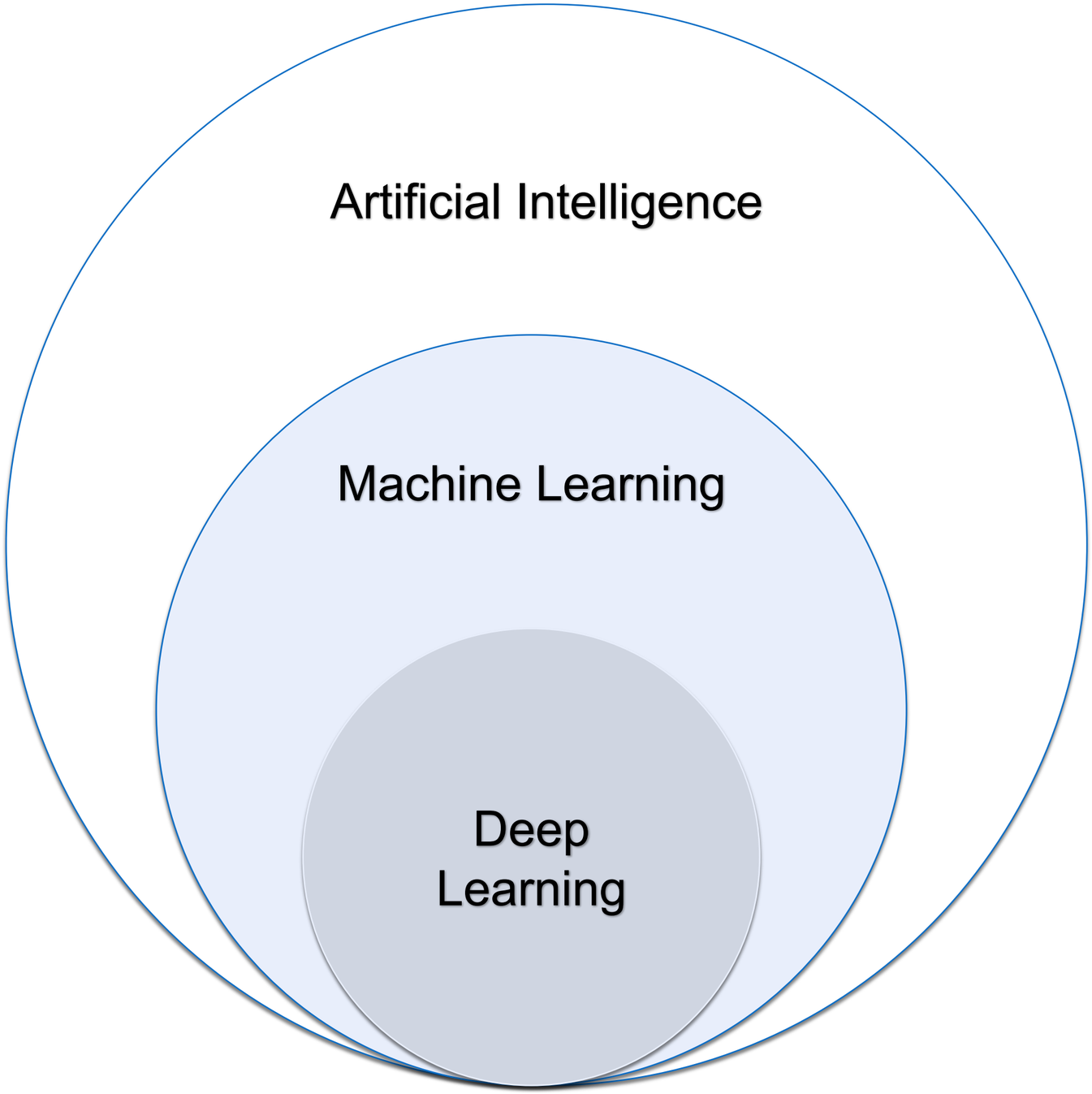
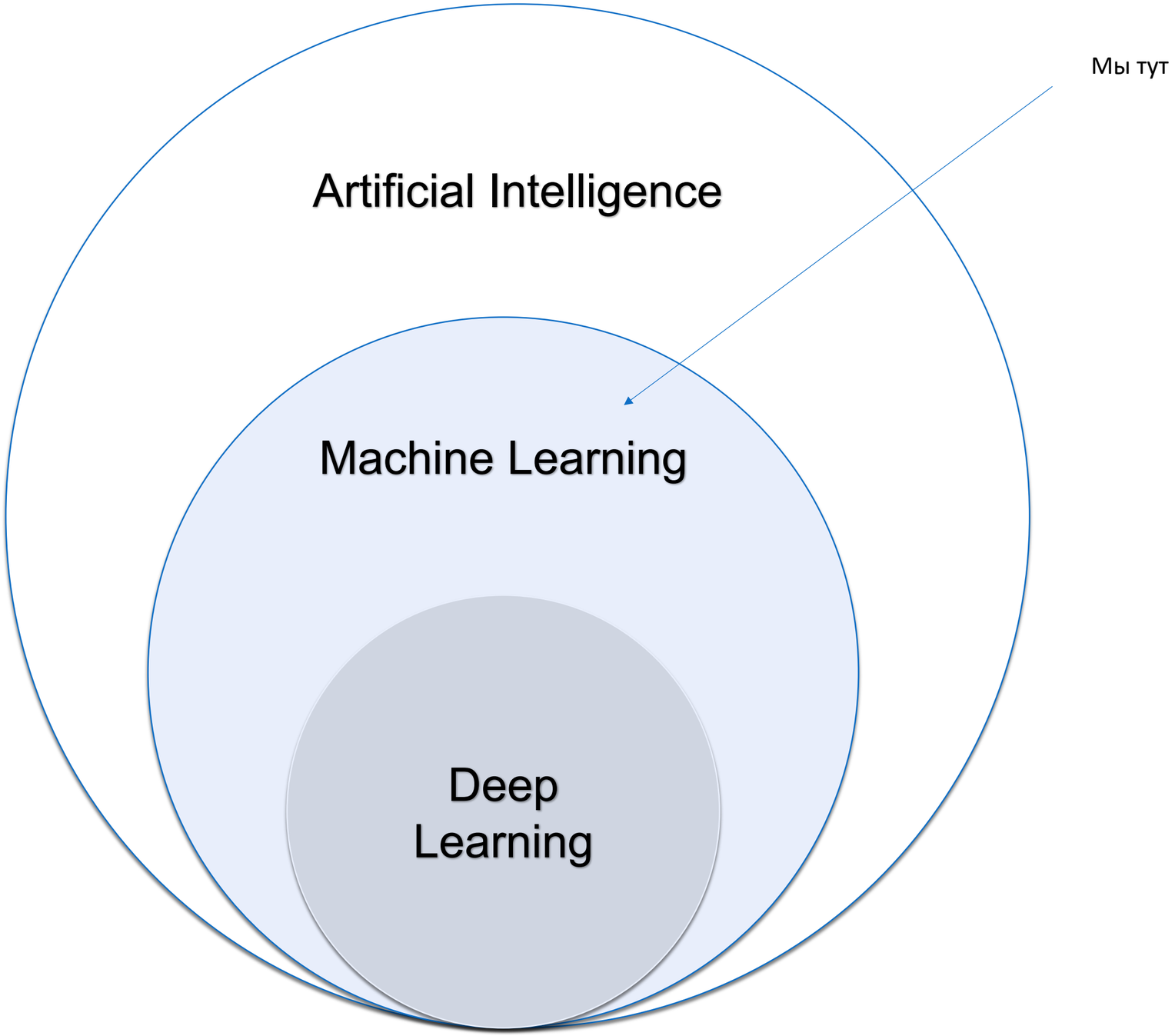
Машинное обучение

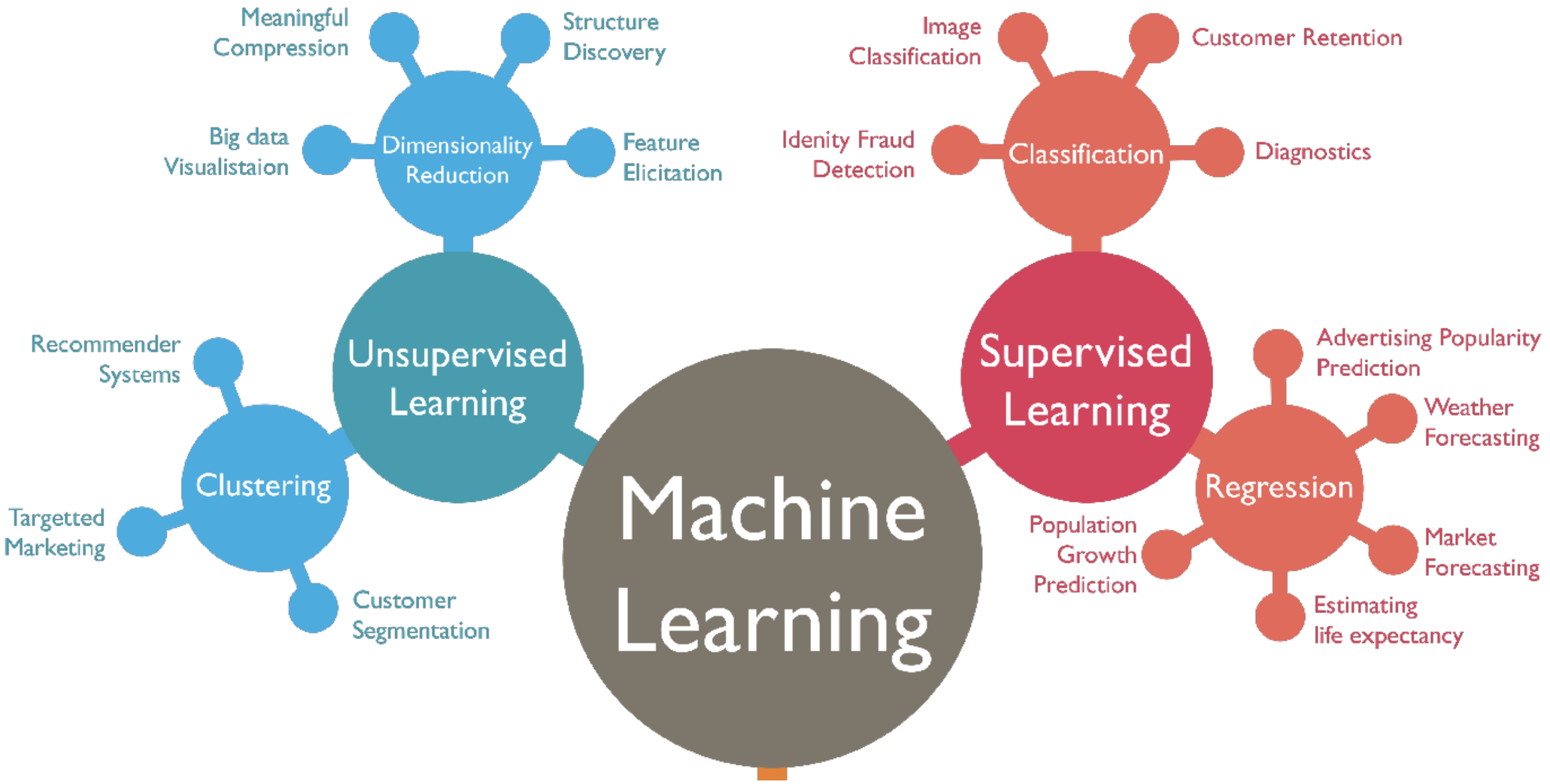
Занятие 5.
Композиции алгоритмов

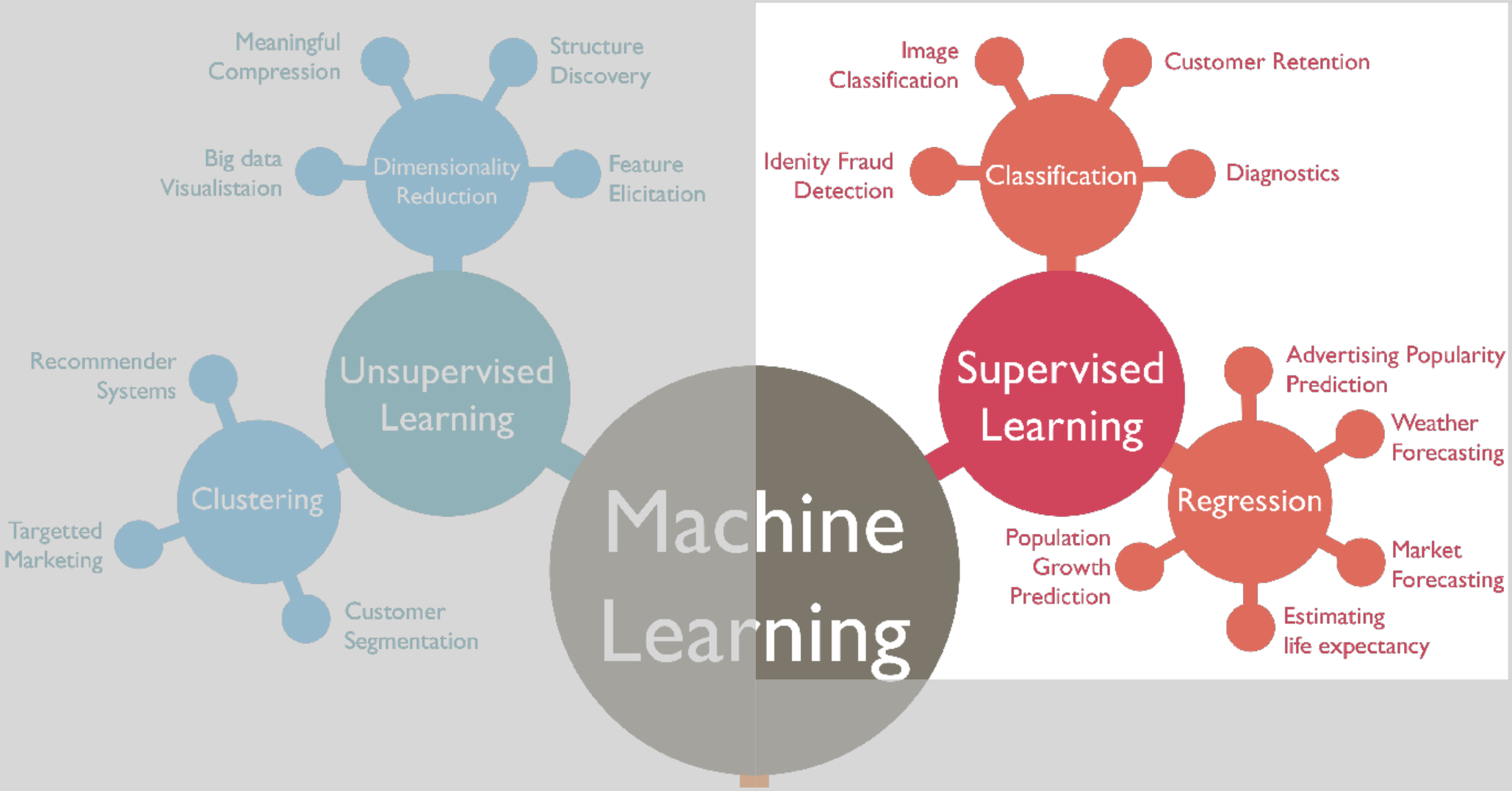
Краткое содержание

предыдущих серий


Artificial Intelligence
Machine Learning
Deep
Learning


Мы тут
Artificial Intelligence
Machine Learning
Deep
Learning


Machine Learning
Unsupervised Learning
Supervised Learning
Dimensionality Reduction
Clustering
Classification
Regression
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Recommender Systems
Targetted Marketing
Customer Segmentation
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy

Различие в задачах

Регрессия

- $y \in \mathbb{R}$

Классификация

- $y -$ класс/категория

Различие в задачах

Регрессия

- $y \in \mathbb{R}$
- Метрики:
  - MAE
  - MAPE
  - MSE/RMSE
  - R^2

Классификация

- $y - \text{класс/категория}$
- Метрики:
  - Accuracy
  - Precision
  - Recall
  - F1-мера
  - ROC-AUC

Различие в задачах

Регрессия

- $y \in \mathbb{R}$
- Метрики:
  - MAE
  - MAPE
  - MSE/RMSE
  - R^2
- Алгоритмы:
  - Линейная регрессия
  - Регресионное дерево

Классификация

- $y - \text{класс/категория}$
- Метрики:
  - Accuracy
  - Precision
  - Recall
  - F1-мера
  - ROC-AUC
- Алгоритмы:
  - Логистическая регрессия
  - Решающее дерево

Как работаем над моделями

- Фиксируем целевую переменную и класс задачи
- Фиксируем метрику для оценки качества
- Делаем EDA
- Разбиваем данные на train/test
- Учим и валидируем модели на train-куске
- Проверяем на test
- Сохраняем/выкатываем/тестируем

Вопросы из аудитории

Можно ли обойтись без математики?

Можно ли обойтись без математики?

- Нет ☹

Можно ли обойтись без математики?

- Нет ☹
- Матан:
  - Дифференциальное исчисление
- Линал
  - Операции над матрицами (сложение, умножение, транспон.)
  - Разложение матриц
- Статистика:
  - Распределения
  - Вероятности

«У меня не работает»

«У меня не работает»

- УРА!

«У меня не работает»

- УРА!
- Кидайте скрин ошибки -- я помогу

«Не работает cross_val_score»

«Не работает cross_val_score»

- Следите, что передаете
  - в model
  - в scoring

Если модель классификации, то и метрики – классификации

Если регрессии, то метрики регресии.

См. слайд 8

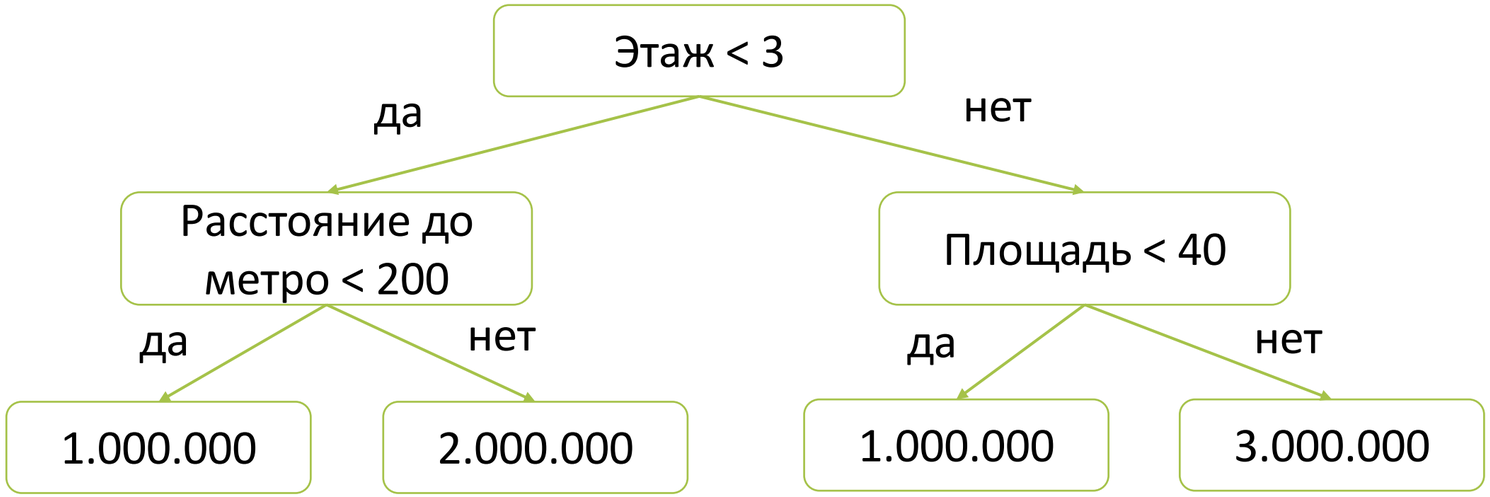
«Почему cross_val_score выдает отр. MAE»

«Почему cross_val_score выдает отр. MAE»

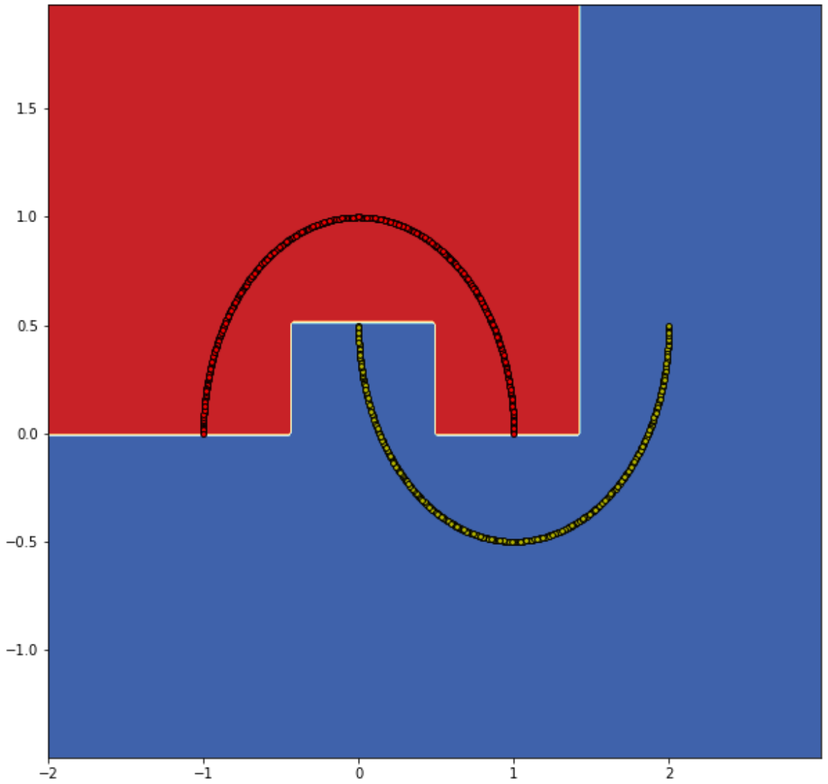
- В переменную scoring передаем neg_mean_absolute_error и neg_mean_squared_error
- cross_val_score считает, что чем больше, тем лучше!
- А MAE и MSE, наоборот, чем меньше, тем лучше!
- Тогда –MAE и –MSE чем больше, тем лучше!

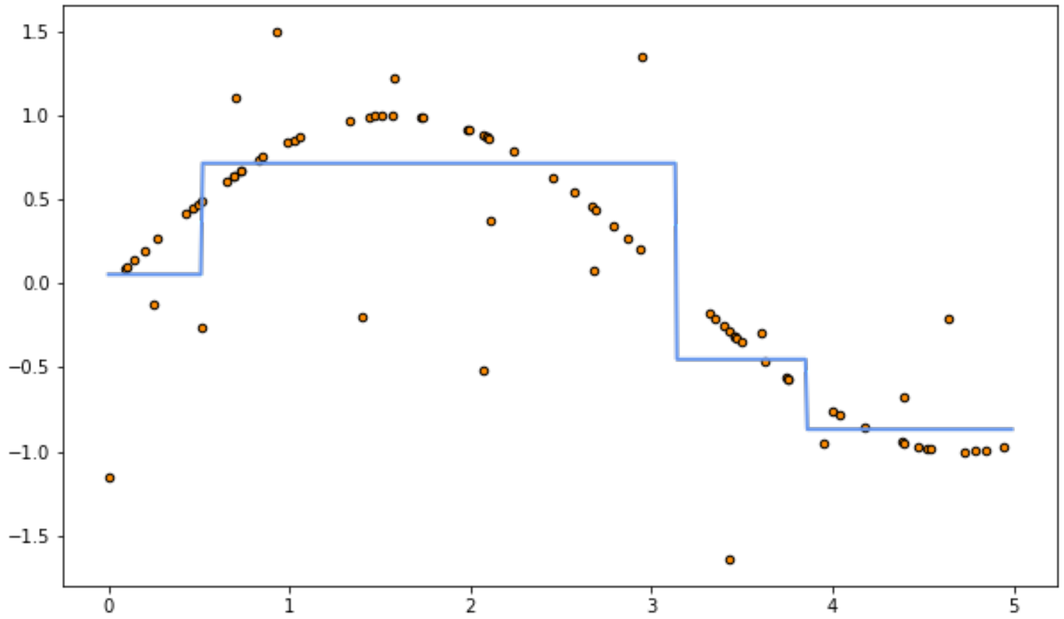
Напоминание о деревьях

Решающее дерево

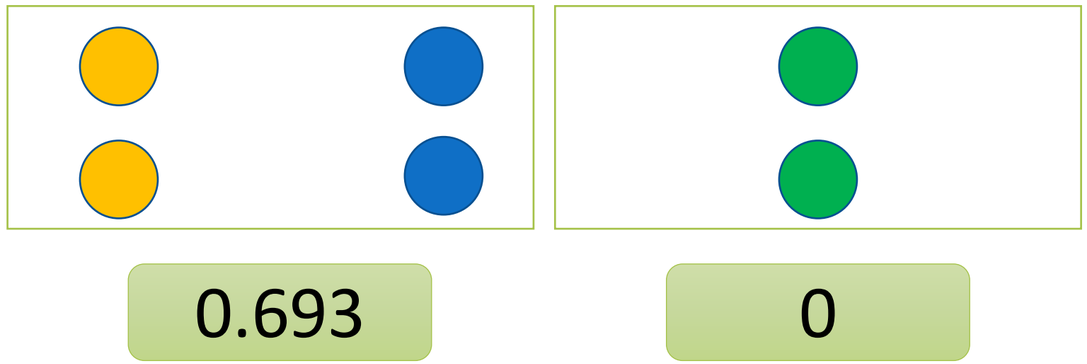


Решающие деревья

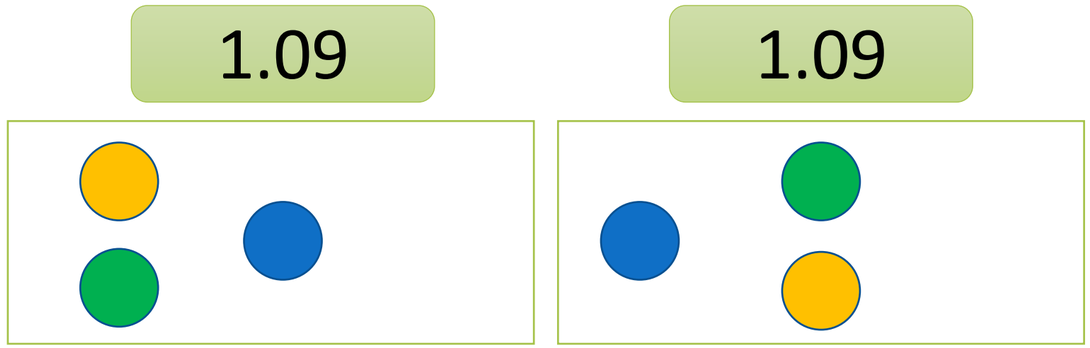




Классификация. Как сравнить разбиения?



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



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

Варианты:

- Энтропия

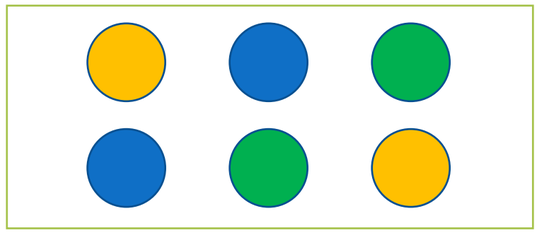
$$H(p_1, \dots, p_K) = -\sum_{i=1}^{K} p_i \log_2 p_i$$

- Джинни

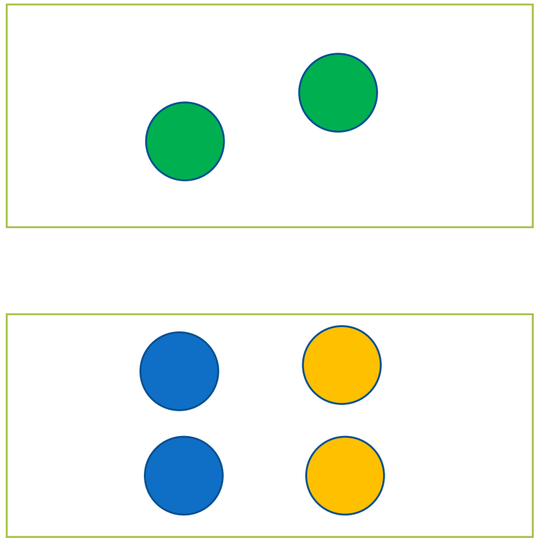
$$H(p_1, \dots, p_K) = \sum_{i=1}^{K} p_i \, (1 - p_i)$$

Критерий информативности

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



против



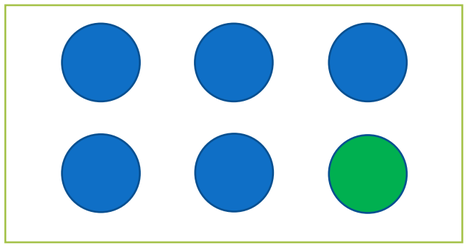
Критерий информативности

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

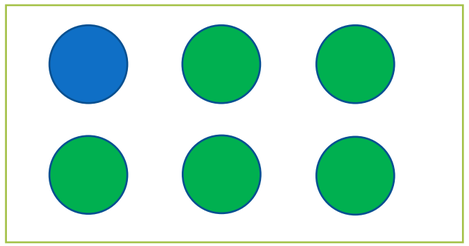
- Или так:

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

Как сравнить разбиения?

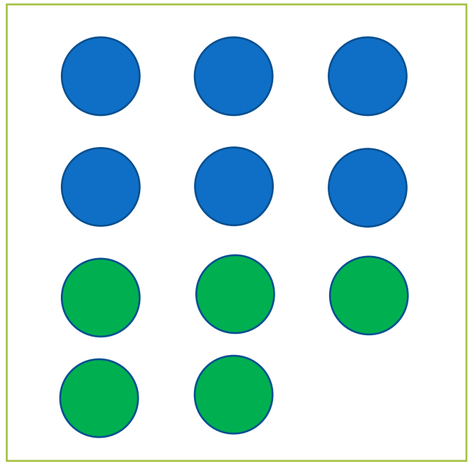


0.65

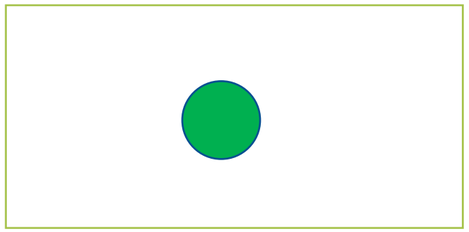


0.65

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



0.994



0

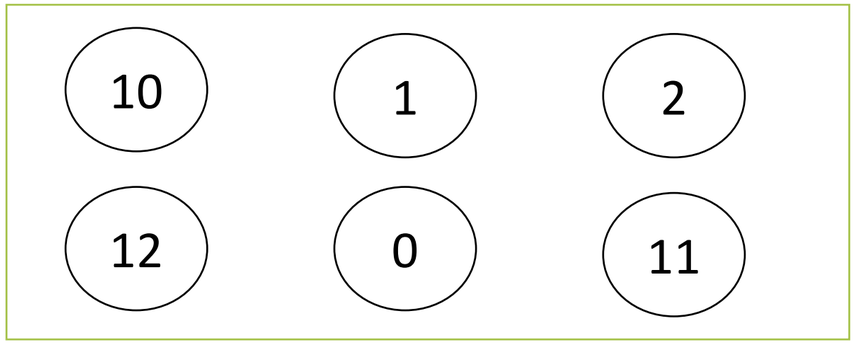
- $(6/11, 5/11)$ и $(0, 1)$
- $\frac{11}{12} * 0.994 + \frac{1}{12} * 0 = 0.911$

Регрессия. Как сравнить разбиения?

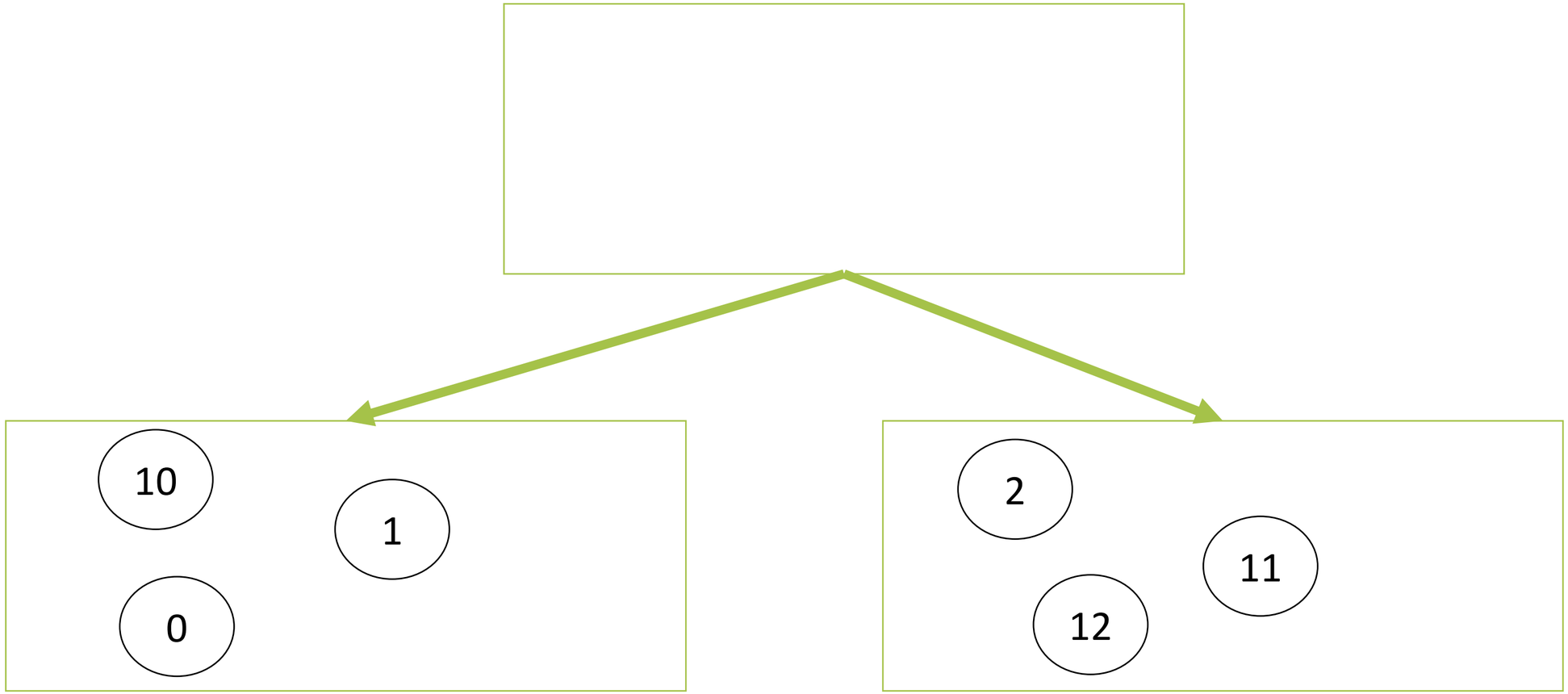
$$H(R) = \frac{1}{|R|} \sum_{(x_i, y_i) \in R} (y_i - y_R)^2$$

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

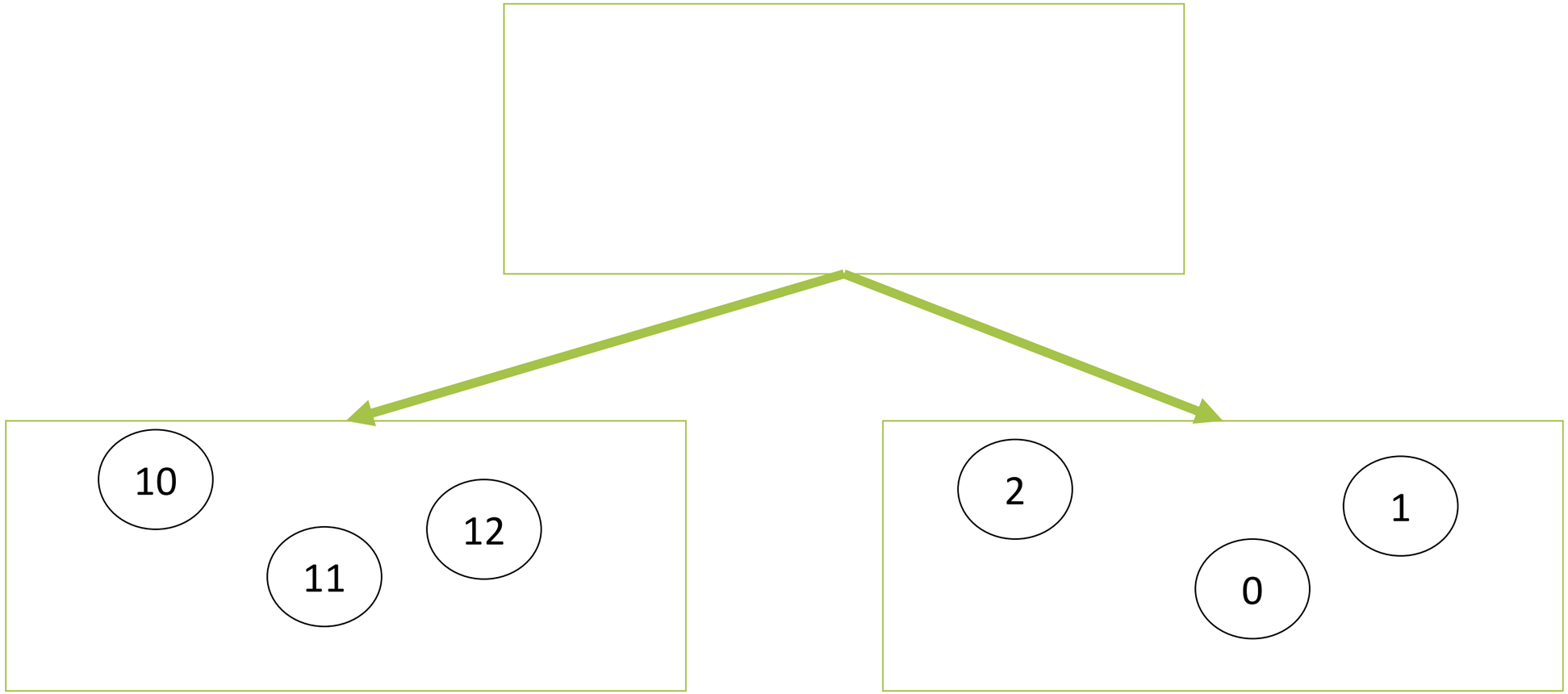
А для регрессии?



А для регрессии?



А для регрессии?

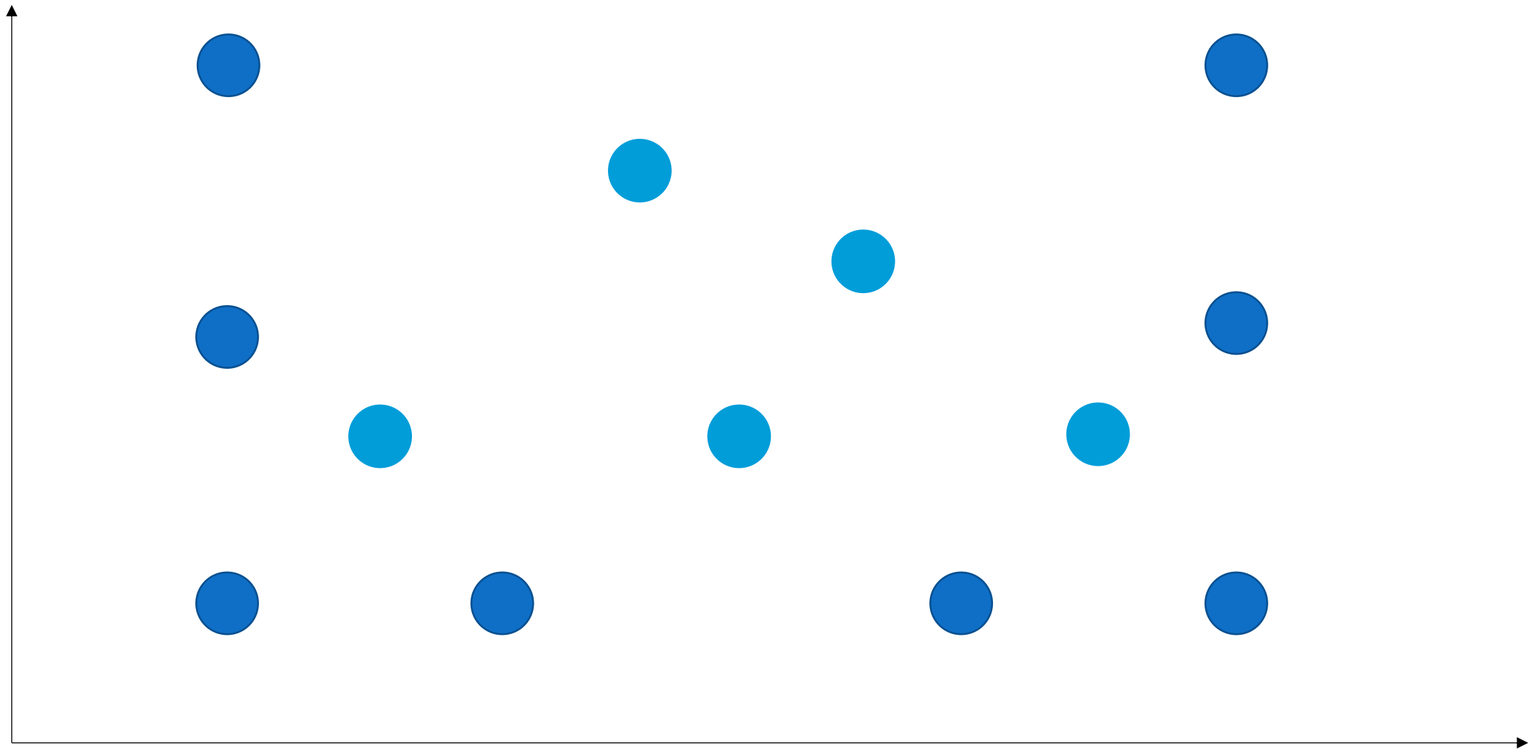


Жадный алгоритм

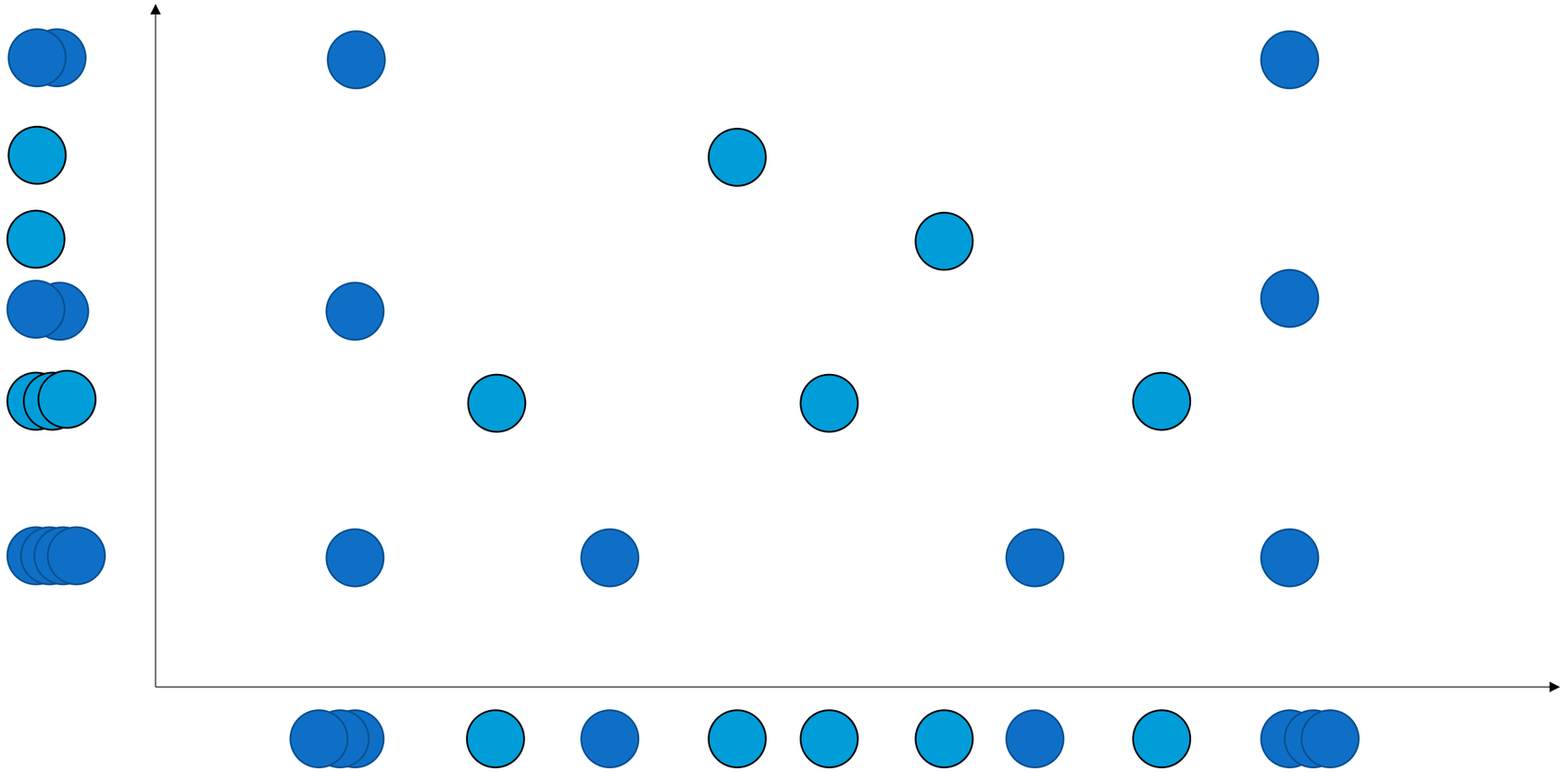
- $\mathrm{SplitNode}(m, R_m)$

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

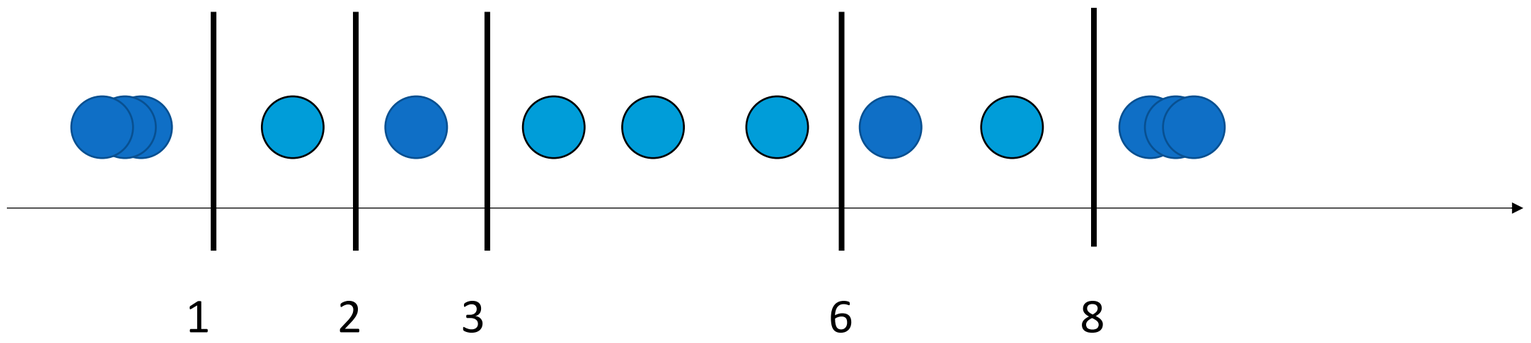
Обучение деревьев



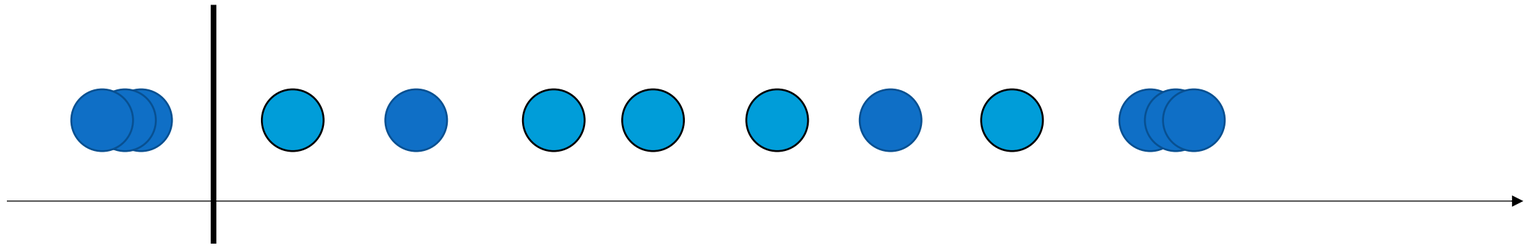
Признаки



Разбиения по признаку 1



Разбиения по признаку 1

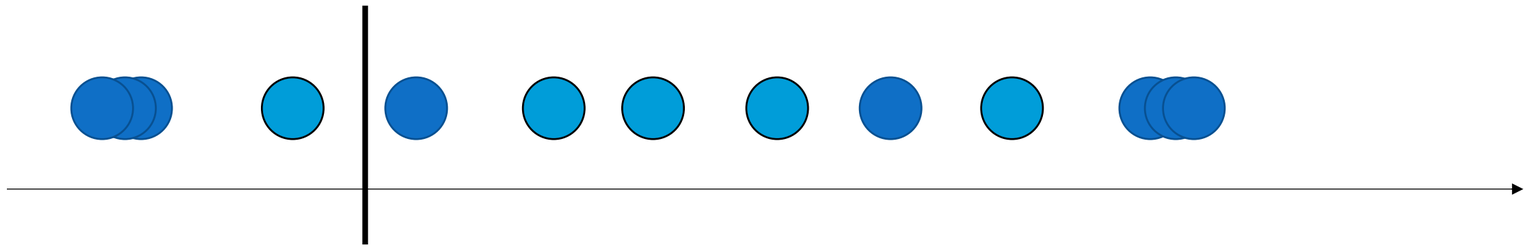


(1, 0)
H(p) = 0

(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

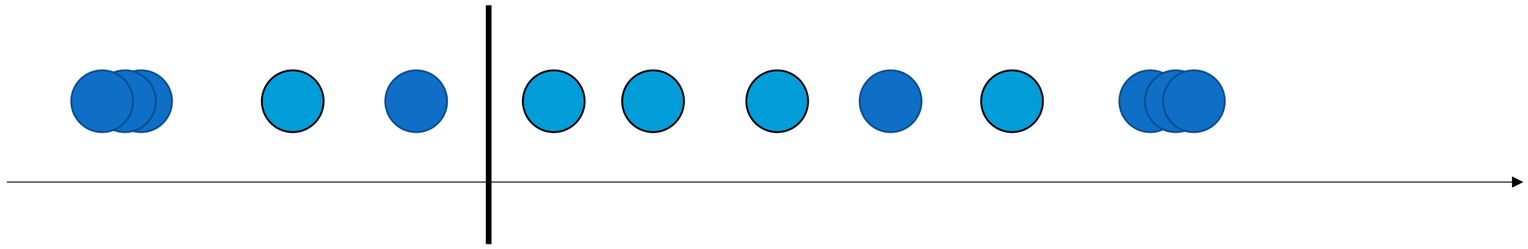


(3/4, 1/4)
H(p) = 0.56

(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

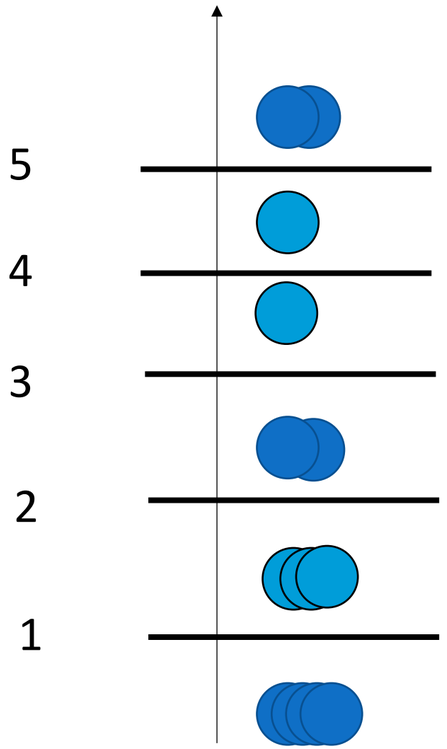


(4/5, 1/5)
H(p) = 0.5

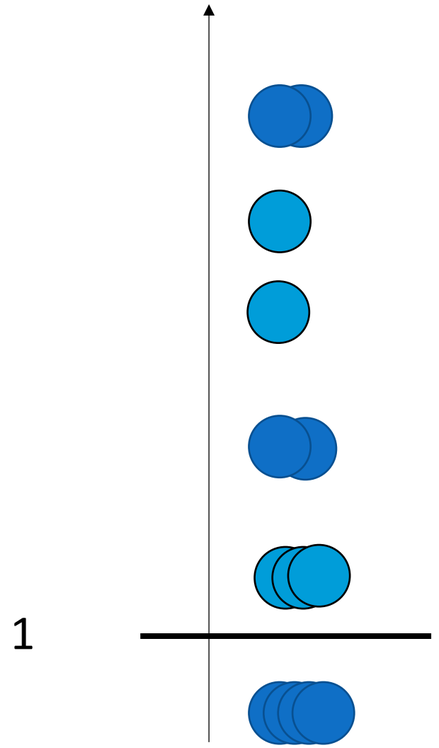
(1/2, 1/2)
H(p) = 0.69

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

Разбиения по признаку 2



Разбиения по признаку 2

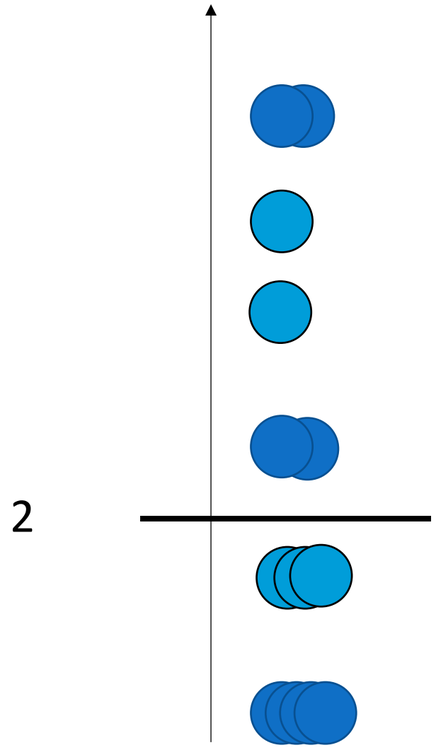


(4/9, 5/9)
H(p) = 0.69

(1, 0)
H(p) = 0

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

Разбиения по признаку 2

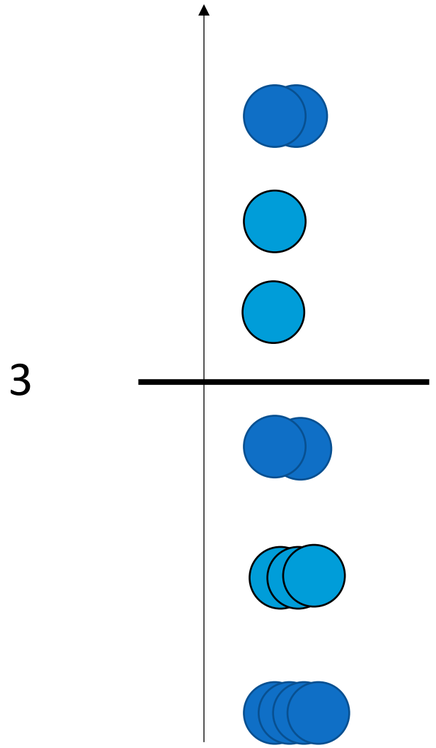


(4/6, 2/6)
H(p) = 0.64

(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

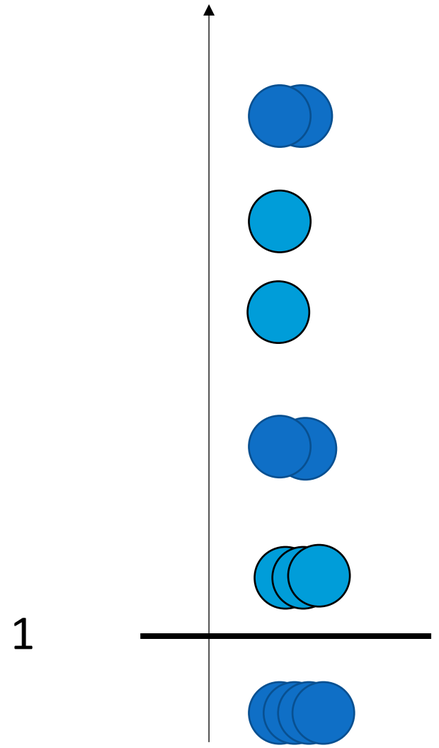


(1/2, 1/2)
H(p) = 0.69

(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



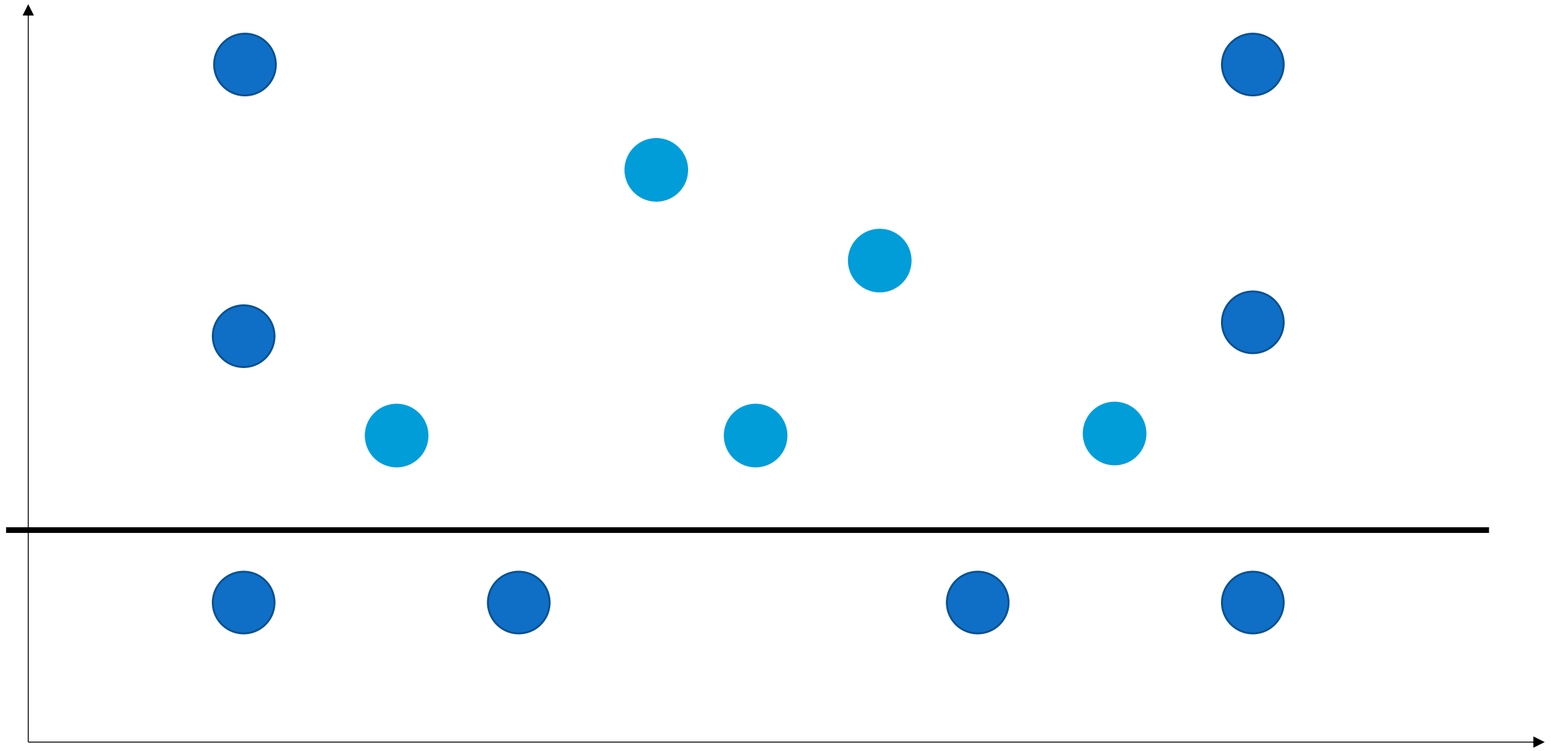
(4/9, 5/9)
H(p) = 0.69

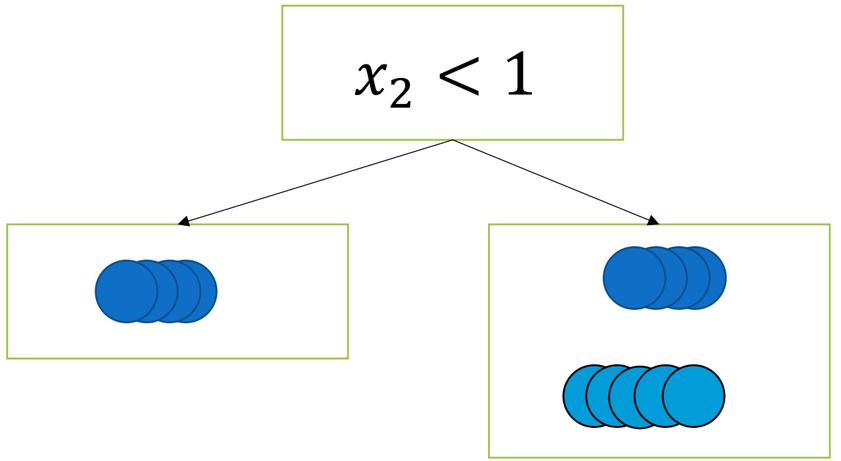
(1, 0)
H(p) = 0

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

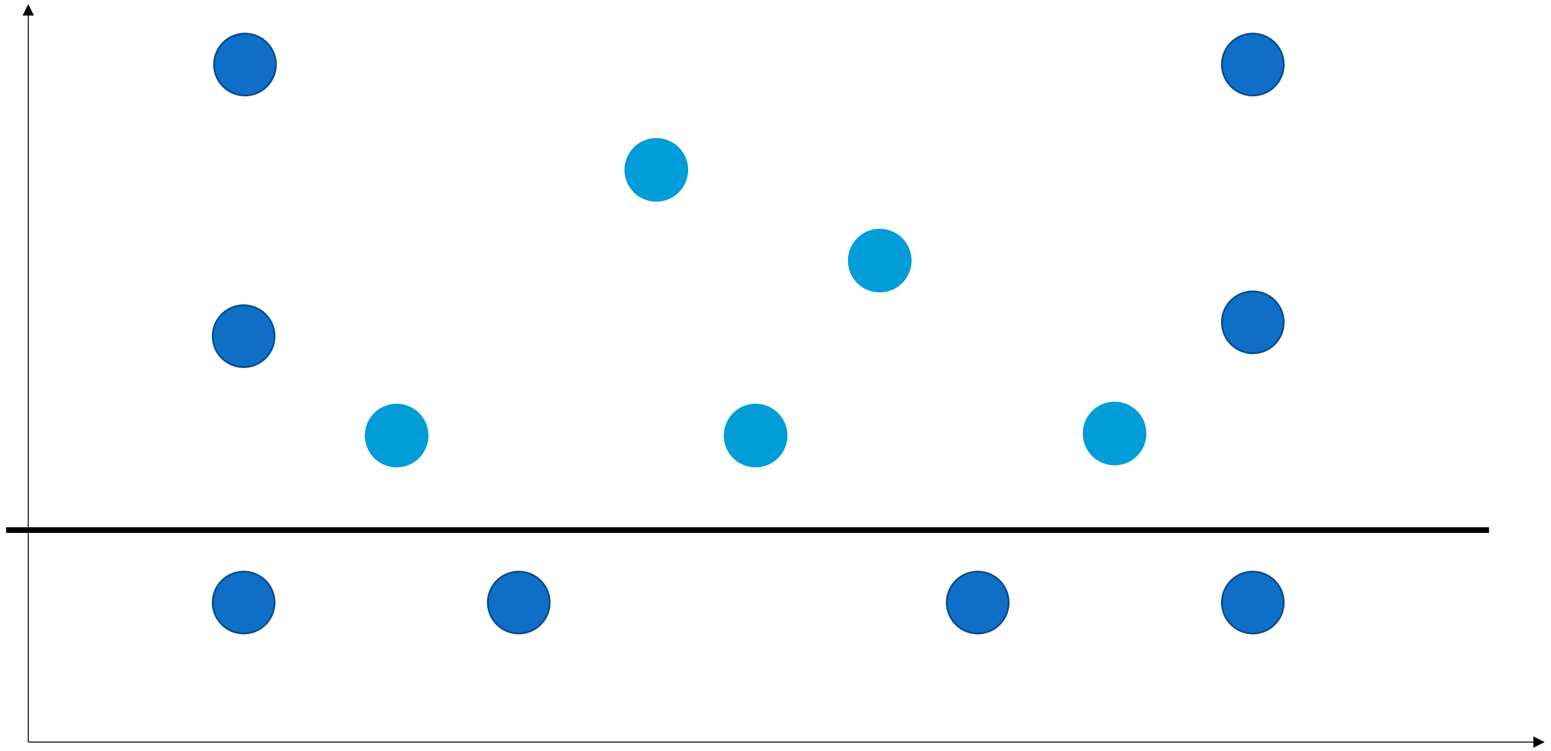
Лучшее разбиение!

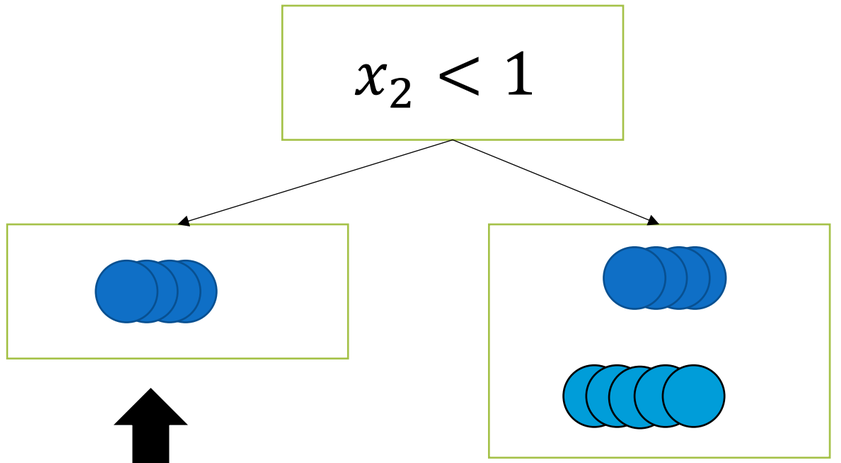
Обучение деревьев





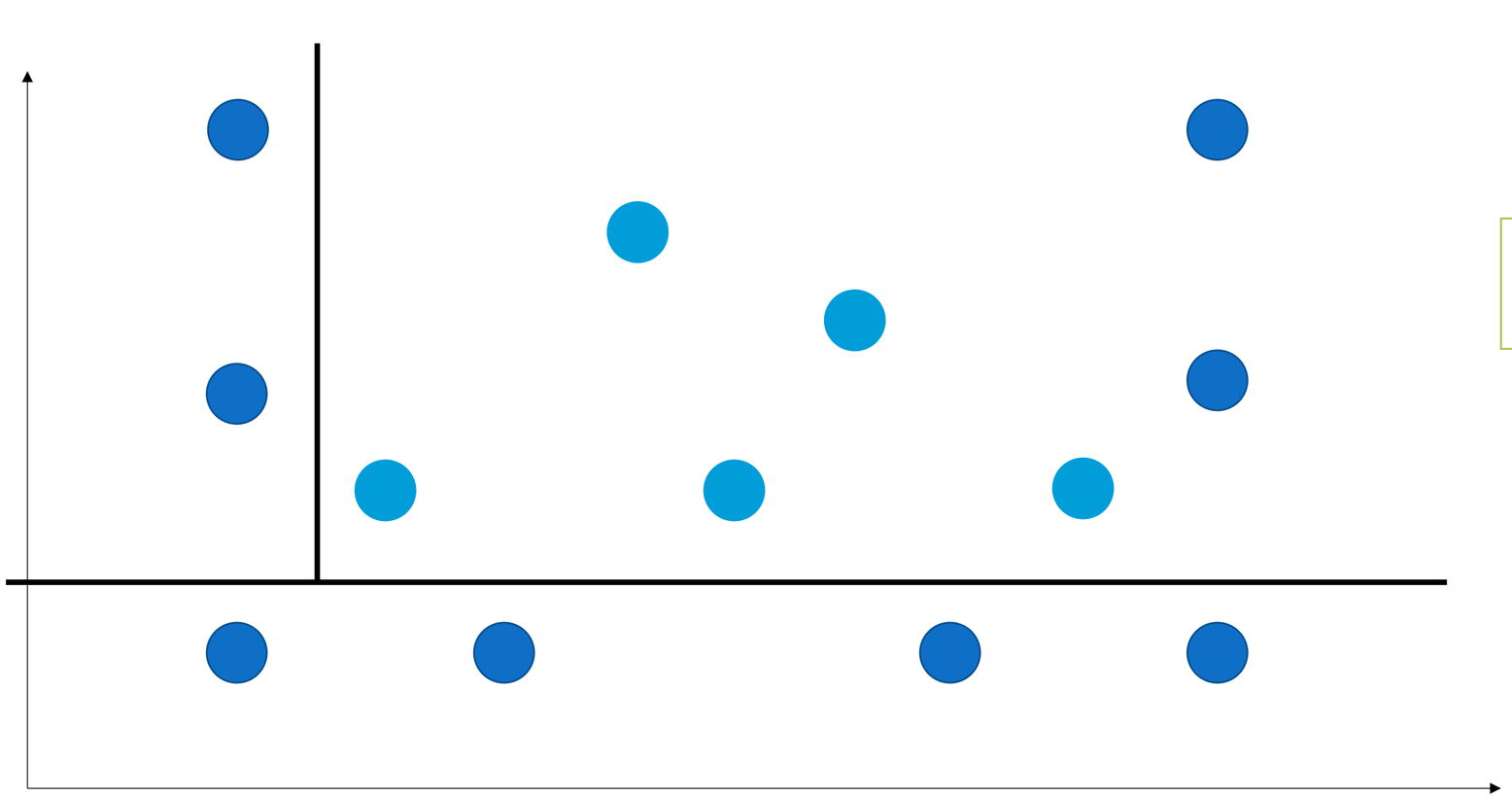
Обучение деревьев

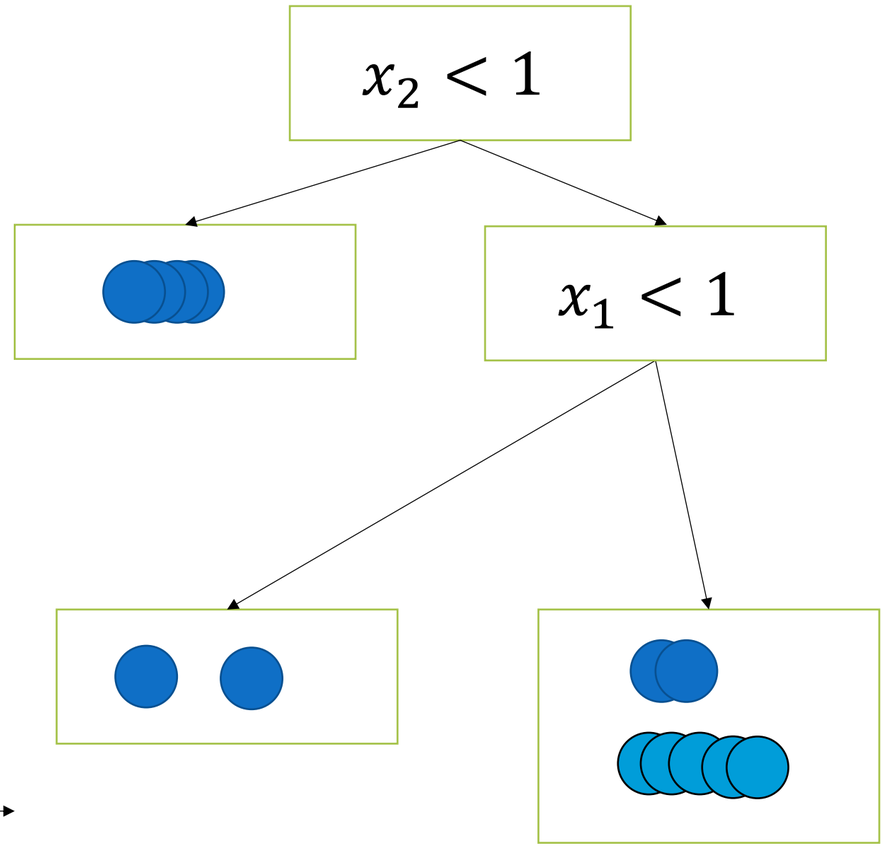




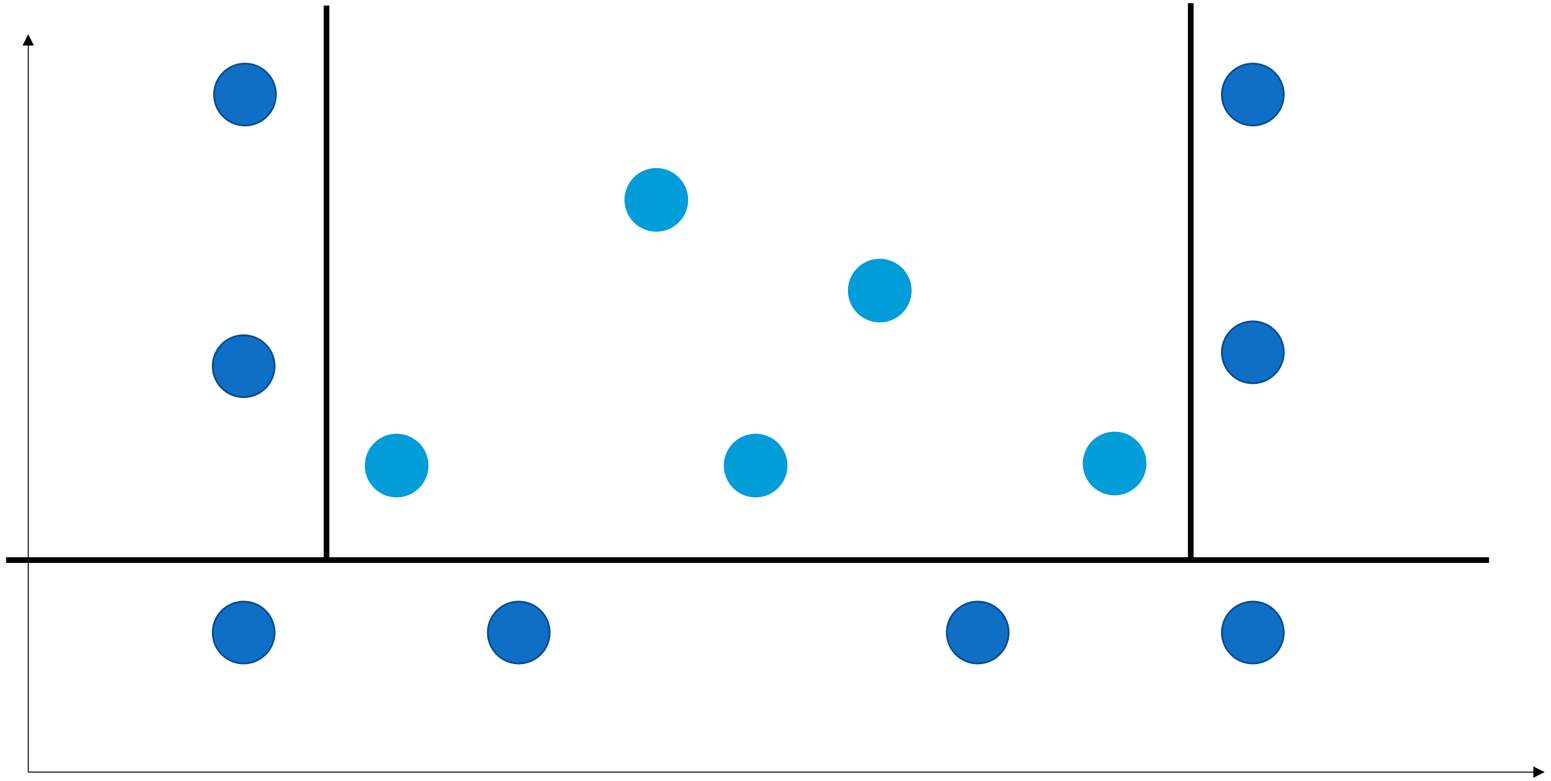
Листовая вершина

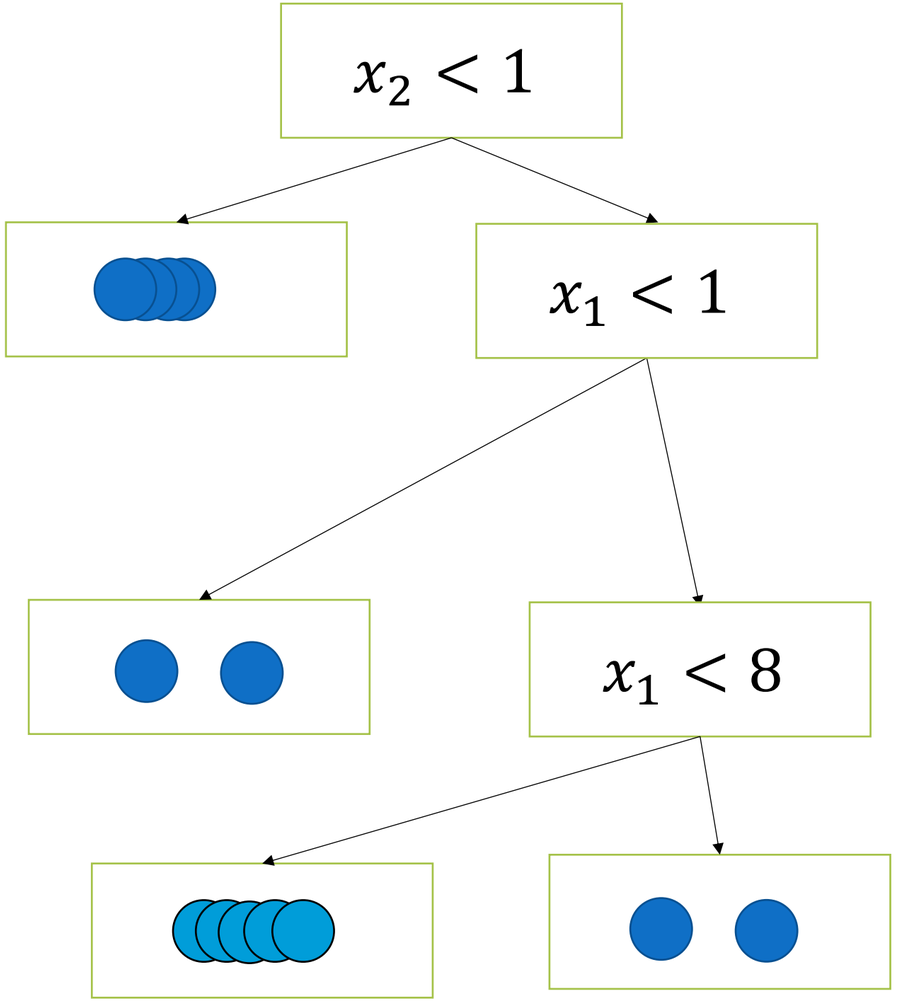
Обучение деревьев





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

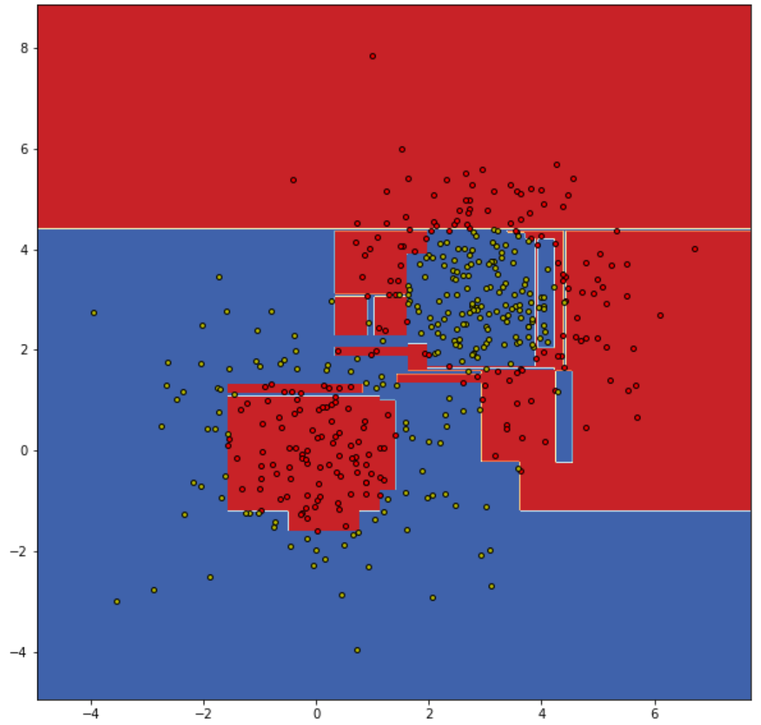
Устойчивость моделей

- $X = (x_i, y_i)_{i=1}^{\ell}$ — 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

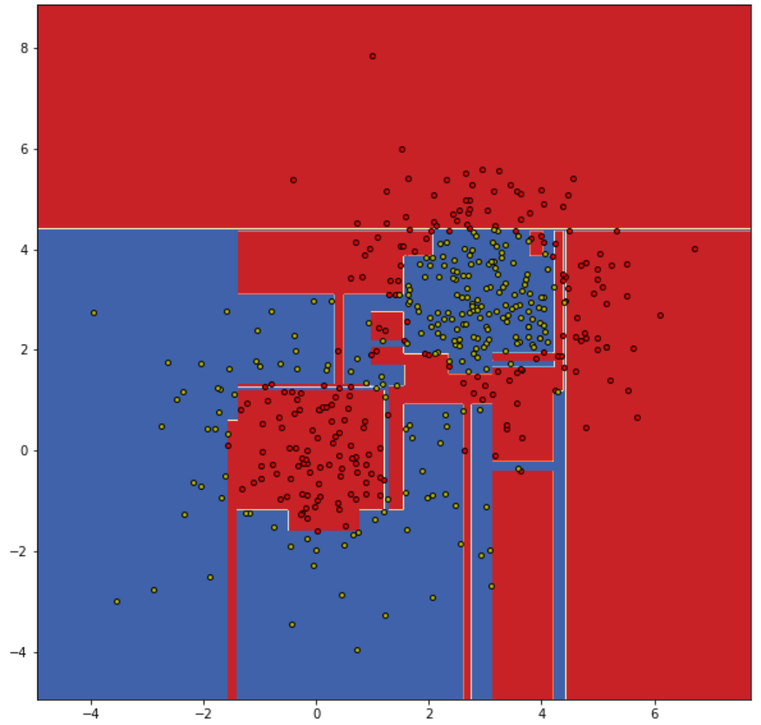
Устойчивость моделей

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

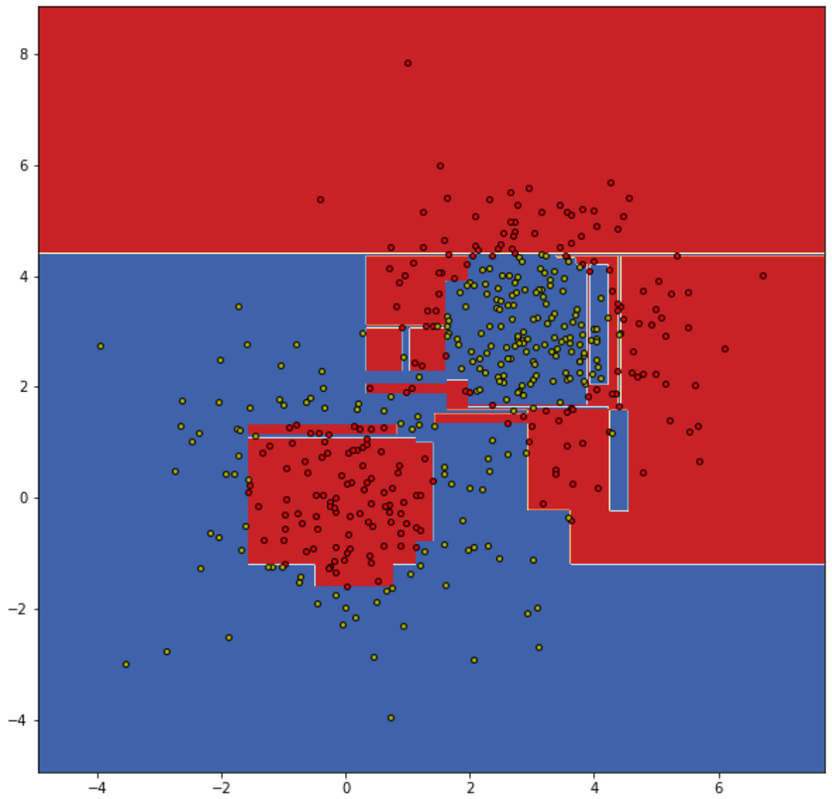
Обучение на подвыборках

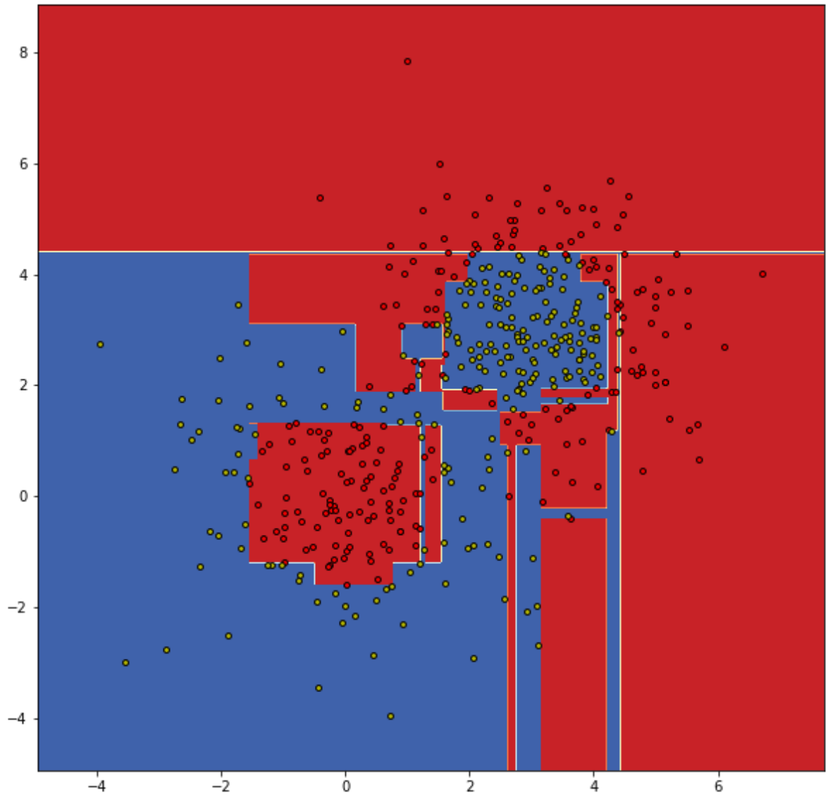


Обучение на подвыборках



Обучение на подвыборках



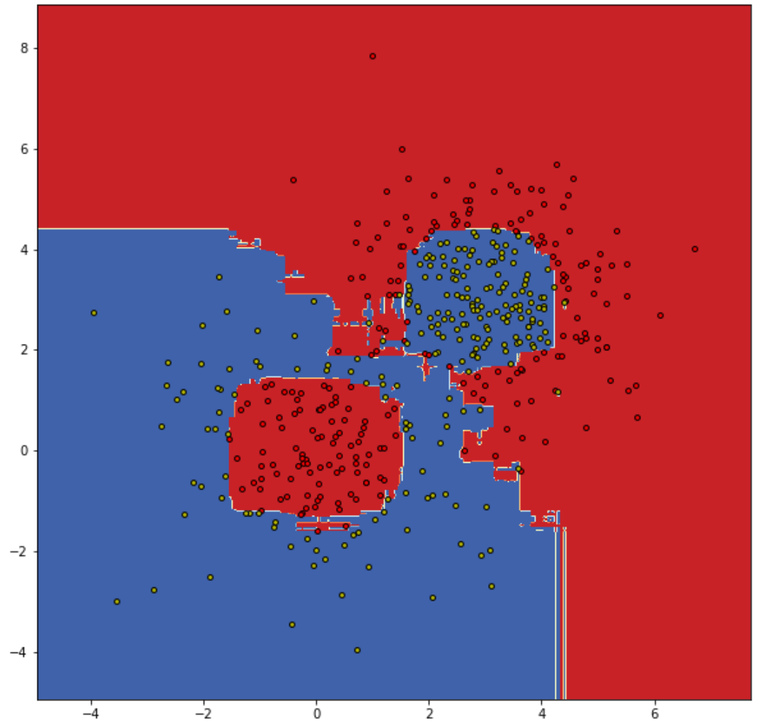


Композиция моделей

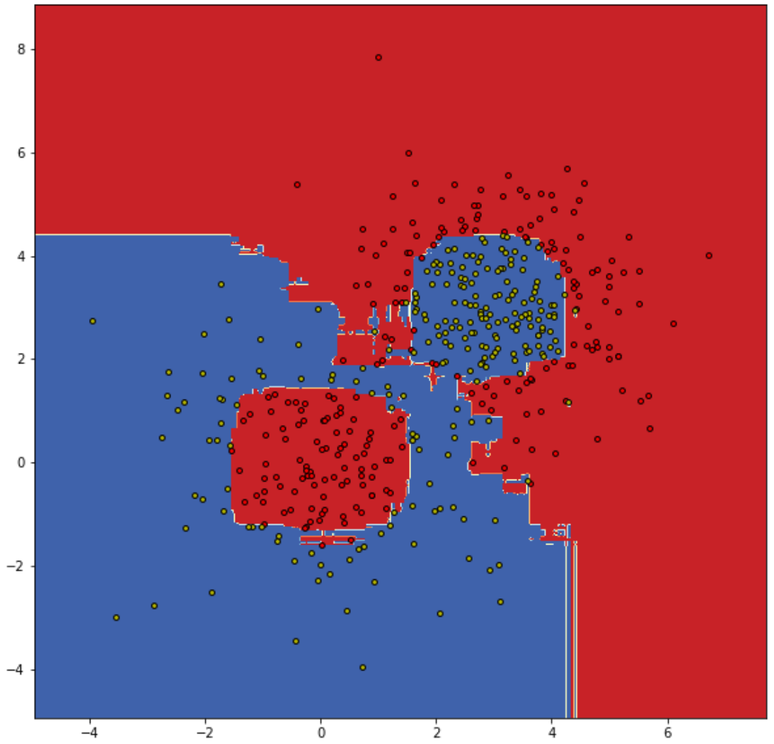
- У нас получилось N деревьев: $b_1(x), \ldots, b_N(x)$
- Объединим их через голосование большинством (majority vote):

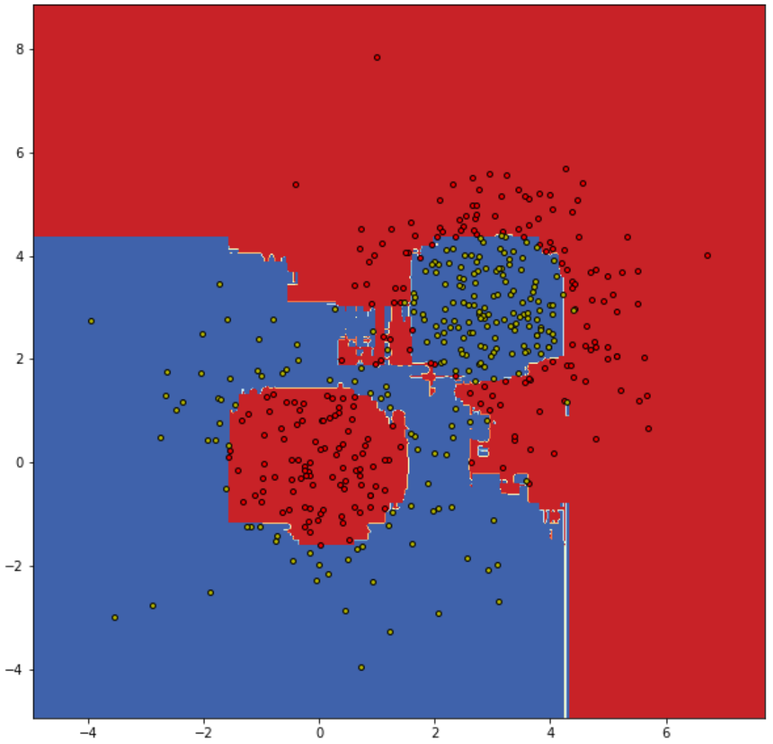
$$a(x) = \arg\max_{y \in \mathbb{Y}} \sum_{n=1}^{N} [b_n(x) = y]$$

Композиция моделей



Композиция моделей

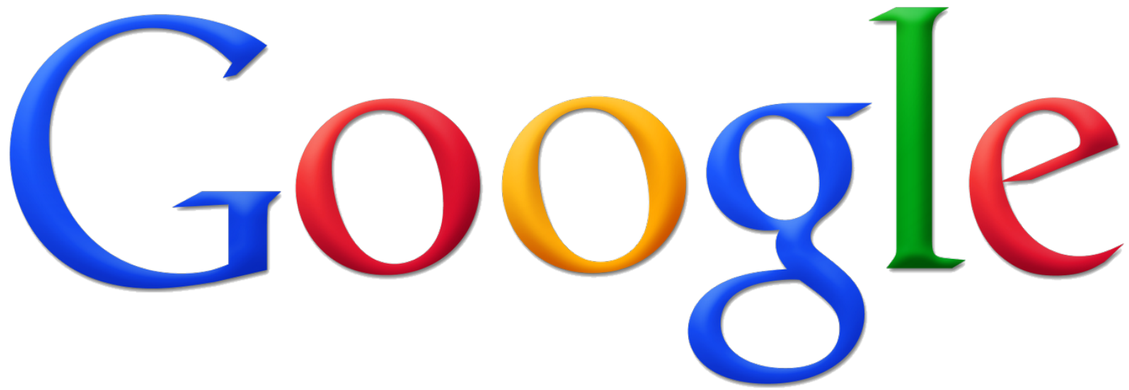




Голосование по большинству и усреднение

Majority vote

- Какой из двух логотипов более старый?



Majority vote

- 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

Усреднение наблюдений

- Сколько метров в 1 сажени?

Усреднение наблюдений

- Сколько всего стран в мире?

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

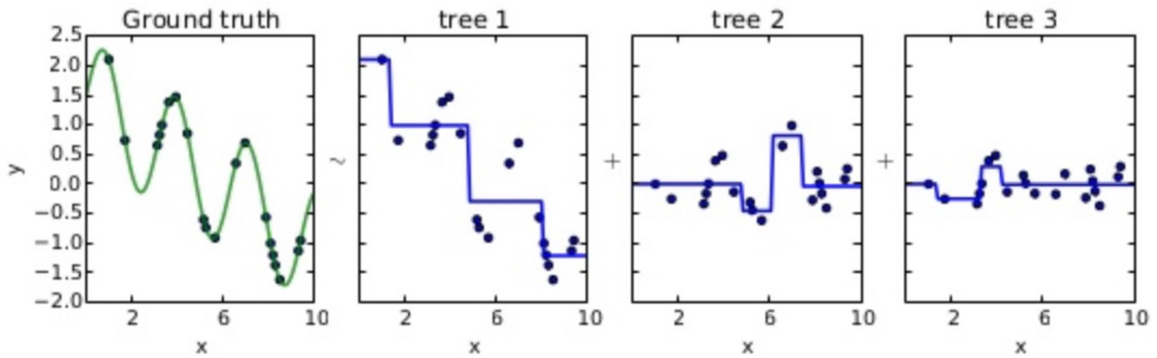
$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

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

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

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

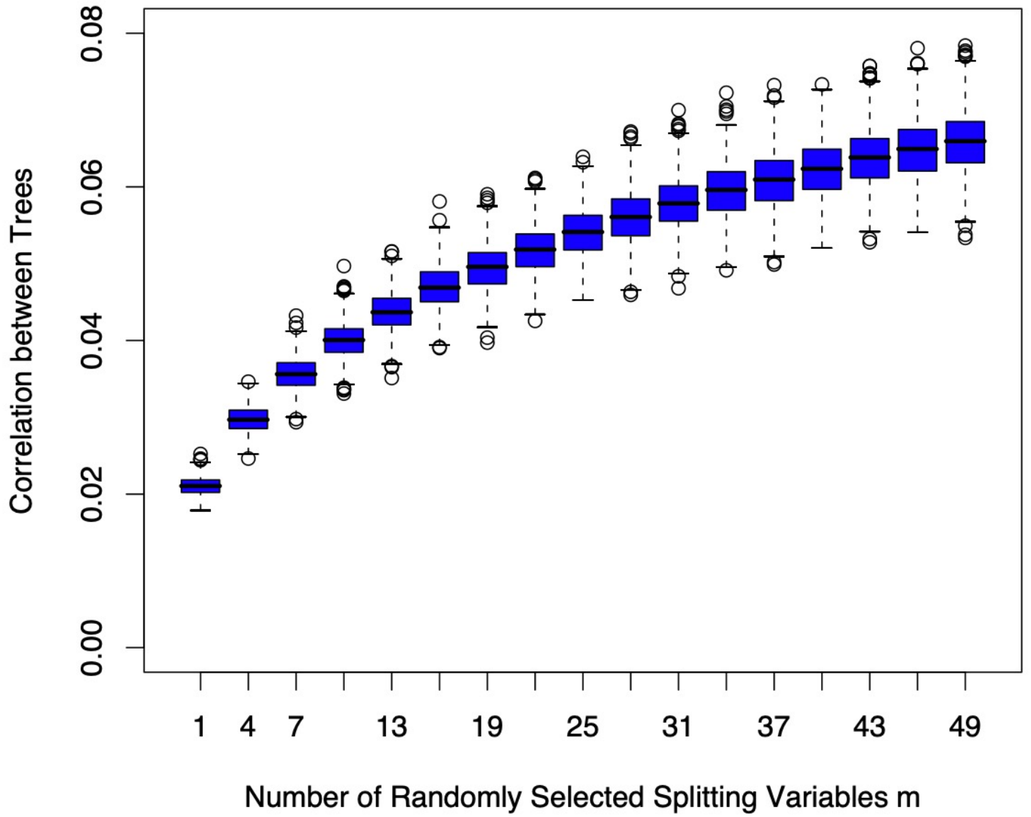
1. Если выполнен критерий останова, то выход
2. Ищем лучший предикат: $j, t = \arg\min_{\mathrm{j,t}} Q(R_m, j, t)$
3. Разбиваем с его помощью объекты: $R_\ell = \left\{\{(x, y) \in R_m | \left[x_j < t\right]\right\}$, $R_r = \left\{\{(x, y) \in R_m | \left[x_j \geq t\right]\right\}$
4. Повторяем для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

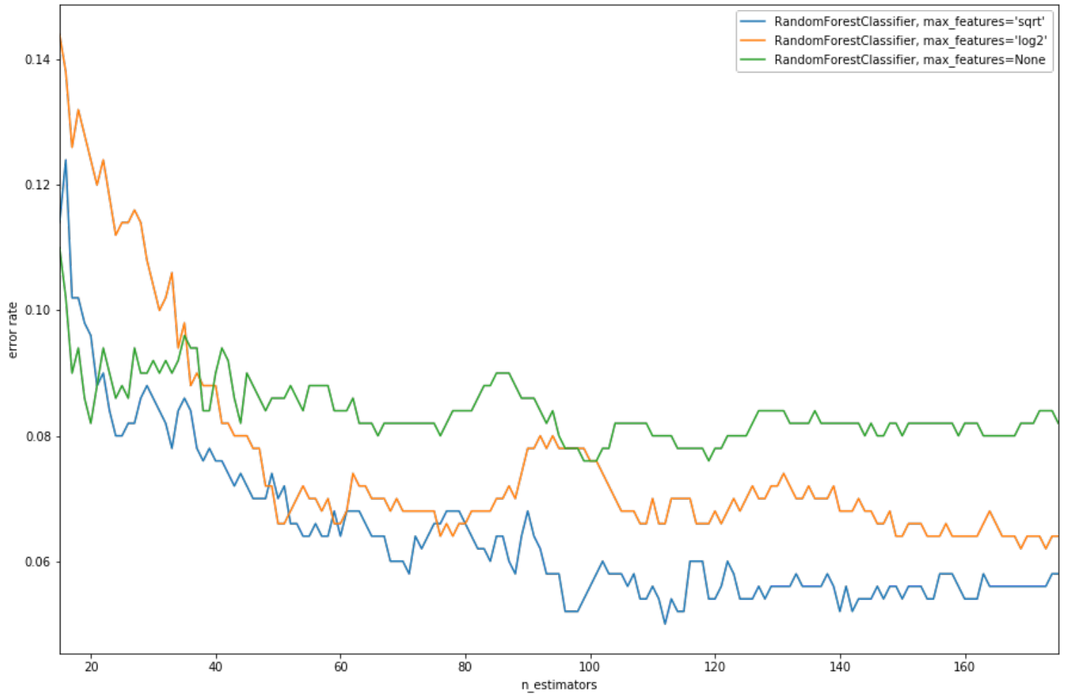
$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$$

- 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



Резюме

- Случайный лес — метод на основе бэггинга, в котором делается попытка повысить разнообразие деревьев
- Метод практически без гиперпараметров

Градиентный бустинг

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:

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

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 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 логистической функции потерь

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

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

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

Гиперпараметры

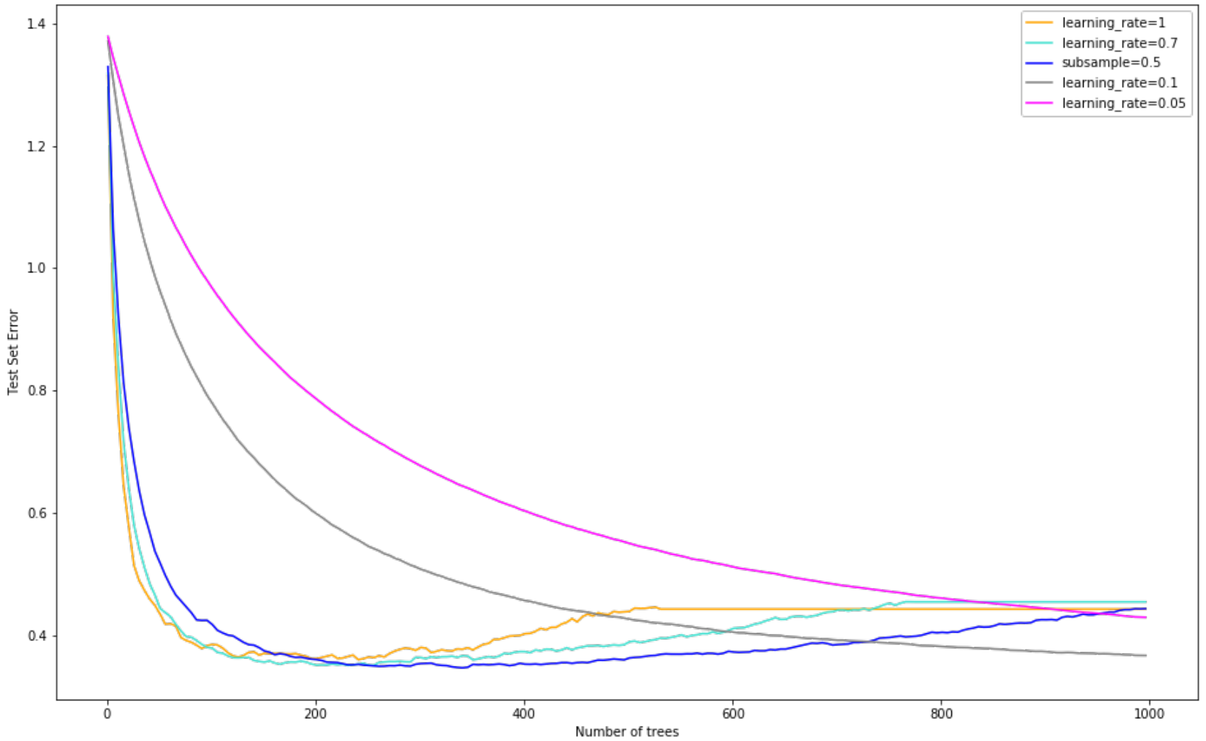
- Глубина базовых деревьев
- Число деревьев N

Длина шага

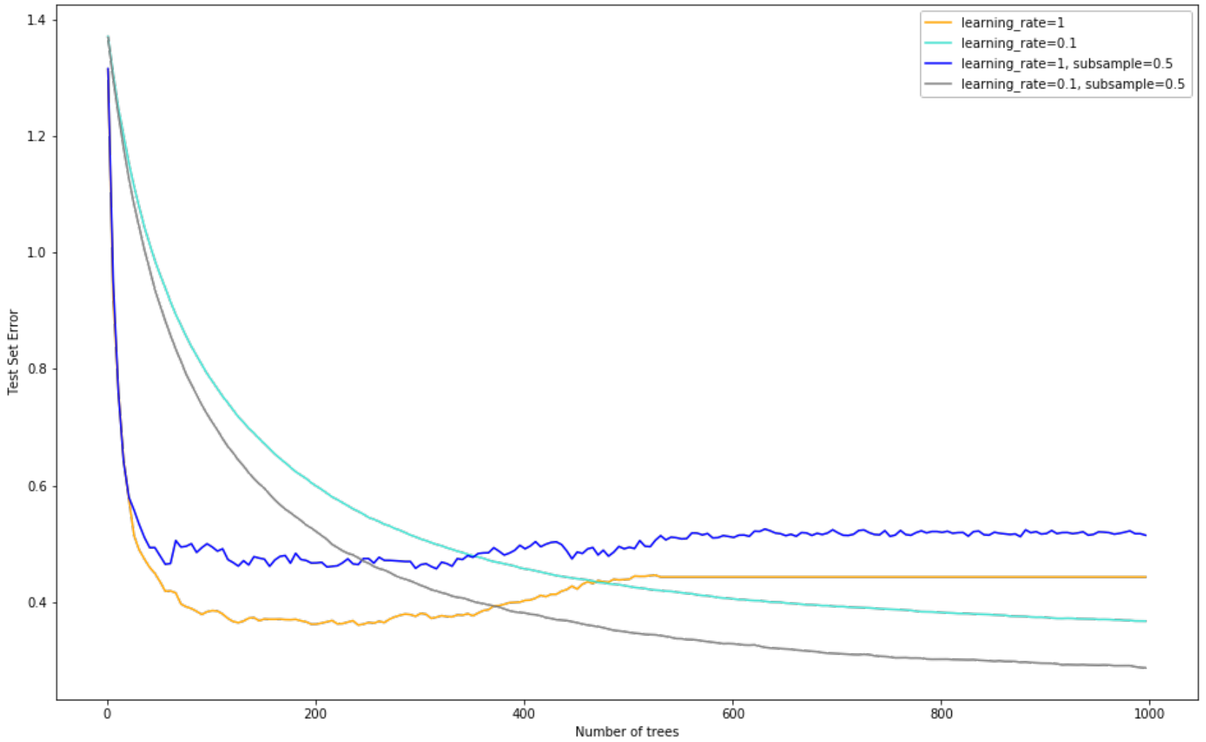
$$a_N(x) = a_{N-1}(x_i) + \eta b_N(x_i)$$

- $\eta \in (0, 1]$ — длина шага
- Можно сказать, что это регуляризация композиции
- Снижает вклад каждой модели в композицию
- Чем меньше η, тем больше надо деревьев

Длина шага


learning_rate=1
learning_rate=0.7
subsample=0.5
learning_rate=0.1
learning_rate=0.05
Test Set Error
Number of trees
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

Рандомизация



Гиперпараметры

- Глубина базовых деревьев
- Число деревьев N
- Длина шага
- Размер подвыборки для обучения
- и т.д.

Резюме

- Чтобы снизить переобучение, можно добавлять модели в композицию с небольшими весами
- Также может помочь обучение моделей на подвыборках

Last Not Least

- LightGMB на MacOS требует libomp
- Если нет инструментов разработчика, надо ставить
- ☹