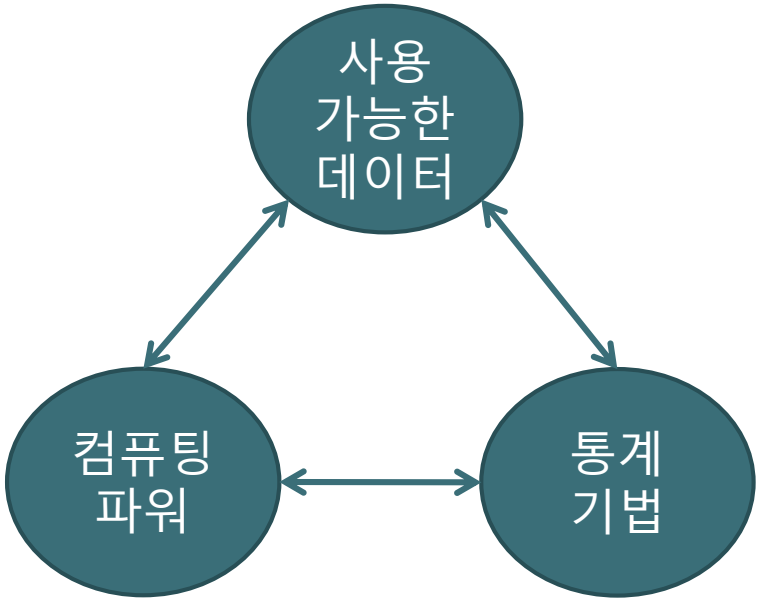
머신러닝(기계 학습) 소개

기계 학습

- 데이터를 지능 행위로 변환하는 컴퓨터 알고리즘을 연구하는 분야
- 기계 학습 알고리즘은 데이터마이닝의 필요 조건

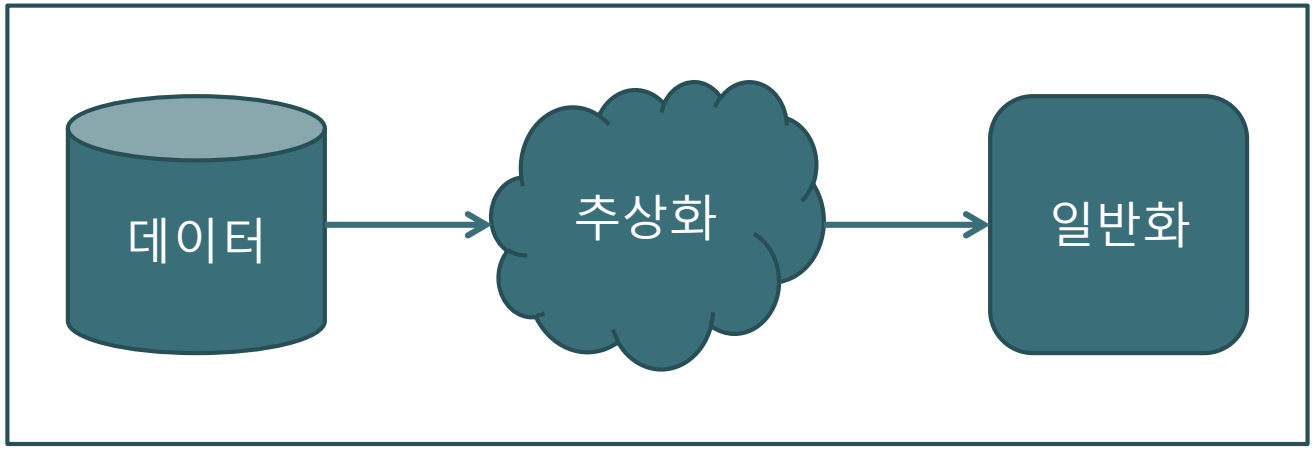


기계 학습의 사용

- 선거의 결과 예상
- 이메일에서 스팸 메시지 분류
- 도로 상태에 따른 신호 조절
- 자연 재해의 경제적 측정 산출
- 서비스 업체를 바꾸려는 고객 조사
- 자동 운전 차량과 자동 항법 비행기 구현
- 기부 능력 여부 판단
- 고객에 따른 타겟 광고

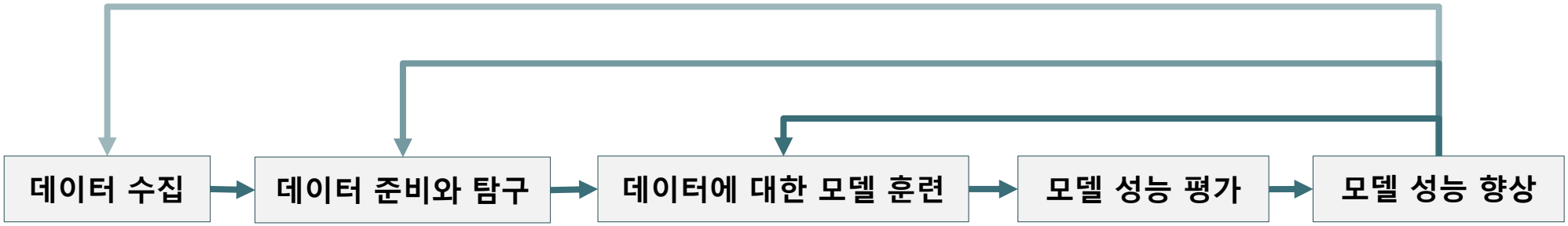
기계 학습 방법

- 데이터 입력(Data input) : 미래의 추론을 위한 사실적 근거를 제공하기 위해 관찰, 기억 공간을 활용
- 추상화(Abstraction) : 넓은 표현성으로 데이터를 변환하는 것과 관련
- 일반화(Generalization) : 실행하기 위해 추상화된 데이터를 사용



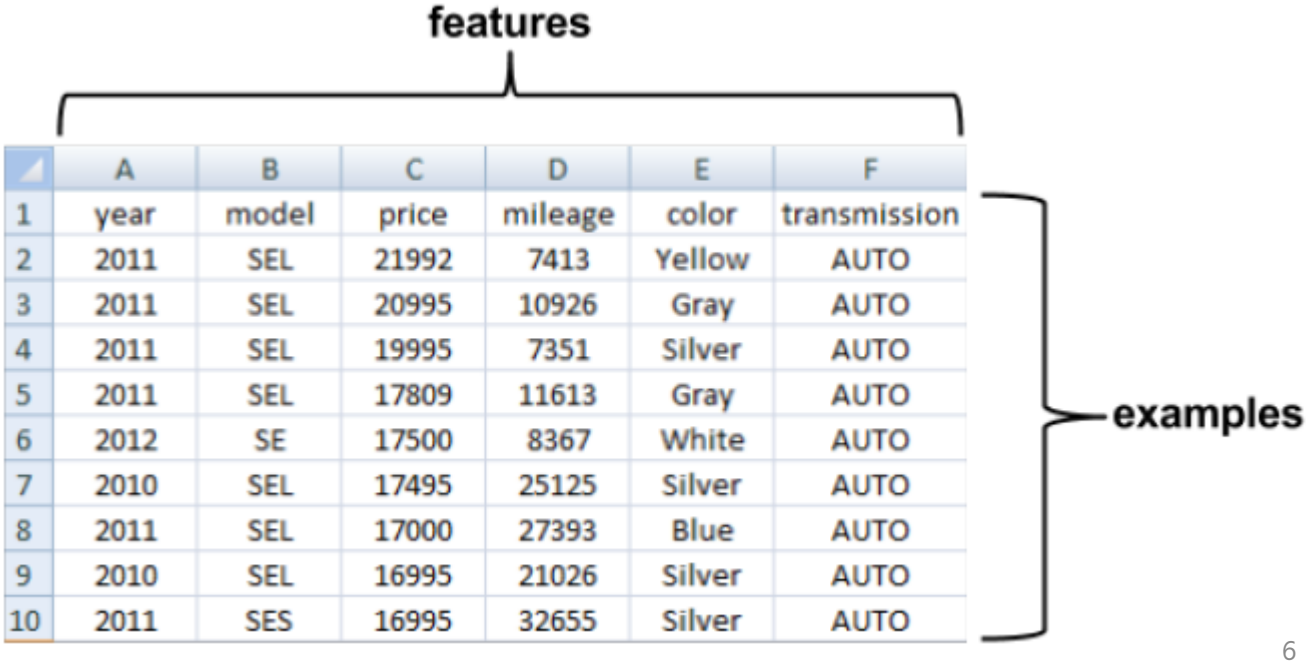
기계 학습을 적용하는 단계

- 데이터 수집
- 데이터 준비와 탐구
  - 80% 의 노력을 데이터에 들어야 함
- 데이터에 대한 모델 훈련
- 모델 성능 평가
- 모델 성능 향상



데이터의 속성

- 숫자로 측정된 특성
  - 수치(numeric)
- 범주로 측정된 특성
  - 범주적(categorical)
  - 명목적(nominal)
- 범주형 변수가 순서화된 리스트
  - 서수적(ordinal)



	A	B	C	D	E	F
1	year	model	price	mileage	color	transmission
2	2011	SEL	21992	7413	Yellow	AUTO
3	2011	SEL	20995	10926	Gray	AUTO
4	2011	SEL	19995	7351	Silver	AUTO
5	2011	SEL	17809	11613	Gray	AUTO
6	2012	SE	17500	8367	White	AUTO
7	2010	SEL	17495	25125	Silver	AUTO
8	2011	SEL	17000	27393	Blue	AUTO
9	2010	SEL	16995	21026	Silver	AUTO
10	2011	SES	16995	32655	Silver	AUTO

기계 학습 알고리즘의 종류

- 지도 학습기(supervised learner)
  - 예측 모델(predictive model)

- 자율 학습기(unsupervised learner)
  - 기술 모델(descriptive model)

데이터에 맞는 적당한 알고리즘 선정

	모델	태스크
지도 학습 알고리즘	최근접 이웃	분류
	나이브 베이즈	분류
	결정 트리	분류
	로지스틱 회귀	분류
	선형 회귀	수치 예측
	모델 트리	수치 예측
	신경망	다중 용도
	서포트 벡터 머신	다중 용도
자율 학습 알고리즘	연관 규칙	패턴 탐지
	K 평균 군집화	군집화

게으른 학습 :
최근접 이웃을 사용한 분류

최근접 이웃을 사용한 분류의 이해

- 유사한 범주의 아이템들은 수치이고 동질 데이터
- 복잡하거나 이해하기 어려운 속성과 범주에 관계된 분류 태스크에 적합
- 개념을 정의하기 힘들다면 최근접 이웃이 적합
- 그룹 사이에 뚜렷한 구별이 없다면 대체로 경계선을 식별하는 데 적합하지 않다
- 예제
  - 이미지나 비디오에서 얼굴과 글자를 인식하는 컴퓨터 비전 애플리케이션
  - 개인별 추천 영화 예측
  - 특정 단백질과 질병을 추출하는데 사용하는 유전자 데이터의 패턴 식별

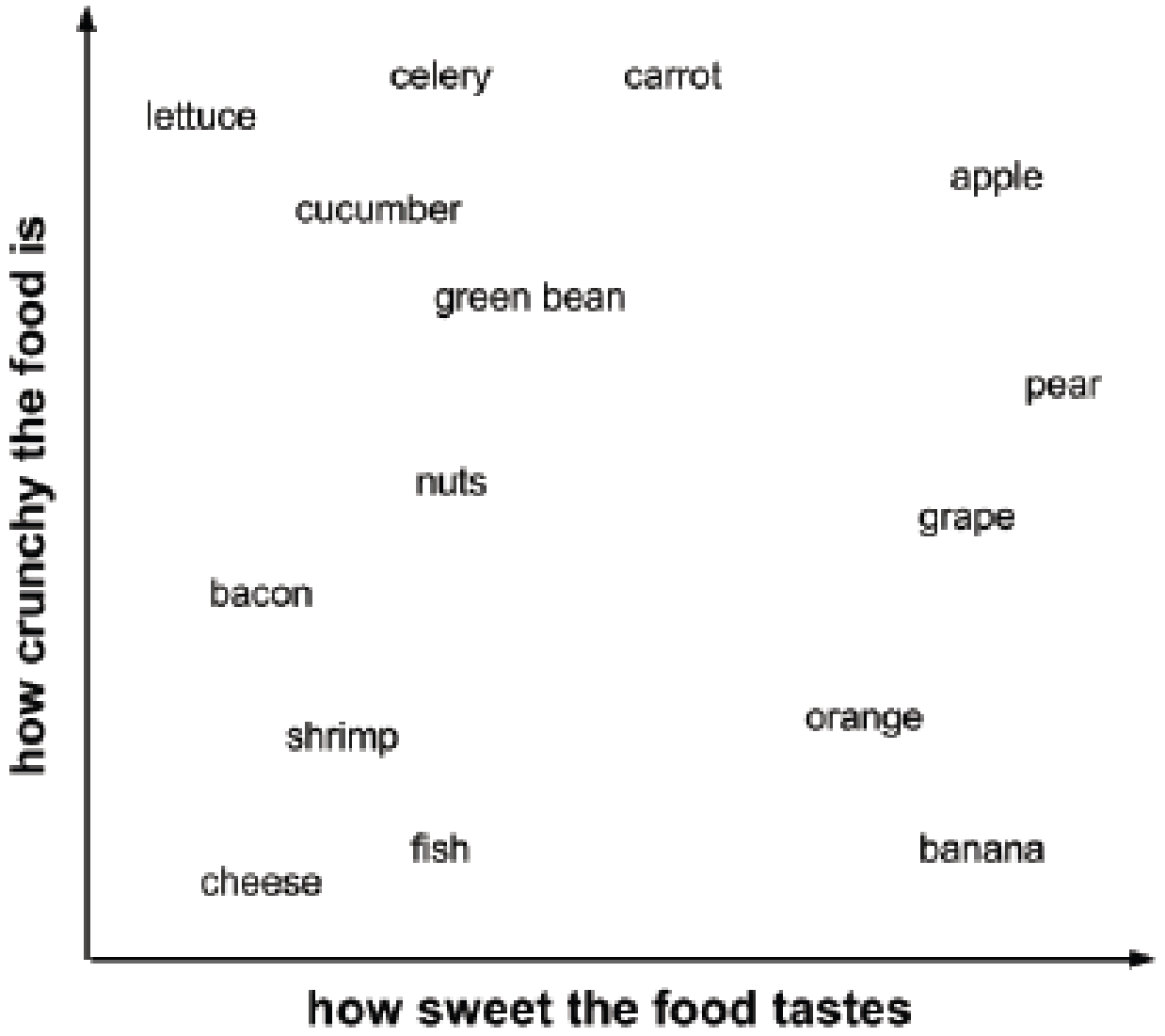
kNN 알고리즘

장점	단점
• 단순하며 효율적이다 • 데이터 분산에 대한 추정을 만들 필요가 없다 • 빠른 훈련 단계	• 모델을 생성하지 않는다. 즉, 속성 사이에 관계에서 기발한 통찰력을 발견하는 능력이 제한된다 • 느린 분류 단계 • 많은 메모리 필요 • 명목형 속성과 결측 데이터는 추가적인 처리가 필요하다

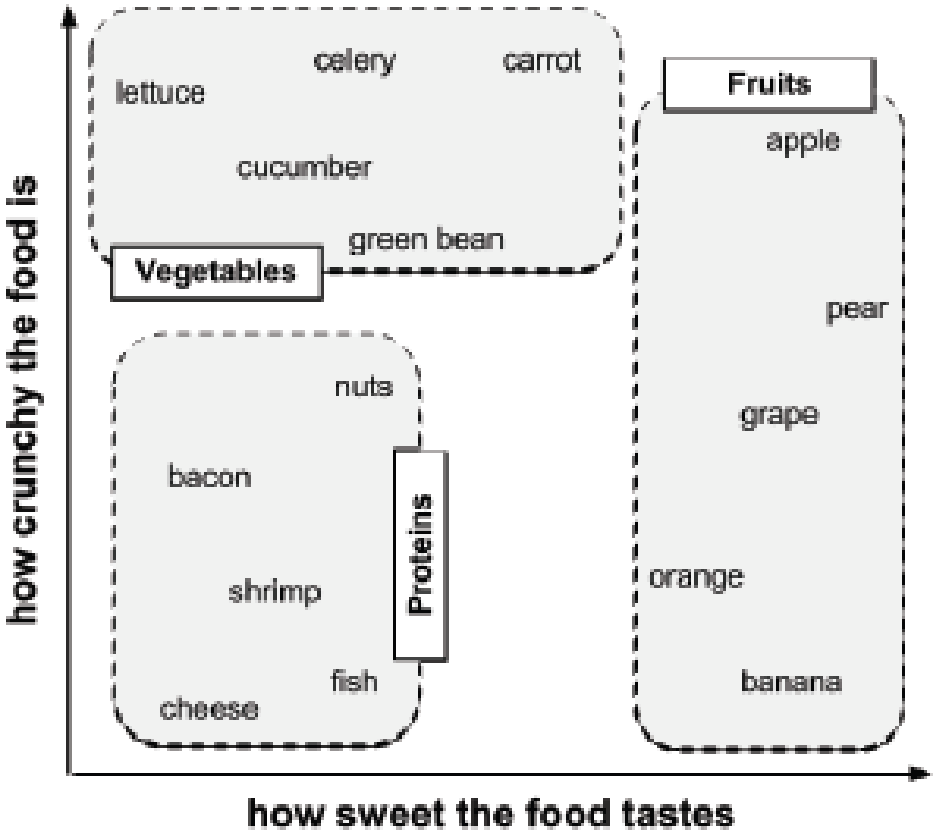
kNN 알고리즘

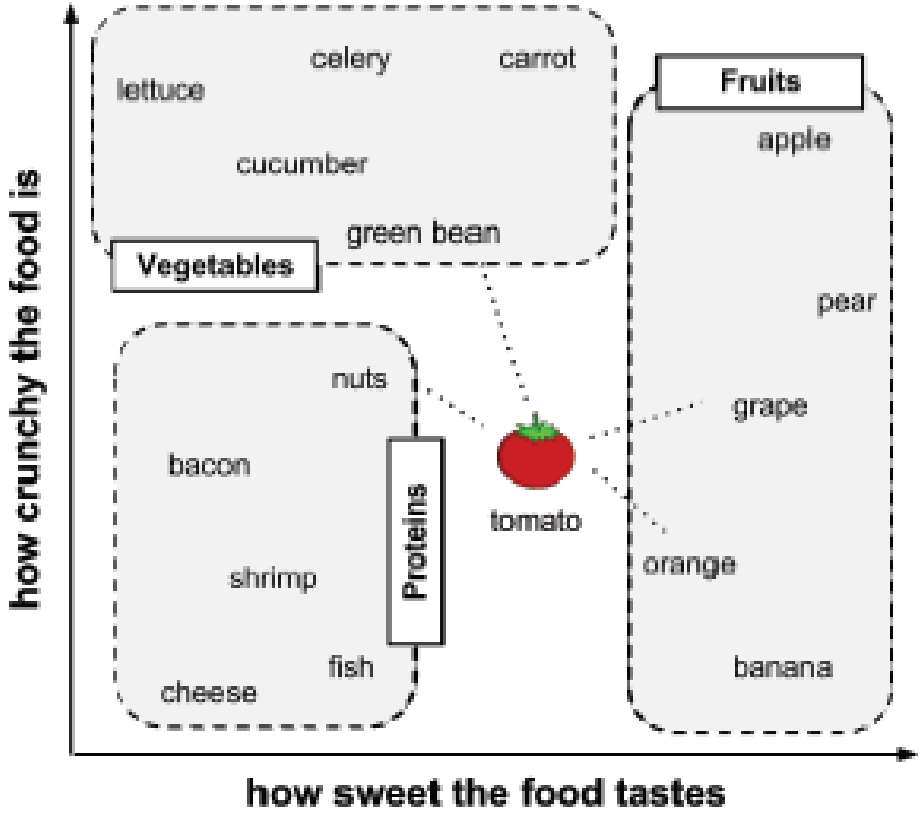
재료	단맛	아삭거림의 정도	음식 종류
Apple	10	9	Fruits
Bacon	1	4	Proteins
Banana	10	1	Fruits
Carrot	7	10	Vegetables
Celery	3	10	Vegetables
cheese	1	1	Proteins

kNN 알고리즘


celery
carrot
lettuce
apple
cucumber
green bean
pear
nuts
grape
bacon
orange
shrimp
fish
banana
cheese
how crunchy the food is
how sweet the food tastes

kNN 알고리즘


how crunchy the food is
how sweet the food tastes
celery
carrot
lettuce
cucumber
green bean
Vegetables
Fruits
apple
pear
grape
orange
banana
nuts
bacon
shrimp
fish
cheese
Proteins


how crunchy the food is
how sweet the food tastes
celery
carrot
lettuce
cucumber
green bean
Vegetables
Fruits
apple
pear
grape
orange
banana
nuts
bacon
shrimp
fish
cheese
Proteins
tomato

거리 계산

$$\text{dist}(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

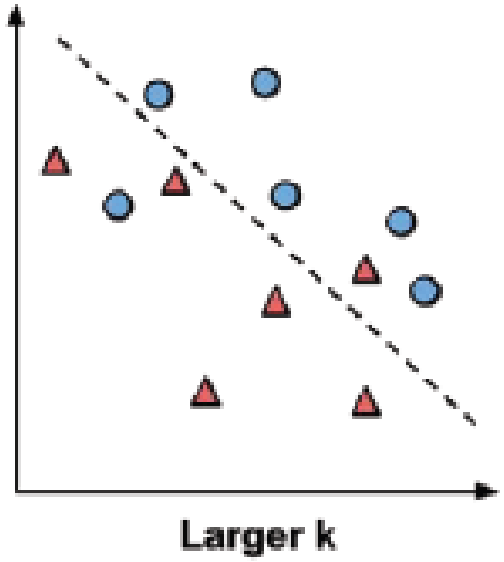
$$\text{dist}(tomato, green\ bean) = \sqrt{(6-3)^2 + (4-7)^2} = 4.2$$

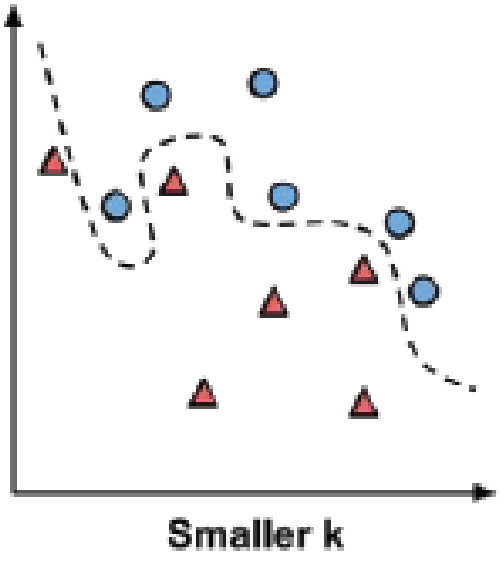
ingredient	sweetness	crunchiness	food type	distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2*
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*

- 토마토는 1NN 일 경우 과일, 3NN 일 경우 과일로 분류된다
- 1NN (k=1) : 오렌지(1.4) => 과일
- 3NN (k=3) : 오렌지(1.4), 포도(2.2), 땅콩(3.6) => 과일(2):단백질(1) => 과일

적당한 k 선택

- Larger K
  - 모든 데이터가 다수결에 참여
  - 가장 가까운 이웃에 상관없이 다수의 범주를 항상 예측
- Smaller K
  - 노이즈 데이터나 이상치에 영향을 받는다
- 데이터 개수의 제곱근을 일반적으로 사용
  - 15개 =〉 4개 ($\sqrt{15} = 3.87$)





kNN을 사용하기 위한 데이터 준비

- 최소-최대 정규화(min-max normalization)

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

- Z 점수 표준화(z-score standardization)

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

- 더미 코딩(dummy coding)

$$\text{male} = \begin{cases} 1 & \text{if } x = \text{male} \\ 0 & \text{otherwise} \end{cases}$$

- n 범주 명목 속성
  - 3가지 범주 온도 변수일 경우 2(=3-1) 속성으로 구성

$$\text{hot} = \begin{cases} 1 & \text{if } x = \text{hot} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{medium} = \begin{cases} 1 & \text{if } x = \text{medium} \\ 0 & \text{otherwise} \end{cases}$$

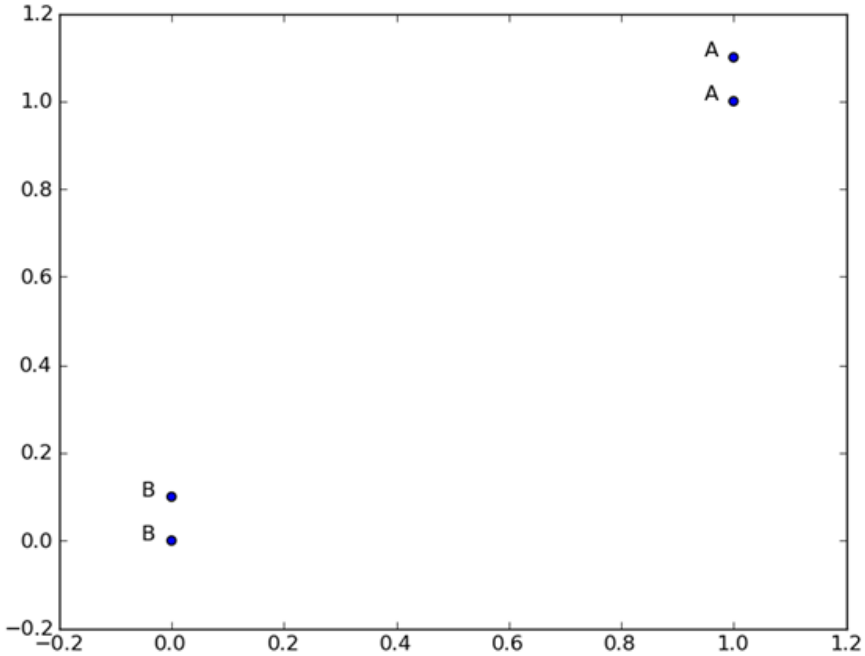
kNN 알고리즘 : 게으른 학습 알고리즘

- 게으른 학습 알고리즘
  - 인스턴스 기반 학습(instance-based learning)
  - 암기 학습(rote learning)

- 인스턴스 기반 학습기
  - 모델을 생성하지 않는다
  - 모수가 없는 비모수(non-parametic) 학습 기법

Python Example

```
>>> group = array([[1.0,1.1],[1.0,1.0],[0,0],[0,0.1]])
>>> labels = ['A','A','B','B']
```

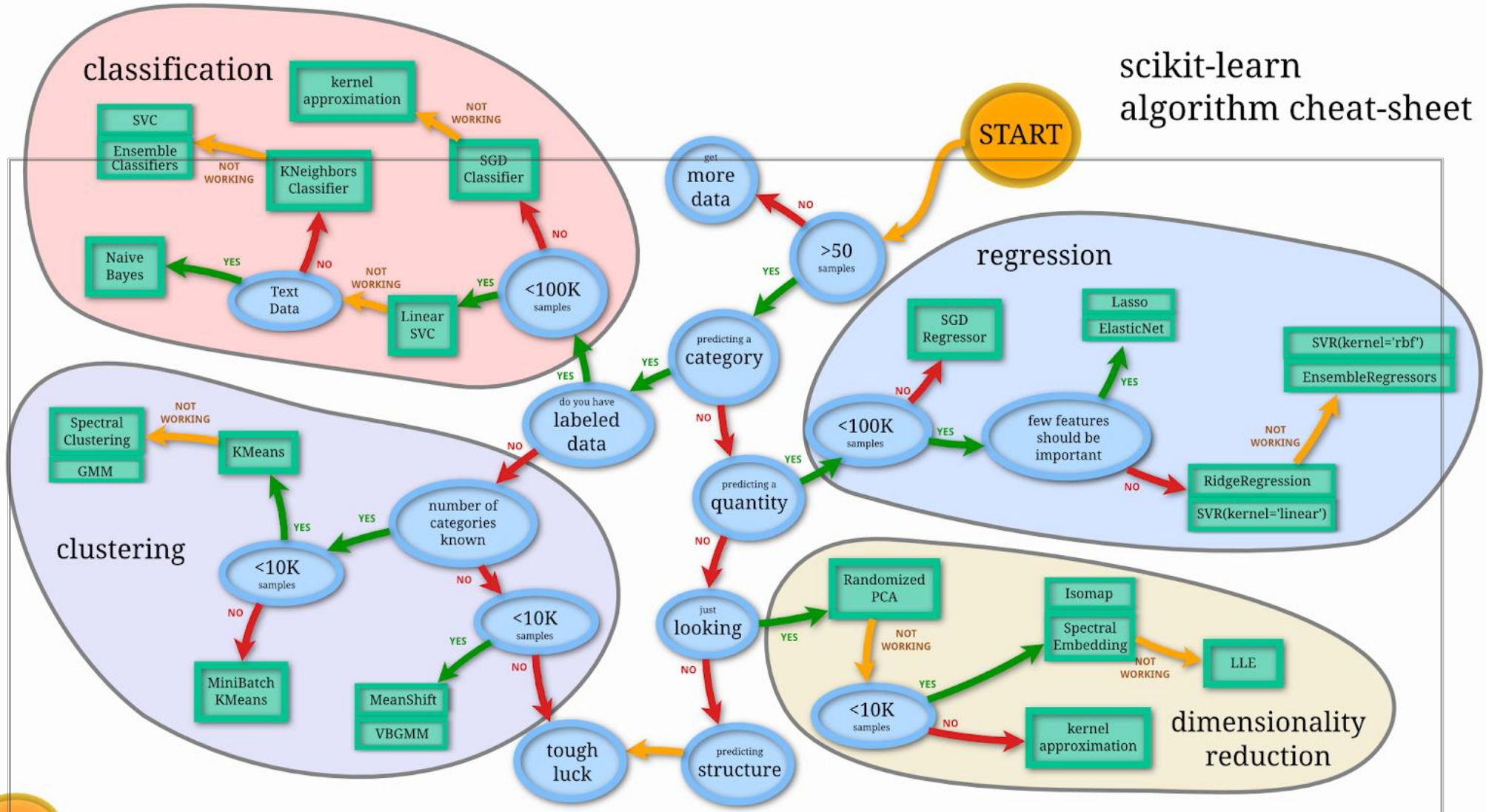



```
>>> classify0([0,0], group, labels, 3)
B
```

분류기를 시험하는 방법

- 우리가 만든 분류기를 테스트하기 위해서는 분류항목이 알려진 데이터셋을 이용하여 에러율을 측정한다.
- 에러율이 0이면 완벽한 분류기이고 1이면 항상 틀린 분류를 한다는 의미이다.

[참고] sklearn 이용 방법: 알고리즘 선택 기준



[참고] sklearn 이용 방법: 필요한 추가모듈 설치

- pip install scikit-learn
- pip install mglearn

[참고] sklearn 이용 방법: KNeighborsClassifier 이용 방법

```
from sklearn.model_selection import train_test_split
X, y = mglearn.datasets.make_forge()
X_train, X_test, y_train, y_test = train_test_split(
                                        X, y, random_state=0)
                                                # 트레인셋/테스트셋 분리
from sklearn.neighbors import KNeighborsClassifier
clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train, y_train)                       # kNN 모델생성
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=3, p=2,
           weights='uniform')
```

```
print("테스트 세트 예측: {}".format(clf.predict(X_test)))   # 분류 예측
```

```
테스트 세트 예측: [1 0 1 0 1 0 0]
```

```
print("테스트 세트 정확도: {:.2f}".format(clf.score(X_test, y_test)))
                                                # 정확도 계산
```

```
테스트 세트 정확도: 0.86
```

kNN 요약

- kNN 알고리즘은 간단하고 데이터를 분류하는 데 효과적인 방법이다.
- kNN은 사례기반 학습(instance-based learning)의 한 예이며, 기계 학습 알고리즘을 수행하기 위해서는 우리 주변에서 다루기 쉬운 데이터 사례가 있어야만 한다.
- 이 알고리즘은 데이터 집합 전체를 다루므로 큰 데이터 집합을 처리하기 위해서는 큰 저장소가 필요하다.
- 데이터 집합 내의 모든 데이터에 대해 거리 측정을 계산해야 하므로 데이터의 크기가 커지면 사용하기 힘들다.
- 데이터 구조에 대한 어떠한 정보도 주지 못한다. 즉, 유사해 보이는 각 항목으로부터 '평균 ' 이나 '모범적 사례 ' 가 어떠한 것인지 알 수 없다.