Data Analytics Week 13 Assignment

202011431 산업공학과 차승현

Classifier (Decision Tree)

Analysis Procedure

Data 및 Module Import

In [70]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
```

In [72]:
```
df = pd.read_csv(r'data_week13.csv')
df
```

Out [72]:

	caseno	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	1	5.1	3.5	1.4	0.2	setosa
1	2	4.9	3.0	1.4	0.2	setosa
2	3	4.7	3.2	1.3	0.2	setosa
3	4	4.6	3.1	1.5	0.2	setosa
4	5	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	virginica
146	147	6.3	2.5	5.0	1.9	virginica
147	148	6.5	3.0	5.2	2.0	virginica
148	149	6.2	3.4	5.4	2.3	virginica
149	150	5.9	3.0	5.1	1.8	virginica

150 rows × 6 columns

SepalLength, SepalWidth, PetalLength, PetalWidth를 독립변수 X로 설정하고, Species를 종속변수 Y로 설정한다.

```
X = df.iloc[:, 1:-1]
y = df.iloc[:, -1]
```

```
X
```

	SepalLength	SepalWidth	PetalLength	PetalWidth
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

```
y
```

```
0          setosa
1          setosa
2          setosa
3          setosa
4          setosa
          ...
145     virginica
146     virginica
147     virginica
148     virginica
149     virginica
Name: Species, Length: 150, dtype: object
```

모델의 학습을 위하여 X데이터와 y데이터를 train용과 test용으로 분할한다. (test size는 전체 데이터의 20%로 설정한다.)

```
In [94]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=11)
```

```
dt_clf = DecisionTreeClassifier(max_depth = 5)
dt_clf.fit(X_train, y_train)
print('max depth가 5인 경우, decision tree의 정확도: ',dt_clf.score(X_test, y_test))
```

```
max depth가 5인 경우, decision tree의 정확도:  0.8666666666666667
```

```
dt_clf4 = DecisionTreeClassifier(max_depth = 4)
dt_clf4.fit(X_train, y_train)
print('max depth가 5인 경우, decision tree의 정확도: ',dt_clf4.score(X_test, y_test))
```

```
max depth가 5인 경우, decision tree의 정확도:  0.9333333333333333
```

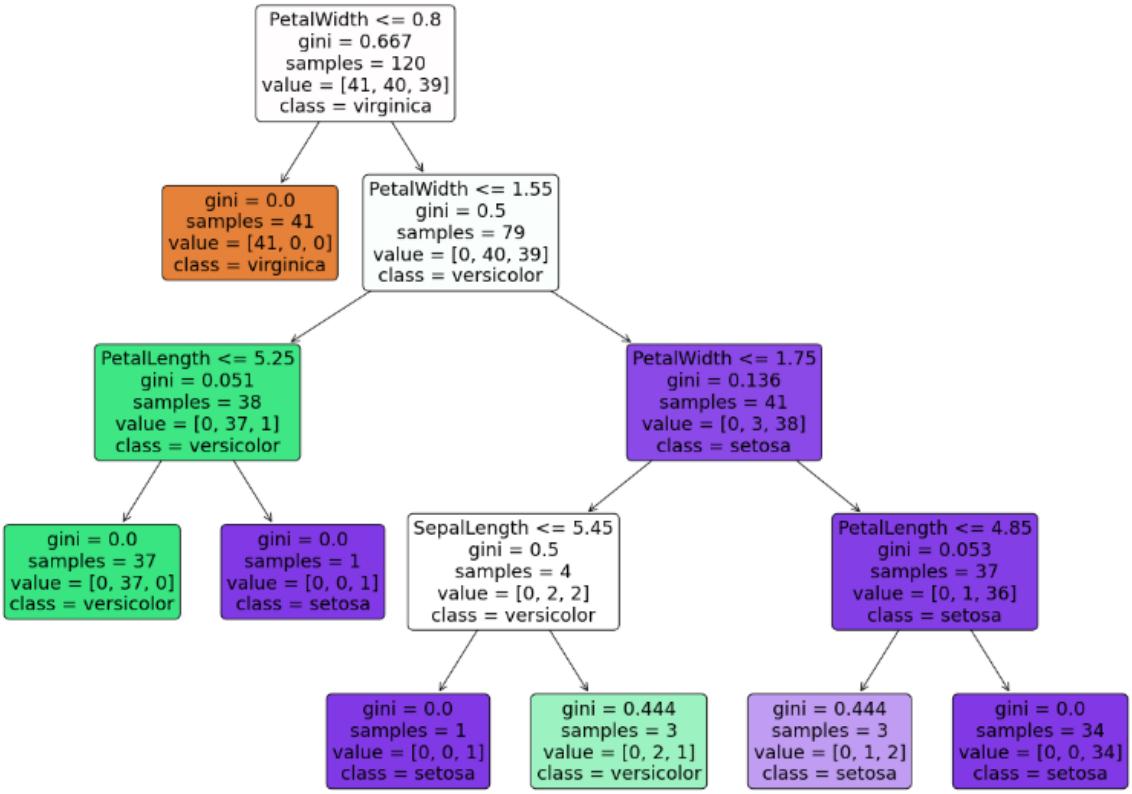
```
dt_clf3 = DecisionTreeClassifier(max_depth = 3)
dt_clf3.fit(X_train, y_train)
print('max depth가 5인 경우, decision tree의 정확도: ',dt_clf3.score(X_test, y_test))
```

```
max depth가 5인 경우, decision tree의 정확도:  0.9333333333333333
```

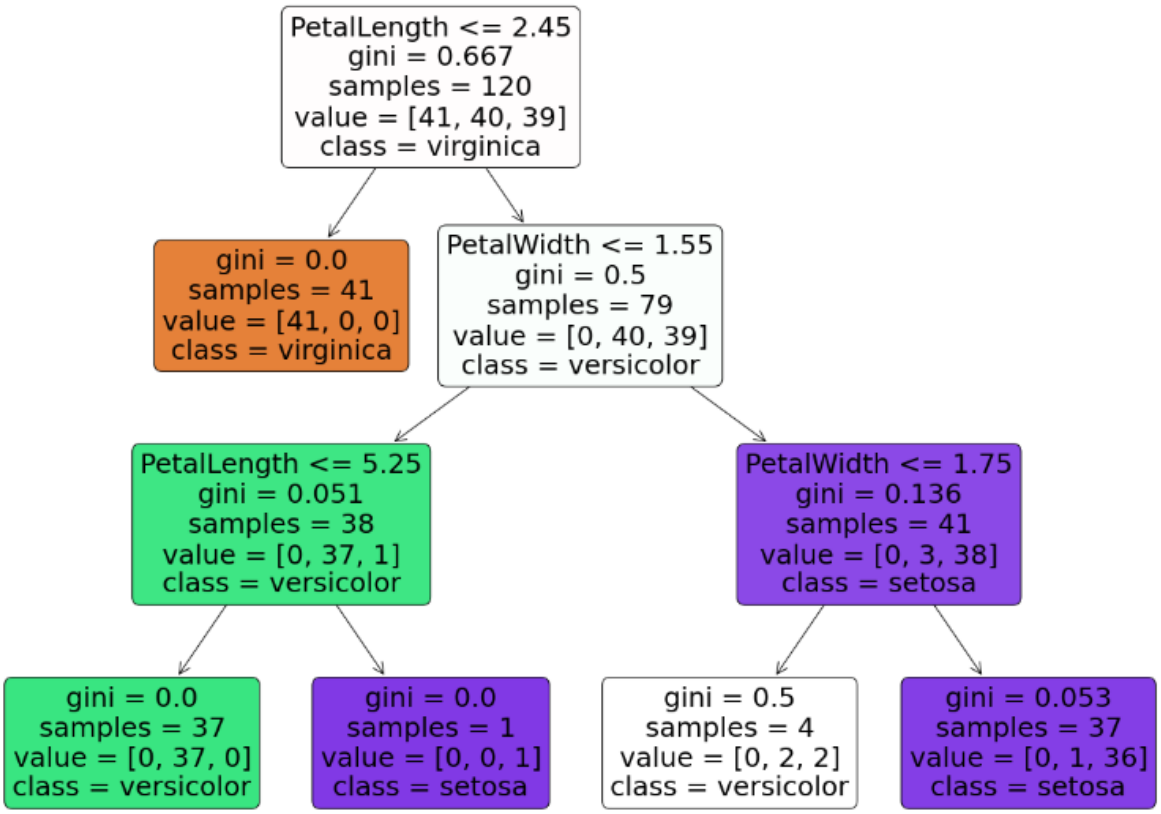
Max depth에 따라서, 각 decision tree model의 정확도를 산출해본 결과, max depth가 4인 경우와 3인 경우의 정확도가 동일하게 나타났고, 해당 모델의 plot을 그려보았다.

```
import matplotlib.pyplot as plt
from sklearn import tree

plt.figure( figsize=(20,15) )
tree.plot_tree(dt_clf4,
               class_names=list(set(y)),
               feature_names=X.columns.to_list(),
               impurity=True, filled=True,
               rounded=True)
```



Max depth가 4인 모델의 plot



Max depth가 3인 모델의 plot