Машинное обучение

Занятие 4.
Логистическая регрессия

Решающее дерево

Линейная регрессия

$$a(x) = \boxed{w_0 + \sum_{j=1}^{d} w_j x_j}$$

Вещественное число!

Линейный классификатор

$$a(x) = \operatorname{sign}\left(w_0 + \sum_{j=1}^{d} w_j x_j\right)$$

Линейный классификатор

$$a(x) = \operatorname{sign}\left(w_0 + \sum_{j=1}^{d} w_j x_j\right)$$

Свободный коэффициент

Веса

Признаки

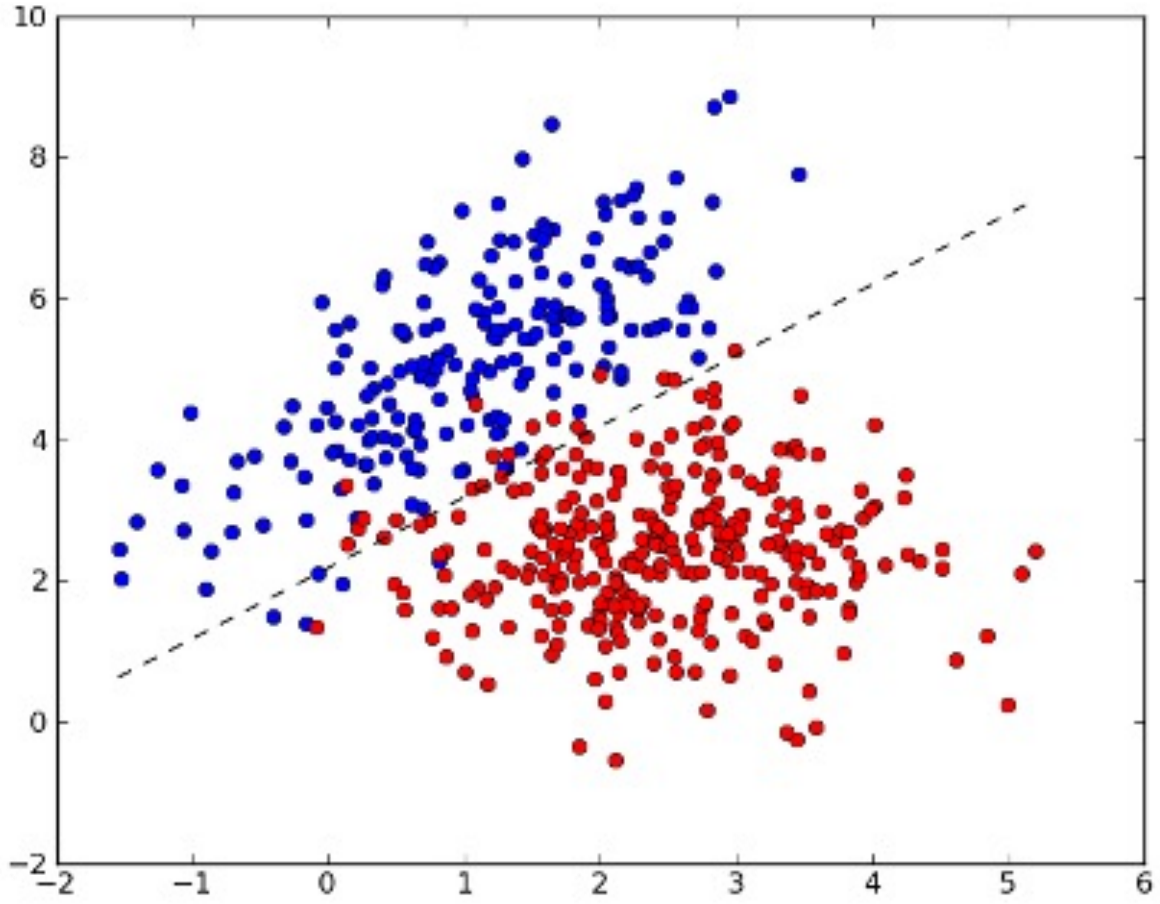
Линейный классификатор

- Будем считать, что есть единичный признак

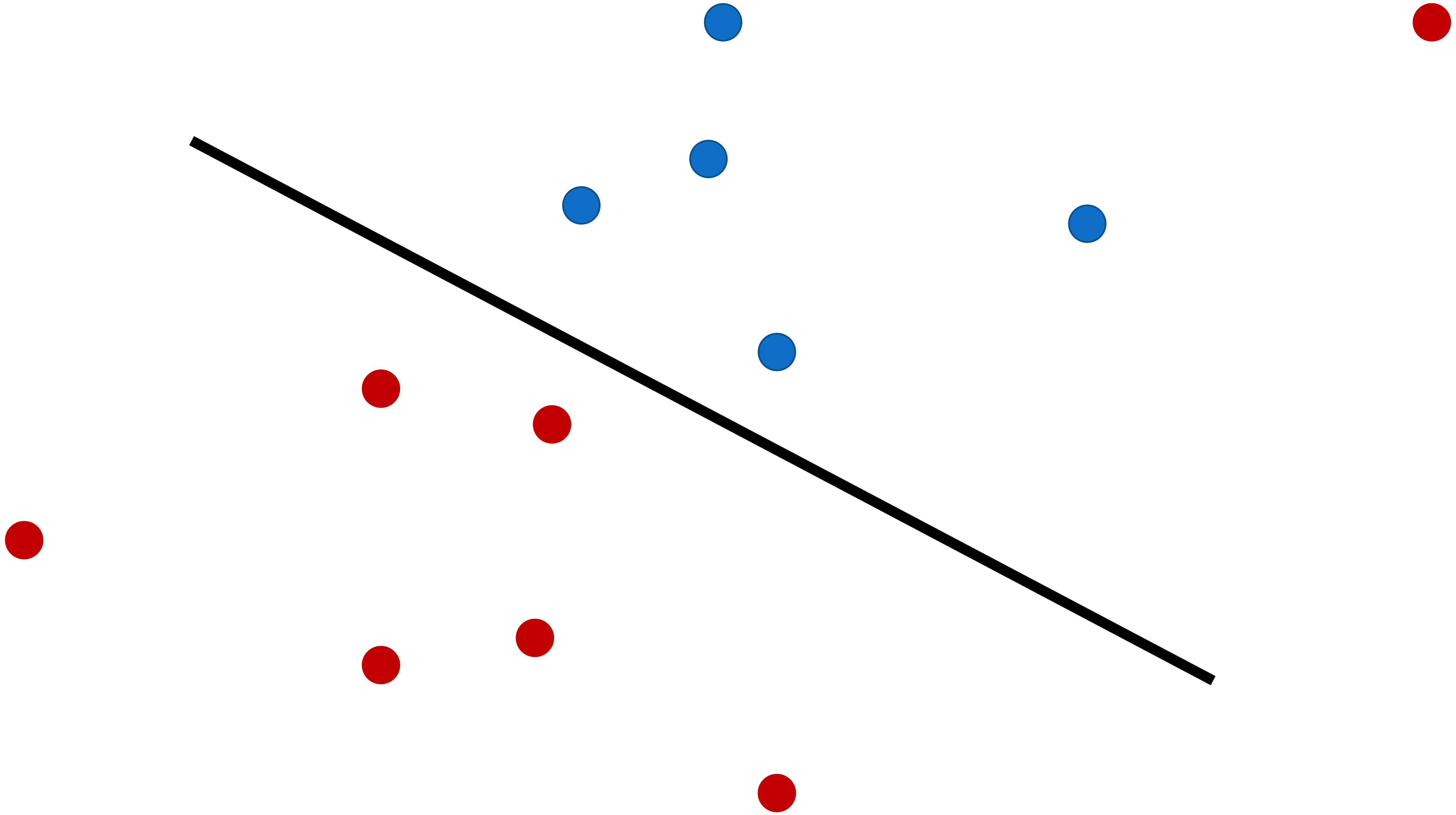
$$a(x) = \operatorname{sign} \sum_{j=0}^{d} w_j x_j = \operatorname{sign} \langle w, x \rangle$$

Геометрия линейного классификатора

- Линейный классификатор проводит гиперплоскость
- $\langle w, x \rangle < 0$ — объект «слева» от неё
- $\langle w, x \rangle > 0$ — объект «справа» от неё

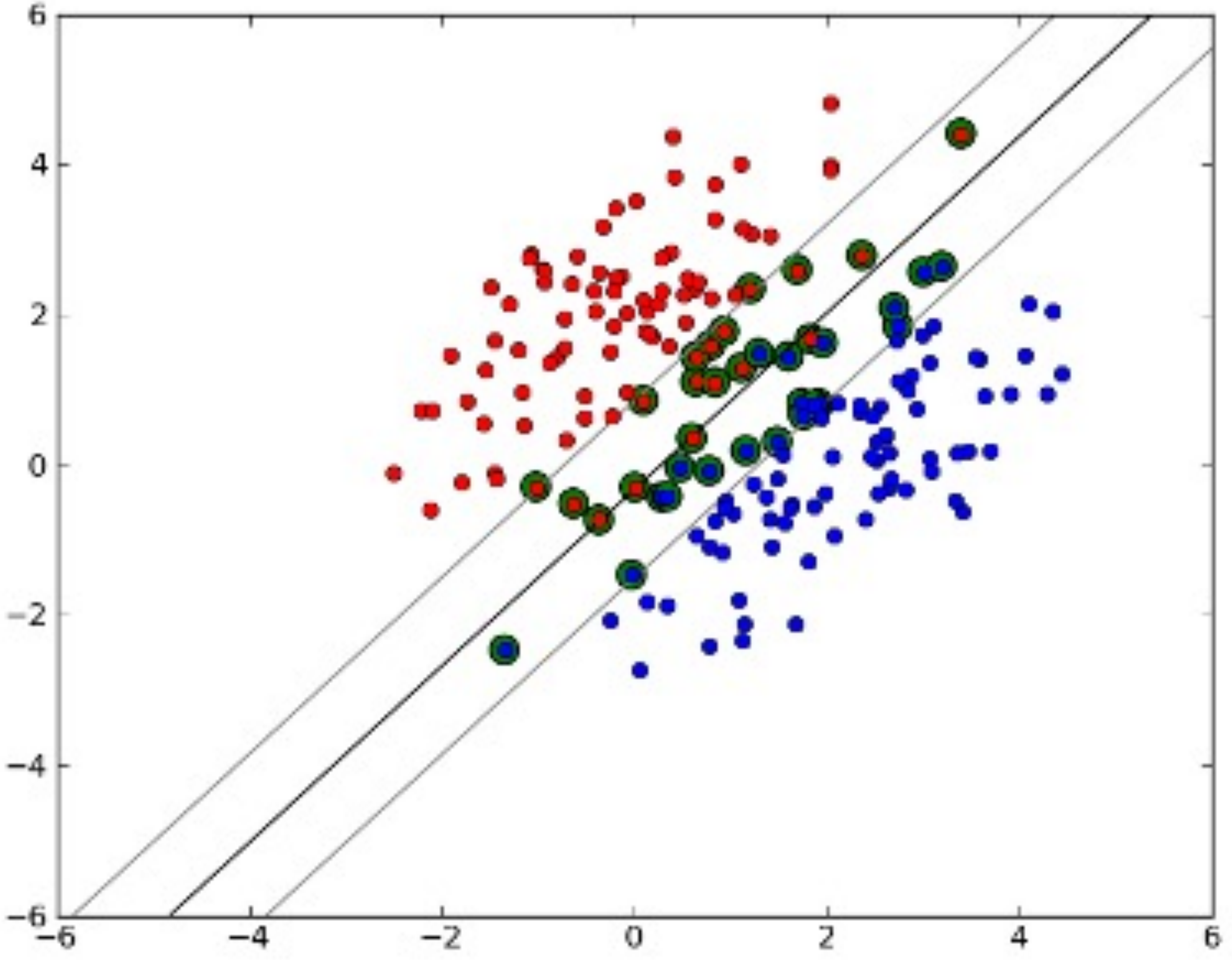


Геометрия линейного классификатора



Отступ

- $M_i = y_i \langle w, x_i \rangle$
- $M_i > 0$ — классификатор дает верный ответ
- $M_i < 0$ — классификатор ошибается
- Чем дальше отступ от нуля, тем больше уверенности

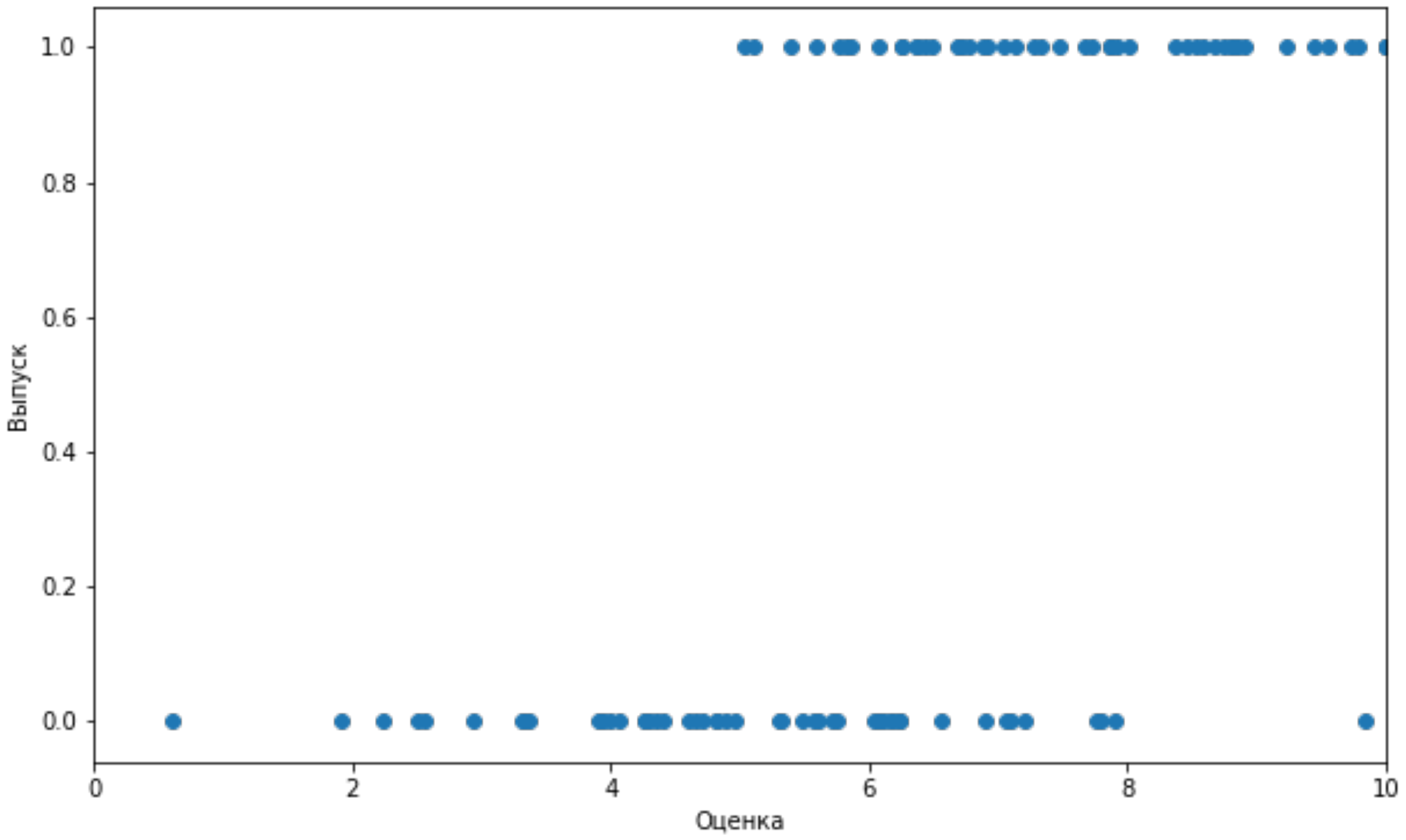


Логистическая регрессия: простое объяснение

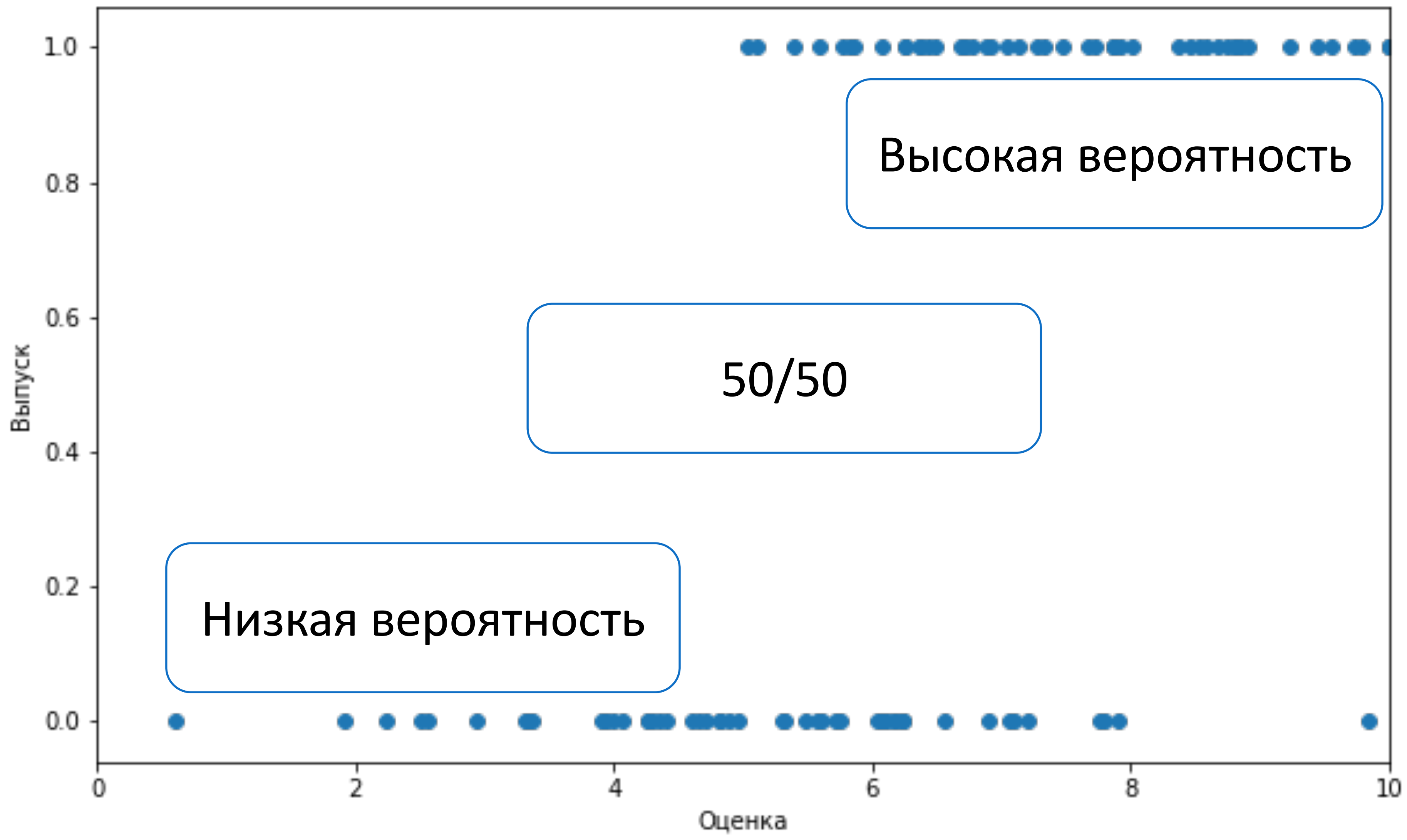
Логистическая регрессия

- Решаем задачу бинарной классификации: $\mathbb{Y} = \{-1, +1\}$

Предсказание вероятностей



Предсказание вероятностей



Предсказание вероятностей

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

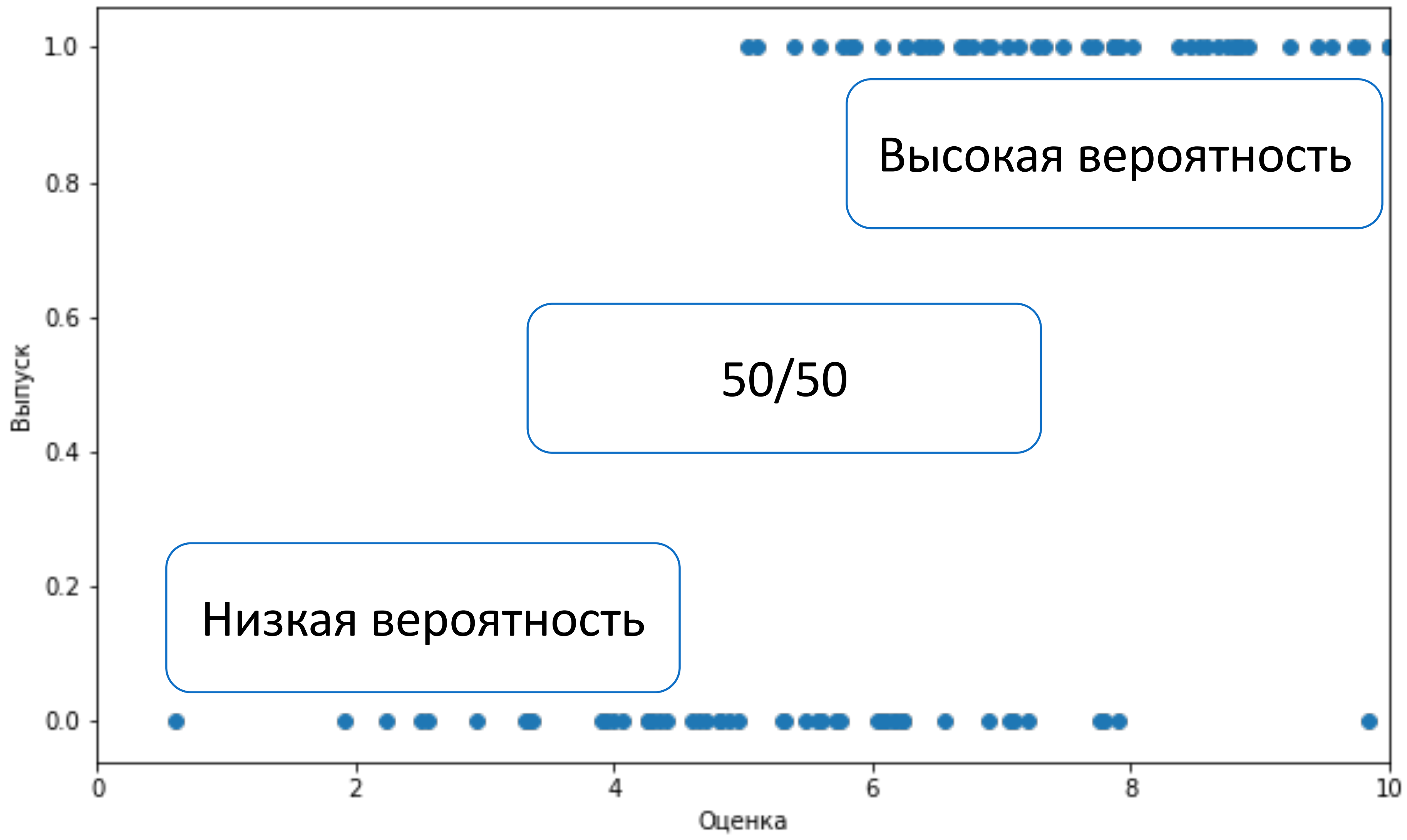
Предсказание вероятностей

- Прогнозирование оттока клиентов
- Медицинская диагностика
- Поисковое ранжирование (насколько веб-страница соответствует запросу?)

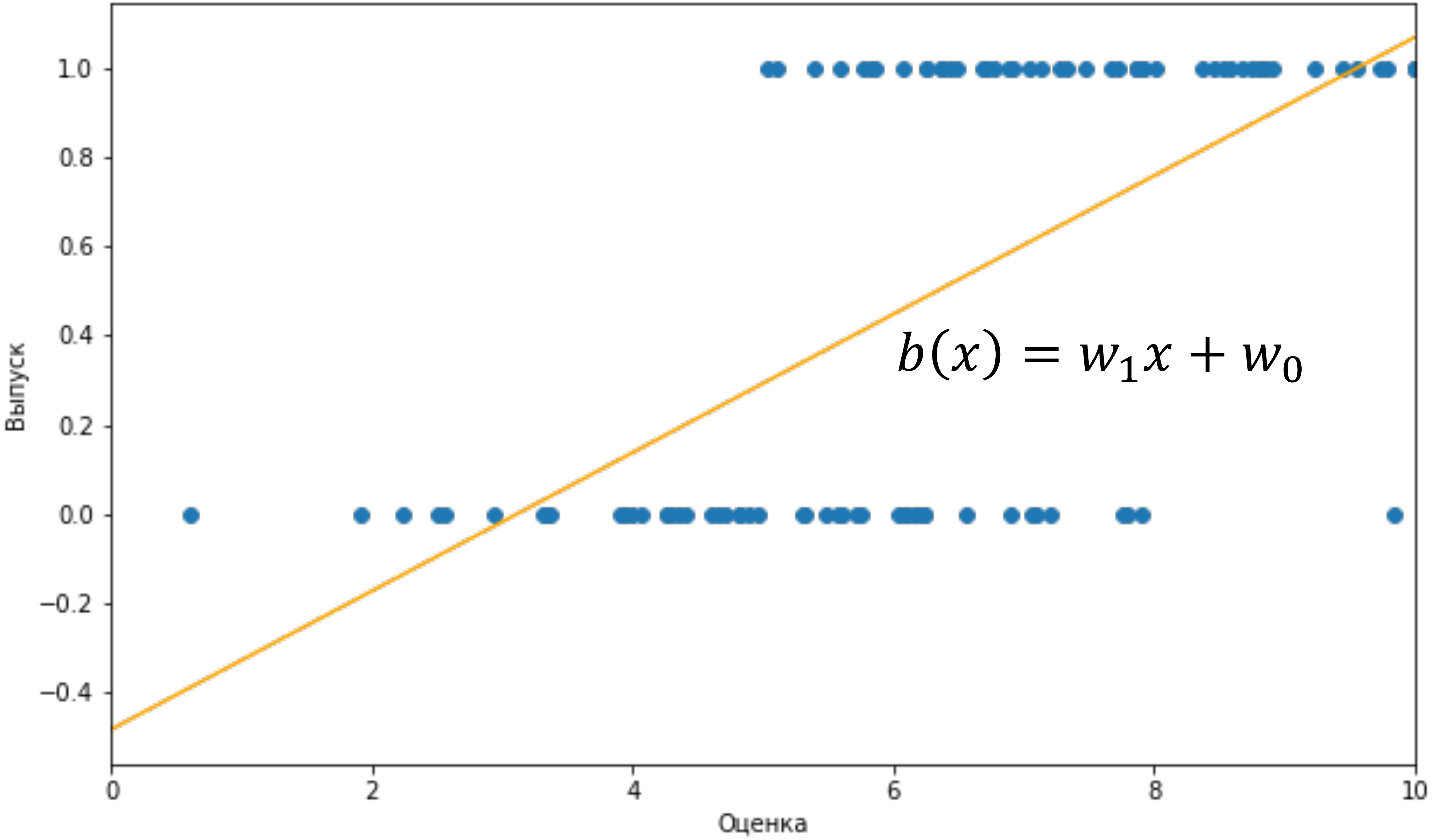
Предсказание вероятностей

Будем говорить, что модель $b(x)$ предсказывает вероятности, если среди объектов с $b(x) = p$ доля положительных равна p.

Предсказание вероятностей



Предсказание вероятностей



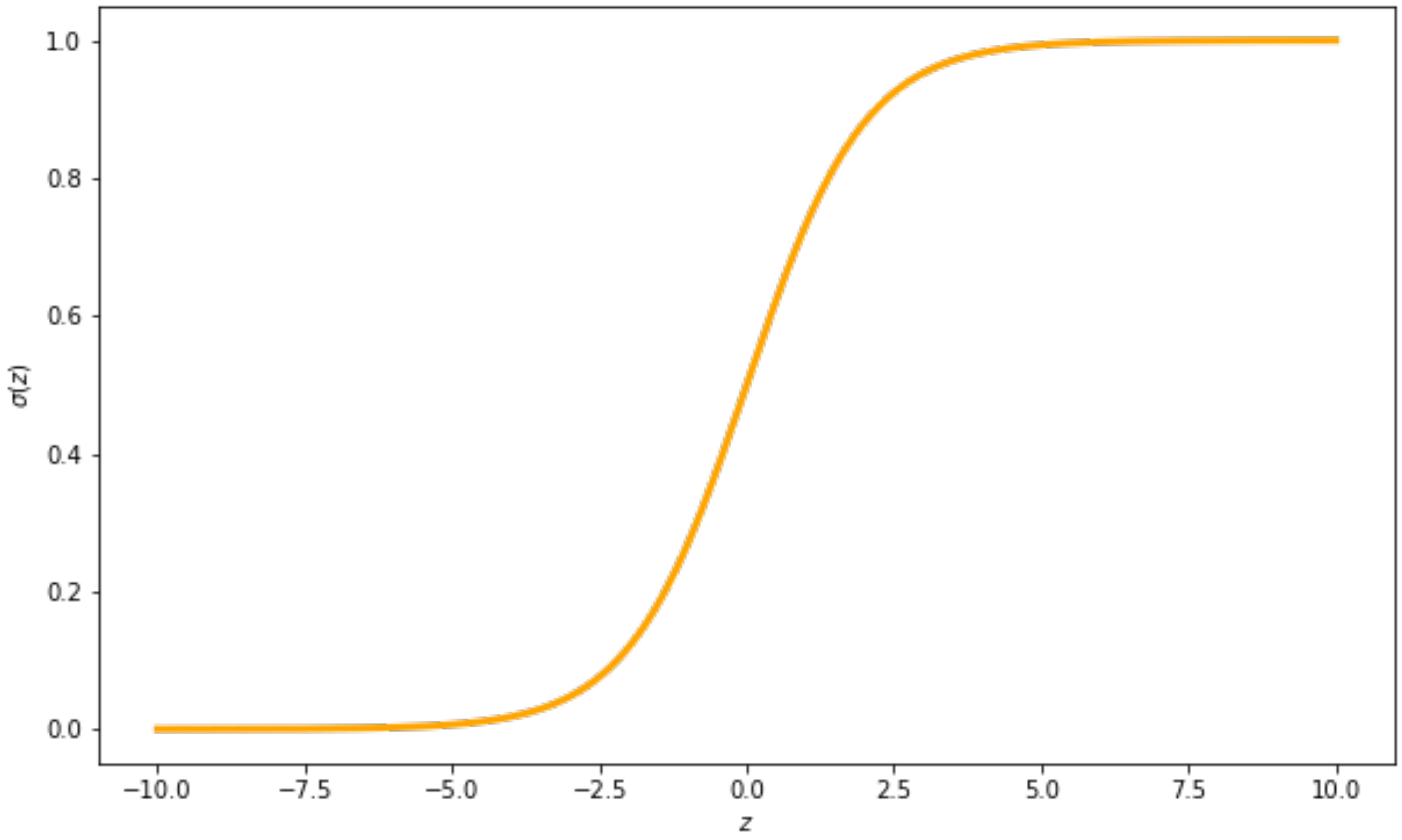
$$b(x) = w_1 x + w_0$$

Линейный классификатор

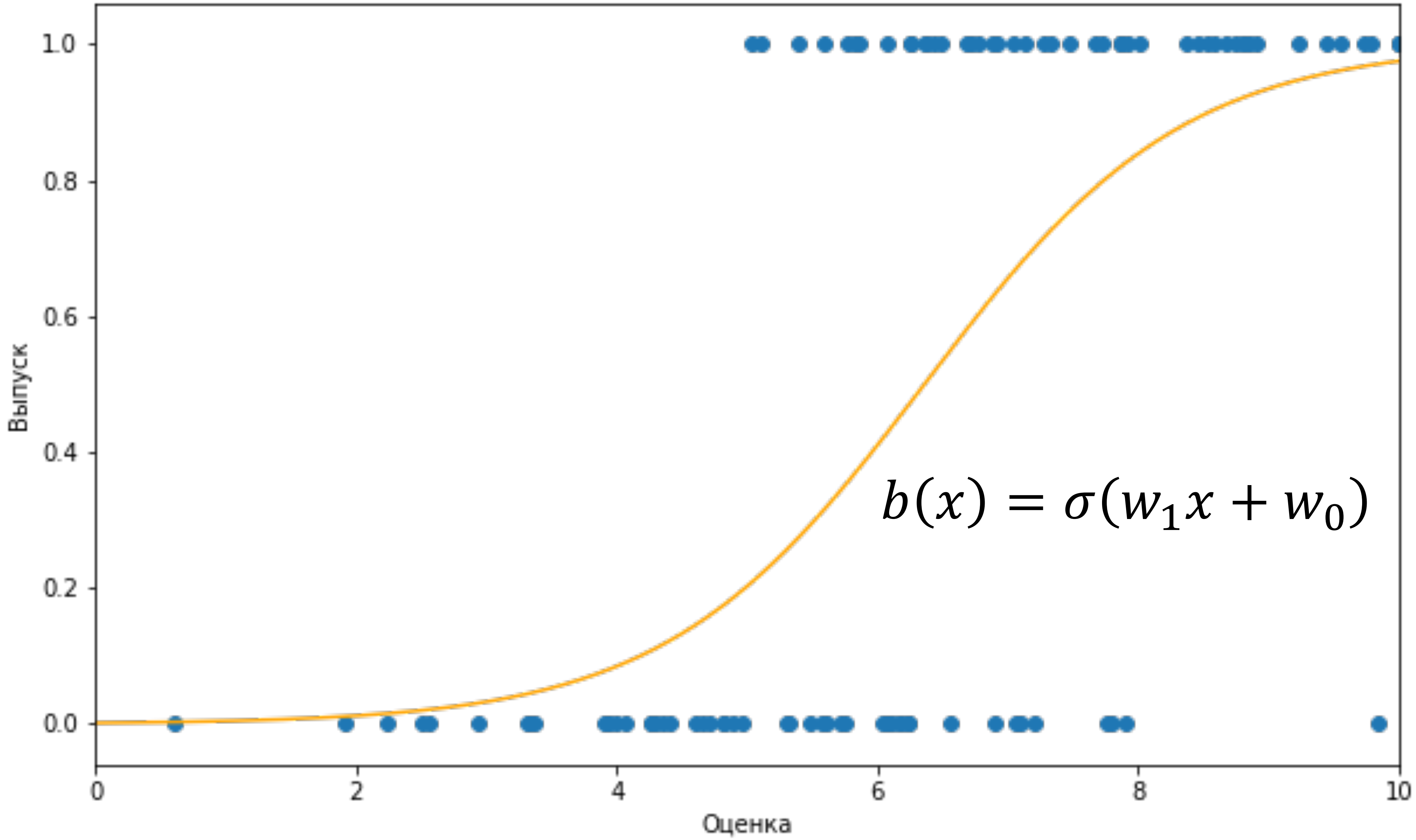
- Переведём выход модели на отрезок [0, 1]
- Например, с помощью сигмоиды:

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

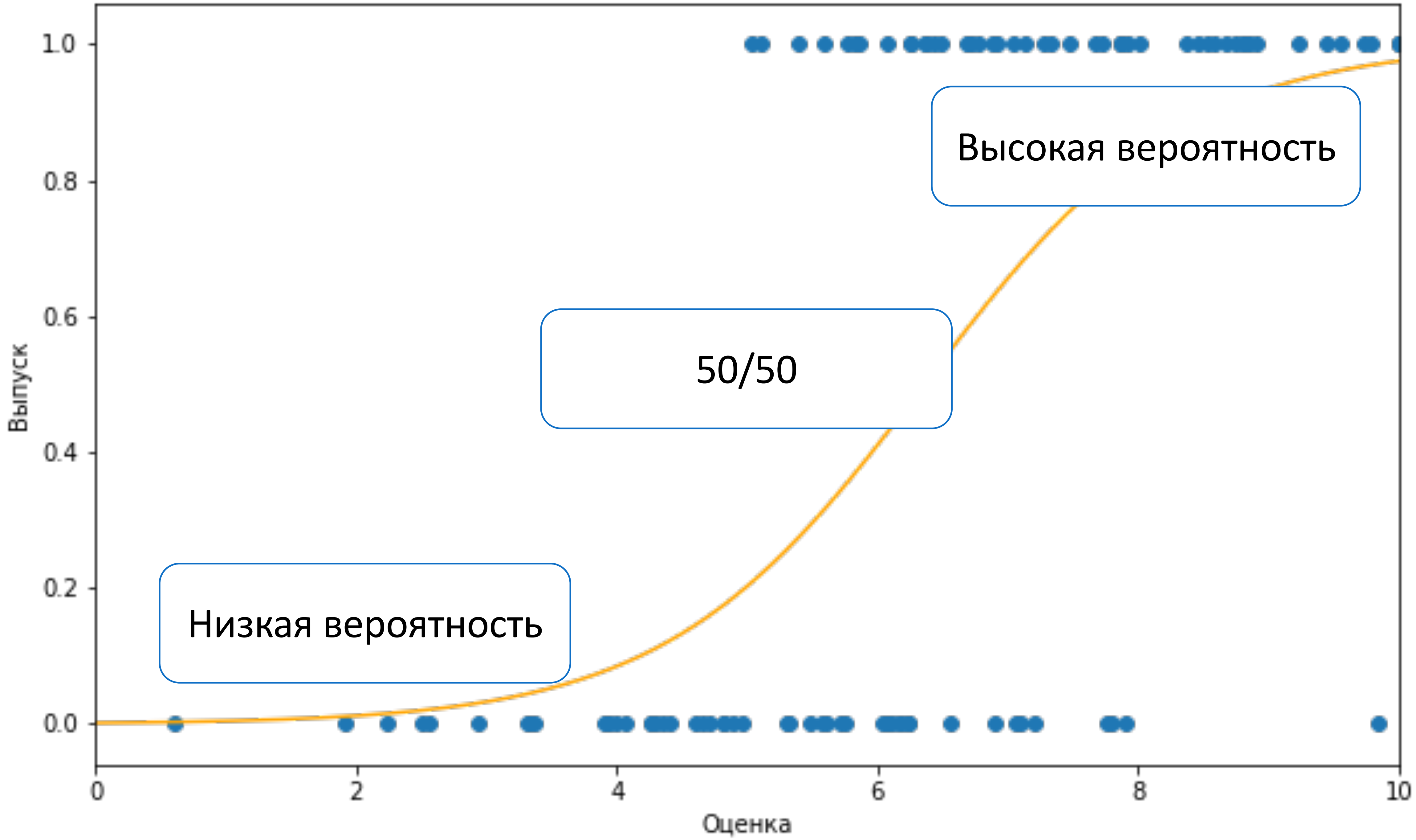
Сигмоида



Предсказание вероятностей



Предсказание вероятностей



Предсказание вероятностей

- Модель для оценивания вероятностей:

$$b(x) = \sigma(\langle w, x \rangle)$$

- Как обучать?

Предсказание вероятностей

- Модель для оценивания вероятностей:

$$b(x) = \sigma(\langle w, x \rangle)$$

- Как обучать?
- Если $y_i = +1$, то $\sigma(\langle w, x_i \rangle) \to 1$
- Если $y_i = -1$, то $\sigma(\langle w, x_i \rangle) \to 0$

Предсказание вероятностей

- Модель для оценивания вероятностей:

$$b(x) = \sigma(\langle w, x \rangle)$$

- Как обучать?

$$\prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \left(1 - b(x_i)^{[y_i=-1]}\right) \to \max$$

Предсказание вероятностей

$$\prod_{i=1}^{l} b(x_i)^{[y_i=+1]} \left(1 - b(x_i)^{[y_i=-1]}\right) \to \max$$

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

$L(y, b) = [y = +1] \log b + [y_i = -1] \log(1 - b)$ - logloss

Логистическая регрессия

$$-\sum_{i=1}^{\ell}\{[y_i = 1]\log\sigma(\langle w, x_i\rangle) + [y_i = -1]\log\big(1 - \sigma(\langle w, x_i\rangle)\big)\} =$$

$$-\sum_{i=1}^{\ell}\left\{[y_i = 1]\log\frac{1}{1 + \exp(-\langle w, x\rangle)} + [y_i = -1]\log\left(1 - \frac{1}{1 + \exp(-\langle w, x\rangle)}\right)\right\} =$$

$$-\sum_{i=1}^{\ell}\left\{[y_i = 1]\log\frac{1}{1 + \exp(-\langle w, x\rangle)} + [y_i = -1]\log\left(\frac{1}{1 + \exp(\langle w, x\rangle)}\right)\right\} =$$

$$\sum_{i=1}^{\ell}\{[y_i = 1]\log(1 + \exp(-\langle w, x\rangle)) + [y_i = -1]\log(1 + \exp(\langle w, x\rangle))\} =$$

$$\sum_{i=1}^{\ell}\log(1 + \exp(-y_i\langle w, x_i\rangle))$$

Градиентный спуск

Градиент

- Градиентом функции $f: \mathbb{R}^n \to \mathbb{R}$ называется вектор его частых производных
- $\nabla f(x_1, \dots, x_n) = \left(\frac{\partial f}{\partial x_i}\right)_{i=1}^{n}$
- Пример: $y = x^2 + z$
- $\nabla y(z) = \left(\frac{\partial y}{\partial x}, \frac{\partial y}{\partial z}\right) = (2x, 1\ )$

Градиент

- Градиент - направление наискорейшего роста функции
- Антиградиент ($-\nabla f$) – направление наискорейшего убывания

Градиентный спуск на словах

- Выберем произвольную точку
- Будем шагами двигаться из нее по направлению антиградиента
- Пересчитаем антиградиент
- Улучшилось – идем дальше
- Не улучшилось, останавливаем и считаем, что нашли точку минимума

Градиентный спуск формально

- $w^{(0)}$ - стартовая точка
- $w^{(k)} = w^{(k-1)} - \eta_k \nabla Q(w^{(k-1)})$
- $Q(\mathrm{w}) \, -$ значение функционала ошибки
- η_k - длина шага
- Критерий останова: || $w^{(k)}$ - $w^{(k-1)}$|| < ε

Область эвристик

- Как выбирать начальную точку
- Как выбрать размер шага
- Когда остановиться
- Как оценить градиент

Стохастический градиентный спуск

- $w^{(0)}$ - стартовая точка
- $\nabla Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \nabla q_i(w)$ – полный градиент

  - трудоемко

  - не очень оправдано

Стохастический градиентный спуск

- $w^{(0)}$ - стартовая точка
- $w^{(k)} = w^{(k-1)} - \eta_k \, \nabla q_{i_k}(w^{(k-1)})$

- менее трудоемко

- медленно сходится

- нужен один объект => можно учиться на больших выборках

Примерчик

Девочка Маша измерила вес трех конфет, $y_1 = 6, y_2 = 6, y_3 = 10.$ Маша прошла курс по машинному обучению и решила спрогнозировать вес следующей конфетки моделью $\hat{y} = \beta$.

Помогите Маше оценить β, сделав 3 шага градиентного спуска. Маша думает использовать MSE для обучения

$$\mathrm{L} = MSE = \sum_{i=1}^{n} (y - \beta)^2$$

И думает шагать $\eta = 0.1$

Примерчик

$y_1 = 6, y_2 = 6, y_3 = 10$

$\mathrm{L} = MSE = \sum_{i=1}^{n}(y - \beta)^2$

1) Выберем какую-то стартовую точку. $\beta_0 = 0$
2) Шаг градиентного спуска выглядит так: $w^{(k)} = w^{(k-1)} - \eta_k \nabla q_{i_k}(w^{(k-1)})$, в наших реалиях:

$$\beta_1 = \beta_0 - 0.1 * MSE'_\beta(y_1)$$

Примерчик

$y_1 = 6, y_2 = 6, y_3 = 10$
$\mathrm{L} = MSE = \sum_{i=1}^{n}(y - \beta)^2$

$\beta_0 = 0$
$\beta_1 = \beta_0 - MSE'_\beta(y_{1,} \beta_0)$

$$MSE'_\beta = \left((y - \beta)^2\right)' = -2 * (y - \beta)$$

$$\beta_1 = 0 - 0.1\big(-2 * (6 - 0)\big) = 1.2$$

Примерчик

$y_1 = 6, y_2 = 6, y_3 = 10$

$\mathrm{L} = MSE = \sum_{i=1}^{n}(y - \beta)^2$

$\beta_0 = 0$

$\beta_1 = 1.2$

$\beta_2 = \beta_1 - MSE'_\beta(y_{2,}\,\beta_1)$

$$\beta_2 = 1.2 - \big(-2 * (6 - 1.2)\big) = 4.8$$

Примерчик

$y_1 = 6, y_2 = 6, y_3 = 10$

$\mathrm{L} = MSE = \sum_{i=1}^{n}(y - \beta)^2$

$\beta_0 = 0$

$\beta_1 = 1.2$

$\beta_2 = 4.8$

$\beta_3 = \beta_2 - 0.1 * MSE'_{\beta}(y_{3,}\, \beta_1)$

$$\beta_3 = 4.8 - \left(-2 * (6 - 4.8)\right) = 7{,}2$$

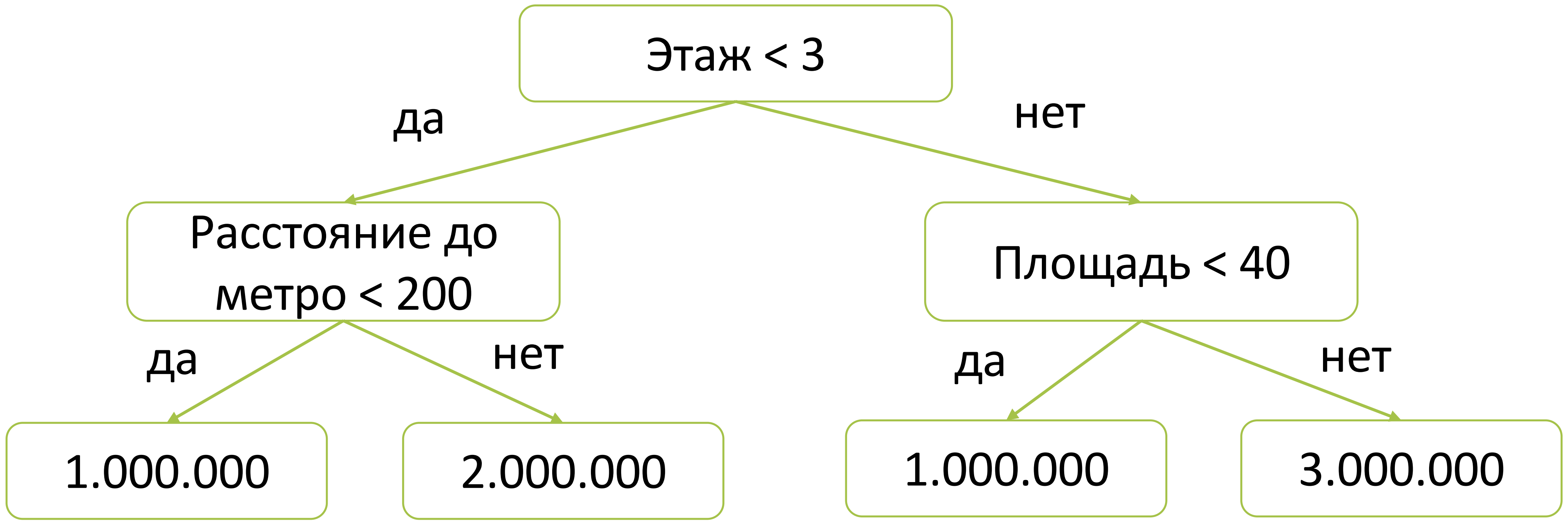
Решающие деревья

Логические правила

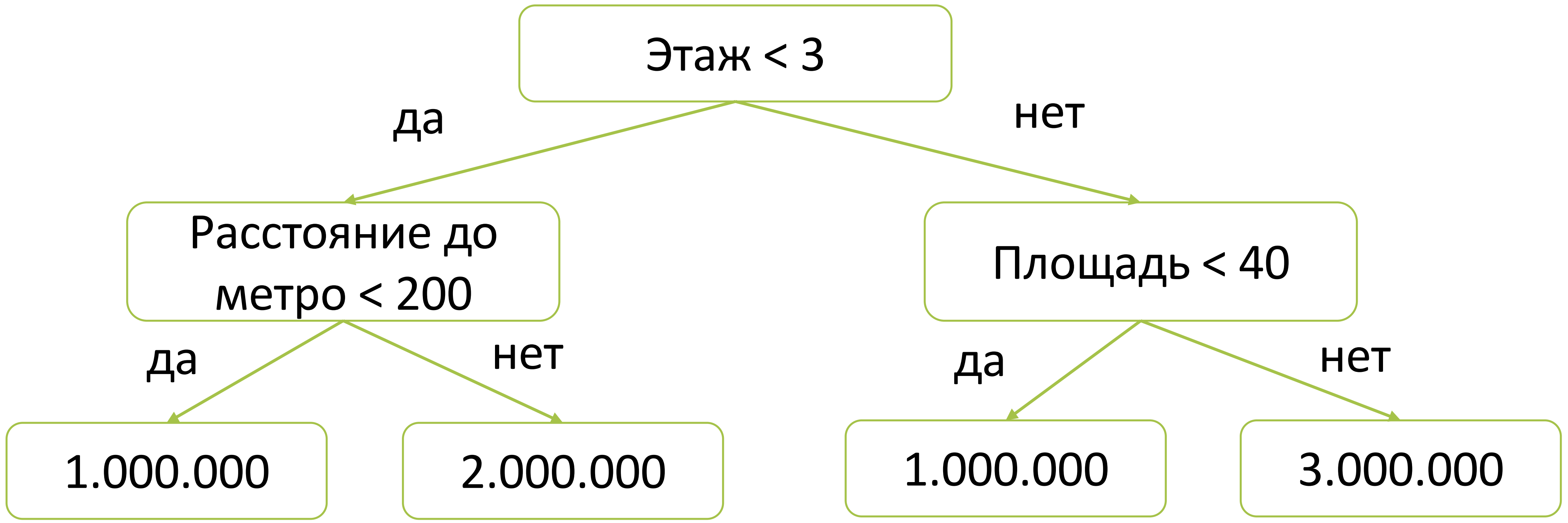
- $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][500 < \text{расстояние до метро} < 1000]$
- Легко объяснить, как работают
- Находят нелинейные закономерности

- Нужно как-то искать хорошие логические правила
- Нужно уметь составлять модели из логических правил

Решающее дерево

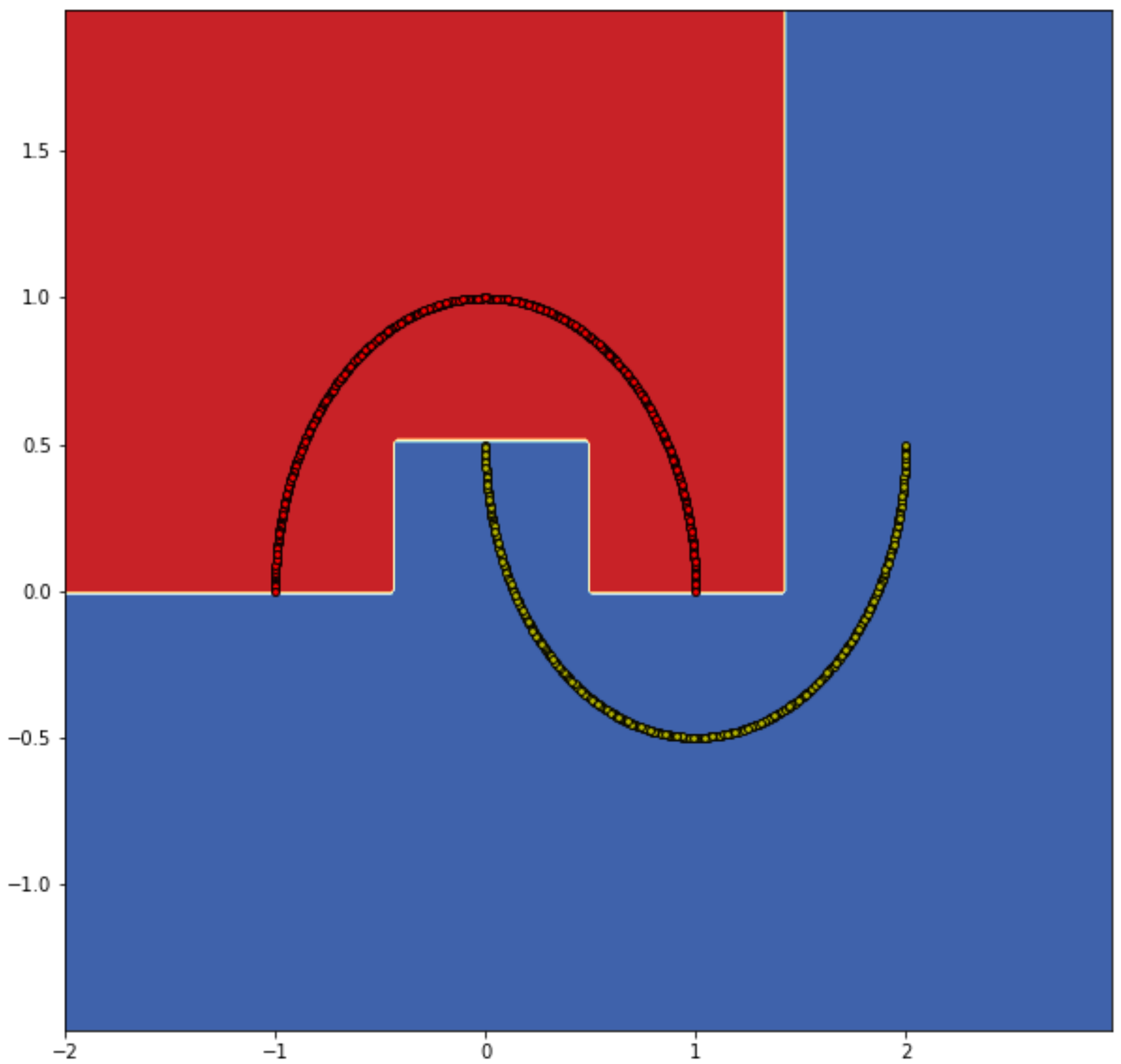


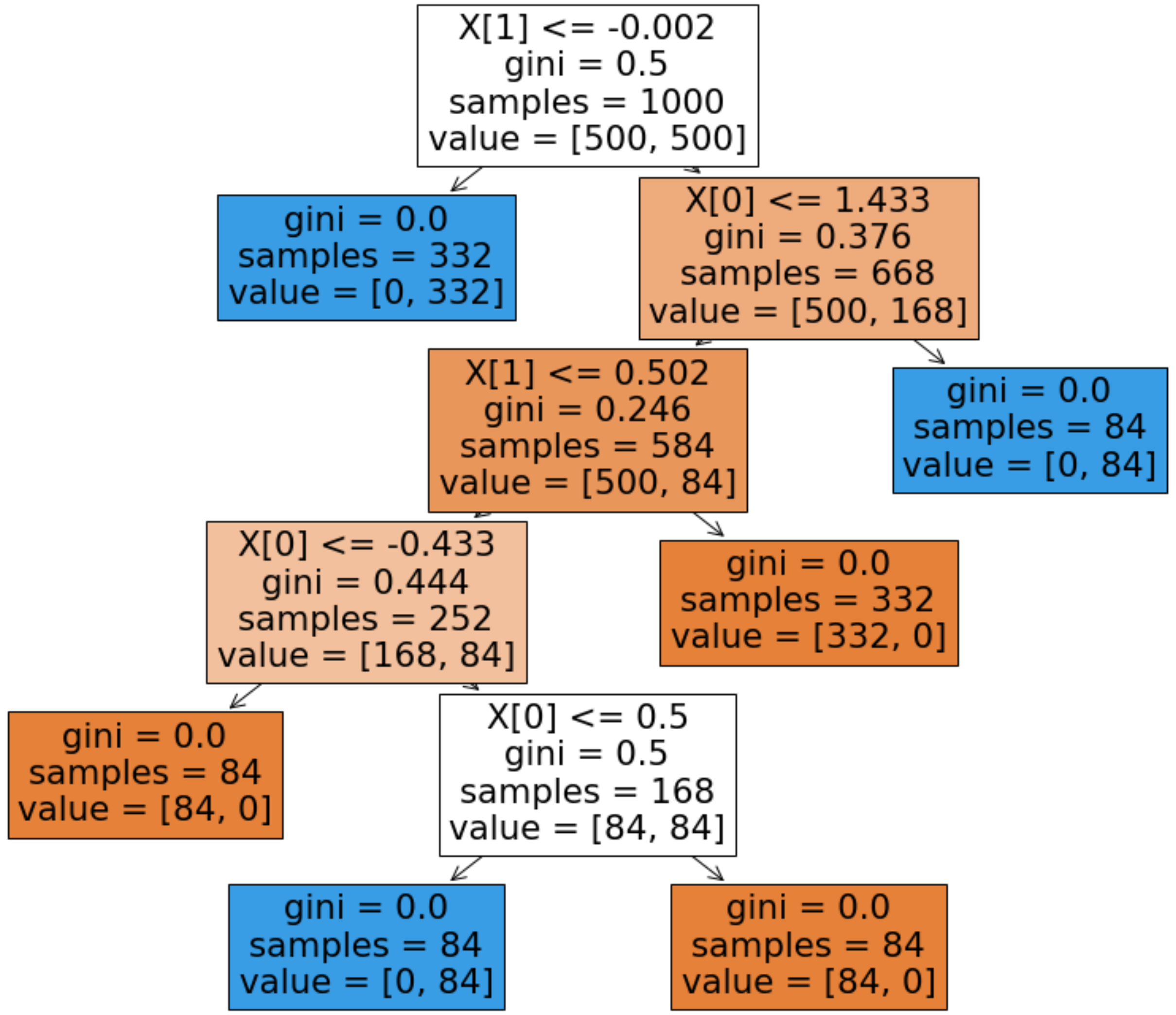
Решающее дерево



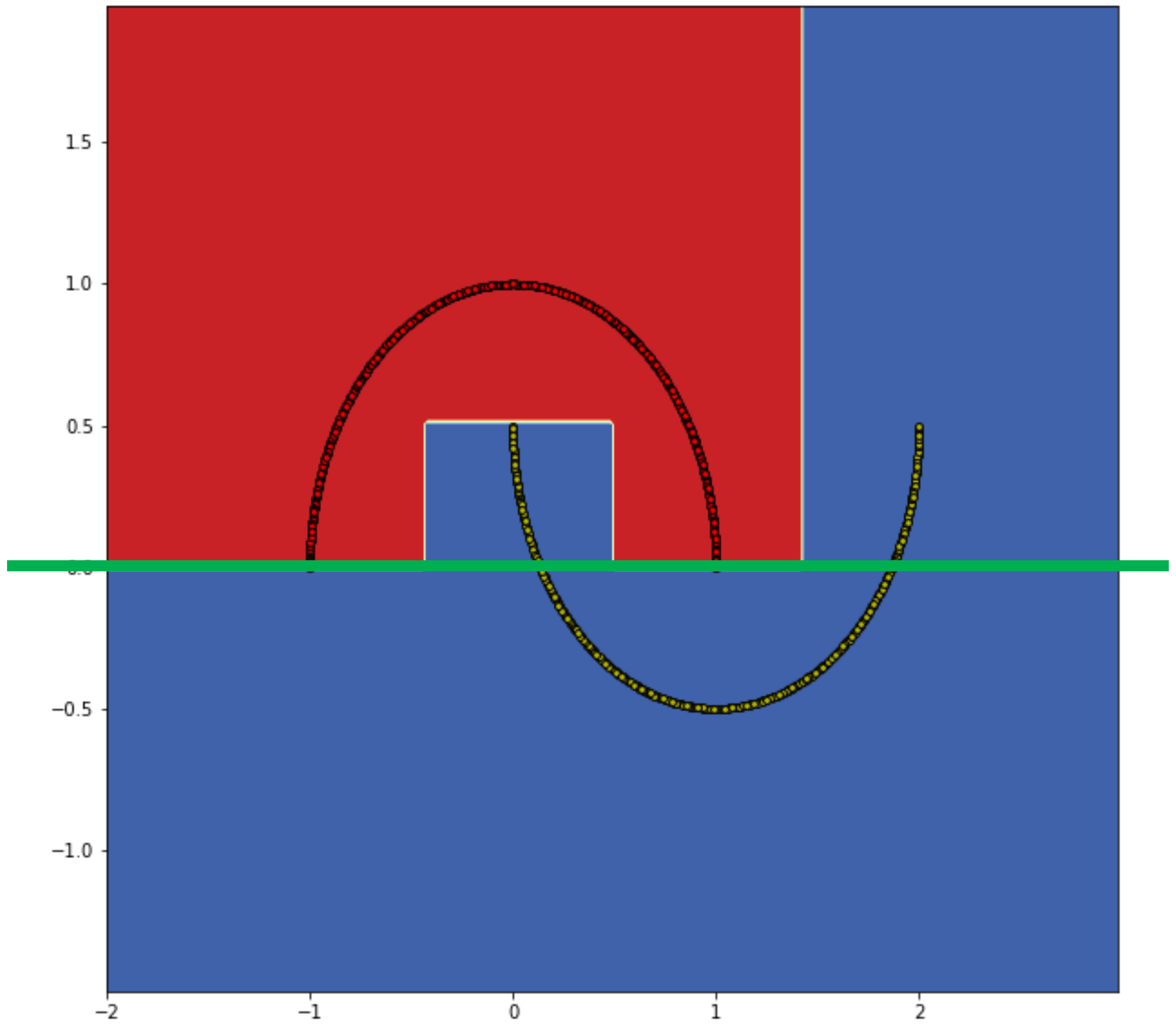
- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

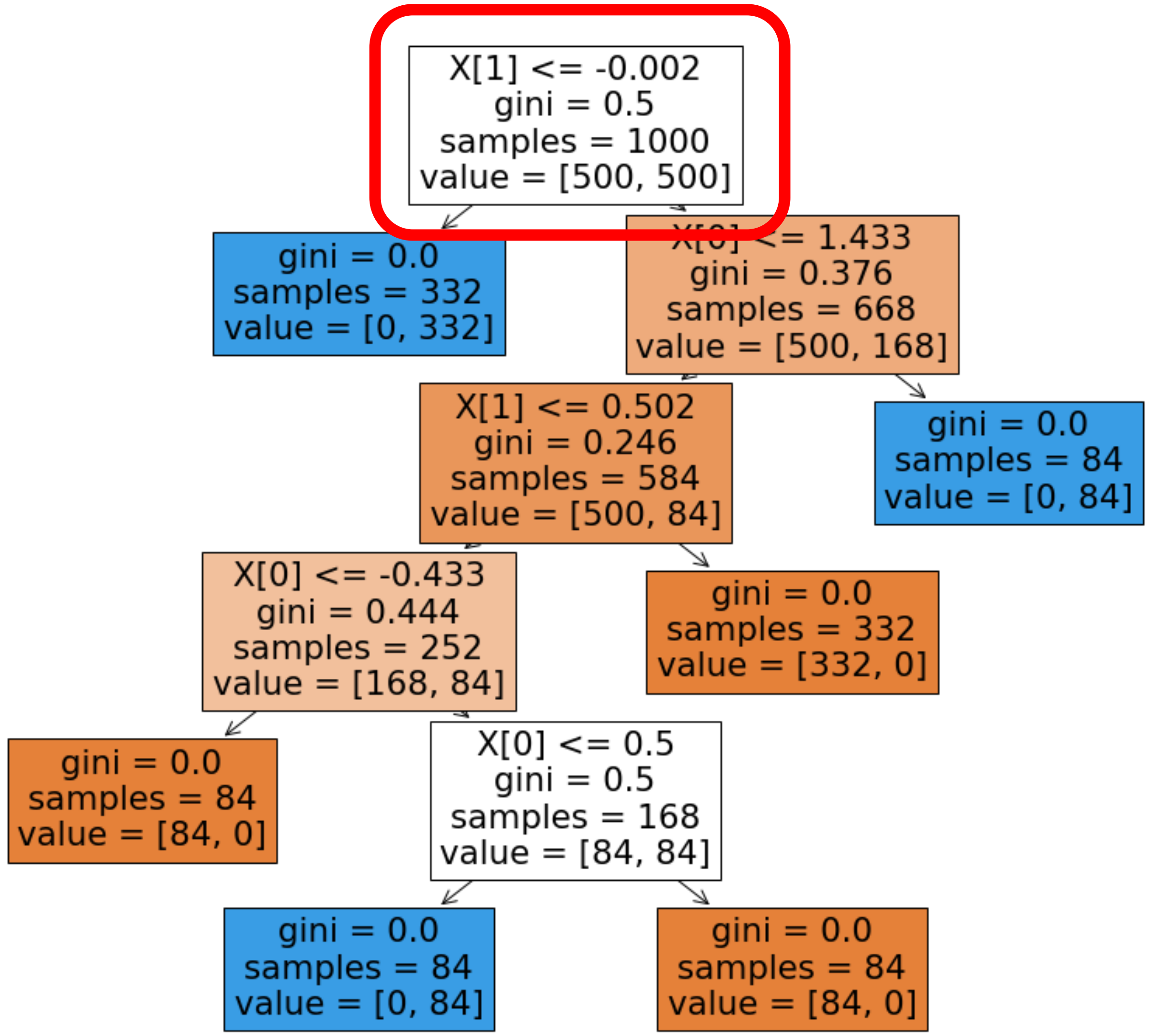
Решающее дерево



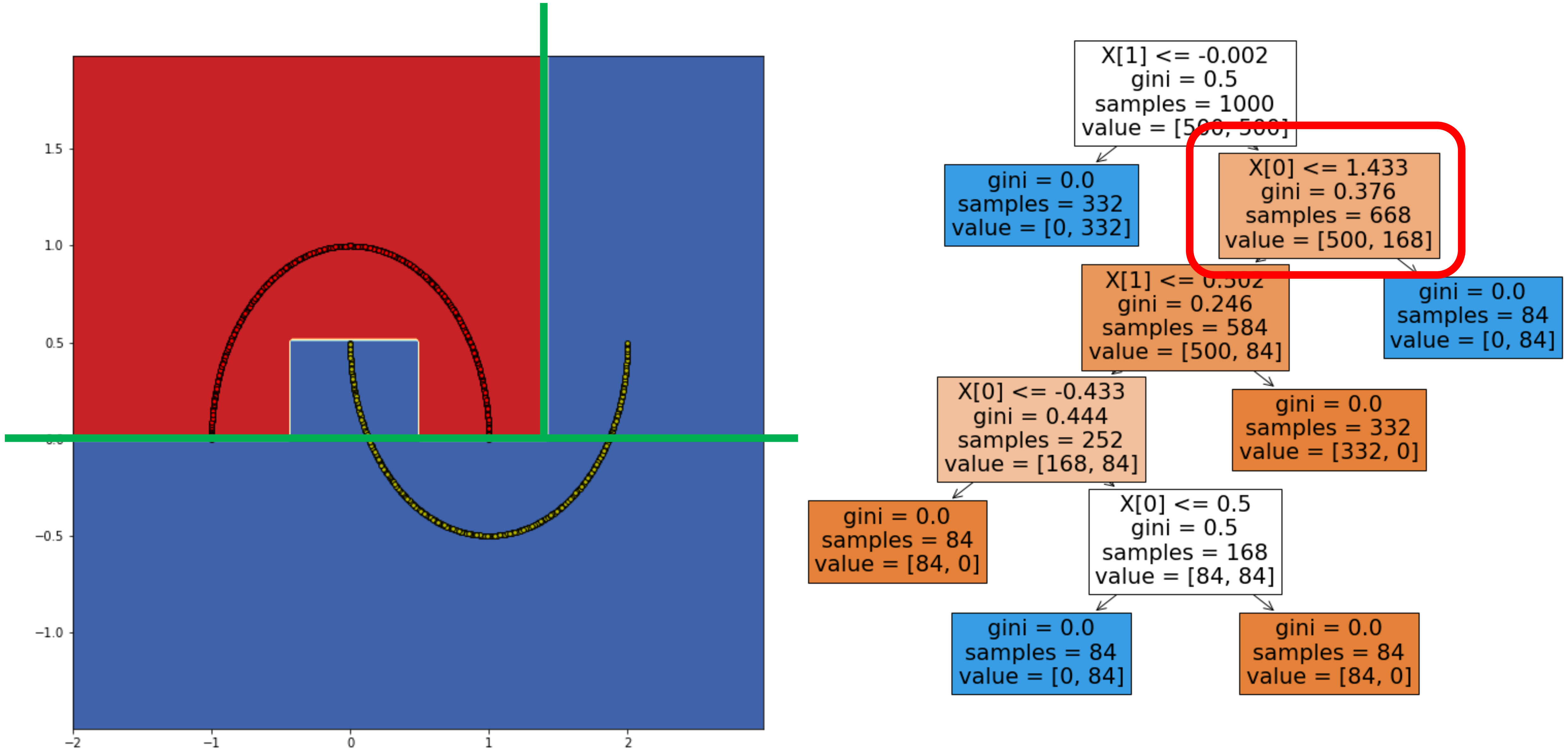


Решающее дерево

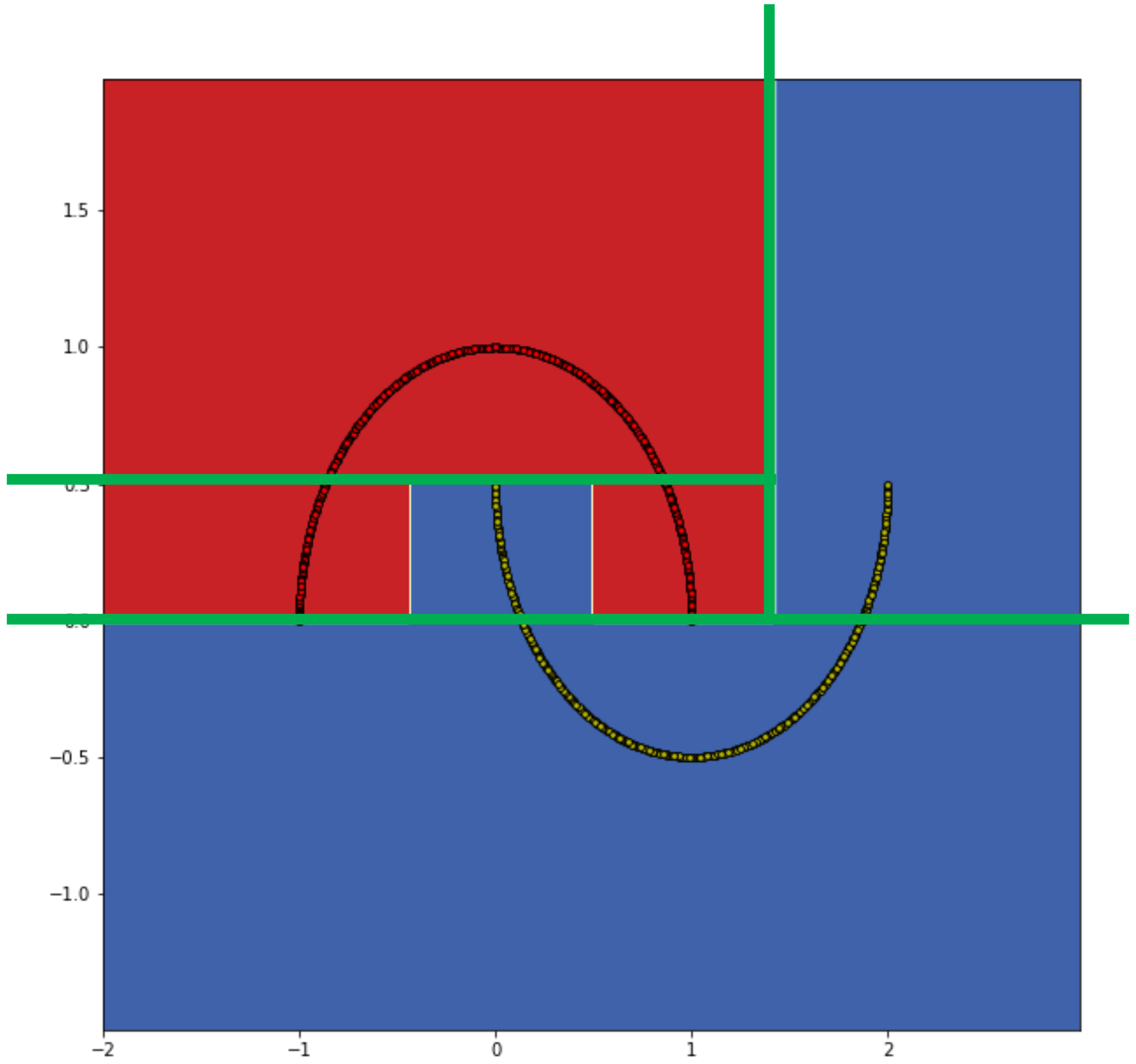


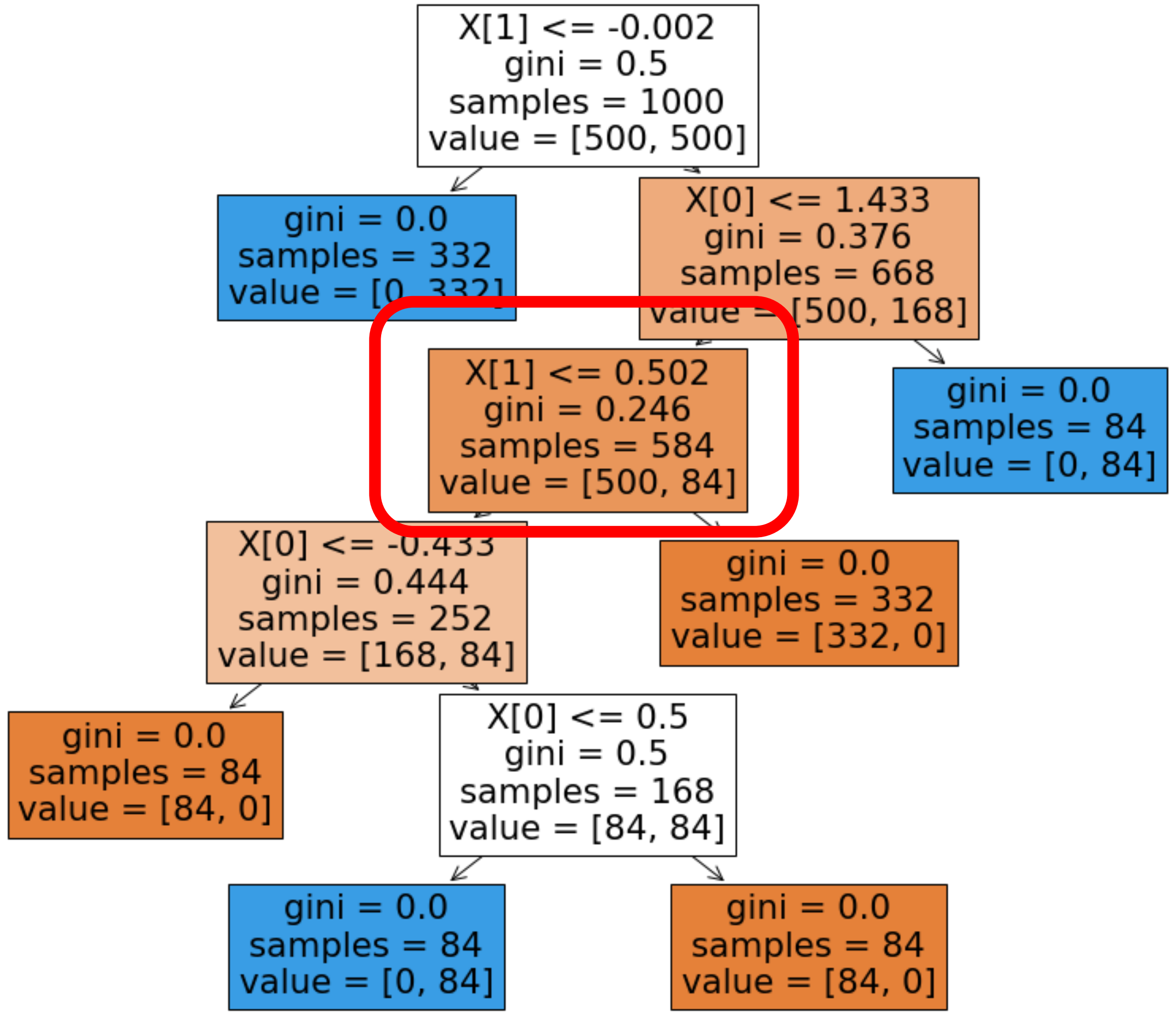


Решающее дерево

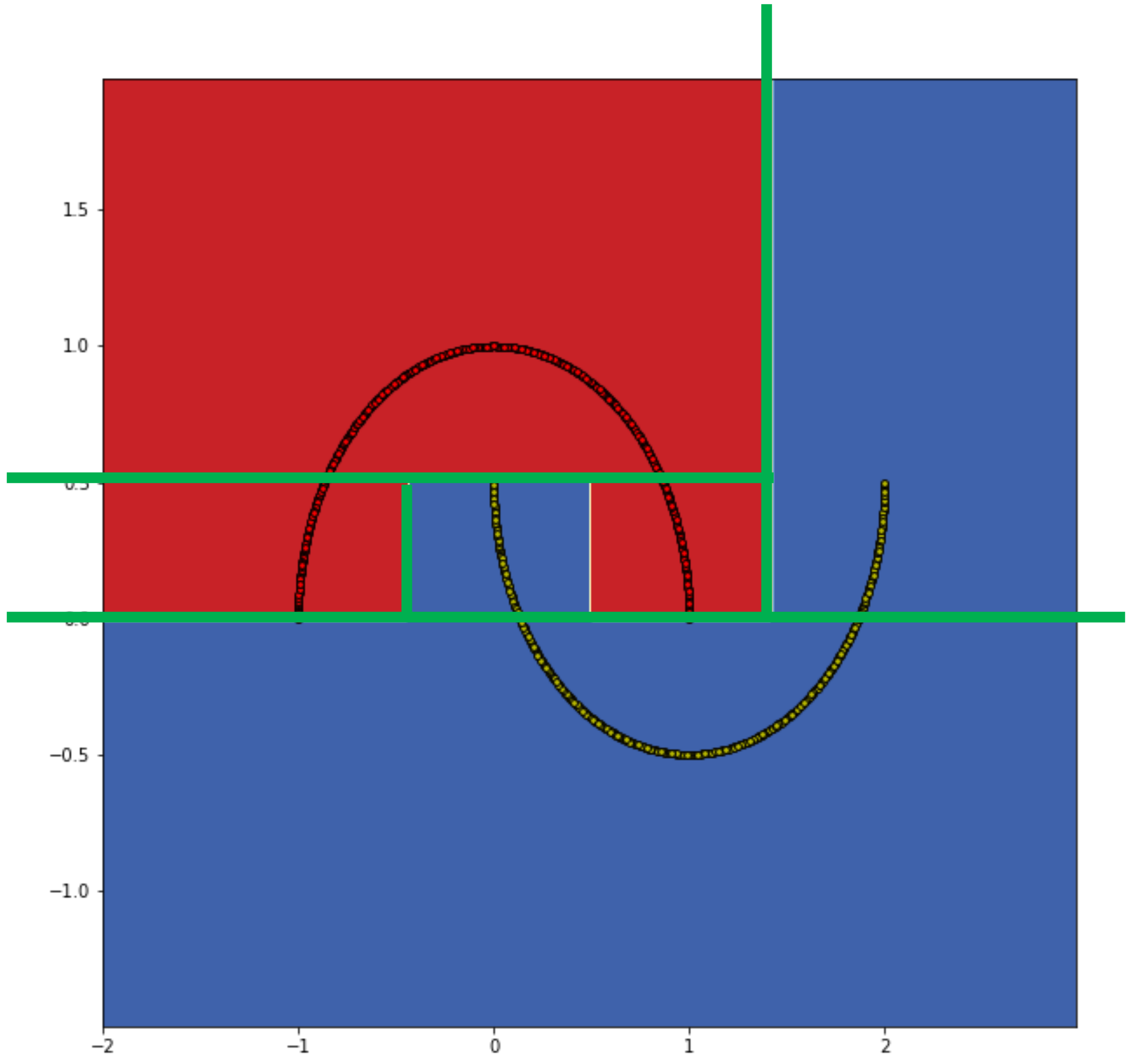


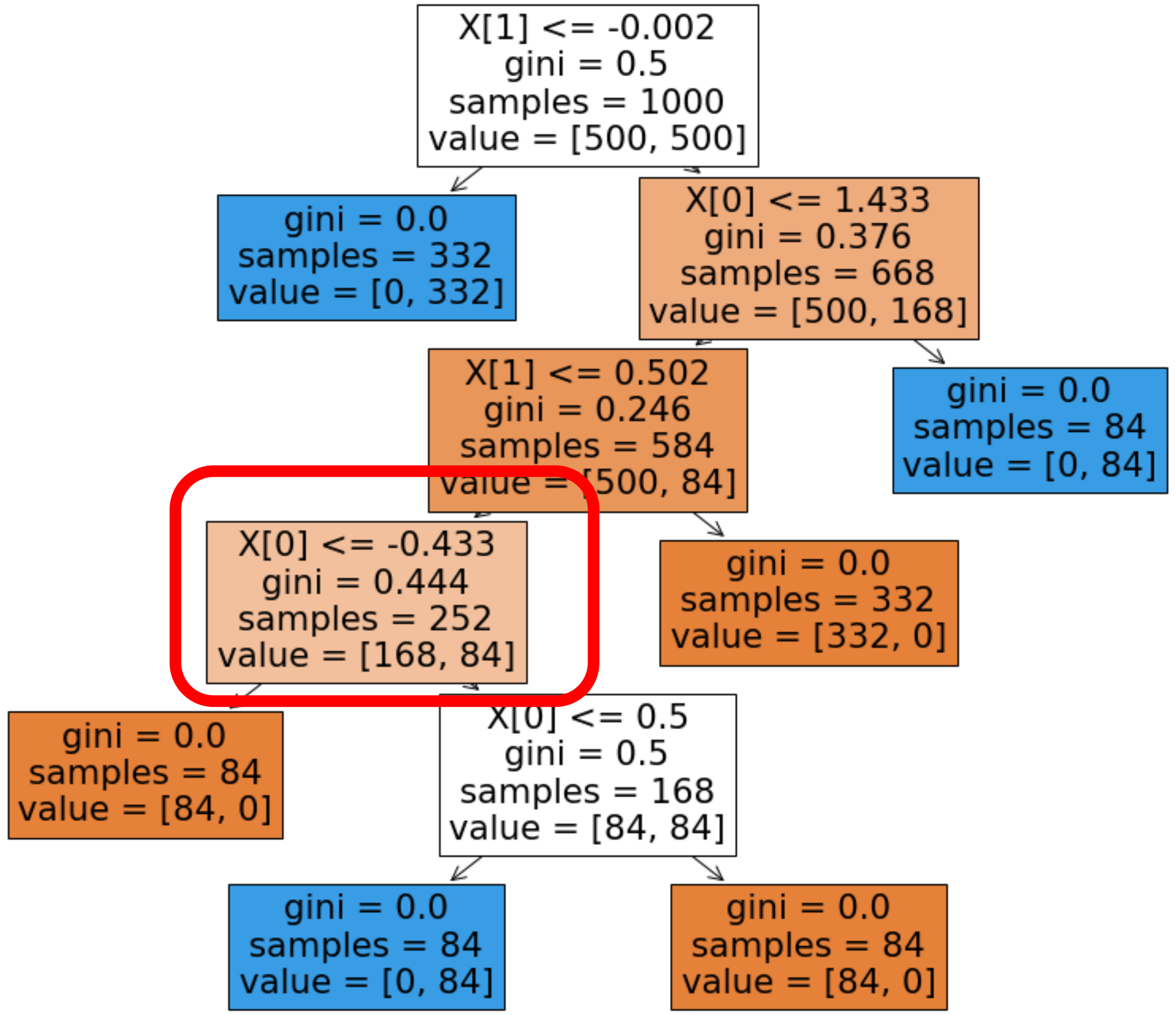
Решающее дерево



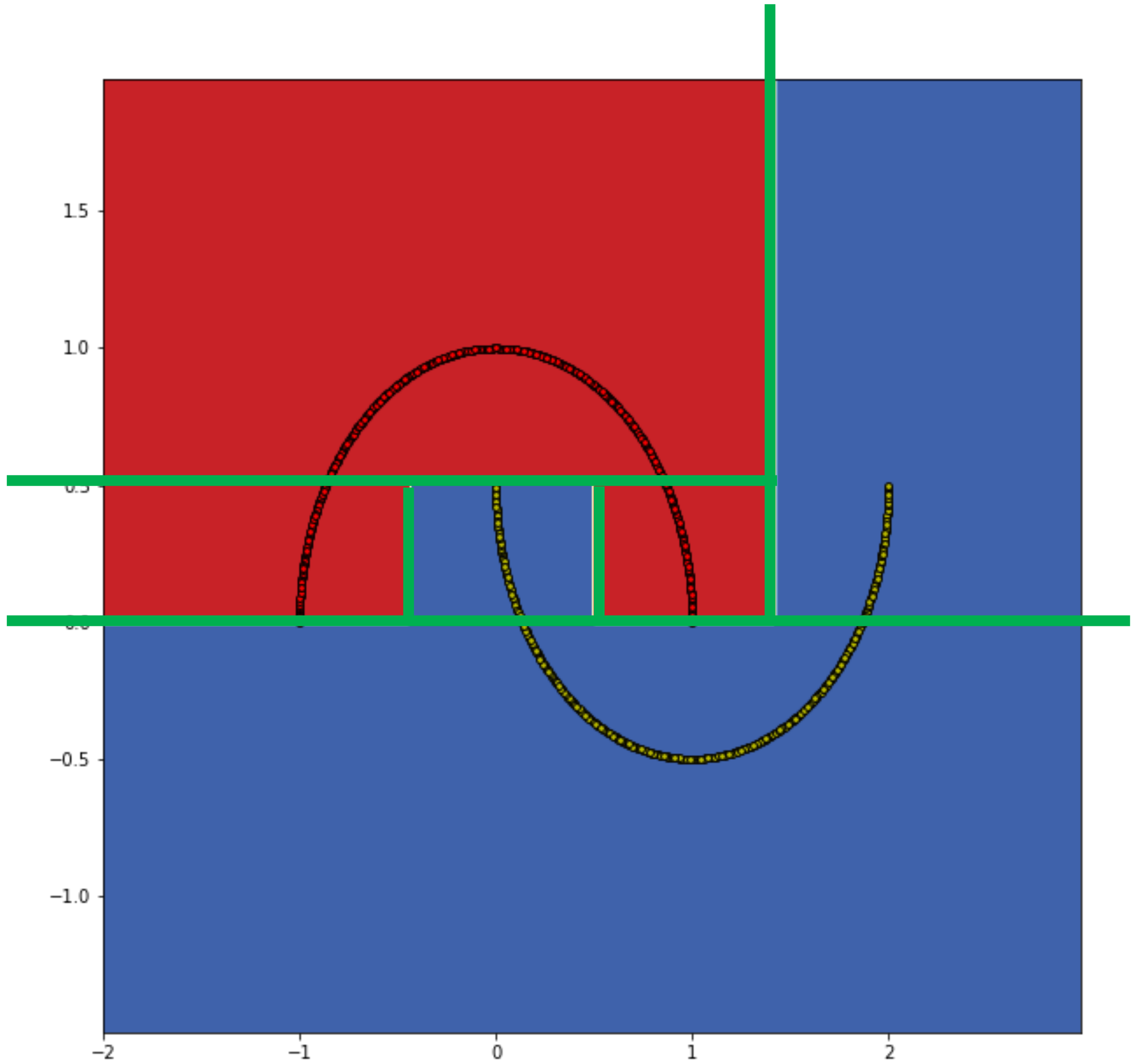


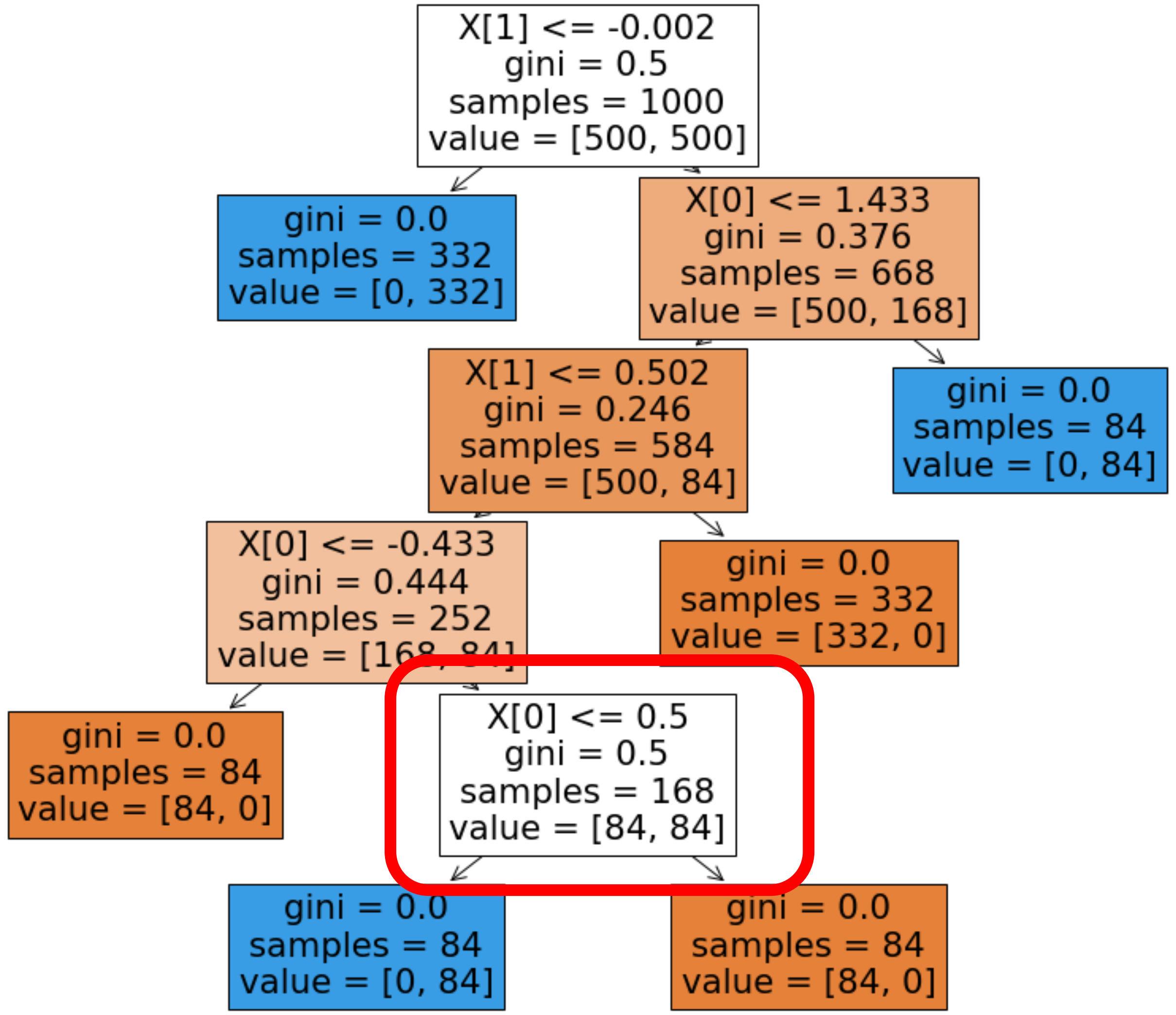
Решающее дерево



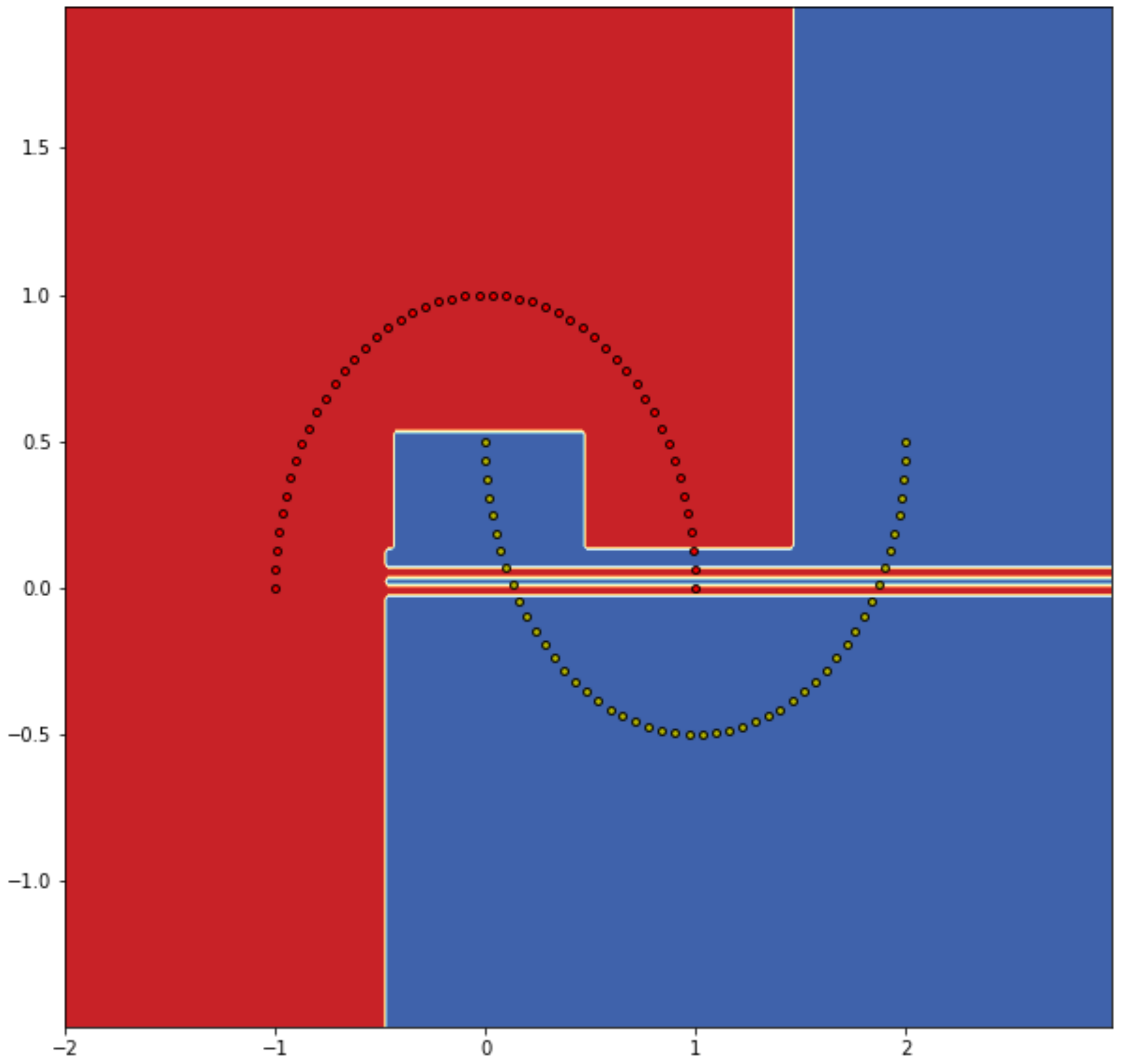


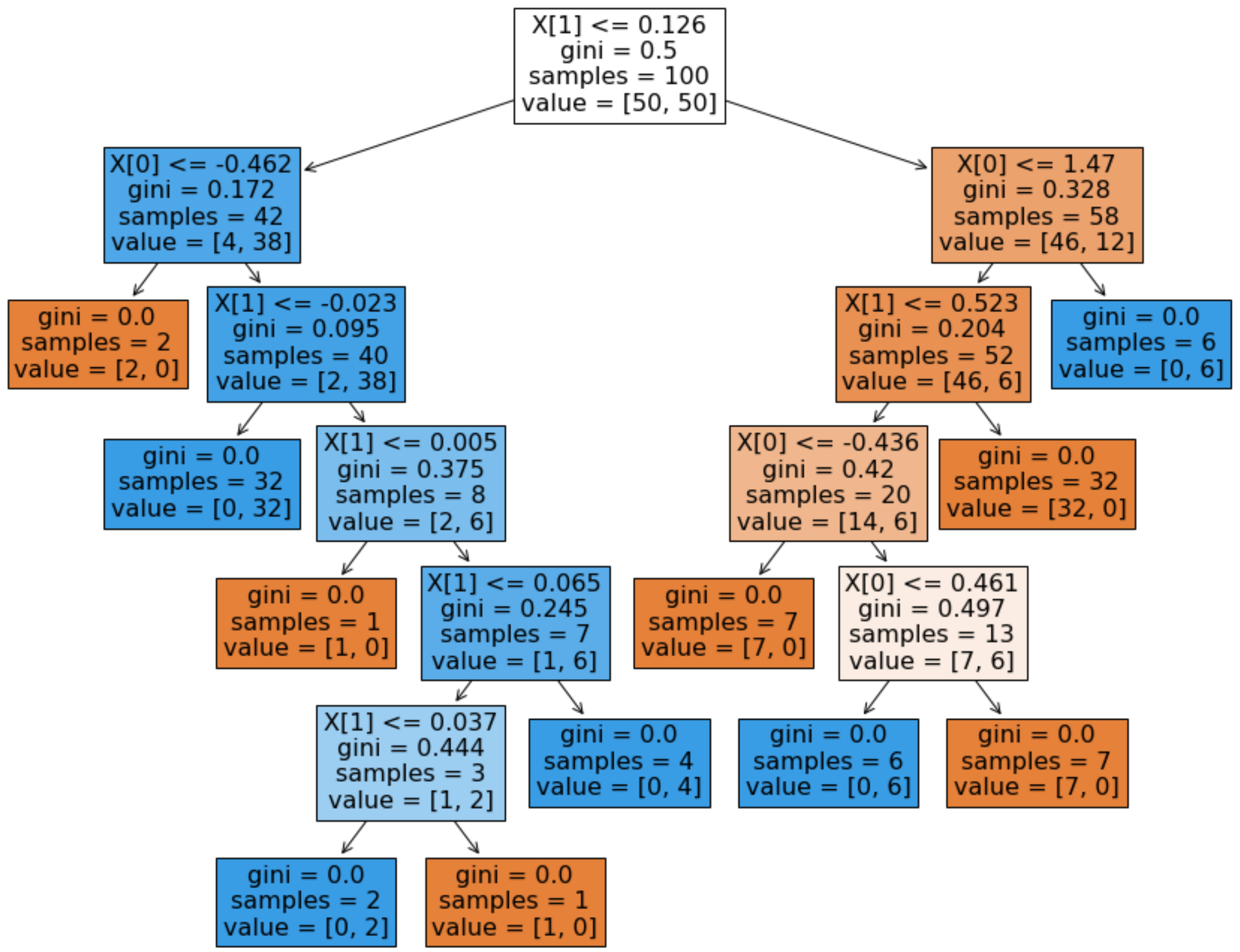
Решающее дерево





Решающее дерево





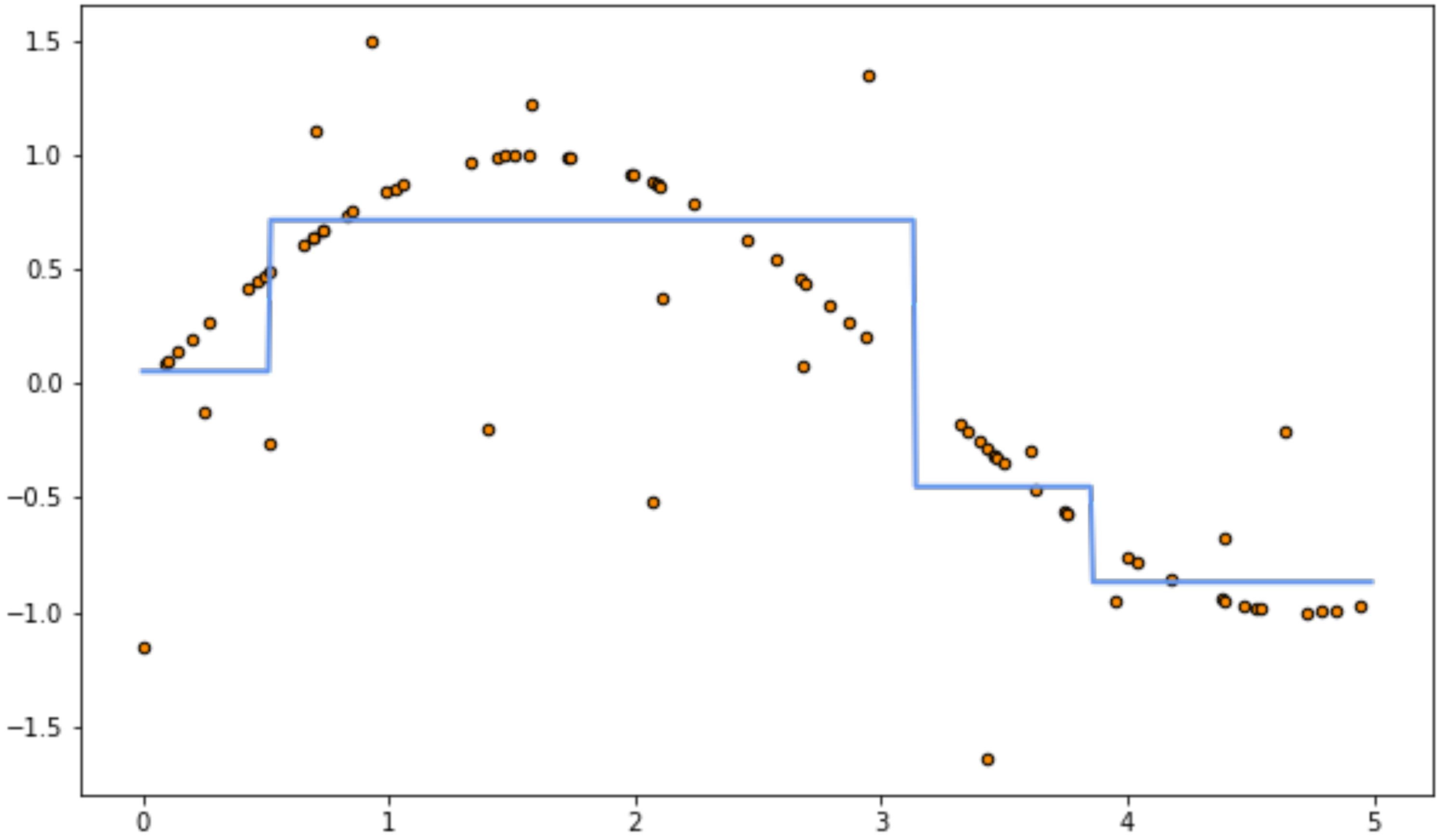
Сложность дерева

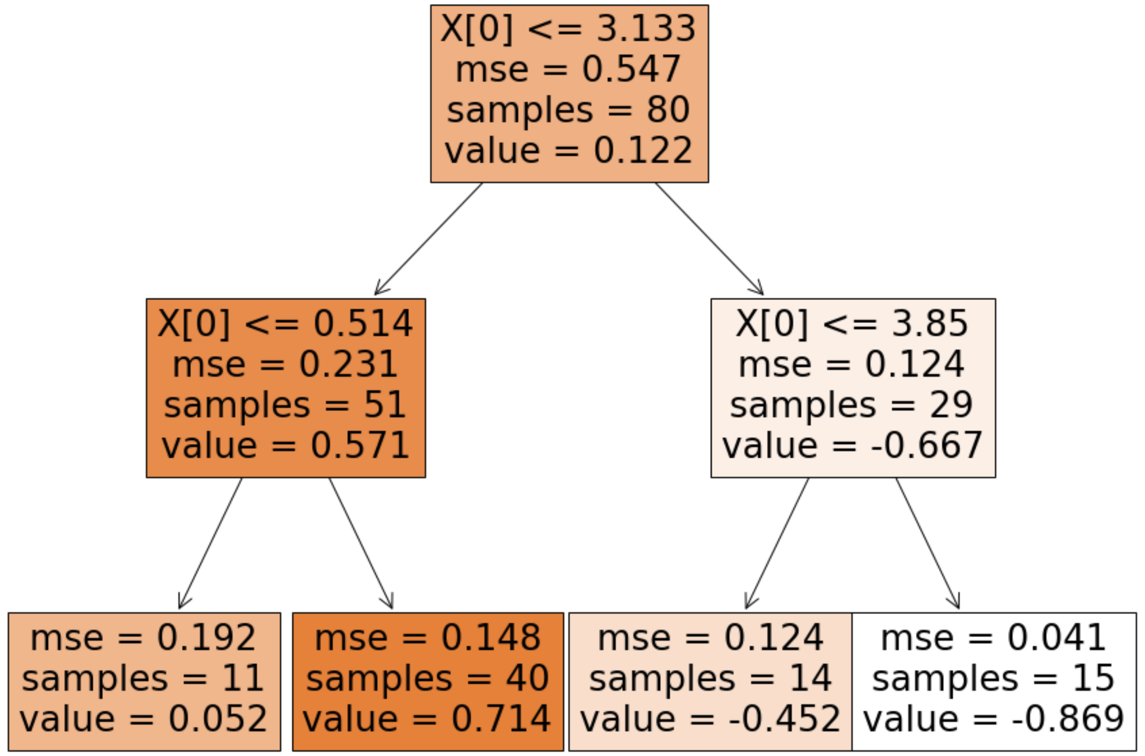
- Решающее дерево можно строить до тех пор, пока каждый лист не будет соответствовать ровно одному объекту

- Деревом можно идеально разделить любую выборку!

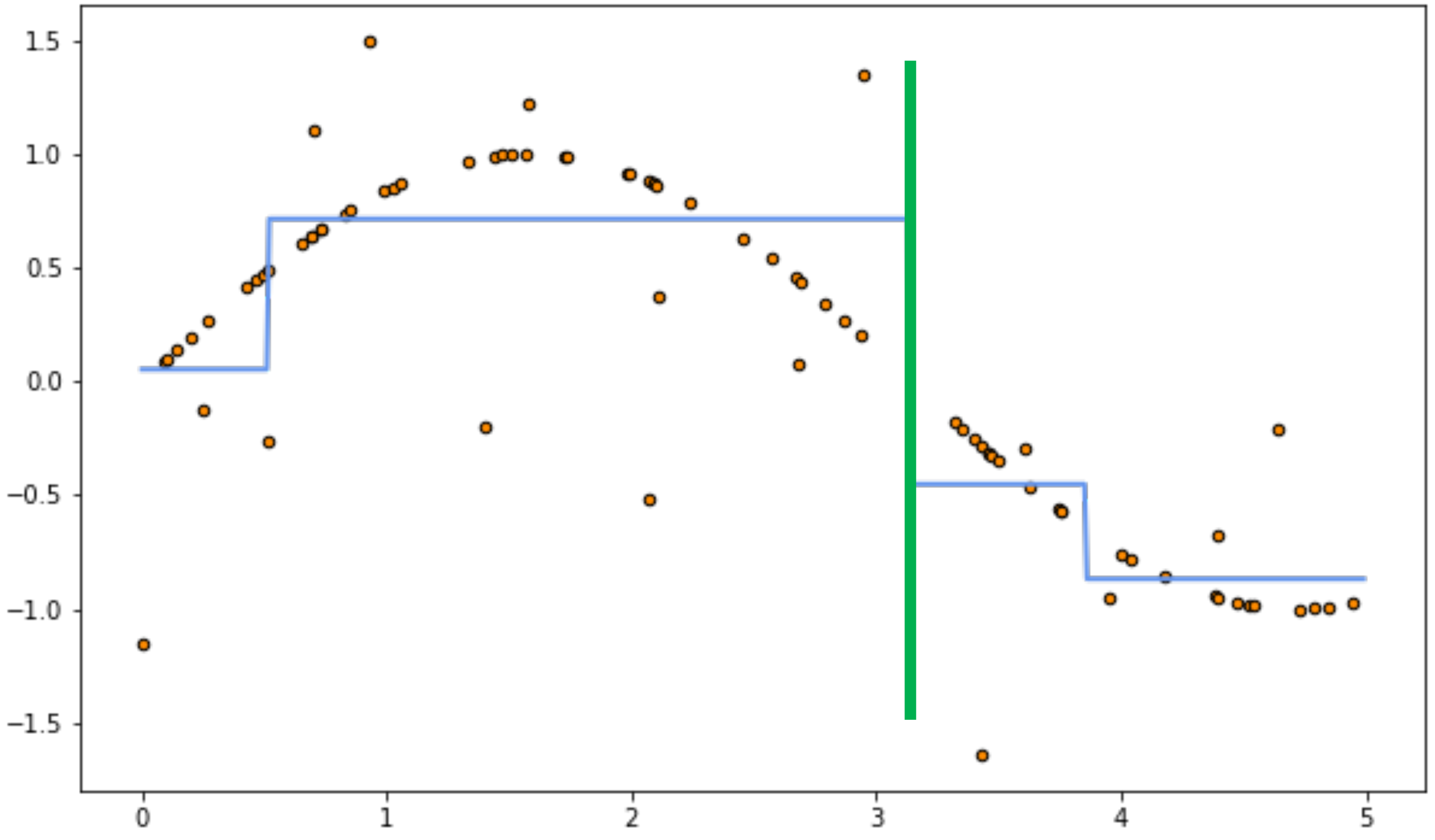
- Если только нет объектов с одинаковыми признаками, но разными ответами

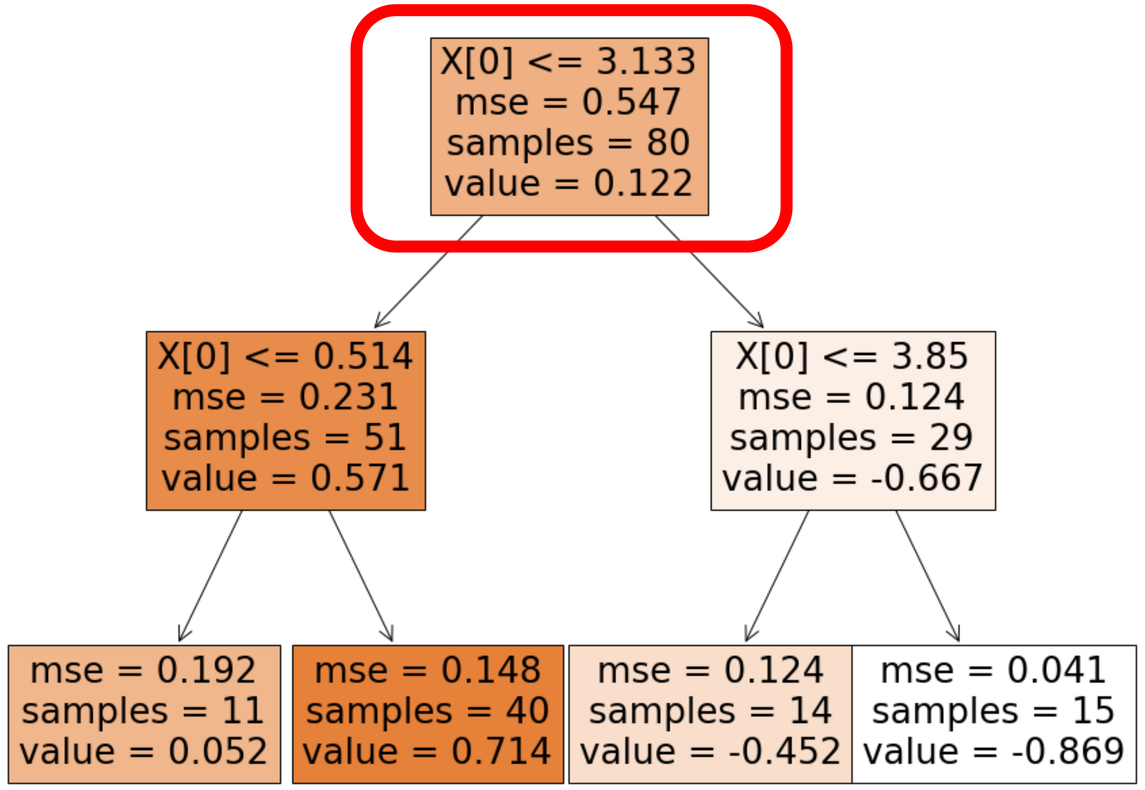
Решающее дерево для регрессии



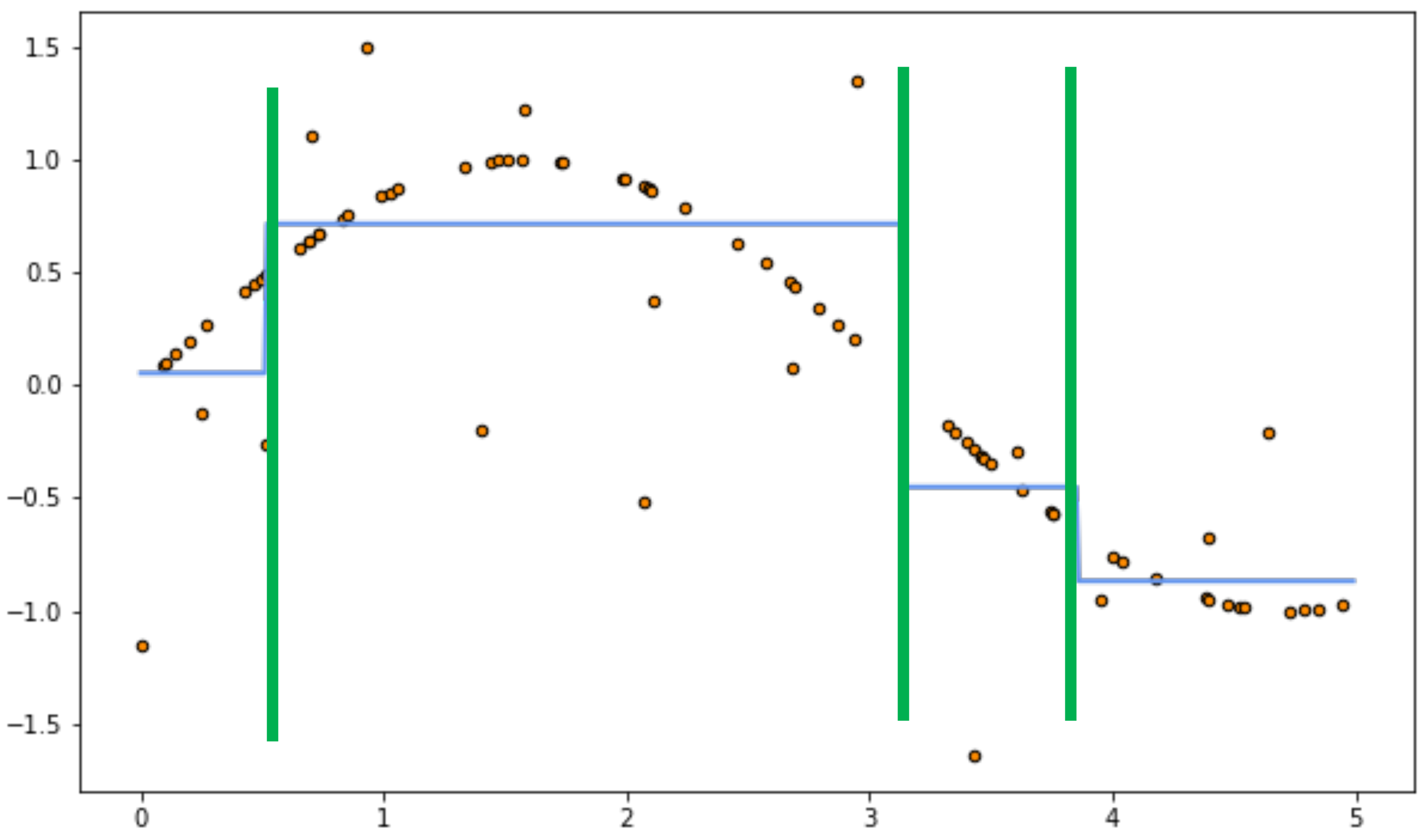


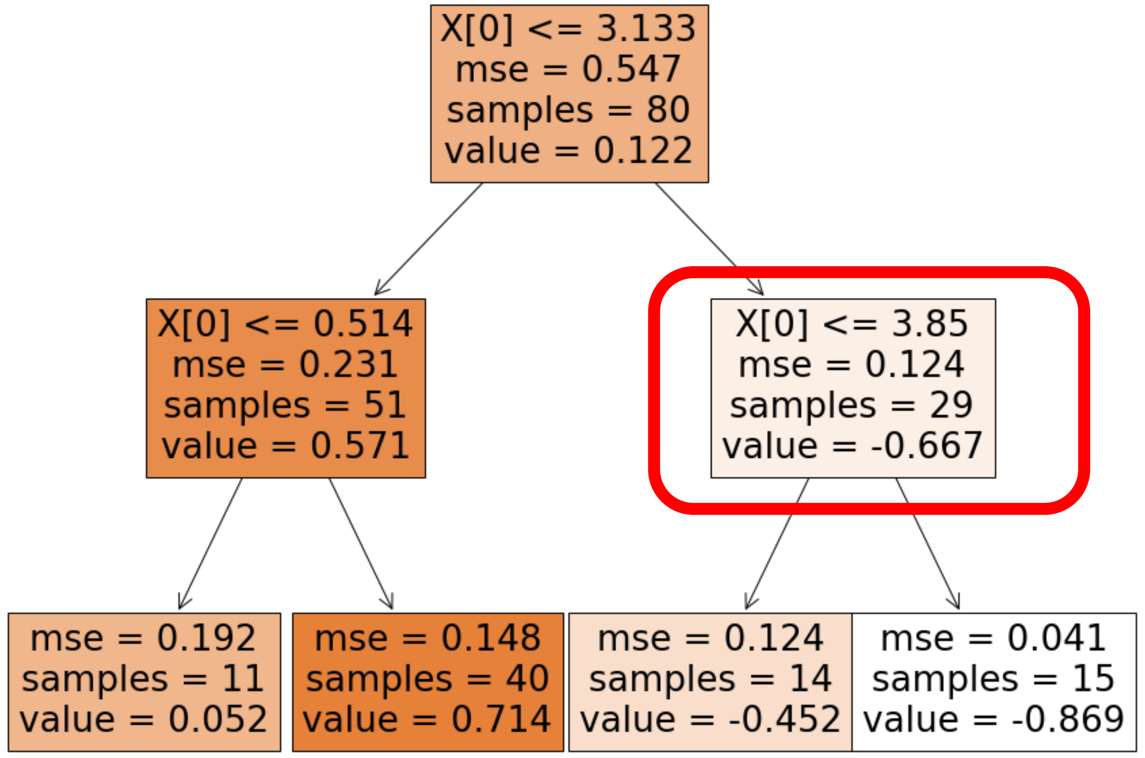
Решающее дерево для регрессии



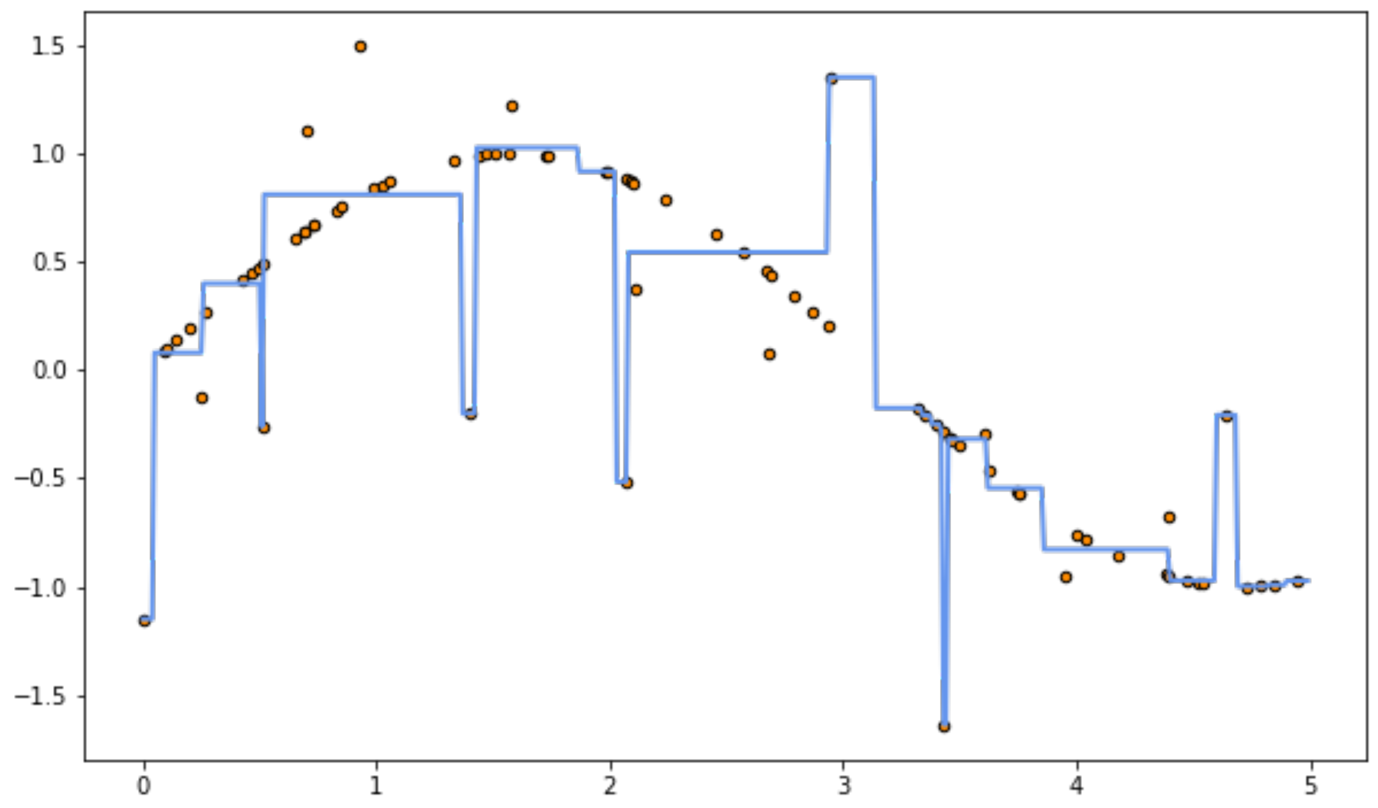


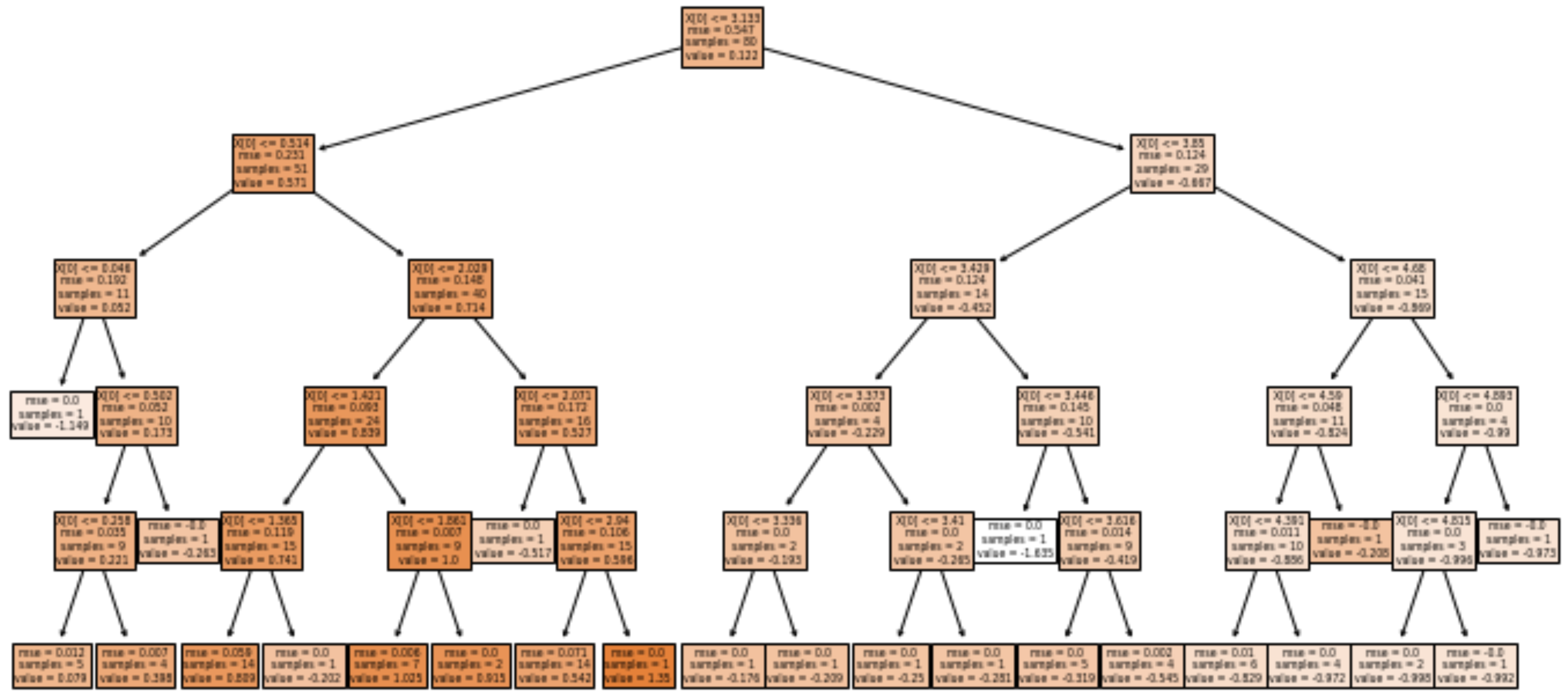
Решающее дерево для регрессии



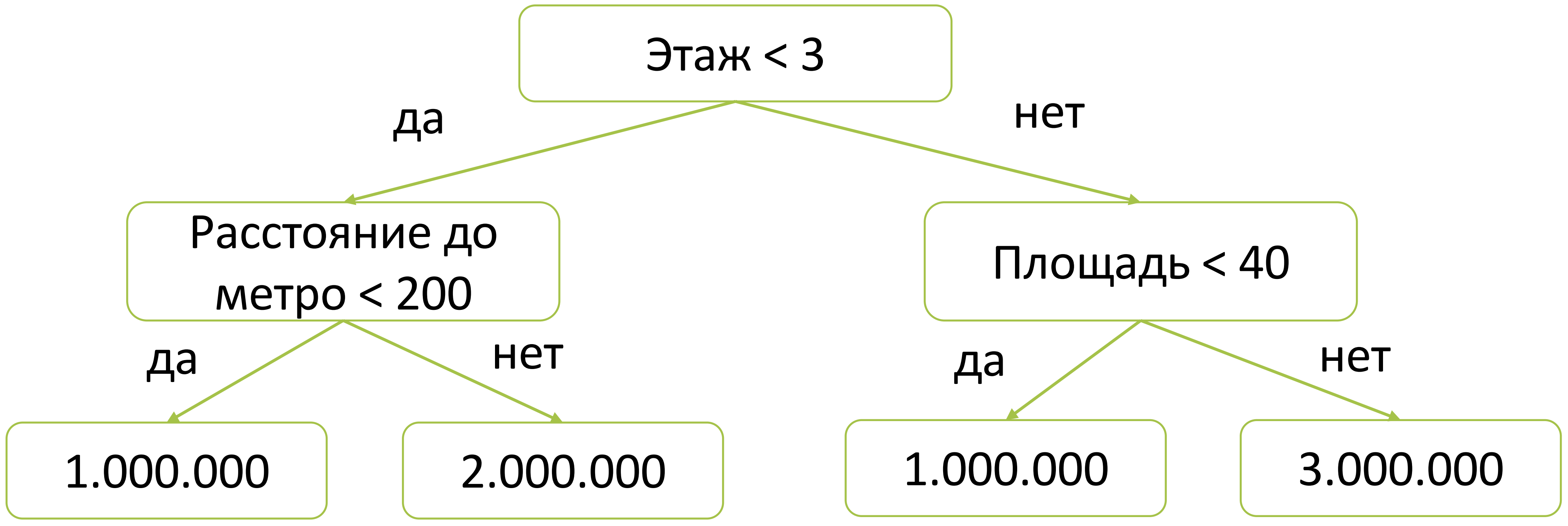


Решающее дерево для регрессии





Решающее дерево



- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

Предикаты

- Порог на признак $\left[x_j < t\right]$ — не единственный вариант
- Предикат с линейной моделью: $[\langle w, x \rangle < t]$
- Предикат с метрикой: $[\rho(x, x_0) < t]$
- И много других вариантов
- Но даже с простейшим предикатом можно строить очень сложные модели

Прогнозы в листьях

- Наш выбор: константные прогнозы $c_v \in \mathbb{Y}$
- Регрессия:

$$c_v = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} y_i$$

- Классификация:

$$c_v = \arg\max_{k \in \mathbb{Y}} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

Прогнозы в листьях

- Наш выбор: константные прогнозы $c_v \in \mathbb{Y}$
- Классификация и вероятности классов:

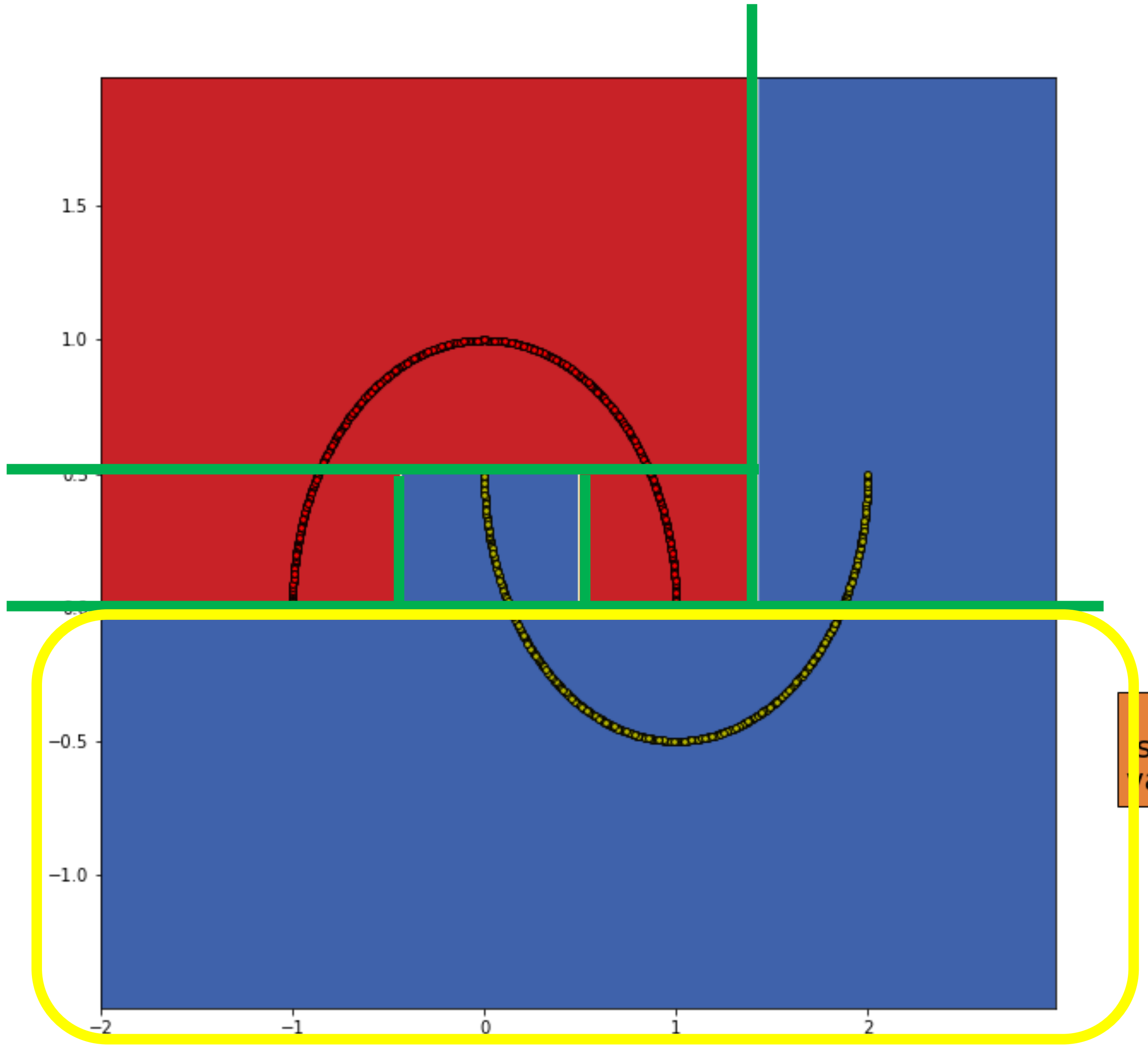
$$c_{vk} = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

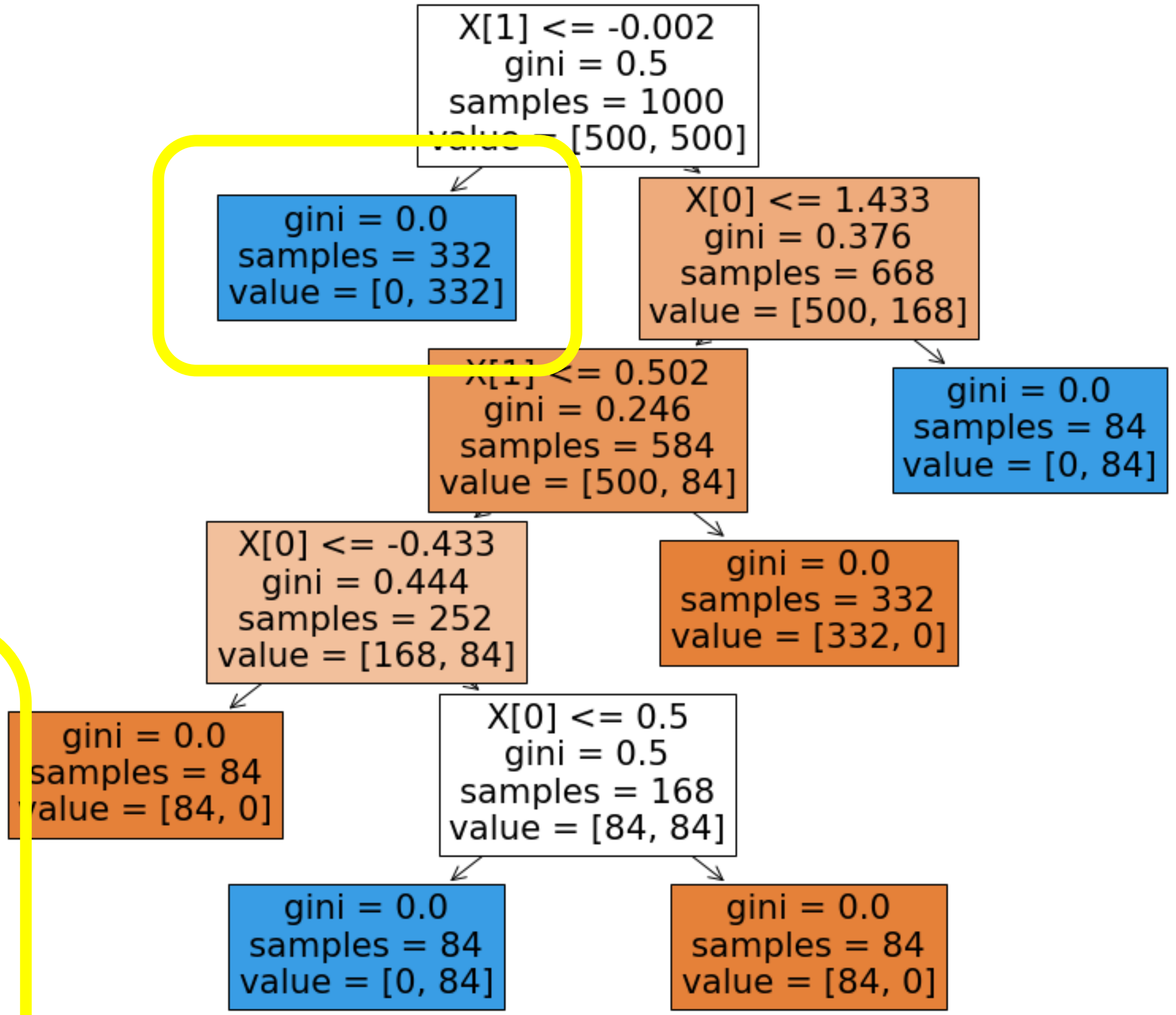
Прогнозы в листьях

- Можно усложнять листья
- Например:

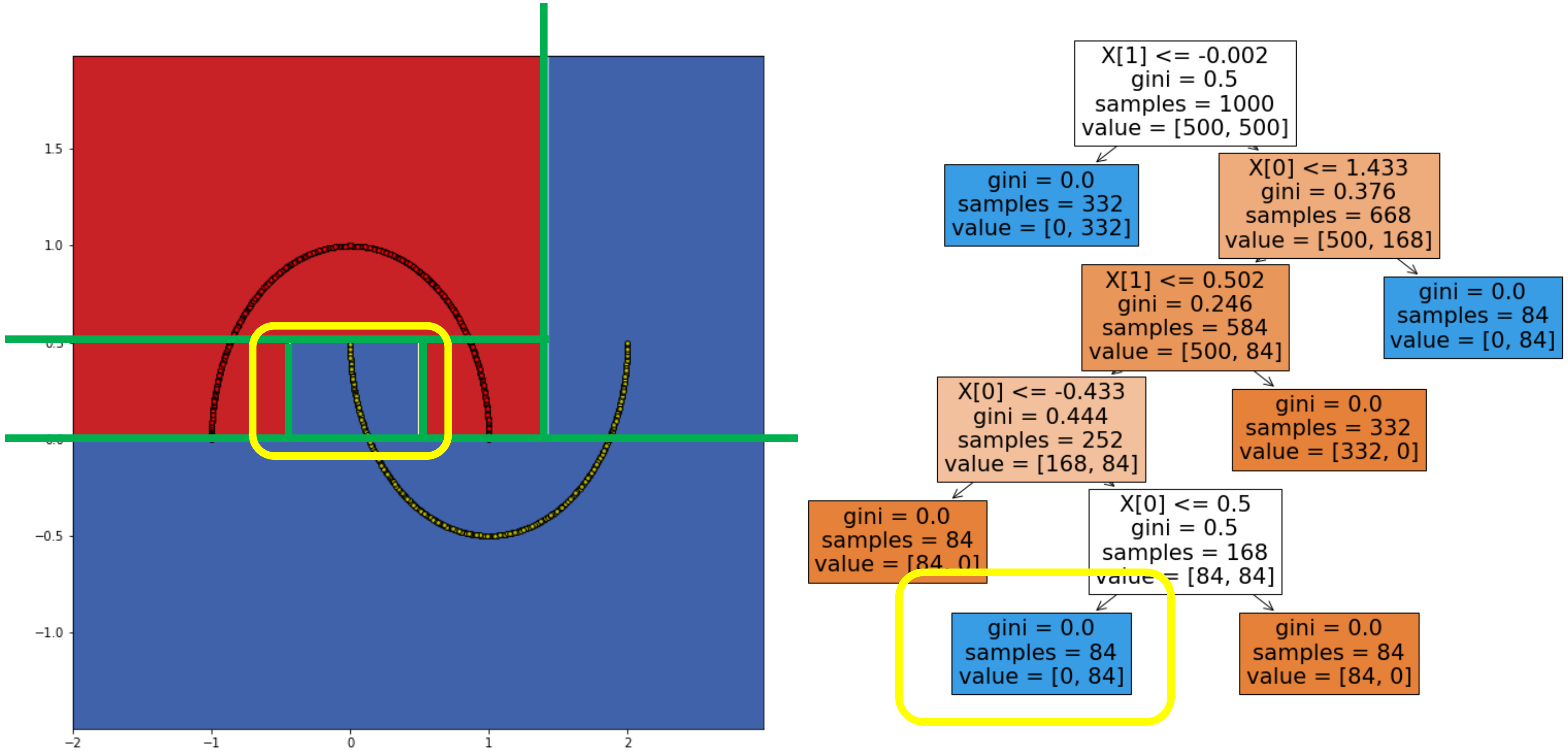
$$c_v(x) = \langle w_v, x \rangle$$

Решающее дерево





Решающее дерево



Формула для дерева

- Дерево разбивает признаковое пространство на области $R_1, \ldots, R_J$
- Каждая область R_j соответствует листу
- В области R_j прогноз c_j константный

$$a(x) = \sum_{j=1}^{J} c_j \left[x \in R_j\right]$$

Формула для дерева

$$a(x) = \sum_{j=1}^{J} c_j \left[x \in R_j\right]$$

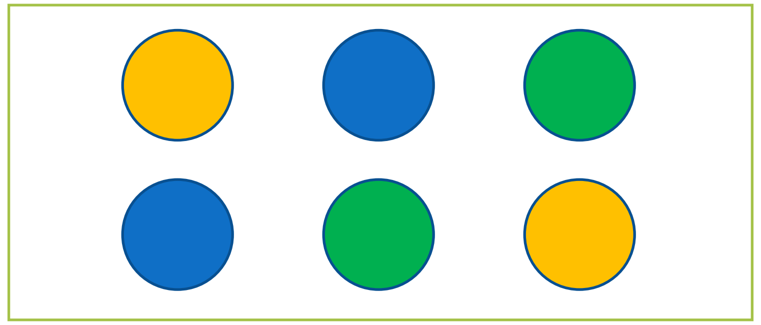
- Решающее дерево находит хорошие новые признаки
- Над этими признаками подбирает линейную модель

Как выбирать предикаты

Жадное построение

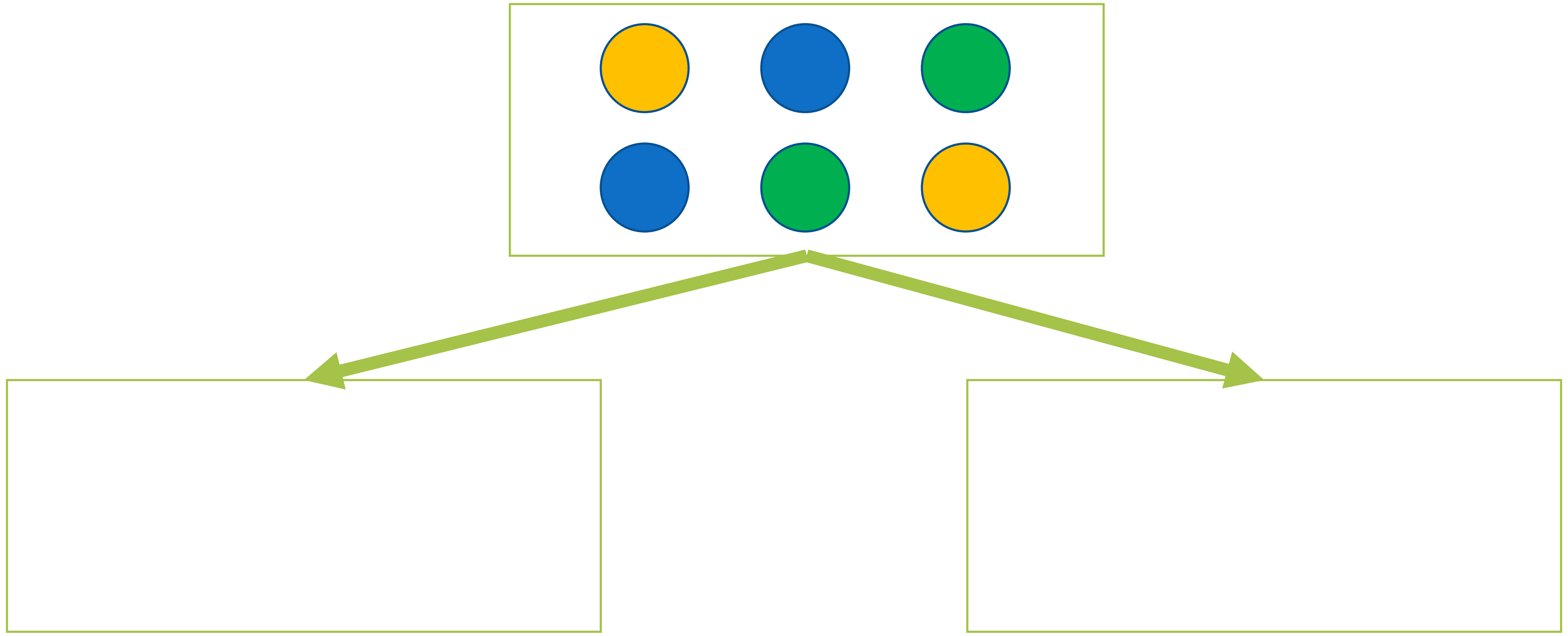
- Разберёмся на примере
- Начнём с задачи классификации

Жадное построение

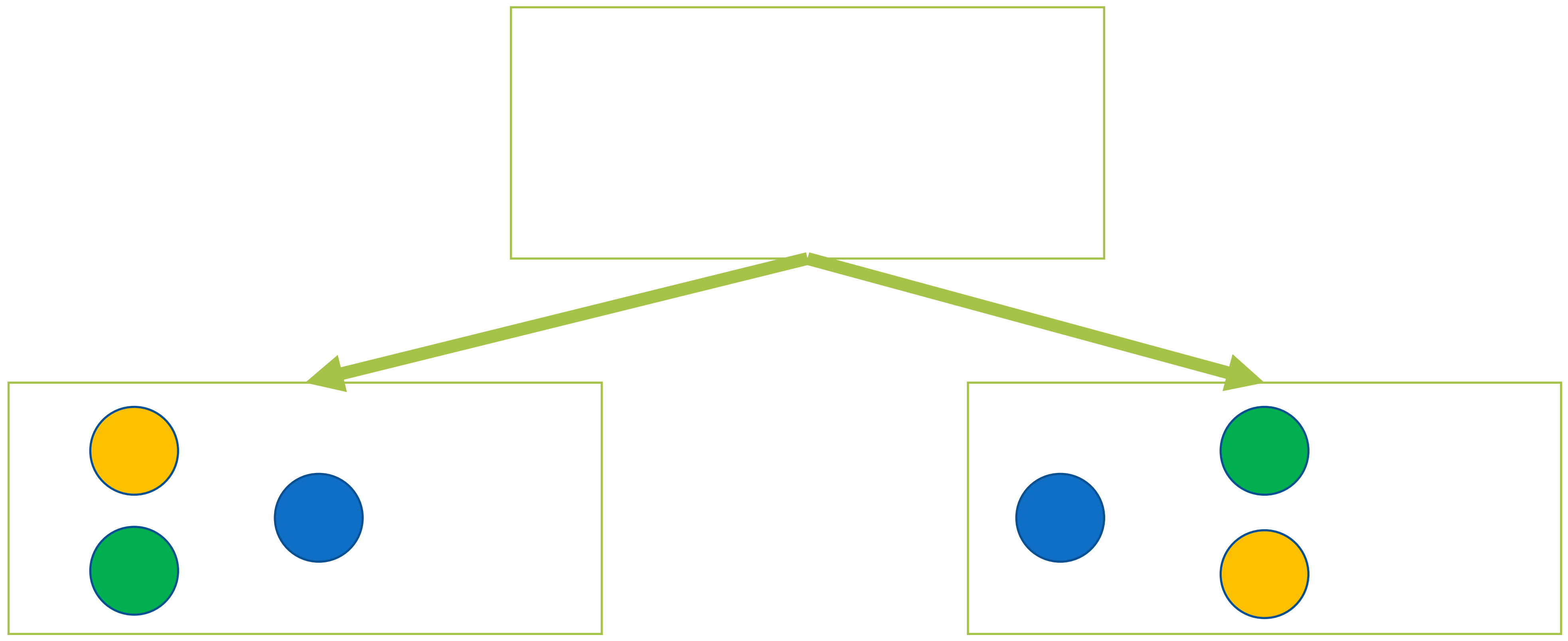


- Как разбить вершину?

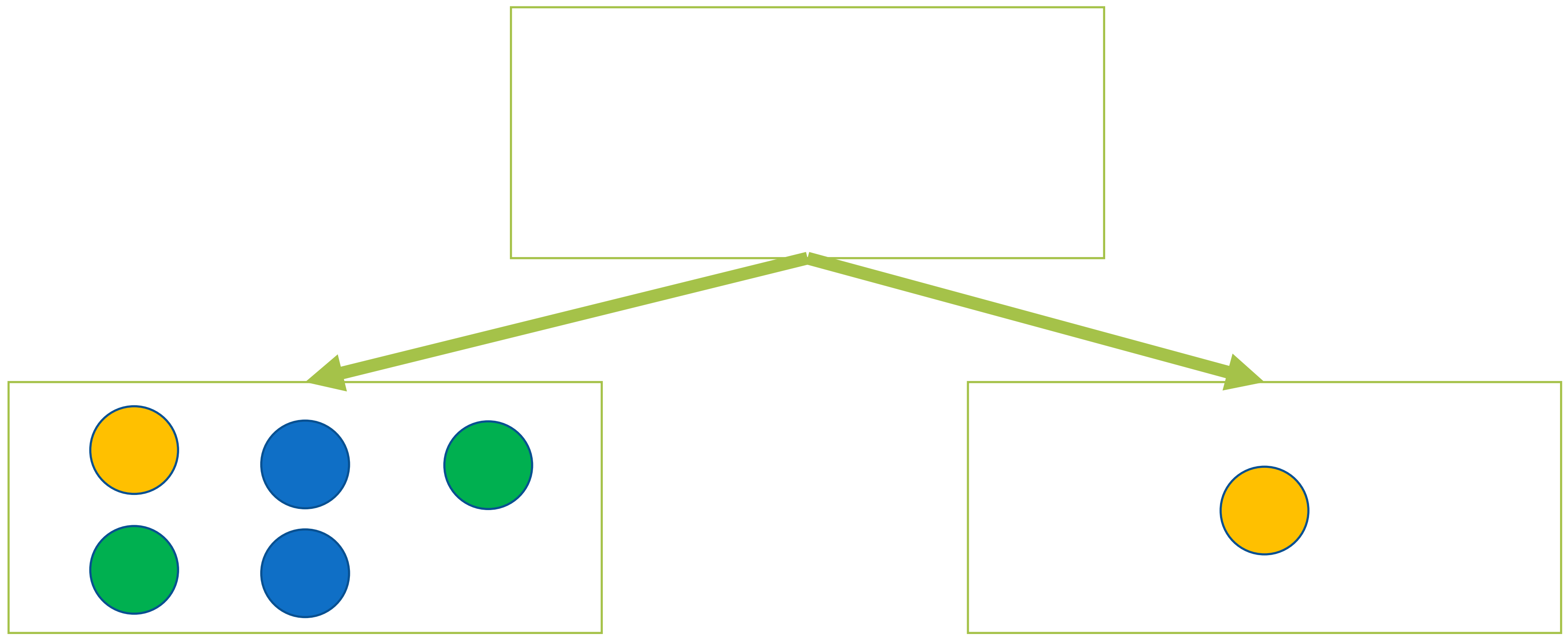
Жадное построение



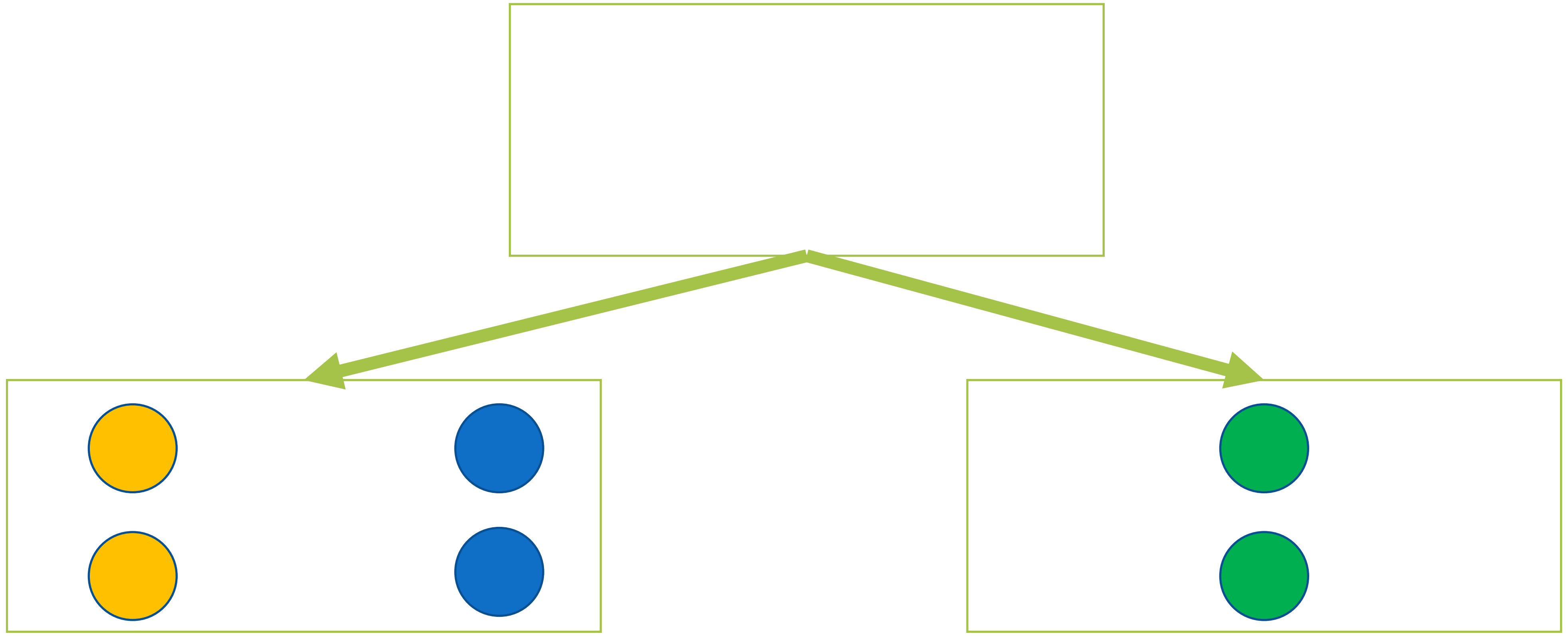
Жадное построение



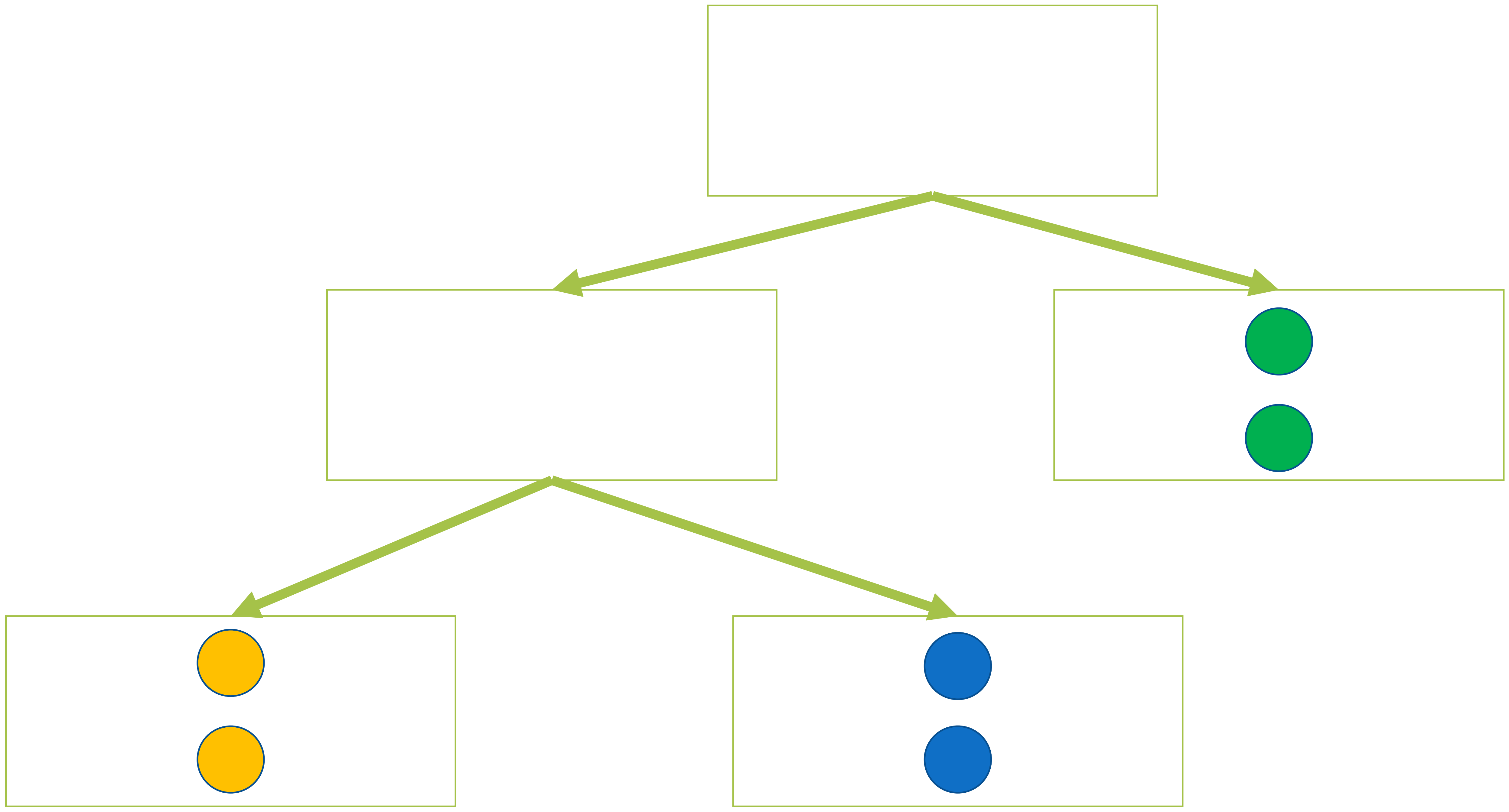
Жадное построение



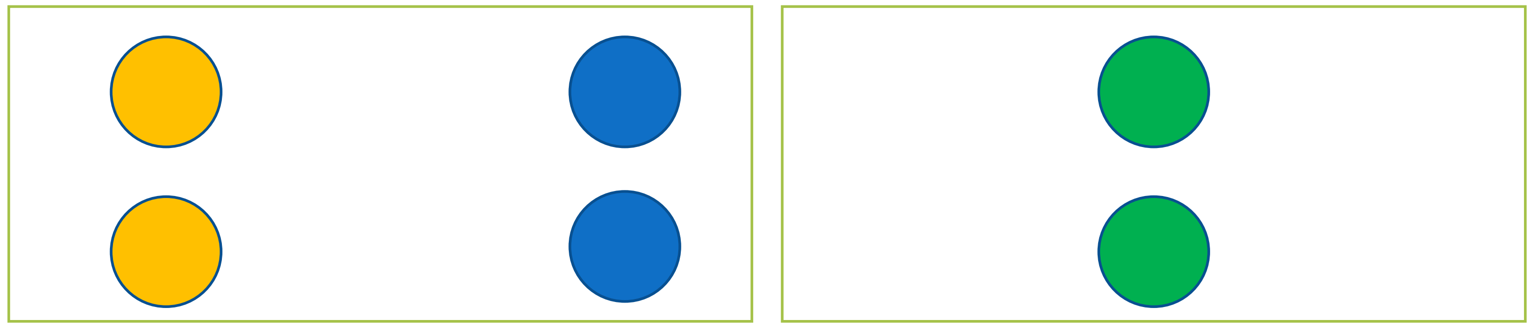
Жадное построение



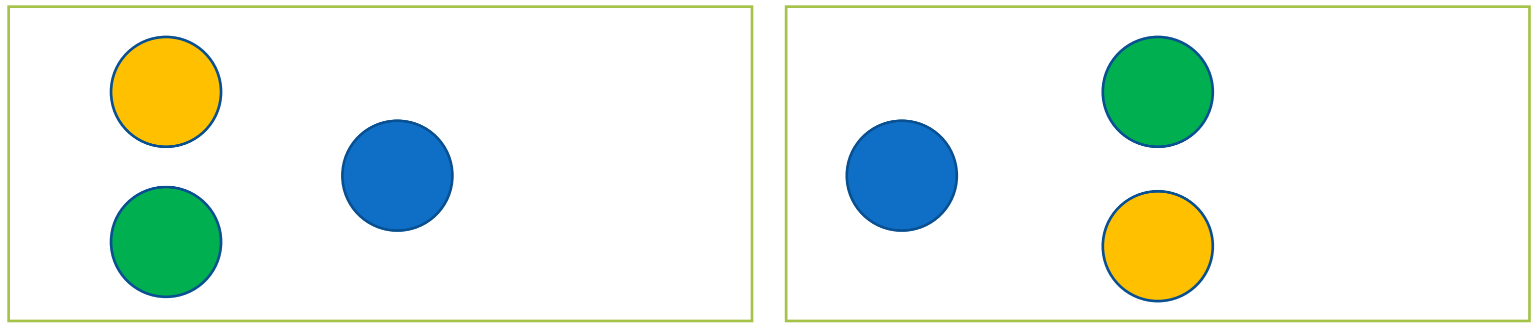
Жадное построение



Как сравнить разбиения?

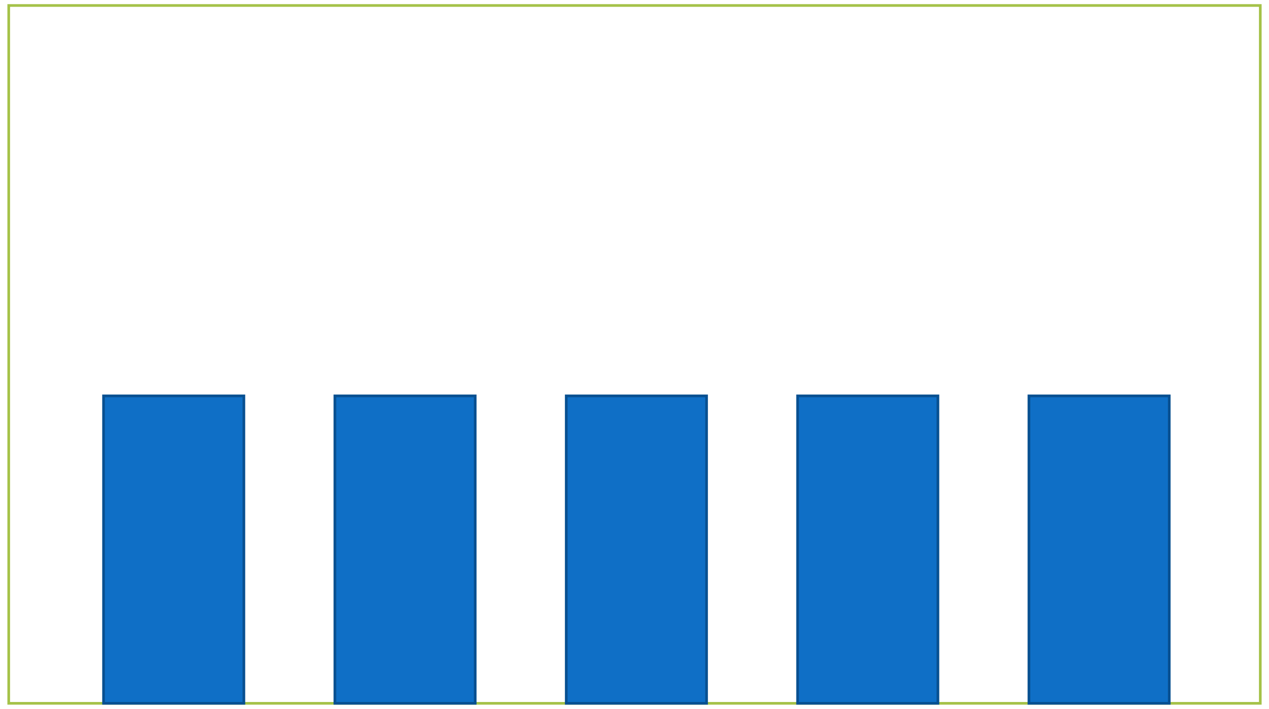


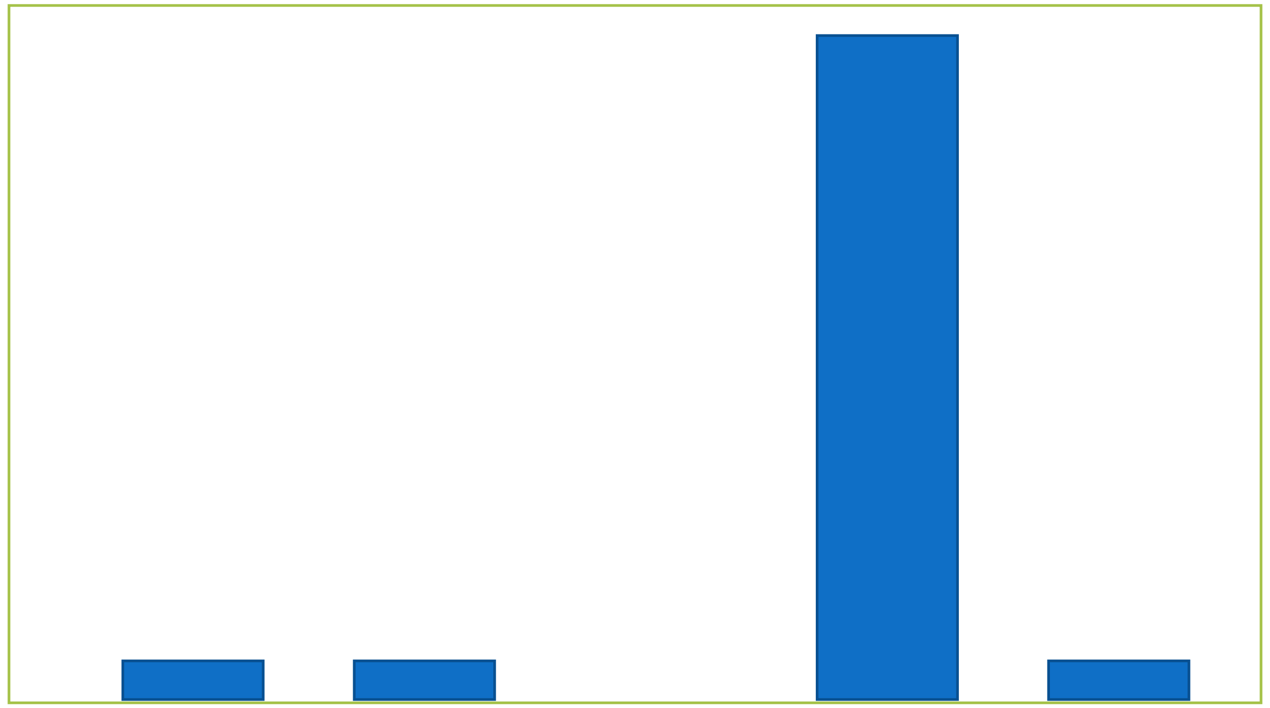
или



Энтропия

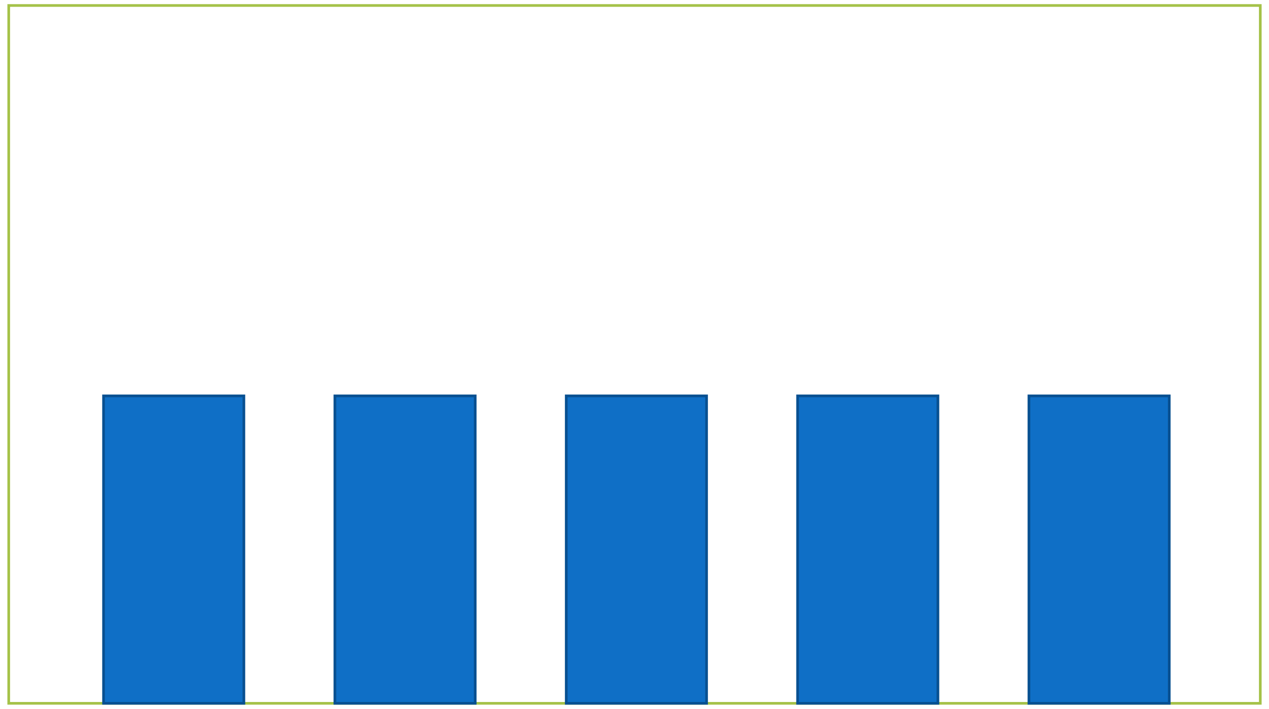
- Мера неопределённости распределения



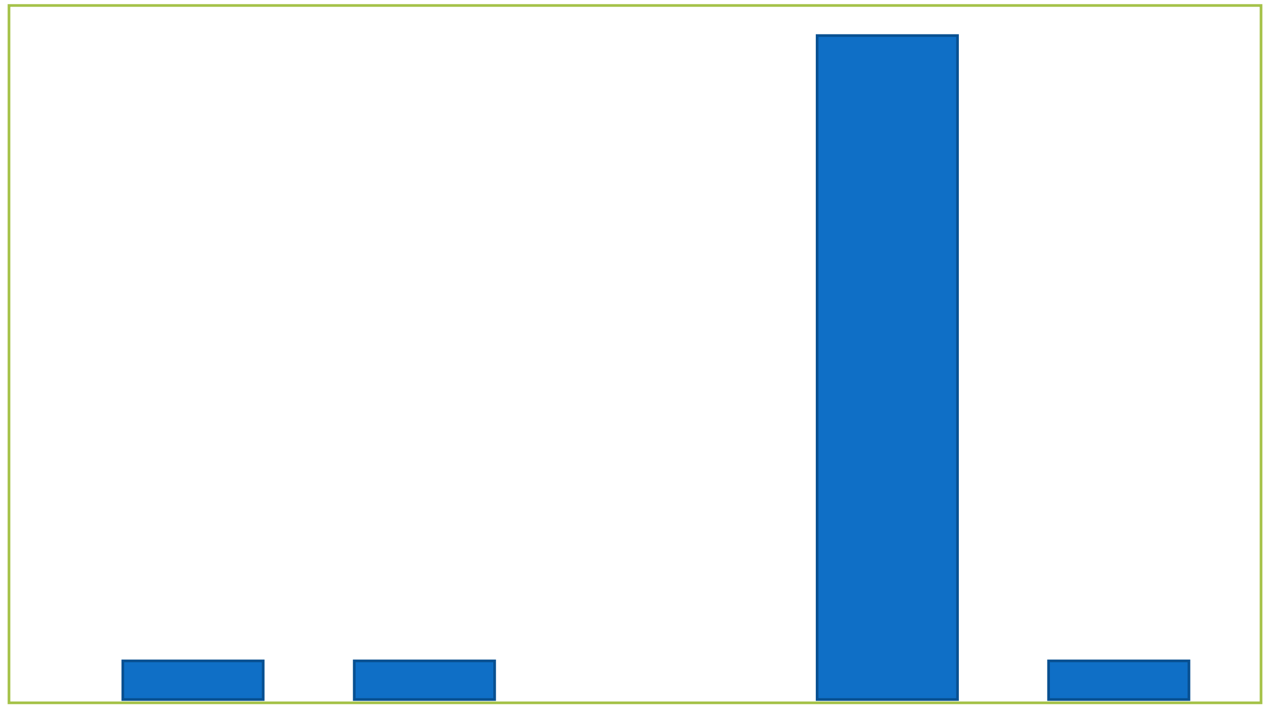


Энтропия

- Мера неопределённости распределения



Высокая энтропия



Низкая энтропия

Энтропия

- Дискретное распределение
- Принимает n значений с вероятностями $p_1, \dots, p_n$
- Энтропия:

$$H(p_1, \dots, p_n) = -\sum_{i=1}^{n} p_i \log p_i$$

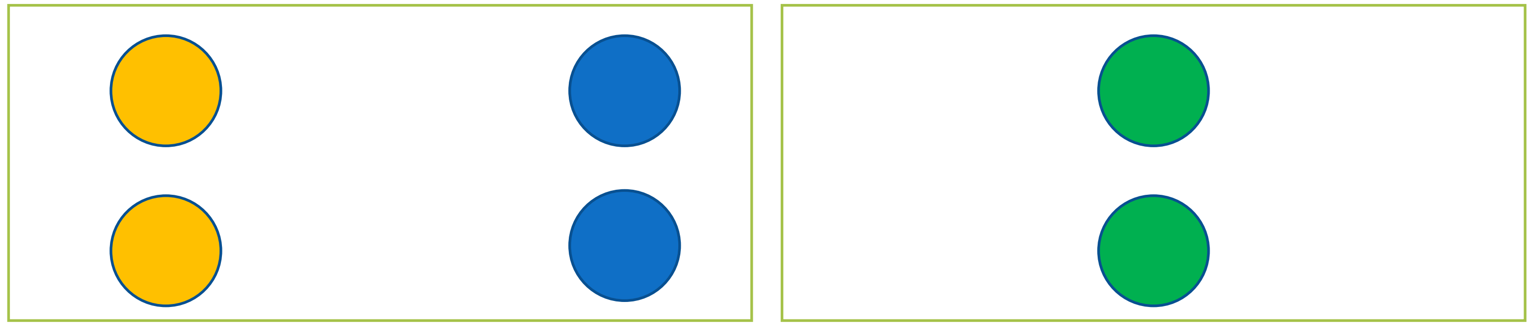
Энтропия

- $(0.2, 0.2, 0.2, 0.2, 0.2)$
- $H = 1.60944 \ldots$

- $(0.9, 0.05, 0.05, 0, 0)$
- $H = 0.394398 \ldots$

- $(0, 0, 0, 1, 0)$
- $H = 0$

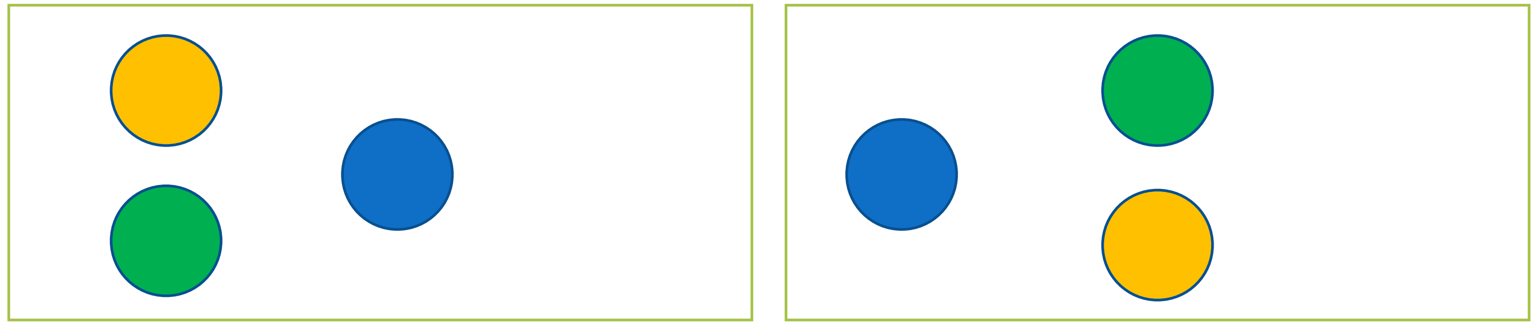
Как сравнить разбиения?



0.693 0

- $(0.5, 0.5, 0)$ и $(0, 0, 1)$
- $H = 0.693 + 0 = 0.693$

1.09 1.09



- $(0.33, 0.33, 0.33)$ и $(0.33, 0.33, 0.33)$
- $H = 1.09 + 1.09 = 2.18$

Энтропия

$$H(p_1, \dots, p_K) = -\sum_{i=1}^{K} p_i \log_2 p_i$$

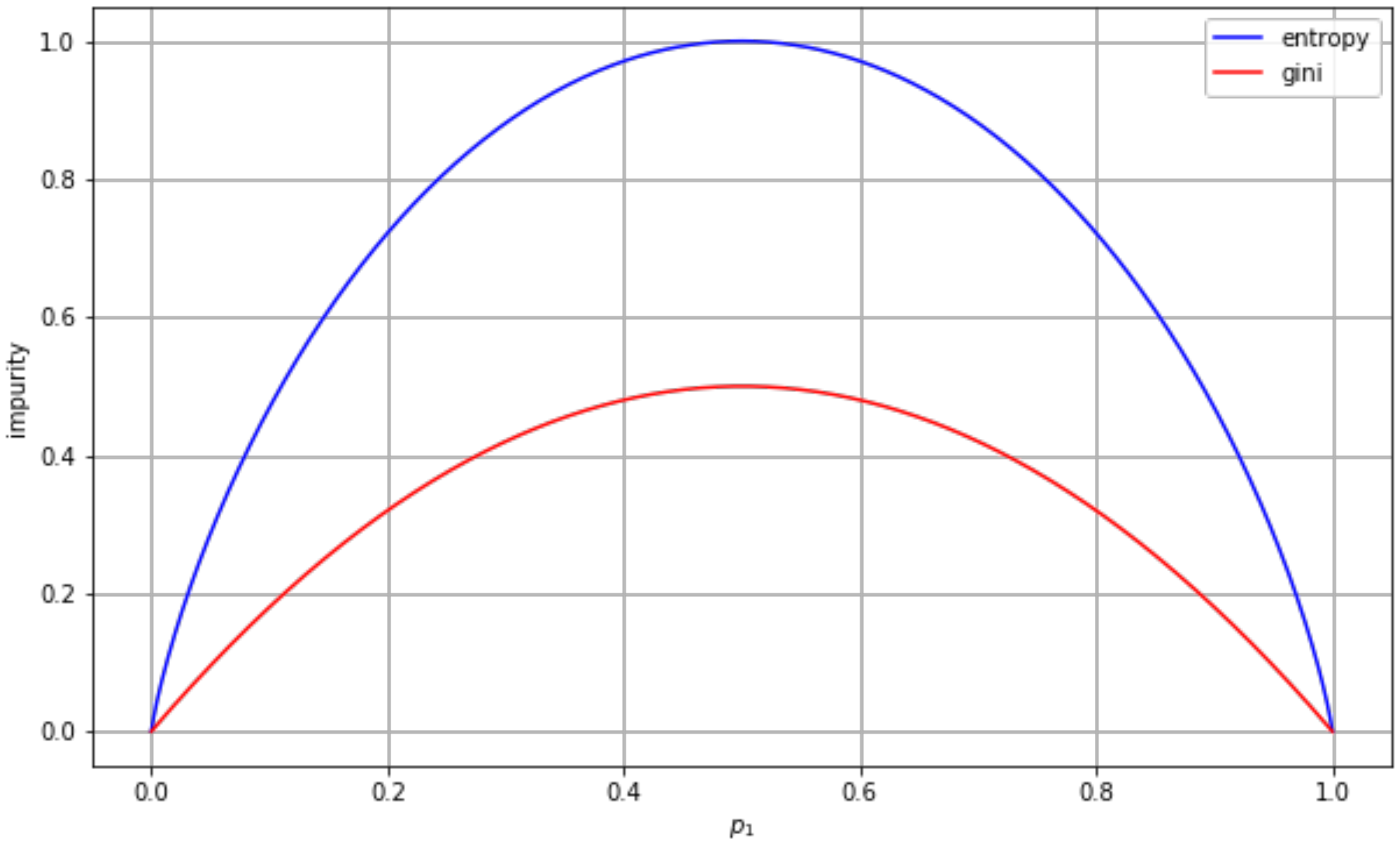
- Характеристика «хаотичности» вершины
- Impurity

Критерий Джини

$$H(p_1, \ldots, p_K) = \sum_{i=1}^{K} p_i \, (1 - p_i)$$

- Вероятность ошибки случайного классификатора, который выдаёт класс k с вероятностью p_k
- Примерно пропорционально количеству пар объектов, относящихся к разным классам

Критерии качества вершины



Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

$$Q(R, j, t) = H(R) - H(R_\ell) - H(R_r) \to \max_{j,t}$$

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

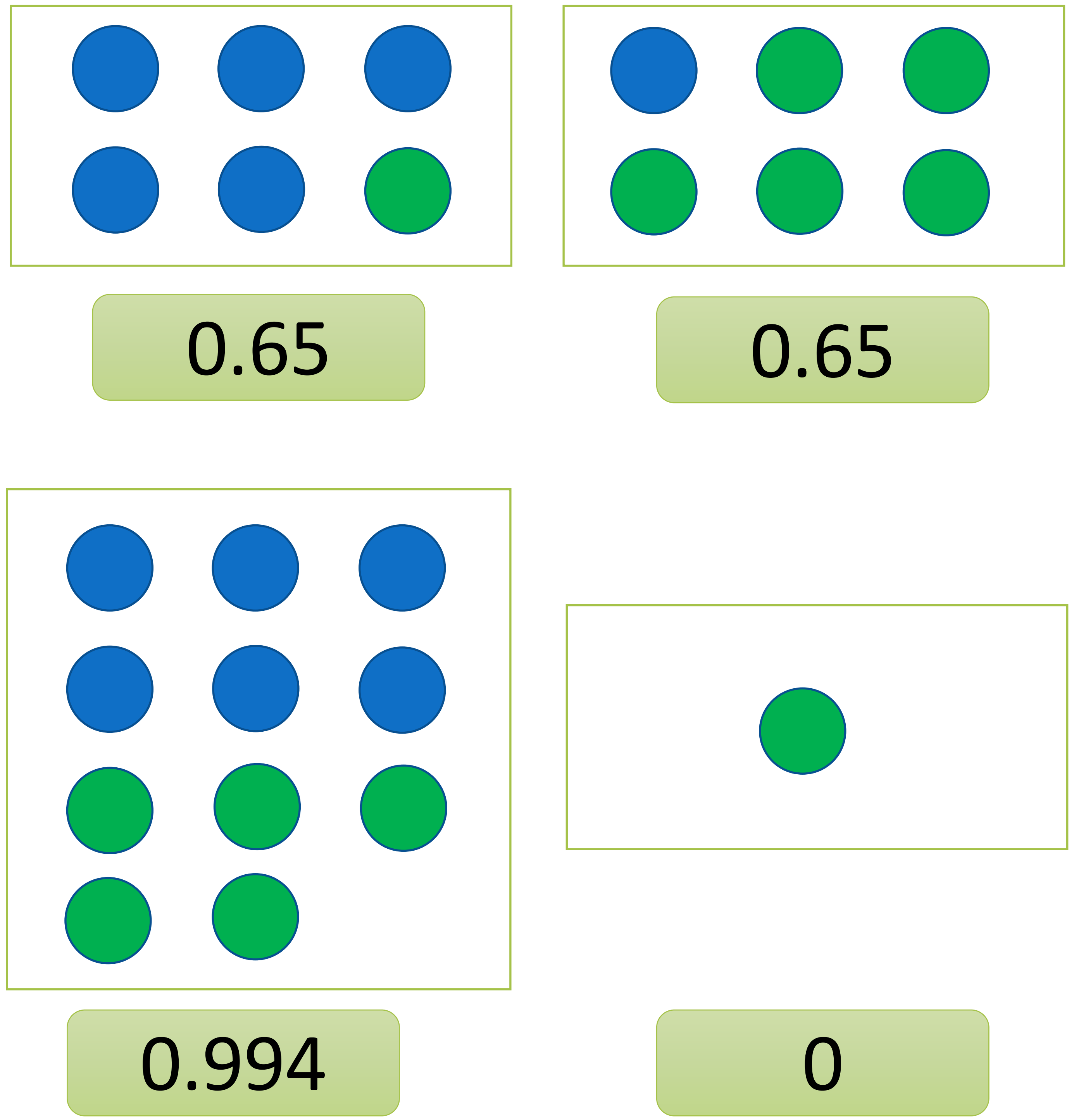
$$Q(R, j, t) = H(R) - H(R_\ell) - H(R_r) \to \max_{j,t}$$

- Или так:

$$Q(R, j, t) = H(R_\ell) + H(R_r) \to \min_{j,t}$$

- (у этих формул есть проблемы!)

Как сравнить разбиения?



- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.65 + 0.65 = 1.3$

- $(6/11, 5/11)$ и $(0, 1)$
- $0.994 + 0 = 0.994$

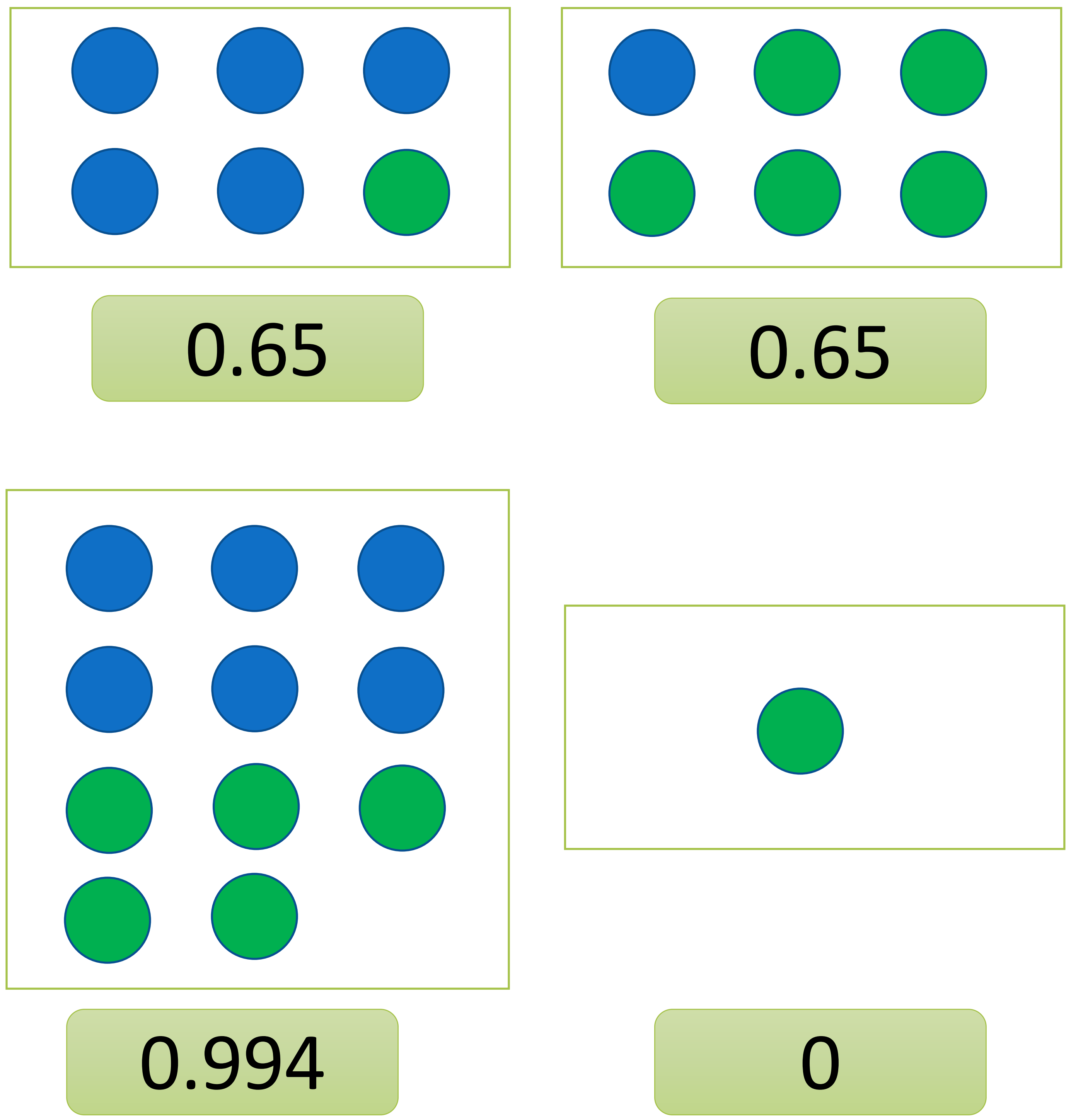
Критерий информативности

$$Q(R, j, t) = H(R) - \frac{|R_\ell|}{|R|} H(R_\ell) - \frac{|R_r|}{|R|} H(R_r) \to \max_{j,t}$$

- Или так:

$$Q(R, j, t) = \frac{|R_\ell|}{|R|} H(R_\ell) + \frac{|R_r|}{|R|} H(R_r) \to \min_{j,t}$$

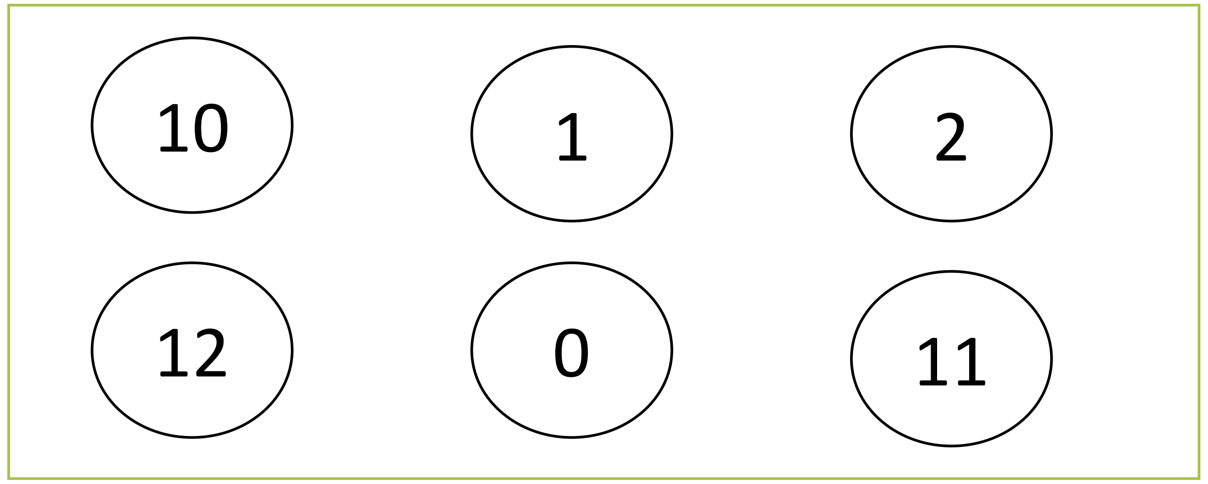
Как сравнить разбиения?



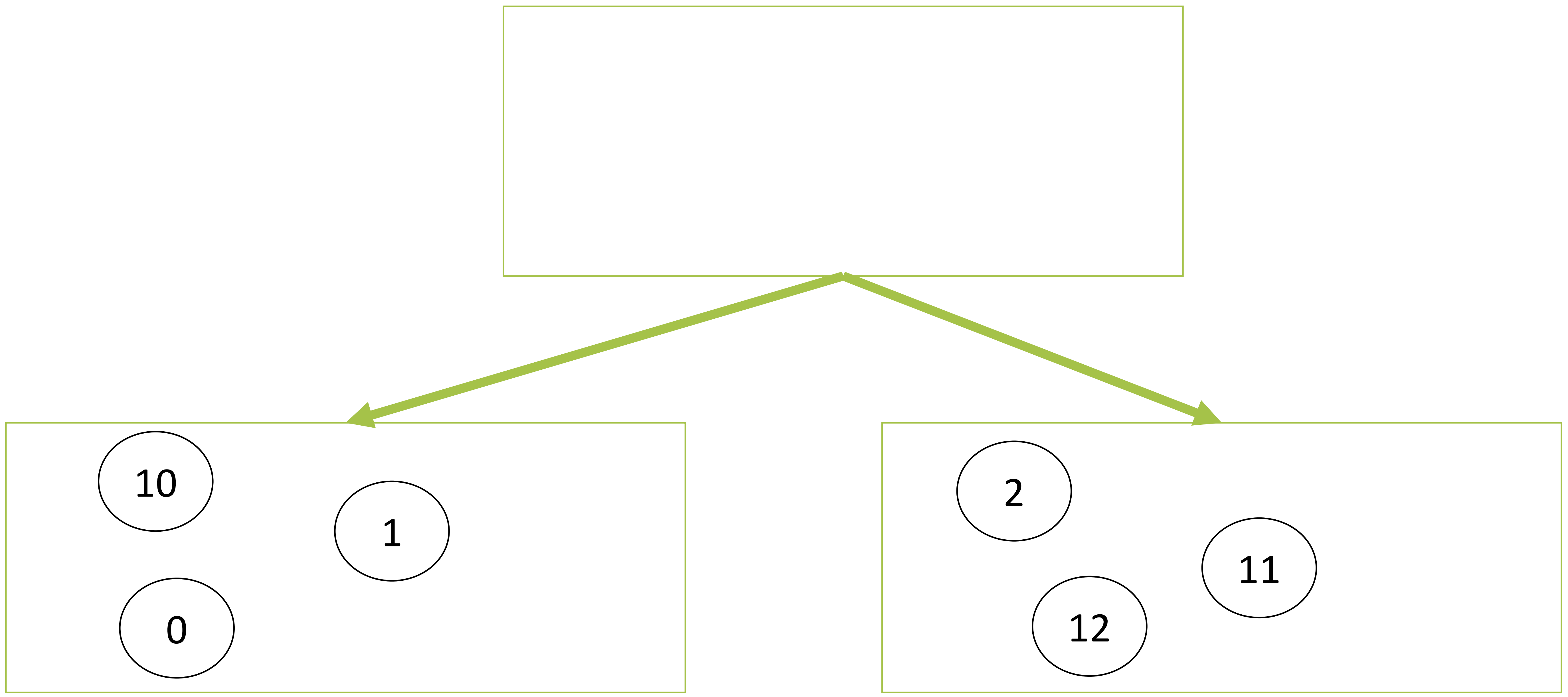
- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.5 * 0.65 + 0.5 * 0.65 = 0.65$

- $(6/11, 5/11)$ и $(0, 1)$
- $\frac{11}{12} * 0.994 + \frac{1}{12} * 0 = 0.911$

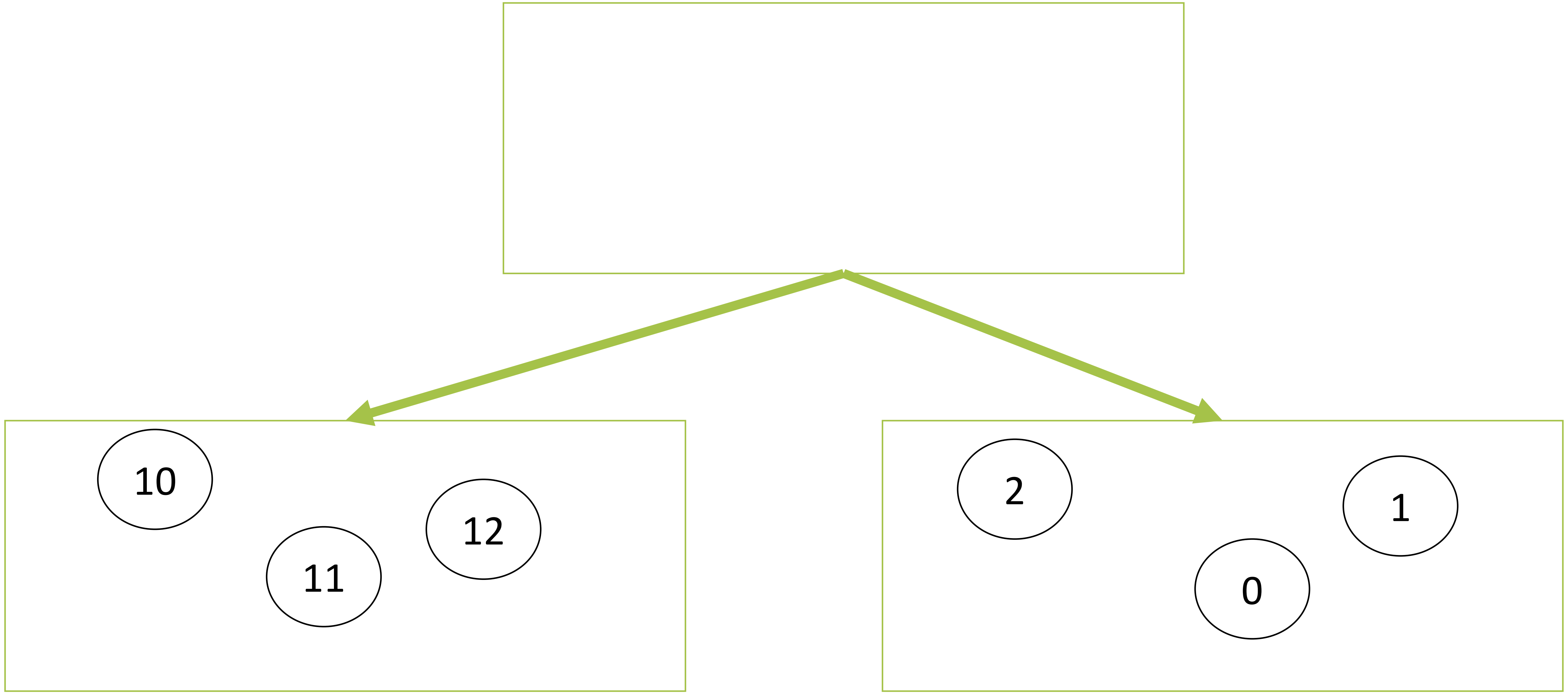
А для регрессии?



А для регрессии?



А для регрессии?



Задача регрессии

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - y_R)^2$$

$$y_R = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i$$

- То есть «хаотичность» вершины можно измерять дисперсией ответов в ней

Жадное построение дерева

Как строить дерево?

- Оптимальный вариант: перебрать все возможные деревья, выбрать самое маленькое среди безошибочных

- Слишком долго

Как строить дерево?

- Мы уже умеем выбрать лучший предикат для разбиения вершины
- Будем строить жадно
- Начнём с корня дерева, будем разбивать последовательно, пока не выполнится некоторый критерий останова

Критерий останова

- Ограничить глубину
- Ограничить количество листьев
- Задать минимальное число объектов в вершине
- Задать минимальное уменьшение хаотичности при разбиении
- И так далее

Жадный алгоритм

1. Поместить в корень всю выборку: $R_1 = X$
2. Запустить построение из корня: $\mathrm{SplitNode}(1, R_1)$

Жадный алгоритм

- $\text{SplitNode}(m, R_m)$

1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg\min_{\mathrm{j,t}} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{\{(x, y) \in R_m | \left[x_j < t\right]\right\}$, $R_r = \left\{\{(x, y) \in R_m | \left[x_j \geq t\right]\right\}$
4. Повторяем для дочерних вершин: $\text{SplitNode}(\ell, R_\ell)$ и $\text{SplitNode}(r, R_r)$

Резюме

- Решающие деревья позволяют строить сложные модели, но есть риск переобучения
- Деревья строятся жадно, на каждом шаге вершина разбивается на две с помощью лучшего из предиктов
- Алгоритм довольно сложный и требует перебора всех предикатов на каждом шаге