

# Основы машинного обучения

## Лекция 11

Линейная классификация. Многоклассовая классификация.

Евгений Соколов

[esokolov@hse.ru](mailto:esokolov@hse.ru)

НИУ ВШЭ, 2025

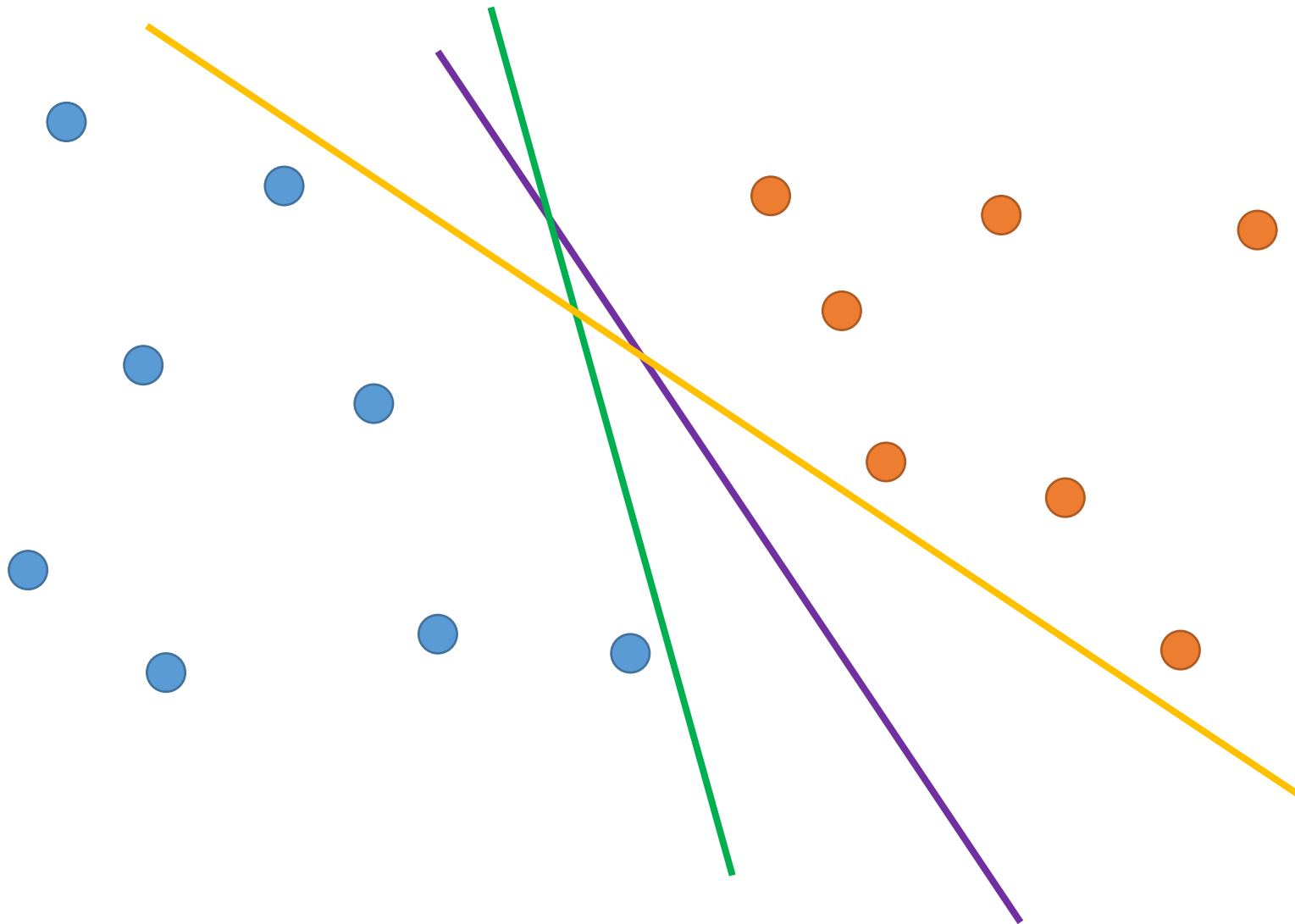
# Метод опорных векторов

# Hinge loss

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

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \max(0, 1 - y_i \langle w, x_i \rangle) \rightarrow \min_w$$

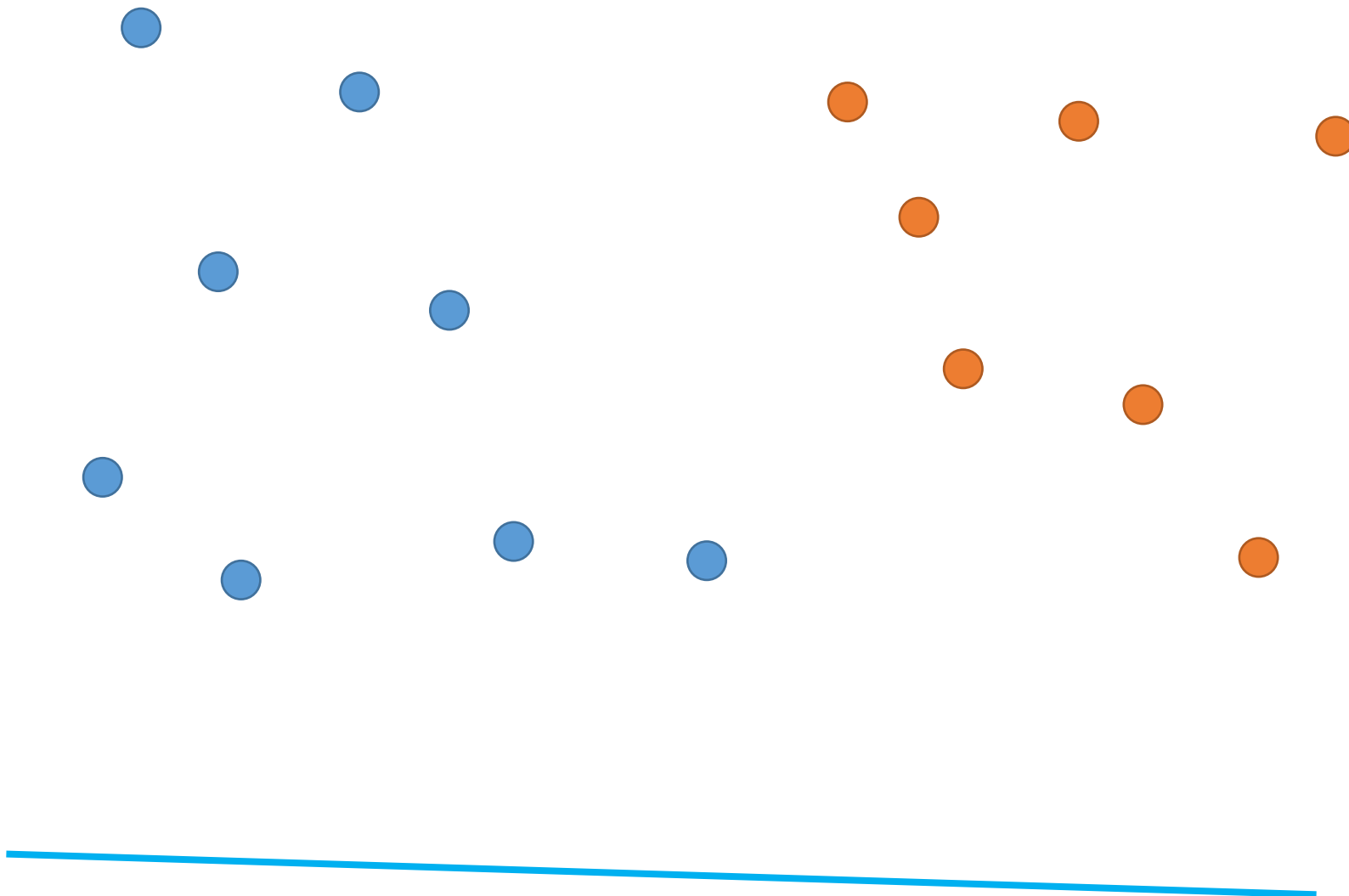
Какой классификатор лучше?



# Отступ классификатора

- Будем максимизировать отступ классификатора — расстояние от гиперплоскости до ближайшего объекта

# Отступ классификатора



# Отступ классификатора

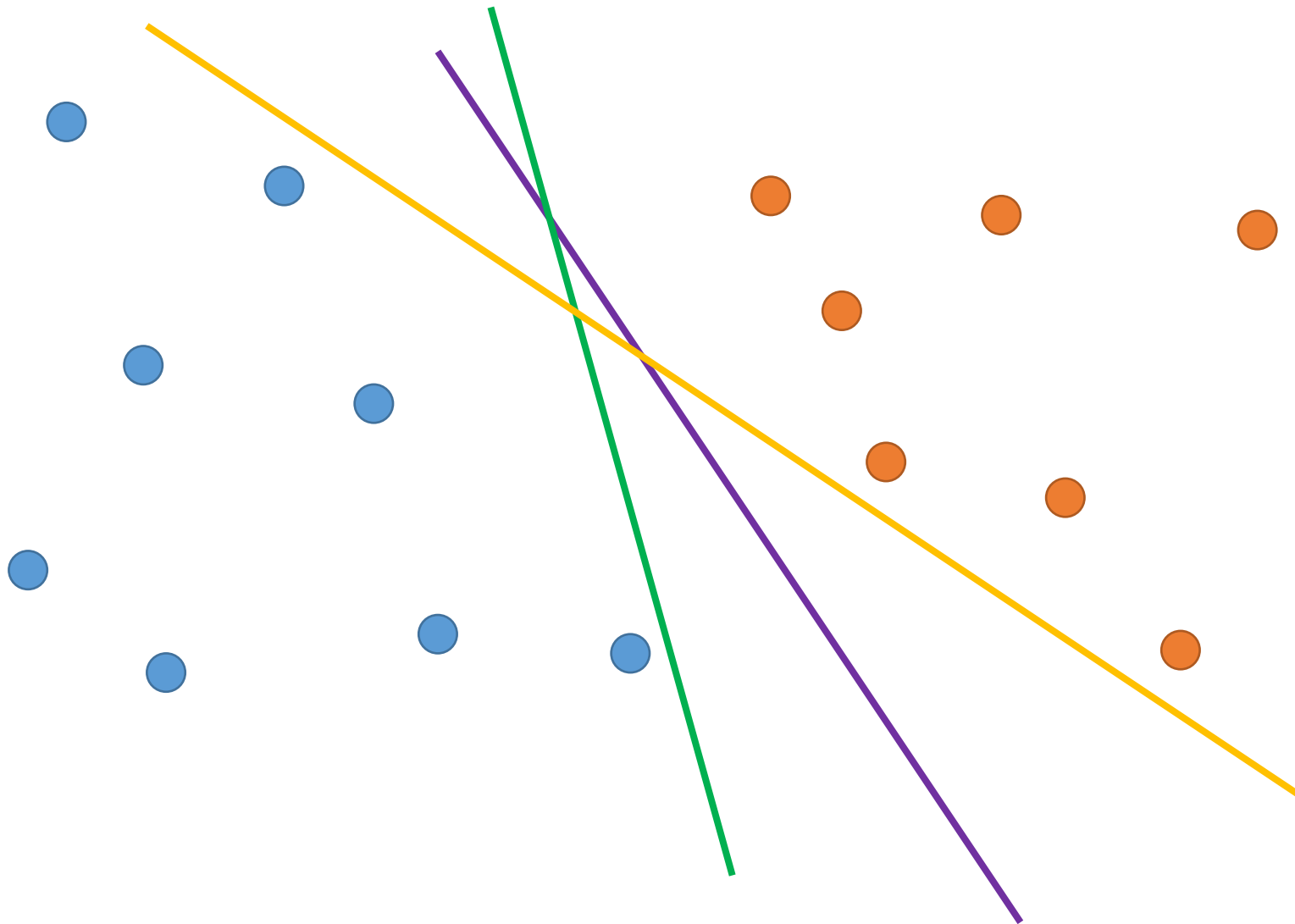
- Будем максимизировать отступ классификатора — расстояние от гиперплоскости до ближайшего объекта
- При этом будет стараться сделать поменьше ошибок
- По сути, делаем как можно меньше предположений о модели, и верим, что это понизит вероятность переобучения

# Простой случай

- Будем считать, что выборка линейно разделима
- Существует линейный классификатор, не допускающий ни одной ошибки



# Линейно разделимый случай



# Линейно разделимый случай

- **Требование 1:**  $y_i(\langle w, x_i \rangle + w_0) > 0$  для всех  $i = 1, \dots, \ell$
- **Требование 2:** максимальный отступ классификатора

# Отступ классификатора

- Расстояние от точки до гиперплоскости  $\langle w, x \rangle + w_0 = 0$ :

$$\frac{|\langle w, x \rangle + w_0|}{\|w\|}$$

- Отступ классификатора:

$$\min_{i=1, \dots, \ell} \frac{|\langle w, x_i \rangle + w_0|}{\|w\|}$$

# Небольшое предположение

- Линейный классификатор:

$$a(x) = \text{sign} (\langle w, x_i \rangle + w_0)$$

- Если мы поделим  $w$  и  $w_0$  на число  $a > 0$ , то выходы классификатора никак не поменяются:

$$a(x) = \text{sign} \left( \frac{\langle w, x_i \rangle + w_0}{a} \right) = \text{sign} (\langle w, x_i \rangle + w_0)$$

# Небольшое предположение

- Поделим  $w$  и  $w_0$  на  $\min_{i=1,\dots,\ell} |\langle w, x_i \rangle + w_0| > 0$ , после этого будет выполнено

$$\min_{i=1,\dots,\ell} |\langle w, x_i \rangle + w_0| = 1$$

# Отступ классификатора

- Расстояние от точки до гиперплоскости  $\langle w, x \rangle + w_0 = 0$ :

$$\frac{|\langle w, x \rangle + w_0|}{\|w\|}$$

- Отступ классификатора:

$$\min_{i=1, \dots, \ell} \frac{|\langle w, x_i \rangle + w_0|}{\|w\|} = \frac{\min_{i=1, \dots, \ell} |\langle w, x_i \rangle + w_0|}{\|w\|} = \frac{1}{\|w\|}$$

# Линейно разделимый случай

- **Требование 1:**  $y_i(\langle w, x_i \rangle + w_0) > 0$  для всех  $i = 1, \dots, \ell$
- **Требование 2:** максимальный отступ классификатора

$$\frac{1}{\|w\|} \rightarrow \max_w$$

# Линейно разделимый случай

- **Требование 1:**  $y_i(\langle w, x_i \rangle + w_0) > 0$  для всех  $i = 1, \dots, \ell$
- **Требование 2:** максимальный отступ классификатора

$$\frac{1}{\|w\|} \rightarrow \max_w$$

- При условии, что  $\min_{i=1, \dots, \ell} |\langle w, x_i \rangle + w_0| = 1$



# Линейно разделимый случай

- **Требование 1:**  $y_i(\langle w, x_i \rangle + w_0) > 0$  для всех  $i = 1, \dots, \ell$
- **Требование 2:** максимальный отступ классификатора

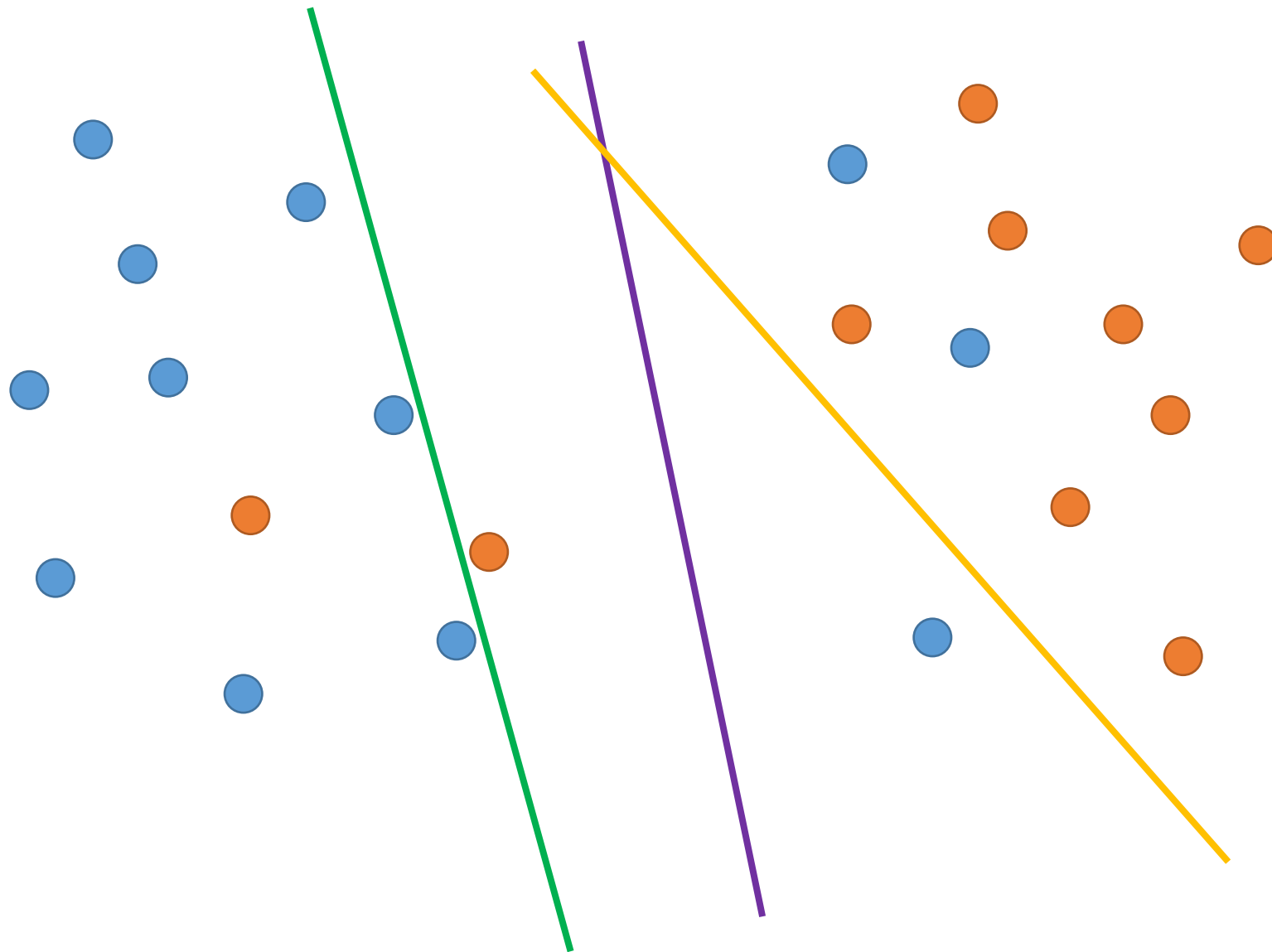
$$\frac{1}{\|w\|} \rightarrow \max_w$$

- При условии, что  $|\langle w, x_i \rangle + w_0| \geq 1$
- И мы минимизируем  $\|w\|$  — тогда где-то модуль отступа будет равен 1

# Метод опорных векторов (SVM)

$$\begin{cases} \|w\|^2 \rightarrow \min_{w, w_0} \\ y_i(\langle w, x_i \rangle + w_0) \geq 1 \end{cases}$$

# Линейно неразделимый случай



# Линейно неразделимый случай

- Любой линейный классификатор допускает хотя бы одну ошибку

$$\begin{cases} \|w\|^2 \rightarrow \min_{w, w_0} \\ y_i(\langle w, x_i \rangle + w_0) \geq 1 \end{cases}$$

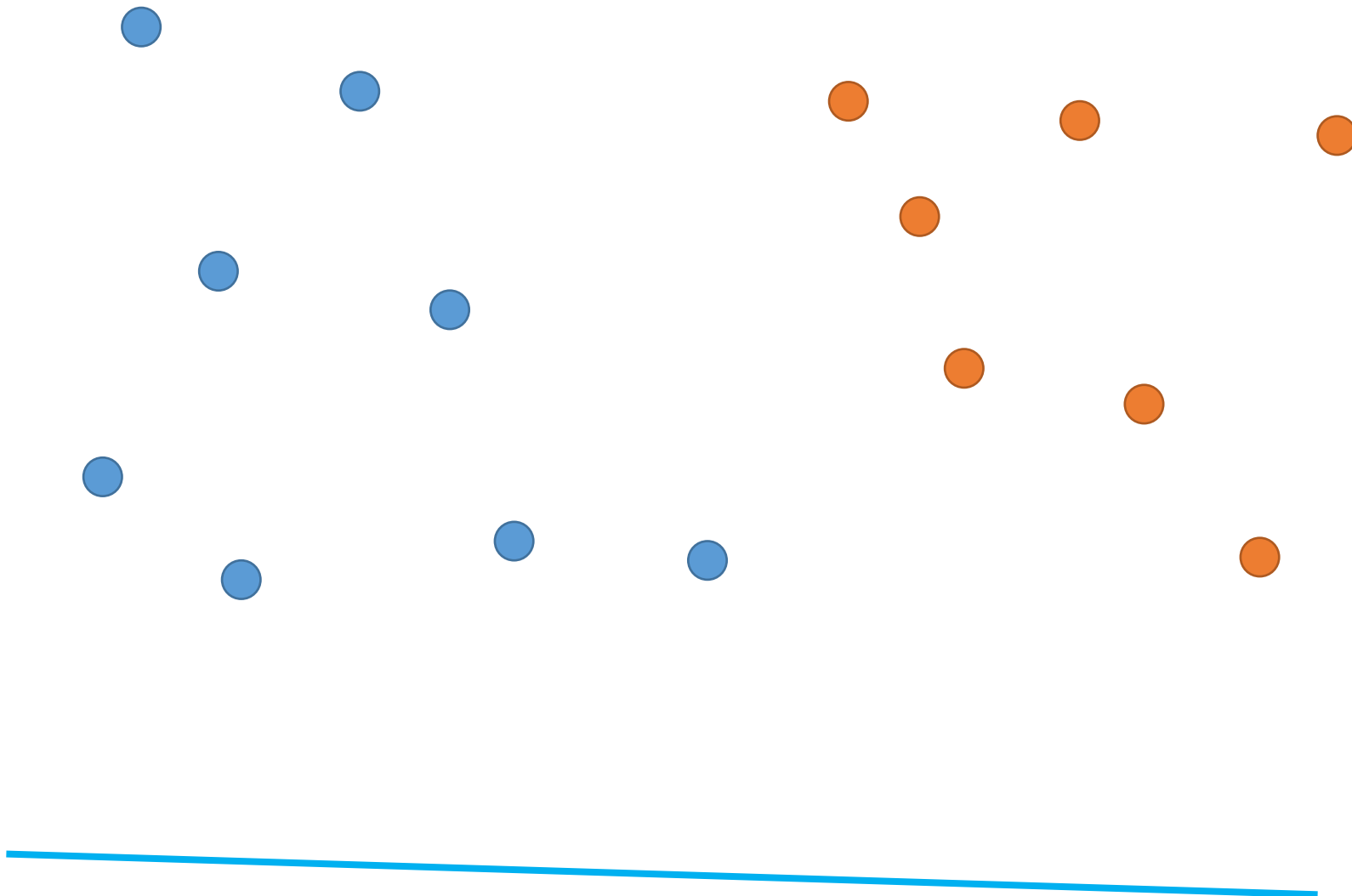
# Линейно неразделимый случай

$$\begin{cases} \|w\|^2 \rightarrow \min_{w, w_0} \\ y_i(\langle w, x_i \rangle + w_0) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

# Линейно неразделимый случай

$$\begin{cases} \|w\|^2 \rightarrow \min_{w, w_0} \\ y_i(\langle w, x_i \rangle + w_0) \geq 1 - 10^{1000} \end{cases}$$

# Отступ классификатора

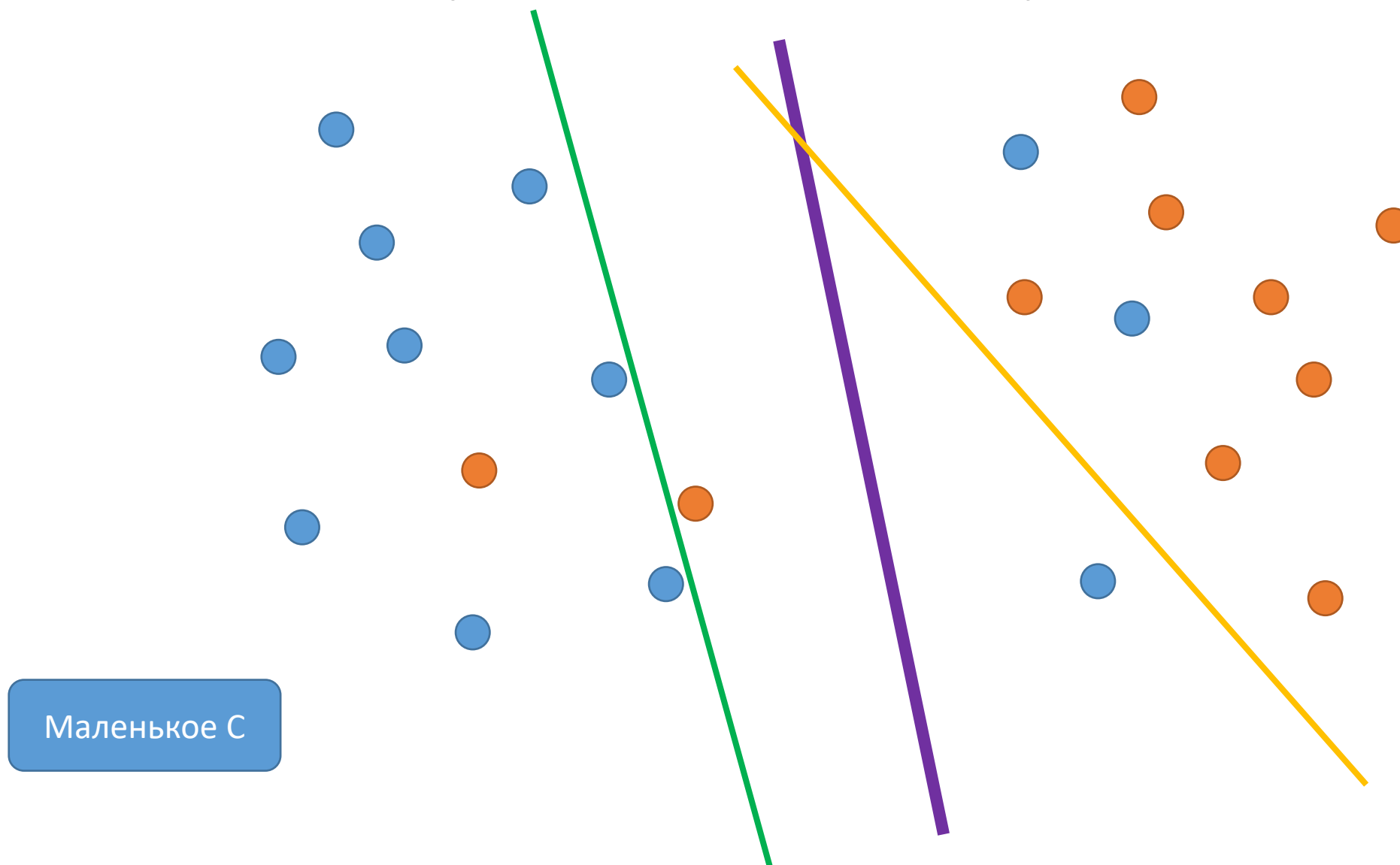


# Метод опорных векторов

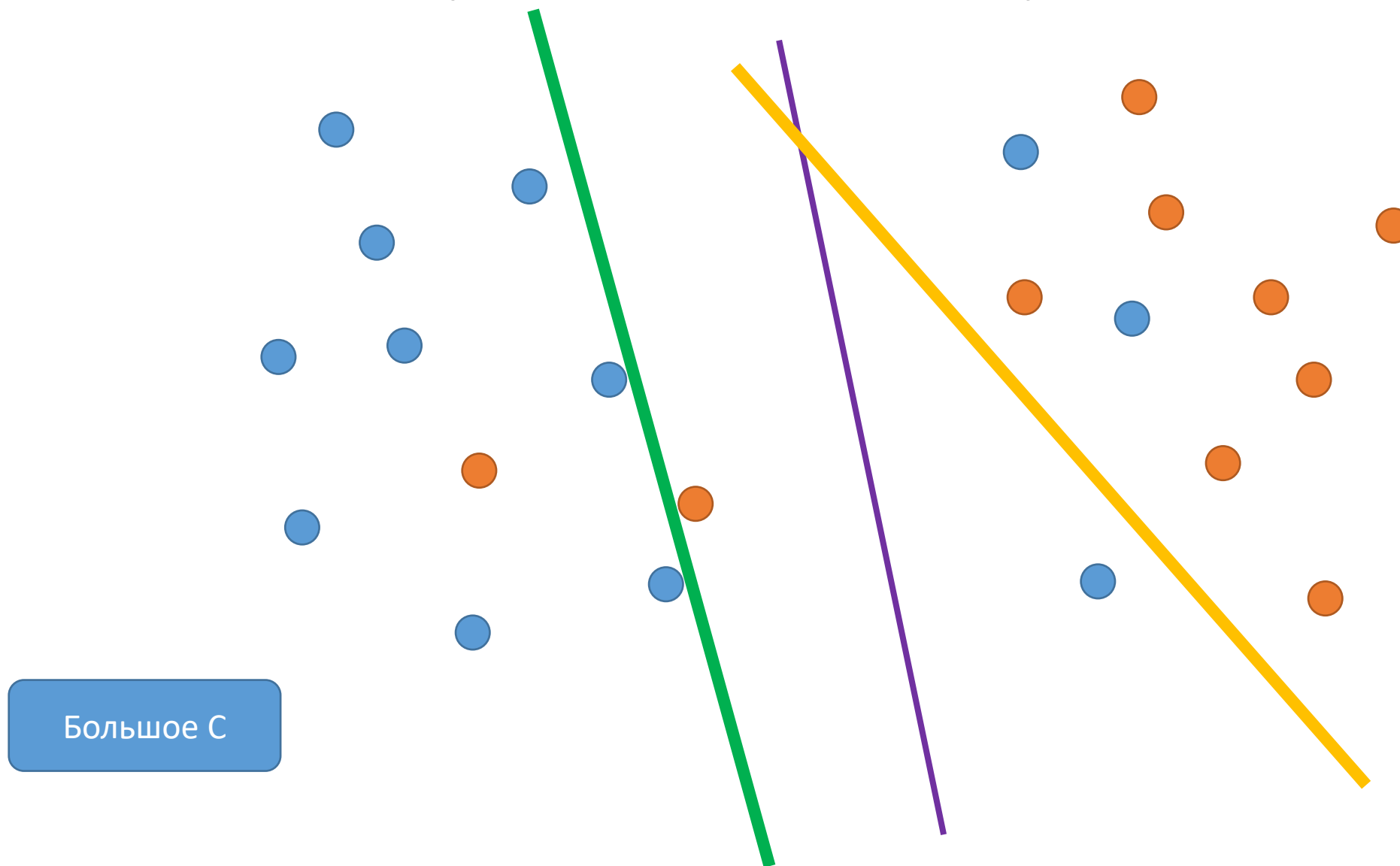
$$\left\{ \begin{array}{l} \|w\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w, w_0, \xi_i} \\ y_i(\langle w, x_i \rangle + w_0) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \right.$$



# Линейно неразделимый случай



# Линейно неразделимый случай



# Метод опорных векторов

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

- Объединим ограничения:

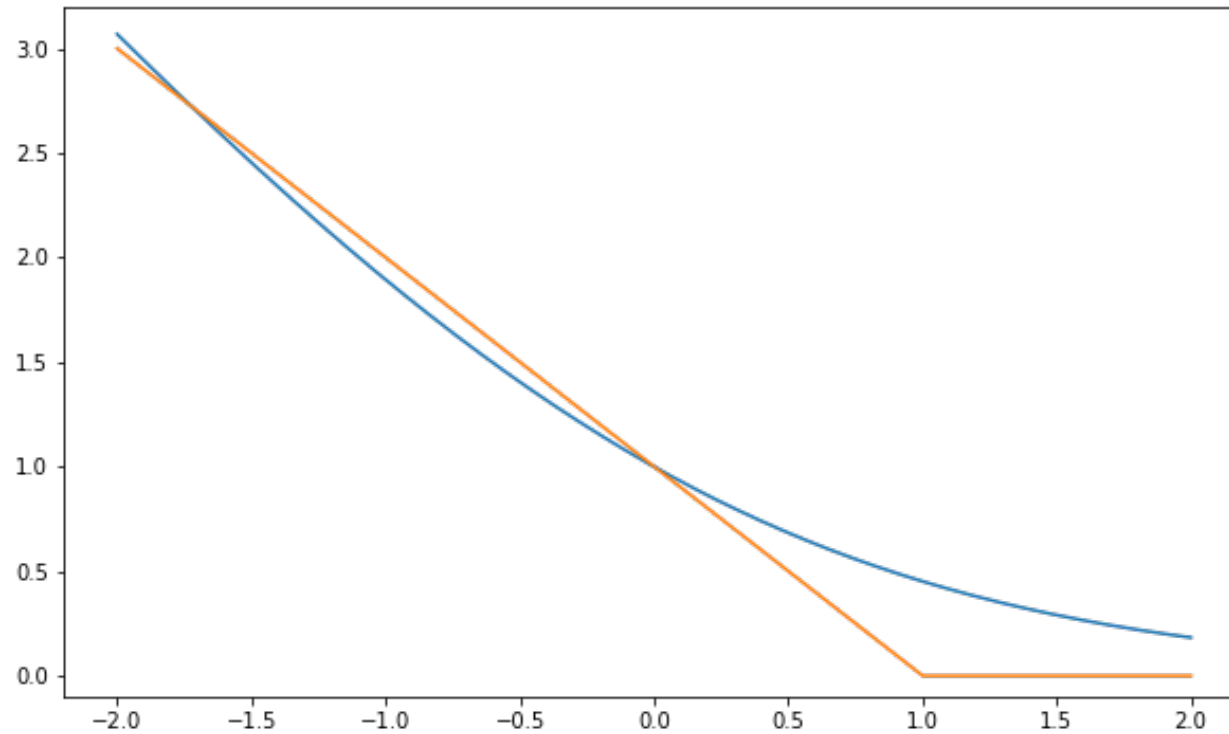
$$\xi_i \geq \max(0, 1 - y_i(\langle w, x_i \rangle + w_0))$$

# Метод опорных векторов

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

- Функция потерь (hinge loss) + регуляризация

# Сравнение логистической регрессии и SVM

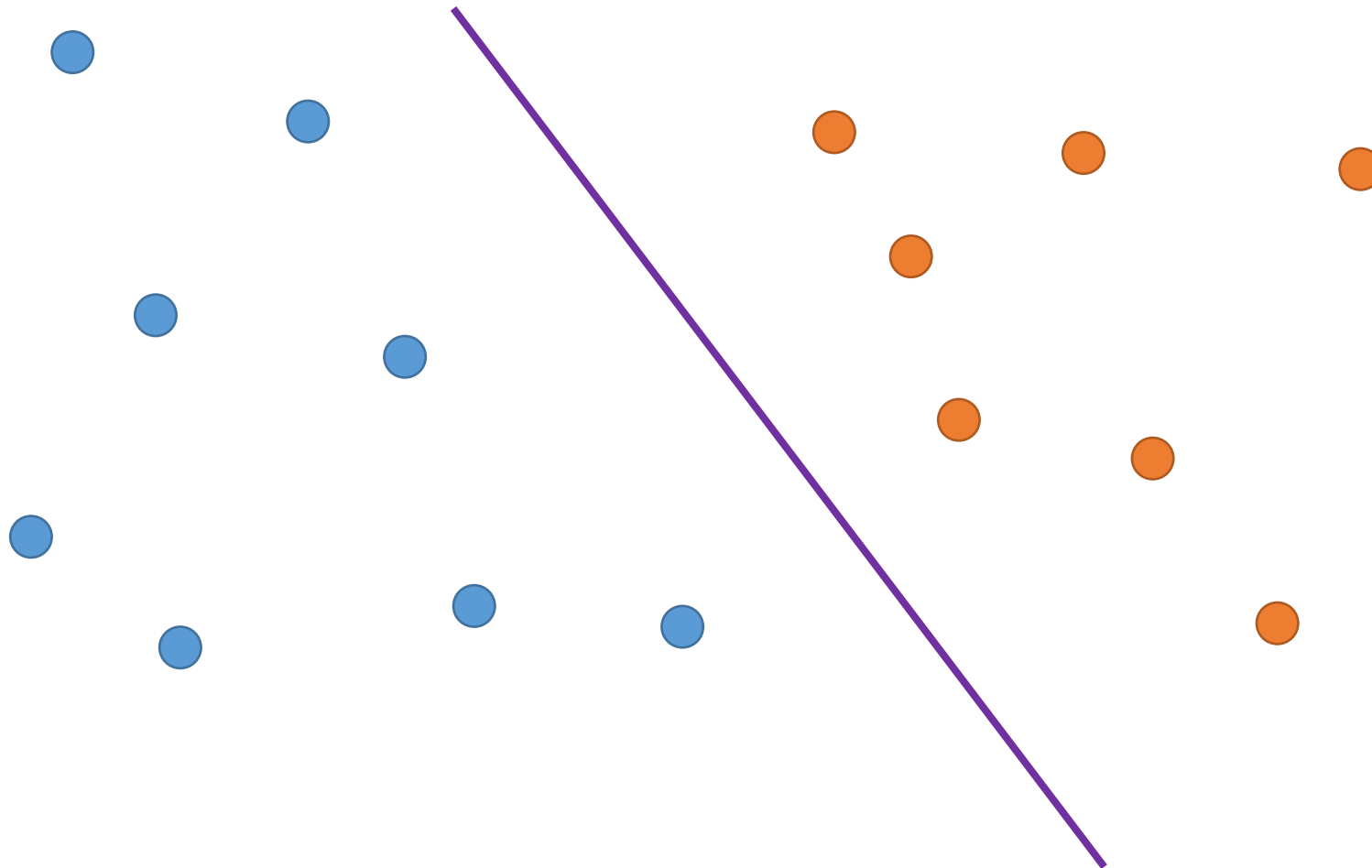


# Резюме

- Логистическая регрессия — обучение модели так, что на объектах с близкими прогнозами эти прогнозы стремятся к доле положительных объектов
- Метод опорных векторов основан на идее максимизации отступа классификатора

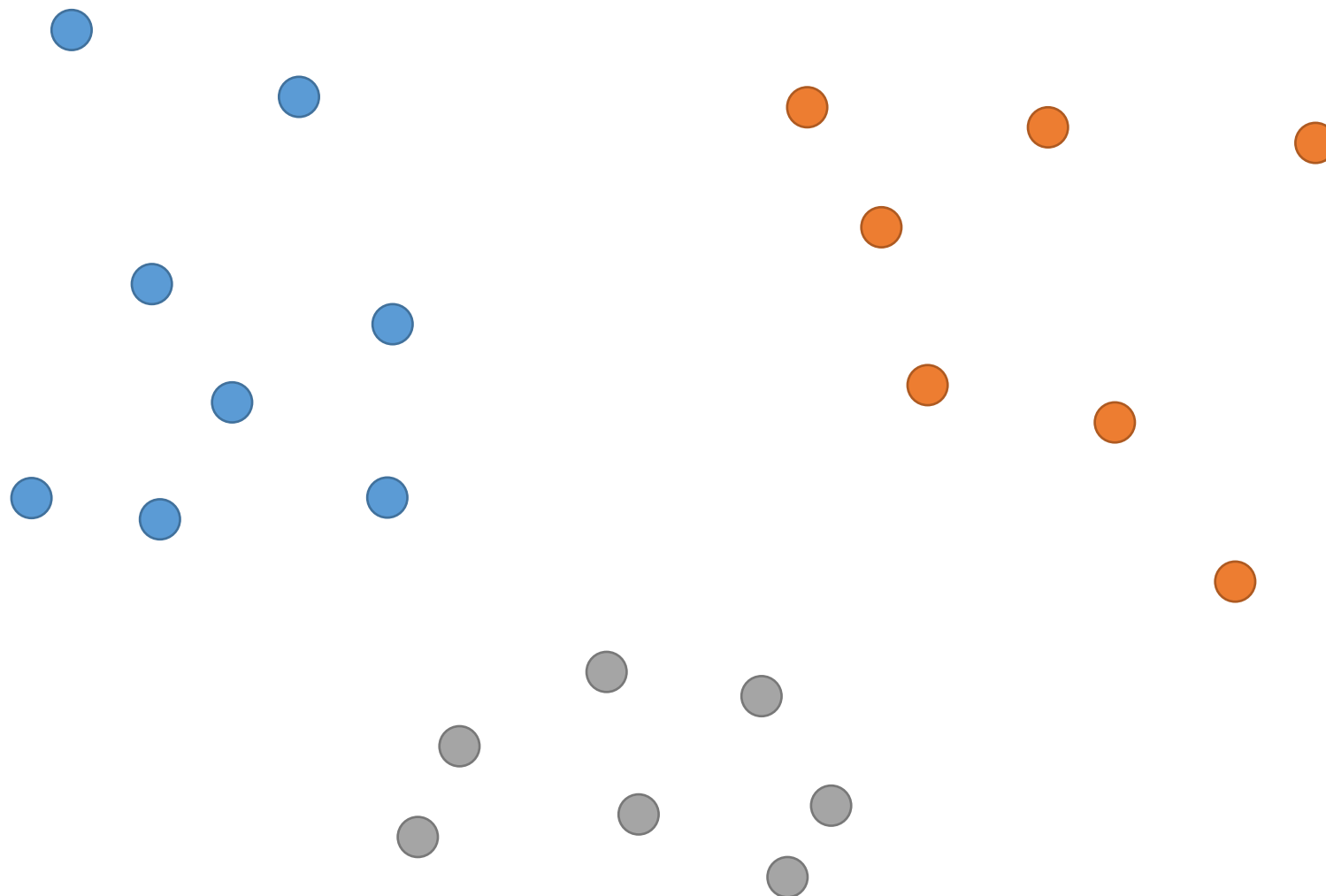
Многоклассовая классификация

# Бинарная классификация

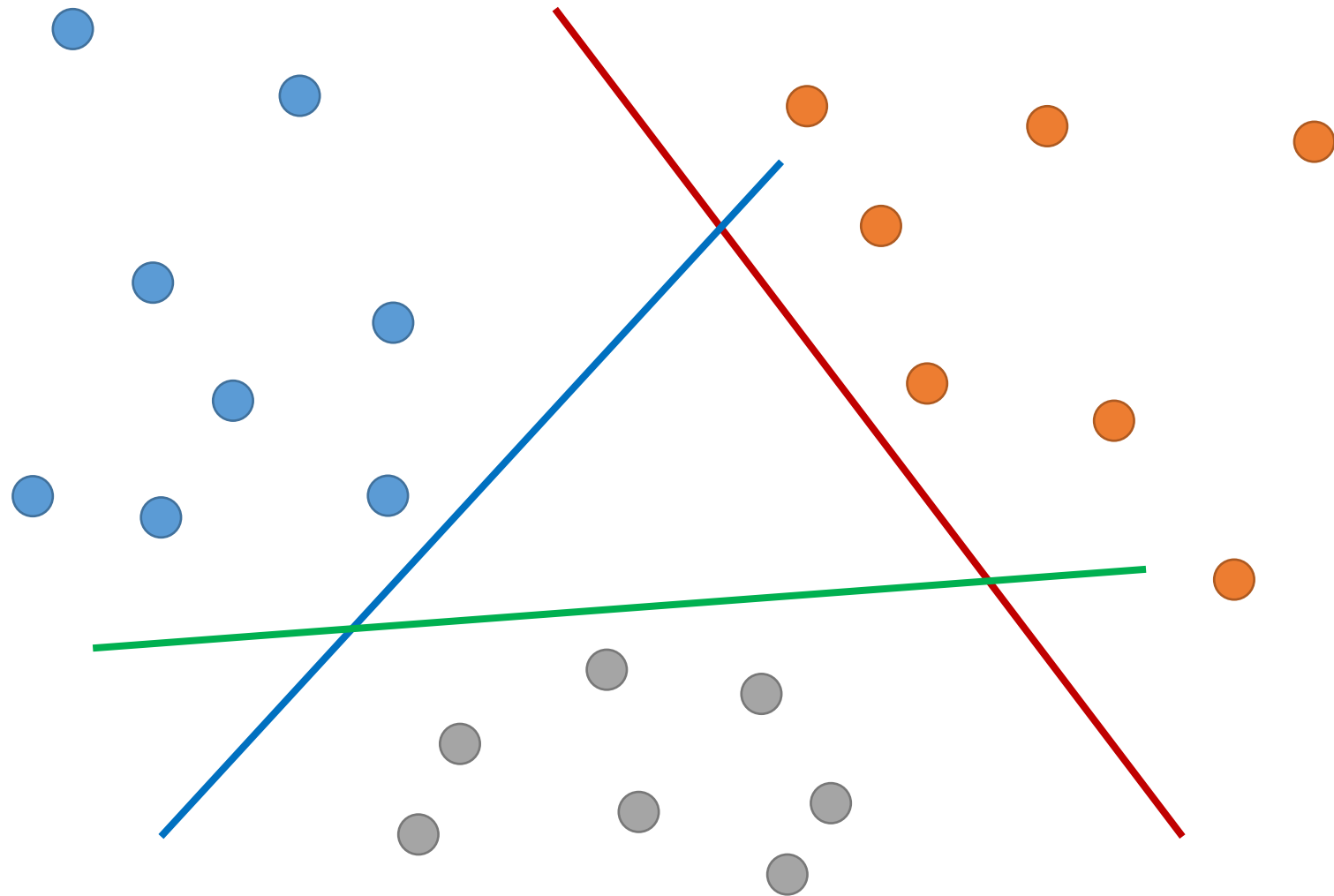




# Многоклассовая классификация



# Многоклассовая классификация



# One-vs-all

- $K$  классов:  $\mathbb{Y} = \{1, \dots, K\}$
- $X_k = (x_i, [y_i = k])_{i=1}^{\ell}$
- Обучаем  $a_k(x)$  на  $X_k$ ,  $k = 1, \dots, K$
- $a_k(x)$  должен выдавать оценки принадлежности классу (например,  $\langle w, x \rangle$  или  $\sigma(\langle w, x \rangle)$ )
- Итоговая модель:

$$a(x) = \arg \max_{k=1, \dots, K} a_k(x)$$

# One-vs-all

- Модель  $a_k(x)$  при обучении не знает, что её выходы будут сравнивать с выходами других моделей
- Нужно обучать  $K$  моделей

# All-vs-all

- $X_{km} = \{(x_i, y_i) \in X \mid y_i = k \text{ или } y_i = m\}$
- Обучаем  $a_{km}(x)$  на  $X_{km}$
- Итоговая модель:

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} \sum_{m=1}^K [a_{km}(x) = k]$$

# All-vs-all

- Нужно обучать порядка  $K^2$  моделей
- Зато каждую обучаем на небольшой выборке

# Доля ошибок

- Функционал ошибки — доля ошибок (error rate)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

- Нередко измеряют долю верных ответов (accuracy):

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i]$$

- Подходит для многоклассового случая!

# Общие подходы

## Микро-усреднение

Вычисляем  $TP_k, FP_k, FN_k, TN_k$  для каждого класса

Суммируем по всем классам, получаем TP, FP, FN, TN

Подставляем их в формулу для precision/recall/...

Крупные классы вносят больший вклад

## Макро-усреднение

Вычисляем нужную метрику для каждого класса (например,  $precision_1, \dots, precision_K$ )

Усредняем по всем классам

Игнорирует размеры классов



Как делать нелинейные модели

# Предсказание стоимости квартиры

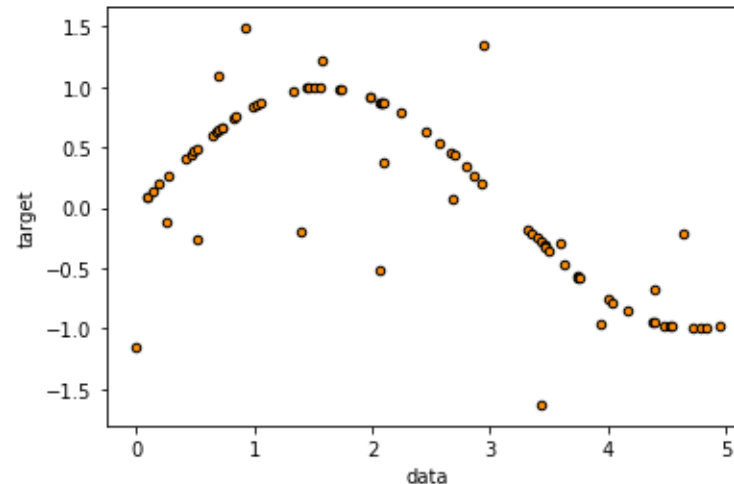
- Признаки: площадь, этаж, расстояние до метро и т.д.
- Целевая переменная: рыночная стоимость квартиры

# Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ + w_3 * (\text{расстояние до метро}) + \dots$$

- Вряд ли признаки линейно связаны с целевой переменной



# Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ + w_3 * (\text{расстояние до метро}) + \dots$$

- Вряд ли признаки не связаны между собой

# Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ & + w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2 \\ & + w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2 \\ & + w_7 * (\text{площадь}) * (\text{этаж}) + \dots \end{aligned}$$

# Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж}) \\ & + w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2 \\ & + w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2 \\ & + w_7 * (\text{площадь}) * (\text{этаж}) + \dots \end{aligned}$$

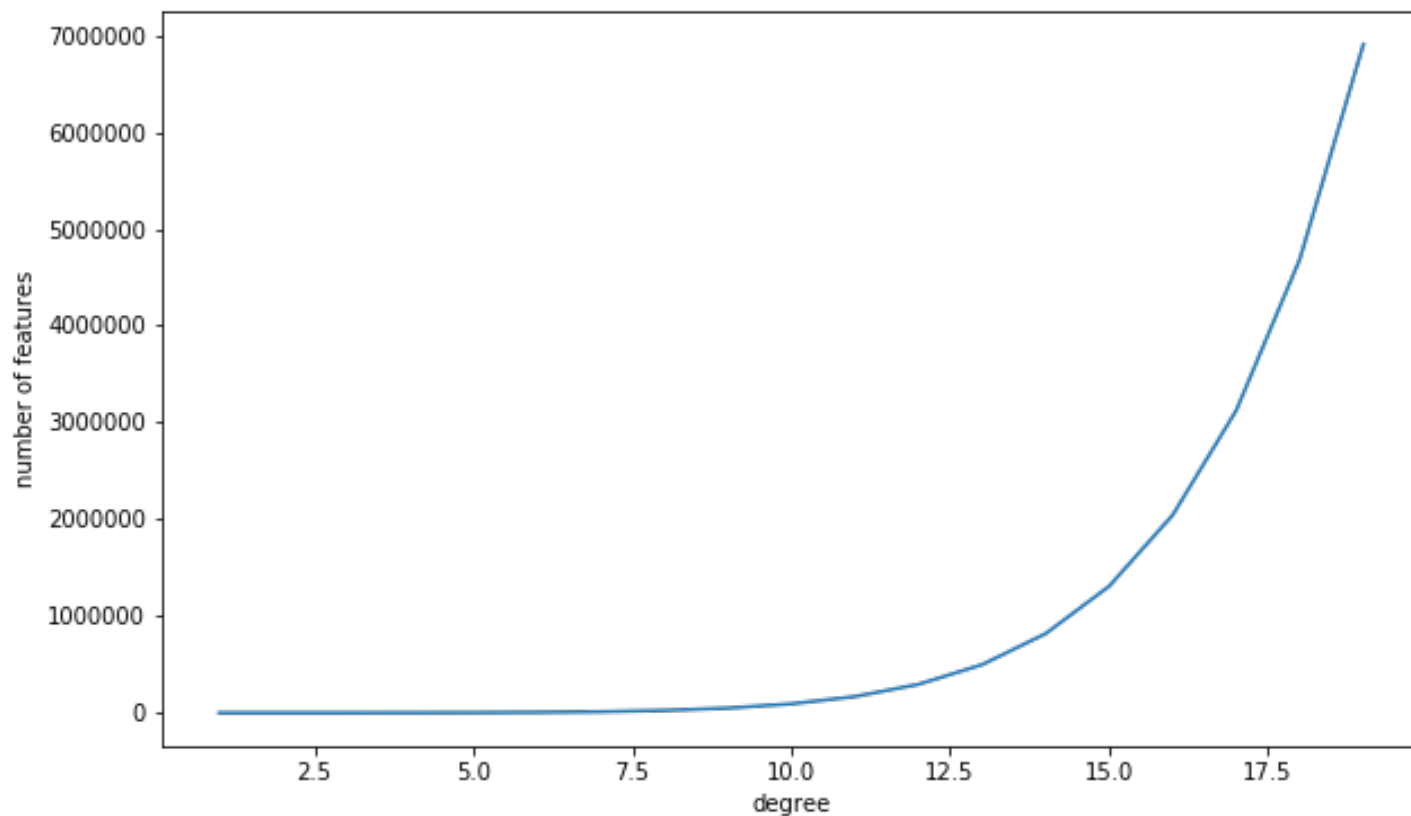
- Может быть сложно интерпретировать модель
- Что такое  $(\text{расстояние до метро}) * (\text{этаж})^2$ ?

# Предсказание стоимости квартиры

- Допустим, изначально имеем 10 признаков
- Полиномиальных степени 2: 55
- Полиномиальных степени 3: 220
- Полиномиальных степени 4: 715

# Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:





# Предсказание стоимости квартиры

- Линейная модель с полиномиальными бинаризованными признаками:

$$a(x) = w_0 + w_1 * [30 < \text{площадь} < 50]$$

$$+ w_2 * [50 < \text{площадь} < 80] + \dots$$

$$+ w_{20} * [2 < \text{этаж} < 5] + \dots$$

$$+ w_{100} * [30 < \text{площадь} < 50][2 < \text{этаж} < 5] + \dots$$

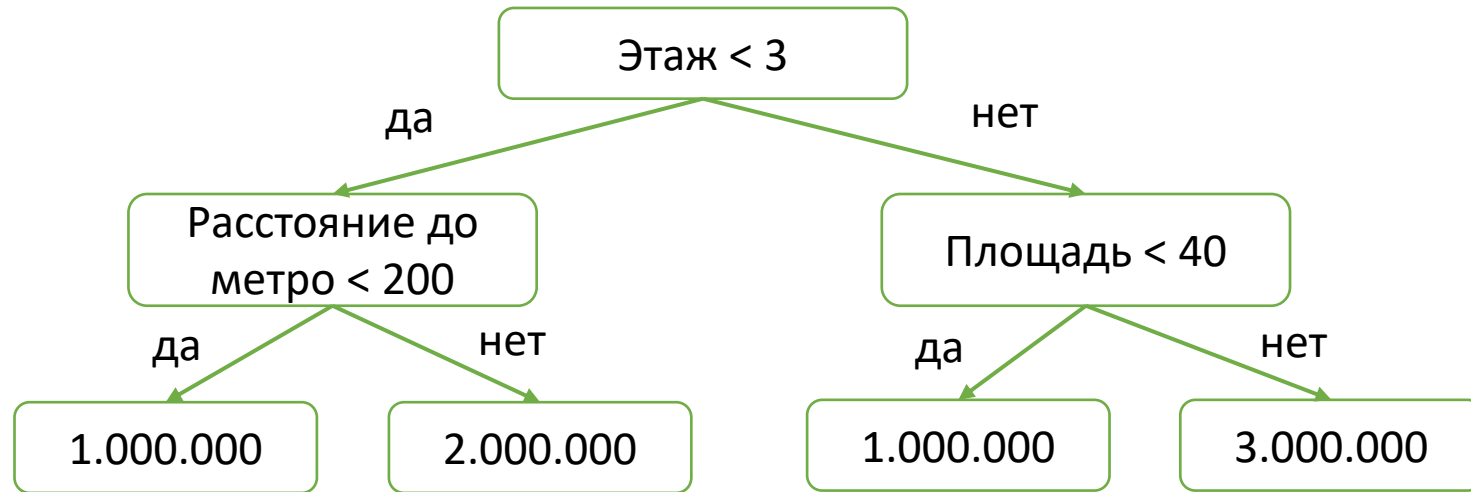
- Признаки интерпретируются куда лучше:  $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][100 < \text{расстояние до метро} < 500]$
- Но их станет ещё больше!

Решающие деревья

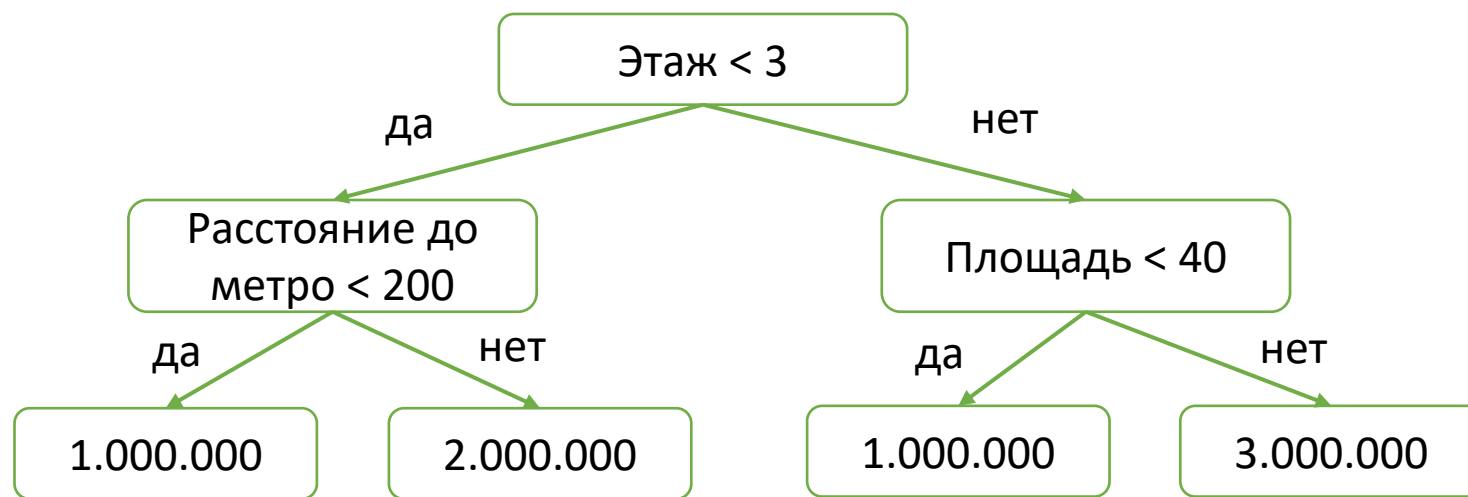
# Логические правила

- $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][500 < \text{расстояние до метро} < 1000]$
- Легко объяснить, как работают
- Находят нелинейные закономерности
- Нужно как-то искать хорошие логические правила
- Нужно уметь составлять модели из логических правил

# Решающее дерево

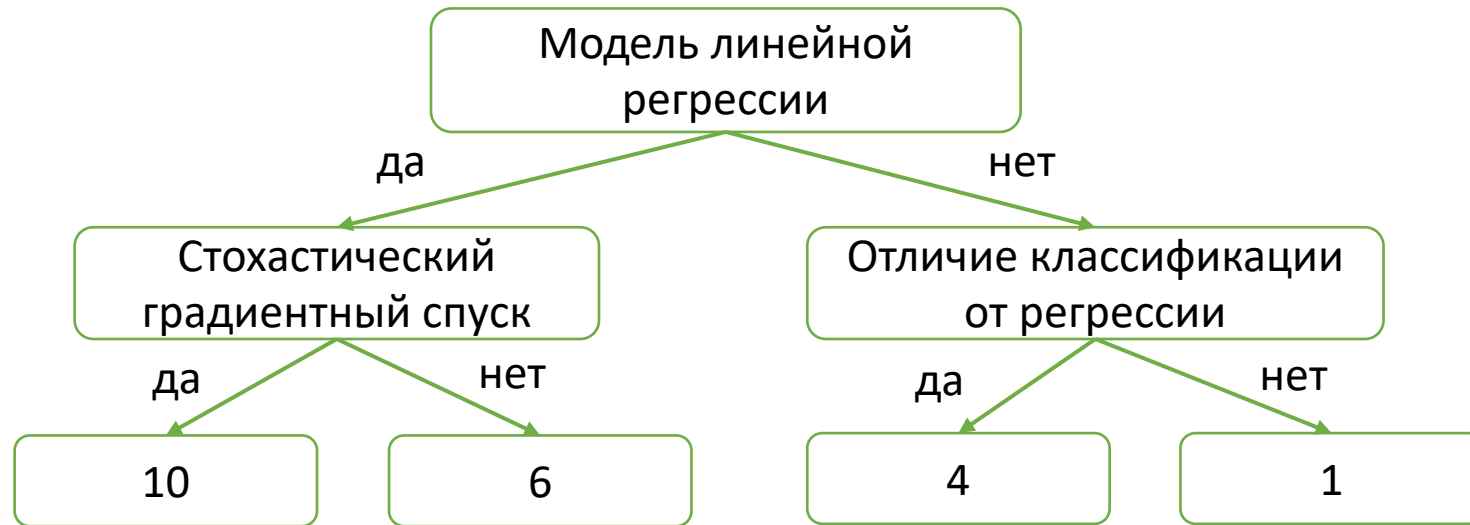


# Решающее дерево

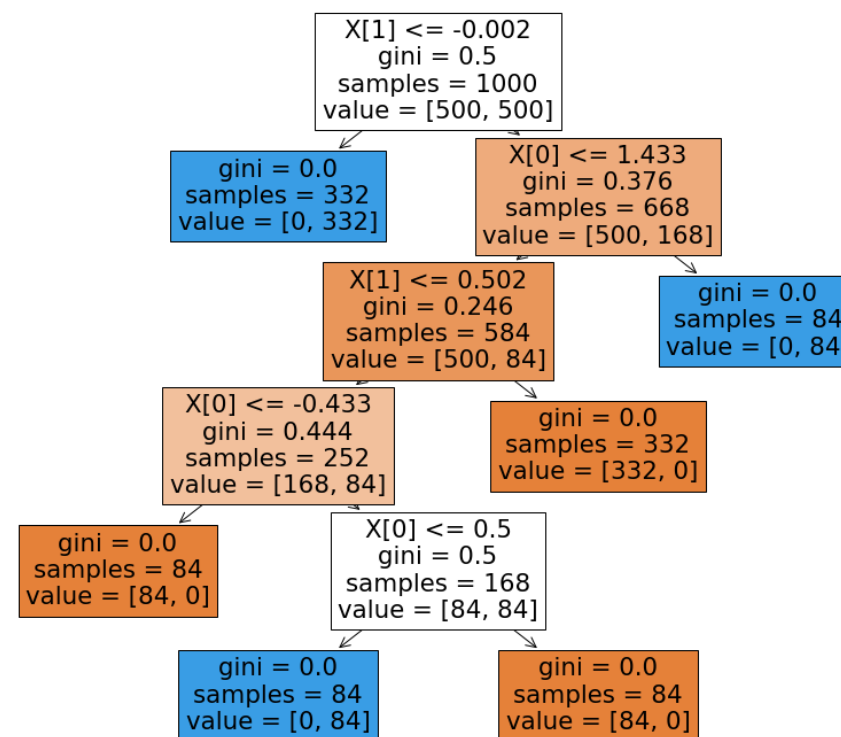
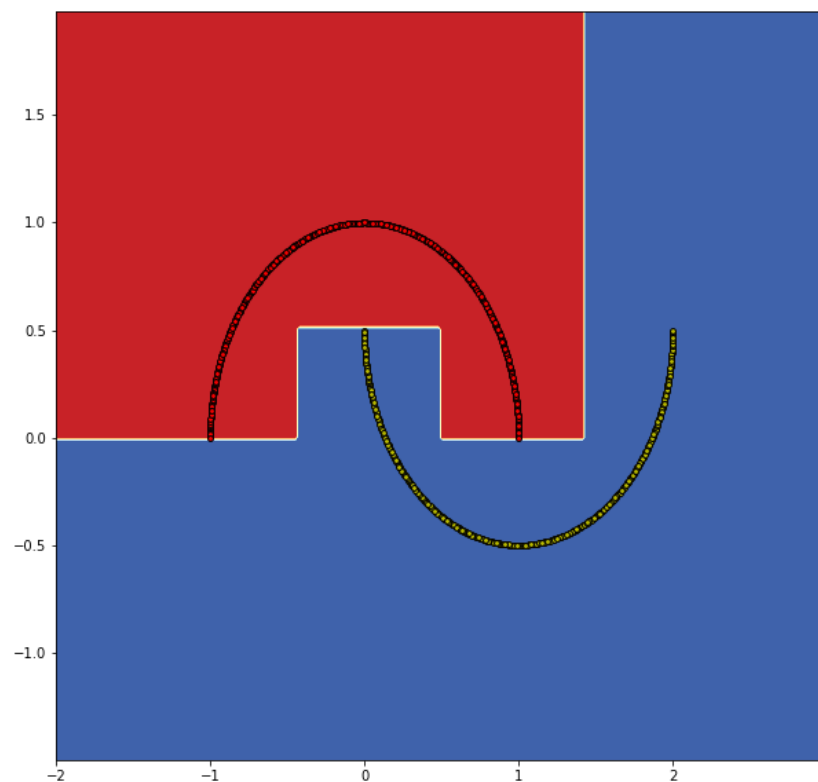


- Внутренние вершины: предикаты  $[x_j < t]$
- Листья: прогнозы  $s \in \mathbb{Y}$

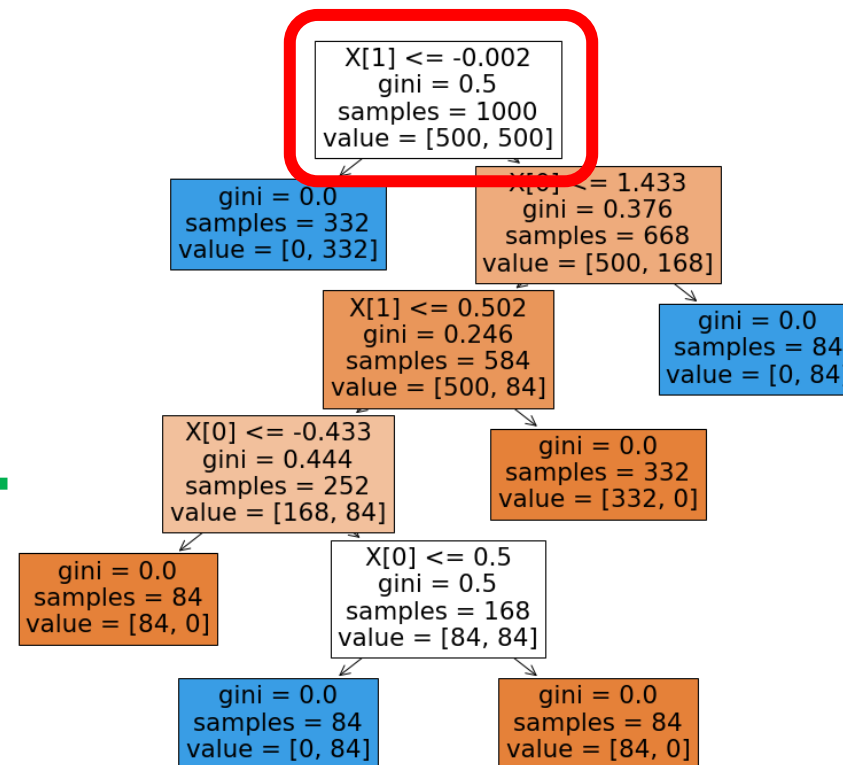
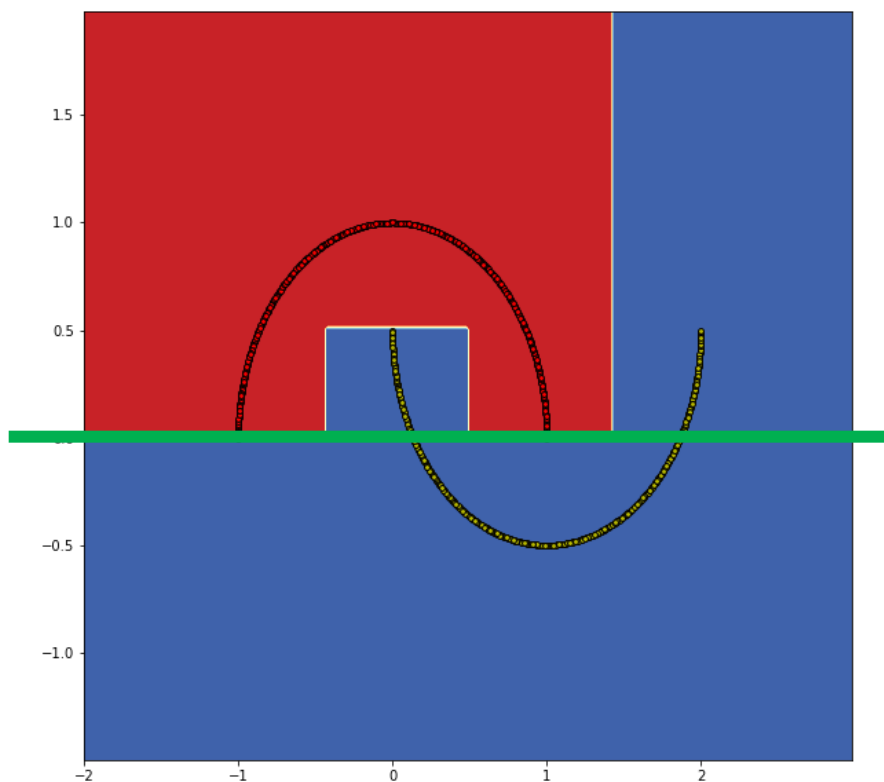
# Решающее дерево



# Решающее дерево

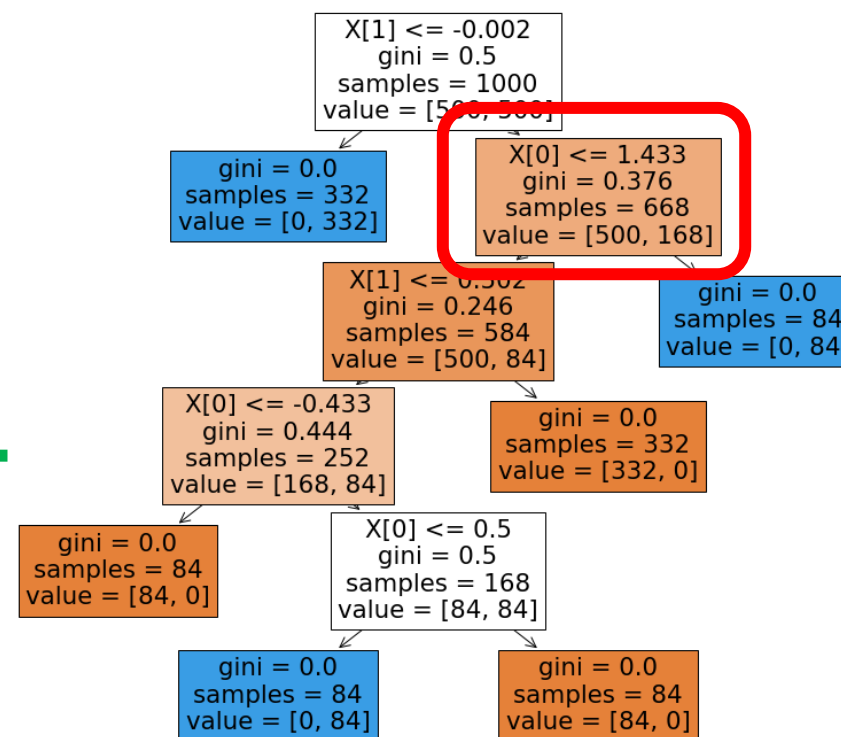
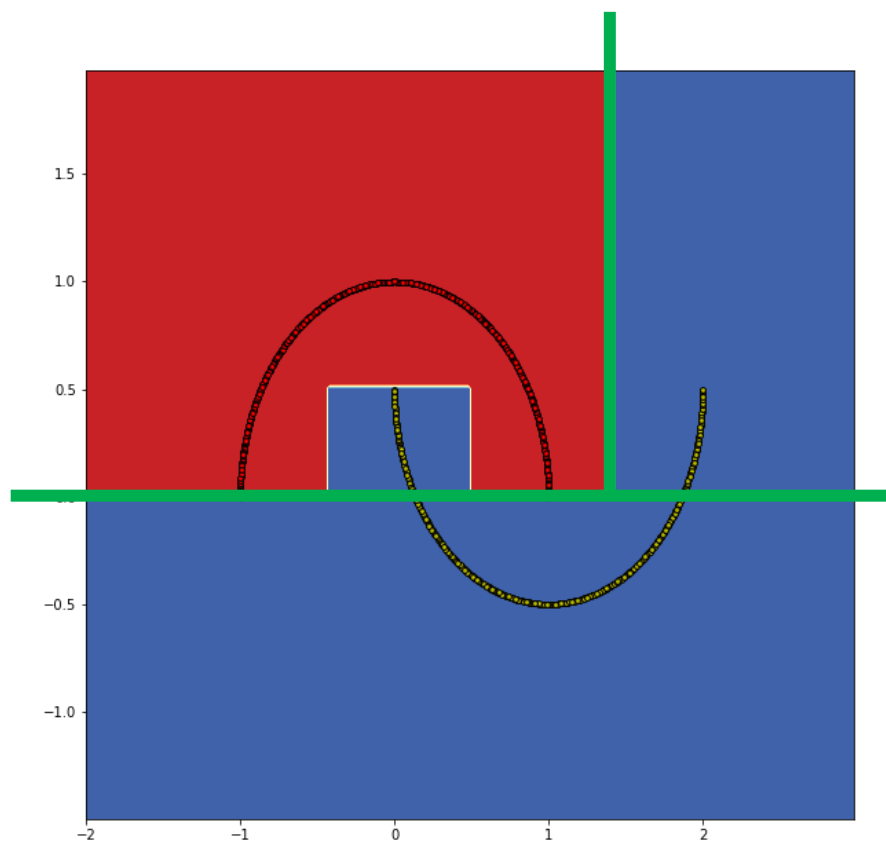


# Решающее дерево

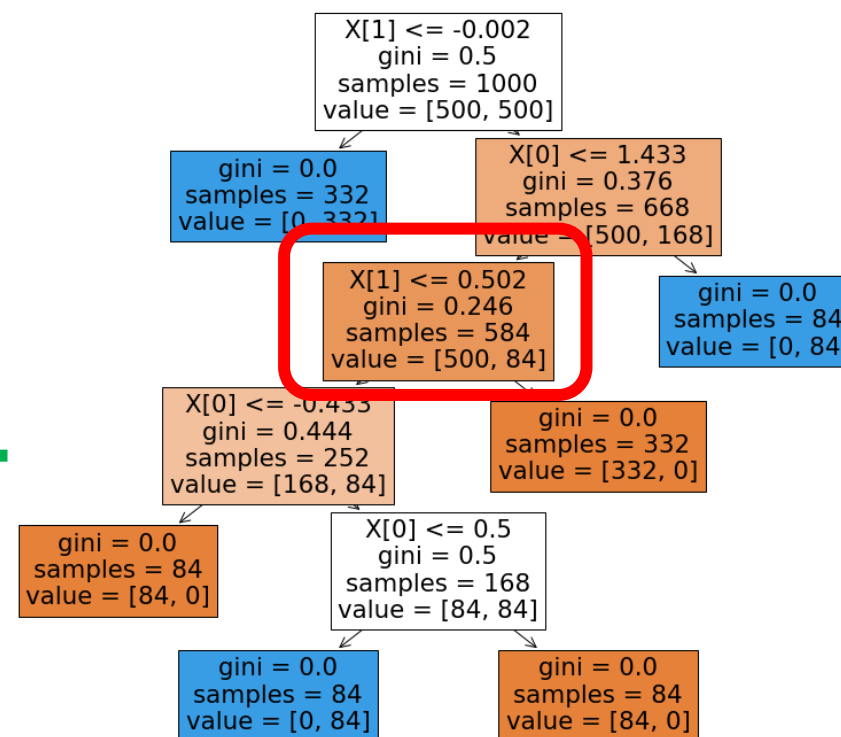
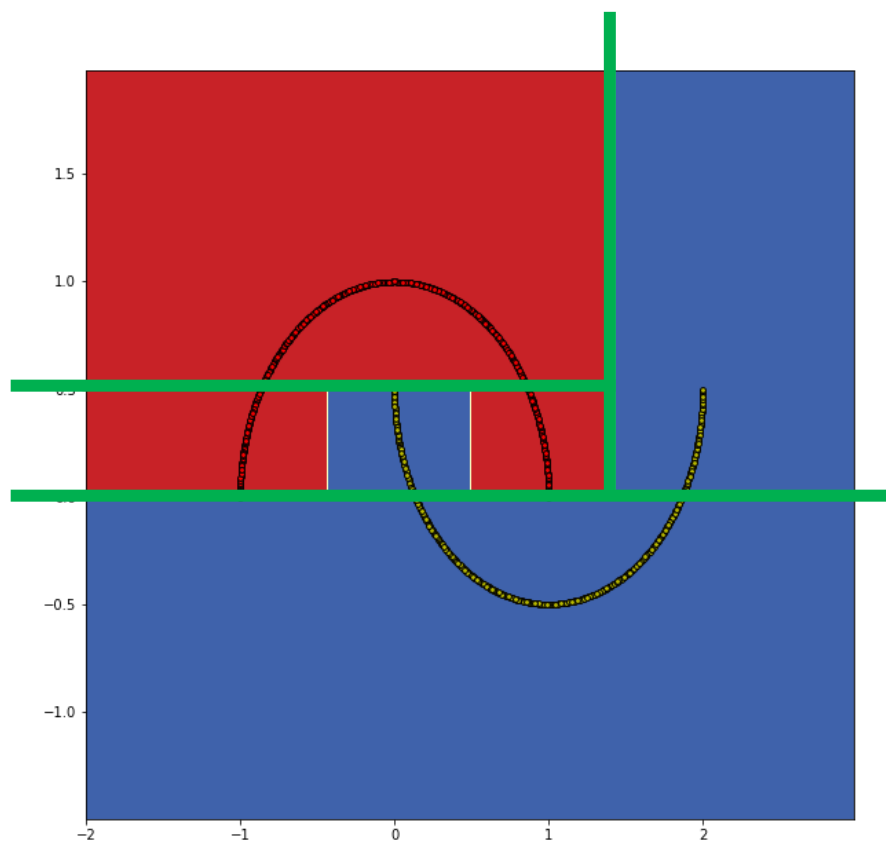




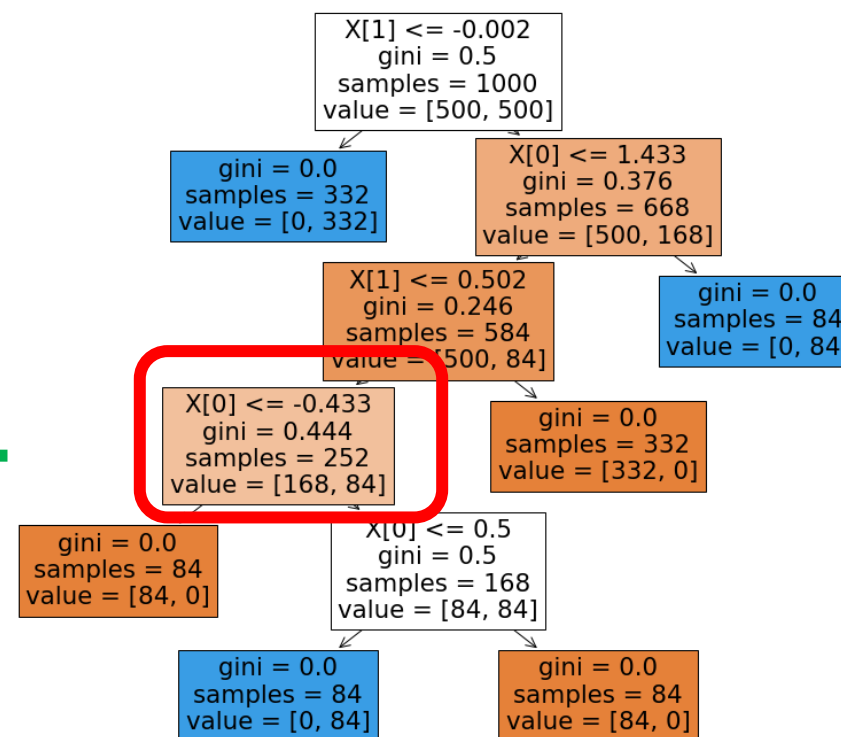
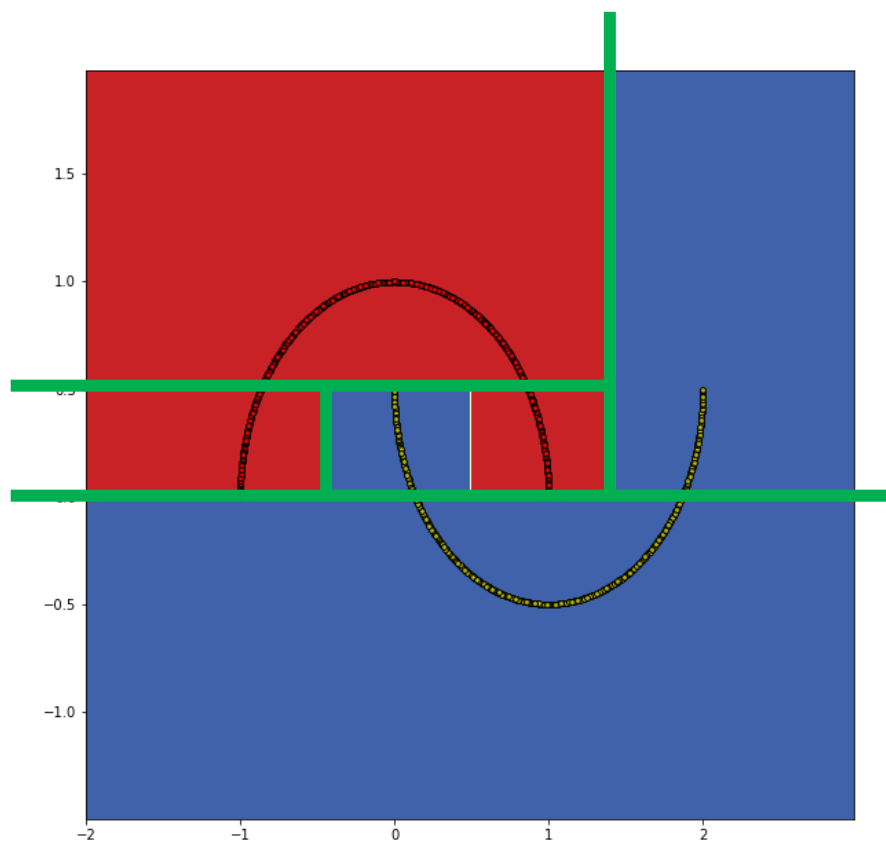
# Решающее дерево



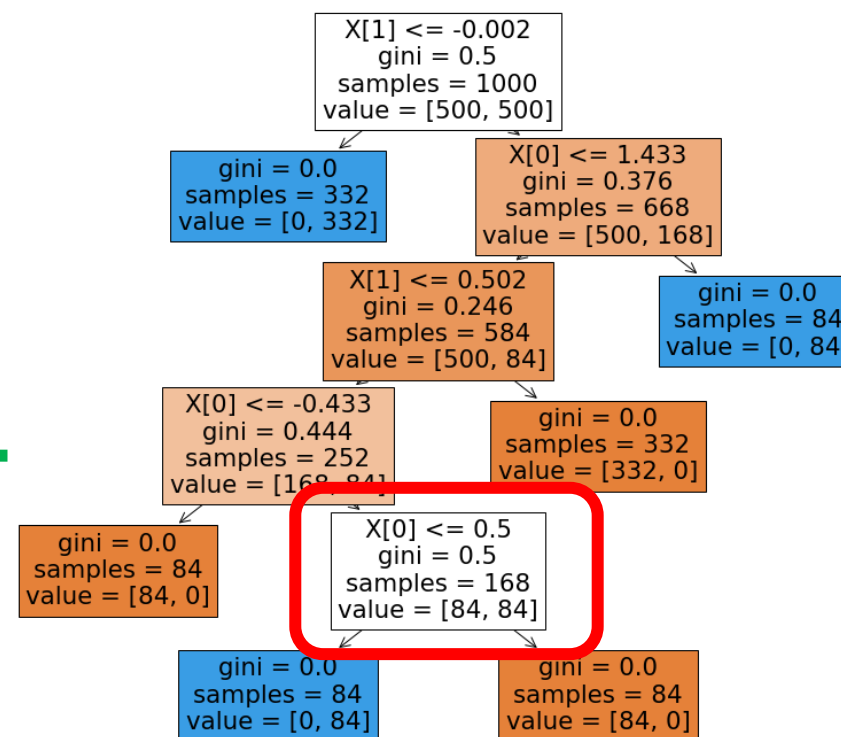
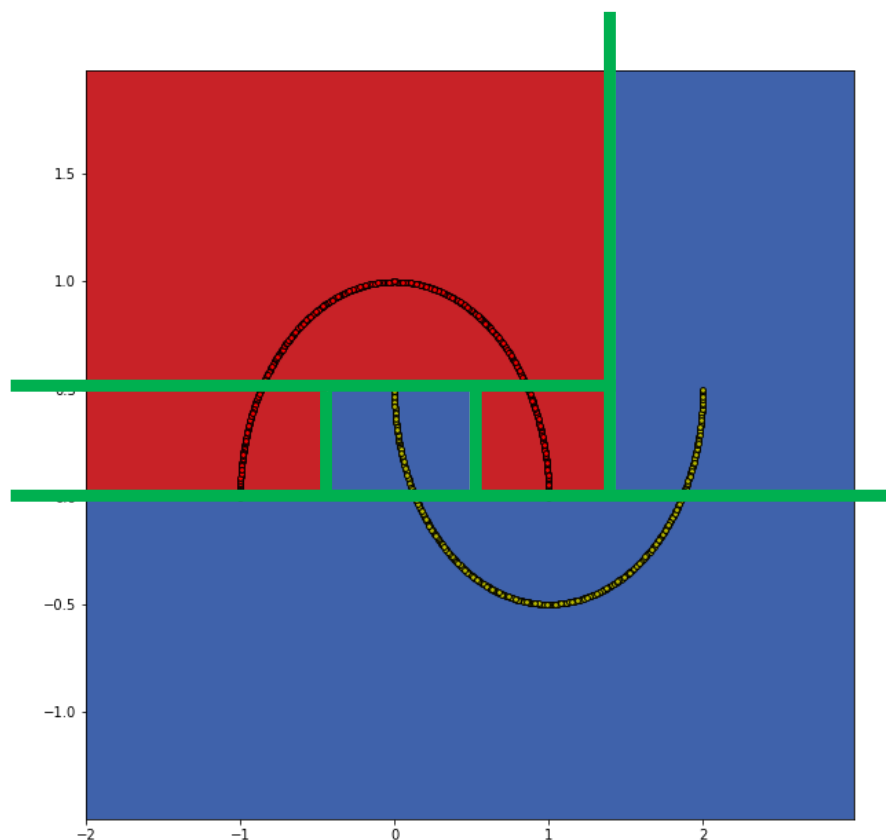
# Решающее дерево



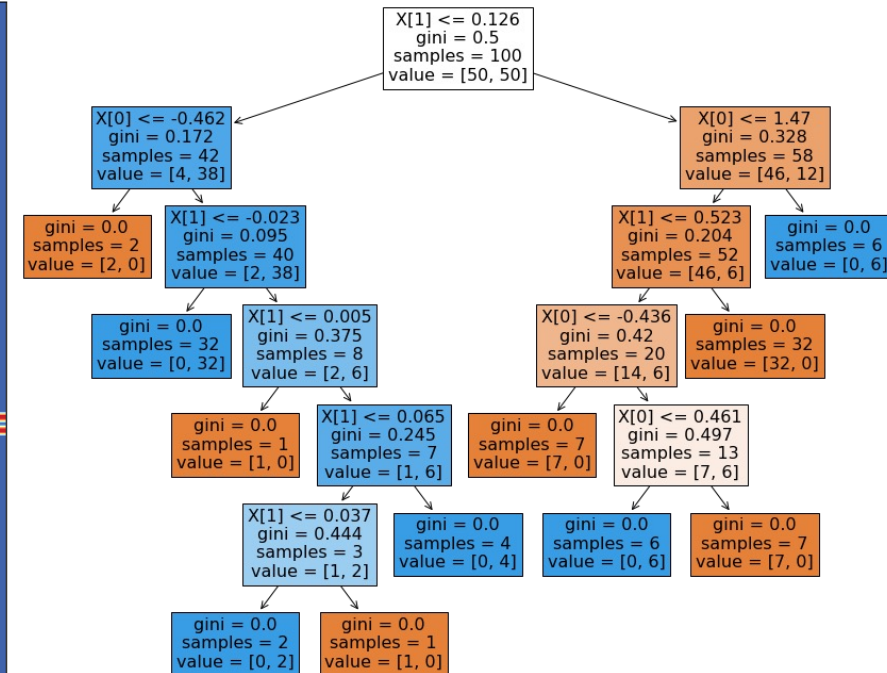
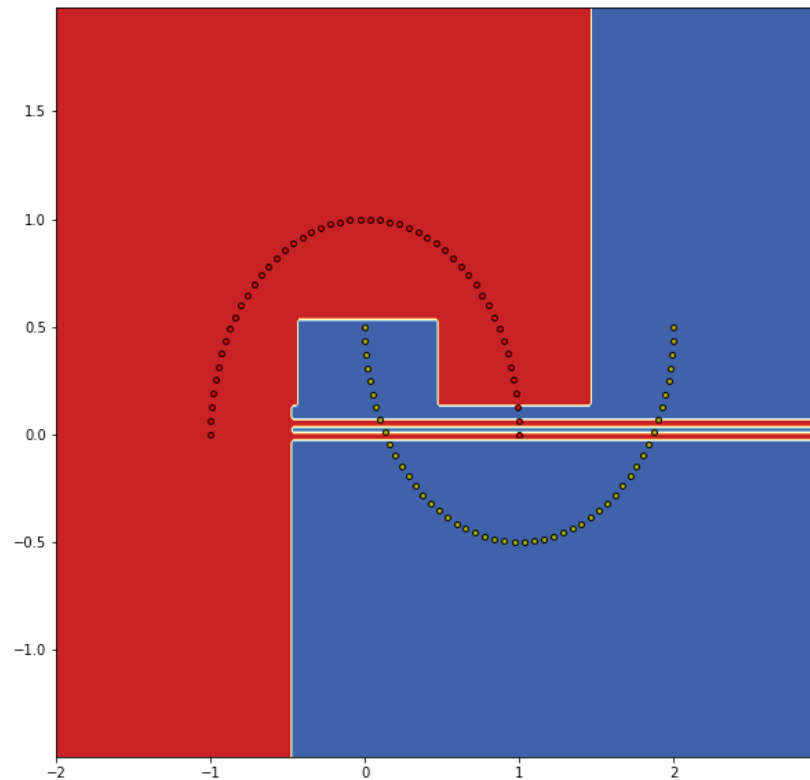
# Решающее дерево



# Решающее дерево



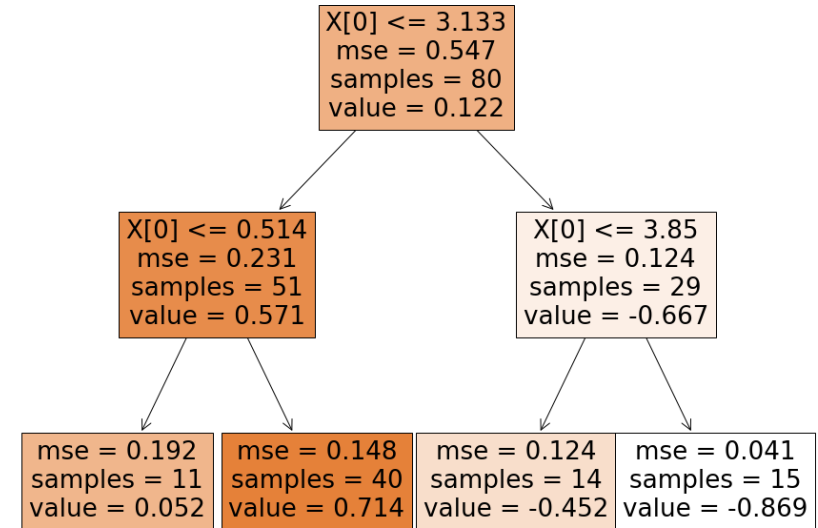
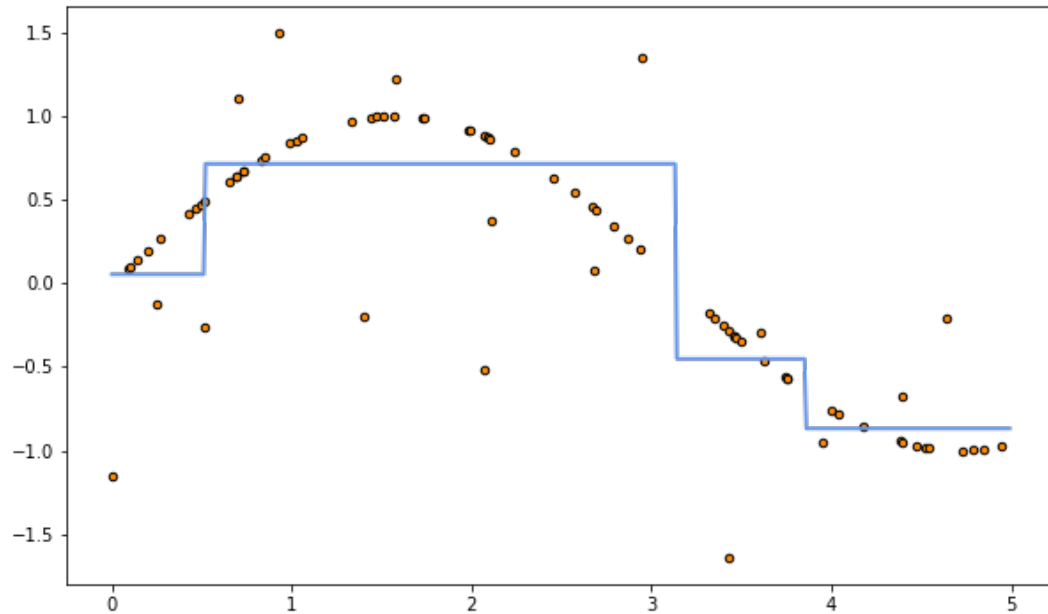
# Решающее дерево



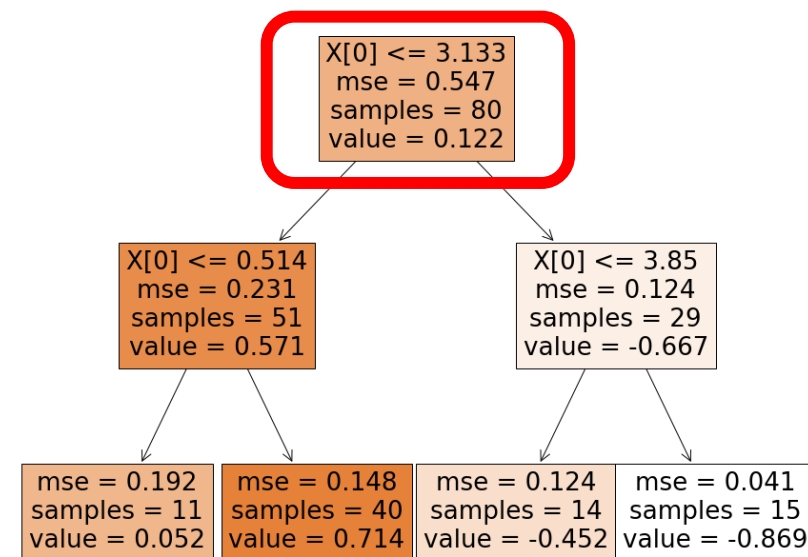
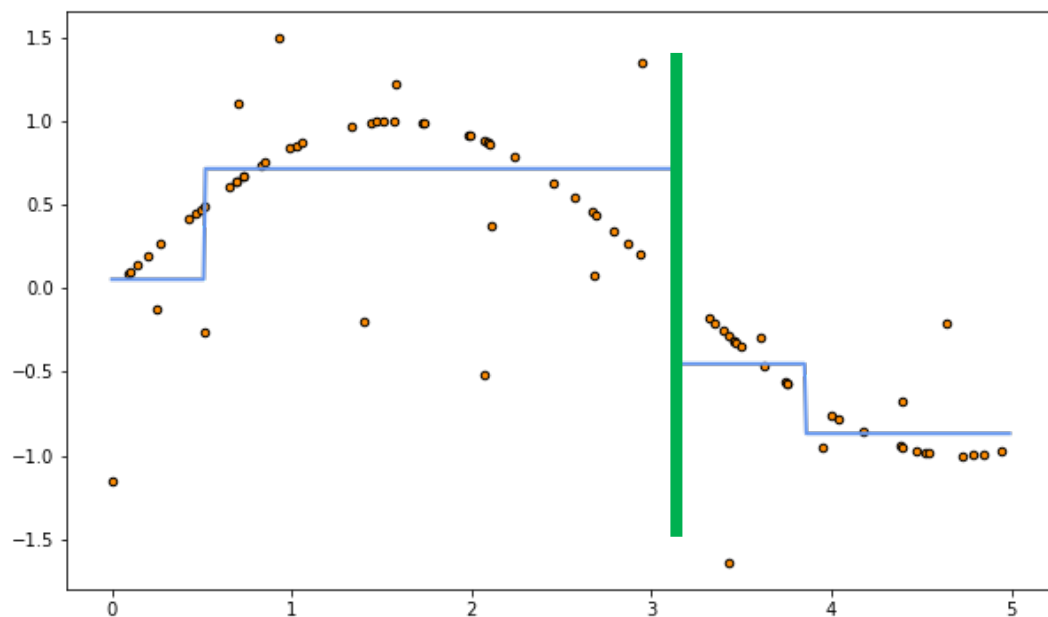
# Сложность дерева

- 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

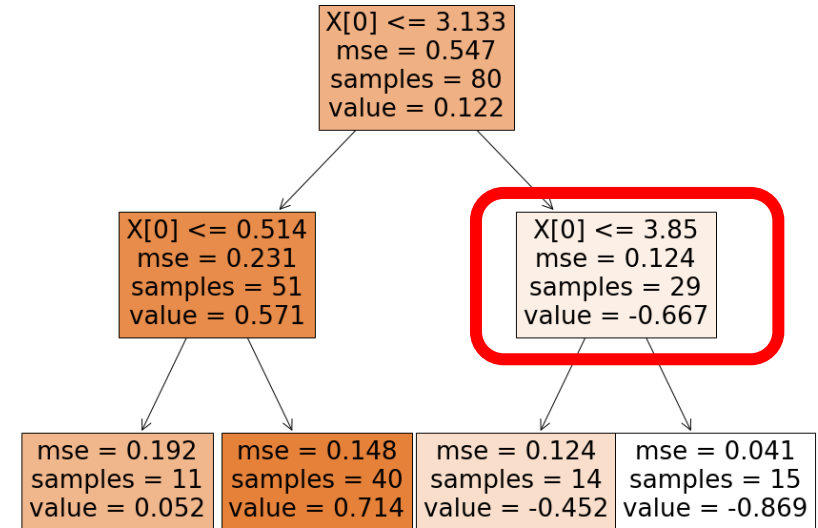
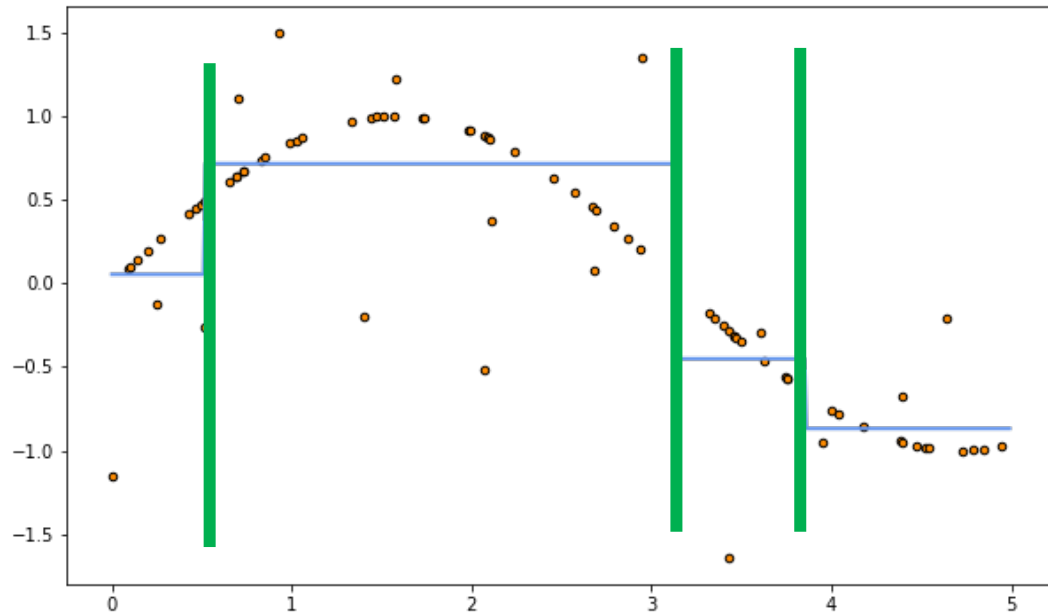


# Решающее дерево для регрессии

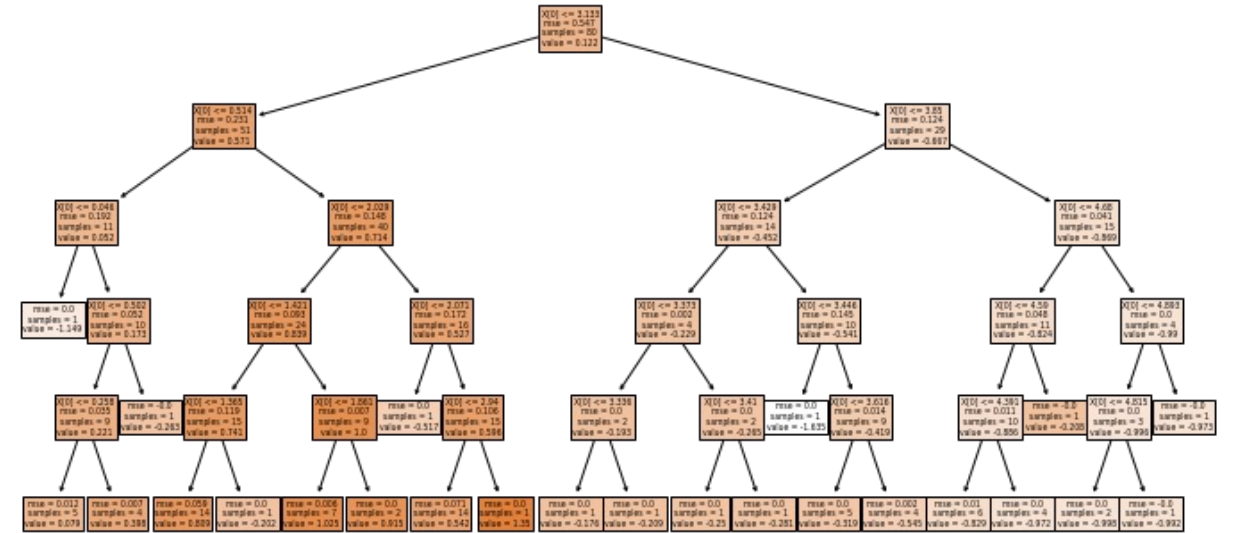
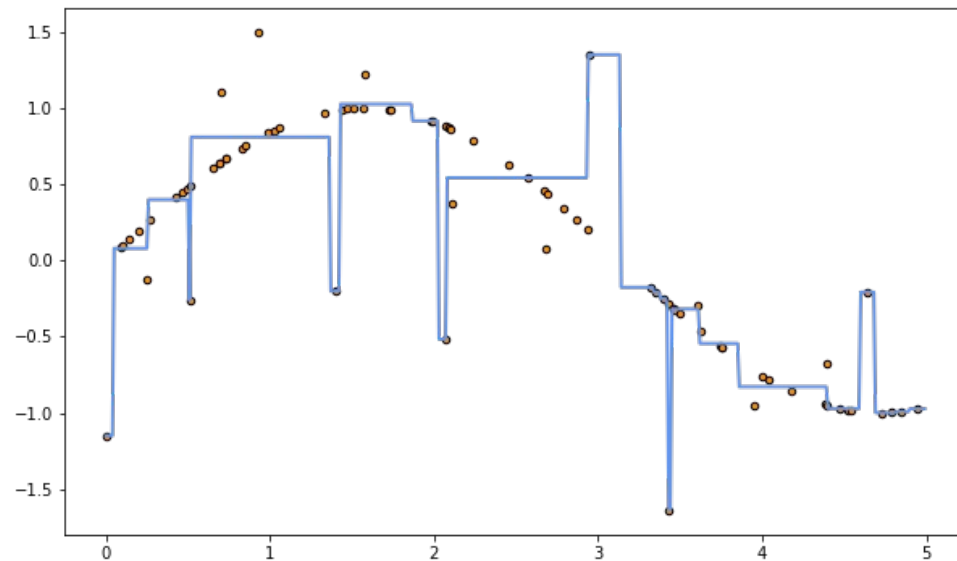




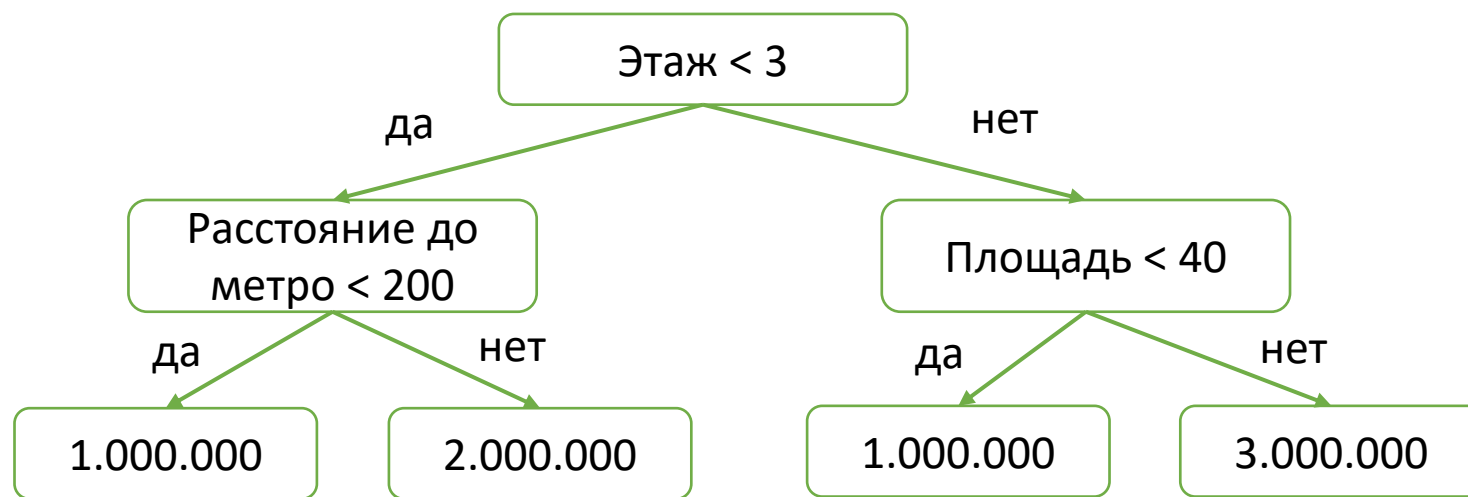
# Решающее дерево для регрессии



# Решающее дерево для регрессии



# Решающее дерево



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

# Предикаты

- Порог на признак  $[x_j < t]$  — 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:

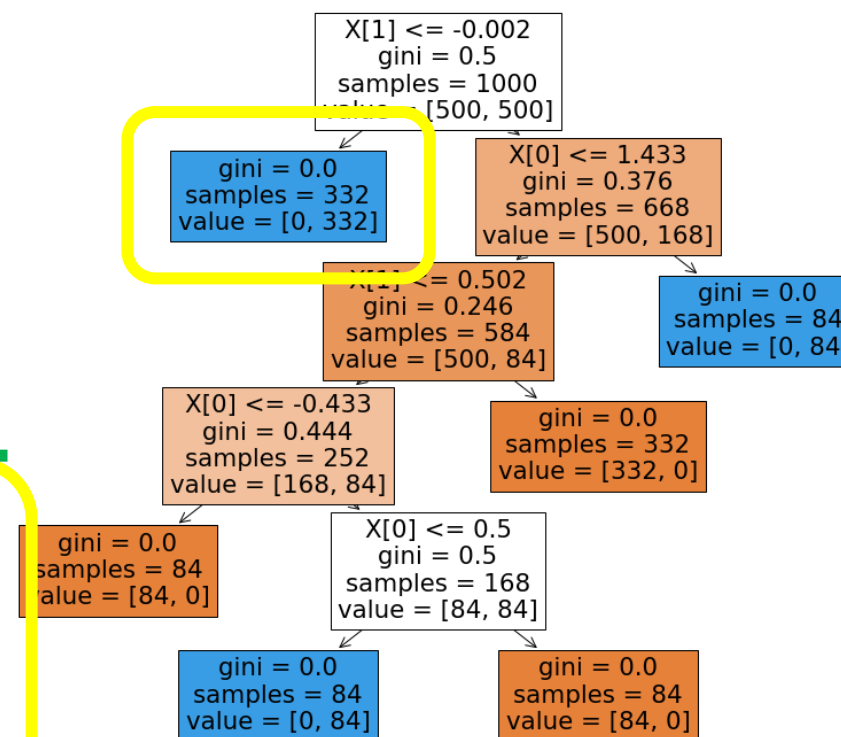
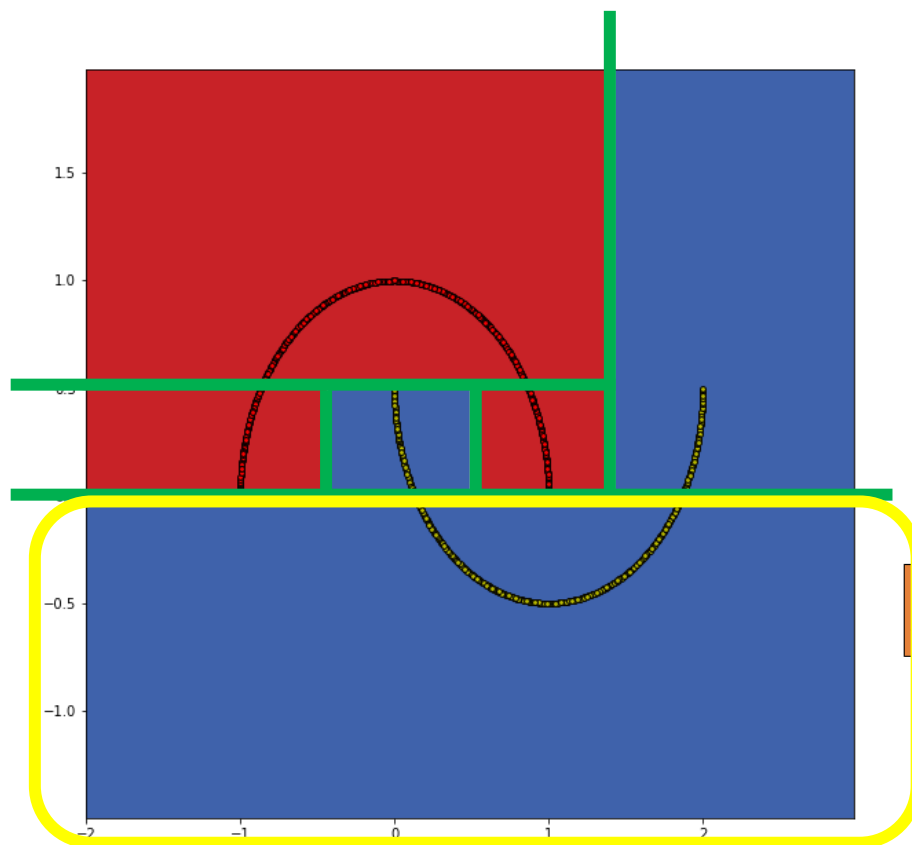
$$c_{vk} = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

# Прогнозы в листьях

- Можно усложнять листья
- Например:

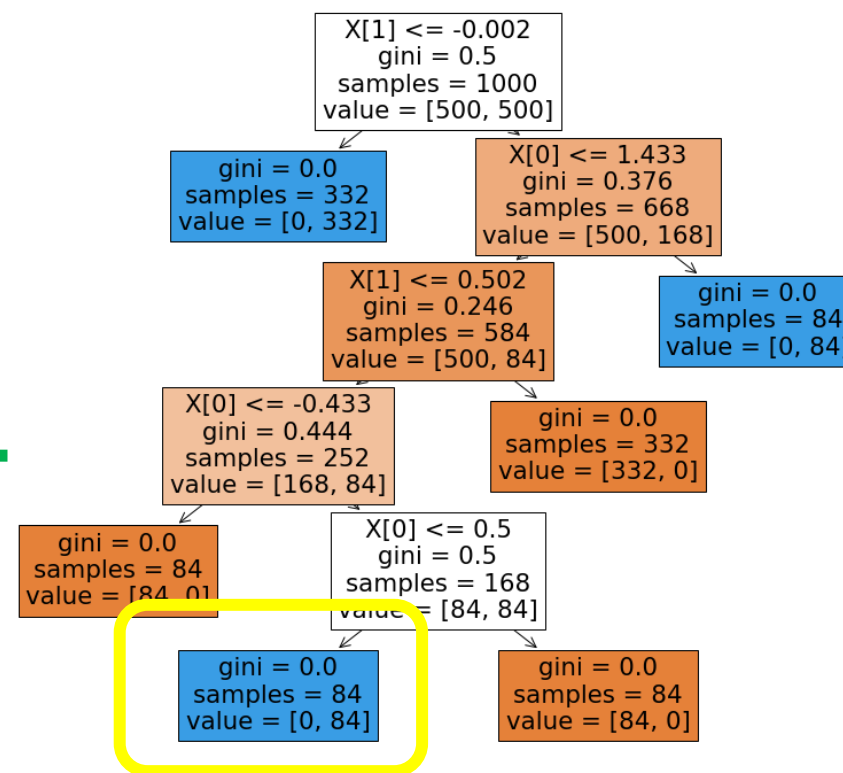
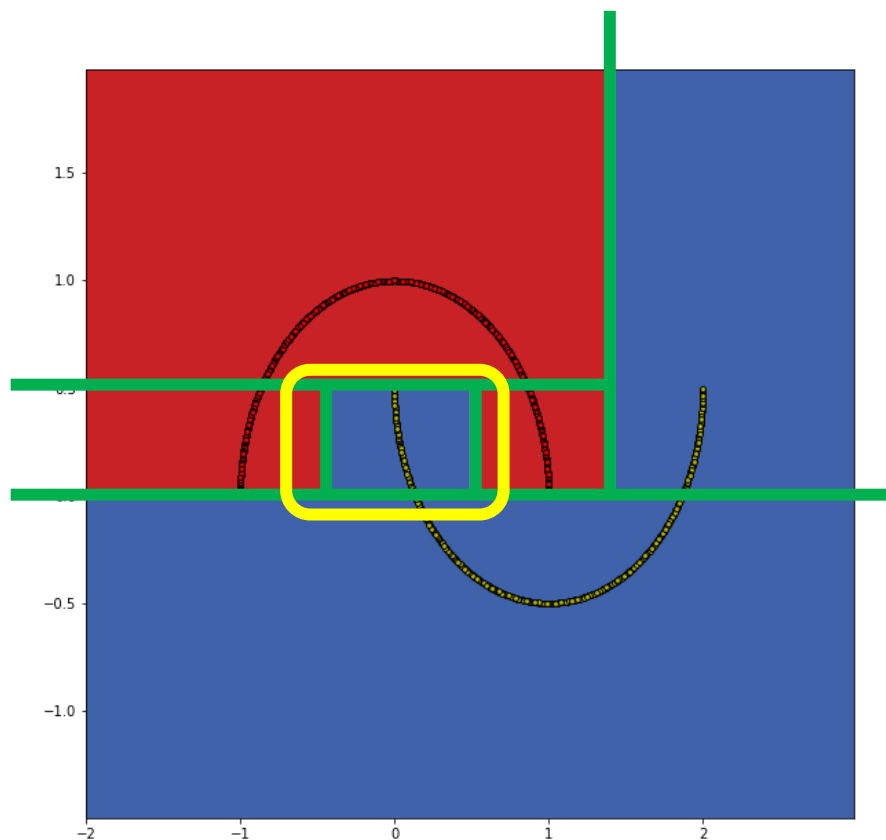
$$c_v(x) = \langle w_v, x \rangle$$

# Решающее дерево





# Решающее дерево



# Формула для дерева

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

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

# Формула для дерева

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

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