예제 데이터

IRIS 데이터

```
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

# Load data and store it into pandas DataFrame objects
iris=load_iris()
X = pd.DataFrame(iris.data[:, :], columns = iris.feature_names[:])
y = pd.DataFrame(iris.target, columns =["Species"])
```

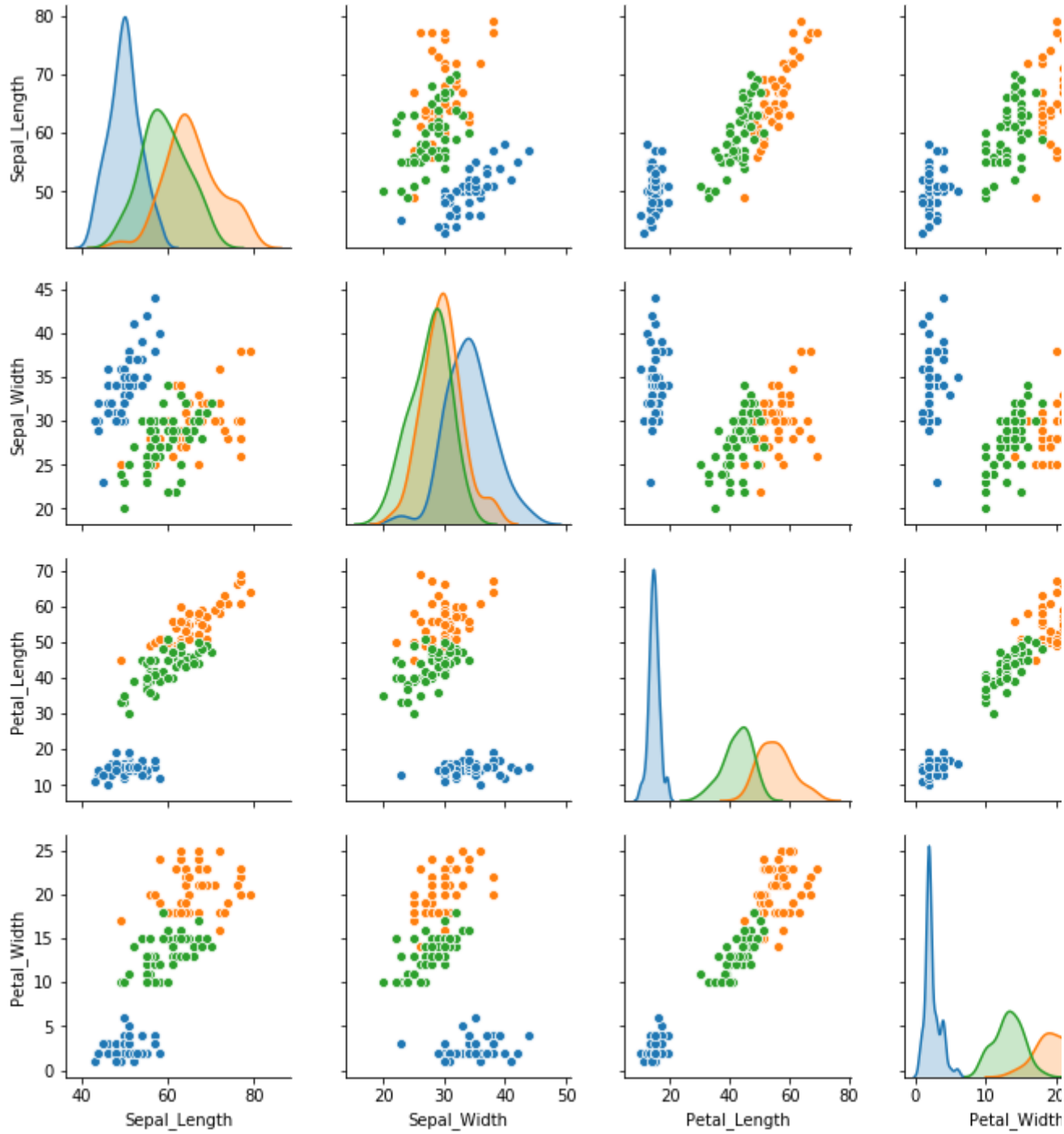
```
import pandas as pd
df=pd.read_csv('http://203.247.53.31/Stat_Notes/example_data/iris.csv')
df.count()
```

```
Sepal_Length    150
Sepal_Width     150
Petal_Length    150
Petal_Width     150
group           150
dtype: int64
```

집단별 판별변수 산점도

```
import seaborn as sns
sns.pairplot(df, hue='group')
```

```
<seaborn.axisgrid.PairGrid at 0x7f680b3f2080>
```



데이터 검증 : 결측치여부, 변수정보

```
1 df.isnull().any()
```

```
Sepal_Length    False
Sepal_Width     False
Petal_Length    False
Petal_Width     False
group           False
dtype: bool
```

```
1 df.dtypes
```

```
Sepal_Length     int64
Sepal_Width      int64
Petal_Length     int64
Petal_Width      int64
group           object
dtype: object
```

판별변수, 목표집단변수 설정

```
X=df.iloc[:,0:4]
y=df.iloc[:,4]
```

방법1

```
from sklearn import tree
clf = tree.DecisionTreeClassifier(criterion='entropy', max_depth=2,min_samples_leaf=5)
clf = clf.fit(X,y)
```

```
from sklearn import metrics
def measure_performance(X,y,clf, show_accuracy=True, show_classification_report=True, show_cor
    y_pred=clf.predict(X)
    if show_accuracy:
        print ("Accuracy:{0:.3f}".format(metrics.accuracy_score(y,y_pred)),"\n")

    if show_classification_report:
        print ("Classification report")
        print (metrics.classification_report(y,y_pred),"\n")

    if show_confusion_matrix:
        print ("Confusion matrix")
        print (metrics.confusion_matrix(y,y_pred),"\n")

measure_performance(X,y,clf, show_classification_report=False, show_confusion_matrix=False)
```

```
Accuracy:0.960
```

test data evaluating

```
clf_dt=tree.DecisionTreeClassifier(criterion='entropy', max_depth=3,min_samples_leaf=5)
clf_dt.fit(X_train,y_train)
measure_performance(X_test,y_test,clf_dt)
```

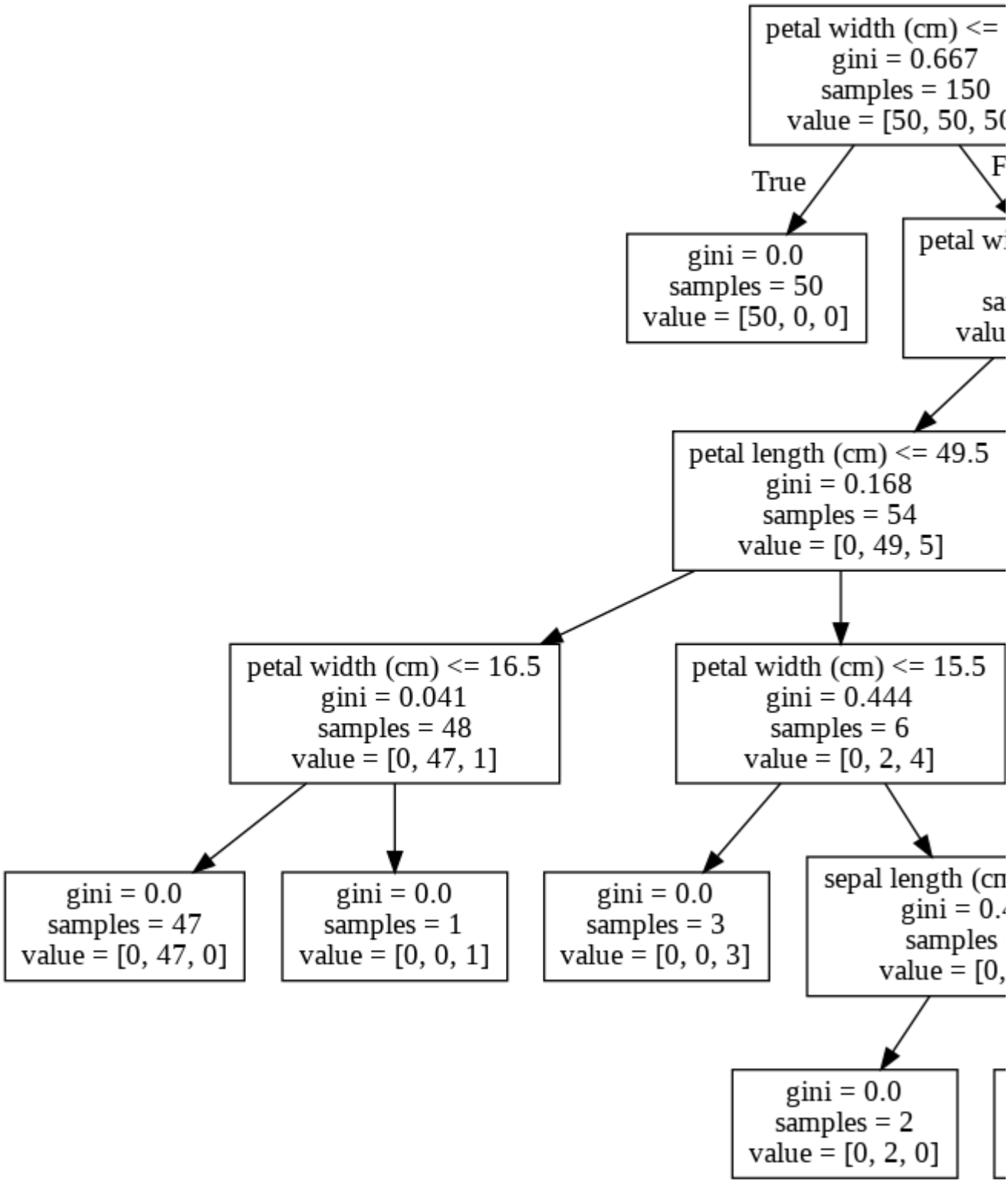
방법2

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
```

```
from pydot import graph_from_dot_data
dt = DecisionTreeClassifier()
dt.fit(X, y)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/externals/six.py:31: Deprecatio
  "(https://pypi.org/project/six/).", DeprecationWarning)
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

```
dot_data = StringIO()
export_graphviz(dt, out_file=dot_data, feature_names=iris.feature_names)
(graph, ) = graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```


petal width (cm) <=
gini = 0.667
samples = 150
value = [50, 50, 5
True
gini = 0.0
samples = 50
value = [50, 0, 0]
petal wi
sa
valu
petal length (cm) <= 49.5
gini = 0.168
samples = 54
value = [0, 49, 5]
petal width (cm) <= 16.5
gini = 0.041
samples = 48
value = [0, 47, 1]
petal width (cm) <= 15.5
gini = 0.444
samples = 6
value = [0, 2, 4]
gini = 0.0
samples = 47
value = [0, 47, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 3
value = [0, 0, 3]
sepal length (cm
gini = 0.
samples
value = [0,
gini = 0.0
samples = 2
value = [0, 2, 0]