

# Машинное обучение

Лекция 5

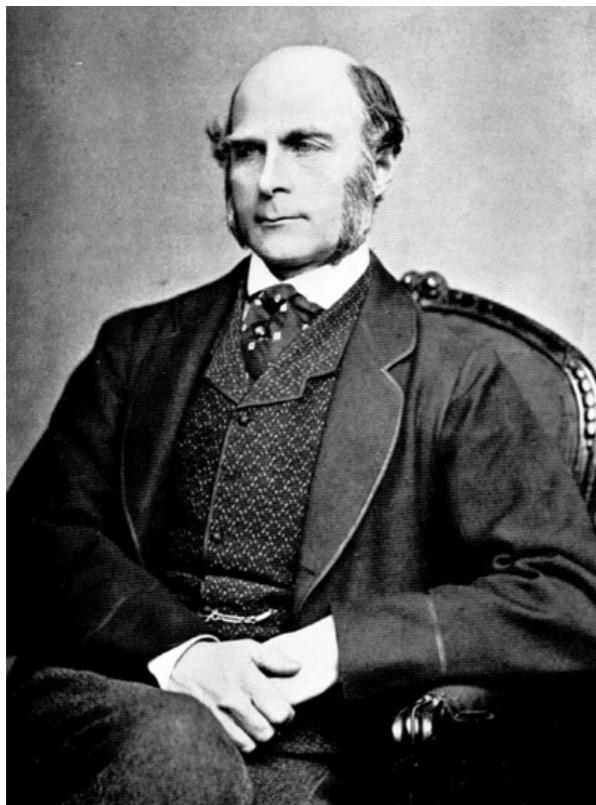
Композиции, ансамблирование, бустинг

*Власов Кирилл Вячеславович*



2019

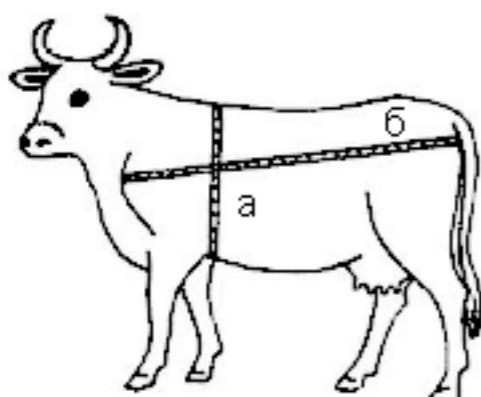
# Феномен: «Мудрость толпы»



В 1906 году британский ученый сэр **Фрэнсис Гальтон** посещал выставку достижений животноводства, где случайно провел крайне важное наблюдение.

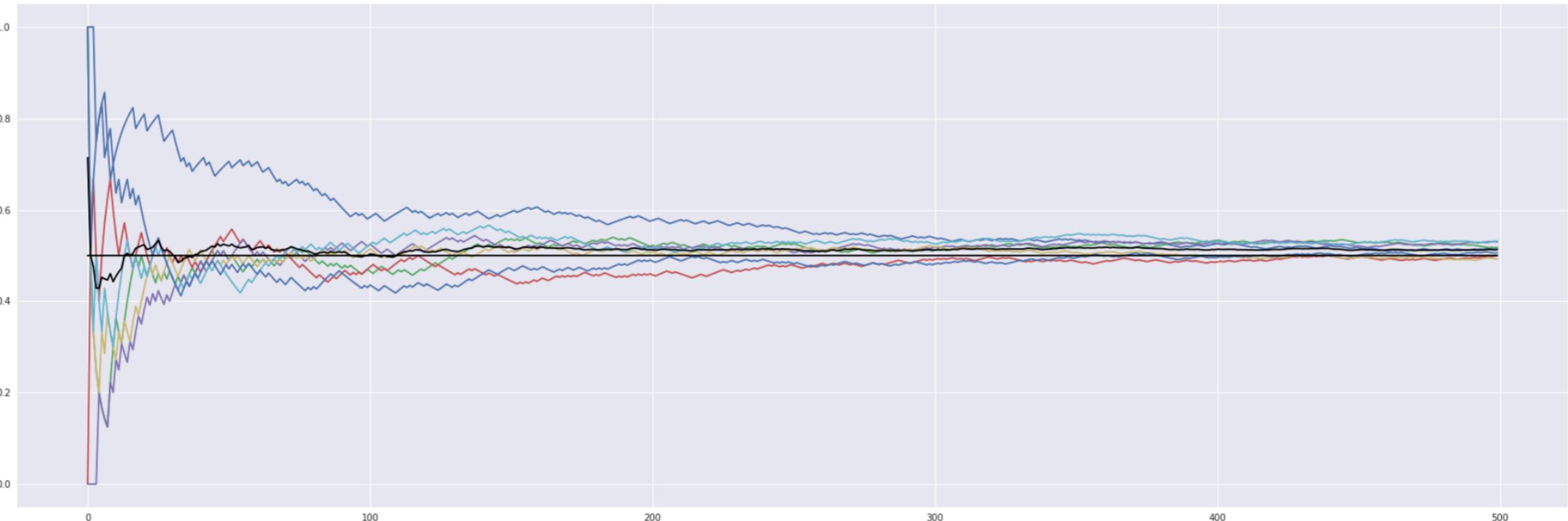
На выставке проводился конкурс, в рамках которого всем желающим предлагалось на глаз угадать точный вес забитого быка. Побеждал тот, кто называл самое близкое к истинному значение.

Полагая, что справиться с подобной задачей под силу только профессионалу, и чтобы доказать некомпетентность толпы, Гальтон посчитал среднее значение из почти восьми сотен догадок посетителей ярмарки. К удивлению ученого, толпа ошиблась меньше, чем на килограмм.



Определение живой массы коровы с помощью обмеров:  
а - обхват туловища  
б - косая длина

# Пример: «Бросание монетки»

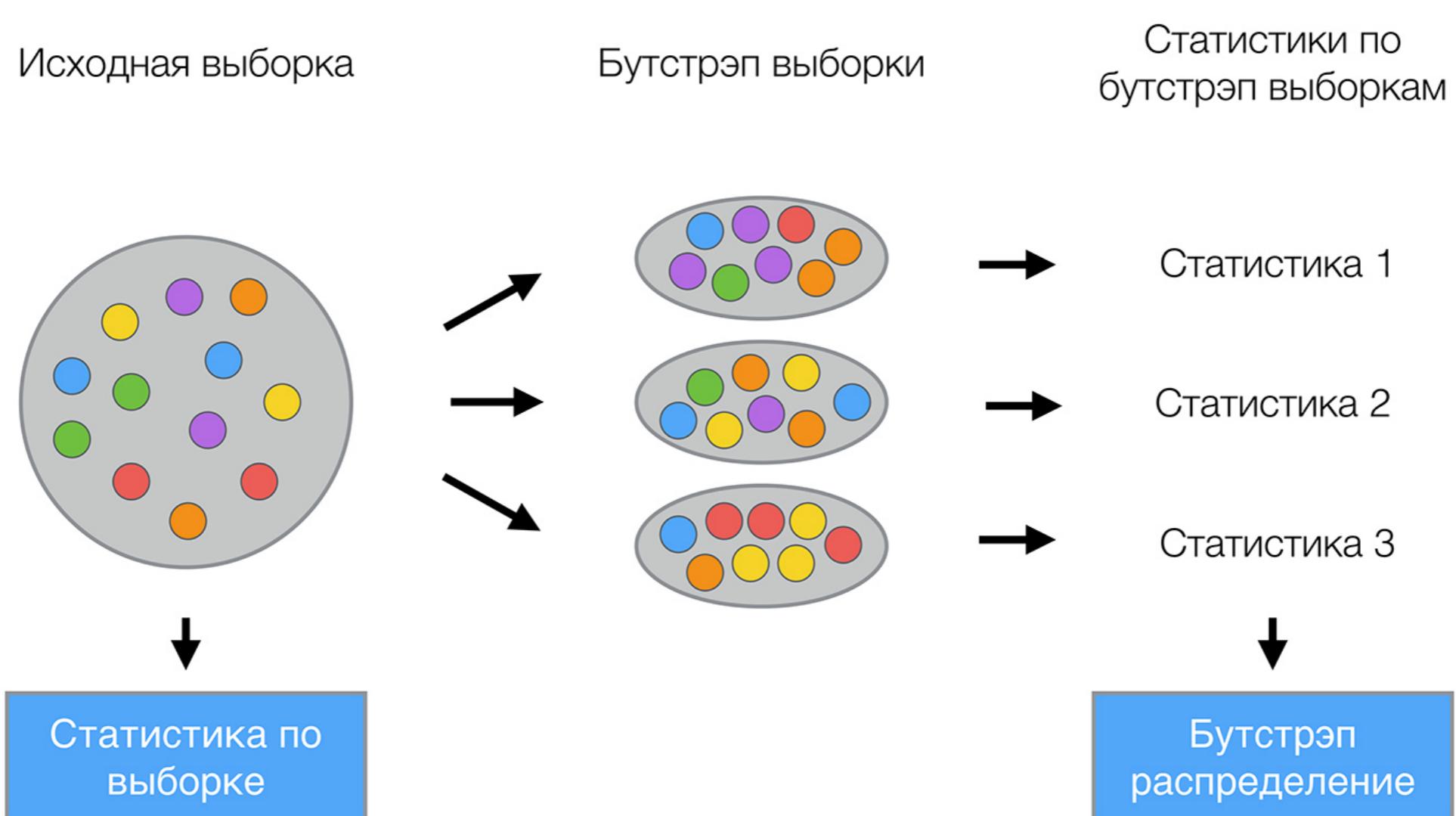


Если будем кидать монетку 500 раз, то  
соотношение «Орлов» и «Решек» с каждым  
броском будем стремиться к 1/2

Если повторить монетку несколько раз, то  
результаты будут отличаться (но все равно  
стремиться к 1/2)



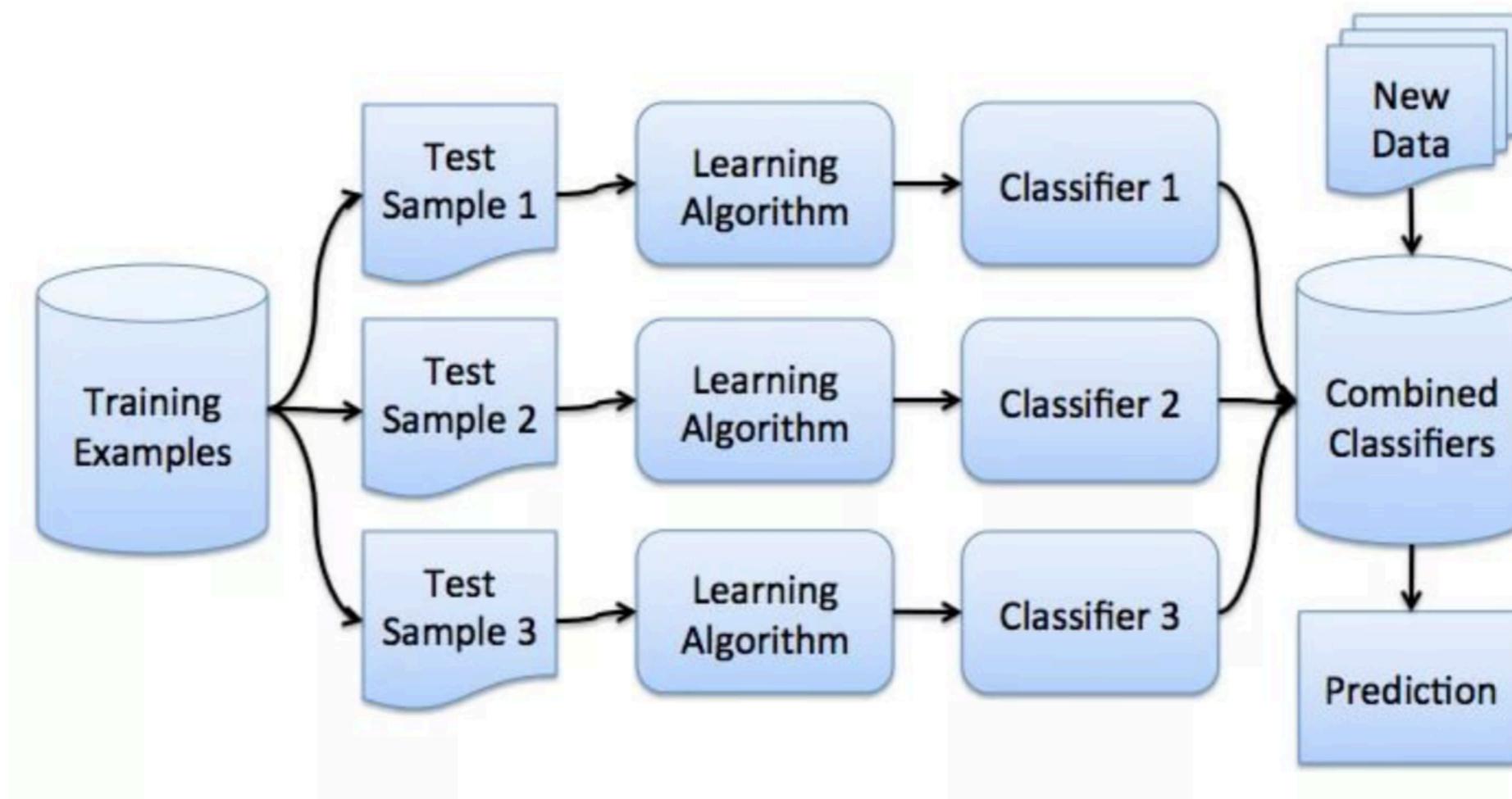
# Bootstrap



# Bootstrap

**Рассмотрим пример!**

# Bagging (Bootstrap aggregating)



# **Bagging (Bootstrap aggregating)**

**Рассмотрим пример!**

# Метод случайных подпространств



## Теорема Кондорсе о присяжных

Если каждый член жюри присяжных имеет **независимое мнение**, и если вероятность правильного решения члена жюри больше 0.5, то тогда вероятность правильного решения присяжных в целом возрастает с увеличением количества членов жюри, и стремиться к единице. Если же вероятность быть правым у каждого из членов жюри меньше 0.5, то вероятность принятия правильного решения присяжными в целом монотонно уменьшается и стремится к нулю с увеличением количества присяжных.

# Метод случайных подпространств



## Теорема Кондорсе о присяжных

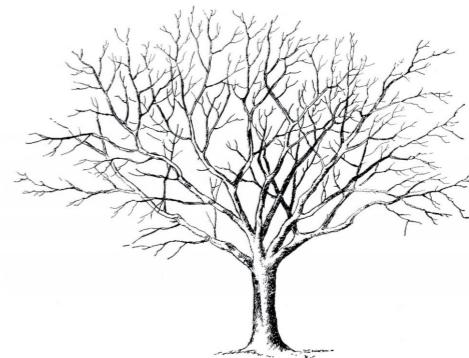
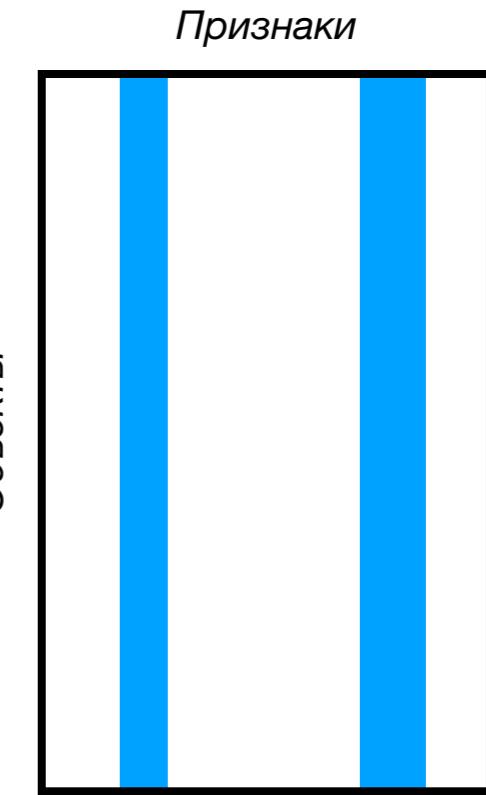
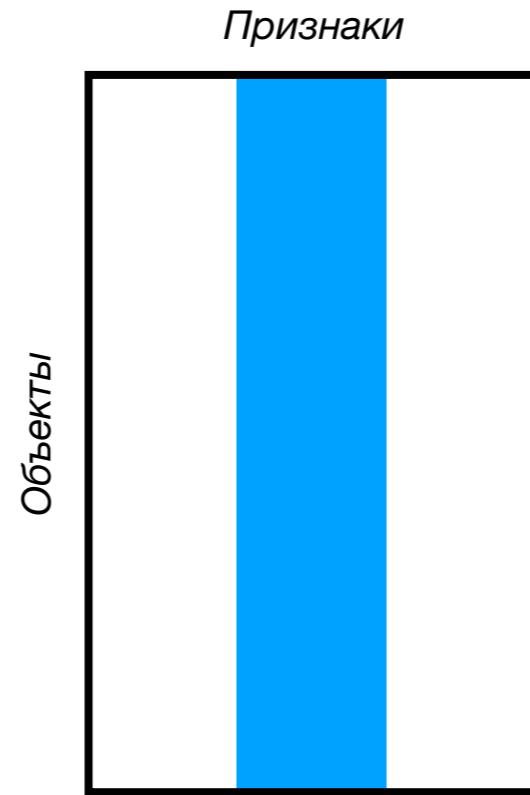
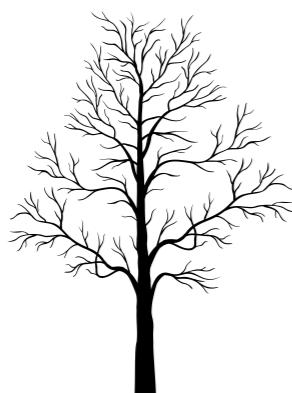
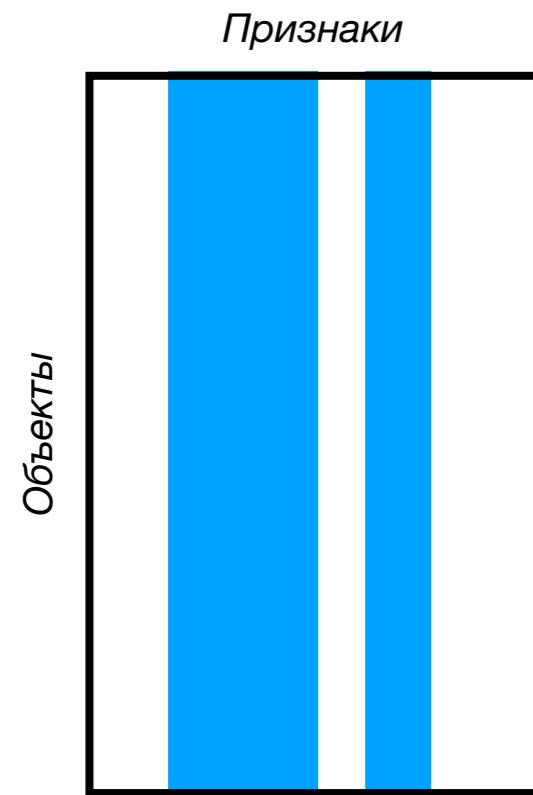
Если каждый член жюри присяжных имеет **независимое мнение**, и если вероятность правильного решения члена жюри больше 0.5, то тогда вероятность правильного решения присяжных в целом возрастает с увеличением количества членов жюри, и стремиться к единице. Если же вероятность быть правым у каждого из членов жюри меньше 0.5, то вероятность принятия правильного решения присяжными в целом монотонно уменьшается и стремится к нулю с увеличением количества присяжных.

$$\mu = \sum_{i=m}^N C_N^i p^i (1-p)^{N-i}$$

$$p > 0.5 \quad \mu > p$$

$$N \rightarrow \infty \quad \mu \rightarrow 1$$

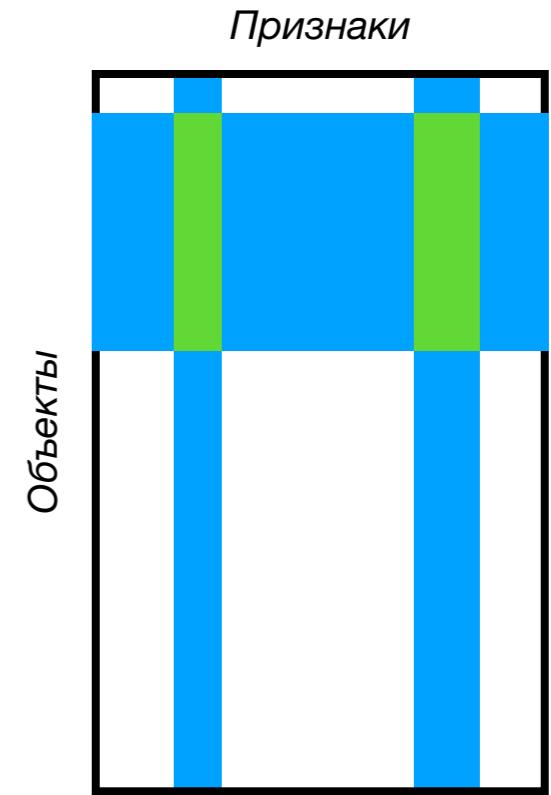
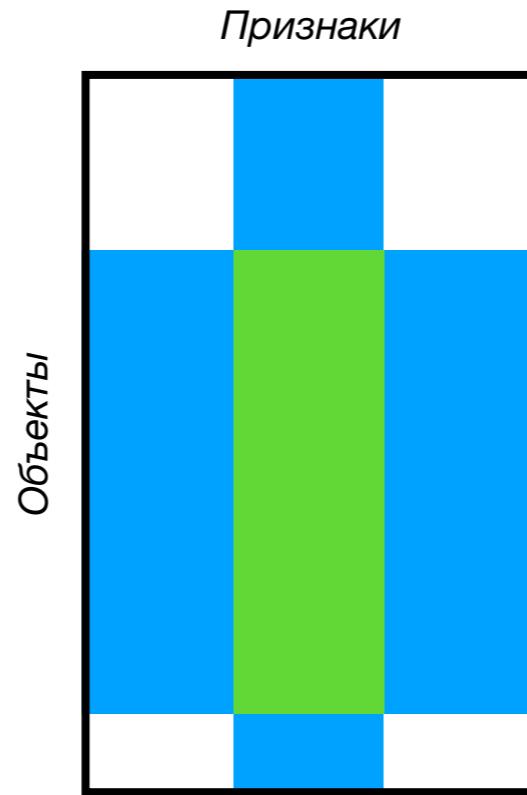
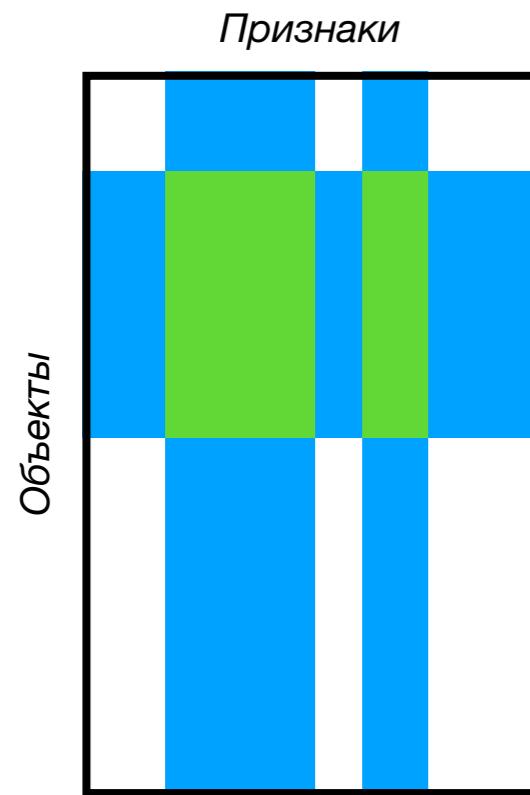
# Метод случайных подпространств



Для классификации:  $m = \sqrt{n}$

Для регрессии:  $m = \frac{n}{3}$

# Случайный лес



## Алгоритм:

1. Генерируем bootstrap выборку
2. Строим дерево, такое что
  - **В каждом узле** выбираем  $m$  случайных признаков из  $n$  возможных и по ним делаем разбиение
  - Дерево строим, до тех пор пока не достигнем определенной глубины дерева или пока в каждом листе больше объектов, чем пороговое
3. Повторяем п. 1 и 2 пока не достигнем заданного количества деревьев
4. Усредняем ответы деревьев

## Для классификации:

$$m = \sqrt{n}$$

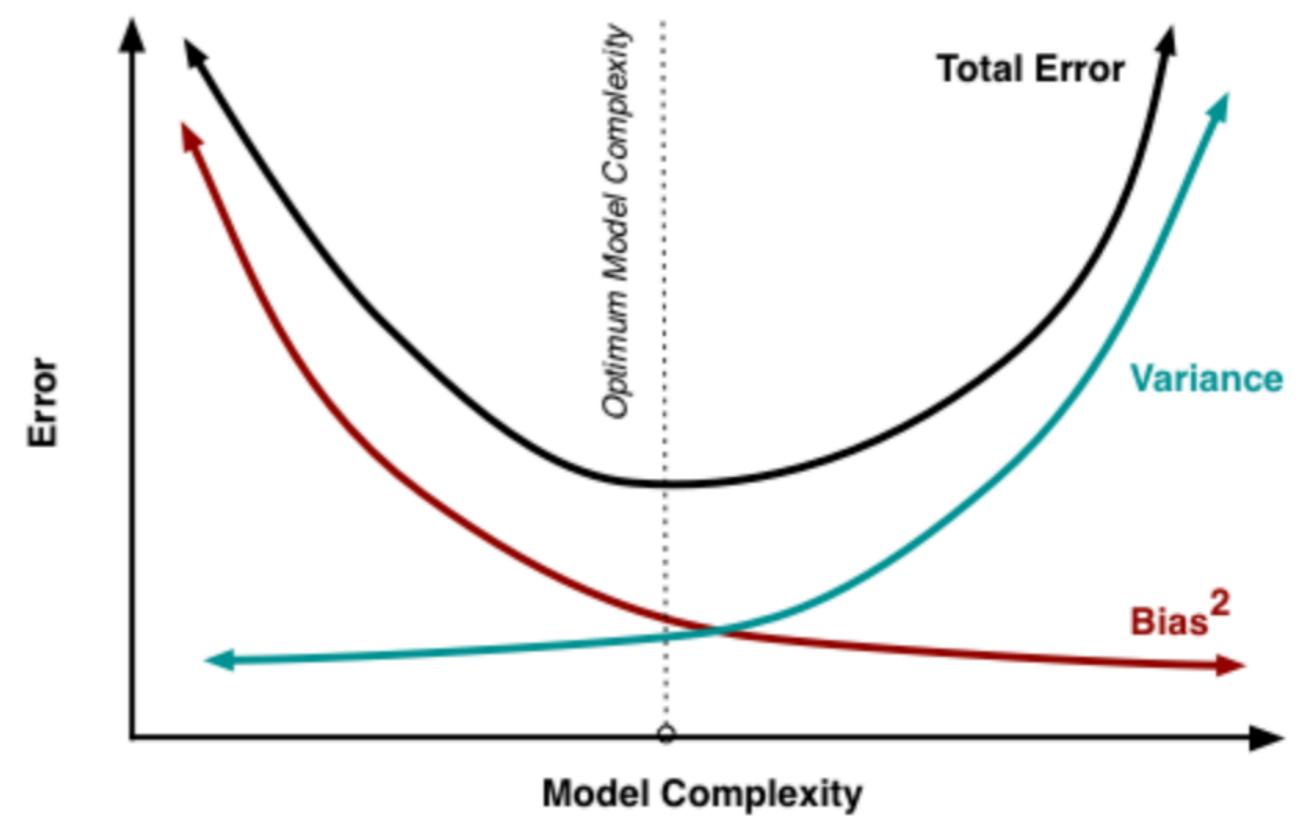
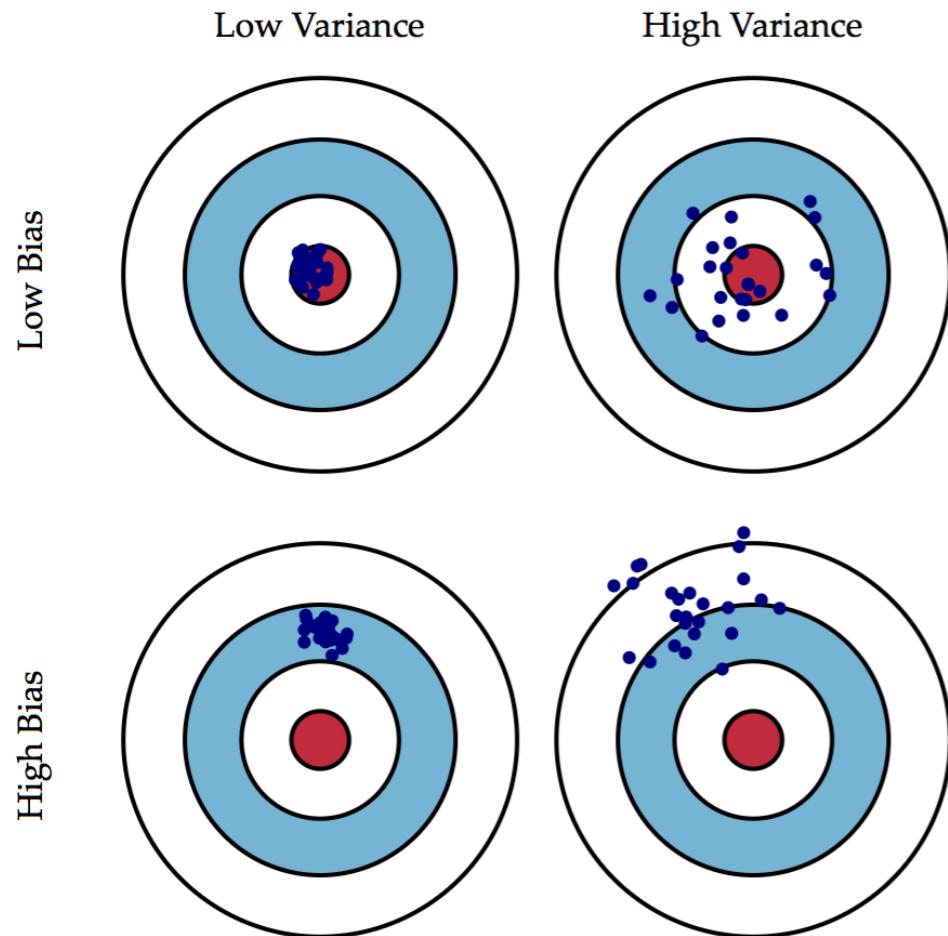
Пока один объект в листе

## Для регрессии:

$$m = \frac{n}{3}$$

Пока пять объектов в листе

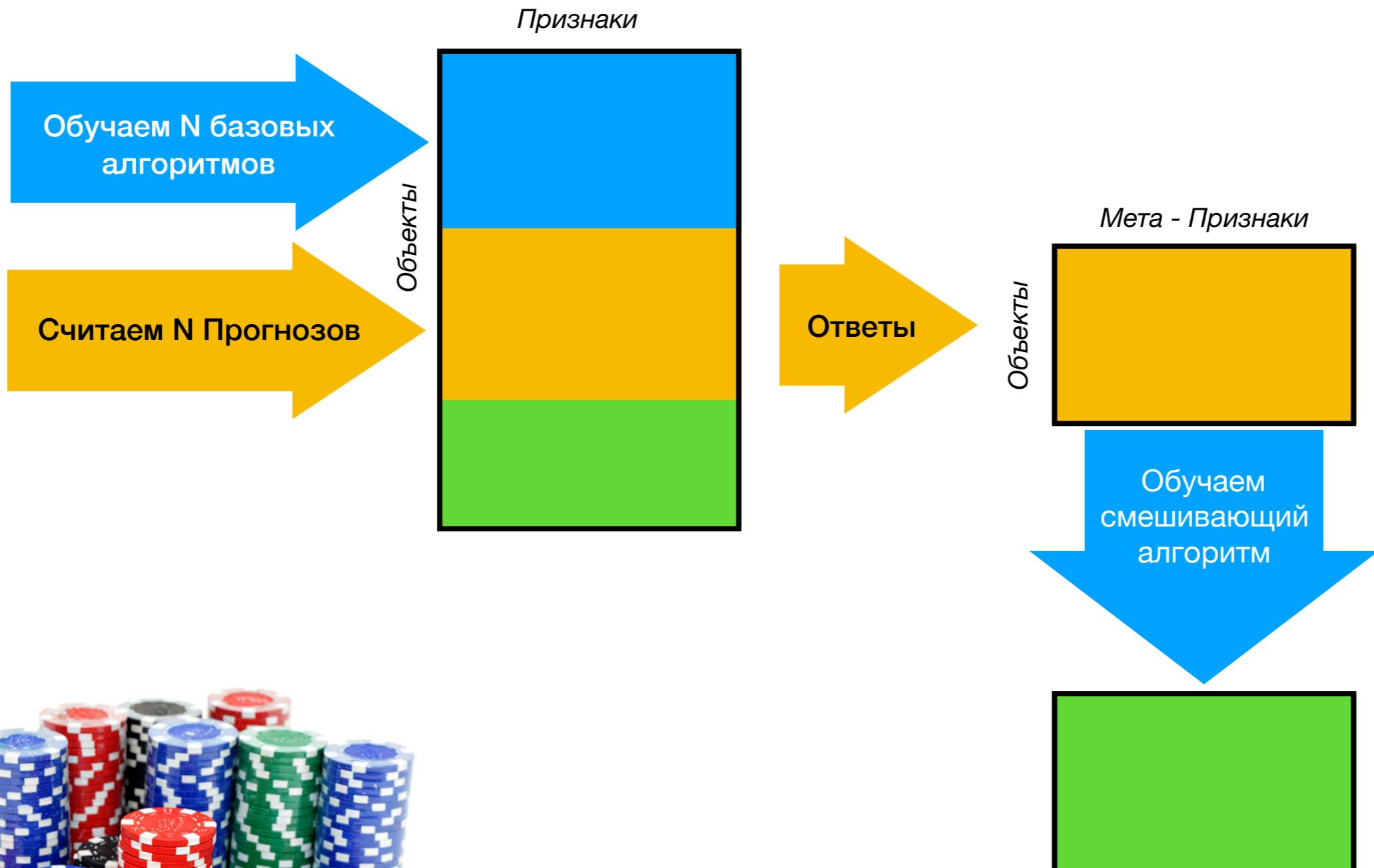
# Bias and Variance tradeoff



$$Err(x) = E[(Y - \hat{f}(x))^2]$$

$$Err(x) = Bias^2 + Variance + IrreducibleError$$

# Stacking



# Blending

**Blending** – частный случай Stacking'a

В качестве решающего алгоритма используется функция:

$$\sum_{i=1}^N a_i f_i(x), \quad \sum_{i=1}^N a_i = 1$$

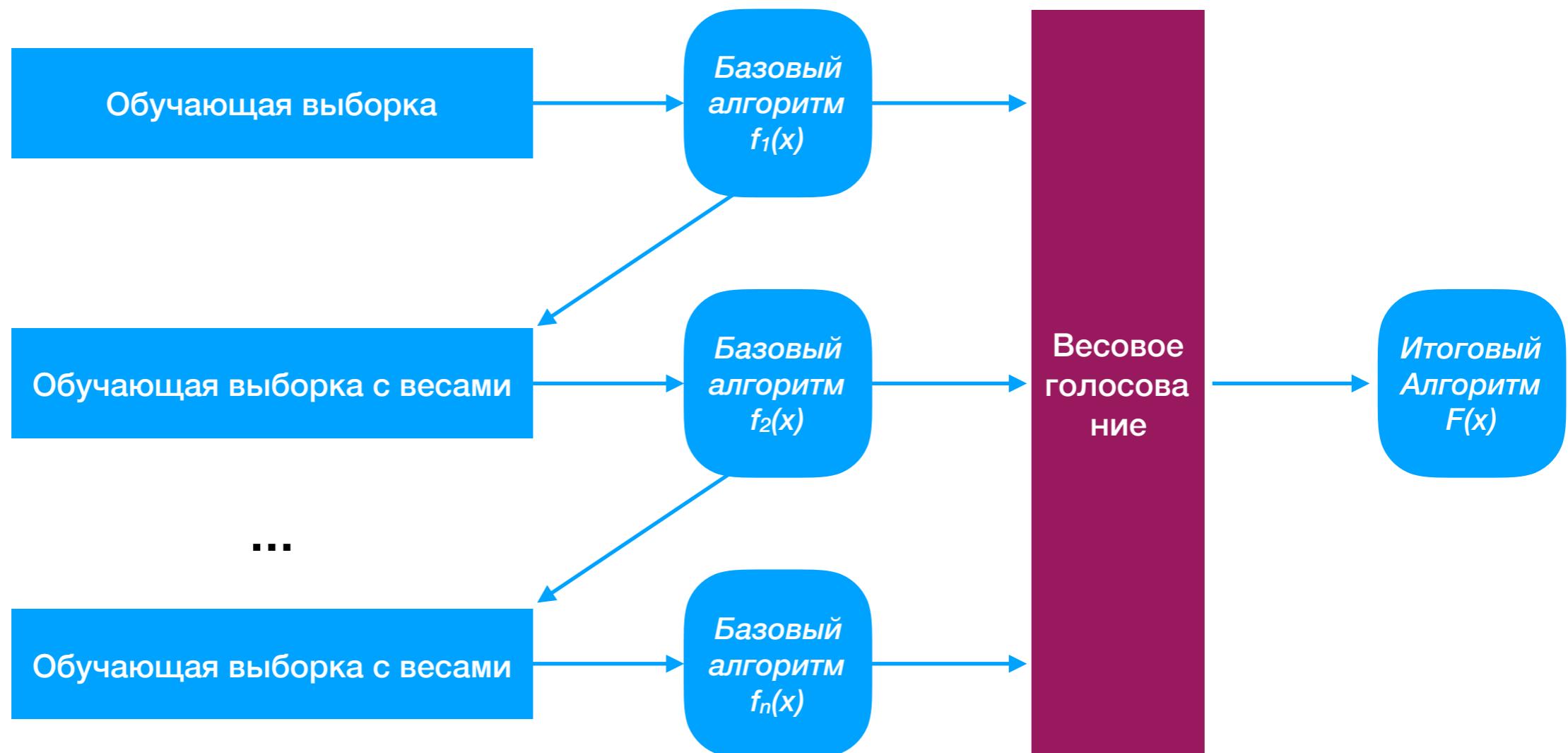
$f_i(x)$  Базовый алгоритм

$N$  Количество базовых алгоритмов



# Бустинг

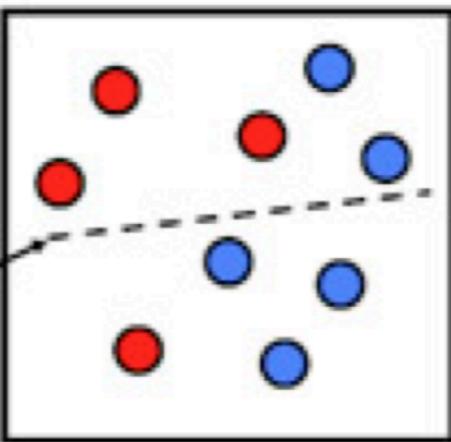
**Boosting** – последовательное добавление в ансамбль алгоритмов, каждый из которых корректирует ошибки предшественника.



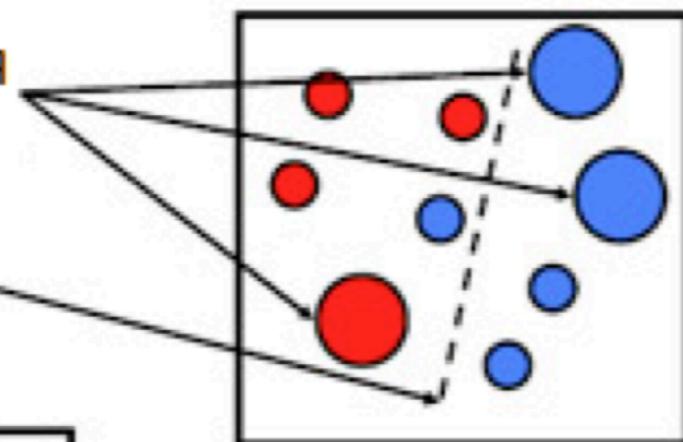
*AdaBoost (сокр. от adaptive boosting)*  
алгоритм, предложенный Йоавом Фройндом и Робертом Шапире Робертом  
Шапире

# AdaBoost

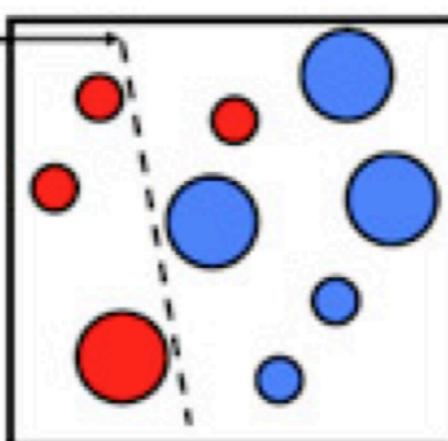
Равномерное  
распределение  
весов  
слабый  
классификатор  
1



Модификация  
весов  
слабый  
классификатор  
2



слабый  
классификатор  
3



Модификация  
весов

$$H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

2. Получаем ответы и отклонения

$$y - \hat{y}$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

2. Получаем ответы и отклонения

$$y - \hat{y}$$

3. Получаем новую обучающую выборку

$$(x_1, y_1 - f(x)) \dots (x_n, y_n - f(x))$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

2. Получаем ответы и отклонения

$$y - \hat{y}$$

3. Получаем новую обучающую выборку

$$(x_1, y_1 - f(x)) \dots (x_n, y_n - f(x))$$

4. Обучаем классификатор

$$a_i(x)$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

2. Получаем ответы и отклонения

$$y - \hat{y}$$

3. Получаем новую обучающую выборку

$$(x_1, y_1 - f(x)) \dots (x_n, y_n - f(x))$$

4. Обучаем классификатор

$$a_i(x)$$

5. Объединяем

$$f(x) + a_i(x) = y$$

# Градиентный бустинг

1. Строим алгоритм

$$\hat{y} = f(x)$$

2. Получаем ответы и отклонения

$$y - \hat{y}$$

3. Получаем новую обучающую выборку

$$(x_1, y_1 - f(x)) \dots (x_n, y_n - f(x))$$

4. Обучаем классификатор

$$a_i(x)$$

5. Объединяем

$$f(x) + a_i(x) = y$$

6. Возвращаемся к п. 2

# Градиентный бустинг

$$\mathcal{Q} = \frac{1}{l} \sum_{i=1}^l \mathcal{L}(y, f(x))$$

*Рассмотрим  $f(x)$  как параметр,  
если Функция MSE*

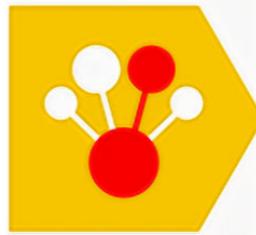
$$\frac{\partial \mathcal{Q}}{\partial f(x_i)} = f(x_i) - y$$

$y - \hat{y}$     *Отрицательный градиент!*

# Градиентный бустинг



**XGBoost**



**H<sub>2</sub>O**

# Ссылки

Открытый курс машинного обучения: [Тема 5](#) и [Тема 10](#)

[Репозитории Евгения Соколова](#)

Статья про [Стекинг \(Stacking\)](#) и [блэндинг \(Blending\)](#) в блоге А.Г. Дьяконова