

# Лекция 5

## Линейные модели классификации. Часть 2

Юлия Конюшенко

тг: @ko\_iulia

[koniushenko.iun@phystech.edu](mailto:koniushenko.iun@phystech.edu)

## Лекция 1.

ML

→ обучение с учителем

обучение без учителя

- категоризация
- снижение размерн.
- визуализация

- классифр.
- регрессия
- ранжирование

## Лекция 2.

1) линейная регрессия

Обучение  $\equiv$  минимизация MSE

почему именно MSE? → есть вероятностная интерпретация

2) градиентного спуск

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

$$MSE$$

$$w^{(k)} = w^{(k-1)} - \eta \triangleright Q(w^{(k-1)})$$

стochastic  $\rightarrow$  по 1 объекту  
mini-batch  $\rightarrow$  по батчу

## Лекция 3. Метрики качества и функционалы

$$MSE \rightarrow RMSE \rightarrow R^2$$

ошибки

$$MAE \rightarrow MSLE \rightarrow MAPE \rightarrow SMAPE$$

квантильная регрессия

онлайн / офлайн / бизнес метрики

Признаки переобучения, регуляризация

- разница качества на train/test
- большие веса

+ оптимумы MSE и

MAE

среднее медиана

$$\begin{aligned} \cdot L_2: & + \sum w_i^2 \\ \cdot L_1: & + \sum |w_i| \end{aligned}$$

## Лекция 4.

### 1) Оценивание качества модели

- отложенная выборка
- кросс-валидация

K-fold

complete

leave-one-out

K = 5

K = 7

K = 10

### 2) Способы кодирования категориальных признаков

- one-hot encoding

- стеммы
  - сжатие
  - подсет на отложенной выборке

### 3) Линейные модели классификации

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

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

Для оптимизации используют верхние оценки эмпирического риска  
 Разные ф-ии потерь соответствуют различным типам моделей

Оптимизируются градиентным спуском

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

вероятности несут больше информации

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия:  $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия:  $\sigma(w^T x)$ ,

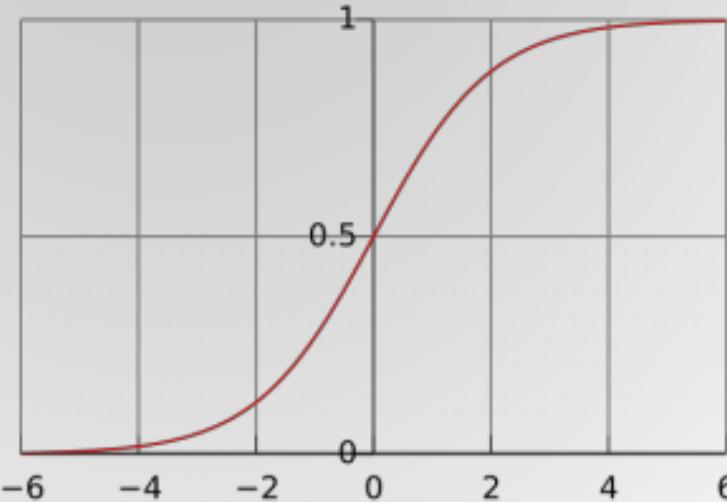
где  $\sigma(z) = \frac{1}{1+e^{-z}}$  - сигмоида (логистическая функция)

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

- Линейная регрессия:  $a(x, w) = (x, w) = w^T x \in \mathbb{R}$
- Логистическая регрессия:  $a(x, w) = \sigma(w^T x)$ ,

где  $\sigma(z) = \frac{1}{1+e^{-z}}$  - сигмоида (логистическая функция),  
 $\sigma(z) \in (0; 1)$ .



Логистическая регрессия:  $a(x, w) = \frac{1}{1+e^{-w^T x}}$

Это линейная модель? Какая здесь разделяющая поверхность?

# РАЗДЕЛЯЮЩАЯ ГРАНИЦА

Предсказываем  $y = +1$ , если  $a(x, w) \geq 0.5$ .



1.  $w^T x$
2.  $\sigma(w^T x)$
- 3.

$$a(x, w) = \sigma(w^T x) \geq 0.5, \text{ если } w^T x \geq 0.$$

Получаем, что

- $y = +1$  при  $w^T x \geq 0$
- $y = -1$  при  $w^T x < 0$ ,

т.е.  $w^T x = 0$  – разделяющая гиперплоскость.

# ВЕРОЯТНОСТНЫЙ СМЫСЛ

**Утверждение.**  $a(x, w)$  – вероятность того, что  $y = +1$  на объекте  $x$ , т.е.

$$a(x, w) = P(y = +1|x; w)$$

**Доказательство.** Дальше в лекции.

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Логистическая регрессия - это линейный классификатор!

# ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Если взять квадратичную функцию потерь

$$L(a, y) = (a - y)^2,$$

то возникнут проблемы:

- $Q(a, X) = \frac{1}{l} \sum_{i=1}^l \left( \frac{1}{1+e^{-w^T x_i}} - y_i \right)^2$  - не выпуклая функция  
(можем не попасть в глобальный минимум при оптимизации)
- На совсем неправильном предсказании маленький штраф  
(пусть предсказали вероятность 0% на объекте класса  $y = +1$ , тогда штраф всего  $(1 - 0)^2 = 1$ )

$$\begin{array}{lll} y = +1 & a = 1 & (1 - 1)^2 = 0 \\ y = +1 & a = 0 & (1 - 0) = 1 \end{array}$$

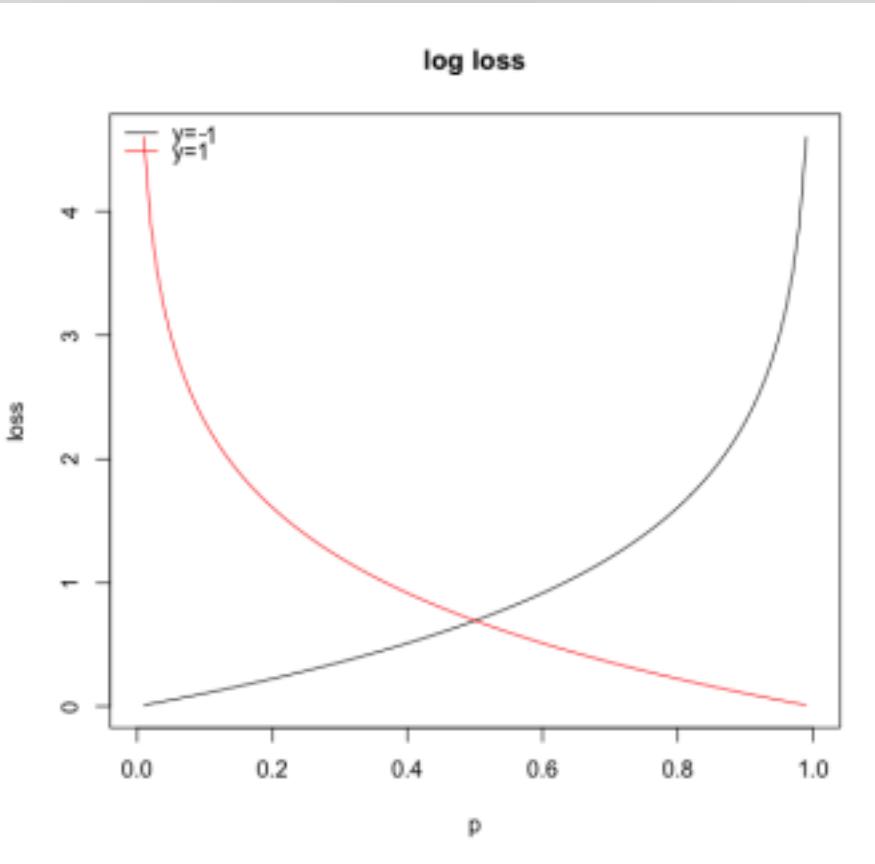
больше одного  
минимума

# ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

Возьмем логистическую функцию потерь (log-loss):

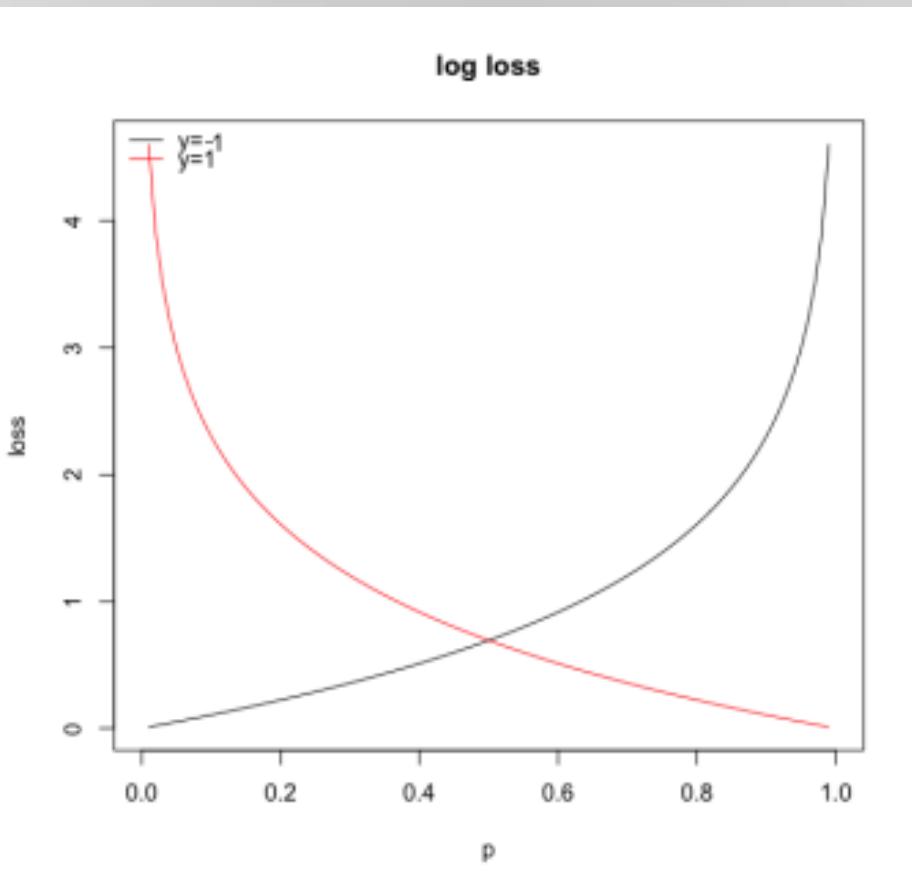
$$Q(w) = - \sum_{i=1}^l ([y_i = +1] \cdot \log(a(x_i, w)) + [y_i = -1] \cdot \log(1 - a(x_i, w)))$$

*индикаторы*  $\rho(+1|x_i)$



*log*  
погану  
здесь *log*,  
уходит?

# ЛОГИСТИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ



- если  $a(x, w) = 1$  и  $y = +1$ , то штраф  $L(a, y) = 0$
- если  $a(x, w) \rightarrow 0$ , а  $y = +1$ , то штраф  $L(a, y) \rightarrow +\infty$

$$Q(w, x) = Q_1 + Q_2 + \dots + Q_n$$

$$1) Q = (x, w) \rightarrow R$$

Ket.  $\downarrow$



$$2) Q = \text{sign}(x, w)$$

3)



$$4) Q = \tilde{\sigma}(x, w)$$

$$5) Q(w, x) - ?$$

MSE - ?

оптимизация?  
больше 1 минимум  
маленькие шаги



ноги ноги

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ: ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

**Предположение:** В каждой точке  $x$  пространства объектов задана вероятность  $p(y = +1|x)$

*Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.*

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

**Предположение:** В каждой точке  $x$  пространства объектов задана вероятность  $p(y = +1|x)$

*Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.*

**Цель:** построить алгоритм  $b(x)$ , в каждой точке  $x$  предсказывающий  $p(y = +1|x)$ .

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

**Предположение:** В каждой точке  $x$  пространства объектов задана вероятность  $p(y = +1|x)$

*Объекты с одинаковым признаковым описанием могут иметь разные значения целевой переменной.*

**Цель:** построить алгоритм  $b(x)$ , в каждой точке  $x$  предсказывающий  $p(y = +1|x)$ .

**Комментарий:** пока что мы будем решать задачу в общем виде, то есть у нас нет ограничений на вид алгоритма  $b(x)$  и на вид функции потерь  $L(y, b)$ .

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект  $x$  встречается в выборке  $n$  раз с ответами  $\{y_1, \dots, y_n\}$ . Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект  $x$  встречается в выборке  $n$  раз с ответами  $\{y_1, \dots, y_n\}$ . Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при  $n \rightarrow \infty$  получаем

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x]$$

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА ЗАДАЧИ

- Пусть объект  $x$  встречается в выборке  $n$  раз с ответами  $\{y_1, \dots, y_n\}$ . Хотим, чтобы алгоритм выдавал вероятность положительного класса:

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1|x)$$

По закону больших чисел при  $n \rightarrow \infty$  получаем

$$b_*(x) = \operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x]$$

Отсюда получаем **условие на функцию потерь**:

$$\operatorname{argmin}_{b \in \mathbb{R}} E[L(y, b)|x] = p(y = +1|x)$$

✓ *переписали  
“цель” в  
терминах  
вероятности*

# ФУНКЦИИ ПОТЕРЬ

Подходят:

Квадратичная

$$L(y, z) = (y - z)^2$$

→ предсказывают вероятность

- Логистическая (log-loss)

$$L(y, z) = [y = +1] \cdot \log(b(x, w)) + [y = -1] \cdot \log(1 - b(x, w))$$

Не подходят:

- Модуль

$$L(y, z) = |y - z|$$

почему log-loss? Почему не что-то иное?

# ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм  $b(x)$ , должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект  $x$  с классом  $y$ :

$$b(x)^{[y=+1]} \cdot (1 - b(x))^{[y=-1]}$$

# ПРАВДОПОДОБИЕ И LOG-LOSS

- Вероятности, которые выдает алгоритм  $b(x)$ , должны согласовываться с выборкой
- Вероятность того, что в выборке встретится объект  $x$  с классом  $y$ :

$$b(x)^{[y=+1]} \cdot (1 - b(x))^{[y=-1]}$$

Правдоподобие выборки:

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]}$$

# ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

# ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

# ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это **log-loss!**

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

# ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

- Для нахождения оптимальных параметров алгоритма можно воспользоваться методом максимума правдоподобия (ММП):

$$(b, X) = \prod_{i=1}^l b(x_i)^{[y_i=+1]} \cdot (1 - b(x_i))^{[y_i=-1]} \rightarrow \max_b$$

- Прологарифмируем правдоподобие и поставим перед ним минус, получим следующую эквивалентную задачу:

Это **log-loss!**

$$-\sum_{i=1}^l ([y_i = +1] \log b(x_i) + [y_i = -1] \log(1 - b(x_i))) \rightarrow \min_b$$

**Вывод:** логистическая функция потерь корректно предсказывает вероятности.

# ВЫБОР АЛГОРИТМА $b(x)$

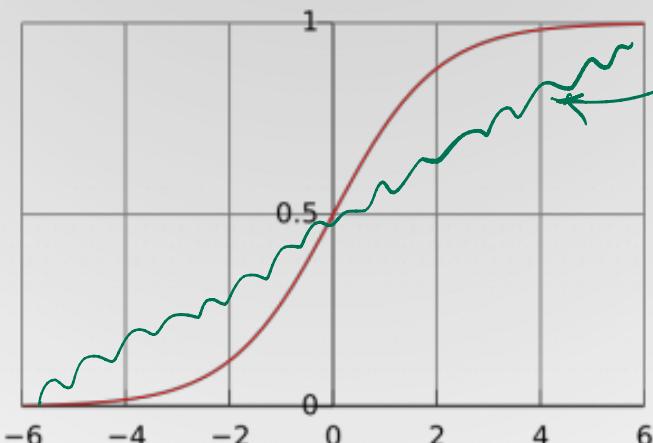
- Хотим, чтобы алгоритм  $b(x)$  возвращал числа из отрезка  $[0, 1]$ .

# ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм  $b(x)$  возвращал числа из отрезка  $[0, 1]$ .
- Можно взять  $b(x) = \sigma(w^T x)$ , где  $\sigma$  – любая монотонно неубывающая функция с областью значений  $[0, 1]$ .

# ВЫБОР АЛГОРИТМА $b(x)$

- Хотим, чтобы алгоритм  $b(x)$  возвращал числа из отрезка  $[0, 1]$ .
- Можно взять  $b(x) = \sigma(w^T x)$ , где  $\sigma$  – любая монотонно неубывающая функция с областью значений  $[0, 1]$ .
- Возьмем **сигмоиду**:  $\sigma(z) = \frac{1}{1+e^{-z}}$



а такую можно?

# СМЫСЛ $(w, x)$ В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке  $x$  предсказывает вероятность того, что  $x$  принадлежит положительному классу  $p(y = +1|x)$ .
- То есть  $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$ . Отсюда можно выразить  $(w, x) = w^T x$ :

$$(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}$$

— логарифм  
отношения  
шансов

# СМЫСЛ $(w, x)$ В ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- Логистическая регрессия в каждой точке  $x$  предсказывает вероятность того, что  $x$  принадлежит положительному классу  $p(y = +1|x)$ .
- То есть  $p(y = +1|x) = \frac{1}{1+e^{-w^T x}}$ . Отсюда можно выразить  $(w, x) = w^T x$ :

$$(w, x) = w^T x = \log \frac{p(y = +1|x)}{p(y = -1|x)}$$

- Величина  $\log \frac{p(y=+1|x)}{p(y=-1|x)}$  называется **логарифм отношения шансов (log odds)**. Из формулы видно, что величина может принимать любое значение.

# ЛОГАРИФМИЧЕСКАЯ ФУНКЦИЯ ПОТЕРЬ

**Утверждение.** Логарифмическая функция потерь может быть записана в виде

$$L(b, X) = \sum_{i=1}^l \log(1 + e^{-y_i(w, x)})$$

**Идея доказательства:**

Подставляем явный вид сигмоиды в логарифмическую функцию потерь:

$$-\sum_{i=1}^l ([y_i = +1] \log \sigma(w^T x_i) + [y_i = -1] \log(1 - \sigma(w^T x_i))) \rightarrow \min_w$$

# ПЕРСЕПТРОН РОЗЕНБЛATTA

*Персепtron* – это простейшая модель классификации, при этом являющаяся предшественником нейронных сетей.

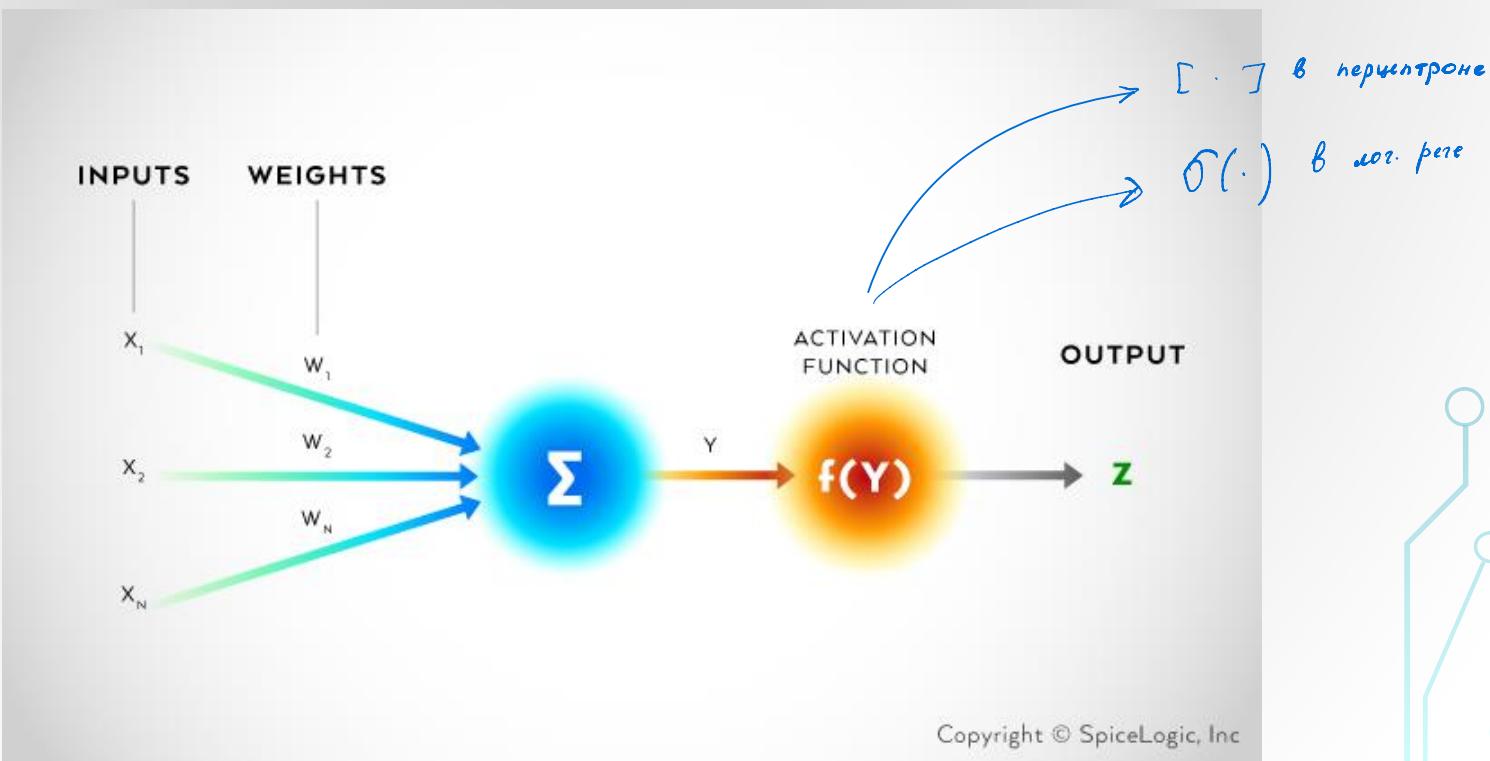
- Задача классификации с двумя классами  $y_i \in \{0,1\}$
- Признаки объектов – бинарные:  $x_i^j \in \{0,1\}$

Алгоритм:  $a(x, w) = [w_1 x_1 + \dots + w_n x_n > 0] = [(w, x) > 0]$

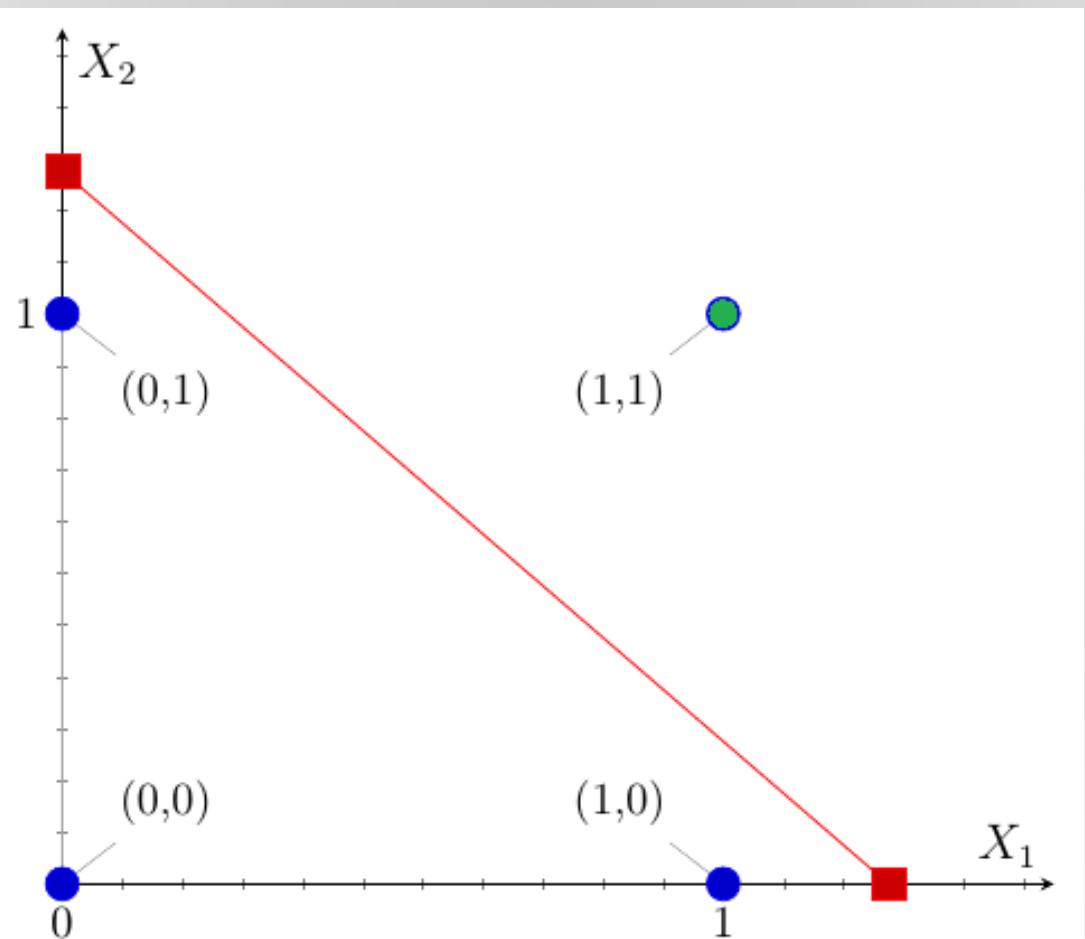
# ПЕРСЕПТРОН РОЗЕНБЛATTA

- Задача классификации с двумя классами  $y_i \in \{0,1\}$
- Признаки объектов – бинарные:  $x_i^j \in \{0,1\}$

Алгоритм:  $a(x, w) = [w_1x_1 + \dots + w_n x_n > 0] = [(w, x) > 0]$



# ПРИМЕР: РЕАЛИЗАЦИЯ ЛОГИЧЕСКОГО AND С ПОМОЩЬЮ ПЕРСЕПТРОНА

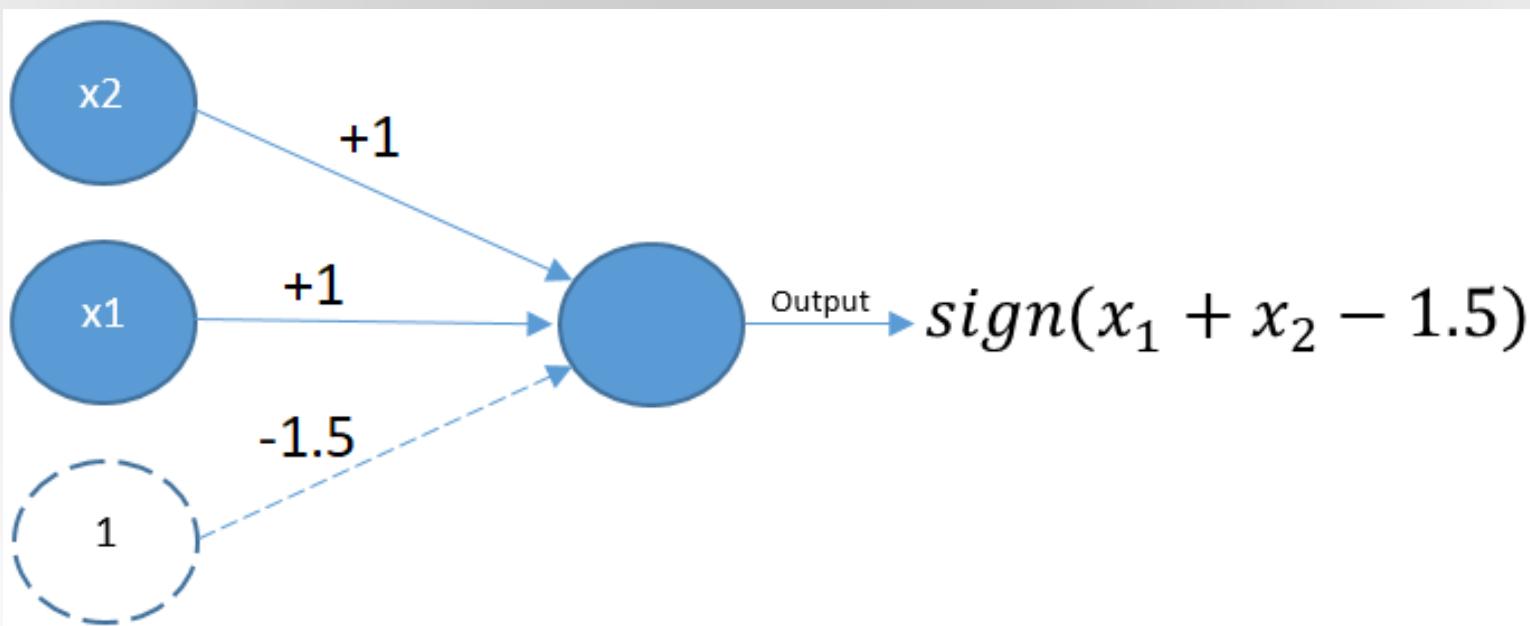


логическое  
 $X_1 \cdot X_2 = 1$   
 $X_1 \cdot X_2 = 0$  шаре

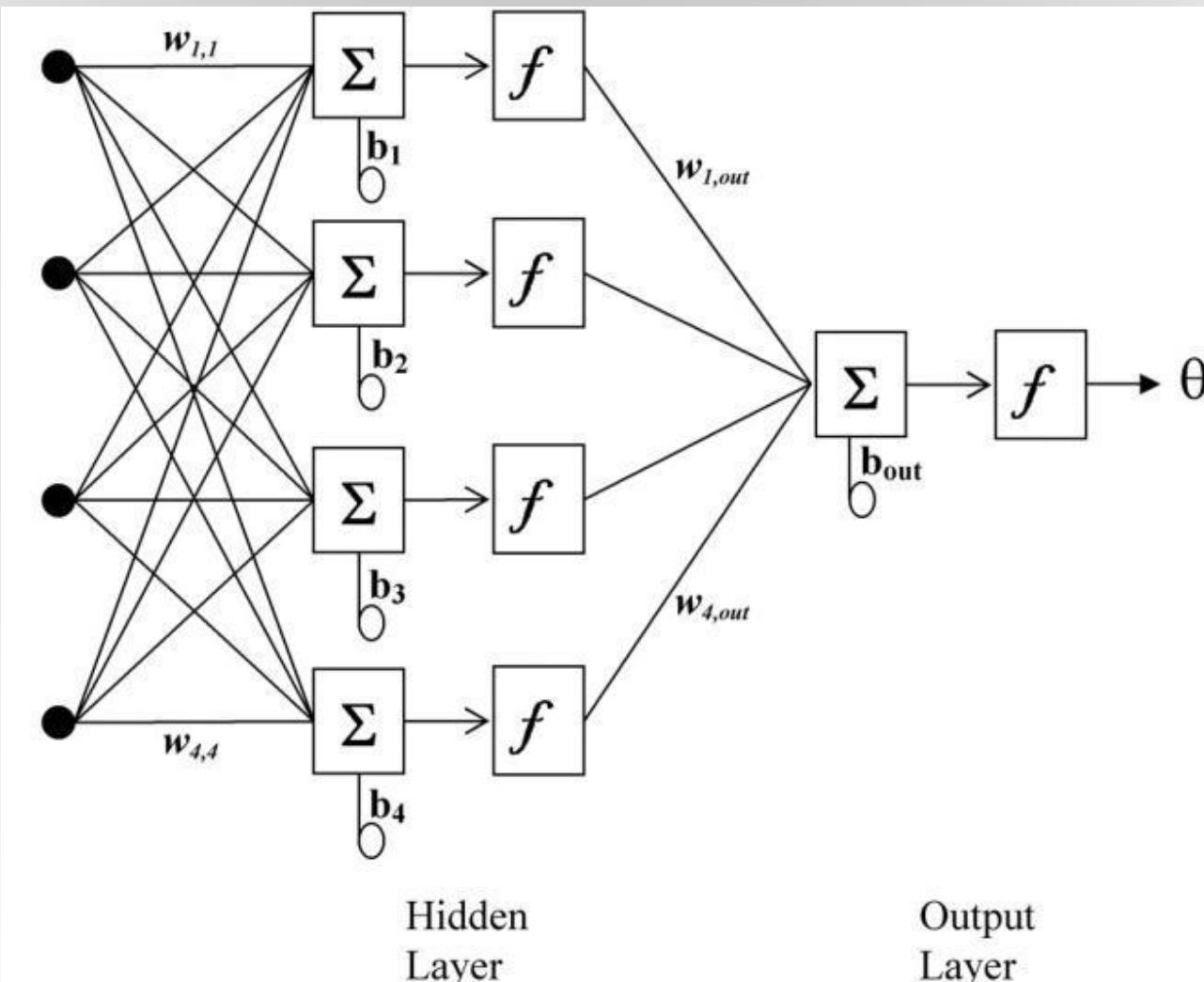
какие веса  
в перцептроне  
решают  
задачу?

# ПРИМЕР: РЕАЛИЗАЦИЯ ЛОГИЧЕСКОГО AND С ПОМОЩЬЮ ПЕРСЕПТРОНА

$$a(x, w) = \text{sign}(x_1 + x_2 - 1.5)$$



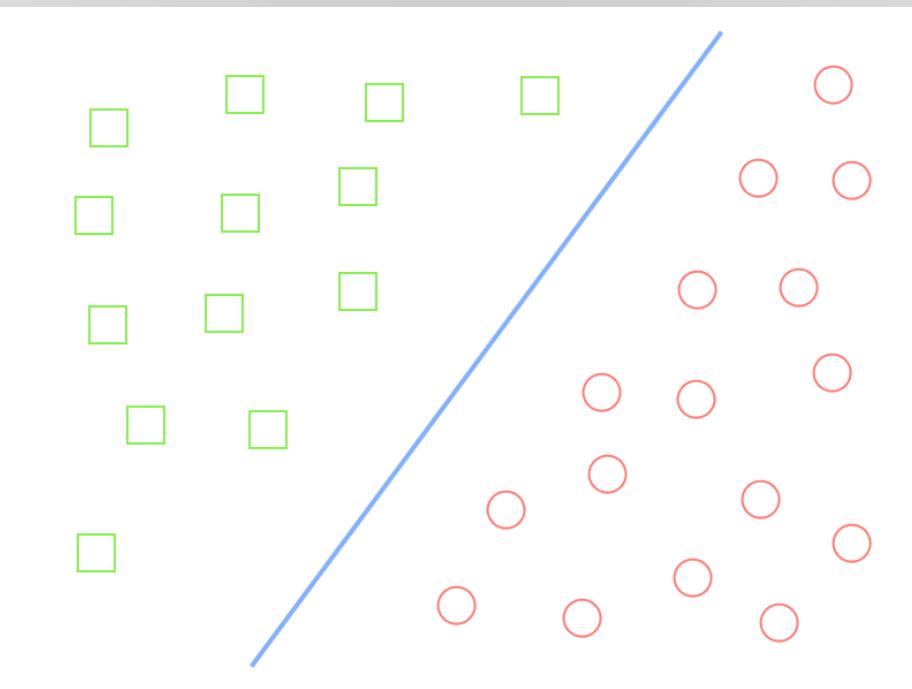
# ПРИМЕР ДВУХСЛОЙНОГО ПЕРСЕПТРОНА



# МЕТОД ОПОРНЫХ ВЕКТОРОВ

# ЛИНЕЙНО РАЗДЕЛИМАЯ ВЫБОРКА

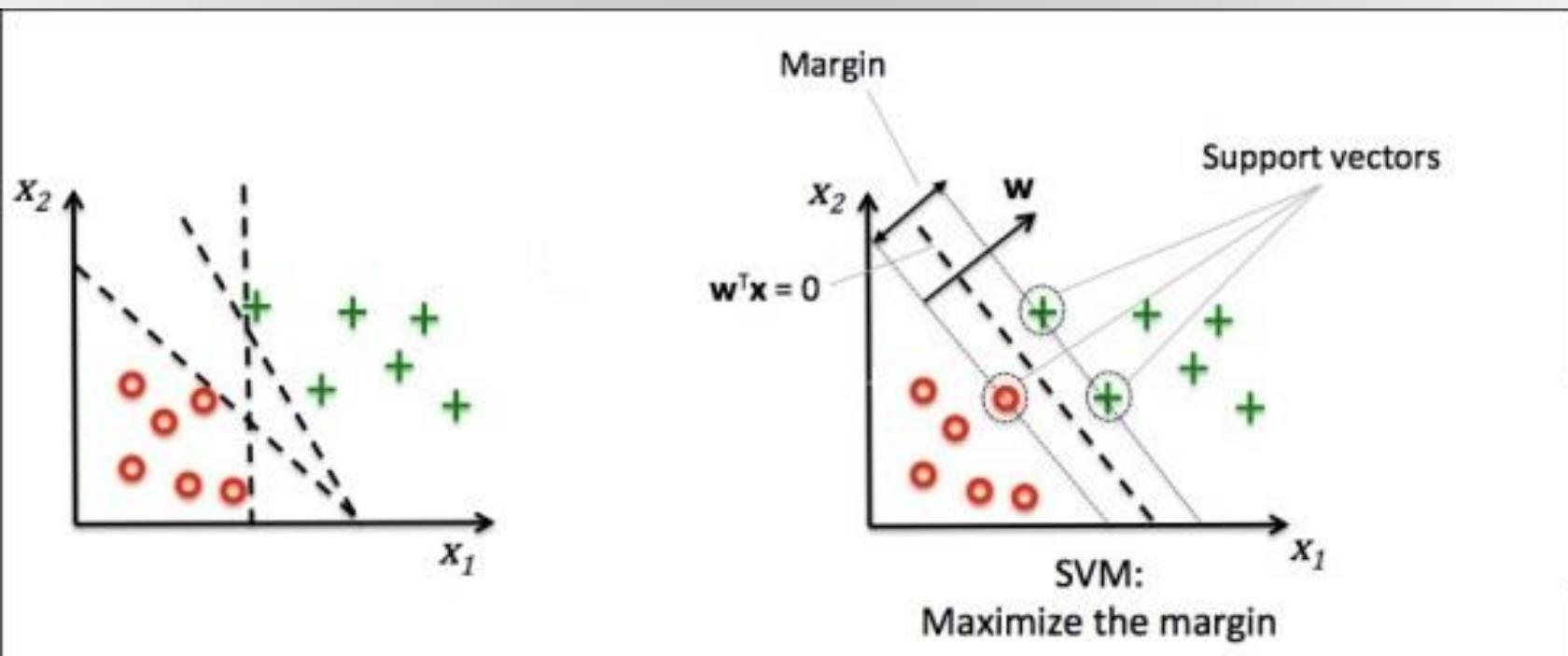
Выборка *линейно разделима*, если существует такой вектор параметров  $w^*$ , что соответствующий классификатор  $a(x)$  не допускает ошибок на этой выборке.



Единственна ли эта разделяющая прямая?

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

Цель метода опорных векторов (Support Vector Machine) –  
максимизировать ширину разделяющей полосы.  
*Какая разделяющая прямая „самая правильная“?*

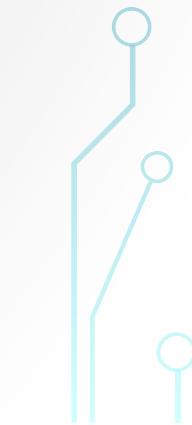


# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ



- $a(x) = sign((w, x) + w_0)$
- Нормируем параметры  $w$  и  $w_0$  так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$



# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

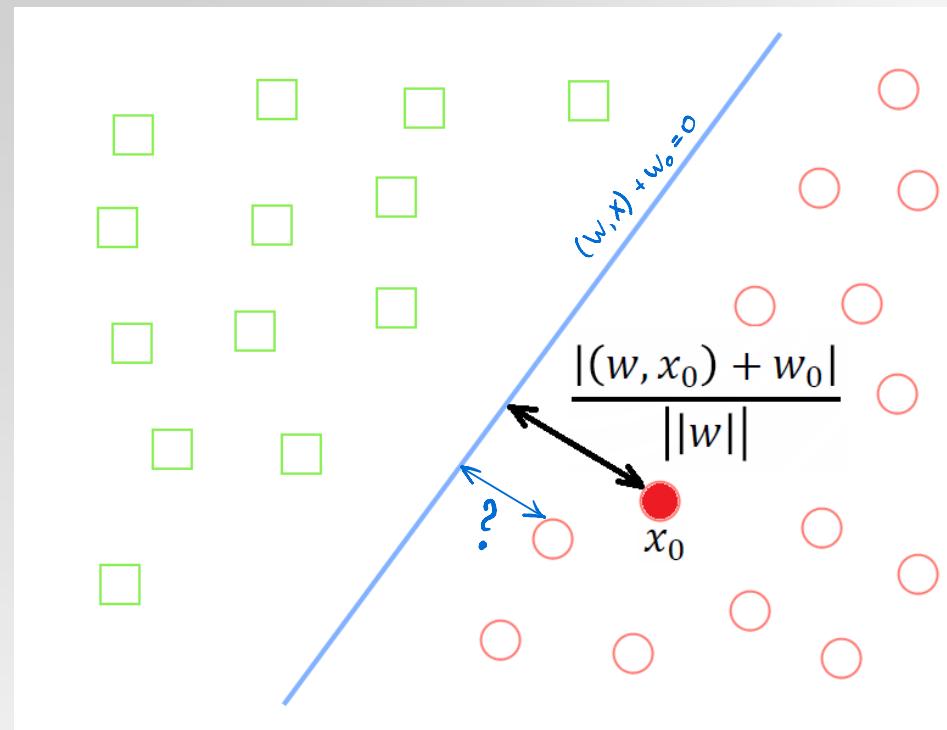
- $a(x) = \text{sign}((w, x) + w_0)$
- Нормируем параметры  $w$  и  $w_0$  так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$

Расстояние от точки  $x_0$  до разделяющей гиперплоскости, задаваемой классификатором:

$$\rho(x_0, a) = \frac{|(w, x_0) + w_0|}{\|w\|}$$

расстояние от точки  $x_0$  до гиперплоскости



# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

- Нормируем параметры  $w$  и  $w_0$  так, что

$$\min_{x \in X} |(w, x) + w_0| = 1$$

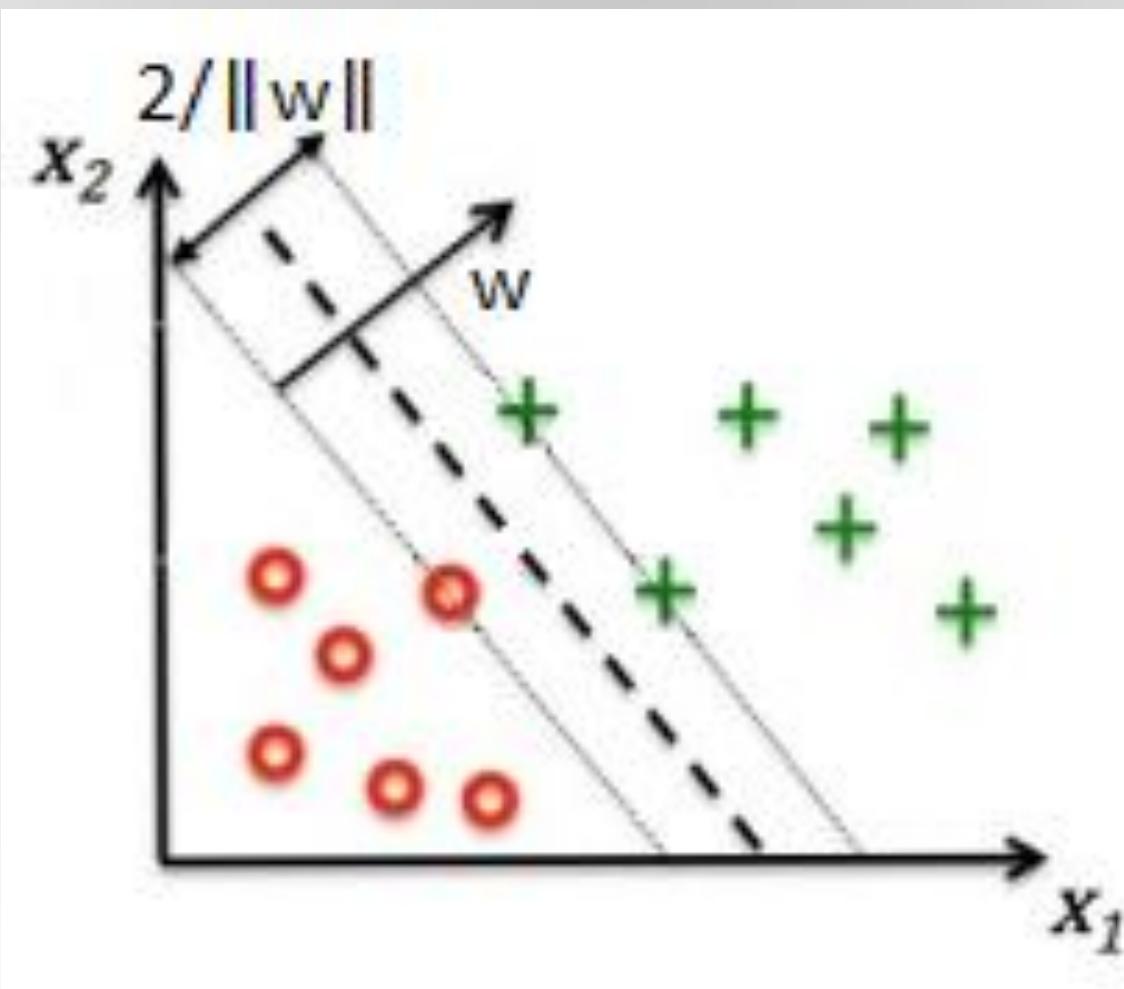
Тогда расстояние от точки  $x_0$  до разделяющей гиперплоскости, задаваемой классификатором:

$$\rho(x_0, a) = \frac{|(w, x_0) + w_0|}{\|w\|}$$

- Расстояние до ближайшего объекта  $x \in X$ :

$$\min_{x \in X} \frac{|(w, x) + w_0|}{\|w\|} = \frac{1}{\|w\|} \min_{x \in X} |(w, x) + w_0| = \frac{1}{\|w\|}$$

# РАЗДЕЛЯЮЩАЯ ПОЛОСА



$$\Rightarrow \frac{2}{\|w\|} \rightarrow_{\max} m_w$$

# ОПТИМИЗАЦИОННАЯ ЗАДАЧА SVM ДЛЯ РАЗДЕЛИМОЙ ВЫБОРКИ

$$\begin{aligned} & \frac{1}{2} \|w\|^2 \rightarrow \min \\ & y_i((w, x_i) + w_0) \geq 1, i = 1, \dots, l \\ & \frac{1}{2} \|w\|^2 \rightarrow \max \end{aligned}$$

максимизация ширины разделяющей полосы

условие, что выборка линейно разделима

M:

**Утверждение.** Данная оптимизационная задача имеет единственное решение.

# ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

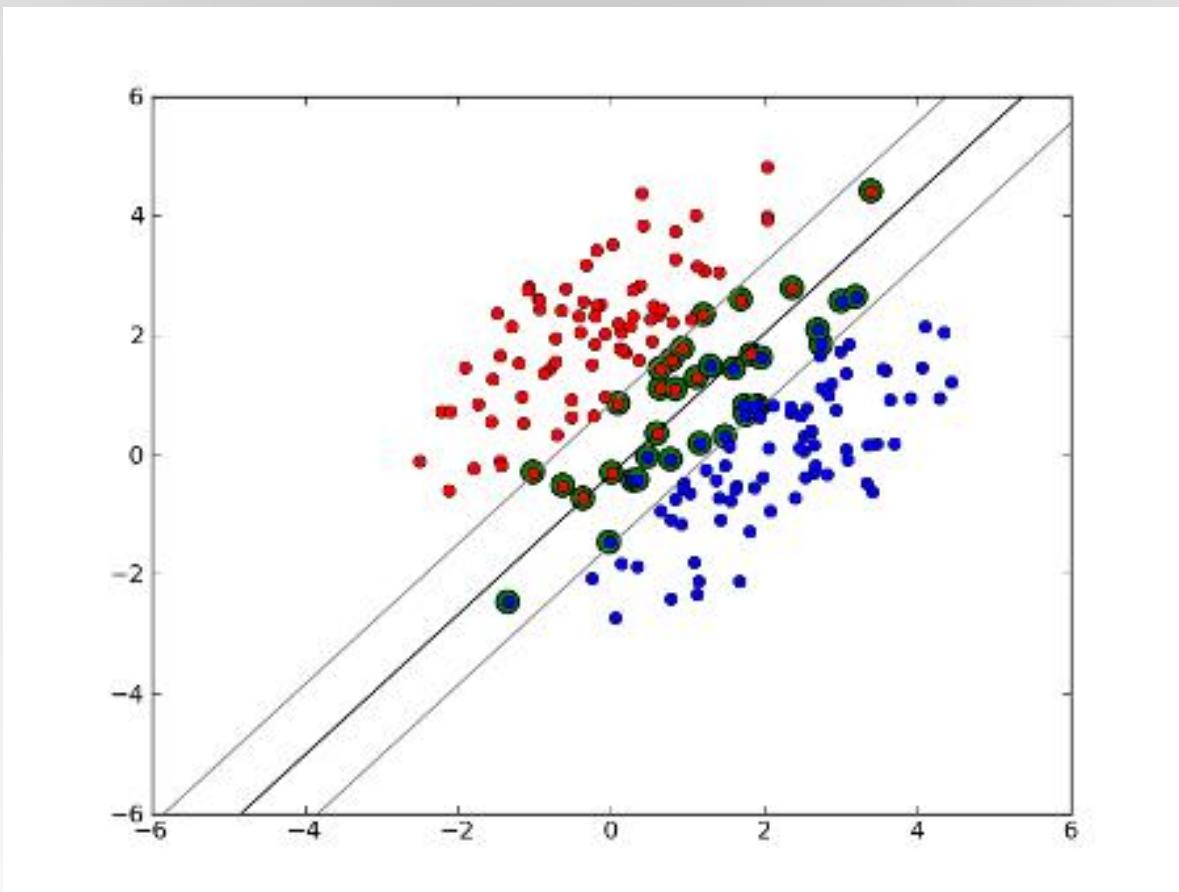
- Существует хотя бы один объект  $x \in X$ , что

$$y_i((w, x_i) + w_0) < 1$$

# ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

- Существует хотя бы один объект  $x \in X$ , что

$$y_i((w, x_i) + w_0) < 1$$



# ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

- Существует хотя бы один объект  $x \in X$ , что

$$y_i((w, x_i) + w_0) < 1$$

Смягчим ограничения, введя штрафы  $\xi_i \geq 0$ :

$$y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l$$

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Хотим:

- Минимизировать штрафы  $\sum_{i=1}^l \xi_i$
- Максимизировать отступ  $\frac{1}{||w||}$

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Хотим:

- Минимизировать штрафы  $\sum_{i=1}^l \xi_i$
- Максимизировать отступ  $\frac{1}{\|w\|}$

Задача оптимизации:

$$\left\{ \begin{array}{l} \text{максимизация отступа} \\ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} \\ \text{минимизация штрафов} \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l \\ \xi_i \geq 0, i = 1, \dots, l \end{array} \right.$$

Как сильно нужно обращать внимание на штрафы?

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

**Утверждение.** Задача

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l \\ \xi_i \geq 0, i = 1, \dots, l \end{cases}$$

Является выпуклой и имеет единственное решение.

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) = 1 - M_i \\ \xi_i \geq 0 \end{cases}$$

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) \\ \xi_i \geq 0 \end{cases} \Rightarrow \xi_i = \max(0, 1 - y_i((w, x_i) + w_0))$$

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} & (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l & (2) \\ \xi_i \geq 0, i = 1, \dots, l & (3) \end{cases}$$

- Перепишем (2) и (3):

$$\begin{cases} \xi_i \geq 1 - y_i((w, x_i) + w_0) \\ \xi_i \geq 0 \end{cases} \Rightarrow \xi_i = \max(0, 1 - y_i((w, x_i) + w_0))$$

Получаем безусловную задачу оптимизации:

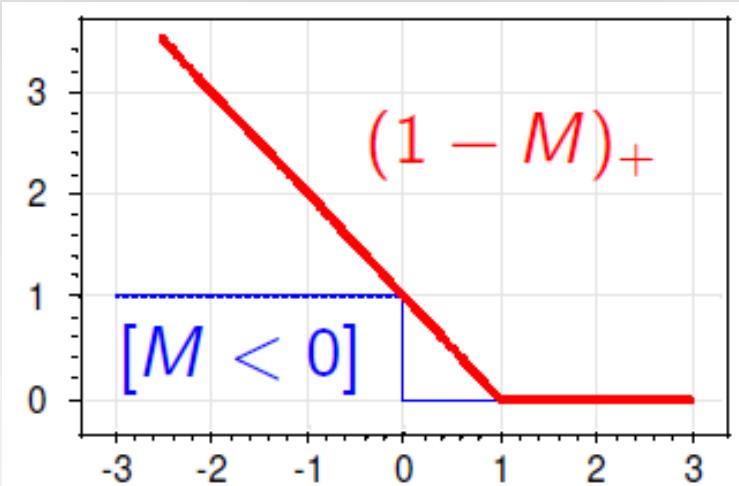
$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \max(0, 1 - y_i((w, x_i) + w_0)) \rightarrow \min_{w, w_0}$$

φ-из потерь  
в SVM

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: ЗАДАЧА ОПТИМИЗАЦИИ

- На задачу оптимизации SVM можно смотреть, как на оптимизацию функции потерь  $L(M) = \max(0, 1 - M) = (1 - M)_+$  с регуляризацией:

$$Q(a, X) = \sum_{i=1}^l (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}$$

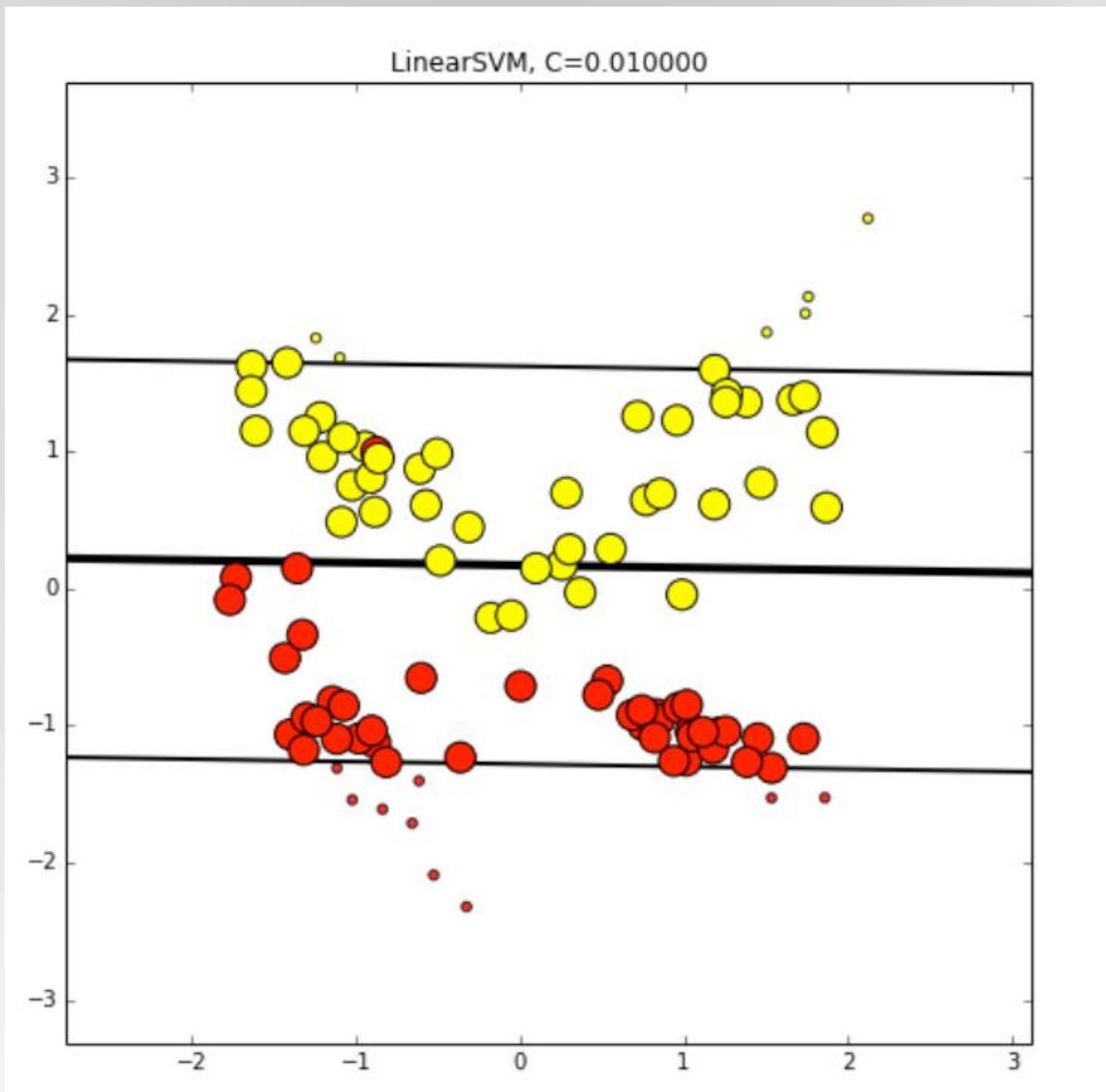


# ЗНАЧЕНИЕ КОНСТАНТЫ С

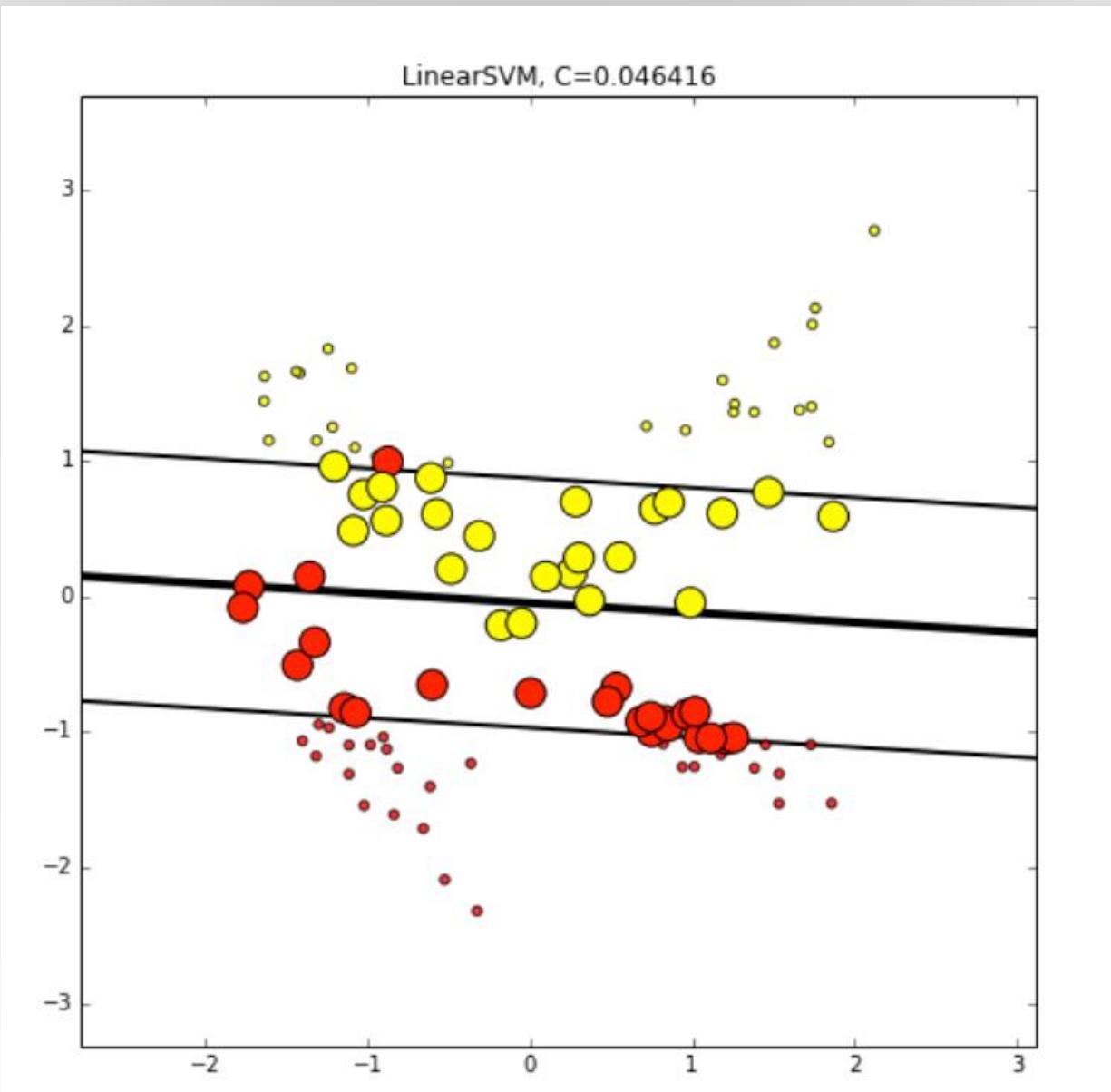
$$\left\{ \begin{array}{l} \frac{1}{2} \|w\|^2 + \textcolor{red}{C} \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{array} \right.$$

Положительная константа  $C$  является управляющим параметром метода и позволяет находить компромисс между максимизацией разделяющей полосы и минимизацией суммарной ошибки.

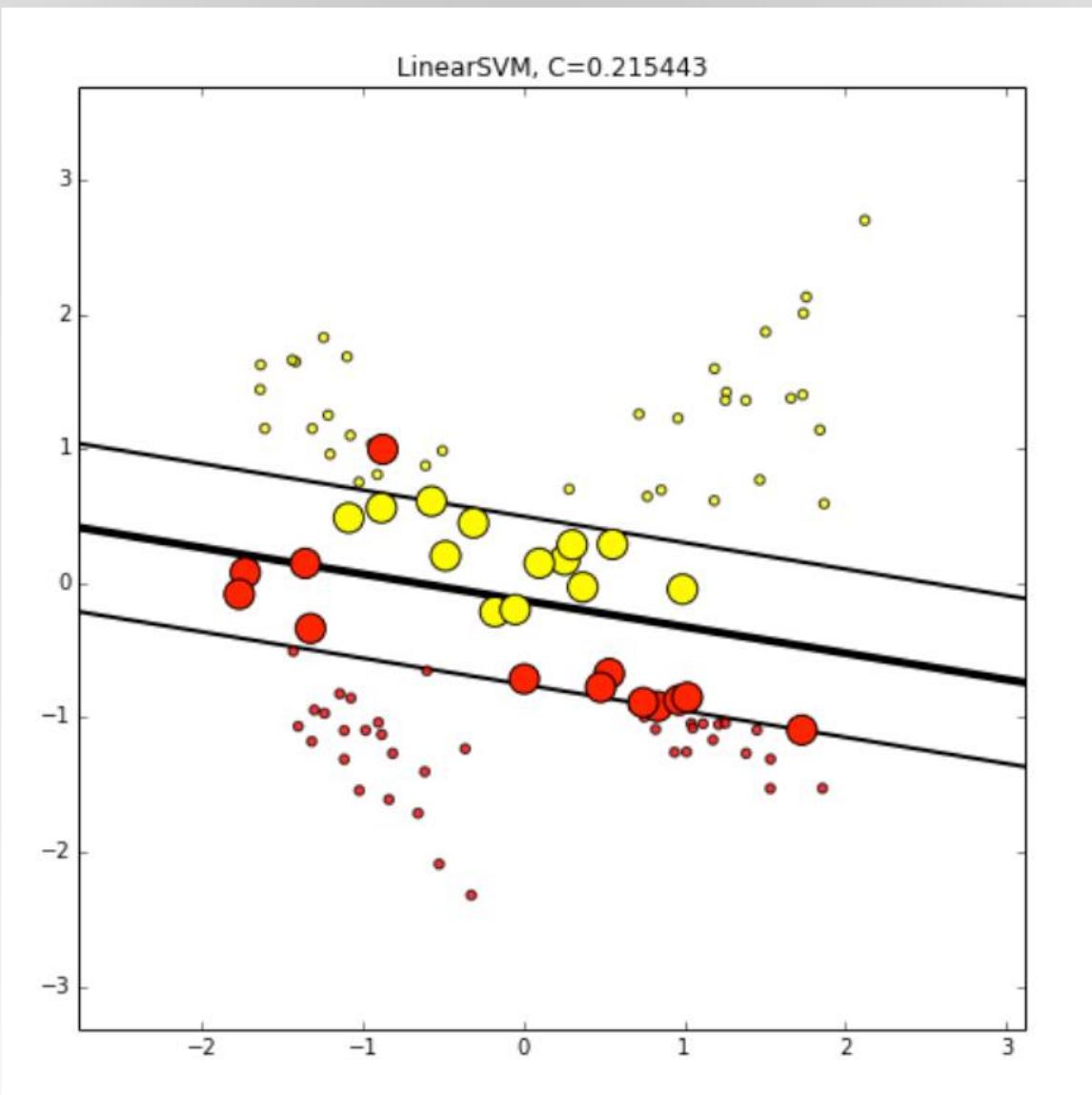
# ЗНАЧЕНИЕ КОНСТАНТЫ С



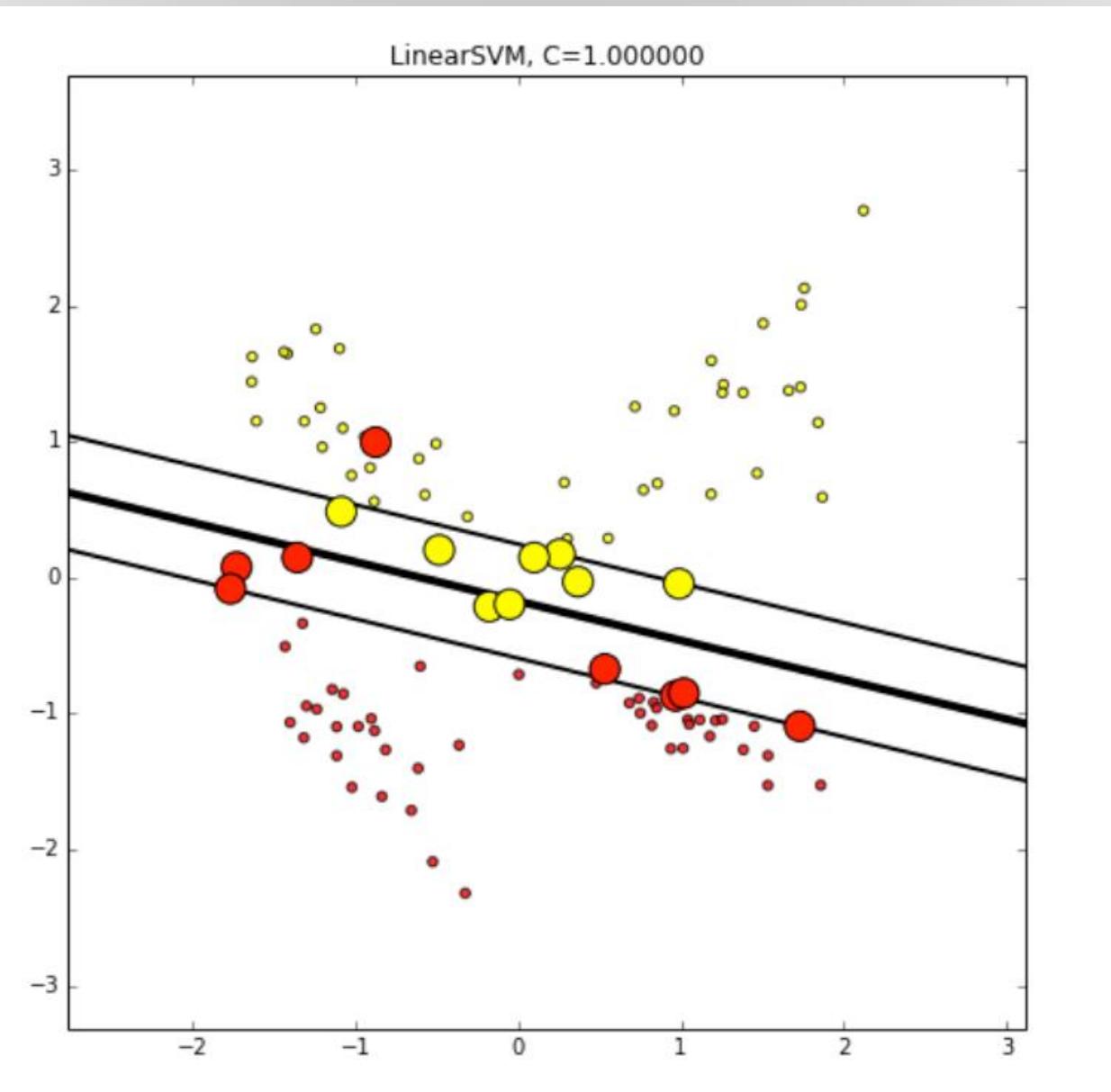
# ЗНАЧЕНИЕ КОНСТАНТЫ С



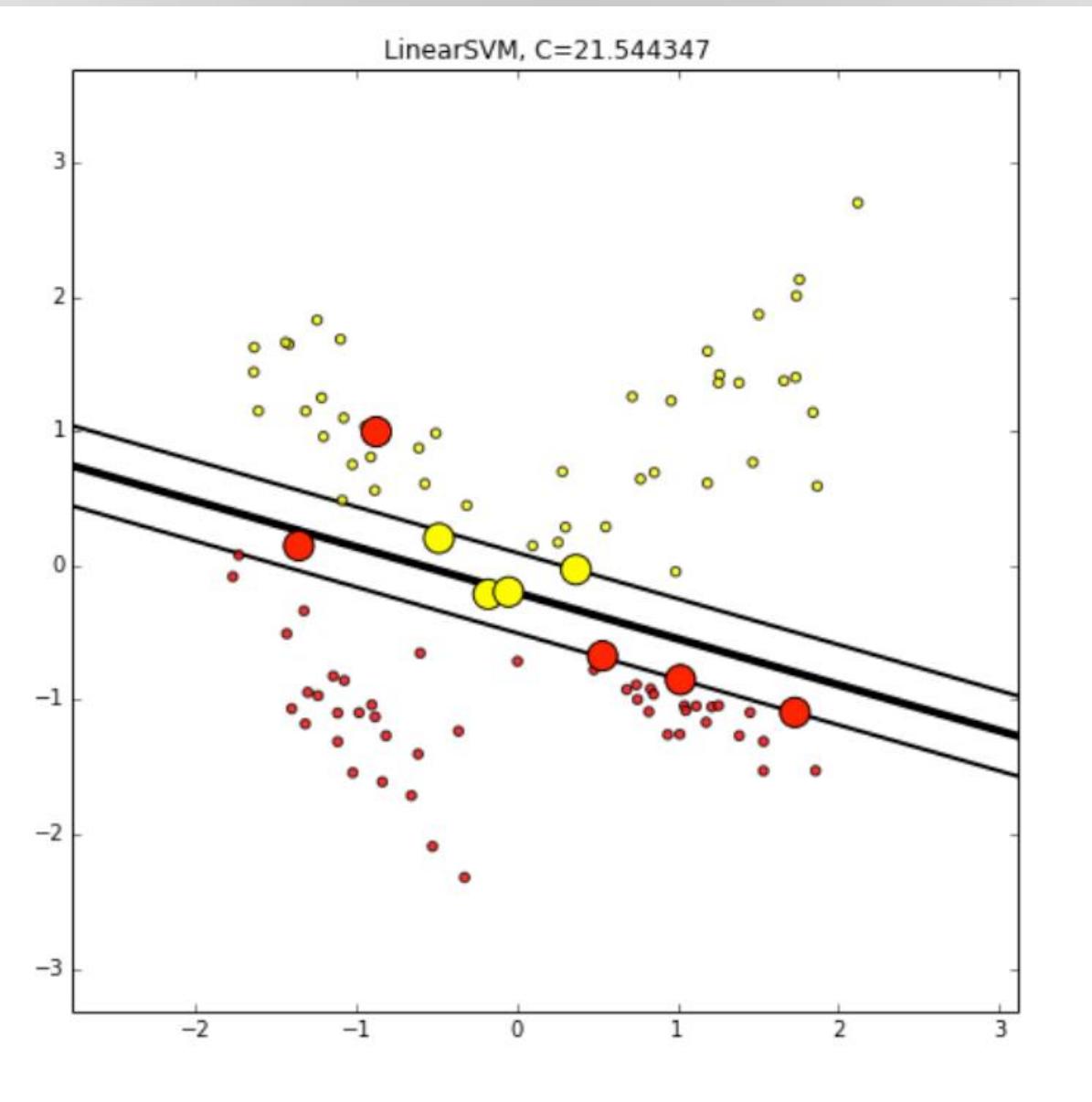
# ЗНАЧЕНИЕ КОНСТАНТЫ С



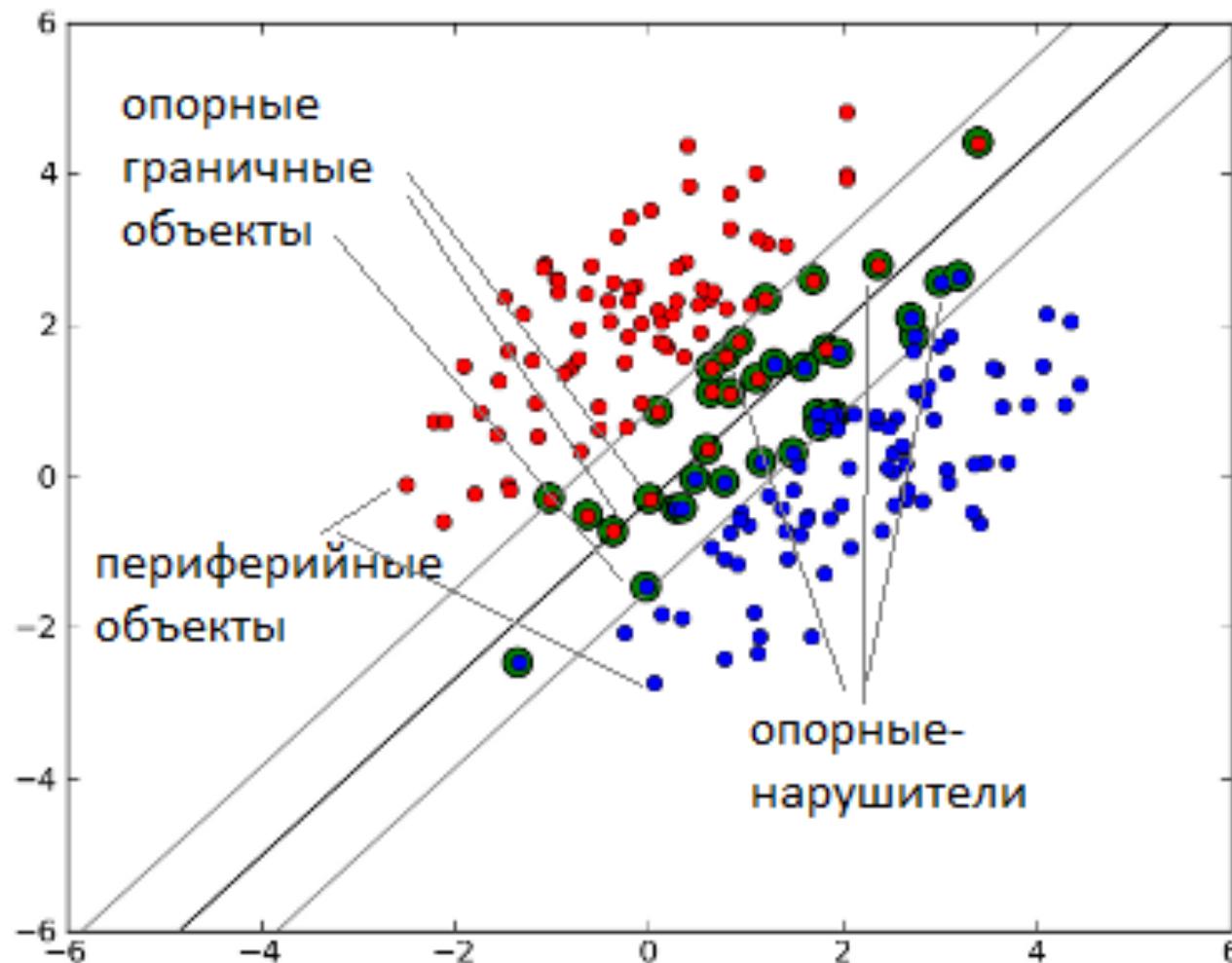
# ЗНАЧЕНИЕ КОНСТАНТЫ С



# ЗНАЧЕНИЕ КОНСТАНТЫ С



# ТИПЫ ОБЪЕКТОВ В SVM



# МЕТРИКИ КАЧЕСТВА

# МЕТРИКИ КАЧЕСТВА КЛАССИФИКАЦИИ

- Accuracy – доля правильных ответов:

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

Когда работает плохо?

# 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*

# МАТРИЦА ОШИБОК

Матрица ошибок (confusion matrix):

реальные  
ответы

		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

→ мои предсказания

## 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  $a_1(x)$ :

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

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	80	20	
$a(x) = -1$ Не получили кредит	20	80	

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

# 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  $a_1(x)$ :

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

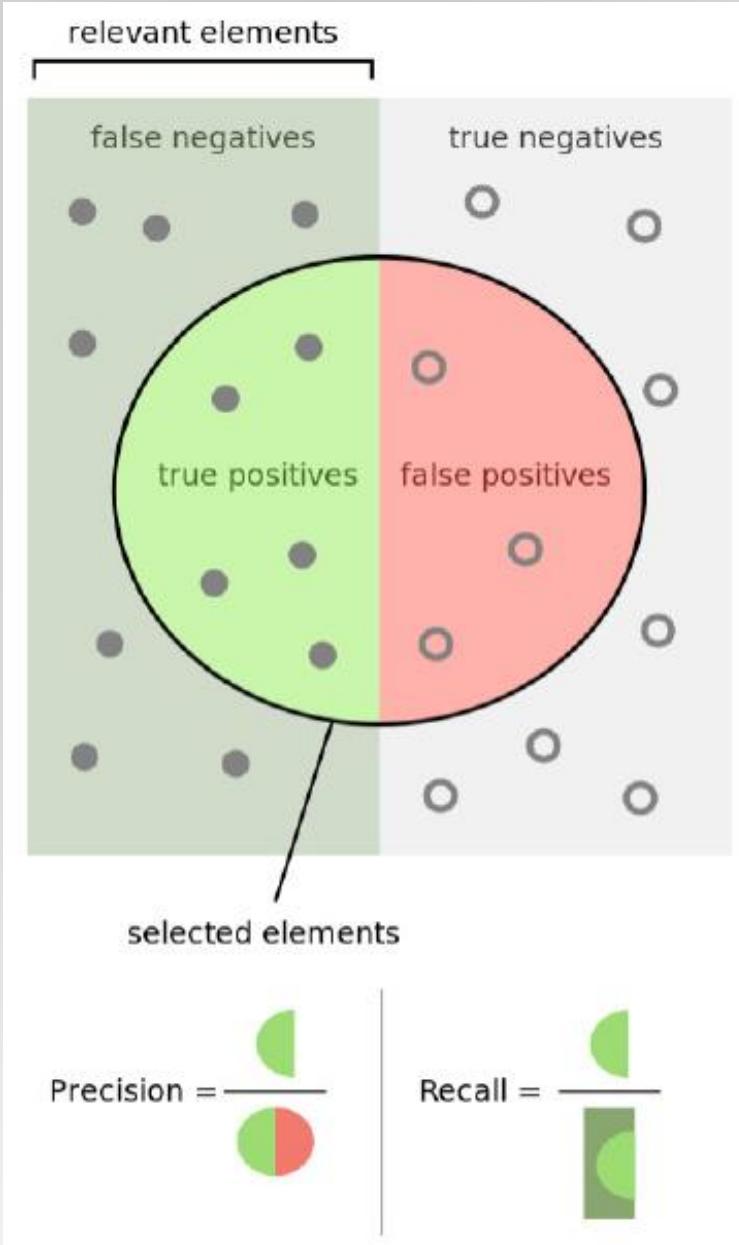
	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20
$a(x) = -1$ Не получили кредит	20	80

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

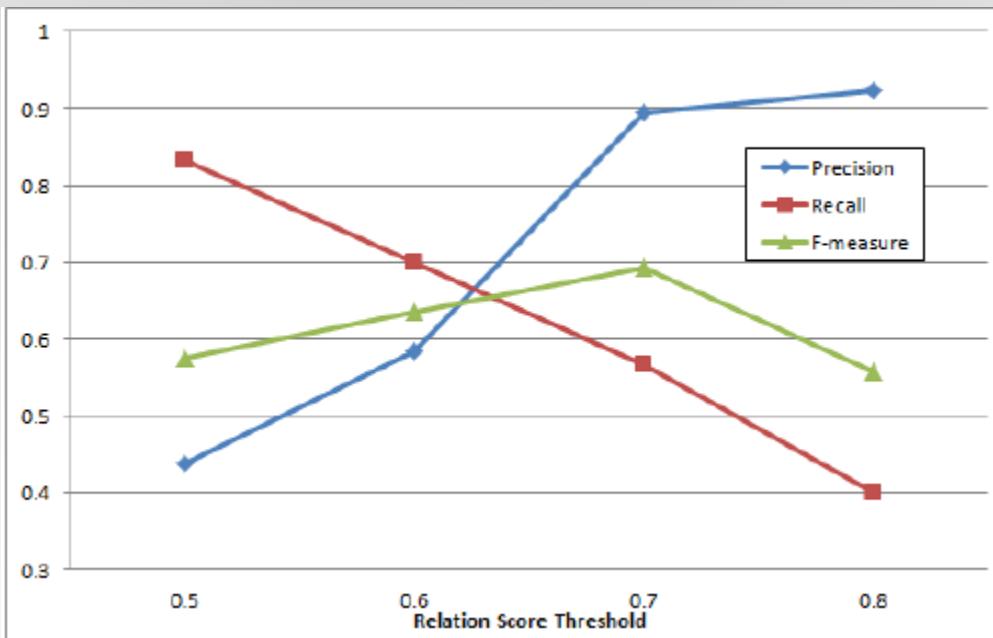
# ТОЧНОСТЬ И ПОЛНОТА



# F-МЕРА

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

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



# РЕГУЛИРУЕМ ТОЧНОСТЬ И ПОЛНОТУ

Пусть  $p(x)$  - уверенность классификатора в том, что объект  $x$  относится к классу +1,  $p(x) \in [0; 1]$ .

Обычно если  $p(x) > 0.5$ , то мы относим объект к положительному классу, а иначе – к отрицательному.

# РЕГУЛИРУЕМ ТОЧНОСТЬ И ПОЛНОТУ

Пусть  $p(x)$  - уверенность классификатора в том, что объект  $x$  относится к классу +1,  $p(x) \in [0; 1]$ .

Обычно если  $p(x) > 0.5$ , то мы относим объект к положительному классу, а иначе – к отрицательному.

Можно изменять этот порог, то есть вместо 0.5 брать другое число из отрезка  $[0; 1]$ .

# РЕГУЛИРУЕМ ТОЧНОСТЬ И ПОЛНОТУ

Пусть  $p(x)$  - уверенность классификатора в том, что объект  $x$  относится к классу +1,  $p(x) \in [0; 1]$ .

Обычно если  $p(x) > 0.5$ , то мы относим объект к положительному классу, а иначе – к отрицательному.

Можно изменять этот порог, то есть вместо 0.5 брать другое число из отрезка  $[0; 1]$ .

Путем изменения порога  $t$  можно регулировать точность и полноту:

➤ Чему будут равны точность и полнота при  $t = 0$ ?

# РЕГУЛИРУЕМ ТОЧНОСТЬ И ПОЛНОТУ

Пусть  $p(x)$  - уверенность классификатора в том, что объект  $x$  относится к классу +1,  $p(x) \in [0; 1]$ .

Обычно если  $p(x) > 0.5$ , то мы относим объект к положительному классу, а иначе – к отрицательному.

Можно изменять этот порог, то есть вместо 0.5 брать другое число из отрезка  $[0; 1]$ .

Путем изменения порога  $t$  можно регулировать точность и полноту:

- при  $t = 0$  мы все объекты относим к положительному классу, то есть **полнота = 1, а точность маленькая**.

# РЕГУЛИРУЕМ ТОЧНОСТЬ И ПОЛНОТУ

Пусть  $p(x)$  - уверенность классификатора в том, что объект  $x$  относится к классу +1,  $p(x) \in [0; 1]$ .

Обычно если  $p(x) > 0.5$ , то мы относим объект к положительному классу, а иначе – к отрицательному.

Можно изменять этот порог, то есть вместо 0.5 брать другое число из отрезка  $[0; 1]$ .

**Путем изменения порога  $t$  можно регулировать точность и полноту:**

- при  $t = 0$  мы все объекты относим к положительному классу, то есть полнота = 1, а точность маленькая.
- **При увеличении  $t$  полнота уменьшается** (могут появиться объекты положительного класса, которые мы не нашли), **а точность возрастает** (появляются объекты положительного класса).

## ИНТЕГРАЛЬНАЯ МЕТРИКА: ROC-AUC

Хотим измерить качество всего семейства классификаторов независимо от выбранного порога.

Для этого будем использовать метрику AUC

**AUC** – *Area Under ROC Curve (площадь под ROC-кривой)*

# ROC-КРИВАЯ

Для каждого значения порога  $t$  вычислим:

- **False Positive Rate** (доля неверно принятых объектов отрицательного класса):

$$FPR = \frac{FP}{FP + TN} = \frac{\sum_i [y_i = -1] [a(x_i) = +1]}{\sum_i [y_i = -1]}$$

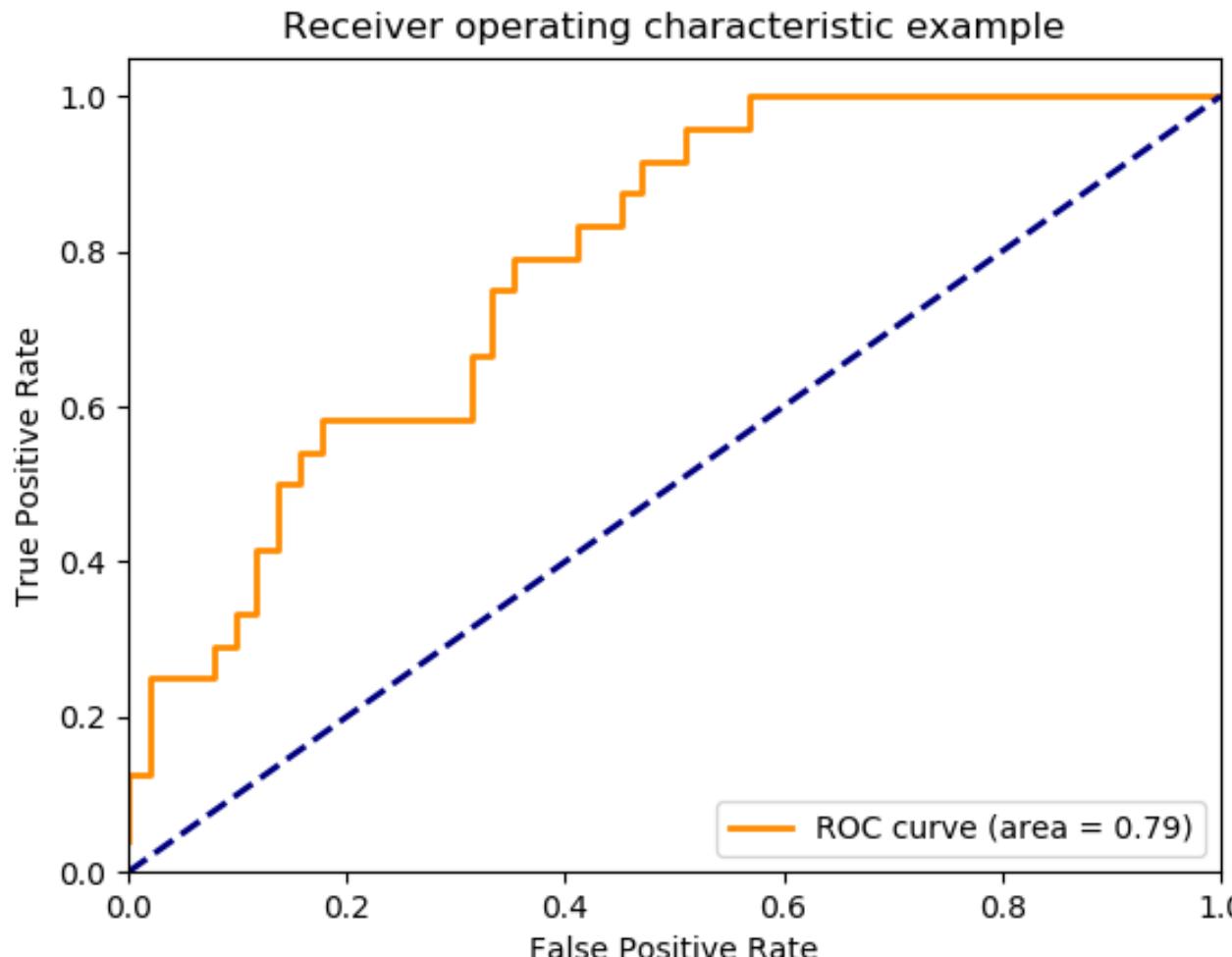
- **True Positive Rate** (доля верно принятых объектов положительного класса):

$$TPR = \frac{TP}{TP+FN} = \frac{\sum_i [y_i=+1] [a(x_i)=+1]}{\sum_i [y_i=+1]}.$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

# ROC-КРИВАЯ

Кривая, состоящая из точек с координатами (FPR,TPR) для всех возможных порогов – это и есть ROC-кривая.

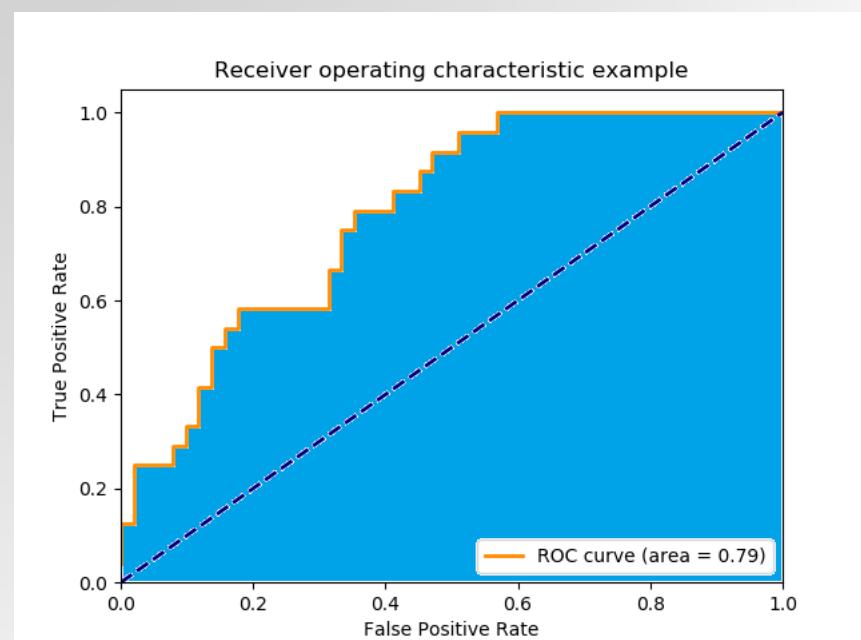


# ROC-КРИВАЯ. AUC.

**AUC (Area Under Curve)** – площадь под ROC-кривой.

$$AUC \in [0; 1].$$

- Чему равен AUC при идеальной классификации?
- Чему равен AUC при случайной классификации?



# ROC-КРИВАЯ. AUC.

**AUC (Area Under Curve)** – площадь под ROC-кривой.

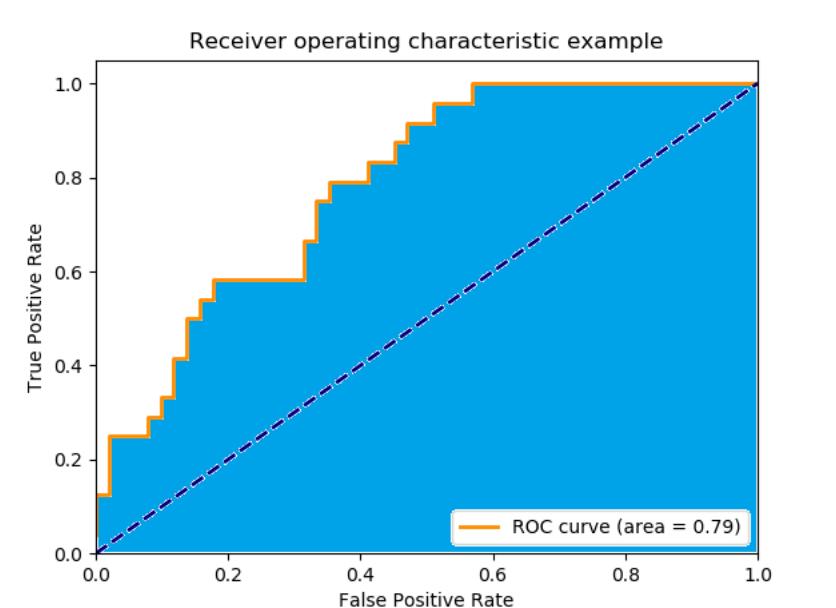
$$AUC \in [0; 1].$$

- $AUC = 1$  –

иdealная классификация

- $AUC = 0.5$  –

случайная классификация



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:  $(0.7, 0.4, 0.2, 0.1, 0.05)$

1 шаг:  $t = 0.7$ , то есть

$$a(x) = [b(x) > 0.7]$$

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:  $(0.7, 0.4, 0.2, 0.1, 0.05)$

1 шаг:  $t = 0.7$ , то есть

$$a(x) = [b(x) > 0.7]$$

$$TPR = \frac{0}{0+3} = 0, \quad FPR = \frac{0}{0+2} = 0.$$

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Пусть есть выборка из 5 объектов и следующие предсказания классификатора оценки принадлежности к классу +1:

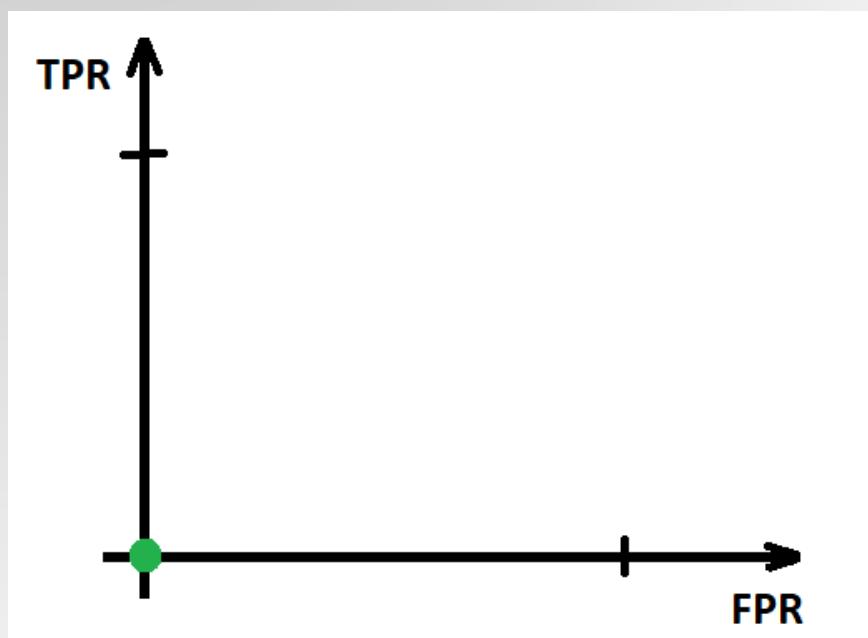
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:  
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

1 шаг:  $t = 0.7$ , то есть  
 $a(x) = [b(x) > 0.7]$

$$TPR = \frac{0}{0+3} = 0,$$

$$FPR = \frac{0}{0+2} = 0.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

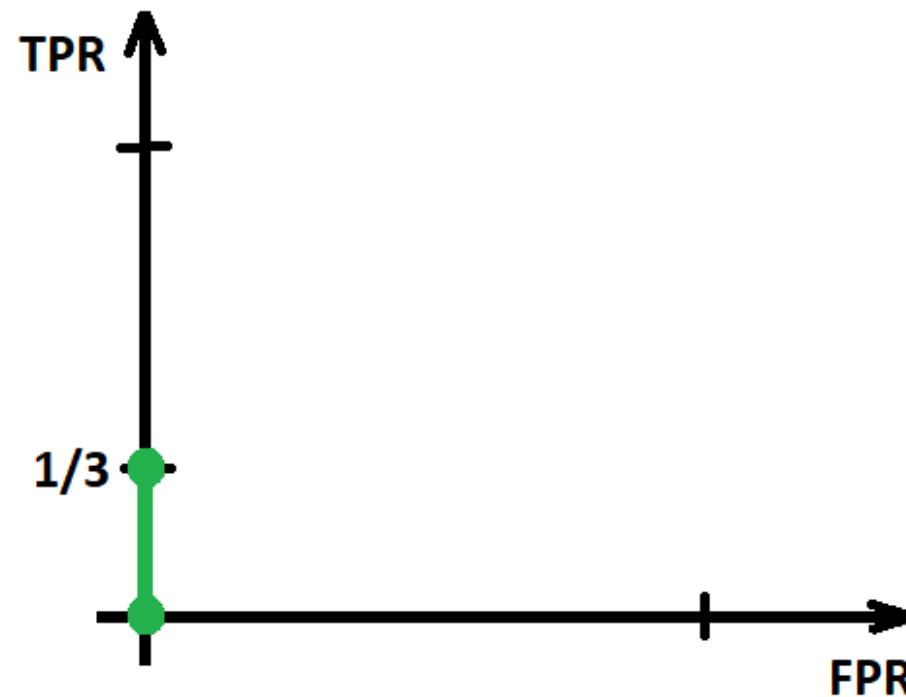
$b(x)$	0.2	0.4	0.1	<b>0.7</b>	0.05
$y$	-1	+1	-1	<b>+1</b>	+1

- Упорядочим объекты по убыванию предсказаний:  
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

**2 шаг:**  $t = 0.4$ , то есть  
 $a(x) = [b(x) > 0.4]$

$$TPR = \frac{1}{1+2} = \frac{1}{3},$$

$$FPR = \frac{0}{0+2} = 0.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

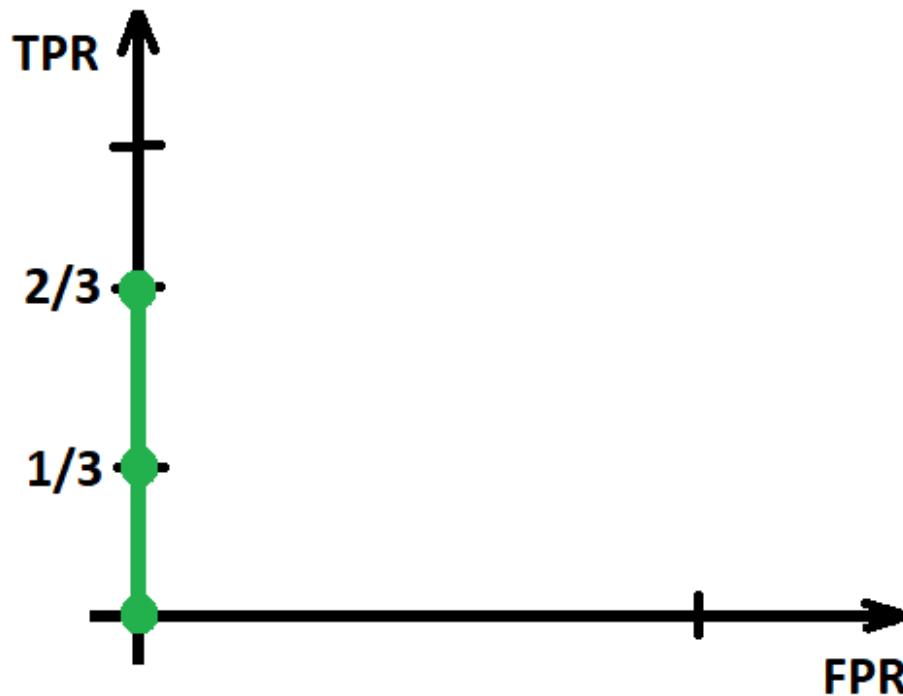
$b(x)$	0.2	<b>0.4</b>	0.1	<b>0.7</b>	0.05
$y$	-1	<b>+1</b>	-1	<b>+1</b>	+1

- Упорядочим объекты по убыванию предсказаний:  
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

3 шаг:  $t = 0.2$ , то есть  
 $a(x) = [b(x) > 0.2]$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{0}{0+2} = 0.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:

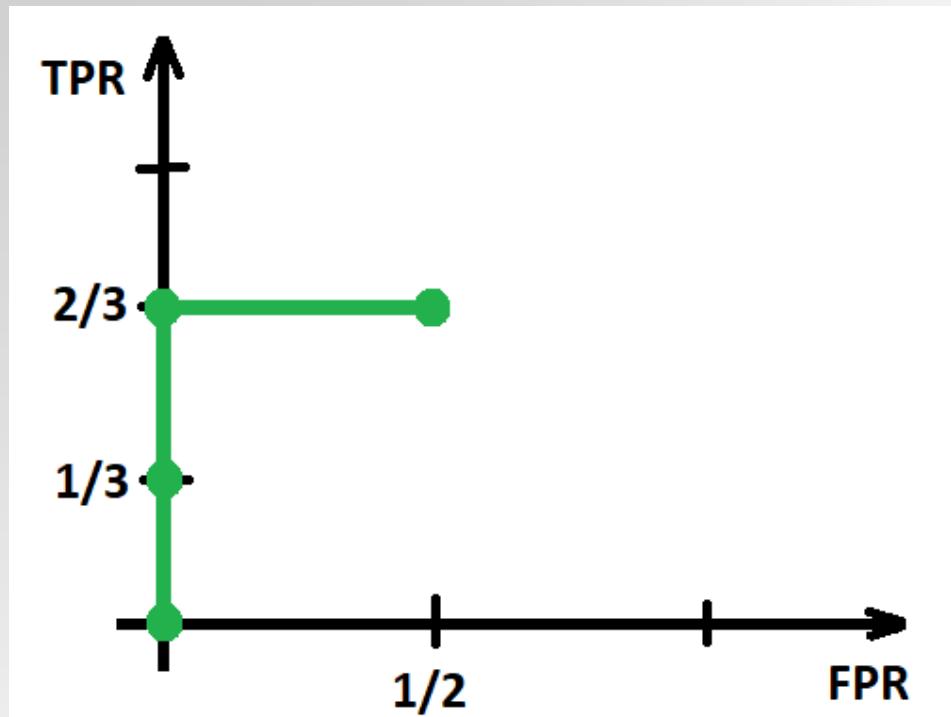
(0.7,0.4,0.2,0.1,0.05)

4 шаг:  $t = 0.1$ , то есть

$a(x) = [b(x) > 0.1]$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{1}{1+1} = \frac{1}{2}.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по

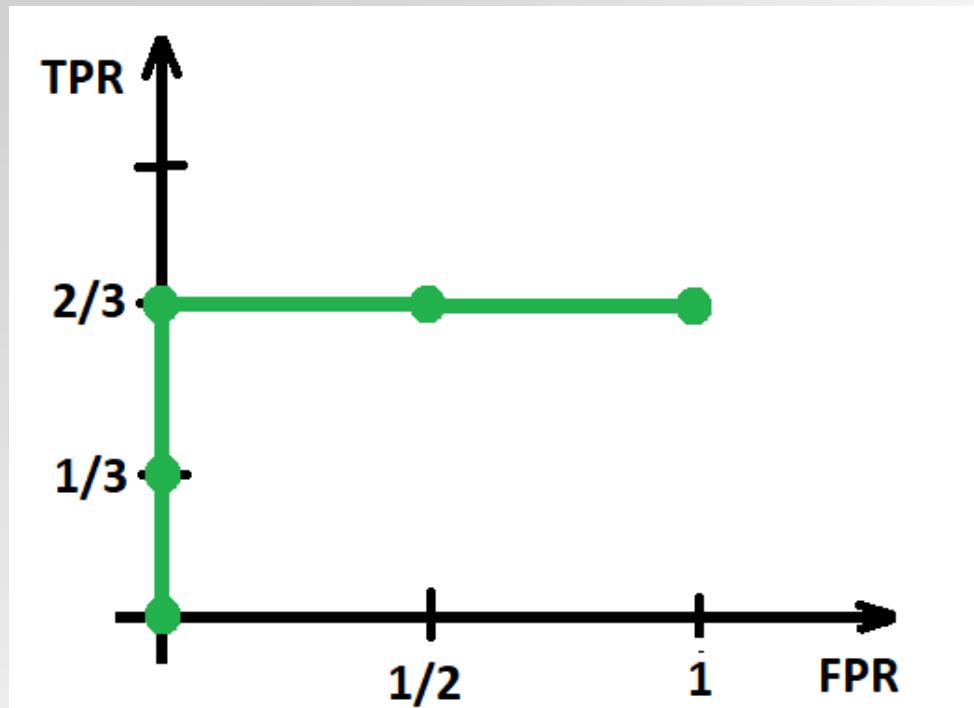
убыванию предсказаний:

(0.7, 0.4, 0.2, 0.1, 0.05)

5 шаг:  $t = 0.05$ , то есть  
 $a(x) = [b(x) > 0.05]$

$$TPR = \frac{2}{2+1} = \frac{2}{3},$$

$$FPR = \frac{2}{2+0} = 1.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

- Оценки принадлежности к классу +1:

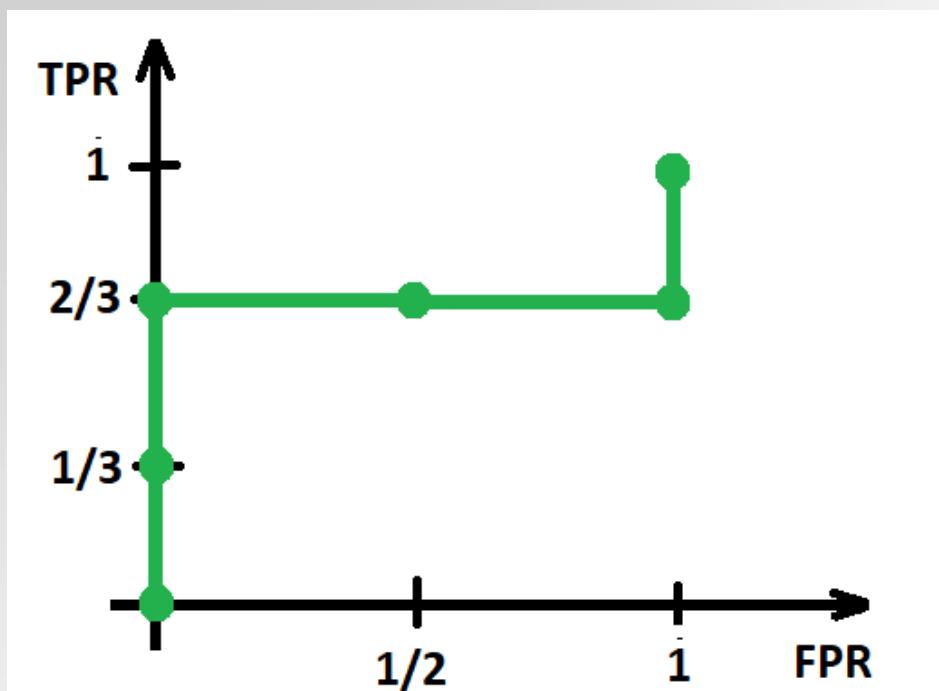
$b(x)$	0.2	0.4	0.1	0.7	0.05
$y$	-1	+1	-1	+1	+1

- Упорядочим объекты по убыванию предсказаний:  
 $(0.7, 0.4, 0.2, 0.1, 0.05)$

**5 шаг:**  $t = 0$ , то есть  
 $a(x) = [b(x) > 0]$

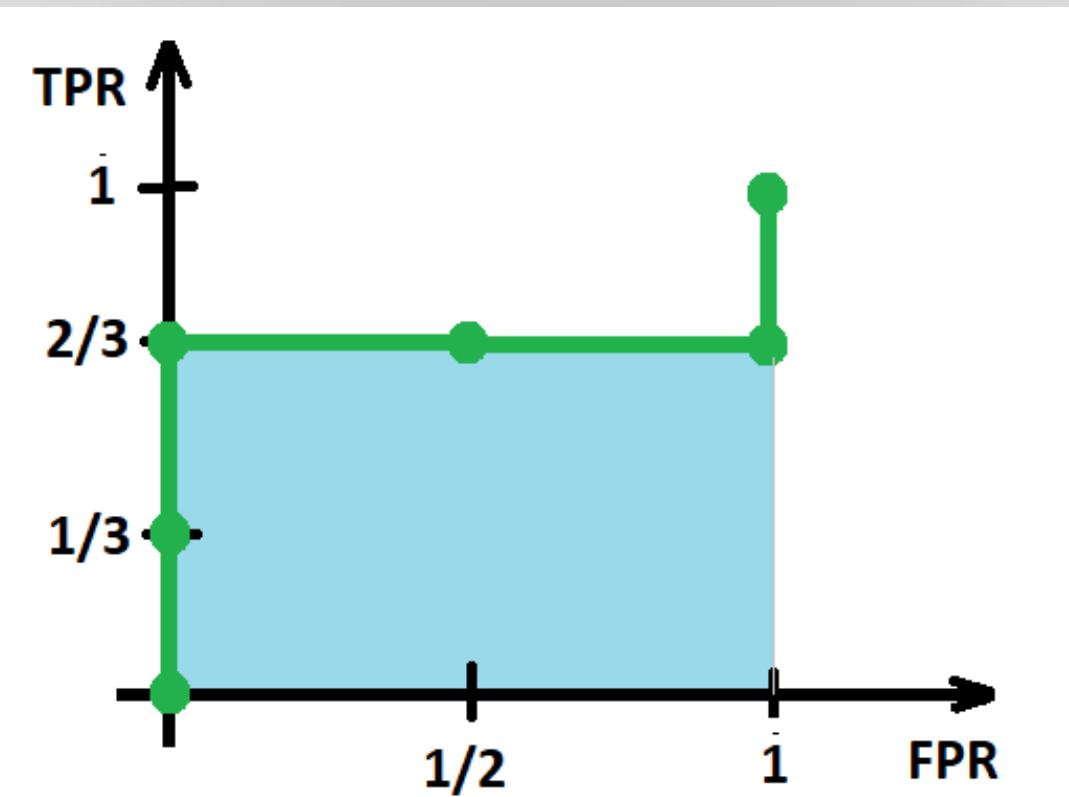
$$TPR = \frac{3}{3+0} = 1,$$

$$FPR = \frac{2}{2+0} = 1.$$



# ПРИМЕР ПОСТРОЕНИЯ ROC-КРИВОЙ

$$AUC = 2/3$$

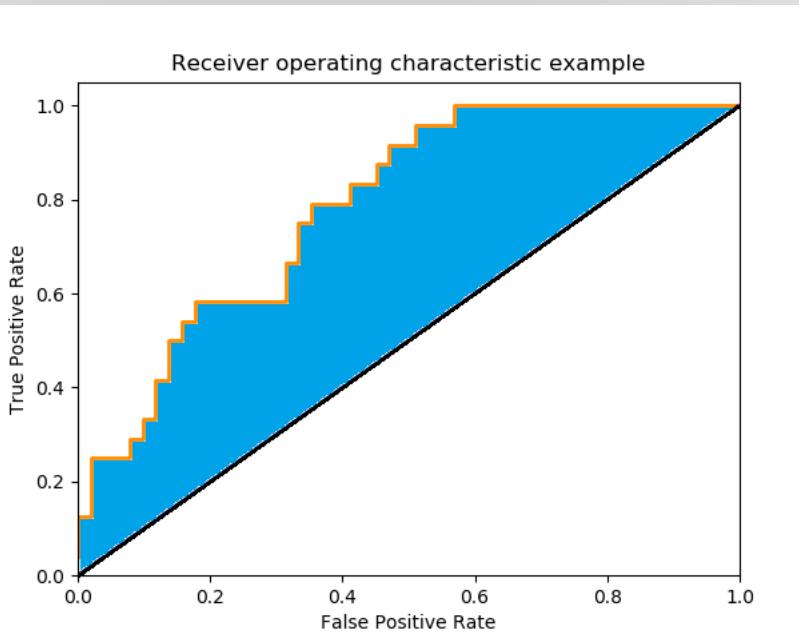


# ИНДЕКС ДЖИНИ

Индекс Джини:

$$Gini = 2 \cdot AUC - 1$$

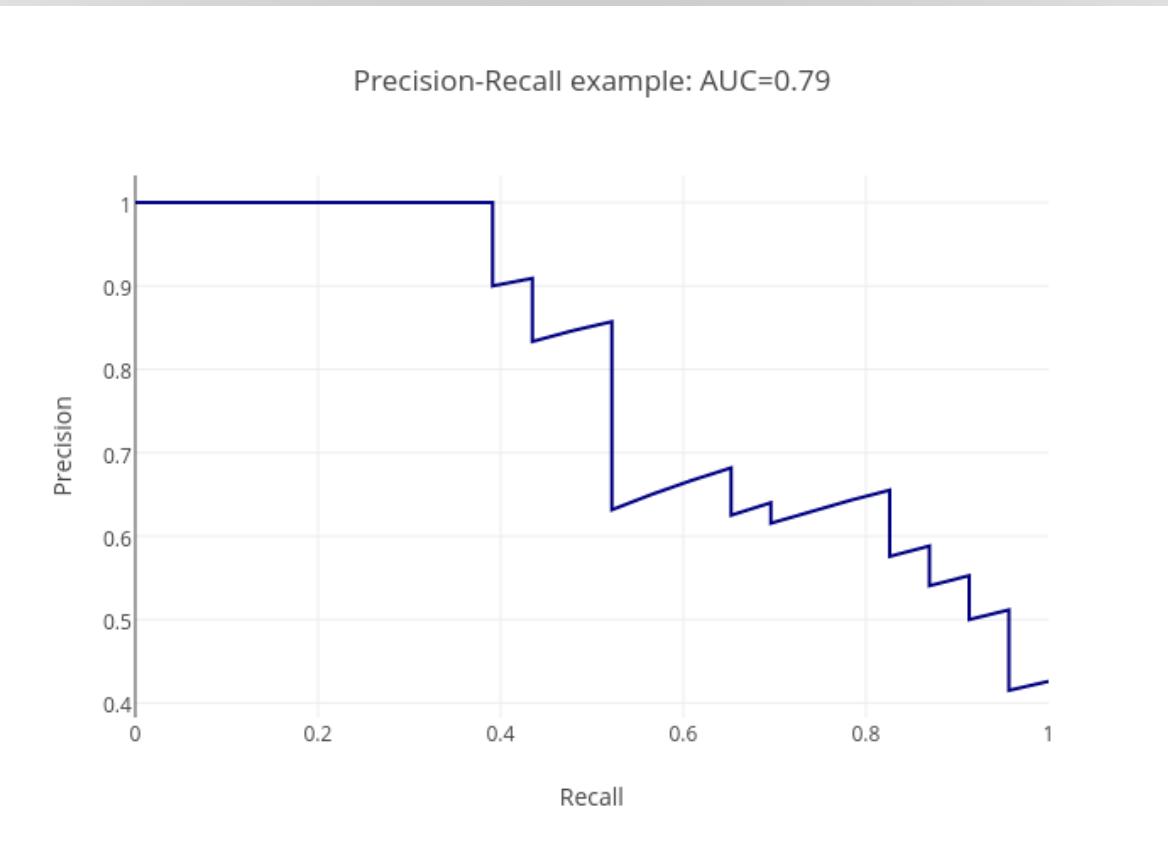
- Индекс Джини – это удвоенная площадь между главной диагональю и ROC-кривой.



# PRECISION-RECALL КРИВАЯ

- В случае малой доли объектов положительного класса AUC-ROC может давать неадекватно хороший результат

Precision-Recall кривая:



# AUC-PR

AUC-PR – площадь под PR-кривой

Precision-Recall example: AUC=0.79

