

# Линейные модели классификации.

# Recap

# ОВЕРФИТТИНГ В ЛИНЕЙНОЙ РЕГРЕССИИ

---

# ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ

- Мы говорили о регрессии с базисными функциями:

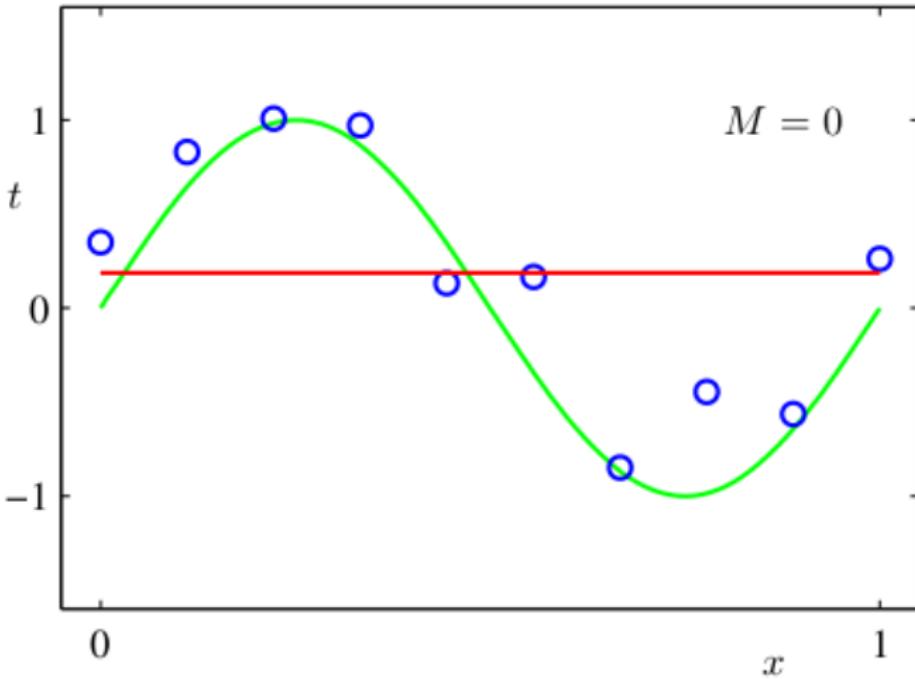
$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^M w_j \phi_j(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}).$$

- Давайте для примера рассмотрим такую регрессию для  $\phi_j(x) = x^j$ , т.е.

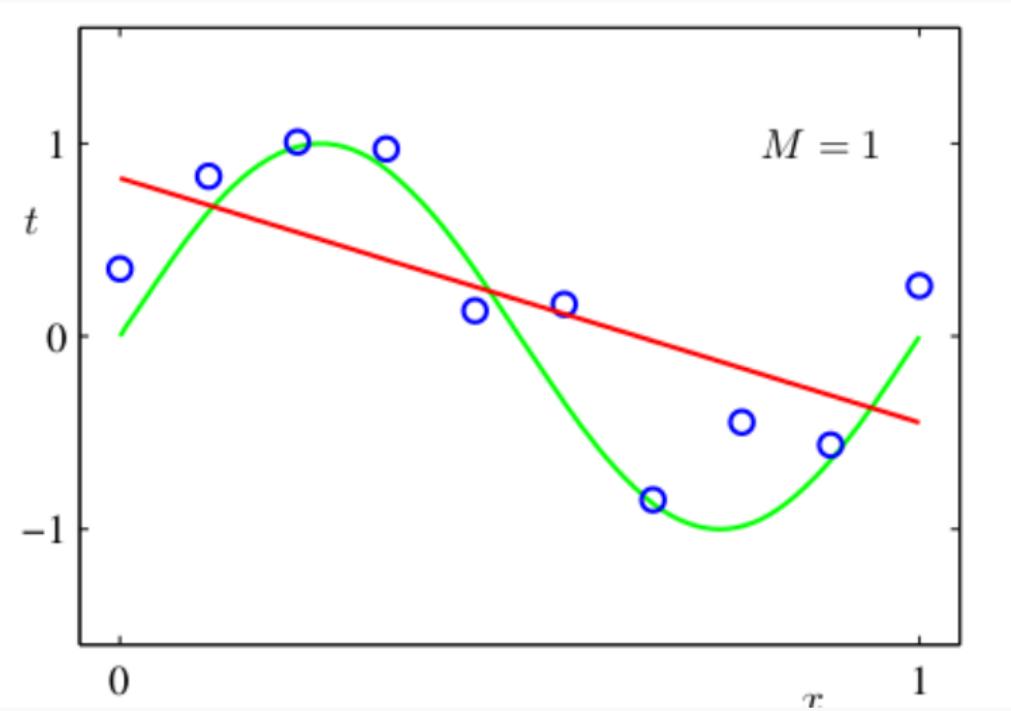
$$f(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M.$$

- И будем, как раньше, минимизировать квадратичную ошибку.
- Пример с кодом.

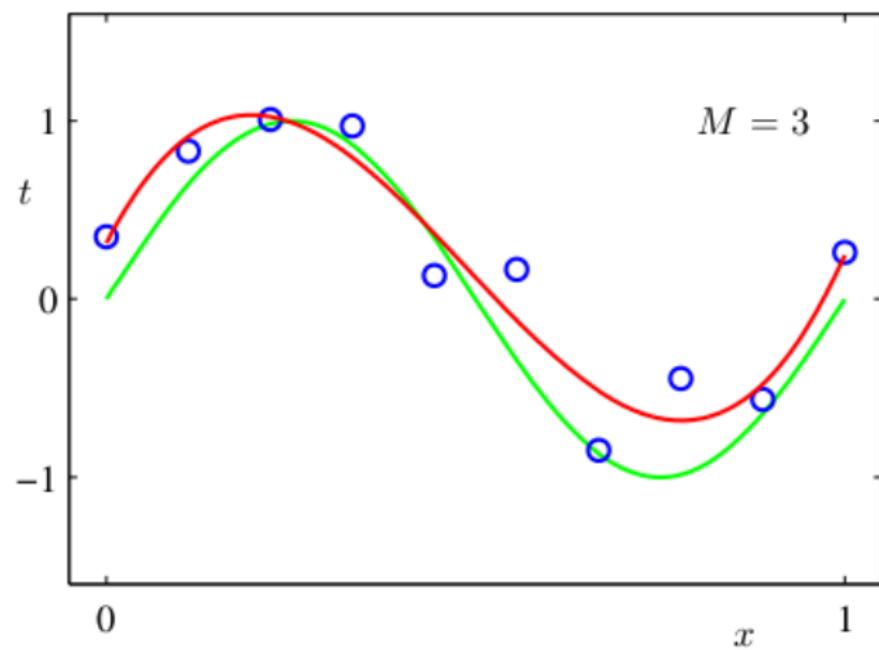
# ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ



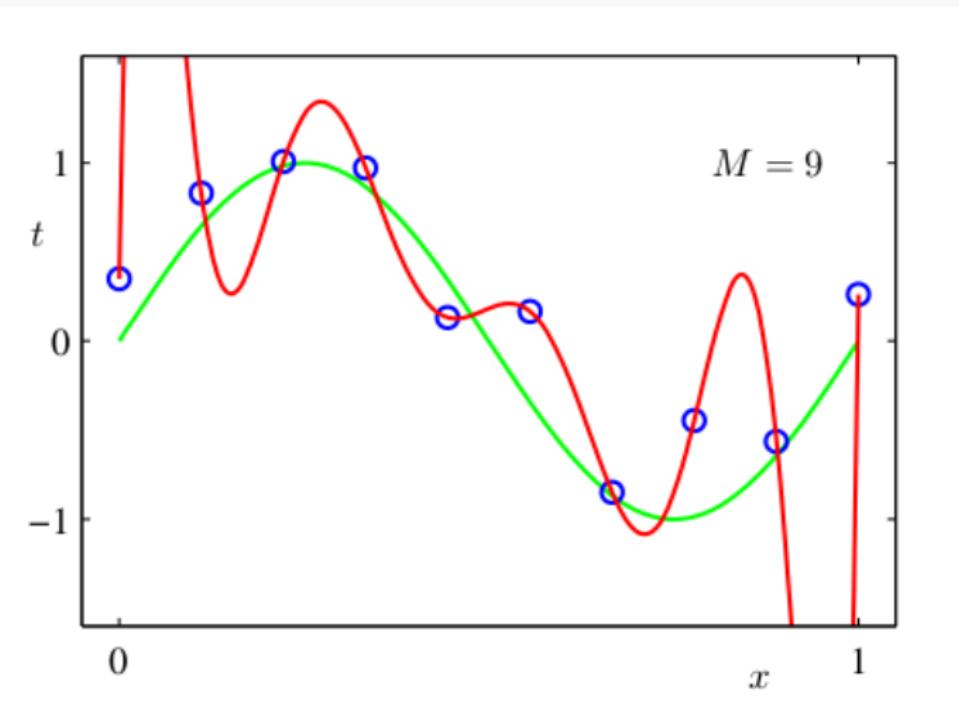
# ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ



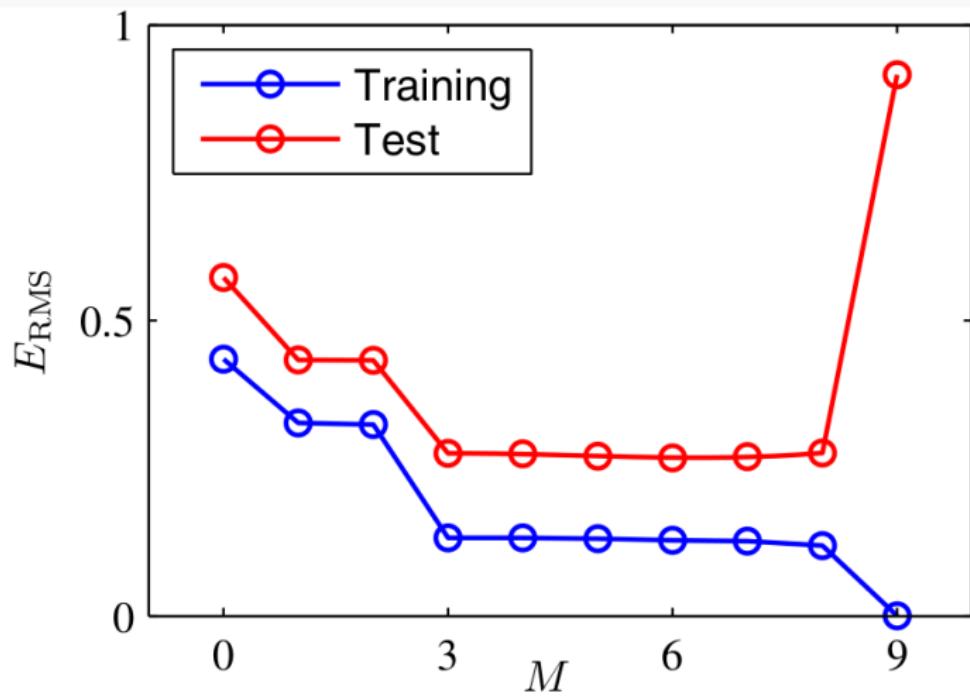
# ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ



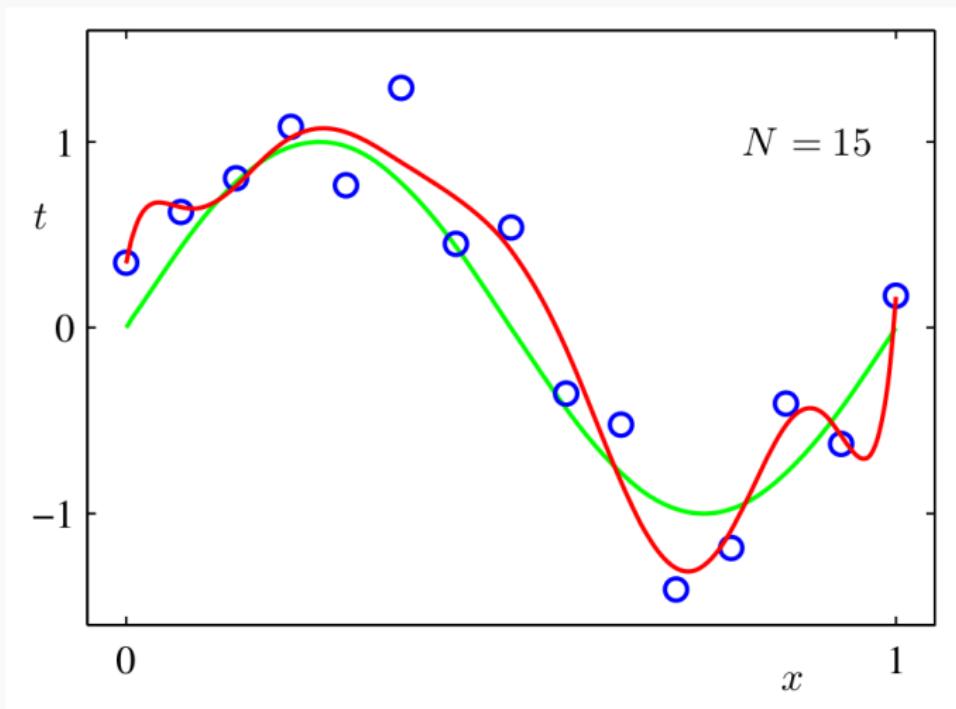
# ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ



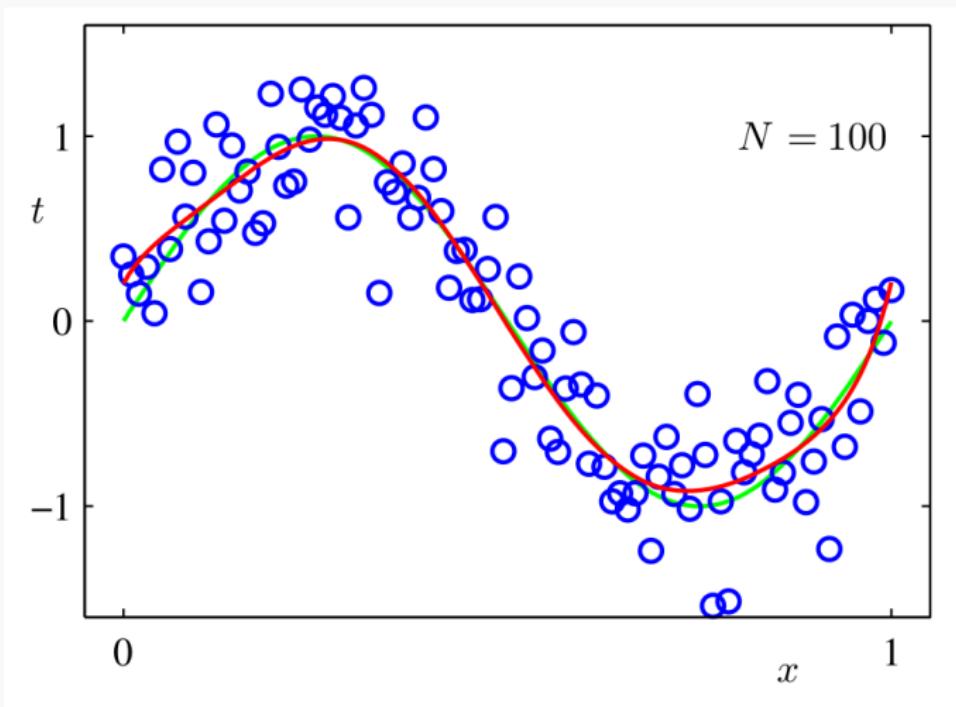
## ЗНАЧЕНИЯ RMS



МОЖНО СОБРАТЬ БОЛЬШЕ ДАННЫХ...



МОЖНО СОБРАТЬ БОЛЬШЕ ДАННЫХ...



## ЗНАЧЕНИЯ КОЭФФИЦИЕНТОВ

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^*$	0.19	0.82	0.31	0.35
$w_1^*$		-1.27	7.99	232.37
$w_2^*$			-25.43	-5321.83
$w_3^*$			17.37	48568.31
$w_4^*$				-231639.30
$w_5^*$				640042.26
$w_6^*$				-1061800.52
$w_7^*$				1042400.18
$w_8^*$				-557682.99
$w_9^*$				125201.43

# ОВЕРФИТТИНГ

- Итак, мы увидели, что даже в линейной регрессии может наступить оверфиттинг.
- Что же делать?..

# РЕГУЛЯРИЗАЦИЯ В ЛИНЕЙНОЙ РЕГРЕССИИ

---

# РЕГУЛЯРИЗАЦИЯ

- Итак, получается, что у нас сильно растут коэффициенты.
- Давайте попробуем с этим бороться. Бороться будем прямолинейно и простодушно: возьмём и добавим размер коэффициентов в функцию ошибки.

# РЕГУЛЯРИЗАЦИЯ

- Было (для тестовых примеров  $\{(x_i, y_i)\}_{i=1}^N$ ):

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2.$$

- Стало:

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \frac{\alpha}{2} \|\mathbf{w}\|^2,$$

где  $\alpha$  – коэффициент регуляризации (его надо будет как-нибудь выбрать).

- Как оптимизировать эту функцию ошибки?

# Регуляризация

- Да точно так же – запишем как

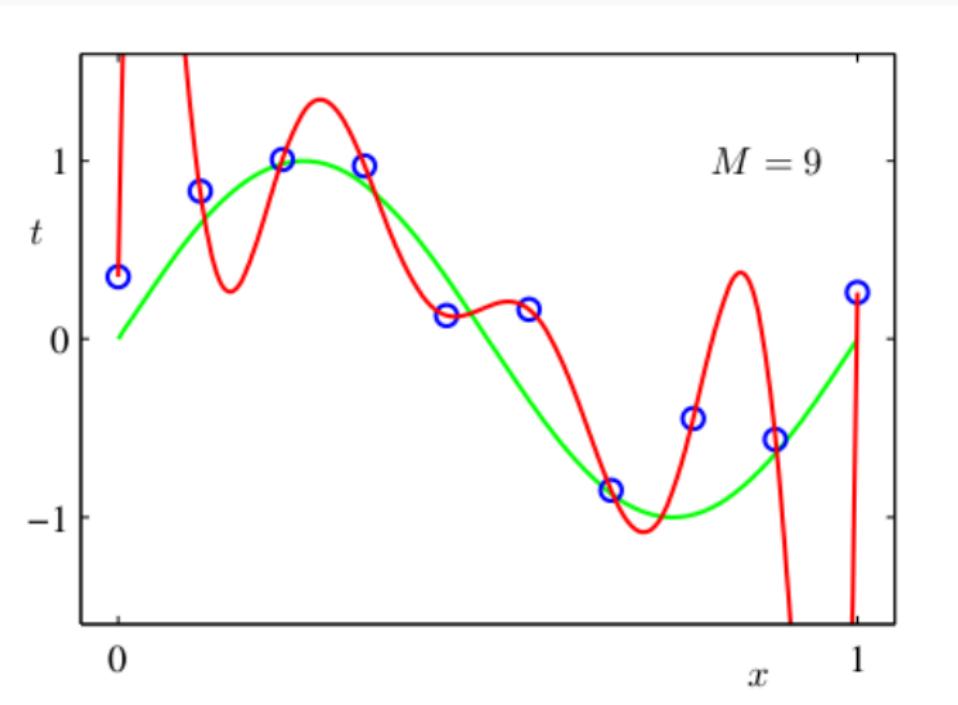
$$\text{RSS}(\mathbf{w}) = \frac{1}{2} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{\alpha}{2} \mathbf{w}^\top \mathbf{w}$$

и возьмём производную; получится

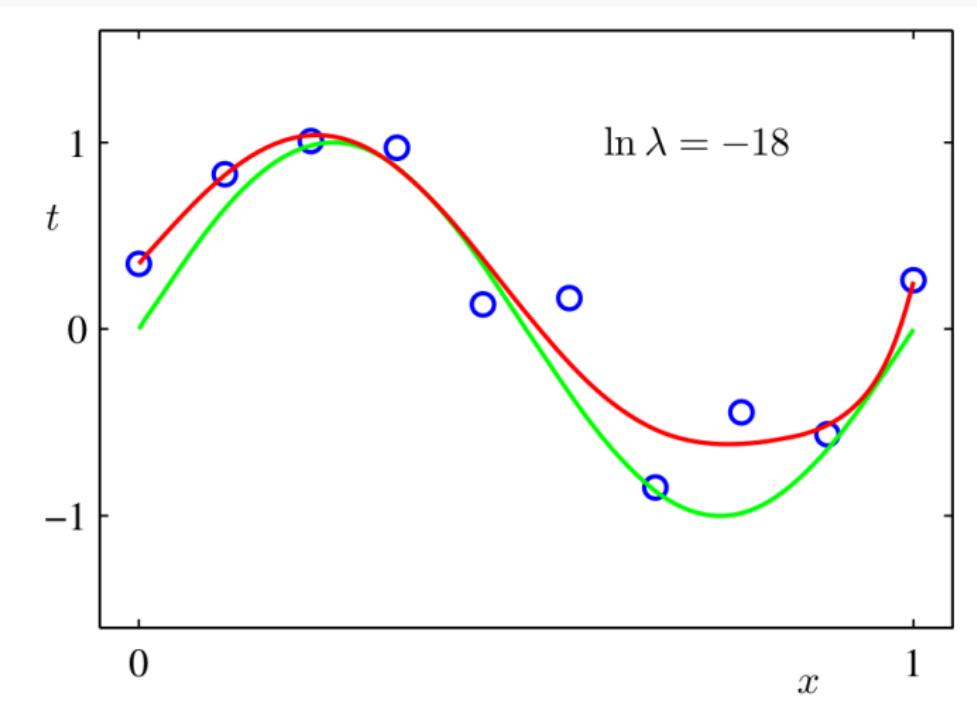
$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \alpha \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}.$$

- Это *гребневая регрессия* (ridge regression); кстати, добавление  $\alpha \mathbf{I}$  к матрице неполного ранга делает её обратимой; это и есть *регуляризация*, и это и было исходной мотивацией для гребневой регрессии.

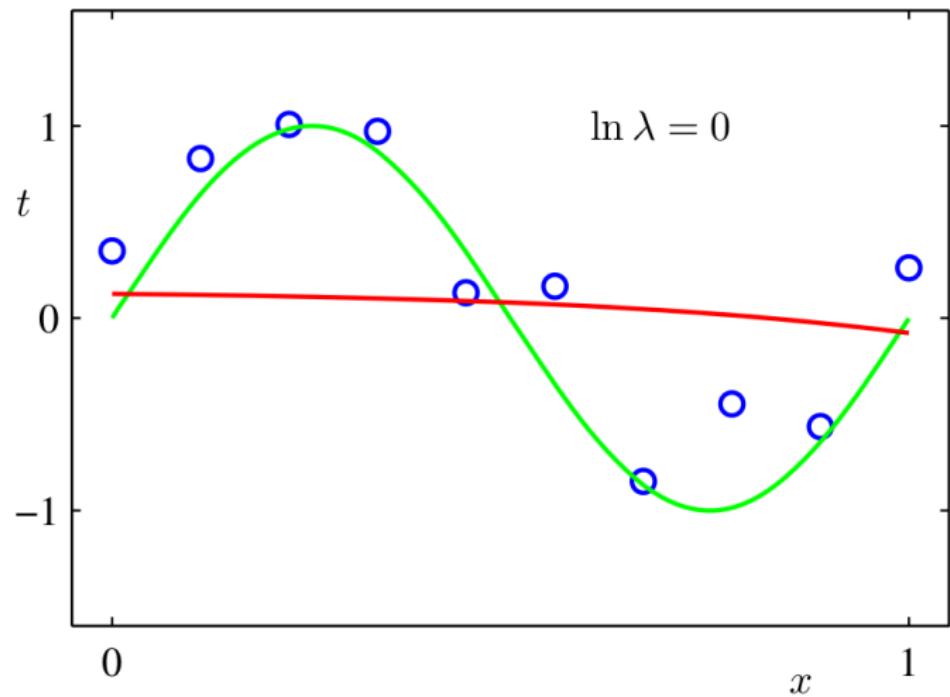
## ГРЕБНЕВАЯ РЕГРЕССИЯ: $\ln \alpha = -\infty$



## ГРЕБНЕВАЯ РЕГРЕССИЯ: $\ln \alpha = -18$



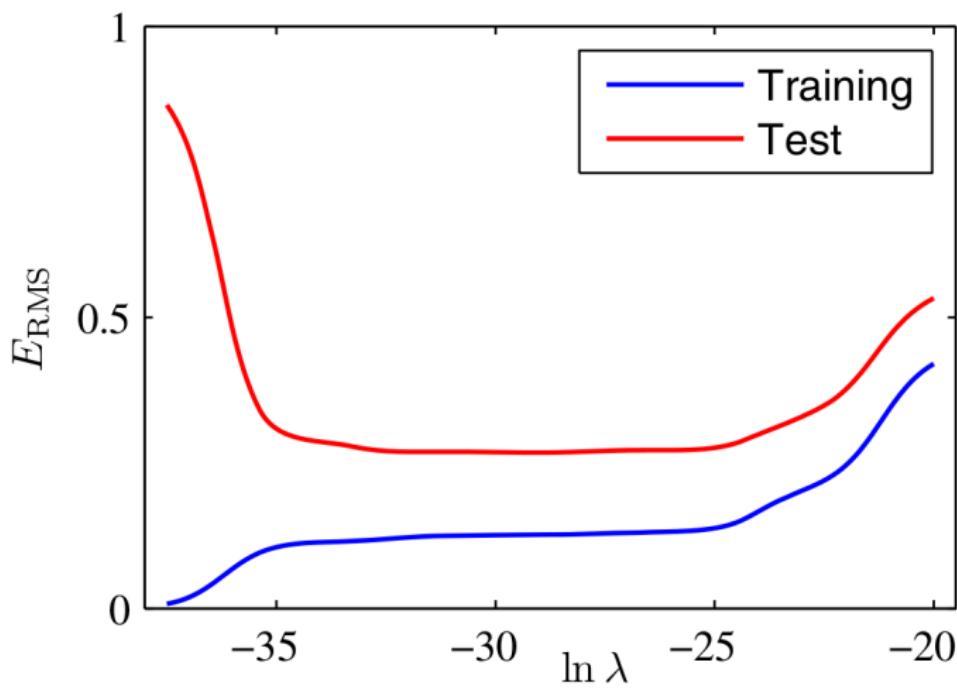
## ГРЕБНЕВАЯ РЕГРЕССИЯ: $\ln \alpha = 0$



## ГРЕБНЕВАЯ РЕГРЕССИЯ: КОЭФФИЦИЕНТЫ

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^*$	0.35	0.35	0.13
$w_1^*$	232.37	4.74	-0.05
$w_2^*$	-5321.83	-0.77	-0.06
$w_3^*$	48568.31	-31.97	-0.05
$w_4^*$	-231639.30	-3.89	-0.03
$w_5^*$	640042.26	55.28	-0.02
$w_6^*$	-1061800.52	41.32	-0.01
$w_7^*$	1042400.18	-45.95	-0.00
$w_8^*$	-557682.99	-91.53	0.00
$w_9^*$	125201.43	72.68	0.01

## ГРЕБНЕВАЯ РЕГРЕССИЯ: RMS



## ДРУГАЯ РЕГУЛЯРИЗАЦИЯ

- Почему именно так? Почему именно  $\frac{\alpha}{2} \|\mathbf{w}\|^2$ ?
- Мы сейчас ответим на этот вопрос, но, вообще говоря, это не обязательно.
- Лассо-регрессия (lasso regression) регуляризует  $L_1$ -нормой, а не  $L_2$ :

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \alpha \sum_{j=0}^M |w_j|.$$

- Есть и другие типы; об этом будем говорить позже.

## РАЗНЫЕ РЕГУЛЯРИЗАТОРЫ

---

## ЕЩЁ РАЗ О РЕГУЛЯРИЗАЦИИ

- Мы знаем, что наименьшие квадраты не всегда хорошо работают. Две причины:
  1. плохая предсказательная сила – часто лучше регуляризовать, пожертвовав bias'ом в пользу variance;
  2. сложности в интерпретации – хотелось бы понимать, что происходит, если переменных с ненулевыми коэффициентами слишком много, не получится.
- Мораль: хотелось бы сделать так, чтобы было поменьше ненулевых компонент в векторе  $w$ .

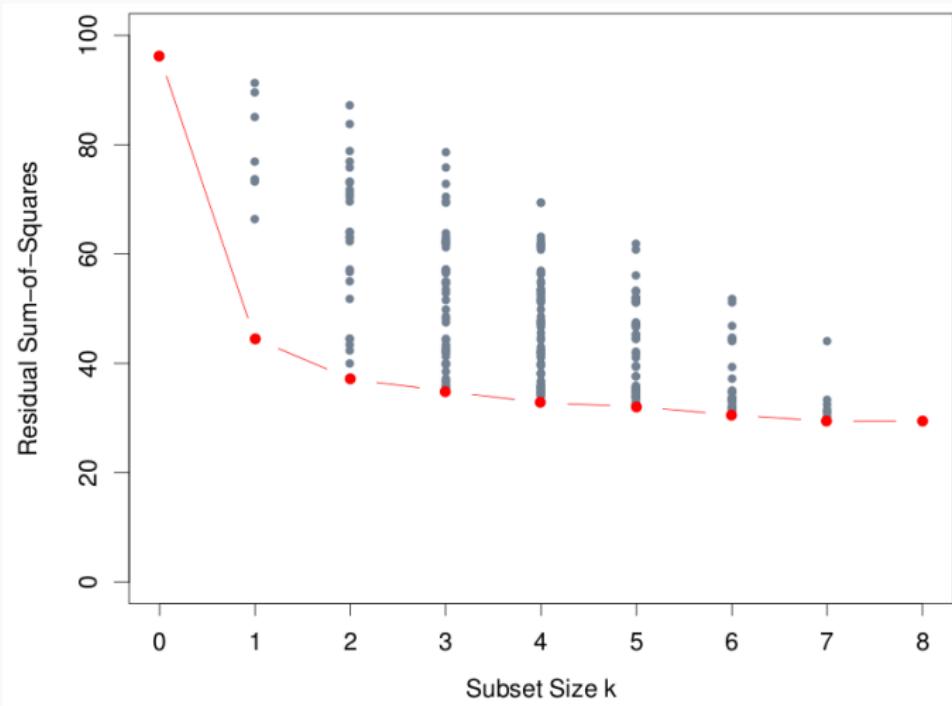
## SUBSET SELECTION

- Может быть, давайте так и сделаем? Будем искать самые лучшие компоненты и делать их ненулевыми.
- Это называется subset selection.
- Можно просто делать best subset selection: выбирать подмножество из  $k$  входных переменных, которые дают самые лучшие результаты.

## SUBSET SELECTION

- Это долго, даже если делать с умом, потому что subsets много.
- Forward-stepwise selection: начинаем со свободного члена, потом добавляет на каждом шаге предиктор, который максимально уменьшает ошибку.
- Т.е. подмножества тут получаются вложенные.
- Backward-stepwise selection: начинаем с полной регрессии и на каждом шаге убираем предиктор, который оказывает меньше всего влияния на ошибку.

## SUBSET SELECTION



## ЛАССО

- Теперь давайте рассмотрим лассо-регрессию:

$$L(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda \sum_{j=0}^p |w_j|.$$

- Главное отличие – теперь форма ограничений (т.е. форма априорного распределения) такова, что весьма вероятно получить строго нулевые  $w_j$ .

- Мы можем переписать регрессию с регуляризатором по-другому:

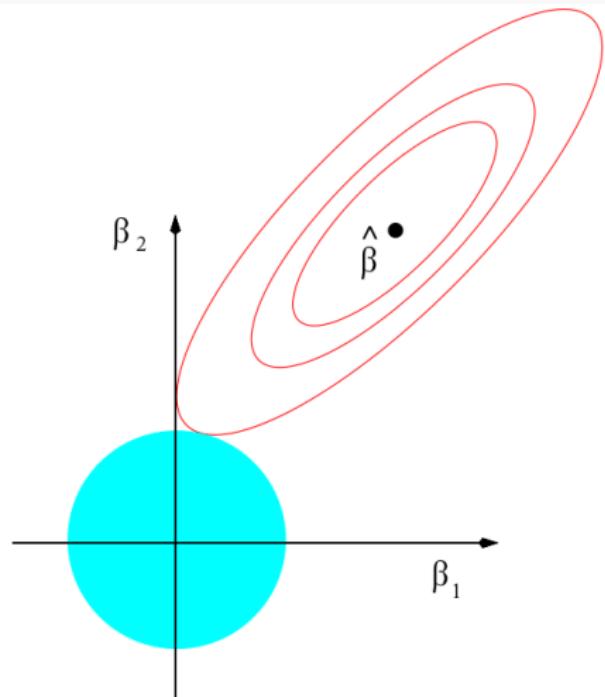
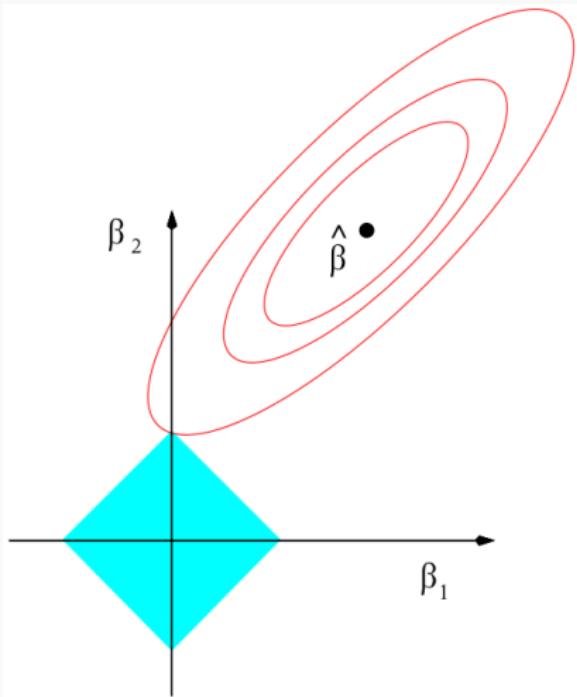
$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda \sum_{j=0}^p |w_j| \right\},$$

эквивалентно

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \left\{ \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 \right\} \text{ при } \sum_{j=0}^p |w_j| \leq t.$$

**Упражнение.** Докажите это.

## ГРЕБЕНЬ И ЛАССО



# ГРЕБЕНЬ И ЛАССО

На данном графике изображены две разные регуляризации для оценки параметров модели (например, линейной регрессии), где:

Красные линии представляют линии уровня функции потерь, зависящей от параметров модели  $\beta_1\beta_1$  и  $\beta_2\beta_2$ . Центр этих линий ( $\beta^*\beta^*$ ) соответствует наилучшей оценке параметров без регуляризации (обычно методом наименьших квадратов).

Голубые области — ограничения, накладываемые регуляризацией, которые сужают пространство возможных значений параметров.

Левый график: L1-регуляризация (Lasso)

Ограничение принимает форму ромба, соответствующего  $|\beta_1|+|\beta_2|\leq\text{constant}$

Углы ромба приводят к разреженным решениям, где многие параметры ( $\beta_1, \beta_2 | \beta_1, \beta_2$ ) обнуляются.

Этот тип регуляризации помогает отбирать только наиболее важные признаки, "заставляя" модель игнорировать остальные.

Правый график: L2-регуляризация (Ridge)

Ограничение представляет собой круг, соответствующий  $\beta_1^2+\beta_2^2\leq\text{constant}$

L2-регуляризация равномерно "сжимает" все параметры, снижая их величины, но не обнуляет их полностью.

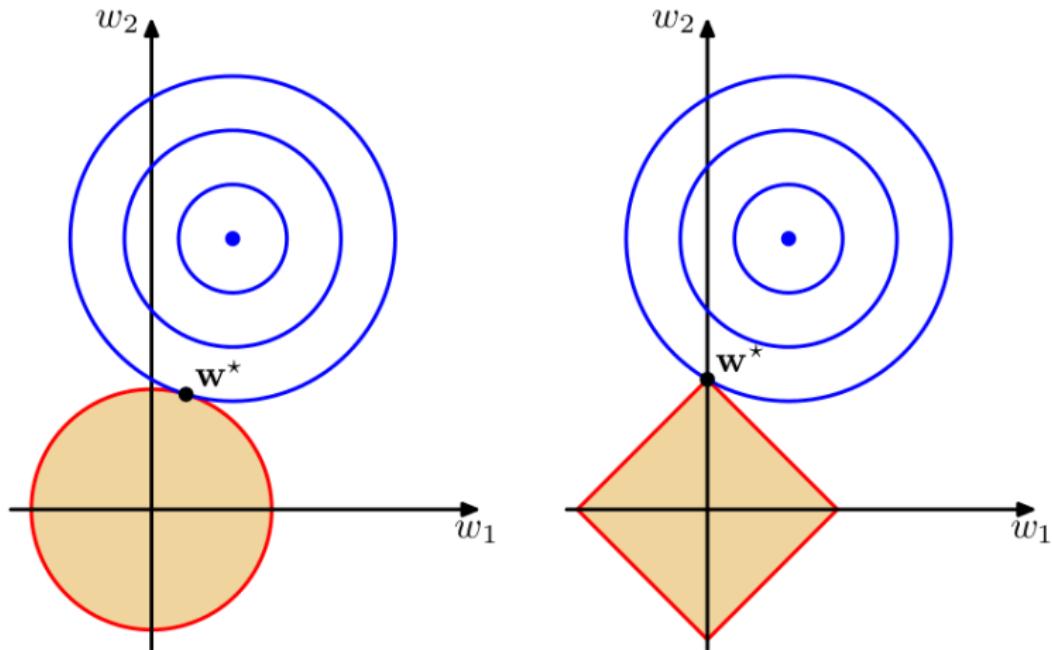
Используется для предотвращения переобучения за счёт уменьшения чувствительности модели к шуму в данных.

Механизм работы

На графиках итоговое значение  $\beta$  находится на пересечении самой близкой к центру линии уровня функции потерь и границы области регуляризации. В зависимости от формы этой границы (ромб или круг), регуляризация по-разному воздействует на параметры.

Это демонстрирует ключевые различия между двумя методами: L1 (Lasso) приводит к более разреженным решениям, а L2 (Ridge) — к сглаживанию параметров.

## ГРЕБЕНЬ И ЛАССО



# ГРЕБЕНЬ И ЛАССО

Этот график иллюстрирует влияние различных типов регуляризации (L2 и L1) на оптимизацию параметров модели машинного обучения.

Синие линии: Представляют собой линии уровня функции потерь (линии уровня функции потерь в машинном обучении визуализируют, как значение функции потерь изменяется в зависимости от параметров модели). Минимум функции потерь отмечен точкой  $w^*w^*$ .

Красные фигуры: Представляют ограничение на параметры модели, которое вводится регуляризацией.

Левая часть: L2-регуляризация (Ridge)

Ограничение показано в виде круга (или эллипса в общем случае). Это происходит потому, что L2-регуляризация добавляет штраф, пропорциональный квадрату нормы весов ( $\|w\|_2^2$ ).

Смысл: L2-регуляризация стремится минимизировать веса модели, равномерно сокращая все компоненты  $w_1w_1$  и  $w_2w_2$ .

Минимум достигается в точке пересечения линий уровня функции потерь с границей круга.

Правая часть: L1-регуляризация (Lasso)

Ограничение показано в виде ромба. Это связано с тем, что L1-регуляризация добавляет штраф, пропорциональный сумме абсолютных значений весов ( $\|w\|_1 + \|w\|_1$ ).

Смысл: L1-регуляризация способствует тому, чтобы многие веса стали нулевыми, создавая разреженные решения. Это видно из углов ромба, которые чаще всего пересекаются с линиями уровня функции потерь.

Основное отличие:

L2-регуляризация (круг): уменьшает веса, но не делает их разреженными (все параметры обычно ненулевые).

L1-регуляризация (ромб): способствует созданию разреженных моделей, где многие веса равны нулю, что полезно для отбора признаков.

# Обобщение

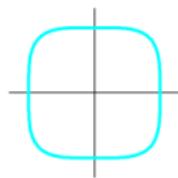
- Можно рассмотреть обобщение гребневой и лассо-регрессии:

$$L(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda \sum_{j=0}^p (|w_j|)^q.$$

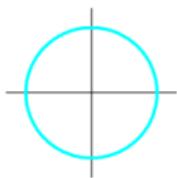
**Упражнение.** Какому априорному распределению на параметры  $\mathbf{w}$  соответствует эта задача?

# РАЗНЫЕ $q$

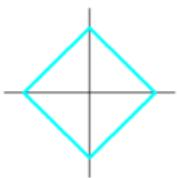
$q = 4$



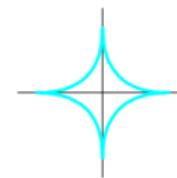
$q = 2$



$q = 1$



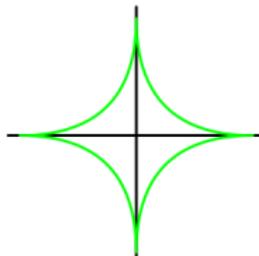
$q = 0.5$



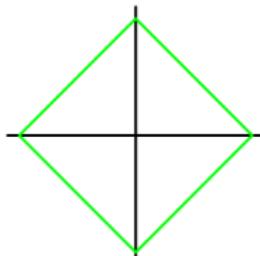
$q = 0.1$



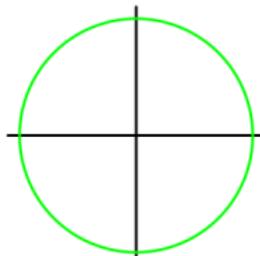
$q = 0.5$



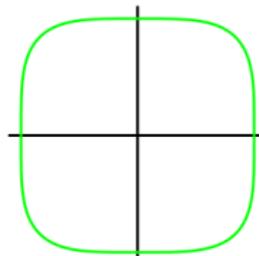
$q = 1$



$q = 2$



$q = 4$



# 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:

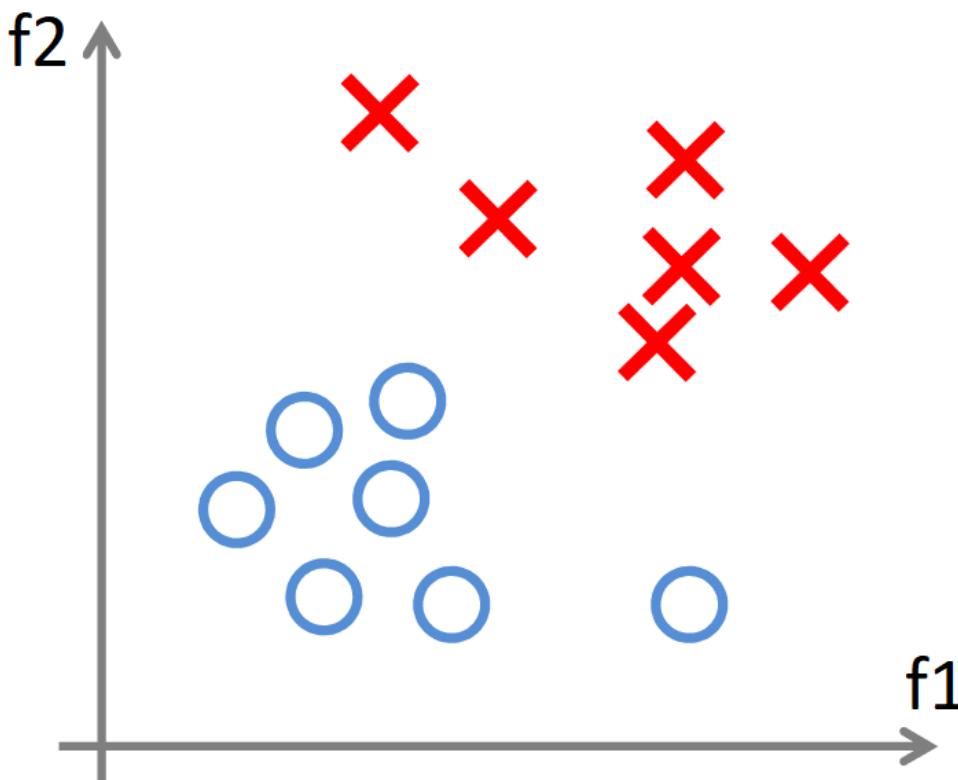
$$a(x, w) = \sum_{i=1}^l w_j x_j$$

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

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

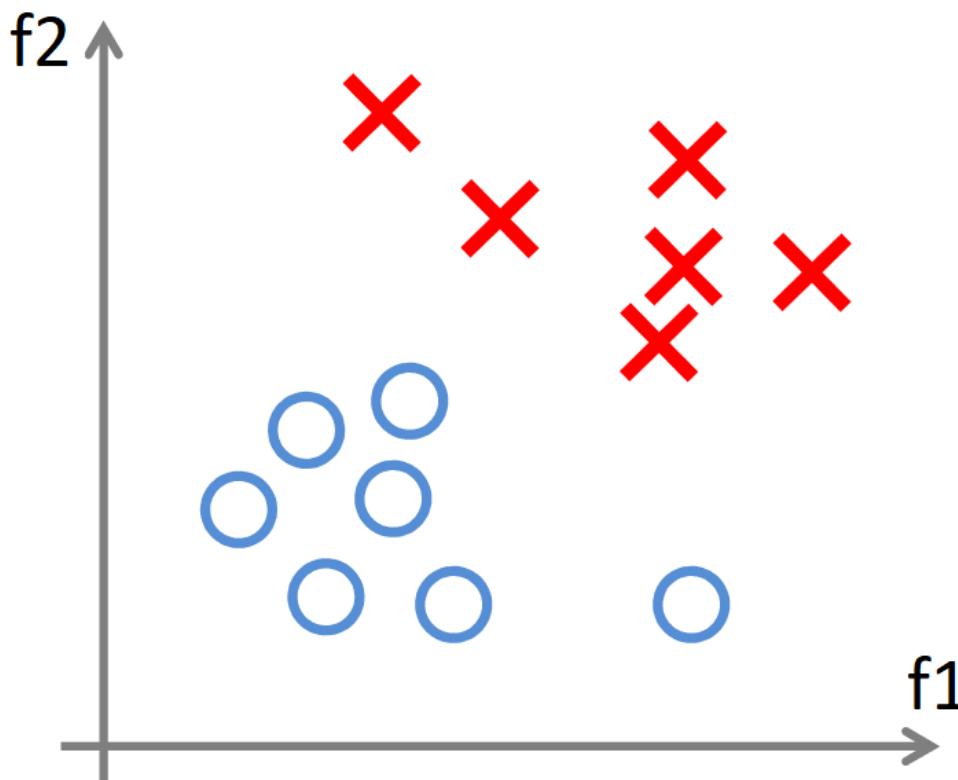
# БИНАРНАЯ КЛАССИФИКАЦИЯ

$y_1, y_2, \dots, y_n$  - 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:

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

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

$\text{sign}(x) = \{1; x > 0; 0; x = 0; -1; x < 0\}$  - (сигнум, от лат. signum — знак) — кусочно-постоянная функция вещественного аргумента

# БИНАРНАЯ КЛАССИФИКАЦИЯ

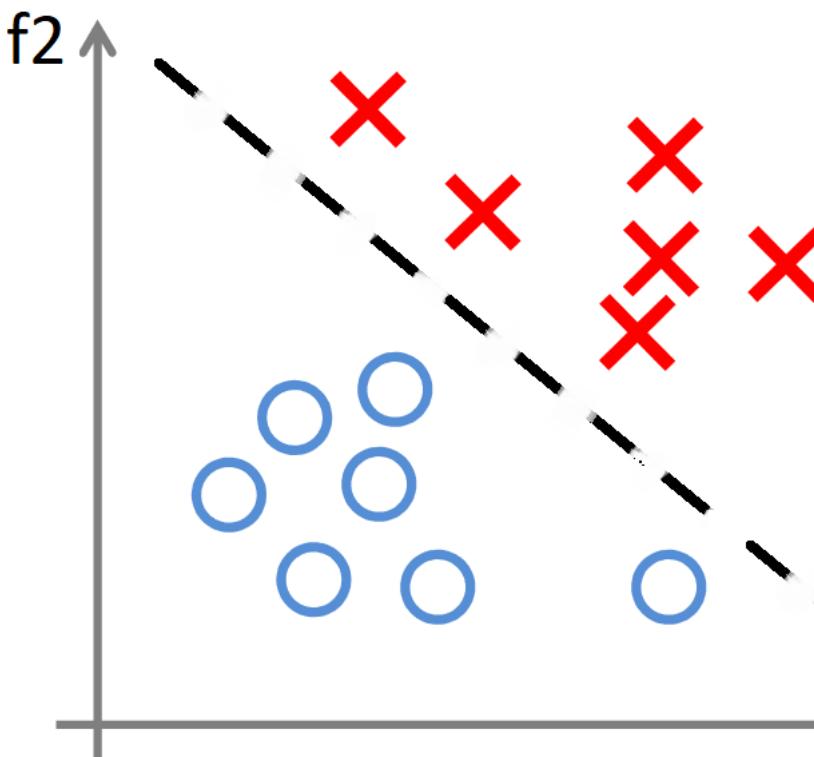
Модель линейного классификатора:

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

Уравнение

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

– уравнение плоскости  
(или прямой).



# ОБУЧЕНИЕ КЛАССИФИКАТОРА

- Обучение - минимизация доли ошибок классификатора:

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

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

- Обозначим  $M_i = y_i \cdot (w, \mathbf{x}_i)$  - *отступ на i-м объекте*.

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

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

## 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:

Случай 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:

Случай 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:

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

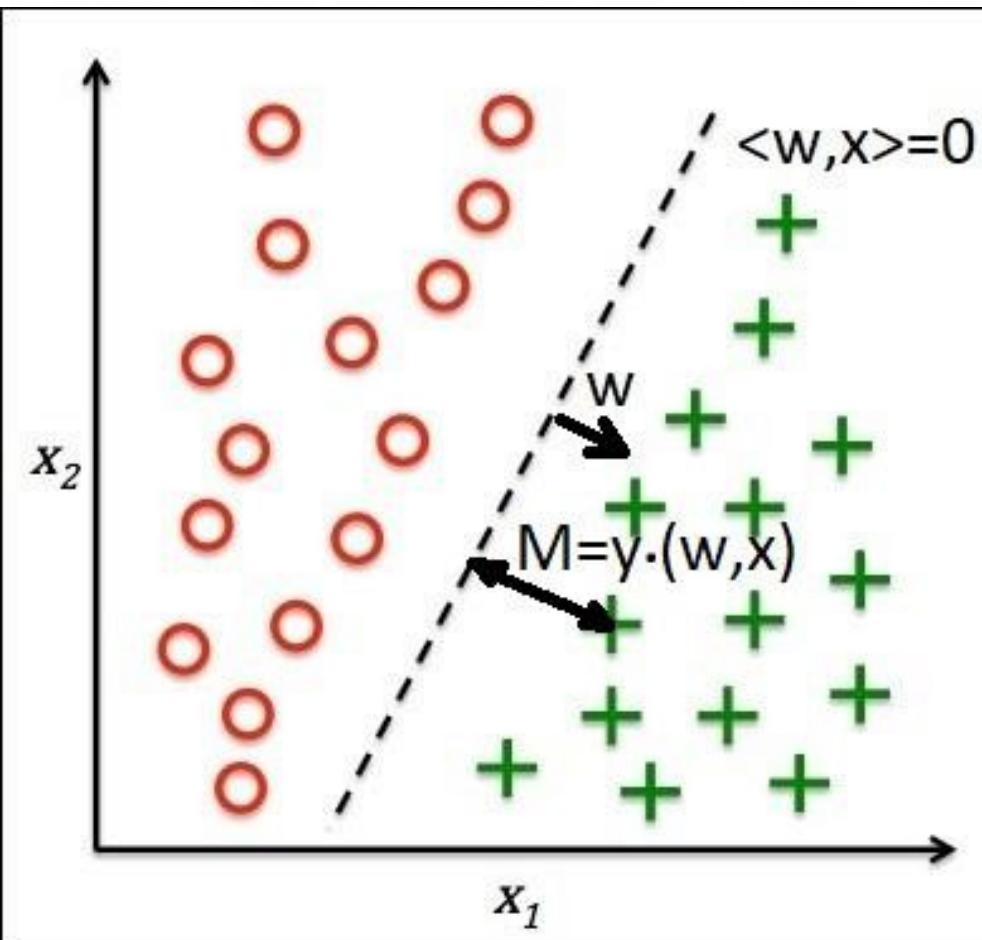
- Если  $(w, x) > 0$  (то есть объект отнесён к классу +1), а  $y = -1$ , то  $M = y \cdot (w, x) < 0$ .
- Аналогично, если  $(w, x) < 0$ , а  $y = +1$ , то  $M = y \cdot (w, x) < 0$ .

Случай 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)

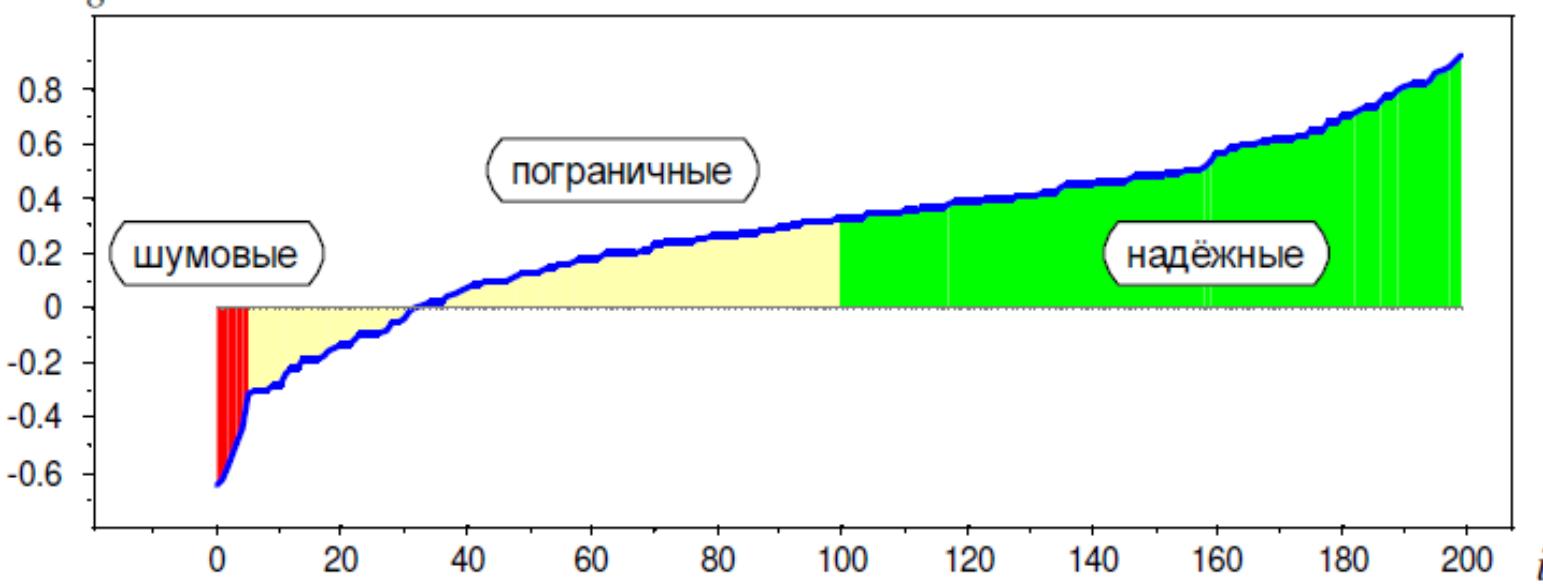
Абсолютная величина отступа  $M$  обозначает степень уверенности классификатора в ответе (чем ближе  $M$  к нулю, тем меньше уверенность в ответе)



# ОТСТУП (MARGIN)

Ранжирование объектов по возрастанию отступа:

*Margin*

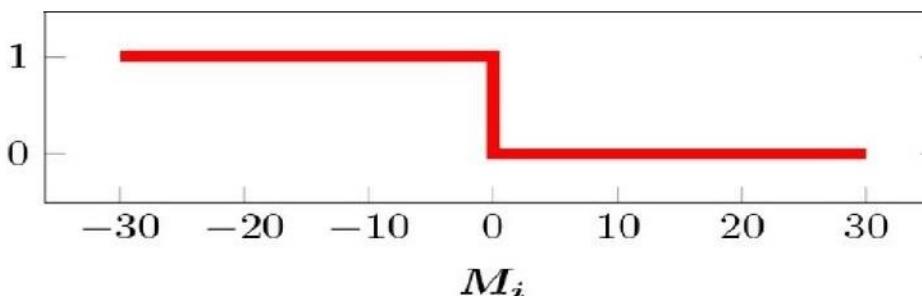


# ПОРОГОВАЯ ФУНКЦИЯ ПОТЕРЬ

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

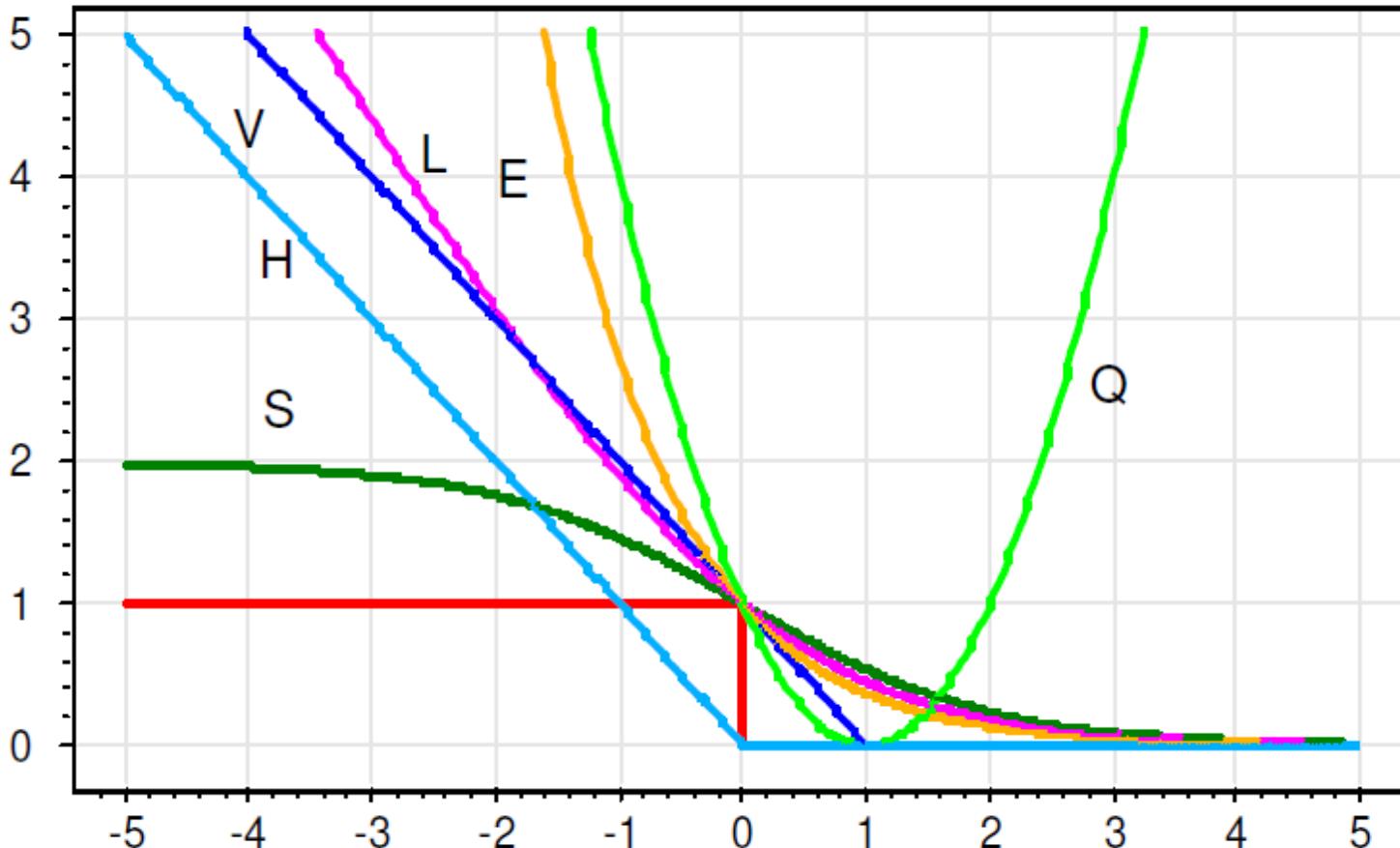
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^n L(y_i \cdot (w, x_i)) \leq \frac{1}{n} \sum_{i=1}^n \tilde{L}(y_i \cdot (w, x_i)) \rightarrow \min$$

# 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:

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

# МЕТРИКИ КАЧЕСТВА

# МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ

- Accuracy – доля правильных ответов:

$$\text{accuracy}(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) = y_i]$$

# МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ

- Accuracy – доля правильных ответов:

$$\text{accuracy}(a, X) = \frac{1}{n} \sum_{i=1}^n [a(x_i) = y_i]$$

*Недостаток: при сильно несбалансированной выборке не отражает качество работы алгоритма*

# ПРИМЕР: КРЕДИТНЫЙ СКОРИНГ

- Пример: Кредитный scoring
- Модель 1: одобряет 100 кредитов
  - 80 кредитов вернули
  - 20 кредитов не вернули
- Модель 2: одобряет 50 кредитов
  - 48 кредитов вернули
  - 2 кредита не вернули
- Какая лучше?

На выборке,  
где 100 вернули,  
100 не вернули

# МАТРИЦА ОШИБОК

Матрица ошибок (confusion matrix):

		Actual Value	
		positives	negatives
Predicted Value	positives	<b>TP</b> True Positive	<b>FP</b> False Positive
	negatives	<b>FN</b> False Negative	<b>TN</b> True Negative

# МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ: PRECISION, RECALL

- Precision (точность):

$$Precision(a, X) = \frac{TP}{TP + FP}$$

Показывает, насколько можно доверять классификатору при  $a(x) = +1$ .

# PRECISION: ПРИМЕР

Модель  $a_1(x)$ :

$$\text{precision}(a_1, X) = 0.8$$

		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20	
	20	80	
$a(x) = -1$ Не получили кредит			

Модель  $a_2(x)$ :

$$\text{precision}(a_2, X) = 0.96$$

		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	48	2	
	52	98	
$a(x) = -1$ Не получили кредит			

# МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ: PRECISION, RECALL

- Precision (точность):

$$Precision(a, X) = \frac{TP}{TP + FP}$$

Показывает, насколько можно доверять классификатору при  $a(x) = +1$ .

- Recall (полнота):

$$Recall(a, X) = \frac{TP}{TP + FN}$$

Показывает, как много объектов положительного класса находит классификатор.

# RECALL: ПРИМЕР

Модель  $a_1(x)$ :

$$\text{recall}(a_1, X) = 0.8$$

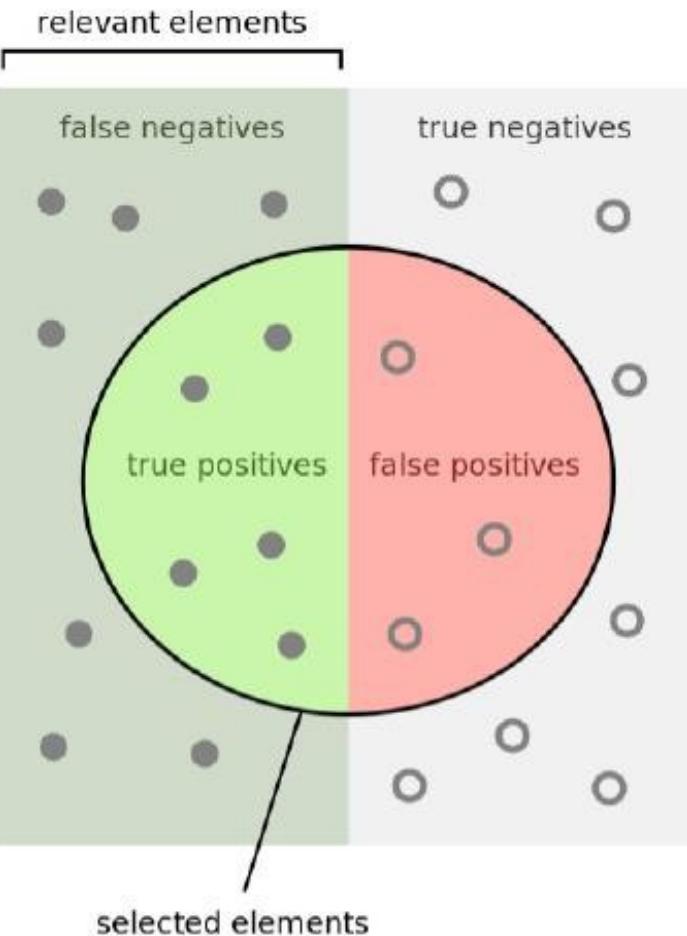
	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20
$a(x) = -1$ Не получили кредит	20	80

Модель  $a_2(x)$ :

$$\text{recall}(a_2, X) = 0.48$$

	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	48	2
$a(x) = -1$ Не получили кредит	52	98

# ТОЧНОСТЬ И ПОЛНОТА



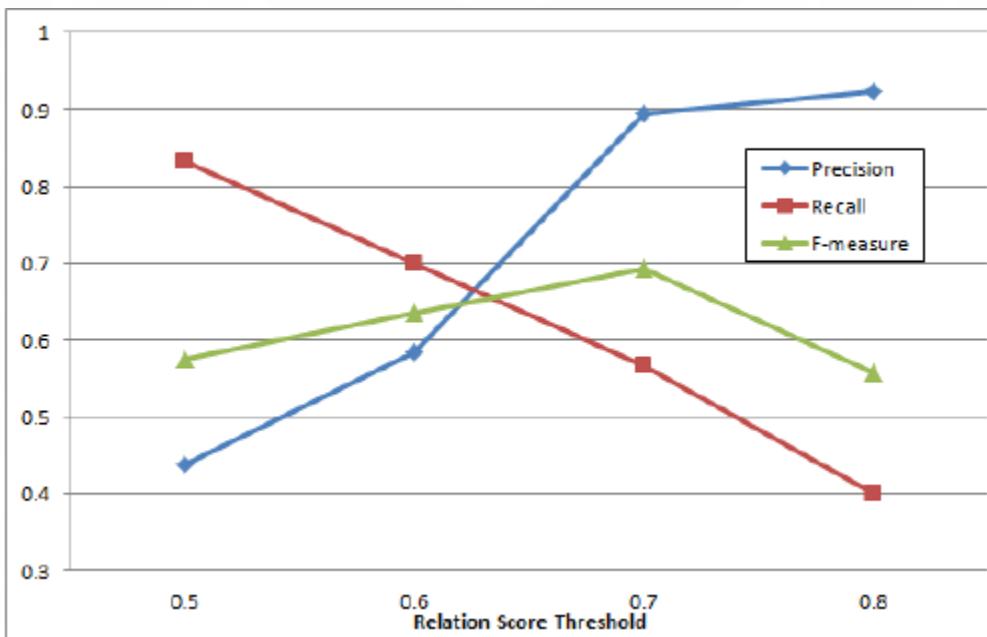
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

# F-МЕРА

F-мера – это метрика качества, учитывающая и точность, и полноту

$$F(a, X) = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$



# ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но *классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.*

# ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.
- Пусть  $p(x)$  – вероятность того, что объект  $x$  принадлежит классу +1. Тогда *если  $p \geq 0.5$ , классификатор относит  $x$  к положительному классу, а иначе – к отрицательному.*



# ТОЧНОСТЬ И ПОЛНОТА

- Для вычисления точности и полноты мы сравниваем на каждом объекте предсказанный класс с истинным классом. Но классификаторы, как правило, кроме классов умеют предсказывать вероятность принадлежности классу.
- Пусть  $p(x)$  – вероятность того, что объект  $x$  принадлежит классу +1. Тогда если  $p \geq 0.5$ , классификатор относит  $x$  к положительному классу, а иначе – к отрицательному.



- Но *мы можем изменять порог вероятности!* Например, относить к положительному классу те объекты, для которых  $p \geq 0.7$ , а остальные объекты – к отрицательному классу. *Тем самым мы будем изменять точность и полноту.*

