МГТУ
им. Н.Э. Баумана
ТЕХНОПАРК
Mail.Ru Group

# ПРОГРАММИРОВАНИЕ CUDA C/C++, АНАЛИЗ ИЗОБРАЖЕНИЙ И DEEP LEARNING

Лекция №6

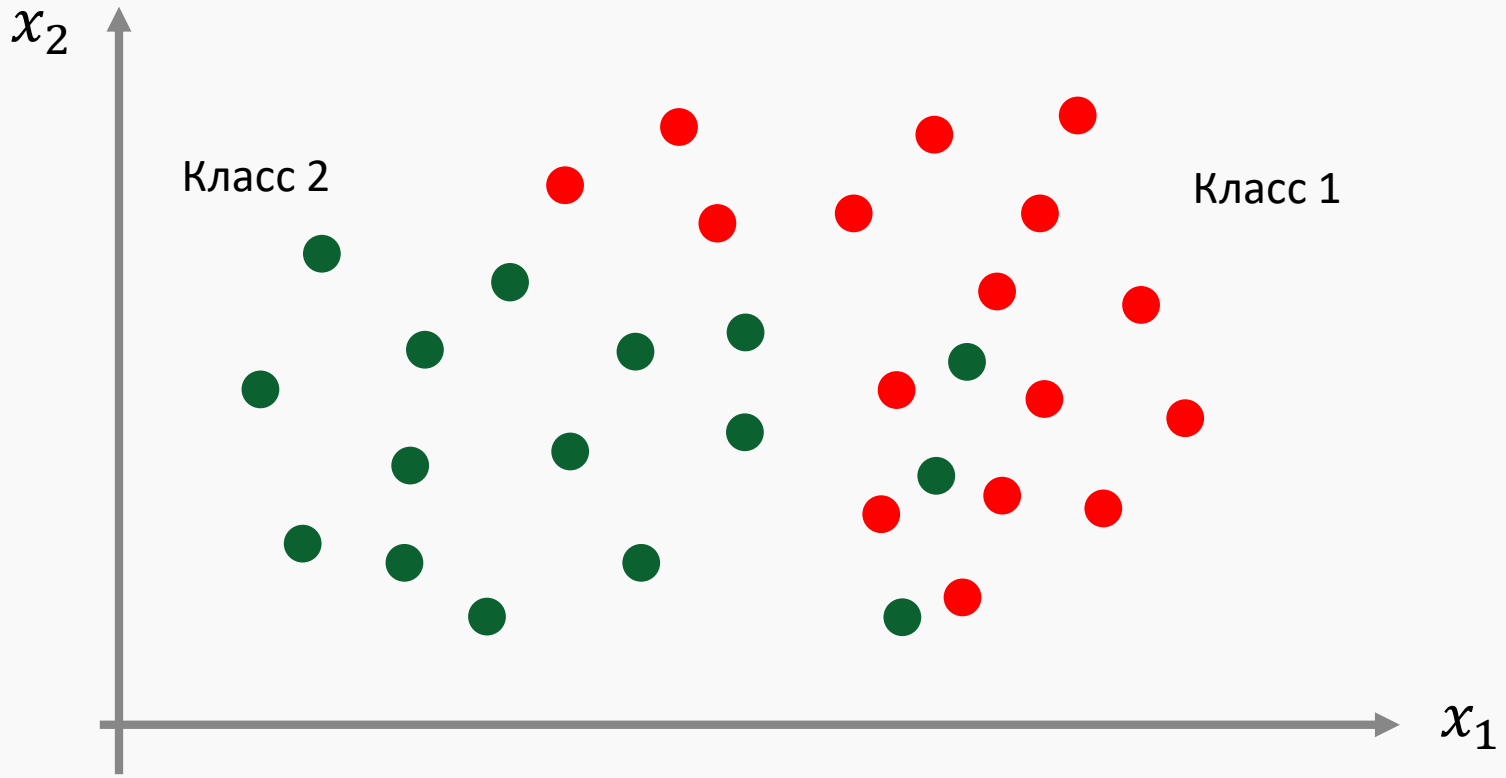
Спасёнов Алексей

# Введение в Машинное обучение

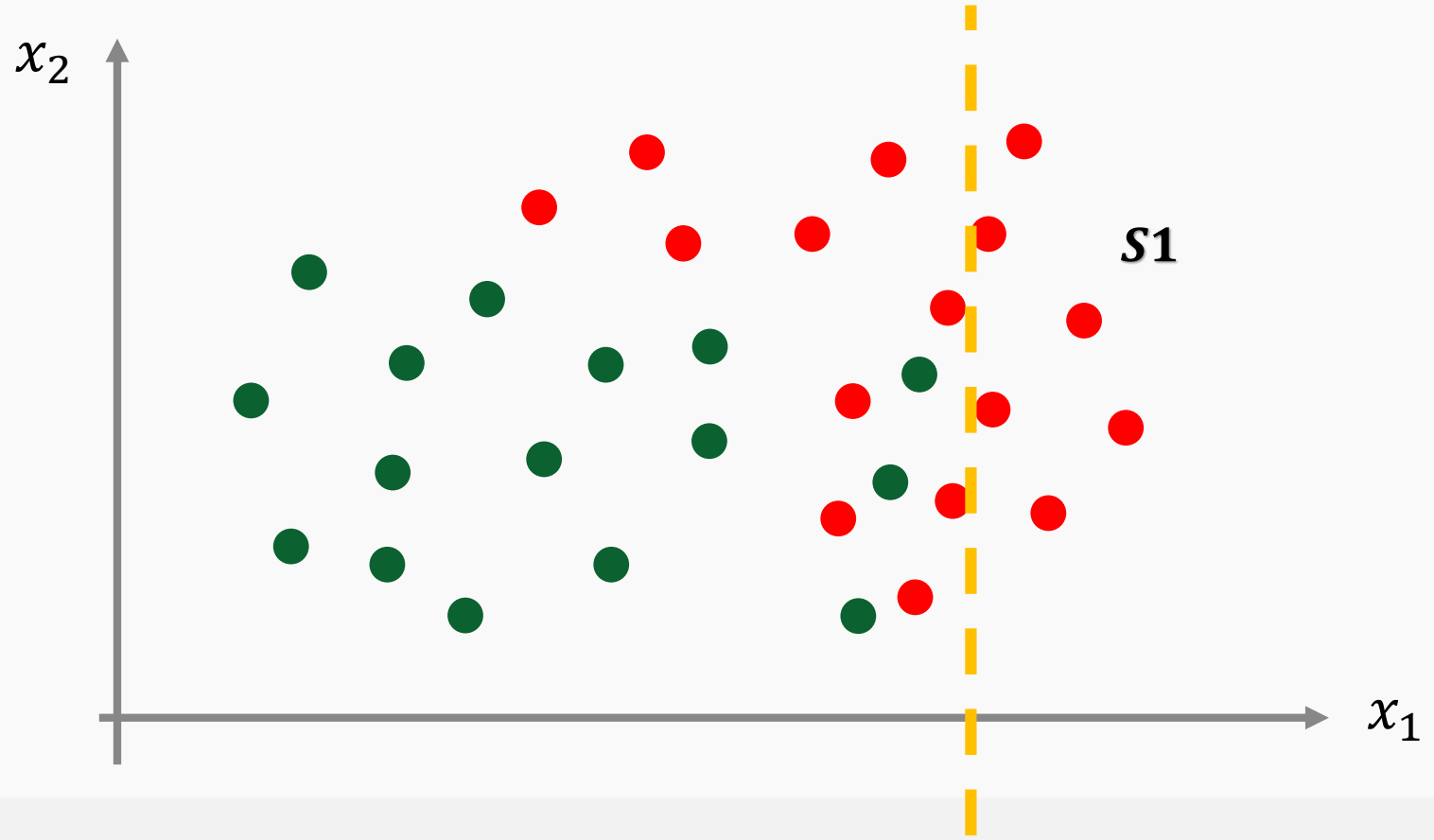

## План занятия

1. Решающие деревья (decision trees)
2. Ансамбли деревьев
3. Случайный лес (random forest)
4. Gradient Boosting

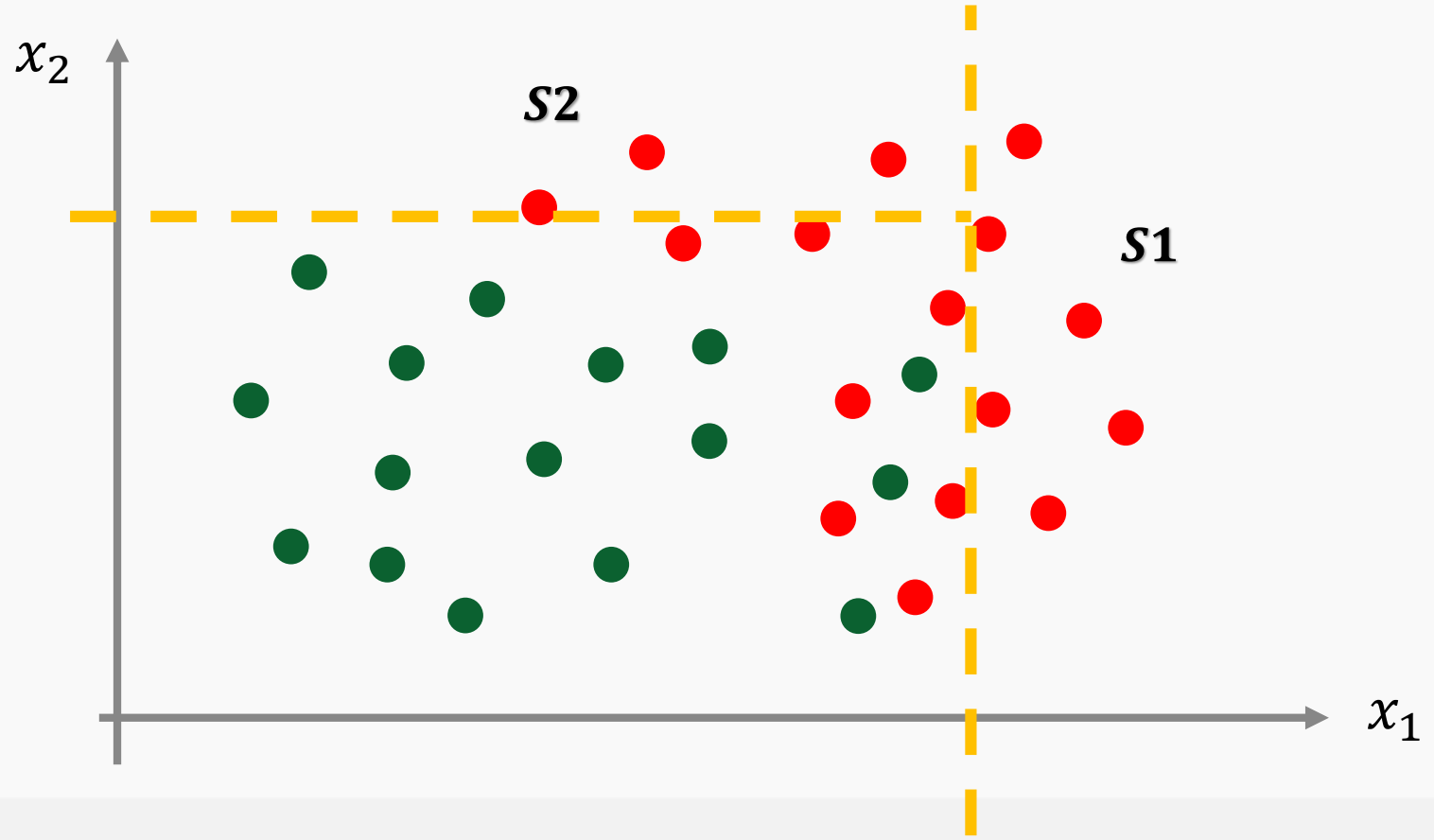
## Бинарное решающее дерево



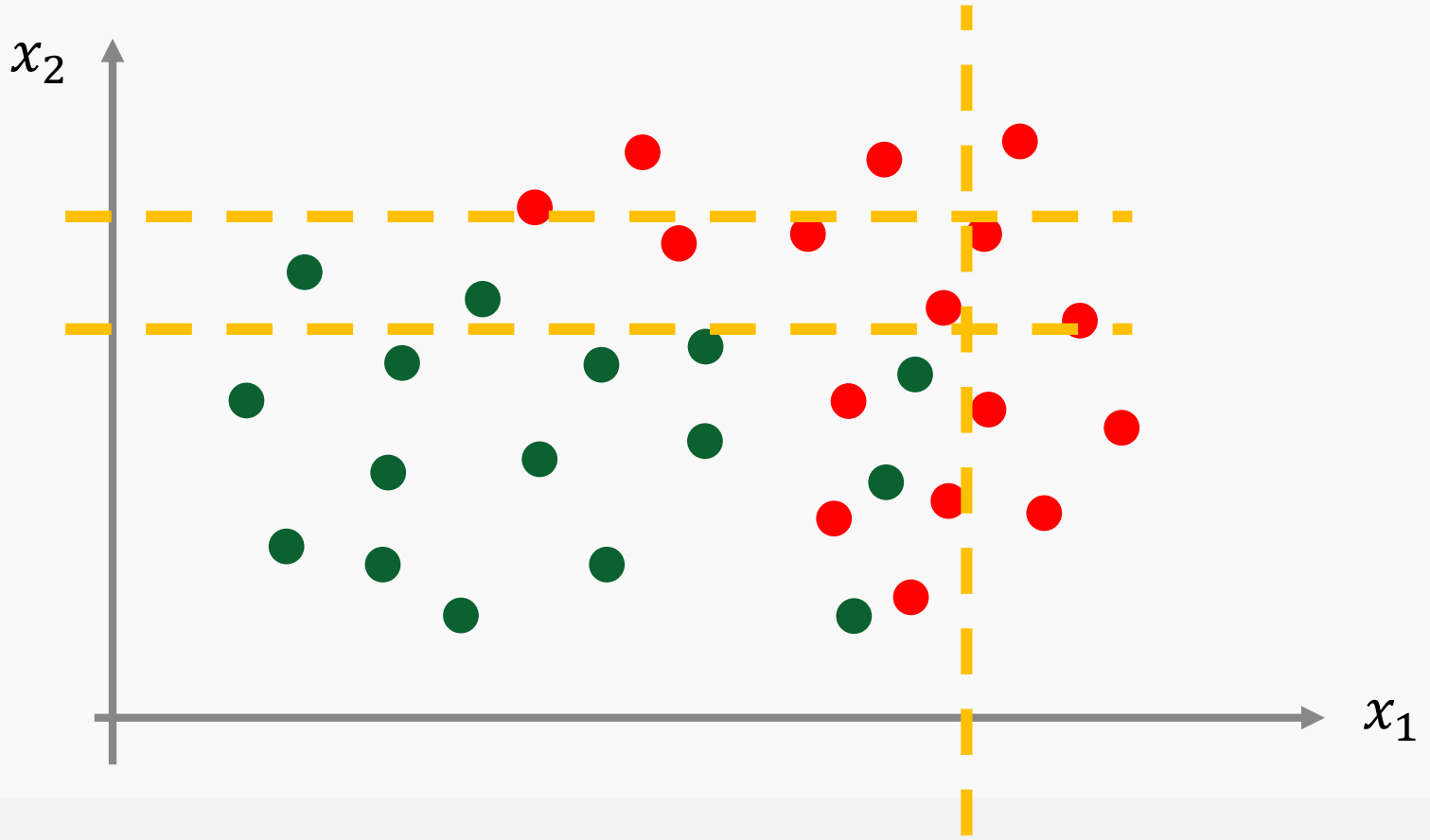
## Бинарное решающее дерево



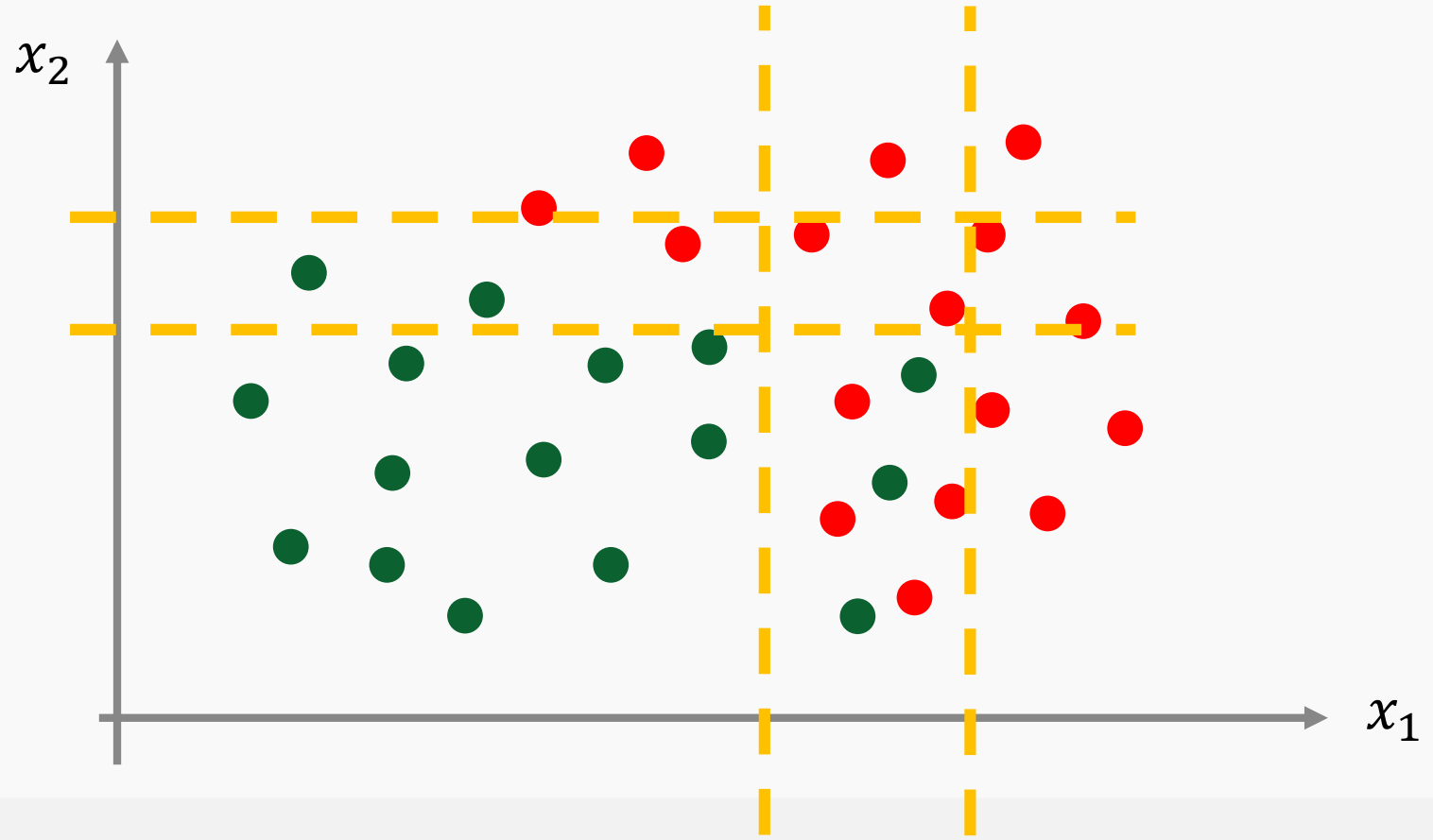
## Бинарное решающее дерево



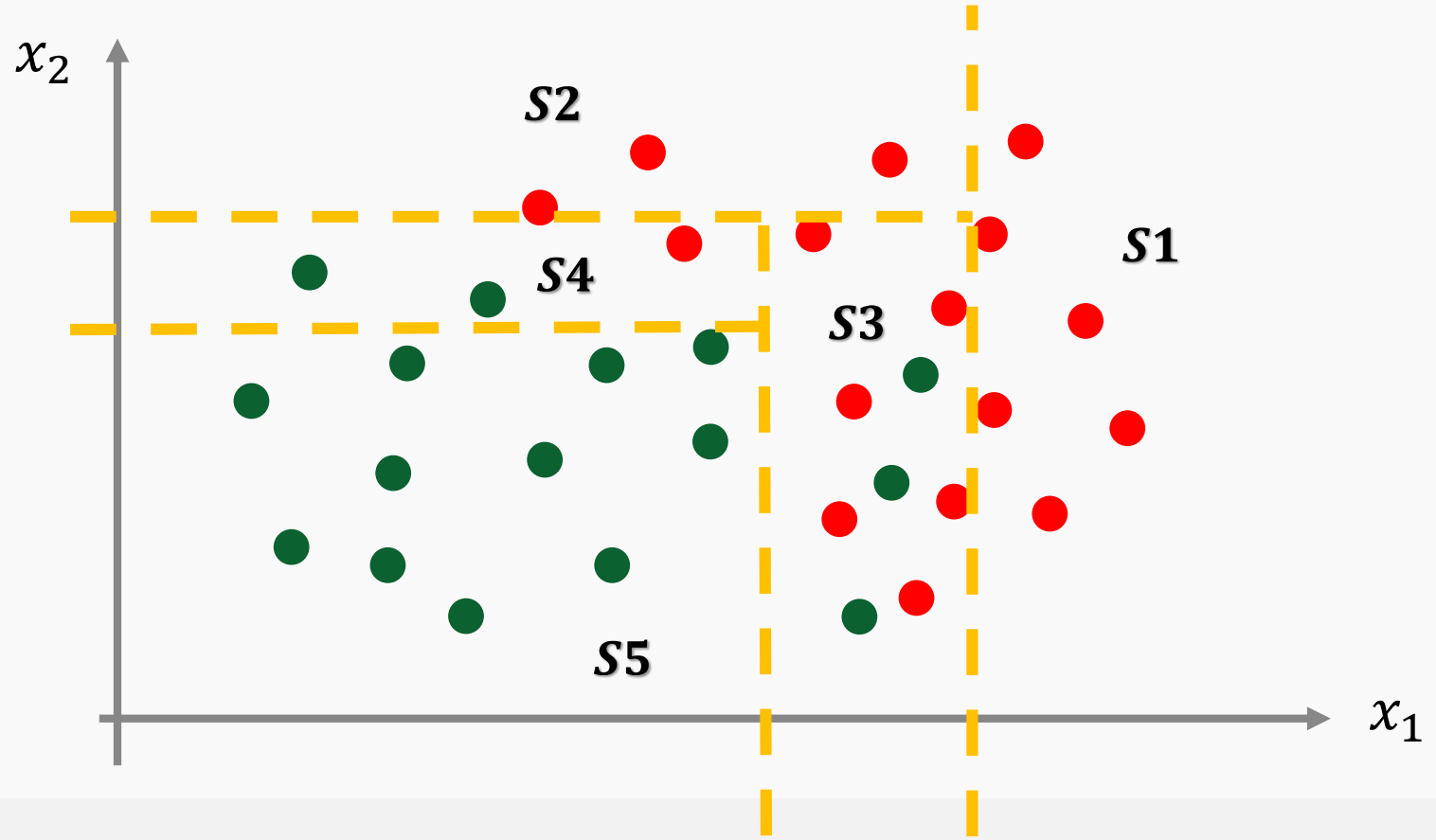
## Бинарное решающее дерево



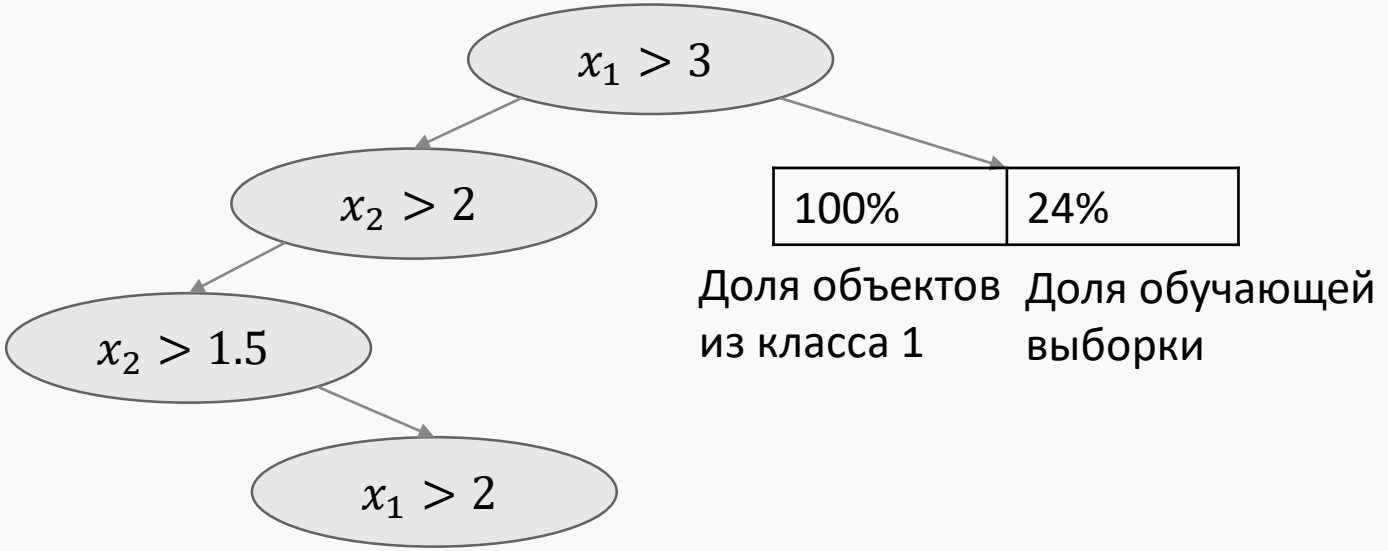
## Бинарное решающее дерево



## Бинарное решающее дерево



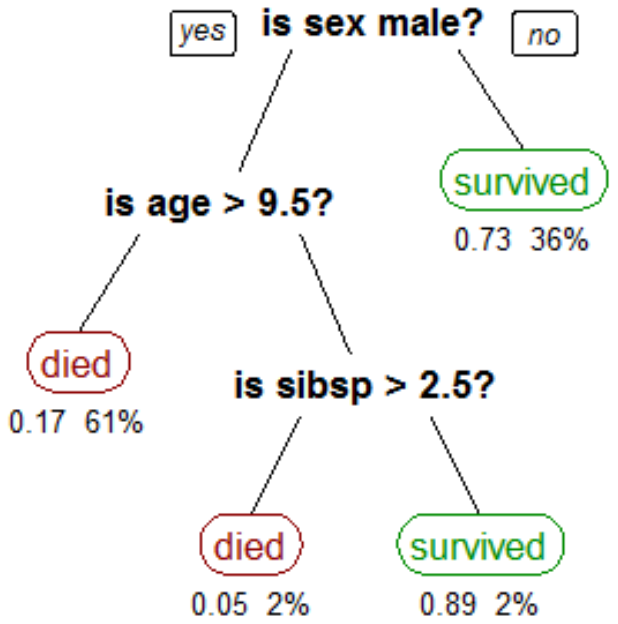
# Бинарное решающее дерево



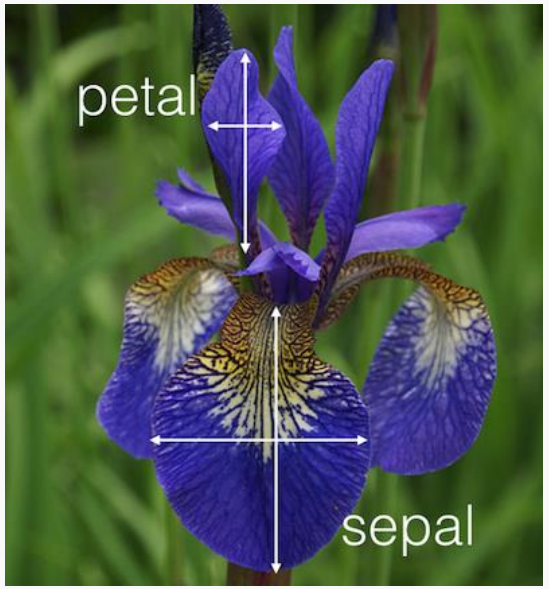
# Бинарное решающее дерево

Titanic dataset

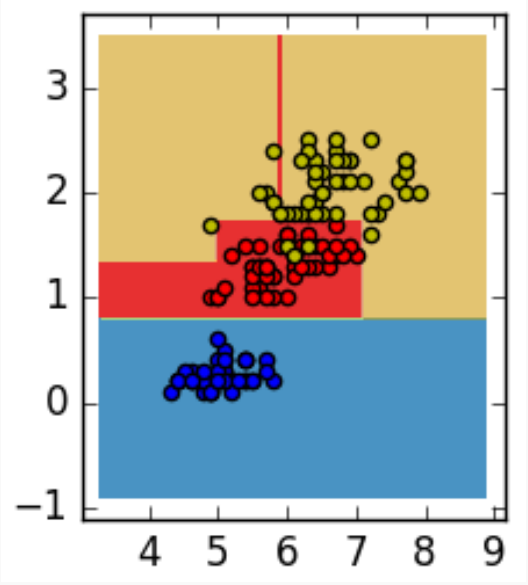
https://www.kaggle.com/c/titanic



# Бинарное решающее дерево, классификация

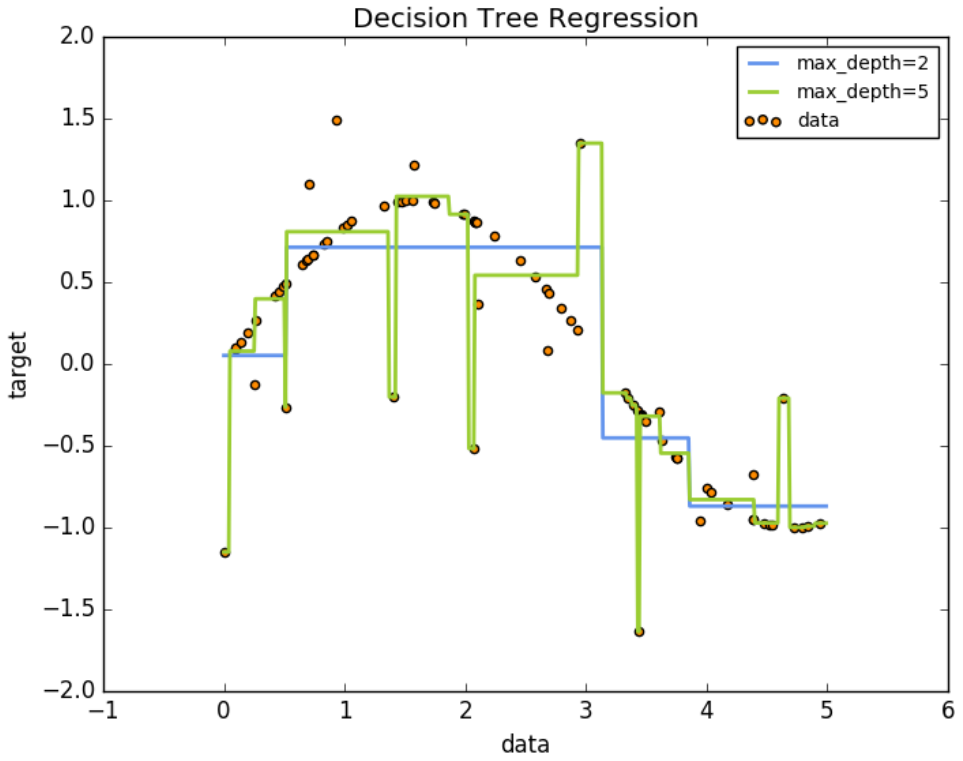



Iris dataset



# Введение в Машинное обучение

## Бинарное решающее дерево, регрессия




# Бинарное решающее дерево

## Критерий разбиения

Имеется множество: $X$

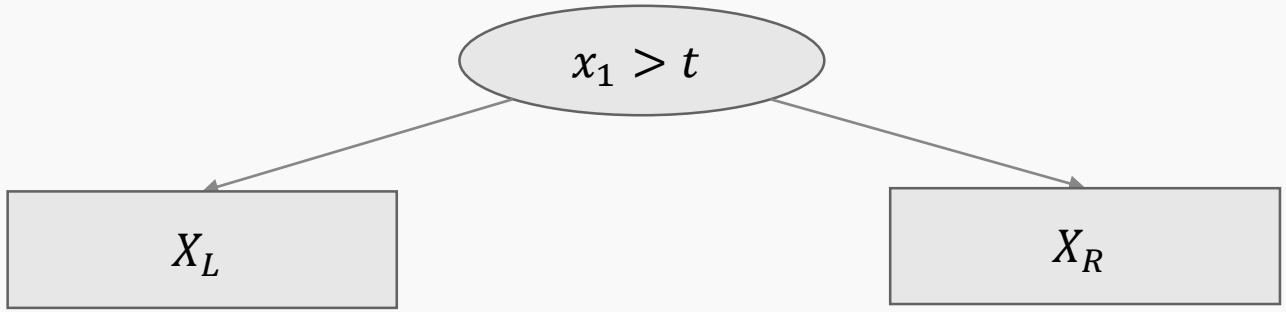
Мера неоднородности: $H(X)$

Решаем задачу бинарной классификации

1) Misclassification criteria: $H(X) = 1 - p_{max}$
2) Entropy criteria: $H(X) = -p_0 \ln(p_0) - p_1 \ln(p_1)$
3) Gini criteria: $H(X) = 2p_0 p_1$

где $p_0$ и $p_1$ - доли объектов из класса 0 и 1

# Бинарное решающее дерево




Уменьшения неопределённости (неоднородности) в узле:

$$G(X) = \frac{|X_L|}{|X|} H(X_L) + \frac{|X_R|}{|X|} H(X_R) \to min$$

## Бинарное решающее дерево

Критерий разбиения

Имеется множество: $X$

Мера неоднородности: $H(X)$

В случае, если имеется $N$ классов:

1) Misclassification criteria: $H(X) = 1 - p_{max}$

2) Entropy criteria: $H(X) = -\sum_{i=1}^{N} p_i \ln(p_i)$

3) Gini criteria: $H(X) = \sum_{i=1}^{N} p_i(1 - p_i)$

где $p_0$ и $p_1$ - доли объектов из класса 0 и 1


## Bootstrap aggregation (bagging)

Бутстреп - способ оценки стандартной ошибки статистик выборочного вероятностного распределения и способ семплирования выборок из набора данных основанный на методе Монте-Карло.

# Введение в Машинное обучение


## Bootstrap

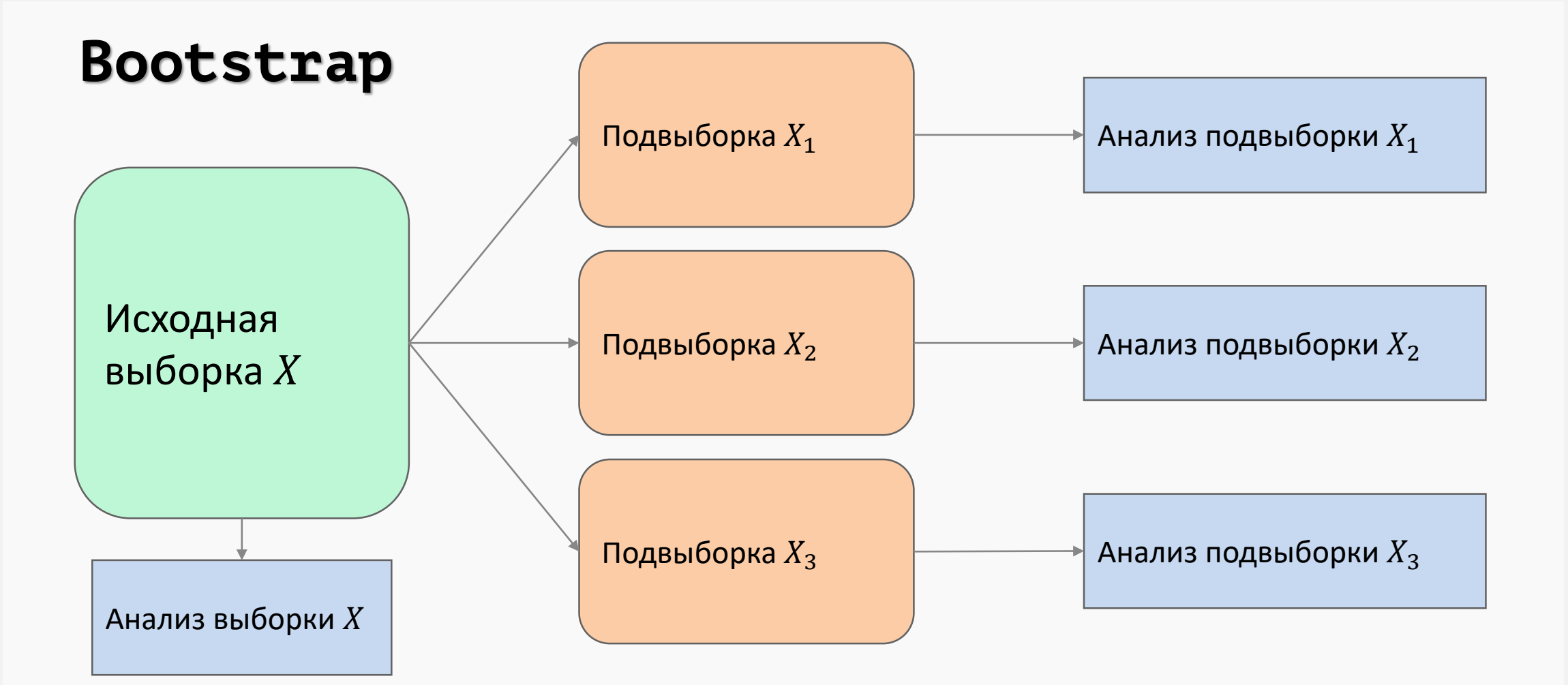
Имеется обучающая выборка $X$ размера $N$

Шаг 1. Равномерно возьмем из выборки $N$ объектов с возвращением. Будем $N$ раз выбирать произвольный объект выборки (считаем, что каждый объект «достается» с одинаковой вероятностью 1/N), причем каждый раз мы выбираем из всех исходных N объектов.

Шаг 2. Повторяем действия из Шага 1 M раз. В результате сгенерируем $M$ подвыборок $X_1, \ldots, X_M$ . (уникальные элементы ~ $(1 - 1/e) \sim 0.63$)

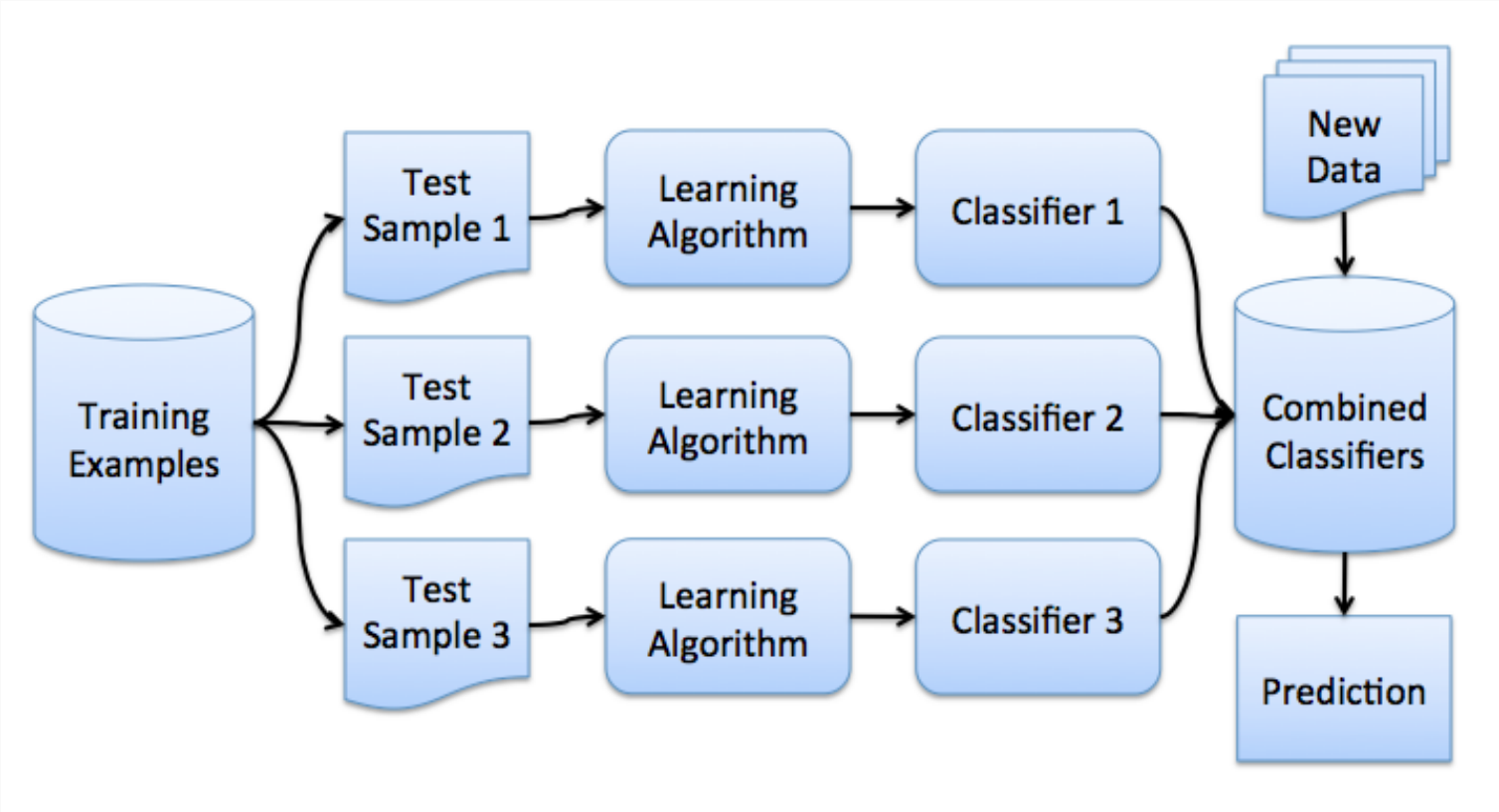
Шаг 3. Оценивать различные статистики исходного распределения.

# Введение в Машинное обучение



# Введение в Машинное обучение

## Bagging



# Bagging

Шаг 1) На основе bootstrap сгенерируем $M$ подвыборок $X_1, \ldots, X_M$ из исходного множества $X$

Шаг 2) На каждой из подвыборок обучим классификатор $h(X)$

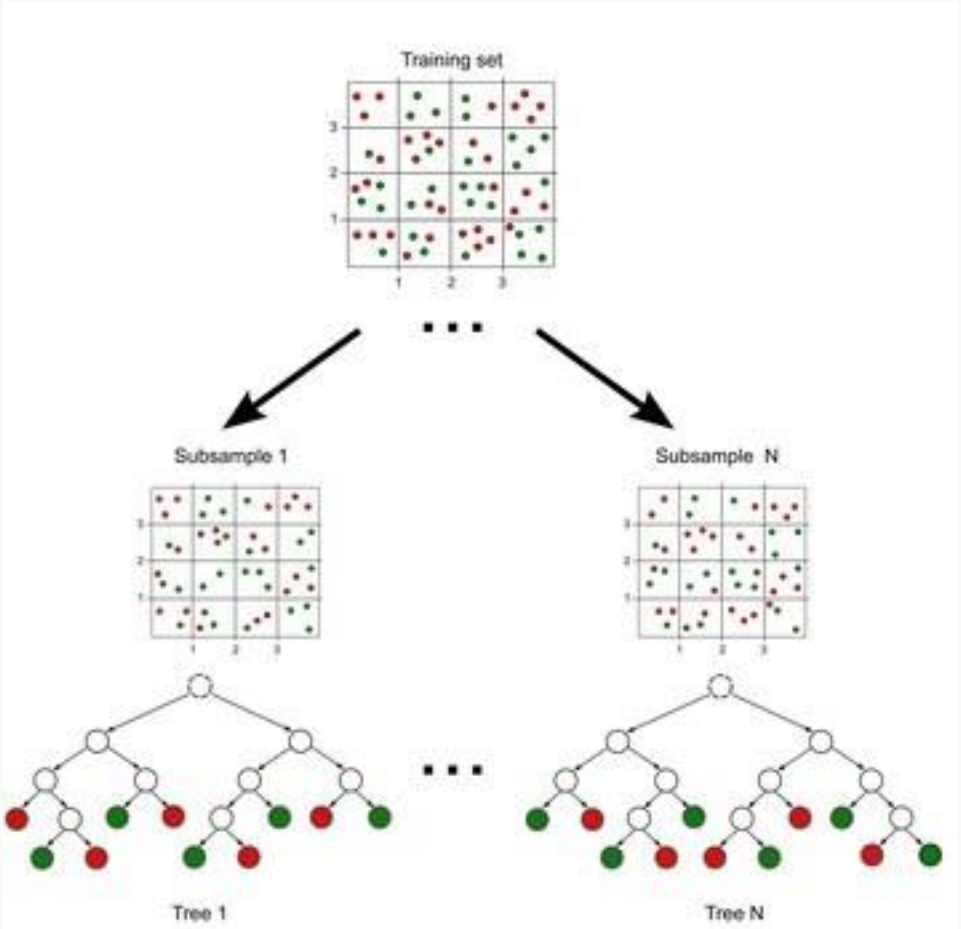
Шаг 3) Итоговый классификатор получается путём усреднения ответов: $h(x) = \frac{1}{M}\sum_{i=1}^{M} h_i(x)$

# Введение в Машинное обучение

## Random forest

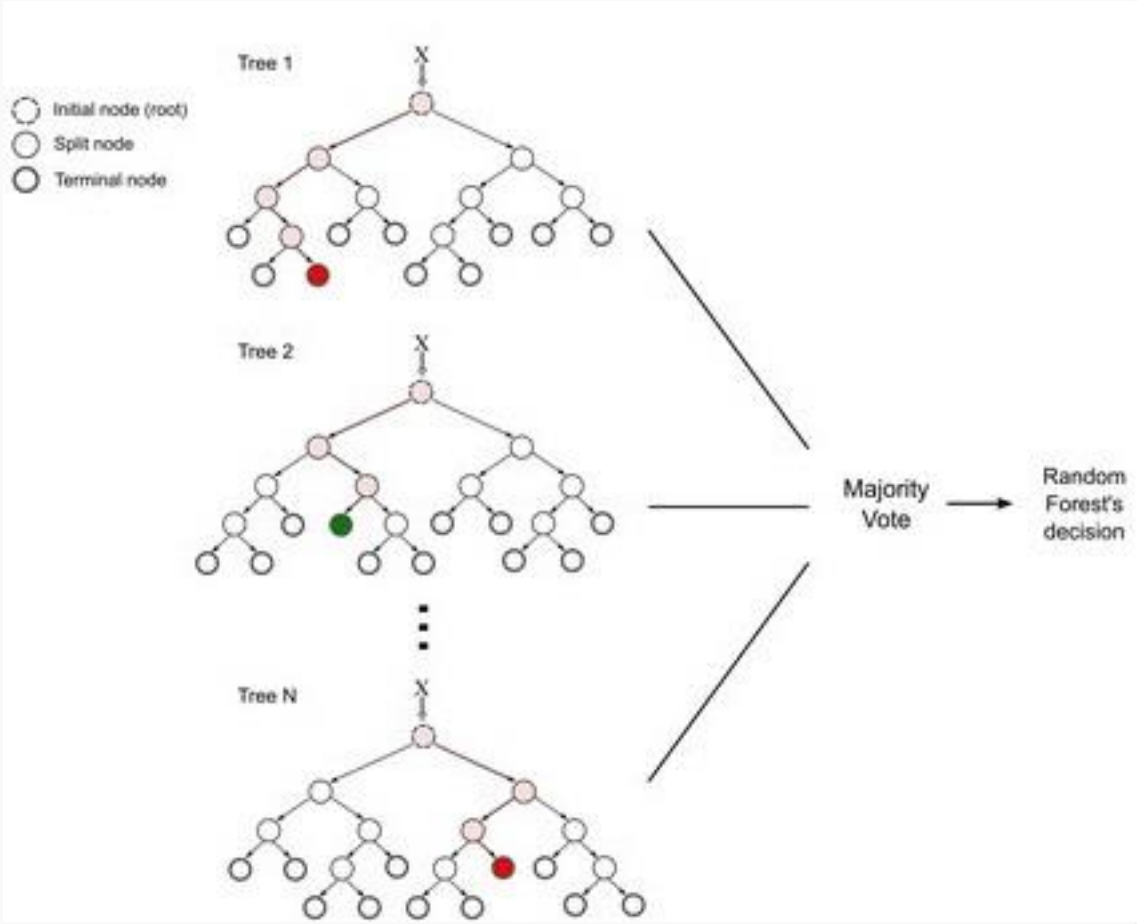
Шаг 1) На основе bootstrap сгенерируем $M$ подвыборок

Шаг 2) Строим на каждой подвыборке дерево, причём в узле будем выбирать $k$ случайных признаков



## Random forest

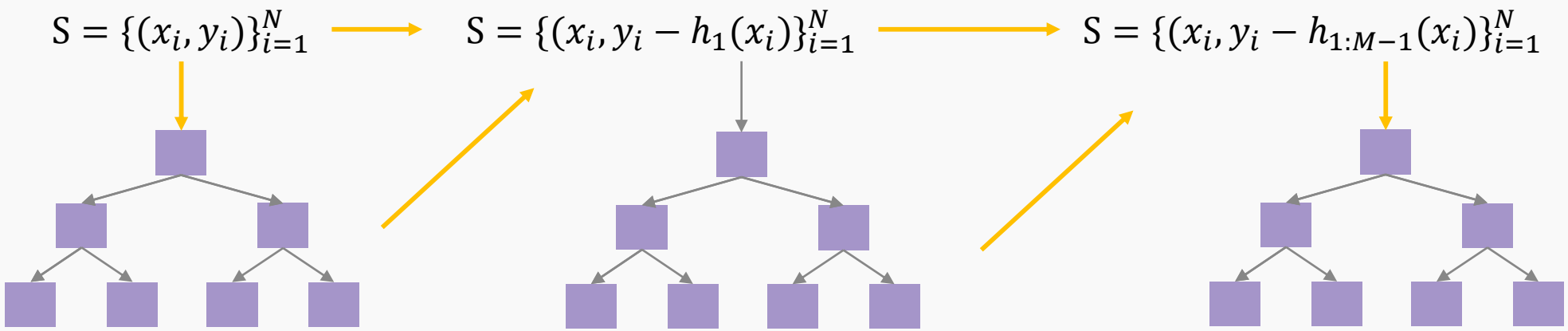
Шаг 3) Усредняем полученный результат от каждого дерева



# Gradient boosting decision trees

Имеется обучающая выборка $S = \{(x_i, y_i)\}_{i=1}^{N}$

Итоговый классификатор: $h(x) = h_1(x) + h_2(x) + \cdots + h_M(x)$



# Gradient boosting decision trees

## Обучение модели

Ошибка для $i$-ого объекта: $J\big(y_i, h(x_i)\big) = (y_i - h(x_i))^2$

Функция потерь:

$$L = \sum_{i=1}^{N} J\big(y_i, h(x_i)\big) = \sum_{i=1}^{N} (y_i - h(x_i))^2$$

# Gradient boosting decision trees

## Обучение модели

Функция потерь:

$$L = \sum_{i=1}^{N} J\big(y_i, h(x_i)\big) = \sum_{i=1}^{N} (y_i - h(x_i))^2$$

$$\frac{\partial L}{\partial h_i} = \sum_{i=1}^{N} \frac{\partial}{\partial h_i} J(y_i, h_i) = 2(y_i - h_i)$$

# Gradient boosting decision trees

## Обучение модели

$$\frac{\partial L}{\partial h_i} = \sum_{i=1}^{N} \frac{\partial}{\partial h_i} J(y_i, h_i) = 2(y_i - h_i)$$

$$h(x_i) = \sum_{j=1}^{M} a_j \, h_j(x_i)$$

1) Каждое новое дерево $j$ обучаем на ответах $y_i - h_{j-1}$
2) Коэффициенты $a_j$ подбираются на основе численной оптимизации функции ошибки $L$


**Контакты:**
a.spasenov@corp.mail.ru
alex_spasenov (Skype)

**Спасибо за внимание!**