实验八：Boosting 集成学习

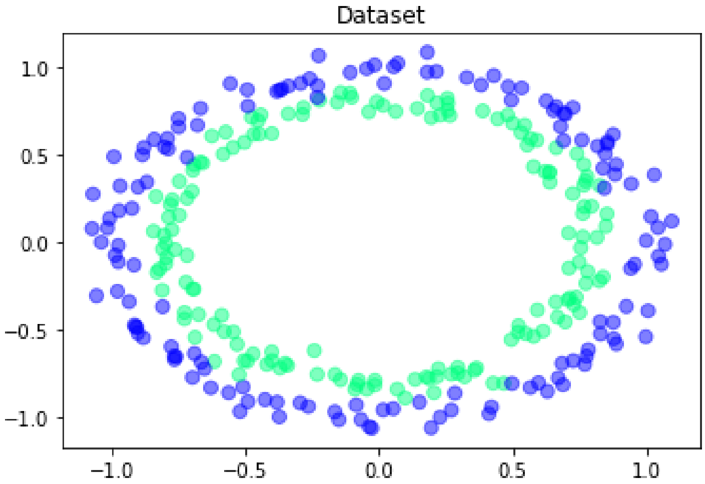
姓名：　　　　　　　　　　　　　　　　　　　　　　学号：

● 实验目的

掌握集成学习思想，掌握 boosting 策略，基于 AdaBoost 实现多分类任务。

● 实验要求

编程实现 AdaBoost 集成方法，对如下数据集进行分类。基模型采用决策树模型，划分属性指标采用信息熵指标，决策树最大深度设置为 3。将基模型数量T依次设置为$1,2,\ldots,20$，计算集成模型在测试集上的精度，并绘制集成模型精度随基模型数量增加的变化曲线。



● 实验环境

Python，numpy，matplotlib，sklearn

● 实验代码

```
import numpy as np
from sklearn import tree
from matplotlib import pyplot as plt

train_data = np.loadtxt('experiment_08_training_set.csv', delimiter=',')
test_data = np.loadtxt('experiment_08_testing_set.csv', delimiter=',')
train_x = train_data[:, 0:2]
train_y = train_data[:, 2]
test_x = test_data[:, 0:2]
test_y = test_data[:, 2]

w = np.full(train_data.shape[0], 1 / train_data.shape[0])
```

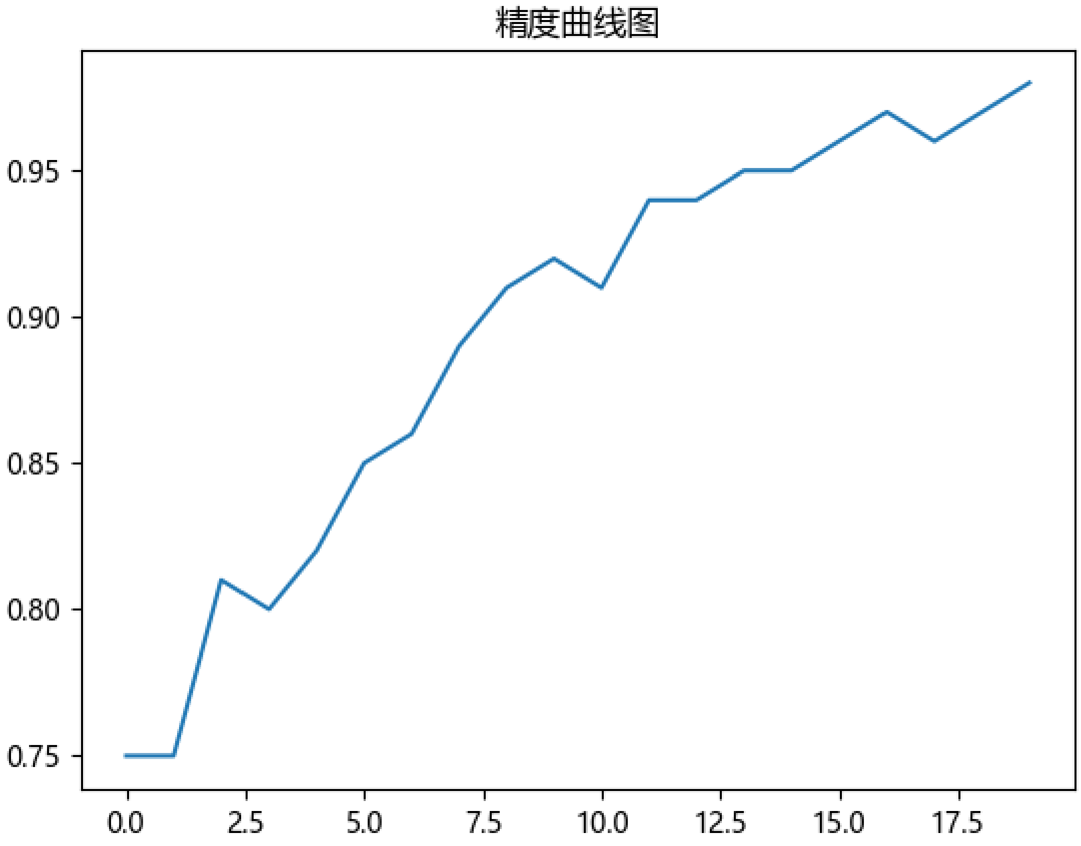
```
model_array = []
at_array = []
for i in range(1, 21):
    model = tree.DecisionTreeClassifier(random_state=1, criterion='entropy', max_depth=3)
    model.fit(train_x, train_y, sample_weight=w)
    predictions_train = model.predict(train_x)
    e_train = np.sum(w[predictions_train != train_y])
    at_train = 1 / 2 * np.log((1 - e_train) / e_train)
    w = w * np.exp(-train_y * at_train * predictions_train)
    w = w / np.sum(w)
    model_array.append(model)
    at_array.append(at_train)
predictions_array = np.zeros(100)
acc_array = []
for i in range(len(model_array)):
    predictions = model_array[i].predict(test_x)
    predictions_array = predictions_array + predictions * at_array[i]
    predictions_array_1 = np.sign(predictions_array)
    accuracy = np.sum((predictions_array_1 == test_y)) / test_y.size
    acc_array.append(accuracy)
    print(f"轮次{i + 1}:", accuracy)

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
plt.title("精度曲线图")
plt.plot(acc_array)
plt.show()
```

● 结果分析

测试集上精度

T	1	2	3	4	5	6	7	8	9	10
精度	0.75	0.75	0.81	0.80	0.82	0.85	0.86	0.89	0.91	0.92
T	11	12	13	14	15	16	17	18	19	20
精度	0.91	0.94	0.94	0.95	0.95	0.96	0.97	0.96	0.97	0.98

精度随T增加的变化曲线


精度曲线图