근주자적 근묵자흑(近朱者赤 近墨者黑)

K-Nearest Neighbors Algorithm

2020학년도 2학기

K-NN 알고리즘 개요

- 배경
  - 아는 정보를 바탕으로 새로운 정보를 평가해야 할 때, 새로운 정보의 속성과 가장 유사한 기존의(아는) 정보를 바탕으로 평가(분류 또는 예측)하자는 아이디어.
  - T. Cover and P. Hart (1967), "Nearest Neighbor Pattern Classification", IEEE Transactions on Information Theory.
- 종류
  - K-NN Classification: 분류 문제 (예: 관측된 발사체는 방사포인가, 탄도미사일인가?)
  - K-NN Regression: 예측 문제 (예: 북한 자주포의 제원은 어떠한가?)

분류 문제 (예: 관측된 발사체는 방사포인가, 탄도미사일인가?)

한미, 北방사포를 탄도미사일로 오판했나...'대북정보력' 논란(종합2보)

송고시간 | 2019-08-01 18:00

김귀근 기자

北 "신형 대구경조종방사포 시험" 발표후 사진공개...軍 "탄도미사일 비행특성"
북한 방사포 전력 갈수록 진화...400㎜급 추정

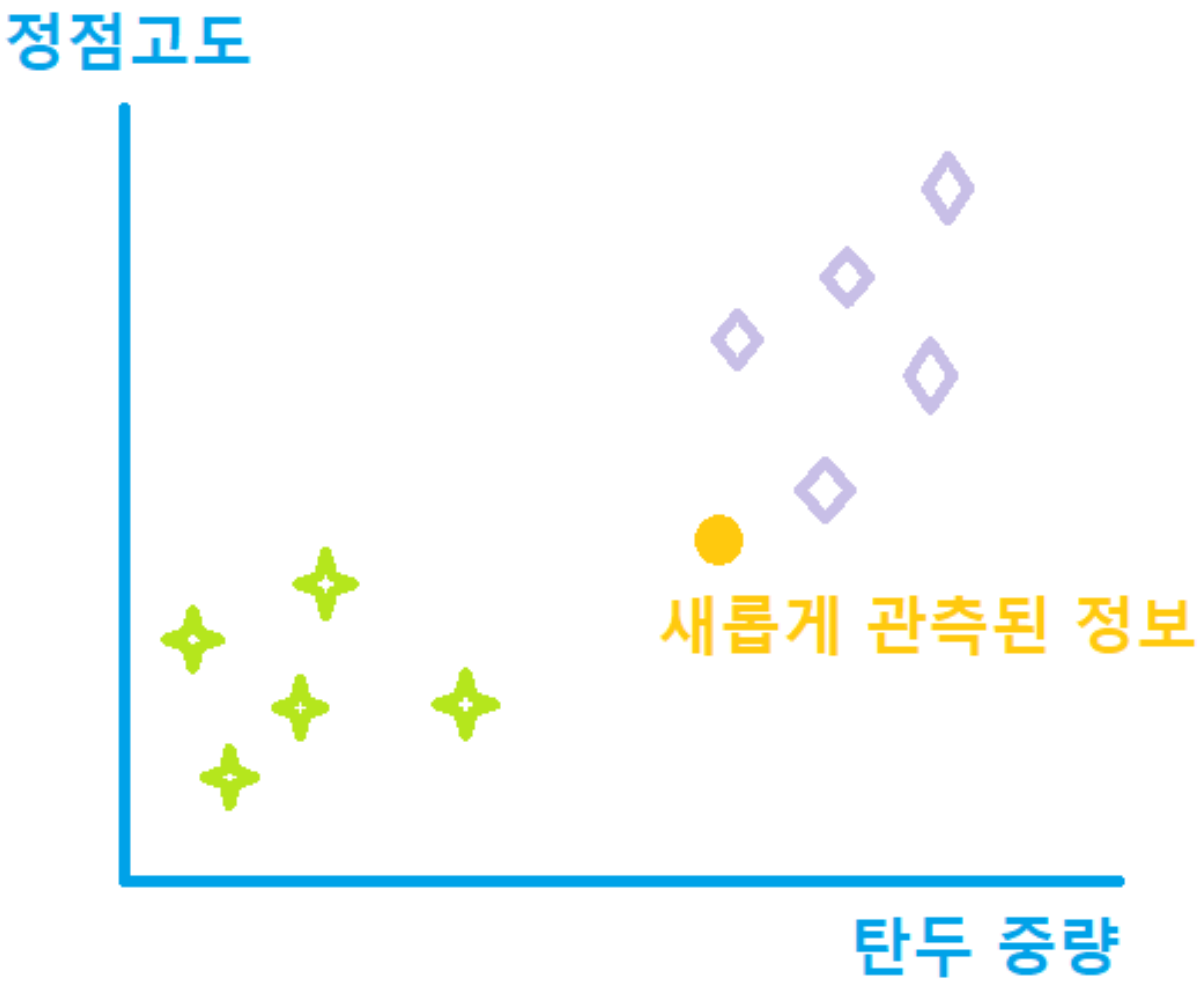
북한 300㎜ 신형 방사포
[연합뉴스 자료사진]

분류 문제 (가상의 데이터)

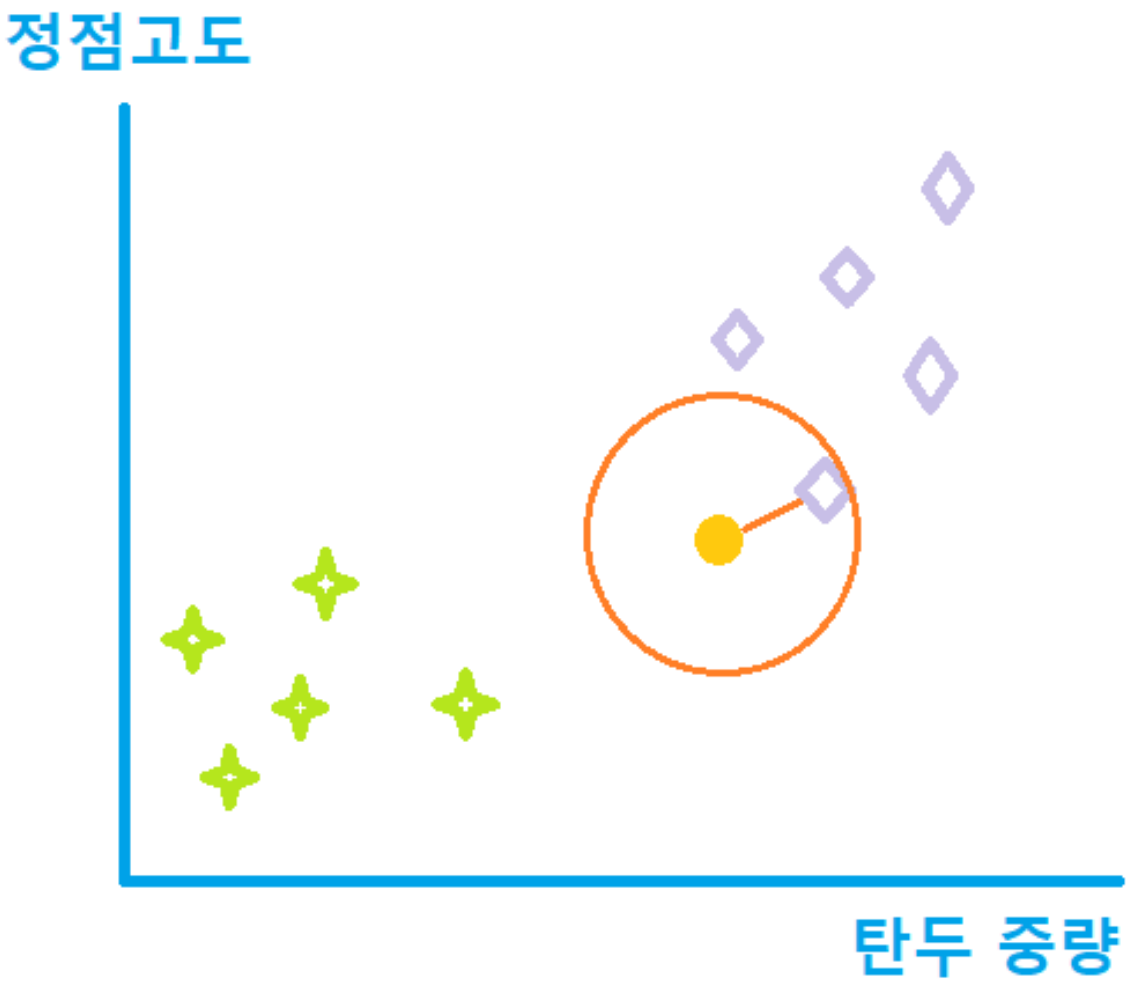
구분	정점고도(km)	탄두 중량(kg)	평가
1	60	800	탄도미사일
2	48	700	탄도미사일
3	20	120	방사포
4	30	150	방사포
5	25	200	방사포
6	37	1010	탄도미사일
7	48	970	탄도미사일
8	30	710	탄도미사일
9	20	400	방사포
10	25	600	방사포
11	27	650	?

“1”-NN 분류 알고리즘 ($k = 1$)

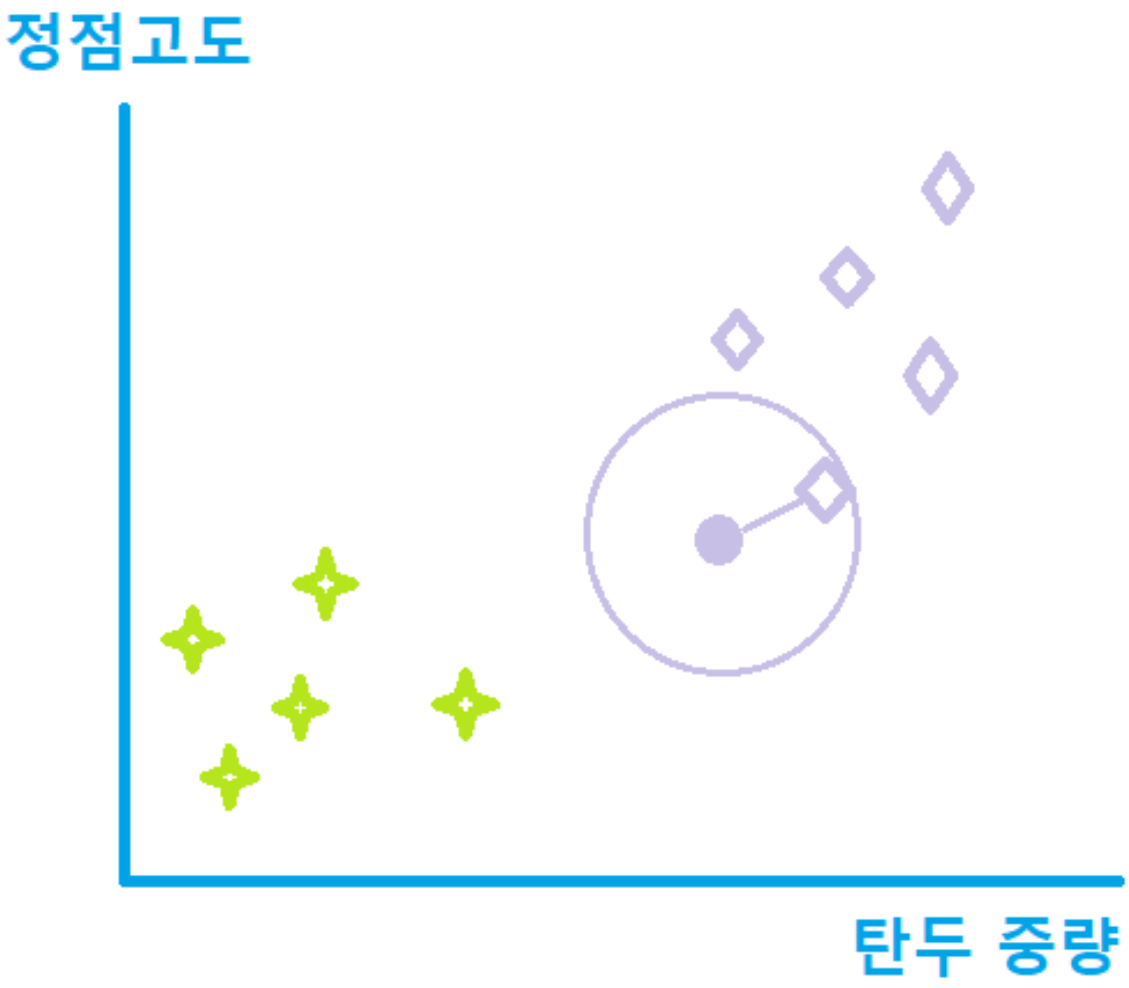
“1”-NN 분류 알고리즘 ($k = 1$)



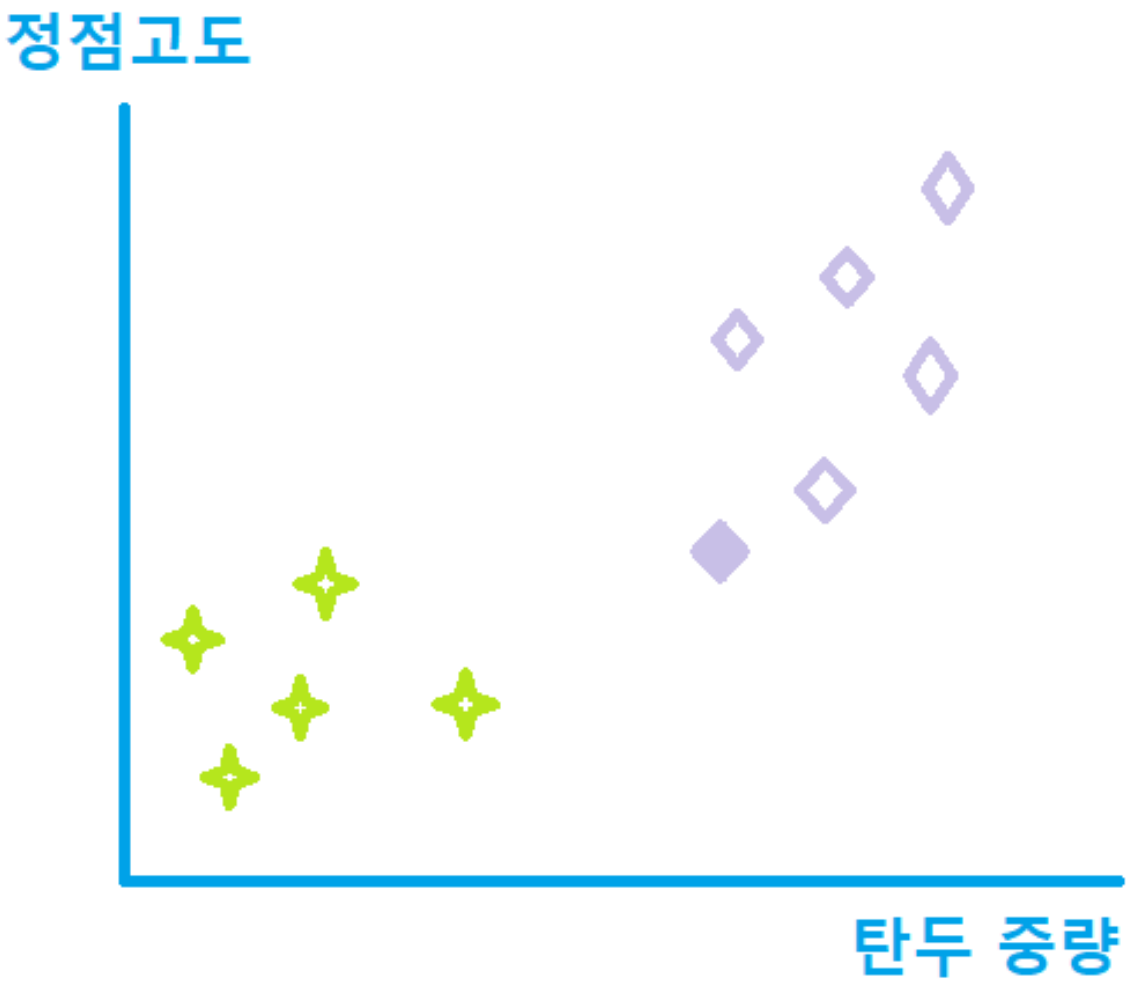
“1”-NN 분류 알고리즘 ($k = 1$)



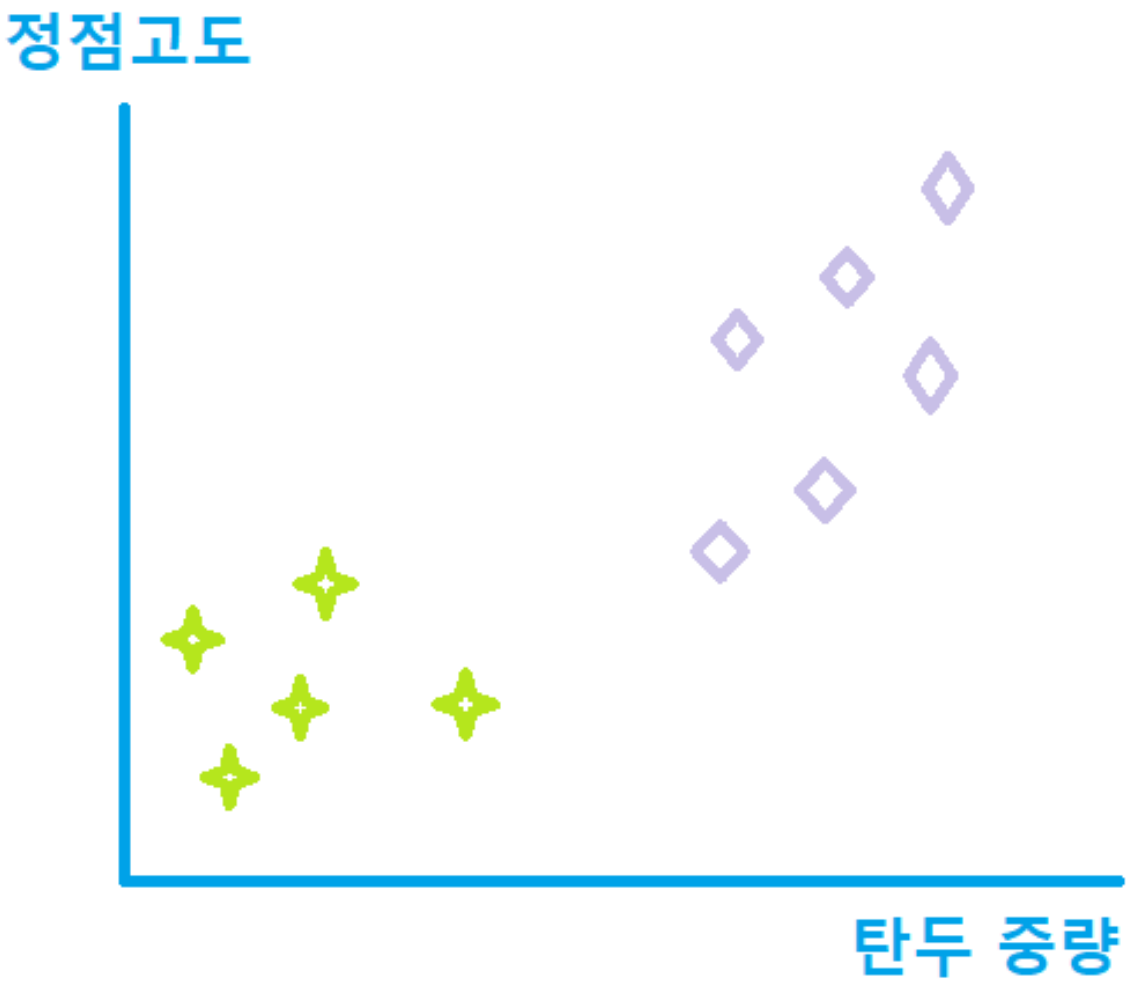
“1”-NN 분류 알고리즘 ($k = 1$)



“1”-NN 분류 알고리즘 ($k = 1$)



"1"-NN 분류 알고리즘 ($k = 1$)

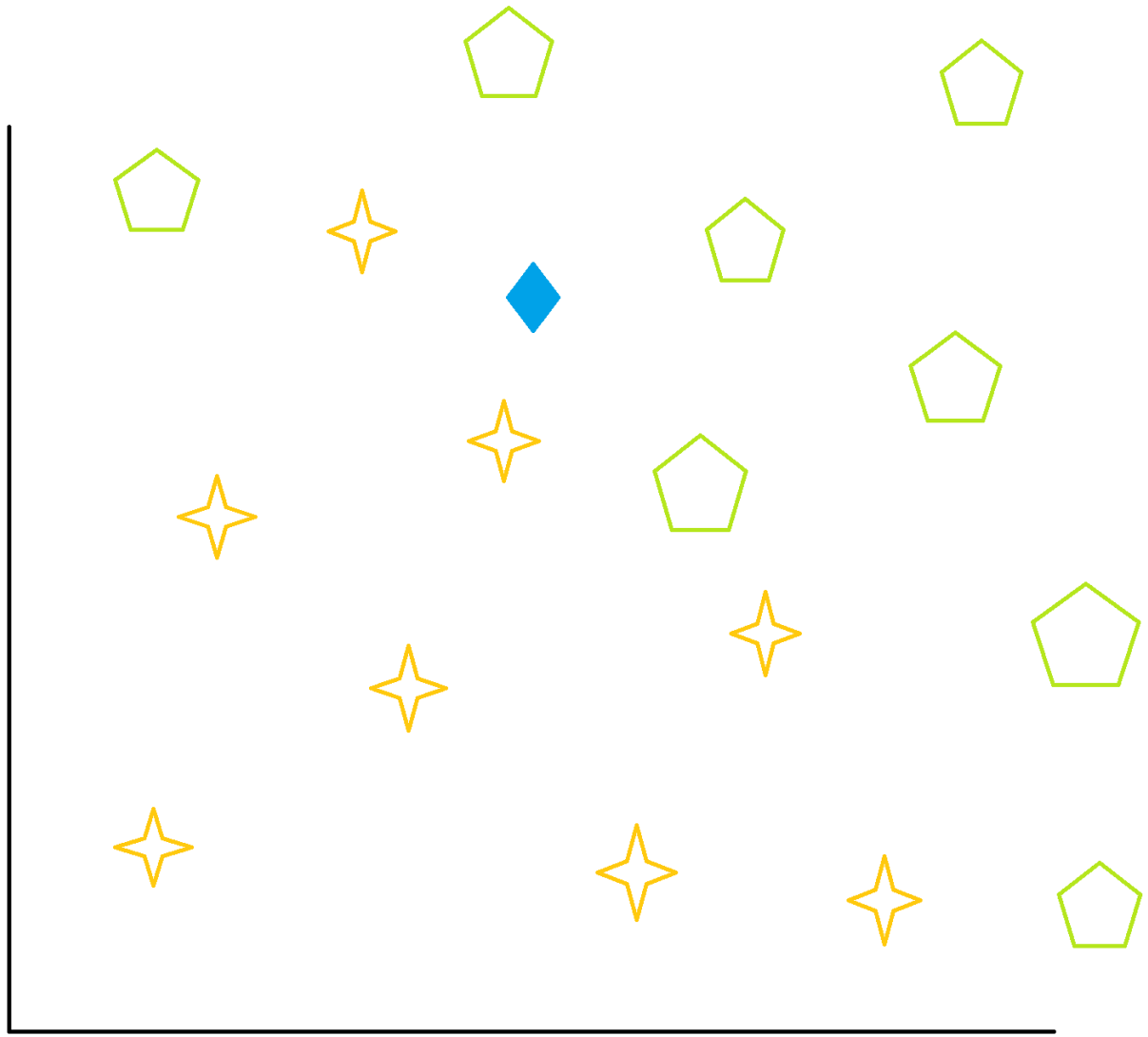


분류 문제 (가상의 데이터)

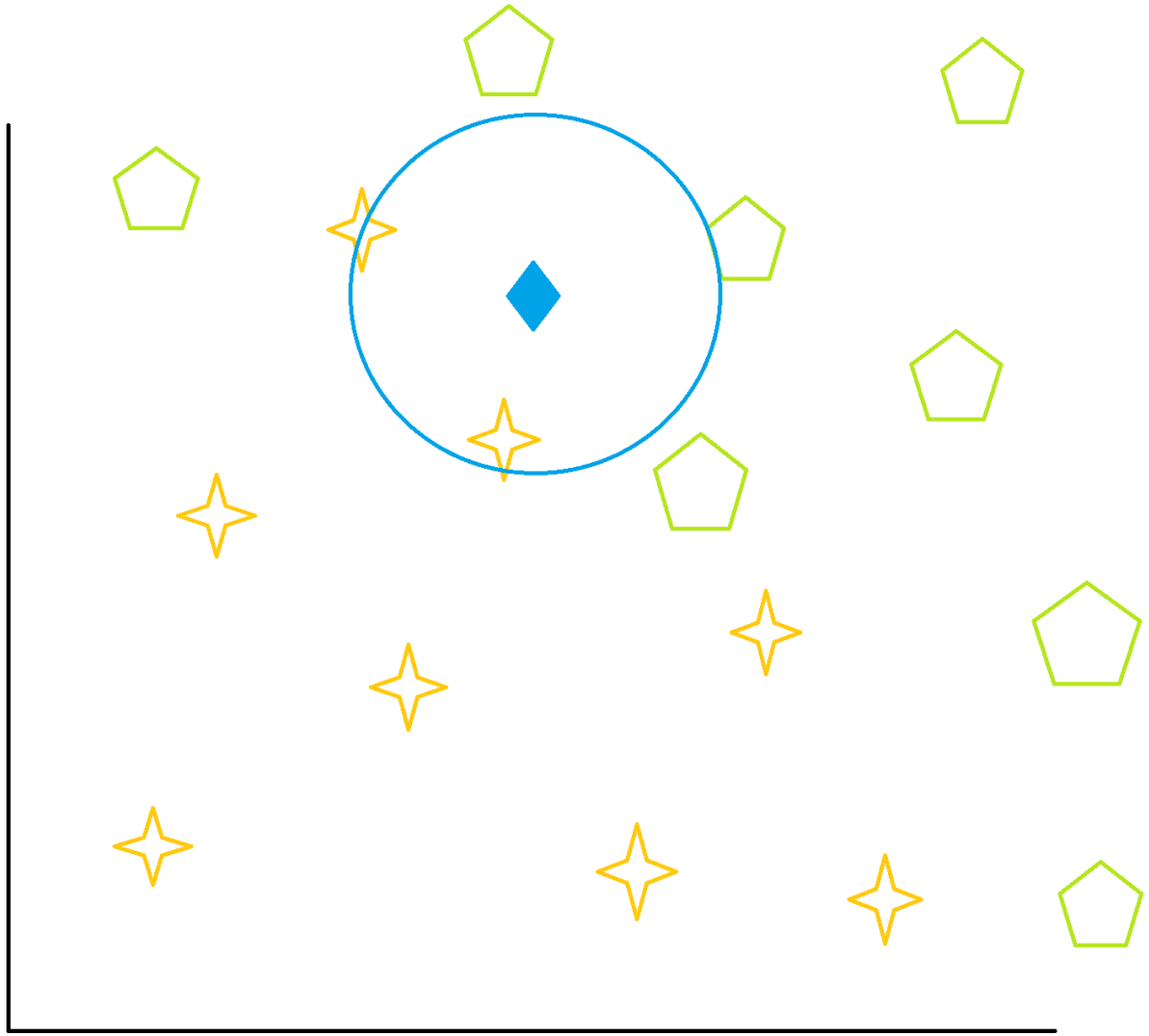
구분	정점고도(km)	탄두 중량(kg)	평가
1	60	800	탄도미사일
2	48	700	탄도미사일
3	20	120	방사포
4	30	150	방사포
5	25	200	방사포
6	37	1010	탄도미사일
7	48	970	탄도미사일
8	30	710	탄도미사일
9	20	400	방사포
10	25	600	방사포
11	27	650	탄도미사일

“3”-NN 분류 알고리즘 ($k = 3$)

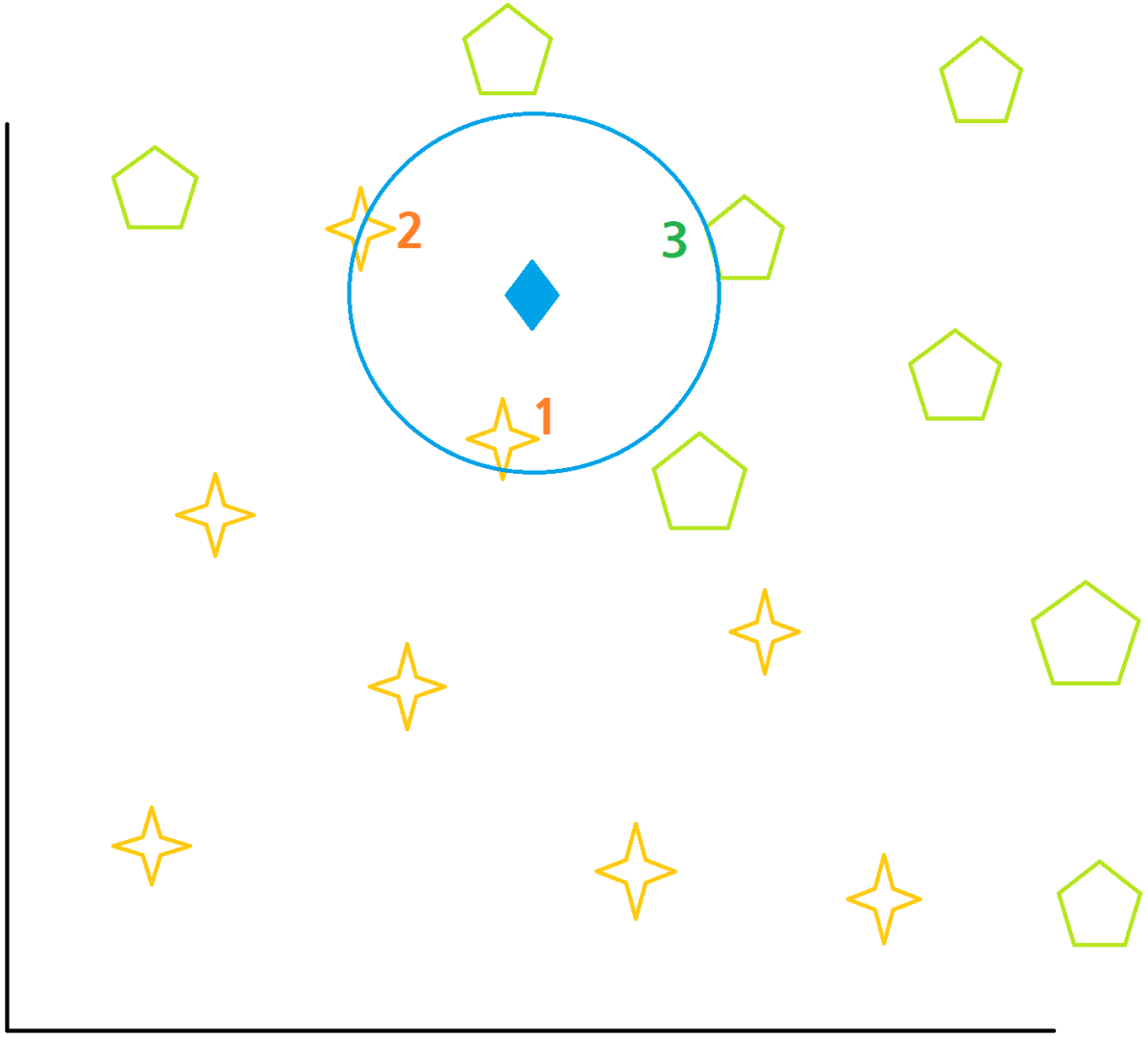
“3”-NN 분류 알고리즘 ($k = 3$)



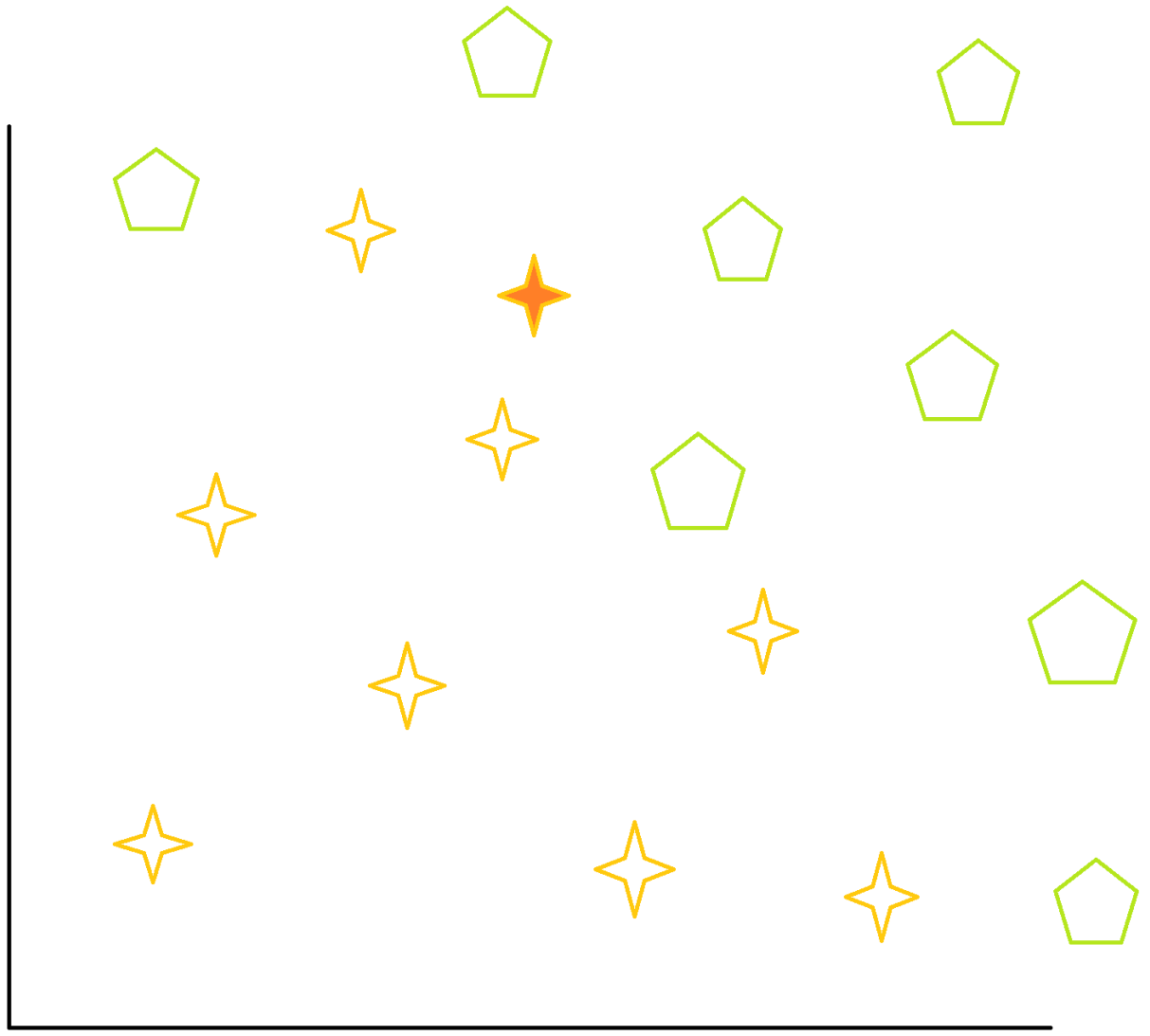
“3”-NN 분류 알고리즘 ($k = 3$)



“3”-NN 분류 알고리즘 ($k = 3$)



“3”-NN 분류 알고리즘 ($k = 3$)

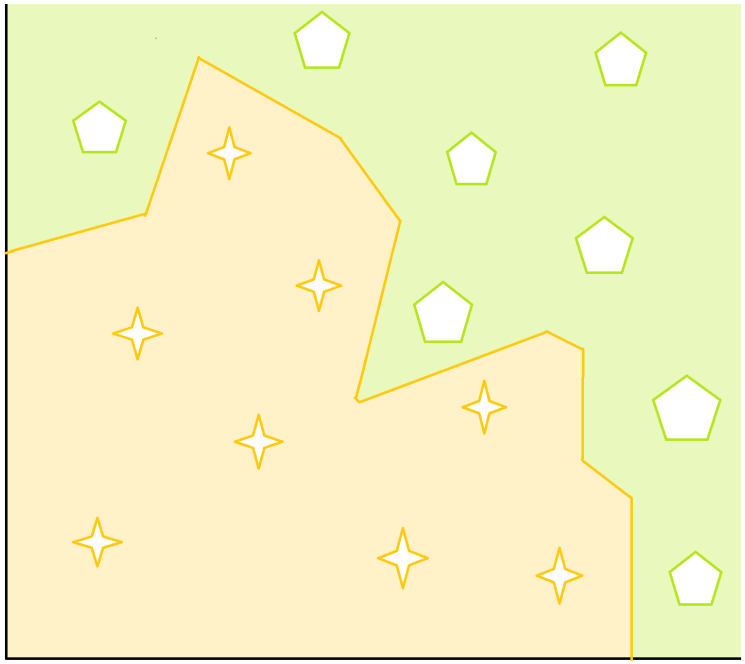


퀴즈!

- $k = 1$일때 평면 위의 가능한 모든 점에 대해서 분류해보기

Decision Boundary

- $k = 1$일때 Decision Boundary



자율 과제

- $k = 3$일때 Decision Boundary 그려보기

K-NN 알고리즘

- 종류
  - K-NN Classification: 분류 문제 (예: 관측된 발사체는 방사포인가, 탄도미사일인가?)
  - **K-NN Regression: 예측 문제 (예: 북한 자주포의 제원은 어떠한가?)**

예측 문제 (예: 북한 자주포의 제원은 어떠한가?)

북한 포 종류 및 제원

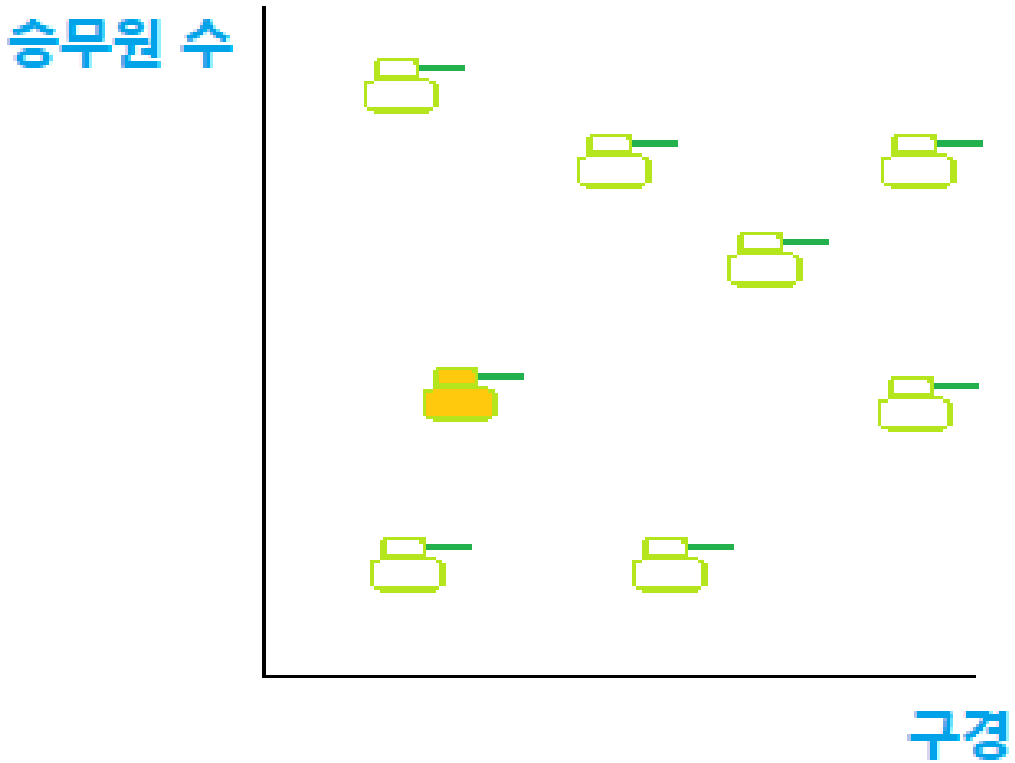
구분	종류	최대사거리(m)	승무원(명)	발사속도(발/분)
자주포	122mm 자주포(M-1977)	15	6	2.5
	122mm 자주포(M-1981)	24	7	5
	122mm 자주포(M-1991)	24	5	–
	130mm 자주포(M-1985)	28	8	3
	130mm 자주포(M-1992)	27	5	6
	152mm 자주포(M-1974)	17	5	2
	170mm 자주포(M-1978)	40	6	1/2
	170mm 자주포(M-1989)	36	6	1

122mm M-1991

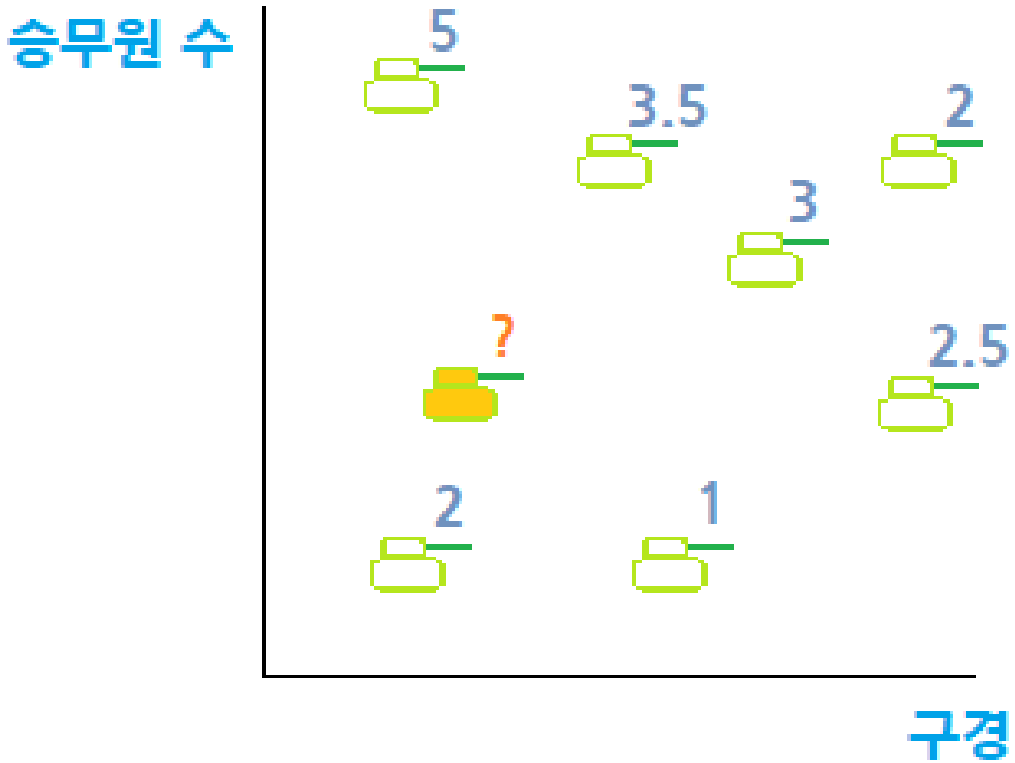
170mm M-1989

연합뉴스

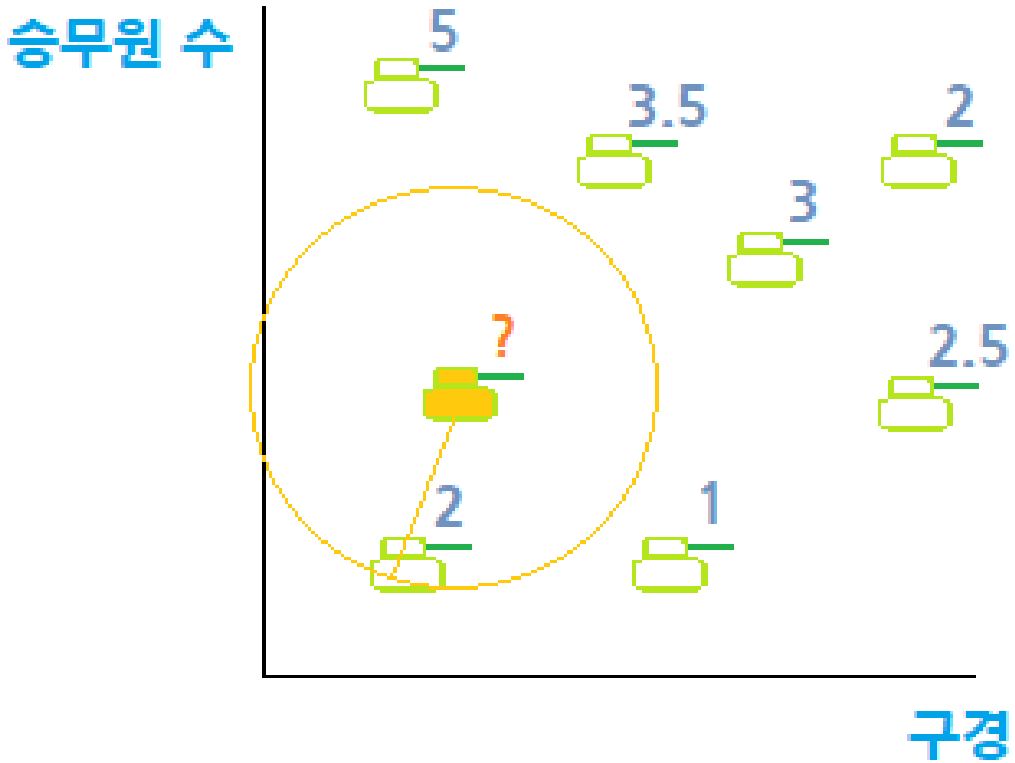
예측 문제 (예: 북한 자주포의 제원은 어떠한가?)



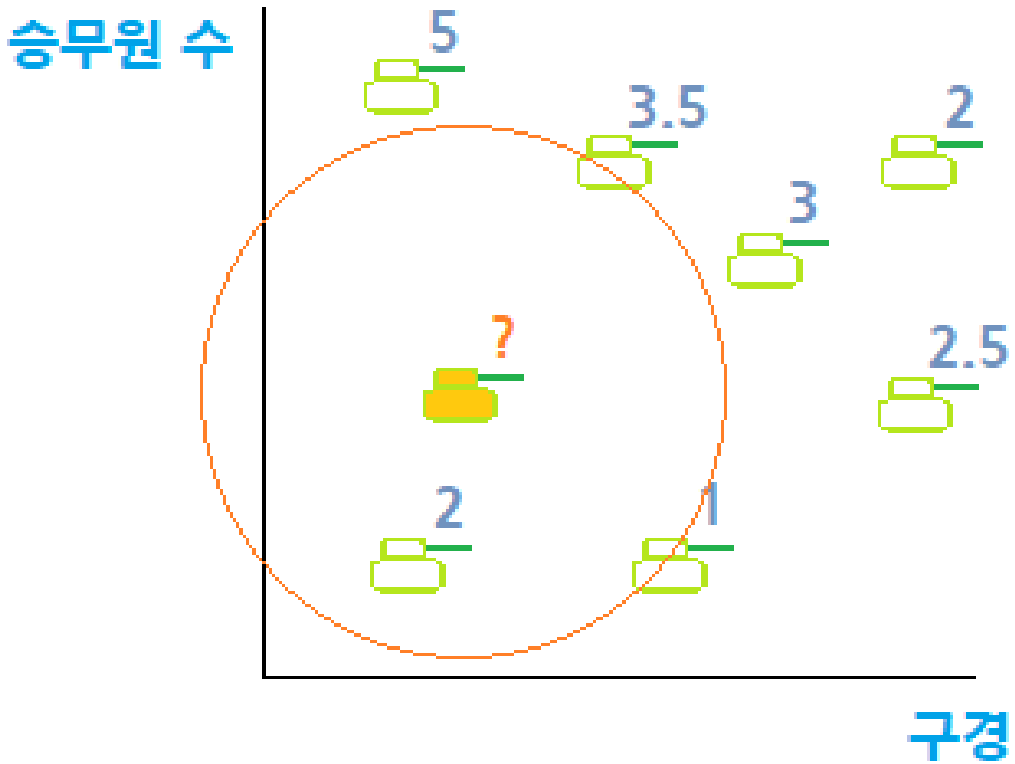
예측 문제 (예: 북한 자주포의 제원은 어떠한가?)



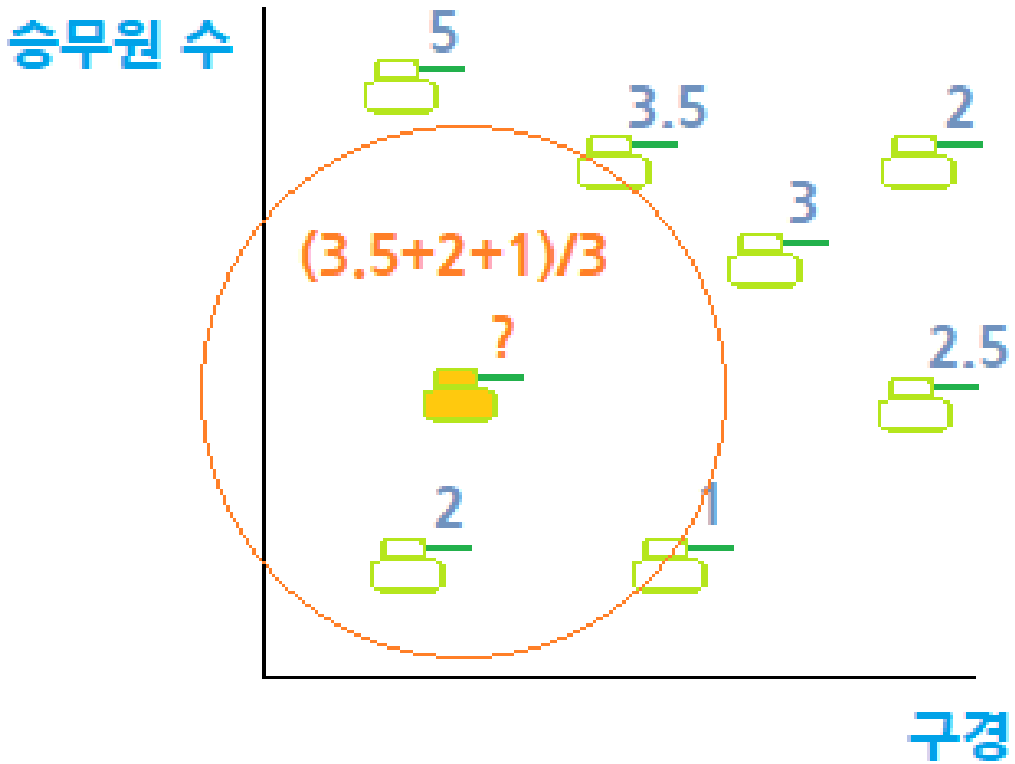
“1”-NN 예측 (가상의 데이터)



“3” -NN 예측 (가상의 데이터)



“3” -NN 예측 (가상의 데이터)



K-NN 알고리즘 개요 (Reprise)

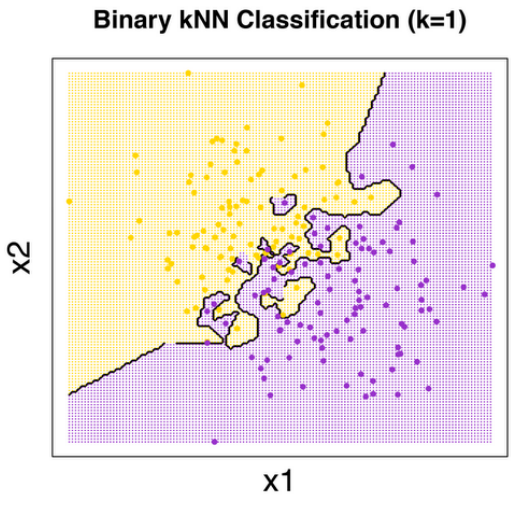
- 배경
  - 아는 정보를 바탕으로 새로운 정보를 평가해야 할 때, 새로운 정보의 속성과 가장 유사한 기존의(아는) 정보를 바탕으로 평가(분류 또는 예측)하자는 아이디어.
  - T. Cover and P. Hart (1967), "Nearest Neighbor Pattern Classification", IEEE Transactions on Information Theory.
- 종류
  - K-NN Classification: 분류 문제 (예: 관측된 발사체는 방사포인가, 탄도미사일인가?)
  - K-NN Regression: 예측 문제 (예: 북한 자주포의 제원은 어떠한가?)

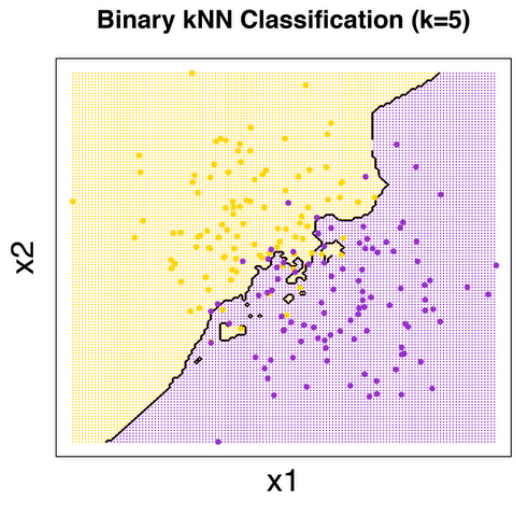
K-NN 알고리즘의 논점

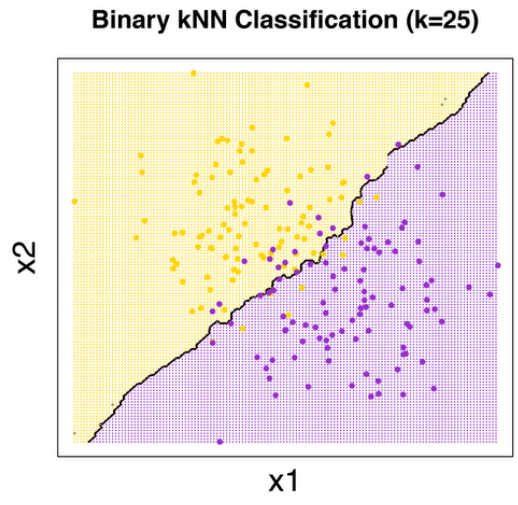
- 논점 1
  - k는 어떻게 설정할 것인가?
- 논점 2
  - 데이터 간 거리를 어떻게 측정할 것인가?
- 논점 3
  - 선택된 근접 데이터를 어떻게 반영할 것인가?

가슴엔 조국을, 두 눈은 세계로!

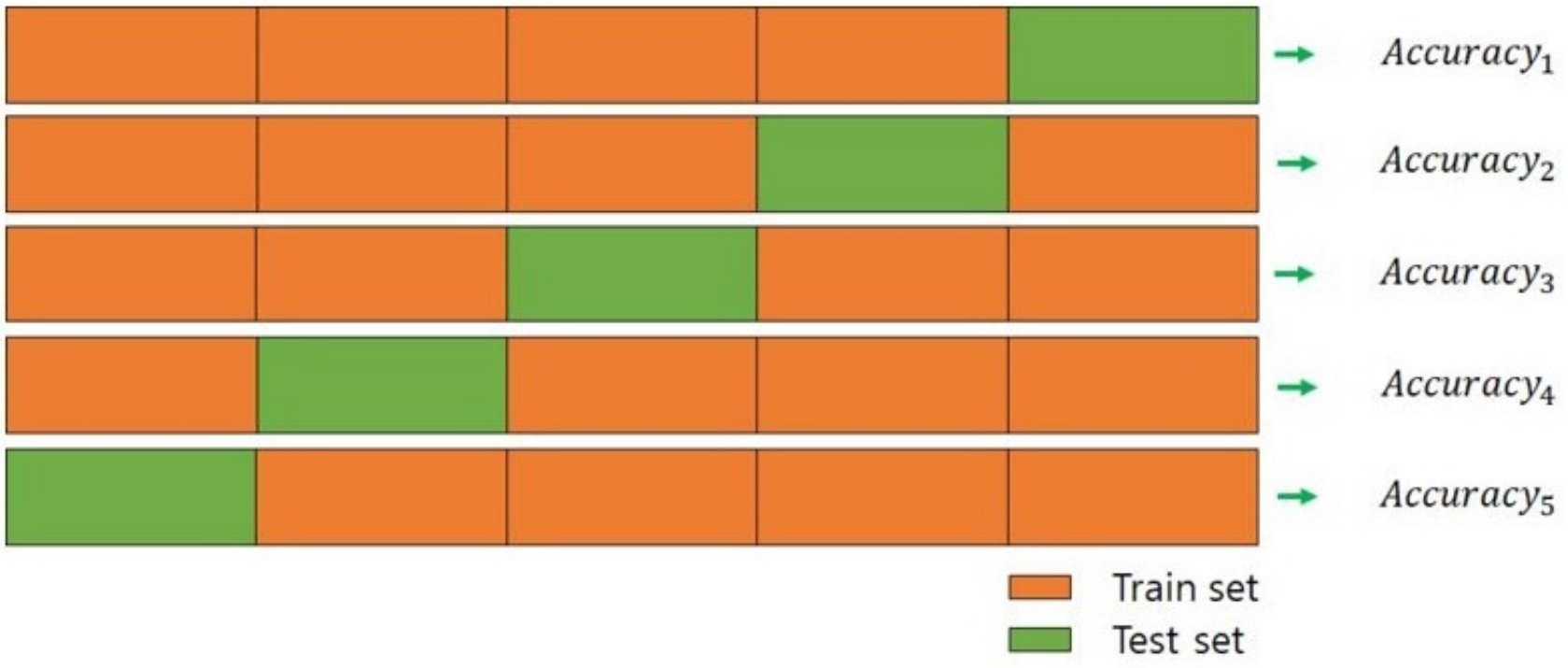
k는 어떻게 설정할 것인가?







5-fold Cross Validation



$$Accuracy = Average(Accuracy_1, \cdots, Accuracy_5)$$

k는 어떻게 설정할 것인가?

- k 선택 방법
  - k값이 1일 때, 2일 때, $\cdots$ 일 때 Accuracy를 각각 따져본다.
  - 가령, 5-fold Cross Validation으로 Accuracy를 측정할 수 있다.

    (참고로 Cross Validation의 종류에도 여러가지가 있다.)
  - 가장 Accuracy가 높은 k를 선택한다.

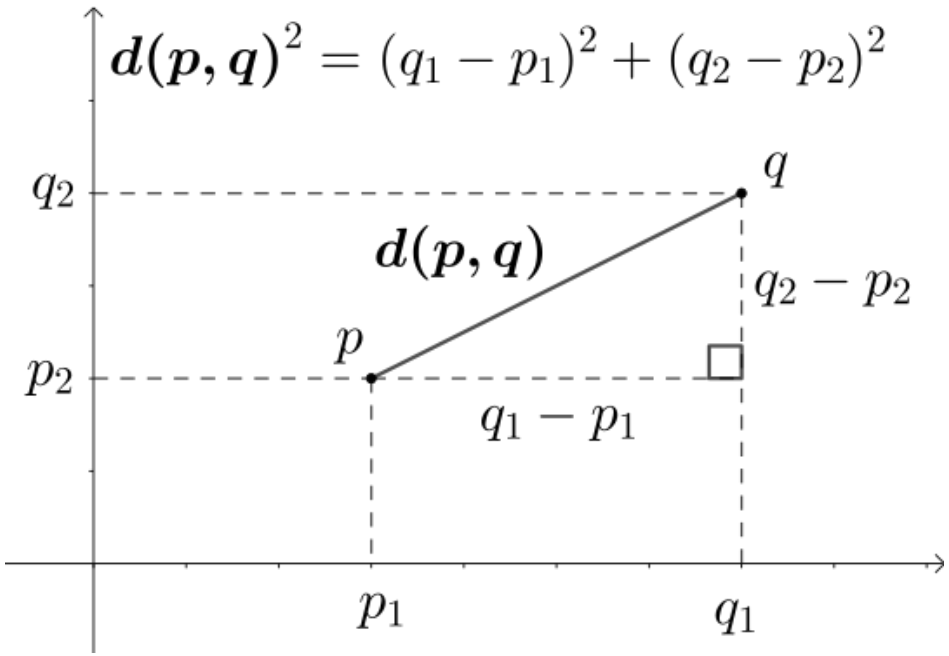
K-NN 알고리즘의 논점 (reprise)

- 논점 1
  - k는 어떻게 설정할 것인가?
- 논점 2
  - 데이터 간 거리를 어떻게 측정할 것인가?
- 논점 3
  - 선택된 근접 데이터를 어떻게 반영할 것인가?

데이터 간 거리를 어떻게 측정할 것인가?

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Hamming Distance
- Cosine Distance
- Mahalanobis Distance
- etc.

Euclidean Distance



Euclidean Distance

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 \ \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

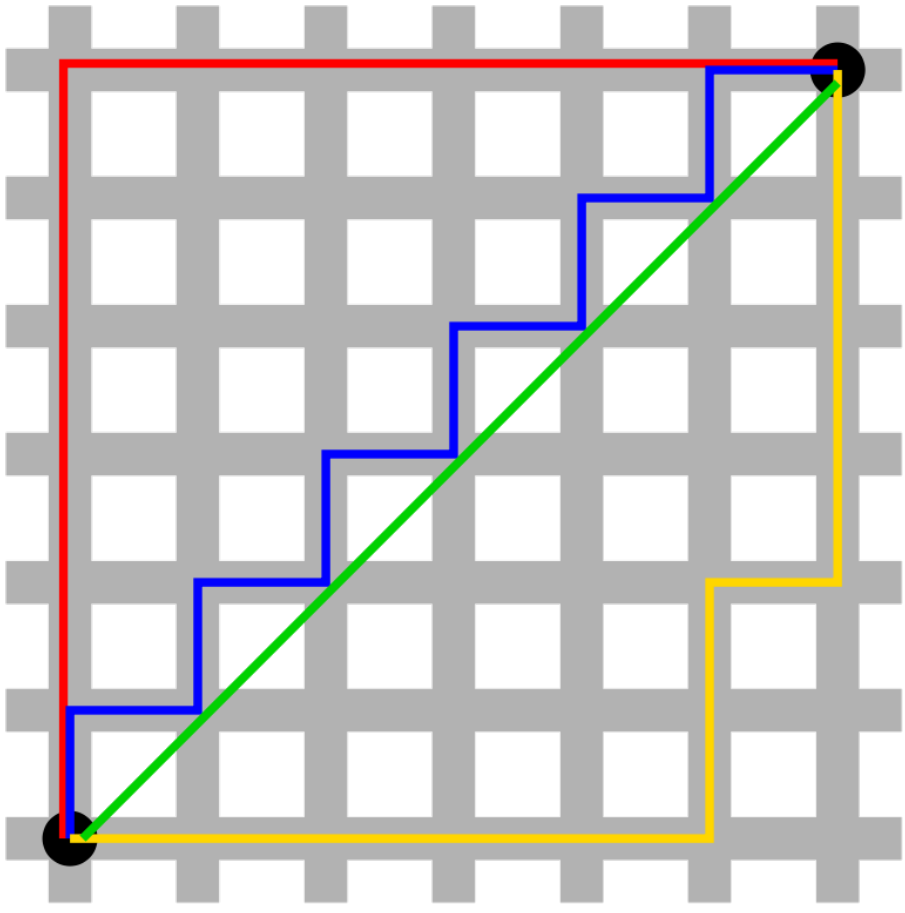
```
from scipy.spatial import distance

print(distance.euclidean([1, 2, 3], [3, 2, 1]))
print(distance.euclidean([1, 0, 0], [0, 1, 1]))
```

```
2.8284271247461903
1.7320508075688772
```

가슴엔 조국을, 두 눈은 세계로!

Manhattan Distance

Manhattan Distance



Manhattan Distance

$$d(p,q) = \|p - q\| = \sum_{i=1}^{n} \|p_i - q_i\|$$

```
from scipy.spatial import distance

print(distance.cityblock([1, 2, 3], [3, 2, 1]))
print(distance.cityblock([1, 0, 0], [0, 1, 1]))
```

```
4
3
```

Minkowski Distance

$$d(X,Y) = \left(\sum_{i=1}^{n} \|x_i - y_i\|^p\right)^{\frac{1}{p}}$$

```
from scipy.spatial import distance

print(distance.minkowski([1, 1, 1], [1, 0, 1], 1))
print(distance.minkowski([1, 1, 0], [0, 1, 1], 2))
print(distance.minkowski([1, 0, 0], [0, 1, 0], 3))
```

```
1.0
1.4142135623730951
1.2599210498948732
```

가슴엔 조국을, 두 눈은 세계로!

Hamming Distance

- '1011101'과 '1001001'사이의 해밍 거리는 2. (1011101, 1001001)
- '2143896'과 '2233796'사이의 해밍 거리는 3. (2143896, 2233796)
- "toned"와 "roses"사이의 해밍 거리는 3. (toned, roses)

```
from scipy.spatial import distance

print(distance.hamming([1, 0, 0], [1, 1, 0]))
print(distance.hamming([1, 0, 0], [1, 1, 1]))
```

```
0.3333333333333333
0.6666666666666666
```

가슴엔 조국을, 두 눈은 세계로!

Cosine Distance

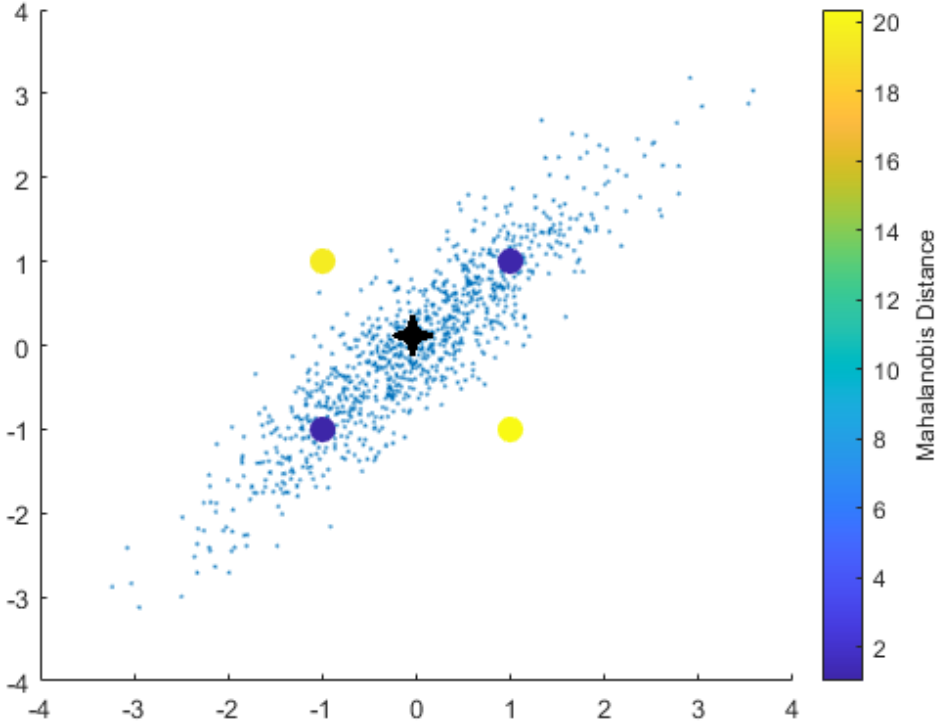
$$Similarity = \cos\theta = \frac{\mathrm{A} \cdot \mathrm{B}}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

```
from scipy.spatial import distance

print(distance.cosine([100, 0, 0], [0, 1, 0]))
print(distance.cosine([1, 1, 0], [0, 1, 0]))
```

```
1.0
0.29289321881345254
```

가슴엔 조국을, 두 눈은 세계로!

Mahalanobis Distance



Mahalanobis Distance

$$\mathbf{d}(p,q) = (p-q)\Sigma^{-1}(p-q)^T$$

Covariance Matrix: Sigma

$$\Sigma_{j,k} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij} - \overline{X}_j)(X_{ik} - \overline{X}_k)$$

scipy.spatial.distance.mahalanobis

scipy.spatial.distance.mahalanobis(*u*, *v*, *VI*)

Computes the Mahalanobis distance between two 1-D arrays.

The Mahalanobis distance between 1-D arrays *u* and *v*, is defined as

$$\sqrt{(u-v)V^{-1}(u-v)^T}$$

where V is the covariance matrix. Note that the argument *VI* is the inverse of V.

Parameters:

u : *(N,) array_like*
Input array.

v : *(N,) array_like*
Input array.

VI : *ndarray*
The inverse of the covariance matrix.

Returns:

mahalanobis : *double*
The Mahalanobis distance between vectors *u* and *v*.

가슴엔 조국을, 두 눈은 세계로!

기타

- Pearson's Correlation Distance
- Spearman's Rank Correlation Distance
- Jaccard Distance

가슴엔 조국을, 두 눈은 세계로!

K-NN 알고리즘의 논점 (reprise)

- 논점 1
  - k는 어떻게 설정할 것인가?
- 논점 2
  - 데이터 간 거리를 어떻게 측정할 것인가?
- 논점 3
  - 선택된 근접 데이터를 어떻게 반영할 것인가?

선택된 근접 데이터를 어떻게 반영할 것인가?

- 과반수 의결
- 거리를 반영하는 방식(예: $\frac{1}{d}$를 곱한 후 의결)

실습