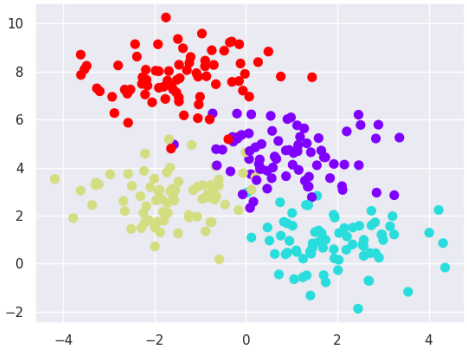
机器学习课程实验十

2022 年 12 月 8 日　苏博南　202000460020

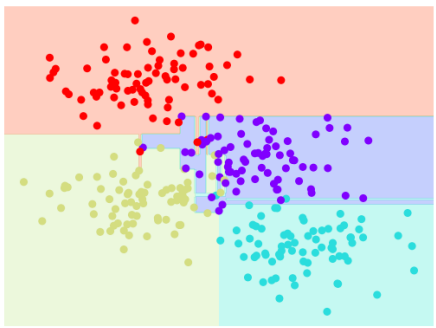
不是很清楚这个实验有啥好写的，把代码跑一遍，结果看一看好了，反之实现都是 import 一步到位（。首先是数据集打印：

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()

from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples = 300, centers = 4, random_state = 0, cluster_std = 1.0)
plt.scatter(X[:, 0], X[:, 1], c = y, s = 50, cmap = 'rainbow')
plt.show()
```



然后 import 一步到位作出决策树：

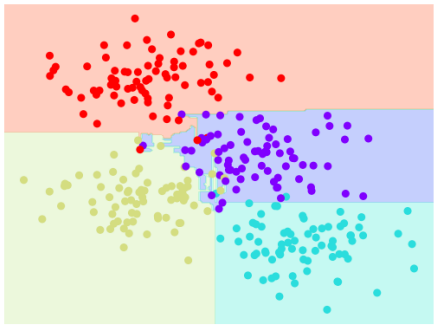
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier().fit(X, y)
visualize_classifier(DecisionTreeClassifier(), X, y)
```



然后会发现有 overfitting 的现象，譬如紫色区域里就会有一条很细很细的青色线。然后

为了求得稳定性和减少过拟合线性，可以考虑使用 bagging 算法，即用多个弱分类算法进行投票。import 下又可以一步到位：

```
1    from sklearn.ensemble import BaggingClassifier
2    from sklearn.ensemble import RandomForestClassifier
3
4    bag = BaggingClassifier(tree, n_estimators = 100, max_samples = 0.8,
   random_state = 1)
5    bag.fit(X, y)
6    visualize_classifier(bag, X, y)
7    plt.show()
```



最后为了提高分类边界的精确度，可以考虑使用 random forest 算法，即在 bagging 算法的基础上使用多个决策树：

```
1    from sklearn.ensemble import RandomForestClassifier
2    model = RandomForestClassifier(n_estimators = 100, random_state = 0)
3    visualize_classifier(model, X, y)
4    plt.show()
```