Математические основы информационной безопасности

Груздев Дмитрий Николаевич

Методы классификации

Классификация

$(x_1, y_1), \ldots, (x_m, y_m)$ – обучающая выборка, $x_i \in X$, $y \in Y$

Y – конечное

Задача:

построить алгоритм A:X–>Y

Метод ближайших соседей

$(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ – объекты-ответы

$\rho(x_i, x_j) \geq 0$ – функция расстояния

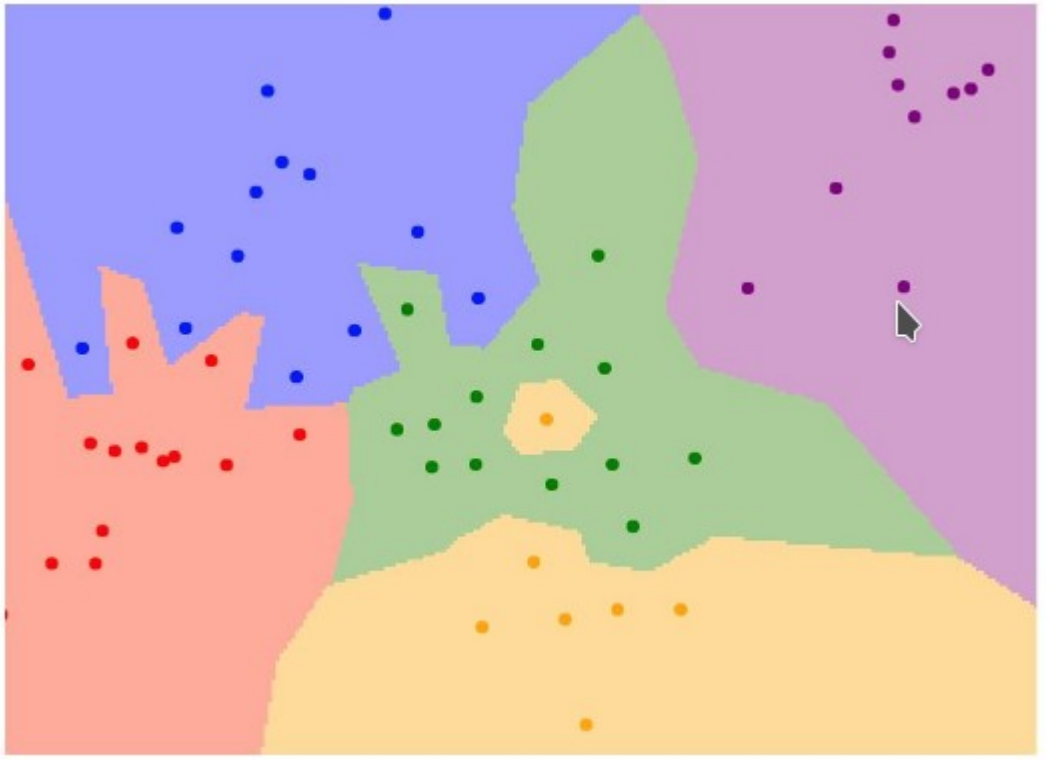
1NN метод:

A(x) = класс ближайшего к x объекта

kNN метод:

A(x) = класс в котором лежат большинство из k ближайших к x объектов

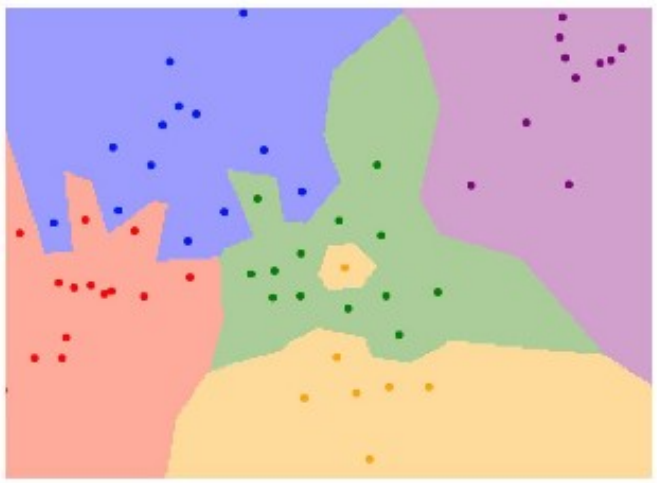
1NN



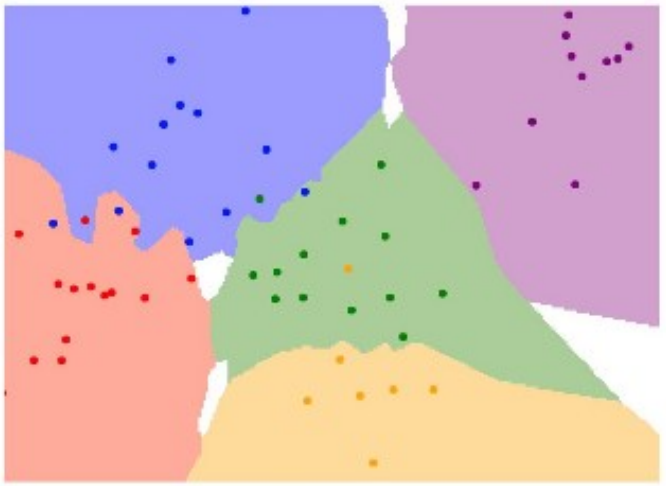
Преимущества: простота реализация, наглядность результатов

Недостатки: неустойчивость к шуму, нужно хранить всю выборку

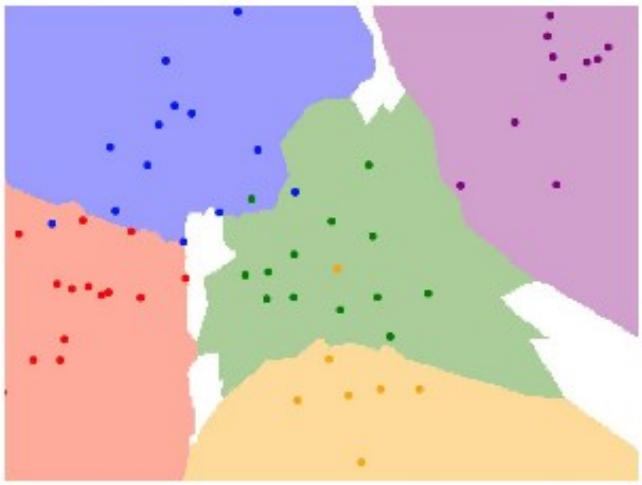
kNN



K = 1



K = 3



K = 5

Белые области – точки, в одинаковой степени относящиеся к нескольким классам.

Метод парзеновского окна

x – исследуемый объект

$\rho(x, x^*_1) \le \rho(x, x^*_2) \le \ldots \le \rho(x, x^*_m)$

Возьмем k ближайших соседей

$w_i = K(\rho(x, x^*_i)/\rho(x, x^*_k))$ вес i-го соседа

K – ядро, невозрастающая, положительная на [0, 1]

$w_{Yi} = \Sigma w_j \mid y_j \in Y_i$

Выбрать класс с наибольшей суммой весов

Отбор эталонов

В обучающей выборке есть излишние объекты.

Количество обучающих объектов влияет на скорость работы алгоритма.

Задача: уменьшить размер обучающей выборки без уменьшения качества классификации.

Алгоритм добавления эталонов:

1. Исключить выбросы из обучающей выборки;
2. Взять по одному объекту в каждом классе (самые удаленные от границ объекты);
3. Добавлять объекты из приграничных объектов, пока не получим классификацию приемлемого качества.

Бинарное решающее дерево

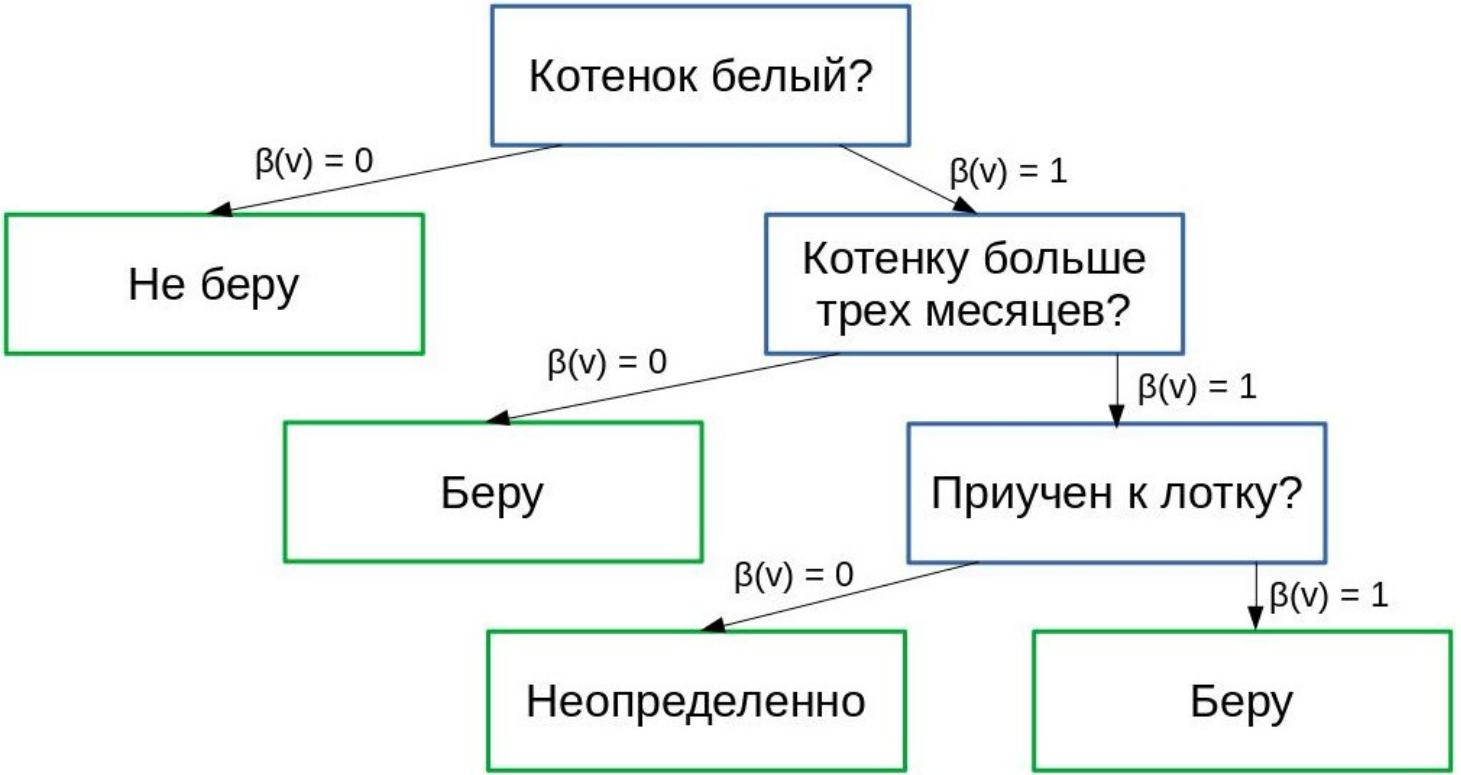
Б.Р.Д. – алгоритм классификации, задающийся бинарным деревом:

Внутренняя вершина – бинарная функция $\beta_v(x) \to \{0,1\}$
Внешняя вершина (лист) – метка класса

Алгоритм классификации:

Для заданного x начиная с корневой вершины вычислять $\beta_v(x)$. Если $\beta_v(x) = 0$, идти в левое поддерево, если $\beta_v(x) = 1$ – в правое. Когда дойдем до листа, то получим нужный класс.

Пример решающего дерева



Котенок = (x1-цвет, x2-возраст, x3-лоточность)
Классы = {беру, не беру, неопределенно}

Разбиение выборки

$(x_1, y_1), \ldots, (x_m, y_m)$ – обучающая выборка, $x_i \in X$, $y \in Y$
$x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})$ – признаки
$\beta(x_i) \to \{0,1\}$ - разбиение

Варианты правил разбиений (условие для $\beta(x_i) = 1$, иначе $\beta(x_i) = 0$):

- Пороговое условие: $x_i^{(j)} = a_j$
- Пороговое условие: $a_j \le x_i^{(j)} \le b_j$; $a_j \le x_i^{(j)}$; $x_i^{(j)} \le b_j$
- Конъюнкция пороговых условий: $\wedge (a_j \le x_i^{(j)} \le b_j)$, $j \in J$
- Синдром: выполняется не меньше d условий из J
- Полуплоскость: $\Sigma w_j x_i^{(j)} \ge w_0$
- Шар: $\rho(x_i, x_0) \le w_0$

Величины, выделенные красным, настраиваются по обучающей выборке.

Эффективность разбиения

Коэффициент Джини:

количество пар объектов принадлежащих одному классу и оказавшихся в одном поддереве:
$I(\beta_v) = |\{(x_i,x_j): \beta_v(x_i) = \beta_v(x_j) \text{ и } y_i = y_j\}| \,/\, |\{x_i,x_j\}|$

Коэффициент В.И.Донского:

количество пар объектов принадлежащих разным классам и оказавшихся в разных поддеревьях:
$I(\beta_v) = |\{(x_i,x_j): \beta_v(x_i) \neq \beta_v(x_j) \text{ и } y_i \neq y_j\}| \,/\, |\{x_i,x_j\}|$

Энтропийный коэффициент

Построение дерева ID3

Алгоритм ID3:

1. Если все объекты выборки U из одного класса, вернуть лист с меткой этого класса.
2. Найти разбиение β с максимальным коэффициентом разбиения и определить $U = U_0 \cup U_1$ по β.
3. Если $U_0 = ø$ или $U_1 = ø$, вернуть метку класса, объектов которого в U больше всего (мажоритарного класса).
4. Рекурсивно построить левое и правое поддерево по U_0 и U_1 соответственно.

Редукция дерева: C4.5, CART

Контрольная выборка длины $k \approx 0.5*m$

Алгоритм редукции (стрижки) дерева:

1. Если ни один объект контрольной выборки не зашел в вершину v, то заменяем ее на лист с мажоритарным классом обучающей подвыборки для этой вершины.
2. Пробуем каждую вершину заменить на ее правое или левое поддерево, или на фиксированный класс. Если количество ошибок классификации уменьшилось, оставляем замену.

Линейный классификатор

$(x_1, y_1), \ldots, (x_m, y_m)$ – обучающая выборка, $x_i \in X$, $y \in Y$

$X = \mathbb{R}^n$, $Y = \{-1, 1\}$

Задача: построить алгоритм классификации вида
$A(x, \theta) = \text{sign } f(x, \theta)$, где

θ – набор параметров
$f(x, \theta)$ – дискриминантная функция

$f(x, \theta) = 0$ – разделяющая поверхность

$M_i(\theta) = y_i * f(x_i, \theta)$ – отступ объекта x_i

$M_i(\theta) < 0 \Leftrightarrow A(x, \theta)$ – ошибается на x_i

Функция ошибок

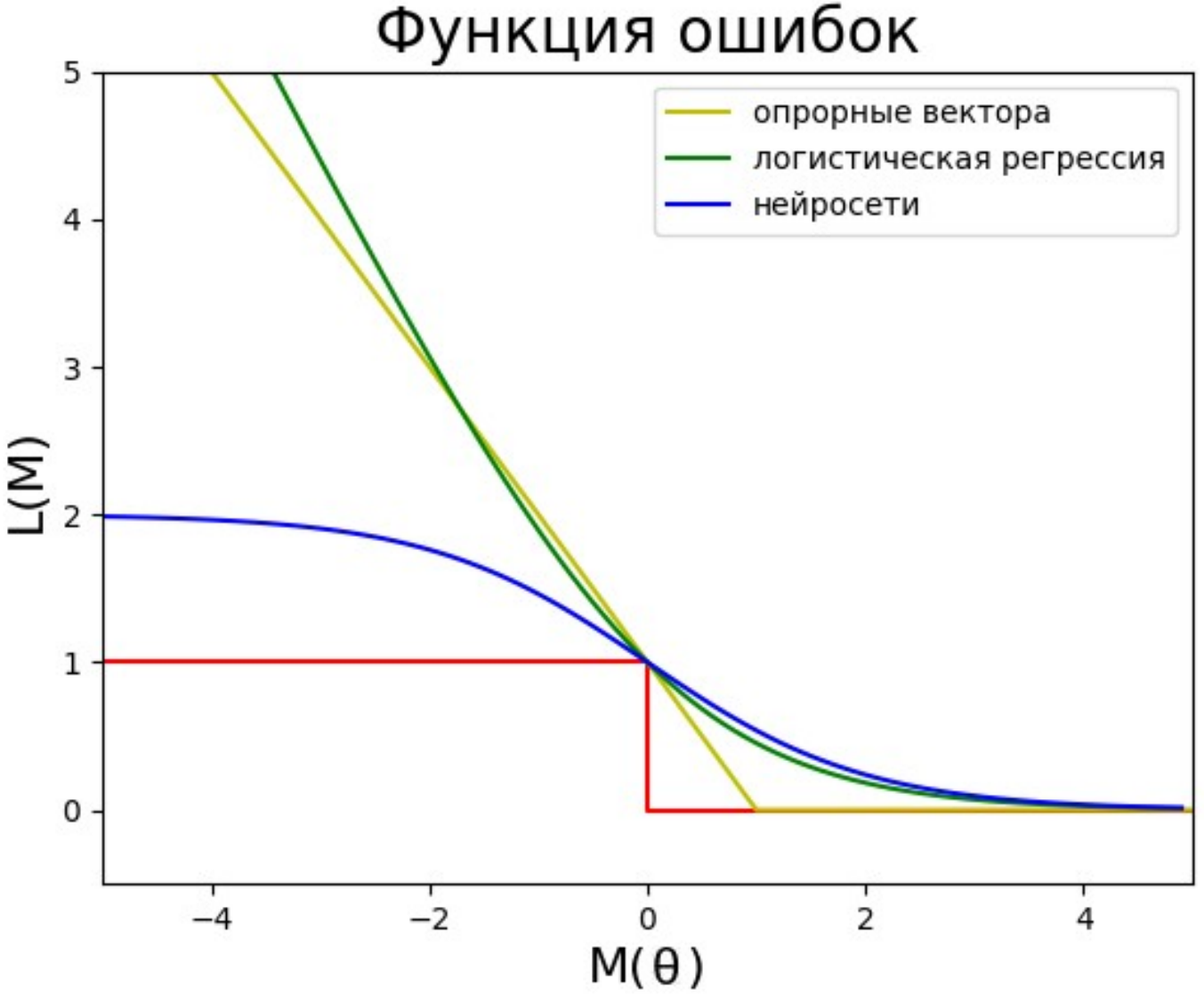
Функция [x]:

[x] = 1, если x – истинно, [x] = 0, если x – ложно

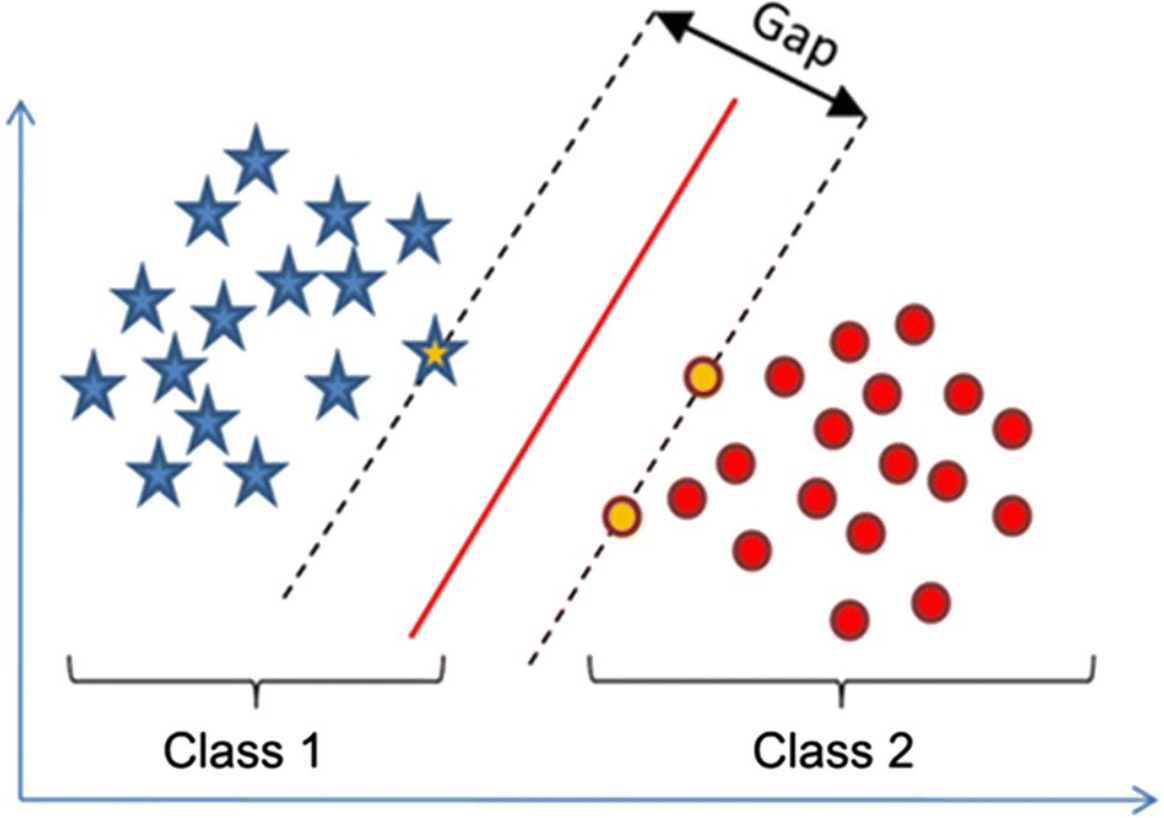
$E(\theta) = \Sigma[M_i(\theta) < 0]$ – количество ошибок на обучающей выборке, дискретная функция ошибки

$E^*(\theta) = \Sigma L(M_i(\theta))$ – непрерывная функция, на которую мы заменяем дискретную функцию ошибки
L(M) – невозрастающая, неотрицательная

Будем минимизировать $E^*(\theta)$


Функция ошибок
опрорные вектора
логистическая регрессия
нейросети
L(M)
M(θ)
5
4
3
2
1
0
−4
−2
0
2
4

Оптимальная разделяющая гиперплоскость



Разделяющая гиперплоскость максимально удалена от разделяемых классов.

Метод опорных векторов

$(x_1, y_1), \ldots, (x_m, y_m)$ – обучающая выборка, $x_i \in X$, $y \in Y$

$X = \mathbb{R}^n$, $Y = \{-1, 1\}$

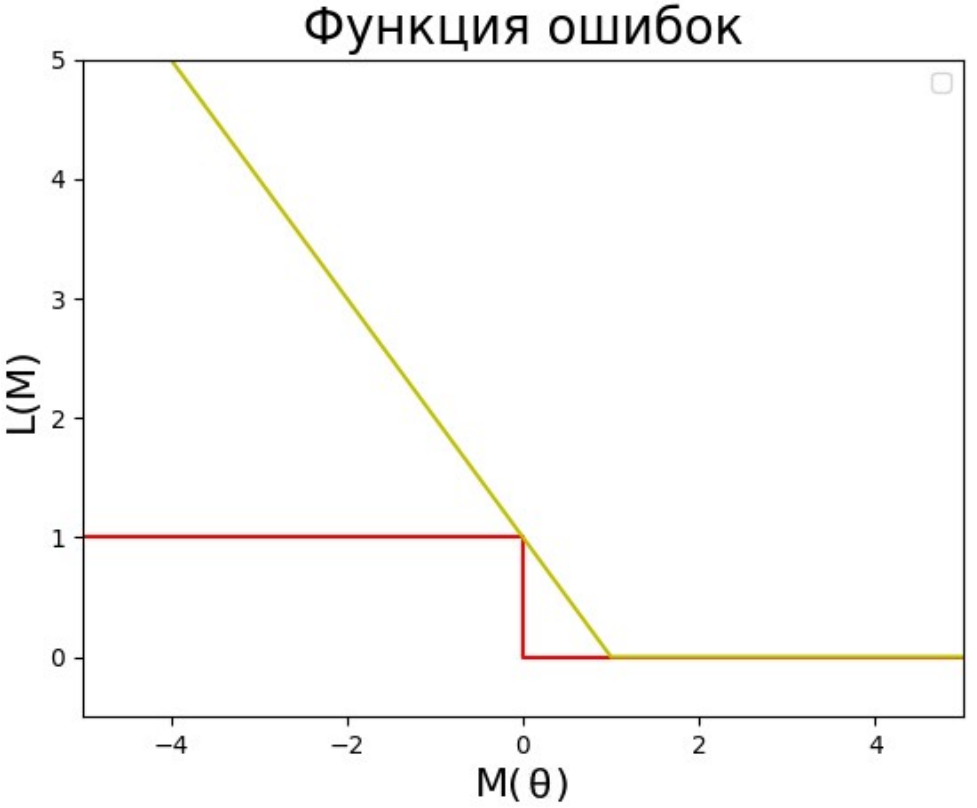
$f(x, \theta) = \langle x, \theta \rangle$

$A(x, \theta) = \text{sign}(\langle x, \theta \rangle)$

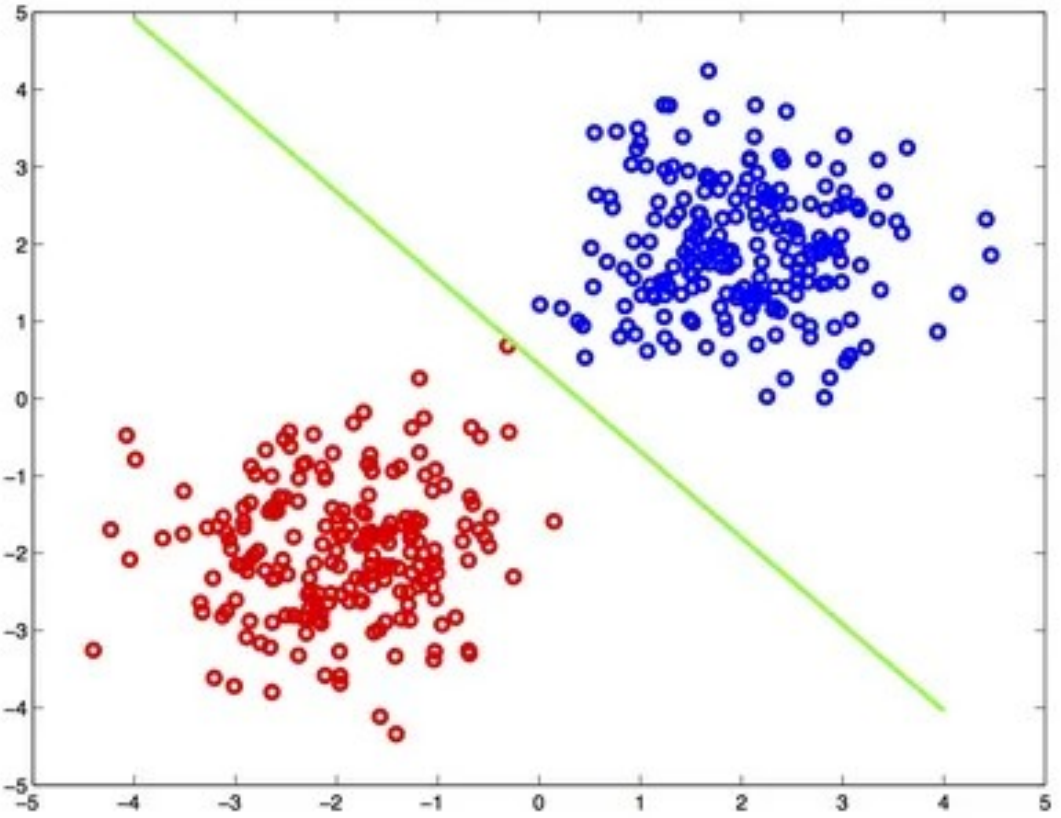
$M_i(\theta) = y_i * \langle x, \theta \rangle$

$L(M_i) = (1 - M_i(\theta))_+$

$E^*(\theta) = \Sigma(1 - M_i(\theta)) \rightarrow \min$



Вероятность ошибки



Разделяющая гиперплоскость позволяет оценить вероятность ошибки классификации.

Логистическая регрессия

$(x_1, y_1), \ldots, (x_m, y_m)$ – обучающая выборка, $x_i \in X$, $y \in Y$

$X = \mathbb{R}^n$, $Y = \{-1, 1\}$

$f(x, \theta) = \langle x, \theta \rangle$

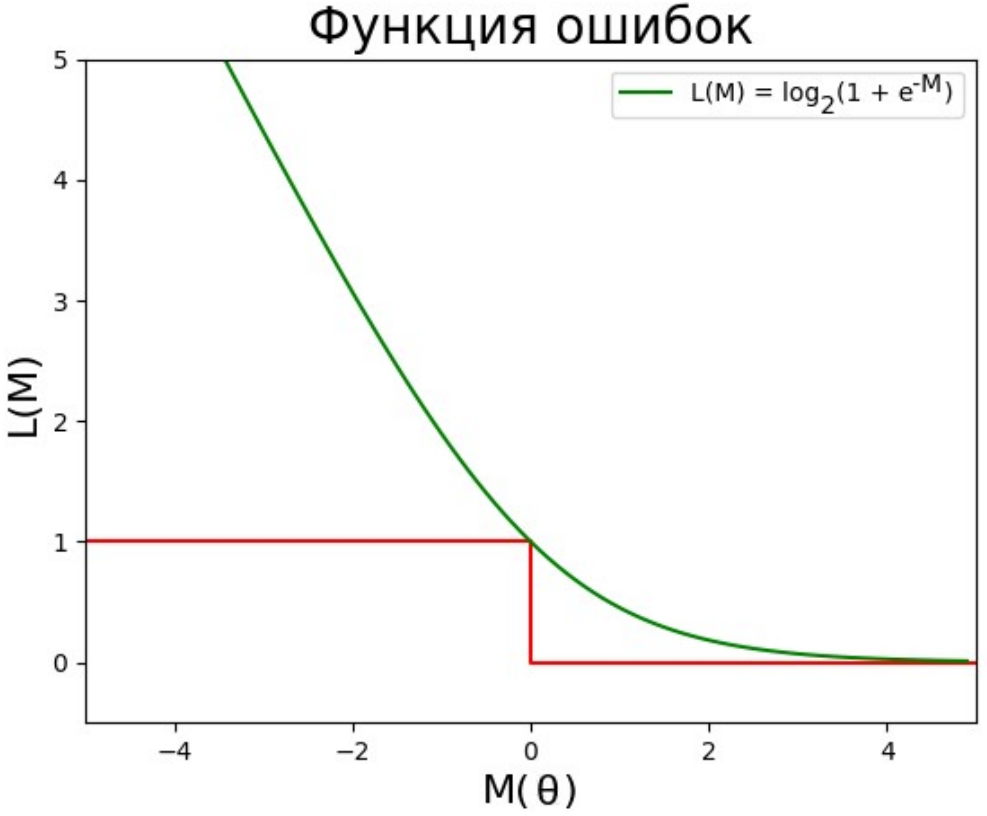
$A(x, \theta) = \text{sign}(\langle x, \theta \rangle)$

$M_i(\theta) = y_i * \langle x_i, \theta \rangle$

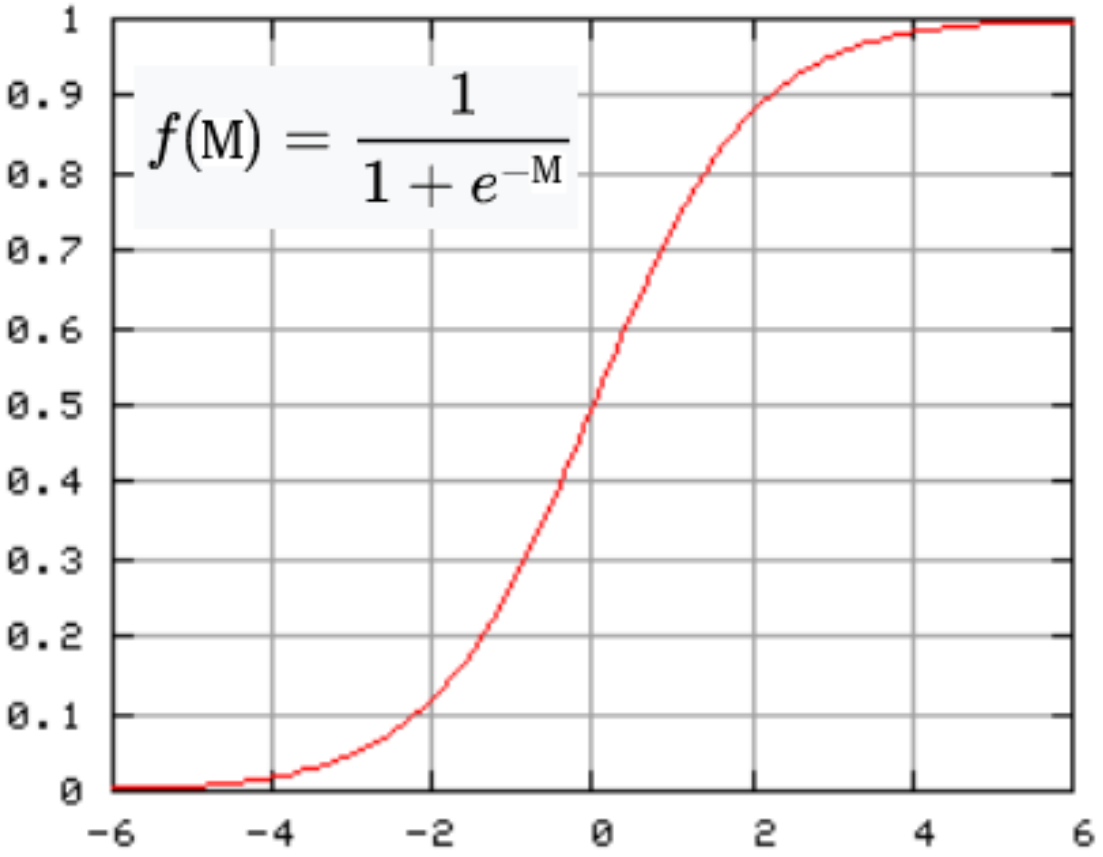
$L(M_i) = \log_2(1 + e^{-M_i})$

$E^*(\theta) = \Sigma \log_2(1 + e^{-M_i}) \rightarrow \min$

$P(y|x) = 1/(1 + \exp(-\langle x, \theta \rangle * y))$



Логистическая функция



M = <x, θ>*y – отступ x

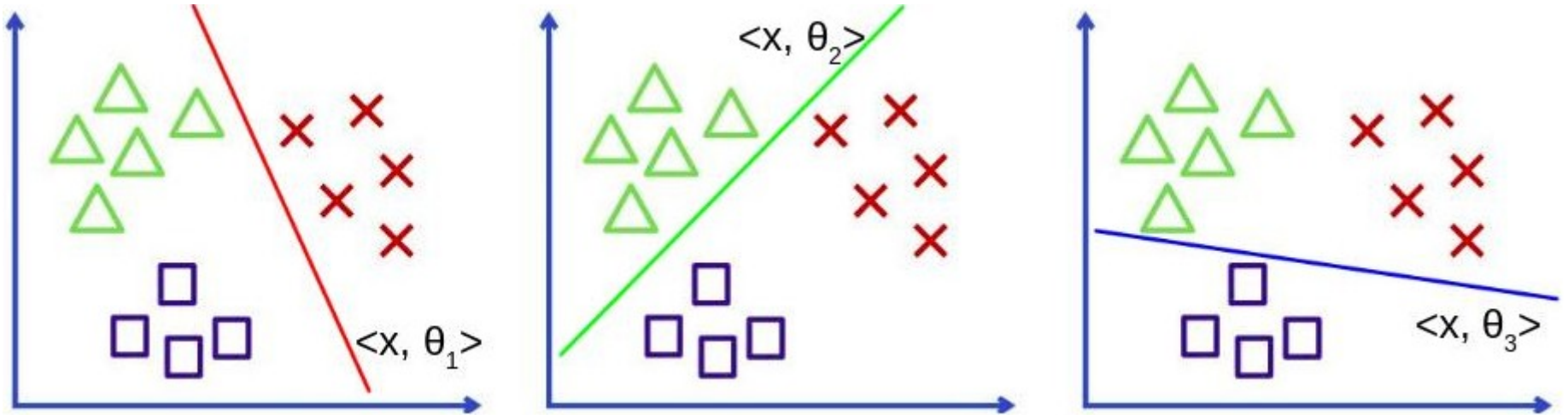
$$P(y|x) = \frac{1}{1 + e^{-\langle x, \theta \rangle * y}}$$

$$P(1|x) = \frac{1}{1 + e^{-\langle x, \theta \rangle}}$$

$$P(-1|x) = \frac{1}{1 + e^{\langle x, \theta \rangle}}$$

$$P(-1|x) + P(1|x) = 1$$

Многоклассовая классификация



1. Построить разделяющие плоскости для каждого класса.
2. Для нового объекта посчитать, с какой вероятностью он относится к каждому классу.
3. Результат алгоритма - класс с максимальной вероятностью.

scikit-learn.org

https://sesc-infosec.github.io/