4.6 로지스틱 회귀

핸즈온 머신러닝 2판
- 오렐리안 제롱

화학소재솔루션센터
김민근

CAUTION_ 이 장에는 기초적인 선형대수와 미분 기호를 사용한 수학 방정식이 꽤 나옵니다. 이 식들을 이해하려면 벡터와 행렬, 전치transpose, 점곱, 역행렬inverse matrix, 편미분partial derivative에 대해 알아야 합니다. 이 개념들이 익숙하지 않다면 주피터 노트북으로 만든 선형대수와 미분에 대한 기초 튜토리얼을 깃허브(https://github.com/rickiepark/handson-ml2)에서 살펴보세요. 정말 수학이 싫다면 이 장을 읽되 방정식은 건너뛰세요. 본문만으로도 대부분의 개념을 이해하는 데 충분히 도움 될 것입니다.

지도학습: Supervised learning

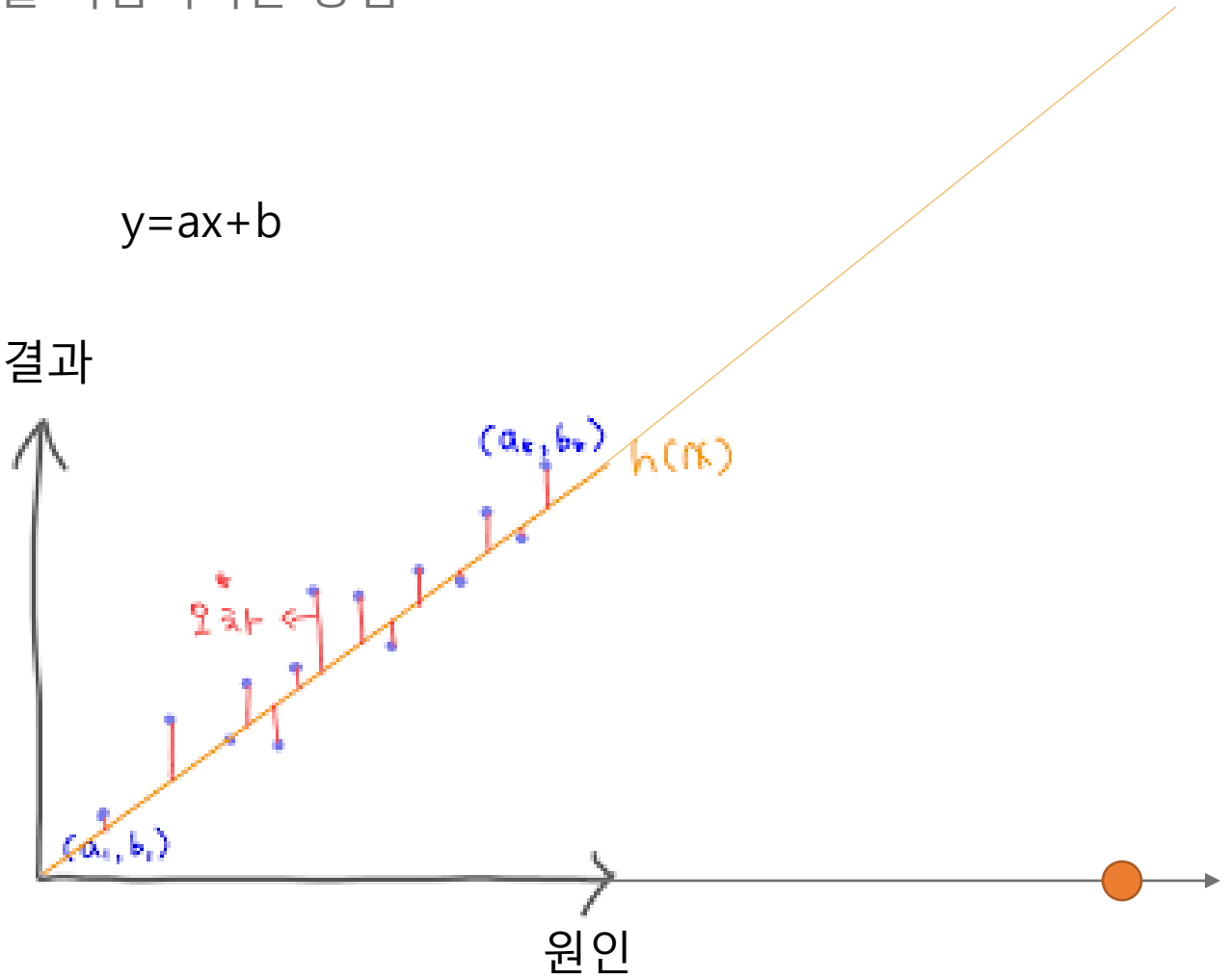
정답이 있는 데이터로 인공지능을 학습시키는 방법

회귀(regression)

지도학습

- k-최근접 이웃
  k-nearest neighbors
- 선형회귀
  linear regression
- 로지스틱 회귀
  logistic regression
- 서포트 벡터 머신
  support vector machine
- 결정 트리와 랜덤 포레스트
  decision tree & random forest
- 신경망
  neural networks

y=ax+b



KRICT 한국화학연구원

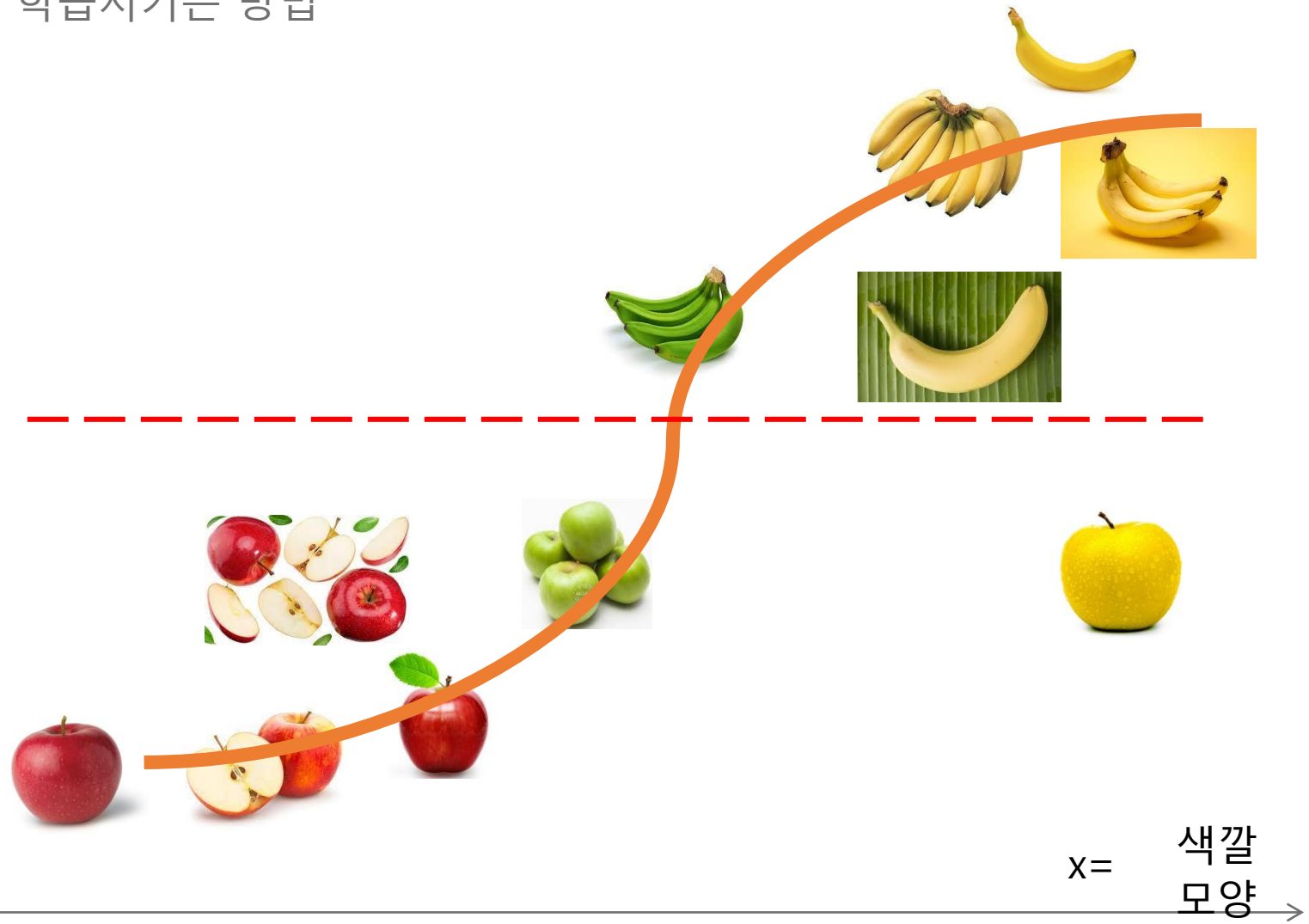
지도학습: Supervised learning

정답이 있는 데이터로 인공지능을 학습시키는 방법

회귀(regression)

지도학습

- k-최근접 이웃
  k-nearest neighbors
- 선형회귀
  linear regression
- 로지스틱 회귀
  logistic regression
- 서포트 벡터 머신
  support vector machine
- 결정 트리와 랜덤 포레스트
  decision tree & random forest
- 신경망
  neural networks

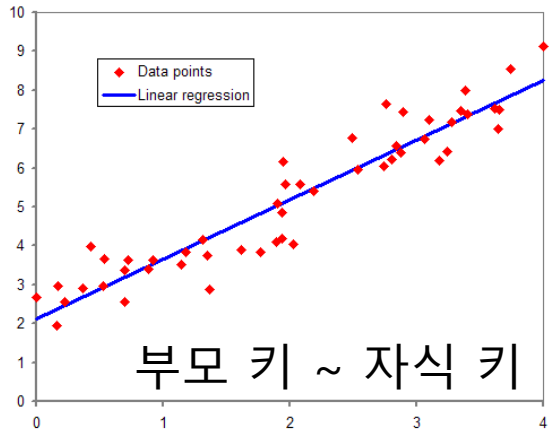


KRICT 한국화학연구원

~~로지스틱~~ 회귀: ~~Logistic~~ regression

역사 [편집]

회귀(영어: regress 리그레스[*])의 원래 의미는 옛날 상태로 돌아가는 것을 의미한다. 영국의 유전학자 프랜시스 골턴은 부모의 키와 아이들의 키 사이의 연관 관계를 연구하면서 부모와 자녀의 키사이에는 선형적인 관계가 있고 키가 커지거나 작아지는 것보다는 전체 키 평균으로 돌아가려는 경향이 있다는 가설을 세웠으며 이를 분석하는 방법을 "회귀분석"이라고 하였다. 이러한 경험적 연구 이후, 칼 피어슨은 아버지와 아들의 키를 조사한 결과를 바탕으로 함수 관계를 도출하여 회귀분석 이론을 수학적으로 정립하였다.

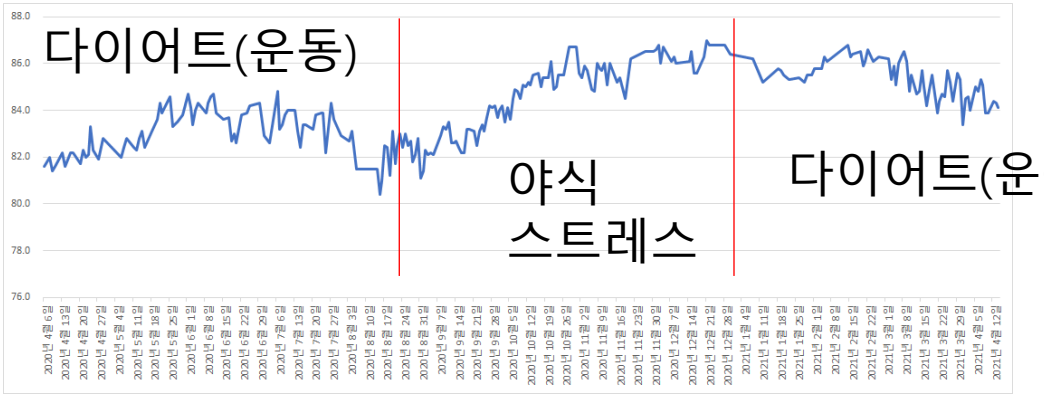


Y = 몸무게

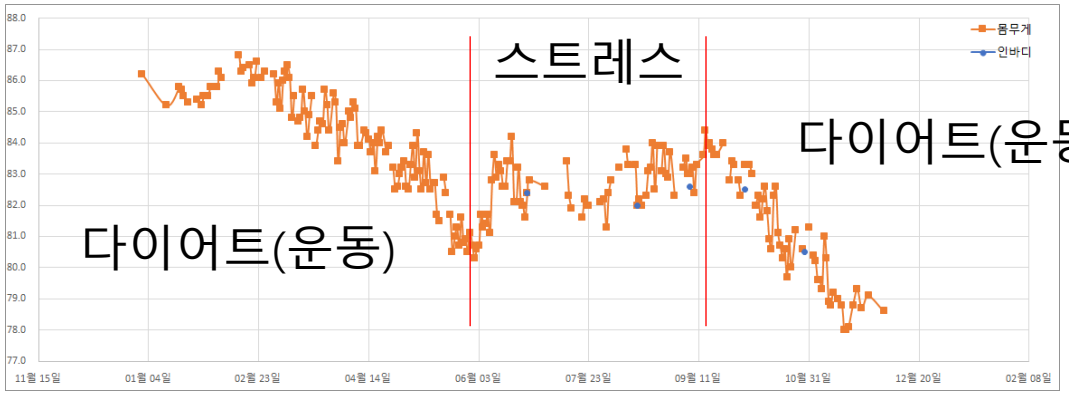
X1=운동
X2=야식
X3=스트레스

Sum_??(X1,X2,X3) > Y(?)

2021: 2학기 졸업준비



2022

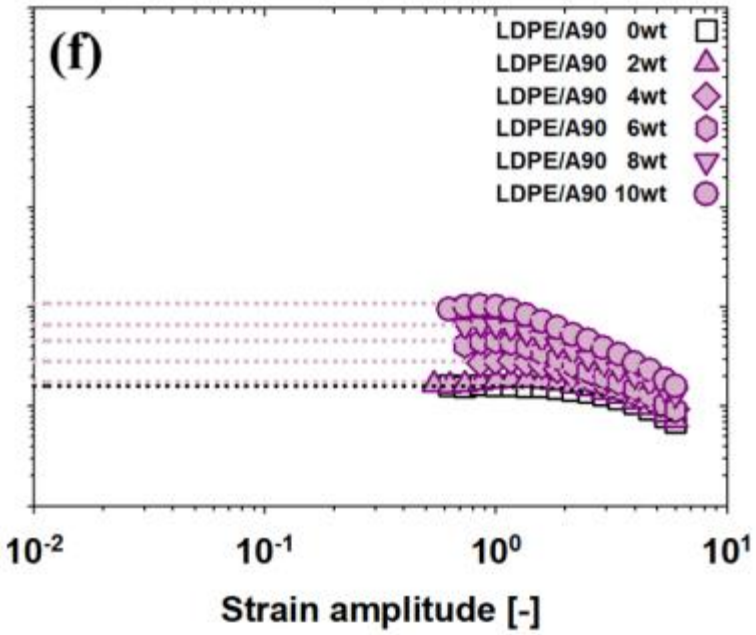


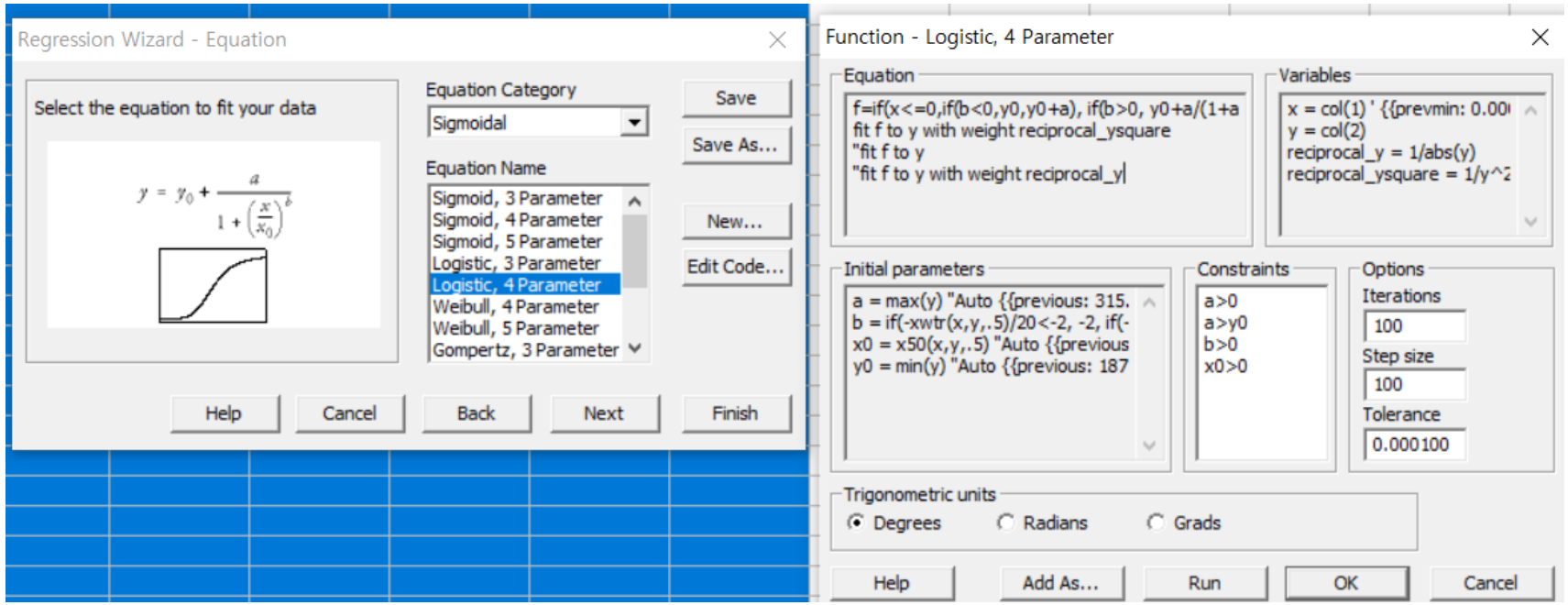
https://ko.wikipedia.org/wiki/%ED%9A%8C%EA%B7%80_%EB%B6%84%EC%84%9D

KRICT 한국화학연구원

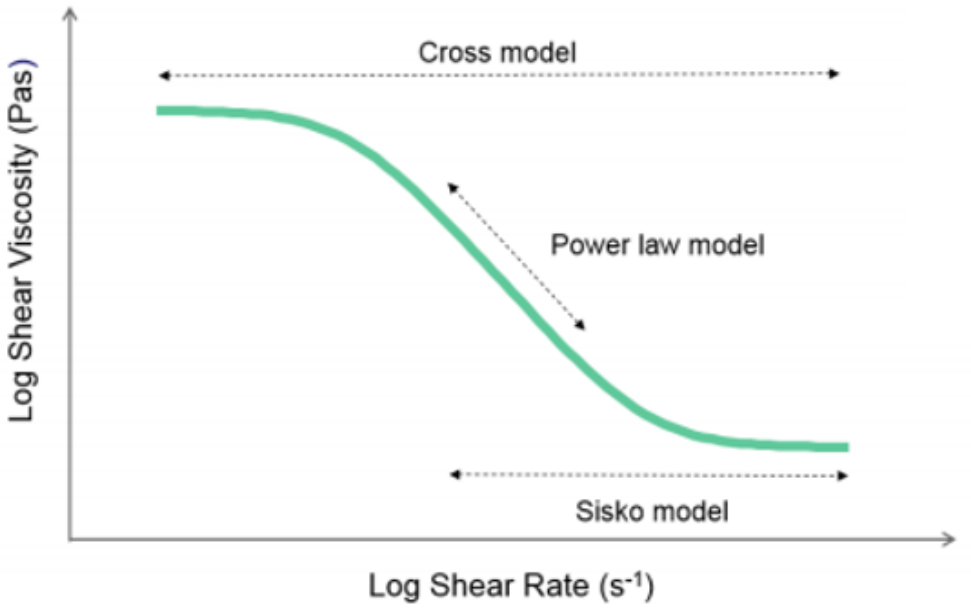
실험 데이터 Fitting

$$Q = Q_0(1 + (C_1\gamma_0)^{C_2})^{(C_3-1)/C_2}$$





실험 데이터 Fitting



Cross model $\dfrac{\eta-\eta_\infty}{\eta_o-\eta_\infty}=\dfrac{1}{1+(K\dot{\gamma})^n}$

Power law model $\sigma=k\dot{\gamma}^n$

Sisko model $\sigma=k\dot{\gamma}^n+\eta_\infty\dot{\gamma}$

Carreau-Yasuda Model : $\eta=\dfrac{\eta_o}{[1+(\lambda\dot{\gamma})^a]^{\frac{(1-n)}{a}}}$

$\eta=\mu$	Newtonian
$\eta=\mu\lvert\lambda\dot{\gamma}\rvert^{n-1}$	Ostwald-deWaele
$\eta=\begin{cases}\mu, & \lvert\lambda\dot{\gamma}\rvert\le 1\\ \mu\lvert\lambda\dot{\gamma}\rvert^{n-1}, & \lvert\lambda\dot{\gamma}\rvert\ge 1\end{cases}$	Spriggs
$\eta=\mu_2+\dfrac{\mu-\mu_2}{1+\lvert\lambda\dot{\gamma}\rvert^{1-n}}$	Cross
$\eta=\mu_2+\dfrac{\mu-\mu_2}{\left[1+(\lambda\dot{\gamma})^2\right]^{\frac{1-n}{2}}}$	Carreau
$\eta=\mu_2+\dfrac{\mu-\mu_2}{\left[1+(\lambda\dot{\gamma})^a\right]^{\frac{1-n}{a}}}$	Carreau-Yasuda
$\eta=\sum_{i=1}^{N}\dfrac{f_i\mu}{\lambda_i\dot{\gamma}}\sinh^{-1}(\lambda_i\dot{\gamma})$ $\sum_{i=1}^{N}f_i=1,\quad N>1$	Ree-Eyring

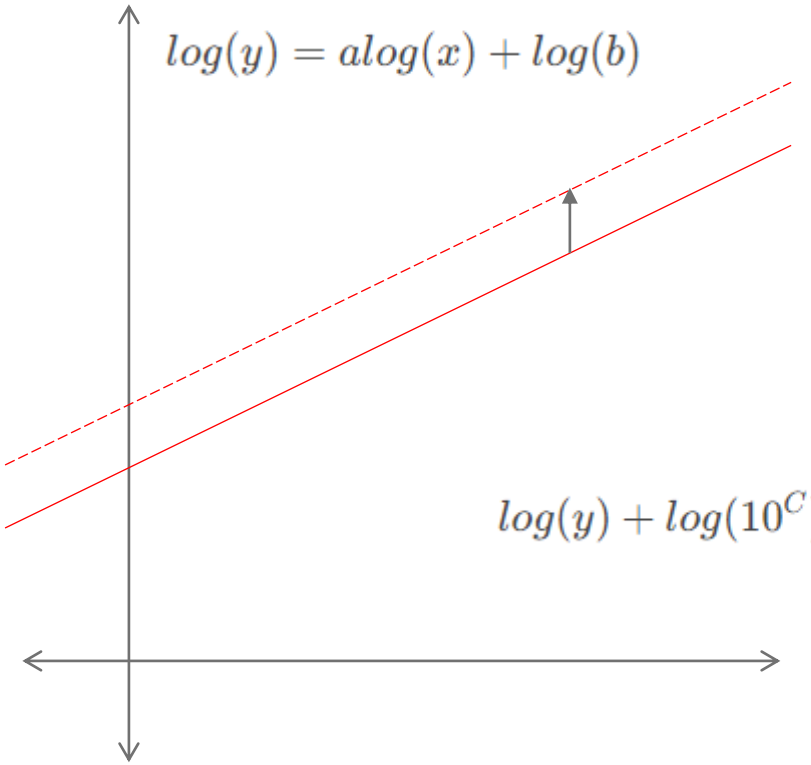
실험 데이터 Fitting

$$y = ax + b$$

$$log(y) = alog(x) + log(b)$$

$$y + c$$

$$log(y) + C$$



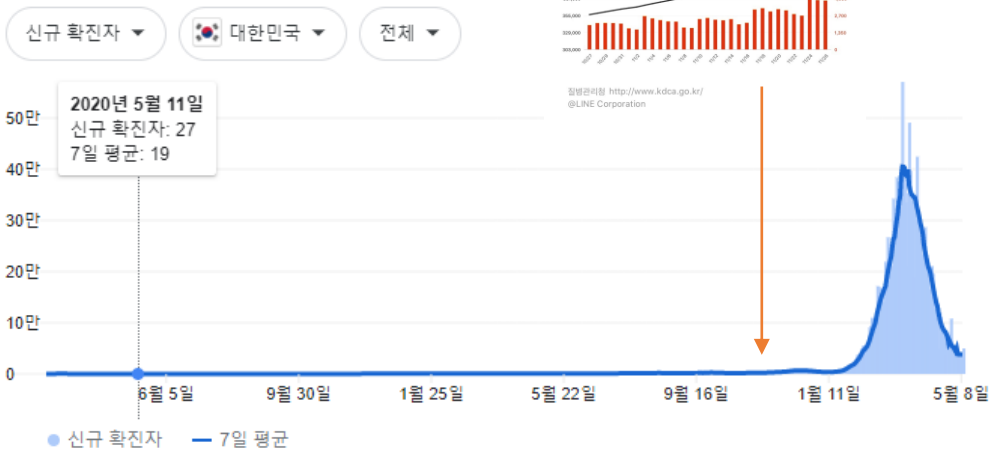
$$log(y) + log(10^C) = log(y * 10^C)$$

코로나바이러스 감염증(COVID-19)

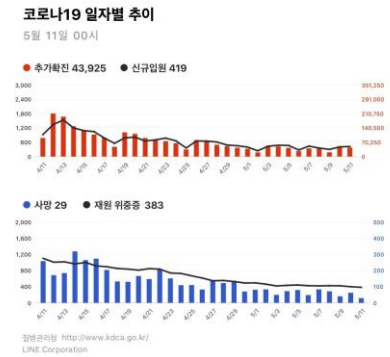
통계

신규 확진자 및 사망자

출처: JHU CSSE COVID-19 Data · 최종 업데이트: 1일 전



각 날짜에는 전날 이후 보고된 신규 확진자가 표시됩니다. · 데이터 정보



KRICT 한국화학연구원

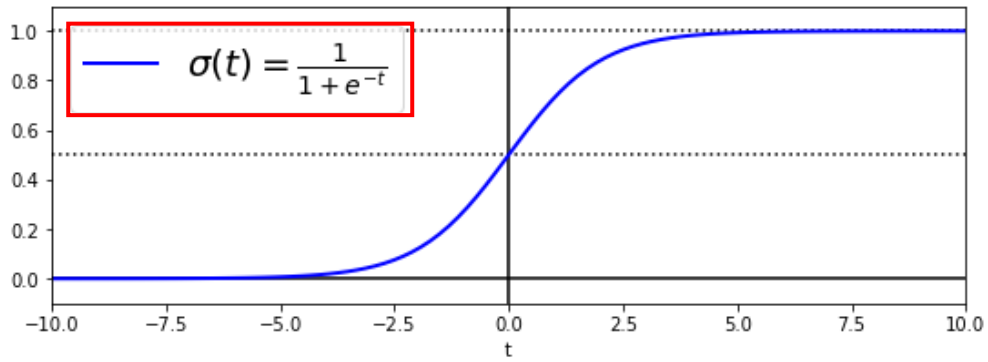
로지스틱 회귀: Logistic regression

식 4-15 로지스틱 회귀 모델 예측

$$\hat{y} = \begin{cases} 0 & \hat{p} < 0.5\text{일 때} \\ 1 & \hat{p} \geq 0.5\text{일 때} \end{cases}$$

4.6 로지스틱 회귀

1장에서도 이야기한 것처럼, 어떤 회귀 알고리즘은 분류에서도 사용할 수 있습니다(또는 그 반대의 경우도 있습니다). **로지스틱 회귀**logistic regression(또는 **로짓 회귀**logit regression)는 샘플이 특정 클래스에 속할 확률을 추정하는 데 널리 사용됩니다(예를 들면 이 이메일이 스팸일 확률은 얼마인가?). 추정 확률이 50%가 넘으면 모델은 그 샘플이 해당 클래스에 속한다고 예측합니다(즉, 레이블이 '1'인 **양성 클래스**positive class). 아니면 클래스에 속하지 않는다고 예측합니다(즉, 레이블이 '0'인 **음성 클래스**negative class). 이를 이진 분류기라고 합니다.



로지스틱 회귀

위키백과, 우리 모두의 백과사전.

로지스틱 회귀(영어: logistic regression)는 영국의 통계학자인 D. R. Cox가 1958년[1]에 제안한 확률 모델로서 독립 변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측하는데 사용되는 통계 기법이다.

로지스틱 회귀의 목적은 일반적인 회귀 분석의 목표와 동일하게 종속 변수와 독립 변수간의 관계를 구체적인 함수로 나타내어 향후 예측 모델에 사용하는 것이다. 이는 독립 변수의 선형 결합으로 종속 변수를 설명한다는 관점에서는 선형 회귀 분석과 유사하다. 하지만 로지스틱 회귀는 선형 회귀 분석과는 다르게 종속 변수가 범주형 데이터를 대상으로 하며 입력 데이터가 주어졌을 때 해당 데이터의 결과가 특정 분류로 나뉘기 때문에 일종의 분류 (classification) 기법으로도 볼 수 있다.

```python
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-10, 10, 100)
sig = 1 / (1 + np.exp(-t))
plt.figure(figsize=(9, 3))
plt.plot([-10, 10], [0, 0], "k-")
plt.plot([-10, 10], [0.5, 0.5], "k:")
plt.plot([-10, 10], [1, 1], "k:")
plt.plot([0, 0], [-1.1, 1.1], "k-")
plt.plot(t, sig, "b-", linewidth=2, label=r"$\sigma(t) = \frac{1}{1 + e^{-t}}$")
plt.xlabel("t")
plt.legend(loc="upper left", fontsize=20)
plt.axis([-10, 10, -0.1, 1.1])
plt.show()
```

KRICT 한국화학연구원

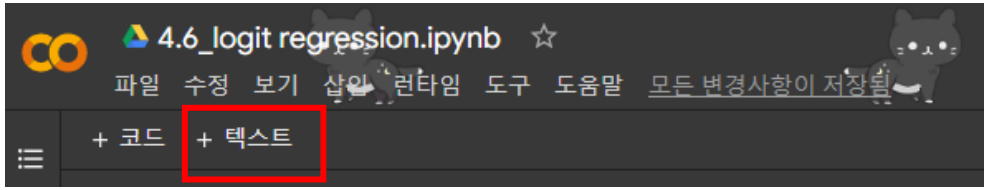
https://ko.wikipedia.org/wiki/%EB%A1%9C%EC%A7%80%EC%8A%A4%ED%8B%B1_%ED%9A%8C%EA%B7%80

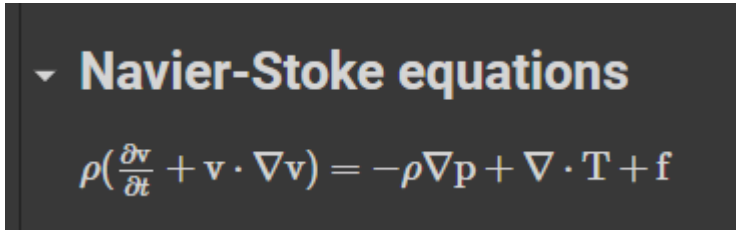
LaTeX

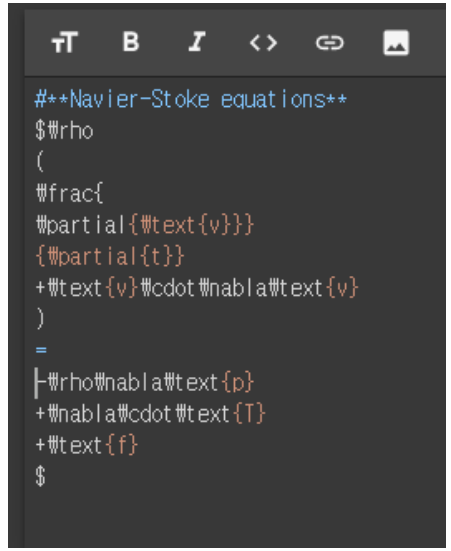
LaTeX – A document preparation system

LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents. LaTeX is available as free software.

- https://www.latex-project.org/
- https://colab.research.google.com/github/bebi103a/bebi103a.github.io/blob/master/lessons/00/intro_to_latex.ipynb
  > "intro_to_latex.ipynb"
- https://www.codecogs.com/latex/eqneditor.php

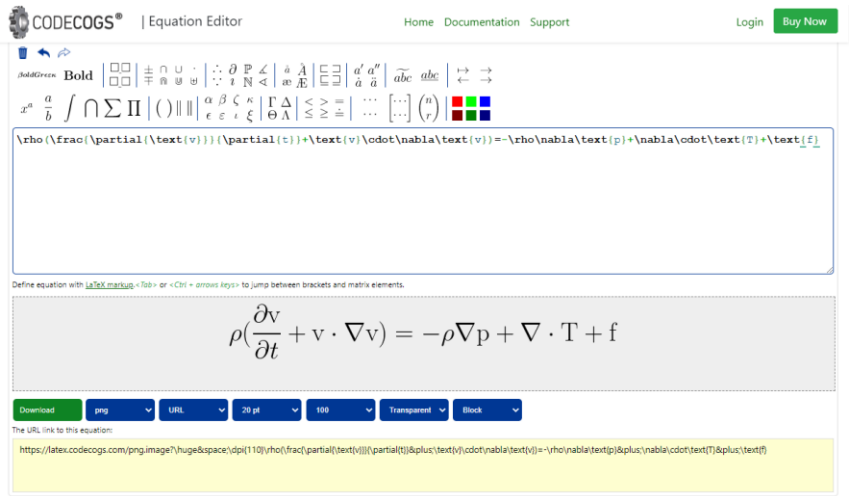






$$\rho(\frac{\partial \text{v}}{\partial t}+\text{v}\cdot\nabla\text{v})=-\rho\nabla\text{p}+\nabla\cdot\text{T}+\text{f}$$

*.png, larger pt size



KRICT 한국화학연구원

Iris: 붓꽃

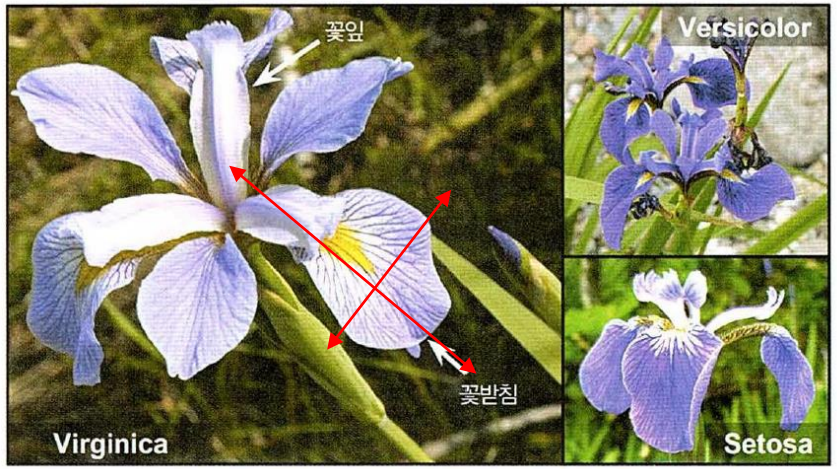
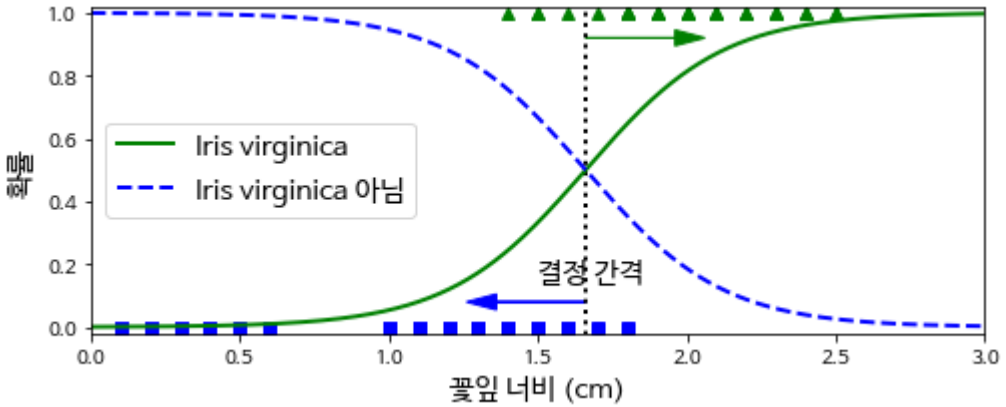


그림 4-22 세 종류의 붓꽃[40]




```
1 from sklearn import datasets
2 iris = datasets.load_iris()
3 list(iris.keys())
```

```
['data',
 'target',
 'frame',
 'target_names',
 'DESCR',
 'feature_names',
 'filename',
 'data_module']
```

```
1 X=iris["data"][:,3:]
2 y=(iris["target"]==2).astype(int)
```

```
1 print(iris.DESCR)
```

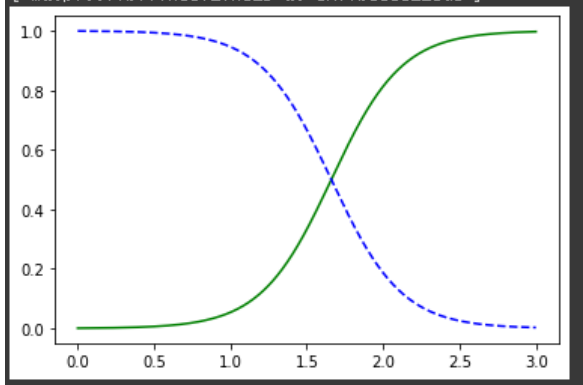
```
**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
```

```
1 from sklearn.linear_model import LogisticRegression
2 log_reg=LogisticRegression(solver="lbfgs",random_state=42)
3 log_reg.fit(X,y)
```

```
LogisticRegression(random_state=42)
```

```
1 X_new=np.linspace(0,3,1000).reshape(-1,1)
2 y_proba=log_reg.predict_proba(X_new)
3
4 plt.plot(X_new,y_proba[:,1],"g-",label="Iris virginica")
5 plt.plot(X_new,y_proba[:,0],"b--",label="Not Iris virginica")
```

```
[<matplotlib.lines.Line2D at 0x7fb9ee3223d0>]
```



```
[ ]  1 decision_boundary
```

```
array([1.66066066])
```

```
[ ]  1 log_reg.predict([[1.7],[1.5]])
```

```
array([1, 0])
```

```
1 plt.rc('font', family='NanumBarunGothic')
2
3 X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
4 y_proba = log_reg.predict_proba(X_new)
5 decision_boundary = X_new[y_proba[:, 1] >= 0.5][0]
6
7 plt.figure(figsize=(8, 3))
8 plt.plot(X[y==0], y[y==0], "bs")
9 plt.plot(X[y==1], y[y==1], "g^")
10 plt.plot([decision_boundary, decision_boundary], [-1, 2], "k:", linewidth=2)
11 plt.plot(X_new, y_proba[:, 1], "g-", linewidth=2, label="Iris virginica")
12 plt.plot(X_new, y_proba[:, 0], "b--", linewidth=2, label="Iris virginica 아님")
13 plt.text(decision_boundary+0.02, 0.15, "결정 간격", fontsize=14, color="k", ha="center")
14 plt.arrow(decision_boundary[0], 0.08, -0.3, 0, head_width=0.05, head_length=0.1, fc='b', ec='b')
15 plt.arrow(decision_boundary[0], 0.92, 0.3, 0, head_width=0.05, head_length=0.1, fc='g', ec='g')
16 plt.xlabel("꽃잎 너비 (cm)", fontsize=14)
17 plt.ylabel("확률", fontsize=14)
18 plt.legend(loc="center left", fontsize=14)
19 plt.axis([0, 3, -0.02, 1.02])
20 plt.show()
```

KRICT 한국화학연구원

한글 깨짐 문제 > 폰트 설치

STEP 1. 나눔 폰트 설치 (Nanum)

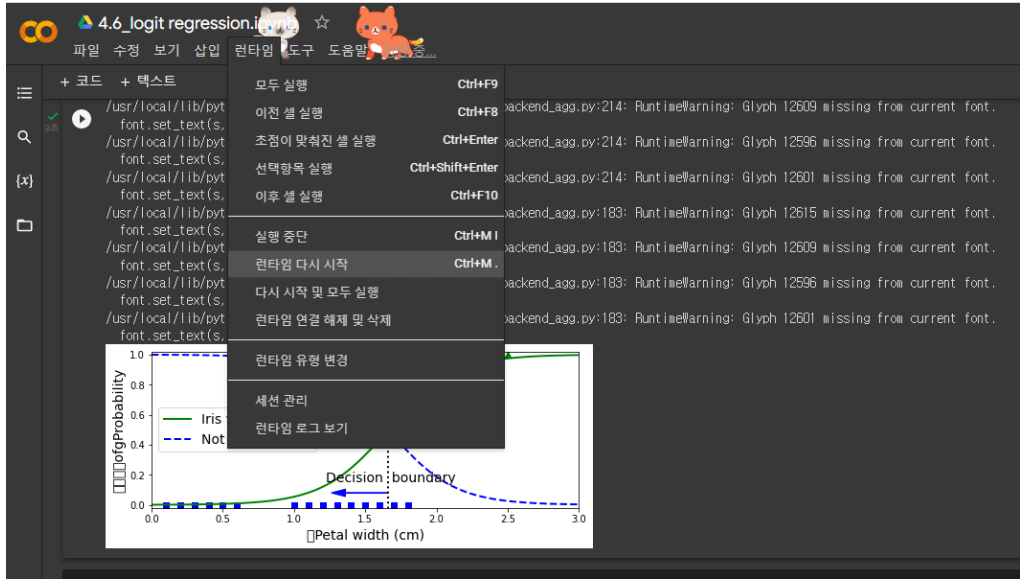
colab 파일을 연 뒤 첫번 째 cell에 아래 코드를 붙혀 넣고 실행합니다.

```
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

```
 1 !sudo apt-get install -y fonts-nanum
 2 !sudo fc-cache -fv
 3 !rm ~/.cache/matplotlib -rf

Reading package lists... Done
Building dependency tree
Reading state information... Done
fonts-nanum is already the newest version (20170925-1).
The following packages were automatically installed and are no longer required:
  libnvidia-common-460 nsight-compute-2020.2.0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 10 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
^C
```

STEP 2. 코랩(Colab)의 런타임을 재시작 합니다.

런타임 - 런타임 다시 시작 을 클릭하여 **런타임을 재시작**합니다.



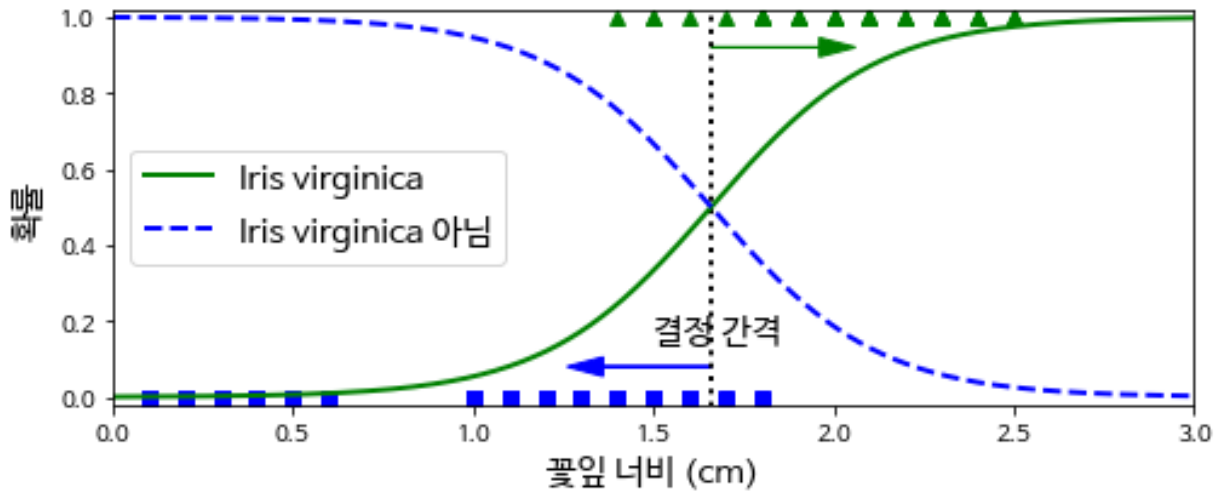
https://teddylee777.github.io/colab/colab-korean

KRICT 한국화학연구원

한글 깨짐 문제 > 폰트 설치

STEP 3. matplotlib의 폰트를 Nanum 폰트로 지정합니다.

```
import matplotlib.pyplot as plt

plt.rc('font', family='NanumBarunGothic')
```




```
1 plt.rc('font', family='NanumBarunGothic')
2
3 X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
4 y_proba = log_reg.predict_proba(X_new)
5 decision_boundary = X_new[y_proba[:, 1] >= 0.5][0]
6
7 plt.figure(figsize=(8, 3))
8 plt.plot(X[y==0], y[y==0], "bs")
9 plt.plot(X[y==1], y[y==1], "g^")
10 plt.plot([decision_boundary, decision_boundary], [-1, 2], "k:", linewidth=2)
11 plt.plot(X_new, y_proba[:, 1], "g-", linewidth=2, label="Iris virginica")
12 plt.plot(X_new, y_proba[:, 0], "b--", linewidth=2, label="Iris virginica 아님")
13 plt.text(decision_boundary+0.02, 0.15, "결정 간격", fontsize=14, color="k", ha="center")
14 plt.arrow(decision_boundary[0], 0.08, -0.3, 0, head_width=0.05, head_length=0.1, fc='b', ec='b')
15 plt.arrow(decision_boundary[0], 0.92, 0.3, 0, head_width=0.05, head_length=0.1, fc='g', ec='g')
16 plt.xlabel("꽃잎 너비 (cm)", fontsize=14)
17 plt.ylabel("확률", fontsize=14)
18 plt.legend(loc="center left", fontsize=14)
19 plt.axis([0, 3, -0.02, 1.02])
20 plt.show()
```

https://teddylee777.github.io/colab/colab-korean

KRICT 한국화학연구원

4.6.4 소프트맥스 회귀

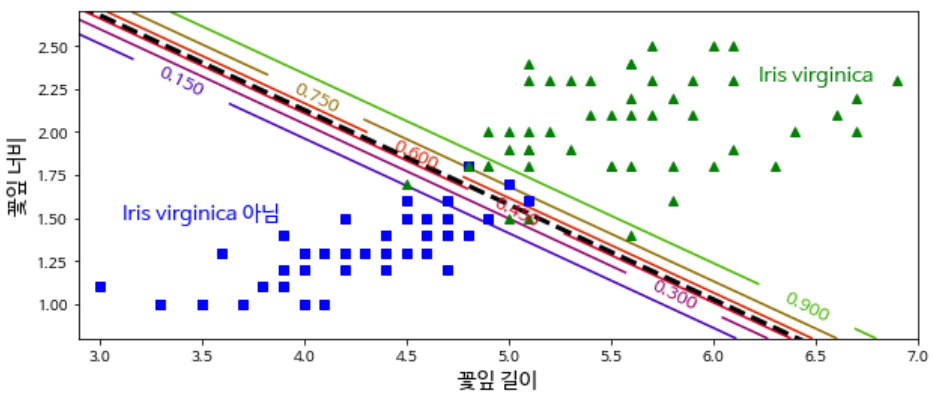
4.6.4 소프트맥스 회귀

로지스틱 회귀 모델은 (3장에서 언급한 것처럼) 여러 개의 이진 분류기를 훈련시켜 연결하지 않고 직접 다중 클래스를 지원하도록 일반화될 수 있습니다. 이를 **소프트맥스 회귀**softmax regression 또는 **다항 로지스틱 회귀**multinomial logistic regression라고 합니다.

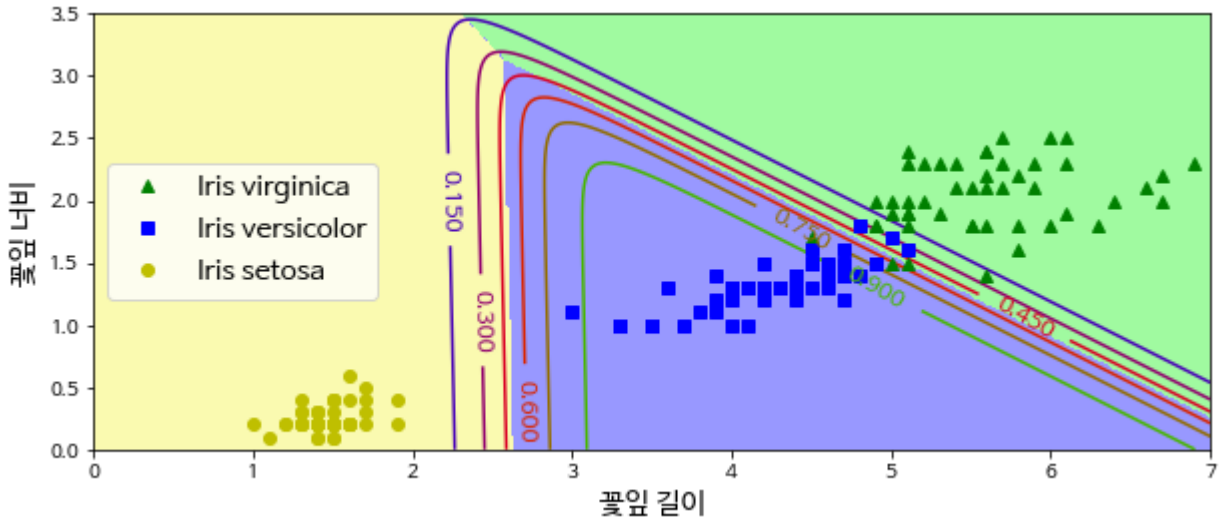
개념은 매우 간단합니다. 샘플 $\mathbf{x}$가 주어지면 먼저 소프트맥스 회귀 모델이 각 클래스 k에 대한 점수 $s_k(\mathbf{x})$를 계산하고, 그 점수에 **소프트맥스 함수**softmax function(또는 **정규화된 지수 함수**normalized exponential function라고 부릅니다)를 적용하여 각 클래스의 확률을 추정합니다. $s_k(\mathbf{x})$를 계산하는 식은 선형 회귀 예측을 위한 식과 매우 비슷해서 친숙할 것입니다(식 4-19).

TIP 소프트맥스 회귀 분류기는 한 번에 하나의 클래스만 예측합니다(즉, 다중 클래스multiclass지 다중 출력multioutput은 아닙니다). 그래서 종류가 다른 붓꽃 같이 상호 배타적인 클래스에서만 사용해야 합니다. 하나의 사진에서 여러 사람의 얼굴을 인식하는 데는 사용할 수 없습니다.

```python
from sklearn.linear_model import LogisticRegression

X = iris["data"][:, (2, 3)]
y = (iris["target"] == 2).astype(int)

log_reg = LogisticRegression(solver="lbfgs", C=10**10, random_state=42)
log_reg.fit(X, y)

x0, x1 = np.meshgrid(
        np.linspace(2.9, 7, 500).reshape(-1, 1),
        np.linspace(0.8, 2.7, 200).reshape(-1, 1),
    )
X_new = np.c_[x0.ravel(), x1.ravel()]

y_proba = log_reg.predict_proba(X_new)

plt.figure(figsize=(10, 4))
plt.plot(X[y==0, 0], X[y==0, 1], "bs")
plt.plot(X[y==1, 0], X[y==1, 1], "g^")

zz = y_proba[:, 1].reshape(x0.shape)
contour = plt.contour(x0, x1, zz, cmap=plt.cm.brg)


left_right = np.array([2.9, 7])
boundary = -(log_reg.coef_[0][0] * left_right + log_reg.intercept_[0]) / log_reg.coef_[0][1]

plt.clabel(contour, inline=1, fontsize=12)
plt.plot(left_right, boundary, "k--", linewidth=3)
plt.text(3.5, 1.5, "Iris virginica 아님", fontsize=14, color="b", ha="center")
plt.text(6.5, 2.3, "Iris virginica", fontsize=14, color="g", ha="center")
plt.xlabel("꽃잎 길이", fontsize=14)
plt.ylabel("꽃잎 너비", fontsize=14)
plt.axis([2.9, 7, 0.8, 2.7])
plt.show()
```



4.6.4 소프트맥스 회귀




```
[ ]  1 softmax_reg.predict([[5,2]])

     array([2])

     1 softmax_reg.predict_proba([[5,2]])

     array([[6.38014896e-07, 5.74929995e-02, 9.42506362e-01]])
```

```
1 x0, x1 = np.meshgrid(
2         np.linspace(0, 8, 500).reshape(-1, 1),
3         np.linspace(0, 3.5, 200).reshape(-1, 1),
4     )
5 X_new = np.c_[x0.ravel(), x1.ravel()]
6
7
8 y_proba = softmax_reg.predict_proba(X_new)
9 y_predict = softmax_reg.predict(X_new)
10
11 zz1 = y_proba[:, 1].reshape(x0.shape)
12 zz = y_predict.reshape(x0.shape)
13
14 plt.figure(figsize=(10, 4))
15 plt.plot(X[y==2, 0], X[y==2, 1], "g^", label="Iris virginica")
16 plt.plot(X[y==1, 0], X[y==1, 1], "bs", label="Iris versicolor")
17 plt.plot(X[y==0, 0], X[y==0, 1], "yo", label="Iris setosa")
18
19 from matplotlib.colors import ListedColormap
20 custom_cmap = ListedColormap(['#fafab0','#9898ff','#a0faa0'])
21
22 plt.contourf(x0, x1, zz, cmap=custom_cmap)
23 contour = plt.contour(x0, x1, zz1, cmap=plt.cm.brg)
24 plt.clabel(contour, inline=1, fontsize=12)
25 plt.xlabel("꽃잎 길이", fontsize=14)
26 plt.ylabel("꽃잎 너비", fontsize=14)
27 plt.legend(loc="center left", fontsize=14)
28 plt.axis([0, 7, 0, 3.5])
29 plt.show()
```

Class 0: $6.38*10^{-7}$ >> $6.38*10^{-5}$ % => **0%**
Class 1: $5.75*10^{-2}$ >> **5.75 %**
Class 2: $9.425*10^{-1}$ >> **94.25 %**

KRICT 한국화학연구원