AI

ITAM
AI Lab MISIS

Лекция 3

Решающие деревья, ансамбли, Random Forest, бэггинг, бустинг, стекинг

План лекции

Решающие деревья 🌴

- Введение. Пример решающего дерева
- Определение решающего дерева
- Почему построение оптимального решающего дерева — сложная задача?
- Жадный алгоритм построения решающего дерева
- Особенности данных
- Методы регуляризации решающих деревьев

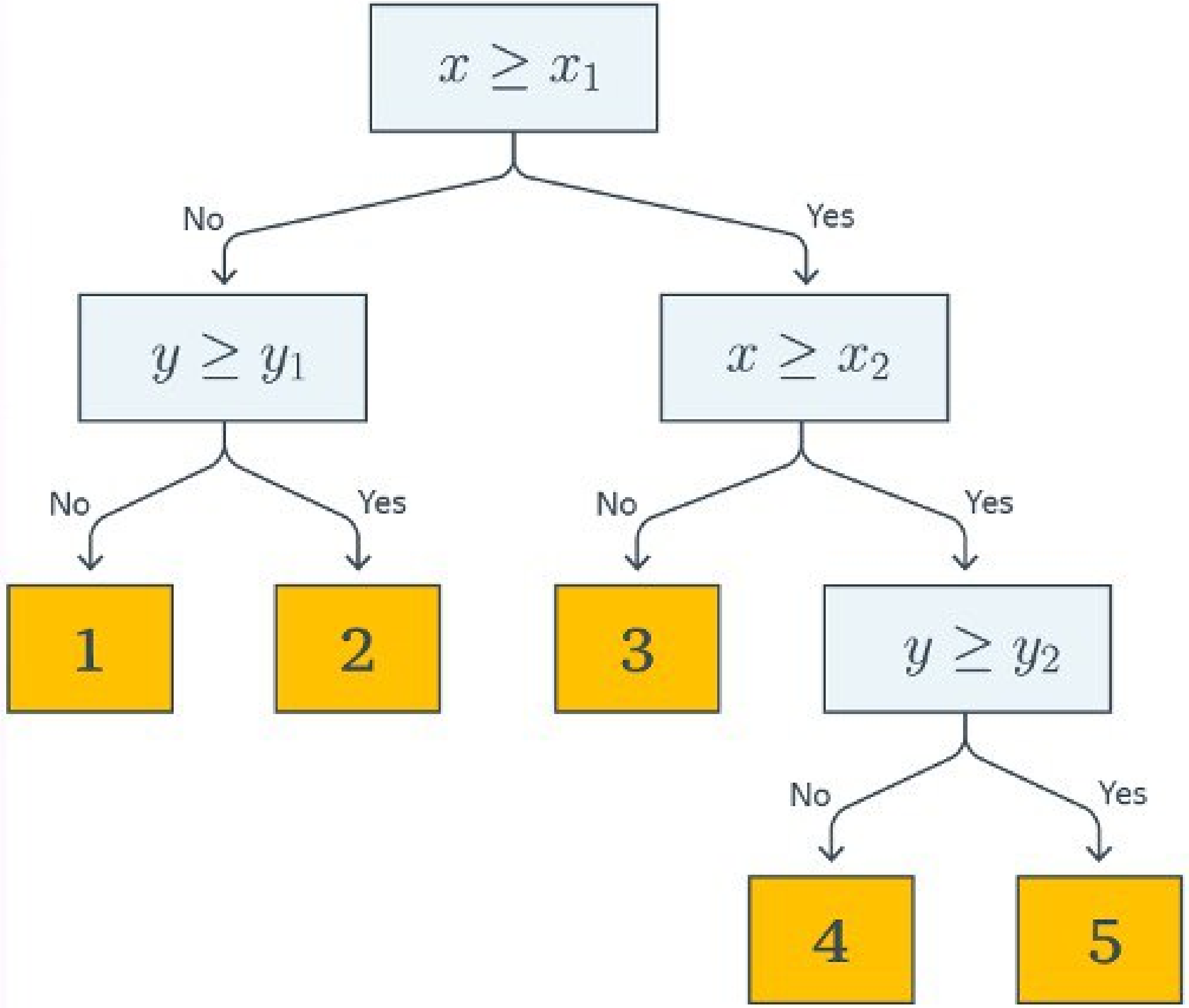
Ансамбли в машинном обучении 🌲🌳🌴

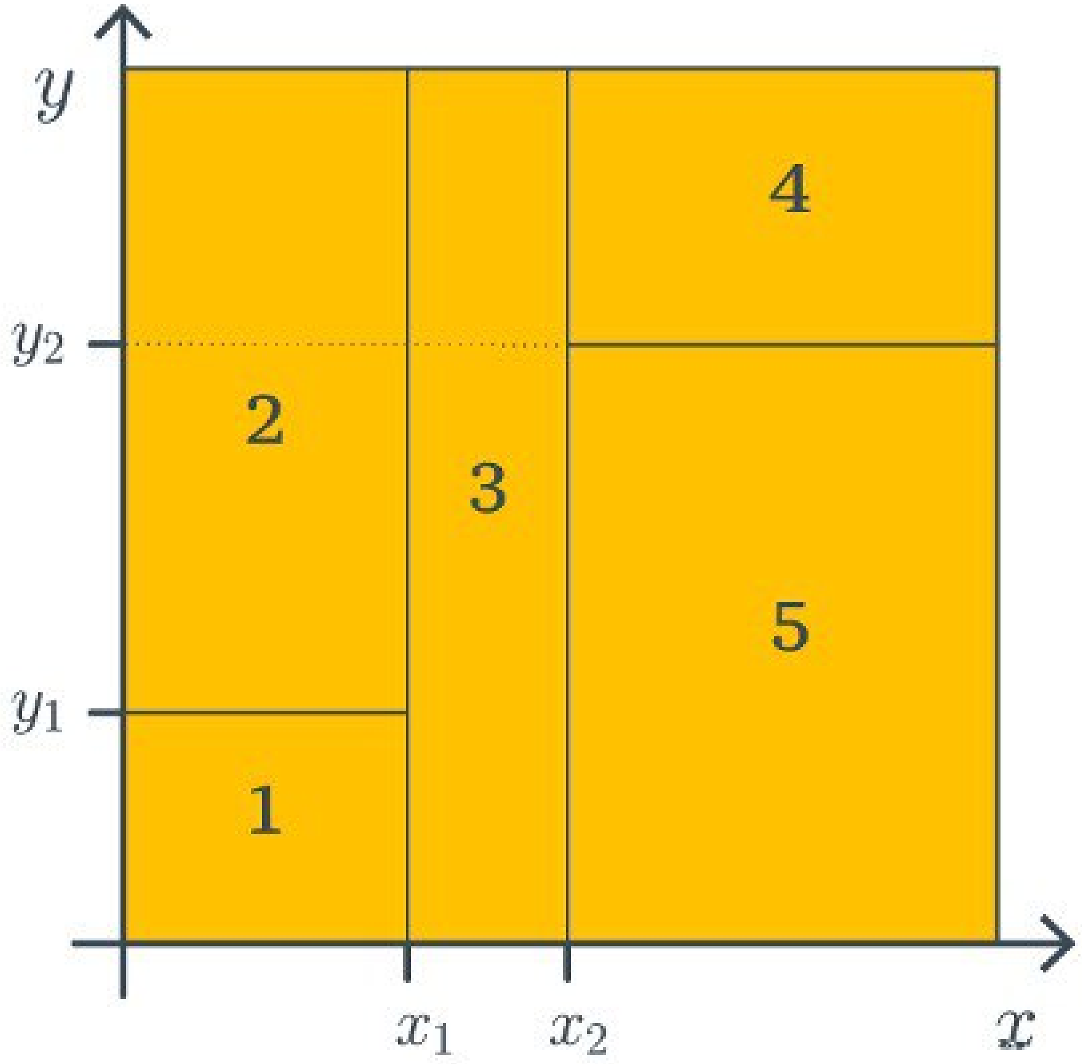
- Смещение и разброс
- Бэггинг. Бэггинг над решающими деревьями
- Random Forest
- Бустинг
- Стекинг

Decision trees

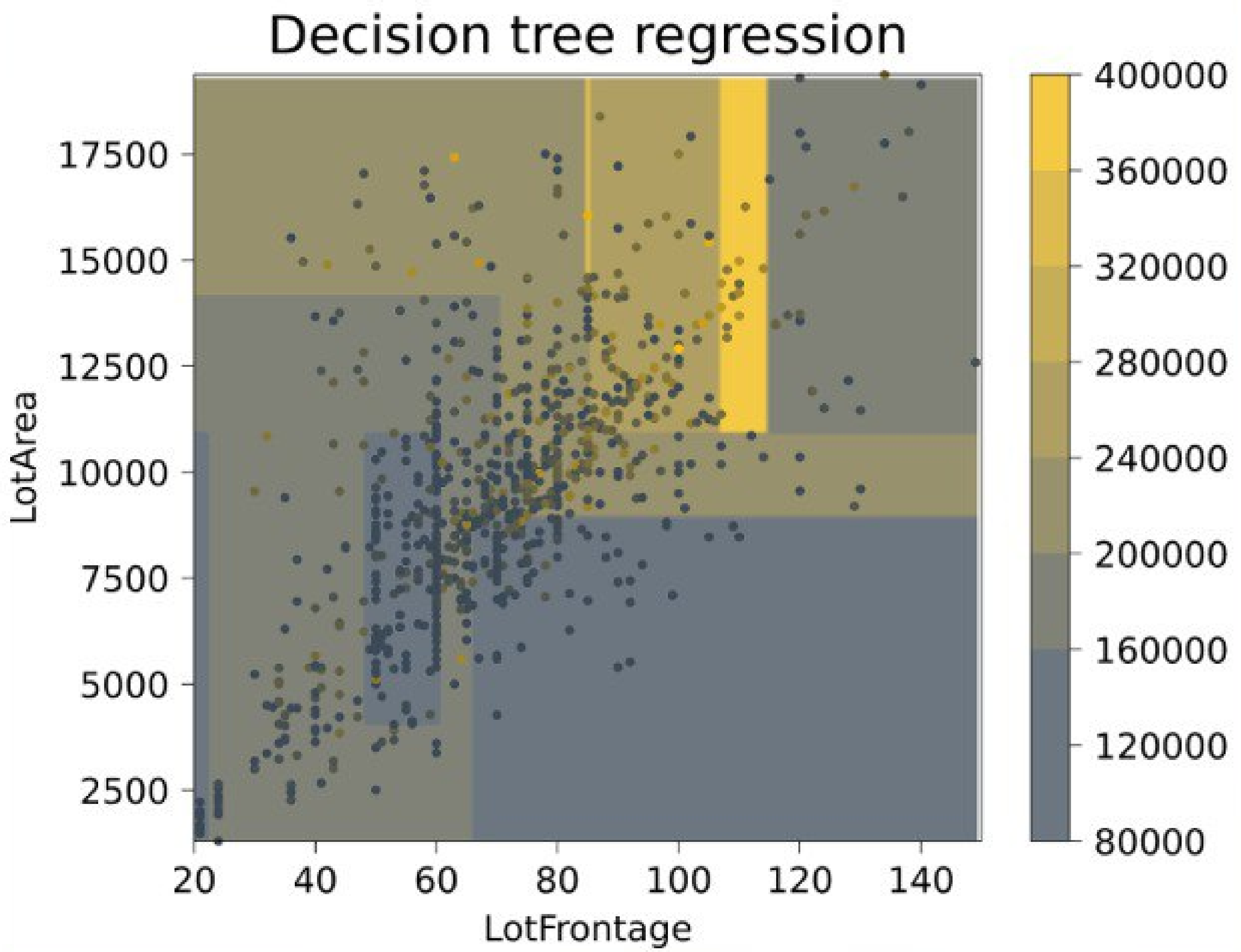
AI Lab MISIS

Решение о том, к какому классу будет отнесён текущий объект выборки, будет приниматься с помощью прохода от корня дерева к некоторому листу.





Decision trees



Дерево глубины 4, построенное для объектов данных из **Ames Hounsing Dataset** (признаковое описание объектов недвижимости)

Фичи:

- ширина фасада (Lot_Frontage)
- площадь (Lot_Area)

Предсказать:

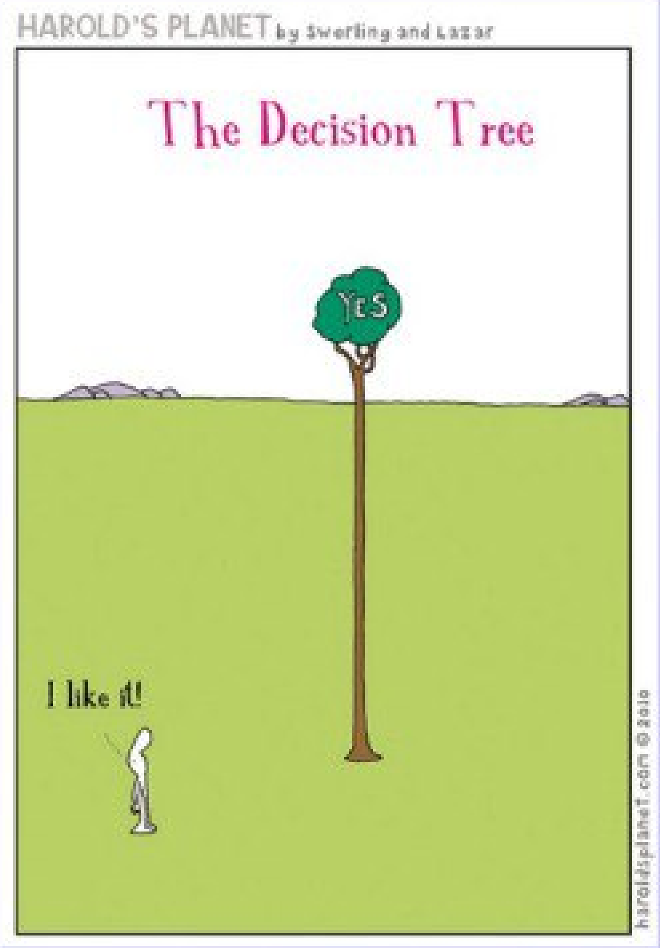
- стоимость недвижимости ?

Decision trees

AI Lab MISIS

Пусть задано **бинарное дерево**, в котором:

- каждой внутренней вершине v приписан предикат $B_v : \mathbb{X} \to \{0, 1\}$
- каждой листовой вершине v приписан прогноз $c_v \in \mathbb{Y}$ где $\mathbb{Y}$ — область значений целевой переменной (в случае классификации листу может быть также приписан вектор вероятностей классов).

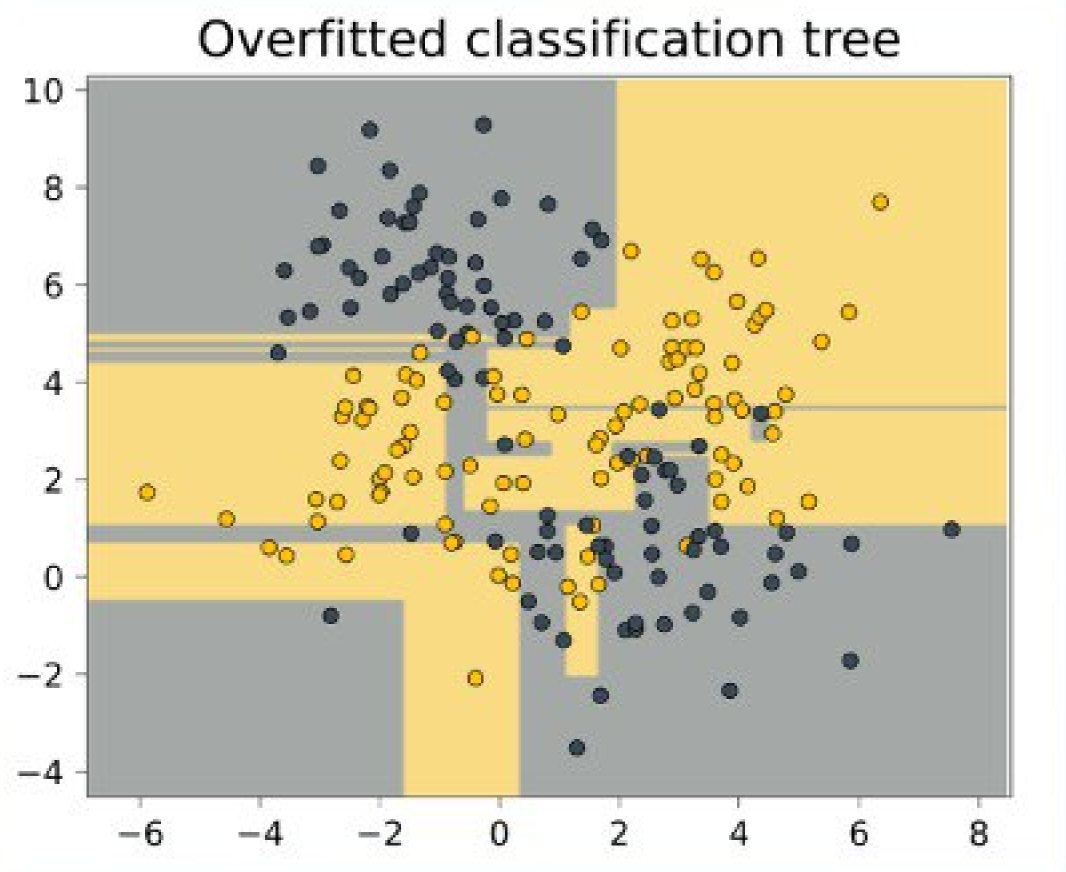


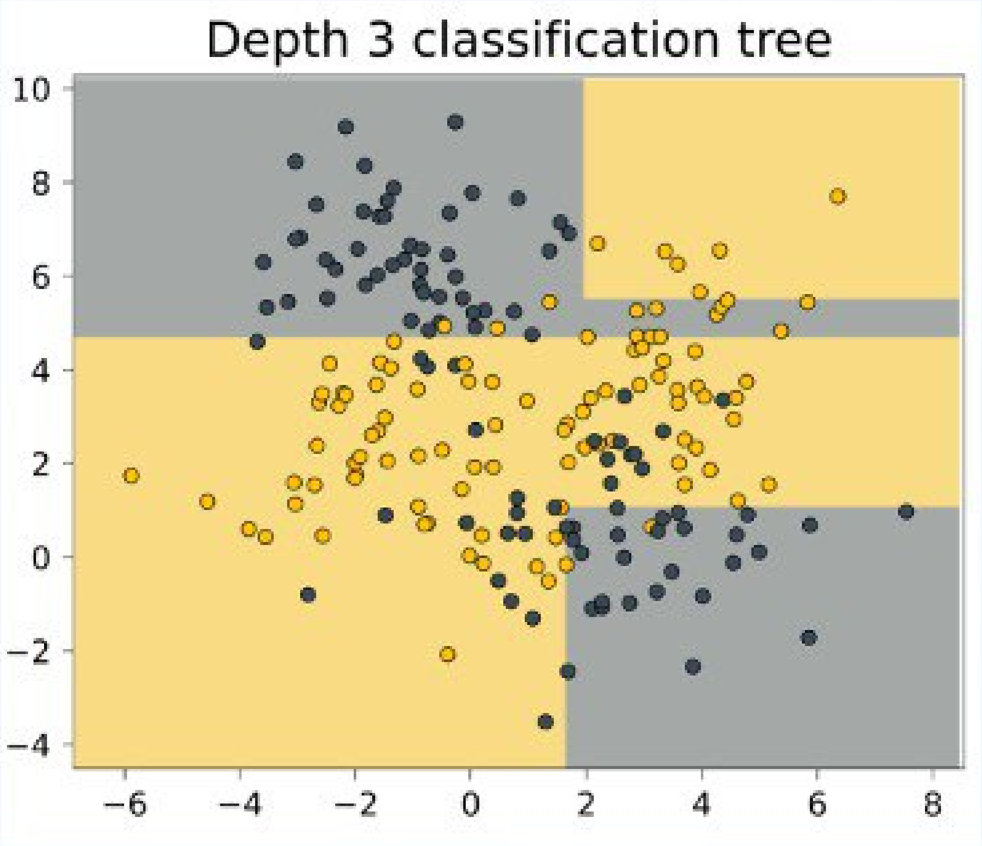
Предикат B_v может иметь, вообще говоря, произвольную структуру, но, как правило, на практике используют просто сравнение с порогом $t \in \mathbb{R}$ по какому-то j-му признаку:

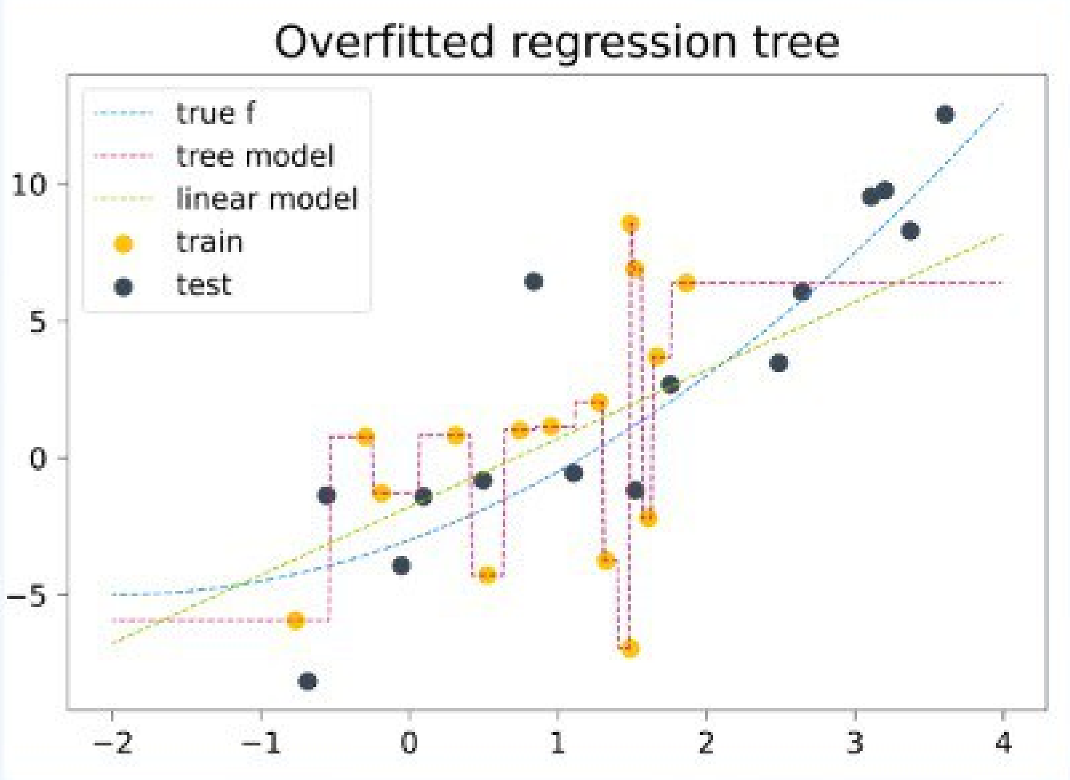
$$B_v(x, j, t) = [x_j \leq t]$$

Decision trees

AI Lab MISIS

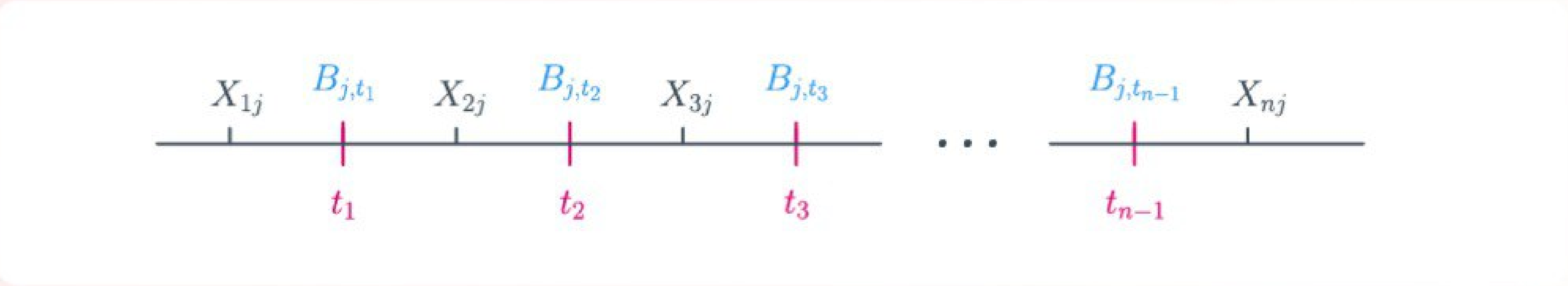






Оптимальное решающее дерево — сложная задача?

Пусть, как обычно, у нас задан датасет (X,y), где $y = \{y_i\}_{i=1}^N \subset \mathbb{R}^N$ — вектор таргетов, а $X = \{x_i\}_{i=1}^N \in \mathbb{R}^{N \times D}, x_i \in \mathbb{R}^D$ — матрица признаков, в которой i-я строка — это вектор признаков i-го объекта выборки. Пусть у нас также задана функция потерь L(f,X,y), которую мы бы хотели минимизировать.



$$(j_{opt}, t_{opt}) = \arg\min_{j,t} L(B_{j,t}, X, y)$$

Жадный алгоритм построения решающего дерева

Пусть X — исходное множество объектов обучающей выборки, а X_m — множество объектов, попавших в текущий лист (в самом начале $X_m = X$). Тогда жадный алгоритм можно верхнеуровнево описать следующим образом:

1. Создаём вершину v.
2. Если выполнен критерий остановки $Stop(X_m)$, то останавливаемся, объявляем эту вершину листом и ставим ей в соответствие ответ $Ans(X_m)$, после чего возвращаем её.
3. Иначе: находим предикат (иногда ещё говорят сплит) $B_{j,t}$, который определит наилучшее разбиение текущего множества объектов X_m на две подвыборки X_ℓ и X_r, максимизируя критерий ветвления $Branch(X_m, j, t)$.
4. Для X_ℓ и X_r рекурсивно повторим процедуру.

Стратегии сплита

Ответы дерева будем кодировать так: $c \in \mathbb{R}$ — для ответов регрессии и меток класса; для случаев, когда надо ответить дискретным распределением на классах, $c \in \mathbb{R}^K$ будет вектором вероятностей:

$$c = (c_1, \ldots, c_K), \quad \sum_{i=1}^{K} c_i = 1$$

Минимизируем среднее значение функции потерь: $$\frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} L(y_i, c)$$

Информативностью (impurity) - чем она ниже, тем лучше объекты в листе можно приблизить константным значением: $$H(X_m) = \min_{c \in Y} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} L(y_i, c)$$

Похожим образом можно определить информативность **решающего пня:**

$$\frac{1}{|X_m|} \left(\sum_{x_i \in X_l} L(y_i, c_l) + \sum_{x_i \in X_r} L(y_i, c_r) \right) = \frac{1}{|X_m|} \left(|X_l| \cdot \frac{1}{|X_l|} \sum_{x_i \in X_l} L(y_i, c_l) + |X_r| \cdot \frac{1}{|X_r|} \sum_{x_i \in X_r} L(y_i, c_r) \right)$$

$$= \frac{|X_l|}{|X_m|} H(X_l) + \frac{|X_r|}{|X_m|} H(X_r) \qquad => Branch(X_m, j, t) = H(X_m) - \frac{|X_l|}{|X_m|} H(X_l) - \frac{|X_r|}{|X_m|} H(X_r)$$

Информативность в задаче регрессии: MSE

Регрессия с минимизацией среднеквадратичной ошибки: $L(y_i, c) = (y_i - c)^2$

Информативность листа будет выглядеть следующим образом: $H(X_m) = \frac{1}{|X_m|} \min_{c \in Y} \sum_{(x_i, y_i) \in X_m} (y_i - c)^2$

Оптимальным предсказанием константного классификатора для задачи минимизации MSE является среднее значение, то есть:

$$c = \frac{\sum y_i}{|X_m|}$$

Подставив в формулу информативности сплита, получаем:

$$H(X_m) = \sum_{(x_i, y_i) \in X_m} \frac{(y_i - \bar{y})^2}{|X_m|}, \text{ где } \bar{y} = \frac{1}{|X_m|} \sum_i y_i$$

Информативность в задаче регрессии: MAE

$$L(y_i, c) = |y_i - c|$$

Случай средней абсолютной ошибки так же прост: в листе надо предсказывать медиану, ведь именно медиана таргетов для обучающих примеров минимизирует MAE констатного предсказателя

Информативность - абсолютное отклонение от медианы:

$$H(X_m) = \sum_{(x_i, y_i) \in X_m} \frac{|y_i - MEDIAN(Y)|}{|X_m|}$$

Критерии ветвления

Пусть есть область R

в ней доли объектов всех классов: $p_1, \ldots, p_l$

Missclassification criteria

$$H(R) = 1 - p_{\max}$$

энтропийный

$$H(R) = -\sum_j p_j \log_2 p_j$$

Джини

$$H(R) = \sum_j p_j (1 - p_j) = 1 - \sum_j p_j^2$$

Мера неоднородности (impurity) минимальна (=0)
только если все объекты принадлежат одному классу

Для двух классов

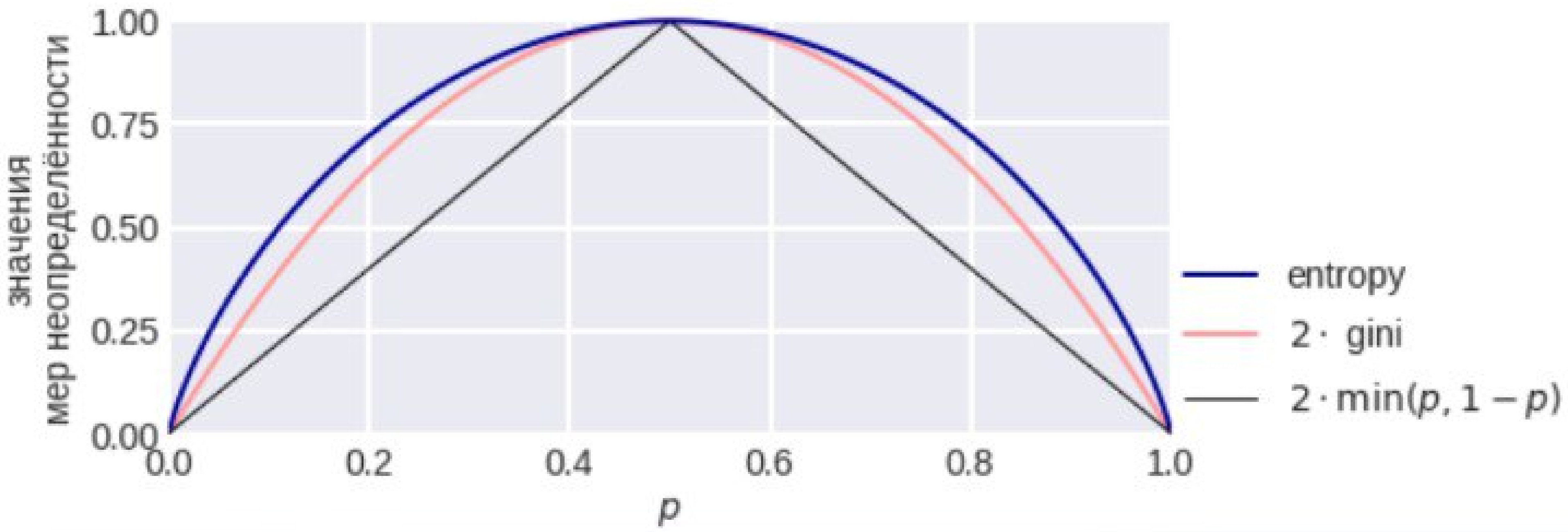
Пусть есть область R

в ней доли объектов всех классов: $p_1 = p, p_2 = 1 - p$

Missclassification criteria $H(R) = \min[p, 1-p]$

энтропийный $H(R) = -p\log_2 p - (1-p)\log_2(1-p)$

Джини $H(R) = 2p(1-p) = 1 - p^2 - (1-p)^2$



Методы регуляризации решающих деревьев

- Ограничение по максимальной глубине дерева;
- Ограничение на минимальное количество объектов в листе;
- Ограничение на максимальное количество листьев в дереве;
- Требование, чтобы функционал качества **Branch** при делении текущей подвыборки на две улучшался не менее чем на **s** процентов.

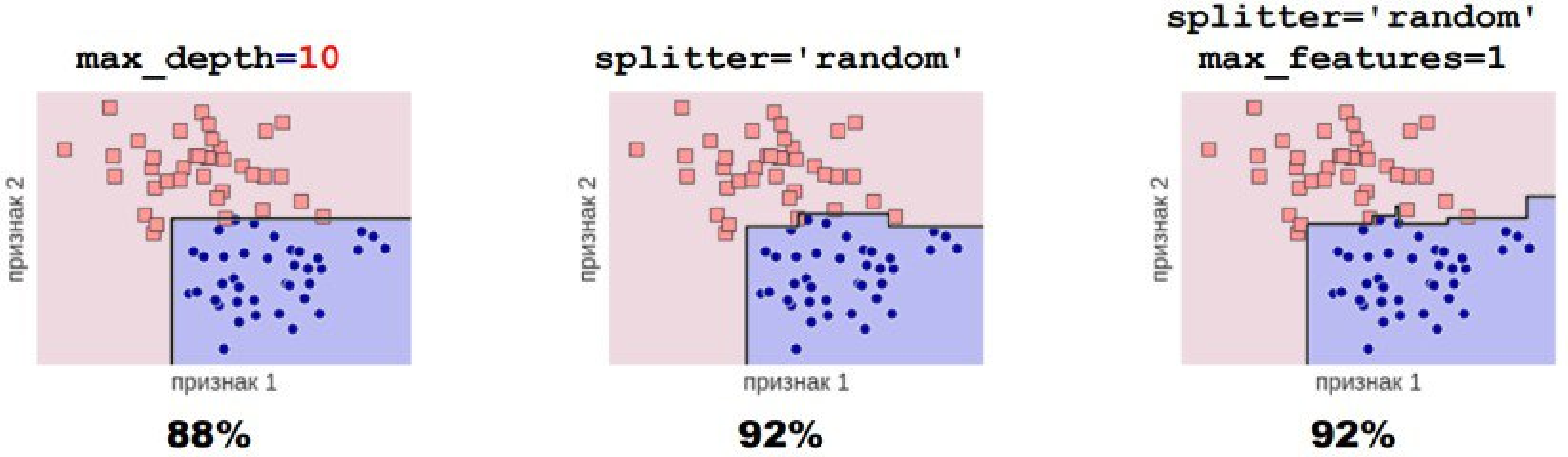
1. Можно проверять критерии прямо во время построения дерева, такой способ называется **pre-pruning** или **early stopping**;
2. А можно построить дерево жадно без ограничений, а затем провести **стрижку (pruning)**, то есть удалить некоторые вершины из дерева так, чтобы итоговое качество упало не сильно, но дерево начало подходить под условия регуляризации. При этом качество стоит измерять на отдельной, отложенной выборке.

Деревья в sklearn

```python
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(criterion='gini', splitter='best',
                               max_depth=None, min_samples_split=2,
                               min_samples_leaf=1,
                               min_weight_fraction_leaf=0.0,
                               max_features=None, random_state=3,
                               max_leaf_nodes=None,
                               min_impurity_decrease=0.0,
                               min_impurity_split=None, class_weight=None,
                               presort=False)
model.fit(X, y)
```

```python
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(max_depth=10)
tree.fit(X_train, y_train)
```

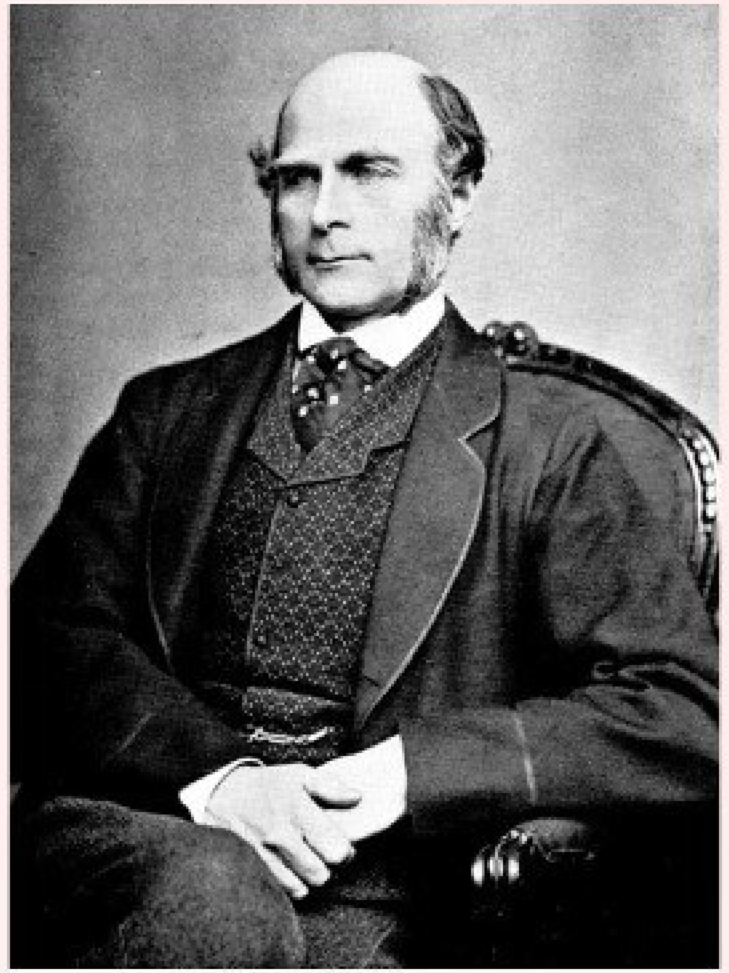


Реализация в scikit-learn

```
sklearn.tree import DecisionTreeClassifier
```

criterion	критерий расщепления «gini» / «entropy»
splitter	разбиение «best» / «random»
max_depth	допустимая глубина
min_samples_split	минимальная выборка для разбиения
min_samples_leaf	минимальная мощность листа
min_weight_fraction_leaf	аналогично с весом
max_features	число признаков, которые смотрим для нахождения разбиения
random_state	инициализация генератора случайных чисел
max_leaf_nodes	допустимое число листьев
min_impurity_decrease	порог изменения «зашумлённости» для разбиения
min_impurity_split	порог «зашумлённости» для останова
class_weight	веса классов («balanced» или словарь, список словарей)

Ансамбли в ML



- **Метод ближайших соседей**
- **Линейные алгоритмы**
- **Решающее дерево**

Идея: построение композиции алгоритмов

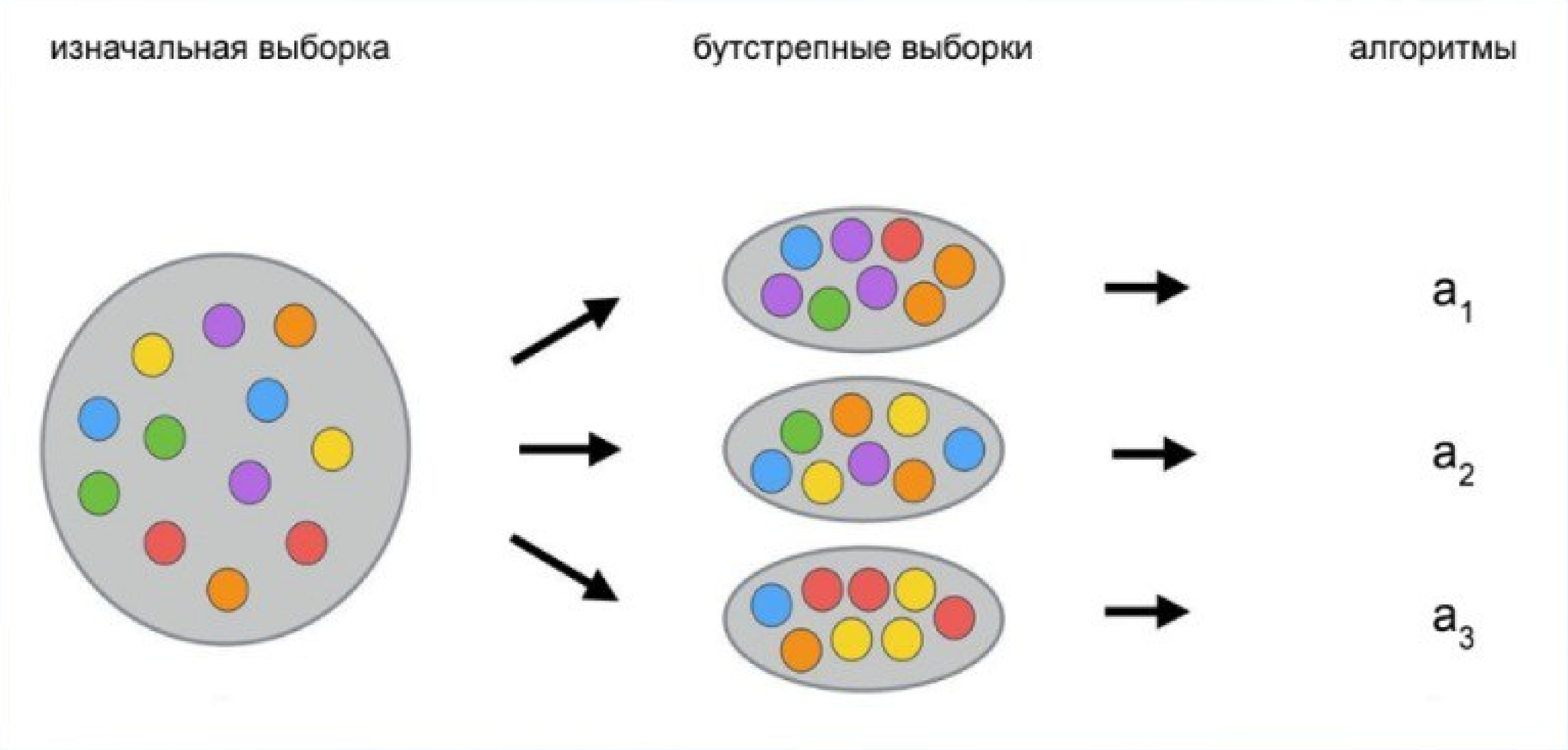
Эксперимент Гальтона:

- Собралось около 800 человек, которые попытались угадать вес быка на ярмарке. Бык весил 1198 фунтов.
- Ни один крестьянин не угадал точный вес быка
- Среднее предсказания оказалось равным 1197 фунтов

Метод простого голосования

- a1, a2, ..., an - несколько обученных алгоритмов
- **Классификация:** относим x к классу, за который проголосовало большинство из a1(x), a2(x), ..., an(x)
- **Регрессия:** ответом является среднее значение a1(x), a2(x), ..., an(x)

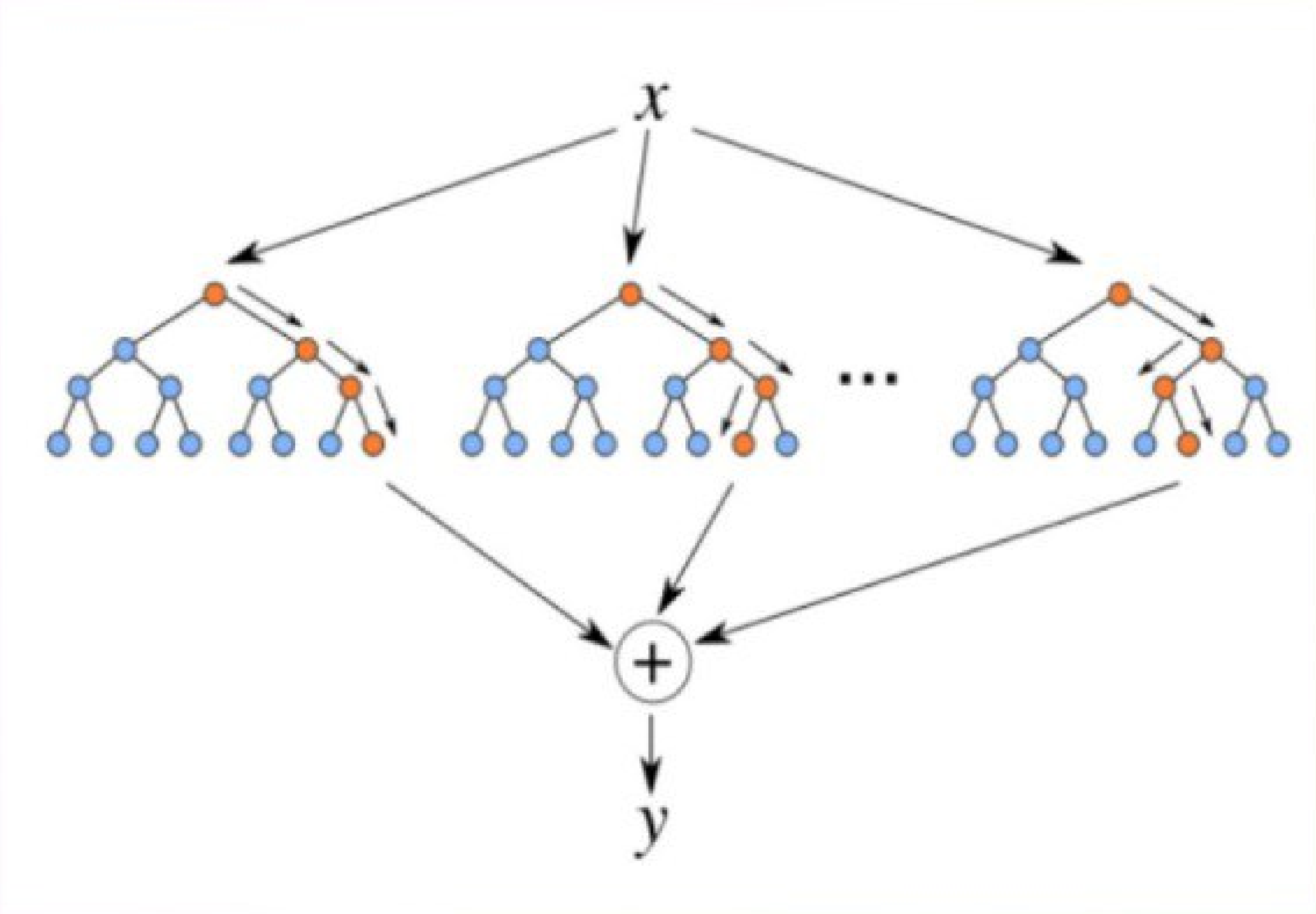
Бутстреп



Бэггинг

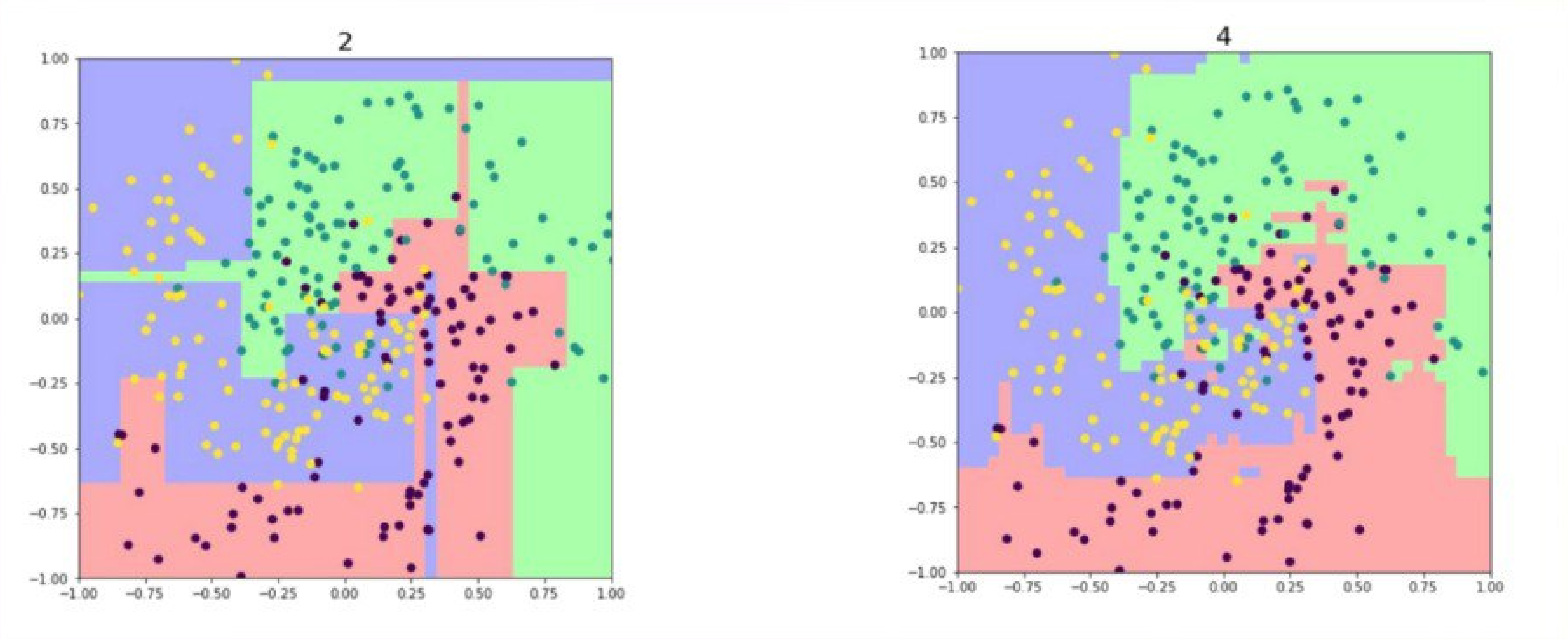
Бэггинг (bagging: bootstrap aggregation) — принцип построения композиции, основанный на простом голосовании

- 100 деревьев
- Бутстрепная выборка для каждого дерева
- Финальное решение принимается простым голосованием



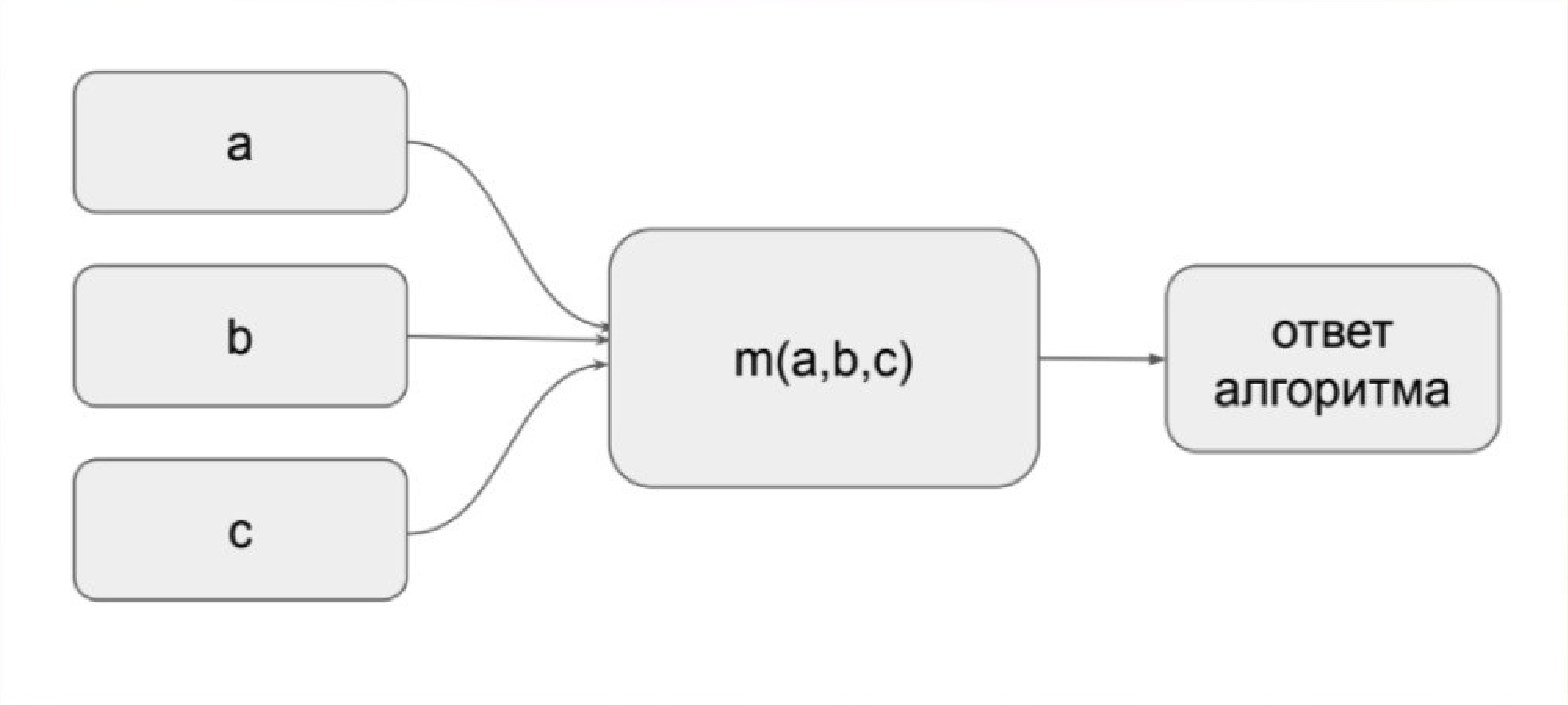
Случайный лес и решающее дерево

AI Lab MISIS

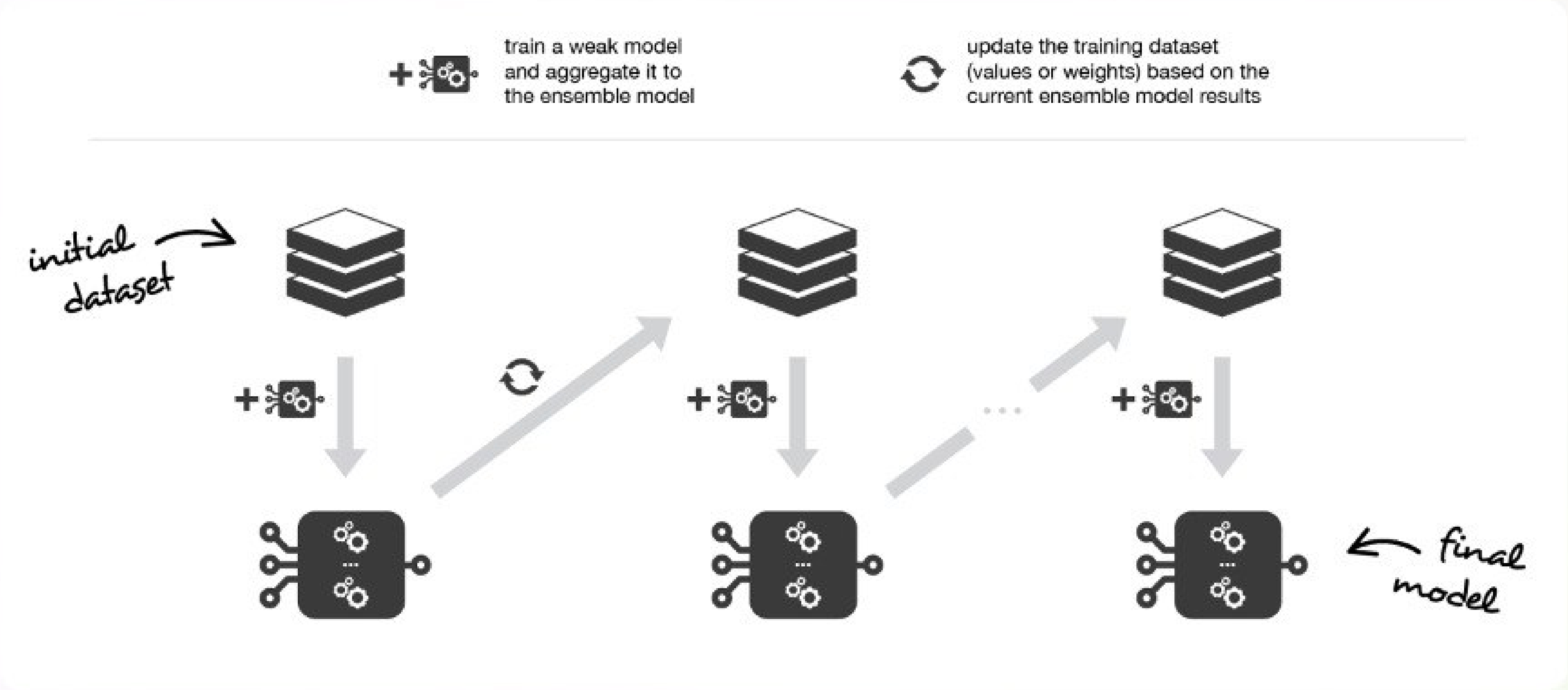


- Слишком долгое и громоздкое вычисление
- В индустрии стараются обойтись без композиций
- Тем не менее, вычисления можно проводить параллельно

Стекинг



Бустинг



- Строим алгоритмы последовательно
- Каждый следующий алгоритм компенсирует ошибку всех предыдущих
- Принимаем решение взвешенным голосованием:
  a(x) = c1a1(x) + c2a2(x) + ... + cn*an(x)
- Сильный метод бустинга — градиентный бустинг над решающими деревьями

Подводя итог

- Бэггинг, стекинг и бустинг используют принцип композиции
- Бэггинг принимает решение простым голосованием
- Стекинг обучает метаалгоритм над разноплановыми алгоритмами
- Бустинг строит базовые модели, компенсирующие ошибки предыдущих

Источники

- Учебник по ML от ШАД, глава decision trees (https://ml-handbook.ru/)
- NP-полнота в задачах обучающих деревьев (https://people.csail.mit.edu/rivest/HyafilRivest-ConstructingOptimalBinaryDecisionTreesIsNPComplete.pdf)
- Решающие деревья, Хабр (https://habr.com/ru/company/ods/blog/322534/)
- Deep Learning School, курс по машинному обучению (https://stepik.org/course/124069/)
- Учебник по ML от ШАД, глава ensembles (https://ml-handbook.ru/)