

.. 새해 복 많이 받으세요!
:::
▽

파이썬 딥러닝

이성주

seongjoo@codebasic.io

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help

In [8]: `import numpy`In [9]: `arr2d = numpy.array(data2d)`In [12]: `type(arr2d)`

ndarray "n차원 배열"

Out[12]: `numpy.ndarray`In [10]: `arr2d`Out[10]: `array([[1, 2],
[3, 4]])`In [11]: `arr2d * 2`Out[11]: `array([[2, 4],
[6, 8]])`

In []:

Trusted

Python 3

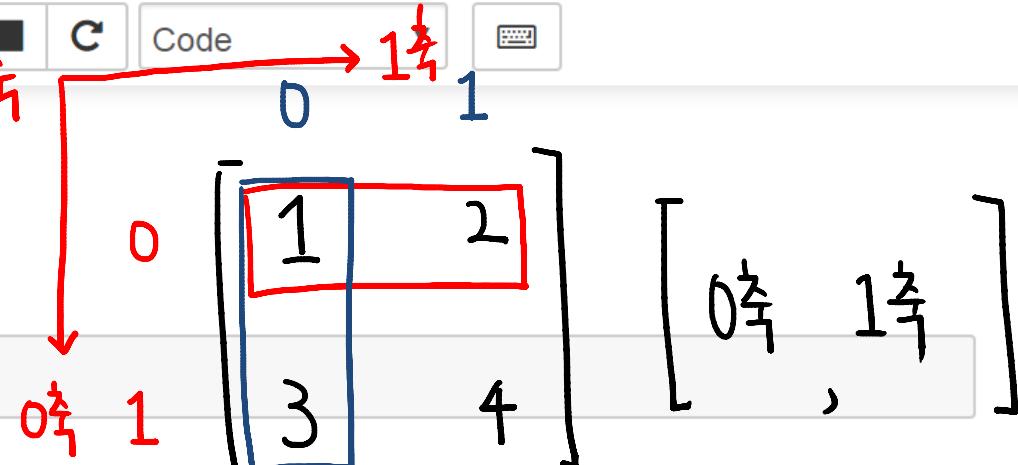
File Edit View Insert Cell Kernel Widgets Help

In [12]: array([[0, 1, 2],
 [3, 4, 5],
 [6, 7, 8]])

행 선택

In [13]: arr2d[0]

Out[13]: array([1, 2])



열 선택

In [14]: arr2d[:, 0]

Out[14]: array([1, 3])

[:, 0]

"모두"

S ≤ E

In []:

Trusted

Python 3

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

NumPy

Pandas

In [15]: `import pandas`

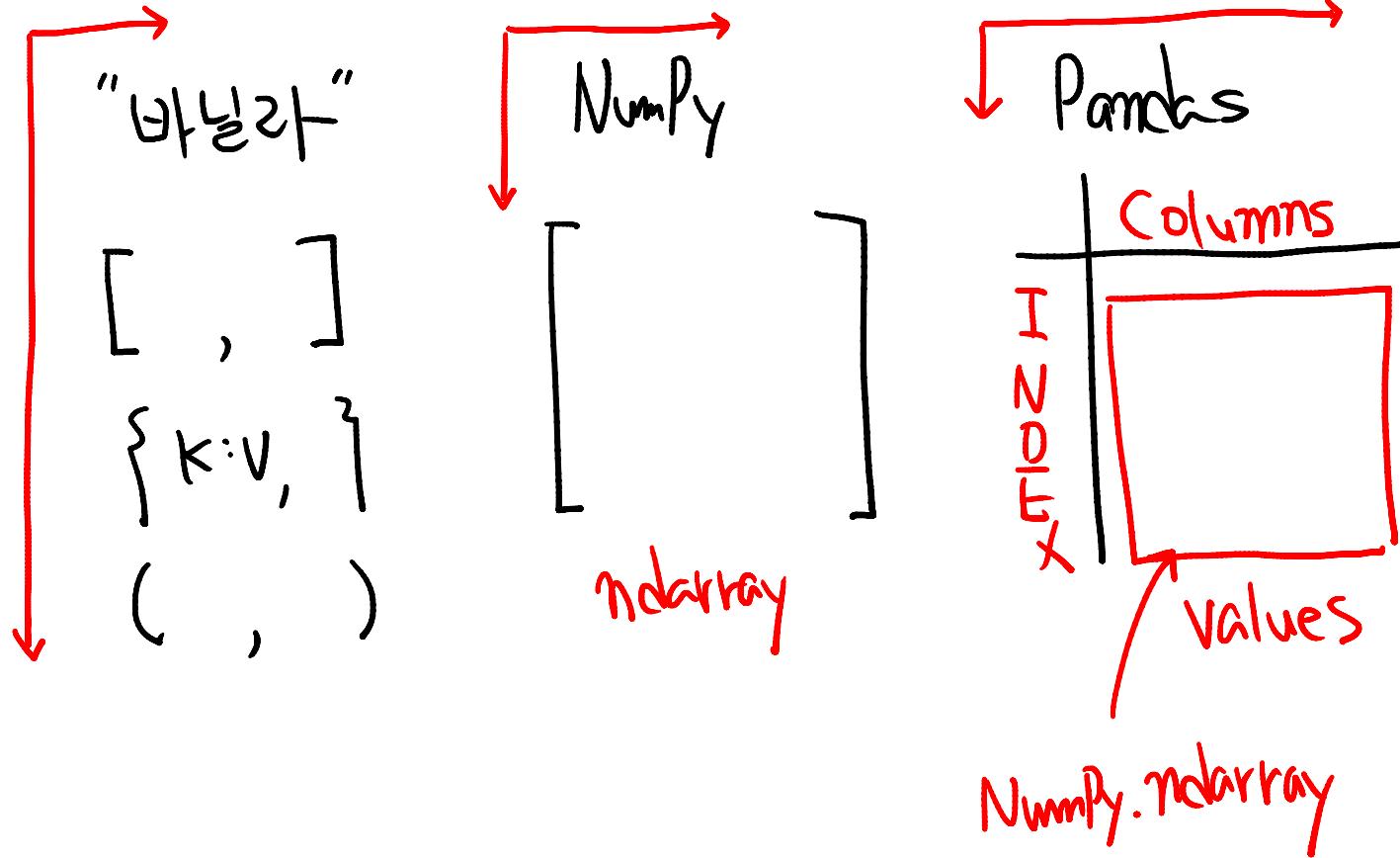
$$\begin{bmatrix} [1, 2], [3, 4] \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

In [17]: `arr2d = numpy.array(data2d)`
`arr2d`Out[17]: `array([[1, 2], [3, 4]])`In [18]: `frame = pandas.DataFrame(data2d)`
`frame`

Out[18]:

	0	1
0	1	2
1	3	4

In []:



1943 McCulloch-Pitts 누런

입력

가중치 (중요도)

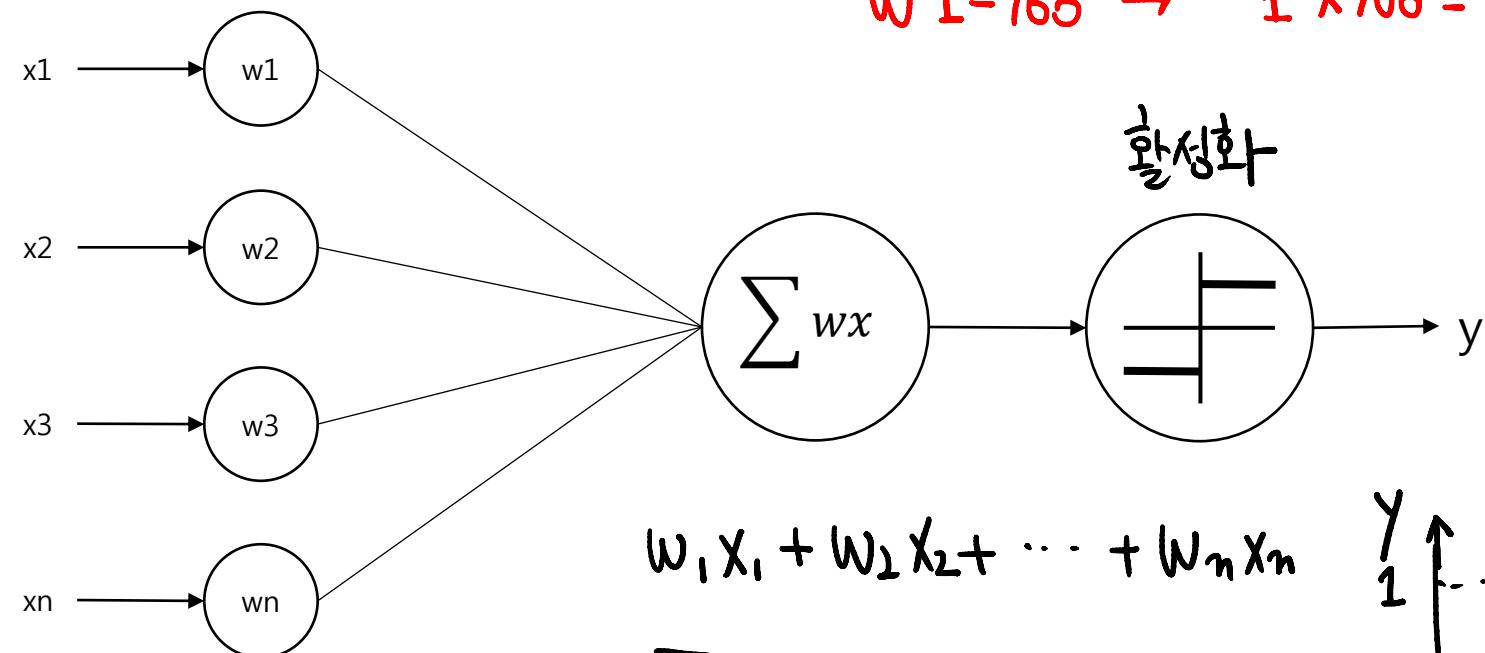
$$x_1 = 1$$

$$w_1 = 0$$

$$\rightarrow x_1 \times w_1 = 0$$

$$w_1 = 100$$

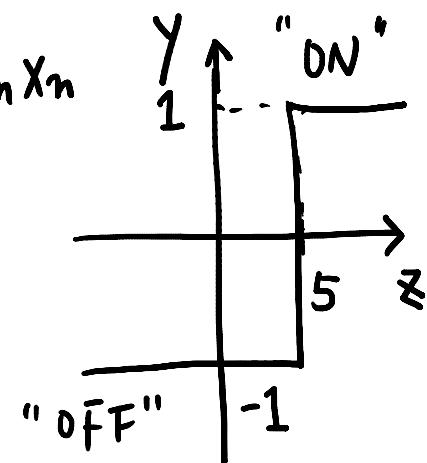
$$\rightarrow 1 \times 100 = 100$$



$$w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

$$= \sum w x$$

$$= z$$



Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help

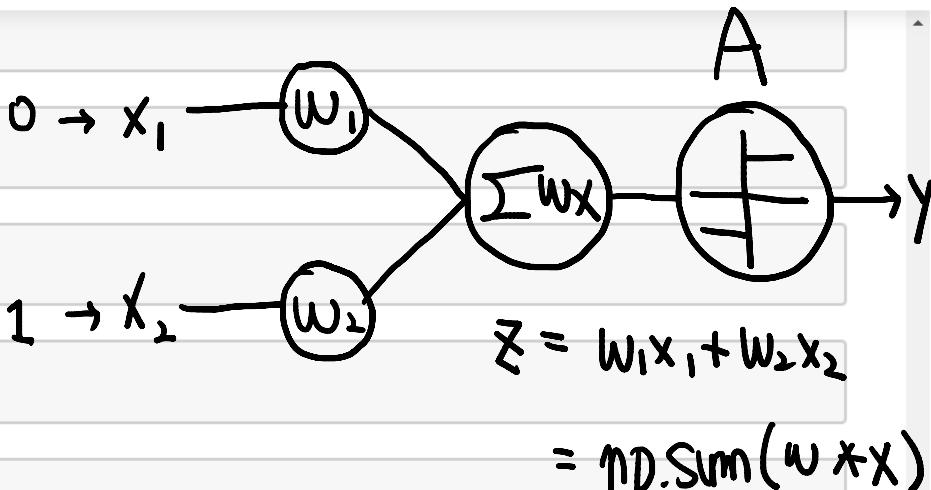
In [20]: `import numpy as np`In [21]: `x = np.array([0, 1])`In [22]: `w = np.array([0.5, 0.5])`In [23]: `b = -0.5`In [24]: `z = np.sum(w*x) + b`

Out[24]: 0.0

In [25]: `y = 1 if z > 0 else -1`

Out[25]: -1

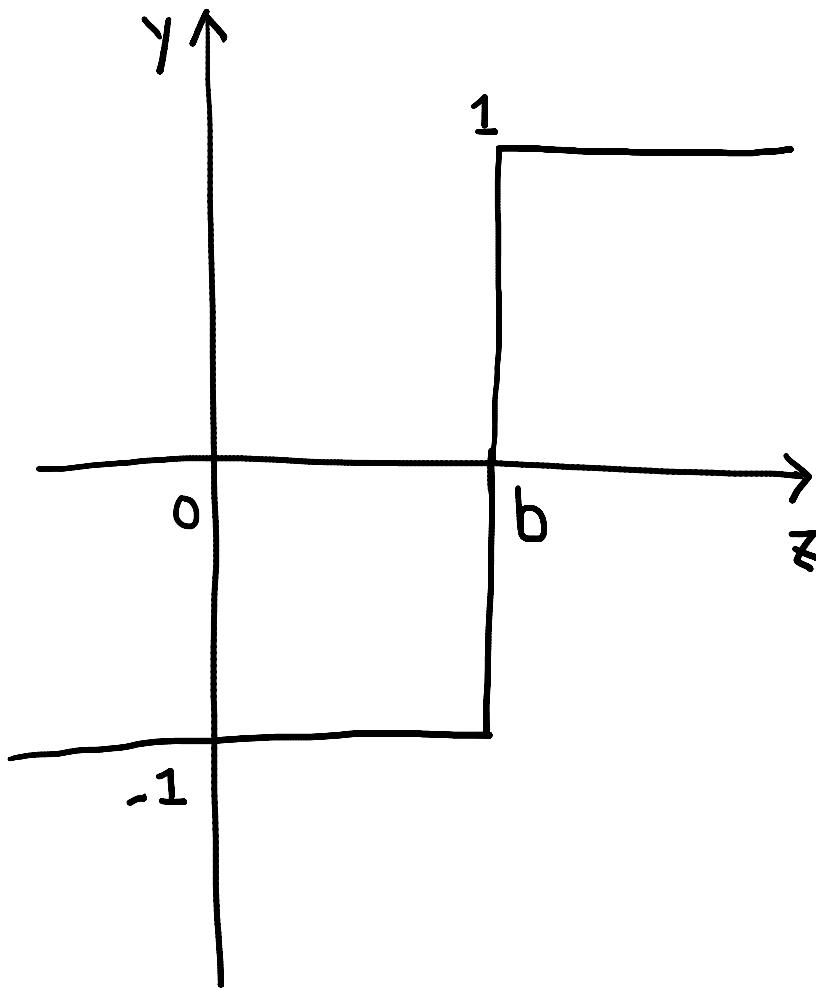
In []:

 $w * x$

$$\text{Sum} ([0 \times 0.5, 1 \times 0.5]) \rightarrow 0.5$$

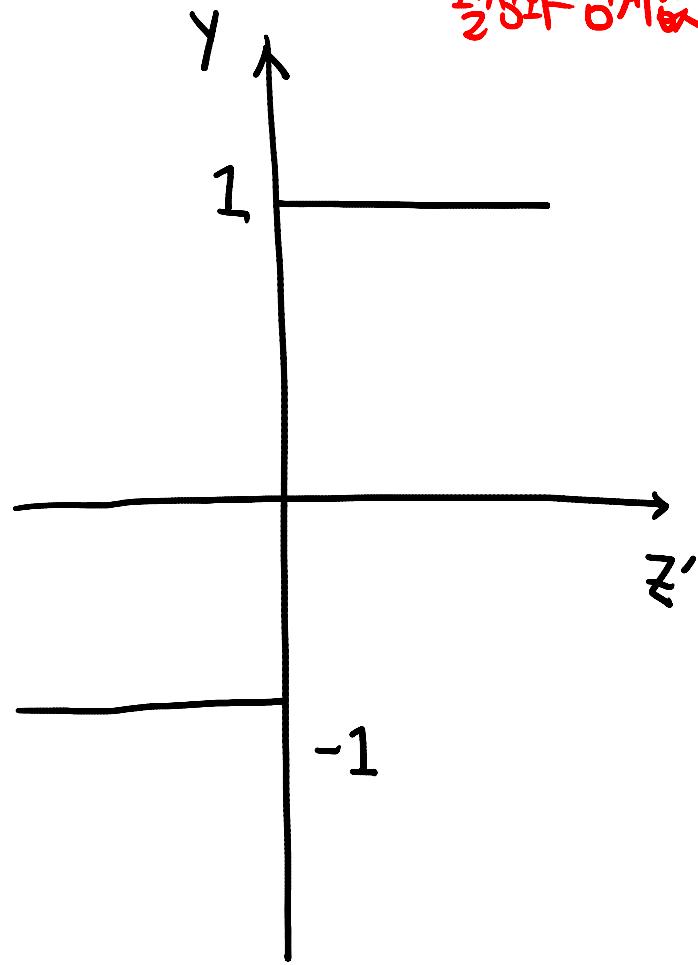
편향(bias)

$$z = \sum w x$$



$$\tilde{z} = \sum w x + b^{\text{bias}}$$

활성화 임계값



Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help



```
In [26]: def AND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7
    z = np.sum(w*x) + b
    y = 1 if z > 0 else 0
    return y
```

```
In [27]: def test(logic):
    for x1, x2 in [(0, 0), (0, 1), (1, 0), (1, 1)]:
        y = logic(x1, x2)
        print(x1, x2, '|', y)
```

```
In [28]: test(AND)
```

0	0		0
0	1		0
1	0		0
1	1		1

```
In [ ]:
```

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help



1 1 | 1

```
In [29]: def OR(x1, x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.2
    z = np.sum(w*x) + b
    y = 1 if z > 0 else 0
    return y
```

```
In [30]: test(OR)
```

0	0		0
0	1		1
1	0		1
1	1		1

```
In [ ]:
```

File Edit View Insert Cell Kernel Widgets Help



In [36]: `x = np.array([1, 1])
w = np.array([0.5, 0.5])
b = -0.5`

In [35]: `def net_input(x, w, b):
 return np.sum(w*x)+b`

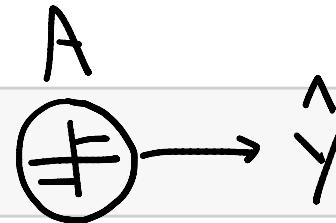
$$\sum w_i x_i$$

In [37]: `z = net_input(x, w, b)`

In [38]: `z`

Out[38]: 0.5

In [39]: `def activation(z):
 return 1 if z > 0 else -1`



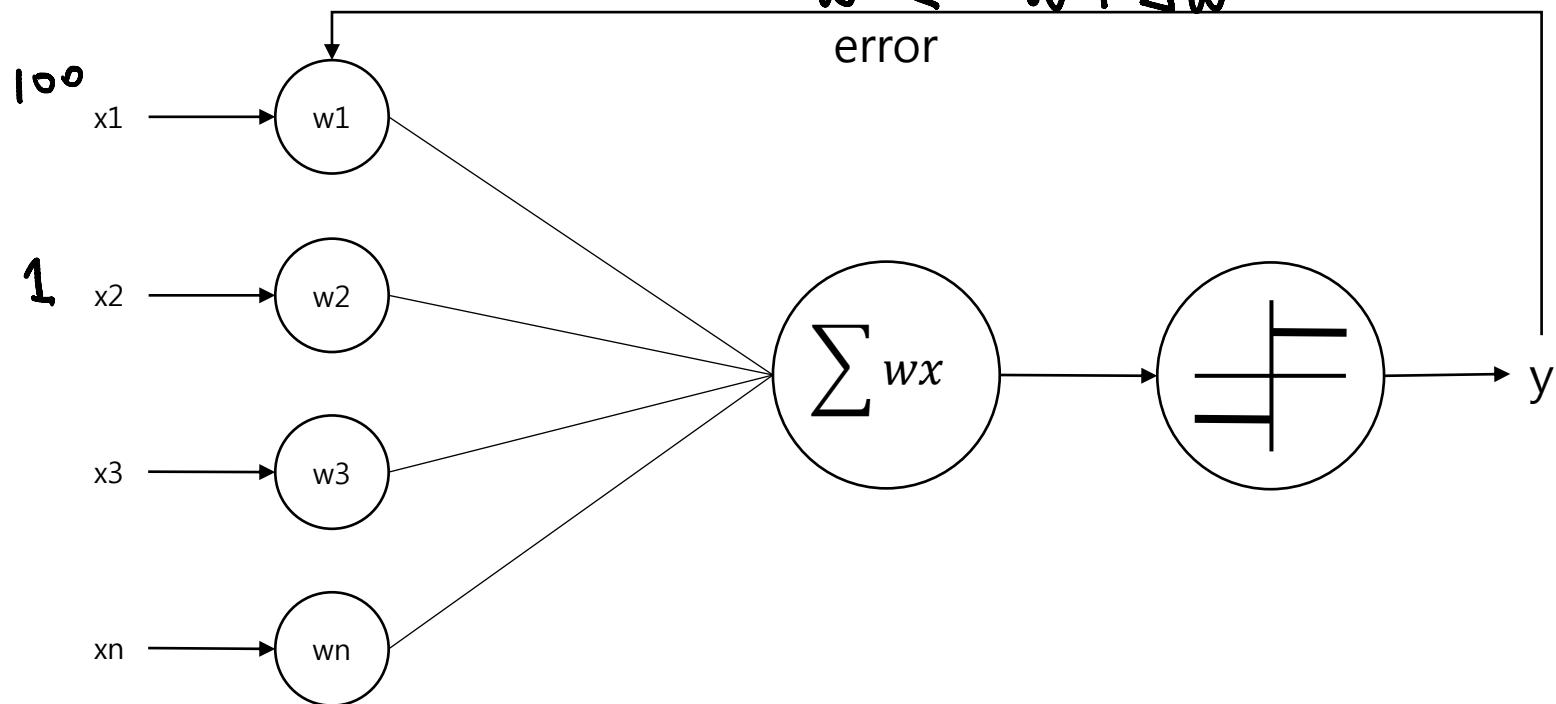
In [41]: `y_pred = activation(z)`

In []:

1958 퍼셉트론

$$\text{ERROR} = y - \hat{y}$$

$$\Delta w_i = \text{ERROR} \times \frac{x_i}{\text{error}}$$



File Edit View Insert Cell Kernel Widgets Help



```
In [42]: def 가중치갱신(y, y_pred, x):
    오류 = y - y_pred
    w_delta = 오류 * x
    return w_delta
```

```
In [46]: x
```

```
Out[46]: array([1, 1])
```

정답과 예측값이 같은 경우

```
In [44]: 가중치갱신(y=1, y_pred=1, x=x)
```

```
Out[44]: array([0, 0])
```

```
In [45]: 가중치갱신(y=1, y_pred=-1, x=x)
```

```
Out[45]: array([2, 2])
```

```
In [ ]:
```



In [55]:

```

def 학습(X, y, 반복=10, 학습률=0.1):
    w = np.array([0.] * X.shape[1])
    b = 0.

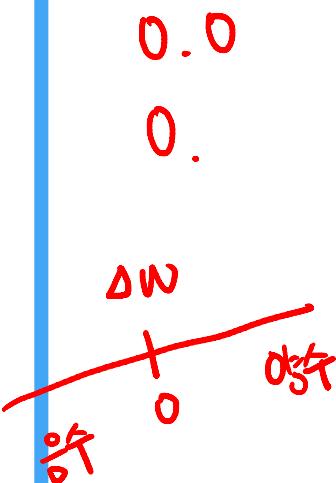
    error_history = []
    for i in range(반복):
        print('{0} 번째 훈련 중 ...'.format(i))
        → z = np.dot(X, w) + b
        → y_pred = np.where(z > 0, 1, -1)

        error = y - y_pred
        sse = np.sum(error**2) # 오류의 제곱합
        error_history.append(sse)

        # 가중치 갱신
        w_delta = 학습률 * error
        w += np.dot(w_delta, X)
        b += np.sum(w_delta)

    print('{0}) w: {1}, b: {2} WtError: {3}'.format(
        i+1, w, b, sse))
    return w, b, error_history

```



In [56]:

X

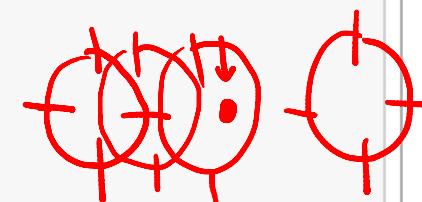
x_1, x_2

$$\begin{bmatrix} 0, 0 \\ 0, 1 \\ 1, 0 \\ 1, 1 \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ w_1 \end{bmatrix}$$

$+ b$

표표

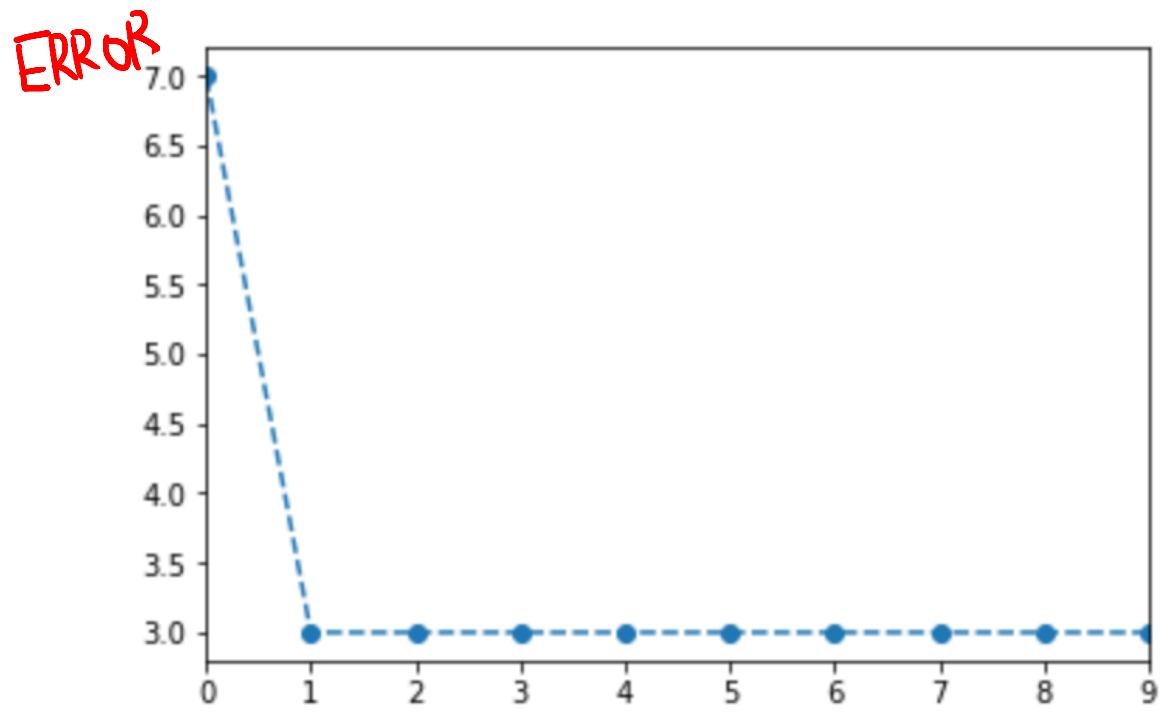


File Edit View Insert Cell Kernel Widgets Help



In [65]: Series(error_history).plot(style='o--')

Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0xa440358>



In []:

File Edit View Insert Cell Kernel Widgets Help



In [74]: `import numpy as np
import pandas as pd
from pandas import Series, DataFrame
%matplotlib inline`

노트: 코드 II 풀기

In [1]: `data = list(range(1, 11))`

In [2]: `data`

Out[2]: `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`

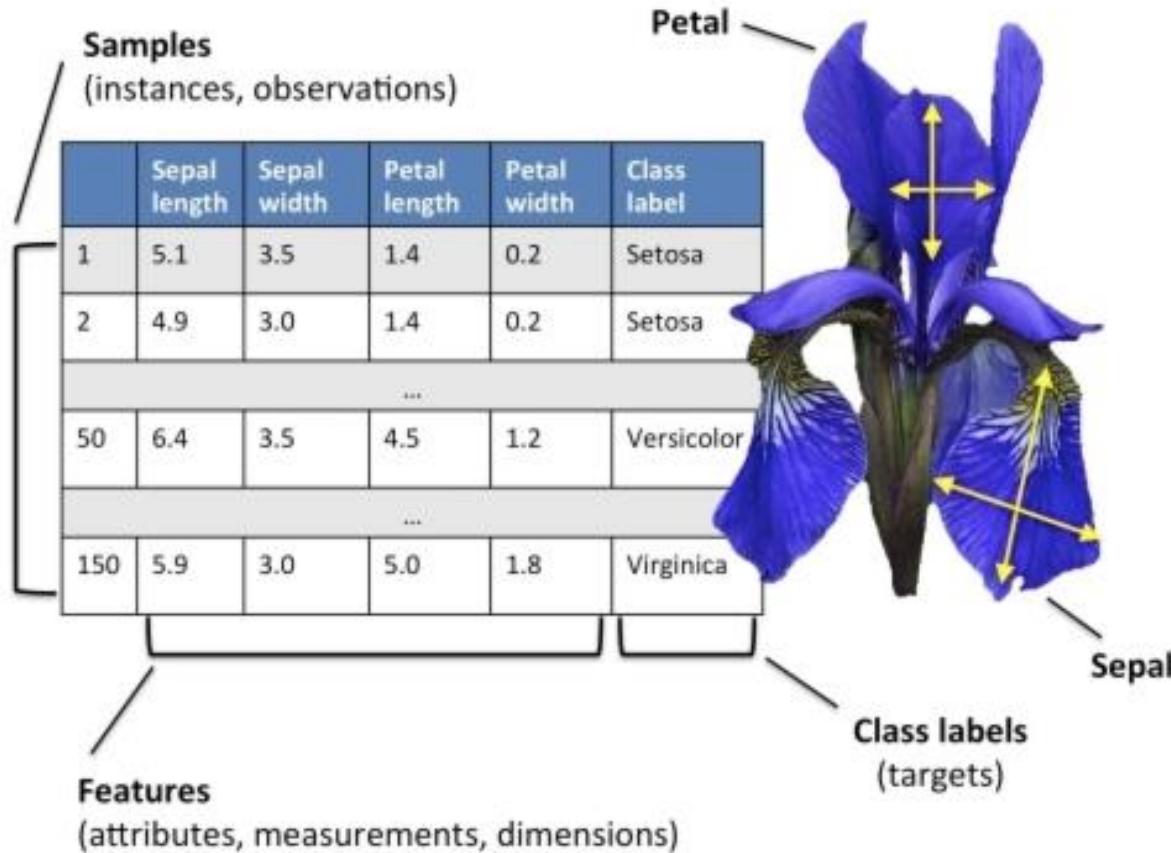
In [3]: `data * 2`

Out[3]: `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`

In [4]: `[n*2 for n in data]`

Out[4]: `[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]`

Iris



File Edit View Insert Cell Kernel Widgets Help



Scikit Learn

In [77]: `붓꽃 = pd.read_csv('data/iris.data', header=None)`

In [78]: `붓꽃[:5]`

Out [78]:

	x_1	x_2	x_3	x_4	
	0	1	2	3	4
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

y "정답"

In []:

File Edit View Insert Cell Kernel Widgets Help



In [77]: `붓꽃 = pd.read_csv('data/iris.data', header=None)`

In [78]: `붓꽃[:5]`

In [82]: `X = 붓꽃.iloc[:, 0:4]`

In [83]: `X[:5]`

Out[83]:

	0	1	2	3
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

모든 행 선택

열 선택
0:4

스칼라

1

'abc'

Series

0

벡터

1

2

3

L4

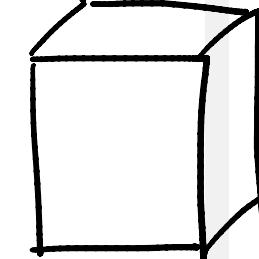
Dataframe

행렬

1 2

3 4

텐서



y X

In [84]: `y = 붓꽃[4]`

File Edit View Insert Cell Kernel Widgets Help



In [91]: `w, b, error_history = 학습(X, y)`

...

In [102]: `def 로직게이트_만들기(w, b):
 def logic_gate(x1, x2):
 x = np.array([x1, x2])
 z = np.sum(w*x) + b
 y = 1 if z > 0 else 0
 return y
 return logic_gate`

In [103]: `w, b`

Out[103]: `(array([0.1, 0.1]), -0.10000000000000003)`

In [104]: `AND = 로직게이트_만들기(w, b)`

In [105]: `test(AND)`

0	0		0
0	1		0
1	0		0
1	1		1

File Edit View Insert Cell Kernel Widgets Help



Code



In [119]: len(붓꽃)

Out[119]: 150

In [120]: data = 붓꽃[:100]

In [121]: X = data.iloc[:, 0:4]

In [122]: X[:5]

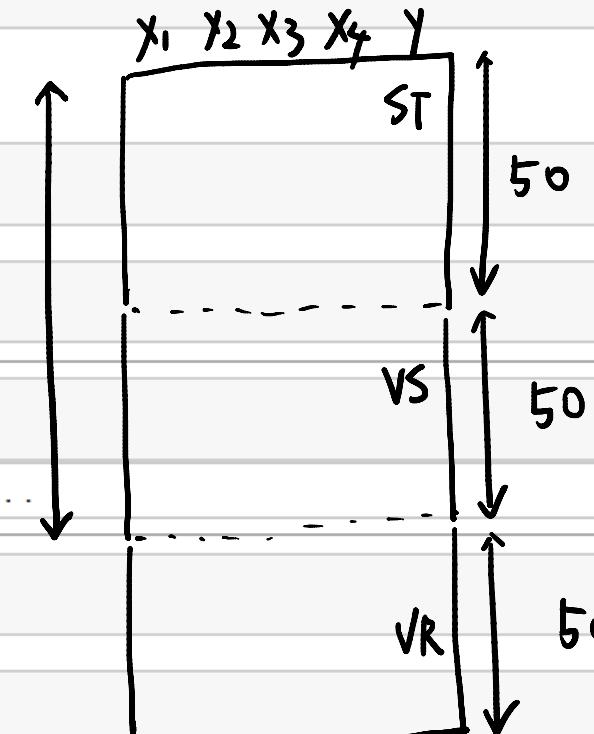
In [123]: y = data[4]

In [124]: y[:5]

In [125]: y.value_counts()

Out[125]: Iris-versicolor 50
Iris-setosa 50
Name: 4, dtype: int64

$$\hat{y} = \begin{cases} 1 \rightarrow ST \\ -1 \rightarrow VS \end{cases}$$





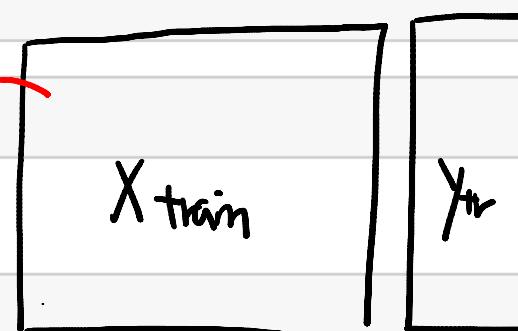
훈련/테스트 데이터 분리

In [126]: `from sklearn.model_selection import train_test_split`
scikit-Learn

In [127]: `X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.3)`

In [128]: `len(X)`
Out [128]: 100
[`:70`] 70%

모의고

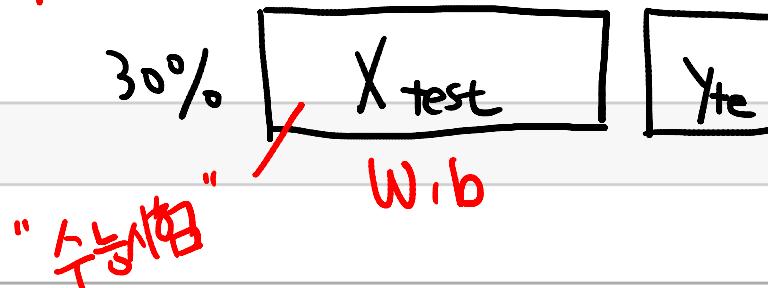


In [129]: `len(X_train)` "split"

Out [129]: 70

In [130]: `len(X_test)`

Out [130]: 30



In []:

File Edit View Insert Cell Kernel Widgets Help



In [131]: `from sklearn.linear_model import Perceptron`

모델 설정

학습률(η)

반복

In [132]: `model = Perceptron(eta0=0.1, n_iter=40)`

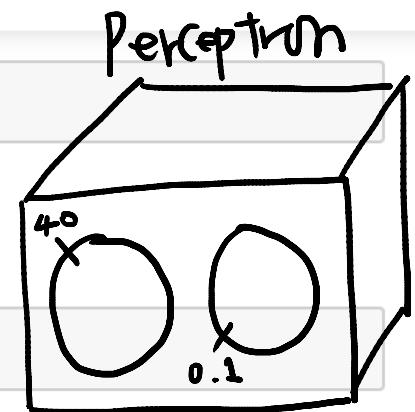
모델 훈련



정답

In [133]: `model.fit(X_train, y_train)`

...



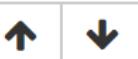
학습결과 설정된 매개변수

In [134]: `w = model.coef_
b = model.intercept_`

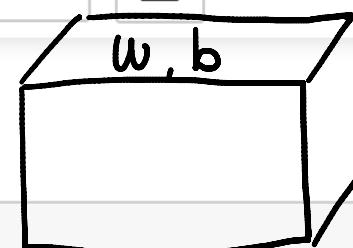
In [135]: `w`

Out[135]: `array([[-0.17, -0.69, 0.98, 0.46]])`

File Edit View Insert Cell Kernel Widgets Help



새로운 데이터에 대해 예측하기



```
In [137]: y_pred = model.predict(X_test)
```

```
In [138]: y_pred[:5]
```

```
Out[138]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-setosa'],
      dtype='|<U15')
```

예측된 결과 평가하기

```
In [141]: y_test.values == y_pred
```

```
Out[141]: array([ True,  True,  True,  True,  True,  True,  True,  True,
       True,  True,  True,  True,  True,  True,  True,  True,  True,
       True,  True,  True,  True,  True,  True,  True,  True,  True,
       True,  True,  True], dtype=bool)
```

File Edit View Insert Cell Kernel Widgets Help



In [138]: `y_pred[:5]`

...

예측된 결과 평가하기

In [143]: `(y_test.values == y_pred).sum() / len(y_test)`

Out[143]: 1.0

In [144]: `model.score(X_test, y_test)`

Out[144]: 1.0

In [159]: `model.loss_function`

Out[159]: <sklearn.linear_model.sgd_fast.Hinge at 0xaa965d0>

"왜 잘 될까?"

In [148]: `X[:3]`

Out[148]:

0 1 2 3

File Edit View Insert Cell Kernel Widgets Help

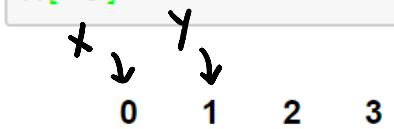


Code



In [148]: X[:3]

Out [148]:



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
---	-----	-----	-----	-----

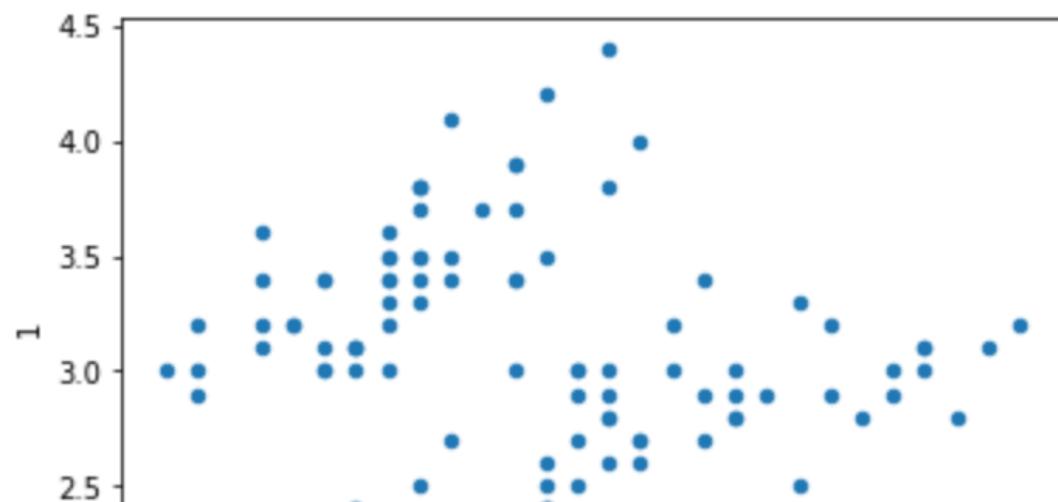
산포도

Pandas

DataFrame.plot

In [146]: X.plot(kind='scatter', x=0, y=1)

Out [146]: <matplotlib.axes._subplots.AxesSubplot at 0xc6c70f0>



File Edit View Insert Cell Kernel Widgets Help



In [149]: `from plot_util import get_colormap`

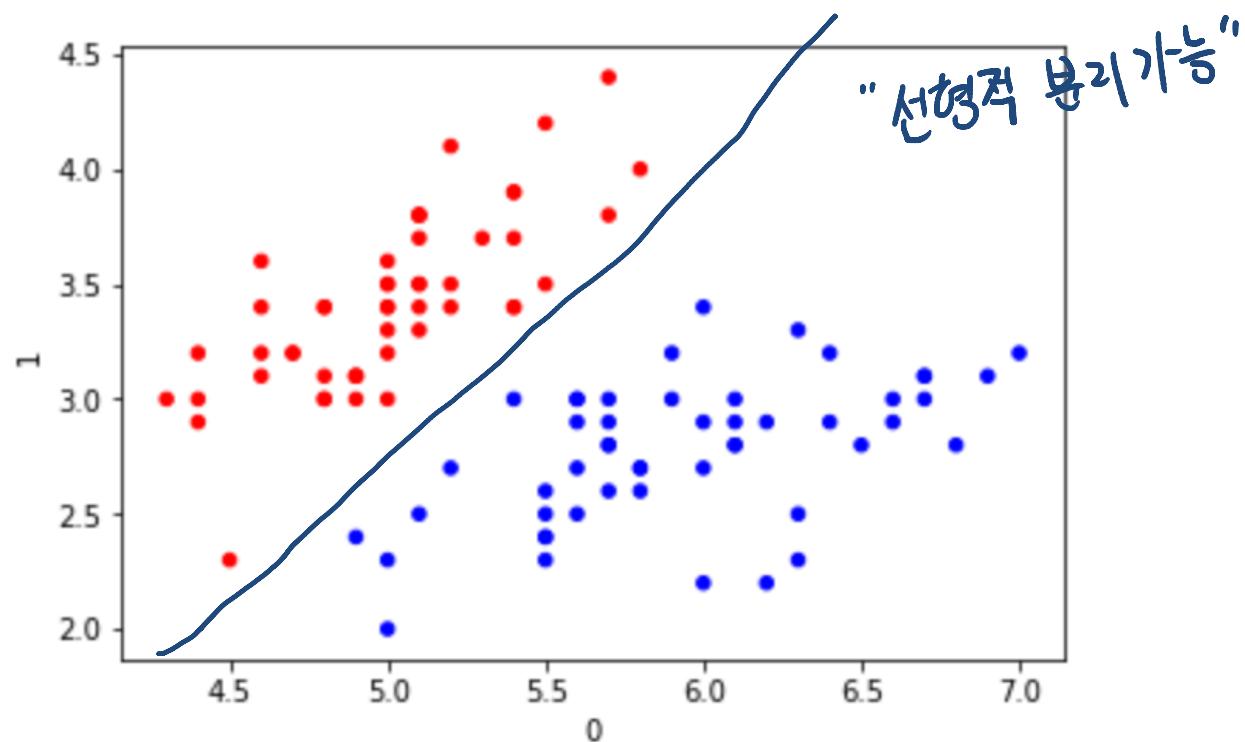
ST → 'red'

In [150]: `colormap = get_colormap(y)`

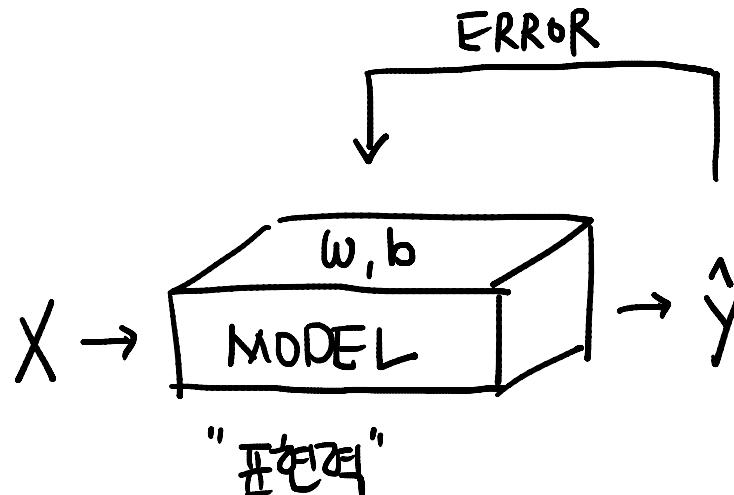
VS → 'blue'

In [151]: `X.plot(kind='scatter', x=0, y=1, c=colormap)`

Out[151]: <matplotlib.axes._subplots.AxesSubplot at 0xc6c7a90>



학습?



1) 최적화 (훈련)

$$\underset{w^*}{\operatorname{argmin}} \text{Error}(w)$$

"오류를 최소화하는 w "

$$\underset{w_{\text{train}}^*}{\operatorname{argmin}} \underset{\text{test}}{\text{Error}}(w)$$

지도학습

$$y \text{ "정답"} \\ \text{ERROR} = y - \hat{y}$$

비지도학습 ~ 데이터 전처리
탐색



$$\text{ERROR} = ? - \hat{y}$$

File Edit View Insert Cell Kernel Widgets Help



pandas 자료구조

In [160]: `data = list(range(1, 10))`

In [161]: `data`

Out[161]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [162]: `Series(data)`

Out[162]:

0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9

`dtype: int64`

:index

.Values

File Edit View Insert Cell Kernel Widgets Help

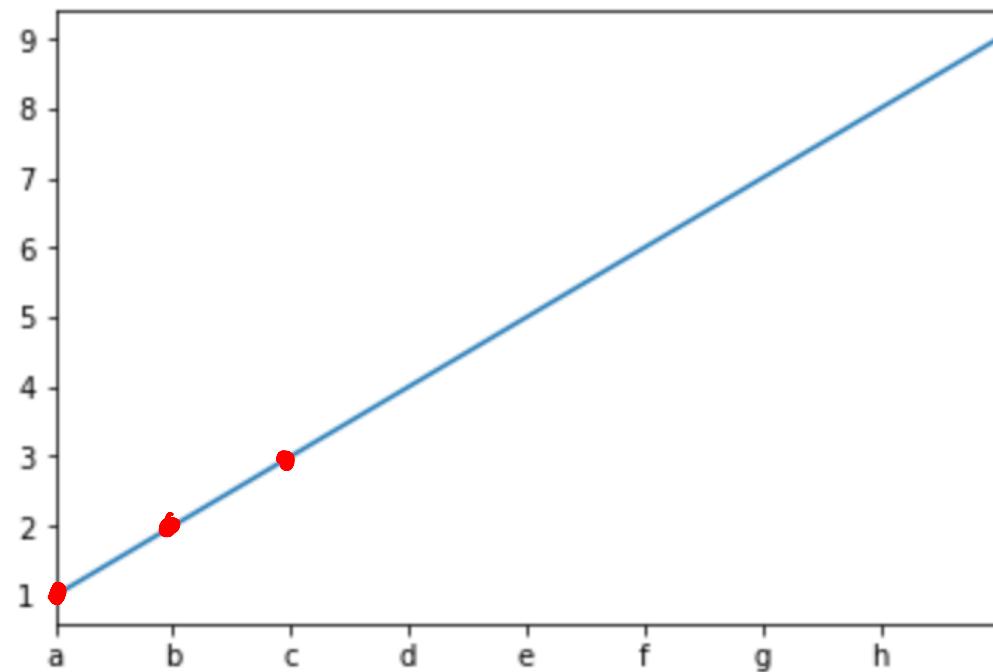


In [164]: `sr = Series(data, index=list('abcdefghi'))`

In [165]: `sr.plot() kind= "line" (기본값)`

Out [165]: <matplotlib.axes._subplots.AxesSubplot at 0xc5b9be0>

Values



index

In []:

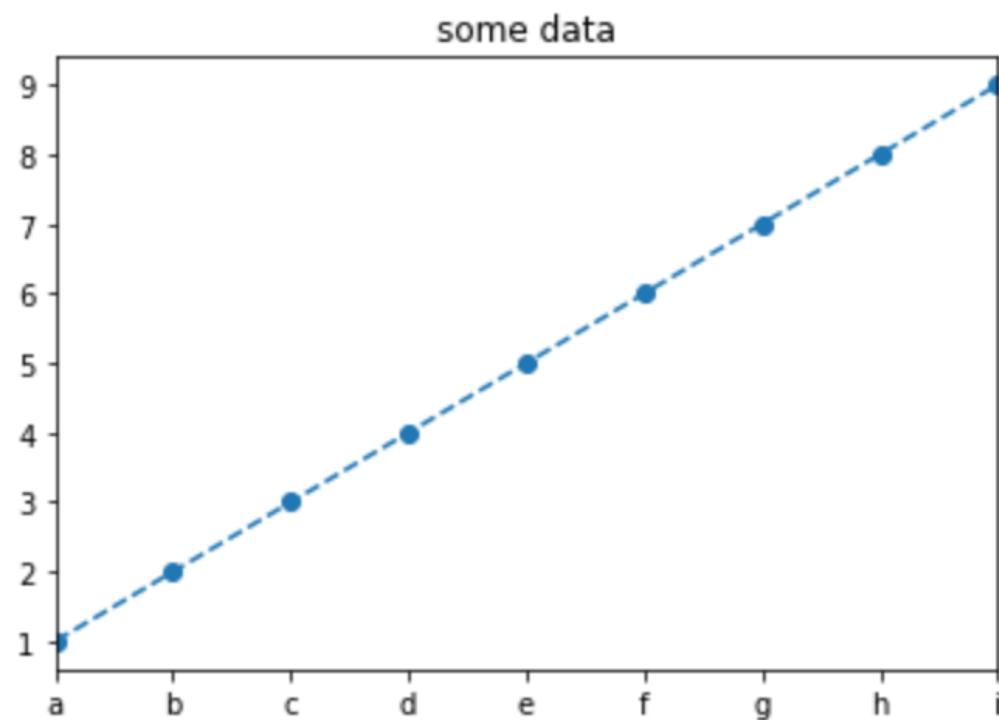
File Edit View Insert Cell Kernel Widgets Help



```
In [164]: sr = Series(data, index=list('abcdefghi'))
```

```
In [167]: sr.plot(title='some data', style='o--')
```

```
Out[167]: <matplotlib.axes._subplots.AxesSubplot at 0xc7e7240>
```



```
In [ ]:
```

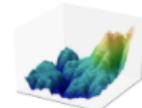
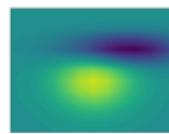
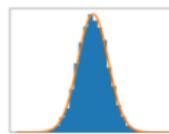


Fork me on GitHub

[home](#) | [examples](#) | [tutorials](#) | [pyplot](#) | [docs](#) »[modules](#) | [index](#)

Introduction

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [Matplotlib installation instructions](#).

Documentation

This is the documentation for Matplotlib version 2.1.1.

To get started, read the [User's Guide](#).

Other versions are available:

- [2.1.1](#) Latest stable version.
- [2.x](#) Latest git master (unstable)
- [2.0.2](#) Previous stable version.
- [1.5.3](#) Older stable version.

Trying to learn how to do a particular kind of plot? Check out the [examples gallery](#) or the [list of plotting commands](#).

powered by [NumFOCUS](#)Depsy [100th percentile](#)Travis-CI: [build](#) [failing](#)[Support matplotlib](#)[Support NumFOCUS](#)

Quick search

Go



[0:5]

DataFrame

In [169]: `야구 = pd.read_excel('data/baseball.xlsx')`In [170]: `야구[:5]`

Out [170]:

	선수명	팀명	경기	타석	타수	안타	홈런	득점	타점	볼넷	삼진	도루	BABIP	타율
0	모창민	NC	108	436	395	109	12	57	51	37	68	16	0.307	0.276
1	이호준	NC	126	508	442	123	20	46	87	60	109	2	0.324	0.278
2	김종호	NC	128	546	465	129	0	72	22	57	100	50	0.352	0.277
3	나성범	NC	104	458	404	98	14	55	64	33	95	12	0.279	0.243

File Edit View Insert Cell Kernel Widgets Help



In [170]: 야구[:5]

Series

In [171]: 야구['홈런']

Out[171]:

	야구['홈런']
0	12
1	20
2	0
3	14
4	6
5	0
6	0
7	0
8	2
9	0
10	4
11	0
12	0
13	0
14	0
15	3
16	0
17	1

홈런

File Edit View Insert Cell Kernel Widgets Help



```
In [169]: 야구 = pd.read_excel('data/baseball.xlsx')
```

```
In [173]: 야구 = 야구.set_index('선수명')
```

```
In [174]: 야구[:5]
```

Out[174]:

팀 경 타 안 훔 득 타 볼 삼 도 BABIP 타율 출루
명 기 석 수 안타 런 점 점 넷 진 류 율

선
수
명

모
창
민

이
호
준

김
종
호

	NC	108	436	395	109	12	57	51	37	68	16	0.307	0.276	0.339
--	----	-----	-----	-----	-----	----	----	----	----	----	----	-------	-------	-------

	NC	126	508	442	123	20	46	87	60	109	2	0.324	0.278	0.362
--	----	-----	-----	-----	-----	----	----	----	----	-----	---	-------	-------	-------

	NC	128	546	465	129	0	72	22	57	100	50	0.352	0.277	0.376
--	----	-----	-----	-----	-----	---	----	----	----	-----	----	-------	-------	-------

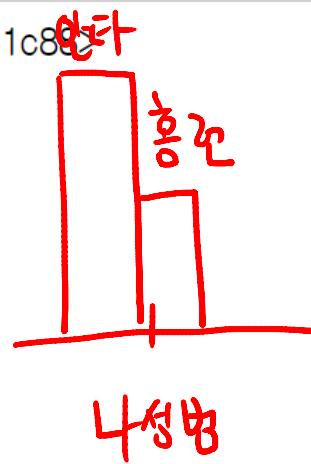
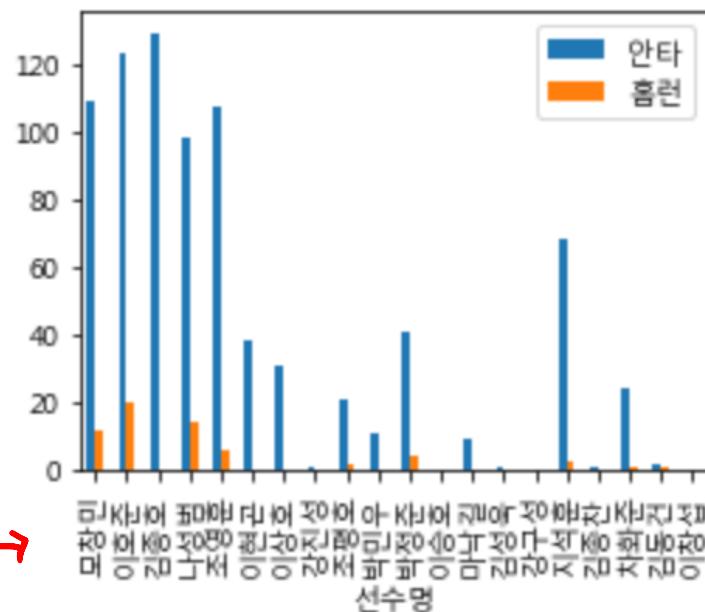


그래프에서, 한글을 지원하는 글씨체 설정

```
In [178]: import matplotlib  
matplotlib.rcParams['font.family'] = 'Malgun Gothic'
```

```
In [185]: 야구[['안타', '홈런']].plot(kind='bar', figsize=(4,3))
```

```
Out[185]: <matplotlib.axes._subplots.AxesSubplot at 0xcfa1c80>
```



index →

나성명

NumPy.ndarray

↓

	0	1
0	1	2
1	3	4

$[\dots, 0]$

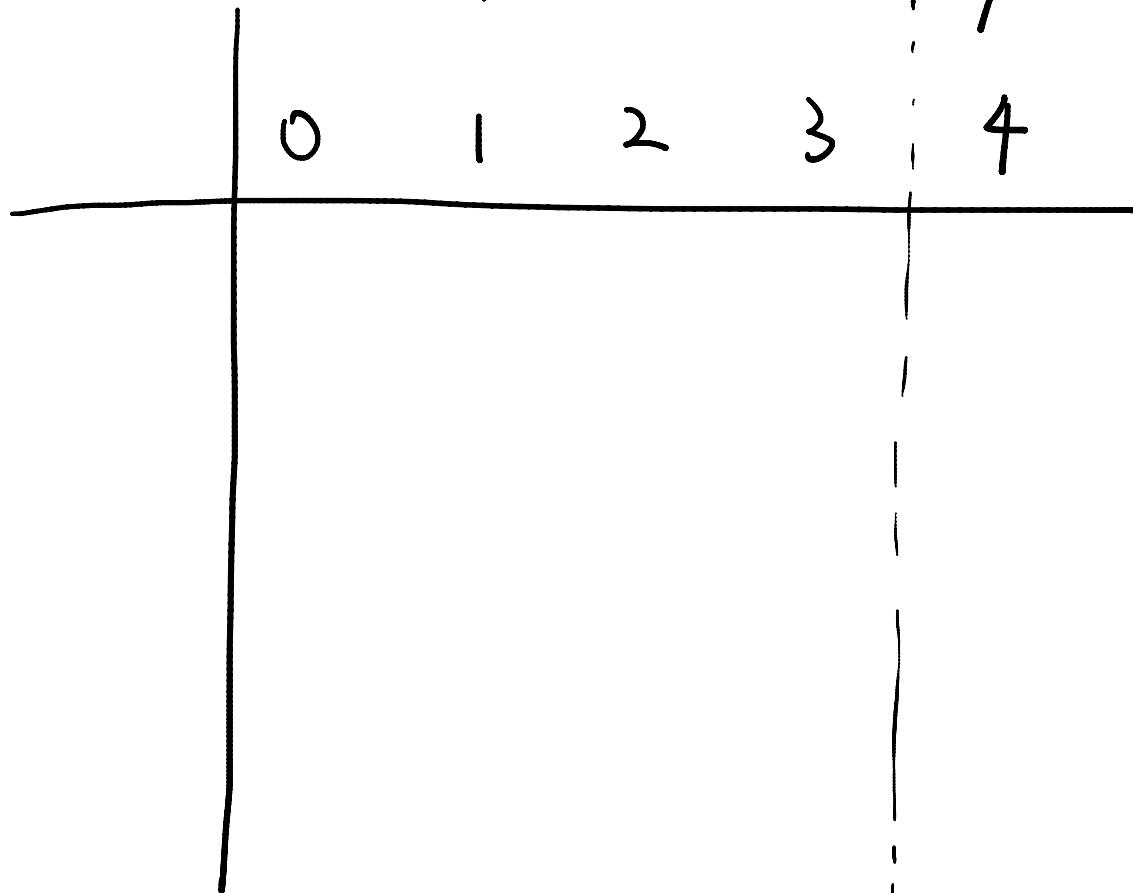
Pandas.DataFrame

	0	1
0	1	2
1	3	4

[0]

$i \text{loc} [0\frac{1}{3}, 1\frac{1}{3}]$

$y \text{loc}[:, 0:4]$



File Edit View Insert Cell Kernel Widgets Help



```
Iris-versicolor      50  
Name: 4, dtype: int64
```

```
In [193]: X = data.iloc[:, 0:4]
```

```
In [194]: X_train, X_test, y_train, y_test = train_test_split(  
          X, y, test_size=0.3)
```

```
In [195]: X_train[:3]
```

Out[195]:

"우와 쓰기"

	0	1	2	3
91	6.1	3.0	4.6	1.4
132	6.4	2.8	5.6	2.2
77	6.7	3.0	5.0	1.7

pandas 자료구조

```
In [160]: data = list(range(1, 10))
```

File Edit View Insert Cell Kernel Widgets Help



```
In [194]: X_train, X_test, y_train, y_test = train_test_split(  
          X, y, test_size=0.3)
```

```
In [195]: X_train[:3]
```

...

```
In [196]: model = Perceptron(eta0=0.1, n_iter=40)
```

```
In [197]: model.fit(X_train, y_train)
```

...

```
In [198]: model.score(X_test, y_test)
```

```
Out[198]: 0.9000000000000002
```

실행한 때마다 다른 수입니다!

pandas 자료구조

```
In [160]: data = list(range(1, 10))
```

```
In [161]: data
```

File Edit View Insert Cell Kernel Widgets Help



In [208]: `model = Perceptron(eta0=0.1, n_iter=40)`

동일 오류
같은 에러

```
results = []
for _ in range(5):
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3)

    model.fit(X_train, y_train)
    score = model.score(X_test, y_test)
    results.append(score)
```

In [209]: `results`

Out [209]: `[0.9333333333333335,`
`0.9666666666666667,`
`0.7666666666666672,`
`0.9666666666666667,`
`0.5333333333333333]`



In [210]: `Series(results).plot(kind='bar')`

Out [210]: <matplotlib.axes._subplots.AxesSubplot at 0xd21b0f0>

File Edit View Insert Cell Kernel Widgets Help

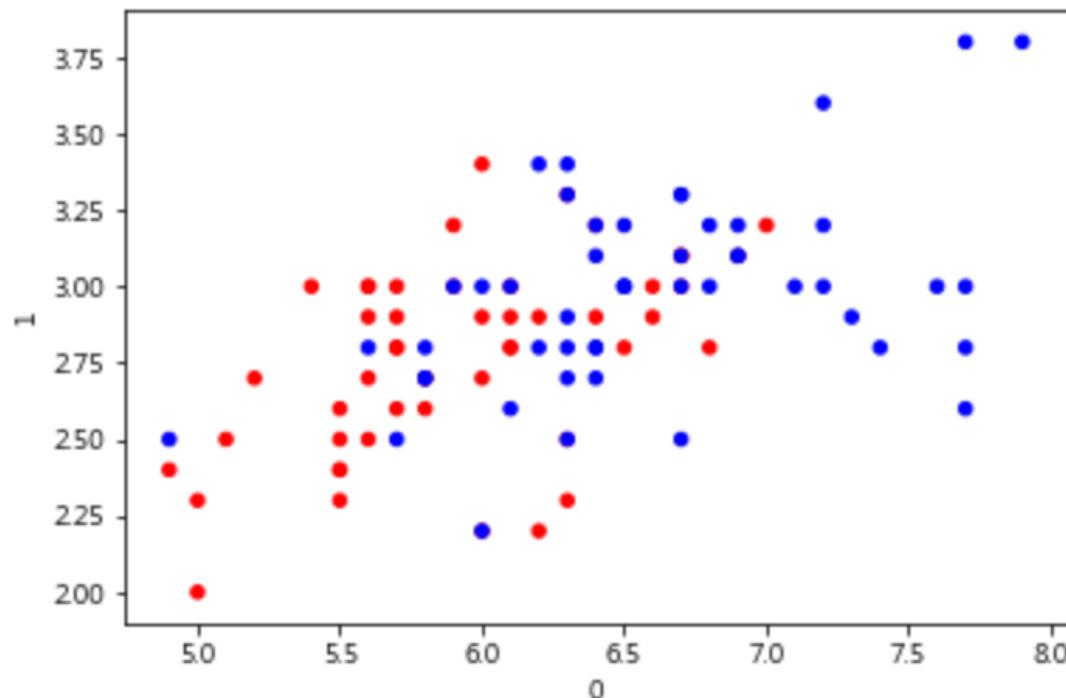


"왜 불안정 할까?"

In [212]: `colormap = get_colormap(y)`

In [213]: `X.plot(kind='scatter', x=0, y=1, c=colormap)`

Out[213]: <matplotlib.axes._subplots.AxesSubplot at 0xd2b4048>



File Edit View Insert Cell Kernel Widgets Help



In [213]: x.plot(kind='scatter', x='x', y='y', color='map')

"훈련 데이터는 백점 맞을까?"

최적화 실패!

In [214]: model.score(X_train, y_train)

Out [214]: 0.55714285714285716

pandas 자료구조

In [160]: data = list(range(1, 10))

In [161]: data

Out [161]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

In [164]: sr = Series(data, index=list('abcdefghi'))

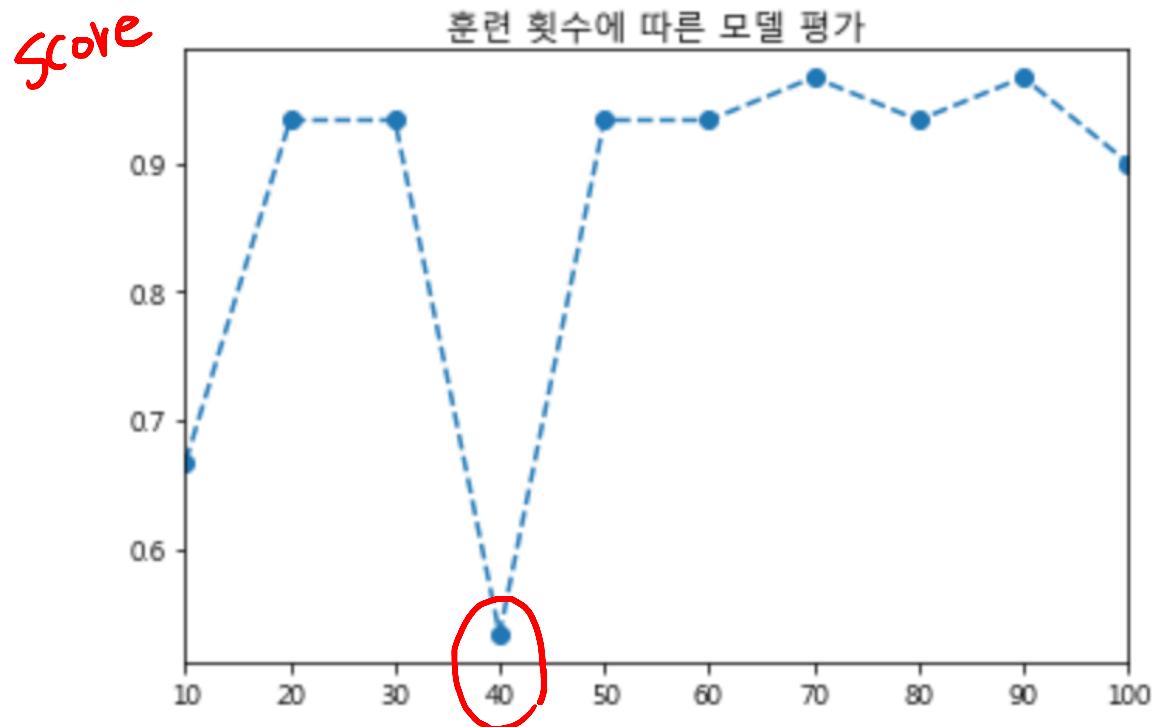
In [167]: sr.plot(title='some data', style='o--')



model.fit(X_train, y_train)

```
score = model.score(X_test, y_test)  
results.append(score)
```

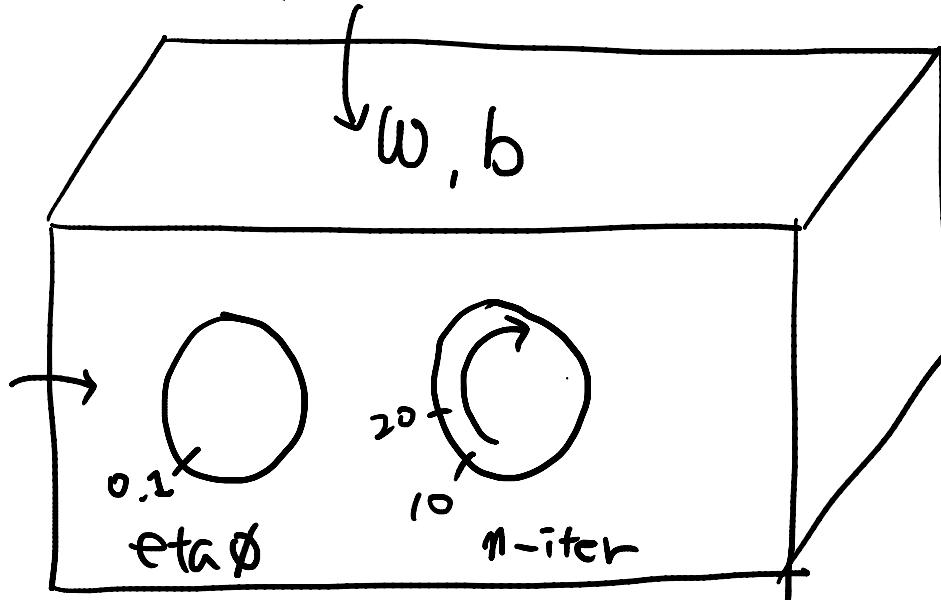
In [224]: report = Series(results, index=훈련횟수_리스트).plot(
style='o--', title='훈련 횟수에 따른 모델 평가')

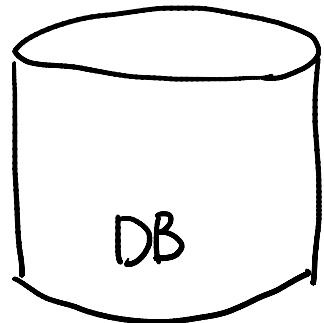
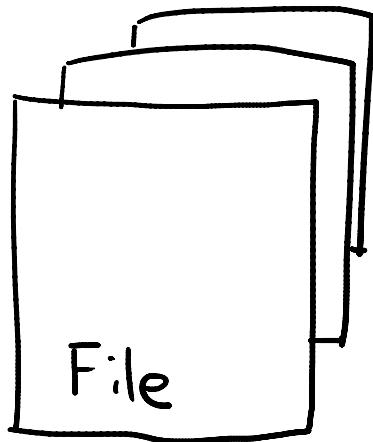


In []:

Parameter Vs. Hyper Parameter

매개변수(계통)

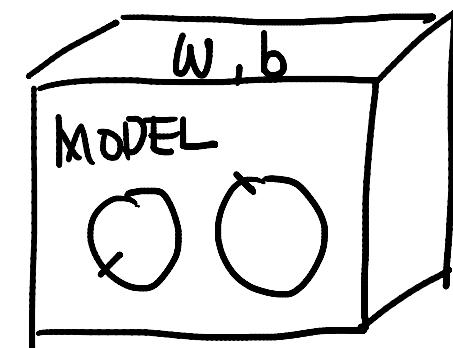
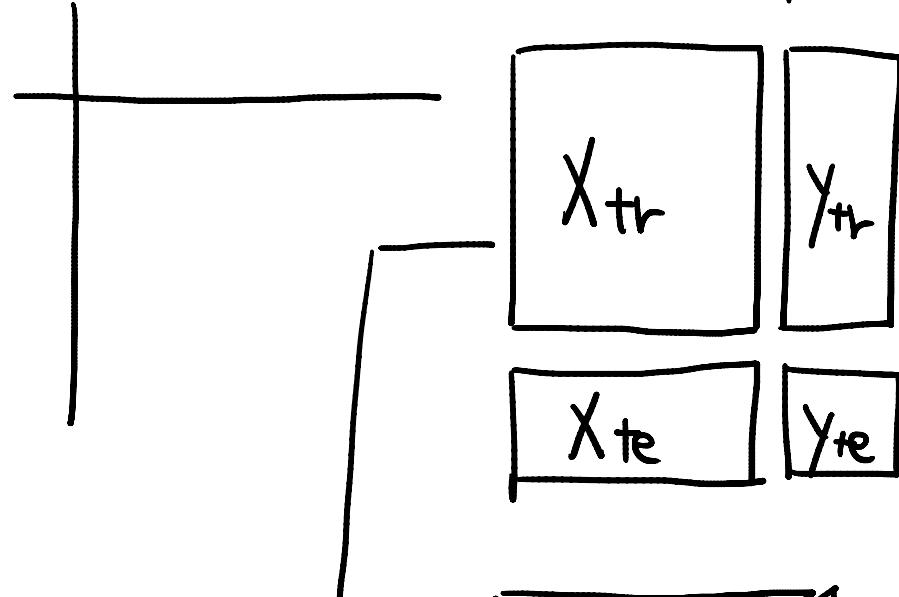




`pd.read_`

DataFrame

sklearn
train_test_Split



www



하이퍼 파라미터, 매개변수, 모델 성능

In [229]: `C_params = [0.001, 0.01, 0.1, 1., 10., 100., 1000.]`

1) 최적화(훈련)

```
In [233]: results = []
for C in C_params:
    model = LogisticRegression(C=C)
    model.fit(X_train, y_train)
    test_score = model.score(X_test, y_test)
    train_score = model.score(X_train, y_train)
    results.append((test_score, train_score))
```

$$\underset{w^*_{\text{train}}}{\operatorname{Argmin}} \text{Error}(w)$$

2) 일반화(테스트)

In [236]: `report = DataFrame(results, index=C_params)`

$$\underset{w^*_{\text{test}}}{\operatorname{Argmin}} \text{Error}(w)$$

In [237]: `report.columns = ['test', 'train']`

$$w^*_{\text{test}}$$

In [238]: `report`

$$w^*_{\text{train}} \approx w^*_{\text{test}}$$

Out[238]:

	test	train
	0.001	0.466667

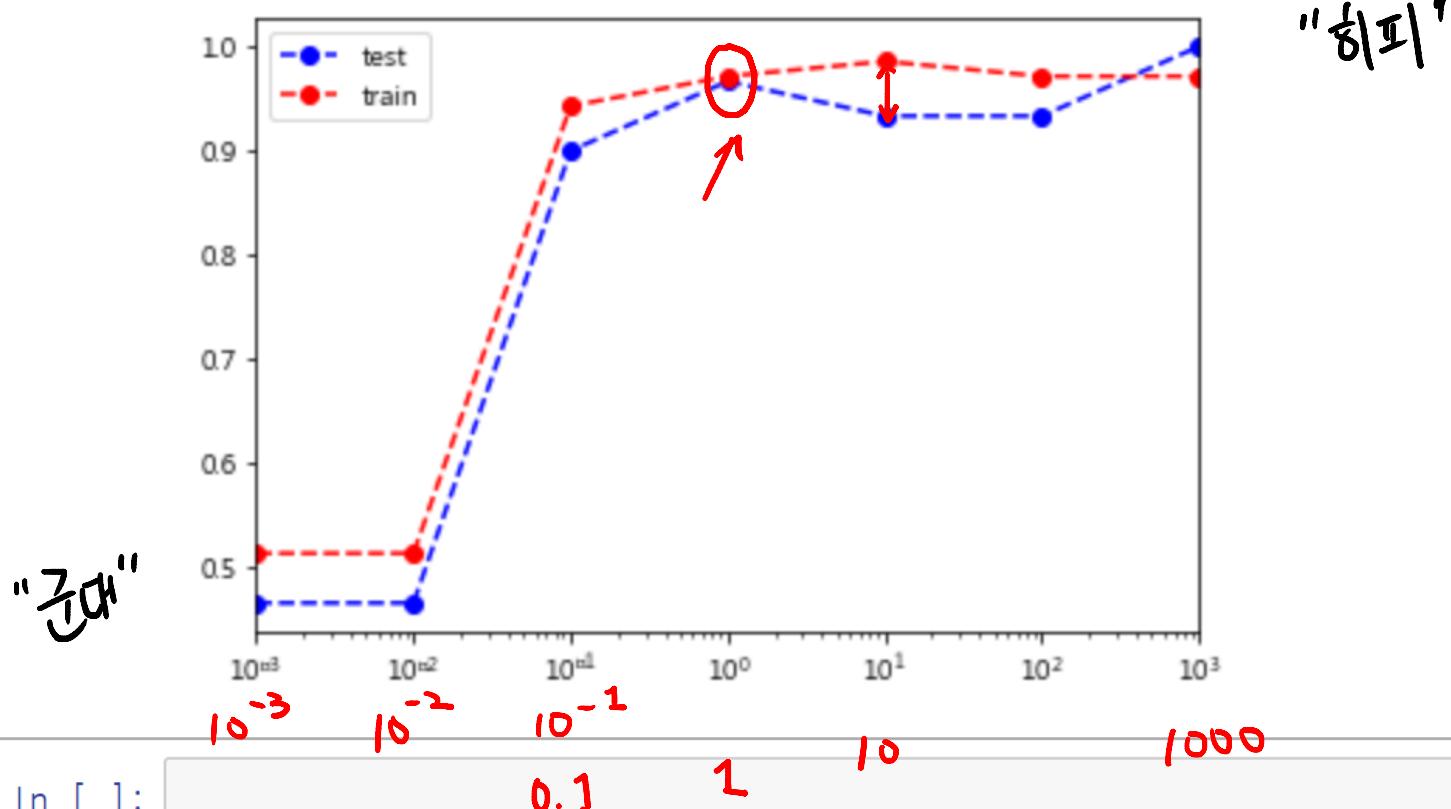
File Edit View Insert Cell Kernel Widgets Help



Code



warn('Substituting with a dummy symbol.', mathextwarning)



File Edit View Insert Cell Kernel Widgets Help



??!

```
w = model.coef_
w2sum = np.sum(w**2)
results.append((test_score, train_score, w2sum))
```

In [244]: report = DataFrame(results, index=C_params)

In [245]: report.columns = ['test', 'train', 'w2sum']

In [246]: report

Out[246]:

	test	train	w2sum
0.001	0.466667	0.514286	0.000414
0.010	0.466667	0.514286	0.021209
0.100	0.900000	0.942857	1.118424
1.000	0.966667	0.971429	13.428179
10.000	0.933333	0.985714	57.559760
100.000	0.933333	0.971429	135.796330
1000.000	1.000000	0.971429	199.025736

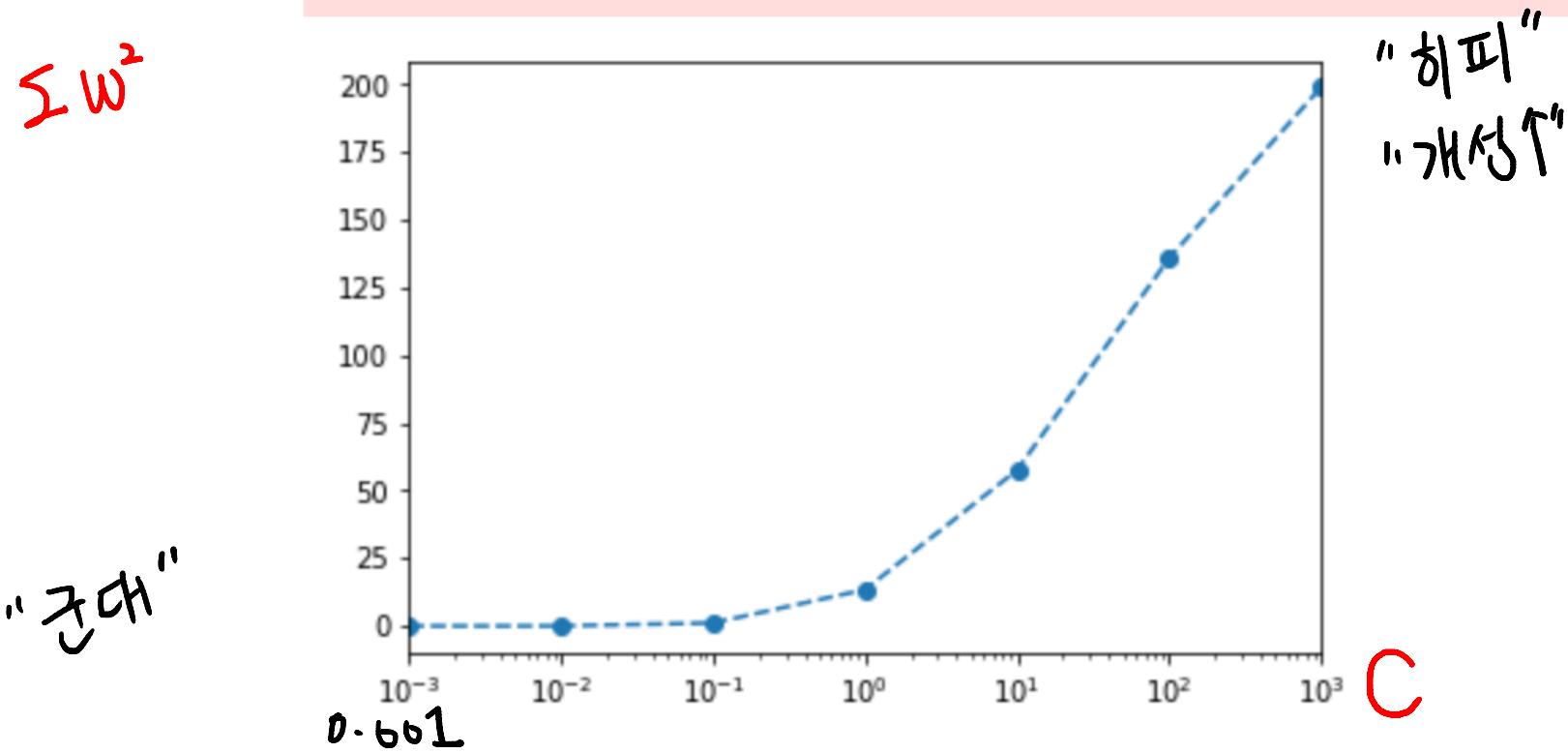
File Edit View Insert Cell Kernel Widgets Help



In [257]: `report['w2sum'].plot(logx=True, style='o--')`

Out [257]: <matplotlib.axes._subplots.AxesSubplot at 0x10353588>

C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\font_manager.py:1297: UserWarning: findfont: Font family ['San-Serif'] not found. Falling back to DejaVu Sans
(prop.get_family(), self.defaultFamily[fontext]))



가중치와 모델의 표현력

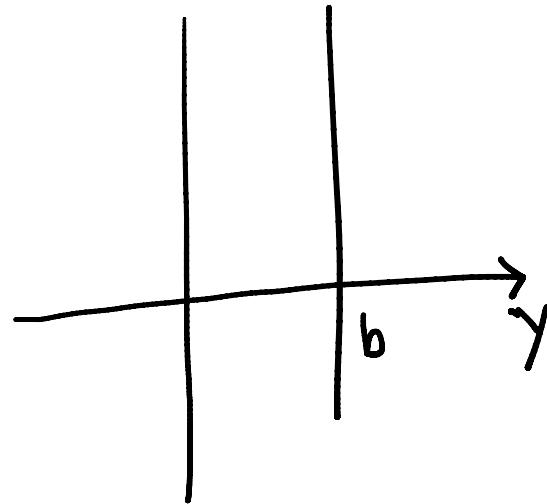
$$y = b + w_1 x_1 + w_2 x_2^2 + \dots$$

$$w_1 = 0, w_2 = 0$$

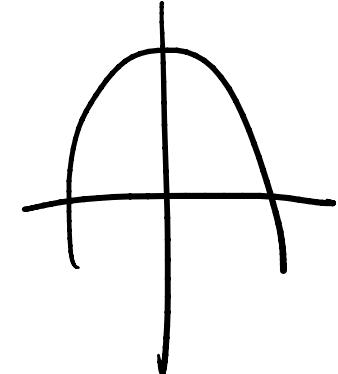
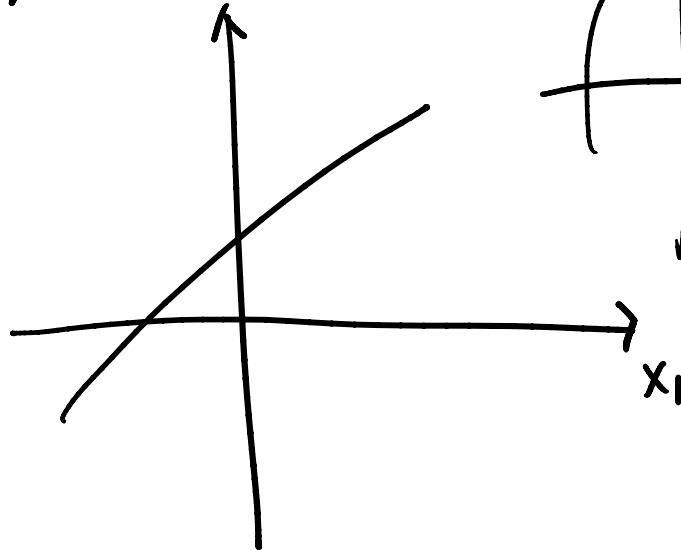
$$w_1 = 1, w_2 = 0$$

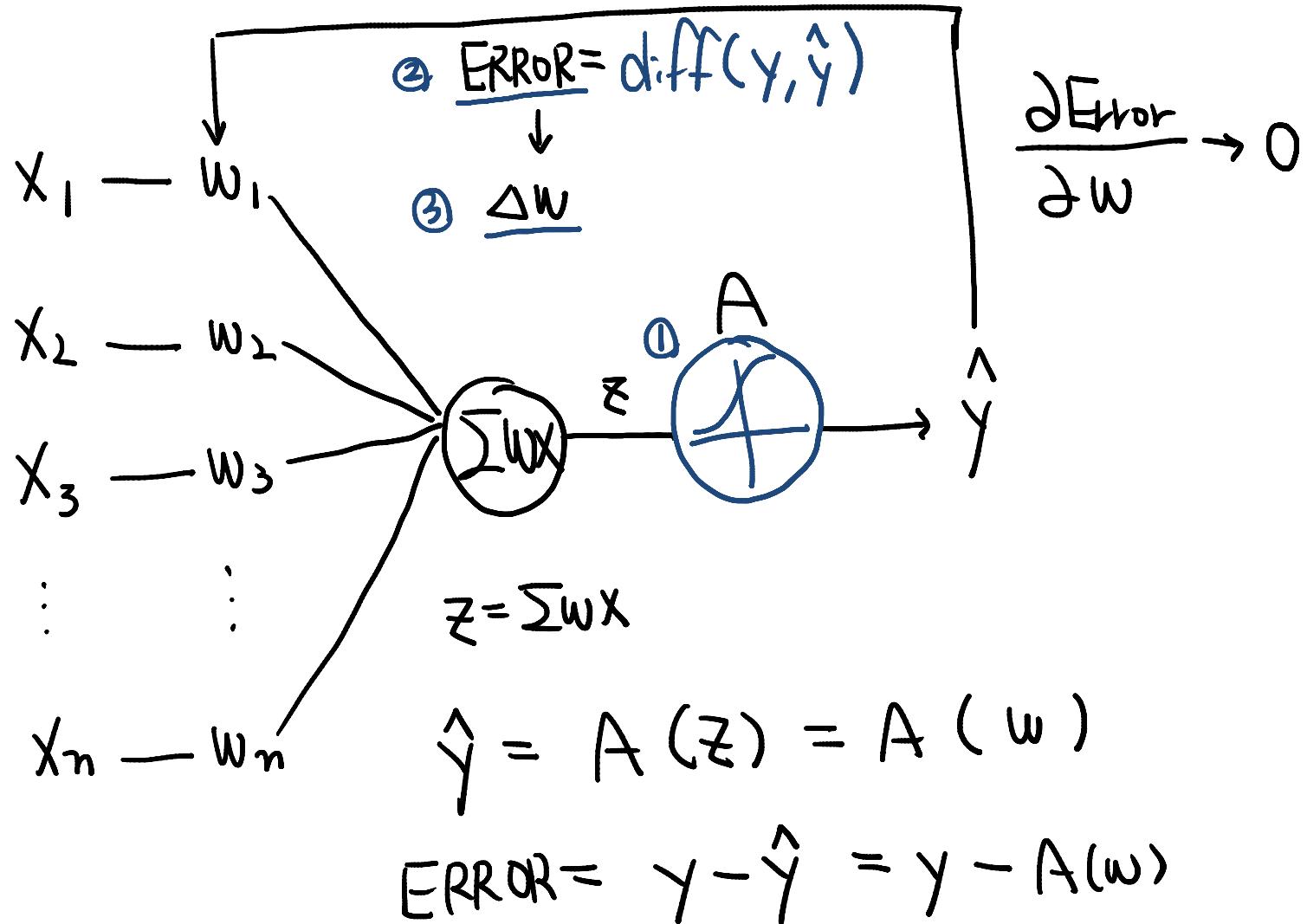
$$w_1 \neq 0, w_2 \neq 0$$

$$y = b$$



$$y = b + w_1 x_1$$







In [257]: `report['w2sum'].plot(logx=True, style='o--')`

...

In [259]: 예측확률 = model.predict_proba(X_test)

In [261]: DataFrame(예측확률)

Out[261]:

y

	0	1
--	---	---

0	0.995893	0.004107
---	----------	----------

1	0.000123	0.999877
---	----------	----------

2	0.272634	0.727366
---	----------	----------

3	0.337795	0.662205
---	----------	----------

4	0.999926	0.000074
---	----------	----------

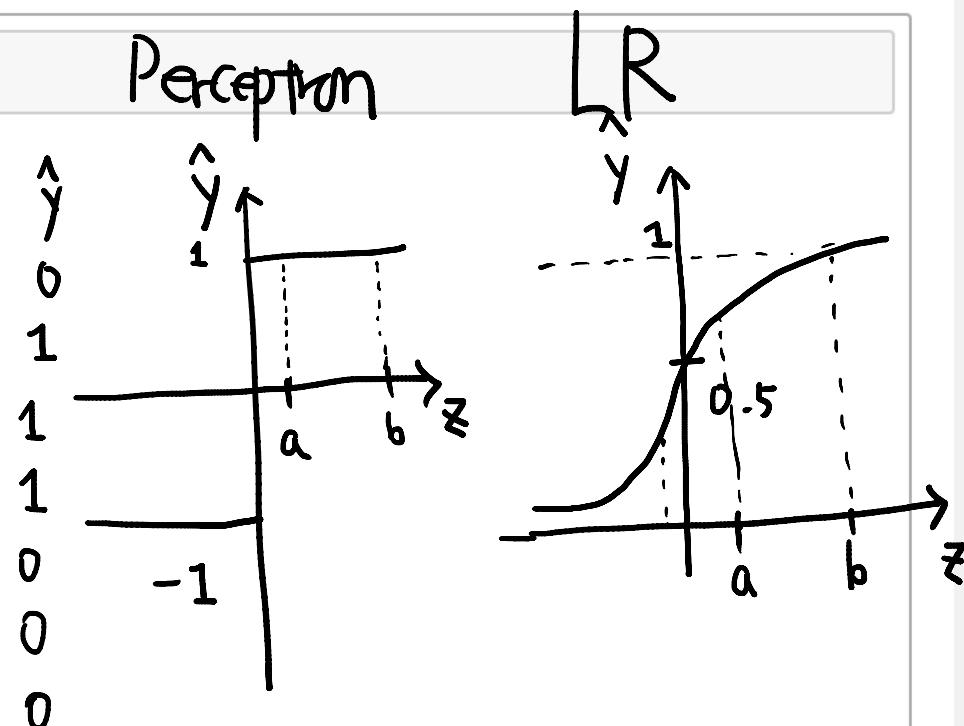
5	0.999981	0.000019
---	----------	----------

6	0.998488	0.001512
---	----------	----------

7	0.018170	0.981830
---	----------	----------

8	0.044919	0.955081
---	----------	----------

9	0.222227	0.777773
---	----------	----------





연습

퍼셉트론과 로지스틱 회귀 (LR; Logistic Regression) 모델을 같은 데이터로 훈련시키고 결과를 비교합니다.

```
In [271]: X_train, X_test, y_train, y_test = train_test_split(  
          X, y, test_size=0.3)
```

```
In [272]: model_ppn = Perceptron()
```

```
In [273]: model_ppn
```

```
Out[273]: Perceptron(alpha=0.0001, class_weight=None, eta0=1.0, fit_intercept  
          =True,  
          n_iter=5, n_jobs=1, penalty=None, random_state=0, shuffle=True,  
          verbose=0, warm_start=False)
```

*기본값으로
잘할수 있을까?*

```
In [ ]:
```

File Edit View Insert Cell Kernel Widgets Help



In [275]: model_ppn = Perceptron()
model_ppn

Out[275]: Perceptron(alpha=0.0001, class_weight=None, eta0=1.0, fit_intercept=True,
n_iter=5, n_jobs=1, penalty=None, random_state=0, shuffle=True,
verbose=0, warm_start=False)

In [274]: model_lr = LogisticRegression()
model_lr

Out[274]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
verbose=0, warm_start=False)

In []:

File Edit View Insert Cell Kernel Widgets Help

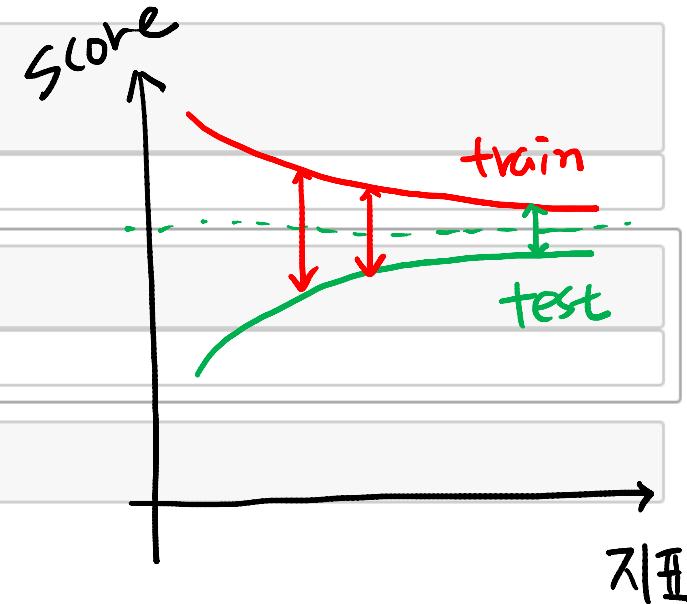


Out[287]: 0.87037037037035

✓ 1) 최적화

과적합 2) 일반화 ↓

In [274]: model_lr = LogisticRegression()
model_lr



In [278]: model_lr.fit(X_train, y_train)

In [279]: model_lr.score(X_test, y_test)

Out[279]: 0.9259259259259259 "수능"

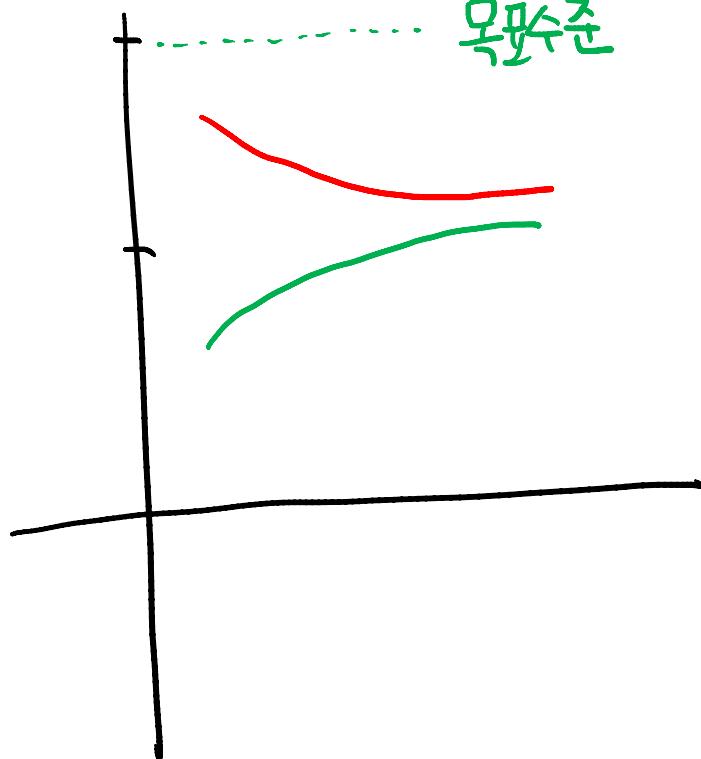
In [281]: model_lr.score(X_train, y_train)

Out[281]: 0.97580645161290325 "모의고사"

In []:

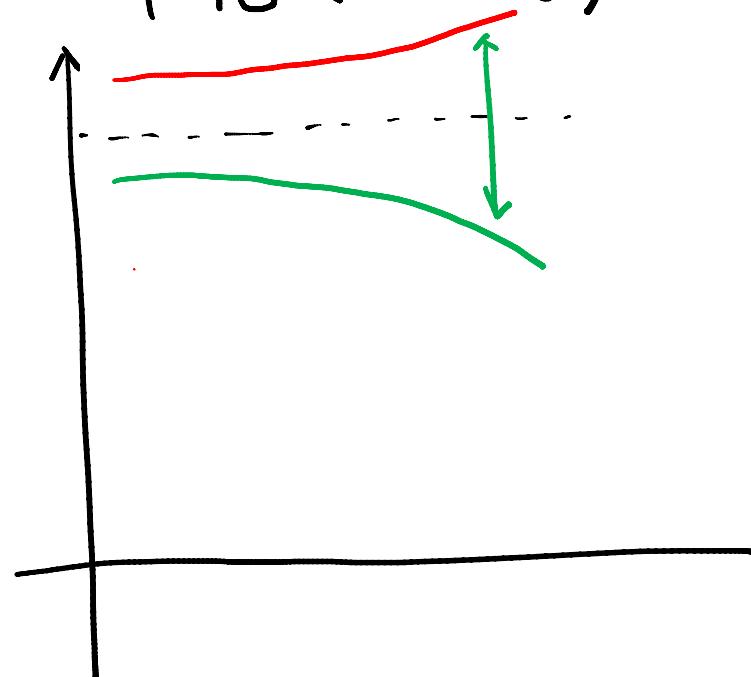
표현력

과소적합(Underfit)



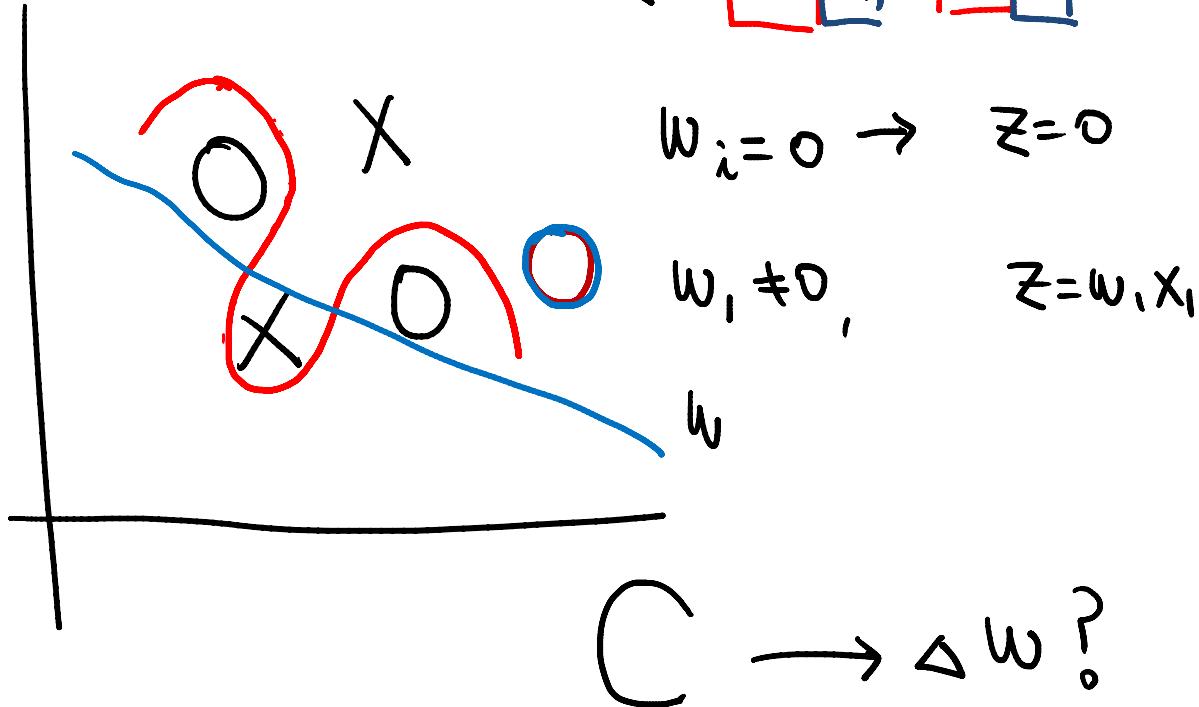
일반화 ~ 정규화 ~ C "군대"..... "하피"

과적합(Overfit)



과적합?

$$z = \boxed{w_1} \boxed{x_1} + \boxed{w_2} \boxed{x_2} + \dots$$





x의 문제 ...

$$\Delta w := \eta \times \text{Error} \times x_i$$

↑ 학습률

In [289]: X[:5]

Out [289]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflav ph
0	14.23	1.71	2.43	15.6	127	2.80	3.06	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	

In []:



Code



단위 조정 (Scaling)

In [290]: `from sklearn.preprocessing import MinMaxScaler`

$$\frac{X - \text{min}}{\text{Max} - \text{Min}}$$

In [291]: `scaler_min_max = MinMaxScaler()`In [292]: `X_mm = scaler_min_max.fit_transform(X)`

$$X \rightarrow \Delta W$$

In [295]: `DataFrame(X_mm, columns=X.columns)[5:]`

$$X_{mm} \rightarrow \Delta W$$

Out [295]:

	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids
0	0.842105	0.191700	0.572193	0.257732	0.619565	0.627586	0.573840
1	0.571053	0.205534	0.417112	0.030928	0.326087	0.575862	0.510549
2	0.560526	0.320158	0.700535	0.412371	0.336957	0.627586	0.611814
3	0.878947	0.239130	0.609626	0.319588	0.467391	0.989655	0.664557
4	0.581579	0.365613	0.807487	0.536082	0.521739	0.627586	0.495781

File Edit View Insert Cell Kernel Widgets Help



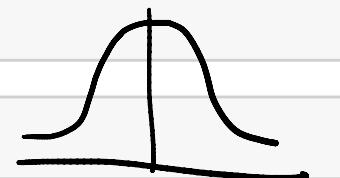
Code



In [296]: `from sklearn.preprocessing import StandardScaler`

$$\frac{x - \mu}{\sigma}$$

In [297]: `scaler = StandardScaler()`



In [298]: `X_std = scaler.fit_transform(X)`

In [299]: `DataFrame(X_std, columns=X.columns)[:5]`

Out [299]:

	Alcohol	Malic acid	Ash	Alkalinity of ash	Magnesium	Total phenols	Flavanoids
0	1.518613	-0.562250	0.232053	-1.169593	1.913905	0.808997	1.0348
1	0.246290	-0.499413	-0.827996	-2.490847	0.018145	0.568648	0.7336
2	0.196879	0.021231	1.109334	-0.268738	0.088358	0.808997	1.2155
3	1.691550	-0.346811	0.487926	-0.809251	0.930918	2.491446	1.4665
4	0.295700	0.227694	1.840403	0.451946	1.281985	0.808997	0.6633

In []: