Lab10　AdaBoost 应用实践

学习自助样本的生成方法和组合分类器、AdaBoost 算法。实验使用的弱分类器是同一类分类器，即是单层决策树。应用 AdaBoost 算法来预测患有马疝病的马匹是否能够存活。该数据集包含 30%的缺失值，相关缺失的数据已用 0 代替。代码文件 adaboost.py。

1. 单层决策树构建弱分类器

单层决策树对每个样本比较样本值和阈值可得到分类结果。在 AdaBoost 算法中，每轮迭代都使用最优的单层决策树，因此需要对于所有特征和特征值遍历，寻找最优分类器：

```
# 实现数据集和类标签导入：
>>> import adaboost
>>> datMat, classLabels = adaboost.loadSimpData()
# 构建多个函数建立单层决策树：
>>> D = mat(ones((5,1))/5)
>>>adaboost.buildStump(dataMat, classLabels, D)
```

2. 完整 AdaBoost 算法实现

```
#  adaBoostTrainDS()函数完成 AdaBoost 分类器的训练：
>>> calssifierArray = adaboost.adaBoostTrainDS(dataMat, classLabels, 9)
# 观察 calssifierArray 值：
>>> calssifierArray
```

3. 测试 AdaBoost 分类器

```
# 如果没有弱分类器数组：
>>> dataArr, lableArr = adaboost.loadSimpData()
>>> calssifierArr = adaboost.adaBoostTrainDS(dataArr, lable1Arr, 30)
# 分类预测
>>> adaboost.adaClassify([0, 0], classifierArr)
# 其他点的分类预测：
>>> adaboost.adaClassify([5, 5], [0, 0], classifierArr)
```


4. 马疝病数据集上应用

马疝病数据集包含训练集和测试集两部分，分别存放在 horseColicTraining2.txt 和 horseColicTest2.txt 两个文件中。每份文件的每一行代表一个数据样本，具体属性没有标明；每一行的最后一列对应数据点的类别，其值只能为+1 或-1。数据存放在 txt 文件中，编写函数读取文件，将数据转化为程序可以处理的样本数据集合和类别标签列表。

```
# 导入数据集
>>> dataArr, labelArr = adaboost.loadDataSet('horseColicTraining2.txt')
>>> testArr, testLabelArr = adaboost.loadDataSet('horseColicTest2.txt')
>>> testArr, testLabelArr = adaboost.loadDataSet('horseColicTest2.txt')
>>> prediction10 = adaboost.adaClassify(testArr,classifierArray)
>>> errArr = mat(ones((67, 1)))
>>> errArr[prediction10! = mat(testLabelArr).T].sum()
```

函数全部整合在 adaboost.py 文件中，其主要的功能是将数据文件中的数据保存为算法可处理的变量，并通过预设一个代表迭代次数的列表，可以从代码中看到随着下标的增大，迭代次数也设定随之增大，也可以自己设定其数值，目的在于方便观察迭代次数对算法的影响；其次，根据训练数据和测试数据，得出在每一次迭代过程中的训练误差率和测试误差率。

5. 作业习题

（1）增大迭代次数（即修改 numIt_list 值），测试 AdaBoost 分类器的稳定性。

（2）将 adaboost.py 中的“from numpy import * ”用“import numpy as np”代替，修改其中对应的代码，使其能够正常执行。

（3）利用 sklearn.tree.DecisionTreeClassifierhe 和 sklearn. ensemble.AdaBoostClassifier 实现该数据集的预测，并比较两种模型的性能（精度和耗时）。

（4）利用下面三种模型实现该数据集的预测，并比较三种模型的性能（精度和耗时）：

sklearn. ensemble.AdaBoostClassifier

sklearn.ensemble.GradientBoostingClassifie

sklearn.ensemble.RandomForestClassifier

（5）实现 PPT 中提升回归树实例。（扩展）

（6）基于 tensorflow 实现随机森林和梯度提升树。（扩展）