

# Лекция 4

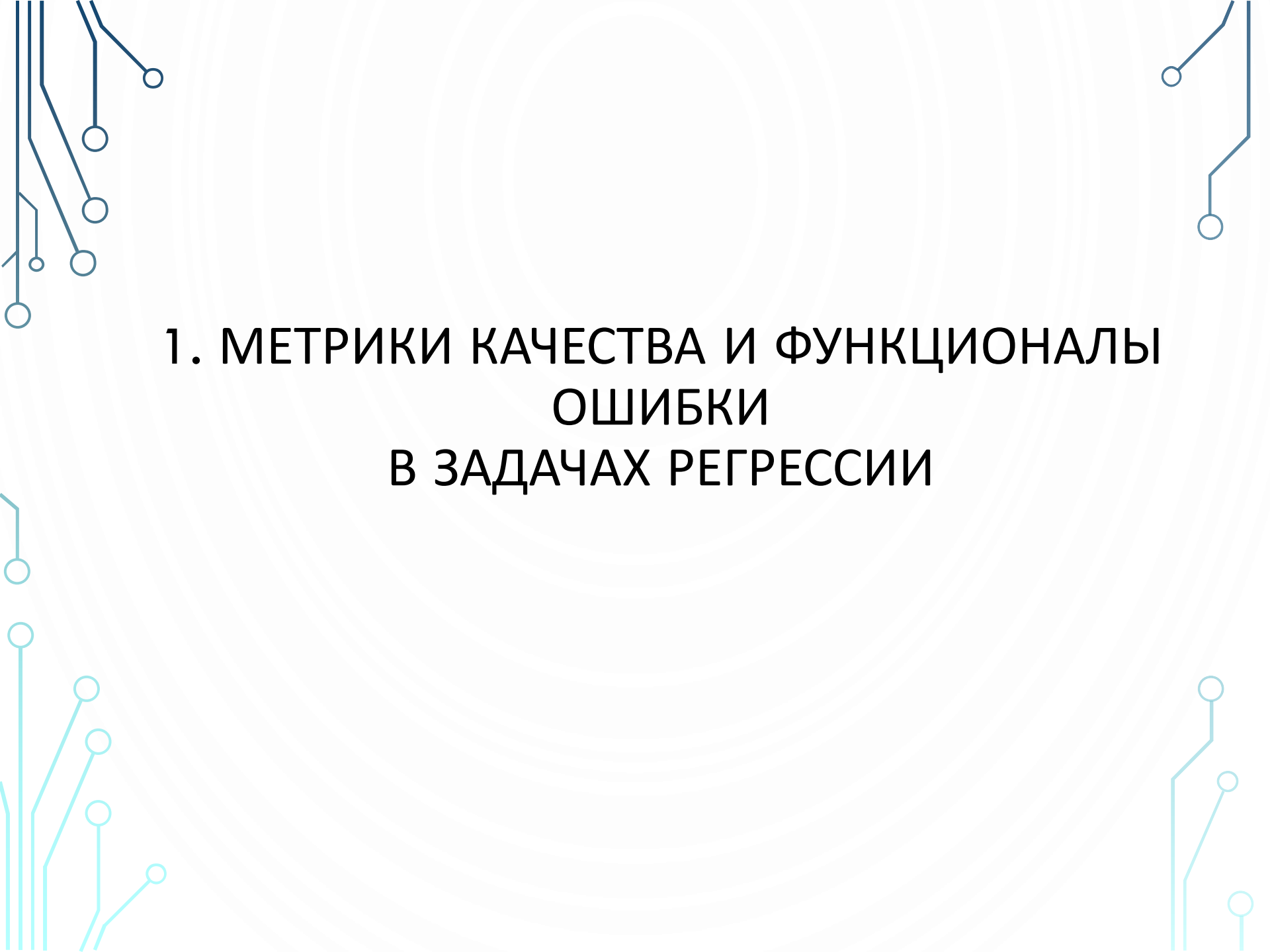
## Линейные методы регрессии. Часть 2.

Кантонистова Е.О.

ВШЭ, 2021

# ПЛАН ЛЕКЦИИ

- метрики качества и функционалы ошибки в задаче регрессии
- признаки переобученной модели и методы выявления переобучения и борьбы с ним: кросс-валидация и регуляризация, полезное свойство  $l_1$ -регуляризации

The slide features a light gray background with a subtle pattern of concentric circles. In the four corners, there are decorative elements resembling circuit board traces or neural network connections, consisting of thin blue lines and small circles.

# 1. МЕТРИКИ КАЧЕСТВА И ФУНКЦИОНАЛЫ ОШИБКИ В ЗАДАЧАХ РЕГРЕССИИ

# МЕТРИКИ КАЧЕСТВА И ФУНКЦИИ ОШИБКИ

- **Функционал (функция) ошибки** – функция, которую минимизируют в процессе обучения модели для нахождения неизвестных параметров (весов).
- **Метрика качества** – функция, которую используют для оценки качества построенной (уже обученной) модели.

# МЕТРИКИ КАЧЕСТВА И ФУНКЦИИ ОШИБКИ

- **Функционал (функция) ошибки** – функция, которую минимизируют в процессе обучения модели для нахождения неизвестных параметров (весов).
- **Метрика качества** – функция, которую используют для оценки качества построенной (уже обученной) модели.

*Иногда одна и та же функция может использоваться и для обучения модели (функция ошибки), и для оценки качества модели (метрика качества).*

# ЛИНЕЙНАЯ РЕГРЕССИЯ

**Линейная регрессия:**

$$a(x) = w_0 + \sum_{j=1}^d w_j x_j$$

**Обучение линейной регрессии** - минимизация  
среднеквадратичной ошибки:

$$\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 \rightarrow \min_w$$

# СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE (MEAN SQUARED ERROR)

Среднеквадратичное отклонение:

$$MSE(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2$$

# СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE (MEAN SQUARED ERROR)

Среднеквадратичное отклонение:

$$MSE(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2$$

Плюсы:

- Позволяет сравнивать модели
- Подходит для контроля качества во время обучения



# СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE

Среднеквадратичное отклонение:

$$MSE(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2$$

Плюсы:

- Позволяет сравнивать модели
- Подходит для контроля качества во время обучения

Минусы:

- Плохо интерпретируется, т.к. не сохраняет единицы измерения (если целевая переменная – кг, то MSE измеряется в кг в квадрате)
- Тяжело понять, насколько хорошо данная модель решает задачу, так как MSE не ограничена сверху.

# RMSE (ROOT MEAN SQUARED ERROR)

Корень из среднеквадратичной ошибки:

$$RMSE(a, X) = \sqrt{\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2}$$

Плюсы:

- Все плюсы MSE
- **Сохраняет единицы измерения (в отличие от MSE)**

Минусы:

- Тяжело понять, насколько хорошо данная модель решает задачу, так как RMSE не ограничена сверху.

# КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ ( $R^2$ )

Коэффициент детерминации:

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^l (a(x_i) - y_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2},$$

где  $\bar{y} = \frac{1}{l} \sum_{i=1}^l y_i$ .

# КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ ( $R^2$ )

Коэффициент детерминации:

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^l (a(x_i) - y_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2},$$

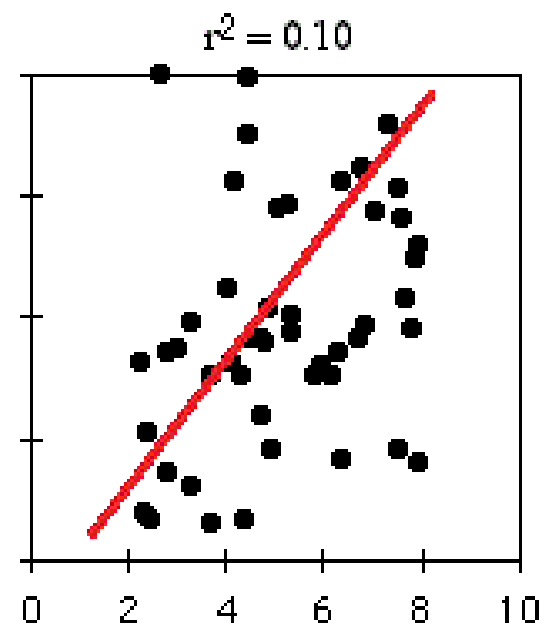
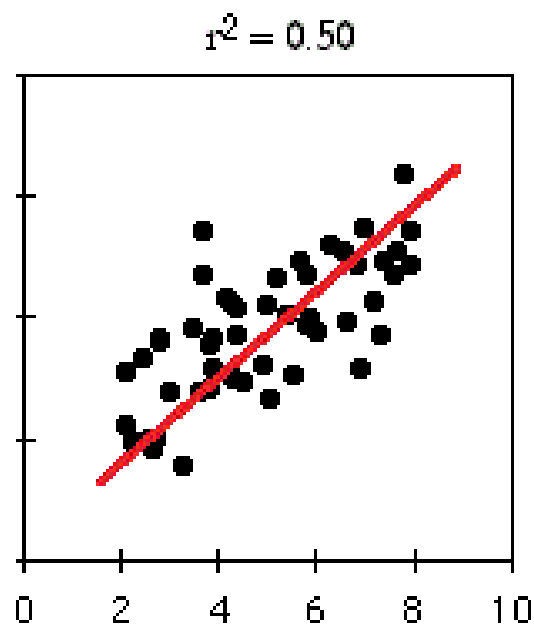
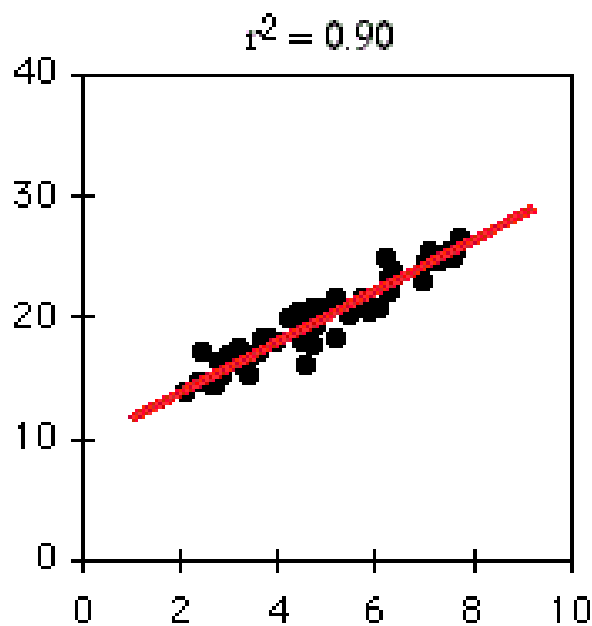
где  $\bar{y} = \frac{1}{l} \sum_{i=1}^l y_i$ .

Коэффициент детерминации это доля дисперсии целевой переменной, объясняемая моделью.

- Чем ближе  $R^2$  к 1, тем лучше модель объясняет данные
- Чем ближе  $R^2$  к 0, тем ближе модель к константному предсказанию
- Отрицательный  $R^2$  говорит о том, что модель плохо решает задачу

# КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ ( $R^2$ )

$$R^2 \leq 1$$



# MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

$$MAE(a, X) = \frac{1}{l} \sum_{i=1}^l |a(x_i) - y_i|$$

# MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

$$MAE(a, X) = \frac{1}{l} \sum_{i=1}^l |a(x_i) - y_i|$$

Плюсы:

- Менее чувствителен к выбросам, чем MSE

# MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

$$MAE(a, X) = \frac{1}{l} \sum_{i=1}^l |a(x_i) - y_i|$$

Плюсы:

- Менее чувствителен к выбросам, чем MSE

Минусы:

- MAE - не дифференцируемый функционал



# ОПТИМУМЫ MSE И MAE

Рассмотрим вероятностную постановку задачи.

Предположим, что на объектах с одинаковым признаковым описанием могут быть разные ответы. В этом случае на всех таких объектах MSE (или MAE) должна выдать один и тот же ответ.

**Теорема.** Пусть даны  $l$  объектов с одинаковым признаковым описанием и значениями целевой переменной  $y_1, \dots, y_l$ .

Тогда:

1. Оптимум MSE достигается на среднем значении ответов:

$$\alpha_{MSE} = \sum_{i=1}^l y_i$$

2. Оптимум MAE достигается на медиане ответов:

$$\alpha_{MAE} = \textit{median}\{y_1, \dots, y_l\}$$

# MSLE (MEAN SQUARED LOGARITHMIC ERROR)

Среднеквадратичная логарифмическая ошибка:

$$MSLE(a, X) = \frac{1}{l} \sum_{i=1}^l (\log(a(x_i) + 1) - \log(y + 1))^2$$

- Подходит для задач с неотрицательной целевой переменной ( $y \geq 0$ )
- Штрафует за отклонения в порядке величин
- Штрафует заниженные прогнозы сильнее, чем завышенные

# MAPE

*MAPE – Mean Absolute Percentage Error:*

$$MAPE(a, X) = \frac{1}{l} \sum_{i=1}^l \frac{|y_i - a(x_i)|}{|y_i|}$$

MAPE измеряет относительную ошибку.

# MAPE

$$MAPE(a, X) = \frac{1}{l} \sum_{i=1}^l \frac{|y_i - a(x_i)|}{|y_i|}$$

Плюсы:

- Ограничена:  $0 \leq MAPE \leq 1$
- Хорошо интерпретируема: например,  $MAPE=0.16$  означает, что ошибка модели в среднем составляет 16% от фактических значений.

# MAPE

$$MAPE(a, X) = \frac{1}{l} \sum_{i=1}^l \frac{|y_i - a(x_i)|}{|y_i|}$$

Плюсы:

- Ограничена:  $0 \leq MAPE \leq 1$
- Хорошо интерпретируема: например,  $MAPE=0.16$  означает, что ошибка модели в среднем составляет 16% от фактических значений.

Минусы:

- По-разному относится к недо- и перепрогнозу. Например, если правильный ответ  $y = 10$ , а прогноз  $a(x) = 20$ , то ошибка  $\frac{|10-20|}{|10|} = 1$ , а если ответ  $y = 30$ , то ошибка  $\frac{|30-20|}{|30|} = \frac{1}{3} \approx 0.33$ .

# SMAPE

SMAPE – *Symmetric Mean Absolute Percentage Error*  
(симметричный вариант MAPE):

$$SMAPE(a, X) = \frac{1}{l} \sum_{i=1}^l \frac{|y_i - a(x_i)|}{(|y_i| + |a(x_i)|)/2}$$

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

# SMAPE

SMAPE – *Symmetric Mean Absolute Percentage Error*  
(симметричный вариант MAPE):

$$SMAPE(a, X) = \frac{1}{l} \sum_{i=1}^l \frac{|y_i - a(x_i)|}{(|y_i| + |a(x_i)|)/2}$$

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

Проверим:

Пусть правильный ответ  $y = 10$ , а прогноз  $a(x) = 20$ , то

ошибка  $\frac{|10-20|}{|10+20|/2} = \frac{2}{3} \approx 0.67$ , а если ответ  $y = 30$ , то ошибка

$$\frac{|30-20|}{|30+20|/2} = \frac{2}{5} = 0.4.$$

# SMAPE

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

Проверим:

Пусть правильный ответ  $y = 10$ , а прогноз  $a(x) = 20$ , то

ошибка  $\frac{|10-20|}{|10+20|/2} = \frac{2}{3} \approx 0.67$ , а если ответ  $y = 30$ , то ошибка

$$\frac{|30-20|}{|30+20|/2} = \frac{2}{5} = 0.4.$$

***Ошибки стали меньше отличаться друг от друга, но всё-таки не равны.***



# SMAPE

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

*“Сейчас уже в среде прогнозистов сложилось более-менее устойчивое понимание, что SMAPE не является хорошей ошибкой. Тут дело не только в завышении прогнозов, но ещё и в том, что наличие прогноза в знаменателе позволяет манипулировать результатами оценки.” (см. [источник](#))*

# КВАНТИЛЬНАЯ РЕГРЕССИЯ

Квантильная функция потерь:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \rho_\tau(y_i - a(x_i))$$

Здесь

$$\rho_\tau(z) = (\tau - 1)[z < 0]z + \tau[z \geq 0]z = (\tau - \frac{1}{2})z + \frac{1}{2}|z|$$

Параметр  $\tau \in [0; 1]$ .

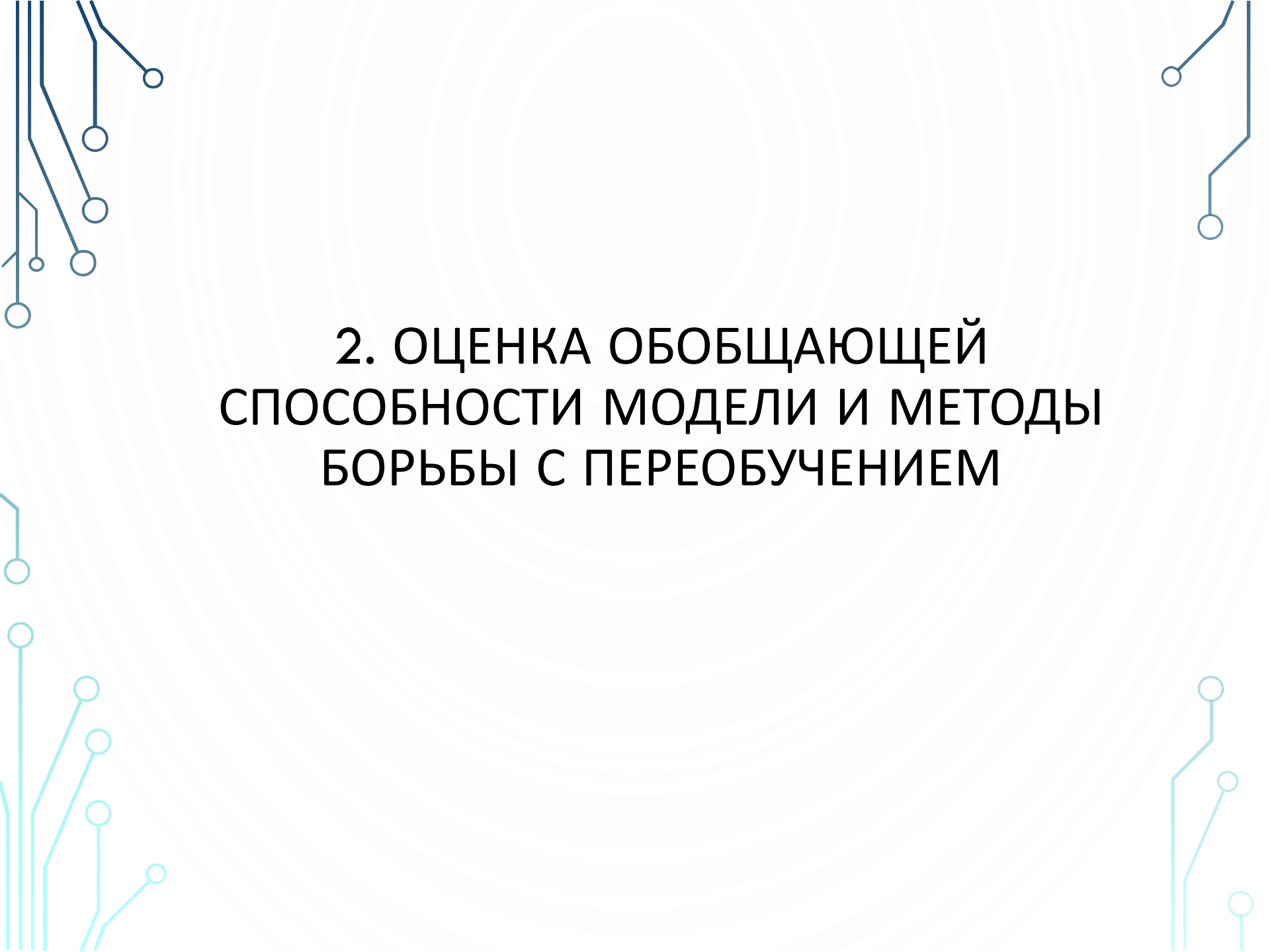
- Чем больше  $\tau$ , тем больше штрафует за занижение прогноза.

# ВЕРОЯТНОСТНЫЙ СМЫСЛ КВАНТИЛЬНОЙ ФУНКЦИИ ПОТЕРЬ

## Теорема.

Пусть в каждой точке  $x \in X$  (пространство объектов) задано распределение  $p(y|x)$  на ответах для данного объекта.

Тогда оптимизация функции потерь  $\rho_\tau(z)$  дает алгоритм  $a(x)$ , приближающий  $\tau$ -квантиль распределения ответов в каждой точке  $x \in X$ .

The slide features a light blue background with a subtle pattern of concentric circles. In the four corners, there are decorative elements resembling circuit board traces or neural network connections, consisting of thin blue lines and small circles.

## 2. ОЦЕНКА ОБОБЩАЮЩЕЙ СПОСОБНОСТИ МОДЕЛИ И МЕТОДЫ БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ

# ОЦЕНКА ОБОБЩАЮЩЕЙ СПОСОБНОСТИ МОДЕЛИ

**Переобучение (overfitting)** – явление, при котором качество модели на новых данных сильно хуже, чем качество на тренировочных данных.

Fitting training data

Degree = 1

— True function  
— Model  
• Training data (MSE = 1.37)

Underfitting

Degree = 2

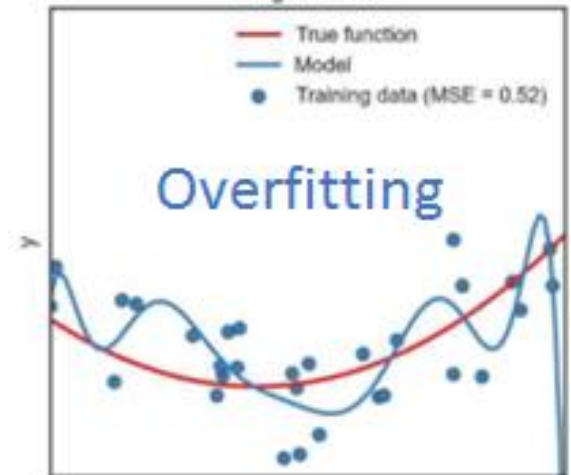
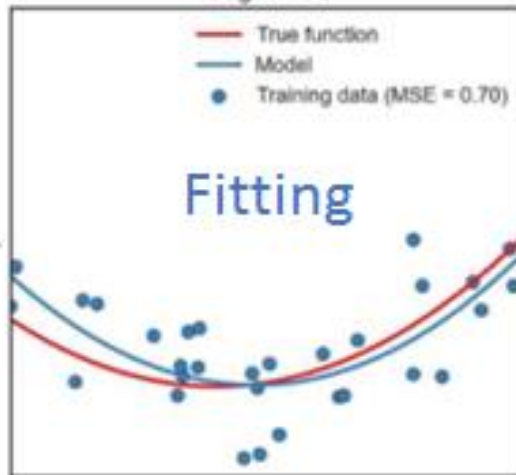
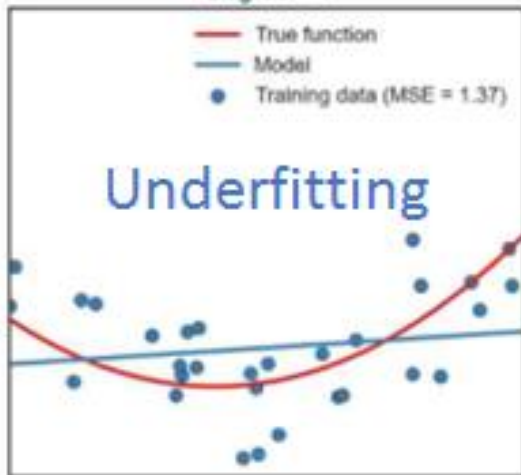
— True function  
— Model  
• Training data (MSE = 0.70)

Fitting

Degree = 10

— True function  
— Model  
• Training data (MSE = 0.52)

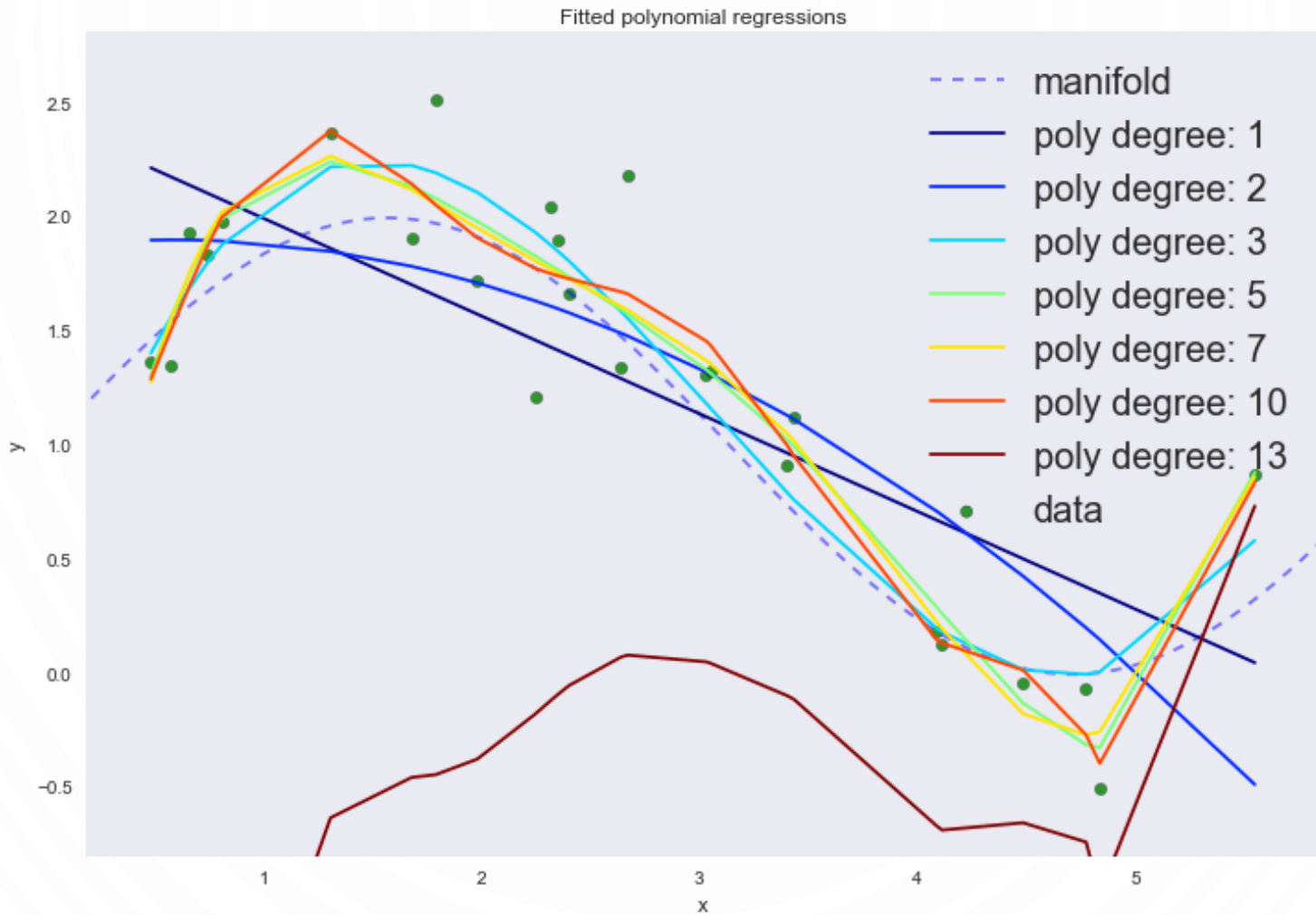
Overfitting



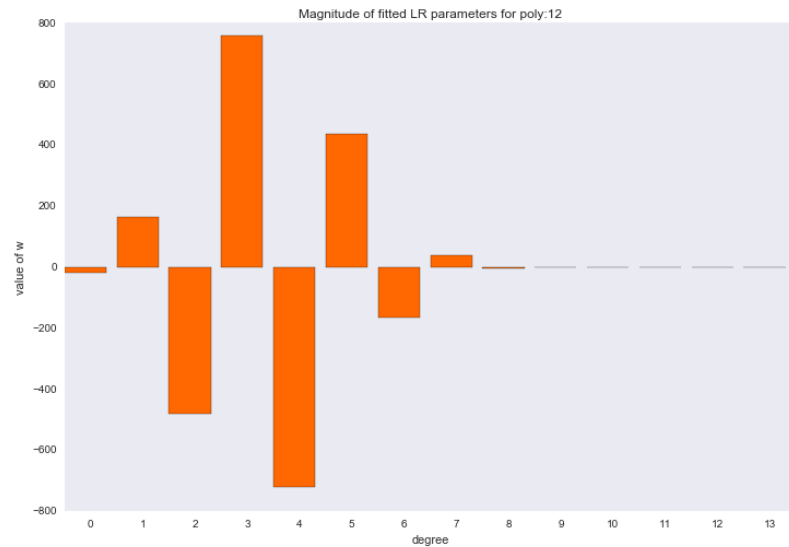
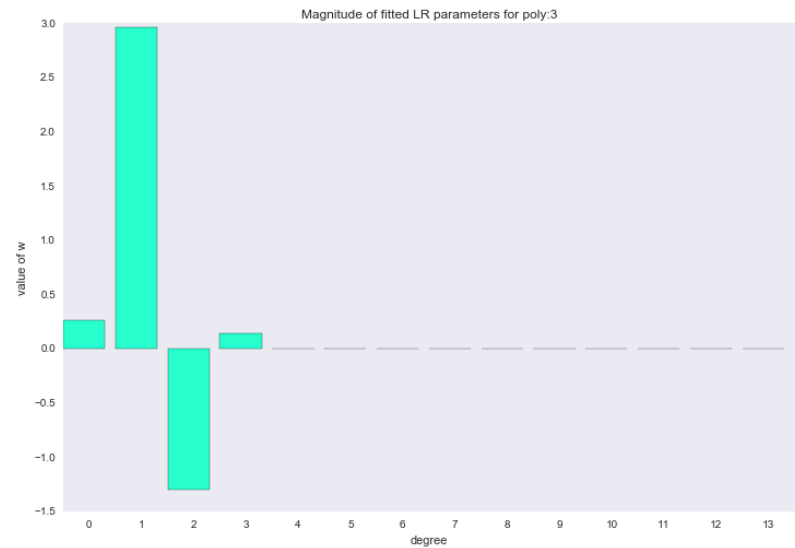
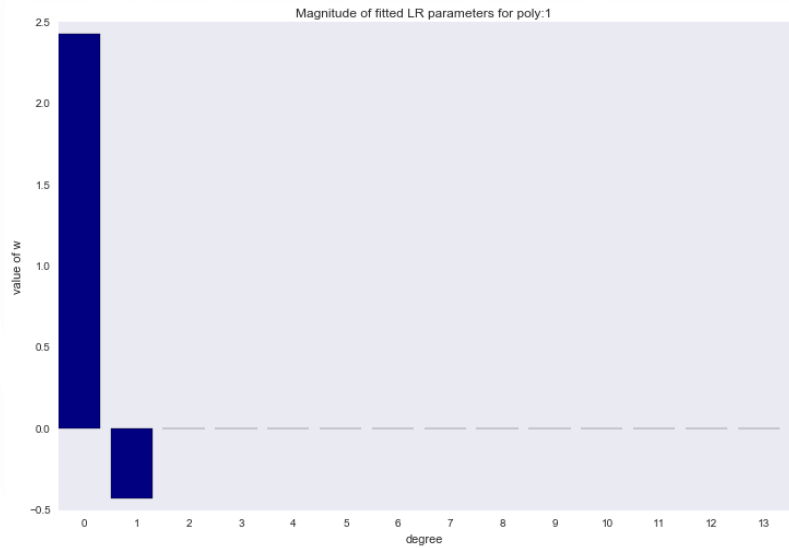
# ПРИЗНАКИ ПЕРЕОБУЧЕННОЙ МОДЕЛИ

- Большая разница в качестве на тренировочных и тестовых данных (модель подгоняется под тренировочные данные и не может найти истинную зависимость)
- Большие значения параметров (весов)  $w_j$  модели
- Неустойчивость дискриминантной (разделяющей) функции  $(w, x)$ .

# ПЕРЕОБУЧЕНИЕ: ПРИМЕР



# ПЕРЕОБУЧЕНИЕ: ПРИМЕР





# ОЦЕНИВАНИЕ КАЧЕСТВА МОДЕЛИ

- Отложенная выборка
- Кросс-валидация

# ОТЛОЖЕННАЯ ВЫБОРКА

Делим тренировочную выборку на две части:

- По первой части обучаем модель (train)
- По оставшимся данным – оцениваем качество (test)

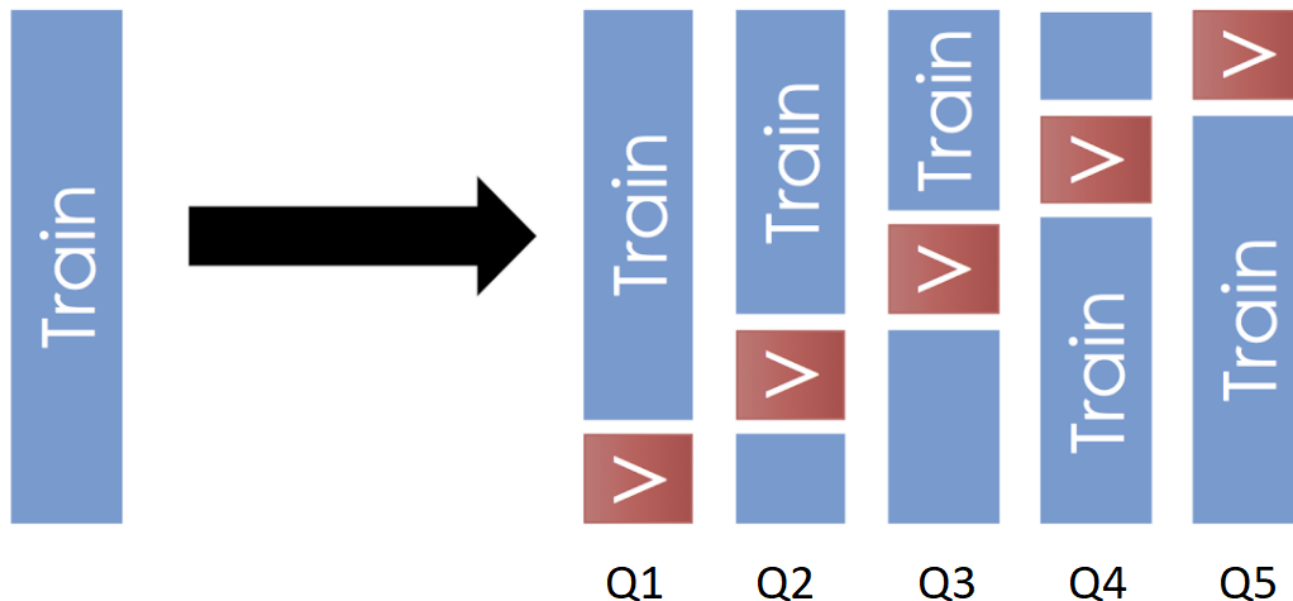


Недостаток:

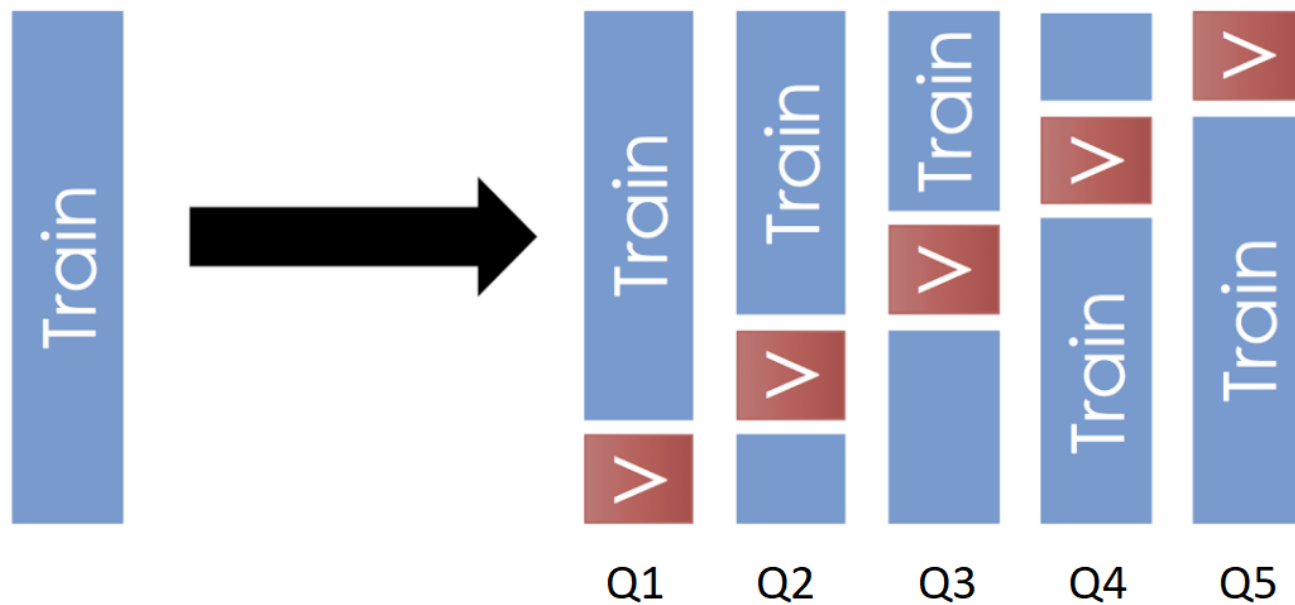
- Результат сильно зависит от разбиения на train и test

# КРОСС-ВАЛИДАЦИЯ

- Разбиваем объекты на тренировку (train) и валидацию (validation) несколько раз (при разбиении  $k$  раз получаем  $k$ -fold кросс-валидацию)
- Для каждого разбиения вычисляем качество на валидационной части
- Усредняем полученные результаты



# КРОСС-ВАЛИДАЦИЯ



$$CV = \frac{1}{k} \sum_{i=1}^k Q(a_i(x), X_i) = \frac{1}{k} \sum_{i=1}^k Q_i$$

# ВИДЫ КРОСС-ВАЛИДАЦИИ

- **k-fold cross-validation** – разбиваем данные на  $k$  блоков, каждый из которых по очереди становится контрольным (валидационным)
- **Complete cross-validation** – перебираем ВСЕ разбиения
- **Leave-one-out cross-validation** – каждый блок состоит из одного объекта (число блоков = числу объектов)

# ВЫБОР КОЛИЧЕСТВА БЛОКОВ В K-FOLD КРОСС-ВАЛИДАЦИИ



- Проблемы при маленьком  $k$ ?
- Проблемы при большом  $k$ ?

# ВЫБОР КОЛИЧЕСТВА БЛОКОВ В K-FOLD КРОСС-ВАЛИДАЦИИ



- Маленькое k – оценка может быть пессимистично занижена из-за маленького размера тренировочной части
- Большое k – оценка может иметь большую дисперсию из-за маленького размера валидационной части

# МЕТОД БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ: РЕГУЛЯРИЗАЦИЯ

**Утверждение.** Если в выборке есть линейно-зависимые признаки, то задача оптимизации  $Q(w) \rightarrow \min$  имеет бесконечное число решений.

- Большие значения параметров (весов) модели  $w$  – признак переобучения.



# МЕТОД БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ: РЕГУЛЯРИЗАЦИЯ

**Утверждение.** Если в выборке есть линейно-зависимые признаки, то задача оптимизации  $Q(w) \rightarrow \min$  имеет бесконечное число решений.

- Большие значения параметров (весов) модели  $w$  – признак переобучения.

Решение проблемы – **регуляризация**.

Будем минимизировать регуляризованный функционал ошибки:

$$Q_{alpha}(w) = Q(w) + \alpha \cdot R(w) \rightarrow \min_w ,$$

где  $R(w)$  - регуляризатор.

# РЕГУЛЯРИЗАЦИЯ

- Регуляризация штрафует за слишком большие веса.

Наиболее используемые регуляризаторы:

- $L_2$ -регуляризатор:  $R(w) = ||w||_2 = \sum_{i=1}^d w_i^2$
- $L_1$ -регуляризатор:  $R(w) = ||w||_1 = \sum_{i=1}^d |w_i|$

# РЕГУЛЯРИЗАЦИЯ

- Регуляризация штрафует за слишком большие веса.

Наиболее используемые регуляризаторы:

- $L_2$ -регуляризатор:  $R(w) = ||w||_2 = \sum_{i=1}^d w_i^2$
- $L_1$ -регуляризатор:  $R(w) = ||w||_1 = \sum_{i=1}^d |w_i|$

Пример регуляризованного функционала:

$$Q(a(w), X) = \frac{1}{l} \sum_{i=1}^l ((w, x_i) - y_i)^2 + \alpha \sum_{i=1}^d w_i^2,$$

где  $\alpha$  – коэффициент регуляризации.

# ПОЛЕЗНОЕ СВОЙСТВО L1-РЕГУЛЯРИЗАЦИИ

*Все ли признаки в задаче нужны?*

- Некоторые признаки могут не иметь отношения к задаче, т.е. они не нужны.
- Если есть ограничения на скорость получения предсказаний, то чем меньше признаков, тем быстрее
- Если признаков больше, чем объектов, то решение задачи будет неоднозначным.

*Поэтому в таких случаях надо делать отбор признаков, то есть убирать некоторые признаки.*

# $L_1$ -РЕГУЛЯРИЗАЦИЯ

**Утверждение.** В результате обучения модели с  $L_1$ -регуляризатором происходит зануление некоторых весов, т.е. отбор признаков.

Можно показать, что задачи

$$(1) \quad Q(w) + \alpha \|w\|_1 \rightarrow \min_w$$

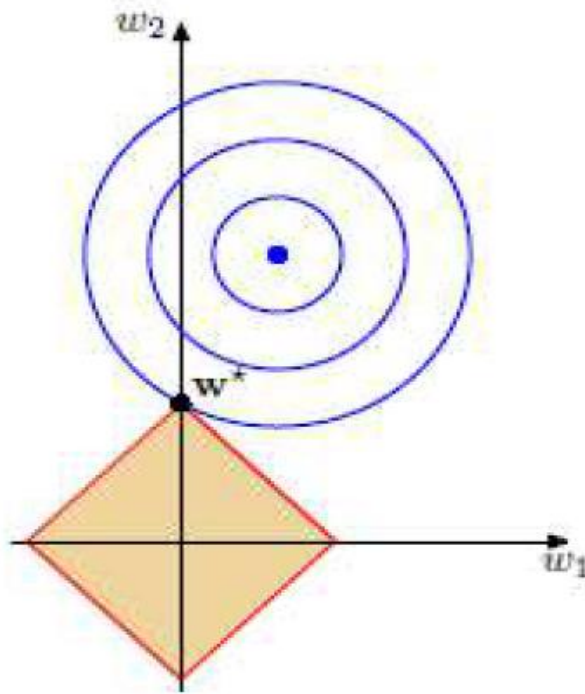
и

$$(2) \quad \begin{cases} Q(w) \rightarrow \min_w \\ \|w\|_1 \leq C \end{cases}$$

эквивалентны.

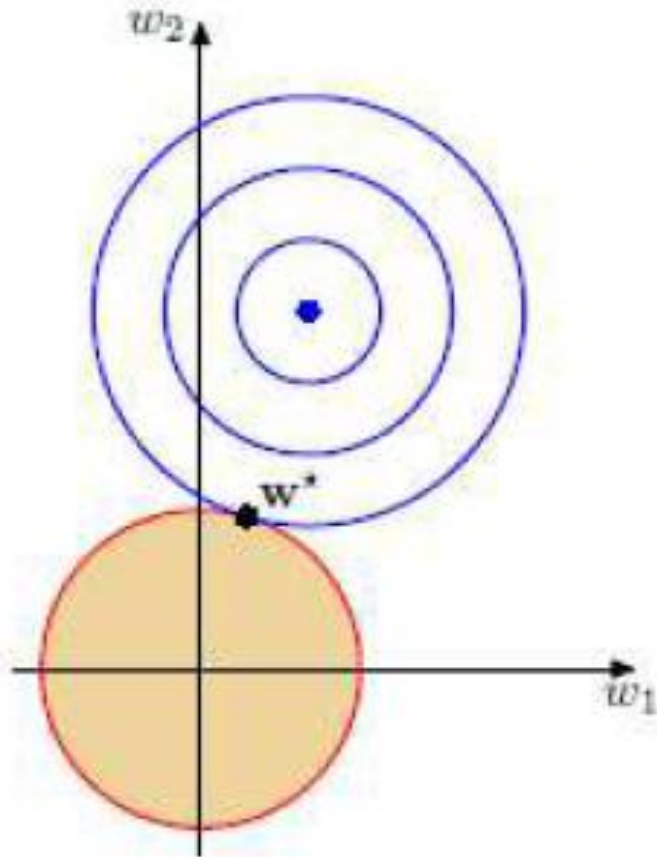
# ОТБОР ПРИЗНАКОВ ПО L1-РЕГУЛЯРИЗАЦИИ

Нарисуем линии уровня  $Q(w)$  и область  $\|w\|_1 \leq C$ :



Если признак незначимый, то соответствующий вес близок к 0. Отсюда получим, что в большинстве случаев решение нашей задачи попадает в вершину ромба, т.е. обнуляет незначимый признак.

# L2-РЕГУЛЯРИЗАЦИЯ НЕ ОБНУЛЯЕТ ПРИЗНАКИ



# РАЗРЕЖЕННЫЕ МОДЕЛИ

Модели, в которых часть весов равна 0, называются ***разреженными моделями***.

- L1-регуляризация зануляет часть весов, то есть делает модель разреженной.