Машинное обучение

Лекция 8

Бэггинг и случайные леса

Ковалев Евгений

ekovalev@hse.ru

НИУ ВШЭ, 2020

Неустойчивость деревьев

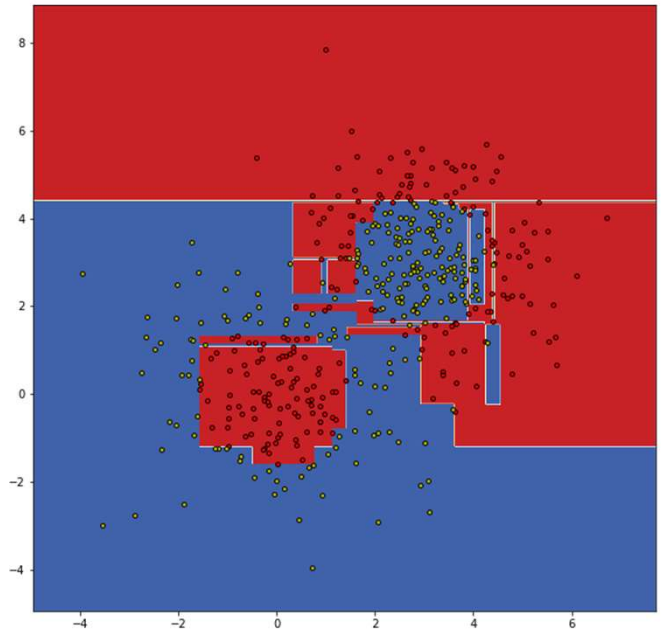
Устойчивость моделей

- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- Обучаем модель $a(x)$
- Ожидаем, что модель устойчивая
- То есть не сильно меняется при небольших изменениях в X
- $\tilde{X}$ — случайная подвыборка, примерно 90% исходной

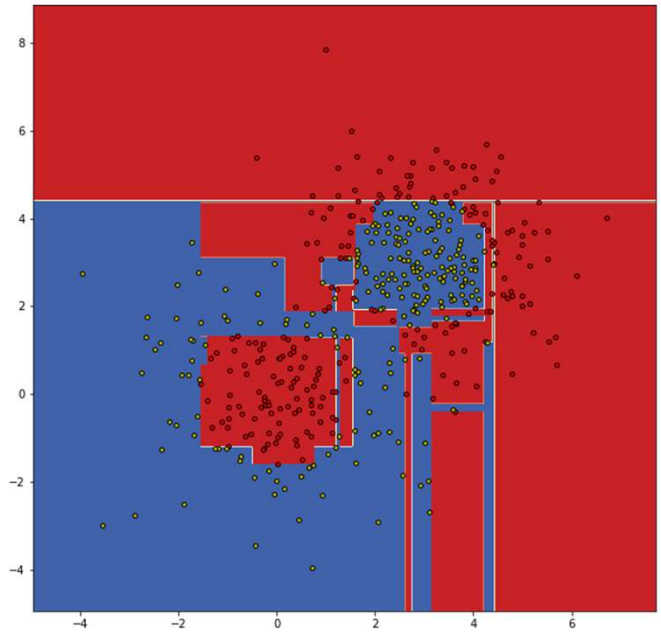
Устойчивость моделей

- $\tilde{X}$ — случайная подвыборка, примерно 90% исходной
- Что будет происходить с деревьями на разных подвыборках?

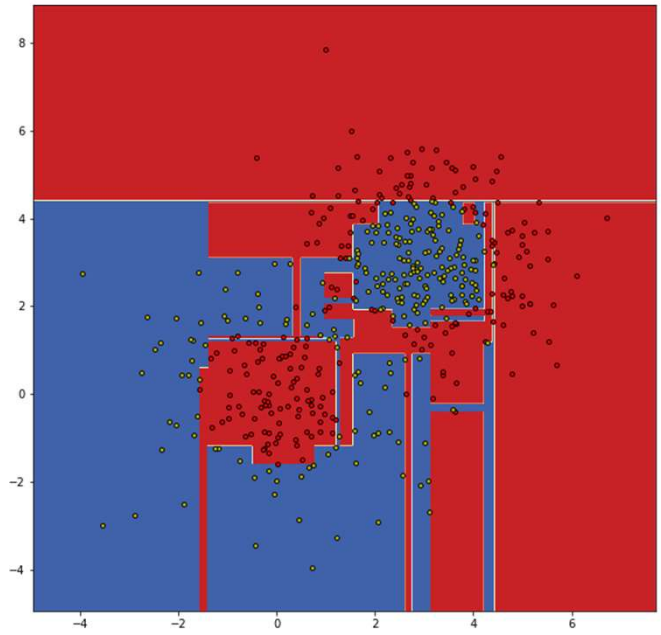
Обучение на подвыборках



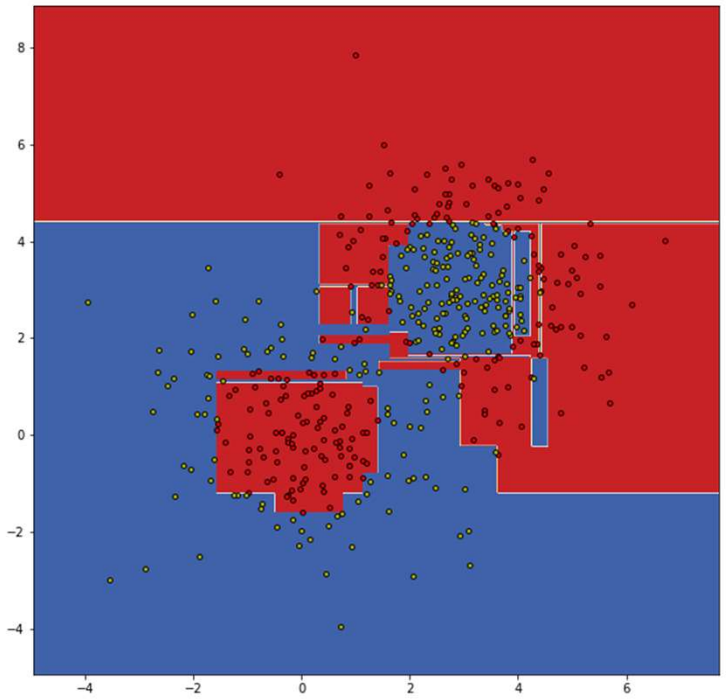
Обучение на подвыборках

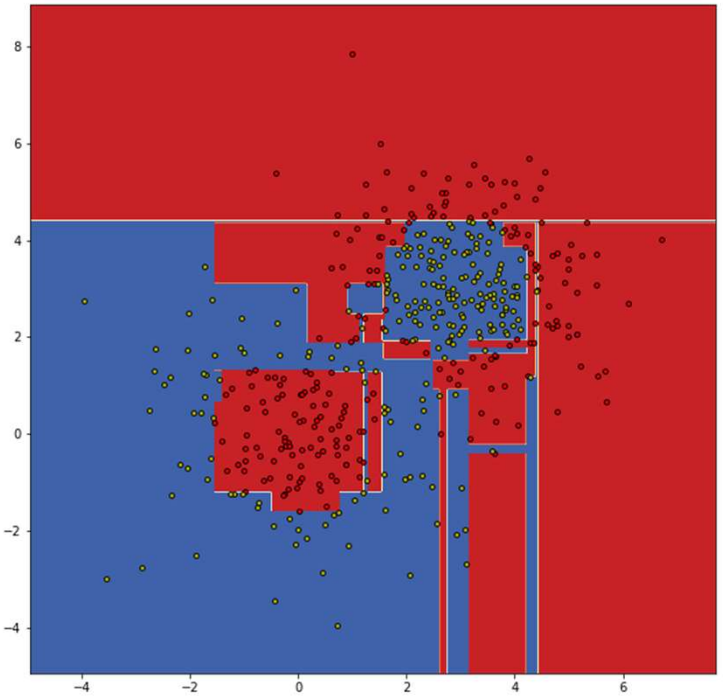


Обучение на подвыборках



Обучение на подвыборках





Композиция моделей

- У нас получилось N деревьев: $b_1(x), \ldots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

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

Композиция моделей

- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

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

Количество деревьев, выдавших класс y

Композиция моделей

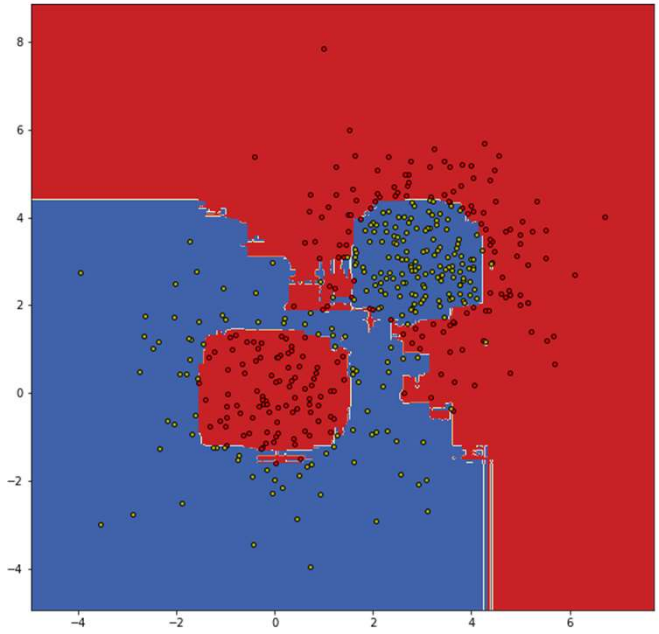
- У нас получилось N деревьев: $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

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

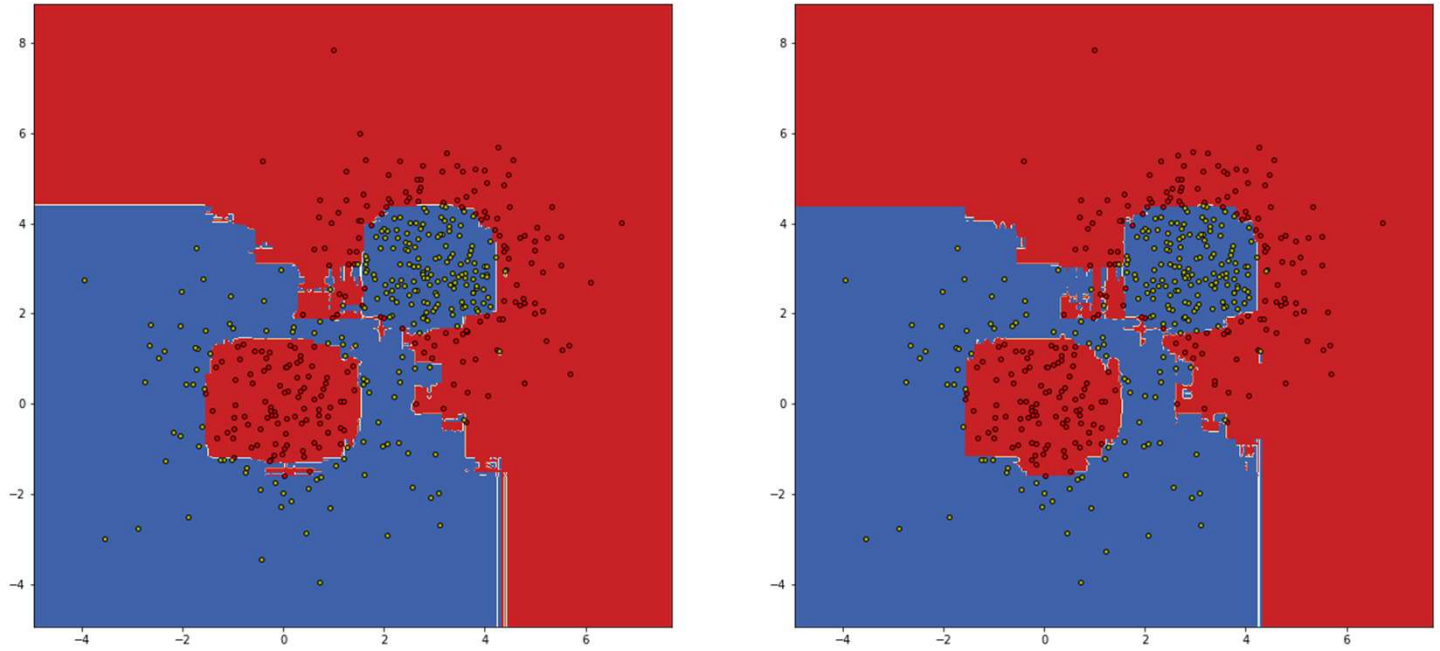
Выбираем класс, который выбрало большинство деревьев

Количество деревьев, выдавших класс y

Композиция моделей



Композиция моделей



Голосование по большинству и усреднение

Majority vote

Majority vote

- Дано: N базовых алгоритмов $b_1(x), \ldots, b_N(x)$
- Композиция: класс, за который проголосовало больше всего базовых алгоритмов

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

Усреднение наблюдений

- Наблюдение: усреднение результатов повышает их точность
- Измерение артериального давления
- Измерение скорости света
- Усреднение соседних пикселей изображения

Композиции моделей

Общий вид: классификация

- $b_1(x), \ldots, b_N(x)$ — базовые модели
- Каждая хотя бы немного лучше случайного угадывания
- Композиция: голосование по большинству (majority vote)

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

Общий вид: регрессия

- $b_1(x), \ldots, b_N(x)$ — базовые модели
- Каждая хотя бы немного лучше случайного угадывания
- Композиция: усреднение

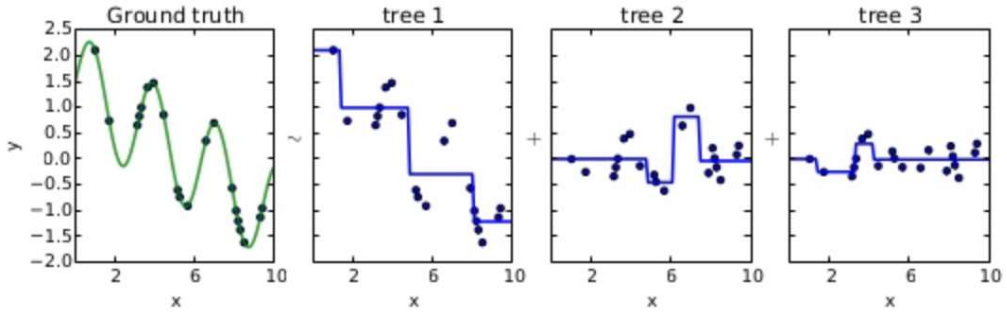
$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

Базовые модели

- $b_1(x), \ldots, b_N(x)$ — базовые модели
- Как на одной выборке построить N различных моделей?
- Вариант 1: обучить их независимо на разных подвыборках
- Вариант 2: обучать последовательно для корректировки ошибок

Бустинг

- Каждая следующая модель исправляет ошибки предыдущих
- Например, градиентный бустинг



Бэггинг

- Bagging (bootstrap aggregating)
- Базовые модели обучаются независимо
- Каждый обучается на подмножестве обучающей выборки
- Подмножество выбирается с помощью бутстрапа

Бутстрап

- Выборка с возвращением
- Берём ℓ элементов из X
- Пример: $\{x_1, x_2, x_3, x_4\} \to \{x_1, x_2, x_2, x_4\}$
- В подвыборке будет ℓ объектов, из них около 63.2% уникальных
- Если объект входит в выборку несколько раз, то мы как бы повышаем его вес

Случайные подпространства

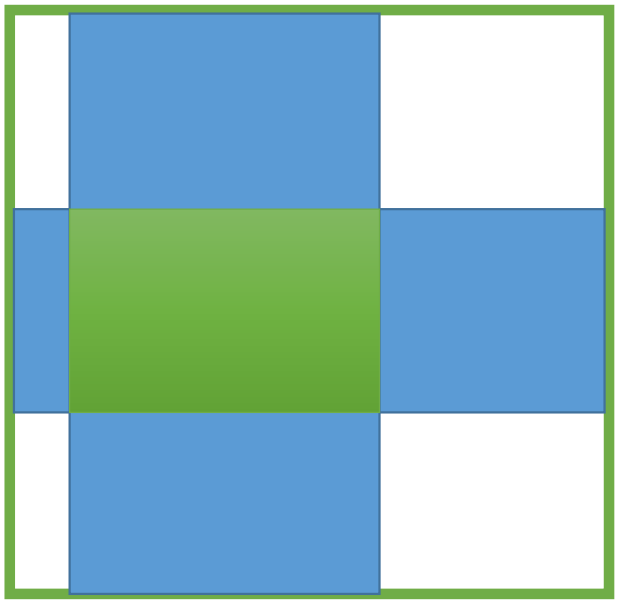
- Выбираем случайное подмножество признаков
- Обучаем модель только на них

Случайные подпространства

- Выбираем случайное подмножество признаков
- Обучаем модель только на них
- Может быть плохо, если имеются важные признаки, без которых невозможно построить разумную модель

Виды рандомизации

- Бэггинг: случайная подвыборка
- Случайные подпространства: случайное подмножество признаков



Резюме

- Будем объединять модели в композиции через усреднение или голосование большинством
- Бэггинг — композиция моделей, обученных независимо на случайных подмножествах объектов
- Можно ещё рандомизировать по признакам
- Как лучше всего?

Смещение и разброс моделей

Разложение ошибки на смещение и разброс

$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y \mid x]\big)^2\Big]}_{\text{шум}} +$$

$$+ \underbrace{\mathbb{E}_x\Big[\big(\mathbb{E}_X\big[\mu(X)\big] - \mathbb{E}[y \mid x]\big)^2\Big]}_{\text{смещение}} + \underbrace{\mathbb{E}_x\Big[\mathbb{E}_X\Big[\big(\mu(X) - \mathbb{E}_X\big[\mu(X)\big]\big)^2\Big]\Big]}_{\text{разброс}}$$

- Разберём на уровне идеи

Разложение ошибки на смещение и разброс

- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных

Разложение ошибки на смещение и разброс

- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных
- Смещение (bias) — способность модели приблизить лучшую среди всех возможных моделей

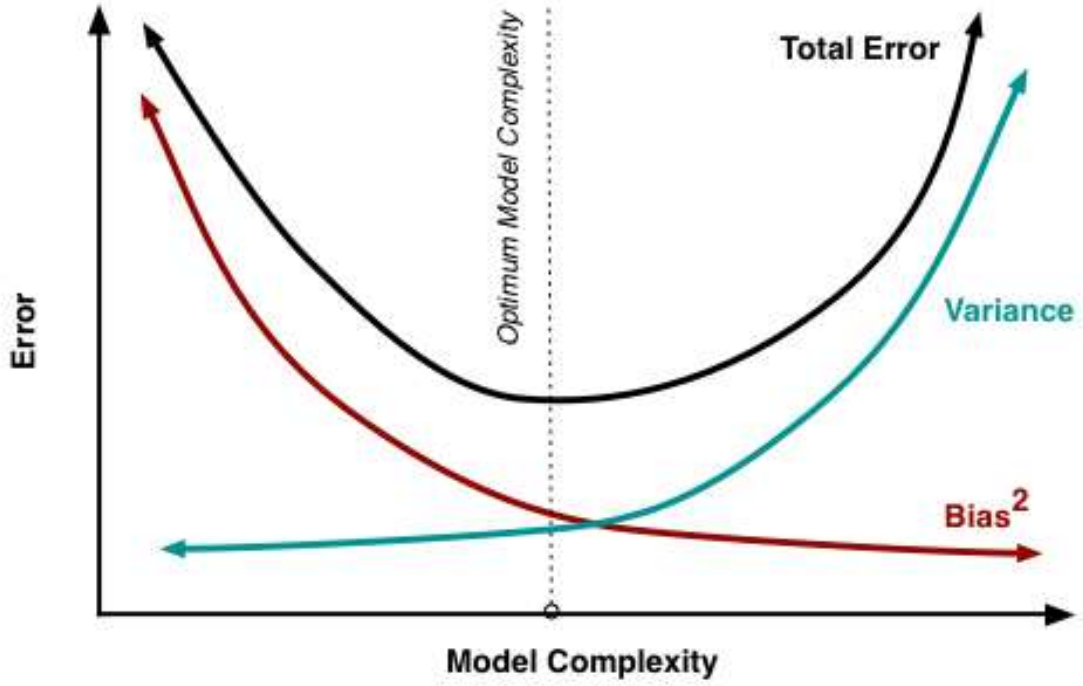
Разложение ошибки на смещение и разброс

- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных
- Смещение (bias) — способность модели приблизить лучшую среди всех возможных моделей
- Разброс (variance) — устойчивость модели к изменениям в обучающей выборке

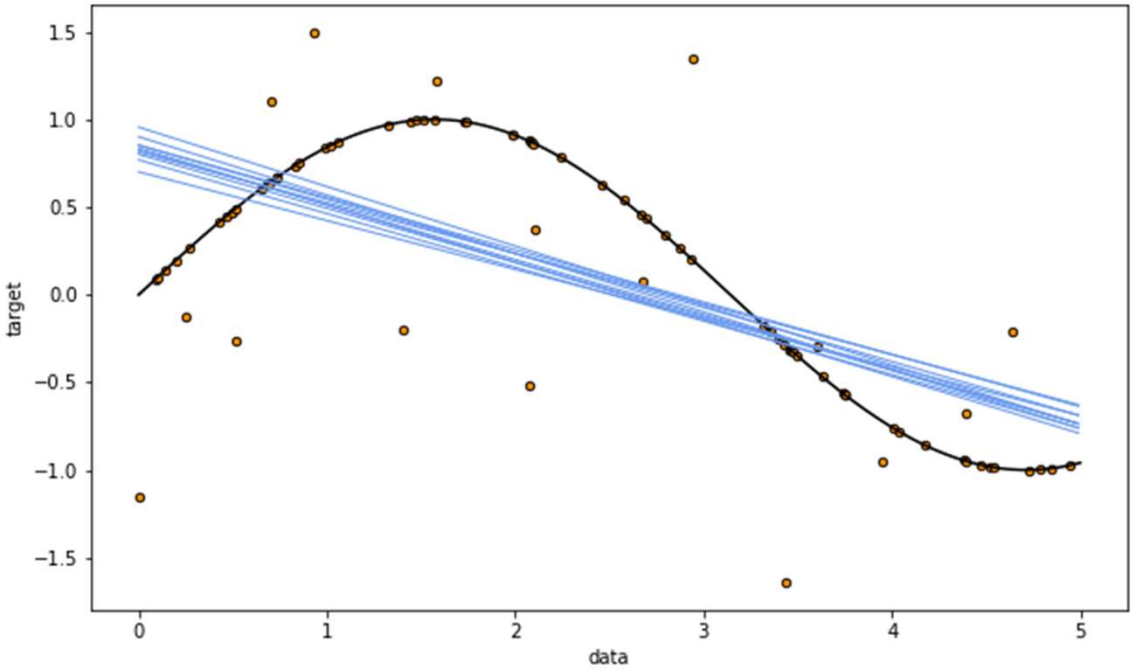
Смещение и разброс

- Высокое смещение может говорить о недообучении (слишком большая ошибка)
- Высокий разброс может говорить о переобучении (слишком сложная модель)

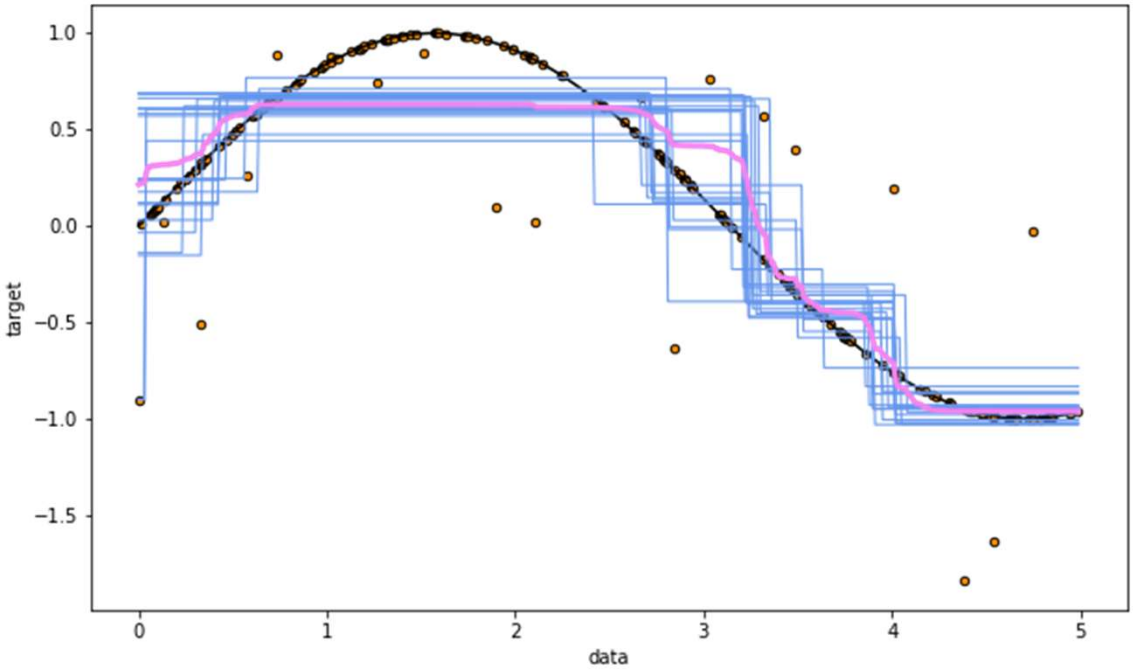
Bias-variance tradeoff



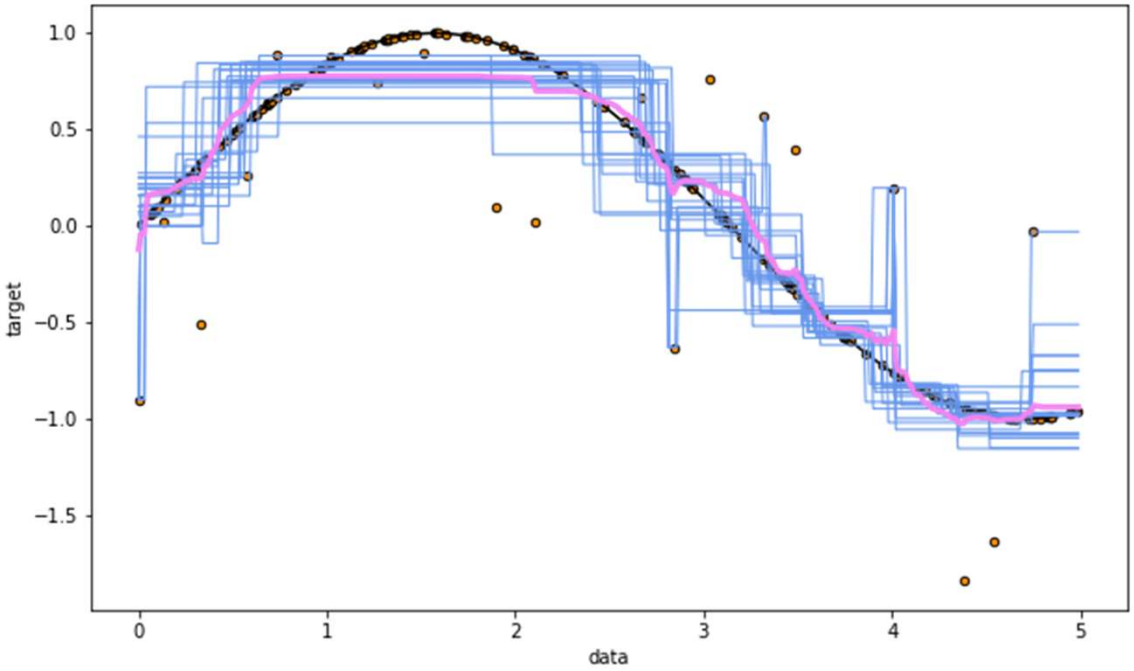
Смещение и разброс: линейная модель



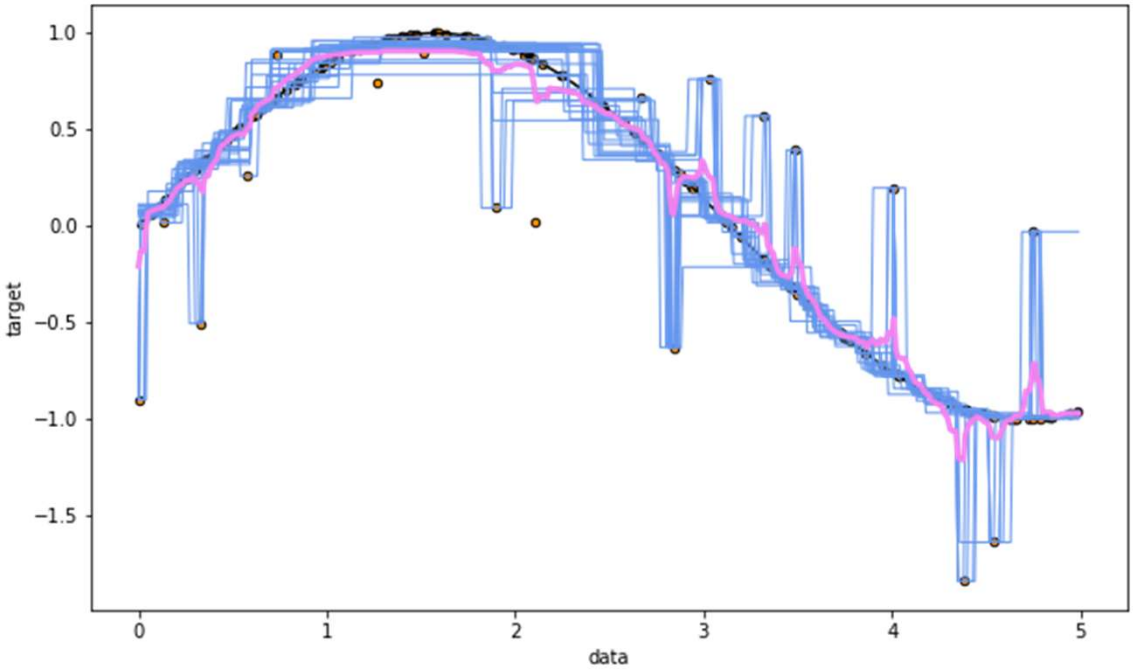
Смещение и разброс: деревья



Смещение и разброс: деревья



Смещение и разброс: деревья



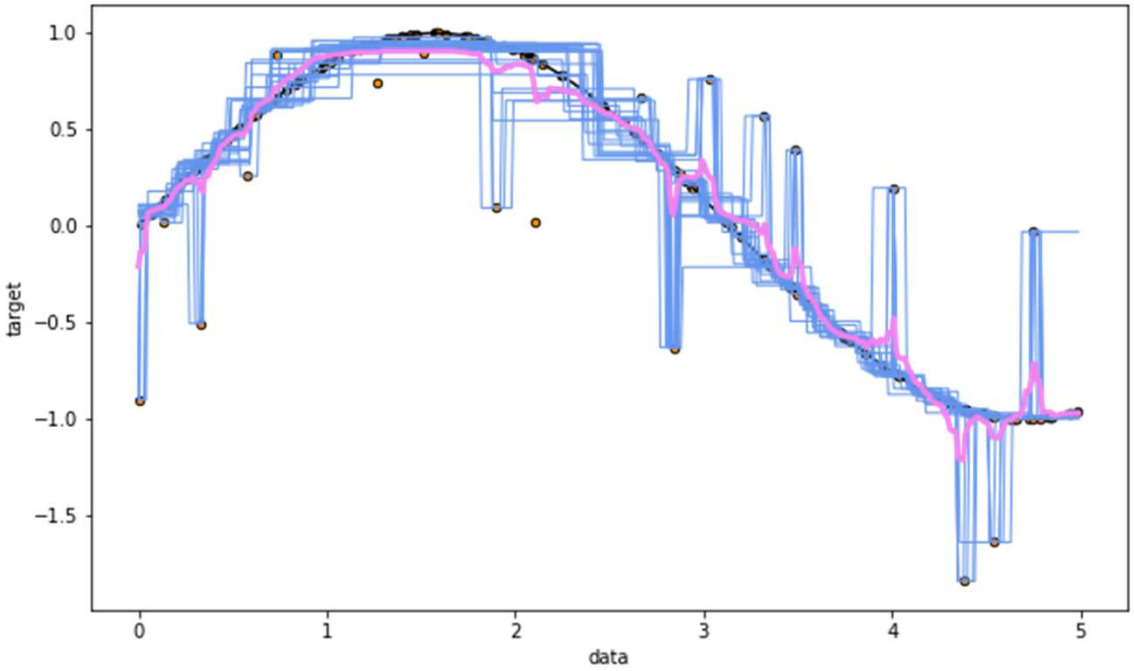
Бэггинг

- Смещение $a_N(x)$ такое же, как у $b_n(x)$
- Разброс $a_N(x)$:

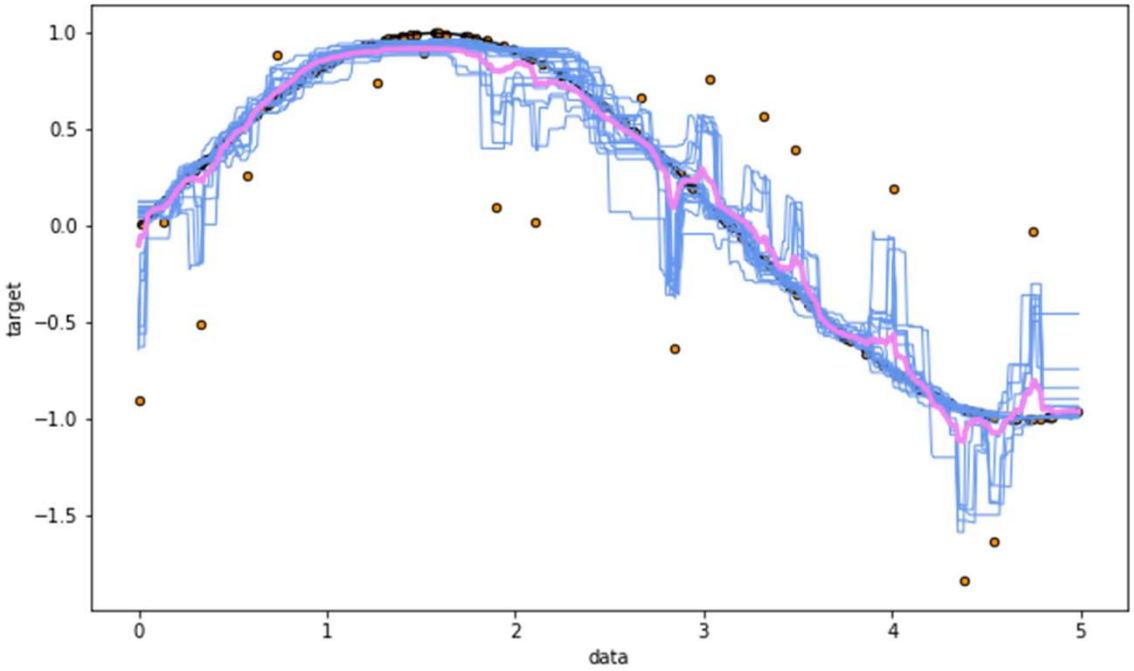
$$\frac{1}{N}\left(\text{разброс } b_n(x)\right) + \text{ковариация}\left(b_n(x), b_m(x)\right)$$

- 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

$\mathrm{SplitNode}(m, R_m)$

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

Жадный алгоритм

$\mathrm{SplitNode}(m, R_m)$

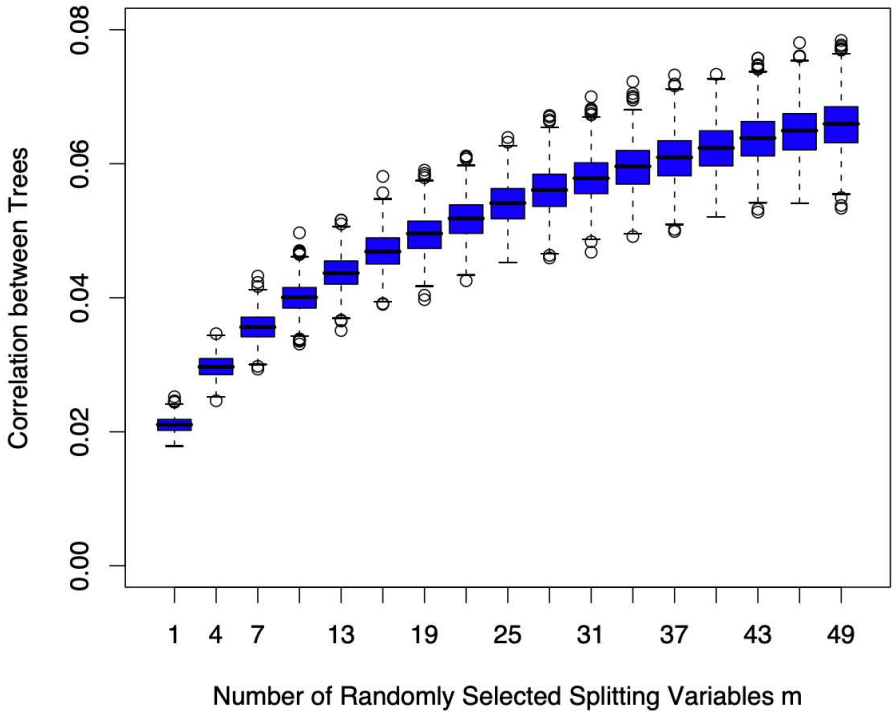
1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg\min_{\mathrm{j,t}} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{\{(x, y) \in R_m | \left[x_j < t\right]\right\}$, $R_r = \left\{\{(x, y) \in R_m | \left[x_j \geq t\right]\right\}$
4. Повторяем для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

Выбор предиката

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

- 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:

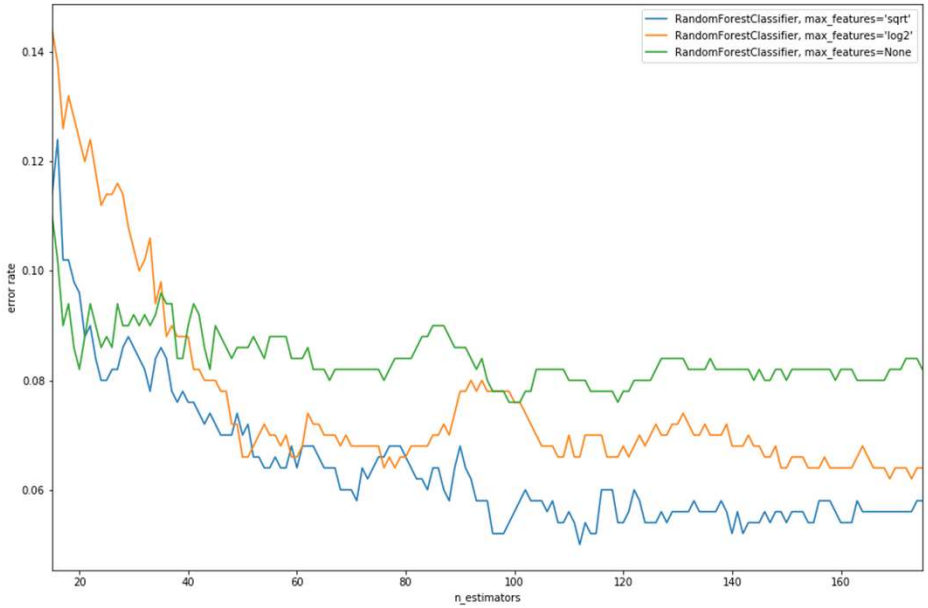
$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

- 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:

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

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:

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

Для каждого объекта выборки

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:

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

Для каждого объекта выборки

Суммируем ответы для всех деревьев, в которые не попал объект

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:

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

Для каждого объекта выборки

Среднее ответов = предсказание на объекте

Суммируем ответы для всех деревьев, в которые не попал объект

Out-of-bag (пример)

- 4 объекта: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)

Out-of-bag (пример)

- 4 объекта: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)
- 3 дерева:
  - Дерево b_1: train on (x_1, y_1), (x_2, y_2)
  - Дерево b_2: train on (x_3, y_3), (x_4, y_4)
  - Дерево b_3: train on (x_1, y_1), (x_2, y_2), (x_3, y_3)

Out-of-bag (пример)

- 4 объекта: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)
- 3 дерева:
  - Дерево b_1: train on (x_1, y_1), (x_2, y_2)
  - Дерево b_2: train on (x_3, y_3), (x_4, y_4)
  - Дерево b_3: train on (x_1, y_1), (x_2, y_2), (x_3, y_3)

$$Q_1 = L\big(y_1, b_2(x_1)\big)$$

Out-of-bag (пример)

- 4 объекта: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)
- 3 дерева:
  - Дерево b_1: train on (x_1, y_1), (x_2, y_2)
  - Дерево b_2: train on (x_3, y_3), (x_4, y_4)
  - Дерево b_3: train on (x_1, y_1), (x_2, y_2), (x_3, y_3)

$$Q_1 = L\big(y_1, b_2(x_1)\big)$$
$$Q_2 = L\big(y_2, b_2(x_2)\big)$$

Out-of-bag (пример)

- 4 объекта: (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)
- 3 дерева:
  - Дерево b_1: train on (x_1, y_1), (x_2, y_2)
  - Дерево b_2: train on (x_3, y_3), (x_4, y_4)
  - Дерево b_3: train on (x_1, y_1), (x_2, y_2), (x_3, y_3)

$$Q_1 = L\big(y_1, b_2(x_1)\big)$$
$$Q_2 = L\big(y_2, b_2(x_2)\big)$$
$$Q_3 = L\big(y_3, b_1(x_3)\big)$$

Out-of-bag (пример)

- 4 объекта: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
- 3 дерева:
  - Дерево b_1: train on $(x_1, y_1), (x_2, y_2)$
  - Дерево b_2: train on $(x_3, y_3), (x_4, y_4)$
  - Дерево b_3: train on $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

$$Q_1 = L\big(y_1, b_2(x_1)\big)$$
$$Q_2 = L\big(y_2, b_2(x_2)\big)$$
$$Q_3 = L\big(y_3, b_1(x_3)\big)$$
$$Q_4 = L\left(y_4, \frac{1}{2}\big(b_1(x_4) + b_3(x_4)\big)\right)$$

Out-of-bag (пример)

$$Q_1 = L\big(y_1, b_2(x_1)\big)$$
$$Q_2 = L\big(y_2, b_2(x_2)\big)$$
$$Q_3 = L\big(y_3, b_1(x_3)\big)$$
$$Q_4 = L\left(y_4, \frac{1}{2}\big(b_1(x_4) + b_3(x_4)\big)\right)$$

$$Q_{test} = \frac{1}{4}(Q_1 + Q_2 + Q_3 + Q_4)$$

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