

Машинное обучение

Лекция 8
Градиентный бустинг

Михаил Гущин
mhushchyn@hse.ru

НИУ ВШЭ, 2025



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Эволюция решающих деревьев



Решающие
деревья

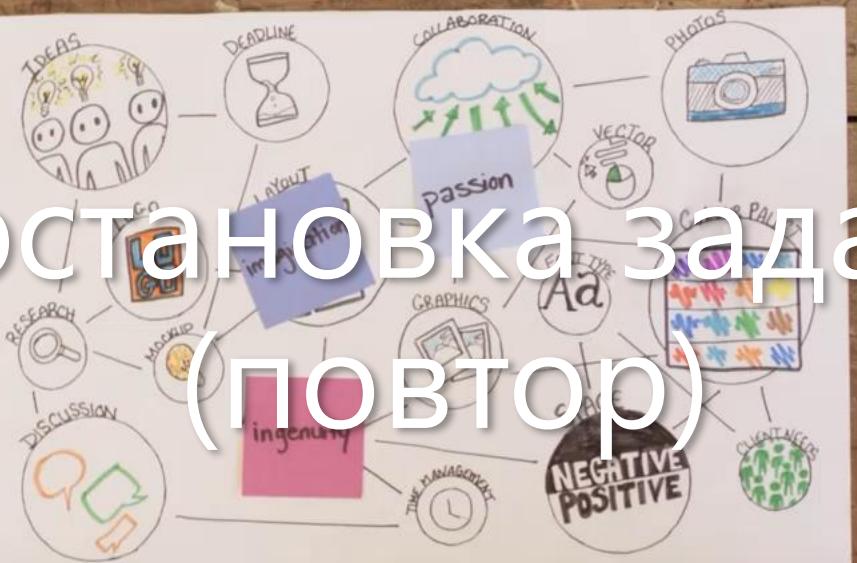


Случайный лес
решающих
деревьев

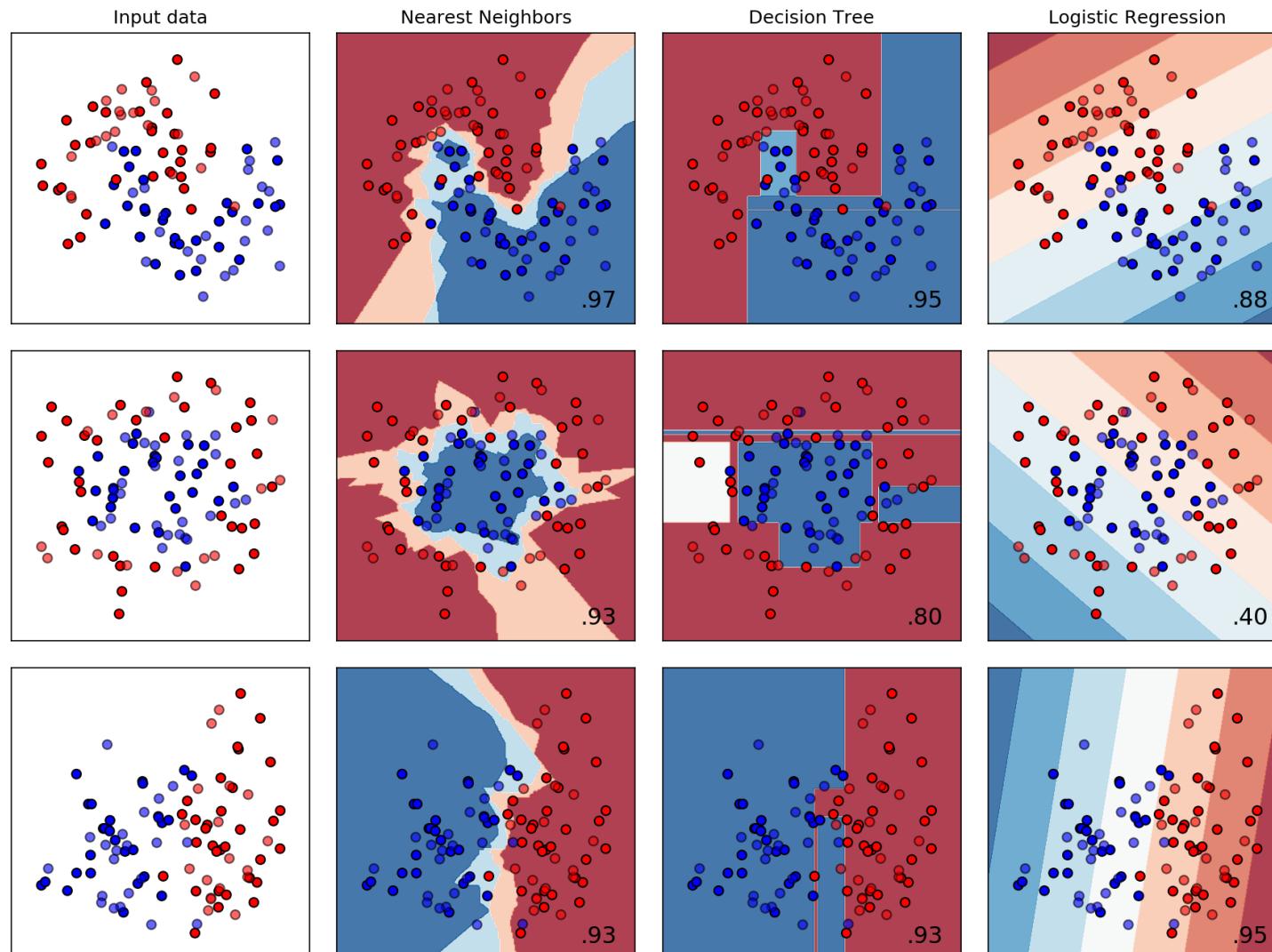


Градиентный
бустинг

Постановка задачи (повтор)



Классификаторы



Задача

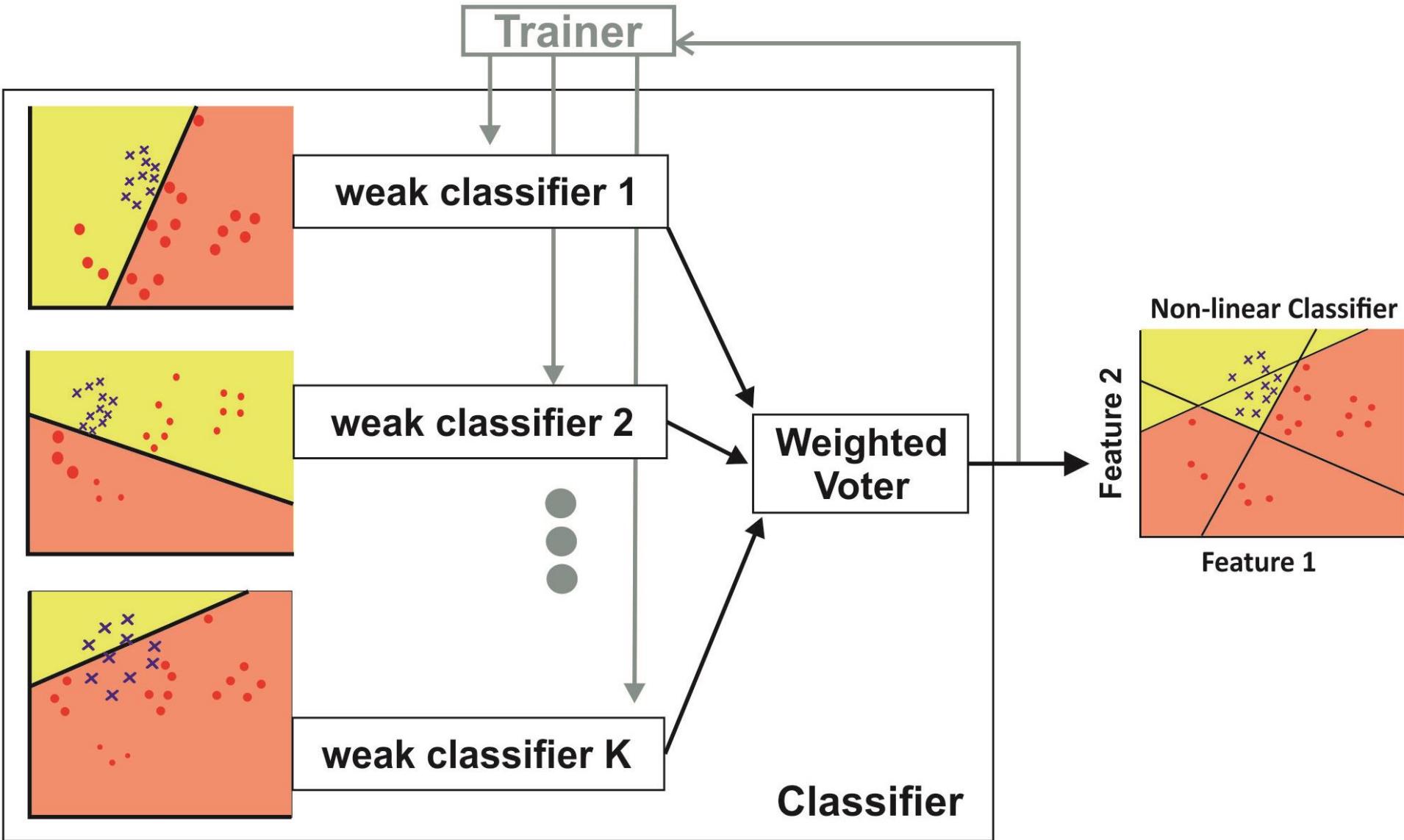
► **Нам дано:**

- Набор данных X, y
- Несколько классификаторов

► **Наши задачи:**

- Как улучшить прогнозы классификаторов?
- Можно ли объединить разные классификаторы в одну модель?

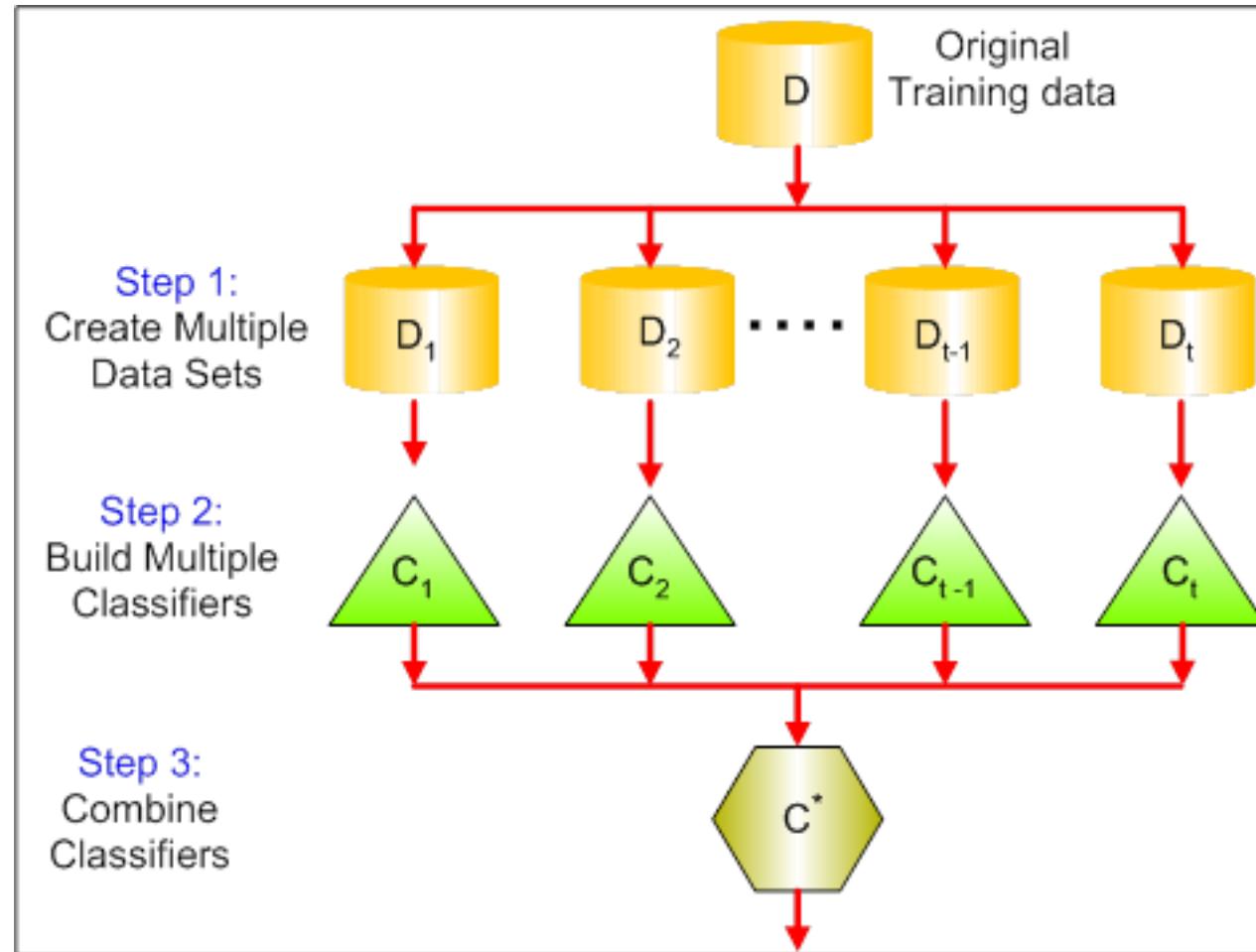
Пример



The background of the slide features a dense, repeating pattern of traditional East Asian motifs. It consists of large, stylized flowers with eight petals and intricate centers, set within rounded square frames. These larger flowers are interspersed with smaller, sun-like shapes with multiple rays and decorative scrollwork. The entire pattern is rendered in a high-contrast black and white color scheme.

БЭГИНГ
(повтор)

Бэггинг (bagging)

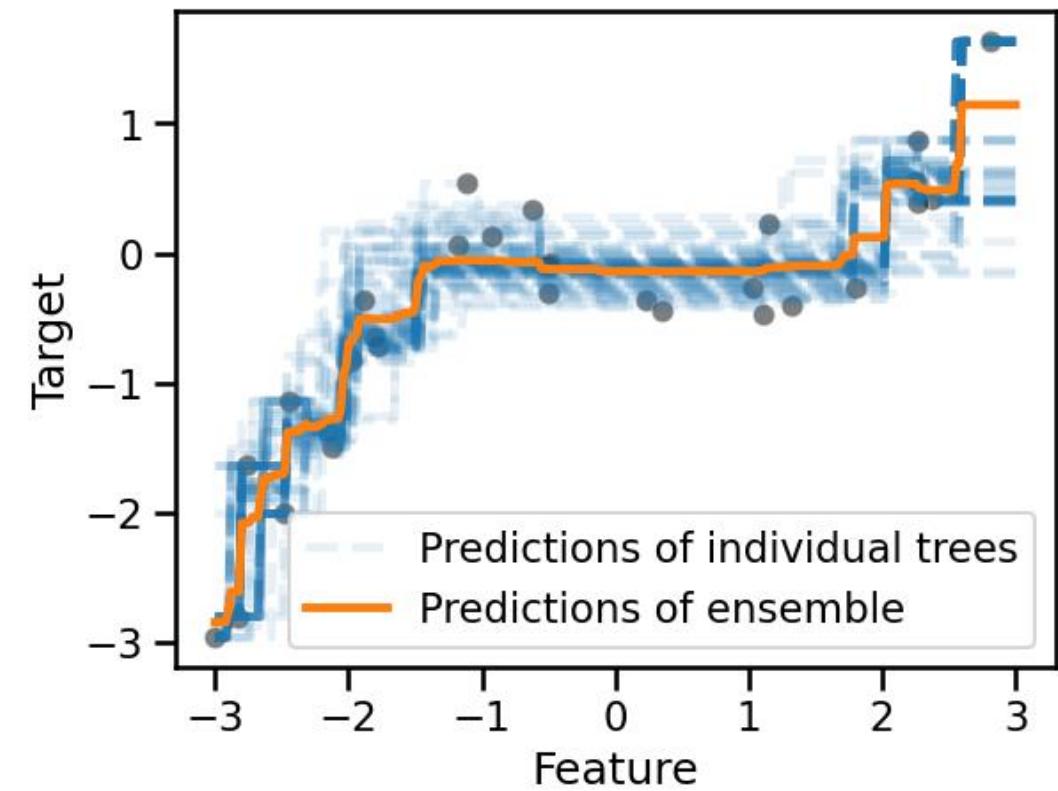
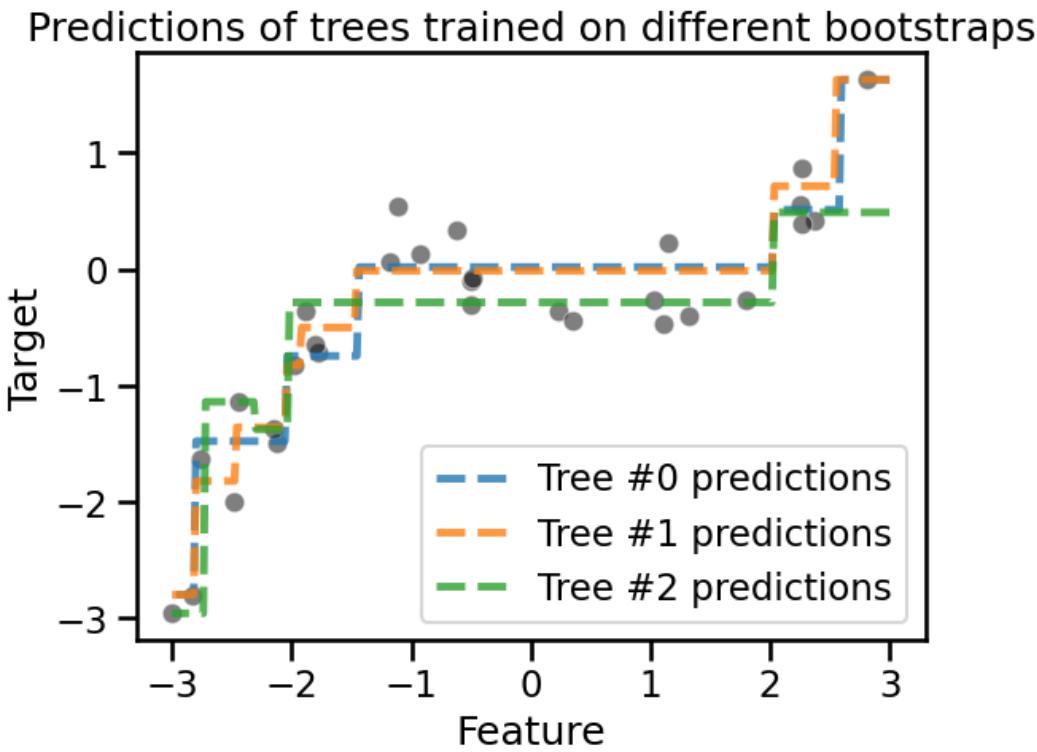


Алгоритм бэггинга

- ▶ Даны выборка данных X, y
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем модель классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Пример



Источник: https://inria.github.io/scikit-learn-mooc/python_scripts/ensemble_bagging.html

Градиентный спуск (повтор)

Повтор

- ▶ Модель логистической регрессии:

$$\hat{y}_i = \sigma(x_i^T w)$$

- ▶ Функция потерь log-loss:

$$L = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

- ▶ Мы хотим минимизировать L :

$$L \rightarrow \min_w$$

- ▶ Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \eta \frac{\partial L(w^{(k)})}{\partial w}$$

Несколько шагов спуска для весов

$$w^{(1)} = w^{(0)} - \eta \frac{\partial L(w^{(0)})}{\partial w}$$

$$w^{(2)} = w^{(1)} - \eta \frac{\partial L(w^{(1)})}{\partial w}$$

$$w^{(3)} = w^{(2)} - \eta \frac{\partial L(w^{(2)})}{\partial w}$$

Можем объединить их так:

$$w^{(3)} = w^{(0)} - \eta \frac{\partial L(w^{(0)})}{\partial w} - \eta \frac{\partial L(w^{(1)})}{\partial w} - \eta \frac{\partial L(w^{(2)})}{\partial w}$$

Т.е. от начальных весов отнимаем градиенты функции потерь в разных точках

Градиентный спуск для весов

- ▶ Веса модели обновляются в процессе градиентного спуска
- ▶ Веса определяют прогноз модели
- ▶ Как меняется прогноз в процессе градиентного спуска?

Градиентный спуск для прогнозов

- ▶ Модель логистической регрессии:

$$\hat{y}_i = \sigma(x_i^T w)$$

- ▶ Тогда прогноз:

$$\hat{y}_i^{(0)} = \sigma(x_i^T w^{(0)})$$

$$\hat{y}_i^{(1)} = \sigma(x_i^T w^{(1)}), \quad w^{(1)} = w^{(0)} - \eta \frac{\partial L(w^{(0)})}{\partial w}$$

$$\hat{y}_i^{(2)} = \sigma(x_i^T w^{(2)}), \quad w^{(2)} = w^{(1)} - \eta \frac{\partial L(w^{(1)})}{\partial w}$$

$$\hat{y}_i^{(3)} = \sigma(x_i^T w^{(3)}), \quad w^{(3)} = w^{(2)} - \eta \frac{\partial L(w^{(2)})}{\partial w}$$

- ▶ Может записать изменение прогнозов как-то иначе?

Градиентный спуск для прогнозов

$$\hat{y}_i^{(1)} = \hat{y}_i^{(0)} - \eta \frac{\partial L(\hat{y}_i^{(0)})}{\partial \hat{y}}$$

$$\hat{y}_i^{(2)} = \hat{y}_i^{(1)} - \eta \frac{\partial L(\hat{y}_i^{(1)})}{\partial \hat{y}}$$

$$\hat{y}_i^{(3)} = \hat{y}_i^{(2)} - \eta \frac{\partial L(\hat{y}_i^{(2)})}{\partial \hat{y}}$$

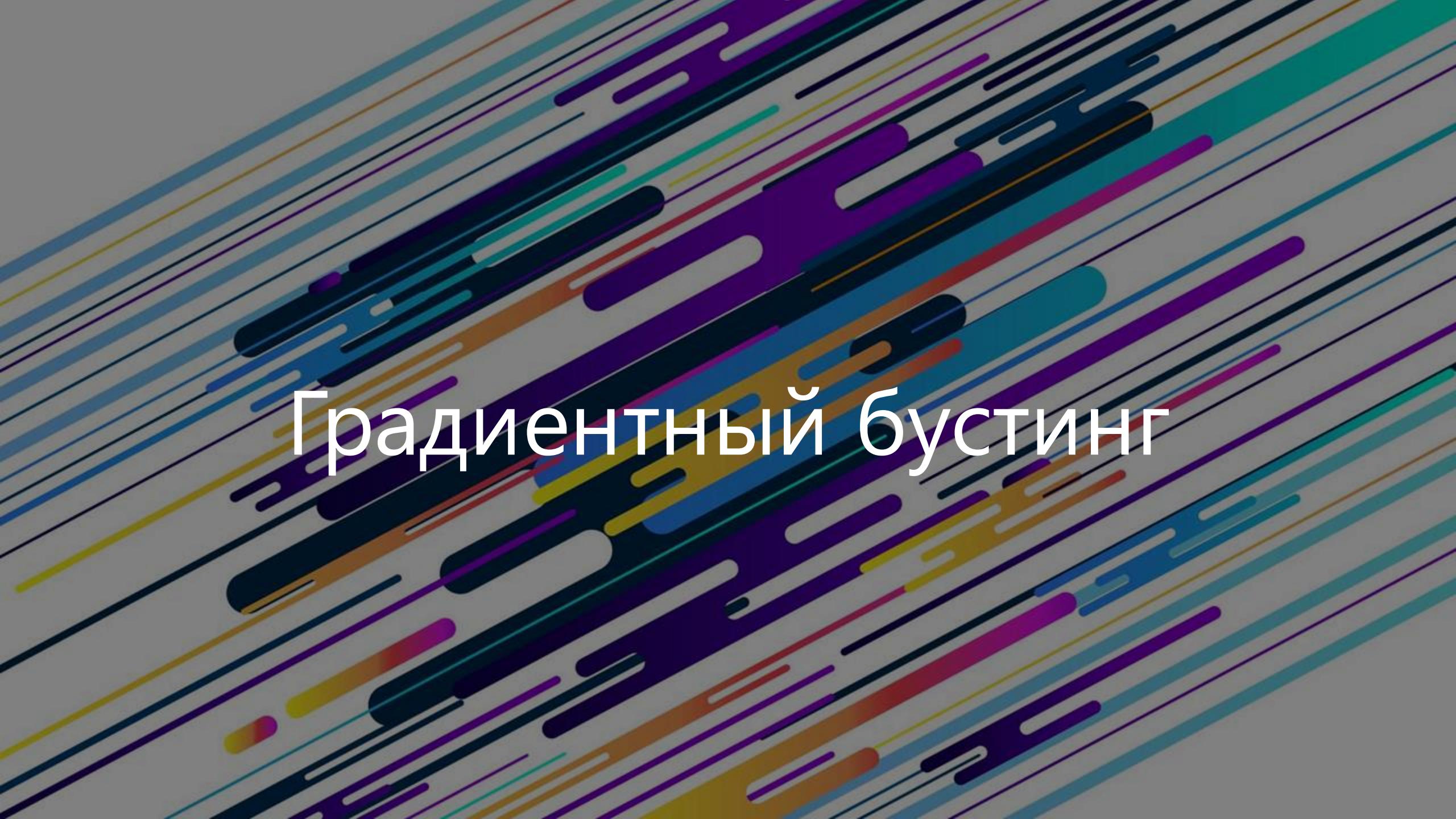
Можем объединить их так:

$$\hat{y}_i^{(3)} = \hat{y}_i^{(0)} - \eta \frac{\partial L(\hat{y}_i^{(0)})}{\partial \hat{y}} - \eta \frac{\partial L(\hat{y}_i^{(1)})}{\partial \hat{y}} - \eta \frac{\partial L(\hat{y}_i^{(2)})}{\partial \hat{y}}$$

От начальных весов отнимаем градиенты функции потерь по прогнозам в разных точках

Градиентный спуск для прогнозов

- ▶ Прогнозы модели зависят от весов
- ▶ Прогнозы так же обновляются в процессе обновления весов
- ▶ Вместо весов мы можем обновлять сразу прогнозы с помощью того же градиентного спуска
- ▶ Только градиенты функции потерь будем брать про прогнозам, а не по весам

The background of the slide features a dynamic pattern of diagonal, elongated, rounded rectangles in various colors, including shades of blue, purple, yellow, and orange, set against a white background.

Градиентный бустинг

Задача

- ▶ Даны выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Будем строить композицию K моделей:

$$a_K(x) = b_0(x) + \sum_{k=1}^K \gamma_k b_k(x)$$

- где $b_0(x)$ – начальный прогноз. Например, константа.
- ▶ Хотим, чтобы $a_K(x)$ минимизировала нашу функцию потерь:

$$L(y, a_K(X)) \rightarrow \min_{\gamma, b}$$

Вопросы

- ▶ Как минимизировать функцию потерь?
- ▶ Как обучать модели в композиции?

Идея решения

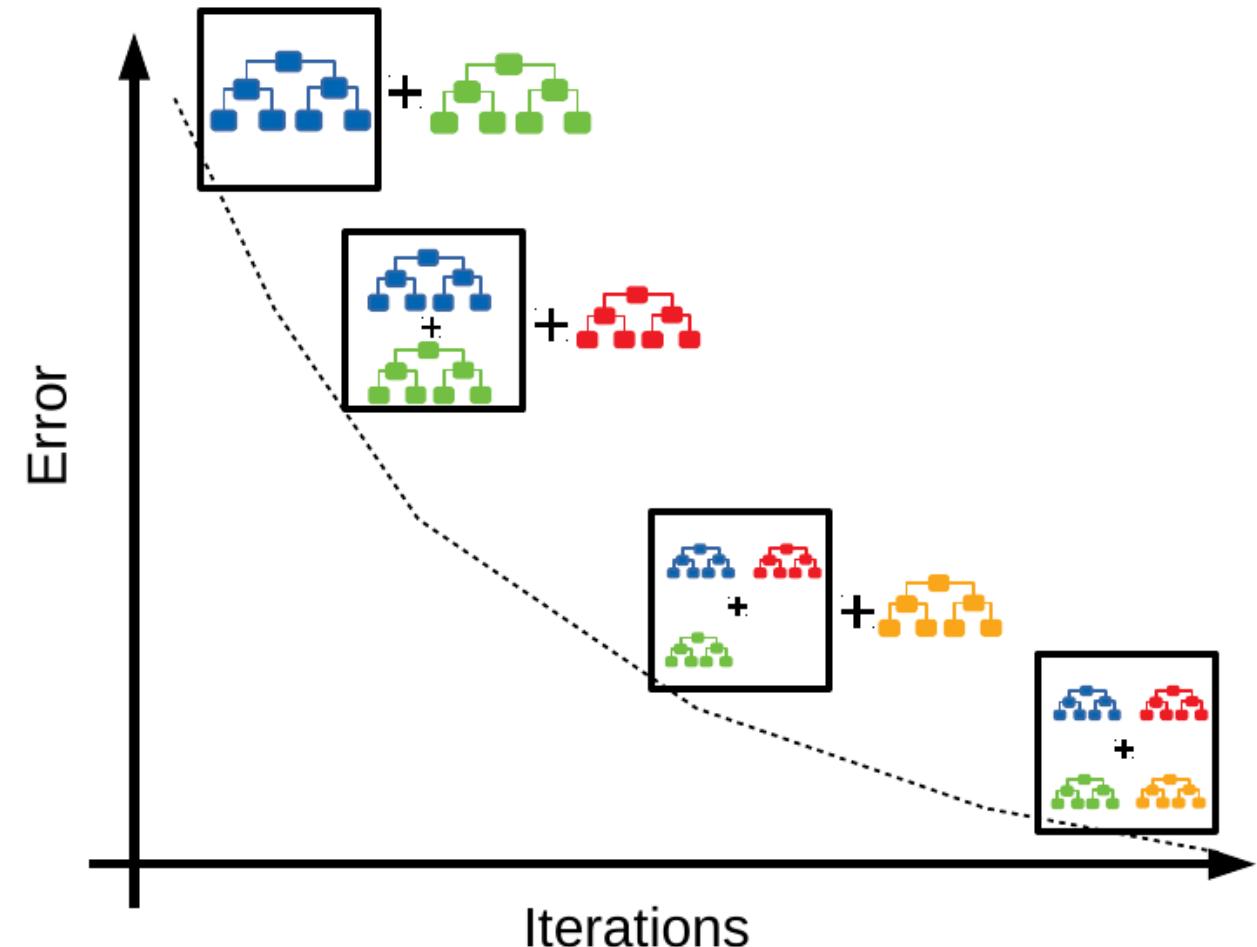
- ▶ Воспользуемся градиентным спуском, только вместо $w^{(k)}$ подставим $a_k(X)$
- ▶ Тогда, для минимизации функции потерь получаем:

$$a_k(X) = a_{k-1}(X) - \eta \frac{\partial L(a_{k-1}(X))}{\partial a}$$

- ▶ Как обучить новую модель $\mathbf{b}_k(X)$, чтобы $a_k(X) = a_{k-1}(X) + \gamma_k \mathbf{b}_k(X)$?

Идея решения

На каждой итерации градиентного спуска добавляем в композицию новую модель, которая учится на ошибках прогноза текущей композиции моделей



Алгоритм градиентного бустинга

- ▶ Даны выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Делаем начальный прогноз $a_0(X) = b_0(X)$
- ▶ Для каждого $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Считаем вектор производных функции потерь (**остатки, сдвиги**):

$$s = -\frac{dL(y^{(k)}, a_{k-1}(X^{(k)}))}{da}$$

- Обучаем модель **регрессии** $b_k(X^{(k)})$ с MSE функцией потерь:

$$(b_k(X^{(k)}) - s)^T (b_k(X^{(k)}) - s) \rightarrow \min_{b_k}$$

- Обновляем композицию алгоритмов:

$$a_k(X^{(k)}) = a_{k-1}(X^{(k)}) + \gamma_k b_k(X^{(k)})$$

Градиентный бустинг для регрессии

- ▶ Функция потерь MSE для задачи регрессии:

$$L(y, a(X)) = \frac{1}{n} (y - a(X))^T (y - a(X)) \rightarrow \min$$

- ▶ Для каждого $k = 1 \dots K$:

- Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Считаем вектор производных функции потерь (**остатки, сдвиги**):

$$s = -\frac{dL(y^{(k)}, a_{k-1}(X^{(k)}))}{da} = -\frac{2}{n} (y^{(k)} - a_{k-1}(X^{(k)}))$$

- Обучаем модель **регрессии** $b_k(X^{(k)})$ с MSE функцией потерь предсказывать остатки:

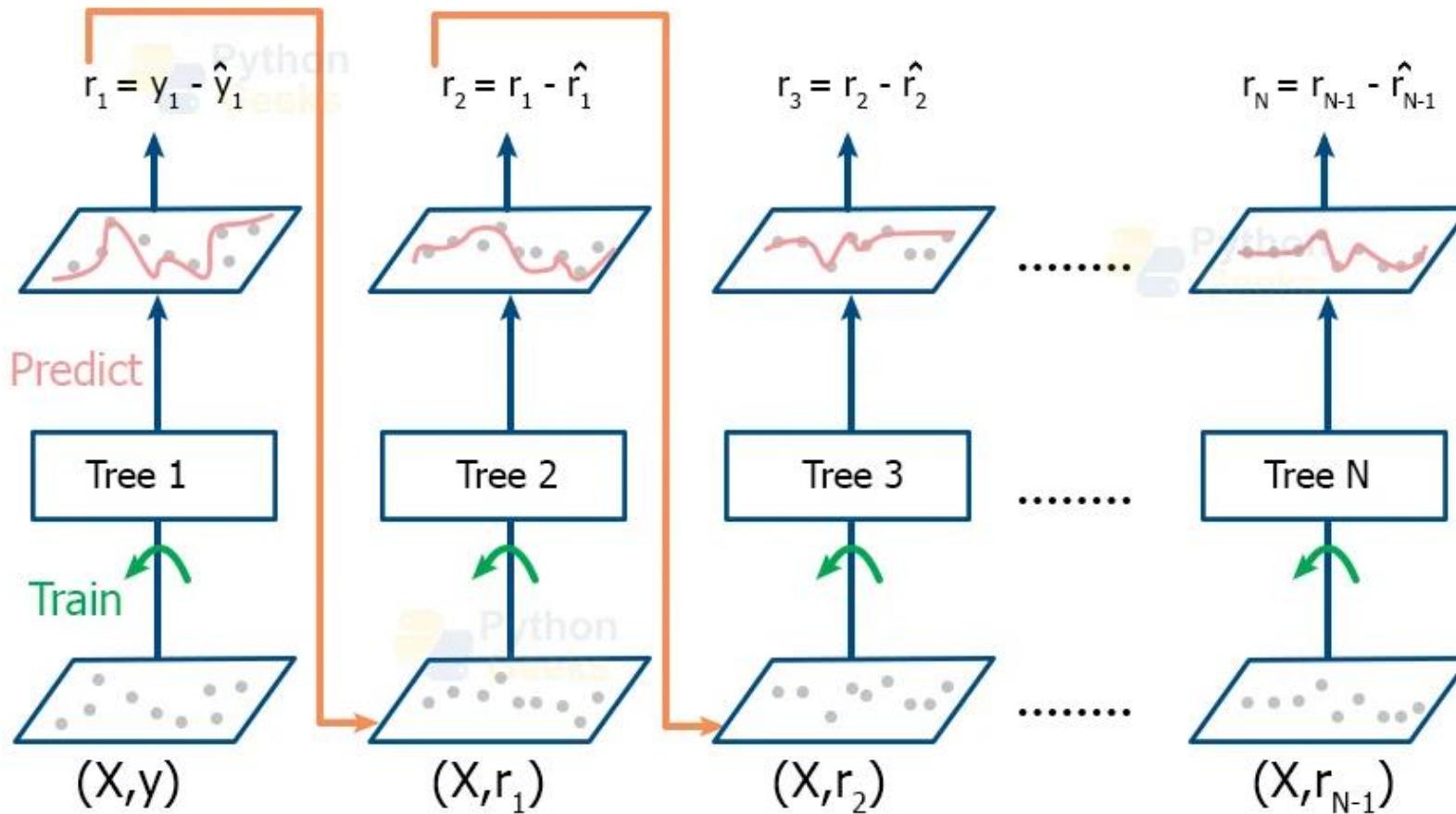
$$(b_k(X^{(k)}) - s)^T (b_k(X^{(k)}) - s) \rightarrow \min_{b_k}$$

- Обновляем композицию алгоритмов:

$$a_k(X^{(k)}) = a_{k-1}(X^{(k)}) + \gamma_k b_k(X^{(k)})$$

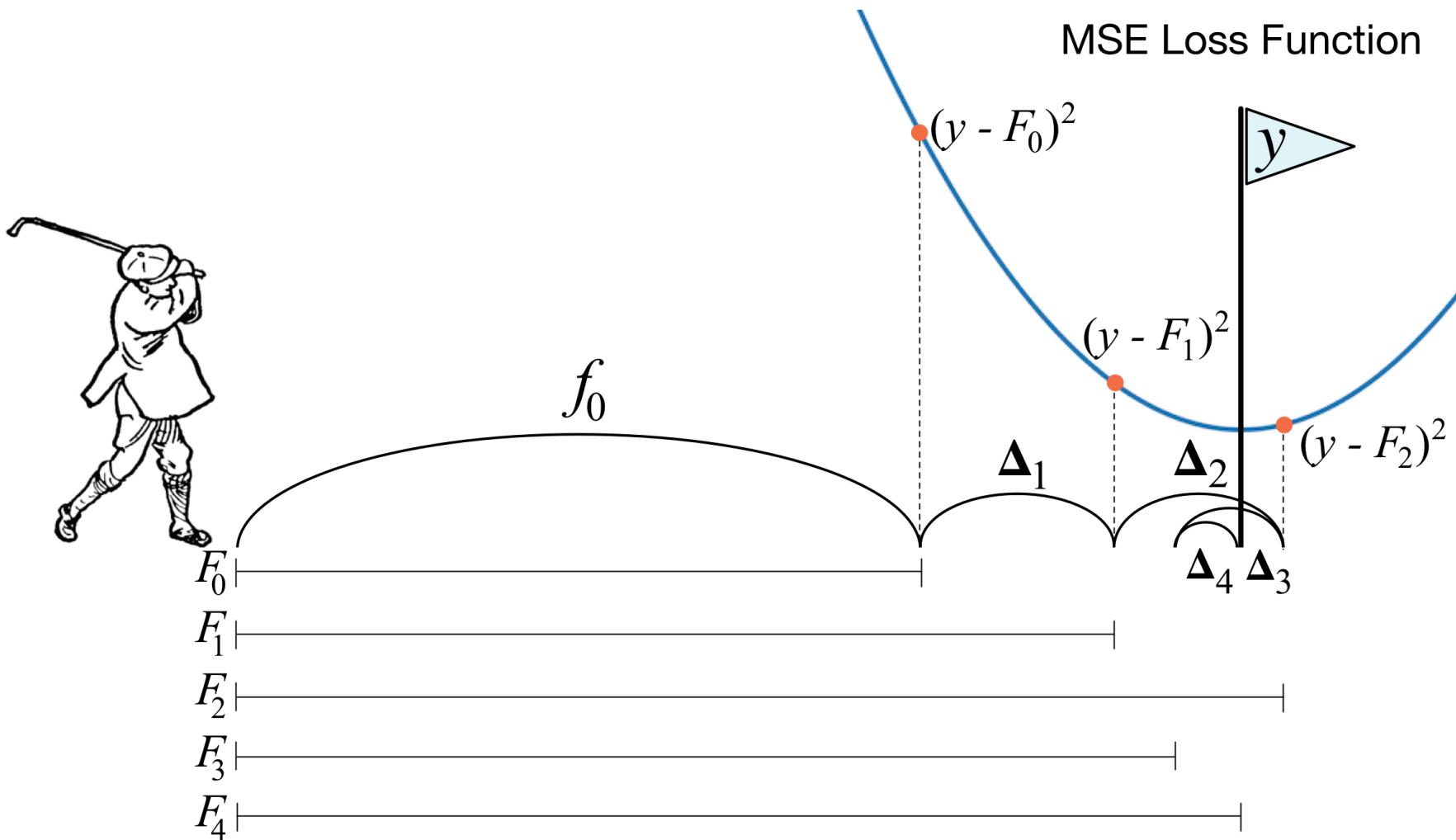
Градиентный бустинг для регрессии

Working of Gradient Boosting Algorithm

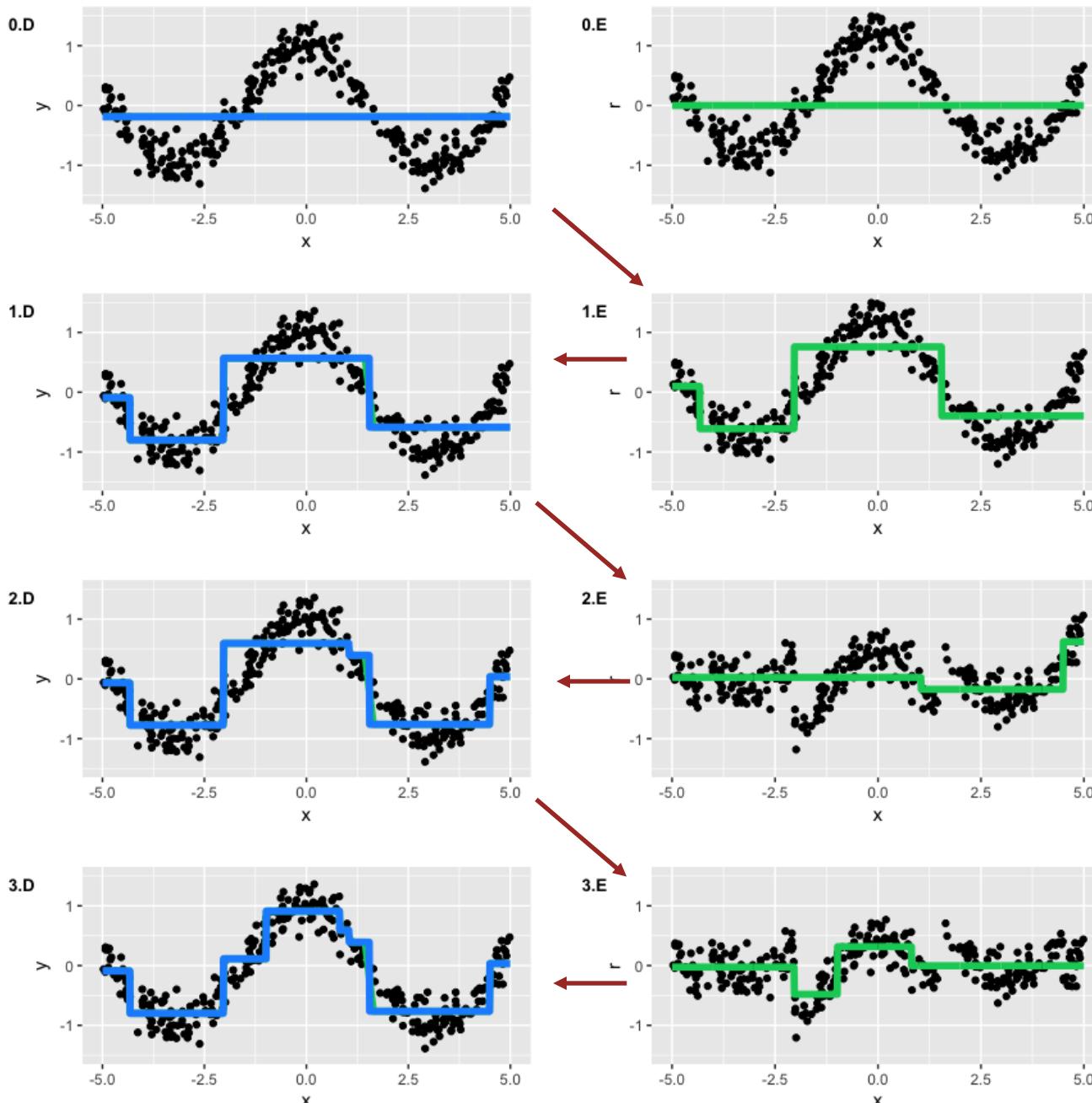


Источник: <https://pythongeeks.org/gradient-boosting-algorithm-in-machine-learning/>

Градиентный бустинг для регрессии



Пример



$$a_k(X) = a_{k-1}(X) + \gamma_k b_k(X)$$

$$s = -\frac{dL(y, z)}{dz} \Big|_{z=a_{k-1}(X)}$$

Градиентный бустинг с оптимальным шагом

- ▶ Делаем начальный прогноз $a_0(X) = b_0(X)$
- ▶ Для каждого $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Считаем вектор **остатков (сдвигов)** $s = -\frac{dL(y^{(k)}, a_{k-1}(X^{(k)}))}{da}$
 - Обучаем модель **регрессии** $b_k(X^{(k)})$ предсказывать остатки
 - Обновляем композицию алгоритмов:

$$L\left(y, a_{k-1}(X^{(k)}) + \gamma_k b_k(X^{(k)})\right) \rightarrow \min_{\gamma_k}$$

$$a_k(X^{(k)}) = a_{k-1}(X^{(k)}) + \eta \gamma_k b_k(X^{(k)})$$

- γ_k - оптимальный шаг градиентного бустинга
- $\eta \in [0, 1]$ – коэффициент сокращения шага (механизм регуляризации)

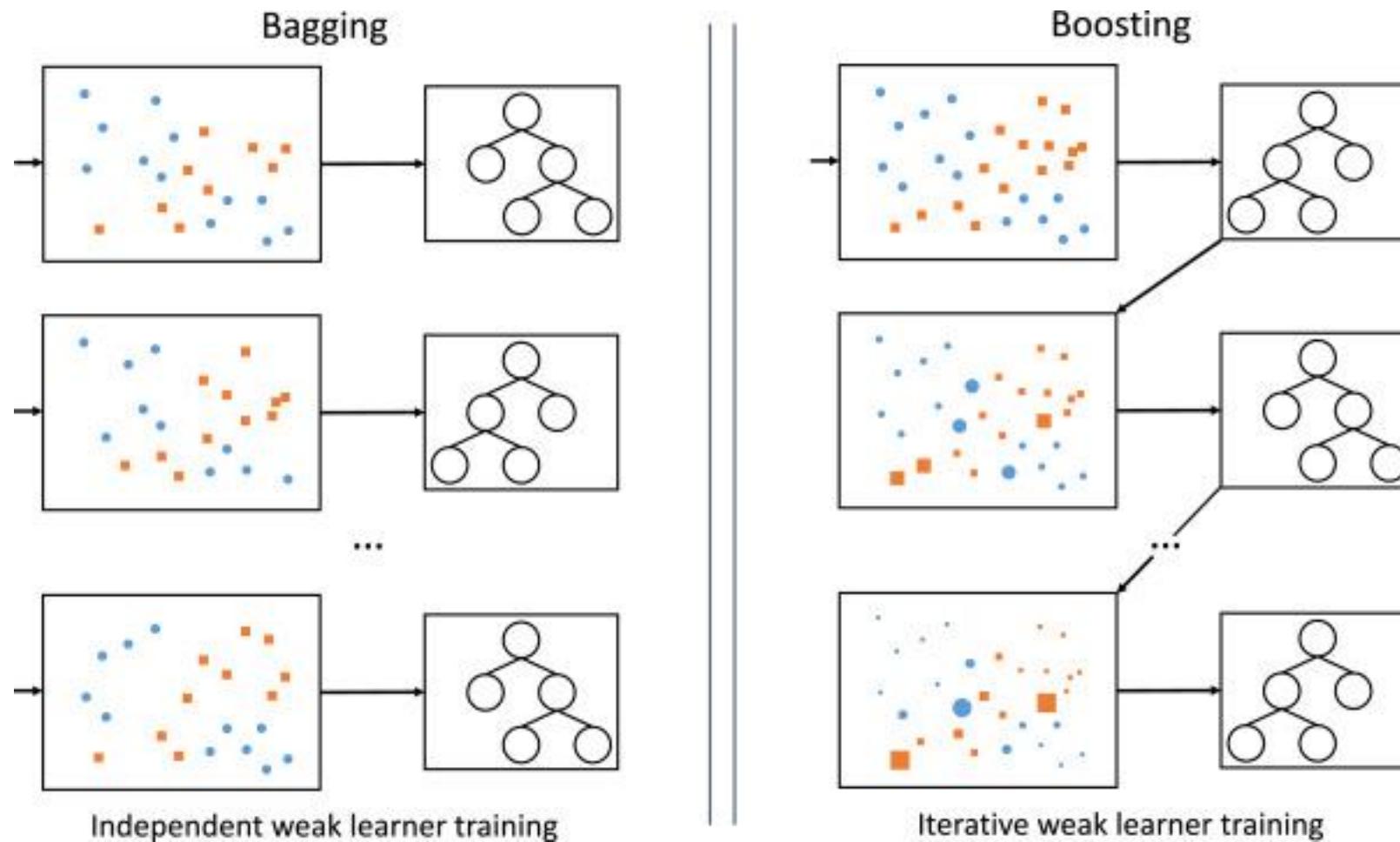
Сокращение шага

Сокращение шага. На практике оказывается, что градиентный бустинг очень быстро строит композицию, ошибка которой на обучении выходит на асимптоту, после чего начинает настраиваться на шум и переобучаться. Это явление можно объяснить одной из двух причин:

- Если базовые алгоритмы очень простые (например, решающие деревья небольшой глубины), то они плохо приближают вектор антиградиента. По сути, добавление такого базового алгоритма будет соответствовать шагу вдоль направления, сильно отличающегося от направления наискорейшего убывания. Соответственно, градиентный бустинг может свестись к случайному блужданию в пространстве.
- Если базовые алгоритмы сложные (например, глубокие решающие деревья), то они способны за несколько шагов бустинга идеально подогнаться подирующую выборку — что, очевидно, будет являться переобучением, связанным с излишней сложностью семейства алгоритмов.

Источник: <https://github.com/esokolov/ml-course-hse/blob/master/2018-fall/lecture-notes/lecture09-ensembles.pdf>

Бэггинг и бустинг



Источник: <https://www.sciencedirect.com/science/article/pii/S1566253520303195>

Важные замечания

- ▶ Неважно какую задачу вы решаете, классификацию или регрессию, в градиентном бустинге **всегда** используются модели **регрессии**
- ▶ Обычно в градиентном бустинге используют решающие деревья
- ▶ Популярные библиотеки:
 - sklearn
 - XGBoost
 - CatBoost
 - LightGBM

Демо

- ▶ [https://arogozhnikov.github.io/2016/07/05/gradient boosting playground.html](https://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html)
- ▶ [https://arogozhnikov.github.io/2016/06/24/gradient boosting explained.html](https://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html)

Заключение



Вопросы

- ▶ Опишите алгоритм построения композиции методом градиентного бустинга. Что такое сдвиги (остатки)?
- ▶ Что такое сокращение шага в градиентном спуске и для чего оно используется?