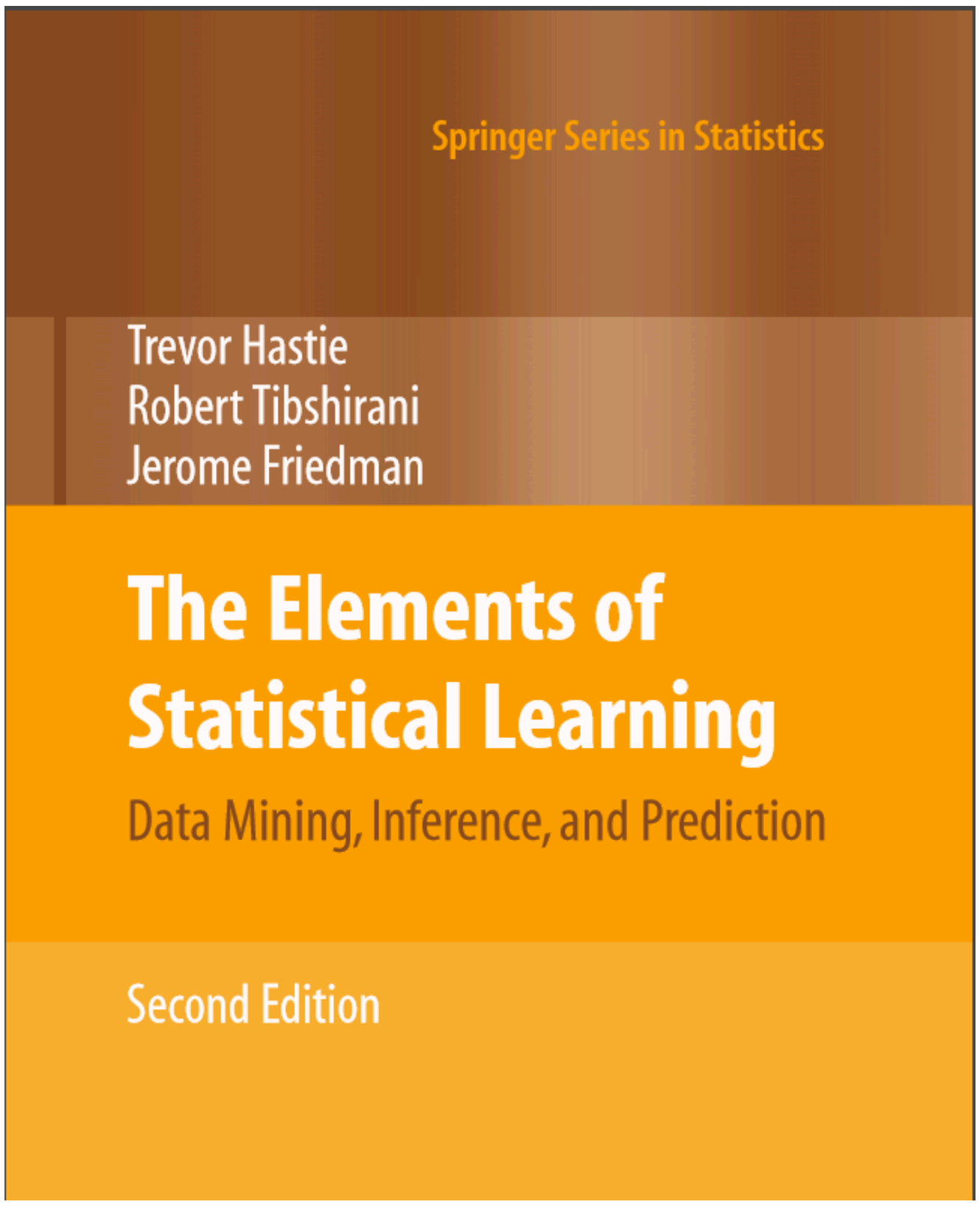
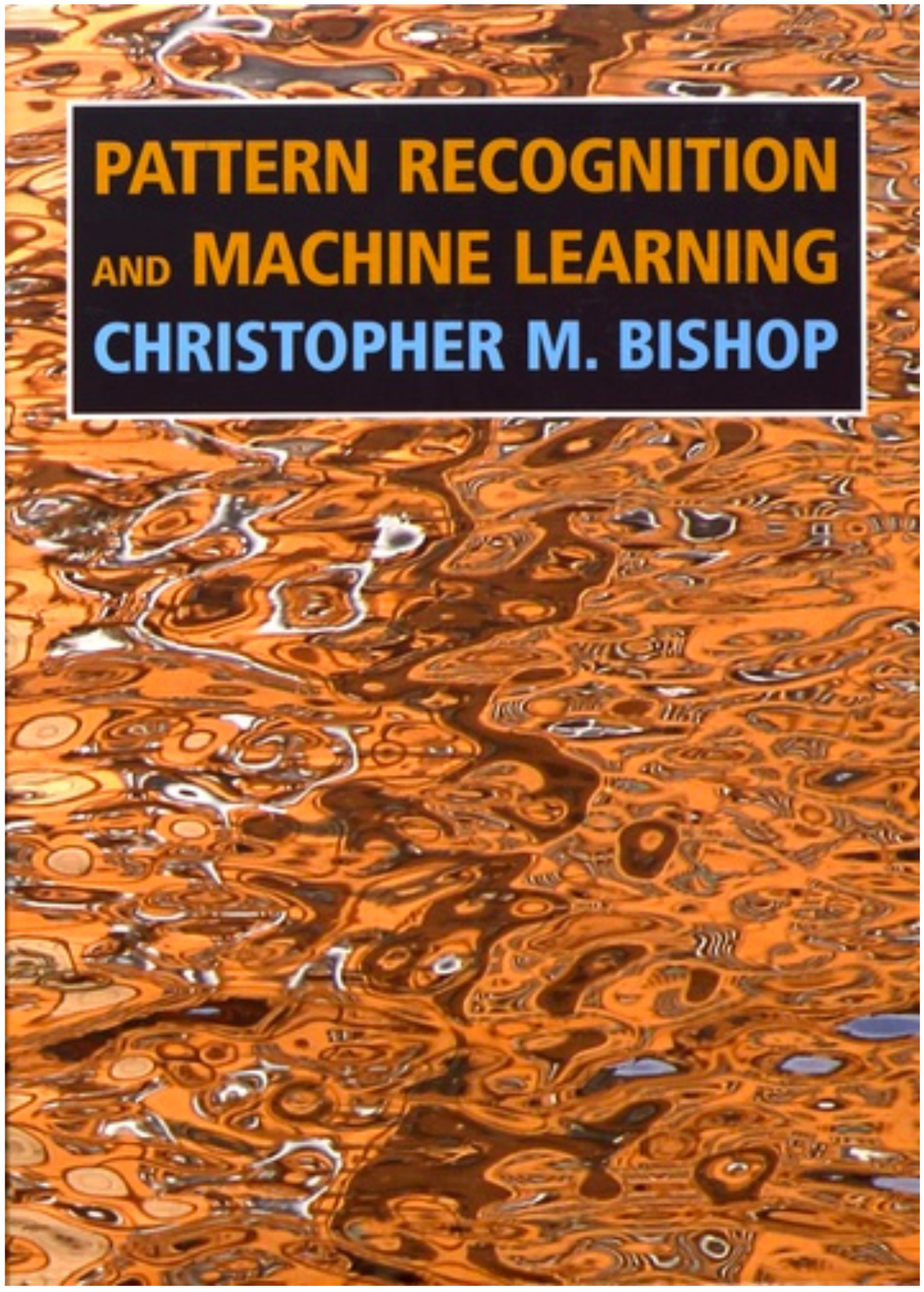
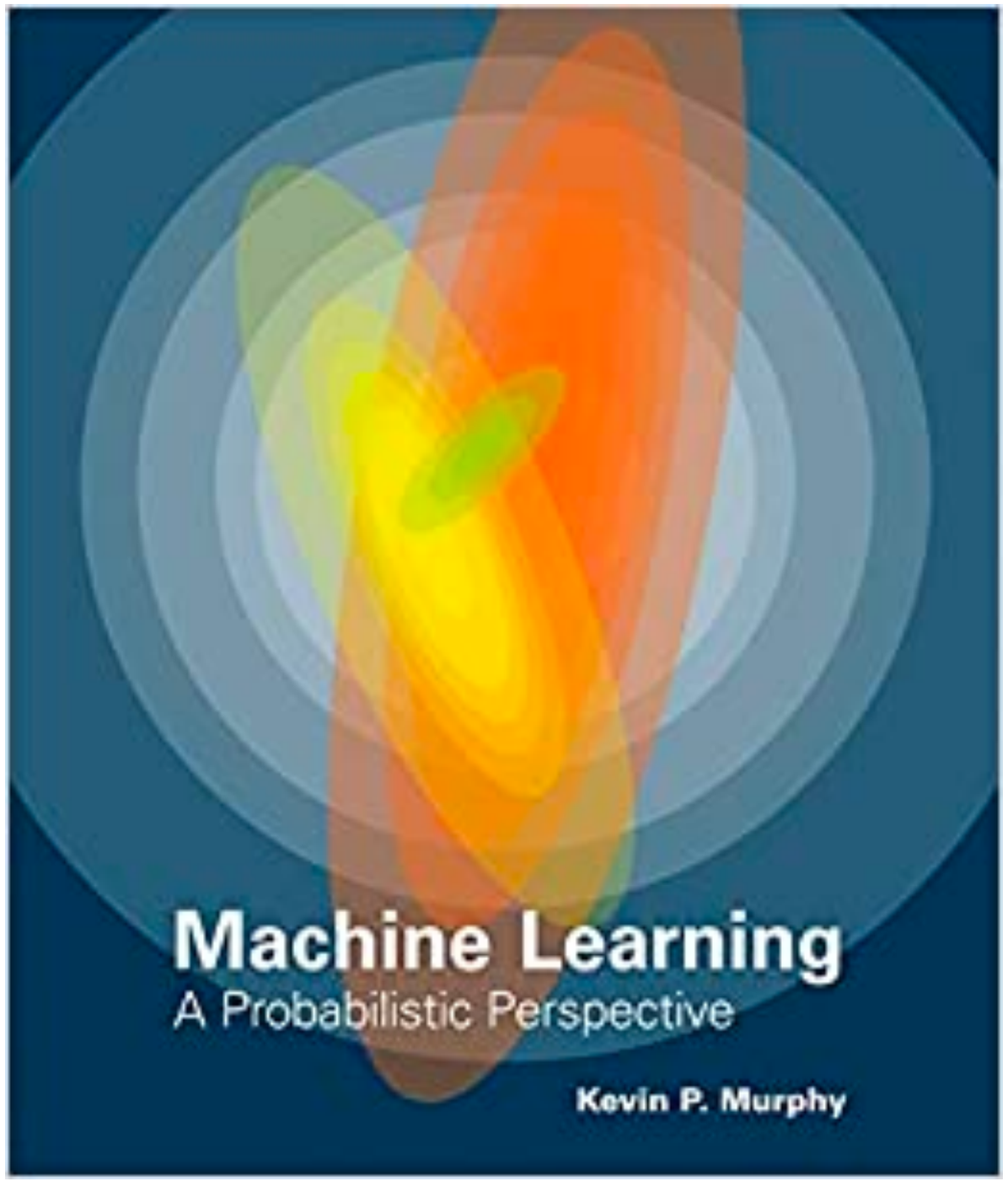
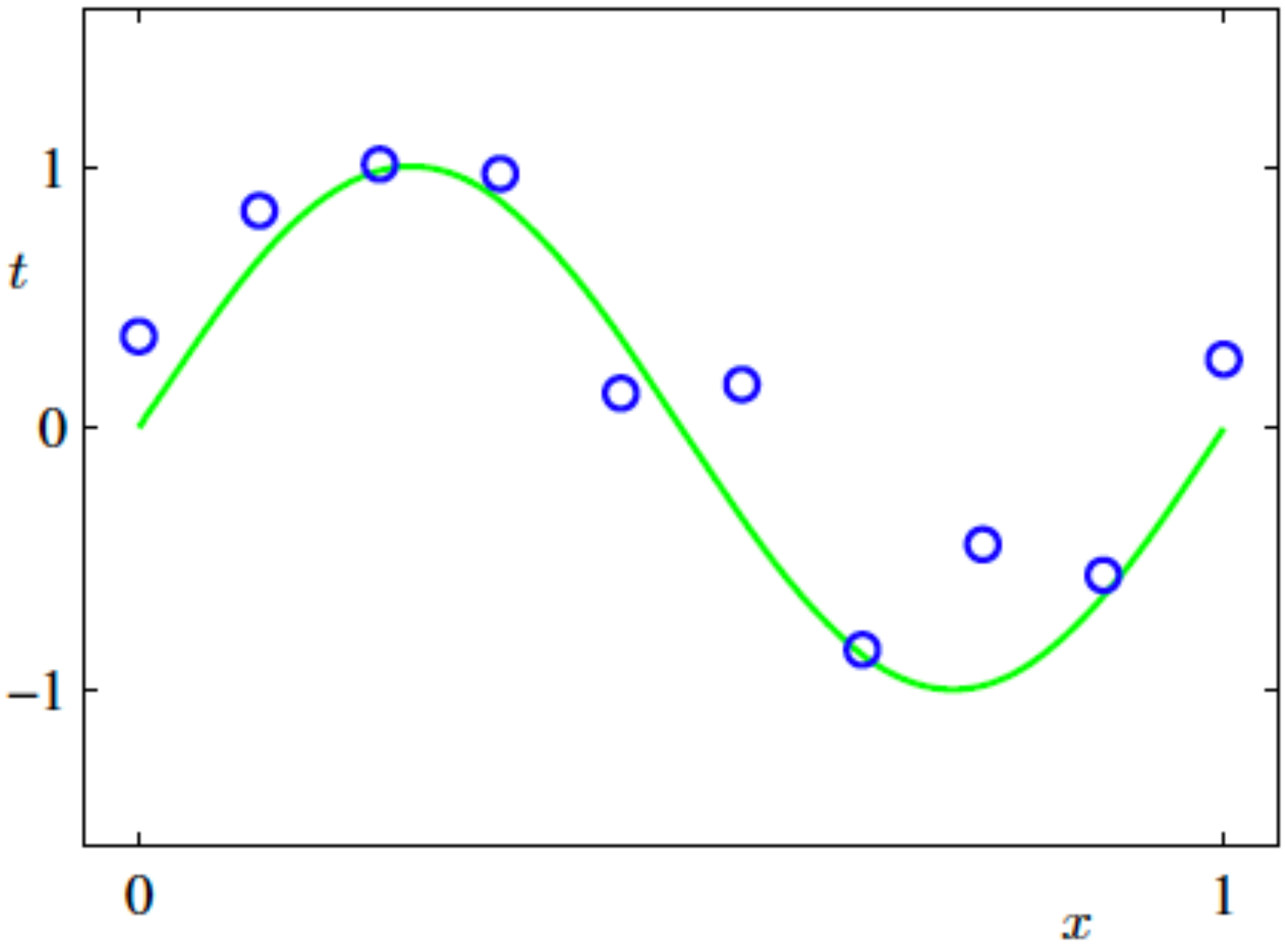
Машинное обучение



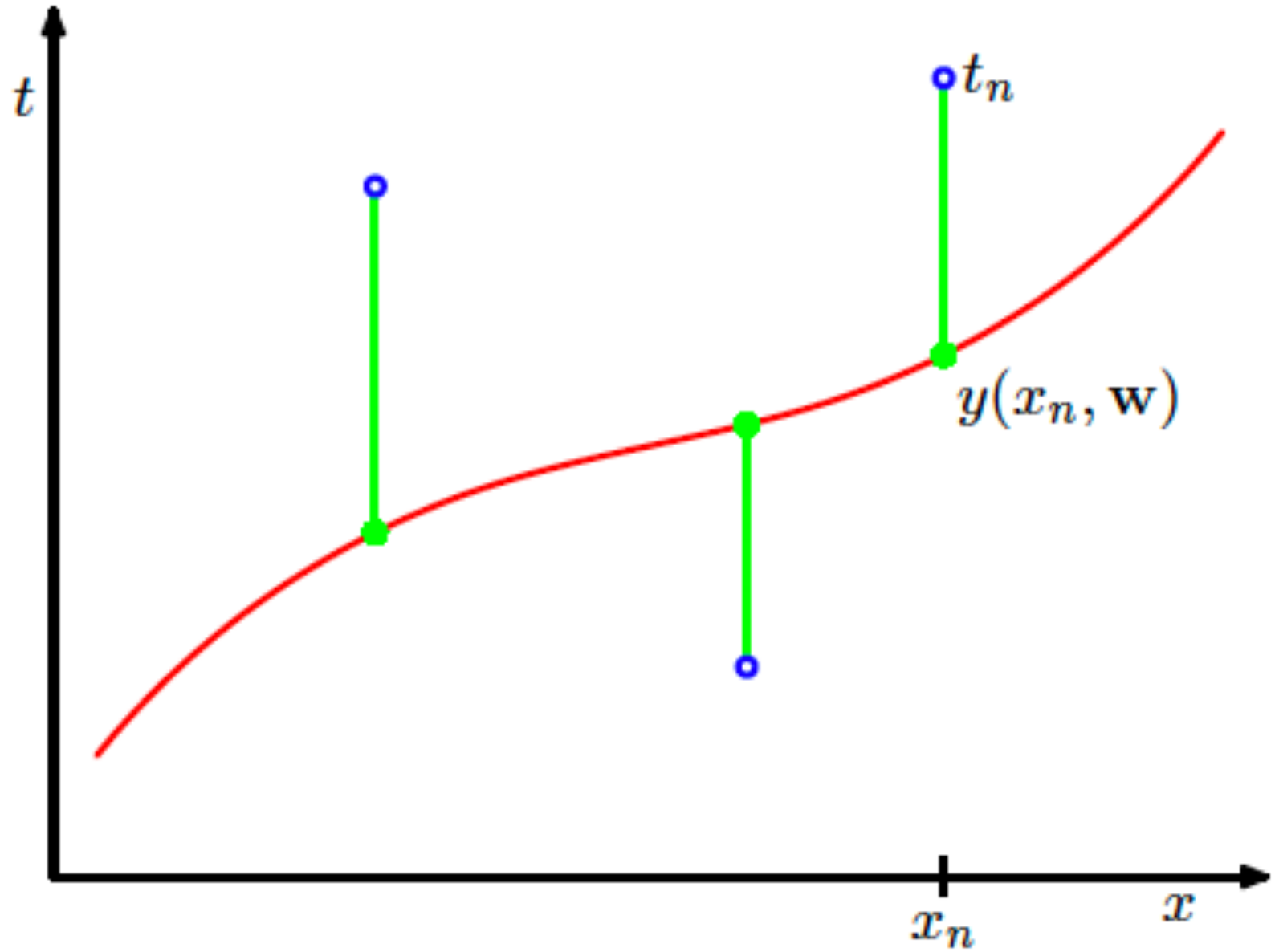

PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP


Machine Learning
A Probabilistic Perspective
Kevin P. Murphy

Пример задачи ML



Оцениваем ошибку

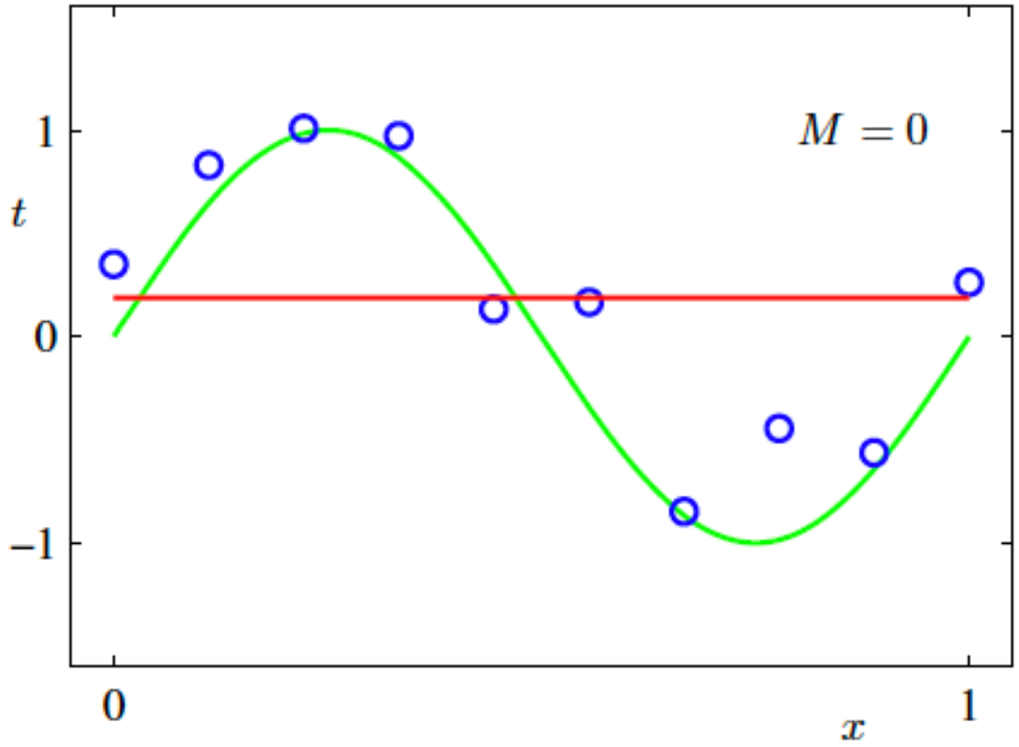


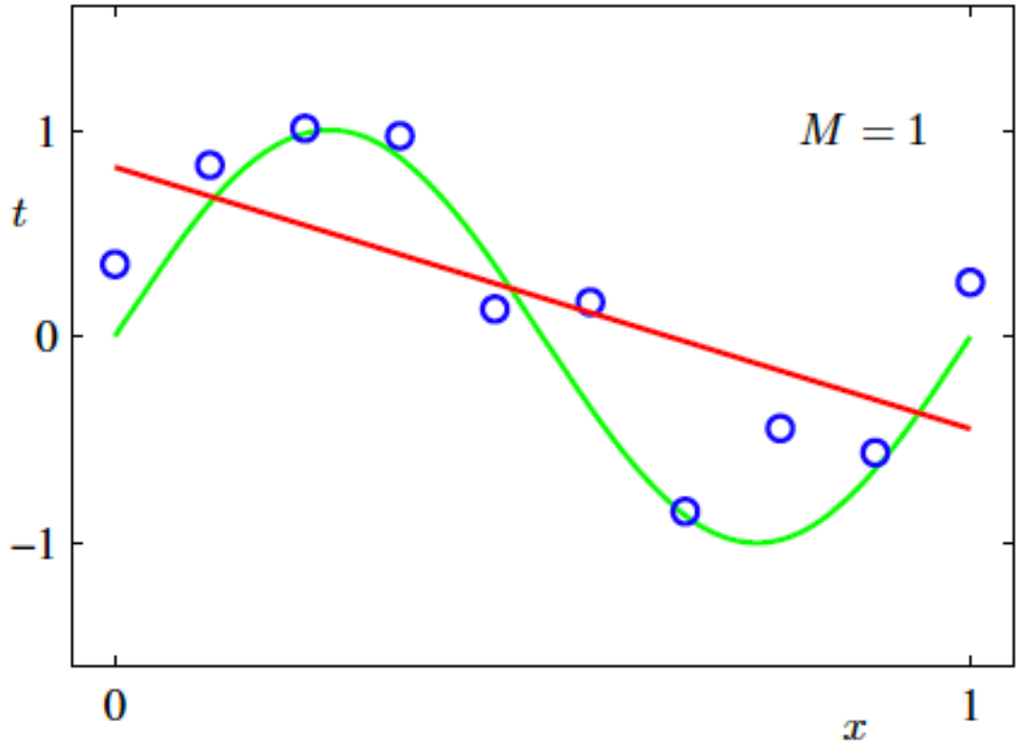
Оцениваем ошибку

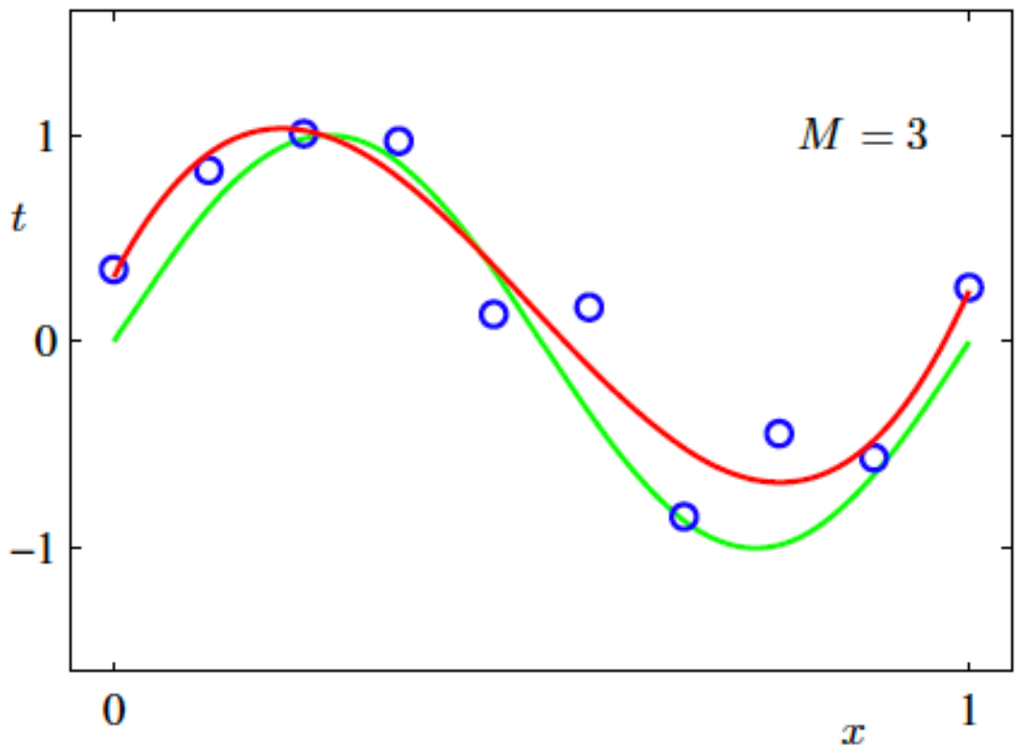
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

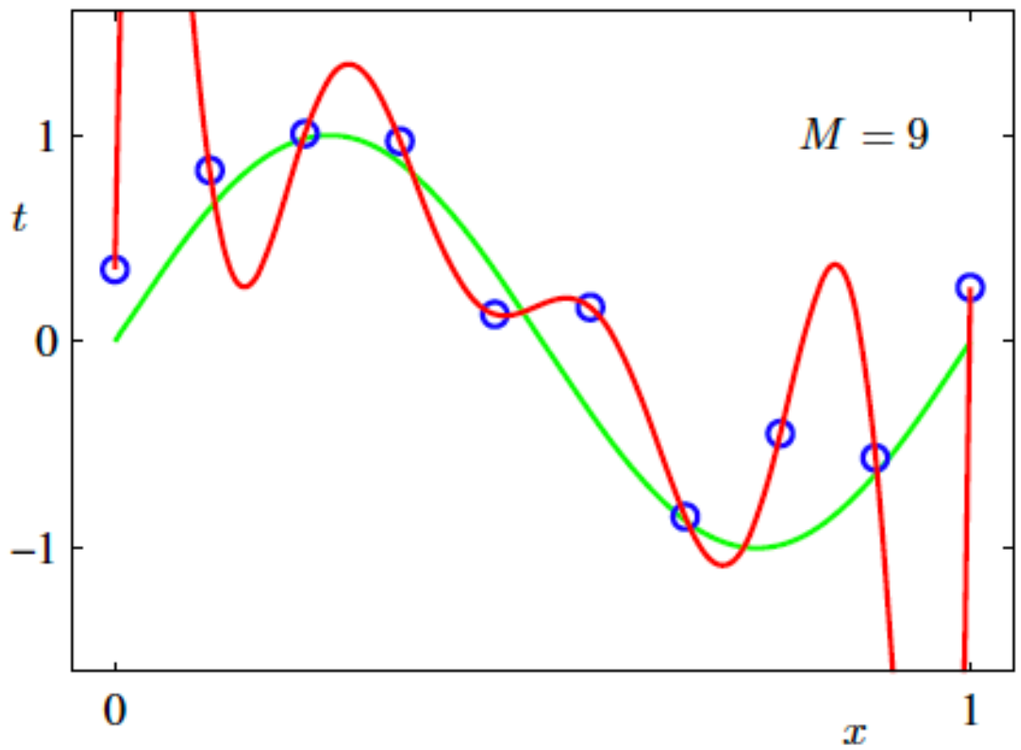
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

Недообучение vs Переобучение

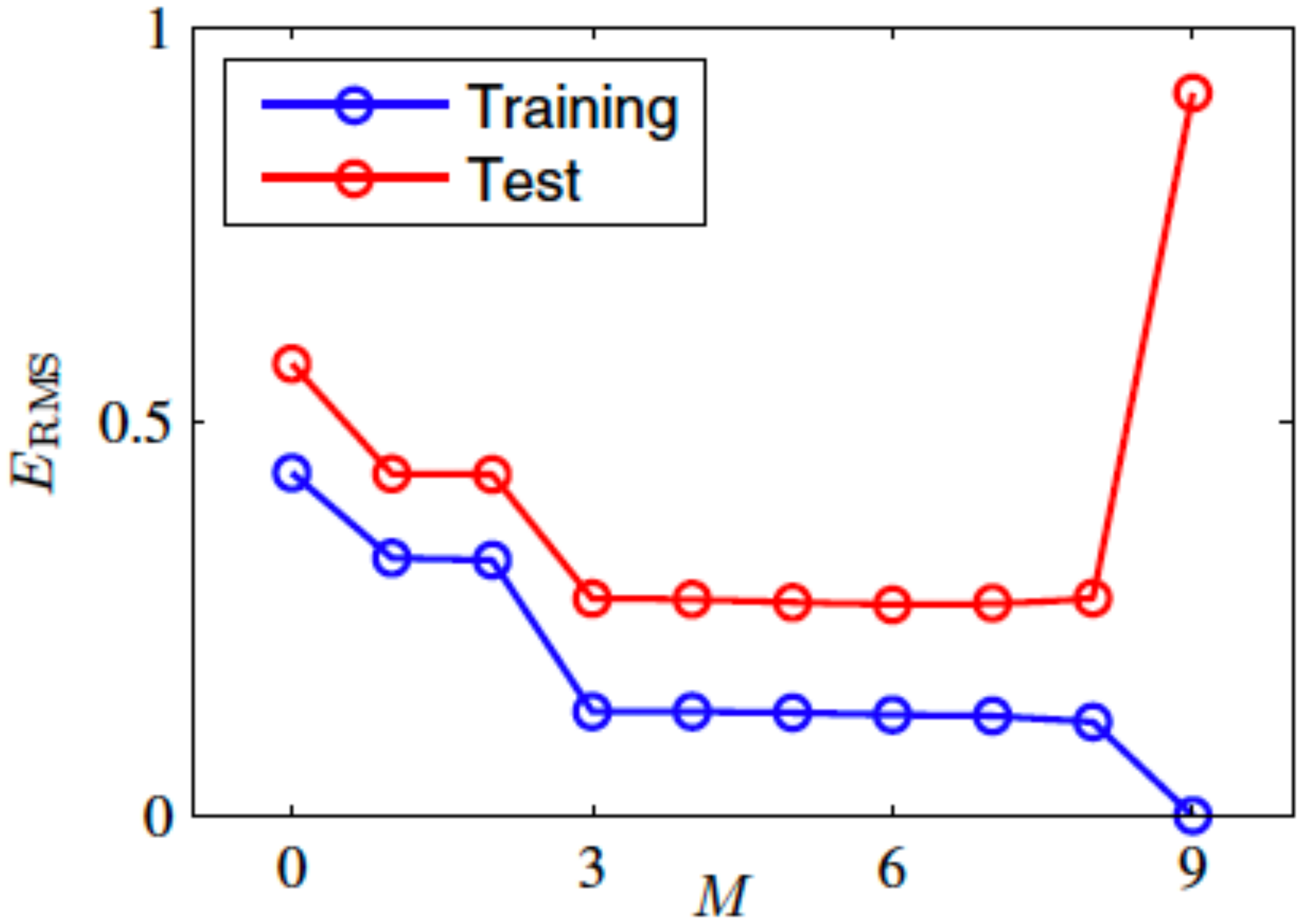








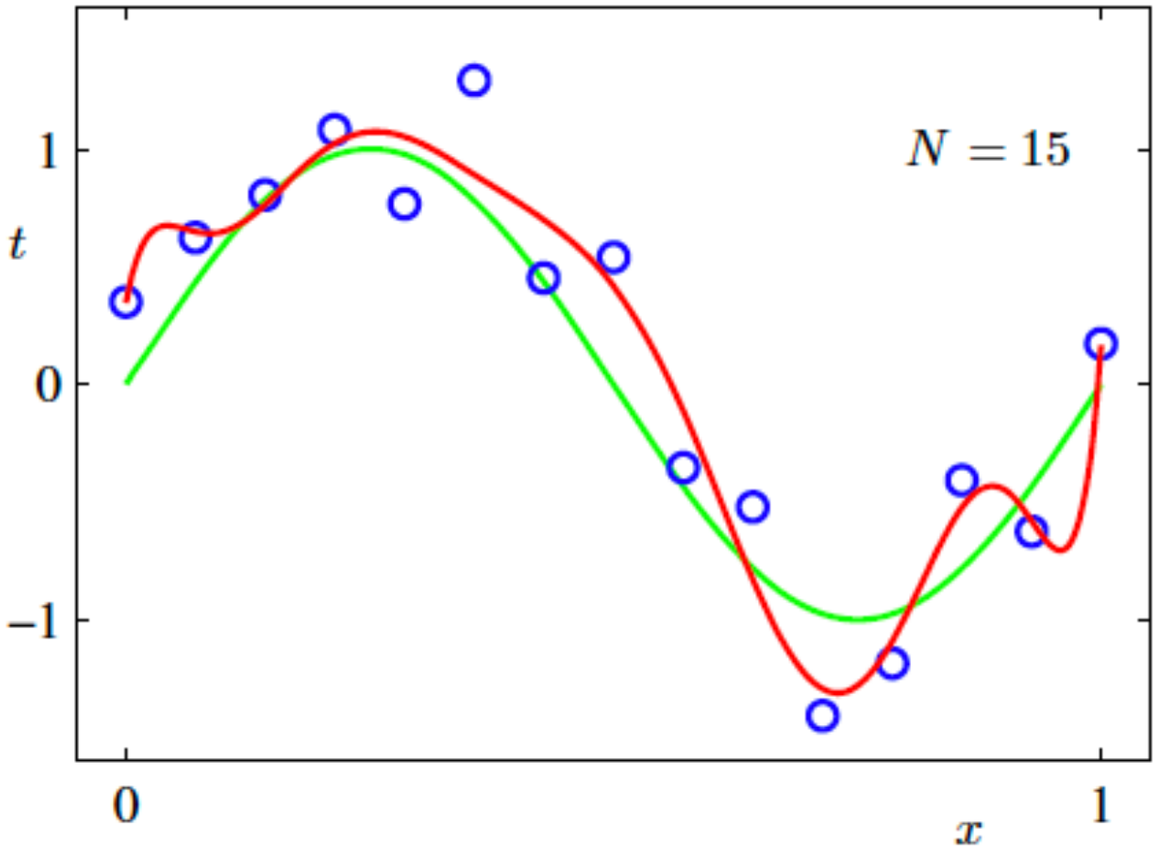
Недообучение vs Переобучение

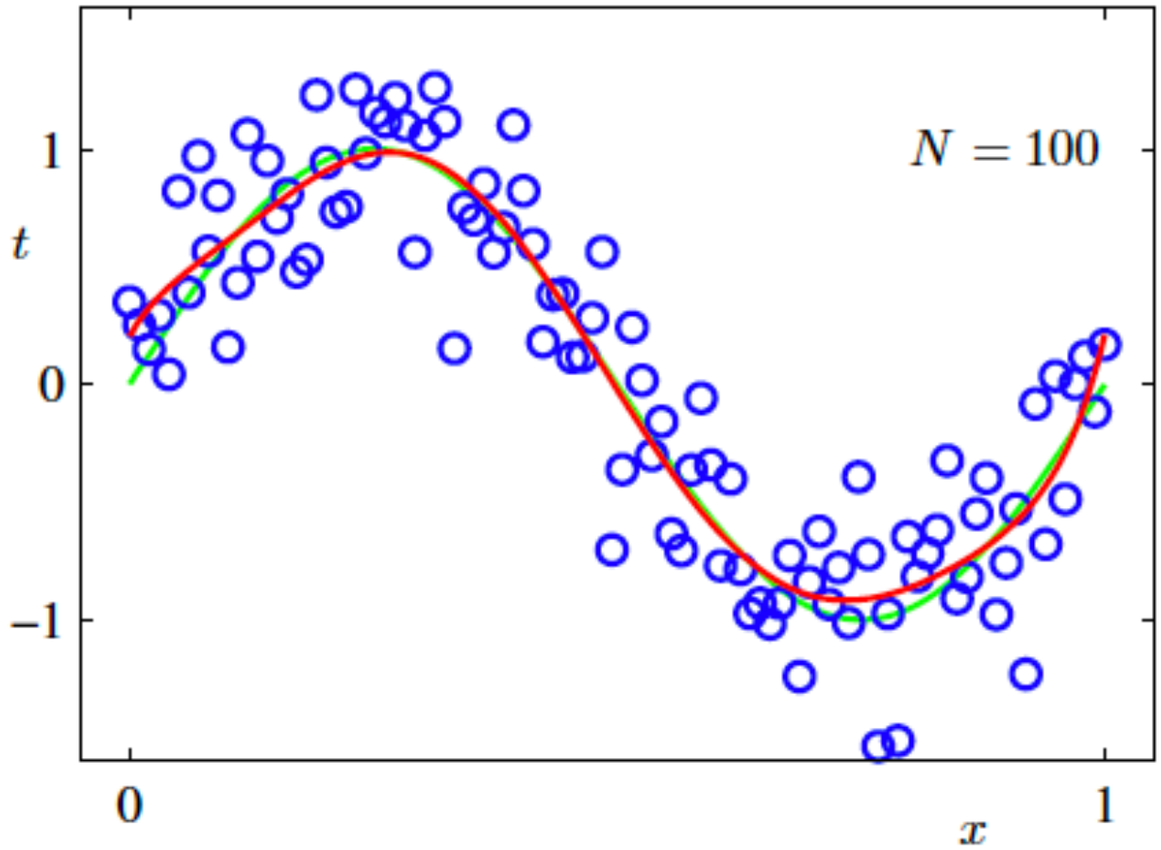


Переобучение

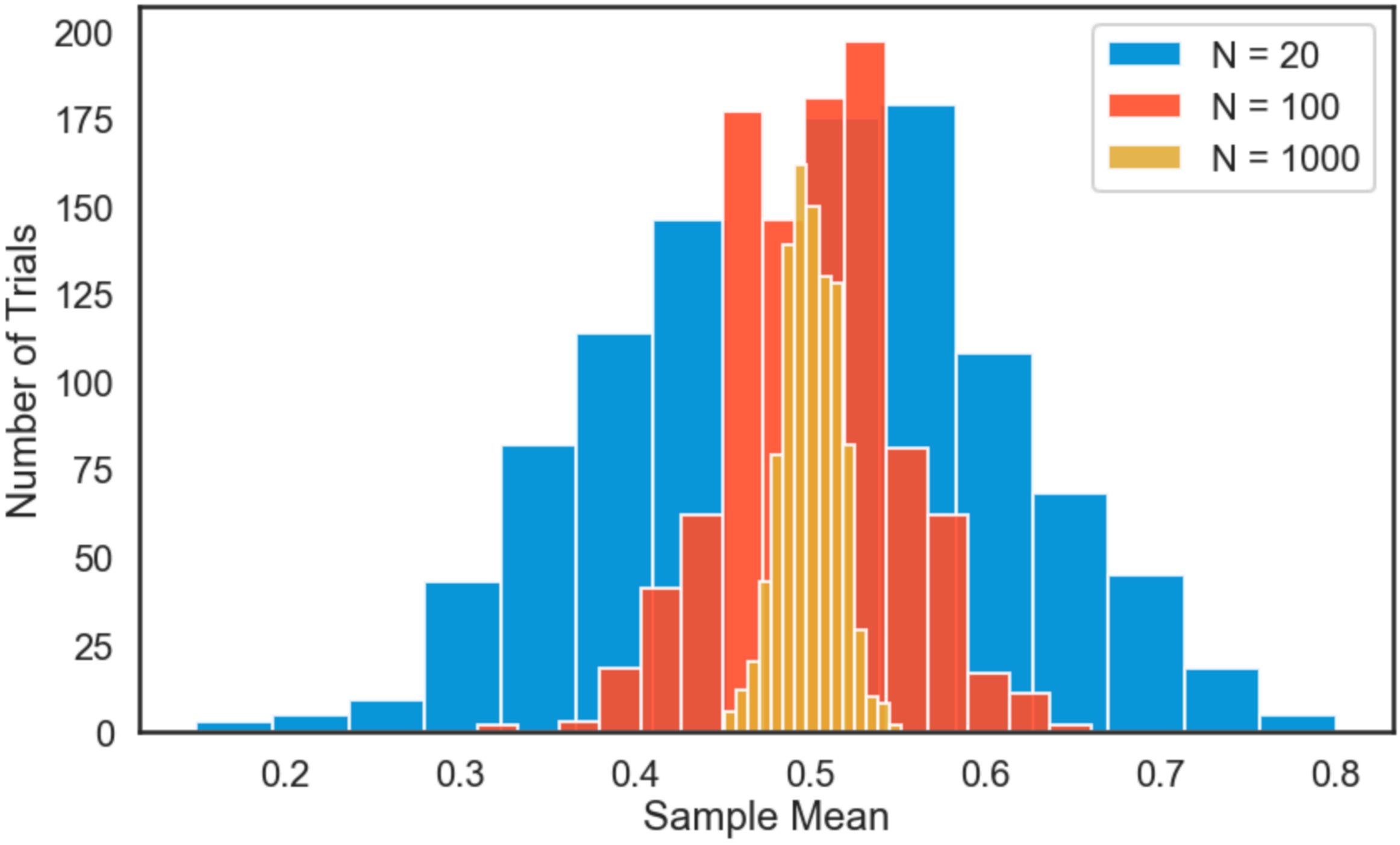
	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Увеличение размера выборки

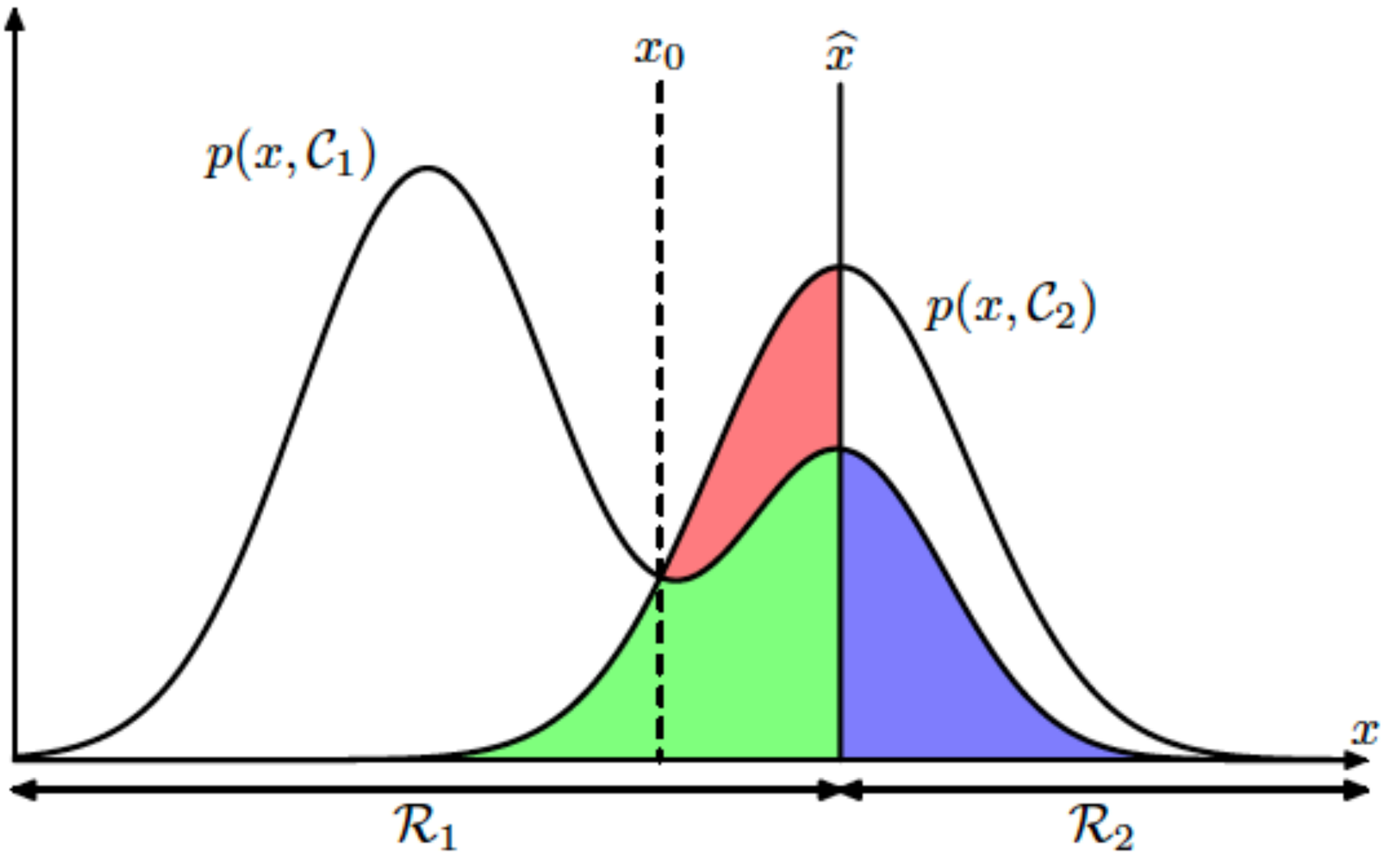




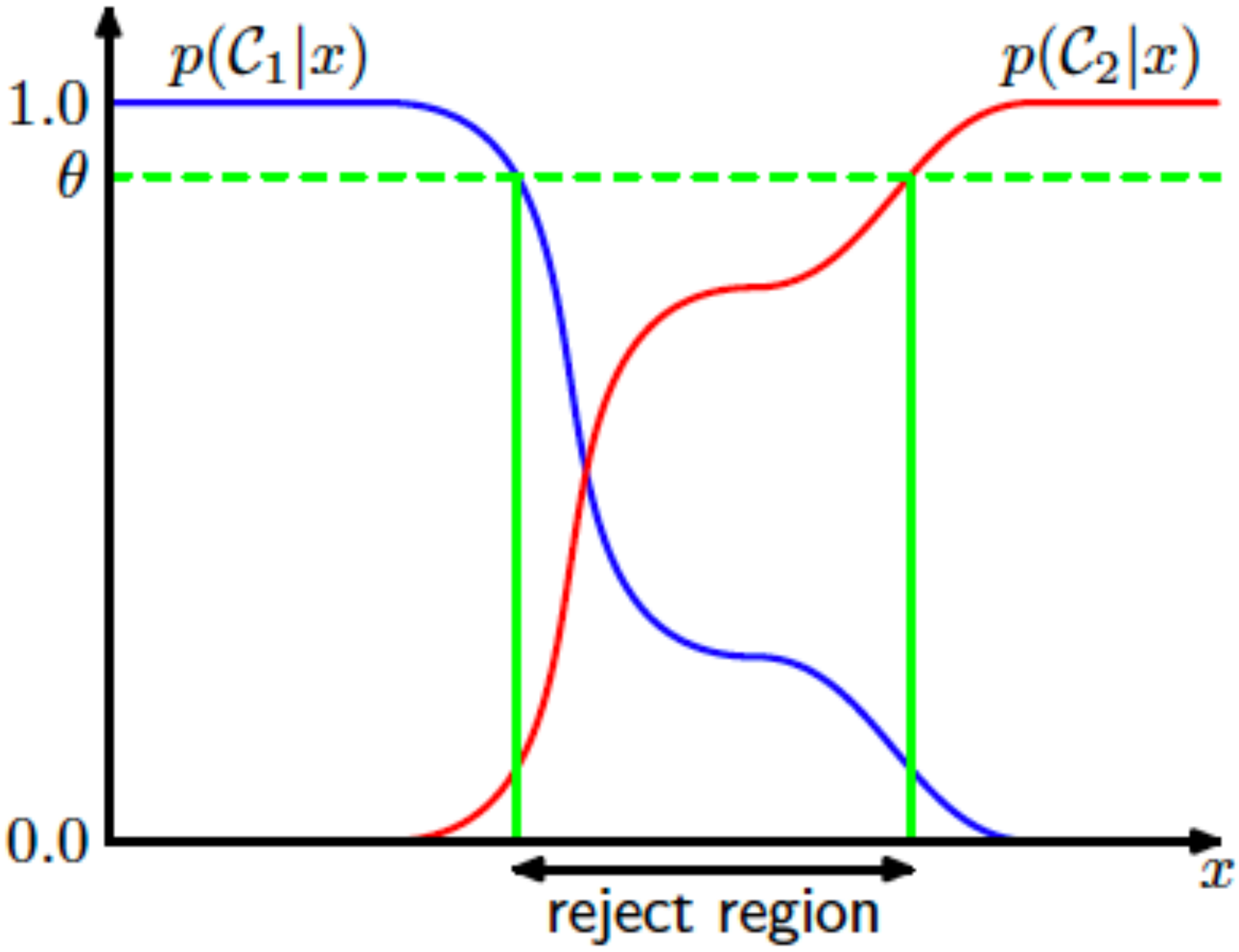
Увеличение размера выборки



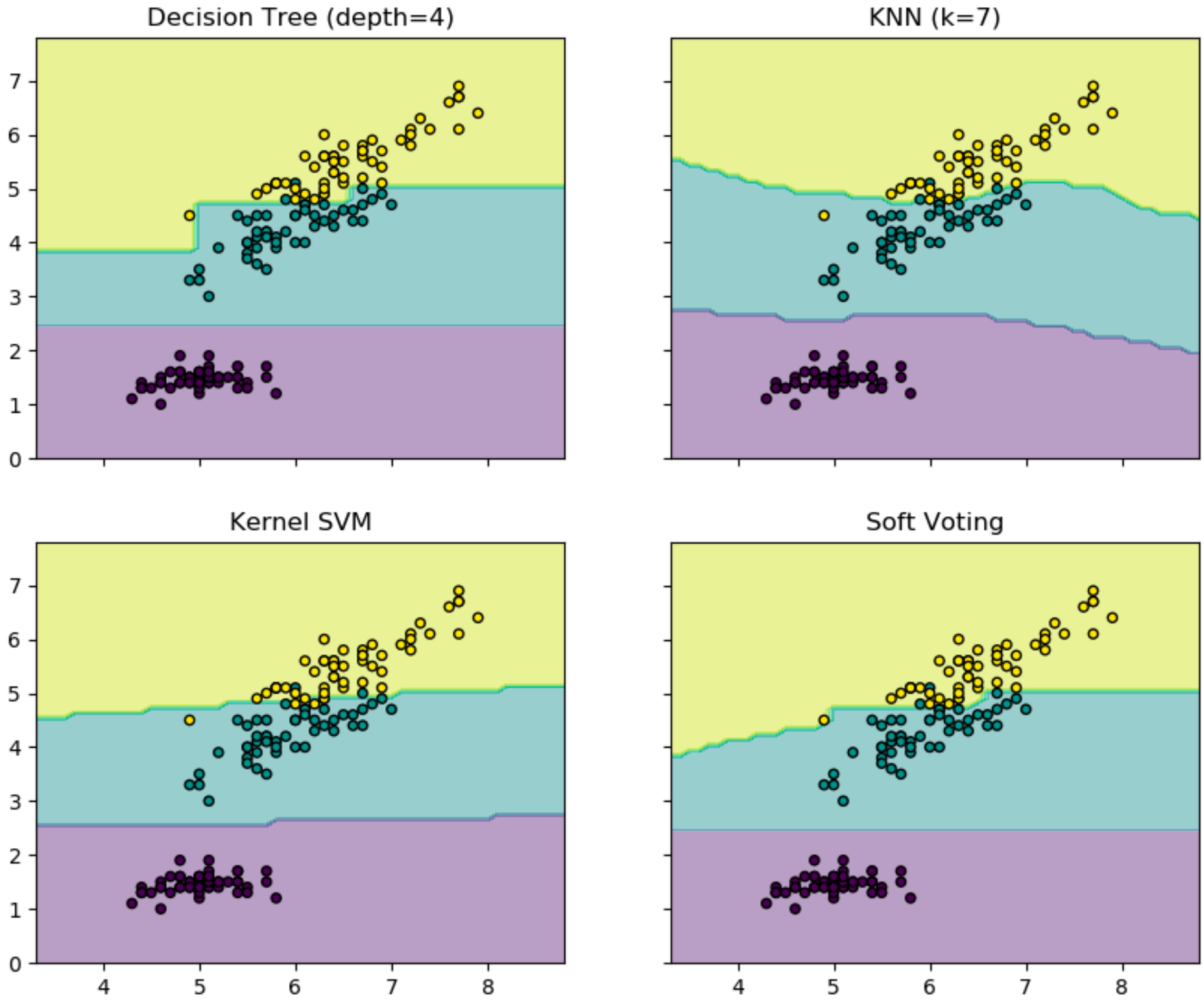
Decision border (Граница решений)



Decision border



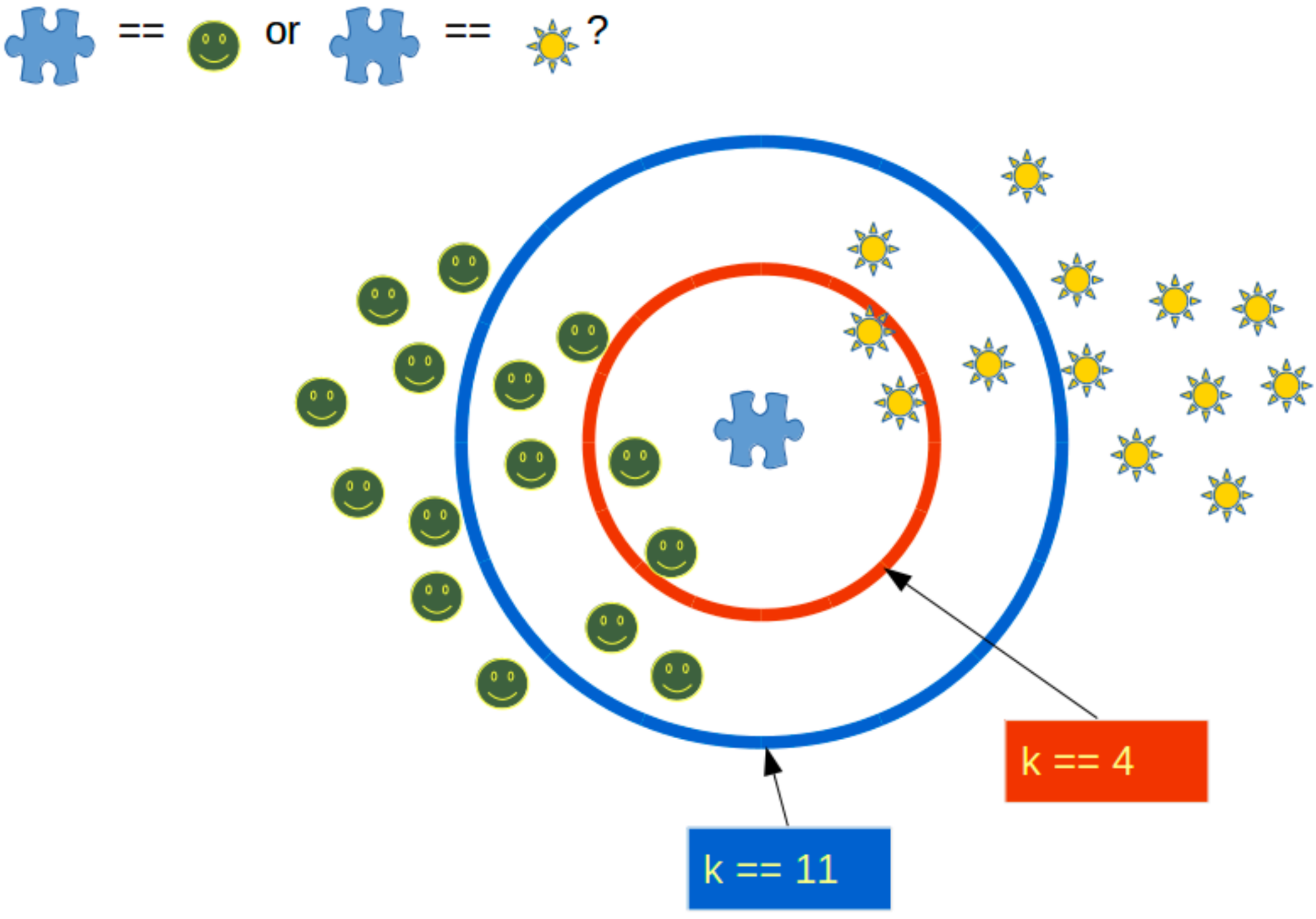
Decision border



K-nearest neighbors

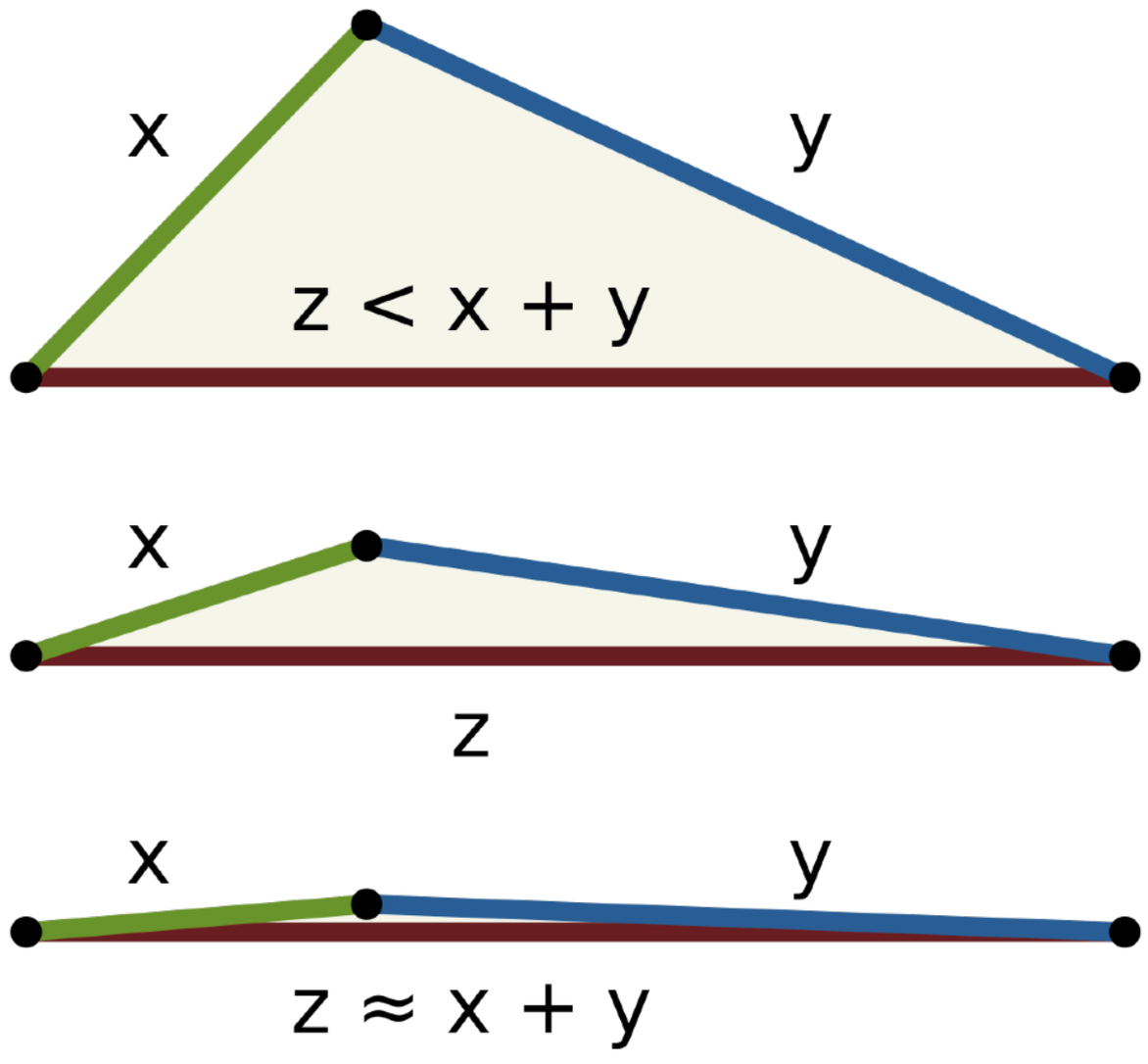
Идея

K-nearest neighbors



Distance = Metric

1. d(x, y) >= 0, d(x, y) = 0 <=> x = y
2. d(x, y) = d(y, x) (Симметричность)
3. d(x, z) <= d(x, y) + d(y, z) (Неравенство треугольника)



Distance

Distance functions

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan

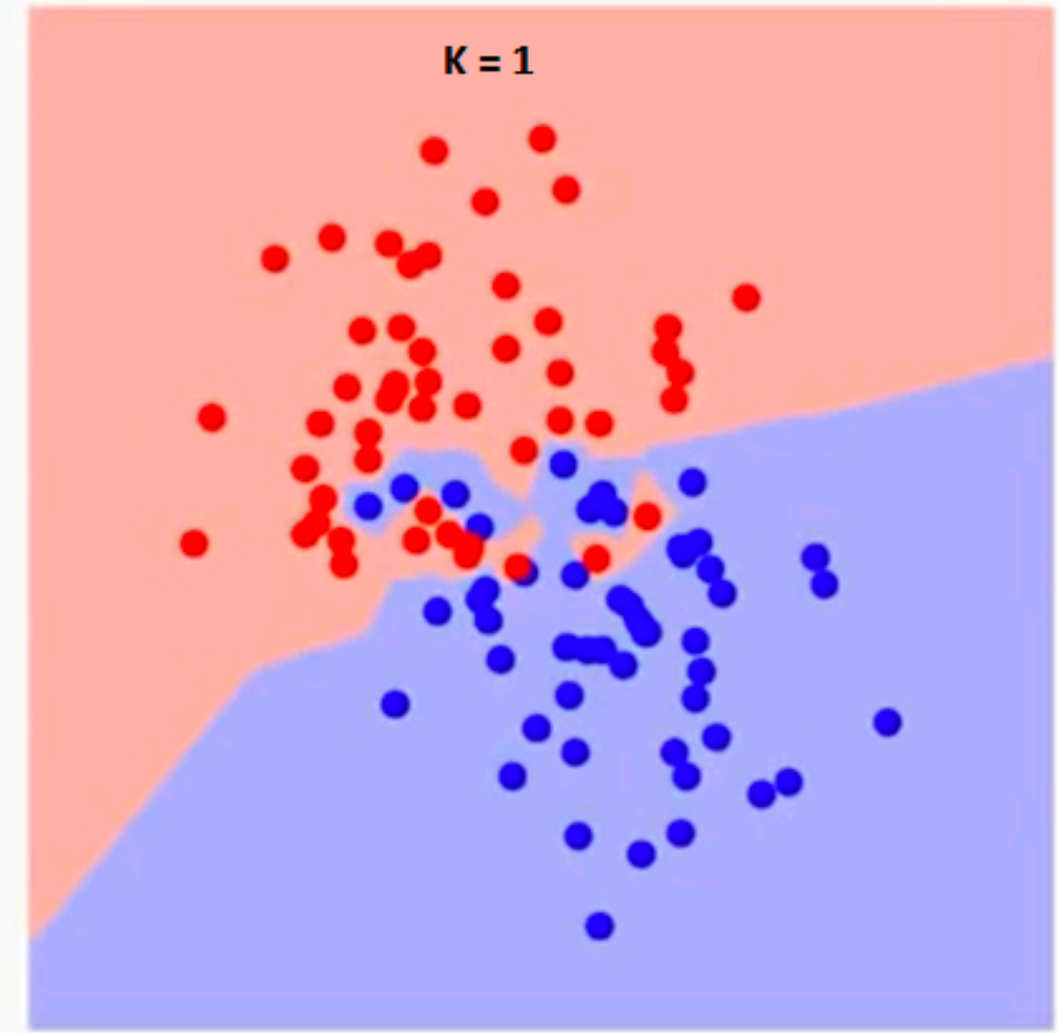
$$\sum_{i=1}^{k}|x_i - y_i|$$

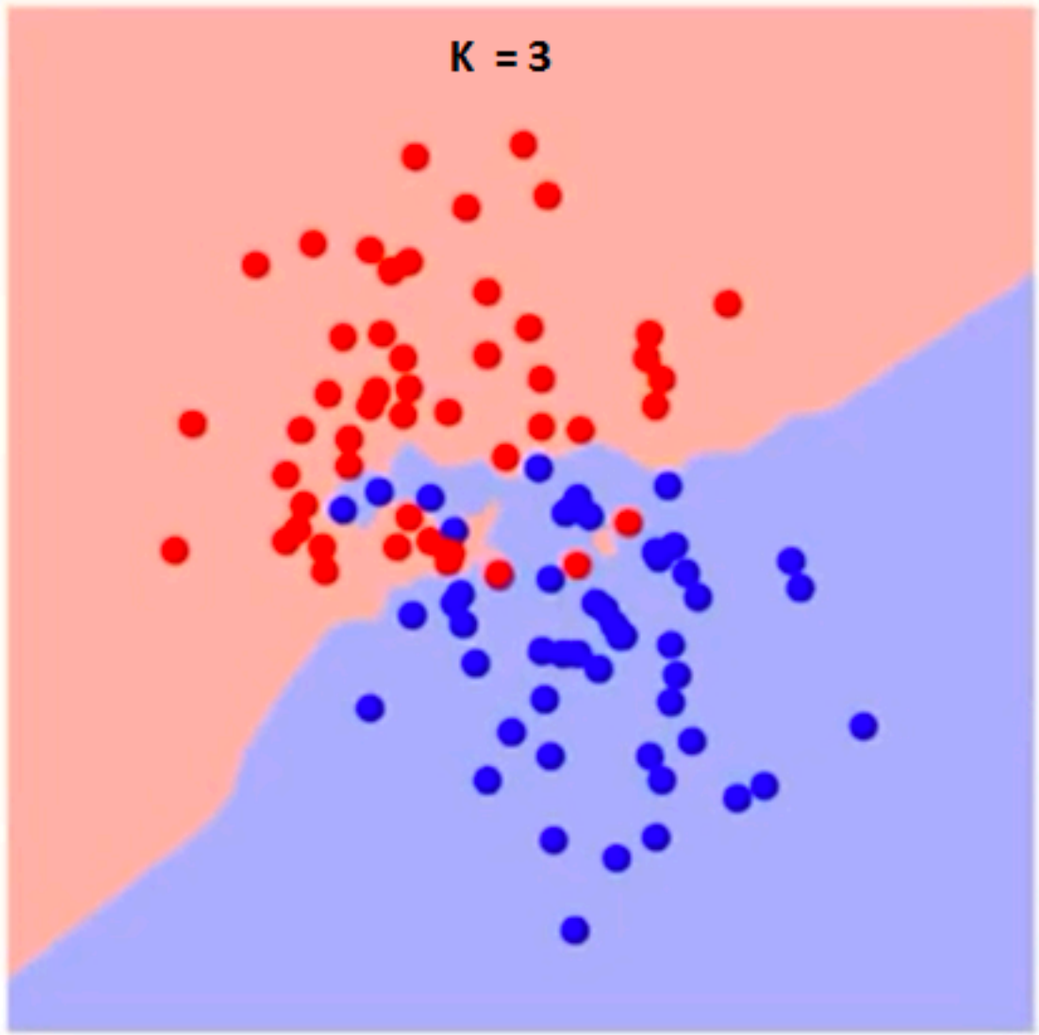
Minkowski

$$\left(\sum_{i=1}^{k}\left(|x_i - y_i|\right)^q\right)^{1/q}$$

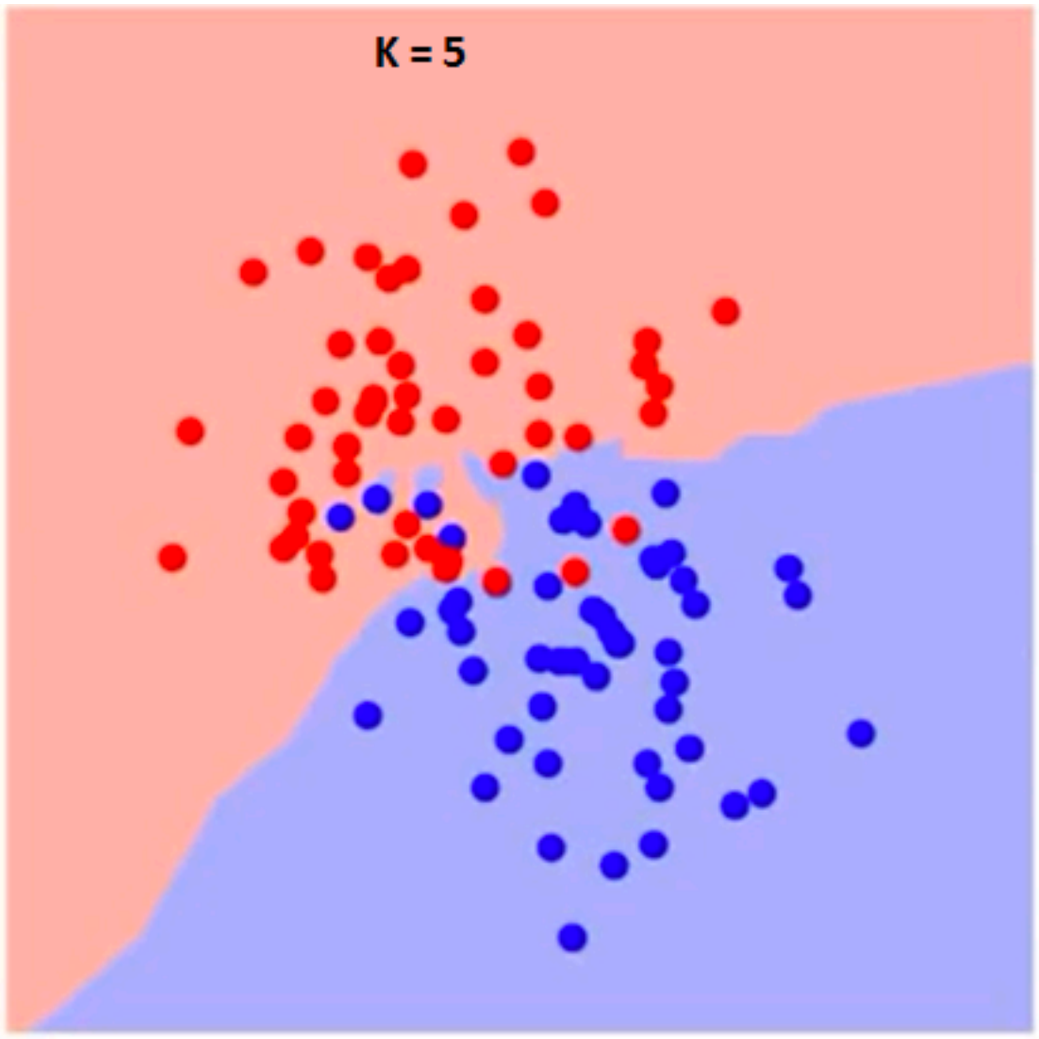
Много их, например можно посмотреть описание https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html

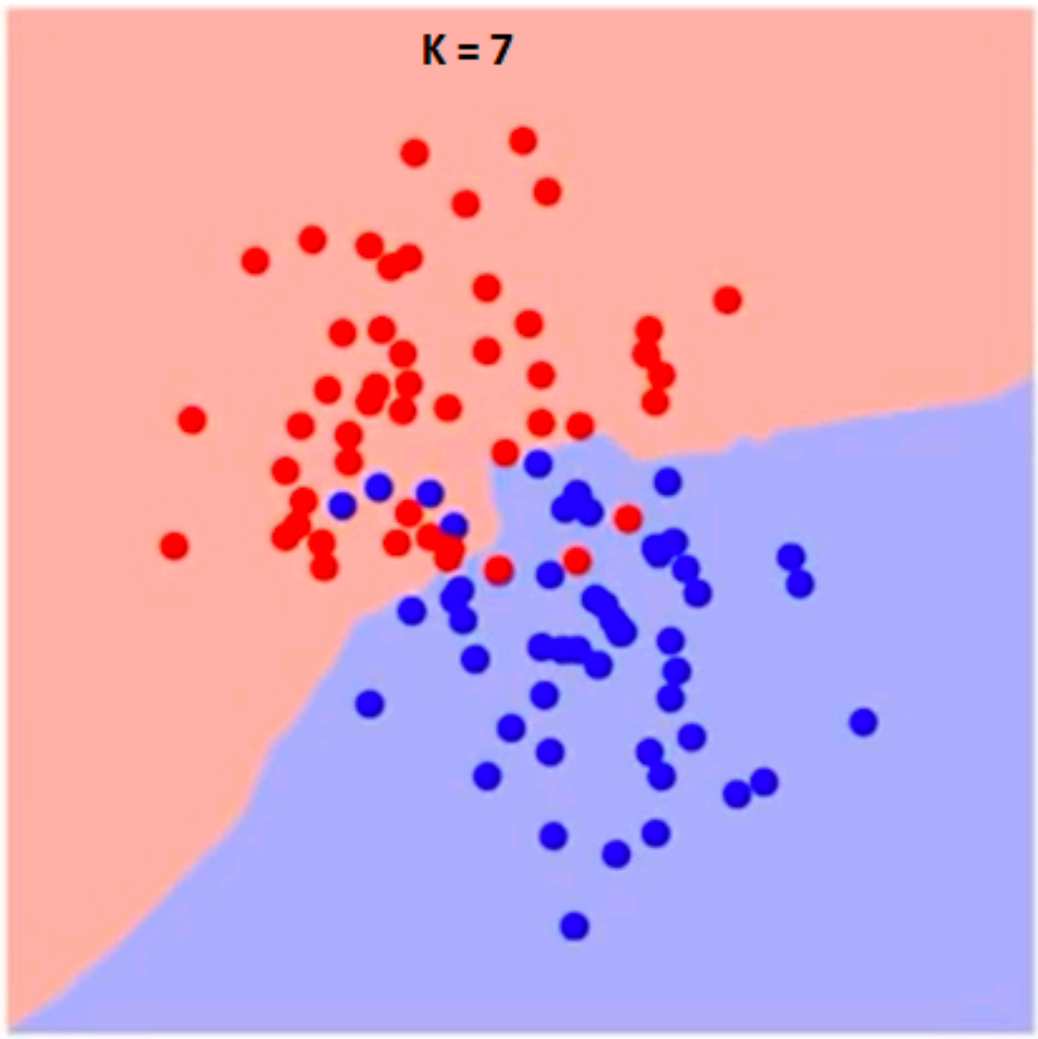
K-nearest neighbors





K-nearest neighbors

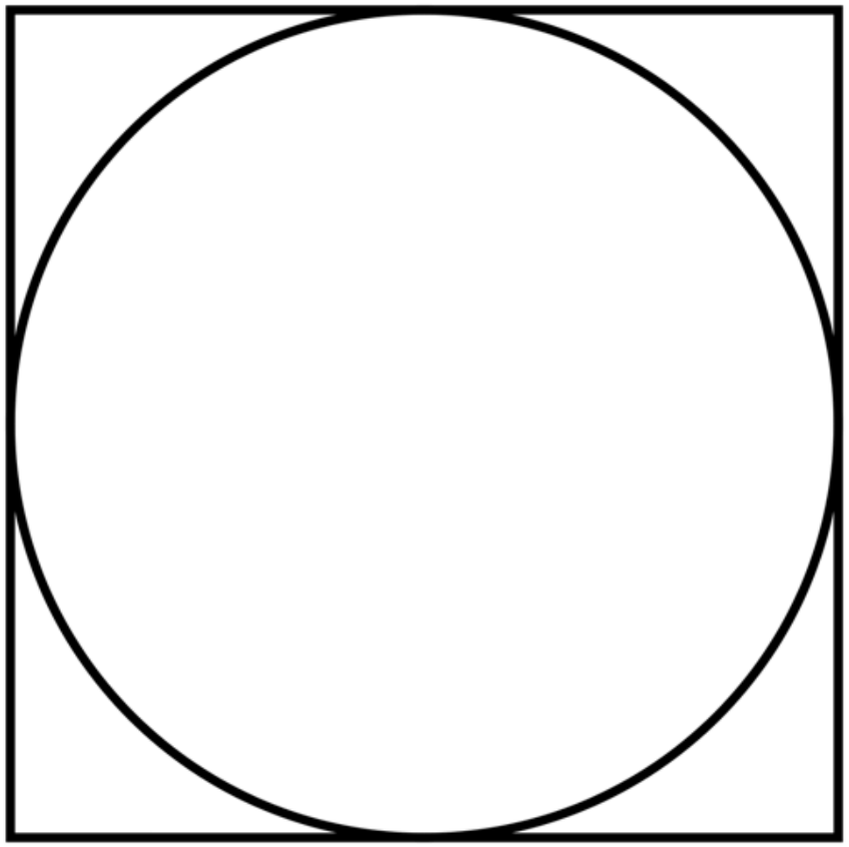


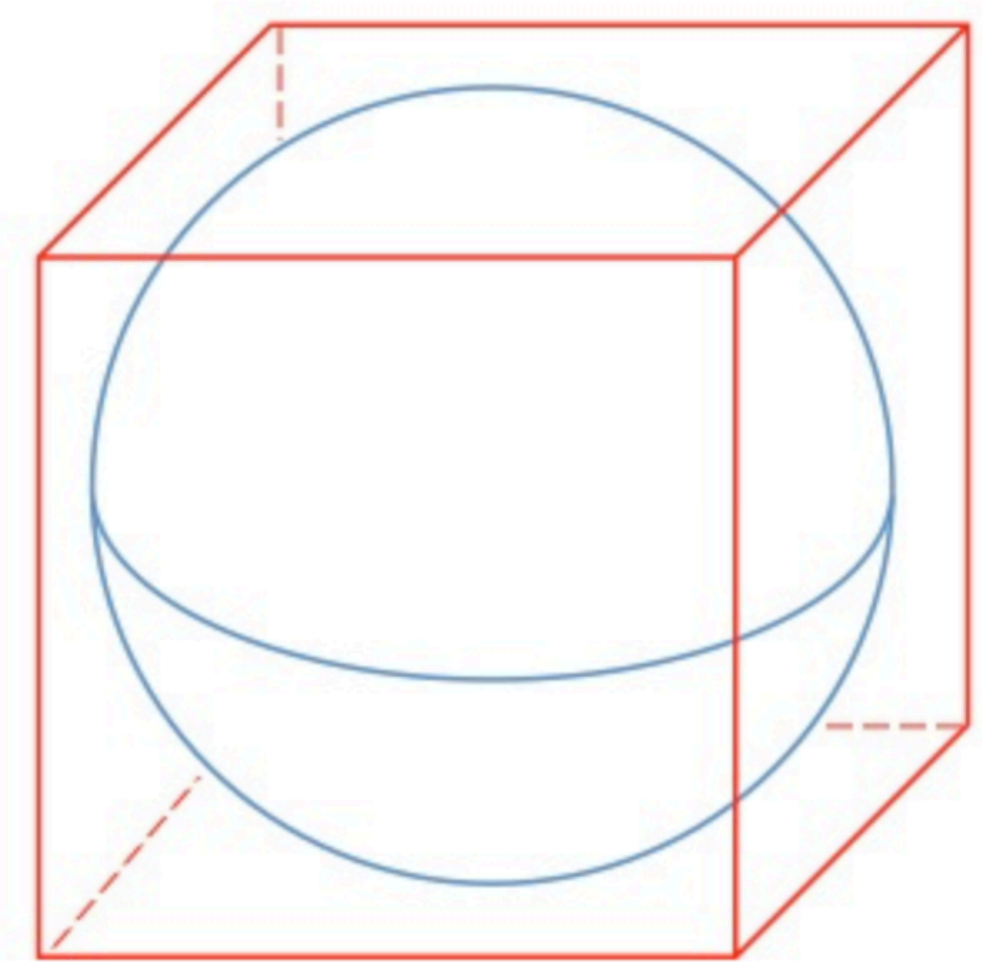


K-nearest neighbors

Какой k будет оптимальным на train?

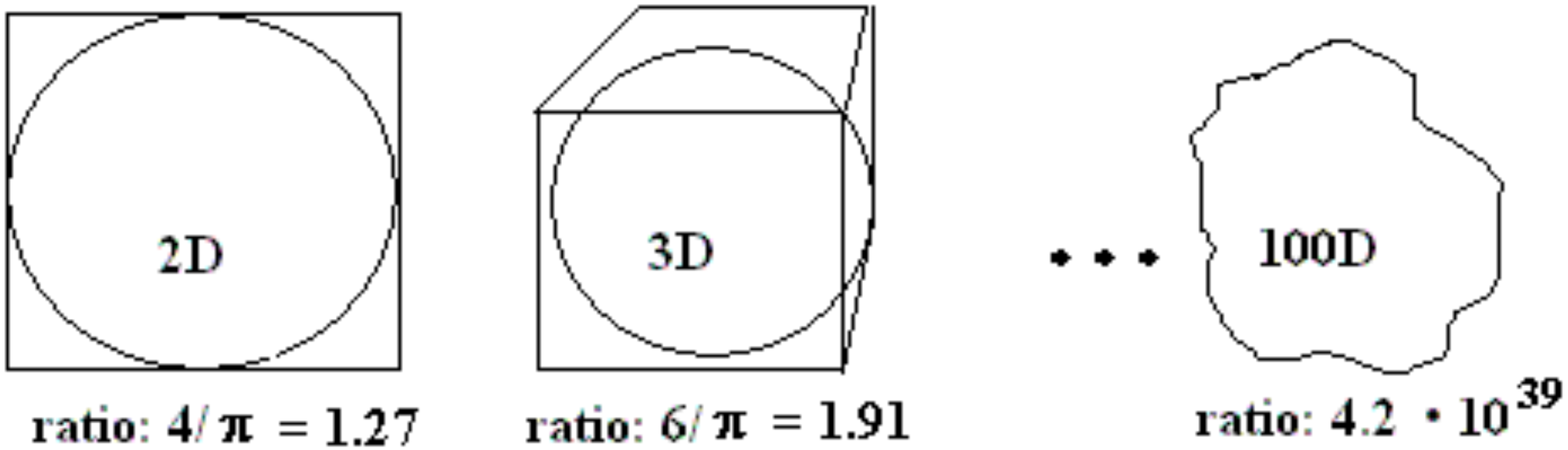
Curse of high dimensionality





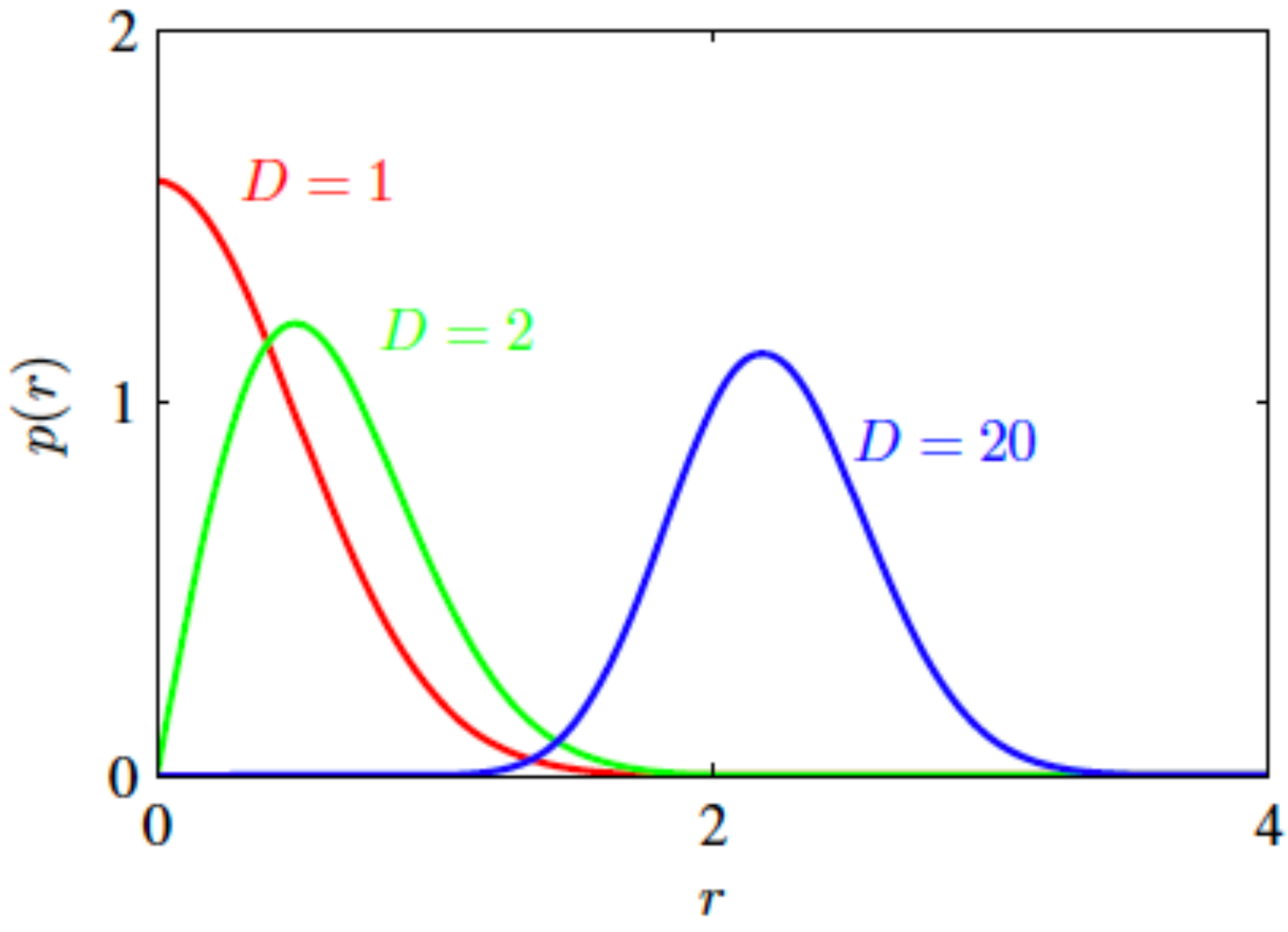
Какова вероятность того, что точка будет вне сферы?

Curse of high dimensionality



В пространстве большой размерности почти все точки находятся “на границе”

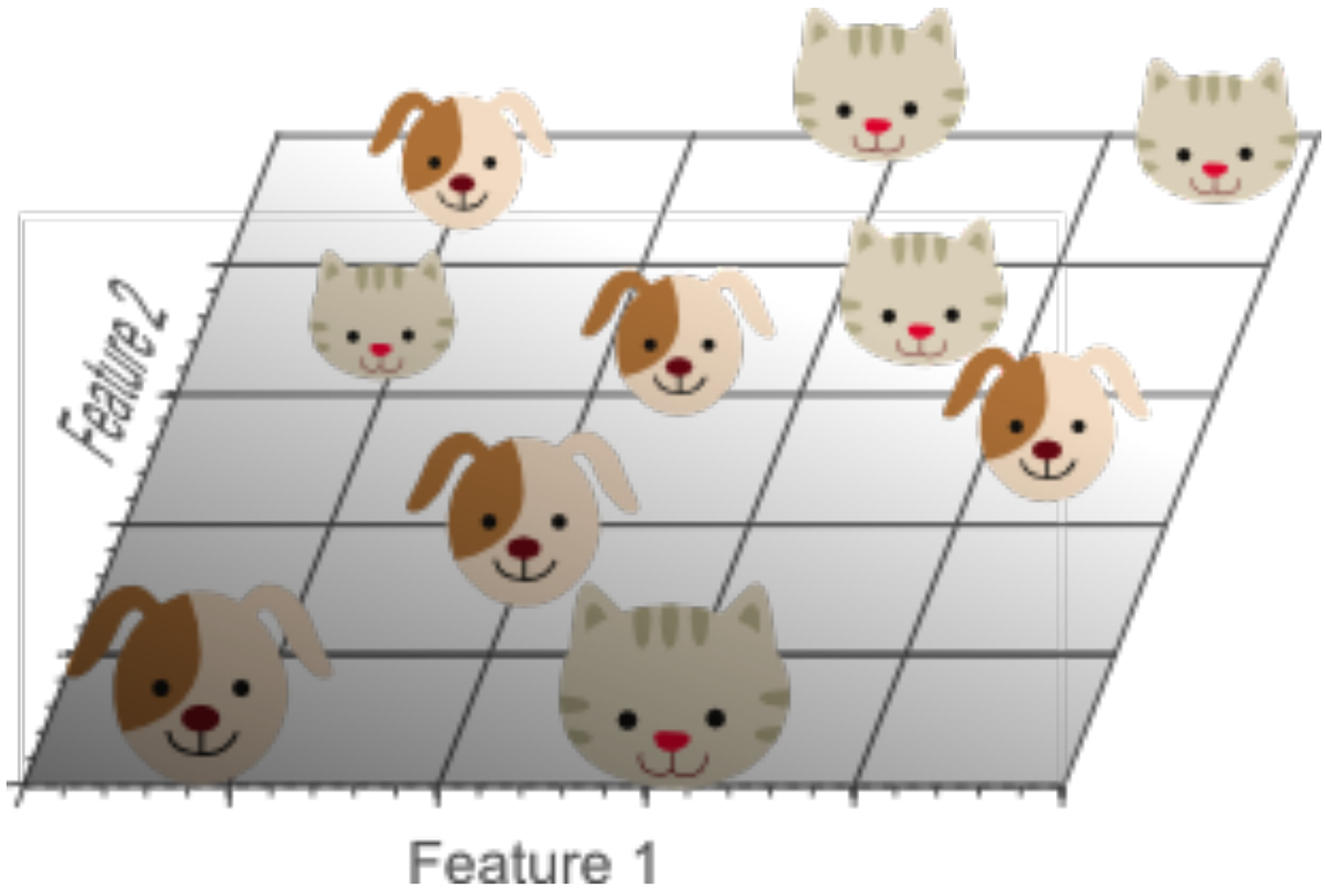
Curse of high dimensionality



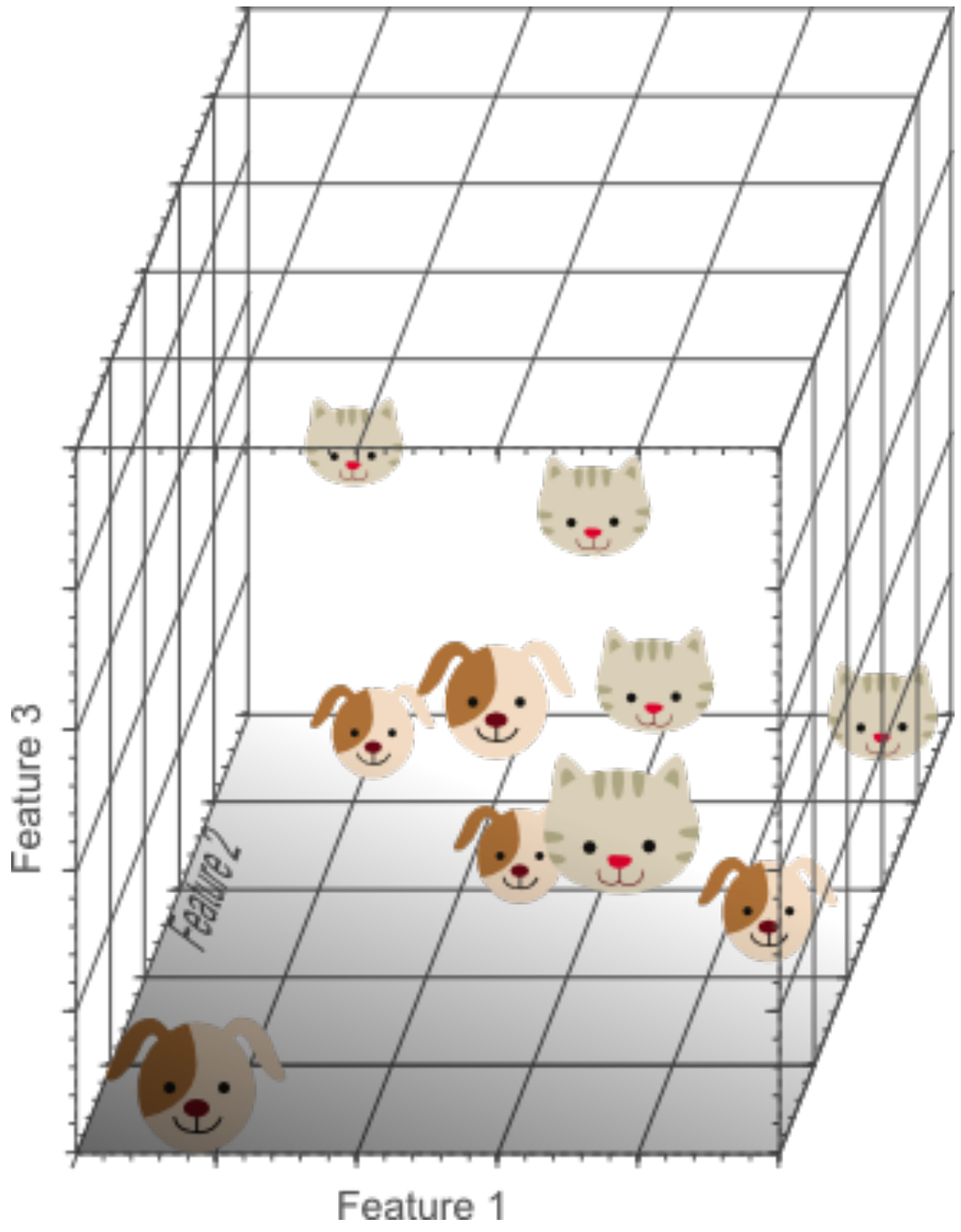
Curse of high dimensionality, ML

Feature 1

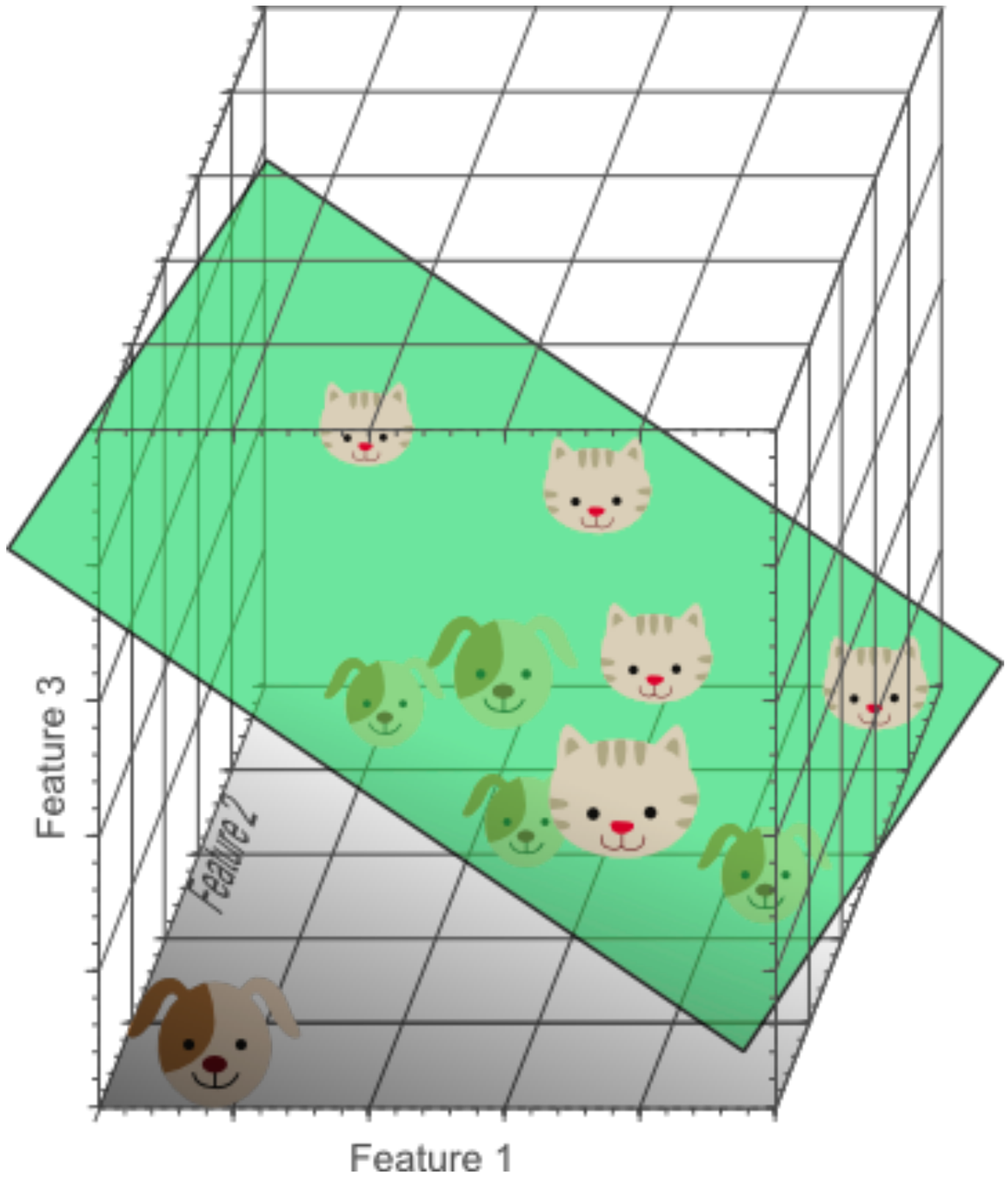
Curse of high dimensionality, ML



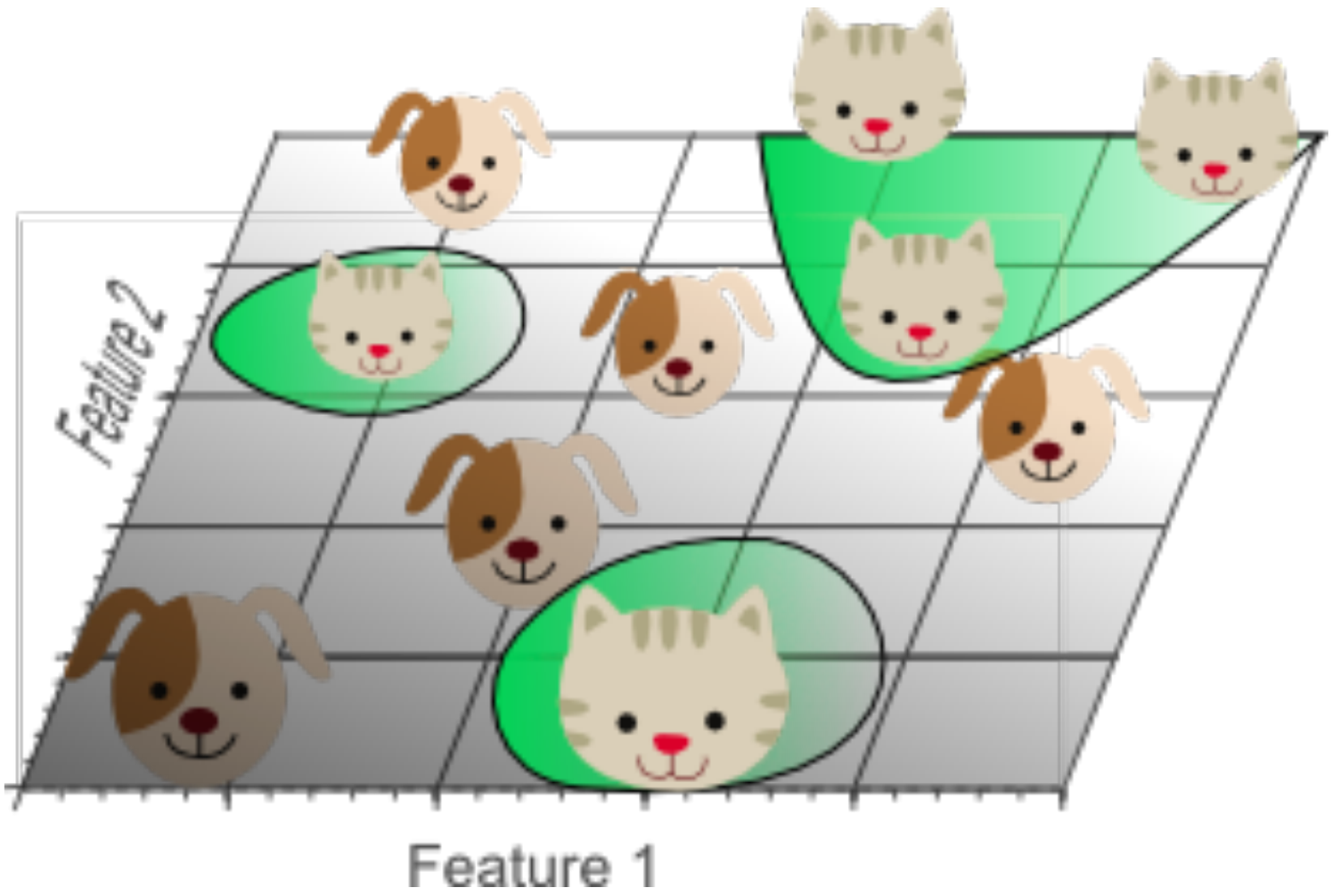
Curse of high dimensionality, ML



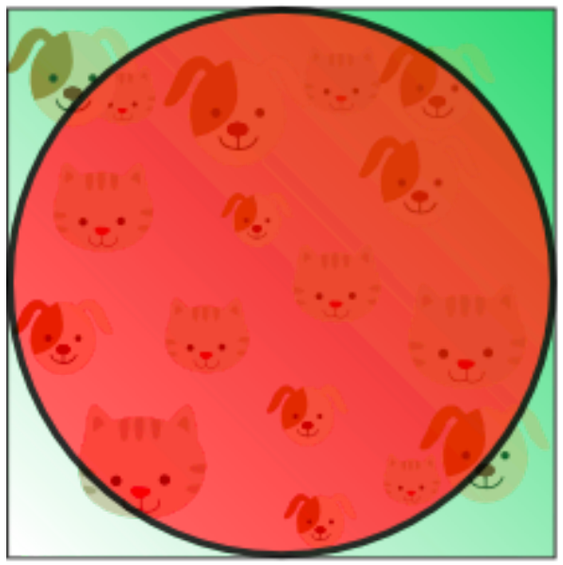
Curse of high dimensionality, ML

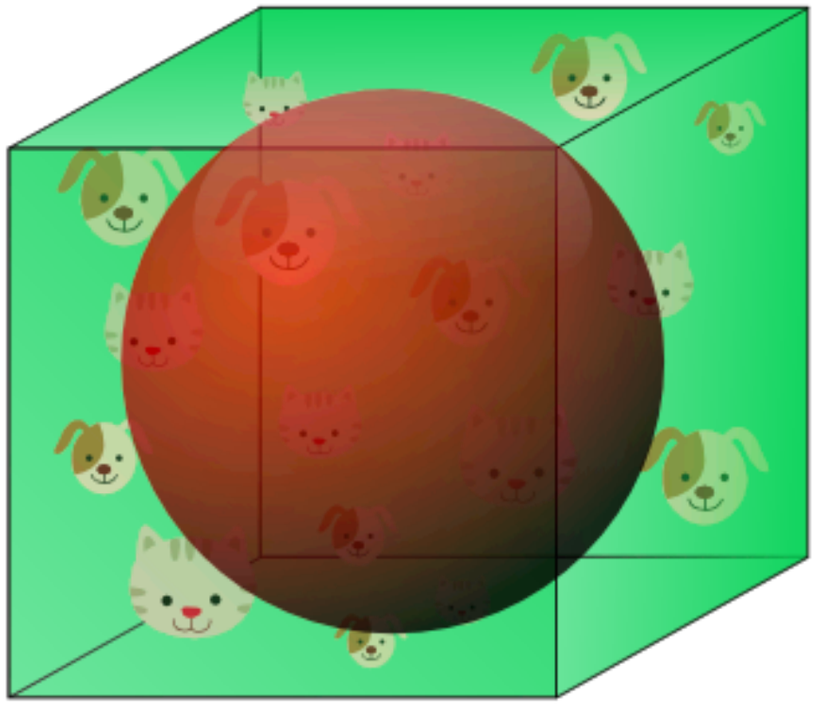


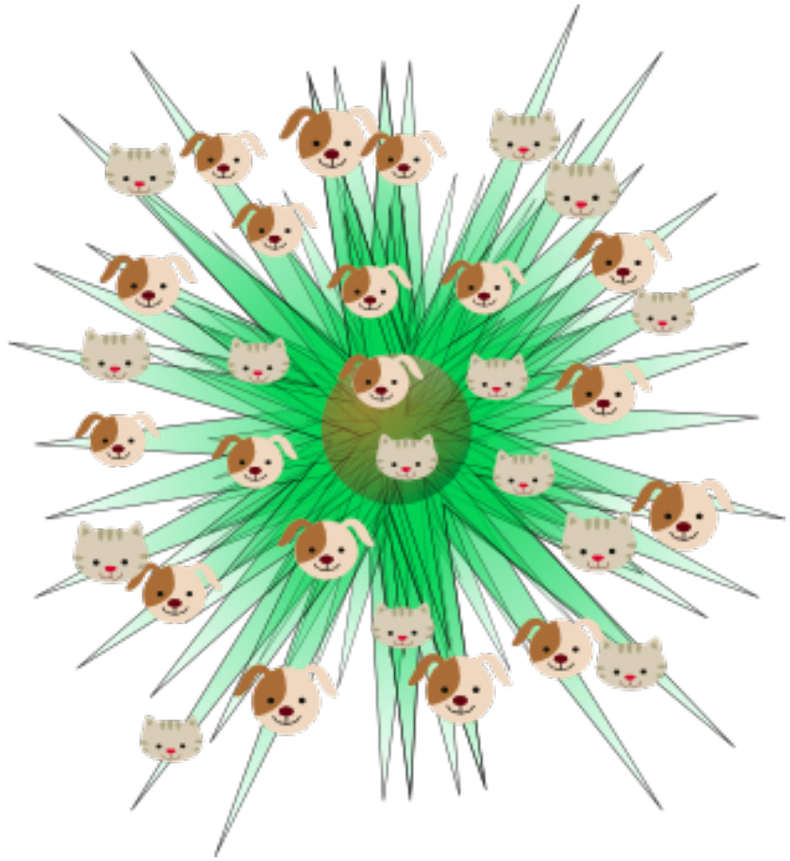
Curse of high dimensionality, ML



Curse of high dimensionality, ML

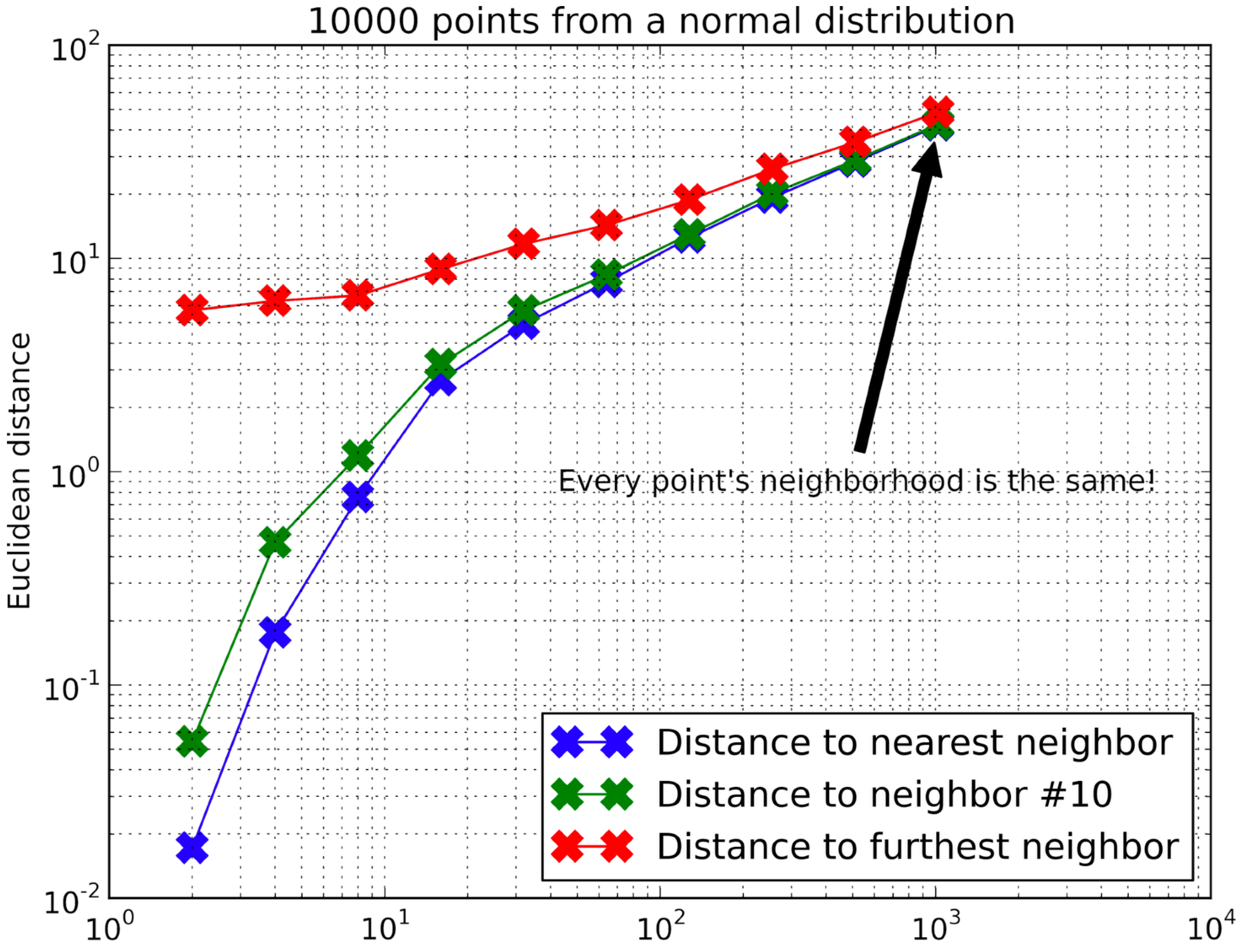




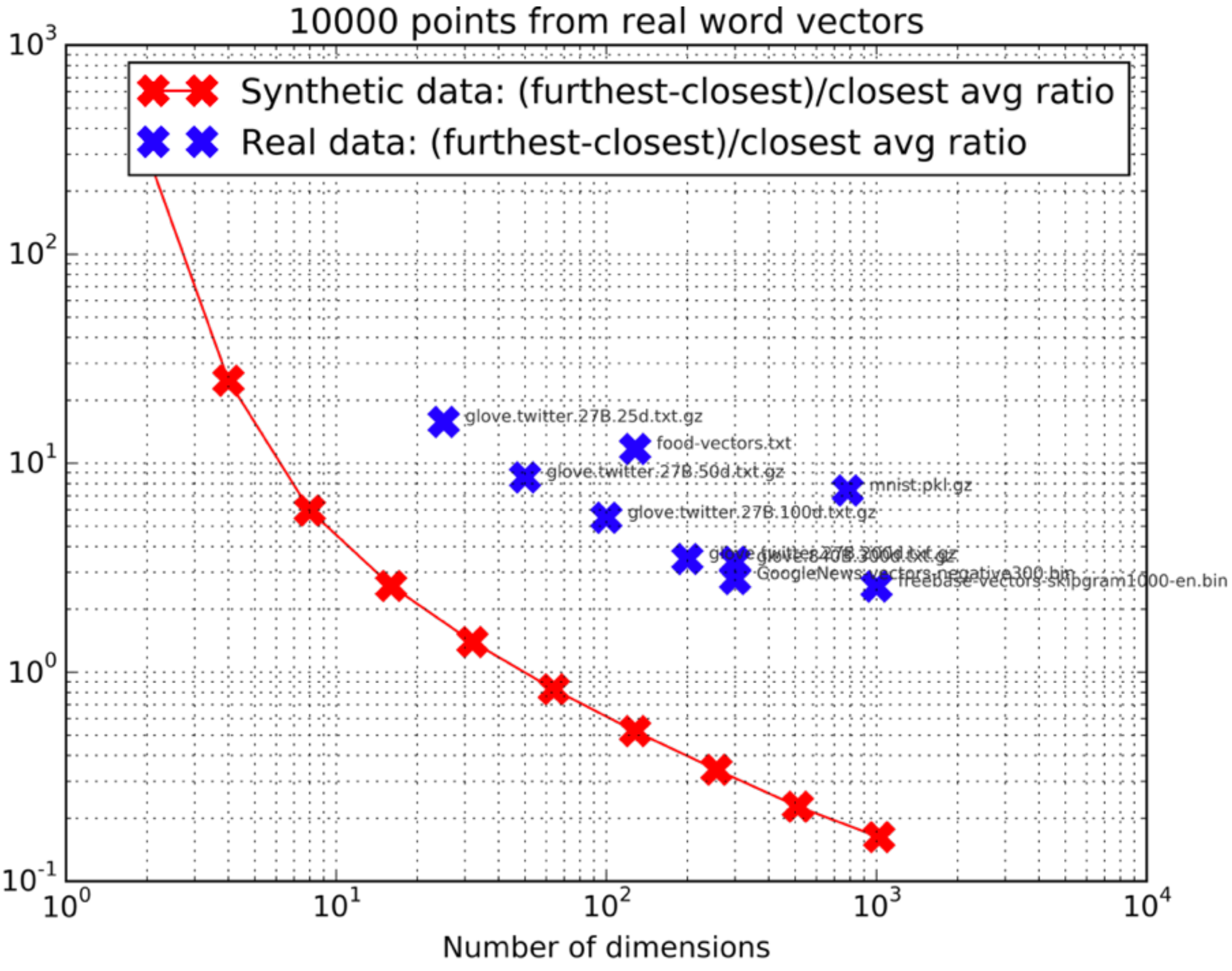


Чем больше признаков - тем больше объекты отличаются друг от друга - тем с какого-то момента сложнее происходит классификация

Симуляция


10000 points from a normal distribution
Euclidean distance
10^2
10^1
10^0
10^{-1}
10^{-2}
10^0
10^1
10^2
10^3
10^4
Every point's neighborhood is the same!
Distance to nearest neighbor
Distance to neighbor #10
Distance to furthest neighbor

Реальные данные

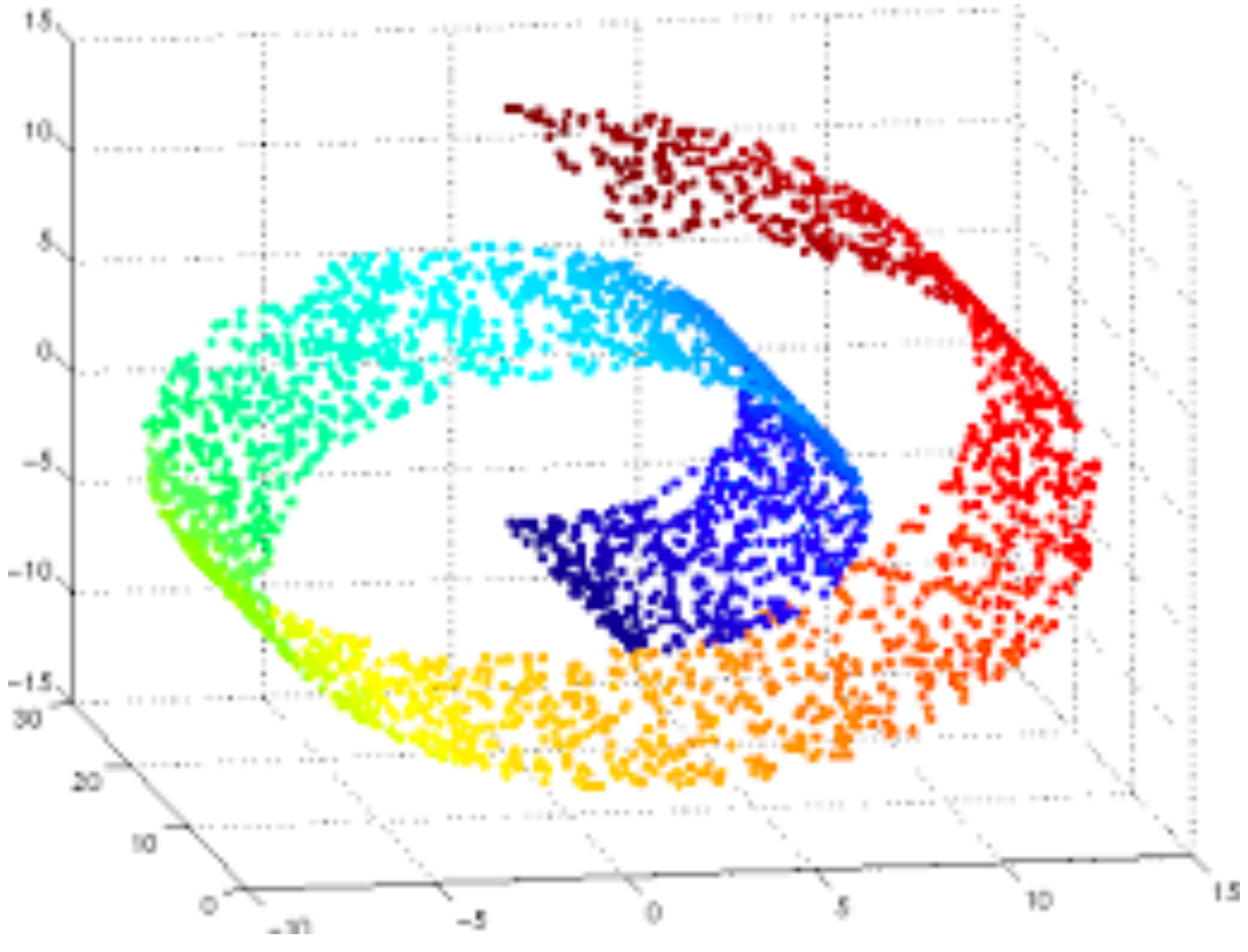


Другие модели и KNN

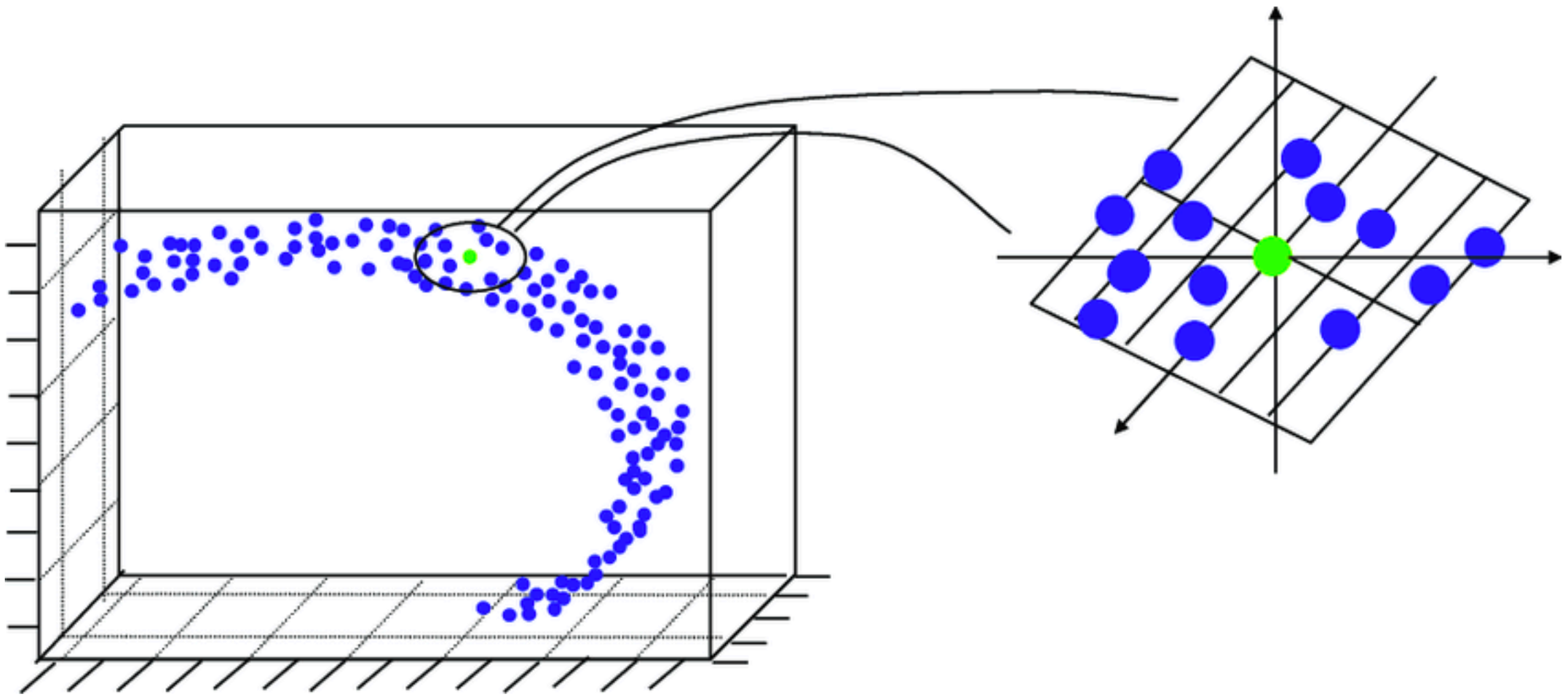
Многие модели ведут себя так же, как и KNN. Особенно в плохих случаях - когда данные сильно зашумлены.

Например, случайный лес и некоторые нейронные сети.

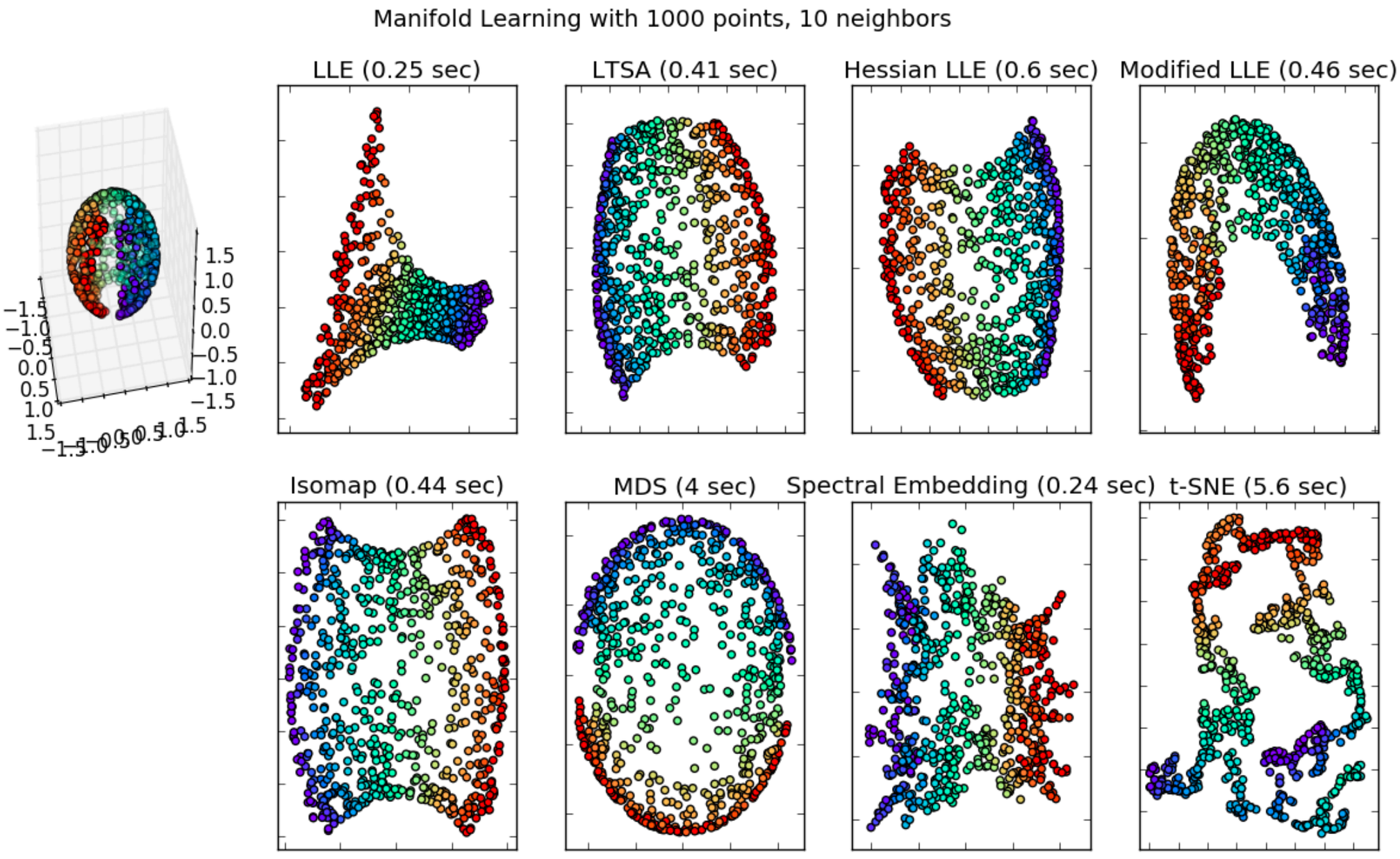
Manifold learning



Manifold learning

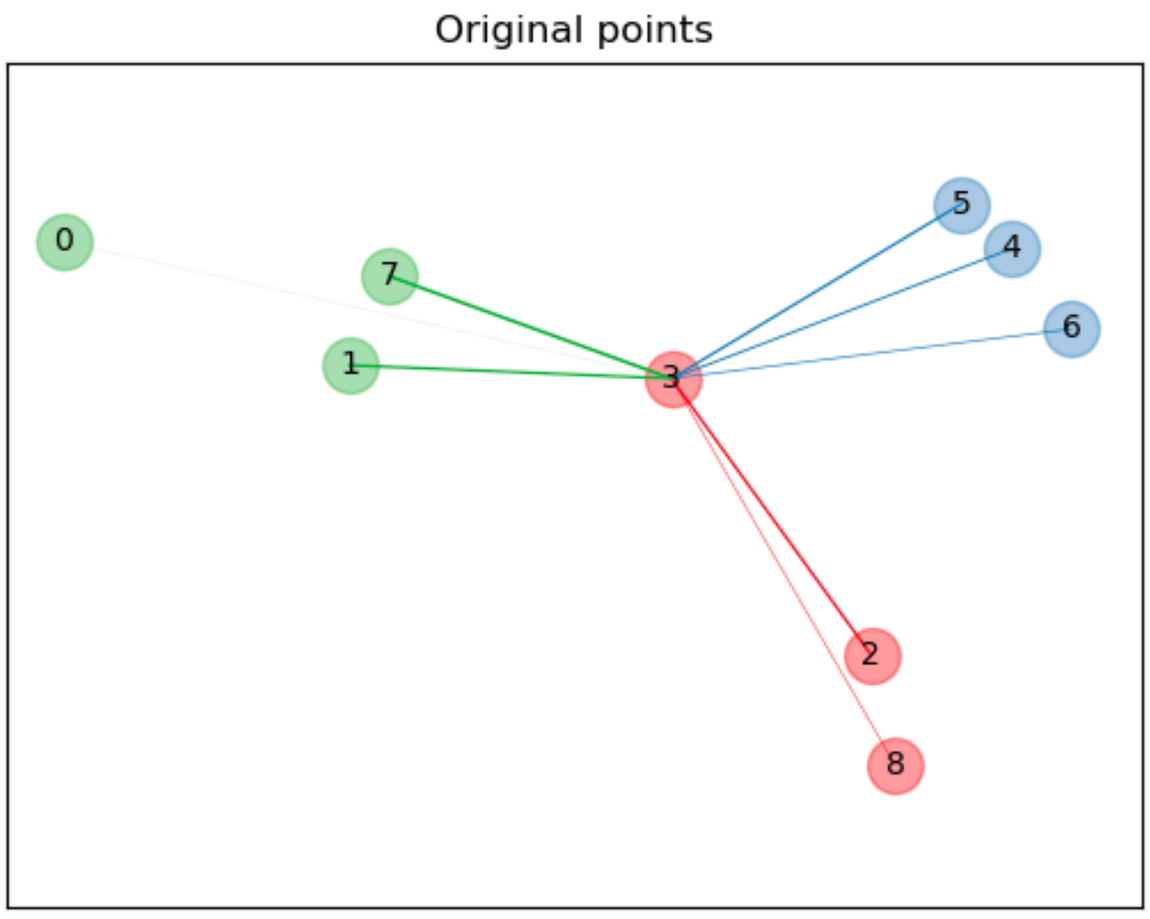


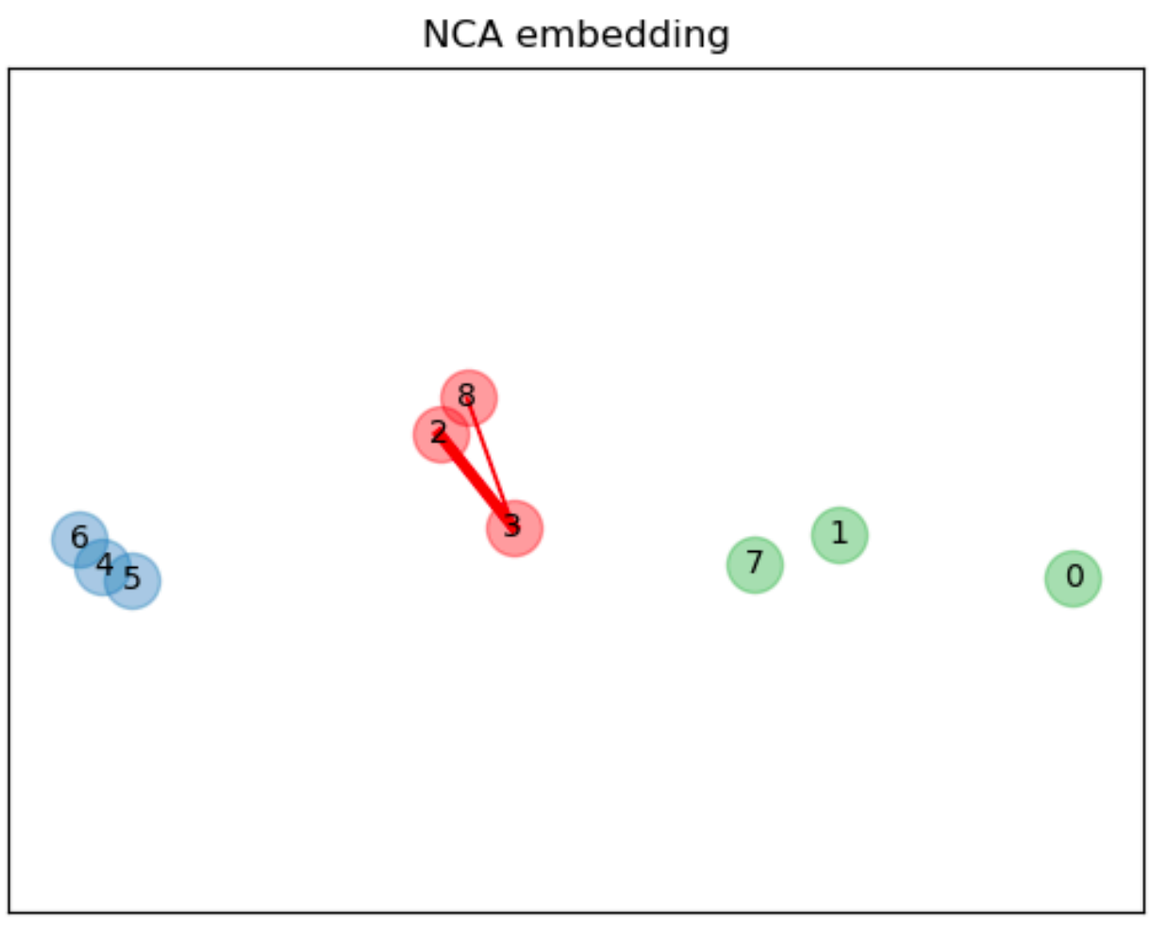
Manifold learning



Много способов, часть разберем позже

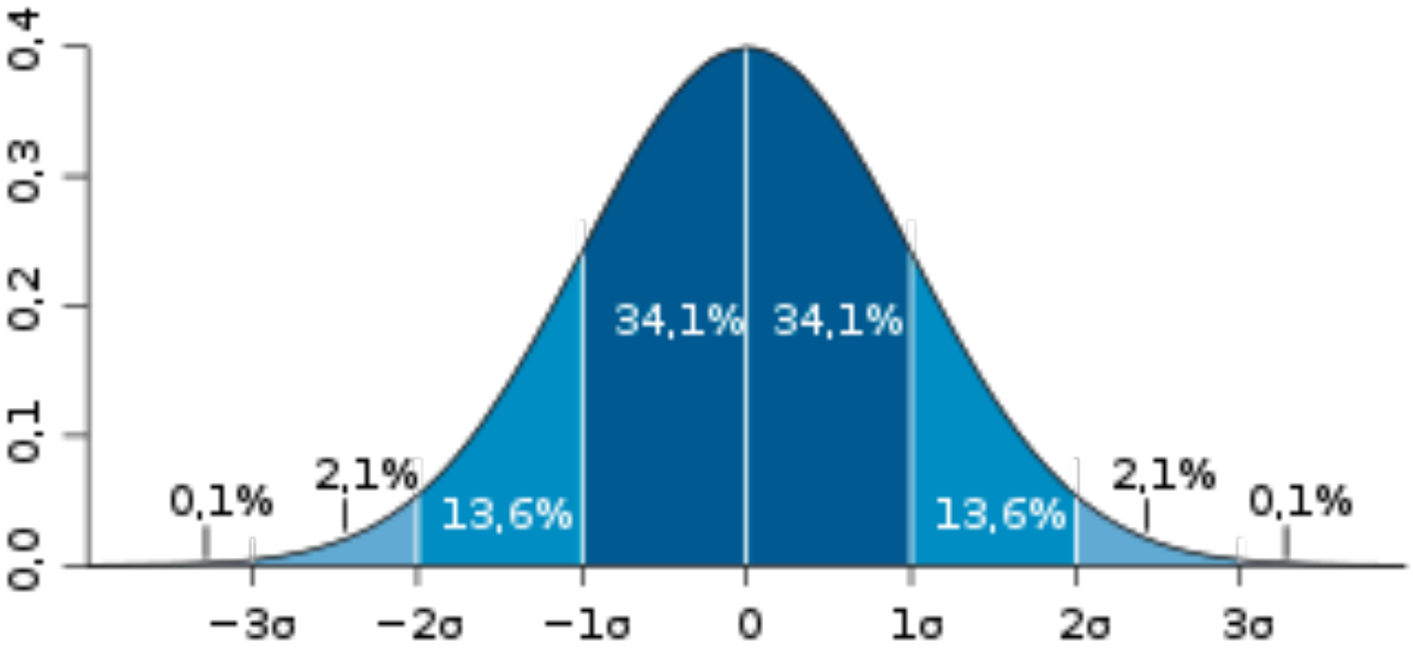
Manifold learning





Neighborhood Components Analysis

Оценка параметра



Генеральная совокупность

Параметры ← **Оценки параметров**

Случай оценки параметра

$$Bias(\hat{\theta}_S, \theta) = \mathbb{E}_{S \sim D^m}[\hat{\theta}_S] - \theta$$

$$Var(\theta_S) = Var_{S \sim D^m}[\theta_S]$$

Мешок - в нем 10% черных шаров.

Вы достаете 10 шаров. Оценить долю черных шаров в мешке?

Взяли монетку. Подбросили три раза. Все три раза выпал орел. Какова вероятность выпадения орла в следующий раз?

Один вариант решения - всегда предполагать, что у нас УЖЕ 1 раз выпал орел ии один раз решка.

Случай оценки параметра

$$\mathbb{E}\left[(\hat{\theta}_S - \theta)^2\right] = \mathbb{E}[\hat{\theta}_S^2] + \theta^2 - 2\mathbb{E}[\hat{\theta}_S]\theta$$

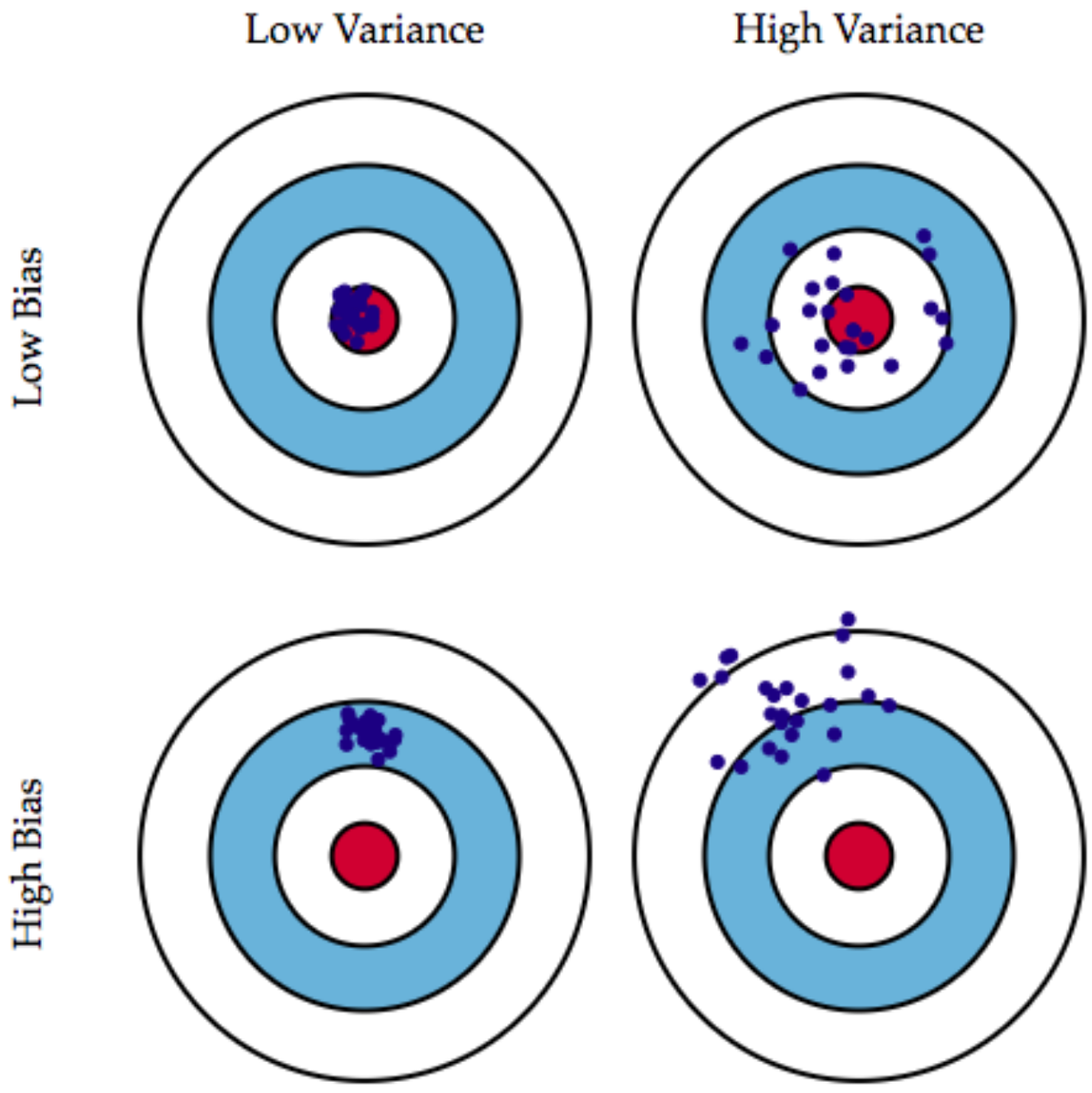
$$Bias^2(\hat{\theta}_S, \theta) = \left(\mathbb{E}[\hat{\theta}_S] - \theta\right)^2$$

$$= \mathbb{E}^2[\hat{\theta}_S] + \theta^2 - 2\mathbb{E}[\hat{\theta}_S]\theta$$

$$Var(\hat{\theta}_S) = \mathbb{E}[\hat{\theta}_S^2] - \mathbb{E}^2[\hat{\theta}_S]$$

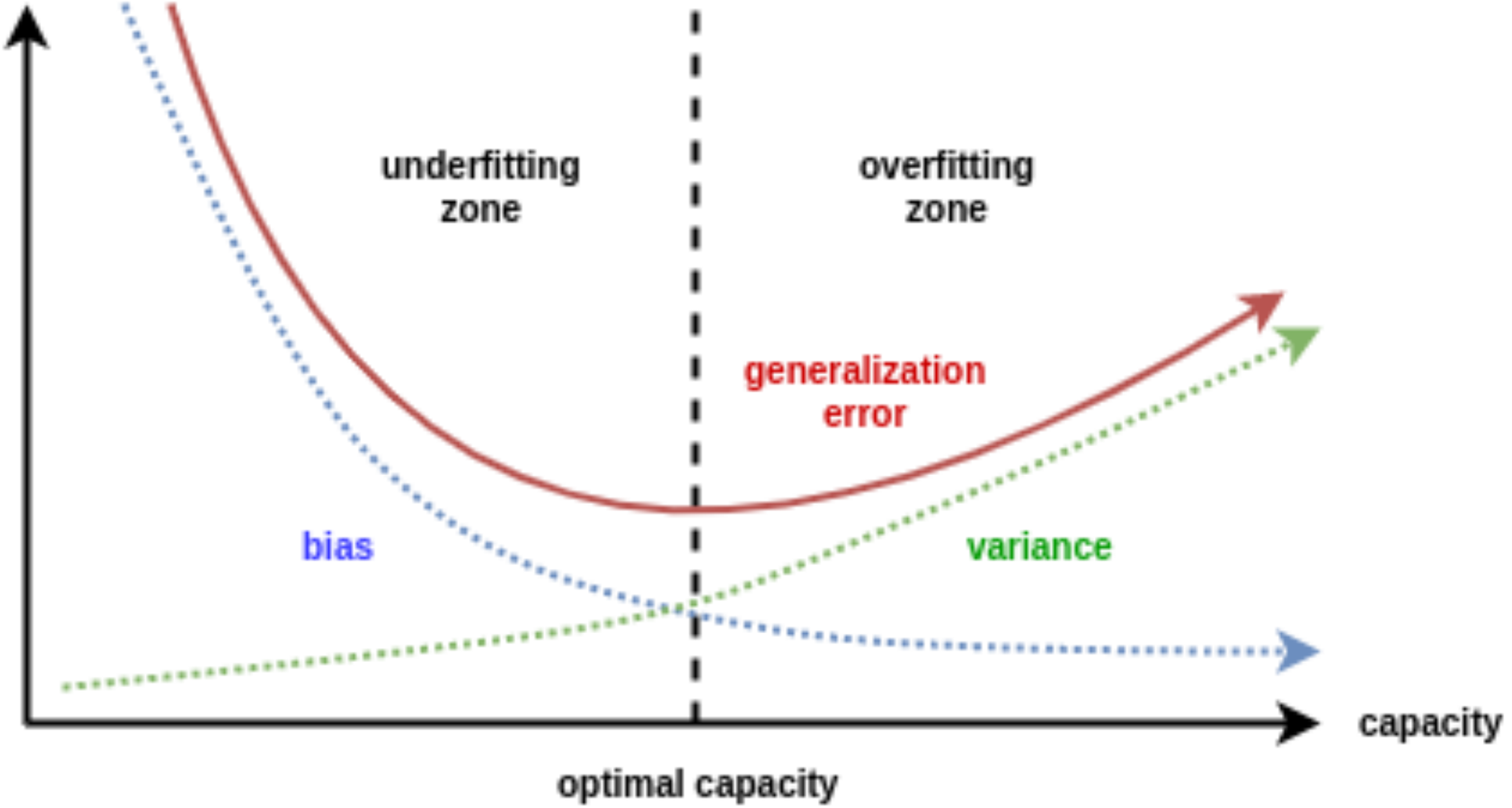
$$MSE = \mathbb{E}\left[(\hat{\theta}_S - \theta)^2\right] = Bias^2(\hat{\theta}_S, \theta) + Var(\hat{\theta}_S)$$

Проблемы при обучении модели

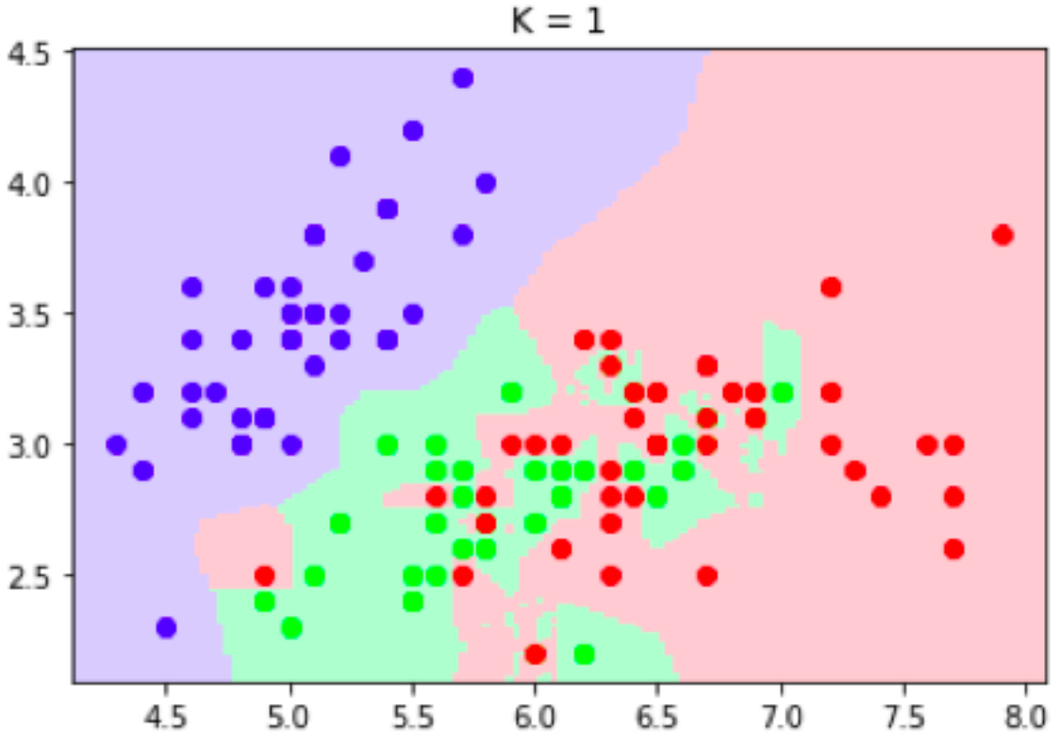


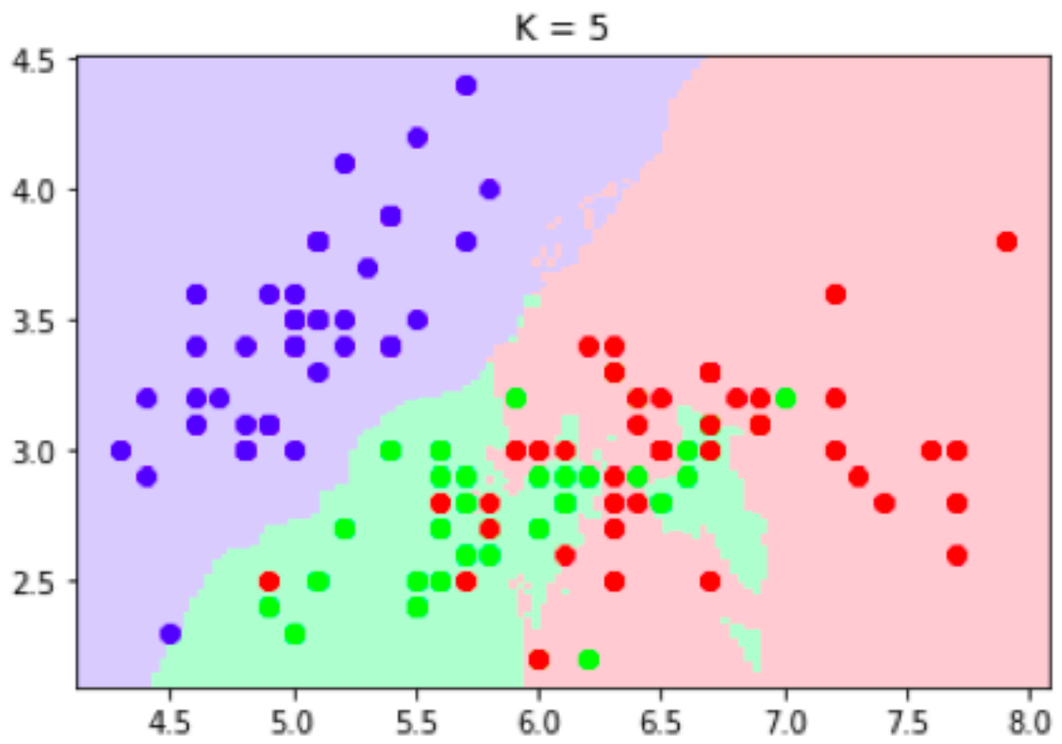
Можно ли предложить подобное для случая ML?

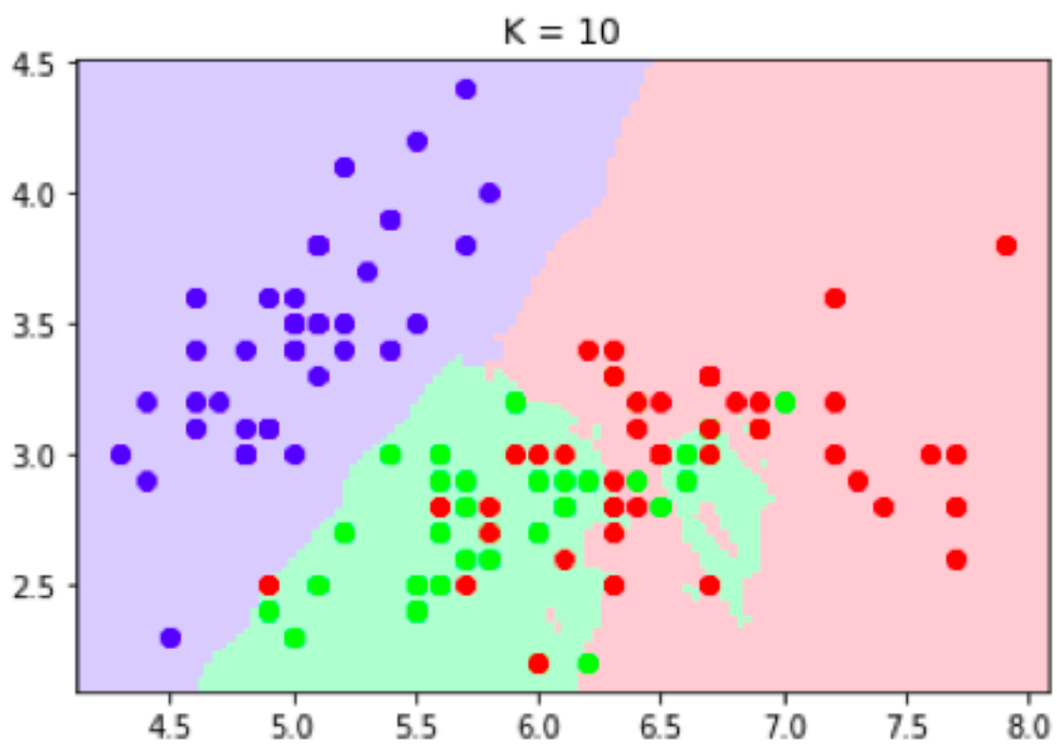
Bias-variance tradeoff

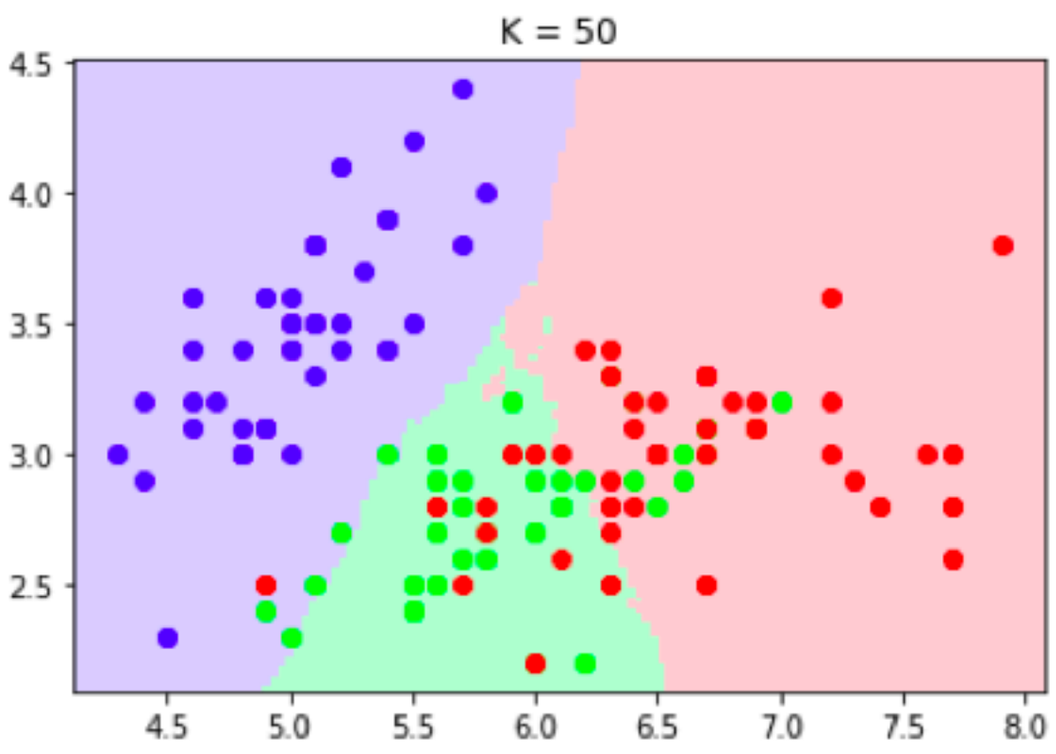


K-nearest neighbors

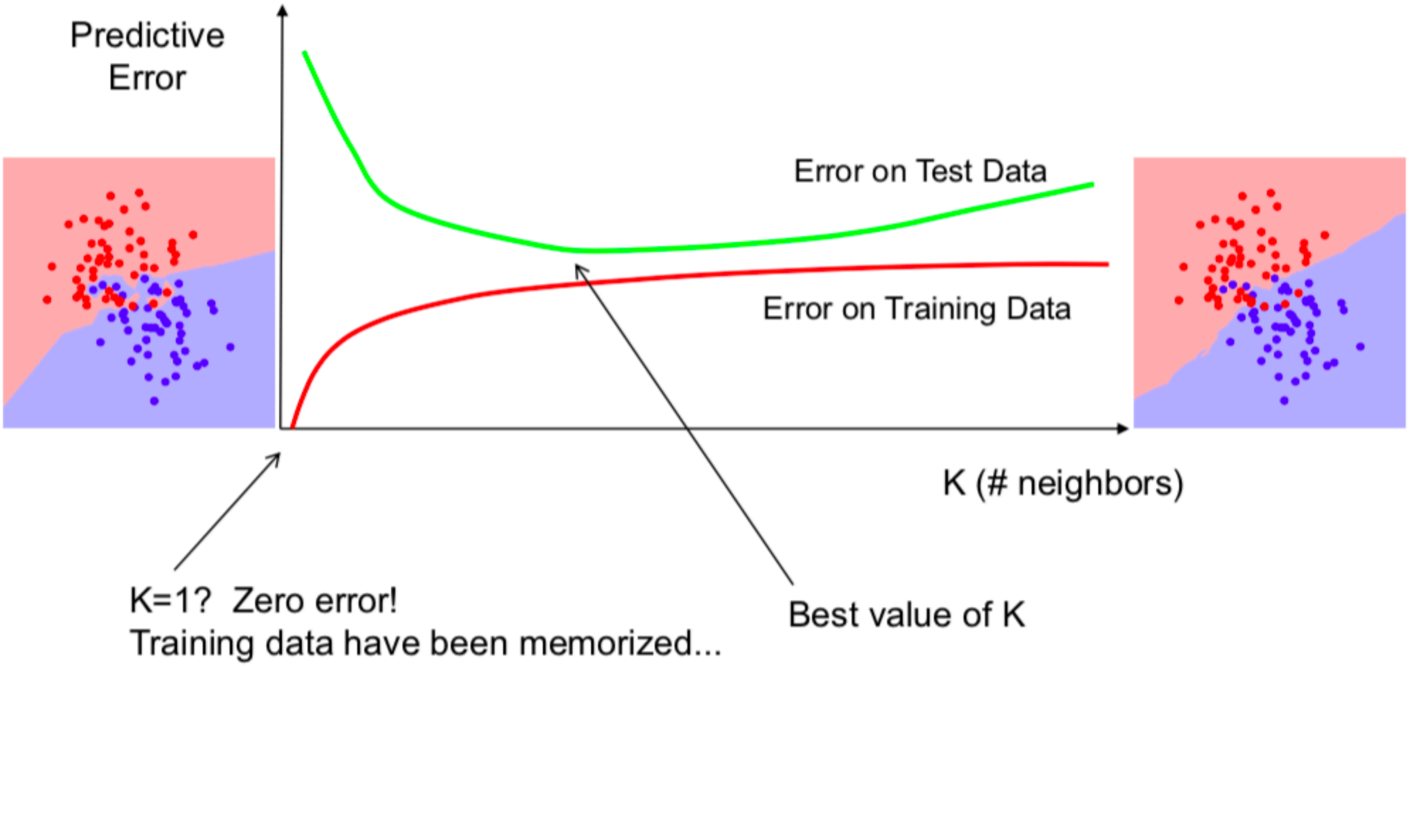








Error rates and K



Bias-variance tradeoff. Случай регрессионой модели.

Напоминание:

$$Var(x) = \mathbb{E}[x^2] - \mathbb{E}^2[x]$$

$$\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y] + Cov(x, y)$$

$$Var(x + y) = Var(x) + Var(y) + 2Cov(x, y)$$

$$Var(x - y) = Var(x) + Var(y) - 2Cov(x, y)$$

$$Cov(x, y) = 0 \text{ if } x \text{ and } y \text{ are independent}$$

Случай регрессора

Оцениваем значение y при помощи функции f(x)

$$MSE = \mathbb{E}\left[(y - \hat{f}_S(x))^2\right] = \mathbb{E}[y^2] + \mathbb{E}[\hat{f}_S^2(x)] - 2\mathbb{E}[y\hat{f}_S(x)]$$
$$= Var(y) + \mathbb{E}^2[y] + Var(\hat{f}_S(x)) + \mathbb{E}^2[\hat{f}_S(x)] - 2\mathbb{E}[y\hat{f}_S(x)]$$
$$= Var(y) + \mathbb{E}^2[y] + Var(\hat{f}_S(x)) + \mathbb{E}^2[\hat{f}_S(x)] - 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)]$$
$$- 2\mathbb{E}[f(x)\hat{f}_S(x)]$$

Случай регрессора

$$
\begin{aligned}
=&Var(y) + \mathbb{E}^2[y] + Var(\hat{f}_S(x)) + \mathbb{E}^2[\hat{f}_S(x)] - 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)] \\
&- 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}_S(x)] - 2Cov(f(x), \hat{f}_S(x)) \\
=&Var(f(x)) + Var(\epsilon) + \mathbb{E}^2[y] + Var(\hat{f}_S(x)) + \mathbb{E}^2[\hat{f}_S(x)] \\
&- 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)] - 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}_S(x)] - 2Cov(f(x), \hat{f}_S(x)) \\
=&Var(f(x) - \hat{f}_S(x)) + Var(\epsilon) + \mathbb{E}^2[y] + \mathbb{E}^2[\hat{f}_S(x)] \\
&- 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)] - 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}_S(x)] \\
=&Var(f(x) - \hat{f}_S(x)) + Var(\epsilon) + \mathbb{E}^2[f(x)] + \mathbb{E}^2[\epsilon] + 2\mathbb{E}[\epsilon]\mathbb{E}[f(x)] \\
&+ \mathbb{E}^2[\hat{f}_S(x)] - 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)] - 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}_S(x)] \\
=&Var(f(x) - \hat{f}_S(x)) + Var(\epsilon) + \Big(\mathbb{E}[f(x)] - \mathbb{E}[\hat{f}_S(x)]\Big)^2 \\
&+ \mathbb{E}^2[\epsilon] + 2\mathbb{E}[\epsilon]\mathbb{E}[f(x)] - 2\mathbb{E}[\epsilon]\mathbb{E}[\hat{f}_S(x)]
\end{aligned}
$$

Случай регрессора

$$
\begin{aligned}
=&Var(f(x) - \hat{f}_S(x)) \\
&+ Var(\epsilon) \\
&+ \left(\mathbb{E}[f(x)] - \mathbb{E}[\hat{f}_S(x)]\right)^2
\end{aligned}
$$

K-nearest neighbours

Как делать регрессию?

K-nearest neighbours

Как делать регрессию?

Просто усредняем значения у соседей

K-nearest neighbours

Как делать регрессию?

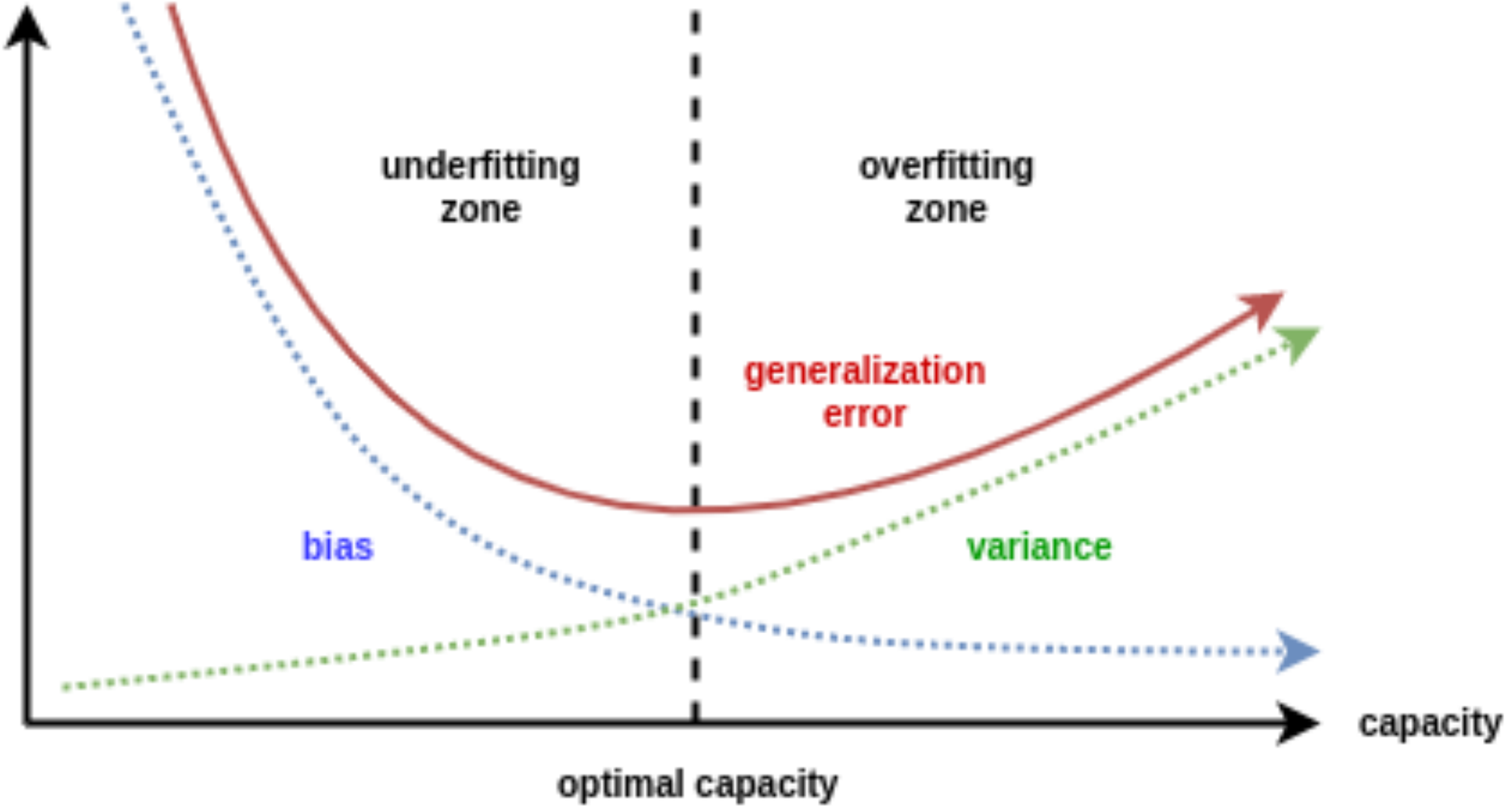
Просто усредняем значения у соседей

Либо можем брать взвешенное среднее - чем сосед ближе, тем больший вклад вносит

K-nearest neighbours

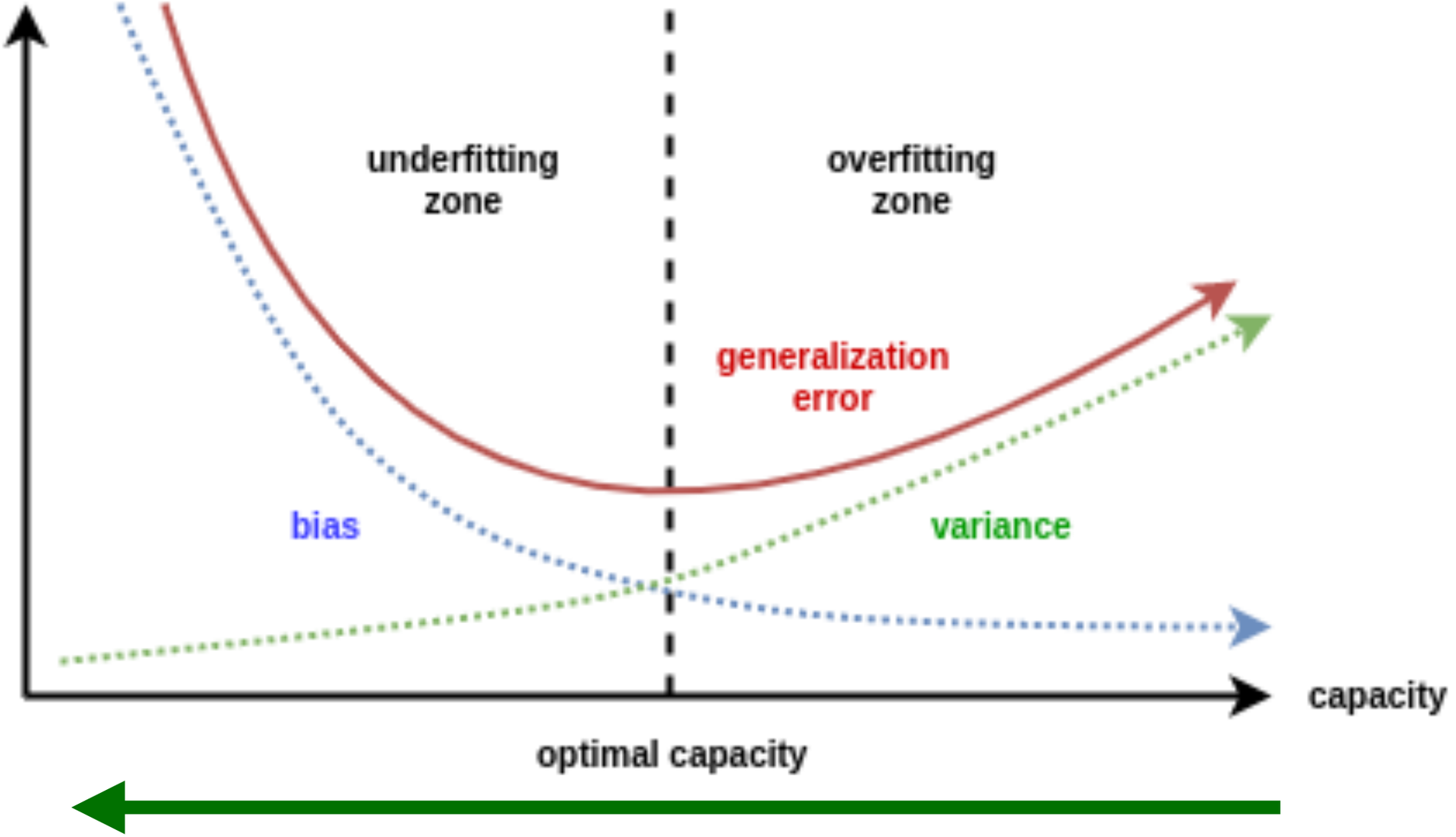
$$Err(x_o) = \sigma_e^2 + [f(x_o) - \frac{1}{k}\sum_{l=1}^{k} f(x_o)]^2 + \frac{\sigma_e^2}{k}$$

Bias-variance tradeoff, KNN



Как тут расположена ось k (число соседей) ?

Bias-variance tradeoff, KNN



Число соседей