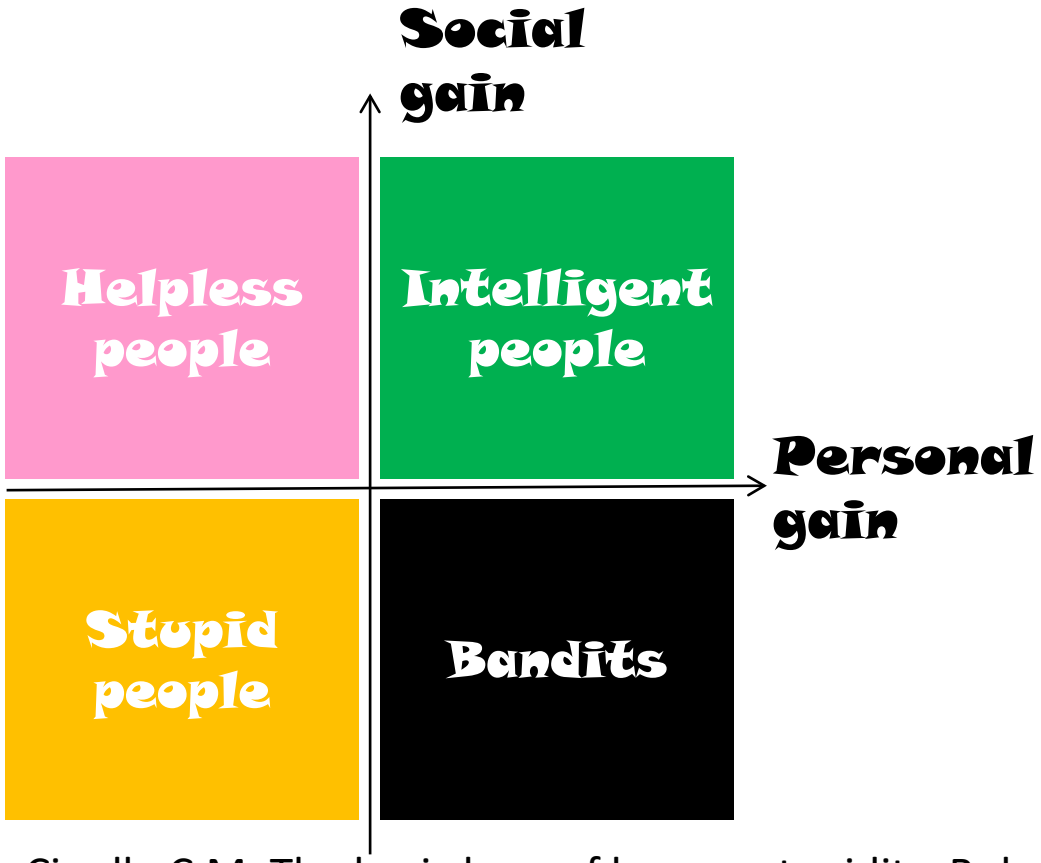
# Задача классификации данных

*Классификация – нить Ариадны в лабиринте природы.*

*Жорж Санд*




Cipolla C.M. The basic laws of human stupidity. Bologna: il Muli, 2011

# Содержание

- **Общий подход к классификации**
- **Деревья решений**
- Ансамблевая классификация
- Оценка качества классификации

# Задача классификации

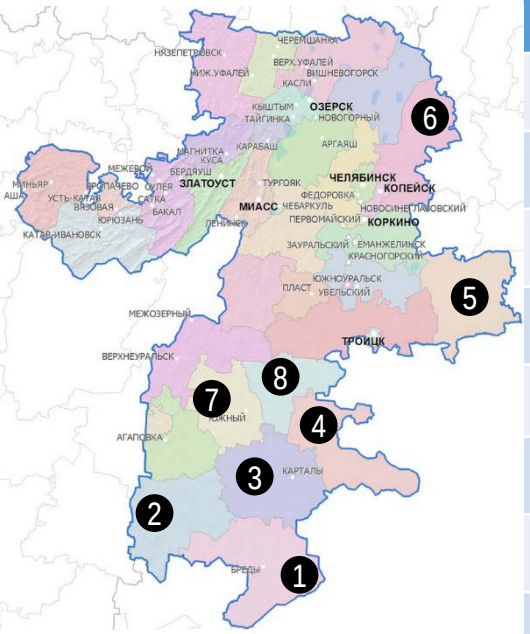
- Построение формальной модели, которая распределяет объекты, имеющие одинаковую структуру по заранее известным группам (классам) в зависимости от похожести атрибутов объектов
- Основные задачи классификации
  - **Предсказание:** назначить корректный класс объекту, который предварительно не был рассмотрен
  - **Описание:** указать способ, с помощью которого можно отличать объекты различных классов

# Пример: анализ оттока клиентов

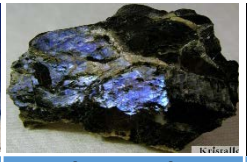
| # | Sex | Age | Day calls | Day charge | Eve calls | Eve charge | Night calls | Night charge | Intl calls | Intl charge | Plan | ... | CHURN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | M | 24 | 110 | 45.07 | 99 | 16.78 | 91 | 11.01 | 3 | 2.7 | A | | YES |
| 2 | M | 28 | 123 | 27.47 | 103 | 16.62 | 103 | 11.45 | 3 | 3.7 | B | | NO |
| 3 | F | 29 | 114 | 41.38 | 110 | 10.3 | 104 | 7.32 | 5 | 3.29 | C | | YES |
| 4 | M | 33 | 71 | 50.9 | 88 | 5.26 | 89 | 8.86 | 7 | 1.78 | A | | YES |
| 5 | M | 53 | 113 | 28.34 | 122 | 12.61 | 121 | 8.41 | 3 | 2.73 | C | | NO |
| 6 | M | 37 | 98 | 37.98 | 101 | 18.75 | 118 | 9.18 | 6 | 1.7 | A | | YES |
| 7 | F | 78 | 88 | 37.09 | 108 | 29.62 | 118 | 9.57 | 7 | 2.03 | B | | NO |
| 8 | M | 63 | 79 | 26.69 | 94 | 8.76 | 96 | 9.53 | 6 | 1.92 | B | | NO |
| 9 | F | 46 | 97 | 31.37 | 80 | 29.89 | 90 | 9.71 | 4 | 2.35 | A | | NO |
| | | | | | ... | | | | | | | | |

Определить, уйдет ли клиент к другому оператору

# Пример: поиск полезных ископаемых





| # | $FeSO_4 \bullet 7H_2O$ | | $NH_4H_2PO_4$ | | $(Na,Ca)(Si,Al)_4O_8$ | | $SiO_2 \bullet nH_2O$ | | $LiAlSi_4O_{10}$ | | … | Минерал |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Наличие | Плотность | Наличие | Плотность | Наличие | Плотность | Наличие | Плотность | Наличие | Плотность | | |
| 1 | Да | 3.40 | Нет | - | Да | 7.80 | Нет | - | Да | 23.92 | | Железо |
| 2 | Нет | - | Да | 7.22 | Да | 2.97 | Да | 5.97 | Да | 16.54 | | Медь |
| 3 | Да | 4.67 | Да | 5.45 | Да | 5.43 | Да | 8.95 | Да | 28.49 | | Серебро |
| 4 | Нет | - | Да | 3.12 | Нет | - | Да | 9.12 | Нет | - | | Цинк |
| 5 | Да | 2.78 | Да | 0.18 | Нет | - | Нет | - | Да | 25.02 | | Железо |
| 6 | Да | 1.02 | Нет | - | Нет | - | Да | 1.23 | Да | 2.12 | | НЕТ |
| 7 | Да | 0.75 | Нет | - | Нет | - | Да | 3.10 | Да | 2.99 | | НЕТ |
| 8 | Нет | - | Да | 0.36 | Да | 2.08 | Нет | - | Нет | - | | НЕТ |

# Пример: классификация позвоночных

| Vertebrate Name | Body Temperature | Skin Cover | Gives Birth | Aquatic Creature | Aerial Creature | Has Legs | Hiber-nates | Class Label |
|---|---|---|---|---|---|---|---|---|
| human | warm-blooded | hair | yes | no | no | yes | no | mammal |
| python | cold-blooded | scales | no | no | no | no | yes | reptile |
| salmon | cold-blooded | scales | no | yes | no | no | no | fish |
| whale | warm-blooded | hair | yes | yes | no | no | no | mammal |
| frog | cold-blooded | none | no | semi | no | yes | yes | amphibian |
| komodo dragon | cold-blooded | scales | no | no | no | yes | no | reptile |
| bat | warm-blooded | hair | yes | no | yes | yes | yes | mammal |
| pigeon | warm-blooded | feathers | no | no | yes | yes | no | bird |
| cat | warm-blooded | fur | yes | no | no | yes | no | mammal |
| leopard shark | cold-blooded | scales | yes | yes | no | no | no | fish |
| turtle | cold-blooded | scales | no | semi | no | yes | no | reptile |
| penguin | warm-blooded | feathers | no | semi | no | yes | no | bird |
| porcupine | warm-blooded | quills | yes | no | no | yes | yes | mammal |
| eel | cold-blooded | scales | no | yes | no | no | no | fish |
| salamander | cold-blooded | none | no | semi | no | yes | yes | amphibian |

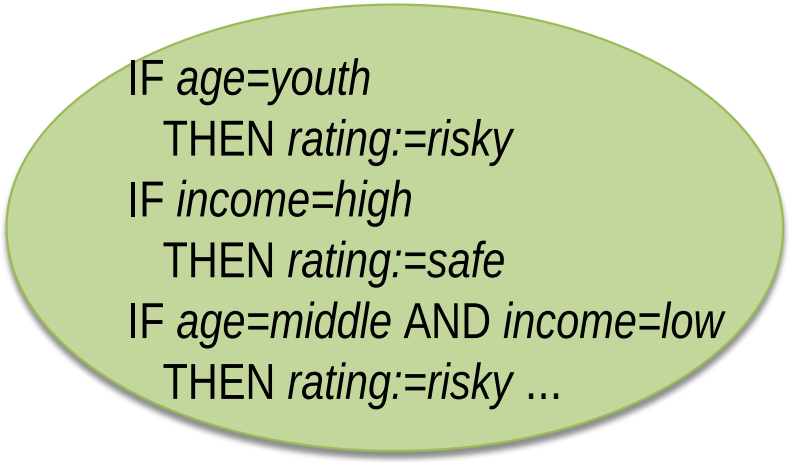
# Атрибуты и метки класса

| Приложение | Набор атрибутов | Метка класса |
|---|---|---|
| Кредитный скоринг клиентов | Пол, возраст и доход клиента, величина, процент и срок кредита | Надежный / ненадежный |
| Предсказание успеваемости студентов | Пол, курс, местный/приезжий, количество посещений, количество сданных заданий | Отлично / хорошо / удовлетворительно / неудовлетворительно |
| Выявление спама | Характеристики, полученные из заголовка и тела сообщения | Спам / не спам |
| Идентификация опухоли | Характеристики, полученные из снимков МРТ | Злокачественная / доброкачественная |
| Классификация галактик | Характеристики, полученные из снимков с телескопа | Эллиптическая / спиральная / нерегулярной формы |

# Процесс классификации: обучение (индукция)

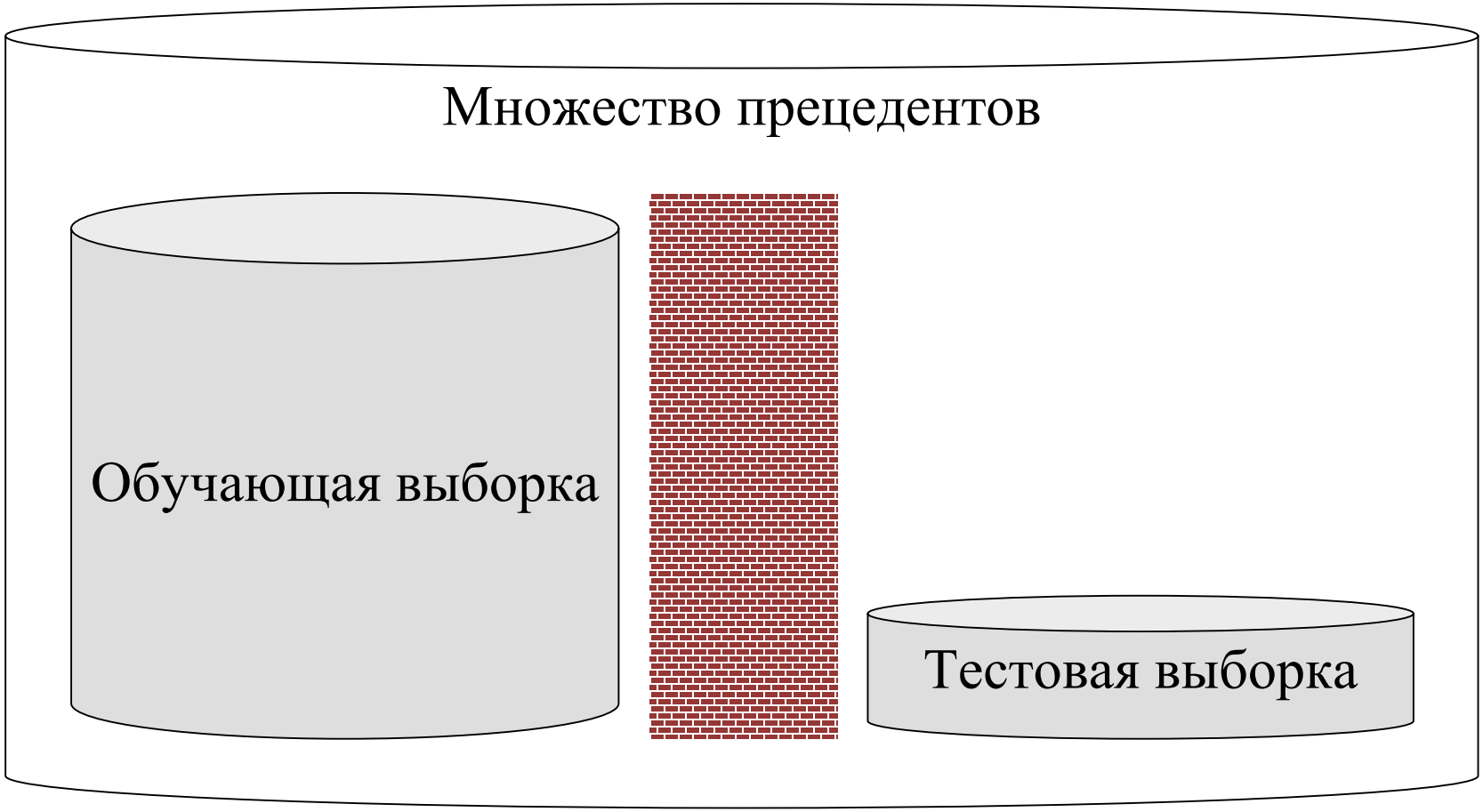
Обучающая выборка

| Name | Income | Age | Credit rating |
|---|---|---|---|
| Peter Parker | low | youth | risky |
| Anakin Skywalker | low | youth | risky |
| Tony Stark | high | middle | safe |
| Han Solo | low | middle | risky |
| Clark Kent | low | senior | risky |
| James Bond | medium | senior | risky |
| Harry Callahan | high | middle | safe |
| Bruce Banner | high | senior | safe |

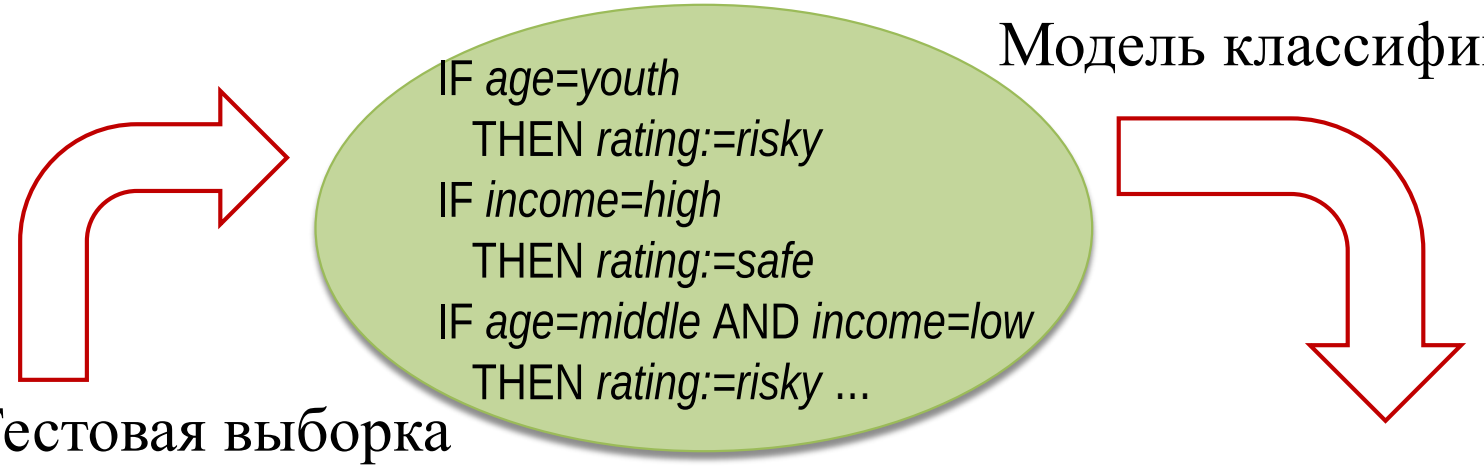




Модель классификации

# Процесс классификации: обучающая *vs.* тестовая выборки



# Процесс классификации: оценка модели




Тестовая выборка

| Name | Income | Age | Credit rating |
|---|---|---|---|
| Kyp Durron | low | youth | risky |
| Mace Windu | low | youth | risky |
| Lea Organa | high | middle | safe |
| Darth Vader | low | middle | risky |
| Joulupukki | low | senior | safe |

| Credit rating |
|---|
| risky |
| risky |
| safe |
| risky |
| risky |

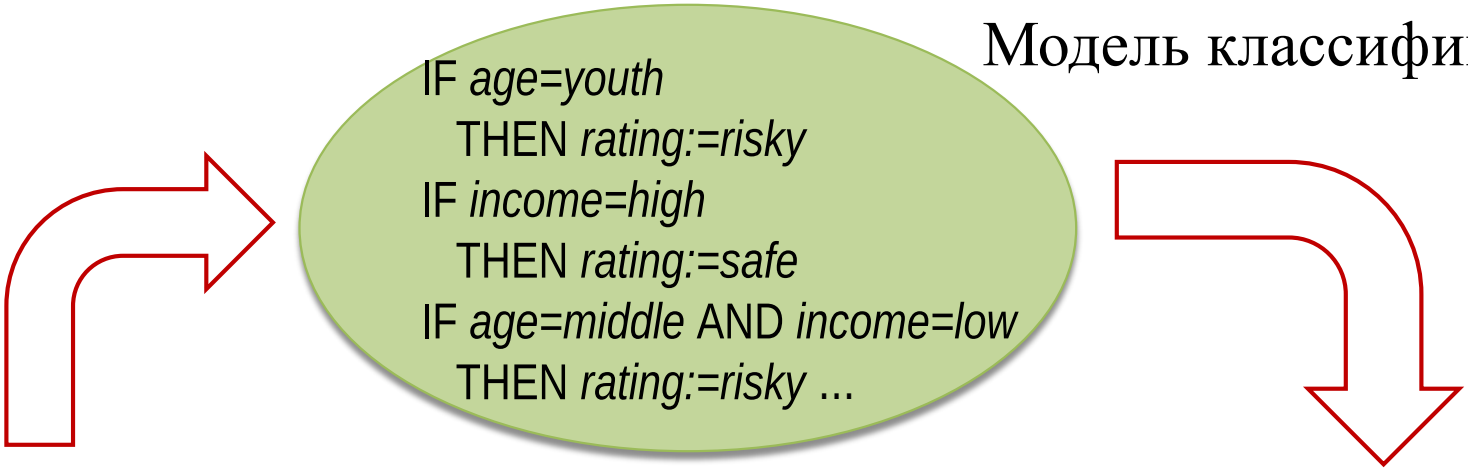
Точность 80%

# Оценка модели: матрица ошибок

| | | Реальный класс | |
|---|---|---|---|
| | | **A** | **B** |
| Предсказанный класс | **A** | $TP$ | $FP$ |
| | **B** | $FN$ | $TN$ |

- $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$
- $Error\ rate = \frac{FP+FN}{TP+FP+FN+TN}$

# Процесс классификации: применение (дедукция)




Неизвестные данные

| Name | Income | Age |
|---|---|---|
| John Doe | low | senior |
| Matti Meikäläinen | high | middle |
| Ivan Ivanov | high | youth |
| Pietro Sconosciuta | low | middle |
| ... | | |

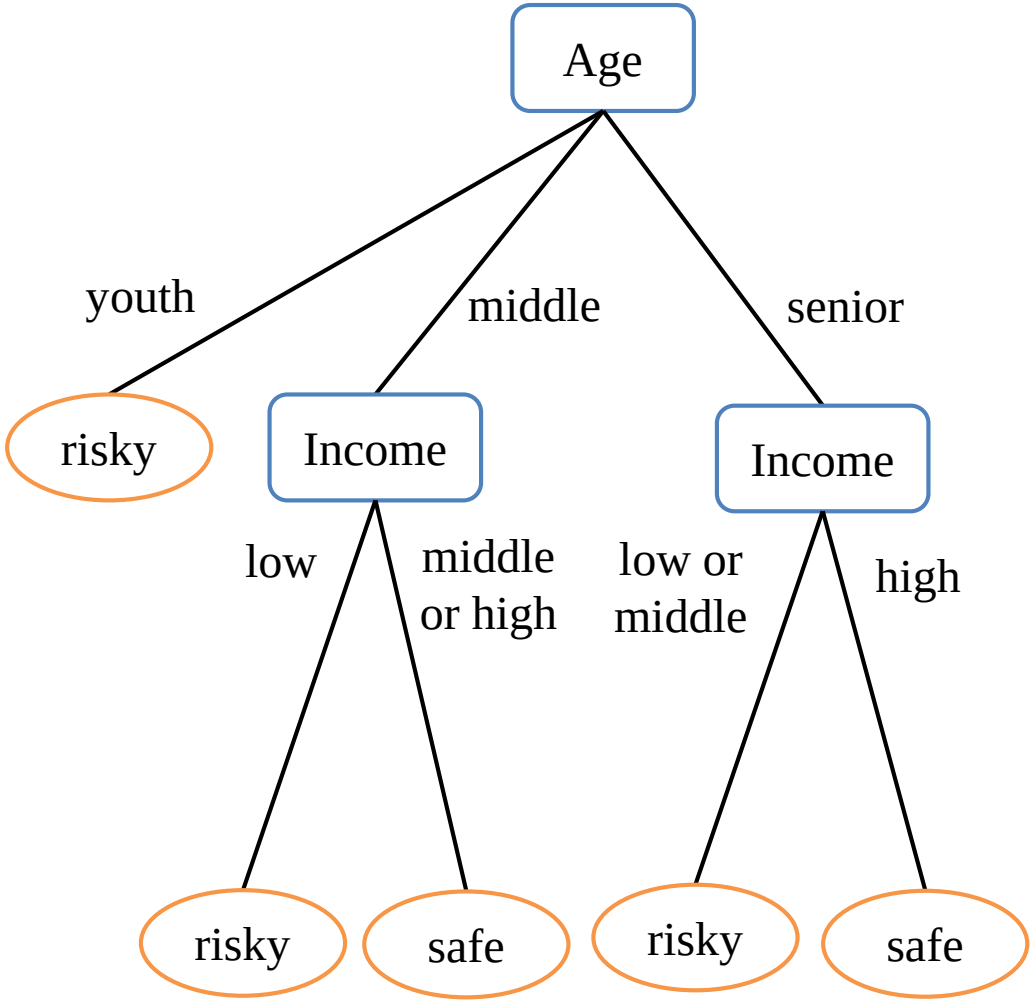
| Credit rating |
|---|
| |
| |
| |
| |
| … |

# Содержание

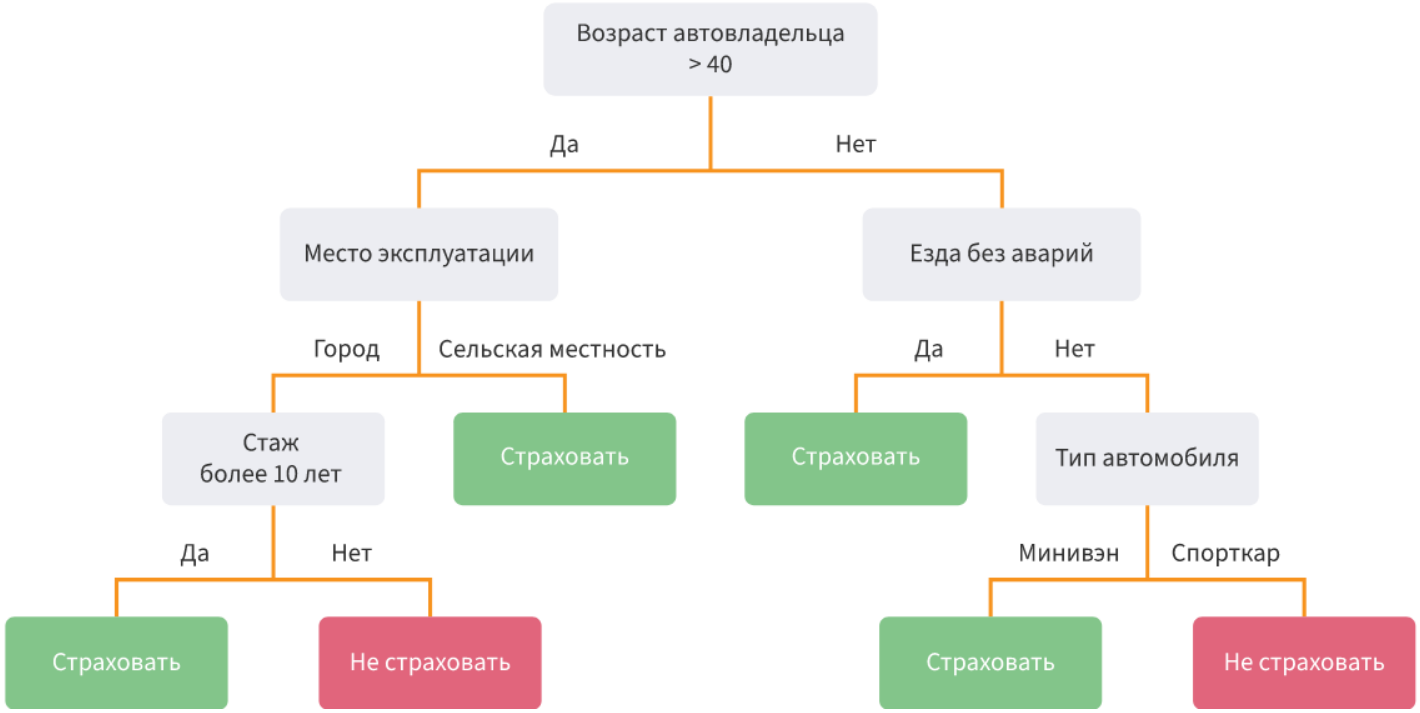
- Общий подход к классификации
- **Деревья решений**
- Ансамблевая классификация
- Оценка качества классификации

# Деревья решений

- Дерево решений – модель классификации в виде дерева, которое имеет
  - корневой узел и внутренние узлы: проверка условия на атрибут объекта
  - узлы-листья: метки классов
  - ребра: переходы по результату проверки



# Деревья решений




- Для применения не требуется компьютер
- Сферы применения: банковское дело, страхование, торговля, медицина, контроль качества продукции

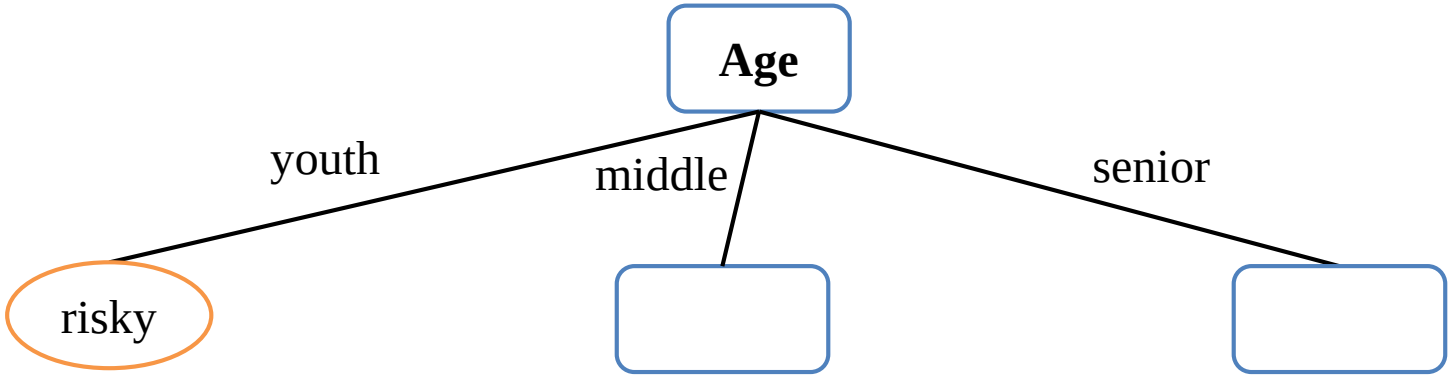
# Построение дерева решений

NULL

| Name | Income | Age | Credit rating |
|---|---|---|---|
| Peter Parker | low | youth | risky |
| Anakin Skywalker | low | youth | risky |
| Tony Stark | high | middle | safe |
| Han Solo | low | middle | risky |
| Clark Kent | low | senior | risky |
| James Bond | medium | senior | risky |
| Harry Callahan | high | middle | safe |
| Bruce Banner | high | senior | safe |

- Если все объекты из одного класса, то создать лист с меткой этого класса, иначе выбрать атрибут для разбиения

# Построение дерева решений




| Name | Income | Credit rating |
|---|---|---|
| Peter Parker | low | risky |
| Anakin Skywalker | low | risky |

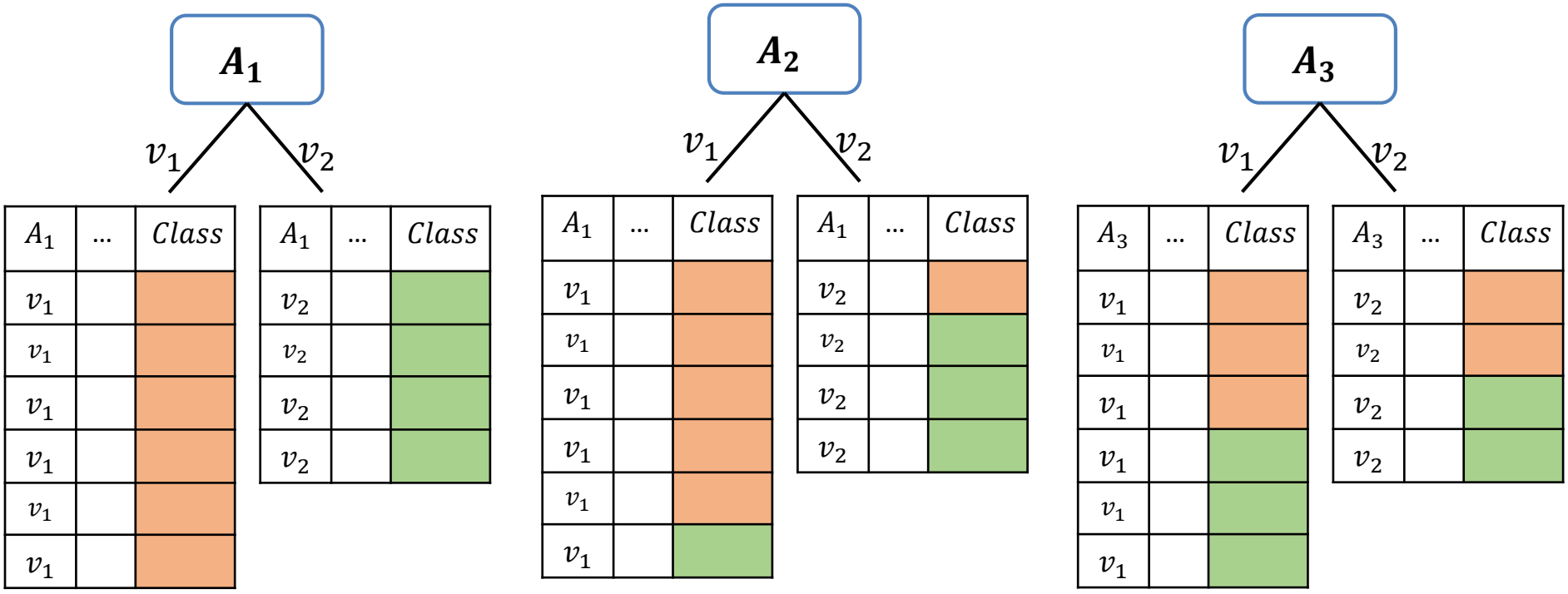
| Name | Income | Credit rating |
|---|---|---|
| Tony Stark | high | safe |
| Han Solo | low | risky |
| Harry Callahan | high | safe |

| Name | Income | Credit rating |
|---|---|---|
| Clark Kent | low | risky |
| James Bond | medium | risky |
| Bruce Banner | high | safe |

- Разбить выборку в соответствии со значениями выбранного атрибута
- Рекурсивно построить поддеревья

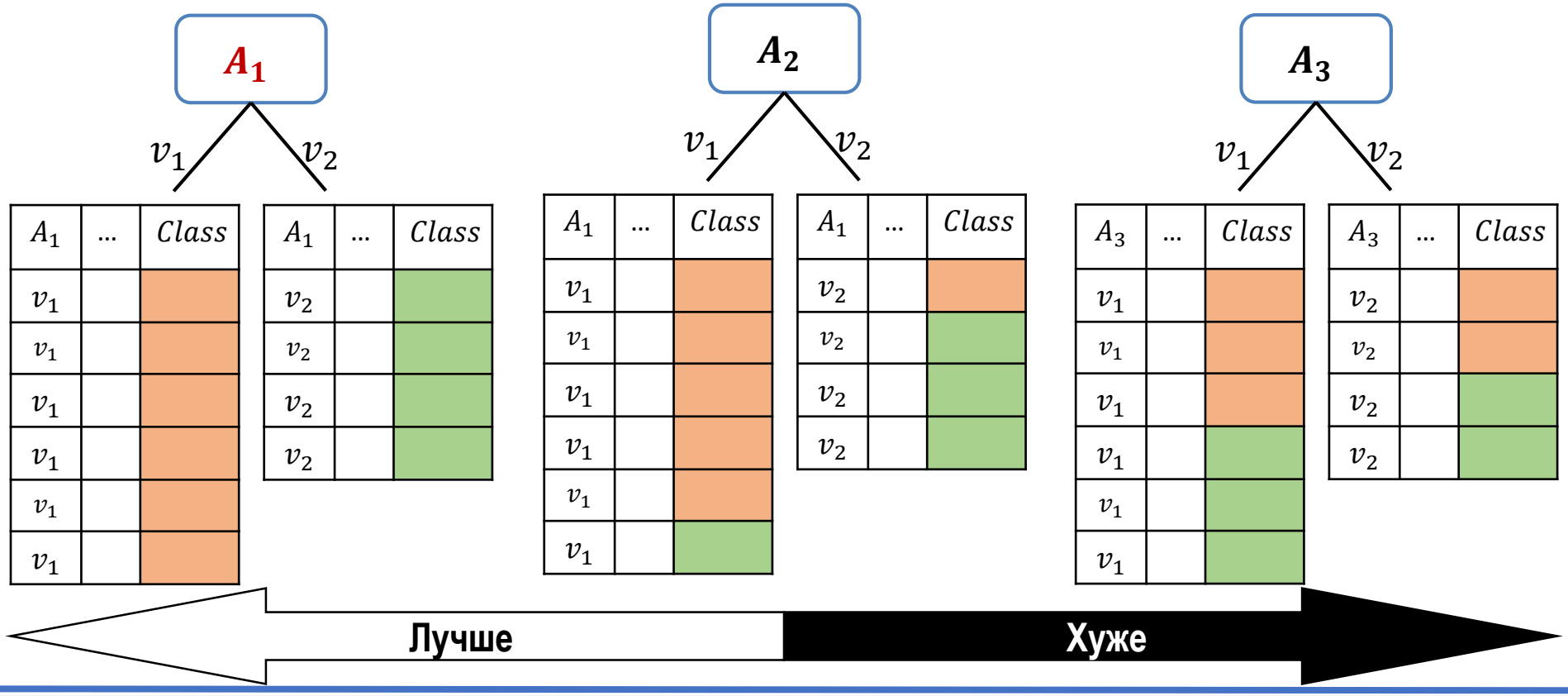
# Выбор атрибута разбиения

- Какой атрибут выбрать для разбиения?

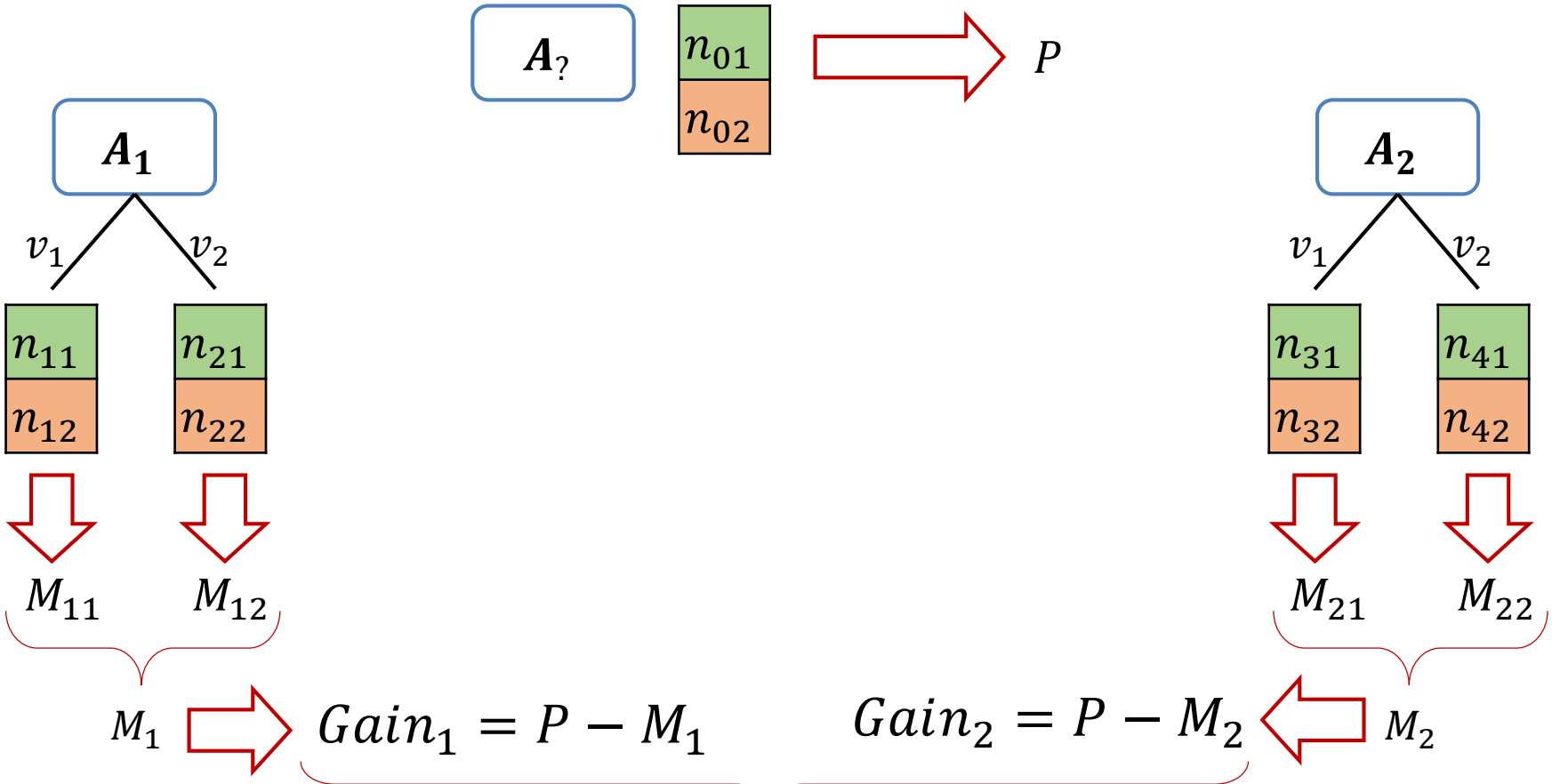


# Выбор атрибута разбиения

- *Жадный подход*: выбрать атрибут, разбивающий выборку на подмножества с минимальной долей «примесей» (объектов иного класса)



# Критерий выбора атрибута – прирост информации



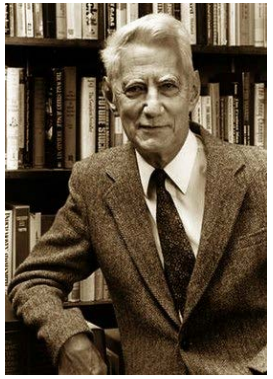

$$A_? = \arg\max\{Gain_1, Gain_2\}$$

# Оценка доли примесей с помощью энтропии

$$Entropy = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$$

$n$ – количество классов

$p_i$ – вероятность объектов $i$-го класса в выборке узла



**Клод Шеннон**
1916-2001

**to be, or not to be, that is the question**

| i | Sym | num | p | -p*log2(p) |
|---|---|---|---|---|
| 1 | space | 9 | 0.22 | 0.48 |
| 2 | comma | 2 | 0.05 | 0.21 |
| 3 | a | 1 | 0.02 | 0.13 |
| 4 | b | 2 | 0.05 | 0.21 |
| 5 | e | 4 | 0.10 | 0.33 |
| 6 | h | 2 | 0.05 | 0.21 |
| 7 | i | 2 | 0.05 | 0.21 |
| 8 | n | 2 | 0.05 | 0.21 |
| 9 | o | 5 | 0.12 | 0.37 |
| 10 | q | 1 | 0.02 | 0.13 |
| 11 | r | 1 | 0.02 | 0.13 |
| 12 | s | 2 | 0.05 | 0.21 |
| 13 | t | 7 | 0.17 | 0.44 |
| 14 | u | 1 | 0.02 | 0.13 |
| | | 41 | | **3.41** |

**aaaaaaaaaaaaaa…a**

| i | Sym | num | p | -p*log2(p) |
|---|---|---|---|---|
| 1 | a | 41 | 1.00 | 0.00 |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| 11 | | | | |
| 12 | | | | |
| 13 | | | | |
| 14 | | | | |
| | | 41 | | **0.00** |

# Алгоритм ID3 (Iterative Dichotomiser 3)

- $Entropy = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$
- $\max Entropy = \log_2 n$, когда объекты равномерно распределены по классам (наименее желательная ситуация)
- $\min Entropy = 0$, когда объекты принадлежат одному классу (наиболее желательная ситуация)

**John Ross Quinlan**

# Вычисление энтропии узла дерева решений

- $Entropy = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$

$0 \cdot \log_2 0$ считается 0

- 0 / 6 $Entropy = -\frac{0}{6} \cdot \log_2 0 - \frac{6}{6} \cdot \log_2 \frac{1}{1} = 0$
- 1 / 5 $Entropy = -\frac{1}{6} \cdot \log_2 \frac{1}{6} - \frac{5}{6} \cdot \log_2 \frac{5}{6} = 0.65$
- 2 / 4 $Entropy = -\frac{2}{6} \cdot \log_2 \frac{2}{6} - \frac{4}{6} \cdot \log_2 \frac{4}{6} = 0.92$

# Вычисление прироста информации после разбиения узла по атрибуту

- $Gain(A) = Entropy(node) - Info(A)$

  $$Entropy(node) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$$

  $$Info(A) = \sum_{i=1}^{k} \frac{n_i}{n} Entropy(child_i)$$

  $A$ – атрибут разбиения с $k$ различных значений

  $n$ – количество объектов в выборке разбиваемого узла-родителя $node$

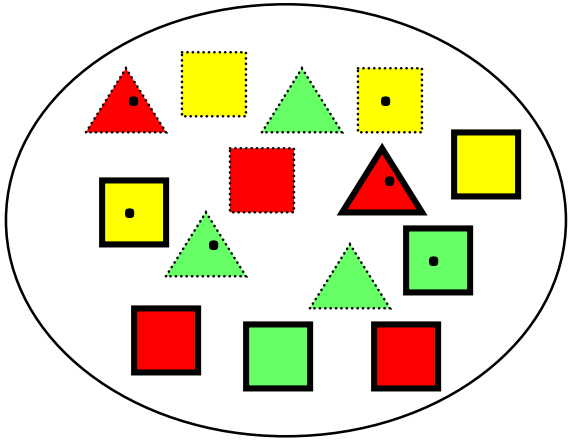
  $n_i$ – количество объектов в выборке узла-потомка $child_i$

- Для разбиения выбирается атрибут

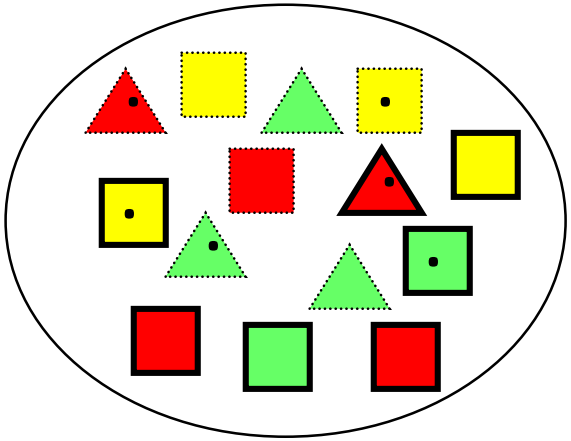
  $$A_{split} = \arg\max_i Gain(A_i) = \arg\min_i Info(A_i)$$

# Пример построения дерева решений по ID3

| # | Attribute | | | Shape |
|---|---|---|---|---|
| | Color | Outline | Dot | |
| 1 | green | dashed | no | triange |
| 2 | green | dashed | yes | triange |
| 3 | yellow | dashed | no | square |
| 4 | red | dashed | no | square |
| 5 | red | solid | no | square |
| 6 | red | solid | yes | triange |
| 7 | green | solid | no | square |
| 8 | green | dashed | no | triange |
| 9 | yellow | solid | yes | square |
| 10 | red | solid | no | square |
| 11 | green | solid | yes | square |
| 12 | yellow | dashed | yes | square |
| 13 | yellow | solid | no | square |
| 14 | red | dashed | yes | triange |



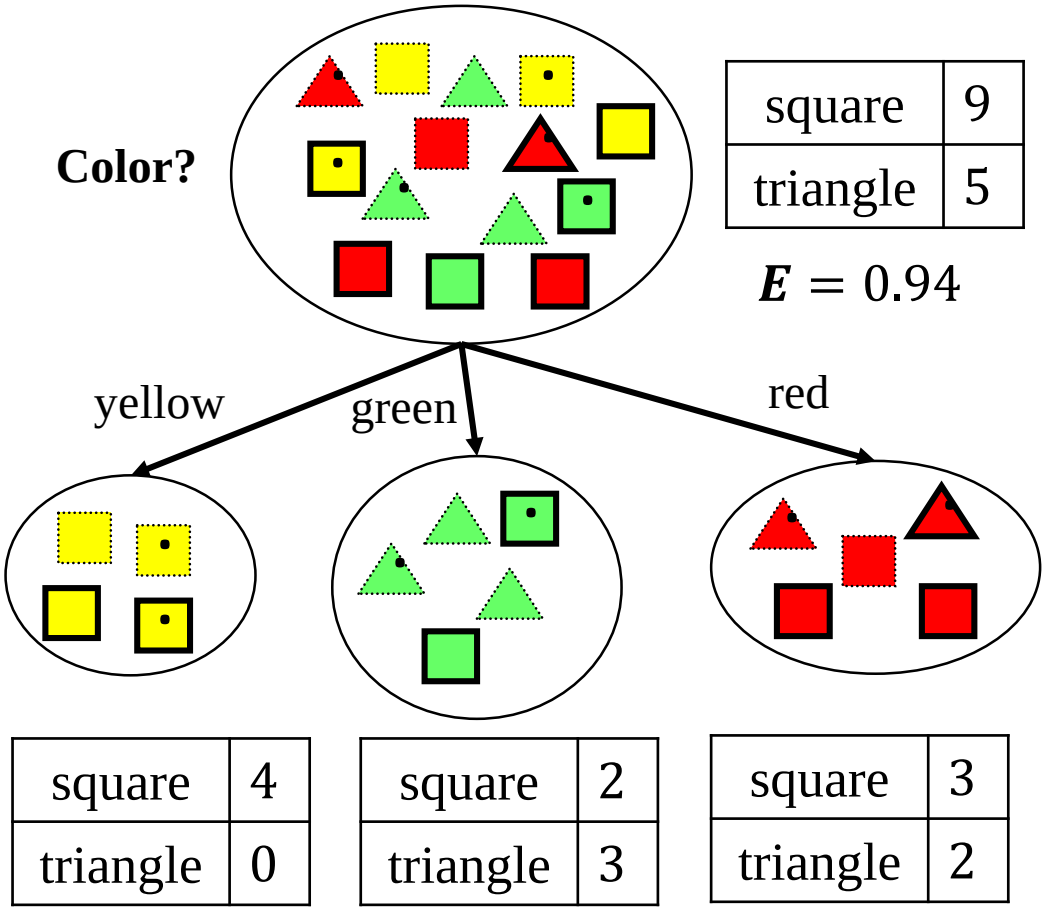
# Пример: вычисление энтропии узла-родителя



| | |
|---|---|
| square | 9 |
| triangle | 5 |

- $\boldsymbol{Entropy} = -\frac{9}{14} \cdot \log_2 \frac{9}{14} - \frac{5}{14} \cdot \log_2 \frac{5}{14} = \mathbf{0.94}$

# Пример: вычисление энтропии узлов-потомков




| | |
|---|---|
| square | 9 |
| triangle | 5 |

$E = 0.94$

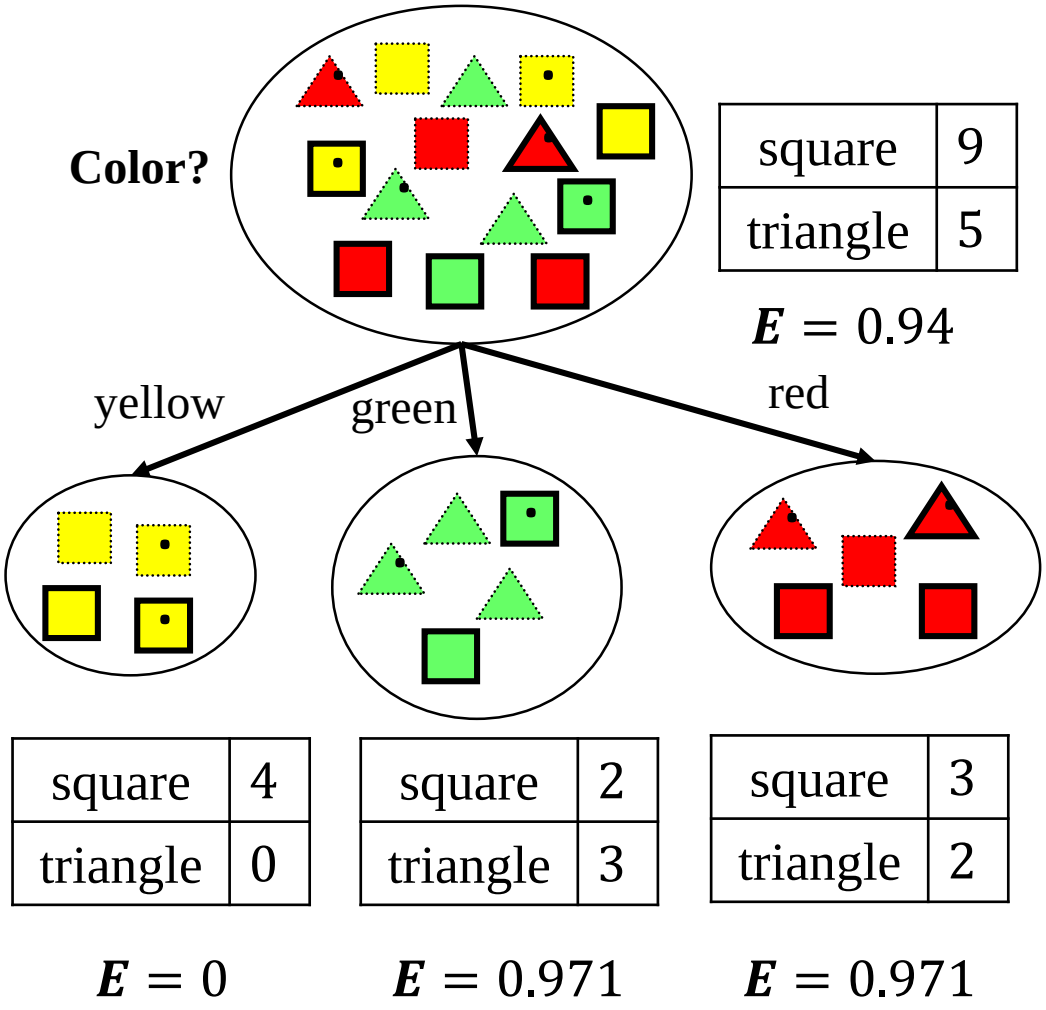
| | |
|---|---|
| square | 4 |
| triangle | 0 |

| | |
|---|---|
| square | 2 |
| triangle | 3 |

| | |
|---|---|
| square | 3 |
| triangle | 2 |

- $Entropy(yellow) = 0$
- $Entropy(green) = -\frac{2}{5} \cdot \log_2 \frac{2}{5} - \frac{3}{5} \cdot \log_2 \frac{3}{5} = \mathbf{0.971}$
- $Entropy(red) = -\frac{3}{5} \cdot \log_2 \frac{3}{5} - \frac{2}{5} \cdot \log_2 \frac{2}{5} = \mathbf{0.971}$

# Пример: вычисление прироста информации




| | |
|---|---|
| square | 9 |
| triangle | 5 |

$E = 0.94$

| | |
|---|---|
| square | 4 |
| triangle | 0 |

$E = 0$

| | |
|---|---|
| square | 2 |
| triangle | 3 |

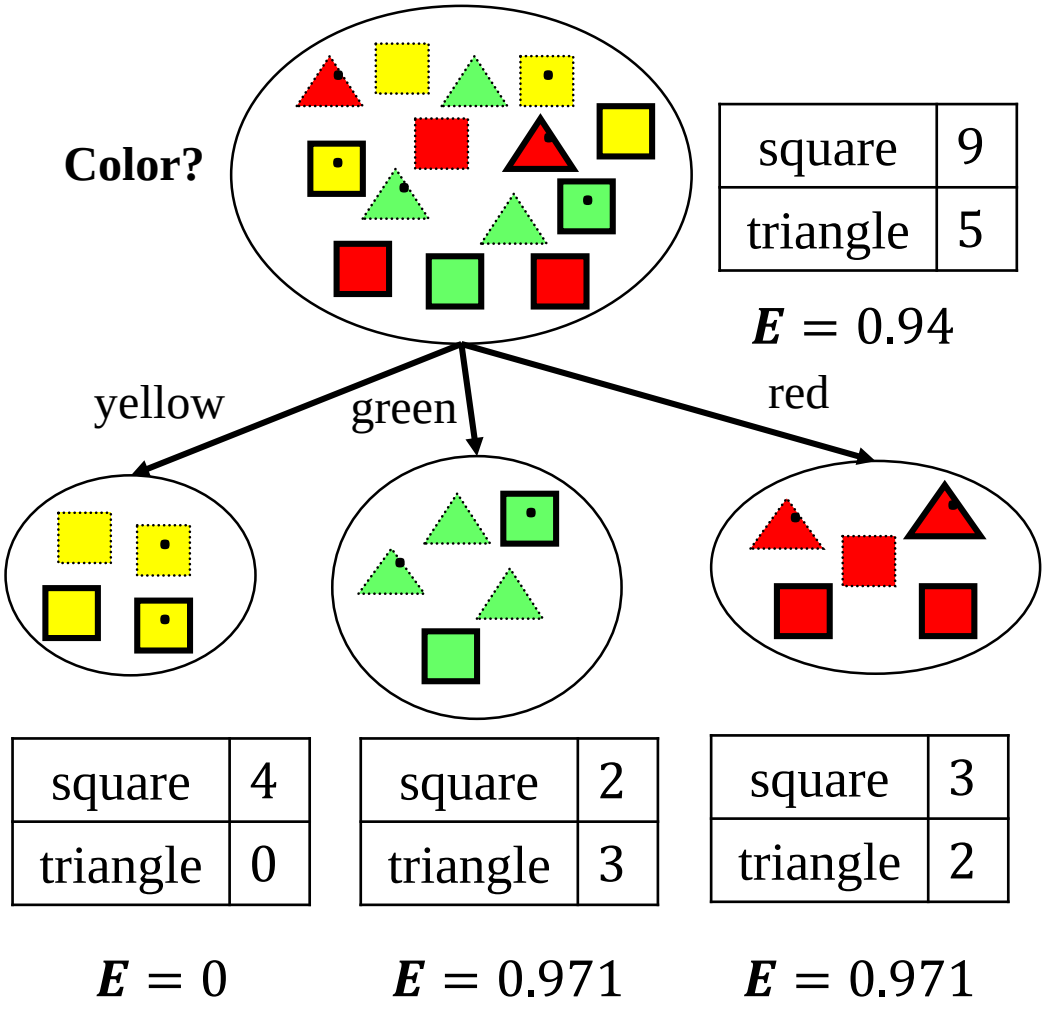
$E = 0.971$

| | |
|---|---|
| square | 3 |
| triangle | 2 |

$E = 0.971$

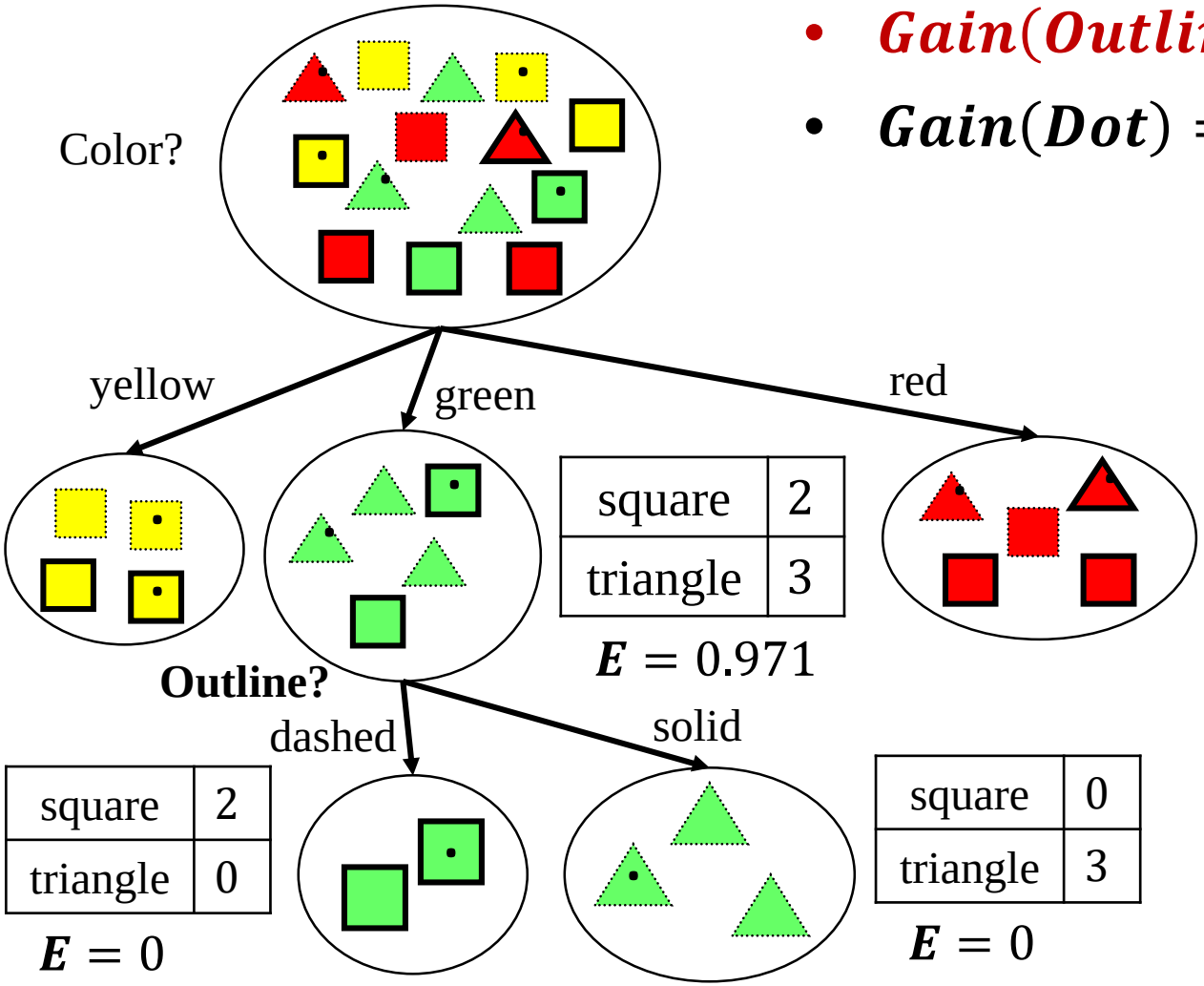
- $\boldsymbol{Info(Color)} = \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.971 + \frac{5}{14} \cdot 0.971 = \mathbf{0.694}$
- $\boldsymbol{Gain(Color)} = 0.94 - 0.694 = \mathbf{0.246}$

# Пример: выбор атрибута разбиения




- $Gain(Color) = 0.246$
- $Gain(Outline) = 0.151$
- $Gain(Dot) = 0.048$

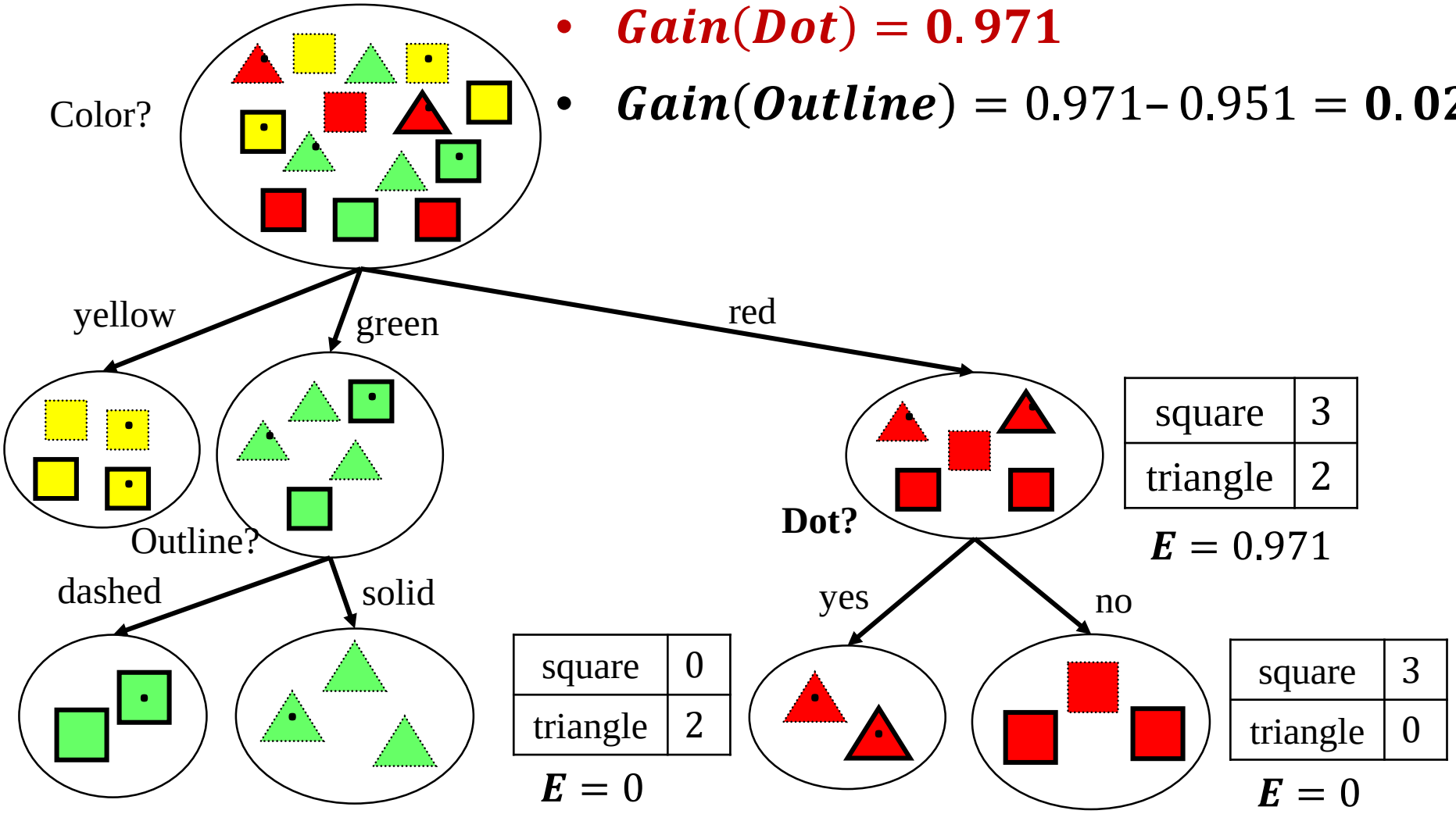
# Пример: выбор атрибута разбиения



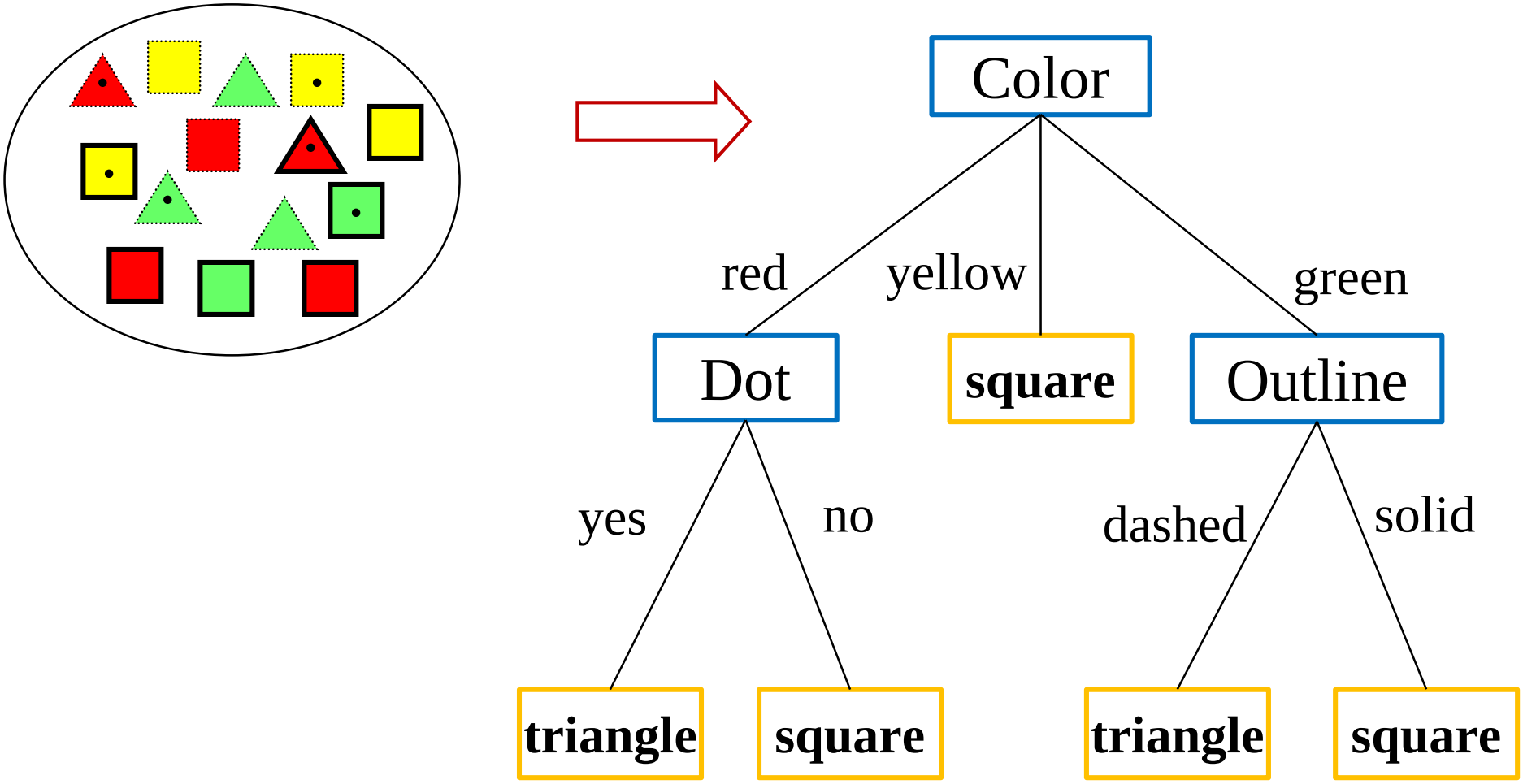

- $\boldsymbol{Gain(Outline) = 0.971}$
- $\boldsymbol{Gain(Dot)} = 0.971 - 0.951 = \mathbf{0.02}$

# Пример: выбор атрибута разбиения

- $Gain(Dot) = 0.971$
- $Gain(Outline) = 0.971 - 0.951 = 0.02$



# Пример: итоговое дерево решений



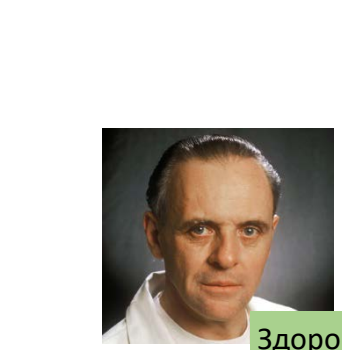
# Алгоритм построения дерева решений

```
TreeGrowth (E, F)
 1: if stopping_cond(E,F) = true then
 2:    leaf = createNode().
 3:    leaf.label = Classify(E).
 4:    return leaf.
 5: else
 6:    root = createNode().
 7:    root.test_cond = find_best_split(E, F).
 8:    let V = {v|v is a possible outcome of root.test_cond }.
 9:    for each v ∈ V do
10:       E_v = {e | root.test_cond(e) = v and e ∈ E}.
11:       child = TreeGrowth(E_v, F).
12:       add child as descendent of root and label the edge (root → child) as v.
13:    end for
14: end if
15: return root.
```

# Содержание

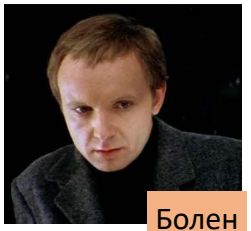
- Общий подход к классификации
- Деревья решений
- **Ансамблевая классификация**
- Оценка качества классификации

# Ансамблевая классификация (идея)


Здоров


Болен


Болен

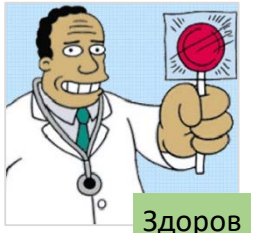

Болен


Здоров


Здоров

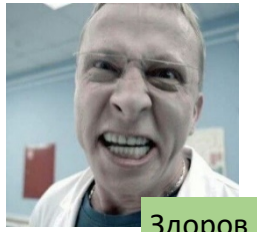

Здоров*


Болен


Здоров


Болен

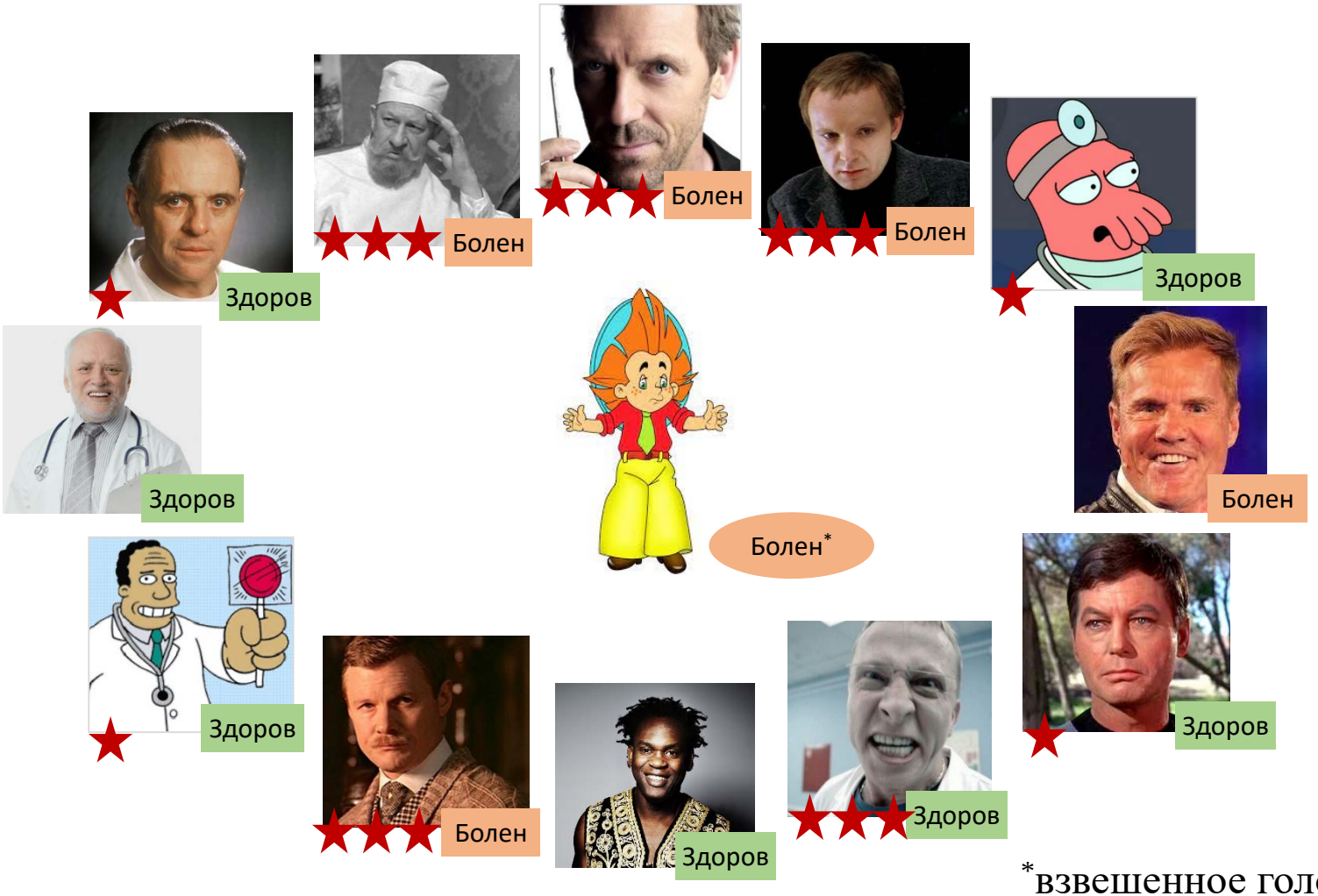

Здоров


Здоров


Здоров

*мажоритарное голосование

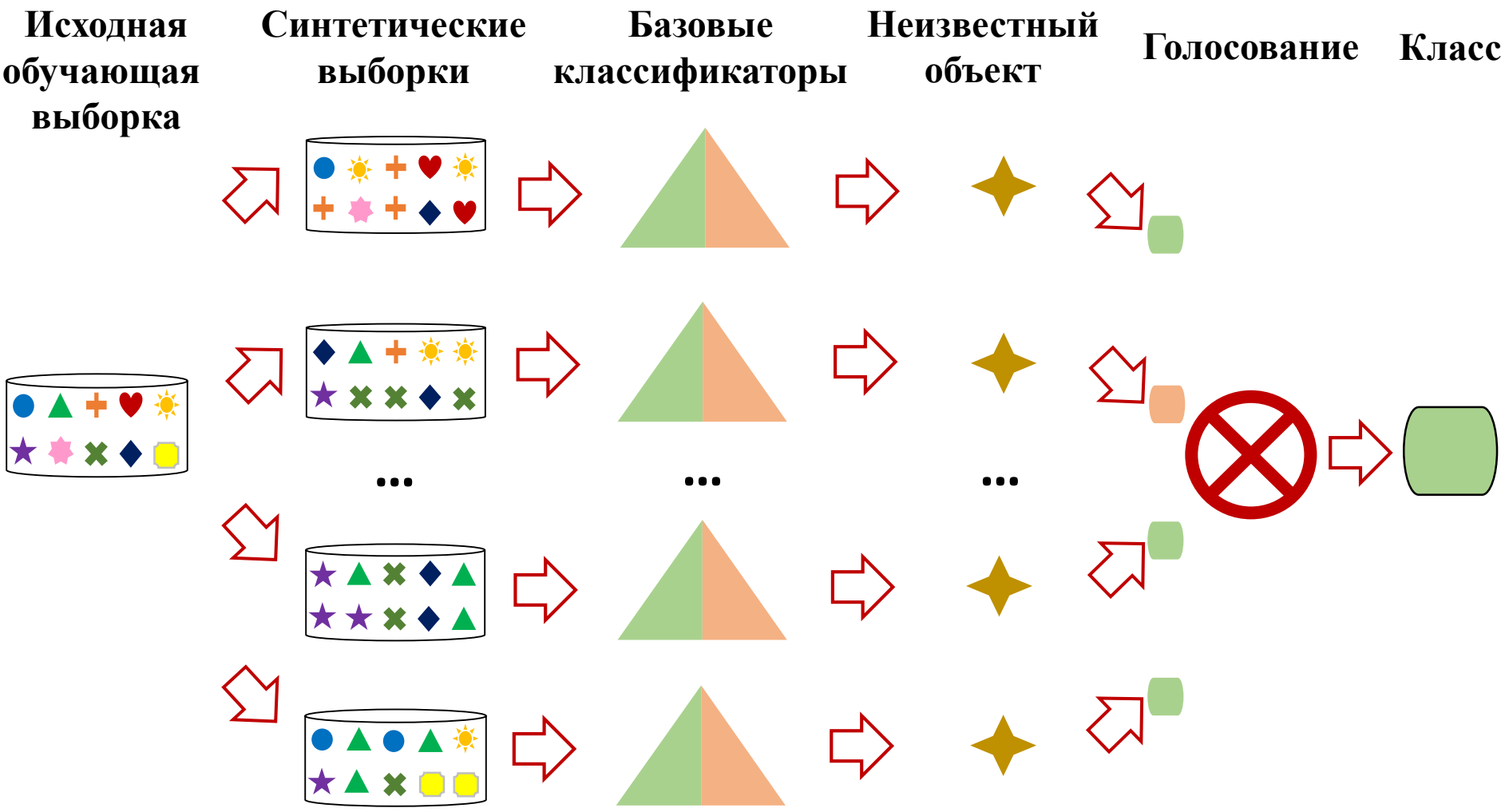
# Ансамблевая классификация (идея)




*взвешенное голосование

# Ансамблевая классификация

$$|D| = n = |D_i|,$$
$$\mathrm{P}(\mathrm{o} \in D_i) = 1 - (1 - 1/n)^n,$$
$$\lim_{n \to +\infty} P(o \in D_i) = 1 - 1/e \approx 0.632$$



# Почему работают ансамбли?

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

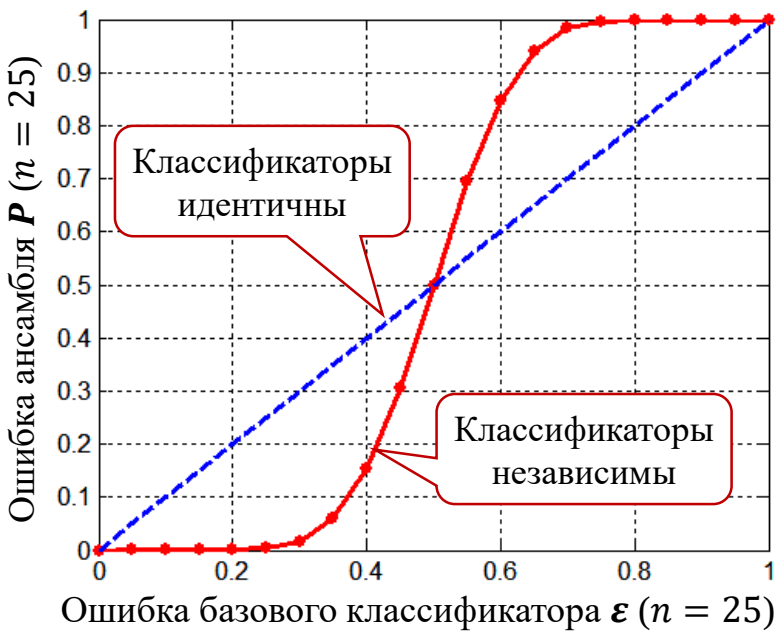
$$P(|wrong| > \lfloor n/2 \rfloor) = \sum_{i=\lfloor n/2 \rfloor+1}^{n} C_i^n \cdot \varepsilon^i \cdot (1-\varepsilon)^{n-i}$$

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

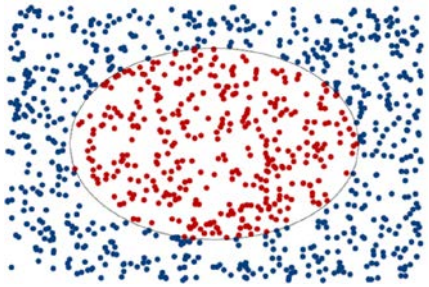
- **Неравенство Хёфдинга**

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

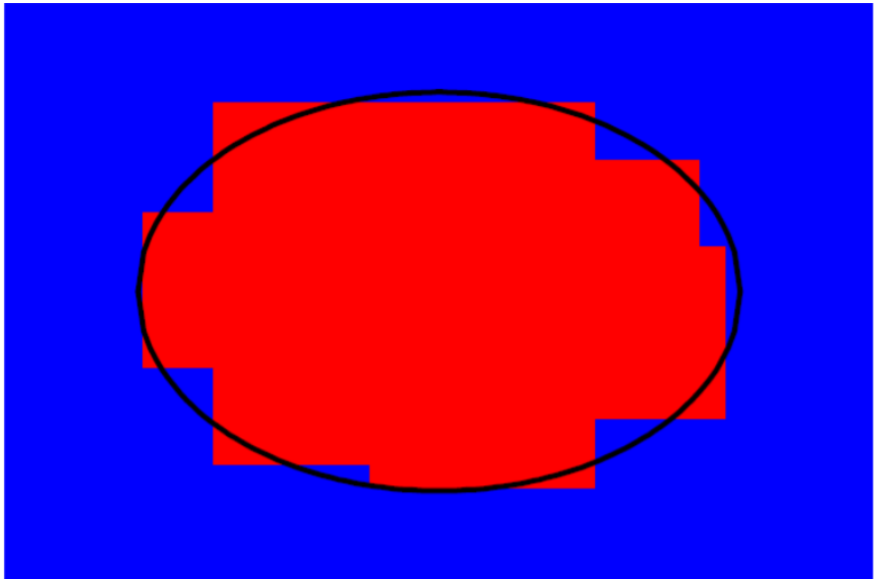
вероятность ошибки ансамбля убывает экспоненциально с ростом числа базовых классификаторов



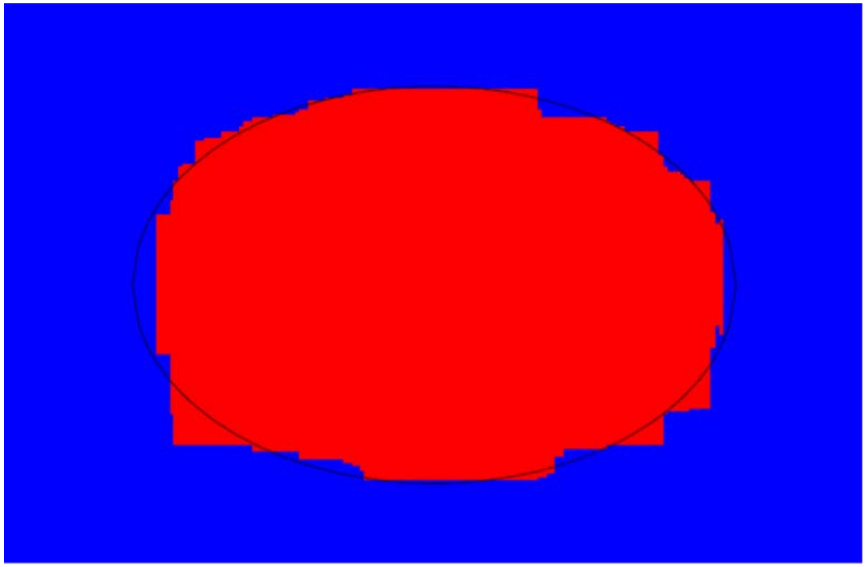
# Пример работы ансамбля классификаторов



Исходное множество



**Одно дерево решений**



**Ансамбль деревьев решений**

# Бэггинг (Bagging, Bootstrap Aggregating)

- Базовые идеи
  - Сэмплинг с замещением при формировании подвыборок
  - Голосование по большинству 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, чему у одного классификатора
  - Устойчивость к шумам в данных при несущественном снижении точности

# Алгоритм бэггинга

1: Let $k$ be the number of bootstrap samples.
2: **for** $i = 1$ to $k$ **do**
3: Create a bootstrap sample of size $N$, $D_i$.
4: Train a base classifier $C_i$ on the bootstrap sample $D_i$.
5: **end for**
6: $C^*(x) = \underset{y}{\operatorname{argmax}} \sum_i \delta\big(C_i(x) = y\big)$.

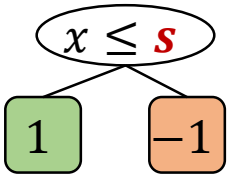
$\{\delta(\cdot) = 1$ if its argument is true and 0 otherwise$\}$.

# 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

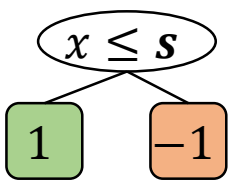
| Уч. | $s$ | $Left$ | $Right$ |
|---|---|---|---|
| 1 | 0.75 | −1 | 1 |
| 2 | ? | ? | ? |
| 3 | ? | ? | ? |
| 4 | ? | ? | ? |
| 5 | ? | ? | ? |
| 6 | ? | ? | ? |
| 7 | ? | ? | ? |
| 8 | ? | ? | ? |
| 9 | ? | ? | ? |
| 10 | ? | ? | ? |

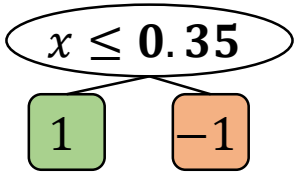
# Пример бэггинга: выборка и обучение 1-го участника ансамбля

| $X$ | $Class$ |
|---|---|
| 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**

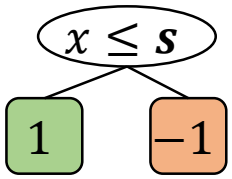
| $X$ | $Class$ |
|---|---|
| 0.1 | 1 |
| 0.2 | 1 |
| 0.2 | 1 |
| 0.3 | 1 |
| 0.4 | -1 |
| 0.4 | -1 |
| 0.5 | -1 |
| 0.6 | -1 |
| 0.9 | 1 |
| 0.9 | 1 |



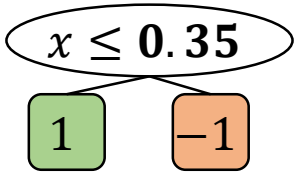
# Пример бэггинга: выборки и обучение 1-5 участников ансамбля

| $X$ | $Class$ |
|---|---|
| 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 |



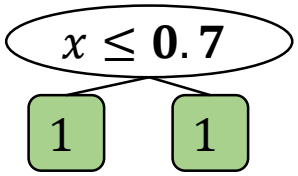

**1**

| $X$ | $Class$ |
|---|---|
| 0.1 | 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 |



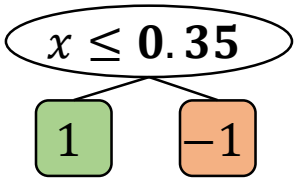

**2**

| $X$ | $Class$ |
|---|---|
| 0.1 | 1 |
| 0.2 | 1 |
| 0.3 | 1 |
| 0.4 | -1 |
| 0.5 | -1 |
| 0.5 | -1 |
| 0.9 | 1 |
| 1.0 | 1 |
| 1.0 | 1 |
| 1.0 | 1 |



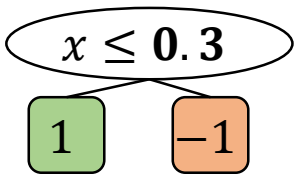

**3**

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



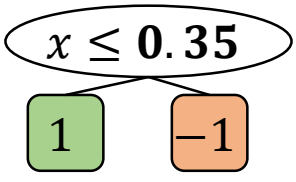

**4**

| $X$ | $Class$ |
|---|---|
| 0.1 | 1 |
| 0.1 | 1 |
| 0.2 | 1 |
| 0.4 | -1 |
| 0.4 | -1 |
| 0.5 | -1 |
| 0.5 | -1 |
| 0.7 | -1 |
| 0.8 | 1 |
| 0.9 | 1 |



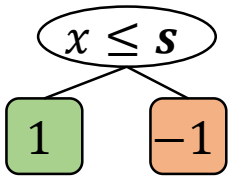

**5**

| $X$ | $Class$ |
|---|---|
| 0.1 | 1 |
| 0.1 | 1 |
| 0.2 | 1 |
| 0.5 | -1 |
| 0.6 | -1 |
| 0.6 | -1 |
| 0.6 | -1 |
| 1.0 | 1 |
| 1.0 | 1 |
| 1.0 | 1 |



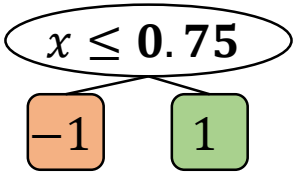
# Пример бэггинга: выборки и обучение 6-10 участников ансамбля

| 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 |



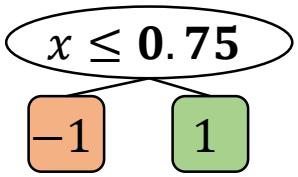

**6**

| X | Class |
|---|---|
| 0.2 | 1 |
| 0.4 | -1 |
| 0.5 | -1 |
| 0.6 | -1 |
| 0.7 | -1 |
| 0.7 | -1 |
| 0.8 | 1 |
| 0.9 | 1 |
| 1.0 | 1 |



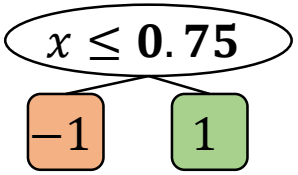

**7**

| X | Class |
|---|---|
| 0.1 | 1 |
| 0.4 | -1 |
| 0.4 | -1 |
| 0.6 | -1 |
| 0.7 | -1 |
| 0.8 | 1 |
| 0.9 | 1 |
| 0.9 | 1 |
| 1.0 | 1 |



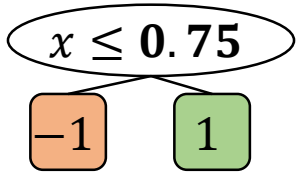

**8**

| X | Class |
|---|---|
| 0.1 | 1 |
| 0.2 | 1 |
| 0.5 | -1 |
| 0.5 | -1 |
| 0.5 | -1 |
| 0.7 | -1 |
| 0.7 | -1 |
| 0.8 | 1 |
| 0.9 | 1 |
| 1.0 | 1 |



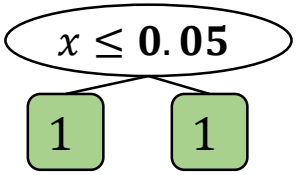

**9**

| X | Class |
|---|---|
| 0.1 | 1 |
| 0.3 | 1 |
| 0.4 | -1 |
| 0.4 | -1 |
| 0.6 | -1 |
| 0.7 | -1 |
| 0.7 | -1 |
| 0.8 | 1 |
| 1.0 | 1 |
| 1.0 | 1 |




**10**

| X | Class |
|---|---|
| 0.1 | 1 |
| 0.1 | 1 |
| 0.1 | 1 |
| 0.1 | 1 |
| 0.3 | 1 |
| 0.3 | 1 |
| 0.8 | 1 |
| 0.8 | 1 |
| 0.9 | 1 |
| 0.9 | 1 |



# 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 |
| ***Vote*** | **2** | **2** | **2** | **−6** | **−6** | **−6** | **−6** | **2** | **2** | **2** |
| ***Class*** | **1** | **1** | **1** | **−1** | **−1** | **−1** | **−1** | **1** | **1** | **1** |

# 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

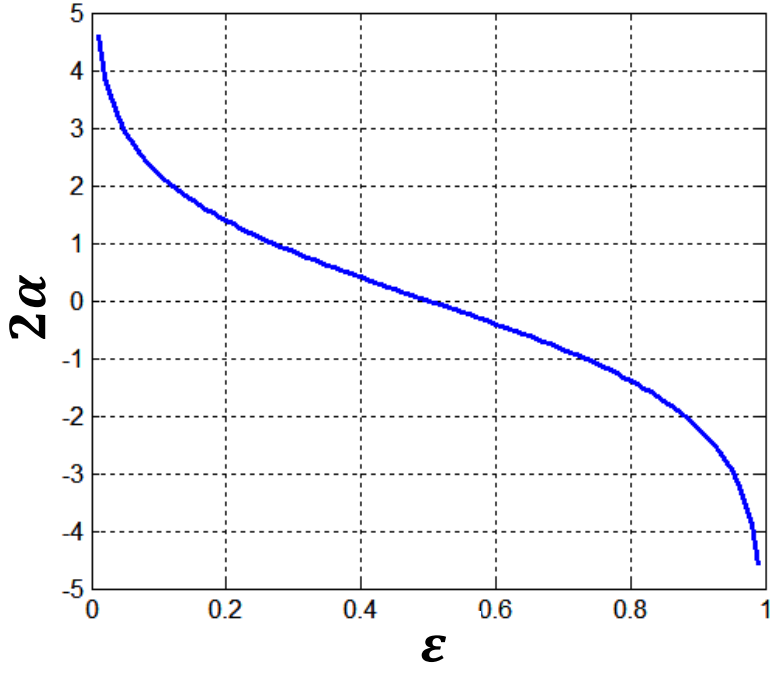
# Алгоритм AdaBoost

- **Обучающая выборка:**
  $(x_1, y_1), \ldots, (x_n, y_n)$, веса $w_1, \ldots, w_n$
- **Ансамбль:** $C_1, \ldots, C_k$
- **Ошибка участника ансамбля:**
  $\varepsilon_i = \frac{1}{n}\sum_{j=1}^{n} w_j \cdot \delta(C_i(x_j) \neq y_j)$
- **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: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$. {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: $\epsilon_i = \frac{1}{N}\left[\sum_j w_j \, \delta\big(C_i(x_j) \neq y_j\big)\right]$ {Calculate the weighted error.}
8: **if** $\epsilon_i > 0.5$ **then**
9: $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \ldots, N\}$.
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: $C^*(\mathbf{x}) = \underset{y}{\operatorname{argmax}} \sum_{j=1}^{T} \alpha_j \delta(C_j(\mathbf{x}) = y)\big)$.

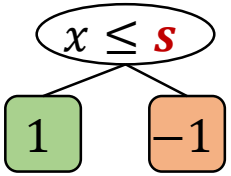
Если ошибка классификации более 50%, то повторить сэмплинг

# 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**

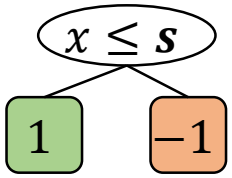
| $X$ | $Class$ | $weight$ |
|---|---|---|
| 0.1 | 1 | 0.1 |
| 0.4 | -1 | 0.1 |
| 0.5 | -1 | 0.1 |
| 0.6 | -1 | 0.1 |
| 0.6 | -1 | 0.1 |
| 0.6 | -1 | 0.1 |
| 0.7 | -1 | 0.1 |
| 0.7 | -1 | 0.1 |
| 0.8 | 1 | 0.1 |
| 1.0 | 1 | 0.1 |

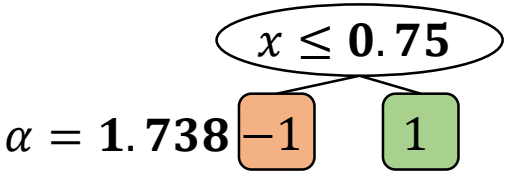
**2**

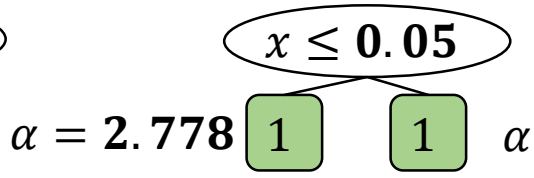
| $X$ | $Class$ | $weight$ |
|---|---|---|
| 0.1 | 1 | 0.311 |
| 0.1 | 1 | 0.311 |
| 0.2 | 1 | 0.311 |
| 0.2 | 1 | 0.01 |
| 0.2 | 1 | 0.01 |
| 0.2 | 1 | 0.01 |
| 0.3 | 1 | 0.01 |
| 0.3 | 1 | 0.01 |
| 0.3 | 1 | 0.01 |
| 0.3 | 1 | 0.01 |

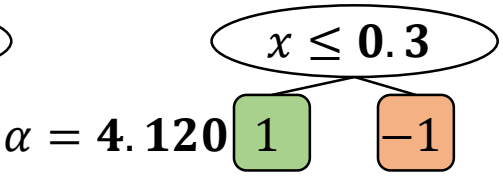
**3**

| $X$ | $Class$ | $weight$ |
|---|---|---|
| 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 |



# 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 |
| $SUM$ | 5.16 | 5.16 | 5.16 | −3.08 | −3.08 | −3.08 | −3.08 | 0.397 | 0.397 | 0.397 |
| $Class$ | 1 | 1 | 1 | −1 | −1 | −1 | −1 | 1 | 1 | 1 |

# Содержание

- Общий подход к классификации
- Деревья решений
- Ансамблевая классификация
- **Оценка качества классификации**

# Матрица ошибок (Confusion matrix)

| Прогноз \ Класс | $C$ | $\neg C$ |
|---|---|---|
| $C$ | $True\ Positives\ (TP)$   | $False\ Positives\ (FP)$   |
| $\neg C$ | $False\ Negatives\ (FN)$   | $True\ Negatives\ (TN)$   |

# Матрица ошибок (Confusion matrix)

| **Прогноз \ Класс** | $\boldsymbol{C}$ | $\boldsymbol{\neg C}$ |
|---|---|---|
| $\boldsymbol{C}$ | $True\ Positives\ (TP)$ | $False\ Positives\ (FP)$ |
| $\boldsymbol{\neg C}$ | $False\ Negatives\ (FN)$ | $True\ Negatives\ (TN)$ |

- $TP$ – верно распознанные объекты класса $C$
- $TN$ – верно распознанные объекты класса $\neg C$
- $FN$ – объекты класса $C$, неверно распознанные как объекты класса $\neg C$
- $FP$ – объекты класса $\neg C$, неверно распознанные как объекты класса $C$
- $P = TP + FN$ – объекты класса $C$
- $N = FP + TN$ – объекты класса $\neg C$

# Accuracy, recognition rate (Аккуратность)
# Error/misclassification rate (Доля ошибок)

- $\boldsymbol{Accuracy = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+FN+FP+TN}}$
- $\boldsymbol{Error = 1 - Accuracy = \frac{FP+FN}{P+N} = \frac{FP+FN}{TP+FN+FP+TN}}$
- Наиболее простой способ оценки качества: доля верных/неверных ответов
- Неадекватен при дисбалансе классов

| **П\К** | $\boldsymbol{C}$ | $\boldsymbol{\neg C}$ |
|---|---|---|
| $\boldsymbol{C}$ | $TP$ | $FP$ |
| $\boldsymbol{\neg C}$ | $FN$ | $TN$ |

# Неадекватность Accuracy и Error при дисбалансе классов в обучающей выборке

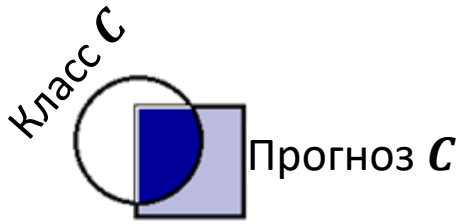
| П\К | Рак = да | Рак = нет |
|---|---|---|
| Рак = да | $TP = 0$ | $FP = 10$ |
| Рак = нет | $FN = 0$ | $TN = 990$ |

- $Accuracy = \frac{0+990}{0+0+10+990} = 99\%, Error = 1\%$
- Классификатор идеально распознает отсутствие рака и плохо распознает рак
- При дисбалансе классов в обучающей выборке нужны отдельные меры качества распознавания объектов из классов $C$ и $\neg C$

# Precision (точность), Recall (полнота)

- *Точность* показывает долю объектов, распознанных как объекты класса $C$, действительно являющихся объектами класса $C$

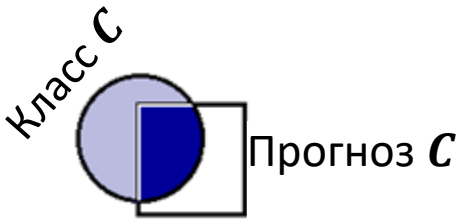
$$\boldsymbol{Precision = \frac{TP}{TP+FP}}$$




| П\К | $C$ | $\neg C$ |
|---|---|---|
| $C$ | $TP$ | $FP$ |
| $\neg C$ | $FN$ | $TN$ |

- *Полнота* показывает долю объектов класса $C$, которые действительно распознаны как объекты класса $C$

$$\boldsymbol{Recall = \frac{TP}{TP+FN}}$$




| П\К | $C$ | $\neg C$ |
|---|---|---|
| $C$ | $TP$ | $FP$ |
| $\neg C$ | $FN$ | $TN$ |

- Точность и полнота имеют обратную зависимость

# Пример

| Прогноз\Класс | Рак = Да | Рак = Нет |
|---|---|---|
| Рак = Да | **90** $TP$ | **140** $FP$ |
| Рак = Нет | **210** $FN$ | **9560** $TN$ |
| Всего | 300 $P$ | 9700 $N$ |

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

- $Accuracy = \frac{90+9560}{300+9700} = 97.7\%, Error = 2.3\%$
- $Precision = \frac{90}{90+140} = 39.13\%$
- $Recall = \frac{90}{90+210} = 30\%$

# Меры $F$ и $F_\beta$

- Мера $F$ ($F_1$ или $F$-score) – гармоническое среднее точности (*precision*) и полноты (*recall*)

$$\boldsymbol{F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}}$$

- $F_\beta$ – взвешенная мера точности (*precision*) и полноты (*recall*)

$$\boldsymbol{F_\beta = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}}$$

  - $\beta > 0$ показывает важность точности (*precision*) по отношению к полноте (*recall*)
  - типичные значения $\beta = 2$ и $\beta = 0.5$

# Пример

| Прогноз\Класс | Рак = Да | Рак = Нет |
|---|---|---|
| Рак = Да | **90** $TP$ | **140** $FP$ |
| Рак = Нет | **210** $FN$ | **9560** $TN$ |
| Всего | 300 $P$ | 9700 $N$ |

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- $Accuracy = \frac{90+9560}{300+9700} = 97.7\%, Error = 2.3\%$
- $Precision = \frac{90}{90+140} = 39.13\%, Recall = \frac{90}{90+210} = 30\%$
- $F = \frac{2 \cdot 0.3913 \cdot 0.3}{0.3913+0.3} = 0.34$

# Меры качества классификации

- $Accuracy = \frac{TP+TN}{P+N}$, $Error = 1 - Accuracy$
- $Precision = \frac{TP}{TP+FP}$
- $Sensitivity = Recall = TPrate = \frac{TP}{TP+FN}$
- $Specificity = TNrate = \frac{TN}{TN+FP}$
- $FPrate = 1 - Specificity$
- $FNrate = 1 - Sensitivity$
- $F = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall}$, $F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision+Recall}$

| **П\К** | $C$ | $\neg C$ |
|---|---|---|
| $C$ | $TP$ | $FP$ |
| $\neg C$ | $FN$ | $TN$ |

# Основные методы подготовки тестовой выборки

- Откладывание (*hold-out*)
  - Множество размеченных данных единожды разбивается на непересекающиеся подмножества: обучающая выборка и тестовая выборка. Типичные соотношения – 70:30, 80:20
- Случайный отбор (random sampling)
  - Повторить откладывание $k$ раз, итоговая точность – среднее
- Перекрестная проверка из $k$ итераций (*$k$-fold cross-validation*), обычно $k = 10$
  - Обучающая выборка разбивается на $k$ непересекающихся частей
  - Выполняется $k$ итераций: обучение проводится на $k - 1$ частях, тестирование проводится на части, не участвовавшей в обучении
  - Итоговая точность – отношение общих кол-в верных ответов и попыток
  - Вариация для маломощных множеств: $k$ равно мощности выборки
- Самонастройка (0.632 *bootstrapping*)
  - По выборке из $d$ объектов $d$ раз строятся случайные выборки с повторением из $d$ объектов – обучающие выборки; не отобранные объекты формируют тестовые выборки
  - $Accuracy(M) = \frac{1}{d}\sum_{i=1}^{d}(0.632 \cdot Accuracy(M_i)_{test_{set}} + 0.368 \cdot Accuracy(M_i)_{train_{set}})$

# Литература


- Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd Edition. Morgan Kaufmann, 2011. 740 p. ISBN: 978-0123814791
  - 8. Classification: Basic Concepts. 8.1. Basic Concepts, 8.2. Decision Tree Induction, 8.5 Model Evaluation and Selection, 8.6 Techniques to Improve Classification Accuracy
- 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
  - 3. Decision Tree Induction. 3.1 Basic Concepts, 3.2 General Framework for Classification, Decision Tree Classifier, 3.5 Model Selection, 3.6 Model Evaluation, 4.10 Ensemble Methods