



# ML. Третье занятие

# План занятия

- 1.Решающие деревья
2. Обучение решающих деревьев
3. Обработка пропусков в решающих деревьях
4. Работа с категориальными признаками в решающих деревьях
5. Стрижка деревьев
6. Композиции моделей
7. Разложение ошибки на смещение и разброс
8. Бэггинг
9. Случайный лес
10. Градиентный бустинг
11. Регуляризация градиентного бустинга
12. Градиентный бустинг над деревьями
13. Блендинг и Стекинг

# Решающие деревья

Лекция 10

# Решающие деревья

## Линейные модели

Плюсы линейных моделей:

- можно найти аналитическое решение
- мало параметров
- можно обучать на больших объемах данных
- можно обучать градиентным спуском

# Решающие деревья

## Линейные модели

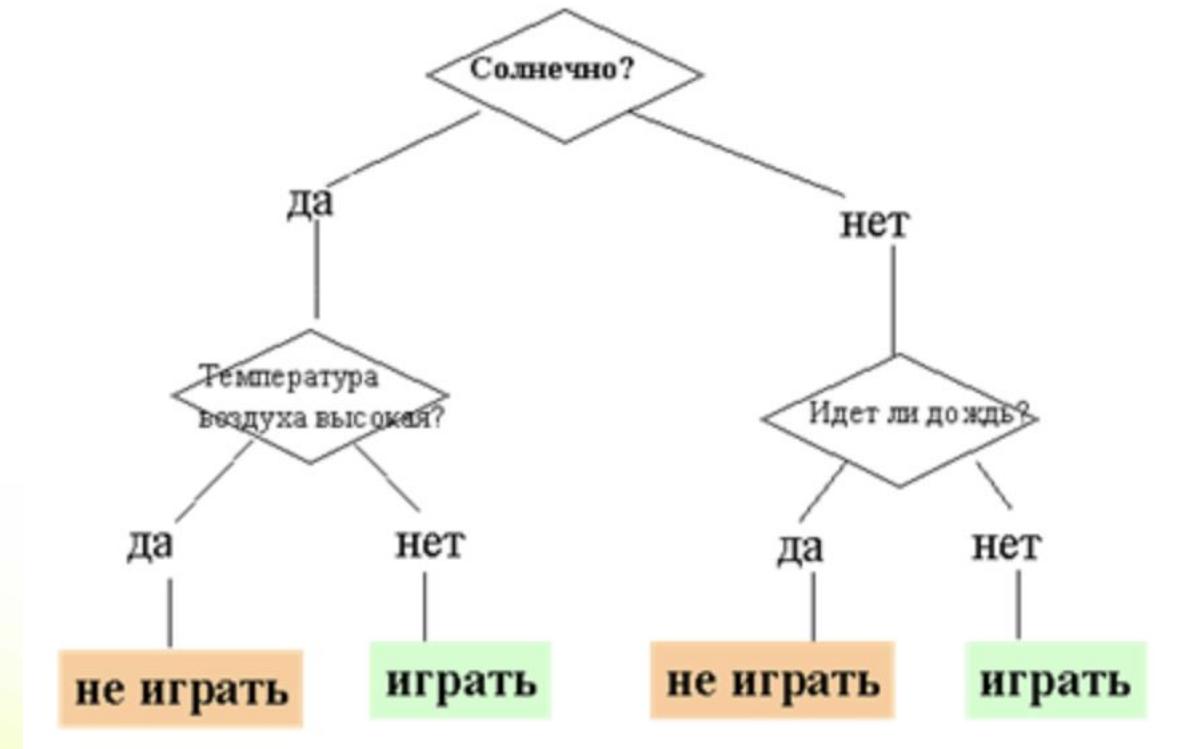
### Плюсы линейных моделей:

- можно найти аналитическое решение
- мало параметров
- можно обучать на больших объемах данных
- можно обучать градиентным спуском

### Минусы линейных моделей:

- нужно правильно подготавливать данные
- нужно обрабатывать пропуски
- предполагает что каждый признак вносит линейный вклад в целевую переменную

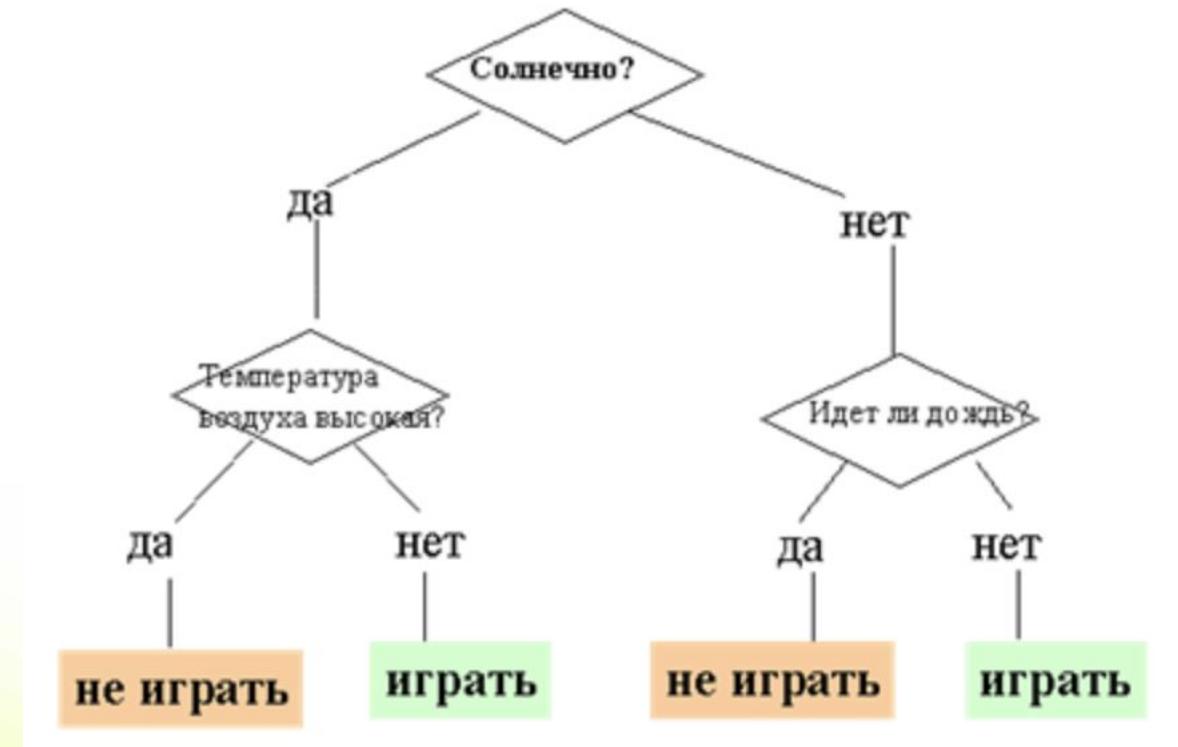
# Решающие деревья



# Решающие деревья

Решающие деревья:

- бинарные деревья



# Решающие деревья

Решающие деревья:

- бинарные деревья
- во внутренних вершинах  $v$  заданы предикаты  $\beta_v$

Предикат – это функция, которая на вход принимает объект и выдает ответ 0 или 1.

$$\beta_v: (x_1, \dots, x_n) \rightarrow \{0,1\}$$



# Решающие деревья

Решающие деревья:

- бинарные деревья
- во внутренних вершинах  $v$  заданы предикаты  $\beta_v$

Предикат – это функция, которая на вход принимает объект и выдает ответ 0 или 1.

$$\beta_v: (x_1, \dots, x_n) \rightarrow \{0,1\}$$

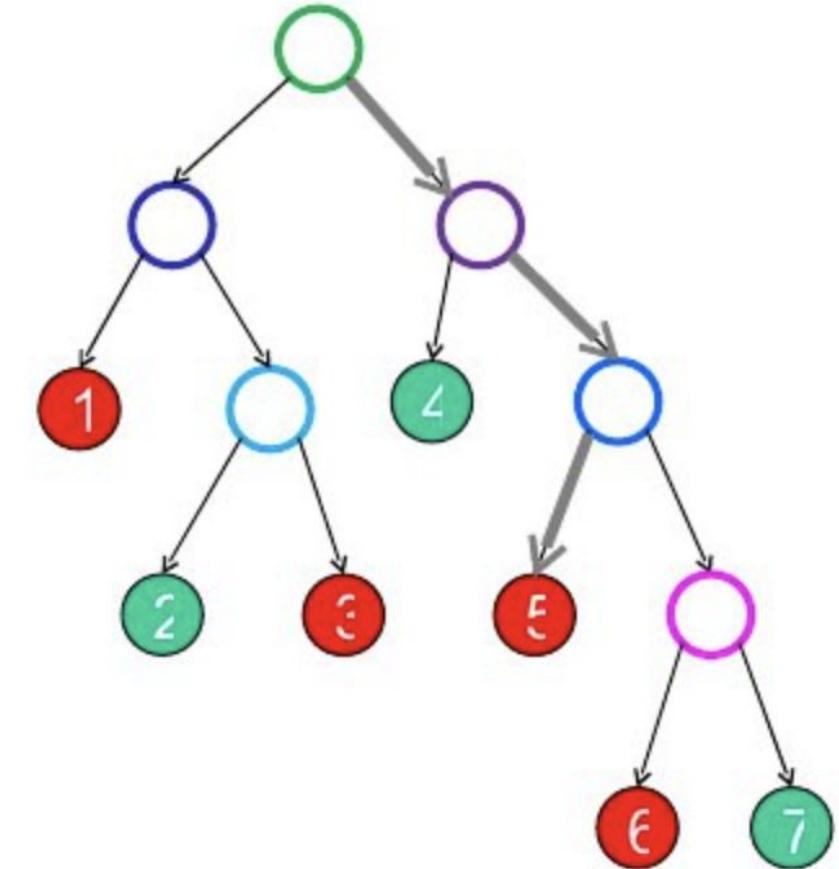
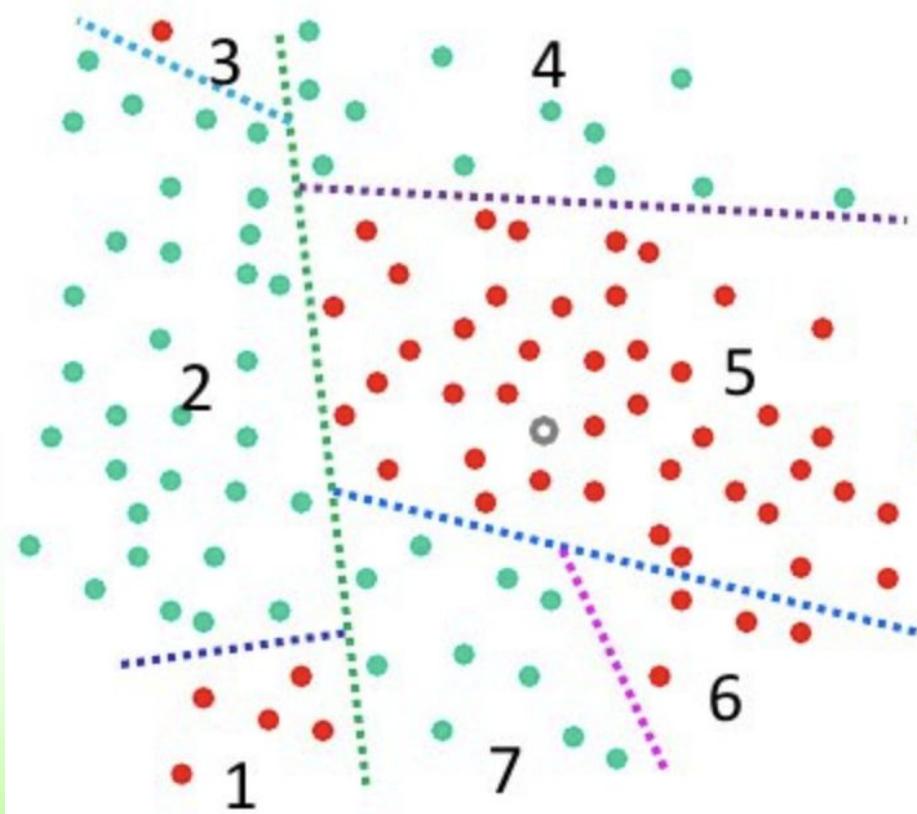
- для листовых вершин заданы прогнозы  $C_v$



# Решающие деревья

Предикаты:

$$\beta_v(x) = [x_j < t]$$



# Обучение решающих деревьев

$F(m, R_m)$

# Обучение решающих деревьев

$F(m, R_m)$

$m$  – номер вершины

$R_m$  - множество объектов в вершине

# Обучение решающих деревьев

$F(m, R_m)$

$m$  – номер вершины

$R_m$  - множество объектов в вершине

если выполнен критерий остановки – выдать прогноз  $C_m$

# Обучение решающих деревьев

$F(m, R_m)$

$m$  – номер вершины

$R_m$  - множество объектов в вершине

если выполнен критерий остановки – выдать прогноз  $C_m$

$j_i, t_i = \operatorname{argmax}_{j,t} Q(R_m, j, t)$

# Обучение решающих деревьев

$F(m, R_m)$

$m$  – номер вершины

$R_m$  - множество объектов в вершине

если выполнен критерий остановки – выдать прогноз  $C_m$

$j_i, t_i = \operatorname{argmax}_{j,t} Q(R_m, j, t)$

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

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

# Обучение решающих деревьев

$F(m, R_m)$

$m$  – номер вершины

$R_m$  - множество объектов в вершине

если выполнен критерий остановки – выдать прогноз  $C_m$

$j_i, t_i = \operatorname{argmax}_{j,t} Q(R_m, j, t)$

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

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

$F(l, R_l)$

$F(r, R_r)$

# Обучение решающих деревьев

Критерий остановки:

# Обучение решающих деревьев

Критерий остановки:

- если все объекты в листе относятся к одному классу

# Обучение решающих деревьев

Критерий остановки:

- если все объекты в листе относятся к одному классу
- если достигли максимальной глубины

# Обучение решающих деревьев

Критерий остановки:

- если все объекты в листе относятся к одному классу
- если достигли максимальной глубины
- если достигли максимального количества листьев

# Обучение решающих деревьев

Критерий остановки:

- если все объекты в листе относятся к одному классу
- если достигли максимальной глубины
- если достигли максимального количества листьев
- если объектов в вершине осталось мало

# Обучение решающих деревьев

Критерий остановки:

- если все объекты в листе относятся к одному классу
- если достигли максимальной глубины
- если достигли максимального количества листьев
- если объектов в вершине осталось мало
- если ошибка меньше заданного порога

# Обучение решающих деревьев

Прогноз -  $C_m$

# Обучение решающих деревьев

Прогноз -  $C_m$

- оптимальный константный прогноз на  $R_m$

# Обучение решающих деревьев

Прогноз -  $C_m$

- оптимальный константный прогноз на  $R_m$
- вероятности классов по долям в  $R_m$

# Обучение решающих деревьев

$H(R_m)$  – критерий информативности (impurity)

# Обучение решающих деревьев

$H(R_m)$  – критерий информативности (impurity)

$H(R_m)$  – большое, если большое разнообразие объектов

# Обучение решающих деревьев

$H(R_m)$  – критерий информативности (impurity)

$H(R_m)$  – большое, если большое разнообразие объектов

$H(R_m)$  – близкое к нулю, если небольшое разнообразие объектов

# Обучение решающих деревьев

$H(R_m)$  – критерий информативности (impurity)

$H(R_m)$  – большое, если большое разнообразие объектов

$H(R_m)$  – близкое к нулю, если небольшое разнообразие объектов

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} L(y, c)$$

# Обучение решающих деревьев

Задача регрессии

# Обучение решающих деревьев

Задача регрессии

$$L(y, c) = (y - c)^2$$

# Обучение решающих деревьев

Задача регрессии

$$L(y, c) = (y - c)^2$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - c)^2$$

# Обучение решающих деревьев

Задача регрессии

$$L(y, c) = (y - c)^2$$

$$\begin{aligned} H(R) &= \min_c \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - c)^2 = \\ &= \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - \bar{y})^2 \end{aligned}$$

$\bar{y}$  = средний ответ на R

# Обучение решающих деревьев

Задача регрессии

$$L(y, c) = (y - c)^2$$

$$\begin{aligned} H(R) &= \min_c \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - c)^2 = \\ &= \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - \bar{y})^2 \end{aligned}$$

$\bar{y}$  = средний ответ на R

$$\frac{1}{|R|} \sum_{(x,y) \in R} (y_i - \bar{y})^2$$

# Обучение решающих деревьев

Задача регрессии

$$L(y, c) = (y - c)^2$$

$$\begin{aligned} H(R) &= \min_c \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - c)^2 = \\ &= \frac{1}{|R|} \sum_{(x,y) \in R} (y_i - \bar{y})^2 \end{aligned}$$

$\bar{y}$  = средний ответ на R

$\frac{1}{|R|} \sum_{(x,y) \in R} (y_i - \bar{y})^2$  - дисперсия

# Обучение решающих деревьев

Задача классификации

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

$$L(y, c) = [y \neq c]$$

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

$$L(y, c) = [y \neq c]$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} [y \neq c]$$

# Обучение решающих деревьев

Задача классификации

1) Ошибка при прогнозе самого частого класса

$$L(y, c) = [y \neq c]$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} [y \neq c]$$

$$p_k = \frac{1}{|R|} \sum_{(x,y) \in R} [y = k] - \text{доля класса } k$$

$$k_* = \operatorname{argmax}_k p_k \quad - \text{самый популярный класс}$$

# Обучение решающих деревьев

Задача классификации

1) Ошибка при прогнозе самого частого класса

$$L(y, c) = [y \neq c]$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} [y \neq c]$$

$$p_k = \frac{1}{|R|} \sum_{(x,y) \in R} [y = k] - \text{доля класса } k$$

$$k_* = \operatorname{argmax}_k p_k \quad - \text{самый популярный класс}$$

$$H(R) = 1 - p_{k_*}$$

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

$$R = (1, 1, 1, 1, 1, 1)$$

$$H(R) = 0$$

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

$$R = (1, 1, 1, 1, 1, 1)$$

$$H(R) = 0$$

$$R = (1, 1, 1, 2, 2, 2)$$

$$H(R) = 0.5$$

# Обучение решающих деревьев

Задача классификации

- 1) Ошибка при прогнозе самого частого класса

$$R = (1, 1, 1, 1, 1, 1)$$
$$H(R) = 0$$

$$R = (1, 1, 1, 2, 2, 2)$$
$$H(R) = 0.5$$

$$R = (1, 1, 1, 2, 3, 4)$$
$$H(R) = 0.5$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^k (c_k - [y_i = k])^2$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^k (c_k - [y_i = k])^2$$

$c_* = (p_1, \dots, p_k)$  - оптимальные вероятности

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^k (c_k - [y_i = k])^2$$

$c_* = (p_1, \dots, p_k)$  - оптимальные вероятности

$$H(R) = \sum_{k=1}^k p_k(1 - p_k)$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^k (c_k - [y_i = k])^2$$

$c_* = (p_1, \dots, p_k)$  - оптимальные вероятности

$$H(R) = \sum_{k=1}^k p_k(1 - p_k) = \sum_{k=1}^k (p_k - p_k^2)$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$c = (c_1, \dots, c_k)$  - вероятности классов

$$\sum_{k=1}^k c_k = 1, c_k \geq 0$$

$$H(R) = \min_c \frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^k (c_k - [y_i = k])^2$$

$c_* = (p_1, \dots, p_k)$  - оптимальные вероятности

$$H(R) = \sum_{k=1}^k p_k (1 - p_k) = \sum_{k=1}^k (p_k - p_k^2)$$

$$H(R) = \sum_{k=1}^k p_k - \sum_{k=1}^k p_k^2 = 1 - \sum_{k=1}^k p_k^2$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$$R = (1, 1, 1, 2, 2, 2)$$

$$H(R) = 1 - (0.5^2 + 0.5^2) = 0.5$$

# Обучение решающих деревьев

Задача классификации

2) Критерий Джини (Gini)

$$R = (1, 1, 1, 2, 2, 2)$$
$$H(R) = 1 - (0.5^2 + 0.5^2) = 0.5$$

$$R = (1, 1, 1, 2, 3, 4)$$
$$H(R) = 1 - \left( (3/6)^2 + (1/6)^2 + (1/6)^2 + (1/6)^2 \right) = 0.66$$

# Обучение решающих деревьев

Задача классификации

3) Энтропийный критерий (entropy)

# Обучение решающих деревьев

Задача классификации

3) Энтропийный критерий (entropy)

$$H(R) = \min_c \left( -\frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^K [y_i = k] \log c_k \right)$$

# Обучение решающих деревьев

Задача классификации

3) Энтропийный критерий (entropy)

$$H(R) = \min_c \left( -\frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^K [y_i = k] \log c_k \right) = \\ = -\sum_{k=1}^K p_k \log p_k \text{ - энтропия распределения классов}$$

# Обучение решающих деревьев

Задача классификации

3) Энтропийный критерий (entropy)

$$H(R) = \min_c \left( -\frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^K [y_i = k] \log c_k \right) = \\ = -\sum_{k=1}^K p_k \log p_k \text{ - энтропия распределения классов}$$

$H(R) = 0$ , если только 1 класс

# Обучение решающих деревьев

Задача классификации

3) Энтропийный критерий (entropy)

$$H(R) = \min_c \left( -\frac{1}{|R|} \sum_{(x,y) \in R} \sum_{k=1}^K [y_i = k] \log c_k \right) = \\ = -\sum_{k=1}^K p_k \log p_k \text{ - энтропия распределения классов}$$

$H(R) = 0$ , если только 1 класс

$H(R)$  = большое, если равномерное распределение объектов

# Обучение решающих деревьев

Выбор лучшего предиката

# Обучение решающих деревьев

Выбор лучшего предиката

$$Q(R_m, j, t) = H(R_m) - \frac{|R_l|}{|R_m|} H(R_l) - \frac{|R_r|}{|R_m|} H(R_r) \rightarrow \max$$

# Обучение решающих деревьев

Выбор лучшего предиката

$$Q(R_m, j, t) = H(R_m) - \frac{|R_l|}{|R_m|}H(R_l) - \frac{|R_r|}{|R_m|}H(R_r) \rightarrow \max$$

⇒ чем больше изменились(уменьшились) критерии информативности у дочерних вершин, по сравнению с родительской вершиной, тем лучше.

# Обработка пропусков в решающих деревьях

1) считаем  $Q(R_m, j, t)$  только для известных объектов

# Обработка пропусков в решающих деревьях

- 1) считаем  $Q(R_m, j, t)$  только для известных объектов
- 2) если  $Q(R_m, j, t)$  оказался лучшим, отправляем пропуски и в  $R_l$  и в  $R_r$

# Обработка пропусков в решающих деревьях

- 1) считаем  $Q(R_m, j, t)$  только для известных объектов
- 2) если  $Q(R_m, j, t)$  оказался лучшим, отправляем пропуски и в  $R_l$  и в  $R_r$

Предсказание на новых объектах:

$$a_m(x) = \frac{|R_l|}{|R_m|} a_l(x) + \frac{|R_r|}{|R_m|} a_r(x)$$

# Обработка пропусков в решающих деревьях

Суррогатный предикат:

# Обработка пропусков в решающих деревьях

Суррогатный предикат:

для  $[x_j < t]$  найдем другой признак  $[x_j < \tilde{t}]$ , который дает похожее разбиение (похожие объекты отправляются в лево и право)

# Обработка пропусков в решающих деревьях

Суррогатный предикат:

для  $[x_j < t]$  найдем другой признак  $[x_j < \tilde{t}]$ , который дает похожее разбиение (похожие объекты отправляются в лево и право)

$[x_j < \tilde{t}]$  – суррогатный предикат

# Работа с категориальными признаками в решающих деревьях

# Работа с категориальными признаками в решающих деревьях

1) предикат с числом потомков, равному числу категорий признака

# Работа с категориальными признаками в решающих деревьях

1) предикат с числом потомков, равному числу категорий признака

$$Q(R_m, j, t) = H(R_m) - \frac{|R_1|}{|R_m|}H(R_1) - \dots - \frac{|R_k|}{|R_m|}H(R_k)$$

# Работа с категориальными признаками в решающих деревьях

1) предикат с числом потомков, равному числу категорий признака

$$Q(R_m, j, t) = H(R_m) - \frac{|R_1|}{|R_m|}H(R_1) - \dots - \frac{|R_k|}{|R_m|}H(R_k)$$

- быстро растет объем дерева

# Работа с категориальными признаками в решающих деревьях

1) предикат с числом потомков, равному числу категорий признака

$$Q(R_m, j, t) = H(R_m) - \frac{|R_1|}{|R_m|}H(R_1) - \dots - \frac{|R_k|}{|R_m|}H(R_k)$$

- быстро растет объем дерева
- категориальный признак будет выбираться чаще, чем числовой

# Работа с категориальными признаками в решающих деревьях

Д) найдем лучшее разбиение среди попарных разбиений категориального признака

# Работа с категориальными признаками в решающих деревьях

2) найдем лучшее разбиение среди попарных разбиений категориального признака

$$\{u_1, \dots, u_k\}, \{u_{k+1}, \dots, u_n\}$$

# Работа с категориальными признаками в решающих деревьях

4) найдем лучшее разбиение среди попарных разбиений категориального признака

$$\{u_1, \dots, u_k\}, \{u_{k+1}, \dots, u_n\}$$

$$\beta_j(x) = [x_j \in \{u_1, \dots, u_k\}]$$

# Работа с категориальными признаками в решающих деревьях

2) найдем лучшее разбиение среди попарных разбиений категориального признака

$$\{u_1, \dots, u_k\}, \{u_{k+1}, \dots, u_n\}$$

$$\beta_j(x) = [x_j \in \{u_1, \dots, u_k\}]$$

- перебор большого числа вариантов

# Работа с категориальными признаками в решающих

**деревьях**

Бинарная классификация

# Работа с категориальными признаками в решающих деревьях

3) бинарная классификация

$R_m(u)$  – объекты, в вершине  $m$ , у которых  $j$  – й признак =  $u$

# Работа с категориальными признаками в решающих деревьях

3) бинарная классификация

$R_m(u)$  – объекты, в вершине  $m$ , у которых  $j$  – й признак =  $u$

$N_m(u) = |R_m(u)|$  - число таких объектов

# Работа с категориальными признаками в решающих деревьях

3) бинарная классификация

$R_m(u)$  – объекты, в вершине  $m$ , у которых  $j$  – й признак =  $u$

$N_m(u) = |R_m(u)|$  - число таких объектов

упорядочим по доле положительных объектов в категориях  $u_{(1)}, \dots, u_{(q)}$ :

$$\frac{1}{N_m(u_{(1)})} \sum_{x_i \in R_m(u_{(1)})} [y_i = +1] \leq \dots \leq \frac{1}{N_m(u_{(q)})} \sum_{x_i \in R_m(u_{(q)})} [y_i = +1]$$

# Работа с категориальными признаками в решающих

**деревьях**

Бинарная классификация

# Работа с категориальными признаками в решающих деревьях

3) бинарная классификация

Заменим категории числами:

$$u_{(1)} \rightarrow 1$$

$$u_{(2)} \rightarrow 2$$

...

$$u_{(q)} \rightarrow q$$

# Работа с категориальными признаками в решающих деревьях

3) бинарная классификация

Заменим категории числами:

$$u_{(1)} \rightarrow 1$$

$$u_{(2)} \rightarrow 2$$

...

$$u_{(q)} \rightarrow q$$

и разбиваем как числовой признак

# Работа с категориальными признаками в решающих деревьях

Д) регрессия

# Работа с категориальными признаками в решающих деревьях

Д) регрессия

сортируем по среднему значению целевой переменной в категории

# Работа с категориальными признаками в решающих деревьях

сортируем по среднему значению целевой переменной в категории

$$\frac{1}{N_m(u_{(1)})} \sum_{x_i \in R_m(u_{(1)})} y_i \leq \dots \leq \frac{1}{N_m(u_{(q)})} \sum_{x_i \in R_m(u_{(q)})} y_i$$

# Стрижка деревьев

Стрижка деревьев – pruning

# Стрижка деревьев

Стрижка деревьев – pruning

- 1) Построим дерево, в каждом листе которого находится только один класс

# Стрижка деревьев

Стрижка деревьев – pruning

1) Построим дерево, в каждом листе которого находится только один класс

2) Введем новый функционал, добавив штраф за количество листьев

$$R_\alpha(T) = R(T) + \alpha|T|$$

# Стрижка деревьев

Стрижка деревьев – pruning

1) Построим дерево, в каждом листе которого находится только один класс

2) Введем новый функционал, добавив штраф за количество листьев

$$R_\alpha(T) = R(T) + \alpha|T|$$

$|T|$  - число листьев в поддереве Т

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$X_1, \dots, X_n$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

$p(x)$  – распределение на  $X$

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

$p(x)$  – распределение на  $X$

$y(x)$  – правильных ответ на  $x$

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

$p(x)$  – распределение на  $X$

$y(x)$  – правильных ответ на  $x$

$$E_x(b_j(x) - y(x))^2 = E_x \varepsilon_j^2(x) – ошибка b_j(x) на всем X$$

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

$p(x)$  – распределение на  $X$

$y(x)$  – правильных ответ на  $x$

$E_x(b_j(x) - y(x))^2 = E_x \varepsilon_j^2(x)$  – ошибка  $b_j(x)$  на всем  $X$

Предположим что:

$E_x \varepsilon_j(x) = 0$  – ошибки несмещенные

# Композиции моделей

$$X = (x_i, y_i)_{i=1}^l$$

bootstrap:

из  $X$  делаем  $n$  случайных подвыборок с возвращением размера  $l$

$$X_1, \dots, X_n$$

$b_1(x), \dots, b_n(x)$  - модели на этих выборках

$p(x)$  – распределение на  $X$

$y(x)$  – правильных ответ на  $x$

$$E_x(b_j(x) - y(x))^2 = E_x \varepsilon_j^2(x) – ошибка b_j(x) на всем X$$

Предположим что:

$$E_x \varepsilon_j(x) = 0 – ошибки несмещенные$$

$$E_x \varepsilon_i(x) \varepsilon_j(x) = 0, i \neq j – ошибки некоррелированы$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2 = E_x \left( \frac{1}{n} \sum_{j=1}^n (b_j(x) - y(x)) \right)^2$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$\begin{aligned} E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2 &= E_x \left( \frac{1}{n} \sum_{j=1}^n (b_j(x) - y(x)) \right)^2 = \\ &= E_x \left( \frac{1}{n} \sum_{j=1}^n \varepsilon_j(x) \right)^2 \end{aligned}$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$\begin{aligned} E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2 &= E_x \left( \frac{1}{n} \sum_{j=1}^n (b_j(x) - y(x)) \right)^2 = \\ &= E_x \left( \frac{1}{n} \sum_{j=1}^n \varepsilon_j(x) \right)^2 = \frac{1}{n^2} \left( E_x \sum_{j=1}^n \varepsilon_j^2(x) + \sum_{j \neq k} E_x \varepsilon_i(x) \varepsilon_j(x) \right) \end{aligned}$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$\begin{aligned} E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2 &= E_x \left( \frac{1}{n} \sum_{j=1}^n (b_j(x) - y(x)) \right)^2 = \\ &= E_x \left( \frac{1}{n} \sum_{j=1}^n \varepsilon_j(x) \right)^2 = \frac{1}{n^2} \left( E_x \sum_{j=1}^n \varepsilon_j^2(x) + \sum_{j \neq k} E_x \varepsilon_i(x) \varepsilon_j(x) \right) = \\ &= \frac{1}{n^2} E_x \sum_{j=1}^n \varepsilon_j^2(x) \end{aligned}$$

# Композиции моделей

$$a(x) = \frac{1}{n} \sum_{j=1}^n b_j(x)$$

$$\begin{aligned} E_x \left( \frac{1}{n} \sum_{j=1}^n b_j(x) - y(x) \right)^2 &= E_x \left( \frac{1}{n} \sum_{j=1}^n (b_j(x) - y(x)) \right)^2 = \\ &= E_x \left( \frac{1}{n} \sum_{j=1}^n \varepsilon_j(x) \right)^2 = \frac{1}{n^2} \left( E_x \sum_{j=1}^n \varepsilon_j^2(x) + \sum_{j \neq k} E_x \varepsilon_i(x) \varepsilon_j(x) \right) = \\ &= \frac{1}{n^2} E_x \sum_{j=1}^n \varepsilon_j^2(x) = \frac{1}{n} E_x \varepsilon_j^2(x) \end{aligned}$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

$$R(a) = E_{x,y}(y - a(x))^2$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

$$R(a) = E_{x,y}(y - a(x))^2 =$$

$$= \int_X \int_y p(x, y) (y - a(x))^2 dx dy$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

$$R(a) = E_{x,y}(y - a(x))^2 =$$

$$= \int_X \int_y p(x, y) (y - a(x))^2 dx dy \text{ – критерий ошибки (среднеквадратичный риск)}$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

$$R(a) = E_{x,y}(y - a(x))^2 =$$

$$= \int_X \int_y p(x, y) (y - a(x))^2 dx dy \text{ – критерий ошибки (среднеквадратичный риск)}$$

Свойства среднеквадратичного риска:

$$1) R(a) = E_{x,y}(y - E(y|x))^2 + E_{x,y}(E(y|x) - a(x))^2$$

# Разложение ошибки на смещение и разброс

$$X = (x_i, y_i)_{i=1}^l$$

$y \in R$  – задача регрессии

$p(x, y)$  – распределение на котором получена выборка

$$L(y, c) = (y - c)^2$$

$$R(a) = E_{x,y}(y - a(x))^2 =$$

$= \int_X \int_y p(x, y) (y - a(x))^2 dx dy$  – критерий ошибки (среднеквадратичный риск)

Свойства среднеквадратичного риска:

$$1) R(a) = E_{x,y}(y - E(y|x))^2 + E_{x,y}(E(y|x) - a(x))^2$$

2) оптимальный алгоритм по среднеквадратичному риску

$$a_*(x) = \int_y y p(y|x) dy = E(y|x)$$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

$$p(X) = \prod_{i=1}^l p(x_i, y_i)$$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

$$p(X) = \prod_{i=1}^l p(x_i, y_i)$$

$$\begin{aligned} L(\mu) &= E_{x,y} (y - E(y|x))^2 + \\ &+ E_x (E_X \mu(X)(x) - E(y|x))^2 + \\ &+ E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 \end{aligned}$$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

$$p(X) = \prod_{i=1}^l p(x_i, y_i)$$

$$\begin{aligned} L(\mu) &= E_{x,y} (y - E(y|x))^2 + \quad \text{шум(noise)} \\ &+ E_x (E_X \mu(X)(x) - E(y|x))^2 + \\ &+ E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 \end{aligned}$$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

$$p(X) = \prod_{i=1}^l p(x_i, y_i)$$

$$\begin{aligned} L(\mu) &= E_{x,y} (y - E(y|x))^2 + && \text{шум(noise)} \\ &+ E_x (E_X \mu(X)(x) - E(y|x))^2 + && \text{смещение(bias)} \\ &+ E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 \end{aligned}$$

# Разложение ошибки на смещение и разброс

$\mu$  – метод обучения

$L(\mu) = E_X E_{x,y} (y - \mu(X)(x))^2$  - ошибка метода обучения  $\mu$

$$p(X) = \prod_{i=1}^l p(x_i, y_i)$$

$L(\mu) = E_{x,y} (y - E(y|x))^2 +$       шум(noise)  
+  $E_x (E_X \mu(X)(x) - E(y|x))^2 +$       смещение(bias)  
+  $E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2$       разброс(variance, дисперсия)

# Разложение ошибки на смещение и разброс

$$L(\mu) = E_{x,y}(y - E(y|x))^2 + \text{шум(noise)}$$

$$+ E_x(E_X\mu(X)(x) - E(y|x))^2 + \text{смещение(bias)}$$

$$+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2 \text{ разброс(variance)}$$

# Разложение ошибки на смещение и разброс

$$L(\mu) = E_{x,y}(y - E(y|x))^2 + \text{шум(noise)}$$

$$+ E_x(E_X\mu(X)(x) - E(y|x))^2 + \text{смещение(bias)}$$

$$+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2 \text{ разброс(variance)}$$

$E(y|x)$  – лучшая модель

# Разложение ошибки на смещение и разброс

$$L(\mu) = E_{x,y}(y - E(y|x))^2 + \text{шум(noise)}$$

$$+ E_x(E_X\mu(X)(x) - E(y|x))^2 + \text{смещение(bias)}$$

$$+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2 \text{ разброс(variance)}$$

$E(y|x)$  – лучшая модель

$E_x\mu(X)(x)$  - средняя обученная модель

# Разложение ошибки на смещение и разброс

$$L(\mu) = E_{x,y}(y - E(y|x))^2 + \text{шум(noise)}$$

$$+ E_x(E_X\mu(X)(x) - E(y|x))^2 + \text{смещение(bias)}$$

$$+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2 \text{ разброс(variance)}$$

$E(y|x)$  – лучшая модель

$E_x\mu(X)(x)$  - средняя обученная модель

$\mu(X)(x)$  - модель на выборке X

# Разложение ошибки на смещение и разброс

$L(\mu) = E_{x,y}(y - E(y|x))^2 +$  шум(noise) - ошибка лучшей модели (показывает насколько трудно решается задача)

+  $E_x(E_X\mu(X)(x) - E(y|x))^2 +$  смещение(bias)

+  $E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2$  разброс(variance)

$E(y|x)$  – лучшая модель

$E_x\mu(X)(x)$  - средняя обученная модель

$\mu(X)(x)$  - модель на выборке X

# Разложение ошибки на смещение и разброс

$$L(\mu) = E_{x,y}(y - E(y|x))^2 + \text{шум(noise)} - \text{ошибка лучшей модели (показывает насколько трудно решается задача)}$$
$$+ E_x(E_X\mu(X)(x) - E(y|x))^2 + \text{смещение(bias) – как модели в среднем отклоняются от лучшего прогноза}$$
$$+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2 \quad \text{разброс(variance)}$$

$E(y|x)$  – лучшая модель

$E_x\mu(X)(x)$  - средняя обученная модель

$\mu(X)(x)$  - модель на выборке X

# Разложение ошибки на смещение и разброс

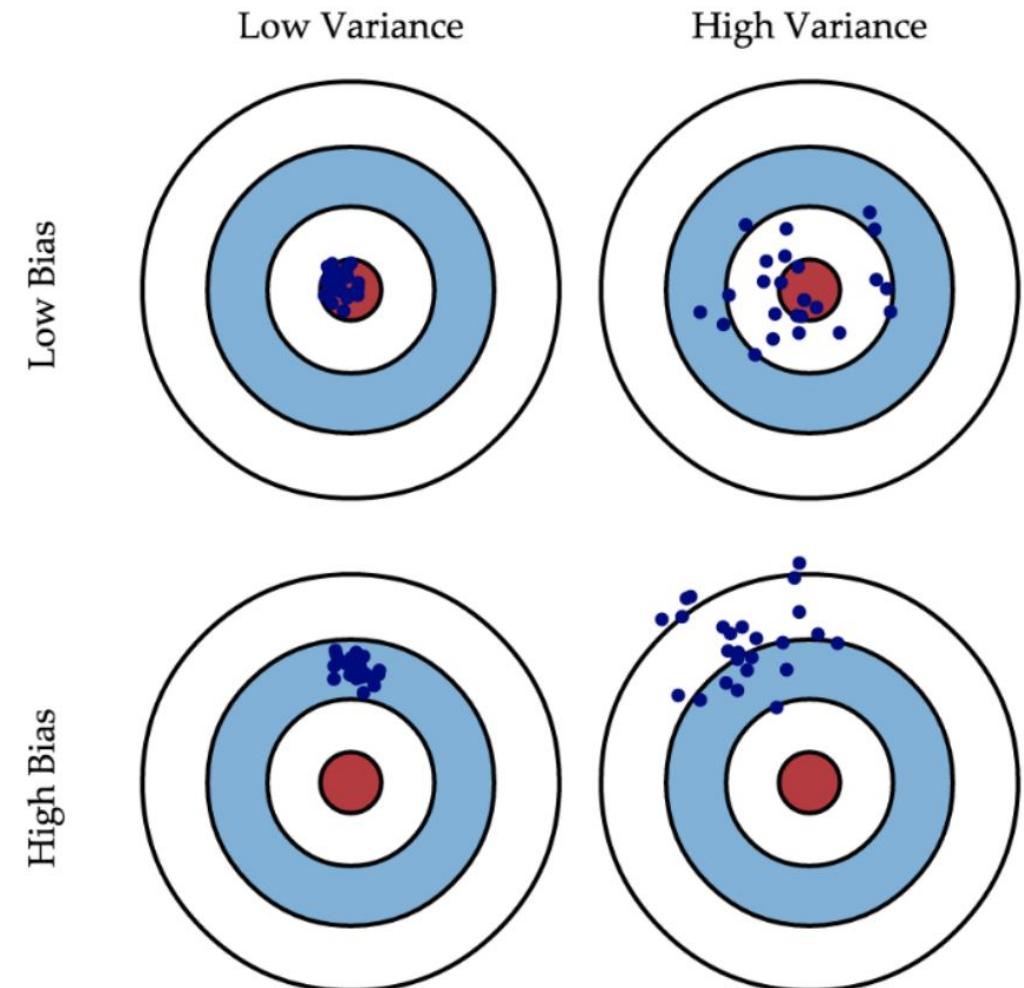
- $L(\mu) = E_{x,y}(y - E(y|x))^2 +$  шум(noise) - ошибка лучшей модели (показывает насколько трудно решается задача)
- $+ E_x(E_X\mu(X)(x) - E(y|x))^2 +$  смещение(bias) – как модели в среднем отклоняются от лучшего прогноза
- $+ E_xE_X(\mu(X)(x) - E_x\mu(X)(x))^2$  разброс(variance) – отклонение модели на X от средней модели (показывает чувствительность к изменениям в выборке)

$E(y|x)$  – лучшая модель

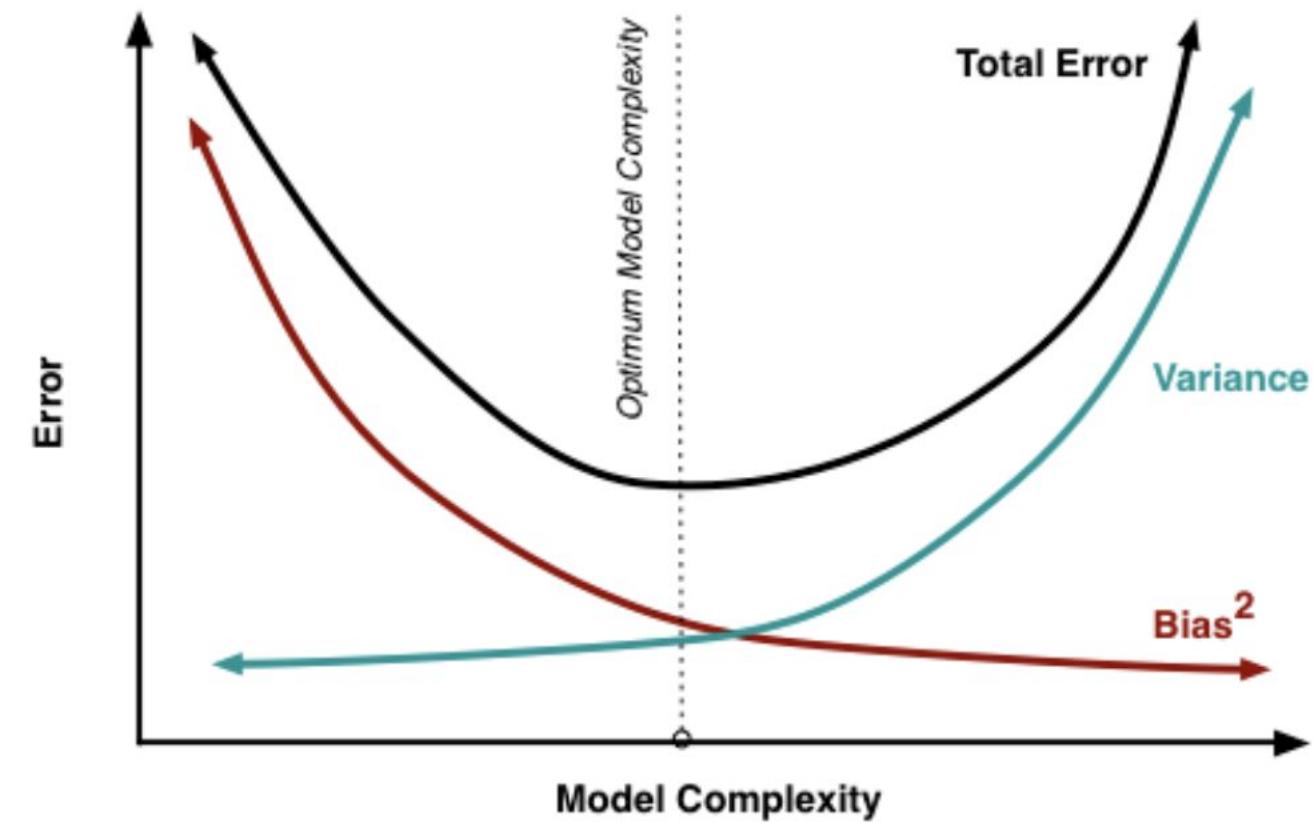
$E_x\mu(X)(x)$  - средняя обученная модель

$\mu(X)(x)$  - модель на выборке X

# Разложение ошибки на смещение и разброс



# Разложение ошибки на смещение и погрешность



# БЭГГИНГ

# БЭГГИНГ

bagging – bootstrap aggregating

# БЭГГИНГ

bagging – bootstrap aggregating

Введем метод обучения  $\tilde{\mu}$ :

- 1) генерируем бутстррапом подвыборку  $\tilde{X}$  из  $X$ ,  $|\tilde{X}| = l$

# БЭГГИНГ

bagging – bootstrap aggregating

Введем метод обучения  $\tilde{\mu}$ :

- 1) генерируем бутстррапом подвыборку  $\tilde{X}$  из  $X$ ,  $|\tilde{X}| = l$
- 2) обучаем  $b(x)$  с помощью  $\mu$  на  $\tilde{X}$

# БЭГГИНГ

bagging – bootstrap aggregating

Введем метод обучения  $\tilde{\mu}$ :

- 1) генерируем бутстррапом подвыборку  $\tilde{X}$  из  $X$ ,  $|\tilde{X}| = l$
- 2) обучаем  $b(x)$  с помощью  $\mu$  на  $\tilde{X}$

С помощью этого метода  $\tilde{\mu}$  строим  $n$  моделей

$b_1, \dots, b_n$  - базовые модели

# БЭГГИНГ

bagging – bootstrap aggregating

Введем метод обучения  $\tilde{\mu}$ :

- 1) генерируем бутстррапом подвыборку  $\tilde{X}$  из  $X$ ,  $|\tilde{X}| = l$
- 2) обучаем  $b(x)$  с помощью  $\mu$  на  $\tilde{X}$

С помощью этого метода  $\tilde{\mu}$  строим  $n$  моделей

$b_1, \dots, b_n$  - базовые модели

Получаем композицию моделей:

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

# Случайный лес



# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

$$\begin{aligned} 2) \text{var } a(x) &= \frac{1}{n} E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 + \\ &+ \frac{n(n-1)}{n^2} E_x E_X (\widetilde{\mu_1}(X)(x) - E_x \widetilde{\mu_1}(X)(x)) (\widetilde{\mu_2}(X)(x) - E_x \widetilde{\mu_2}(X)(x)) \end{aligned}$$

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

2)  $\text{var } a(x) = \frac{1}{n} E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 +$  - разброс одной базовой модели  
+  $\frac{n(n-1)}{n^2} E_x E_X (\widetilde{\mu_1}(X)(x) - E_x \widetilde{\mu_1}(X)(x)) (\widetilde{\mu_2}(X)(x) - E_x \widetilde{\mu_2}(X)(x))$

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

2)  $\text{var } a(x) = \frac{1}{n} E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 +$  - разброс одной базовой модели  
 $+ \frac{n(n-1)}{n^2} E_x E_X (\widetilde{\mu_1}(X)(x) - E_x \widetilde{\mu_1}(X)(x)) (\widetilde{\mu_2}(X)(x) - E_x \widetilde{\mu_2}(X)(x))$  - ковариация моделей обученных с помощью  $\widetilde{\mu}$

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

2)  $\text{var } a(x) = \frac{1}{n} E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 +$  - разброс одной базовой модели  
 $+ \frac{n(n-1)}{n^2} E_x E_X (\widetilde{\mu_1}(X)(x) - E_x \widetilde{\mu_1}(X)(x)) (\widetilde{\mu_2}(X)(x) - E_x \widetilde{\mu_2}(X)(x))$  - ковариация моделей обученных с помощью  $\widetilde{\mu}$

$\Rightarrow n \rightarrow \infty$

# Случайный лес

1)  $\text{bias } a(x) = \text{bias } b_n(x)$

$\Rightarrow \mu$  должен быть с низким смещением, нужно использовать глубокие деревья

2)  $\text{var } a(x) = \frac{1}{n} E_x E_X (\mu(X)(x) - E_x \mu(X)(x))^2 +$  - разброс одной базовой модели  
 $+ \frac{n(n-1)}{n^2} E_x E_X (\widetilde{\mu_1}(X)(x) - E_x \widetilde{\mu_1}(X)(x)) (\widetilde{\mu_2}(X)(x) - E_x \widetilde{\mu_2}(X)(x))$  - ковариация моделей обученных с помощью  $\widetilde{\mu}$

$\Rightarrow n \rightarrow \infty$ , выходы моделей должны быть менее коррелированы

# Случайный лес

Случайный лес (Random Forest)

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом

- обучение на подвыборке признаков(часть базовых моделей может обучиться на плохих признаках, получится высокое смещение)

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом
- обучение на подвыборке признаков(часть базовых моделей может обучиться на плохих признаках, получится высокое смещение)  $\Rightarrow$  лучший предикат для каждой вершины выбираем из случайного подмножества признаков(для каждой вершины свое)

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом
- обучение на подвыборке признаков(часть базовых моделей может обучиться на плохих признаках, получится высокое смещение)  $\Rightarrow$  лучший предикат для каждой вершины выбираем из случайного подмножества признаков(для каждой вершины свое)

$m$  – число признаков, на которых выбираем лучший предикат

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом

- обучение на подвыборке признаков(часть базовых моделей может обучиться на плохих признаках, получится высокое смещение)  $\Rightarrow$  лучший предикат для каждой вершины выбираем из случайного подмножества признаков(для каждой вершины свое)

$m$  – число признаков, на которых выбираем лучший предикат

для классификации –  $m = \sqrt{d}$

# Случайный лес

Случайный лес (Random Forest)

Чтобы получить низкое смещение, нужно строить глубокие деревья.

Чтобы получить низкую ковариацию:

- обучение бутстррапом

- обучение на подвыборке признаков(часть базовых моделей может обучиться на плохих признаках, получится высокое смещение)  $\Rightarrow$  лучший предикат для каждой вершины выбираем из случайного подмножества признаков(для каждой вершины свое)

$m$  – число признаков, на которых выбираем лучший предикат

для классификации –  $m = \sqrt{d}$

для регрессии -  $m = \frac{d}{3}$

# Случайный лес

Out-of-bag estimation

# Случайный лес

Out-of-bag estimation

$X_n$  - выборка, на которой обучалась модель  $b_n(x)$

# Случайный лес

Out-of-bag estimation

$X_n$  - выборка, на которой обучалась модель  $b_n(x)$

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

# Случайный лес

Out-of-bag estimation

$X_n$  - выборка, на которой обучалась модель  $b_n(x)$

$$OOB = \sum_{i=1}^l l \left( y_i, \frac{1}{\sum_{n=1}^l [x_i \notin X_n]} \sum_{n=1}^l [x_i \notin X_n] b_n x_i \right) \rightarrow LOO, \text{ при } n \rightarrow \infty$$

# Градиентный бустинг

Сергей Смирнов

Учебный курс

Математическое моделирование

Методы машинного обучения

Градиентный бустинг

# Градиентный бустинг

Gradient boosting machine

# Градиентный бустинг

Gradient boosting machine

1) MSE

# Градиентный бустинг

Gradient boosting machine

1) MSE

$$a_n(x) = \sum_{n=1}^n b_n(x)$$

# Градиентный бустинг

Gradient boosting machine

1) MSE

$$a_n(x) = \sum_{n=1}^n b_n(x)$$

$$\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 \rightarrow \min$$

# Градиентный бустинг

Gradient boosting machine

1) MSE

$$a_n(x) = \sum_{n=1}^n b_n(x)$$

$$\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 \rightarrow \min$$

$b_1(x)$ :

# Градиентный бустинг

Gradient boosting machine

1) MSE

$$a_n(x) = \sum_{n=1}^n b_n(x)$$

$$\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 \rightarrow \min$$

$b_1(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_1(x_i) - y_i)^2 \rightarrow \min$$

# Градиентный бустинг

Gradient boosting machine

1) MSE

$$a_n(x) = \sum_{n=1}^n b_n(x)$$

$$\frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 \rightarrow \min$$

$b_1(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_1(x_i) - y_i)^2 \rightarrow \min$$

$$S_i^{(1)} = y_i - b_1(x_i) \text{ - остаток, сдвиг}$$

# Градиентный бустинг

$b_2(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_2(x_i) - s_i^{(1)})^2 \rightarrow \min$$

# Градиентный бустинг

$b_2(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_2(x_i) - s_i^{(1)})^2 \rightarrow \min$$

Если  $b_2$  идеально обучится, то  $b_1 + b_2$

# Градиентный бустинг

$b_2(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_2(x_i) - s_i^{(1)})^2 \rightarrow \min$$

Если  $b_2$  идеально обучится, то  $b_1 + b_2 = y_i$

# Градиентный бустинг

$b_2(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_2(x_i) - s_i^{(1)})^2 \rightarrow \min$$

Если  $b_2$  идеально обучится, то  $b_1 + b_2 = y_i$

$$s_i^{(2)} = y_i - b_1(x_i) - b_2(x_i)$$

# Градиентный бустинг

$b_2(x)$ :

$$\frac{1}{l} \sum_{i=1}^l (b_2(x_i) - s_i^{(1)})^2 \rightarrow \min$$

Если  $b_2$  идеально обучится, то  $b_1 + b_2 = y_i$

$$s_i^{(2)} = y_i - b_1(x_i) - b_2(x_i)$$

...

$b_n(x)$ :

# Градиентный бустинг

2) Произвольная функция потерь

# Градиентный бустинг

2) Произвольная функция потерь

$L(y, z)$  – дифференцируемая функция потерь

# Градиентный бустинг

2) Произвольная функция потерь

$L(y, z)$  – дифференцируемая функция потерь

$b_1(x)$  – первая модель

# Градиентный бустинг

2) Произвольная функция потерь

$L(y, z)$  – дифференцируемая функция потерь

$b_1(x)$  – первая модель

обучаем  $b_n(x)$ :

$$a_{n-1}(x) = \sum_{n=1}^{n-1} b_n(x)$$

# Градиентный бустинг

2) Произвольная функция потерь

$L(y, z)$  – дифференцируемая функция потерь

$b_1(x)$  – первая модель

обучаем  $b_n(x)$ :

$$a_{n-1}(x) = \sum_{l=1}^{n-1} b_l(x)$$

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + b_n(x_i)) \rightarrow \min_{b_n(x)}$$

# Градиентный бустинг

2) Произвольная функция потерь

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + S_i) \rightarrow \min_{b_n(x)}$$

# Градиентный бустинг

2) Произвольная функция потерь

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + S_i) \rightarrow \min_{b_n(x)}$$

1)  $S_i = y_i - a_{n-1}(x_i)$  - не будет учитывать особенности функции потерь

# Градиентный бустинг

2) Произвольная функция потерь

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + S_i) \rightarrow \min_{b_n(x)}$$

1)  $S_i = y_i - a_{n-1}(x_i)$  - не будет учитывать особенности функции потерь

$$2) S_i = -\frac{\partial L(y_i, z)}{\partial z} \Big|_{z=a_{n-1}(x_i)}$$

z – значение композиции в предыдущей точке

# Градиентный бустинг

2) Произвольная функция потерь

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + S_i) \rightarrow \min_{b_n(x)}$$

1)  $S_i = y_i - a_{n-1}(x_i)$  - не будет учитывать особенности функции потерь

$$2) S_i = -\frac{\partial L(y_i, z)}{\partial z} \Big|_{z=a_{n-1}(x_i)}$$

z – значение композиции в предыдущей точке

$$\sum_{i=1}^l (b_n(x_i) - S_i)^2 \rightarrow \min_{b_n(x)} - \text{задача обучения } n\text{-й модели в градиентном бустинге}$$

# Градиентный бустинг

2) Произвольная функция потерь

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + S_i) \rightarrow \min_{b_n(x)}$$

1)  $S_i = y_i - a_{n-1}(x_i)$  - не будет учитывать особенности функции потерь

$$2) S_i = -\frac{\partial L(y_i, z)}{\partial z} \Big|_{z=a_{n-1}(x_i)}$$

z – значение композиции в предыдущей точке

$$\sum_{i=1}^l (b_n(x_i) - S_i)^2 \rightarrow \min_{b_n(x)} -$$
 задача обучения n-й модели в градиентном бустинге

Обучаем базовую модель на градиент функционала по выходам композиции.

# Градиентный бустинг

По сравнению с Random Forest:

# Градиентный бустинг

По сравнению с Random Forest:

-используем неглубокие деревья, тем самым обучение идет быстрее

# Градиентный бустинг

По сравнению с Random Forest:

- используем неглубокие деревья, тем самым обучение идет быстрее
- за счет корректировки ошибки, с каждой итерацией уменьшается смещение базовых моделей

# Градиентный бустинг

По сравнению с Random Forest:

- используем неглубокие деревья, тем самым обучение идет быстрее
- за счет корректировки ошибки, с каждой итерацией уменьшается смещение базовых моделей
- в градиентном бустинге количество гиперпараметров больше

# Градиентный бустинг

По сравнению с Random Forest:

- используем неглубокие деревья, тем самым обучение идет быстрее
- за счет корректировки ошибки, с каждой итерацией уменьшается смещение базовых моделей
- в градиентном бустинге количество гиперпараметров больше
- если в градиентном бустинге взять сложную модель, то ошибка сразу будет маленькой на обучающей выборке и не получится дальше корректировать модель

# Регуляризация градиентного бустинга

Сергей Смирнов  
smirnov.serg@yandex.ru

# Регуляризация градиентного бустинга

- 1) сокращение длины шага

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

$$a_n(x_i) = a_{n-1}(x_i) + \eta b_n(x_i)$$

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

$$a_n(x_i) = a_{n-1}(x_i) + \eta b_n(x_i)$$

$\eta \in (0,1]$  – длина шага

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

$$a_n(x_i) = a_{n-1}(x_i) + \eta b_n(x_i)$$

$\eta \in (0,1]$  – длина шага

Но при уменьшении шага растет количество базовых алгоритмов

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

$$a_n(x_i) = a_{n-1}(x_i) + \eta b_n(x_i)$$

$\eta \in (0,1]$  – длина шага

Но при уменьшении шага растет количество базовых алгоритмов

2) SGD

# Регуляризация градиентного бустинга

1) сокращение длины шага

-если используем простые модели, то они не очень хорошо улучшают ошибку на предыдущем шаге

-если используем сложные модели, то после добавления  $b_n(x)$  в сдвигах останется только шум

$$a_n(x_i) = a_{n-1}(x_i) + \eta b_n(x_i)$$

$\eta \in (0,1]$  – длина шага

Но при уменьшении шага растет количество базовых алгоритмов

2) SGD

$b_n$  обучаем на случайной подвыборке

# Градиентный бустинг над деревьями

$$b_n(x) = \sum_{j=1}^{J_n} b_{nj} [x \in R_{nj}]$$

$b_{nj}$  - прогноз j-го листа

$R_{nj}$  - j-я область

$$a_n(x_i) = a_{n-1}(x_i) + \sum_{j=1}^{J_n} b_{nj} [x \in R_{nj}]$$

подберем  $b_{nj}$  так, чтобы они были оптимальны с точки зрения исходной функции потерь L

# Градиентный бустинг над деревьями

$$\sum_{i=1}^l L(y_i, a_{n-1}(x_i) + \sum_{j=1}^{J_n} b_{nj}[x \in R_{nj}]) \rightarrow \min$$

Задача сводится к одномерным задачам по всем листьям

j=1,2,...,J<sub>n</sub>:

$$\sum_{(x_i, y_i) \in R_{nj}} L(y_i, a_{n-1}(x_i) + b_{nj}) \rightarrow \min$$

При такой модификации алгоритм быстрее сходится

# Градиентный бустинг над деревьями

В бэггинге:

# Градиентный бустинг над деревьями

В бэггинге:

-смещение не меняется

# Градиентный бустинг над деревьями

В бэггинге:

- смещение не меняется
- разброс уменьшается в зависимости от количества базовых алгоритмов и их скоррелированности

# Градиентный бустинг над деревьями

В бэггинге:

- смещение не меняется
- разброс уменьшается в зависимости от количества базовых алгоритмов и их скоррелированности

в бустинге:

# Градиентный бустинг над деревьями

В бэггинге:

- смещение не меняется
- разброс уменьшается в зависимости от количества базовых алгоритмов и их скоррелированности

в бустинге:

- смещение уменьшается с каждой новой базовой моделью

# Градиентный бустинг над деревьями

В бэггинге:

- смещение не меняется
- разброс уменьшается в зависимости от количества базовых алгоритмов и их скоррелированности

в бустинге:

- смещение уменьшается с каждой новой базовой моделью
- разброс увеличивается с каждой новой базовой моделью

# Градиентный бустинг над деревьями

В бэггинге:

- смещение не меняется
- разброс уменьшается в зависимости от количества базовых алгоритмов и их скоррелированности

в бустинге:

- смещение уменьшается с каждой новой базовой моделью
  - разброс увеличивается с каждой новой базовой моделью
- ⇒ нужно использовать неглубокие решающие деревья

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

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

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

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

$$a(x) = c(b_1(x), \dots b_n(x))$$

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

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

$a(x) = c(b_1(x), \dots b_n(x))$

$c(z_1, \dots z_n)$  - метамодель

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

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

$a(x) = c(b_1(x), \dots b_n(x))$

$c(z_1, \dots z_n)$  - метамодель

$$\frac{1}{l} \sum_{l=1}^l L(y_i, c(b_1(x), \dots b_n(x))) \rightarrow \min_c$$

# Блендинг и Стекинг

$b_1(x), \dots b_n(x)$

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

$a(x) = c(b_1(x), \dots b_n(x))$

$c(z_1, \dots z_n)$  - метамодель

$$\frac{1}{l} \sum_{l=1}^l L(y_i, c(b_1(x), \dots b_n(x))) \rightarrow \min_c$$

$b(x)$  - может переобучиться и следовательно распределение прогнозов на тесте будет не таким, как на обучении, если  $c$  обучена на той же выборке, что и  $b_1(x), \dots b_n(x)$ .

# Блендинг и Стекинг

Блендинг (blending)

# Блендинг и Стекинг

Блендинг (blending)

- делим исходную выборку  $X$  на две  $X_1$  и  $X_2$

# Блендинг и Стекинг

Блендинг (blending)

- делим исходную выборку  $X$  на две  $X_1$  и  $X_2$
- на  $X_1$  учим  $b_1(x), \dots b_n(x)$

# Блендинг и Стекинг

Блендинг (blending)

- делим исходную выборку  $X$  на две  $X_1$  и  $X_2$
- на  $X_1$  учим  $b_1(x), \dots b_n(x)$
- на  $X_2$  учим метамодель  $c$

# Блендинг и Стекинг

Стекинг (stacking)

# Блендинг и Стекинг

## Стекинг (stacking)

- делим исходную выборку  $X$  на  $k$  частей  $X_1, \dots, X_k$

# Блендинг и Стекинг

## Стекинг (stacking)

- делим исходную выборку  $X$  на  $k$  частей  $X_1, \dots, X_k$
- $b_n^{(k)}(x)$  учим на всей выборке без  $X_k$  (нужно обучить  $n^*k$  моделей)

# Блендинг и Стекинг

## Стекинг (stacking)

- делим исходную выборку  $X$  на  $k$  частей  $X_1, \dots, X_k$
- $b_n^{(k)}(x)$  учим на всей выборке без  $X_k$  (нужно обучить  $n^*k$  моделей)
- учим метамодель  $c$  на  $X_k$

# Блендинг и Стекинг

## Стекинг (stacking)

- делим исходную выборку  $X$  на  $k$  частей  $X_1, \dots, X_k$
- $b_n^{(k)}(x)$  учим на всей выборке без  $X_k$  (нужно обучить  $n^*k$  моделей)
- учим метамодель  $c$  на  $X_k$

$$\sum_{k=1}^k \sum_{(x_i, y_i) \in X_k} L(y_i, c(b_1^{(k)}(x), \dots, b_n^{(k)}(x))) \rightarrow \min_c$$

# Блендинг и Стекинг

Для чего используется:

# Блендинг и Стекинг

Для чего используется:

- 1) базовые модели из разных семейств

# Блендинг и Стекинг

Для чего используется:

- 1) базовые модели из разных семейств
- 2) разные наборы признаков у базовых моделей

# Вопросы