**作业8.5**

1. **实验要求**

试编程实现bagging,以决策树桩为基学习器，在西瓜数据集3.0a上，训练一个bagging集成。

1. **实验原理**

给定包含m个样本的数据集，随机取出一个样本放入采样集，然后放回数据集中，再进行下一次采样，采样m次得到一个包含m个数据集的采样集。这样有的样本可以在采样集中多次出现，有的可能一次都出现不了。

照这样，可以训练T个这样的采样集，然后对每个采样集训练一个基学习器，再对这些基学习器进行组合，对于分类任务，采取简单投票法即可解决问题。

1. **实验步骤与代码**

对于本题，直接从sklearn机器学习包里调用决策树和bagging进行训练，得到划分结果，并进行表达，过程如下：

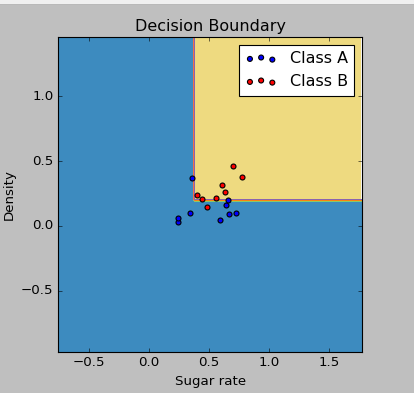
1. 读取数据并进行处理
2. 设定基学习器的数目，对数据和标签进行训练。
3. 将得出的决策结果画图显示出来。

代码如下：

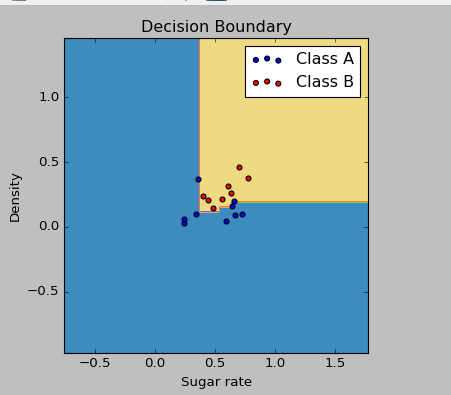
import numpy as np  
import matplotlib.pyplot as plt  
  
from sklearn.ensemble import BaggingClassifier  
from sklearn.tree import DecisionTreeClassifier  
  
file1 = open('w3.csv','r')  
data = [line.strip('\n').split(',') for line in file1]  
data = np.array(data)  
#X = [[float(raw[-7]),float(raw[-6]),float(raw[-5]),float(raw[-4]),float(raw[-3]), float(raw[-2])] for raw in data[1:,1:-1]]  
  
X = [[float(raw[-3]), float(raw[-2])] for raw in data[1:]]  
y = [1 if raw[-1]=='1' else 0 for raw in data[1:]]  
X = np.array(X)  
y = np.array(y)  
  
  
# Create and fit an AdaBoosted decision tree,不剪枝决策树  
bdt = BaggingClassifier(DecisionTreeClassifier(),  
  
 n\_estimators=10)#最后这个参数表示基学习器的个数  
  
bdt.fit(X, y)  
  
plot\_colors = "br"  
plot\_step = 0.02  
class\_names = "AB"  
  
plt.figure(figsize=(10, 5)) #调图的尺寸  
  
# Plot the decision boundaries  
plt.subplot(121)  
x\_min, x\_max = X[:, 0].min() - 1, X[:, 0].max() + 1  
y\_min, y\_max = X[:, 1].min() - 1, X[:, 1].max() + 1  
xx, yy = np.meshgrid(np.arange(x\_min, x\_max, plot\_step),  
 np.arange(y\_min, y\_max, plot\_step))  
  
Z = bdt.predict(np.c\_[xx.ravel(), yy.ravel()])  
Z = Z.reshape(xx.shape)  
cs = plt.contourf(xx, yy, Z, cmap=plt.cm.Paired)  
plt.axis("tight")  
  
# Plot the training points  
for i, n, c in zip(range(2), class\_names, plot\_colors):  
 idx = np.where(y == i)  
 plt.scatter(X[idx, 0], X[idx, 1],  
 c=c, cmap=plt.cm.Paired,  
 label="Class %s" % n)  
plt.xlim(x\_min, x\_max)  
plt.ylim(y\_min, y\_max)  
plt.legend(loc='upper right')  
plt.xlabel('Sugar rate')  
plt.ylabel('Density')  
plt.title('Decision Boundary')  
  
  
plt.tight\_layout()  
plt.subplots\_adjust(wspace=0.35)  
plt.show()

1. **实验结果**

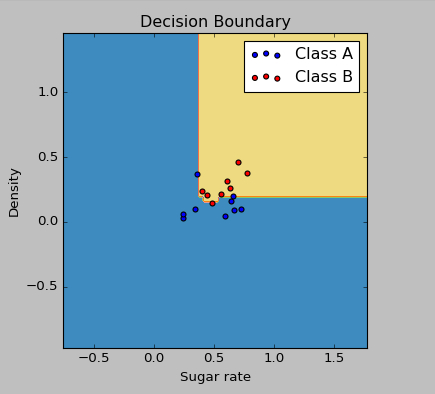
当基学习器数目为3时，得到划分结果如下图：



当基学习器数目为5时，得到结果如下图：



当基学习器数目为10时：



可以看出，基学习器数目越多，划分效果越好，但是当基学习器数目超过一定数量后，改善效果不是很大。