# Лекция 5
# Линейные (и не совсем) модели классификации. Часть 2.

Марк Блуменау

На основе материалов Кантонистовой Е.О.

ВШЭ, 2025

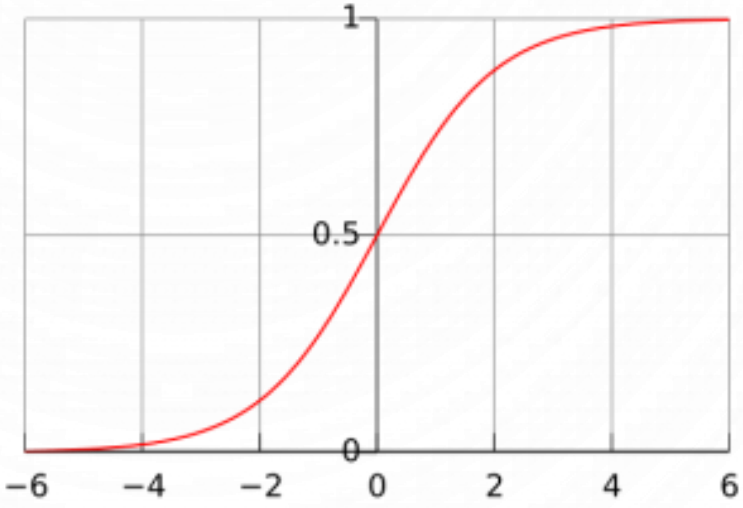
# ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Хотим предсказывать не классы, а вероятности классов.

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

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

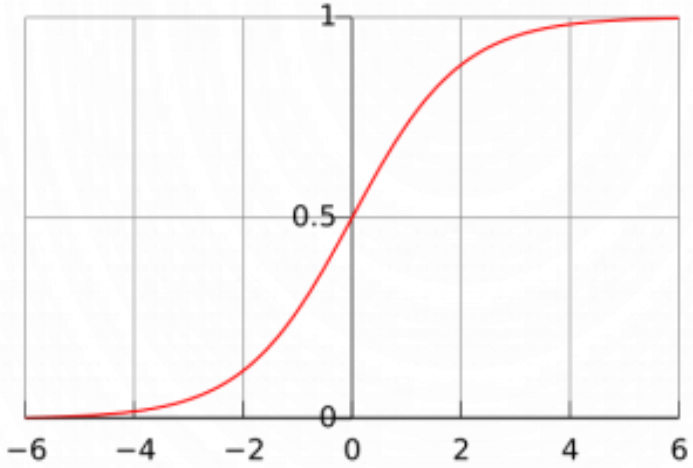
$\sigma(z) \in (0; 1)$ .




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

# РАЗДЕЛЯЮЩАЯ ГРАНИЦА

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




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

Получаем, что

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

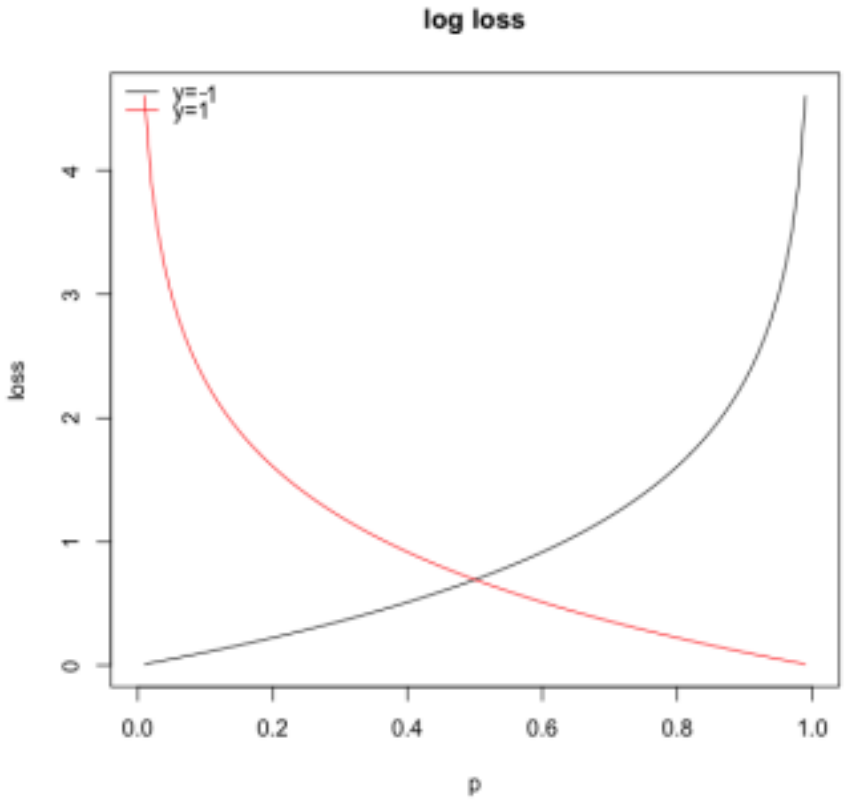
т.е. $w^T x = 0$ – разделяющая гиперплоскость.

**Логистическая регрессия - это линейный классификатор!**

# ФУНКЦИЯ ПОТЕРЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

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

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



# 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:

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

*По закону больших чисел* при $n \to \infty$ получаем

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

Отсюда получаем ***условие на функцию потерь***:

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

# ЗАКОН БОЛЬШИХ ЧИСЕЛ

- Закон больших чисел (ЗБЧ) в теории вероятностей — принцип, описывающий результат выполнения одного и того же эксперимента много раз. Согласно закону, среднее значение конечной выборки из фиксированного распределения близко к математическому ожиданию этого распределения.

# ПРАВДОПОДОБИЕ И LOG-LOSS

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

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

Правдоподобие выборки:

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

# ПРАВДОПОДОБИЕ?

- Правдоподобие (likelihood function) — это вероятность получить наблюдаемую выборку при конкретном значении параметра.
- Оценка максимального правдоподобия — значение параметра, которое максимизирует правдоподобие.
- Если вероятность позволяет нам предсказывать неизвестные результаты, основанные на известных параметрах, то правдоподобие позволяет нам оценивать неизвестные параметры, основанные на известных результатах.

«Какова вероятность выпадения 12 очков в каждом из ста бросков двух костей?»

«Насколько правдоподобно, что кости честно, если из ста бросков в каждом выпало 12 очков?»

# ФУНКЦИЯ ПОТЕРЬ ДЛЯ ОБУЧЕНИЯ

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

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

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

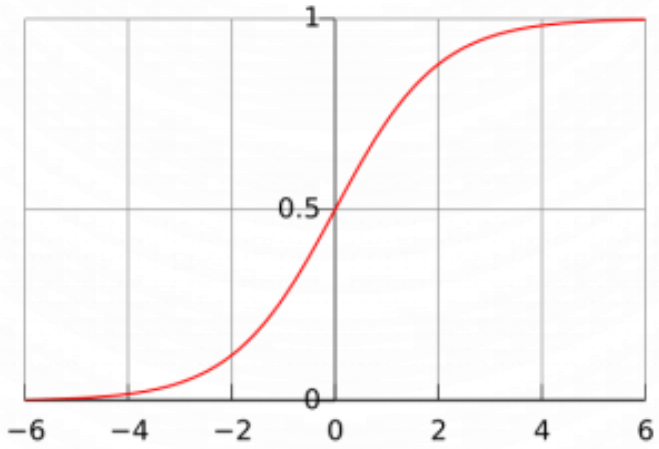
**Это log-loss!**

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

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

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

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

- 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

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

**Идея доказательства:**

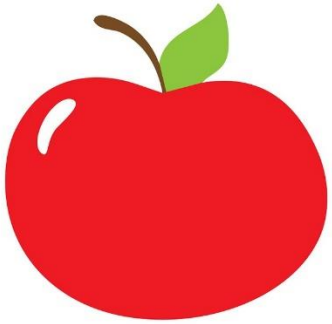
Подставляем явный вид сигмоиды в логарифмическую функцию потерь:

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

# НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

# НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР

**Наивный байесовский классификатор –** это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков.



Пример: фрукт может считаться яблоком, если:

1) он красный
2) круглый
3) его диаметр составляет порядка 8 см

Предполагаем, что признаки вносят независимый вклад в вероятность того, что фрукт является яблоком.

# ТЕОРЕМА БАЙЕСА

Теорема Байеса:

$$\boldsymbol{P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}}$$

- $\boldsymbol{P(c|x)}$ - вероятность того,

что объект со значением признака $x$

принадлежит классу $c$.

- $\boldsymbol{P(c)}$ – априорная вероятность класса $c$.
- $\boldsymbol{P(x|c)}$ - вероятность того, что значение признака равно $x$ при условии, что объект принадлежит классу $c$.
- $\boldsymbol{P(x)}$ – априорная вероятность значения признака $x$.

# ПРИМЕР РАБОТЫ БАЙЕСОВСКОГО АЛГОРИТМА

Пример: на основе данных о погодных условиях необходимо определить, состоится ли матч.

| Weather | Play |
|---|---|
| Sunny | No |
| Overcast | Yes |
| Rainy | Yes |
| Sunny | Yes |
| Sunny | Yes |
| Overcast | Yes |
| Rainy | No |
| Rainy | No |
| Sunny | Yes |
| Rainy | Yes |
| Sunny | No |
| Overcast | Yes |
| Overcast | Yes |
| Rainy | No |

- Преобразуем набор данных в следующую таблицу:

| Weather | No | Yes |
|---|---|---|
| Overcast | 0 | 4 |
| Rainy | 3 | 2 |
| Sunny | 2 | 3 |
| Grand Total | 5 | 9 |

# ПРИМЕР РАБОТЫ БАЙЕСОВСКОГО АЛГОРИТМА

Решим задачу с помощью теоремы Байеса:

$$P(Yes|Sunny) = P(Sunny|Yes) \cdot P(Yes)/P(Sunny)$$

| Таблица частот | | | | |
|---|---|---|---|---|
| **Weather** | **No** | **Yes** | | |
| Overcast | 0 | 4 | **=4/14** | **0.29** |
| Rainy | 3 | 2 | **=5/14** | **0.36** |
| Sunny | 2 | 3 | **=5/14** | **0.36** |
| Grand Total | 5 | 9 | | |
| | **=5/14** | **=9/14** | | |
| | **0.36** | **0.64** | | |

- $P(Sunny|Yes) = \frac{3}{9}, P(Sunny) = \frac{5}{14}, P(Yes) = \frac{9}{14}.$
- $P(Yes|Sunny) = \frac{3}{9} \cdot \frac{9}{14} : \frac{5}{14} = \frac{3}{5} = 0{,}6 \Rightarrow 60\%.$

# В СЛУЧАЕ НЕСКОЛЬКИХ ПРИЗНАКОВ

Пусть $x_1, \dots, x_n$ - признаки объекта, $y$ – целевая переменная.

Тогда теорема Байеса записывается в виде

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y) \dots P(x_n|y)P(y)}{P(x_1)P(x_2) \dots P(x_n)}.$$

Вероятности в правой части формулы вычисляются с помощью частотных таблиц, как и в одномерном случае.

# БАЙЕСОВСКИЙ АЛГОРИТМ ДЛЯ КЛАССИФИКАЦИИ

Плюсы и минусы:

+ классификация быстрая и простая

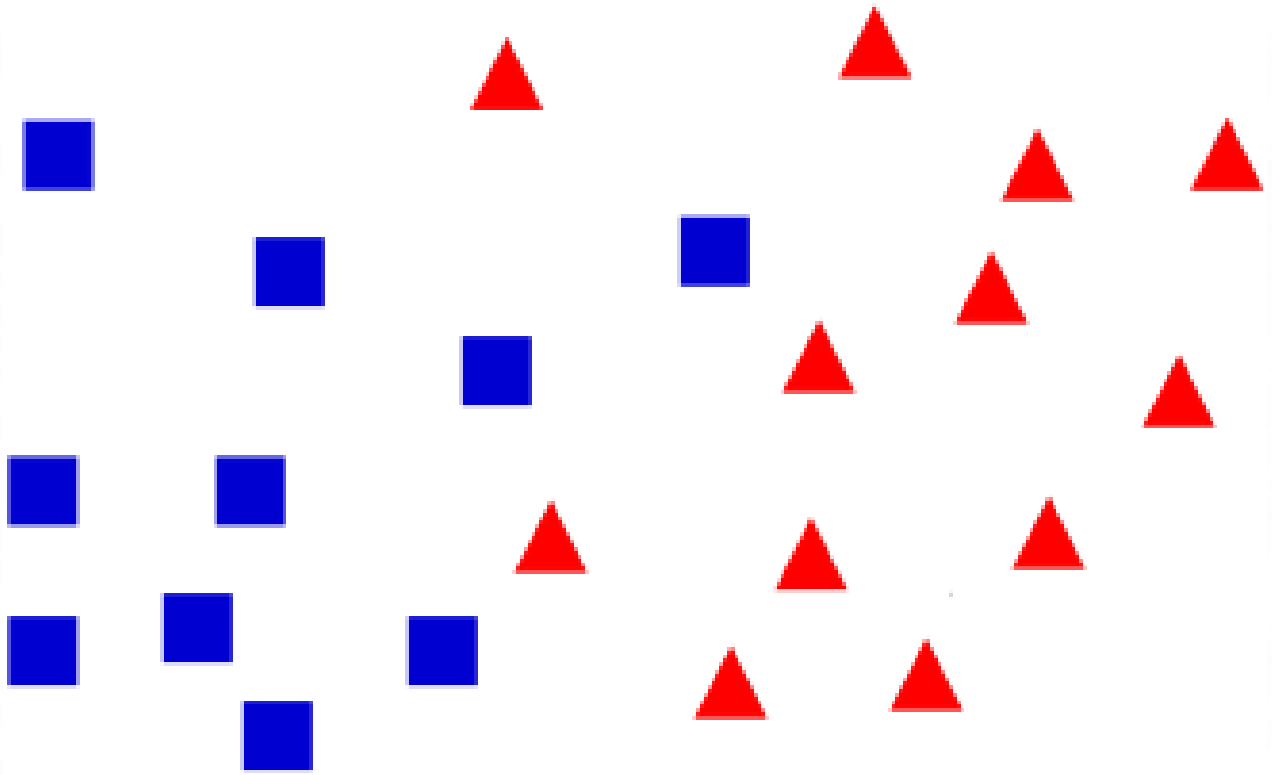
+ в случае, если выполняется предположение о независимости, классификатор показывает очень высокое качество

- если в тестовых данных присутствует категория, не встречавшаяся в данных для обучения, модель присвоит ей нулевую вероятность

# МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

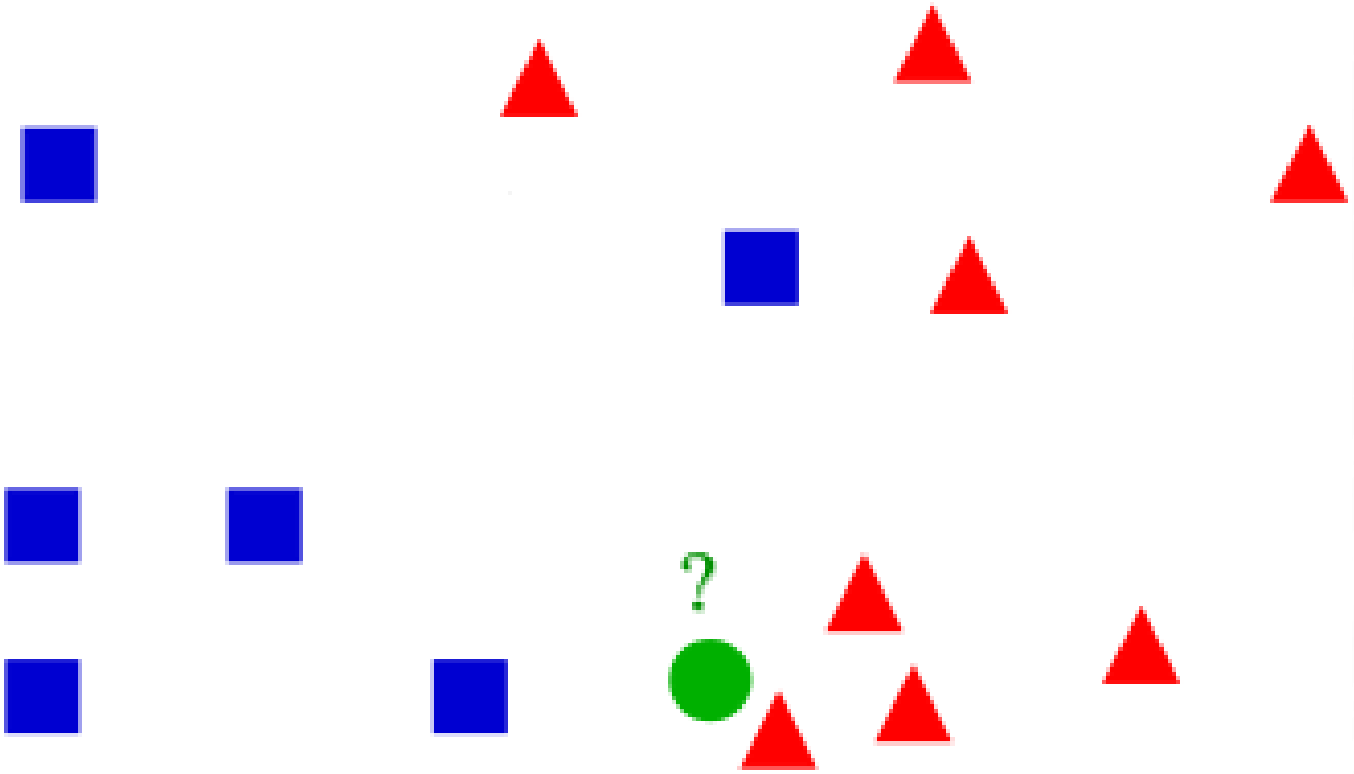
# МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

**Идея:** схожие объекты находятся близко друг к другу в пространстве признаков.



# МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

*Как классифицировать новый объект?*



# МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

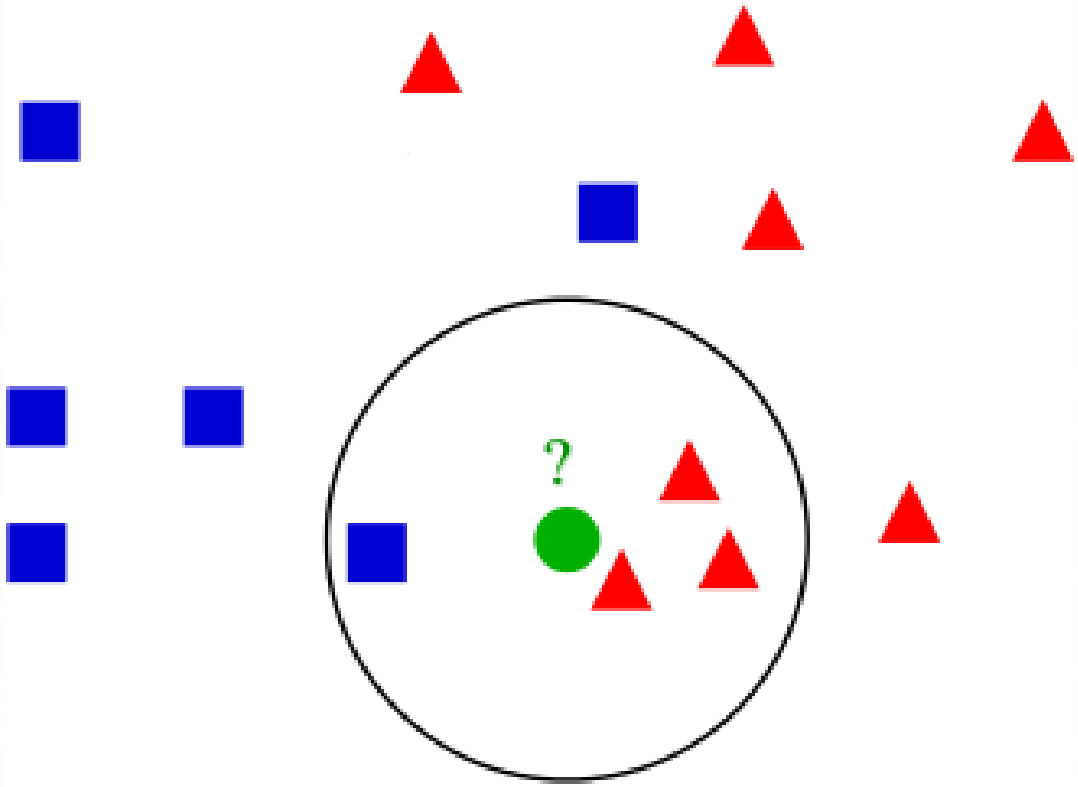
Чтобы классифицировать новый объект, нужно:

- Вычислить расстояние до каждого из объектов обучающей выборки.
- Выбрать k объектов обучающей выборки, расстояние до которых минимально.
- Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей.

# МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

*Число ближайших соседей k – гиперпараметр метода.*

Например, для $k = 4$ получим:




То есть объект будет отнесён к классу *треугольников*.

# ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть $k$ – количество соседей. Для каждого объекта $u$ возьмём $k$ ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта $u$ определяется следующим образом:

$$\boldsymbol{a(u) = \operatorname*{argmax}_{y \in Y} \sum_{i=1}^{k} [y(x_{(i;u)}) = y]}.$$

# ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть $k$ – количество соседей. Для каждого объекта $u$ возьмём $k$ ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта $u$ определяется следующим образом:

$$a(u) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} [y(x_{(i;u)}) = y].$$

*Ближайшие объекты* – это объекты, расстояние от которых до данного объекта наименьшее по некоторой метрике $\rho$.

# ФОРМАЛИЗАЦИЯ МЕТОДА

Пусть $k$ – количество соседей. Для каждого объекта $u$ возьмём $k$ ближайших к нему объектов из тренировочной выборки:

$$x_{(1;u)}, x_{(2;u)}, \dots, x_{(k;u)}.$$

Тогда класс объекта $u$ определяется следующим образом:

$$a(u) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} [y(x_{(i;u)}) = y].$$

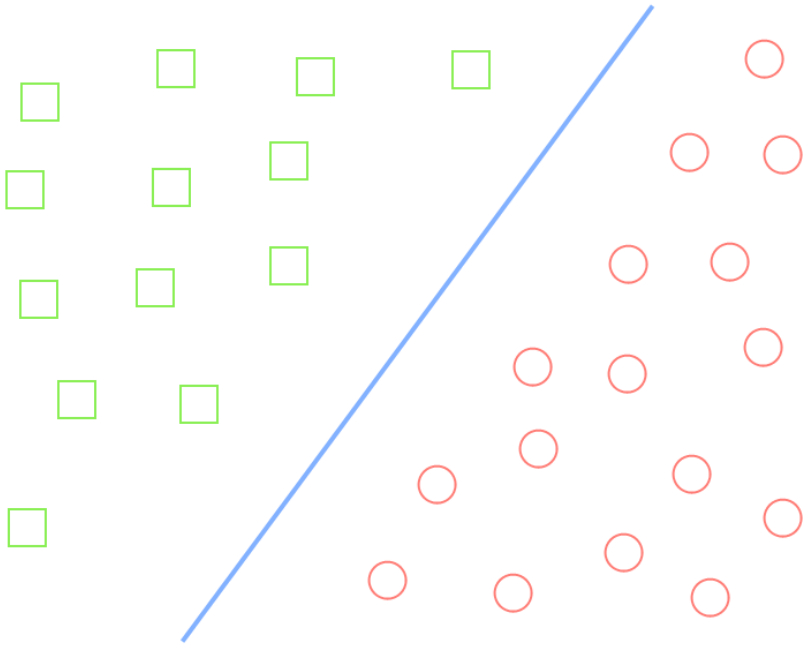
*Ближайшие объекты* – это объекты, расстояние от которых до данного объекта наименьшее по некоторой метрике $\rho$.

- В качестве метрики $\rho$ как правило используют евклидово расстояние, но можно использовать и другие метрики.
- Перед использованием метода необходимо масштабировать данные, иначе признаки с большими числовыми значениями будут доминировать при вычислении расстояний.

# МЕТОД ОПОРНЫХ ВЕКТОРОВ

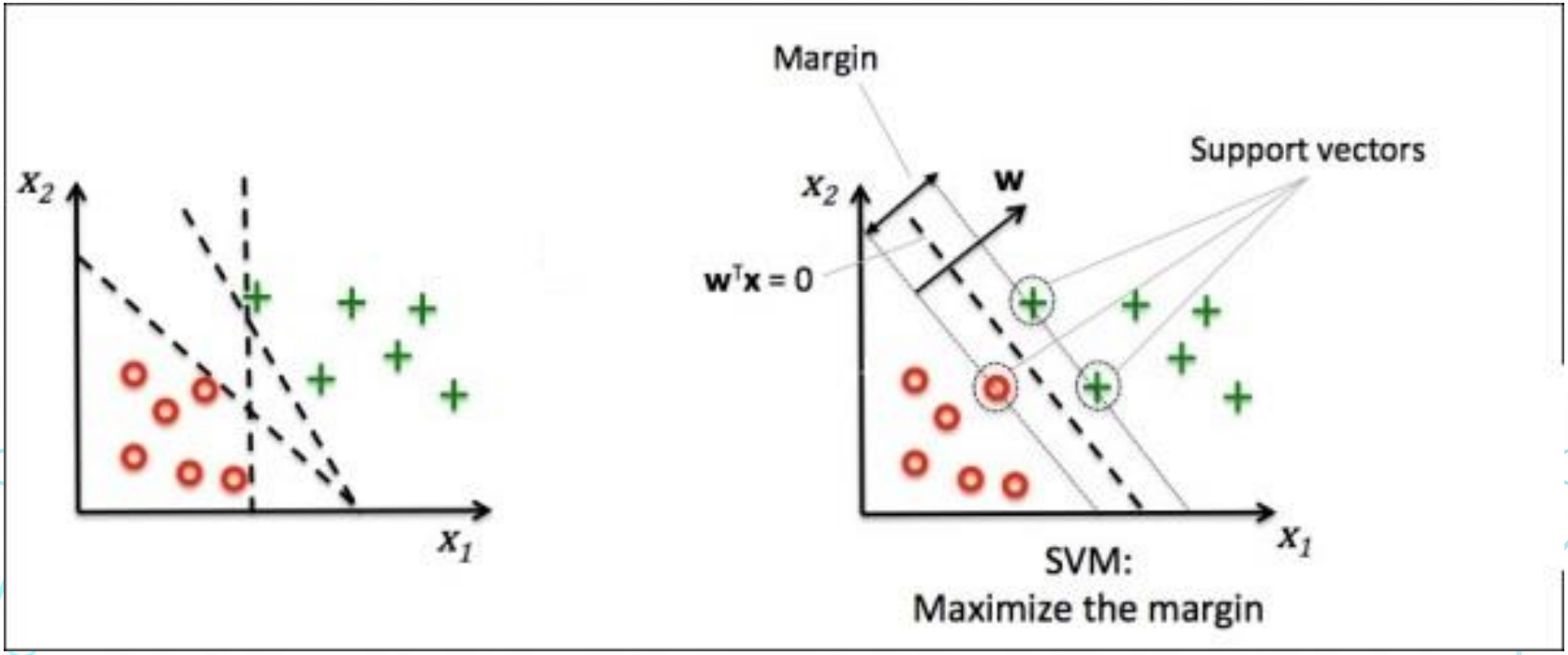
# ЛИНЕЙНО РАЗДЕЛИМАЯ ВЫБОРКА

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

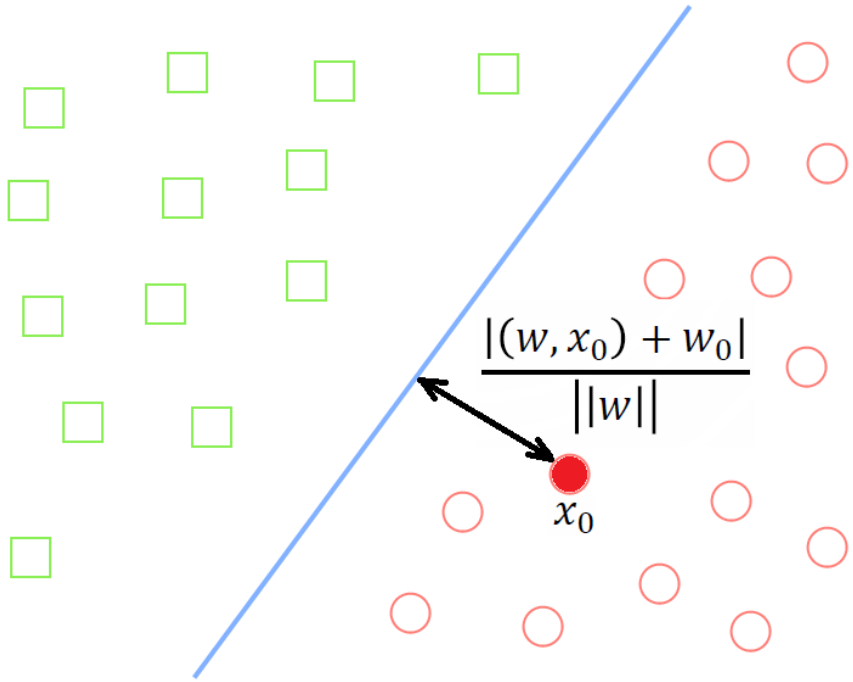
- $a(x) = sign\big((w, x) + w_0\big)$
- Нормируем параметры $w$ и $w_0$ так, что

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

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

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

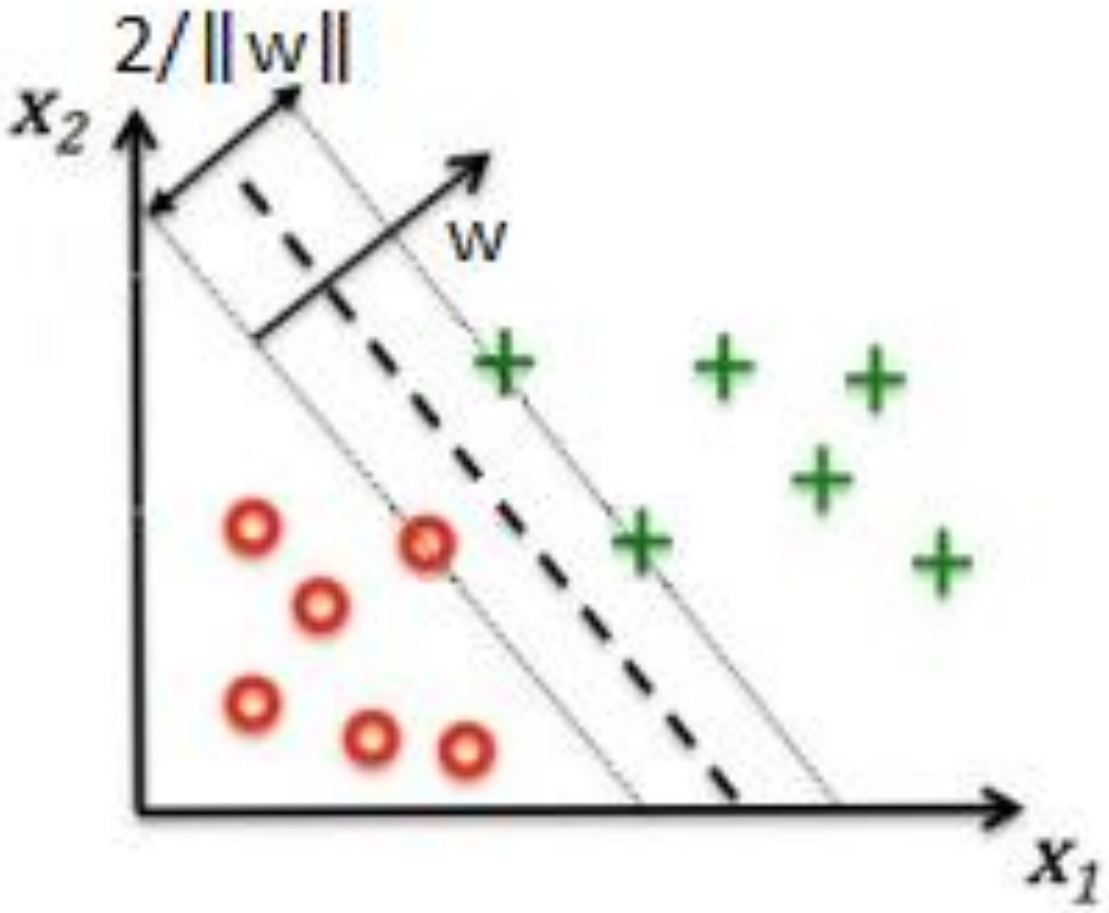
Тогда расстояние от точки $x_0$ до разделяющей гиперплоскости, задаваемой классификатором:

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

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

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

# РАЗДЕЛЯЮЩАЯ ПОЛОСА



# ОПТИМИЗАЦИОННАЯ ЗАДАЧА SVM ДЛЯ РАЗДЕЛИМОЙ ВЫБОРКИ

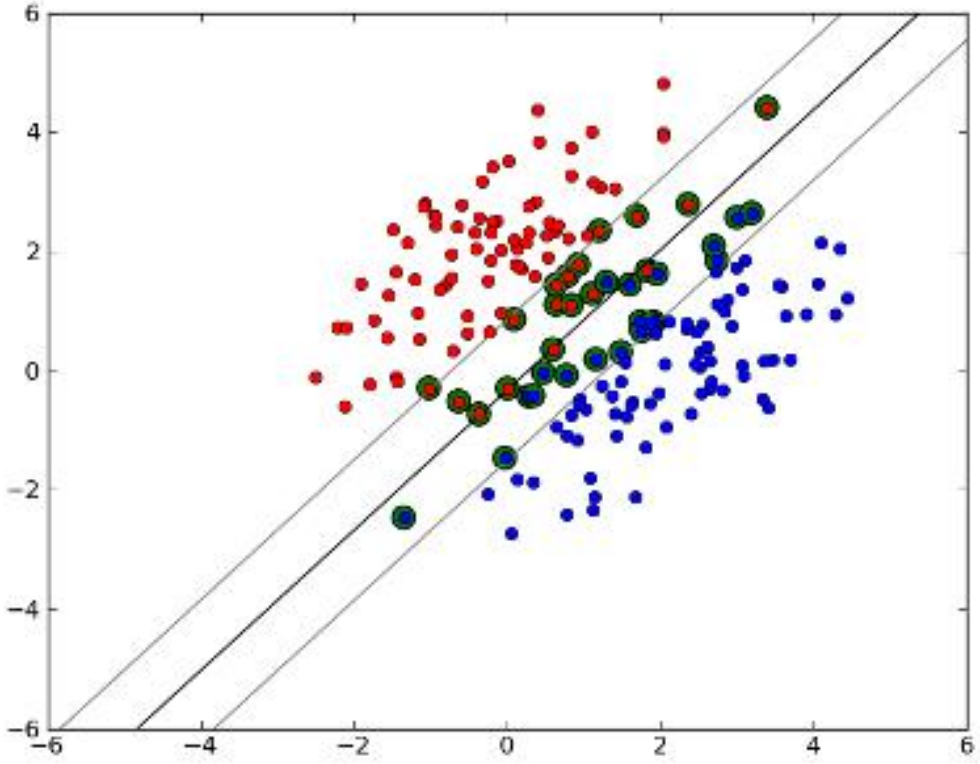
$$\begin{cases} \frac{1}{2}\left|\left|w\right|\right|^2 \to \min\limits_{w} \\ y_i\left((w, x_i) + w_0\right) \geq 1, i = 1, \ldots, l \end{cases}$$

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

# ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

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

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



# ЛИНЕЙНО НЕРАЗДЕЛИМАЯ ВЫБОРКА

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

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

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

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

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

Хотим:

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

Задача оптимизации:

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

# МЕТОД ОПОРНЫХ ВЕКТОРОВ: НЕРАЗДЕЛИМЫЙ СЛУЧАЙ

**Утверждение.** Задача

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

Является выпуклой и имеет единственное решение.

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

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

- Перепишем (2) и (3):

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

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{l}\xi_i \to \min\limits_{w,w_0,\xi_i} (1) \\ y_i\big((w,x_i)+w_0\big) \geq 1-\xi_i, i=1,\dots,l\ (2) \\ \xi_i \geq 0, i=1,\dots,l\ (3) \end{cases}$$

- Перепишем (2) и (3):

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

# СВЕДЕНИЕ К БЕЗУСЛОВНОЙ ЗАДАЧЕ

$$\begin{cases} \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{l}\xi_i \to \min\limits_{w,w_0,\xi_i} (1) \\ y_i\big((w,x_i)+w_0\big) \geq 1-\xi_i, i=1,\dots,l\ (2) \\ \xi_i \geq 0, i=1,\dots,l\ (3) \end{cases}$$

- Перепишем (2) и (3):

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

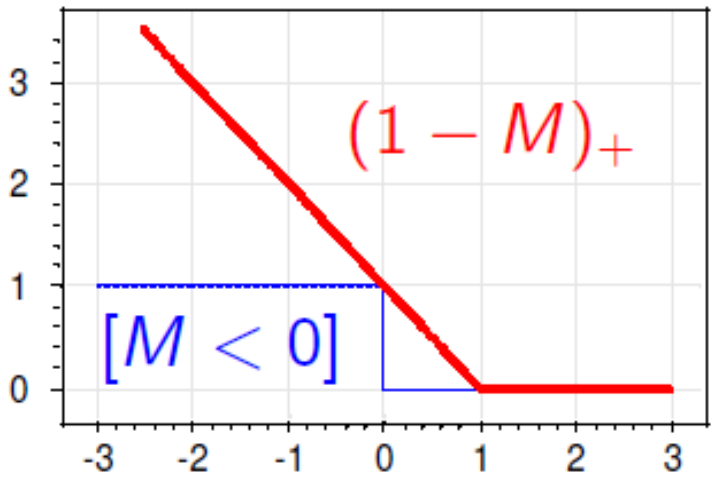
Получаем безусловную задачу оптимизации:

$$\boldsymbol{\frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{l}\max\Big(0, 1 - y_i\big((w,x_i)+w_0\big)\Big) \to \min\limits_{w,w_0}}$$

# 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*:

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

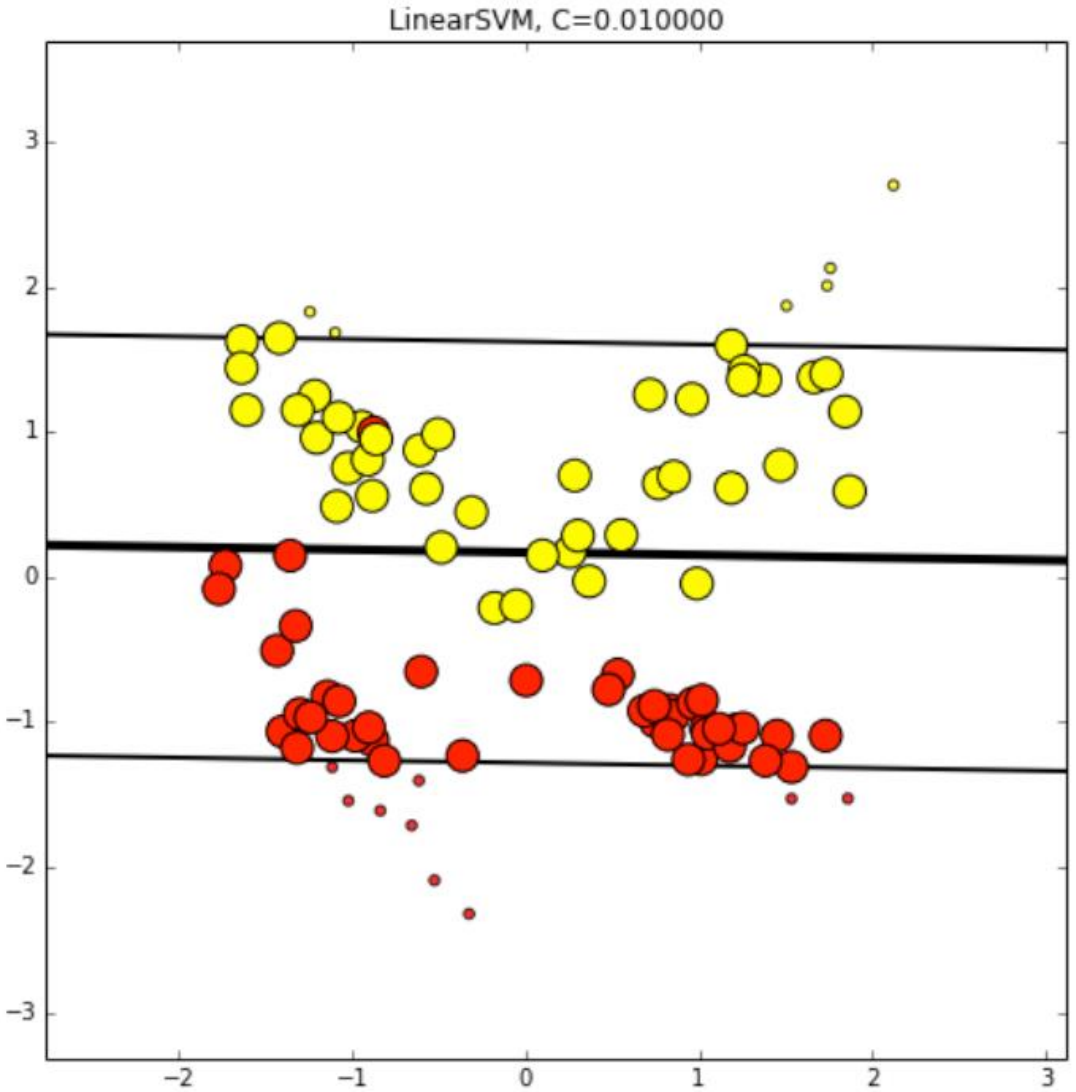


# ЗНАЧЕНИЕ КОНСТАНТЫ C

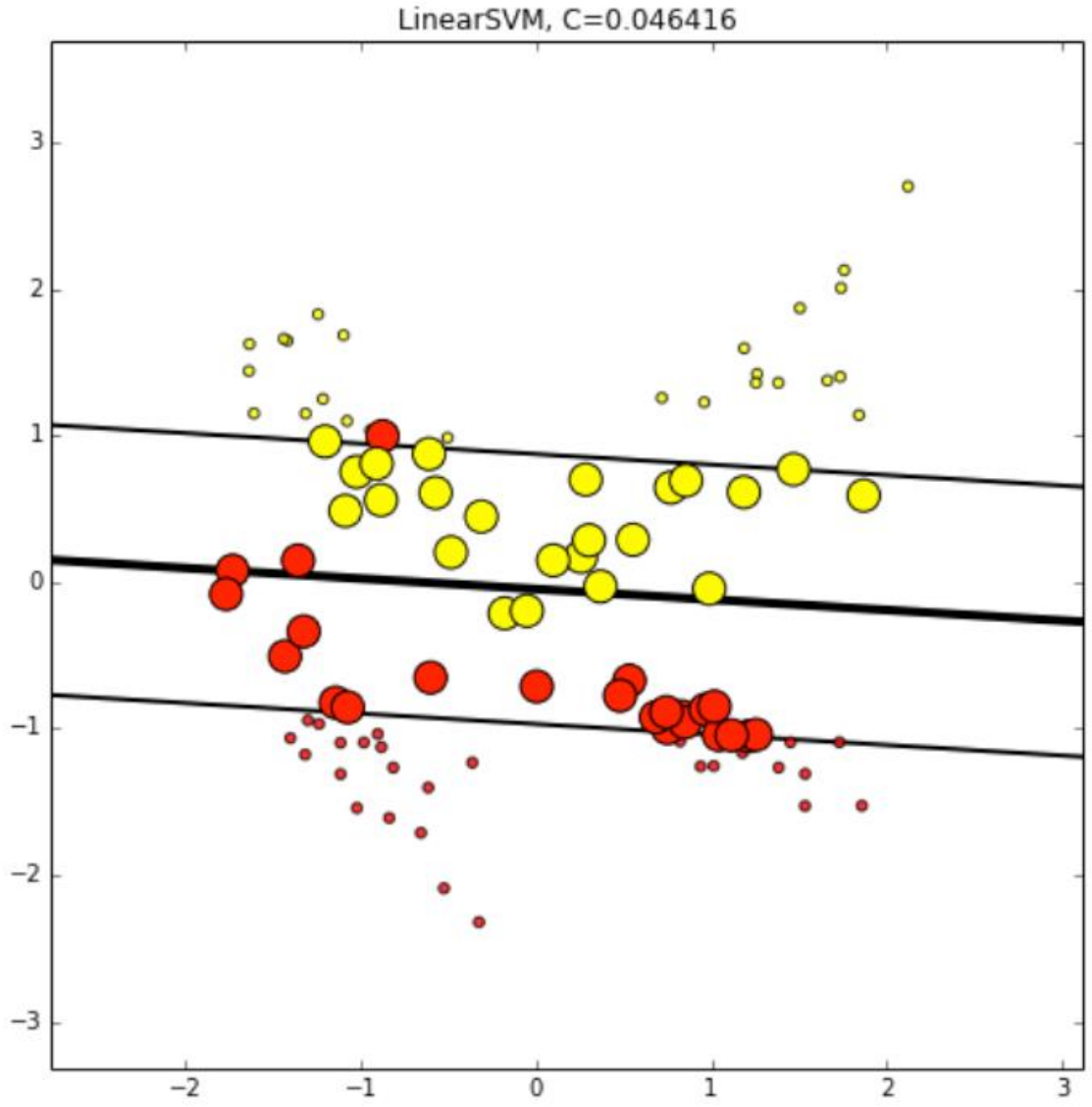
$$\begin{cases} \frac{1}{2}||w||^2 + \boldsymbol{C}\sum_{i=1}^{l}\xi_i \to \min\limits_{w,w_0,\xi_i} (1) \\ y_i\big((w,x_i)+w_0\big) \geq 1-\xi_i, i=1,\dots,l\ (2) \\ \xi_i \geq 0, i=1,\dots,l\ (3) \end{cases}$$

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

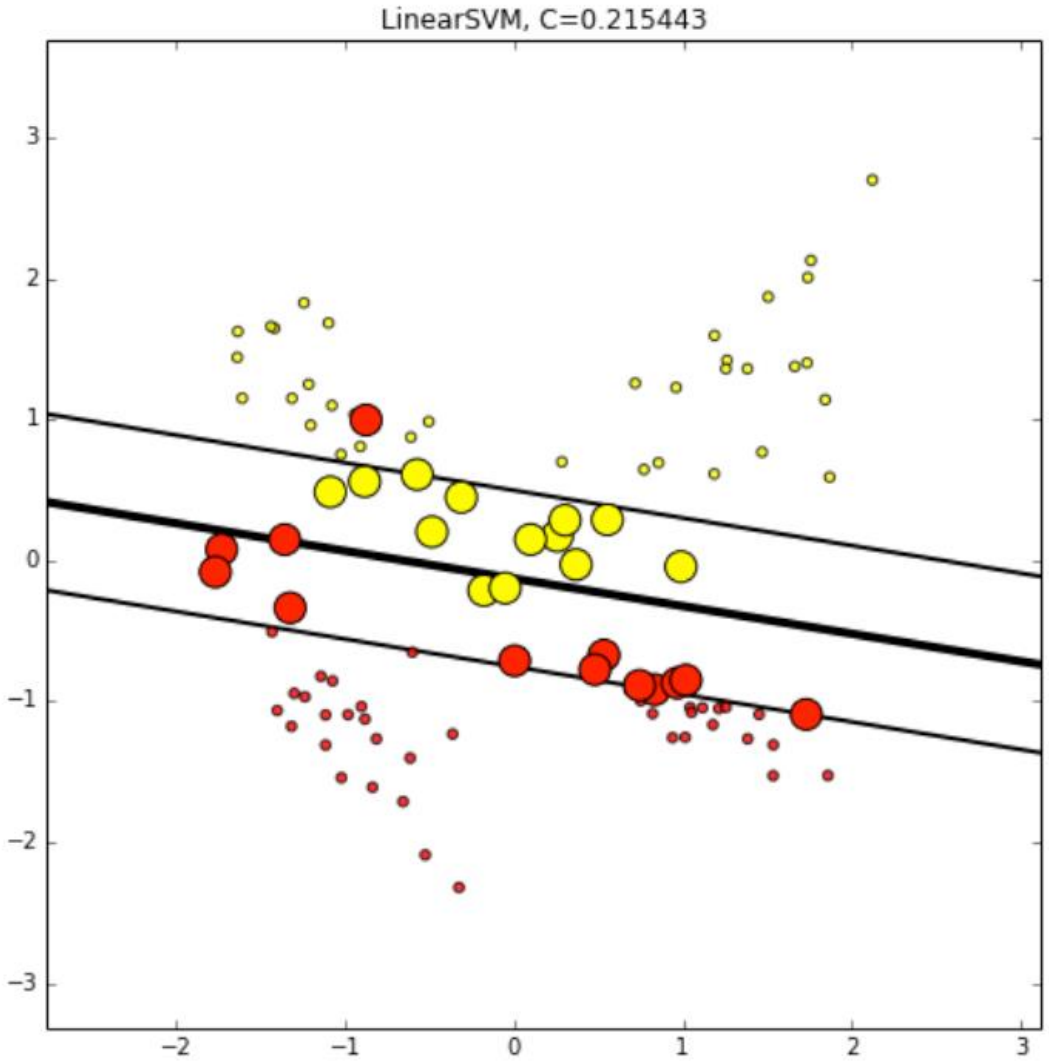
# ЗНАЧЕНИЕ КОНСТАНТЫ C



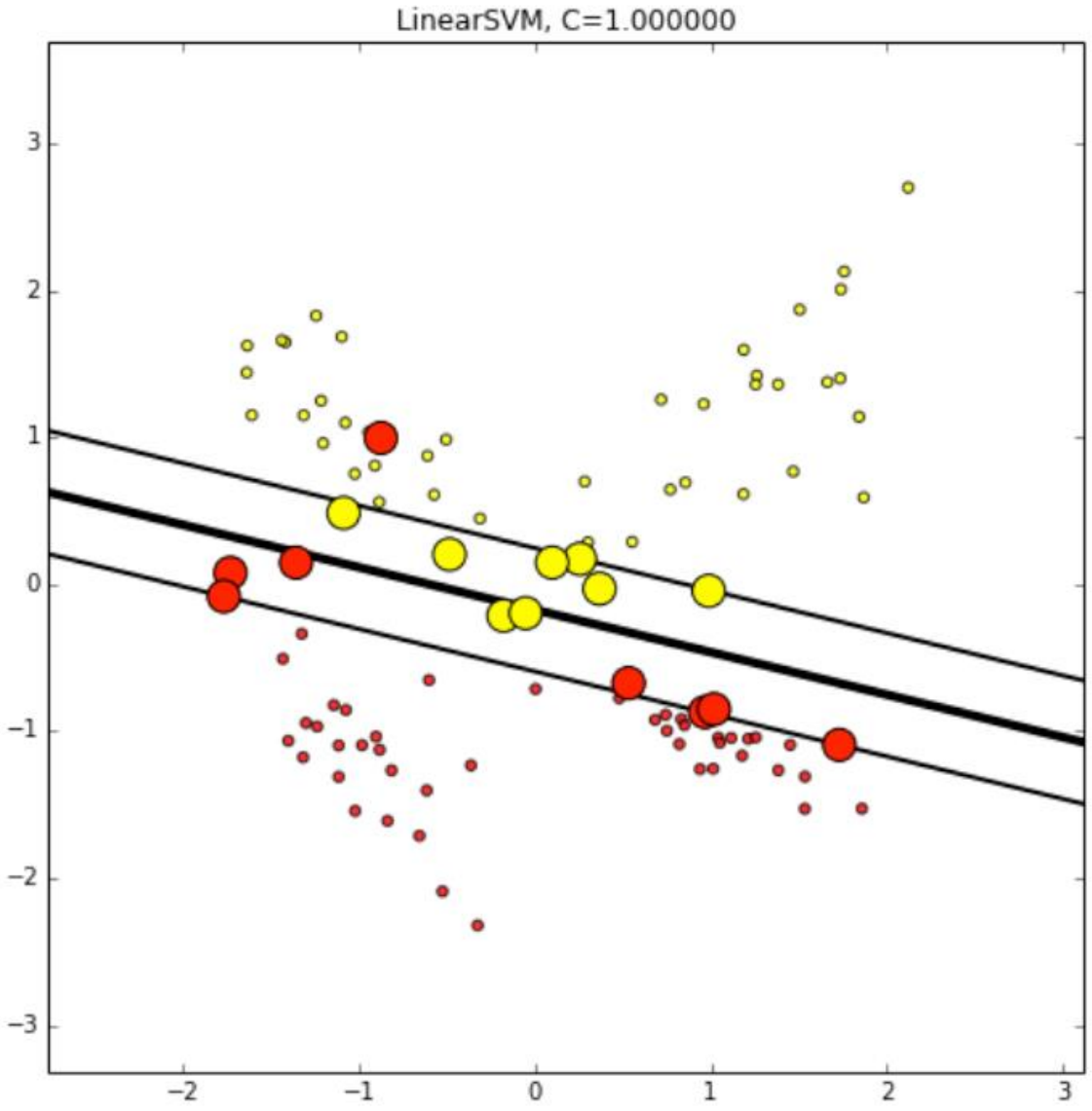
# ЗНАЧЕНИЕ КОНСТАНТЫ C



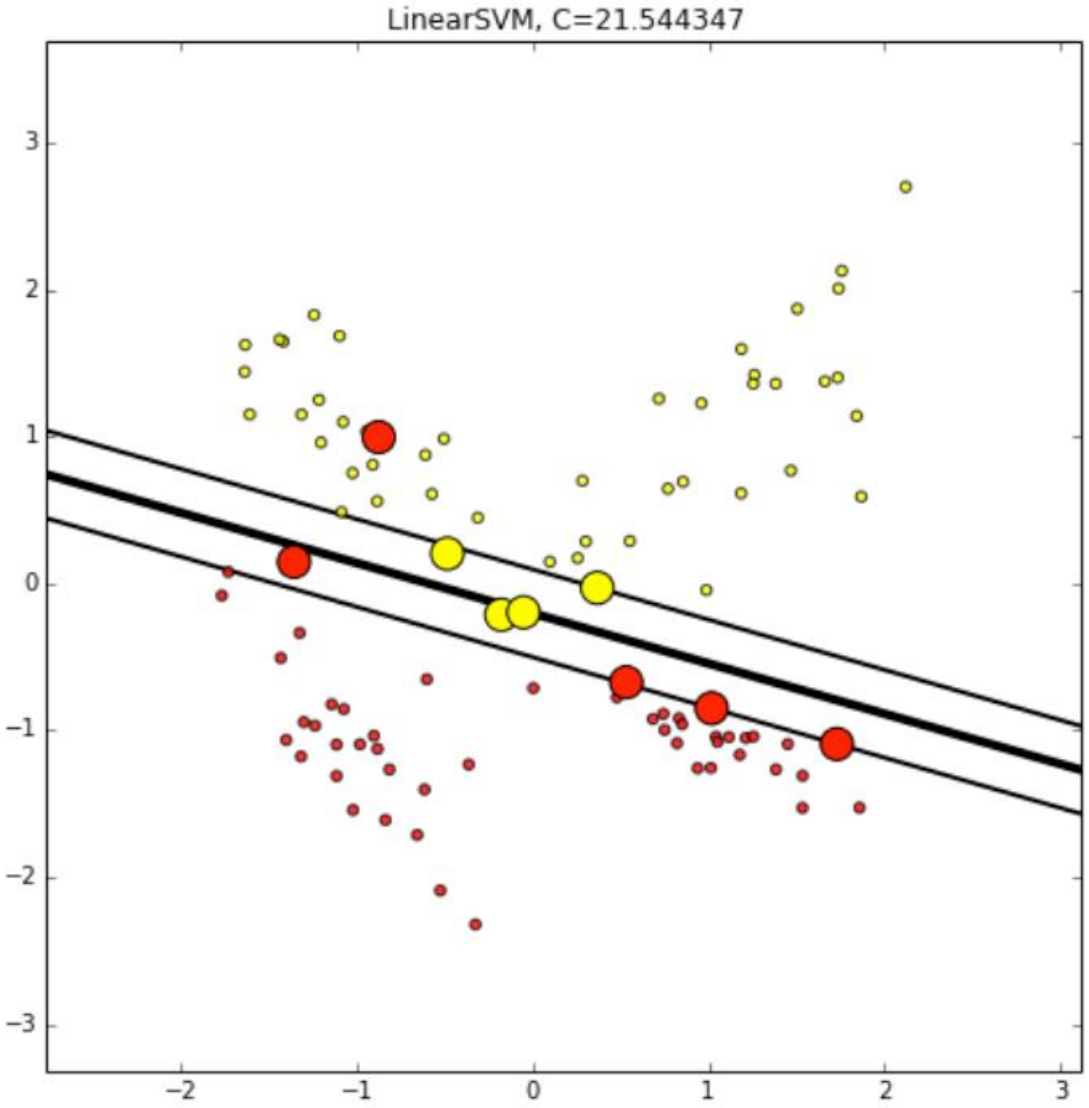
# ЗНАЧЕНИЕ КОНСТАНТЫ C



# ЗНАЧЕНИЕ КОНСТАНТЫ C



# ЗНАЧЕНИЕ КОНСТАНТЫ C



# ТИПЫ ОБЪЕКТОВ В SVM