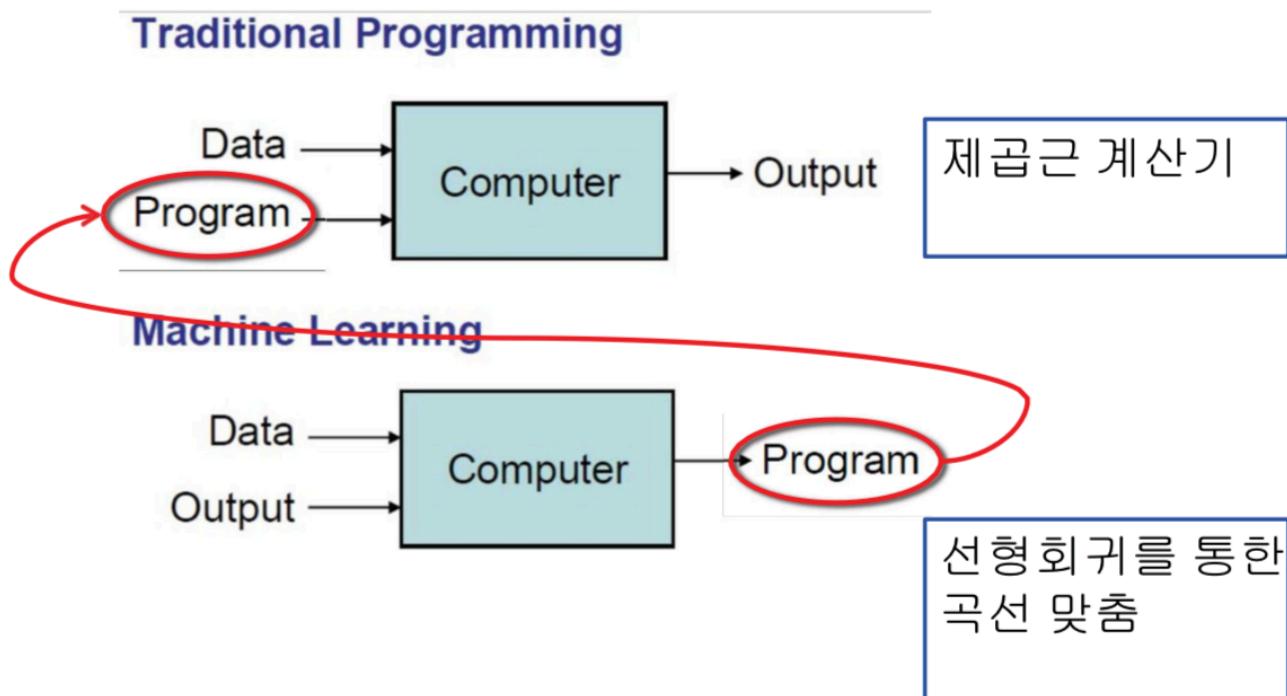


Chapter 11. Introduction to Machine Learning

발표자: 김재민

머신러닝

- 컴퓨터가 명시적으로 프로그래밍 되지 않고서 학습하게 만드는 연구 분야 – Arthur Samuel (1959)



학습 방법

- 선언적 지식
 - 기억 (Memorization)
 - 사실의 축적
 - 사실 관측 시간의 제한
 - 기억 장치 용량의 제한
 - 기말고사에 나올만한 것들을 기억하고 외우지 못한 사실들이 나오지 않기를 바라는 것

학습 방법

- 절차적 지식
 - 일반화 (Generalization)
 - 이전 사실로부터의 새로운 사실을 유추
 - 유추 방식의 정확도에 대한 제한
- 데이터의 내적 패턴으로부터 알고리즘이 그 패턴을 알아내고 그것을 이용해 프로그램을 작성해서 새로운 데이터에 대한 정보를 유추하는 것.

기본 패러다임

- 데이터 관측
 - 훈련 데이터
 - 질량에 따른 용수철의 이동 거리
- 코드 작성
 - 시스템이 데이터를 생성하는 과정을 유추할 수 있도록 할 것인가
 - 선형 회귀를 통한 다향식 곡선 맞추기
- 예측
 - 테스트 데이터
 - 다른 무게에 따른 이동거리 예측

기본 패러다임

- 데이터
 - 뜻볼 선수들의 포지션 라벨, 키, 몸무게
- 데이터를 만든 과정에 대한 유추
 - 통계를 바탕으로 포지션에 대한 정규 모델 유추
- 새로운 데이터 예측
 - 새로운 선수의 포지션 예측
- 패러다임 변형
 - 지도학습 : 특성/라벨 쌍의 집단을 가지고 새로운 입력값에 대한 레이블을 예측
 - 비지도 학습 : 특성 벡터(라벨 x)를 가지고 클러스터(집단에 대한 레이블) 형성

분류와 군집화에 대한 예

- 뉴잉글랜드 패트리어츠에 대한 데이터

- 이름, 키, 몸무게
 - 포지션의 종류로 레이블

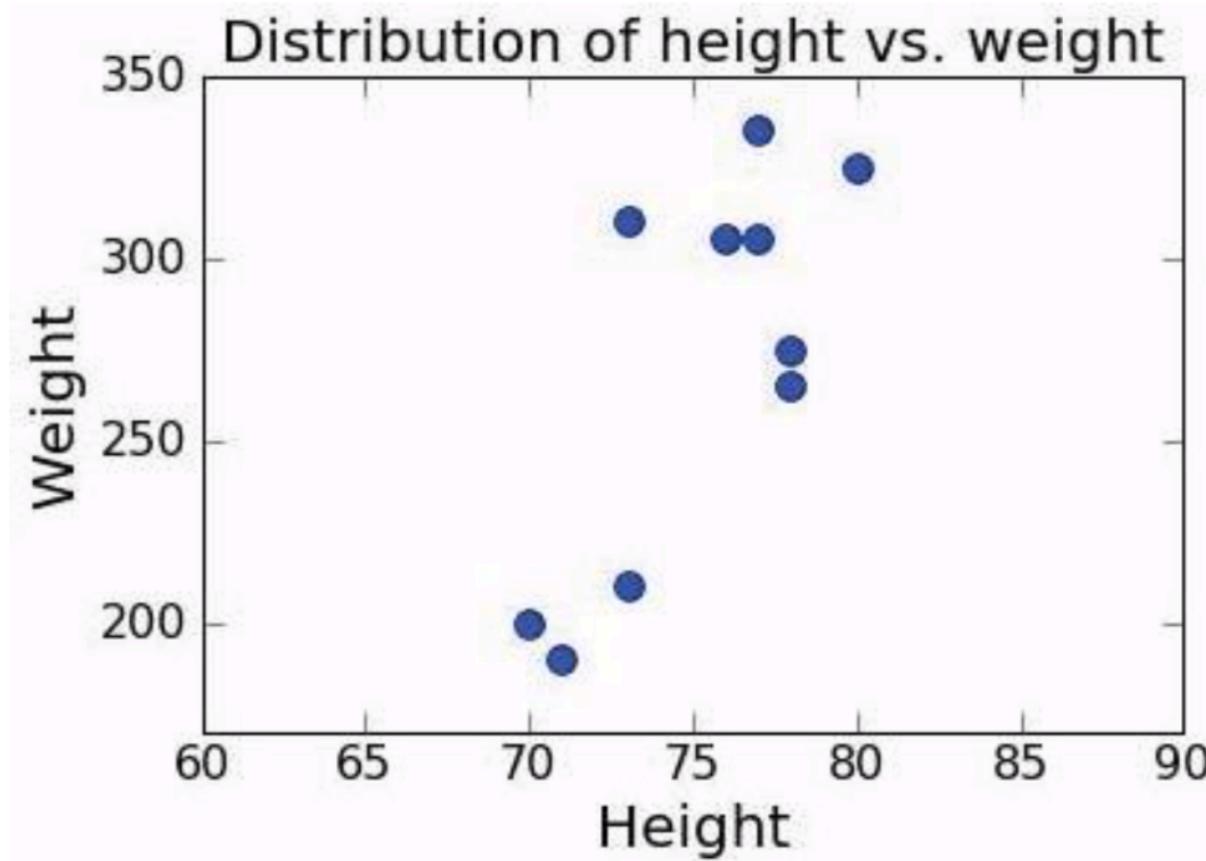
- 리시버:

- edelman = ['edelman', 70, 200]
 - hogan = ['hogan', 73, 210]
 - gronkowski = ['gronkowski', 78, 265]
 - amendola = ['amendola', 71, 190]
 - bennett = ['bennett', 78, 275]

- 라인맨:

- cannon = ['cannon', 77, 335]
 - solder = ['solder', 80, 325]
 - mason = ['mason', 73, 310]
 - thuney = ['thuney', 77, 305]
 - karras = ['karras', 76, 305]

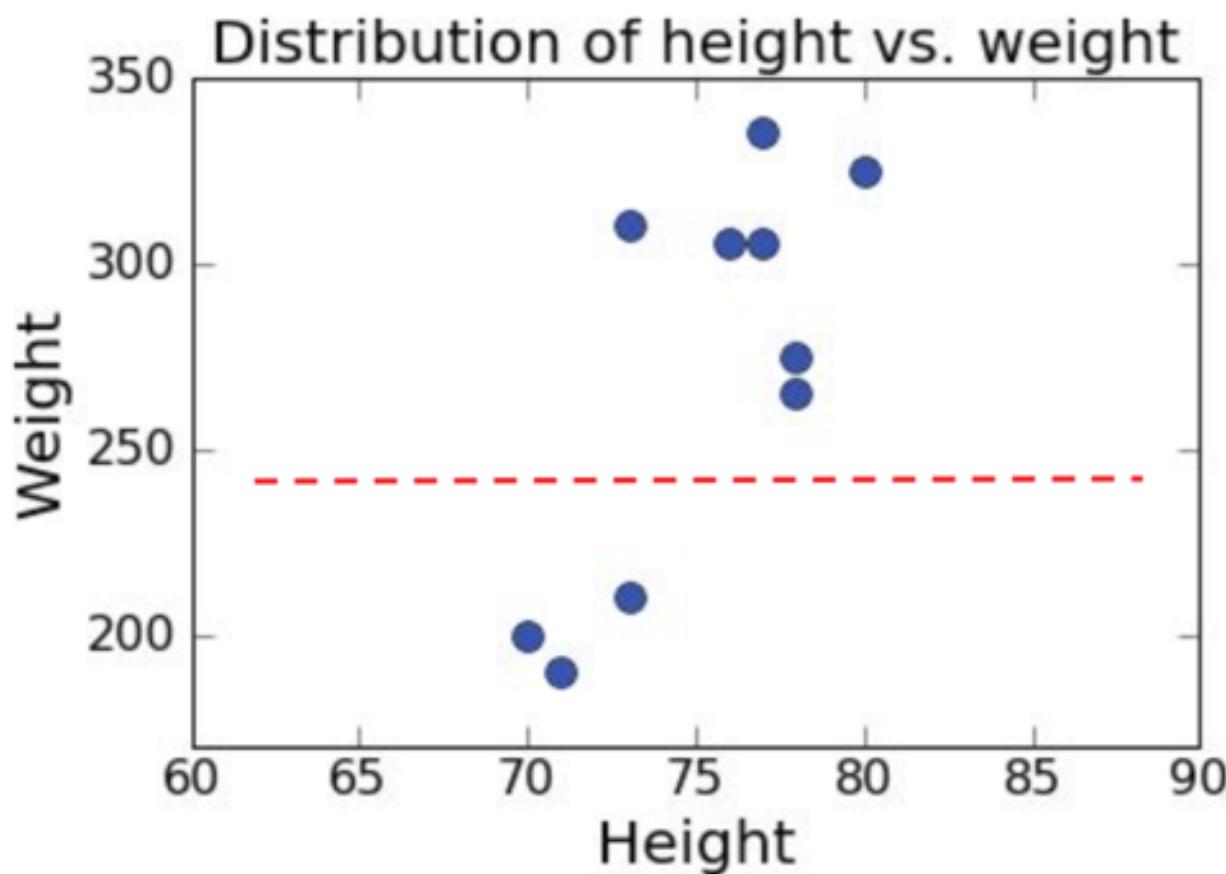
레이블이 없는 데이터



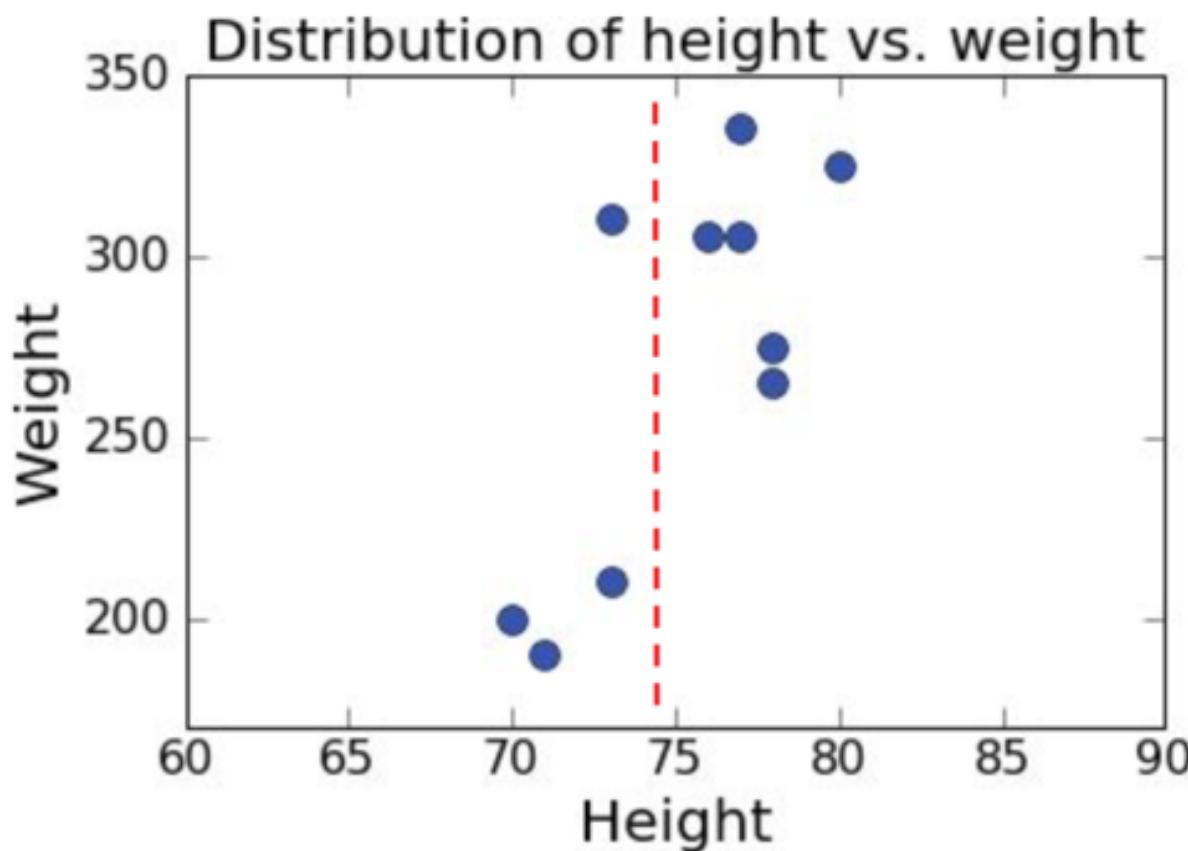
군집화 예제

- 예시들 간의 유사도를 결정해서, 자연스러운 집단으로 분류
 - 유사도 = 거리 측정
- 훈련데이터에 k개의 집단이 있고 레이블은 알 수 없음
 - k개의 샘플을 대표값으로 설정
 - 남은 샘플들을 대표값과 가장 가까운(거리가 최소) 집단에 넣기
 - 집단의 중간값을 새로운 대표값으로 설정
 - 변화가 생기지 않을 때 까지 반복

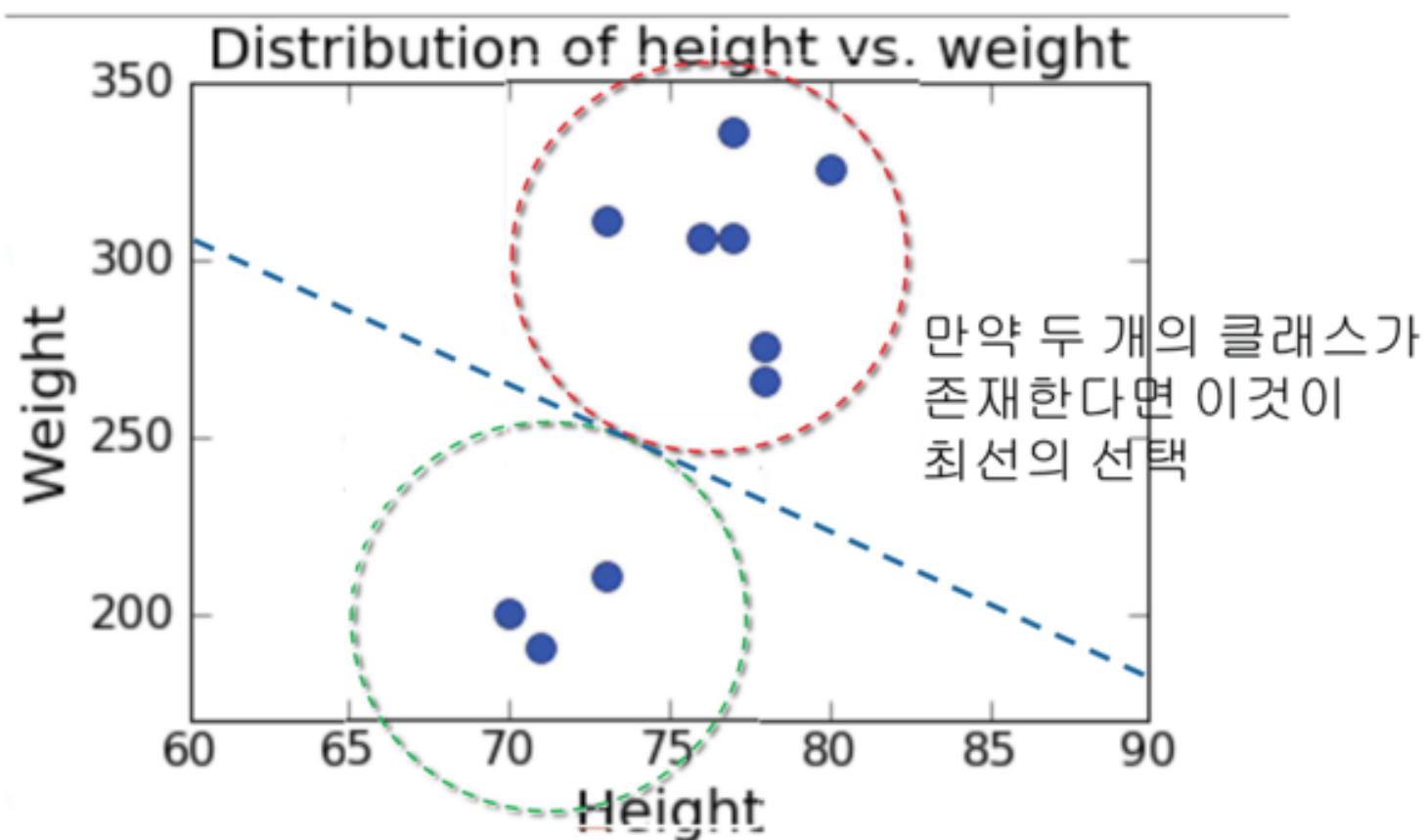
유사도 : 무게 기준



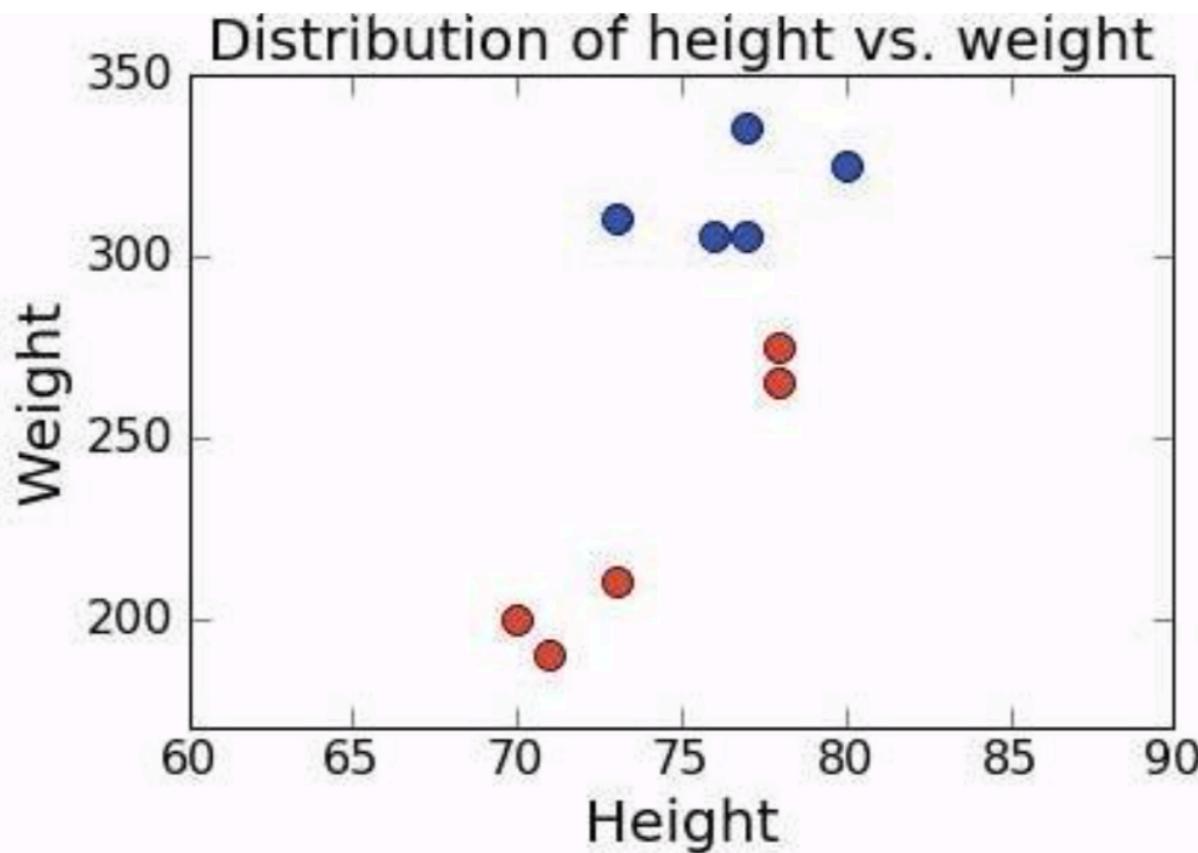
유사도 키 기준



두 속성(키, 몸무게)를 이용한 집단 형성



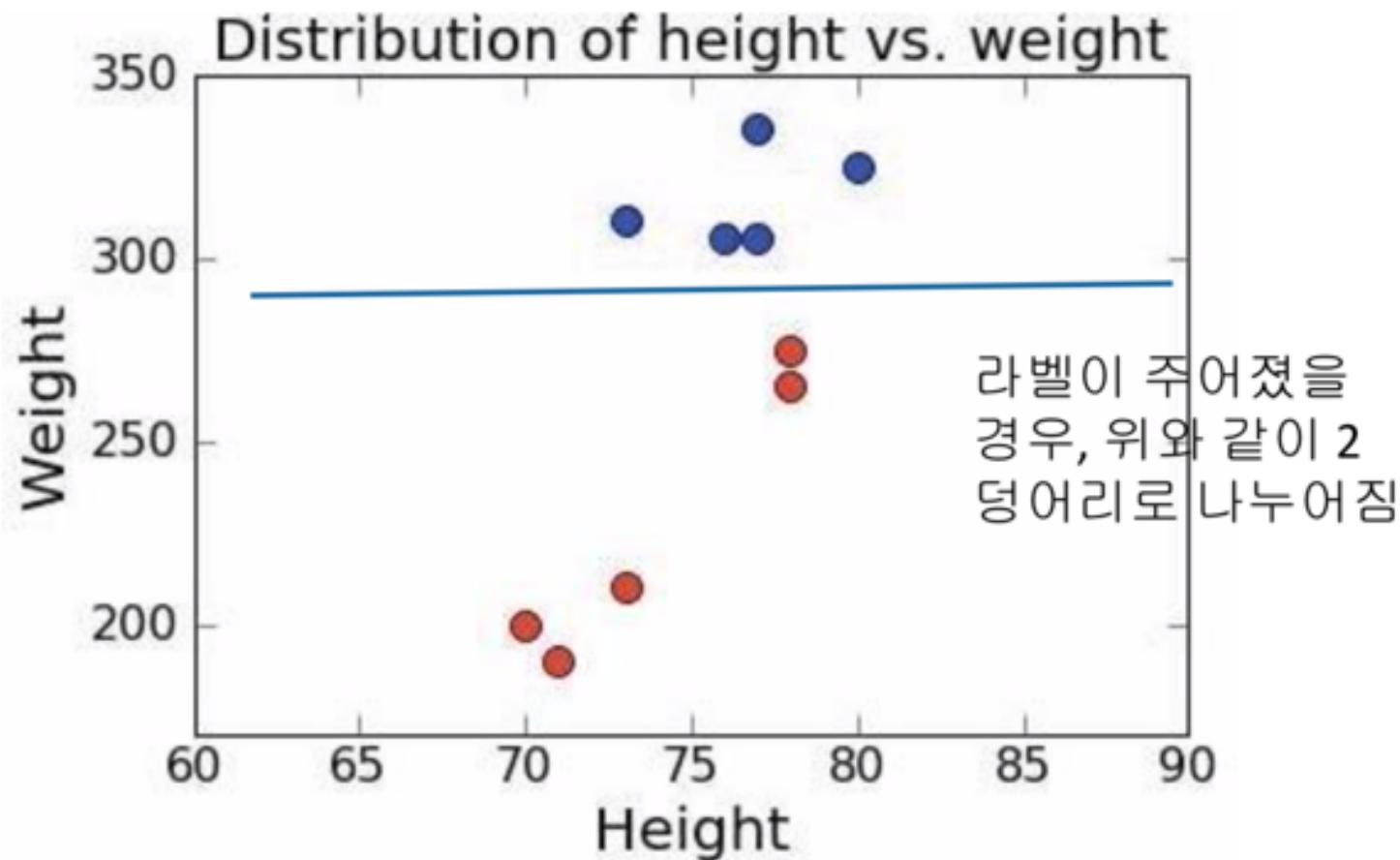
레이블이 있는 데이터



분급 표면 찾기

- 레이블이 있는 집단에서 집단을 구분하는 sub surface 찾기
 - 복잡도에 대한 제한 조건 존재
- 2차원 경우 어떤 선을 찾아야 하나의 레이블의 예시로부터 또 다른 레이블을 구분할 수 있을지 찾는 것.
- 예시가 잘 구분되었을 때는 매우 직관적이며
- 레이블된 집단의 예시가 겹칠 때에는 긍정 오류 또는 부정 오류 발생 가능

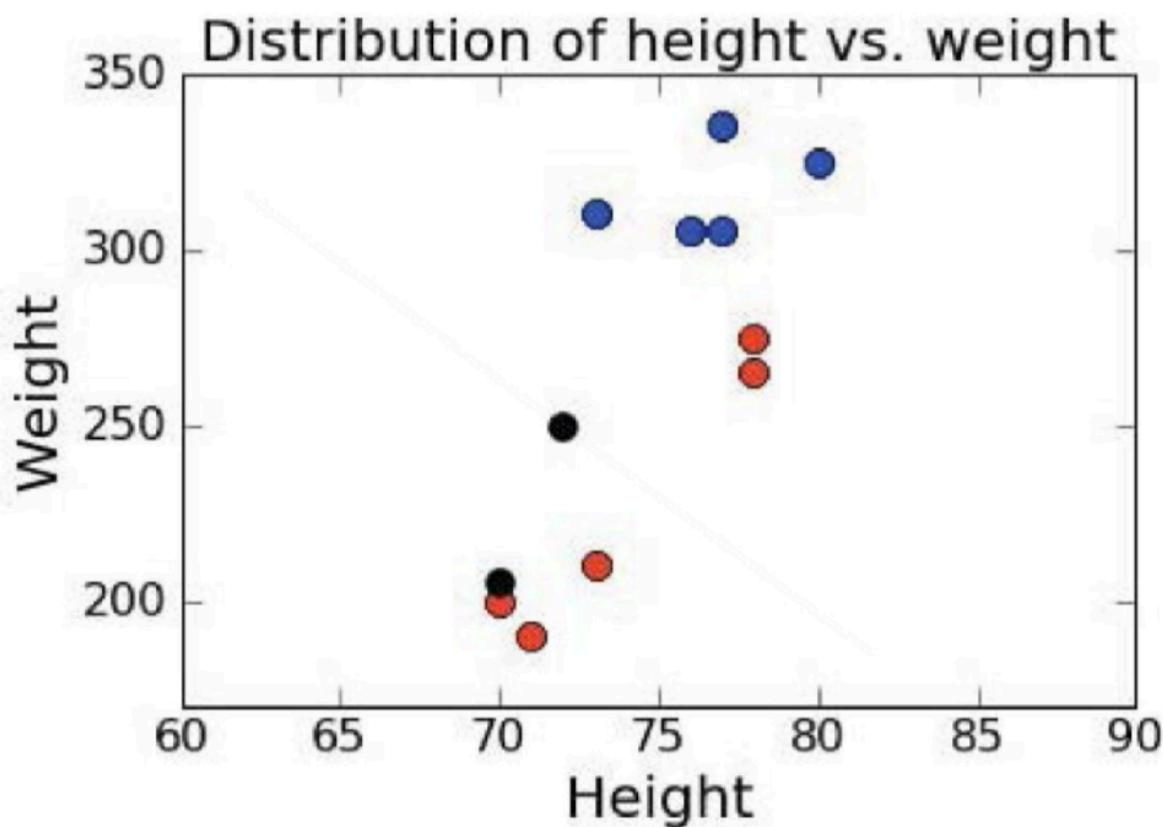
레이블이 있는 데이터



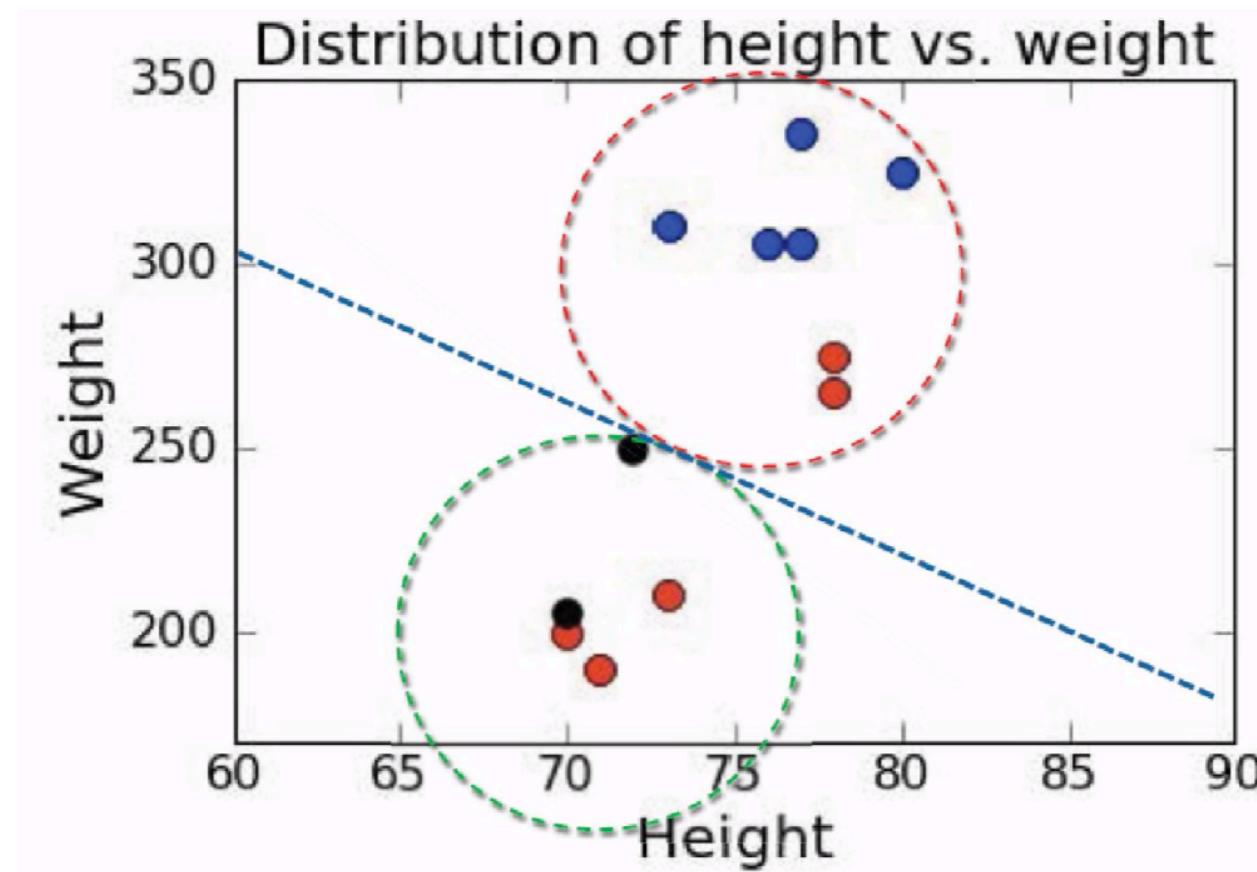
새로운 데이터 추가

- 리시버, 라인맨 구분법을 학습했다고 가정
- 새로운 예시가 다음과 정해졌을 시 어느 것에 가까운지 결정
 - blount = ['blount', 72, 250]
 - white = ['white', 70, 205]

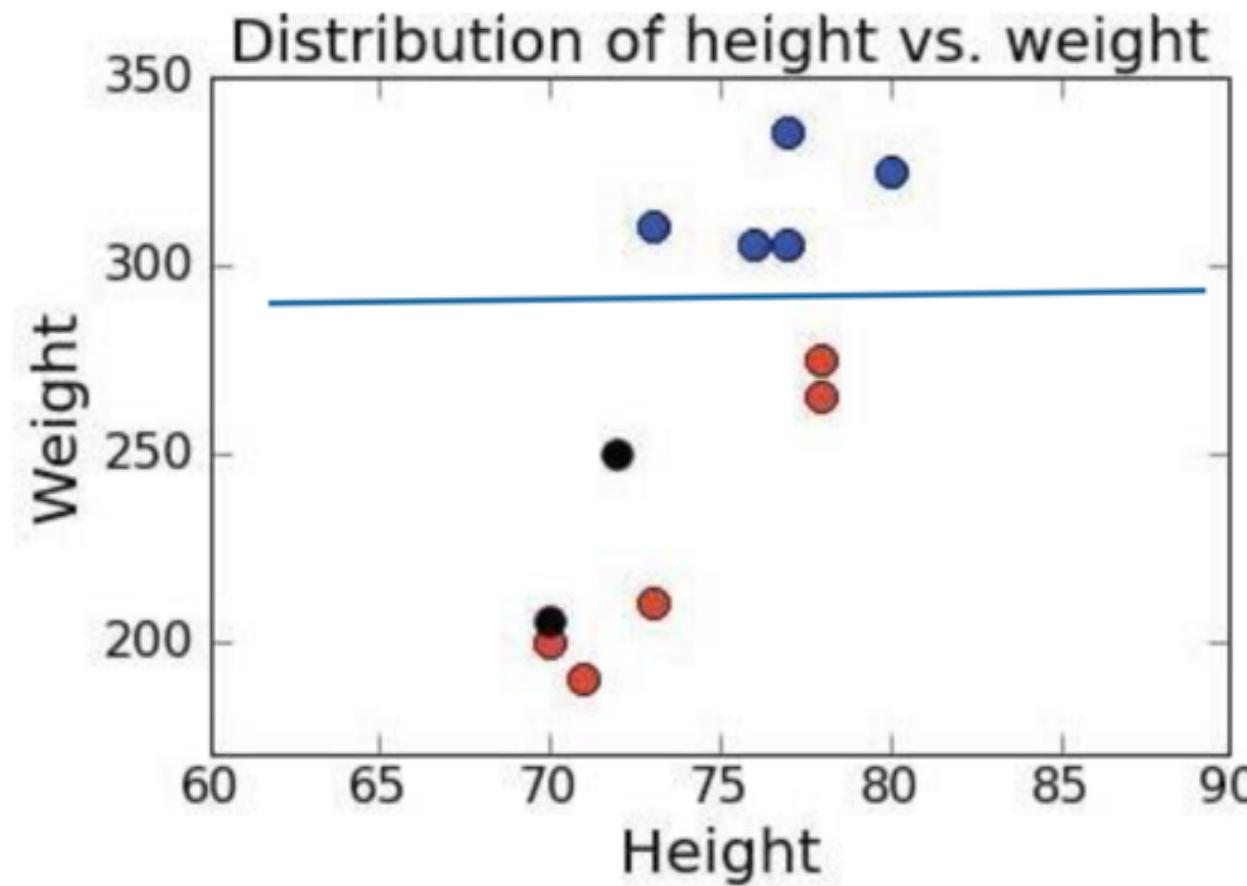
새로운 데이터 추가



레이블 없는 자료의 군집화



레이블 있는 자료의 분류



머신러닝 방식의 예

- 결과 집단으로 새로운 데이터에 레이블을 지정
- 레이블 있는 유사한 데이터 집단을 다른 집단과 구분하는 모델 학습
 - Overfitting 없이는 완벽히 집단을 구분하기는 어려움
 - 긍정오류, 부정오류를 비용으로 결정을 내릴 수 있음
 - 새로운 데이터에 레이블 지정

모든 머신러닝 방식의 요구사항

- 훈련 데이터와 평가 방식 결정
- 특성의 요소 결정
- 특성 벡터의 거리 계측법
- 목적 함수와 제한 조건
- 모델 학습의 최적화 방식

특성 표시 방식

- 특성 설계
 - 일반화 할수 있도록 특성 벡터로 예시
 - 예시
 - 과거의 100개의 예시를 이용해서 학기 초에 수업에서 누가 A를 받을지 예측
 - 몇몇 특성은 유용함 : GPA, 프로그래밍 경험
 - Overfitting : 출생 월, 눈 색상
 - 어떻게 특성들을 선택 할 것인가?
 - 신호 대 잡음비를 최대화
 - 가장 많은 정보를 가진 특성을 최대화하고 그렇지 않은 것들을 제거하는 것

예시

특성						레이블
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes

초기 모델:

- 일반화하기에는 정보가 부족

예시

특성							레이블
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile	
Cobra	True	True	True	True	0	Yes	
Rattlesnake	True	True	True	True	0	Yes	

초기 모델:

- 산란
- 비늘
- 독
- 냉혈동물
- 다리 없음

예시

특성						레이블
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes

현재 모델:

- 비늘
- 냉혈동물
- 다리 없음

보아뱀은 모델에 맞지 않지만 파충류로 분류
모델 수정이 필요함

예시

Name	Egg-laying	특성			레이블	
		Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No

현재 모델:

- 비늘
- 냉혈동물
- 다리 없음

예시

특성						레이블
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes

현재 모델:

- 비늘
- 냉혈동물
- 0 혹은 4개의 다리

악어는 모델에 맞지 않지만 파충류로
분류
모델 수정이 필요함

예시

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No

현재 모델:

- 비늘
- 냉혈동물
- 0 혹은 4개의 다리

예시

특성

레이블

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

현재 모델:

- 비늘
- 냉혈동물
- 0 혹은 4개의 다리

연어와 비단뱀을 구분하는 규칙을 추가하기 어려움
(동일한 특성값을 가지기 때문)

예시

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

좋은 모델:

- 비늘
- 냉혈동물

완벽하지는 않지만 부정 오류(파충류가 아닌 것으로 분류된 것이 올바르게 레이블된 것)가 없음; 조금의 긍정오류가 있음(어떤 동물을 파충류로 잘못 레이블할 수 있음)

특성들 간의 거리를 측정해야함

- 어느 정도의 부정오류와 긍정 오류를 용납할 것인가?
- 적절한 특성을 찾아서 사용하려 할 때
- 어떤 집단을 분류하거나 하나의 분리선을 찾아서 구분 짓기 위함
- 따라서
 - 어떤 특성을 포함할지와 어떤 것을 제외할 지 결정
 - 훈련 예시들 같은 거리를 측정하는 방법을 결정
 - 특성 베터의 요소들의 가중치를 결정

동물들간의 거리 측정

- 동물 예시는 4개의 2진 특성(Boolean)과 한개의 정수로 이루어
짐

```
rattlesnake = [1,1,1,1,0]  
boa constrictor = [0,1,0,1,0]  
dart Frog = [1,0,1,0,4]
```

민코프스키 계측법

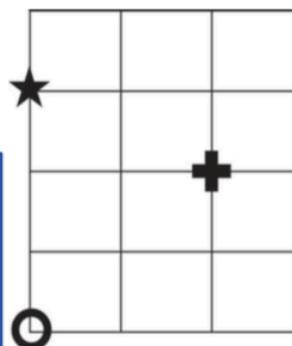
$$dist(X1, X2, p) = \left(\sum_{k=1}^{\text{len}} abs(X1_k - X2_k)^p \right)^{1/p}$$

p = 1: Manhattan Distance

p = 2: Euclidean Distance

특성 벡터 간의 거리를
측정해야 함

일반적으로는
유클리드 계측법을
사용하고 맨해턴은
여러 요소들을 서로
비교하기 어려울 때
유용하다



원이 십자가와 별 중
어느 것과 가까울까?

- 유클리드 거리
 - 십자가 - 2.8
 - 별 - 3
- 맨해턴 거리
 - 십자가 - 4
 - 별 - 3

동물들 간의 유클리드 거리

```
rattlesnake = [1,1,1,1,0]  
boa constrictor = [0,1,0,1,0]  
dartFrog = [1,0,1,0,4]
```

	rattlesnake	boa constrictor	dart frog
rattlesnake	-	1.414	4.243
boa constrictor	1.414	-	4.472
dart frog	4.243	4.472	-

유클리드 거리를 이용해서 방울뱀과 보아뱀이
독침개구리보다 훨씬 서로 가깝다는 것을 알 수 있음

악어 추가

- alligator = Animal('alligator', [1,1,0,1,4])
- animals.append(alligator)
- compareAnimals(animals, 3)

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	--	1.414	4.243	4.123
boa constrictor	1.414	--	4.472	4.123
dart frog	4.243	4.472	--	1.732
alligator	4.123	4.123	1.732	--

악어는 뱀보다 독침개구리와 가깝다 - 왜 그럴까?

- 개구리와 3개의 특성이 다르지만, 보아뱀과는 두 개 뿐이다
- “다리 개수”의 규모는 0부터 4이고, 다른 특성은 0부터 1
- “다리 개수”의 규모가 불균형적으로 큼

이진특성 사용

```
rattLesnake = [1,1,1,1,0]  
boa constrictor = [0,1,0,1,0]  
dartFrog = [1,0,1,0,1]  
ALLigator = [1,1,0,1,1]
```

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	-	1.414	1.732	1.414
boa constrictor	1.414	-	2.236	1.414
dart frog	1.732	2.236	-	1.732
alligator	1.414	1.414	1.732	-

이제 악어가 개구리보다 뱀에 가까워짐

- 더 타당함

특성 설계의 중요성

지도학습 vs 비지도 학습

- 군집
 - 레이블이 없는 데이터를 가지고 예시들이 서로 이웃한 집단을 찾는것
 - 집단의 도심을 학습 클래스의 정의로 사용
 - 새로운 데이터를 가장 가까운 집단으로 지정
- 분류
 - 레이블이 있는 데이터를 가지고 surface 복잡도의 제한 조건을 만족하면서 예시들을 가장 잘 분리하는 방법(overfitting은 피해야함)

모델을 학습 할 때 고려할 것

- 예시들 간의 거리 계측법
- 특성 벡터 선택
- 모델의 복잡도에 대한 제한 조건
 - 지정된 수의 집단
 - 표면 구분 복잡도
 - overfitting을 피해야함(각자의 예시가 자기 자신의 집단인 경우, 너무 복잡한 구분 표면)

군집화 접근법

- 최적의 집단 수를 어떻게 결정할 것인가?
- 최적의 특성과 거리 계측법을 어떻게 결정할 것인가?

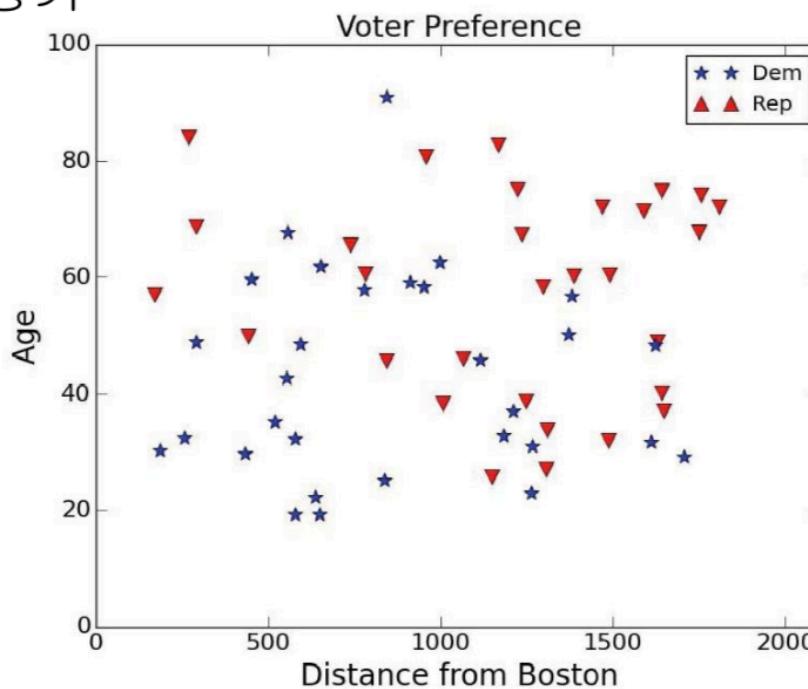
분류 접근법

- 레이블 있는 예시들의 클래스를 구분하는 특성 공간의 경계선을 찾아야 함
 - 클래스를 구분하는 단순한 표면 찾기
 - 클래스를 구분하는 복잡한 표면 찾기
 - 투표 방식을 사용
 - K-최근접 훈련 예시를 찾고 다수 투표를 이용해 레이블 결정
- 문제점
 - 어떻게 데이터의 Overfitting을 피할 것인가?
 - 어떻게 성능을 평가할 것인가?
 - 어떻게 최적의 특성을 선택할 것인가?

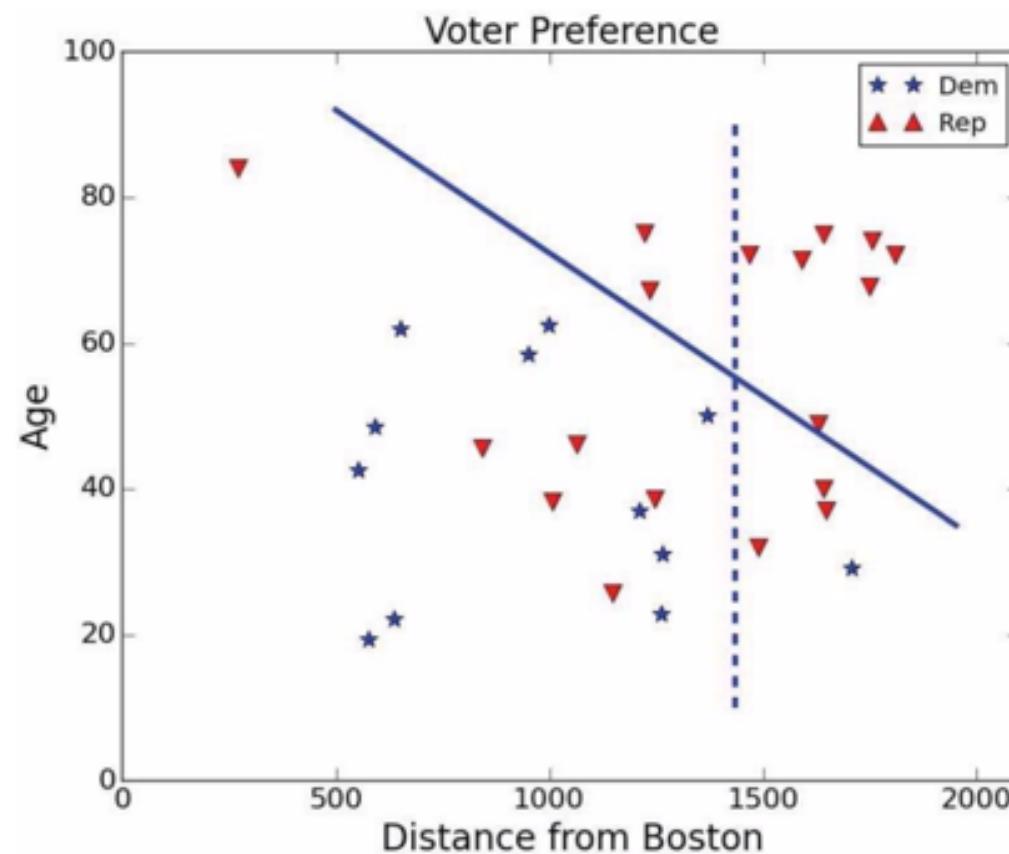
분류

- 훈련 데이터의 오류를 최소화
 - 데이터에 곡선 맞추기와 유사
- 훈련 데이터를 평가

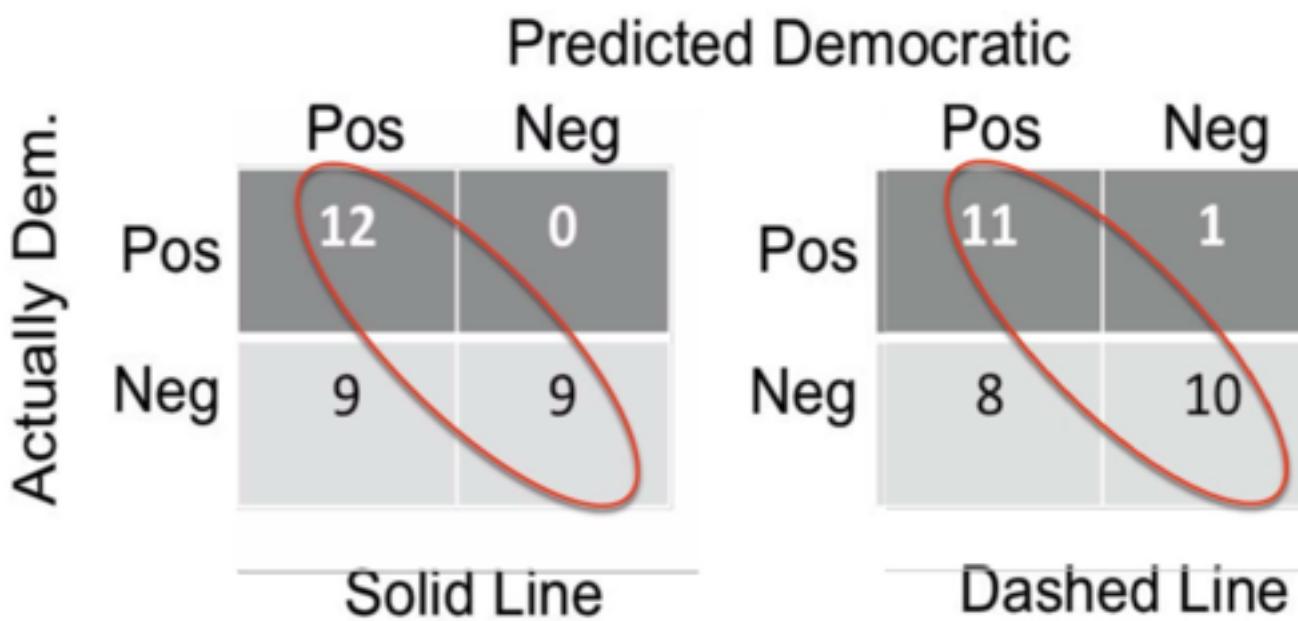
투표 선호도,
나이와
보스턴으로부터
의 거리



훈련 세트의 두가지 모델



오차행렬

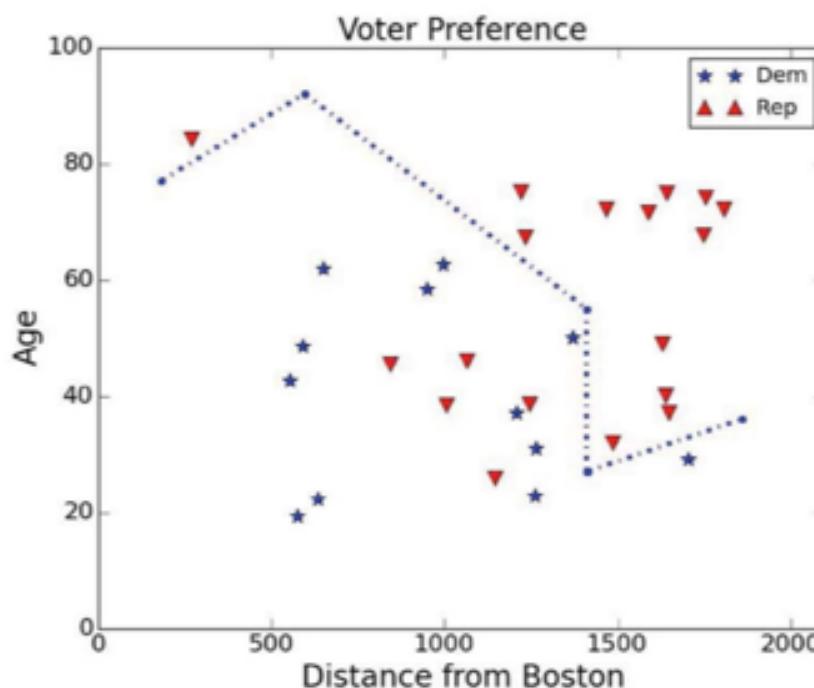


정확도 (accuracy)

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

- 두 개의 모델 모두 0.7
 - 어느 것이 나을까?
- 훈련 오류가 더 적은 모델을 찾을 수 있을까?

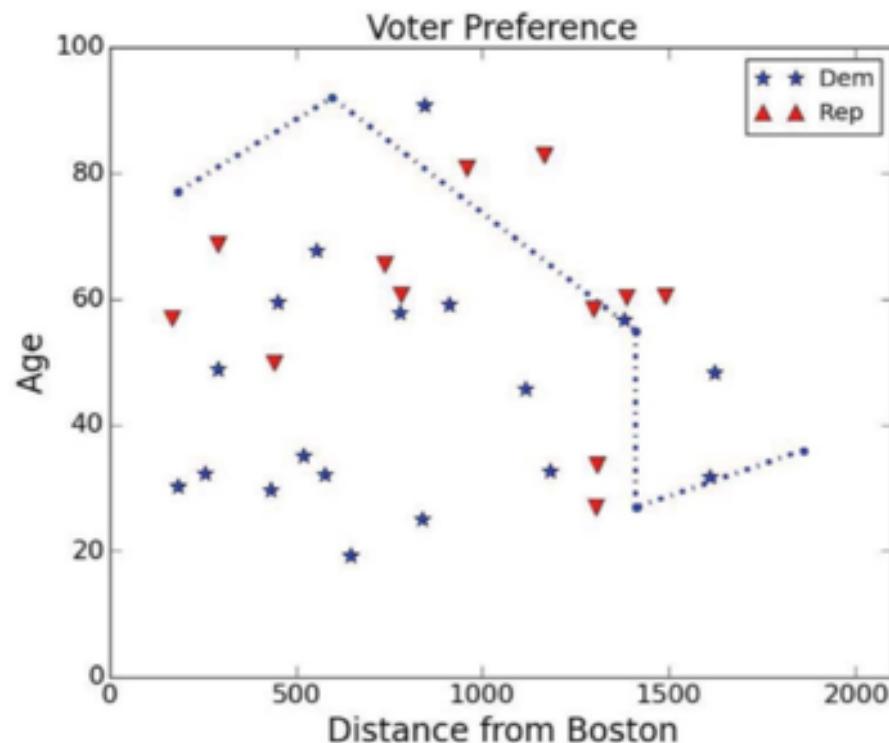
더 복잡한 모델



TP = 12, FP = 5, TN = 13, FN = 0

Accuracy = 25/30 = 0.833

테스트 데이터에 모델 적용



PPV, sensitivity, Specificity

$$\text{positive predictive value} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

- 진한 선 모델: .57
- 점선 모델: .58
- 복합 모델, (훈련): .71
- 복합 모델, (실험): .78
- “민감도” v.s. “특정성”

$$\text{sensitivity} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

$$\text{specificity} = \frac{\text{true negative}}{\text{true negative} + \text{false positive}}$$

올바르게
찾은 비율

올바르게
제외한 비율