

# Основы машинного обучения

Лекция 14

Случайные леса. Градиентный бустинг.

Евгений Соколов

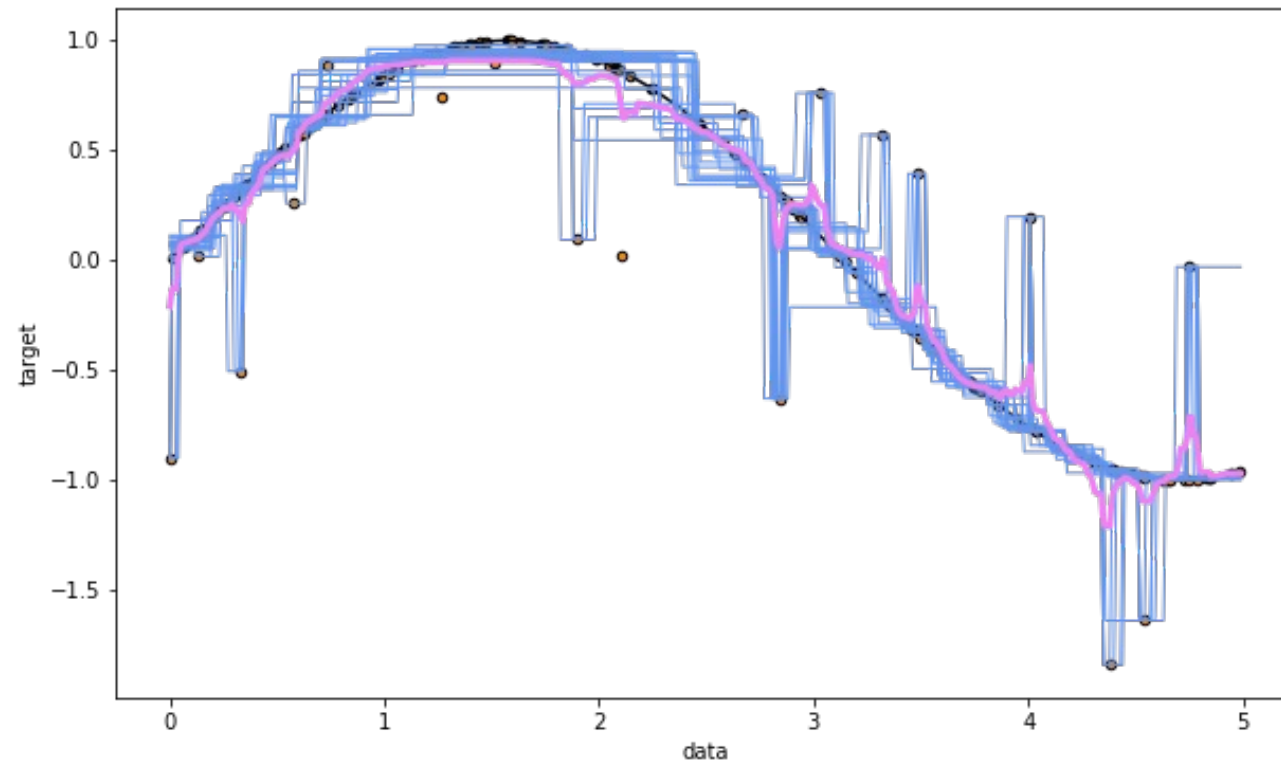
[esokolov@hse.ru](mailto:esokolov@hse.ru)

НИУ ВШЭ, 2025

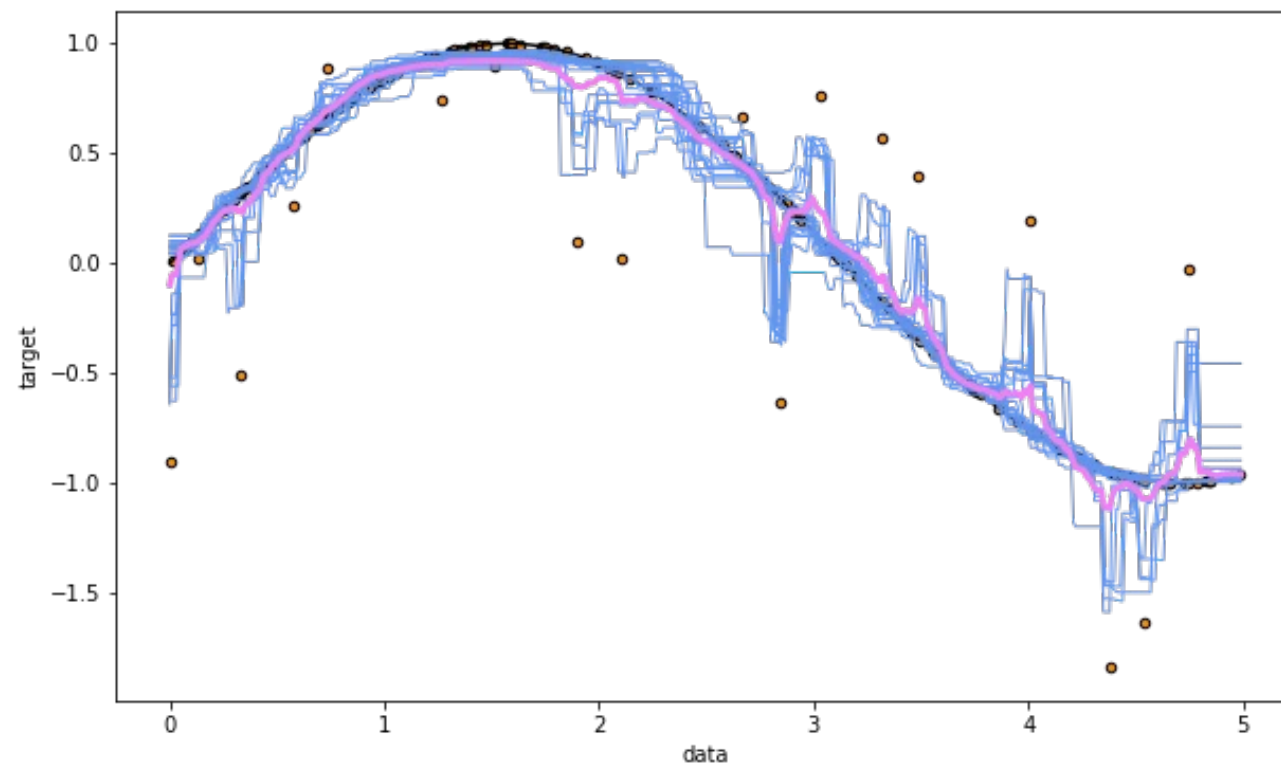
# Бэггинг

- Смещение  $a_N(x)$  такое же, как у  $b_n(x)$
- Разброс  $a_N(x)$ :
- $\frac{1}{N} (\text{разброс } b_n(x)) + \text{ковариация}(b_n(x), b_m(x))$
- Если базовые модели независимы, то разброс уменьшается в  $N$  раз!
- Чем более похожи выходы базовых моделей, тем меньше эффект от построения композиции

# Смещение и разброс: деревья



# Смещение и разброс: бэггинг



Случайный лес

# Жадный алгоритм

SplitNode( $m, R_m$ )

1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат:  $j, t = \arg \min_{j, t} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты:  $R_\ell = \{(x, y) \in R_m \mid [x_j < t]\}$ ,  
 $R_r = \{(x, y) \in R_m \mid [x_j \geq t]\}$
4. Повторяем для дочерних вершин: SplitNode( $\ell, R_\ell$ ) и SplitNode( $r, R_r$ )

# Жадный алгоритм

SplitNode( $m, R_m$ )

1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат:  $j, t = \arg \min_{j, t} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты:  $R_\ell = \{(x, y) \in R_m \mid [x_j < t]\}$ ,  
 $R_r = \{(x, y) \in R_m \mid [x_j \geq t]\}$
4. Повторяем для дочерних вершин: SplitNode( $\ell, R_\ell$ ) и SplitNode( $r, R_r$ )

# Выбор предиката

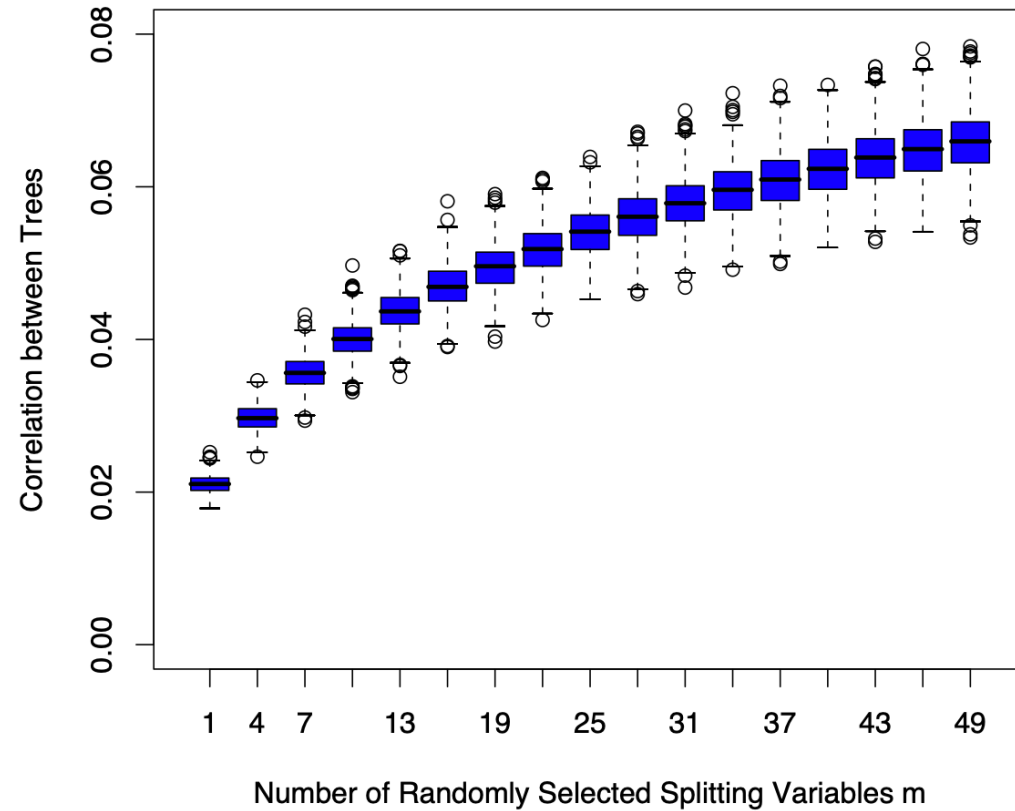
$$j, t = \arg \min_{j, t} Q(R_m, j, t)$$

- Будем искать лучший предикат среди случайного подмножества признаков размера  $q$





# Корреляция между деревьями



Hastie, Tibshirani, Friedman. The Elements of Statistical Learning.

# Корреляция между деревьями

Рекомендации для  $q$ :

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

# Случайный лес (Random Forest)

Для  $n = 1, \dots, N$ :

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

# Случайный лес (Random Forest)

Для  $n = 1, \dots, N$ :

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

Выбираются заново при каждом разбиении!

# Случайный лес (Random Forest)

- Регрессия:

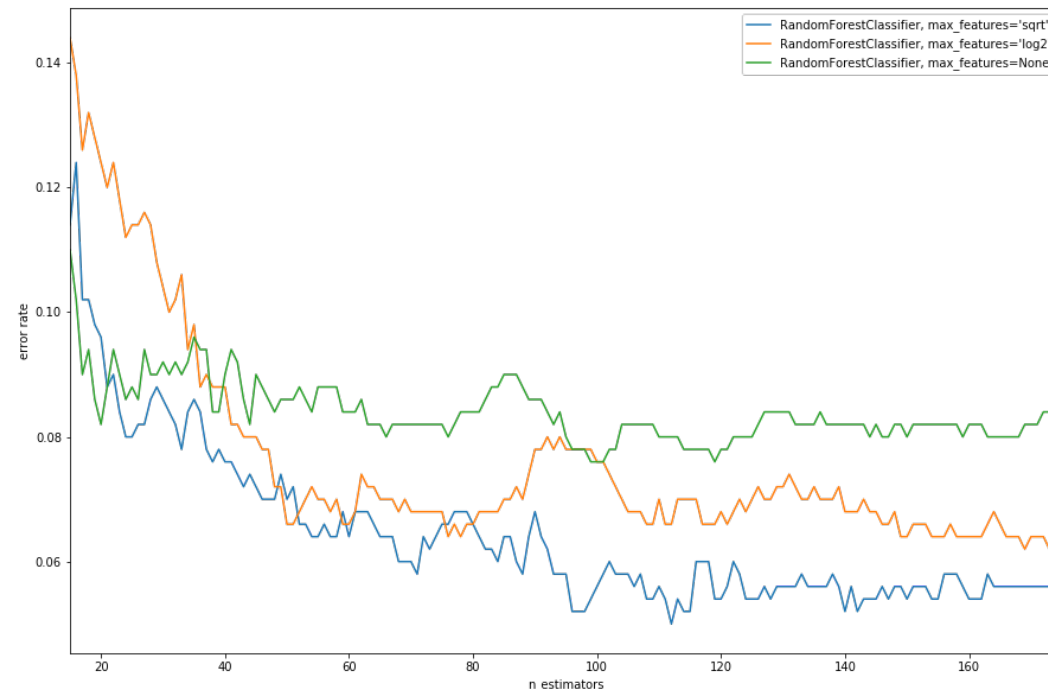
$$a(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

- Классификация:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{n=1}^N [b_n(x) = y]$$

# Универсальный метод

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



# Out-of-bag

- Каждое дерево обучается примерно на 63% данных
- Остальные объекты — как бы тестовая выборка для дерева
- $X_n$  — обучающая выборка для  $b_n(x)$
- Можно оценить ошибку на новых данных:

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

# Важность признаков

- Перестановочный метод для проверки важности  $j$ -го признака
- Перемешиваем соответствующий столбец в матрице «объекты-признаки» для тестовой выборки
- Измеряем качество модели
- Чем сильнее оно упало, тем важнее признак



# Резюме

- Случайный лес — метод на основе бэггинга, в котором делается попытка повысить разнообразие деревьев
- Метод практически без гиперпараметров
- Можно оценить обобщающую способность без тестовой выборки

Исправление ошибок моделей  
и идея бустинга

# Проблемы бэггинга

- Если базовая модель окажется смещённой, то и композиция не справится с задачей
- Базовые модели долго обучать и применять, дорого хранить

# Идея бустинга

- Возьмём простые базовые модели
- Будем строить композицию последовательно и жадно
- Каждая следующая модель будет строиться так, чтобы максимально корректировать ошибки построенных моделей

# Идея бустинга

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

- Обучение первой модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, b_1(x_i)) \rightarrow \min_{b_1(x)}$$

# Идея бустинга

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

# Идея бустинга

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

# Идея бустинга

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

- Непонятно, как обучать дерево на такое в общем случае



# Резюме

- В бустинге базовые модели обучаются последовательно
- Каждая следующая корректирует ошибки уже построенных
- В общем случае получается функционал, на который может быть сложно обучать деревья

Бустинг для  
среднеквадратичной ошибки

# Идея бустинга

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

# Бустинг для MSE

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (a_{N-1}(x_i) + b_N(x_i) - y_i)^2 \rightarrow \min_{b_N(x)}$$

# Бустинг для MSE

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - (y_i - a_{N-1}(x_i)) \right)^2 \rightarrow \min_{b_N(x)}$$

# Бустинг для MSE

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - \underbrace{(y_i - a_{N-1}(x_i))}_{s_i^{(N)}} \right)^2 \rightarrow \min_{b_N(x)}$$

# Бустинг для MSE

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - s_i^{(N)} \right)^2 \rightarrow \min_{b_N(x)}$$

- $s_i^{(N)} = y_i - a_{N-1}(x_i)$  — остатки

# Первая итерация

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (b_1(x_i) - y_i)^2 \rightarrow \min_{b_1(x)}$$



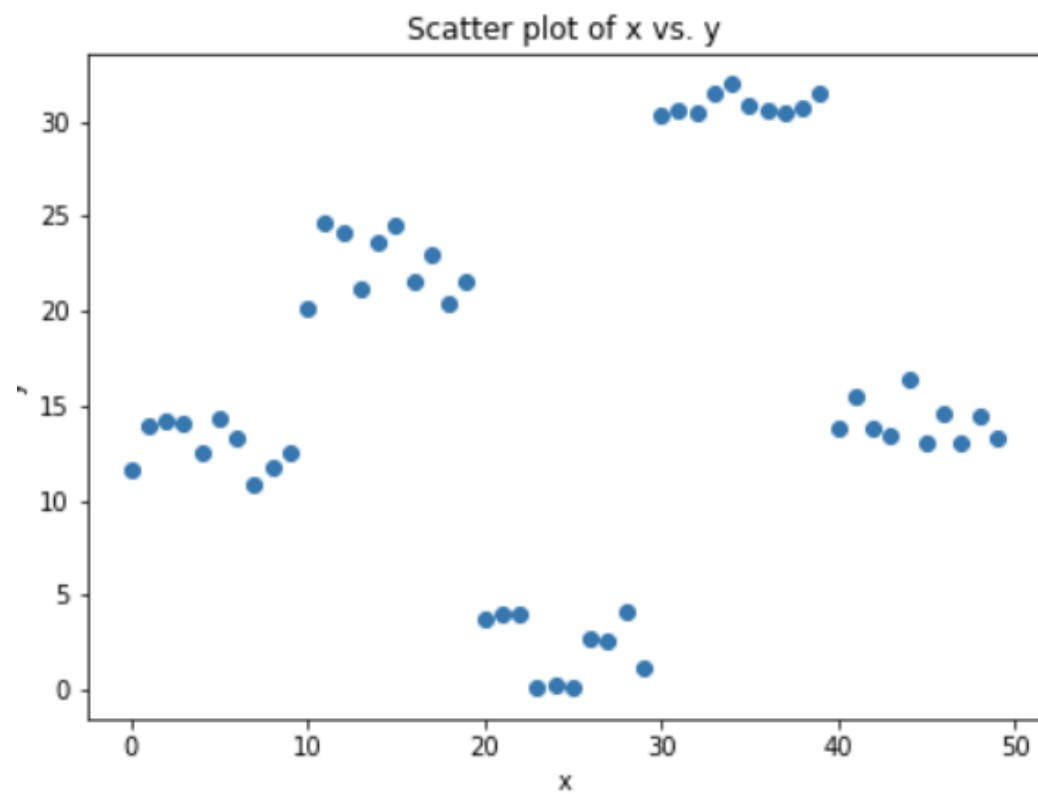
## Вторая итерация

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_2(x_i) - (y_i - b_1(x_i)) \right)^2 \rightarrow \min_{b_2(x)}$$

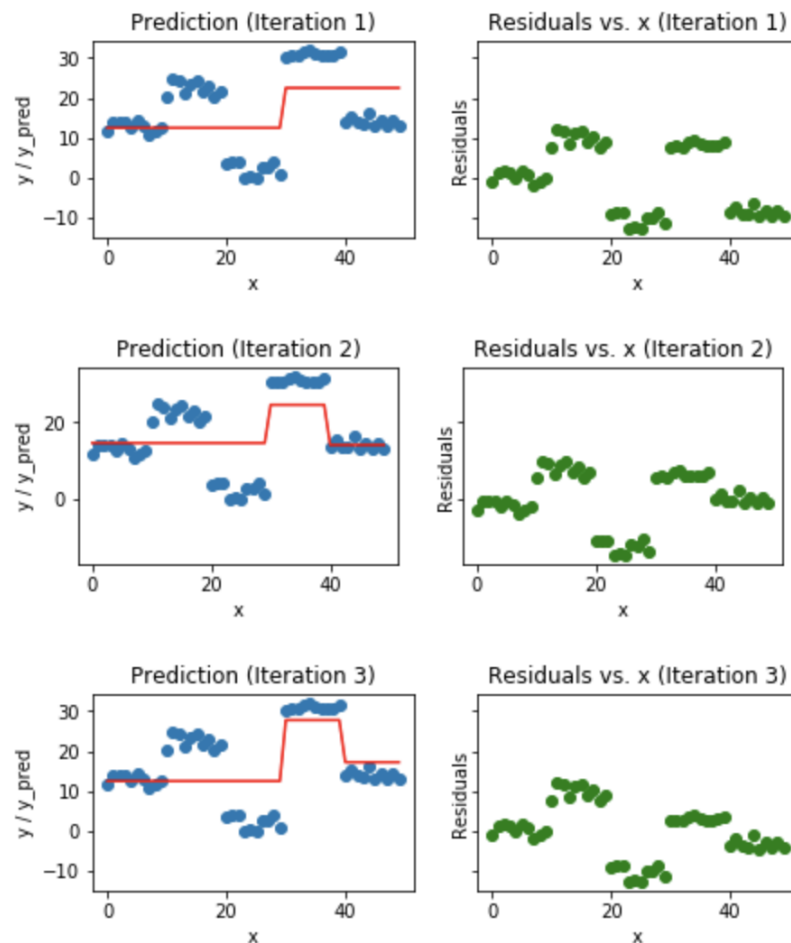
# Третья итерация

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_3(x_i) - (y_i - b_1(x_i) - b_2(x_i)) \right)^2 \rightarrow \min_{b_3(x)}$$

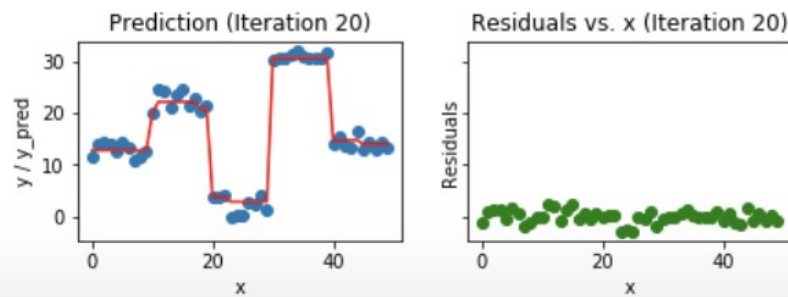
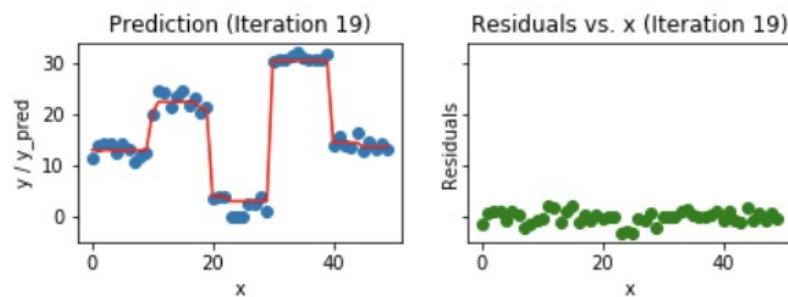
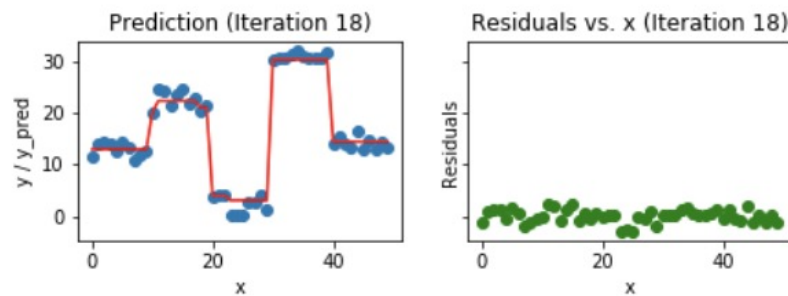
# Визуализация



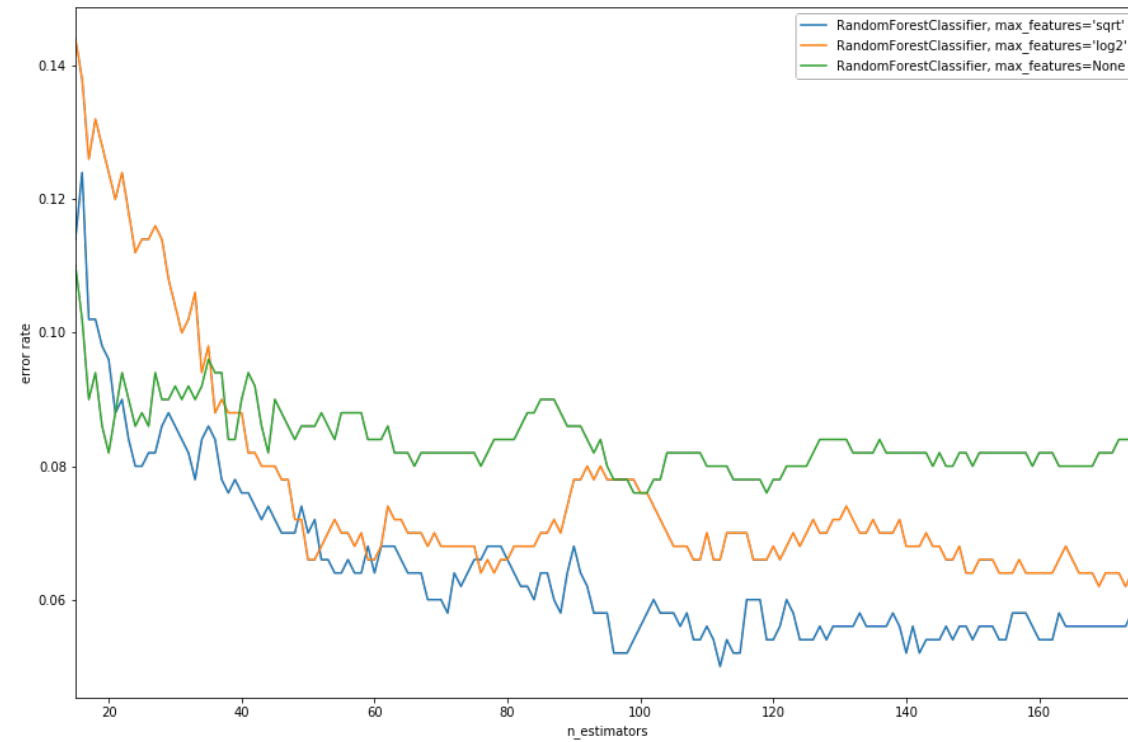
# Визуализация



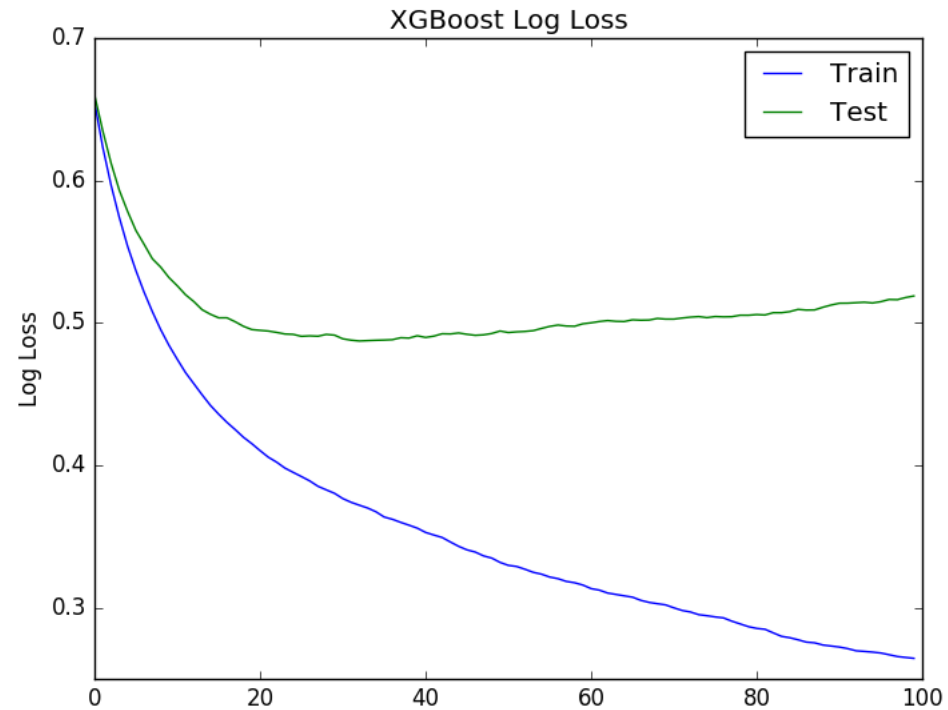
# Визуализация



# Random Forest



# Ошибка бустинга на обучении и тесте



# Резюме

- В случае с MSE обучение базовых моделей сводится к обычной процедуре обучения с заменой целевой переменной
- Бустинг может переобучаться, поэтому надо следить за ошибкой на тестовой выборке



Сложности с произвольной  
функцией потерь

# Задача обучения базовой модели

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

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

# Задача обучения базовой модели

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

- Может, просто обучаться на остатки, как в MSE?

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i - a_{N-1}(x_i), b_N(x_i)) \rightarrow \min_{b_N(x)}$$

# Логистическая функция потерь

$$a_N(x) = \text{sign} \sum_{n=1}^N b_n(x)$$

$$L(y, z) = \log(1 + \exp(-yz))$$

- Может, просто обучаться на остатки, как в MSE?

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \log \left( 1 + \exp \left( - (y_i - a_{N-1}(x_i)) b_N(x_i) \right) \right) \rightarrow \min_{b_N(x)}$$

- Если  $y_i = a_{N-1}(x_i)$ , то объект не участвует в обучении
- Иначе  $y_i - a_{N-1}(x_i) = \pm 2$

# Логистическая функция потерь

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \log \left( 1 + \exp \left( -\frac{y_i - a_{N-1}(x_i)}{2} b_N(x_i) \right) \right) \rightarrow \min_{b_N(x)}$$

- Если  $y_i = a_{N-1}(x_i)$ , то объект не участвует в обучении
- Если  $y_i \neq a_{N-1}(x_i)$ , то базовая модель учится выдавать корректный класс

# Логистическая функция потерь

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \log \left( 1 + \exp \left( - \frac{y_i - a_{N-1}(x_i)}{2} b_N(x_i) \right) \right) \rightarrow \min_{b_N(x)}$$

- $y_i = +1, \sum_{n=1}^{N-1} b_n(x_i) = -0.5 \rightarrow \text{надо } b_N(x_i) > 0.5$
- $y_i = +1, \sum_{n=1}^{N-1} b_n(x_i) = -100 \rightarrow \text{надо } b_N(x_i) > 100$
- Но на обоих объектах будет одинаково максимизироваться отступ
- На объектах с корректными ответами никак не контролируется выход  $b_N(x)$

# Резюме

- Нельзя заменить обучение добавки к композиции на обучение базовой модели на отклонение от ответов
- Не учитываются особенности функции потерь

# Градиентный бустинг в общем виде



# Задача обучения базовой модели

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

- Ошибка на объекте  $x_i$  при прогнозе новой модели, равном  $z$  :

$$L(y_i, a_{N-1}(x_i) + z)$$

- Как посчитать, куда и как сильно сдвигать  $a_{N-1}(x_i)$ , чтобы уменьшить ошибку?

# Задача обучения базовой модели

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N(x)}$$

- Ошибка на объекте  $x_i$  при прогнозе новой модели, равном  $z$  :

$$L(y_i, a_{N-1}(x_i) + z)$$

- Как посчитать, куда и как сильно сдвигать  $a_{N-1}(x_i)$ , чтобы уменьшить ошибку?
- Посчитать производную

# Задача обучения базовой модели

- Ошибка на объекте  $x_i$  при прогнозе новой модели, равном  $z$  :

$$L(y_i, a_{N-1}(x_i) + z)$$

- Посчитаем производную:

$$s_i^{(N)} = - \frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)}$$

# Задача обучения базовой модели

- Посчитаем производную:

$$s_i^{(N)} = -\frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)}$$

- Знак показывает, в какую сторону сдвигать прогноз на  $x_i$ , чтобы уменьшить ошибку композиции на нём
- Величина показывает, как сильно можно уменьшить ошибку, если сдвинуть прогноз
- Если ошибка почти не сдвинется, то нет смысла что-то менять

# Градиентный бустинг

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - s_i^{(N)} \right)^2 \rightarrow \min_{b_N(x)}$$

$$s_i^{(N)} = - \frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} \text{ — сдвиги}$$

# Градиентный бустинг

- Обучение  $N$ -й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - s_i^{(N)} \right)^2 \rightarrow \min_{b_N(x)}$$

$$s_i^{(N)} = - \frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} \text{ — сдвиги}$$

- Как бы градиентный спуск в пространстве ответов на обучающей выборке
- Базовая модель будет делать корректировки на объектах так, чтобы как можно сильнее уменьшить ошибку композиции
- Сдвиги учитывают особенности функции потерь

# Градиентный бустинг для MSE

$$\begin{aligned} s_i^{(N)} &= -\frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} = -\frac{\partial}{\partial z} \frac{1}{2} (z - y_i)^2 \Big|_{z=a_{N-1}(x_i)} = \\ &= -(a_{N-1}(x_i) - y_i) = y_i - a_{N-1}(x_i) \end{aligned}$$

# Градиентный бустинг для MSE

$$\begin{aligned} s_i^{(N)} &= -\frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} = -\frac{\partial}{\partial z} \frac{1}{2} (z - y_i)^2 \Big|_{z=a_{N-1}(x_i)} = \\ &= -(a_{N-1}(x_i) - y_i) = y_i - a_{N-1}(x_i) \end{aligned}$$

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - (y_i - a_{N-1}(x_i)) \right)^2 \rightarrow \min_{b_N(x)}$$



# Градиентный бустинг для MSE

$$s_i^{(N)} = y_i - a_{N-1}(x_i)$$

- $y_i = 10, a_{N-1}(x_i) = 5: s_i = 5$
- $y_i = 10, a_{N-1}(x_i) = 15: s_i = -5$

# Градиентный бустинг для асимметричной функции

$$L(y, z) = \frac{1}{2} ([z < y](z - y)^2 + 5[z \geq y](z - y)^2)$$

$$\begin{aligned} s_i^{(N)} &= - \frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} = \\ &= [z < y](y - z) + 5[z \geq y](y - z) \end{aligned}$$

# Градиентный бустинг для асимметричной функции

$$s_i^{(N)} = [z < y](y - z) + 5[z \geq y](y - z)$$

- $y_i = 10, a_{N-1}(x_i) = 5: s_i = 5$
- $y_i = 10, a_{N-1}(x_i) = 15: s_i = -25$

# Градиентный бустинг для логистической функции потерь

$$\begin{aligned} s_i^{(N)} &= -\frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} = \\ &= -\frac{\partial}{\partial z} \log(1 + \exp(-y_i z)) \Big|_{z=a_{N-1}(x_i)} = \\ &= \frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \end{aligned}$$

# Градиентный бустинг для логистической функции потерь

$$\begin{aligned} s_i^{(N)} &= -\frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a_{N-1}(x_i)} = \\ &= -\frac{\partial}{\partial z} \log(1 + \exp(-y_i z)) \Big|_{z=a_{N-1}(x_i)} = \\ &= \frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \end{aligned}$$

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - \frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \right)^2 \rightarrow \min_{b_N(x)}$$

# Градиентный бустинг для логистической функции потерь

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - \frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \right)^2 \rightarrow \min_{b_N(x)}$$

- Отступ большой положительный:  $\frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \approx 0$
- Отступ большой отрицательный:  $\frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \approx \pm 1$

# Градиентный бустинг для логистической функции потерь

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left( b_N(x_i) - \frac{y_i}{1 + \exp(y_i a_{N-1}(x_i))} \right)^2 \rightarrow \min_{b_N(x)}$$

- $y_i = +1, a_{N-1}(x_i) = -0.7: s_i = 0.67$
- $y_i = +1, a_{N-1}(x_i) = 2: s_i = 0.12$

# Резюме

- Чтобы учесть особенности функции потерь, можно посчитать её производные в точке текущего прогноза композиции
- Базовую модель будем обучать на эти производные (со знаком минус)