Bayes Classifier

- Y = f(X) 에서 Y가 카테고리(클래스) 변수이고 클래스가 c1, c2, … c10 처럼 10개의 카테고리로 되어있을 때 f가 Classification을 해야 하고,

- predictor X가 'x_0' 일 때, response Y가 카테고리 c2일 확률을
  Pr(Y=c2 | X = x_0) 인 conditional probability(조건부 확률)로 표현

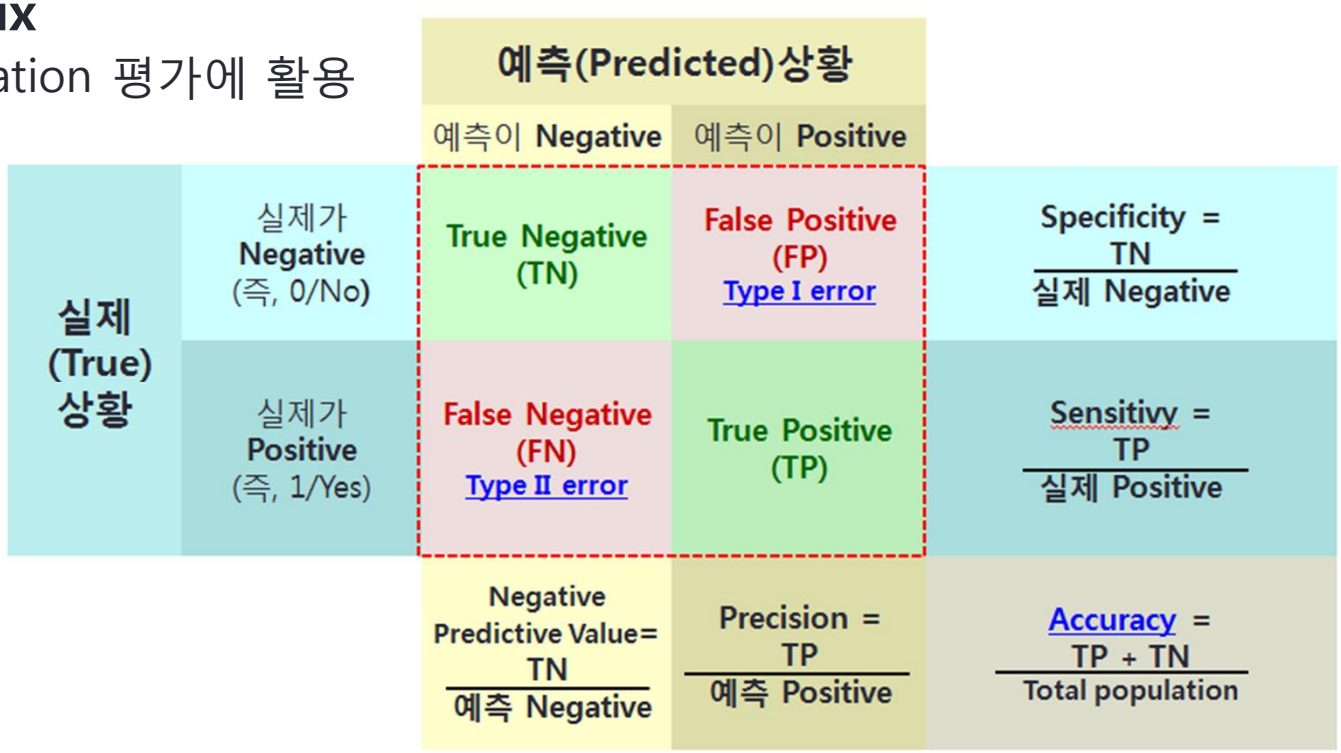
- Bayes Classifier는 X = x_0 인 observation을 어떤 클래스로 분류할지를 판단할 때, 모든 카테고리 c1, c2,… c10 각각에 대해 조건부 확률을 계산해서 그 중 가장 큰 값을 보이는 클래스로 observation을 분류한다

- Bayes Classifier는 평균적으로 가장 낮은 test error rate (클래스를 틀리게 분류하는 비율) 를 보인다

- **즉, 앞의 조건부 확률들을 구할 수만 있으면 Bayes Classifier는 평균적으로 가장 좋은 성능을 보인다 (잘 분류한다). 그런데 실제로는 그게 안된다. 따라서, 이 조건부 확률(분포)들을 잘~ 추정해서 가장 그럴 듯한 클래스에 observation을 배정하는 방법을 쓴다**

Confusion Matrix

- Binary Classification 평가에 활용



찾고자 하는 (관심이 있는) 클래스를 Positive, 1로 놓는다

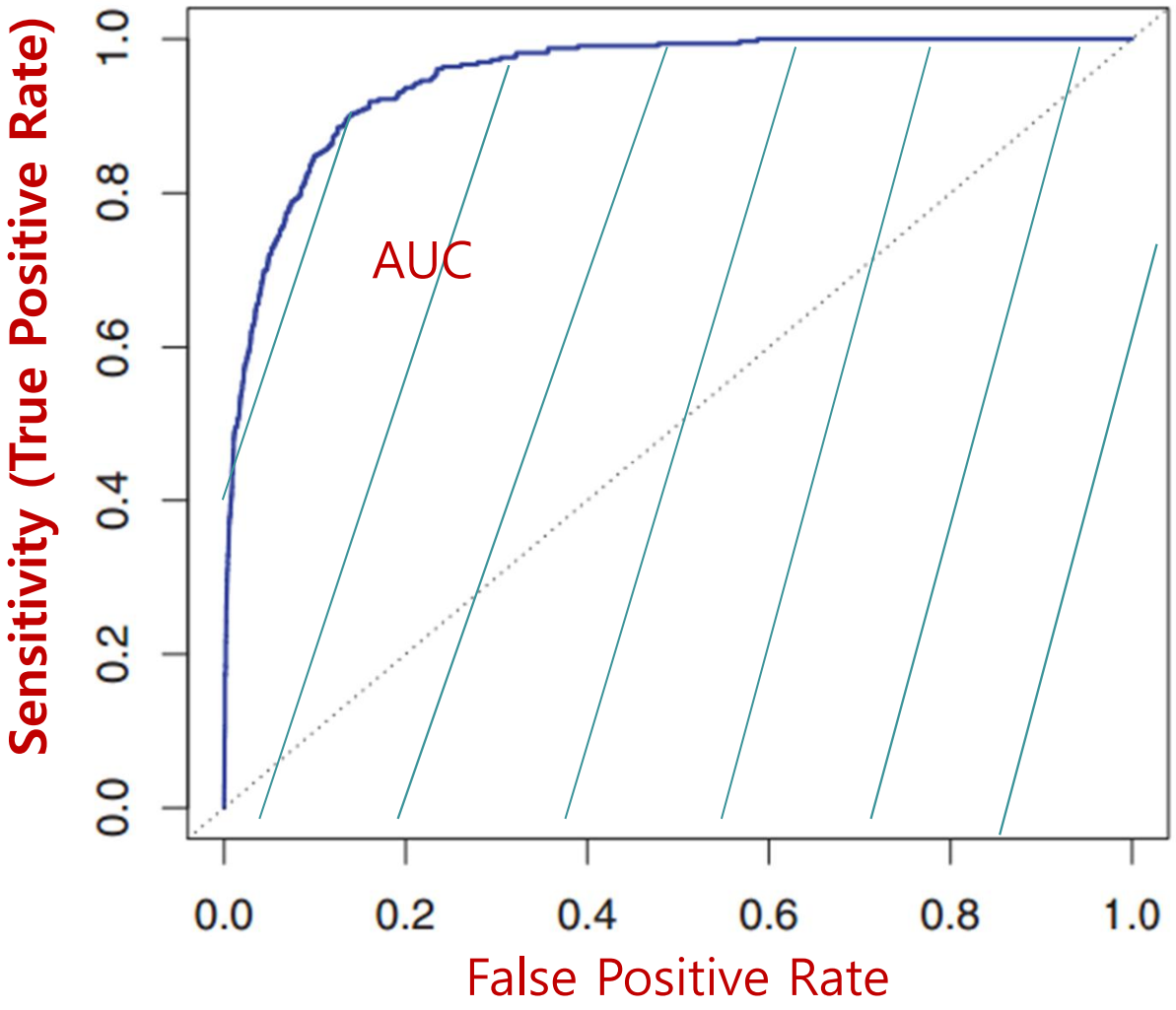
- **예를 들어, 암 진단, 비행기 탐지, 채무 불이행자 예측 시 암발견, 비행기 발견, 채무 불이행자로 판단을 Positive 1로**

- **TP** : 실제 암이 있고 이를 있다고 **맞게** 예측한 경우
- **TN** : 실제 암이 없고 이를 없다고 **맞게** 예측한 경우
- **FP** : 실제는 암이 없는데 이를 있다고 **틀리게** 예측한 경우
- **FN** : 실제는 암이 있는데 이를 없다고 **틀리게** 예측한 경우

❖ **Accuracy (정확도)** : 예측이 맞은 비율 (전체 예측 중에 TP와 TN 합의 비율)
❖ **Sensitivity (민감도**, True Positive Rate, Recall): 실제 암을 얼마나 예측이 이를 맞추었는지 (탐지했는지)
❖ **Precision** : 암이라고 (Positive) 예측을 한 것 중 실제로 **맞은 비율 (Positive 예측이 얼마나 정확한지?)**
❖ **Specificity (특이도, True Negative Rate) :** 실제로 암이 **아닐 때**, 얼마나 이를 **맞게** 예측한 비율
❖ **False Positive Rate (FPR)** : 실제는 암이 아닌데 (Negative), 암이라고 (Positive) **틀리게** 예측한 비율 :
FPR = 1 – Specificity = FP/(FP + TN)

ROC (Receiver Operating Characteristic) Curve

-Threshold 를 변화함에 따라 TRP과 FPR 점들을 그린 것. 커브가 좌상단 모서리에 붙을수록 좋다



ROC :
Binary Classifier의 성능을 간략하게 나타낼 수 있는 지표로서, positive 예측의 threshold(임계치)를 변화시킴에 따른 sensitivity와 False Positive Rate의 변화를 그린 곡선

AUC (Area Under the Curve) :
ROC 아래부분 면적
- 일반적으로, 0.5 < AUC < 1.0

• **False Positive Rate** : 실제 negative 중에서 FP로 잘 못 식별한 비율 (1 – specificity)

K-Nearest Neighbors (KNN) Classification & Regression

Idea :

어떤 feature 값 x_0를 갖는 observation의 response를 추정하고 싶을 때, training set observation 중에서 x_0와 가장 가까운 K개의 observation을 추려,

- Regression이면 위의 K개 observation의 **평균값**
- Classification이면 위의 K개 observation들의 class/카테고리 중에 **최다수를** 차지하는 클래스를

답으로 내 놓는다

K-Nearest Neighbors (KNN) Classification 과 Regression 은

세상에서 가장 단순한 학습 모델!

왜?

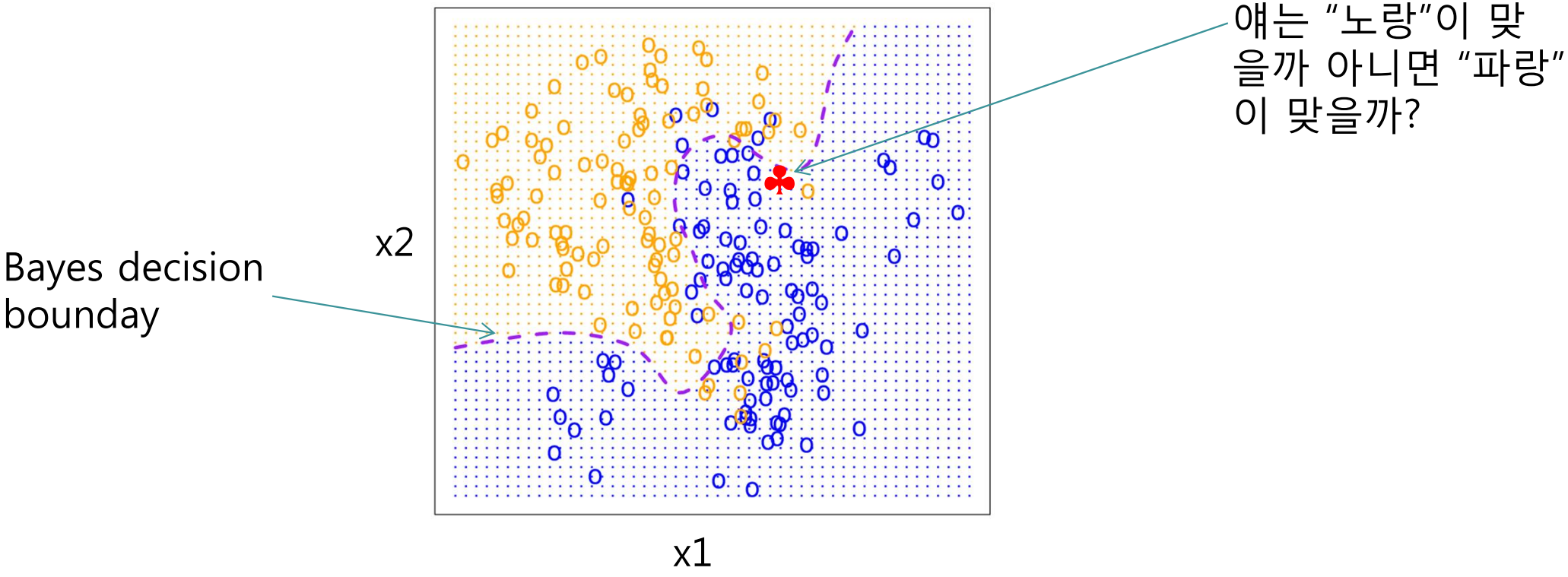
- 학습이라고 할 만한 것이 없다

- '학습' 때 한 일은 training set의 observation 들의 위치를 기억한 것 밖에 없으니

K-Nearest Neighbor Classifier

- Classification 모델 Y = f(X) 에서 'n'을 observation 개수, 'p'를 predictor의 feature dimension 개수, 'k'를 카테고리 개수로 나타낸다 하자.

- k=2 즉 클래스가 2개(노랑, 파랑)로 나누어지고, p=2 즉 predictor는 2개의 feature (x1, x2)로 구성되고, n=100 즉 100개의 training observation이 주어졌다고 생각하자. 그리고 이 정보를 아래와 같이 나타내보자



- **이 세상에서 가장 단순한 생각** : ♣ **주변**에 어떤 것이 있나 봐서 그들 중에 가장 많이 보이는 클래스에 ♣ 의 클래스를 삼음

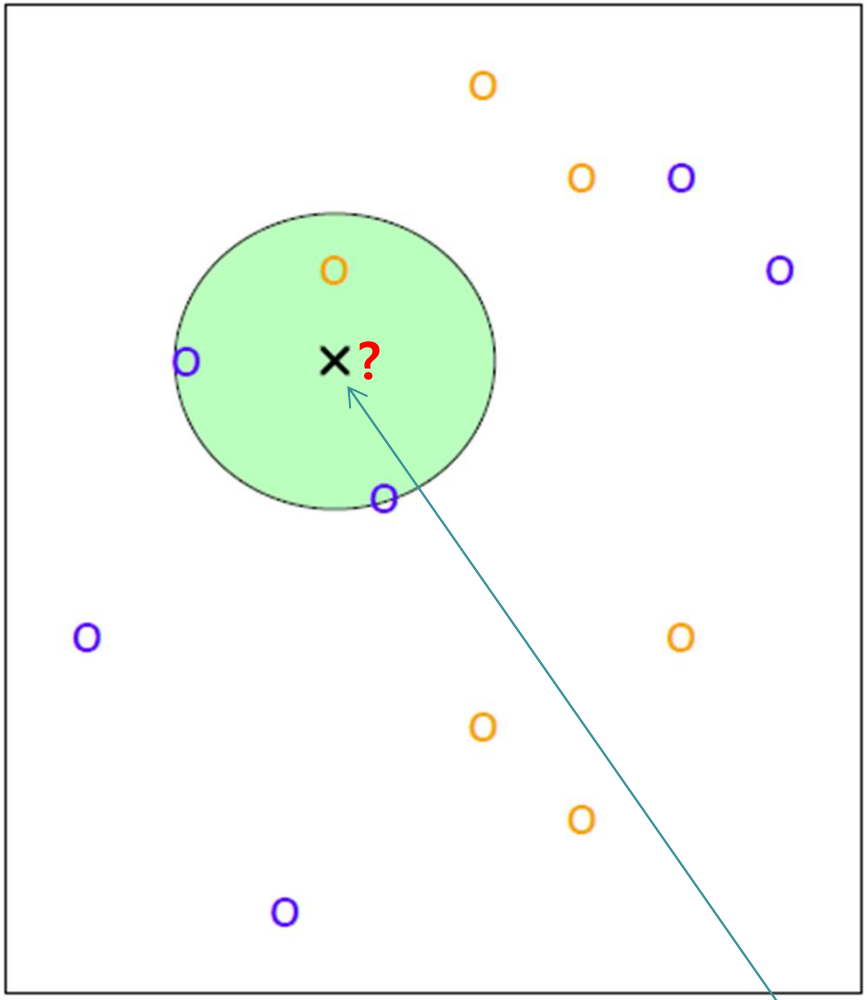
- **이슈** : 그러면, 그 "주변"이 어느 정도 크기면 적당한가? ♣ 가까이에 노랑, 파랑 샘플이 같은 수로 있을 때는?

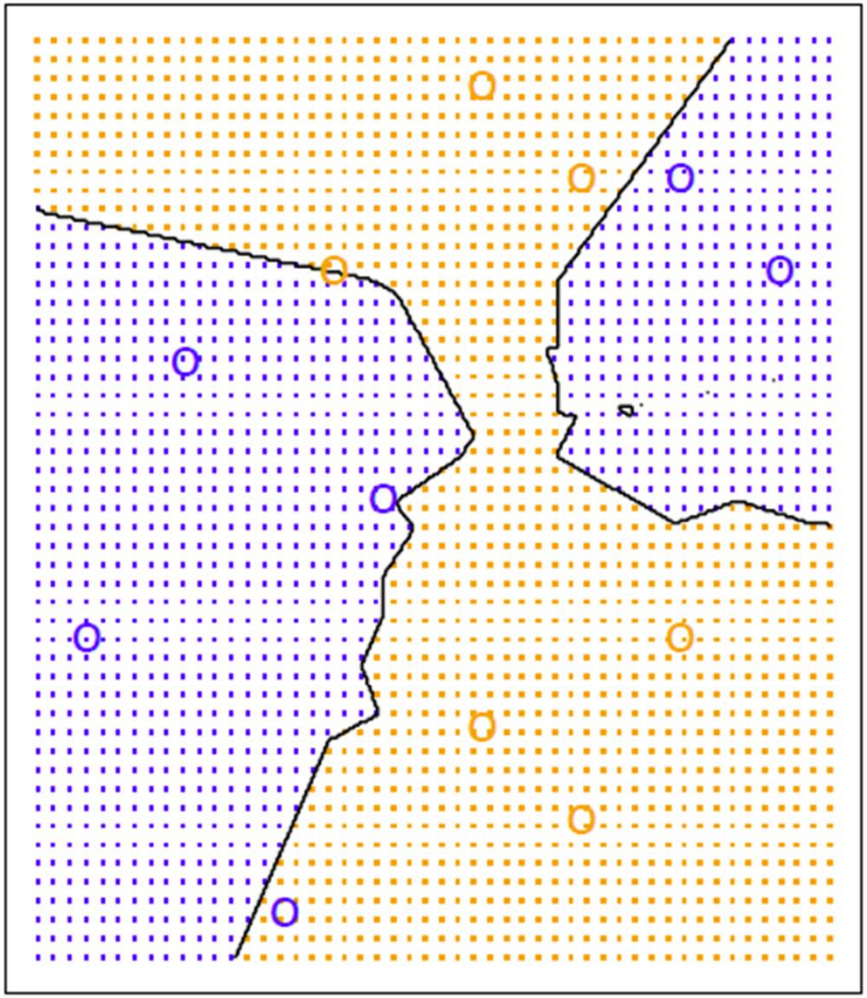
- **KNN 식으로 말하자면** : ♣ 에서 가장 가까운 K개의 observation들을 추리고, 그 K개의 observation들을 클래스별로 집계해서 가장 많은 클래스에 ♣ 이 속한다고 함

- **KNN** 에서의 이슈 :
  - K를 몇으로 정하면 좋을까? (여기서 **K**는 앞 장의 k, 즉 카테고리 갯수가 아님)
  - 가까운 정도를 계산하려면 "거리" 계산이 필요한데, 어떤 방식으로 하면 좋은가?
  - predictor의 dimension p가 커서, 가령 30 이면 30개의 feature들이 다 같은 정도로 중요한가?
  - feature 들의 scale 차이가 크면? 가령 어떤 feature는 값이 0 ~ 1 사이이고, 또 다른 것은 -10000 ~ 20000 이면?
  - category 형 feature간 거리는 어떻게 계산?

KNN with K=3

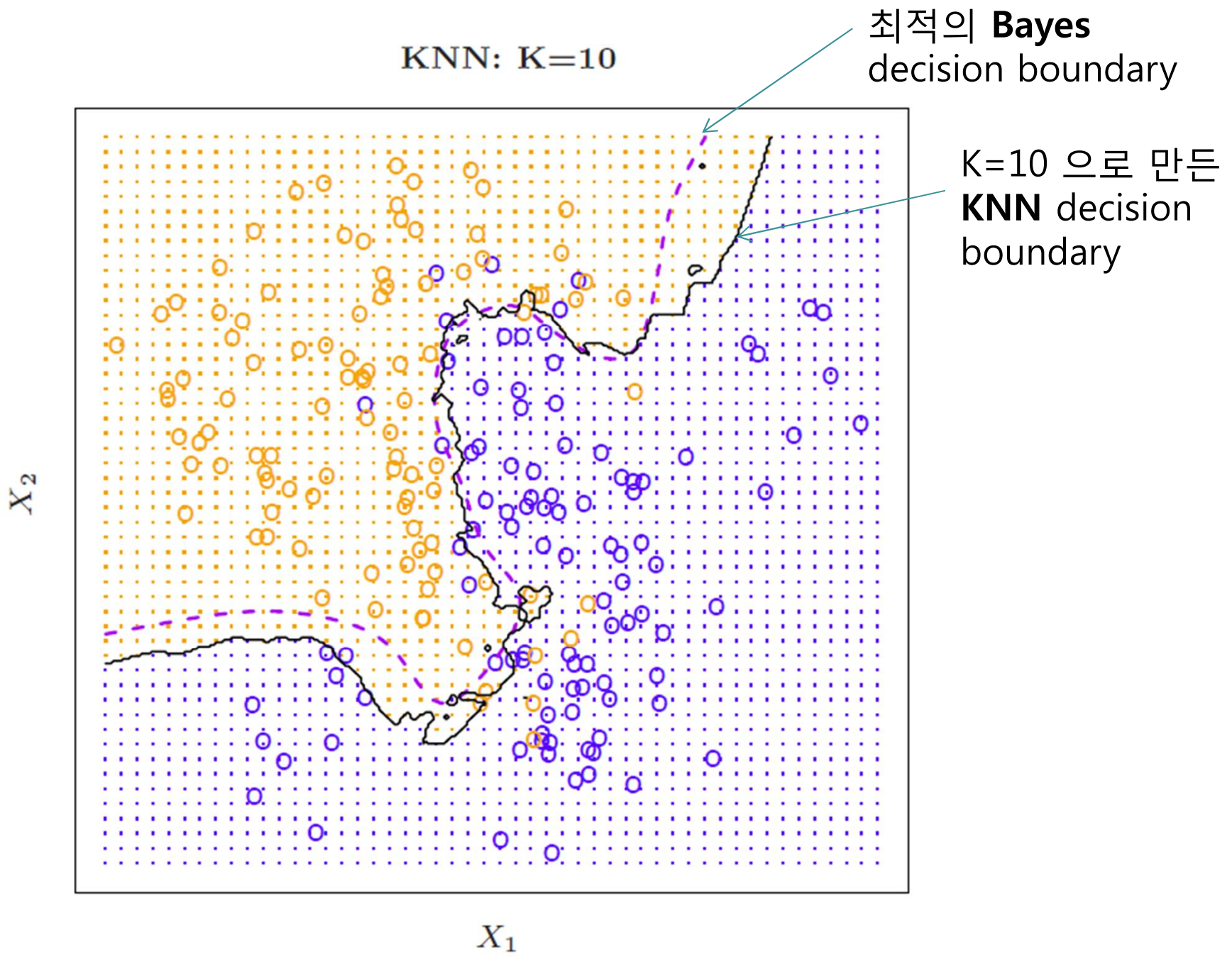
2 classes (yellow, blue) with n=12 traning samples





모든 (x1, x2)에 대해서 분류

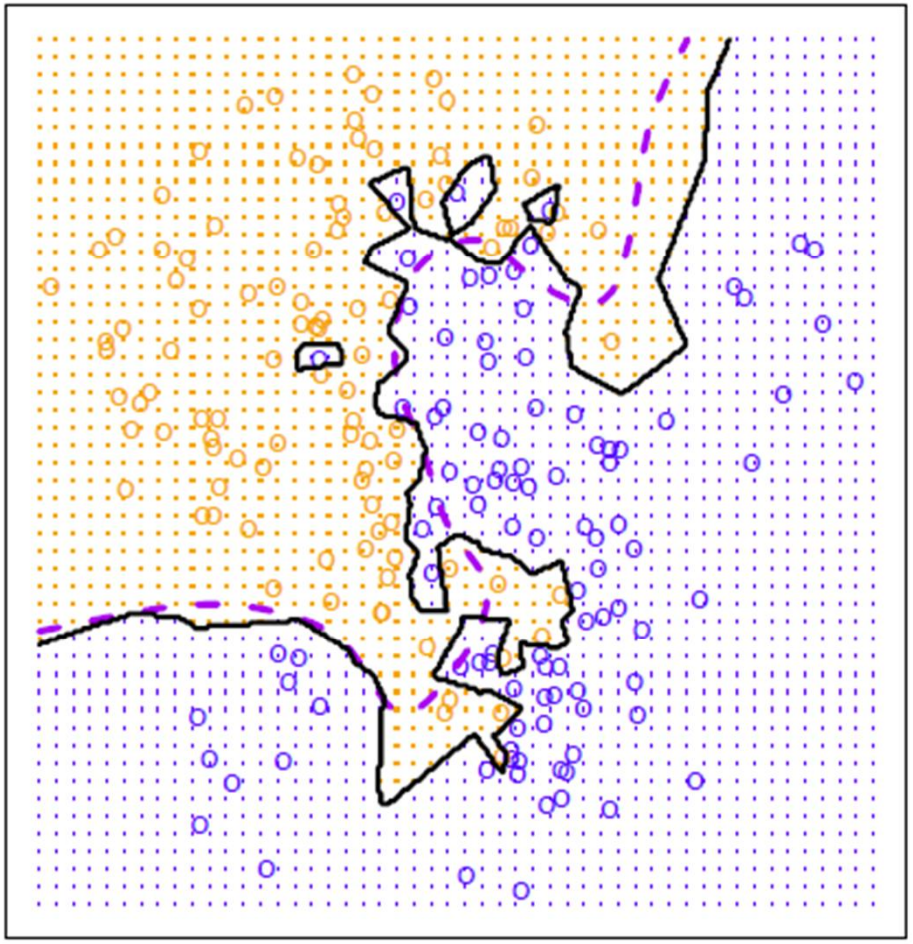
X 주변 가장 가까운 3개의 샘플을 보니 2개가 “파랑” 1개가 “노랑” 클래스. 따라서 X는 “파랑”으로 분류



최적의 **Bayes** decision boundary

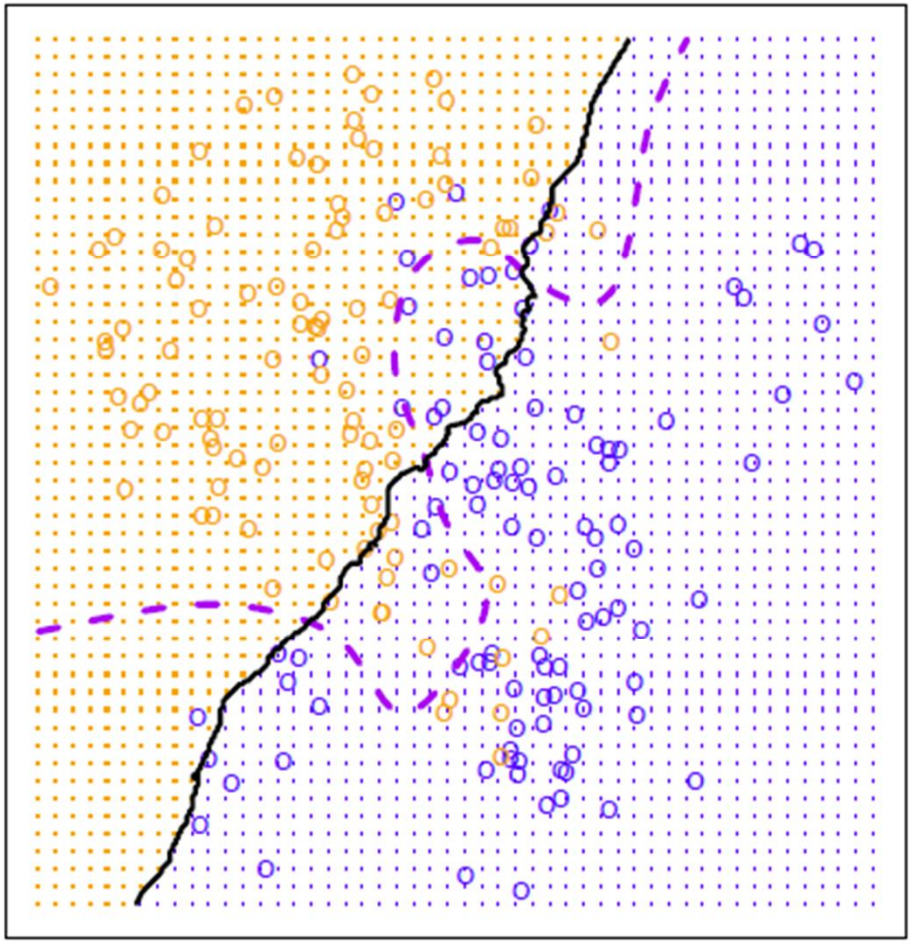
K=10 으로 만든 **KNN** decision boundary

KNN: K=1



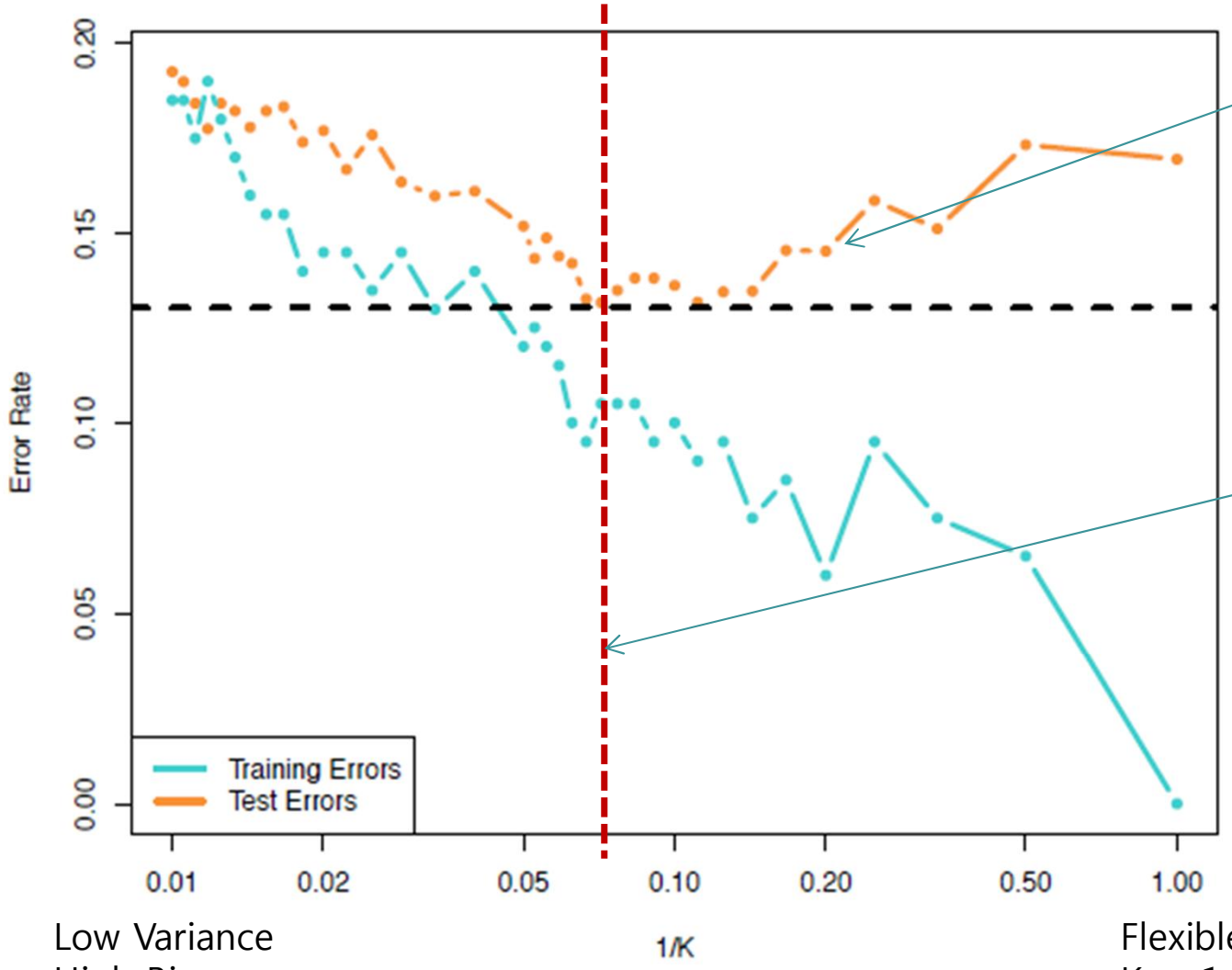
Overfit
(high variance)

KNN: K=100



Underfit
(high bias)

Training vs. Test Error Rates w.r.t. Model Complexity



K가 작아짐에 따라 Test error가 처음에는 감소하다가 다시 증가

1/K = 0.075, 즉 K=14 정도가 적당한 것 같음

Low Variance
High Bias
낮은 Complexity

Flexible 증가
K → 1
High variance
Overfit
Low Bias

K-Nearest Neighbors Regression

KNN Regression

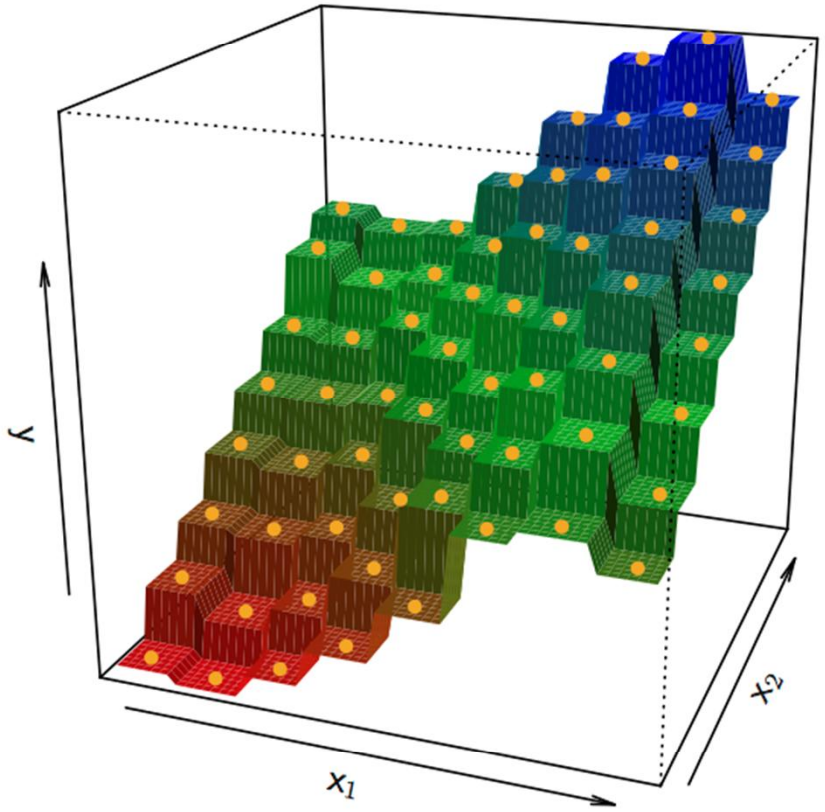
- KNN Regression은 KNN classifier와 유사

- Predictor가 X=x_0 일때, Y 추정치 $\hat{f}(x_0)$를 추정하려면 x_0 에서 가장 가까운 K개의 observation들을 training set에서 식별해 이를 $\mathcal{N}_0$라 한다. 그리고는, $\mathcal{N}_0$ 에 속한 observation 들의 response 들을 구해 그것들의 평균을 구한다. 즉,

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i$$

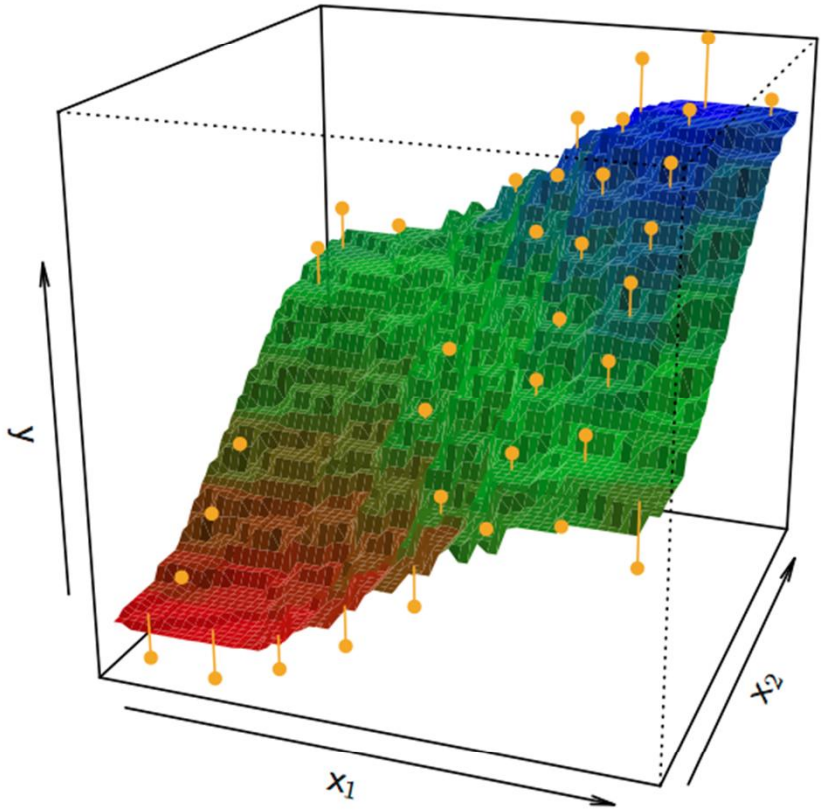
- KNN regression은 f가 어떤 "특정한 식"인 것을 가정하지 않으므로 **Non-Parametric**

K = 1



- High Variance
- 실제 Test Data 에서는 안좋은 결과를 낼 수 있음

K = 9



- 괜찮아 보임

Parametric vs. Non Parametric

- 어떤 경우에 KNN과 같은 Non parametric 방법을, 또는 Linear Regression과 같은 Parametric 모델을 쓰면 좋은가?

- Parametric 모델에서 우리가 가정(선택)한 식이 **실제 f와 비슷하면** parametric 접근이 더 나은 결과를 보인다

- Feature dimension p가 크지 않고, 가령 $p < 10$, training observation 개수가 충분히 크고, 가령 $n > 10000$, 실제 f가 non linear 하면 적당한 K를 써서 parametric 경우보다 KNN으로 더 나은 결과를 보일 수 있다

- 한편, p가 크고 n이 작으면 실제 f가 linear가 아니더라도 KNN의 결과가 좋지 않다 – **curse of dimensionality**