


ВВЕДЕНИЕ В МАШИННОЕ ОБУЧЕНИЕ

Лекция №5



План лекции

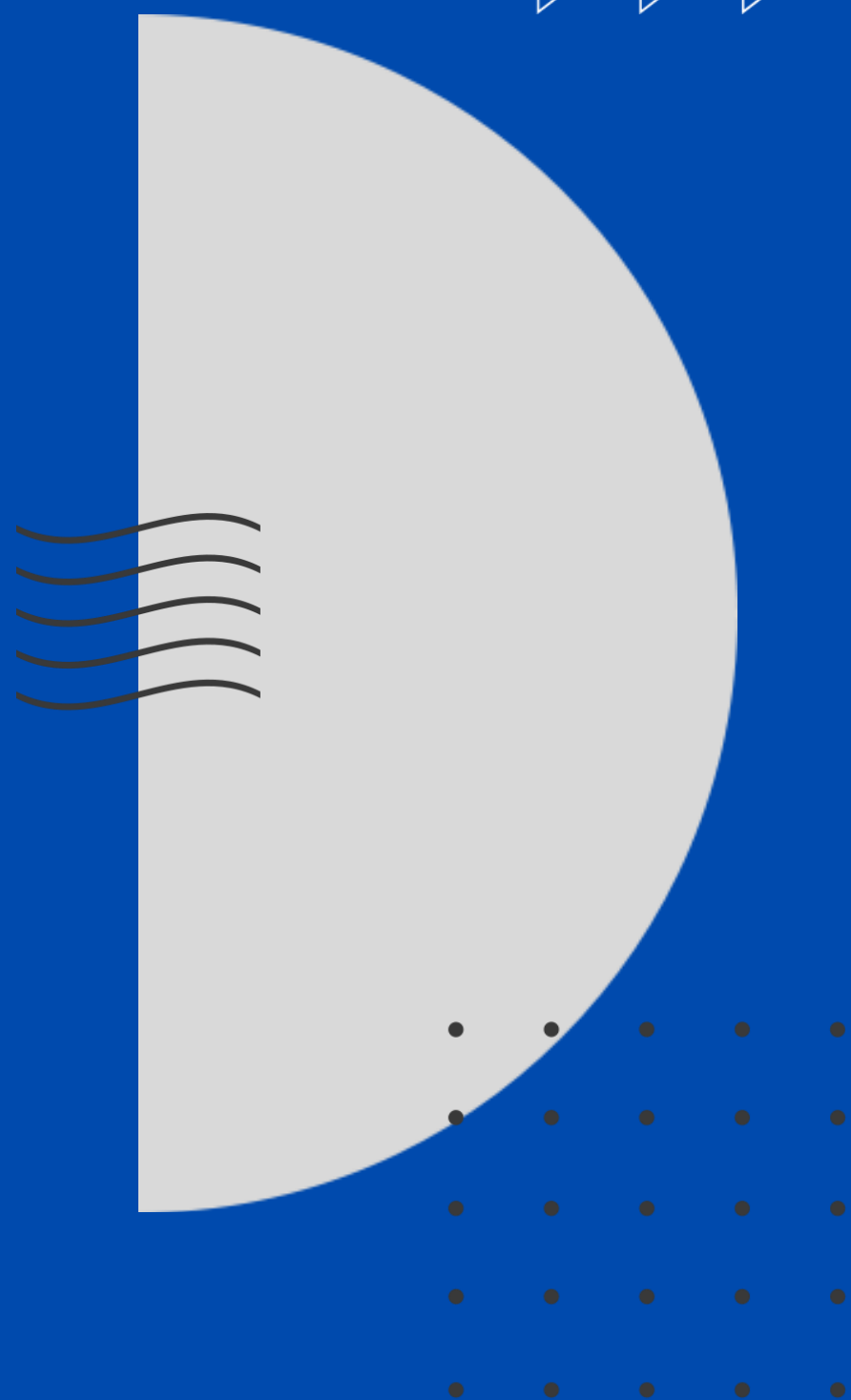
1. Решающие деревья
2. Выбор предикатов
3. Композиция моделей
4. Бэггинг
5. Смещение и разброс
6. Алгоритм случайного леса
7. Особенности леса





01

Решающие деревья

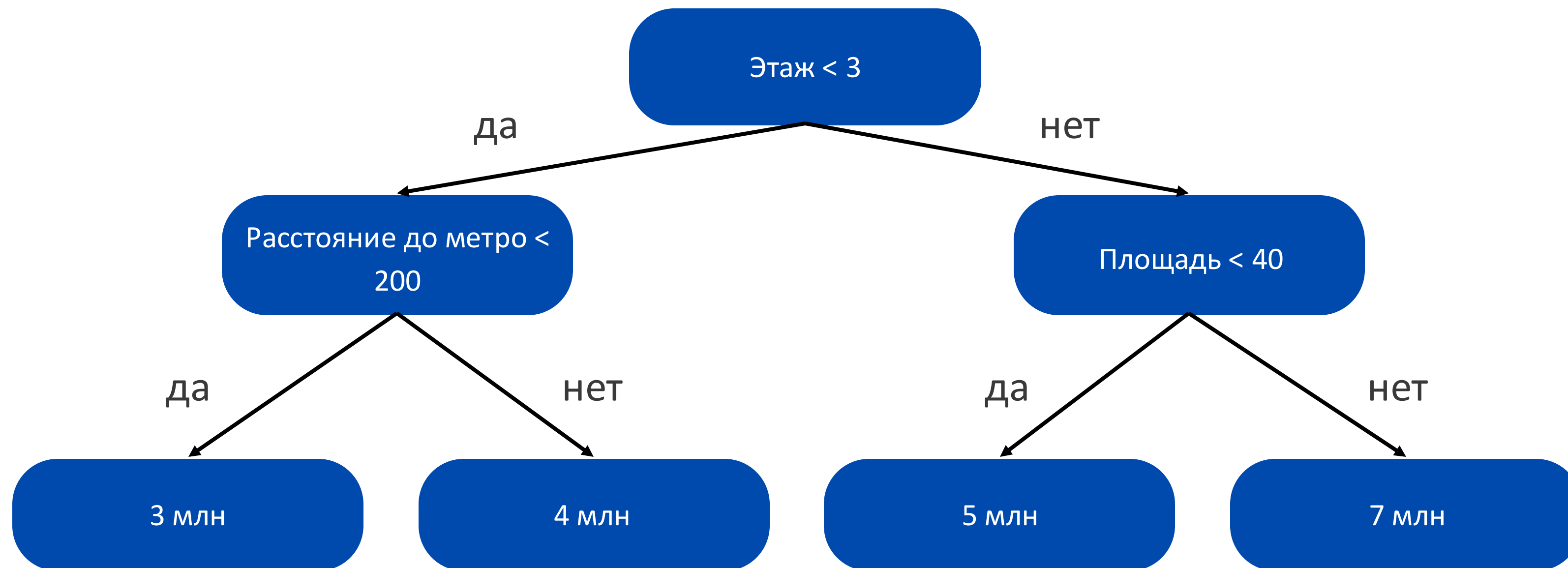


Логические правила

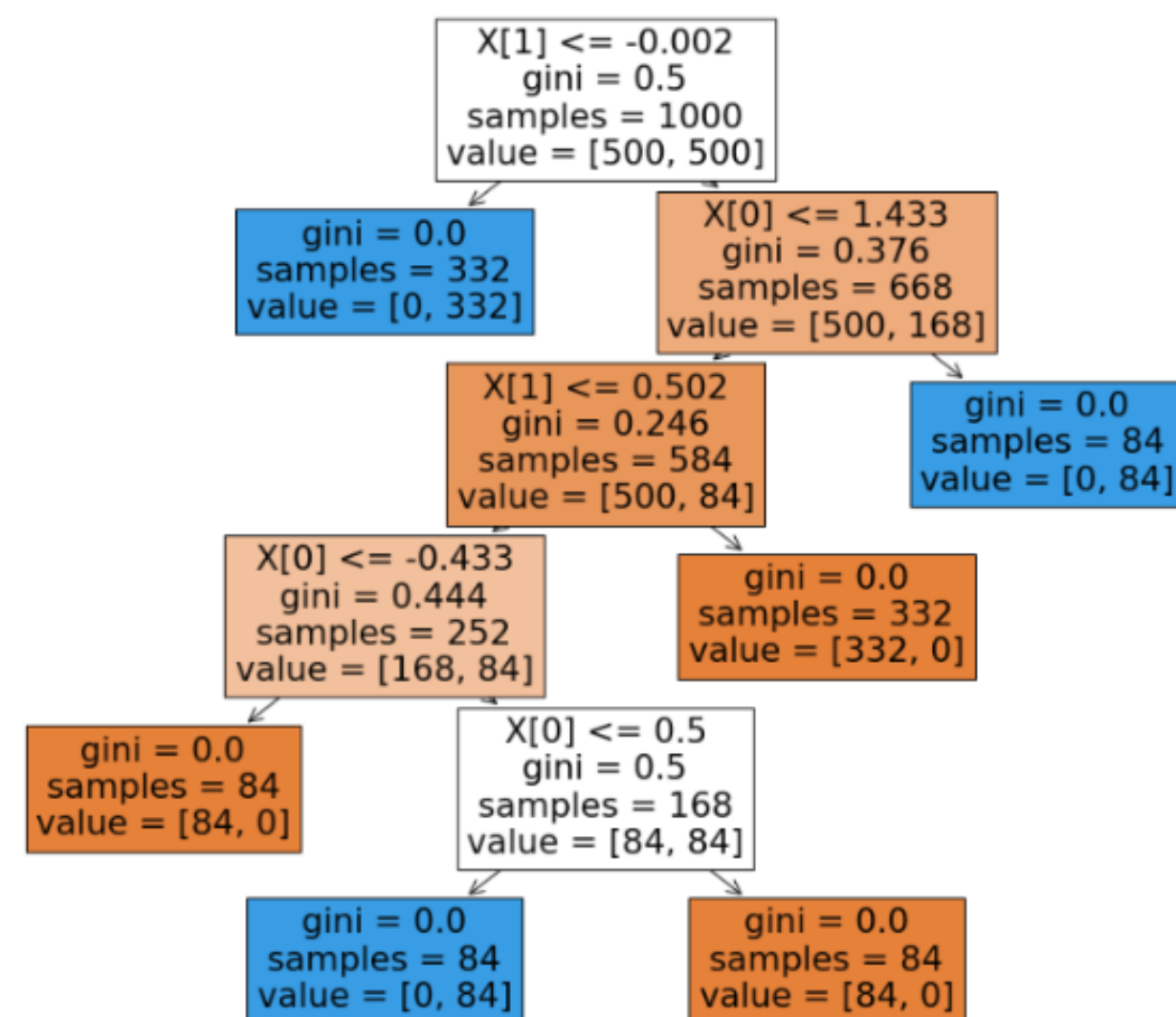
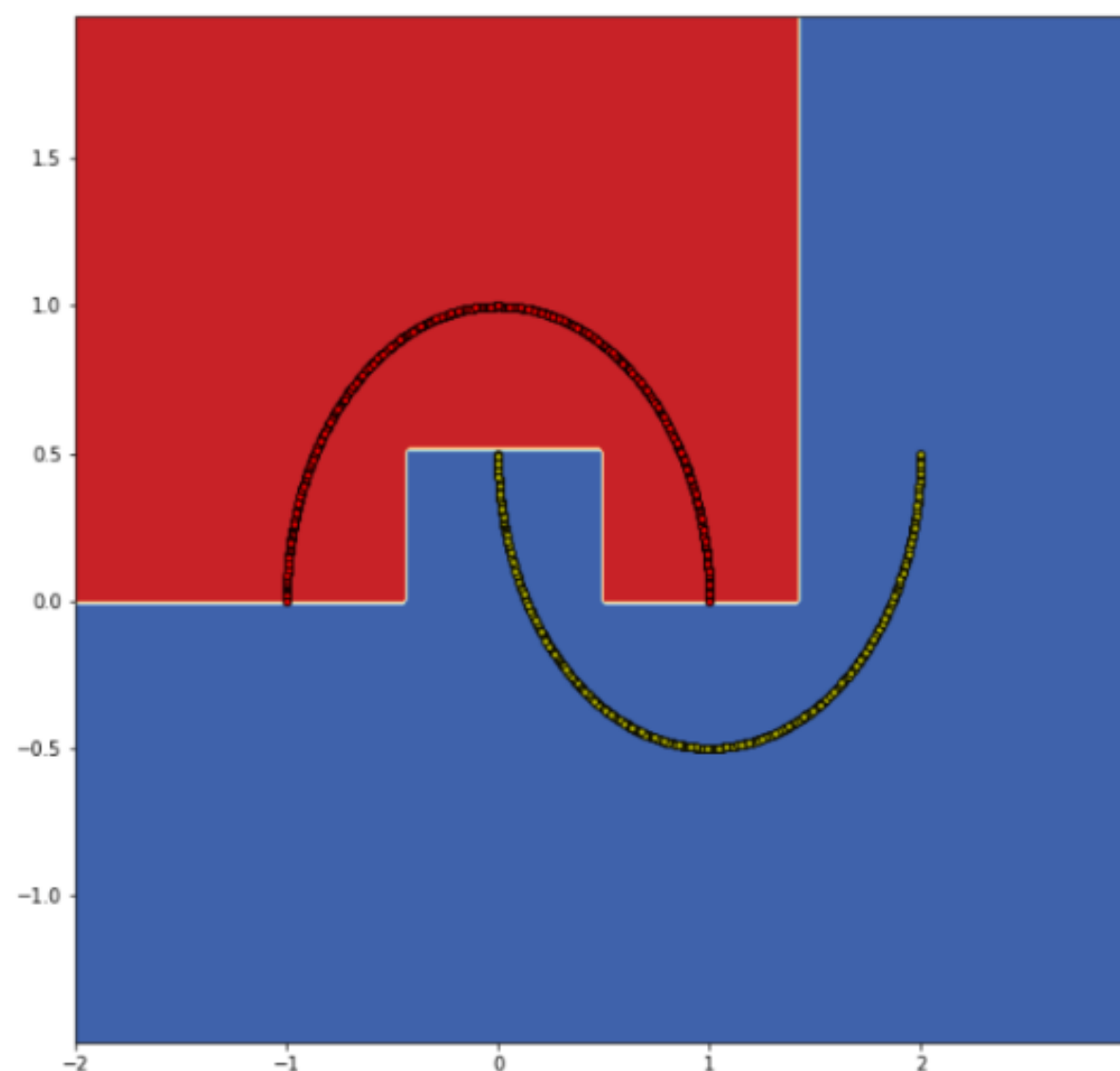
- $[100 > \text{площадь} > 50] \ \& \ [20 > \text{этаж} > 10] \ \& \ [200 > \text{расстояние до метро}]$
- Легко интерпретировать
- Нелинейные закономерности

- Как искать правила?
- Как из моделей составлять правила?

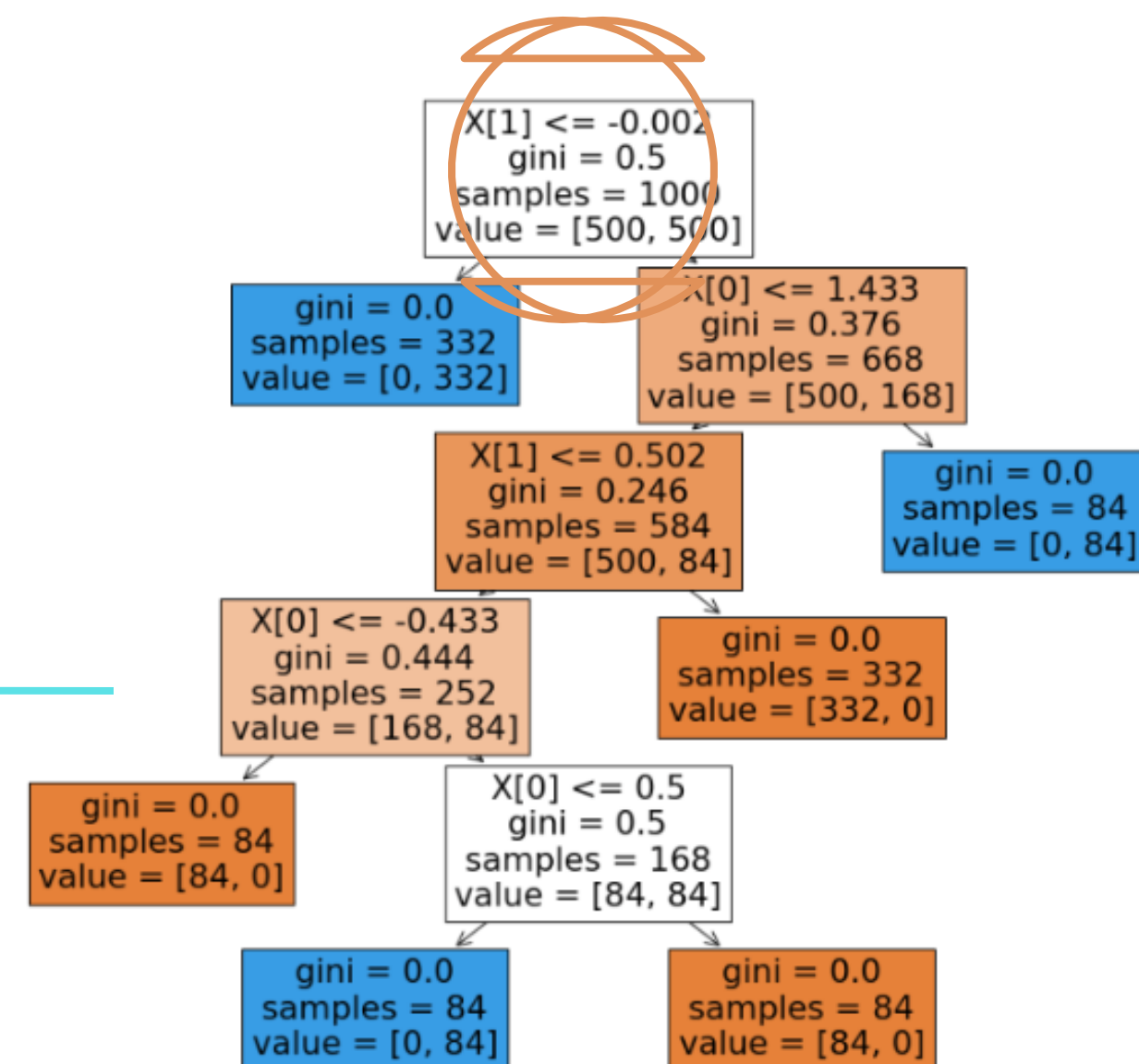
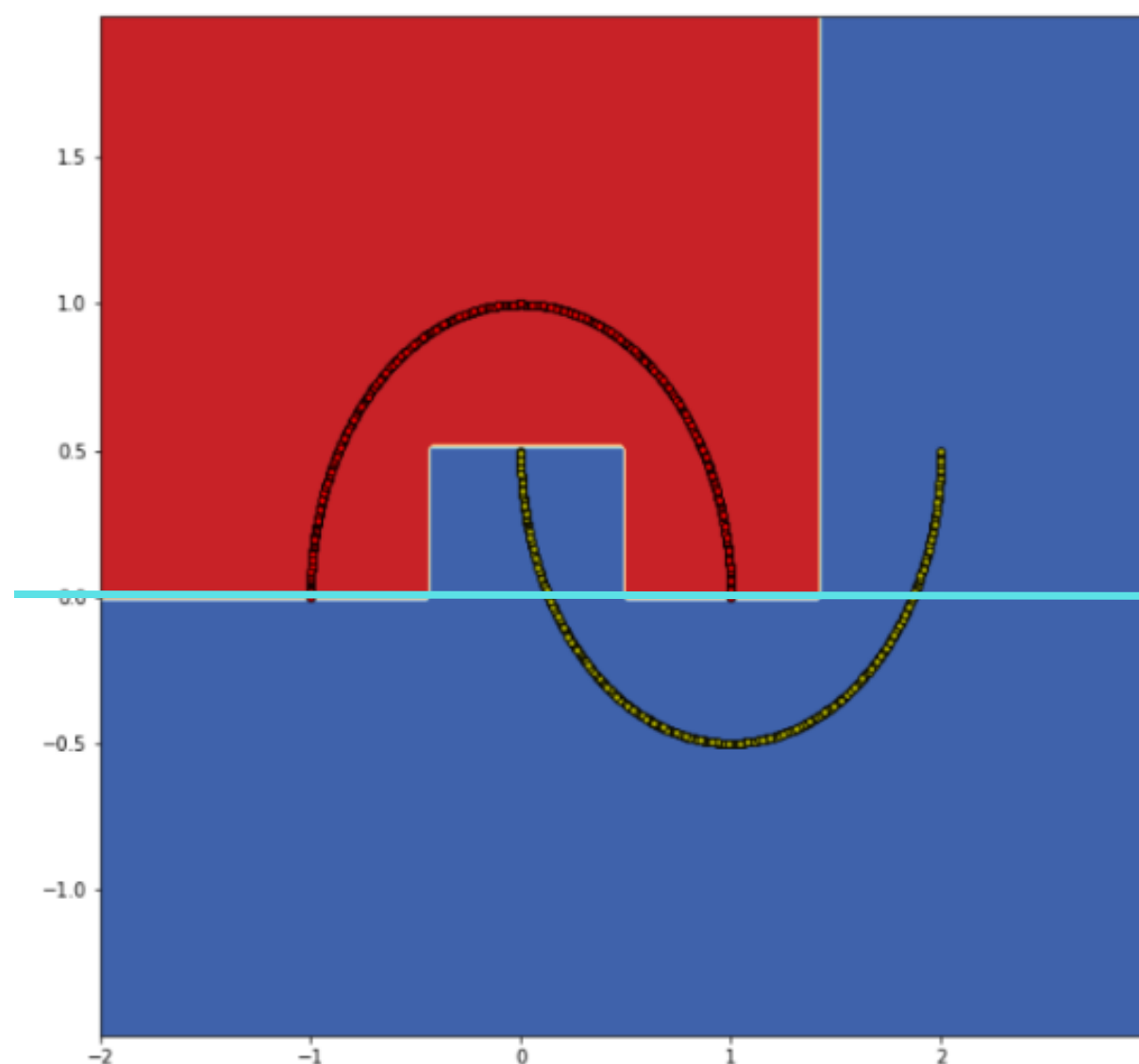
Решающее дерево



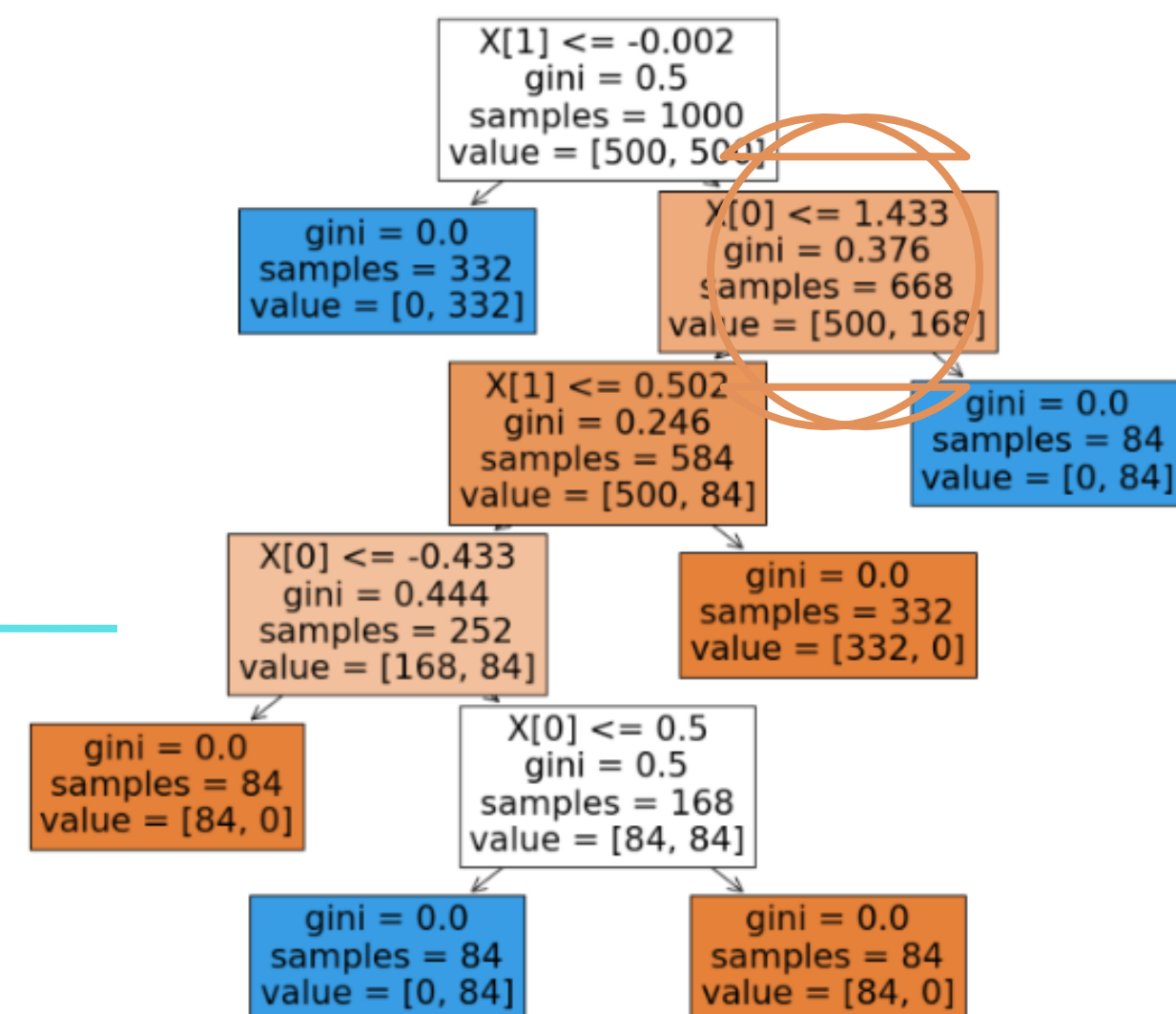
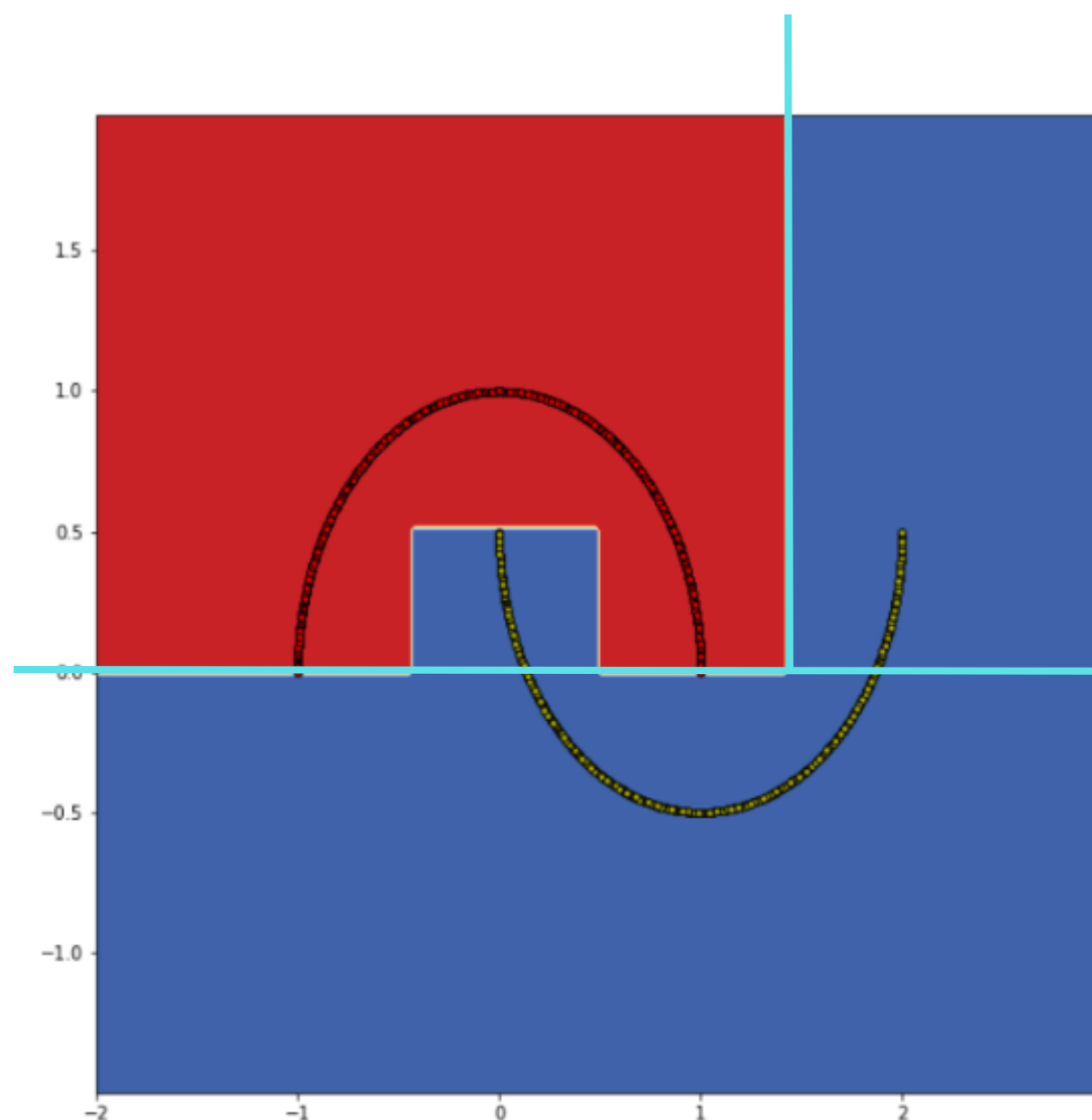
Решающее дерево



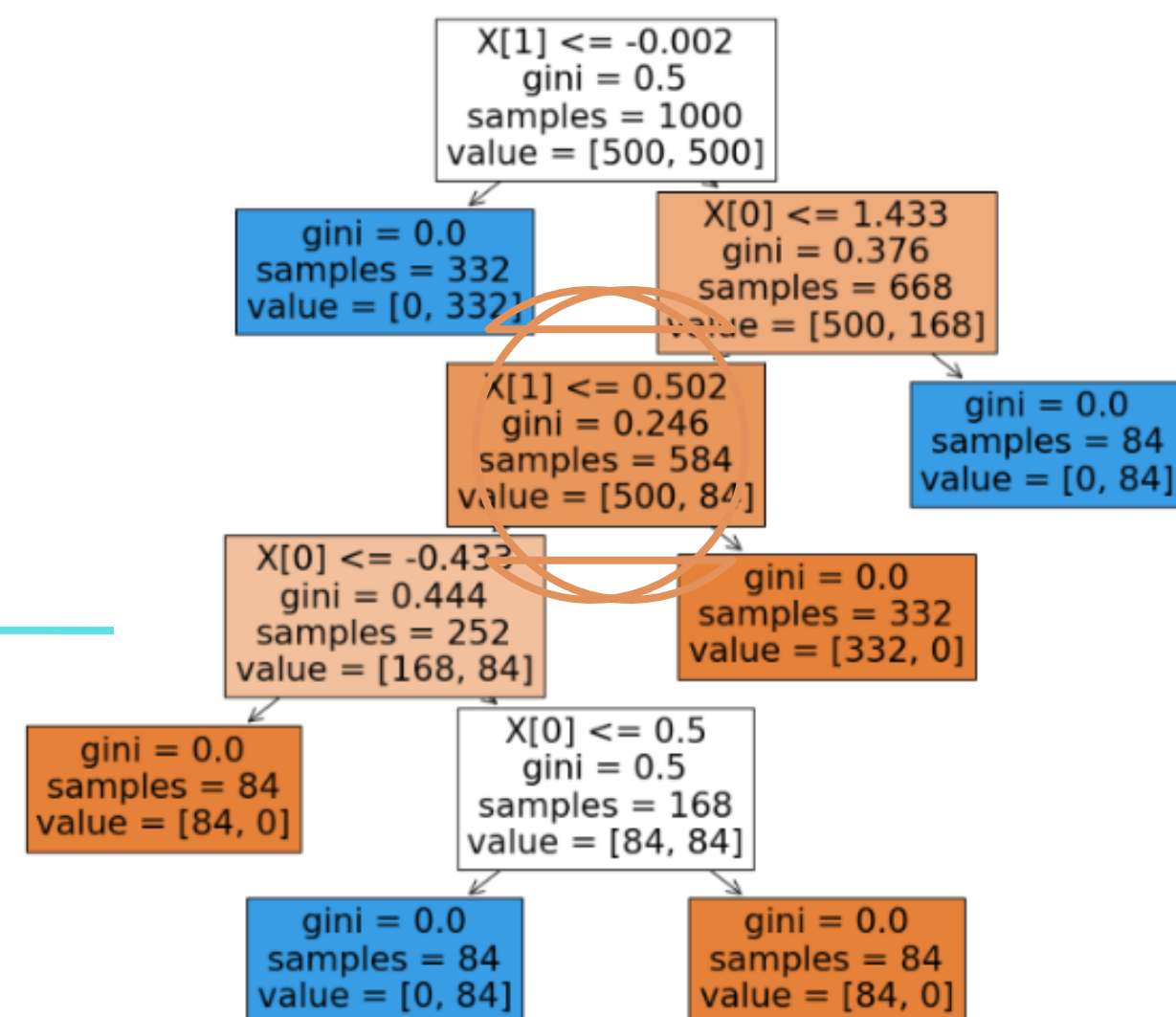
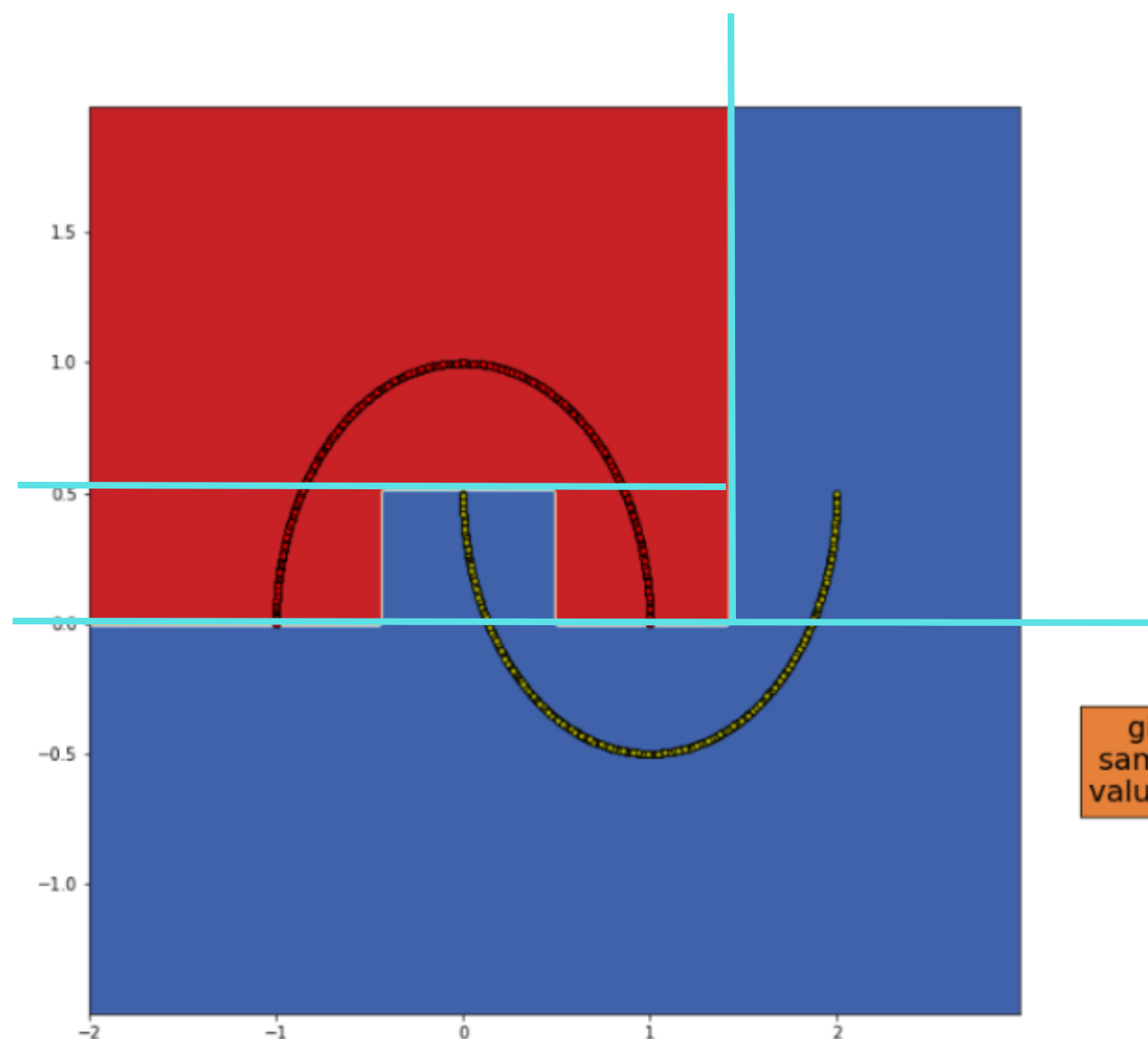
Решающее дерево



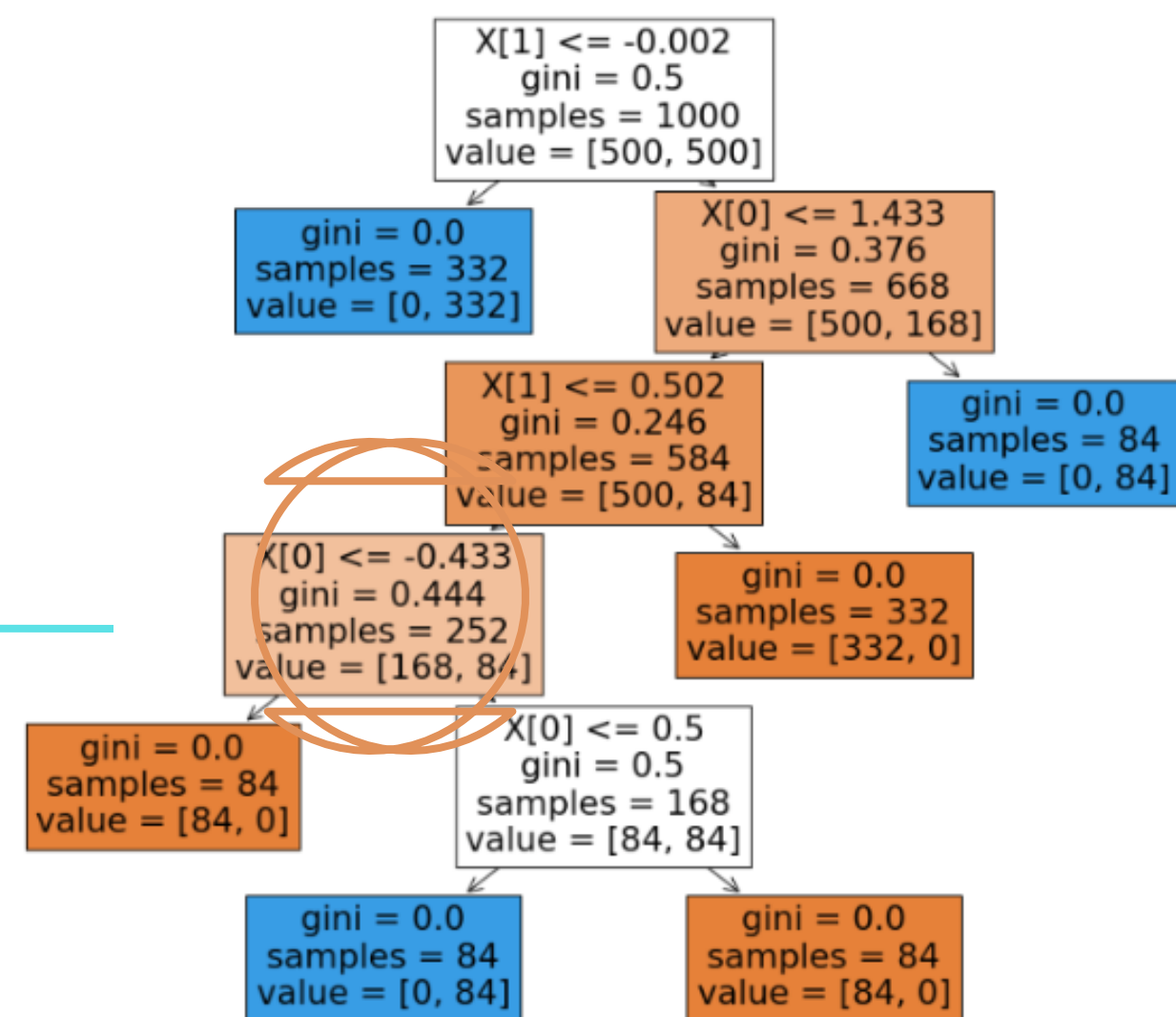
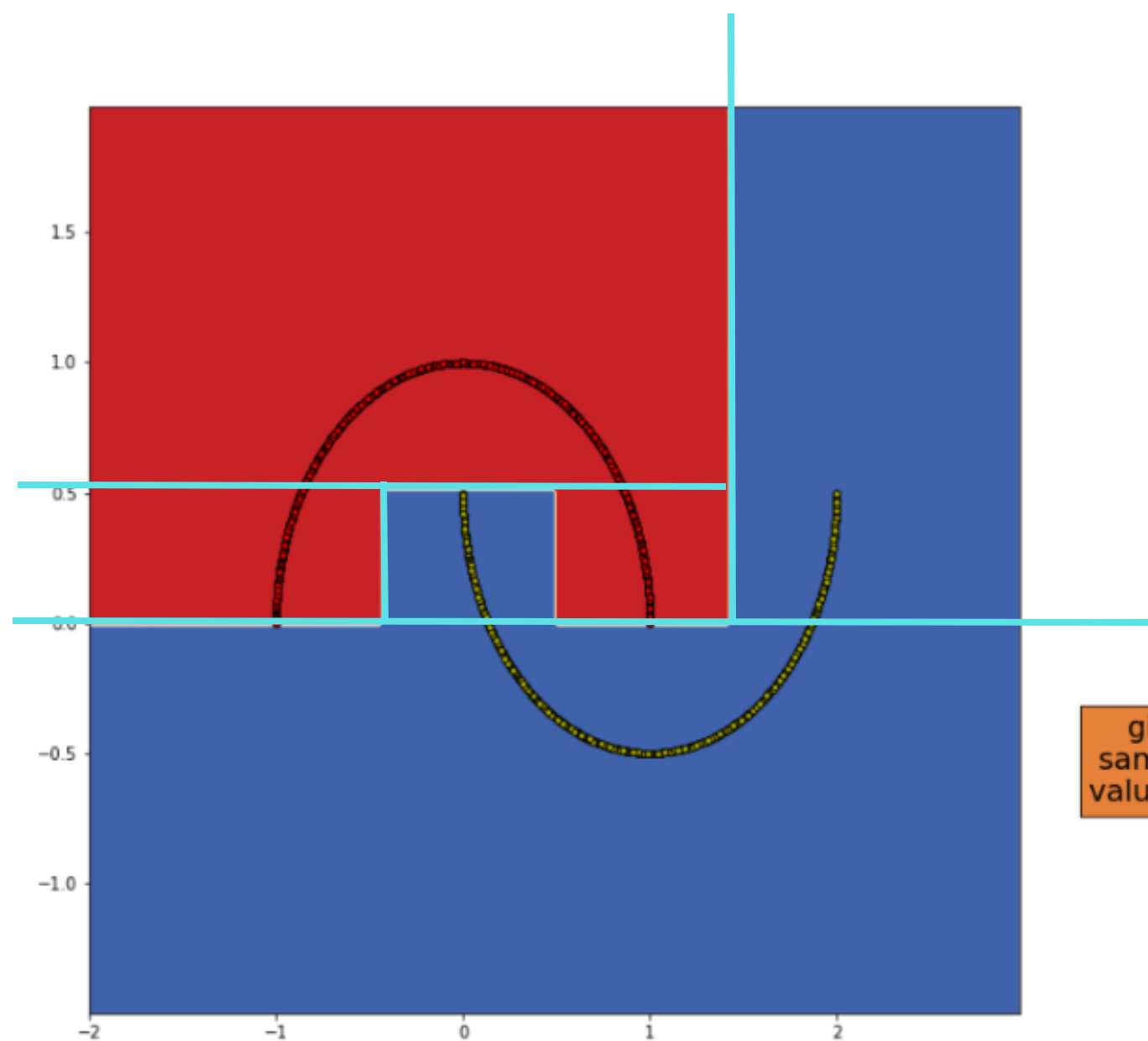
Решающее дерево



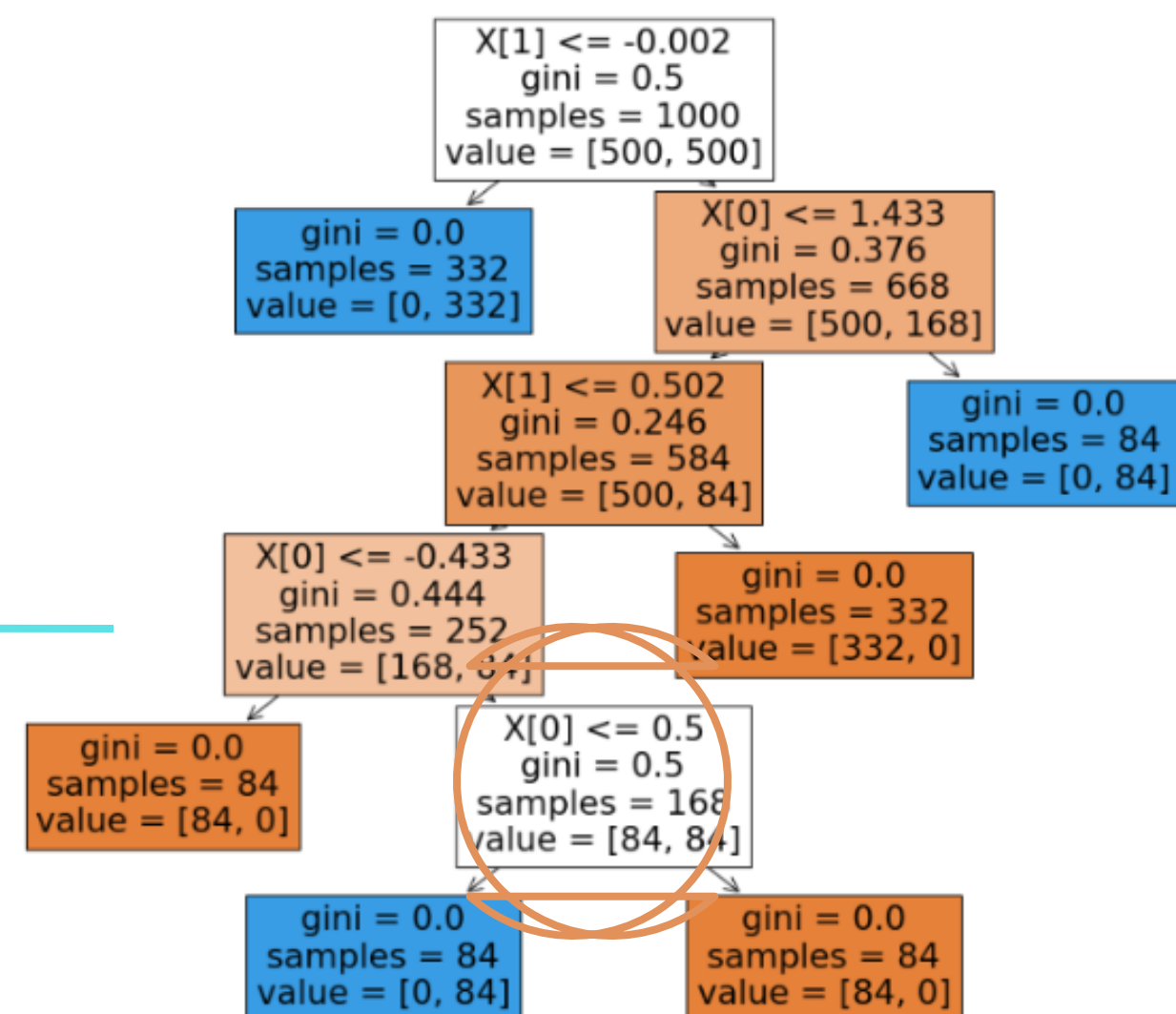
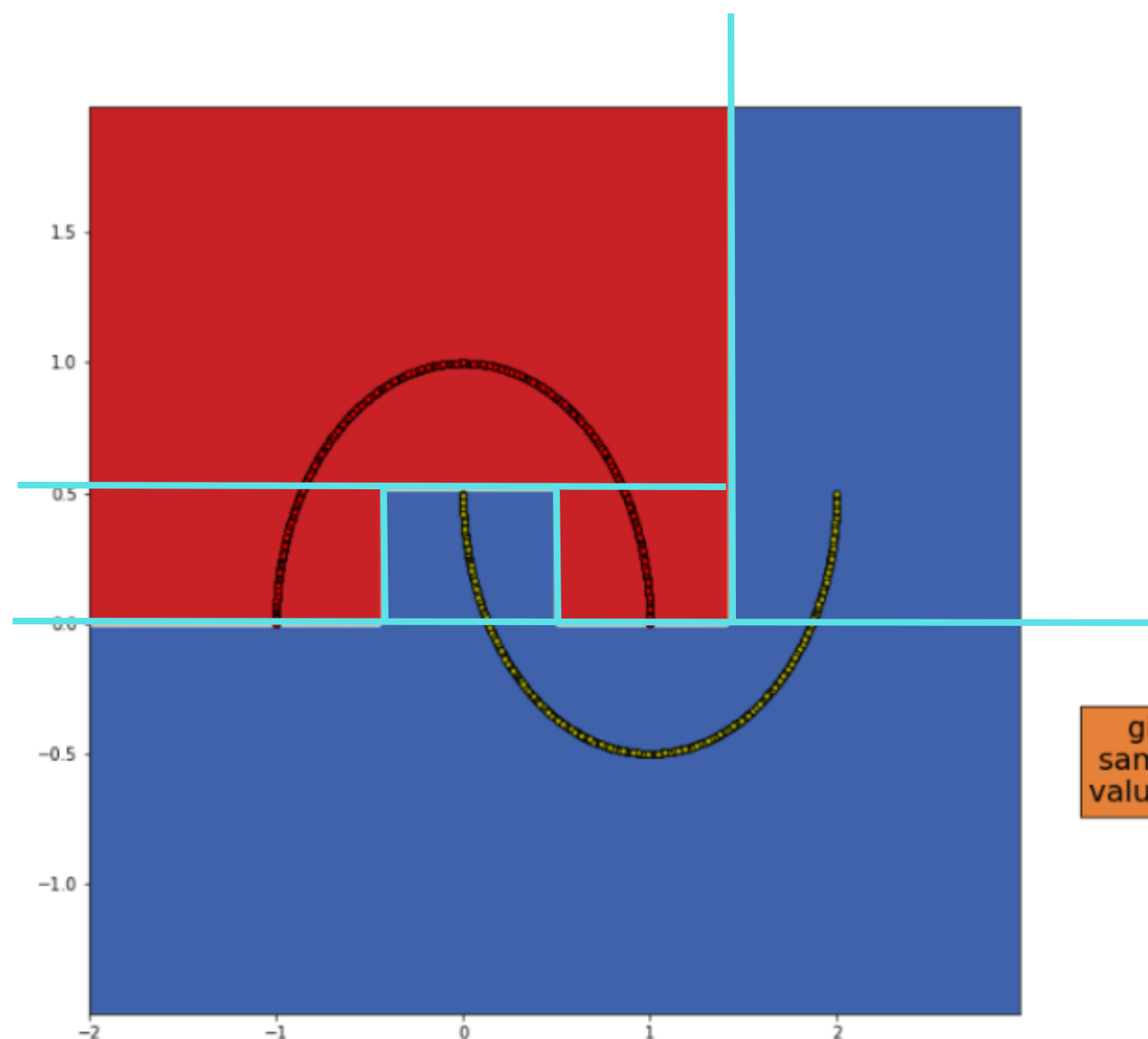
Решающее дерево



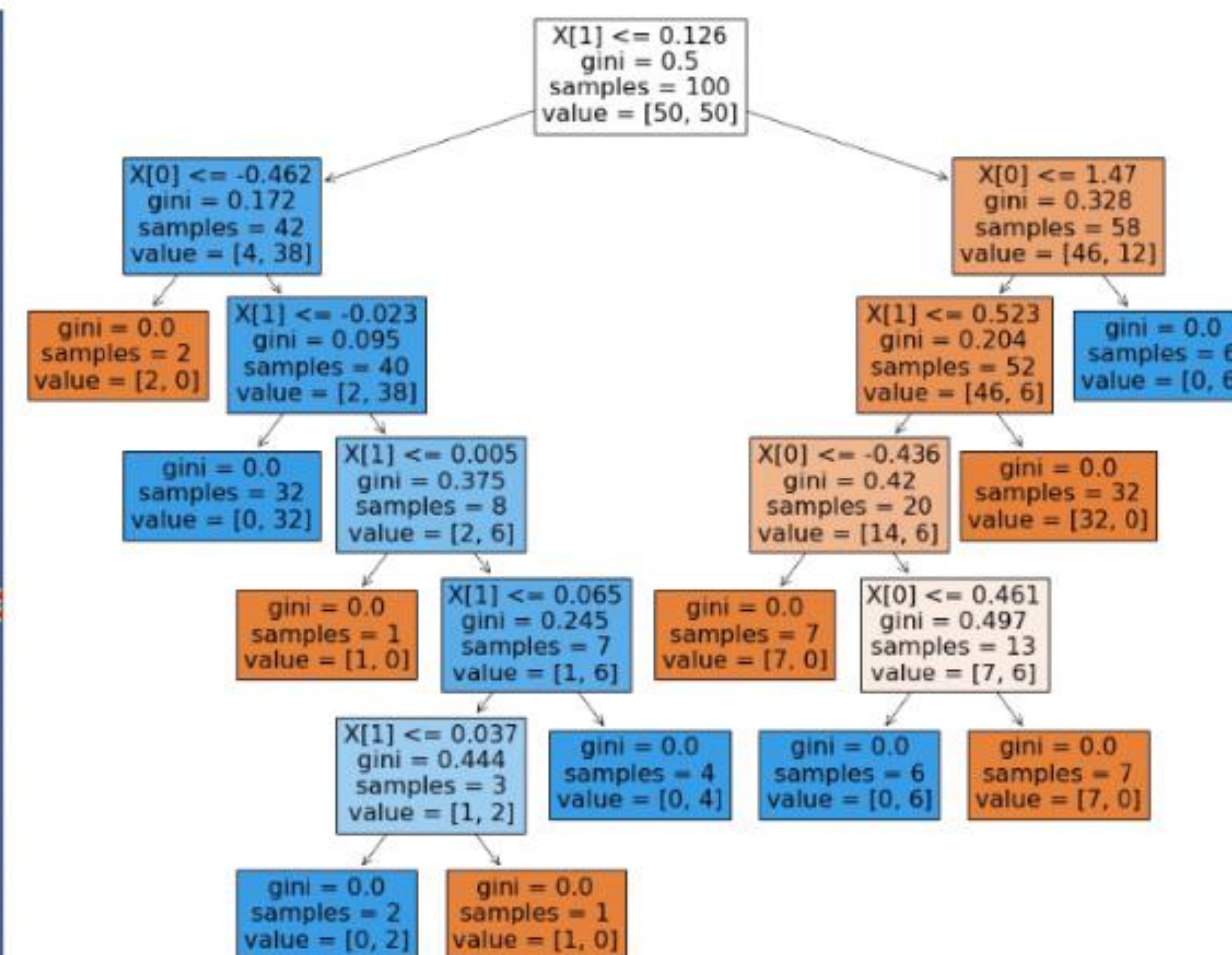
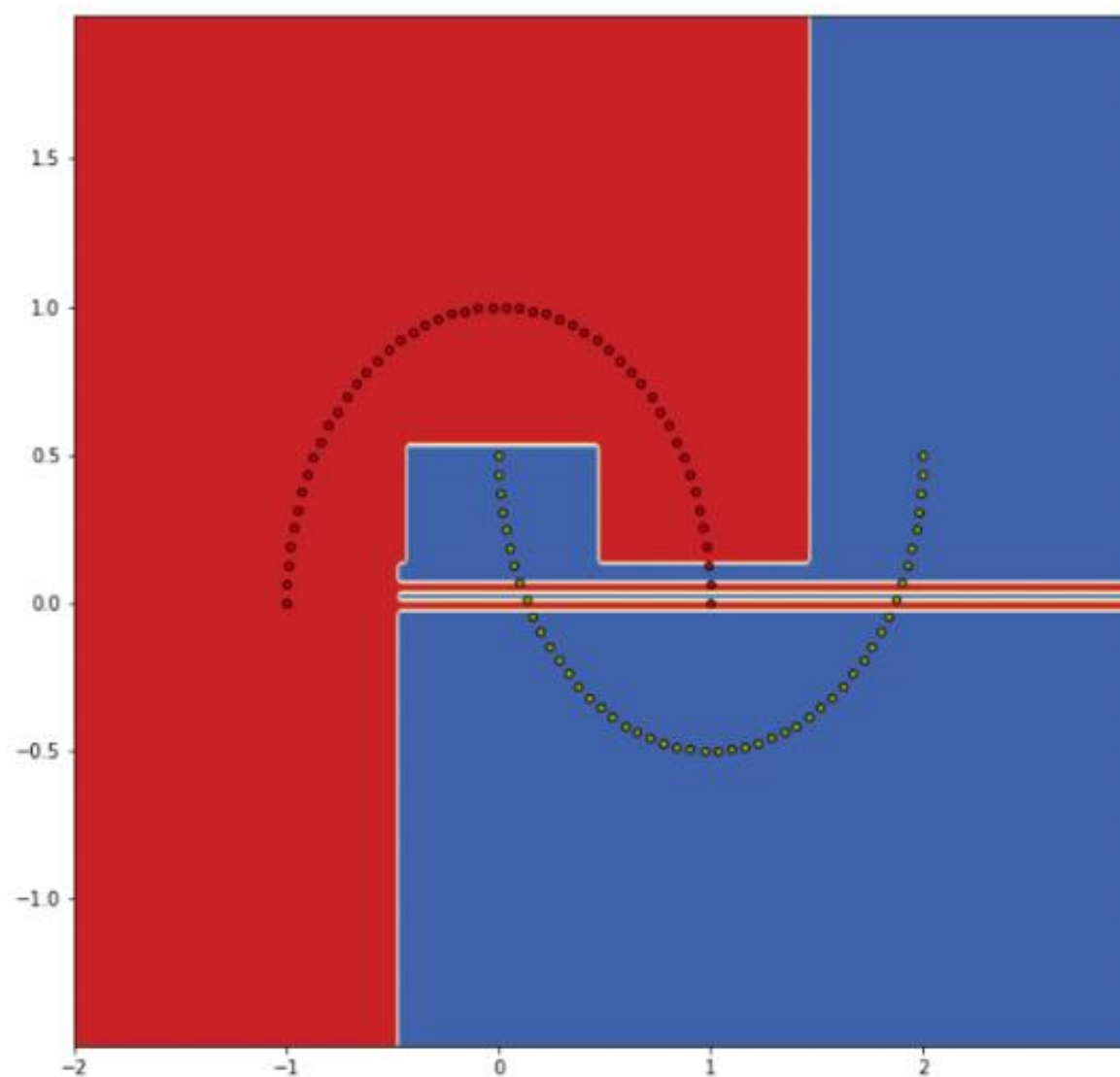
Решающее дерево



Решающее дерево

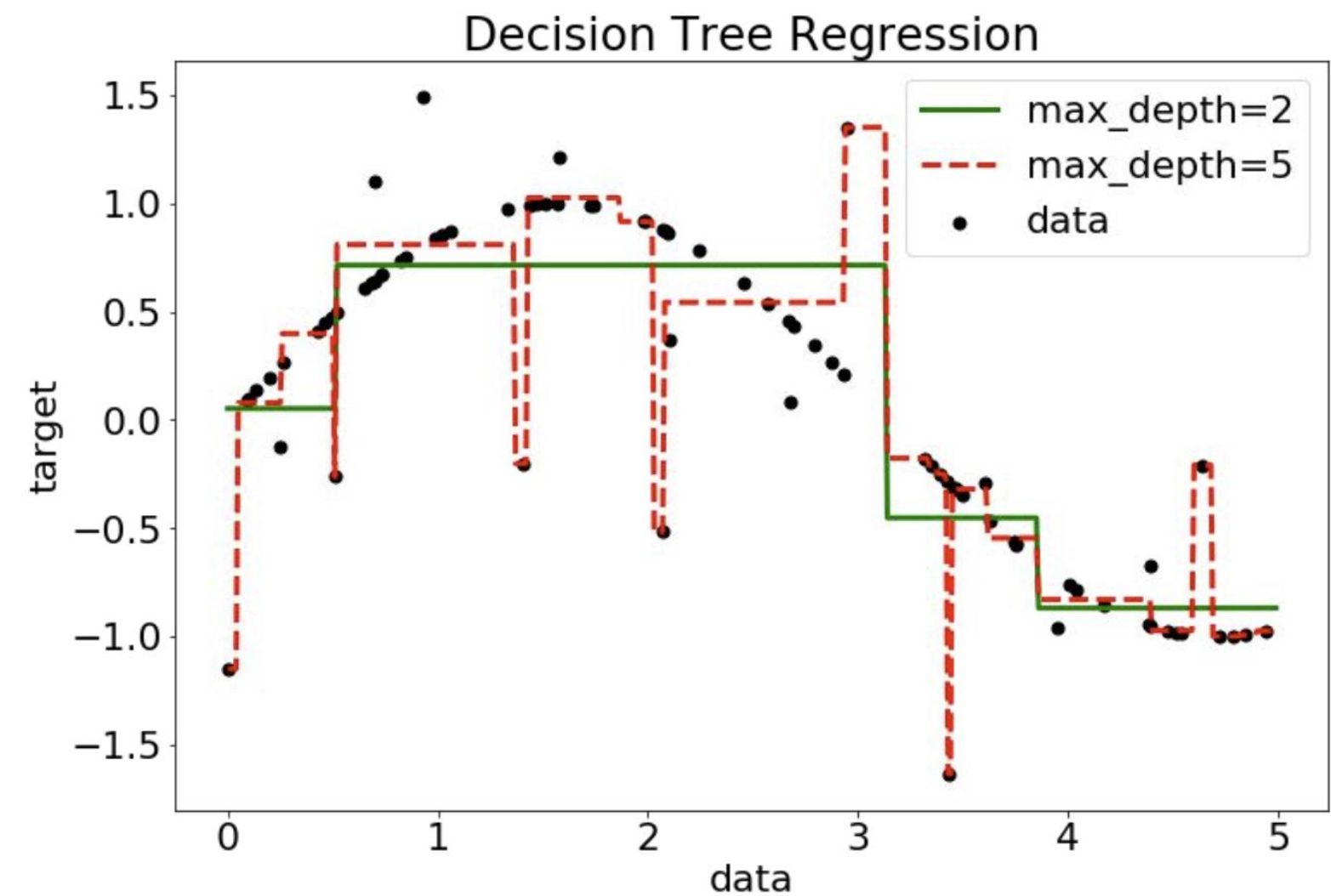


Решающее дерево



Сложность дерева

- Дерево можно строить до тех пор, пока каждый лист не будет соответствовать ровно одному объекту
- Деревом можно идеально разбить любую выборку, если нет объектов с разными признаками и одинаковыми ответами



Предикаты

- Порог на признак $[x_j < t]$ - не единственный вариант
 - Предикат с линейной моделью: $[\langle w, x \rangle < t]$
 - Предикат с метрикой: $[\rho(x, x_0) < t]$
 - ...
-
- Даже с простейшим предикатом можно строить хорошие модели!

Прогнозы в листьях

- Константные прогнозы
- Регрессия:

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

- Классификация:

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

- Классификация и вероятности классов:

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

Прогнозы в листьях

- Константные прогнозы
- Регрессия:

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

- Классификация:

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

- Классификация и вероятности классов:

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

- Можно усложнять листья
- Пример:

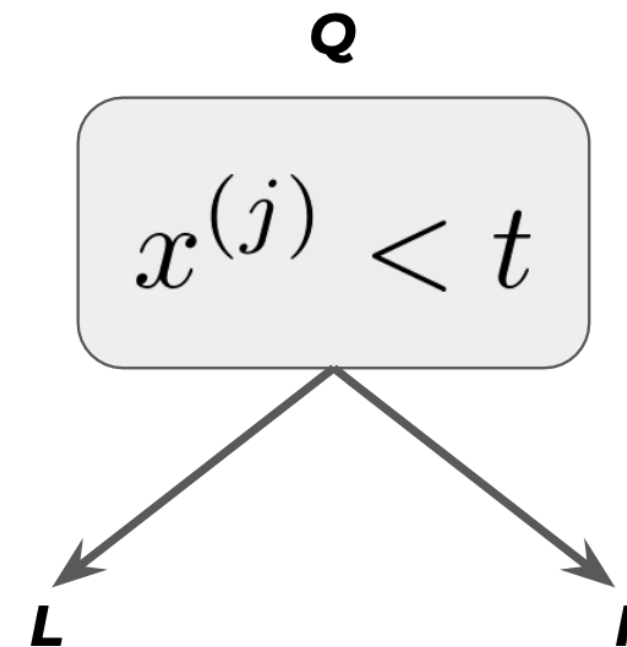
$$c_v(x) = \langle w_v, x \rangle$$

Формула для дерева

- Дерево разбивает признаковое пространство на области R_1, \dots, R_j
- Каждая область R_i соответствует листу
- В области R_i прогноз c_i константный

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

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

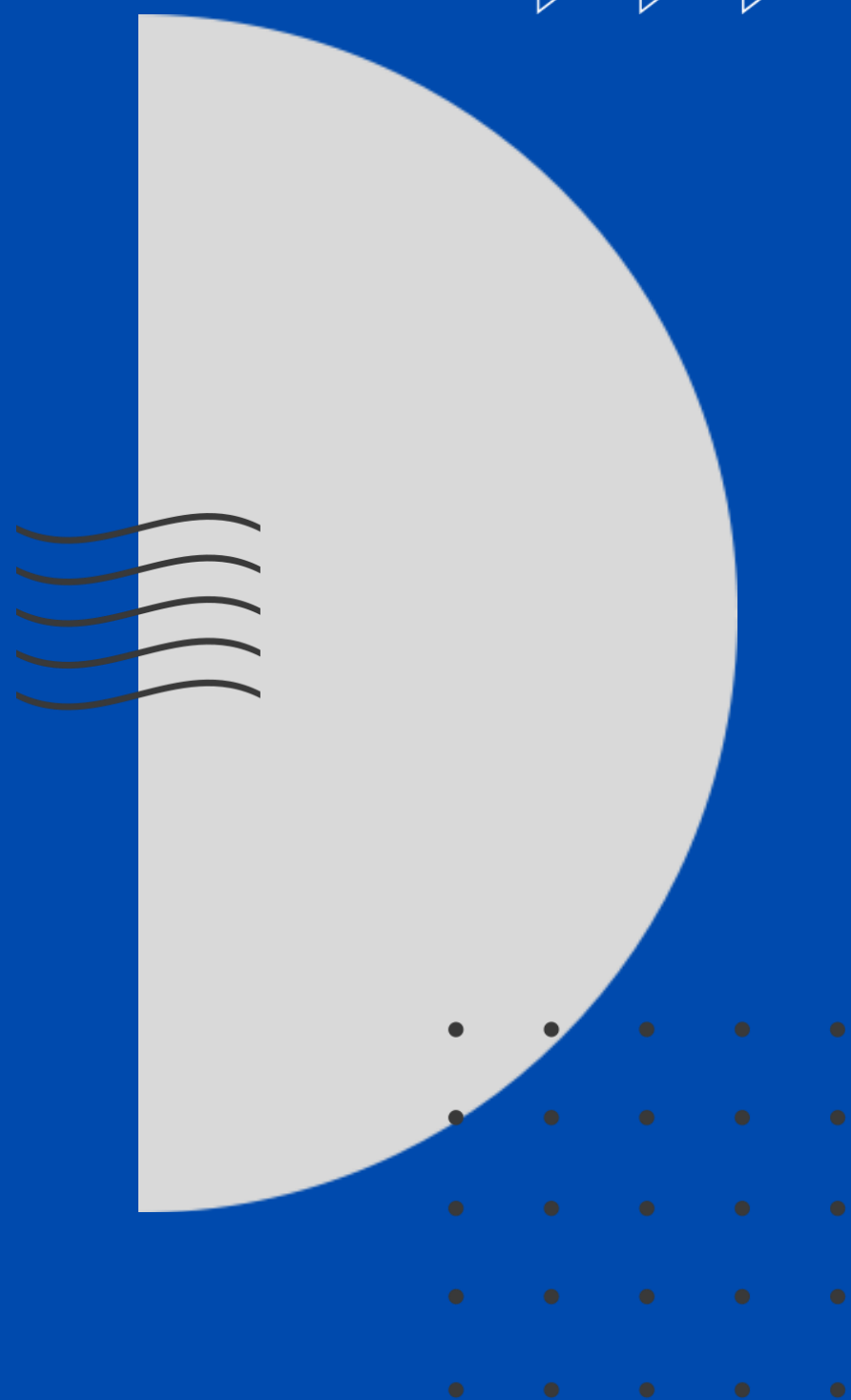


What is H?

$$\frac{|L|}{|Q|} H(L) + \frac{|R|}{|Q|} H(R) \longrightarrow \min_{j,t}$$

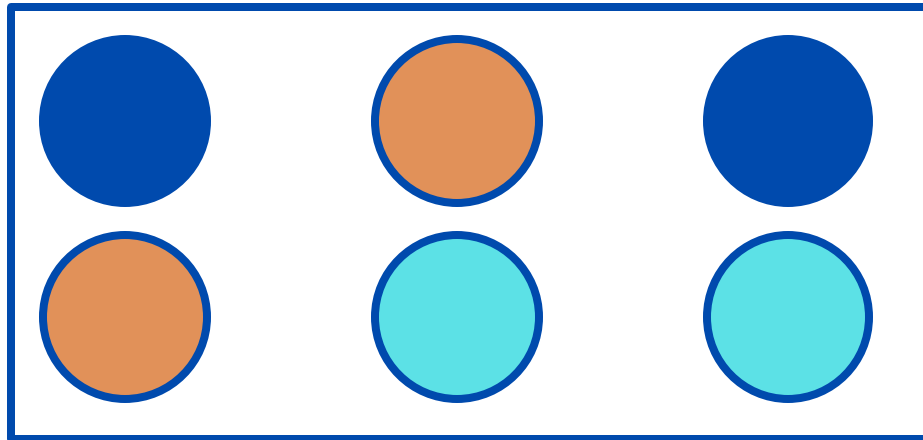


02



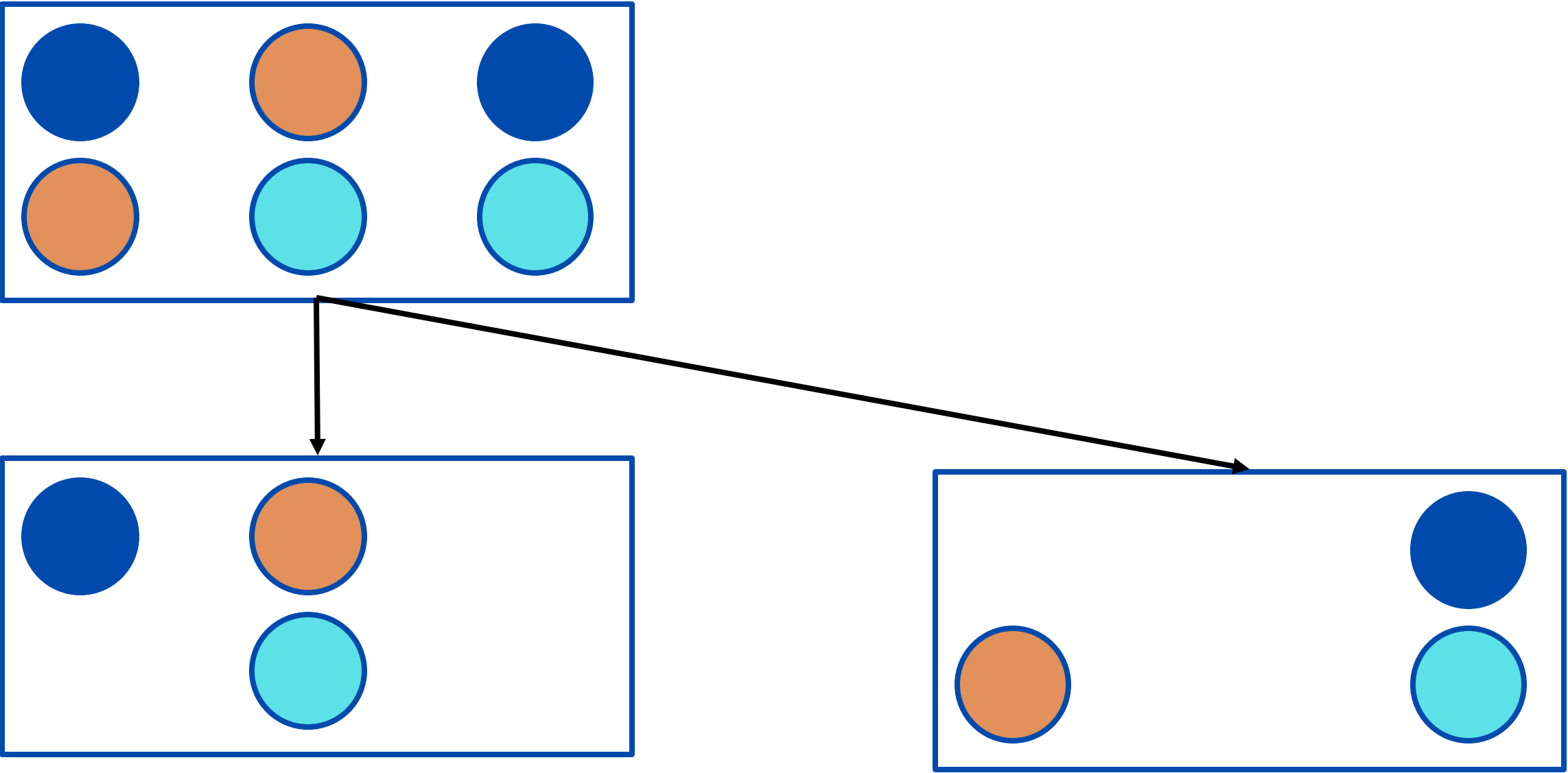
Как выбирать
предикаты?

Жадное построение

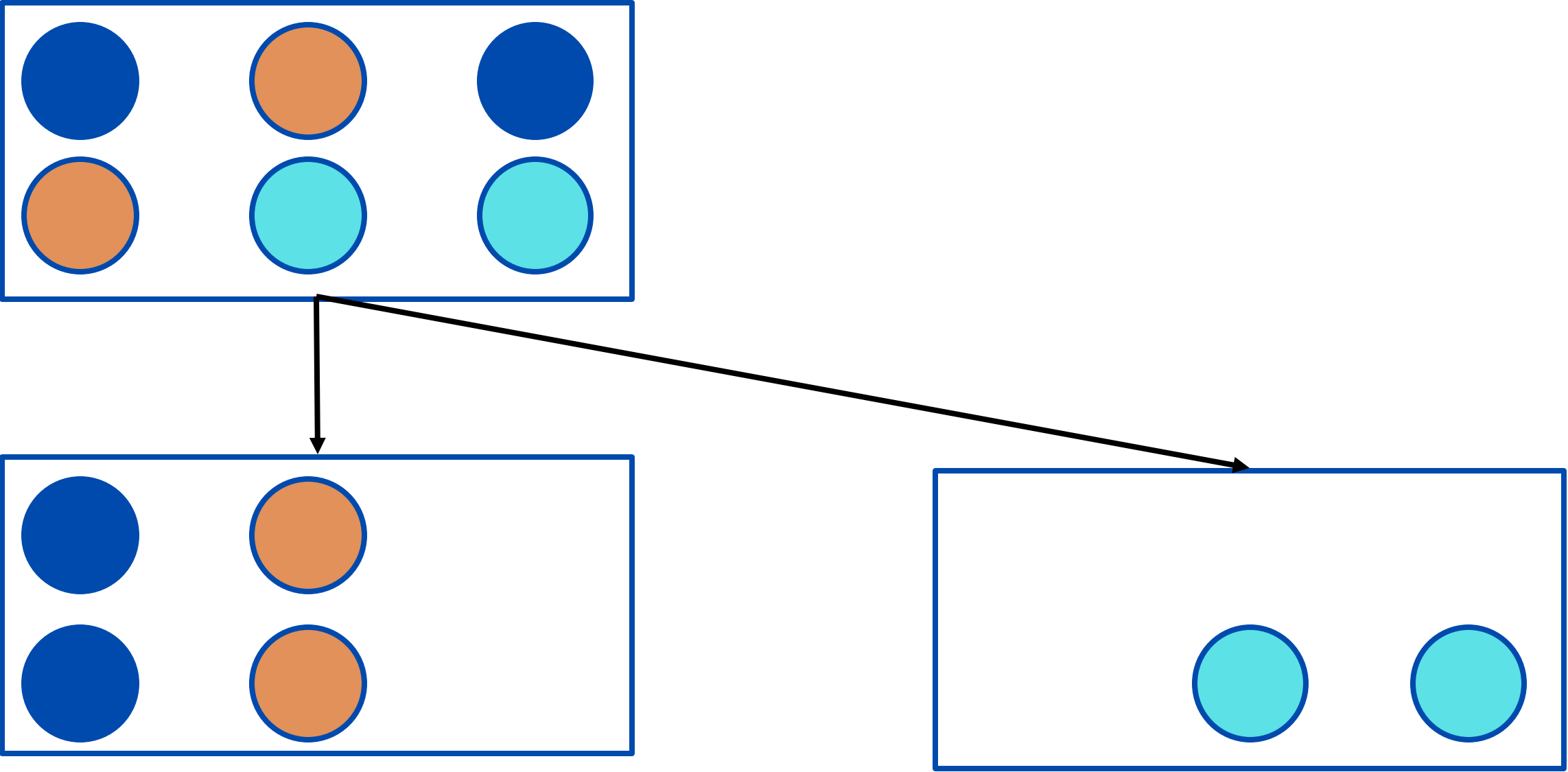


- Как разбить вершину?

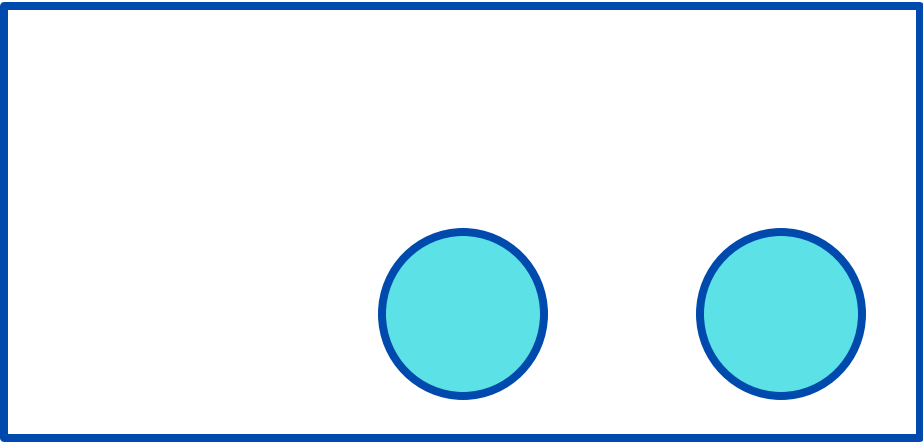
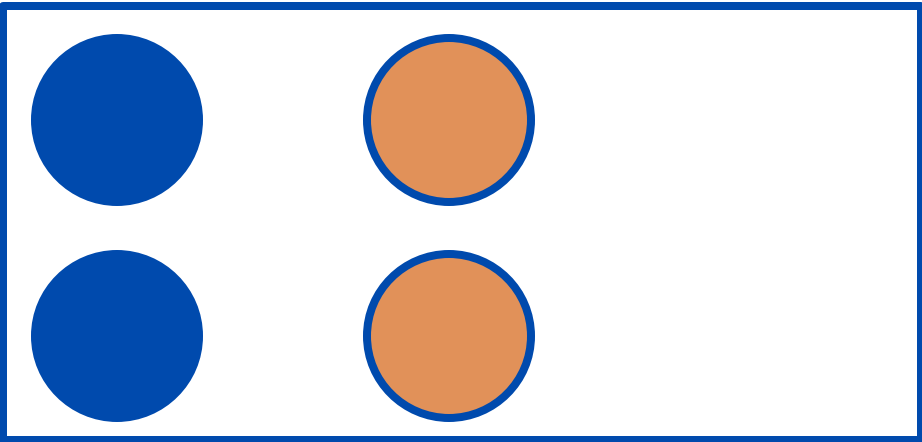
Жадное построение



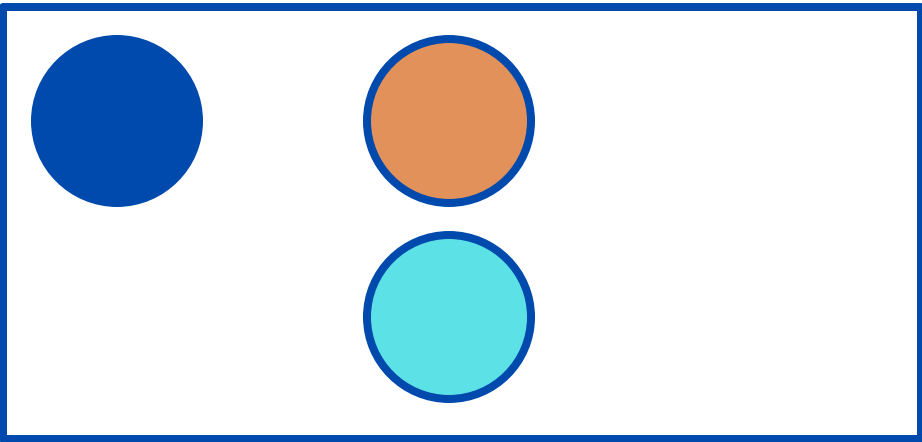
Жадное построение



Как сравнить разбиения?

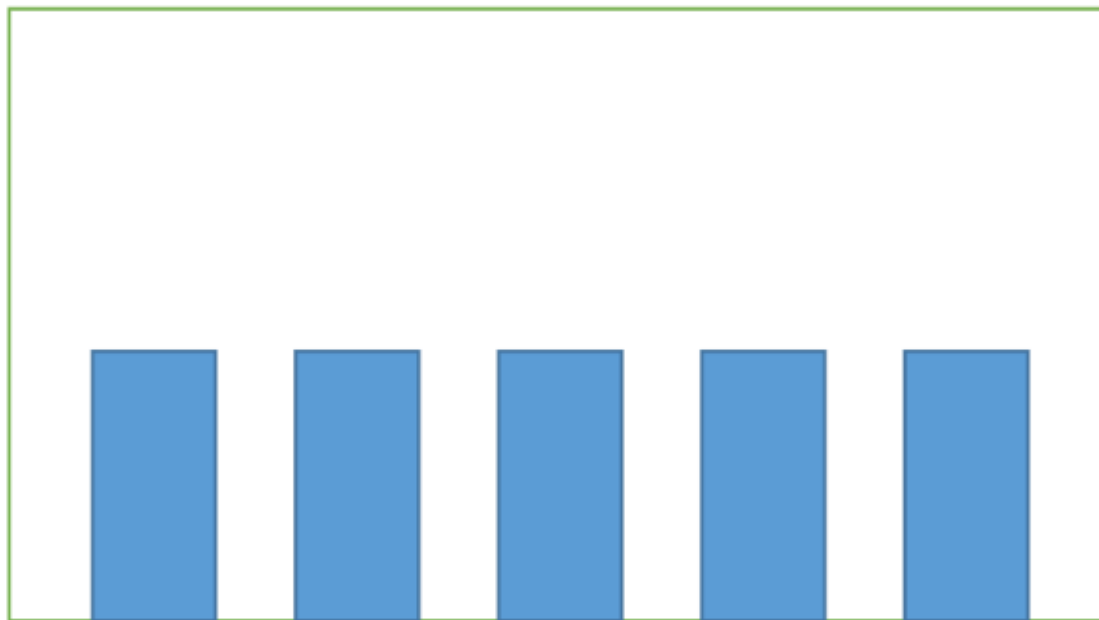


или

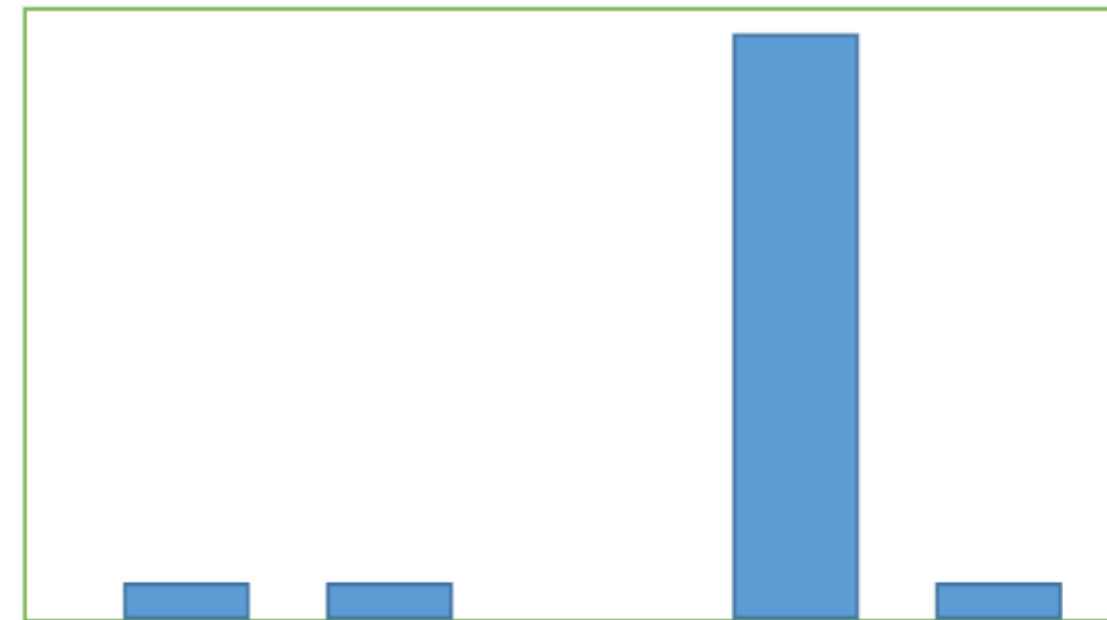


Энтропия

- Мера неопределенности распределения



Высокая энтропия



Низкая энтропия

Энтропия

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

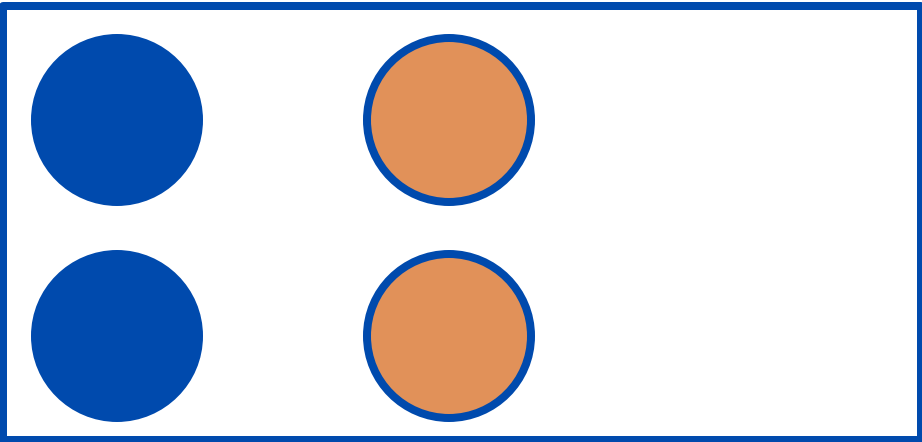
$$H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i$$

- | | | |
|-----------------------------|---------------------------|-------------------|
| • (0.2, 0.2, 0.2, 0.2, 0.2) | • (0.9, 0.05, 0.05, 0, 0) | • (0, 0, 0, 1, 0) |
| • $H = 1.609 \dots$ | • $H = 0.3943 \dots$ | • $H = 0$ |

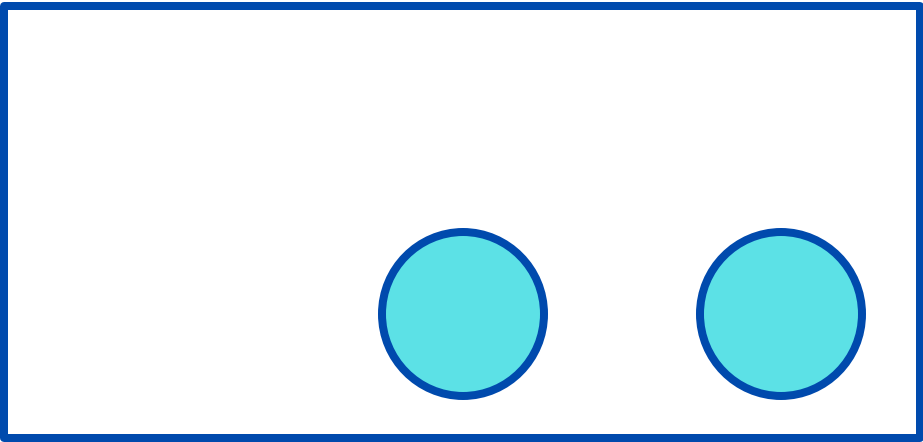
Энтропия для бинарной классификации

$$H(R) = -p_0 \log p_0 - p_1 \log p_1$$

Энтропия

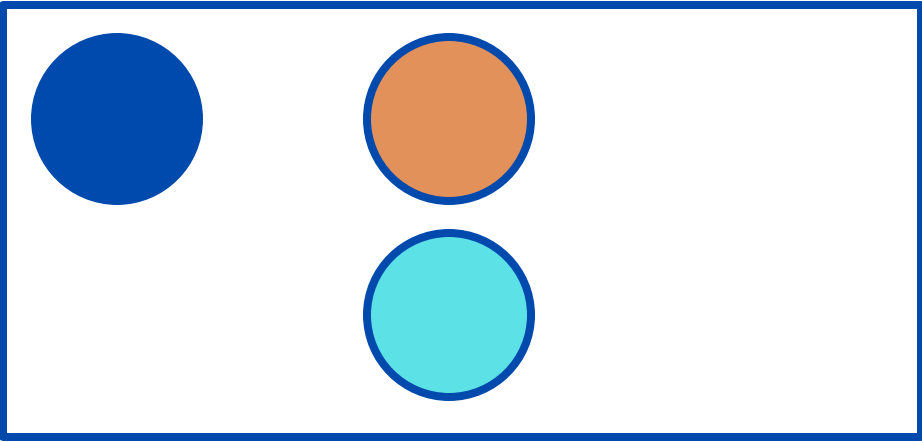


0.693



0

$H = 0.693$



1.09



1.09

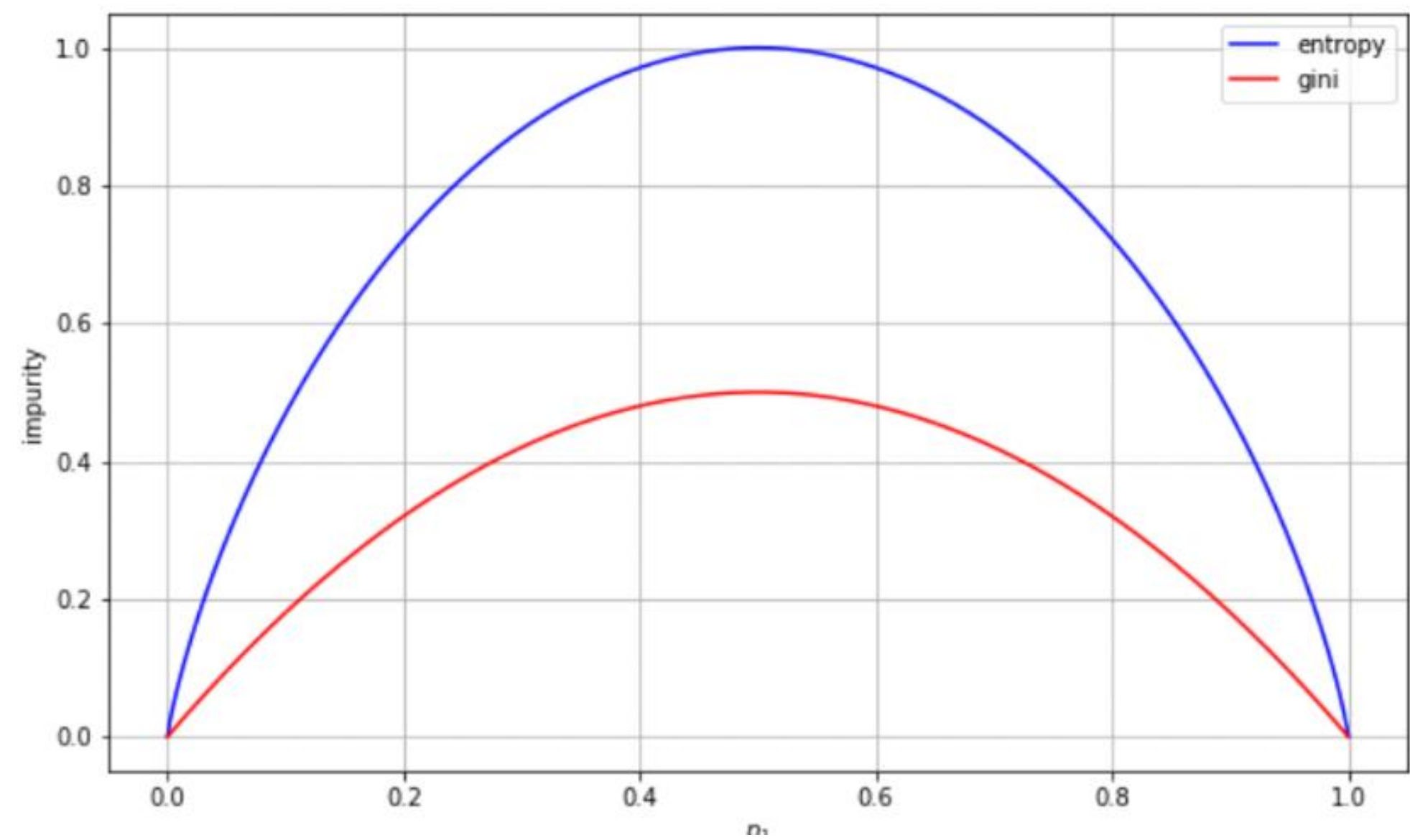
$H = 2.18$

Критерий Джини

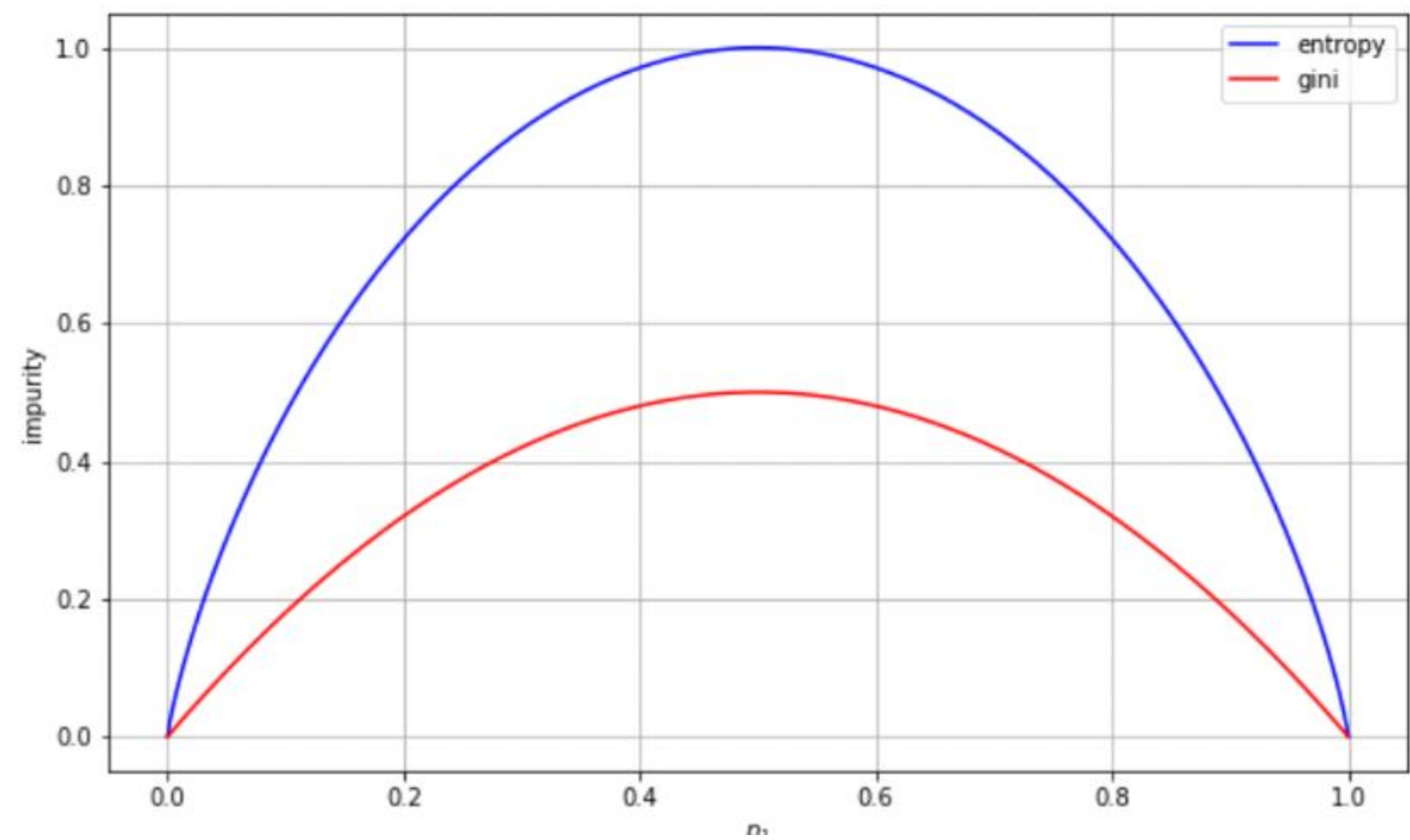
- Вероятность ошибки случайного классификатора, который выдаёт класс k с вероятностью p
- Примерно пропорционально количеству пар объектов, относящихся к разным классам
- Максимизация Неопределенности Джини = максимизация числа пар объектов одного класса, оказавшихся в одном поддереве

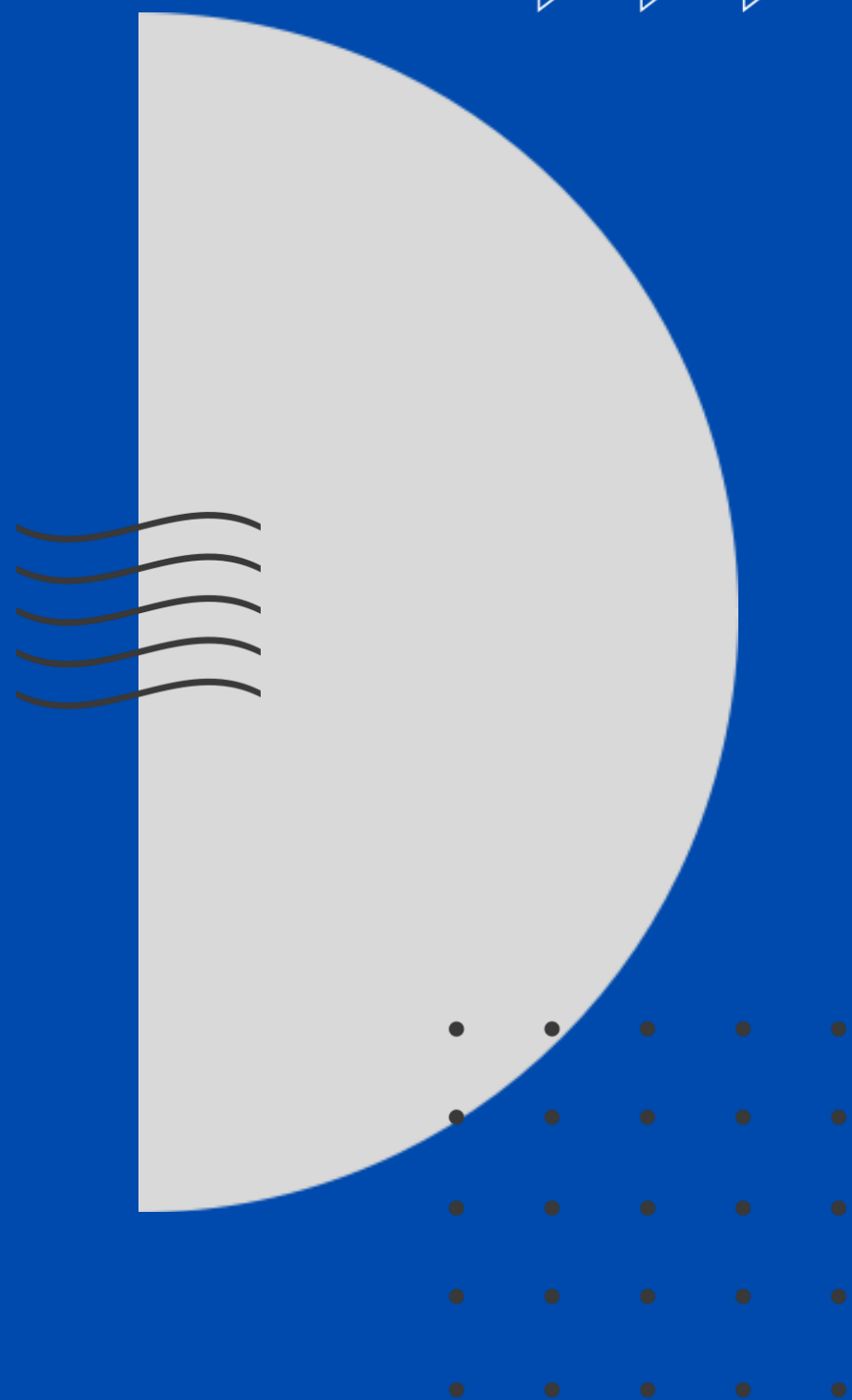
$$I_G(p) = \sum_{i=1}^J \left(p_i \sum_{k \neq i} p_k \right) = \sum_{i=1}^J p_i (1 - p_i) = \sum_{i=1}^J (p_i - p_i^2) = \sum_{i=1}^J p_i - \sum_{i=1}^J p_i^2 = 1 - \sum_{i=1}^J p_i^2.$$

Критерий качества вершины



Критерий качества вершины





03

Композиции моделей

Композиция

Аналогично с моделями машинного обучения:

- Если мы возьмем N базовых алгоритмов $b_1(x), \dots, b_N(x)$
- То композиция будет выглядеть следующим образом

$$a(x) = \arg \max_{y \in Y} \sum_{n=1}^N [b_n(x) = y]$$

Композиция

Аналогично с моделями машинного обучения:

- Если мы возьмем N базовых алгоритмов $b_1(x), \dots, b_N(x)$
- То композиция будет выглядеть следующим образом

$$a(x) = \arg \max_{y \in Y} \sum_{n=1}^N [b_n(x) = y]$$

- Либо же для регрессии

$$a(x) = \frac{1}{N} \sum_{n=1}^N (b_n(x))$$

Композиция

- Если каждая базовая модель хоть немного лучше угадывания, то рост количества моделей устремляет ответ к истинному

Композиция

- Если каждая базовая модель хоть немного лучше угадывания, то рост количества моделей устремляет ответ к истинному
- Теорема о “жюри присяжных”:

N — количество присяжных

p — вероятность правильного решения присяжного

μ — вероятность правильного решения всего жюри

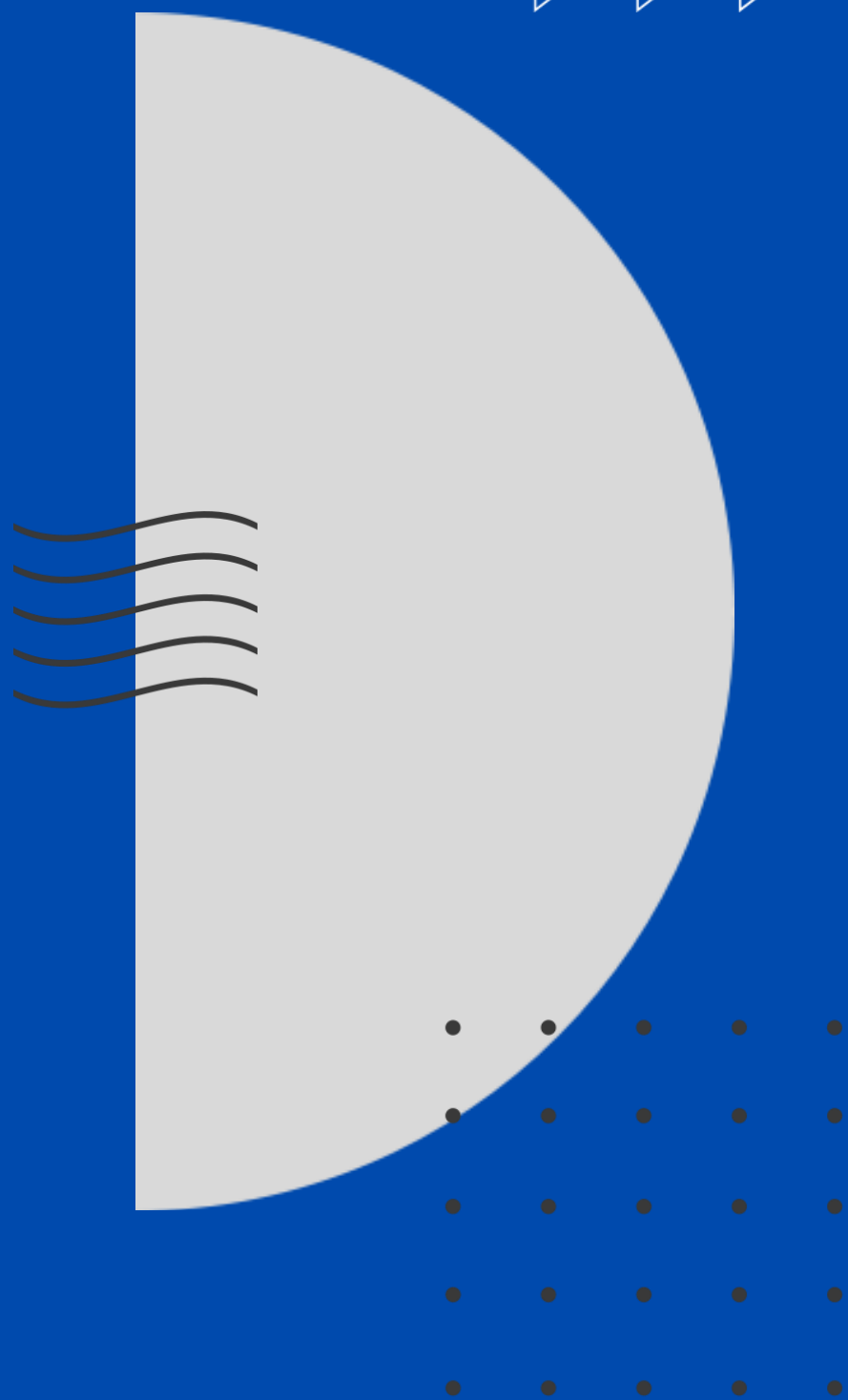
m — минимальное большинство членов жюри, $m = \text{floor}(N/2) + 1$

C_N^i — число сочетаний из N по i

$$\mu = \sum_{i=m}^N C_N^i p^i (1-p)^{N-i}$$

Если $p > 0.5$, то $\mu > p$

Если $N \rightarrow \infty$, то $\mu \rightarrow 1$



04

БЭГГИНГ

Бэггинг

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Как на одной выборке построить N различных моделей?

БЭГГИНГ

- $b_1(x), \dots, b_N(x)$ — базовые модели
- Как на одной выборке построить N различных моделей?

Обучим их независимо на разных подвыборках

БЭГГИНГ

- Bagging (bootstrap aggregating)
- Базовые модели обучаются независимо
- Подмножество выбирается с помощью бутстрапа

$$\begin{aligned}\varepsilon_j(x) &= b_j(x) - y(x), \quad j = 1, \dots, N, \\ \mathbb{E}_x(b_j(x) - y(x))^2 &= \mathbb{E}_x \varepsilon_j^2(x). \\ E_1 &= \frac{1}{N} \sum_{j=1}^N \mathbb{E}_x \varepsilon_j^2(x). \\ E_N &= \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^N b_j(x) - y(x) \right)^2 = \\ &= \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^N \varepsilon_j(x) \right)^2 = \\ &= \frac{1}{N^2} \mathbb{E}_x \left(\sum_{j=1}^N \varepsilon_j^2(x) + \underbrace{\sum_{i \neq j} \varepsilon_i(x) \varepsilon_j(x)}_{=0} \right) = \\ &= \frac{1}{N} E_1.\end{aligned}$$

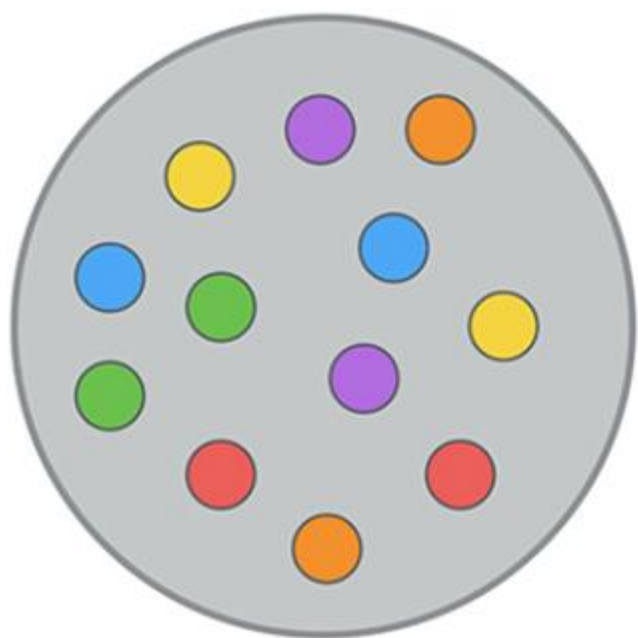
Бутстрап

- Выборка с возвращением
- Берём ℓ элементов из X
- Пример: $\{x_1, x_2, x_3, x_4\} \rightarrow \{x_1, x_2, x_2, x_4\}$
- В подвыборке будет ℓ объектов, из них около 63.2% уникальных
- Если объект входит в выборку несколько раз, то мы как бы повышаем его вес

Бутстрап

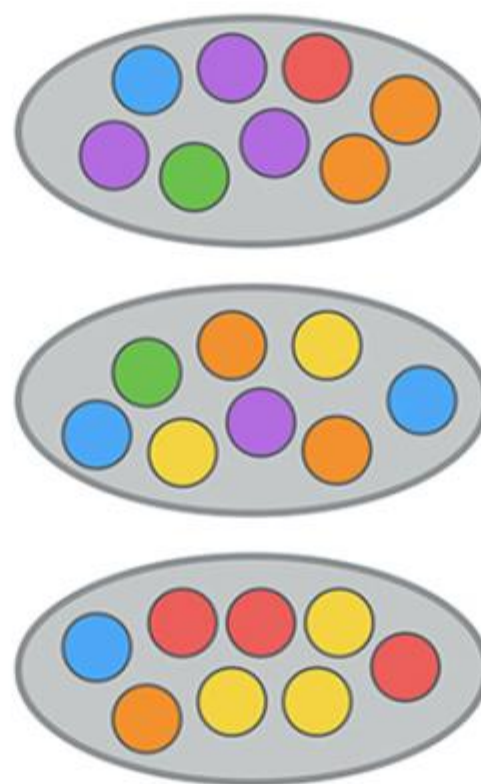


Исходная выборка



Статистика по
выборке

Бутстрэп выборки



Статистика 1



Статистика 2



Статистика 3



Бутстрэп
распределение



Случайные подпространства

- Выбираем случайное подмножество признаков
- Обучаем модель только на них

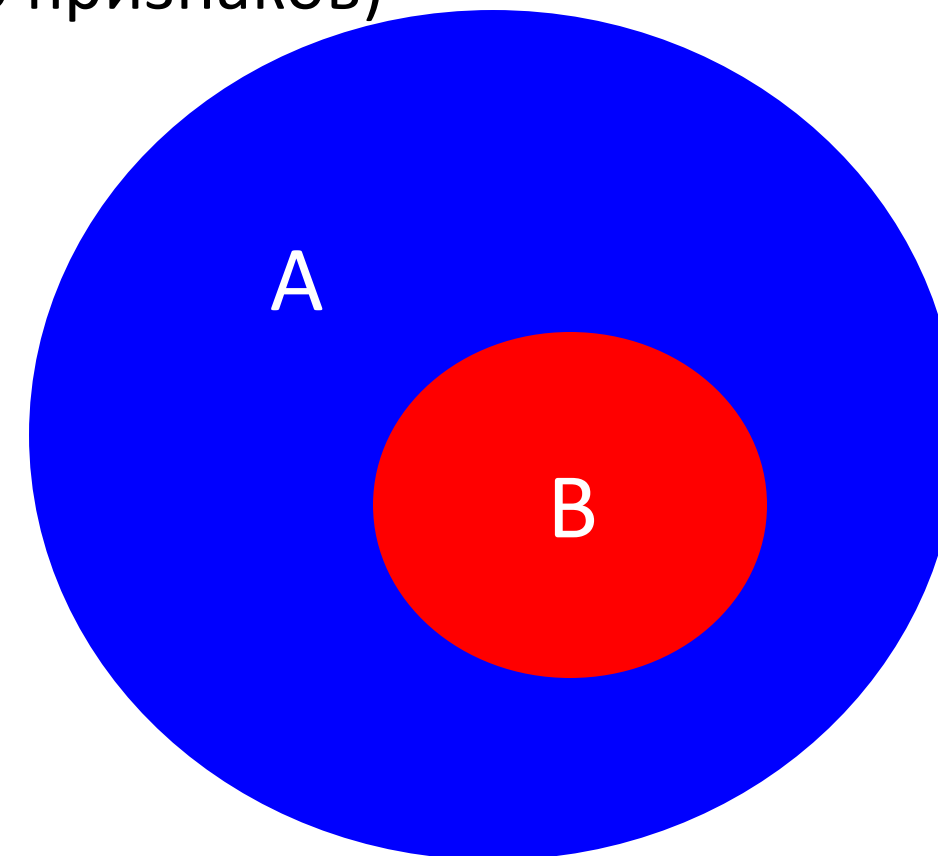
Случайные подпространства

- Выбираем случайное подмножество признаков
- Обучаем модель только на них
- Может быть плохо, если в подмножество не попадут важные признаки



Виды рандомизации

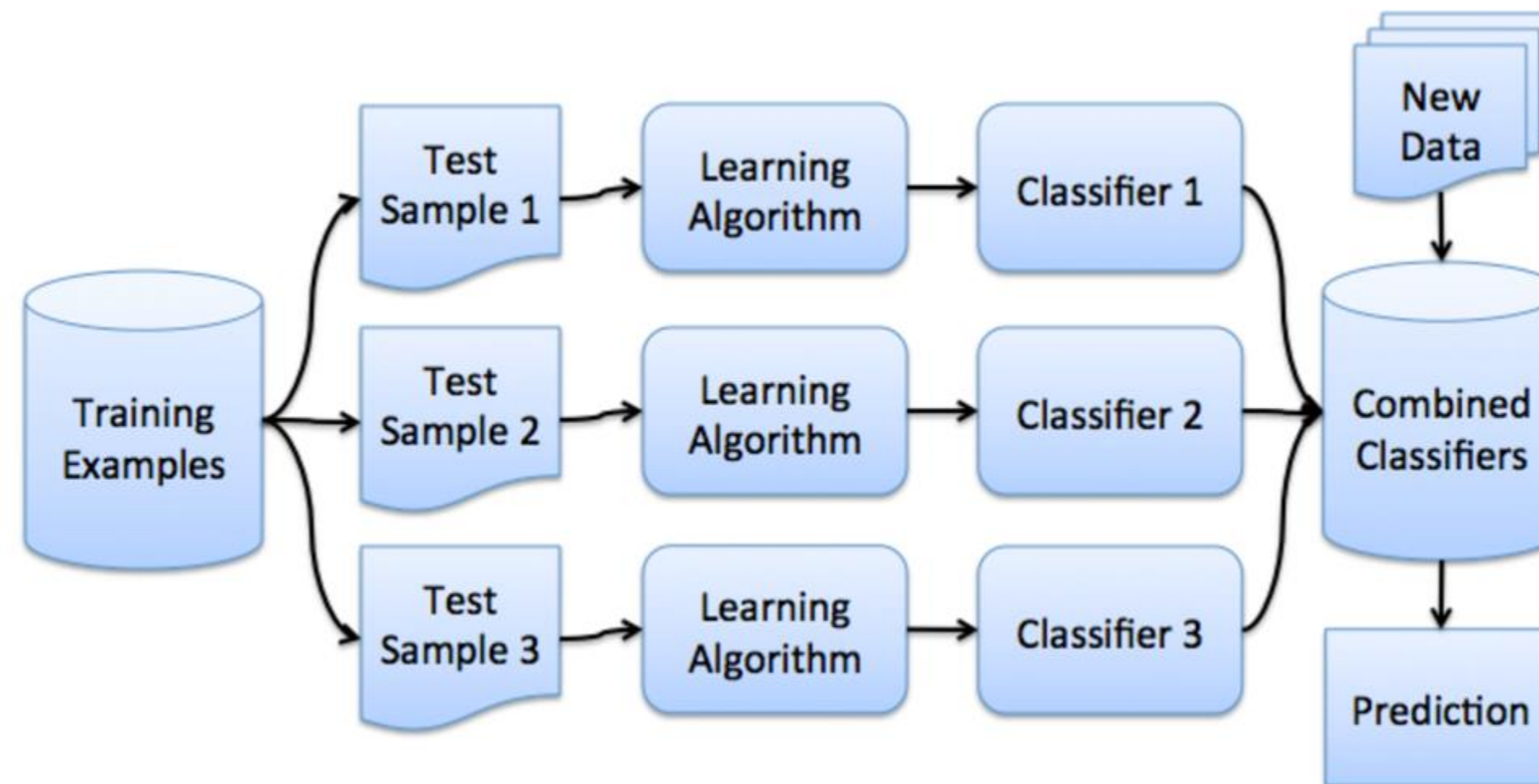
- Бэггинг (случайные подвыборки)
- Случайные подпространства (случайное множество признаков)

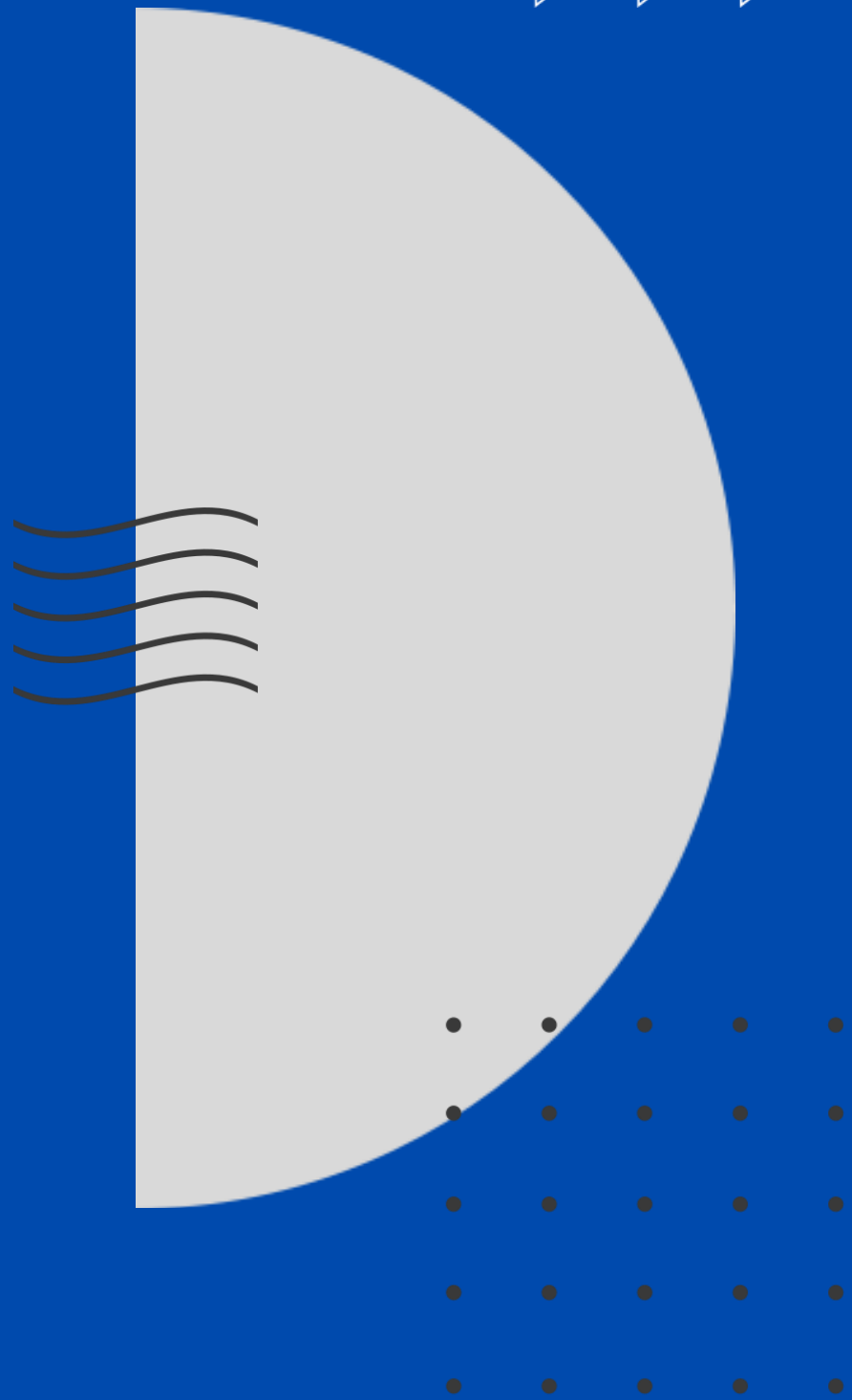




Резюме

- Будем объединять модели в композиции через усреднение или голосование большинством
- Бэггинг - композиция моделей, обученных независимо на случайных подмножествах объектов
- Рандомизируем признаки, на которых обучаются модели





05

Разложение на смещение
и разброс

Bias-variance decomposition

- Разберем на уровне идеи
- Ошибка модели складывается из трех компонент:

Bias-variance decomposition

- Разберем на уровне идеи
- Ошибка модели складывается из трех компонент:
- Шум (noise) - характеристика сложности и противоречивости данных

Bias-variance decomposition

- Разберем на уровне идеи
- Ошибка модели складывается из трех компонент:
- Шум (noise) - характеристика сложности и противоречивости данных
- Смещение (bias) - способность модели приблизить лучшую среди всех возможных моделей

Bias-variance decomposition

- Разберем на уровне идеи
- Ошибка модели складывается из трех компонент:
- Шум (noise) - характеристика сложности и противоречивости данных
- Смещение (bias) - способность модели приблизить лучшую среди всех возможных моделей
- Разброс (variance) - устойчивость модели к изменениям в обучающей выборке

Bias-variance decomposition

$$L(\mu) = \underbrace{\mathbb{E}_{x,y} \left[(y - \mathbb{E}[y | x])^2 \right]}_{\text{шум}} + \underbrace{\mathbb{E}_x \left[(\mathbb{E}_X [\mu(X)] - \mathbb{E}[y | x])^2 \right]}_{\text{смещение}} + \underbrace{\mathbb{E}_x \left[\mathbb{E}_X \left[(\mu(X) - \mathbb{E}_X [\mu(X)])^2 \right] \right]}_{\text{разброс}}$$

$$\text{bias} := \mathbb{E}(\hat{y}) - y.$$

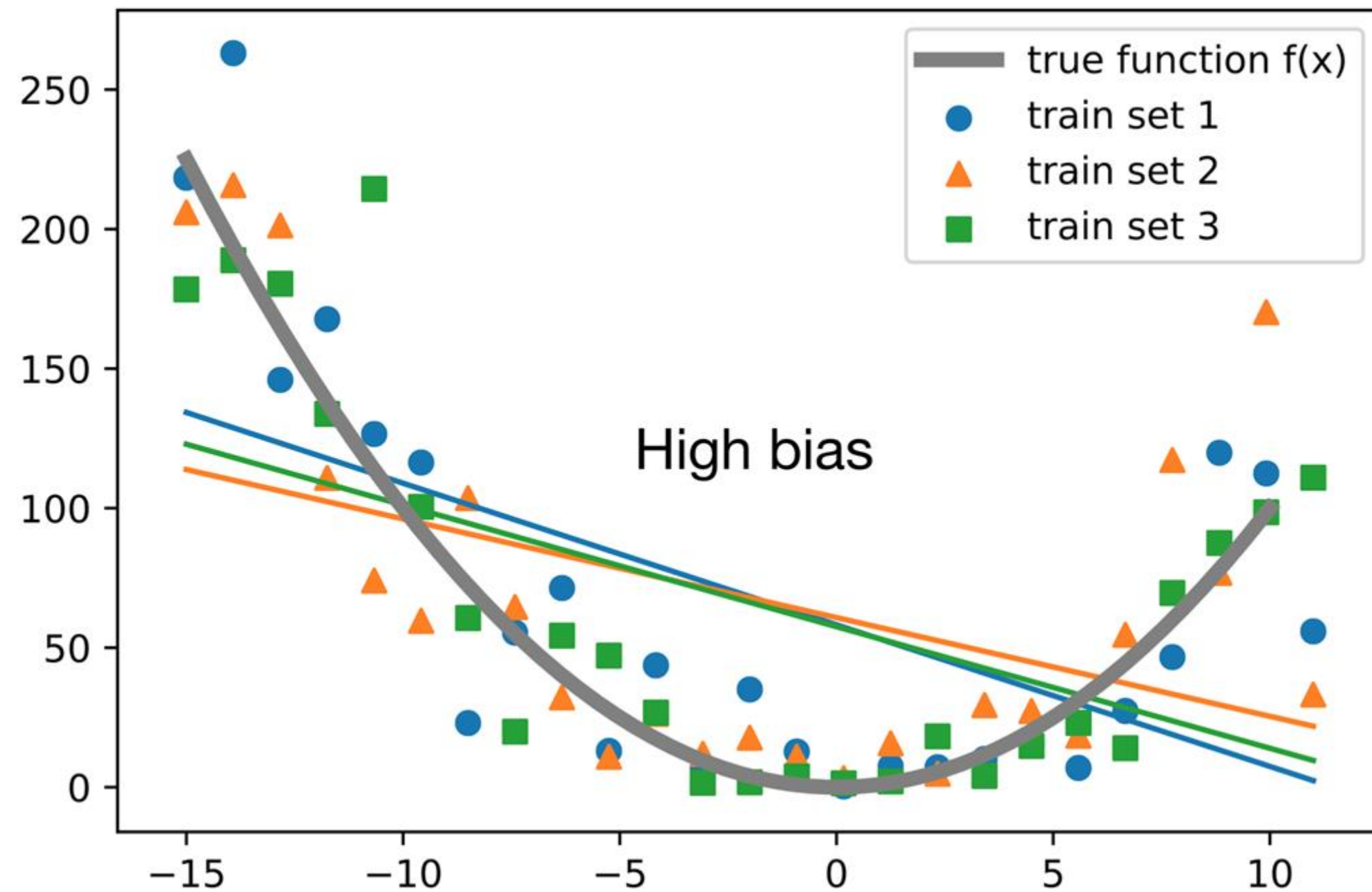
$$\text{variance} := \mathbb{E}[\mathbb{E}(\hat{y}) - \hat{y}]^2$$

$$\text{noise} := \mathbb{E}[y - \mathbb{E}(y)]^2$$



Bias-variance decomposition

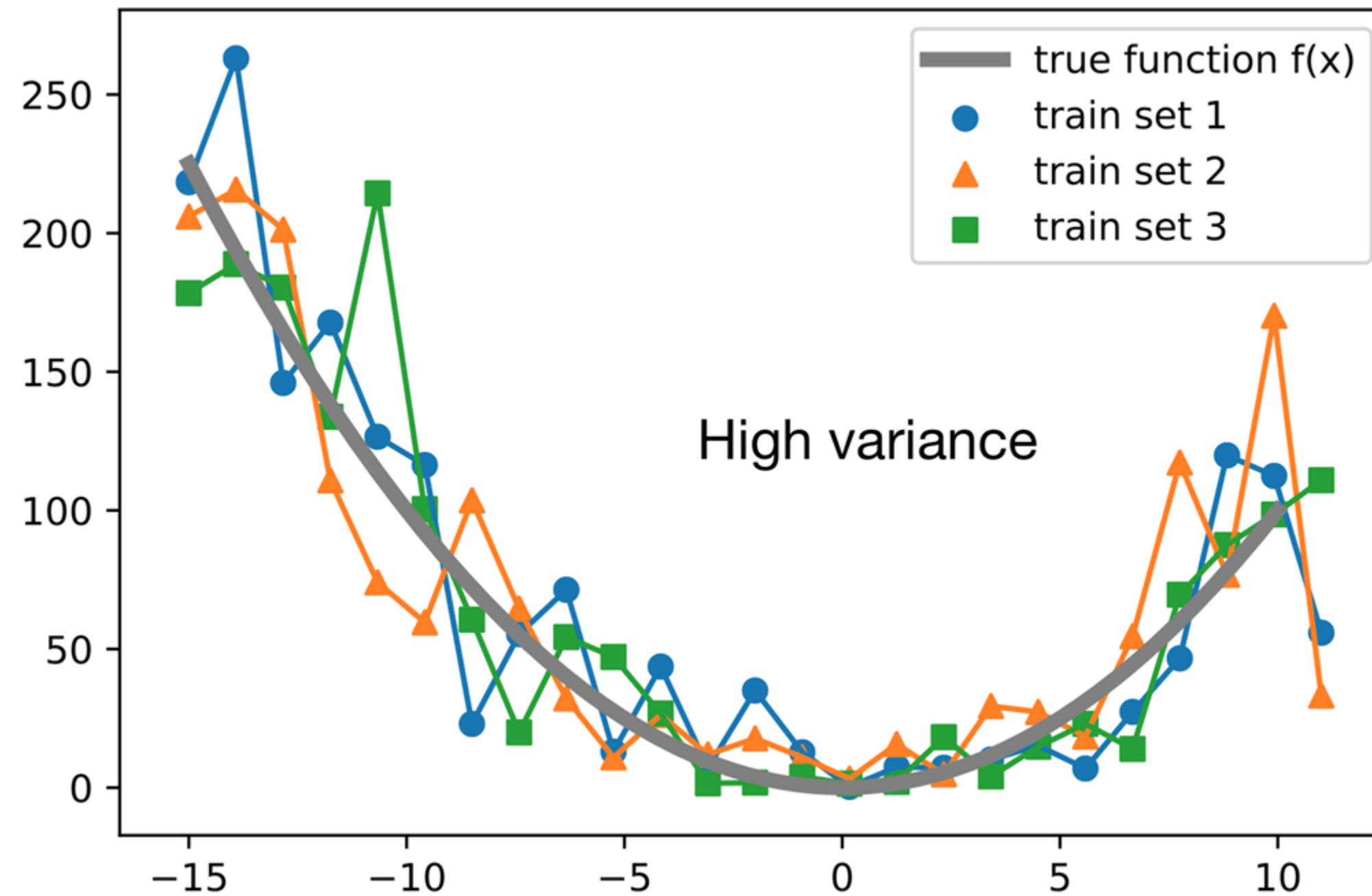
- Пример высокого смещение у линейной модели





Bias-variance decomposition

- Пример высокого разброса у более сложной модели

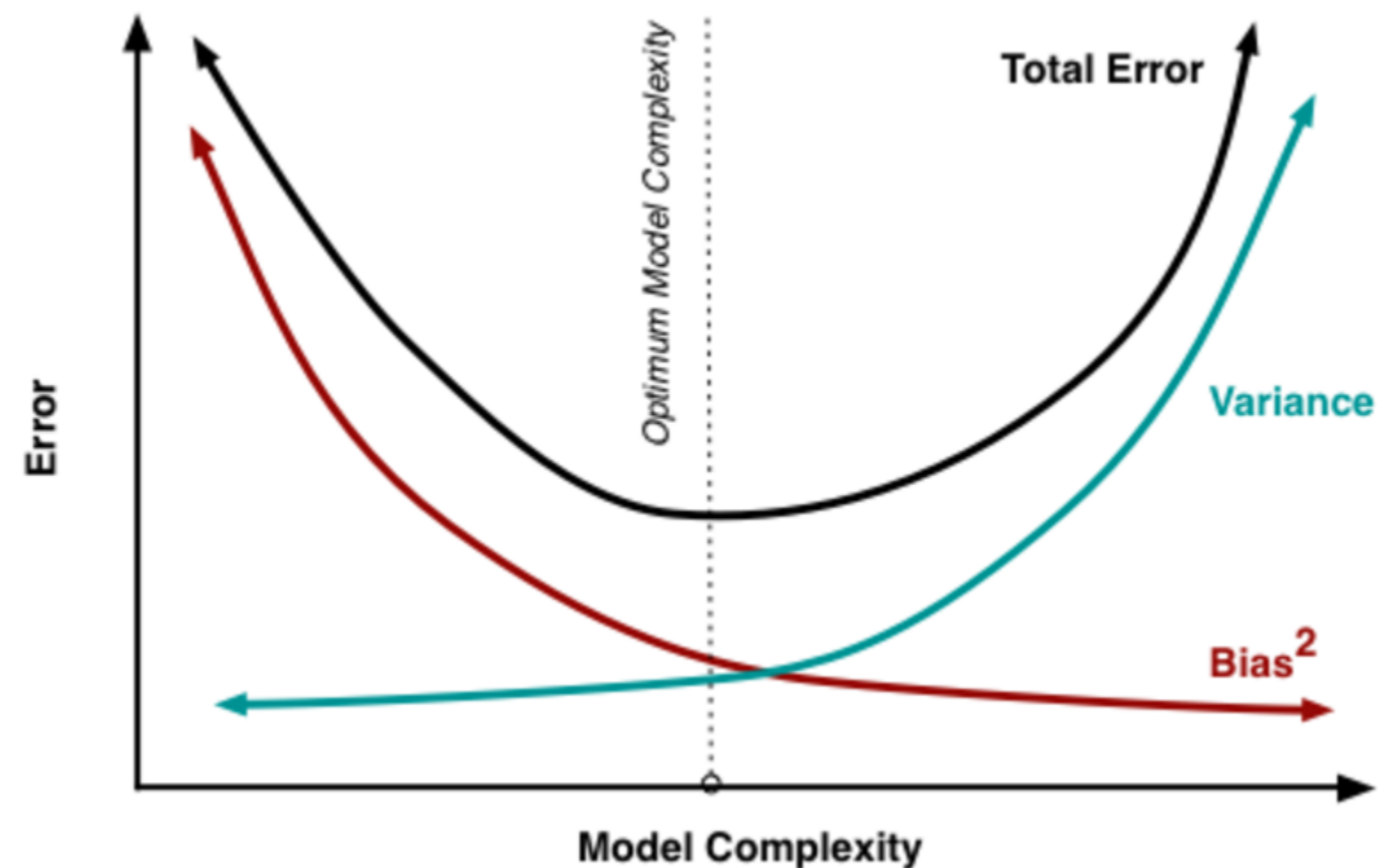


Bias-variance tradeoff

- Недообученная модель имеет низкий разброс, но высокое смещение
- Переобученная модель имеет высокий разброс, но низкое смещение

Bias-variance tradeoff

- Недообученная модель имеет низкий разброс, но высокое смещение
- Переобученная модель имеет высокий разброс, но низкое смещение
- Необходимо искать золотую середину

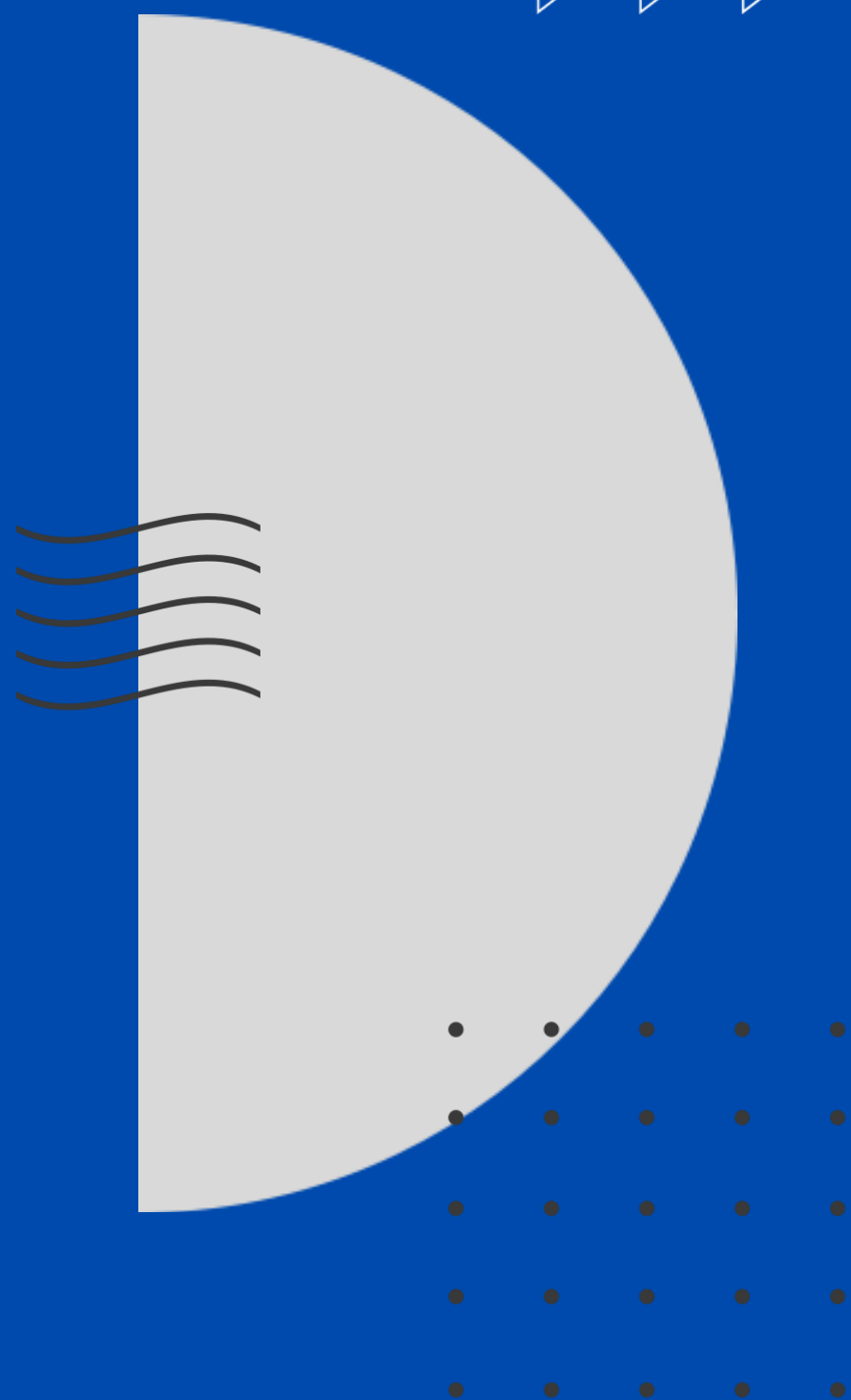


Bias-variance + бэггинг

- Смещение $a_N(x)$ такое же, как у $b_n(x)$
- Разброс $a_N(x)$:

$$\frac{1}{N} (\text{разброс } b_n(x)) + \text{ковариация}(b_n(x), b_m(x))$$

- Если базовые модели независимы, то разброс уменьшается в N раз!
- Чем более похожи выходы базовых моделей, тем меньше эффект от построения композиции



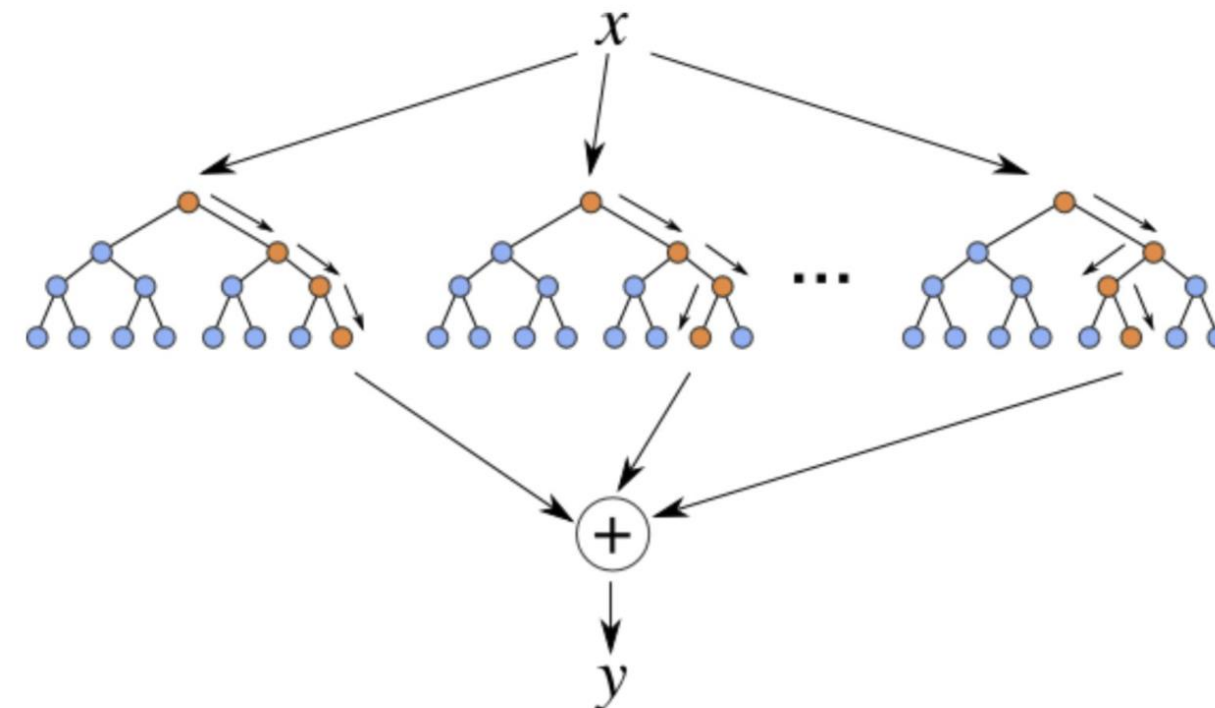
06

Алгоритм случайного леса

Алгоритм

Для $n = 1, \dots, N$:

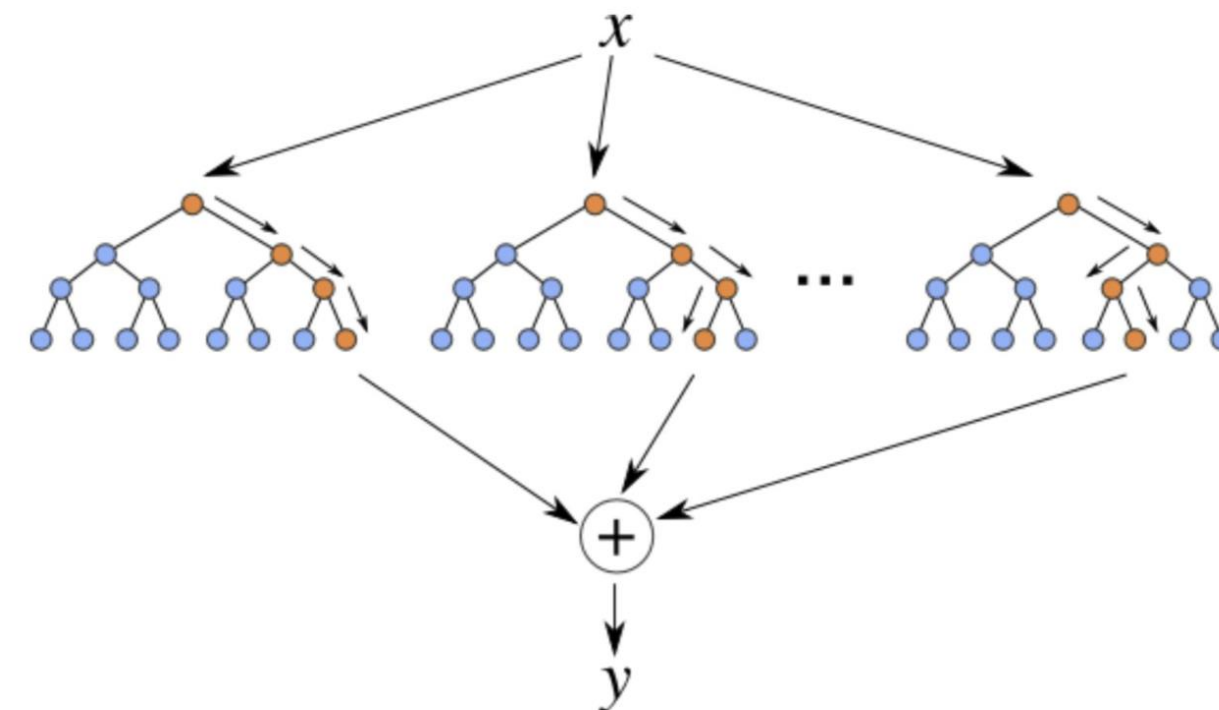
1. Генерируем выборку \mathbf{X}' с помощью бутстрапа
2. Строим решающее дерево $b_n(x)$ по выборке \mathbf{X}'
3. Строим дерево, пока не выполнится критерий остановки (обычно пока не достигнет n_{\min} объектов в листах)
4. Оптимальное разбиение ищется среди q случайных признаков, в каждом узле (не дереве) обновляется набор признаков



Алгоритм

Для $n = 1, \dots, N$:

1. Генерируем выборку \mathbf{X}' с помощью бутстрапа
2. Строим решающее дерево $b_n(x)$ по выборке \mathbf{X}'
3. Строим дерево, пока не выполнится критерий остановки (обычно пока не достигнет n_{\min} объектов в листах)
4. Оптимальное разбиение ищется среди q случайных признаков, в каждом узле (не дереве) обновляется набор признаков



Выбор предиката

4. Оптимальное разбиение ищется среди q случайных признаков, в каждом узле (не дереве) обновляется набор признаков

$$j, t = \arg \min_{j, t} Q(R_m, j, t)$$

Будем искать лучший предикат среди случайного подмножества признаков размера q



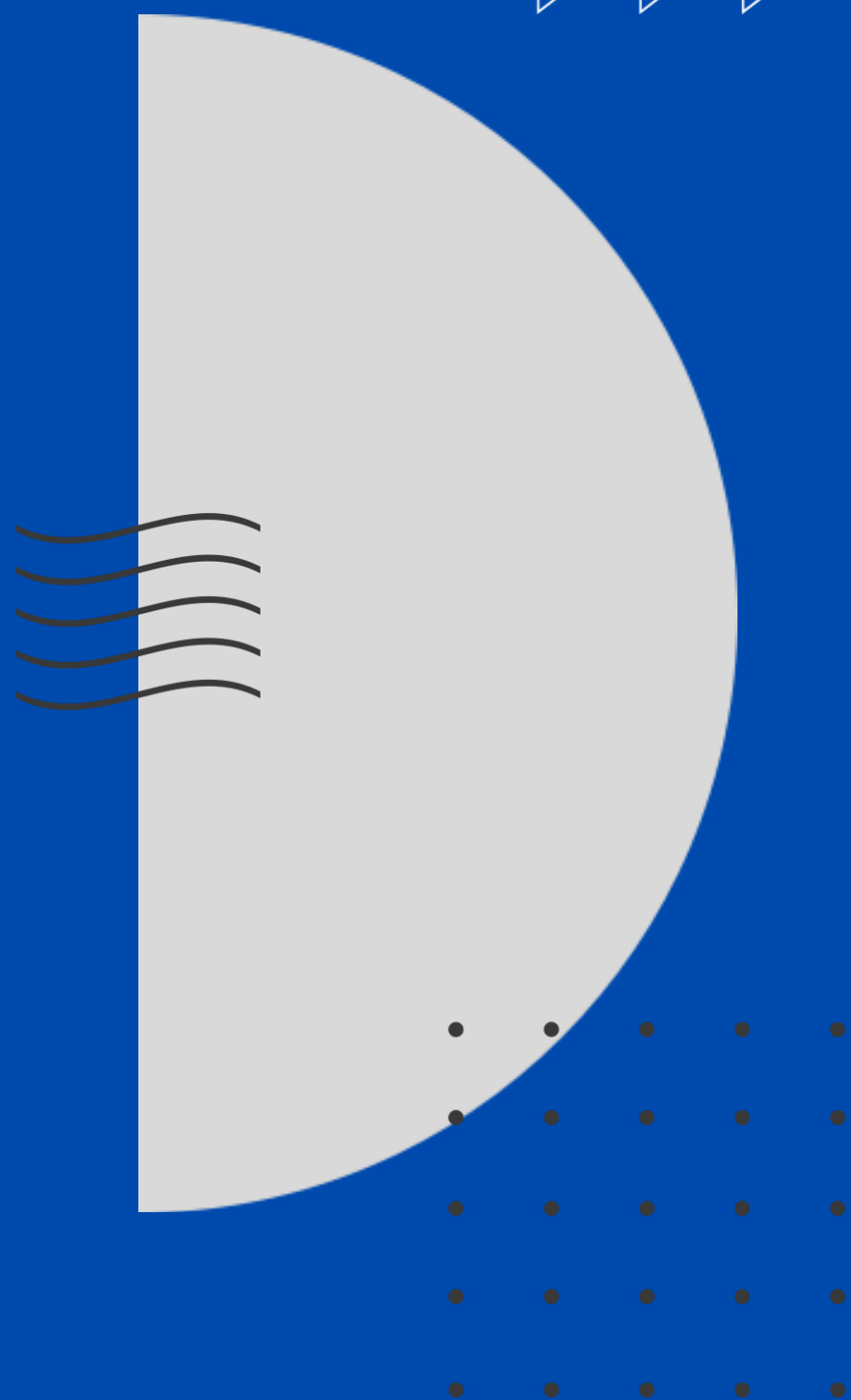
Рекомендации для q :

- Регрессия:

$$q = \frac{d}{3}$$

- Классификация:

$$q = \sqrt{d}$$



07

Особенности применения

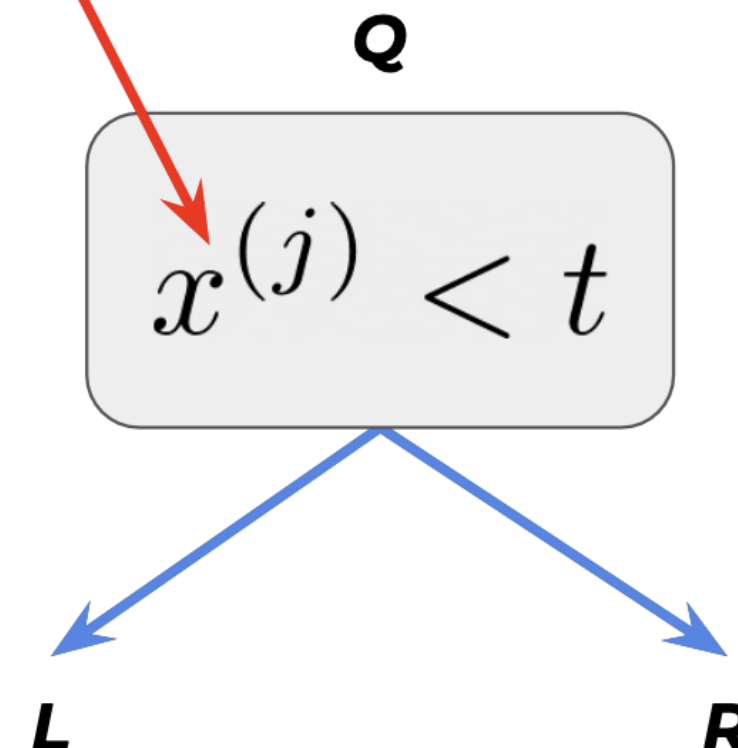
Пропуск значений

Если значение отсутствует, можно было бы использовать оба поддерева и усреднить их прогнозы. Предлагается отправить его в каждую из дальнейших веток и получить по ним предсказания. Эти предсказания мы усредним с

весами

$$\hat{y} = \frac{|L|}{|Q|} \hat{y}_L + \frac{|R|}{|Q|} \hat{y}_R$$

Missing value



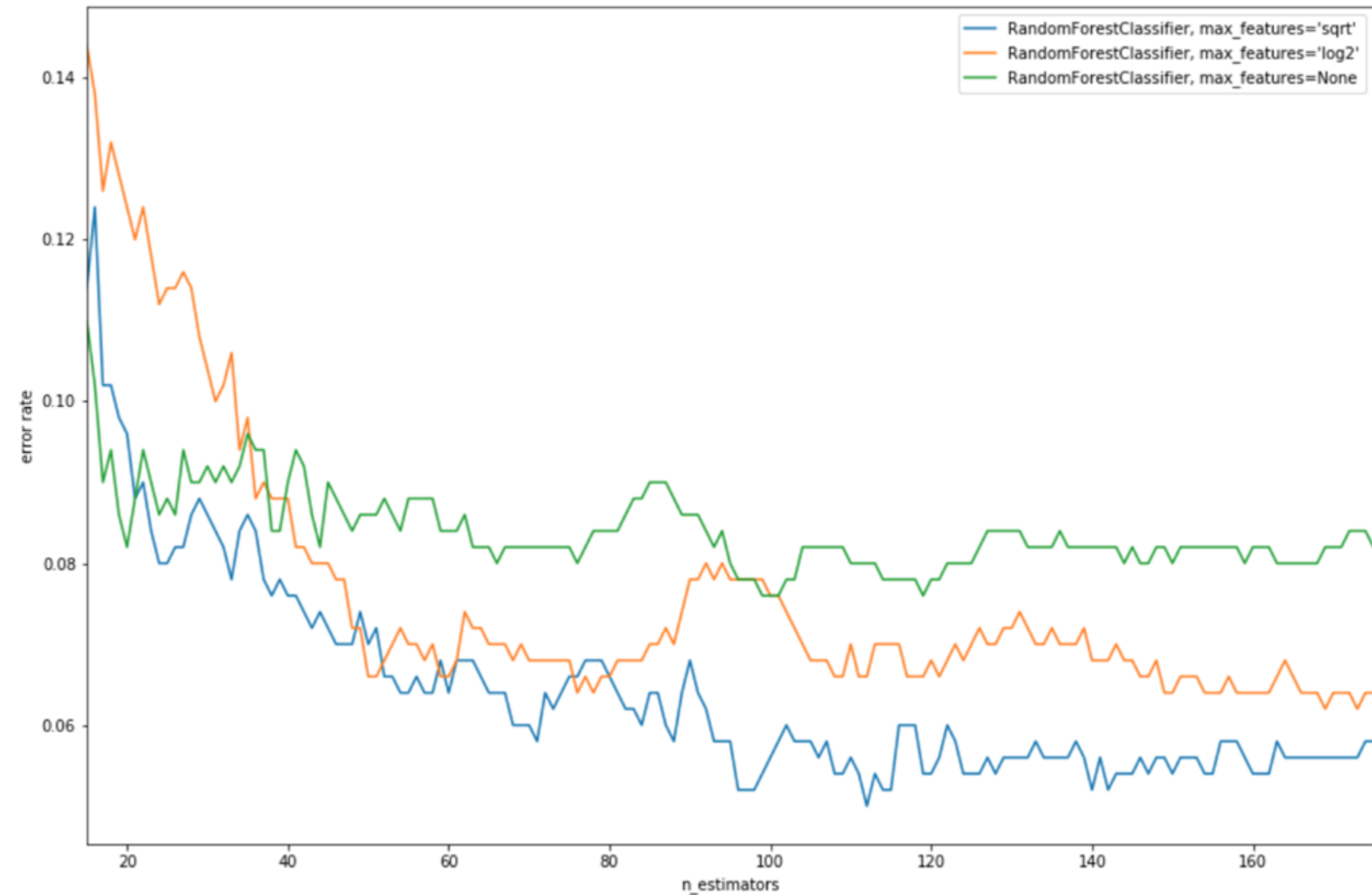
Переобучение

- Мы уже узнали, что увеличение количества базовых моделей приводит к уменьшению разброса

Получается мы можем брать неограниченное количество базовых моделей и уменьшать ошибку?

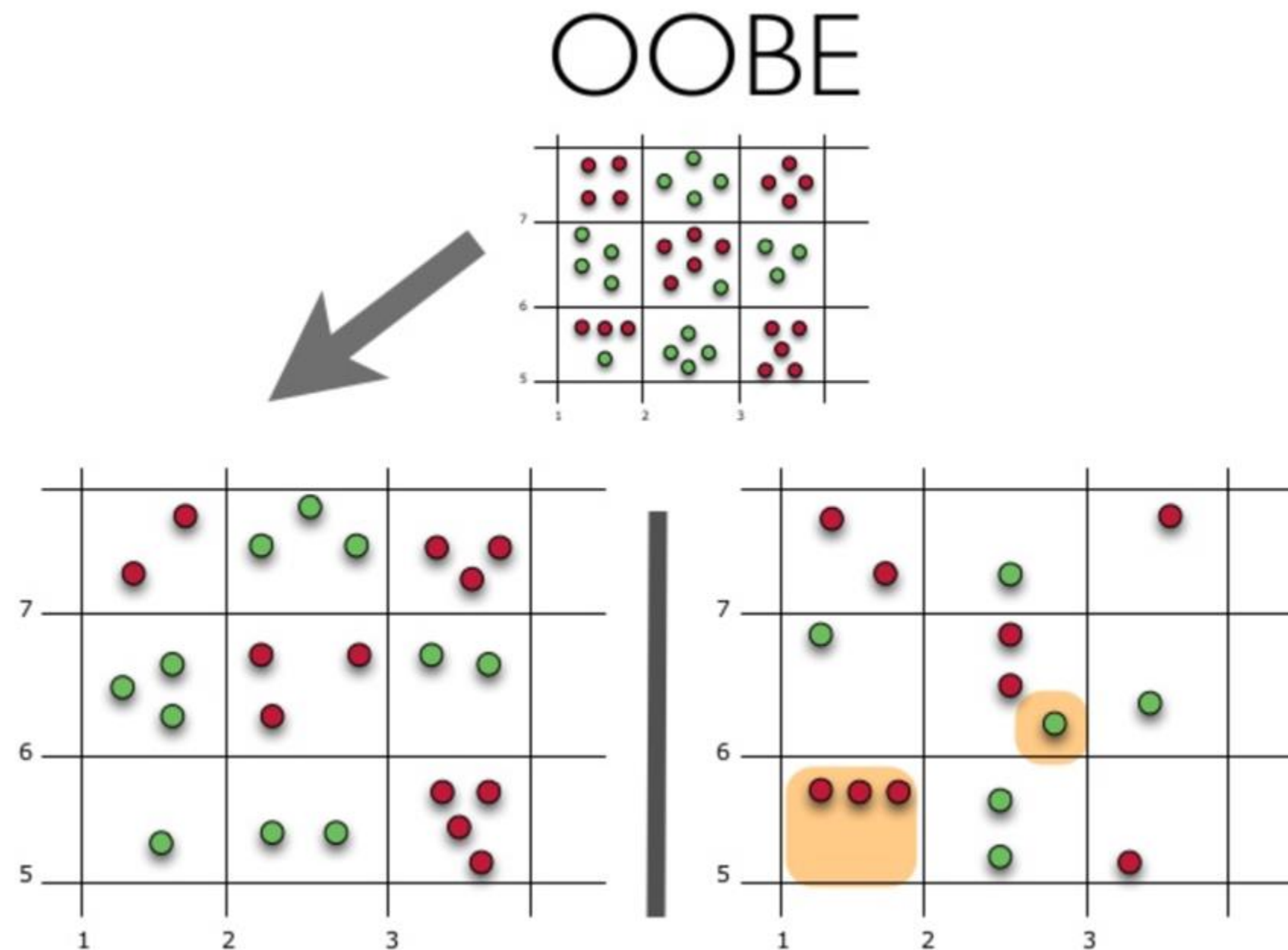
Переобучение

- Ошибка сначала убывает, а затем выходит на один уровень
- Случайный лес не переобучается при росте N



Out-of-bag error

- Благодаря особенности построения случайного леса, потребность в кросс-валидации отсутствует



Out-of-bag error

- Благодаря особенности построения случайного леса, потребность в кросс-валидации отсутствует
- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

Out-of-bag error

- Благодаря особенности построения случайного леса, потребность в кросс-валидации отсутствует
- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- X_n — обучающая выборка для $b_n(x)$
- Можно оценить ошибку на новых данных:

$$Q_{test} = \frac{1}{\ell} \sum_{i=1}^{\ell} L \left(y_i, \frac{1}{\sum_{n=1}^N [x_i \notin X_n]} \sum_{n=1}^N [x_i \notin X_n] b_n(x_i) \right)$$

Важность признаков

- Перестановочный метод для проверки важности j -го признака
- Перемешиваем соответствующий столбец в матрице «объекты-признаки» для тестовой выборки
- Измеряем качество модели
- Чем сильнее оно упало, тем важнее признак

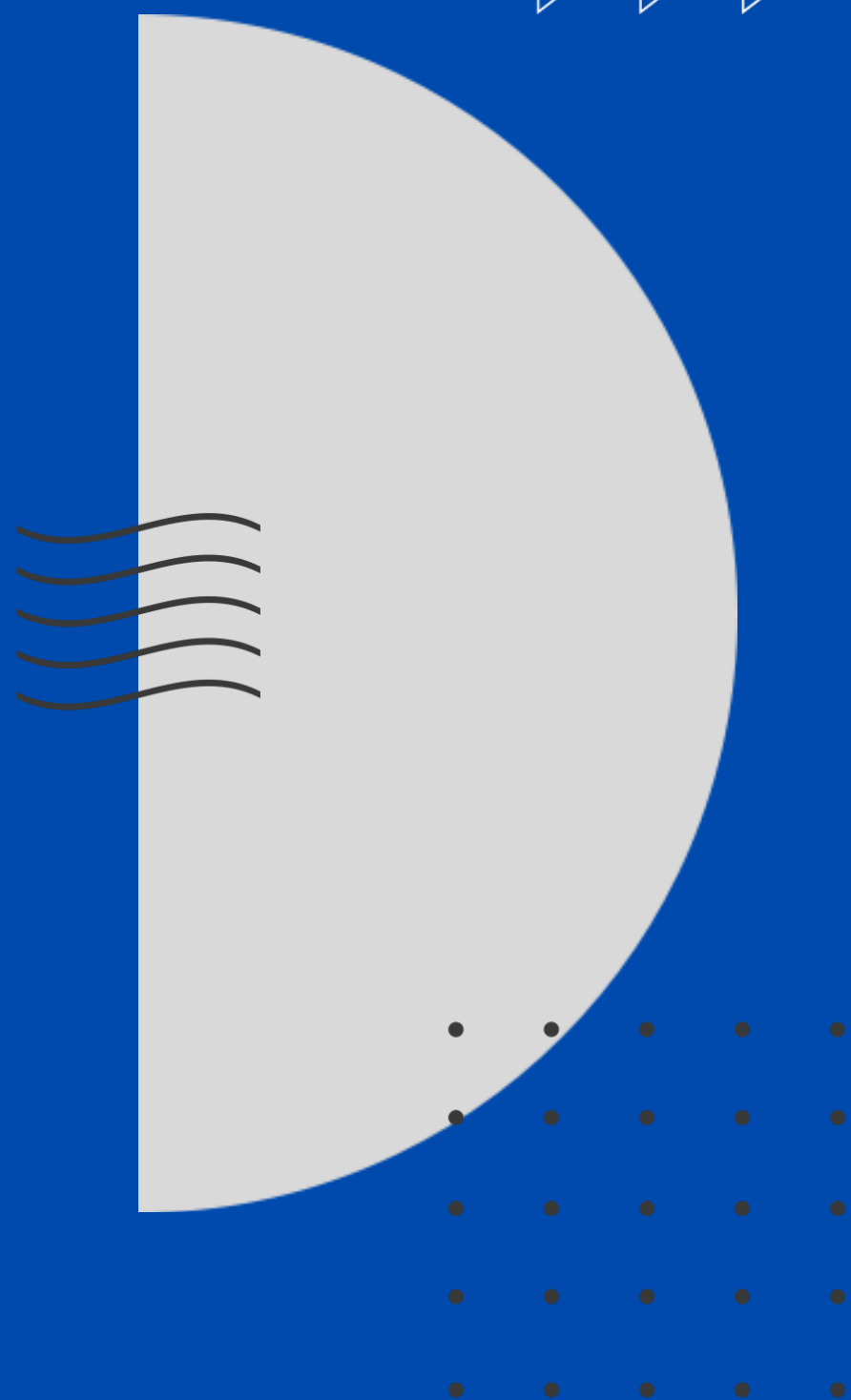
Объект / Признак	Возраст	Вес	Рост
0	17	60	165
1	24	86	193
2	28	98	185



Объект / Признак	Возраст	Вес	Рост
0	17	86	165
1	24	98	193
2	28	60	185

Резюме

- Случайный лес — метод на основе бэггинга, в котором делается попытка повысить разнообразие деревьев
- Метод практически без гиперпараметров
- Можно оценить обобщающую способность без тестовой выборки



Место для ваших
вопросов