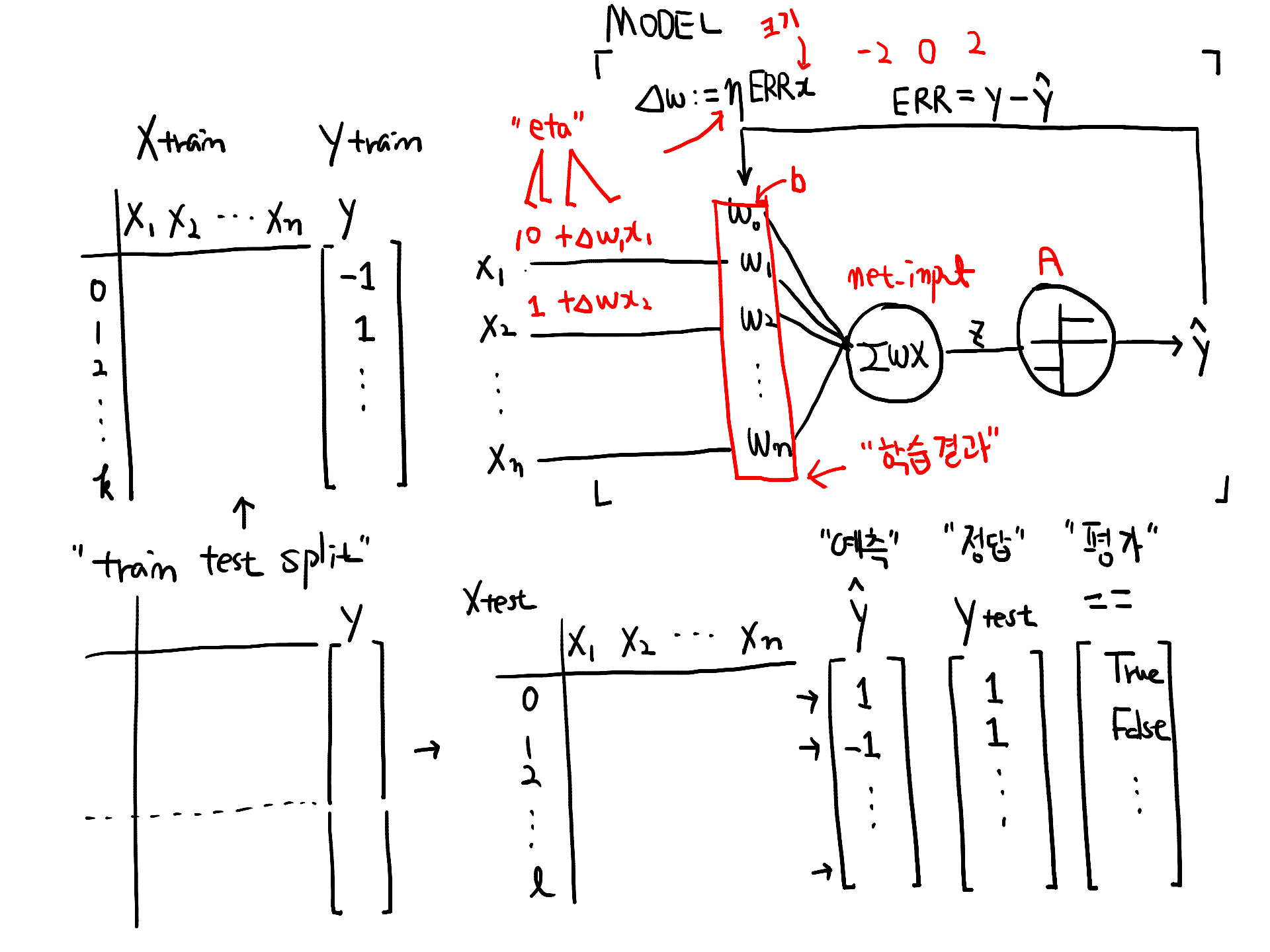
파이썬 딥러닝

이성주

seongjoo@codebasic.io


MODEL
초기
-2 0 2
Δw := η ERR x
ERR = y − ŷ
"eta"
b
10 + Δw₁x₁
1 + Δwx₂
x₁
x₂
xₙ
w₀
w₁
w₂
wₙ
net-input
Σwx
z
A
ŷ
"학습결과"
Xtrain
ytrain
X₁ X₂ ··· Xn
y
0
1
2
k
-1
1
"train test split"
y
Xtest
X₁ X₂ ··· Xn
0
1
2
l
"예측"
"정답"
"평가"
ŷ
ytest
==
1
-1
1
1
True
False

jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

```
Name: 4, dtype: int64
```

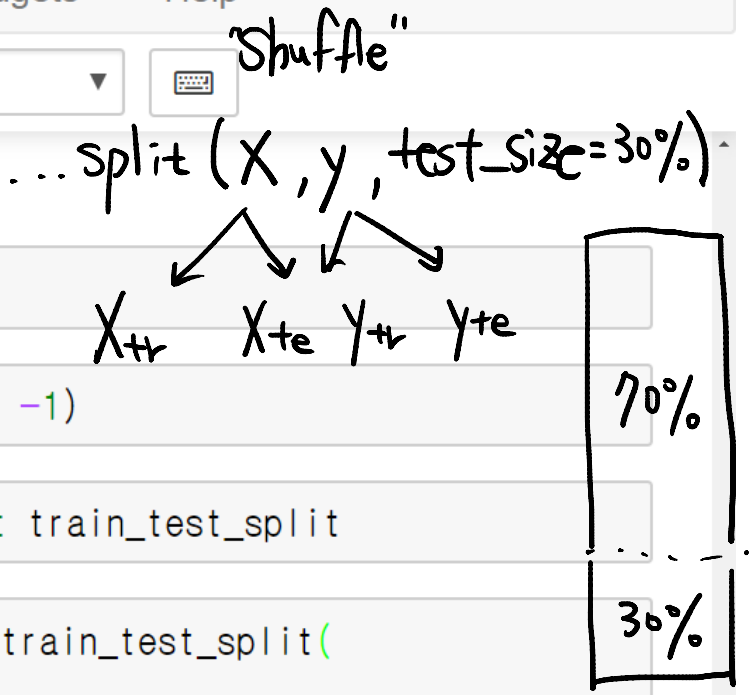
In [14]:
```
y = iris[:100][4]
```

In [16]:
```
y = np.where(y == 'Iris-setosa', 1, -1)
```

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3)
```

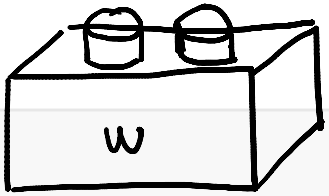
In [ ]:
```

```



jupyter

Day 3 상위 매개변수

Hyper Parameter → 매개변수

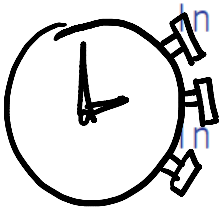


Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code



In [19]:
```
from sklearn.linear_model import Perceptron
```

In [20]:
```
model = Perceptron(n_iter=10, eta0=0.1)
```

In [21]:
```
model.fit(X_train, y_train)
```

$\Delta w := \eta \, Err \, x$

Out[21]:
```
Perceptron(alpha=0.0001, class_weight=None, eta0=0.1, fit_intercept
=True,
      n_iter=10, n_jobs=1, penalty=None, random_state=0, shuffle=Tr
ue,
      verbose=0, warm_start=False)
```

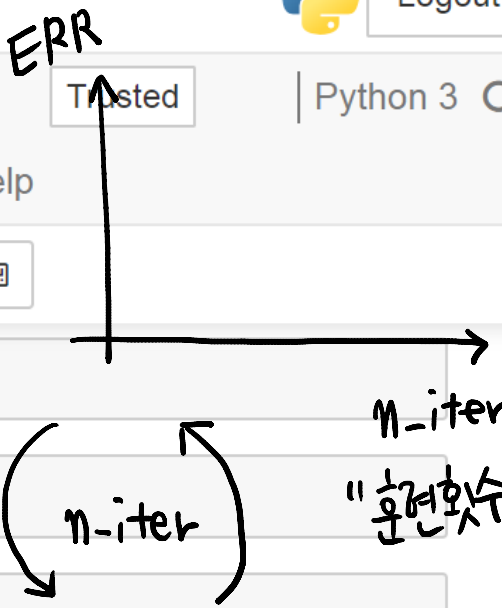
w → model.coef_, b → model.intercept_

In [24]:
```
model.coef_, model.intercept_
```

Out[24]:
```
(array([[ 0.01,  0.4 , -0.66, -0.31]]), array([ 0.1]))
```

w_1 w_2 w_3 w_4 $w_0 = b$

In [ ]:



jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

연습

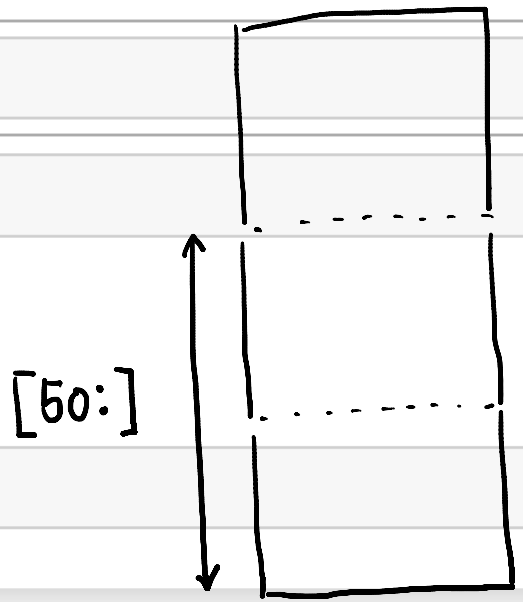
아래 구성된 data 대해 퍼셉트론 모델을 훈련하고 평가합니다.

In [38]:
```
data = iris[50:]
```

In [39]:
```
data[4].value_counts()
```

Out[39]:
```
Iris-versicolor    50
Iris-virginica     50
Name: 4, dtype: int64
```

In [ ]:



jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

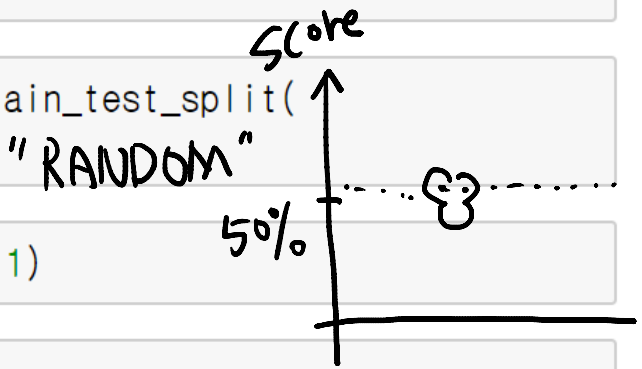
이진 분류

$y = \begin{cases} 1 \\ -1 \end{cases}$

In [42]:
```
X = data.loc[:, 0:3].values
```

In [45]:
```
y = np.where(data[4] == 'Iris-versicolor', 1, -1)
```

In [46]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3)
```



In [47]:
```
model = Perceptron(n_iter=10, eta0=0.1)
```

In [48]:
```
model.fit(X_train, y_train)
```

.predict(Xtest) == Ytest

왜?

In [49]:
```
score = model.score(X_test, y_test)
```

In [50]:
```
score
```

Out[50]:
```
0.73333333333333328
```

In [ ]:

jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

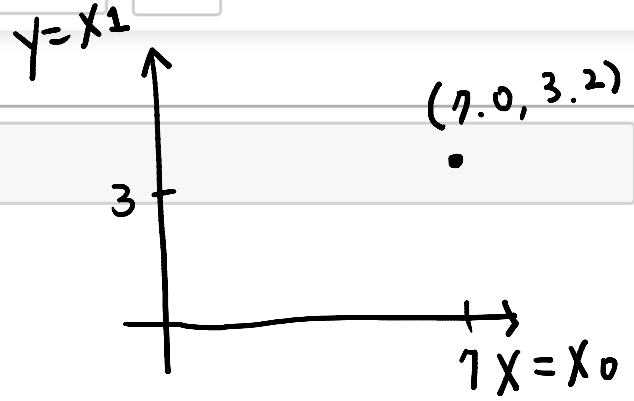
Code

Out[50]: 0.73333333333333328

In [51]:
```
data.plot(kind='scatter', x=0, y=1)
```

Out[51]:

	0	1	2	3	4
50	7.0	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
53	5.5	2.3	4.0	1.3	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor
56	6.3	3.3	4.7	1.6	Iris-versicolor
57	4.9	2.4	3.3	1.0	Iris-versicolor
58	6.6	2.9	4.6	1.3	Iris-versicolor
59	5.2	2.7	3.9	1.4	Iris-versicolor



jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

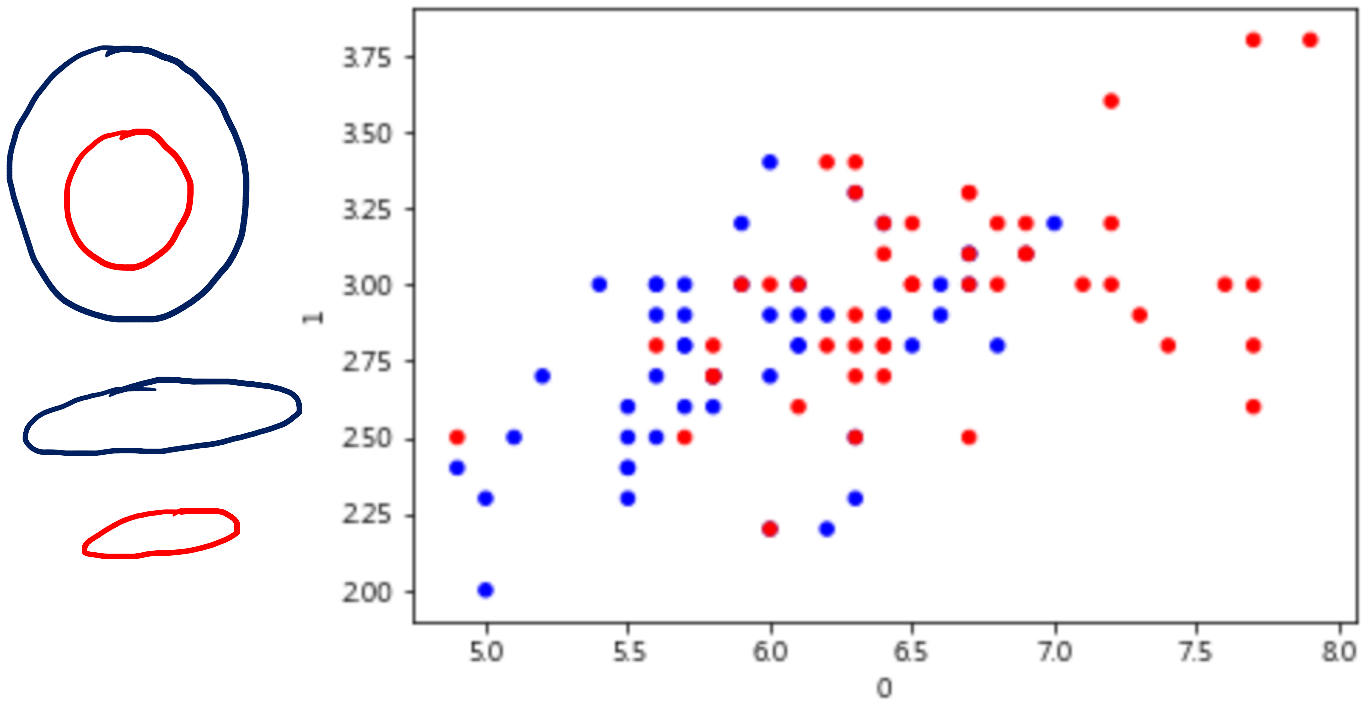
Code

X y c

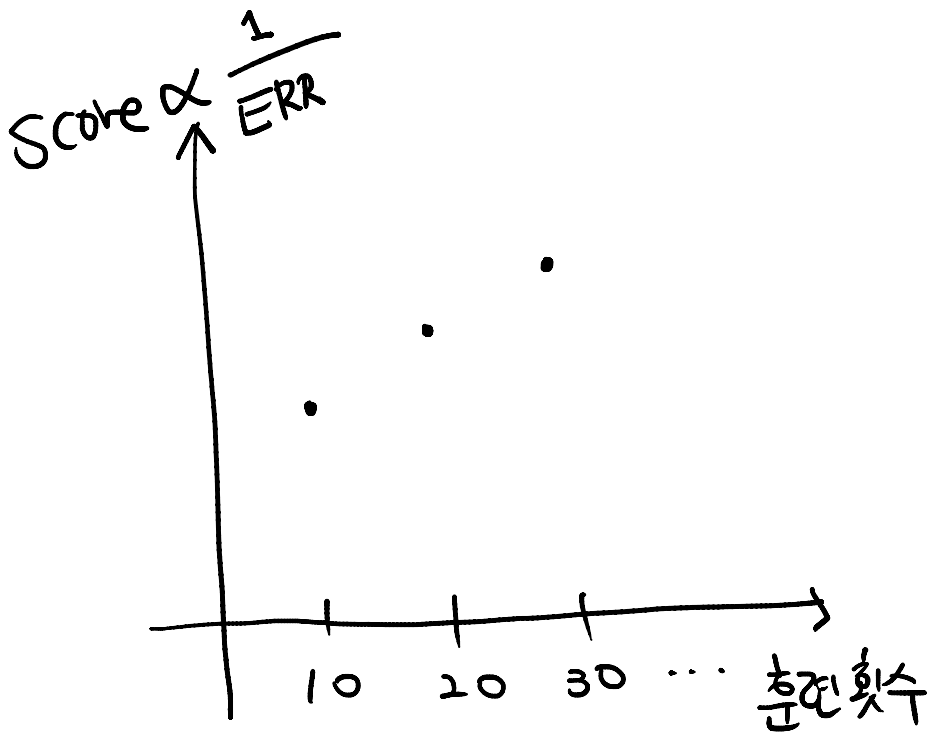
[ ] [1 -1 ⋮] ['r' 'b' ⋮]

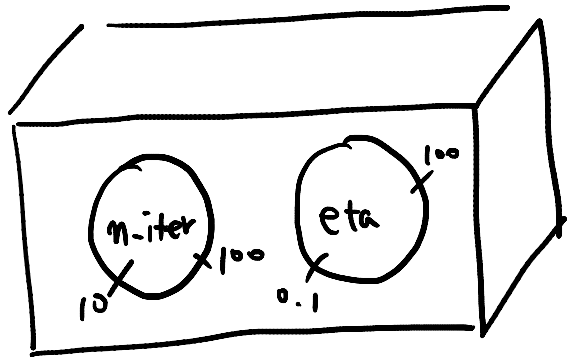
In [54]:
```
c = get_colormap(y, colors='rb')
data.plot(kind='scatter', x=0, y=1, c=c)
```

Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0xbc35630>



"비선형적 분포"



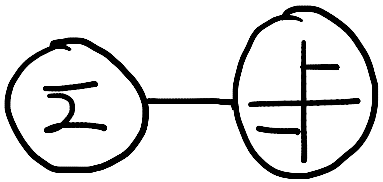


Hyper parameter Tuning

1958

Perceptron

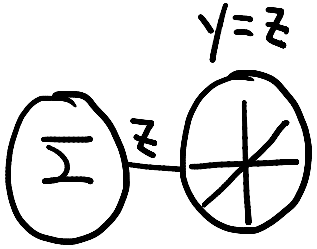
$\Delta w \leftarrow ERR$

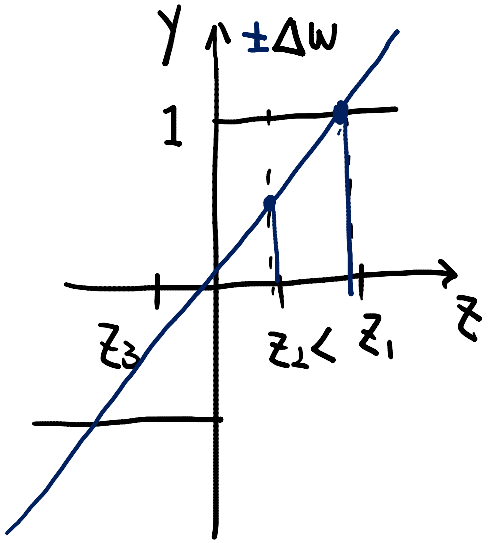


$$\frac{\partial ERR}{\partial w} \rightarrow 0$$

1960

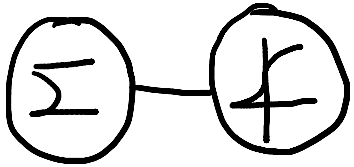
Adaline

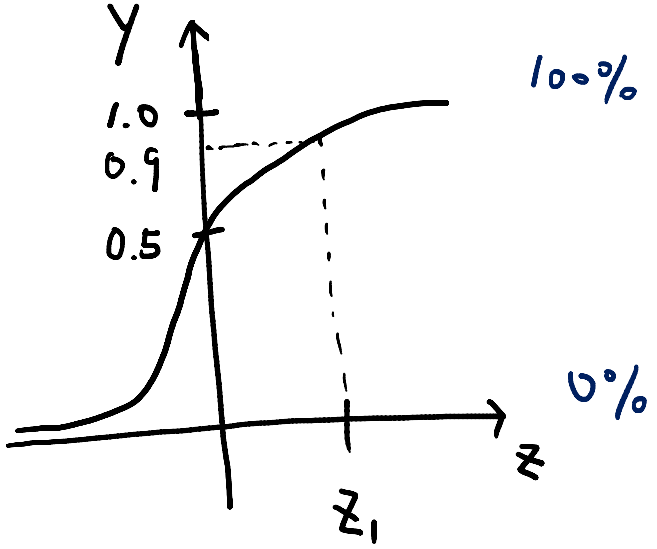




~~1962~~

Logistic Regression





jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [68]:
```
score = model.score(X_test, y_test)
```

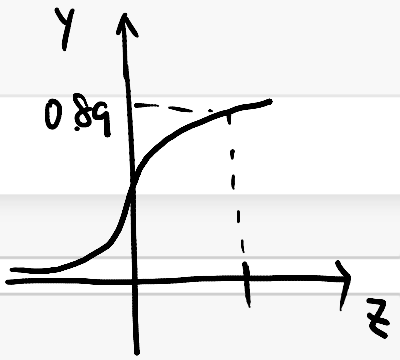
In [69]:
```
score
```

Out[69]: 0.96666666666666667

In [70]:
```
model.predict_proba(X_test)
```

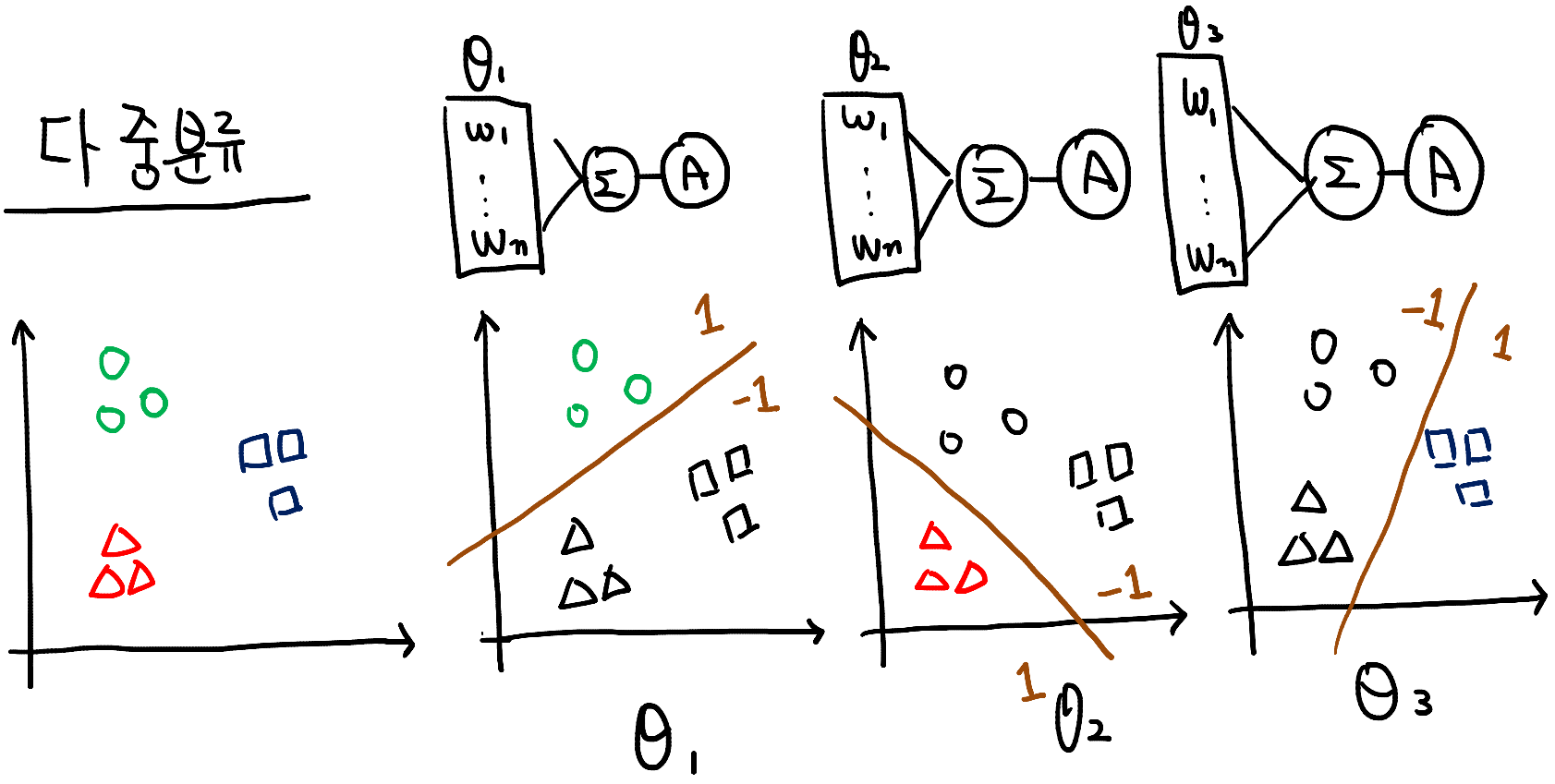
확률

Out[70]:
```
array([[ 0.11402482,  0.88597518],
       [ 0.77026774,  0.22973226],
       [ 0.12516969,  0.87483031],
       [ 0.90225135,  0.09774865],
       [ 0.51947142,  0.48052858],
       [ 0.22031136,  0.77968864],
       [ 0.75909506,  0.24090494],
       [ 0.56949791,  0.43050209],
       [ 0.46446273,  0.53553727],
       [ 0.90902316,  0.09097684],
       [ 0.9274843 ,  0.0725157 ],
       [ 0.30379255,  0.69620745],
       [ 0.53949166,  0.46050834],
       [ 0.89395922,  0.10604078],
```



.predict_proba()		.predict()
1	-1	ŷ
11%	89%	-1

다중분류



One-Vs-{ ALL → OvA
Rest → OvR

jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [81]:
```
score = model.score(X_test, y_test)
```

In [82]:
```
score
```

Out[82]: 0.9555555555555556

In [83]:
```
model.coef_
```

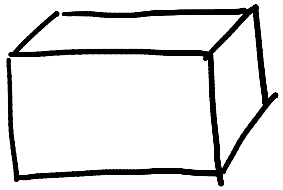
x_1 x_2 x_3 x_4

w_1 w_2 w_3 w_4

Out[83]:
```
array([[ 0.38607279,  1.35044459, -2.08715693, -0.94242481],
       [ 0.28235063, -1.4213786 ,  0.83819553, -1.7033084 ],
       [-1.43394126, -1.24905873,  1.93244322,  2.41635473]])
```

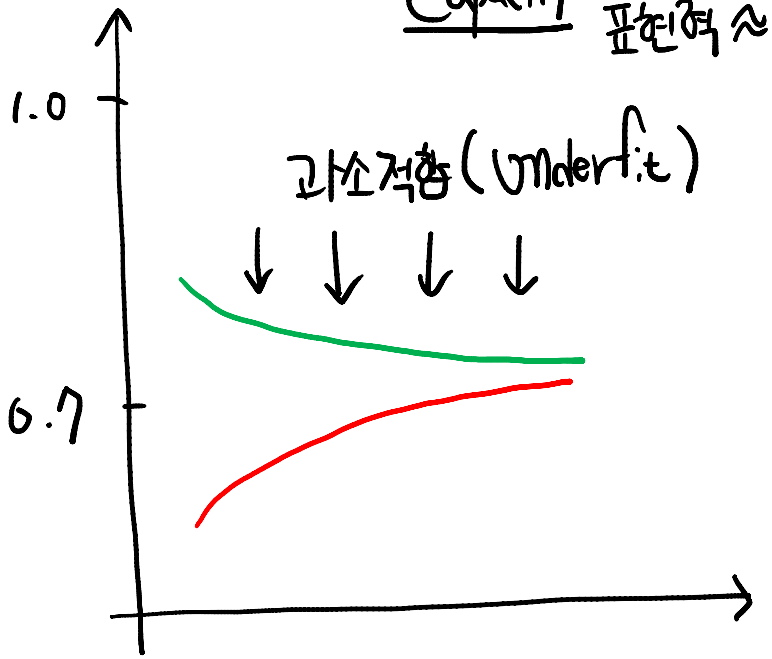
θ_1 θ_2 θ_3

In [ ]:
```

```

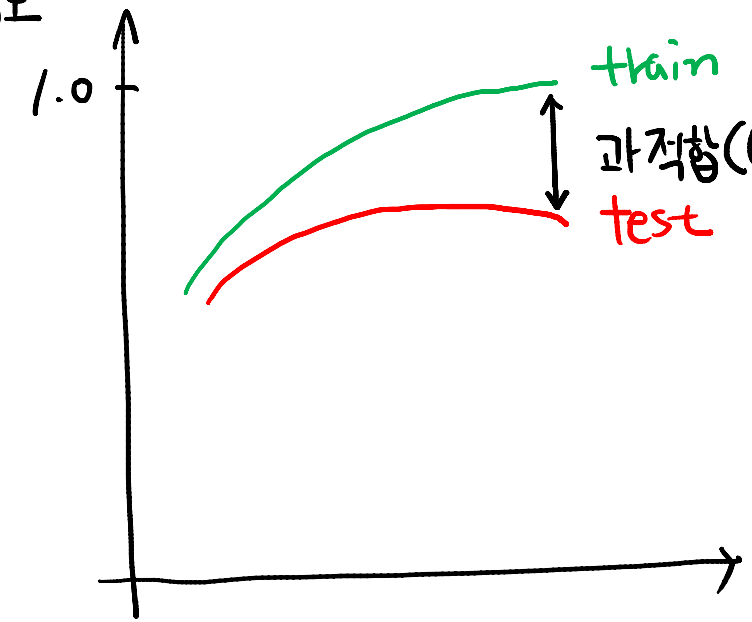


Capacity

표현력 ≈ 복잡도



"단순"



"정교"

일반성 ⇓

jupyter Day 3

Logout

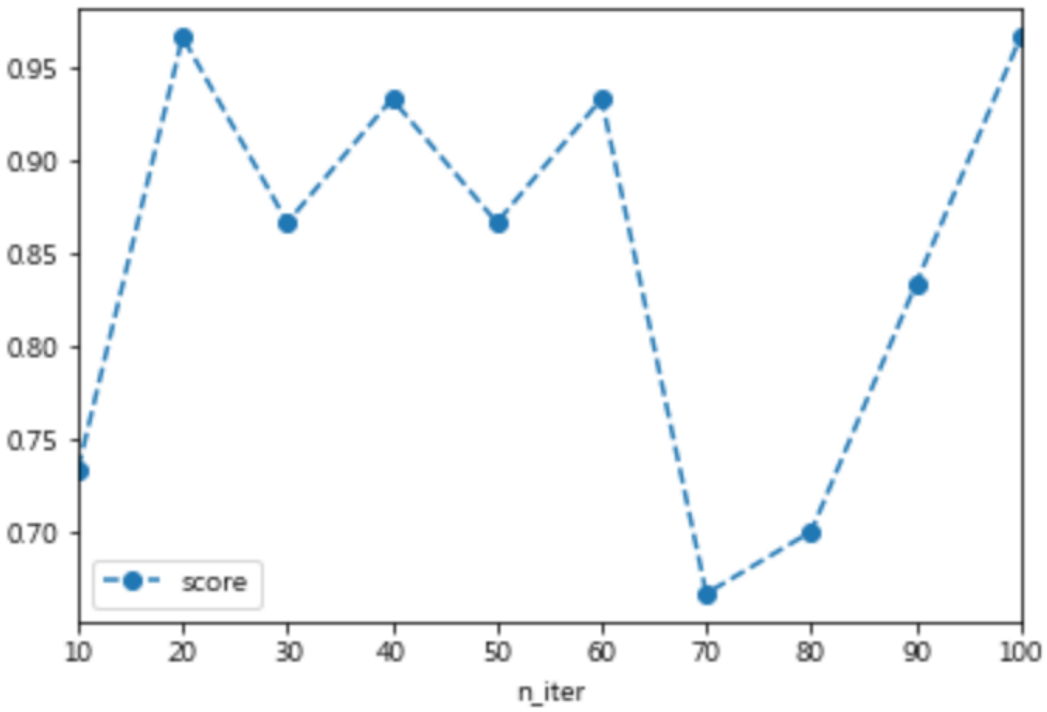
Trusted | Python 3

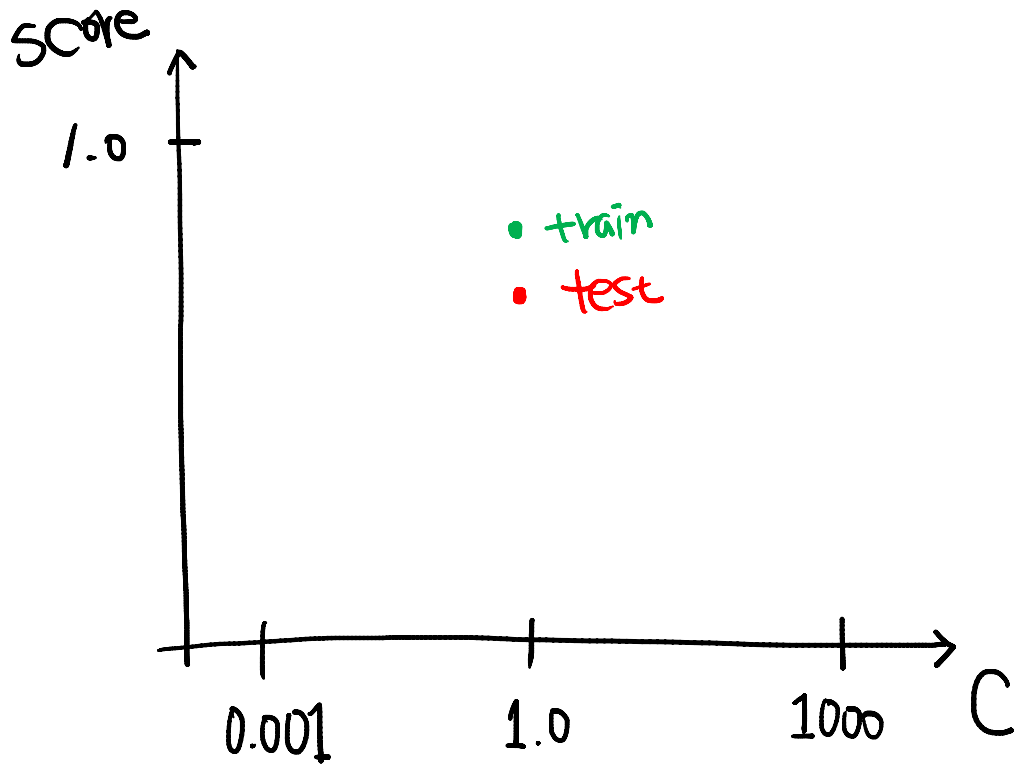
File Edit View Insert Cell Kernel Widgets Help

Code

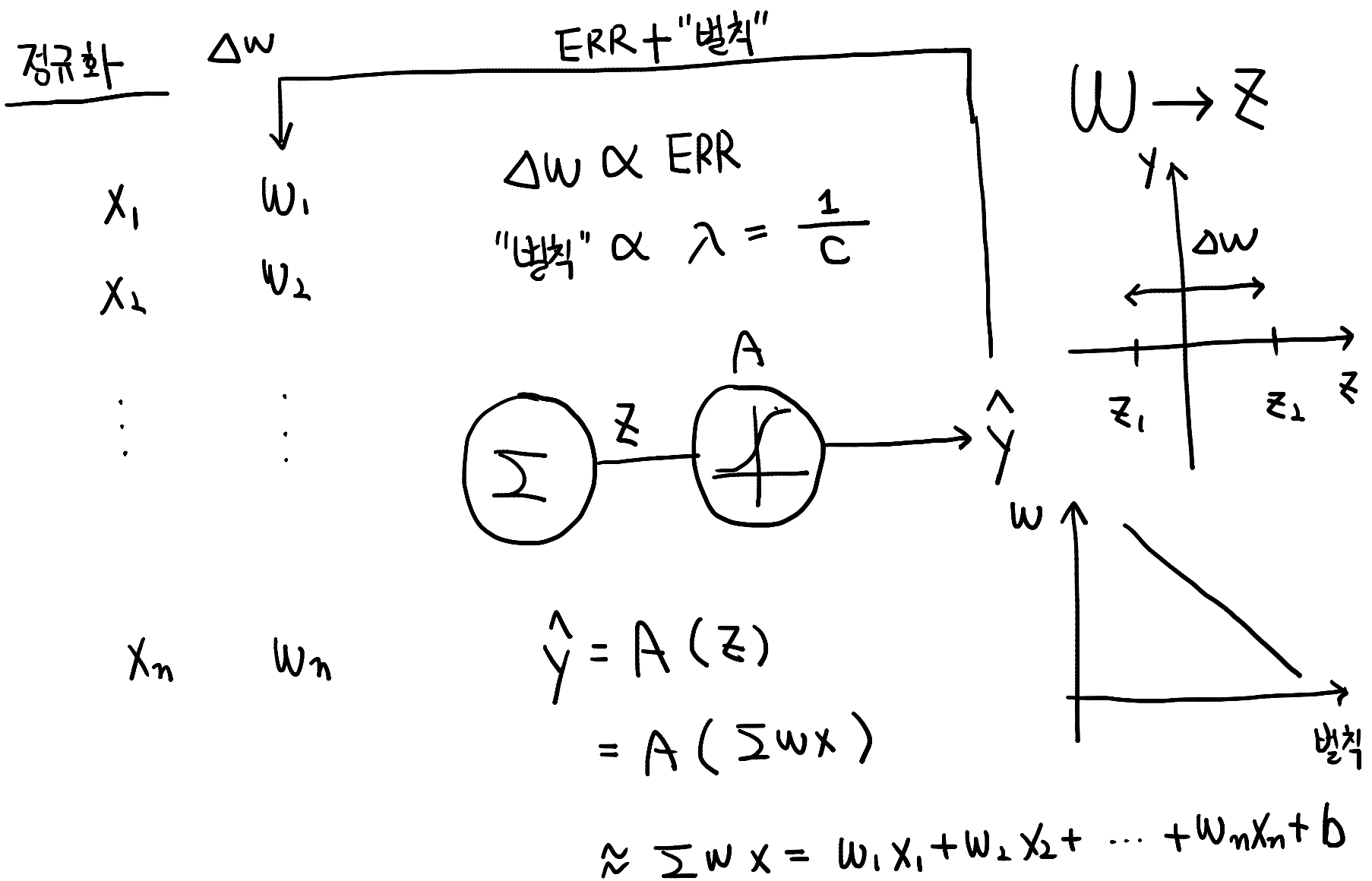
In [64]:
```
report.plot(style='o--')
```

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0xbde5438>




score
1.0
train
test
0.001
1.0
1000
C

정규화

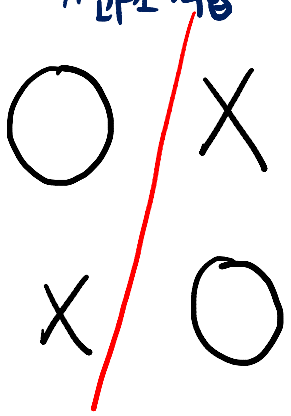

ΔW
ERR + "벌칙"
$\Delta w \propto ERR$
"벌칙" $\propto \lambda = \frac{1}{C}$
x_1 w_1
x_2 w_2
x_n w_n
Σ
Z
A
$\hat{y}$
$\hat{y} = A(z)$
$= A(\Sigma wx)$
$\approx \Sigma wx = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$
W → Z
y
Δw
z_1
z_2
z
w
벌칙

정규화 ~ 모델 표현력 ⇝ 일반성

$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$

$C \propto \Sigma w$

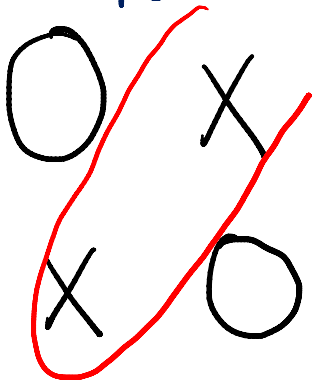
"과소적합"



$z = w_1 x_1 + b$

$C \propto \Sigma w$

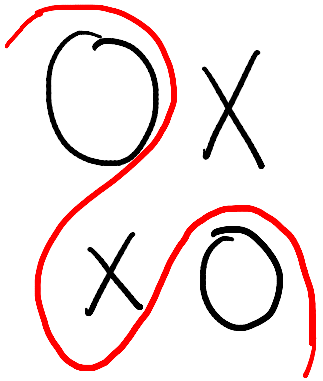
"적합"



$z = w_1 x_1 + w_2 x_2 + b$

$C \propto \Sigma w$

"과적합"



$z = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$

	x_1	x_2	x_3
	성씨	이름	성별
	이	성주	남
	김	성주	여
	박	성주	남
	최	성주	여

MODEL

$w_1 x_1$

$w_1 x_1 + w_2 x_2$

$w_1 x_1 + w_2 x_2 + w_3 x_3$

⋮

jupyter Day 3

$w_1 x_1 \rightarrow z$

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [75]: wine

$\Delta w_i := ERR \cdot x_i$

Out[75]:

x_1 x_2 ... x_5

	Class label	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69
5	1	14.20	1.76	2.45	15.2	112	3.27	3.39
6	1	14.39	1.87	2.45	14.6	96	2.50	2.52
7	1	14.06	2.15	2.61	17.6	121	2.60	2.51
8	1	14.83	1.64	2.17	14.0	97	2.80	2.98
9	1	13.86	1.35	2.27	16.0	98	2.98	3.15
10	1	14.10	2.16	2.30	18.0	105	2.95	3.32

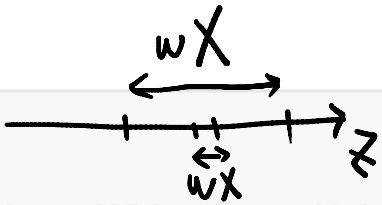
jupyter Day 3

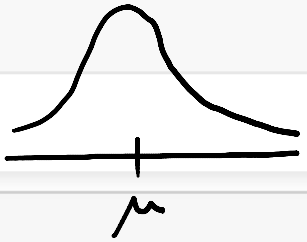
Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code





전처리

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

Before	After
$wX \rightarrow Z$	$wx \rightarrow z$

```
In [78]: from sklearn.preprocessing import StandardScaler
In [79]: scaler = StandardScaler()
In [80]: Xstd = scaler.fit_transform(X)
In [81]: DataFrame(Xstd)
```

Out[81]:

Alcohol

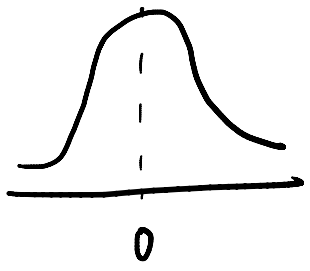
	0	1	2	3	4	5	
0	1.518613	-0.562250	0.232053	-1.169593	1.913905	0.808997	1.0348
1	0.246290	-0.499413	-0.827996	-2.490847	0.018145	0.568648	0.7336
2	0.196879	0.021231	1.109334	-0.268738	0.088358	0.808997	1.2155
3	1.691550	-0.346811	0.487926	-0.809251	0.930918	2.491446	1.4665
4	0.295700	0.227694	1.840403	0.451946	1.281985	0.808997	0.6633
5	1.481555	-0.517367	0.305159	-1.289707	0.860705	1.562093	1.3661
6	1.716255	-0.418624	0.305159	-1.469878	0.262708	0.328298	0.4926

$w_1 x_1 + w_2 x_2$

$0.1 \cdot (-1)$ $0.1 \cdot 1$

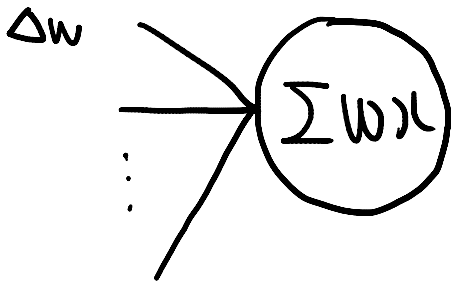
✓ Standard

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



MinMax

$$\frac{X - X_{min}}{X_{max} - X_{min}} > 0$$



$Z = \Sigma w x \leftarrow \Delta w = \eta ERR x$

$\sum_{i}^{n} (y_i - \hat{y}_i) +$ "벌칙"

$+ \Sigma |w|$

$+ \cdots$

$\Sigma |w|^2$

ΣW_x

Σwx

Model

H

H

$X_1 \cdots X_n$

$w X$

wx

ΔW

Δw

연습

스케일의 효과를 확인하기 위해, 스케일을 하지 않은 데이터와 스케일을 한 데이터에 대해 같은 모델, 같은 설정으로 훈련하고 평가합니다.

In [82]:
```
X_tr, X_te, Xstd_tr, Xstd_te, y_tr, y_te = train_test_split(
    X, Xstd, y, test_size=0.3)
```

In [ ]:

In [ ]:

jupyter Day 3

[(C, train, test)
(0.01, 0.97, 0.98),
(0.1, 0.98, 0.99),
⋮

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [87]:
```
results = {'raw': [], 'std': []}
for C in [0.01, 0.1, 1., 10., 100., 1000.]:
    model = LogisticRegression(C=C)
    model.fit(X_tr, y_tr)
    results['raw'].append((
        C, model.score(X_tr, y_tr), model.score(X_te, y_te)))

    model.fit(Xstd_tr, y_tr)
    results['std'].append((
        C, model.score(Xstd_tr, y_tr), model.score(Xstd_te, y_te)))
```

In [90]:
```
DataFrame(results['raw'])
```

...

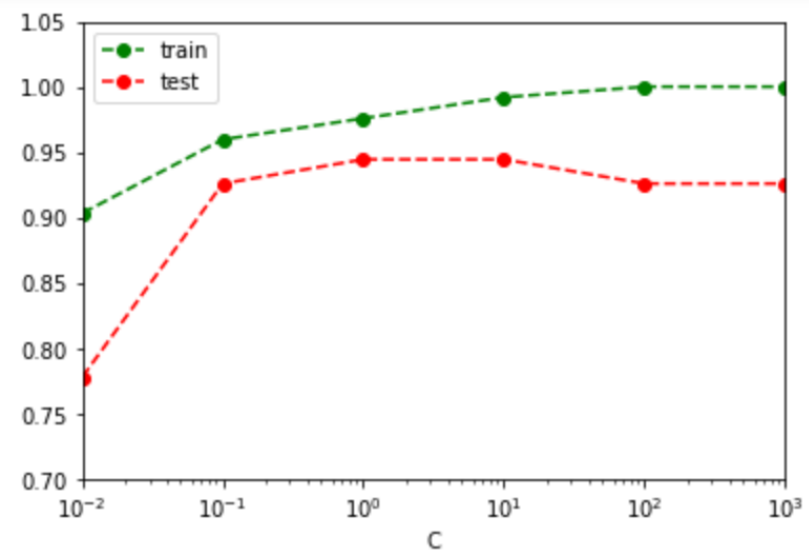
In [91]:
```
DataFrame(results['std'])
```

Out[91]:

	0	1	2
0	0.01	0.975806	0.981481
1	0.10	0.991935	1.000000

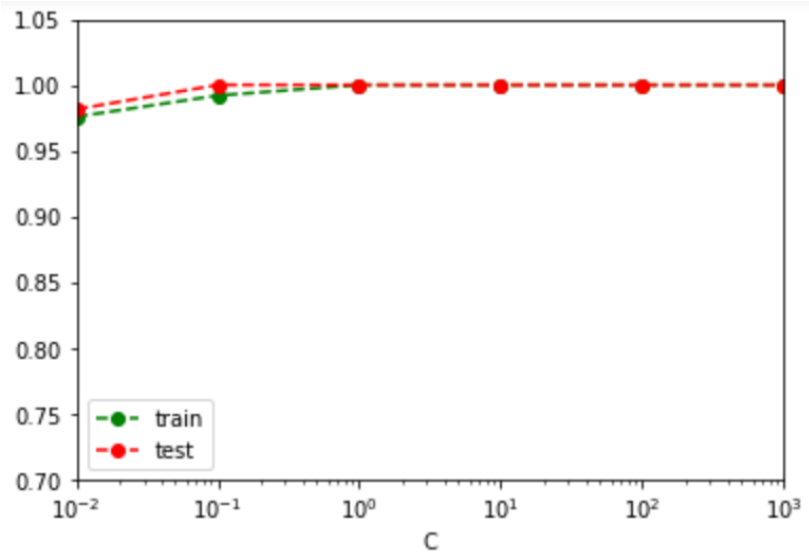
$\Sigma w X$

raw

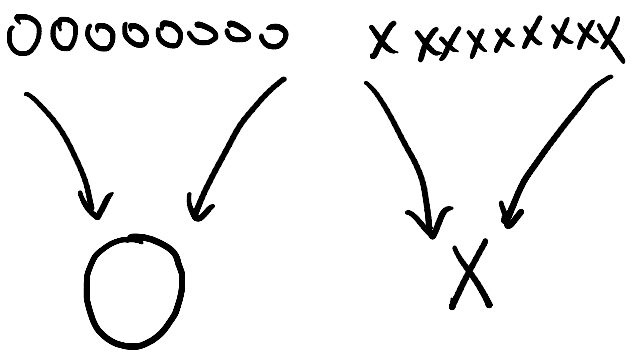


$\Sigma w x$

STD



차원축소



$X_1 \; X_2 \; \cdots \; X_n$

$\downarrow \; \downarrow \qquad \downarrow$

$W_1 \; W_2 \qquad W_n$

$X_1' \; X_2' \; \cdots \; X_d' \quad (d \ll n)$

$W_1' \; W_2' \; \cdots \; W_d'$

jupyter Day 3

지도학습 | 비지도학습

X, y | X, ?

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [103]:
```
from sklearn.decomposition import PCA
```

In [104]:
```
pca = PCA(n_components=2)
```

In [105]:
```
Xpca = pca.fit_transform(X)
```

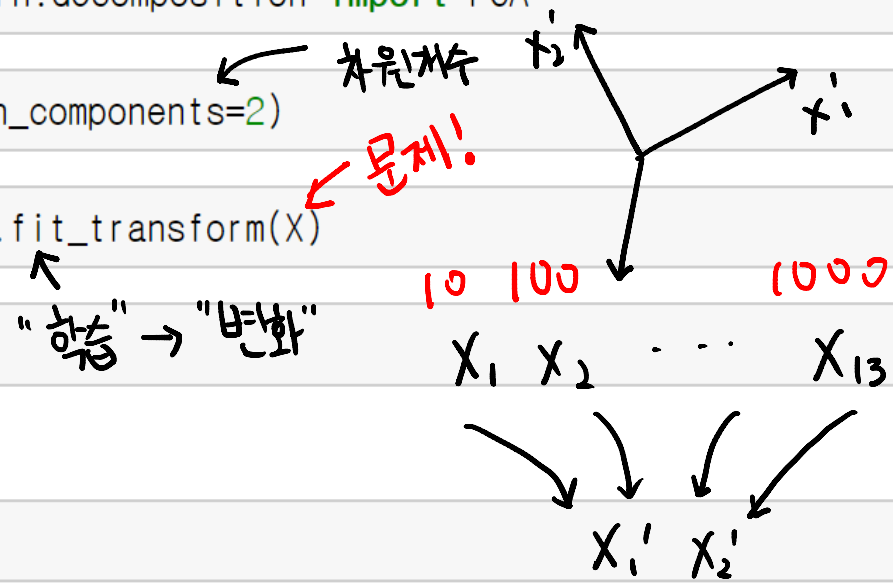
In [106]:
```
X.shape
```

Out[106]: (178, 13)

In [107]:
```
Xpca.shape
```

Out[107]: (178, 2)

In [ ]:



jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [116]:
```
X_tr, X_te, Xpca_tr, Xpca_te, y_tr, y_te = train_test_split(
    X, Xpca, y, test_size=0.3)
```

In [117]:
```
model = LogisticRegression(C=1.0)
```

$w_1 x_1 + w_2 x_2 + \cdots + w_{13} x_{13}$

In [118]:
```
model.fit(X_tr, y_tr).score(X_te, y_te)
```

$n = 13$

Out[118]: 0.9259259259259259

$w_1 x_1' + w_2 x_2'$

In [119]:
```
model.fit(Xpca_tr, y_tr).score(Xpca_te, y_te)
```

$d = 2$

Out[119]: 0.9444444444444442

In [ ]:

jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code

In [120]:
```
pca = PCA(n_components=None)
```
13 → 13

In [121]:
```
pca.fit(Xstd)
```
"학습(탐색)"

θ

$Y = \begin{cases} 1 \\ -1 \end{cases}$

In [123]:
```
기여도 = pca.explained_variance_ratio_
```

In [125]:
```
Series(기여도, index=wine.columns[1:]).plot(kind='barh')
```

Out[125]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xe547cc0>
```

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

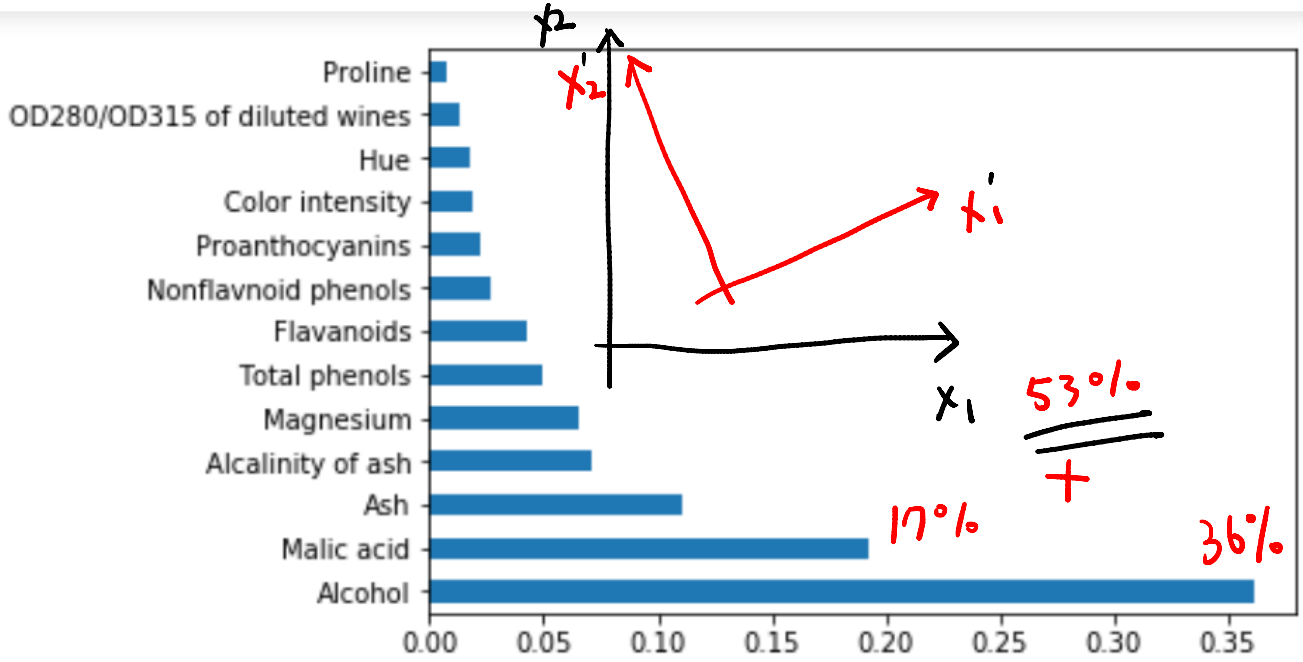
jupyter Day 3

Logout

Trusted | Python 3

File Edit View Insert Cell Kernel Widgets Help

Code



In [ ]: