Pandas 라이브러리 IRIS 데이터 셋 실습해보기

학습 내용

- scikit-learn를 활용한 머신러닝 모델 구축 실습

01 데이터 준비

In [20]:
```
import pandas as pd
import seaborn as sns
import numpy as np

print(pd.__version__)
iris = sns.load_dataset("iris")
iris
```

```
2.0.3
```

Out[20]:

	sepal_length	sepal_width	petal_length	petal_width	species
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

02 라이브러리 불러오기

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
```

03 모델 구축 및 예측, 평가

특징 선택 및 데이터 처리

In [22]:
```
# 데이터 입력과 출력으로 나누기
X = iris.iloc[:, 0:4]   # 1열~4열
y = iris.iloc[:, 4]     # 5열 - species
```

In [22]:
```
X.shape, y.shape
```

Out[22]:
```
((150, 4), (150,))
```

In [23]:
```
y
```

Out[23]:
```
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
```

In [24]:
```
# 데이터를 학습용과 검증용으로 분류
# test 30%, train 70% 로 분할
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.3)
```

모델 선택 및 학습

In [25]:
```
# 모델 선택 - 앙상블
model = RandomForestClassifier(max_depth=5, n_estimators=10)
model.fit(X_train, y_train)
```

Out[25]:
```
RandomForestClassifier(max_depth=5, n_estimators=10)
```

In [26]:
```
# 학습한 모델로 예측 수행
y_pred = model.predict(X_test)
print( len(y_pred) )
print( y_pred[0:10] )
```

```
45
['versicolor' 'versicolor' 'virginica' 'versicolor' 'setosa' 'versicolor'
 'setosa' 'versicolor' 'versicolor' 'setosa']
```

In [27]:
```
df_iris = pd.DataFrame(list(zip(y_pred, y_test)), columns=['pred_val', 'actual'])
df_iris.head()
```

Out[27]:

	pred_val	actual
0	versicolor	versicolor
1	versicolor	versicolor
2	virginica	virginica
3	versicolor	versicolor
4	setosa	setosa

In [28]:
```
df_iris.loc[ df_iris['pred_val'] == df_iris['actual'], 'correct' ] = 1
df_iris.head(10)
```

Out[28]:

	pred_val	actual	correct
0	versicolor	versicolor	1.0
1	versicolor	versicolor	1.0
2	virginica	virginica	1.0
3	versicolor	versicolor	1.0
4	setosa	setosa	1.0
5	versicolor	versicolor	1.0
6	setosa	setosa	1.0
7	versicolor	versicolor	1.0
8	versicolor	versicolor	1.0
9	setosa	setosa	1.0

In [29]:
```
df_iris.correct.value_counts()
```

Out[29]:
```
correct
1.0    40
Name: count, dtype: int64
```

In [30]:
```
np.mean( df_iris['correct'] )
```

Out[30]:
```
1.0
```

In [ ]:
```

```