# Линейные модели

Лекция 3

# Повторение

$X$ – множество объектов (их признаковое описание)

$Y$ – множество истинных ответов

$\hat{Y}$ – множество предсказанных ответов. Получаем по формуле:

$$\hat{Y} = f(X),$$

где $f(X)$ – модель машинного обучения

# Линейная регрессия

Предсказания по формуле:

$$\hat{y} = kX + b$$

$$\hat{y} = \sum_{j=1}^{M} k_j X_j + b$$

$$\hat{y} = \sum_{j=1}^{M} w_j X_j + b$$

$$\hat{y} = \sum_{j=1}^{M} w_j X_j + w_0 \cdot 1$$

# Линейная регрессия

$$\hat{y} = \sum_{j=0}^{M} w_j X_j$$

(представляем, что есть $X_0 = 1$)

Вспомним про $X$:

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

# Линейная регрессия

С учетом того, что добавляем $X_0$

$$X = \begin{pmatrix} 1 & \cdots & 1 \\ x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

Вес каждого признака можно представить в виде вектора:

$$W = (w_0, w_1, \dots, w_m)$$

# Линейная регрессия

Результат предсказания:

$$\hat{y} = (y_1, \dots, y_n)$$

В итоге имеем:

- Матрица признаков $X$ размера (M, N)
- Вектор весов $W$ размера (M)

Хотим:

- Вектор предсказаний размера (N)

# Линейная регрессия. Аналитическое решение

Результат предсказания

$$\hat{y} = X^T W$$
$$(N) = (M, N)^T \cdot (M)$$

Вспоминаем, что хотим приближать $\hat{y}$ к $y$

$$Q(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y - \hat{y})^2$$

# Линейная регрессия. Аналитическое решение

$$Q(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2$$

$$Q(y, X, W) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i^T W\right)^2$$

Перепишем в виде

$$Q(y, X, W) = (Y - X^T W)^2$$

# Линейная регрессия. Аналитическое решение

Продолжим

$$Q(y, X, W) = (Y - XW)^T (Y - XW)$$

$$\frac{dQ(y, X, W)}{dW} = -2X^T (Y - XW)$$

$$X^T (Y - XW) = 0$$

$$\boldsymbol{W} = \left(\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{X}\right)^{-\mathbf{1}} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y}$$

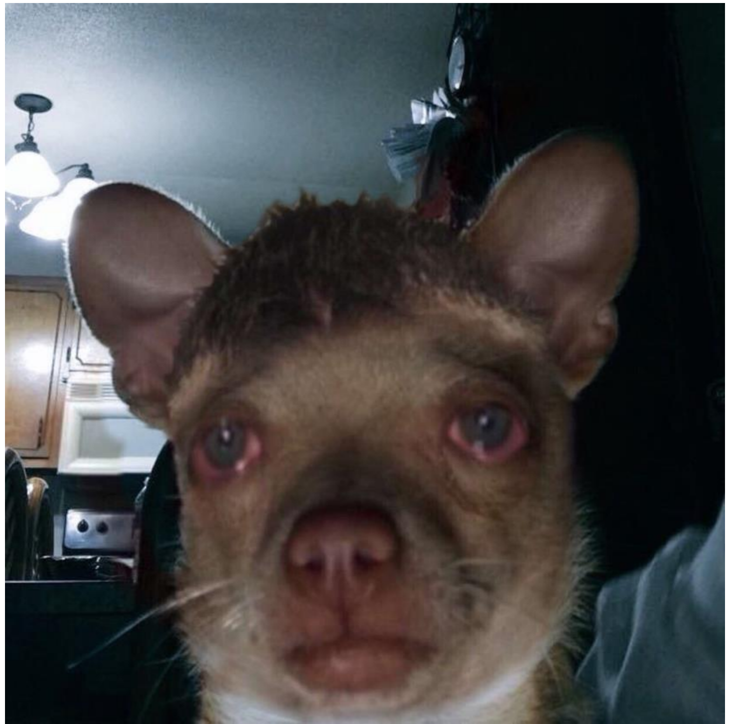
Есть пара **НО**

# Линейная регрессия. Градиентный спуск

Используем итеративный алгоритм:

$$\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{\eta} \cdot \nabla_W \boldsymbol{Q}(\boldsymbol{W})$$

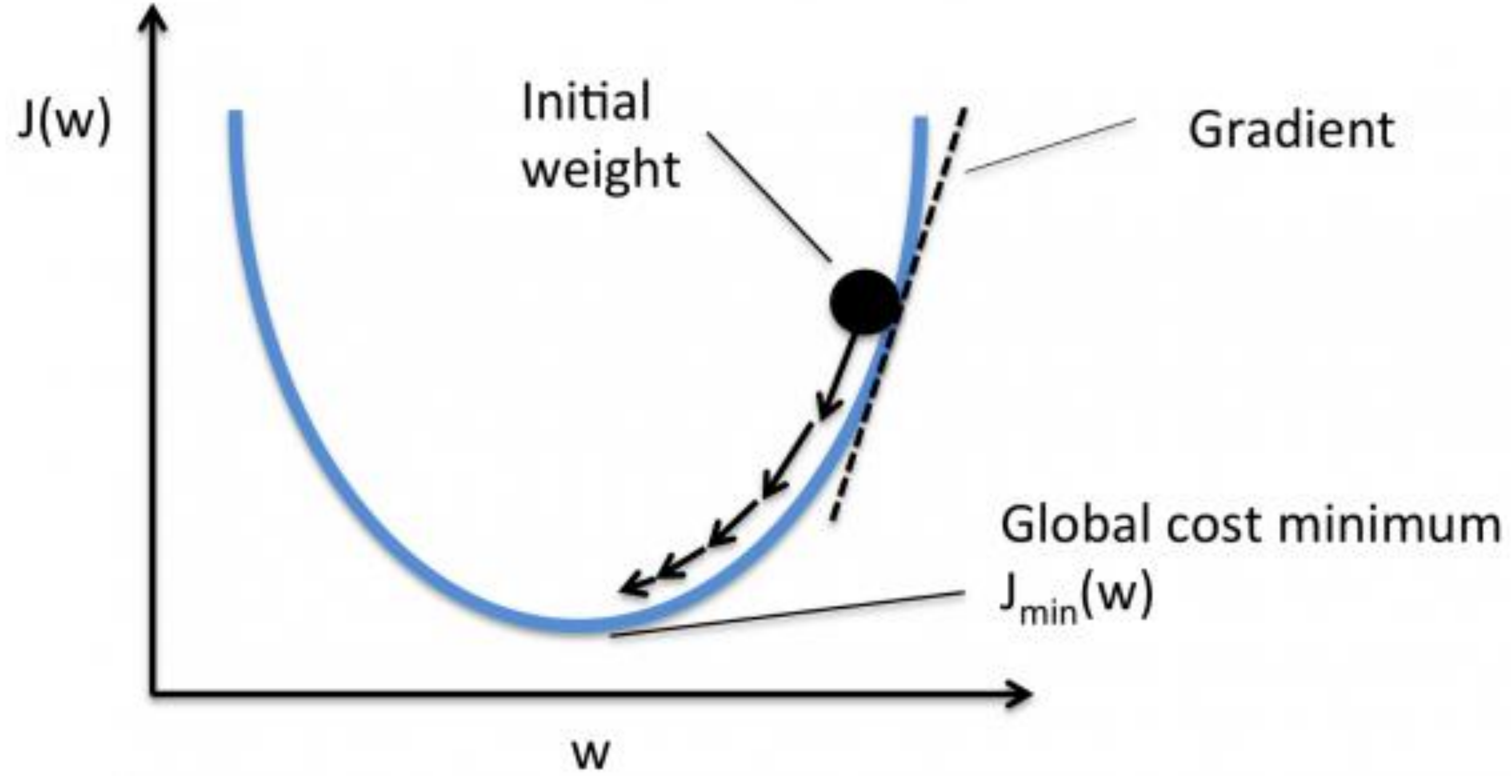
Как мы до этого дошли:



# Линейная регрессия. Градиентный спуск

По определению, градиент показывает рост функции.

Если будем идти по антиградиенту, то придем к минимуму функции.



# Линейная регрессия. Градиентный спуск

Возвращаемся к формуле:

$$\boldsymbol{W = W - \eta \cdot \nabla_W Q}$$

Как найти заветный градиент:

$$\nabla_W Q = \frac{d\left(\sum_{i=1}^{N}\left(y_i - x_i^T W\right)^2\right)}{dW}$$

# Линейная регрессия. Градиентный спуск

Продолжаем

$$\frac{d((Y - X^T W)^2)}{dW}$$

$$\left(\boldsymbol{Y} - \boldsymbol{X^T W}\right)\boldsymbol{X^T}$$

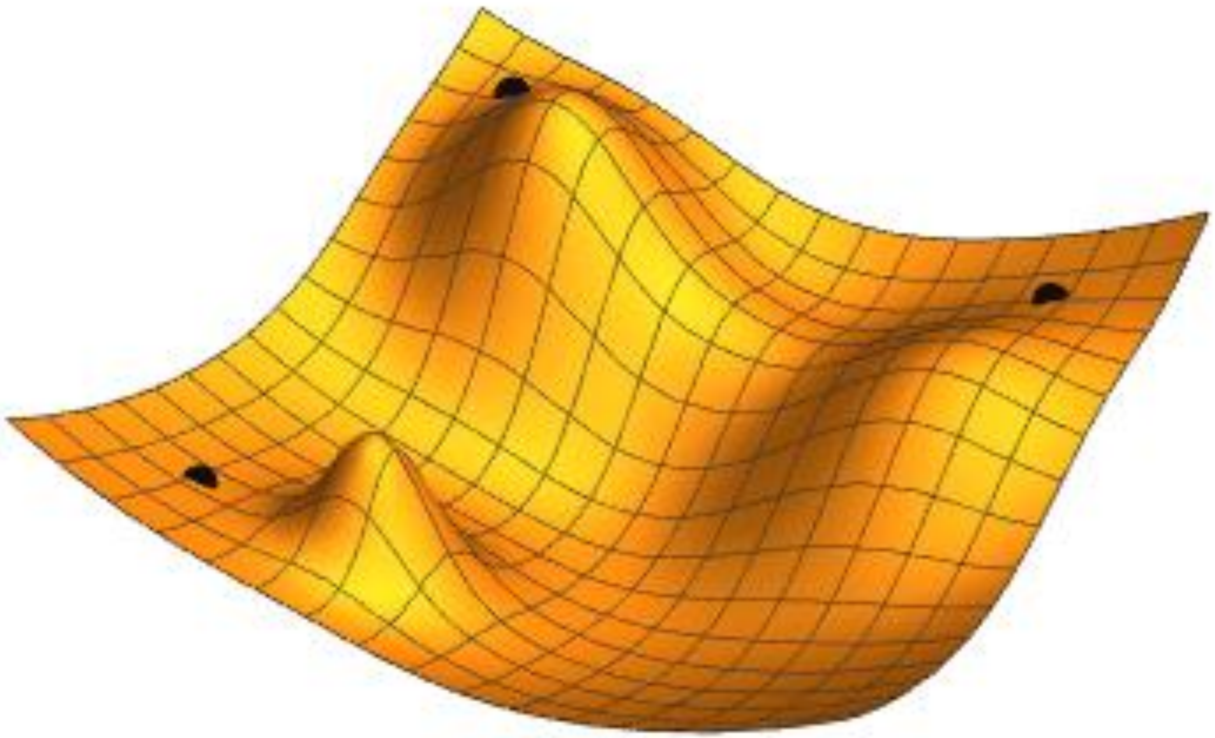
По размерностям:

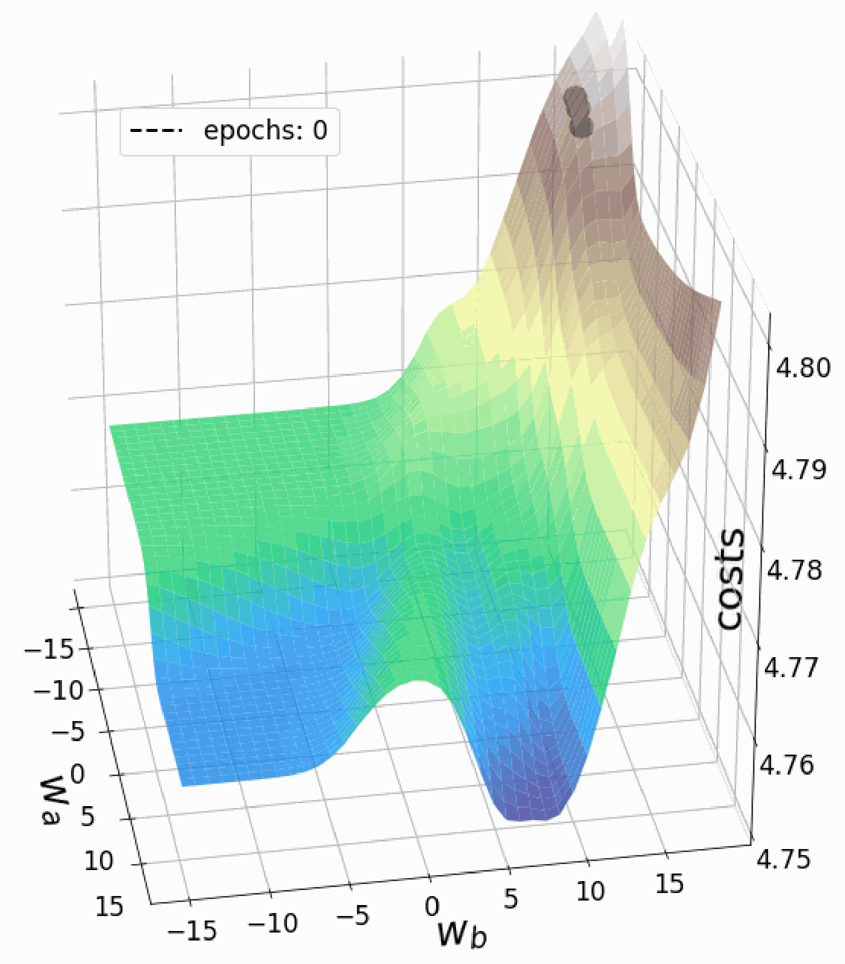
$$\left((N) - (N)\right)(N, M)$$

**В итоге:**

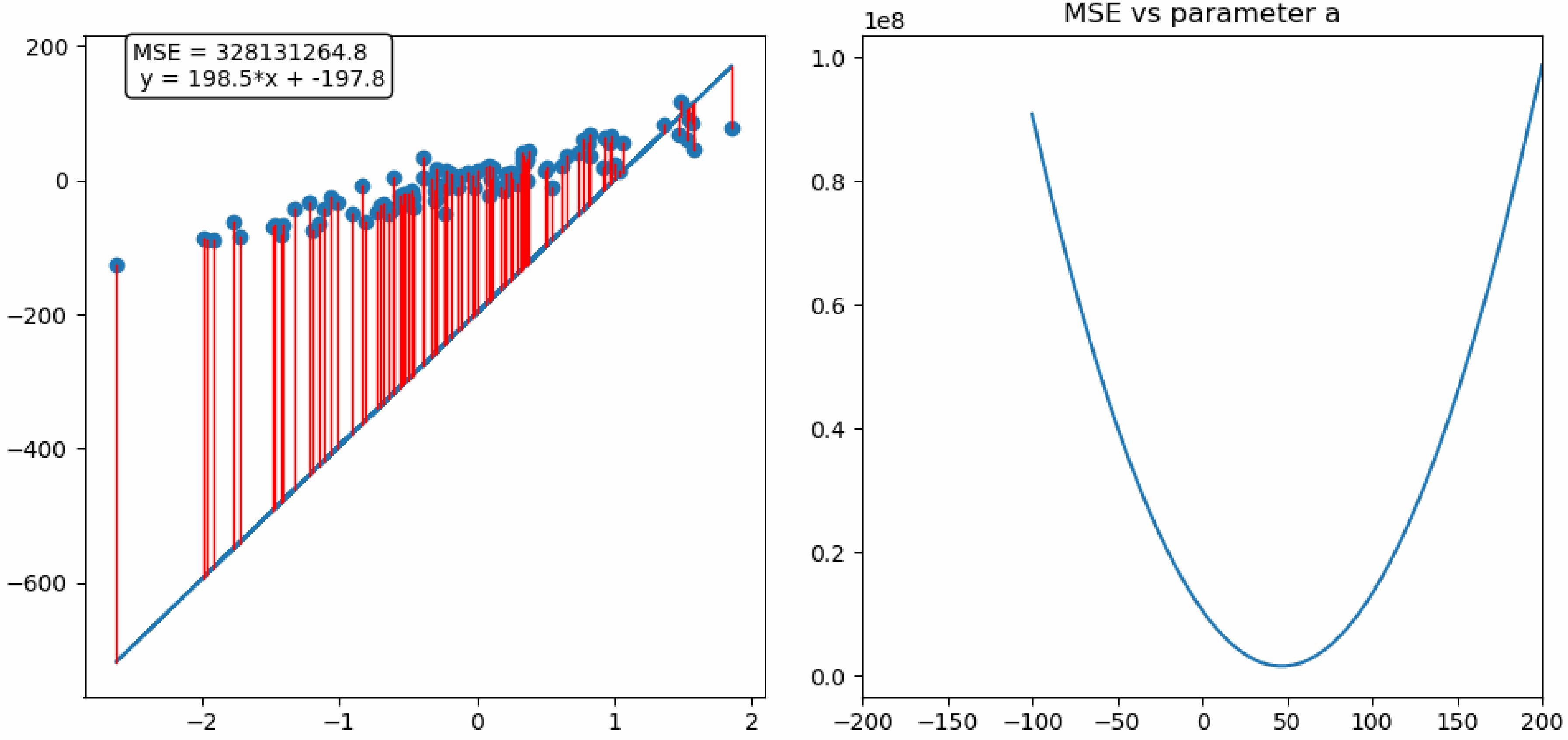
$$\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{X}(\boldsymbol{Y} - \widehat{\boldsymbol{Y}})$$

# Немного визуализации





# Еще визуализация, именно регрессия



# Логистическая регрессия

Решаем задачу классификации

$$Y \in \{0, 1\}$$

При использовании регрессии, получаем:

$$\hat{Y} \in (-\infty;\ +\infty)$$

Необходимо сделать так, чтобы $\hat{Y}$ был похож на $Y$, поэтому примем:

$$\hat{Y} = \sigma(X^T W) \in (0; 1)$$

# Логистическая регрессия

Немного переиначим целевую переменную

$$Y \in \{-1, 1\}$$

И введем понятие отступа (margin) при классификации:

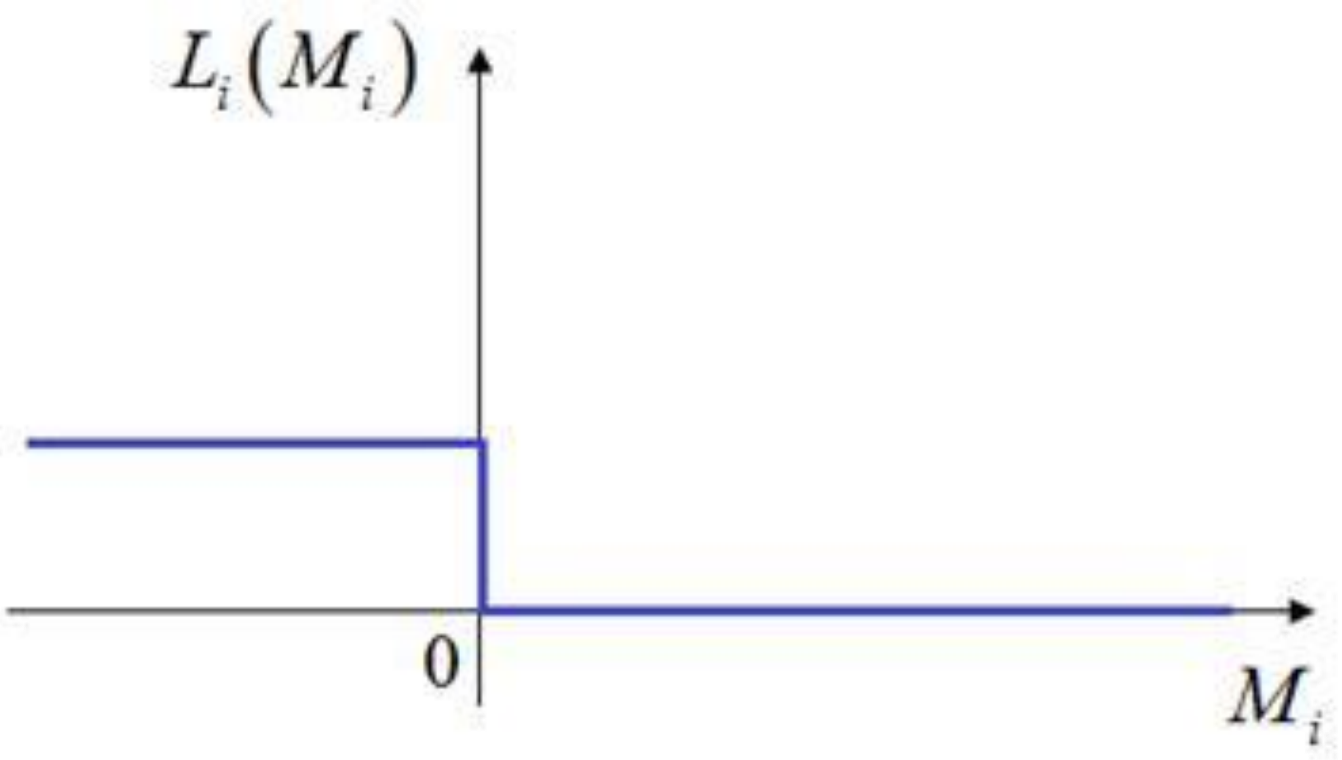
$$M(x_i) = y_i \cdot \langle x_i, w \rangle$$

Тогда функция ошибки:

$$Q = \sum_{i=1}^{N} [M(x_i) < 0]$$

# Логистическая регрессия

Посмотрим на функцию ошибки для одного наблюдения

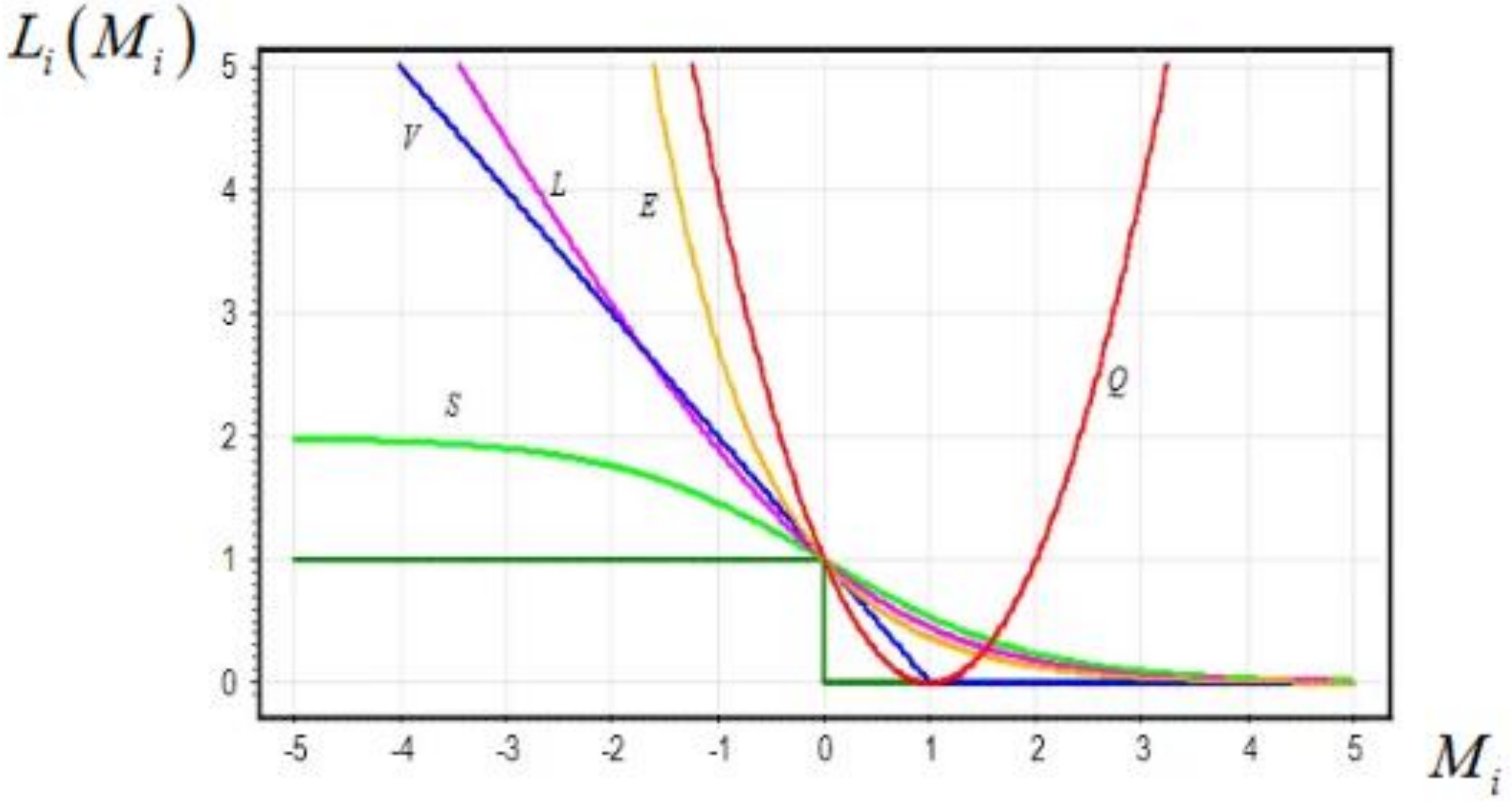


# Логистическая регрессия

$\boldsymbol{V(M) = 1 - M}$. Кусочно-линейная (SVM)

$\boldsymbol{L(M) = log_2(1 - \mathrm{e}^{-\mathrm{M}})}$. Логарифмическая (Log Reg)

$\boldsymbol{E(M) = \mathrm{e}^{-\mathrm{M}}}$. Экспоненциальная (AdaBoost)



# Логистическая регрессия. Еще один вывод логлосс

$Y \in \{0, 1\}$

Обучающая выборка – реализация обобщенной схемы Бернулли. Мы стараемся максимально **правдоподобно** приблизить оценками алгоритма ($\hat{y}$, либо $a$) обучающую выборку.

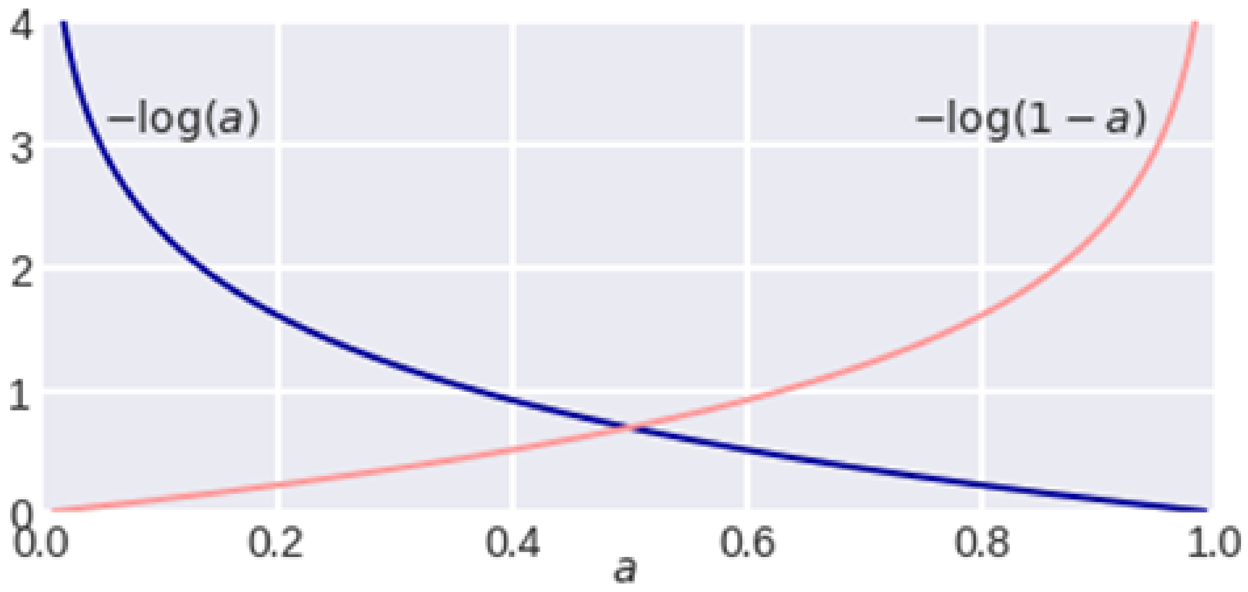
$$p(y|X, w) = \prod_{i=1}^{N} p(y_i|x_i, w) = \prod_{i=1}^{N} a_i^{y_i}(1 - a_i)^{1-y_i} \rightarrow max$$

$$\log\big(p(y|X, w)\big) = -\sum_{i=1}^{N} y_i \log a_i + (1 - y_i) \log(1 - a_i) \rightarrow min$$

# Логистическая регрессия. Вникаем в логлосс

$$-\begin{cases} \log a_i, & y_i = 1, \\ \log(1 - a_i), & y_i = 0. \end{cases}$$

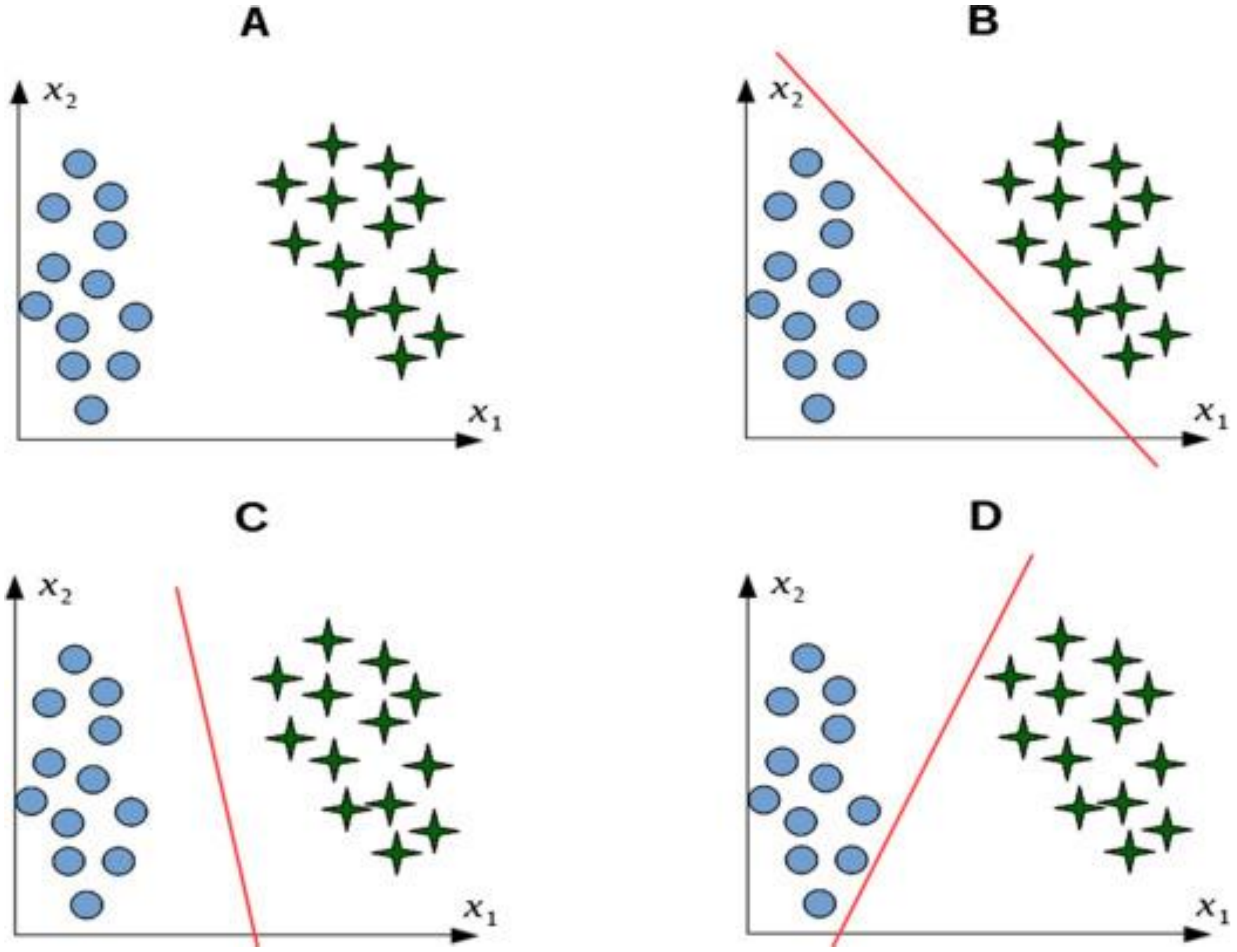
$$\log\big(p(y|X,w)\big) = -\sum_{i=1}^{N} y_i \log a_i + (1 - y_i)\log(1 - a_i) \to min$$



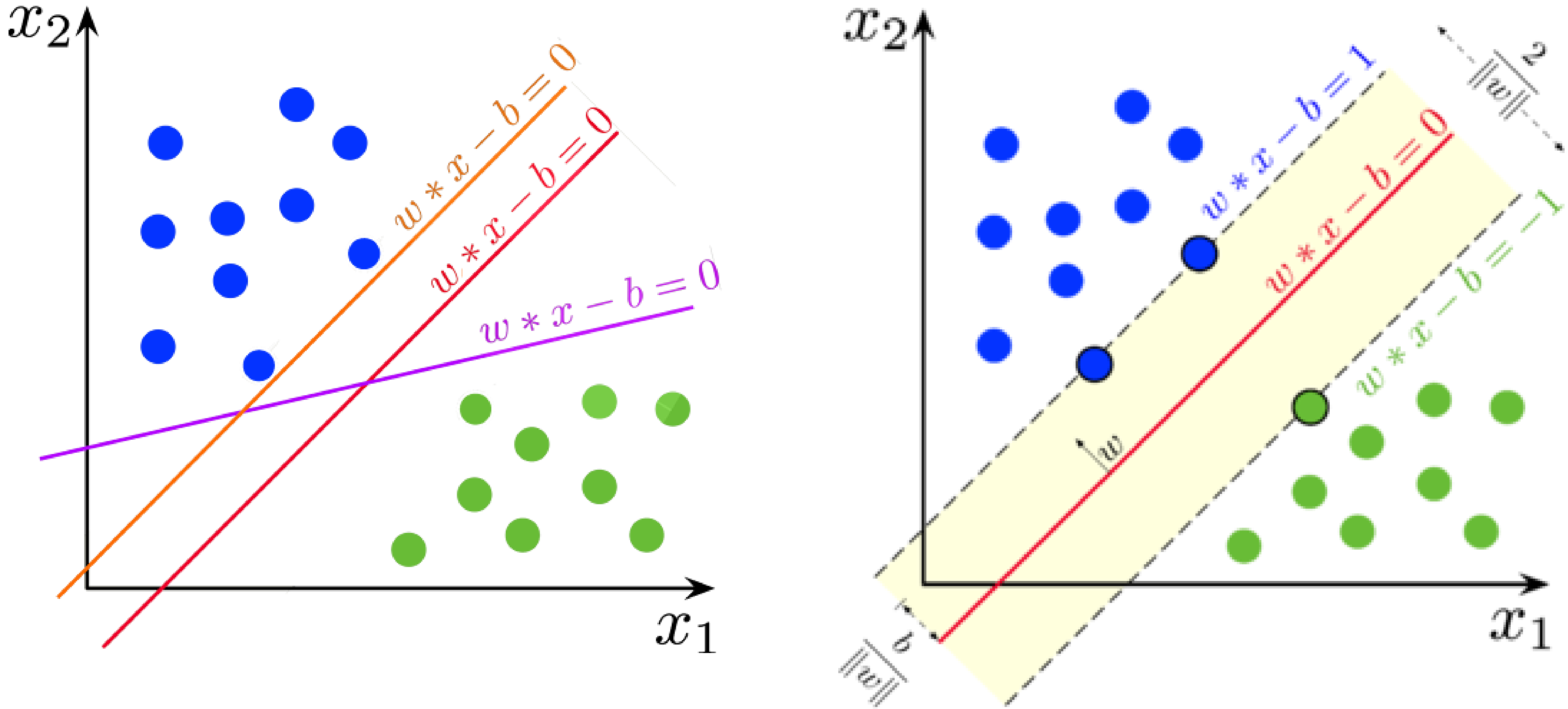
# Метод опорных векторов

$\boldsymbol{V(M) = 1 - M}$. Кусочно-линейная (SVM)

$\boldsymbol{Q(M) = 1 - M + ||W||}$. Функция потерь с максимизацией ширины (SVM)

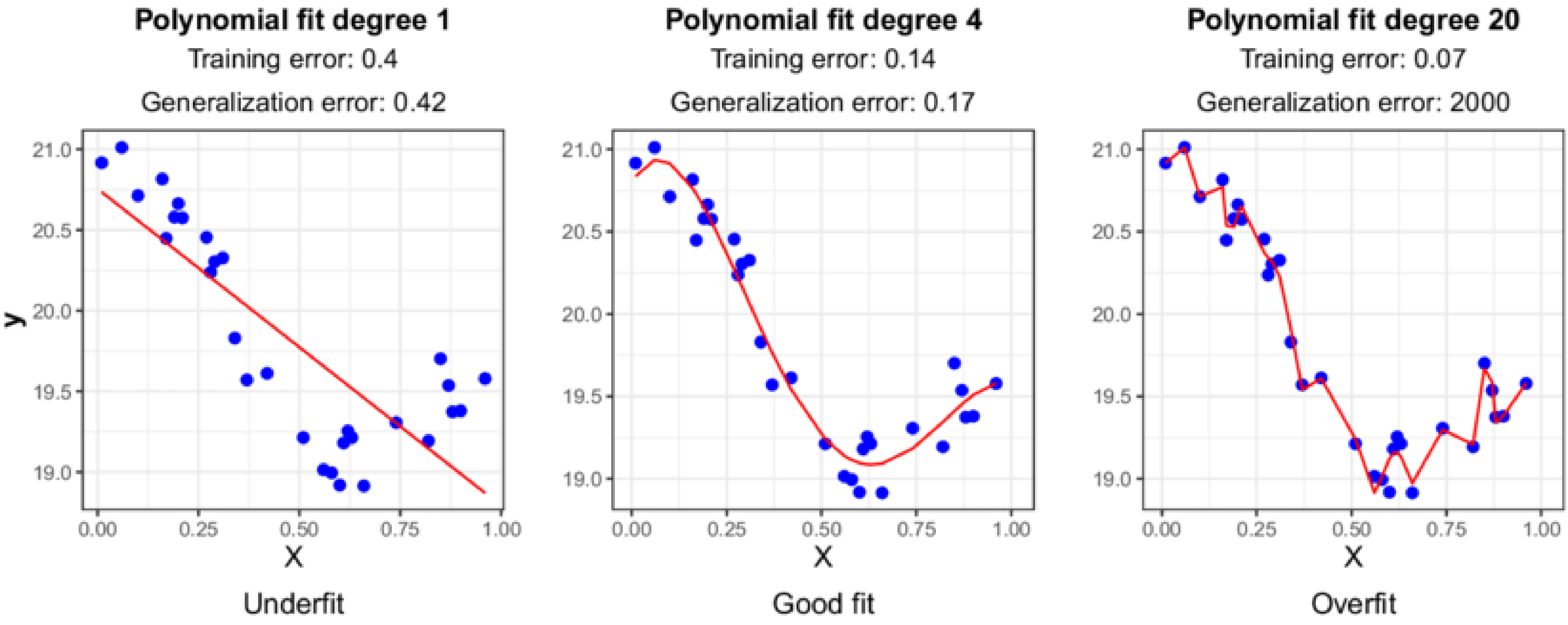


# Метод опорных векторов



# Переобучение при полиномиальной регрессии

$f(X) = w_0 + w_1 x + w_2 x^2 + ... + w_N x^N$



# Регуляризация

Полином 20 степени:

$$f(X) = 1053525 + 2356474x + 45645347x^2 + ... + 5674567x^{20}$$

Полином 3 степени:

$$f(X) = 0.17 + 5.3x + 2.6x^2 - 12.9x^3$$

Делаем вывод:

**Большие веса приводят к переобучению**

# Регуляризация. Определение

**Регуляризация** (англ. *regularization*) в статистике, машинном обучении, теории обратных задач — метод добавления некоторых дополнительных ограничений к условию с целью решить некорректно поставленную задачу или предотвратить переобучение.
Чаще всего эта информация имеет вид штрафа за сложность модели.

# Регуляризация. L1 и L2

Можем модифицировать функцию потерь

L1 регуляризация (LASSO, *Least Absolute Shrinkage and Selection Operator*):

$$Q(y, \hat{y}, W) = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2 + \lambda\sum_{j=0}^{M}|W_j|$$

L2 регуляризация (Ridge, гребневая):

$$Q(y, \hat{y}, W) = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2 + \lambda\sum_{j=0}^{M}W_j^2$$

# Регуляризация. Elastic Net

Elastic Net (L1 + L2):

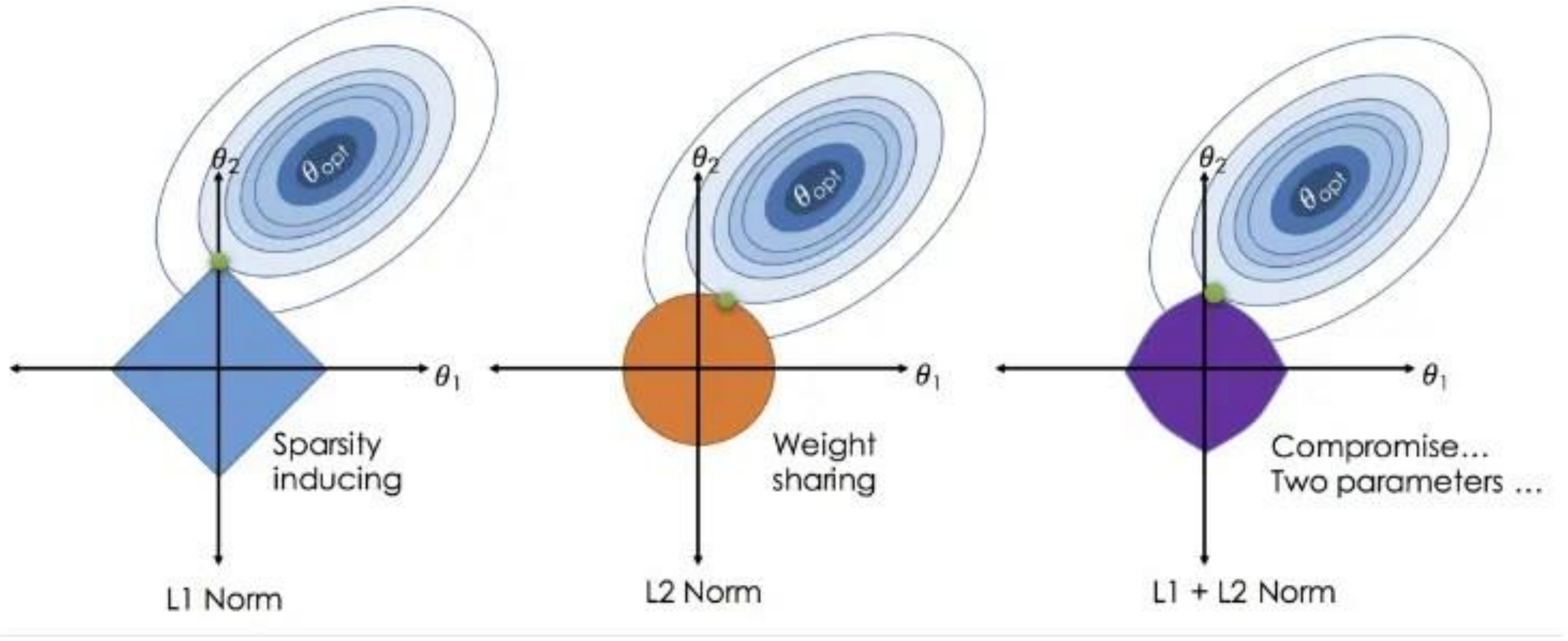
$$Q(y, \hat{y}, W) = \frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2 + \lambda_1 \sum_{j=0}^{M} |W_j| + \lambda_2 \sum_{j=0}^{M} W_j^2$$

Особенности:

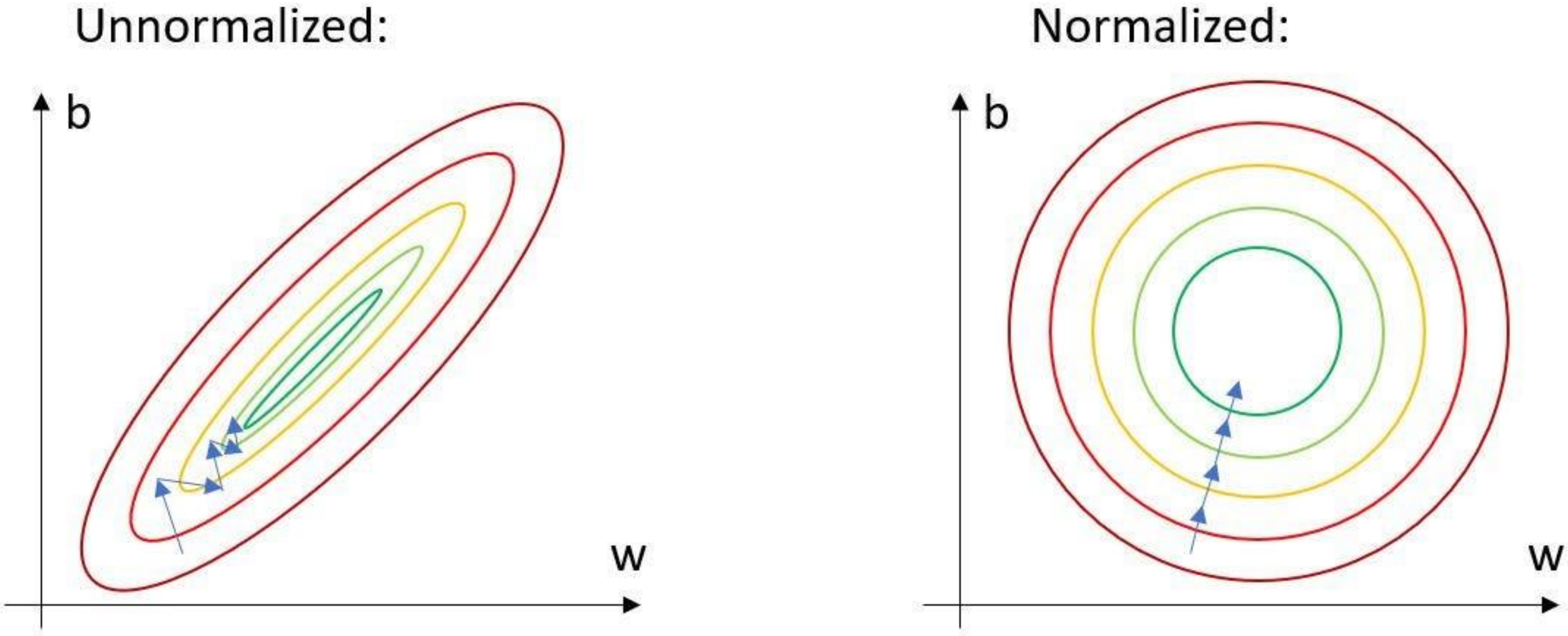
L1-регуляризация зануляет скоррелированные признаки (один из двух). Можно использовать ее в задаче отбора признаков (feature selection).

L2-регуляризация распределит веса скоррелированных признаков равномерно

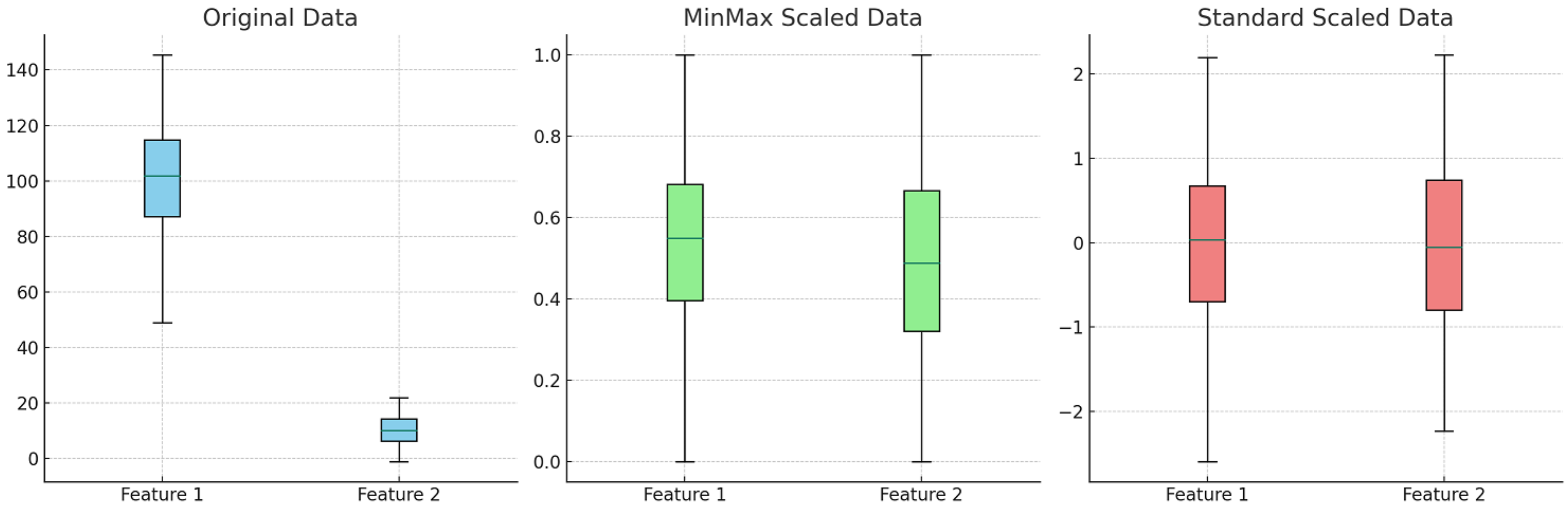
# Регуляризация. Геометрически



# Предобработка признаков. Нормализация



# Предобработка признаков. Нормализация



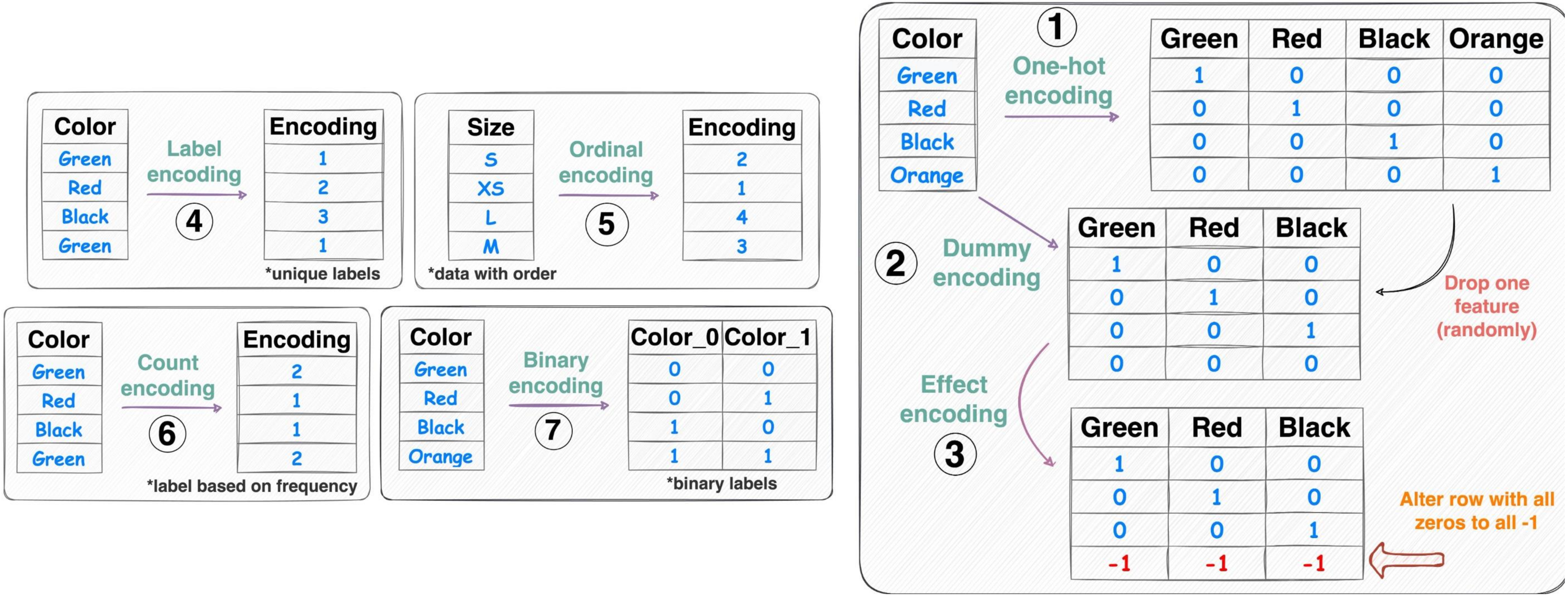

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$z = \frac{x - \mu}{\sigma}$$

$\mu =$ Mean

$\sigma =$ Standard Deviation

# Предобработка признаков. Кодировки категориальных признаков



# Предобработка признаков

| | Color | Target_1 |
|---|---|---|
| 0 | Red | 1 |
| 1 | Red | 1 |
| 2 | Red | 0 |
| 3 | Red | 0 |
| 4 | Red | 0 |
| 5 | Green | 1 |
| 6 | Green | 0 |
| 7 | Green | 0 |

| | Color_Target_1 |
|---|---|
| 0 | 0.400000 |
| 1 | 0.400000 |
| 2 | 0.400000 |
| 3 | 0.400000 |
| 4 | 0.400000 |
| 5 | 0.333333 |
| 6 | 0.333333 |
| 7 | 0.333333 |

Leave-One-Out target encoding