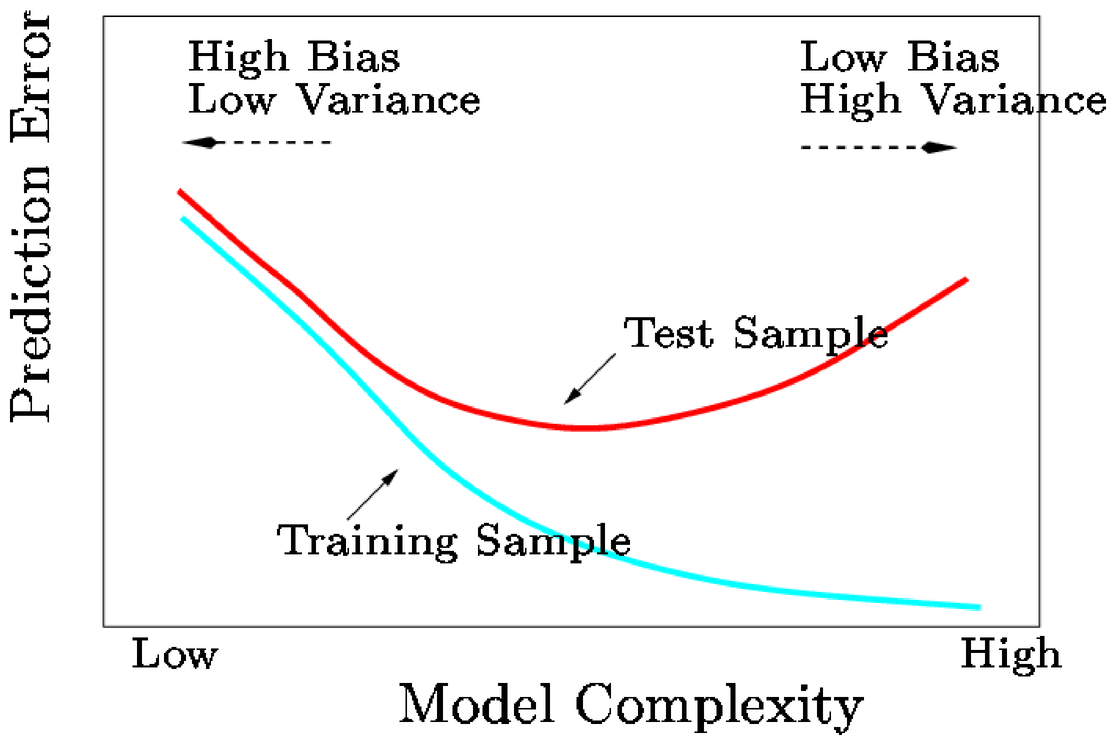
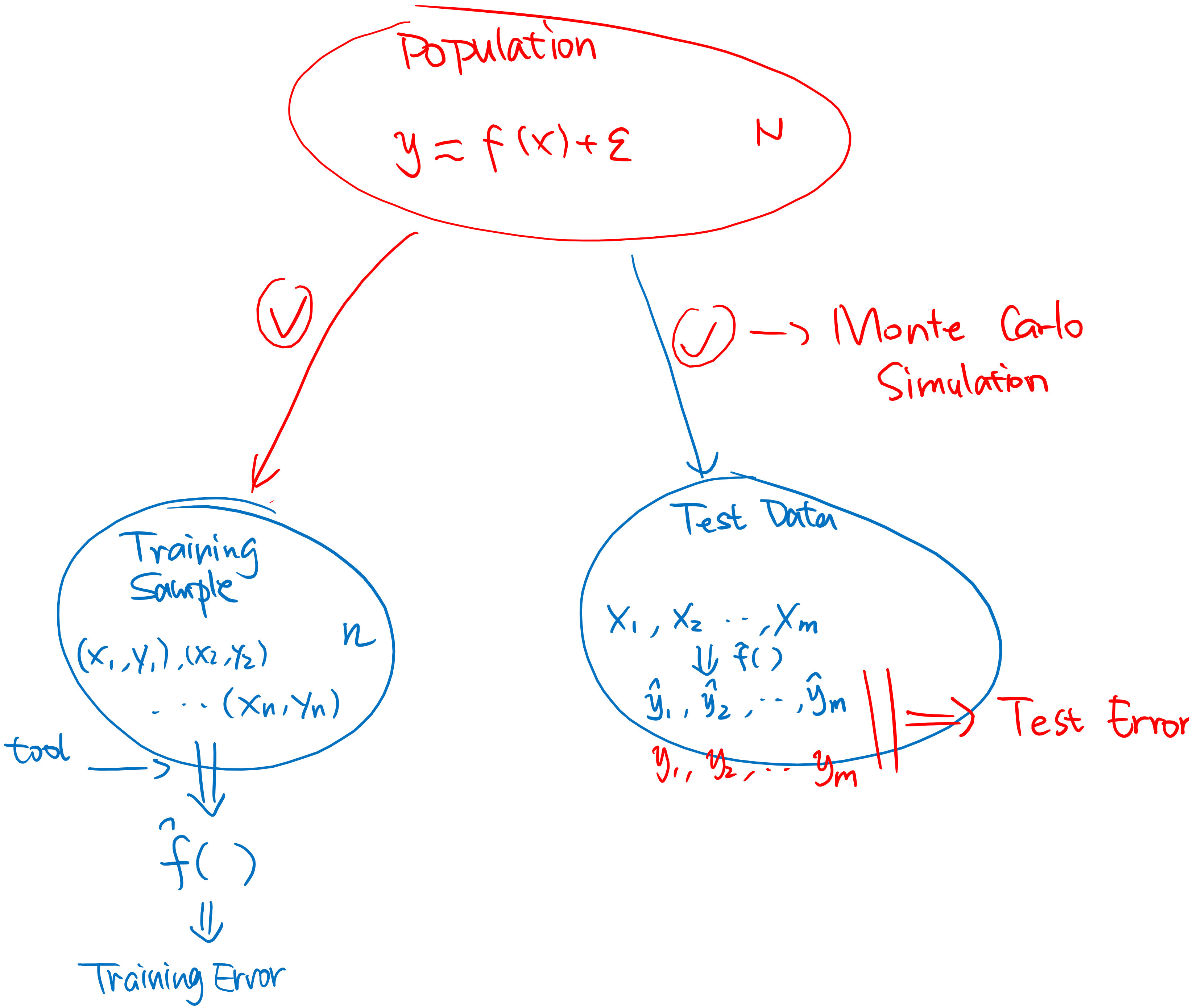
DSC5103 Statistics

Session 2. The K-Nearest Neighbor Algorithm

Last time

- Out-of-sample prediction performance as the correct measure
- The Bias-Variance decomposition and trade-off



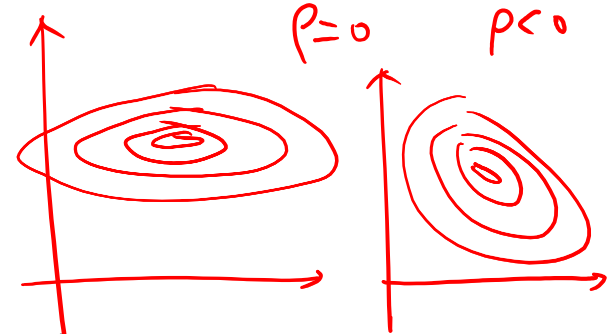

Population
$y = f(x) + \varepsilon$
N
✓
✓ → Monte Carlo Simulation
Training Sample
n
$(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$
tool
$\hat{f}(\ )$
Training Error
Test Data
$x_1, x_2 \ldots, x_m$
$f(\ )$
$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m$
$y_1, y_2, \ldots, y_m$
Test Error

Simulation Fundamentals

f( )

- Play the God's role and generate data with a known mechanism
- Stochastic models
  - Random variables, their distributions and parameters
  - The relationship among variables, dependency structures
- Monte Carlo simulation
  - Use computer to simulate outcomes of a stochastic model
  - To mimic the process of obtaining a sample from the population
  - By comparing the known population parameters and the estimates made from the sample, we can better evaluate our estimation methods

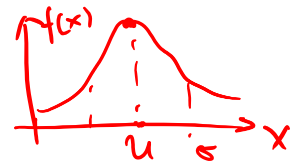
Simulation in R



- Simulating random variables with a known distribution
  - Random sampling   sample( )
  - Uniform $[min, max]$
  - Poisson λ (mean), $\{0, 1, 2, \cdots\}$
  - Binomial n, p $\{0, 1, 2, \cdots, n\}$
  - Normal μ, σ^2 $-\infty, +\infty$
  - Multivariate Normal

(μ_H, μ_W) $\begin{bmatrix} \sigma_H^2 & \rho\sigma_H\sigma_W \\ \rho\sigma_H\sigma_W & \sigma_W^2 \end{bmatrix}$

$\rho>0$



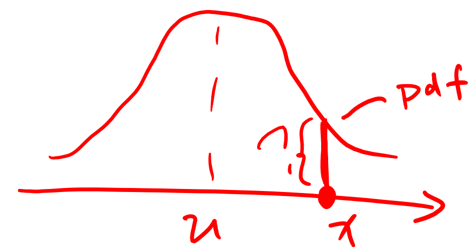
$\varepsilon \sim N(0, \sigma^2)$

- Simulating stochastic models
  - A 1-D linear model $y = \beta_0 + \beta_1 X + \varepsilon$ $\boxed{\beta_0}$, $\boxed{\beta_1}$ $\varepsilon \sim \underline{N}(0, \sigma^2)$, $\boxed{\sigma^2}$
  - A 2-D classification example $X \sim \underline{Unif}(a, b)$, $\boxed{a}$, $\boxed{b}$

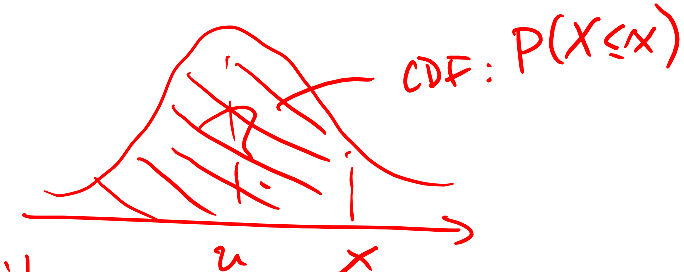
Prob. Functions in R: unif, binom, norm, pois, mvnorm

$\underline{r}\,\underline{\text{norm}}(n, \overset{\text{mean}}{\mu}, \overset{\text{sd}}{\sigma})$: Simulate n points from the dist

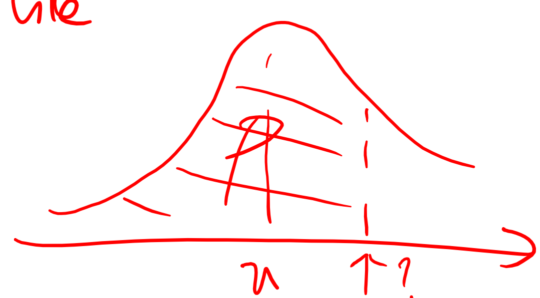
$\underline{d}\,\text{norm}(x, \mu, \sigma)$: PDF (Prob. density func)

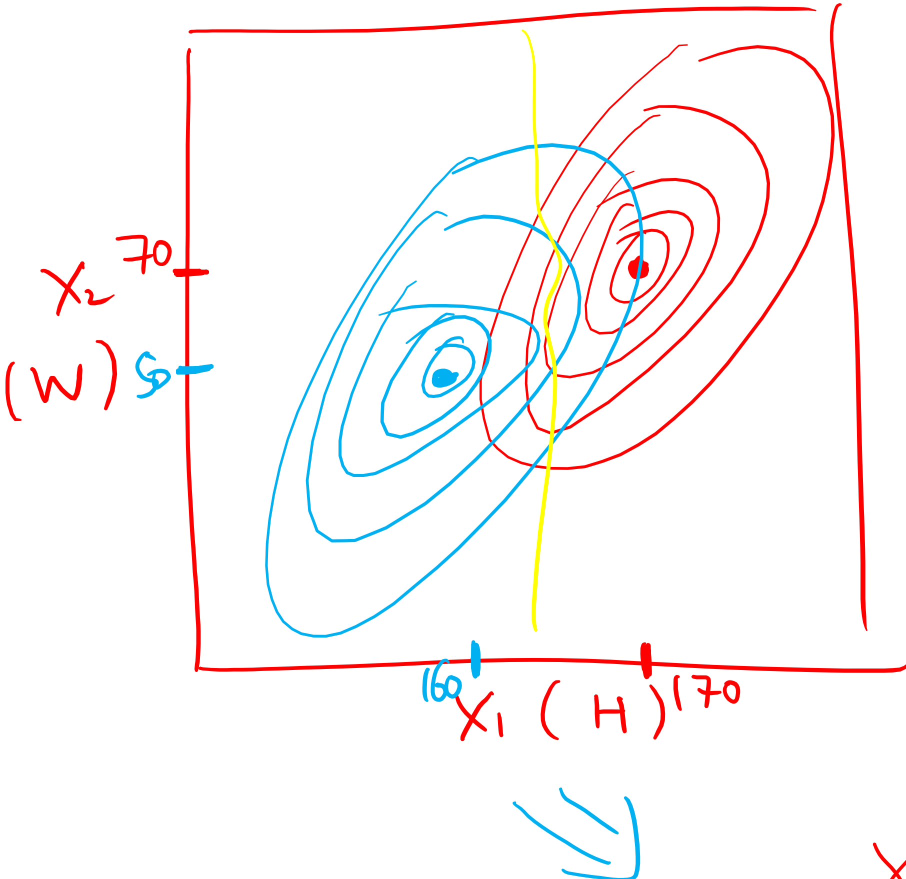


$\underline{p}\,\text{norm}(x, \mu, \sigma^2)$: CDF



$q\,\text{norm}(p, \mu, \sigma^2)$: quantile



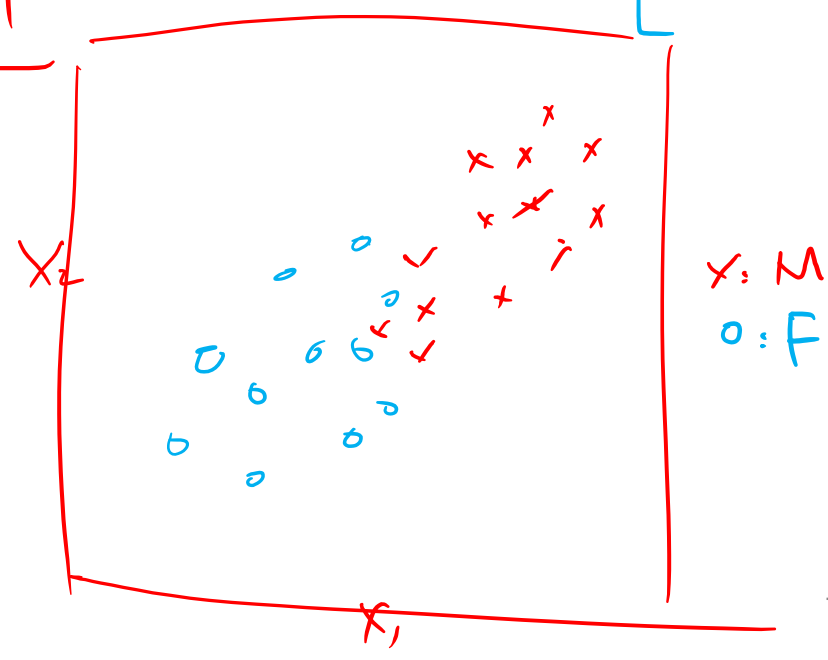


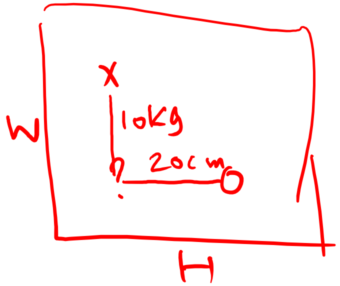
$y: \{M, F\}$
$\quad\;\; 0, 1$

$-1 \le \rho \le 1$

$M: \quad (u_1^M, u_2^M), \quad \begin{bmatrix} \sigma_{1m}^2 & \rho\sigma_{1m}\sigma_{2m} \\ \rho\sigma_{1m}\sigma_{2m} & \sigma_{2m}^2 \end{bmatrix}$

$F: \quad (u_1^F, u_2^F), \quad \begin{bmatrix} \cdot & \cdot \\ & \end{bmatrix}$





K-Nearest Neighbors (KNN)

- k Nearest Neighbors is a flexible *nonparametric* approach for both regression and classification.

- For any given *X*, we find the k closest neighbors to *X* in the training data, and examine their corresponding *Y*. We use
  - the **average** of the neighbors' *Y* as prediction for regression;
  - the **majority votes** of the neighbors' *Y* as prediction for classification.

    ⇒ proportion ⇒ prob

- The smaller that k is, the more flexible the method will be.

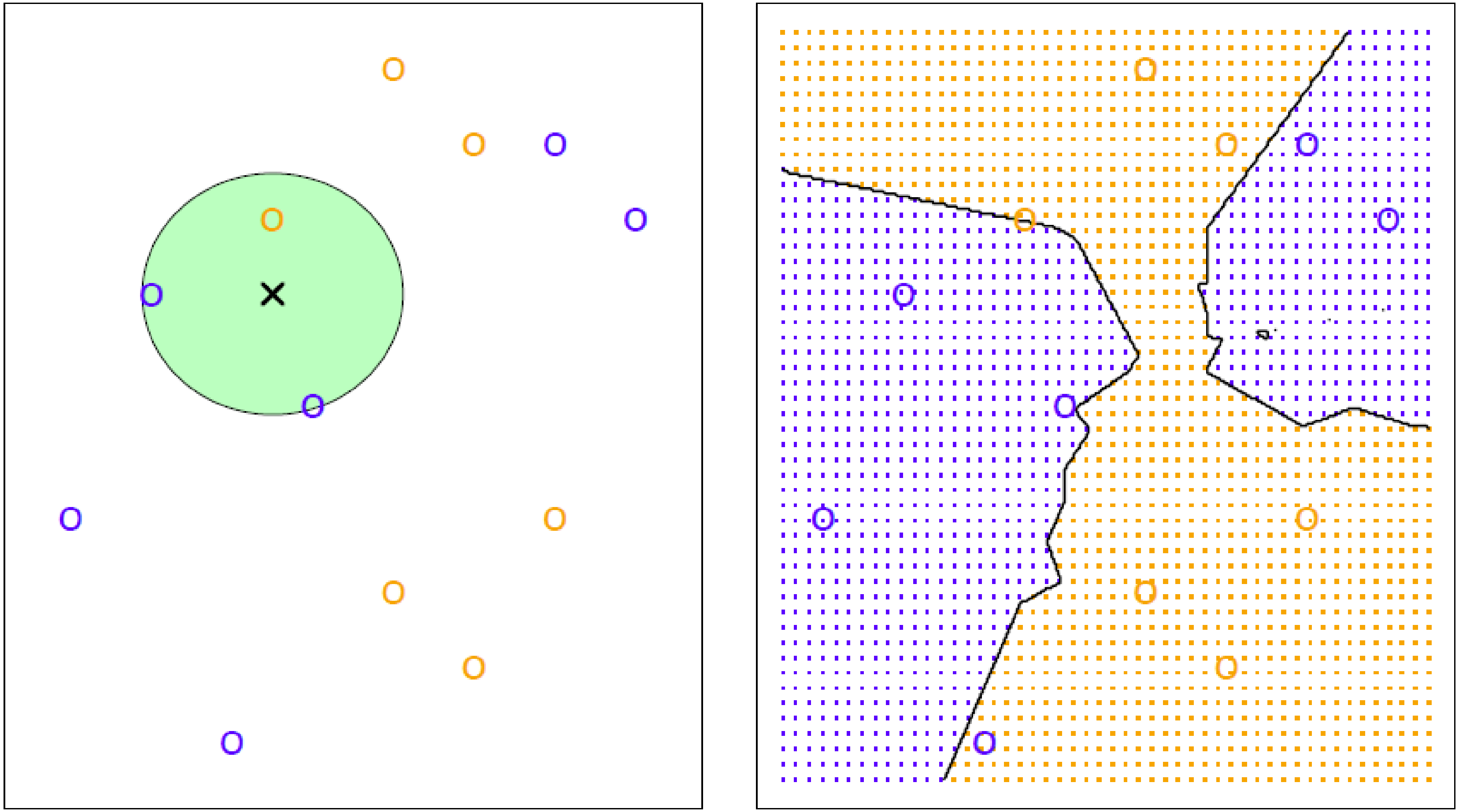
  (High Var, Low Bias) k=1 most flexible —— k^* —— K=n $(\hat{y} = \bar{y})$ (high Bias, low Var) Least flexible

- !!!BE CAREFUL!!!
  - How to define "closest"? How to measure distance in the space of *X*? ⇒ Normalize
  - Categorical X?
  - Dimensionality??
  - Variable selection???

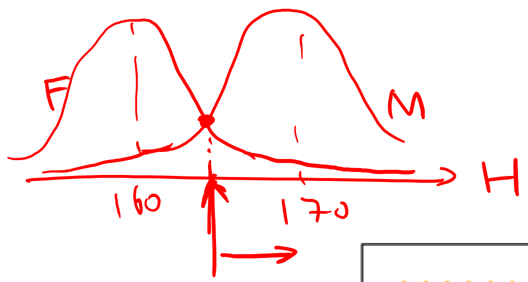
    ⇒ PCA

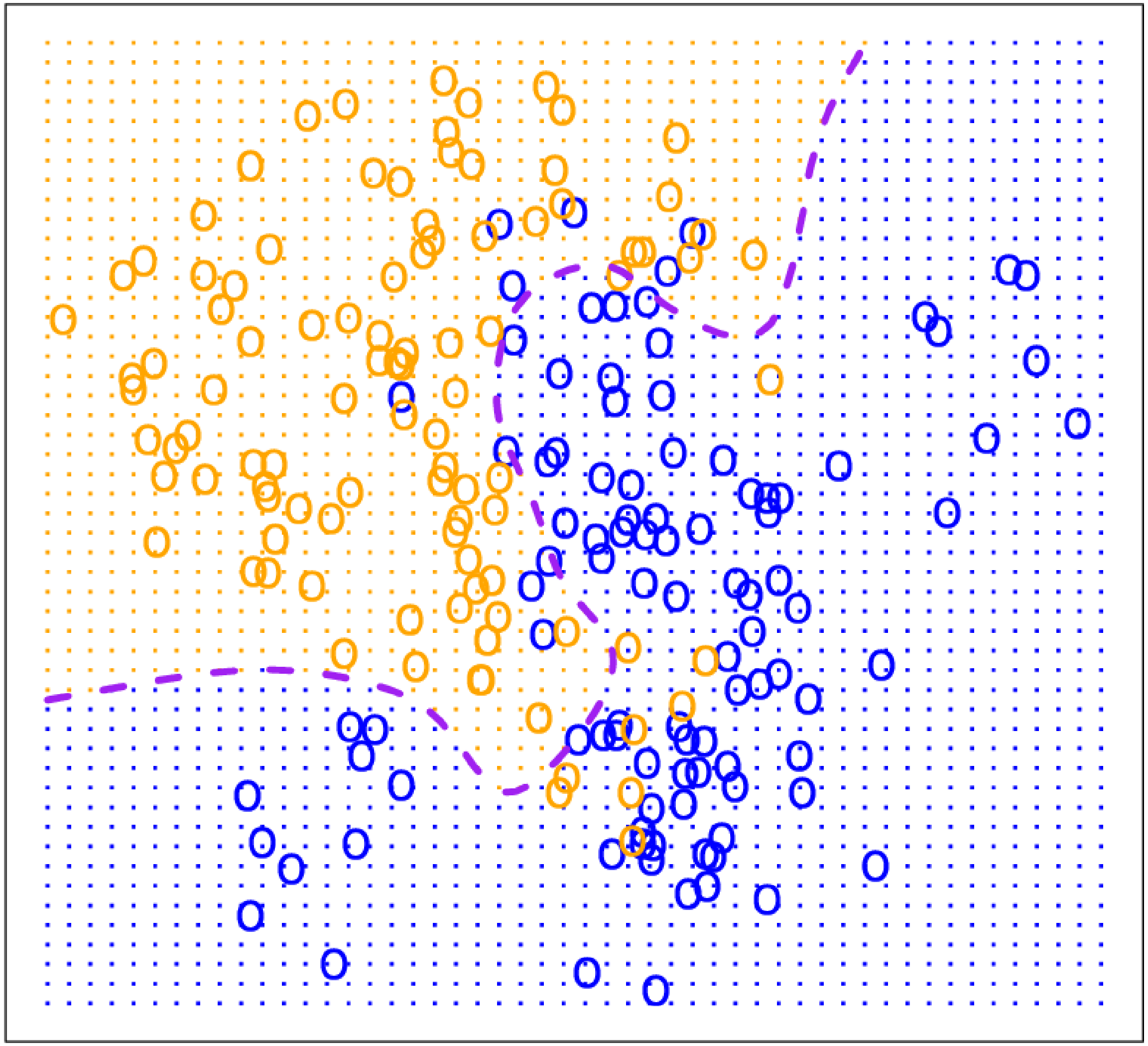
  + Capture Nonlinearity

KNN Example with k = 3

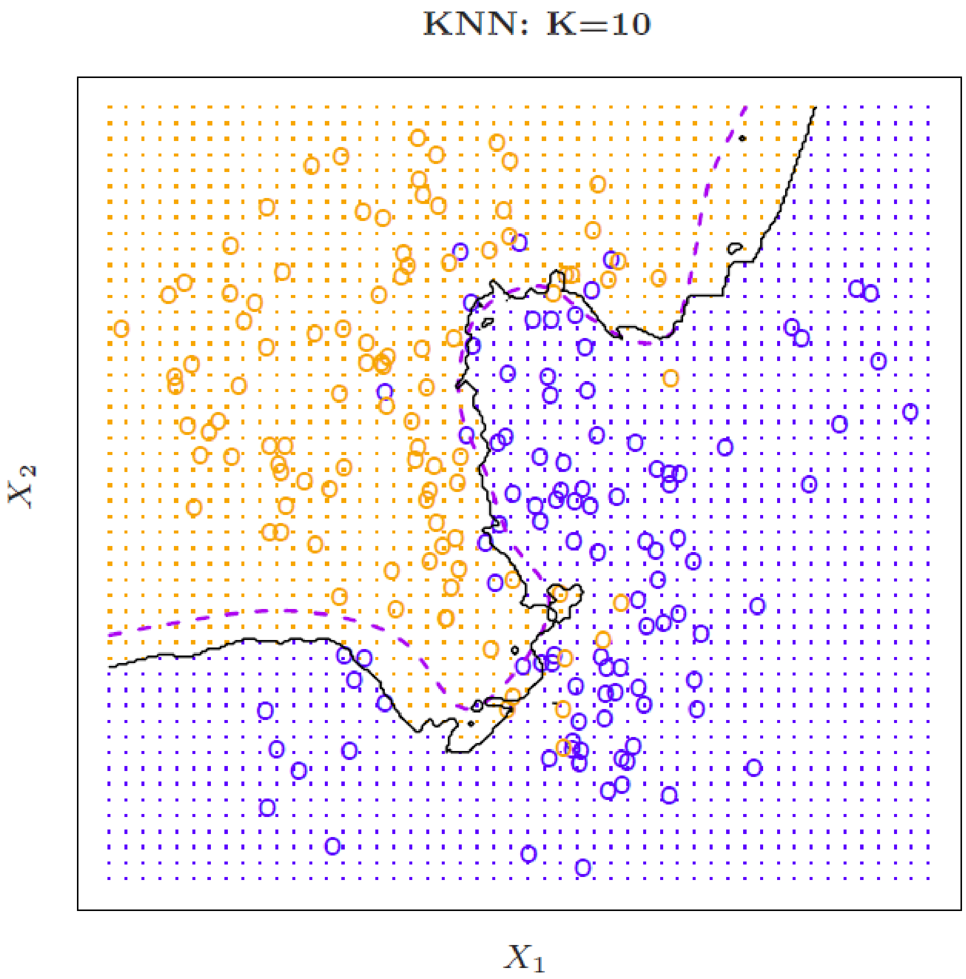


Optimal Classifier

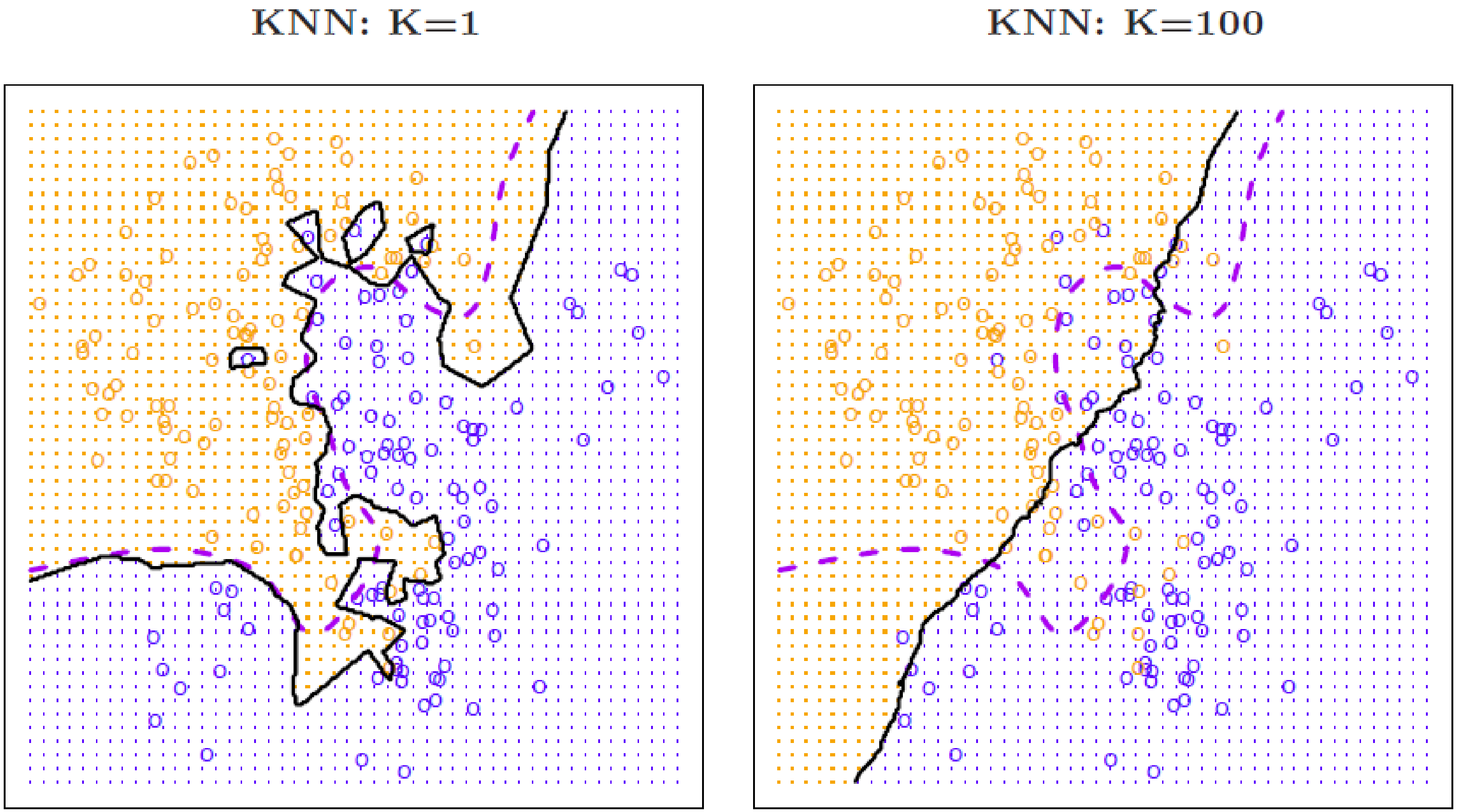




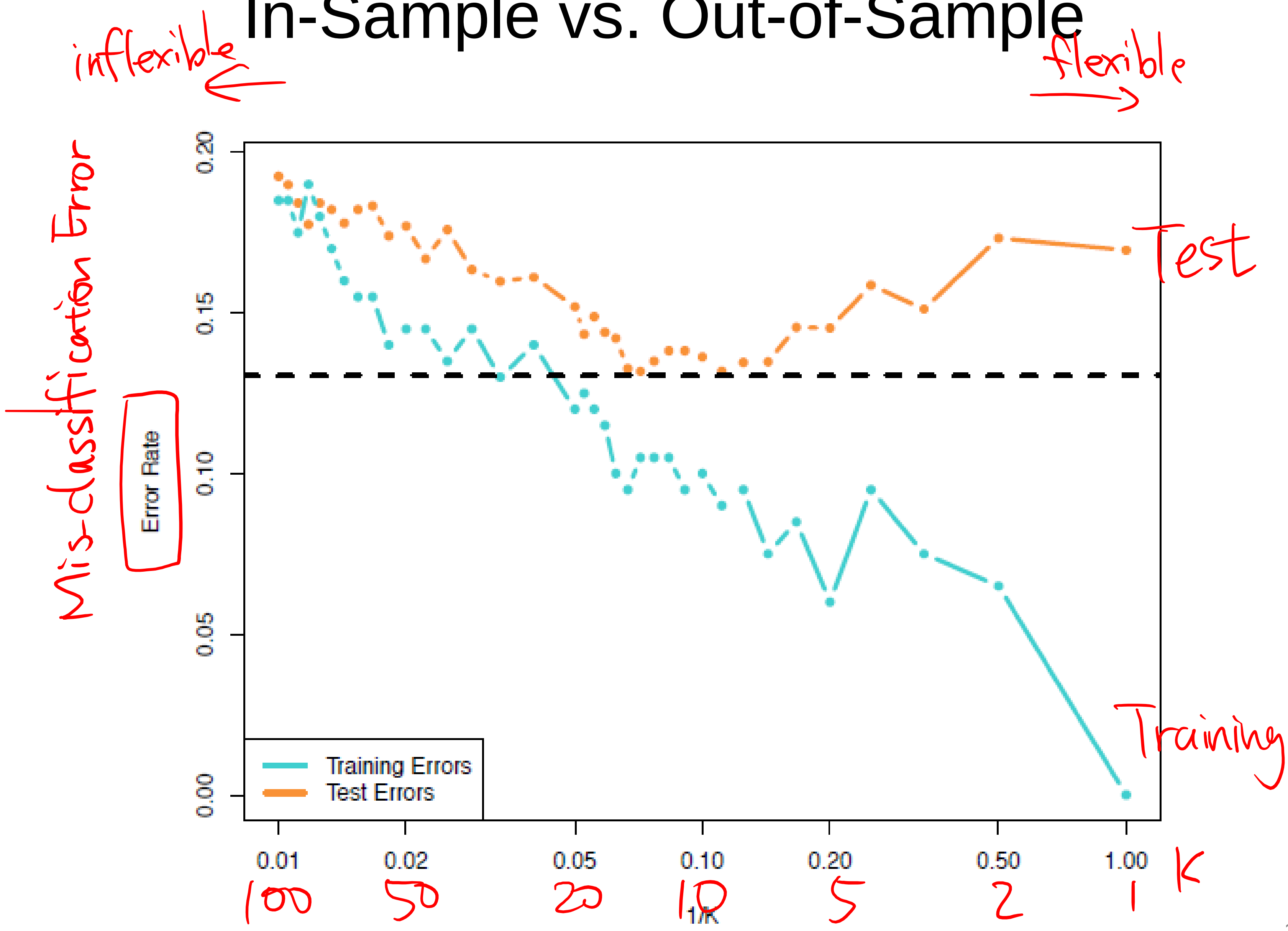
Simulated Data: K = 10



K = 1 and K = 100



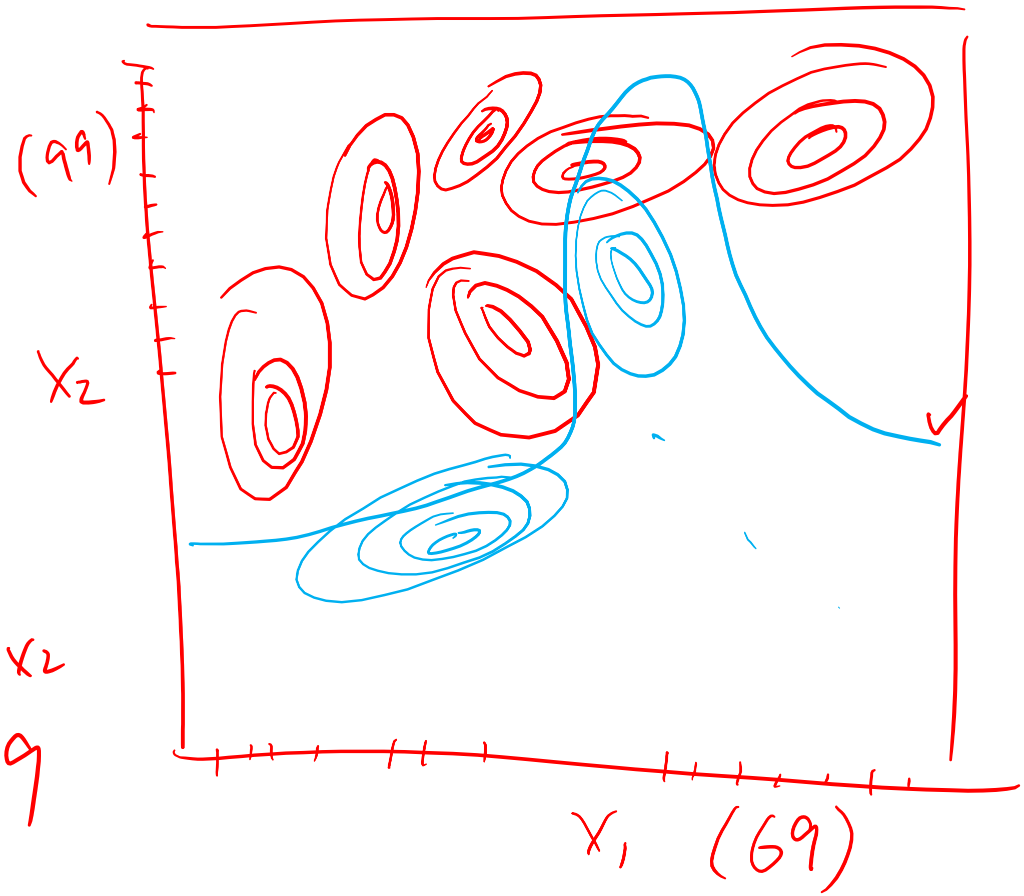
In-Sample vs. Out-of-Sample



KNN Demo in R

- A regression problem using KNN

- Applying KNN on the Mixture Example
  - https://web.stanford.edu/~hastie/ElemStatLearn/datasets/mixture.example.info.txt



$y: \{0, 1\}$

$P X_1$ $P X_2$

Grid: 69x99

6831

- Homework 1
  - Learn RMarkdown (http://rmarkdown.rstudio.com/)
  - Test the curse of dimensionality of the KNN algorithm

- Next time:
  - Linear Regression