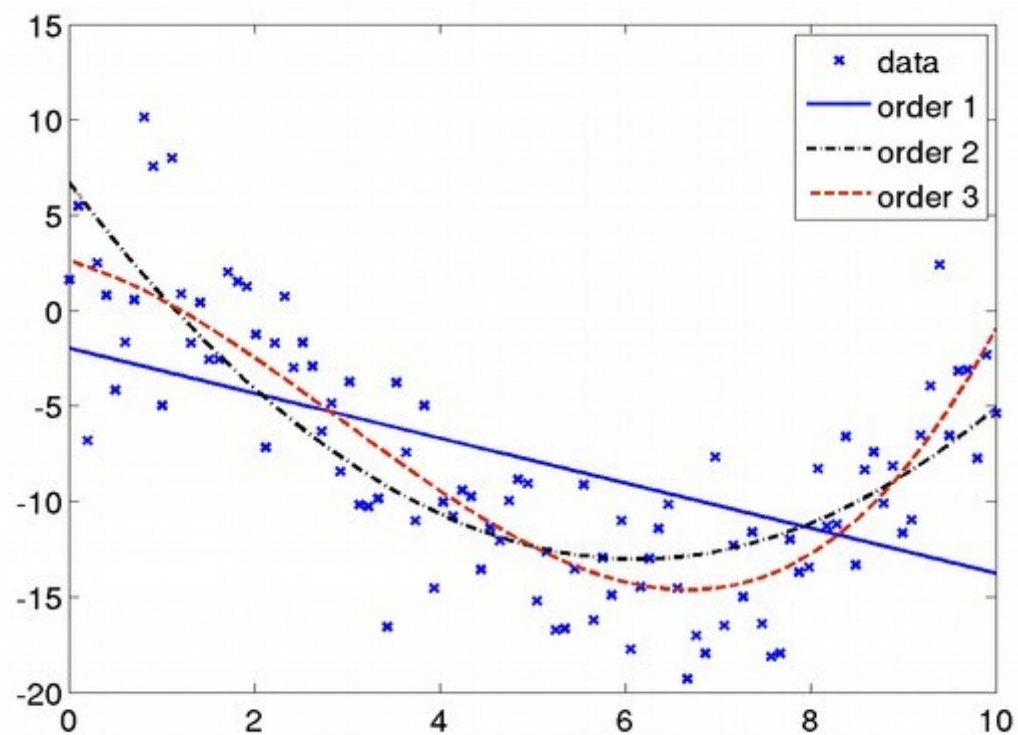


# Машинное обучение

## Основные понятия



# Содержание лекции

- Задача обучения
- Матрица объектов-признаков
- Модель алгоритмов и метод обучения
- Функционал качества
- Проблема переобучения

# Задача обучения

$X$  — множество объектов

$Y$  — множество ответов

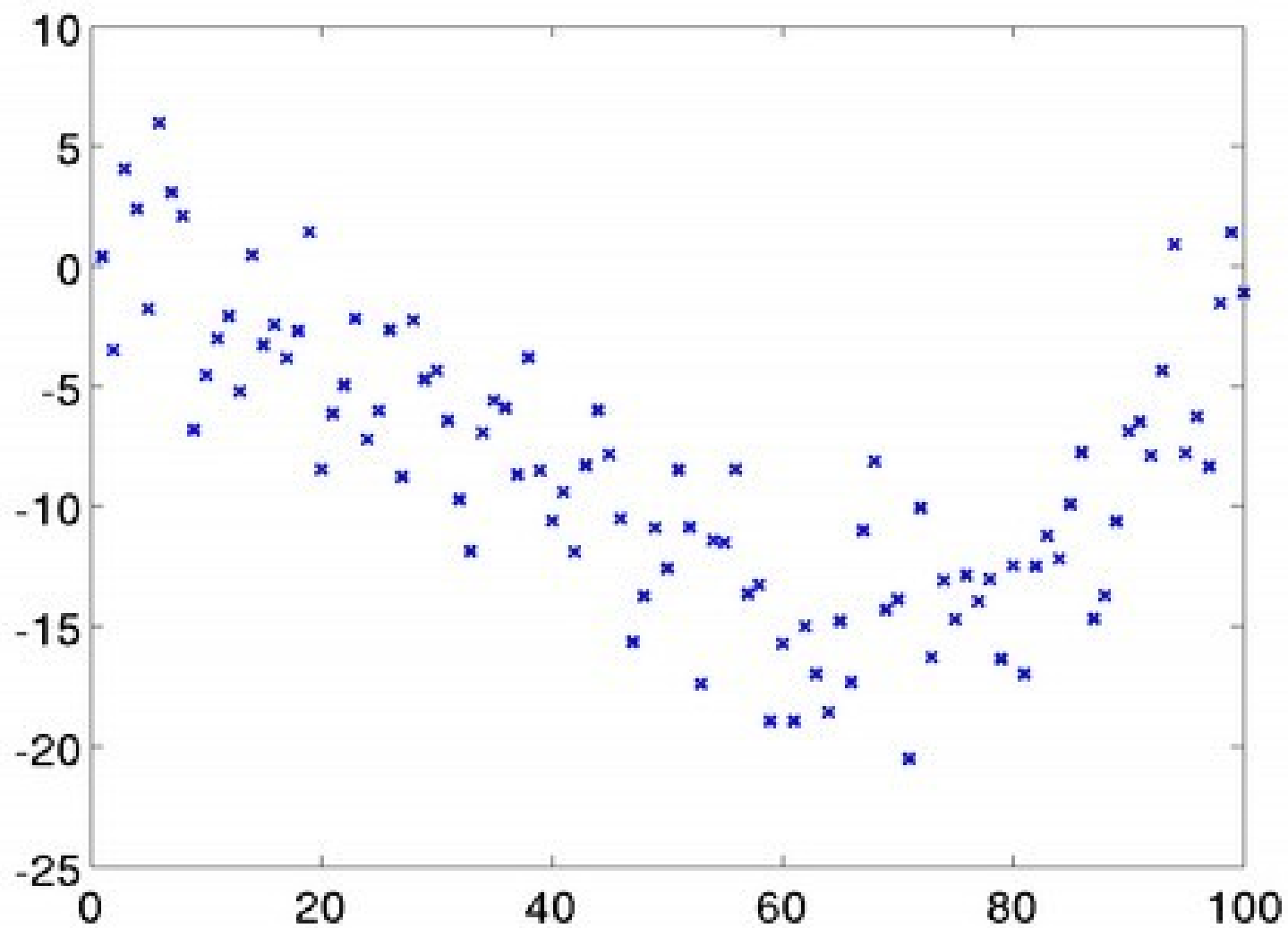
$y : X \rightarrow Y$  — неизвестная зависимость  
(target function)

**Дано:**

$\{x_1, \dots, x_\ell\} \subset X$  — обучающая выборка  
(training sample)

$y_i = y(x_i), i = 1, \dots, \ell$  — известные ответы

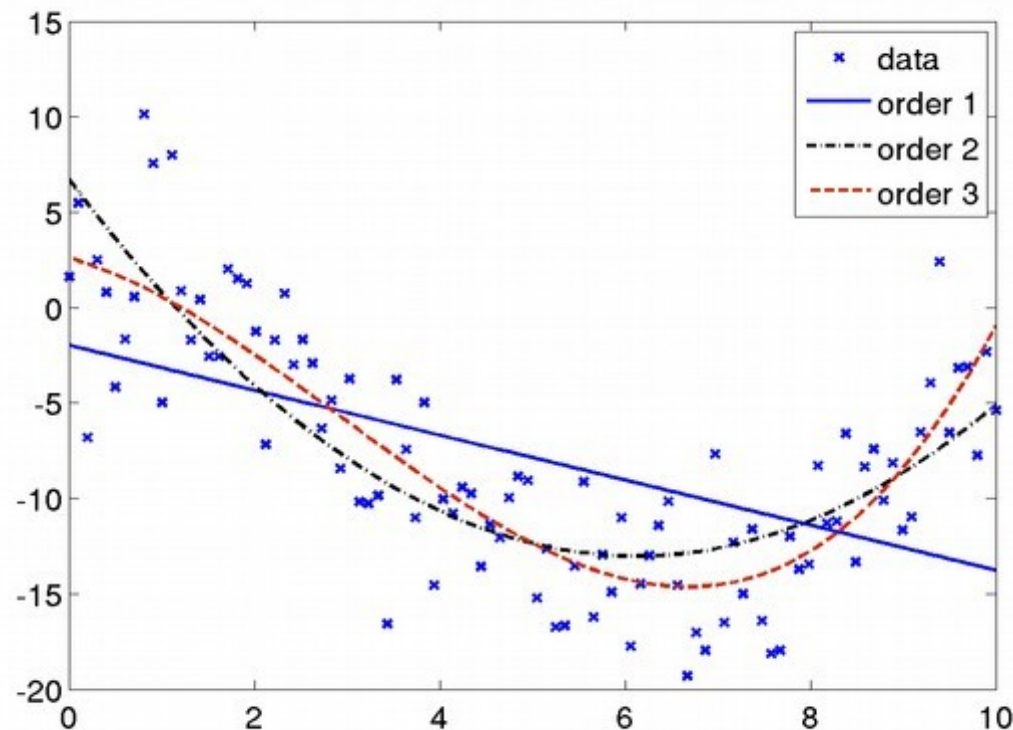
# Задача обучения



# Задача обучения

Найти:

$a : X \rightarrow Y$  — алгоритм, решающую функцию (decision function), приближающую  $y$  на всём множестве  $X$



# Типы задач

## Задачи классификации (classification):

$Y = \{-1, +1\}$  — классификация на 2 класса

$Y = \{1, \dots, M\}$  — на  $M$  непересекающихся классов (multi-class classification)

$Y = \{0, 1\}^M$  — на  $M$  классов, которые могут пересекаться (multi-label classification).

## Задачи восстановления регрессии (regression):

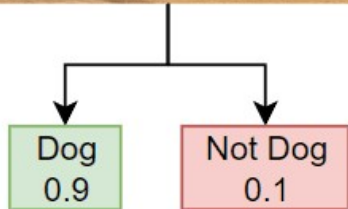
$Y = \mathbb{R}$  или  $Y = \mathbb{R}^m$

## Задачи ранжирования (ranking):

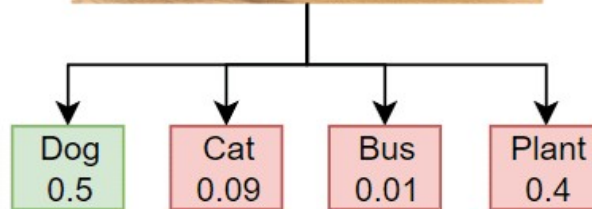
$Y$  — конечное упорядоченное множество

# Типы классификаций

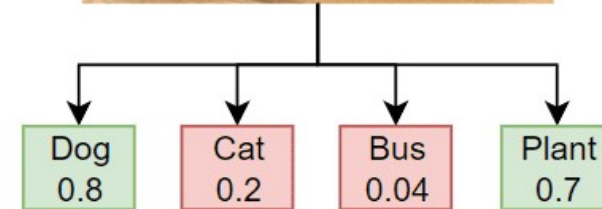
Binary Classification



Multiclass Classification



Multilabel Classification

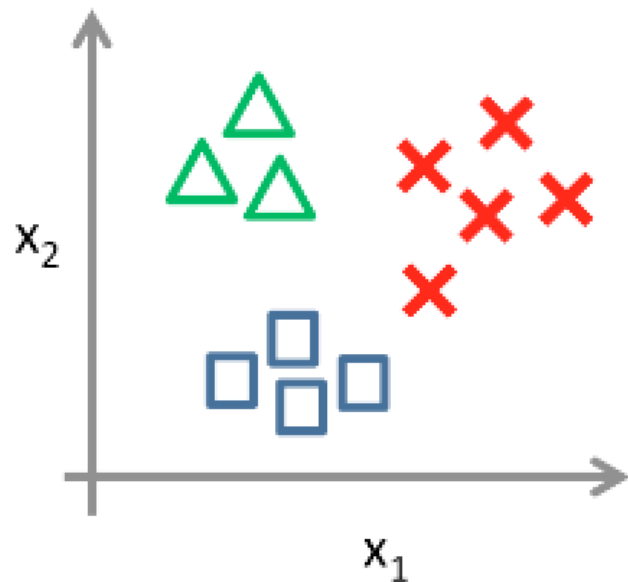





# Выбор между multi-class и multi-label

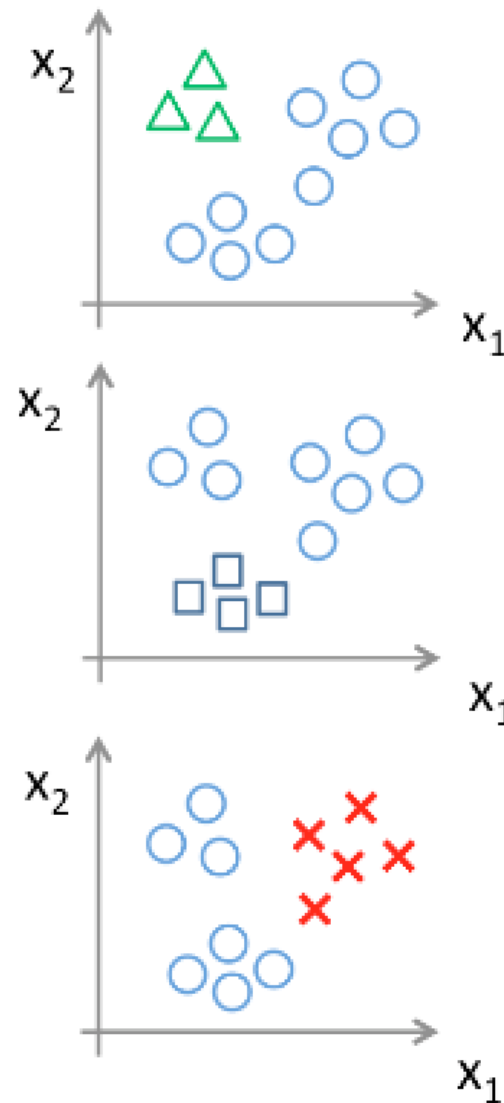
- Пример нетривиальной задачи: классификация участков гистологий. Патологоанатомы разметили знакомые характерные участки в БД изображений названиями известных патологий. Нужно натренировать нейросеть делать это автоматически
- Чем будут отличаться результаты предсказания multi-class и multi-label нейросетей?
- Если мы впоследствии будем сортировать по вероятности принадлежности тому или иному классу, какой тип классификации правильно будет применять?

# Сведение многоклассовой к бинарной классификации

One-vs-all (one-vs-rest):









Class 1:   
Class 2:   
Class 3: 









# Кодирование класса

## Multi-Class

C = 3	
  	Samples
	  
	Labels $[0 \ 0 \ 1]$ $[1 \ 0 \ 0]$ $[0 \ 1 \ 0]$ one-hot encoding

## Multi-Label

C = 3	
  	Samples
	  
	Labels $[1 \ 0 \ 1]$ $[0 \ 1 \ 0]$ $[1 \ 1 \ 1]$

# Признаки

- Компьютер всегда имеет дело с признаковым описанием объектов. Например: пациента можно описать признаками: имя, возраст, номер полиса, жалобы, давление, температура, результаты анализов

- $f : X \rightarrow D_f$

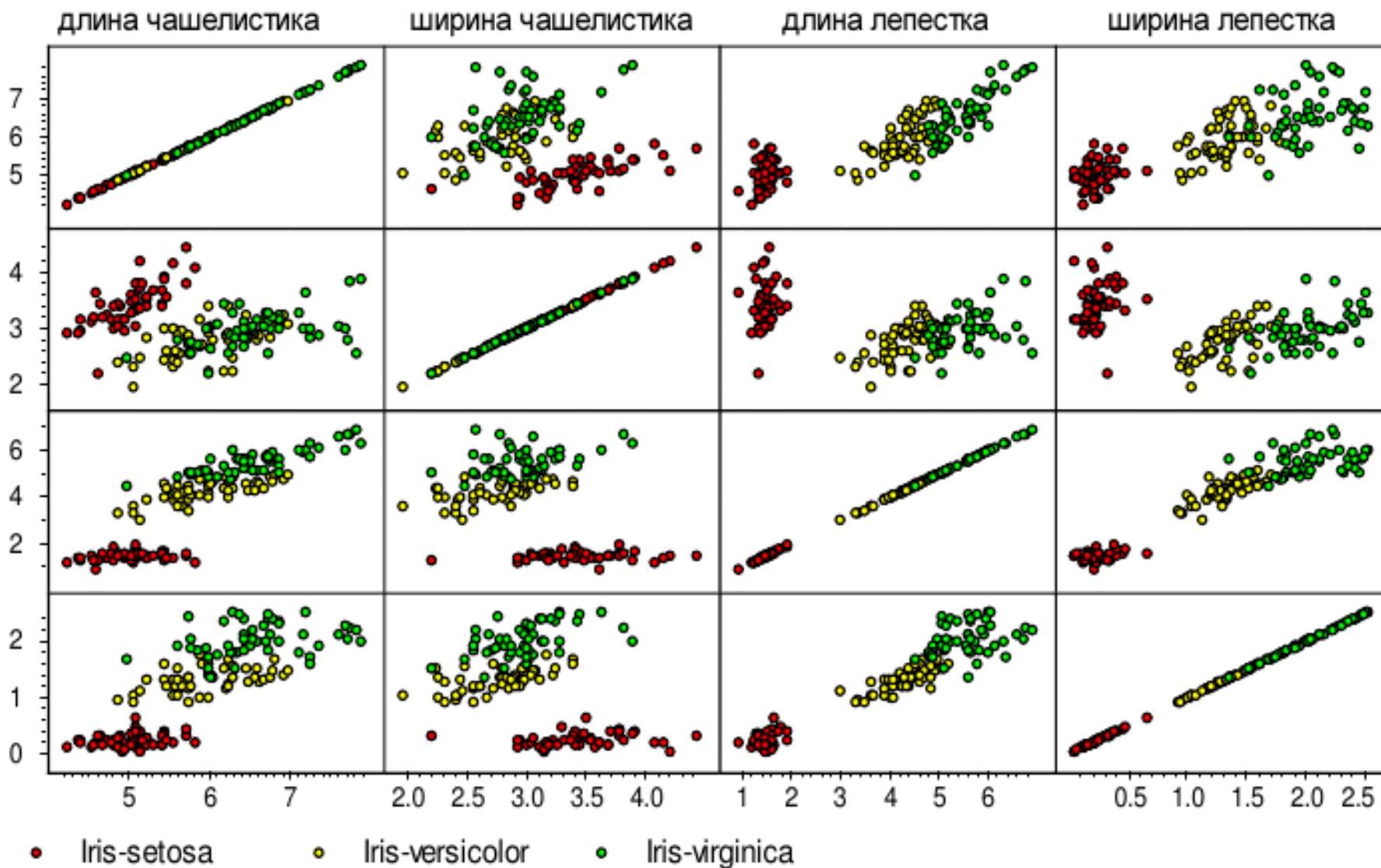
- Типы признаков:

- бинарный
- номинальный
- порядковый
- количественный

Матрица объектов-признаков:

$$\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

# Пример. Задача классификации видов ириса (Фишер 1936)



# Модель и алгоритм обучения

- **Модель** – это семейство “гипотез”

$$A = \{g(x, \theta) \mid \theta \in \Theta\}$$

одна из которых (как мы надеемся)  
хорошо приближает целевую функцию

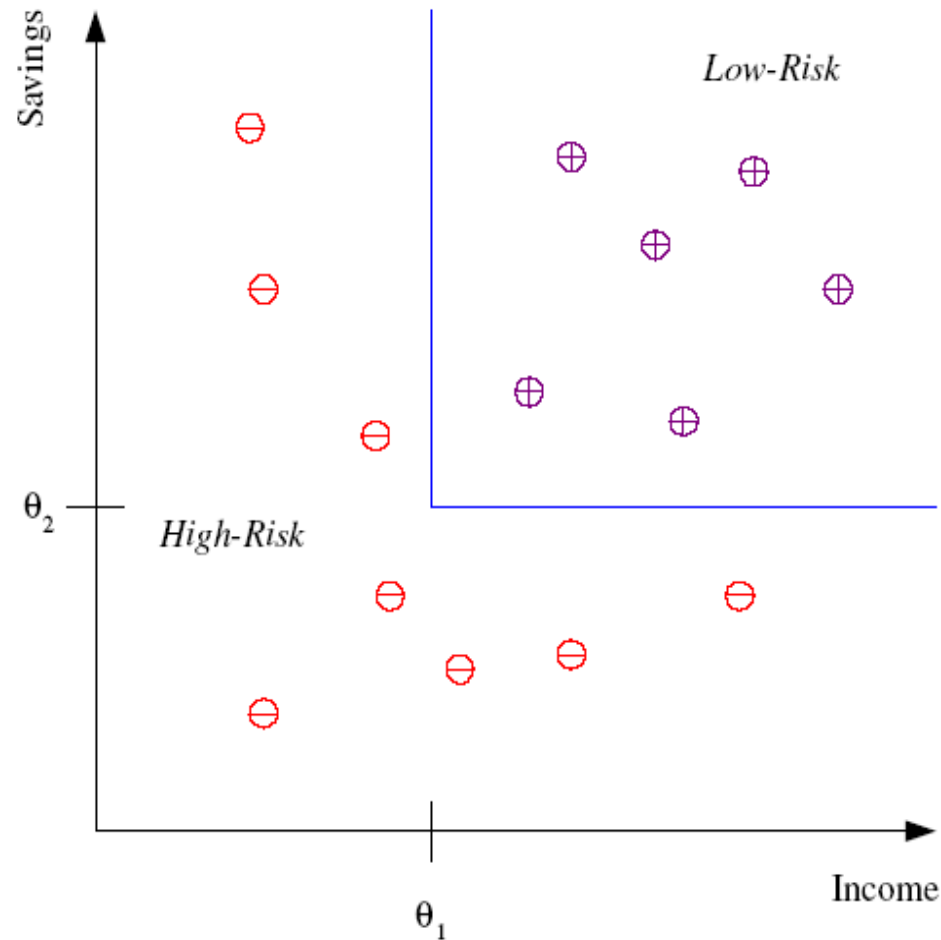
- **Алгоритм обучения**

$$\mu: (X \times Y)^\ell \rightarrow A$$

находит гипотезу в модели, которая  
наилучшим образом приближает  
целевую функцию, используя известные  
значения (обучающую выборку)

# Пример - классификация

- Кредитный скоринг
- Разделение клиентов на **low-risk** и **high-risk** по их зарплате и сбережениям

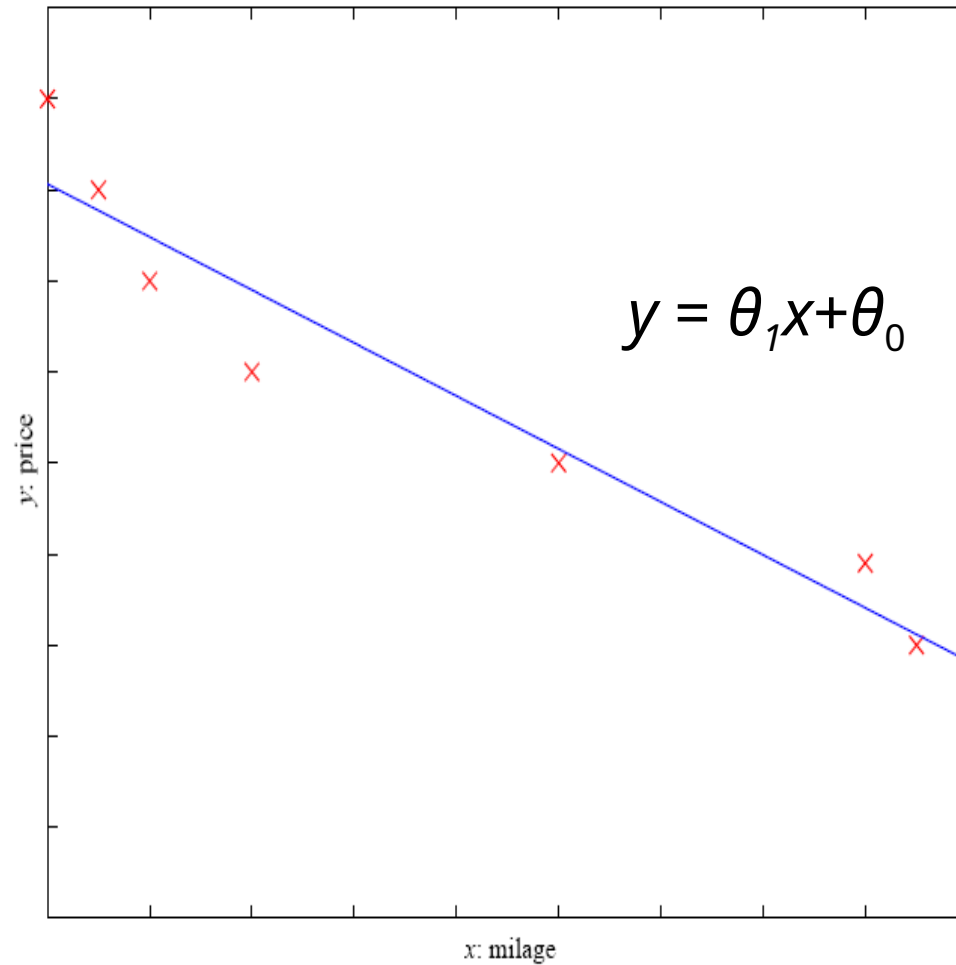


IF  $income > \theta_1$  AND  $savings > \theta_2$   
THEN **low-risk** ELSE **high-risk**

Модель

# Пример - регрессия

- $y$  - цена автомобиля
- $x$  - пробег
- $y = \theta_1 x + \theta_0$  - модель
- $\theta_0, \theta_1$  - параметры



# Обучение на основе минимизации эмпирического риска

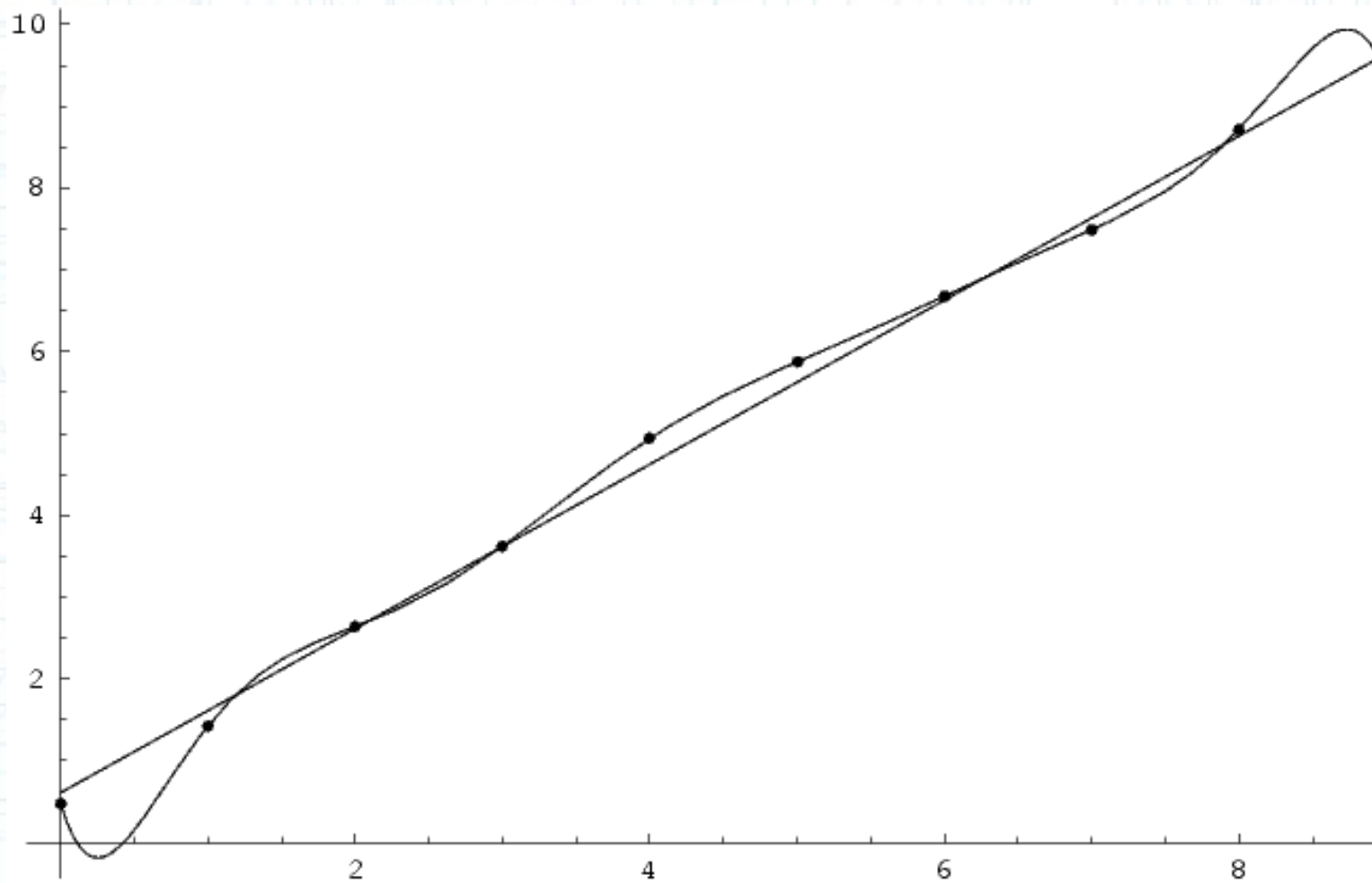
- Функция потерь  $\mathcal{L}(a(x), y^*(x))$  - величина ошибки гипотезы  $a$  на объекте  $x$ .  
Примеры:
  - бинарная (где используется?)
  - $\mathcal{L}(a(x), y^*(x)) = |a(x) - y^*(x)|$
  - $\mathcal{L}(a(x), y^*(x)) = (a(x) - y^*(x))^2$
- Эмпирический риск:  $Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(a(x_i), y_i)$
- Самый популярный алгоритм обучения – минимизация эмпирического риска:

$$\mu(X^\ell) = \arg \min_{a \in A} Q(a, X^\ell)$$

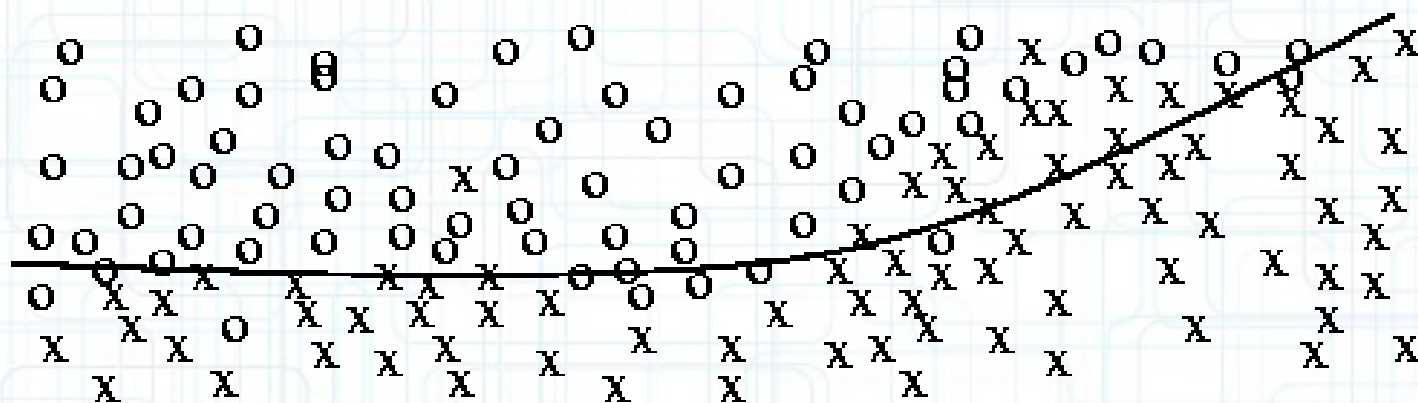
# Степени обученности модели

- Недообученная модель
  - Модель, слишком сильно упрощающая закономерность  $X \rightarrow Y$ .
- Переобученная модель
  - Модель, слишком сильно настроенная на особенности обучающей выборки (на шум в наблюдениях), а не на реальную закономерность  $X \rightarrow Y$ .

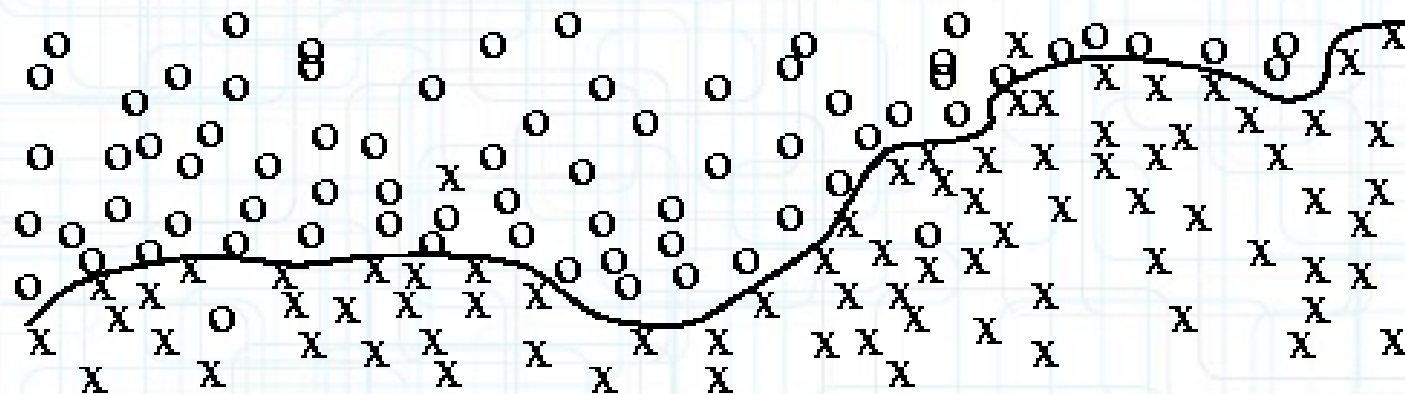
# Переобучение



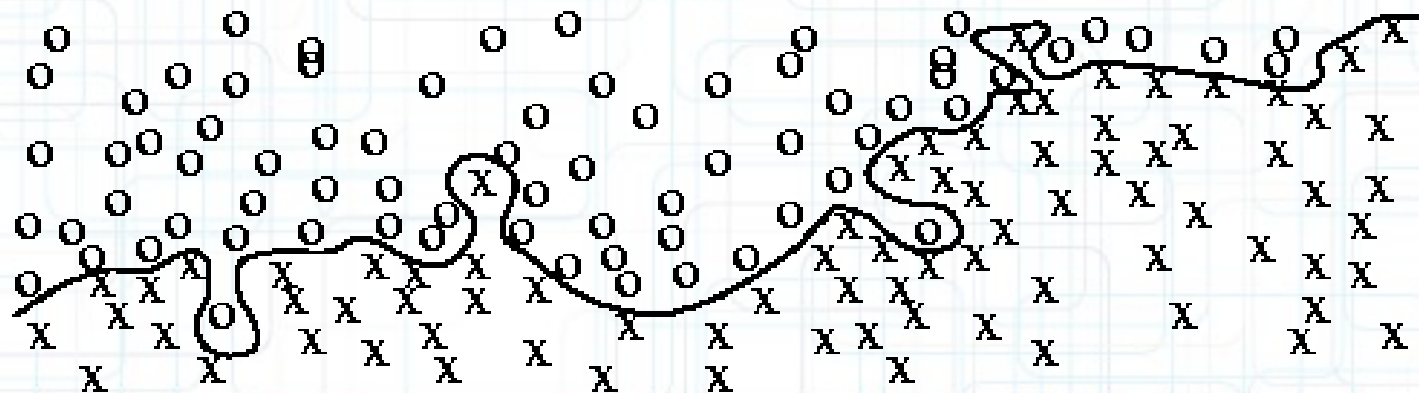
# Переобучение



Under-Trained

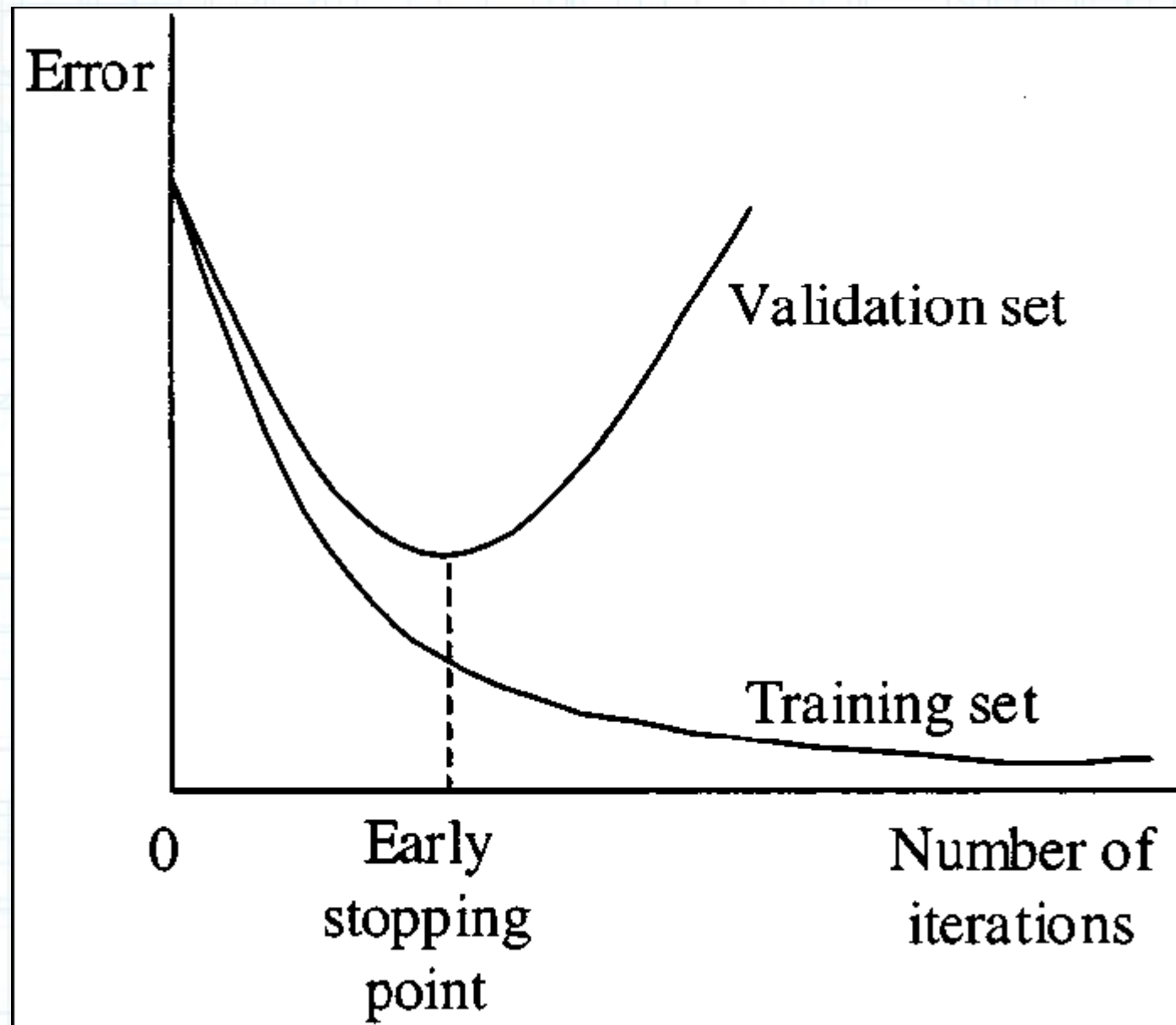


Well-Trained



Overfitted

# Когда нужно заканчивать обучение?



# Контроль переобучения

- Для оценки обобщающей способности алгоритма обучения  $\mu$  используют:
  - Эмпирический риск на тестовых данных (hold-out):

$$\text{HO}(\mu, X^\ell, X^k) = Q(\mu(X^\ell), X^k) \rightarrow \min$$

- Скользящий контроль (leave-one-out),  $L=\ell+1$ :

$$\text{LOO}(\mu, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(\mu(X^\ell \setminus \{x_i\})(x_i), y_i)$$

- Кросс-проверка (cross-validation):

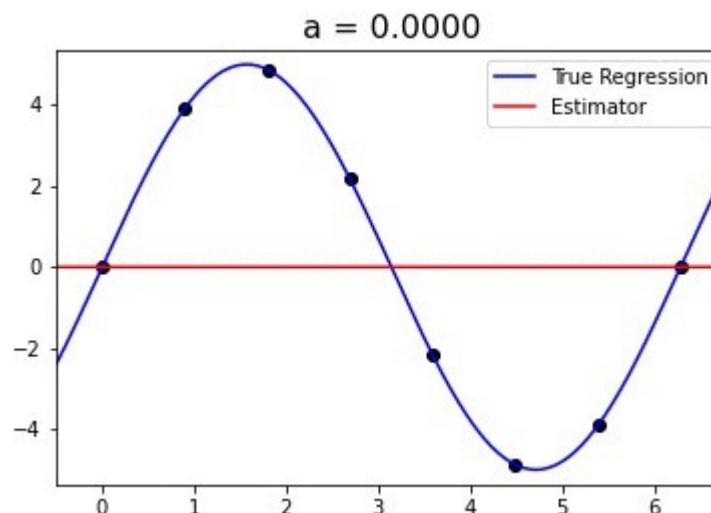
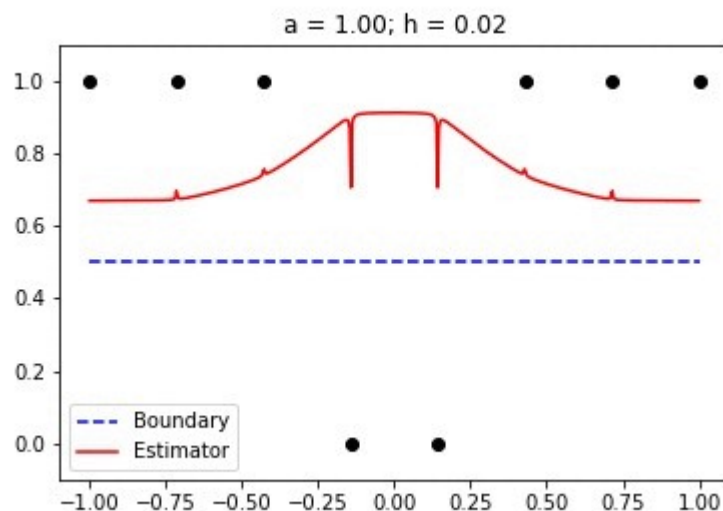
$$\text{CV}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q(\mu(X_n^\ell), X_n^k) \rightarrow \min$$

- Оценка вероятности переобучения:

$$Q_\varepsilon(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} \left[ Q(\mu(X_n^\ell), X_n^k) - Q(\mu(X_n^\ell), X_n^\ell) \geq \varepsilon \right] \rightarrow \min$$

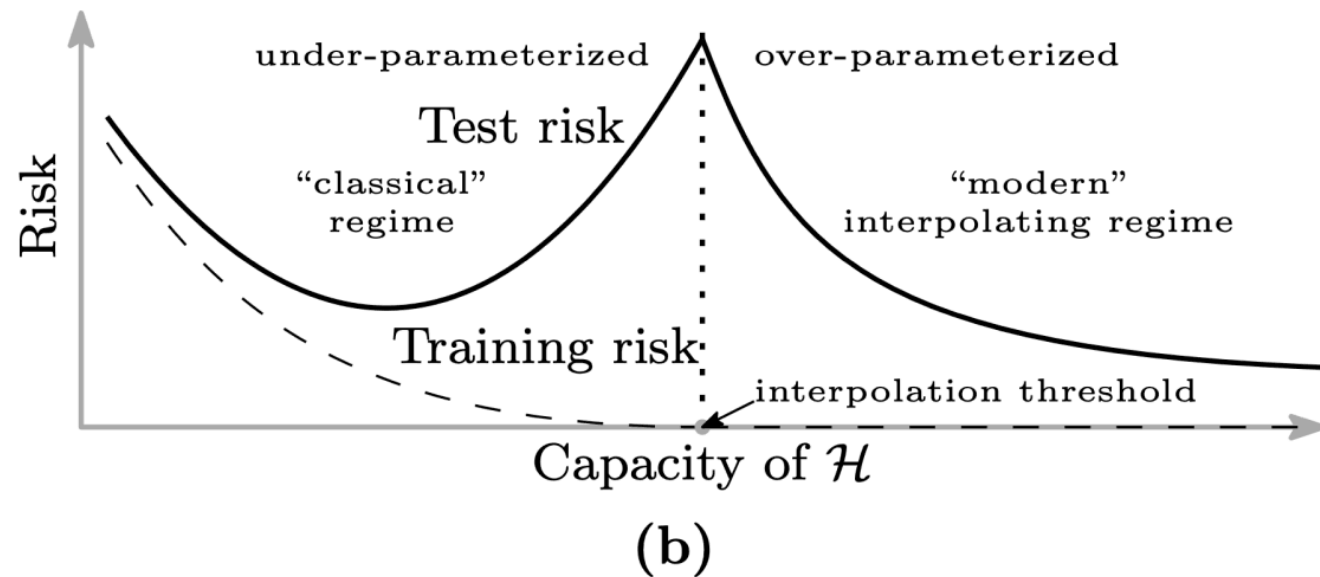
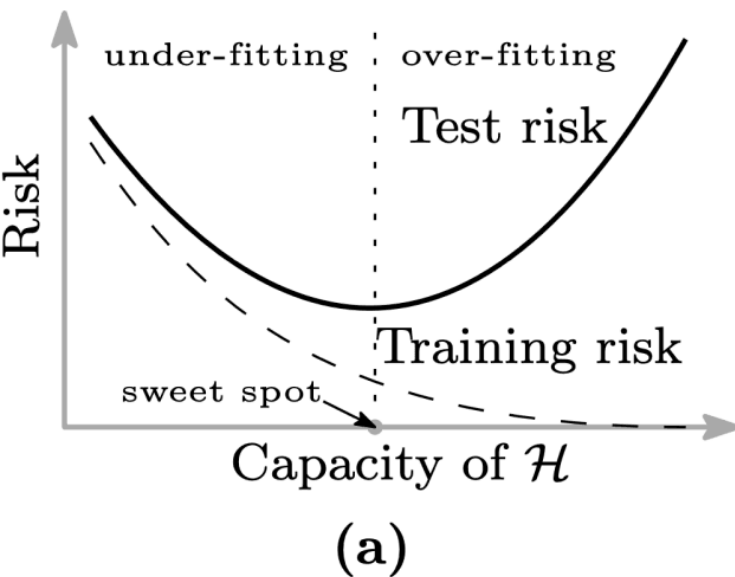
# Переобучение переобучению - рознь

Даже когда модель “запомнила” обучающую выборку, если ее правильно тренировать дальше, она повышает свою обобщающую способность



# Кривая риска для современных алгоритмов

**Double descent risk curve** - после достижения interpolation threshold алгоритм умеет идеально запоминать обучающую выборку, но неограниченного роста переобучения не происходит



# Ошибки кросс-валидации

- Данные, зависящие от времени, нужно разделять на "прошрое" и "будущее", а не пользоваться `train_test_split` да еще и с `shuffle=True`
- Аугментацию нужно проводить не перед, а после разделения на Train и Test (пример: генерация смесей, аугментация фото)

# Ошибки кросс-валидации

- Bias - статистическое отличие Train от Test (или от данных, к которым модель собираются применять в будущем).
- Примеры:
  - обучение на фото, собранных в ясную погоду;
  - наем сотрудников в Amazon (м/ж)
  - наш опыт сбора данных по титановым покрытиям
  - конкурсы Яндекса: пробки, панорамы