# Classification

## Classification Problem

- Classification은 “Target variable이 discontinuous한 regression”이라고 생각하면 쉽다.
- **특히 binary classification**의 경우, 전체 target variable의 평균은 곧 $Y = 1$인 observation들의 비율이 되고, 이는 다시 말해 1개의 observation을 뽑았을 때 **해당 observation의 $Y = 1$일 확률**을 의미한다.
  - 따라서 $\mathbb{E}\left(Y = 1 \mid X = x\right) = \Pr\left(Y = 1\right)$이 성립한다.

## Logistic Regression

- 각각의 feature $X_1, X_2, \ldots, X_p$에 대해서 선형식을 최대한 따르되 추정되는 값이 0에서 1 사이를 항상 만족하도록 추정되는 회귀분석
  - 아래와 같이 Sigmoid function의 형태를 따른다.

$$\begin{aligned}\mathbb{E}\left(Y = 1 \mid X = x\right) &= \Pr\left(Y = 1\right) \\ &= \frac{\exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)}{1 + \exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)}\end{aligned}$$

  - 이를 다시 re-arrange하면 다음과 같이 정리된다.

$$\begin{aligned}1 - \Pr\left(Y = 1\right) &= \Pr\left(Y = 0\right) \\ &= \frac{1}{1 + \exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)}\end{aligned}$$

$$\frac{\Pr\left(Y = 1\right)}{1 - \Pr\left(Y = 1\right)} = \exp\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\right)$$

$$\log\left(\frac{\Pr\left(Y = 1\right)}{1 - \Pr\left(Y = 1\right)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  - $\frac{p}{1-p}$를 보통 odds ratio라고 부르며, 여기에 $\log$를 붙인 값을 log odds 또는 logit이라 부른다. 즉 **logistic regression의 계수는 “다른 모든 변수가 고정되어 있을 때 $X_j$의 1 unit 증가로 인한 평균적인 logit의 변화분"**을 의미한다.

### How to estimate the logistic regression

- Maximum Likelihood Estimation이라는 principle을 기준으로 추정하며, likelihood function이란 보통 PDF를 모든 $i.i.d$ sample에 대해서 모두 곱해준 값을 일컫는다.
- 그에 따라서 logistic regression의 likelihood function은 다음과 같이 나타내어진다.

$$\prod_{y_i=1} \frac{\exp\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right)}{1 + \exp\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right)} \cdot \prod_{y_i=0} \frac{1}{1 + \exp\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right)}$$

- 보통 이를 $\log$ 변환한 log-likelihood를 maximize하는 방향으로 추정되며, $\log$가 증가함수이므로 $\arg\max$ 값은 변화하지 않는다는 점을 이용한다.

$$\begin{aligned}
&\sum_{y_i=1} \log p\left(x_i, \beta\right) + \sum_{y_i=0} \log\left(1 - p\left(x_i, \beta\right)\right) \\
&= \sum_{i=1}^{N} \left[y_i \log p + \left(1 - y_i\right) \log\left(1 - p\right)\right] \\
&= \sum_{i=1}^{N} \left[y_i \log \frac{p}{1-p} + \log\left(1 - p\right)\right] \\
&= \sum_{i=1}^{N} \left[y_i \left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right) - \log\left(1 + \exp\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right)\right)\right]
\end{aligned}$$

  - 이 식을 활용해 optimization problem을 풀어야 하므로, First Order Condition을 사용한다 (즉, 편미분해서 0이 되는 $\beta$값을 찾는다.)

- 이를 풀어내는 구체적인 알고리즘으로는 **`newton-cg`** 등 다양한 방법이 있으며, 공통점은 **모두 numerical apporach**라는 점이다.
  - 즉 어떤 구체적인 함수의 형태를 찾지 않고 값을 대입하면서 partial derivative 값이 더 이상 변화하지 않고 수렴하는 지점까지 반복하는 알고리즘
- 그리고 이를 통해 계수가 추정된 경우, 계수와 현재 주어진 feature들의 값을 위의 probability 식에 대입하여 해당 observation이 1로 분류될 확률을 구체적으로 계산할 수 있으며, 이렇게 계산된 분류 확률이 곧 classification에서의 predicted value가 된다.
- 이렇게 계산된 predicted probability에 대해서 threshold 값을 지정하게 되면, 해당 threshold보다 높은 predicted probability를 갖는 observation은 1로, 그렇지 않은 observation은 0으로 분류된다.

## Confounding factor의 제거

- Default 여부(0 or 1)를 target 변수로 두고, 이를 student(0 or 1)에 대해서 logistic regression을 수행하면 양의 계수가 나온다.
  - 하지만 balance와 함께 수행할 경우 student의 계수는 음수가 된다.
- 이는 왜냐하면 balance가 추가될 경우 student 변수의 계수의 의미는 “나머지 변수는 모두 고정된 상태에서 학생 여부가 default 여부에 미치는 영향"을 의미하게 되는데, balance가 같은 수준에서는 학생인 경우가 아닌 경우에 비해 default 상황이 될 확률이 낮아지기 때문이다.
  - 이 상황이 student 여부와 default 여부 사이의 true relationship이며,

- 이렇듯 balance라는 변수를 생략함으로써 predictor의 계수에 발생하는 bias를 Omitted Variable Bias라고 부르며, 생략된 변수(balance)를 confounding factor, 그리고 이를 model에 포함시키는 것을 confounding factor에 대한 통제라고 한다.

## Multiple Logistic Regression vs Multinomial Logistic Regression

- Multiple Logistic Regression: Binary target & **More than 1 predictors**
- Multinomial Logistic Regression: **More than two categories in the target**
  - 설명변수의 개수와 multinomial 여부는 관계가 없음
  - 한편 multinomial logistic regression에서 사용되는 parametric function의 형태는 sigmoid가 아닌 **softmax**라고 부른다.

# Discriminant Analysis

- Logistic regression이 확률 추정을 위해 sigmoid function의 argument에 선형식을 plug-in한다면, Discriminant Analysis는 Bayes theorem을 활용
- Bayes theorem이 $\Pr(Y = k \mid X = x)$를 추정하기 위해 $\Pr(Y = k)$와 $\Pr(X = x \mid Y = k)$를 사용하는 것을 활용
  - 여기서 $\Pr(X = x \mid Y = k)$를 conditional density, 즉 $x$가 $k$번째 category로 분류될 density, $\Pr(Y = k)$를 prior probability라고 한다.
- Discriminant Analysis의 구체적인 식은 다음과 같다.

$$\Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

  - 이때 prior probability $\pi_k$는 이미 데이터로부터 주어지는 정보이므로, 결국 우리는 $Y = k$라는 정보가 주어졌을 때 특정한 observation $x$가 해당 class에서 등장할 확률(or density)을 최대화하는 class로 해당 observation을 분류하고자 한다.
  - 한편 이때 conidtional density $f_k(x)$는 Gaussian density로 가정한다.
    - Predictor가 1개일 때의 Gaussian density

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right]$$

      - $x$는 개별 observation, $\sigma_k$는 $k$번째 class 내에 속한 모든 observation의 표준편차, $\mu_k$는 $k$번째 class 내에 속한 모든 observation의 mean
    - Predictor가 2개 이상일 때의 Gaussian density

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} \det(\Sigma_k)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\mu_k)^T \Sigma_k^{-1}(\boldsymbol{x}-\mu_k)\right]$$

    - 여기서 $\Sigma_k$는 $k$번째 class에 속한 모든 observation들 사이의 variance-covariance matrix, $\mu_k$는 observation들의 mean vector

- Advantages of DA over Logistic regression
    - Class들이 서로 겹치지 않고 잘 분류되어 있는 경우, 오히려 logistic regression은 그 결과가 unstable한 경우가 많다.
    - 전체 observation의 수가 적거나 각 feature들이 대체로 정규분포를 따르고 있는 경우에도 LDA가 보다 stable한 예측결과를 만들어낼 수 있다.
    - 또한 LDA란 결국 두 집단을 나누는 하나의 기준을 만드는 작업이므로 low-dimensional 시각화에도 좋은 성능을 보인다.

# Estimation of LDA

- 위에서 정의한 Gaussian density를 Discriminant Analysis 식에 대입해서 사용하며, 모든 parameter들은 given dataset으로부터 계산한 추정값을 사용한다.
    - **즉 $\pi_k$는 주어진 데이터에서 전체 $Y$ 가운데 $Y=k$인 observation들의 비율인 $\hat{\pi}_k$를 대입**
    - **$\mu_k$는 $Y=k$인 모든 observation에 대한 feature들의 sample mean**
    - **$\sigma_k$ 역시 마찬가지로 $Y=k$인 모든 observation에 대한 feature들의 sample standard deviation**
        - 다만 LDA의 경우 모든 class에 걸쳐 standard deviation이 같다고 가정하므로, 개별 class에 대한 sample standard deviation $\sigma_k$들의 가중평균으로 구한다.

$$\hat{\sigma}^2 = \sum_{k=1}^{K} \frac{n_k - 1}{n-K} \sigma_k^2 \;\Rightarrow\; \hat{\sigma} = \sqrt{\sum_{k=1}^{K} \frac{n_k - 1}{n-K} \sigma_k^2}$$

- 그에 따라서 final output $k^*$는 $\frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$를 최대화하는 $\arg\max$로 계산되며, likelihood function과 마찬가지로 $\log$ 변환한 뒤 $k$에 관련되지 않은 부분을 다시 정리하면 아래와 같은 식이 도출된다.

$$\delta_k(x) = x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \log(\pi_k)$$

    - 그리고 $\delta_k(x)$을 최대화하는 $k$를 찾아 해당 class로서 $x$를 분류한다.
    - 만일 전체 class가 2개이고 $\pi_1 = \pi_2 = 0.5$인 경우 decision boundary $x = \frac{\mu_1+\mu_2}{2}$이다. 단, 두 class에 대해 표준편차는 같다고 가정한다.
        - **decision boundary에서는 모든 $\delta_k(x)$가 서로 같아야 하기 때문이다.**

- 따라서

$$
\begin{aligned}
&\delta_1 = \delta_2 \\
&\Leftrightarrow x\frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + \log(0.5) = x\frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} + \log(0.5) \\
&\Leftrightarrow x(\mu_1 - \mu_2) = \frac{1}{2}\left(\mu_1^2 - \mu_2^2\right) \\
&\therefore x = \frac{\mu_1 + \mu_2}{2}
\end{aligned}
$$

- 나아가, 위에서 계산된 estimate $\hat{\mu}_k$와 $\hat{\sigma}_k$를 사용하면 $\hat{\delta}_k(x)$ 역시 추정할 수 있으며, 그에 따라서 개별 observation $x$가 특정 class $k$에 속할 확률 역시 추정할 수 있다.

$$
\hat{\Pr}(Y = k \mid X = x) = \frac{\exp\left[\hat{\delta}_k(x)\right]}{\sum_{l=1}^{K}\hat{\delta}_l(x)}
$$

- 이렇게 추정된 확률에 대해서
  - Binary classification의 경우 주로 $Y = 1$일 확률을 추정하고, pre-defined threshold와 비교하여 그 값보다 크면 1, 작으면 0으로 분류한다.
  - Multi-class classifcation의 경우 주로 전체 확률을 추정하고, 그 가운데 가장 높은 추정 확률을 보이는 class로 분류한다.

# Metrics for Classification

- For binary classfication

## Accuracy

- $y_i$를 True class, $\hat{y}_i$를 Predicted class라고 하면, “전체 $n$개 가운데 $y_i = \hat{y}_i$를 만족하는 observation의 비율"로써 측정
- (1 - accuracy) = misclasfication rate이 성립한다.
- But 개별 예측 결과 자체에는 신경쓰지 않고 오로지 서로 같은지 다른지만 측정하기 때문에, 특히 default 예시와 같이 “1”로 정확하게 분류하는 것이 중요한, 즉 분류 결과가 어떻게 나왔는지에 대해서도 관심을 가져야 하는 경우 Accuracy는 한계가 있다.

## Confusion Matrix

- 특히 Binary classifcation에 대해서 $2 \times 2$의 행렬로 정의되며, 각각의 row는 predicted class, 각각의 column은 true class를 assign하여 각각에 해당하는 observation의 수를 센다.

| | True No | True Yes |
|---|---|---|
| Predicted No | True Negative | False Negative |

| | True No | True Yes |
|---|---|---|
| Predicted Yes | False Positive | True Positive |

- “True / False”는 True class를 기준으로, “Positive / Negative”는 Predicted class를 기준으로 본다.
  - True class와 일치하면 True, True class와 일치하지 않으면 False
  - Predicted class가 Yes이면 Positive, Predicted class가 No이면 Negative
  - 즉 False positive란 “예측 결과 Yes이지만 True label과 일치하지 않음"이라는 뜻
- 그리고 confusion matrix로부터 계산되는 다양한 metric은 다음이 있다.
  - False positive rate = $\frac{FP}{FP+TN}$
    - 전체 No(0) 가운데 Yes(1)로 잘못 분류하는 비율
  - False negative rate = $\frac{FN}{FN+TP}$
    - 전체 Yes(1) 가운데 No(0)로 잘못 분류하는 비율
    - 특히 default case와 같이 1의 비중이 아주 작은 경우에는 False negative rate을 줄이는 것이 중요하다.

## ROC Curve

- False positive rate - True positive rate 사이의 관계를 나타내는 plot으로, curve 아래의 넓이(AUC: Area Under Curve)가 넓으면 넓을수록 1과 0 사이의 경계가 뚜렷해짐을 나타냄
- Higher AUC is good, but increasing training ROC AUC can lead to overfitting

## Other forms of Discriminant Analysis

- QDA: class별 variance(or variance-covariance matrix)가 서로 다르다는 것을 가정하는 경우
- Naive Bayes: density를 observation 전체 단위가 아닌 개별 predictor 단위의 곱으로 가정하는 경우; 즉 모든 predictor들의 independence를 가정하는 경우

## Logistic vs LDA

- Binary class의 경우, LDA에 $\log$를 취하면 Logistic regression과 동일한 form을 갖는다.
- 다만 Logistic은 conditional likelihood, LDA는 full joint likelihood에 기반을 두고 추정한다는 차이점이 있음
- 또한 Logistic에 high-order term을 포함시킴으로써 QDA와도 비슷한 성능을 보일 수 있음

# SVM

## Hyperplane

- $(p-1)$-dimensional affine subspace in a $p$-dimensional vector space
- $\beta$, $X$의 선형결합, 즉 $X\beta = 0$이라는 식에 의해서 이루어진 공간으로서 $\beta_0 = 0$인 경우 원점을 지나는 완전한 벡터공간, $\beta_0 \neq 0$인 경우 원점을 지나지 않는 Affine subspace
- 한편 $X\beta = 0$이므로, $\beta = (\beta_1, \ldots, \beta_p)$는 $X_1, X_2, \ldots, X_p$의 벡터들이 이루는 평면에 직교하는 normal vector이다.
- given sample을 separate하는 hyperplane은 infinitely many

## Maximal Margin Classifier

- separate되는 두 집단 사이의 거리를 가장 크게 만드는 classifier
- Constrained Optimization

  $\max_{\beta_0,\ldots,\beta_p} M$ → $M$: margin(separating hyper-plane과 closest data point 사이의 거리)

  - Constraints
    - $y_i \cdot f(x_i) \geq M$ where $f(x_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$
      - True value와 estimate이 서로 같은 방향(correct side)에 있으며 - since $M > 0$, hyperplane으로부터 적어도 $M$만큼 떨어져 있어야 한다.
        - ∵ misclassification이 발생했을 경우 두 곱은 음수가 되기 때문
    - $\sum_{i=1}^{p} \beta_j^2 = 1$
      - 이 값이 1이면 위의 constraint에서 $y_i \cdot f(x_i)$ 자체가 hyperplane $f(X)$과의 거리를 의미
- Hyper-plane과 margin 사이에 data point가 들어오는 것을 억제
- What if the data points are non-linearly separable or noisy?
  - the MMC can be very unreliable(has a high variance)

## Soft Margin SVM

- Hyper-plane과 margin 사이에 data point가 들어오는 것을 허용 → bias를 좀 허용하더라도 robust한 classification 결과를 얻고자 하는 것
- So introduce $\xi_i \geq 0$, a slack variable, to the first constraint so as to allow the points to be inside the margin $M$

$$
\begin{aligned}
\max_{\beta_0,\ldots,\beta_p,\xi_1,\ldots,\xi_n} \quad & M \\
\text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 = 1, \\
& y_i \cdot f\left(x_i\right) \geq M\left(1-\xi_i\right), \quad \forall i=1,\ldots,n \\
& \xi_i \geq 0, \qquad\qquad\qquad\quad \forall i=1,\ldots,n \\
& \sum_{i=1}^{n} \xi_i \leq C
\end{aligned}
$$

  - while $\xi_i$ cannot be infinitely large, add a constraint on the total amount of those slack variables: $\sum_{i=1}^{n} \xi_i \leq C$
    - this means that it allows a bit of misclassification, while try to obtain a robust classifier
  - As $C$ becomes larger, the margin becomes wider.
  - From this perspective, $C$ can be interpreted as a regularization parameter
- 그림 보고 non-zero slack을 갖는 data point 몇 개인지 찾는 문제

## Inner products and Linear SVC

- Support vector: hyperplane을 기준으로 가장 가까이에 있는 vectors
  - i.e. the data points placed within the margin
- Linear support vector classifier can be expressed as a linear combination of the inner products

$$
f\left(x\right) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle
$$

  - 여기서 $x$는 임의의 데이터, $x_i$는 기존의 training data를 나타냄
    - $f\left(x_1\right) = \beta_0 + \alpha_1 \langle x_1, x_2 \rangle + \cdots + \alpha_n \langle x_1, x_n \rangle$
    - $f\left(x_2\right) = \beta_0 + \alpha_1 \langle x_2, x_1 \rangle + \cdots + \alpha_n \langle x_2, x_n \rangle$
    - 이를 $x_1, \ldots, x_n$에 대해서 모두 작성하여 $\alpha_1, \alpha_2, \ldots, \alpha_n$을 구할 수 있음
    - 그런데 이때 inner product은 교환법칙이 성립하며, 자기 자신과의 내적은 제외하므로 전체 필요한 inner product의 개수는 $(n-1) + (n-2) + \cdots + 1 = \binom{n}{2} = \frac{n(n-1)}{2}$
  - 그리고 이때 inner product 앞의 계수는 대부분 0이 되는데, 이는 margin 바깥에 있는 observation에 대해서는 hyperplane이 고려하지 않기 때문이며, 대부분의 벡터는 support vector가 아니라는 것을 의미함

    - 따라서 전체 support vector들의 집합인 support set $\mathcal{S}$에 대해서 다음과 같이도 나타낼 수 있음

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i \langle x, x_i \rangle$$

- $\hat{\alpha}_i \neq 0$인 data point의 개수 == Support set $\mathcal{S}$의 크기 $|\mathcal{S}|$ == support vector의 개수 == data points within the margin(including the boundary)

## Kernel and SVM

- Kernel: a generalization of the inner product of the form $K(x_i, x_{i'})$
    - Ex. polynomial kernel of degree $d$

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

    - and the kernel can replace the inner product in the Linear SVC function
    - with the kernel, the hyperplane approach can be applied to the higher degree than the linear SVC
- One of the well-known kernels is the Radial kernel, with a tuning parameter $\gamma$

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right)$$

    - Test observation $x$가 기존의 training observation $x_i$로부터 많이 떨어져 있을 경우, $\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$의 값이 커지게 되고, $\exp$의 성질에 따라서 kernel의 값은 exponentially 줄어든다.
        - This means that $x_i$ has little influence on the classifcation of the test data $x$. - 즉 margin으로부터 떨어져 있는 point는 classifcation에 큰 영향이 없다
    - Radial kernel has very local behavior: test set 기준으로 가까이에 있는 training observation들이 영향을 미치며, 그 거리가 멀어지면 멀어질수록 영향력이 줄어든다.
    - What if the tuning parameter becomes smaller? Kernel의 값이 줄어드는 속도가 느려지며, more training observations have influence on the classification of the test data in other words.
        - which means that the decision boundary is smooth - high bias, low variance
    - Tuning parameter가 크면 반대로 decision boundary becomes wiggly; training point 하나 하나가 classifcation에 미치는 영향이 커지게 된다.

- Training data 기준으로 볼 경우, large $\gamma$에 대해서 더 좋은 성능이 나올 수도 있다 - why? low bias를 achieve할 수 있기 때문

## SVM for more than 2 classes

- OVA: One Versus All (or OVR: One Versus Rest)
  - $k = 1, \ldots, K$에 대해서 $k$th class와 나머지 $K-1$개의 class에 대해 한 번의 binary classification을 수행 → 총 $K$번의 binary classification을 수행하게 됨
    - 한 번 수행할 때마다 $k$th class에 대한 estimation $\hat{f}_k\left(x\right)$을 얻게 되며, 이렇게 모든 binary classification을 수행한 뒤 새로운 test data $x^*$에 대해서는 $\hat{f}_k\left(x^*\right)$가 가장 크게 되는 class로 분류한다.
- OVO: One Versus One
  - 서로 다른 2개의 class 모두 - 즉 $\binom{k}{2}$개 - 에 대해서 training을 수행 → 새로운 test data $x^*$에 대해서는 가장 많은 pairwise competition을 이기는 class로 분류
  - $K$가 많지 않으면 OVO가 좋다

## SVM vs Logistic

- SVM: margin의 바깥에 있는 값들에 대해서는 상관하지 않고, margin의 안쪽에 있는 값들에 대해서 regularized minimization
  - $f\left(X\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$에 대해서 SVC classifier optimization can be rephrased as

$$\min_{\beta_0, \beta_1, \ldots, \beta_p} \left\{ \sum_{i=1}^{n} \max\left[0, 1 - y_i f\left(x_i\right)\right] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

    - 0보다 작게 되면 0으로 취급 → support vector가 아니라는 의미
    - 즉 support vector에 대해서만 고려를 하여 loss minimization을 수행하겠다는 의미
    - $\max\left[0, 1 - y_i f\left(x_i\right)\right]$를 hinge loss라고도 부름
  - Hinge loss를 사용하기 때문에 SVM은 특히 class가 well-separated된 경우 loss를 정확히 0으로 만들 수 있다는 특징이 있으며, 다만 data가 dense한 경우에는 logistic regression과 SVM이 비슷하게 동작할 수 있다.
- Which to use
  - well-separated인 경우 → SVM performs better
  - more overlapping regimes → Logistic Regression performs better
  - wish to estimate probabilieis → Logistic Regression performs better
  - non-linear boundaries → kernel SVM is popular, while LR and kernel SVM can be used altogether