

实验二RL报告

**江辉--171250506**



**2020-4-14**

**南京大学**

**软件学院**

目录

[一、 实验内容描述 1](#_Toc37721707)

[1、 项目目录描述 1](#_Toc37721708)

[二、 实验结果和树图形 1](#_Toc37721709)

[1、 生成的决策树 1](#_Toc37721710)

[2、 决策树图形 2](#_Toc37721711)

[三、 决策树算法原理阐述 2](#_Toc37721712)

[四、 决策树部分核心代码 3](#_Toc37721713)

[1、 ID3算法类 3](#_Toc37721714)

[2、 递归生成决策树流程代码 3](#_Toc37721715)

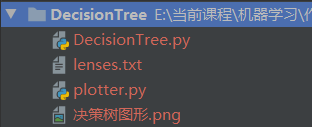
[3、 最佳特征选择函数 4](#_Toc37721716)

[4、 熵的计算 4](#_Toc37721717)

# 实验内容描述

本程序用python编写，手写ID3算法的实现，完成决策树实验。给定隐形眼镜小量数据集，构造决策树预测患者佩戴隐形眼镜类型，并通过 Matplotlib绘制树图形

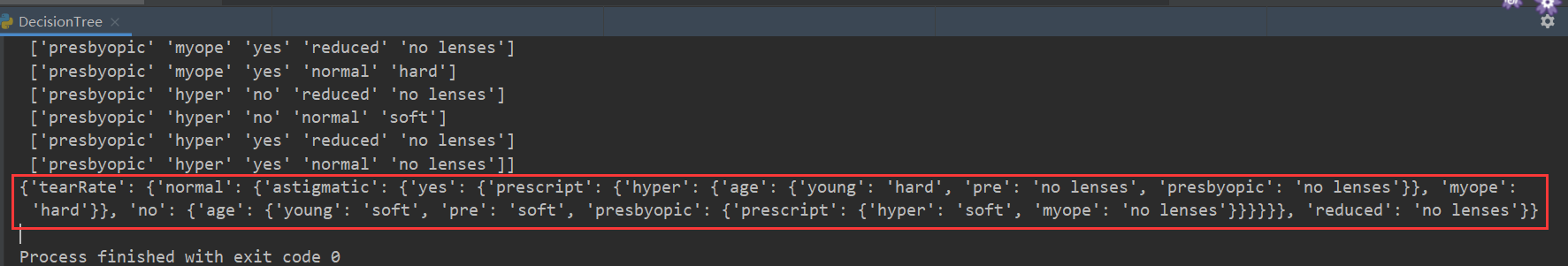
## 项目目录描述

本项目推荐使用pycharm打开

**DecisionTree.py**里写了main函数和决策树实现的代码，**plotter.py**是画树图形的代码

# 实验结果和树图形

## 生成的决策树

以下是控制台输出的结果，我也用文本编辑器转换为了用json格式查看的结果



## 决策树图形

’以下是用plotter.py画出的树图形

# 决策树算法原理阐述

决策树的目的是构建一棵树，从根节点出发，每一个非叶节点都是一个属性判断分支，直至叶节点确定分类，即在样本的多维属性中不断选择具有区分度的属性进行判断，逐步将样本精确至分类结果的算法。其中决策树的生成包含以下几个主要步骤

1. 特征属性选择：从目前有的特征中选一个作为当前节点的分支选项，每个选项会达到下一层的子树或叶节点

一般认为能越好地区分数据集的特征属性越应该优先进行判断，如ID3和C4.5以信息增益和信息增益率来选择能最大化地区分样本的特征，而CART中以类似信息熵的悉尼指数来简化计算，本质上都是为了选择最能提现类别不同的特征

1. 递归生成完整的决策树：对每个非叶节点递归进行步骤1中的操作，直至没有可再选择分支的特征
2. 可选的剪枝：决策树构建完后容易过拟合，所以需要限制分支的数量，有些算法会在生成决策子树的时候进行筛选，即预剪枝，有些会在在最后会对生成的决策树进行合并或裁剪，即后剪枝，目的都是增强泛化能力，本次实验中我采用ID3算法，没有做剪枝操作

# 决策树部分核心代码

## ID3算法类

下图是我实现的ID3算法类，\_\_init\_\_中可以看到就是存了数据集合属性集

calculate\_tree是递归生成决策树的函数

cal\_entropy是熵的计算函数

max\_count\_label是在没有可划分属性时选择占比最大的类别的函数

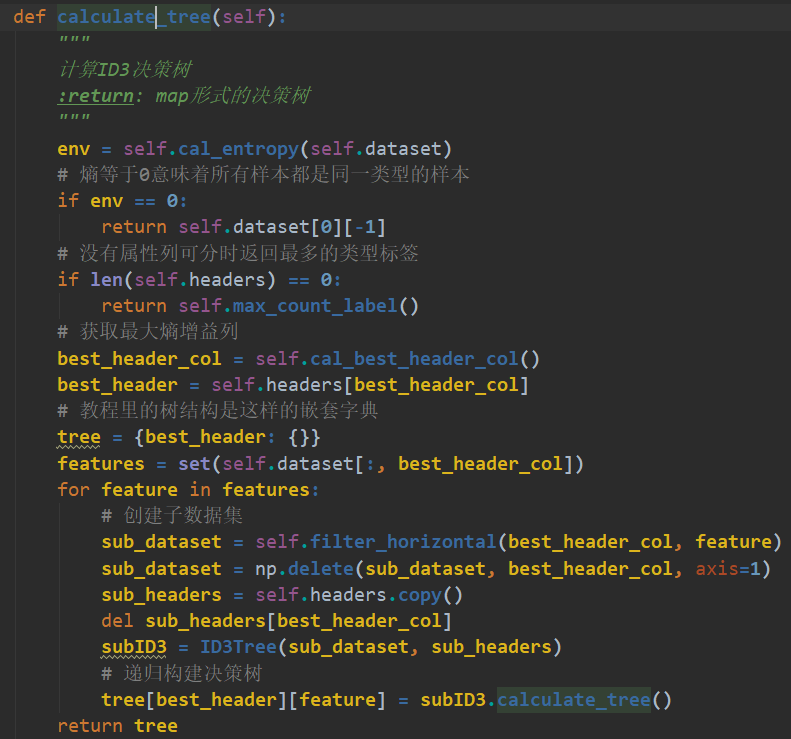
filter\_horizontal是选择特征分支时将数据集根据属性分割成不同的子数据集

delete\_col是选完一个特征后将其从数据集中移除

col\_best\_header\_col是选择最佳特征

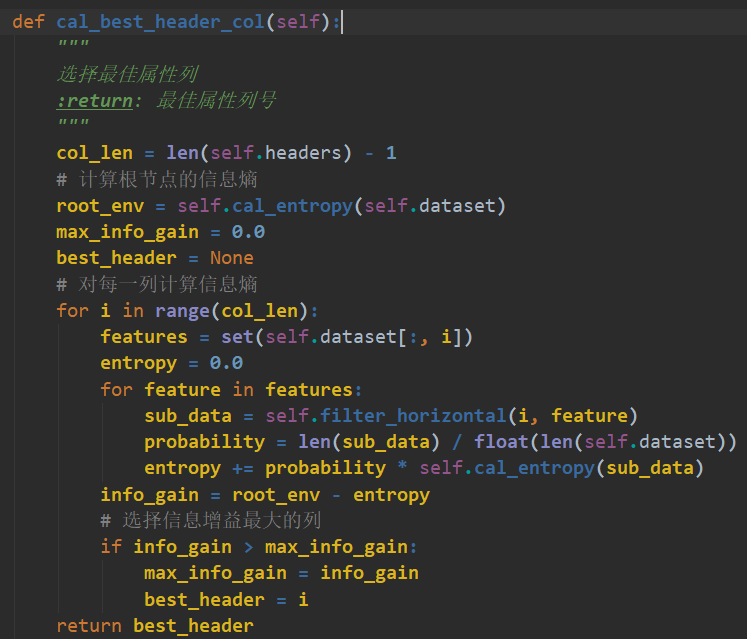
下面我对关键的函数进行简单描述

## 递归生成决策树流程代码

下面这个方法是递归主体，先判断是否是叶节点，是叶节点就返回标签，不是叶节点就选择最佳特征，然后对特征内每个属性构建子数据集并递归构造子树

## 最佳特征选择函数

先计算当前根节点的熵，再对每个特征计算条件熵，选出信息增益最大的特征返回特征列号



## 熵的计算

对每一类样本进行统计并根据熵的公式计算熵

