

Машинное обучение

Лекция 2
Линейная регрессия

Михаил Гущин
mhushchyn@hse.ru

НИУ ВШЭ, 2023

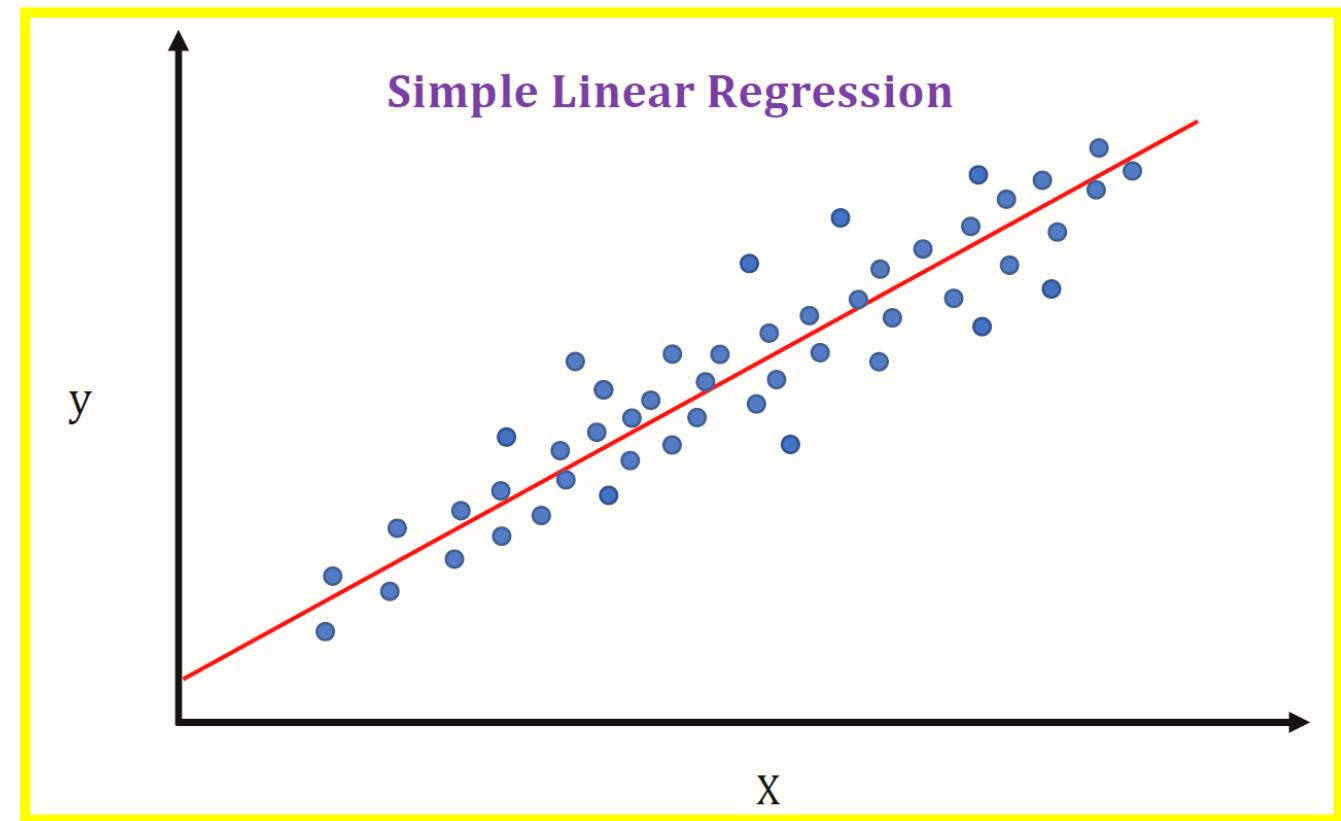


НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

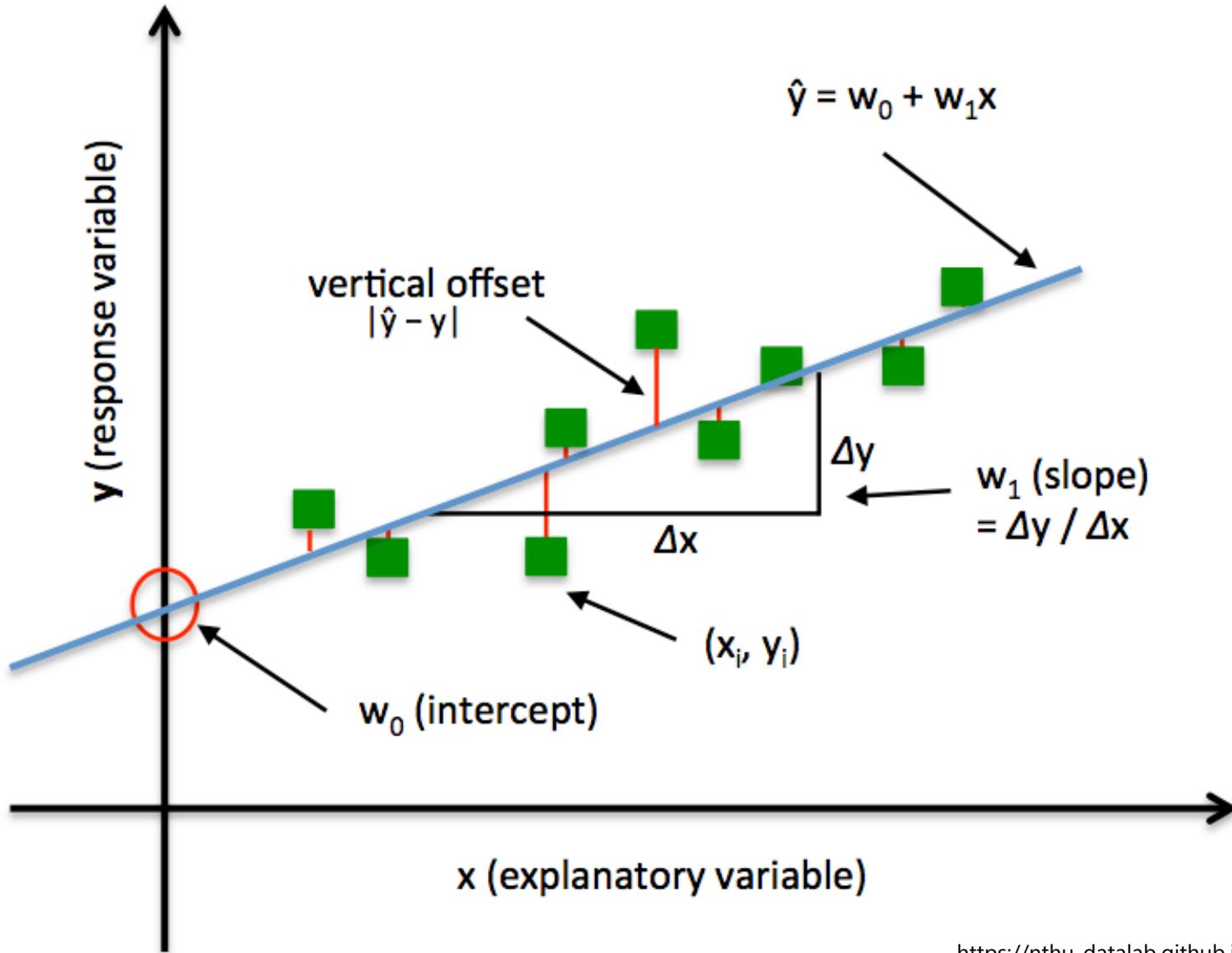
Линейная регрессия. Аналитическое решение.

Задача регрессии

- ▶ Есть объекты (X)
- ▶ Нужно предсказать некоторую величину (y)
- ▶ Функция, которая описывает зависимость y от X - **модель регрессии**



Линейная регрессия



<https://nchu-datalab.github.io>

Векторная форма

- ▶ Пусть дан набор из n точек: $\{x_i, y_i\}_{i=1}^n$, где
 - $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ - вектор из d признаков объекта;
 - y_i - скалярная величина, которую хотим предсказать для объекта.

- ▶ Модель линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^d w_j x_{ij}$$

- w_j - веса модели;
 - \hat{y}_i - прогноз для объекта;
- ▶ Квадрат ошибки прогноза модели для объекта: $(\hat{y}_i - y_i)^2$

Матричная форма

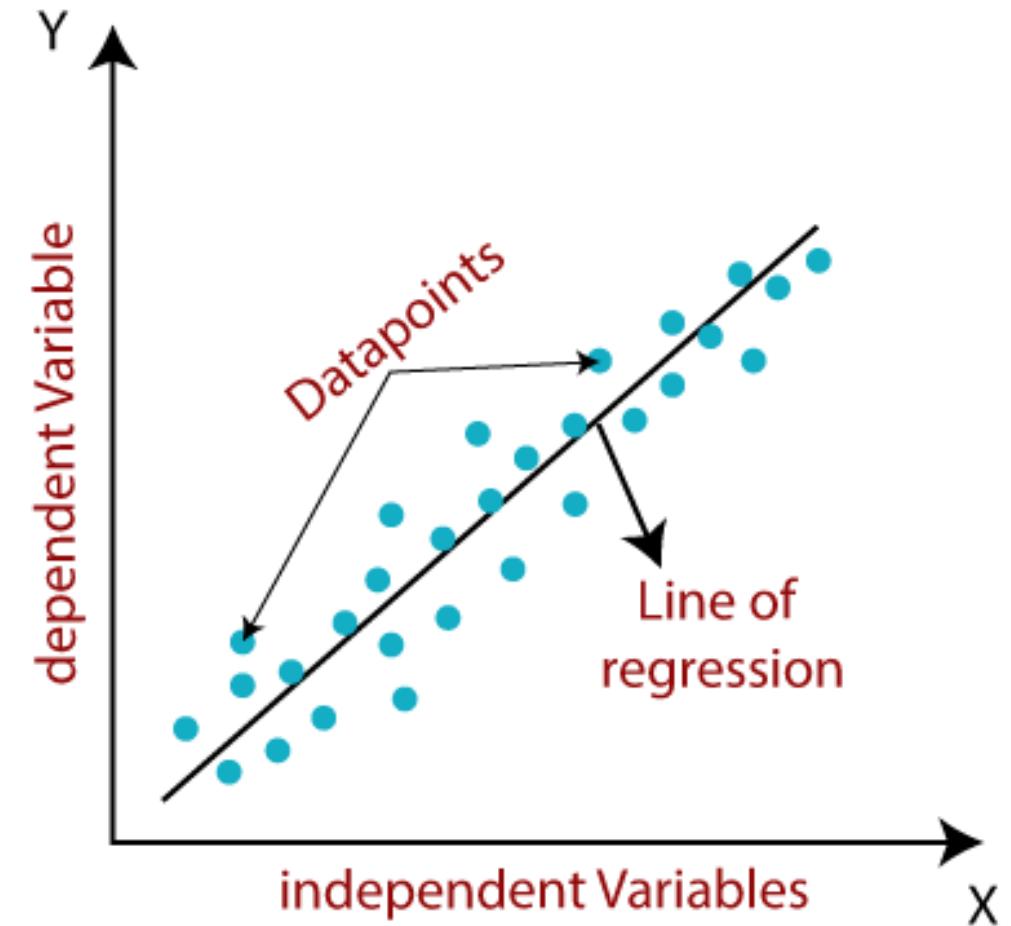
- Модель линейной регрессии:

$$\hat{y} = Xw$$

- $X = \begin{pmatrix} \mathbf{1} & x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & & \vdots \\ \mathbf{1} & x_{n1} & \cdots & x_{nd} \end{pmatrix}$ - матрица признаков объектов $(n, d + 1)$;
 - $w = (w_0, w_1, \dots, w_d)^T$ - вектор $(d + 1)$ весов модели;
 - $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)^T$ - вектор прогнозов модели для (n) объектов;
-
- Вектор квадратов ошибок прогнозов модели: $(\hat{y} - y)^2$

Задача

- ▶ Хотим, чтобы средняя квадратичная ошибка прогнозов $(\hat{y} - y)^2$ была минимальной
- ▶ **Как найти** оптимальные веса w модели?



Решение

- ▶ **Функция потерь (Loss function)** (скалярная и векторная формы):

$$L = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 = \frac{1}{n} (\hat{\mathbf{y}} - \mathbf{y})^T (\hat{\mathbf{y}} - \mathbf{y})$$

- ▶ Значение L – среднеквадратичная ошибка (**Mean Squared Error (MSE)**)
- ▶ Мы хотим минимизировать L :

$$L \rightarrow \min_w$$

Аналитическое решение

$$L = (\hat{y} - y)^T (\hat{y} - y) = (\mathbf{X}\mathbf{w} - y)^T (\mathbf{X}\mathbf{w} - y)$$

Чтобы найти минимум L , надо:

$$\frac{\partial L}{\partial \mathbf{w}} = 0$$

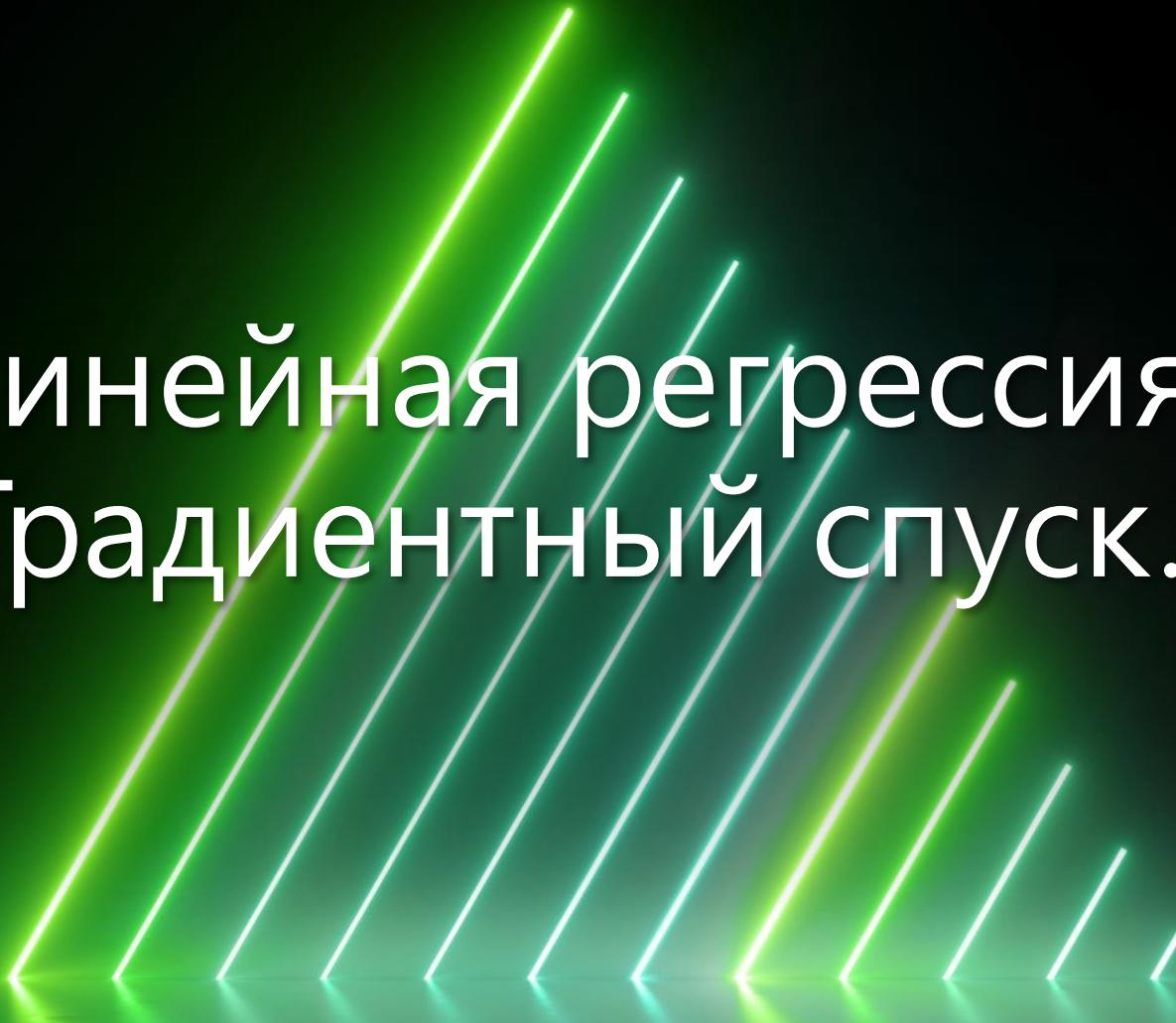
Тогда

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - y) = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^Ty = 0$$

Получаем оптимальные веса w линейной регрессии:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Линейная регрессия.
Градиентный спуск.

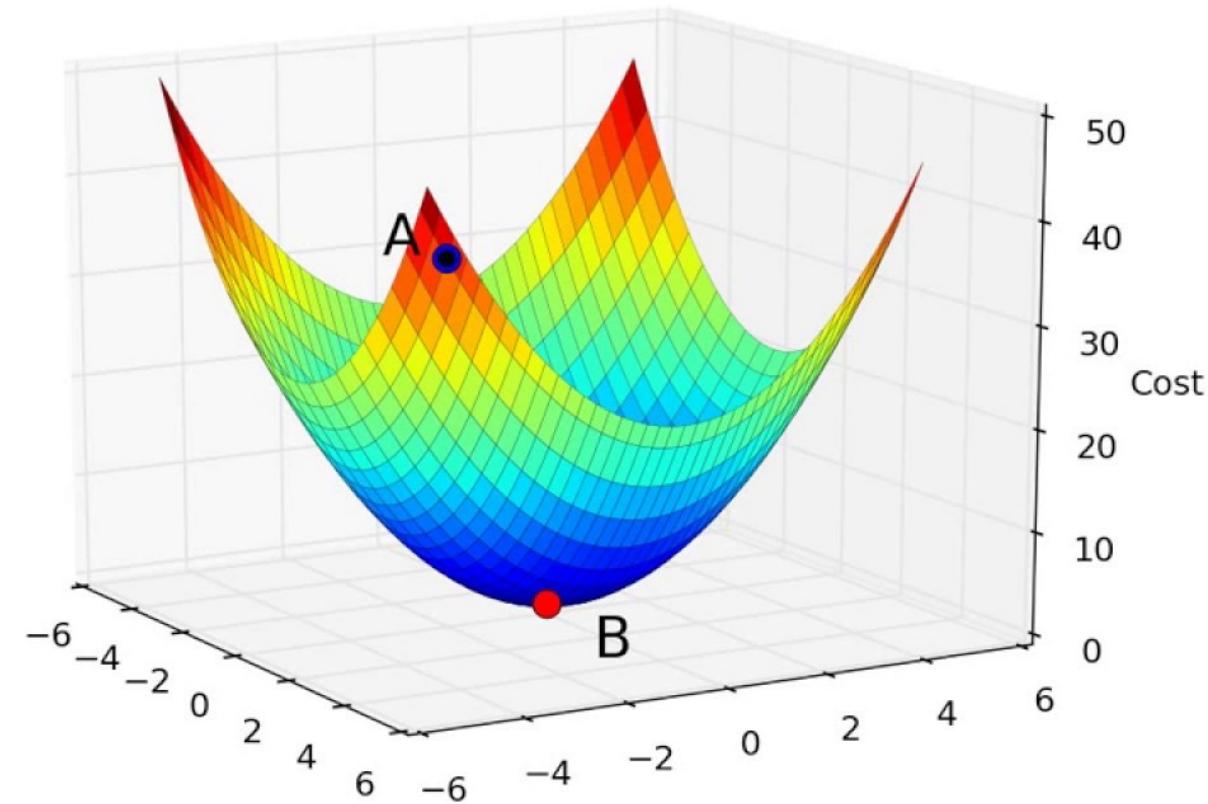


Задача

- ▶ Есть функция $L(w)$
- ▶ Хотим найти ее минимум:

$$L \rightarrow \min_w$$

- ▶ Мы умеем считать ее производную $\frac{\partial L}{\partial w}$
- ▶ Но не умеем (или не хотим) решать уравнение $\frac{\partial L}{\partial w} = 0$



Градиент функции

- ▶ Градиент функции (∇L) – вектор первых частных производных функции:

$$\nabla L(w_0, w_1, \dots, w_d) = \left(\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_d} \right)$$

- ▶ В векторной форме мы будем писать:

$$\nabla L(w) = \frac{\partial L(w)}{\partial w}$$

Градиентный спуск

- ▶ Есть функция $L(w)$, минимум которой хотим найти
- ▶ Пусть $w^{(0)}$ - начальный вектор параметров. Например, $w^{(0)} = 0$
- ▶ Тогда **градиентный спуск** состоит в повторении:

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)})$$

- η – длина шага градиентного спуска (**learning rate**) (мы сами его задаем)
- k – номер итерации
- $\nabla L(w^{(k)})$ – градиент функции потерь на итерации k

Пример

- ▶ Модель линейной регрессии:

$$\hat{y} = Xw$$

- ▶ Функция потерь MSE:

$$L = \frac{1}{n}(\hat{y} - y)^T(\hat{y} - y) = \frac{1}{n}(Xw - y)^T(Xw - y)$$

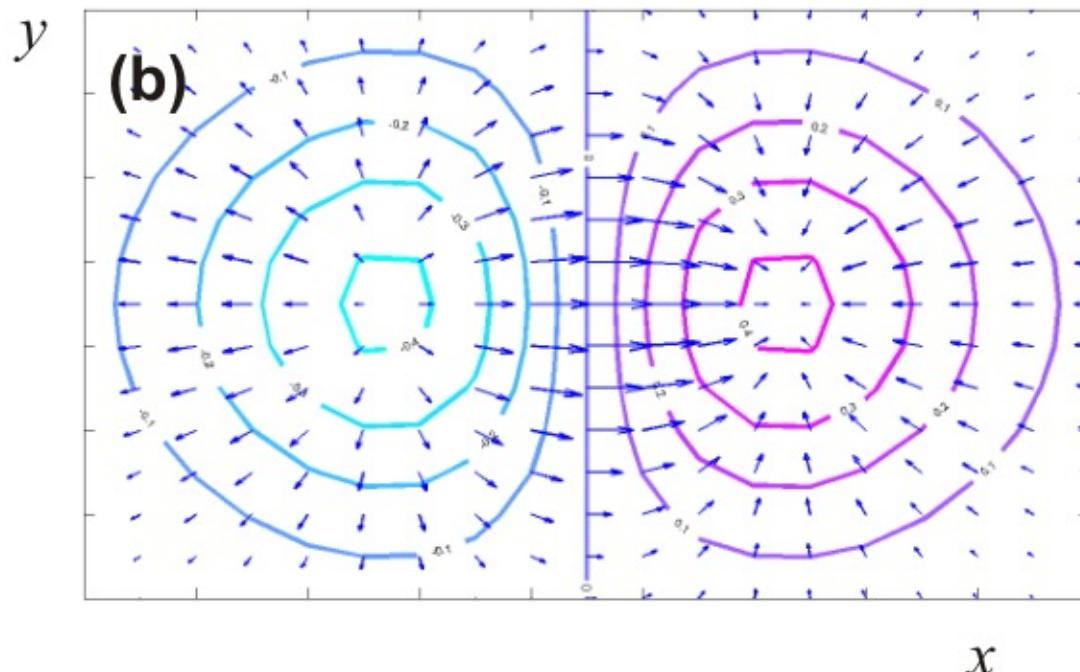
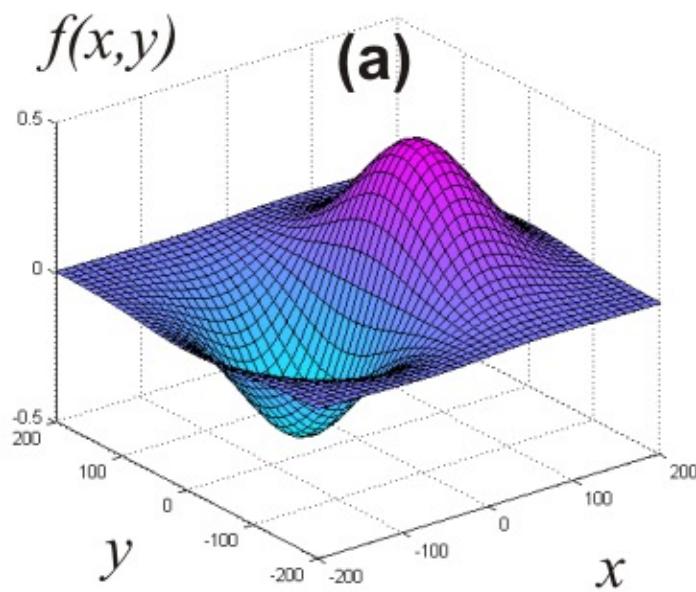
- ▶ Градиент:

$$\nabla L(w) = \frac{2}{n}X^T(Xw - y)$$

- ▶ Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)}) = w^{(k)} - \eta \frac{2}{n}X^T(Xw^{(k)} - y)$$

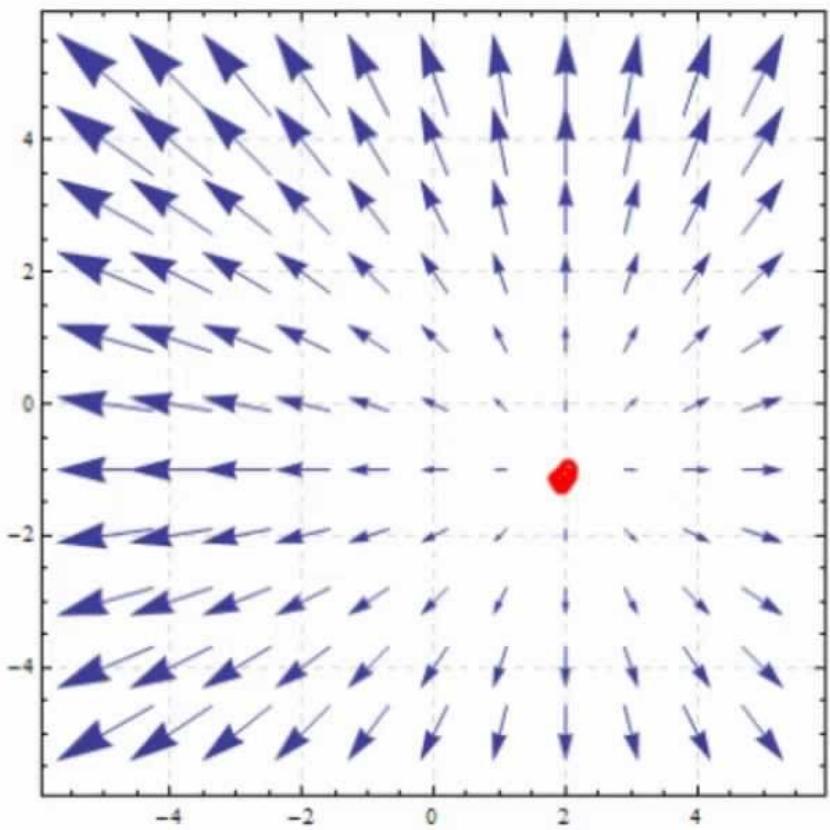
Свойства градиента



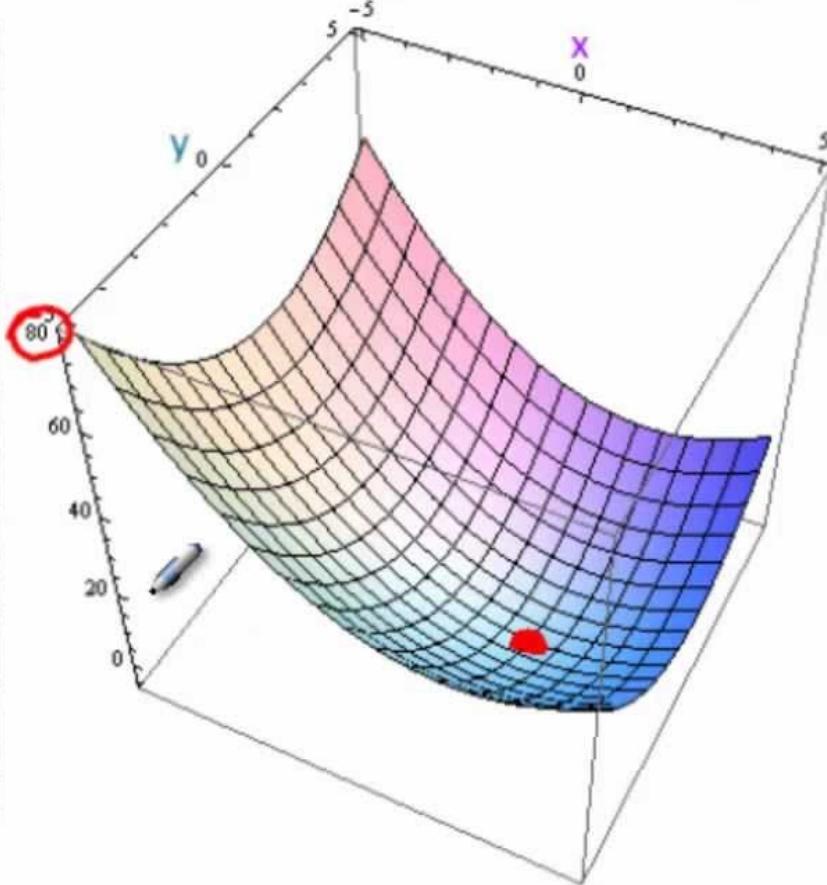
Карта градиентов и линии уровня
функции

Свойства градиента

The gradient field $\langle 2x-4, 2y+2 \rangle$

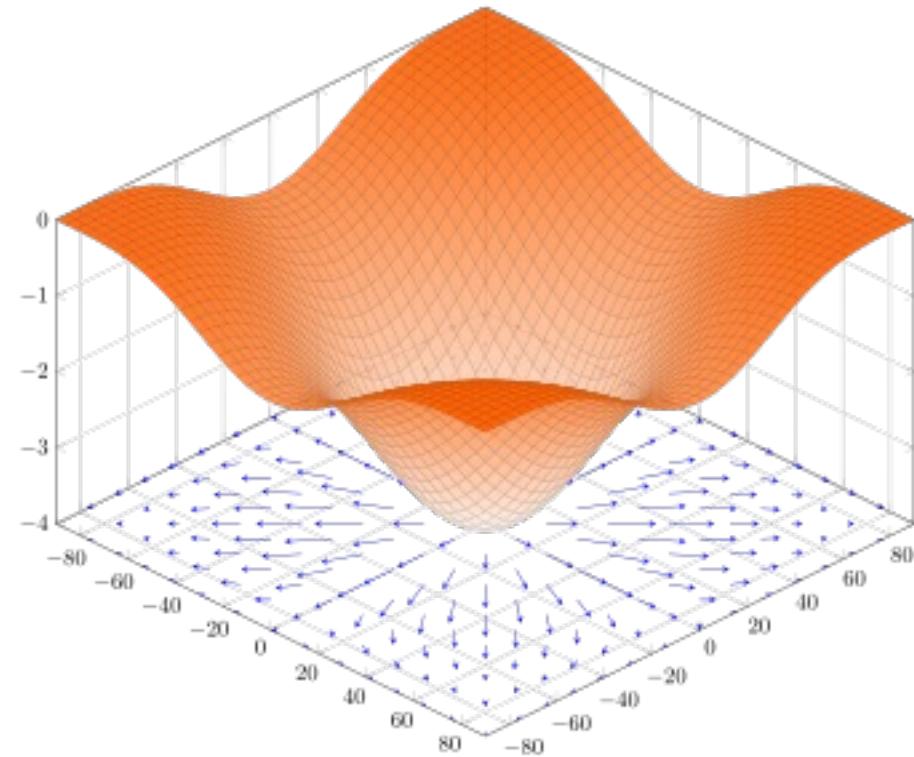


of the function $f = x^2 - 4x + y^2 + 2y$.



Свойства градиента

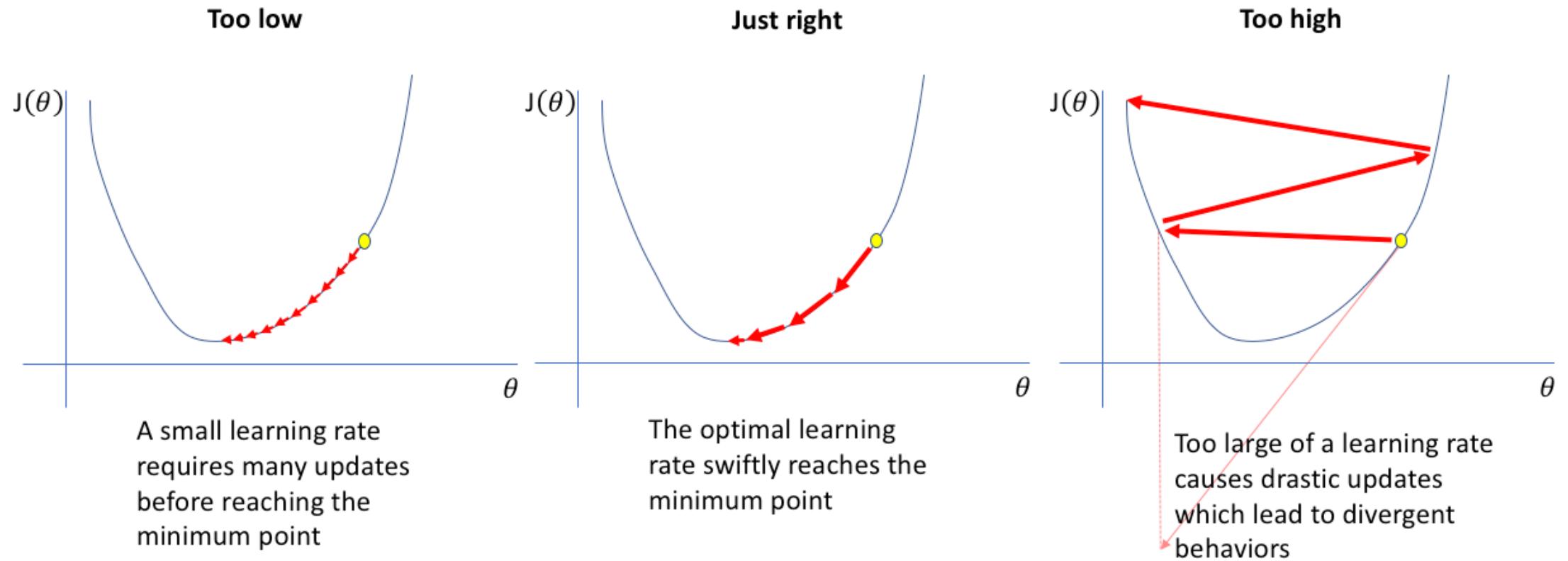
- ▶ Градиент функции в некоторой точке ортогонален линии уровня, проходящей через эту точку
- ▶ Градиент функции указывает направление наискорейшего возрастания функции в данной точке
- ▶ Направление антиградиента указывает направление наискорейшего убывания функции в данной точке



Градиентный спуск

- ▶ Как выбрать длину градиентного спуска?
- ▶ Сколько итераций делать?

Выбор шага



Выбор шага

- ▶ Константа: $\eta = \text{const}$
- ▶ Уменьшение с каждой итерацией k : $\eta_k = \frac{1}{k}$
- ▶ Другие варианты: $\eta_k = \lambda \left(\frac{s_0}{s_0 + k} \right)^p$
 - λ, s_0, p – некоторые значения
 - как правило $s_0 = 1, p = 0.5$

Критерии остановки

- ▶ Близость градиента к нулю: $\nabla L \approx 0$
- ▶ Малое изменение вектора весов:
 $|w^{(k+1)} - w^{(k)}| \approx 0$



Стохастический градиентный спуск

- ▶ Модель линейной регрессии:

$$\hat{y} = Xw$$

- ▶ Функция потерь MSE:

$$L(w) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

- ▶ Мы хотим минимизировать L :

$$L \rightarrow \min_w$$

- ▶ Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)})$$

Стохастический градиентный спуск

- ▶ Полный градиентный спуск:

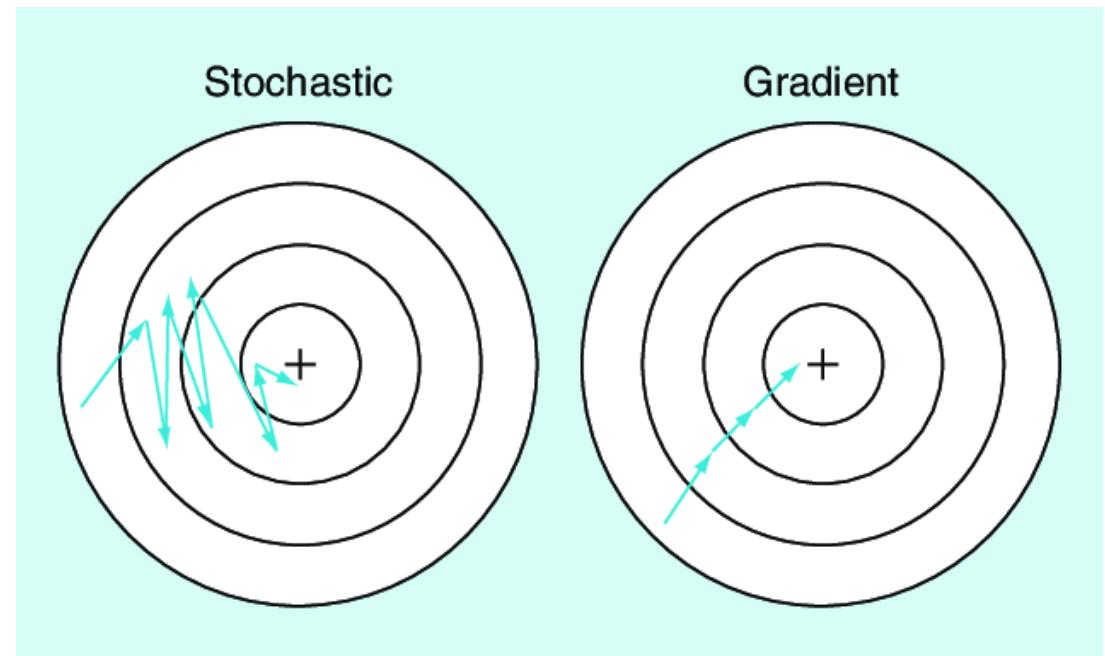
$$L(w) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)})$$

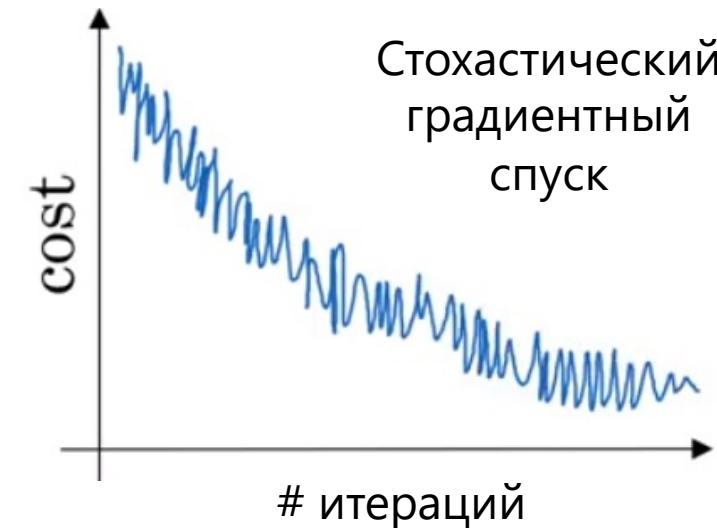
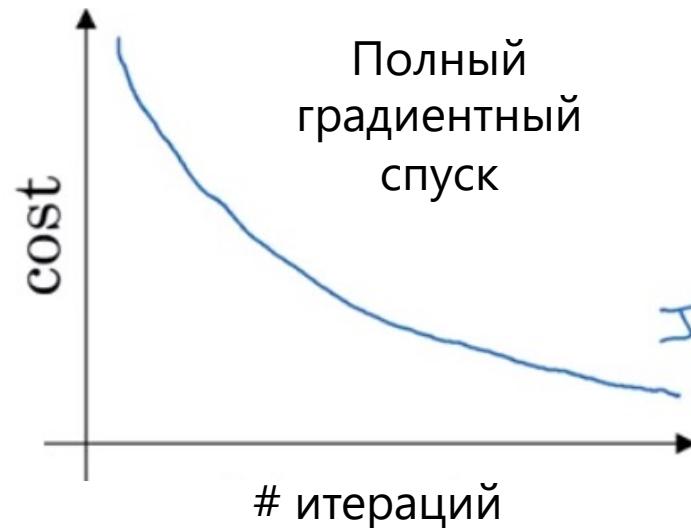
- ▶ **Стохастический** градиентный спуск:

$$L_i(w) = (\hat{y}_i - y_i)^2$$

$$w^{(k+1)} = w^{(k)} - \eta \nabla L_i(w^{(k)})$$



Стохастический градиентный спуск



- ▶ Стохастический ГС требует меньше вычислительных операций
- ▶ В полном ГС обучение стабильнее
- ▶ Полный ГС требует меньше итераций, но больше вычислительных операций

Нормализация данных

- ▶ Модель линейной регрессии:

$$\hat{y}_i = w_0 + w_1 x_{i1} + w_2 x_{i2}$$

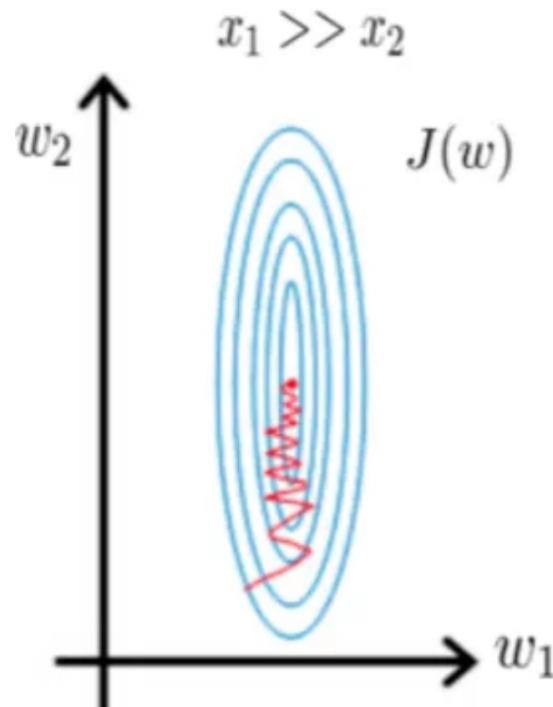
- ▶ Функция потерь MSE:

$$L(w) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \rightarrow \min_w$$

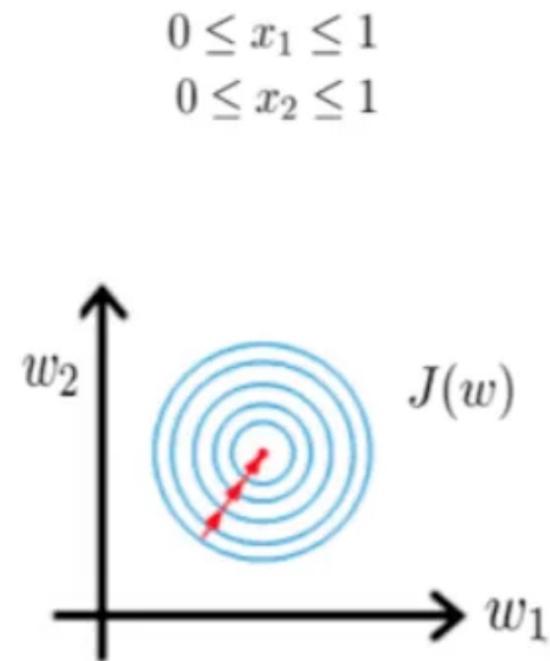
- ▶ Пусть признаки разные по масштабу:
 - например $|x_1| \approx 1000, |x_2| \approx 1$
- ▶ Тогда малые изменения dw_2 приводят к малым изменениям $L(w)$
- ▶ Малые изменения dw_1 приводят к **большим** изменениям $L(w)$

Нормализация данных

Gradient descent
without scaling



Gradient descent
after scaling variables



- ▶ Нормализация данных стабилизирует градиентный спуск
- ▶ Обучение происходит быстрее

The background of the image is a dense, abstract pattern of numerous thin, curved lines in various colors, primarily shades of red, orange, and pink, set against a dark, almost black, background. These lines create a sense of depth and motion, resembling a complex network or a microscopic view of organic structures.

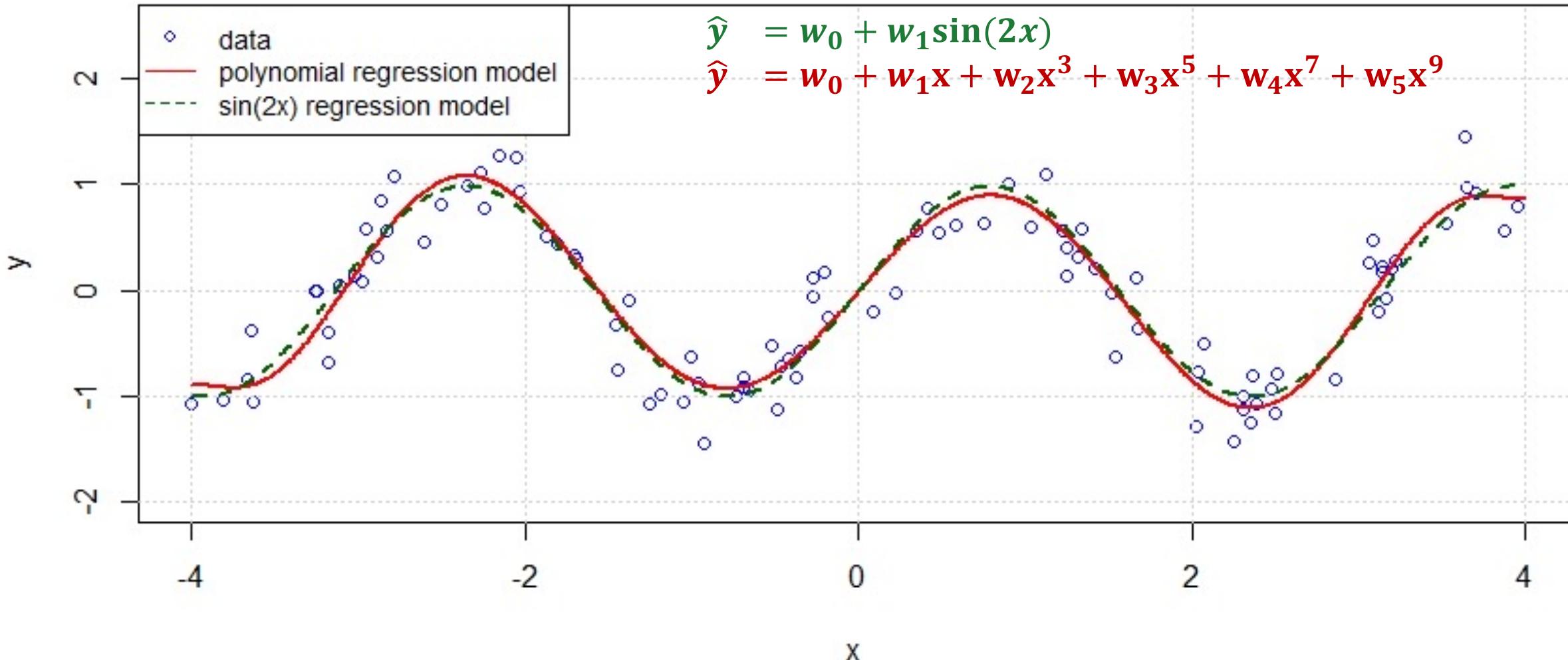
Нелинейные зависимости

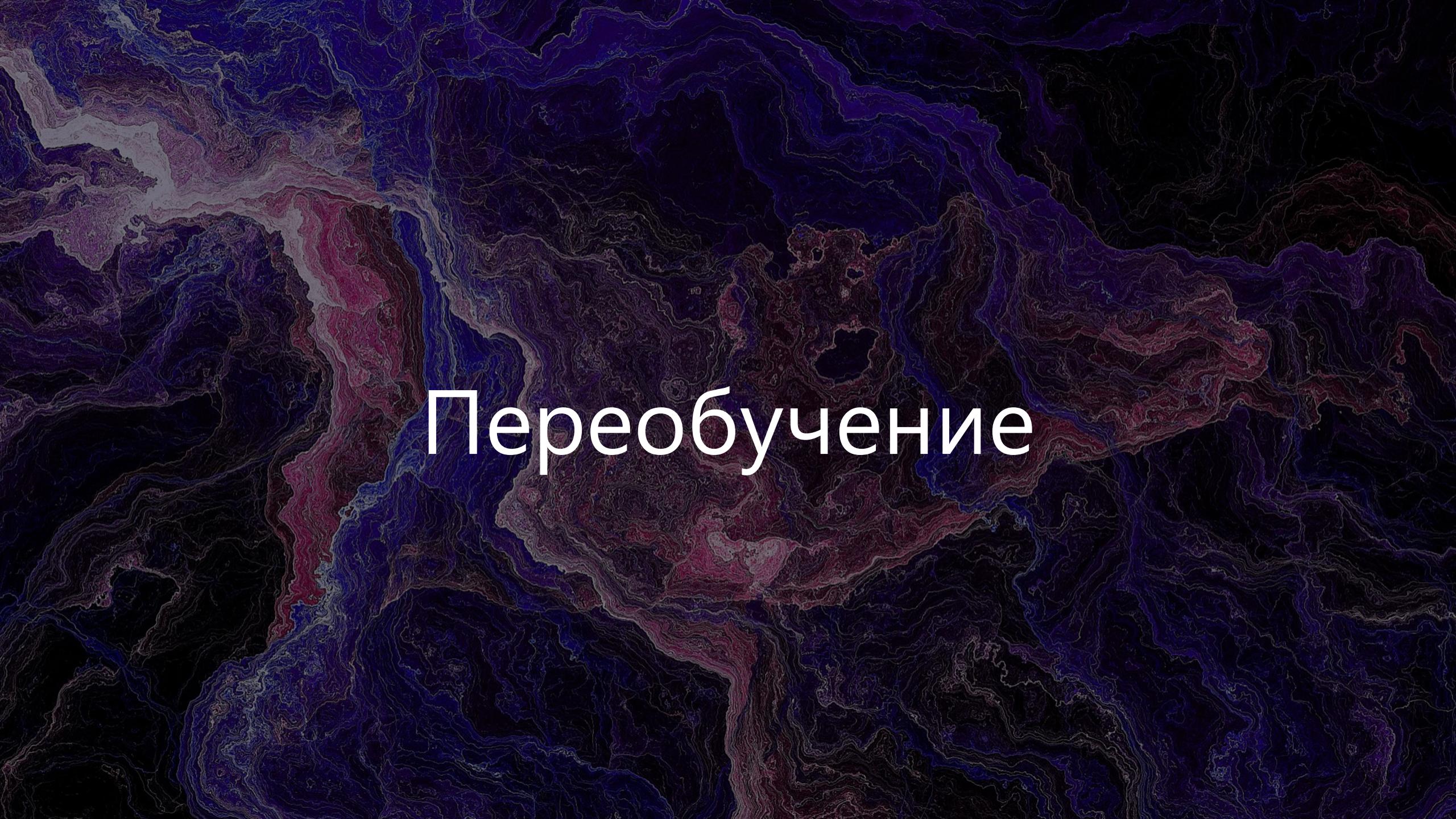
Нелинейные зависимости

- ▶ Пусть дан вектор признаков x_i
- ▶ Выберем любые нелинейные функции от x : $\phi_0(x), \phi_1(x), \dots, \phi_M(x)$
 - Например, $x^2, \sin(x), e^{-x}$
- ▶ Тогда модель линейной регрессии:

$$\hat{y}_i = \sum_{j=0}^{\textcolor{blue}{k}} w_j \phi_j(x_i)$$

Пример





Переобучение

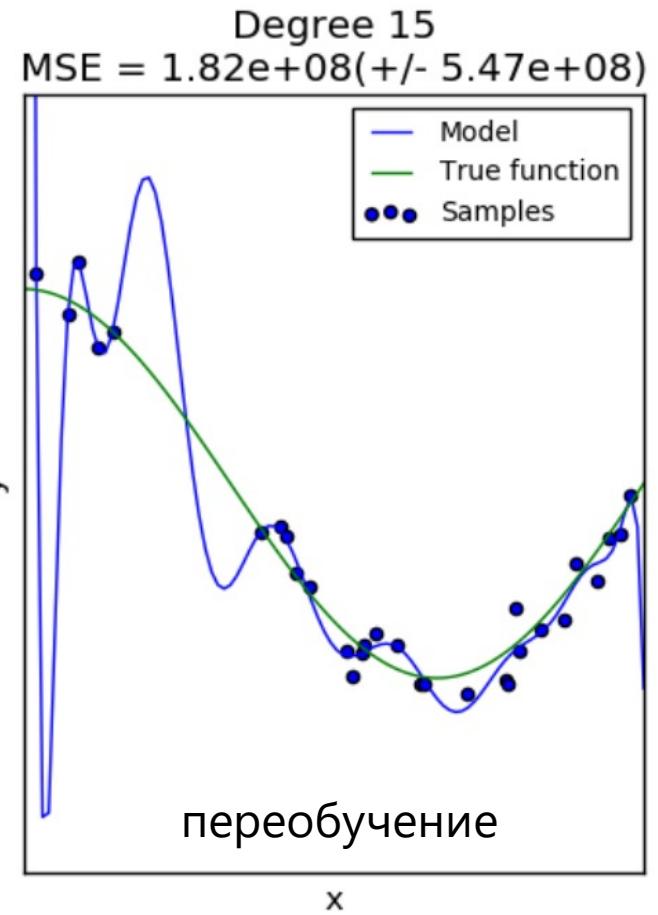
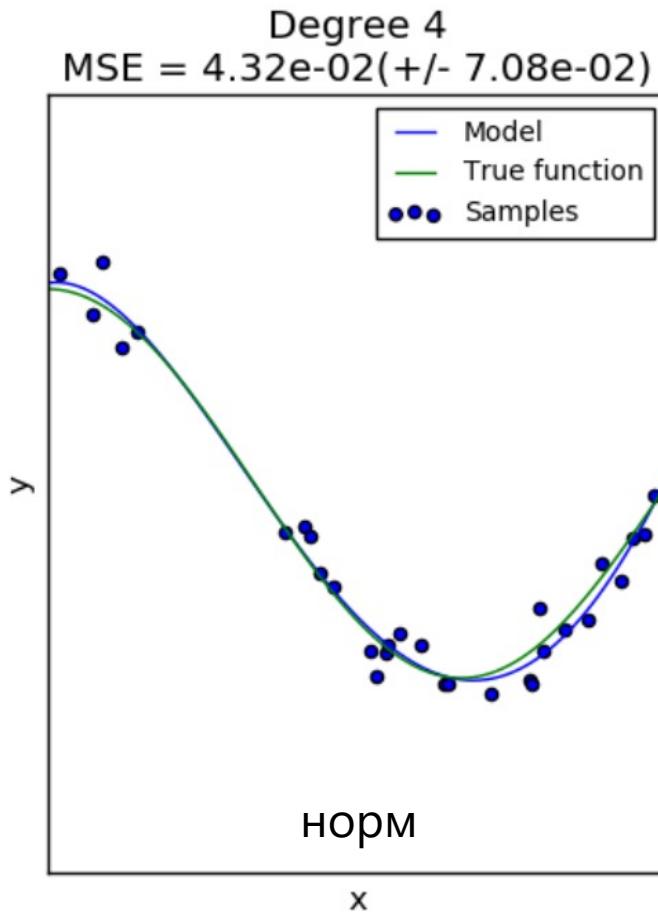
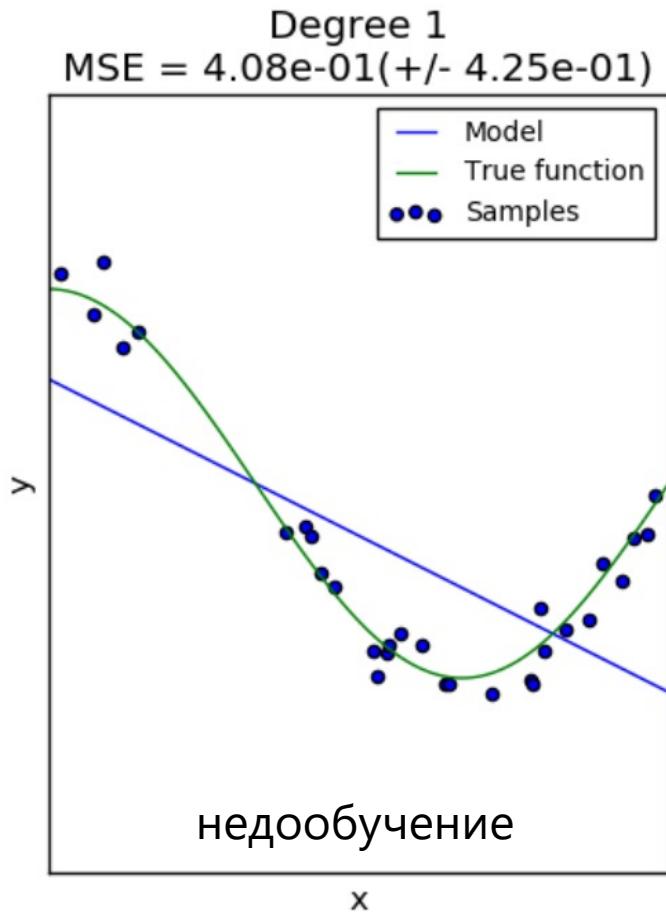
Задача

- ▶ Пусть дан набор из n точек: $\{x_i, y_i\}_{i=1}^n$, где $x_i \in \mathbb{R}^1$
- ▶ Для каждого x_i создадим дополнительные признаки:
 - $x_i, x_i^2, x_i^3, \dots, x_i^k$
- ▶ Рассмотрим модель полиномиальной линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^{\mathbf{k}} w_j x_i^j$$

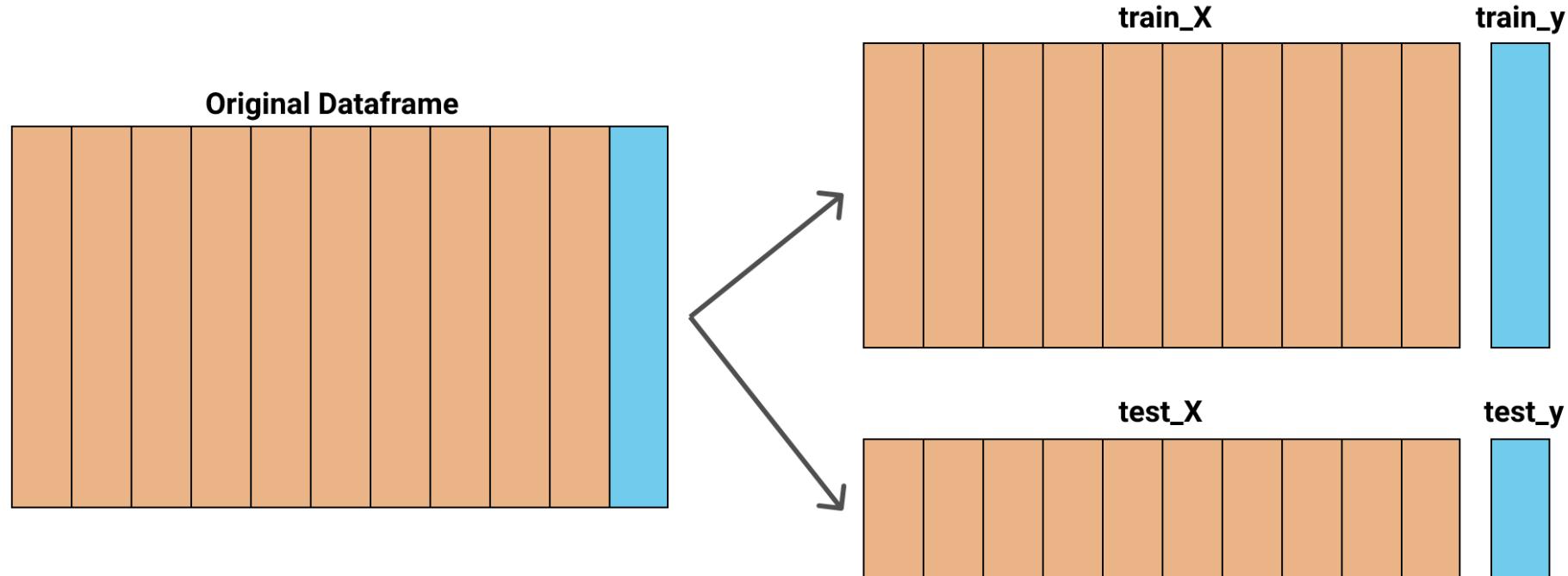
- ▶ Максимальную степень полинома \mathbf{k} будем менять от 1 до 15

Решение

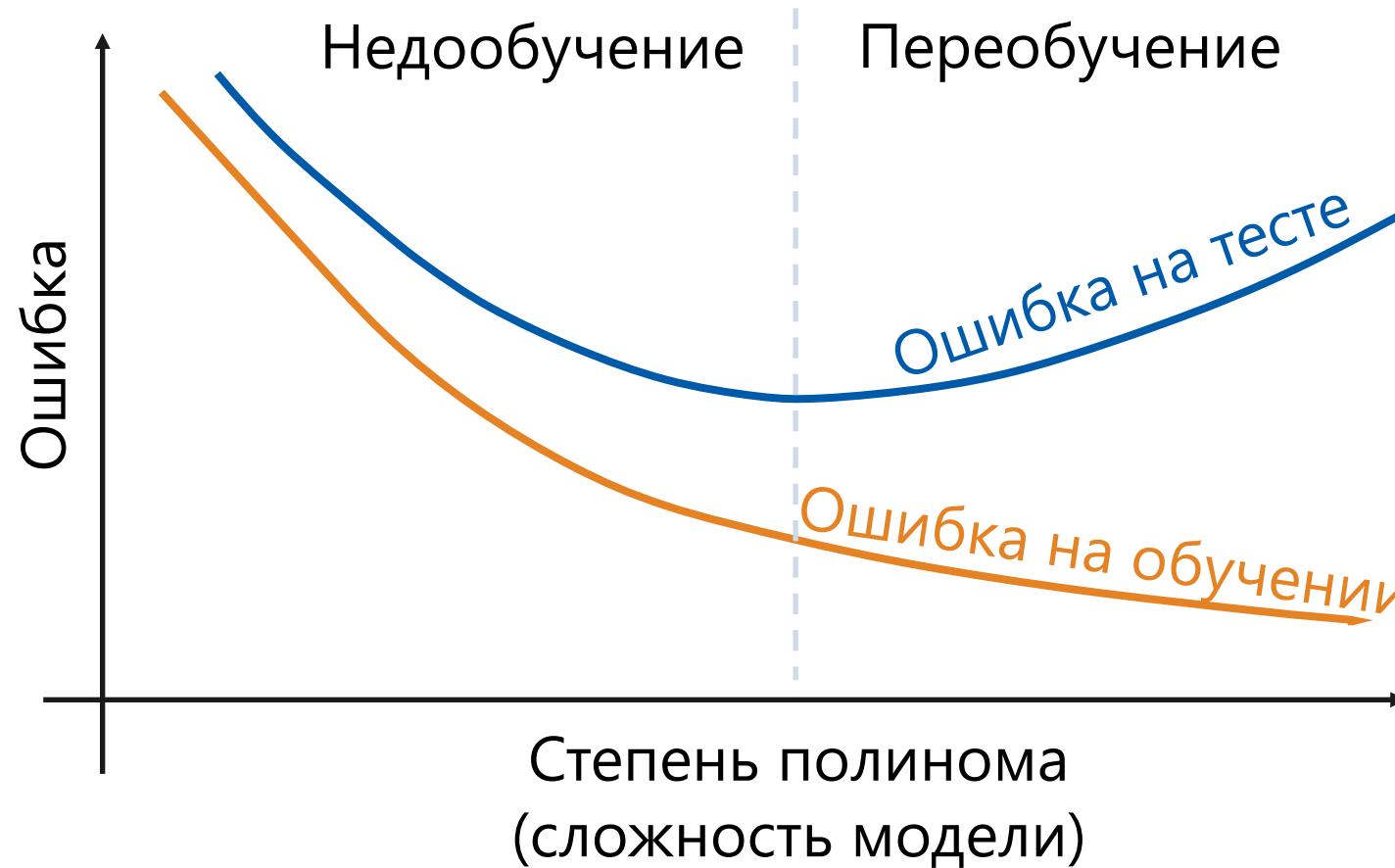


Обучение и тест

- ▶ **Обучающая выборка (train):** для обучения модели
- ▶ **Тестовая (отложенная) выборка (test):** для измерения качества модели



Переобучение



Регуляризация

Проблема переобучения

- ▶ Модель линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^d w_j x_{ij}$$

- ▶ Ошибка прогноза модели для объекта: $|\hat{y}_i - y_i|$
- ▶ Пусть значение некоторых весов очень большие по модулю, например $|w_k| > 10^3$
- ▶ Тогда малые изменения dx_{ik} приводят к очень большим изменениям $|d\hat{y}_i| = |w_k dx_{ik}|$

Регуляризация

- ▶ Давайте добавим к функции потерь $L(w)$ **штраф** $R(w)$ на **величину весов** модели:

$$L_\alpha(w) = L(w) + \alpha R(w)$$

- α – коэффициент регуляризации (подбираем сами)
- ▶ Регуляризация не позволяет весам модели принимать слишком большие значения

Виды регуляризации

- ▶ L_1 регуляризация (Lasso):

$$R_1(w) = \sum_{j=1}^d |w_j|$$

- ▶ L_2 регуляризация (Ridge):

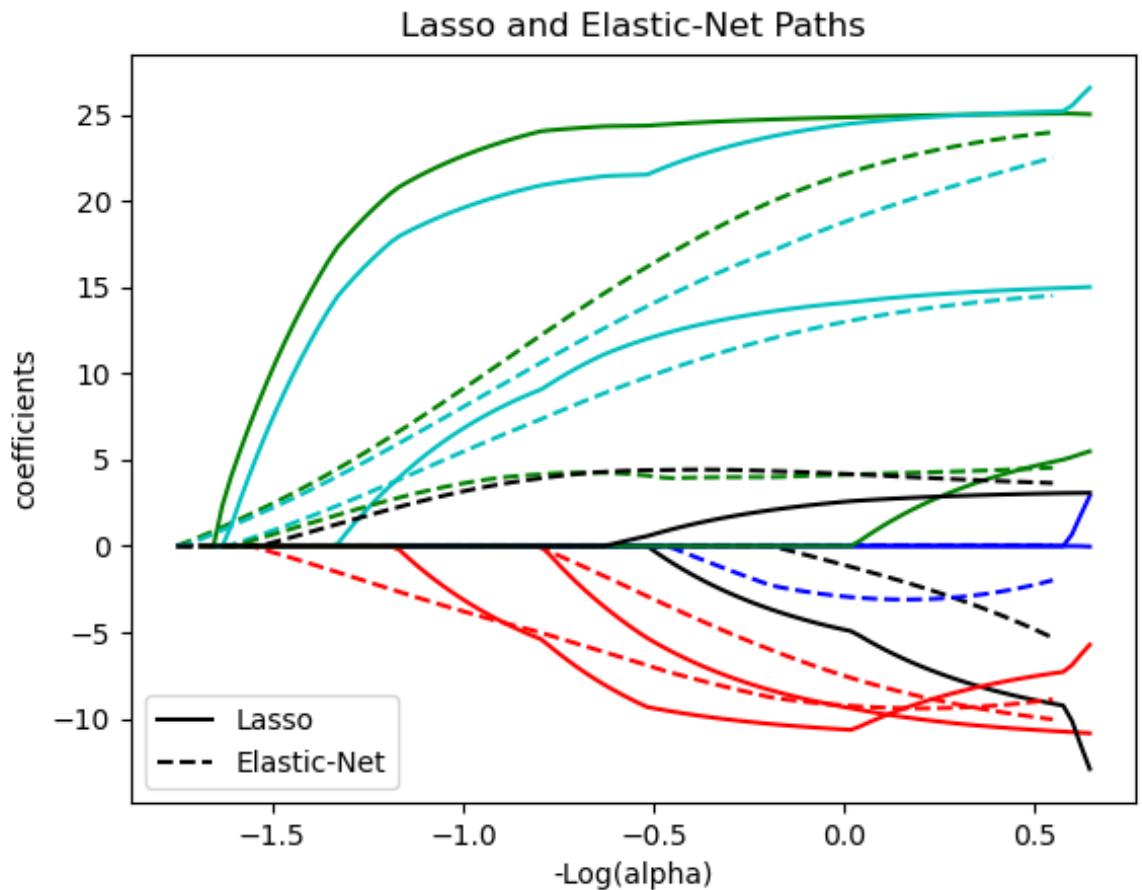
$$R_2(w) = \sum_{j=1}^d w_j^2$$

- ▶ $L_1 + L_2$ регуляризация (Elastic Net):

$$L_\alpha(w) = L(w) + \alpha_1 R_1(w) + \alpha_2 R_2(w)$$

Свойства регуляризации

- ▶ L_2 регуляризация стремится уменьшить веса модели
- ▶ L_1 позволяет проводить **отбор признаков**
- ▶ **L_1 обнуляет веса** для наименее информативных признаков



<https://scikit-learn.org>

Гиперпараметры и параметры

- ▶ Рассмотрим пример функции потерь с регуляризацией:

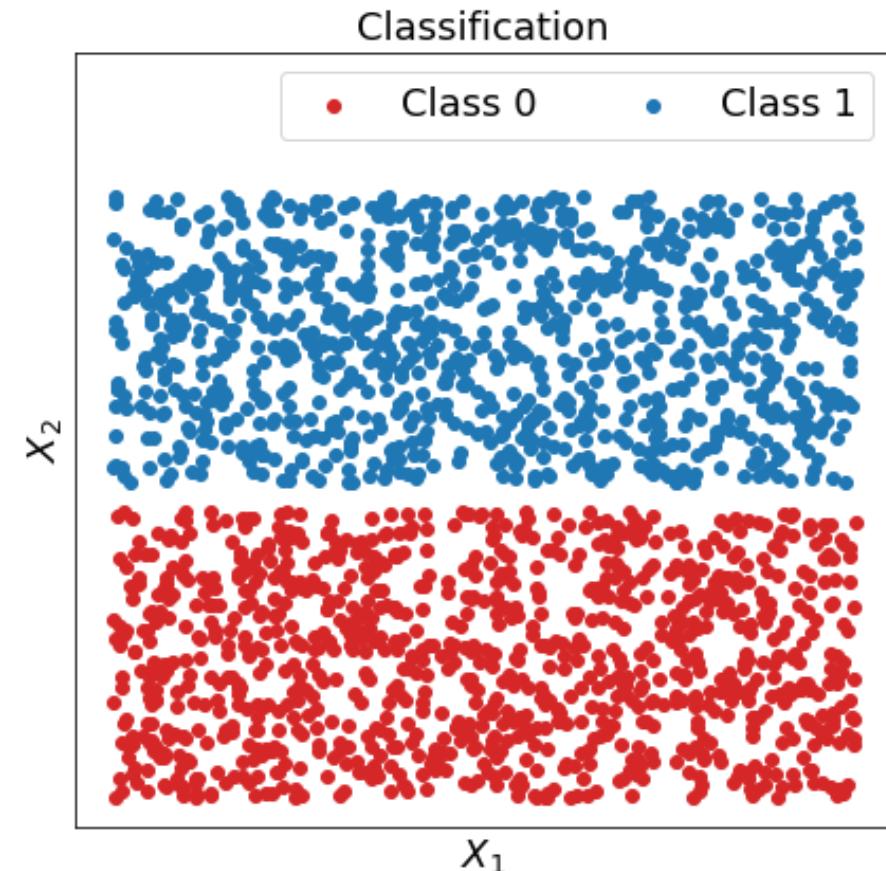
$$L_\alpha(w) = L(w) + \alpha R(w)$$

- ▶ Здесь w – веса нашей модели. Их будем называть **параметрами** модели. Они **определяются в процессе обучения**.
- ▶ α – коэффициент регуляризации. Его **значение задаем мы сами**. Такие параметры будем называть **гиперпараметрами**.

Важность признаков (Feature importance)

Интуиция

- ▶ Не все признаки одинаково полезны для решения задачи
- ▶ Некоторые из них более информативны, чем другие
- ▶ Например, X_1 неинформативна для классификации
- ▶ **Цель** – определить **важность каждого признака**



Линейные модели

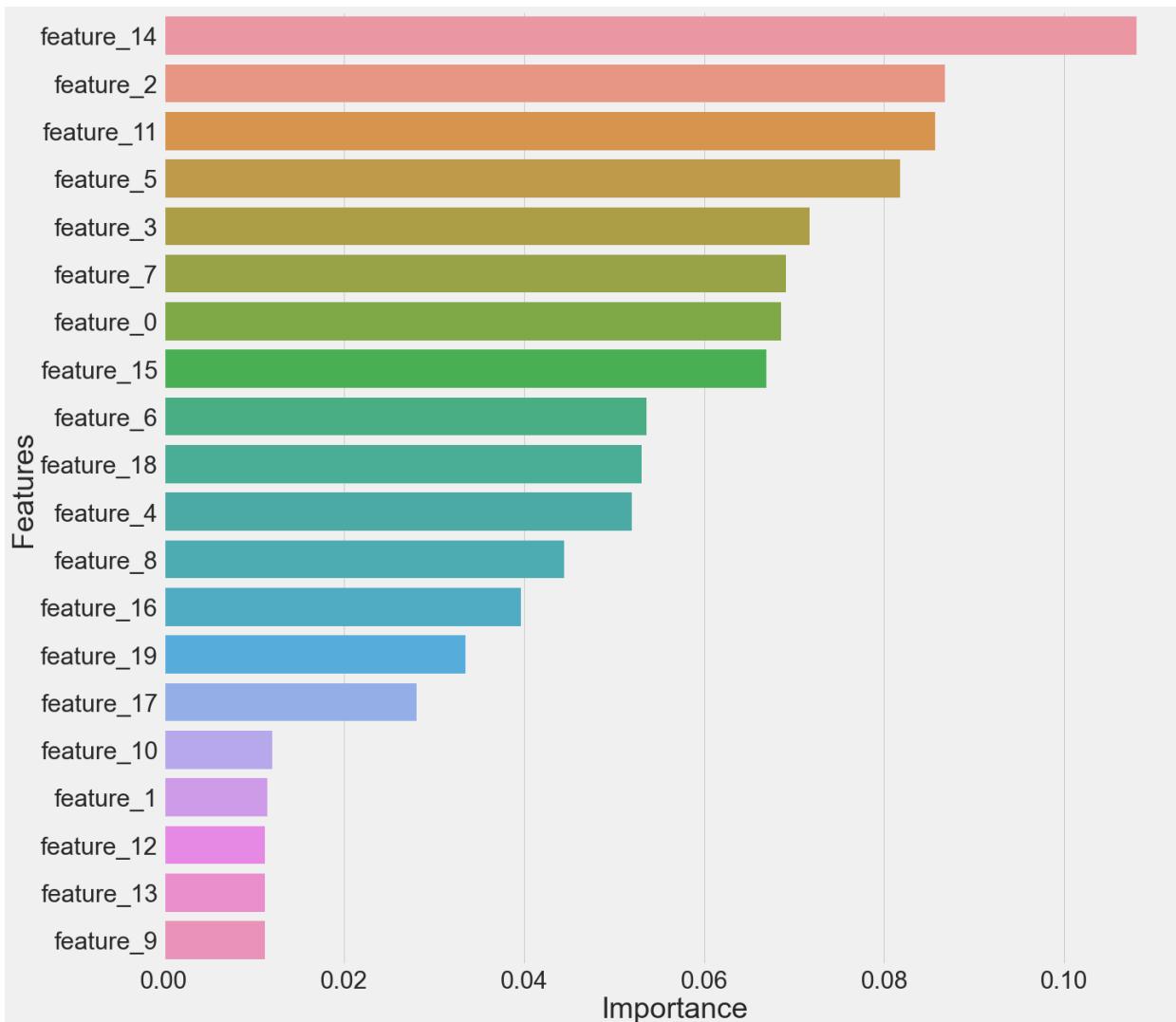
Рассмотрим линейную модель с регуляризацией (L_1 или L_2):

$$\hat{y} = w_0 + w_1 f_1 + w_2 f_2 + \cdots + w_k f_k$$

Если признаки нормированы (значения одного масштаба), то
важность признака f_i равна:

$$Imp(f_i) = |w_i|$$

Пример



Заключение



Вопросы

- ▶ Что такое объект, целевая переменная, признак, модель, функция потерь, функционал ошибки и обучение?
- ▶ Что такое переобучение и недообучение? Как отличить переобучение от недообучения?
- ▶ Что такое кросс-валидация и для чего она используется? Чем применение кросс-валидации лучше, чем разбиение выборки на обучение и контроль?
- ▶ Чем гиперпараметры отличаются от параметров?
- ▶ Запишите формулы для линейной модели регрессии и для среднеквадратичной ошибки. Запишите среднеквадратичную ошибку в матричном виде.
- ▶ Что такое градиент? Какое его свойство используется при минимизации функций?
- ▶ Запишите алгоритм градиентного спуска. Приведите примеры критериев остановки. Как длина шага влияет на процесс оптимизации?
- ▶ Для чего нужно нормировать данные при обучении линейных моделей? Какие способы нормировки вы знаете?
- ▶ Что такое регуляризация? Для чего ее используют в линейных моделях? Запишите L1- и L2-регуляризаторы. Почему L1-регуляризация отбирает признаки?