02 决策树

一、利用决策树对红酒分类

1.1. 导入所需的模块

```
from sklearn import tree
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
import pandas as pd
```

1.2 导入数据

```
wine = load_wine()
#wine.data.shape
```

```
pd.concat([pd.DataFrame(wine.data),pd.DataFrame(wine.target)],axis=1)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	0
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	1.74	740.0	2
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	1.56	750.0	2
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	1.56	835.0	2
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	1.62	840.0	2
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	1.60	560.0	2

178 rows × 14 columns

1.3 拆分训练集和测试集

```
X_train,X_test,y_train,y_test = train_test_split(wine.data,wine.target,test_size=0.3)
X_train.shape
X_test.shape
```

1.4 建立模型

```
estimator = tree.DecisionTreeClassifier(criterion='entropy')
model = estimator.fit(X_train,y_train)
```

```
score = model.score(X_test,y_test)
score
```

```
0.9074074074074074
```

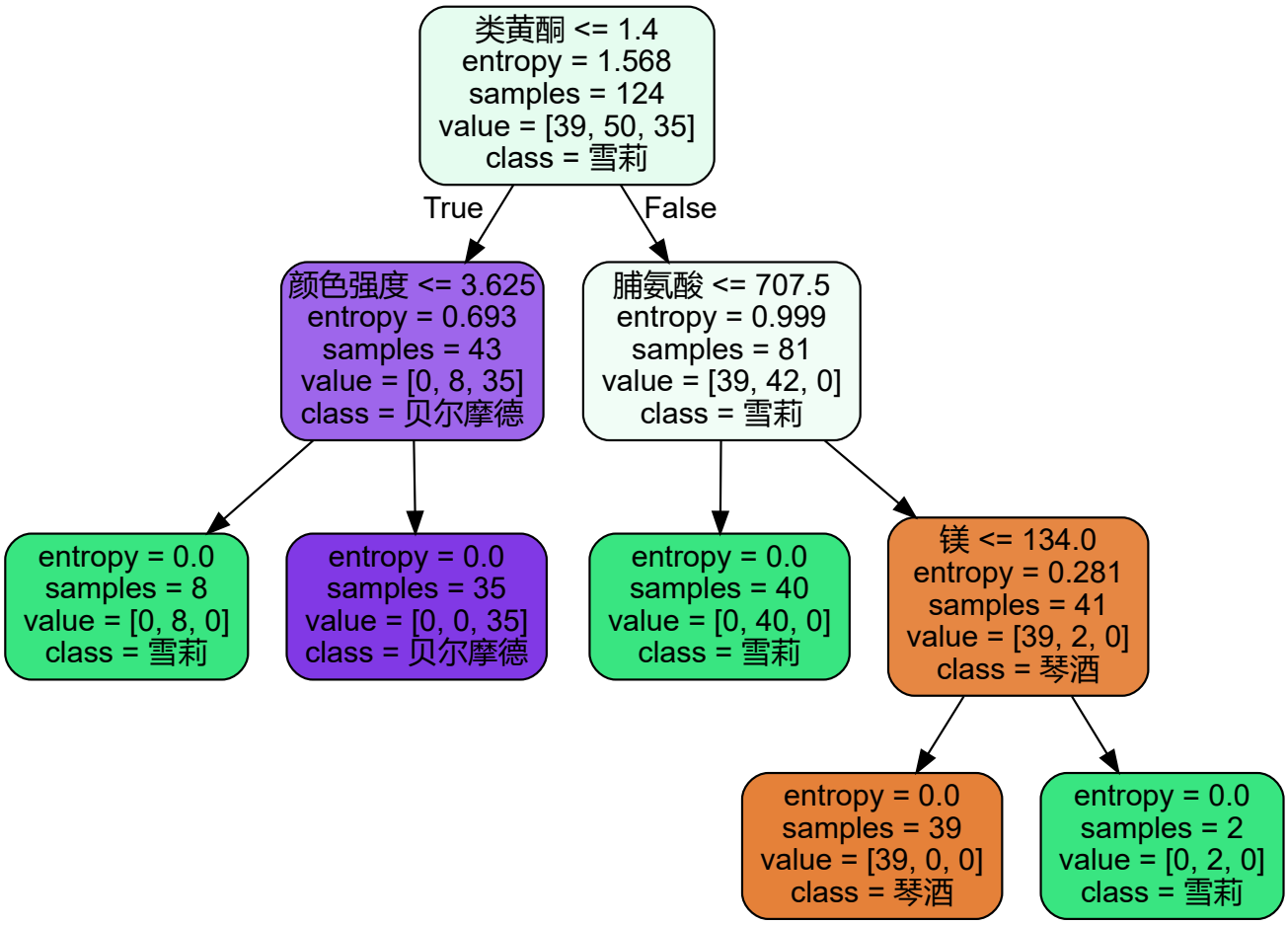
```
# conda install python-graphviz
# pip install python-graphviz
```

1.5 可视化决策树

```
import graphviz
```

```
feature_name = ['酒精', '苹果酸', '灰', '灰的碱性', '镁', '总酚', '类黄酮',
                '非黄烷类酚类', '花青素', '颜色强度', '色调', 'od280/od315稀释葡萄酒', '脯氨酸']

dot_data = tree.export_graphviz(
    model,
    out_file=None,
    feature_names=feature_name,
    class_names=["琴酒","雪莉","贝尔摩德"],
    filled=True,
    rounded=True
)
graph = graphviz.Source(dot_data)
graph
```



1.6 探索决策树

```
model.feature_importances_ #特征重要性
```

```
array([0.        , 0.        , 0.        , 0.        , 0.05928334,
       0.        , 0.4306412 , 0.        , 0.        , 0.1532585 ,
       0.        , 0.        , 0.35681696])
```

```
[*zip(feature_name,model.feature_importances_)]
```

```
[('酒精', 0.0),
 ('苹果酸', 0.0),
 ('灰', 0.0),
 ('灰的碱性', 0.0),
 ('镁', 0.059283337236752734),
 ('总酚', 0.0),
 ('类黄酮', 0.43064120044882187),
 ('非黄烷类酚类', 0.0),
 ('花青素', 0.0),
 ('颜色强度', 0.1532585046752736),
 ('色调', 0.0),
 ('od280/od315稀释葡萄酒', 0.0),
 ('脯氨酸', 0.3568169576391518)]
```

看一下模型性能

```
model.apply(X_test)
```

```
array([8, 7, 3, 2, 5, 7, 7, 3, 3, 5, 3, 2, 7, 7, 7, 3, 5, 5, 7, 7, 5, 7,
       5, 7, 7, 5, 7, 5, 5, 5, 7, 3, 3, 7, 7, 7, 7, 7, 7, 5, 3, 7, 5, 5,
       3, 7, 3, 5, 5, 7, 5, 3, 3, 5], dtype=int64)
```

```
model.predict(X_test)
```

```
array([1, 0, 2, 1, 1, 0, 0, 2, 2, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 0, 1, 1, 1, 0, 2, 2, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 1,
       2, 0, 2, 1, 1, 0, 1, 2, 2, 1])
```

```

```