Лекция 2
Логические классификаторы

Машинное обучение

15/02/22

Кабалянц Петр Степанович

Задача обучения классификатора

X — множество *объектов*;
Y — множество *ответов*;
$y\colon X \to Y$ — неизвестная зависимость (target function).

Дано:
$\{x_1, \dots, x_\ell\} \subset X$ — *обучающая выборка* (training sample);
$y_i = y(x_i)$, $i = 1, \dots, \ell$ — известные ответы.

Найти:
$a\colon X \to Y$ — алгоритм, решающую функцию (decision function), приближающую y на всём множестве X.

Весь курс машинного обучения — это конкретизация:

- как задаются объекты и какими могут быть ответы;
- в каком смысле «a приближает y»;
- как строить функцию a.

Векторное описание объектов

$f_j \colon X \to D_j, \;\; j = 1, \dots, n$ — признаки объектов (features).

Типы признаков:

- $D_j = \{0, 1\}$ — *бинарный* признак f_j;
- $|D_j| < \infty$ — *номинальный* признак f_j;
- $|D_j| < \infty$, D_j упорядочено — *порядковый* признак f_j;
- $D_j = \mathbb{R}$ — *количественный* признак f_j.

Вектор $(f_1(x), \dots, f_n(x))$ — *признаковое описание* объекта x.

Матрица «*объекты–признаки*» (feature data)

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

Векторное описание объектов

$f_j\colon X \to D_j,\ \ j = 1, \dots, n$ — признаки объектов (features).

Типы признаков:

- $D_j = \{0, 1\}$ — *бинарный* признак f_j;
- $|D_j| < \infty$ — *номинальный* признак f_j;
- $|D_j| < \infty$, D_j упорядочено — *порядковый* признак f_j;
- $D_j = \mathbb{R}$ — *количественный* признак f_j.

Вектор $(f_1(x), \dots, f_n(x))$ — *признаковое описание* объекта x.

Матрица «*объекты–признаки*» (feature data)

$$F = \big\| f_j(x_i) \big\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

Векторное описание объектов

$f_j \colon X \to D_j$, $j = 1, \dots, n$ — признаки объектов (features).

Типы признаков:

- $D_j = \{0, 1\}$ — *бинарный* признак f_j;
- $|D_j| < \infty$ — *номинальный* признак f_j;
- $|D_j| < \infty$, D_j упорядочено — *порядковый* признак f_j;
- $D_j = \mathbb{R}$ — *количественный* признак f_j.

Вектор $(f_1(x), \dots, f_n(x))$ — *признаковое описание* объекта x.

Матрица «*объекты–признаки*» (feature data)

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

Функционалы качества классификации и функции потерь

Индикатор ошибки на одном объекте:

$$\mathscr{L}(a,x) = [a(x) \neq y(x)]$$

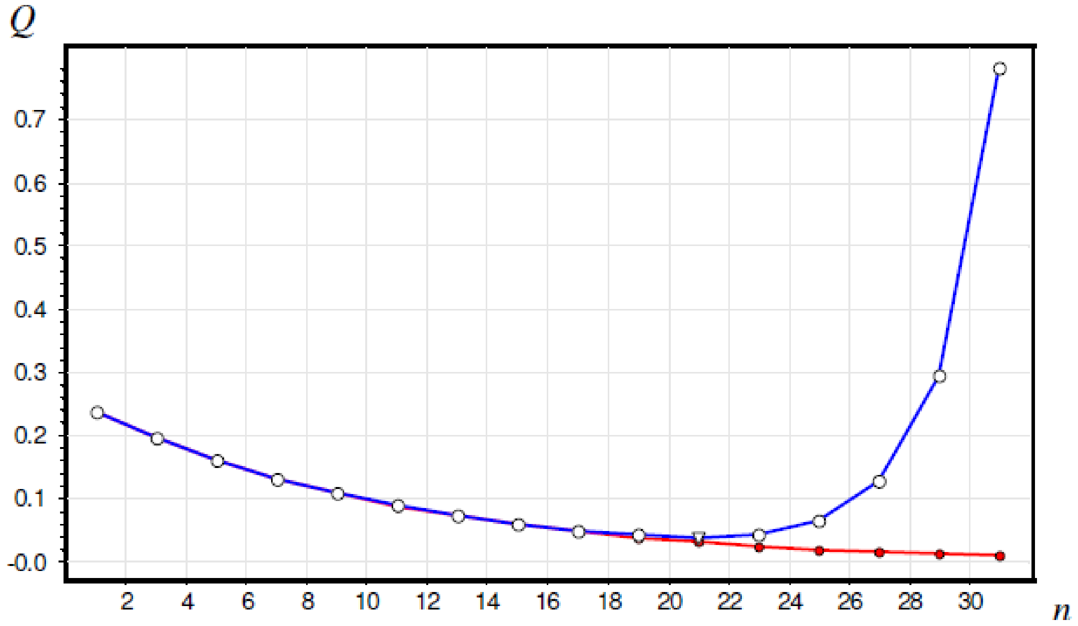
Среднее ошибок алгоритма на всей выборке:

$$Q(a,X^\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(a,x_i)$$

Обучение как минимизация ошибок:

$$\mu(X^\ell) = \arg\min_{a\in A} Q(a,X^\ell)$$

Переобучение: качество классификации на обучающей выборке существенно больше чем качество классификации на контрольных (тестовых) данных



Скользящий контроль (leave-one-out) и кросс-проверка (cross-validation)

$$\mathrm{LOO}(\mu, X^L) = \frac{1}{L} \sum_{i=1}^{L} \mathscr{L}(\mu(X^L \setminus \{x_i\}), x_i) \to \min$$

$$\mathrm{CV}(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q(\mu(X_n^\ell), X_n^k) \to \min$$

Линейные алгоритмы классификации в случае двух классов

$$a(x,w) = \operatorname{sign}\langle x, w\rangle = \operatorname{sign}\sum_{j=1}^{n} w_j f_j(x)$$

Разделяющий классификатор: $a(x,w) = \operatorname{sign} g(x,w)$

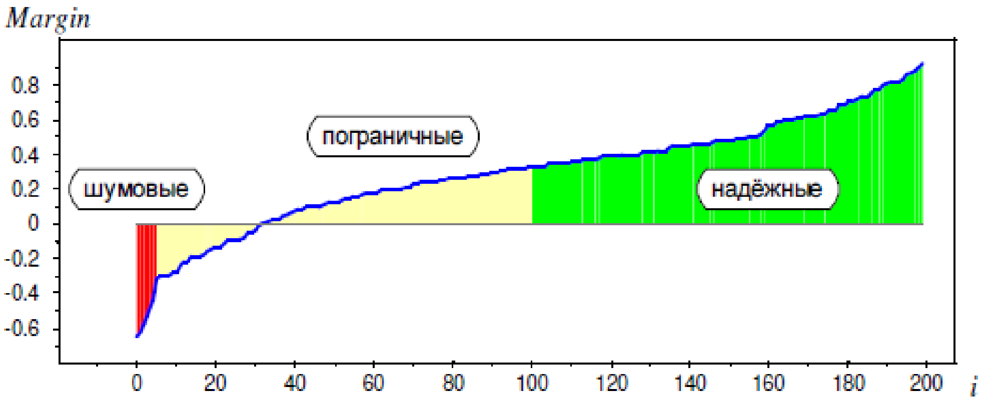
$g(x,w)$ — *разделяющая (дискриминантная) функция*

$g(x,w) = 0$ — уравнение разделяющей поверхности

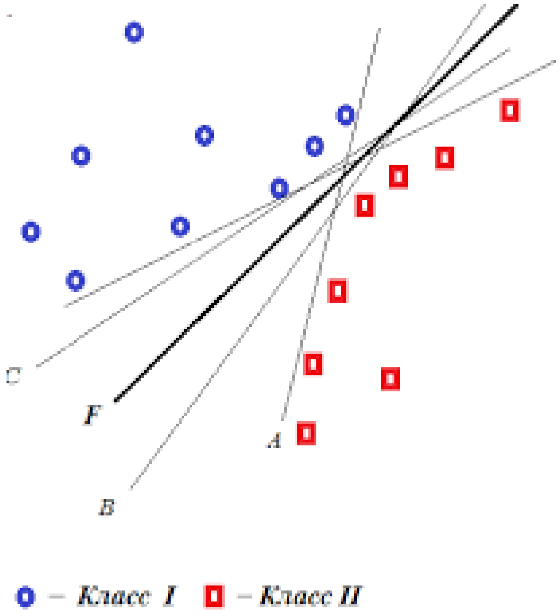
Ранжирование объектов по величине отступа от линии классификации

$M_i(w) = g(x_i, w)y_i$ — *отступ* (margin) объекта x_i

$M_i(w) < 0 \iff$ алгоритм $a(x, w)$ ошибается на x_i

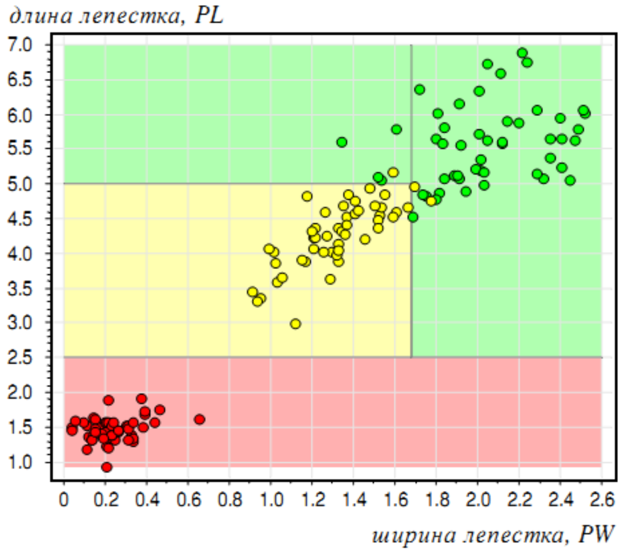


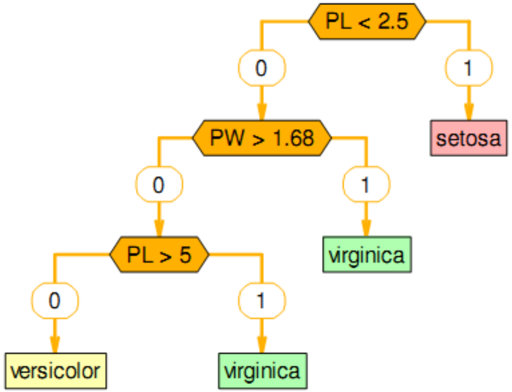
Линейно разделимые классы: какую выбрать прямую?

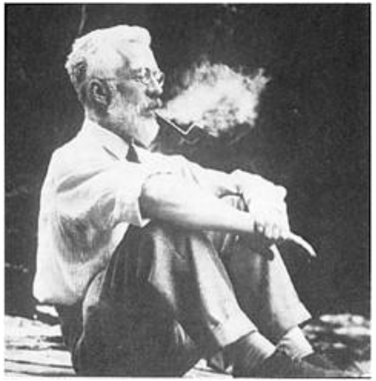


Нелинейный классификатор логического дерева

Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.







На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

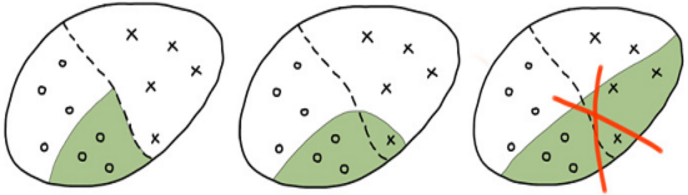
Логическая закономерность

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — обучающая выборка, $y_i = y(x_i)$.

Логическая закономерность (правило, rule) — это предикат $R\colon X \to \{0,1\}$, удовлетворяющий двум требованиям:

1. *интерпретируемость*:
   1) R записывается на естественном языке;
   2) R зависит от небольшого числа признаков (1–7);
2. *информативность* относительно одного из классов $c \in Y$:
   $p_c(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i=c\} \to \max$;
   $n_c(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i\neq c\} \to \min$;

Если $R(x) = 1$, то говорят «R выделяет x» (R covers x).



Требование интерпретируемости

1) $R(x)$ записывается на естественном языке;
2) $R(x)$ зависит от небольшого числа признаков (1–7);

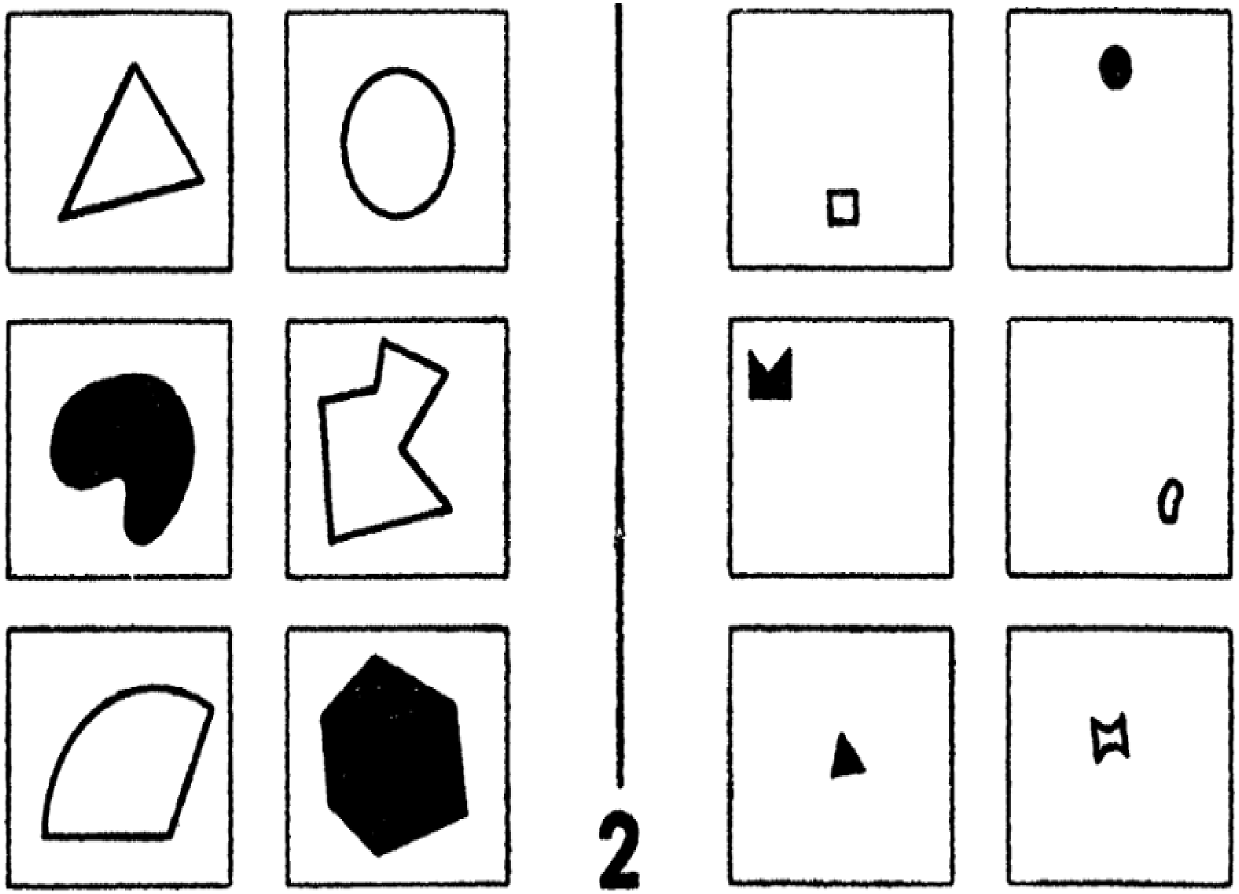
Пример (из области медицины)

Если *«возраст > 60»* ***и*** *«пациент ранее перенёс инфаркт»,* ***то*** *операцию не делать, риск отрицательного исхода 60%.*

Пример (из области кредитного скоринга)

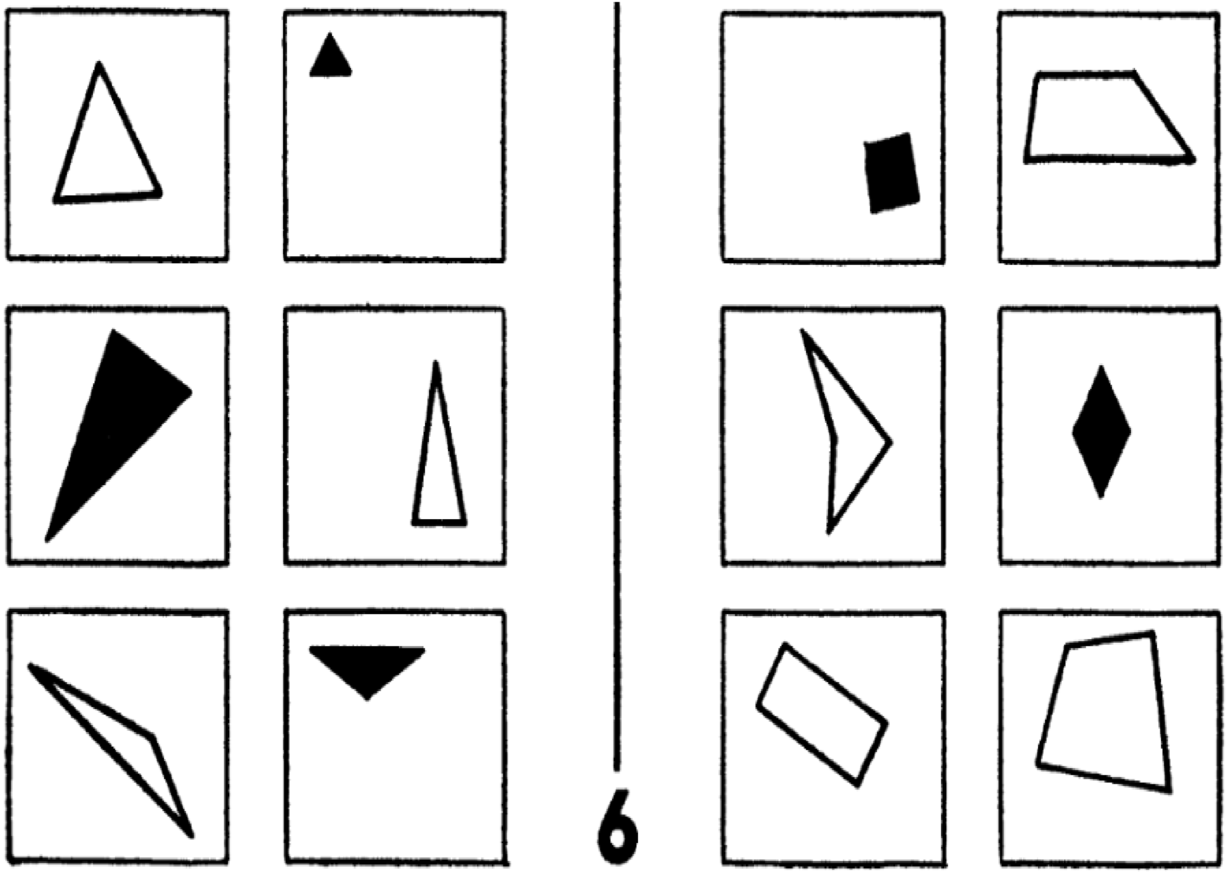
Если *«в анкете указан домашний телефон»*
и *«зарплата > \$2000»* ***и*** *«сумма кредита < \$5000»*
то *кредит можно выдать, риск дефолта 5%.*

Тесты М. М. Бонгарда [Проблема узнавания, 1967]



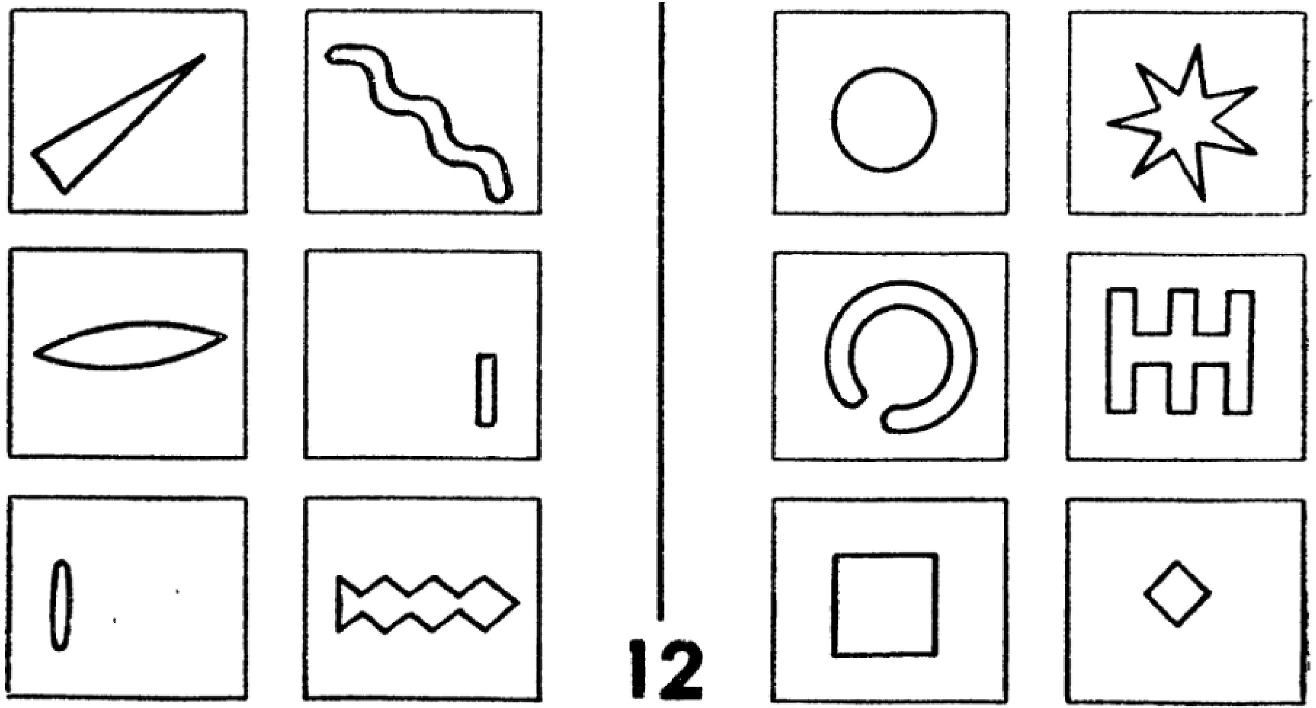
2. Большие фигуры

Тесты М. М. Бонгарда [Проблема узнавания, 1967]



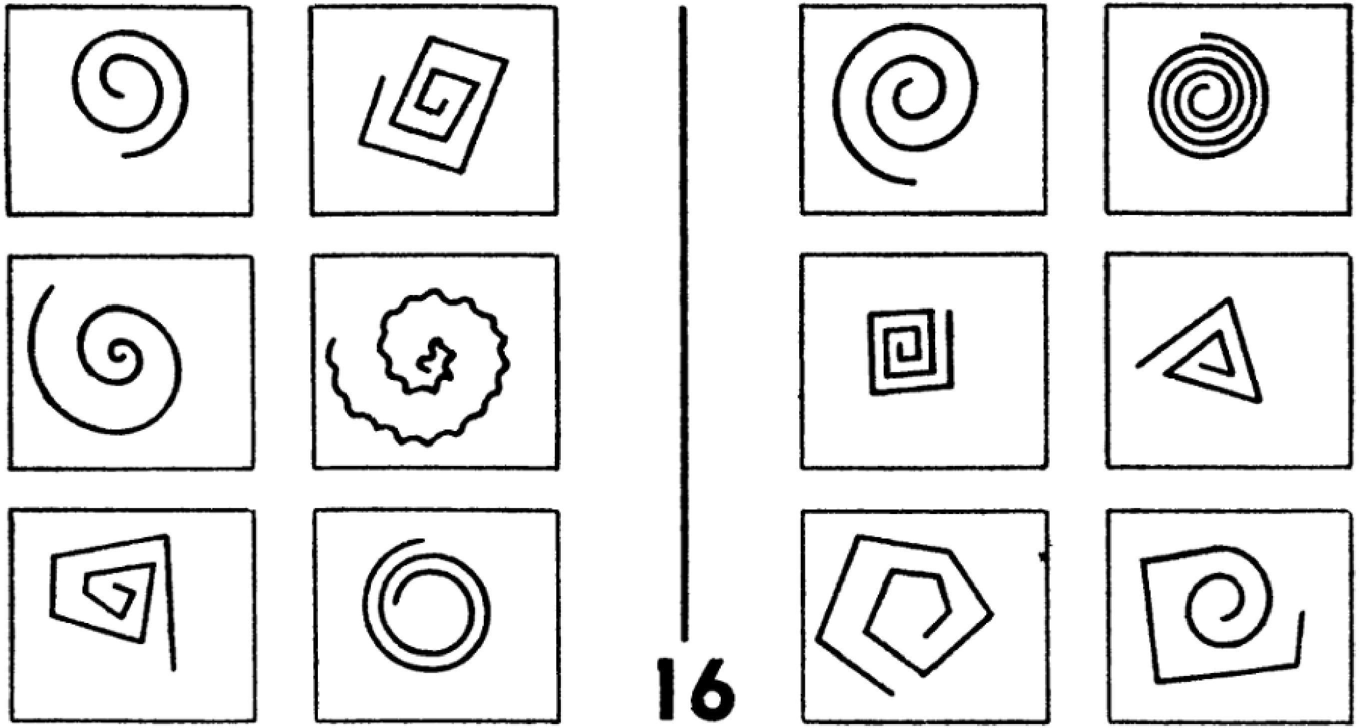
6. Треугольники

Четырехугольники

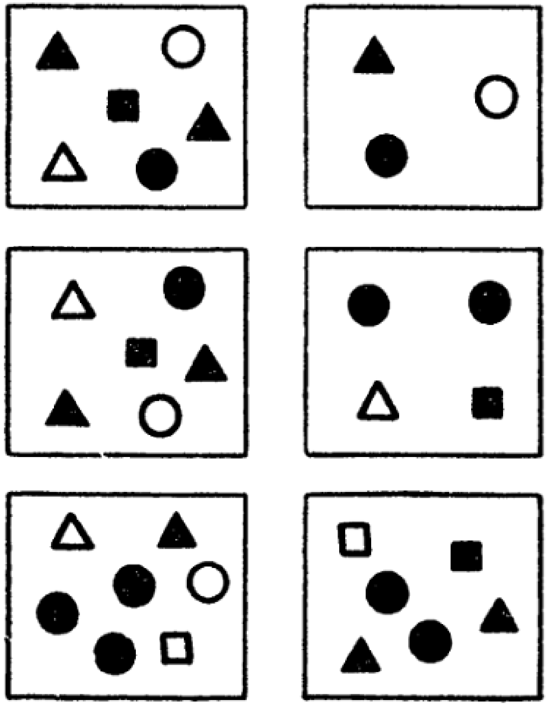

12

12. Выпуклая оболочка фигуры сильно вытянута

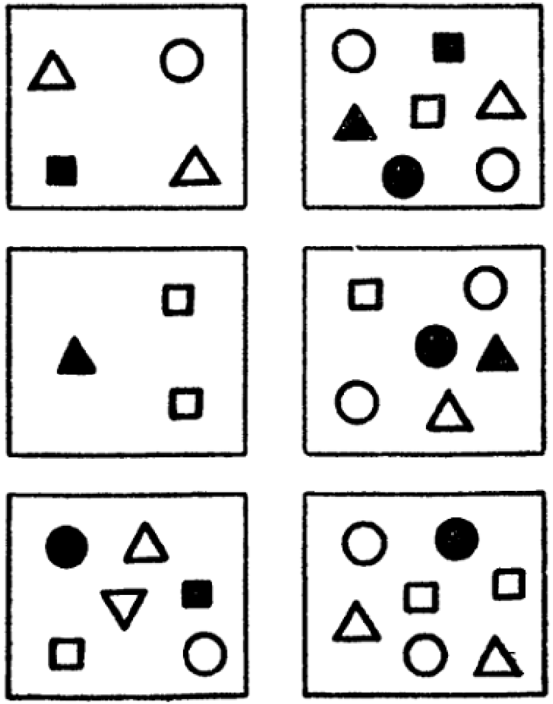
Оболочка слабо вытянута


16

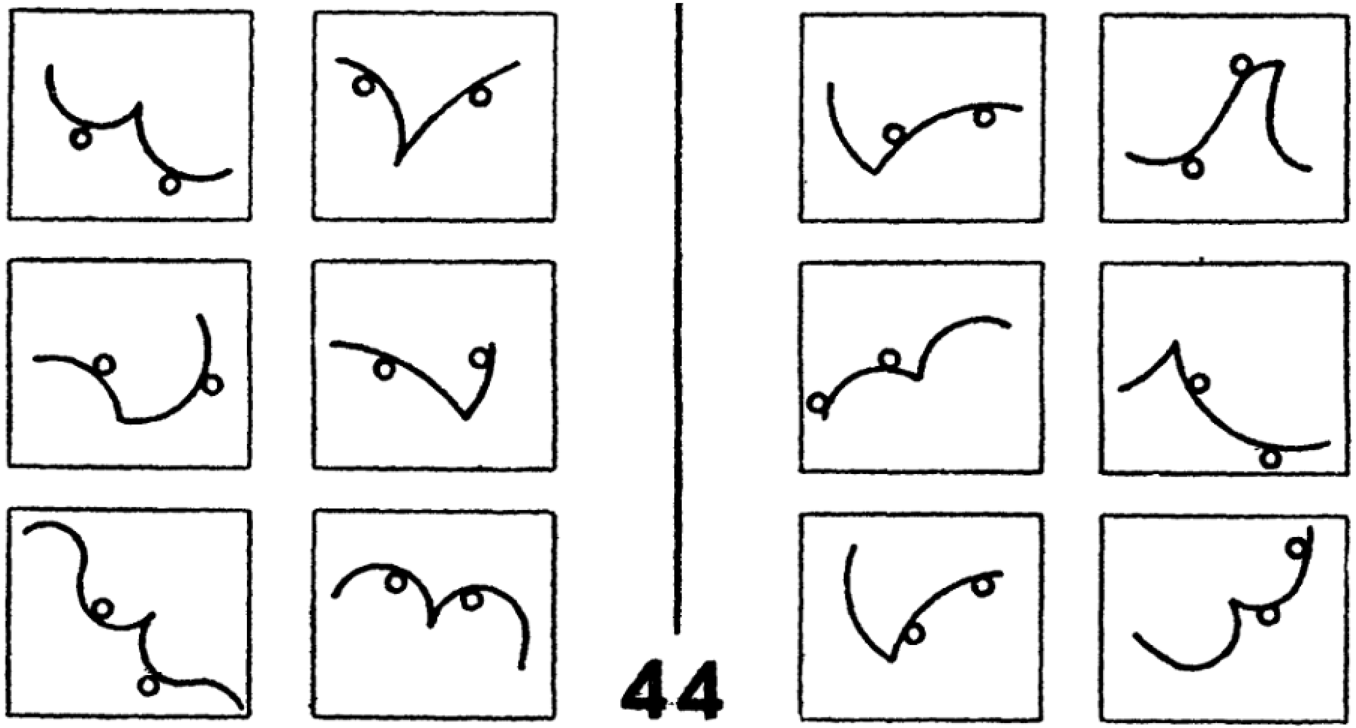
16. Спираль закручивается влево — Спираль закручивается вправо



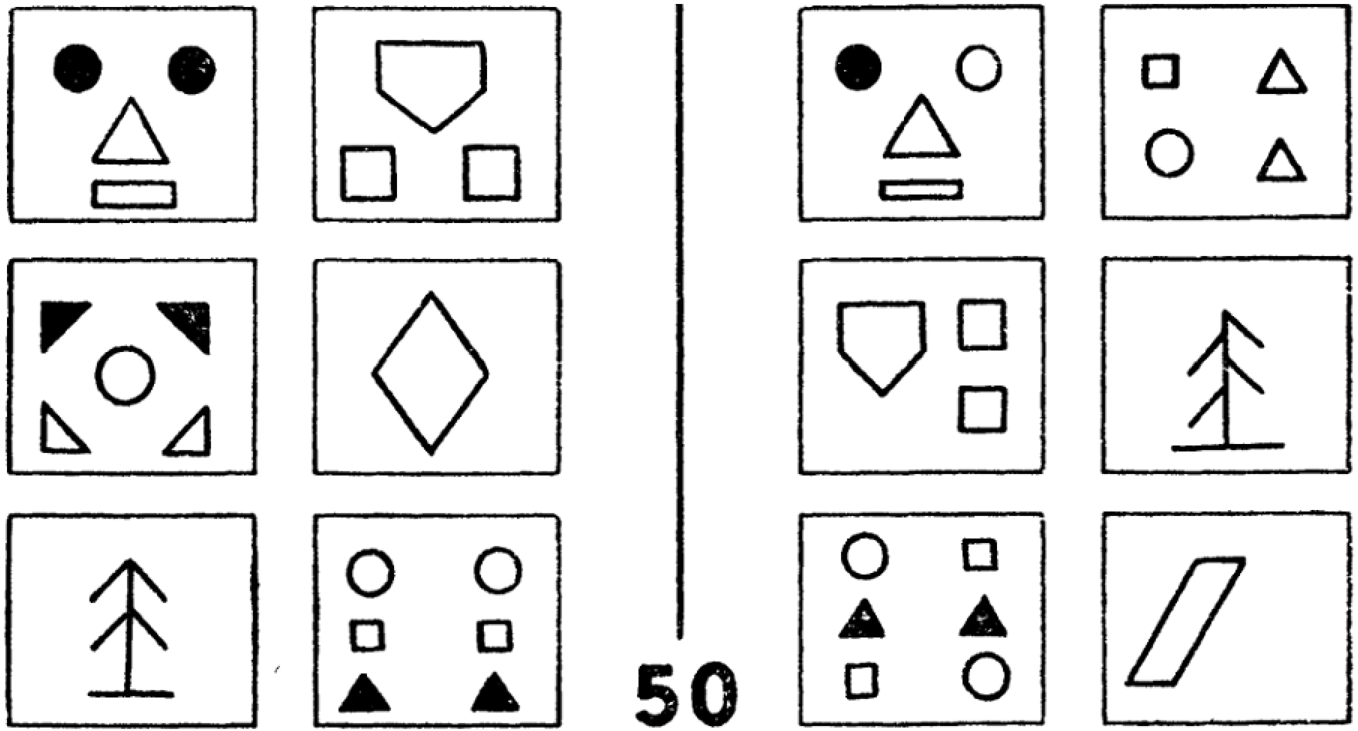
27



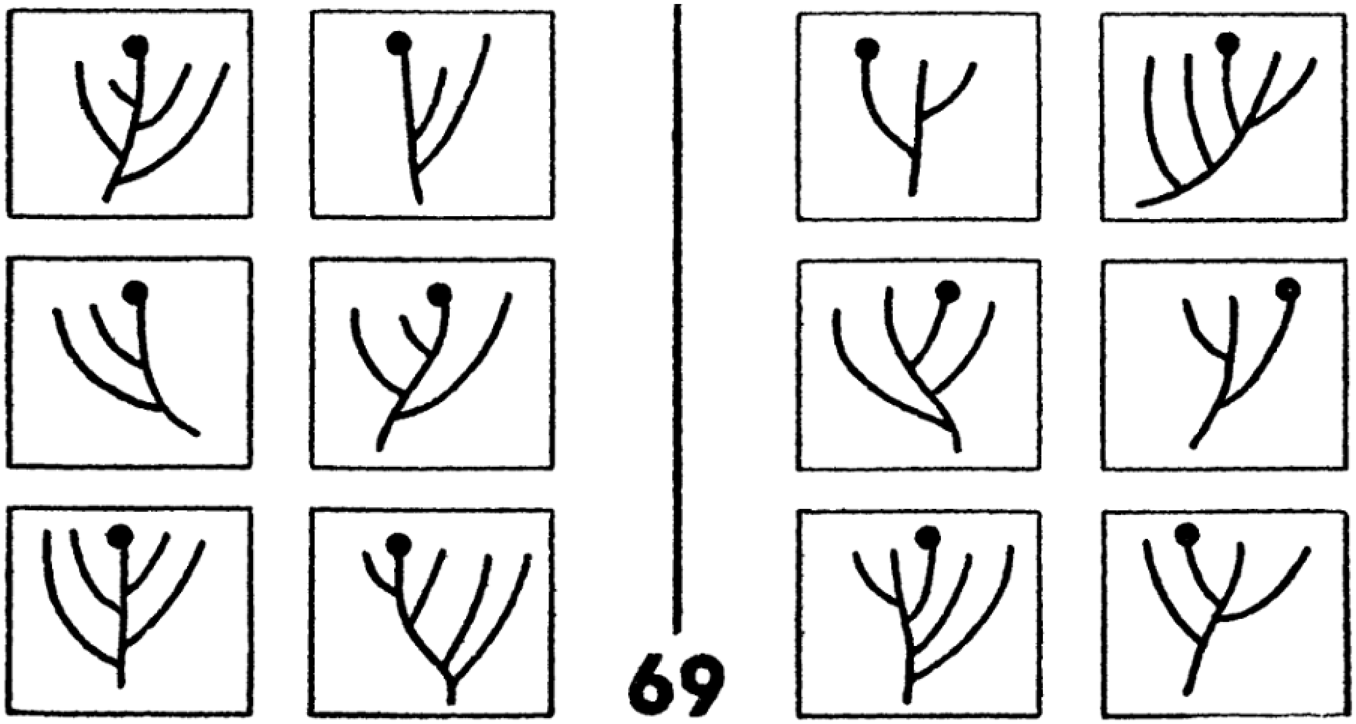
27. Черных фигур больше, чем белых — Белых фигур больше, чем черных


44

44. Кружки на разных дугах — Кружки на одной дуге


50

50. Есть оси симметрии — Нет оси симметрии


69

69. «Ягода растет на главной ветке» — «Ягода растет на боковой ветке»

Виды закономерностей

1. Пороговое условие (решающий пень, decision stump):

$$R(x) = \big[f_j(x) \leqslant a_j\big] \text{ или } \big[a_j \leqslant f_j(x) \leqslant b_j\big].$$

2. *Конъюнкция* пороговых условий:

$$R(x) = \bigwedge_{j \in J} \big[a_j \leqslant f_j(x) \leqslant b_j\big].$$

3. *Синдром* — выполнение не менее d условий из $|J|$, (при $d = |J|$ это конъюнкция, при $d = 1$ — дизъюнкция):

$$R(x) = \Big[\sum_{j \in J} \big[a_j \leqslant f_j(x) \leqslant b_j\big] \geqslant d\Big],$$

Параметры J, a_j, b_j, d настраиваются по обучающей выборке путём оптимизации *критерия информативности*.

Виды закономерностей

4. *Полуплоскость* — линейная пороговая функция:

$$R(x) = \Big[\sum_{j \in J} w_j f_j(x) \geqslant w_0 \Big].$$

5. *Шар* — пороговая функция близости:

$$R(x) = \big[\rho(x, x_0) \leqslant w_0 \big],$$

АВО — алгоритмы вычисления оценок [Ю. И. Журавлёв, 1971]:

$$\rho(x, x_0) = \max_{j \in J} w_j \big| f_j(x) - f_j(x_0) \big|.$$

SCM — машины покрывающих множеств [M. Marchand, 2001]:

$$\rho(x, x_0) = \sum_{j \in J} w_j \big| f_j(x) - f_j(x_0) \big|^{\gamma}.$$

Параметры J, w_j, w_0, x_0 настраиваются по обучающей выборке путём оптимизации *критерия информативности*.

Пример:

при $P = 200$, $N = 100$ и различных p и n.

Простые эвристики не всегда адекватны:

p	n	$p-n$	$p-5n$	$\frac{p}{P}-\frac{n}{N}$	$\frac{p}{n+1}$	IStat$\cdot\ell$	IGain$\cdot\ell$	$\sqrt{p}-\sqrt{n}$
50	0	**50**	50	0.25	50	22.65	23.70	7.07
100	50	**50**	−150	0	1.96	2.33	1.98	2.93
50	9	41	**5**	0.16	**5**	7.87	7.94	4.07
5	0	5	**5**	0.03	**5**	2.04	3.04	2.24
100	0	**100**	100	**0.5**	100	52.18	53.32	10.0
140	20	**120**	40	**0.5**	6.67	37.09	37.03	7.36

Адекватные, но неочевидные критерии:

- энтропийный критерий прироста информации:

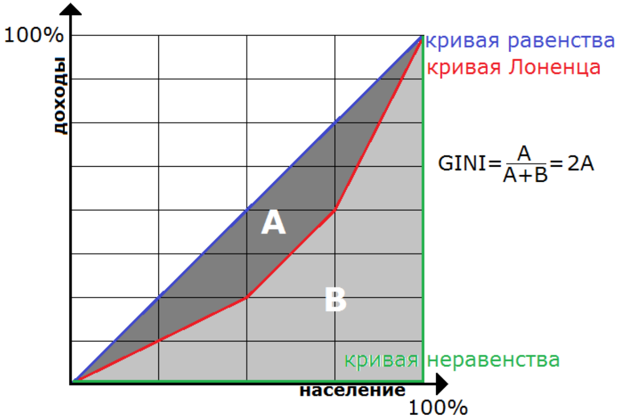
  $$\mathrm{IGain}(p, n) = h\left(\tfrac{P}{\ell}\right) - \tfrac{p+n}{\ell} h\left(\tfrac{p}{p+n}\right) - \tfrac{\ell-p-n}{\ell} h\left(\tfrac{P-p}{\ell-p-n}\right) \to \max,$$

  где $h(q) = -q \log_2 q - (1-q) \log_2 (1-q)$
- критерий Джини (Gini impurity):
  $\mathrm{IGini}(p, n) = \mathrm{IGain}(p, n)$ при $h(q) = 4q(1-q)$
- точный статистический тест Фишера (Fisher's Exact Test):

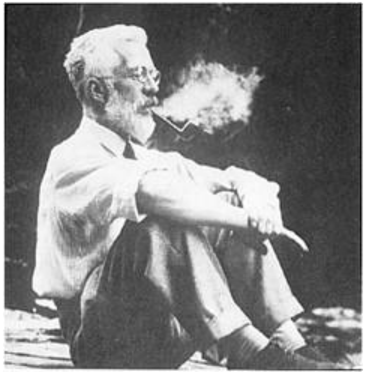
  $$\mathrm{IStat}(p, n) = -\tfrac{1}{\ell} \log_2 \frac{C_P^p C_N^n}{C_{P+N}^{p+n}} \to \max$$
- критерий бустинга:
  $\sqrt{p} - \sqrt{n} \to \max$
- нормированный критерий бустинга:
  $\sqrt{p/P} - \sqrt{n/N} \to \max$

Допустим, в компании работают 4 человека с суммарным доходом 8000$. Равномерное распределение дохода — это 2000$+2000$+2000$+2000$, неравномерное — 0$+0$+0$+8000$. А как оценить неравномерность, скажем, для случая 1000$+1000$+2000$+4000$? Упорядочим сотрудников по возрастанию дохода. Построим кривую (Лоренца) в координатах [процент населения, процент дохода этого населения] — идём по всем сотрудникам и откладывает точки. Для первого — [25%, 12.5%] — это сколько он составляет процентов от всего штата и сколько процентов составляет его доход, для первого и второго — [50%, 25%] — это сколько они составляют процентов и сколько процентов их доход, для первых трёх — [75%, 50%], для всех — [100%,

«Научные основы фашизма» (1927)



Fisher said:



Точный тест Фишера. Пусть X — в.п., выборка X^ℓ — i.i.d.
Гипотеза H_0: $y(x)$ и $R(x)$ — независимые случайные величины.
Тогда вероятность реализации пары (p, n) описывается *гипергеометрическим распределением*:

$$\mathrm{P}(p, n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}, \quad 0 \leqslant p \leqslant P, \quad 0 \leqslant n \leqslant N,$$

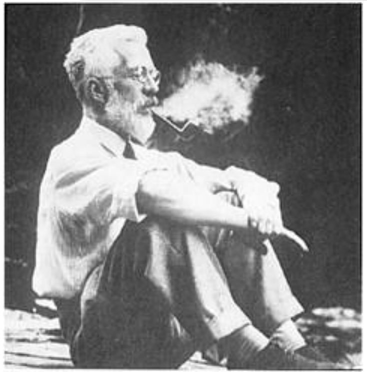
где $C_N^n = \frac{N!}{n!(N-n)!}$ — биномиальные коэффициенты.

Определение

Информативность предиката $R(x)$ относительно класса $c \in Y$:

$$\mathrm{IStat}(p, n) = -\frac{1}{\ell} \log_2 \frac{C_P^p C_N^n}{C_{P+N}^{p+n}},$$

$\mathrm{IStat}(p, n) \geqslant I_0$ — *статистическая закономерность класса c.*




Если Вы не знаете, зачем Вы делаете то, что делаете, лучше полежите спокойно и подумайте.
Незачем зазря увеличивать энтропию.

Энтропийный критерий (вывод из теории информации)

Пусть ω_0, ω_1 — два исхода с вероятностями q и $1 - q$.
Количество информации: $I_0 = -\log_2 q$, $I_1 = -\log_2(1-q)$.
Энтропия — математическое ожидание количества информации:

$$h(q) = -q\log_2 q - (1-q)\log_2(1-q).$$

Энтропия выборки X^ℓ, если исходы — это классы $y{=}c$, $y{\neq}c$:

$$H(y) = h\left(\frac{P}{\ell}\right).$$

Энтропия выборки X^ℓ после получения информации $R(x_i)_{i=1}^{\ell}$:

$$H(y|R) = \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) + \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right).$$

Прирост информации (Information gain, IGain):

$$\mathrm{IGain}(p,n) = H(y) - H(y|R).$$

Определение

Предикат R — закономерность по энтропийному критерию, если $\mathrm{IGain}(p, n) > G_0$ *при некотором* G_0.

Теорема

Энтропийный критерий IGain *асимптотически эквивалентен статистическому* IStat*:*

$$\mathrm{IStat}(p, n) \to \mathrm{IGain}(p, n) \quad \text{при } \ell \to \infty.$$

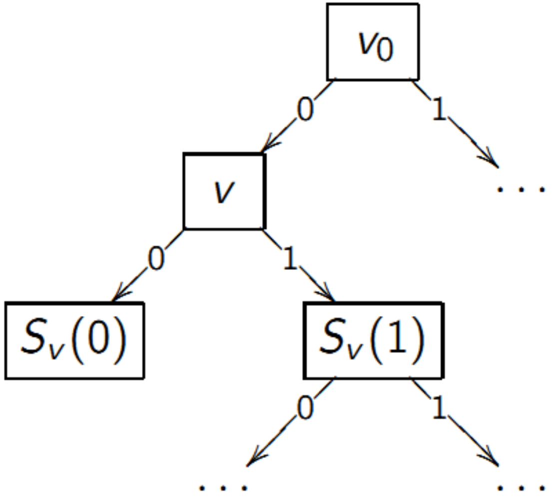
Доказательство:

применить формулу Стирлинга к критерию IStat.

Решающее дерево — алгоритм классификации $a(x)$, задающийся *деревом* (связным ациклическим графом):

1) $V = V_{\text{внутр}} \sqcup V_{\text{лист}}$, $v_0 \in V$ — корень дерева;
2) $v \in V_{\text{внутр}}$: функции $f_v \colon X \to D_v$ и $S_v \colon D_v \to V$, $|D_v| < \infty$;
3) $v \in V_{\text{лист}}$: метка класса $y_v \in Y$.

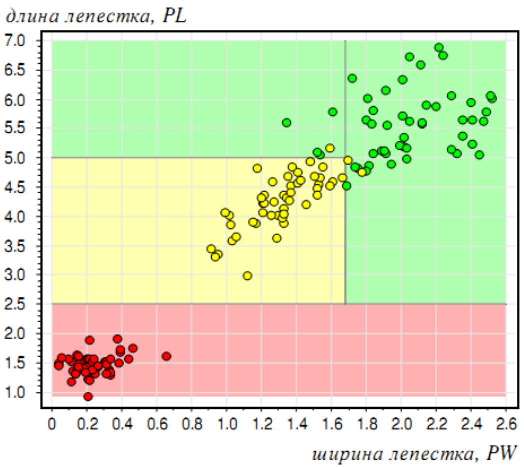
$v := v_0$;
пока $v \in V_{\text{внутр}}$
    $v := S_v(f_v(x))$;
вернуть y_v;

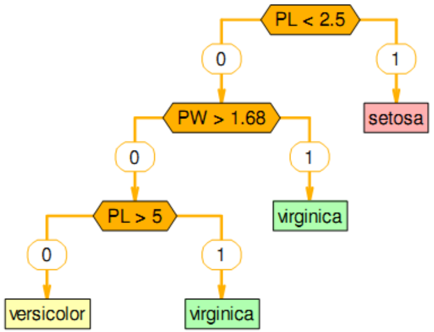


Частный случай: $D_v \equiv \{0, 1\}$
— бинарное решающее дерево

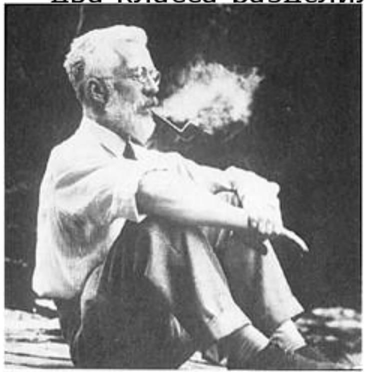
Пример: $f_v(x) = \left[f_j(x) \geqslant \theta_j\right]$

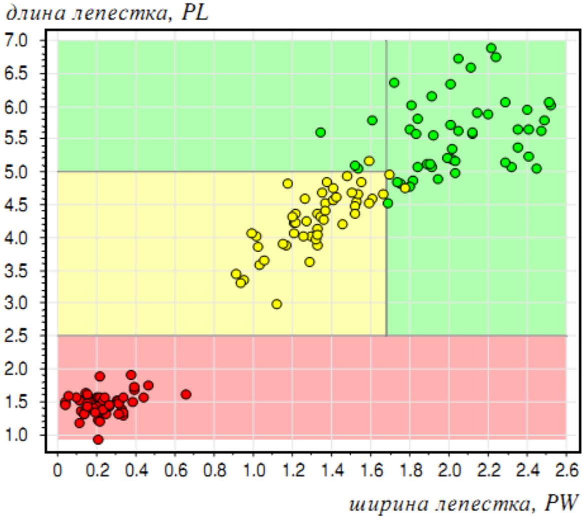
Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.

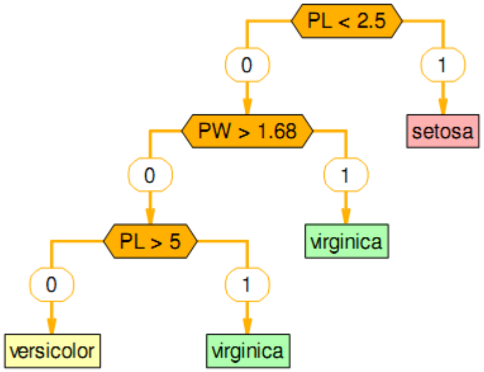




На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

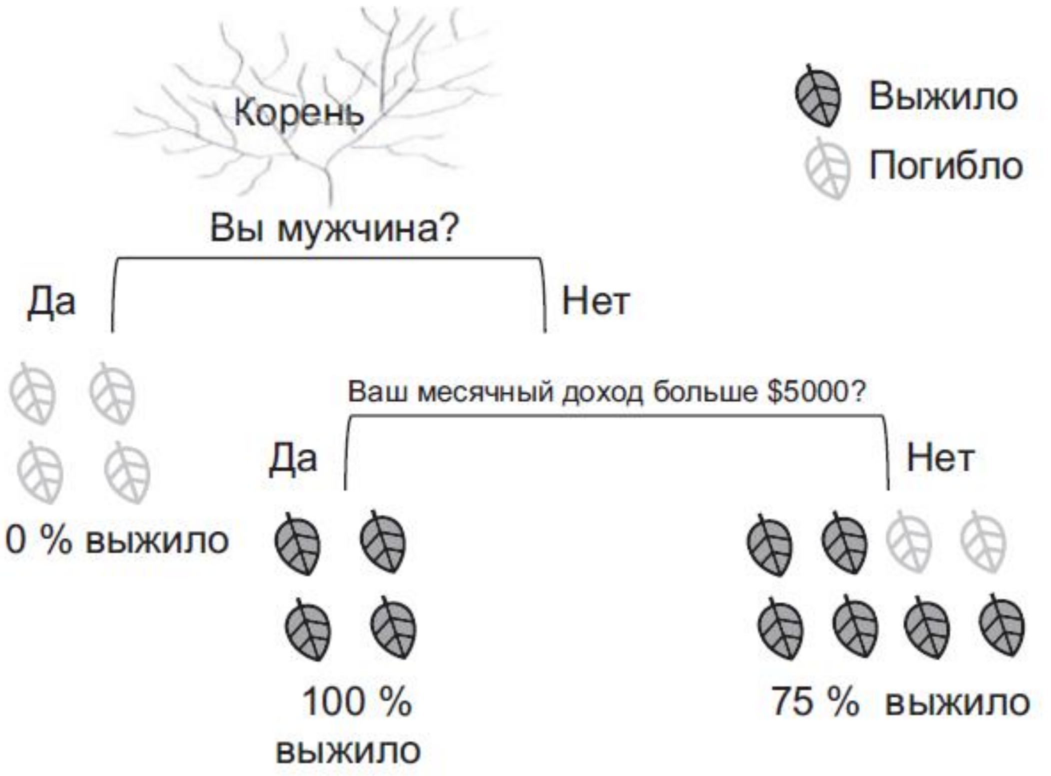




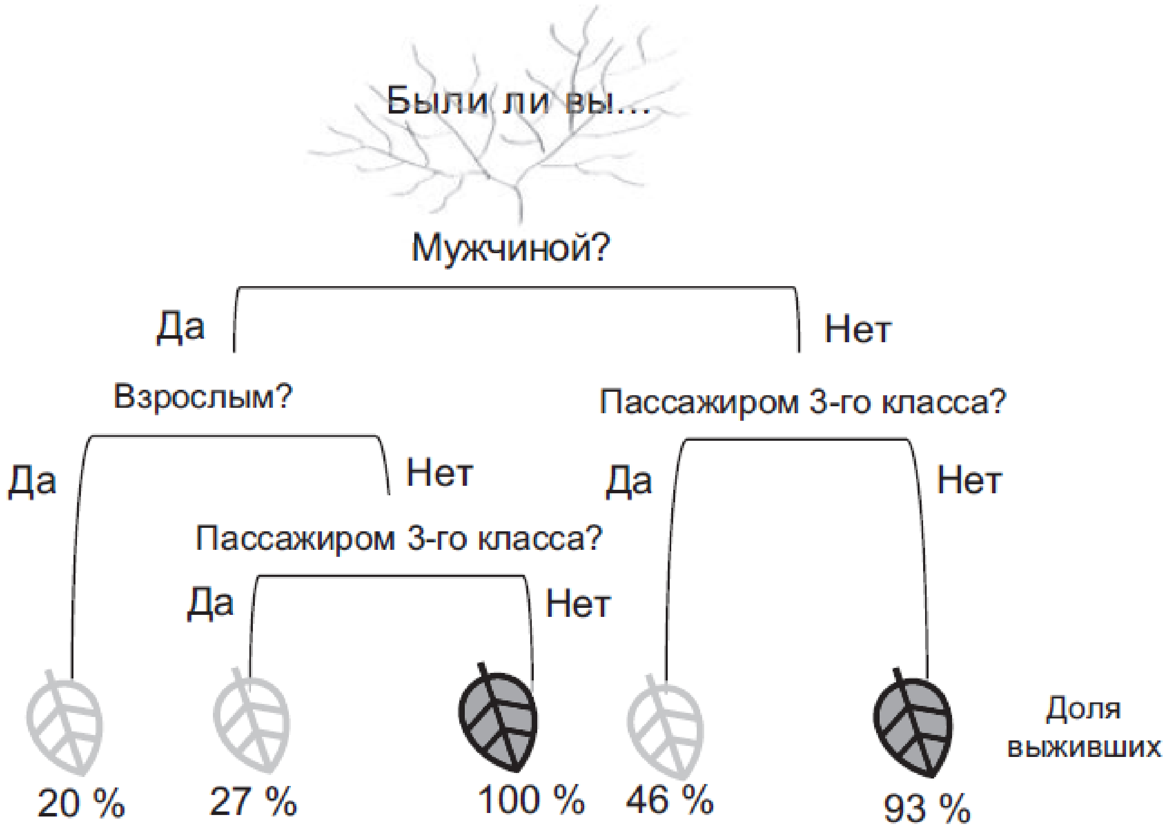


setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

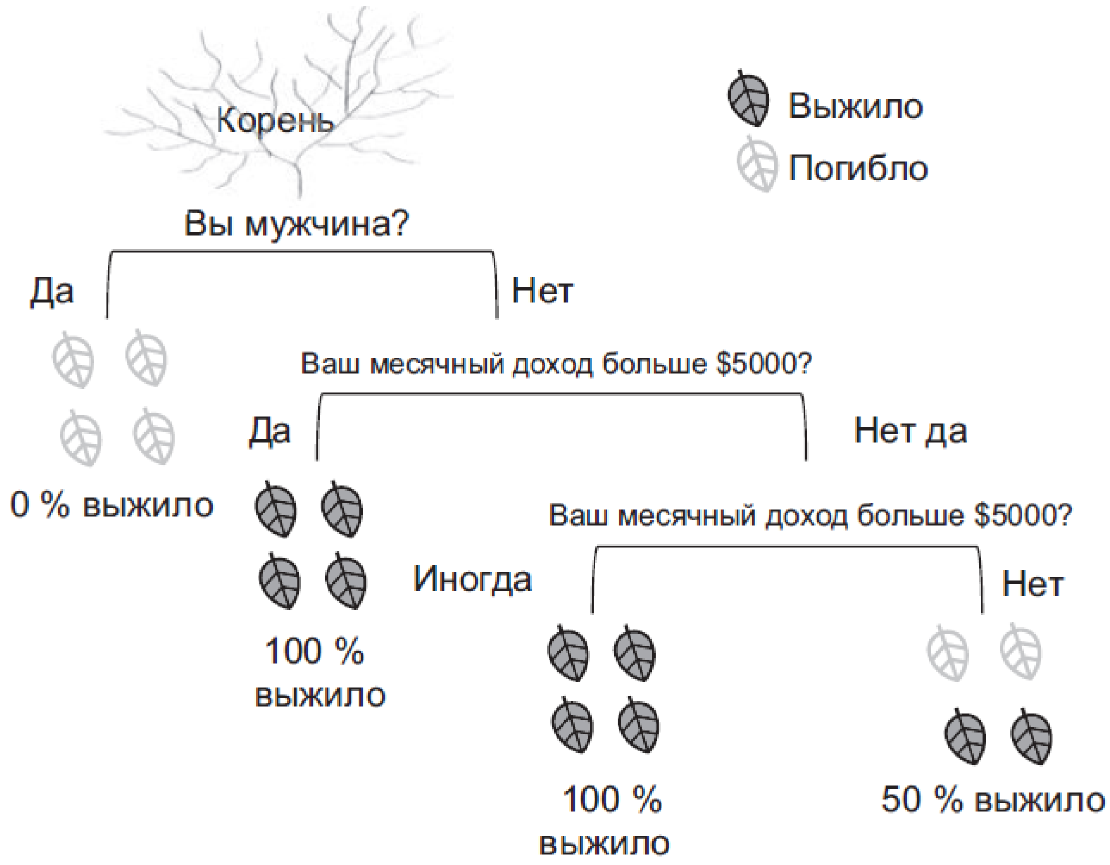
Титаник: почему они спаслись?



Титаник: ты бы выжил?



Титаник: ты бы выжил?



Класс DecisionTreeClassifier в Scikit-learn Python

Полное описание с конструктором по умолчанию:

class sklearn.tree.DecisionTreeClassifier (criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Класс DecisionTreeClassifier в Scikit-learn Python

Параметры:

- max_depth - максимальная глубина дерева
- max_features - максимальное число признаков, по которым ищется лучшее разбиение в дереве (это нужно потому, что при большом количестве признаков будет "дорого" искать лучшее (по критерию типа прироста информации) разбиение среди *всех* признаков)
- min_samples_leaf - минимальное число объектов в листе. У этого параметра есть понятная интерпретация: скажем, если он равен 5, то дерево будет порождать только те классифицирующие правила, которые верны как минимум для 5 объектов

Класс DecisionTreeClassifier в Scikit-learn Python

criterion=’gini’ или ’entropy’

splitter=’best’, max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)

Библиотека anytree для Python

- установка: pip install anytree
- включение, например: from anytree import Node, RenderTree
- создание дерева: pm=Node(“PM”, parent=fmi)…


time for a boost