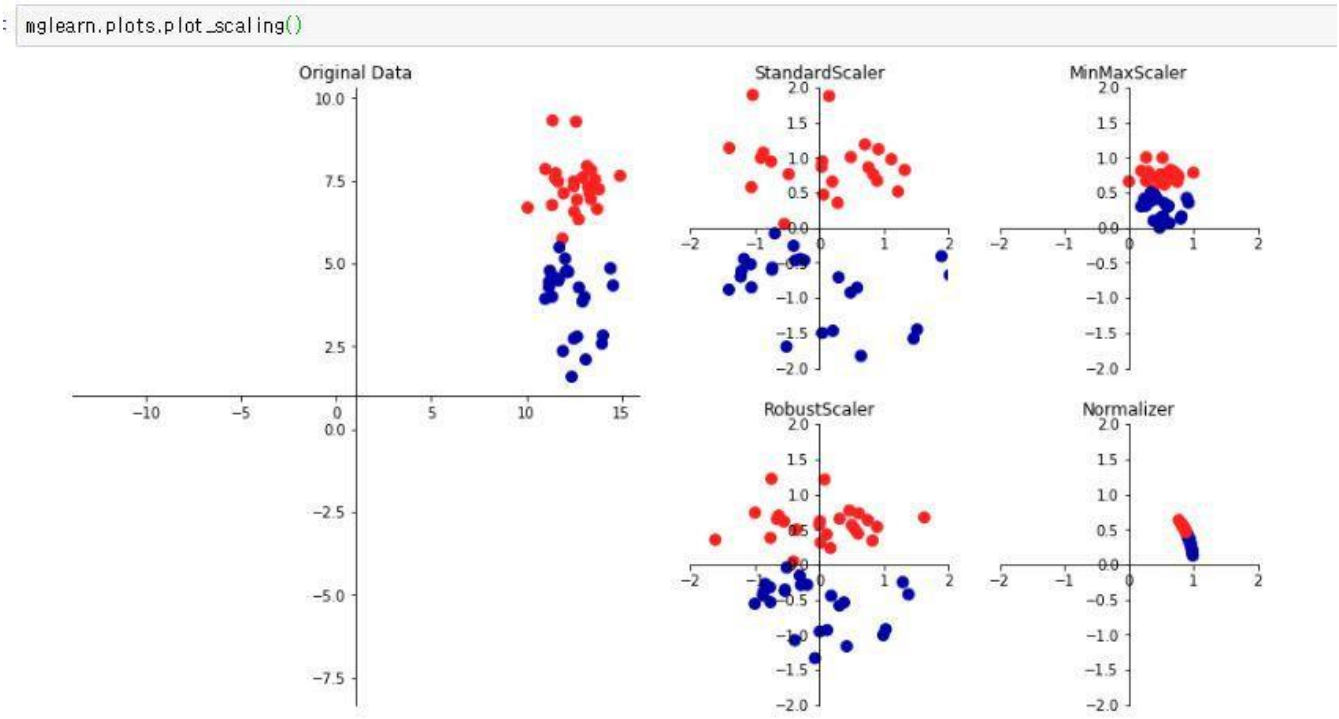
데이터 스케일링 (Data Scaling)

데이터 스케일링이란 데이터 전처리 과정의 하나입니다.

데이터 스케일링을 해주는 이유는 데이터의 값이 너무 크거나 혹은 작은 경우에 모델 알고리즘 학습과정에서 0 으로 수렴하거나 무한으로 발산해버릴 수 있기 때문입니다.

따라서, scaling 은 데이터 전처리 과정에서 굉장히 중요한 과정입니다.
가볍게 살펴보도록 하겠습니다.

1. What is Scaler?



(1) StandardScaler
각 feature 의 평균을 0, 분산을 1 로 변경합니다. 모든 특성들이 같은 스케일을 갖게 됩니다.

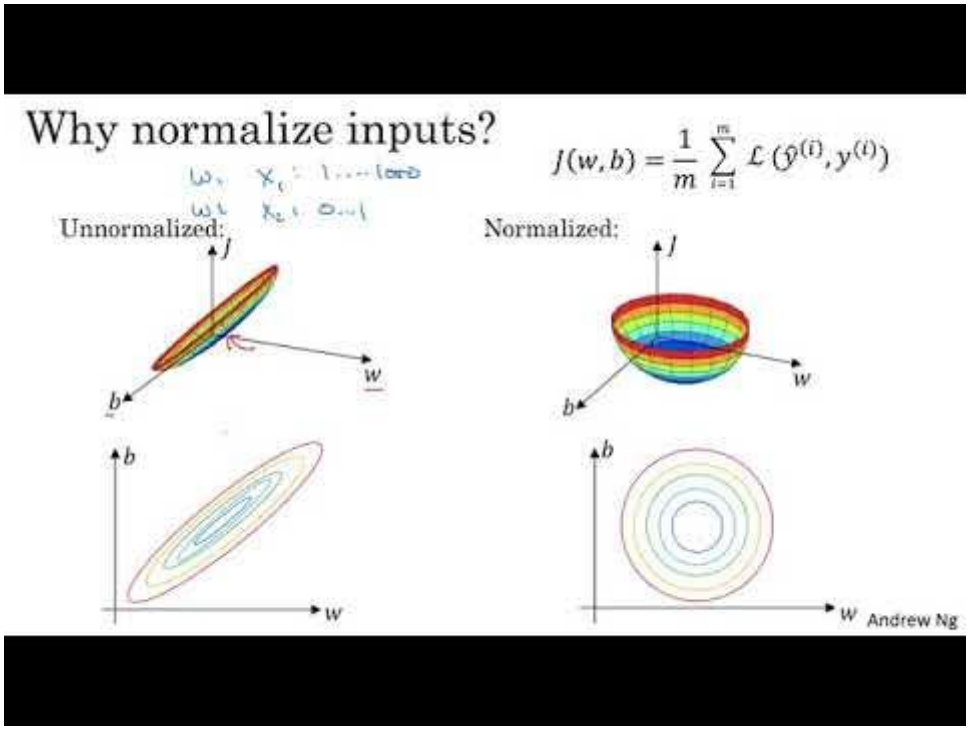
(2) RobustScaler
모든 특성들이 같은 크기를 갖는다는 점에서 StandardScaler 와 비슷하지만,
평균과 분산 대신 median 과 quartile 을 사용합니다.
RobustScaler 는 이상치에 영향을 받지 않습니다.

(3) MinMaxScaler
모든 feature 가 0 과 1 사이에 위치하게 만듭니다.
데이터가 2 차원 셋일 경우,
모든 데이터는 x 축의 0 과 1 사이에, y 축의 0 과 1 사이에 위치하게 됩니다.

(4) Normalizer

StandardScaler, RobustScaler, MinMaxScaler 가 각 columns 의 통계치를 이용한다면
Normalizer 는 row 마다 각각 정규화됩니다.

Normalizer 는 유클리드 거리가 1 이 되도록 데이터를 조정합니다.
(유클리드 거리는 두 점 사이의 거리를 계산할 때 쓰는 방법, L2 Distance)



(출처 : Andrew NG lecture)

Normalize 를 하게 되면
Spherical contour(구형 윤곽)을 갖게 되는데,
이렇게 하면 좀 더 빠르게 학습할 수 있고 과대적합 확률을 낮출 수 있습니다.

2. Code

scikit-learn 에 있는 유방암 데이터셋으로 데이터 스케일링을 해보겠습니다.

데이터를 학습용과 테스트용으로 분할했습니다.

scaler 를 사용하기 이전에 주의 해야될 점을 먼저 살펴보겠습니다.
scaler 는 fit 과 transform 메서드를 지니고 있습니다.

fit 메서드로 데이터 변환을 학습하고,
transform 메서드로 실제 데이터의 스케일을 조정합니다.

이때, fit 메서드는 학습용 데이터에만 적용해야 합니다.
그 후, transform 메서드를 학습용 데이터와 테스트 데이터에 적용합니다.

scaler 는 fit_transform()이란 단축 메서드를 제공합니다.
학습용 데이터에는 fit_transform()메서드를 적용하고,
테스트 데이터에는 transform()메서드를 적용합니다.

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split

cancer = load_breast_cancer()

X_train, X_test, Y_train, Y_test = train_test_split(cancer.data,
                                                    cancer.target,
                                                    stratify=cancer.target,
                                                    random_state=2019)
```

(1) StandardScaler code

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_scale = scaler.fit_transform(X_train)

print('스케일 조정 전 features MIN value : \n {}'.format(X_train.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train.max(axis=0)))
print('스케일 조정 전 features MIN value : \n {}'.format(X_train_scale.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train_scale.max(axis=0)))
```

```
스케일 조정 전 features MIN value :
 [7.691e+00 9.710e+00 4.792e+01 1.704e+02 5.263e-02 1.938e-02 0.000e+00
 0.000e+00 1.060e-01 4.996e-02 1.115e-01 3.602e-01 7.570e-01 6.802e+00
 1.713e-03 2.252e-03 0.000e+00 0.000e+00 7.882e-03 8.948e-04 8.678e+00
 1.202e+01 5.449e+01 2.236e+02 7.117e-02 2.729e-02 0.000e+00 0.000e+00
 1.565e-01 5.504e-02]
스케일 조정 전 features MAX value :
 [2.811e+01 3.928e+01 1.885e+02 2.501e+03 1.398e-01 3.454e-01 4.264e-01
 1.878e-01 3.040e-01 9.575e-02 2.873e+00 4.885e+00 2.198e+01 5.422e+02
 3.113e-02 1.064e-01 3.960e-01 5.279e-02 7.895e-02 2.984e-02 3.604e+01
 4.954e+01 2.512e+02 4.254e+03 2.184e-01 1.058e+00 1.105e+00 2.910e-01
 5.558e-01 2.075e-01]
스케일 조정 전 features MIN value :
 [-1.82649679 -2.2589088  -1.81061958 -1.3691575  -3.215235   -1.62335942
 -1.12197837 -1.25606443 -2.78510532 -1.86692203 -1.04672829 -1.56117279
 -1.03399046 -0.70924554 -1.7826557  -1.34782746 -1.08275944 -1.9538913
 -1.63235399 -1.04695123 -1.58231847 -2.28985773 -1.58015783 -1.16210312
 -2.68079139 -1.4699755  -1.33593337 -1.74189712 -2.26454054 -1.61168047]
스케일 조정 전 features MAX value :
 [ 3.92198171  4.66212461  3.92107226  5.14879805  3.26368239  4.59780886
  4.19055919  3.51431177  4.63124424  4.88303938  8.81099111  6.83901816
  9.36282817 10.57726999  8.2120306   4.71802057 12.3424089   6.81172053
  7.95051055  9.39297199  4.08450004  3.98653819  4.27044371  5.92712333
  3.81411327  5.1687493   4.02300871  2.62476297  4.6334501   6.88134572]
```

(2) RobustScaler code

```
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

X_train_scale = scaler.fit_transform(X_train)

print('스케일 조정 전 features MIN value : \n {}'.format(X_train.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train.max(axis=0)))
print('스케일 조정 전 features MIN value : \n {}'.format(X_train_scale.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train_scale.max(axis=0)))
```

```
스케일 조정 전 features MIN value :
 [7.691e+00 9.710e+00 4.792e+01 1.704e+02 5.263e-02 1.938e-02 0.000e+00
 0.000e+00 1.060e-01 4.996e-02 1.115e-01 3.602e-01 7.570e-01 6.802e+00
 1.713e-03 2.252e-03 0.000e+00 0.000e+00 7.882e-03 8.948e-04 8.678e+00
 1.202e+01 5.449e+01 2.236e+02 7.117e-02 2.729e-02 0.000e+00 0.000e+00
 1.565e-01 5.504e-02]
스케일 조정 전 features MAX value :
 [2.811e+01 3.928e+01 1.885e+02 2.501e+03 1.398e-01 3.454e-01 4.264e-01
 1.878e-01 3.040e-01 9.575e-02 2.873e+00 4.885e+00 2.198e+01 5.422e+02
 3.113e-02 1.064e-01 3.960e-01 5.279e-02 7.895e-02 2.984e-02 3.604e+01
 4.954e+01 2.512e+02 4.254e+03 2.184e-01 1.058e+00 1.105e+00 2.910e-01
 5.558e-01 2.075e-01]
스케일 조정 전 features MIN value :
 [-1.33658537 -1.69251825 -1.32644768 -1.03759197 -2.41103604 -1.11331811
 -0.62718806 -0.60975498 -2.22813688 -1.35235294 -0.88778365 -1.15241464
 -0.93436645 -0.6881635  -1.6023605  -0.98593272 -0.9442091  -1.54518441
 -1.35092416 -1.0390557  -1.1166301  -1.63349515 -1.06289193 -0.86029044
 -2.10980392 -0.98094992 -0.88124594 -1.0379344  -1.83911439 -1.20344456]
스케일 조정 전 features MAX value :
 [ 3.40650407  3.70346715  3.43413478  5.19812709  2.49718468  3.84893455
  3.5309864   2.83059308  3.79467681  4.03470588 10.55766242  5.80667487
 11.96323306 19.57479421  8.60921635  4.55311794 13.33249211  5.83674183
  7.70809433 12.22942929  3.68582712  2.91990291  3.72761035  6.47871808
  3.13903743  4.34021683  3.21627885  1.94959191  4.05461255  6.09041981]
```

(3) MinMaxScaler code

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_train_scale = scaler.fit_transform(X_train)

print('스케일 조정 전 features MIN value : \n {}'.format(X_train.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train.max(axis=0)))
print('스케일 조정 전 features MIN value : \n {}'.format(X_train_scale.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train_scale.max(axis=0)))
```

```
스케일 조정 전 features MIN value :
 [7.691e+00 9.710e+00 4.792e+01 1.704e+02 5.263e-02 1.938e-02 0.000e+00
 0.000e+00 1.060e-01 4.996e-02 1.115e-01 3.602e-01 7.570e-01 6.802e+00
 1.713e-03 2.252e-03 0.000e+00 0.000e+00 7.882e-03 8.948e-04 8.678e+00
 1.202e+01 5.449e+01 2.236e+02 7.117e-02 2.729e-02 0.000e+00 0.000e+00
 1.565e-01 5.504e-02]
스케일 조정 전 features MAX value :
 [2.811e+01 3.928e+01 1.885e+02 2.501e+03 1.398e-01 3.454e-01 4.264e-01
 1.878e-01 3.040e-01 9.575e-02 2.873e+00 4.885e+00 2.198e+01 5.422e+02
 3.113e-02 1.064e-01 3.960e-01 5.279e-02 7.895e-02 2.984e-02 3.604e+01
 4.954e+01 2.512e+02 4.254e+03 2.184e-01 1.058e+00 1.105e+00 2.910e-01
 5.558e-01 2.075e-01]
스케일 조정 전 features MIN value :
 [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.]
스케일 조정 전 features MAX value :
 [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.]
```

(4) Normalizer code

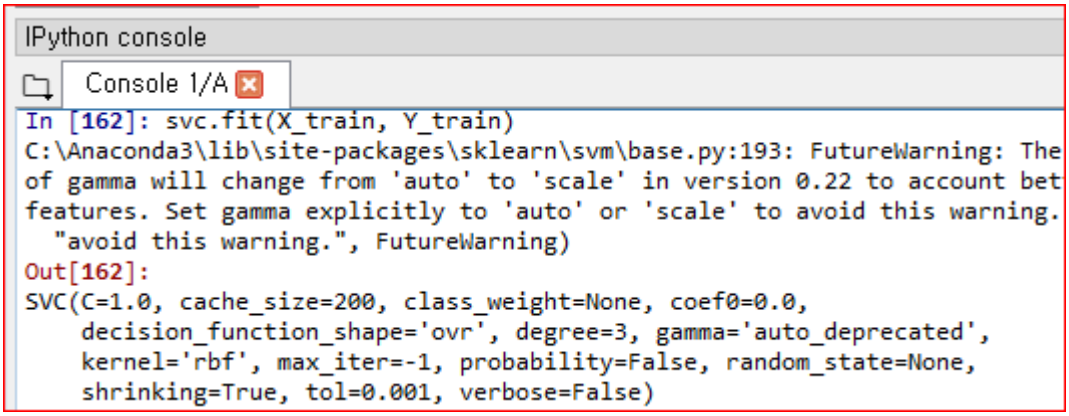
```
from sklearn.preprocessing import Normalizer

scaler = Normalizer ()

X_train_scale = scaler.fit_transform(X_train)

print('스케일 조정 전 features MIN value : \n {}'.format(X_train.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train.max(axis=0)))
print('스케일 조정 전 features MIN value : \n {}'.format(X_train_scale.min(axis=0)))
print('스케일 조정 전 features MAX value : \n {}'.format(X_train_scale.max(axis=0)))
```

```
스케일 조정 전 features MIN value :
 [7.691e+00 9.710e+00 4.792e+01 1.704e+02 5.263e-02 1.938e-02 0.000e+00
 0.000e+00 1.060e-01 4.996e-02 1.115e-01 3.602e-01 7.570e-01 6.802e+00
 1.713e-03 2.252e-03 0.000e+00 0.000e+00 7.882e-03 8.948e-04 8.678e+00
 1.202e+01 5.449e+01 2.236e+02 7.117e-02 2.729e-02 0.000e+00 0.000e+00
 1.565e-01 5.504e-02]
스케일 조정 전 features MAX value :
 [2.811e+01 3.928e+01 1.885e+02 2.501e+03 1.398e-01 3.454e-01 4.264e-01
 1.878e-01 3.040e-01 9.575e-02 2.873e+00 4.885e+00 2.198e+01 5.422e+02
 3.113e-02 1.064e-01 3.960e-01 5.279e-02 7.895e-02 2.984e-02 3.604e+01
 4.954e+01 2.512e+02 4.254e+03 2.184e-01 1.058e+00 1.105e+00 2.910e-01
 5.558e-01 2.075e-01]
스케일 조정 전 features MIN value :
 [5.51189319e-03 4.57286305e-03 3.75701254e-02 4.39372749e-01
 2.17902707e-05 2.37805701e-05 0.00000000e+00 0.00000000e+00
 4.14296567e-05 1.13032004e-05 1.38230416e-04 2.28118010e-04
 8.95949678e-04 1.14342671e-02 1.17600717e-06 2.76335624e-06
 0.00000000e+00 0.00000000e+00 3.33747542e-06 5.12412074e-07
 7.24466195e-03 5.15445862e-03 5.04955350e-02 6.96047105e-01
 2.72780418e-05 4.21129601e-05 0.00000000e+00 0.00000000e+00
 4.59910547e-05 1.49295517e-05]
스케일 조정 전 features MAX value :
 [2.61835034e-02 8.66088059e-02 1.64570349e-01 6.97400762e-01
 3.41167747e-04 5.03168358e-04 8.25977854e-04 1.60675994e-04
 6.93483246e-04 2.63877694e-04 1.65778661e-03 1.13783731e-02
 1.07454414e-02 1.46680208e-01 5.26665970e-05 2.42624716e-04
 7.96220132e-04 1.06142578e-04 9.35810942e-05 5.99980019e-05
 2.95436799e-02 1.08567406e-01 1.85507619e-01 8.93239684e-01
 5.43347698e-04 1.47441480e-03 1.65195571e-03 3.96782489e-04
 9.93708656e-04 4.44506641e-04]
```

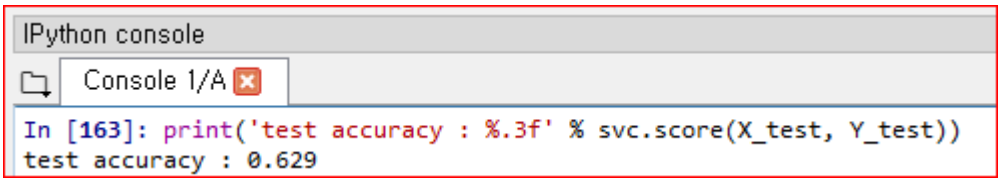
3. 적용해보기

SVC 로 cancer 데이터셋을 학습해보겠습니다.
먼저, 데이터 스케일링을 적용하지 않은 채 진행하겠습니다.

```
from sklearn.svm import SVC

X_train, X_test, Y_train, Y_test = train_test_split(cancer.data,
                                                    cancer.target,
                                                    random_state=0)

svc = SVC()

svc.fit(X_train, Y_train)
```

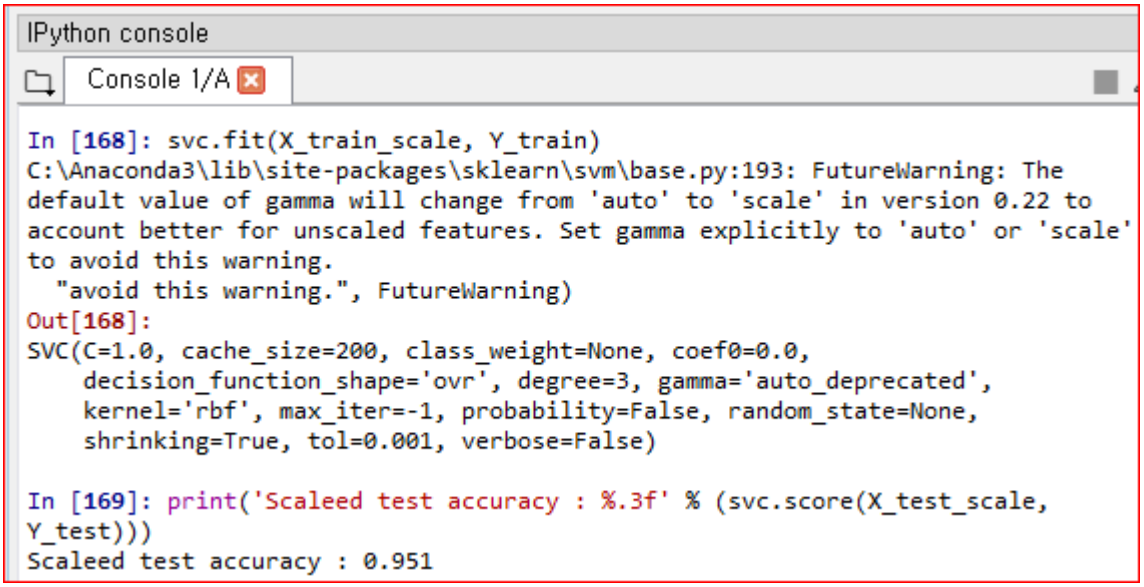



```
print('test accuracy : %.3f' % svc.score(X_test, Y_test))
```

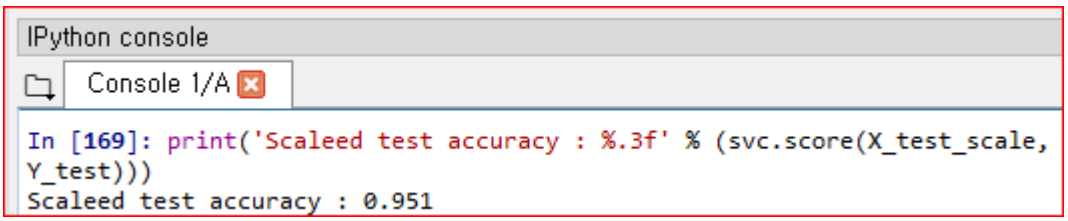


다음은 데이터를 MinMaxScaler 로 스케일을 조정하고 SVC 모델로 학습시켜보겠습니다.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X_train_scale = scaler.fit_transform(X_train)

X_test_scale = scaler.transform(X_test)

svc.fit(X_train_scale, Y_train)
```




```
print('Scaleed test accuracy : %.3f' % (svc.score(X_test_scale, Y_test)))
```



성능이 더 좋아졌습니다.

(참고 : introduction to MachineLearing with Python)