实验报告

1. 实验目的

本次实验的目的是以决策树为基学习器，实现集成算法中的AdaBoost算法和随机森林算法，对UCI数据集Adult进行分类。同时，以AUC为分类器的评价指标，通过5折交叉验证为两类算法选择最优的参数设置。

2. 实验过程

2.1 算法介绍

2.1.1 AdaBoost算法

AdaBoost算法基于加性模型，即基学习器的线性组合

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

来进行分类。

AdaBoost的伪代码如下：

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程：

1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4: $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5: **if** $\epsilon_t > 0.5$ **then break**
6: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
7: $\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
$= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: **end for**

输出： $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$

图1：AdaBoost算法

初始时，各样本和各基学习器的权值都相等。AdaBoost算法依次串行训练各基学习器：

- 基于样本权值 $\mathcal{D}_t$ 从数据集中训练出分类器 h_t
- 基于各基学习器权值 α_i 计算出 h_t 预测结果的误差 ϵ_t
- 依据 ϵ_t 计算出基学习器 h_t 的权值 α_t
- 按照以下公式更新样本权值

  $$\mathcal{D}_{t+1}(\mathbf{x}) = \frac{\mathcal{D}_t(\mathbf{x})}{Z_t} \exp\{-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))\}$$

  其中，Z_t 为规范化因子

  $$Z_t = \sum_{i=1}^{m} \mathcal{D}_t(\mathbf{x}) \exp\{-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))\}$$

训练完所有基学习器后，根据基学习器权值α_i将它们的预测结果线性结合，作为整个分类器的预测结果。

2.1.2 随机森林算法

随机森林算法并行训练各基学习器，采用自主采样法对原训练集采样，对每个基学习器提供不尽相同的训练集。最后，对每个基学习器应用相同的权值，线性结合它们的预测结果。

在本次实验中，随机森林算法的伪代码如下：

输入： 训练集 $D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\dots(\mathbf{x}_m,y_m)\}$

决策树算法 $\mathcal{L}$

采样算法 $\mathcal{S}$

基学习器个数 T

过程：

1：**for** $t=1,2,\dots,T$ **do**

2：$\mathcal{D}_{bs}=\mathcal{S}(D)$

3：$h_t=\mathcal{L}(\mathcal{D},\mathcal{D}_{bs})$

4：**end for**

输出： $H(\mathbf{x})=\frac{1}{T}\sum_{t=1}^{T}h_t(\mathbf{x})$

2.2 模块划分

在AdaBoost.py和RandomForestMain.py中都定义了以下函数：

```
def load_data()
def preprocessing(train_raw, test_raw)
def cross_validation(train_x, train_y, test_x, test_y, max_learner_num)
def sign(pred)
```

各函数的功能如下：

- **load_data：** 读取训练集和测试集数据
- **preprocessing：** 预处理数据集，分为以下三步：
  - 补充缺失值：使用`sklearn.impute`中的`SimpleImputer`类，采用`most_frequent`策略补充属性缺失值。
  - 对属性编码：使用`sklearn.preprocessing`中的`LabelEncoder`类，将非数字属性和标记编码为数字。其中，标记"<=50K"和">50K"被编码为0和1，继续处理将它们编码为-1和1。
  - 分割属性和标记
- **cross_validation：** 采用5折交叉验证，以AUC为评价标准，在一定范围内选择最佳的基学习器数目。
- **sign：** 处理分类器的预测结果，结果小于0则标记为-1，否则标记为1。

两种算法的main函数都依次调用`load_data`、`preprocessing`和`cross_validation`，处理数据集并进行交叉验证，选出最佳基学习器数目。最后，分别调用`adaboost`和`random_forest`函数对Adult数据集进行训练和测试，上述两个函数的伪代码参见2.1节。AdaBoost算法和随机森林算法都使用`sklearn.tree`中的`DecisionTreeClassifier`类作为基学习器。

2.3 参数设置

本次实验调用sklearn库，涉及的参数有

- sklearn.tree.DecisionTreeClassifier

  使用此类作为AdaBoost算法和随机森林算法中的基学习器。

  - **min_samples_split**：决策树内部结点分裂所需的最少样本数
  - **max_depth**：决策树最大深度
  - **random_state**：控制决策树训练的随机性，为了确保模型的稳定性，实验中令random_state为一固定整数
  - **max_features**：决策树结点分裂考虑的最大特征数目（仅随机森林算法使用）
- sklearn.model_selection.KFold
  - **n_splits**：交叉验证的折数
  - **shuffle**：分割训练集时是否对数据进行洗牌

在上述参数中，根据实验要求和课本中的算法的细节，可确定以下参数的值：

```
random_state=1
max_features="log2"
n_splits=5
shuffle=True
```

因此，对最终的强分类器准确率有影响的参数有

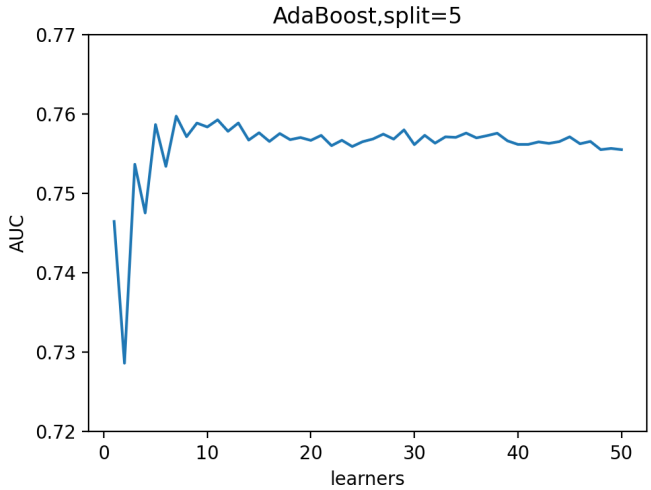
- min_samples_split
- max_depth
- 基学习器数量

max_depth参数对决策树深度进行限制，防止模型过拟合。考虑到Adult数据集的特征数较少，且从实际数据角度而言，不限制决策数深度得到的AUC和准确率指标更好，因此本次实验选择不设置max_depth。

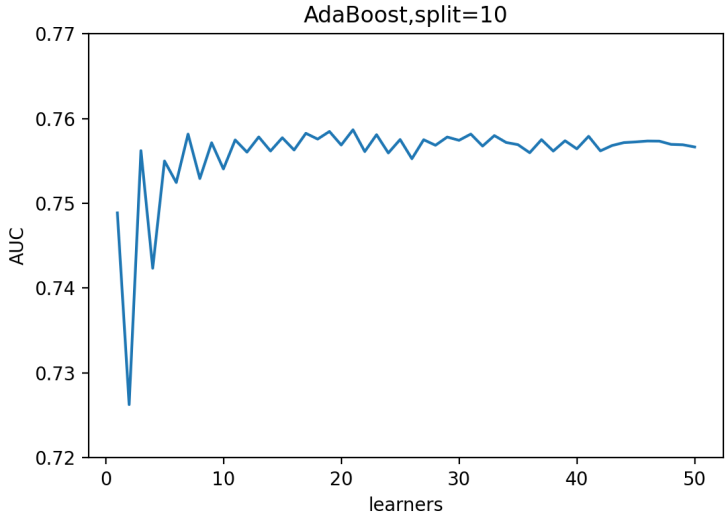
选定若干min_samples_split值，通过交叉验证计算AUC值，在1~50的范围内选择最优的基学习器数量：

- AdaBoost算法

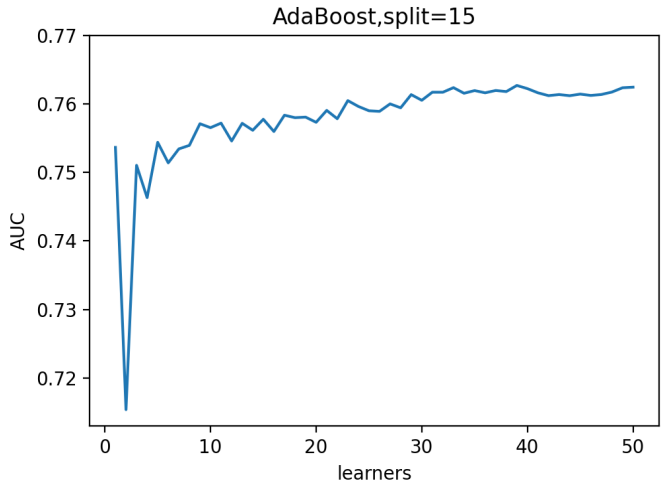
  min_samples_split=5：



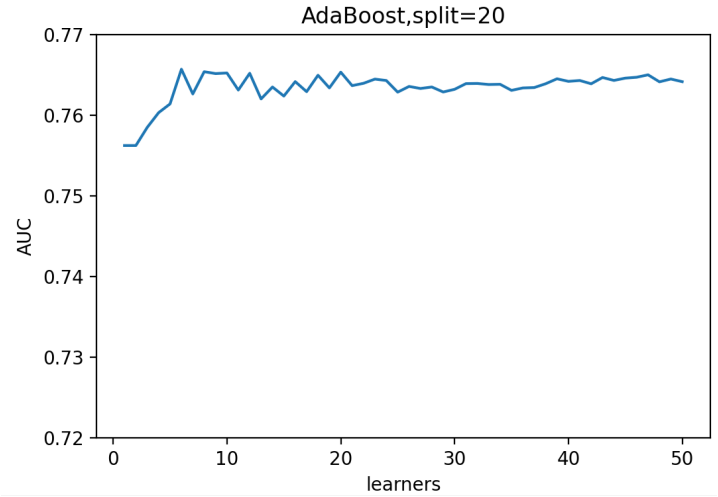
  min_samples_split=10：



min_samples_split=15：



min_samples_split=20：



对每个min_samples_split值，曲线趋于收敛后，最优的AUC指标和相应的基学习器数目如下表所示：

min_samples_split	基学习器数目	AUC
5	29	0.7580
10	21	0.7587
15	39	0.7627
20	20	0.7653

根据以上AUC曲线和数据，在AdaBoost算法中，基学习器数目在1~15之间时，AUC大致呈上升趋势，但波动较大。此后，AUC小幅度震荡，逐渐趋于收敛。

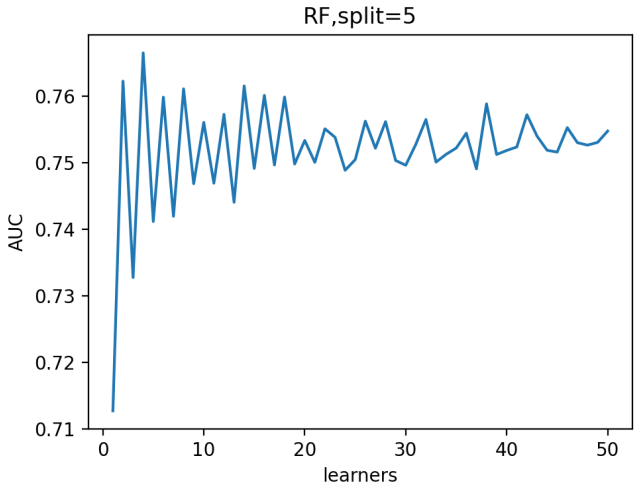
在与其他同学讨论以及查阅相关资料的过程中，了解到AUC曲线应当呈上升趋势，且基本不会出现下降的情况。考虑我得到的曲线出现震荡的原因，首先考虑交叉验证时设置的参数shuffle=True，设置这一参数会导致使用不同数目的基学习器训练时，划分数据集并不完全一致，因而可能出现小范围波动。但是，令shuffle=False后，AUC曲线依旧是在大致呈上升趋势的同时小幅震荡，并未完全消除震荡。

接下来，考虑数据预处理和AdaBoost算法设计对AUC曲线的影响。在数据预处理方面，用该属性最常出现的元素代替缺失值，这一做法会影响训练集的真实性，进而影响分类器的性能。在AdaBoost算法设计方面，基学习器权重计算方法的选取，以及样本权值规范化方式的选取，在基学习器增加时，会一定程度削弱已有基分类器的权重和性能。

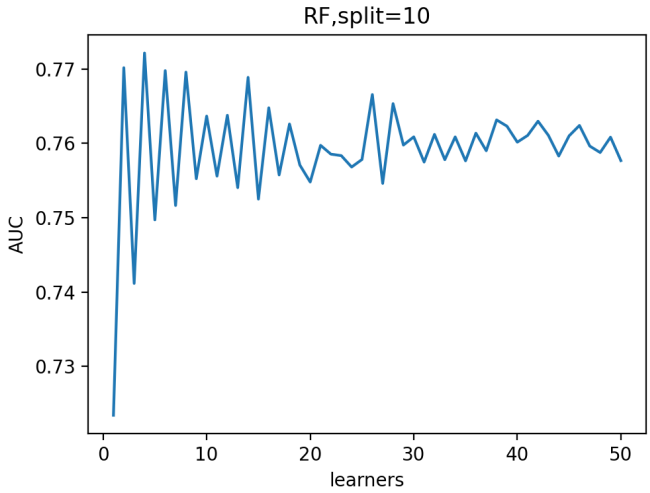
在基学习器数大致为29左右时会出现局部峰值，考虑到基学习器数继续增加后，AUC较基学习器数为29时提升不显著，且基学习器增多导致模型训练速度减缓，因此AdaBoost算法最终选用`29个基学习器`。此外，模型最优AUC值随着min_samples_split增加而增加，因此选择`min_samples_split=20`。

- 随机森林算法

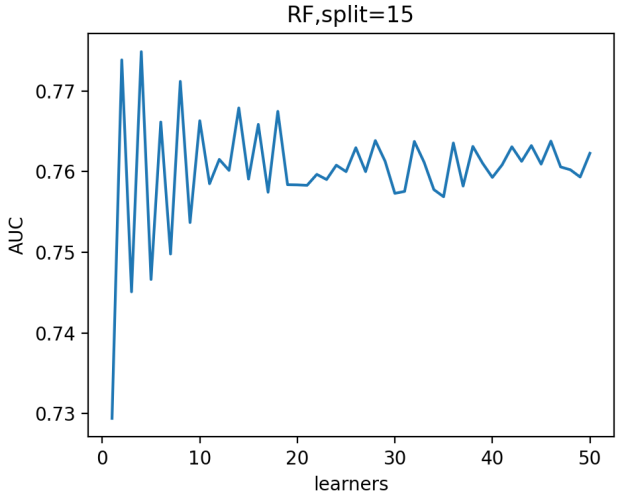
  min_samples_split=5：



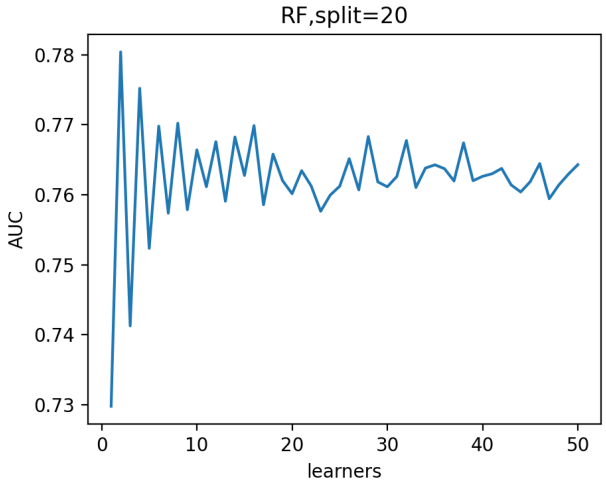
  min_samples_split=10：



min_samples_split=15：



min_samples_split=20：



对每个min_samples_split值，曲线趋于收敛后，最优的AUC指标和相应的基学习器数目如下表所示：

min_samples_split	基学习器数目	AUC
5	32	0.7658
10	26	0.7675
15	39	0.7686
20	28	0.7783

根据以上AUC曲线和数据，在随机森林算法中，基学习器数目在1~20之间时，AUC震荡幅度相对较大。基学习器数目大于20时，AUC值小幅度震荡，趋于收敛，震荡的原因已在上文分析过。当基学习器数为28左右时，会出现一个局部峰值。因此随机森林算法最终选用`28个基学习器`，并令`min_samples_split=20`。

3. 实验结果

3.1 AdaBoost算法

取基学习器数目为29，min_samples_split=20，在测试集上的实验结果为

AUC	准确率
0.756	0.841

3.2 随机森林算法

取基学习器数目为28，min_samples_split=20，在测试集上的实验结果为

AUC	准确率
0.776	0.859