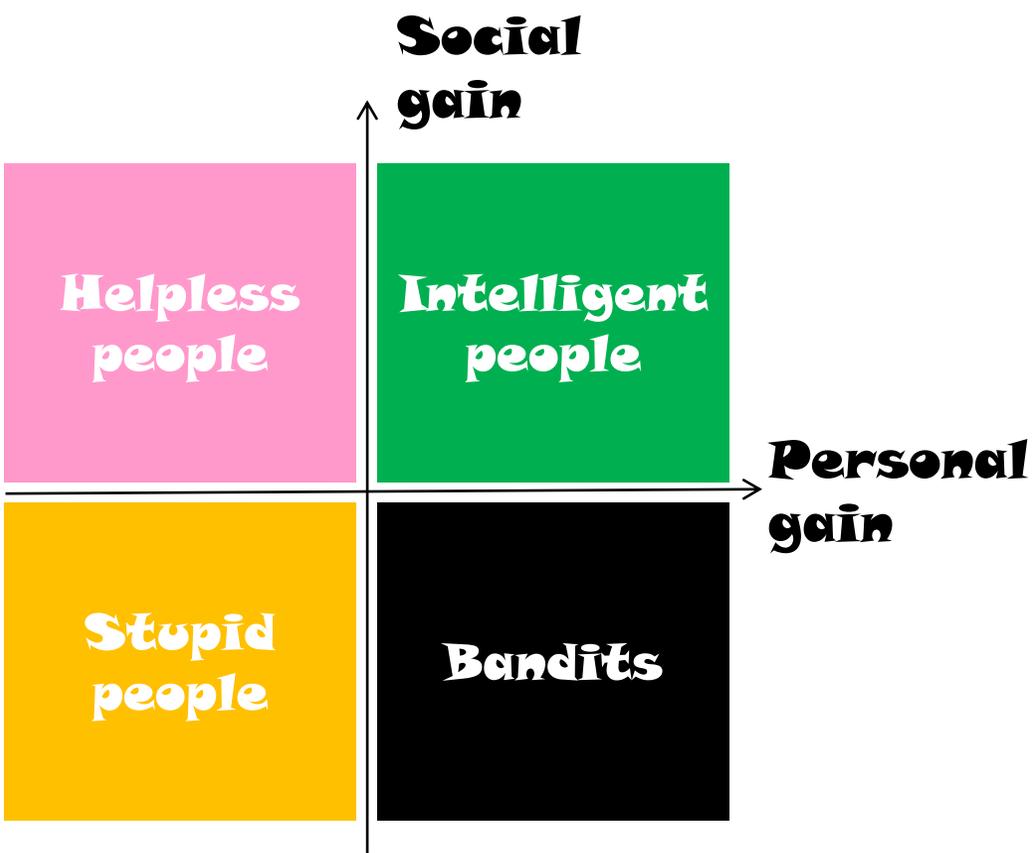


# Задача классификации данных



*Классификация – нить Ариадны в лабиринте природы.*

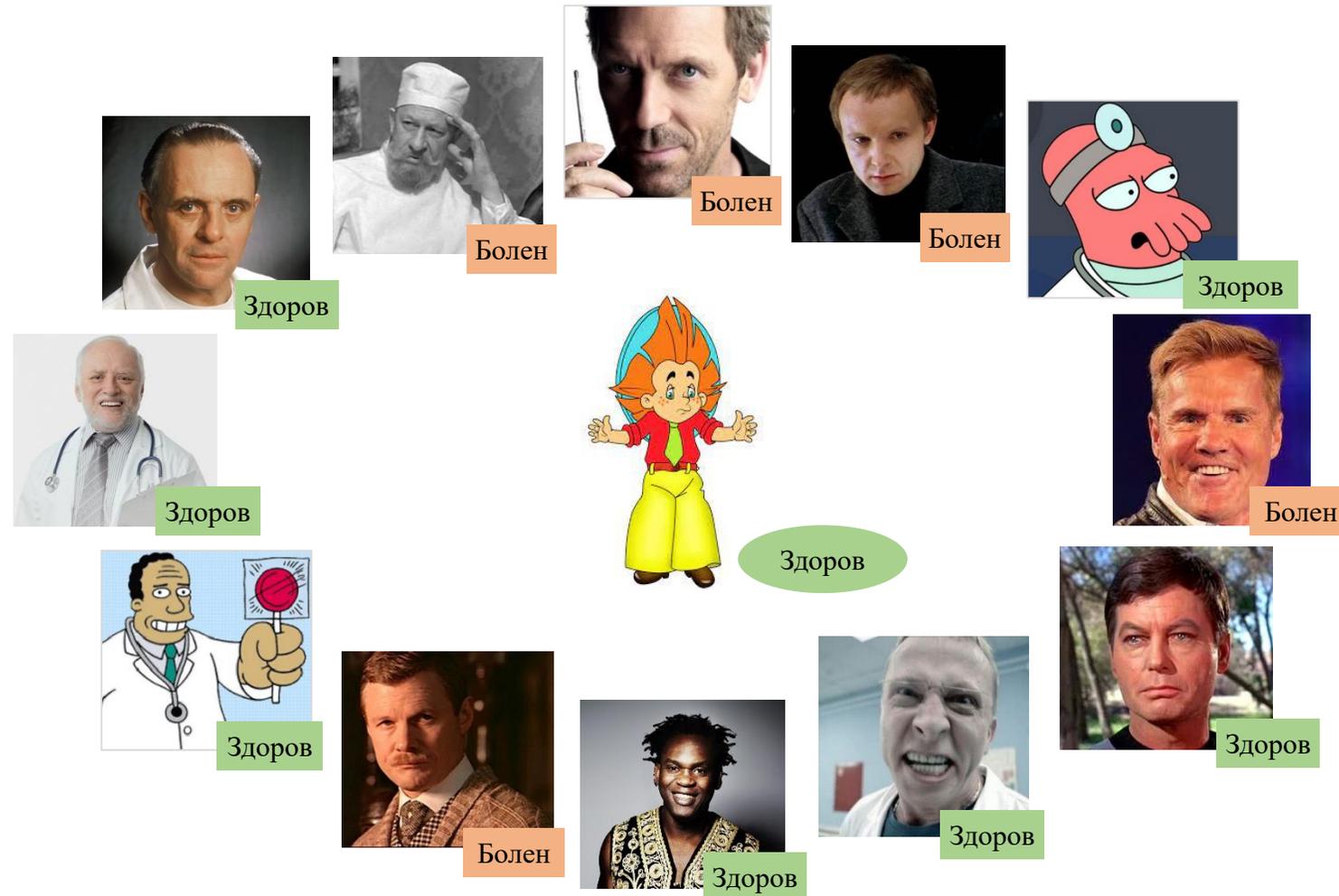
*Жорж Санд*

[Cipolla C.M. The basic laws of human stupidity.](#) Bologna: il Mulino, 2011 ([текст](#))

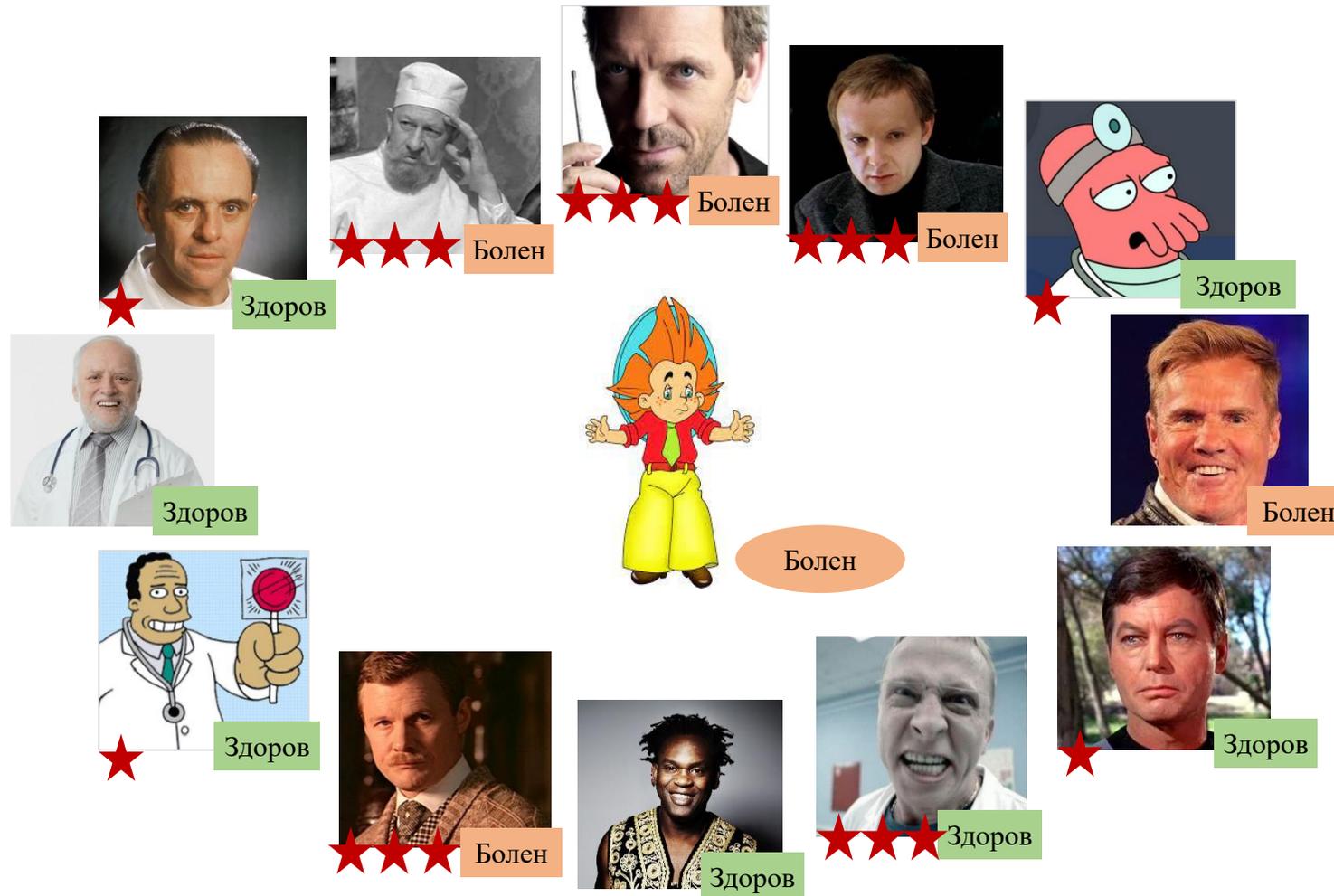
# Содержание

- Основные понятия
- Деревья решений
- Байесовская классификация
- Классификация по ближайшим соседям
- Оценка качества классификации
- **Ансамблевая классификация**
  - **Бэггинг**
  - **Бустинг**
  - **Случайный лес**

# Ансамблевая классификация с мажоритарным голосованием



# Ансамблевая классификация со взвешенным голосованием

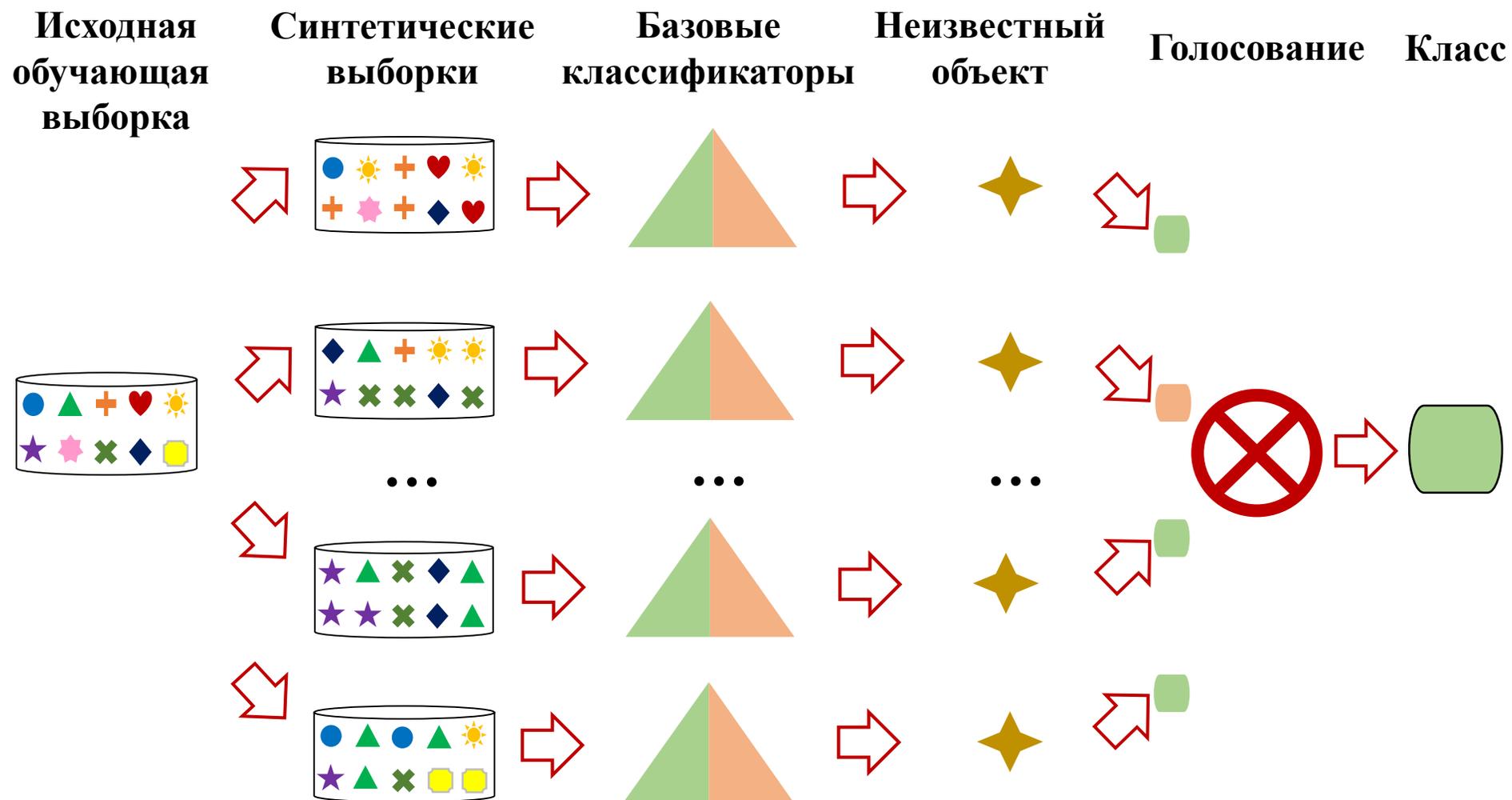


# Ансамблевая классификация

$$|D| = n = |D_i|,$$

$$\forall o \in D: P(o \in D_i) = 1 - (1 - 1/n)^n,$$

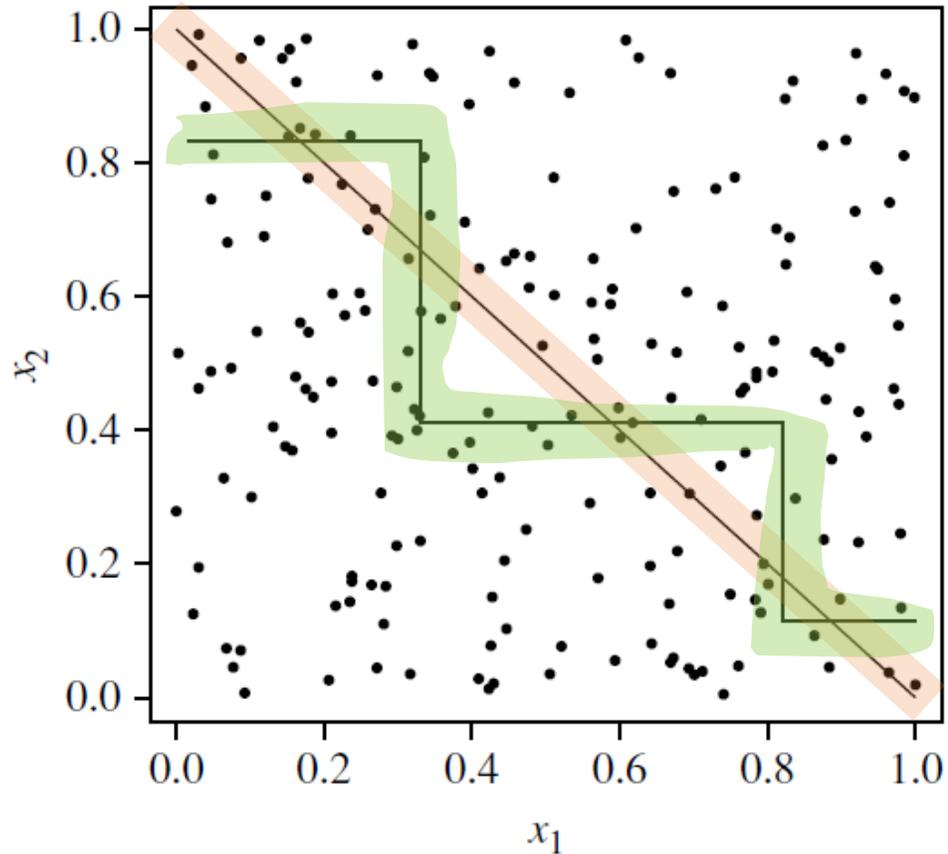
$$\lim_{n \rightarrow +\infty} P(o \in D_i) = 1 - 1/e \approx 0.632$$



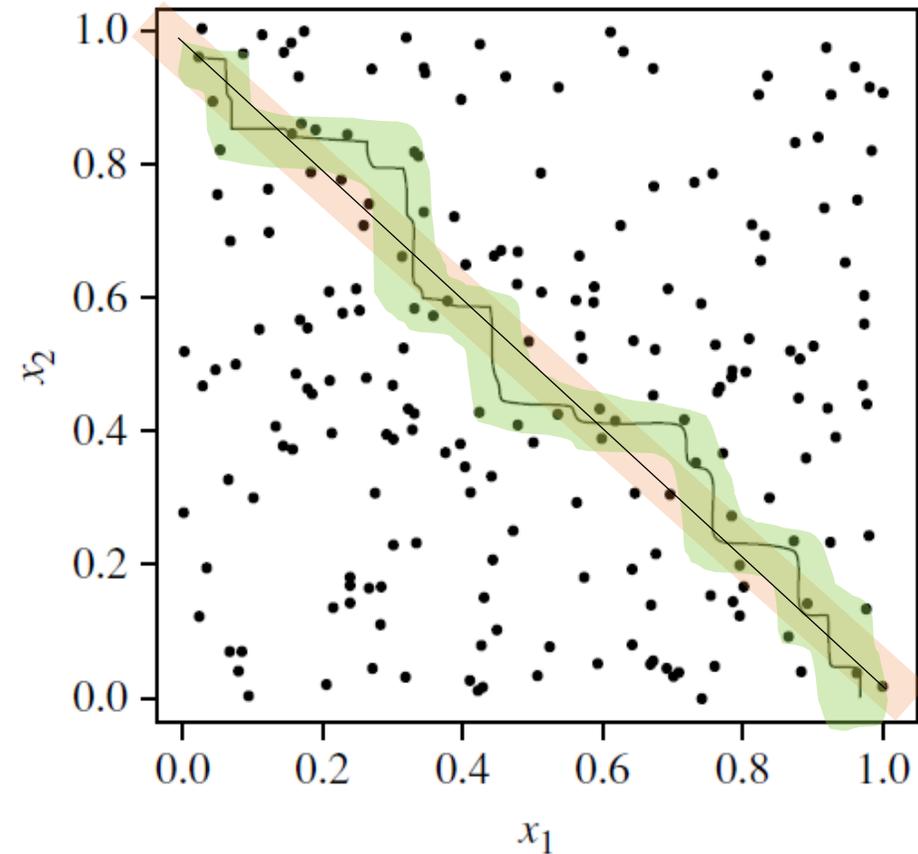
## Виды ансамблевых методов

- Манипуляции с распределением исходной обучающей выборки по выборкам участников
  - Бэггинг (bagging), бустинг (boosting)
- Манипуляции со структурой объектов в выборках участников
  - Случайный лес (random forest)
- Манипуляции с метками классов
  - Кодирование вывода с исправлением ошибок (error-correcting output coding)

# Насколько хорошо работают ансамбли?

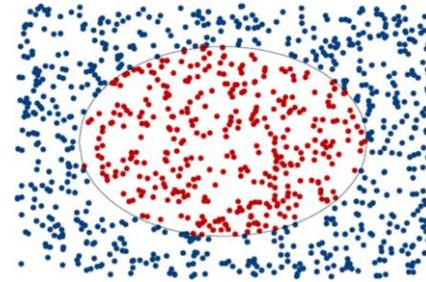


Одно дерево решений

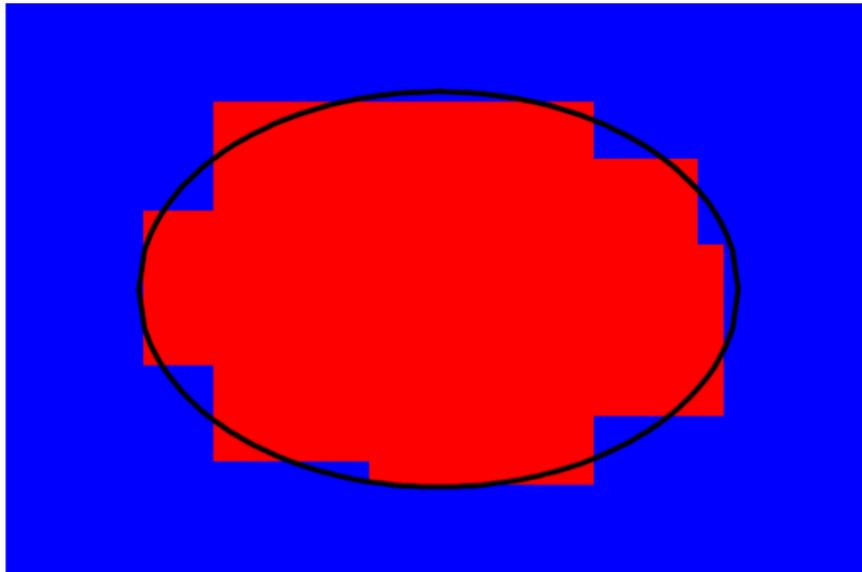


Ансамбль деревьев решений

# Насколько хорошо работают ансамбли?



Исходное множество



Одно дерево решений



Ансамбль деревьев решений

# Почему ансамбли хорошо работают?

- Дано  $n$  базовых классификаторов с вероятностью ошибки  $\varepsilon$  у каждого, *между их ошибками отсутствует корреляция*

- Тогда вероятность ошибки ансамбля

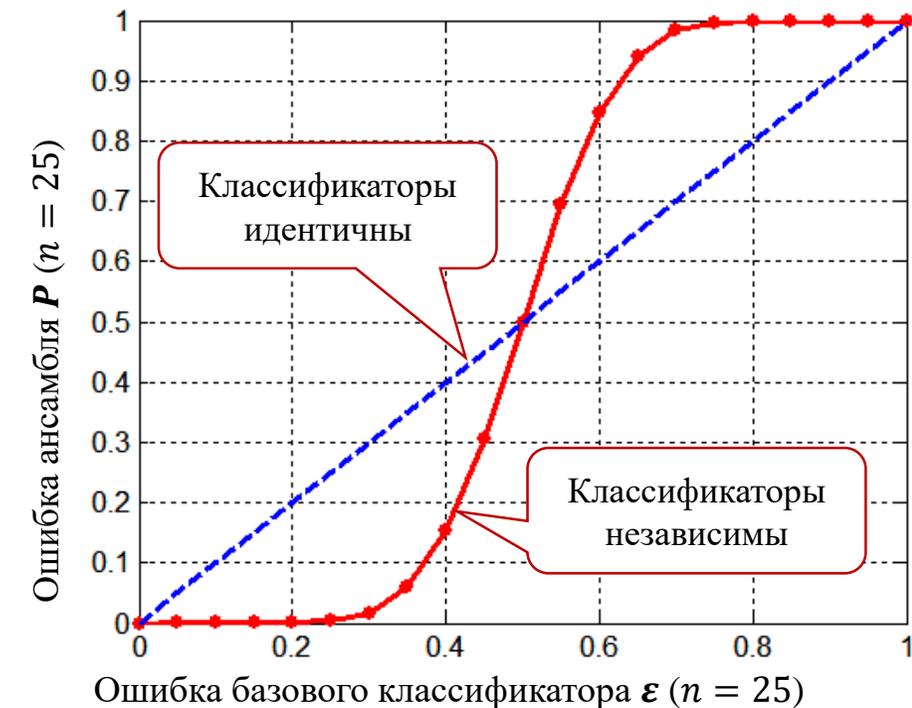
$$P(|wrong| > \lfloor n/2 \rfloor) = \sum_{i=\lfloor n/2 \rfloor + 1}^n C_i^n \cdot \varepsilon^i \cdot (1 - \varepsilon)^{n-i}$$

$$n = 25 \text{ и } \varepsilon = 0.35: P = 0.06 (!!)$$

- **Неравенство Хёфдинга**

$$P(|wrong| > \lfloor n/2 \rfloor) \leq e^{-0.5(2\varepsilon-1)^2}$$

вероятность ошибки ансамбля убывает экспоненциально с ростом числа базовых классификаторов



# Бэггинг (Bagging, Bootstrap Aggregating)

- Базовые идеи
  - Сэмплинг с повторением при формировании выборок
  - Мажоритарное голосование при назначении метки класса
- Отличительные черты
  - Может использоваться для предсказания непрерывных значений (усреднение результатов, выданных участниками ансамбля)
  - Часто существенно более высокая точность, чем у одного классификатора. Лучшая точность при предсказании, чем у одного классификатора
  - Устойчивость к шумам в данных при несущественном снижении точности

# Алгоритм бэггинга

- Вход:

- Обучающая выборка  $D$
- Базовые классификаторы:  $C_1, \dots, C_k$

- Метод

**for**  $i := 1$  **to**  $k$  **do**

Создать случайную выборку с повторением  $D_i, |D_i| = |D|$

Обучить классификатор  $C_i$  на выборке  $D_i$

- Результат

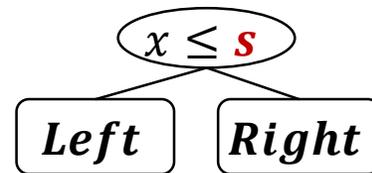
$$C^*(x) = \arg \max_y \sum_{i=1}^k \delta(C_i(x) = y) \quad \delta(\cdot) = \begin{cases} 1, & \cdot - \text{Истина} \\ 0, & \cdot - \text{Ложь} \end{cases}$$

# Пример бэггинга: задача и обучающая выборка

Обучающая  
выборка

$X$	$Class$
0.1	1
0.2	1
0.3	1
0.4	-1
0.5	-1
0.6	-1
0.7	-1
0.8	1
0.9	1
1.0	1

Примитивная  
классификация

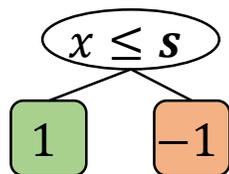


Ансамбль примитивных  
классификаторов

Уч.	$s$	$Left$	$Right$
1	0.75	-1	1
2	?	?	?
3	?	?	?
4	?	?	?
5	?	?	?
6	?	?	?
7	?	?	?
8	?	?	?
9	?	?	?
10	?	?	?

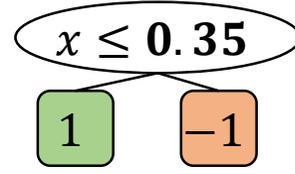
# Пример бэггинга: выборка и обучение 1-го участника ансамбля

$X$	$Class$
0.1	1
0.2	1
0.3	1
0.4	-1
0.5	-1
0.6	-1
0.7	-1
0.8	1
0.9	1
1.0	1

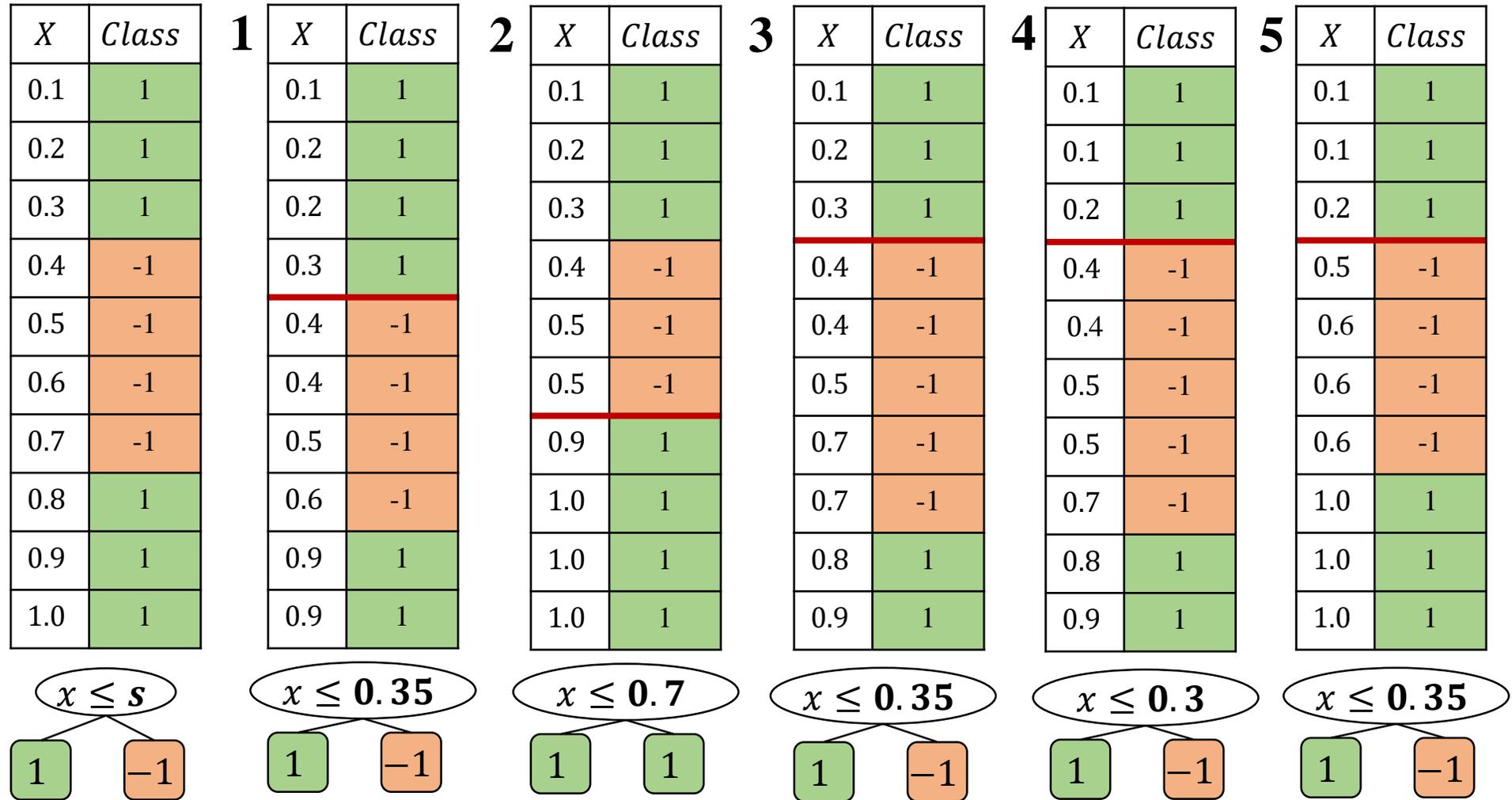


**1**

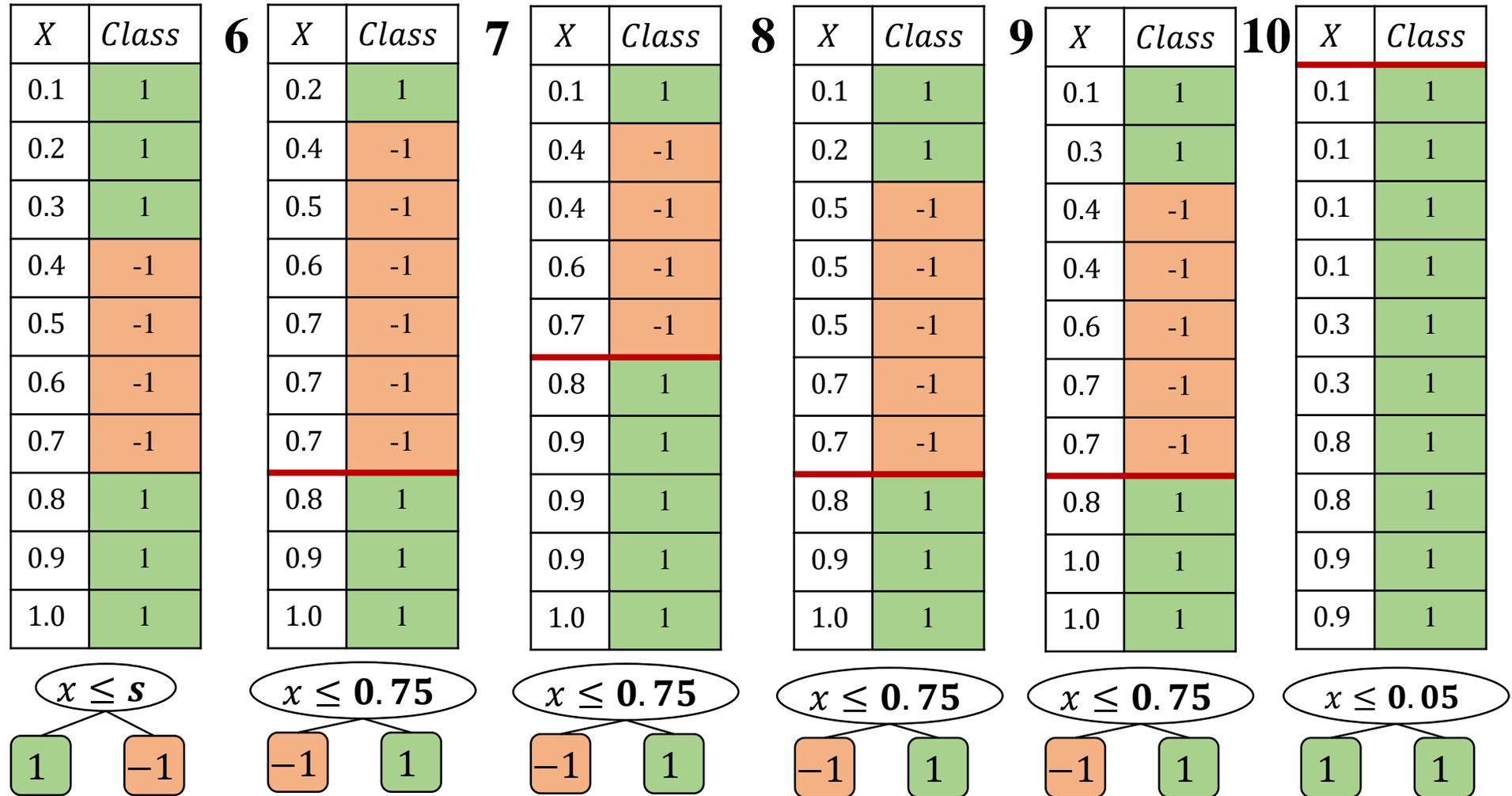
$X$	$Class$
0.1	1
0.2	1
0.2	1
0.3	1
0.4	-1
0.4	-1
0.5	-1
0.6	-1
0.9	1
0.9	1



# Пример бэггинга: выборки и обучение 1-го – 5-го участников ансамбля



# Пример бэггинга: выборки и обучение 6-го – 10-го участников ансамбля



# Пример бэггинга: итоговый ансамбль

Участник	$s$	$L$	$R$
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

# Пример бэггинга: проверка ансамбля

Уч.	$s$	$L$	$R$
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

Уч.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
<b>Vote</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>-6</b>	<b>-6</b>	<b>-6</b>	<b>-6</b>	<b>2</b>	<b>2</b>	<b>2</b>
<b>Class</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>1</b>	<b>1</b>	<b>1</b>

# Достоинства и недостатки бэггинга

- Плюсы
  - Простота реализации
  - Возможность параллельного обучения
  - Увеличение точности по сравнению с одним классификатором, устойчивость к шумам
- Минусы
  - Недетерминированность результата (выборки формируются случайно)
  - Сложность интерпретации результатов по сравнению с одним классификатором
  - Отсутствие строгого математического обоснования условий улучшения прогноза ансамбля

# Boosting: основные идеи

- Веса объектов выборки
  - Вес объекта влияет на вероятность включения объекта в обучающую выборку участника ансамбля
  - Сначала объекты имеют одинаковые веса. Затем вес объекта, неверно классифицированного участником, увеличивается, иначе – уменьшается
- Обучение участников
  - Последовательно (один за другим)
  - Обучающая выборка участника формируется с помощью сэмплинга с замещением
  - Перед переходом к следующему участнику выполняется оценка точности текущего участника на всех объектах исходной выборки и затем пересчитываются их веса
- Классификация
  - Класс неизвестного объекта определяется взвешенным голосованием участников
  - Вес участника зависит от его точности классификации

# Алгоритм AdaBoost

- **Обучающая выборка:**  $(x_1, y_1), \dots, (x_n, y_n)$ , веса  $w_1, \dots, w_n$

- **Ансамбль:**  $C_1, \dots, C_k$

- **Ошибка участника ансамбля:**

$$\varepsilon_i = \frac{1}{n} \sum_{j=1}^n w_j \cdot \delta(C_j(x_i) \neq y_i)$$

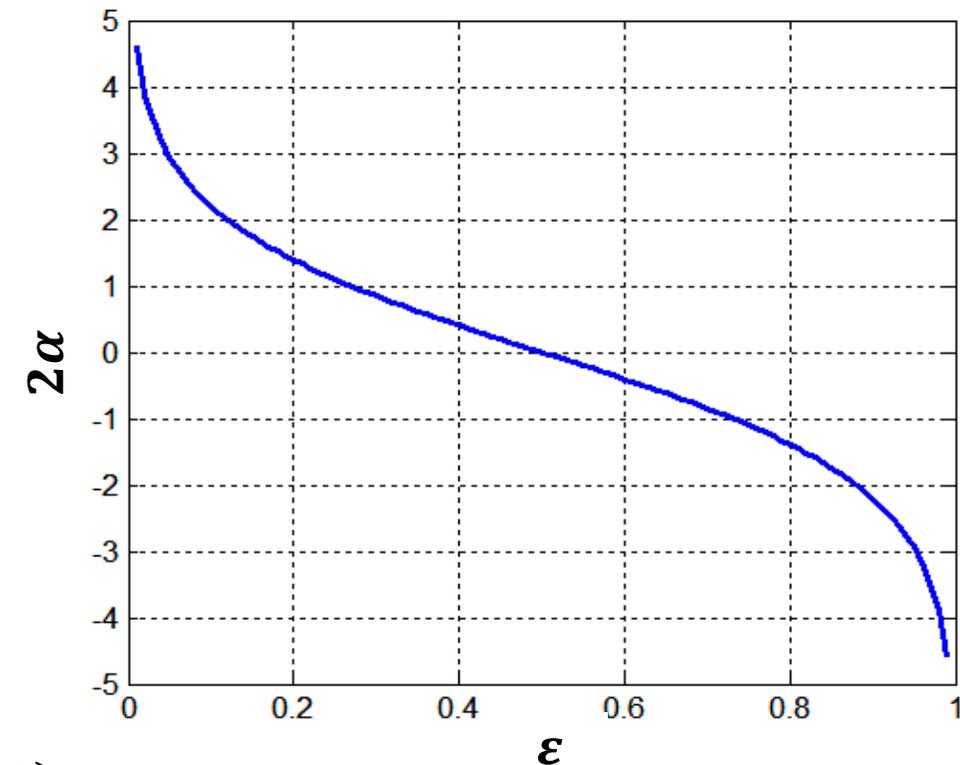
- **Вес участника:**  $\alpha_i = \frac{1}{2} \ln \frac{1-\varepsilon_i}{\varepsilon_i}$

- **Обновление весов:**

$$w_i^{(0)} = 1/n$$

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \cdot \begin{cases} e^{-\alpha_j}, & C_j(x_i) = y_i \\ e^{\alpha_j}, & C_j(x_i) \neq y_i \end{cases}$$

где  $Z_j$  – нормализующий множитель ( $\sum_i w_i^{(j+1)} = 1$ )



# Алгоритм AdaBoost

- 1:  $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$ . {Initialize the weights for all  $N$  examples.}
- 2: Let  $k$  be the number of boosting rounds.
- 3: **for**  $i = 1$  to  $k$  **do**
- 4:   Create training set  $D_i$  by sampling (with replacement) from  $D$  according to  $\mathbf{w}$ .
- 5:   Train a base classifier  $C_i$  on  $D_i$ .
- 6:   Apply  $C_i$  to all examples in the original training set,  $D$ .
- 7:    $\epsilon_i = \frac{1}{N} [\sum_j w_j \delta(C_i(x_j) \neq y_j)]$  {Calculate the weighted error.}
- 8:   **if**  $\epsilon_i > 0.5$  **then**
- 9:      $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$ .
- 10:    Go back to Step 4.
- 11:   **end if**
- 12:    $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$ .
- 13:   Update the weight of each example
- 14: **end for**
- 15:  $C^*(\mathbf{x}) = \operatorname{argmax}_y \sum_{j=1}^T \alpha_j \delta(C_j(\mathbf{x}) = y)$ .

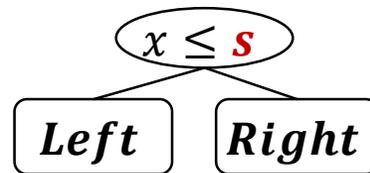
Если ошибка классификации более 50%,  
то повторить сэмплинг

# Пример бустинга

Обучающая  
выборка

$X$	$Class$
0.1	1
0.2	1
0.3	1
0.4	-1
0.5	-1
0.6	-1
0.7	-1
0.8	1
0.9	1
1.0	1

Примитивная  
классификация

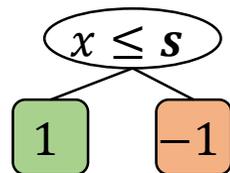


Ансамбль примитивных  
классификаторов

Уч.	$s$	$Left$	$Right$	$\alpha$
1	0.75	-1	1	?
2	?	?	?	?
3	?	?	?	?

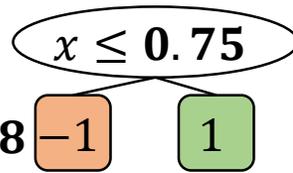
# Пример бустинга: выборки и обучение участников ансамбля

$X$	$Class$
0.1	1
0.2	1
0.3	1
0.4	-1
0.5	-1
0.6	-1
0.7	-1
0.8	1
0.9	1
1.0	1



**1**

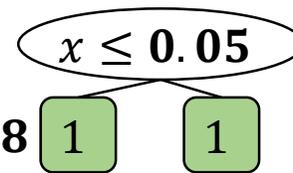
$X$	$Class$	$weight$
0.1	1	0.1
0.4	-1	0.1
0.5	-1	0.1
0.6	-1	0.1
0.6	-1	0.1
0.6	-1	0.1
0.7	-1	0.1
0.7	-1	0.1
0.7	-1	0.1
0.8	1	0.1
1.0	1	0.1



$\alpha = 1.738$

**2**

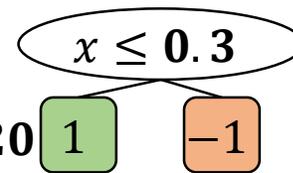
$X$	$Class$	$weight$
0.1	1	0.311
0.1	1	0.311
0.2	1	0.311
0.2	1	0.01
0.2	1	0.01
0.2	1	0.01
0.2	1	0.01
0.3	1	0.01
0.3	1	0.01
0.3	1	0.01



$\alpha = 2.778$

**3**

$X$	$Class$	$weight$
0.2	1	0.029
0.2	1	0.029
0.4	-1	0.029
0.4	-1	0.228
0.4	-1	0.228
0.4	-1	0.228
0.5	-1	0.228
0.6	-1	0.009
0.6	-1	0.009
0.7	-1	0.009



$\alpha = 4.120$

# Пример бустинга: проверка ансамбля

Уч.	$s$	$L$	$R$	$\alpha$
1	0.75	-1	1	1.738
2	0.05	1	1	2.778
3	0.3	1	-1	4.120

Уч.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	-1	-1	-1	-1	-1	-1	-1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
<b>SUM</b>	5.16	5.16	5.16	-3.08	-3.08	-3.08	-3.08	0.397	0.397	0.397
<b>Class</b>	1	1	1	-1	-1	-1	-1	1	1	1

## Бустинг: ошибка обучения ансамбля

- $\varepsilon_{ensemble} \leq \prod_i \sqrt{\varepsilon_i(1 - \varepsilon_i)}$
- Если ошибка участника ансамбля  $\varepsilon_i < 0.5$ , то мы можем ввести  $\gamma_i$  – меру улучшения классификации участником по сравнению с угадыванием:  
 $\varepsilon_i = 0.5 - \gamma_i$ . Тогда

$$\varepsilon_{ensemble} \leq \prod_i \sqrt{1 - 4\gamma_i^2} \leq e^{-2 \sum_i \gamma_i^2}$$

- Если  $\exists \gamma^* \forall i \gamma_i < \gamma^*$ , то ошибка обучения ансамбля убывает экспоненциально (быстрая сходимость алгоритма)

# Случайный лес (Random Forest)

- Вариация бэггинга
  - Участники ансамбля – деревья решений
  - Для обучения участника выбирается случайное подмножество атрибутов исходной выборки (обычно мощность одна для всех участников)
    - Forest-RI: большое число атрибутов  $p$ , число выбираемых атрибутов  $m = \lceil \log_2 p \rceil$  или  $m = \lceil \sqrt{p} \rceil$
    - Forest-RC: малое число атрибутов; генерация  $f$  искусственных атрибутов как линейных комбинаций случайных коэффициентов, равномерно распределенных в отрезке  $[-1, 1]$  и выбор из них  $m$  случайных атрибутов
  - Класс неизвестного объекта определяется мажоритарным голосованием участников
- Основная идея
  - Увеличение вариации деревьев в ансамбле, снижение корреляции результатов их классификации;
  - Блокада потенциально доминирующих атрибутов (с вероятностью  $\frac{p-m}{p}$ )
- Преимущества и недостатки
  - Высокая скорость обучения; распараллеливаемость
  - Устойчивость к выбросам и несбалансированным данным
  - Неинтерпретируемость, высокие требования к памяти

# Сравнение точности ансамблевых методов

Data Set	Number of (Attributes, Classes, Records)	Decision Tree (%)	Bagging	Boosting	RF
			(%) <b>№ 3</b>	(%) <b>№ 2</b>	(%) <b>№ 1</b>
Anneal	(39, 6, 898)	92.09	94.43	95.43	95.43
Australia	(15, 2, 690)	85.51	87.10	85.22	85.80
Auto	(26, 7, 205)	81.95	85.37	85.37	84.39
Breast	(11, 2, 699)	95.14	96.42	97.28	96.14
Cleve	(14, 2, 303)	76.24	81.52	82.18	82.18
Credit	(16, 2, 690)	85.8	86.23	86.09	85.8
Diabetes	(9, 2, 768)	72.40	76.30	73.18	75.13
German	(21, 2, 1000)	70.90	73.40	73.00	74.5
Glass	(10, 7, 214)	67.29	76.17	77.57	78.04
Heart	(14, 2, 270)	80.00	81.48	80.74	83.33
Hepatitis	(20, 2, 155)	81.94	81.29	83.87	83.23
Horse	(23, 2, 368)	85.33	85.87	81.25	85.33
Ionosphere	(35, 2, 351)	89.17	92.02	93.73	93.45
Iris	(5, 3, 150)	94.67	94.67	94.00	93.33
Labor	(17, 2, 57)	78.95	84.21	89.47	84.21
Led7	(8, 10, 3200)	73.34	73.66	73.34	73.06
Lymphography	(19, 4, 148)	77.03	79.05	85.14	82.43
Pima	(9, 2, 768)	74.35	76.69	73.44	77.60
Sonar	(61, 2, 208)	78.85	78.85	84.62	85.58
Tic-tac-toe	(10, 2, 958)	83.72	93.84	98.54	95.82
Vehicle	(19, 4, 846)	71.04	74.11	78.25	74.94
Waveform	(22, 3, 5000)	76.44	83.30	83.90	84.04
Wine	(14, 3, 178)	94.38	96.07	97.75	97.75
Zoo	(17, 7, 101)	93.07	93.07	95.05	97.03

# Литература

- Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd Edition. Morgan Kaufmann, 2011. 740 p. ISBN: 978-0123814791  
8.6 Techniques to Improve Classification Accuracy, pp. 377-385
- Tan P.-N., Steinbach M., Karpatne A., Kumar V. Introduction to Data Mining. 2nd Edition. Pearson, 2019. 839 p. ISBN: 978-0-13-312890-1  
4.10 Ensemble Methods, pp. 296-312