머신러닝 요약

3.1 지도학습과 비지도학습

3.1.1 지도학습

- 정답을 알려주면서 진행되는 학습.
- 학습시 데이터와 함께 레이블(정답)이 항상 제공돼야 함.

3.1.2 비지도학습

- 레이블(정답)이 없이 진행되는 학습.
- 학습할 때 레이블 없이 데이터만 필요.

3.2 분류와 회귀

3.2.1 분류

- 데이터가 입력됐을 때 지도학습을 통해 미리 학습된 레이블 중 하나 또는 여러 개의 레이블로 예측.

  Ex) 이진 분류, 다중 분류

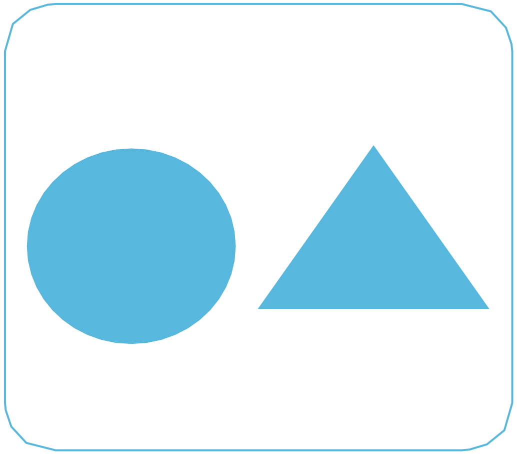


그림 3.1 다중 레이블 분류 테스트 데이터 예제

3.2.2 회귀

- 입력된 데이터에 대해 연속된 값으로 예측.

  Ex) 날씨 예측

3.5 머신러닝 모델의 성능 평가

3.5.1 TP(true positive) – 맞는 것을 올바르게 예측한 것

- 데이터를 입력했을 때 데이터의 실제값을 올바르게 예측한 케이스

표 3.5 혼동행렬에서 TP 찾기

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

3.5 머신러닝 모델의 성능 평가

3.5.2 TN(true negative) – 틀린 것을 올바르게 예측한 것

- 데이터를 입력했을 때 틀린 것을 올바르게 예측한 것

표 3.6 혼동행렬에서 A 클래스의 TN 찾기

		예측값			
		A	B	C	D
실제값	A	9	1	0	0
	B	1	15	3	1
	C	5	0	24	1
	D	0	4	1	15

4.1 머신러닝 알고리즘 실습 개요

4.1.1 알고리즘 선정 이유

- 최고의 알고리즘은 존재하지 않음.
- 데이터의 특징과 상황에 따라 가장 적합한 알고리즘을 선택하는 것이 중요.

표 4.1.1 머신러닝 알고리즘의 장단점 비교

알고리즘	장점	단점
k-최근접 이웃	• 구현이 쉽다. • 알고리즘을 이해하기 쉽다. • 하이퍼파라미터[1]가 적다.	• 예측 속도가 느리다. • 메모리를 많이 쓴다. • 노이즈 데이터[2]에 예민하다.
서포트 벡터 머신	• 상대적으로 적은 데이터로도 높은 정확도를 낸다. • 예측 속도가 빠르다. • 고차원 데이터를 처리하기가 쉽다.	• 결정경계선이 많이 겹칠 때 정확도가 낮아진다. • 수학적 이해 없이는 모델의 분류 결과를 이해하기 어렵다. • 커널 트릭 오사용 시 과대적합되기 쉽다.

4.1 머신러닝 알고리즘 실습 개요

4.1.1 알고리즘 선정 이유

알고리즘	장점	단점
의사결정트리	• 모델의 추론 과정을 시각화하기 쉽다. • 데이터에서 중요한 특성이 무엇인지 쉽게 알아낼 수 있다. • 학습 및 예측 속도가 빠르다.	• 과대적합되기 쉽다. • 조정해야 할 하이퍼파라미터가 많다.
랜덤포레스트	• 앙상블 효과로 의사결정 트리의 과대적합 단점을 보완한다.	• 조정해야 할 하이퍼파라미터가 많다.
나이브베이즈	• 고차원 데이터를 처리하기가 쉽다. • 구현하기 쉽다. • 학습 및 추론 시간이 빠르다.	• 모든 변수가 독립변수라는 가설하에 작동함으로써 데이터가 가설과 다를 경우 정확도가 낮아진다.
선형회귀	• 수집된 데이터를 통해 새롭게 관측된 데이터의 예측값(수치값)을 구할 수 있다.	• 데이터 특징들이 선형 관계에 있다는 가설하에 작동함으로써 데이터 특징이 가설과 다를 경우 정확도가 낮아진다.

4.1 머신러닝 알고리즘 실습 개요

4.1.1 알고리즘 선정 이유

알고리즘	장점	단점
로지스틱회귀	• 데이터를 분류할 때 확률을 제공한다.	• 데이터 특징이 많을 경우 학습이 어려워과소적합되기 쉽다.
K 평균	• 데이터 크기에 상관 없이 군집화에 사용할 수 있다. • 구현하기 쉽다.	• 군집화 결과에 대한 확률을 제공하지 않는다. • 데이터의 분포가 균일하지 않을 경우 정확도가 떨어진다.
주성분 분석	• 고차원 데이터를 저차원 데이터로 축소할 때 사용된다. • 구현이 쉽다.	• 차원 축소 시 정보의 손실이 있다.

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

4.2.1 [이론] k-최근접 이웃 알고리즘(kNN)

- 기존의 데이터에서 현재 데이터로부터 가까운 k개의 데이터를 찾아 k개의 레이블 중 가장 많이 분류된값으로 현재의 데이터를 분류
- Ex) 이곳이 강남일까요, 강북일까요? (k=5일 경우)
- 머신러닝에서 사용되는 공간이란 개념은 사실벡터 공간을 의미
    - 현실 공간: 평면 이동 및 수직 이동이 가능한 3차원 공간
    - 벡터 공간: 벡터 연산이 가능한 N차원 공간

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

4.2.1 [이론] k-최근접 이웃 알고리즘(kNN)

- Ex) 농구선수의 포지션이센터인지 슈팅가드인지 예측

표 4.2.1 농구선수 데이터

선수 이름	3점슛 성공 횟수	블로킹 성공 횟수	선수 포지션
이정하	1	6	센터
유옥중	2	5	센터
오일두	2	7	센터
김제영	3	7	센터
황영희	4	1	슈팅가드
김윤석	4	4	센터
오현화	4	6	센터
박예은	5	2	슈팅가드
최예원	6	1	슈팅가드

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

4.2.1 [이론] k-최근접 이웃 알고리즘(kNN)

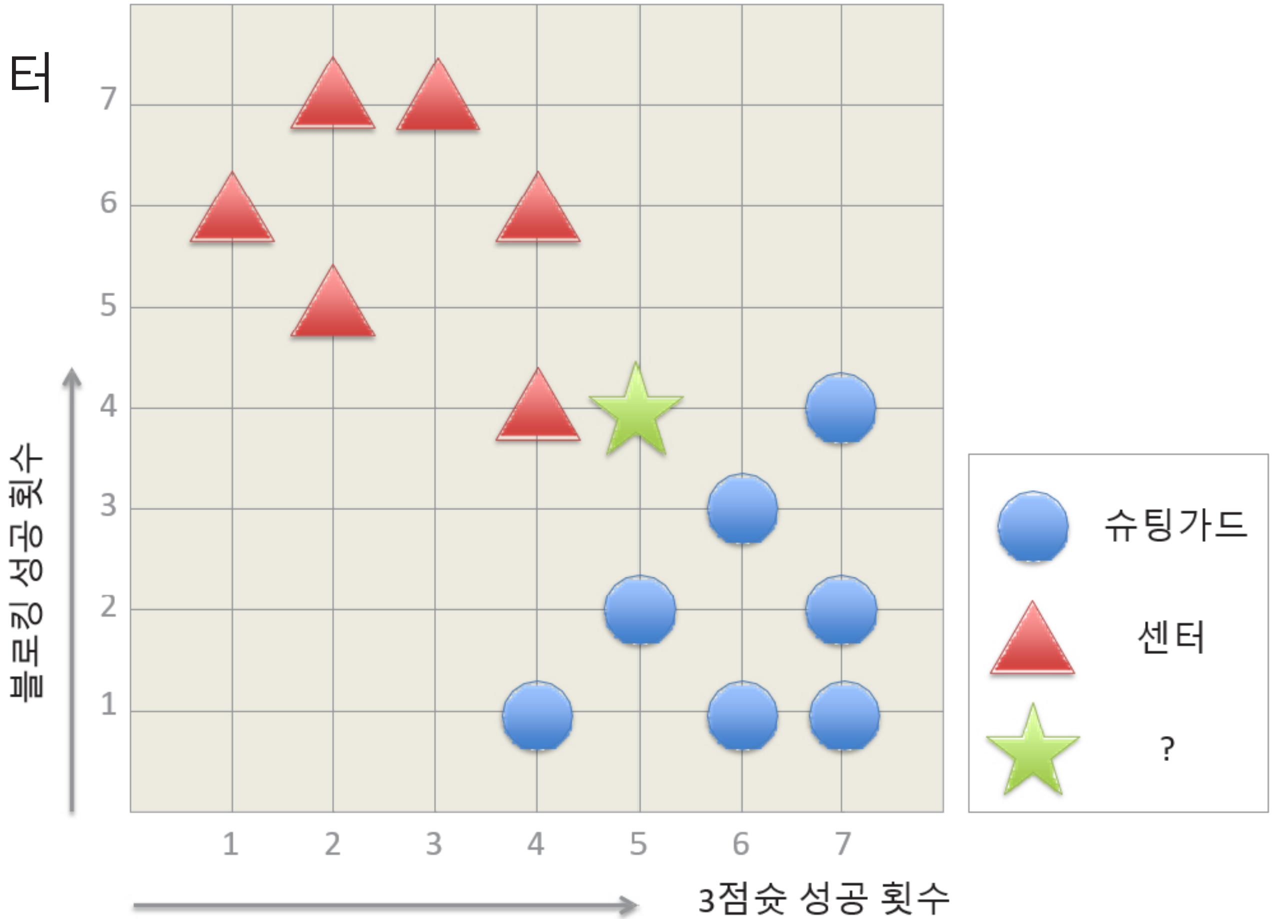
- Ex) 농구선수의 포지션이센터인지 슈팅가드인지 예측

선수 이름	3점슛 성공 횟수	블로킹 성공 횟수	선수 포지션
최경자	6	3	슈팅가드
최선옥	7	1	슈팅가드
오광희	7	2	슈팅가드
오광민	7	4	슈팅가드
홍길동	5	4	?

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

- 3점슛 성공 횟수를 벡터 공간의 x축으로, 블로킹 성공 횟수를 y축으로 2차원 벡터 공간에 시각화

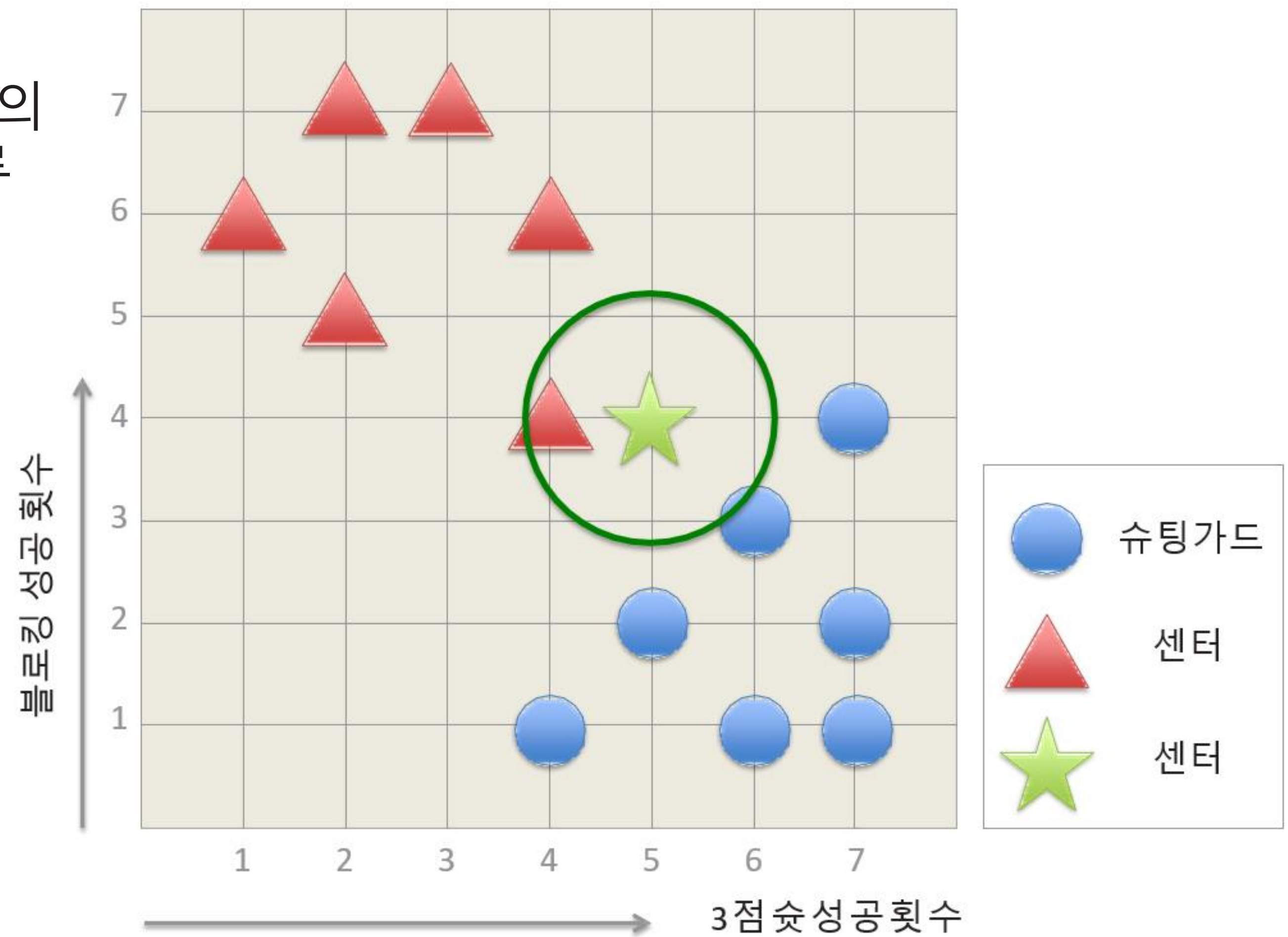
표 4.2.1 농구선수 데이터



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

- 제일 먼저 k가 1일 경우, 가장 가까운 농구선수 데이터 1개를 찾아 포지션을 예측

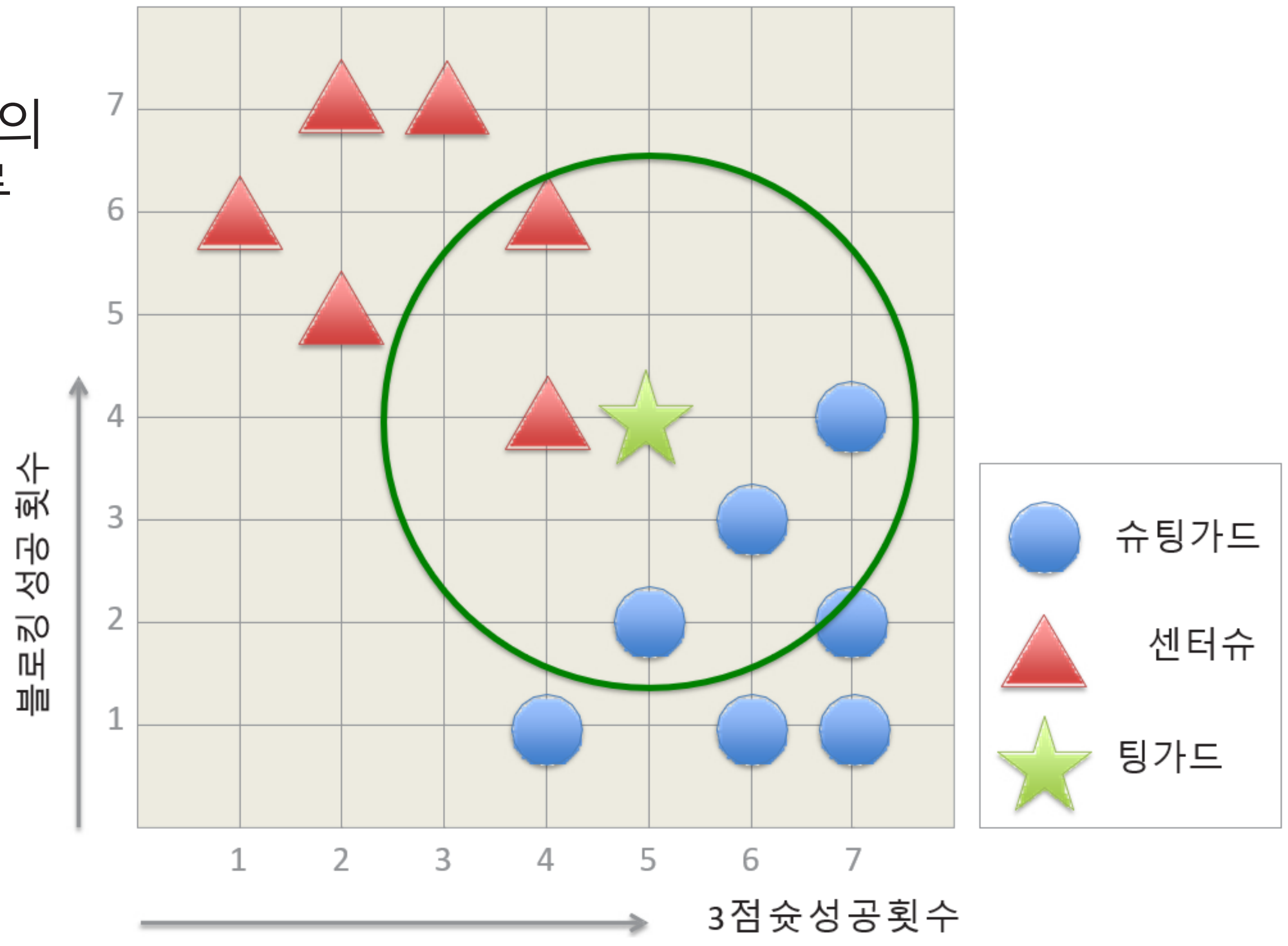
표 4.2.2 k = 1인 경우의 kNN 알고리즘의 분류



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

- k가 5일 경우, 슈팅가드로 분류

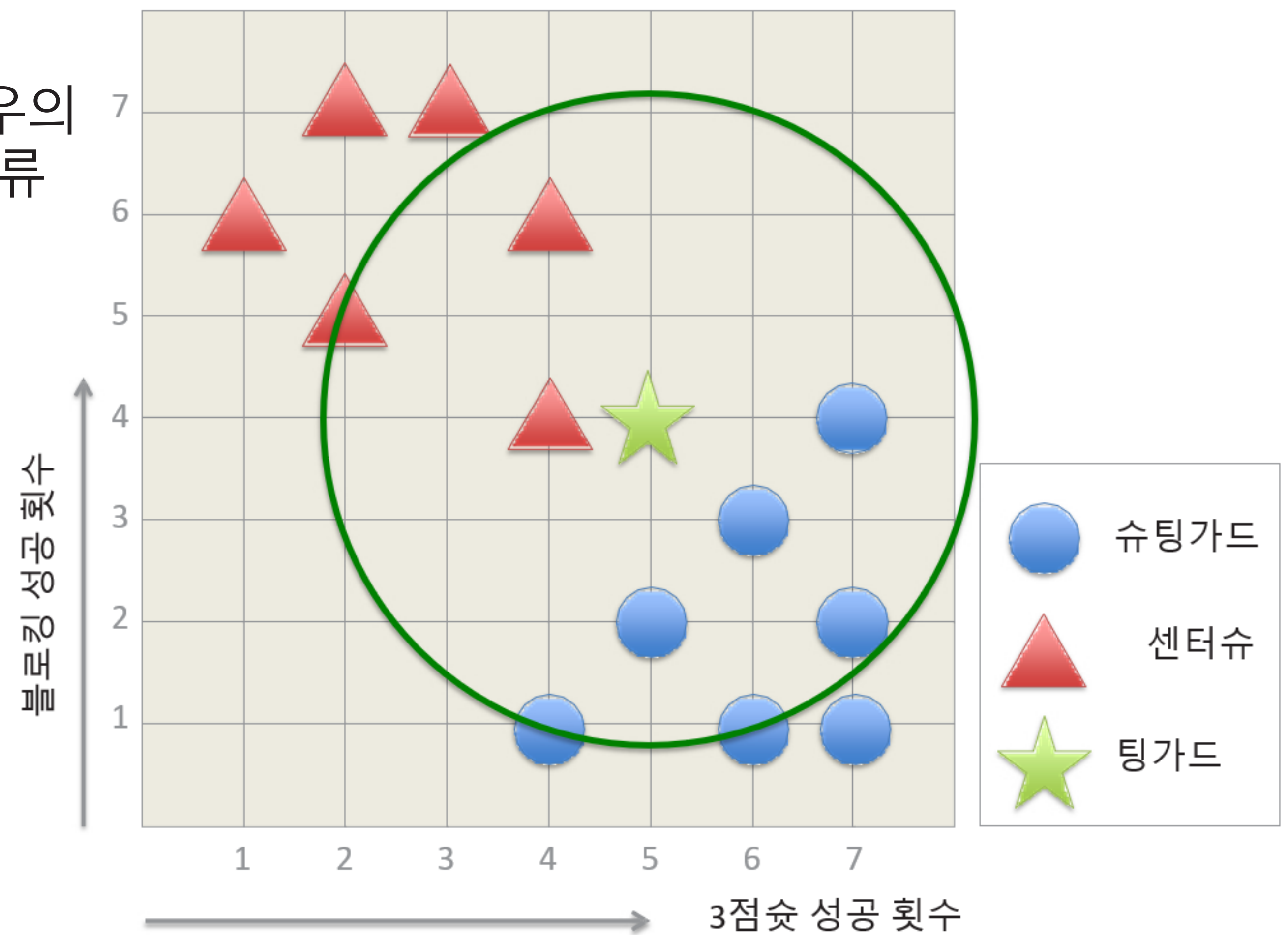
표 4.2.3 k = 5인 경우의 kNN 알고리즘의 분류



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

- k가 9일 경우, 슈팅가드로 분류

표 4.2.4 k = 9인 경우의 kNN 알고리즘의 분류



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

- k(탐색할 이웃의 개수)에 따라 데이터를 다르게 예측
- 보통 k는 1로 설정하지 않음.
- 보통 k는 홀수로 설정.

다중분류

- 분류는 보통 이진 분류(binary classification)와 다중 분류(multiclass classification)로 나뉨.
  - 이진 분류는 두 가지 중 하나로 분류.
  - 다중 분류는 여러 개의 가능한 레이블 중 하나로 분류.

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

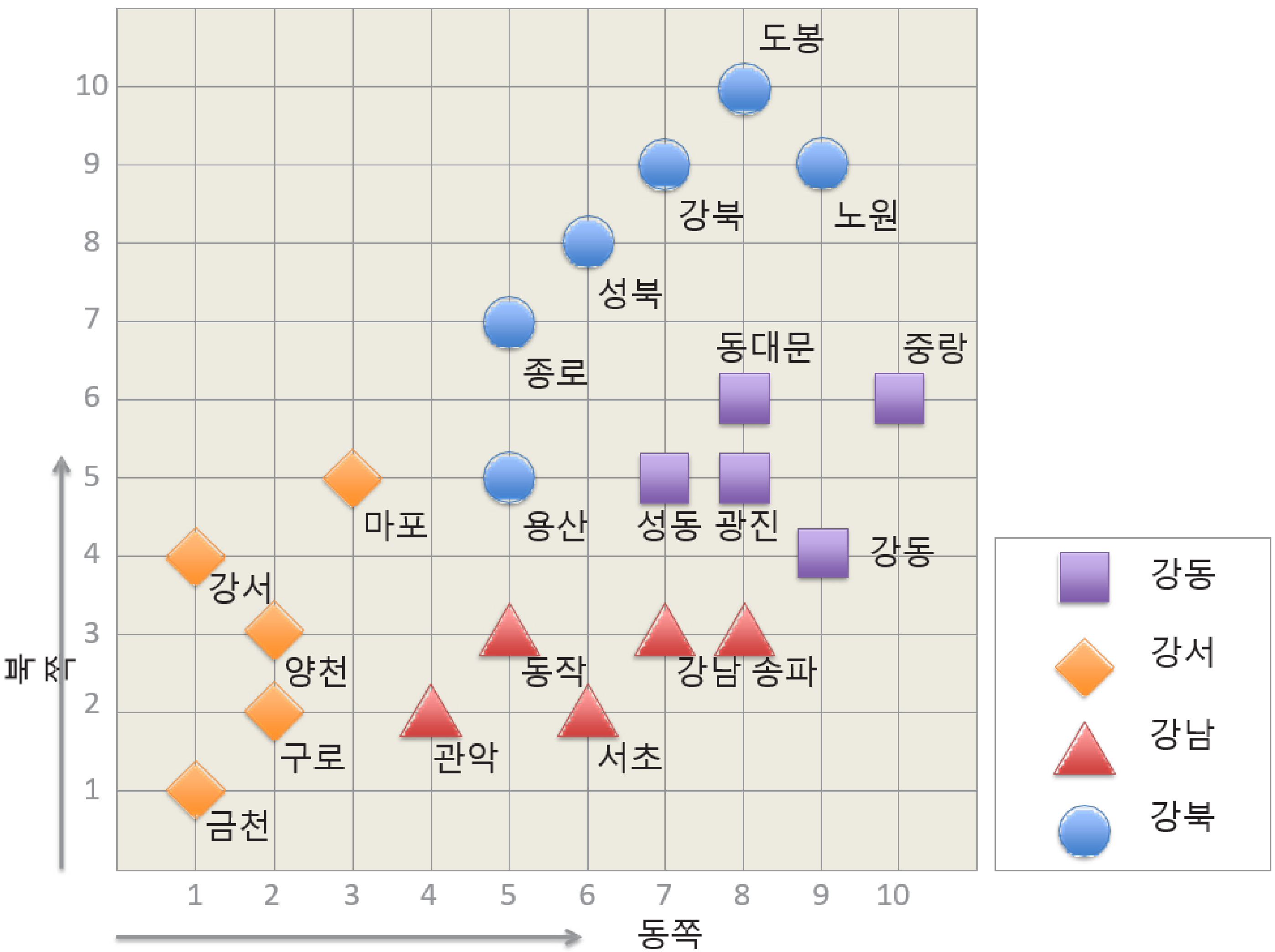
표 4.2.2 이진 분류와 다중 분류 예제

이진 분류	다중분류
악성 코드 분류(일반 파일 또는 악성 코드)	임의의 손글씨 숫자가 입력됐을 때 1에서9 중 가장가까운 숫자로 분류하는 모델
위조 지폐 분류(일반 지폐 또는 위조 지폐)	서울의 도시가 입력됐을 때 강동, 강서, 강남, 강북 중한 곳으로 도시를 분류하는 모델
문장에서 사람의 감정을 분류할 때 행복 또는 슬픔으로만 분류하는 모델	문장에서 사람의 감정을 분류할 때 행복, 슬픔, 화남으로 분류하는 모델

- kNN 알고리즘은 다중 분류에도 탁월한 성능을 보임

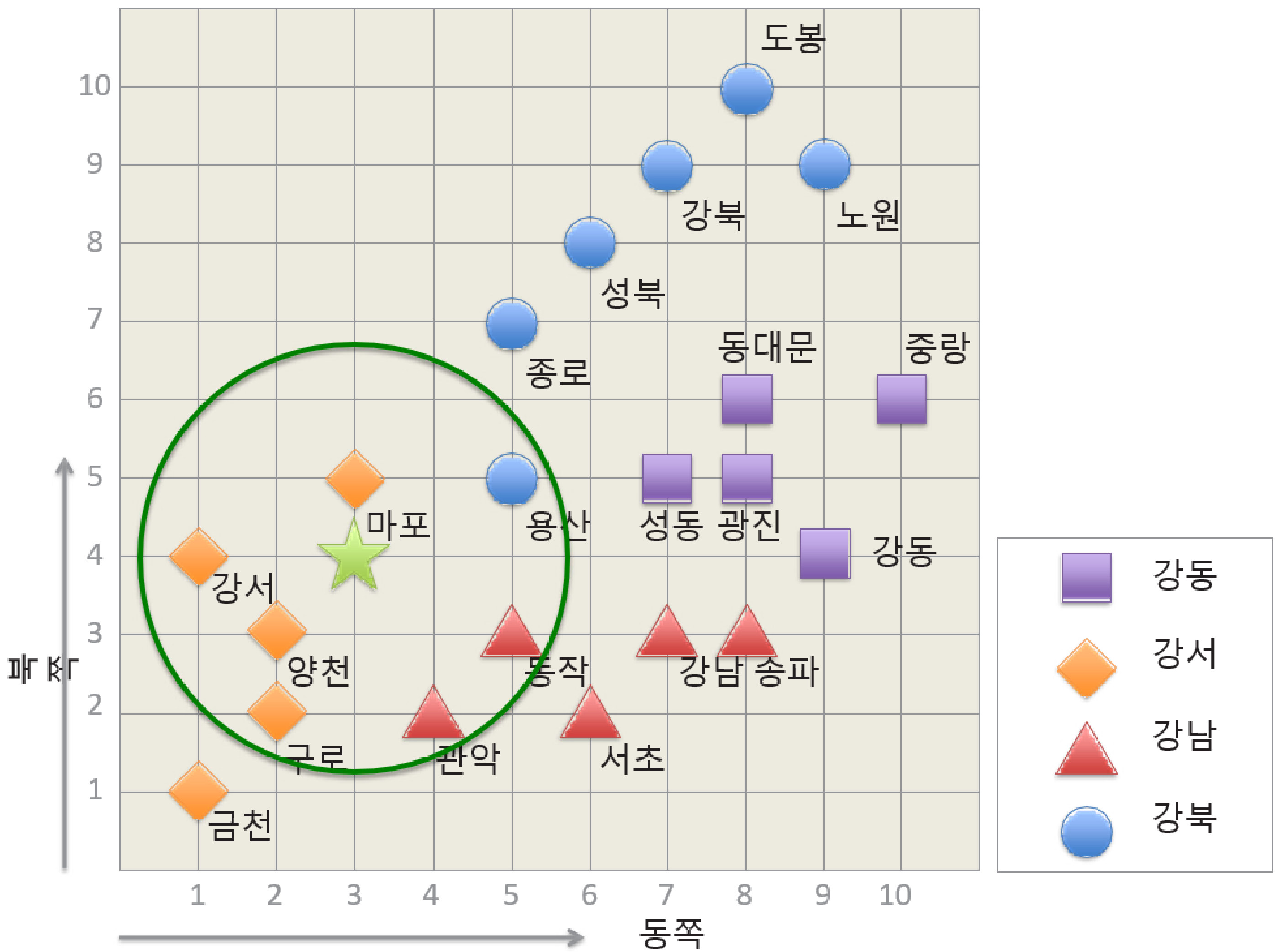
4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

그림 4.2.5 서울시 행정구역에 대한 다중 분류 예



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

그림 4.2.6 입력 (3,4)가 들어왔을 때 kNN(k=7) 분류를 시각화한 결과



4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

kNN 알고리즘의 수학적 이해

- 농구선수 분류 예제에서 3점슛 성공 횟수와 블로킹이라는 두 가지 속성을 사용한 공간 속에서 최근접 이웃을 찾음.
- 두 가지 속성은 2차원 공간에 나타낼 수 있음.
- 2차원 공간에서의 거리는 피타고라스의 정리를 적용

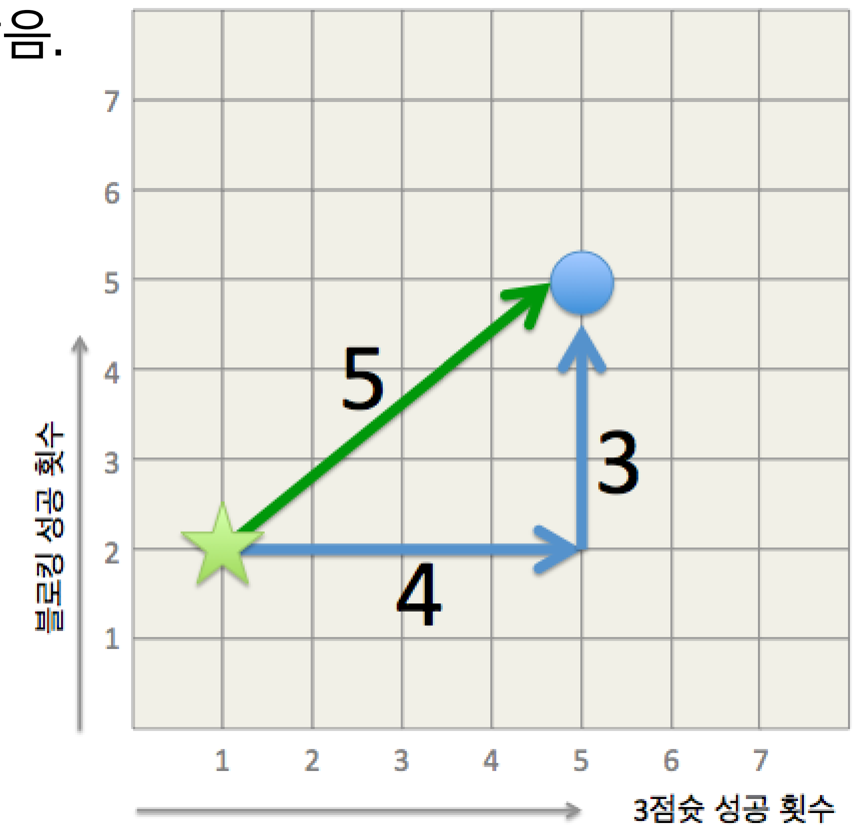
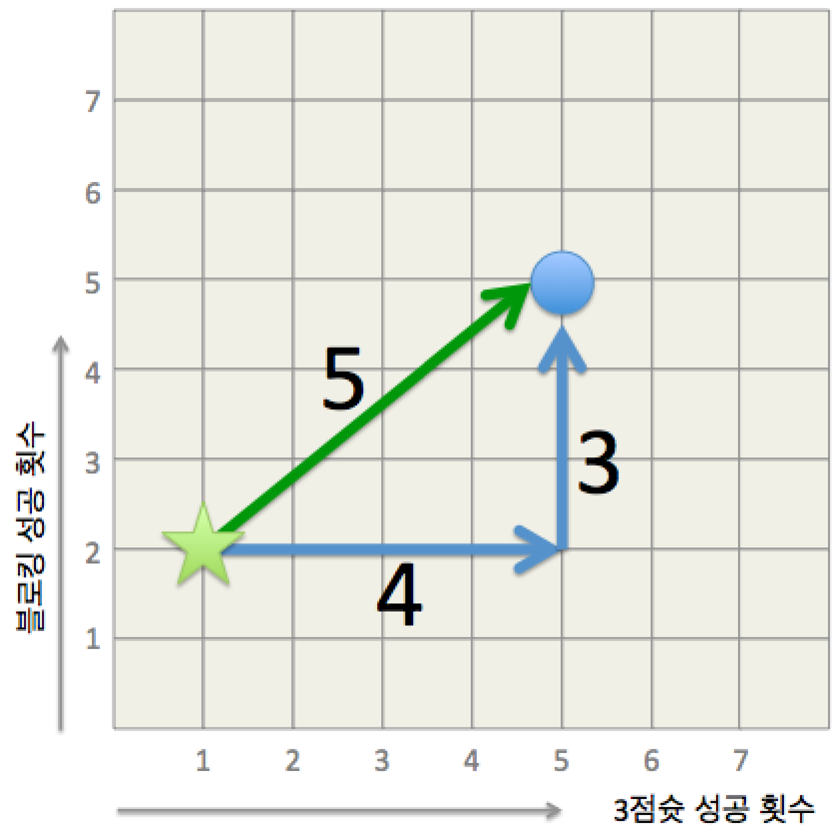


그림 4.2.7 벡터 공간의 거리 계산

4.2 k-최근접 이웃(k-Nearest Neighbor, kNN)

그림 4.2.7 벡터 공간의 거리 계산



- 별과 동그라미 사이의 거리

별 (1,2), 동그라미 (5,5)

$$\sqrt{(1-5)^2 + (2-5)^2} = 5$$

- 속성이 3개인 경우,

$$\sqrt{(x_1)^2 + (x_2)^2 + (x_3)^2} = y$$

- 3차원을 넘는 N차원(N개의 속성) 공간에서도 거리를 계산하는 방법은 동일
- 이 5차원 벡터공간에서의 두 데이터의 거리는

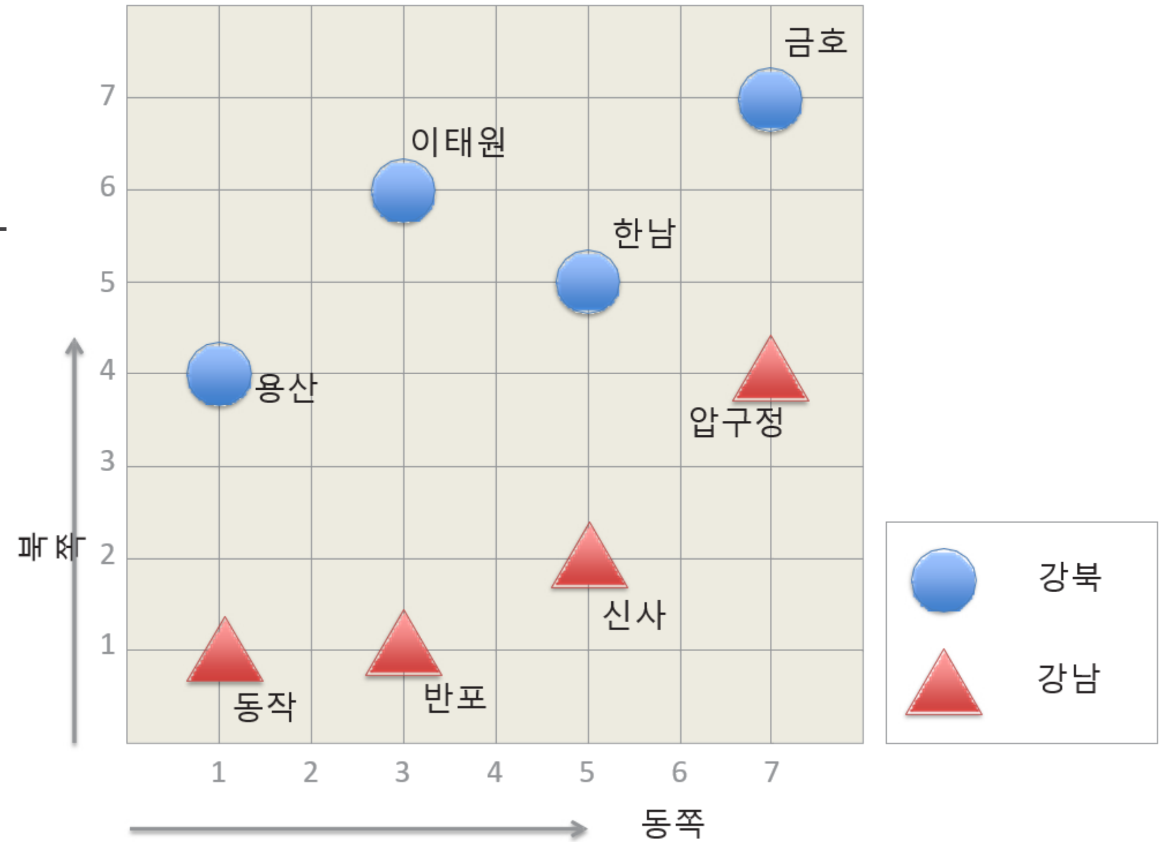
$$\sqrt{(x_1)^2 + (x_2)^2 + (x_3)^2 + (x_4)^2 + (x_5)^2} = y$$

4.3 서포트 벡터머신(SVM)

4.3.1 [이론] 서포트 벡터 머신

- 한강은 도시가 강북인지, 강남인지를 구분하는 결정 경계선(decision boundary)이고, 한강 위는 강북,아래는 강남이 됨.

그림 4.3.1 강남과 강북 도시를 표시한 데이터 차트



4.3 서포트 벡터머신(SVM)

- 한강의 위치 후보

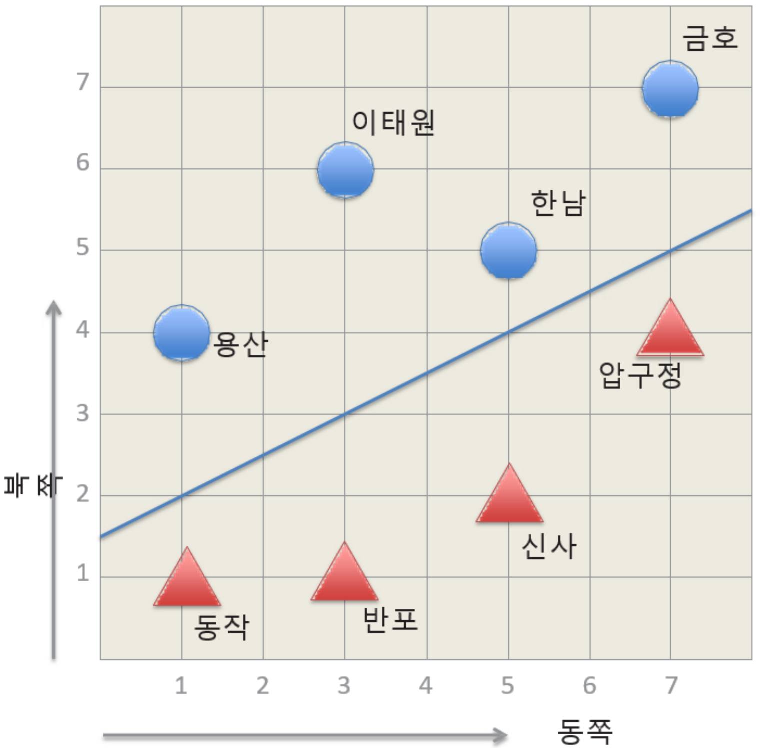


그림 4.3.2 한강의 위치 후보 1

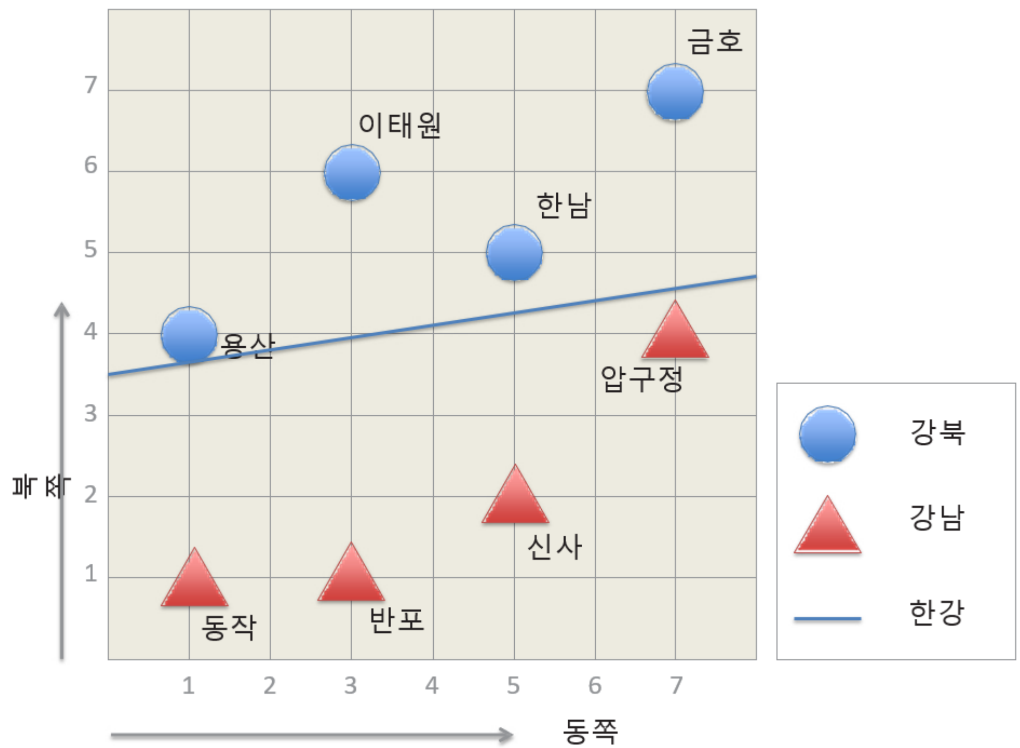


그림 4.3.3 한강의 위치 후보 2

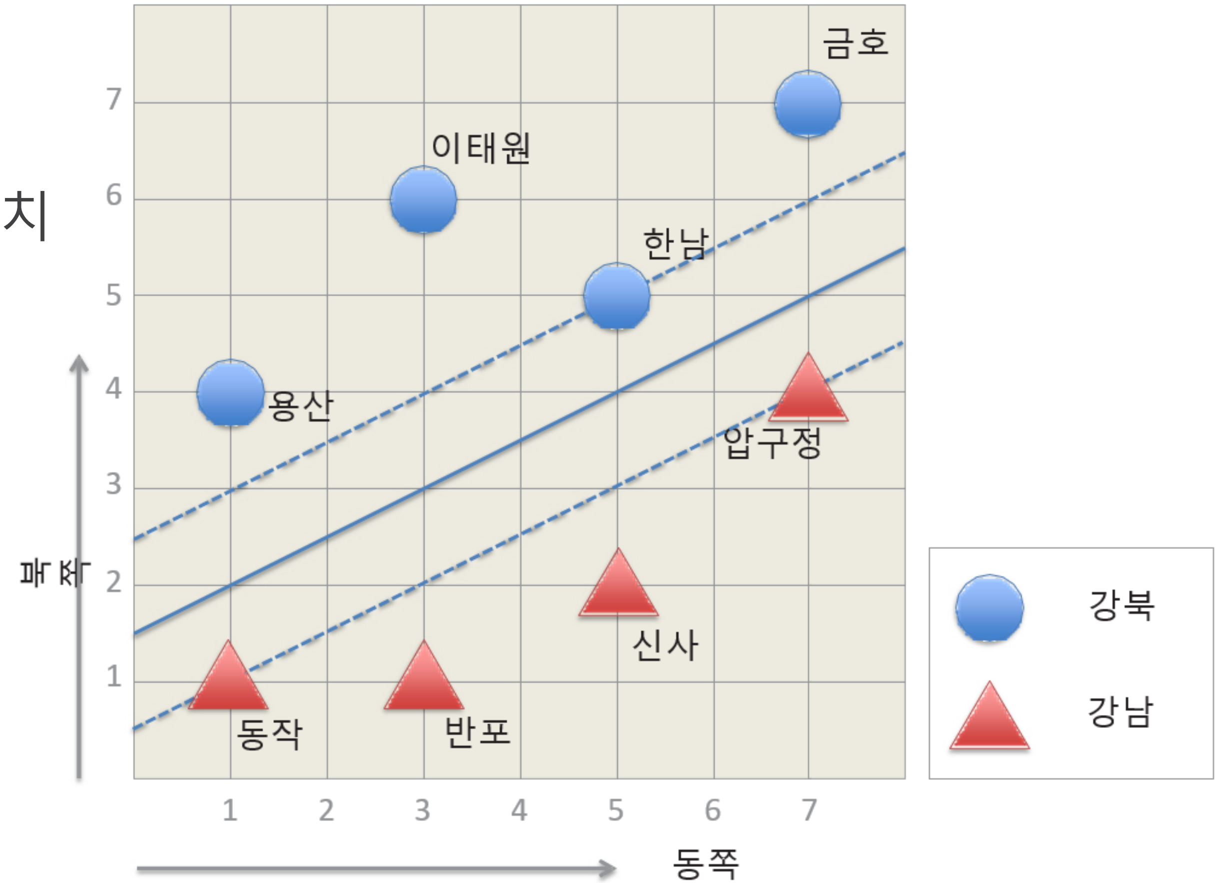
- 후보 1이 후보 2보다 강남과 강북을 구분하는 결정 경계선으로 더 나아 보임.

4.3 서포트 벡터머신(SVM)

서포트 벡터

- 벡터는 바로 2차원 공간 상에 나타난 데이터 포인트를 의미.
- 서포트 벡터는 결정 경계선과 가장 가까이 맞닿은 데이터 포인트를 의미.

그림 4.3.4 서포트 벡터의 위치



4.3 서포트 벡터머신(SVM)

- 감마의 크기에 따른 결정 경계 곡률의 차이

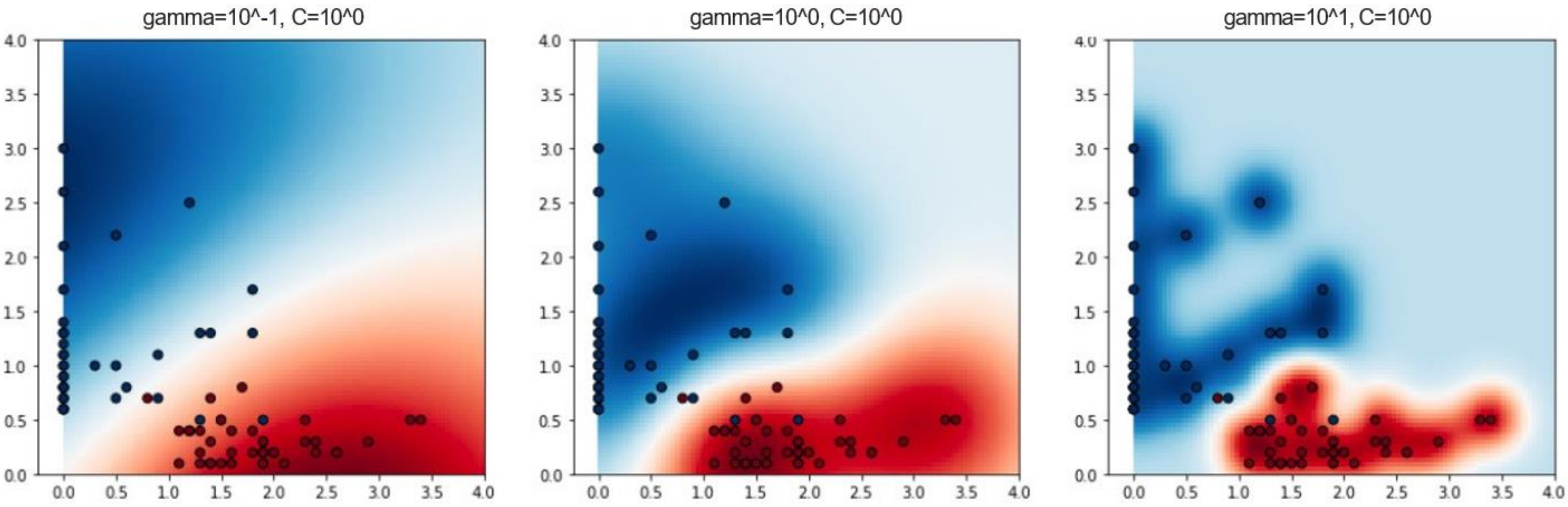


그림 4.3.14 감마의 크기에 따른 결정 경계의 곡률 차이

4.4 의사결정트리

4.4.1 [이론] 의사결정 트리

- 데이터 분류 및 회귀에 사용되는 지도학습 알고리즘
- 장점은 다른 알고리즘에 비해 결괏값이 왜, 어떻게 나왔는지 이해하기가 상당히 쉽다는 것.
- 높은 정확도 역시 상당히 큰 장점.
- 하지만과대적합되기 쉬운 알고리즘이라는 단점도 있음.
- 스무고개라는 놀이와 유사.

4.4 의사결정트리

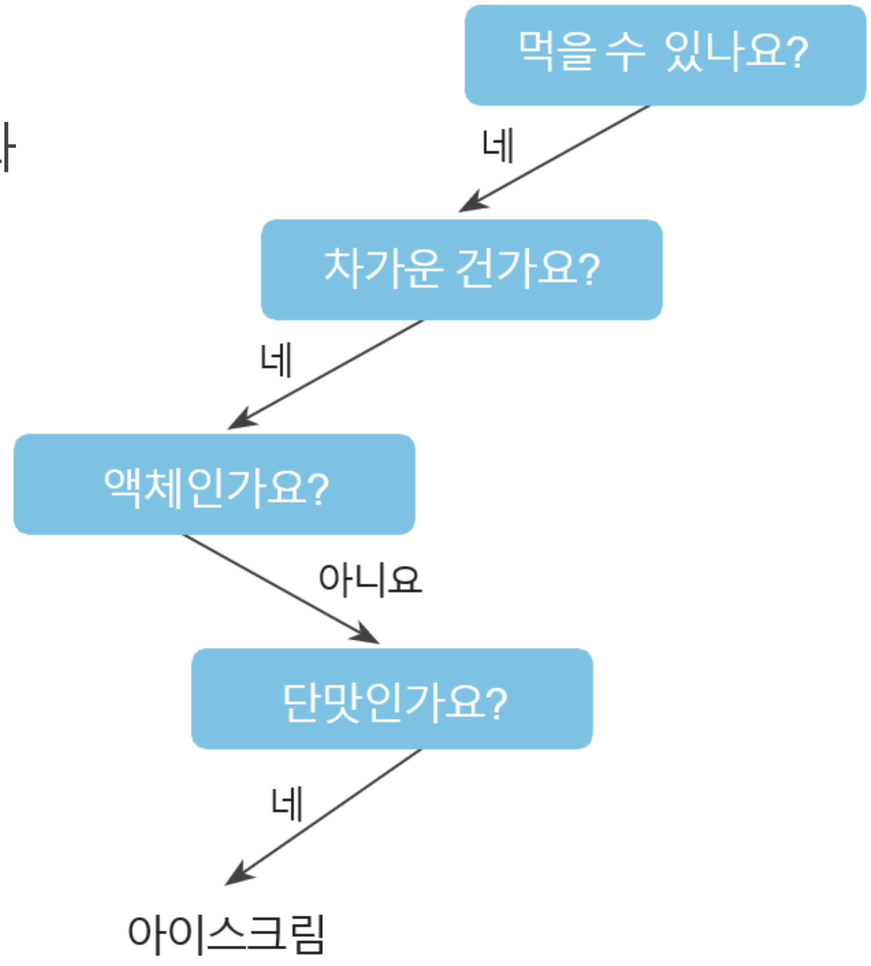


그림 4.4.1 스무고개 놀이의 시각화

4.4 의사결정트리

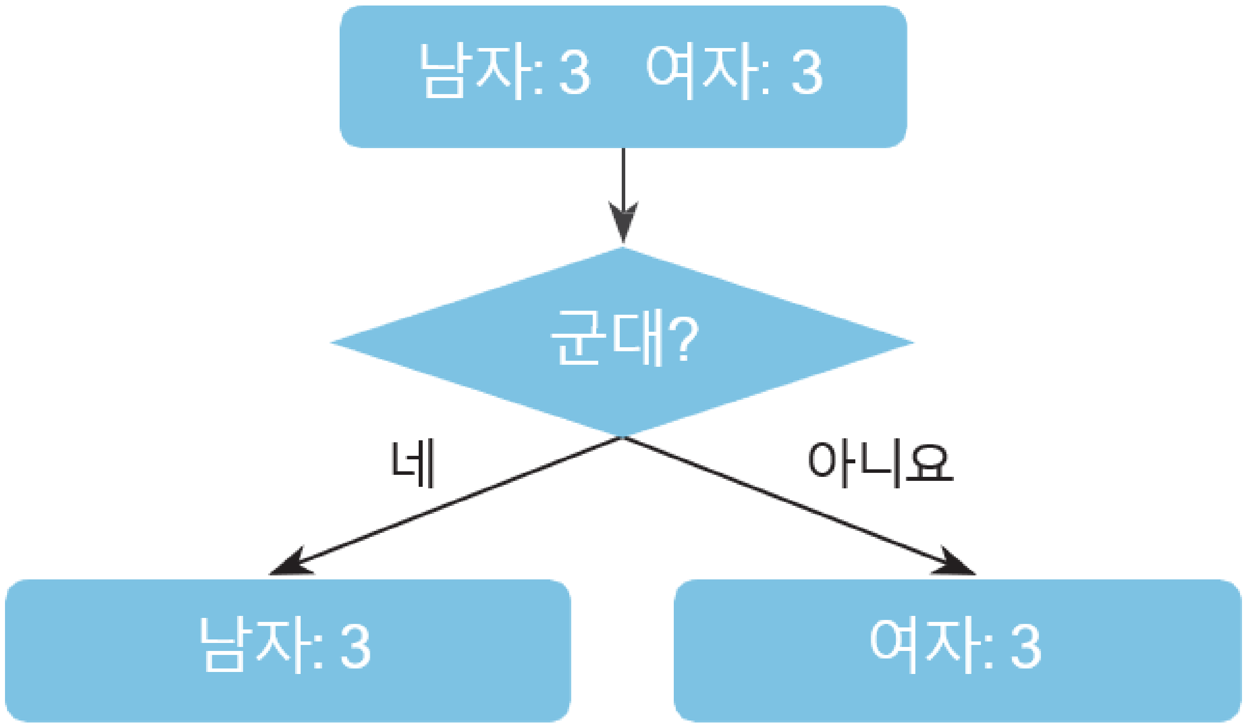
- 다음과 같은 데이터가 있을 때 어떤 질문을 먼저 하는 것이 그 사람이 남자인지 여자인지 구분하기에 더 효율적인가?

표 4.4.1 남자와 여자를 구분하기 위한 데이터

이름	군대를 다녀왔는가?	긴 생머리인가?	성별
김덕수	네	아니요	남자
이쁜이	아니요	아니요	여자
박장군	네	아니요	남자
최빛나	아니요	네	여자
최강민	네	아니요	남자
지화자	아니요	아니요	여자

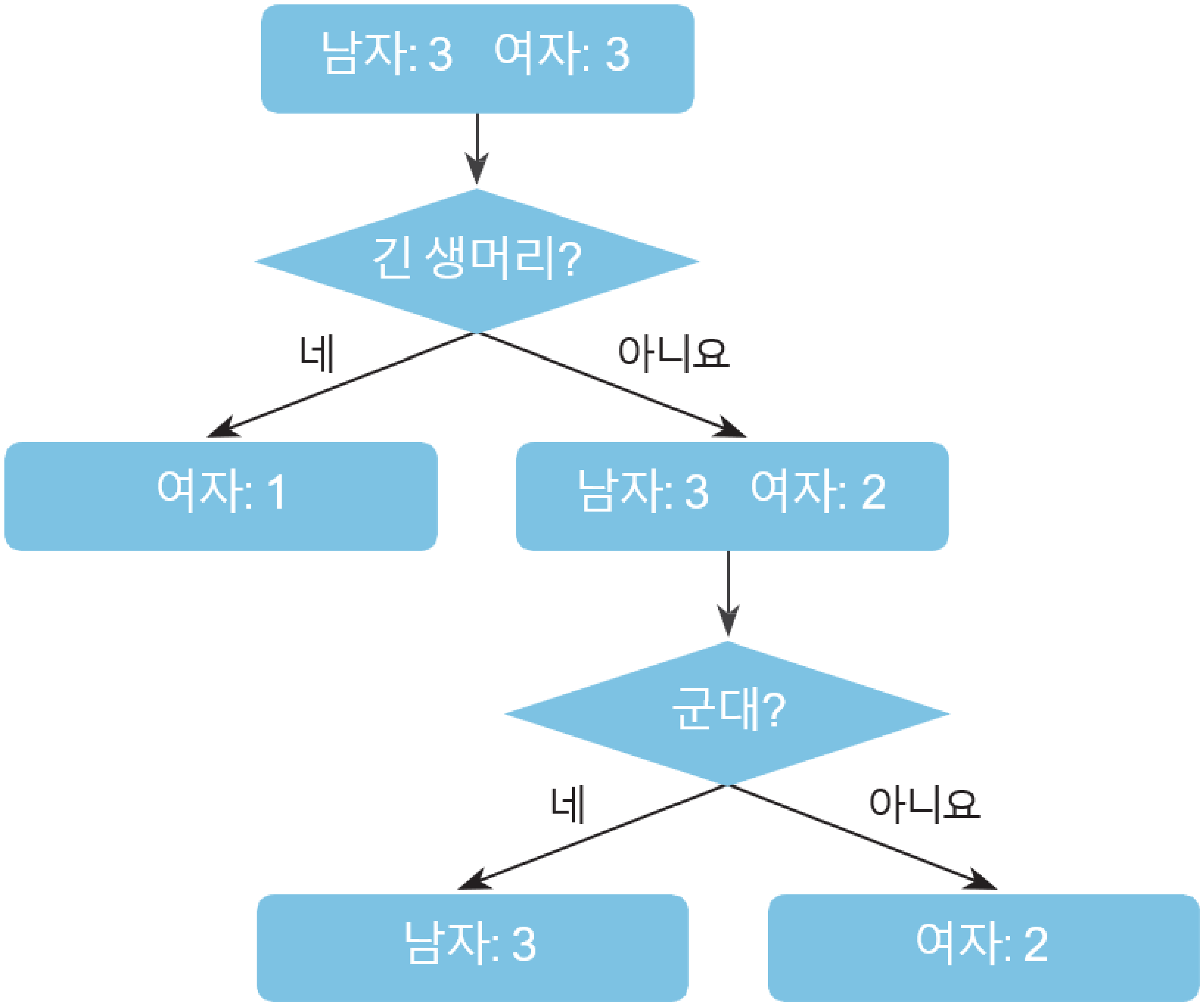
4.4 의사결정트리

그림 4.4.2 군대를 다녀왔는지를 먼저 물어보는 경우



4.4 의사결정트리

그림 4.4.3 긴 생머리인지를 먼저 물어보는 경우



4.4 의사결정트리

- 예제와 같은 데이터가 있을 때 군대를 다녀왔는지를 먼저 물어보는 것이 더 효율적이라고 할 수 있음.
- 머신러닝에서 의미 있는 질문이란 데이터의 특징 중 의미 있는 특징.
- , 의사결정 트리의 핵심은 바로 영향력이 큰 특징을 상위 노드로, 영향력이 작은 특징은 하위 노드로 선택하는 것.
- 의사결정 트리는 특징별 영향력의 크고 작음을 비교하기 위해 다음과 같은 두 가지 방법 중 하나를 사용함.

4.5 나이브베이즈

4.5.1 [이론] 나이브 베이즈

나이브 베이즈 알고리즘의 이해

- 데이터를 나이브(단순)하게 독립적인 사건으로 가정하고, 독립 사건들을 베이즈 이론에 대입시켜 가장 높은 확률의 레이블로 분류를 실행하는 알고리즘
- 베이즈 이론 공식

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

  - $P(A|B)$: 어떤 사건 B가 일어났을 때 사건 A가 일어날 확률
  - $P(B|A)$: 어떤 사건 A가 일어났을 때 사건 B가 일어날 확률
  - $P(A)$: 어떤 사건A가 일어날 확률

4.5 나이브베이즈

- 조건부 확률 공식

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

- 조건부 확률 예시

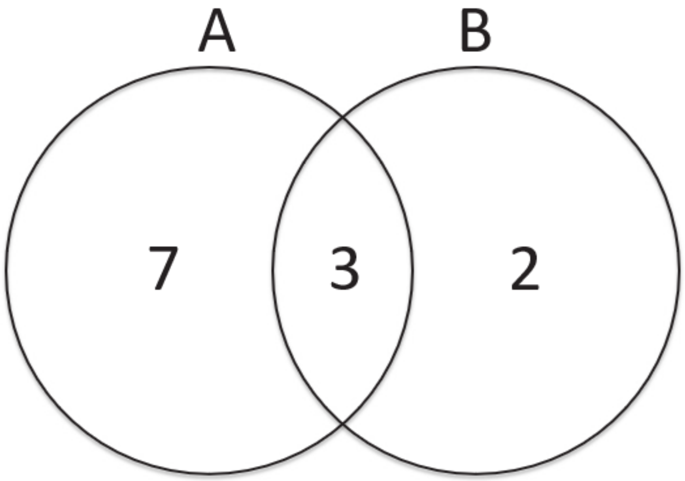


그림 4.5.1 조건부 확률을 이해하기 위한 벤 다이어그램

4.6 앙상블

- 앙상블 기법이란 여러 개의 분류 모델을 조합해서 더 나은성능을 내는 방법.

4.6.1 [이론] 배깅

- 배깅은 과대적합이 쉬운 모델에 상당히 적합한 앙상블
- 배깅은 한 가지 분류 모델을 여러 개 만들어서 서로 다른 학습 데이터로 학습시킨 후(부트스트랩), 동일한 테스트 데이터에 대한 서로 다른 예측값들을 투표를 통해(어그리게이팅) 가장 높은 예측값으로 최종 결론을 내리는 앙상블 기법
- 배깅의 어원은 부트스트랩(bootstrap)과 어그리게이팅(aggregating)에서 옴.

4.7 군집화

- 군집화(clustering)는 비지도학습
- 데이터의 특징만으로 비슷한 데이터들끼리 모아 군집된 클래스로 분류

4.7.1 [이론] k 평균 알고리즘

- k 평균 알고리즘32은 간단하면서도 강력한 군집화 알고리즘
- 다음과 같은 순서로 진행

1. 데이터 준비
2. 몇 개의 클래스로 분류할 것인지 설정
3. 클러스터의 최초 중심 설정
4. 데이터를 가장 가까운 클러스터로 지정
5. 클러스터 중심을 클러스터에 속한 데이터들의 가운데 위치로 변경
6. 클러스터 중심이 바뀌지 않을 때까지 4번부터 5번 과정을 반복적으로 수행

4.7 군집화

데이터 준비

- k 평균 알고리즘은 데이터 간의 거리를 사용해 가까운 거리에 있는 데이터끼리 하나의 클래스로 묶는 알고리즘
- 한 교실에 있는 학생들의 키와 몸무게 값으로 세 개의 군집으로 분류하는 예제를

표 4.7.1 학생들의 키와 몸무게 데이터

학생	키	몸무게
1	185	60
2	180	60
3	185	70
4	165	63
5	155	68
6	170	75
7	175	80

4.7 군집화

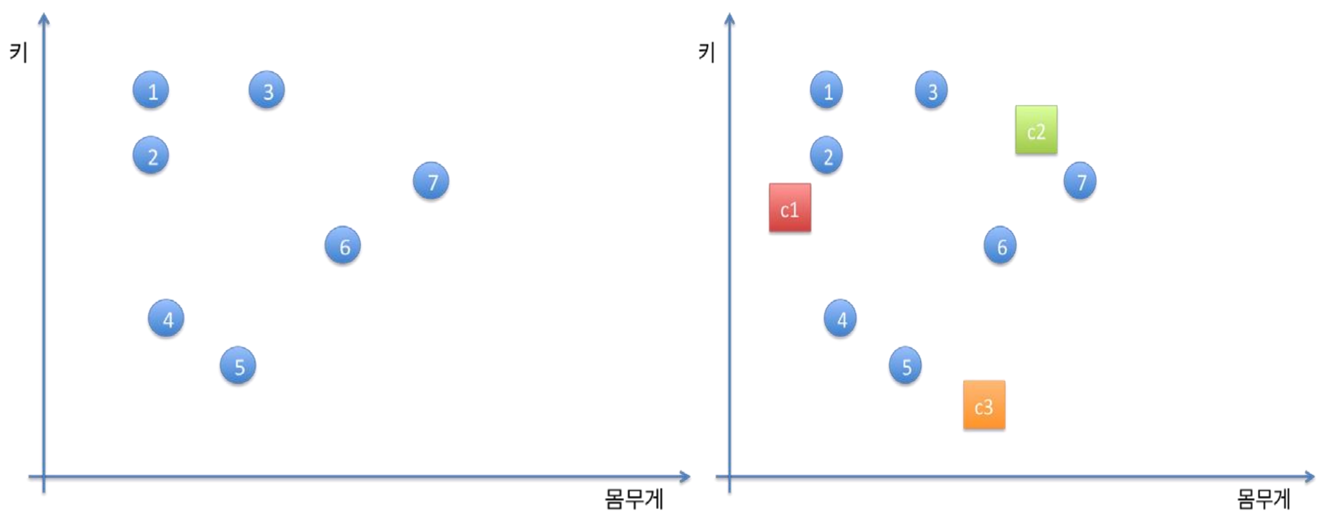
몇 개의 클래스로 분류할 것인지 정하기

- k 평균 알고리즘의 k는 몇 개의 클래스로 분류할 것인지를 나타내는 변수
- 이번 예제에서는 k를 3으로 설정

클러스터의 최초 중심 설정

- k 평균의 표준 알고리즘은 클러스터의 최초 중심을 무작위로 설정
- 하지만 경우에 따라 최초 중심을 k 평균 모델에 부여할 수 있음.

그림 4.7.1 무작위로 클러스터 중심 정하기



4.7 군집화

- 데이터의 순회가 완료되면 다시 클러스터의 중심을 소속된 데이터들의 중앙으로 옮김.
- 이 과정을 소속된 데이터가 변경되지 않을 때까지 또는 클러스터의 중심이 변경되지 않을 때까지 반복

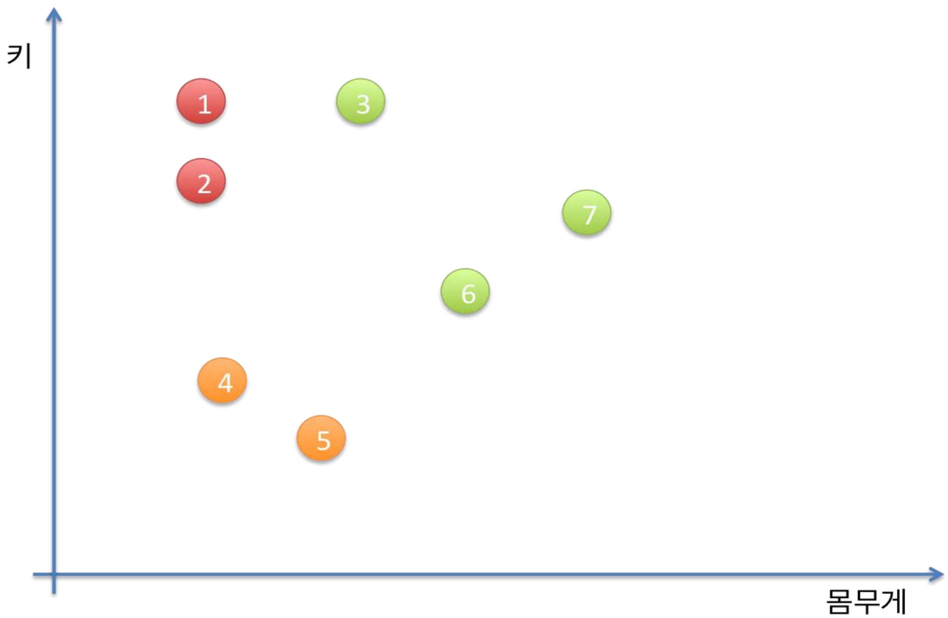


그림 4.7.7 kmean 완료 후 군집화된 데이터

4.8 선형회귀

- 선형회귀(linear regression) 분석이란 관찰된 데이터들을 기반으로 하나의 함수를 구해서 관찰되지 않은 데이터의 값을 예측하는 것

4.8.1 [이론] 선형회귀

- 선형회귀 모델이란 회귀 계수를 선형적으로 결합할 수 있는 모델을 의미
- 예제: 한 축구 선수의 슛 횟수와 획득한 점수표

표 4.8.1 축구선수의 슛 횟수와 득점표

슛 횟수	득점
1	1
3	3
5	5

4.8 선형회귀

- 이 축구 선수가 4번의 슛을 시도할 경우 몇 점을 획득?
- y = ax + b라는 1차함수를 생각할 경우 손쉽게 y = x라는 함수공식이 관찰된 데이터에 100%적용된다는 것을 확인할 수 있음

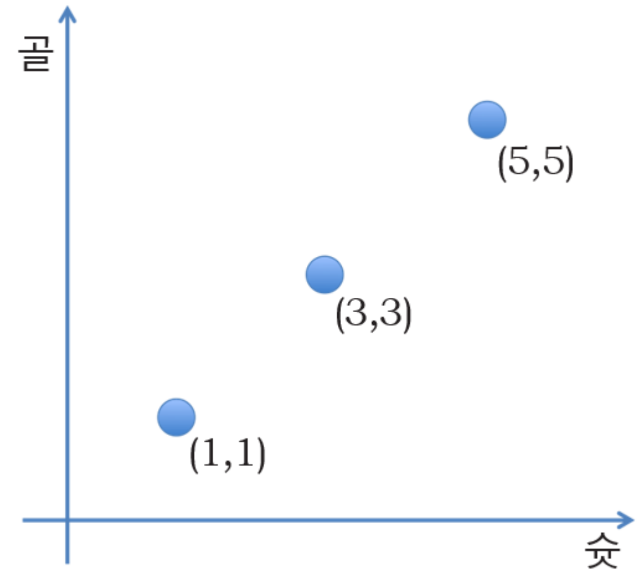


그림 4.8.1 축구선수 데이터 2차원 시각화

- 이 축구 선수가 4번의 슛을 쏠 경우 y = x라는 함수에 대입했을 때, 총 4개의 골을 성공할 것이라고 예측할

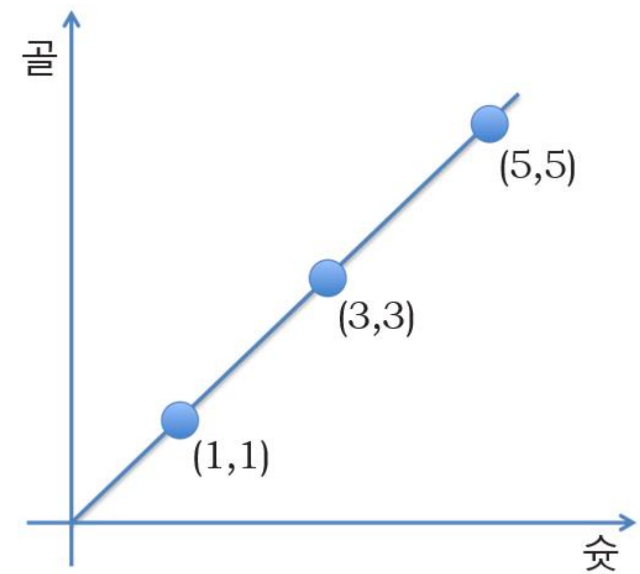


그림 4.8.2 축구 선수 데이터와 일치하는 함수 시각화

4.8 선형회귀

평균 제곱 오차로 더 나은 회귀 함수 선택하기

- $y=ix$ 라는 한 가지의 회귀 계수만을 사용한 함수로 표현해야 한다고 가정할 경우 가장 적당한 i 구해야 함

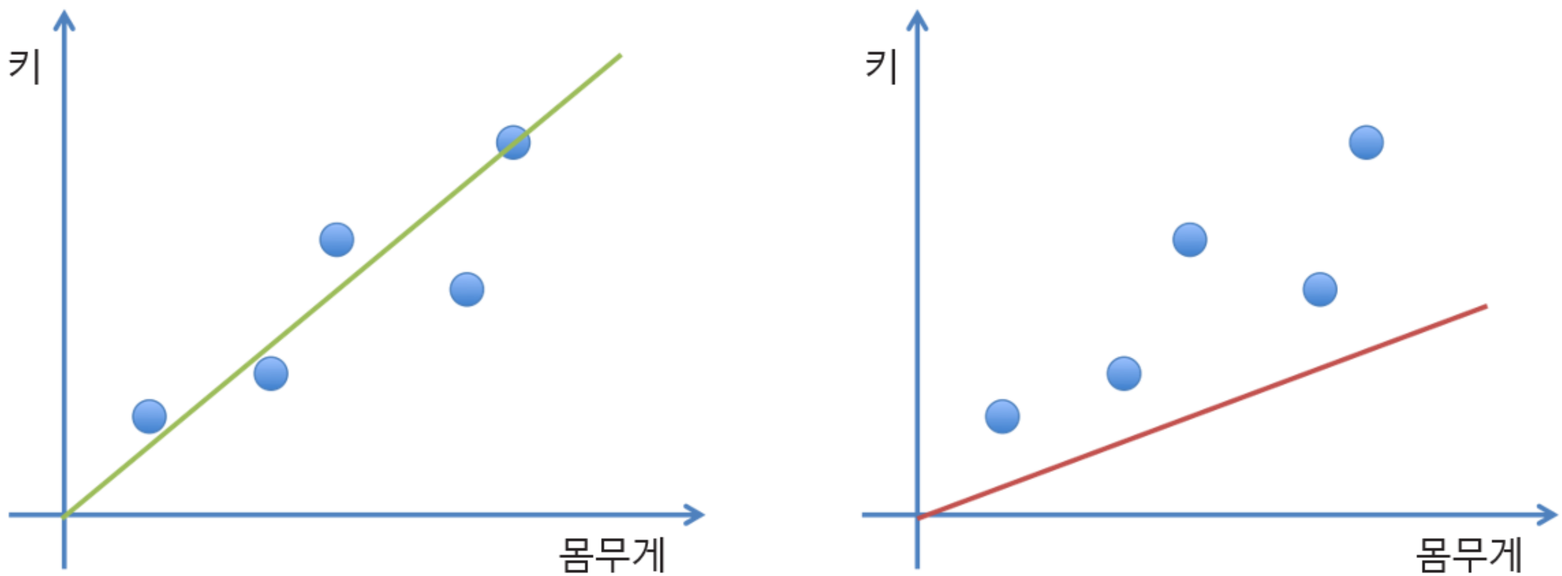


그림 4.8.7 몸무게 키의 상관관계에 적합한 함수 찾기

- 왼쪽과 오른쪽 선 중에서 왼쪽 선이 데이터와 더욱 밀접해 있고, 그에 따라 데이터 분포를 더 잘 표현한 것임을 확인할 수 있음.

4.8 선형회귀

- 데이터포인트에서 함수의 선까지의 거리를 더하는 방법으로 왼쪽의 함수가 더 나은 회귀함수라고 알 수 있음

$$\text{error} = (y_i - \ i x_i)$$

- 실무에서는 이 에러값보다는 에러값을 제곱한 값을 더 많이 사용.

$$\text{square error} = (y_i - \ i x_i)^2$$

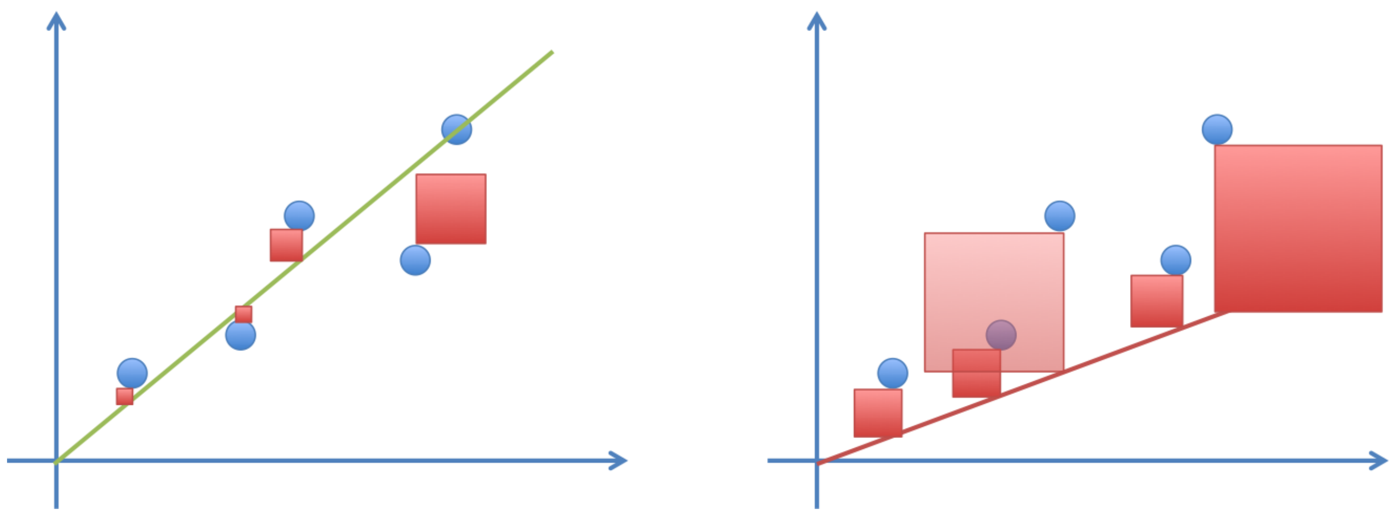


그림 4.8.8 회귀 함수와 실제 데이터 간의 거리 시각화

4.9 로지스틱 회귀

- 선형 회귀의 예측값은 수치값으로 나와서 참 또는 거짓을 분류하는 문제에 적합하지 않음
- 배울로지스틱 회귀[36] 모델은 선형 회귀를 입력으로 받아 특정 레이블로 분류하는 모델

4.9.1 [이론] 로지스틱 회귀

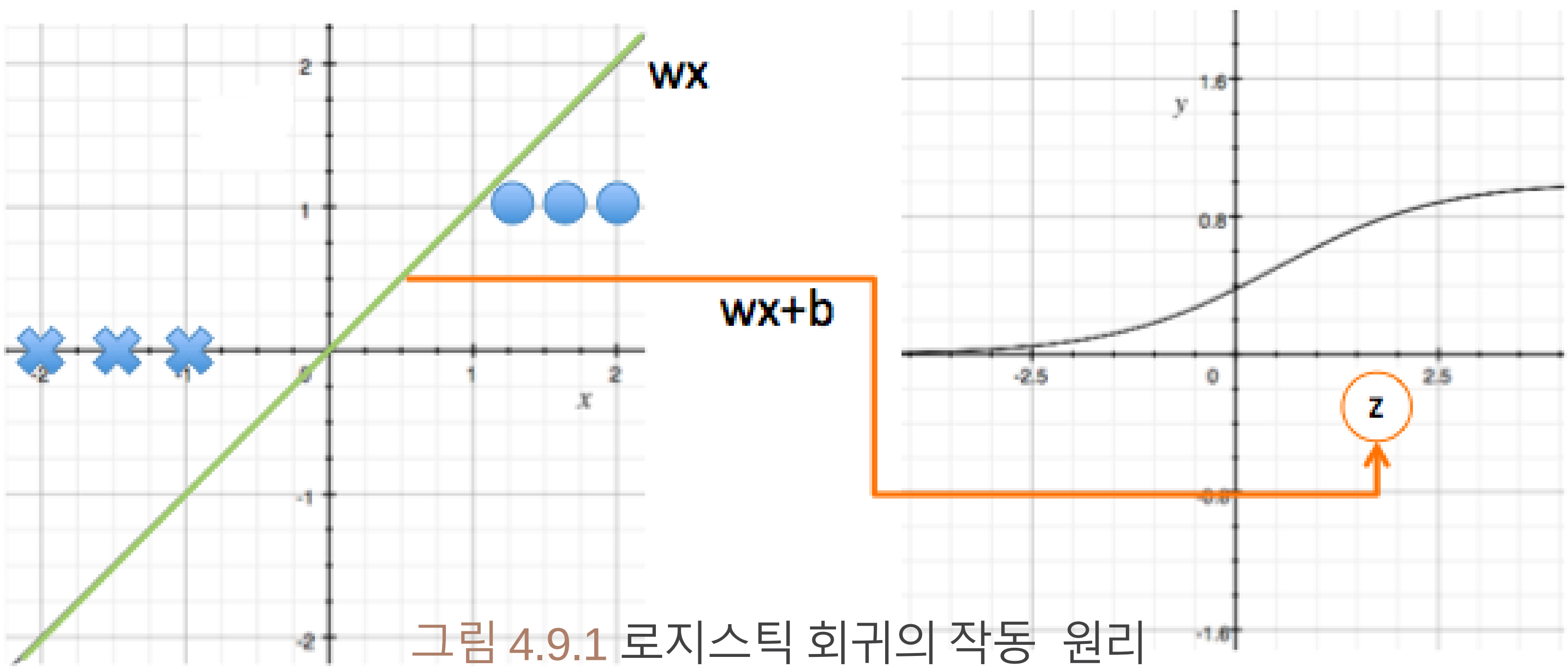


그림 4.9.1 로지스틱 회귀의 작동 원리

4.9 로지스틱 회귀

- 왼쪽 선형 회귀 그래프는 x라는 입력이 들어왔을 때 wx라는 결괏값을 출력.
- 결괏값 wx와 편향값 b가 합쳐진 값(z)는 다시 오른쪽 그래프의 입력값으로 들어가서 결괏값 y를 출력.
- 오른쪽 그래프의 이름은 시그모이드 함수.
- 오른쪽 그래프에서 볼 수 있듯이, 시그모이드 함수는 입력값이 크면 클수록 1이라는 값으로 수렴하고, 입력값이 작으면 작을수록 0이라는 값으로 수렴.
- 0부터 1까지의 값을 가지는 특성 때문에 시그모이드의 출력값은 확률로 사용될 수 있고, 출력값이 0.5 이상일 경우에는 참, 0.5 이하일 경우에는 거짓이라고분류하는 분류 모델로 사용할 수 있음.

4.10 주성분 분석

- 고차원의 데이터를 저차원의 데이터로차원 축소하는 알고리즘
- 주로 고차원의 데이터를 3차원 이하의 데이터로 바꿔서 시각화하는 데 많이 사용되고, 유용한 정보만 살려서 적은 메모리에 저장하거나 데이터의 노이즈를 줄이고 싶을 때도 사용되는 알고리즘

4.10.1 [이론] 주성분 분석

- 아래 2차원 공간의 데이터들을 1차원 공간, 즉 직선상의 데이터로 변환

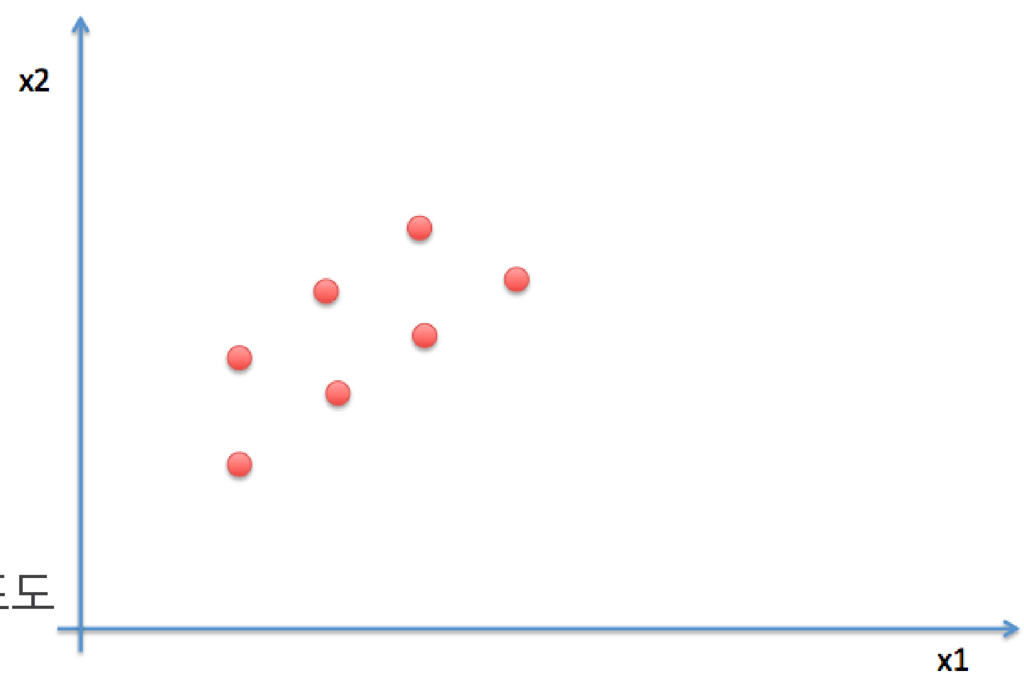


그림 4.10.1 2차원 데이터포인트 분포도

4.10 주성분 분석

- 주성분 분석의 특징은 데이터의 분산을 최대한 유지하면서 저차원으로 데이터를 변환하는 것.
- 만약 위의 2차원 데이터를 x1축으로 옮긴다면 다음과 같을 것.

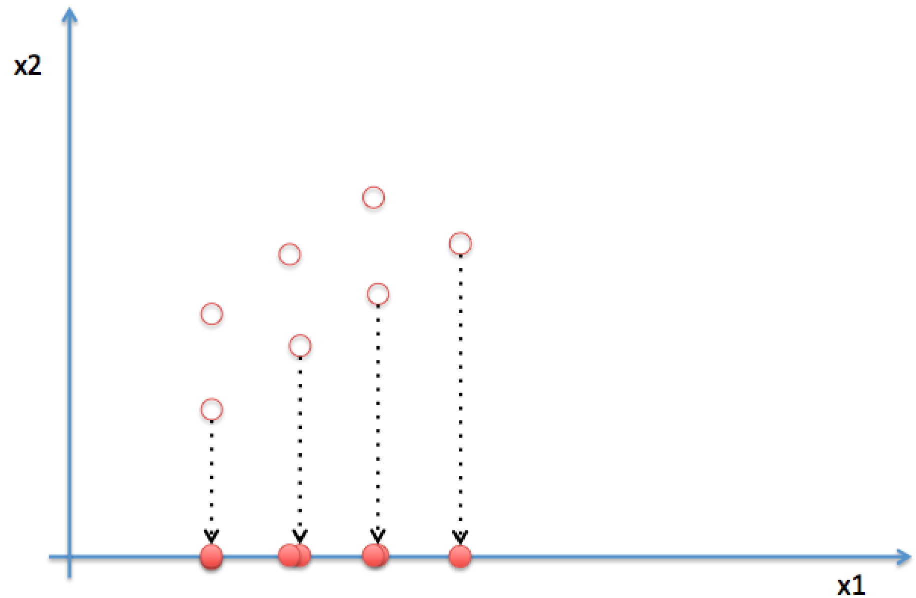


그림 4.10.2 2차원 데이터를 x1축에 사영시켰을 때의 시각화