부제 : 붓꽃 구분하기

결정트리로 머신러닝을 구현해보자

91714167 유재겸

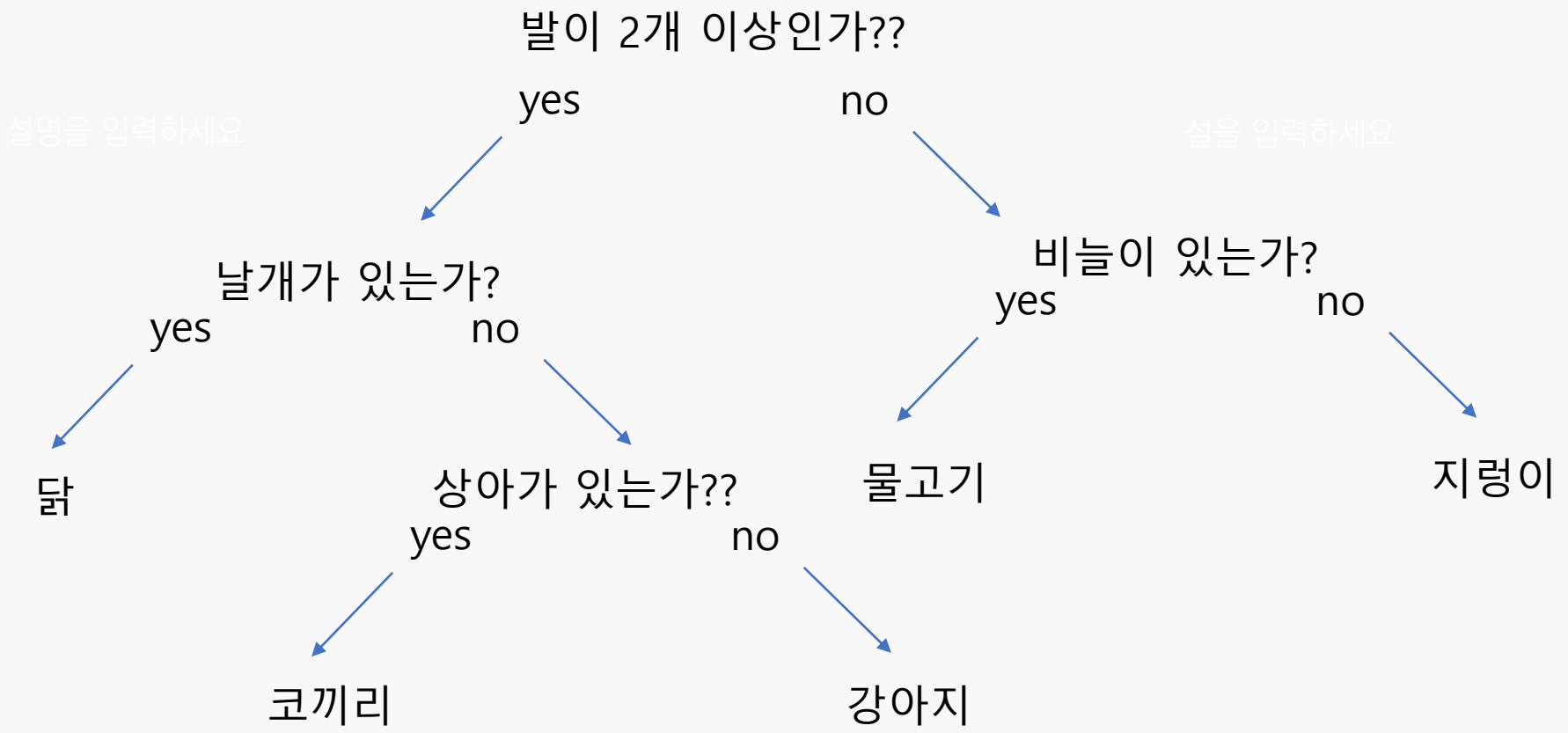
CONTENTS

CONTENTS

CONTENTS

CONTENTS

01 DecisionTreeClassifier



02 붓꽃 구분하기

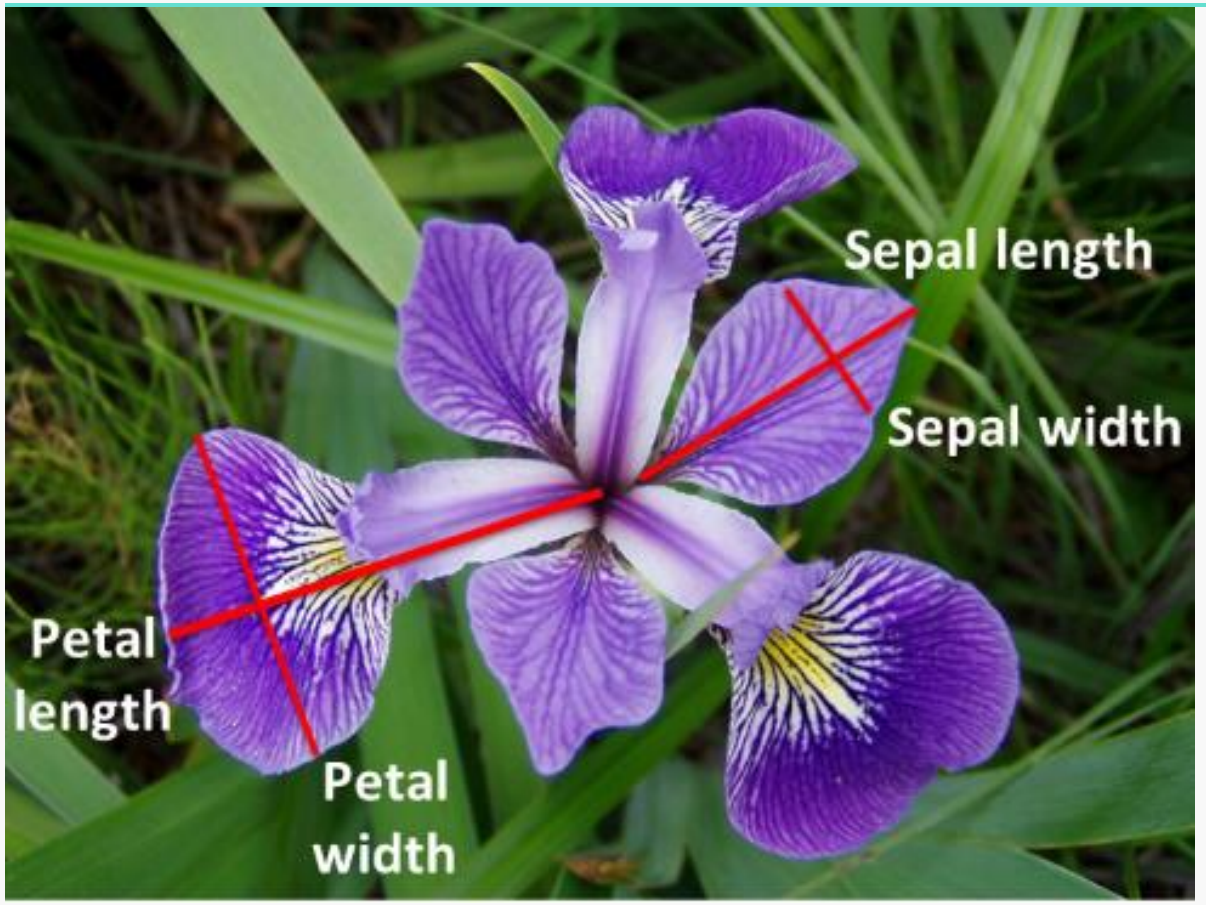
02 붓꽃 품종을 예측해보자

세토사

버시칼라

버지니카

02 붓꽃 품종을 예측해보자



X(0) = Sepal length(꽃받침 너비)
X(1) = Sepal width (꽃받침 길이)
X(2) = Petal length (꽃잎 너비)
X(3) = Petal width (꽃잎 길이)

03 코드설명

```
In [1]: import sklearn
```

```
In [2]: from sklearn.datasets import load_iris
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.model_selection import train_test_split
```

```
In [3]: import pandas as pd

        iris = load_iris()
        iris_data = iris.data
        iris_label = iris.target
        print('iris target값:', iris_label)
        print('iris target명:', iris.target_names)

        iris_df = pd.DataFrame(data=iris_data, columns=iris.feature_names)
        iris_df['label'] = iris.target
        iris_df.head(3)
```

데이터 로드

```
iris target값: [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 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]
iris target명: ['setosa' 'versicolor' 'virginica']
```

Out [3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0

```
In [4]: X_train, X_test, y_train, y_test = train_test_split(iris_data, iris_label,
                                                            test_size = 0.2, random_state=10)
```

학습데이터 와 테스트
데이터 분류

```
In [5]: dt_clf = DecisionTreeClassifier(random_state=11)

        dt_clf.fit(X_train, y_train)

Out[5]: DecisionTreeClassifier(random_state=11)
```

학습

```
In [6]: pred = dt_clf.predict(X_test)
```

```
In [7]: pred

Out[7]: array([1, 2, 0, 1, 0, 1, 2, 1, 0, 1, 1, 2, 1, 0, 0, 2, 1, 0, 0, 0, 2, 2,
               2, 0, 1, 0, 1, 1, 1, 2])
```

예측수행

```
In [8]: from sklearn.metrics import accuracy_score
        print('예측 정확도: {0:.4f}' .format(accuracy_score(y_test,pred)))

예측 정확도: 0.9667
```

예측 정확도 확인

04 시각화를 해보자

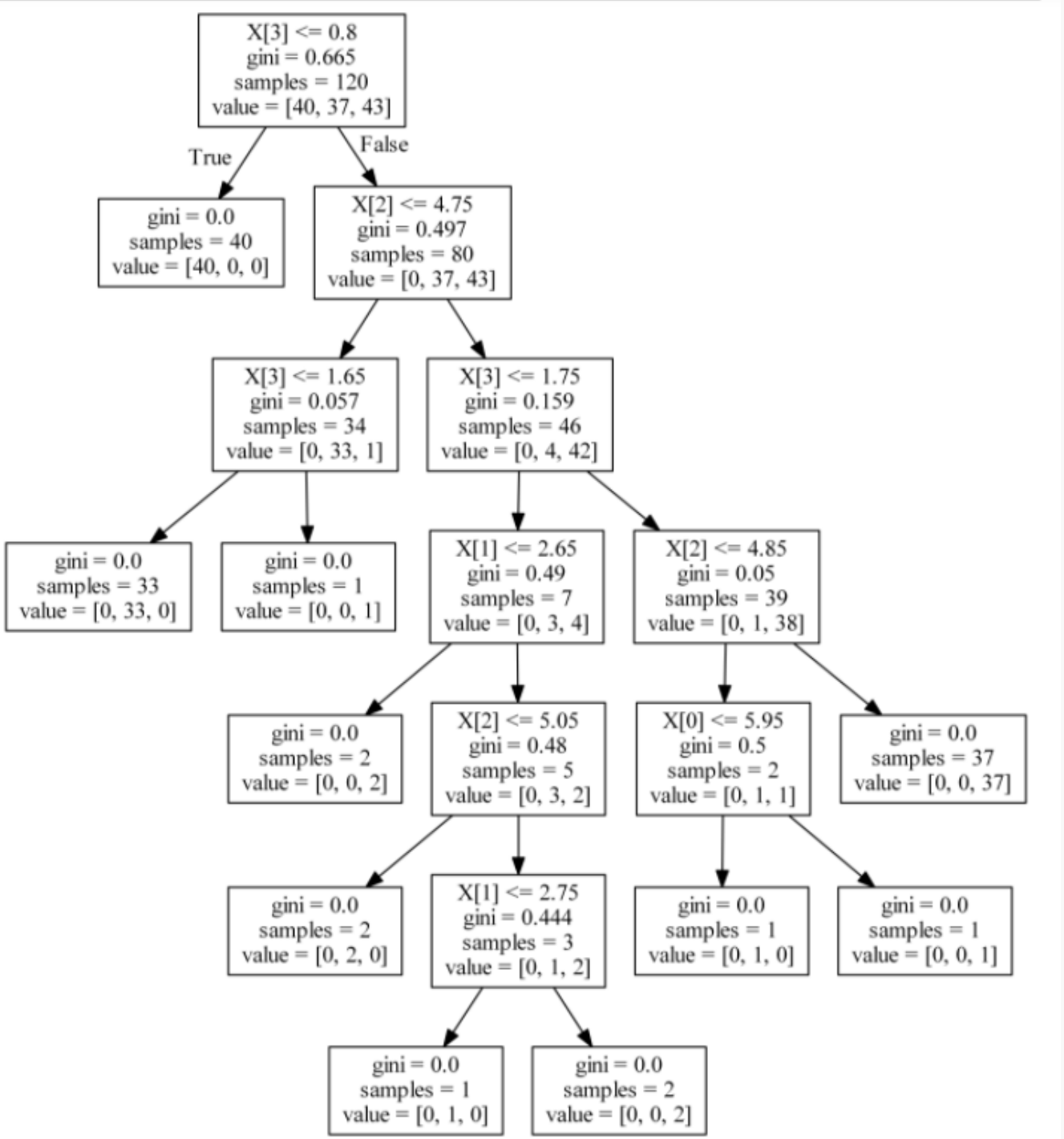
In [15]:

```
from sklearn.tree import export_graphviz
export_graphviz(dt_clf, out_file='tree.dot')


from subprocess import call
call(['dot','-Tpng', 'tree.dot', '-o', 'decistion-tree.png', '-Gdpi=600'])

from IPython.display import Image
Image(filename = 'decistion-tree.png')
```

Out[11]:


X[3] <= 0.8
gini = 0.665
samples = 120
value = [40, 37, 43]
True
False
gini = 0.0
samples = 40
value = [40, 0, 0]
X[2] <= 4.75
gini = 0.497
samples = 80
value = [0, 37, 43]
X[3] <= 1.65
gini = 0.057
samples = 34
value = [0, 33, 1]
X[3] <= 1.75
gini = 0.159
samples = 46
value = [0, 4, 42]
gini = 0.0
samples = 33
value = [0, 33, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
X[1] <= 2.65
gini = 0.49
samples = 7
value = [0, 3, 4]
X[2] <= 4.85
gini = 0.05
samples = 39
value = [0, 1, 38]
gini = 0.0
samples = 2
value = [0, 0, 2]
X[2] <= 5.05
gini = 0.48
samples = 5
value = [0, 3, 2]
X[0] <= 5.95
gini = 0.5
samples = 2
value = [0, 1, 1]
gini = 0.0
samples = 37
value = [0, 0, 37]
gini = 0.0
samples = 2
value = [0, 2, 0]
X[1] <= 2.75
gini = 0.444
samples = 3
value = [0, 1, 2]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 2
value = [0, 0, 2]

Q & A

Q&A

THANK YOU –

감 사 합 니 다.