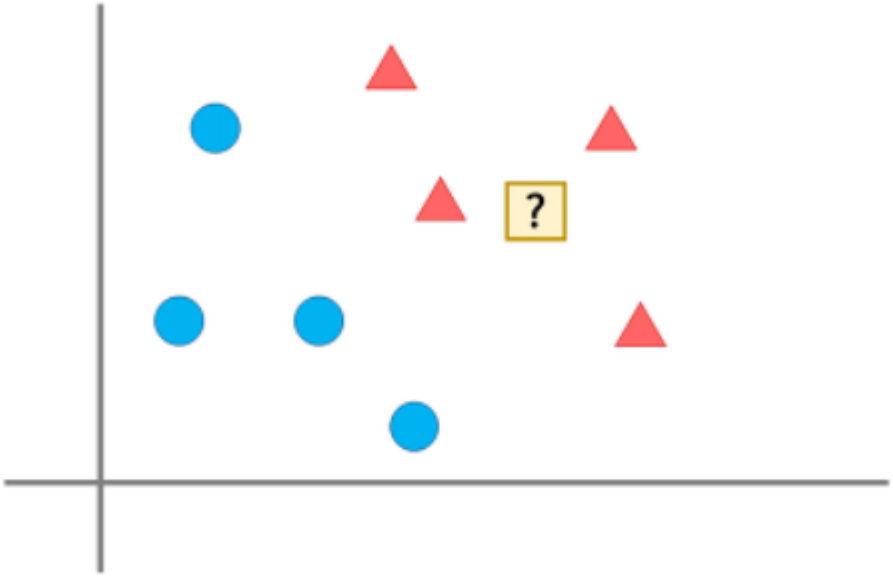
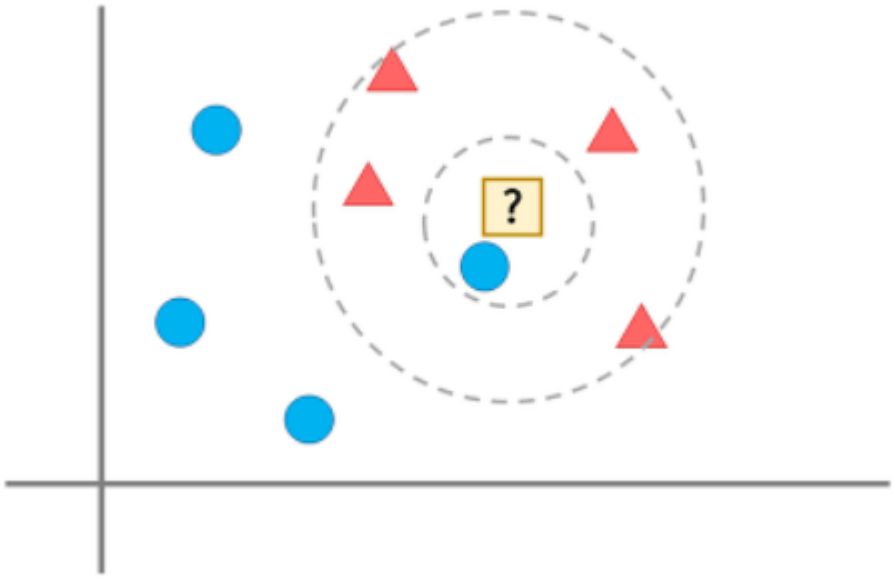
KNN(K -Nearest Neighbors)

Theory

K-최근접 이웃법은 한마디로 **유유상종** 이라고 표현 할 수 있다. 새로운 데이터를 입력 받았을 때 가장 가까이 있는 것이 무엇이냐를 중심으로 새로운 데이터의 종류를 정해주는 알고리즘이다.



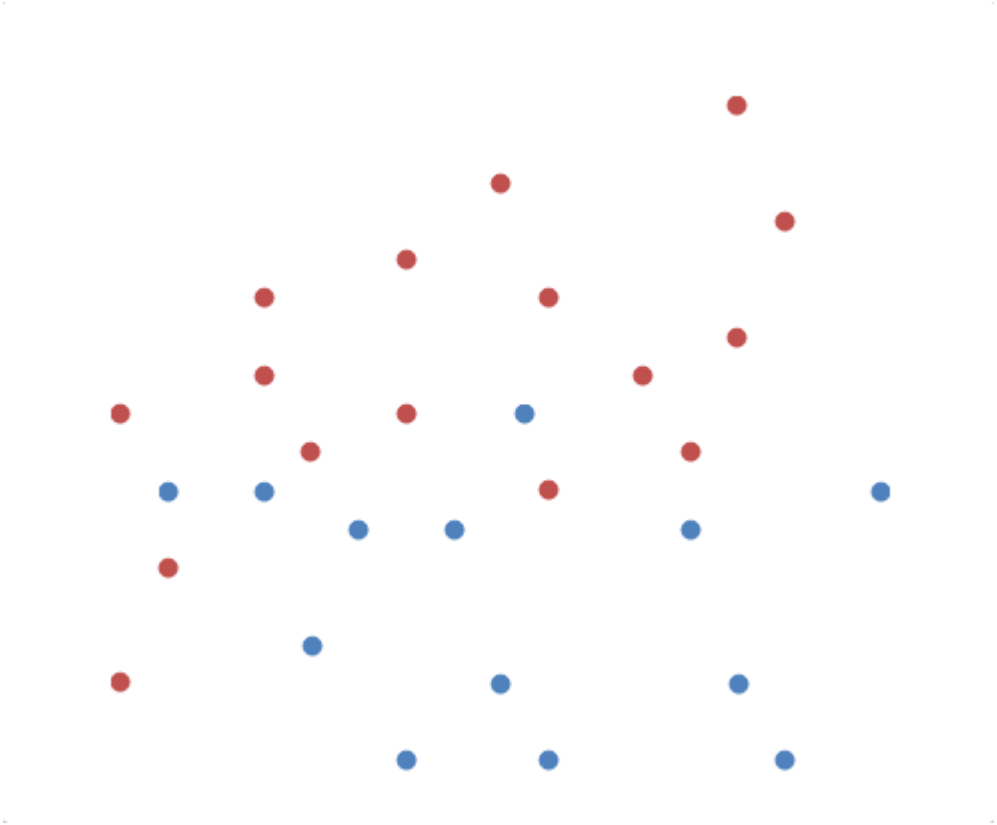
- 저 물음표에는 세모와 동그라미 중에 어떤게 들어갈까?
- KNN은 '?'의 주변에 있는 것이 세모이기 때문에 세모라고 판단하는 알고리즘이다. 매우 간단하고 직관적인 알고리즘이다.
- 하지만 단순히 가장 가까이에 있는 것과 같게 선택하는 것이 옳은 분류가 될까?



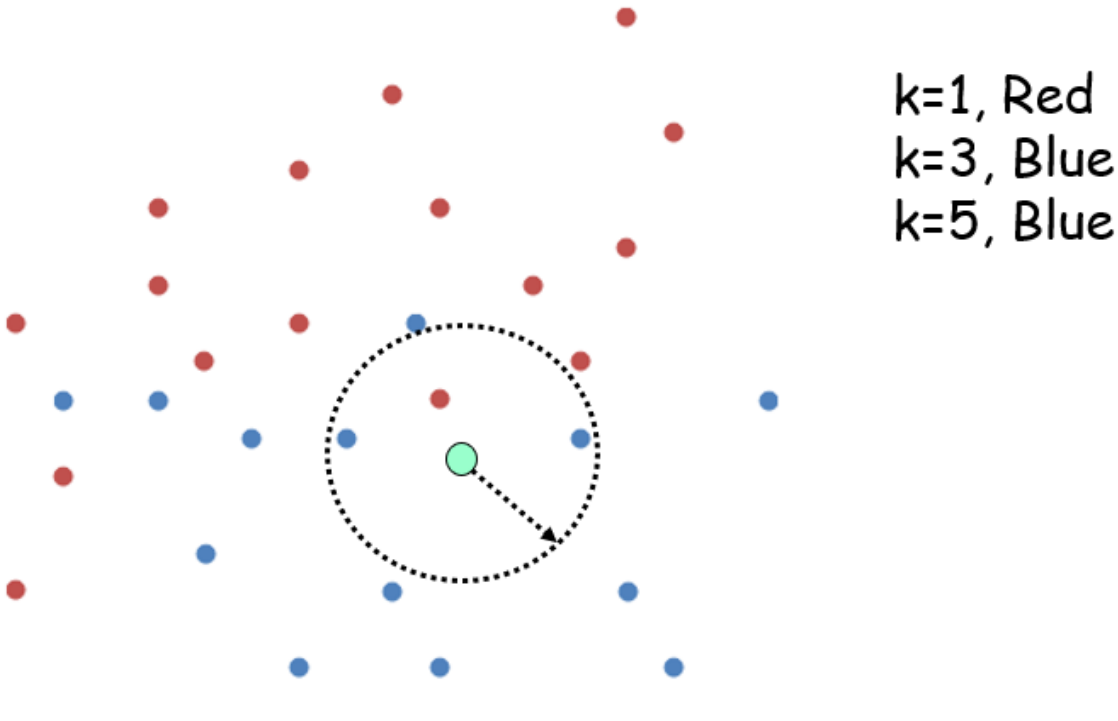
- ? 에 가장 가까운 것은 파란 동그라미 이다. 하지만 조금 더 넓혀서 본다면 뭔가 부적절하다는 것이 느껴진다.
- 이렇기 때문에 단순히 주변에 무엇이 가장 가까이 있는가를 보는 것이 아니라 주변에 있는 몇개의 것들을 같이 봐서 가장 많은 것을 골라내는 방식을 사용하게 된다. 이 방식을 KNN이라고 부른다. KNN에서 K는 주변의 개수를 의미한다.

Classification

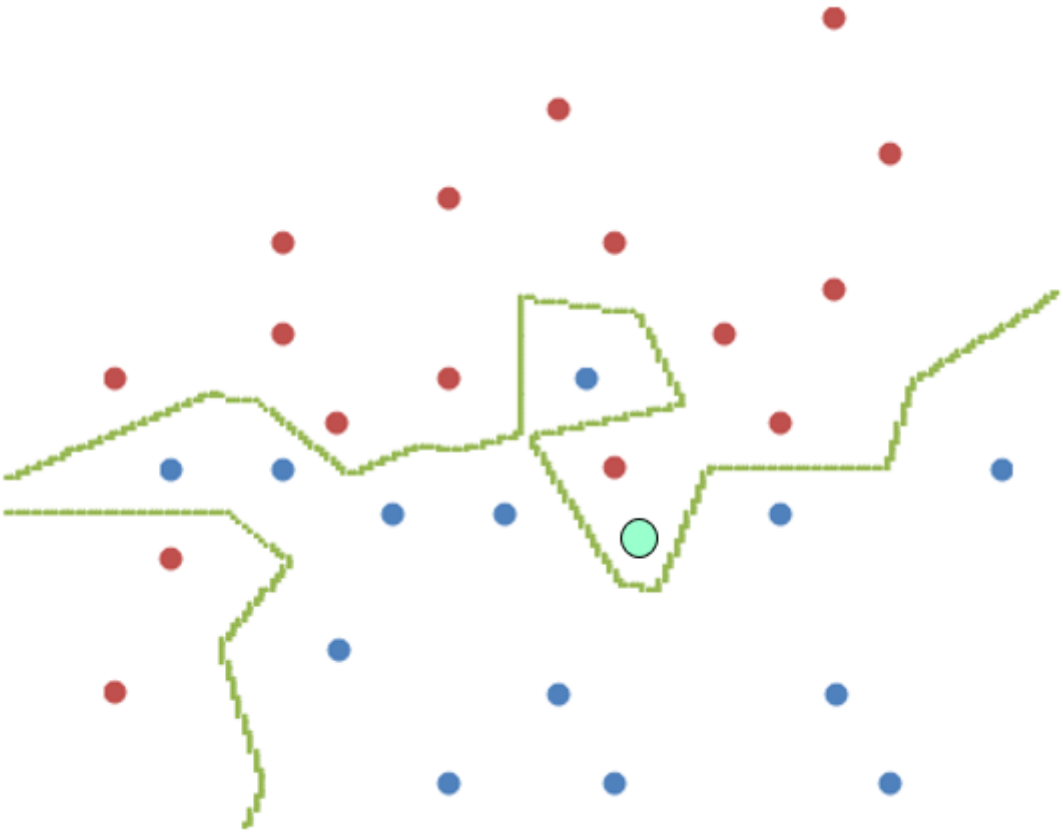
How to Predict Class of UnKnown Data?



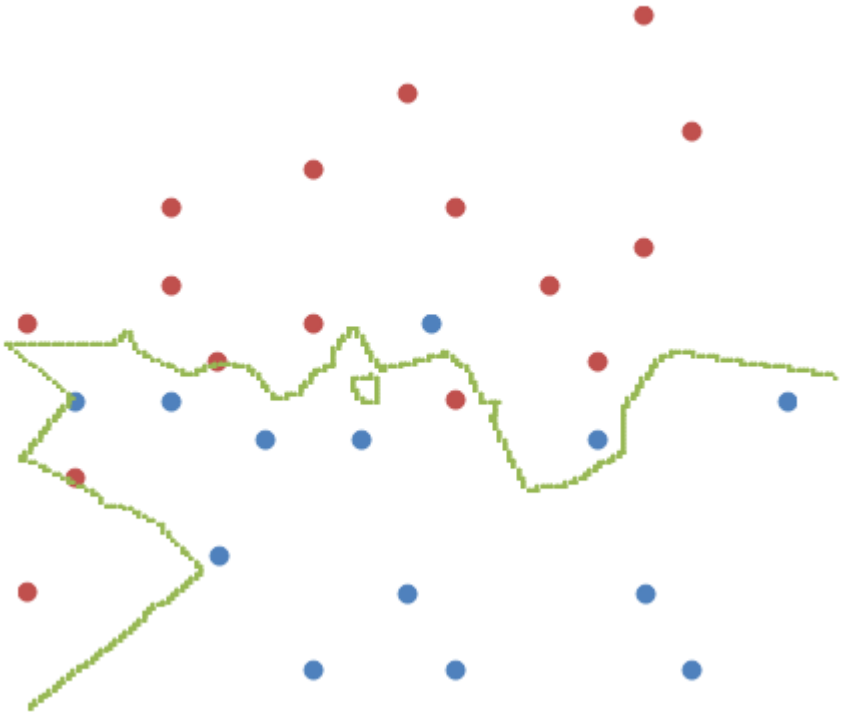
- Choose K nearest neighbors
- Determine the class based on the majority



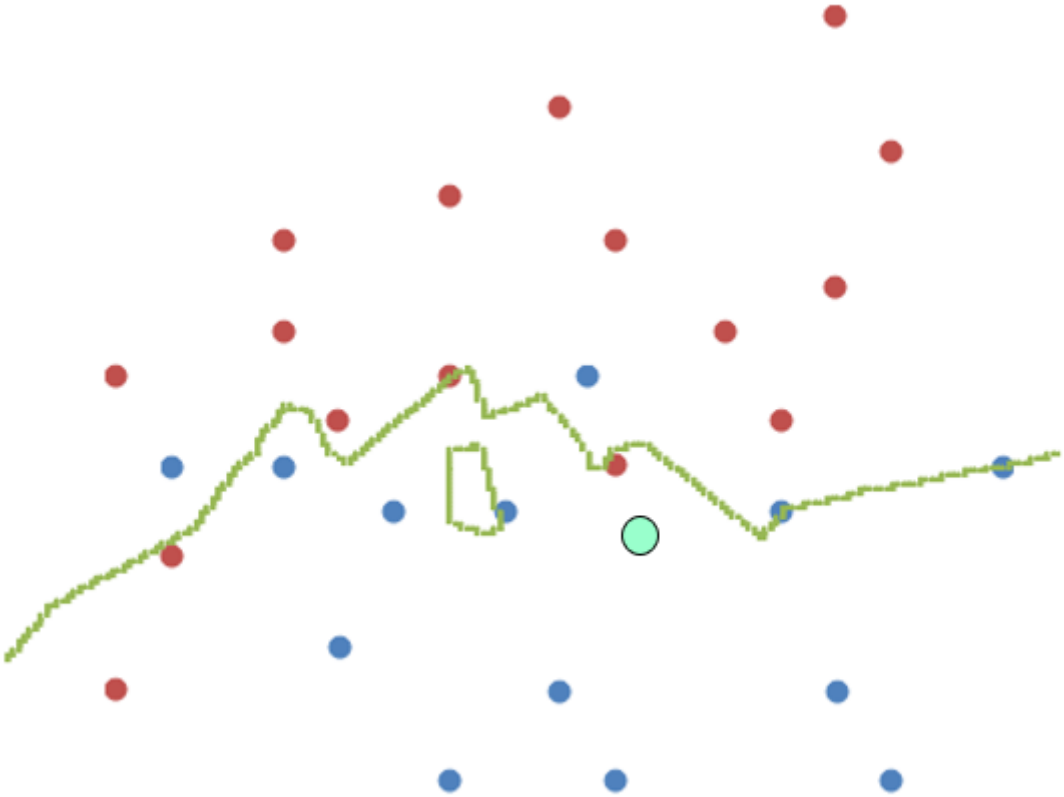
K=1일때



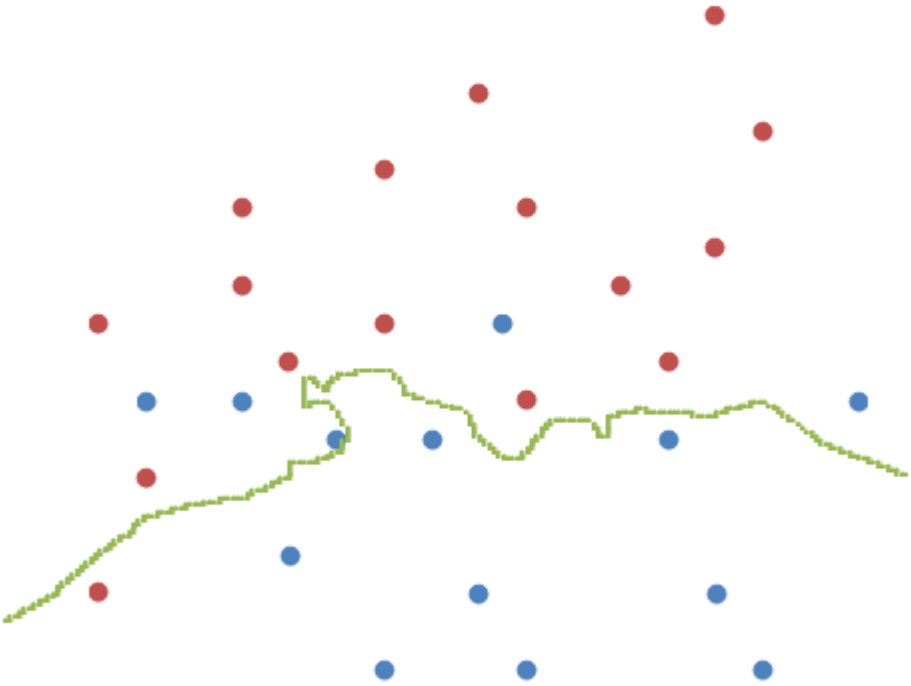
K=3일때



K=5일때



K=10일때



언뜻 보면 K가 적을때 거의 error가 없지만 overfitting의 문제가 있다.

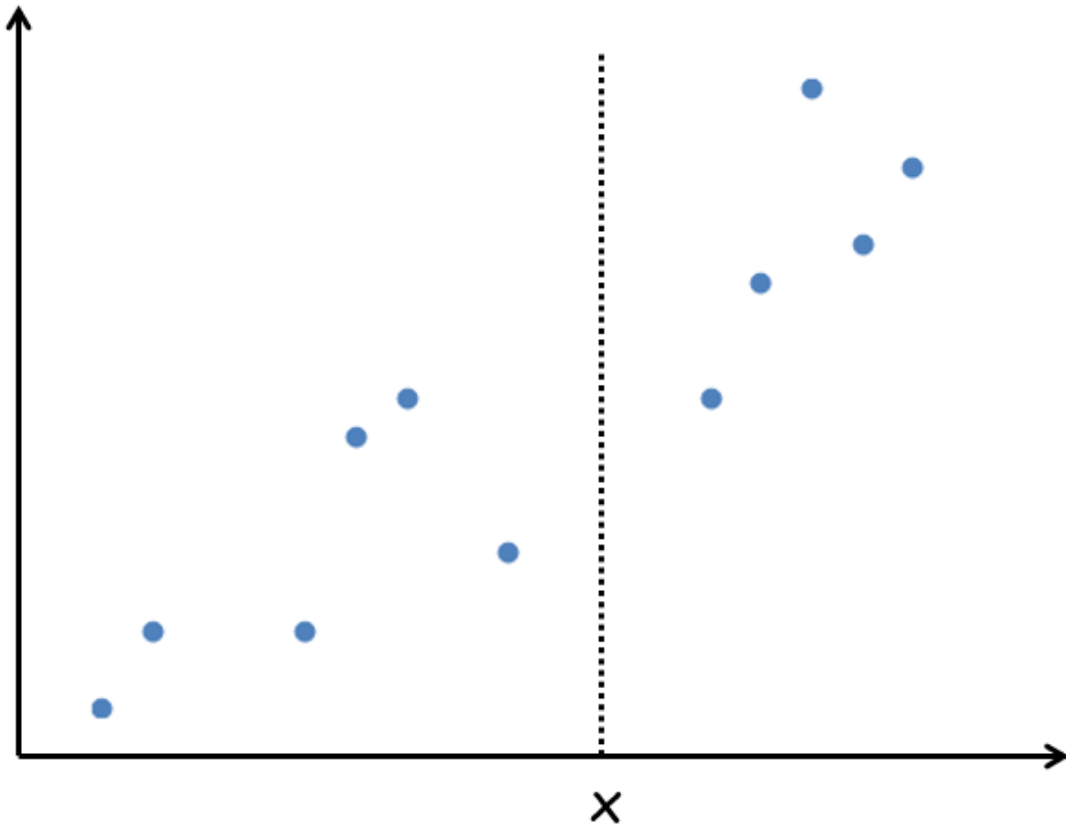
>>unknown데이터는 잘 못맞춘다는뜻이다.

Regression

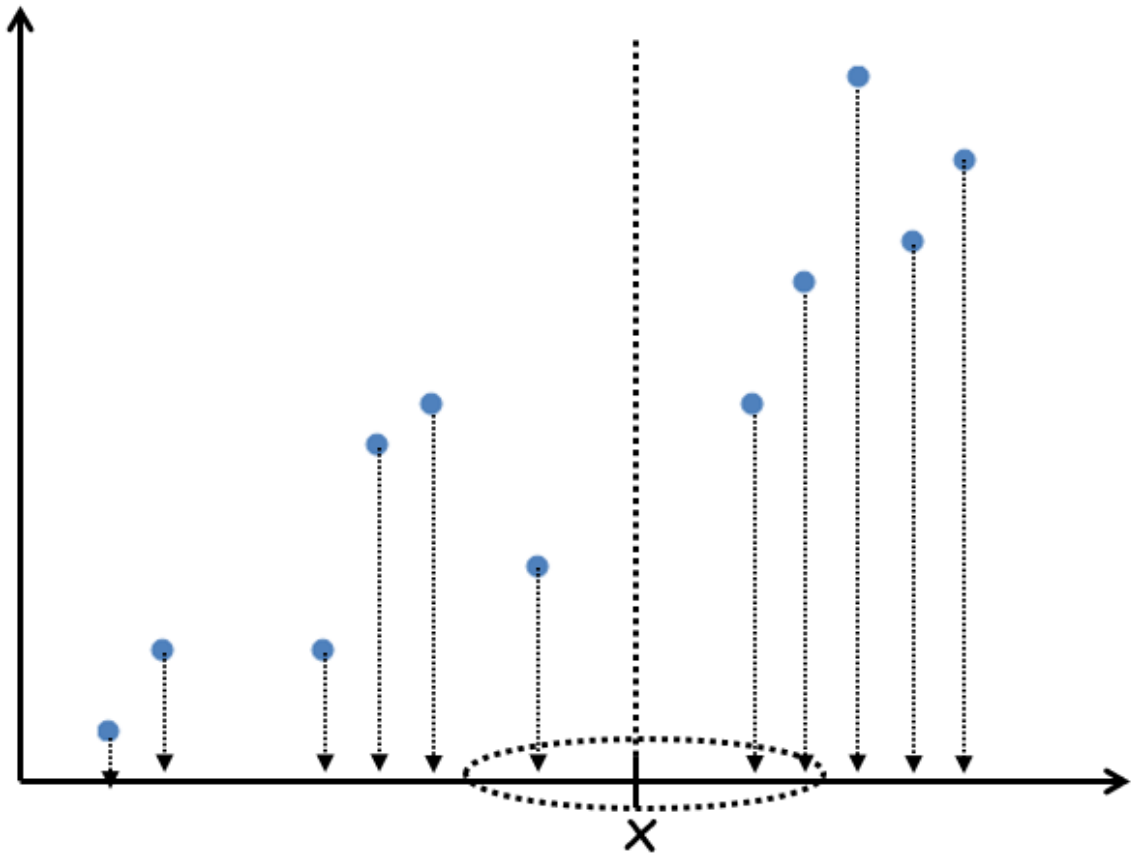
How to predict 'y' value of unknown data

Classification(분류)는 연속적이지 않은 레이블, 다시 말해 '무엇'인지를 예측하지만, 회귀(regression)는 연속된 수치, 즉 얼마나를 예측하는 것이다.

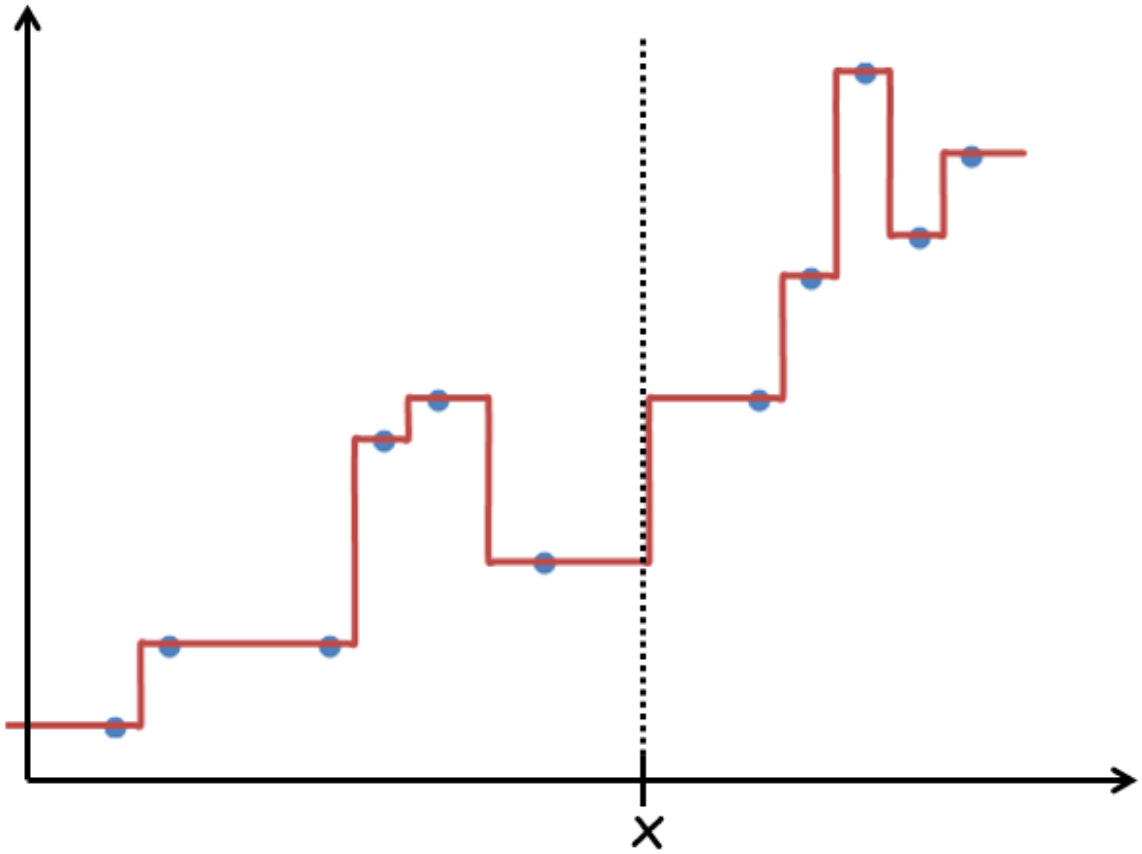
분류에서는 **이웃의 레이블** 개수를 확인해서 다수결로 정했지만, 회귀에서는 **이웃들의 평균**을 계산한다는 점에서 차이가 있다.



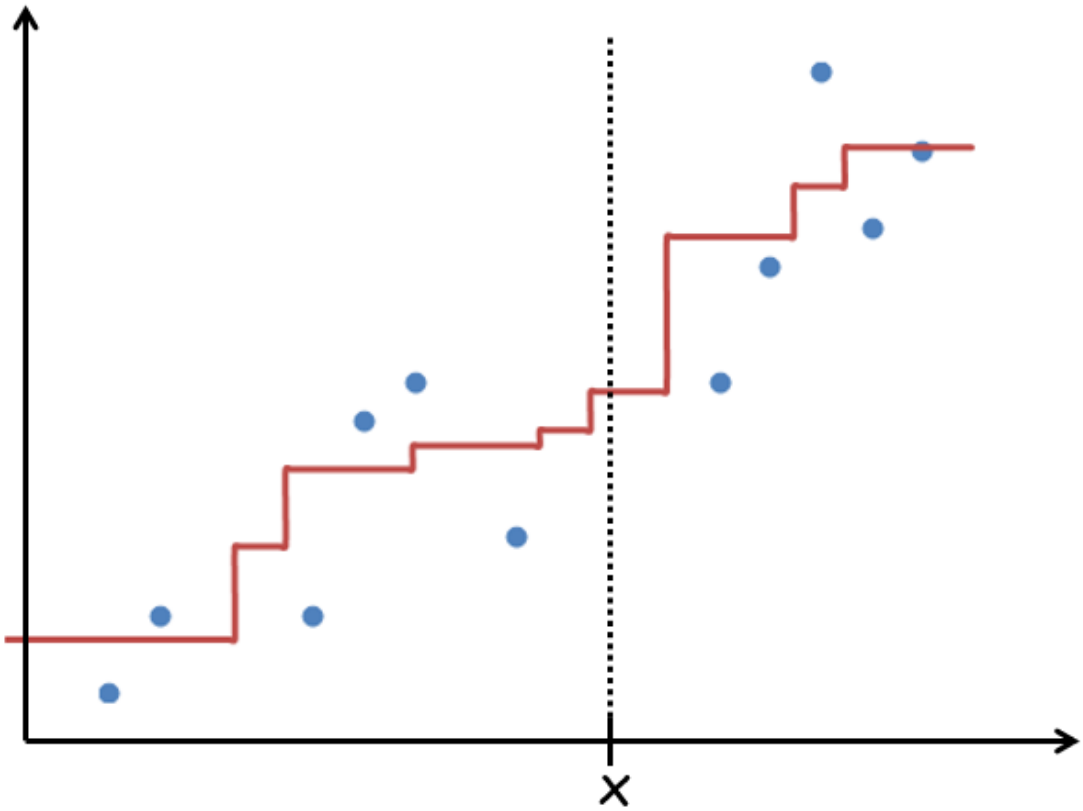
- Choose K nearest neighbors in X axis
- Predict the average of their 'y'



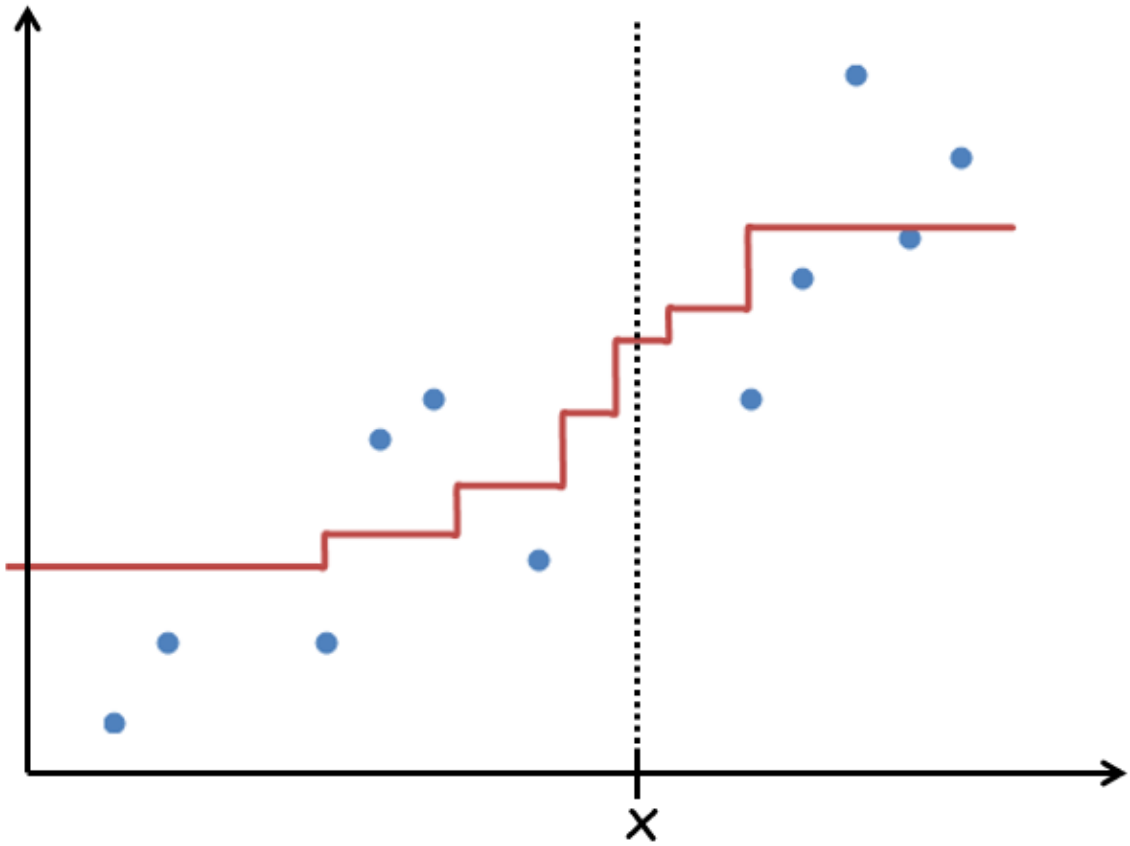
K=1일때



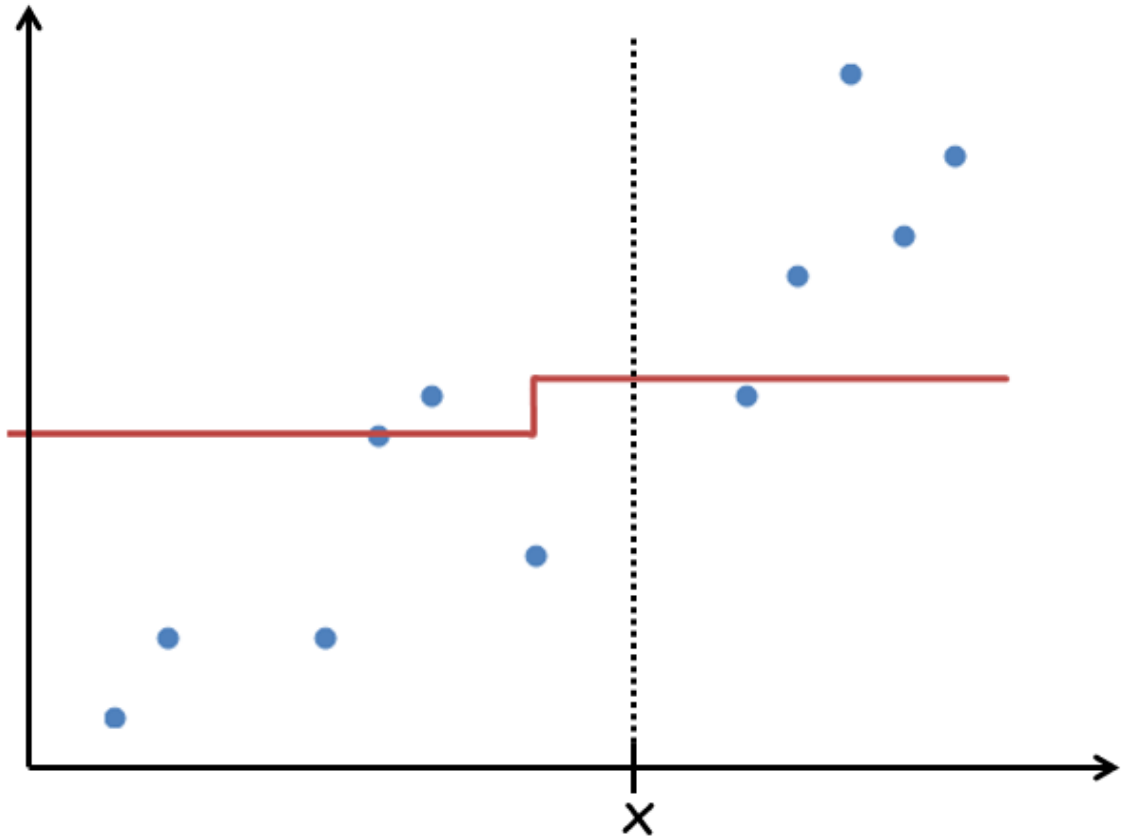
K=3일때



K=5일때



K=10일때

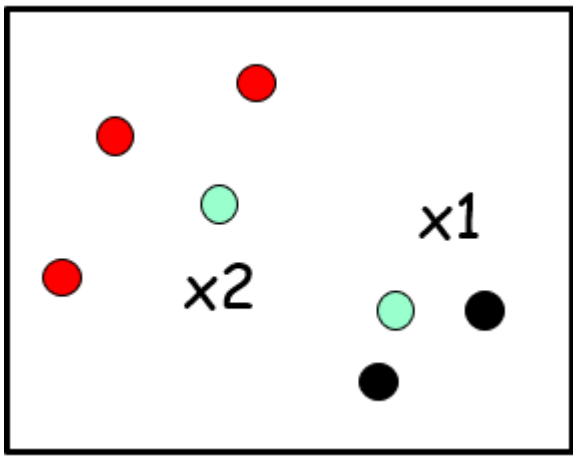


가까운 이웃들의 단순한 평균을 구하는게 아니라 각 이웃이 얼마나 가까이 있는지에 따라 가중 평균을 구하면 어떨까? 거리가 가까울수록 데이터가 더 유사할 것 이라고 보고 가중치를 부여하는것이다.

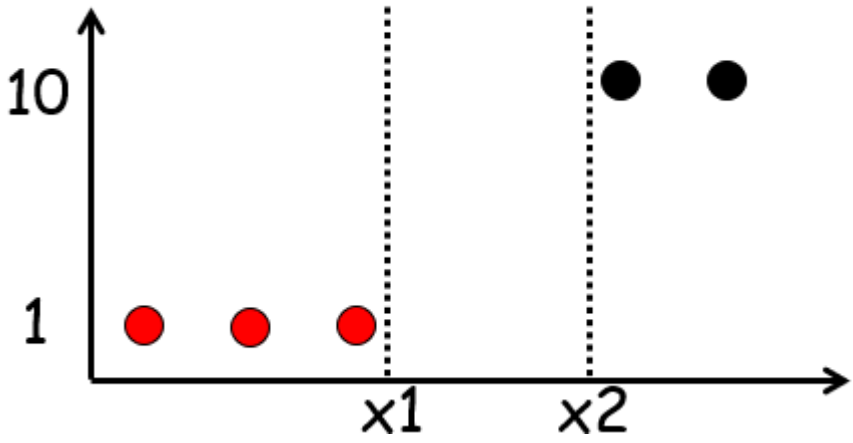
Variation

Why 'just' counting or averaging?

- Classification(k=5)
  - x1 : Red or Black? >>단순히 counting한다면 x1은 red이다.
  - x2 : Red or Black?>>마찬가지로 x2도 red이다.



- Regression(k=5)
  - x1 : close to 10 or 1
  - x2 : close to 10 or 1



More weight to closer one

- **Different weight depending on the distance** from x'

Classification : Not just counting

$$S(\mathbf{x}', R) = \sum_{\mathbf{x} \in N(\mathbf{x}', R)} w(\mathbf{x})$$

$$S(\mathbf{x}', B) = \sum_{\mathbf{x} \in N(\mathbf{x}', B)} w(\mathbf{x})$$

$$\text{if } S(\mathbf{x}', R) > S(\mathbf{x}', B) \text{ then } \mathbf{x}' \text{ is } R$$
$$\text{else } \mathbf{x}' \text{ is } B$$

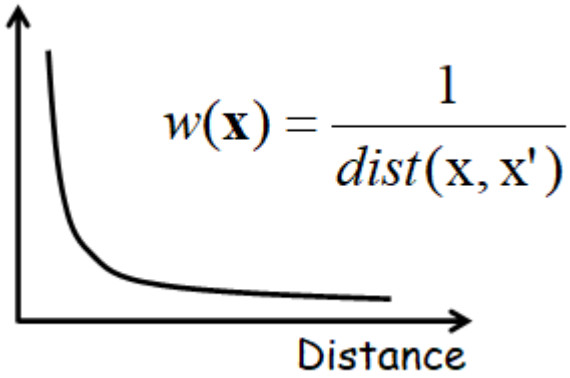
여기서 N(x',R) : the set of Red data among the nearest neighbors of x'

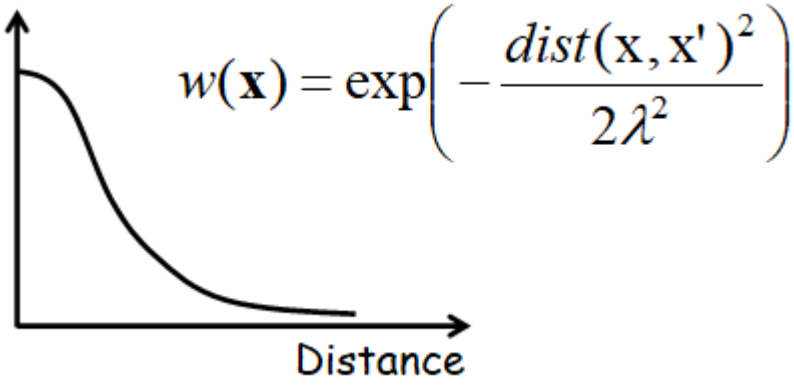
Regression : Not just averaging

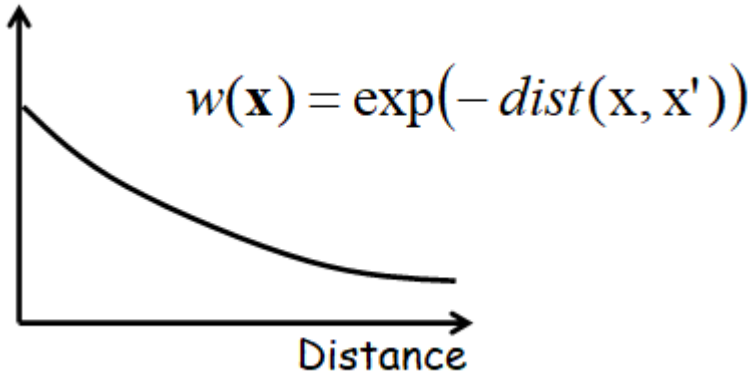
$$f(\mathbf{x}') = \frac{\sum_{\mathbf{x} \in N(\mathbf{x}')} w(\mathbf{x}) \cdot f(\mathbf{x})}{\sum_{\mathbf{x} \in N(\mathbf{x}')} w(\mathbf{x})}$$

N(x') : the nearest neighbors of x'

How to determine weight considering distance







여기 세개중 두번쨔 사진인 Gaussian을 제일 많이 사용한다.

가우시안 식에서 λ가 hyper-parameter이다.

λ와 k의 역할이 유사하다.

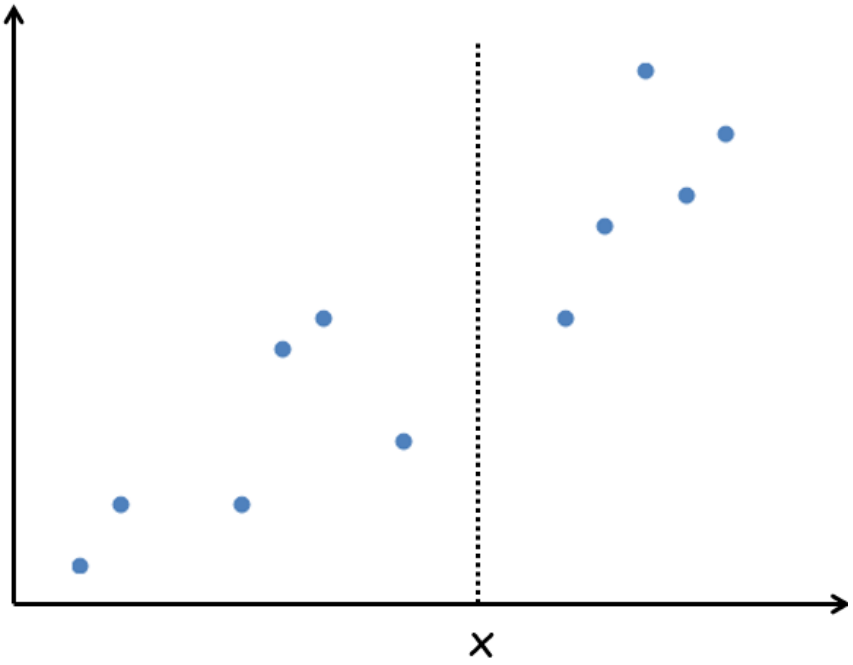
따라서 굳이 k를 정할 필요가 없다.

Do we need to choose K ?

- yes ,if you want
- **Not necessarily**

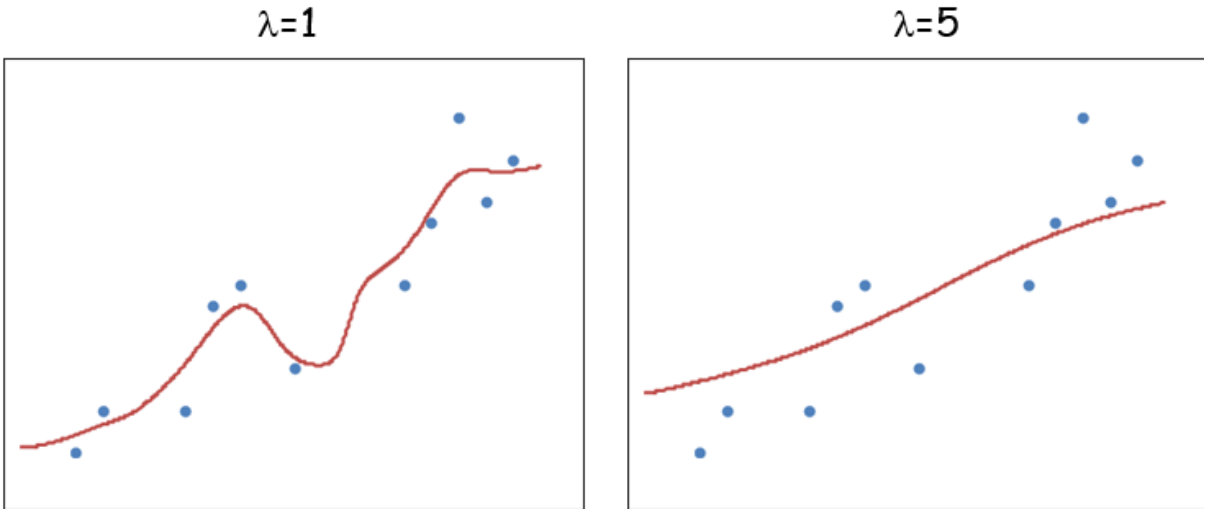
Kernel Regression

- Use Gaussian weight function
- Use all the given data



$$w(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{dist(\mathbf{x},\mathbf{x}')^2}{2\lambda^2}\right)$$

$$f(\mathbf{x}') = \frac{\sum_{\mathbf{x} \in \mathbf{X}} w(\mathbf{x}, \mathbf{x}') \cdot f(\mathbf{x})}{\sum_{\mathbf{x} \in \mathbf{X}} w(\mathbf{x}, \mathbf{x}')}$$

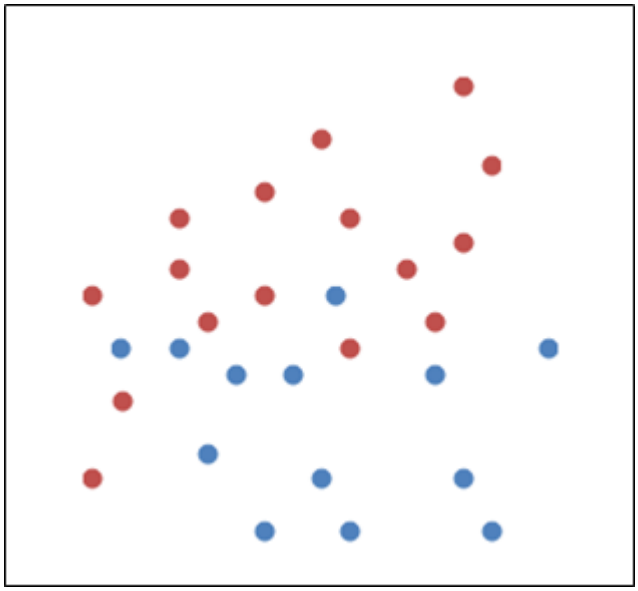


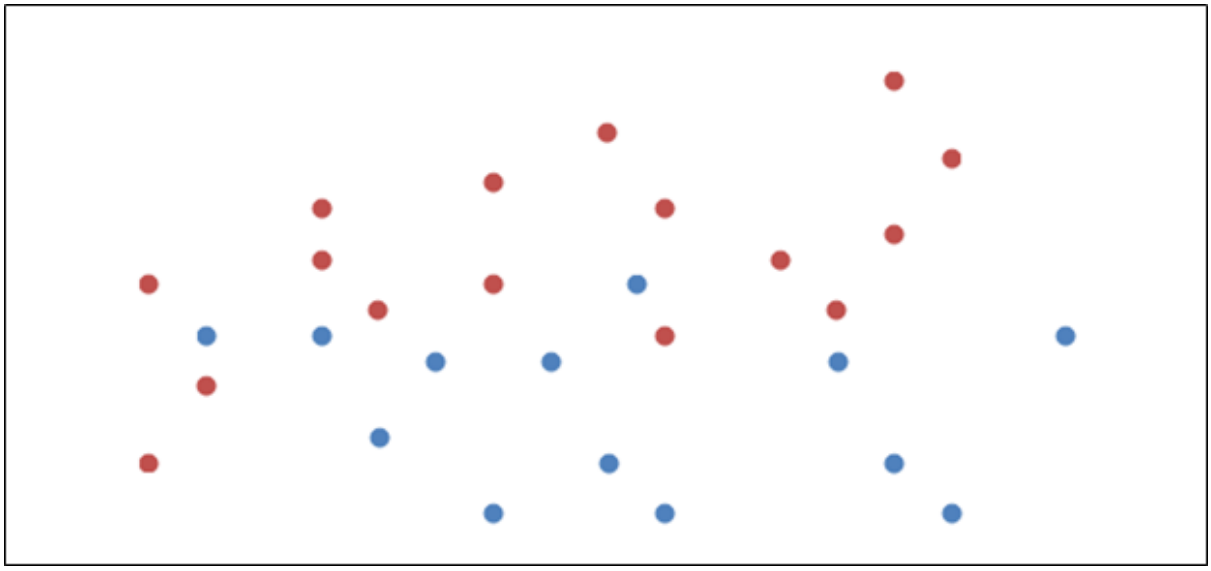
그렇다면 Distance Measure를 어떻게 정할까

Distance Measure

Two data sets

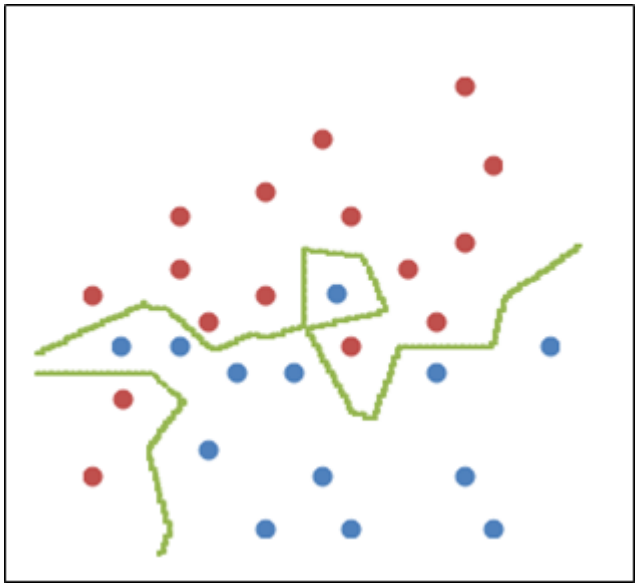
- Compare the boundaries in D1 and D2(1-NN)
- Are they scaled from each other?

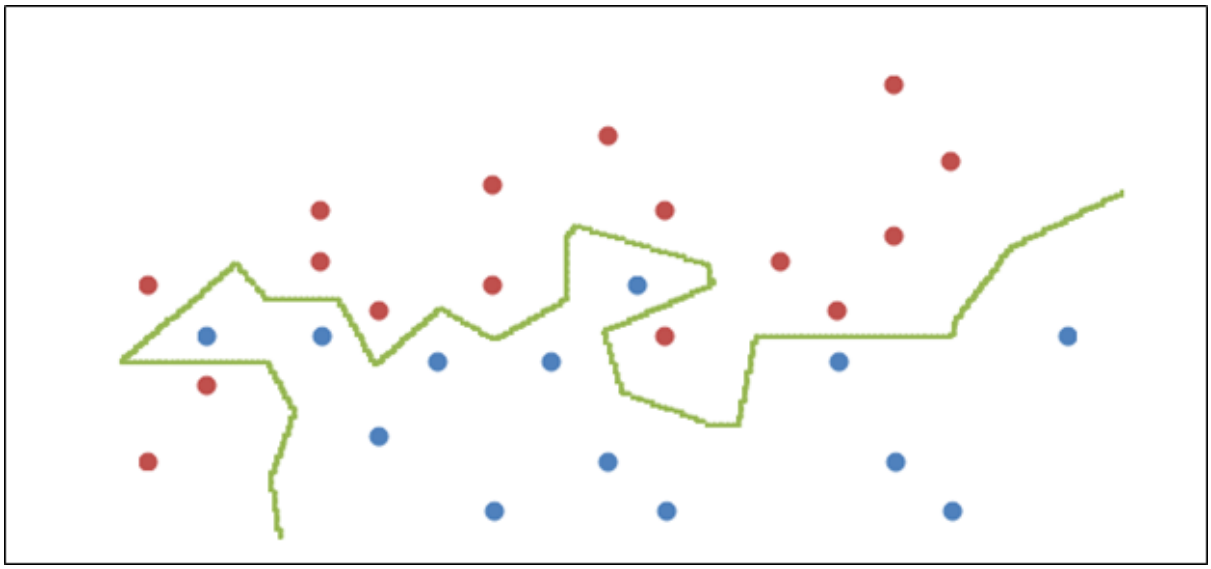




위가 D1 아래가 D2인데 D2는 D1을 x축으로 두배 늘린것이다.

- The data are scaled, the boundary change!!
- You may need to choose **other distance measures**





Why always Euclidian distance?

- Generalized Euclidian distance
- Mahalanobis distance
- Cosine similarity
- Pearson correlation
- ...

Can K-NN be applied only to numerical data??

>>No

>>If we can define a distance, we can apply it to any types of data including sets, text, etc...

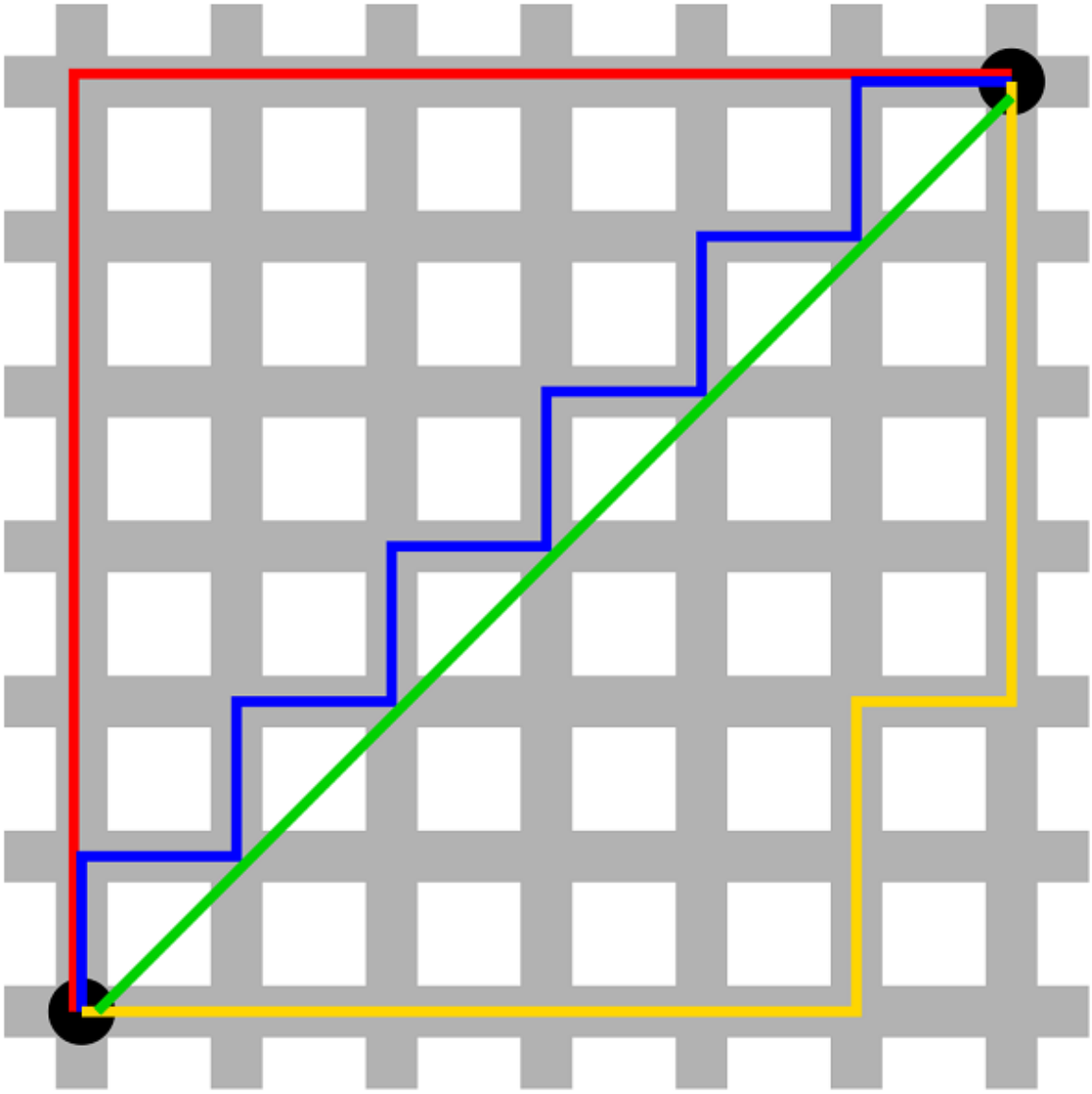
이제 각 distance measure을 알아보자@

Generalized Euclidian distance

우선 Norm이란 무엇일까??

Norm이란 벡터공간에서 벡터의 크기 또는 길이를 측정할 때 사용되는 개념이다.

- L1 norm(Manhattan distancee)



좌측 아래 점과 우측 상단 점을 각각 시작점과 도착점이라 봤을 때, 최단거리는 유클리디안 거리로 구하면 된다. 즉, 초록색은 유클리드 거리를 선으로 그은것과 같다.

그러나 좌표가 아니라 도시라고 생각을 해보자, 도시가 유클리드 거리를 연산하는 것처럼 뻥 뚫려 있을리가 없다. 그렇기 때문에 현실적인 거리를 구해야 하는데 건물이 위 이미지 모양처럼 있다면 우리는 가장 가까운 거리는 대부분 파란색이라 착각할 수 있다.

하지만, 파란색과 노란색과 붉은색 선들은 모두 총길이가 같다는 함정이 있다. 이것이 바로 맨하탄거리이다.

왼쪽 아래의 좌표를 (x1, y1)이라 하고, 우측 상단의 좌표를 (x2, y2)라고 할 때, 맨하탄 좌표의 공식을 적어보면 다음과 같다.

|x1-x2| +|y1-y2|

- L2 norm: 우리가 알고 있는 일반적인 유클리드안 거리

$$dist(\mathbf{x},\mathbf{y}) = \sqrt{\sum (x_i - y_i)^2}$$

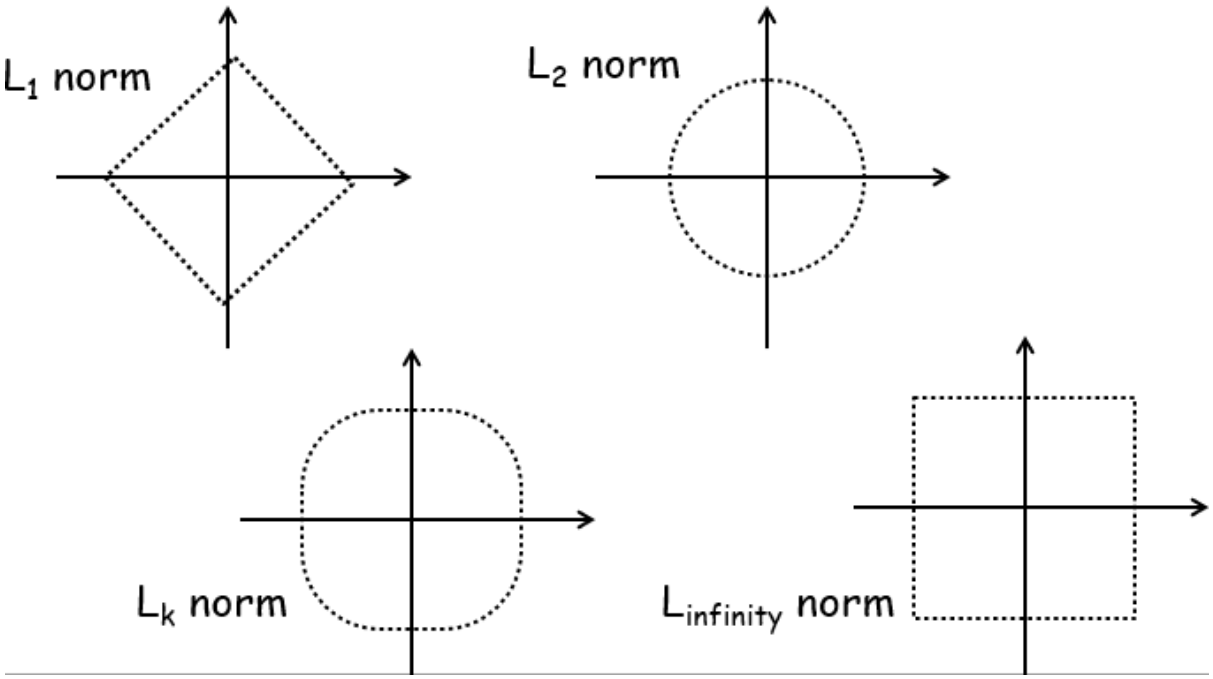
- Lk norm

$$dist(\mathbf{x},\mathbf{y}) = \sqrt[k]{\sum (x_i - y_i)^k}$$

- L infinity norm

$$dist(\mathbf{x},\mathbf{y}) = \sqrt[\infty]{\sum (x_i - y_i)^\infty} = \max_i |x_i - y_i|$$

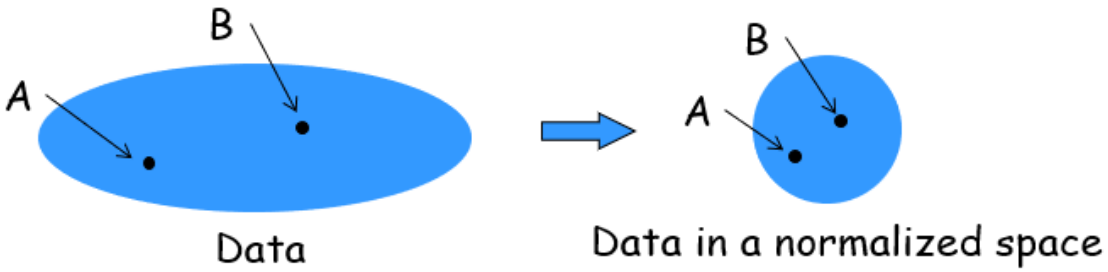
$$\mathbf{x} = (x_1, x_2, \cdots, x_n),\ \mathbf{y} = (y_1, y_2, \cdots, y_n)$$



Mahalanobis distance

- Euclidian distance in a normalized space(un-correlated case)

$$dist(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}} \qquad \mathbf{x} = (x_1, x_2, \cdots, x_n), \quad \mathbf{y} = (y_1, y_2, \cdots, y_n)$$



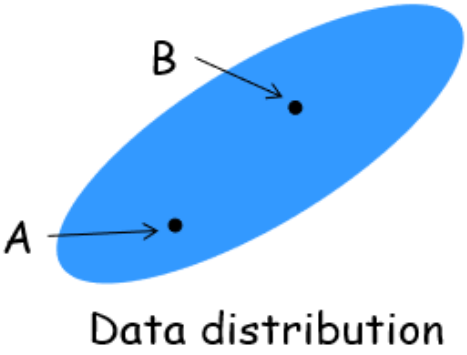
(x_1, x_2) $\qquad x_1^M = \frac{x_1 - \mu_1}{\sigma_1} \qquad x_2^M = \frac{x_2 - \mu_2}{\sigma_2}$

- Euclidian distance in a normalized space(correlated case)

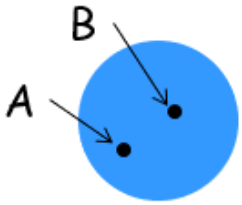
$$dist(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})\Sigma^{-1}(\mathbf{x}-\mathbf{y})^T}$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n),$$
$$\mathbf{y} = (y_1, y_2, \cdots, y_n)$$

Σ is the covariance matrix of given data



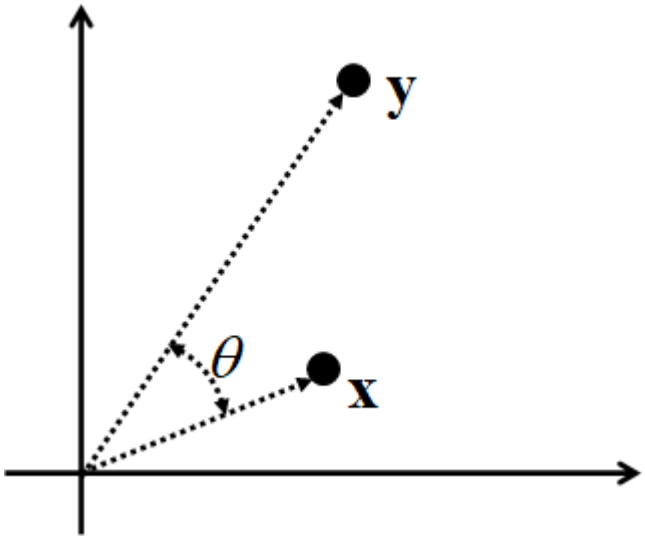
Data distribution



Data distribution

Cosine similarity

$$dist(\mathbf{x},\mathbf{y}) = \cos\theta = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$



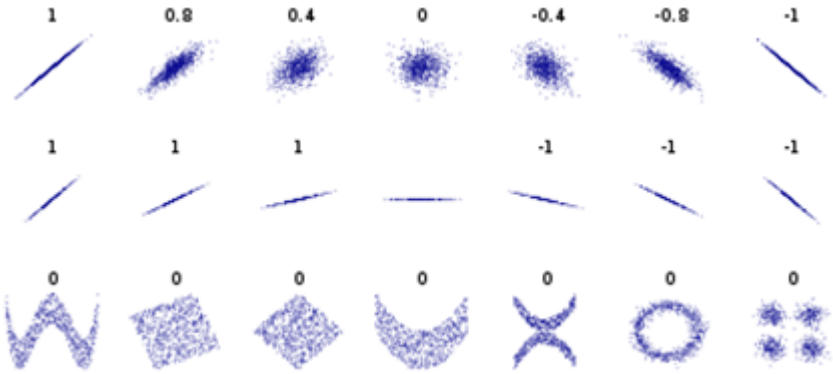
$$\mathbf{x} = (x_1, x_2, \cdots, x_n),$$
$$\mathbf{y} = (y_1, y_2, \cdots, y_n)$$

Pearson Correlation Coefficient

$$dist(\mathbf{x}, \mathbf{y}) = corr(\mathbf{x}, \mathbf{y}) = \frac{\sigma^2_{\mathbf{x},\mathbf{y}}}{\sigma_{\mathbf{x}} \cdot \sigma_{\mathbf{y}}} = \frac{\mathrm{cov}(\mathbf{x}, \mathbf{y})}{\sigma_{\mathbf{x}} \cdot \sigma_{\mathbf{y}}} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \cdot \sum (y_i - \bar{y})^2}}$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n), \mathbf{y} = (y_1, y_2, \cdots, y_n)$$

+1은 완벽한 양의 선형 상관 관계, 0은 선형 상관 관계 없음, -1은 완벽한 음의 선형 상관 관계를 의미한다.



Jaccard Similarity

- Similarity measure between two sets

$$dist(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Summary

Which K is better?

- Small K : higher variance (less stable)>>overfitting
- Large K : higher bias(less preise) >>over-generalization

Proper choice of K

- Depending on the data
- Use Cross-validation

K-Nearest Neighbors is a kind of

- Case-based reasoning(Instance-based, Memory -based)

- Lazy learning(lazy learning이 뭐지??)
- Non-parametric method(↔model based method)

Lazy vs Eager

- Lazy : wait for query before generalizing
  - instance-based learning
  - test sample 이 들어오면 그때 test sample과 연관성이 높은 instance를 memory 에서 찾는다.
  - test sample에 기반해 뽑힌 데이터로 local model을 만들고 inference를 한다.
  - 학습 시간 보다 예측 시간이 더 걸린다.
- Eager : generalize before seeing query
  - training time에 training data만으로 모델이 만들어진다.
  - new query 가 들어오면 미리 만들어둔 모델으로 예측을 하기 때문에 test sample 은 모델의 구축에 영향을 미치지 못한다.

Parametric vs Non-parametric

- parametric : build a model with given data
- Non-parametric : keep given data

Advantage

- No training(Only inference step)
- Complexity of target functions do not matter

Disadvantage

- Have to keep all data →Memory space
- Sensitive to noise

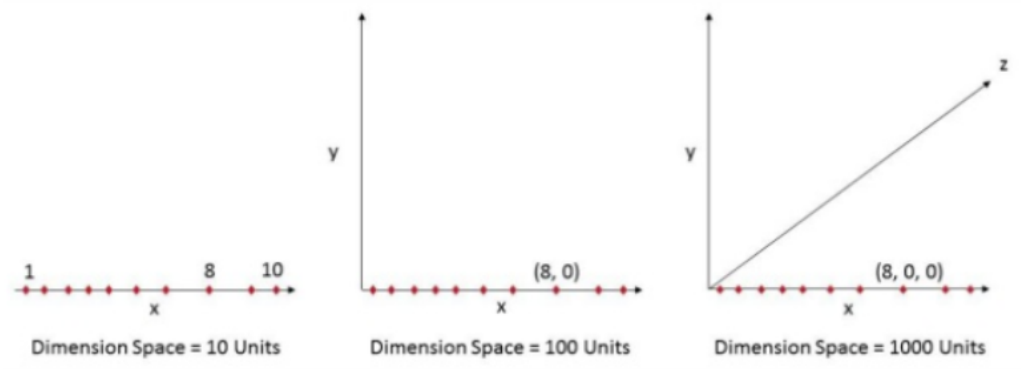
- If training data is imbalancced, major class may dominate
- Need to calculate the distance from all training data → Time
- In high dimensional space, nearest neighbors are not near
  - curse of dimensionality
  - 차원이 커질수록 빈공간이 커져 데이터간 거리가 멀어지게 된다.

차원의 저주 (Curse of dimensionality)

: 수학적 공간 차원(=변수 개수)이 늘어나면서, 문제 계산법이 지수적으로 커지는 상황

: 차원이 높아질수록 데이터 사이의 거리가 멀어지고, 빈공간이 증가하는 공간의 성김 현상(Sparsity)을 보임

*KNN(K-Nearest Neighbors) 분류 알고리즘에서 흔하게 발생하는 문제



(좌측 그림) - 1차원에 10개의 데이터가 존재 (10^1=10)

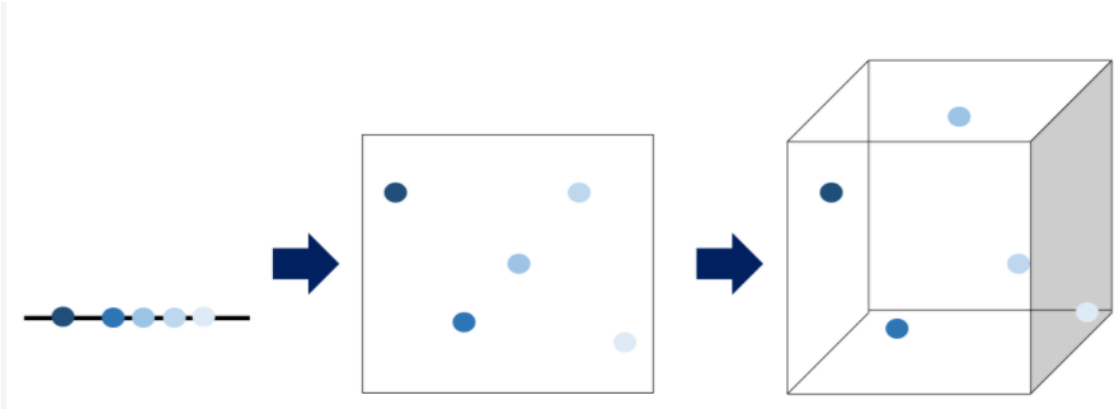
(중앙 그림) - 2차원에 100개의 데이터가 존재 (10^2=100)

(우측 그림) - 3차원에 1,000개의 데이터가 존재 (10^3=1,000)

→ '8'의 위치를 설명하는 상황에서, 차원이 커질수록 설명 공간이 지수적으로 늘어남

→ Feature가 많아질수록, 동일한 데이터를 설명하는 빈 공간이 증가함

→ 차원의 저주로 인해, 알고리즘 모델링 과정에서 저장 공간과 처리 시간이 불필요하게 증가됨 (성능 저하)



Reference

https://bioinformaticsandme.tistory.com/197

https://datapedia.tistory.com/15