

머신러닝과 딥러닝

Report6

소프트웨어학과

2016312568 정희운

```
In [5]: from sklearn.datasets import load_iris
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sn

In [2]: iris = load_iris()
iris_frame = pd.DataFrame(data=np.c_[iris['data'], iris['target']], columns = iris['feature_names'] + ['target'],
iris_frame['target'] = iris_frame['target'].map({1:"versicolor",0:"setosa",2:"virginica"})
X = iris_frame.iloc[:,1:-1]
Y = iris_frame.iloc[:,[-1]]
iris_frame

Out[2]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa
10	5.4	3.7	1.5	0.2	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
17	5.1	3.5	1.4	0.3	setosa
18	5.7	3.8	1.7	0.3	setosa
19	5.1	3.8	1.5	0.3	setosa
20	5.4	3.4	1.7	0.2	setosa
21	5.1	3.7	1.5	0.4	setosa
22	4.6	3.6	1.0	0.2	setosa
23	5.1	3.3	1.7	0.5	setosa
24	4.8	3.4	1.9	0.2	setosa
25	5.0	3.0	1.6	0.2	setosa

지난 실습에서 사용한 Iris 데이터를 가져온다.

```
In [3]: from sklearn.tree import DecisionTreeClassifier

##아래 하이퍼파라미터 수정으로 결과 확인
clf = DecisionTreeClassifier(random_state=0, criterion='gini', max_depth=5)
import matplotlib.colors as colors

df1 = iris_frame[["sepal length (cm)", "sepal width (cm)", "target" ]]
X = df1.iloc[:,0:2]
Y = df1.iloc[:,2].replace({'setosa':0,'versicolor':1,'virginica':2}).copy()

clf.fit(X,Y)
N=100
```

Iris 데이터셋 중에서 sepal과 관련된 데이터만을 이용한다. 여기서 max depth의 크기에 따라 tree의 depth가 달라지게 된다.

```
In [6]: X = np.linspace(4,8,N)
Y = np.linspace(1.5,5,N)
X_,Y_ = np.meshgrid(X_,Y_)

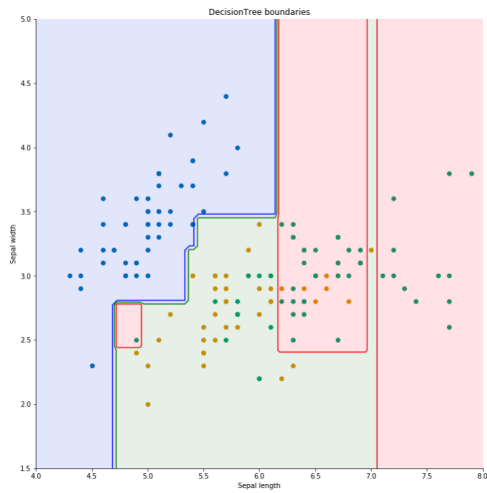
color_list = ['Blues','Greens','Reds']
my_norm = colors.Normalize(vmin=-1,vmax=1)
g = sns.FacetGrid(iris_frame, hue = 'target', size = 10,
                  palette = 'colorblind').map(plt.scatter, "sepal length (cm)", "sepal width (cm)", "target")

my_ax = g.ax
zz = np.array([clf.predict([[xx,yy]]) [0] for xx,yy in zip(np.ravel(X_),np.ravel(Y_))])
Z = zz.reshape(X_.shape)

my_ax.contourf(X_,Y_,Z,2,alpha=.1, colors = ('blue','green','red'))
my_ax.contour(X_,Y_,Z,2,alpha=.1, colors = ('blue','green','red'))

my_ax.set_xlabel('Sepal length')
my_ax.set_ylabel('Sepal width')
my_ax.set_title('DecisionTree boundaries')
plt.show()
```

C:\Users\user\Anaconda3\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: g: The size parameter has been renamed to height; please update your code. warnings.warn(msg, UserWarning)



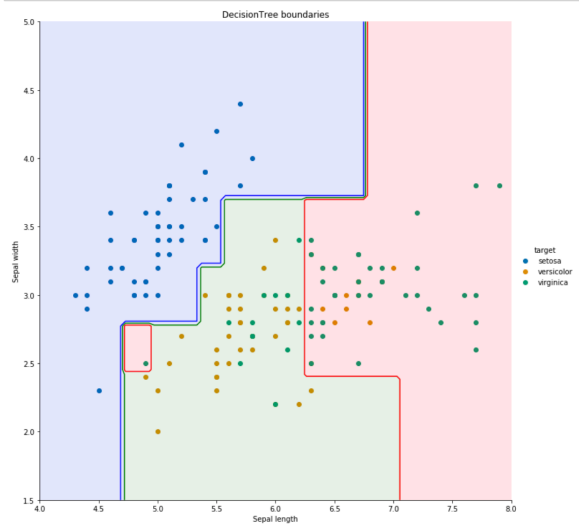
```
!]: X = np.linspace(4,8,N)
Y = np.linspace(1.5,5,N)
X_,Y_ = np.meshgrid(X_,Y_)

color_list = ['Blues','Greens','Reds']
my_norm = colors.Normalize(vmin=-1,vmax=1)
g = sns.FacetGrid(iris_frame, hue = 'target', height = 10,
                  palette = 'colorblind').map(plt.scatter, "sepal length (cm)", "sepal width (cm)", "target").add_legend()

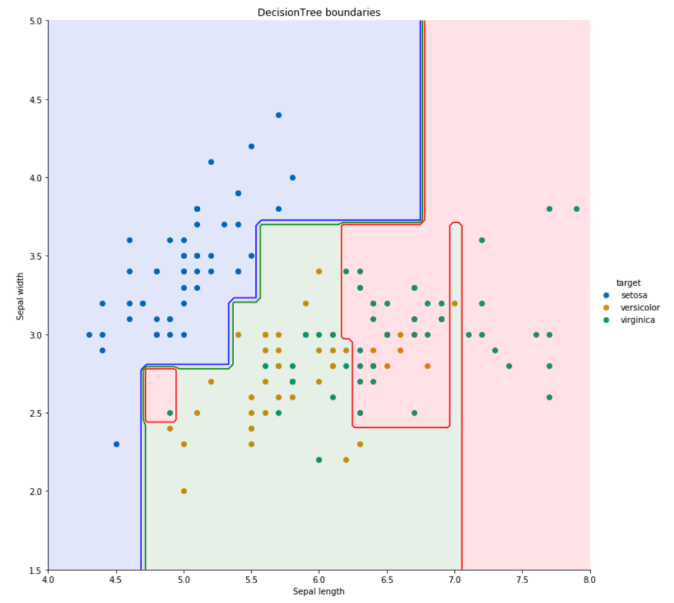
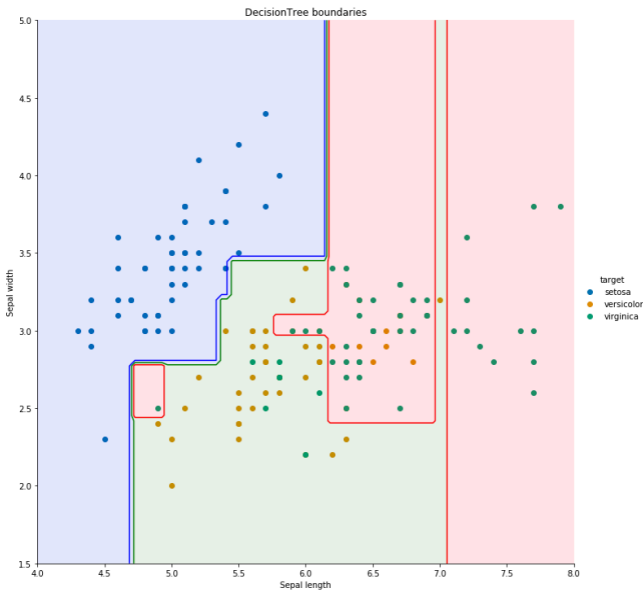
my_ax = g.ax
zz = np.array([clf.predict([[xx,yy]]) [0] for xx,yy in zip(np.ravel(X_),np.ravel(Y_))])
Z = zz.reshape(X_.shape)

my_ax.contourf(X_,Y_,Z,2,alpha=.1, colors = ('blue','green','red'))
my_ax.contour(X_,Y_,Z,2,alpha=.1, colors = ('blue','green','red'))

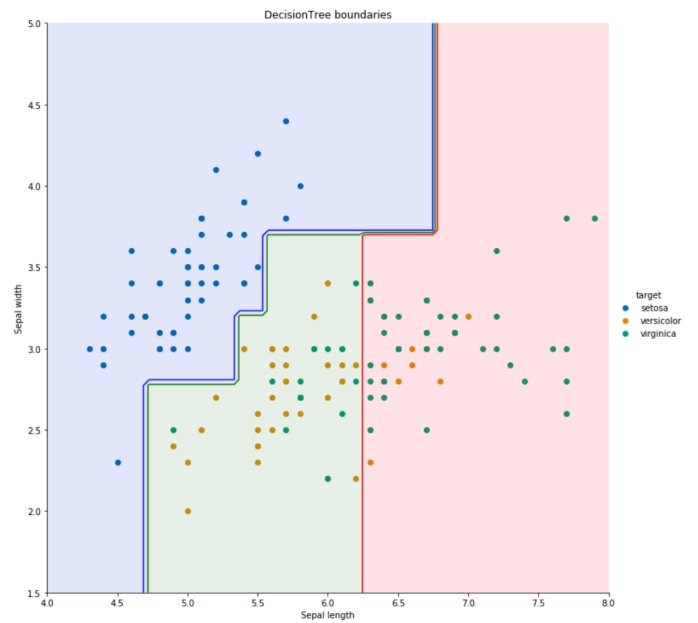
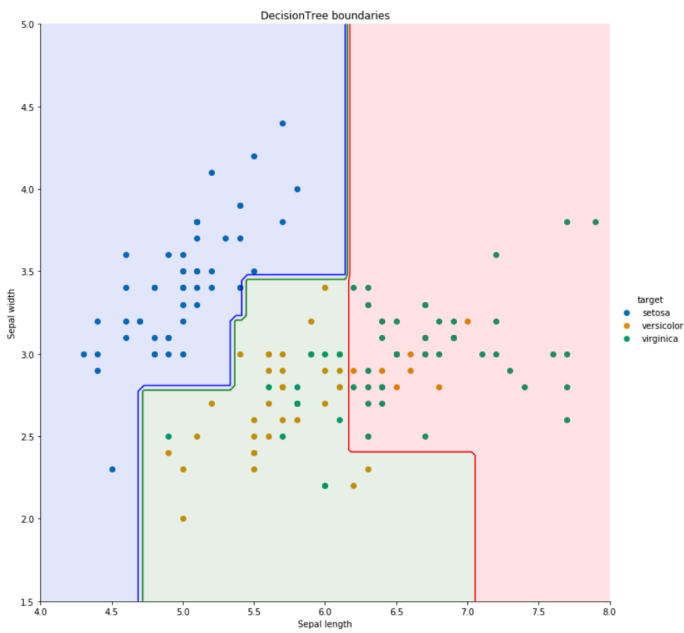
my_ax.set_xlabel('Sepal length')
my_ax.set_ylabel('Sepal width')
my_ax.set_title('DecisionTree boundaries')
plt.show()
```



학습한 데이터를 표현한 것이다. 이때의 Max depth는 5이다. 왼쪽은 의사결정 트리의 criterion = gini일 경우이고, 우측은 criterion = entropy일 경우이다.



위 그림은 Max depth를 6으로 했을 경우이다. 좌측은 criterion = gini, 우측은 criterion = entropy일 경우이다.



위 그림은 Max depth를 4로 했을 경우이다. 좌측은 criterion = gini, 우측은 criterion = entropy일 경우이다.