教程6:CART决策树

提交日期：2020.8.1

提交人：詹紫琦

目录

[1. 题目 3](#_Toc26562)

[1.1决策树 3](#_Toc11601)

[1.2运行环境 3](#_Toc19300)

[2. 算法阐述或实验步骤说明 3](#_Toc18139)

[2.1变量说明 3](#_Toc11667)

[2.2CART决策树分类阐述及步骤说明 4](#_Toc8065)

[3. 实验结果与截图 5](#_Toc802)

[3.1CART决策树未剪枝分类的实验结果 5](#_Toc26865)

[3.2CART决策树使用全部数据集分类实验结果 6](#_Toc3746)

[3.3CART决策树进行后剪枝分类实验结果 8](#_Toc15524)

[4. 总结 13](#_Toc10256)

[4.1总结 13](#_Toc19817)

[5. 参考文献 13](#_Toc3769)

# 题目

## 1.1决策树

决策树[1]是一类常见的机器学习方法，以一个二分类任务为例，决策树能作为一个树形分类器通过筛选各类特征属性的值从而判断这个二分类任务的结果，即是与不是，而实现决策树算法的最终目标就是构建这一类树形判断结构。树形结构理所当然是由点集合与边集合以及之间的关系构成，所以构建决策树的步骤就在于构建结点，构建结点之间的边，而每一个结点之间都存在着一定的信息，这类信息就用于决策当中，使用这类信息进行决策，找到下一结点的位置，再进行递归决策，最终找到不存在后续结点的叶结点，该叶节点存储的便是类别信息，这样，通过一个树形结构，以依次访问根节点，孩子结点，结点之间的边，最终访问到包含类别信息的叶节点后成功实现决策树的分类任务。显然，决策过程的最终结论对应了我们所希望的判定结果。决策过程中提出的每一个判定问题都是对某一个属性的测试。通过不停的测试，最终得到分类结果。对判定问题定义的选择则是选择不同属性值的结点，通过信息增益或者是信息增益率，亦或者是基尼系数来进行判定结点的选择。

## 1.2运行环境

系统：Ubuntu20.04，python3.7，Anaconda集成工具Jupyter编写。GUI图形系统由Pycharm编写。

# 算法阐述或实验步骤说明

## 2.1变量说明

：代表数据集合D。

是CART决策树使用的基尼系数，代表数据集D的基尼系数，反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率，所以，当基尼系数越小的时候，则说明数据集D的纯度越高。

：代表着数据集中不同类别的符号。

是数据集D中的属性a的基尼系数

:是数据集中各类属性的集合。

：称之为最优划分属性。

：属性a的离散划分点。

## 2.2CART决策树分类阐述及步骤说明

首先选定初始数据集合，计算数据集合中不同属性的基尼系数，根据数据集合中不同属性的取值不同，就存在着不同的处理方法，在本次实验中使用的是连续型数据集，所以需要采取连续值处理。即将连续属性离散化，在这里采用的是二分法对连续属性进行处理。根据训练集中连续属性a而言，找出a的n个不同的取值，此时能得到n-1个离散划分点，找出针对该连续属性而言的离散划分点集合如公式2-1：

 （2-1）

该集合中包含了n-1个离散划分点，然后分别计算这n-1个离散划分点下的属性a在数据集合D下的基尼系数，选择能得到最小基尼系数的离散划分点。计算基尼系数的公式如2-2，2-3：

 （2-2）

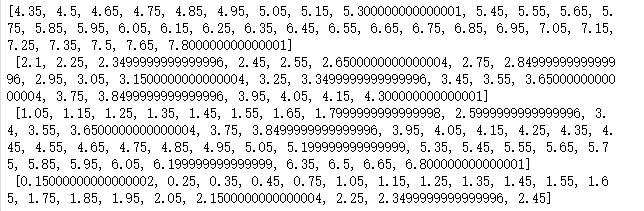
 （2-3）

找到能得到最小基尼系数的离散划分点后计算所有属性的基尼系数，确定最小基尼系数的属性a，将该属性a与相应的离散划分点作为决策树的根结点，这样就能将初始的集合根据最优划分属性的值能划分为两个集合，一个集合中属性的值全部都不小于属性的离散划分点，另一个集合中属性的值全部都小于属性的离散划分点的值。这样就能得到使树形结构的一个根结点生出左右两个孩子结点，然后依次对左右两个孩子结点做以上的操作，即把两个左右孩子结点当作新的根节点，继续递归进行，直到左孩子或者右孩子是一个叶结点，就停止该结点的递归，判断决策树中结点为叶结点的条件就是该结点中的数据集合的所有数据的类别都一致。这样就能得到一个存储着类别信息的叶节点，直到所有叶节点生成，并且并集与初始集合相等，则说明决策树构建完毕。构建完决策树后，由于训练出的模型可能会存在过拟合的情况，或者是模型的泛化能力不强，此时就需要进行剪枝操作，剪枝操作又分为预剪枝和后剪枝，预剪枝是在构建决策树的过程中进行剪枝，后剪枝则是在构建决策树之后再进行剪枝。剪枝操作实质上就是为了防止决策树模型过拟合，将某些叶结点与其双亲结点并为一个新的叶结点，既能加快决策树的构建速度同时也能提高决策树模型的泛化能力。

# 实验结果与截图

## 3.1CART决策树未剪枝分类的实验结果

本次实验中使用的数据集是鸢尾花数据集，总共有150条数据，总共有三个类别，其中每一个类别都包含着50个数据。取每一个类别的最后十条数据用作验证集。将标签值数据转化为类别0，1，2，分别代表着不同的三种鸢尾花种类。由于鸢尾花数据总共有四种属性，而且都为连续值属性，这里进行离散划分，能得到四个二元离散划分点集合。如图3-1：

图3-1四种属性的二元离散划分点集合

然后计算每一种属性的二元离散划分点后的基尼系数，得到最小的基尼系数，同时记录此时的二元离散划分点的大小。得到的每一种属性的最小基尼系数的划分点为：5.45，3.34999，2.59999，0.75，然后进行根结点的选择，找出的能带来最小基尼系数的属性为第三个属性，同时基尼系数的值为0.3333。然后不停的进行递归操作构建CART决策树。最后得到的各个结点信息如下如图3-2：（这里列举出前五个结点）

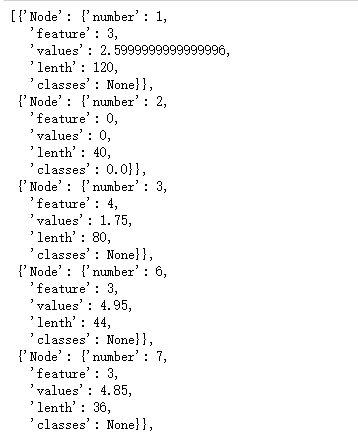


图3-2生成的CART决策树的前五结点信息

生成的决策树模型上通过30条验证集数据进行验证，得到的分类正确率为100%，这里我们可以发现由于数据项数太少，模型的泛化能力不强。所以需要后续的剪枝步骤。

## 3.2CART决策树使用全部数据集构建分类系统

通过上面的实验，虽然决策树模型能实现100%分类正确率的效果，但是由于训练数据较少，鸢尾花数据集只有150项，这里使用所有的150数据当作训练集，得到的决策树模型结点信息如图：3-3

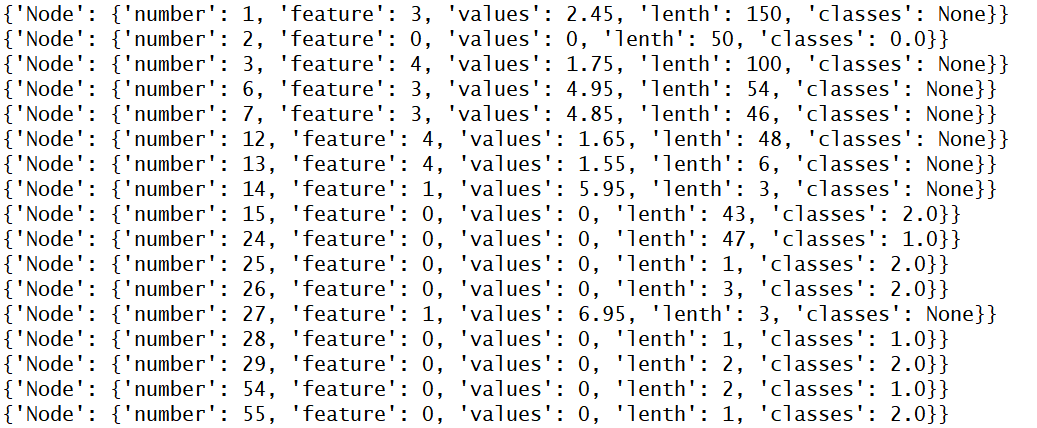


图3-3使用全部数据用作训练集后产生的所有决策树结点信息

将其设计为一个GUI系统，通过人为输入数据来进行验证，GUI页面如图3-4：

通过输入不同的测试用例，来取得预测结果，例如输入花萼长度为3，花萼宽度为3，花瓣长度为4，花瓣宽度为2，得到的预测结果如图3-5：

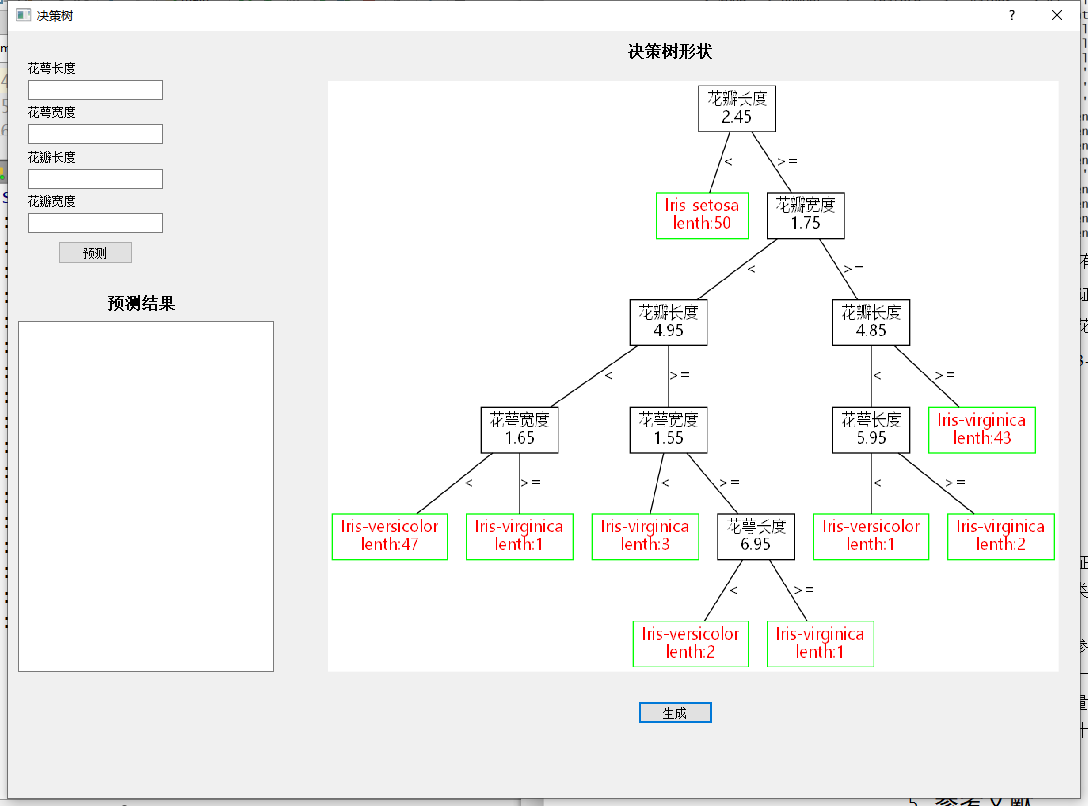


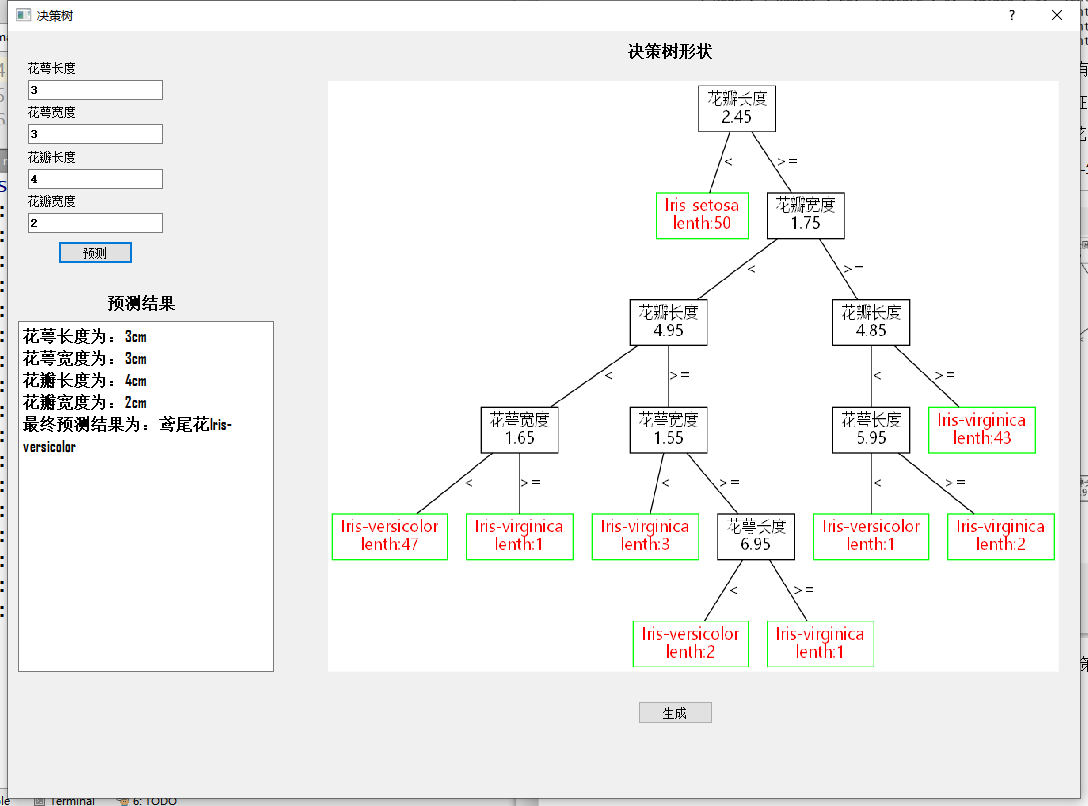
图3-4鸢尾花数据集训练出的决策树模型

图3-5输入测试用例预测后得到的结果

根据该测试用例，我们可以发现预测结果为鸢尾花种类为Iris-versicolor。由于模型的泛化能力较弱，需要进行后面的剪枝操作。

## 3.3CART决策树进行后剪枝分类实验结果

虽然前面分类正确率能达到100%，但由于训练集只有120项，验证集只有30项，可以看出训练出的决策树模型较为复杂，可能会产生过拟合的现象，这里我们采用后剪枝操作来对CART决策树进行剪枝。未剪枝生成的CART决策树如图3-6

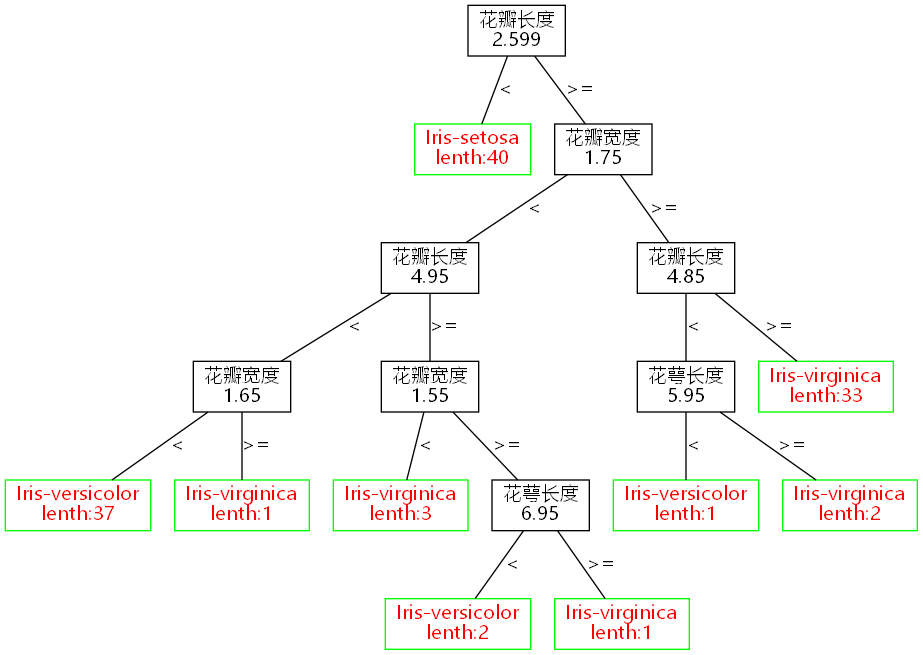


图3-6未剪枝的决策树

得到的验证集分类结果正确率为100%，经过不停的剪枝后，直到第七次剪枝，分类正确率为66.7%，前六次剪枝分类正确率全为100%，得到的剪枝步骤图如图3-7，3-8，3-9，3-10，3-11，3-12：最终得到的CART分类正确率在验证集上也能达到100%的正确率。

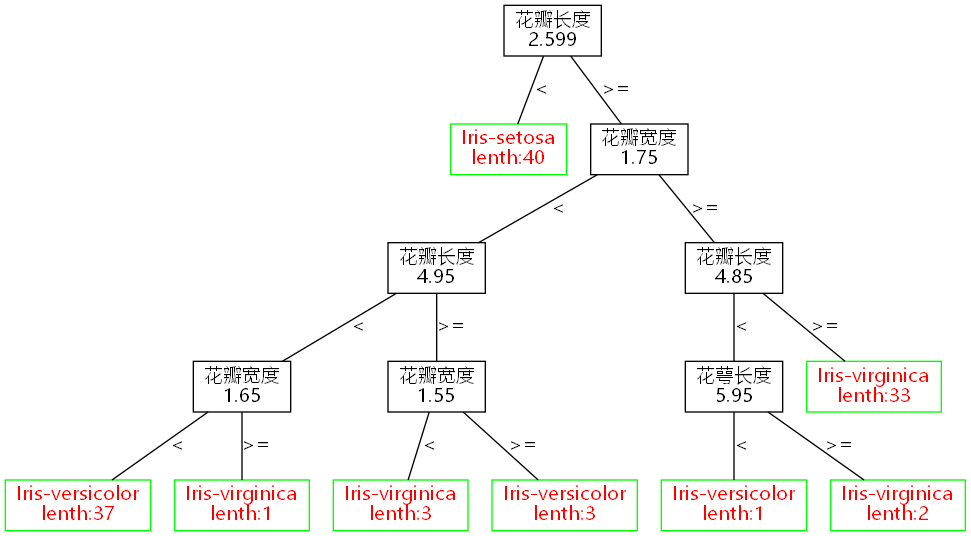


图3-7后剪枝一次后的决策树

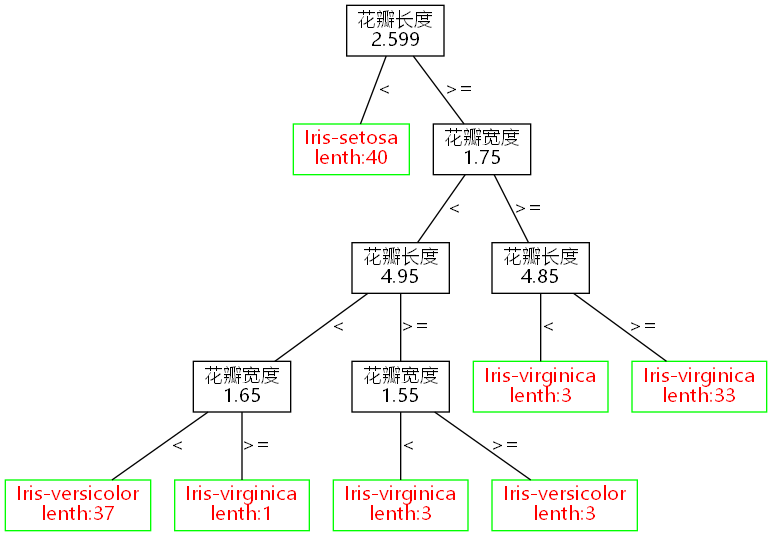


图3-8后剪枝两次的决策树

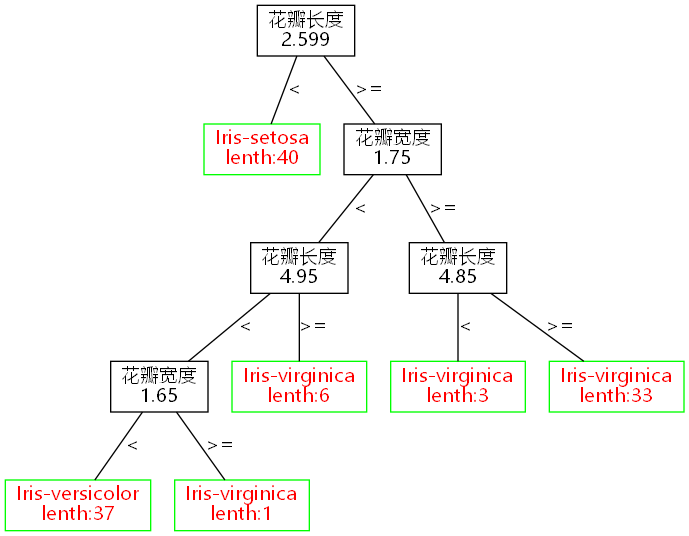


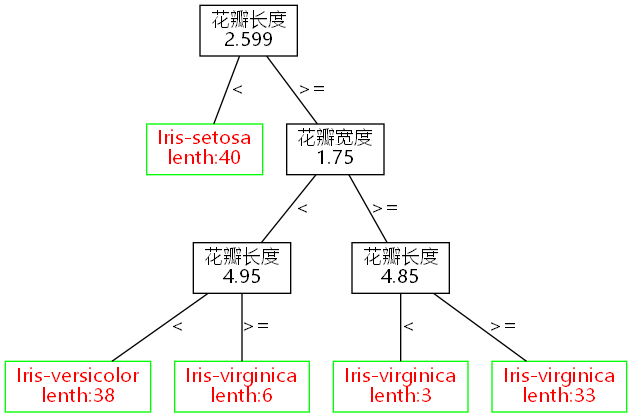
图3-9后剪枝三次的决策树

图3-10后剪枝四次的决策树

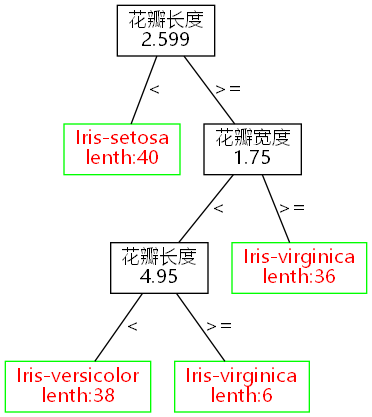


图3-11后剪枝五次的决策树

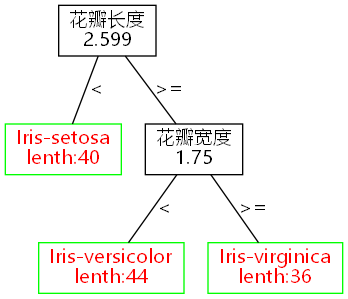


图3-12后剪枝六次，最终得到剪枝后的决策树

# 总结

## 4.1总结

CART决策树是决策树算法中的一种，使用的结点选择标准是基尼系数，通过选择能带来最小基尼系数的属性来选择结点，即选择最优特征，根据这种选择原则递归的来构建出一颗CART决策树，在实验中可以看出CART决策树分类的准确率很高，而且通过剪枝后能获得更为精简同时不失准确率的树模型。同时应用决策树进行分类任务时也较为容易理解，本文中涉及的只是单变量CART决策树，后续还可以实现多变量决策树，以及决策树回归等应用。

# 参考文献

[1]周志华等.机器学习（西瓜书）