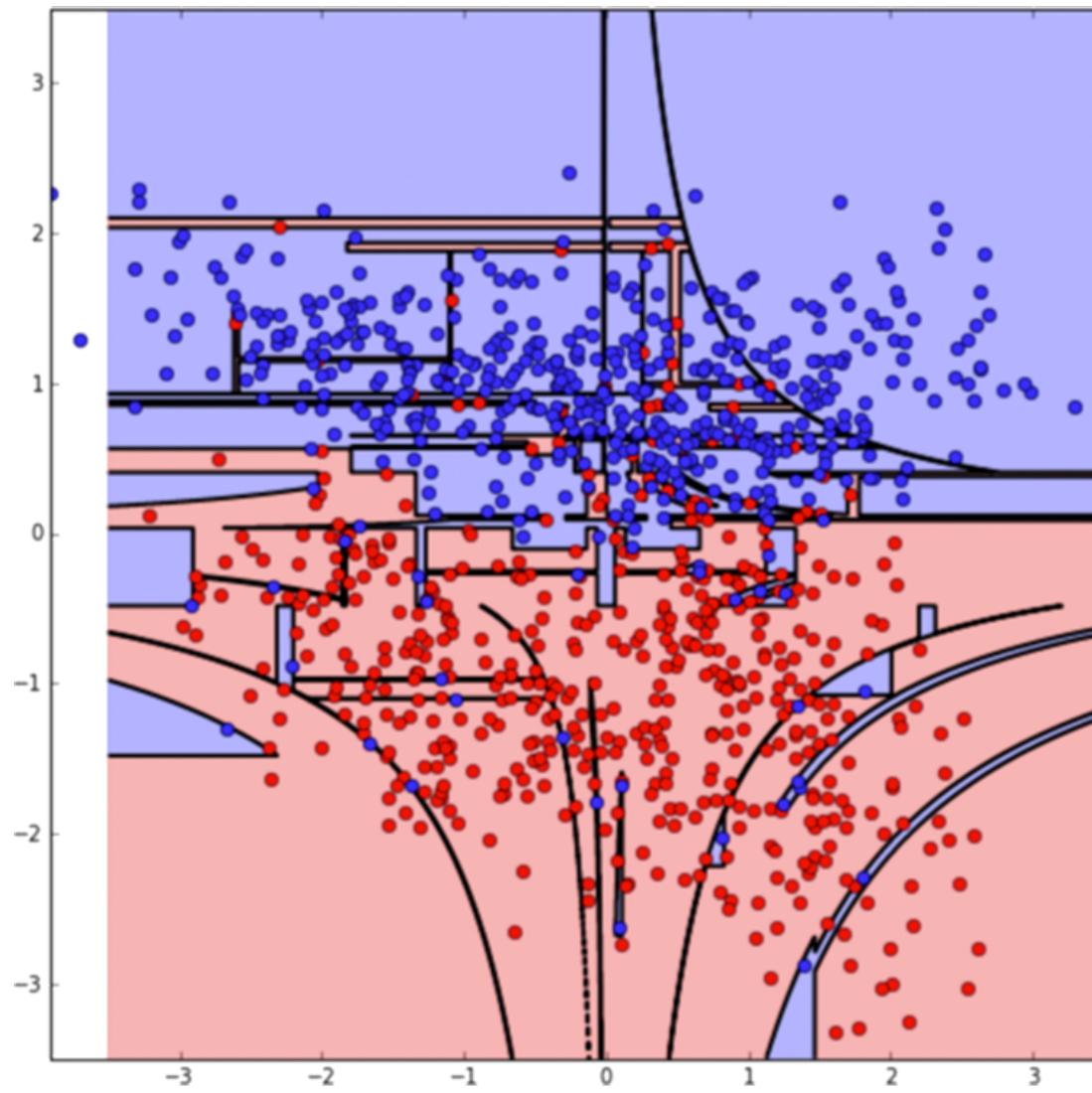


# КОМПОЗИЦИИ ДЕРЕВЬЕВ

---

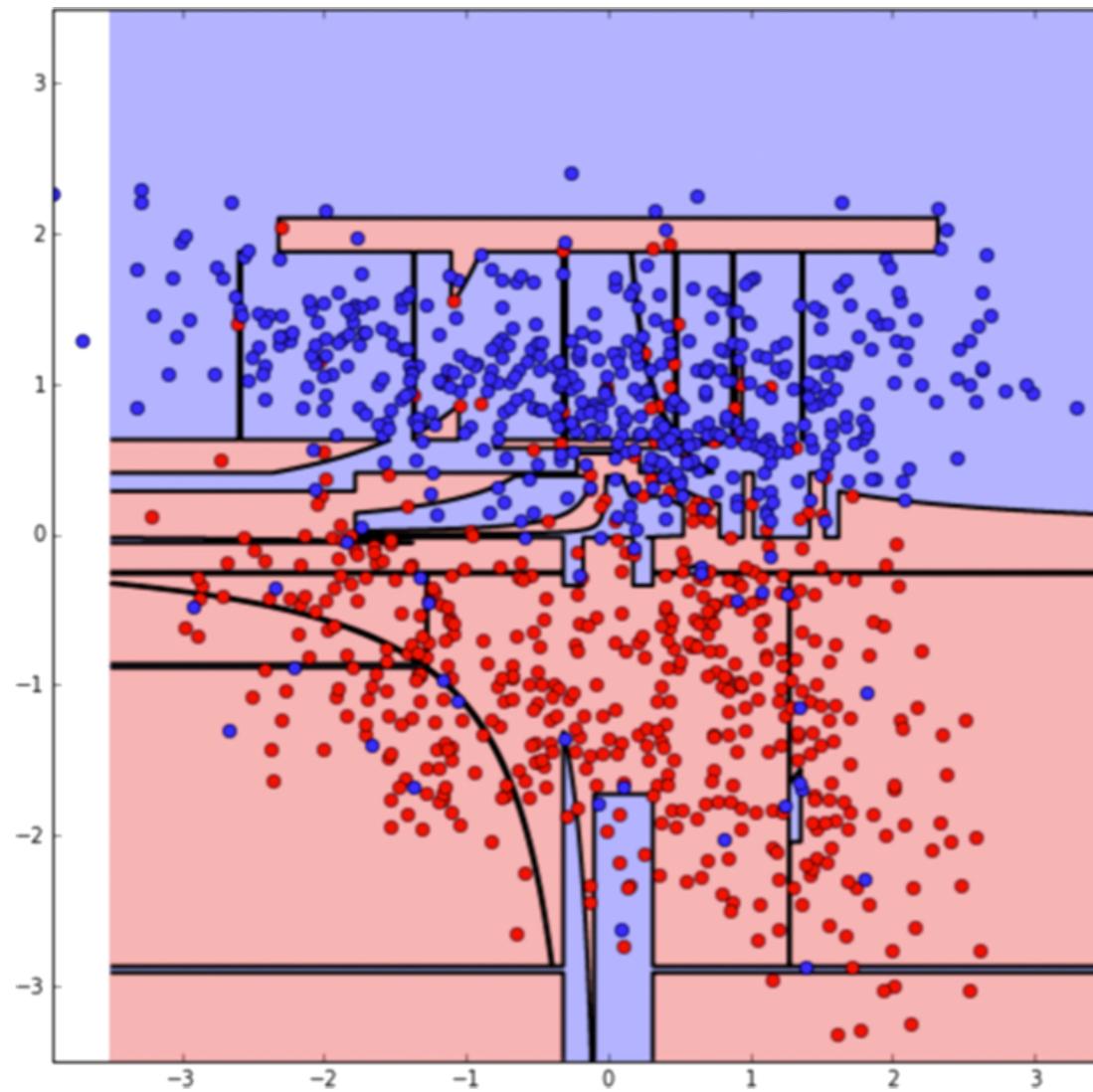
# РЕШАЮЩЕЕ ДЕРЕВО

---



# РЕШАЮЩЕЕ ДЕРЕВО

---



# РЕШАЮЩЕЕ ДЕРЕВО

---

- › Сильно переобучается
- › Сильно меняется при небольшом изменении выборки

# КОМПОЗИЦИЯ ДЕРЕВЬЕВ

---

› Идея:

- ▶ Обучим много деревьев  $b_1(x), \dots, b_N(x)$
- ▶ Усредним ответы:

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

# КОМПОЗИЦИЯ ДЕРЕВЬЕВ

---

› Идея:

- ▶ Обучим много деревьев  $b_1(x), \dots, b_N(x)$
- ▶ Усредним ответы:

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

# ПРИМЕР

---

› Прогнозы деревьев:  $-1, -1, 1, -1, 1, -1$

$$a(x) = \text{sign } -\frac{2}{6} = -1$$

# РАНДОМИЗАЦИЯ

---

- › Как сделать деревья разными?
- › Обучать по подвыборкам!

# РАНДОМИЗАЦИЯ

---

- › Популярный подход: бутстрэп
- › Выбираем из обучающей выборки  $\ell$  объектов с возвращением
- › Пример:  $\{x_1, x_2, x_3, x_4\} \rightarrow \{x_1, x_2, x_3, x_4\}$
- › Примерно  $0.632 * \ell$  различных объектов

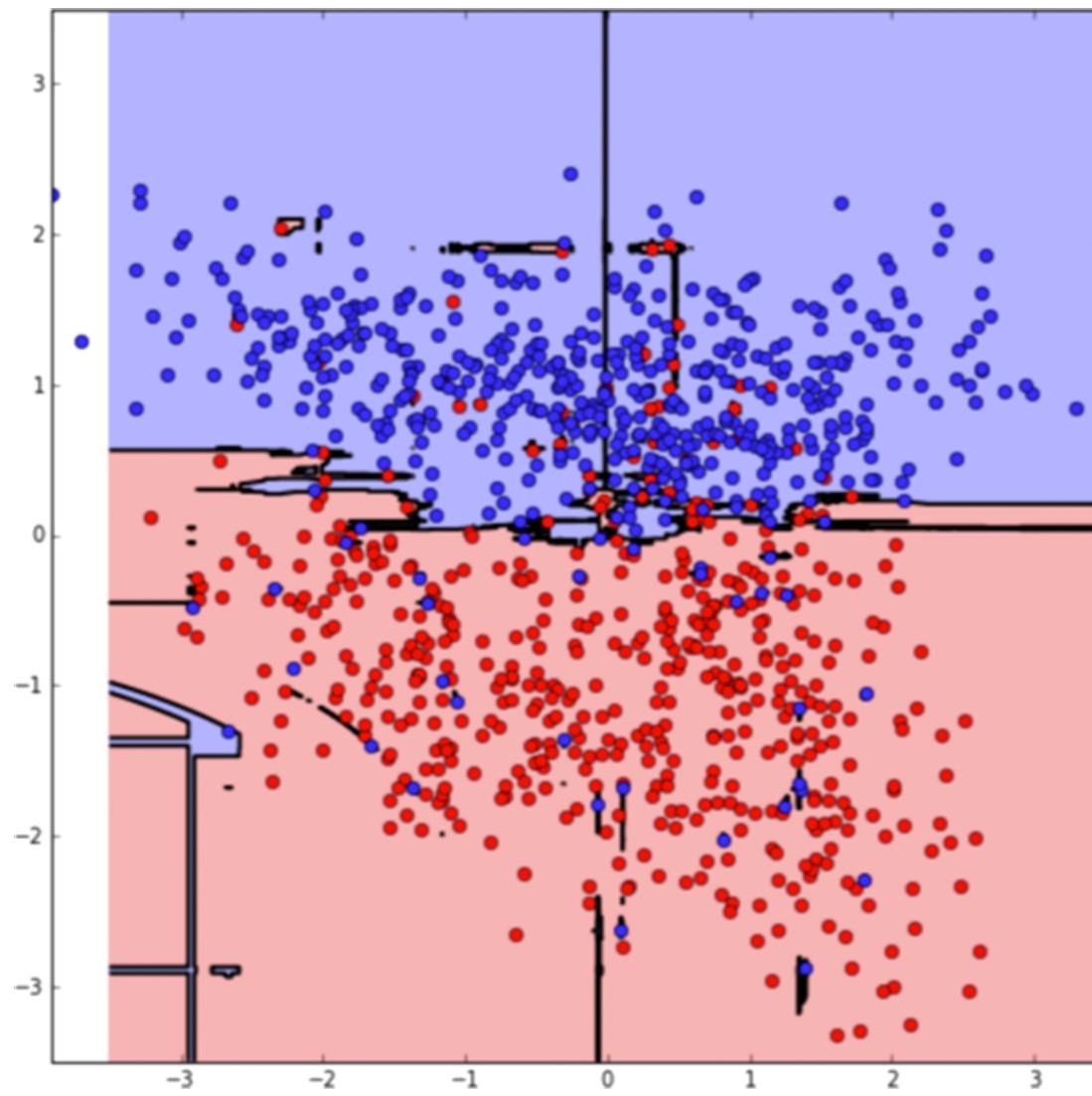
# РАНДОМИЗАЦИЯ

---

- › Другой подход: выбор случайного подмножества объектов
- › Гиперпараметр: размер подмножества

# КОМПОЗИЦИЯ ДЕРЕВЬЕВ

---



# РАНДОМИЗАЦИЯ

---

- › Деревья сильно переобучаются и являются неустойчивыми
- › Усреднение ответов нескольких деревьев повышает качество
- › Рандомизация при обучении: бутстрэп или случайные подвыборки

# СМЕЩЕНИЕ И РАЗБРОС

---

# КОМПОЗИЦИИ ДЕРЕВЬЕВ

---

- › Усреднение ответов дерева повышает качество

# РАЗЛОЖЕНИЕ ОШИБКИ

---

- › Ошибка на новых данных = *Шум* +  
+ Смещение + Разброс
- › *Шум* — ошибка лучшей из всех возможных  
моделей  $y(x)$

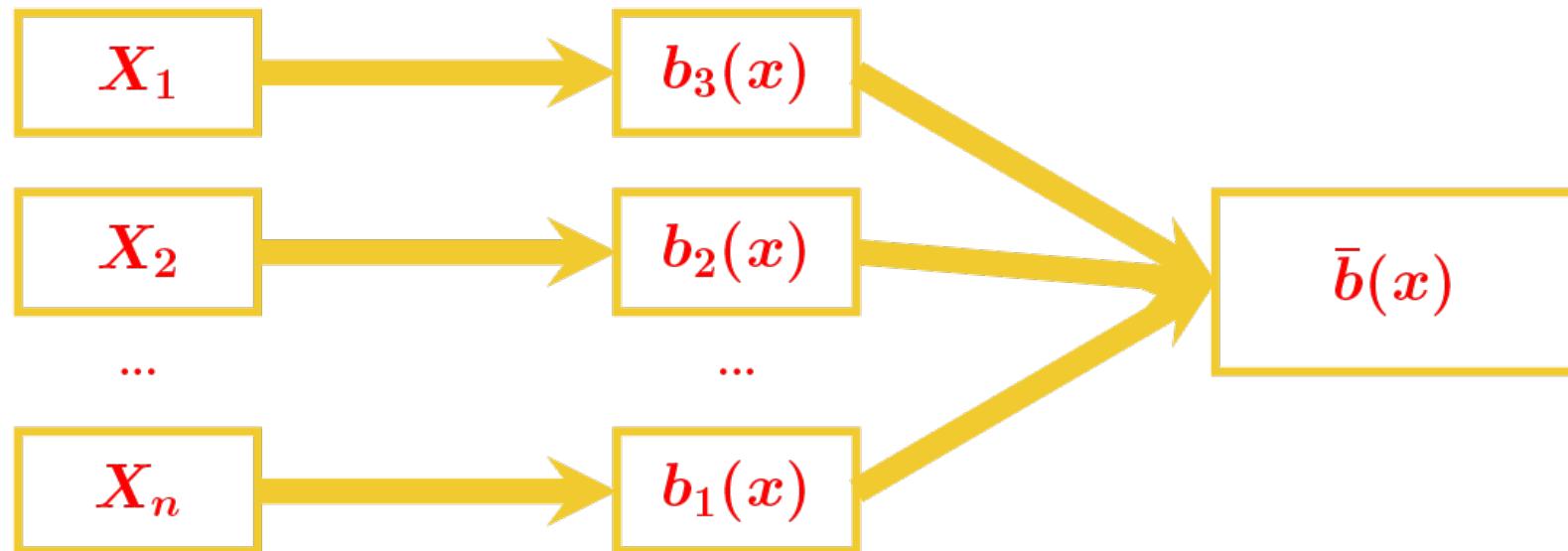
# РАЗЛОЖЕНИЕ ОШИБКИ

---

- › Ошибка на новых данных = Шум +  
+ *Смещение* + Разброс
- › Генерируем много обучающих выборок
- › На каждой обучаем модель
- › *Смещение* — отклонение средних ответов  
нашей модели от ответов лучшей  
модели  $y(x)$

# РАЗЛОЖЕНИЕ ОШИБКИ

---



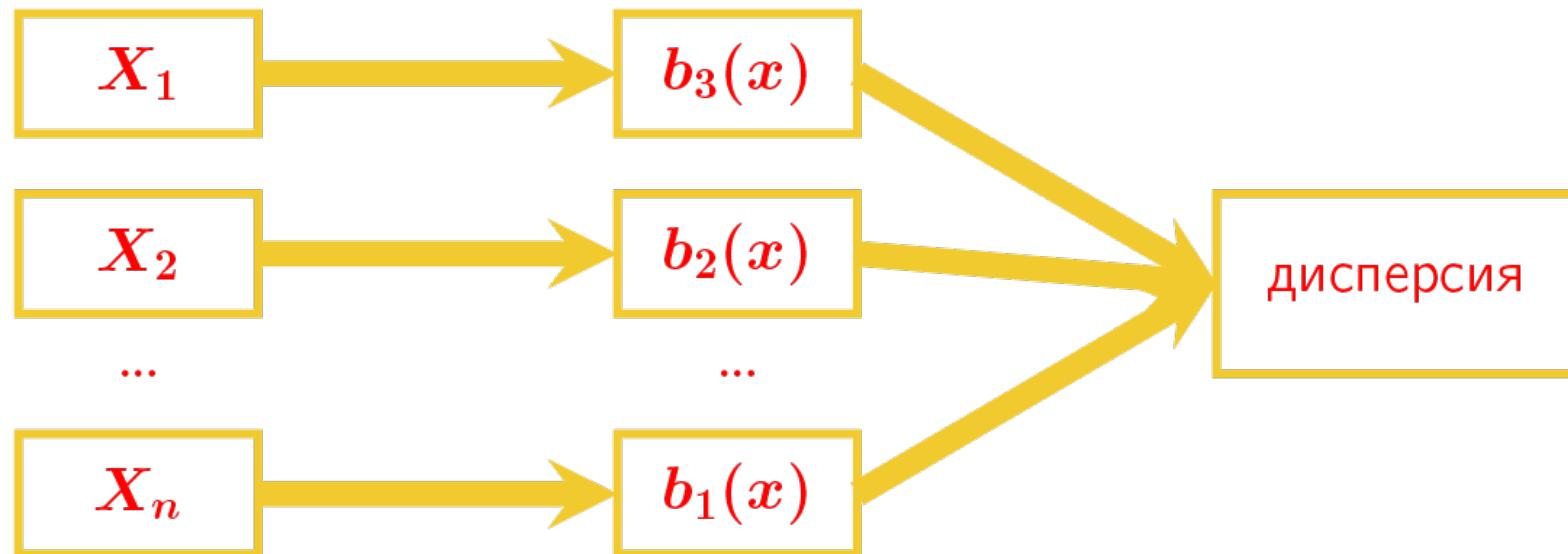
# РАЗЛОЖЕНИЕ ОШИБКИ

---

- › Ошибка на новых данных = Шум +  
+ Смещение + *Разброс*
- › Генерируем много обучающих выборок
- › На каждой обучаем модель
- › *Разброс* — дисперсия ответов наших моделей

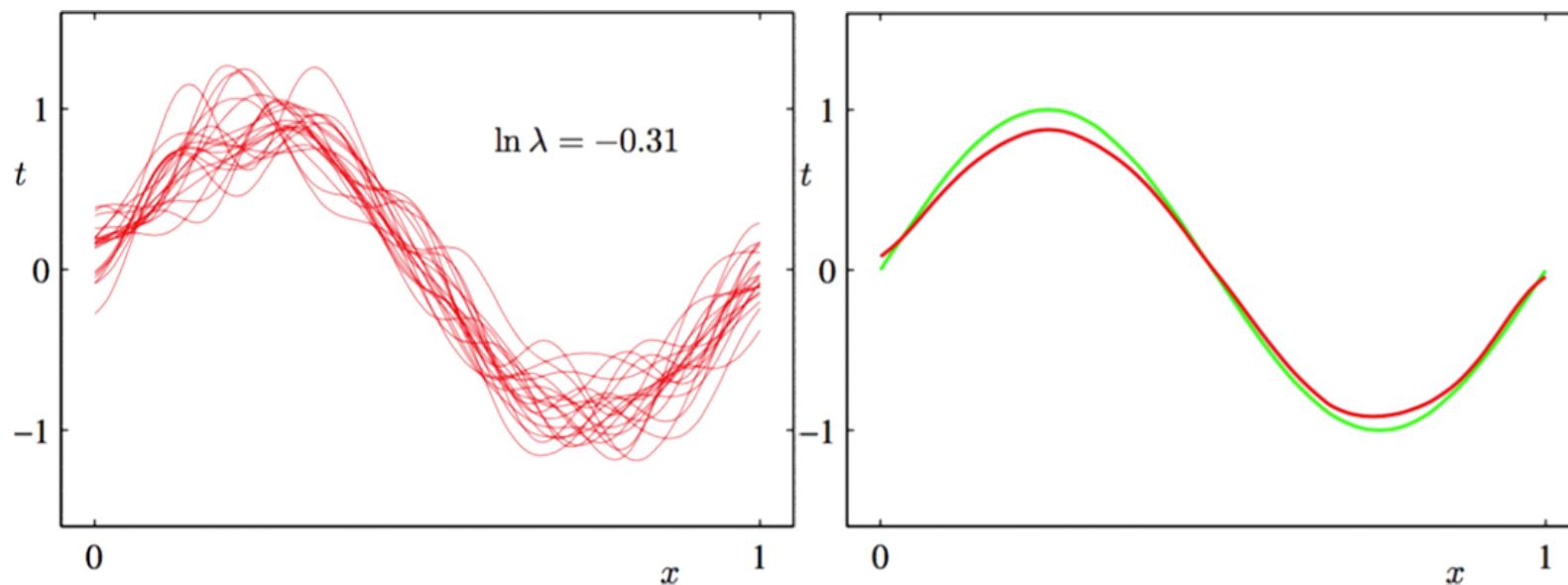
# РАЗЛОЖЕНИЕ ОШИБКИ

---



# ПРИМЕР

---



# РЕШАЮЩИЕ ДЕРЕВЬЯ

---

- › Низкое смещение
- › Большой разброс

# ЛИНЕЙНЫЕ МОДЕЛИ

---

- › Смещение может быть большим
- › Низкий разброс

# УСРЕДНЕНИЕ АЛГОРИТМОВ

- › Не меняет смещение
- › Разброс =  $1/N$  (разброс базового алгоритма) + (корреляция между базовыми алгоритмами)

# УСРЕДНЕНИЕ АЛГОРИТМОВ

---

- › Не меняет смещение
- › Разброс =  $1/N$  (разброс базового алгоритма) + (корреляция между базовыми алгоритмами)
- › Если алгоритмы независимые: разброс уменьшается в  $N$  раз!

# НЕЗАВИСИМОСТЬ АЛГОРИТМОВ

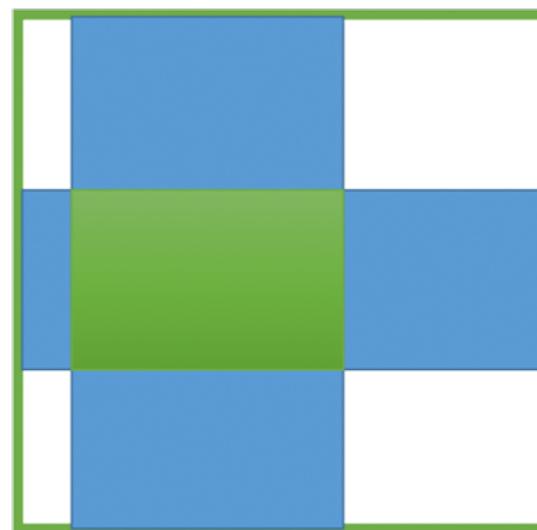
---

- › Базовые алгоритмы обучаются на одних и тех же данных
- › Сложно сделать независимыми

# НЕЗАВИСИМОСТЬ

---

- › Бэггинг: обучаем на случайной подвыборке
- › Метод случайных подпространств: обучаем на случайном подмножестве признаков
- › Размер подвыборки/подмножества — гиперпараметр



# РЕЗЮМЕ

---

- › Ошибка складывается из смещения и разброса
- › Усреднение алгоритмов не меняет смещение и уменьшает разброс
- › Чем меньше корреляция между ответами базовых алгоритмов, тем сильнее уменьшение разброса
- › Бэггинг и метод случайных подпространств

# СЛУЧАЙНЫЕ ЛЕСА

---

# КОМПОЗИЦИЯ АЛГОРИТМОВ

---

- › Ошибка складывается из смещения и разброса
- › Усреднение алгоритмов не меняет смещение и уменьшает разброс
- › Чем меньше корреляция между ответами базовых алгоритмов, тем сильнее уменьшение разброса

# РАНДОМИЗАЦИЯ

---

- › Бэггинг
- › Метод случайных подпространств

# РАНДОМИЗАЦИЯ

---

- › Этого недостаточно
- › Как можно рандомизировать сам процесс построения дерева?

# ПОИСК РАЗБИЕНИЯ

---

- › Пусть в вершине  $m$  оказалась выборка  $X_m$
- ›  $Q(X_m, j, t)$  — критерий ошибки  
условия  $[x^j \leq t]$
- › Ищем лучшие параметры  $j$  и  $t$  перебором:

$$Q(X_m, j, t) \rightarrow \min_{j,t}$$

# ПОИСК РАЗБИЕНИЯ

---

› Пусть в вершине  $m$  оказалась выборка  $X_m$

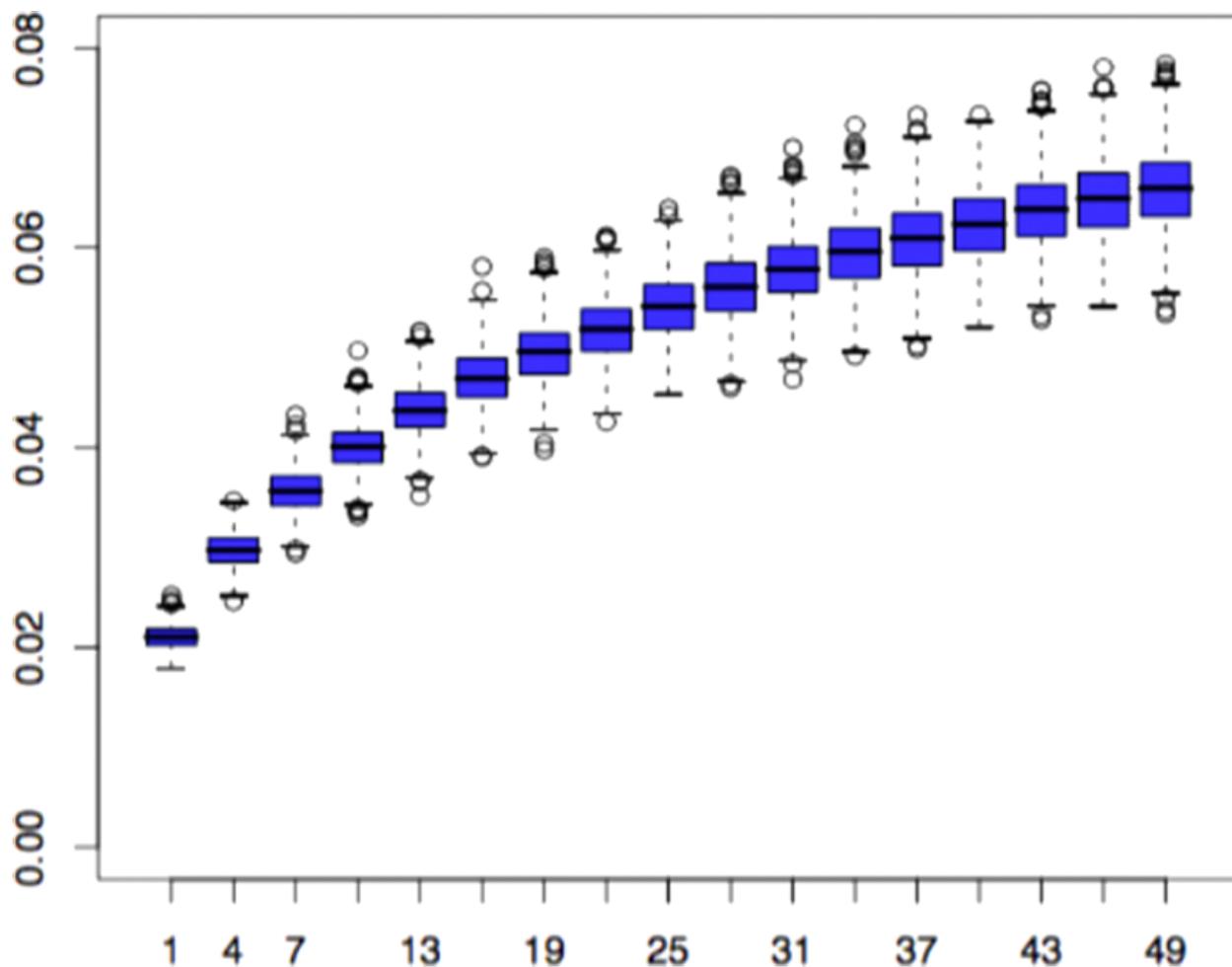
›  $Q(X_m, j, t)$  — критерий ошибки  
условия  $[x^j \leq t]$

› Ищем лучшие параметры  $j$  и  $t$  перебором:

$$Q(X_m, j, t) \rightarrow \min_{j, t}$$

› Случайный лес: выбираем  $j$  из случайного  
подмножества признаков размера  $q$

# КОРРЕЛЯЦИЯ МЕЖДУ ДЕРЕВЬЯМИ



# КОРРЕЛЯЦИЯ МЕЖДУ ДЕРЕВЬЯМИ

---

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

# Случайный лес (RANDOM FOREST)

---

- » Для  $n = 1, \dots, N$ :
  - ▶ Сгенерировать выборку  $\tilde{X}$  с помощью бутстрата
  - ▶ Построить решающее дерево  $b_n(x)$  по выборке  $\tilde{X}$
  - ▶ Дерево строится, пока в каждом листе не окажется не более  $n_{min}$  объектов

# Случайный лес (RANDOM FOREST)

---

Для  $n = 1, \dots, N$ :

- › Оптимальное разбиение ищется среди  $q$  случайных признаков

# СЛУЧАЙНЫЙ ЛЕС (RANDOM FOREST)

---

» Регрессия:

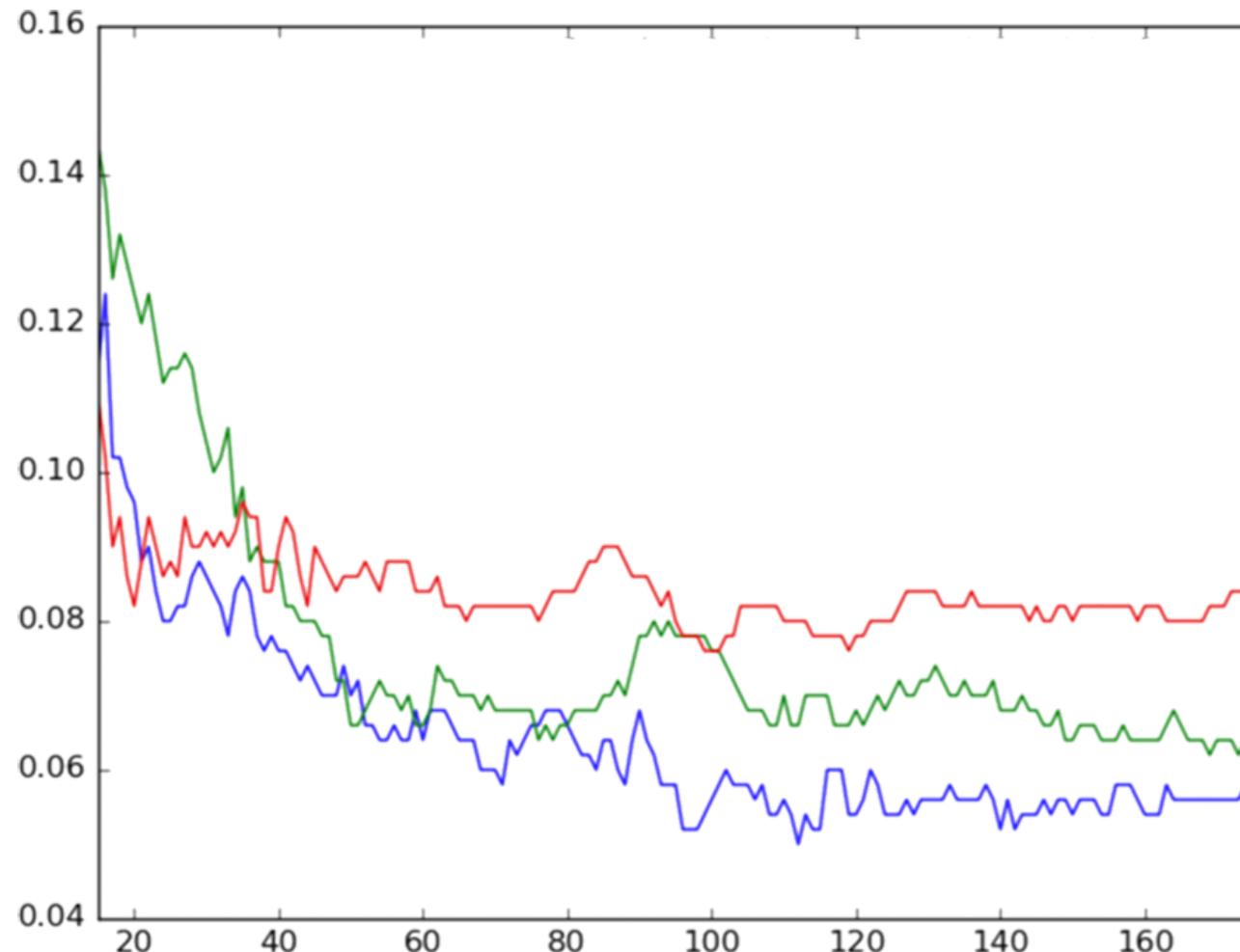
$$a(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

» Классификация:

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

# КАЧЕСТВО НА ТЕСТЕ

---



# РЕЗЮМЕ

---

- › Случайный лес: обучение на случайных подвыборках и рандомизация при поиске разбиения
- › Низкая корреляция между деревьями
- › Не переобучается при росте числа деревьев

# ТРЮКИ СО СЛУЧАЙНЫМИ ЛЕСАМИ

---

# СЛУЧАЙНЫЙ ЛЕС (RANDOM FOREST)

---

- › Для  $n = 1, \dots, N$ :
- › Сгенерировать выборку  $\tilde{X}$  с помощью бутстрапа
- › Построить решающее дерево  $b_n(x)$  по выборке  $\tilde{X}$
- › Дерево строится, пока в каждом листе не окажется не более  $n_{min}$  объектов
- › Оптимальное разбиение ищется среди  $q$  случайных признаков

# ПАРАЛЛЕЛЬНОЕ ПОСТРОЕНИЕ

---

- › Каждое дерево строится независимо от остальных
- › Можно строить отдельные деревья на разных ядрах/компьютерах
- › Идеальное распараллеливание

# ОЦЕНИВАНИЕ ОШИБКИ

---

- › Каждое дерево обучается на подвыборке
- › Подвыборка генерируется бутстррапом
- › Использует в среднем 63.2% объектов

# OUT-OF-BAG

---

- › Лес из  $N$  деревьев
- › Подвыборки  $X_1, \dots, X_N$
- › Прогноз для  $x_i$  найдем по деревьям, которые не обучались на  $x_i$
- › Не нужна дополнительная выборка!

# OUT-OF-BAG

---

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

# OUT-OF-BAG

---

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

Доля деревьев, которые  
не обучались на  $x_i$

# OUT-OF-BAG

---

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

$b_n(x_i)$ , если  $n$ -е дерево  
не обучалось на  $b_n(x_i)$

# OUT-OF-BAG

---

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

Средний ответ деревьев, которые  
не обучались на  $x_i$

# ВАЖНОСТЬ ПРИЗНАКОВ

---

- › По out-of-bag можно оценить важность признаков
- › В следующем курсе!

# РЕЗЮМЕ

---

- › Обучение случайного леса хорошо параллелится
- › Out-of-bag — оценивание обобщающей способности без кросс-валидации