



“Бустинг” (Boosting)



Бустинг

- Ранее мы увидели, как можно улучшить работу деревьев решений с помощью случайных лесов.
- Далее мы рассмотрим другую методологию по улучшению работы отдельного дерева решений – называется “бустинг” (boosting).



Бустинг

- В этом разделе мы изучим:
 - Бустинг и мета-обучение
 - Теория AdaBoost (адаптивный бустинг)
 - Пример AdaBoost
 - Теория градиентного бустинга
 - Пример градиентного бустинга



“Бустинг” (Boosting)

Мотивация и история



Бустинг

- Бустинг – это не алгоритм машинного обучения, а методология для применения к существующим алгоритмам машинного обучения, чаще всего к деревьям решений.
- Рассмотрим идею мета-обучения...



Бустинг

- Основная формула для бустинга:

$$F_T(x) = \sum_{t=1}^T f_t(x)$$



Бустинг

- Основная формула для бустинга:

$$F_T(x) = \sum_{t=1}^T f_t(x)$$

$f_t(x) = \alpha_t h(x)$



Бустинг

- Основная формула для бустинга:

$$F_T(x) = \sum_{t=1}^T f_t(x) \quad f_t(x) = \alpha_t h(x)$$

- Здесь подразумевается, что комбинация объектов estimator с некоторыми коэффициентами может выдавать результаты для всего ансамбля.



Бустинг

$h(x)$ может быть любым алгоритмом машинного обучения

- Основная формула для бустинга:

$$F_T(x) = \sum_{t=1}^T f_t(x)$$

$f_t(x) = \alpha_t h(x)$

- Здесь подразумевается, что комбинация объектов estimator с некоторыми коэффициентами может выдавать результаты для всего ансамбля.



Бустинг

- Может ли ансамбль из слабых (простых) моделей стать сильной моделью?



Бустинг

- Может ли ансамбль из слабых (простых) моделей стать сильной моделью?
- Для деревьев решений мы можем в качестве $h(x)$ использовать простые деревья, объединив их потом с некоторыми коэффициентами



Бустинг

- Идея градиентного бустинга была высказана Лео Брейманом в 1990х годах - он увидел, что бустинг можно применять как алгоритм оптимизации для стоимостных функций
- Позже Джером Фридман и многие другие исследователи разработали более явные формулировки для градиентного бустинга.



Бустинг

- Также в конце 1990х годов Йоав Фройнд и Роберт Шапире разработали алгоритм AdaBoost (адаптивный бустинг), который также объединяет более слабые модели в единый ансамбль для усиления общей модели.



Бустинг

- Далее мы посмотрим AdaBoost и узнаем, как можно объединить несколько слабых моделей в ансамбль для усиления общей модели.
- Также мы узнаем, почему деревья решений так хорошо подходят для бустинга.



AdaBoost

Теория



Бустинг

- AdaBoost (адаптивный бустинг) использует ансамбли слабых моделей и затем объединяет их с помощью взвешенной суммы.
- Адаптация состоит в том, что ранее созданные слабые модели используются при построении следующих слабых моделей, уточняя их поведение для неправильно классифицированных точек.



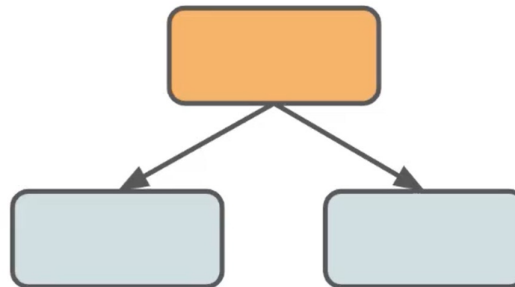
Бустинг

- Что такое слабая модель (weak learner)?
 - Это слишком простая модель – она сама по себе не даёт достаточно хороших результатов



Бустинг

- Что такое слабая модель (weak learner)?
 - Это слишком простая модель – она сама по себе не даёт достаточно хороших результатов
 - Простейшее дерево решений – один корневой узел и два листовых узла:





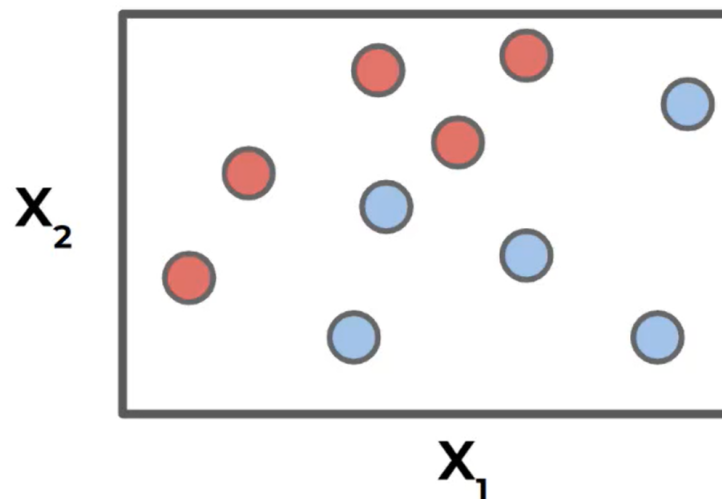
Бустинг

- В отличие от одного дерева решений, которое обучается сразу на всех данных, AdaBoost агрегирует несколько слабых моделей, позволяя обучать результирующую ансамблевую модель постепенно.
- Давайте посмотрим, как это выглядит с точки зрения данных...



Бустинг

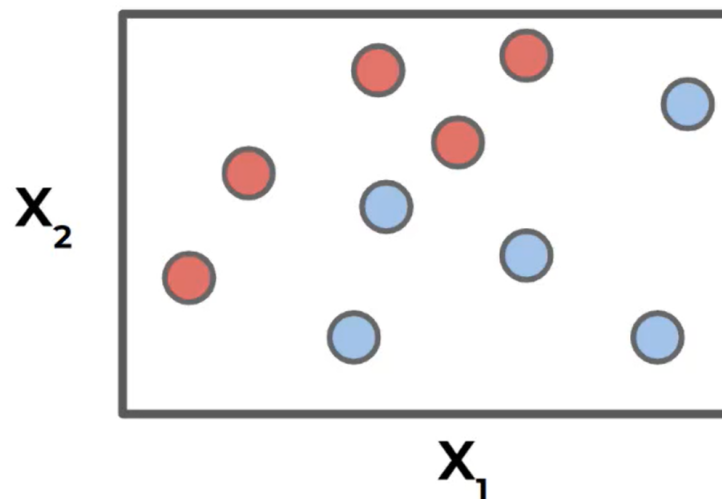
- Представим себе задачу классификации:





Бустинг

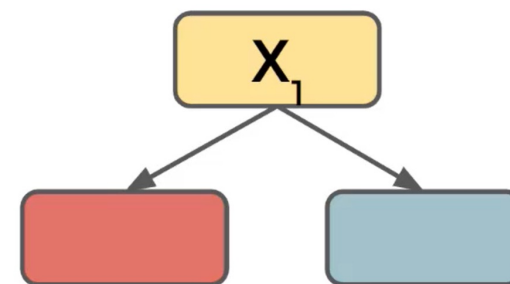
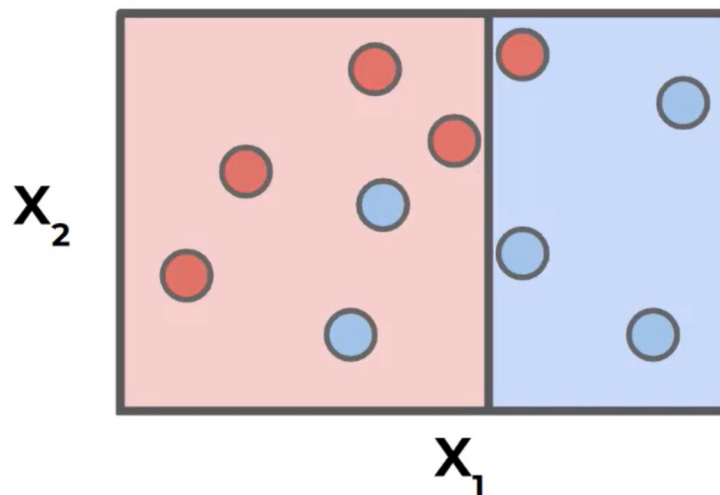
- Как может выглядеть простейшее дерево?





Бустинг

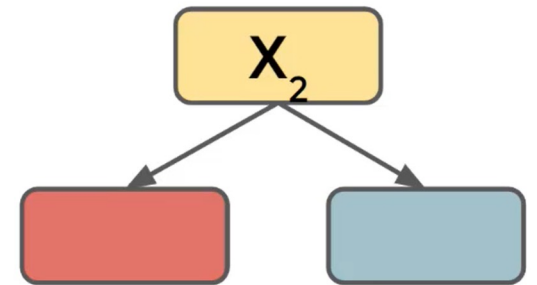
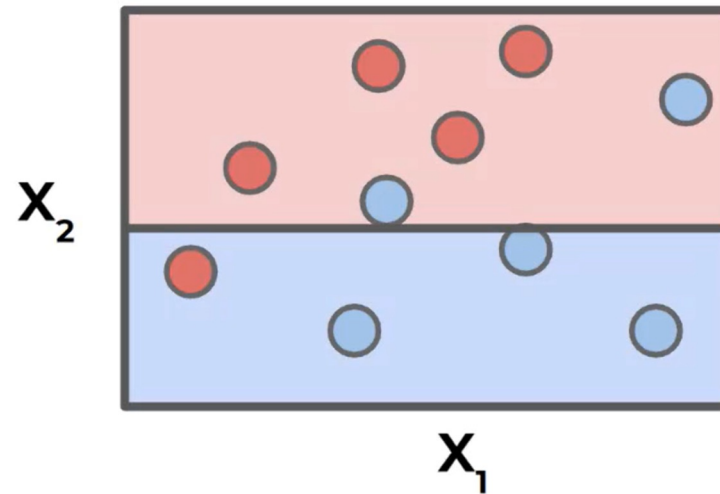
- Как может выглядеть простейшее дерево?





Бустинг

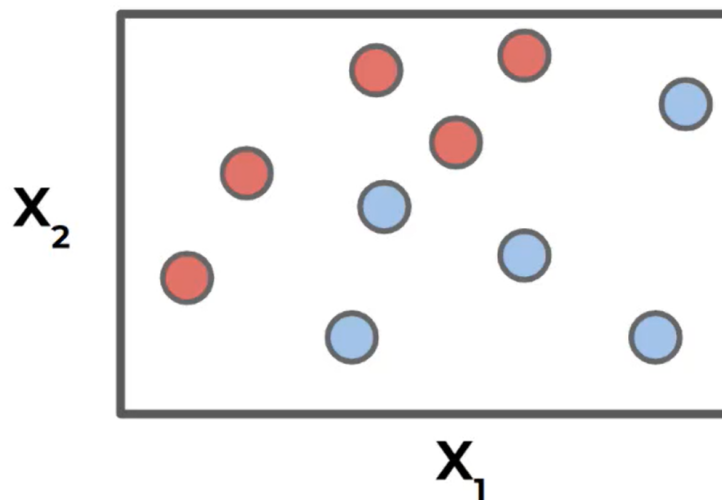
- Как может выглядеть простейшее дерево?





Бустинг

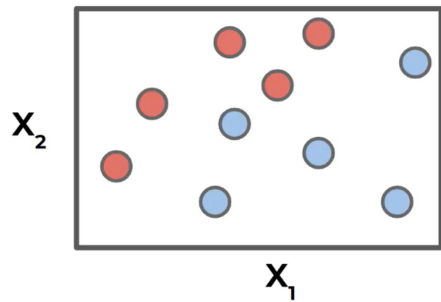
- Как мы можем объединить простейшие деревья?





Бустинг

- Процесс AdaBoost:

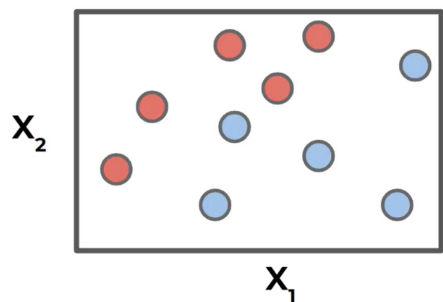


- Основные формулы
- Шаги алгоритма
- Визуальное представление алгоритма



Бустинг

- Процесс AdaBoost: Основные формулы

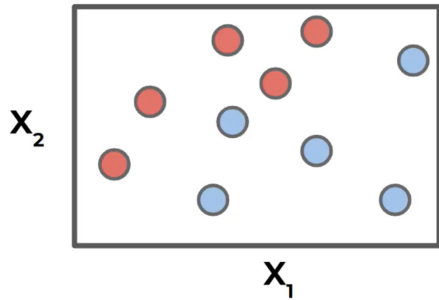


$$F_T(x) = \sum_{t=1}^T f_t(x)$$



Бустинг

- Процесс AdaBoost:



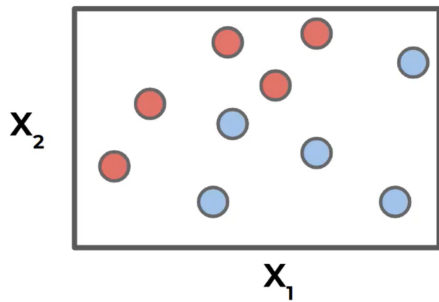
$$F_T(x) = \sum_{t=1}^T f_t(x)$$

$f_t(x) = \alpha_t h(x)$



Бустинг

- Процесс AdaBoost:



$$F_T(x) = \sum_{t=1}^T f_t(x)$$

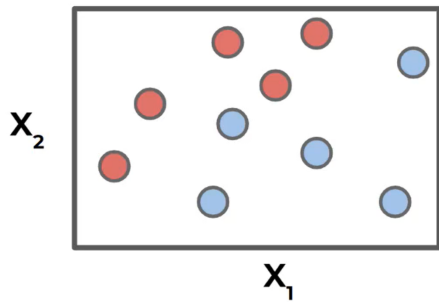
$$f_t(x) = \alpha_t h(x)$$

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$



Бустинг

- Процесс AdaBoost:



$$F_T(x) = \sum_{t=1}^T f_t(x)$$

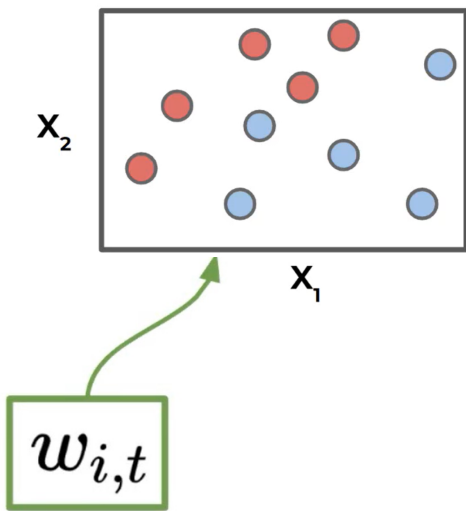
$$f_t(x) = \alpha_t h(x)$$

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$



Бустинг

- Процесс AdaBoost:



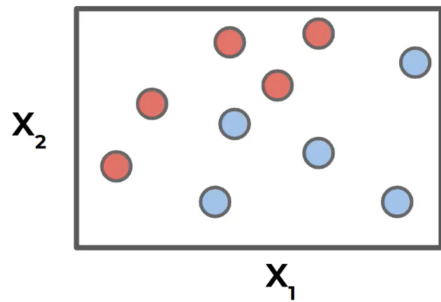
$$F_T(x) = \sum_{t=1}^T f_t(x) \quad f_t(x) = \alpha_t h(x)$$

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$



Бустинг

- Процесс AdaBoost: Шаги алгоритма



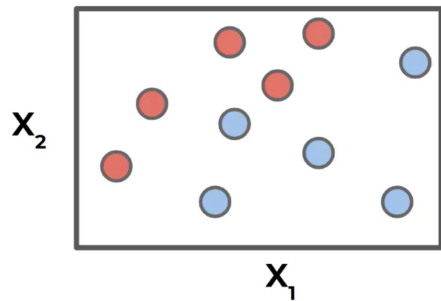
Терминология для алгоритма

- Точки данных $x_1 \dots x_n$
- Требуемые результаты $y_1 \dots y_n, y \in \{-1, 1\}$
- Начальные веса $w_{1,1} \dots w_{n,1}$ равны $\frac{1}{n}$
- Функция ошибок $E(f(x), y, i) = e^{-y_i f(x_i)}$
- Слабые модели $h: x \rightarrow \{-1, 1\}$



Бустинг

- Процесс AdaBoost:



Для t от 1 до T :

- Выбираем $h_t(x)$:
 - Находим слабую модель $h_t(x)$, которая минимизирует взвешенную сумму ошибок для неправильно классифицированных точек:

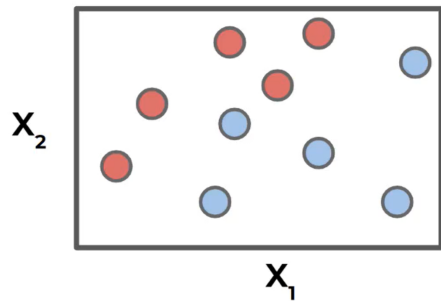
$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^n w_{i,t}$$

- Выбираем $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$



Бустинг

- Процесс AdaBoost:



Для t от 1 до T :

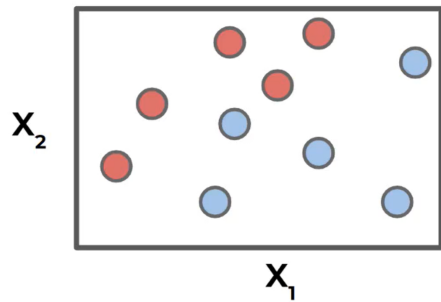
- Добавляем в ансамбль:
 - $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$
- Обновляем веса:
 - $w_{i,t+1} = w_{i,t} e^{-y_i \alpha_t h_t(x_i)}$ для i от 1 до n
 - Нормализуем $w_{i,t+1}$ так, чтобы

$$\sum_i w_{i,t+1} = 1$$



Бустинг

- Процесс AdaBoost: Визуализация алгоритма



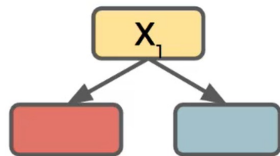
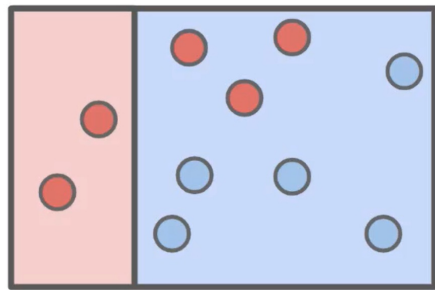
Терминология для алгоритма

- Точки данных $x_1 \dots x_n$
- Требуемые результаты $y_1 \dots y_n, y \in \{-1, 1\}$
- Начальные веса $w_{1,1} \dots w_{n,1}$ равны $\frac{1}{n}$
- Функция ошибок $E(f(x), y, i) = e^{-y_i f(x_i)}$
- Слабые модели $h: x \rightarrow \{-1, 1\}$



Бустинг

- Процесс AdaBoost:



Для t от 1 до T :

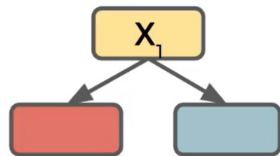
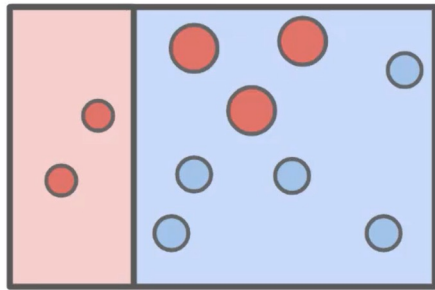
- Выбираем $h_t(x)$:
- Находим слабую модель $h_t(x)$, которая минимизирует взвешенную сумму ошибок для неправильно классифицированных точек:

$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^n w_{i,t}$$



Бустинг

- Процесс AdaBoost:



Для t от 1 до T :

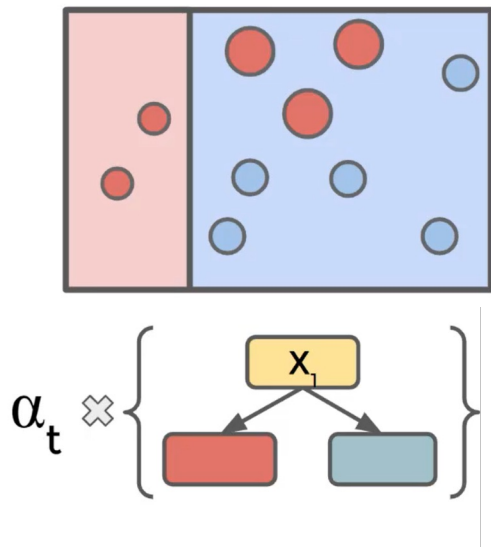
- Выбираем $h_t(x)$:
- Находим слабую модель $h_t(x)$, которая минимизирует взвешенную сумму ошибок для неправильно классифицированных точек:

$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^n w_{i,t}$$



Бустинг

- Процесс AdaBoost:



Для t от 1 до T :

- Выбираем $h_t(x)$:
 - Находим слабую модель $h_t(x)$, которая минимизирует взвешенную сумму ошибок для неправильно классифицированных точек:

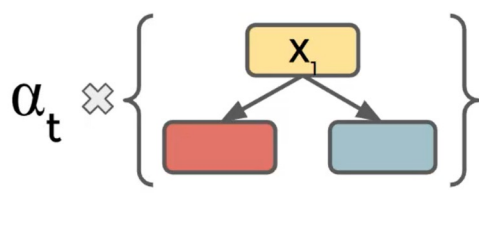
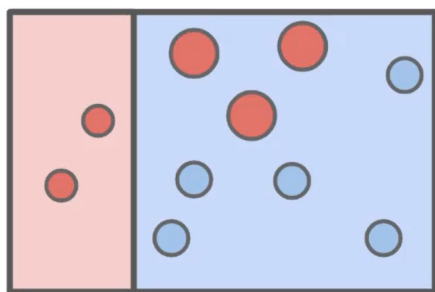
$$\epsilon_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^n w_{i,t}$$

- Выбираем $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$



Бустинг

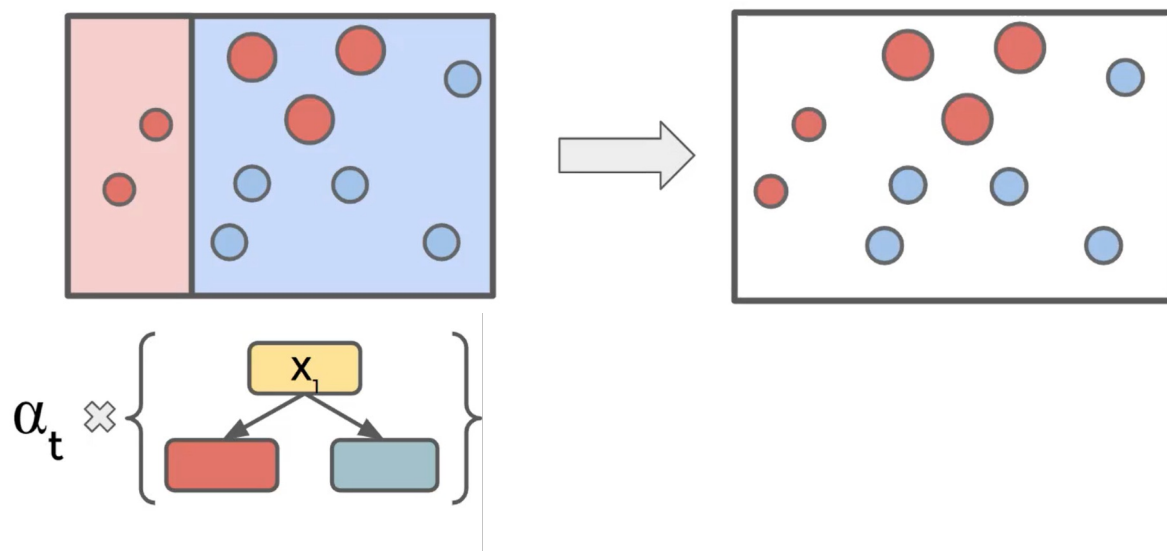
- Процесс AdaBoost:





Бустинг

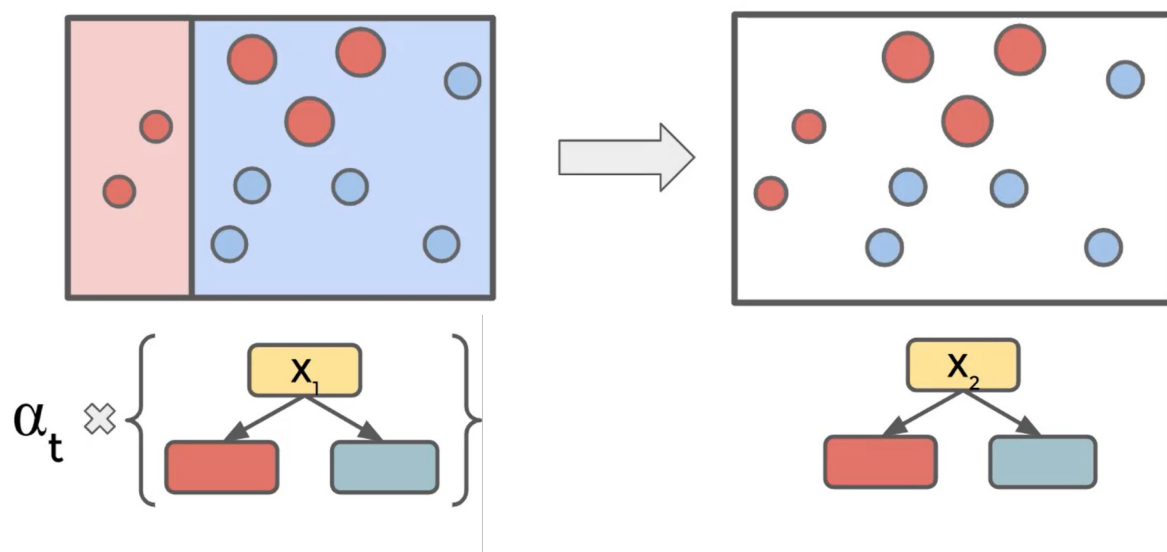
- Процесс AdaBoost:





Бустинг

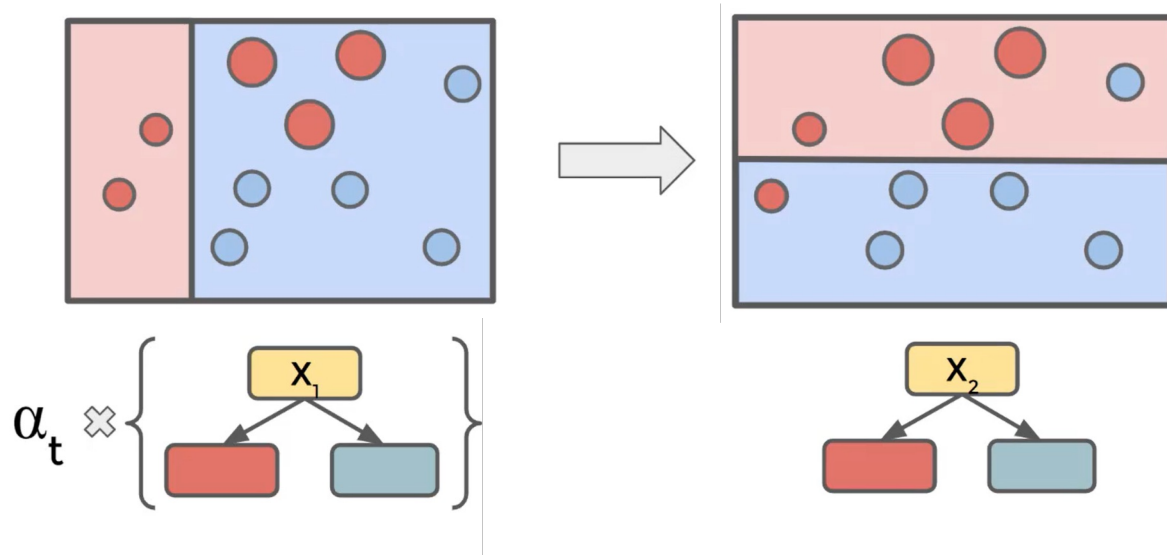
- Процесс AdaBoost:





Бустинг

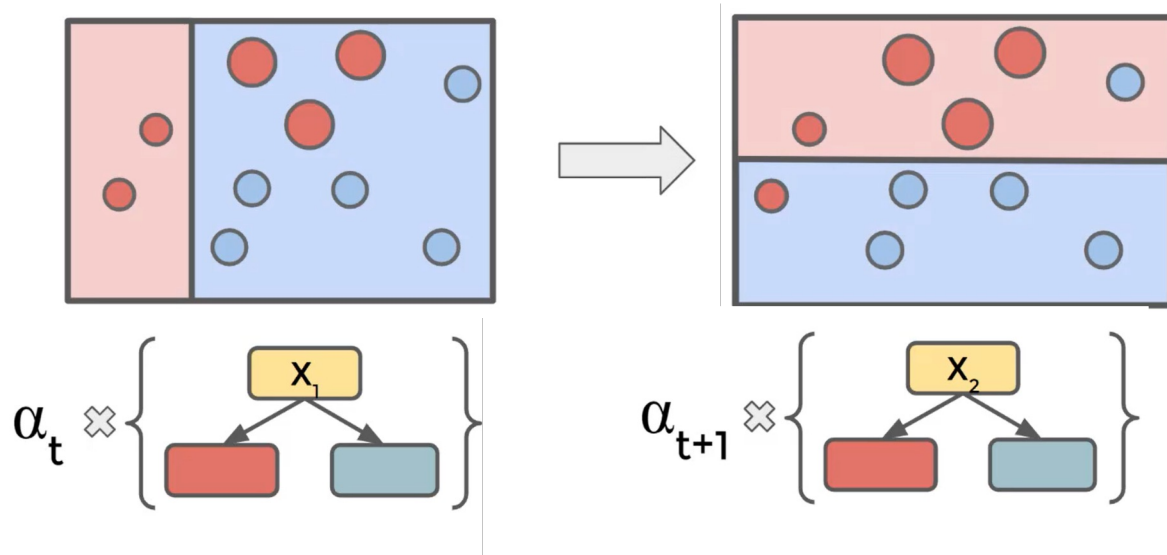
- Процесс AdaBoost:





Бустинг

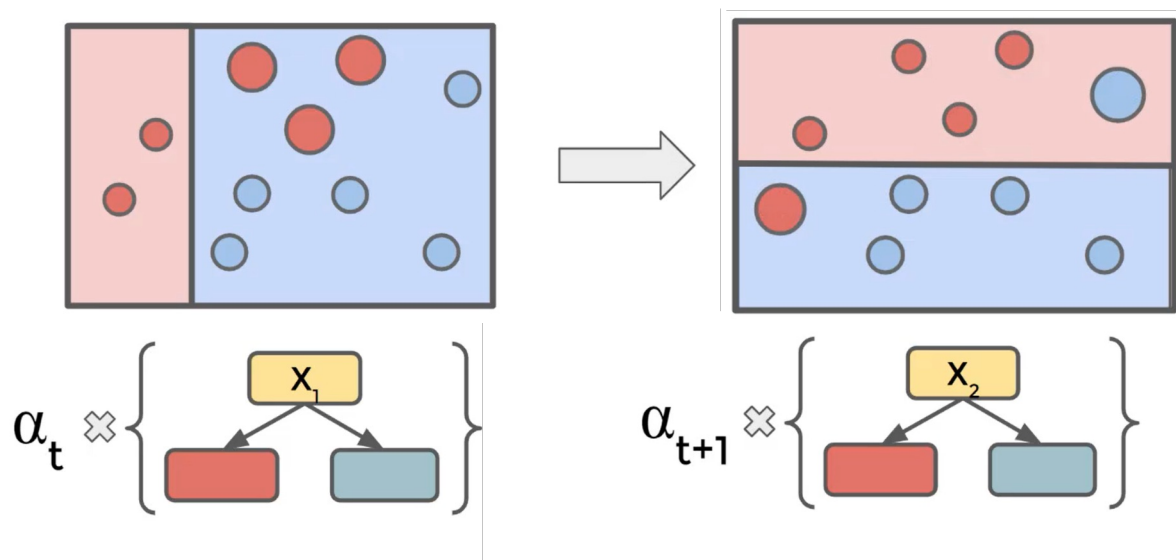
- Процесс AdaBoost:





Бустинг

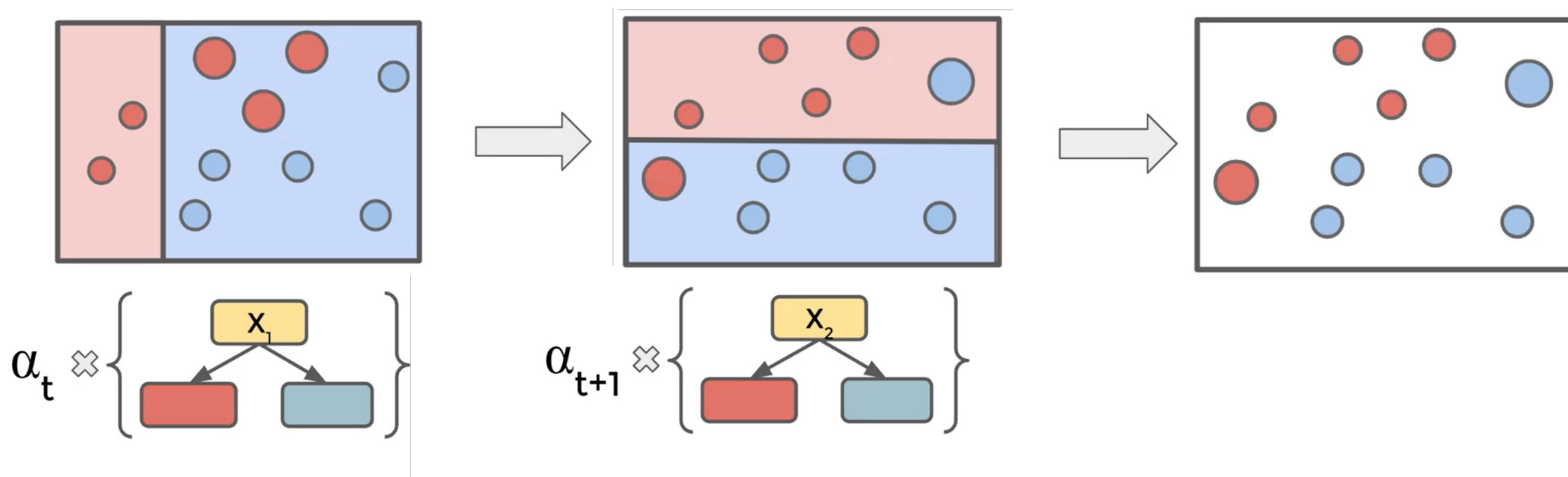
- Процесс AdaBoost:





Бустинг

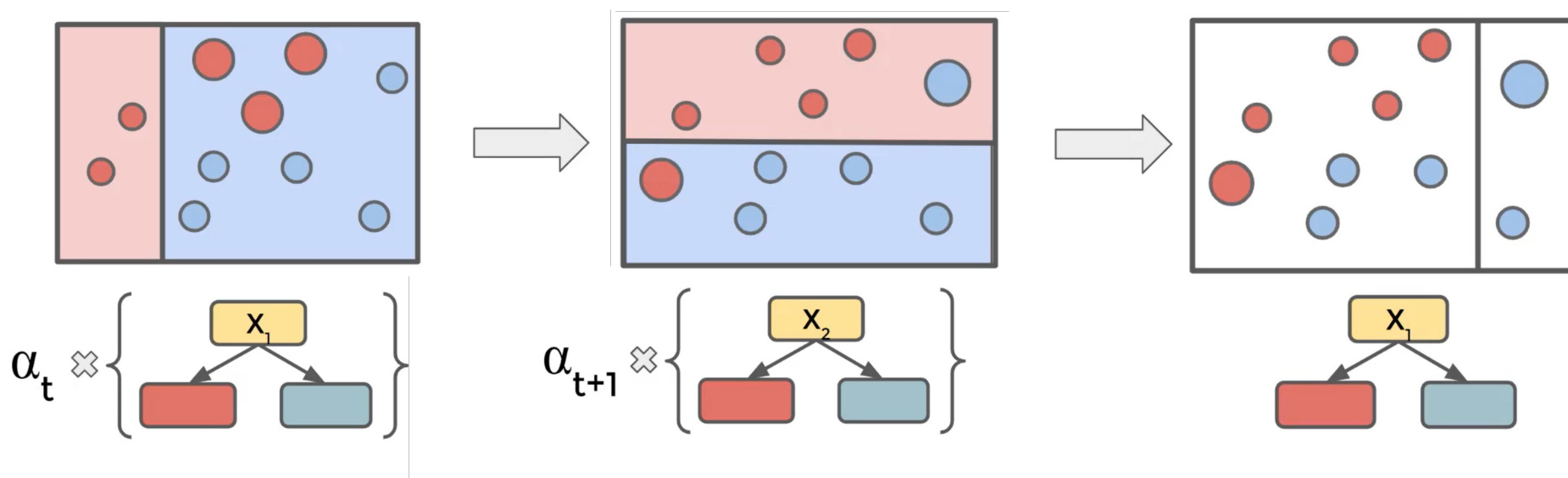
- Процесс AdaBoost:





Бустинг

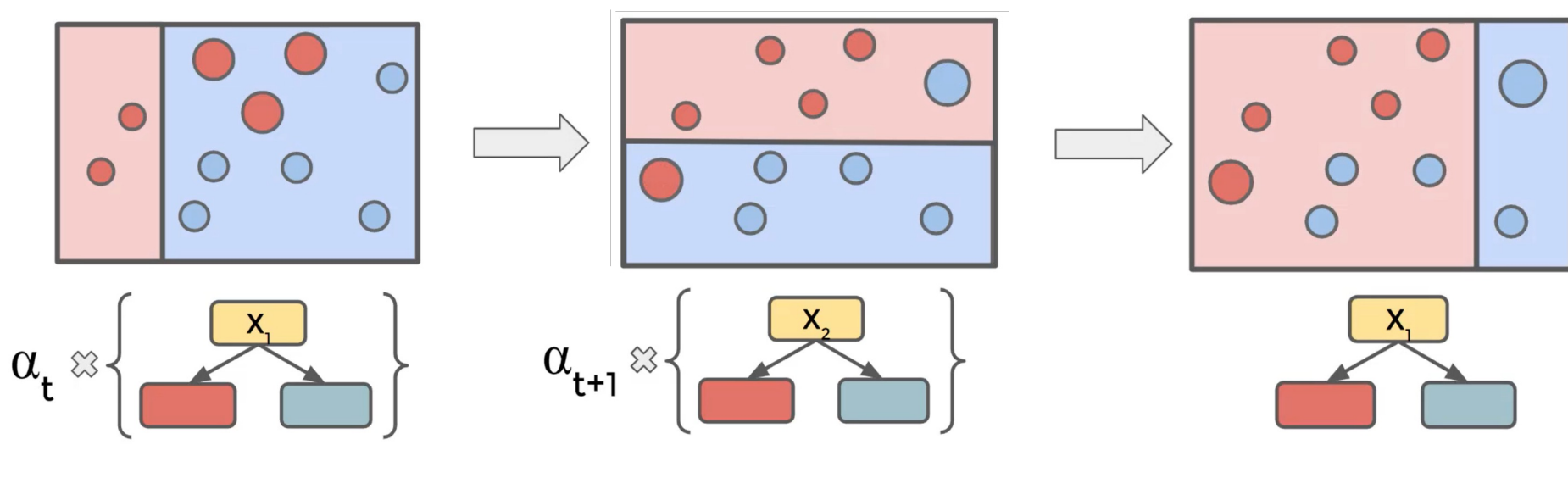
- Процесс AdaBoost:





Бустинг

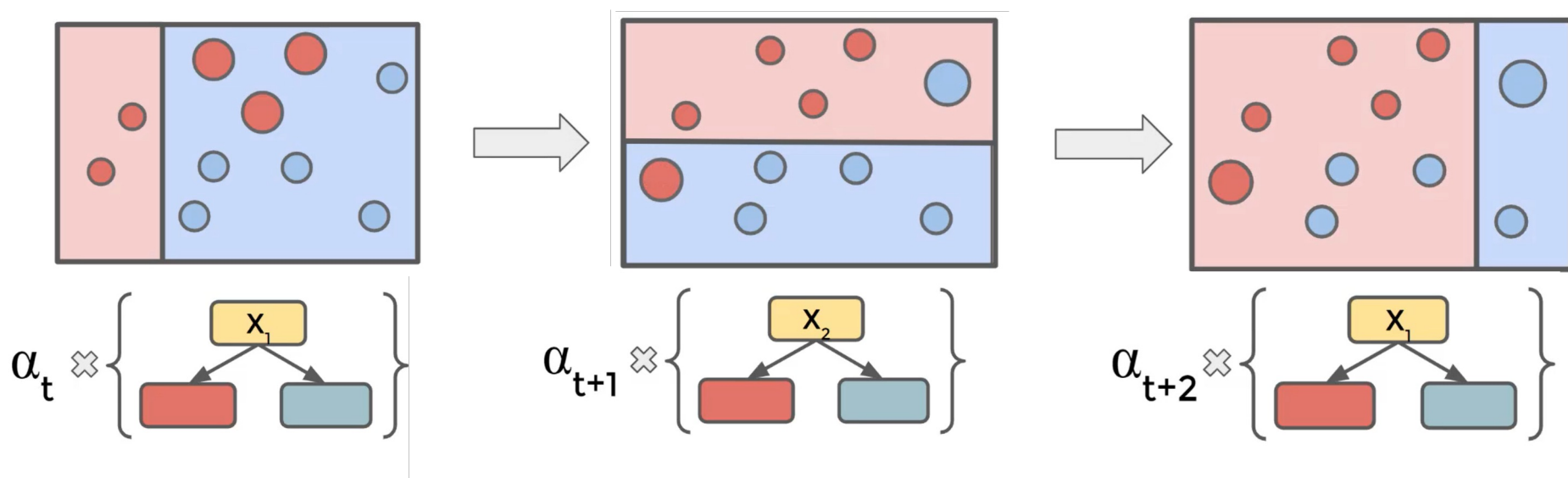
- Процесс AdaBoost:





Бустинг

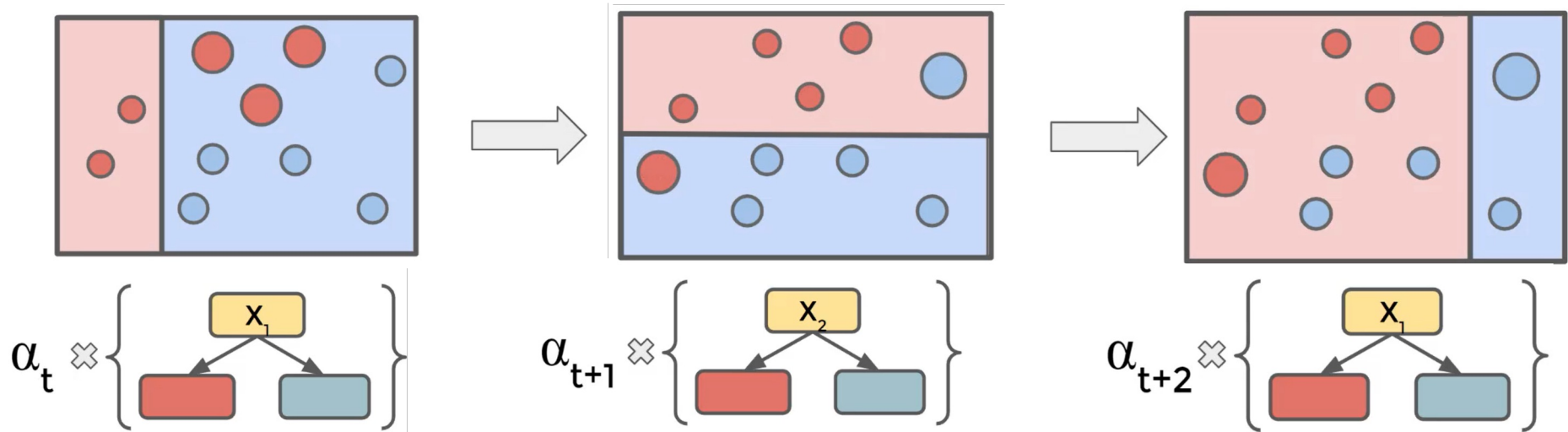
- Процесс AdaBoost:





Бустинг

- Процесс AdaBoost:



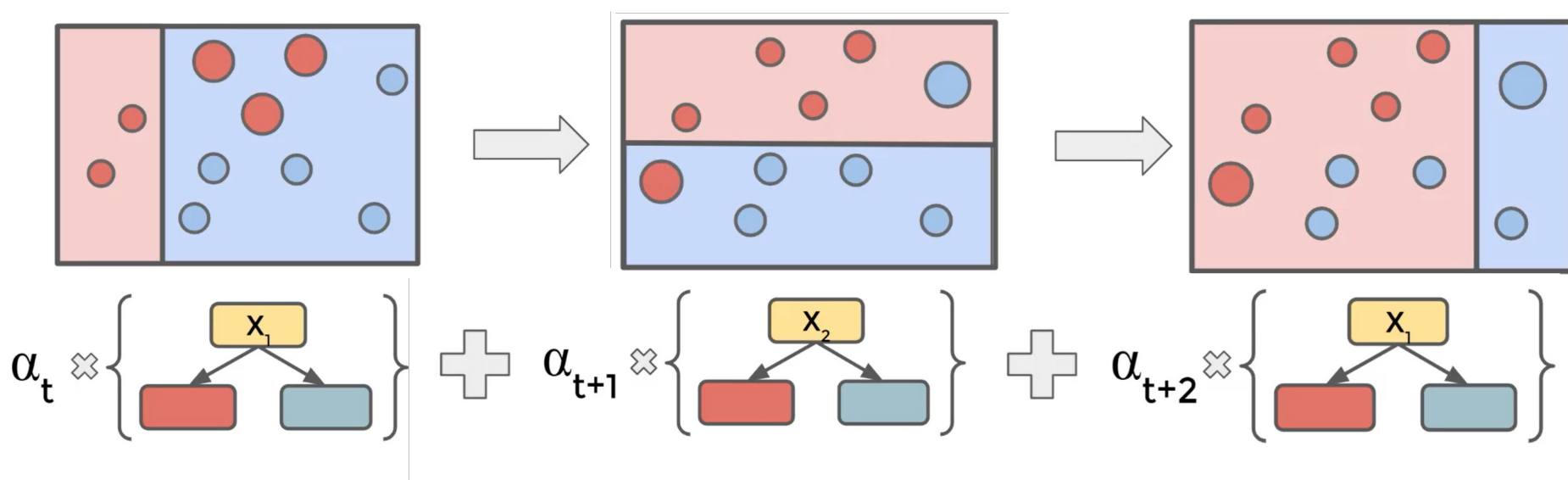
$$F_T(x) = \sum_{t=1}^T f_t(x)$$

$$f_t(x) = \alpha_t h(x)$$



Бустинг

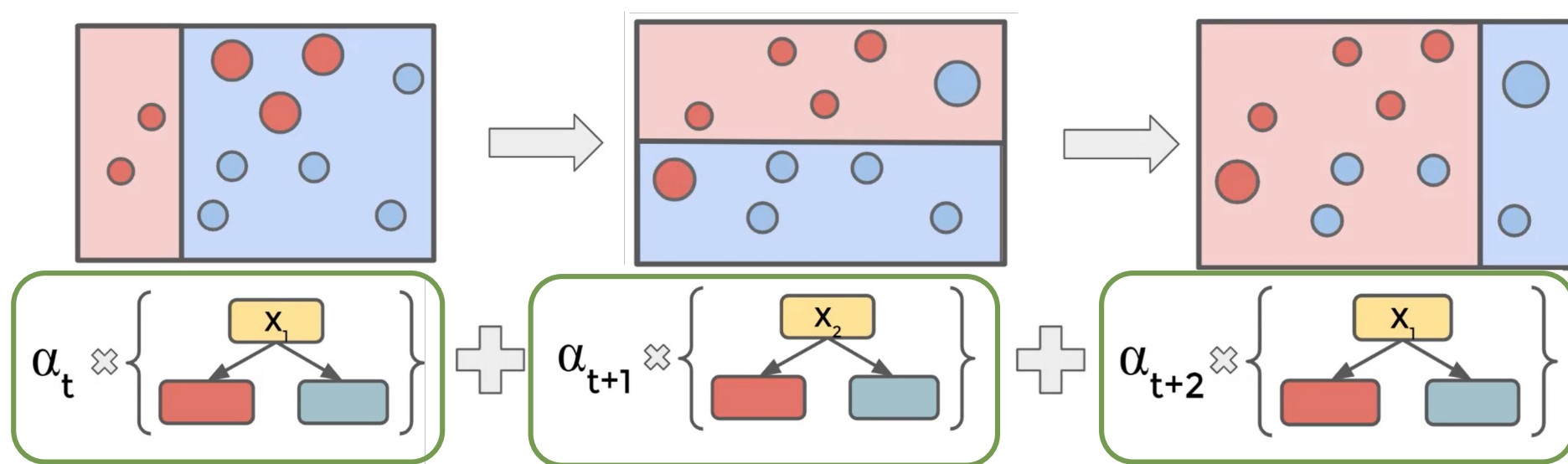
- Процесс AdaBoost:





Бустинг

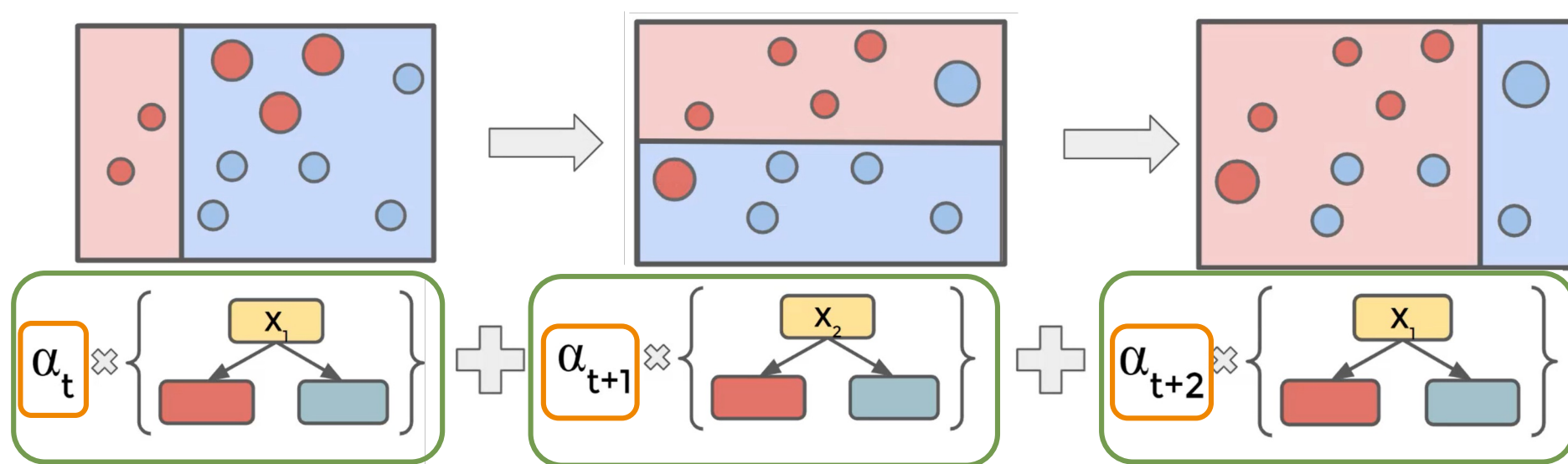
- Процесс AdaBoost:





Бустинг

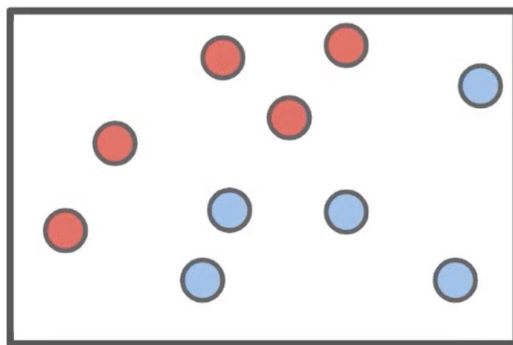
- Процесс AdaBoost:





Бустинг

- Процесс AdaBoost:

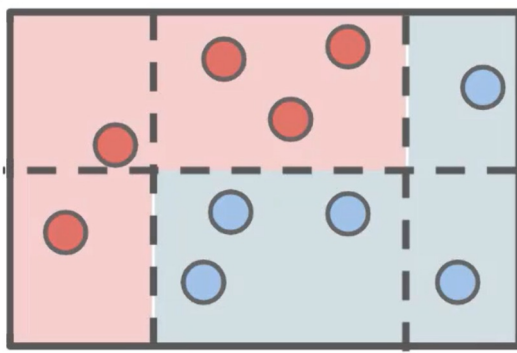


$$\alpha_t \times \left\{ \begin{array}{c} \text{yellow box } x_1 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\} + \alpha_{t+1} \times \left\{ \begin{array}{c} \text{yellow box } x_2 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\} + \alpha_{t+2} \times \left\{ \begin{array}{c} \text{yellow box } x_1 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\}$$



Бустинг

- Процесс AdaBoost:



$$\alpha_t \times \left\{ \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\} + \alpha_{t+1} \times \left\{ \begin{array}{c} x_2 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\} + \alpha_{t+2} \times \left\{ \begin{array}{c} x_1 \\ \swarrow \quad \searrow \\ \text{red box} \quad \text{blue box} \end{array} \right\}$$



Бустинг

- AdaBoost использует ансамбли слабых моделей, которые постепенно (медленно) обучаются.
- Некоторые слабые модели вносят в финальный результат бОльший вклад, чем другие – это учитывается умножением на параметр альфа.
- Каждая последующая модель t обучается на заново взвешенных данных от предыдущей модели $t-1$.



Бустинг

- Интуитивная интерпретация:
 - Каждое отдельное простое дерево представляет силу предсказания того или иного признака
 - Постепенное построение таких деревьев и применение параметров альфа позволяет нам комбинировать вместе важность различных признаков.



Бустинг

- Замечания:
 - В отличие от случайных лесов, AdaBoost может приводить к переобученности (overfitting).
 - Обычно ошибка стабилизируется намного раньше, чем количество деревьев становится насколько большим, что возникает переобученность.



AdaBoost

Пишем код – часть 1 - данные



Бустинг

- В этом проекте по машинному обучению мы будем рассматривать пример, в котором построение предиктивной модели не является основной целью.
- Давайте взглянем на данные, чтобы лучше понять рамки проекта.



Бустинг

- Данные о грибах - mushrooms.
 - Множество категориальных признаков для грибов
 - Можем ли мы применить машинное обучение для двух задач:
 - Предсказать, съедобный ли гриб или ядовитый
 - Создать предупреждающие советы для собирателей грибов



Бустинг

- Методы на основе деревьев решений имеют возможность определять важность признаков.
- AdaBoost на отдельных шагах работает с простыми деревьями, это может быть полезно для создания советов собирателям грибов.
- Посмотрим на данные в блокноте!



AdaBoost

Пишем код – часть 2 – модель



Градиентный бустинг

Теория



Бустинг

- Идея градиентного бустинга похожа на AdaBoost – слабые модели создаются постепенно, и затем объединяются в ансамбль.
- При обучении градиентный бустинг использует ошибки остатков (residual errors).



Бустинг

- Градиентный бустинг v.s. AdaBoost:
 - В градиентном бустинге допустимы более глубокие деревья
 - Коэффициент “learning rate” одинаков для всех слабых моделей
 - Обучение очередной модели выполняется с использованием остатков (residuals) предыдущей модели.



Бустинг

- Пример - градиентный бустинг для регрессии

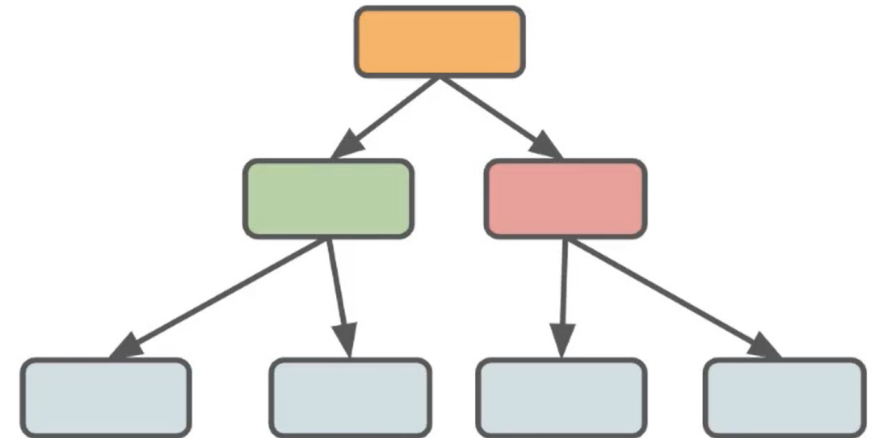
Area m ²	Bedrooms	Bathrooms	Price
200	3	2	\$500,000
190	2	1	\$462,000
230	3	3	\$565,000



Бустинг

- Обучаем на этих данных дерево решений

Area m ²	Bedrooms	Bathrooms	Price
200	3	2	\$500,000
190	2	1	\$462,000
230	3	3	\$565,000

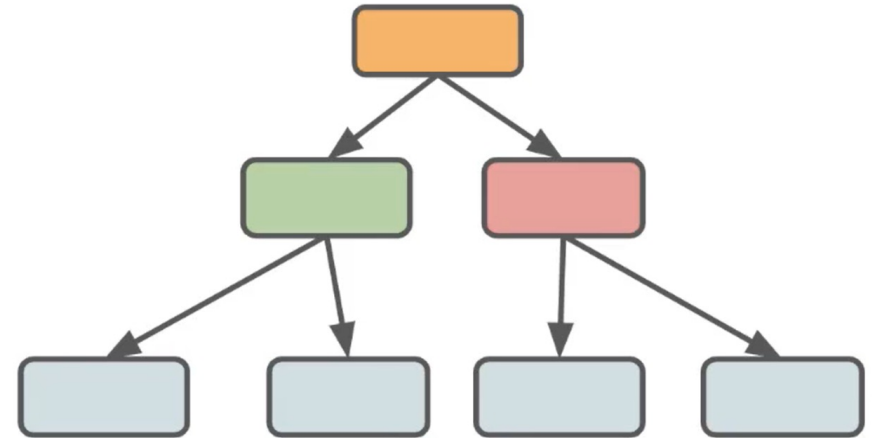




Бустинг

- Получаем прогнозируемые значения $\hat{y}$

y	$\hat{y}$
\$500,000	\$509,000
\$462,000	\$509,000
\$565,000	\$509,000

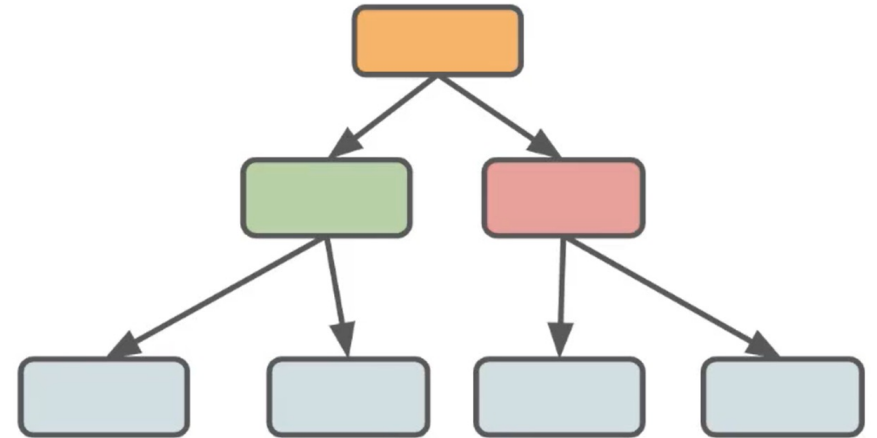




Бустинг

- Вычисляем остатки (residuals): $e = y - \hat{y}$

y	$\hat{y}$	e
\$500,000	\$509,000	-\$9,000
\$462,000	\$509,000	-\$47,000
\$565,000	\$509,000	\$56,000





Бустинг

- Создаём новую модель для предсказания ошибки

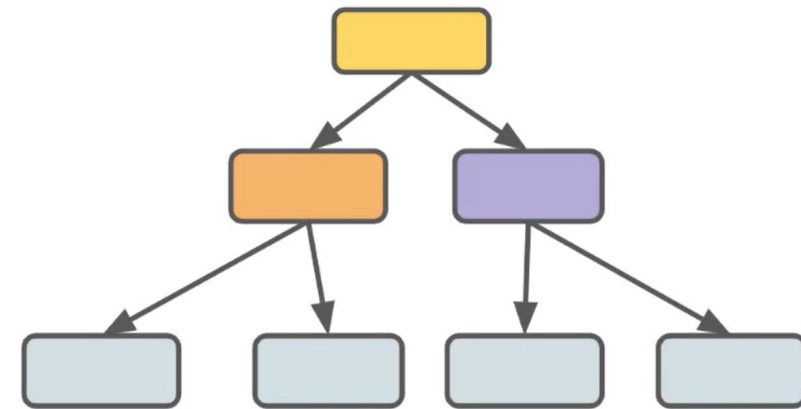
y	$\hat{y}$	e
\$500,000	\$509,000	-\$9,000
\$462,000	\$509,000	-\$47,000
\$565,000	\$509,000	\$56,000



Бустинг

- Создаём новую модель для предсказания ошибки

y	$\hat{y}$	e
\$500,000	\$509,000	-\$9,000
\$462,000	\$509,000	-\$47,000
\$565,000	\$509,000	\$56,000



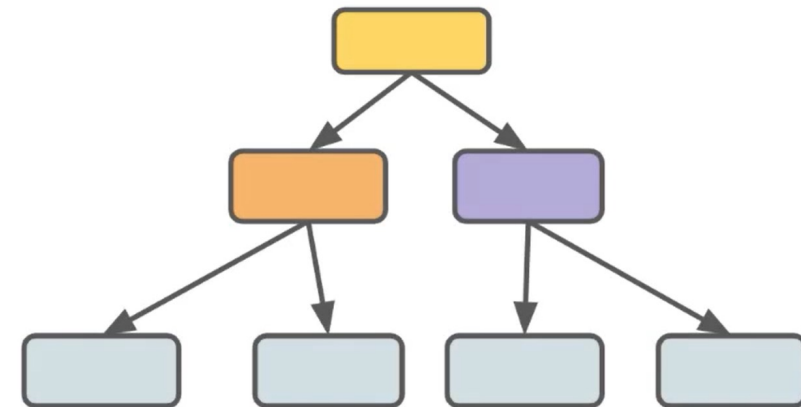


Бустинг

- Создаём новую модель для предсказания ошибки

y	$\hat{y}$	e	$f1$
\$500,000	\$509,000	-\$9,000	-\$8,000
\$462,000	\$509,000	-\$47,000	-\$50,000
\$565,000	\$509,000	\$56,000	\$50,000

Area m ²	Bedrooms	Bathrooms
200	3	2
190	2	1
230	3	3



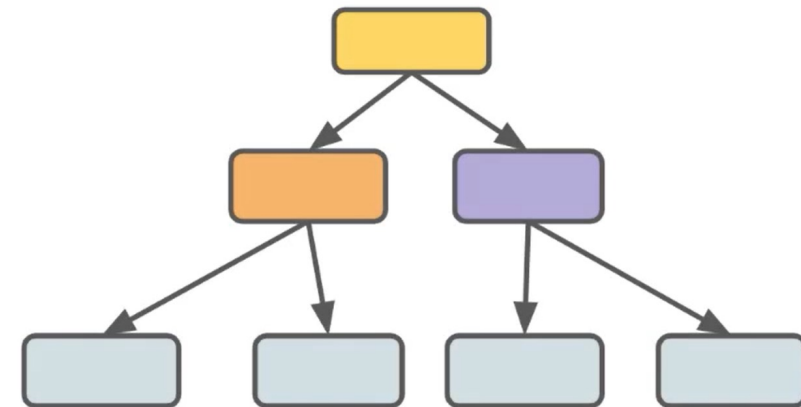


Бустинг

- Создаём новую модель для предсказания ошибки

y	$\hat{y}$	e	$f1$
\$500,000	\$509,000	-\$9,000	-\$8,000
\$462,000	\$509,000	-\$47,000	-\$50,000
\$565,000	\$509,000	\$56,000	\$50,000

Area m ²	Bedrooms	Bathrooms
200	3	2
190	2	1
230	3	3

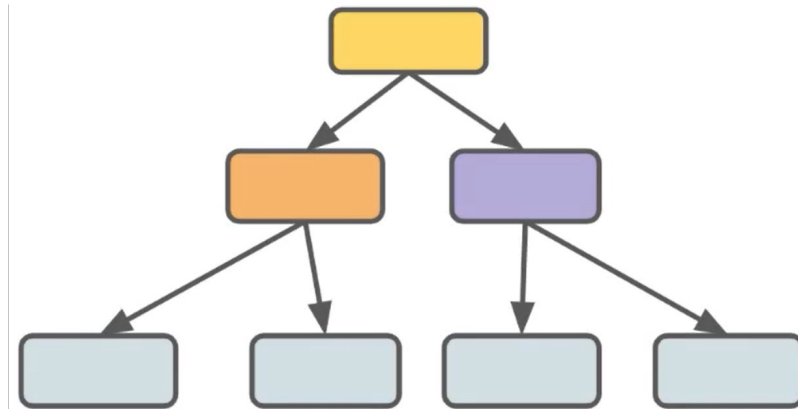




Бустинг

- Уточняем предсказываемое значение

y	$\hat{y}$	e	$f1$	$F1 = \hat{y} + f1$
\$500,000	\$509,000	-\$9,000	-\$8,000	
\$462,000	\$509,000	-\$47,000	-\$50,000	
\$565,000	\$509,000	\$56,000	\$50,000	

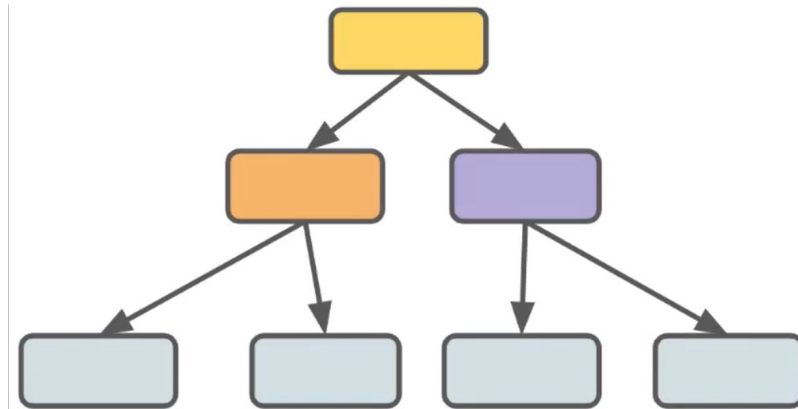




Бустинг

- Уточняем предсказываемое значение

y	$\hat{y}$	e	$f1$	$F1 = \hat{y} + f1$
\$500,000	\$509,000	-\$9,000	-\$8,000	\$501,000
\$462,000	\$509,000	-\$47,000	-\$50,000	\$459,000
\$565,000	\$509,000	\$56,000	\$50,000	\$559,000

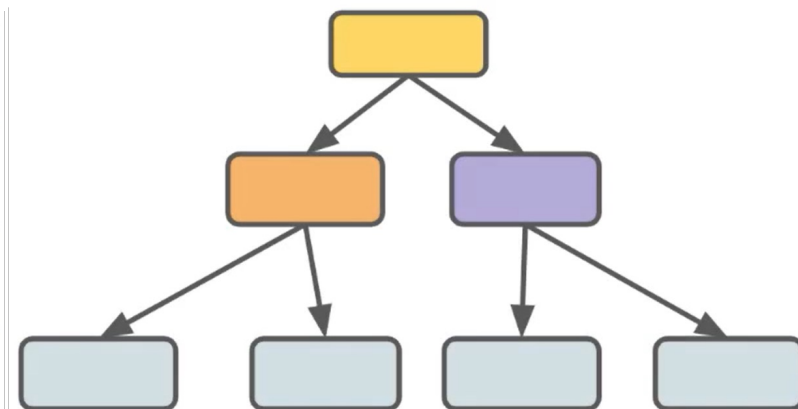




Бустинг

- Можем далее повторять этот процесс

y	$\hat{y}$	e	$f1$	$F1 = \hat{y} + f1$
\$500,000	\$509,000	-\$9,000	-\$8,000	\$501,000
\$462,000	\$509,000	-\$47,000	-\$50,000	\$459,000
\$565,000	\$509,000	\$56,000	\$50,000	\$559,000





Бустинг

- Процесс градиентного бустинга

$$F_m = F_{m-1} + f_m$$



Бустинг

- Процесс градиентного бустинга

$$F_m = F_{m-1} + f_m$$

$$F_m = F_{m-1} + (\text{learning rate} * f_m)$$



Бустинг

- Процесс градиентного бустинга
 - Создаём начальную модель f_0
 - Обучаем другую модель на ошибке
 - $e = y - f_0$
 - Создаём новое прогнозируемое значение
 - $F_1 = f_0 + \eta f_1$
 - Повторяем при необходимости
 - $F_m = f_{m-1} + \eta f_m$



Бустинг

- Для задач классификации мы можем применять метрику ошибки на основе логарифма:

$$\hat{y} = \log \left(\frac{\hat{p}}{1 - \hat{p}} \right) \qquad \hat{p} = \frac{1}{1 + e^{-\hat{y}}}$$



Бустинг

- Параметр “learning rate” в градиентном бустинге одинаковый для каждой последующей модели (в отличие от параметров альфа в AdaBoost).
- Градиентный бустинг достаточно устойчив к переобучению (overfitting), что позволяет выставить параметр number of estimators высоким по умолчанию (~100).



Бустинг

- Интерпретация градиентного бустинга
 - Мы оптимизируем последовательность деревьев, обучаясь на остатках – последующие деревья пытаются исправить ошибки предыдущих деревьев.



Бустинг

- Интерпретация градиентного бустинга
 - Баланс, который нужно соблюсти – это время обучения.
 - Параметр “learning rate” от 0 до 1. Низкие значения дают меньший вес последующим деревьям, это приводит к увеличению количества деревьев и увеличению времени на обучение.



Градиентный бустинг

Пишем код