机器学习之决策树和异常检测实战

1. 决策树

In [13]:
```
%matplotlib inline

from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import accuracy_score
from matplotlib import pyplot as plt
```

In [3]:
```
import numpy as np
import pandas as pd
data = pd.read_csv('data\iris_data.csv')

# define x, y
X = data.drop(['target','label'], axis = 1)
y = data.loc[:, 'label']
```

In [4]:
```
X
```

Out[4]:

	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

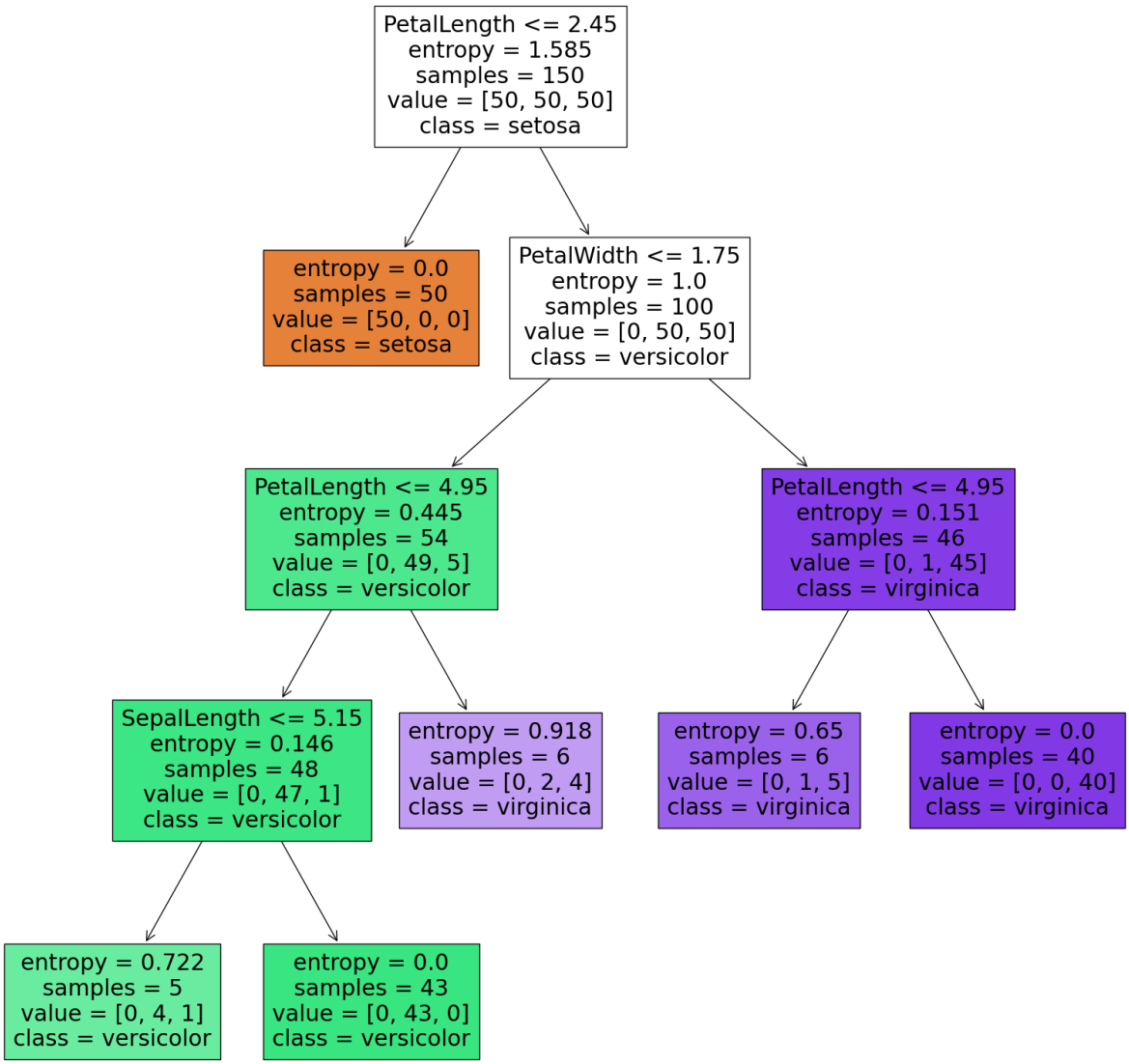
In [9]:
```
#建构并训练模型
dc_tree = DecisionTreeClassifier(criterion = 'entropy', min_samples_leaf = 5)
# 参数criterion = 'entropy'表示使用信息熵作为特征选择的度量标准，是ID3决策树模型;
# 参数min_samples_leaf = 5表示决策树分裂出来的叶子最少要有5个样本，如果少于5个样本节点就
```

```
dc_tree.fit(X, y)

# 预测数据
y_predict = dc_tree.predict(X)
print(y_predict)
# 评估性能（准确度）
accuracy = accuracy_score(y, y_predict)
print(accuracy)
```

```
[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 2 1 1 1
 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 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]
0.9733333333333334
```

In [15]:
```
fig = plt.figure(figsize = (20,20))
tree.plot_tree(dc_tree, filled = True, feature_names = ['SepalLength', 'SepalWidth'
               class_names = ['setosa', 'versicolor', 'virginica']);
```



In [24]:
```
# 预测一个测试鸢尾花：5.0 3.0 1.3 0.3
x_new = pd.DataFrame( [[5.0,3.0,1.3,0.3]], columns=X.columns)
y_predict_test = dc_tree.predict( x_new )
print(y_predict_test)
```

```
[0]
```

可见花萼长度sepal length为5.0、花萼宽度sepal width为3.0、花瓣长度petal length为1.3、花瓣宽度petal width0.3的鸢尾花属于类别0，即属于山鸢尾花Iris-setosa

In [21]:
```
target = data[['label','target']].drop_duplicates(keep='first')
target
```

Out[21]:

	label	target
0	0	Iris-setosa
50	1	Iris-versicolor
100	2	Iris-virginica

In [ ]:
```

```

2. 异常检测

In [7]:
```
%matplotlib inline

import pandas as pd
from sklearn.covariance import EllipticEnvelope
import matplotlib.pyplot as plt

data = pd.read_csv('data/anomaly_data.csv')
data
```

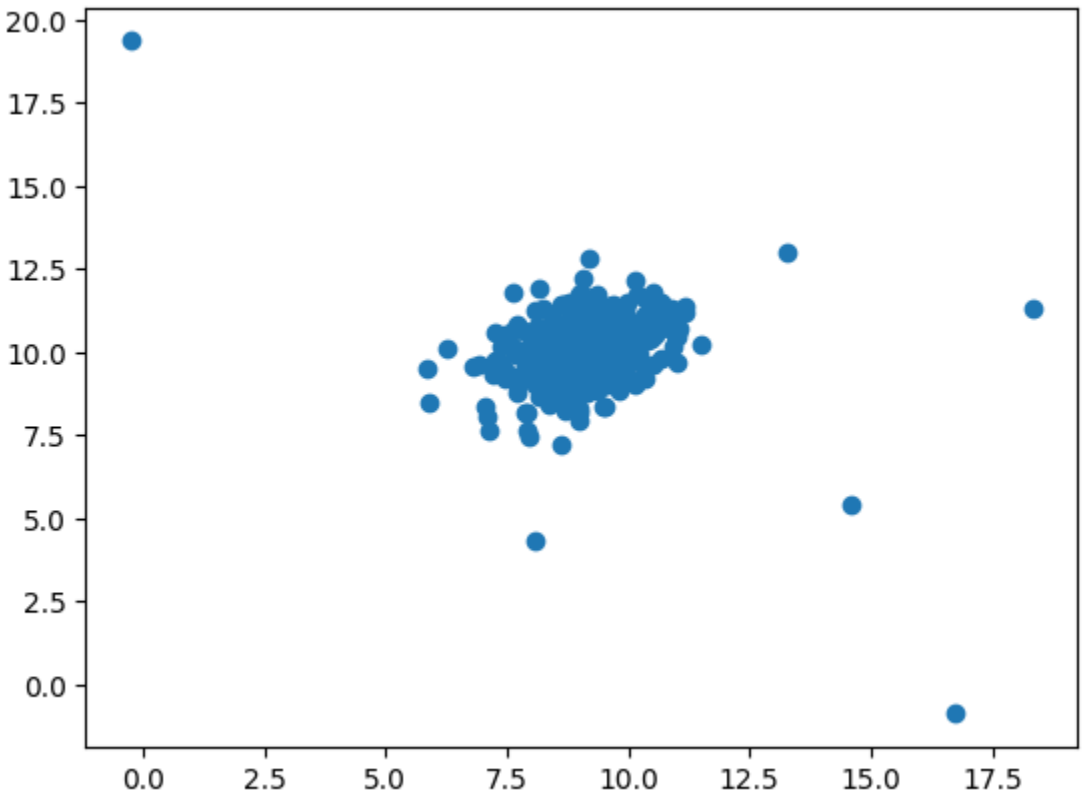
Out[7]:

	x1	x2
0	8.046815	9.741152
1	8.408520	8.763270
2	9.195915	10.853181
3	9.914701	11.174260
4	8.576700	9.042849
...	...	...
302	7.476629	9.459370
303	14.582573	5.411619
304	18.339868	11.298874
305	13.261188	12.978309
306	-0.247387	19.350407

307 rows × 2 columns

In [4]:
```
fig, ax = plt.subplots()
ax.scatter(data['x1'],data['x2'])
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x8d5a340>
```



In [17]:
```
ad_model = EllipticEnvelope() # anomaly detection --> ad
```

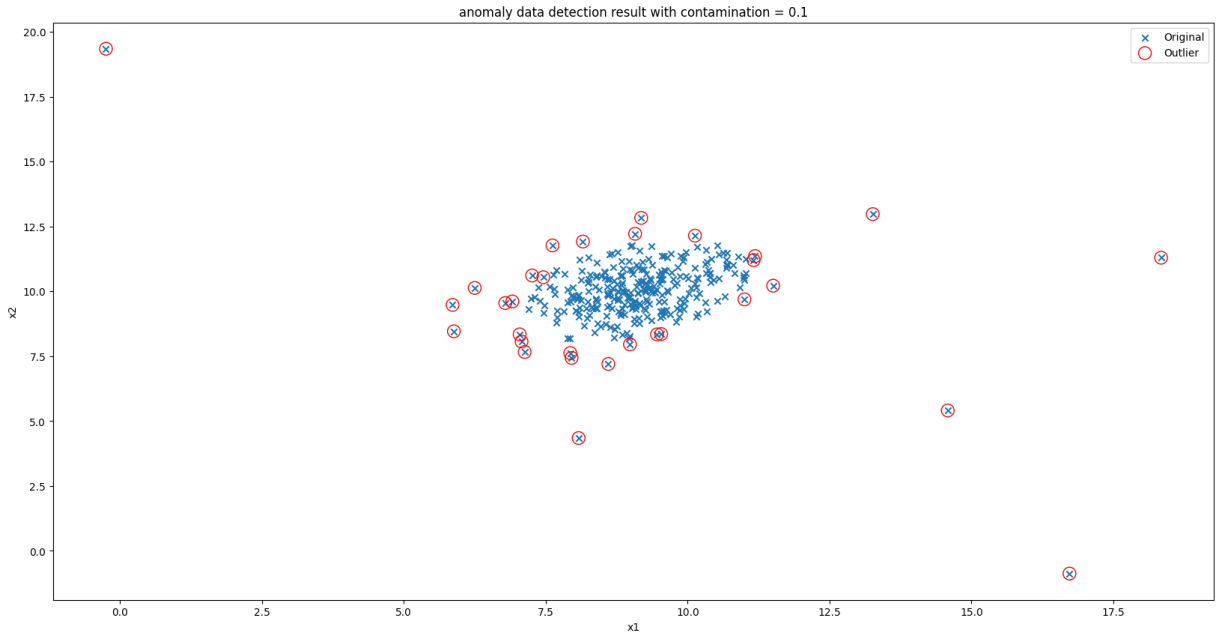
```
ad_model.fit(data)
y_predict = ad_model.predict(data) # 预测结果y_predict中只有1和-1，表示正常和异常
print(pd.value_counts(y_predict))
```

```
 1    276
-1     31
dtype: int64
```

In [16]:
```
fig, ax = plt.subplots(figsize = (20, 10))
ax.scatter(data.loc[:,'x1'], data.loc[:, 'x2'], marker = 'x', label='Original')
ax.scatter(data.loc[:,'x1'][y_predict == -1], data.loc[:, 'x2'][y_predict == -1], m
           facecolor='none',edgecolor='red',s=150, label='Outlier')

plt.title('anomaly data detection result with contamination = 0.1')
plt.xlabel('x1')
plt.ylabel('x2')
plt.legend()
```

Out[16]: <matplotlib.legend.Legend at 0x2dc69100>



In [ ]: