1 红酒类别分类 ¶

2 1.导入库

In [1]:
```
from sklearn import tree
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split fdsfsd
```

3 2.导入数据

In [2]:
```
wine=load_wine()
```

In [3]:
```
wine
```

Out[3]:
```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
         1.185e+03],
        ...,
        [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
         5.600e+02]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2]),
 'frame': None,
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': '.. _wine_dataset:\n\nWine recognition dataset\n------------------------\n\n**Data Set Characteristics:**\n\n    :Number of Instances: 178 (50 in each of three classes)\n    :Number of Attributes: 13 numeric, predictive attributes and the class\n    :Attribute Information:\n \t\t- Alcohol\n \t\t- Malic acid\n \t\t- Ash\n\t\t- Alcalinity of ash  \n \t\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids\n \t\t- Nonflavanoid phenols\n \t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD280/OD315 of diluted wines\n \t\t- Proline\n\n    - class:\n            - class_0\n            - class_1\n            - class_2\n\t\t\n    :Summary Statistics:\n    \n    ============================== ==== ===== ======= =====\n                                   Min   Max   Mean     SD\n    ============================== ==== ===== ======= =====\n    Alcohol:                       11.0  14.8    13.0   0.8\n    Malic Acid:                    0.74  5.80    2.34  1.12\n    Ash:                           1.36  3.23    2.36  0.27\n    Alcalinity of Ash:             10.6  30.0    19.5   3.3\n    Magnesium:                     70.0 162.0    99.7  14.3\n    Total Phenols:                 0.98  3.88    2.29  0.63\n    Flavanoids:                    0.34  5.08    2.03  1.00\n    Nonflavanoid Phenols:          0.13  0.66    0.36  0.12\n    Proanthocyanins:               0.41  3.58    1.59  0.57\n    Colour Intensity:              1.3  13.0     5.1   2.3\n    Hue:                           0.48  1.71    0.96  0.23\n    OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71\n    Proline:                       278  1680     746   315\n    ============================== ==== ===== ======= =====\n\n    :Missing Attribute Values: None\n    :Class Distribution: class_0 (59), class_1 (71), class_2 (48)\n    :Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n    :Date: July, 1988\n\nThis is a copy of UCI ML Wine recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data\n\nThe data is the results of a chemical analysis of wines grown in the same\nregion in Italy by three different cultivators. There are thirteen different\nmeasurements taken for different constituents found in the three types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVUS - \nAn Extendible Package for Data Exploration, Classification and Correlation. \nInstitute of Pharmaceutical and Food Analysis and Technologies,\nVia Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/ml]. Irvine, CA: University of California,\nSchool of Information and Computer Science. \n\n.. topic:: References\n\n  (1) S. Aeberhard, D. Coomans and O. de Vel, \n  Comparison of Classifiers in High Dimensional Settings, \n  Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of  \n  Mathematics and Statistics, James Cook University of North Queensland. \n  (Also submitted to Technometrics). \n\n  The data was used with many others for comparing various \n  classifiers. The classes are separable, though only RDA \n  has achieved 100% correct classification. \n  (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) \n  (All results using the leave-one-out technique) \n\n  (2) S. Aeberhard, D. Coomans and O. de Vel, \n  "THE CLASSIFICATION PERFORMANCE OF RDA" \n  Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \n  Mathematics and Statistics, James Cook University of North Queensland. \n  (Also submitted to Journal of Chemometrics).\n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline']}
```

In [4]:
```
wine.data
```

```
array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
        1.065e+03],
       [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
        1.050e+03],
       [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
        1.185e+03],
       ...,
       [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
        8.350e+02],
       [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
        8.400e+02],
       [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
        5.600e+02]])
```

In [5]:
```
wine.target
```

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

In [6]:
```
wine.data.shape
```

```
(178, 13)
```

3.1 2.1把行列数据合并成表格形式

In [7]:
```
import pandas as pd
```

In [8]:
```
pd.concat([pd.DataFrame(wine.data),pd.DataFrame(wine.target)],axis=1)
```

	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

In [9]:
```
wine.feature_names
```

```
['alcohol',
 'malic_acid',
 'ash',
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']
```

In [10]:
```
wine.target_names
```

```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

3.2 2.2分割训练集和测试集

In [11]:
```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(wine.data,wine.target,test_size=0.3)
```

In [12]:
```
Xtrain.shape
```

```
(124, 13)
```

In [13]:
```
Ytrain
```

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

4 3.训练模型

4.1 3.1搭建模型

In [14]:
```
clf = tree.DecisionTreeClassifier(criterion="entropy"
                                  # 随机性
                                  ,random_state=30
                                  ,splitter="random"
                                  # 剪枝
                                  ,max_depth = 4 #最大层数为3 不含根
                                  # ,min_samples_leaf=5 #叶子结点包含最少样本数
                                  # ,min_samples_split=20 #中间节点包含最少样本数
                                  )
clf = clf.fit(Xtrain, Ytrain)
score = clf.score(Xtest, Ytest) #返回预测的准确度accuracy
score
```

```
0.9074074074074074
```

4.2 3.2画树

In [15]:
```
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
import graphviz
dot_data = tree.export_graphviz(clf
                                ,feature_names = feature_name
                                ,class_names=["琴酒","雪莉","贝尔摩德"]
                                ,filled=True #填充颜色，颜色深度和不纯度有关
                                ,rounded=True #框的形状
                               )
```

In [16]:
```
graph = graphviz.Source(dot_data)
graph
```

```
<graphviz.files.Source at 0x1ab6e1bbc48>
```

4.3 3.3测试各个特征值的权重

In [17]:
```
clf.feature_importances_
```

```
array([0.14919183, 0.02028786, 0.02624354, 0.01995807, 0.        ,
       0.        , 0.49343547, 0.        , 0.01523205, 0.01116594,
       0.        , 0.18185863, 0.08262662])
```

4.4 3.4给特征值添加名称

In [18]:
```
[*zip(feature_name,clf.feature_importances_)]
```

```
[('酒精', 0.14919183271291966),
 ('苹果酸', 0.020287855558148975),
 ('灰', 0.026243535272012935),
 ('灰的碱性', 0.01995807000571754),
 ('镁', 0.0),
 ('总酚', 0.0),
 ('类黄酮', 0.49343546921712306),
 ('非黄烷类酚类', 0.0),
 ('花青素', 0.015232053722644873),
 ('颜色强度', 0.011165938842047928),
 ('色调', 0.0),
 ('od280/od315稀释葡萄酒', 0.1818586259206247),
 ('脯氨酸', 0.0826266187487606)]
```

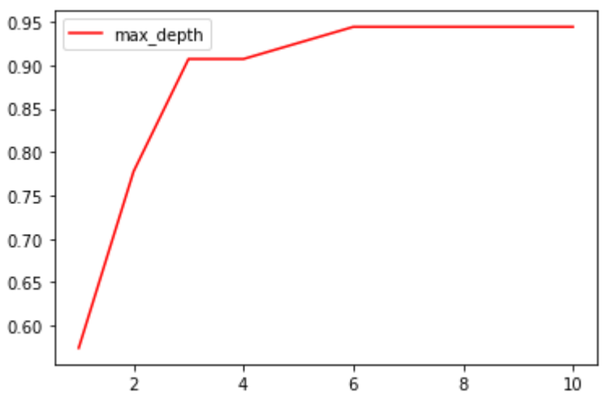
In [19]:
```
score = clf.score(Xtrain,Ytrain)
score
```

```
0.9596774193548387
```

5 4.用matplotlib画图，调参数

In [20]:
```
import matplotlib.pyplot as plt
test = []
for i in range(10):
    clf = tree.DecisionTreeClassifier(max_depth=i+1
                                      ,criterion="entropy"
                                      ,random_state=30
                                      ,splitter="random"
                                      )
    clf = clf.fit(Xtrain, Ytrain)
    score = clf.score(Xtest, Ytest)
    test.append(score)
plt.plot(range(1,11),test,color="red",label="max_depth")
plt.legend()
plt.show()
```

```
Matplotlib is building the font cache; this may take a moment.
```



In [21]:
```
#apply返回每个测试样本所在的叶子节点的索引
clf.apply(Xtest)
```

```
array([ 7,  9, 23,  9, 31, 14, 23, 20,  7,  7, 20, 20, 20,  4, 31,  7,  7,
        7, 23,  7,  7, 31, 23,  7, 14, 23, 23, 31, 20, 31, 31, 31,  7, 23,
       23, 31,  9,  7,  7,  7, 31, 14, 20, 20, 31,  7, 28, 14, 31, 28, 11,
       28, 31, 23], dtype=int64)
```

In [22]:
```
#predict返回每个测试样本的分类/回归结果
clf.predict(Xtest)
```

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

In [ ]:
```

```