Ensembles

Revision

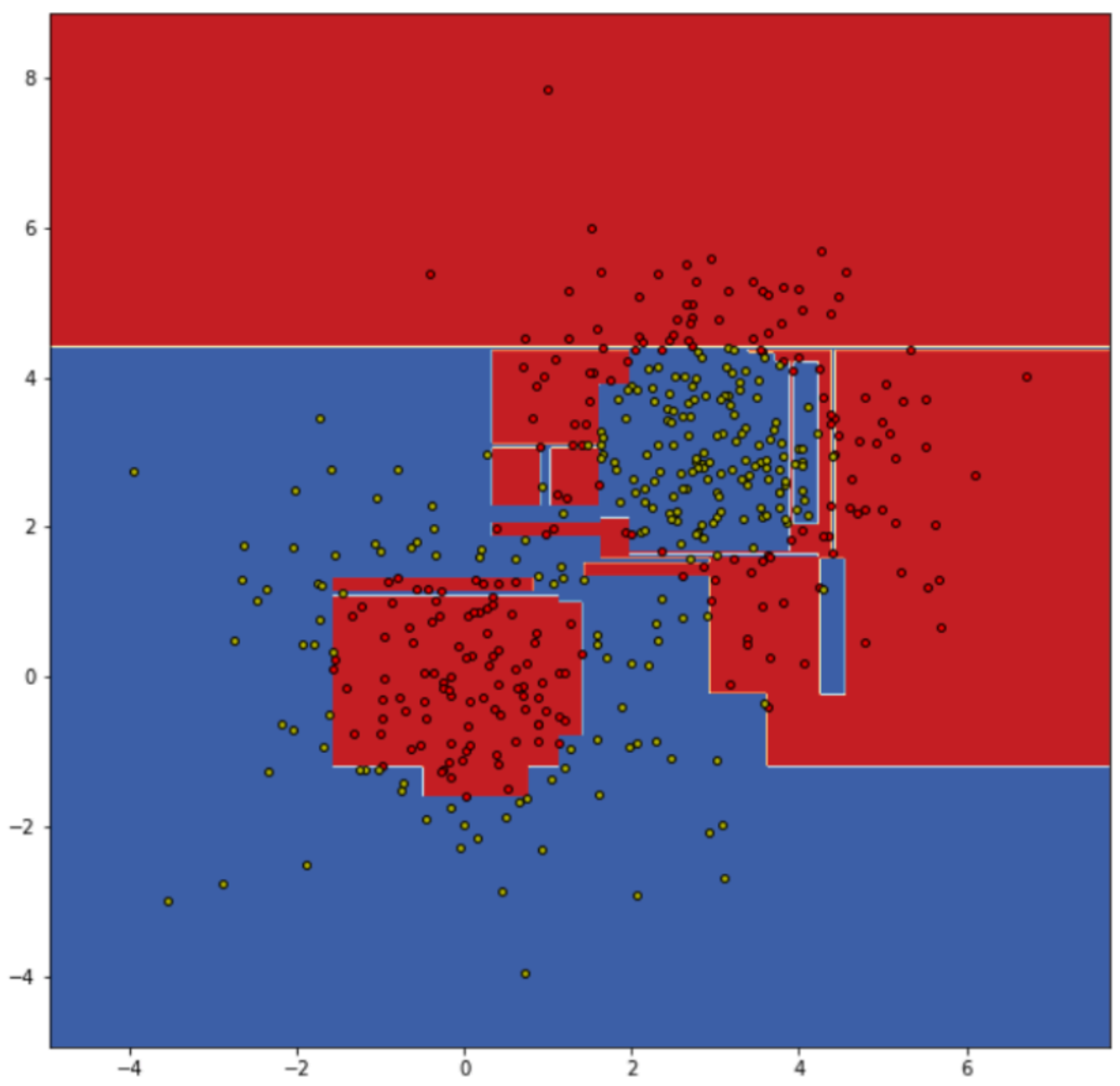
0. Unstable Decision Trees

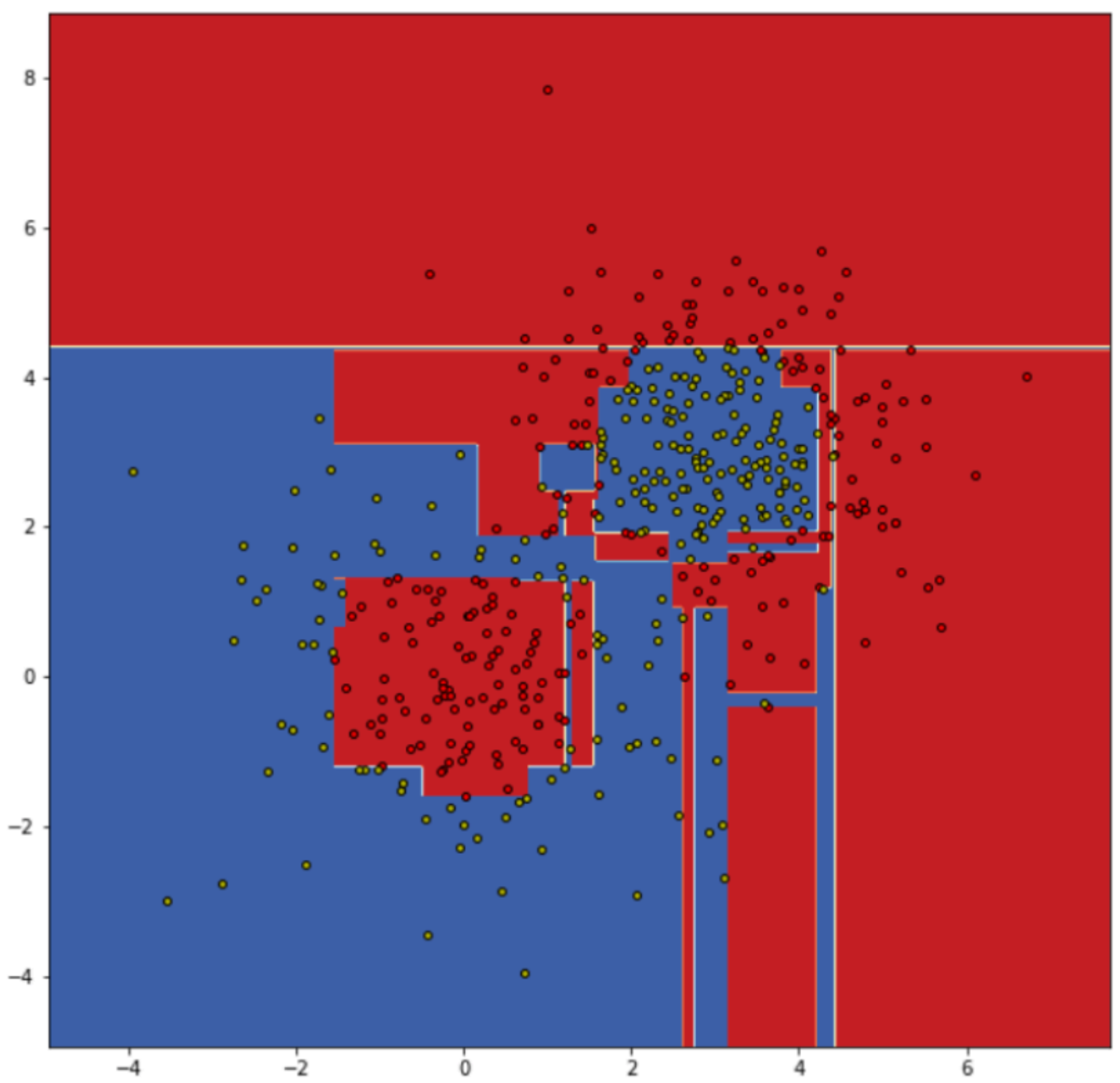
- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- Обучаем модель $a(x)$
- Ожидаем, что модель устойчивая
- То есть не сильно меняется при небольших изменениях в X
- $\tilde{X}$ — случайная подвыборка, примерно 90% исходной

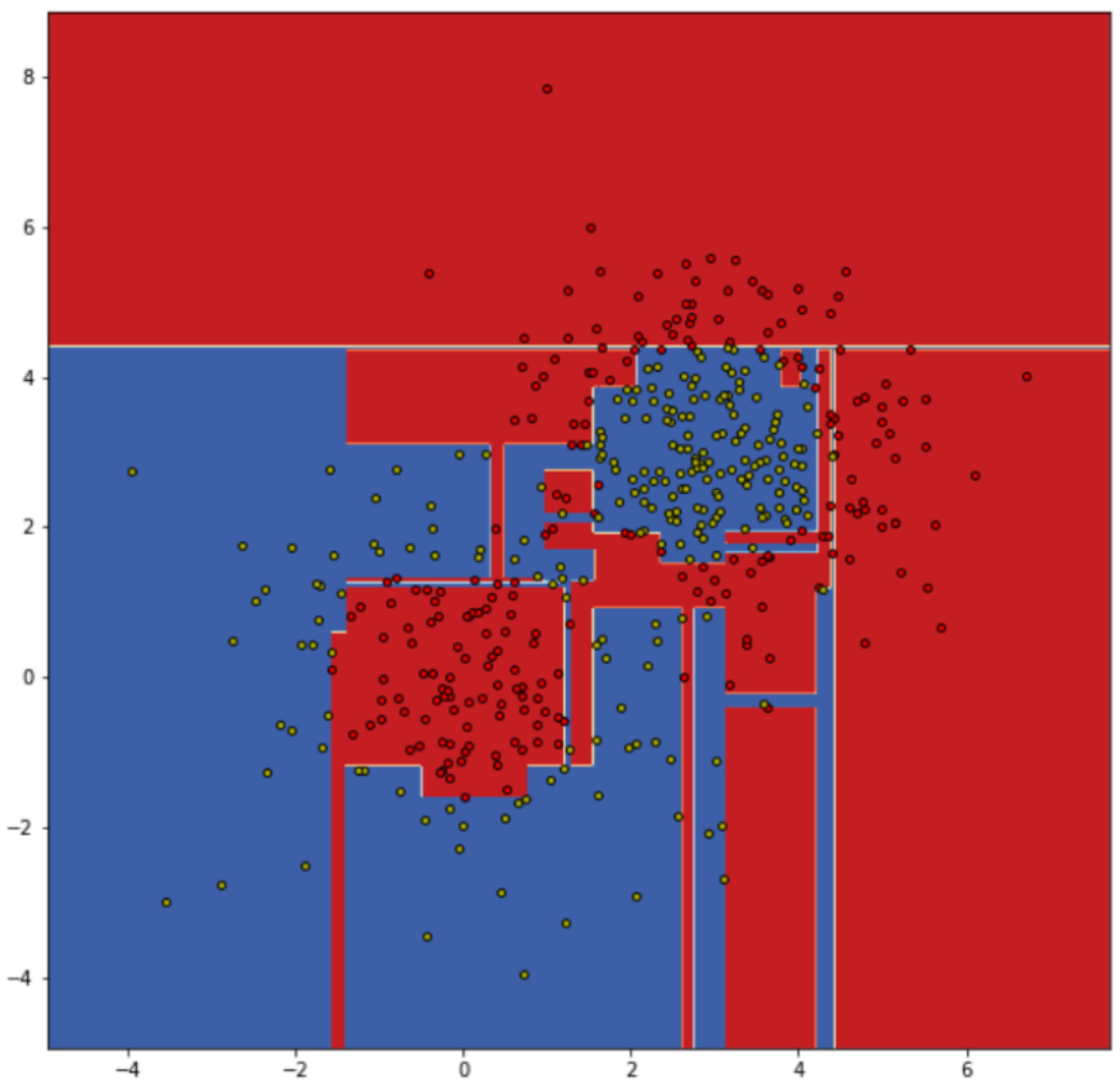
- *Что будет происходить с деревьями на разных подборках?*

0. Unstable Decision Trees

- *Что будет происходить с деревьями на разных подборках?*







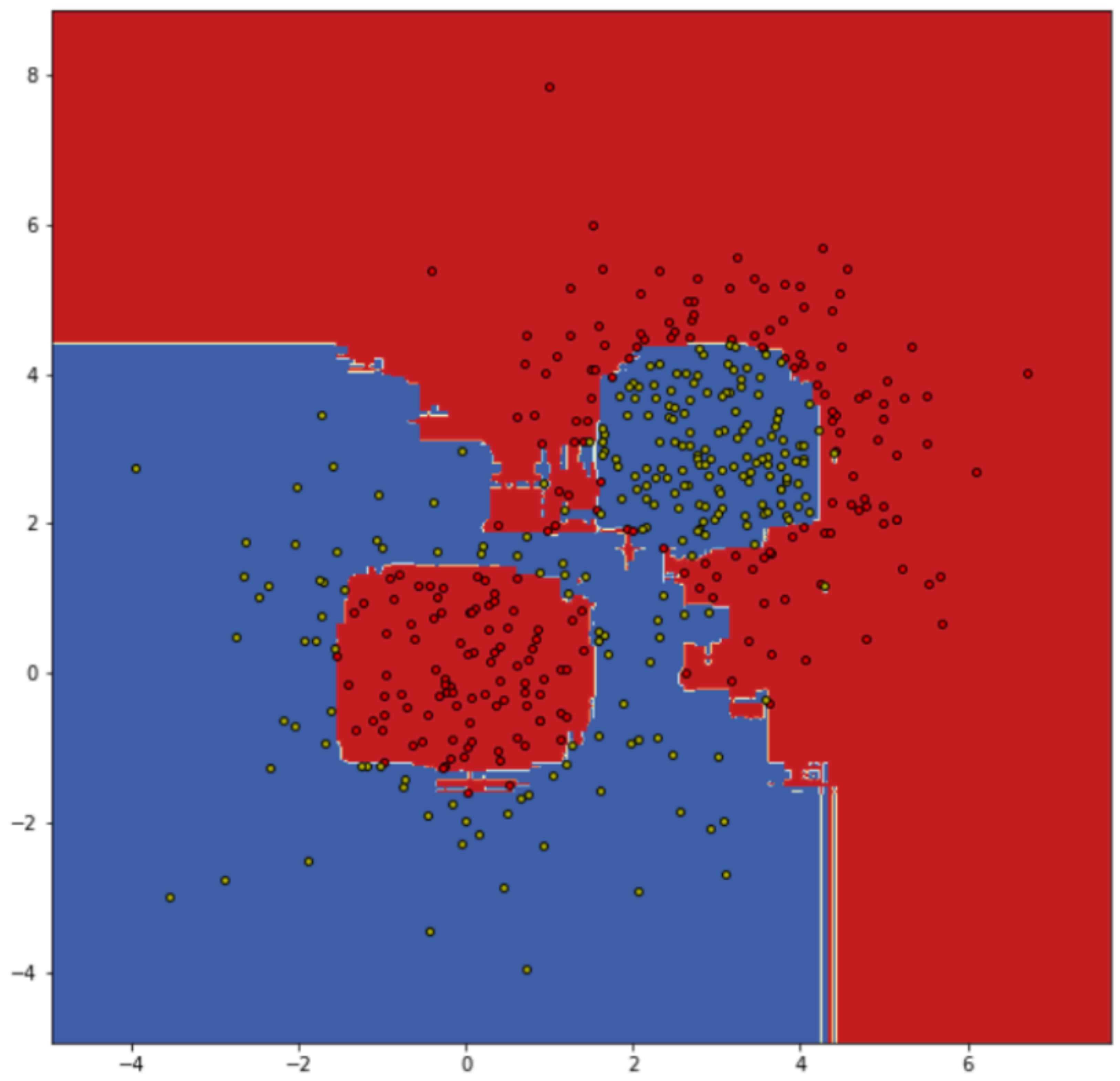
0. **Unstable** Decision Trees

- *Что будет происходить с деревьями на разных подборках?*
- *А если на всех моделях построить ансамбль и голосовать большинством?*

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

За какой класс больше моделей проголосовало, тот и победил

Предсказание класса от модели под номером ***n***



0. Как сделать ансамбль?

- *Классификация*
  - *Базовые модели:* $b_1(x), \ldots, b_N(x)$
  - *Каждая - лучше случайного угадывания*
  - ***Majority Voice***

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

- *Регрессия*
  - *Базовые модели:* $b_1(x), \ldots, b_N(x)$
  - *Каждая - лучше случайного угадывания*
  - ***Усреднение наблюдений***

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

0. Как сделать ансамбль?

- *Классификация*
  - *Базовые модели:* $b_1(x), \ldots, b_N(x)$
  - *Каждая - лучше случайного угадывания*
  - ***Majority Voice***

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

- *Регрессия*
  - *Базовые модели:* $b_1(x), \ldots, b_N(x)$
  - *Каждая - лучше случайного угадывания*
  - ***Усреднение наблюдений***

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

Откуда взять базовые модели?

Вариант Uno: ***независимо*** *обучить на разных данных*

Вариант Dos: ***последовательно*** *обучить чтобы улучшать предшественника*

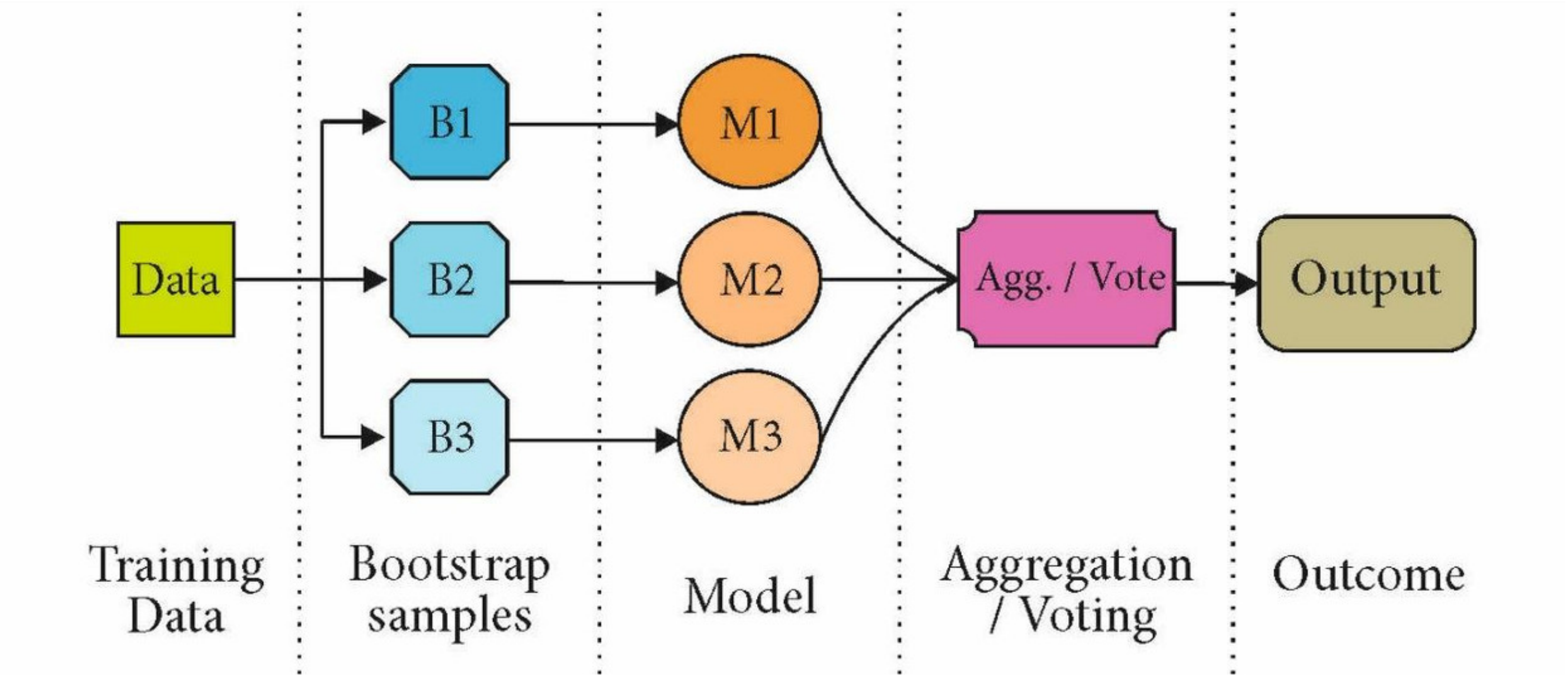
0. Как сделать ансамбль?

Вариант Uno:
***независимо** обучить на разных данных*

Варианты рандомизации:

- **Бэггинг: случайная подвыборка**
- **Случайные подпространства: случайное подмножество признаков**

0. Bagging: bootstrap + aggregation

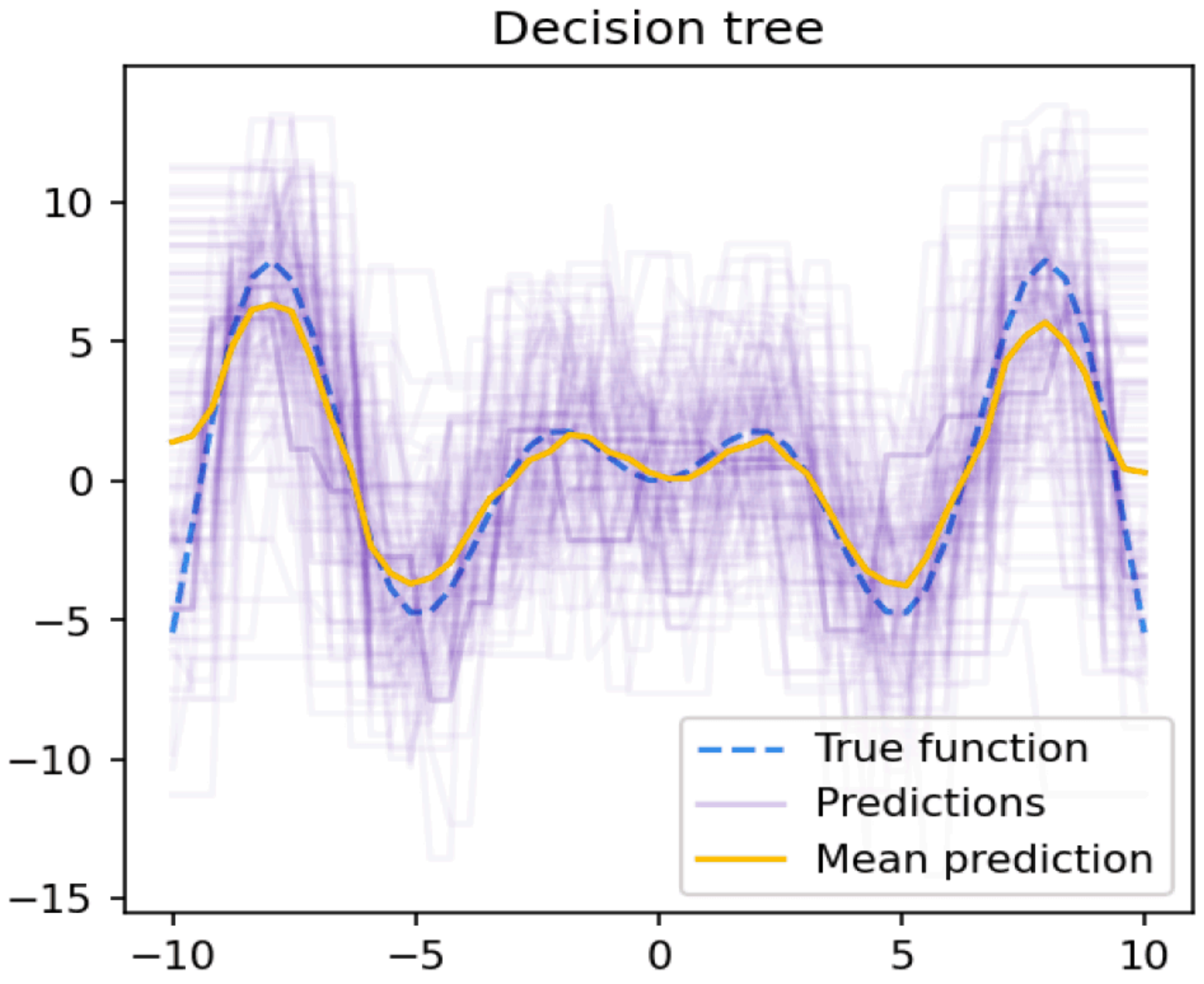


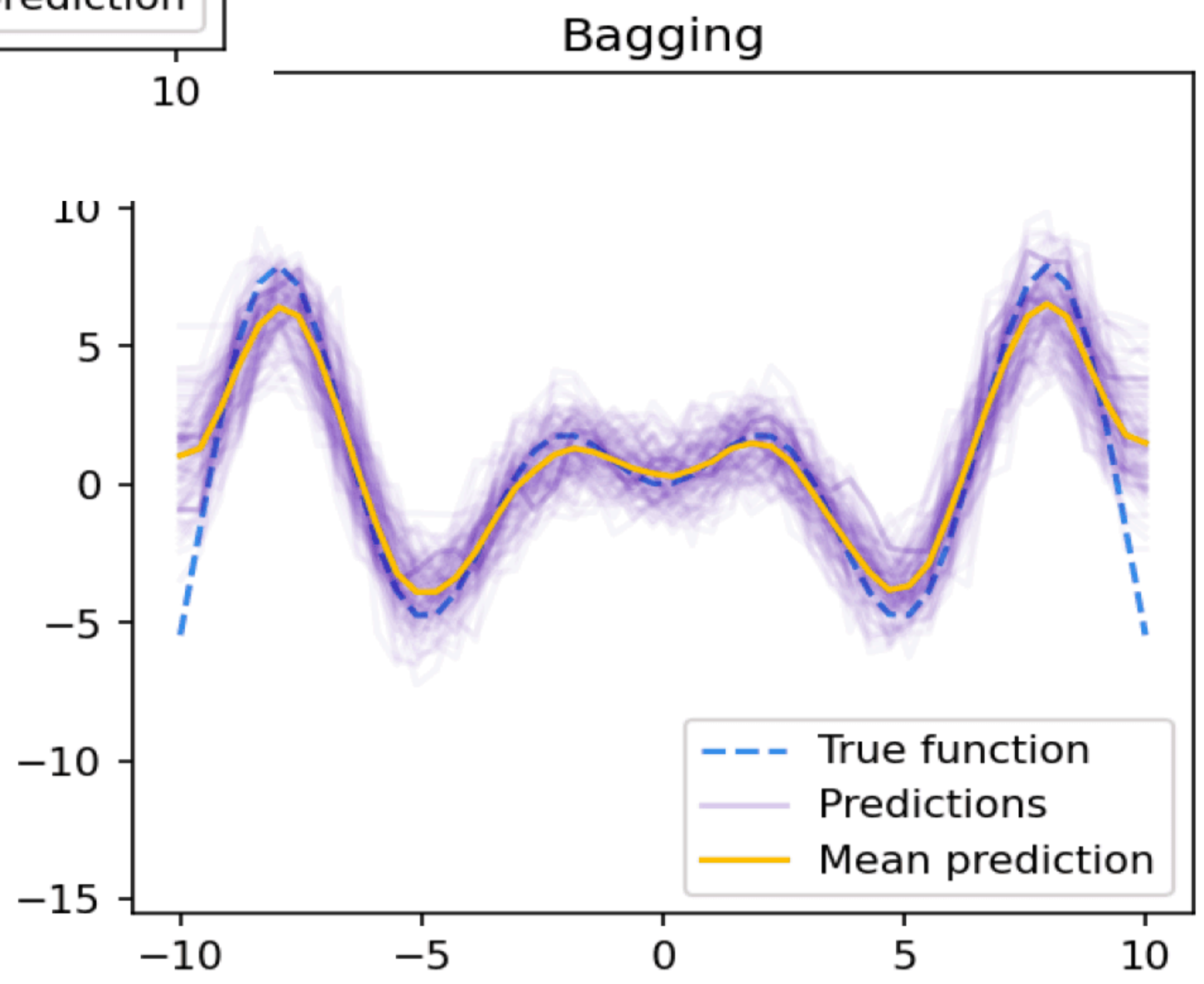
- Aggregation

  На каждом датасете обучим выбранную нами базовую архитектуру: $b_i(x) = b(x, X^i)$

  Объединим все модели в единый ансамбль: $a(x) = \frac{1}{K}(b_1(x) + \ldots + b_k(x))$

0. **Bagging:** bootstrap + aggregation





- Функция:
  $y(x, \varepsilon) = x\sin(x) + \mathcal{N}(0,9)$

- Обучим 100 решающих деревьев глубины 7 на случайных выборках размера 20

- Обучим 100 раз бэггинг над решающими деревьями на случайных выборках размера 20

0. **Bias-**Variance decomposition

- *Функционал ошибки с MSE:*

$$Q(a) = \mathbb{E}_x \mathbb{E}_{X,\varepsilon}[y(x,\varepsilon) - a(x,X)]^2$$

$\sigma^2 = \mathbb{E}_x \mathbb{E}_\varepsilon [y(x,\varepsilon) - f(x)]^2$
неустранимый ***шум*** в данных

- *Аналогичное представление функционала:*

$$Q(a) = \mathbb{E}_x(\mathrm{bias}^2_X a(x,X)) + \mathbb{E}_x \mathbb{V}_X[a(x,X)] + \sigma^2$$

$\mathrm{bias}_X a(x,X) = f(x) - \mathbb{E}_X[a(x,X)]$
смещение предсказания алгоритма, усреднённого по всем возможным обучающим выборкам, относительно истины

$\mathbb{V}_X[a(x,X)] = \mathbb{E}_X[a(x,X) - \mathbb{E}_X[a(x,X)]]^2$
разброс предсказаний алгоритма в зависимости от обучающей выборки

Random Forest

1. Случайный лес **/предпосылки**

- Будем объединять модели в композиции через усреднение или голосование большинством
- Бэггинг — композиция моделей, обученных независимо на случайных подмножествах объектов
- Можно ещё рандомизировать по признакам
- Как лучше всего?

1. Случайный лес /**классика**

FitNode (R_m, m):

If Stop_Criteria:

C_m - prediction

m - leaf node

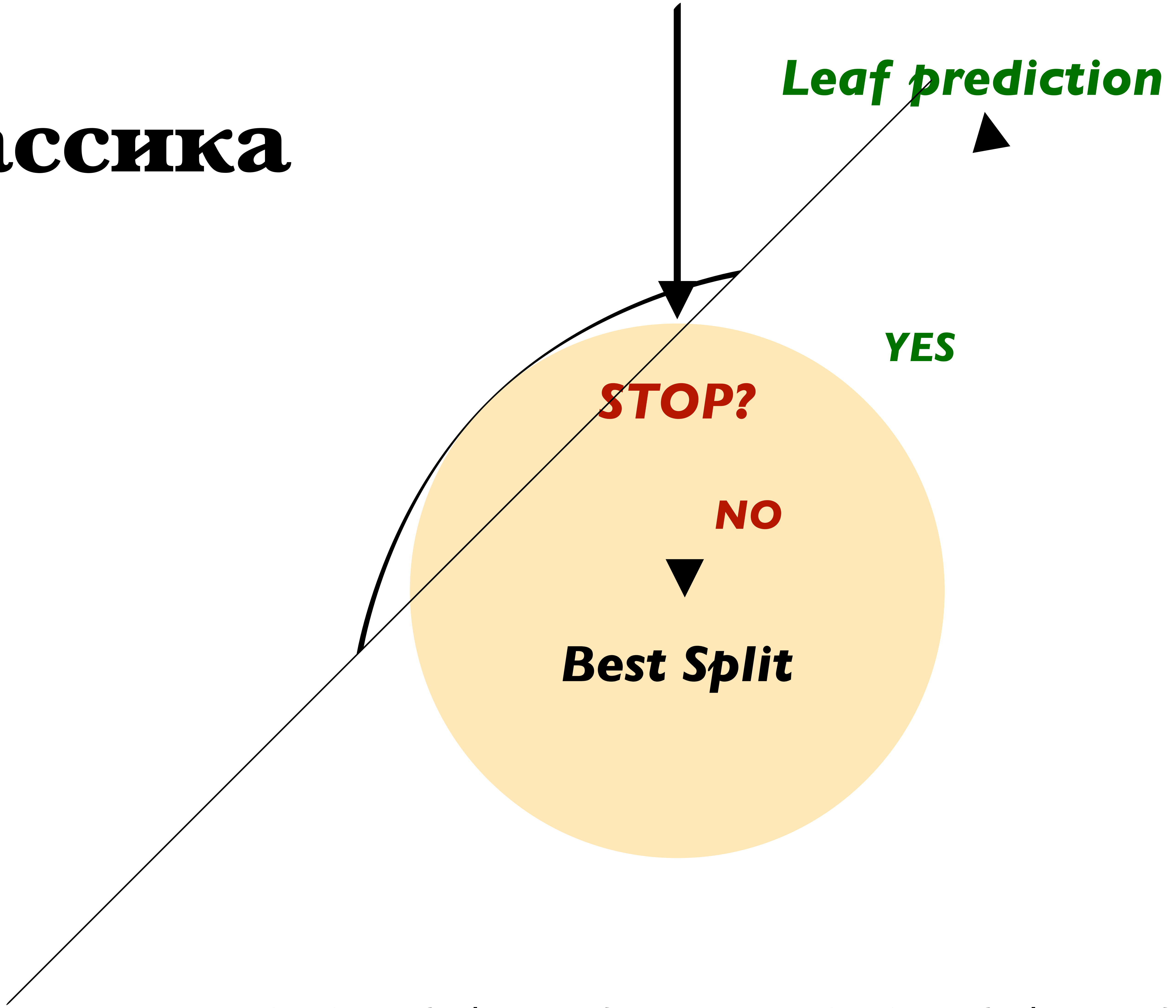
$$j, t = argmax\ Q(R_m, j, t) \text{ (Quality)}$$

$$R_l = \{(x, y) \in R_m \,|\, [x_j < t]\}$$

$$R_r = \{(x, y) \in R_m \,|\, [x_j \geq t]\}$$

FitNode(R_l, l)

FitNode(R_r, r)



$$R_1(j, t) = \{x \,|\, x_j < t\} \qquad R_2(j, t) = \{x \,|\, x_j \geq t\}$$

1. Случайный лес /**классика**

FitNode (R_m, m):

If Stop_Criteria:

C_m - prediction

m - leaf node

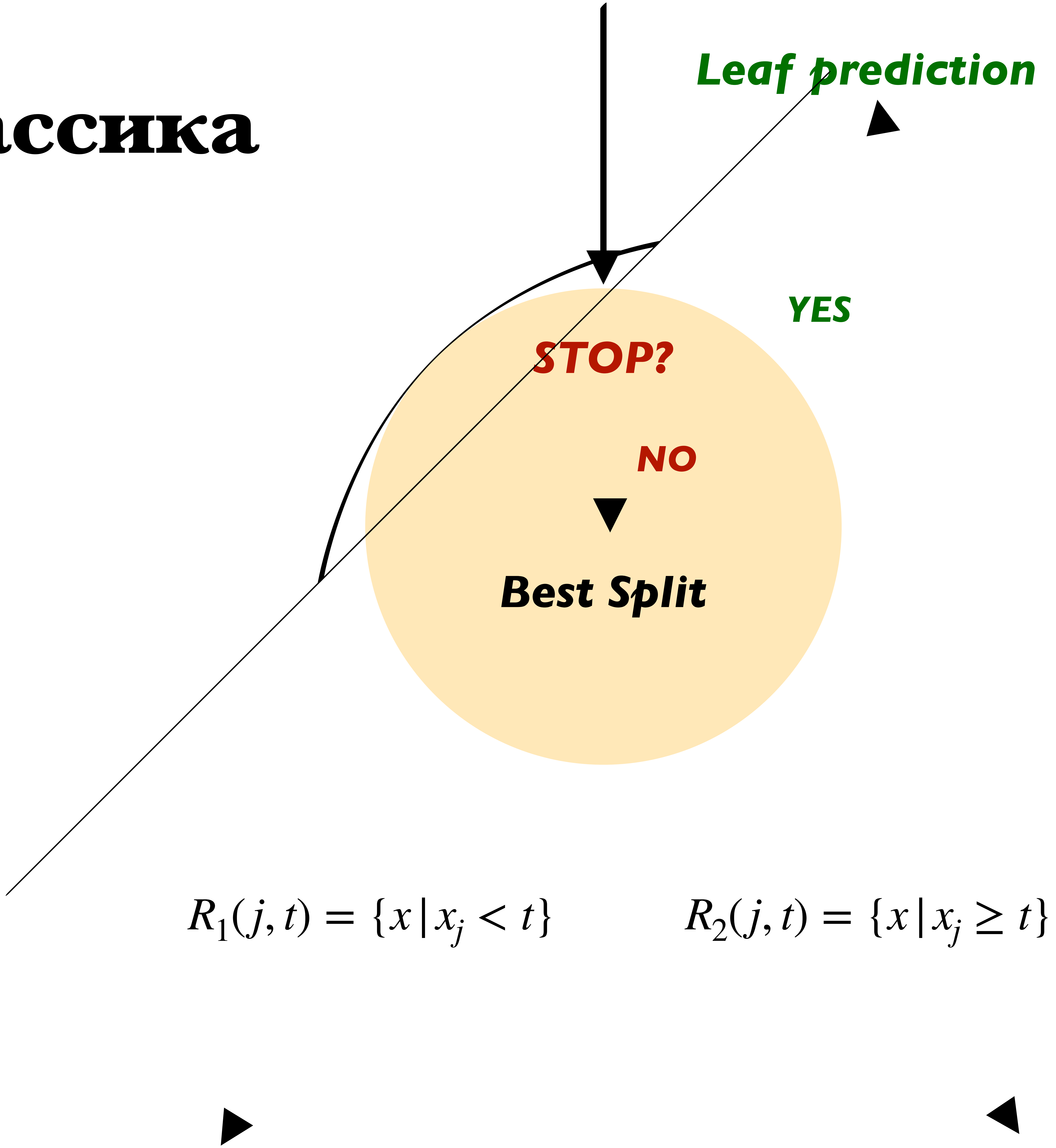
$$j, t = \color{red}{\text{argmax } Q(R_m, j, t)} \text{ (Quality)}$$

$$R_l = \{(x, y) \in R_m \,|\, [x_j < t]\}$$

$$R_r = \{(x, y) \in R_m \,|\, [x_j \geq t]\}$$

FitNode(R_l, l)

FitNode(R_r, r)



1. Случайный лес **/случайное подпространство**

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

- Будем искать лучший предикат среди случайного подмножества признаков размера q

1. Случайный лес /**алгоритм**

Для $n = 1, \dots, N$:

1. Сгенерировать выборку $\tilde{X}$ с помощью бутстрапа
2. Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}$
3. Дерево строится, пока в каждом листе не окажется не более n_{min} объектов
4. Оптимальное разбиение ищется среди q случайных признаков

Выбираются заново при каждом разбиении!

1. Случайный лес **/алгоритм**

Алгоритм 3.1. Random Forest

1: **для** $n = 1, \dots, N$

2: Сгенерировать выборку $\tilde{X}_n$ с помощью бутстрэпа

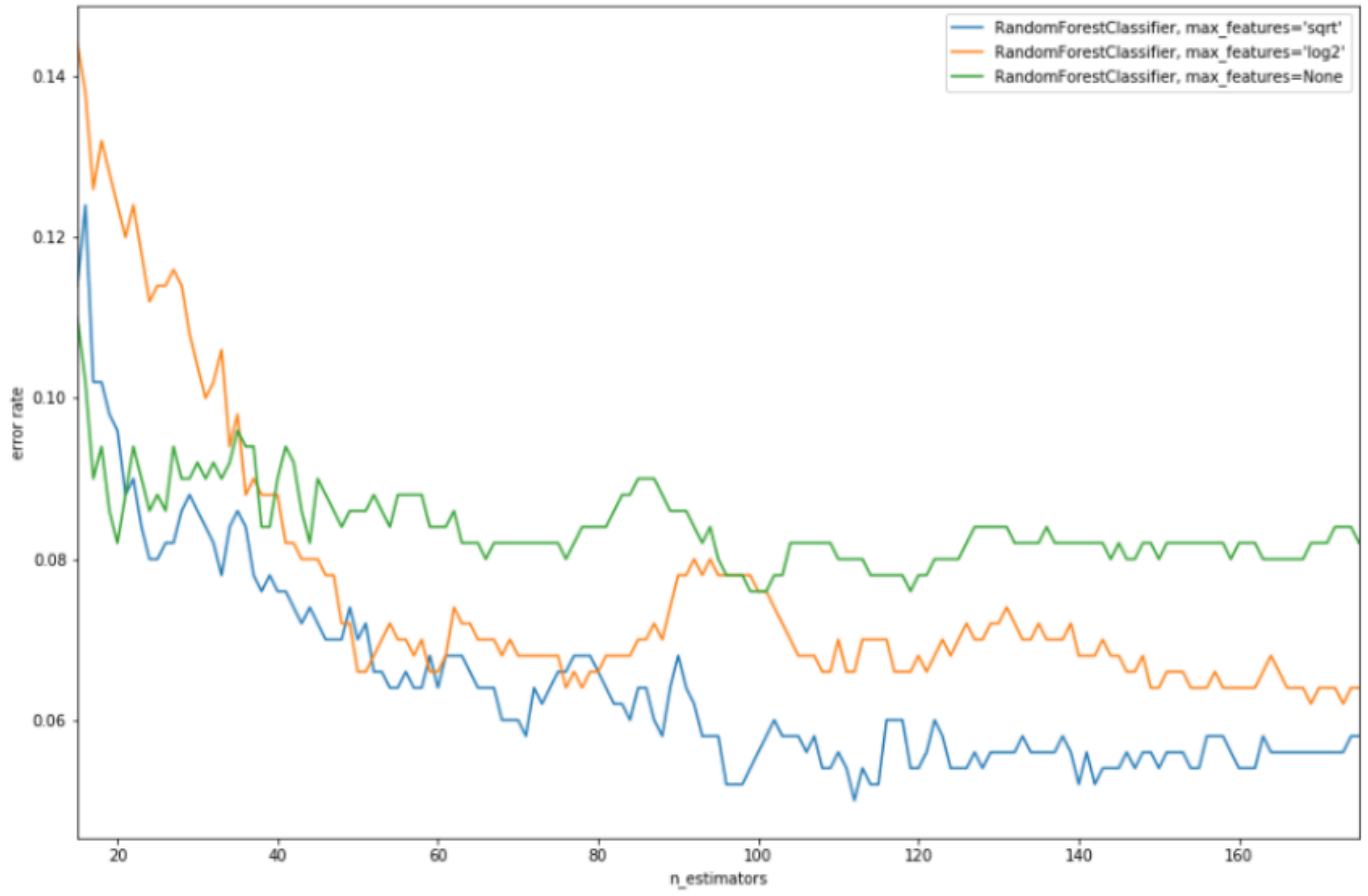
3: Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}_n$:

- дерево строится, пока в каждом листе не окажется не более $n_{\min}$ объектов
- при каждом разбиении сначала выбирается m случайных признаков из p, и оптимальное разделение ищется только среди них

4: Вернуть композицию $a_N(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$

1. Случайный лес **/observation**

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



1. Случайный лес **/out-of-bag**

- 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:

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

1. Случайный лес **/FAQ**

- *Какая должна быть глубина деревьев в случайном лесу?*
    - Используя бэггинг над обычными решающими деревьями, мы не могли добиться некоррелированность моделей
    - Теперь, благодаря Random Forest базовые модели в большей степени отличаются друг от друга
    - Bagging может уменьшить разброс моделей, но не решает проблему смещенности (BVD)
    - Неглубокие деревья имеют меньшую тенденцию к переобучению, с низким разбросом, но высокой смещенностью

    - ***Ответ***: Используем Глубокие деревья

1. Случайный лес **/FAQ**

- *Сколько признаков надо подавать дереву для обучения?*
    - Чем больше признаков, тем больше корреляция между деревьями и тем меньше чувствуется эффект от ансамблирования.
    - Чем меньше признаков, тем слабее сами деревья.

    - ***Ответ***: практический совет:

Рекомендации для q:

- Регрессия: $q = \frac{d}{3}$
- Классификация: $q = \sqrt{d}$

1. Случайный лес **/FAQ**

- *Сколько должно быть деревьев в случайном лесе?*
    - Увеличение числа элементарных алгоритмов в ансамбле не меняет смещения и уменьшает разброс
    - Число признаков и варианты подвыборок, на которых строятся деревья в случайном лесе, ограничены — уменьшать разброс до бесконечности не получится
    - Кстати, можно параллелизацией процесса выиграть на времени работы

    - ***Ответ***: замер значения качества от числа деревьев, выбор в качестве гиперпараметра n_estimators, когда метрика достигла плато

1. Случайный лес **/feature importance**

- 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

1. Случайный лес **/similarity metric**

- *Предсказания строится на основе меток похожих объектов из обучения. Докажем:*
    - Пусть в регрессии $T_n(x)$ - номер листа n-ого дерева из Random Forest, в который попал объект *x*
    - Ответ дерева n на объекте *x* будет равен среднему ответу объектов обучающей выборки в этом листе:

$$b_n(x) = \sum_i w_n(x, x_i) y_i = \sum_i \frac{[T_n(x) = T_n(x_i)]}{\sum_j [T_n(x) = T_n(x_j)]} y_i$$

    - Ответ Random Forest - сумма ответов всех объектов train с весами схожести каждого с искомым *x*:

$$a_N(x) = \sum_i \left(\frac{1}{N} \sum_n^N w_n(x, x_i)\right) \cdot y_i$$

1. Случайный лес **/resume**

- Случайный лес — метод на основе бэггинга, в котором делается попытка повысить разнообразие деревьев

- Метод практически без гиперпараметров

- Можно оценить обобщающую способность без тестовой выборки

Gradient Boosting