

# Основы машинного обучения

Лекция 13

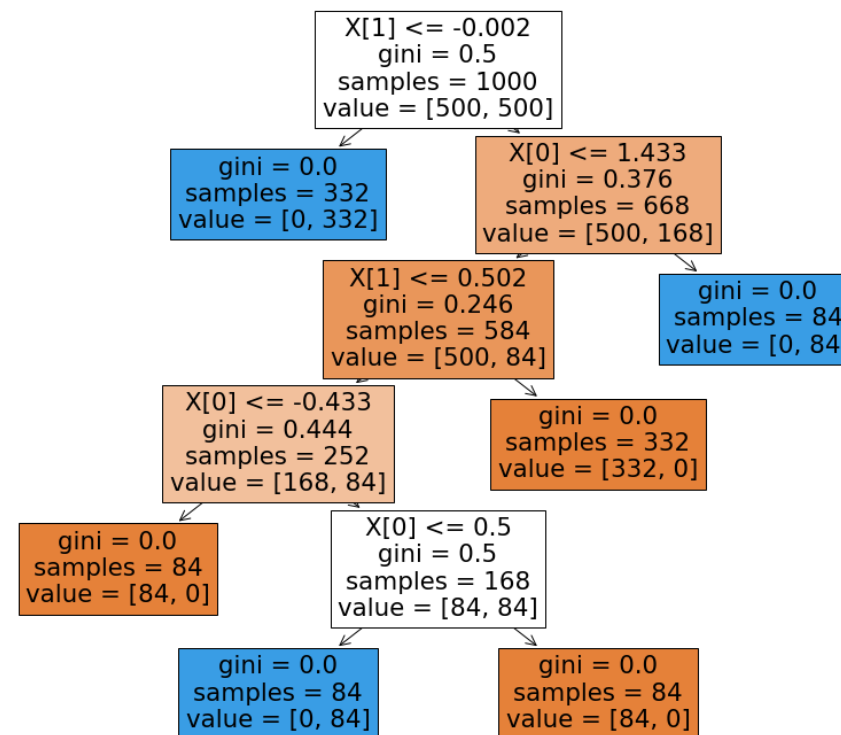
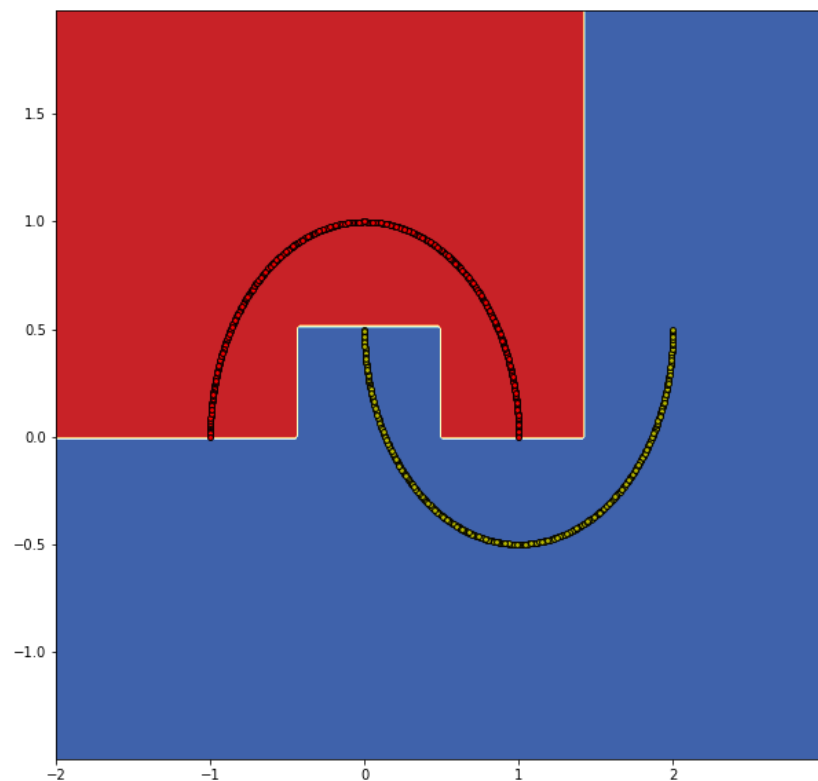
Решающие деревья. Композиции моделей.

Евгений Соколов

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

НИУ ВШЭ, 2024

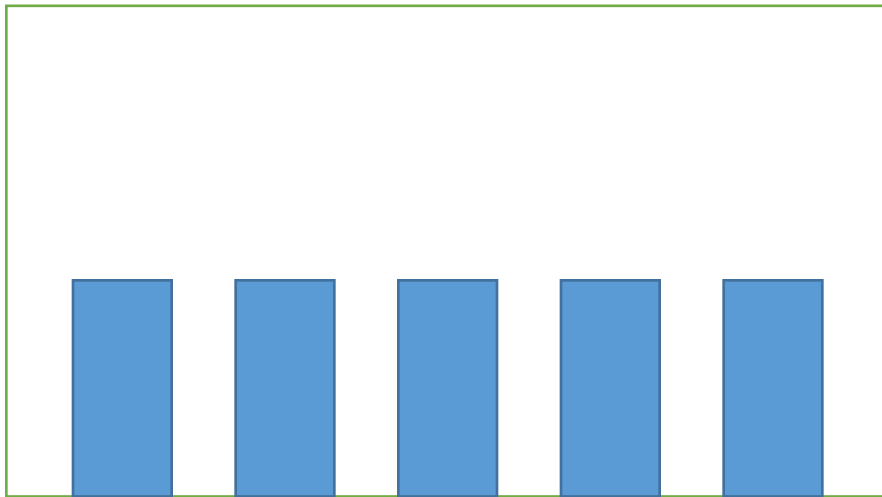
# Решающее дерево



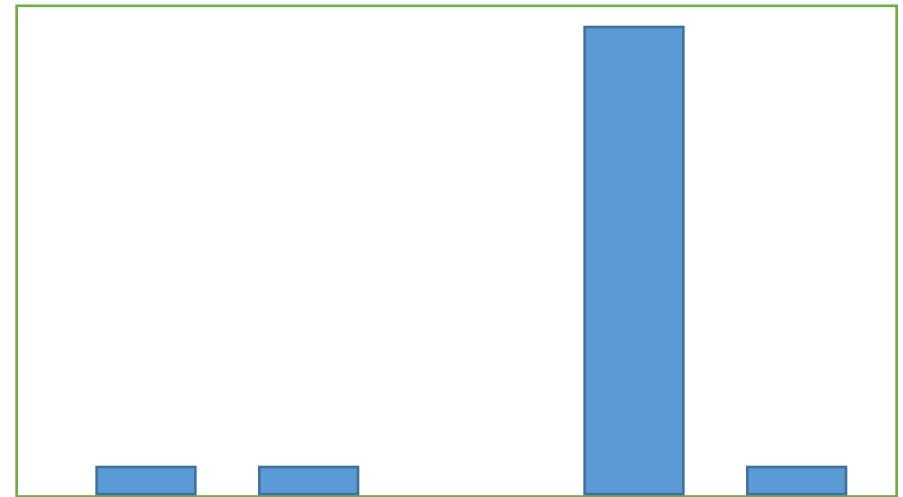
Как выбирать предикаты

# Энтропия

- Мера неопределённости распределения



Высокая энтропия



Низкая энтропия

# Энтропия

- Дискретное распределение
- Принимает  $n$  значений с вероятностями  $p_1, \dots, p_n$
- Энтропия:

$$H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i$$

# Энтропия

$$H(p_1, \dots, p_K) = - \sum_{i=1}^K p_i \log_2 p_i$$

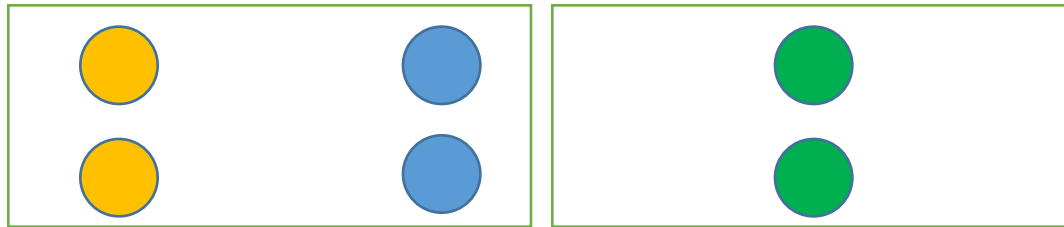
- Характеристика «хаотичности» вершины
- **Impurity**

# Критерий Джини

$$H(p_1, \dots, p_K) = \sum_{i=1}^K p_i (1 - p_i)$$

- Вероятность ошибки случайного классификатора, который выдаёт класс  $k$  с вероятностью  $p_k$
- Примерно пропорционально количеству пар объектов, относящихся к разным классам

# Как сравнить разбиения?



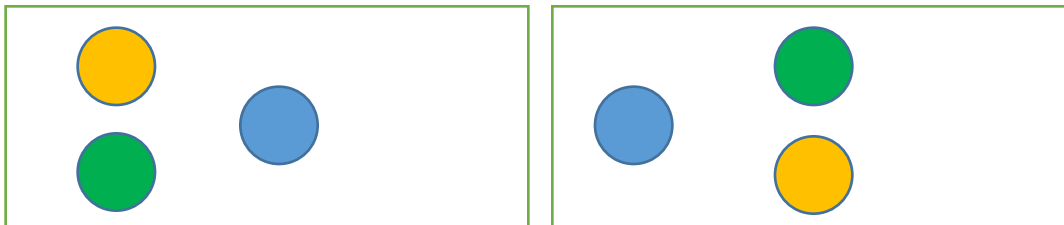
0.693

0

- $(0.5, 0.5, 0)$  и  $(0, 0, 1)$
- $H = 0.693 + 0 = 0.693$

1.09

1.09



- $(0.33, 0.33, 0.33)$  и  $(0.33, 0.33, 0.33)$
- $H = 1.09 + 1.09 = 2.18$



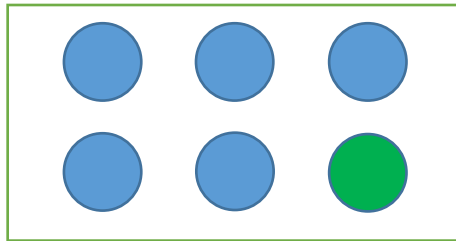
# Критерий информативности

$$Q(R, j, t) = H(R) - \frac{|R_\ell|}{|R|} H(R_\ell) - \frac{|R_r|}{|R|} H(R_r) \rightarrow \max_{j,t}$$

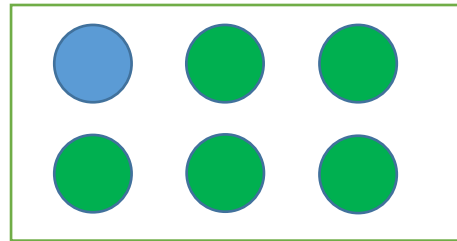
- Или так:

$$Q(R, j, t) = \frac{|R_\ell|}{|R|} H(R_\ell) + \frac{|R_r|}{|R|} H(R_r) \rightarrow \min_{j,t}$$

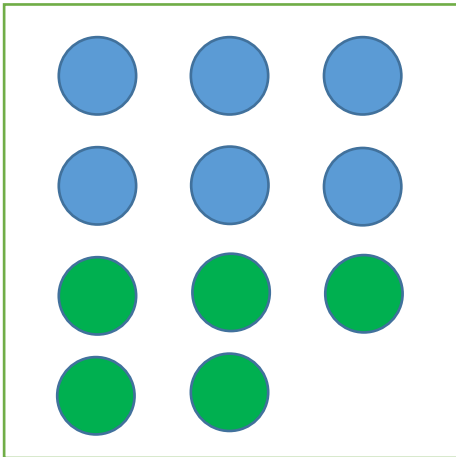
# Как сравнить разбиения?



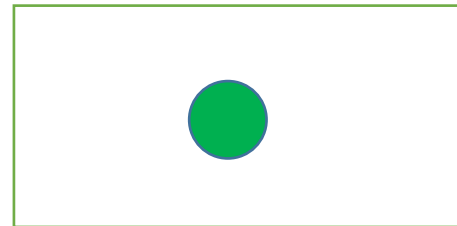
0.65



0.65



0.994



0

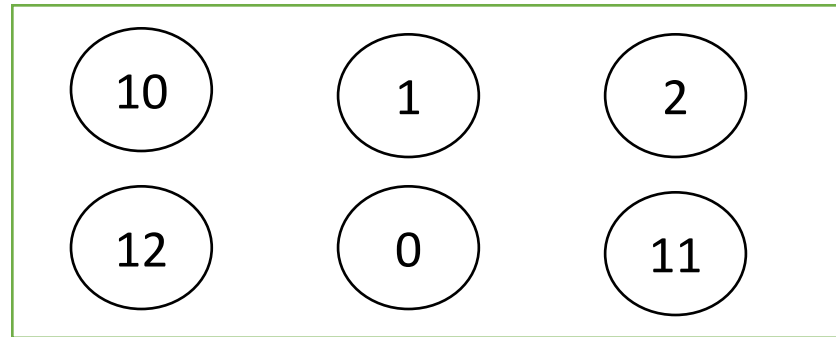
- $(5/6, 1/6)$  и  $(1/6, 5/6)$

- $0.5 * 0.65 + 0.5 * 0.65 = 0.65$

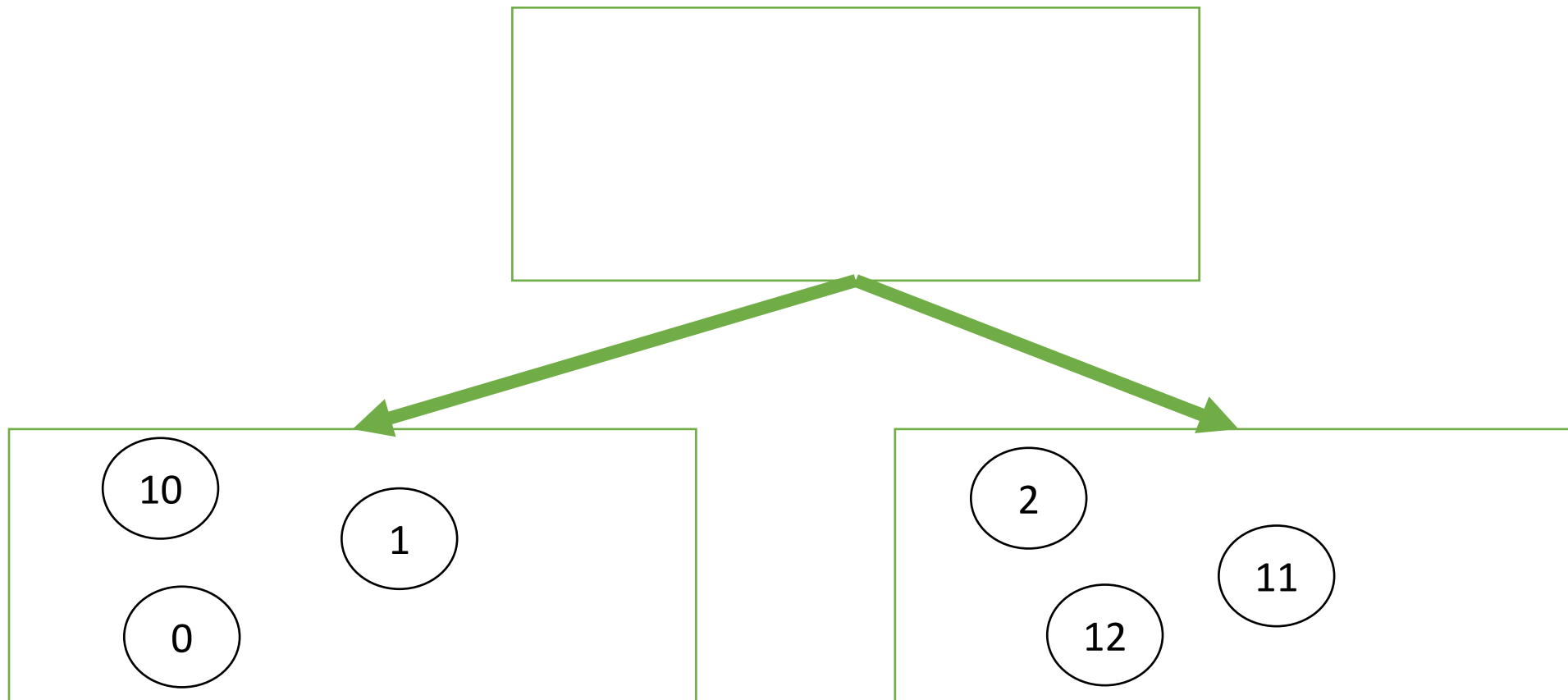
- $(6/11, 5/11)$  и  $(0, 1)$

- $\frac{11}{12} * 0.994 + \frac{1}{12} * 0 = 0.911$

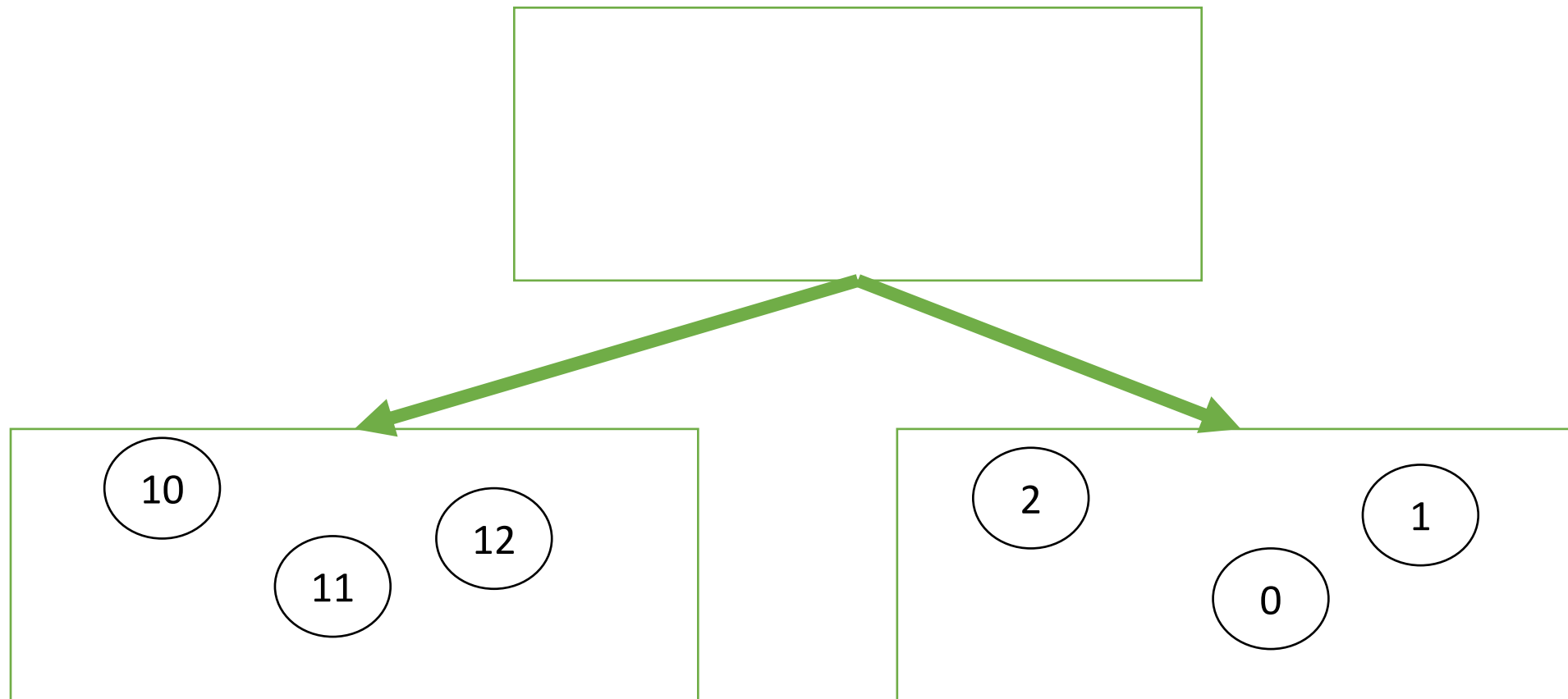
А для регрессии?



А для регрессии?



А для регрессии?



# Задача регрессии

$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - y_R)^2$$

$$y_R = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} y_i$$

- То есть «хаотичность» вершины можно измерять дисперсией ответов в ней

Жадное построение дерева

# Как строить дерево?

- Оптимальный вариант: перебрать все возможные деревья, выбрать самое маленькое среди безошибочных
- Слишком долго



# Как строить дерево?

- Мы уже умеем выбрать лучший предикат для разбиения вершины
- Будем строить жадно
- Начнём с корня дерева, будем разбивать последовательно, пока не выполнится некоторый критерий останова

# Критерий останова

- Ограничить глубину
- Ограничить количество листьев
- Задать минимальное число объектов в вершине
- Задать минимальное уменьшение хаотичности при разбиении
- И так далее

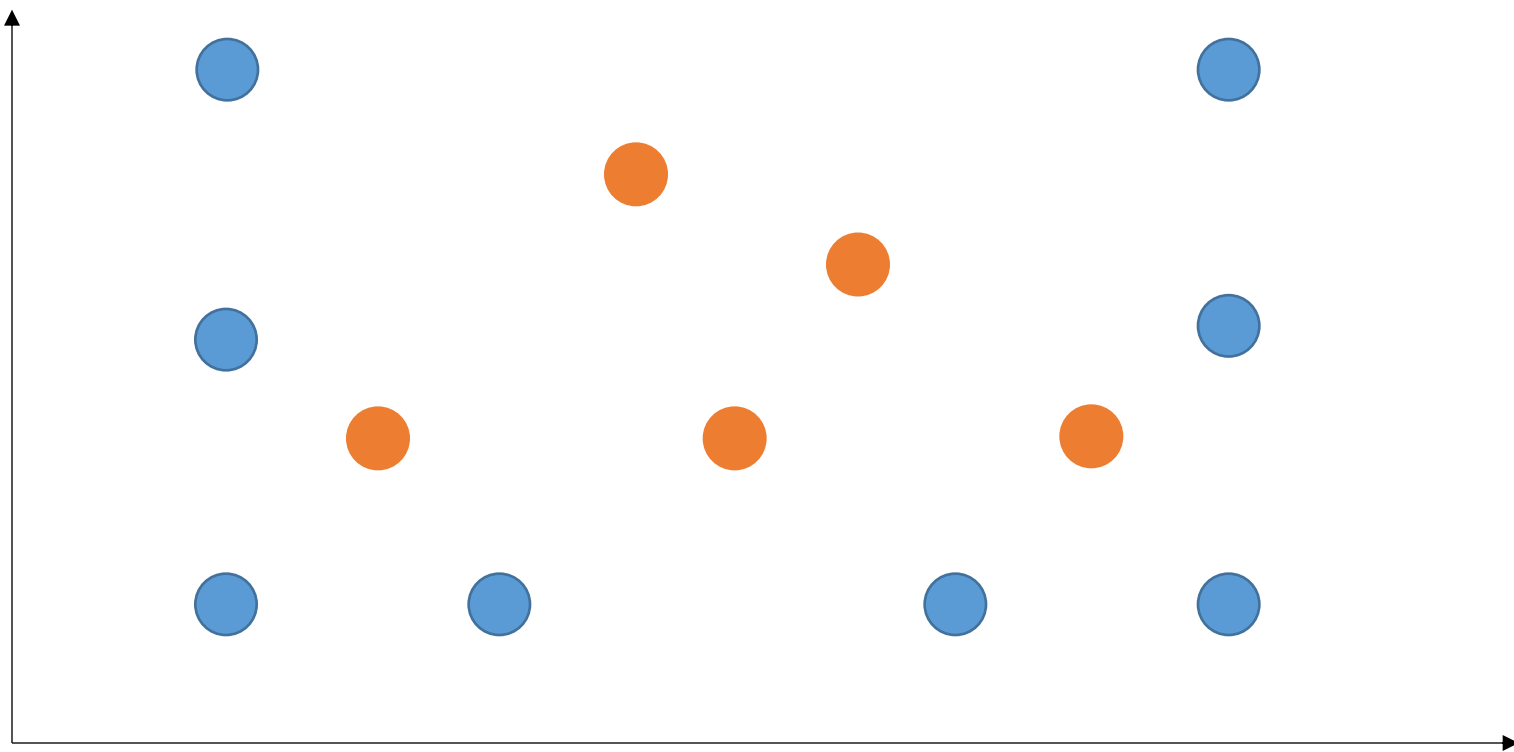
# Жадный алгоритм

1. Поместить в корень всю выборку:  $R_1 = X$
2. Запустить построение из корня:  $\text{SplitNode}(1, R_1)$

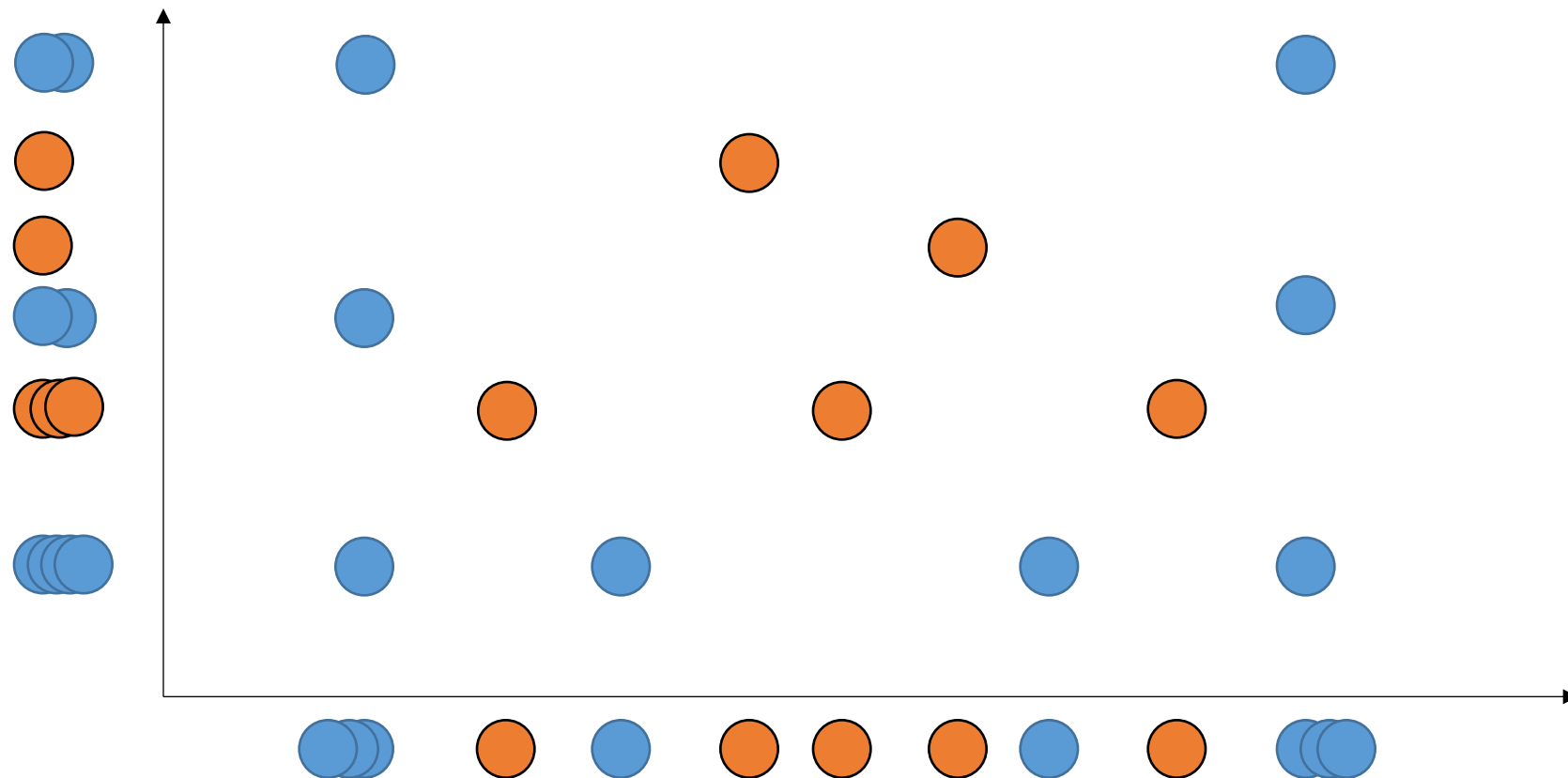
# Жадный алгоритм

- $\text{SplitNode}(m, R_m)$ 
  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:  $R_\ell = \left\{ \{(x, y) \in R_m \mid [x_j < t] \} \right\},$   
 $R_r = \left\{ \{(x, y) \in R_m \mid [x_j \geq t] \} \right\}$
  4. Повторяем для дочерних вершин:  $\text{SplitNode}(\ell, R_\ell)$  и  $\text{SplitNode}(r, R_r)$

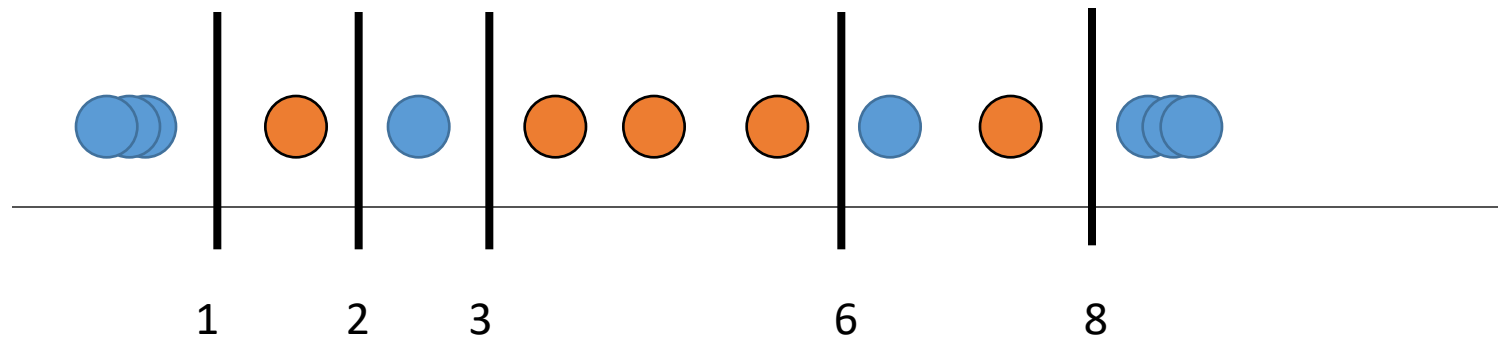
# Обучение деревьев



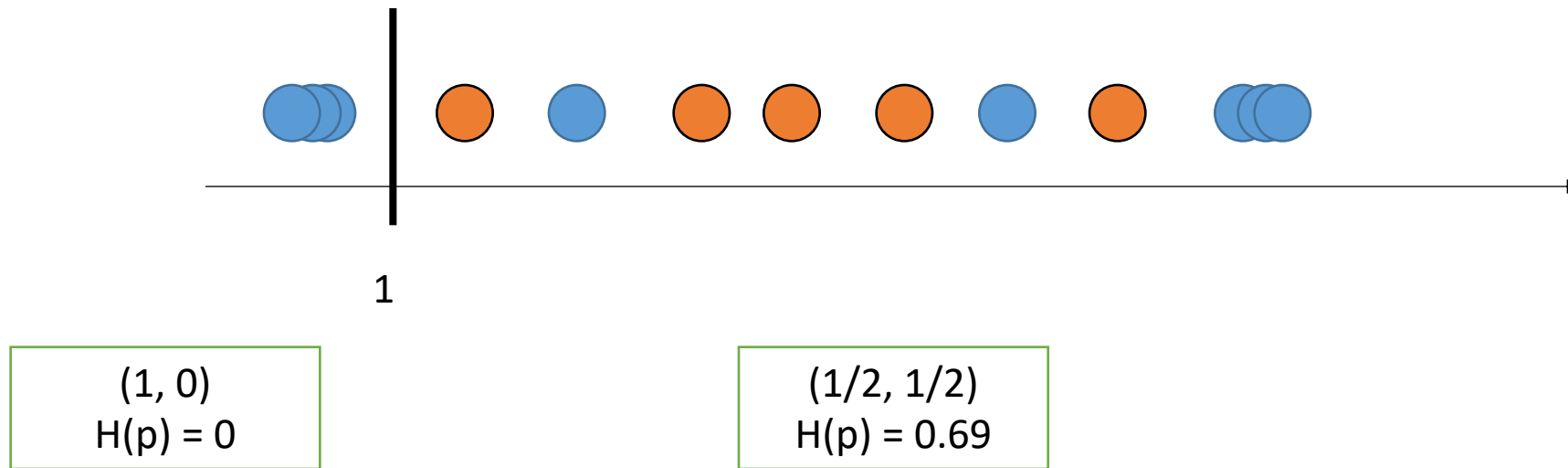
## Признаки



# Разбиения по признаку 1



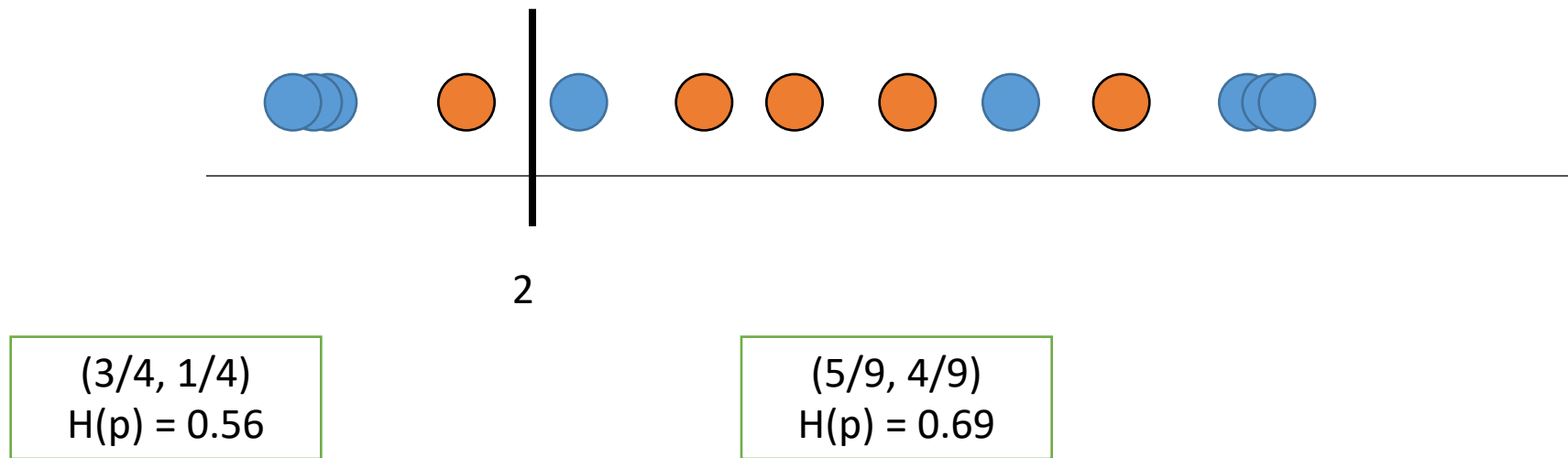
# Разбиения по признаку 1



$$\frac{3}{13}H(p_l) + \frac{10}{13}H(p_r) = 0.53$$

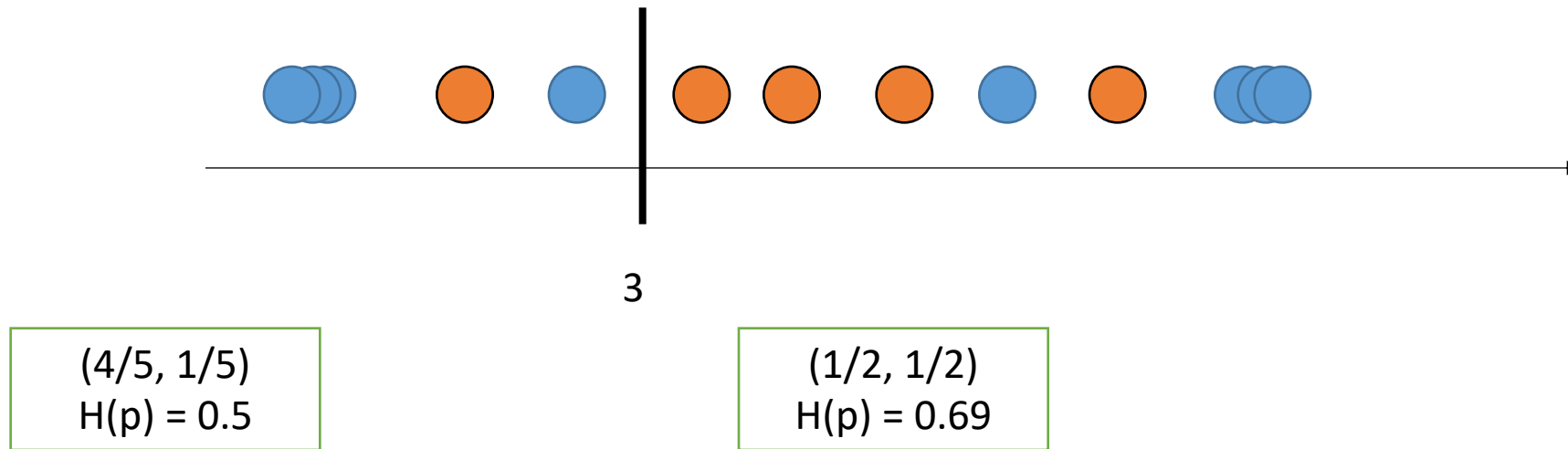


# Разбиения по признаку 1



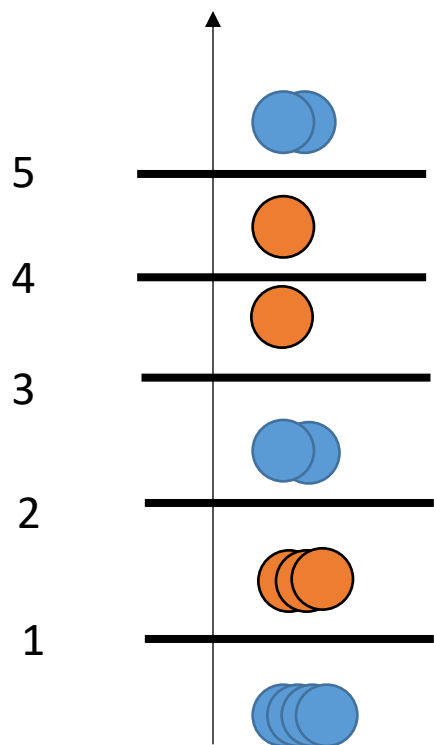
$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.65$$

# Разбиения по признаку 1

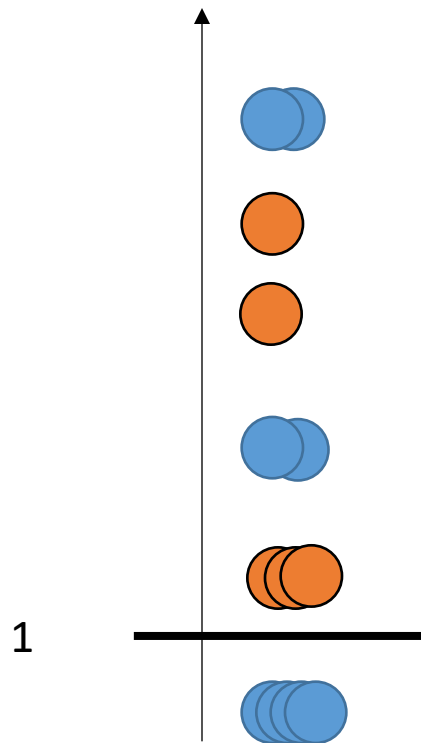


$$\frac{5}{13}H(p_l) + \frac{8}{13}H(p_r) = 0.62$$

# Разбиения по признаку 2



# Разбиения по признаку 2

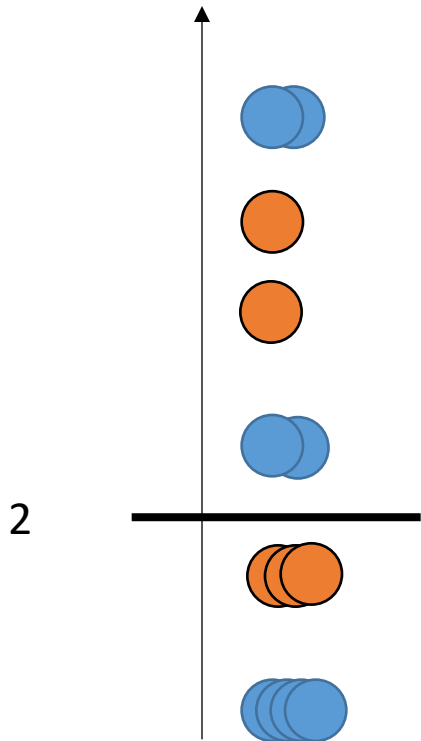


$(4/9, 5/9)$   
 $H(p) = 0.69$

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

$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.47$$

# Разбиения по признаку 2

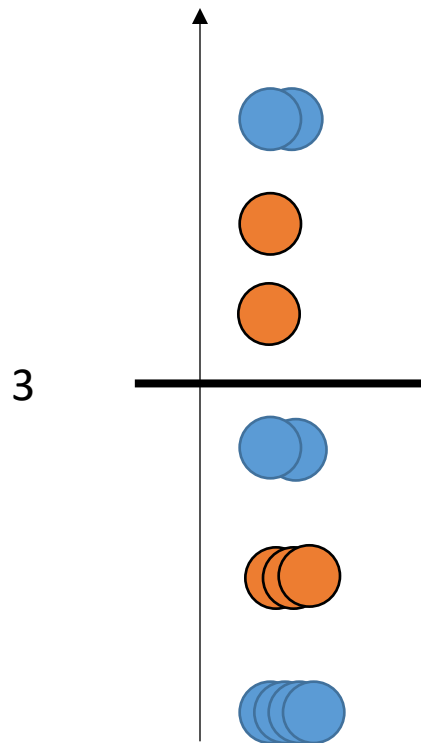


$(4/6, 2/6)$   
 $H(p) = 0.64$

$(4/7, 3/7)$   
 $H(p) = 0.68$

$$\frac{7}{13}H(p_l) + \frac{6}{13}H(p_r) = 0.66$$

# Разбиения по признаку 2

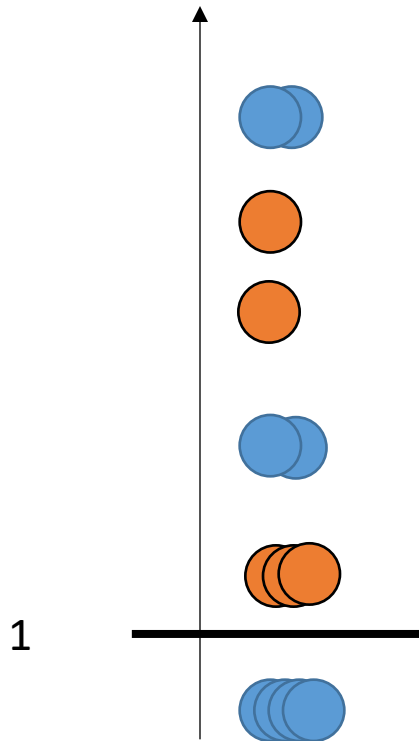


$(1/2, 1/2)$   
 $H(p) = 0.69$

$(6/9, 3/9)$   
 $H(p) = 0.46$

$$\frac{9}{13}H(p_l) + \frac{4}{13}H(p_r) = 0.53$$

# Разбиения по признаку 2



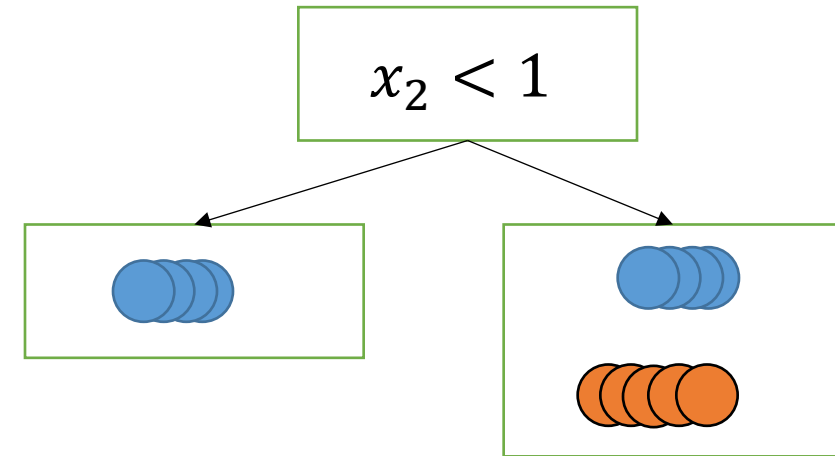
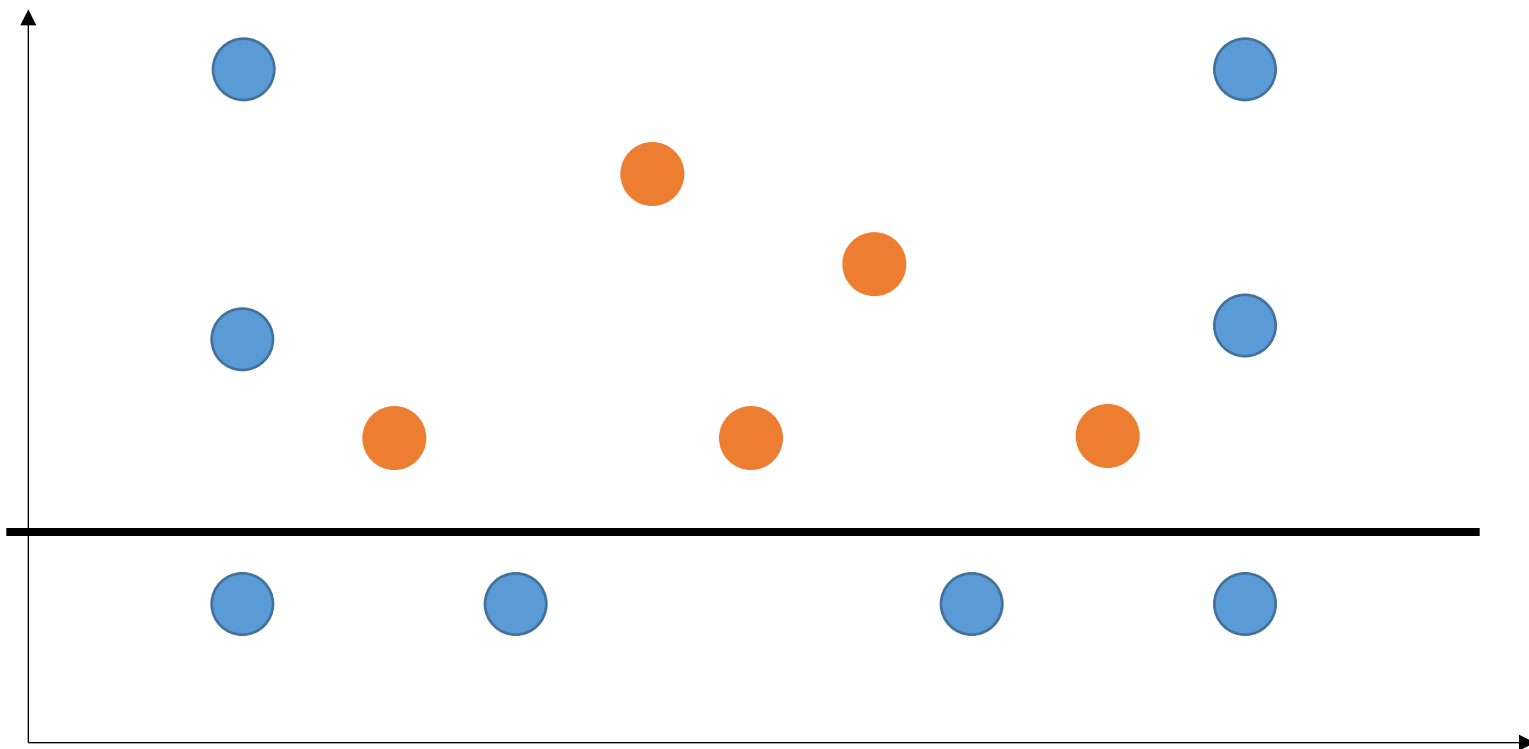
$(4/9, 5/9)$   
 $H(p) = 0.69$

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

$$\frac{4}{13}H(p_l) + \frac{9}{13}H(p_r) = 0.47$$

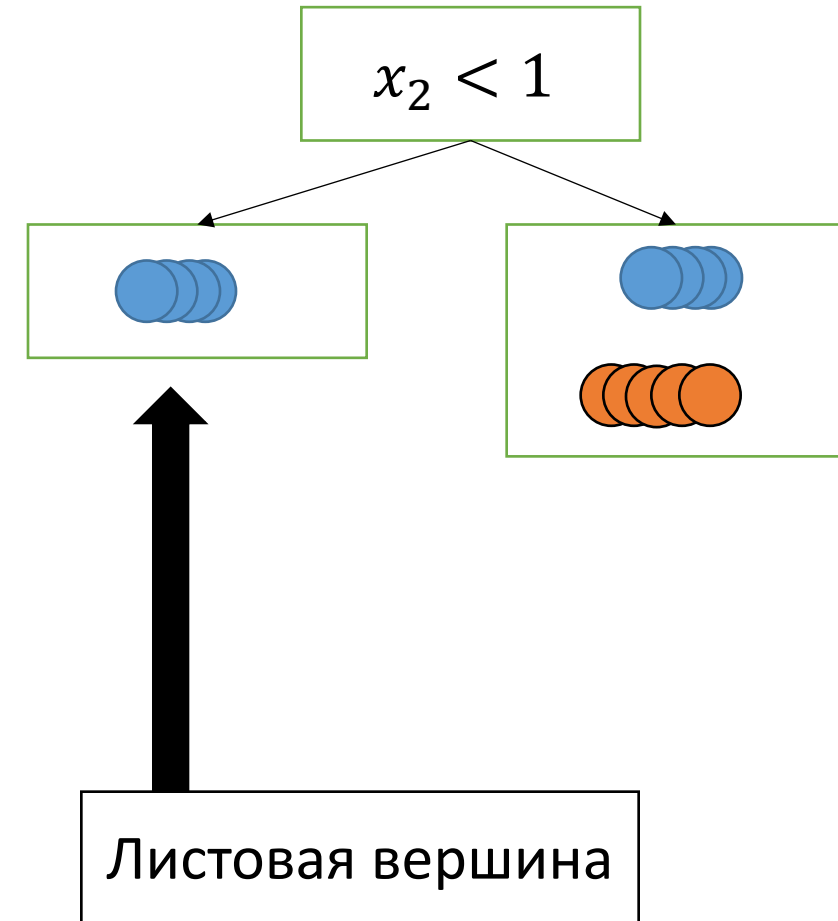
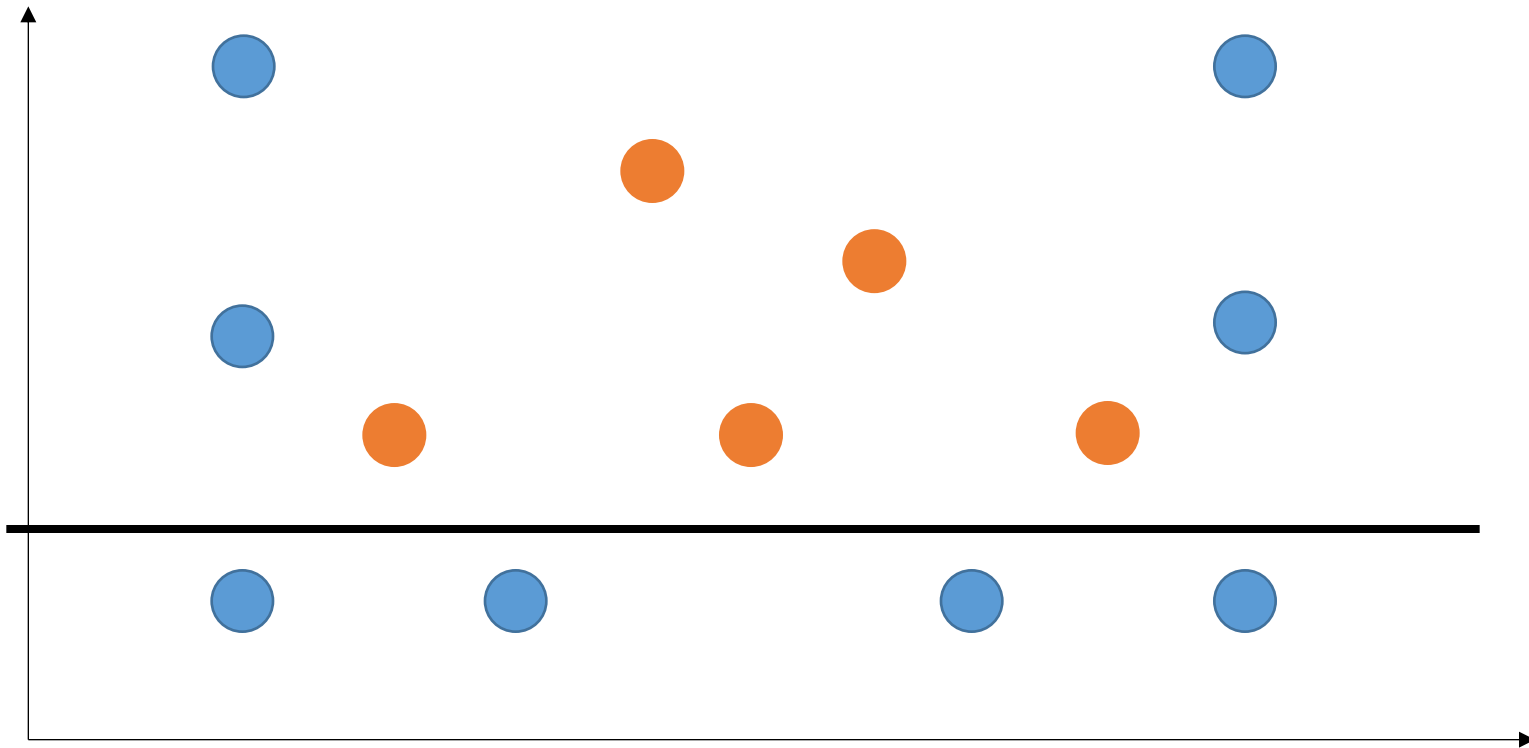
Лучшее разбиение!

# Обучение деревьев

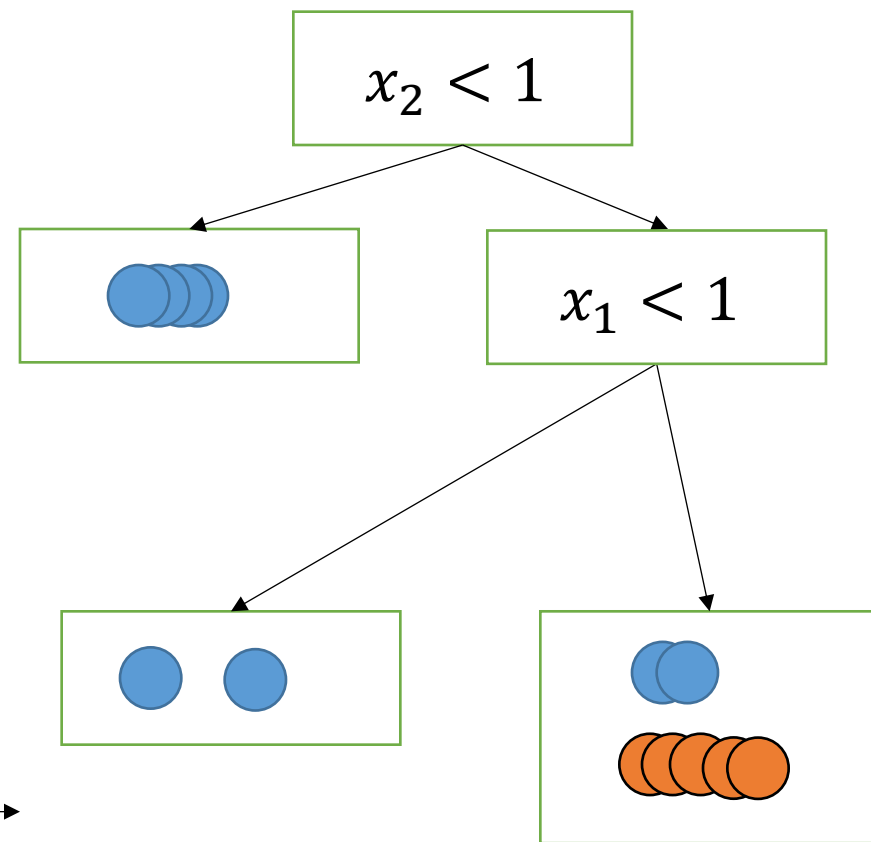
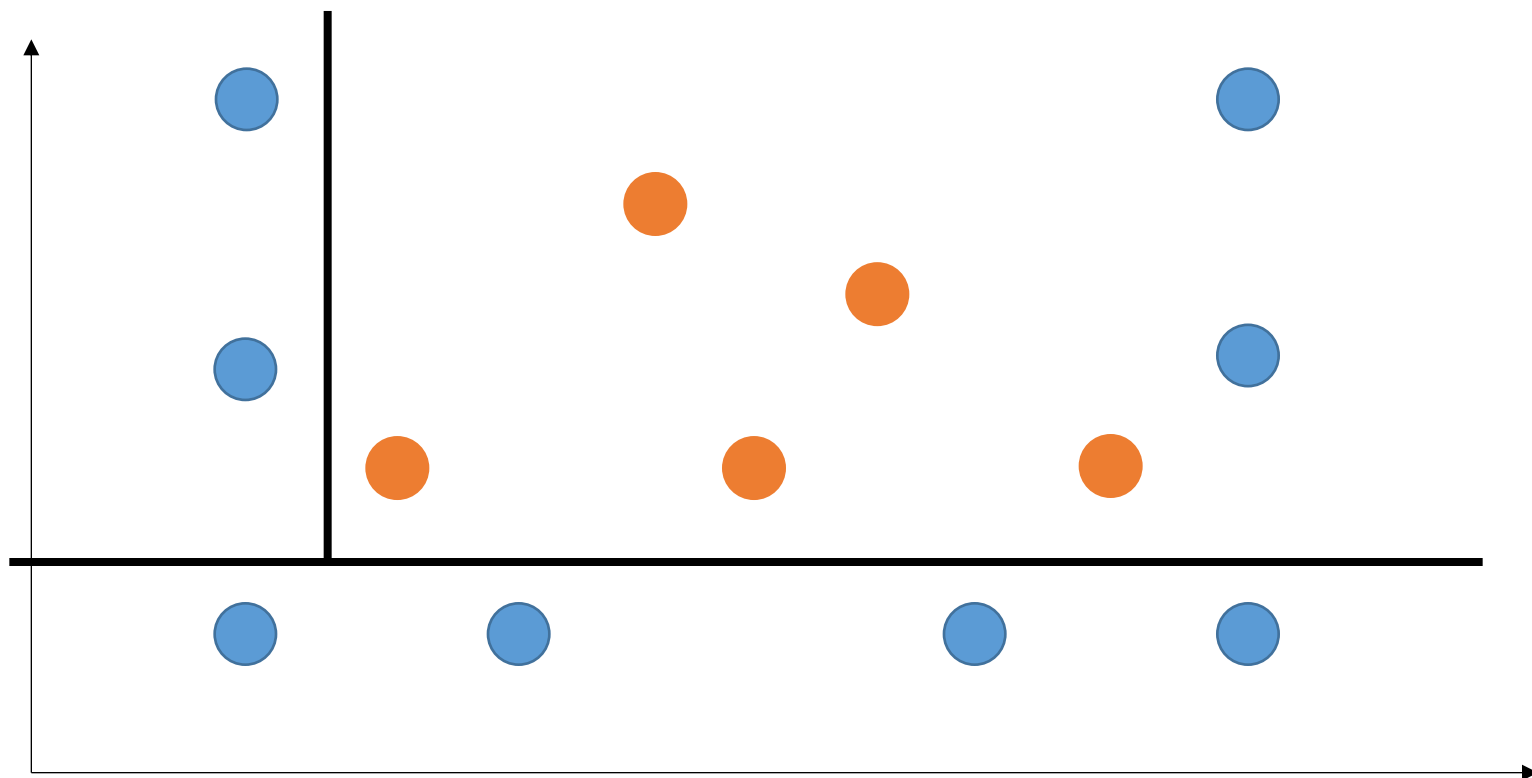




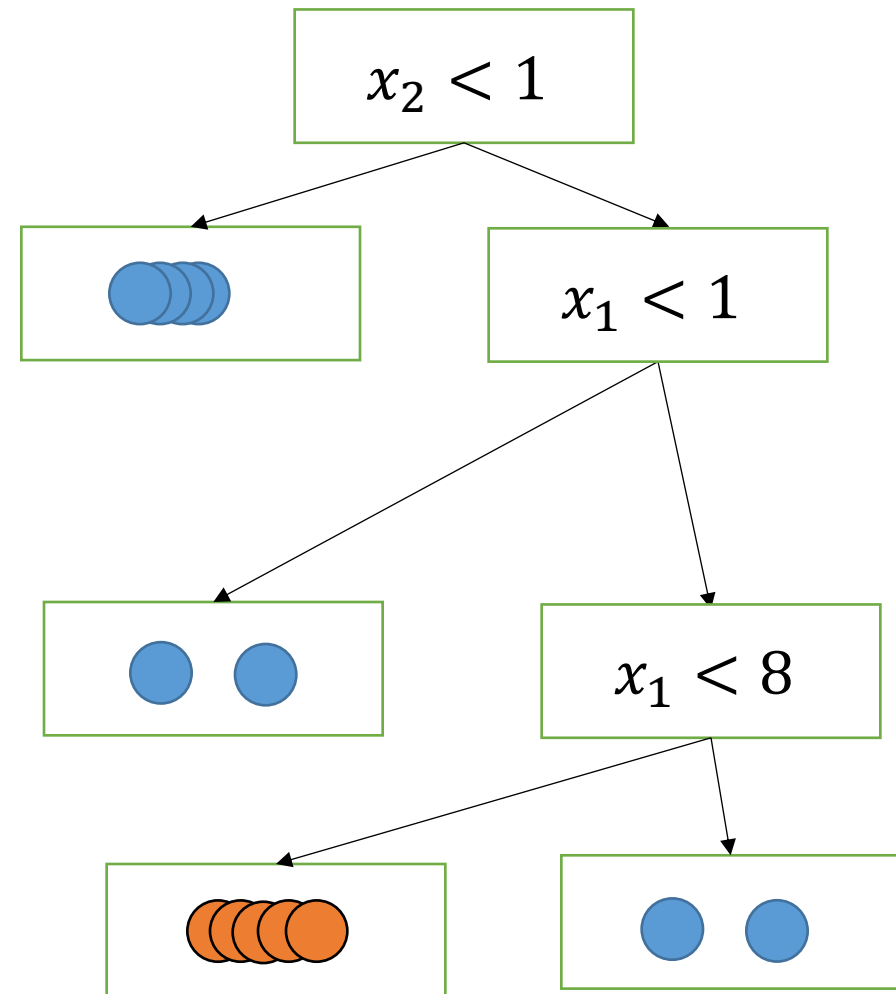
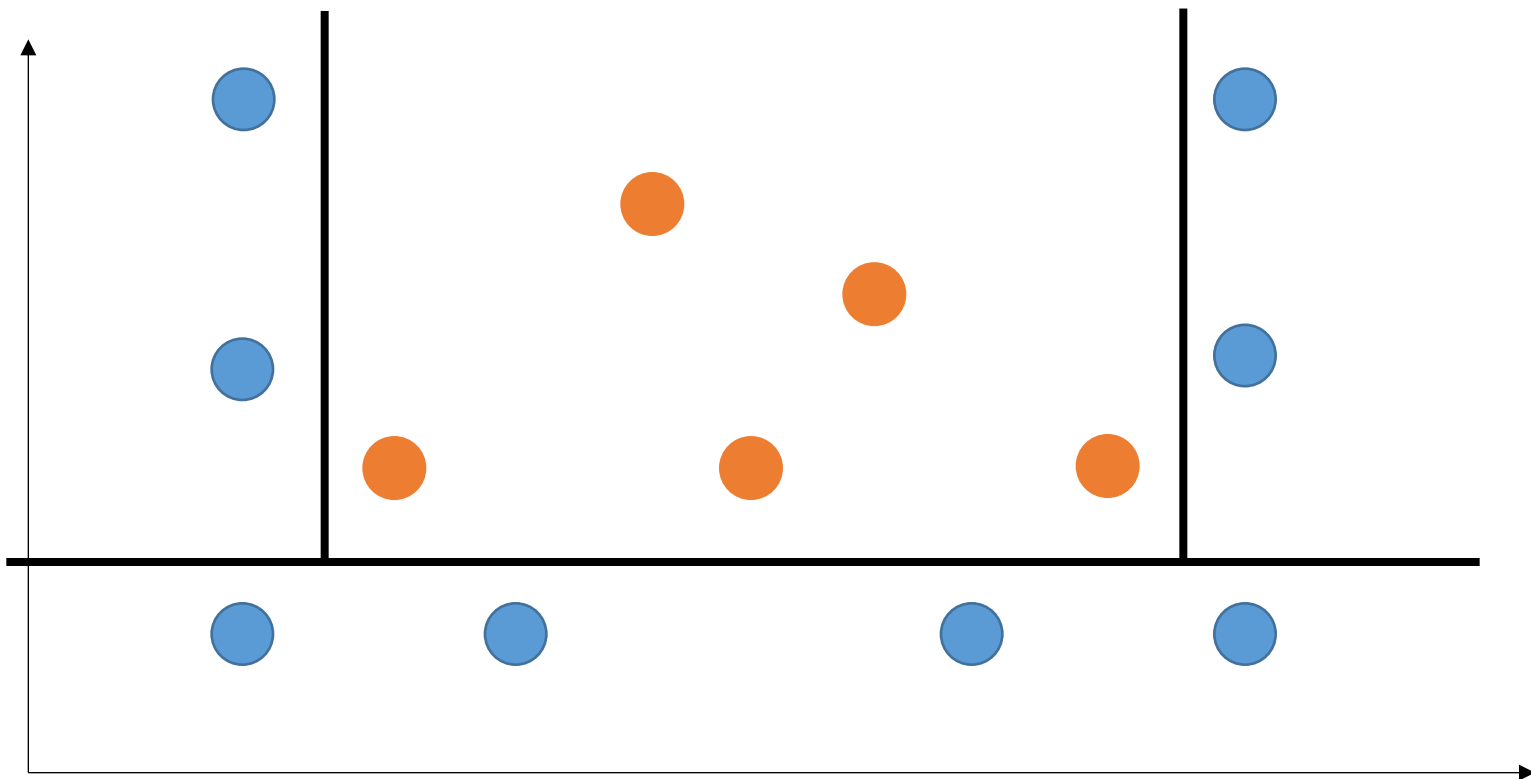
# Обучение деревьев



# Обучение деревьев



# Обучение деревьев



# Резюме

- Решающие деревья позволяют строить сложные модели, но есть риск переобучения
- Деревья строятся жадно, на каждом шаге вершина разбивается на две с помощью лучшего из предиктов
- Алгоритм довольно сложный и требует перебора всех предикатов на каждом шаге

Неустойчивость деревьев

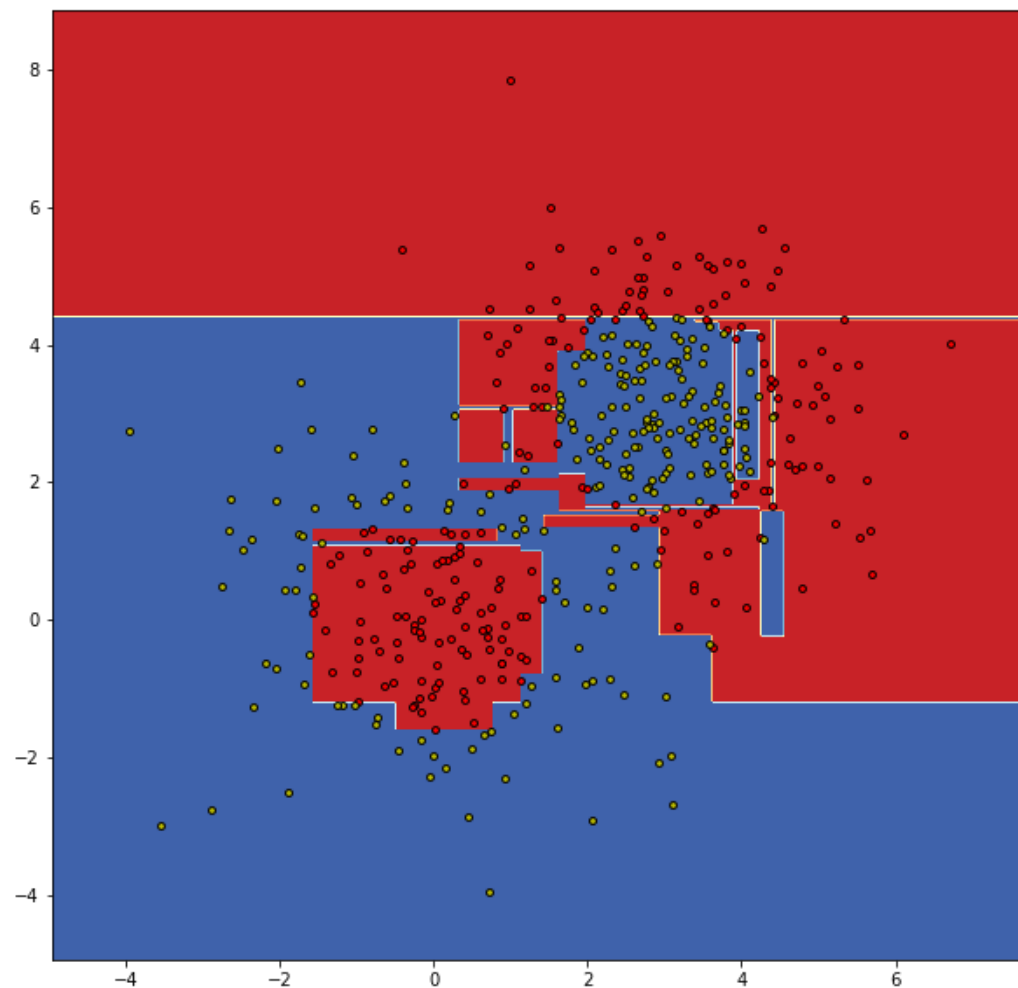
# Устойчивость моделей

- $X = (x_i, y_i)_{i=1}^{\ell}$  — 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

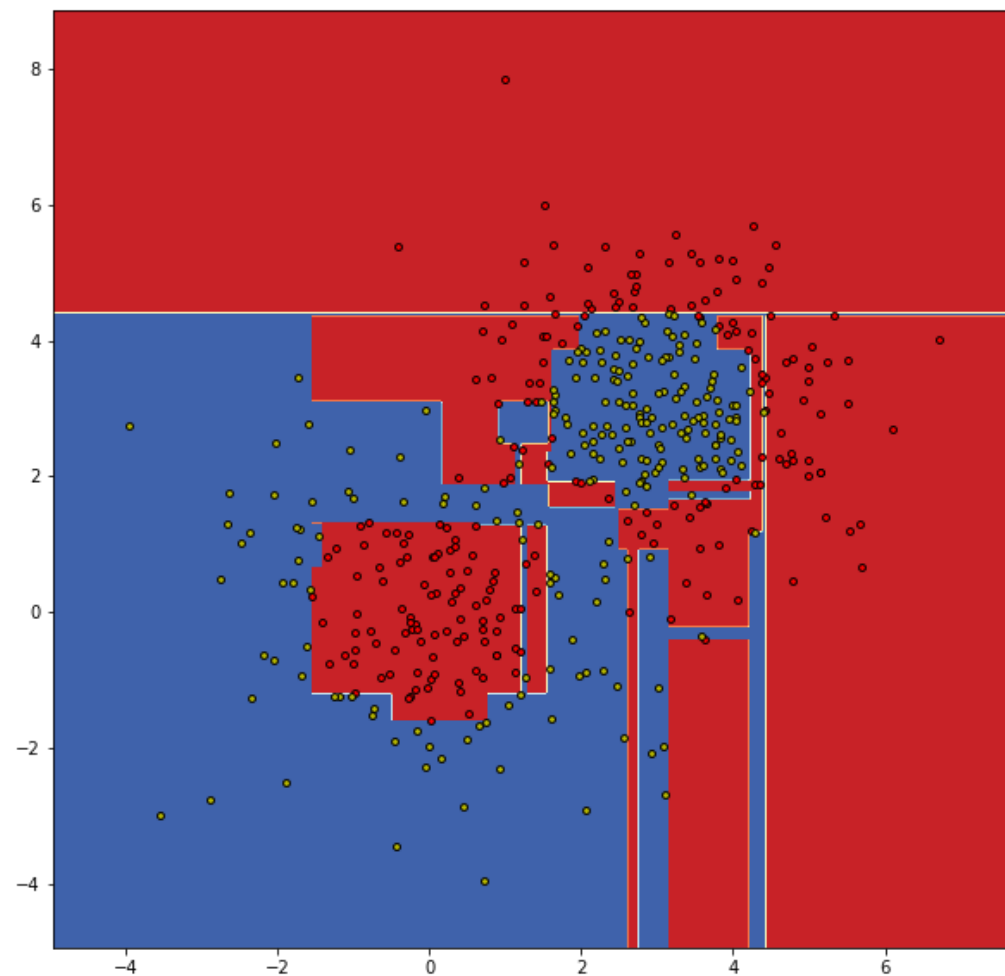
- $\tilde{X}$  — случайная подвыборка, примерно 90% исходной
- Что будет происходить с деревьями на разных подвыборках?

# Обучение на подвыборках

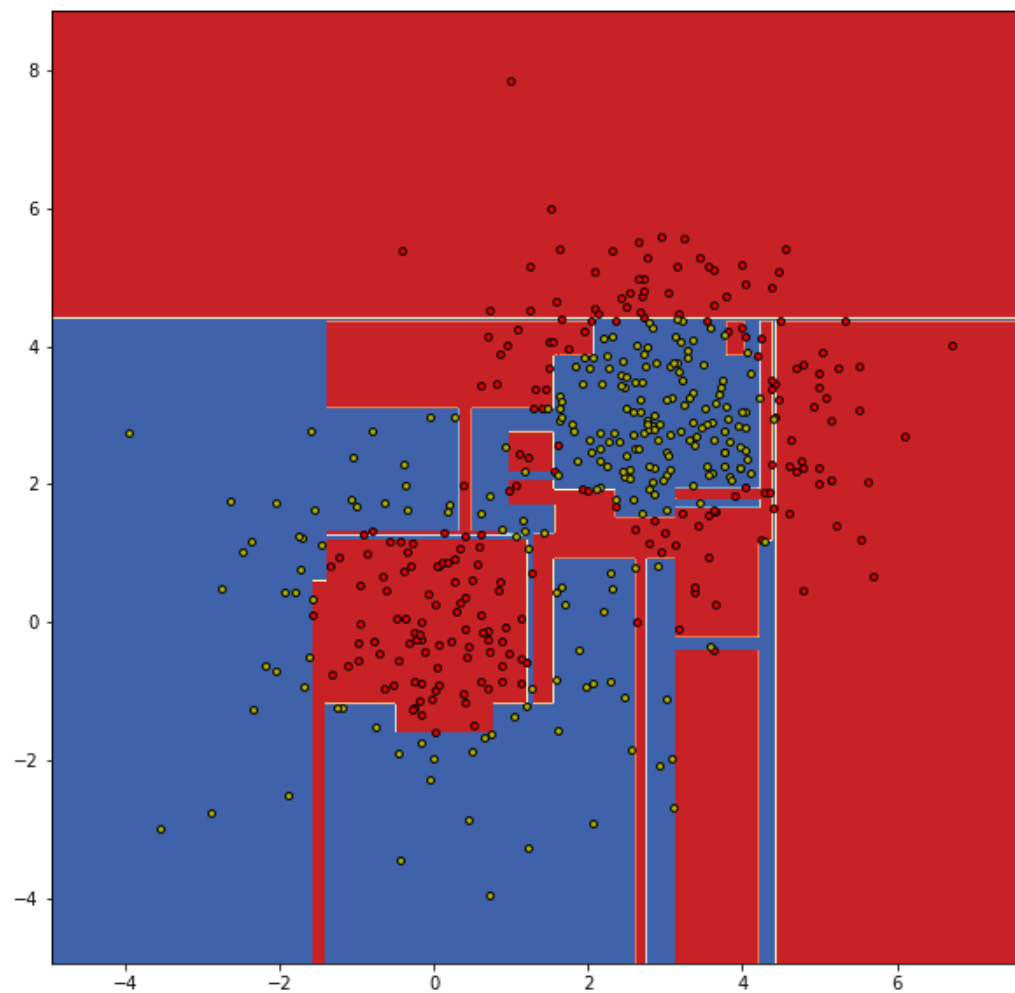




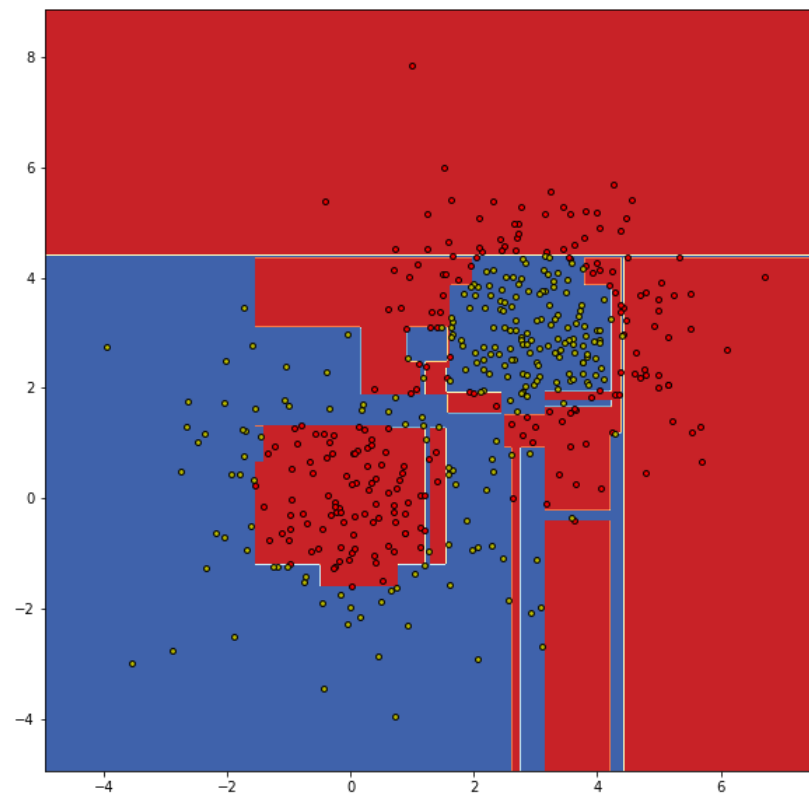
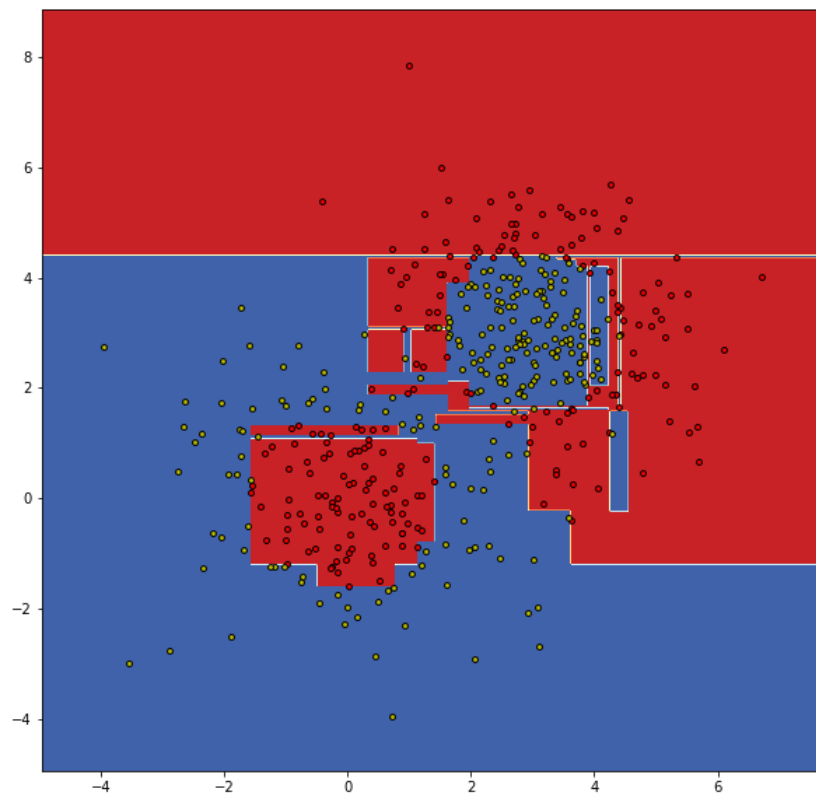
# Обучение на подвыборках



# Обучение на подвыборках



# Обучение на подвыборках

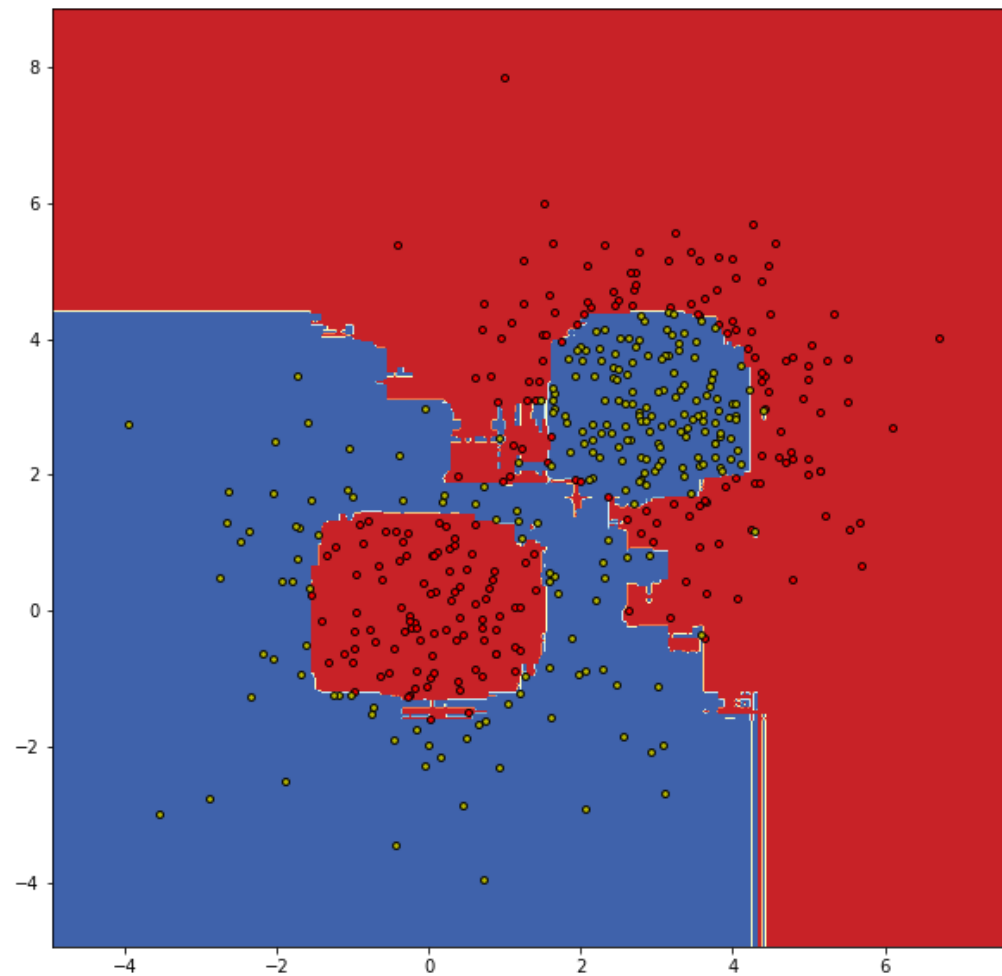


# Композиция моделей

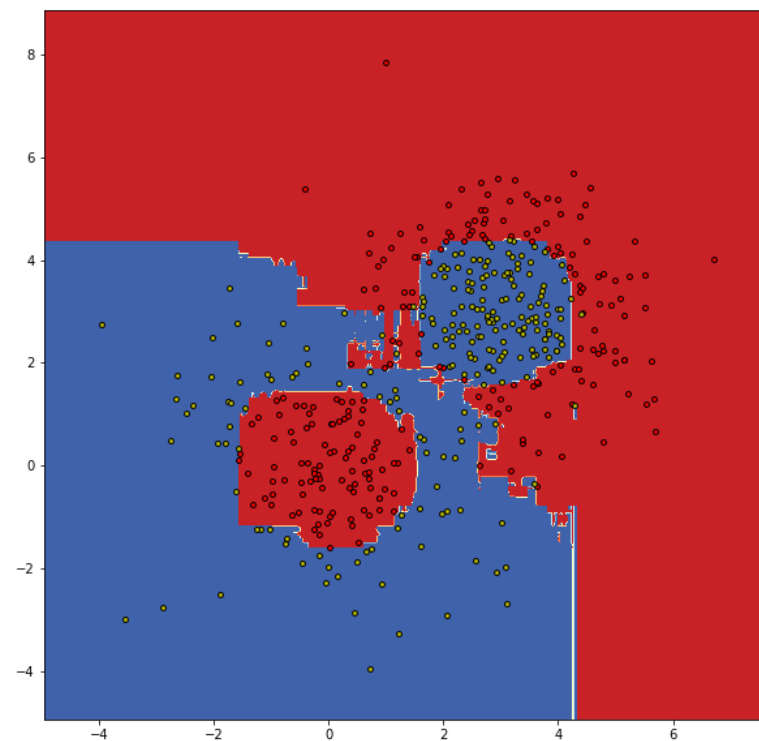
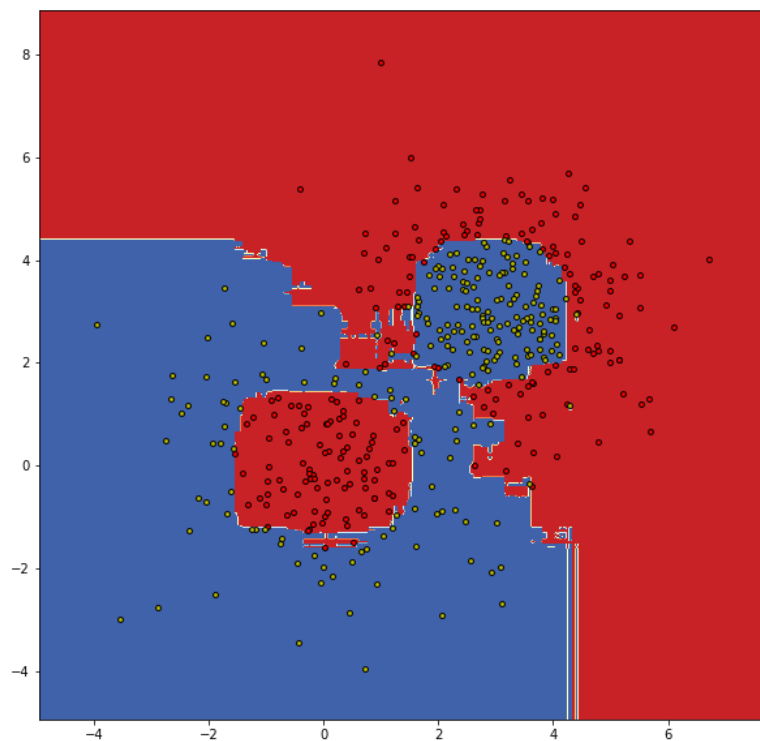
- У нас получилось  $N$  деревьев:  $b_1(x), \dots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

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

# Композиция моделей



# Композиция моделей



Голосование по большинству и  
усреднение

# Majority vote

- Какой из двух логотипов более старый?





# Majority vote

- Как выглядит корпус Вышки в Перми?



# Majority vote

- Покоординатный спуск — это метод оптимизации 1-го или 2-го порядка?

# Majority vote

- Дано:  $N$  базовых алгоритмов  $b_1(x), \dots, b_N(x)$
- 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

# Усреднение наблюдений

- Сколько лет факультету компьютерных наук?

# Усреднение наблюдений

- Сколько метров в 1 сажени?

# Усреднение наблюдений

- Сколько лет лектору?

Композиции моделей



# Общий вид: классификация

- $b_1(x), \dots, b_N(x)$  — 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

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

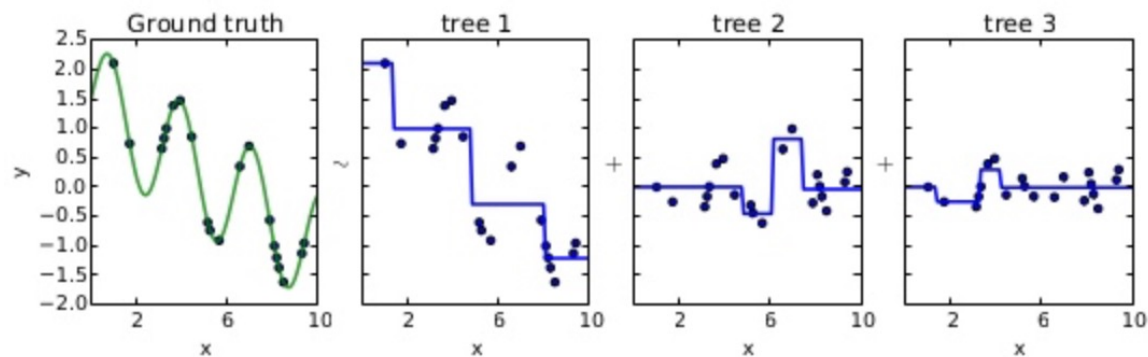
$$a_N(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

# Базовые модели

- $b_1(x), \dots, b_N(x)$  — 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  $\ell$  элементов из  $X$
- Пример:  $\{x_1, x_2, x_3, x_4\} \rightarrow \{x_1, x_2, x_2, x_4\}$
- В подвыборке будет  $\ell$  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

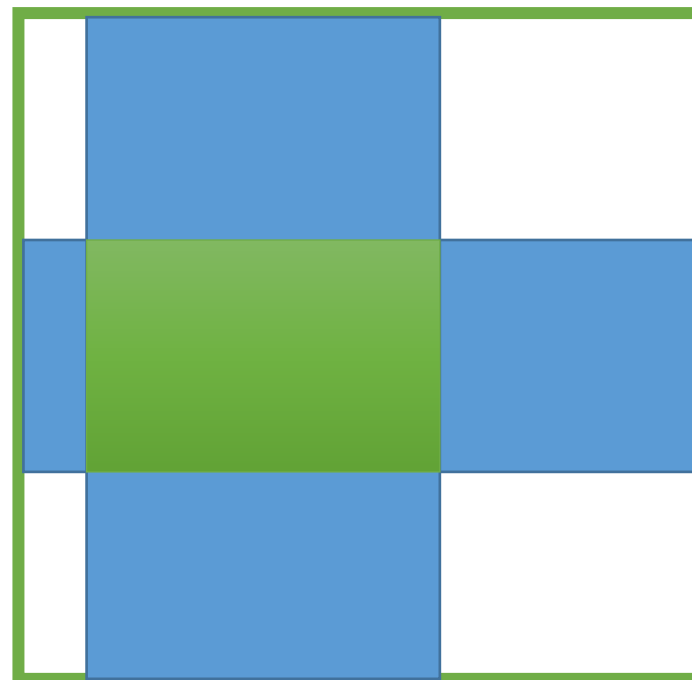
# Случайные подпространства

- 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

$$\begin{aligned} L(\mu) = & \underbrace{\mathbb{E}_{x,y} \left[ (y - \mathbb{E}[y | x])^2 \right]}_{\text{шум}} + \\ & \underbrace{\mathbb{E}_x \left[ (\mathbb{E}_X [\mu(X)] - \mathbb{E}[y | x])^2 \right]}_{\text{смещение}} + \underbrace{\mathbb{E}_x \left[ \mathbb{E}_X \left[ (\mu(X) - \mathbb{E}_X [\mu(X)])^2 \right] \right]}_{\text{разброс}} \end{aligned}$$

- Разберём на уровне идеи

# Разложение ошибки на смещение и разброс

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

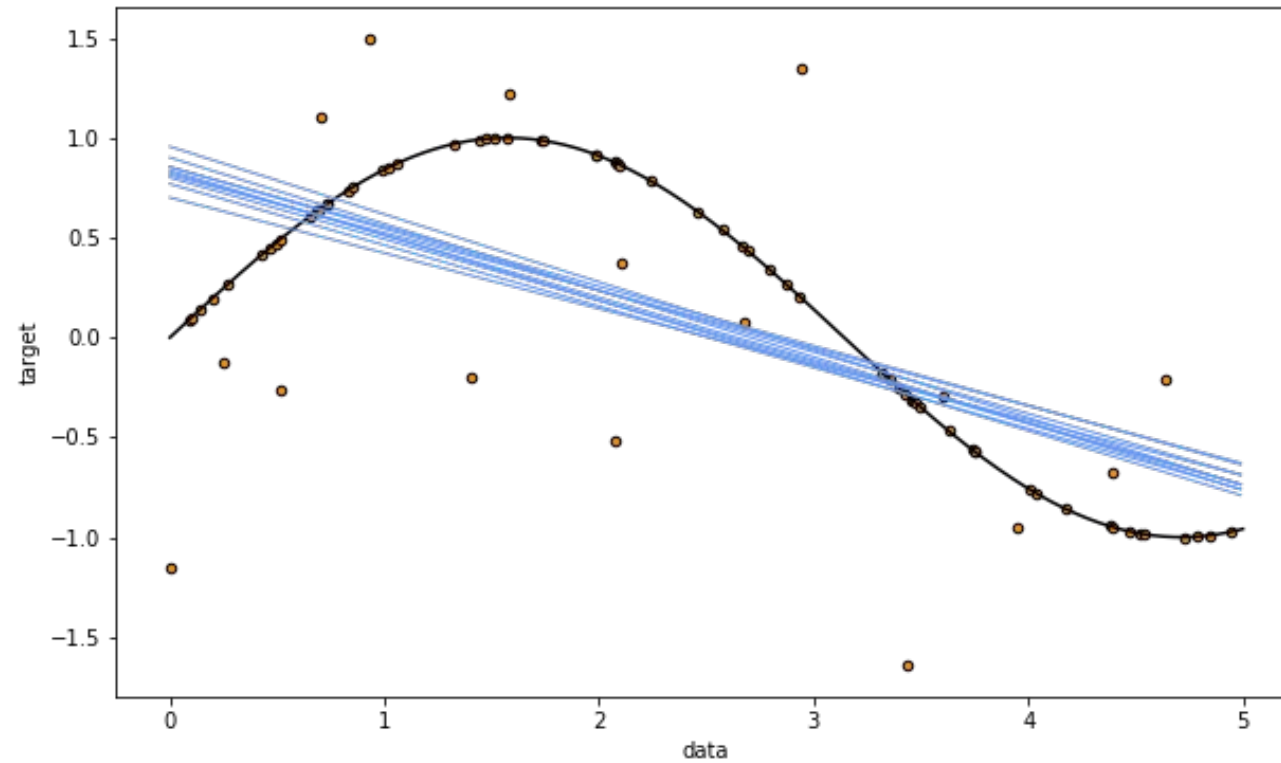
# Разложение ошибки на смещение и разброс

- 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

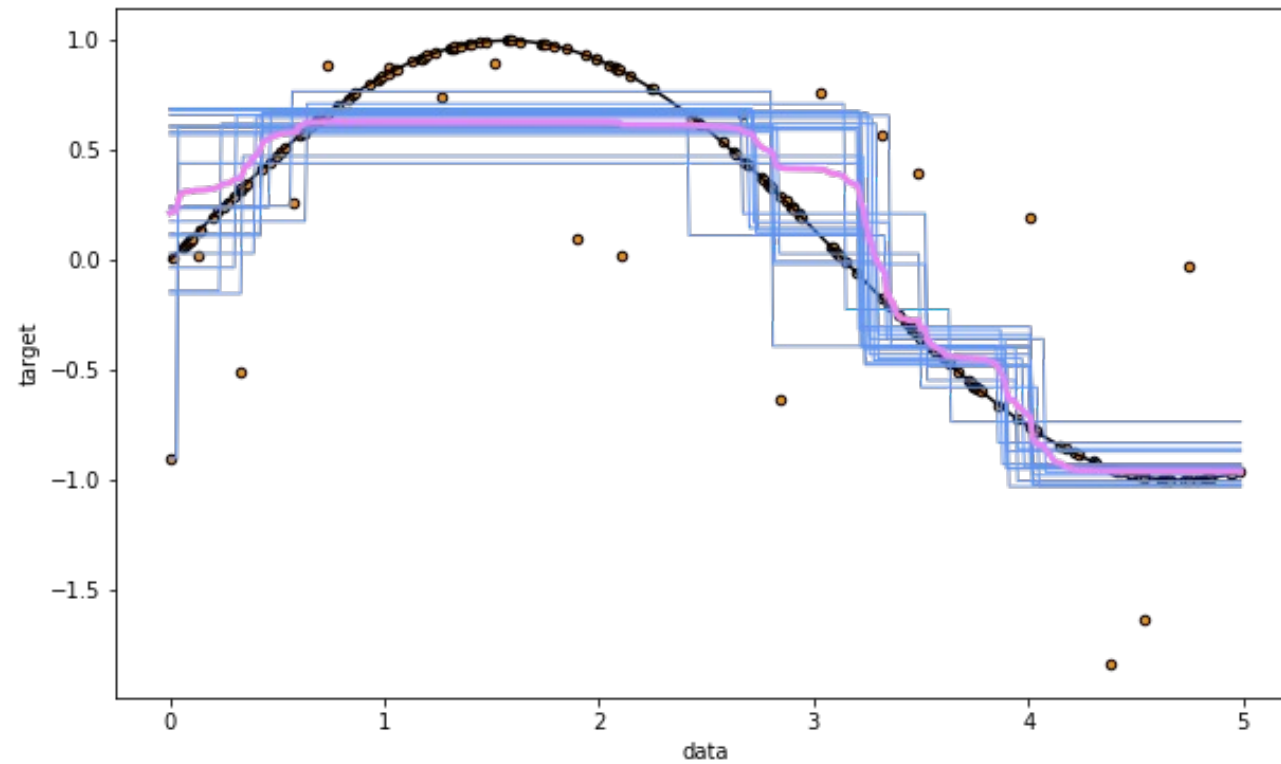
- Ошибка модели складывается из трёх компонент
- Шум (noise) — характеристика сложности и противоречивости данных
- Смещение (bias) — способность модели приблизить лучшую среди всех возможных моделей
- Разброс (variance) — устойчивость модели к изменениям в обучающей выборке

# Смещение и разброс: линейная модель

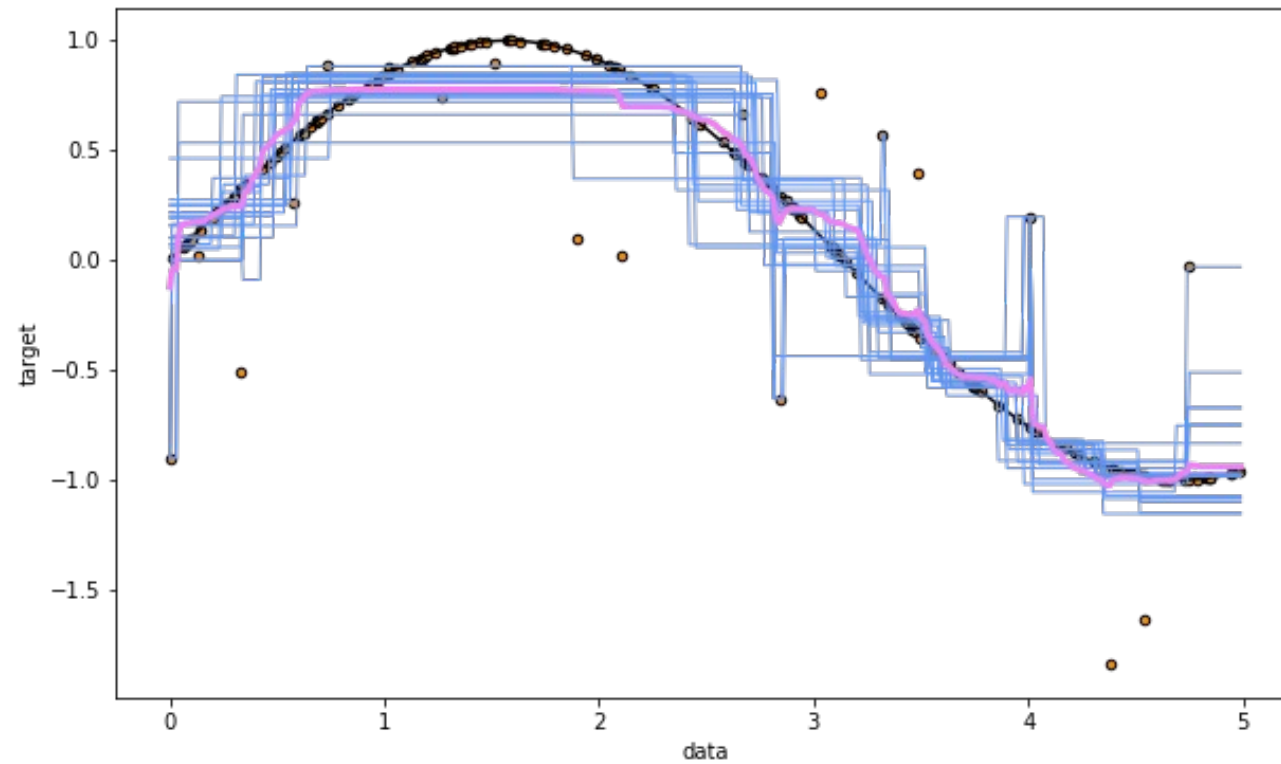




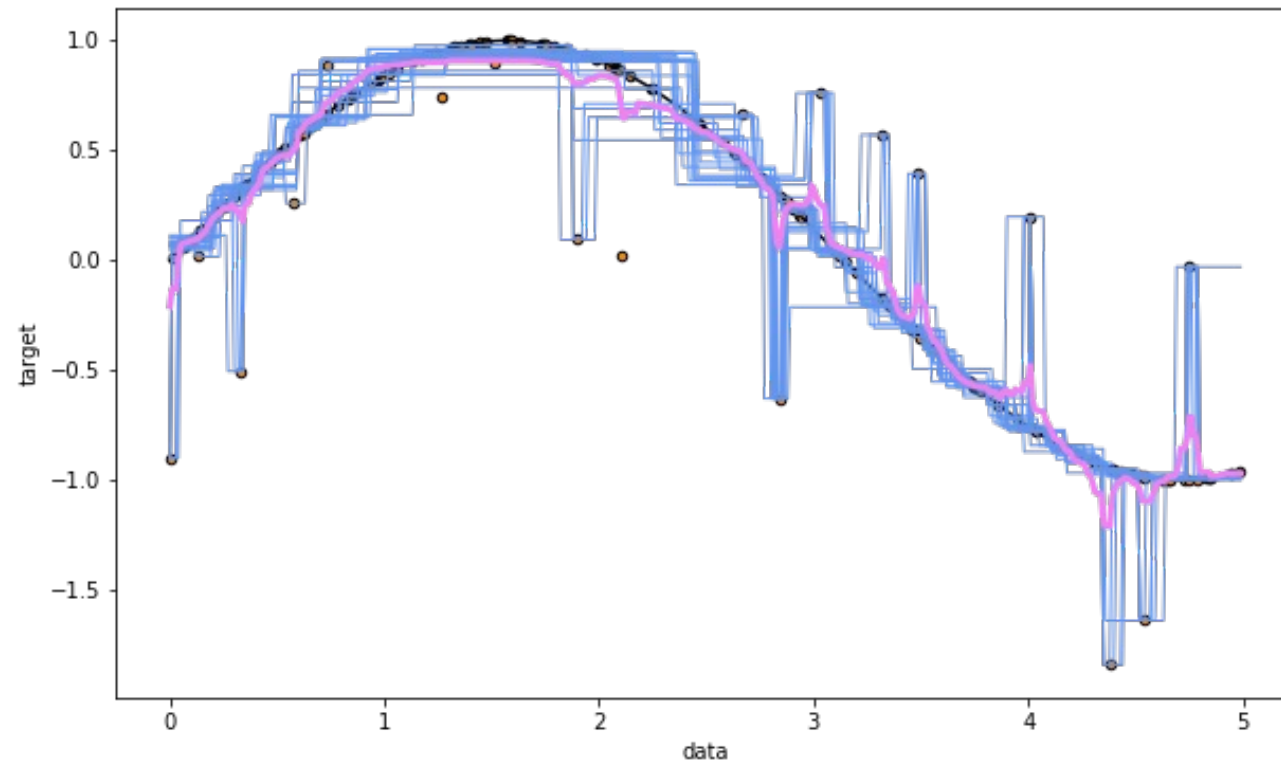
# Смещение и разброс: деревья



# Смещение и разброс: деревья



# Смещение и разброс: деревья



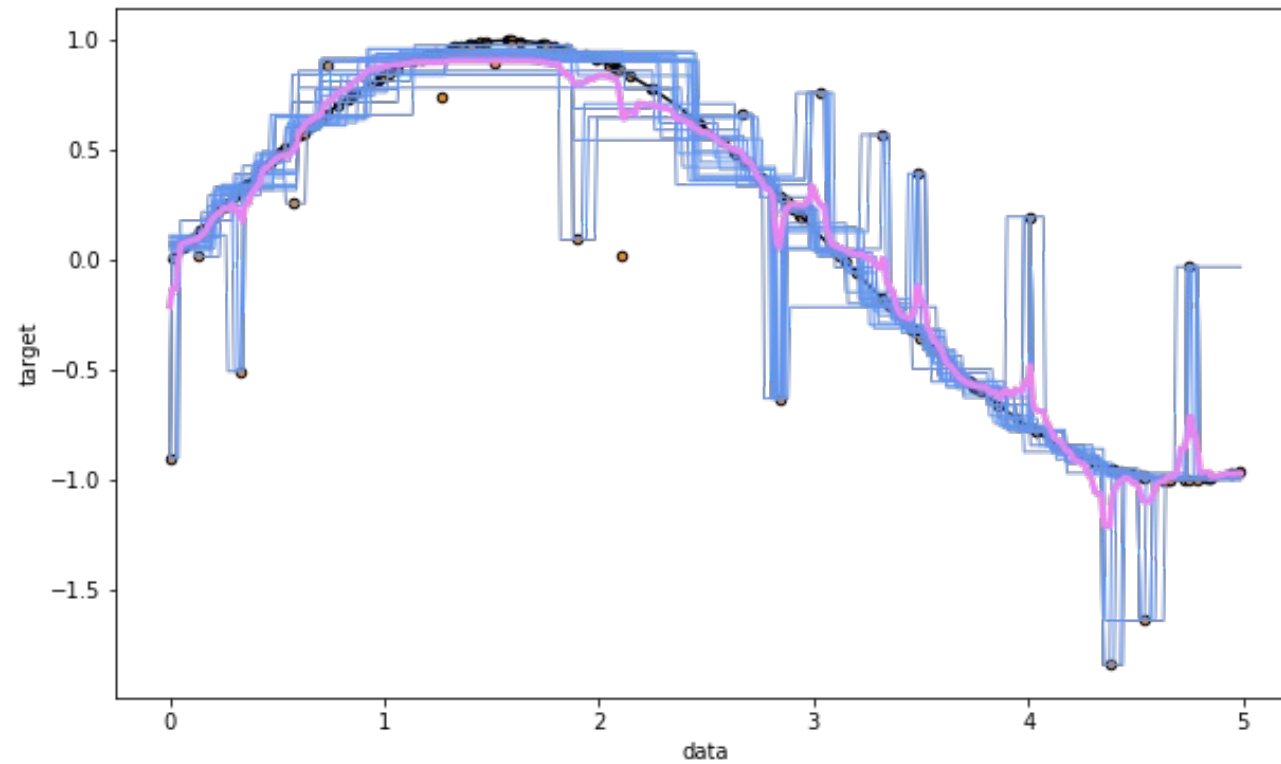
# Бэггинг

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

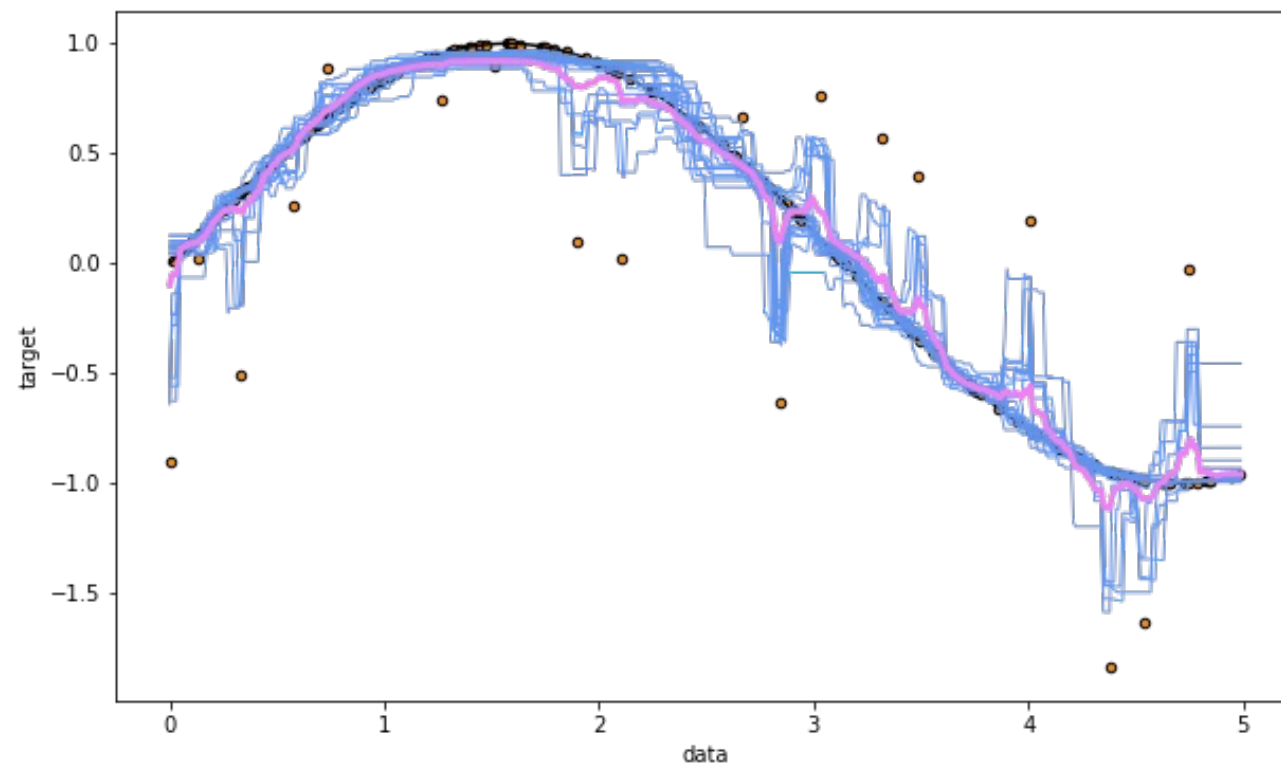
$$\frac{1}{N} (\text{разброс } b_n(x)) + \text{ковариация}(b_n(x), b_m(x))$$

- 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( $r, R_r$ )

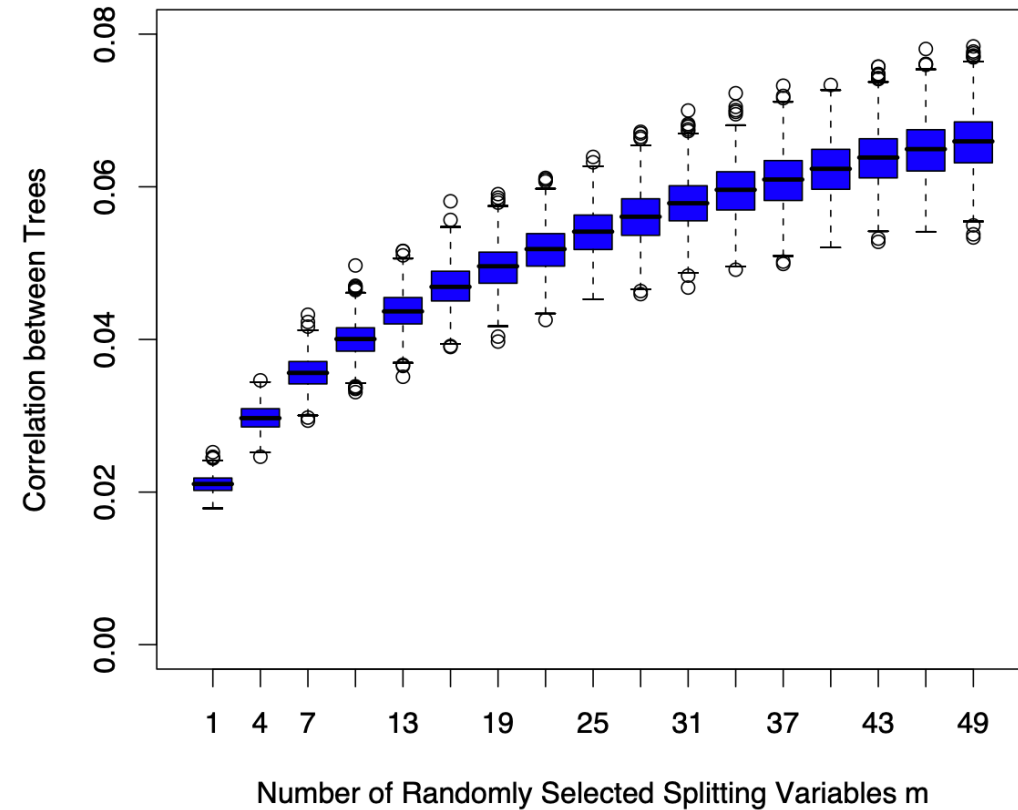
# Выбор предиката

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

- 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:

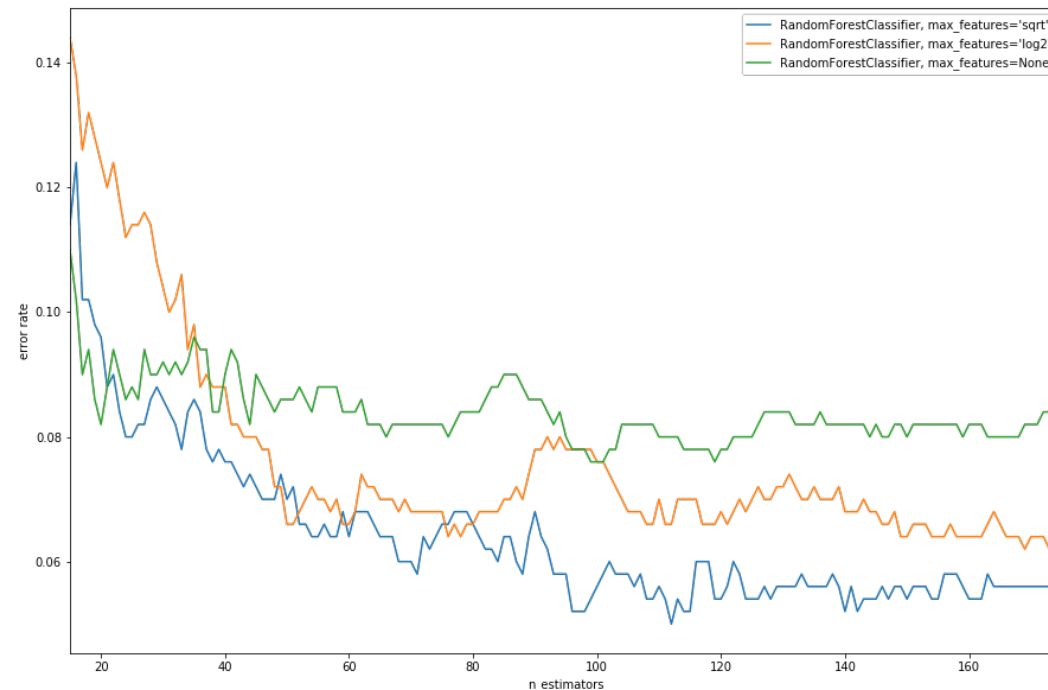
$$a(x) = \frac{1}{N} \sum_{n=1}^N b_n(x)$$

- 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