

# Линейные модели классификации

# Логистическая регрессия

# Оценивание вероятностей

- $P(y = 1 | x) = \pi(x)$

# Оценивание вероятностей

- Кредитный скоринг
- Стратегия: выдавать кредит только клиентам с  $\pi(x) > 0.9$
- 10% невозвращённых кредитов — нормально

# Оценивание вероятностей

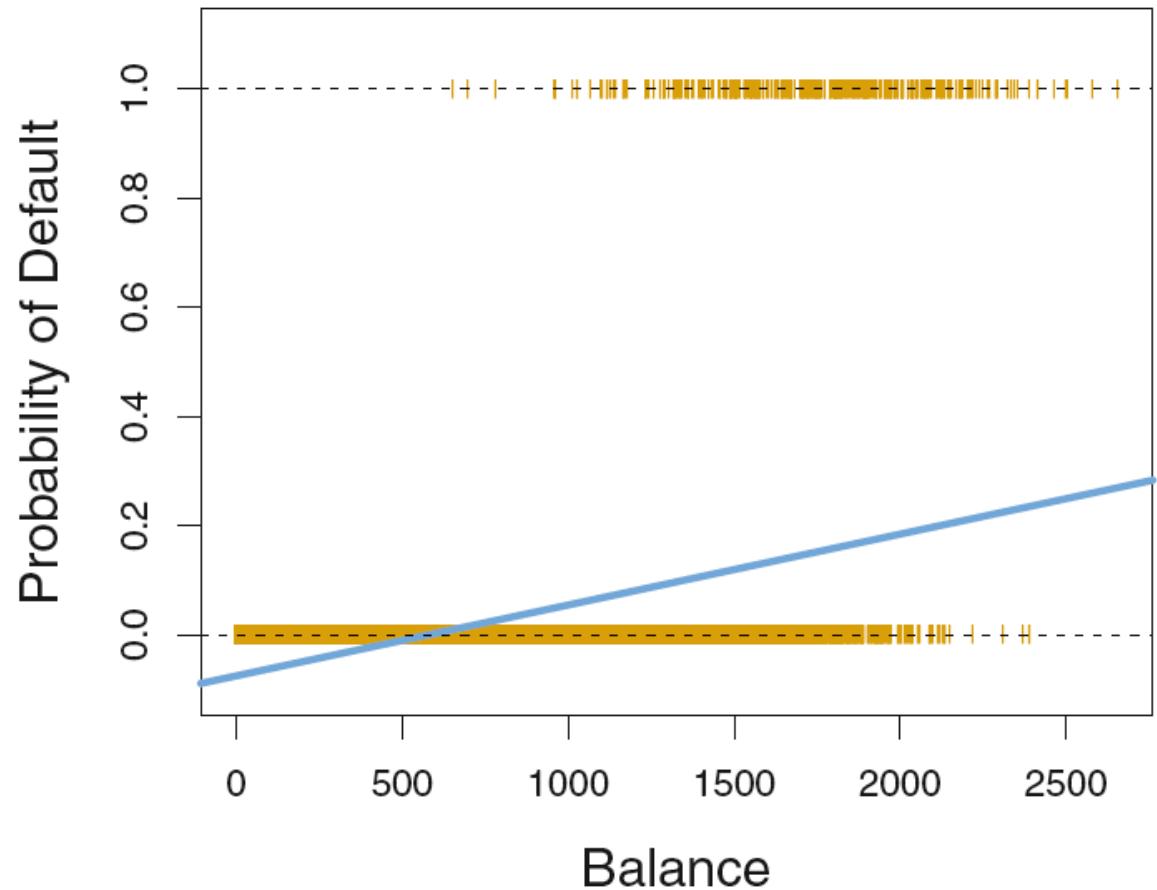
- Баннерная реклама
- $\pi(x)$  — вероятность, что пользователь кликнет по рекламе
- $c(x)$  — прибыль в случае клика
- $\pi(x)c(x)$  — хотим оптимизировать

# Оценивание вероятностей

- $P(y = 1 | x) = \pi(x)$
- $\pi(x)$  — вещественное число
- Классификатор не подходит

# Регрессия?

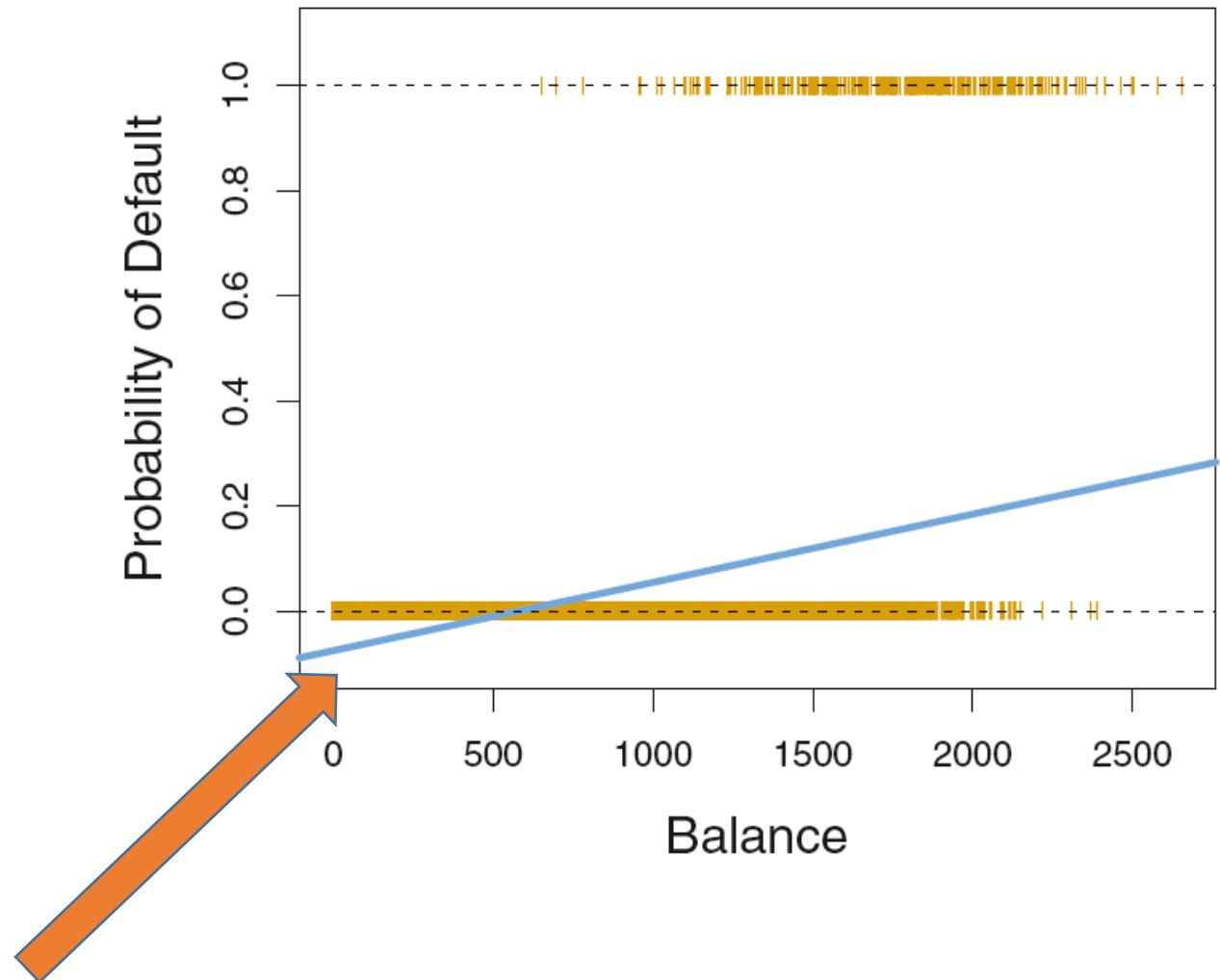
- $\pi(x) \approx \langle w, x \rangle = w_1 x + w_0$



# Регрессия?

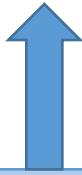
- $\pi(x) \approx \langle w, x \rangle = w_1 x + w_0$

Отрицательная вероятность о\_О



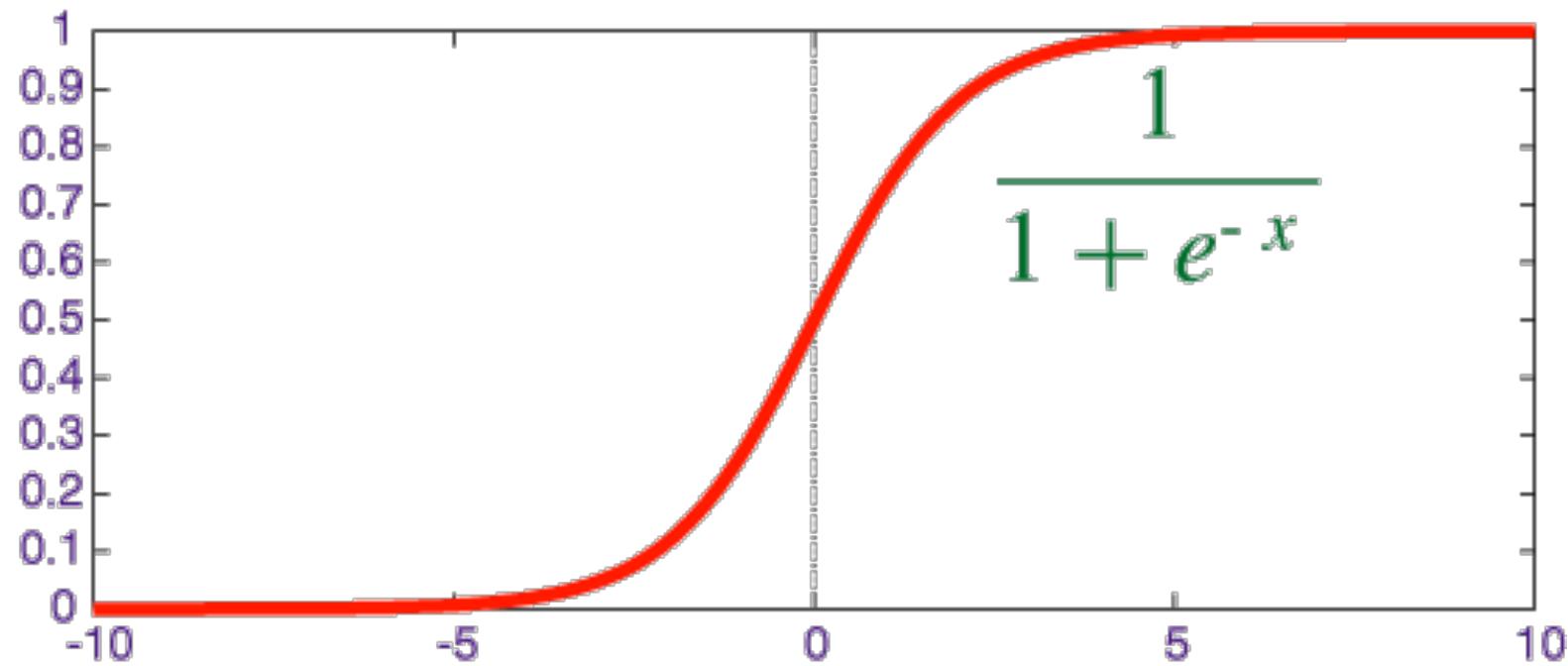
# Регрессия?

$$\pi(x) \approx \sigma(\langle w, x \rangle) = \frac{1}{1 + \exp(-\langle w, x \rangle)}$$

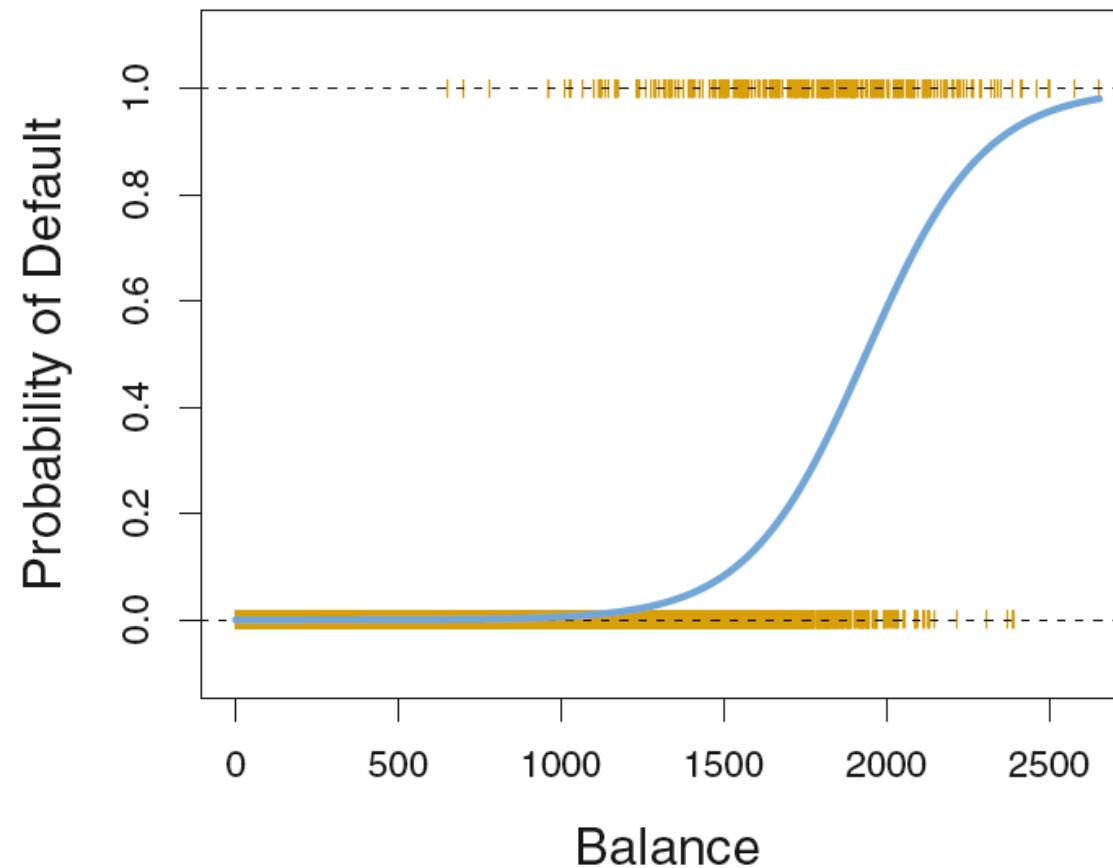


Сигмоида

# Сигмоида



# Логистическая регрессия



# Логистическая регрессия

- Линейная модель классификации:  $a(x) = \text{sign} \langle w, x \rangle$
- Позволяет оценивать вероятности:  $\pi(x) = \sigma(\langle w, x \rangle)$
- Обучение: градиентный спуск

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

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

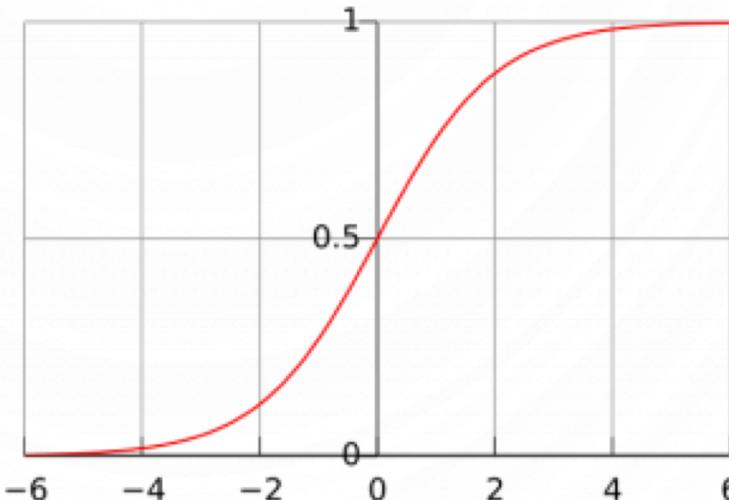
где  $\sigma(z) = \frac{1}{1+e^{-z}}$  - сигмоида (логистическая функция)

# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

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

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



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

# ВЕРОЯТНОСТНЫЙ СМЫСЛ

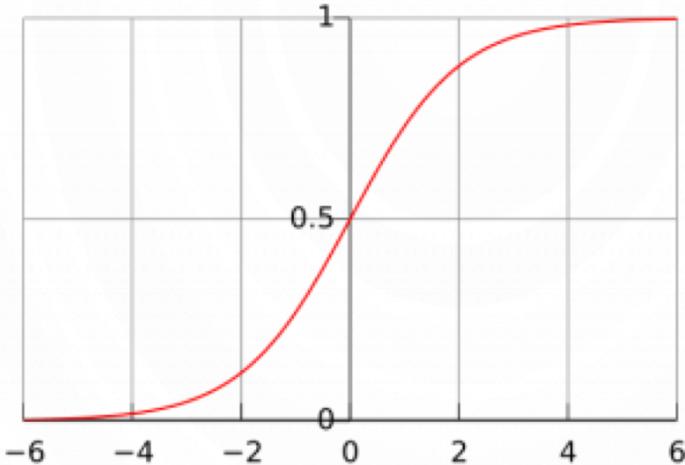
**Утверждение.**  $a(x, w)$  – вероятность того, что  $y = +1$  на объекте  $x$ , т.е.

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

**Доказательство.** Дальше в лекции.

# РАЗДЕЛЯЮЩАЯ ГРАНИЦА

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



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

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.

# 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:

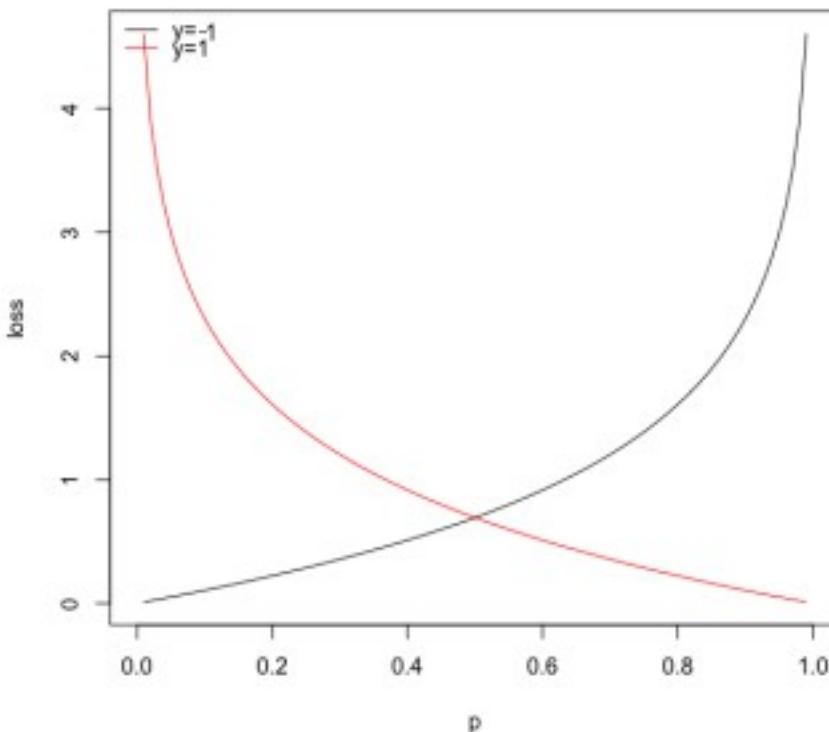
- $Q(a, X) = \frac{1}{l} \sum_{i=1}^l \left( \frac{1}{1+e^{-w^T x_i}} - y_i \right)^2$  - не выпуклая функция  
(можем не попасть в глобальный минимум при оптимизации)
- На совсем неправильном предсказании маленький штраф  
(пусть предсказали вероятность 0% на объекте класса  $y = +1$ , тогда штраф всего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

# ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

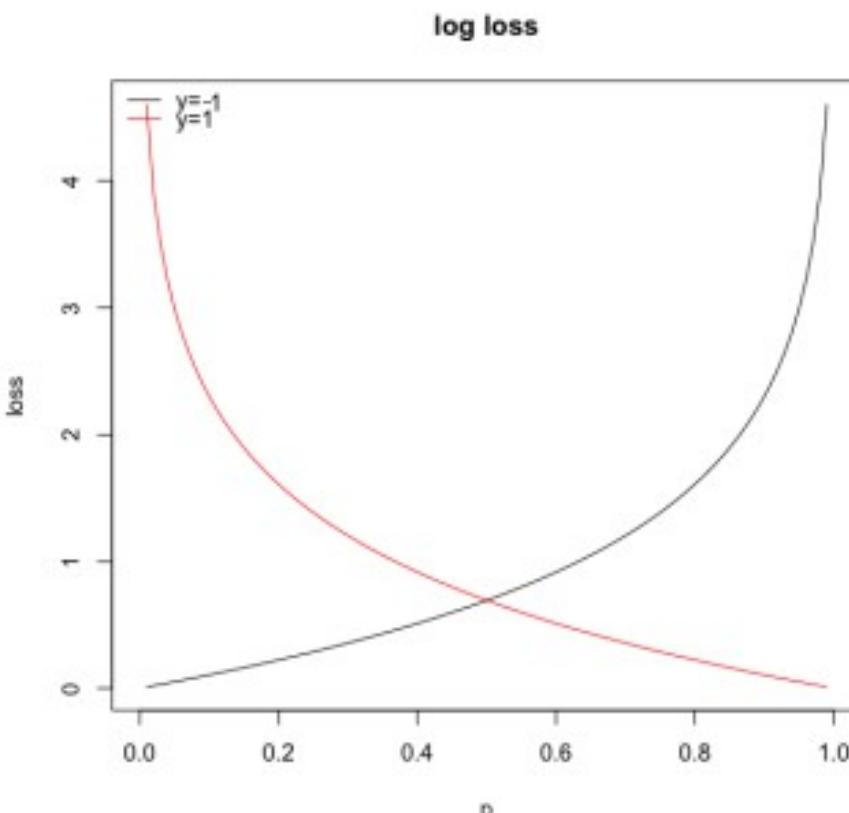
Возьмем логистическую функцию потерь (**log-loss**):

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

log loss



# 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  $L(a, y) \rightarrow +\infty$

# 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**:

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

# ФУНКЦИИ ПОТЕРЬ

Подходят:

• Квадратичная

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

• Логистическая (log-loss)

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

Не подходят:

• Модуль

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

# 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 $b(x)$

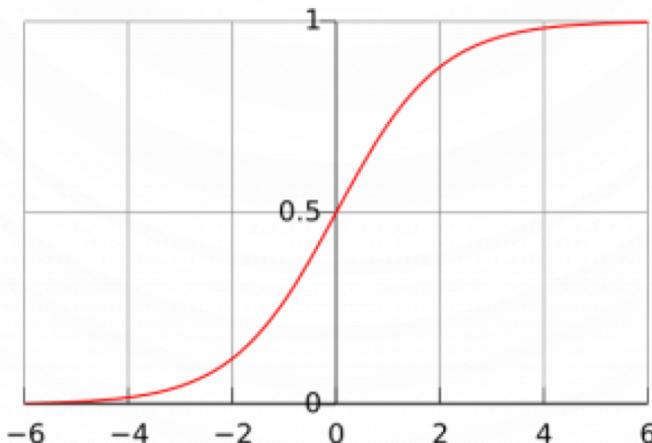
- Хотим, чтобы алгоритм  $b(x)$  возвращал числа из отрезка  $[0, 1]$ .

# 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**:  $\sigma(z) = \frac{1}{1+e^{-z}}$



# 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  $(w, x) = w^T x$ :

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

# 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

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

**Идея доказательства:**

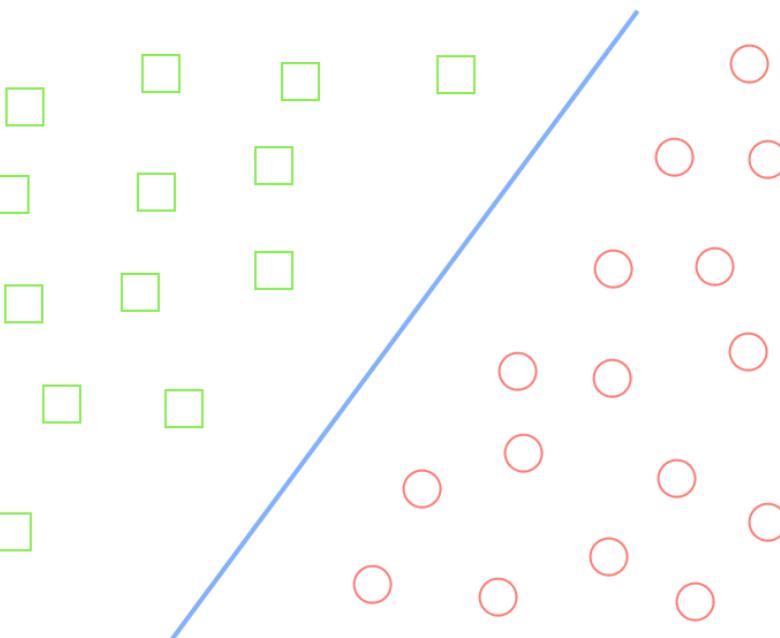
Подставляем явный вид сигмоиды в логарифмическую функцию потерь:

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

# МЕТОД ОПОРНЫХ ВЕКТОРОВ

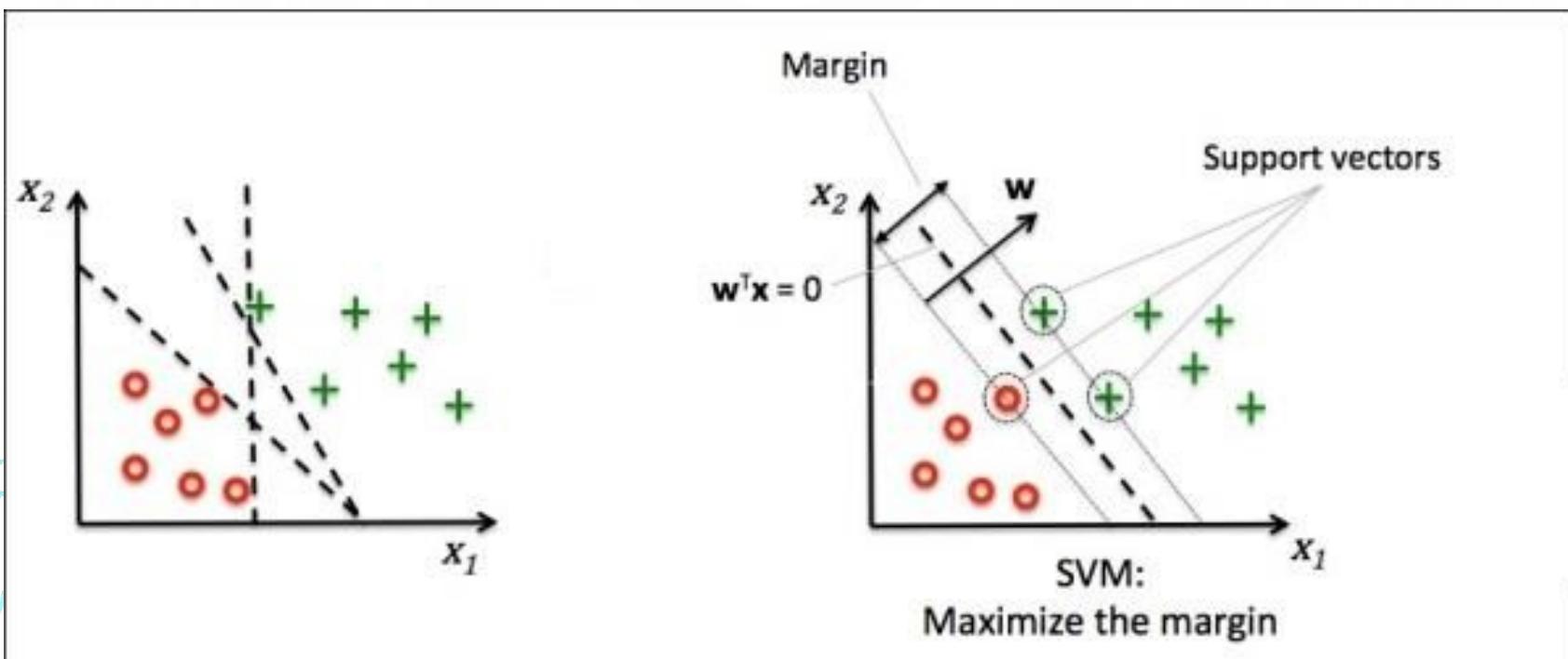
# ЛИНЕЙНО РАЗДЕЛИМАЯ ВЫБОРКА

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



# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

Цель метода опорных векторов (Support Vector Machine) –  
максимизировать ширину разделяющей полосы.



# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

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

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

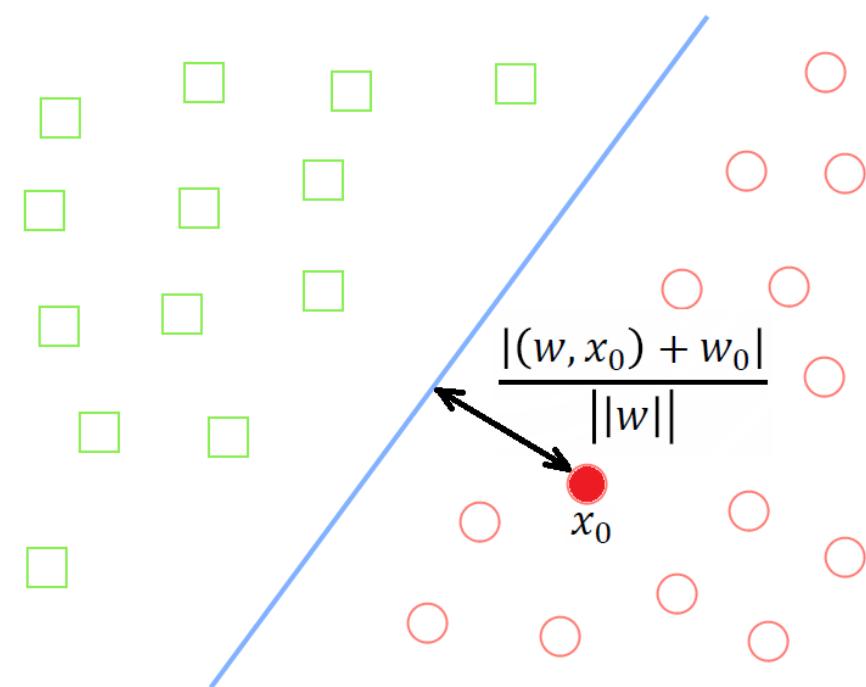
# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

- $a(x) = \text{sign}((w, x) + w_0)$
- 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:

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



# МЕТОД ОПОРНЫХ ВЕКТОРОВ: РАЗДЕЛИМЫЙ СЛУЧАЙ

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

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

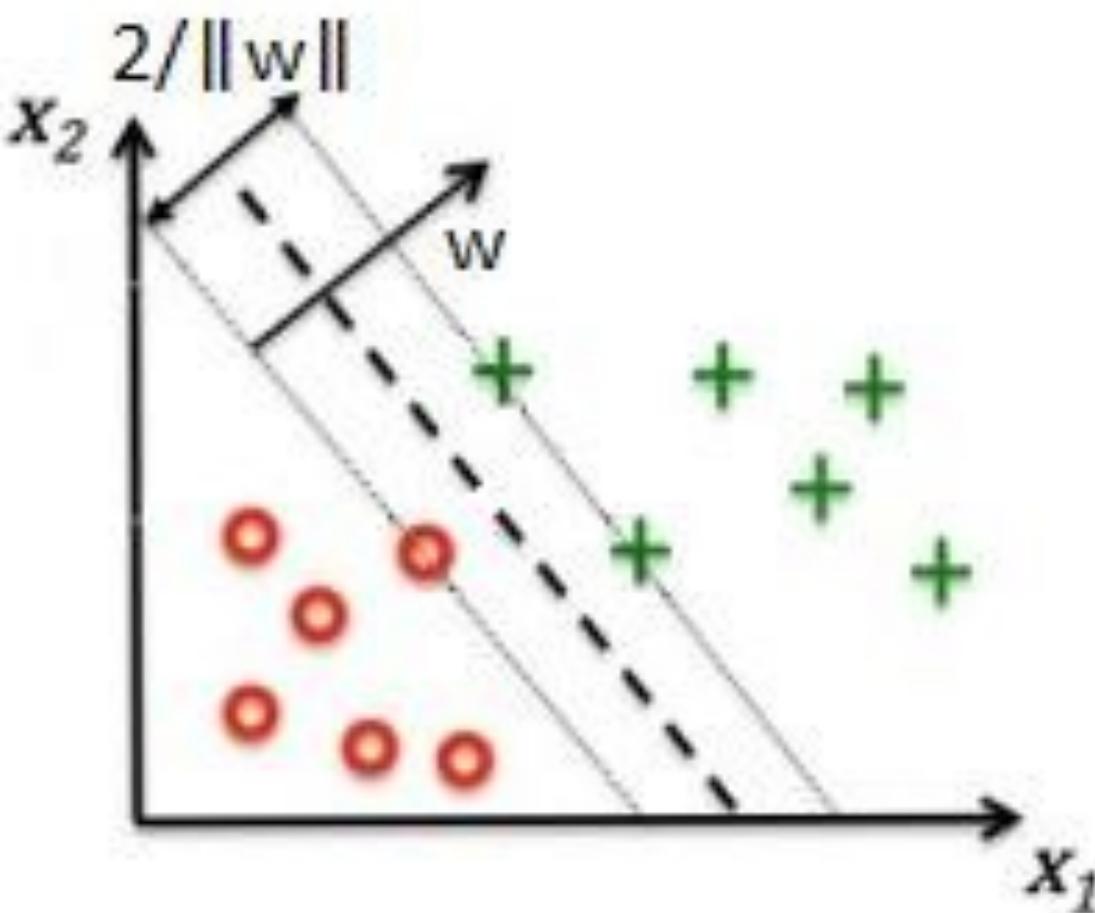
Тогда расстояние от точки  $x_0$  до разделяющей гиперплоскости, задаваемой классификатором:

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

- 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



# ОПТИМИЗАЦИОННАЯ ЗАДАЧА SVM ДЛЯ РАЗДЕЛИМОЙ ВЫБОРКИ

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

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

## ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

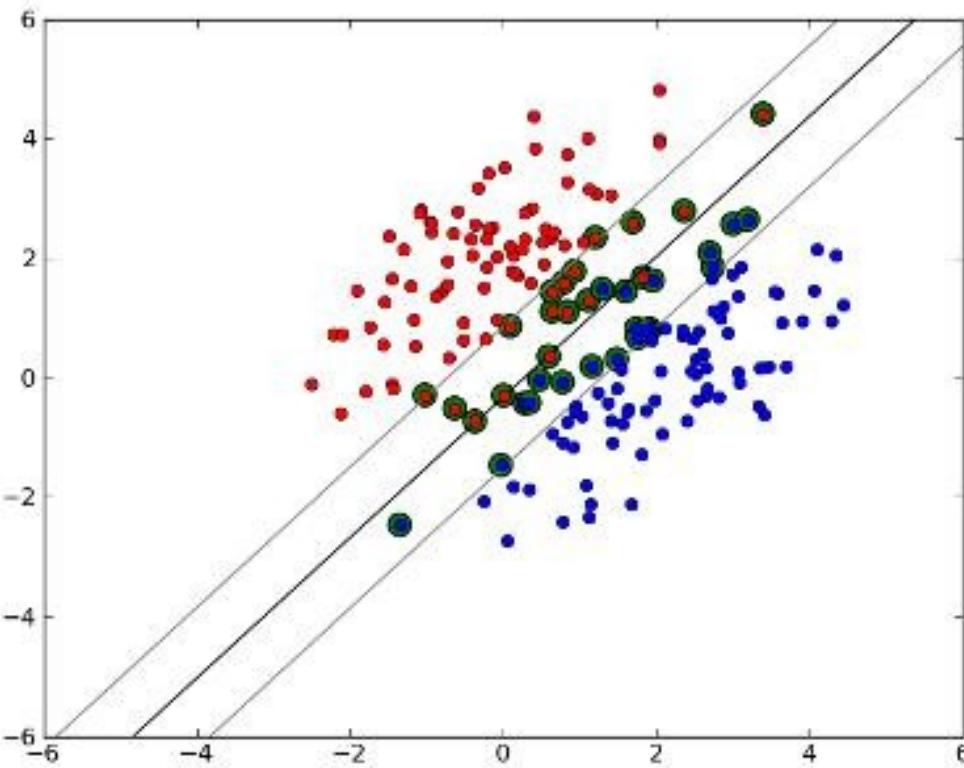
- Существует хотя бы один объект  $x \in X$ , что

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

# 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 штраф  $\sum \xi_i$
- 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:

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

# 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):

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

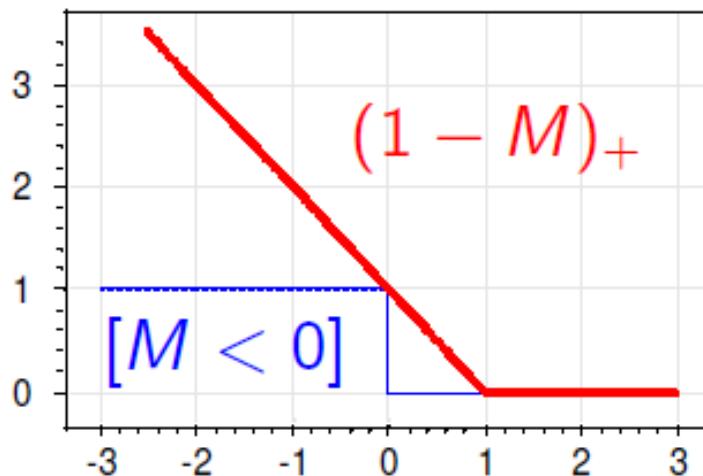
Получаем безусловную задачу оптимизации:

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

# 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:

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

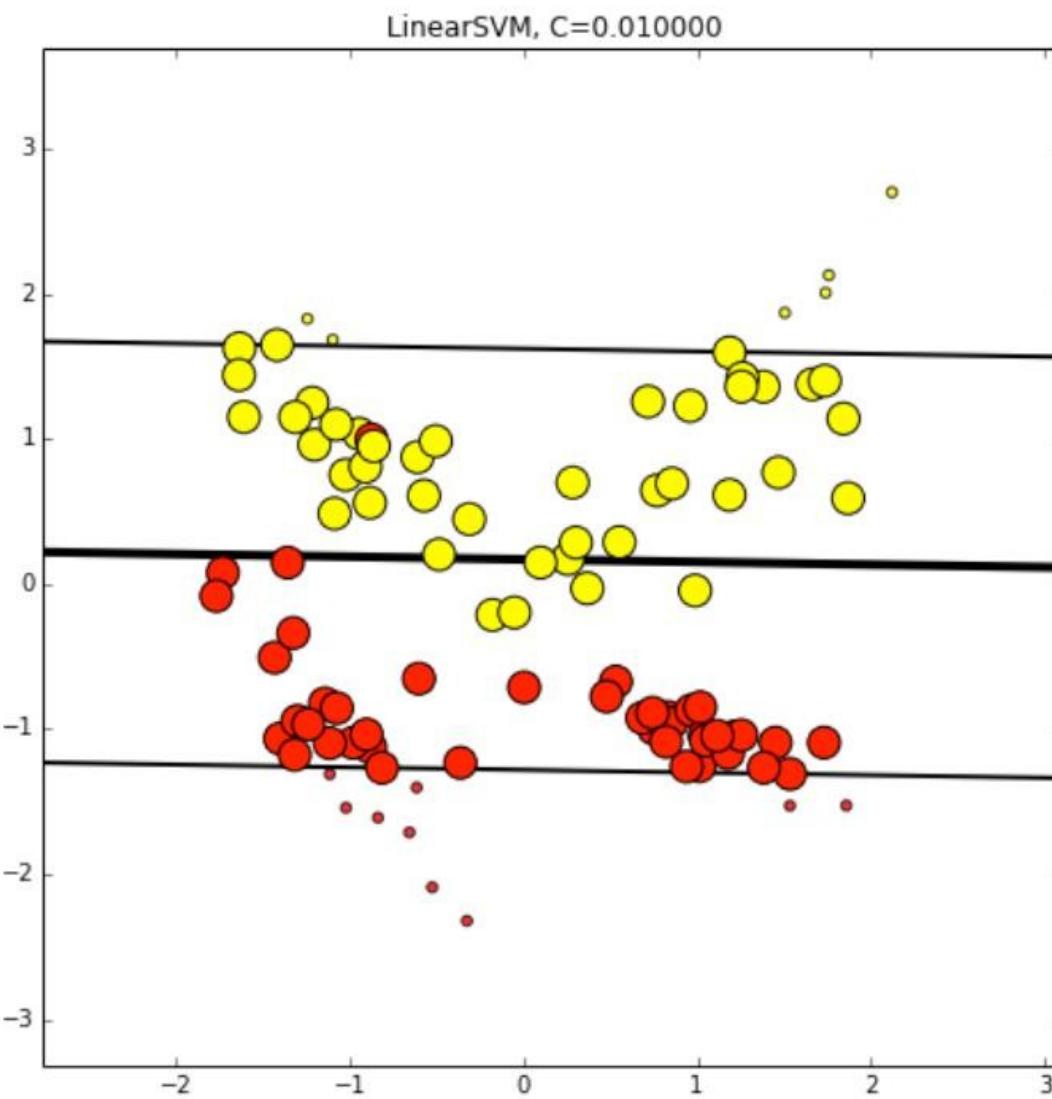


# ЗНАЧЕНИЕ КОНСТАНТЫ С

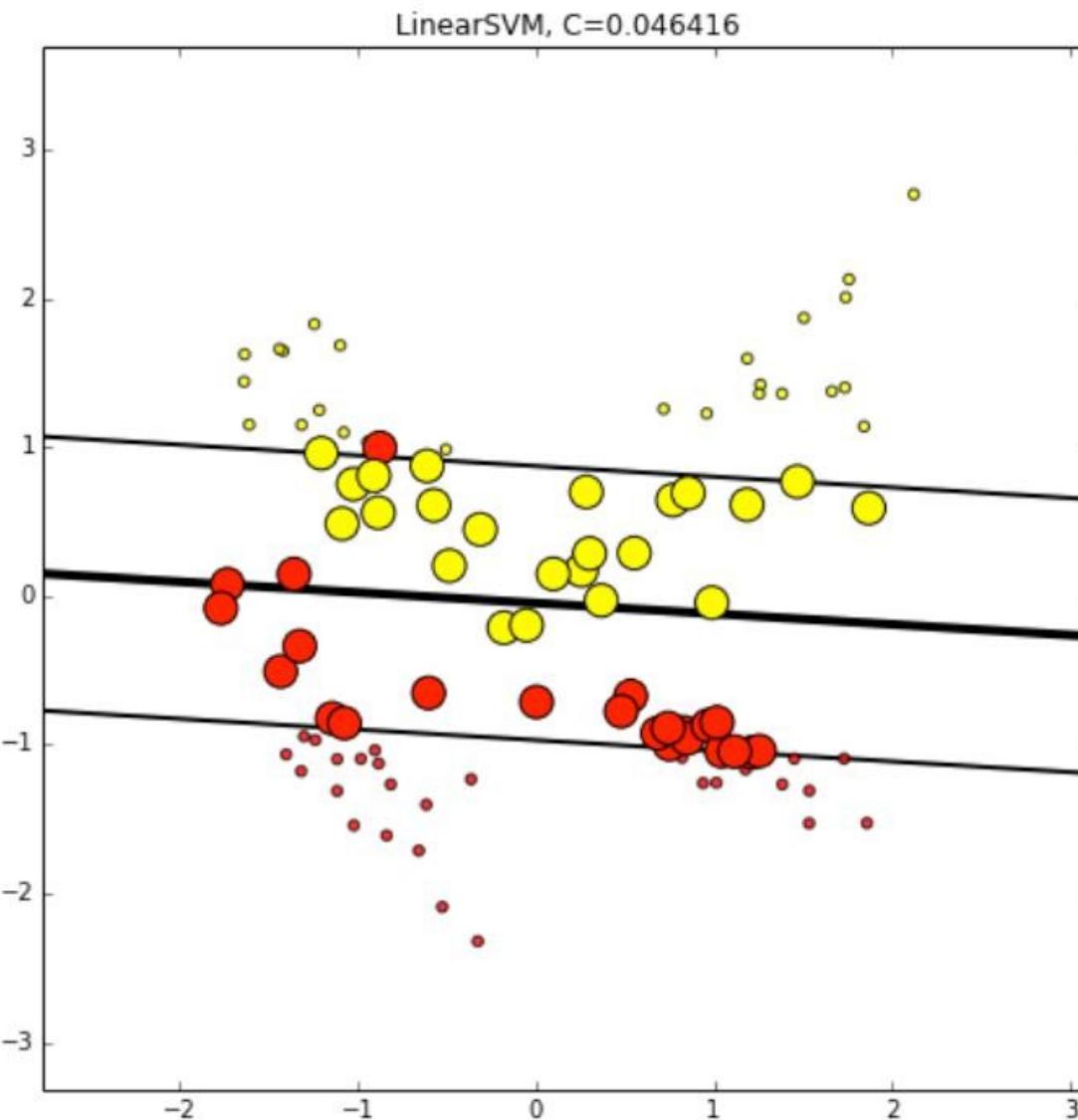
$$\begin{cases} \frac{1}{2} \|w\|^2 + \textcolor{red}{C} \sum_{i=1}^l \xi_i \rightarrow \min_{w, w_0, \xi_i} (1) \\ y_i((w, x_i) + w_0) \geq 1 - \xi_i, i = 1, \dots, l (2) \\ \xi_i \geq 0, i = 1, \dots, l (3) \end{cases}$$

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

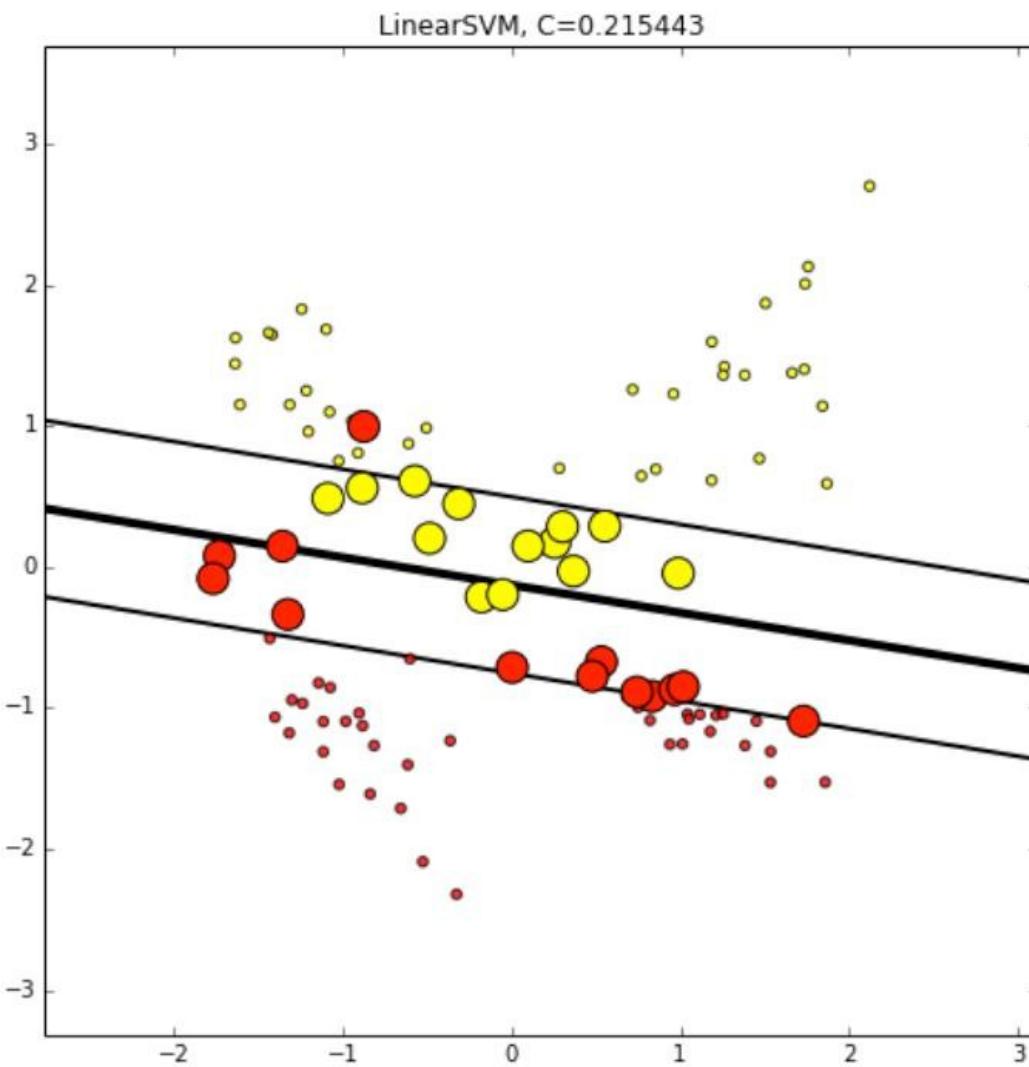
# ЗНАЧЕНИЕ КОНСТАНТЫ С



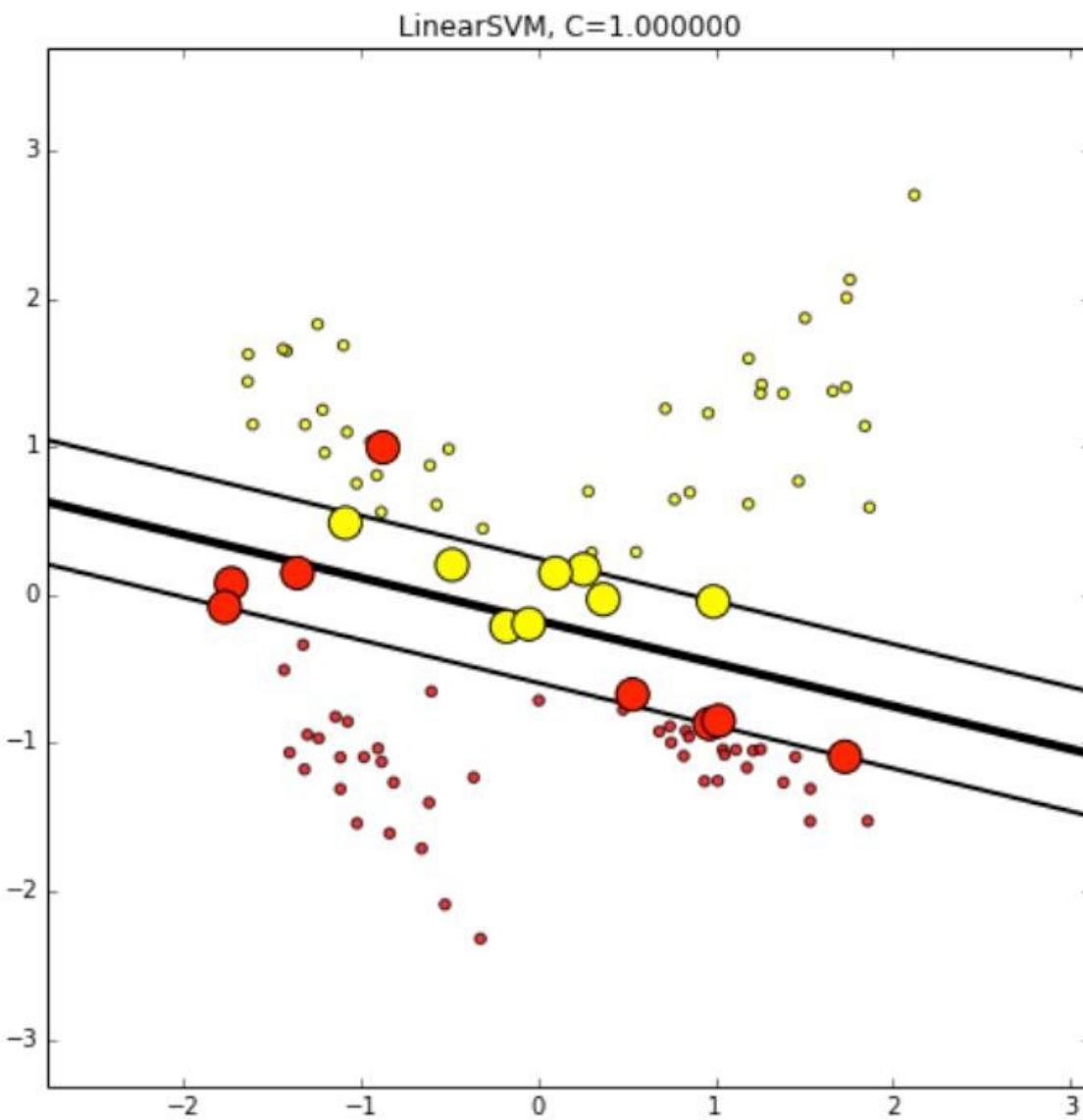
# ЗНАЧЕНИЕ КОНСТАНТЫ С



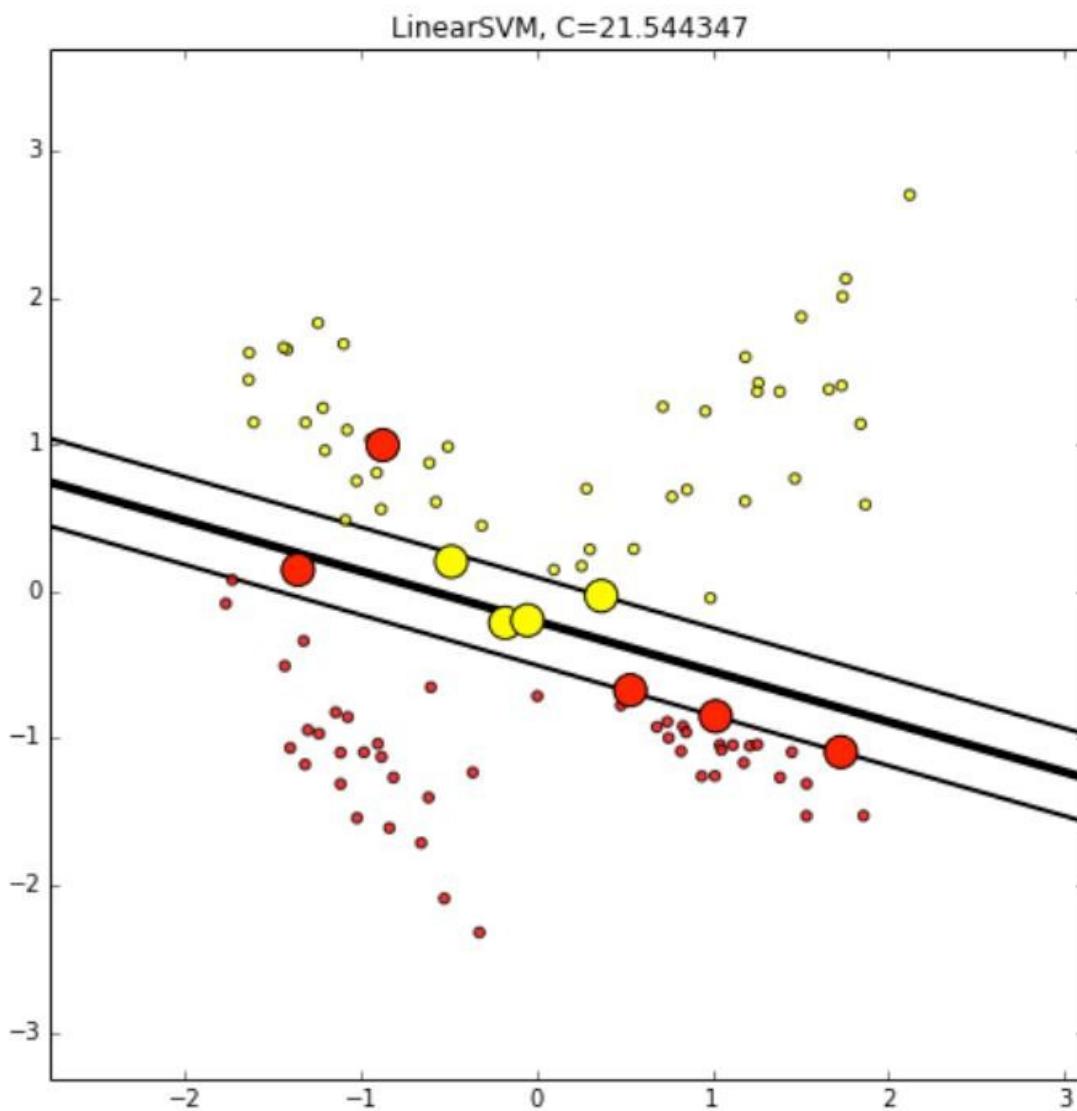
# ЗНАЧЕНИЕ КОНСТАНТЫ С



# ЗНАЧЕНИЕ КОНСТАНТЫ С



# ЗНАЧЕНИЕ КОНСТАНТЫ С



# ТИПЫ ОБЪЕКТОВ В SVM

