A decorative graphic on the left side of the slide, consisting of a network of blue and teal lines and circles, resembling a circuit board or a neural network diagram.

# Занятие 5

## Линейные модели классификации.

Елена Кантонистова

[elena.kantonistova@yandex.ru](mailto:elena.kantonistova@yandex.ru)

ВШЭ, 2023

The image features a light blue background with decorative circuit-like lines in the corners. These lines are composed of straight segments and small circles, resembling a stylized electronic circuit. They are located in the top-left, top-right, bottom-left, and bottom-right corners, framing the central text.

# СПОСОБЫ КОДИРОВАНИЯ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ

# КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак  $f_j(x)$  принимает  $m$  различных значений:  $C_1, C_2, \dots, C_m$ .

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

# КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак  $f_j(x)$  принимает  $m$  различных значений:  $C_1, C_2, \dots, C_m$ .

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

- Заменяем категориальный признак на  $m$  бинарных признаков:  $b_i(x) = [f_j(x) = C_i]$  (индикатор события).

Тогда One-Hot кодировка для нашего примера будет следующей:

*горький* = (1,0,0,0), *сладкий* = (0,1,0,0),

*солёный* = (0,0,1,0), *кислый* = (0,0,0,1).

# СЧЁТЧИКИ

**Счётчик** (*mean target encoding*) – это вероятность получить значение целевой переменной для данного значения категориального признака.

# СЧЁТЧИКИ (ПРИМЕР)

	feature	target
<b>0</b>	Moscow	0
<b>1</b>	Moscow	1
<b>2</b>	Moscow	1
<b>3</b>	Moscow	0
<b>4</b>	Moscow	0
<b>5</b>	Tver	1
<b>6</b>	Tver	1
<b>7</b>	Tver	1
<b>8</b>	Tver	0
<b>9</b>	Klin	0
<b>10</b>	Klin	0
<b>11</b>	Tver	1

# СЧЁТЧИКИ (ПРИМЕР)

	feature	target
0	Moscow	0
1	Moscow	1
2	Moscow	1
3	Moscow	0
4	Moscow	0
5	Tver	1
6	Tver	1
7	Tver	1
8	Tver	0
9	Klin	0
10	Klin	0
11	Tver	1



	feature	feature_mean	target
0	Moscow	0.4	0
1	Moscow	0.4	1
2	Moscow	0.4	1
3	Moscow	0.4	0
4	Moscow	0.4	0
5	Tver	0.8	1
6	Tver	0.8	1
7	Tver	0.8	1
8	Tver	0.8	0
9	Klin	0.0	0
10	Klin	0.0	0
11	Tver	0.8	1

# СЧЁТЧИКИ: ПРИМЕР

city	target	0	1	2
Moscow	1	$1/4$	$1/2$	$1/4$
London	0	$1/2$	0	$1/2$
London	2	$1/2$	0	$1/2$
Kiev	1	$1/2$	$1/2$	0
Moscow	1	$1/4$	$1/2$	$1/4$
Moscow	0	$1/4$	$1/2$	$1/4$
Kiev	0	$1/2$	$1/2$	0
Moscow	2	$1/4$	$1/2$	$1/4$



## ○ СЧЁТЧИКИ В ЗАДАЧЕ БИНАРНОЙ КЛАССИФИКАЦИИ ○

В случае бинарной классификации счётчики можно задать формулой:

$$Likelihood = \frac{Goods}{Goods + Bads} = mean(target),$$

где *Goods* – число единиц в столбце *target*,

*Bads* – число нулей в столбце *target*.

# СЧЁТЧИКИ (ОБЩАЯ ФОРМУЛА)

- Пусть целевая переменная  $y$  принимает значения от 1 до  $K$ .
- Закодируем категориальную переменную  $f(x)$  следующим способом:

$$counts(u, X) = \sum_{(x,y) \in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y) \in X} [f(x) = u] [y = k], \quad k = 1, \dots, K$$

Тогда кодировка:

$$mean\_target_k(x, X) = \frac{successes_k(f(x), X)}{counts(f(x), X)} \approx p(y = k \mid f(x))$$

# СЧЁТЧИКИ (ОБЩАЯ ФОРМУЛА)

$$counts(u, X) = \sum_{(x,y) \in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y) \in X} [f(x) = u] [y = k], \quad k = 1, \dots, K$$

Тогда кодировка:

$$mean\_target_k(x, X) = \frac{successes_k(f(x), X)}{counts(f(x), X)}$$

Недостаток? Когда такой способ кодирования переобучит наш алгоритм?

# СЧЁТЧИКИ (ОБЩАЯ ФОРМУЛА)

$$counts(u, X) = \sum_{(x,y) \in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y) \in X} [f(x) = u] [y = k], \quad k = 1, \dots, K$$

Тогда кодировка:

$$mean\_target_k(x, X) = \frac{successes_k(f(x), X)}{counts(f(x), X)}$$

*Недостаток? Когда такой способ кодирования переобучит наш алгоритм?*

*Ответ: если в данных много редких категорий.*

# СЧЁТЧИКИ + СГЛАЖИВАНИЕ

Используем счётчики (mean target encoding) со сглаживанием:

$$\frac{\text{mean}(\text{target}) \cdot n_{\text{rows}} + \text{global mean} \cdot \alpha}{n_{\text{rows}} + \alpha},$$

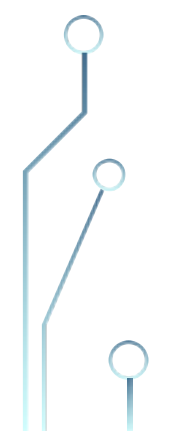
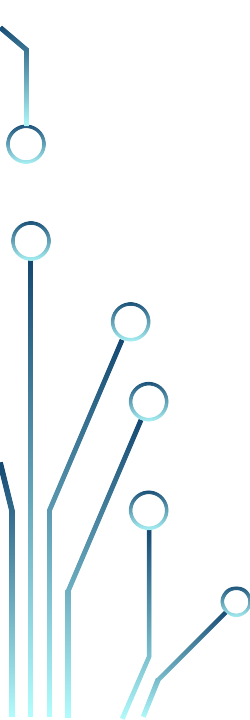

$n_{\text{rows}}$  - количество строк в категории,

$\alpha$  – параметр регуляризации.



# СЧЁТЧИКИ: ОПАСНОСТЬ ПЕРЕОБУЧЕНИЯ

*Вычисляя счётчики, мы закладываем в признаки информацию о целевой переменной и, тем самым, переобучаемся!*



# СЧЁТЧИКИ: КАК ВЫЧИСЛЯТЬ

- Можно вычислять счётчики так:

city	target	
Moscow	1	Вычисляем счетчики по этой части
London	0	
London	2	
Kiev	1	
Moscow	1	Кодируем признак вычисленными счётчиками и обучаемся по этой части
Moscow	0	
Kiev	0	
Moscow	2	

# СЧЁТЧИКИ: КАК ВЫЧИСЛЯТЬ

Более продвинутый способ (по кросс-валидации):

1) Разбиваем выборку  
на  $m$  частей  $X_1, \dots, X_m$

2) На каждой части  $X_i$

значения признаков

вычисляются по

оставшимся частям:

$$x \in X_i \Rightarrow g_k(x) = g_k(x, X \setminus X_i)$$





# БОРЬБА С ПЕРЕОБУЧЕНИЕМ В СЧЁТЧИКАХ

- Вычисление счётчиков по кросс-валидации
- Сглаживание
- Добавление случайных шумов
- Expanding mean

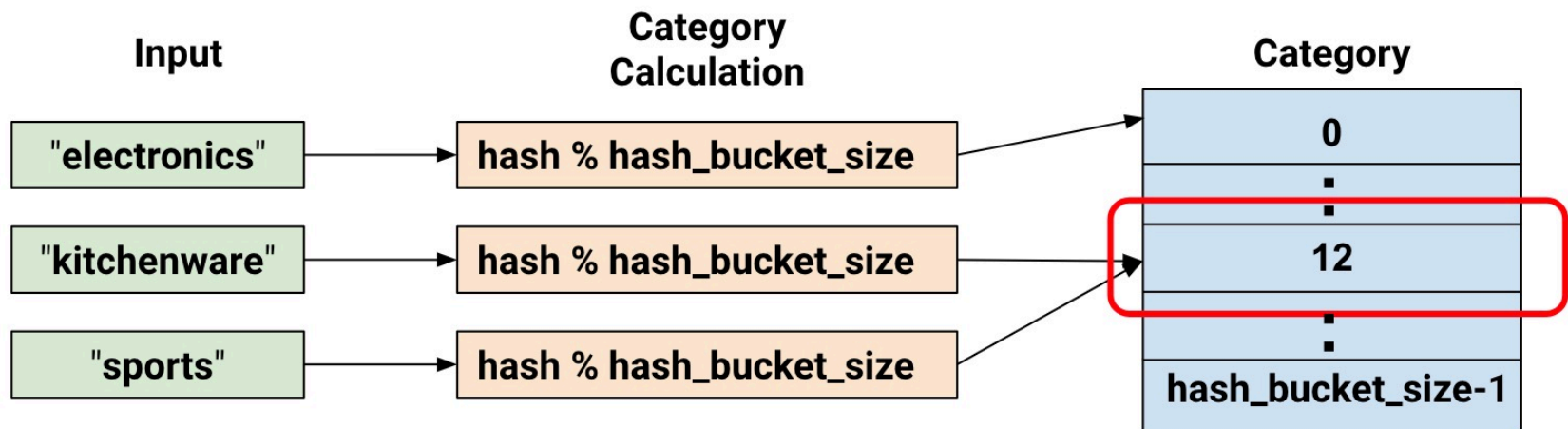
<https://necromuralist.github.io/kaggle-competitions/posts/mean-encoding/#org01e0376>

# ХЭШИРОВАНИЕ ПРИЗНАКОВ

- Если у категориального признака слишком много значений, скажем, миллион, то после применения one-hot кодировки мы получим миллион новых столбцов. С такой огромной матрицей тяжело работать.
- Хэширование развивает идею one-hot кодирования, но позволяет получать любое заранее заданное число новых числовых столбцов после кодировки.

# АЛГОРИТМ ХЭШИРОВАНИЯ

- 1) Для каждого значения признака вычисляем значение некоторой функции – хэш-функции (hash)
- 2) Задаем `hash_bucket_size` – ИТОГОВОЕ количество различных значений категориального признака.
- 3) Берем остаток:  $\text{hash} \% \text{hash\_bucket\_size}$  – тем самым кодируем каждое значение признака числом от 0 до  $\text{hash\_bucket\_size}-1$ .

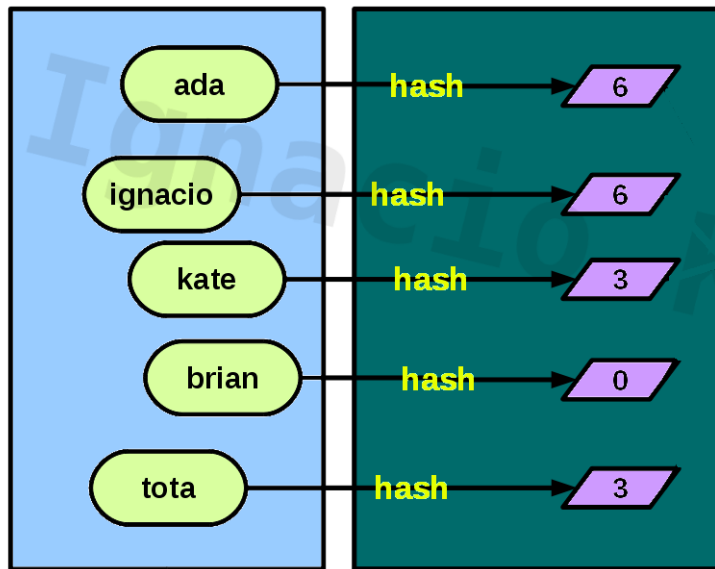


# ЧТО ДЕЛАЕТ ХЭШ-ФУНКЦИЯ

Идея: хэш-функция группирует значения категориального признака:

- часто встречающиеся значения признака формируют отдельные группы
- редко встречающиеся значения попадают в одну группу при группировке

# ХЭШИРОВАНИЕ ПРИЗНАКОВ: ПРИМЕР



elements

hash function

0	1	2	3	4	5	6
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	1	0	0	0
1	0	0	0	0	0	0
0	0	0	1	0	0	0

# ХЭШИРОВАНИЕ

- Хэширование – это способ кодирования категориальных данных, принимающих множество различных значений, показывающий хорошие результаты на практике.
- **Хэширование позволяет закодировать любое значение категориального признака (в том числе то, которого не было в тренировочной выборке).**

Статья про хэширование:

<https://arxiv.org/abs/1509.05472>

The image features a minimalist design with decorative elements in the corners. These elements consist of thin, light blue lines that branch out and terminate in small circles, resembling a stylized circuit board or a network diagram. The lines are positioned in the top-left, top-right, bottom-left, and bottom-right corners, framing the central text.

# ЛИНЕЙНЫЕ МОДЕЛИ КЛАССИФИКАЦИИ

# ОБУЧЕНИЕ ЛИНЕЙНОЙ РЕГРЕССИИ (НАПОМИНАНИЕ)

Обучающая выборка:

пусть  $x$  – объект ( $x_1, x_2, \dots, x_l$  - его признаки), а  $y$  – ответ на объекте (произвольное число),  $n$  – количество объектов.

Модель линейной регрессии:

$$a(x, w) = \sum_{j=1}^l w_j x_j$$

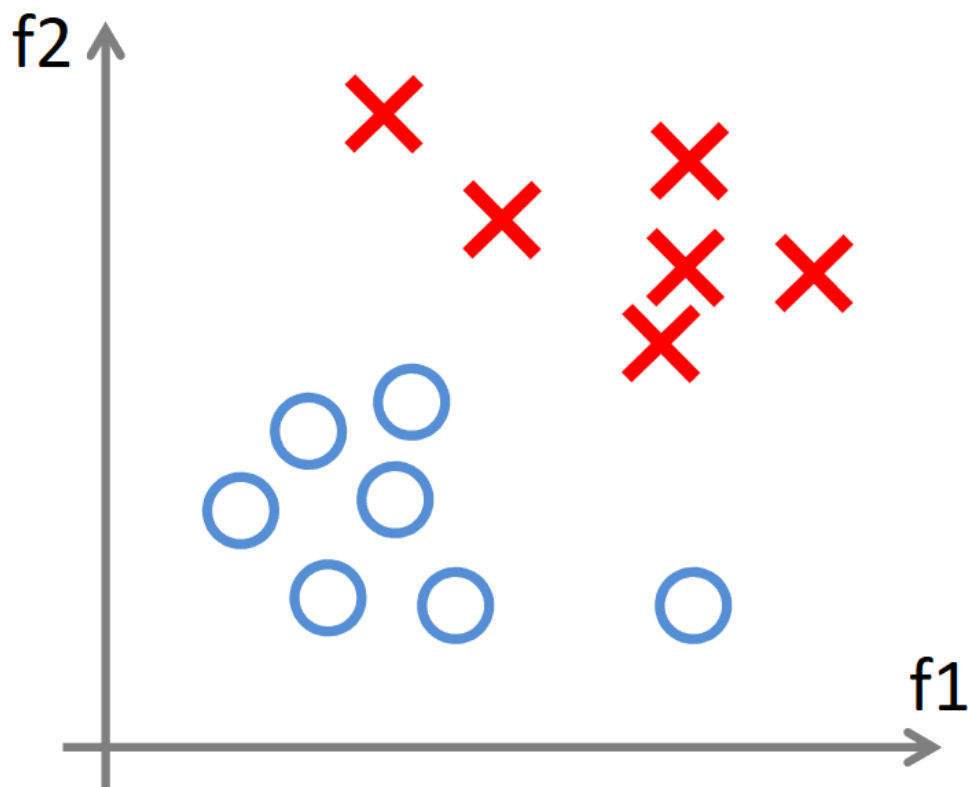
- Метод обучения – метод наименьших квадратов (**минимизируем разность между предсказанием и правильным ответом**):

$$Q(w) = \sum_{i=1}^n (a(x_i, w) - y_i)^2 \rightarrow \min_w$$



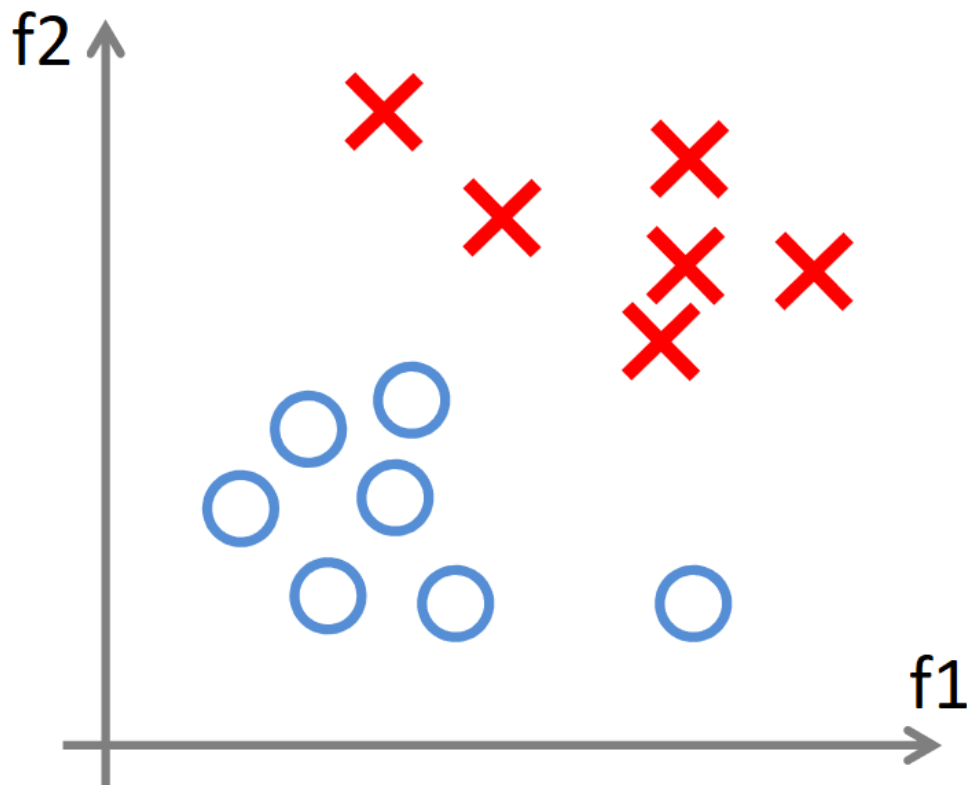
# БИНАРНАЯ КЛАССИФИКАЦИЯ

$y_1, y_2, \dots, y_n$  - ОТВЕТЫ (+1 или -1).



# БИНАРНАЯ КЛАССИФИКАЦИЯ

$y_1, y_2, \dots, y_n$  - ОТВЕТЫ (+1 или -1).



Как выглядит модель линейного классификатора:

$$a(x, w) = ?$$

# БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \textit{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

# БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \textit{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

- если  $\sum_{j=1}^l w_j x_j > 0$ , то  $\textit{sign}\left(\sum_{j=1}^l w_j x_j\right) = +1$ , то есть объект отнесён к положительному классу
- если  $\sum_{j=1}^l w_j x_j < 0$ , то  $\textit{sign}\left(\sum_{j=1}^l w_j x_j\right) = -1$ , то есть объект отнесён к отрицательному классу

# БИНАРНАЯ КЛАССИФИКАЦИЯ

Модель линейного классификатора:

$$a(x, w) = \text{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

- если  $\sum_{j=1}^l w_j x_j > 0$ , то  $\text{sign}\left(\sum_{j=1}^l w_j x_j\right) = +1$ , то есть объект отнесён к положительному классу
- если  $\sum_{j=1}^l w_j x_j < 0$ , то  $\text{sign}\left(\sum_{j=1}^l w_j x_j\right) = -1$ , то есть объект отнесён к отрицательному классу
- значит,  $\sum_{j=1}^l w_j x_j = 0$  – уравнение разделяющей границы между классами. Это уравнение плоскости (или прямой в двумерном случае), поэтому классификатор является линейным.

# БИНАРНАЯ КЛАССИФИКАЦИЯ

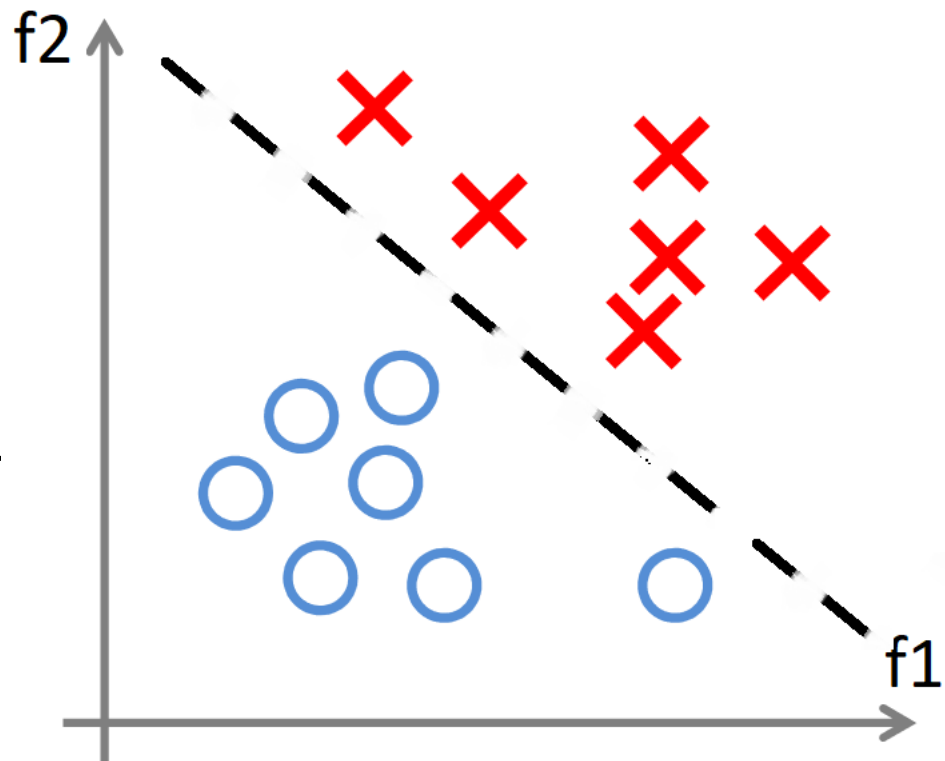
Модель линейного классификатора:

$$a(x, w) = \text{sign}\left(\sum_{j=1}^l w_j x_j\right)$$

Уравнение

$$\sum_{j=1}^l w_j x_j = 0$$

– уравнение плоскости  
(или прямой).





# ОБУЧЕНИЕ КЛАССИФИКАТОРА

*Как обучить линейный классификатор?*



# ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min,$$

где  $[a(x_i) \neq y_i] = 1$ , если предсказание на объекте неверное, то есть  $a(x_i) \neq y_i$ , и 0 иначе.



# ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min (*),$$

где  $[a(x_i) \neq y_i] = 1$ , если предсказание на объекте неверное, то есть  $a(x_i) \neq y_i$ , и 0 иначе.

- Обозначим  $M_i = y_i \cdot (w, x_i)$  - **отступ** на  $i$ -м объекте.

# ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] \rightarrow \min (*),$$

где  $[a(x_i) \neq y_i] = 1$ , если предсказание на объекте неверное, то есть  $a(x_i) \neq y_i$ , и 0 иначе.

- Обозначим  $M_i = y_i \cdot (w, x_i)$  - **отступ** на  $i$ -м объекте.

**Утверждение.** Решение задачи (\*) эквивалентно решению задачи

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

# ДОКАЗАТЕЛЬСТВО УТВЕРЖДЕНИЯ

- Обучение - минимизация доли ошибок классификатора:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) \neq y_i] = \frac{1}{n} \sum_{i=1}^n [\text{sign}(w, x_i) \neq y_i] \rightarrow \min$$

Функционал  $Q$  можно переписать в виде:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [y_i \cdot (w, x_i) < 0] = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- $M_i = y_i \cdot (w, x_i)$  - **отступ**

## ОТСТУП (MARGIN)

Знак отступа  $M = y \cdot (w, x)$  говорит о корректности классификации на объекте:

## ОТСТУП (MARGIN)

Знак отступа  $M = y \cdot (w, x)$  говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

- Если  $(w, x) > 0$  (то есть объект отнесён к классу  $+1$ ), а  $y = -1$ , то  $M = y \cdot (w, x) < 0$ .

## ОТСТУП (MARGIN)

Знак отступа  $M = y \cdot (w, x)$  говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

- Если  $(w, x) > 0$  (то есть объект отнесён к классу  $+1$ ), а  $y = -1$ , то  $M = y \cdot (w, x) < 0$ .
- Аналогично, если  $(w, x) < 0$ , а  $y = +1$ , то  $M = y \cdot (w, x) < 0$ .

## ОТСТУП (MARGIN)

Знак отступа  $M = y \cdot (w, x)$  говорит о корректности классификации на объекте:

Случаи неверной классификации (предсказание не совпадает с правильным ответом):

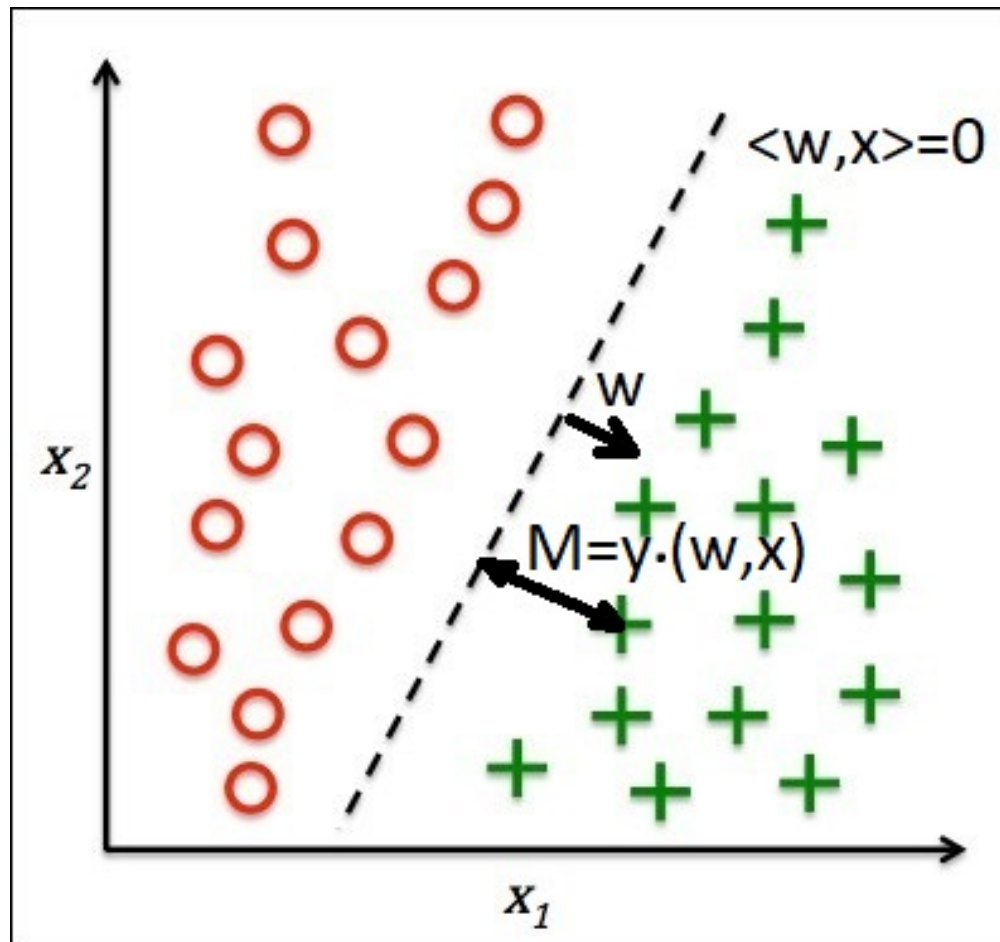
- Если  $(w, x) > 0$  (то есть объект отнесён к классу  $+1$ ), а  $y = -1$ , то  $M = y \cdot (w, x) < 0$ .
- Аналогично, если  $(w, x) < 0$ , а  $y = +1$ , то  $M = y \cdot (w, x) < 0$ .

Случаи верной классификации:

- Если  $(w, x) > 0$  и  $y = +1$  или  $(w, x) < 0$  и  $y = -1$  получаем  $M = y \cdot (w, x) > 0$ .

## ОТСТУП (MARGIN)

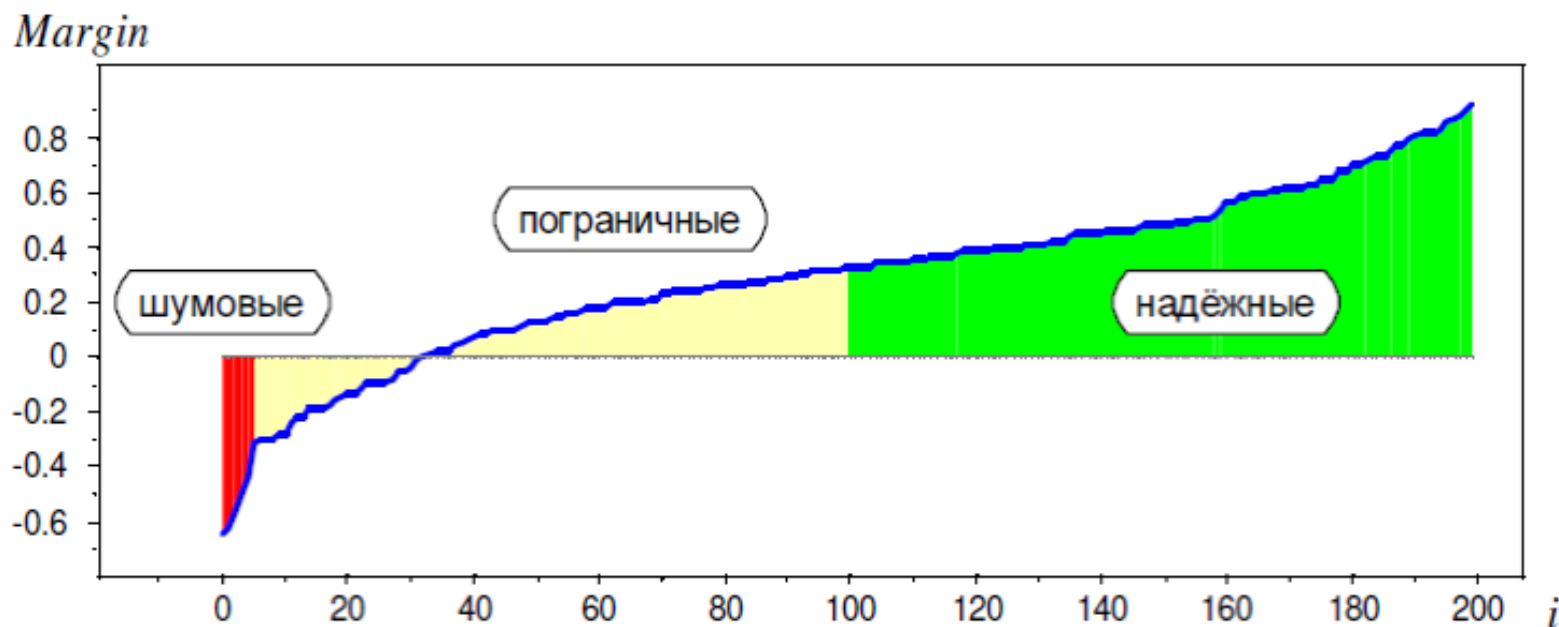
**Абсолютная величина отступа  $M$  обозначает степень уверенности классификатора в ответе (чем ближе  $M$  к нулю, тем меньше уверенность в ответе)**





# ОТСТУП (MARGIN)

Ранжирование объектов по возрастанию отступа:



# ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация **пороговой функции потерь**:

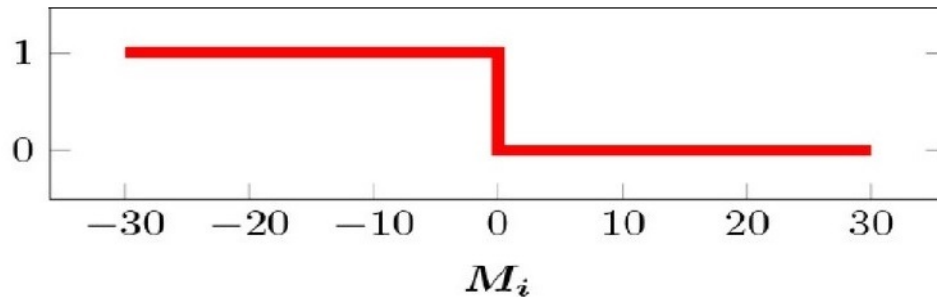
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [\mathbf{M}_i < \mathbf{0}] \rightarrow \min$$

# ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация **пороговой функции потерь**:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [\mathbf{M}_i < \mathbf{0}] \rightarrow \min$$

- Пороговая функция потерь **разрывна**, и этот факт сильно затрудняет процесс минимизации.

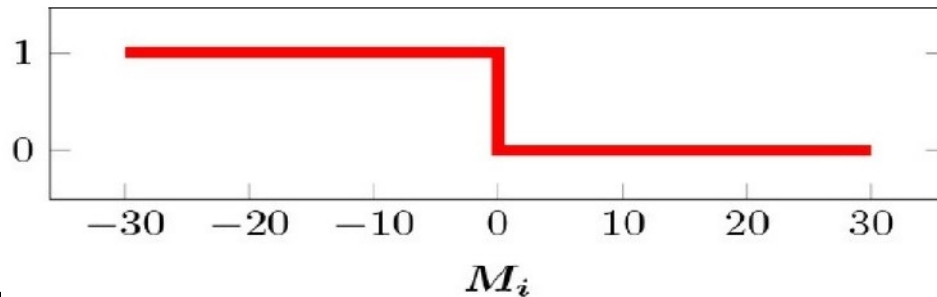


# ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

- Ранее мы показали, что обучение классификатора – это минимизация **пороговой функции потерь**:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.



- Для решения этой проблемы используют **другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции**.

# ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

Ранее мы показали, что обучение классификатора – это минимизация **пороговой функции потерь**:

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n [M_i < 0] \rightarrow \min$$

- Пороговая функция потерь разрывна, и этот факт сильно затрудняет процесс минимизации.
- Для решения этой проблемы используют другие функции потерь – непрерывные или гладкие, как правило, являющиеся верхними оценками пороговой функции.
- Задача минимизации некоторой функции потерь называется **минимизацией эмпирического риска** (сама функция потерь – эмпирический риск).

# ВЕРХНИЕ ОЦЕНКИ ЭМПИРИЧЕСКОГО РИСКА

- $L(a, y) = L(M) = [M < 0]$  – разрывная функция потерь

Оценим

$L(M) \leq \tilde{L}(M)$ , где  $\tilde{L}(M)$  - непрерывная или гладкая функция потерь.

- Тогда

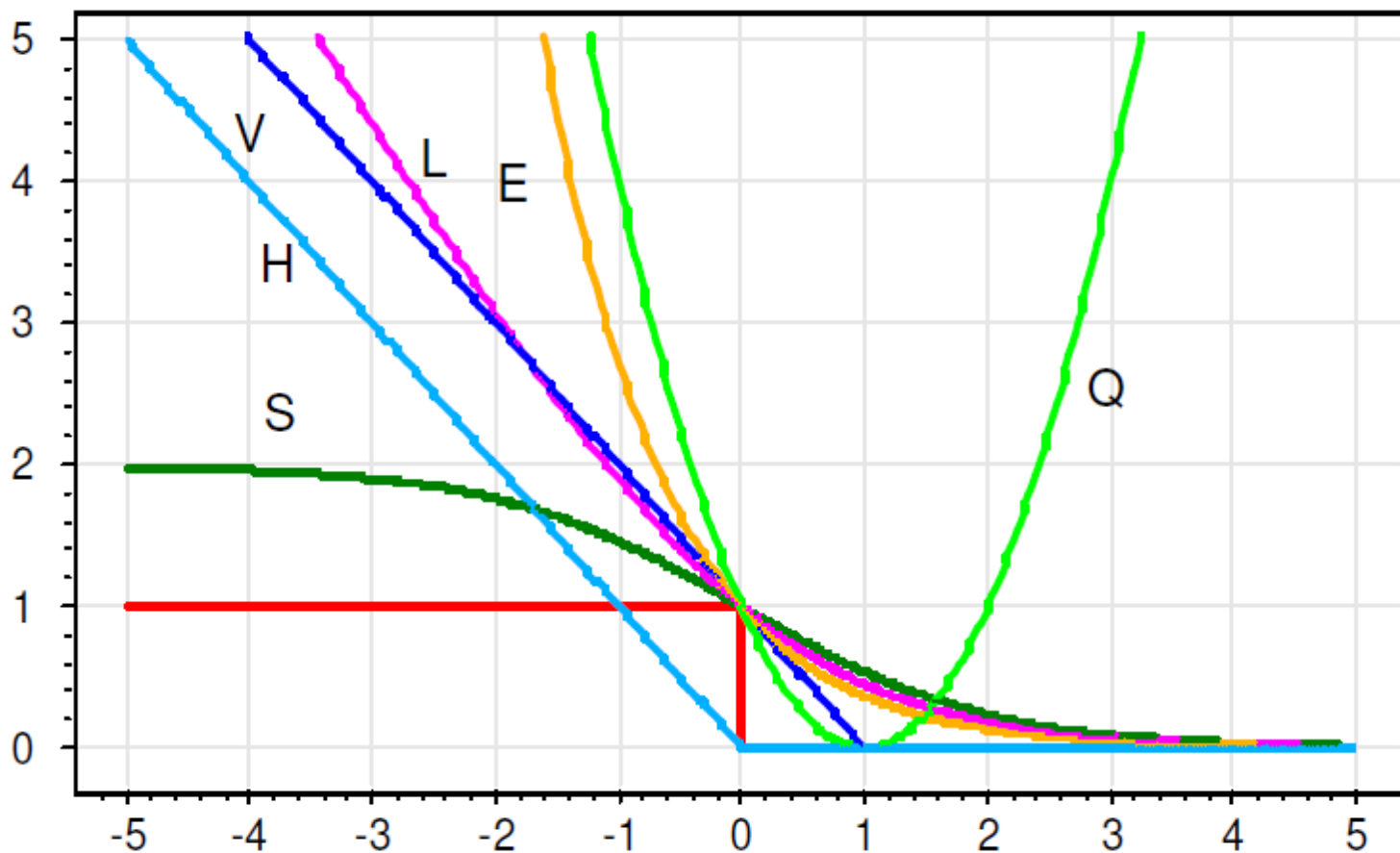
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n L(y_i \cdot (w, x_i)) \leq \frac{1}{n} \sum_{i=1}^n \tilde{L}(y_i \cdot (w, x_i)) \rightarrow \min$$

# ФУНКЦИИ ПОТЕРЬ

Минимизируя различные функции потерь, получаем разные результаты. Поэтому разные функции потерь определяют различные классификаторы.

- $L(M) = \log(1 + e^{-M})$  – логистическая функция потерь
- $V(M) = (1 - M)_+ = \max(0, 1 - M)$  – кусочно-линейная функция потерь (метод опорных векторов)
- $H(M) = (-M)_+ = \max(0, -M)$  – кусочно-линейная функция потерь (персептрон)
- $E(M) = e^{-M}$  - экспоненциальная функция потерь
- $S(M) = \frac{2}{1 + e^{-M}}$  - сигмоидная функция потерь
- $[M < 0]$  – пороговая функция потерь

# ФУНКЦИИ ПОТЕРЬ



$M$



# ОПТИМИЗАЦИЯ ФУНКЦИОНАЛА ПОТЕРЬ

- Нахождение минимума функции потерь  $Q$  происходит с помощью метода градиентного спуска:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \eta \cdot \nabla Q(\mathbf{w}^{(k-1)})$$