Python을 활용한 데이터 분석 강의

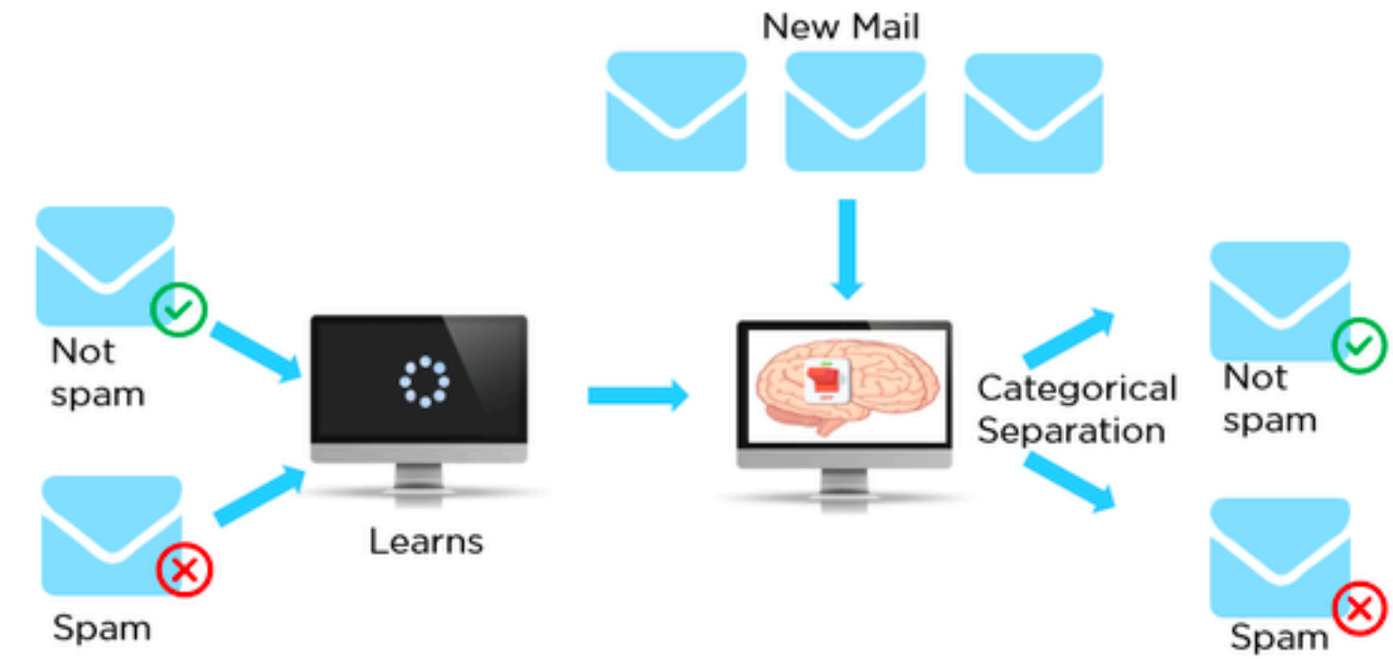
Supervised Learning

HYUNDAI MOBIS

Supervised Learning

- 답을 알고 있는 문제에 적용

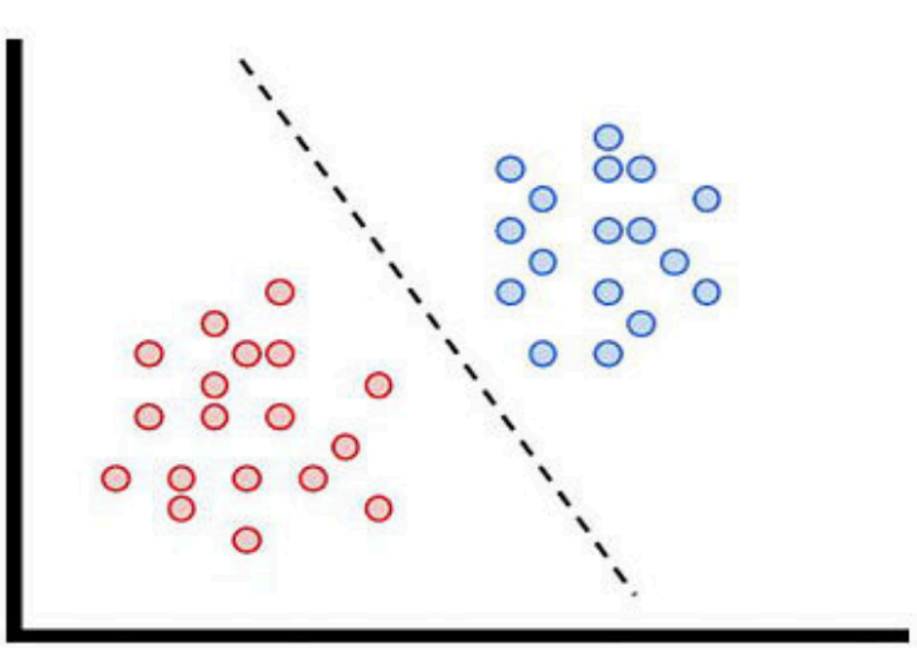
  - ex. spam 메일 걸러내기, titanic 승객 생존 여부 예측, 대출 심사



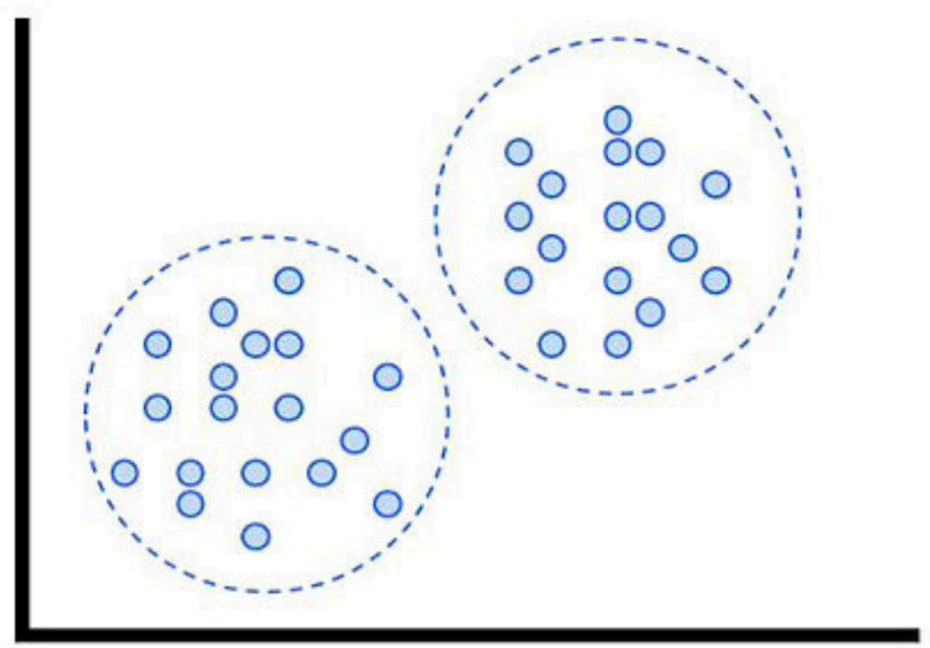
Enables the machine to be trained to classify observations into some class

Supervised Learning

- Classification



Classification



Clustering

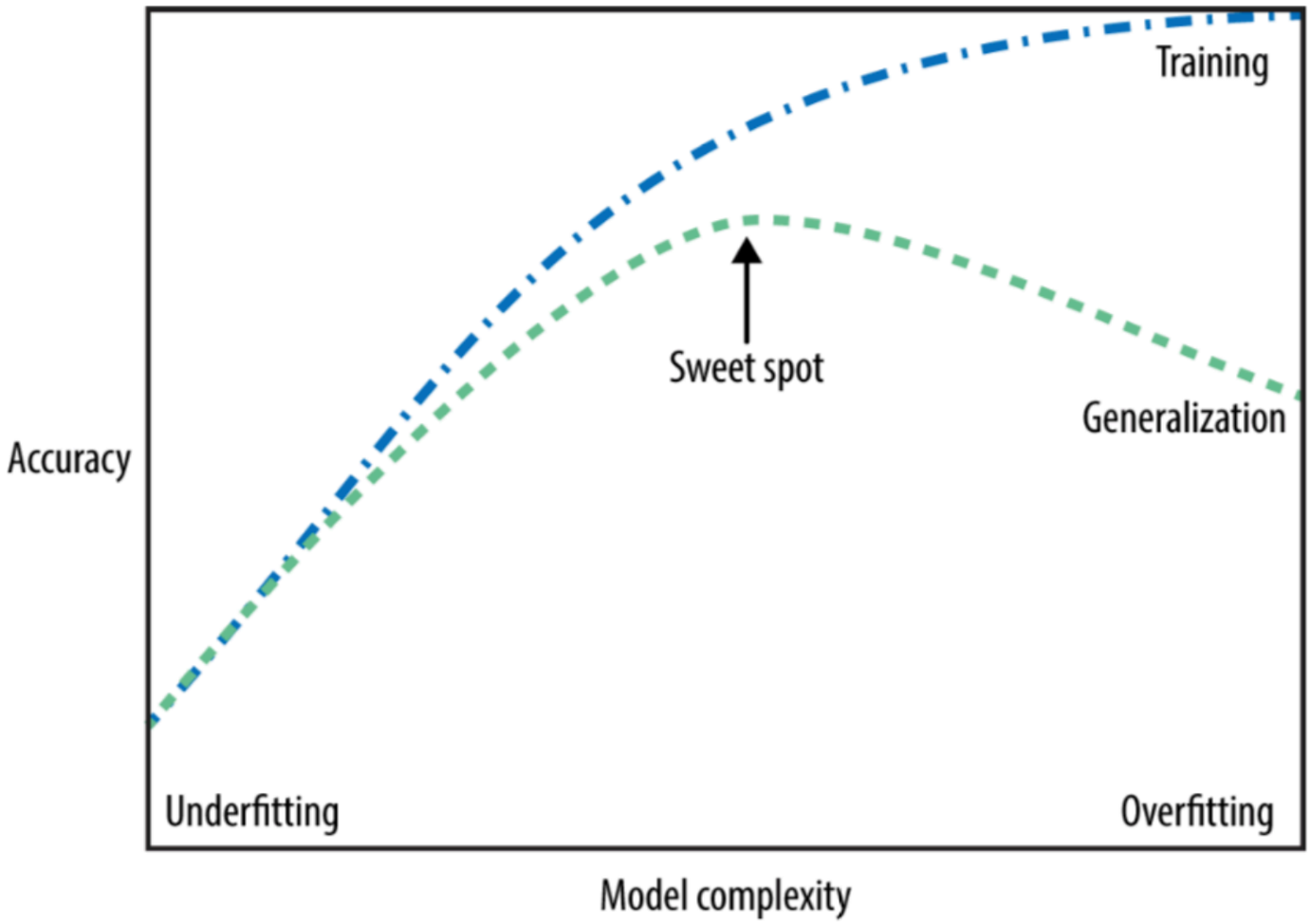
Supervised Learning

- Generalization
  - 만든 모델을 다른 데이터(새로운 데이터)에 적용시켰을 때 유효성이 있느냐

- Overfitting
  - train data set에 너무 fit된 모델이어서 다른 데이터에 적용시켰을 때의 정확도가 떨어지는 상태

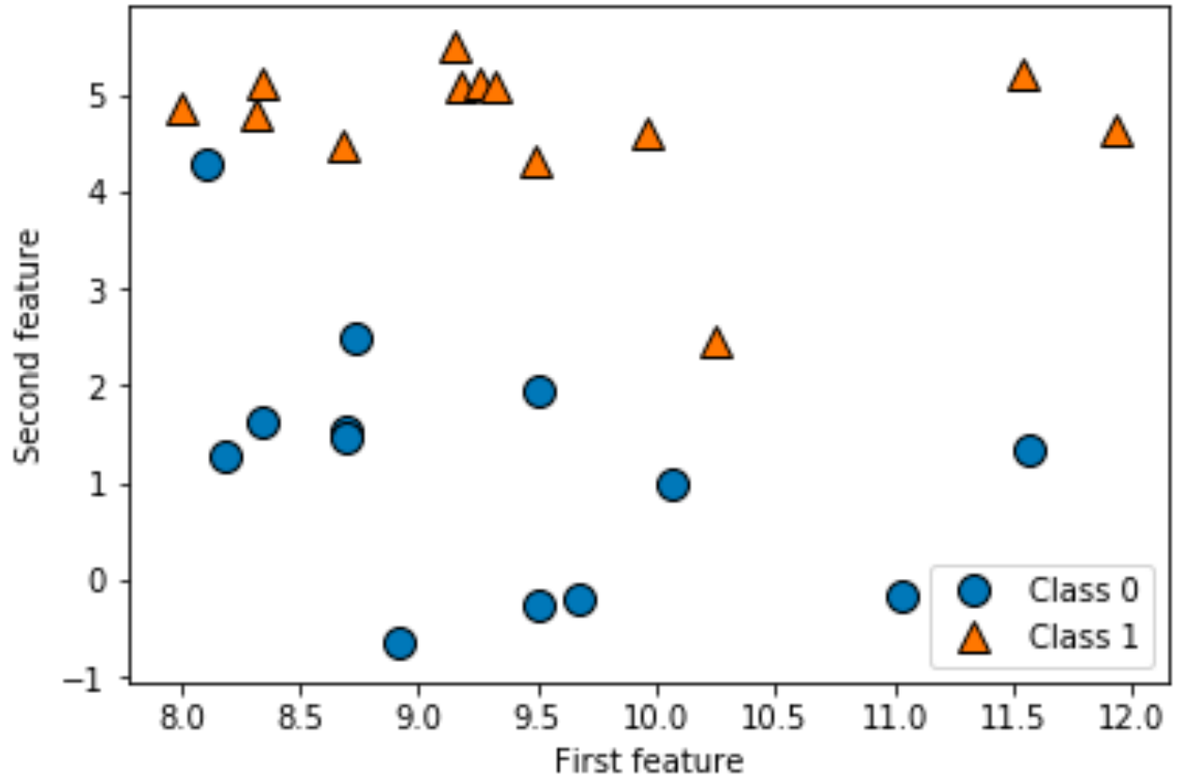
- Underfitting
  - 모델이 너무 간단/엉성해서 input과 output의 correlation을 잘 설명할 수 없는 상태

Supervised Learning



k-Nearest Neighbors

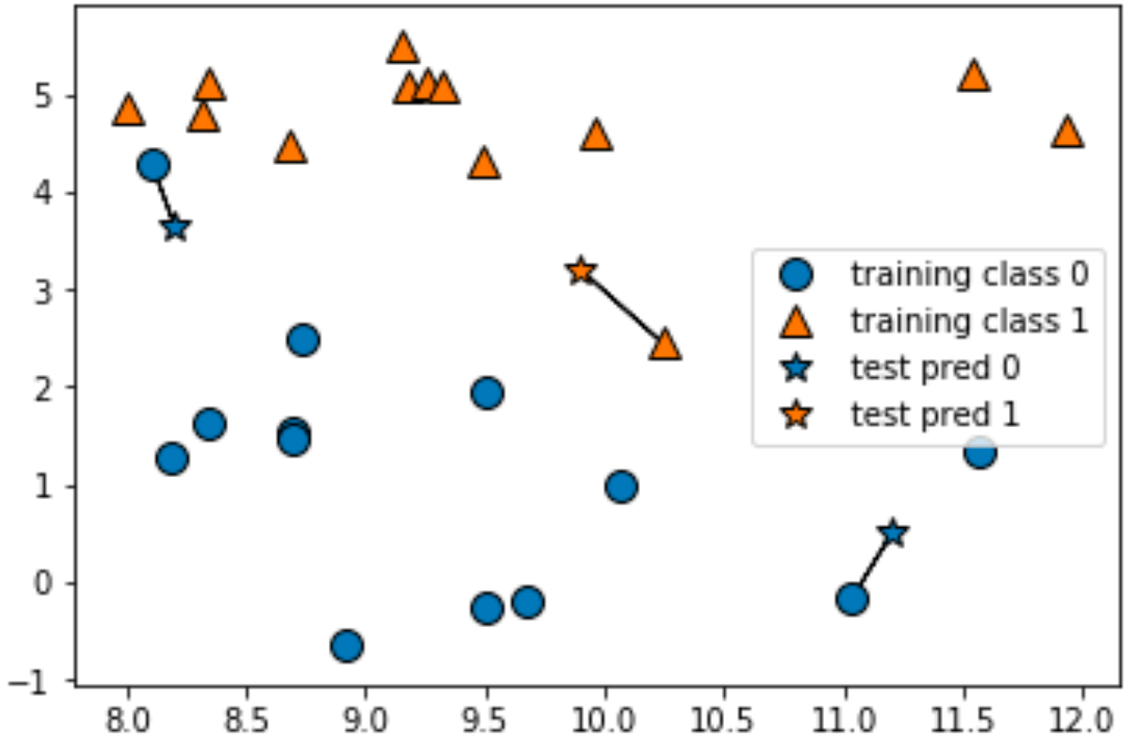
- 가장 간단한 ML 알고리즘 중 하나
- 새로운 데이터가 들어오면, train 데이터의 포인트들 중 가장 가까운 포인트를 찾아서 예측 - “nearest neighbors”



n_neighbors=1

k-Nearest Neighbors

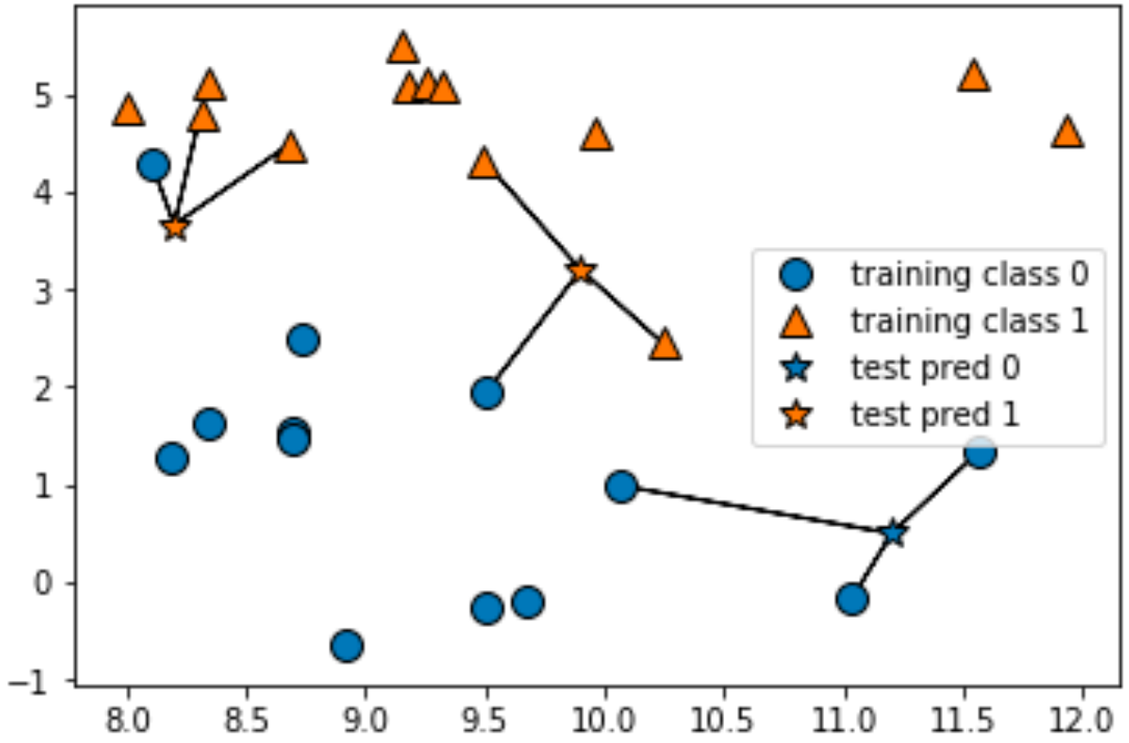
- 가장 간단한 ML 알고리즘 중 하나
- 새로운 데이터가 들어오면, train 데이터의 포인트들 중 가장 가까운 포인트를 찾아서 예측 - “nearest neighbors”



n_neighbors=1

k-Nearest Neighbors

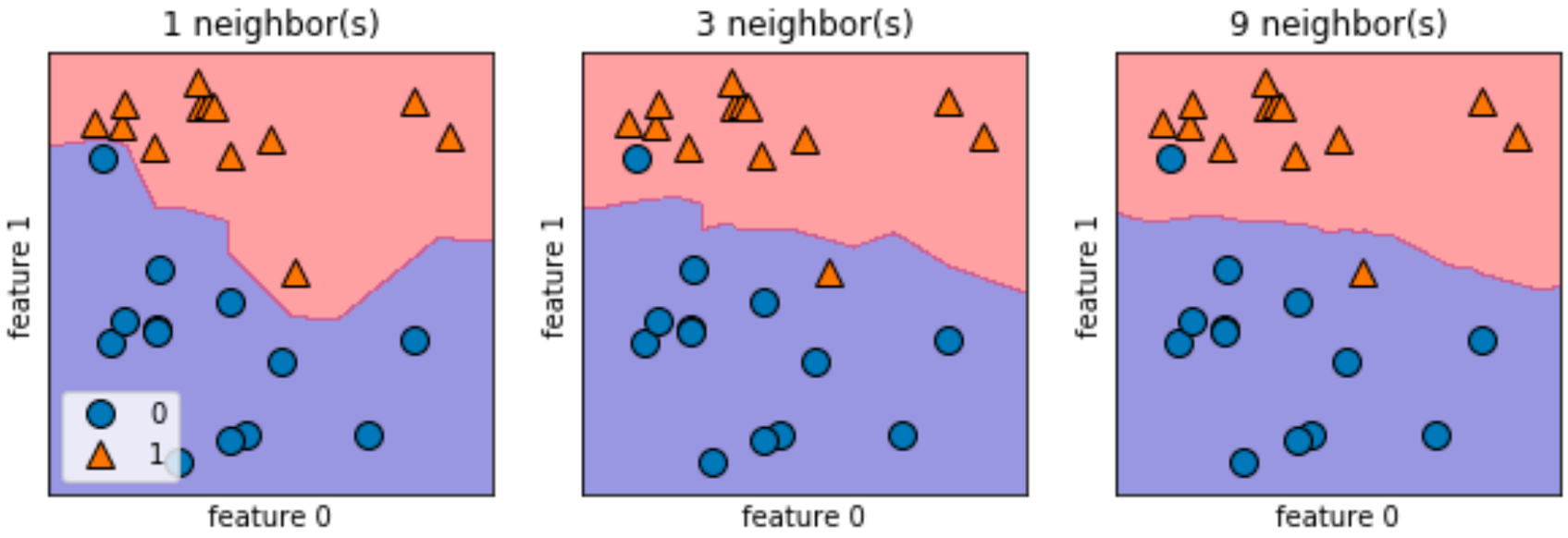
- 가장 간단한 ML 알고리즘 중 하나
- 새로운 데이터가 들어오면, train 데이터의 포인트들 중 가장 가까운 포인트를 찾아서 예측 - “nearest neighbors”



n_neighbors=3

k-Nearest Neighbors

- neighbor 개수를 늘릴수록 경계선이 부드러워진다
- 경계선이 부드럽다 = 모델이 simple하다



k-Nearest Neighbors

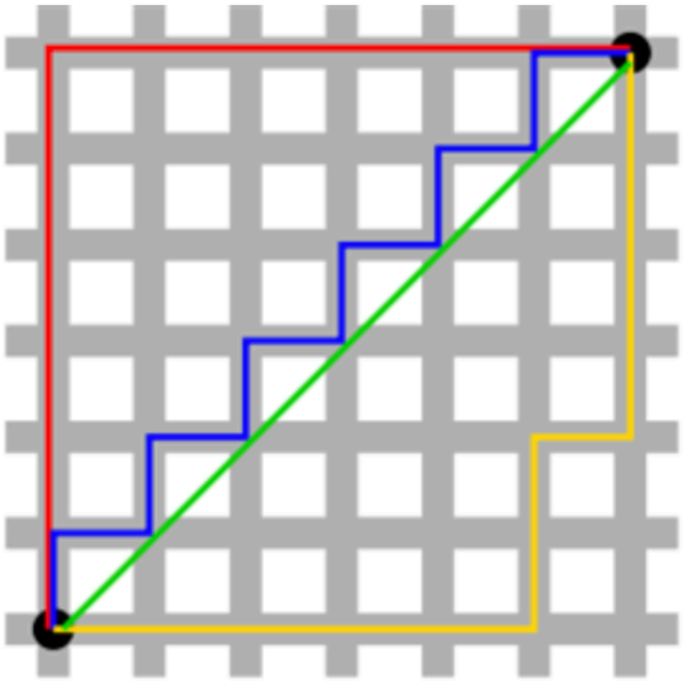
- Pros
  - 실행하기 쉬움
  - training 과정이 필요 없기 때문에 속도가 매우 빠름
  - 두 개의 parameter만 필요(# of neighbors, distance function(e.g. Euclidean or Manhattan))
- Cons
  - 높은 차원의 데이터에서는 잘 작동하지 않음(거리 계산이 어렵기 때문에)
  - 데이터가 많아질수록 cost 증가
  - 범주형 feature에서는 잘 작동하지 않음

cf) Euclidean & Manhattan

- Euclidean distance

$$L_2 = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \cdots + (p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$$

- Manhattan distance

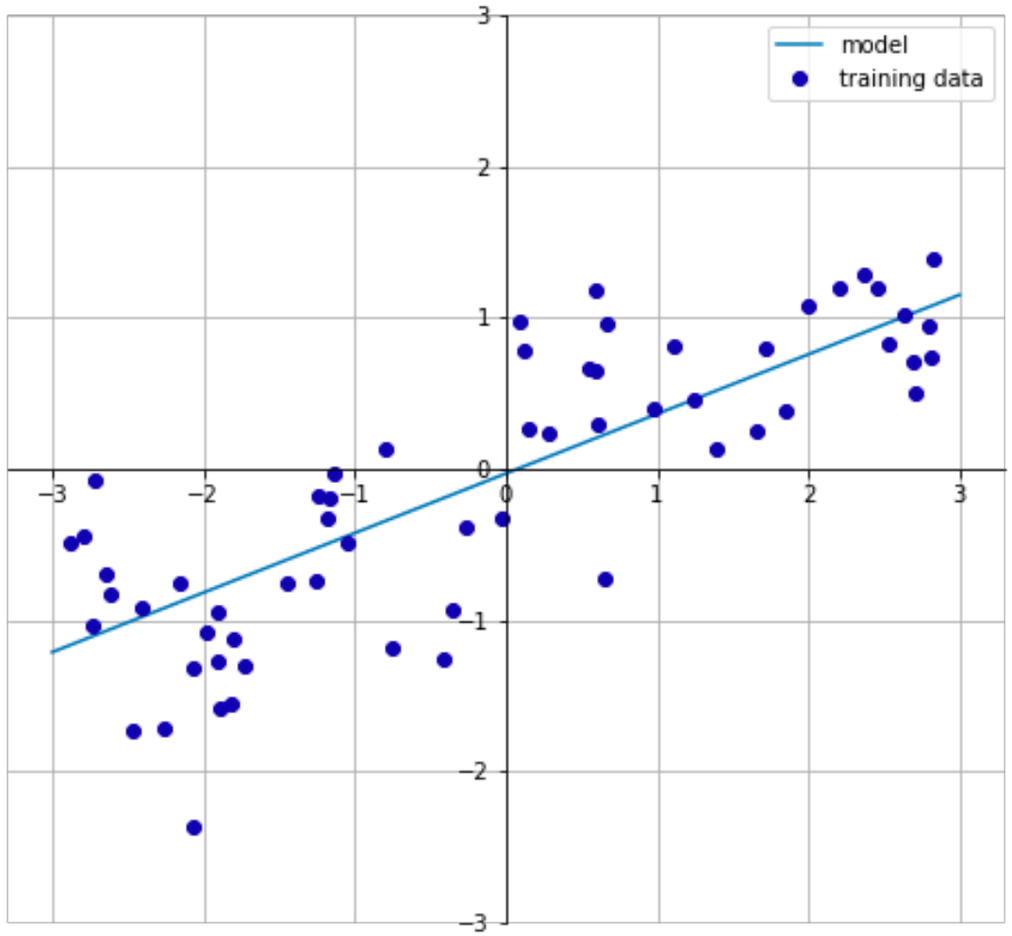


$$L_1 = |x_1-x_2| + |y_1-y_2|$$

HYUNDAI MOBIS

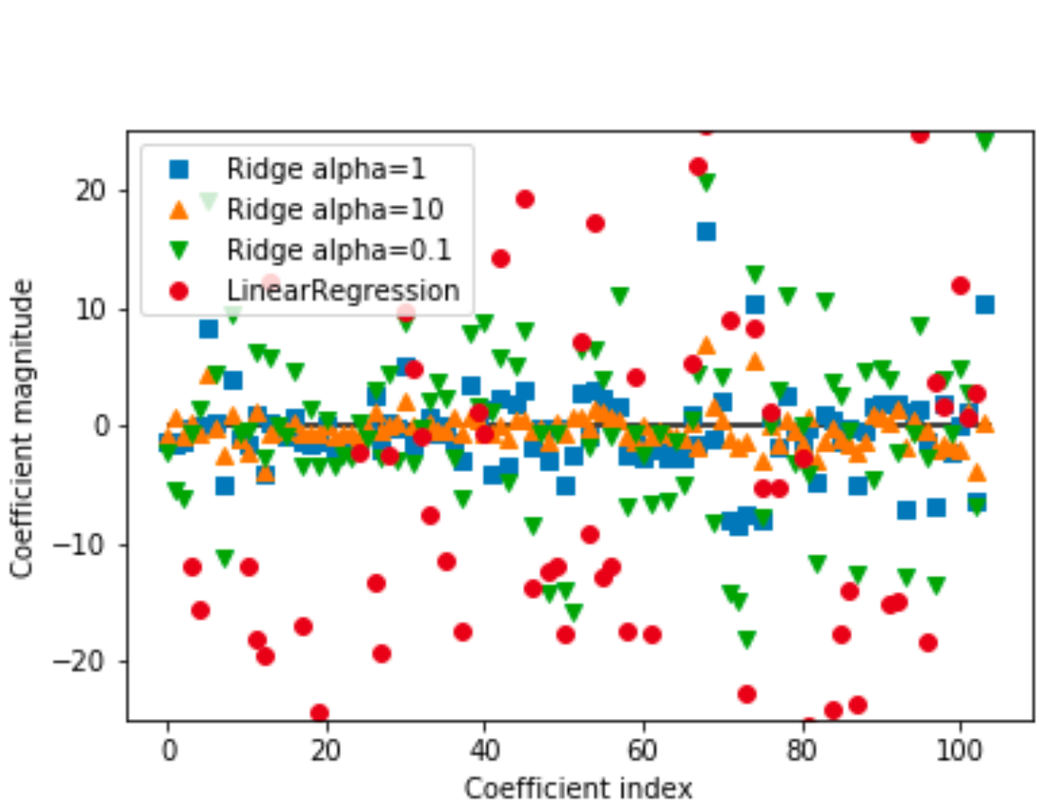
Linear Models

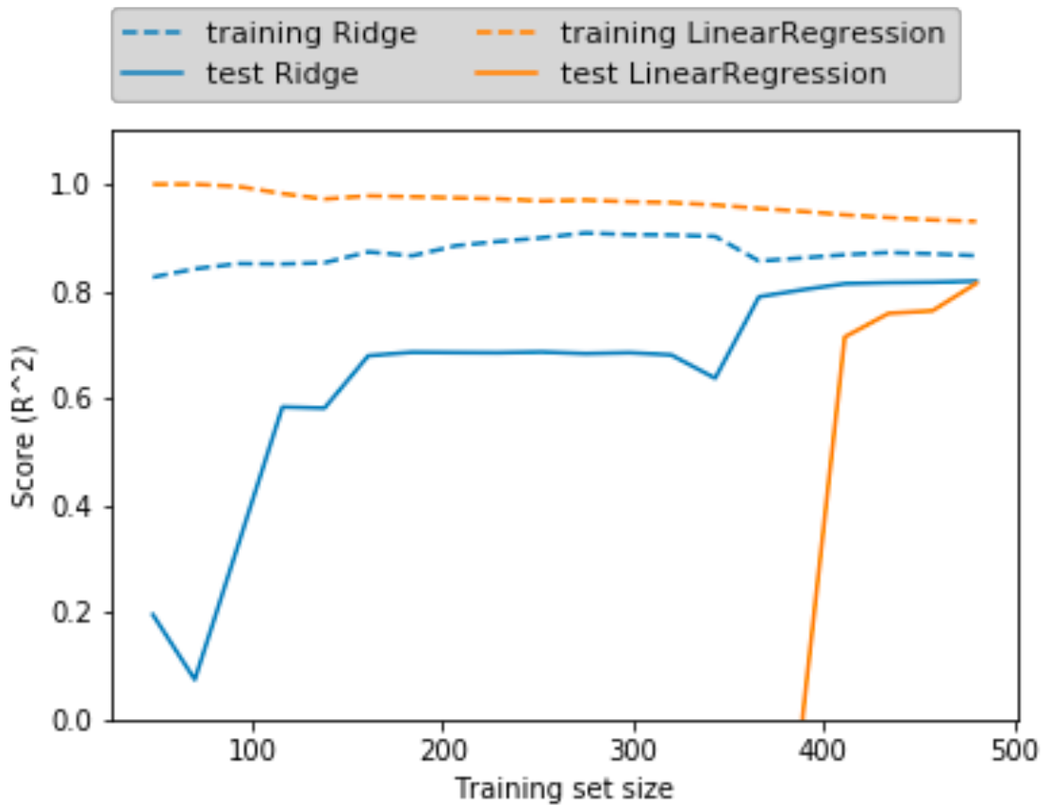
- 선형 모델
  - 가장 대표적인 것이 linear regression(OLS)



Linear Models

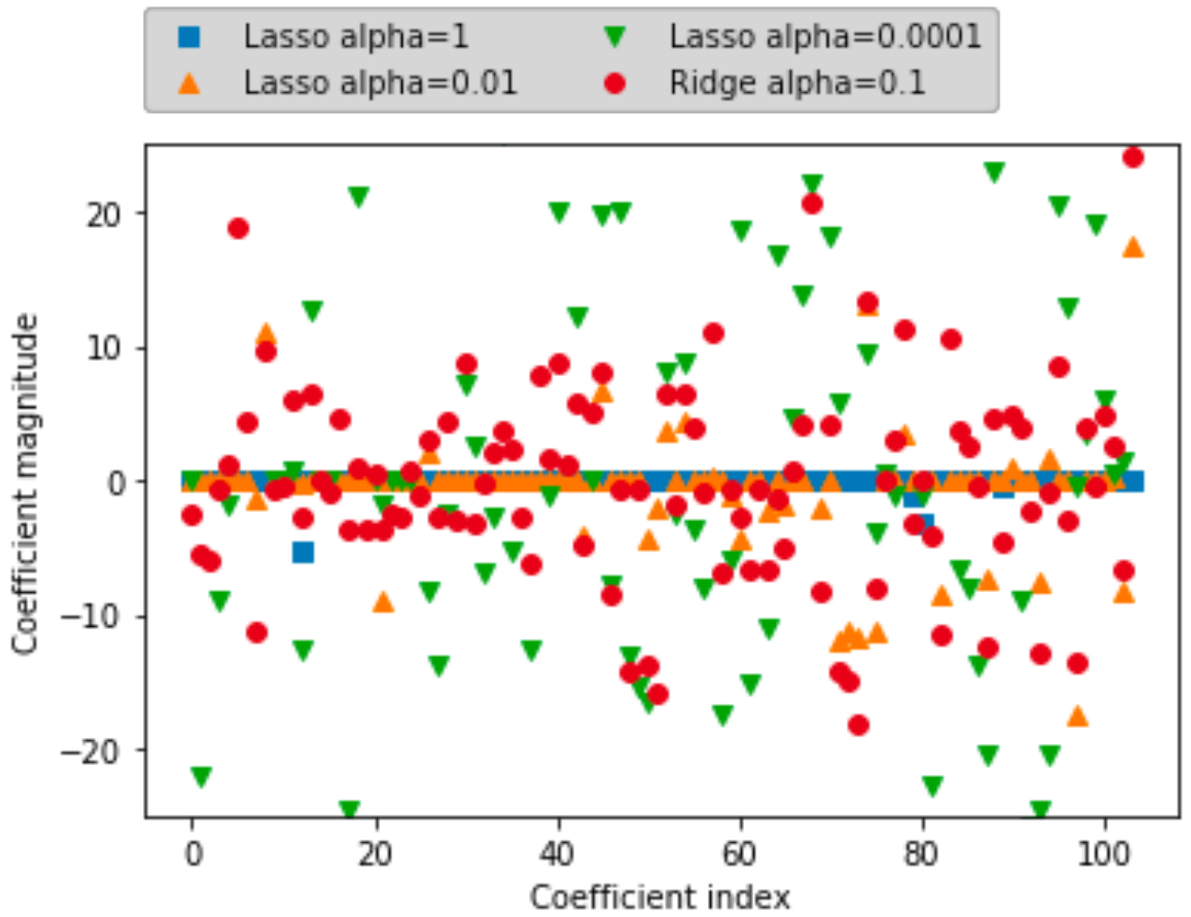
- regression에서 overfitting 문제가 발생했을 때
  - ridge





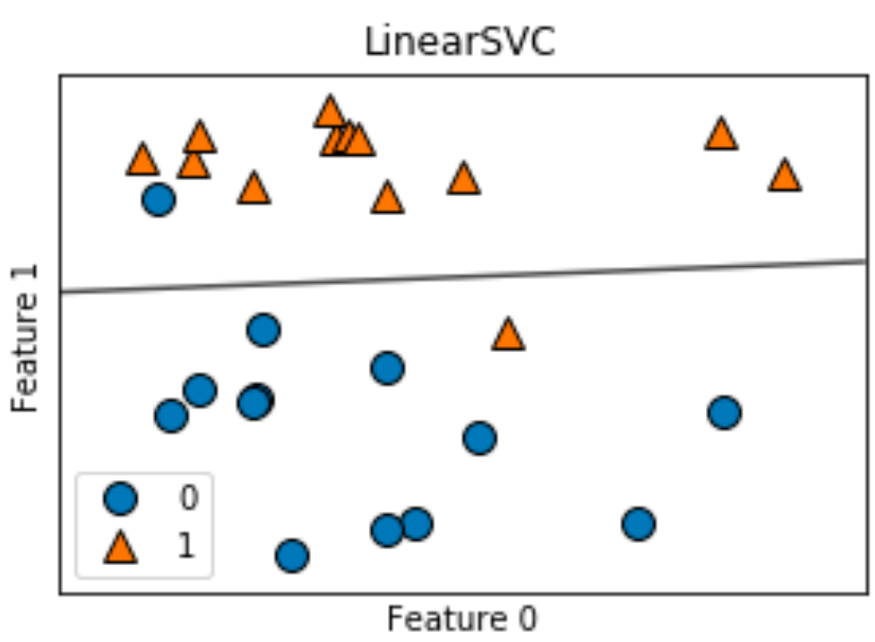
Linear Models

- regression에서 overfitting 문제가 발생했을 때
  - lasso



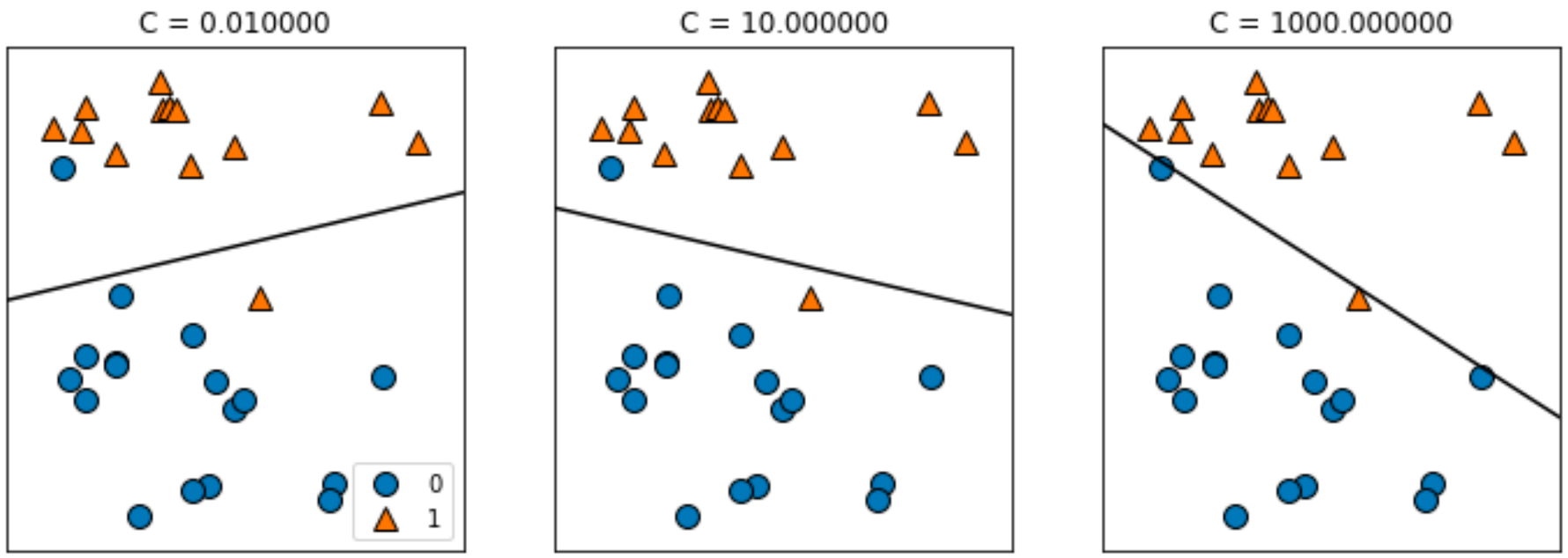
Linear Models

- Linear models for classification
  - Logistic regression
  - Linear SVMs(Support Vector Machines)



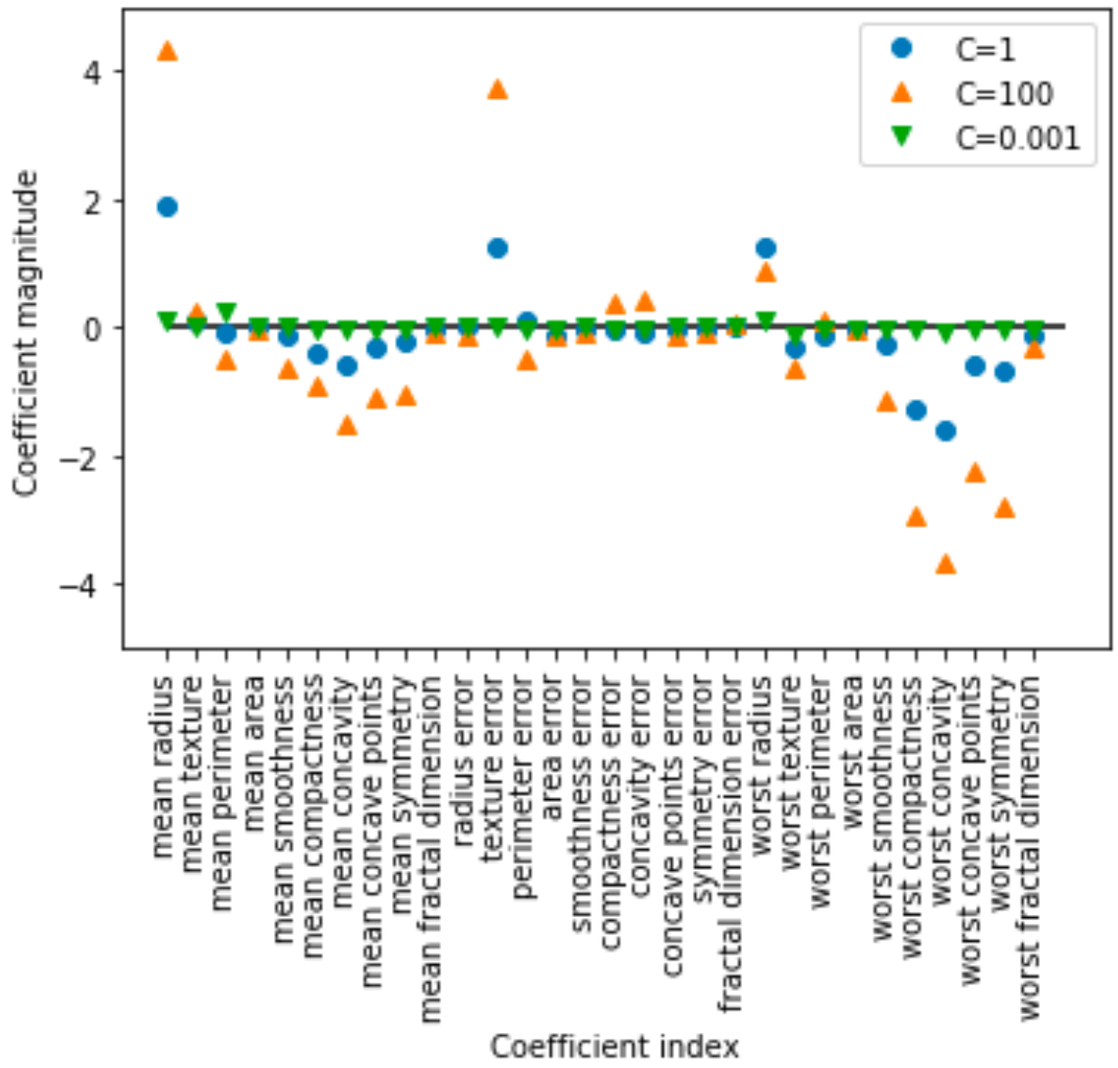
Linear Models

- L2 regularization : C



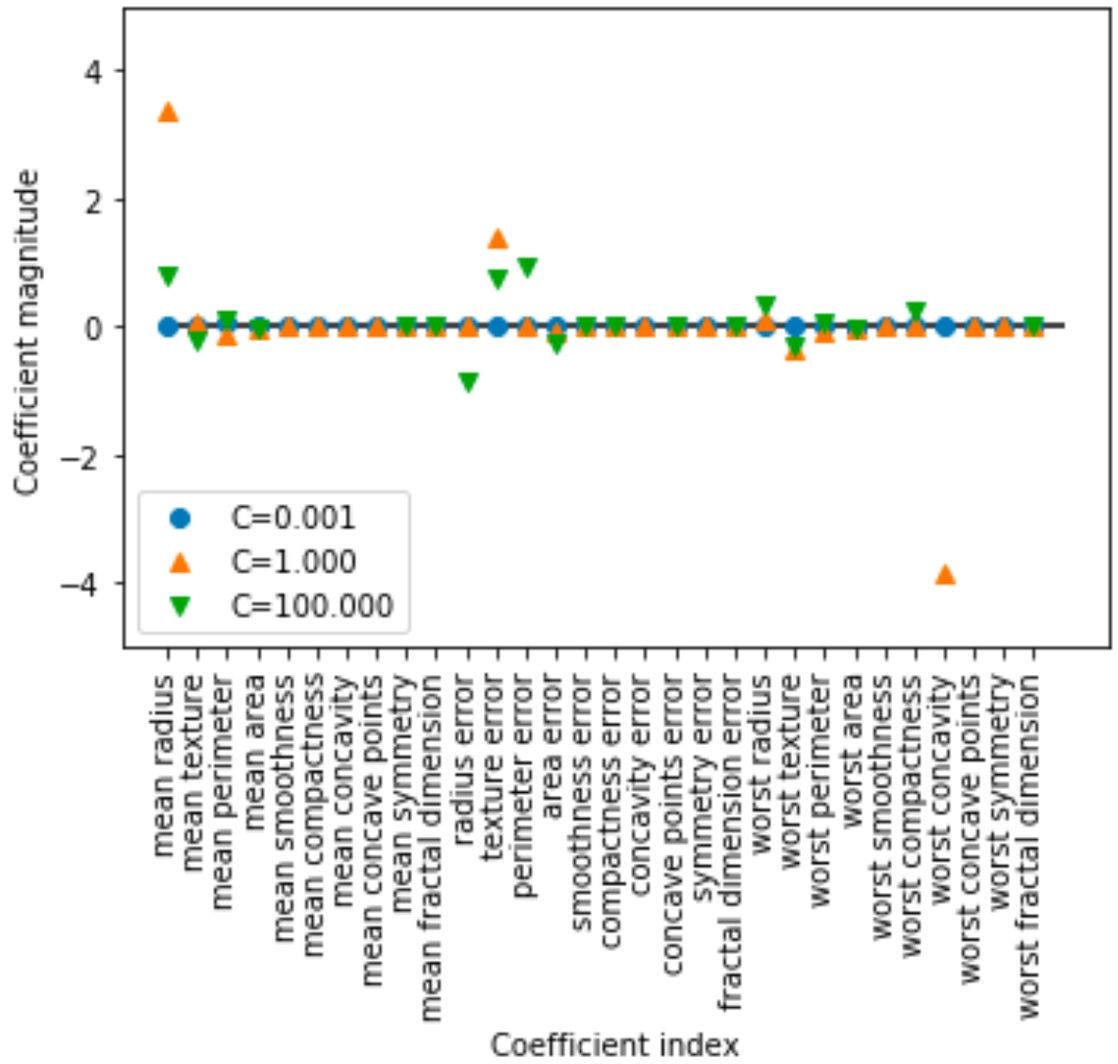
- C가 낮으면 : ‘대부분’의 데이터가 맞도록
- C가 높으면 : 각각의 데이터가 정확히 맞도록

Linear Models



Linear Models

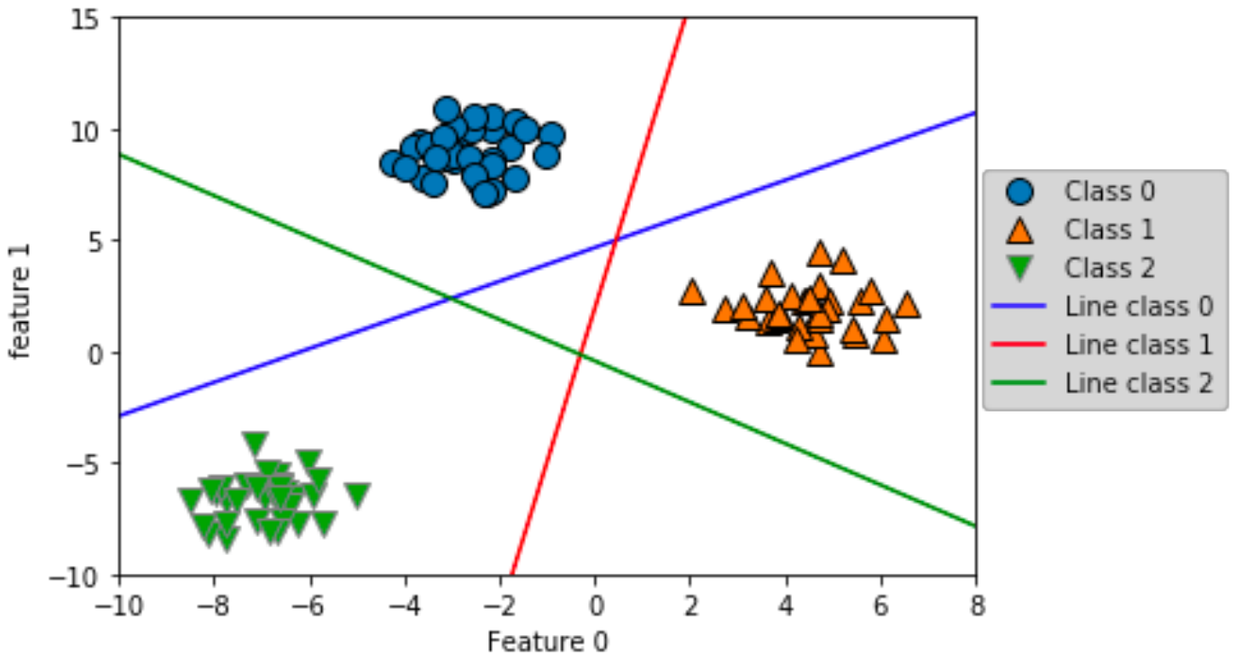
- L2 : default
- L1
  - feature들 중 몇 가지만 중요하다고 생각할 경우
  - 모델의 해석력이 중요한 경우 유용함(feature effect를 볼 때)

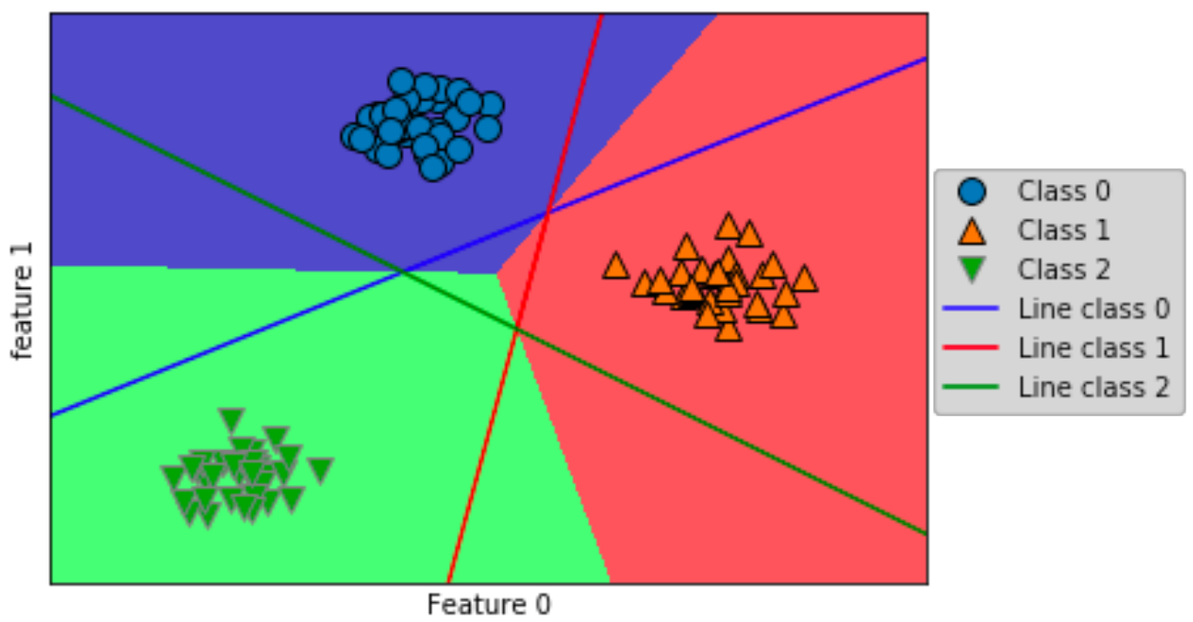


HYUNDAI MOBIS

Linear Models

- Linear models for multiclass classification





Linear Models

- Pros
  - 속도가 빠름
  - 실제 예측이 어떻게 이루어지는지 이해하기 쉬움(formula)

- Cons
  - coefficient가 왜 그렇게 도출되었는지 명확하지 않음
  - 변수들 간 correlation이 높을 경우 해석 문제가 생길 수 있음
    - multicollinearity

Naive Bayes Classifiers

- Naive? - 순수한, 순진한
  - 데이터의 모든 feature들이 동등하고 독립적이라고 가정

- GaussianNB
  - continuous data
- BernoulliNB
  - binary data
- MultinomialNB
  - count data(each feature represents an integer count)
    - ex. 단어 수 count

Bayes' theorem

- 두 확률 변수의 사전 확률과 사후 확률 사이의 관계를 나타내는 정리
- 사전 확률로부터 사후 확률을 구할 수 있음

- 조건부 확률

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

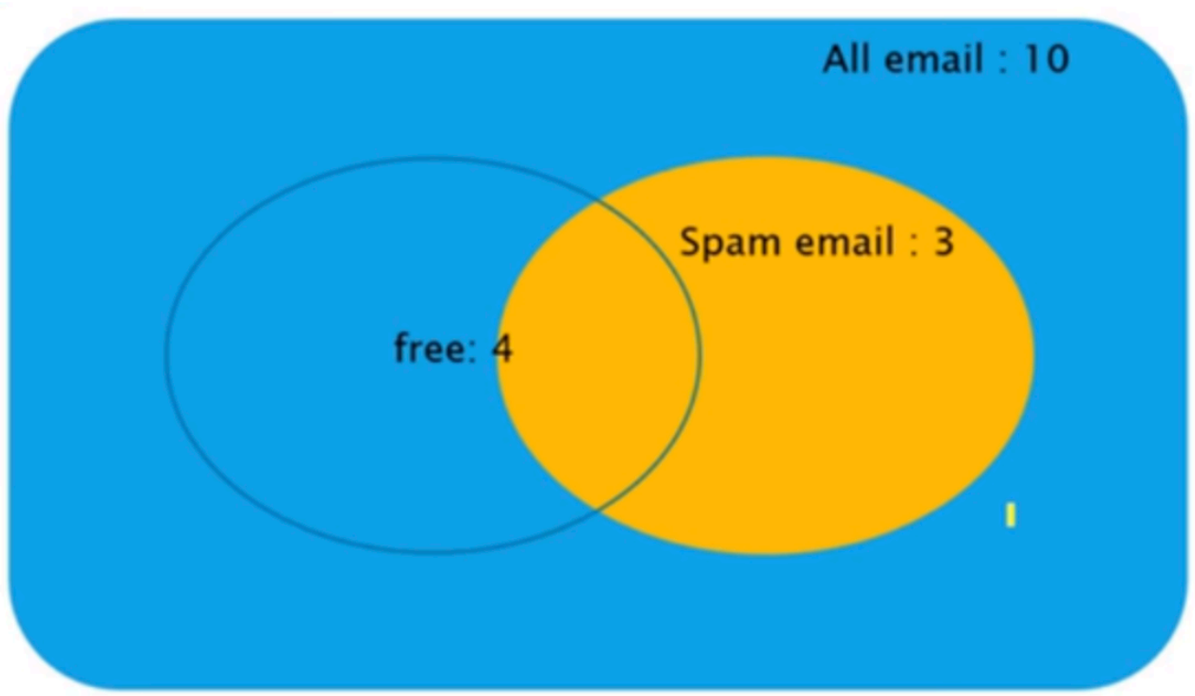
$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' theorem

- 조건부 확률

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

- P(spam|"free") = ?

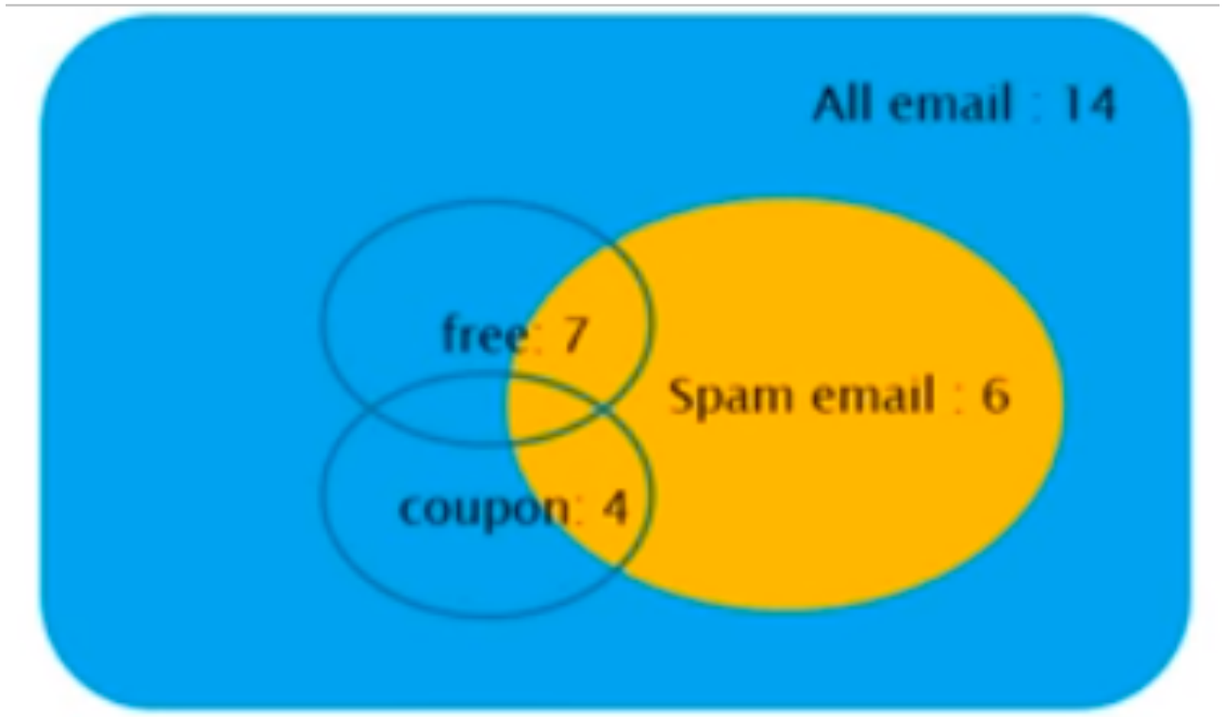


index	Email
1	I got free two movie ticket from your boy friend
2	free coupon from xx.com
3	watch free new movie from freemovie.com
4	Best deal, promo code here
5	There will be free pizza today 2pm meeting - your boss
6	Scheduled meeting tomorrow
7	Can we have lunch today?
8	I miss you
9	thanks my friend
10	It was good to see you today

HYUNDAI MOBIS

Bayes' theorem

- P(spam|"free"∩"coupon") = ?



index	Email
1	I got free two movie ticket from your boy friend
2	free coupon from xx.com
3	watch free new movie from freemovie.com
4	Best deal, promo code here
5	There will be free pizza
6	Scheduled meeting tomorrow
7	Can we have lunch today?
8	I miss you
9	thanks my friend
10	It was good to see you today
11	Free coupon, last deal
12	Free massage coupon
13	I sent the coupon you asked, it is not free
14	Coupon, promo code here!

Bayes' theorem

- Naive Bayes Algorithm을 사용하는 데 있어서, 독립적인 사전 요소(w0, w1, w2, w3, …)이 일어났을 때 사후 요소의 확률은

$$P(\text{spam} \mid w0,w1,w2..wn) = \frac{P(w0|\text{spam}) * \ldots P(wn|\text{spam}) * P(\text{spam})}{P(w0) * P(w1) * P(w2) * \ldots * P(wn)}$$

Bayes' theorem

Q. 쿠키가 들어 있는 그릇1, 그릇2

집었더니 바닐라 쿠키가 나왔다. 그릇1에서 나왔을 확률은?

Naive Bayes Classifiers

- Pros
  - 계산의 복잡성이 낮음(확률값만 있으면 식은 간단)
  - 높은 차원의 데이터에 적합

- Cons
  - feature의 수가 늘어날수록 0으로 수렴하는 경향
  - 가정 자체에 문제가 있을 수 있음(naive?)