Machine Learning

Logistic Regression

김선녕(ksycafe@gmail.com)

삶의 만족도 = $\theta_0 + \theta_1 \times 1인당GDP$

Table 1-1. Does money make people happier?

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

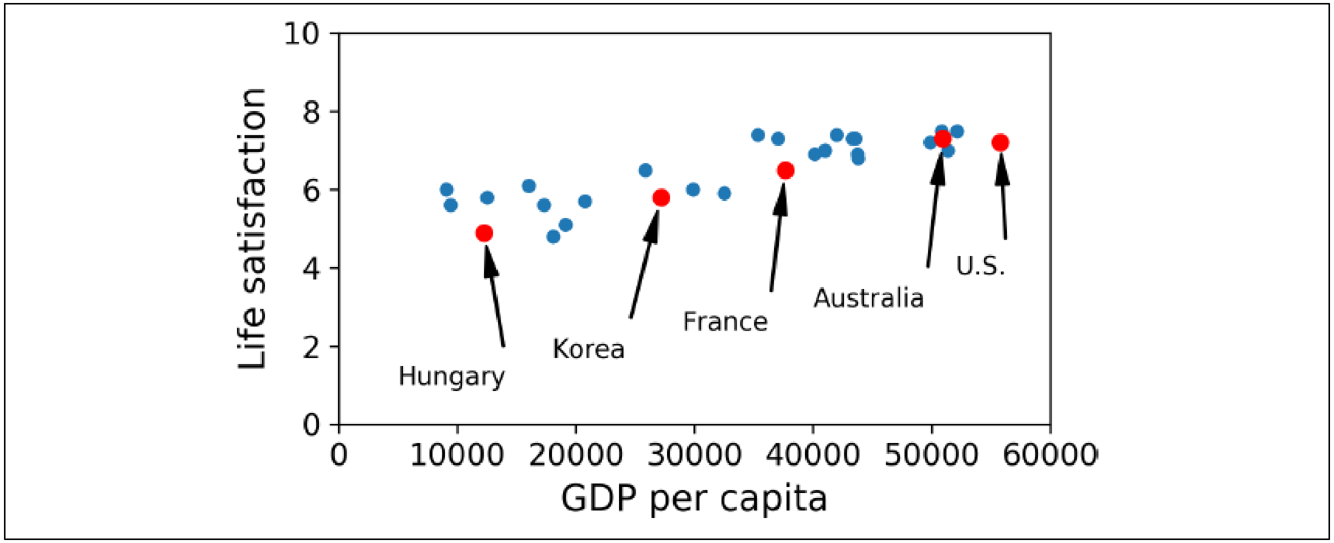


Figure 1-17. Do you see a trend here?

Ref : hands on machine learning 2nd - oreily

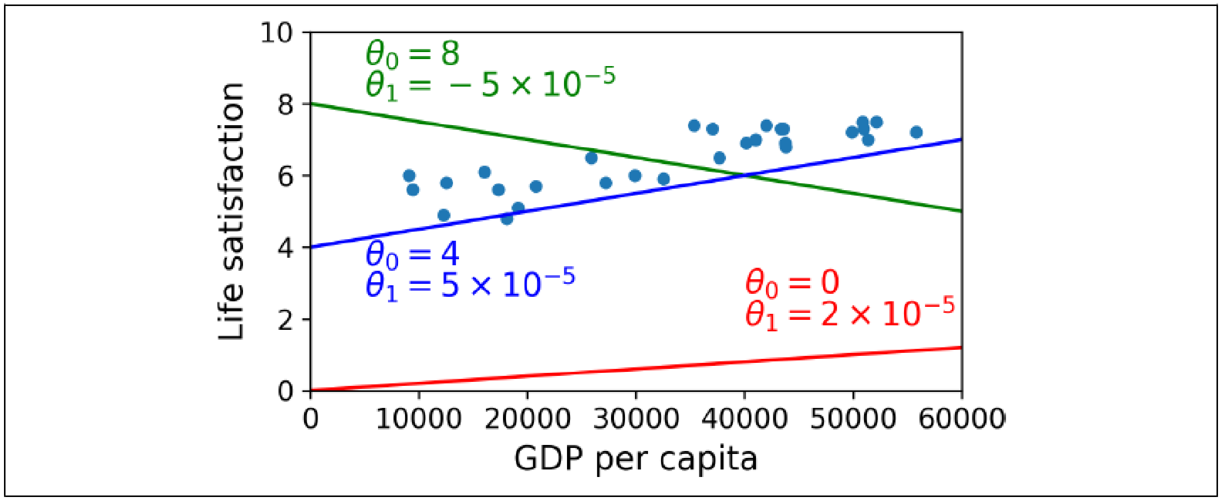


Figure 1-18. A few possible linear models

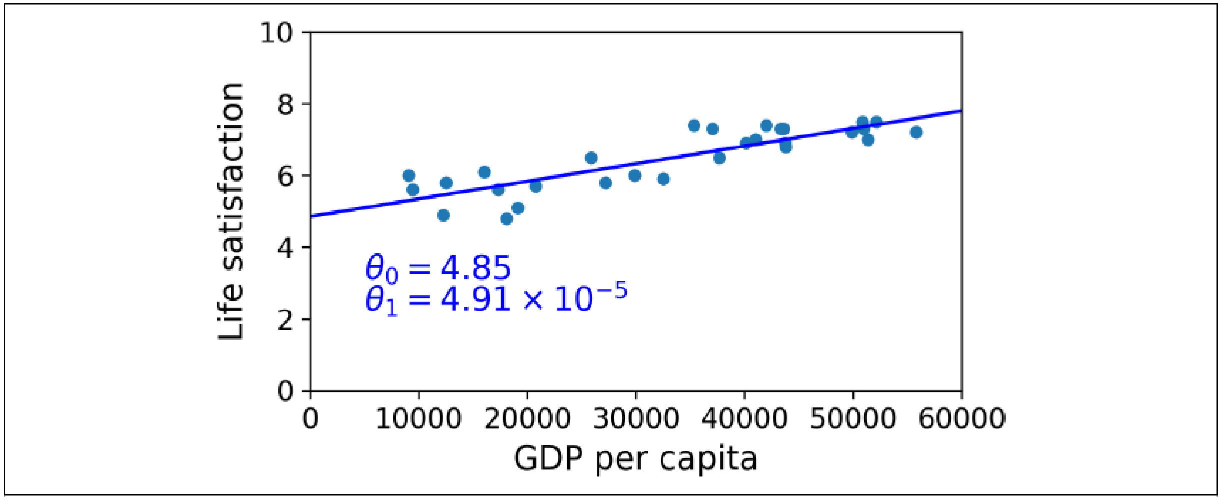


Figure 1-19. The linear model that fits the training data best

Ref : hands on machine learning 2nd - oreily

사각형 데이터 추가

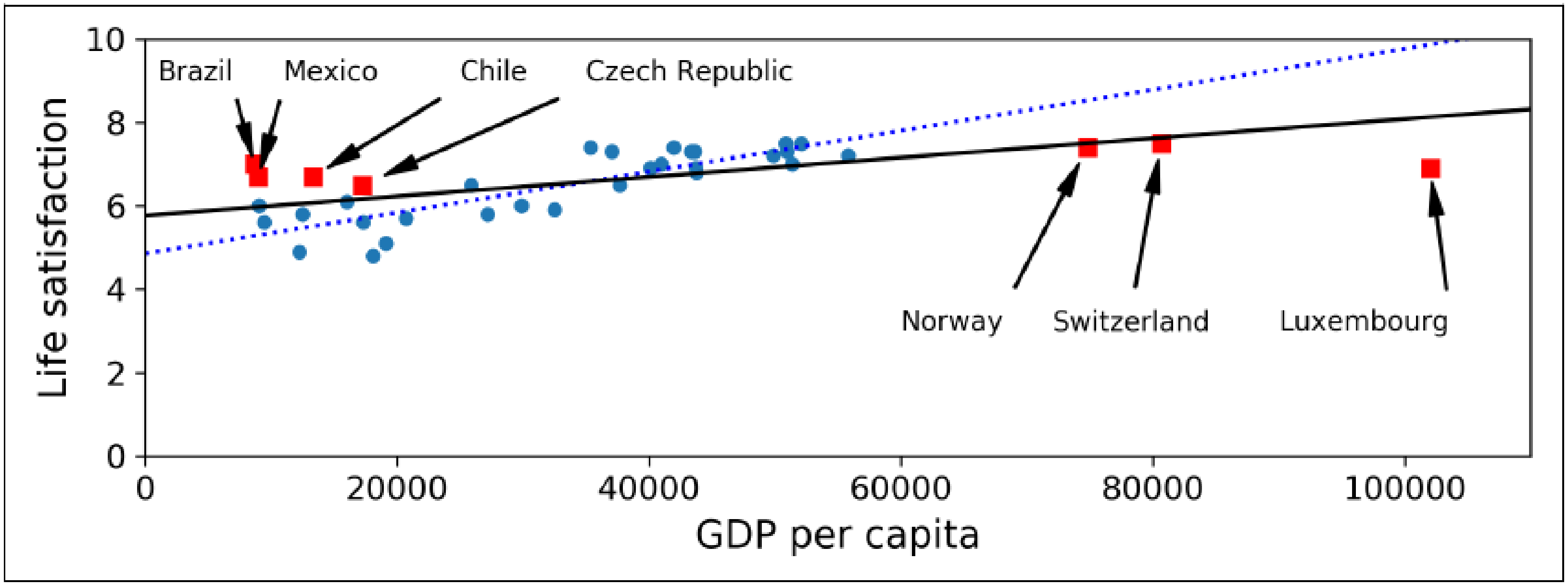
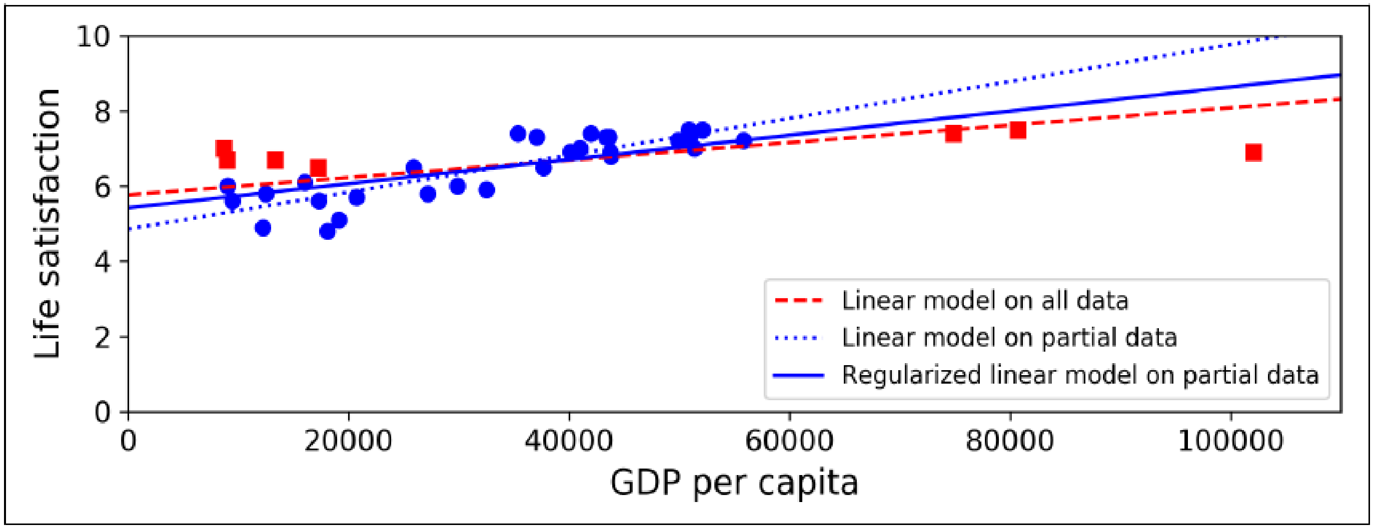


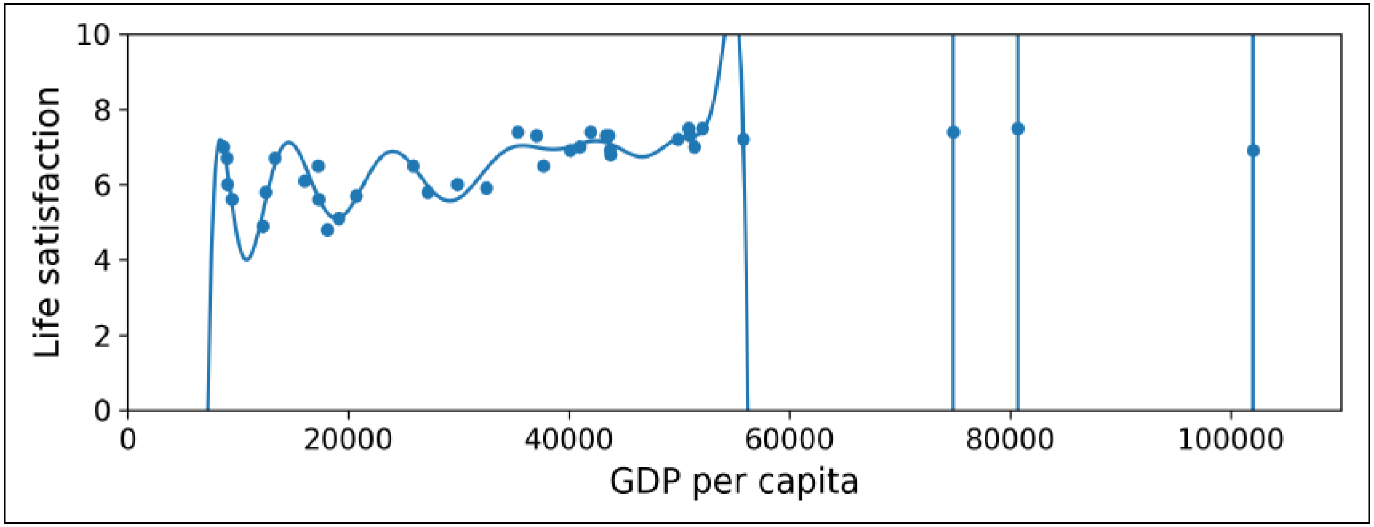
Figure 1-21. A more representative training sample

규제를 적용해 만든 선형 모델

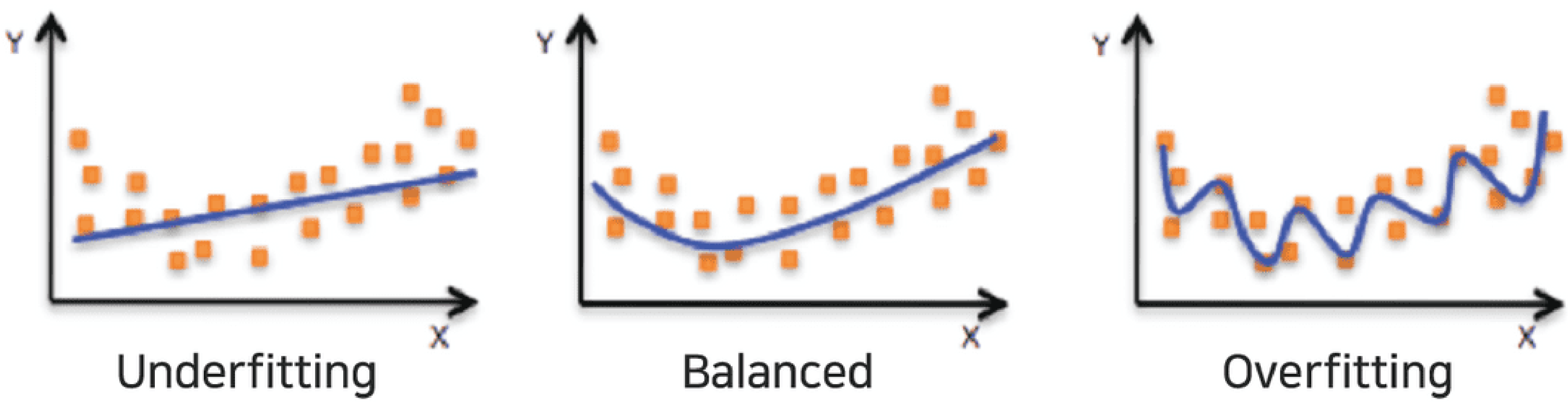


- 훈련데이터(동그라미)에는 덜 맞지만 새로운 샘플(사각형)에는 더 일반화
- 하이퍼파라미터(hyperparameter)
  - (모델이 아니라) 학습알고리즘의 파라미터
  - 학습 알고리즘의 영향을 받지 않으며, 훈련전에 미리 지정되고, 훈련하는 동안 상수로 남아있는다.
- 규제의 양은 하이퍼파라미터가 결정
  - 가중치 제한
  - 릿지(ridge), 라쏘(lasso)

과대적합/과소적합

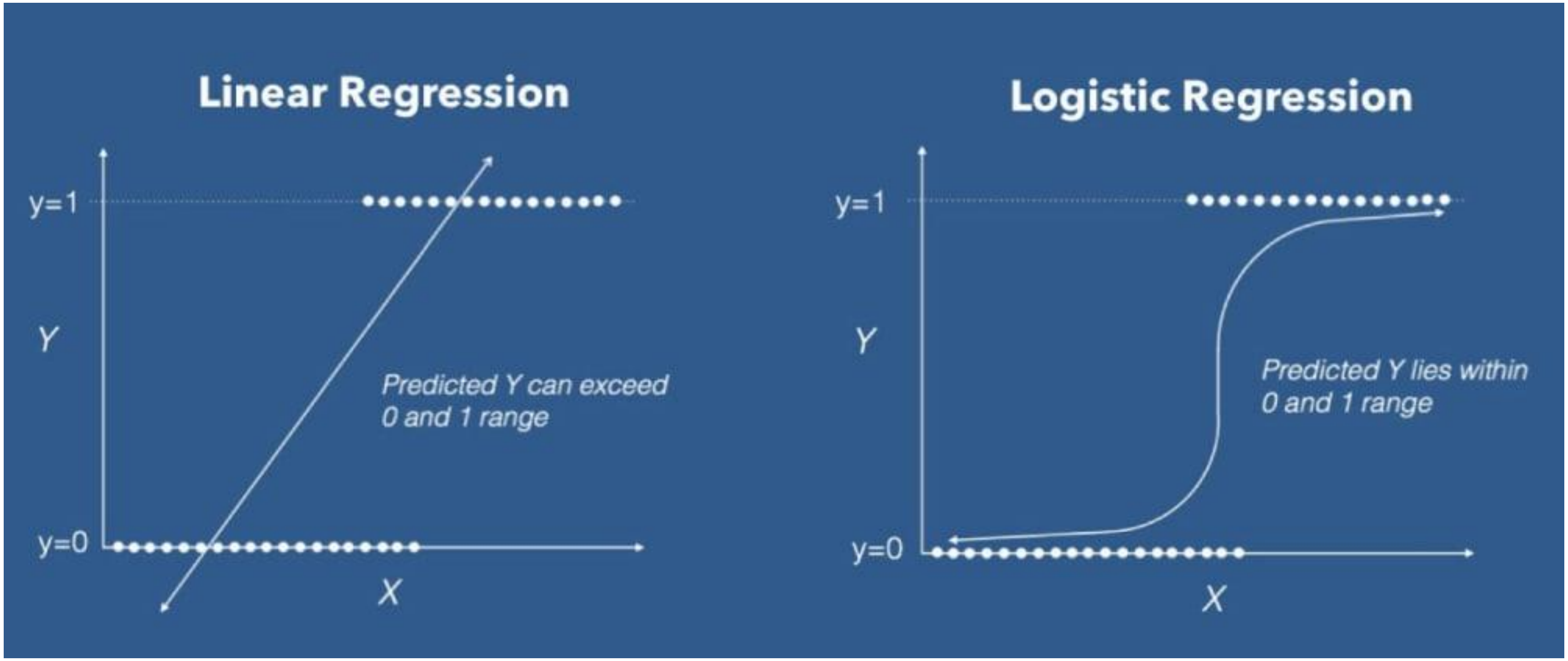


10
8
6
4
2
0
Life satisfaction
0
20000
40000
60000
80000
100000
GDP per capita



Y
X
Underfitting
Y
X
Balanced
Y
X
Overfitting

- 회귀를 사용하여 데이터가 어떤 **범주**에 속할 확률을 0에서 1사이의 값으로 예측
- 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류해주는 알고리즘
- 로지스틱 회귀 분석은 **이진 분류**를 수행하는 데 사용.
  - 데이터 샘플을 양성(1) 또는 음성(0) 클래스 둘 중 어디에 속하는지 예측한다.
- 로지스틱 회귀 모델의 확률추정
  - $\hat{y} = h_w(x) = \sigma(w^T x)$



Odds

- Odds
  - 임의의 사건 A가 발생하지 않을 확률 대비 일어날 확률의 비율을 뜻하는 개념
  - 실패에 비해 성공할 확률의 비.

$$odds = \frac{P(A)}{1 - P(A)}$$

- (예) 게임에서 이길 확률 1/5

$$odds = \frac{1/5}{1 - 1/5} = \frac{1}{4}$$

(5번 게임에서 4번 질 동안 1번 이긴다)

example

	의약품 A	의약품 B	합계
생존율(0)	32	24	56
생존율(1)	20	42	62
	52	66	118

- $Odds(A)$
  - $P(A)$ = 20/52 = 0.38
  - $Odds(A) = 0.38/1 - 0.38 = 0.61$
  - A를 복용하면 100명 사망할 동안 61명 생존
- $Odds(B)$
  - $P(B) = 42/66 = 0.63$
  - $Odds(B) = 0.63/1 - 0.63 = 1.7$
  - B를 복용하면 100명 사망할 동안 170명 생존
- B에 대한 A의 $Odds\ ratio\ =\ 0.61/1.7\ =\ 0.36$
  - B에 비해 A일 때 생존(성공)이 0.36배 = 64%가 생존율(성공율)이 떨어진다

Logistic Function

$$odds = \frac{p}{1-p}$$

- $0 < p < 1, \qquad 0 < 1 - p < 1$
- p가 0에 가까우면 $\dfrac{0}{1-0} = 0$, p가 1에 가까우면 $\dfrac{1}{1-1} = \infty$(무한대)
- 음의 무한대를 포함시키기 위하여 log를 취한다

  입력 값의 범위가 [0,1]일 때 출력 값의 범위를 $[-\infty, \infty]$로 조정

$$-\infty < \log\left(\frac{p}{1-p}\right) < \infty$$

[참고]

- $\log_e 0 = ?$: 정의되지 않는다. $e^x = 0$를 만족시키는 x는 없다
- x가 양의 변 (0+)에서 0에 가까워 질 때 x의 자연 로그 한계는 마이너스 무한대

  $\lim_{x \to 0} \log_e x = -\infty$

Log-Odds

$linear\ regression:\ y_i = \boldsymbol{\beta_0 + \beta_1 x_i + \epsilon_i}$

$logistic\ regression:\ \log\left(\frac{p}{1-p}\right) = \boldsymbol{\beta_0 + \beta_1 x_i + \epsilon_i}$

$$y = \beta_0 + \beta_1 x = W \cdot x$$

$$log\frac{p}{1-p} = W \cdot x$$

$$\frac{p}{1-p} = e^{Wx}$$

$$p = e^{Wx}(1-p) = e^{Wx} - e^{Wx}p$$

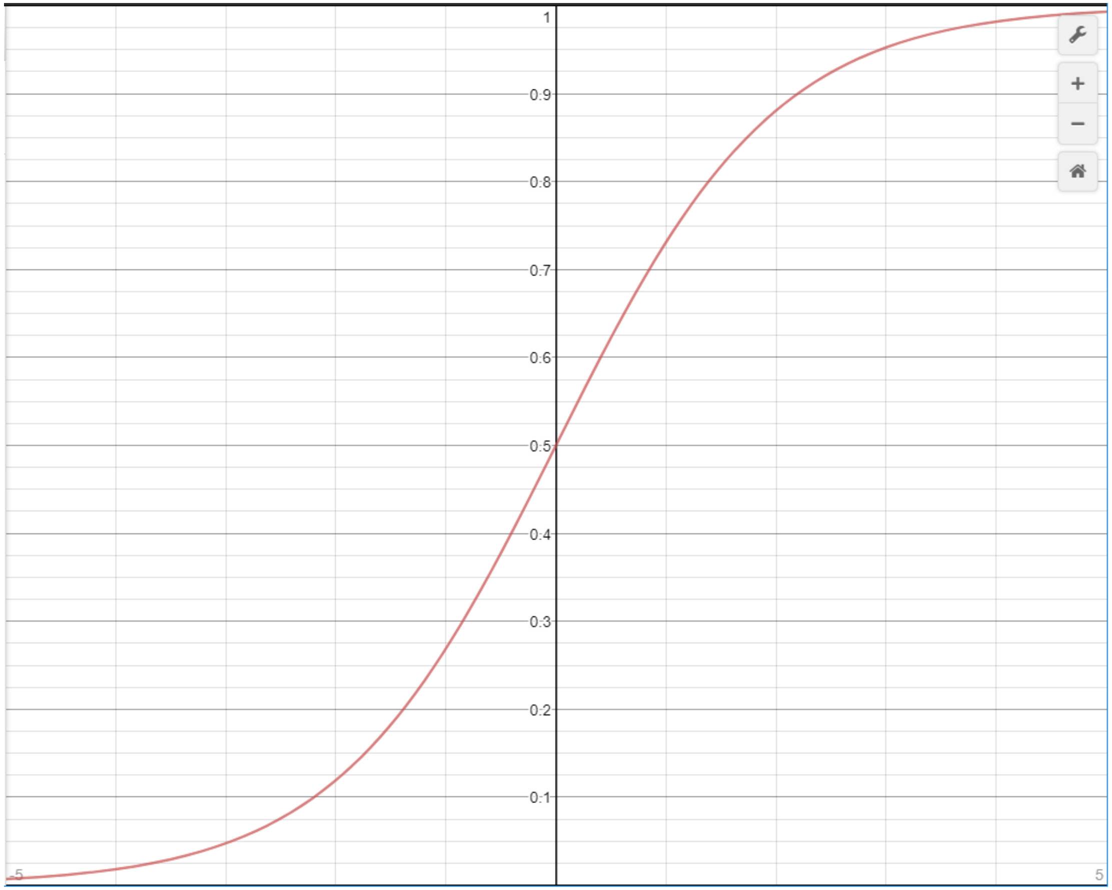
$$p + e^{Wx}p = e^{Wx}, \quad p(1+e^{Wx}) = e^{Wx}$$

$$\boldsymbol{p = \frac{e^{Wx}}{1+e^{Wx}} = \frac{1}{1+e^{-Wx}}}$$

$cf:\ y = a^x \Rightarrow x = log_a y,\quad y = e^x \Rightarrow x = log_e y$

Sigmoid Function/Logistic Function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



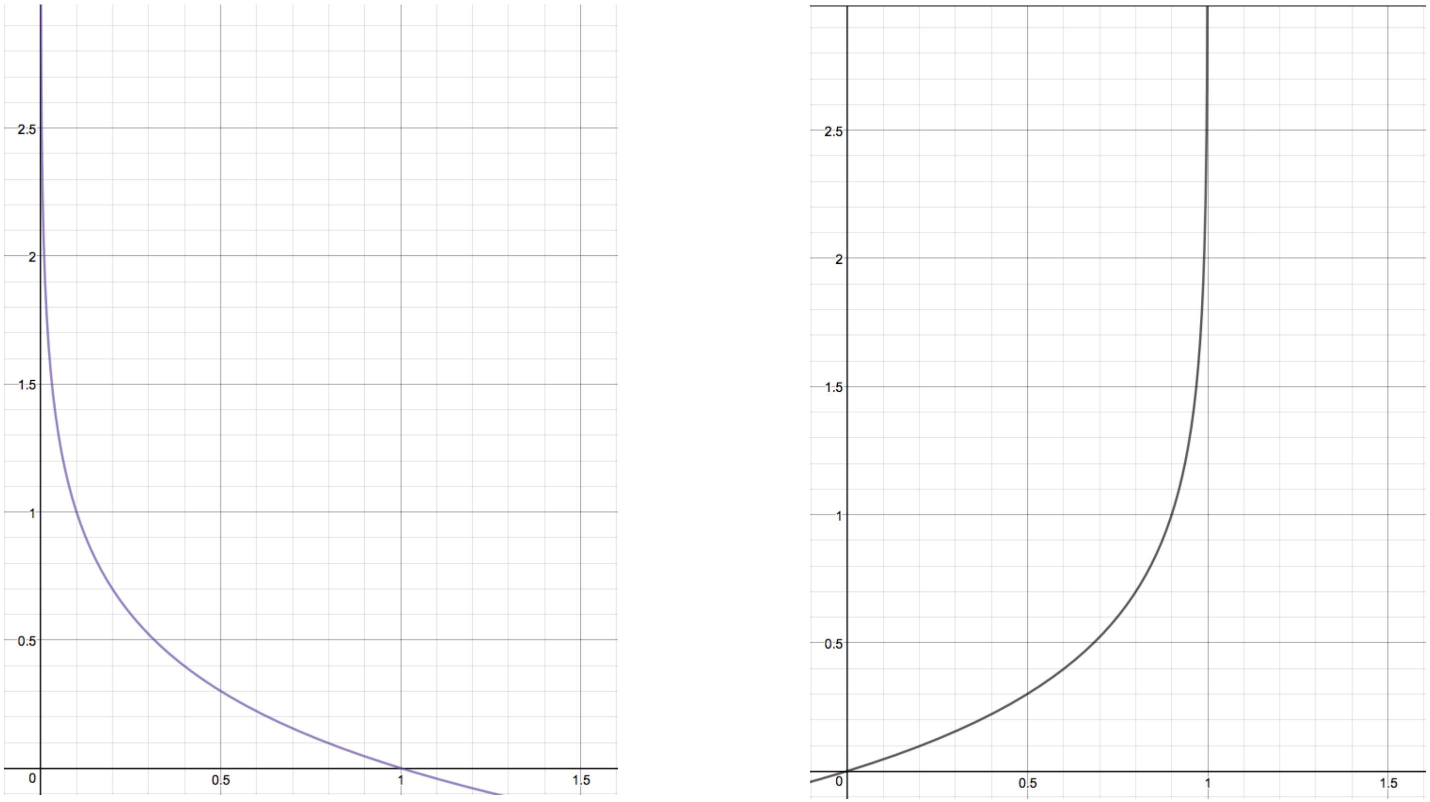
- $0 < \sigma(x) < 1$
- e : $2.7182\cdots$ (자연상수)
- $\sigma(1) = 0.731 \ldots$
- $\sigma(2) = 0.880 \ldots$
- **회귀모델예측**

$$\hat{y} = \begin{cases} \mathbf{0} & \boldsymbol{if} \quad \boldsymbol{\sigma(x) < 0.5} \\ \mathbf{1} & \boldsymbol{if} \quad \boldsymbol{\sigma(x) \geq 0.5} \end{cases}$$

Cost Function

양성샘플($y = 1$)에 대해서는 높은 확률을, 음성샘플($y = 0$)에 대해서는 낮은 확률을 추정하는 모델의 파라미터 θ를 찾는 것

$$cost(\theta) = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1 - \hat{y}) & if\ \ y = 0 \end{cases}$$



$y = 1 \quad \hat{y} = 1, \quad cost(\theta) = 0$

$\qquad\ \ \hat{y} = 0, \quad cost(\theta) = \infty$

$y = 0 \quad \hat{y} = 0, \quad cost(\theta) = 0$

$\qquad\ \ \hat{y} = 1, \quad cost(\theta) = \infty$

Cost function of Logistic Regression

- 하나의 훈련 샘플에 대한 비용함수

$$cost(\theta) = \begin{cases} -log(\hat{y}) & if\ y = 1 \\ -log(1-\hat{y}) & if\ y = 0 \end{cases}$$

- 전체 훈련세트에 대한 비용함수(log loss, cross entropy)

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1-y_i)\,log(1-\hat{y}_i)]$$

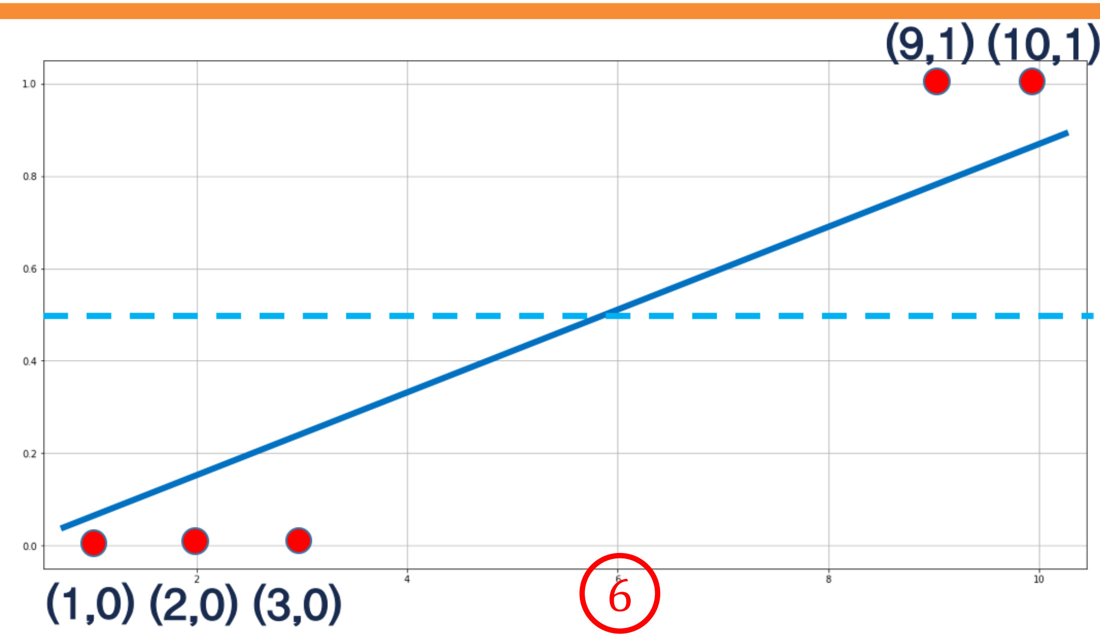
- 로지스틱 회귀 모델 훈련
  - 최솟값을 계산하는 알려진 해가 없다.
  - 하지만 위 비용함수는 블록 함수이므로 경사 하강법이 전역 최솟값을 찾는 것을 보장한다

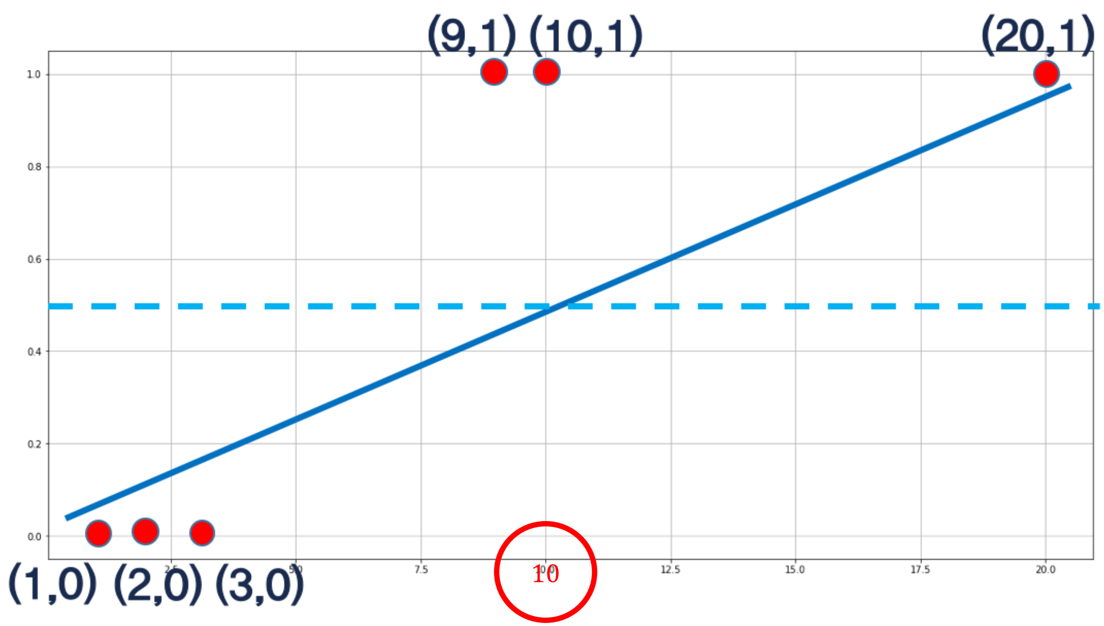
$$\frac{\delta}{\delta\theta}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\sigma(\theta^T x_i) - y_i)\,x_i$$

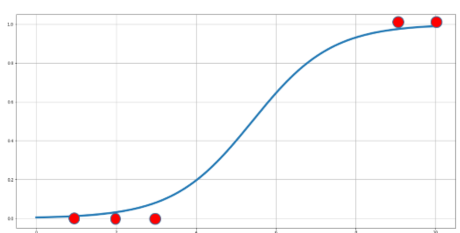
$\therefore\ y = 1,\quad f = -log(\hat{y})\qquad$ X

$\therefore\ y = 0,\quad f = \qquad$ X $\qquad -1 * log(1-\hat{y})$

이진 분류(Binary Classification)

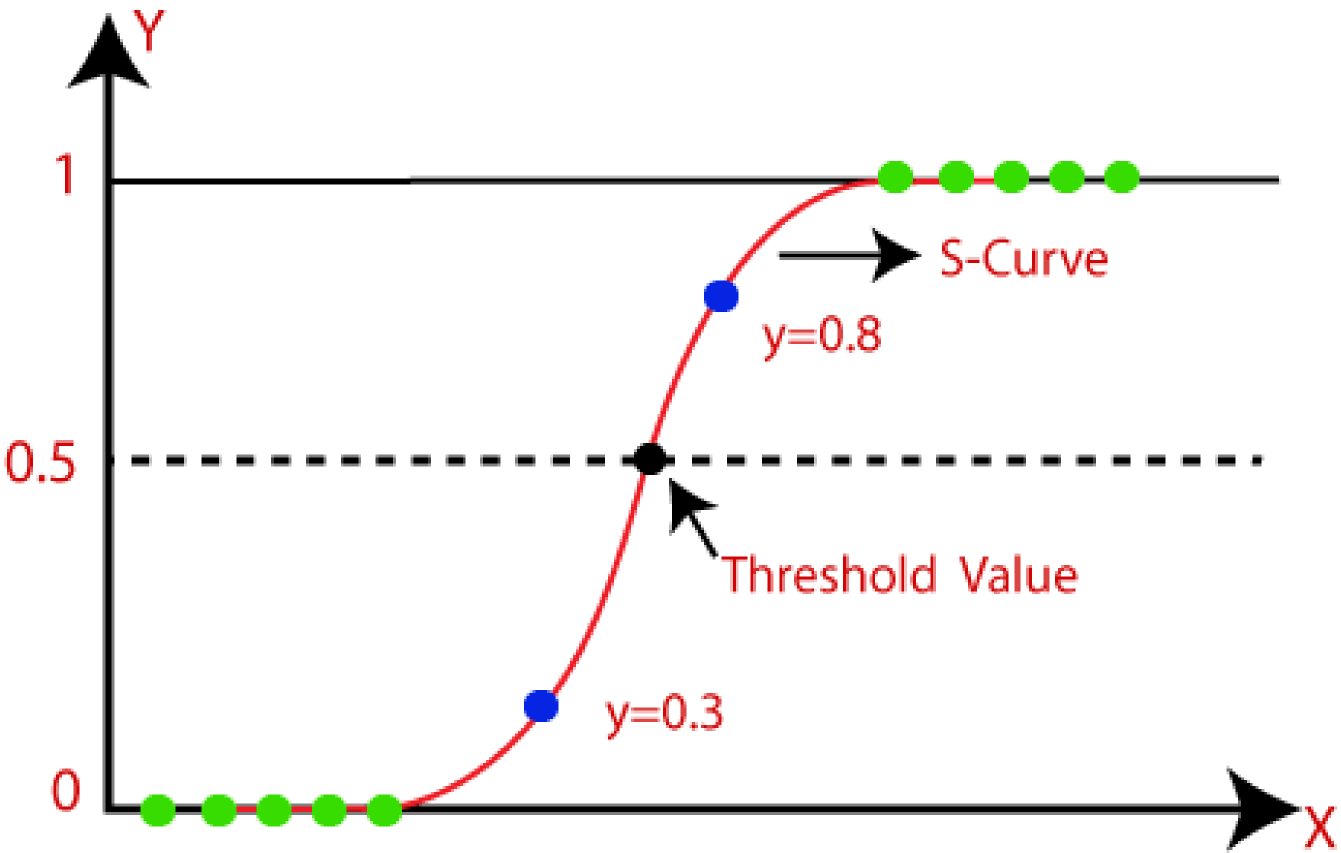






https://icim.nims.re.kr/post/easyMath/64

Threshold



https://www.javatpoint.com/logistic-regression-in-machine-learning

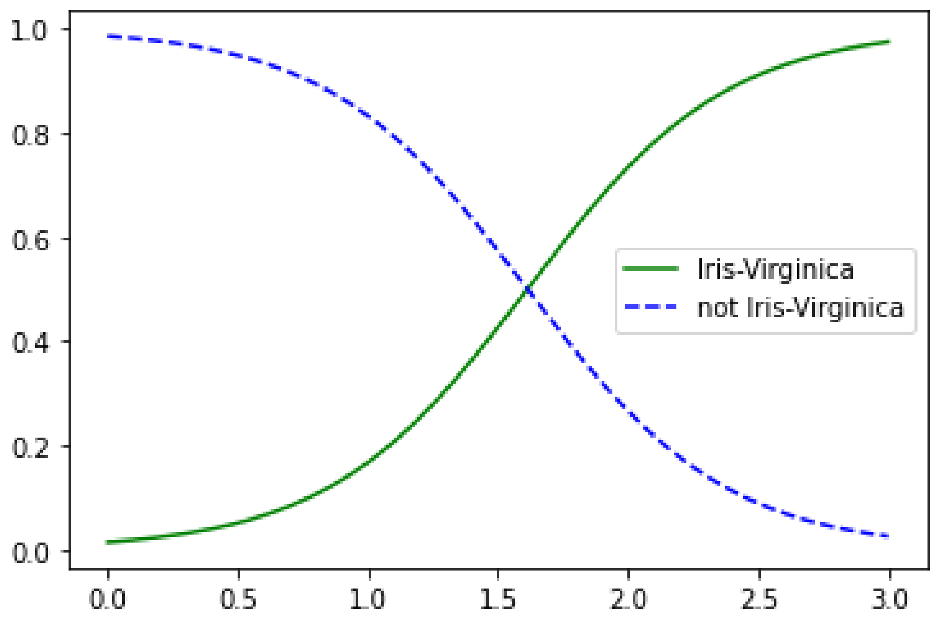
결정경계 - Iris Versinica 분류


```
# 붓꽃 데이터 로드
from sklearn import datasets

iris = datasets.load_iris()
list(iris.keys())
```

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

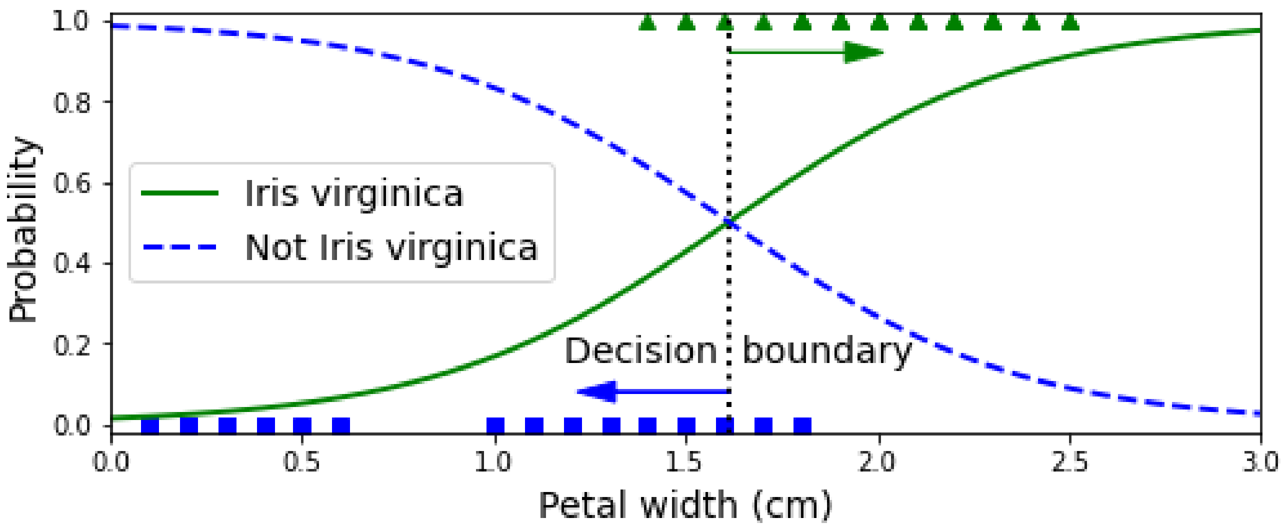
```
X = iris["data"][:,3:]     # 꽃잎의 너비만 사용
y = (iris["target"]==2).astype("int") #iris-Versinica이면 1,아니면 0

# 로지스틱 회귀모델 훈련
from sklearn.linear_model import LogisticRegression

# 향후 버전이 바뀌더라도 동일한 결과를 만들기 위해
# 사이킷런 0.22 버전의 기본값인 solver="lbfgs"로 지정
log_reg = LogisticRegression(solver="lbfgs", random_state=42)
log_reg.fit(X,y)
```

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

# 꽃잎의 너비가 0~3cm인 꽃에 대해 모델의 추정확률을 계산
X_new = np.linspace(0,3,1000).reshape(-1,1)
y_proba = log_reg.predict_proba(X_new)
plt.plot(X_new, y_proba[:,1], "g-", label = "Iris-Virginica")
plt.plot(X_new, y_proba[:,0], "b--", label = "not Iris-Virginica")
plt.legend()
plt.show()
```



```
X_new = np.linspace(0, 3, 1000).reshape(-1, 1)
y_proba = log_reg.predict_proba(X_new)
decision_boundary = X_new[y_proba[:, 1] >= 0.5][0]

plt.figure(figsize=(8, 3))
plt.plot(X[y==0], y[y==0], "bs")  # 음성범주 pointing
plt.plot(X[y==1], y[y==1], "g^")  # 양성범주 pointing

# 결정경계 표시
plt.plot([decision_boundary, decision_boundary], [-1, 2], "k:", linewidth=2)

# 추정확률 plotting
plt.plot(X_new, y_proba[:, 1], "g-", linewidth=2, label="Iris virginica")
plt.plot(X_new, y_proba[:, 0], "b--",linewidth=2, label="Not Iris virginica")

plt.text(decision_boundary+0.02, 0.15, "Decision  boundary", fontsize=14, co
lor="k", ha="center")
plt.arrow(decision_boundary, 0.08, -
0.3, 0, head_width=0.05, head_length=0.1, fc='b', ec='b')
plt.arrow(decision_boundary, 0.92, 0.3, 0, head_width=0.05, head_length=0.1,
 fc='g', ec='g')
plt.xlabel("Petal width (cm)", fontsize=14)
plt.ylabel("Probability", fontsize=14)
plt.legend(loc="center left", fontsize=14)
plt.axis([0, 3, -0.02, 1.02])

plt.show()
```

- Iris-Verginica(y=1)의 꽃잎 너비 : 1.4~2.5cm 사이에 분포(초록 삼각형)
- Iris-Verginica가 아닌 붓꽃의 꽃잎 너비 : 0.1~1.8cm에 분포(파란 사각형)
- 중첩되는 구간이 존재




```
decision_boundary
```

array([1.61561562])

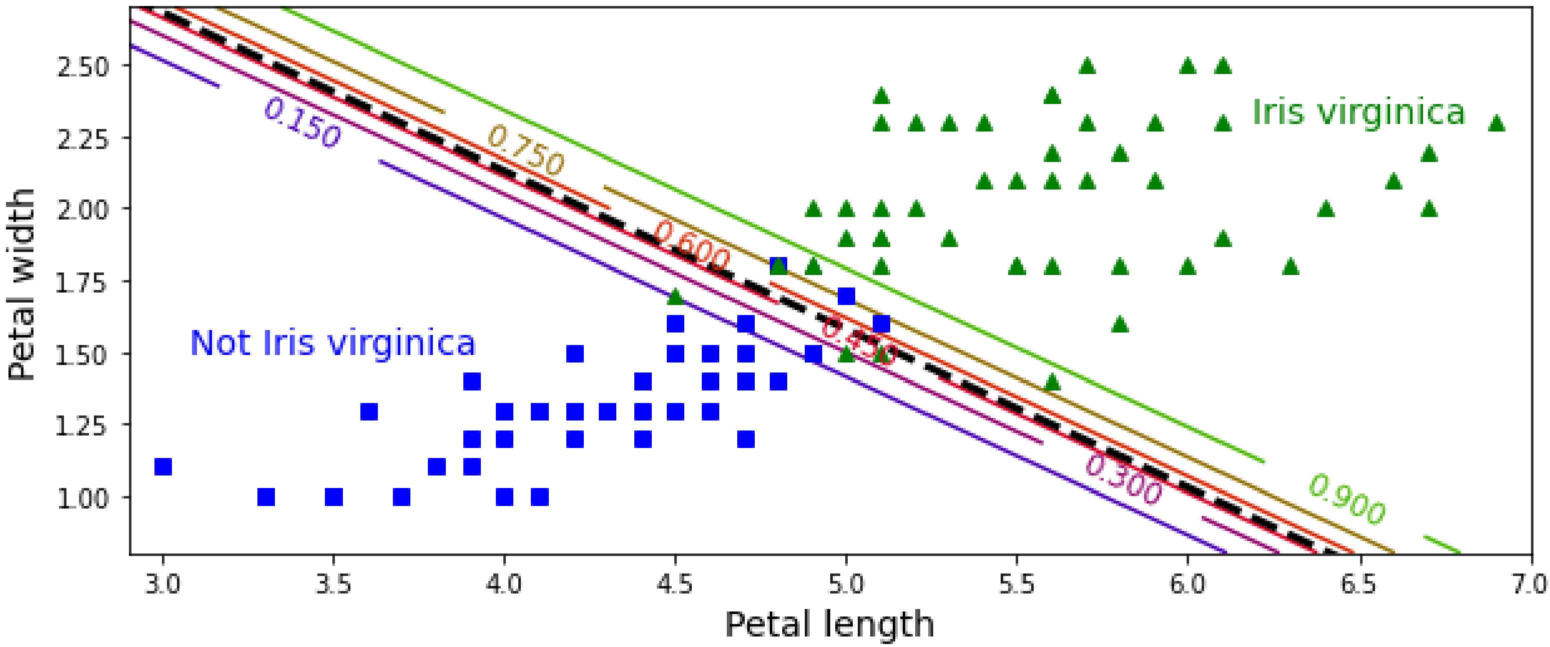
```
# 양쪽의 확률이 50%가 되는 1.6cm 근방에서 결정경계가 만들어지고,
# 분류기는 1.6cm보다 크면 Iris-Verginica로 분류하고 작으면 아니라고 예측한다.
log_reg.predict([[1.7], [1.5]])
```

array([1, 0])

꽃잎 너비와 꽃잎 길이 훈련

```
# 두번째 꽃잎 너비와 꽃잎 길이 2개의 특성을 이용해 훈련
from sklearn.linear_model import LogisticRegression

X = iris["data"][:, (2, 3)]  # petal length, petal width
y = (iris["target"] == 2).astype(np.int)

# LogisticRegression모델의 규제강도를 조절하는 하이퍼파라미터는 alpha가 아니라 그 역수에 해당하는 C이다. C가 높을수록 모델의 규제가 줄어든다.
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, "Not Iris virginica", fontsize=14, color="b", ha="center")
plt.text(6.5, 2.3, "Iris virginica", fontsize=14, color="g", ha="center")
plt.xlabel("Petal length", fontsize=14)
plt.ylabel("Petal width", fontsize=14)
plt.axis([2.9, 7, 0.8, 2.7])

plt.show()
```



- 점선은 모델이 50% 확률을 추정하는 지점으로, 이 모델의 결정경계이다.
- 이 경계는 $\theta_0 + \theta_1 x_1 + \theta_1 x_2 = 0$을 만족하는 (x_1, x_2)의 집합인 선형경계이다
- Iris-Verginica에 속할 확률을 15%부터 90%까지 나타내고 있다.