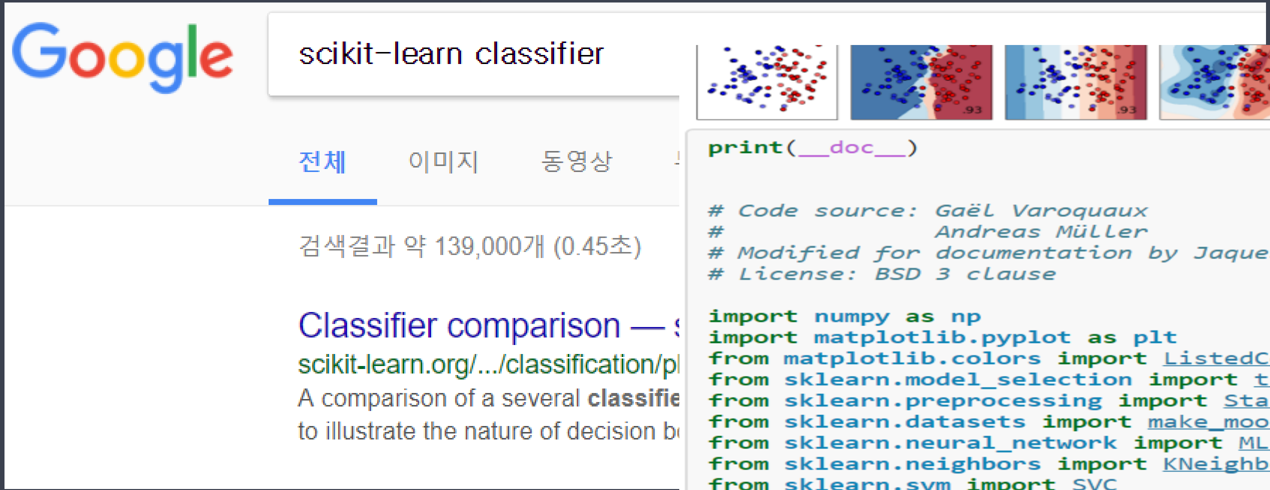
Machine Learning

Support Vecter Machine(SVM)

알고리즘 변경

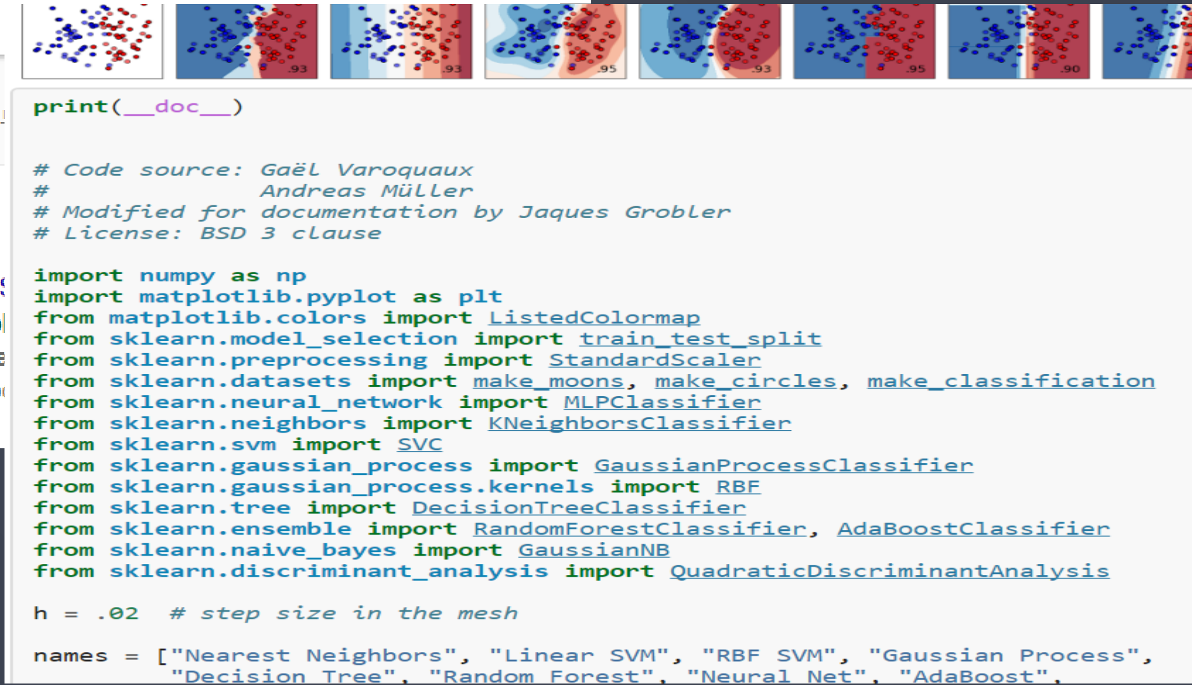


scikit-learn classifier

전체 이미지 동영상

검색결과 약 139,000개 (0.45초)

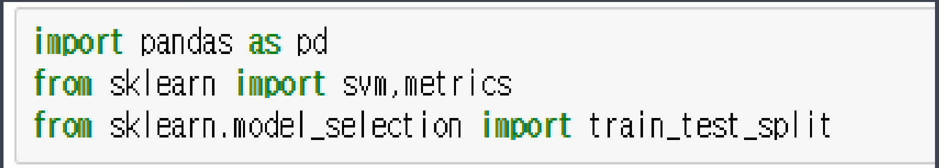
Classifier comparison — s
scikit-learn.org/.../classification/pl
A comparison of a several **classifie**
to illustrate the nature of decision b



```
print(__doc__)


# Code source: Gaël Varoquaux
#              Andreas Müller
# Modified for documentation by Jaques Grobler
# License: BSD 3 clause

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import make_moons, make_circles, make_classification
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis

h = .02  # step size in the mesh

names = ["Nearest Neighbors", "Linear SVM", "RBF SVM", "Gaussian Process",
         "Decision Tree", "Random Forest", "Neural Net", "AdaBoost",
```

Deep Learning

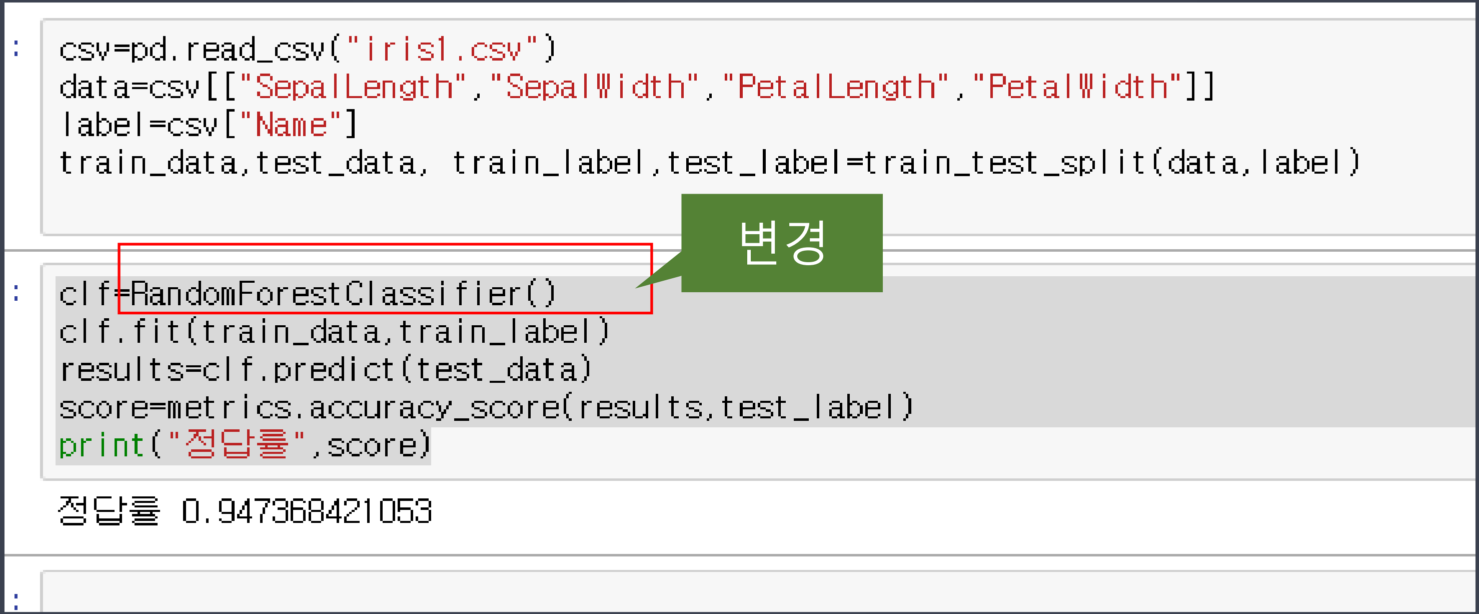
알고리즘 변경



```
import pandas as pd
from sklearn import svm,metrics
from sklearn.model_selection import train_test_split
```

from sklearn.ensemble import RandomForestClassifier

추가



```
csv=pd.read_csv("iris1.csv")
data=csv[["SepalLength","SepalWidth","PetalLength","PetalWidth"]]
label=csv["Name"]
train_data,test_data, train_label,test_label=train_test_split(data,label)
```

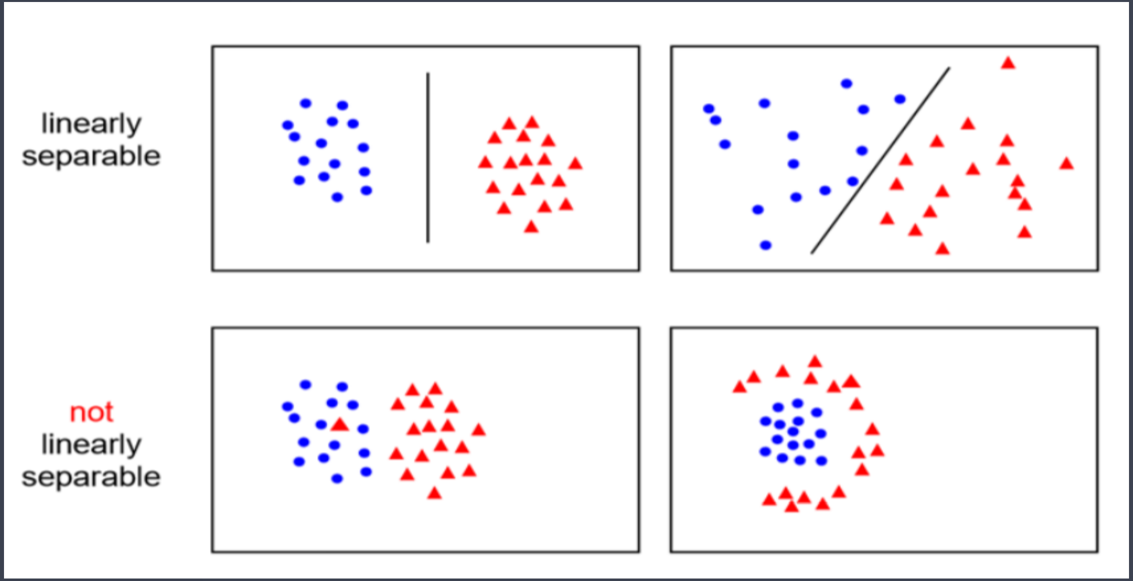
변경

```
clf=RandomForestClassifier()
clf.fit(train_data,train_label)
results=clf.predict(test_data)
score=metrics.accuracy_score(results,test_label)
print("정답률",score)
```

```
정답률 0.947368421053
```

SVM

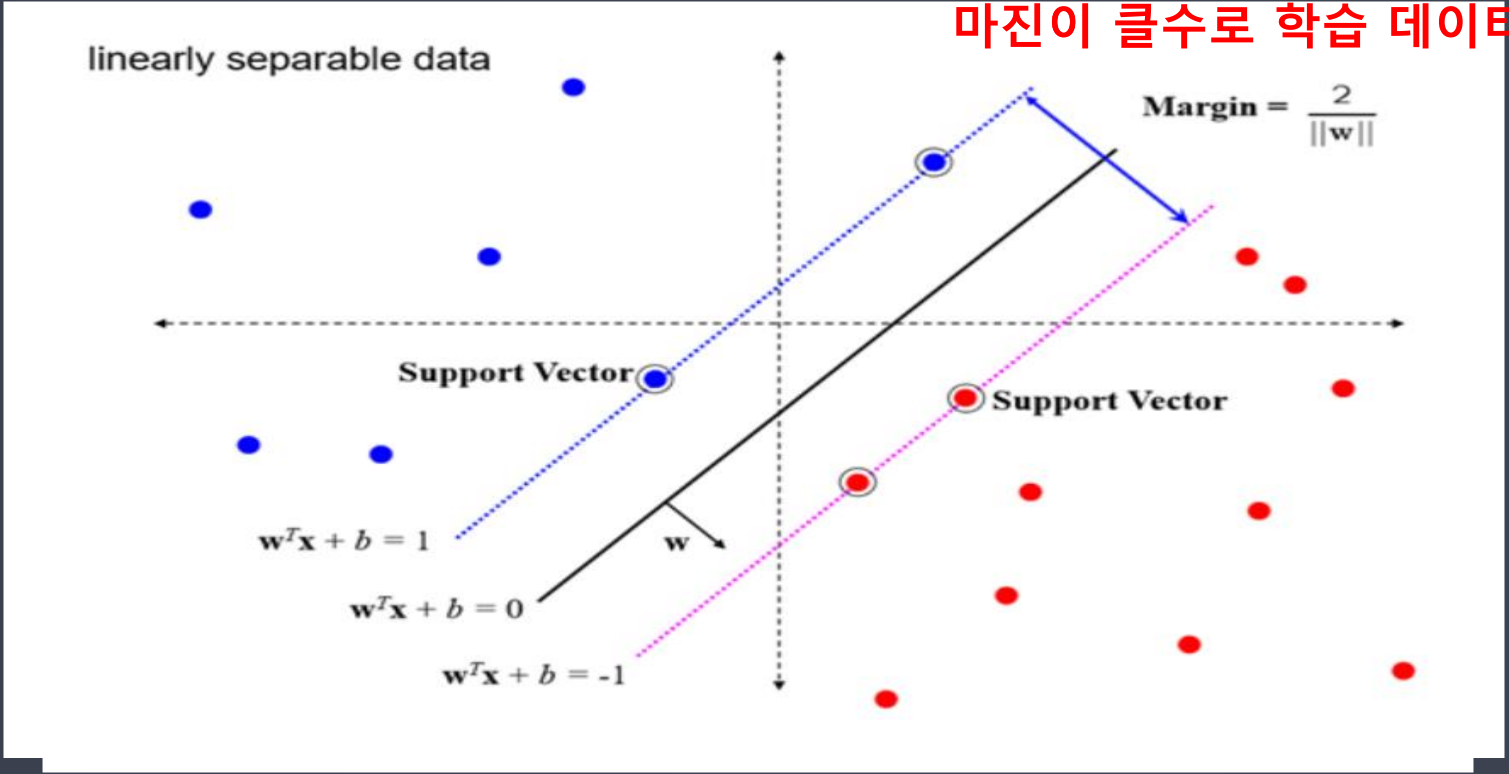
- 분류 ,회귀분석에 사용
- 지도학습 알고리즘
- hyper-plane(초평면)을 이용해 카테고리를 나눔



SVM

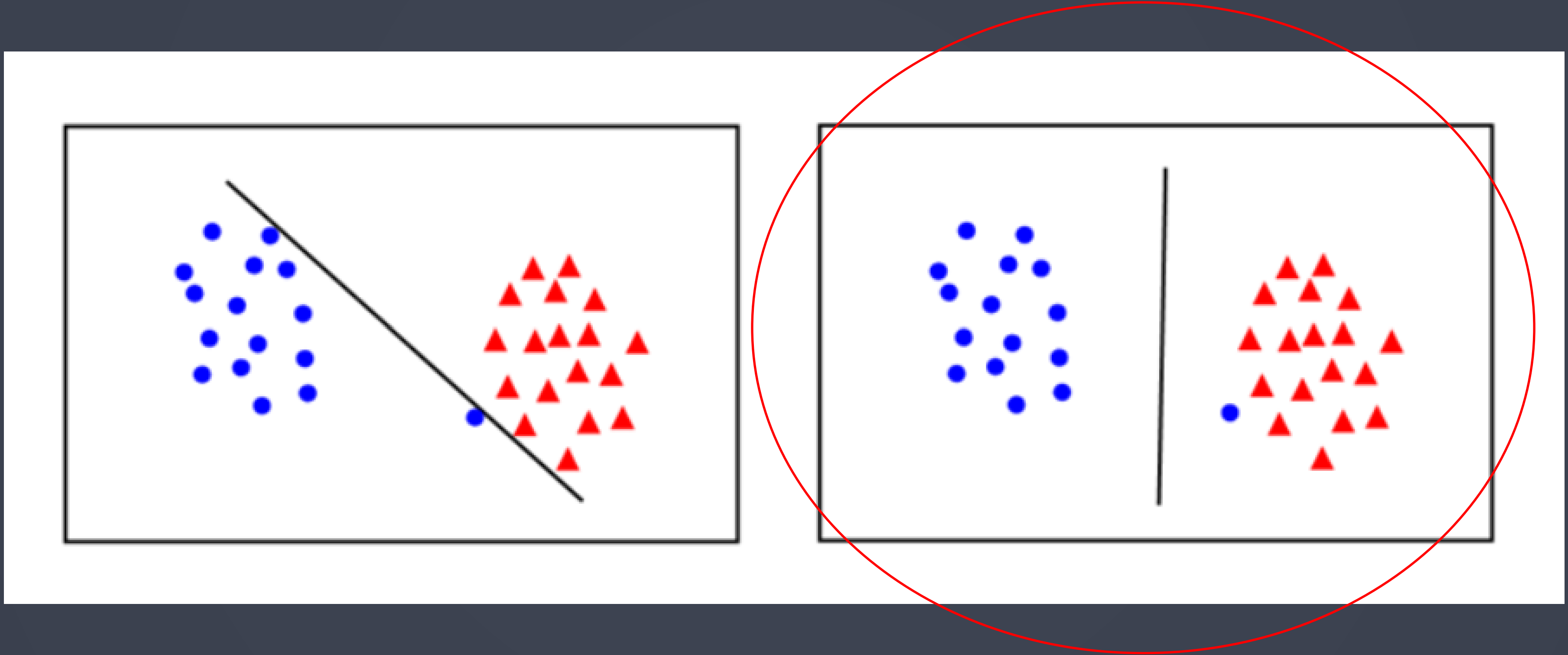
최적으로 나누는 것은 어떻게 할수 있을까 ??

마진이 클수로 학습 데이터를 잘 분류


linearly separable data
Margin = $\frac{2}{\|\mathbf{w}\|}$
Support Vector
Support Vector
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$
$\mathbf{w}$

SVM

- 어떤 경우가 최적일까 ?



XOR연산 학습하기

데이터

데이터

답[레이블]

P	Q	P xor Q
0	0	0
1	0	1
0	1	1
1	1	0

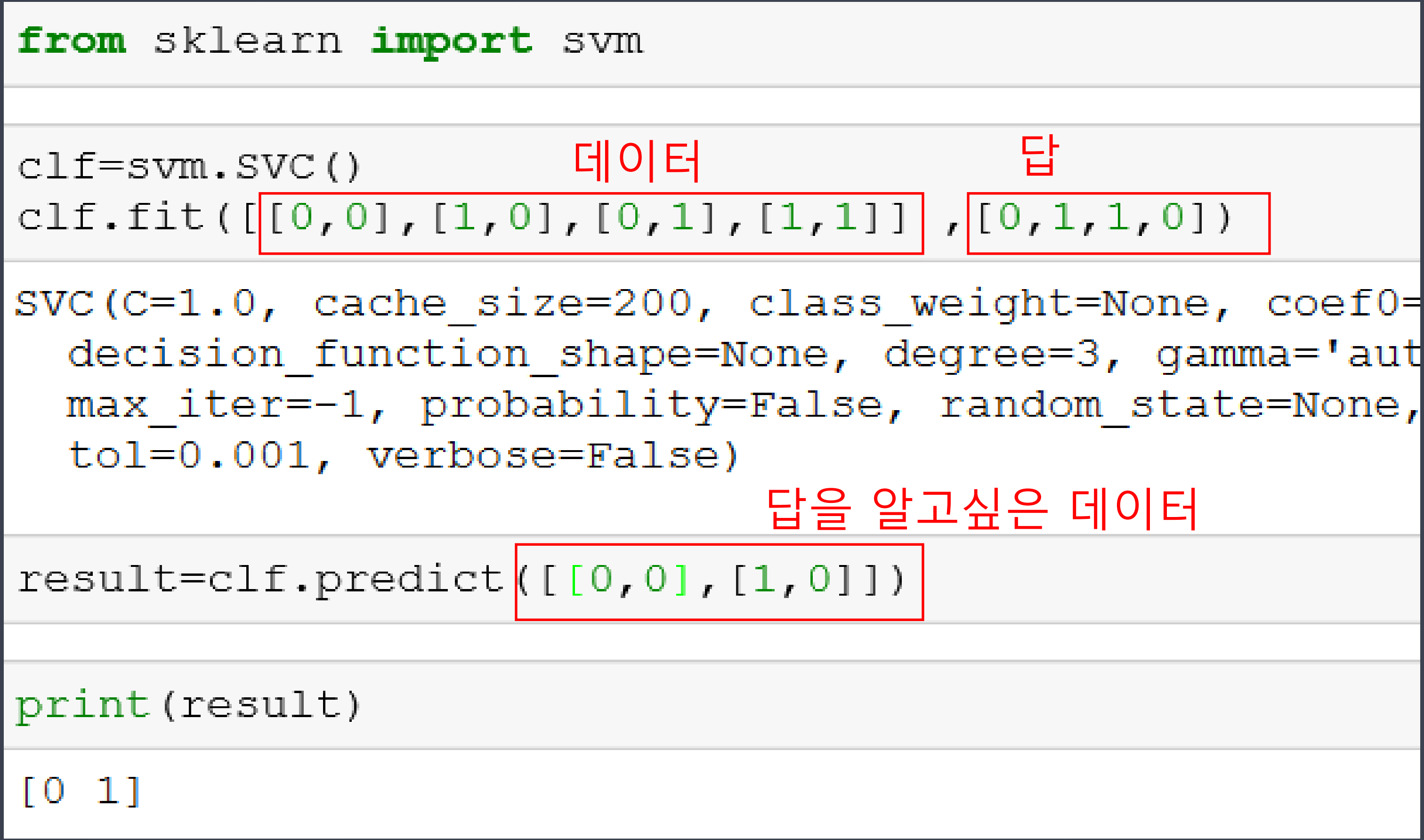
XOR연산 학습하기 1

```
clf=svm.SVC()
clf.fit(데이터 , 답)
clf.predict(값을 얻고 싶은데이터 )
score=metrics.accuracy_score(실제답, 예측결과 )
```

fit()메소드: 학습 기계에 데이터를 학습
predict()메소드 : 데이터를 넣어 학습

Deep Learning

XOR연산 학습하기 1



```
from sklearn import svm
```

```
clf=svm.SVC()
clf.fit([[0,0],[1,0],[0,1],[1,1]] ,[0,1,1,0])
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=
  decision_function_shape=None, degree=3, gamma='aut
  max_iter=-1, probability=False, random_state=None,
  tol=0.001, verbose=False)
```

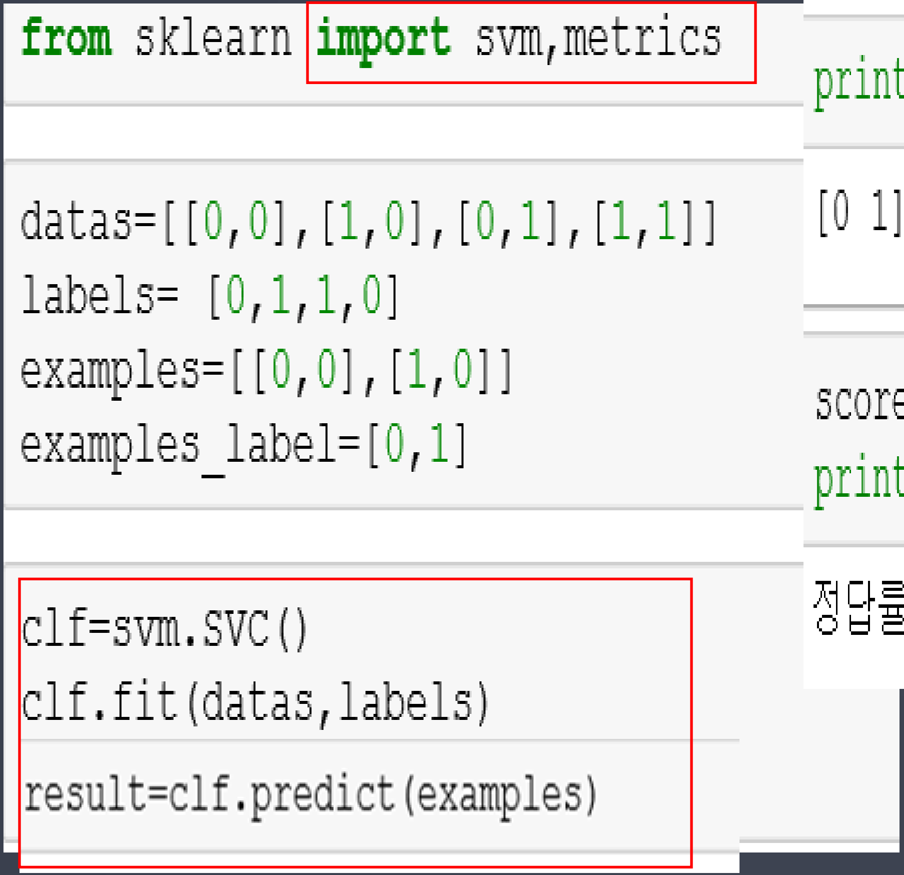
```
result=clf.predict([[0,0],[1,0]])
```

```
print(result)
```

```
[0 1]
```

```
from sklearn import model_selection, svm ,metrics


clf=svm.SVM()          #기계학습 알고리즘 선택
clf.fit()              #학습
predict=clf.predict()  #예측
                                                    # 정답률 구하기
score=metrics.accuracy_score(실제답 ,( predict )
pirnt("정답률",score)
```

XOR연산 학습하기2

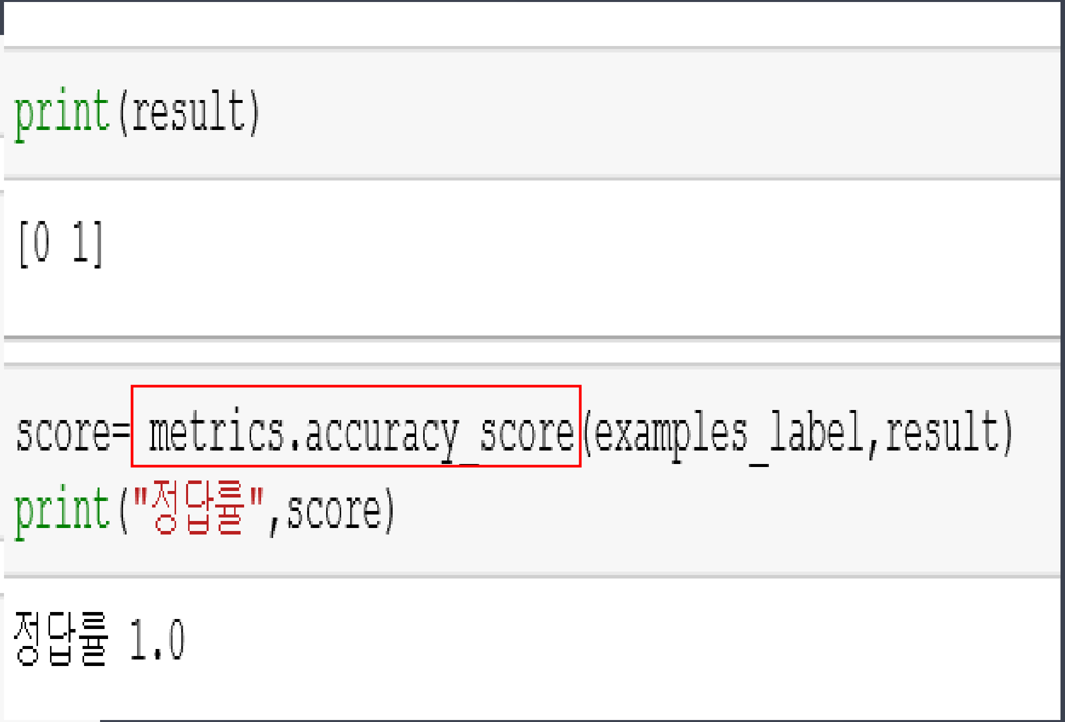


```
from sklearn import svm,metrics
```

```
datas=[[0,0],[1,0],[0,1],[1,1]]
labels= [0,1,1,0]
examples=[[0,0],[1,0]]
examples_label=[0,1]
```

```
clf=svm.SVC()
clf.fit(datas,labels)
```

```
result=clf.predict(examples)
```



```
print(result)
```

```
[0 1]
```

```
score= metrics.accuracy_score(examples_label,result)
print("정답률",score)
```

```
정답률 1.0
```

Iris Dataset

붓꽃의 품종분류

setosa, versicolor, virginica 종 분류

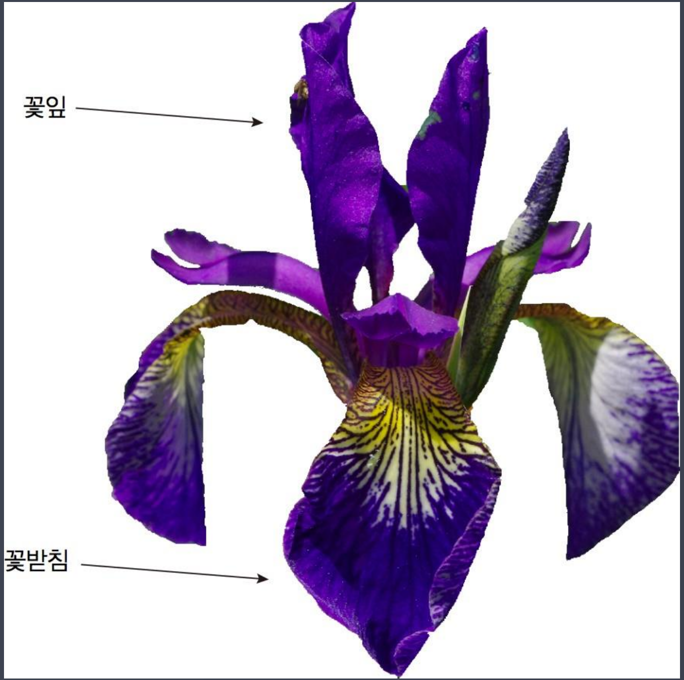
꽃잎petal, 꽃받침sepal의 폭과 길이

사전에 준비한 데이터를 이용하므로 지도 학습

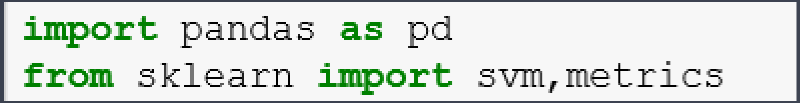
3개의 붓꽃 품종에서 고르는 분류classification

클래스class: 가능한 출력값. 즉 세개의 붓꽃 품종

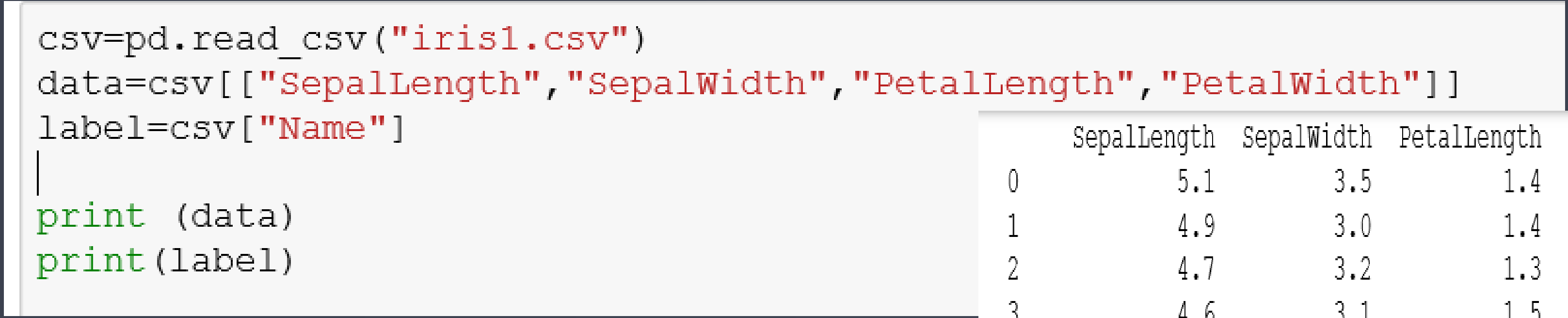
레이블label: 데이터 포인트 하나에 대한 출력



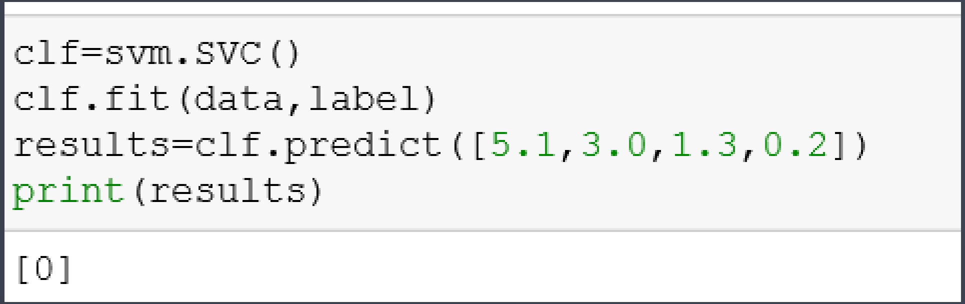
분류



```python
import pandas as pd
from sklearn import svm,metrics
```

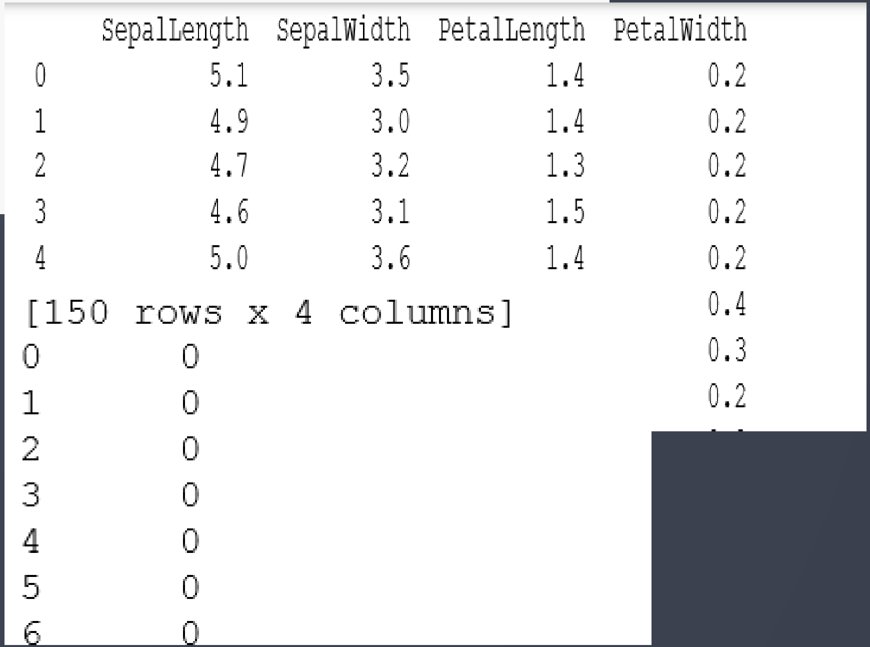


```python
csv=pd.read_csv("iris1.csv")
data=csv[["SepalLength","SepalWidth","PetalLength","PetalWidth"]]
label=csv["Name"]

print (data)
print(label)
```



```python
clf=svm.SVC()
clf.fit(data,label)
results=clf.predict([5.1,3.0,1.3,0.2])
print(results)
```

```
[0]
```

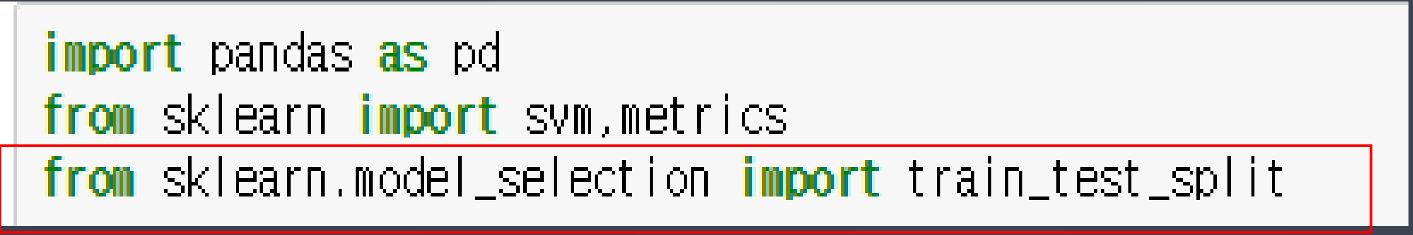


```
   SepalLength  SepalWidth  PetalLength  PetalWidth
0          5.1         3.5          1.4         0.2
1          4.9         3.0          1.4         0.2
2          4.7         3.2          1.3         0.2
3          4.6         3.1          1.5         0.2
4          5.0         3.6          1.4         0.2
                                                0.4
                                                0.3
                                                0.2
```

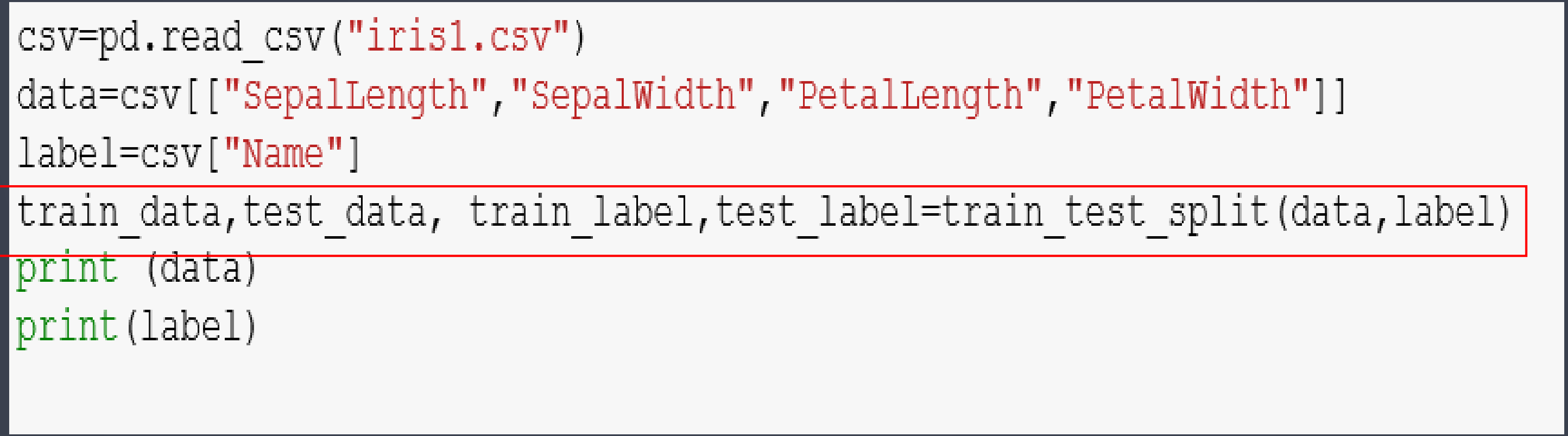
```
[150 rows x 4 columns]
0      0
1      0
2      0
3      0
4      0
5      0
6      0
```

훈련 데이터와 테스트데이터 분할하는 메소드

train_test_split()메소드 사용

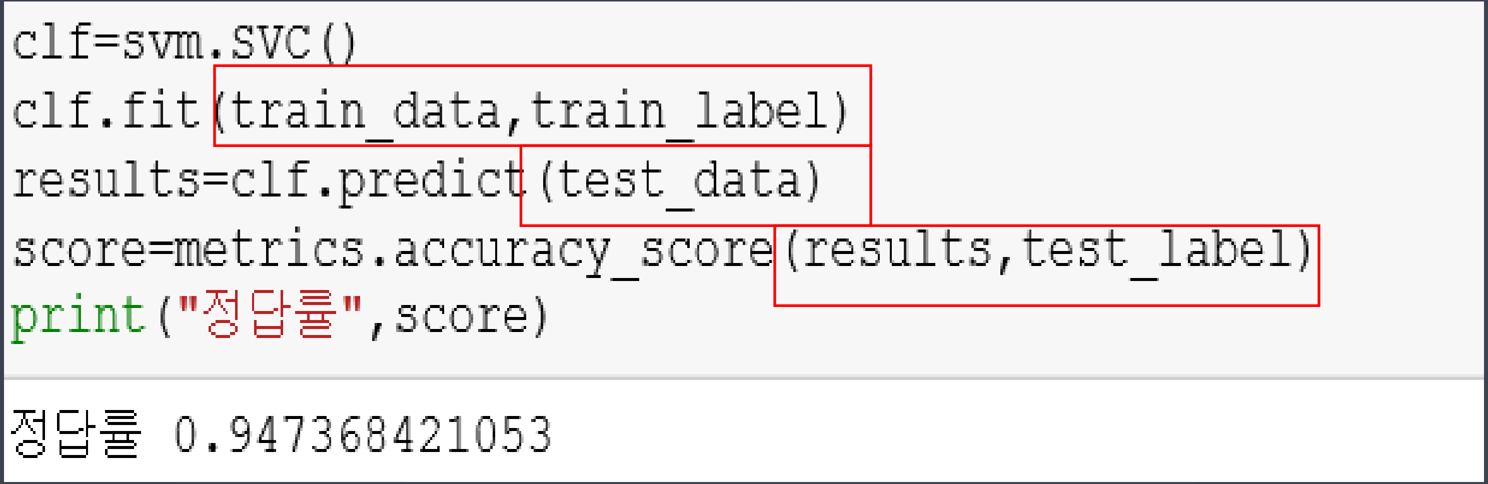


```python
import pandas as pd
from sklearn import svm,metrics
from sklearn.model_selection import train_test_split
```



```python
csv=pd.read_csv("iris1.csv")
data=csv[["SepalLength","SepalWidth","PetalLength","PetalWidth"]]
label=csv["Name"]
train_data,test_data, train_label,test_label=train_test_split(data,label)
print (data)
print(label)
```

평가



```
clf=svm.SVC()
clf.fit(train_data,train_label)
results=clf.predict(test_data)
score=metrics.accuracy_score(results,test_label)
print("정답률",score)
```

```
정답률 0.947368421053
```