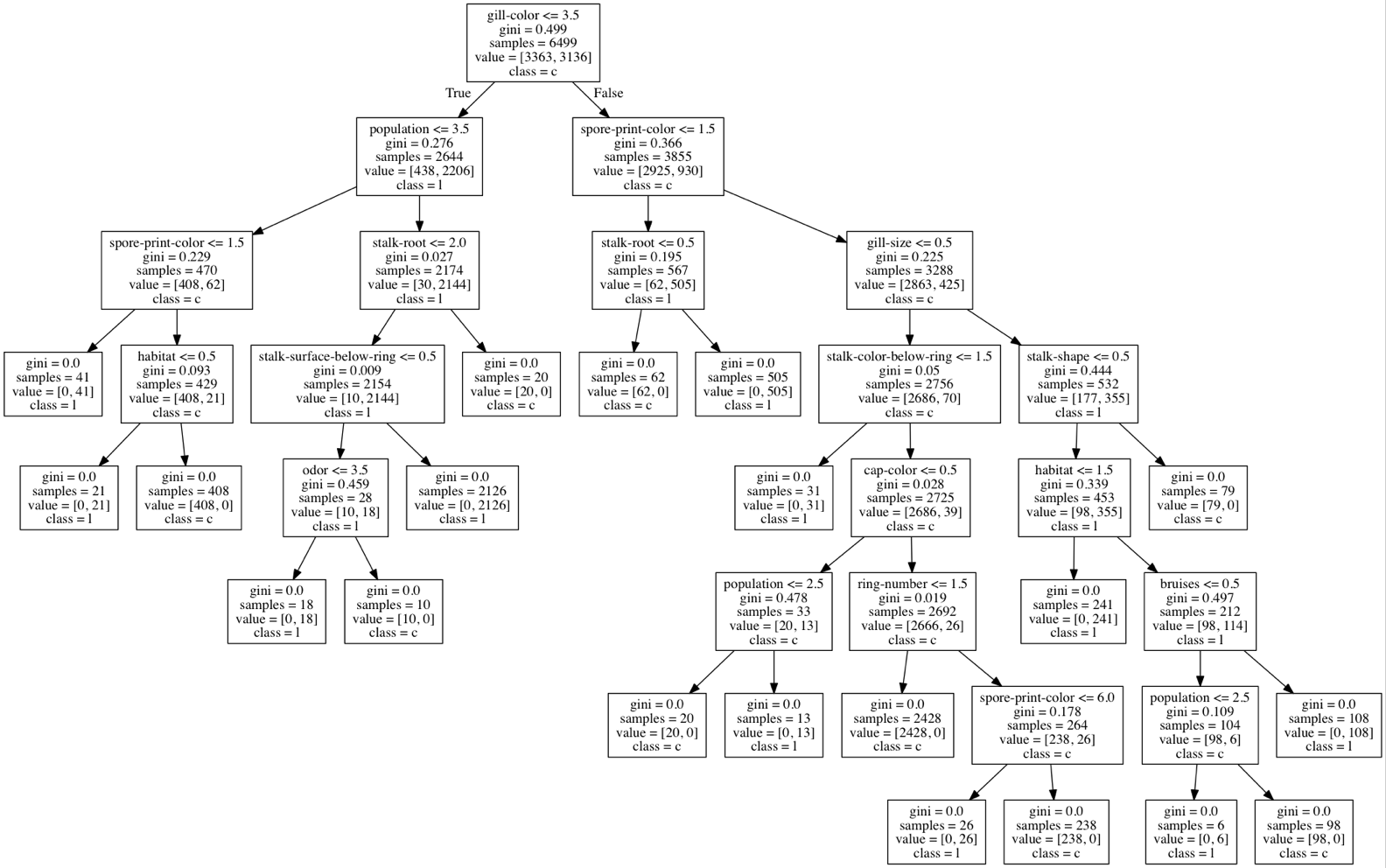


图6-6 Mushroom数据集上得到的决策树模型。

## **标题2**  6.6 案例分析——共享单车骑行量预测

这节我们在共享单车数据集上，采用决策树模型实现共享单车骑行量的预测，这是一个回归任务。数据说明和特征工程同2.7节相同。

不同超参数下决策树交叉验证估计的均方误差如图6-7所示。类似蘑菇分类任务上的表现，决策树模型的性能受参数影响较大，受参数影响较小。决策树在共享单车数据集上的效果并不好，交叉验证的误差估计和在测试集上的性能均远低于线性回归模型SVR。这可能是由于这个任务太复杂，单棵树并不能很好地完成分类任务，第7章我们将探讨采用随机森林和GBDT模型，融合多棵树的结果，性能有很大提升。

另外，对离散型特征，亦可采用标签编码（将不同的特征取值转换成不同的整数），而不是独热编码。共享单车数据集上的试验结果表明，这两种编码方式的性能相差不大。共享单车数据集的特征有很大冗余，决策树模型的特征重要性也表明了这一点。在该数据集的33维特征中，只有13维特征的重要性非0。

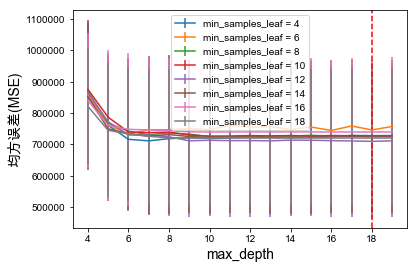


图6-7 不同超参数的决策树模在共享单车数据集上的性能。

## **标题2**  6.7 小结

决策树是一种常用的机器学习模型。它是非线性模型，学习能力强，但这也提醒我们要控制模型复杂度。可以通过对叶子结点包含的样本数目、树的深度、叶子结点的数目、以及每次划分带来的收益等因素来控制模型的复杂度。

因为决策树可简可繁，使得决策树模型是集成学习中很好的基学习器，如决策树和Bagging相结合，得到随机森林模型；决策树和基于梯度的提升算法结合，得到基于树的梯度提升（Gradient Boosting Decision Tree, GBDT）模型。

## **标题2**  6.8 练习

1. 关于决策树的超参数（树的最大深度），下面说法哪些正确？
2. 如果验证准确率相同，值越低越好；
3. 如果验证准确率相同，值越高越好；
4. max\_depth增加可能会导致过拟合；
5. max\_depth增加可能会导致欠拟合。
6. 对5.5节中练习1的数据集，分别采用ID3、C4.5和CART算法，建立深度为1的决策树模型（亦被称为树桩模型）。
7. 请用决策树模型对3.9节的第10题的数据进行建模，并比较模型与Logistic回归模型和SVM的性能。
8. 请用决策树模型对2.9节的第6题的数据进行建模，并比较这些模型与线性回归模型和SVR的性能。