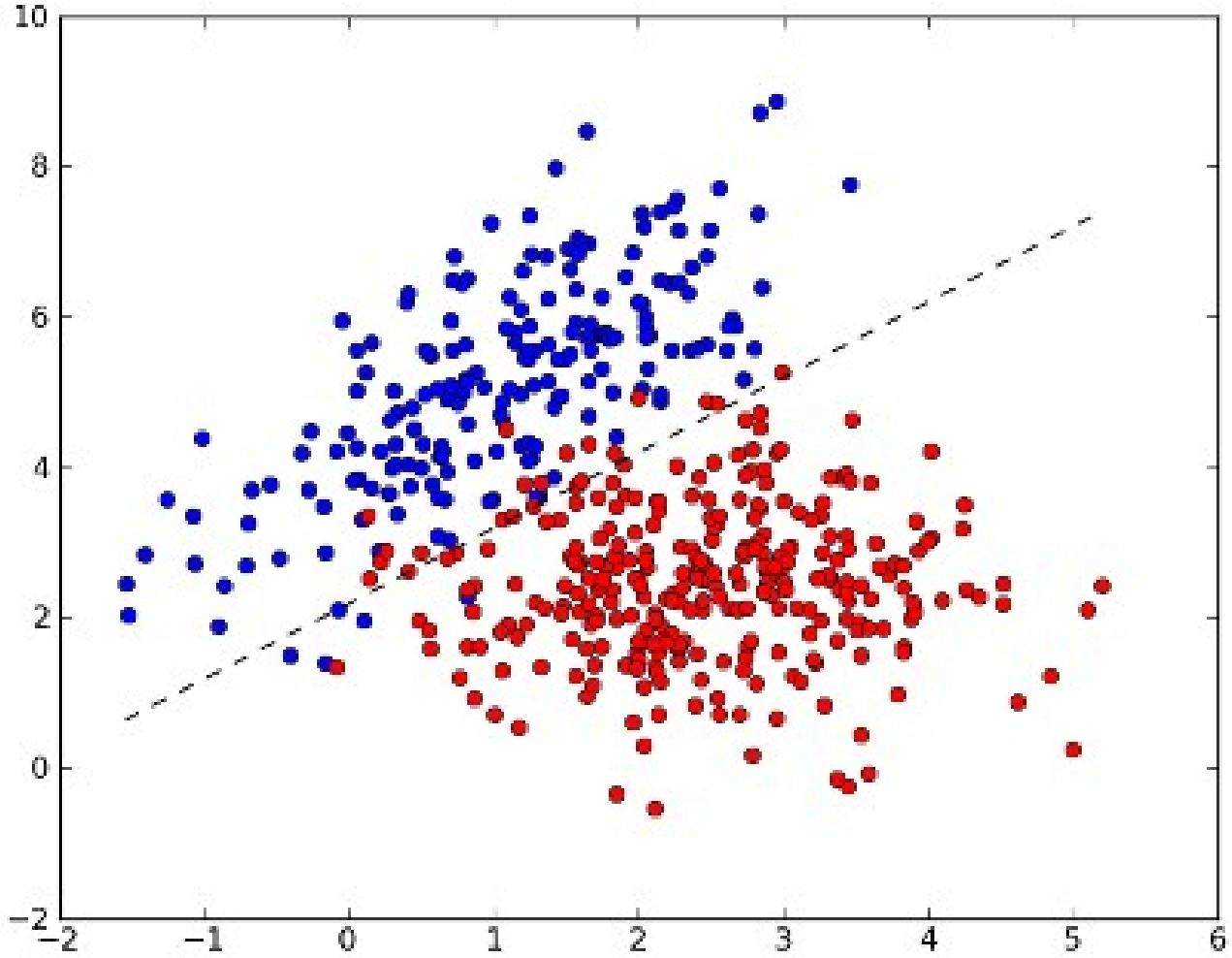
Машинное обучение
Линейные алгоритмы классификации



Содержание лекции

- Общая формула линейного классификатора
- Метод стохастического градиента
- Частные случаи
- Обоснование метода СГ
- Выступ объекта для лин. классификатора
- ROC и PR-кривые

Классификация линейной функцией

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell, \quad x_i \in \mathbb{R}^n, \quad y_i \in \{-1, +1\}$

1. Модель классификации — *линейная*:

$$a(x, w) = \text{sign}\langle x, w\rangle$$

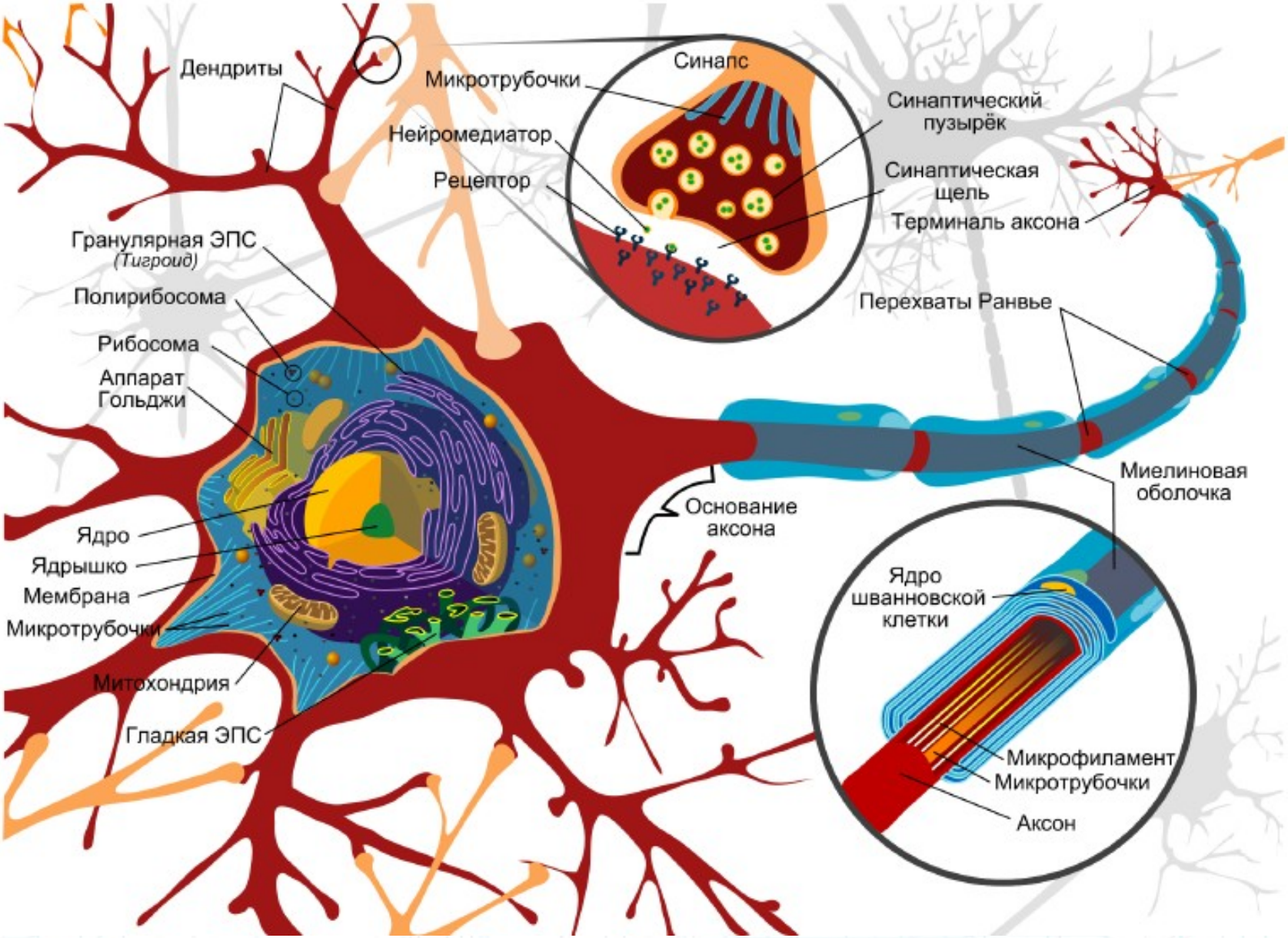
2. Функция потерь — бинарная или её аппроксимация:

$$\mathscr{L}(a, y) = \big[\langle x_i, w\rangle y_i < 0\big] \leqslant \mathscr{L}\big(\langle x_i, w\rangle y_i\big)$$

3. Метод обучения — *минимизация эмпирического риска*:

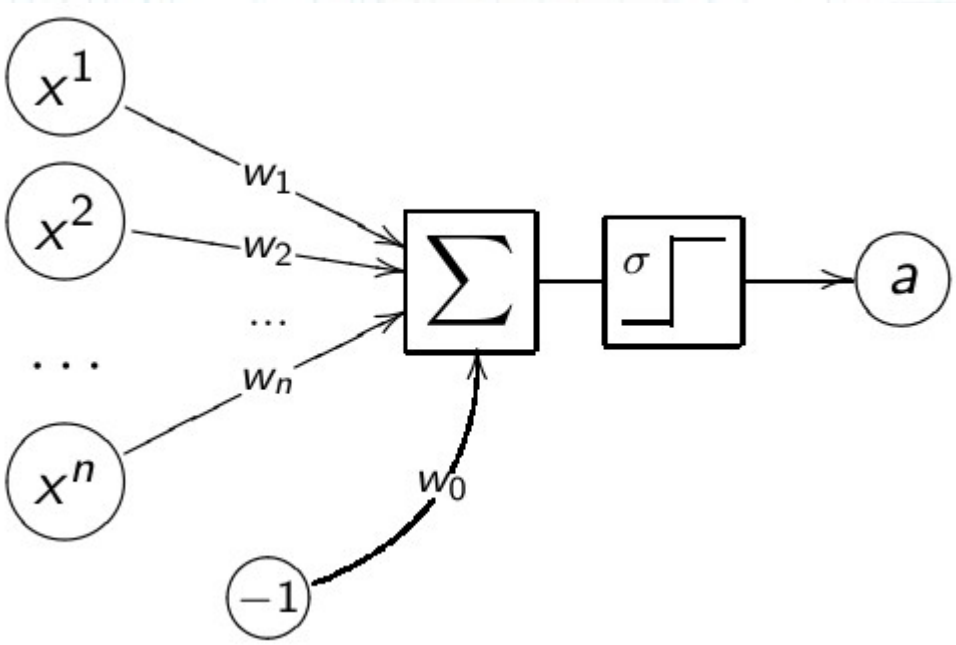
$$Q(w) = \sum_{i=1}^{\ell} \big[a(x_i, w) y_i < 0\big] \leqslant \sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle y_i\big) \to \min_w$$

Линейный классификатор – математическая модель нейрона



Линейная модель нейрона МакКаллока-Питтса (1943)

$$a(x, w) = \sigma(\langle w, x \rangle) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0 \right)$$



Градиентный метод численной минимизации

Минимизация эмпирического риска:

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(g(w, x_i), y_i\big) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) \to \min_w .$$

Численная минимизация методом *градиентного спуска*:

$$w^{(0)} := \text{начальное приближение;}$$

$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}), \qquad \nabla Q(w) = \left(\frac{\partial Q(w)}{\partial w_j}\right)_{j=0}^{n},$$

где h — *градиентный шаг*, называемый также *темпом обучения*.

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{\ell} \nabla \mathscr{L}_i\big(w^{(t)}\big).$$

Идея ускорения сходимости:

брать (x_i, y_i) по одному и сразу обновлять вектор весов.

Метод стохастического градиента

Вход: выборка X^ℓ, темп обучения h, темп забывания λ

Выход: вектор весов w

1: инициализировать веса w_j, $j = 0, \ldots, n$;

2: инициализировать оценку функционала: $\bar{Q} := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}_i(w)$;

3: **повторять**

4: выбрать объект x_i из X^ℓ случайным образом;

5: вычислить потерю: $\varepsilon_i := \mathscr{L}_i(w)$;

6: сделать градиентный шаг: $w := w - h\nabla \mathscr{L}_i(w)$;

7: оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda\varepsilon_i$;

8: **пока** значение $\bar{Q}$ и/или веса w не сойдутся;

Пересчет функционала

Проблема: после каждого шаг w по одному объекту x_i, не хотелось бы оценивать Q по всей выборке $x_1, \ldots, x_\ell$.

Решение: использовать рекуррентную формулу.

Среднее арифметическое $\bar{Q}_m = \frac{1}{m}\sum_{i=1}^{m} \varepsilon_i$:

$$\bar{Q}_m = (1 - \tfrac{1}{m})\bar{Q}_{m-1} + \tfrac{1}{m}\varepsilon_m.$$

Экспоненциальное скользящее среднее

$$\bar{Q}_m = \lambda\varepsilon_m + \lambda(1-\lambda)\varepsilon_{m-1} + \lambda(1-\lambda)^2\varepsilon_{m-2} + \lambda(1-\lambda)^3\varepsilon_{m-3} + \ldots$$

$$\bar{Q}_m := (1 - \lambda)\bar{Q}_{m-1} + \lambda\varepsilon_m.$$

Чем больше λ, тем быстрее забывается предыстория ряда.
Параметр λ называется *темпом забывания*.

Правило Хэбба

Задача классификации: $x_i \in \mathbb{R}^{n+1}$, $y_i \in \{-1, +1\}$,

$$a(x, w) = \text{sign}\langle w, x\rangle, \qquad \mathscr{L}_i(w) = \big(-\langle w, x_i\rangle y_i\big)_+.$$

Градиентный шаг SG — правило Хэбба [1949]:

$$\textbf{если } \langle w, x_i\rangle y_i < 0 \textbf{ то } w := w + h x_i y_i,$$

То же самое для случая $y_i \in \{0, 1\}$,

$$a(x, w) = \big[\langle w, x\rangle > 0\big], \qquad \mathscr{L}_i(w) = \big(a(x_i, w) - y_i\big)\langle w, x_i\rangle,$$

Градиентный шаг SG — персептрон Розенблатта [1957]:

$$w := w - h\big(a(x_i, w) - y_i\big)x_i.$$

Дельта-правило ADALINE

Задача регрессии: $x_i \in \mathbb{R}^{n+1}$, $y_i \in \mathbb{R}$

Адаптивный линейный элемент ADALINE [Видроу, Хофф 1960]:

$$a(x, w) = \langle w, x \rangle, \qquad \mathscr{L}_i(w) = (\langle w, x_i \rangle - y_i)^2.$$

Градиентный шаг SG — дельта-правило (delta-rule):

$$w := w - h\underbrace{(\langle w, x_i \rangle - y_i)}_{\Delta_i} x_i,$$

Δ_i — ошибка алгоритма $a(x, w)$ на объекте x_i.

Формально совпадает с правилом персептрона Розенблатта!

Правило Хэбба

$$a(x, \mathrm{w}) = \mathrm{sign}\langle \mathrm{w}, x\rangle$$

$$\text{если} \quad y_i\langle \mathrm{w}, x_i\rangle < 0, \quad \text{то} \quad \mathrm{w} := \mathrm{w} + hx_iy_i$$

$$\langle \mathrm{w} + hx_iy_i, x_i\rangle = \langle \mathrm{w}, x_i\rangle + hy_i||x_i||^2$$

Теорема (Новиков, 1962)

Пусть выборка X^ℓ линейно разделима:

$$\exists \tilde{w}, \ \exists \delta > 0\colon \langle \tilde{w}, x_i\rangle y_i > \delta \ \text{ для всех } \ i = 1, \ldots, \ell.$$

Тогда Алгоритм SG с правилом Хэбба находит вектор весов w,

- разделяющий обучающую выборку без ошибок;
- при любом начальном положении $w^{(0)}$;
- при любом темпе обучения $h > 0$;
- независимо от порядка предъявления объектов x_i;
- за конечное число исправлений вектора w;
- если $w^{(0)} = 0$, то число исправлений $t_{\max} \leqslant \frac{1}{\delta^2} \max \|x_i\|^2$.

Доказательство

Рассмотрим $\cos(\widehat{\tilde{w}, w^t}) = \dfrac{\langle \tilde{w}, w^t \rangle}{\|w^t\|}$ после t-го исправления w^t, при $\|\tilde{w}\| = 1$.

При t-м исправлении $\langle x_i, w^{t-1} \rangle y_i < 0$. В силу линейной разделимости

$$\langle \tilde{w}, w^t \rangle = \langle \tilde{w}, w^{t-1} \rangle + h\langle \tilde{w}, x_i \rangle y_i > \langle \tilde{w}, w^{t-1} \rangle + h\delta > \langle \tilde{w}, w^0 \rangle + th\delta.$$

В силу ограниченности выборки, $\|x_i\| < D$:

$$\|w^t\|^2 = \|w^{t-1}\|^2 + h^2\|x_i\|^2 + 2h\langle w^{t-1}, x_i \rangle y_i < \|w^{t-1}\|^2 + h^2 D^2 < \|w^0\|^2 + th^2 D^2.$$

Подставим эти соотношения в выражение для косинуса:

$$\cos(\widehat{\tilde{w}, w^t}) > \frac{\langle \tilde{w}, w^0 \rangle + th\delta}{\sqrt{\|w^0\|^2 + th^2 D^2}} \to \infty \text{ при } t \to \infty.$$

$\cos \leqslant 1$, значит при некотором t не найдётся ни одного $x_i \in X^\ell$ такого, что $\langle w^t, x_i \rangle y_i < 0$, то есть выборка окажется поделенной безошибочно.

Если $w^0 = 0$, то из условия $\cos = \dfrac{\sqrt{t}\delta}{D} \leqslant 1$ находим $t_{\max} = \left(\dfrac{D}{\delta}\right)^2$.

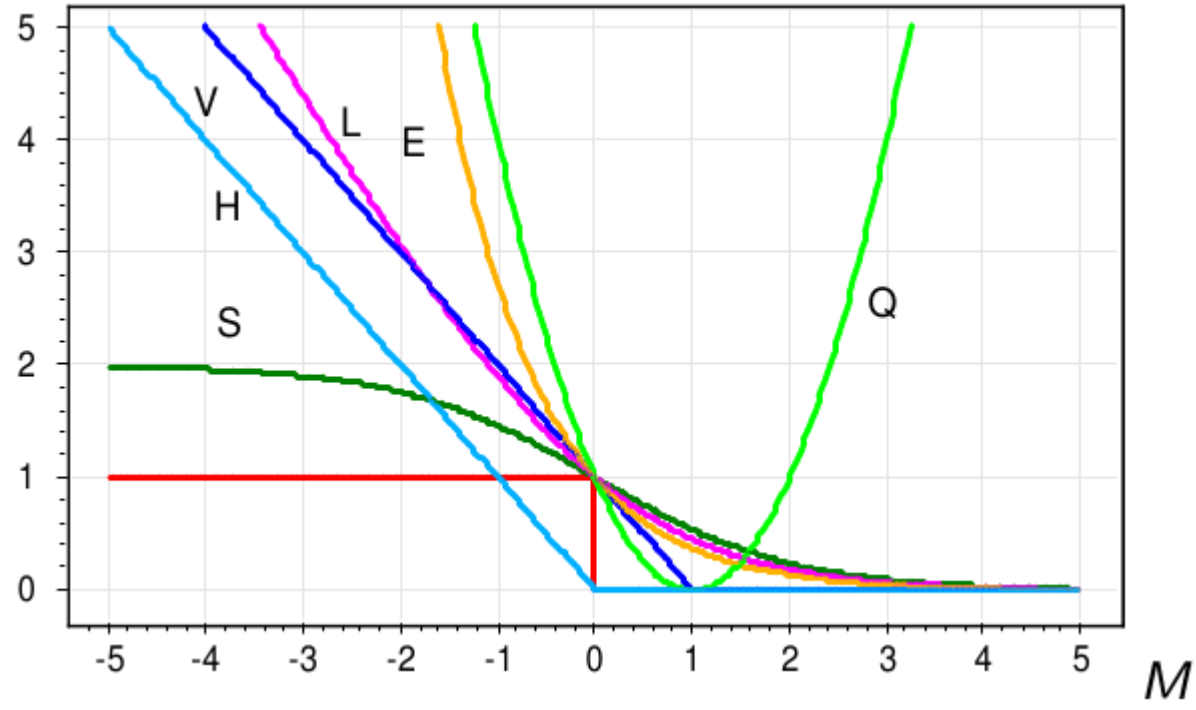
Понятие выступа для линейного классификатора

- Линейный классификатор:

$$a(x, w) = \text{sign}(x, w)$$

- $(x, w) = 0$ — разделяющая гиперплоскость,
- $M_i(w) = (w, x_i)\, y_i$ - отступ/выступ объекта x_i

Часто используемые непрерывные функции потерь



$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);

$H(M) = (-M)_+$ — кусочно-линейная (Hebb's rule);

$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR);

$Q(M) = (1 - M)^2$ — квадратичная (FLD);

$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная (ANN);

$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);

$[M < 0]$ — пороговая функция потерь.

Начальное значение w

1. $w_j := 0$ для всех $j = 0, \ldots, n$;
2. небольшие случайные значения:
   $w_j := \text{random}\left(-\frac{1}{2n}, \frac{1}{2n}\right)$;
3. $w_j := \frac{\langle y, f_j \rangle}{\langle f_j, f_j \rangle}$, $f_j = \left(f_j(x_i)\right)_{i=1}^{\ell}$ — вектор значений признака.

   Упражнение: доказать, что оценка w оптимальна, если
   1) функция потерь квадратична и
   2) признаки некоррелированы, $\langle f_j, f_k \rangle = 0$, $j \neq k$.
4. $w_j := \ln \frac{\sum_i [y_i = +1] f_j(x_i)}{\sum_i [y_i = -1] f_j(x_i)}$ — для классификации, $Y = \{-1, +1\}$
5. обучение по небольшой случайной подвыборке объектов;
6. мультистарт: многократные запуски из разных случайных начальных приближений и выбор лучшего решения.

Порядок предъявления x_i

Возможны варианты:

1. *перетасовка объектов* (shuffling):
   попеременно брать объекты из разных классов;
2. чаще брать те объекты, на которых была допущена бóльшая ошибка
   (чем меньше M_i, тем больше вероятность взять объект)
   (чем меньше $|M_i|$, тем больше вероятность взять объект);
3. вообще не брать «хорошие» объекты, у которых $M_i > \mu_+$
   (при этом немного ускоряется сходимость);
4. вообще не брать объекты-«выбросы», у которых $M_i < \mu_-$
   (при этом может улучшиться качество классификации);

Параметры μ_+, μ_- придётся подбирать.

Выбор шага h

1. сходимость гарантируется (для выпуклых функций) при
$$h_t \to 0, \quad \sum_{t=1}^{\infty} h_t = \infty, \quad \sum_{t=1}^{\infty} h_t^2 < \infty,$$
в частности можно положить $h_t = 1/t$;
2. *метод скорейшего градиентного спуска*:
$$\mathscr{L}_i\big(w - h\nabla\mathscr{L}_i(w)\big) \to \min_h,$$
позволяет найти *адаптивный шаг* h^*;

   Упражнение: доказать, что при квадратичной функции потерь $h^* = \|x_i\|^{-2}$.
3. пробные случайные шаги
   — для «выбивания» из локальных минимумов;

Достоинства и недостатки

Достоинства:

1. легко реализуется;
2. легко обобщается на любые $g(x, w)$, $\mathscr{L}(a, y)$;
3. возможно динамическое (потоковое) обучение;
4. на сверхбольших выборках можно получить неплохое решение, даже не обработав все (x_i, y_i);
5. всё чаще применяется для Big Data

Недостатки:

1. возможна расходимость или медленная сходимость;
2. застревание в локальных минимумах;
3. подбор комплекса эвристик является искусством;
4. проблема переобучения;

Проблема переобучения

Возможные причины переобучения:

1. слишком мало объектов; слишком много признаков;
2. линейная зависимость (мультиколлинеарность) признаков:
   пусть построен классификатор: $a(x, w) = \text{sign}\langle w, x\rangle$;
   мультиколлинеарность: $\exists u \in \mathbb{R}^{n+1}$: $\forall x \;\; \langle u, x\rangle \equiv 0$;
   тогда $\forall \gamma \in \mathbb{R} \quad a(x, w) = \text{sign}\langle w + \gamma u, x\rangle$

Симптоматика:

1. слишком большие веса $|w_j|$ разных знаков;
2. неустойчивость $a(x, w)$;
3. $Q(X^\ell) \ll Q(X^k)$;

Терапия:

1. регуляризация (сокращение весов, weight decay);
2. ранний останов (early stopping);

Регуляризация

Штраф за увеличение нормы вектора весов:

$$\widetilde{\mathscr{L}}_i(w) = \mathscr{L}_i(w) + \frac{\tau}{2}\|w\|^2 = \mathscr{L}_i(w) + \frac{\tau}{2}\sum_{j=1}^{n} w_j^2 \to \min_w .$$

Градиент:

$$\nabla\widetilde{\mathscr{L}}_i(w) = \nabla\mathscr{L}_i(w) + \tau w.$$

Модификация градиентного шага:

$$w := w(1 - h\tau) - h\nabla\mathscr{L}_i(w).$$

Разные штрафы за ошибки

Задача классификации на два класса, $y_i \in \{-1, +1\}$.
Модель классификации: $a(x; w, w_0) = \text{sign}\big(g(x, w) - w_0\big)$.
Чем меньше w_0, тем больше x_i: $a(x_i) = +1$.

Пусть λ_y — штраф за ошибку на объекте класса y.
Функция потерь теперь зависит от штрафов:

$$\mathscr{L}(a, y) = \lambda_{y_i}\big[a(x_i; w, w_0) \neq y_i\big] = \lambda_{y_i}\big[(g(x_i, w) - w_0)y_i < 0\big].$$

Проблема

На практике штрафы $\{\lambda_y\}$ могут пересматриваться

- Нужен удобный способ выбора w_0 в зависимости от $\{\lambda_y\}$, не требующий построения w заново.
- Нужна характеристика качества модели $g(x, w)$, не зависящая от штрафов $\{\lambda_y\}$ и численности классов.

ROC-кривая

- ROC – receiver operating characteristic
- Каждая точка кривой соответствует одному значению порога (цен за ошибки, w_0)
- По оси X:
  FPR (false positive rate) – процент объектов с y=-1 и a(x)=+1 среди всех y=-1
- По оси Y:
  TPR (true positive rate) – процент объектов с y=+1 и a(x)=+1 среди всех y=+1

FPR и TPR

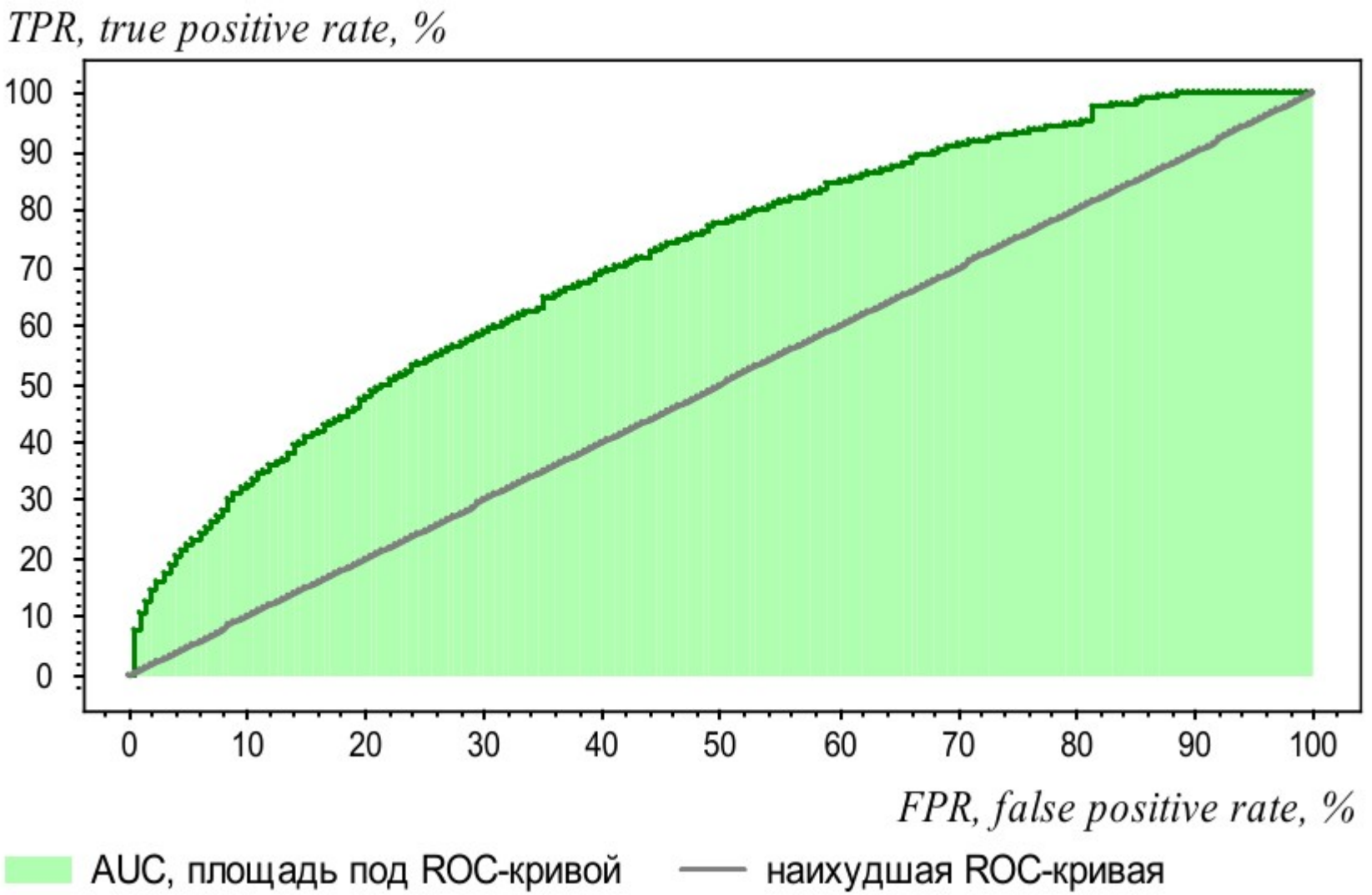
		Predicted	
		Positive (PP)	Negative (PN)
Actual	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Другие названия:
полнота (recall),
чувствительность (sensitivity)

Пример


TPR, true positive rate, %
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
FPR, false positive rate, %
AUC, площадь под ROC-кривой
наихудшая ROC-кривая

Алгоритм построения ROC-кривой

Вход: выборка X^ℓ; дискриминантная функция $g(x, w)$;

Выход: $\{(\mathrm{FPR}_i, \mathrm{TPR}_i)\}_{i=0}^{\ell}$, AUC — площадь под ROC-кривой.

1: $\ell_y := \sum_{i=1}^{\ell} [y_i = y]$, для всех $y \in Y$;

2: упорядочить выборку X^ℓ по убыванию значений $g(x_i, w)$;

3: поставить первую точку в начало координат:
$(\mathrm{FPR}_0, \mathrm{TPR}_0) := (0, 0)$; $\mathrm{AUC} := 0$;

4: **для** $i := 1, \ldots, \ell$

5: **если** $y_i = -1$ **то** сместиться на один шаг вправо:

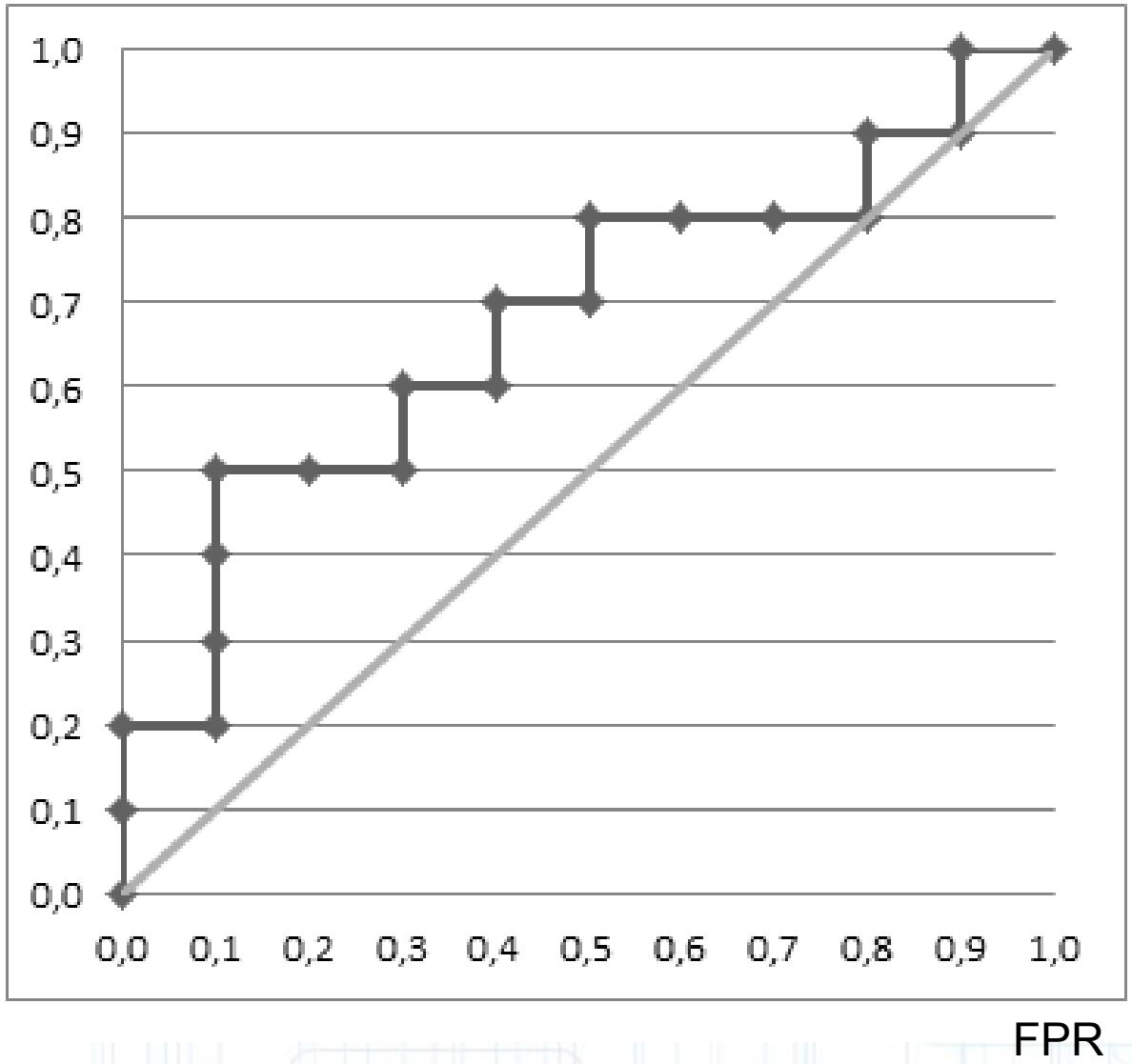
6: $\mathrm{FPR}_i := \mathrm{FPR}_{i-1} + \frac{1}{\ell_-}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1}$;
$\mathrm{AUC} := \mathrm{AUC} + \frac{1}{\ell_-}\mathrm{TPR}_i$;

7: **иначе** сместиться на один шаг вверх:

8: $\mathrm{FPR}_i := \mathrm{FPR}_{i-1}$; $\mathrm{TPR}_i := \mathrm{TPR}_{i-1} + \frac{1}{\ell_+}$;

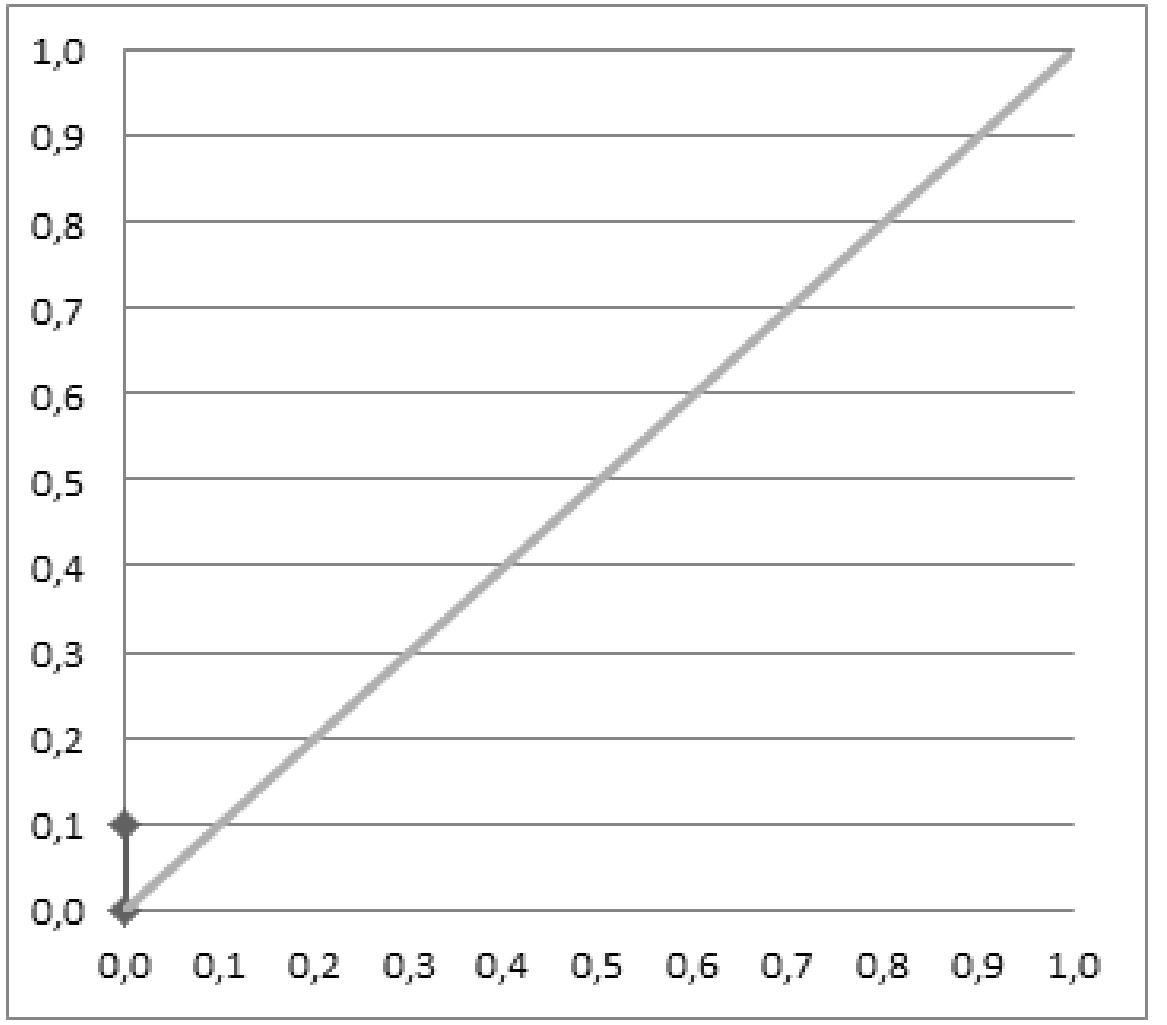
Пример

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1

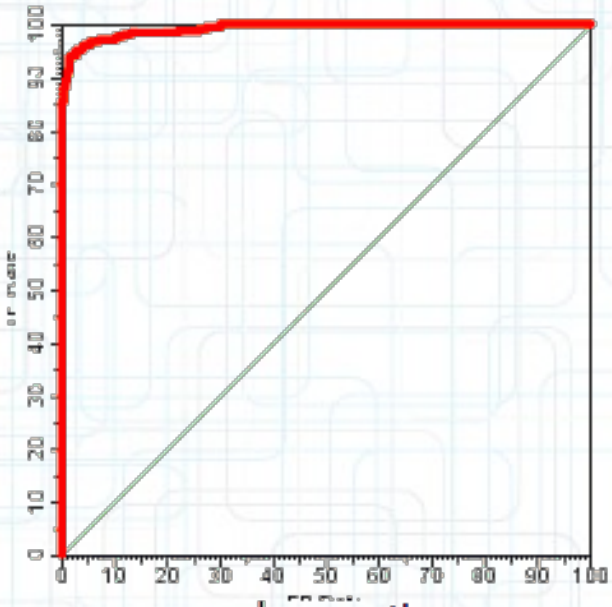


Пример

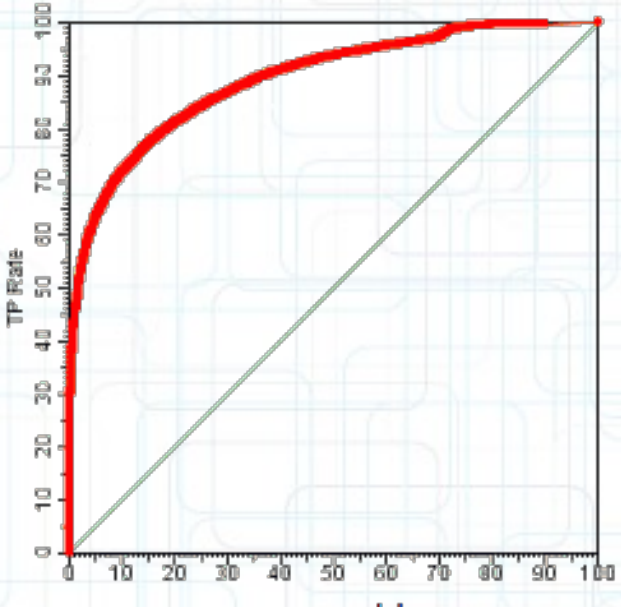
#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



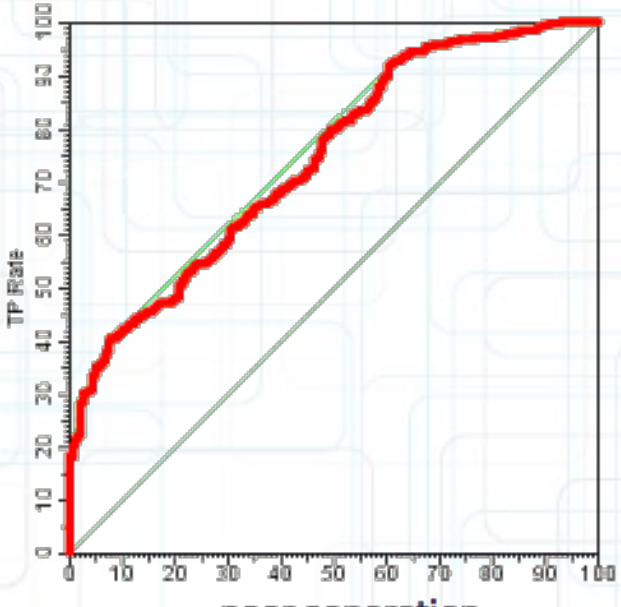
Еще примеры



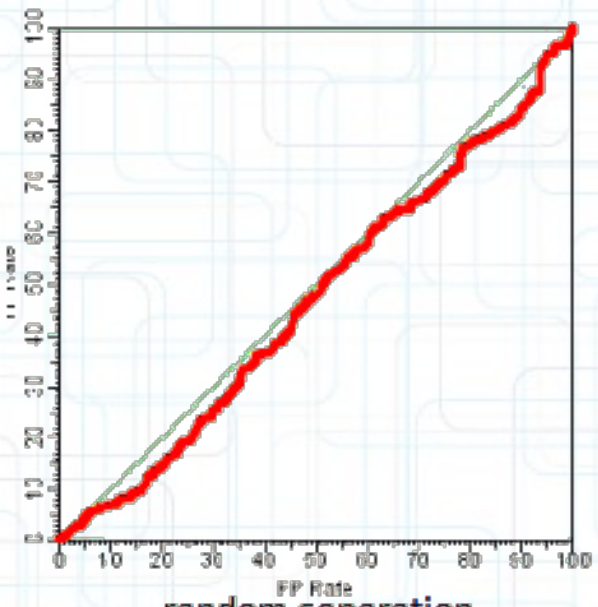
good separation



reasonable



poor separation



random separation

Градиентная максимизация AUC

Модель: $a(x_i, w, w_0) = \text{sign}\big(g(x_i, w) - w_0\big)$.

AUC — это доля правильно упорядоченных пар (x_i, x_j):

$$\text{AUC}(w) = \frac{1}{\ell_-}\sum_{i=1}^{\ell}[y_i = -1]\,\text{TPR}_i =$$

$$= \frac{1}{\ell_-\ell_+}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}[y_i < y_j]\,[g(x_i, w) < g(x_j, w)] \to \max_w$$

Явная максимизация аппроксимированного AUC:

$$\text{AUC}(w) \leqslant Q(w) = \sum_{i,j:\ y_i<y_j} \mathscr{L}\big(\underbrace{g(x_j, w) - g(x_i, w)}_{M_{ij}(w)}\big) \to \min_w,$$

где $\mathscr{L}(M)$ — гладкая убывающая функция отступа,
$M_{ij}(w)$ — новое понятие отступа для пар объектов.

Алгоритм стохастического градиента для AUC

Возьмём для простоты линейный классификатор:

$$g(x, w) = \langle x, w \rangle, \qquad M_{ij}(w) = \langle x_j - x_i, w \rangle.$$

Вход: выборка X^ℓ, темп обучения h, темп забывания λ

Выход: вектор весов w

1: инициализировать веса w_j, $j = 0, \ldots, n$;
2: инициализировать оценку: $\bar{Q} := \frac{1}{\ell_+ \ell_-} \sum_{i,j:\, y_i < y_j} \mathscr{L}(M_{ij}(w))$;
3: **повторять**
4: выбрать пару объектов (i, j): $y_i < y_j$, случайным образом;
5: вычислить потерю: $\varepsilon_{ij} := \mathscr{L}(M_{ij}(w))$;
6: сделать градиентный шаг: $w := w - h\mathscr{L}'(M_{ij}(w))(x_j - x_i)$;
7: оценить функционал: $\bar{Q} := (1 - \lambda)\bar{Q} + \lambda\varepsilon_{ij}$;
8: **пока** значение $\bar{Q}$ и/или веса w не сойдутся;

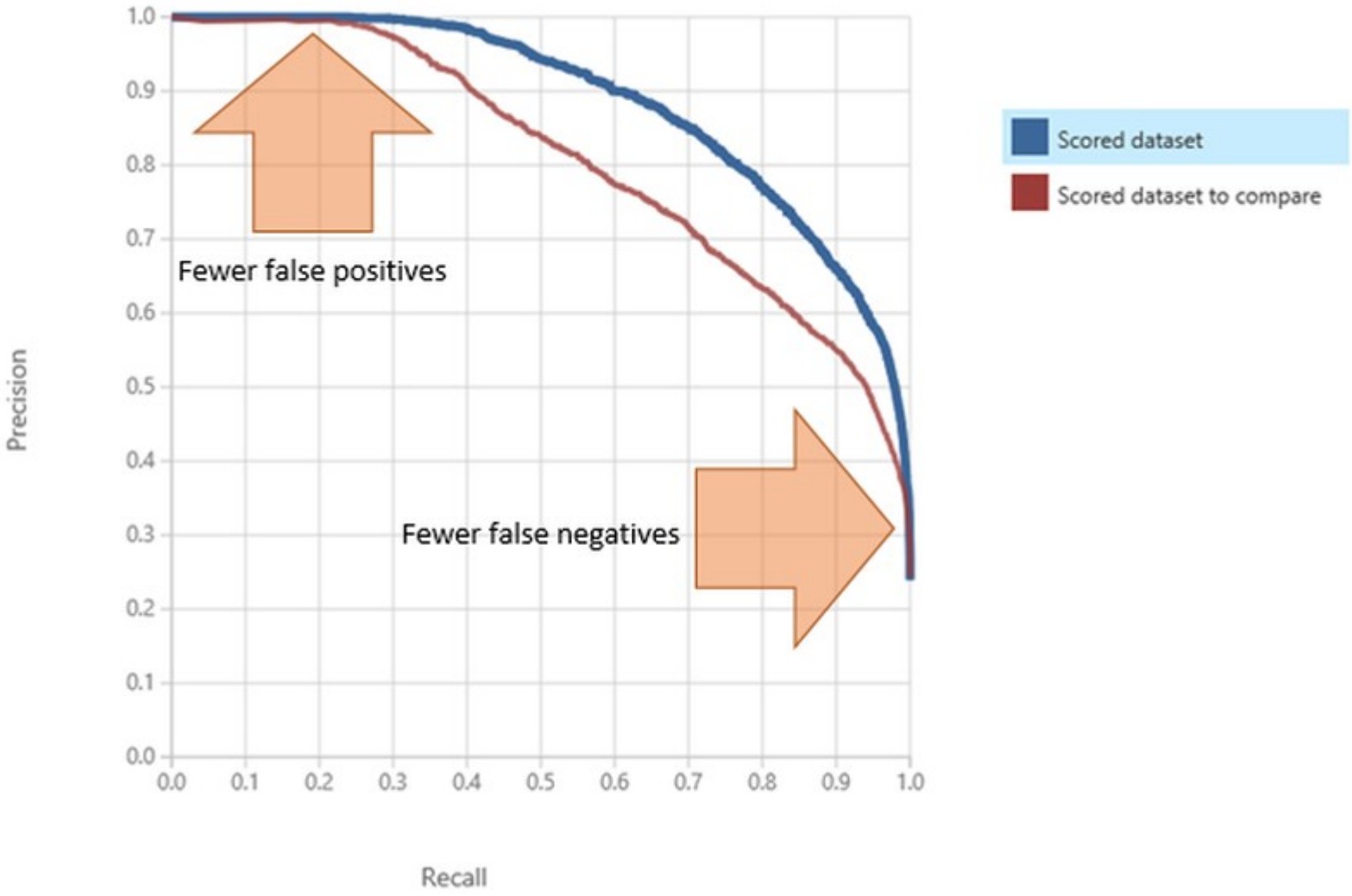
Точность (precision) и полнота (recall)

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

		Predicted	
		Positive (PP)	Negative (PN)
Actual	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

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

PR-кривая

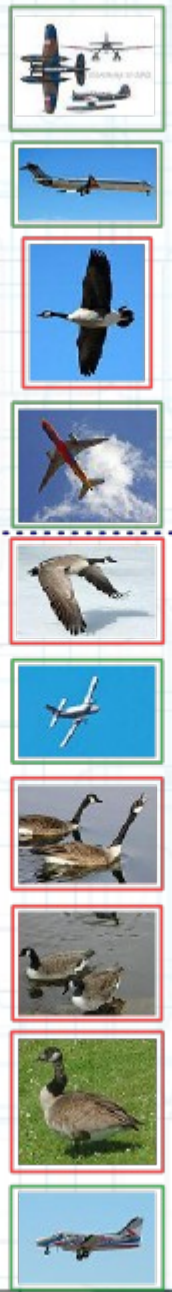


Средняя точность (AveP) – площадь под PR-кривой.

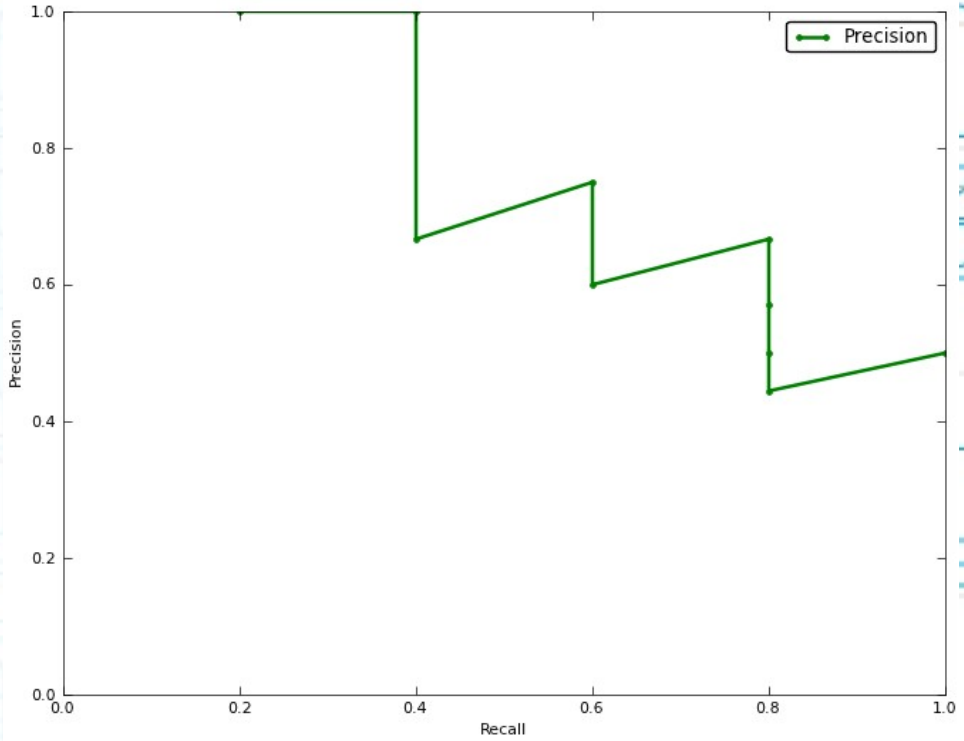
Как выглядит график для случайного порядка?

Пример расчета PR-кривой в задаче определения самолетов

Результат ранжирования



Порог	Precision	Recall
Top 1	100%	20%
Top 2	100%	40%
Top 3	66%	40%
Top 4	75%	60%
Top 5	60%	60%
Top 6	66%	80%
Top 7	57%	80%
Top 8	50%	80%
Top 9	44%	80%
Top 10	50%	100%



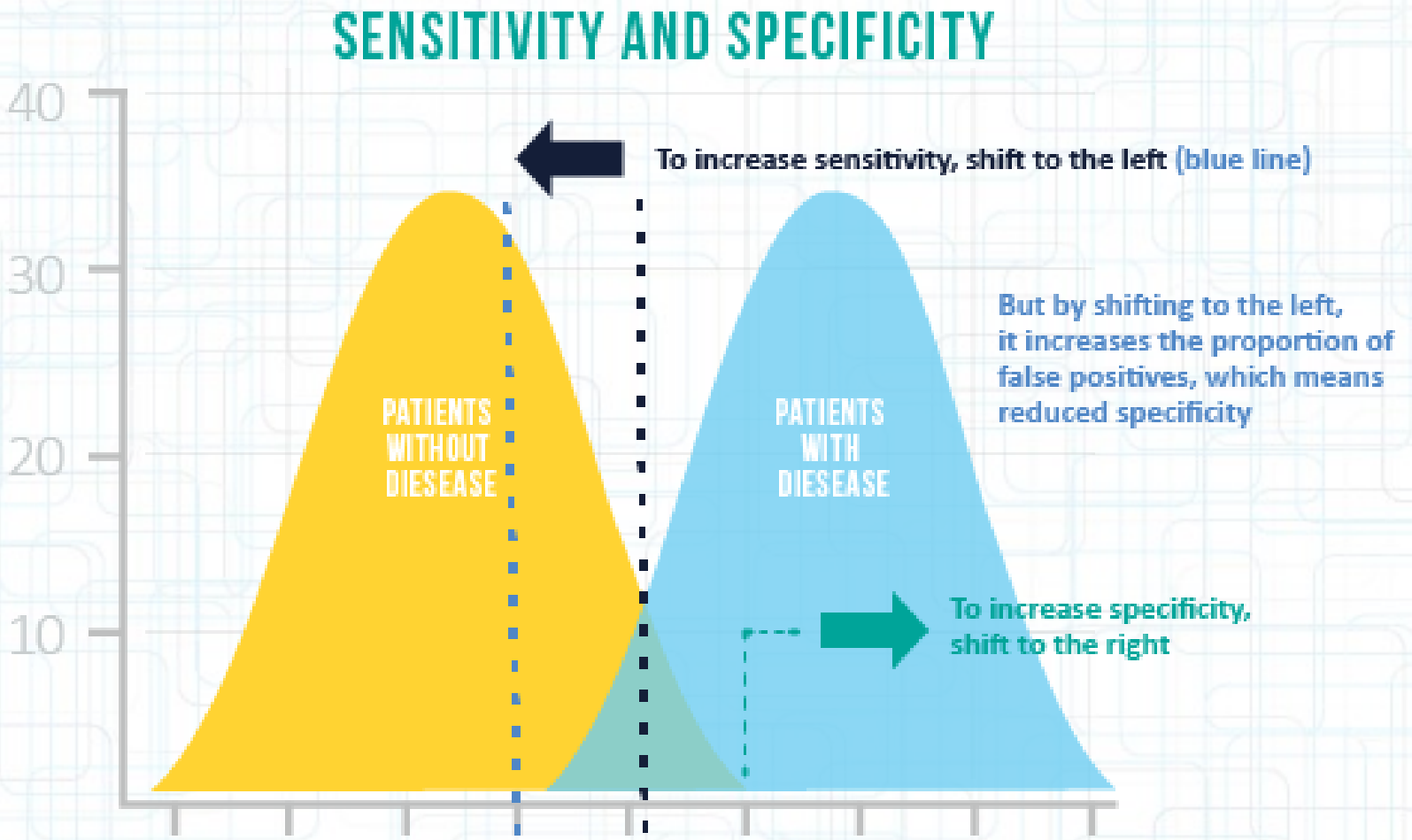
Чувствительность (sensitivity) и специфичность (specificity)

$$\text{Sensitivity} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Чувствительность = Полнота (recall)
Специфичность = 1-FPR, другое название: селективность.
Специфичность - это та же полнота, но для класса 0

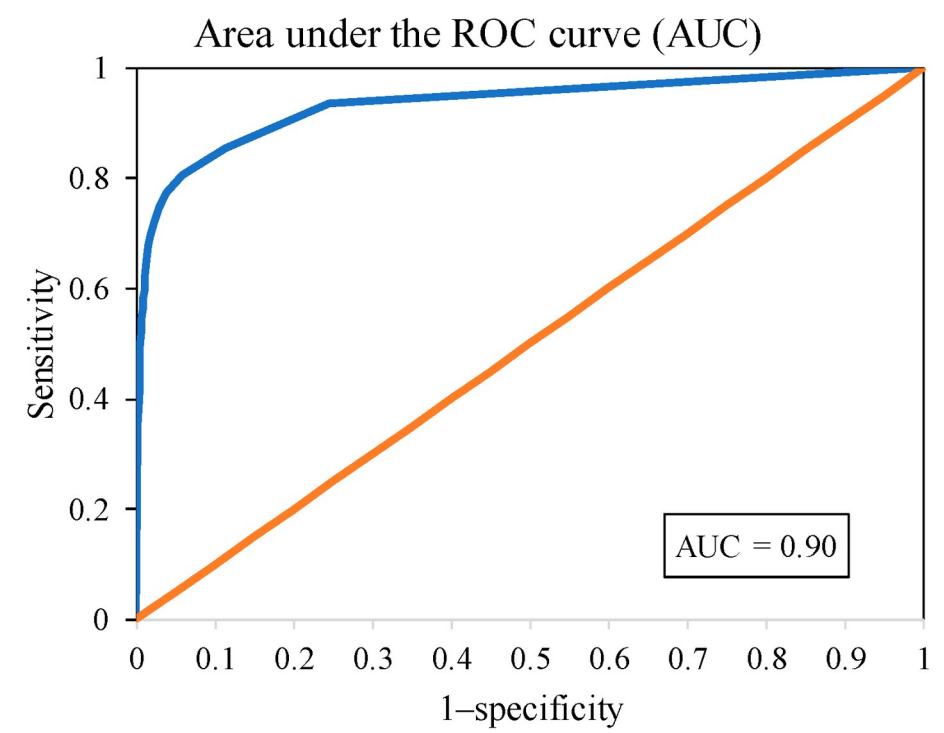
Порог задает баланс между полнотой для класса 1 и класса 0



По оси абсцисс отложены значения анализа (вероятности болезни)
По оси ординат – количество обследованных пациентов
Мы предсказываем болезнь, если вероятность больше порога

Чувствительность (sensitivity) и специфичность (specificity)

Так как Специфичность = 1-FPR, то ROC-кривая также отражает баланс между полнотой в классе 1 (sensitivity) и полнотой в классе 0 для разных значений порога

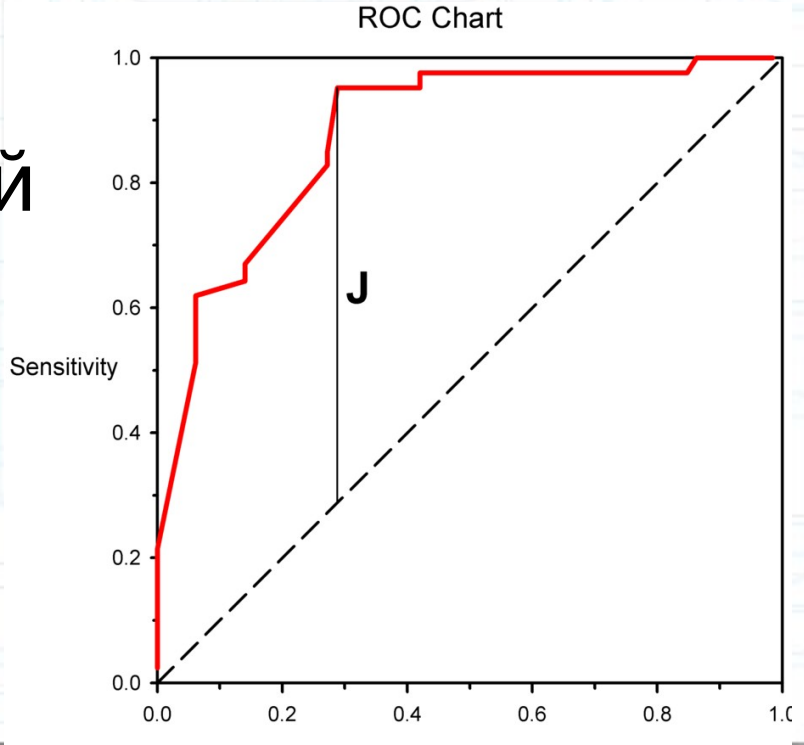


Проблемы метрик качества

- Популярная Accuracy является плохой оценкой для датасетов с несбалансированными классами (пример: 90% класса 1 и 10% класса 0, предсказываем всем 1)
- Точность (Precision) равна 1, если брать маленький порог в отсортированном ряду предсказаний
- Полнота (Recall) равна 1, если брать большой порог
- F1-мера = $2/(P^{-1}+R^{-1})$ – очень популярна, но не зависит от TN (True Negative)

Универсальные метрики

- Balanced accuracy – среднее значение между Sensitivity и Specificity (полнотой в классе 0 и 1). Меняется от 0.5 (random) до 1.
- Индекс Юдена = Sensitivity+Specificity-1. Почти то же самое, но меняется от 0 1.
- Равен высоте точки ROC над биссектрисой
- Реализация scikit-learn: balanced_accuracy_score



Вопросы для самоконтроля

- Сгенерируйте случайную выборку из 10 объектов с классами +1 и -1. Придумайте значения для (w,x_i) и вычислите предсказания $a(x)=sign(w,x_i)$. Нарисуйте ROC и PR-кривые, вычислите площадь под ROC-кривой.
- Вычислите Precision, Recall, Sencitivity, Specificity, Balanced accuracy, индекс Юдена для вышеуказанного классификатора
- Посмотрите внимательно на график PR-кривой на слайде с утками и самолетами. Найдите точку (0.4,0.66). Верно ли, что среди 40% объектов в верхней части таблицы находится 66% процентов самолетов?

Вопросы для самоконтроля

- Сгенерируйте случайную выборку из 10 объектов с классами +1 и -1 и двумя признаками: x_1, x_2. Выберите какую-нибудь функцию ошибок со слайда 14. Придумайте начальное значение для вектора $w=(w_1,w_2)$ и выполните один шаг метода стохастического градиента с темпом обучения h=0.1.