KUGGLE WEEK04

회귀/분류

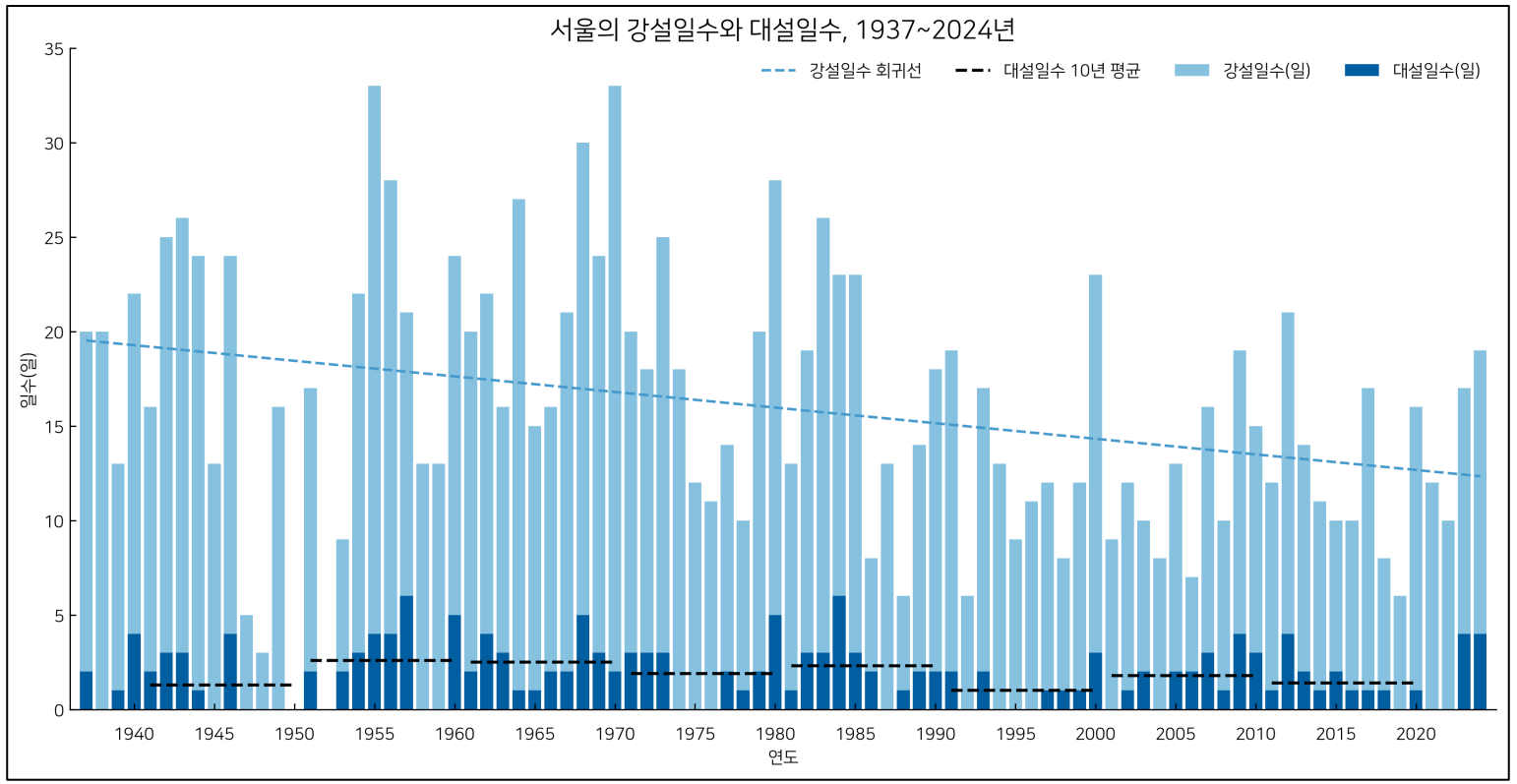
KUGGLE 11기
배지원 · 오영은

목차

회귀분석 정의 및 종류

회귀분석이란?
: 한 변수의 값으로부터 다른 변수의 값을 예측

ex) 시간 -> 서울의 강설일수



서울의 강설일수: 종속변수(반응변수)

시간: 독립변수(설명변수)

➔ 확률변수인 종속변수를 독립변수의 함수로서 설명하려는 것이 목적

회귀분석 정의 및 종류

회귀분석의 종류

- 독립변수 개수 -> 단순회귀 vs 다중회귀

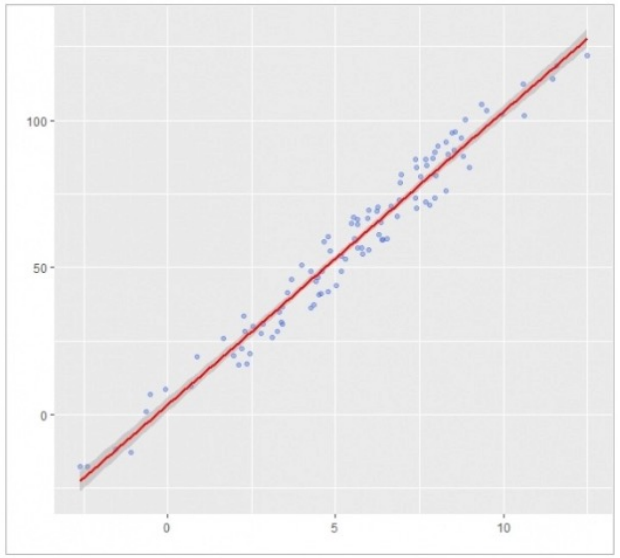
단순회귀

$$y = B_0 + B_1 x + \epsilon$$

다중회귀

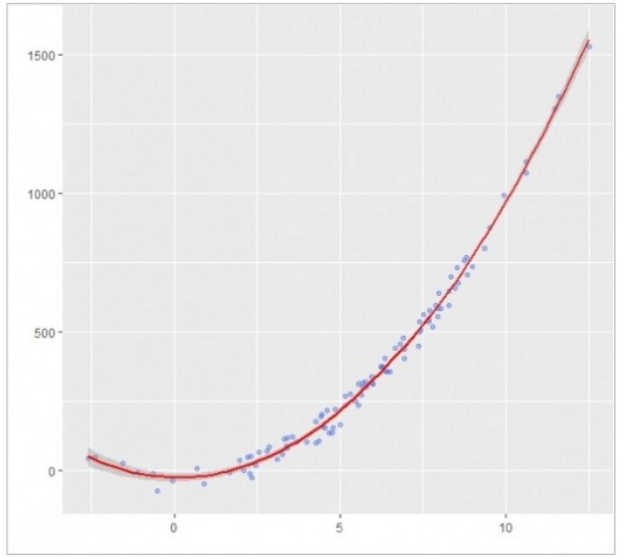
$$y = B_0 + B_1 x_1 + B_2 x_2 + \cdots + B_n x_n + \epsilon$$

- 결합 방식 -> 선형 vs 비선형



선형회귀

$$y = B_0 + B_1 x + \epsilon$$



비선형회귀

$$y = B_0 + B_1 x^2 + \epsilon$$

회귀 평가 지표

01. MSE(Mean Squared Error, 평균 제곱 오차)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- 실제 값과 예측 값의 차이를 제곱하여 평균을 낸 값
- 값이 작을수록 예측이 정확함을 의미
- 오차를 제곱하기 때문에 이상치에 민감

02. RMSE(Root Mean Squared Error , 평균 제곱근 오차)

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

- 단위가 종속변수와 같아 직관적 해석이 가능
- 여전히 이상치에 민감

03. MAE(Mean Absolute Error , 평균 절대 오차)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- 오차의 절댓값을 평균낸 값
- MSE보다 이상치의 영향을 덜 받음
- 미분이 어려워 최적화에 불리할 수 있음

04. R^2(결정계수, 설명력)

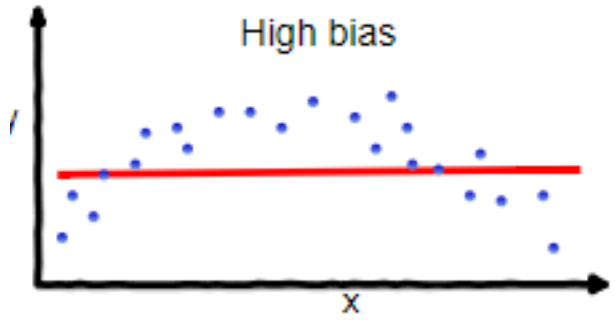
$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- 모델이 데이터를 얼마나 잘 설명하는지를 나타내는 지표
- 1에 가까울수록 모델의 설명력이 높음
- 0이면 모델이 무의미하며, 예측이 평균보다 못한 경우 음수도 나올 수 있음

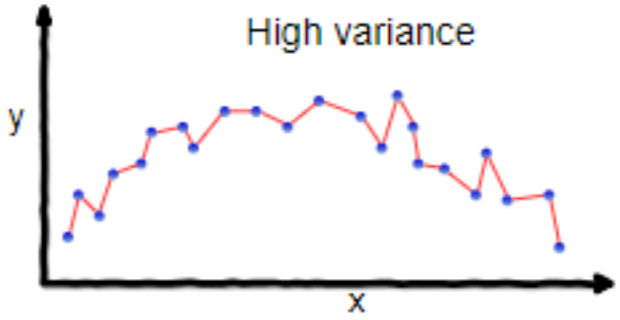
과대적합 & 과소적합

1. 편향과 분산(Bias-Variance Tradeoff)

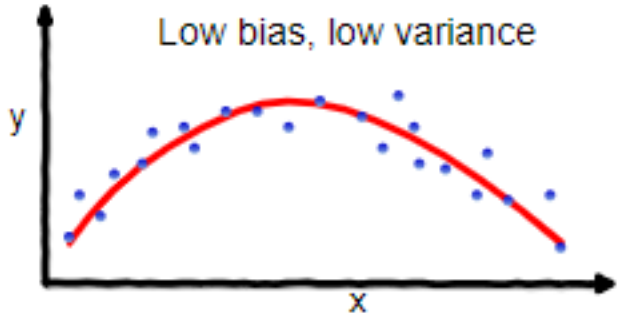
- 편향(Bias): 예측값(선)과 실제값(점) 간의 차이
- 분산(Variance): 동일한 모델에 다른 데이터셋 사용 시 예측값(선)의 변동가능성(복잡성)



편향이 크면 모델이 데이터의 패턴을 충분히 학습하지 못함
➔ 과소적합 발생



분산이 크면 모델이 훈련 데이터에 지나치게 맞춰짐
➔ 과대적합 발생

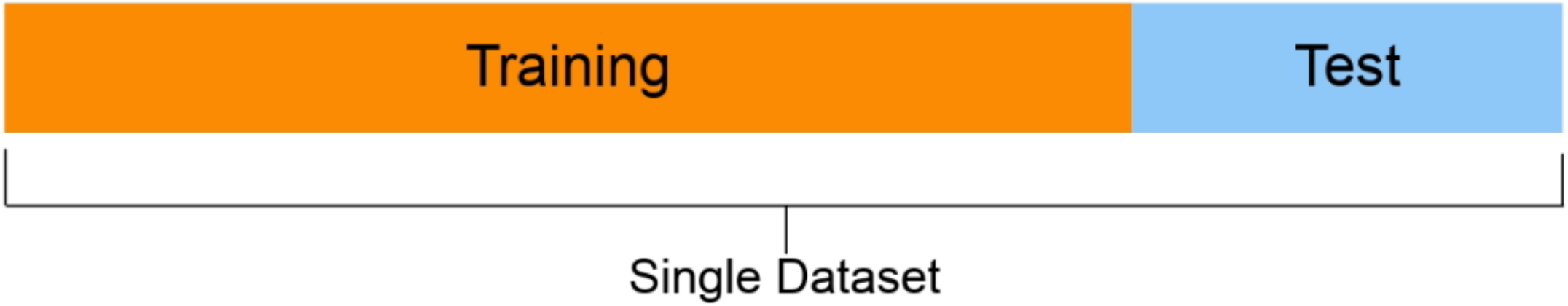


편향과 분산 적절히

과대적합 & 과소적합

2. 훈련 세트와 테스트 세트

- 훈련 세트를 활용하여 모델 훈련 후 훈련 세트와 테스트 세트에서 평가
- 훈련 세트와 테스트 세트를 보통 7:3, 8:2로 분류



- 훈련 세트로 훈련했기 때문에 훈련 세트의 점수가 더 높게 나오는 경향이 있음
  (score: 결정계수 값, 1에 가까울수록 좋음)
- 훈련 세트 점수가 테스트 세트 점수보다 훨씬 높은 경우 -> 과대적합
- 테스트 세트 점수가 훈련 세트 점수보다 높거나, 두 점수 모두 낮은 경우 -> 과소적합

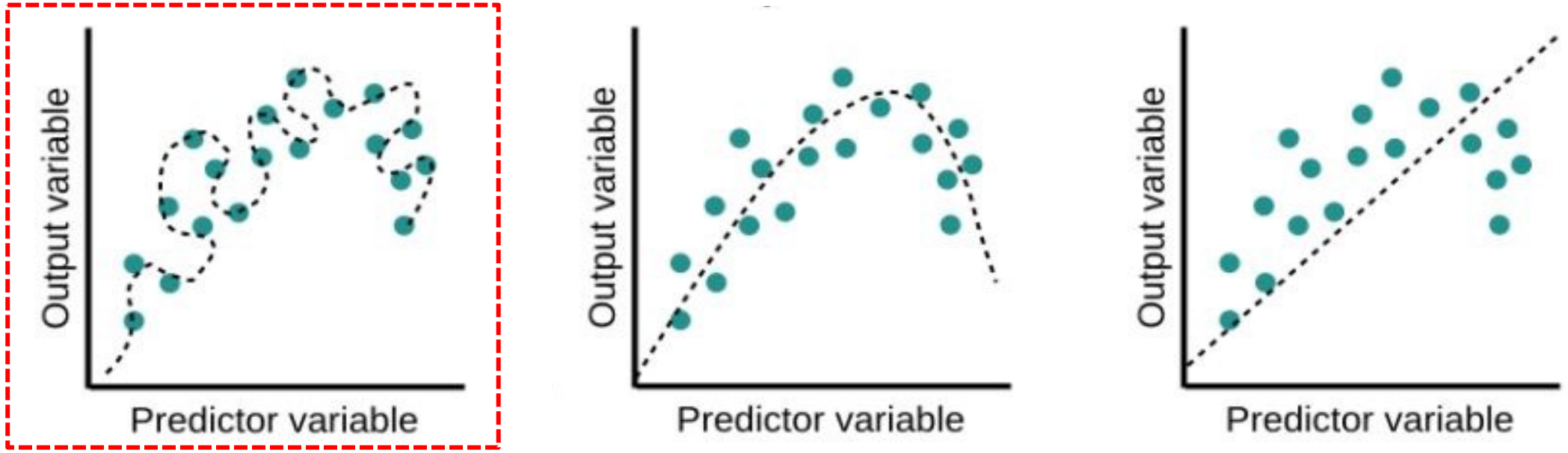
과대적합 & 과소적합

3. 과대적합(Overfitting)

- 모델이 훈련 세트를 너무 잘 학습해서, 노이즈까지 학습하는 현상

- 훈련 세트에 대한 성능 >> 테스트 데이터에 대한 성능
- 설명 변수가 너무 많음

- 복잡한 모델에서 자주 발생 (ex. 너무 많은 변수, 너무 깊은 신경망 등)



과대적합 & 과소적합

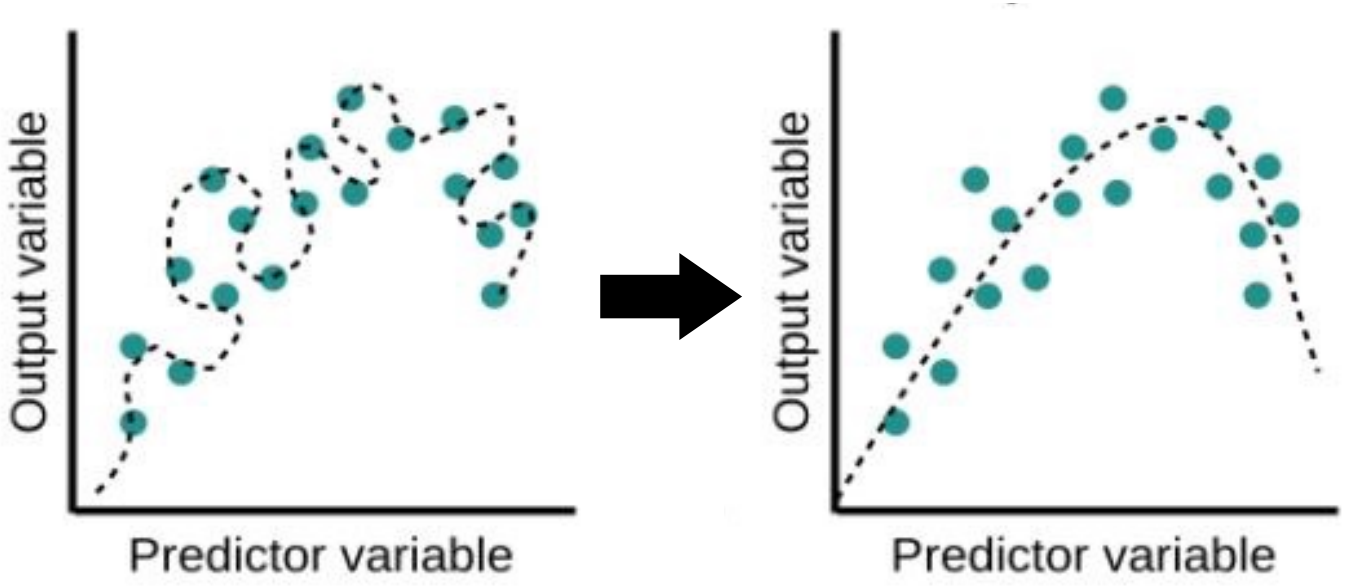
4. 과대적합(Overfitting) 해결 방법

(1) 모델 복잡도 감소 → 모델 단순화

(2) 더 많은 데이터 사용

(3) Early Stopping 적용 → 과대적합 전에 학습 중단

(4) 정규화 적용 → L1, L2 정규화 활용



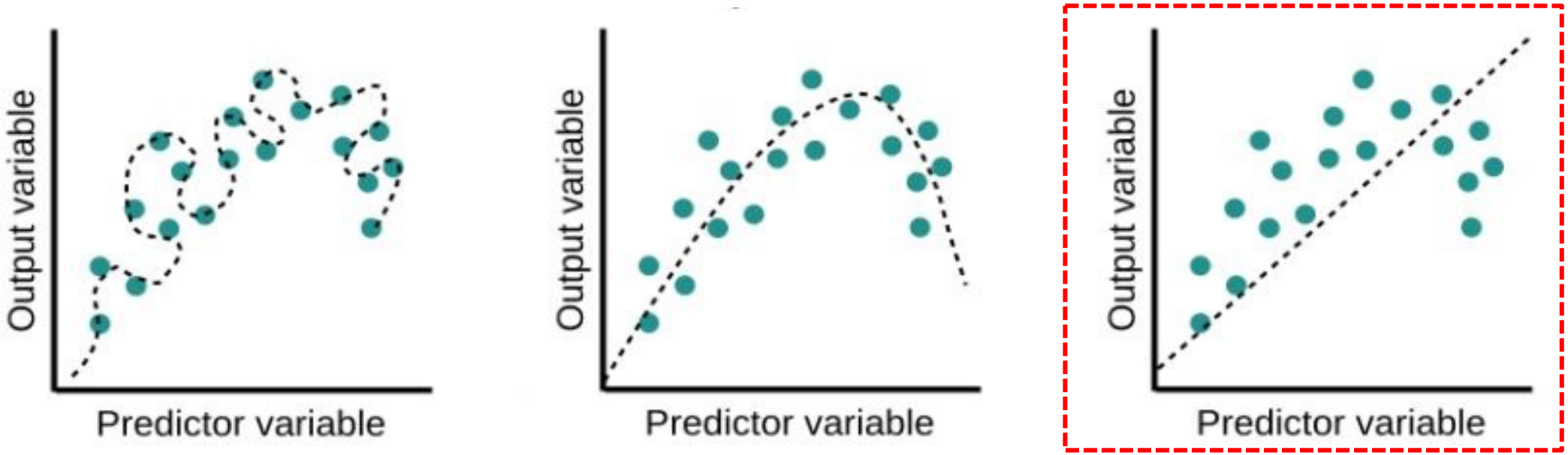
과대적합 & 과소적합

5. 과소적합(Underfitting)

- 모델이 데이터의 패턴을 충분히 학습하지 못함

- 훈련 세트에 대한 성능 << 테스트 세트에 대한 성능
- 훈련 세트에 대한 성능 ↓ 테스트 세트에 대한 성능 ↓

- 너무 단순한 모델에서 자주 발생 (ex. 선형 회귀로 비선형 관계를 학습하려는 경우)

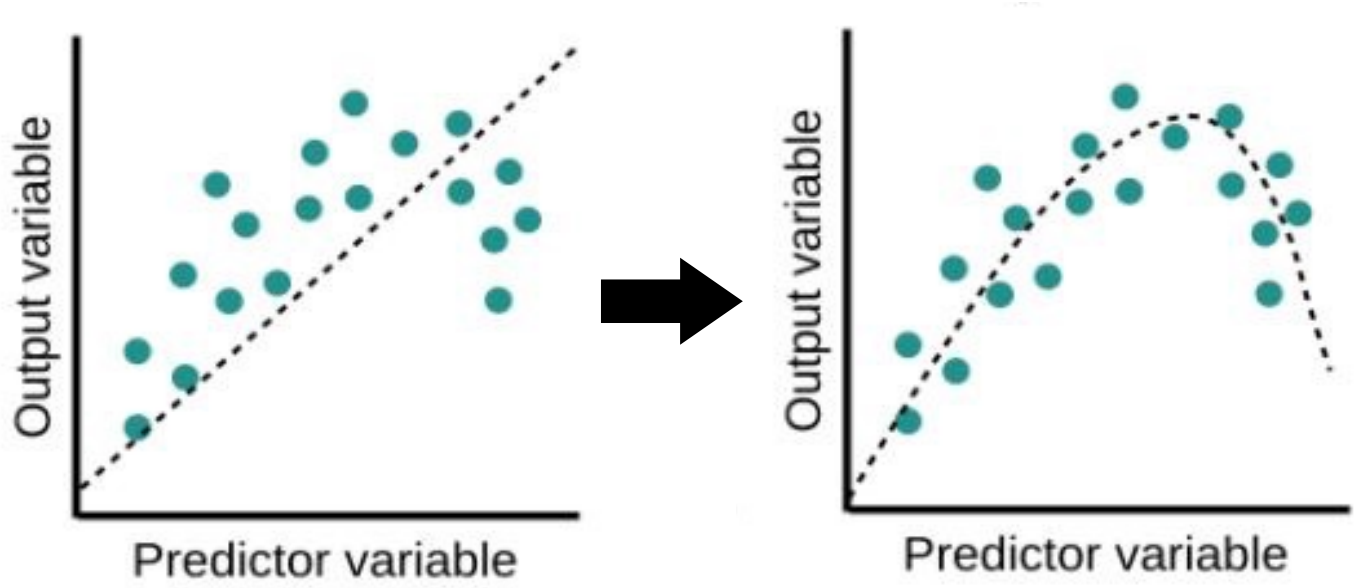


과대적합 & 과소적합

6. 과소적합(Underfitting) 해결 방법

(1) 모델 복잡도 증가 → 더 복잡한 모델 사용

(2) 더 많은 Feature 추가 → 유의미한 변수 추가



파이썬을 활용한 회귀

< LinearRegression을 이용한 예측 코드 >

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# 데이터 준비
X =
y =

# 데이터 분할
# Test size는 0.3 (난수는 자유)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

파이썬을 활용한 회귀

< LinearRegression을 이용한 예측 코드 >

```
# 모델 생성
lr = LinearRegression()

# 학습
lr.fit(X_train, y_train)

# 예측
y_preds = lr.predict(X_test)

# 평가
mse = mean_squared_error(y_test, y_preds)
rmse = np.sqrt(mse)
```

파이썬을 활용한 회귀

< LinearRegression을 이용한 예측 코드 >

```
# MSE값 출력
print('MSE: {0:.3f}, RMSE:{1:3f}'.format(mse, rmse))
print('Variance score:
{0:.3f}'.format(r2_score(y_test,y_preds)))

# 절편 값
print('절편 값:', lr.intercept_)

# 회귀계수 값
print('회귀계수 값:', np.round(lr.coef_, 1)
```

파이썬을 활용한 회귀

< 과적합 방지를 위한 릿지 모델 >

```
from sklearn.linear_model import Ridge

ridge = Ridge() # 릿지 모델 불러오기
ridge.fit(train_scaled, train_target) # 모델 훈련하기
print(ridge.score(train_scaled. train_target)) # 훈련세트 점수
print(ridge.score(train_scaled, test_target)) # 테스트세트 점
수

# Alpha 값으로 규제의 강도 조절 가능
ridge = Ridge(alpha=10)
```

파이썬을 활용한 회귀

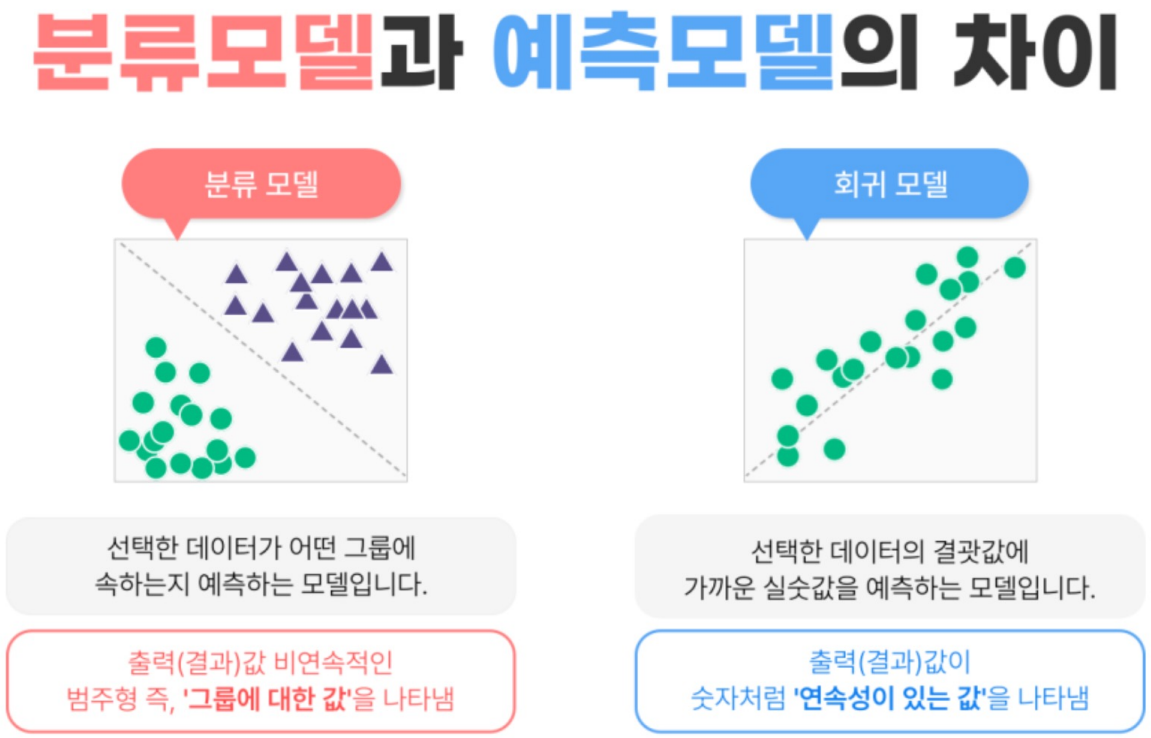
< 과적합 방지를 위한 라쏘 모델 >

```
from sklearn.linear_model import Lasso
Lasso = Lasso()
lasso.fit(train_scaled, train_target) # 모델 훈련하기
print(lasso.score(train_scaled, train_target)) # 훈련세트 점수
print(lasso.score(train_scaled, test_target)) # 테스트세트 점
수

# Alpha 값으로 규제의 강도 조절 가능
lasso = Lasso(alpha=10)
```

분류의 정의 및 종류

분류란?
: 지도학습의 일종으로 데이터를 미리 정의된
여러 범주로 분류하는 것



분류 모델 (Classification)	**비연속적인 범주형 값을 분류하는 것** 출력값이 연속성이 없음 (숫자가 아닌 이름과 문자 형태) - ex : 사과와 바나나, 포도처럼 중간값이 없음
회귀 모델 (Regression)	**연속된 값을 예측하는 것** 출력값이 연속성이 있음 (숫자와 같이 1,2,3... 등)

모델 유형 구분		출력값	
분류 모델	내일 날씨가 추울까요? 더울까요?	덥다 (혹은) 춥다	→ **문자 형태** (비연속성)
회귀 모델	내일 기온은 몇 도일까요?	32도	→ **숫자 형태** (연속성)

분류의 정의 및 종류

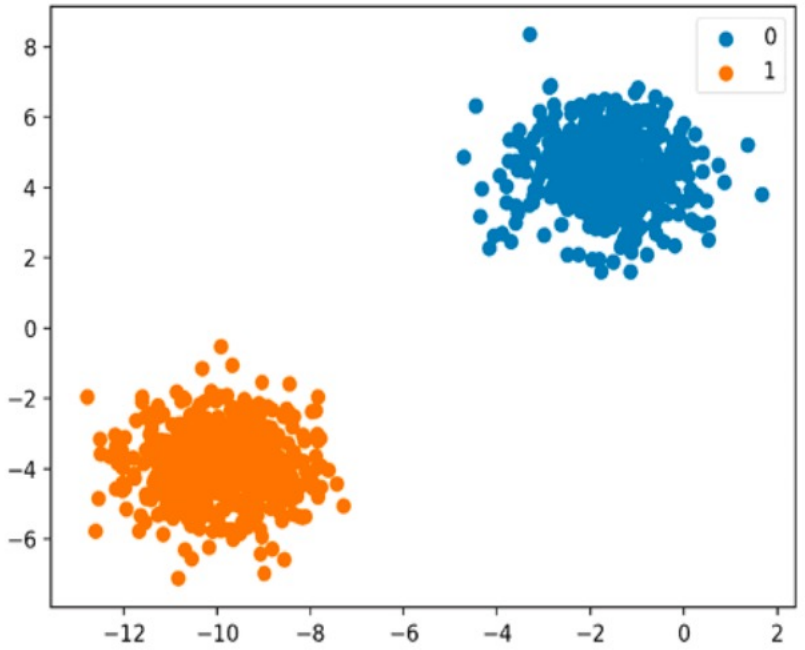
분류의 종류

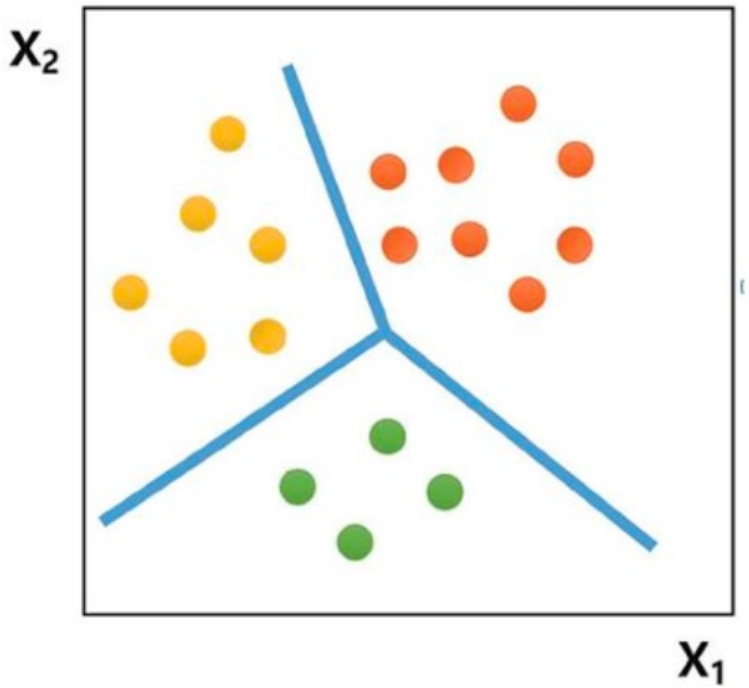
- 이진 분류

  분류 카테고리가 두 가지인 경우 사용 (0 or 1)

- 다중 분류

  분류 카테고리가 여러 가지인 경우 사용

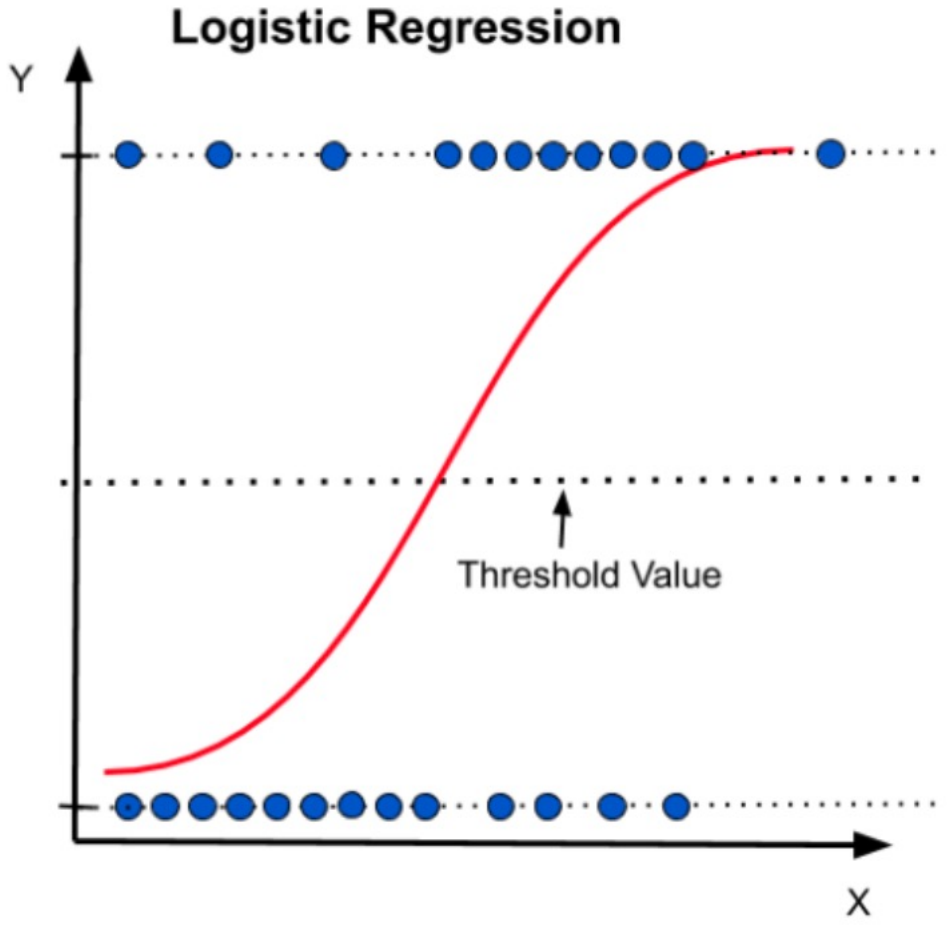




분류 알고리즘

로지스틱 회귀 (Logistic Regression)

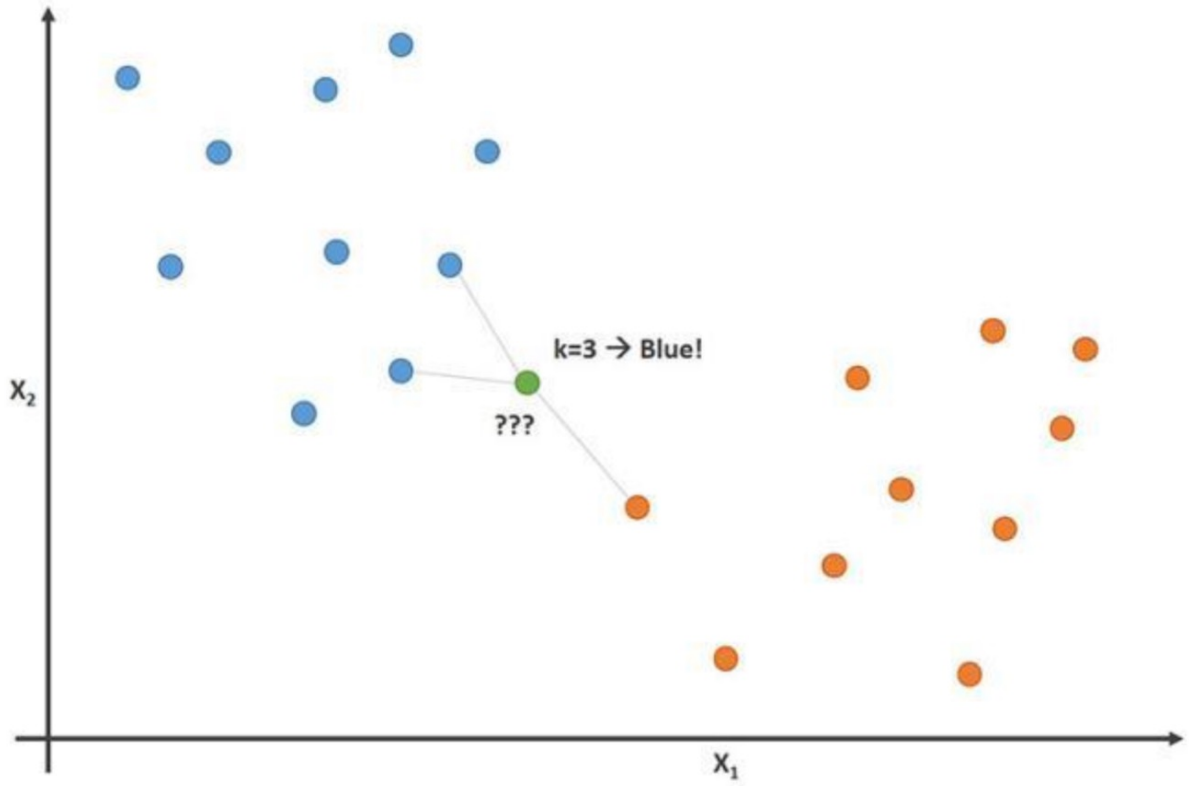
- 회귀를 사용하여 데이터가 어떤 범주에 속할 확률을 0에서1사이의 값으로 예측하고, 그 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류해주는 알고리즘
- 이진 분류 문제를 해결하는데 적합
- 분류를 확률로 해석하기 위해 시그모이드 함수를 이용한다
- 예측 확률이 0.5이상이면 1, 0.5미만이면 0으로 분류된다



분류 알고리즘

KNN(K-Nearest Neighbor)

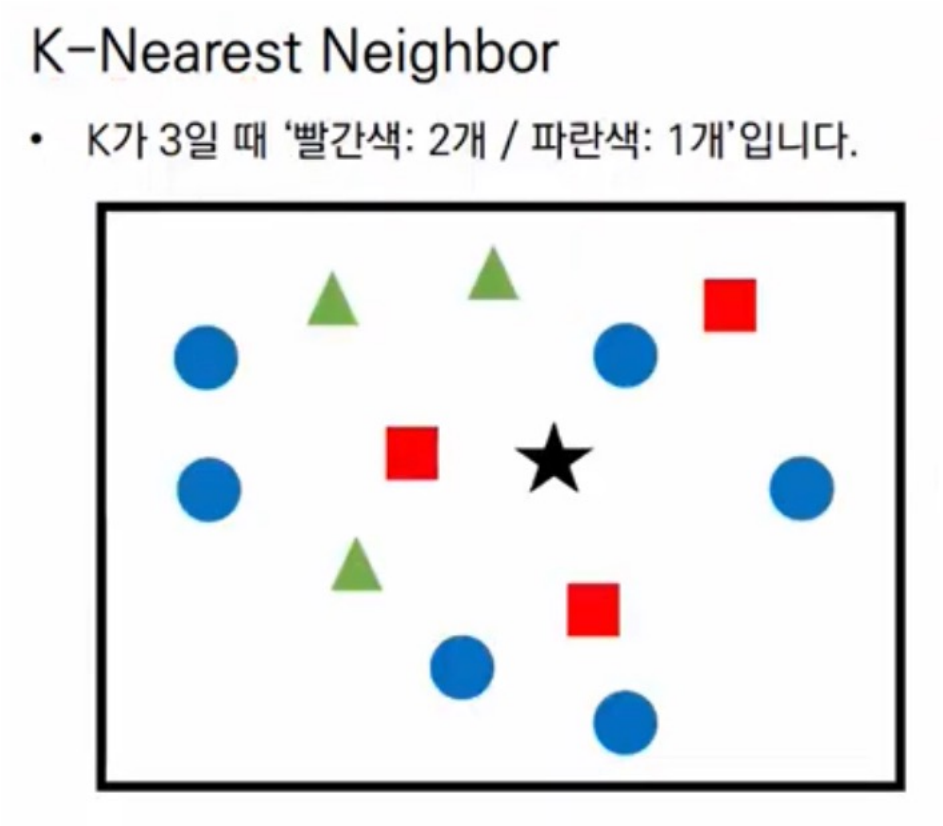
- 다양한 레이블의 데이터 중에서, 자신과 가까운 데이터를 찾아 자신의 레이블을 결정하는 방식
- 단순히 “주변에 가장 가까이 있는게 무엇인가?’ 가 아닌 “주변에 가장 가깝고, 많이 있는 것이 무엇인가?”라는 방식을 사용함
- 다양한 거리 계산 알고리즘에 대한 논의가 필요함(ex. 유클리드, 해밍..)



분류 알고리즘

KNN(K-Nearest Neighbor)

- 2차원 평면 상에 3개의 레이블이 놓여있다고 가정
- 별표가 들어왔을 때 어떤 레이블에 가까운지 판단
- 여기서는 빨간색 2개와 파란색 1개에 가깝기 때문에 빨간색으로 라벨링



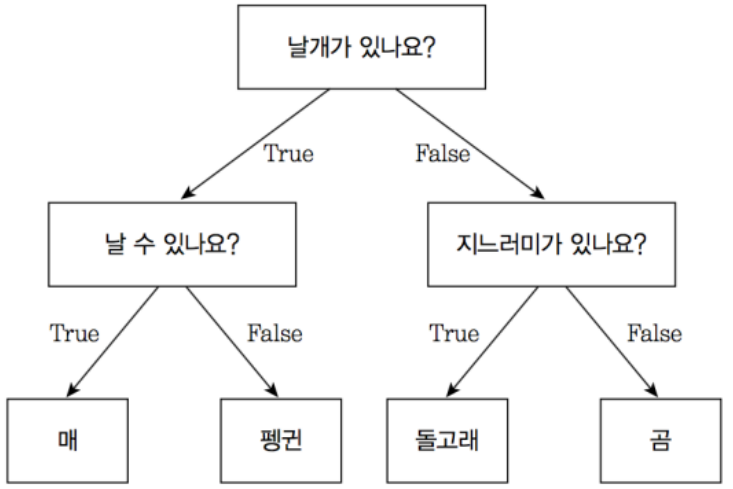
분류 알고리즘

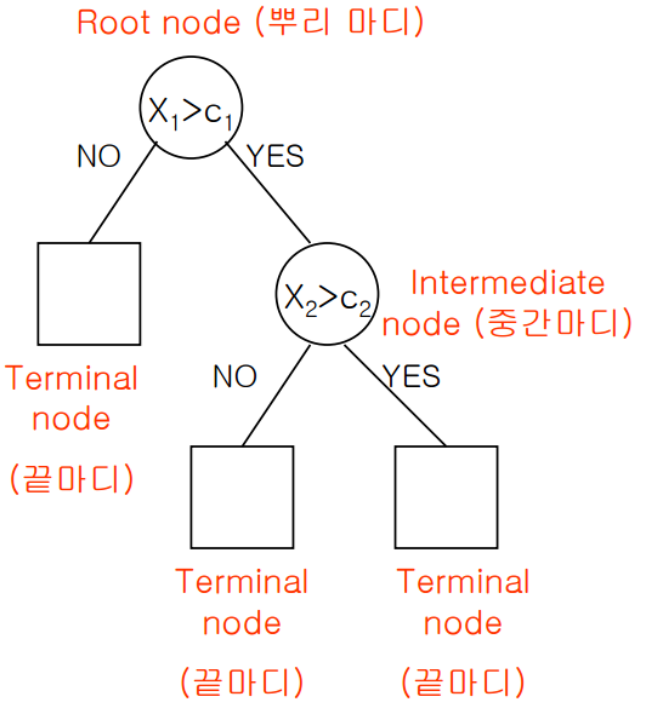
의사결정트리 (Decision Tree)

- 특정 기준에 따라 데이터를 구분하는 모델

- 상위에서 하위로 갈 수록 집단 내에는 동질성이, 집단 간에는 이질성이 커져야 함

- 훈련용 데이터에 따라 지나칠 정도로 복잡한 모델이 만들어 질 수 있음(과적합)





분류 알고리즘

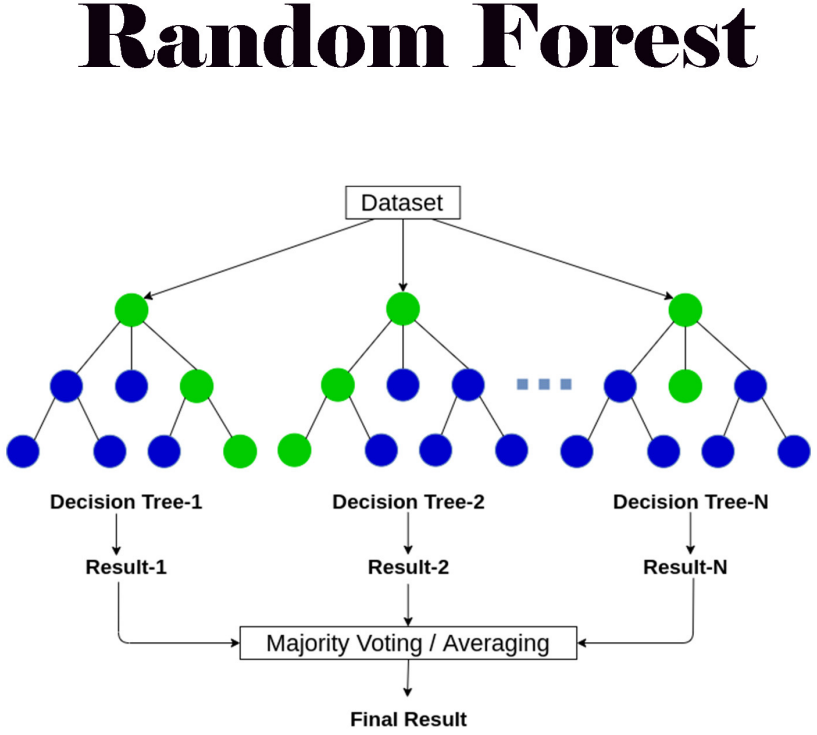
랜덤 포레스트 (Random Forest)

- 여러 개의 의사결정트리를 조합하여 결과를 도출하는 앙상블 기법 중 하나

- 배깅(Bagging) : 동일한 알고리즘을 여러 번 훈련시키되, 데이터를 무작위로 샘플링해서 다양성 확보

- 단일 트리가 가질 수 있는 과적합 문제를 해결하고, 결과의 안정성이 높아짐

- 여러 개의 결정트리를 사용하기 때문에 메모리 사용량이 많음



분류 평가 지표

평가 지표

		예측 클래스	
		Positive(1)	Negative(0)
실제 클래스	Positive(1)	**TP**	**FN**
	Negative(0)	**FP**	**TN**

- TP: 모델이 positive라고 예측, 실제 정답이 positive (정답)
- TN: 모델이 negative라고 예측, 실제 정답이 negative (정답)
- FP: 모델이 positive라고 예측, 실제 정답이 negative (오답)
- FN: 모델이 negative라고 예측, 실제 정답이 positive (오답)

분류 평가 지표

정확도(Accuracy)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- 모델이 전체 문제 중에서 정답을 맞춘 비율
- 데이터가 불균형할 때는 Accuracy만으로 제대로 분류가 어려워서, 정밀도와 재현율 사용

정밀도(Precision)

$$Precision = \frac{TP}{TP + FP}$$

- 모델이 positive라고 예측한 것들 중에서 실제로 정답이 positive인 비율
- 실제 정답이 negative인 데이터를 positive라고 잘못 예측하면 안 되는 경우에 중요한 지표
- Precision을 높이기 위해선 FP(모델이 positive라고 예측했는데 정답은 negative인 경우)를 낮추는 것이 중요

분류 평가 지표

재현율(Recall)

$$Recall = \frac{TP}{TP + FN}$$

- 실제로 정답이 positive인 것들 중에서 모델이 positive라고 예측한 비율
- 실제 정답이 positive인 데이터를 negative라고 잘못 예측하면 안 되는 경우에 중요한 지표
- Recall를 높이기 위해선 FN(모델이 negative라고 예측했는데 정답이 positive인 경우)을 낮추는 것이 중요

F1 Score

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Recall과 Precision의 조화평균
- Precision과 Recall이 한쪽으로 치우쳐지지 않고 모두 클 때 큰 값을 가짐

분류 평가 지표

오분류율 (Error Rate)

$$= \frac{FP + FN}{TP + TN + FP + FN}$$

- 모델이 전체 데이터에서 잘못 맞춘 비율

특이도 (Specificity)

$$Specificity = \frac{TN}{TN + FP}$$

- 실제 정답이 negative인 것들 중에서 모델이 negative라고 예측한 비율

과제