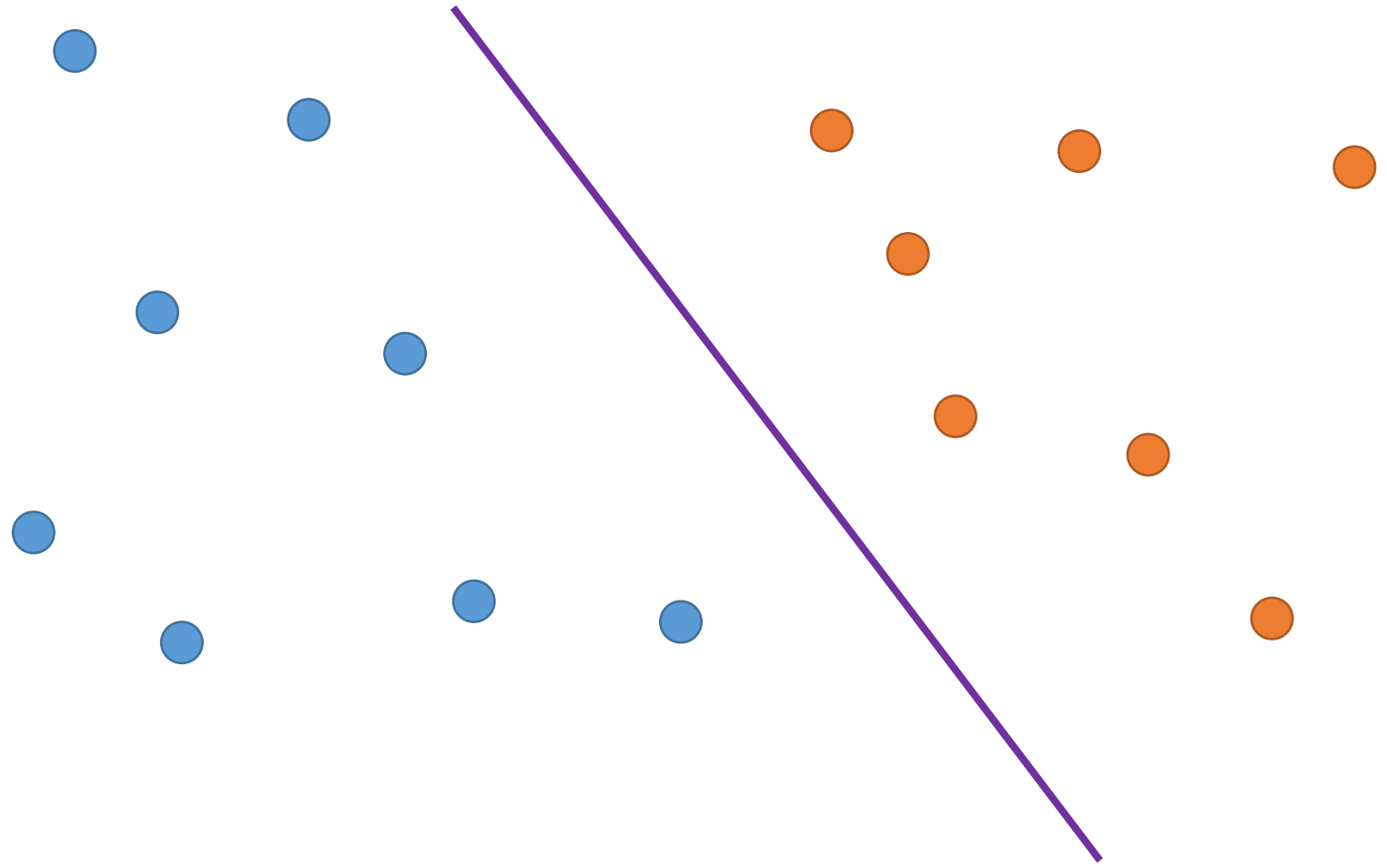
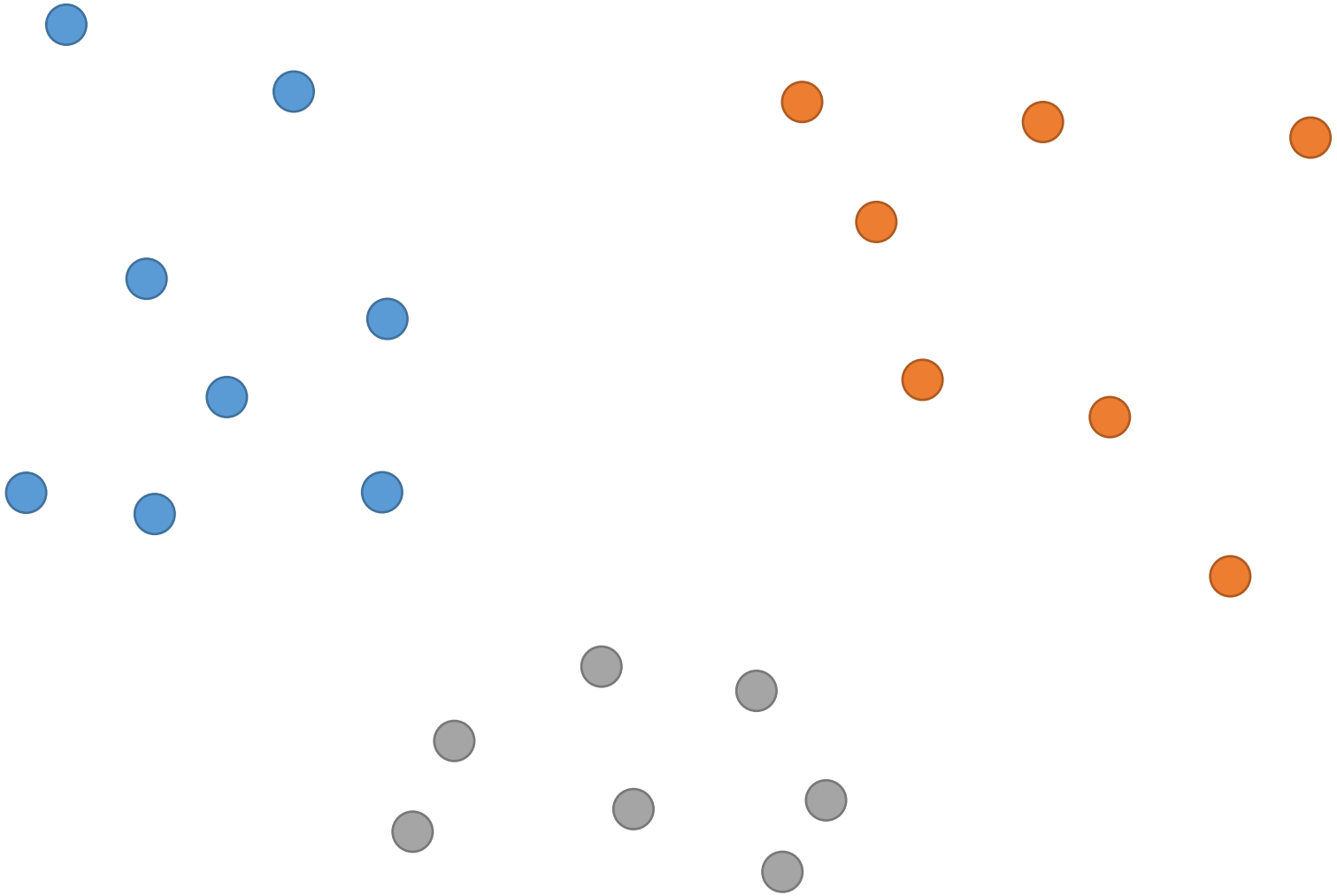
Решающие деревья и Ансамбли

Многоклассовая классификация

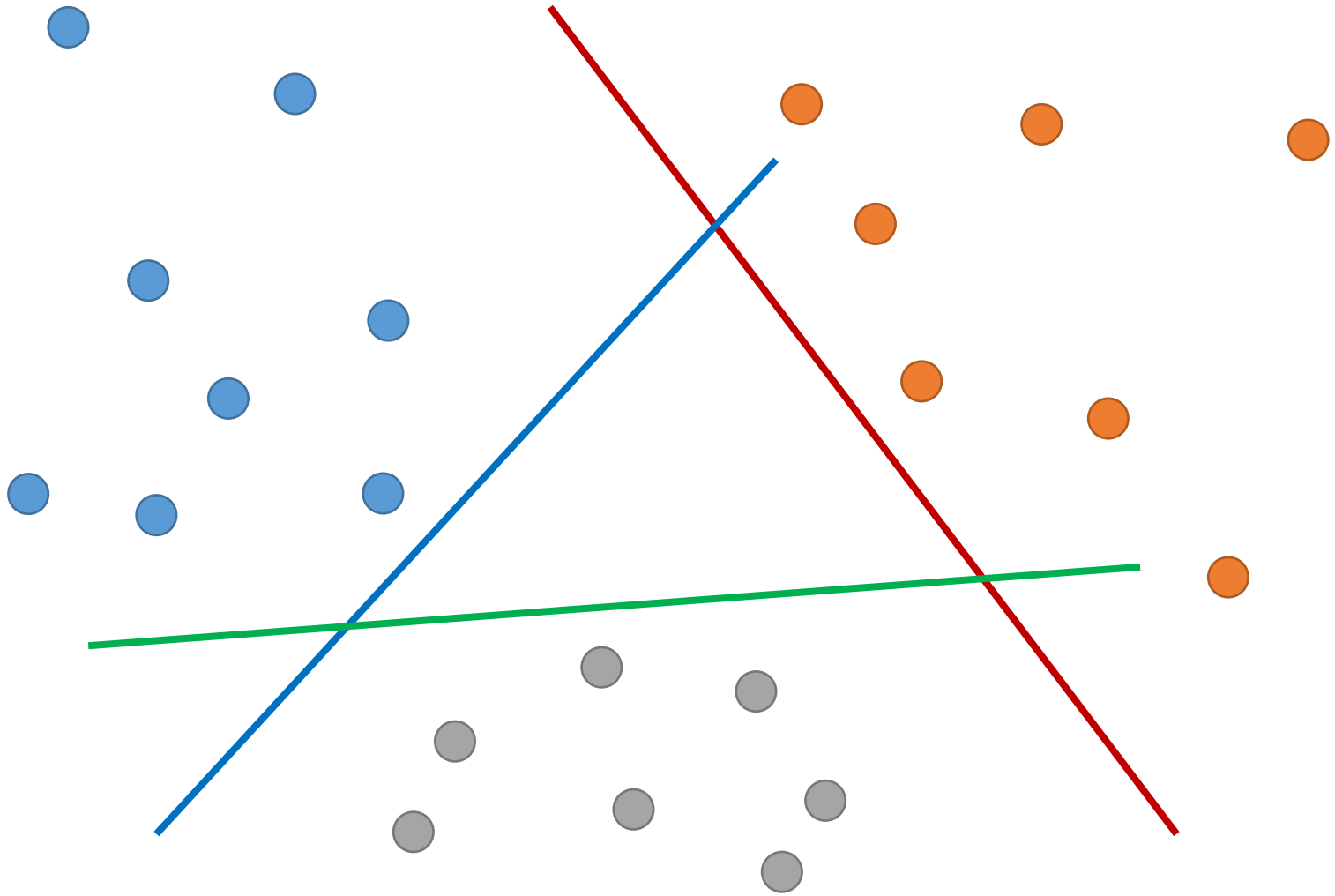
Бинарная классификация



Многоклассовая классификация



Многоклассовая классификация



One-vs-all

- K классов: $\mathbb{Y} = \{1, \dots, K\}$
- $X_k = (x_i, [y_i = k])_{i=1}^{\ell}$
- Обучаем $a_k(x)$ на X_k, $k = 1, \dots, K$
- $a_k(x)$ должен выдавать оценки принадлежности классу (например, $\langle w, x \rangle$ или $\sigma(\langle w, x \rangle)$)
- Итоговая модель:

$$a(x) = \arg \max_{k=1,\dots K} a_k(x)$$

One-vs-all

- Модель $a_k(x)$ при обучении не знает, что её выходы будут сравнивать с выходами других моделей
- Нужно обучать K моделей

All-vs-all

- $X_{km} = \{(x_i, y_i) \in X \mid y_i = k \text{ или } y_i = m\}$
- Обучаем $a_{km}(x)$ на X_{km}
- Итоговая модель:

$$a(x) = \arg \max_{k \in \{1,\ldots,K\}} \sum_{m=1}^{K} [a_{km}(x) = k]$$

All-vs-all

- Нужно обучать порядка K^2 моделей
- Зато каждую обучаем на небольшой выборке

Доля ошибок

- Функционал ошибки — доля ошибок (error rate)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

- Нередко измеряют долю верных ответов (accuracy):

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i]$$

- Подходит для многоклассового случая!

Общие подходы

Микро-усреднение	Макро-усреднение
Вычисляем $\mathrm{TP}_k, \mathrm{FP}_k, \mathrm{FN}_k, \mathrm{TN}_k$ для каждого класса	Вычисляем нужную метрику для каждого класса (например, $\mathrm{precision}_1, \ldots, \mathrm{precision}_K$)
Суммируем по всем классам, получаем $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}, \mathrm{TN}$	Усредняем по всем классам
Подставляем их в формулу для precision/recall/…	
Крупные классы вносят больший вклад	Игнорирует размеры классов

Как делать нелинейные модели

Предсказание стоимости квартиры

- Признаки: площадь, этаж, расстояние до метро и т.д.
- Целевая переменная: рыночная стоимость квартиры

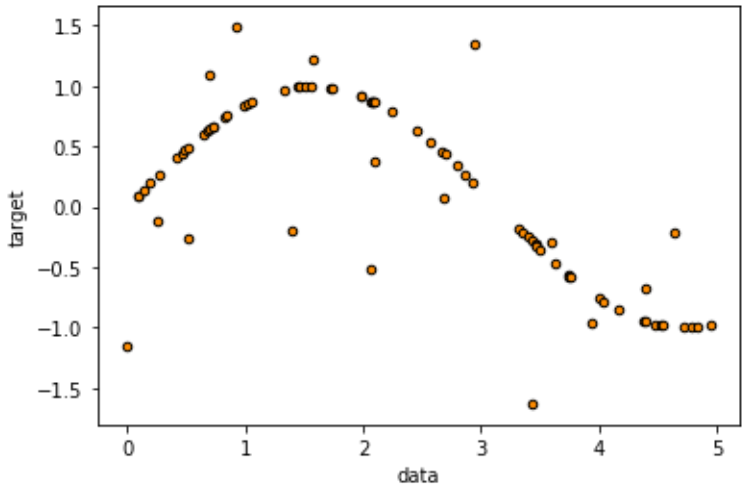
Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$

$$+w_3 * (\text{расстояние до метро}) + \cdots$$

- Вряд ли признаки линейно связаны с целевой переменной



Предсказание стоимости квартиры

- Линейная модель:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+w_3 * (\text{расстояние до метро}) + \cdots$$

- Вряд ли признаки не связаны между собой

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2$$
$$+w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2$$
$$+w_7 * (\text{площадь}) * (\text{этаж}) + \cdots$$

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:

$$a(x) = w_0 + w_1 * (\text{площадь}) + w_2 * (\text{этаж})$$
$$+w_3 * (\text{расстояние до метро}) + w_4 * (\text{площадь})^2$$
$$+w_5 * (\text{этаж})^2 + w_6 * (\text{расстояние до метро})^2$$
$$+w_7 * (\text{площадь}) * (\text{этаж}) + \cdots$$

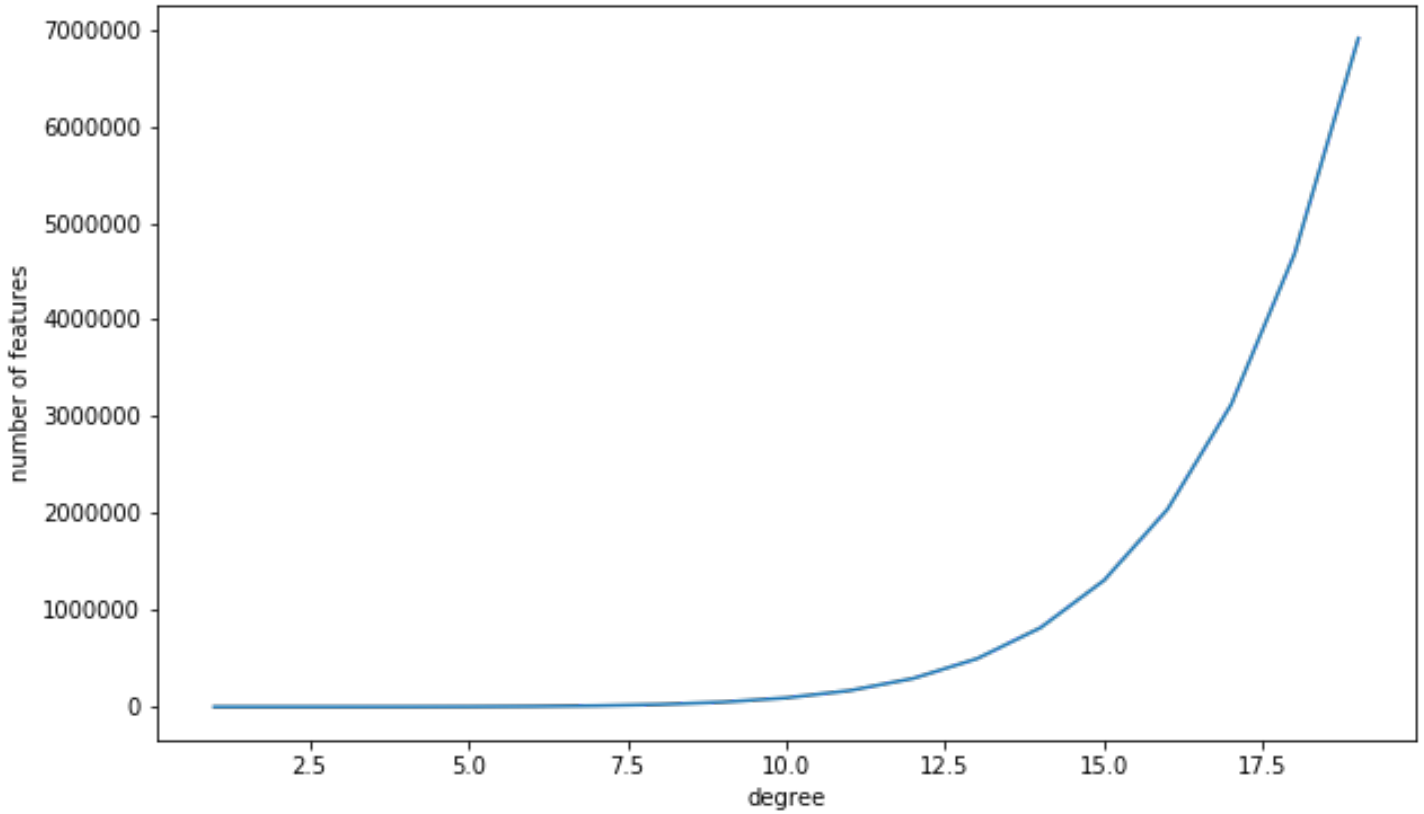
- Может быть сложно интерпретировать модель
- Что такое $(\text{расстояние до метро}) * (\text{этаж})^2$?

Предсказание стоимости квартиры

- Допустим, изначально имеем 10 признаков
- Полиномиальных степени 2: 55
- Полиномиальных степени 3: 220
- Полиномиальных степени 4: 715

Предсказание стоимости квартиры

- Линейная модель с полиномиальными признаками:



Предсказание стоимости квартиры

- Линейная модель с полиномиальными бинаризованными признаками:

$$a(x) = w_0 + w_1 * [30 < \text{площадь} < 50]$$

$$+w_2 * [50 < \text{площадь} < 80] + \cdots$$

$$+w_{20} * [2 < \text{этаж} < 5] + \cdots$$

$$+w_{100} * [30 < \text{площадь} < 50][2 < \text{этаж} < 5] + \cdots$$

- Признаки интерпретируются куда лучше: $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][100 < \text{расстояние до метро} < 500]$
- Но их станет ещё больше!

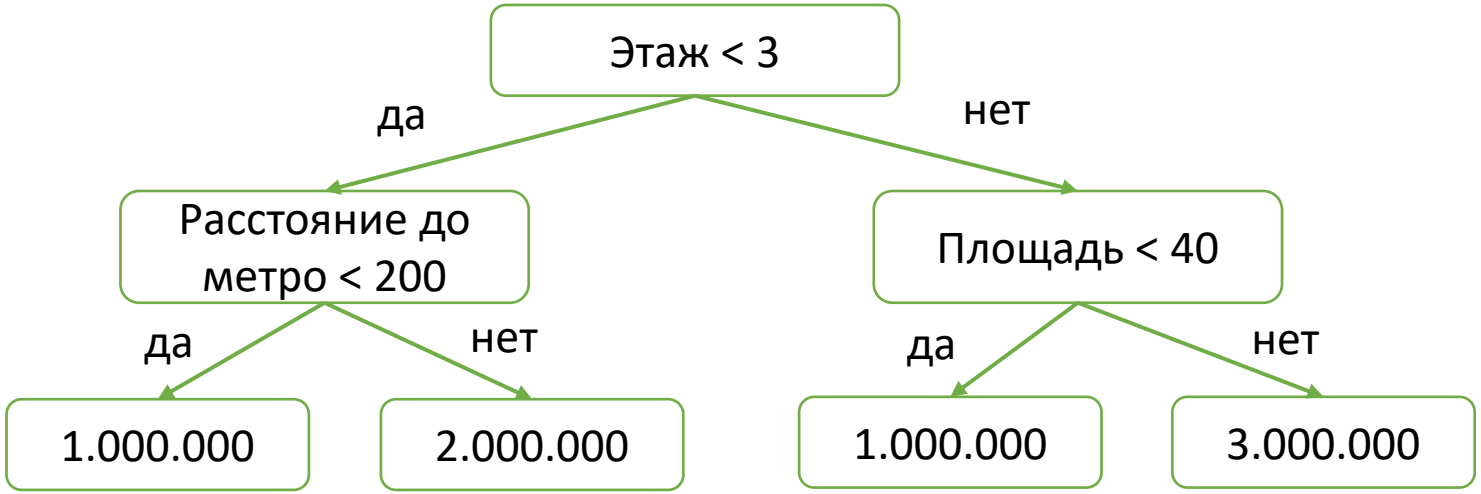
Решающие деревья

Логические правила

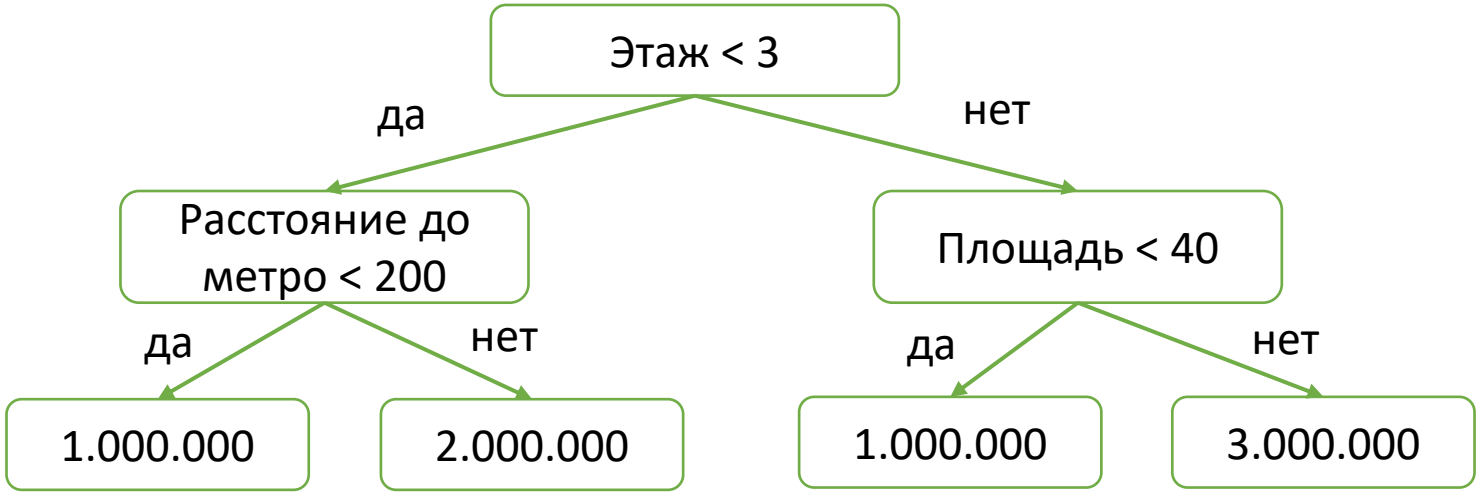
- $[30 < \text{площадь} < 50][2 < \text{этаж} < 5][500 < \text{расстояние до метро} < 1000]$
- Легко объяснить, как работают
- Находят нелинейные закономерности

- Нужно как-то искать хорошие логические правила
- Нужно уметь составлять модели из логических правил

Решающее дерево

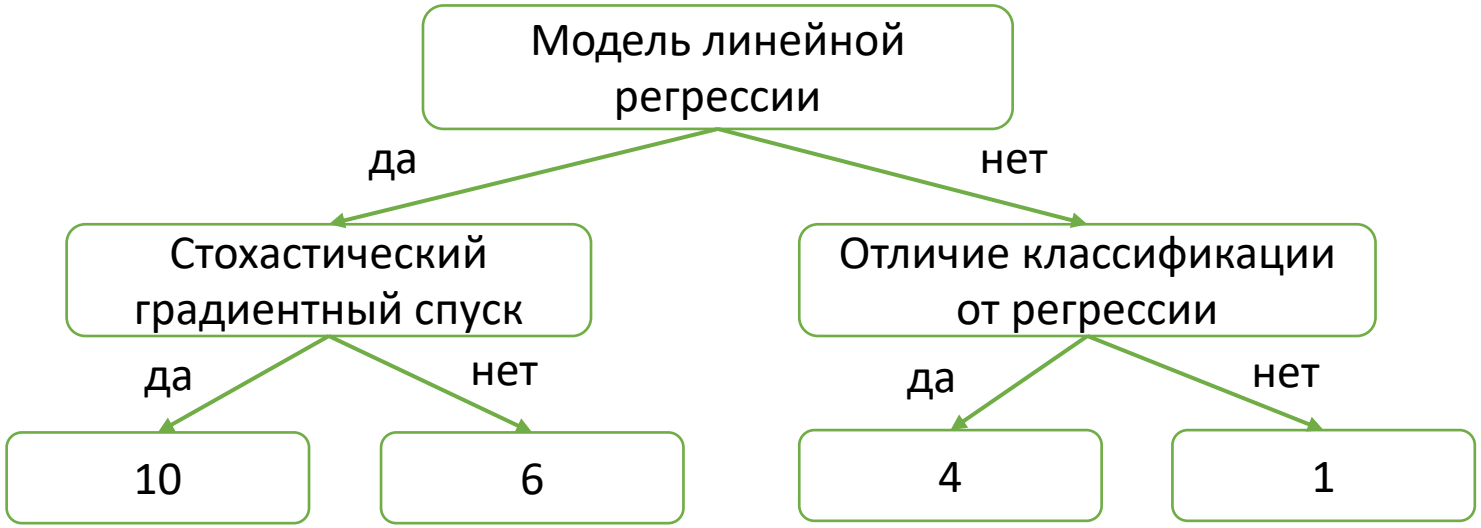


Решающее дерево

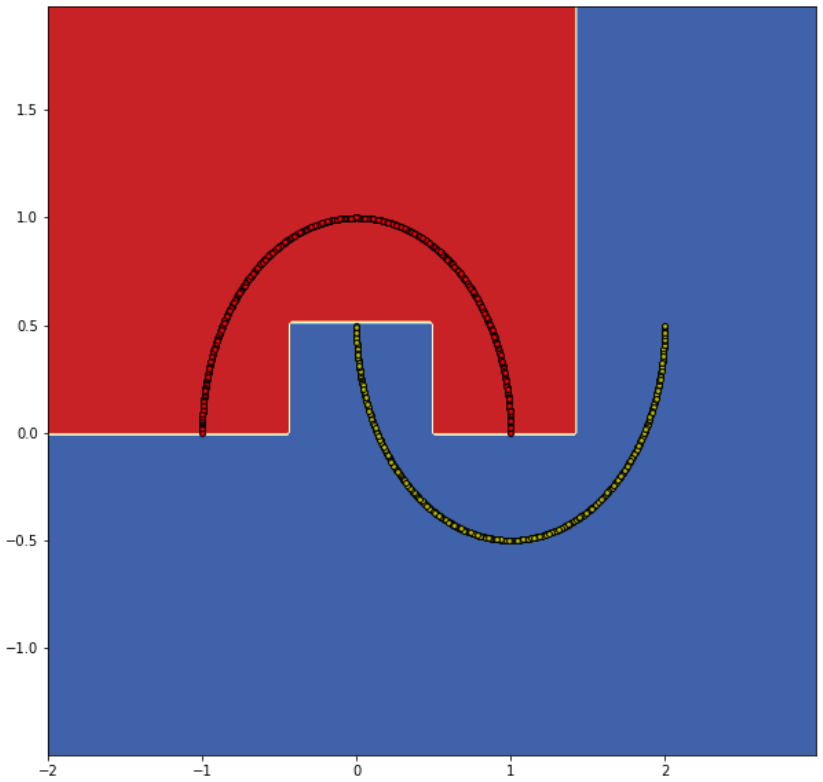


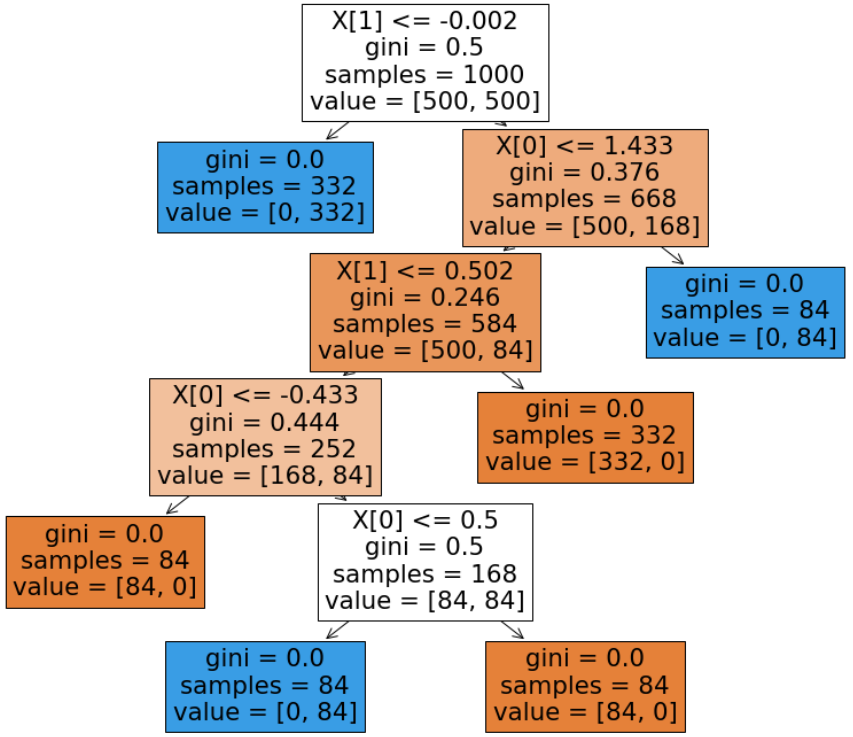
- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

Решающее дерево

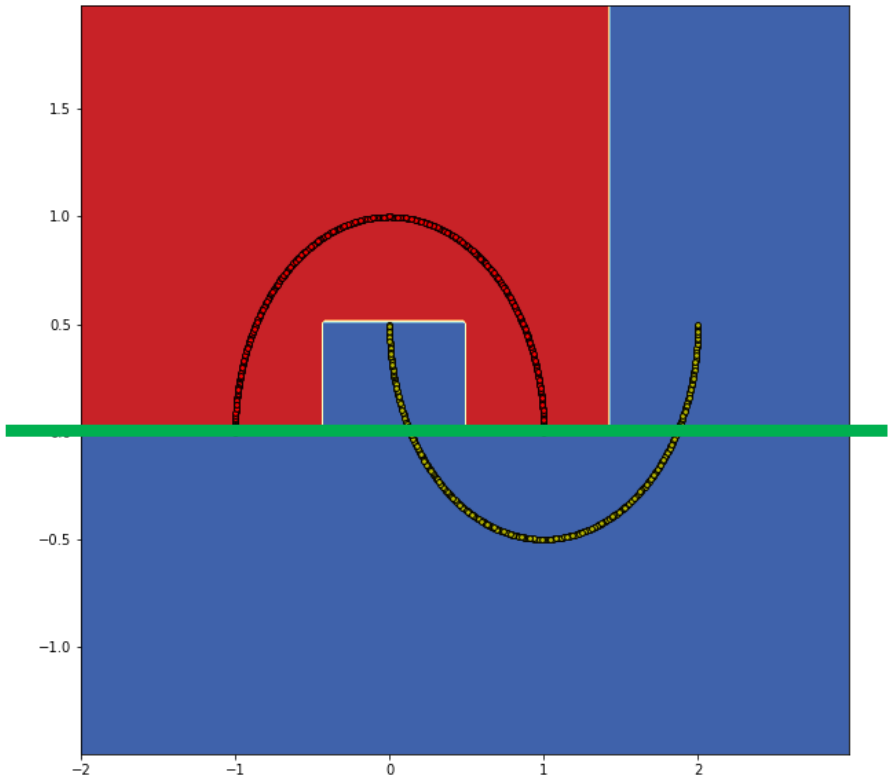


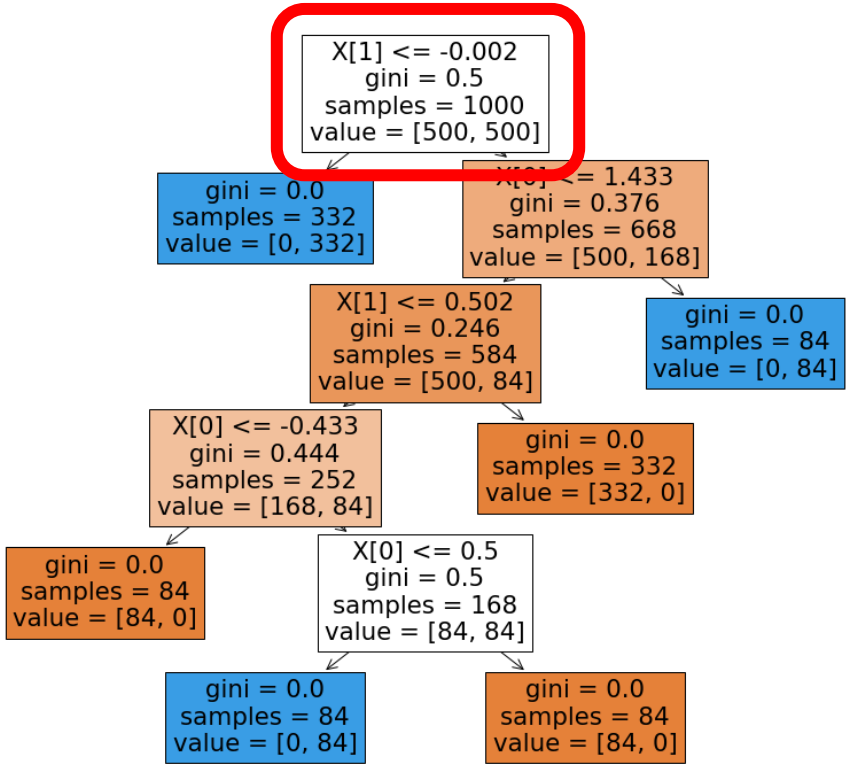
Решающее дерево



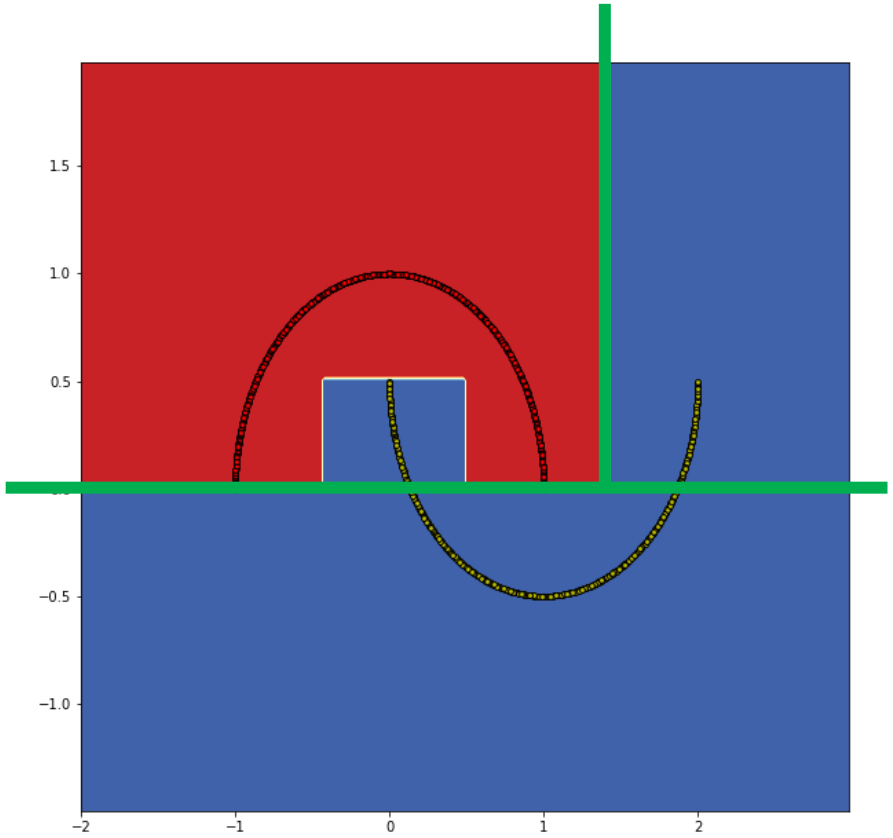


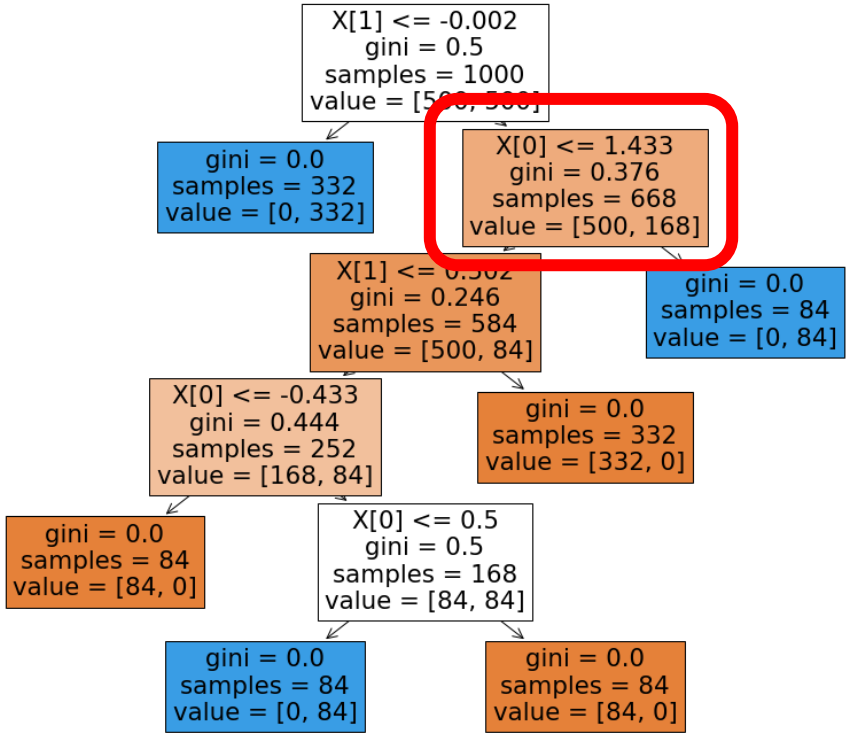
Решающее дерево



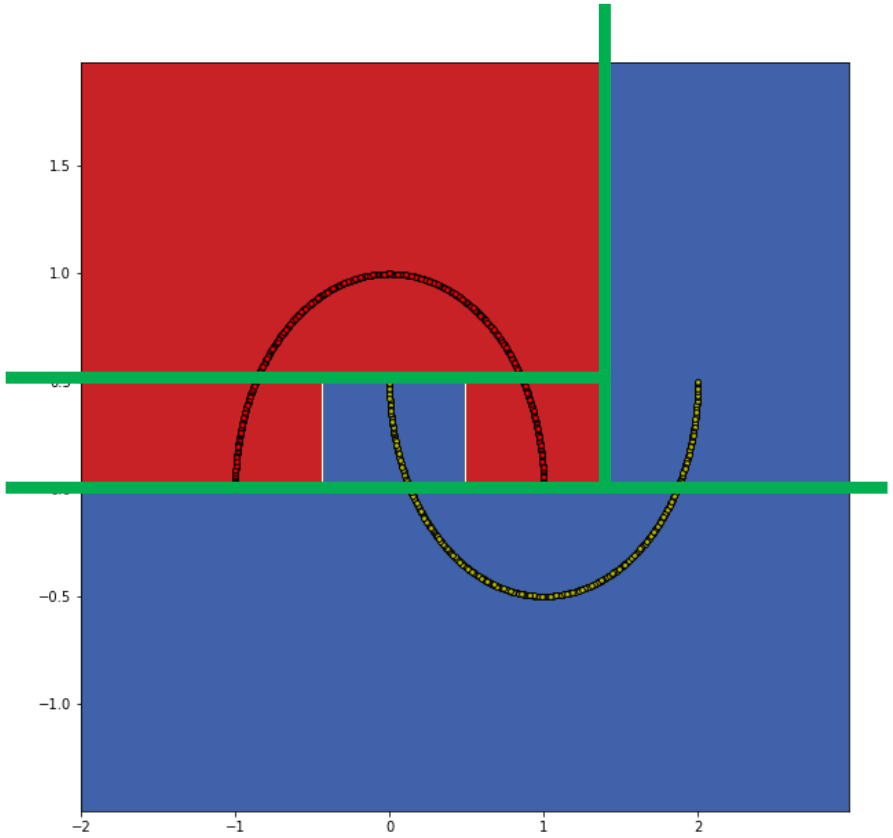


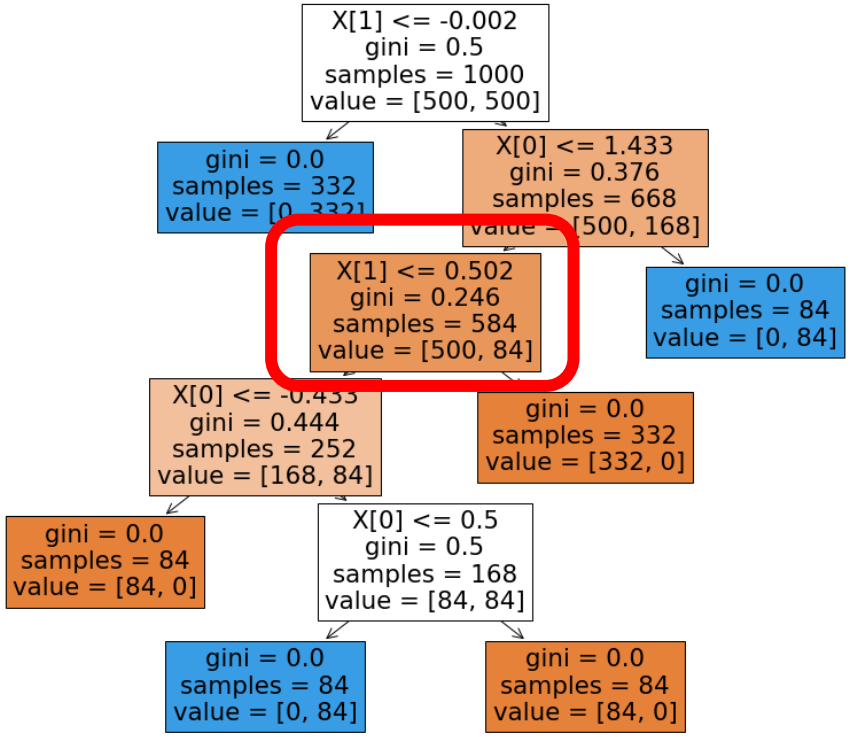
Решающее дерево



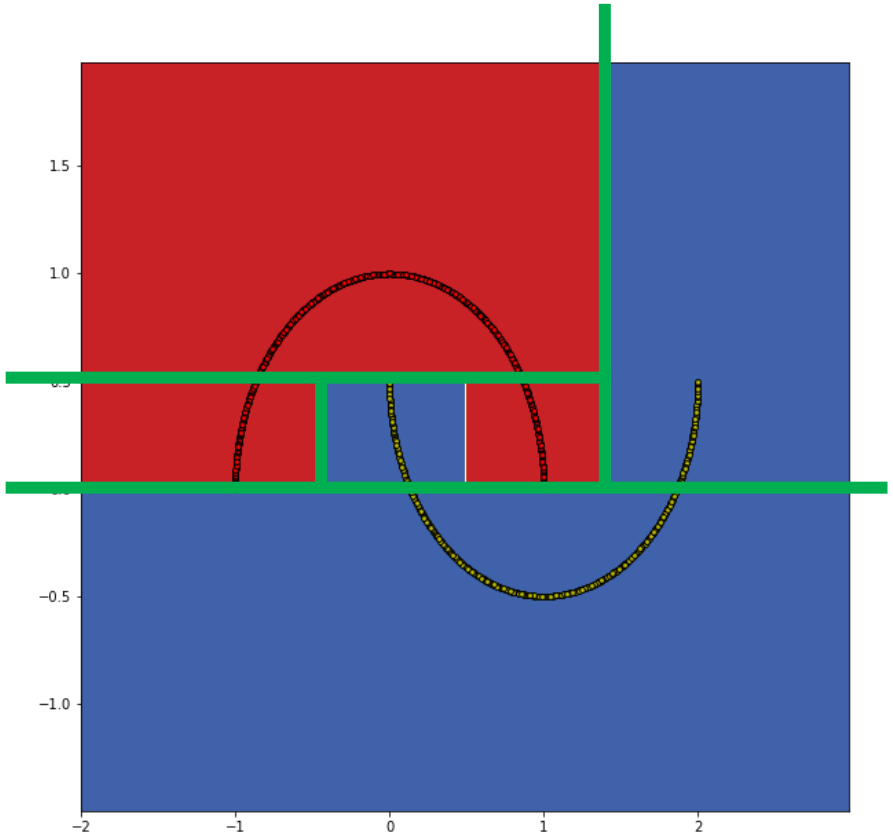


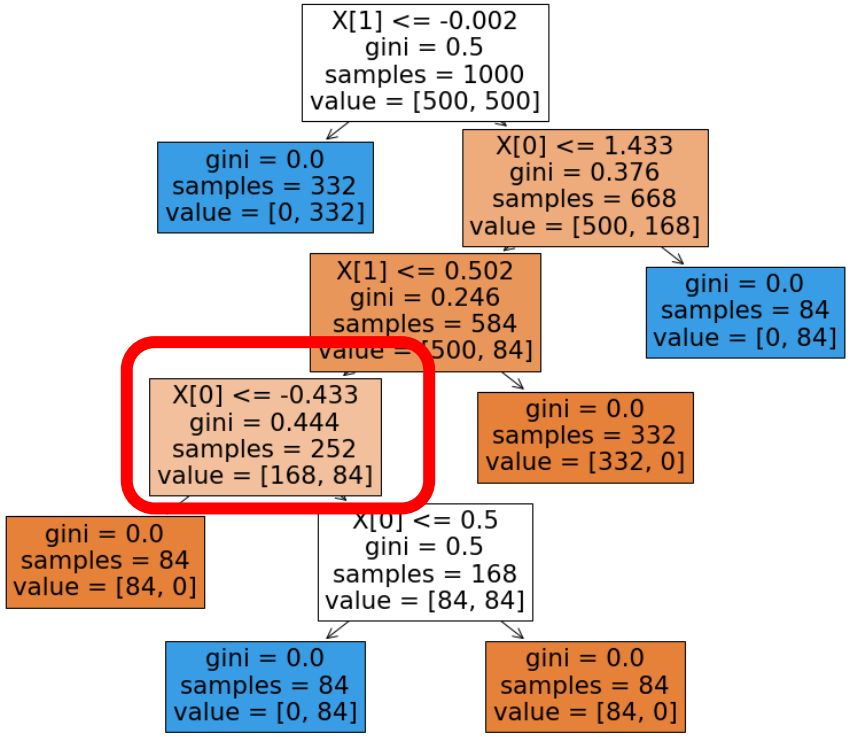
Решающее дерево



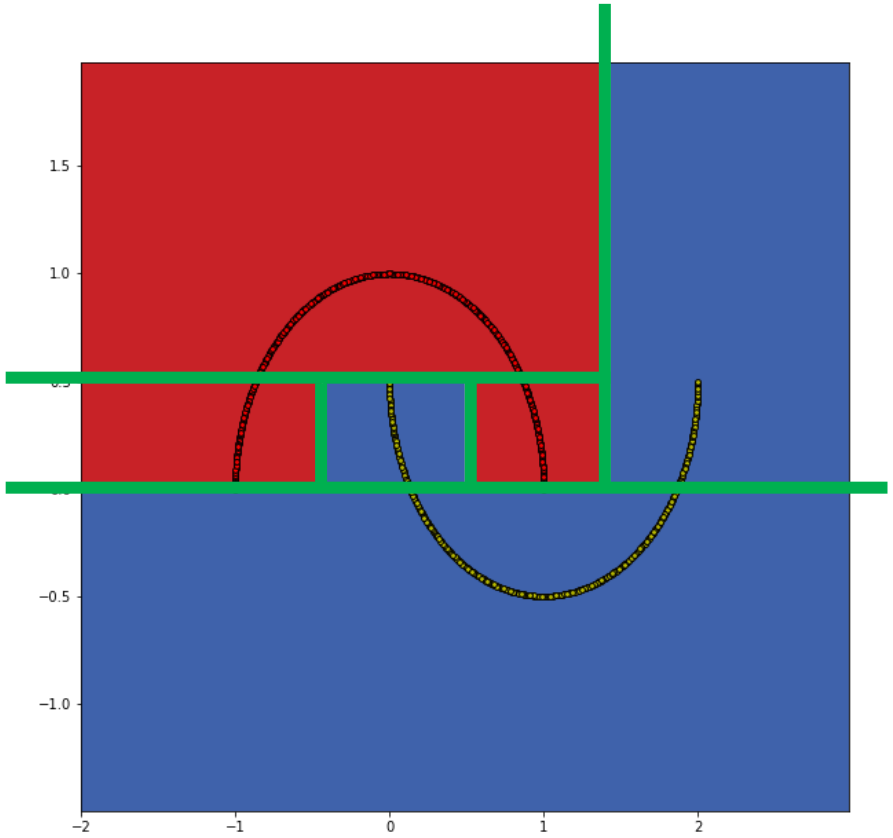


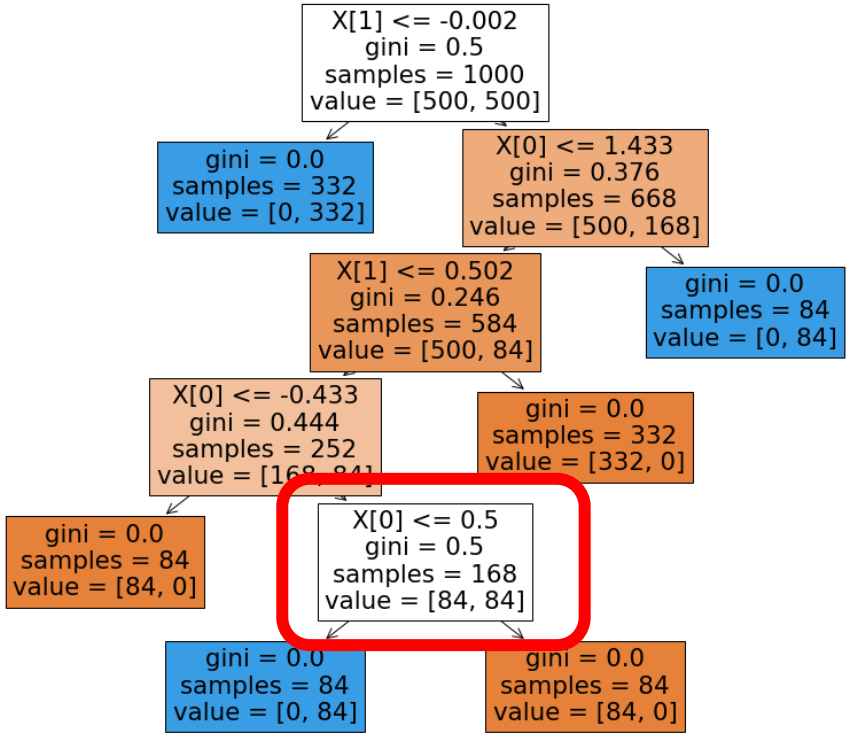
Решающее дерево



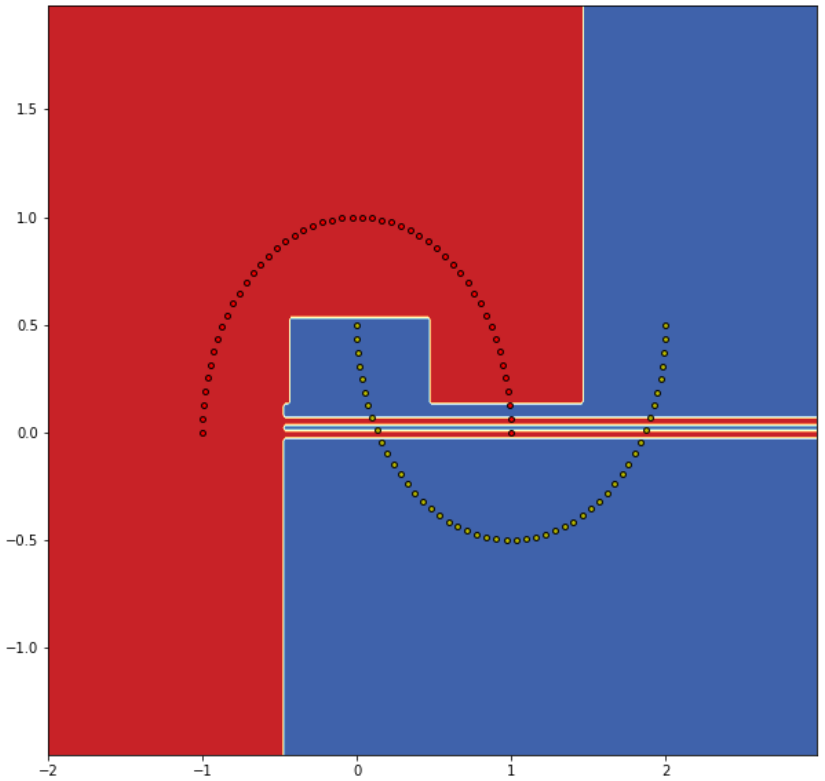


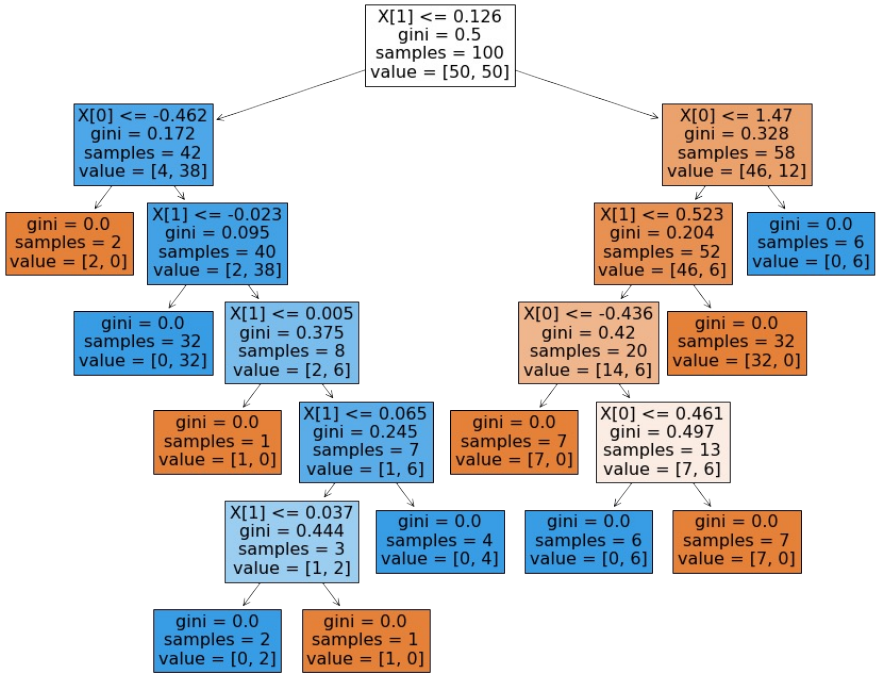
Решающее дерево





Решающее дерево

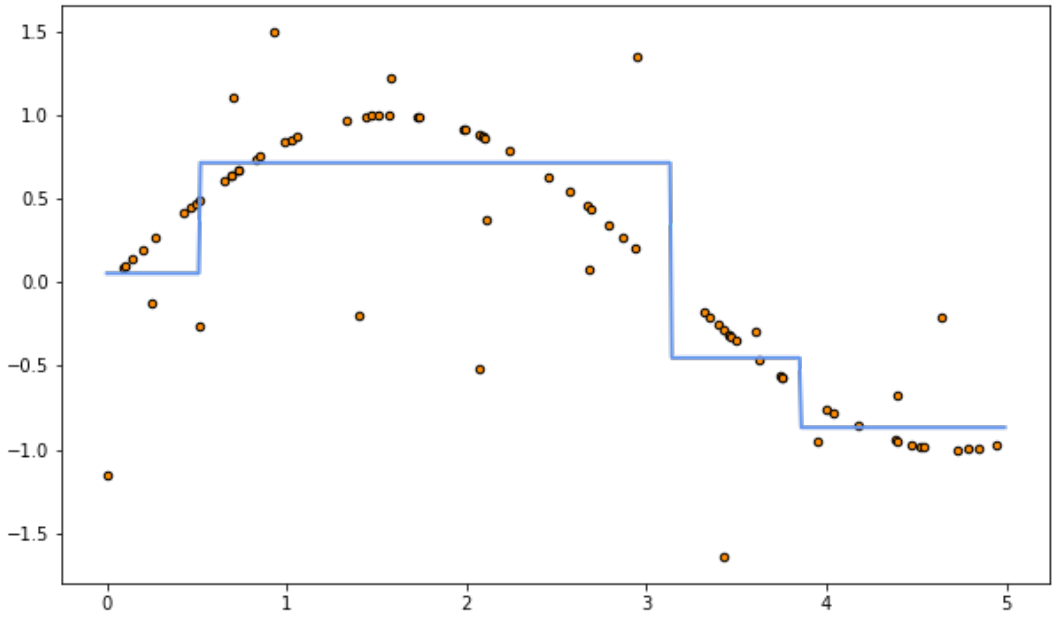


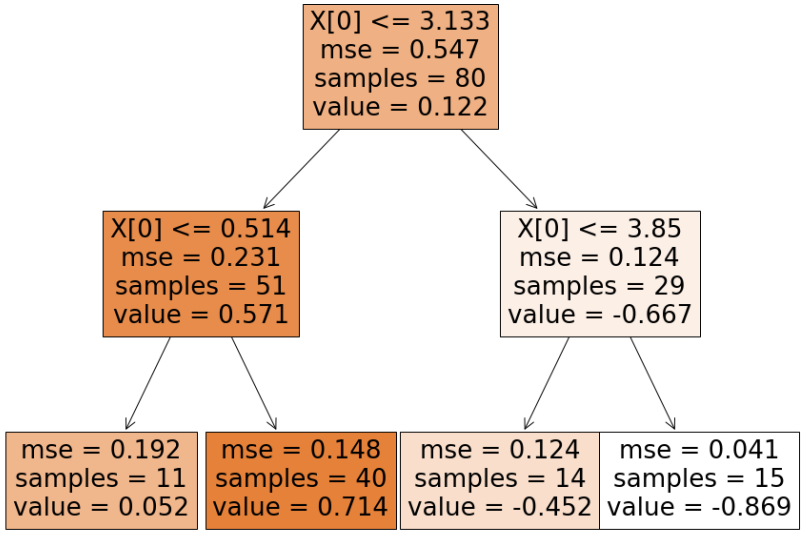


Сложность дерева

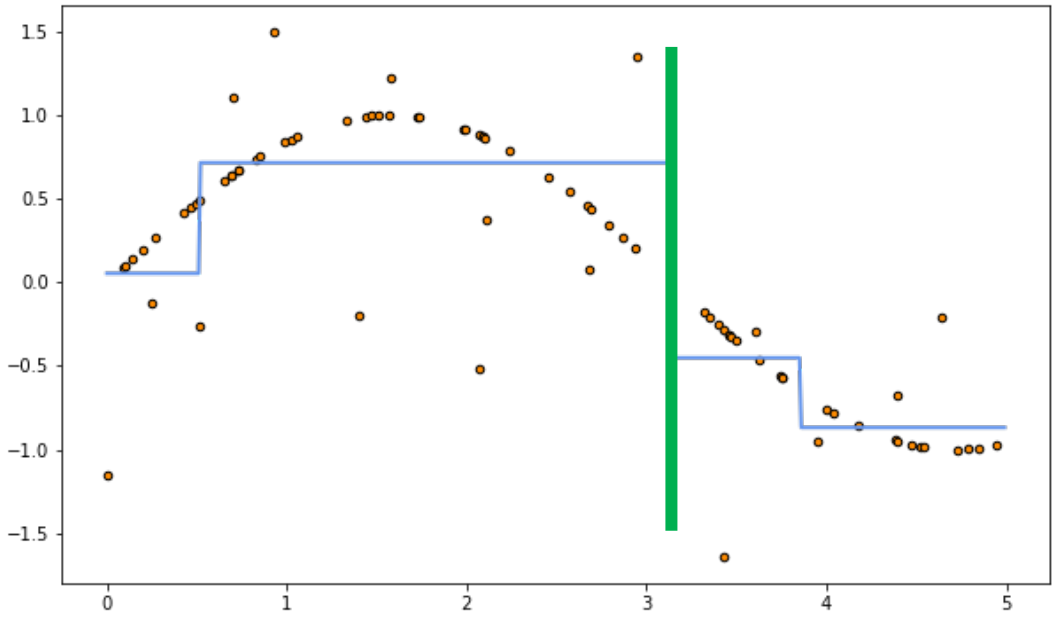
- Решающее дерево можно строить до тех пор, пока каждый лист не будет соответствовать ровно одному объекту
- Деревом можно идеально разделить любую выборку!
- Если только нет объектов с одинаковыми признаками, но разными ответами

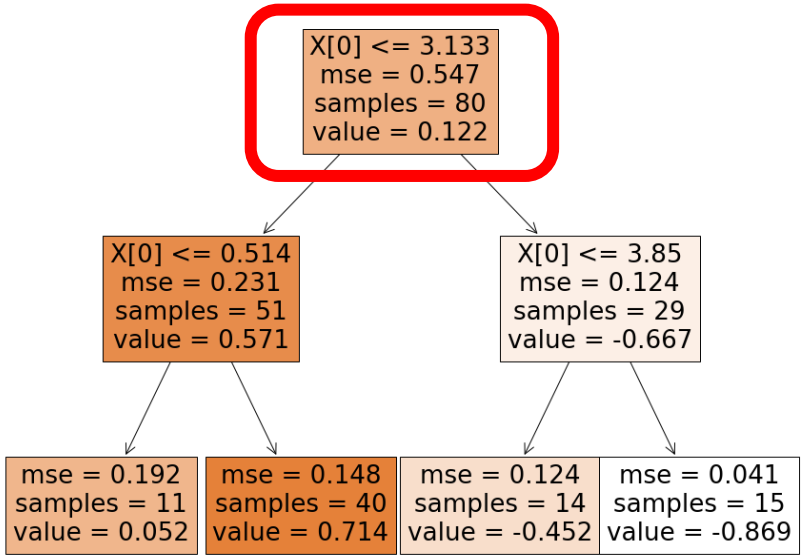
Решающее дерево для регрессии



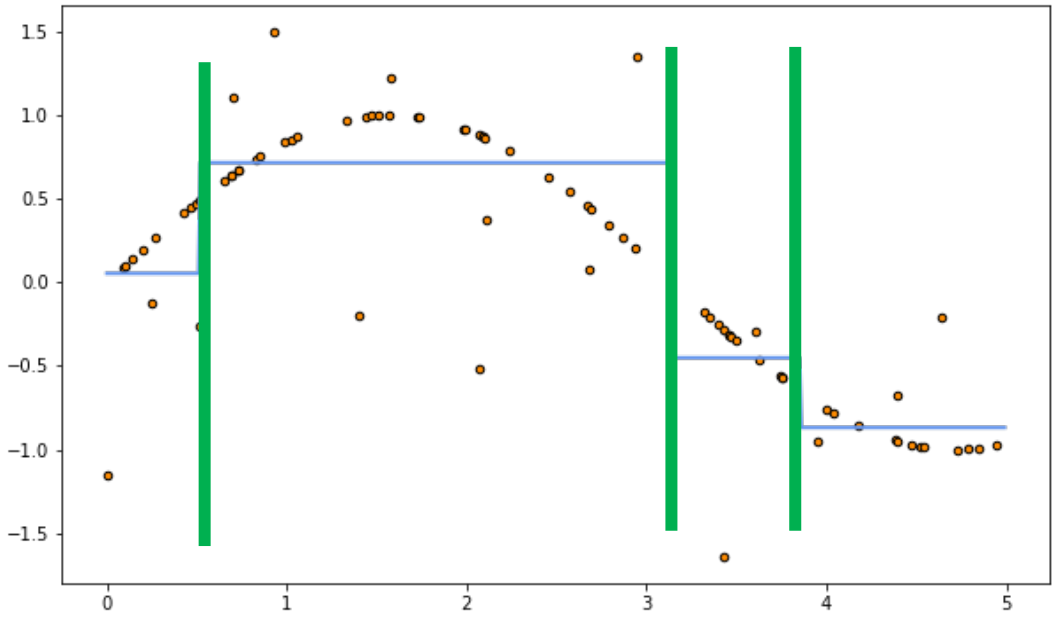


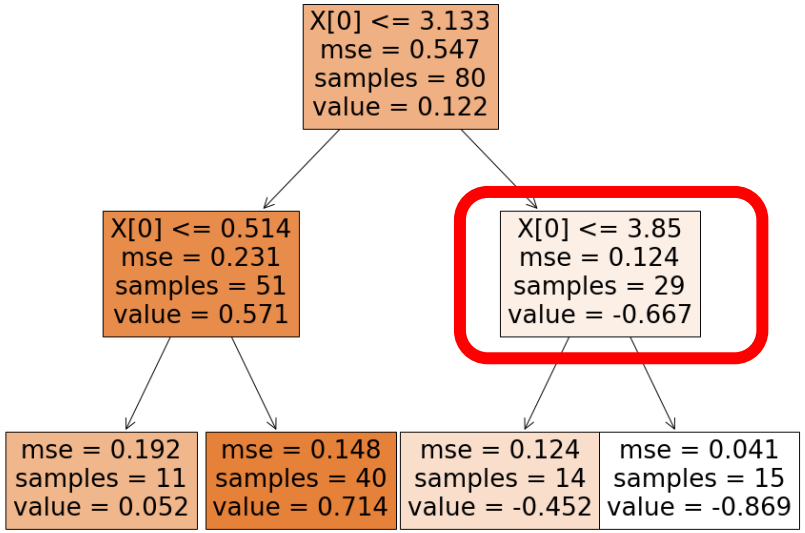
Решающее дерево для регрессии



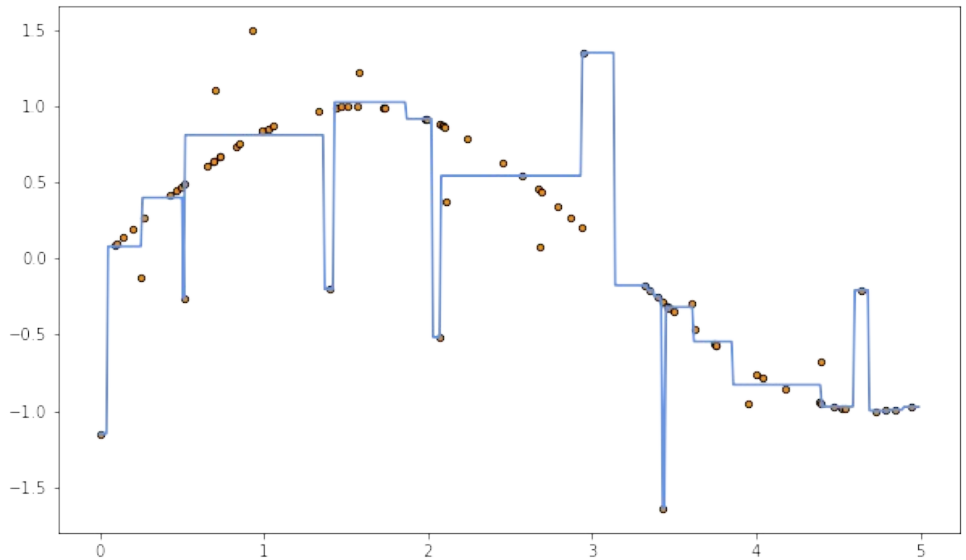


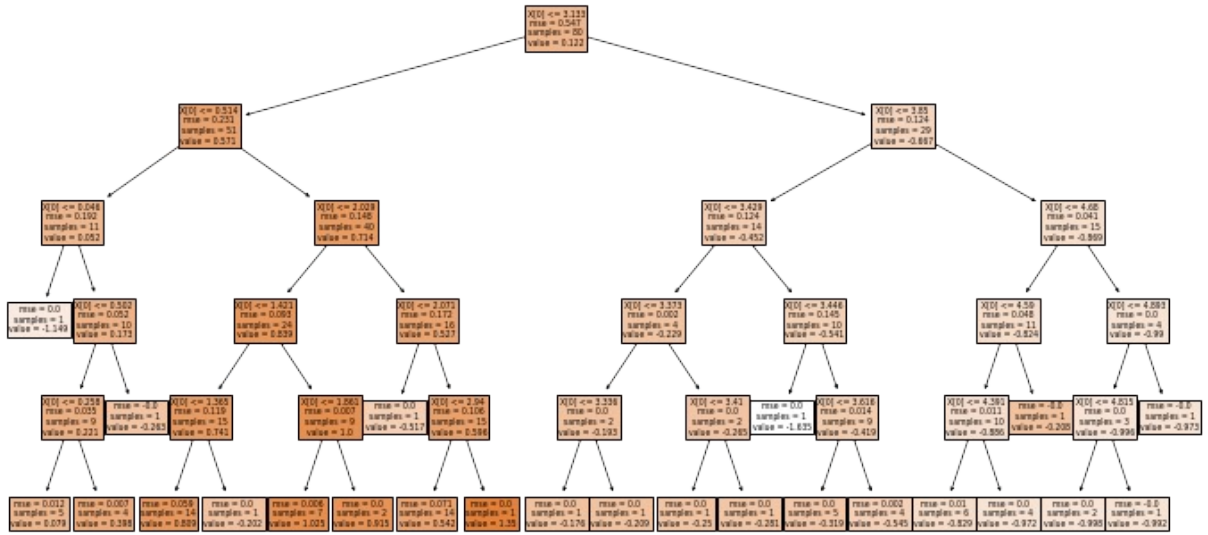
Решающее дерево для регрессии



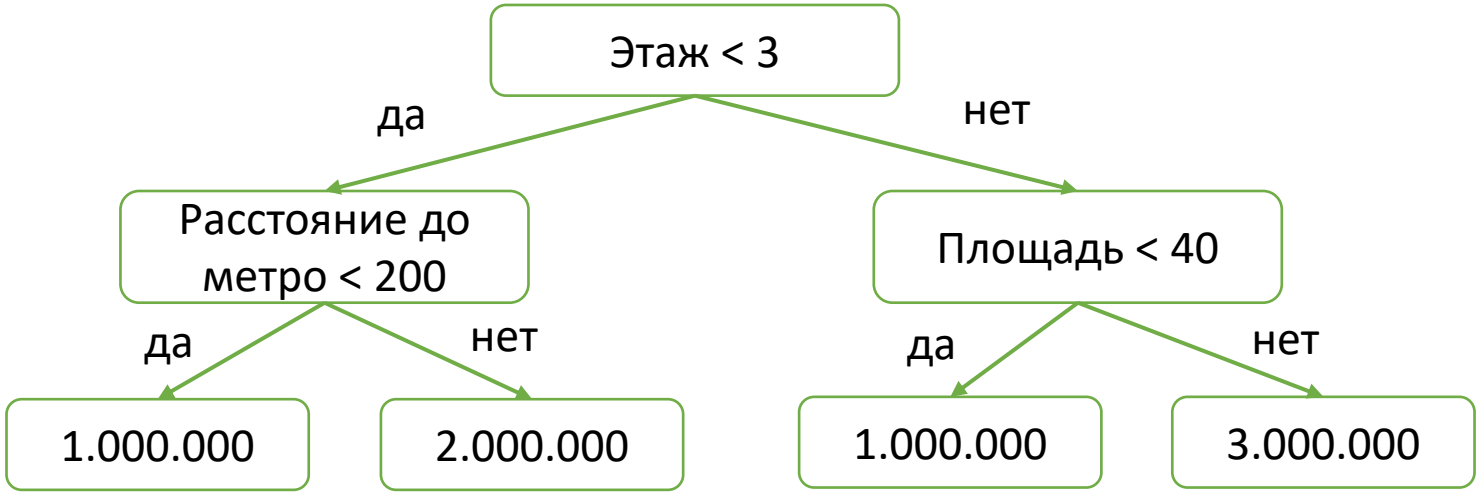


Решающее дерево для регрессии





Решающее дерево



- Внутренние вершины: предикаты $[x_j < t]$
- Листья: прогнозы $c \in \mathbb{Y}$

Предикаты

- Порог на признак $[x_j < t]$ — не единственный вариант
- Предикат с линейной моделью: $[\langle w, x \rangle < t]$
- Предикат с метрикой: $[\rho(x, x_0) < t]$
- И много других вариантов
- Но даже с простейшим предикатом можно строить очень сложные модели

Прогнозы в листьях

- Наш выбор: константные прогнозы $c_v \in \mathbb{Y}$
- Регрессия:

$$c_v = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} y_i$$

- Классификация:

$$c_v = \arg\max_{k \in \mathbb{Y}} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

Прогнозы в листьях

- Наш выбор: константные прогнозы $c_v \in \mathbb{Y}$
- Классификация и вероятности классов:

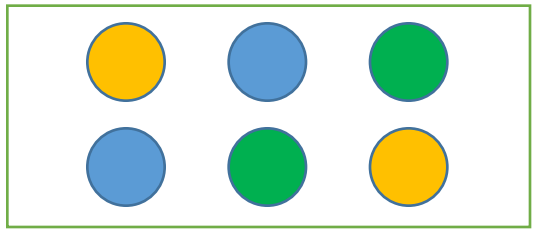
$$c_{vk} = \frac{1}{|R_v|} \sum_{(x_i, y_i) \in R_v} [y_i = k]$$

Как выбирать предикаты

Жадное построение

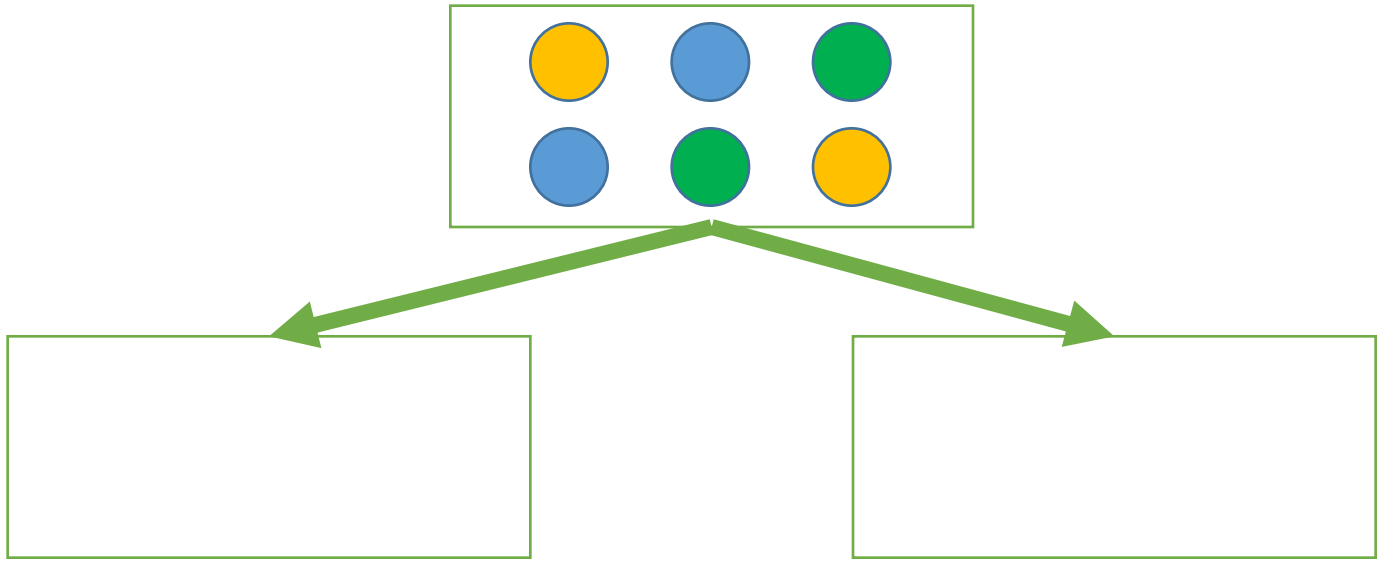
- Разберёмся на примере
- Начнём с задачи классификации

Жадное построение

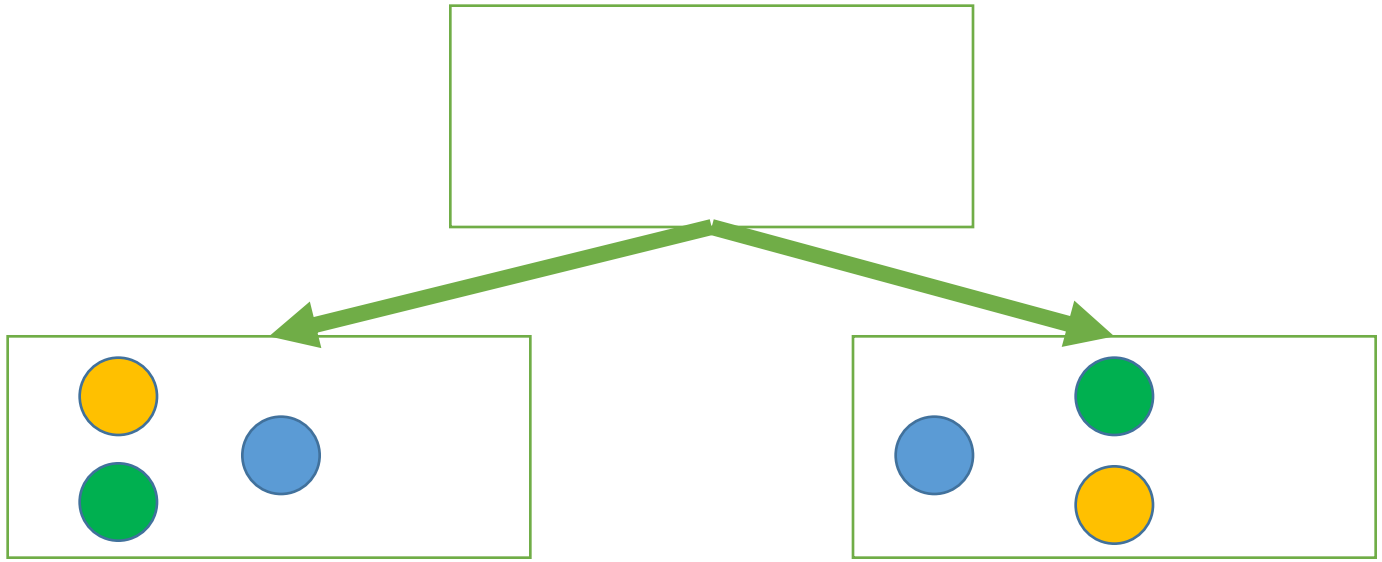


- Как разбить вершину?

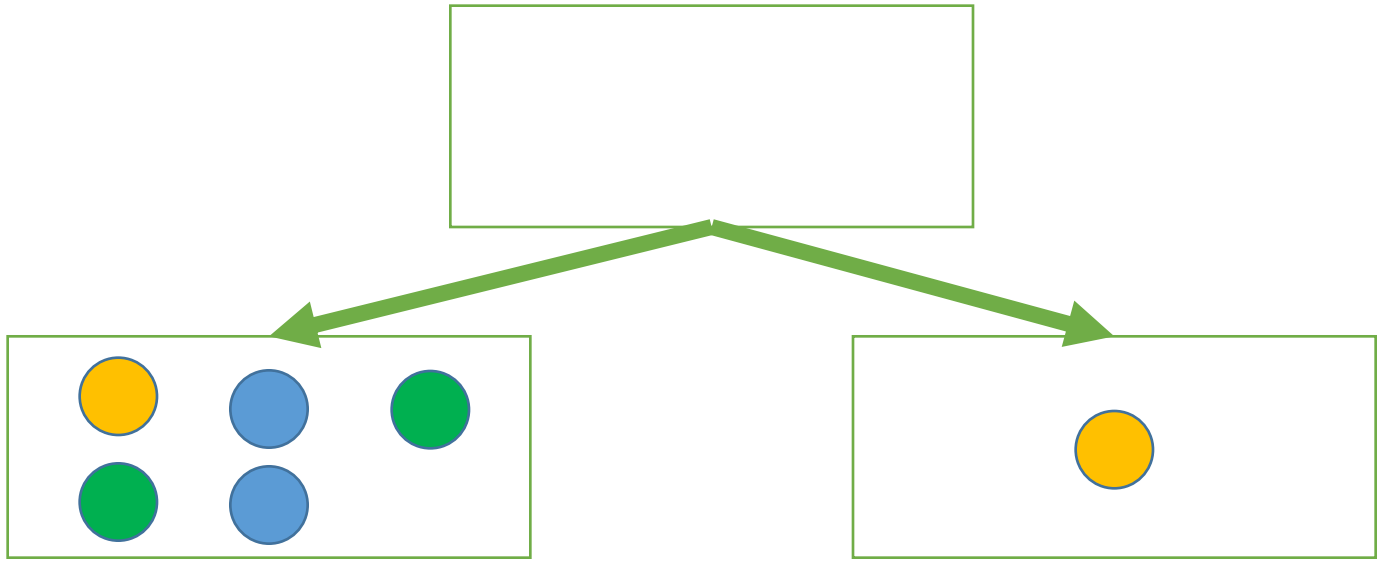
Жадное построение



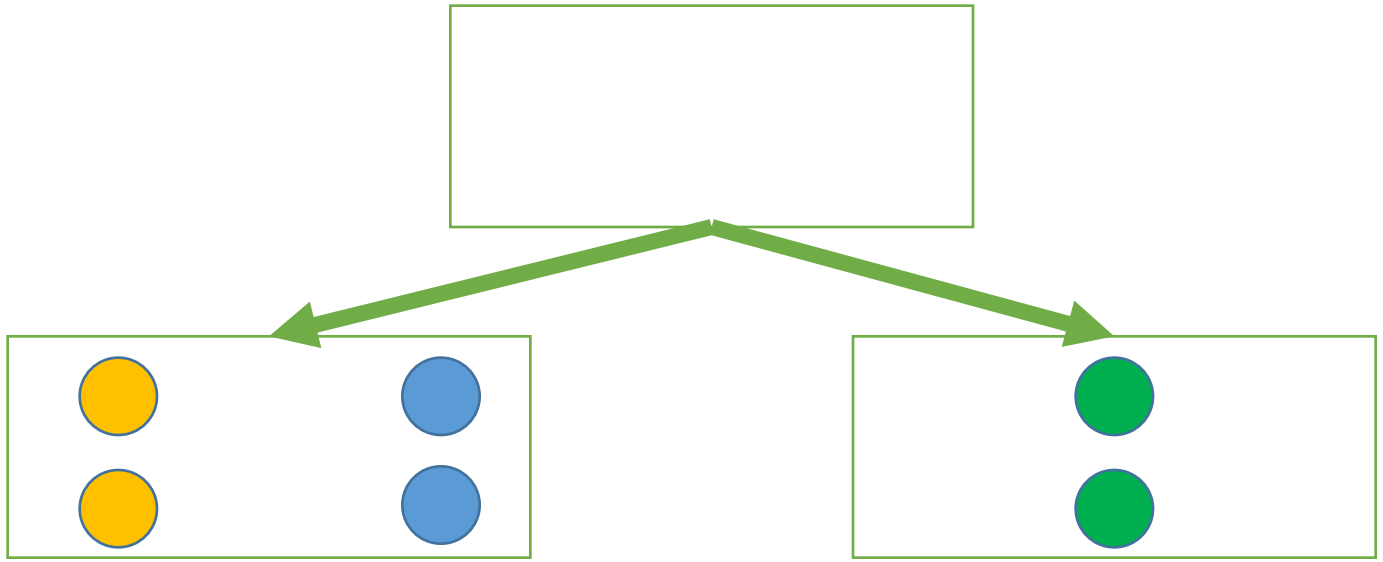
Жадное построение



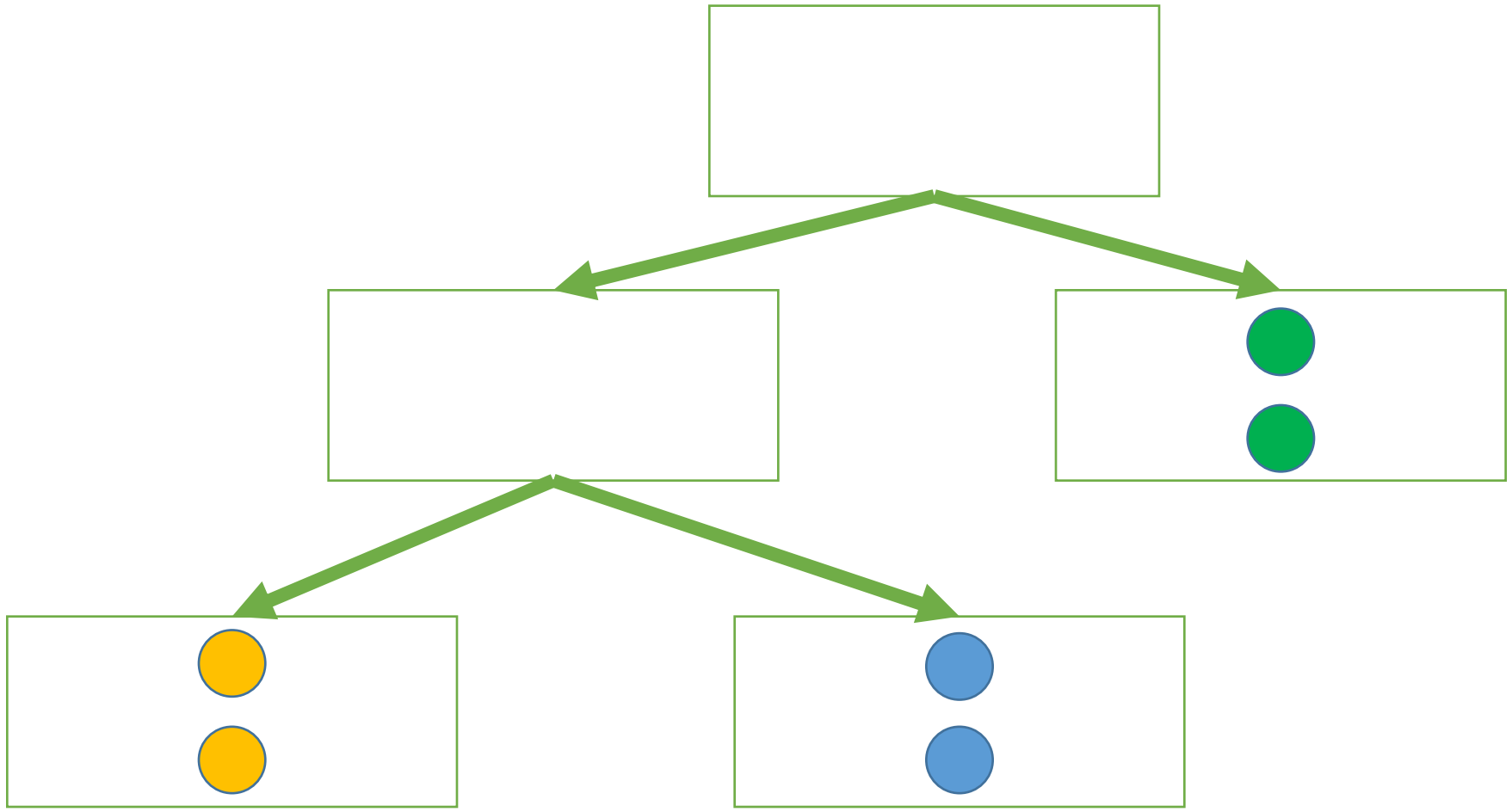
Жадное построение



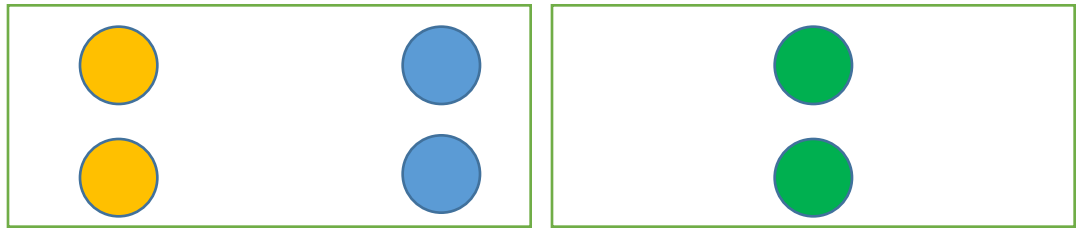
Жадное построение



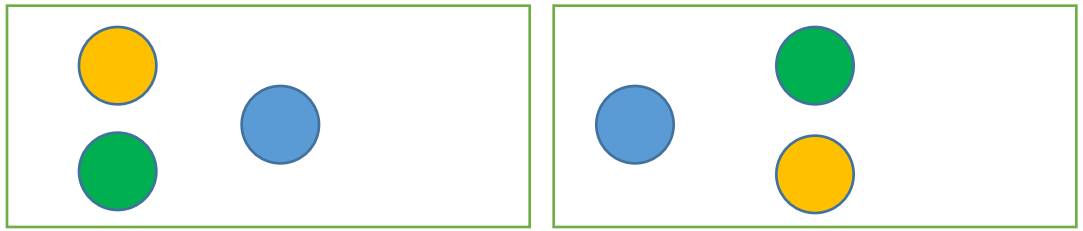
Жадное построение



Как сравнить разбиения?

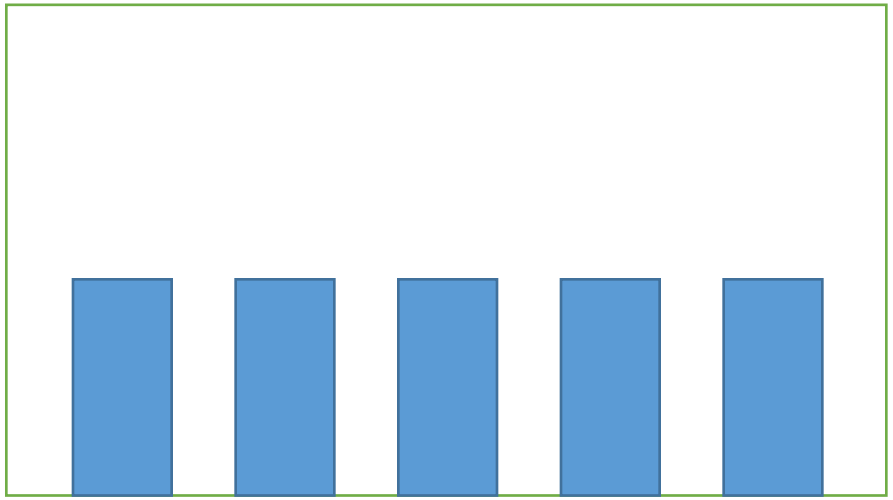


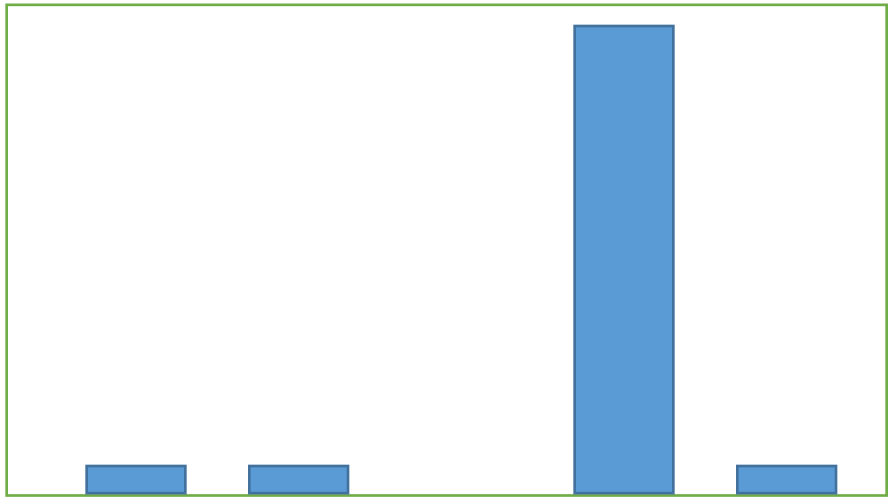
или



Энтропия

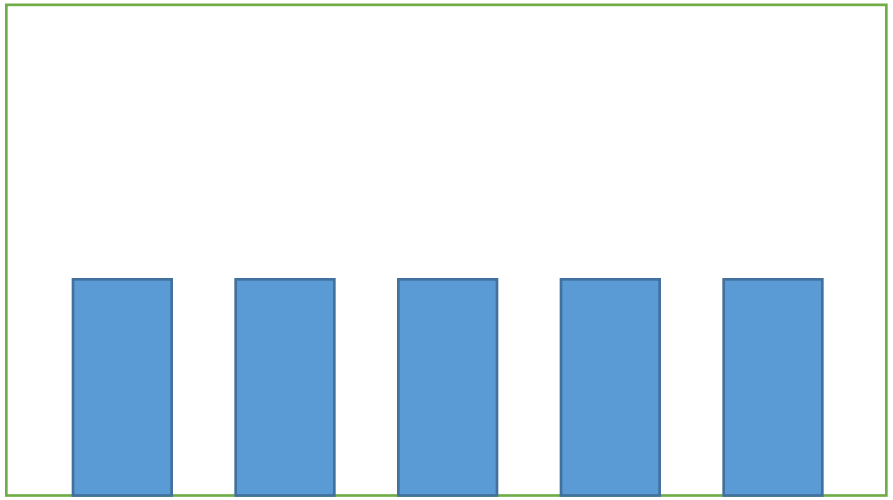
- Мера неопределённости распределения



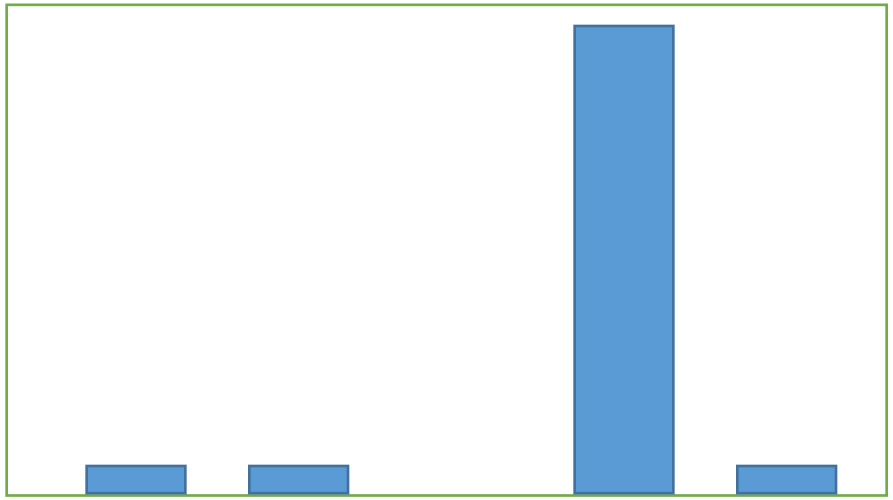


Энтропия

- Мера неопределённости распределения



Высокая энтропия



Низкая энтропия

Энтропия

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

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

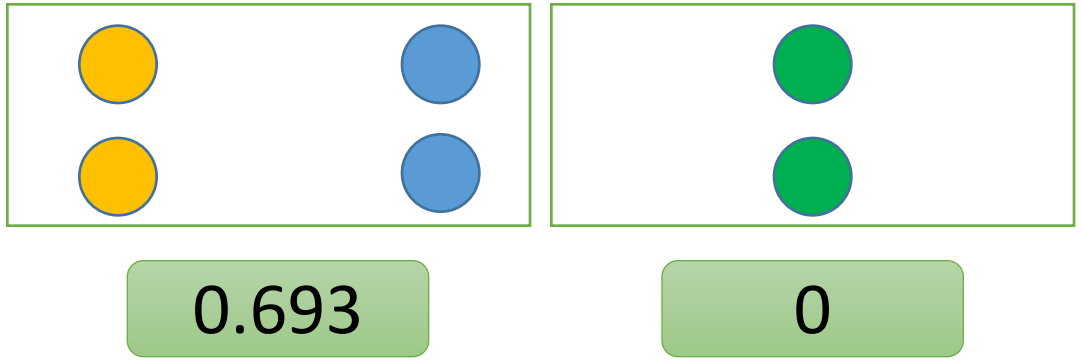
Энтропия

- $(0.2, 0.2, 0.2, 0.2, 0.2)$
- $H = 1.60944 \ldots$

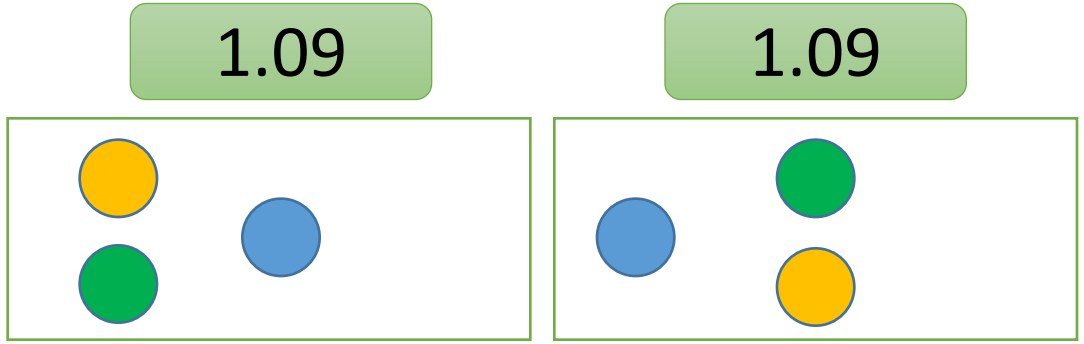
- $(0.9, 0.05, 0.05, 0, 0)$
- $H = 0.394398 \ldots$

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

Как сравнить разбиения?



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



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

Энтропия

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

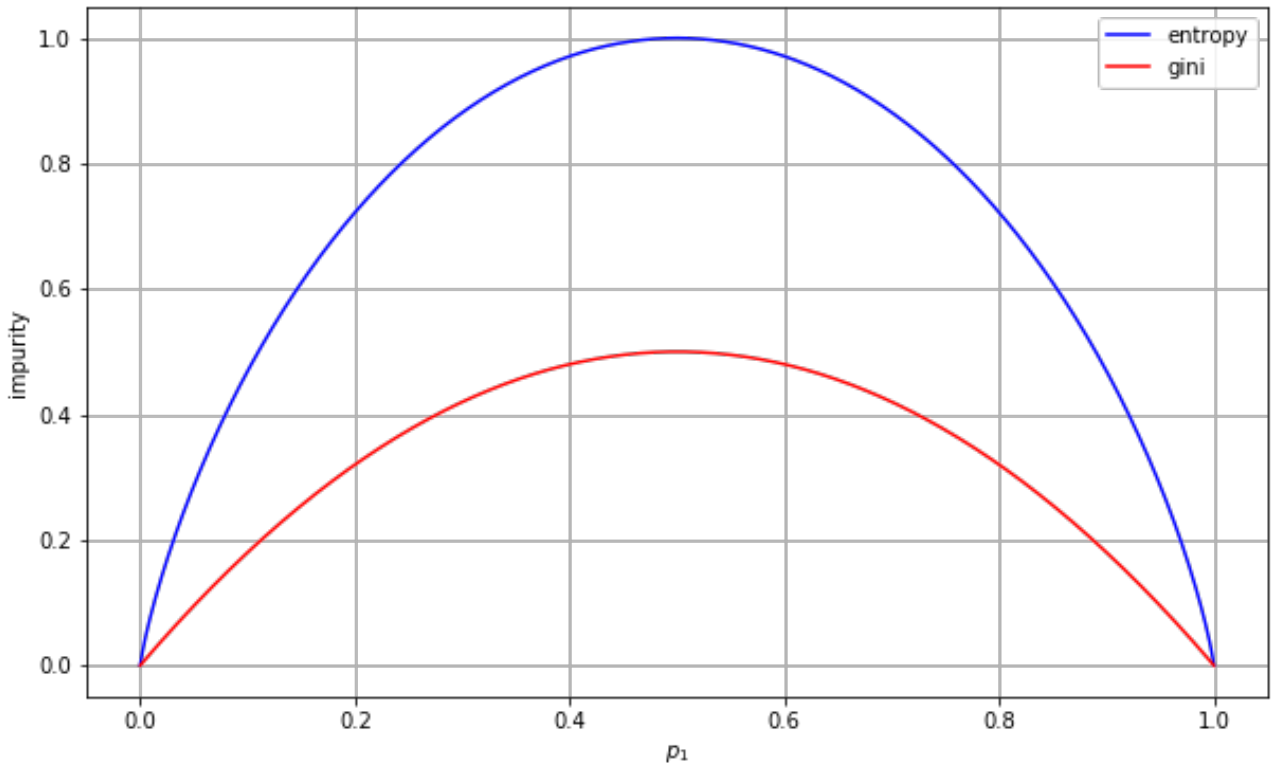
- Характеристика «хаотичности» вершины
- **Impurity**

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

Критерии качества вершины

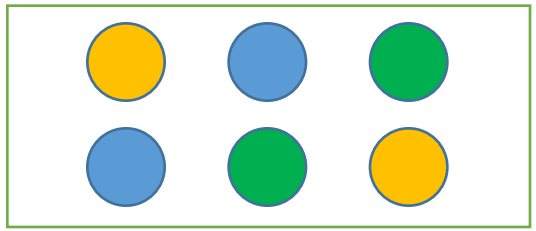


Как выбирать предикаты

Жадное построение

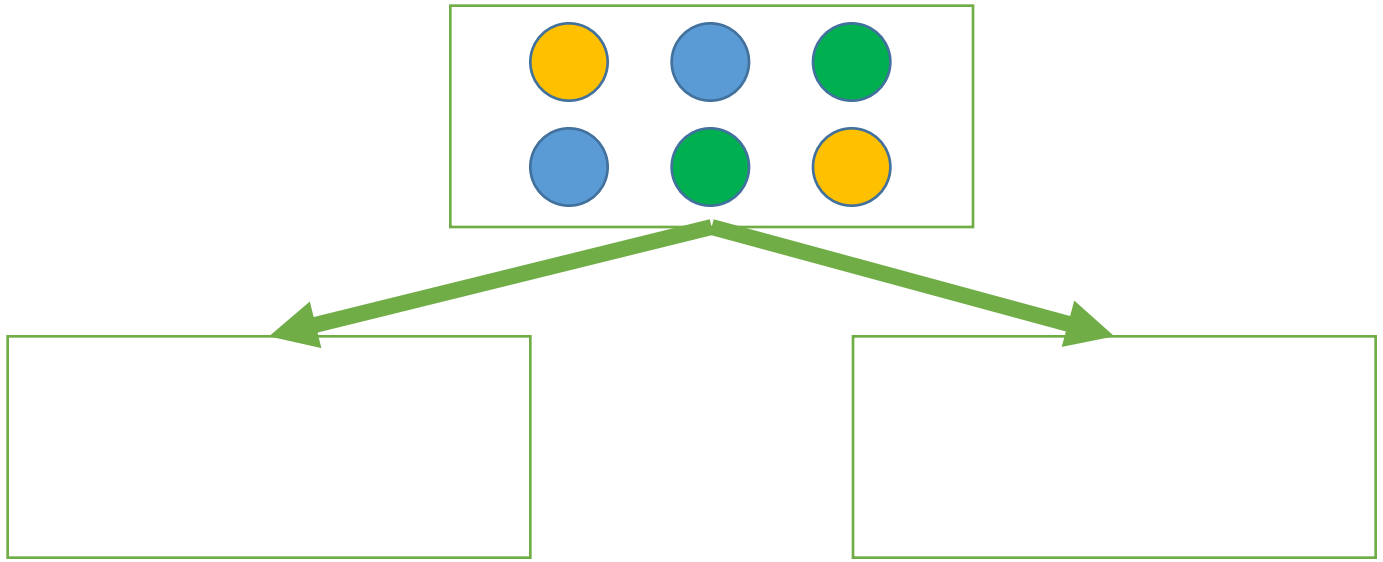
- Разберёмся на примере
- Начнём с задачи классификации

Жадное построение



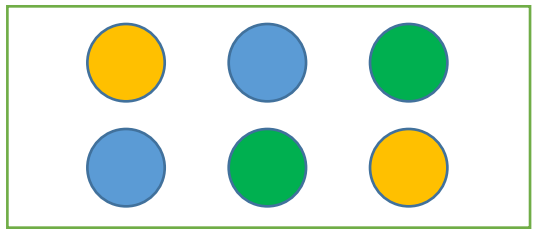
- Как разбить вершину?

Жадное построение

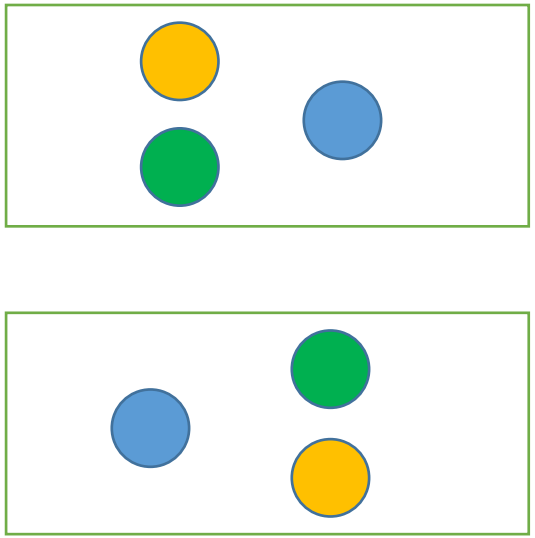


Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

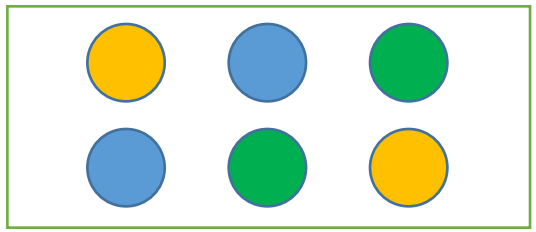


против

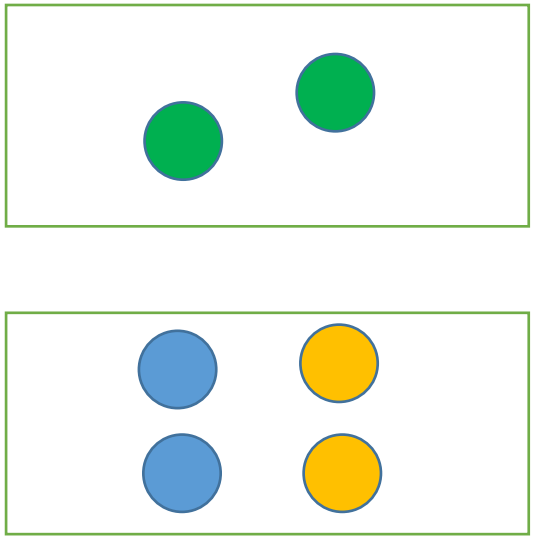


Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!



против



Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

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

Критерий информативности

- Как понять, какой предикат лучше?
- Сравнить хаотичность в исходной вершине и в двух дочерних!

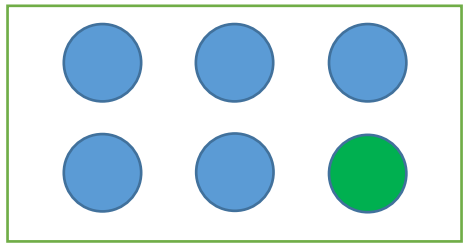
$$Q(R, j, t) = H(R) - H(R_\ell) - H(R_r) \to \max_{j,t}$$

- Или так:

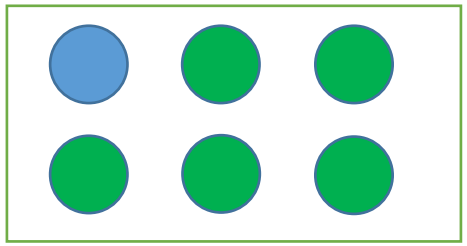
$$Q(R, j, t) = H(R_\ell) + H(R_r) \to \min_{j,t}$$

- (у этих формул есть проблемы!)

Как сравнить разбиения?

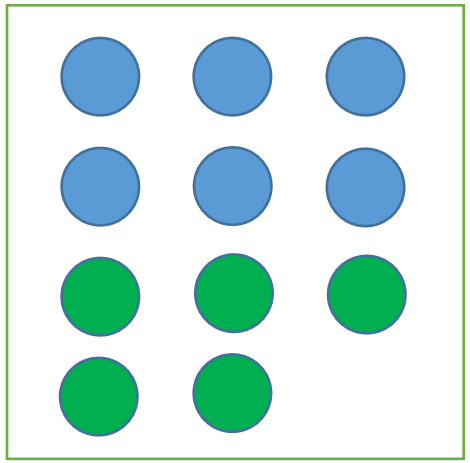


0.65

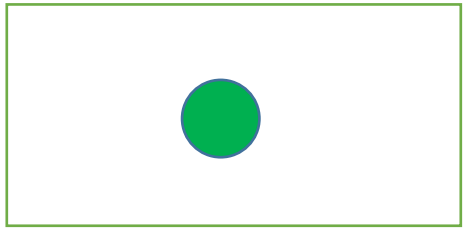


0.65

- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.65 + 0.65 = 1.3$



0.994



0

- $(6/11, 5/11)$ и $(0, 1)$
- $0.994 + 0 = 0.994$

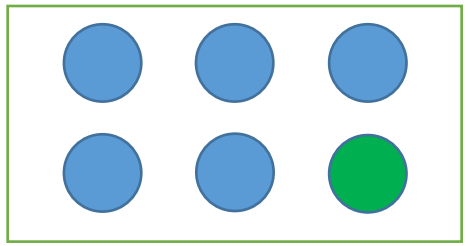
Критерий информативности

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

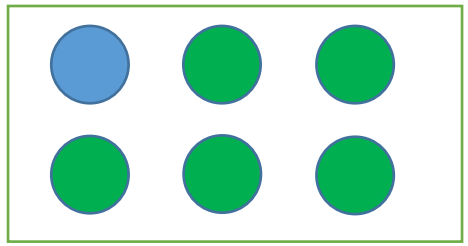
- Или так:

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

Как сравнить разбиения?

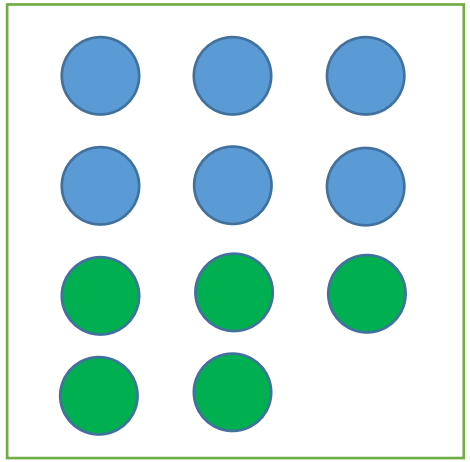


0.65

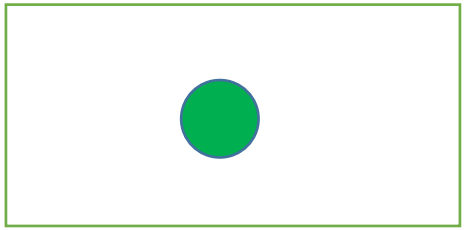


0.65

- $(5/6, 1/6)$ и $(1/6, 5/6)$
- $0.5 * 0.65 + 0.5 * 0.65 = 0.65$



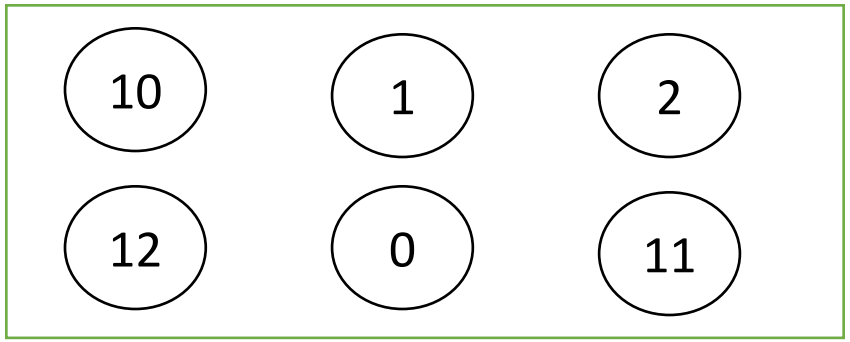
0.994



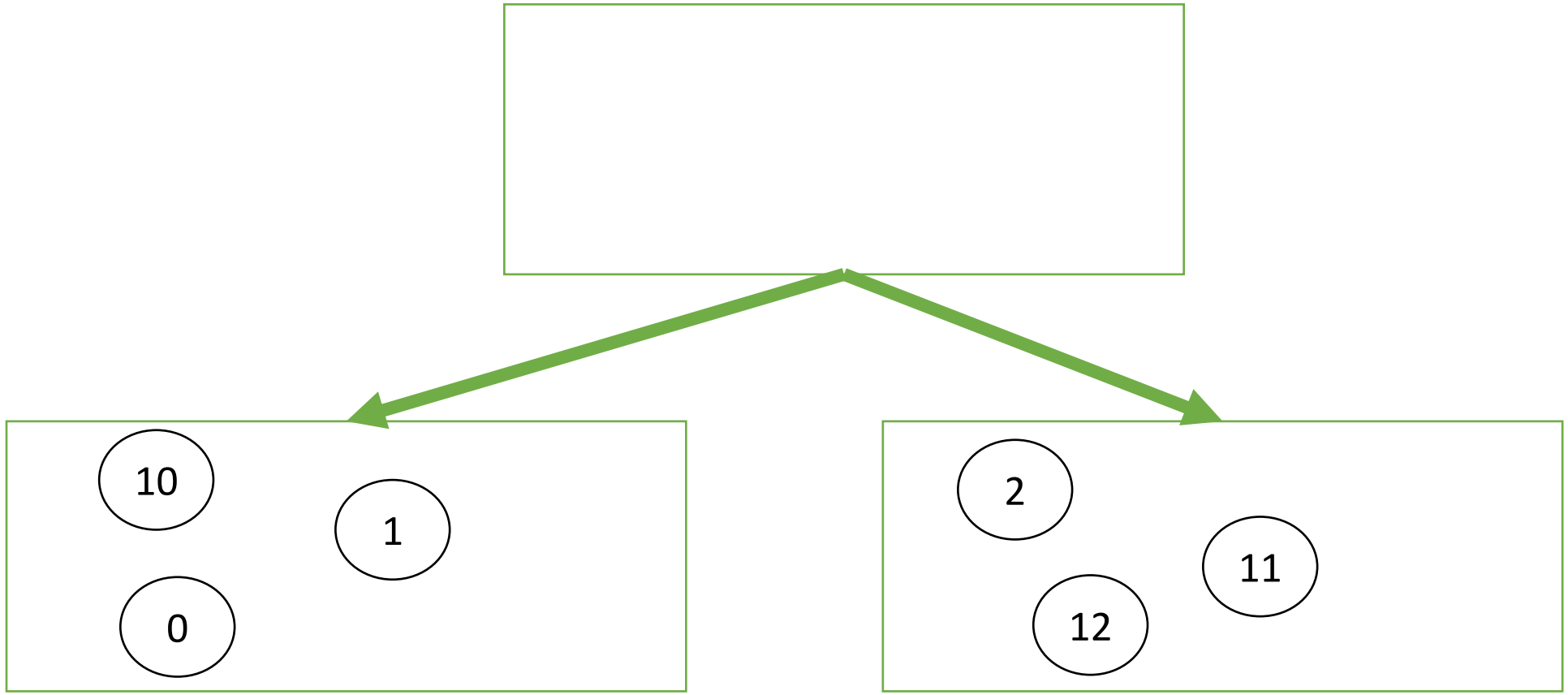
0

- $(6/11, 5/11)$ и $(0, 1)$
- $\frac{11}{12} * 0.994 + \frac{1}{12} * 0 = 0.911$

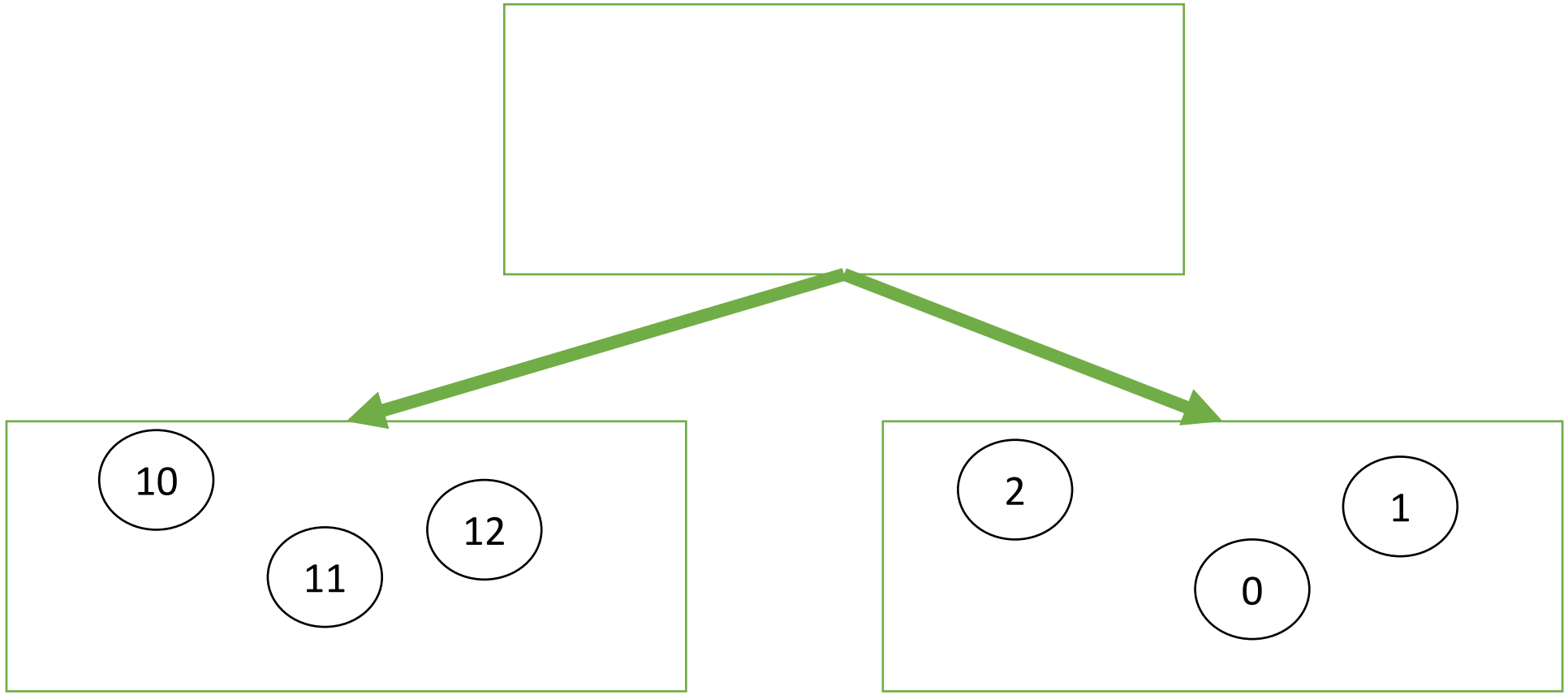
А для регрессии?



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

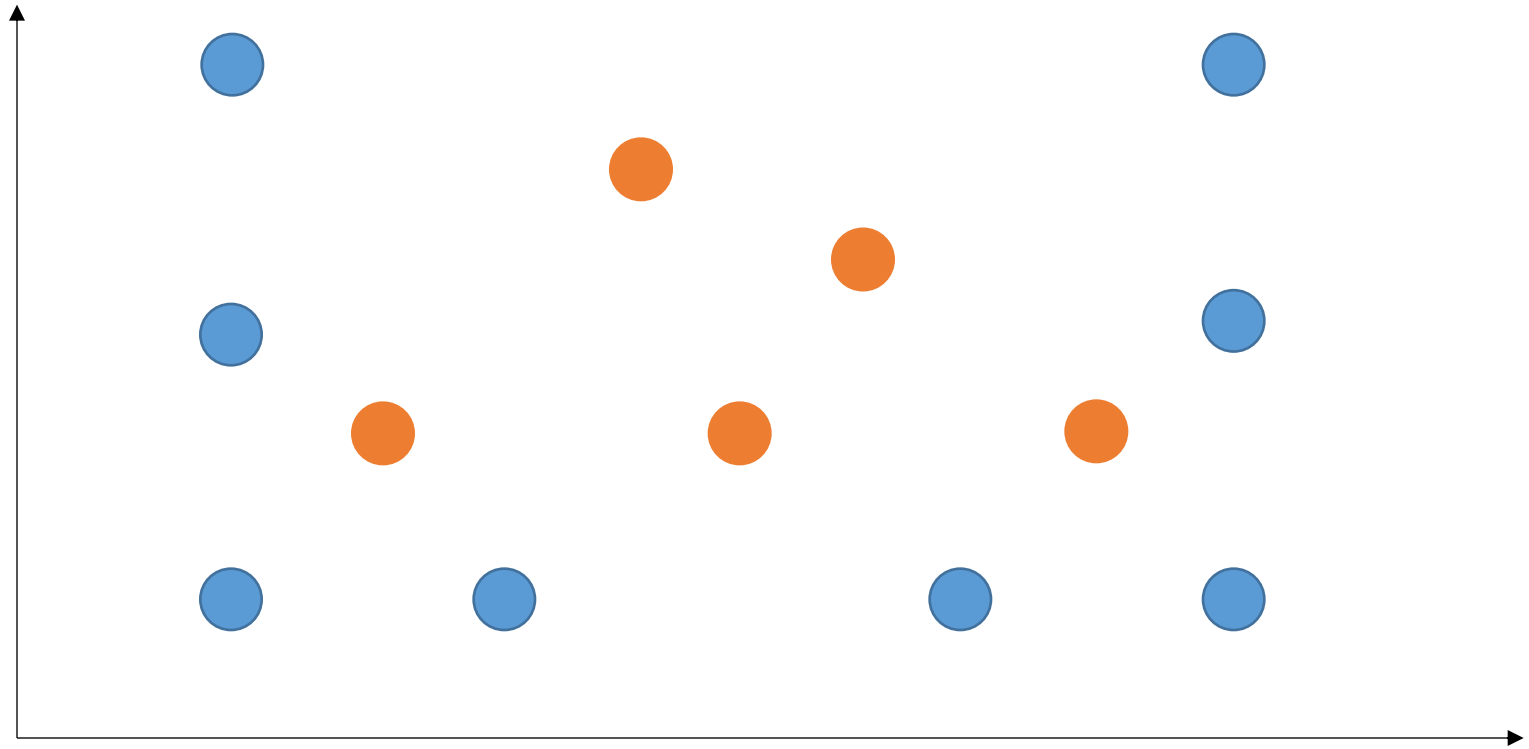
1. Поместить в корень всю выбору: $R_1 = X$
2. Запустить построение из корня: $\mathrm{SplitNode}(1, R_1)$

Жадный алгоритм

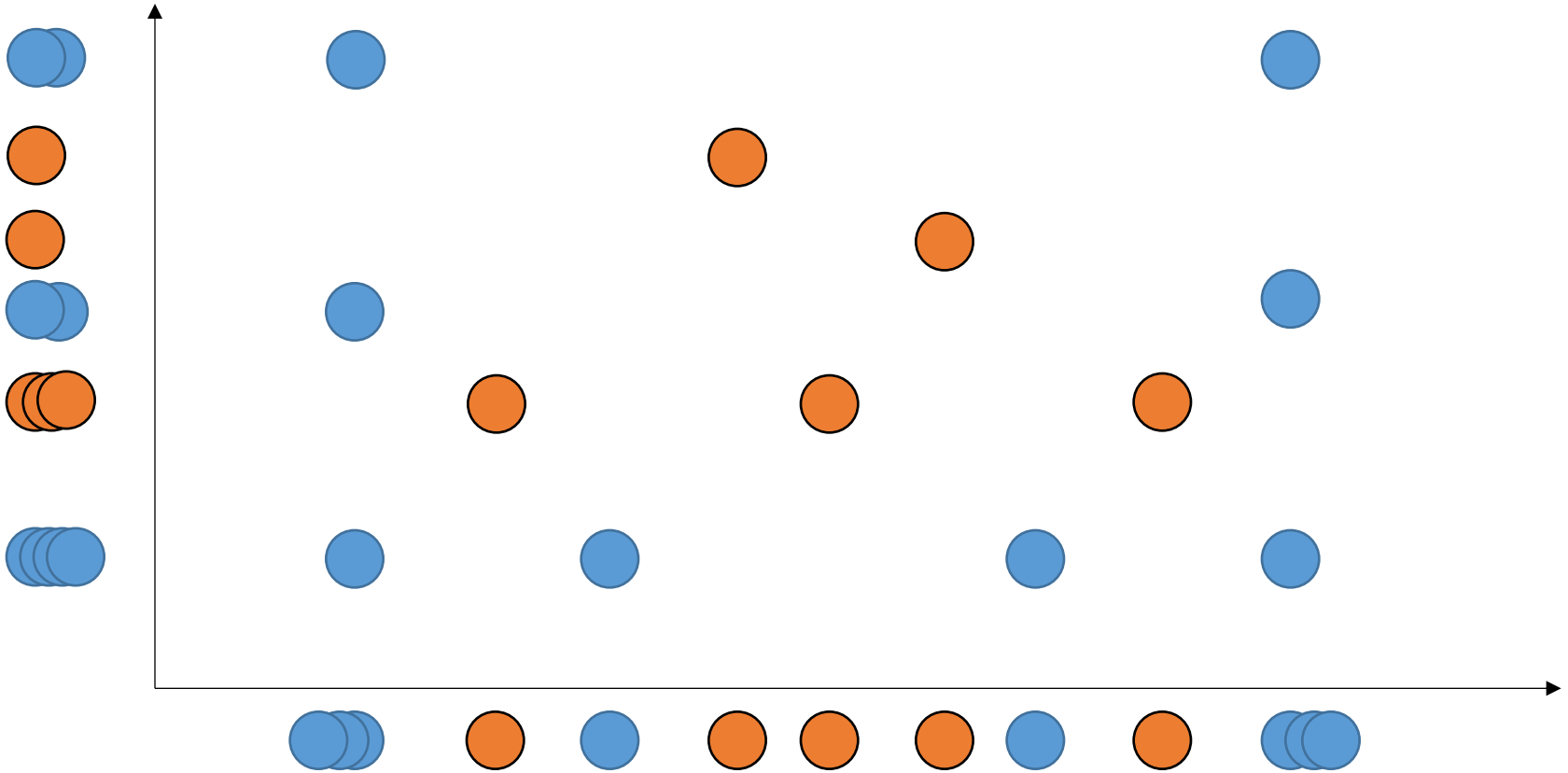
- $\mathrm{SplitNode}(m, R_m)$

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 $\mathrm{SplitNode}(r, R_r)$

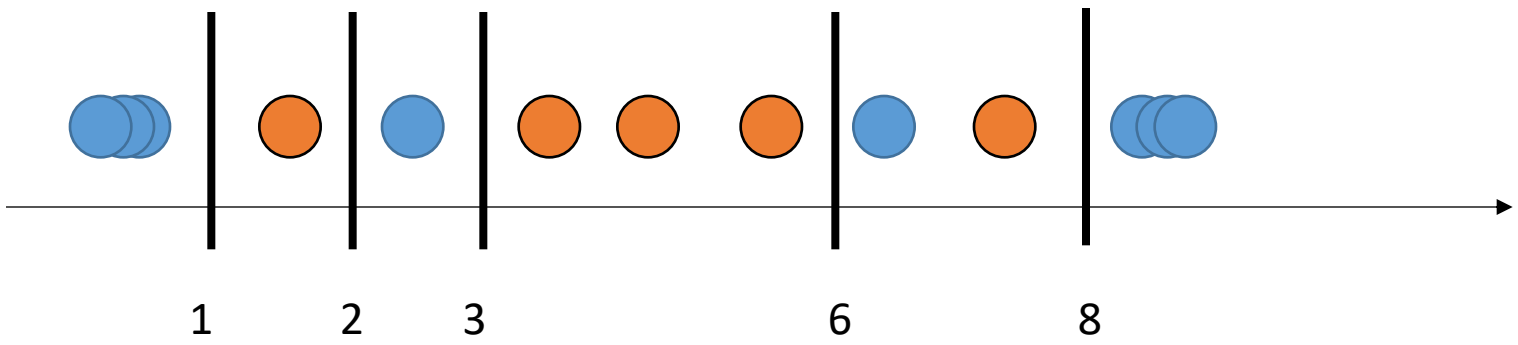
Обучение деревьев



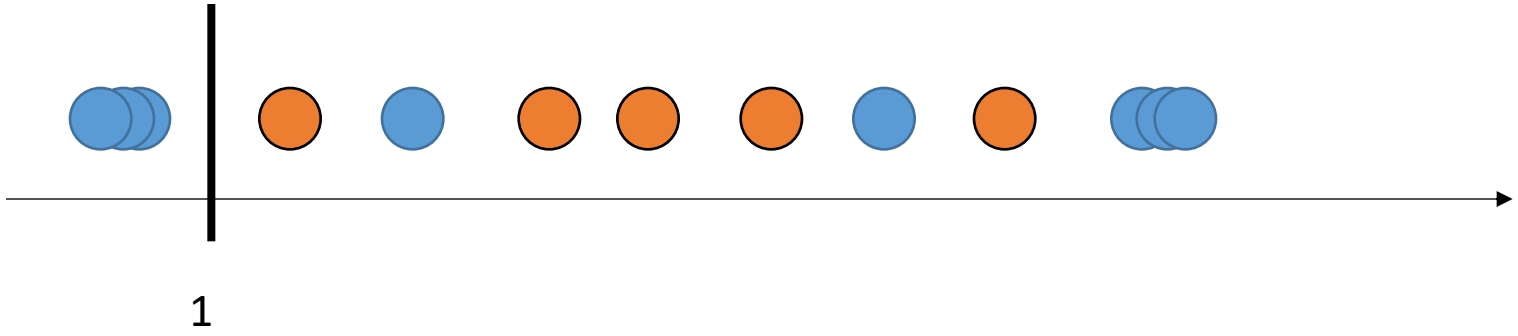
Признаки



Разбиения по признаку 1



Разбиения по признаку 1

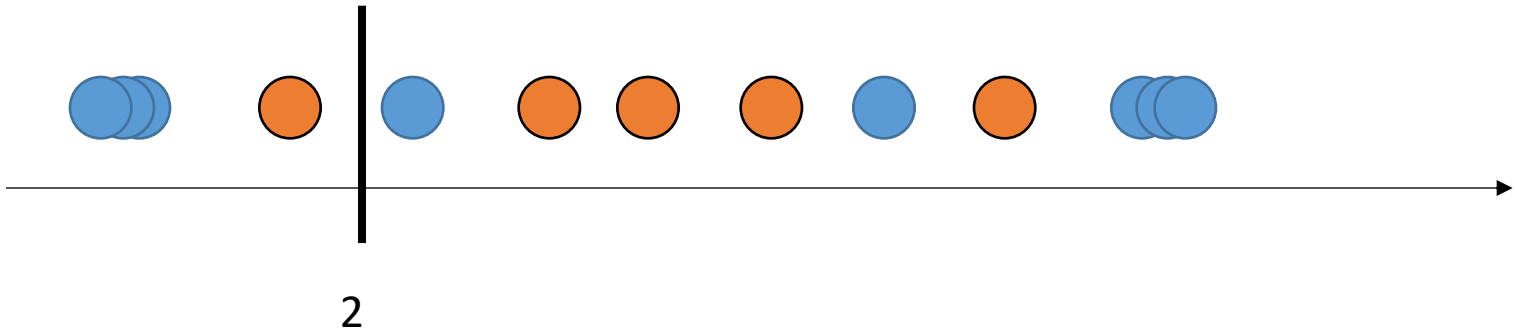


(1, 0)
H(p) = 0

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

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

Разбиения по признаку 1

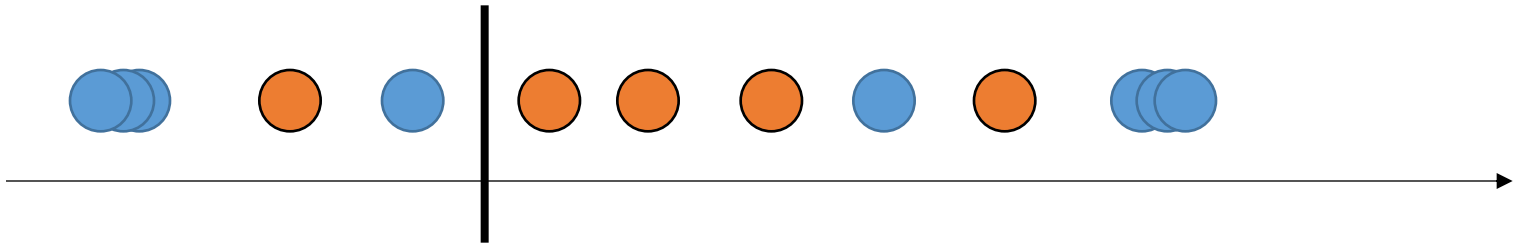


(3/4, 1/4)
H(p) = 0.56

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

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

Разбиения по признаку 1

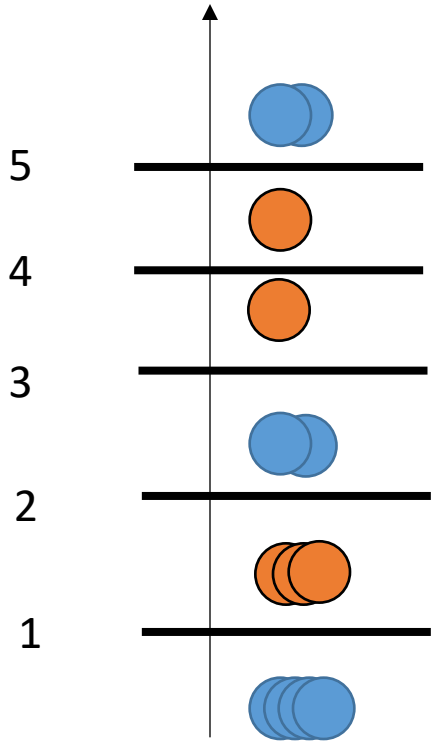


(4/5, 1/5)
H(p) = 0.5

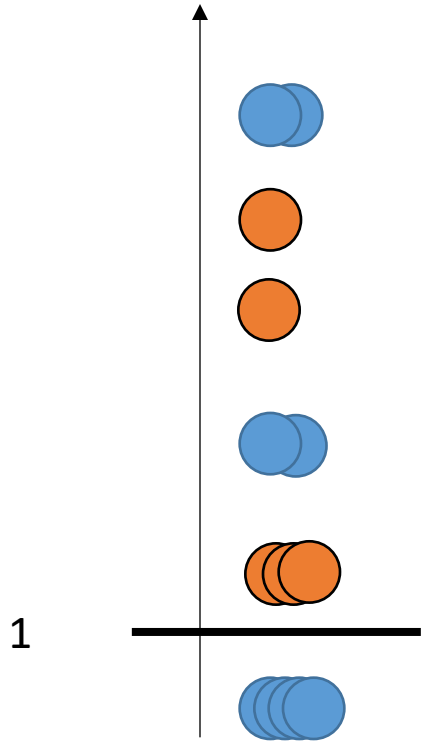
(1/2, 1/2)
H(p) = 0.69

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

Разбиения по признаку 2



Разбиения по признаку 2

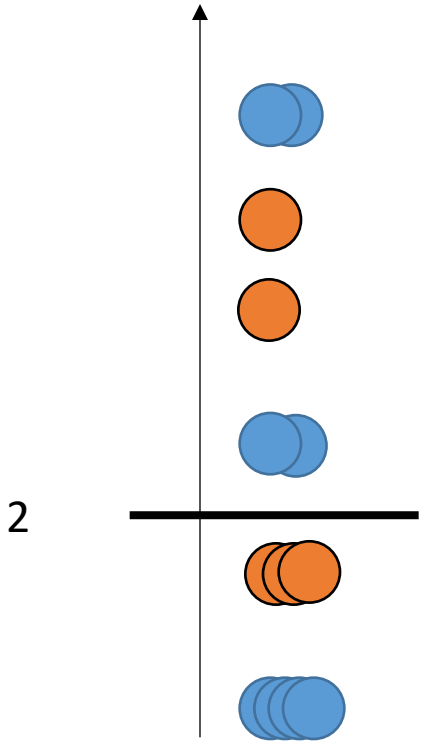


(4/9, 5/9)
H(p) = 0.69

(1, 0)
H(p) = 0

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

Разбиения по признаку 2

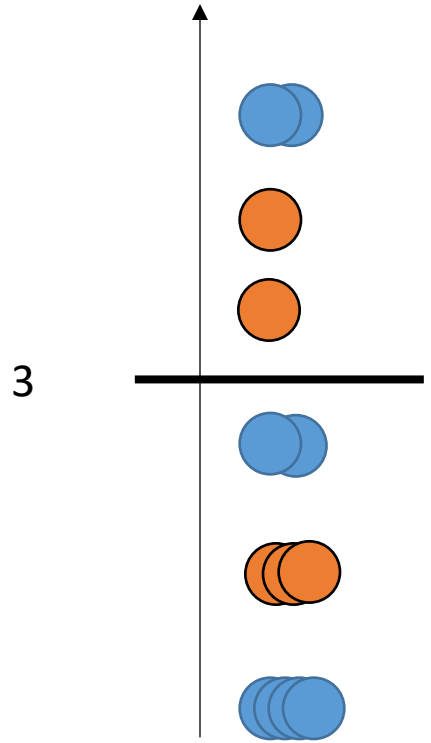


(4/6, 2/6)
H(p) = 0.64

(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

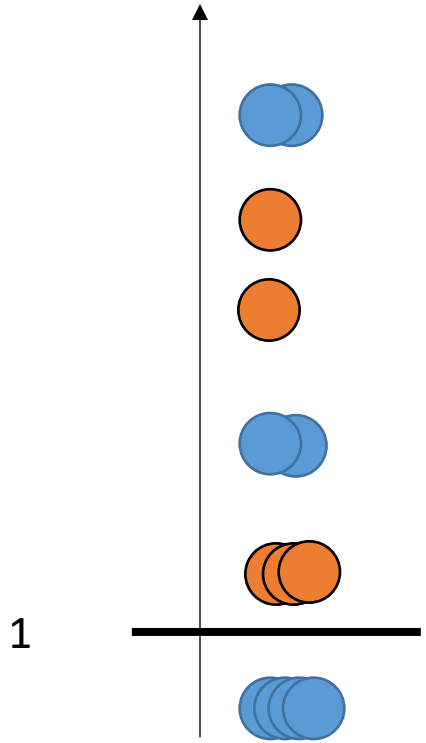


(1/2, 1/2)
H(p) = 0.69

(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



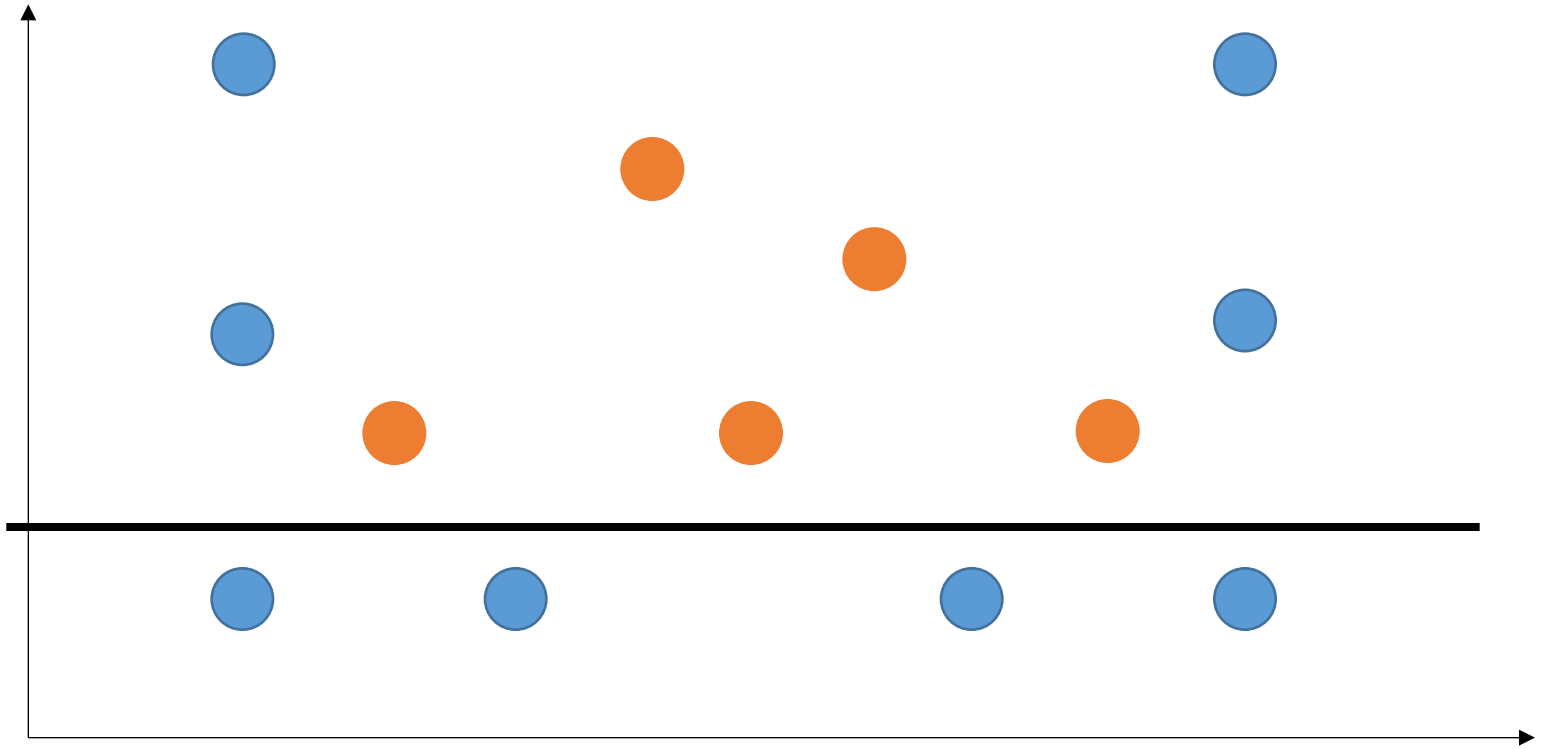
(4/9, 5/9)
H(p) = 0.69

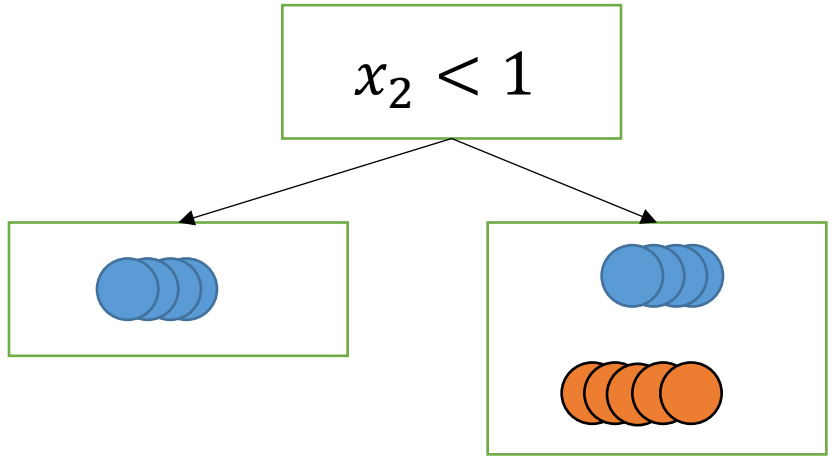
(1, 0)
H(p) = 0

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

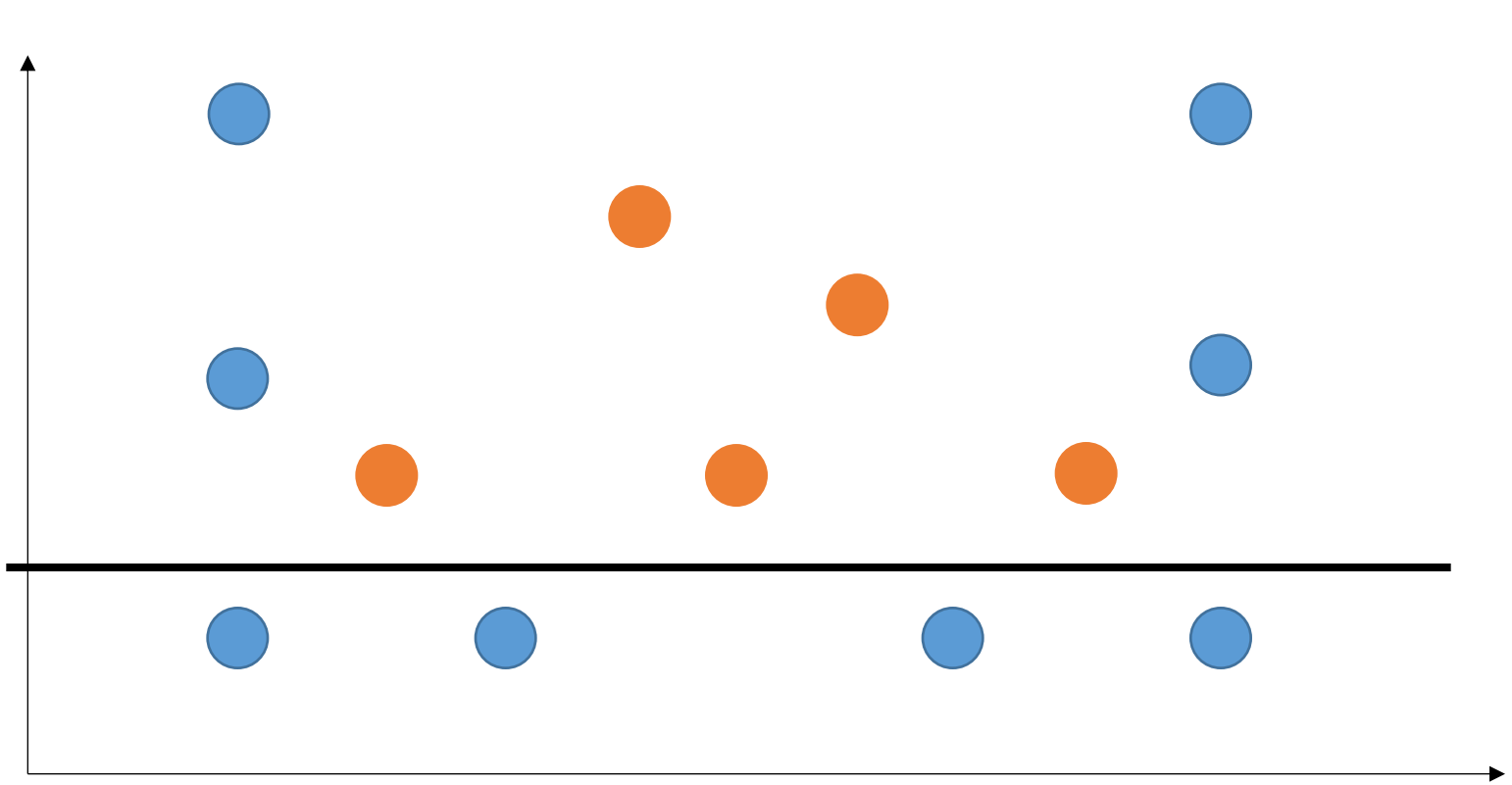
Лучшее разбиение!

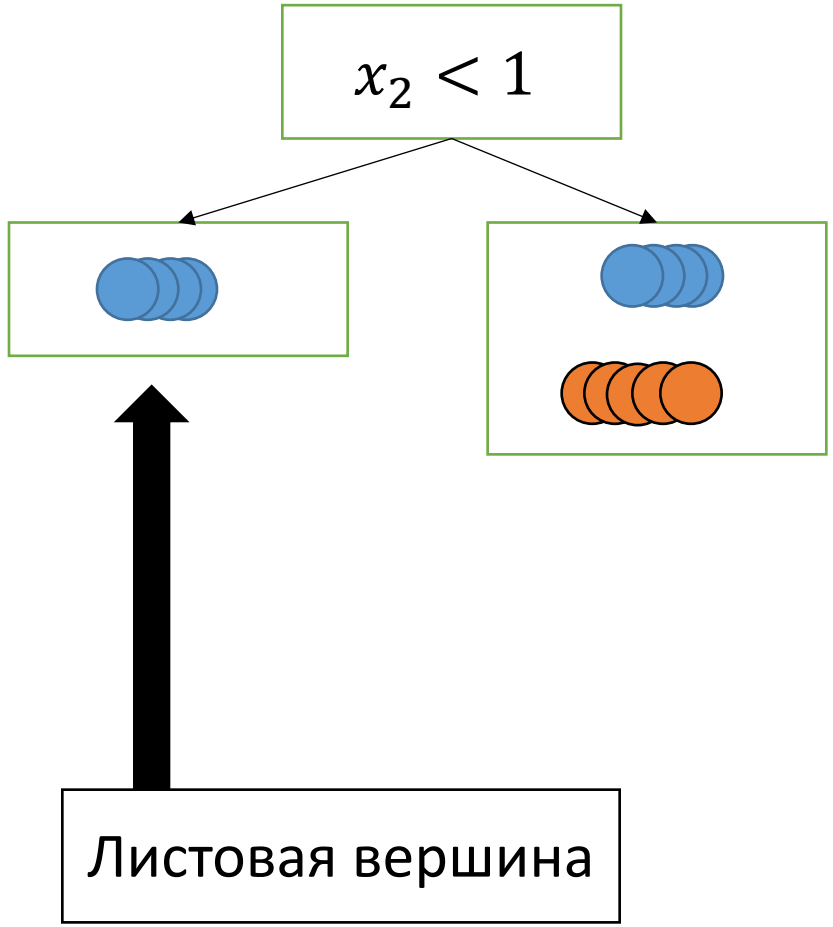
Обучение деревьев



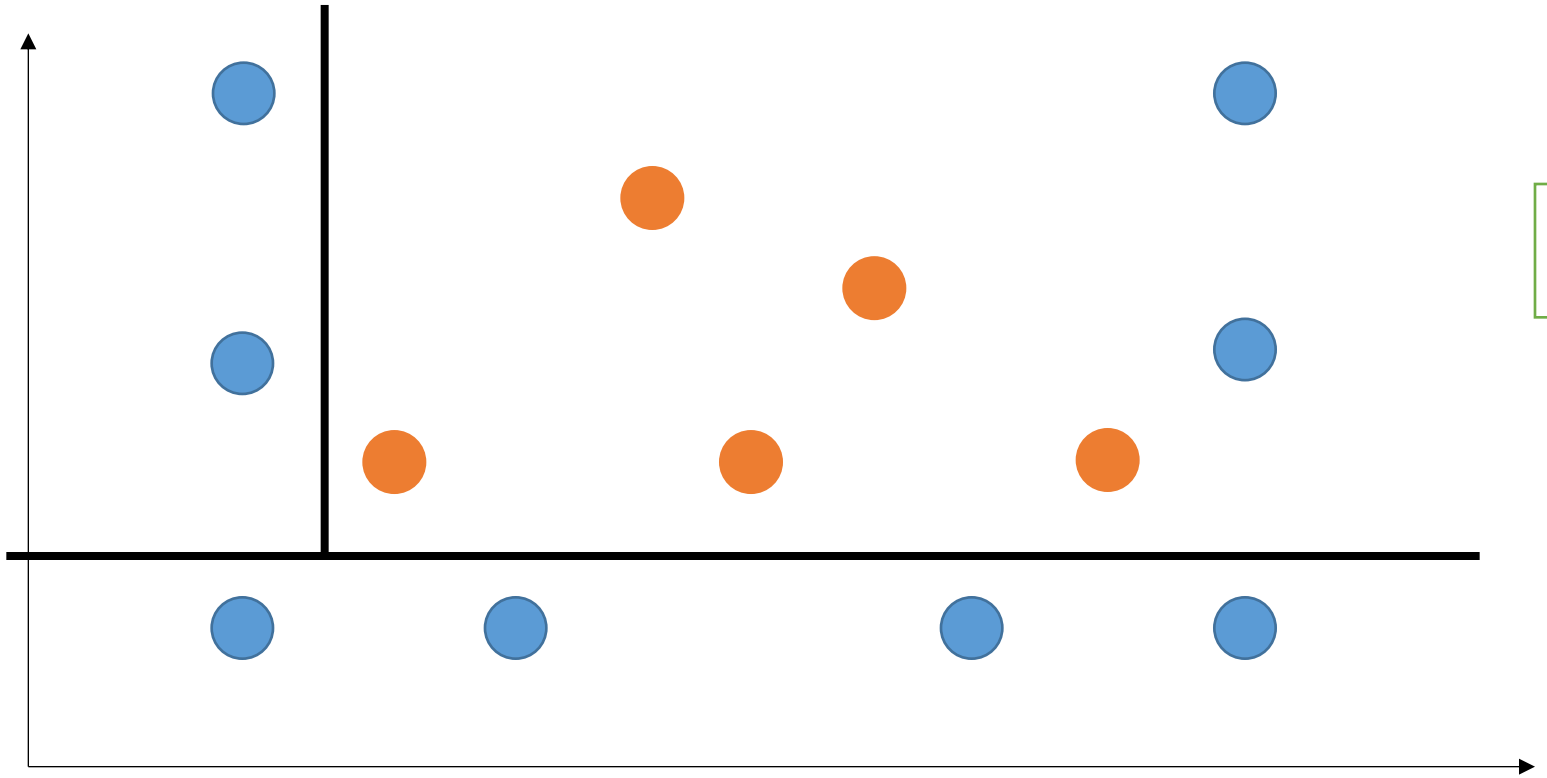


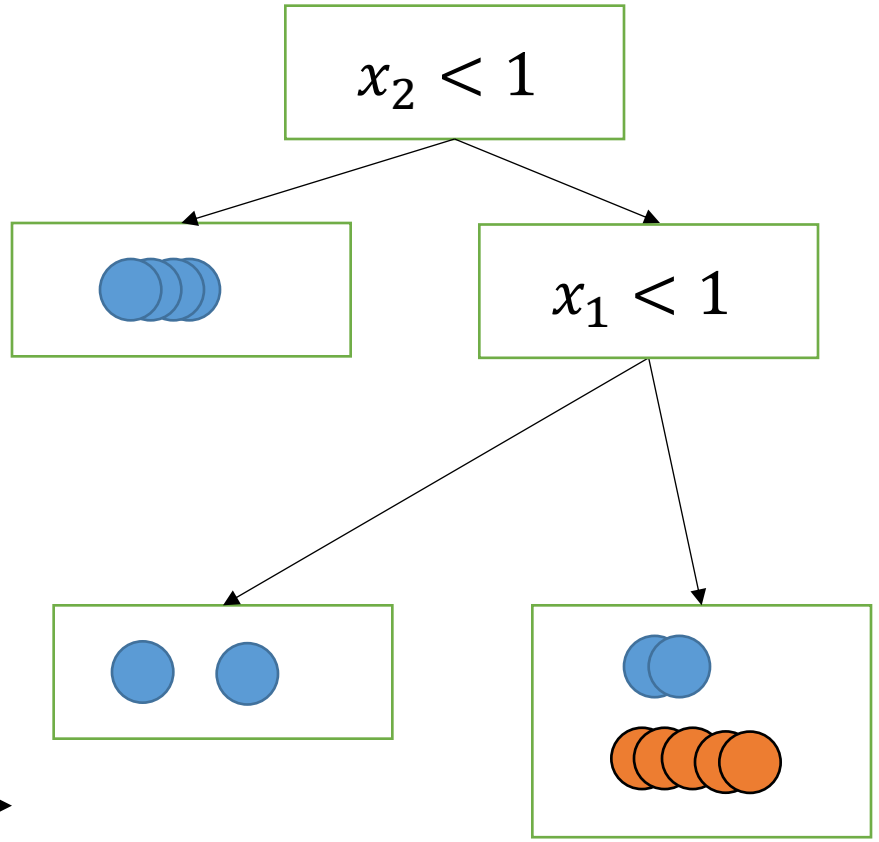
Обучение деревьев



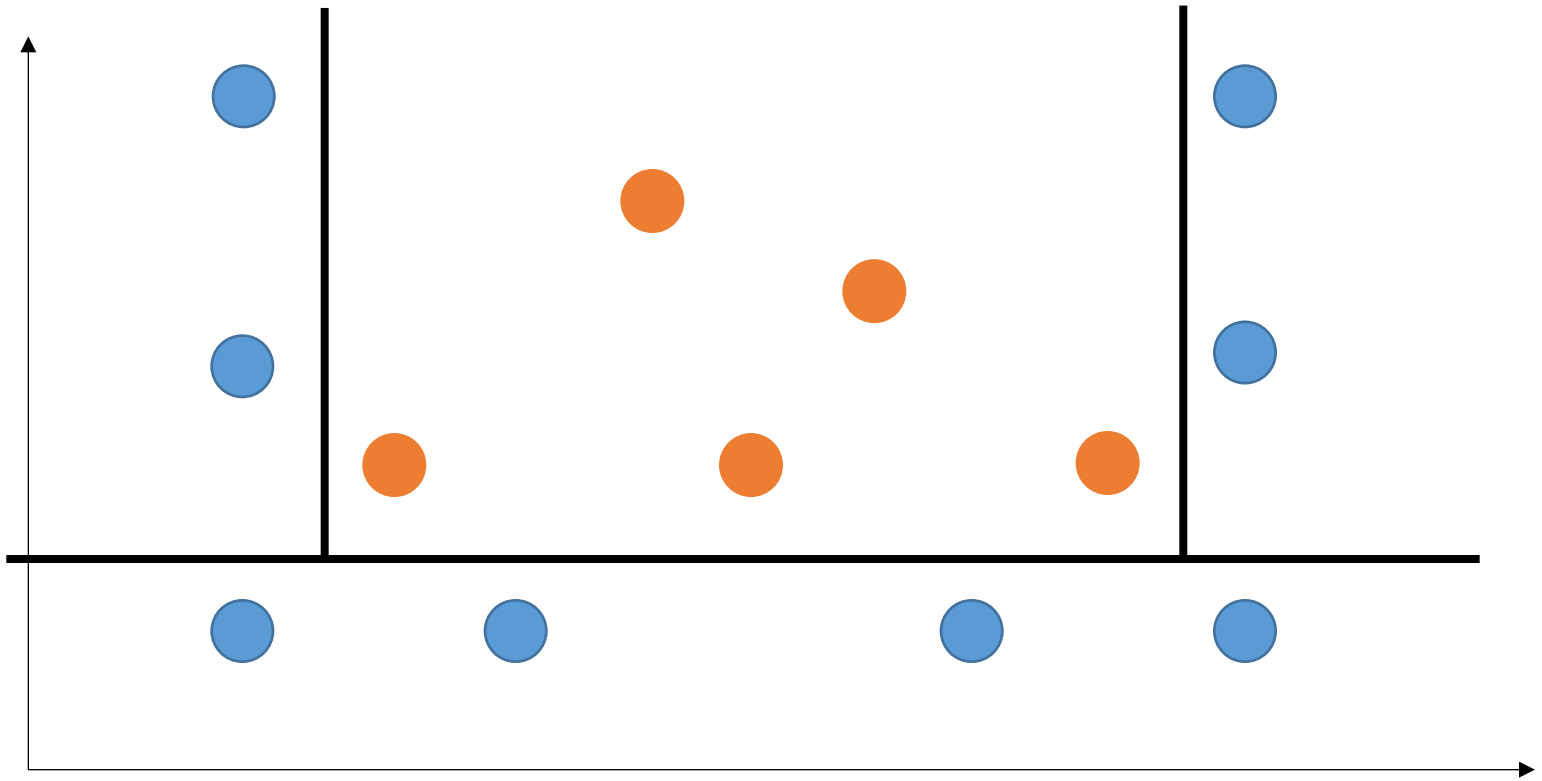


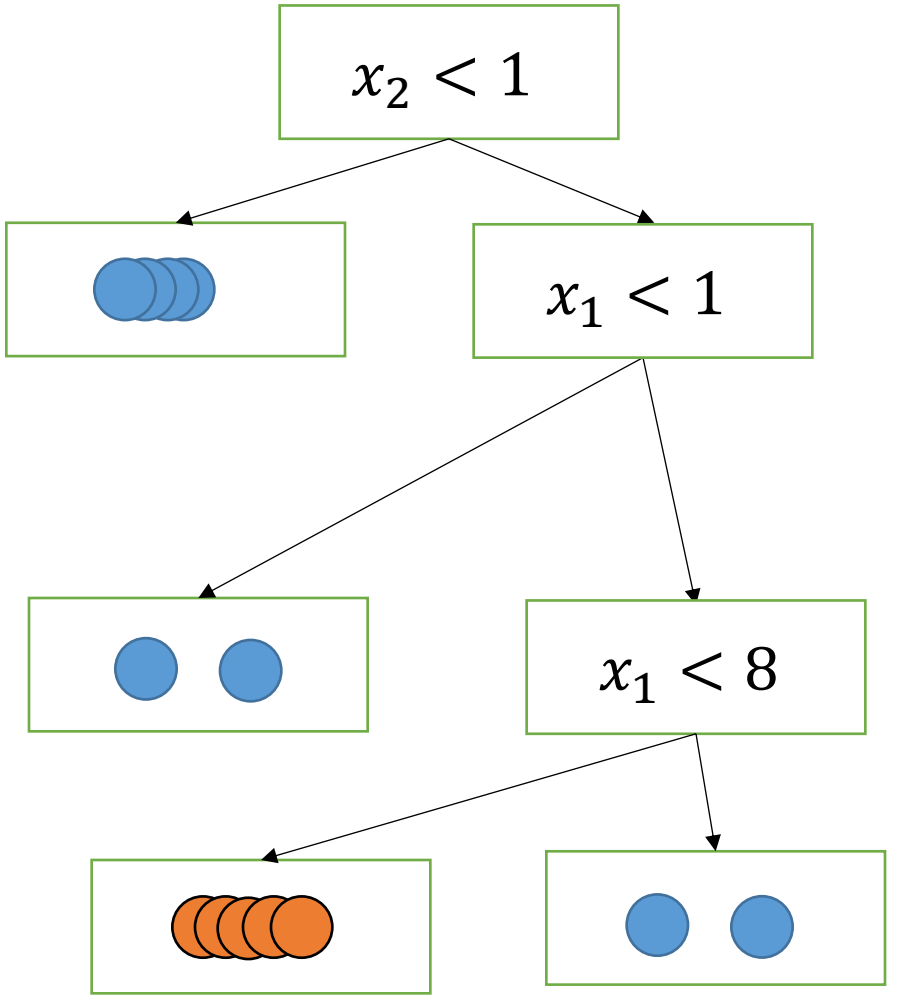
Обучение деревьев





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

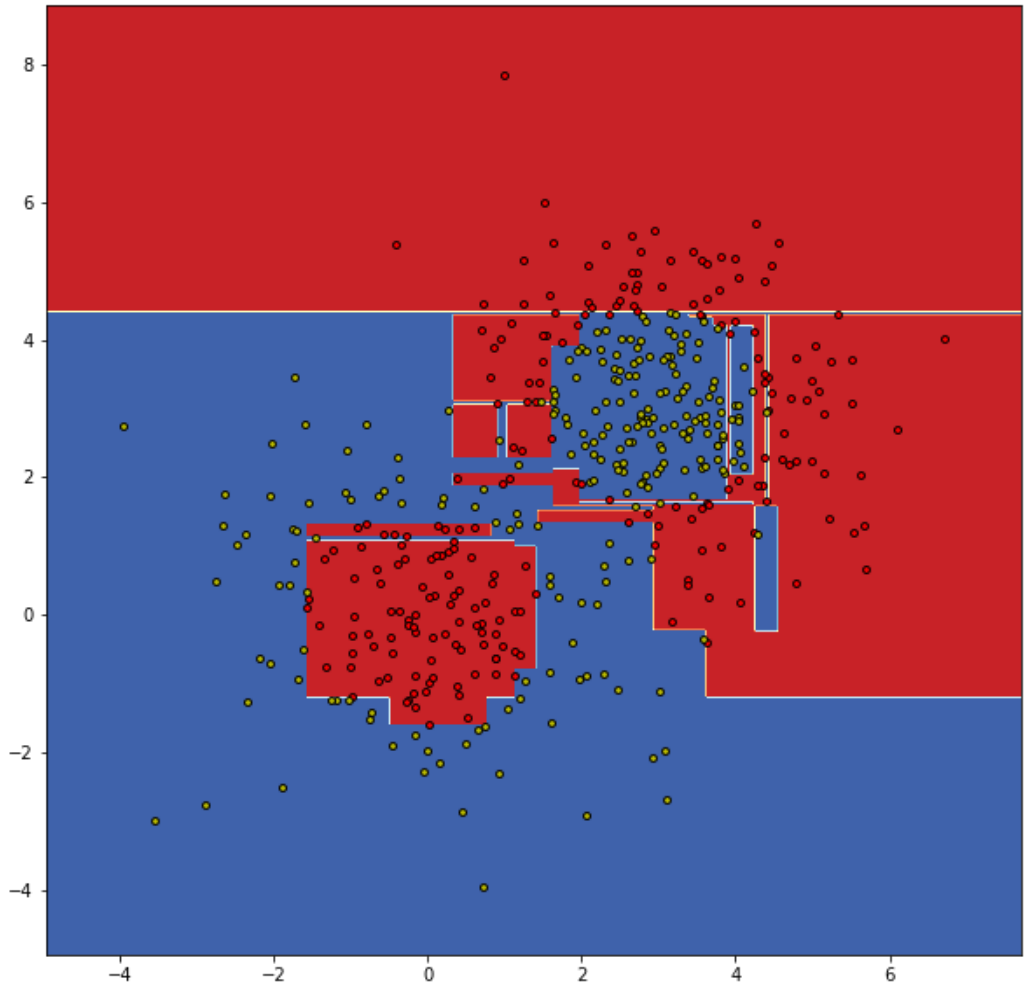
Устойчивость моделей

- $X = (x_i, y_i)_{i=1}^{\ell}$ — 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

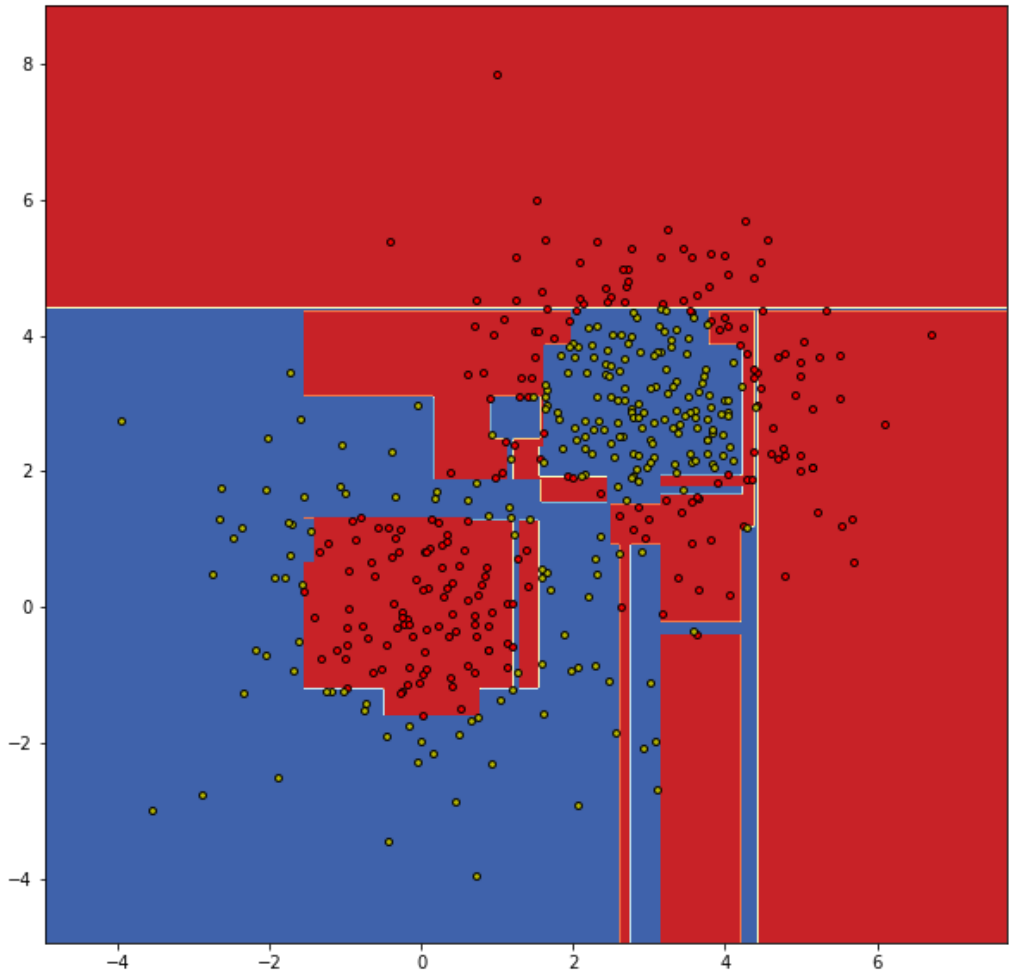
Устойчивость моделей

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

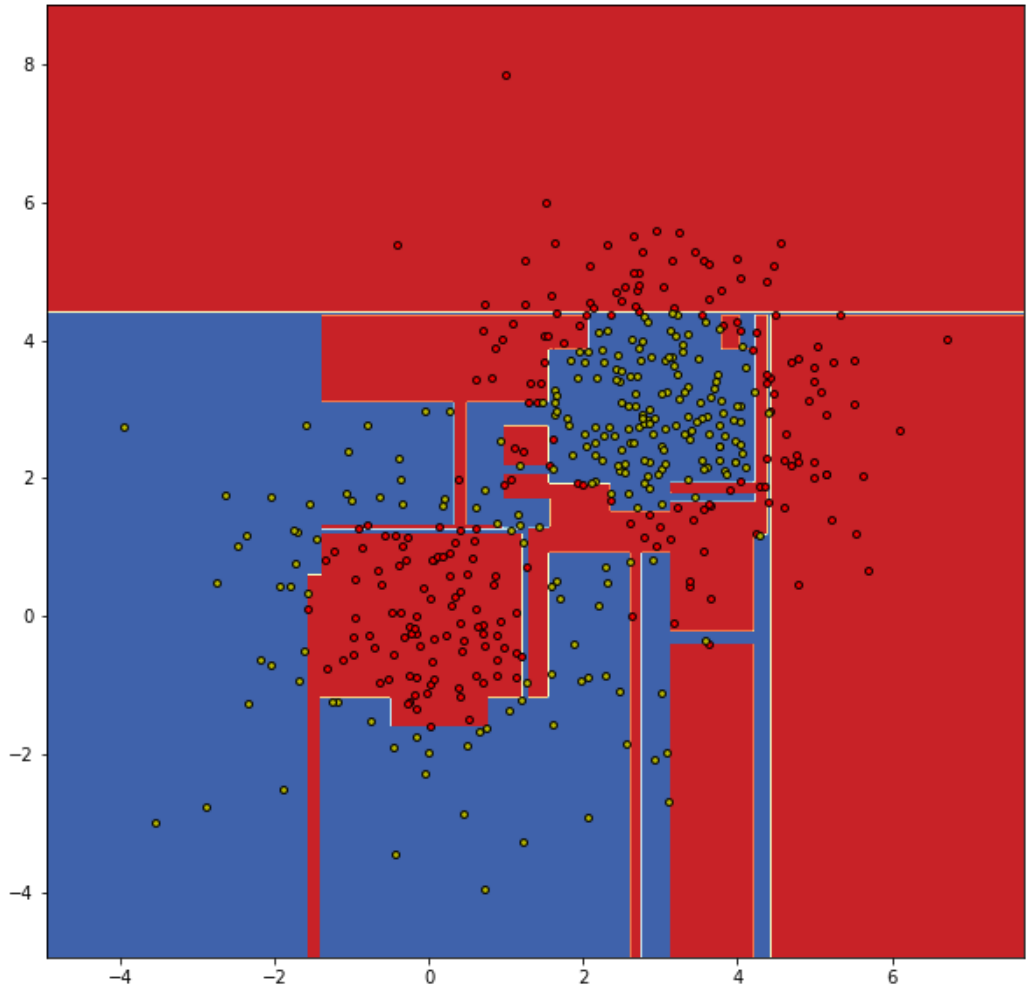
Обучение на подвыборках



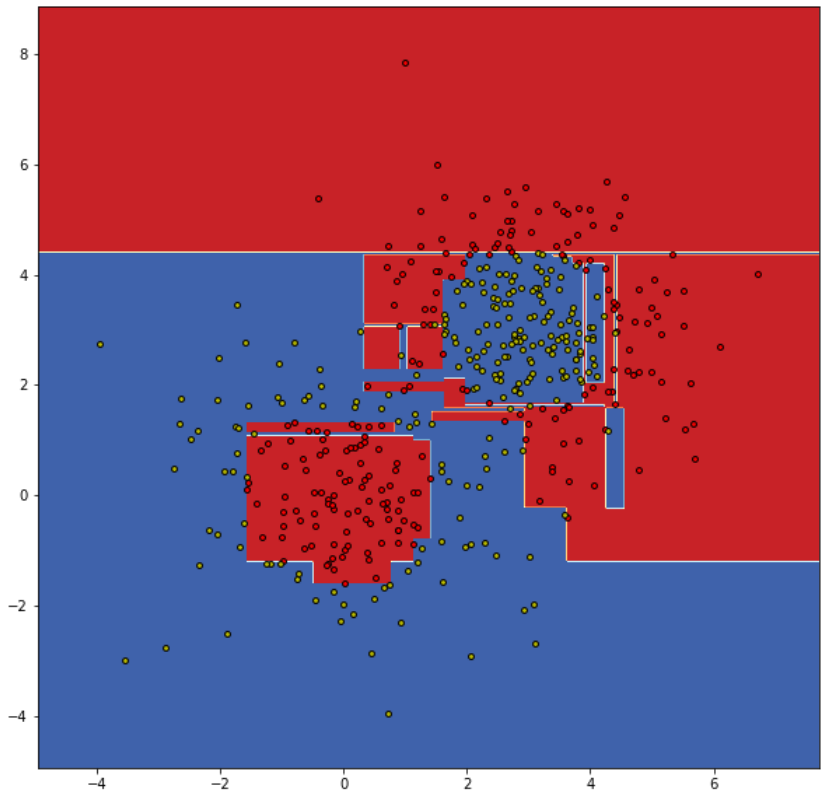
Обучение на подвыборках

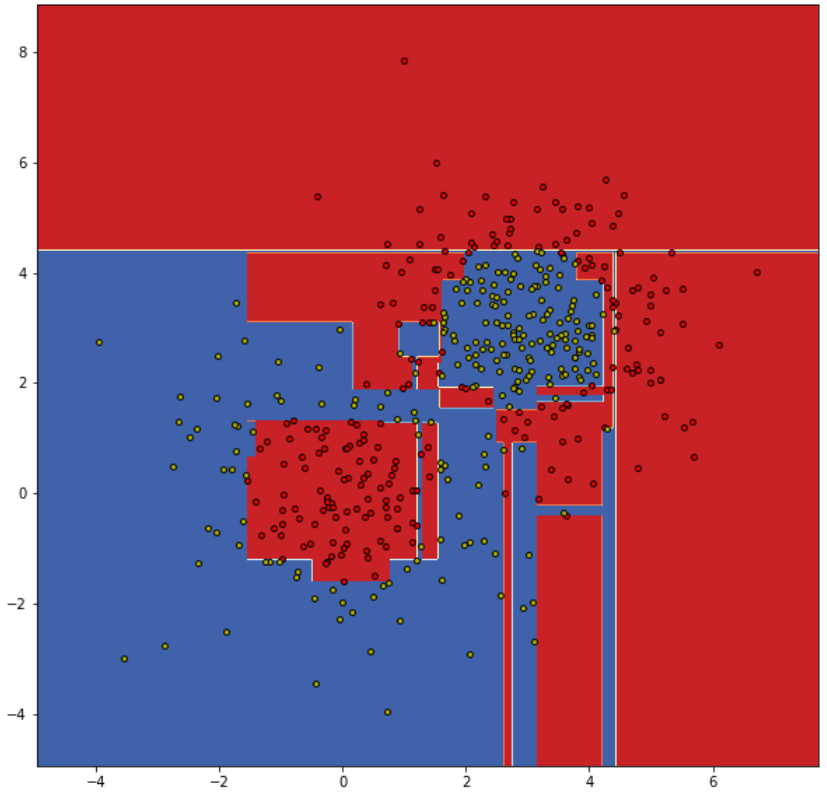


Обучение на подвыборках



Обучение на подвыборках



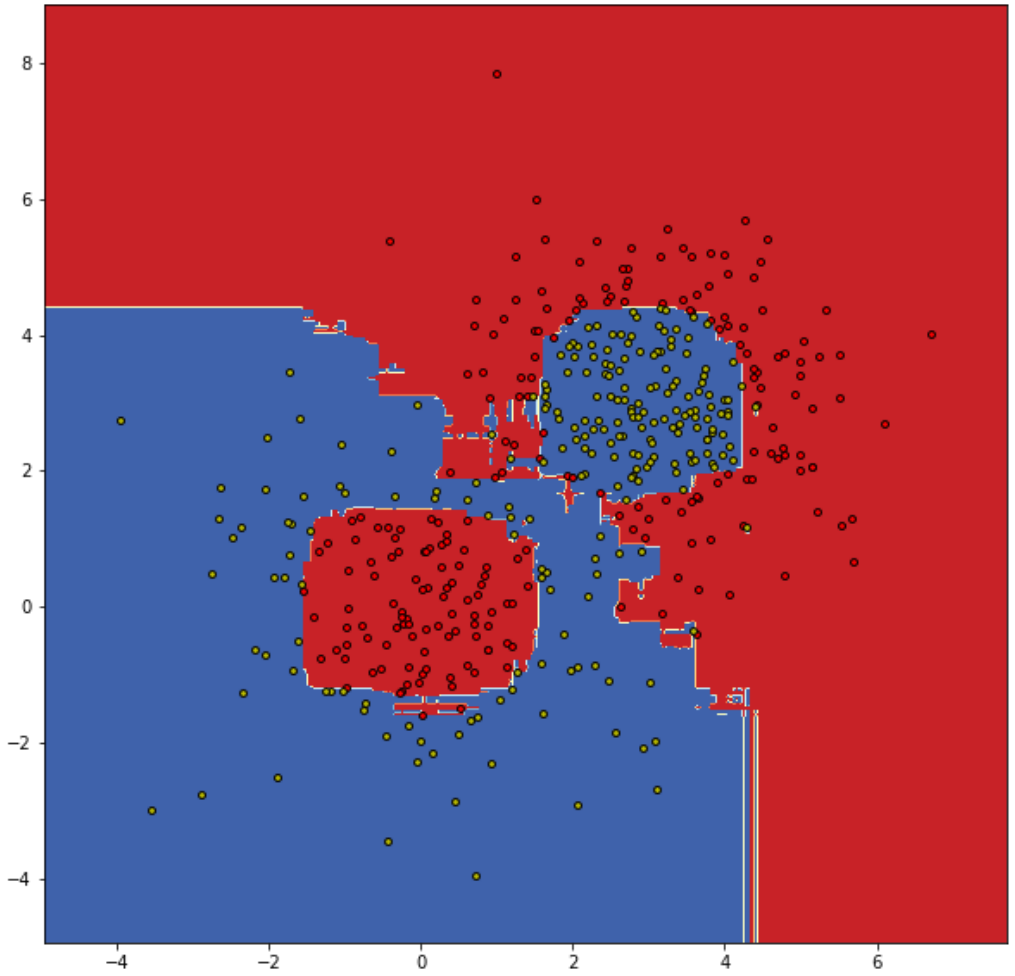


Композиция моделей

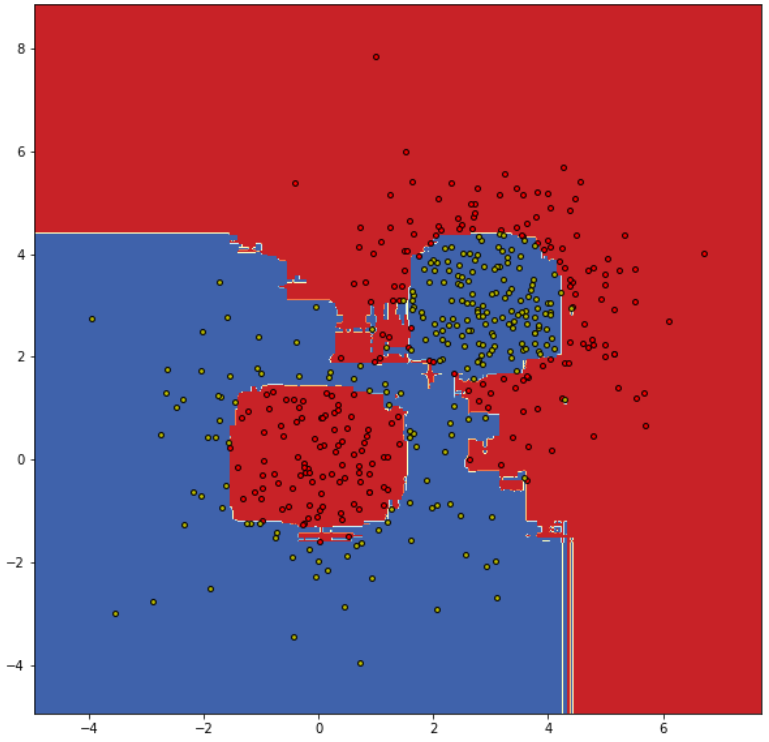
- У нас получилось N деревьев: $b_1(x), \ldots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

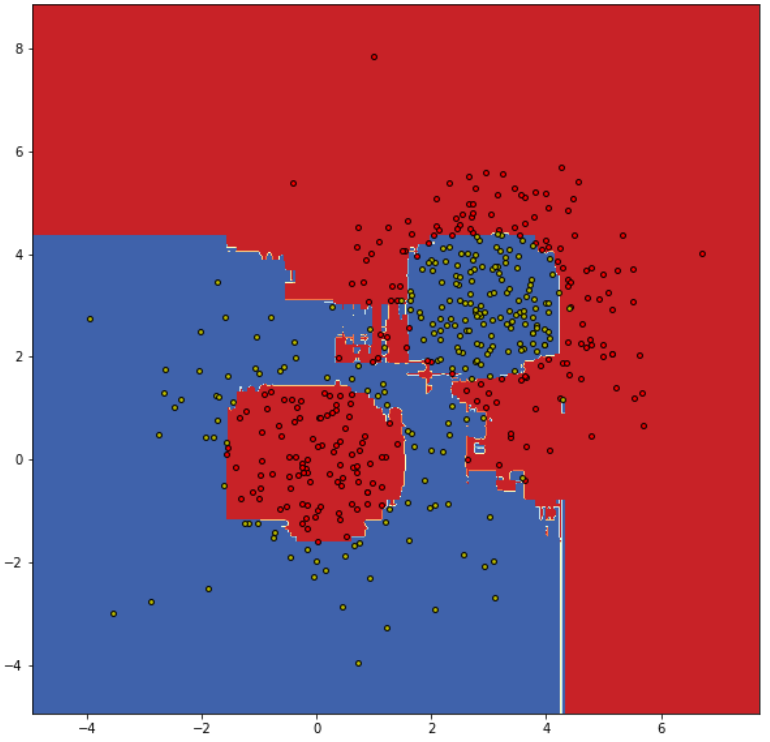
$$a(x) = \arg\max_{y \in \mathbb{Y}} \sum_{n=1}^{N} [b_n(x) = y]$$

Композиция моделей



Композиция моделей

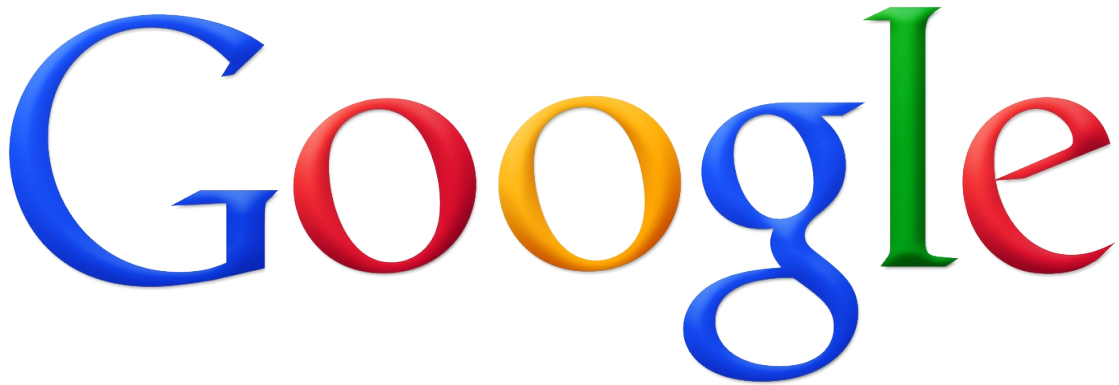




Голосование по большинству и усреднение

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 $b_1(x), \ldots, b_N(x)$
- 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

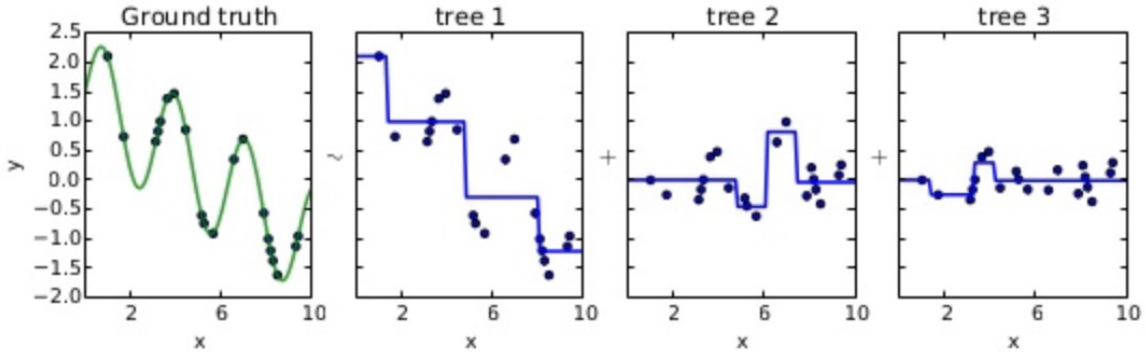
$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

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

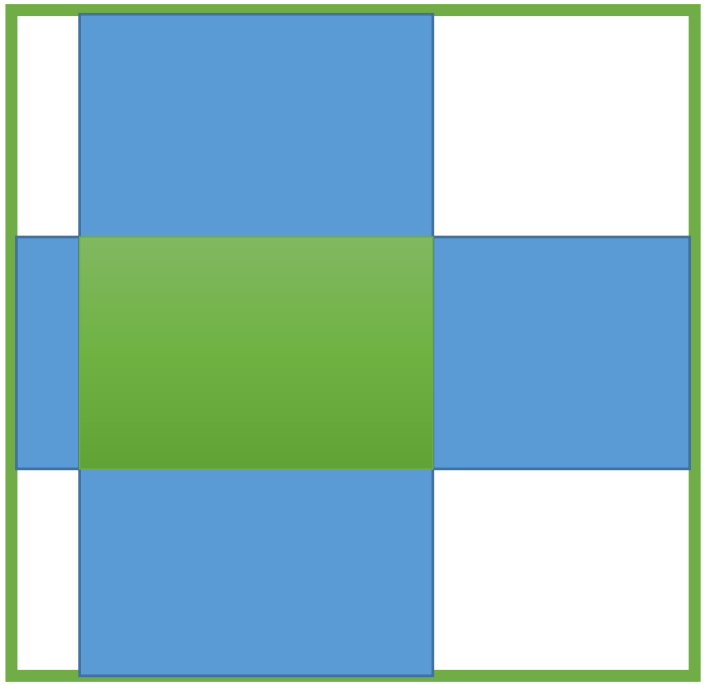
- Выбираем случайное подмножество признаков
- Обучаем модель только на них

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

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

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

- Разберём на уровне идеи

Разложение ошибки на смещение и разброс

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

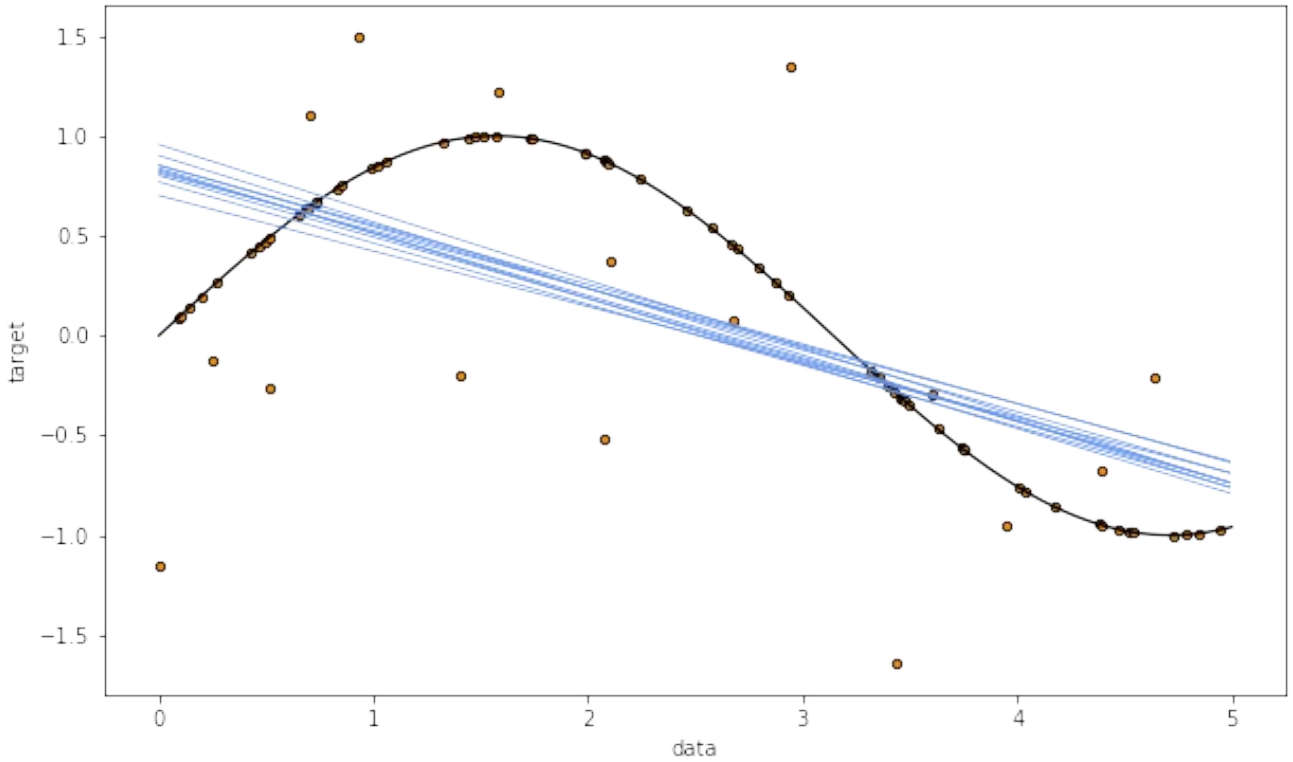
Разложение ошибки на смещение и разброс

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

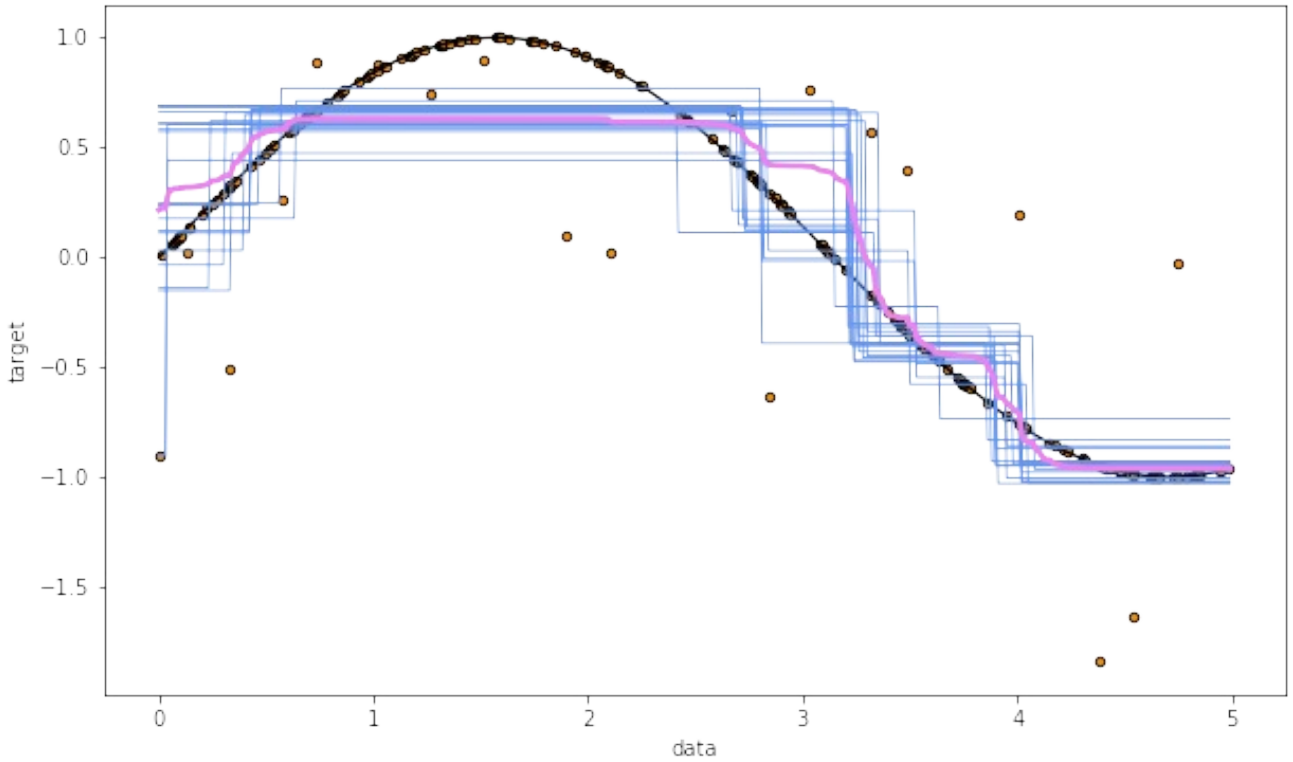
Разложение ошибки на смещение и разброс

- 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

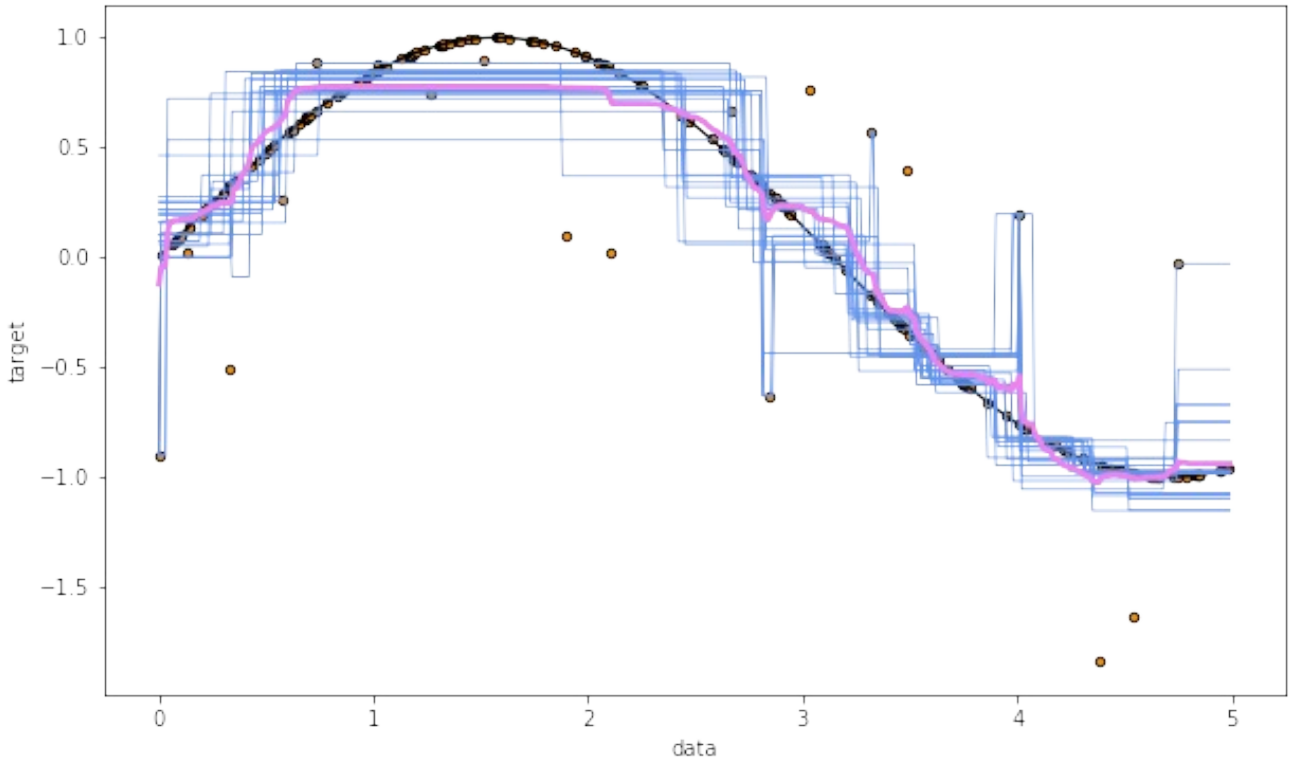
Смещение и разброс: линейная модель



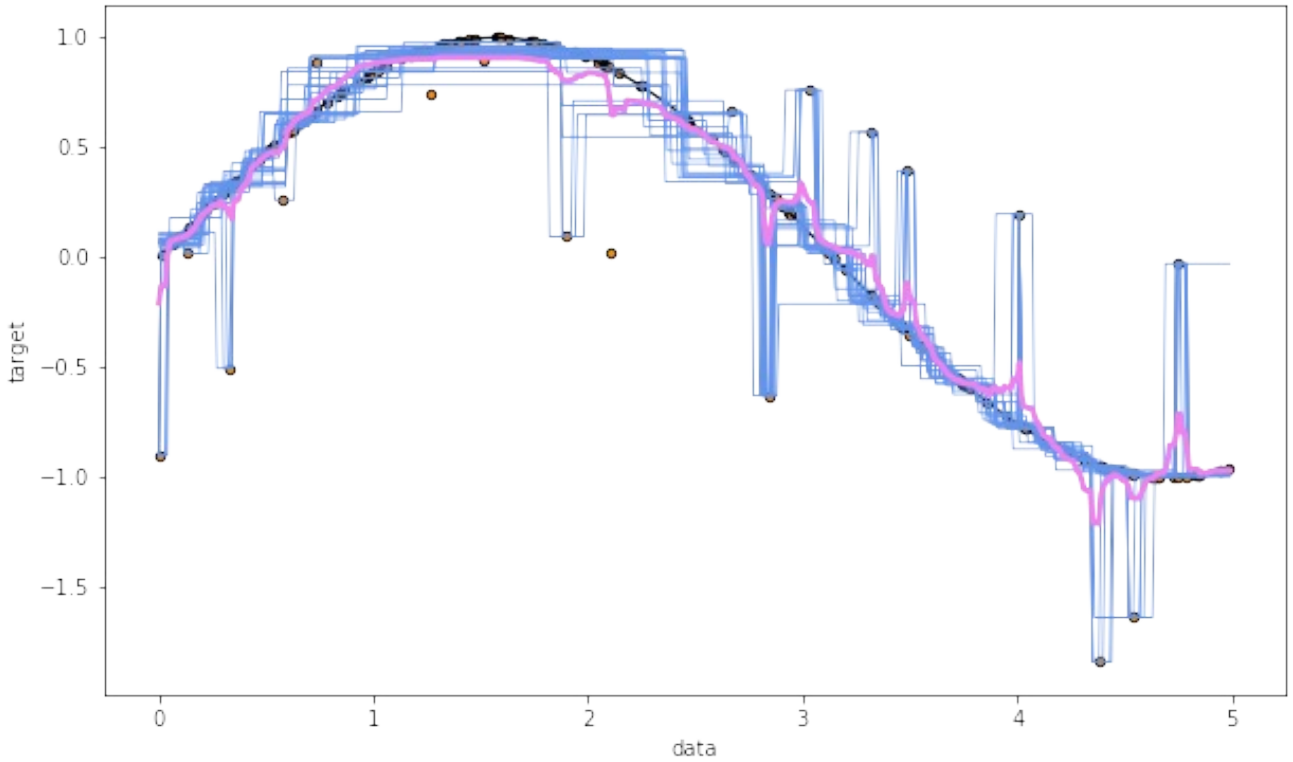
Смещение и разброс: деревья



Смещение и разброс: деревья



Смещение и разброс: деревья



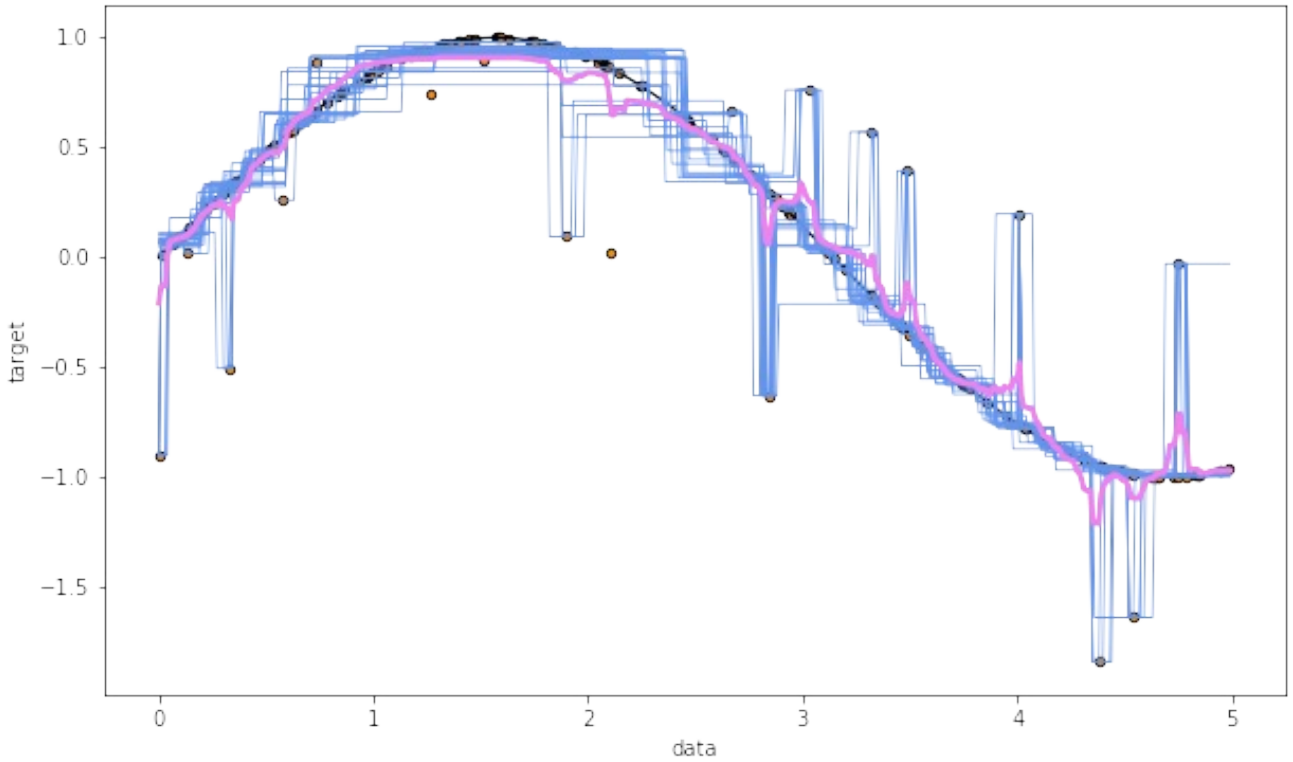
Бэггинг

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

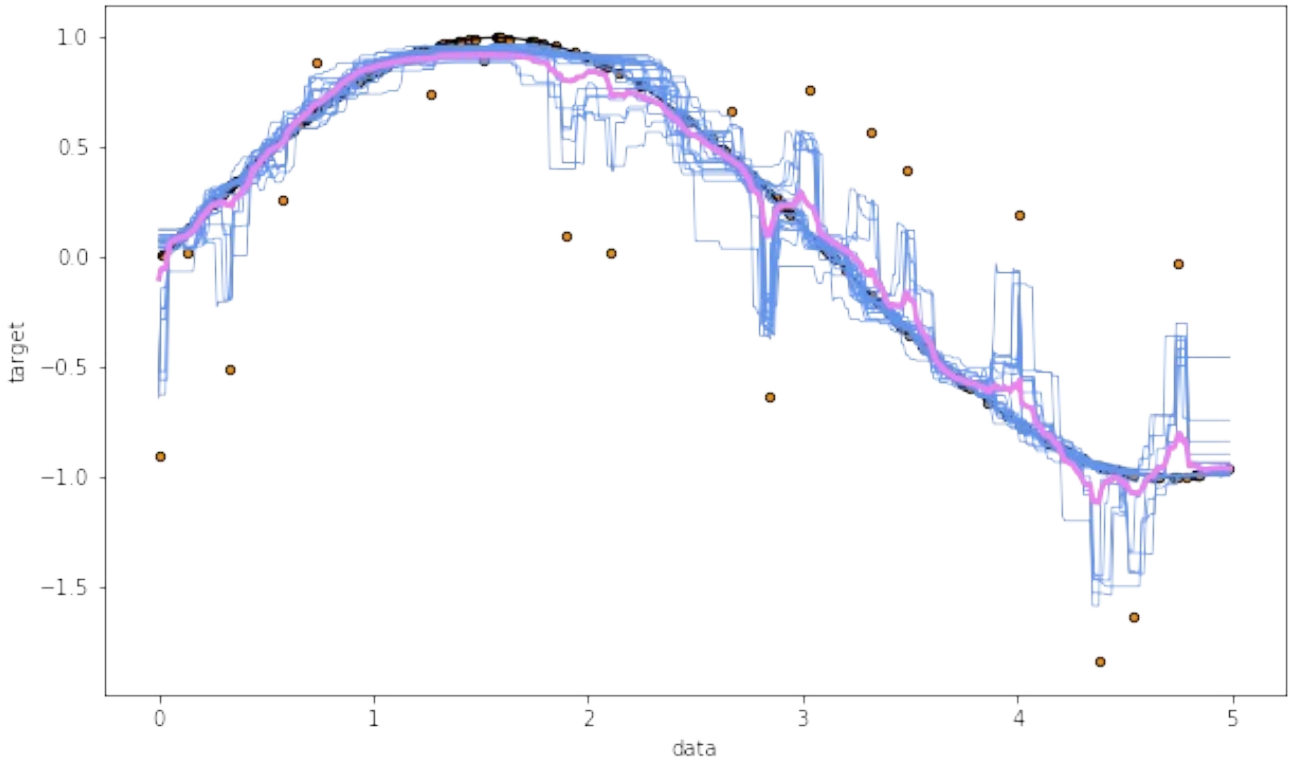
$$\frac{1}{N}\left(\text{разброс } b_n(x)\right) + \text{ковариация}\left(b_n(x), b_m(x)\right)$$

- 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

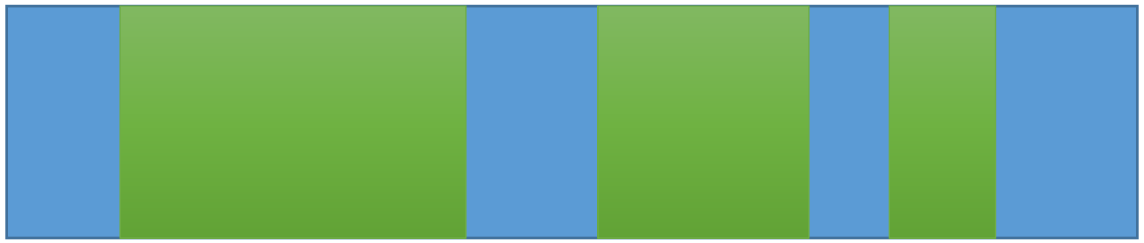
$\mathrm{SplitNode}(m, R_m)$

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 $\mathrm{SplitNode}(r, R_r)$

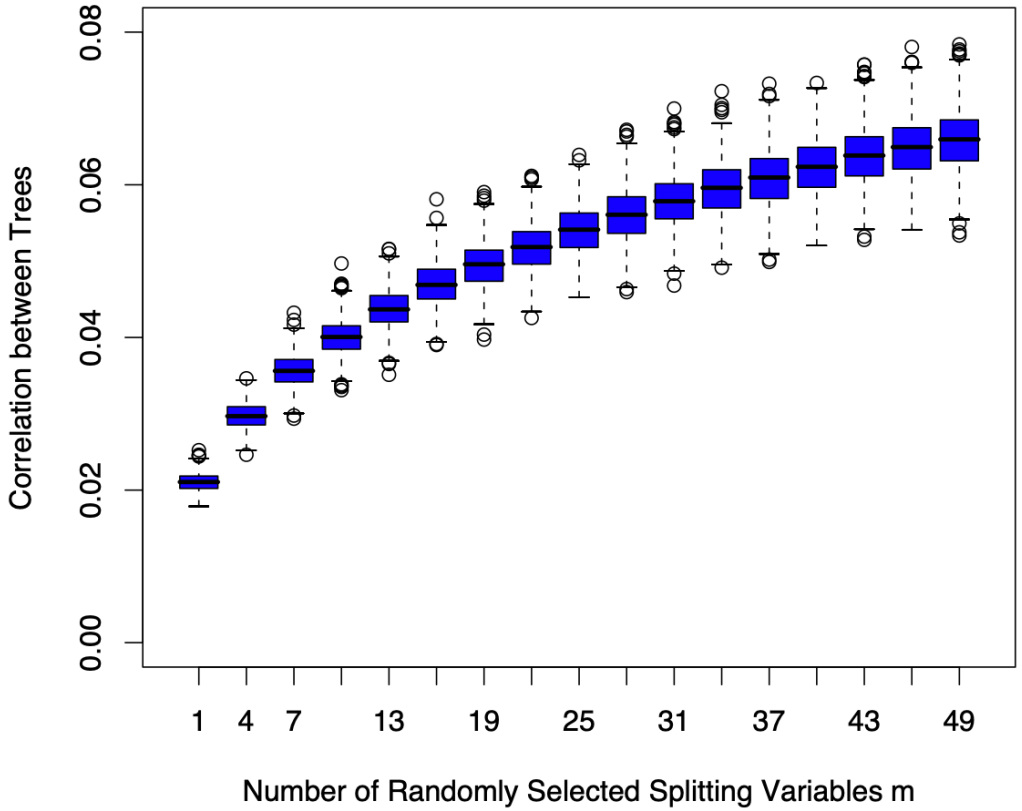
Выбор предиката

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

- 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:

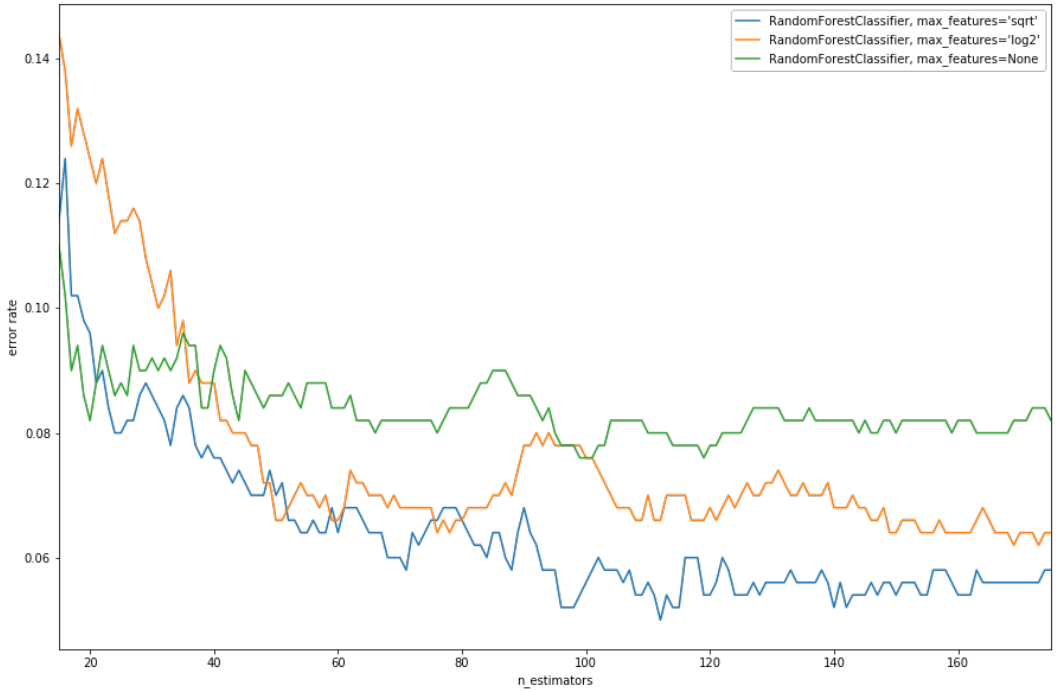
$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

- 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:

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

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

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:

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

Идея бустинга

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

- Обучение N-й модели:

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

Идея бустинга

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

- Обучение N-й модели:

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

Идея бустинга

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

- Обучение N-й модели:

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

- 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:

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

Бустинг для MSE

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

- Обучение N-й модели:

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

Бустинг для MSE

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

- Обучение N-й модели:

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

Бустинг для MSE

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

- Обучение N-й модели:

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

Бустинг для MSE

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

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

Первая итерация

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

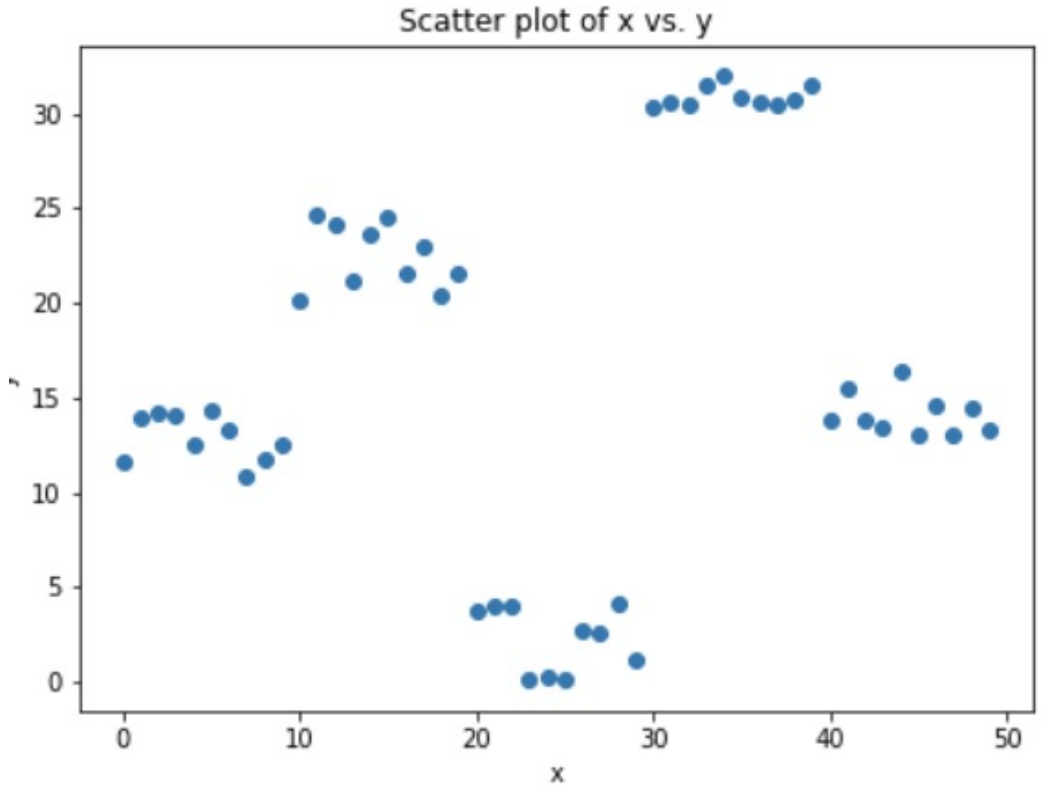
Вторая итерация

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\Big(b_2(x_i)-\big(y_i-b_1(x_i)\big)\Big)^2 \to \min_{b_2(x)}$$

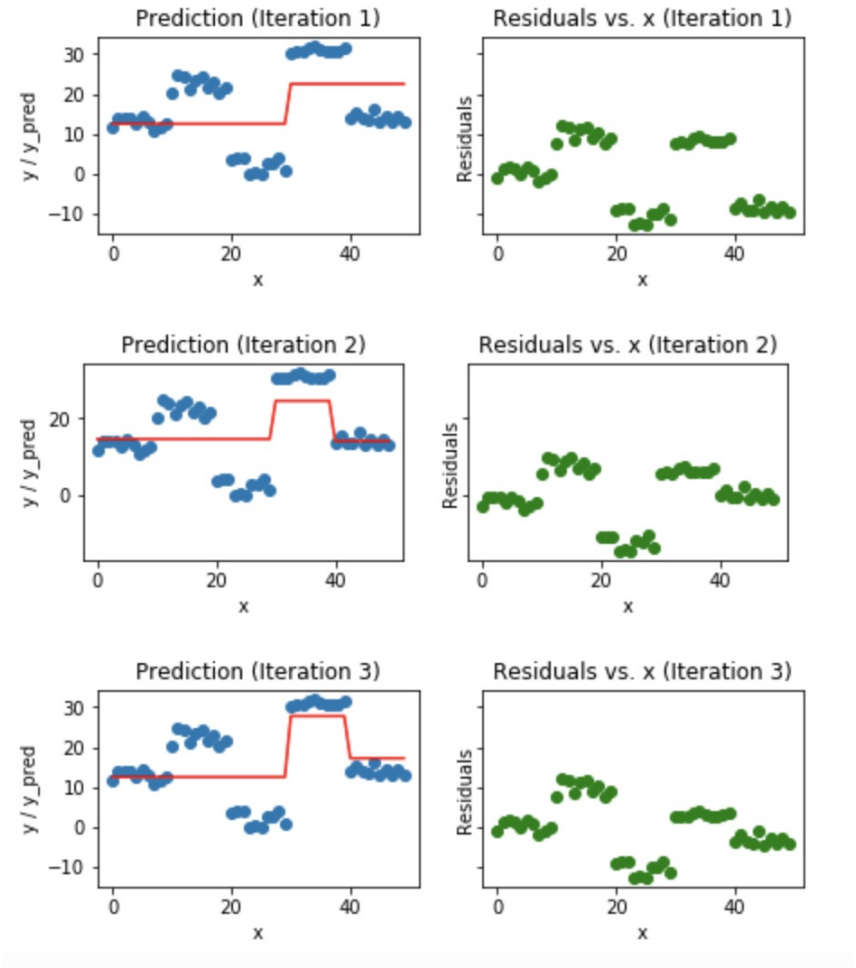
Третья итерация

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\Big(b_3(x_i)-\big(y_i-b_1(x_i)-b_2(x_i)\big)\Big)^2 \to \min_{b_3(x)}$$

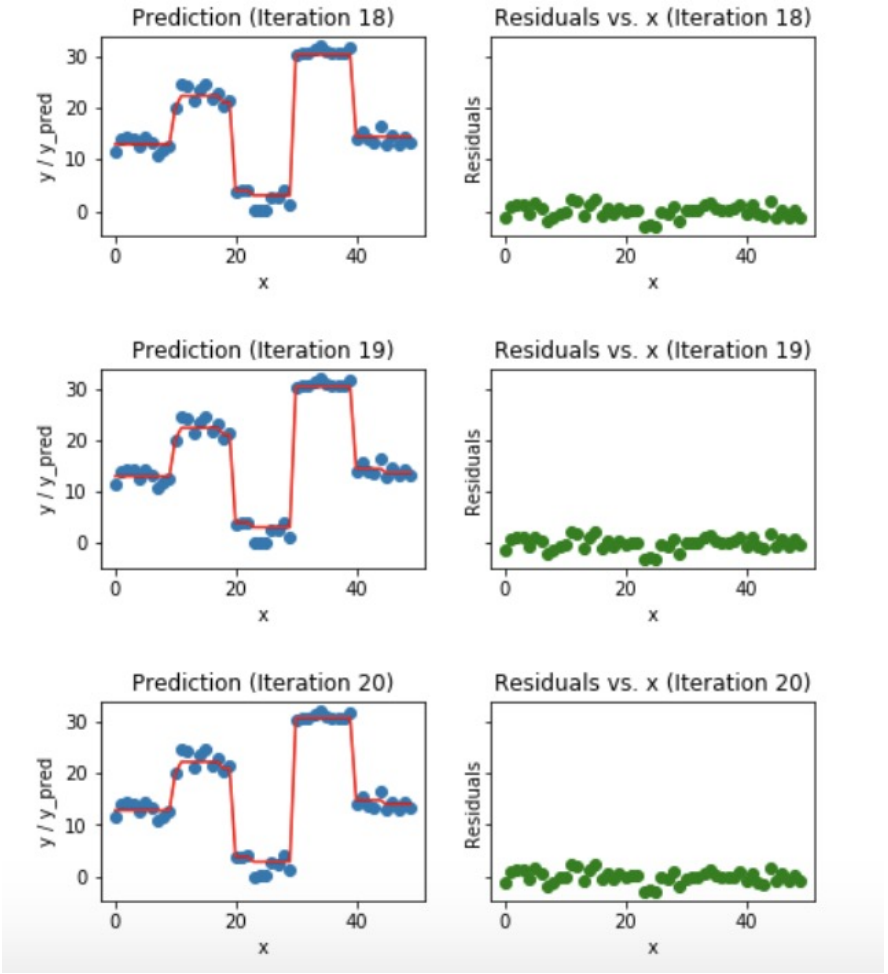
Визуализация



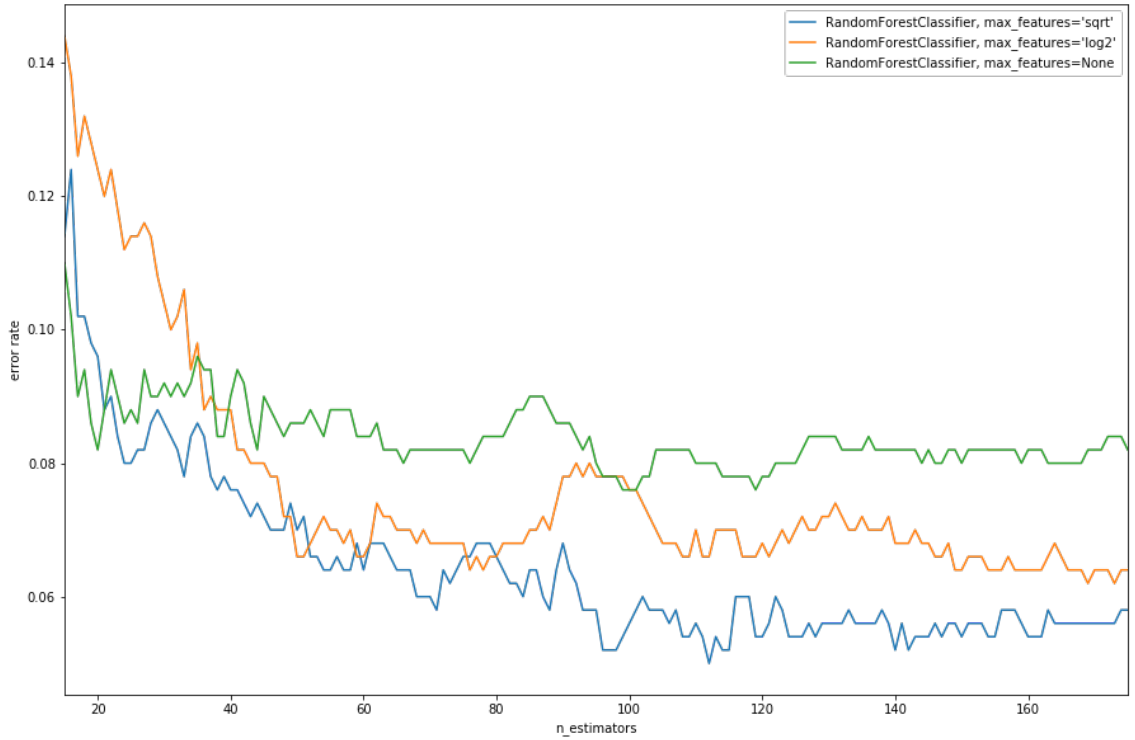
Визуализация



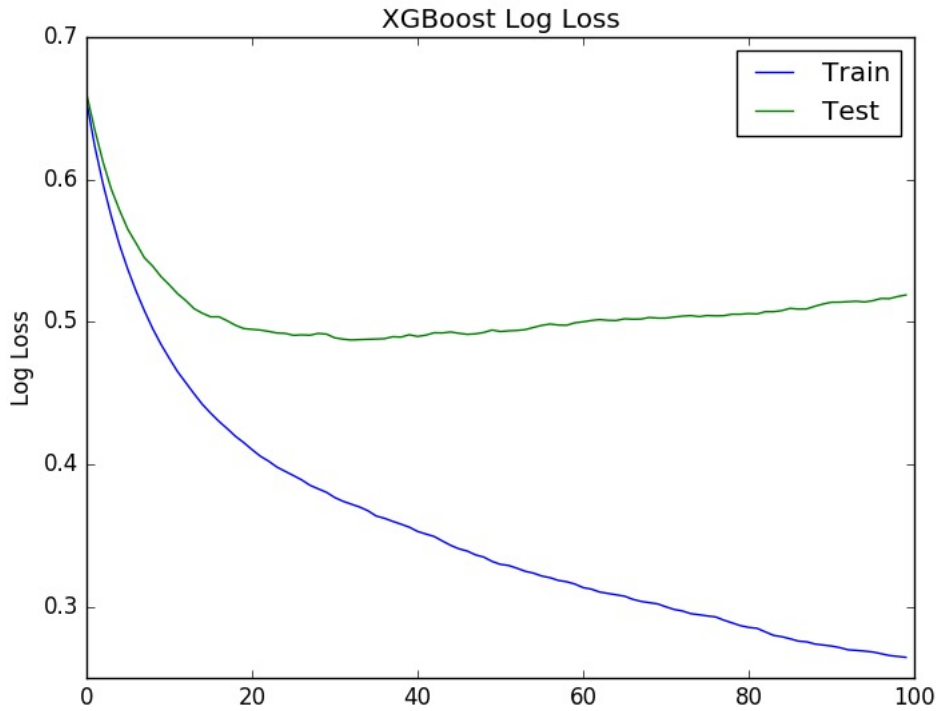
Визуализация



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:

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

Задача обучения базовой модели

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

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

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

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?

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

- 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

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

- 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

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

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

MSLE

- Mean Squared Logarithmic Error (среднеквадратичная логарифмическая ошибка)

$$L(y, z) = (\log(z + 1) - \log(y + 1))^2$$

MSLE

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

$$L(y, z) = (\log(z + 1) - \log(y + 1))^2$$

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

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

MSLE

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

$$L(y, z) = (\log(z + 1) - \log(y + 1))^2$$

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

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

- Аргумент второго логарифма может оказаться отрицательным

MSLE

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

y_i	$a_{N-1}(x_i)$	$b_N(x_i)$	**Улучшение MSLE композиции**	**Улучшение функционала базовой модели**
1000	100	2	0.09	13.7
2	0	2	1.2	1.2

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:

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

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

Задача обучения базовой модели

- Обучение N-й модели:

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

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

- Посчитать производную

Задача обучения базовой модели

- Обучение N-й модели:

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

- 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:

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

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

Градиентный бустинг

- Обучение N-й модели:

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

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

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

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

Градиентный бустинг для MSE

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

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

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

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

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

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

$$= [z < y](y-z) + 5[z \geq y](y-z)$$

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

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

$$= -\frac{\partial}{\partial z} \log(1 + \exp(-y_i z))\Big|_{z=a_{N-1}(x_i)} =$$

$$= \frac{y_i}{1 + \exp\left(y_i a_{N-1}(x_i)\right)}$$

Градиентный бустинг для логистической функции потерь

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

$$= -\frac{\partial}{\partial z} \log(1 + \exp(-y_i z))\Big|_{z=a_{N-1}(x_i)} =$$

$$= \frac{y_i}{1 + \exp\big(y_i a_{N-1}(x_i)\big)}$$

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

Градиентный бустинг для логистической функции потерь

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

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

Градиентный бустинг для логистической функции потерь

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

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

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)