курс «Машинное обучение»

Функции ошибки / функционалы качества

Александр Дьяконов

14 декабря 2021 года

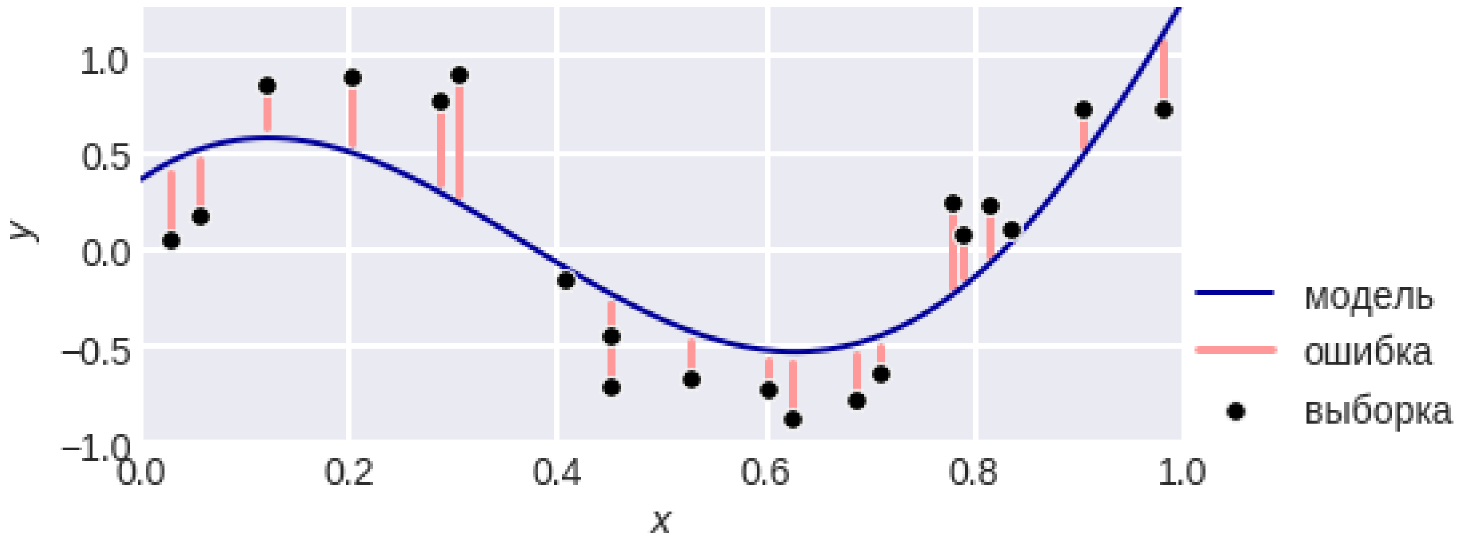
План

задача регрессии

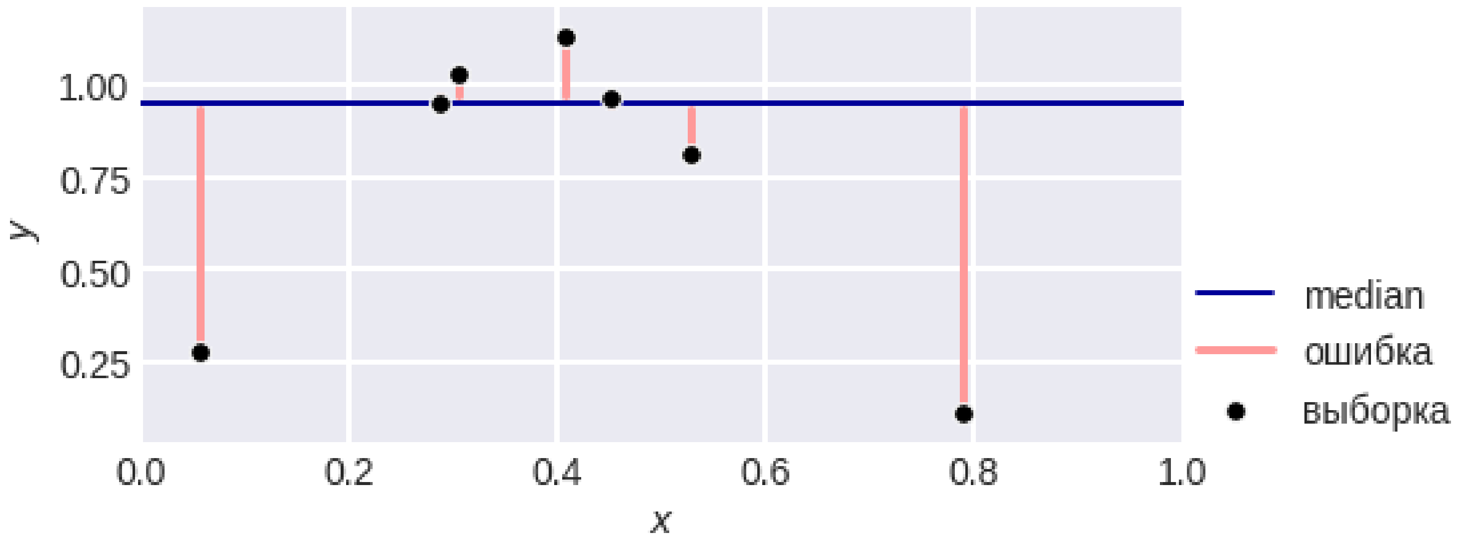
задача бинарной классификации
- **чёткая классификация**
- **скоринговые функции**

задача классификации с несколькими классами

Задача регрессии



Средний модуль отклонения – Mean Absolute Error (MAE), Mean Absolute Deviation (MAD)



$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |$$

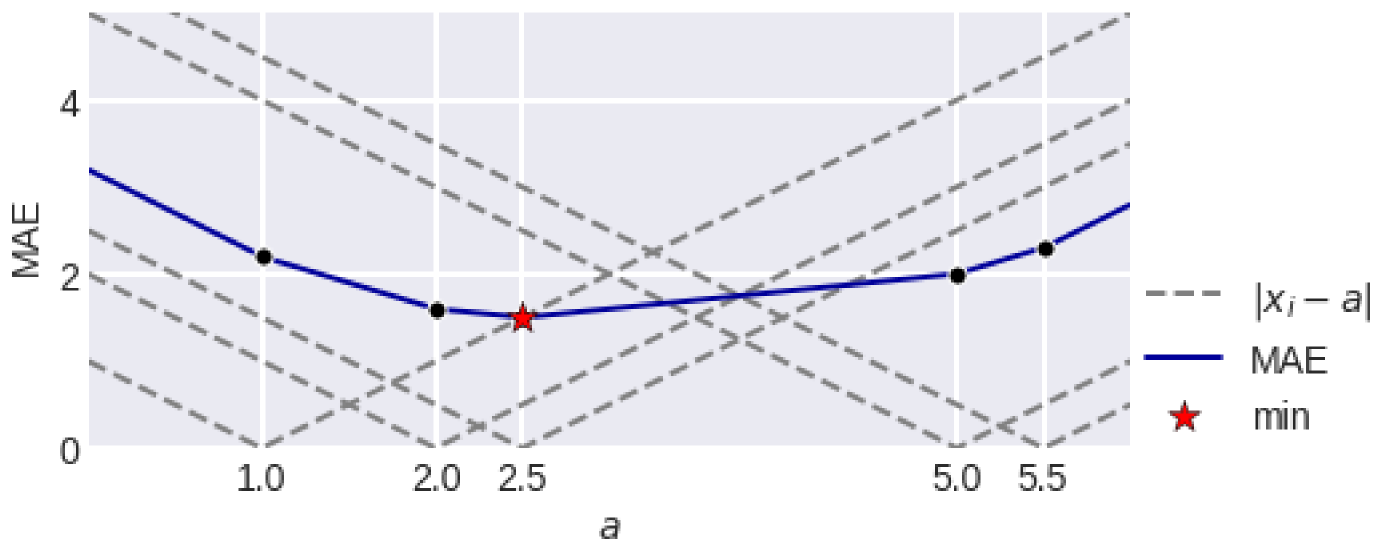
Напоминание:

$$\frac{1}{m}\sum_{i=1}^{m} | a - y_i | \to \min$$

$$a = \mathrm{median}(\{y_i\}_{i=1}^{m})$$

Это открывает смысл решений!

Средний модуль отклонения

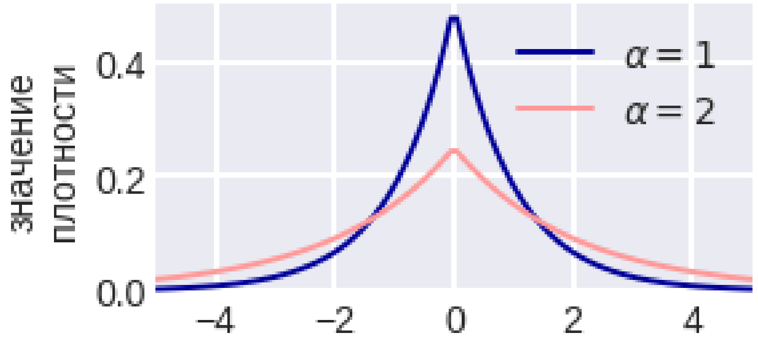


Откуда берётся МАЕ

$$y = a_w(x) + \varepsilon$$

w **– параметры алгоритма** $a_w(x)$

$$\varepsilon \sim \mathrm{laplace}(0, \alpha)$$



Для оценки параметров выписываем правдоподобие модели

$$p(y \mid x, w) = \frac{\alpha}{2} \exp\left[-\alpha \mid y - a_w(x) \mid\right]$$

Метод максимального правдоподобия:

$$\log L(w) = \log \prod_{i=1}^{m} p(y_i \mid x_i, w) =$$

$$= \sum_{i=1}^{m} \left[\log \frac{\alpha}{2} - \alpha \mid y_i - a_w(x_i) \mid \right] \to \max$$

Откуда берётся MAE

Получаем

$$\alpha \sum_{i=1}^{m} | y_i - a_w(x_i) | \to \min$$

т.е. задачу минимизации MAE!

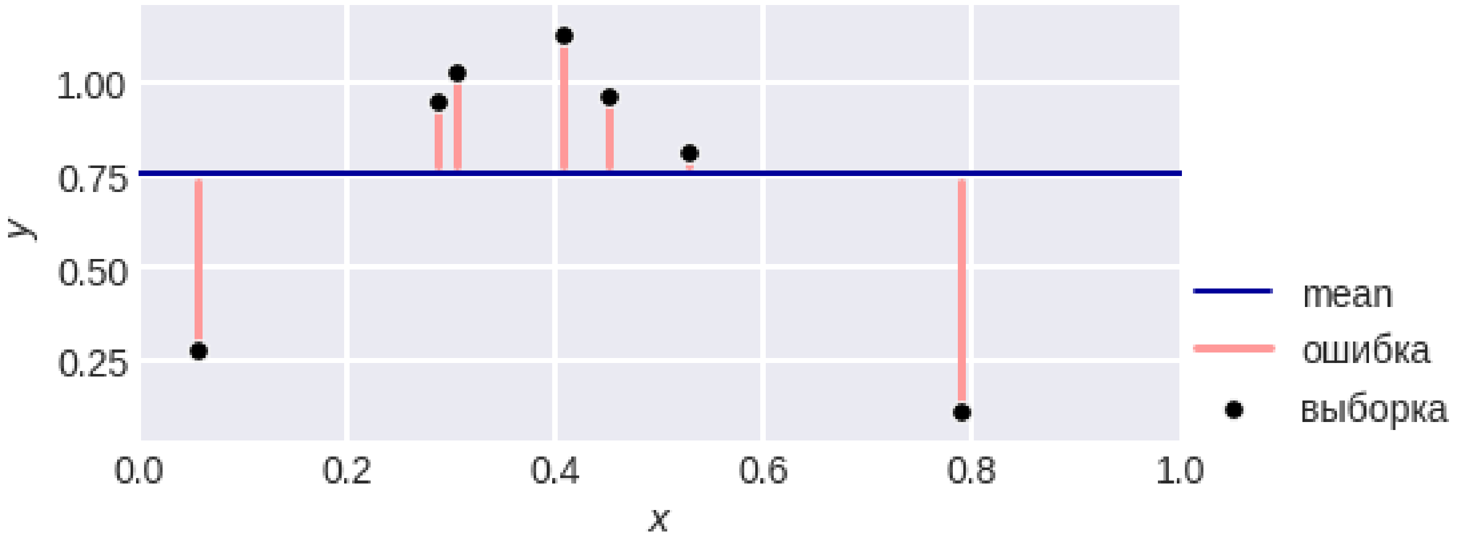
- **не зависит от природы модели**
- **зависит от распределения ошибок**

(почему Residual Plots)

Максимизация правдоподобия эквивалентна минимизации MAE!

Средний квадрат отклонения ~ Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |^2$$



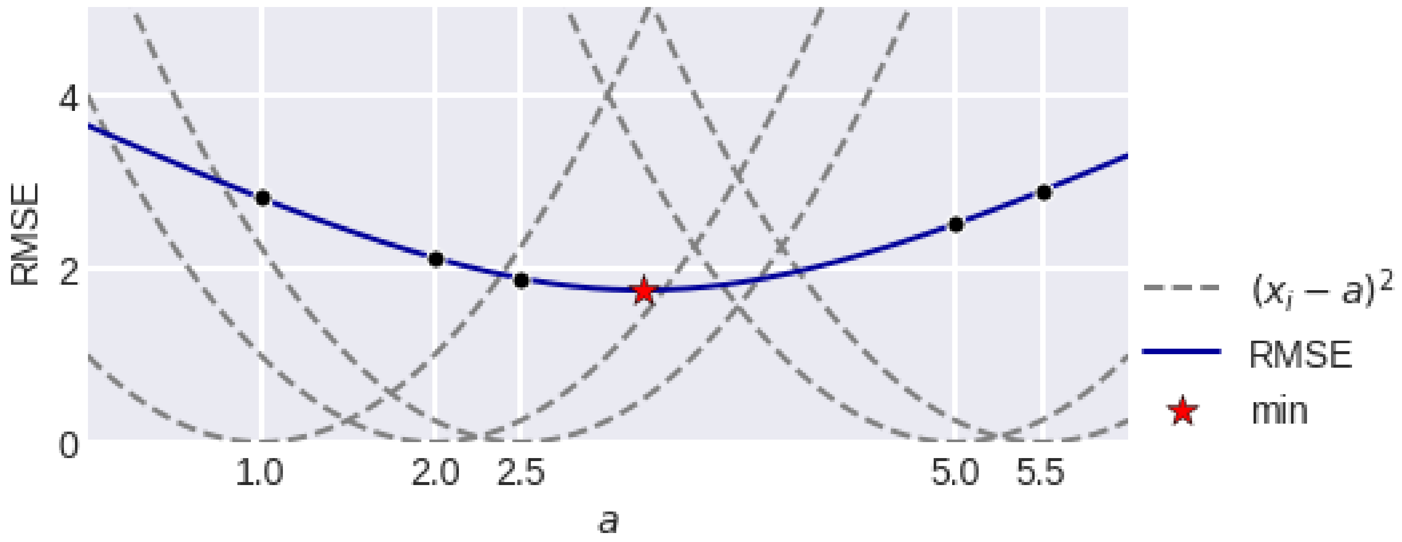
$$\frac{1}{m}\sum_{i=1}^{m} | a - y_i |^2 \to \min$$

$$a = \frac{1}{m}\sum_{i=1}^{m} y_i$$

Root Mean Squared Error (RMSE) / Root Mean Square Deviation (RMSD)

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |^2}$$

Средний квадрат отклонения ~ Mean Squared Error (MSE)



Нормированная версия: коэффициент детерминации R²
(Coefficient of Determination)

$$R^2 = 1 - \frac{\sum_{i=1}^{m} | a_i - y_i |^2}{\sum_{i=1}^{m} | \overline{y} - y_i |^2}$$

$$\overline{y} = \frac{1}{m} \sum_{i=1}^{m} y_i$$

В общем случае (в статистике) коэффициент детерминации:

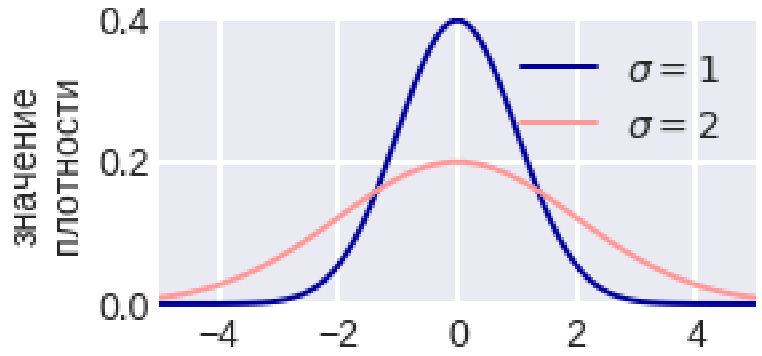
$$R^2 = 1 - \frac{\mathbf{D}(y \,|\, x)}{\mathbf{D}(y)}$$

Откуда берётся (R)MSE

$$y = a_w(x) + \varepsilon$$

w **– параметры алгоритма** $a_w(x)$

$$\varepsilon \sim \mathrm{norm}(0, \sigma^2)$$



Для оценки параметров выписываем правдоподобие модели

$$p(y \mid x, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y - a_w(x))^2}{2\sigma^2}\right]$$

Метод максимального правдоподобия:

$$\log L(w) = \log \prod_{i=1}^{m} p(y_i \mid x_i, w) =$$

$$= \sum_{i=1}^{m} \left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_i - a_w(x_i))^2}{2\sigma^2}\right] \to \max$$

Откуда берётся (R)MSE

Получаем

$$\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - a_w(x_i))^2 \to \min$$

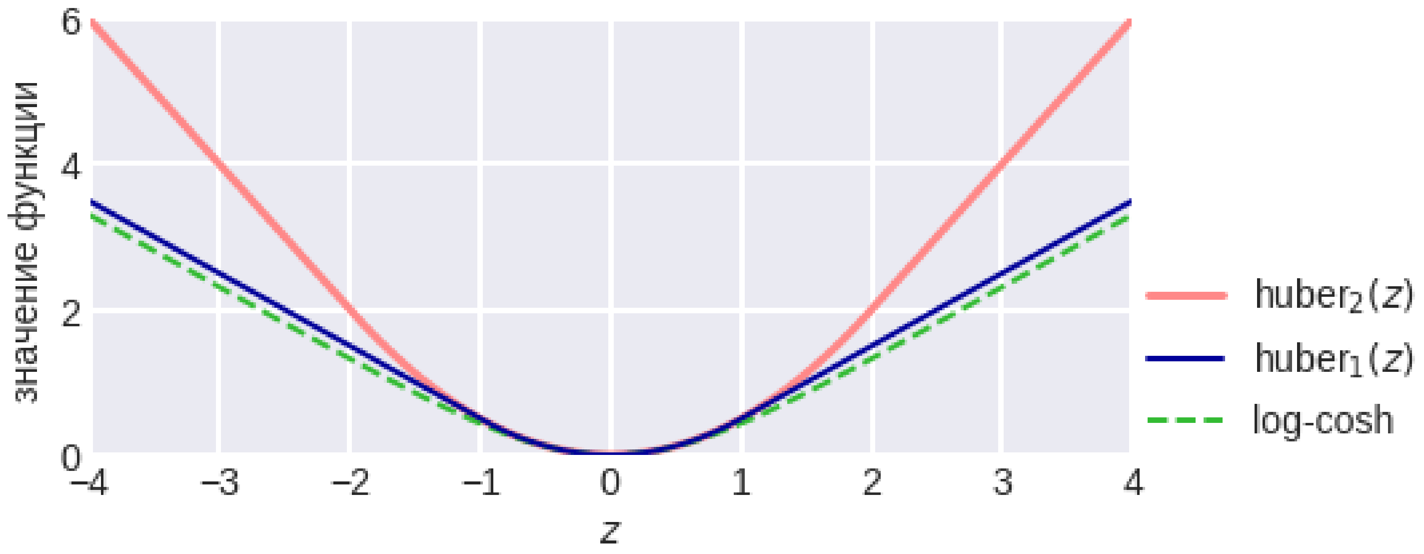
т.е. задачу минимизации MSE!

- **не зависит от природы модели**
- **зависит от распределения ошибок**

(почему Residual Plots)

Максимизация правдоподобия эквивалентна минимизации среднеквадратичной ошибки!

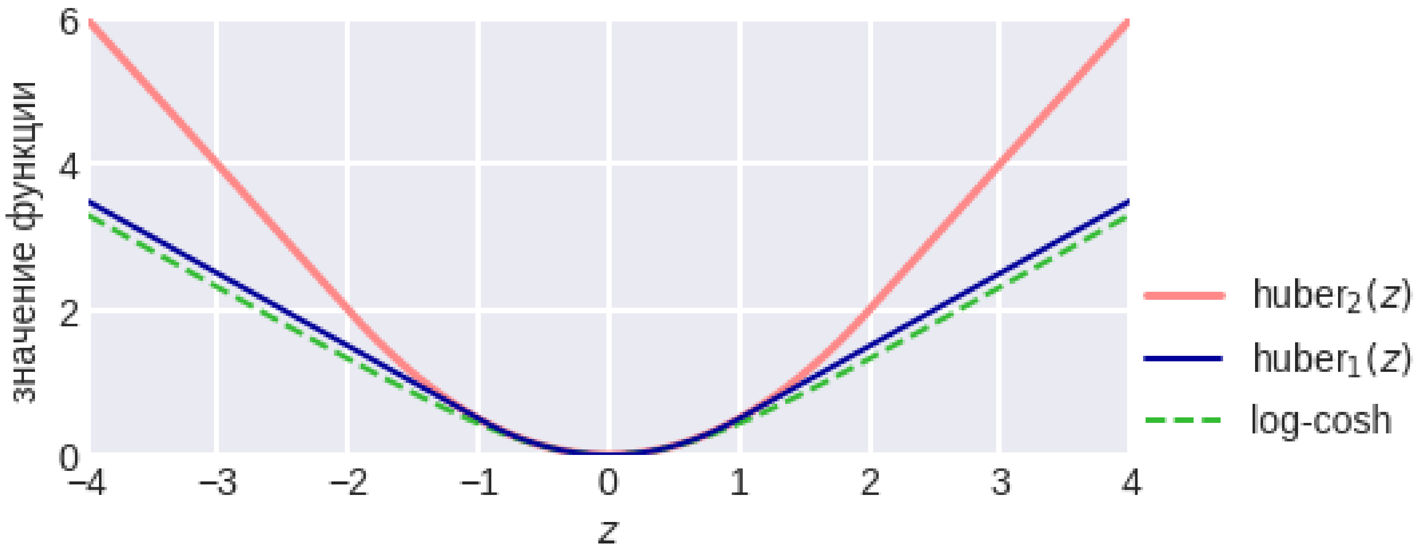
Функция Хьюбера и logcosh



$$\mathrm{huber}_{\delta}(z)=\begin{cases}\dfrac{1}{2}z^2, & |z|\le\delta,\\ \delta\left(|z|-\dfrac{1}{2}\delta\right), & |z|>\delta.\end{cases}$$

Как только что вывели:
когда отклонение мало – ошибка квадратичная
когда велико (в т.ч. выбросы) – линейная

Функция Хьюбера и logcosh

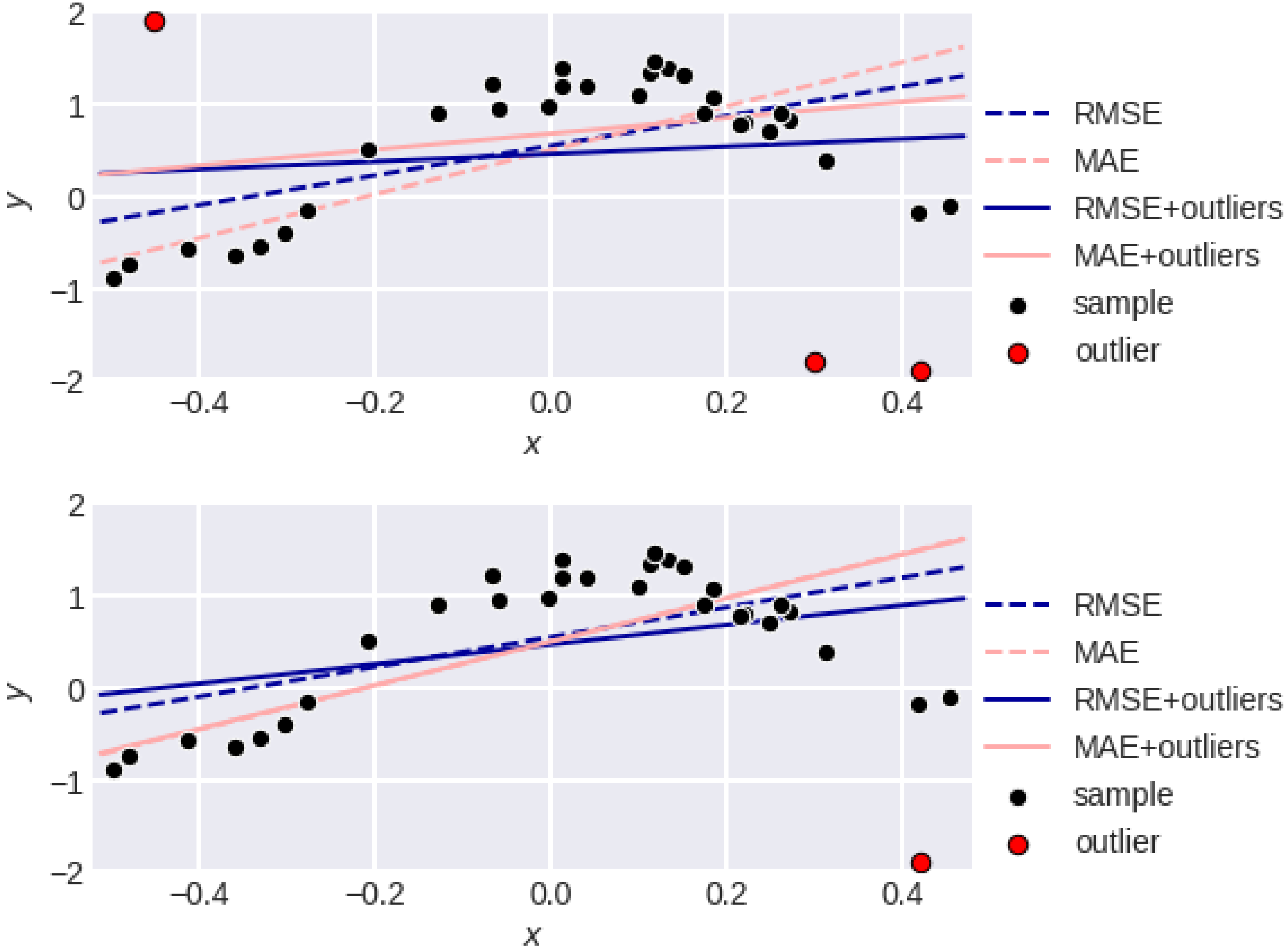


$$\mathrm{logcosh} = \log\left(\frac{\exp(z)+\exp(-z)}{2}\right)$$

непараметрическая,
но используется редко.

Различия MSE и MAE

Устойчивость к выбросам…



Symmetric mean absolute percentage error (SMAPE or sMAPE)

$$\mathrm{SMAPE} = \frac{2}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{y_i + a_i} = 100\% \cdot \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{(y_i + a_i)/2}$$

Mean Absolute Percent Error (MAPE)

$$\mathrm{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{|y_i|}$$

Метрики в регрессии: минутка кода

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import median_absolute_error
from sklearn.metrics import explained_variance_score
# R^2
print (r2_score(y, a),
       1 - np.mean((y - a) ** 2) / np.mean((y - np.mean(y)) ** 2))
# MAE
print (mean_absolute_error(y, a),
       np.mean(np.abs(y - a)))
# MSE
print (mean_squared_error(y, a),
       np.mean((y - a) ** 2))
# MSLp1E
print (mean_squared_log_error(y, a),
       np.mean((np.log1p(y) - np.log1p(a)) ** 2))
# MedAE
print (median_absolute_error(y, a),
       np.median(np.abs(y - a)))
```

Итоги

Функции ошибки имеют вероятностное обоснование

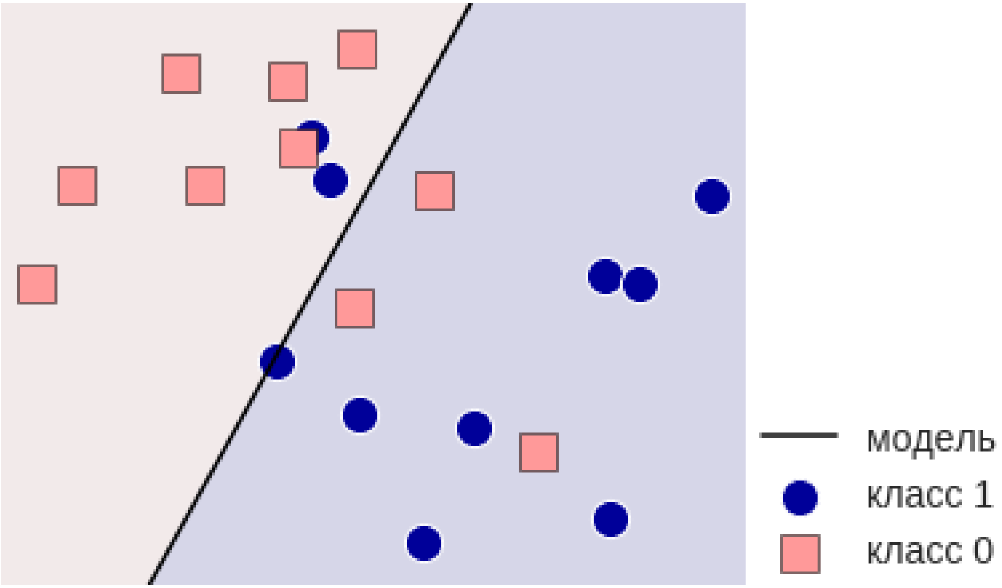
(через правдоподобие)

средний модуль отклонения MAE (MAD)

средний квадрат отклонения MSE

+ RMSE, коэффициент детерминации R2, функция Хьюбера, Logcosh

Задача классификации



сначала – чёткая классификация

«Confusion Matrix» – матрица ошибок / несоответствий

ответы

	y	a
0	1	1
1	1	1
2	1	2
3	2	1
4	2	3
5	3	2
6	3	3
7	3	3
8	1	2
9	2	2

матрица ошибок

a y	1	2	3
1	2	2	0
2	1	1	1
3	0	1	2

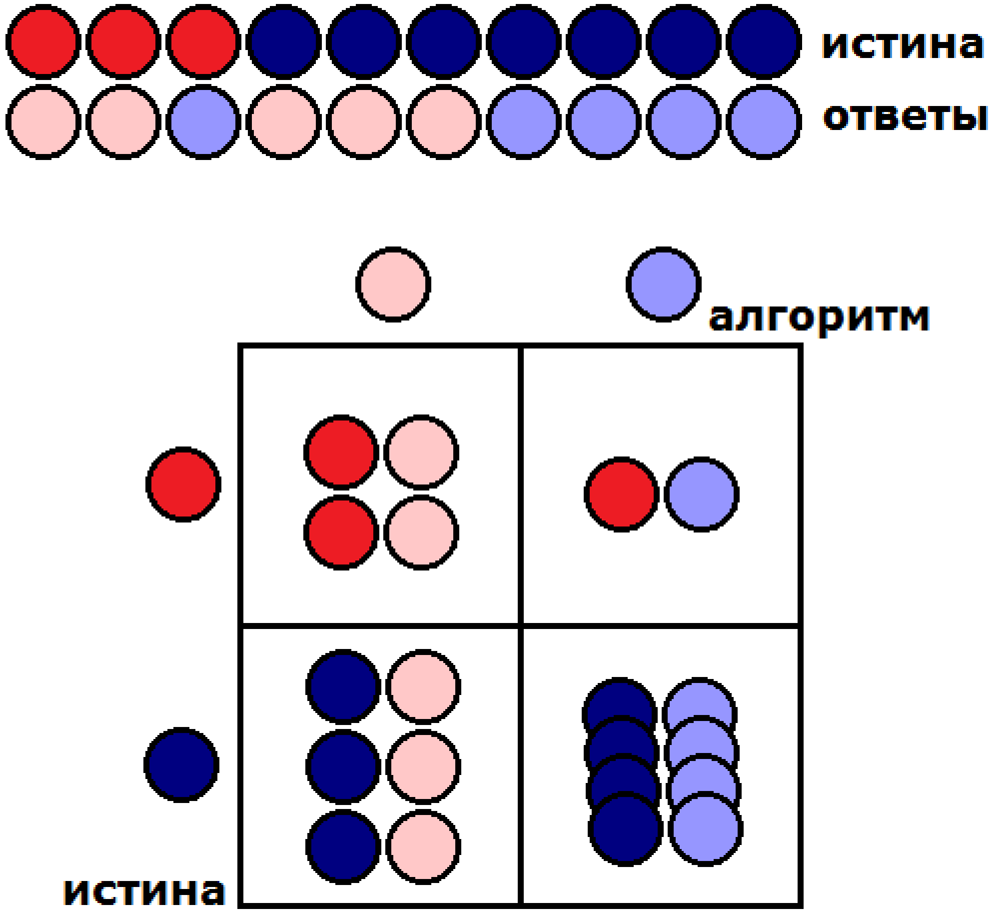
Для классов $\{1, 2, \ldots, l\}$

$$N = \| m_{ij} \|_{l \times l}$$

$$m_{ij} = \sum_{t=1}^{m} I[a_t = i] I[y_t = j]$$

```
from sklearn.metrics import confusion_matrix
n = confusion_matrix(df.y, df.a) # 1й способ
n = pd.crosstab(df.y, df.a) # 2й способ
```

«Confusion Matrix» в задаче классификации с двумя классами

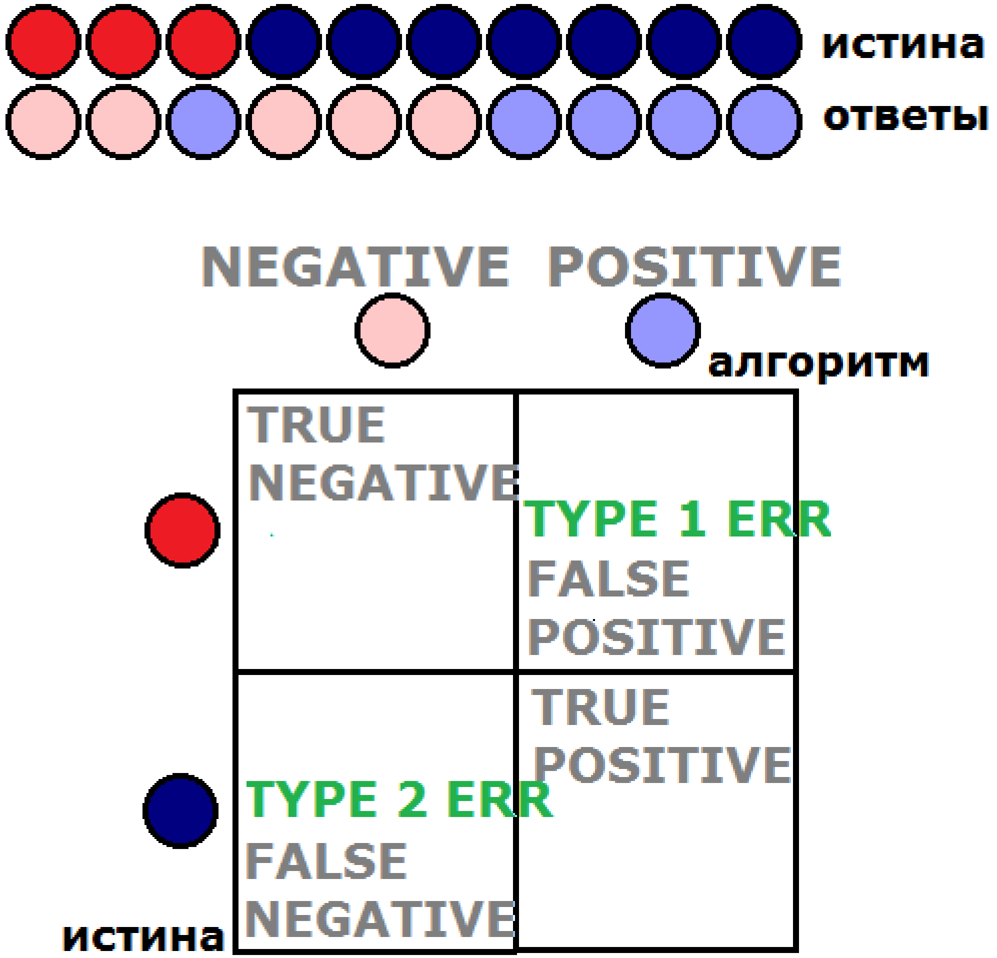


	$a=0$	$a=1$
$y=0$	13599	2600
$y=1$	898	903

в scikit-learn-е такая ориентация!
Иногда: наоборот!

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, a_test)
```

Задача классификации с двумя классами




```
tn, fp, fn, tp = confusion_matrix(y, a).ravel() # вычисление tn, ...
```

Как запомнить названия ошибок

1 рода – не учил, но сдал (= знает по мнению экзаменатора)
2 рода – учил, но не сдал (= не знает по мнению экзаменатора)

Ошибка 1 рода

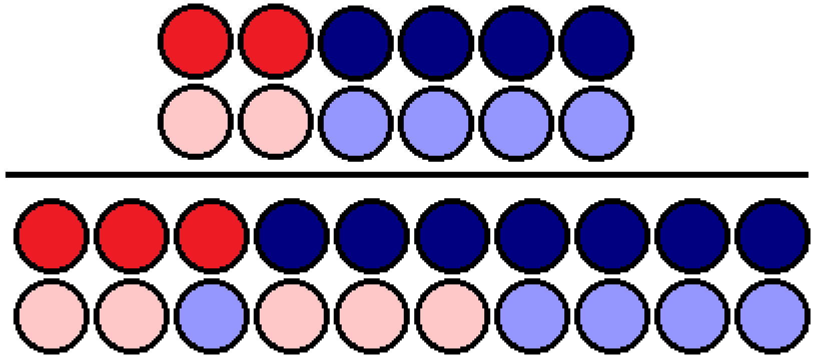
$\mathrm{FP}/\,m$

Ошибка 2 рода

$\mathrm{FN}/\,m$

Точность Accuracy

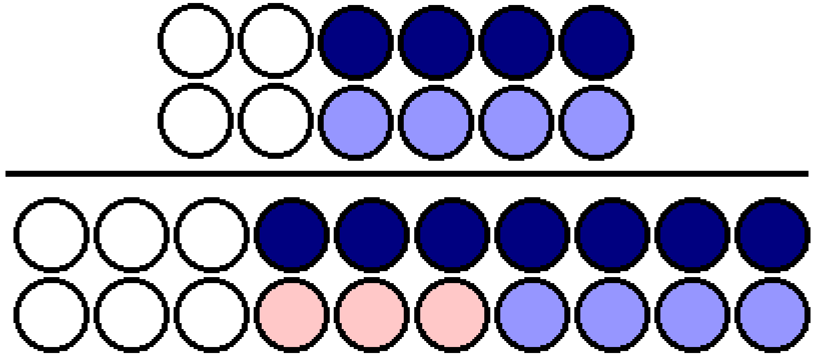
	$a=0$	$a=1$
$y=0$	$\boxed{13599}$	2600
$y=1$	898	$\boxed{903}$

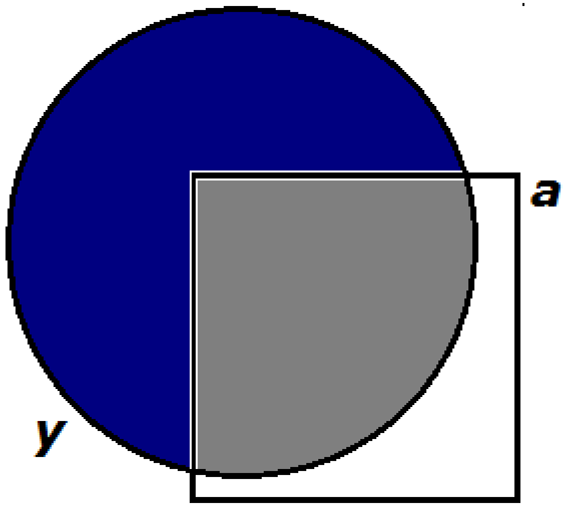


$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FN} + \text{TP} + \text{FP}}$$

Полнота (Sensitivity, True Positive Rate, Recall, Hit Rate)

	$a=0$	$a=1$
$y=0$	13599	2600
$y=1$	898	$\boxed{903}$



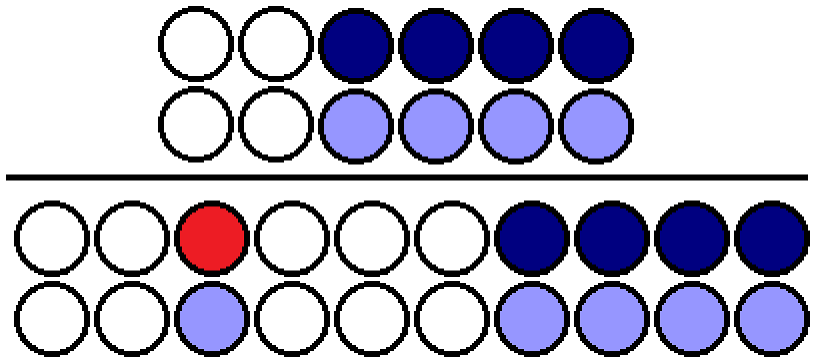


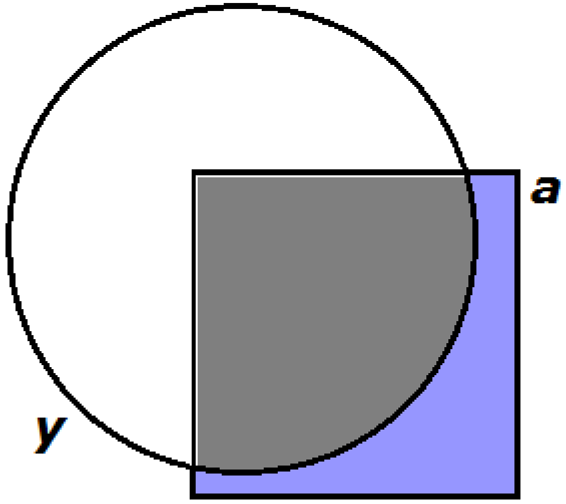
$$\mathrm{TPR} = R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

какой процент объектов положительного класса
мы правильно классифицировали

Точность (Precision, Positive Predictive Value)

	$a=0$	$a=1$
$y=0$	13599	2600
$y=1$	898	$\boxed{903}$





$$\mathrm{PPV} = P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

какой процент положительных объектов (т.е. тех, что мы считаем положительными) правильно классифицирован

False Positive Rate (FPR, fall-out, false alarm rate)

	$a=0$	$a=1$
$y=0$	13599	$\boxed{2600}$
$y=1$	898	903

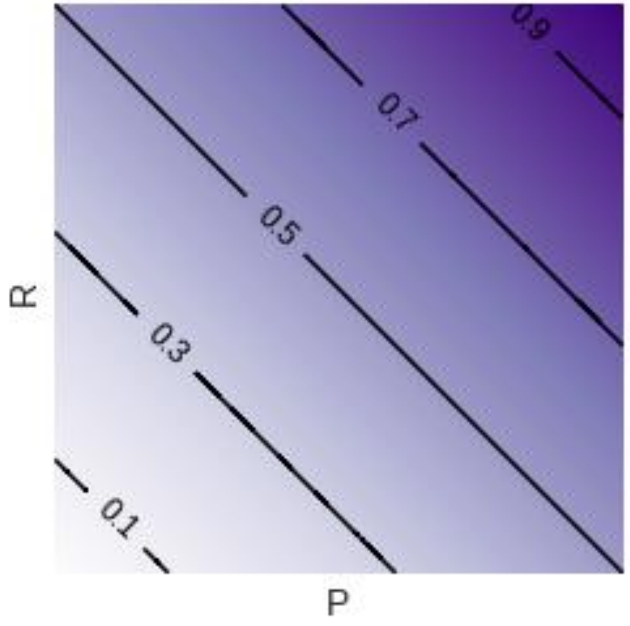
$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}=1-\mathrm{TNR}=1-\mathrm{Specificity}$$

доля объектов негативного класса, которых мы ошибочно отнесли к положительному

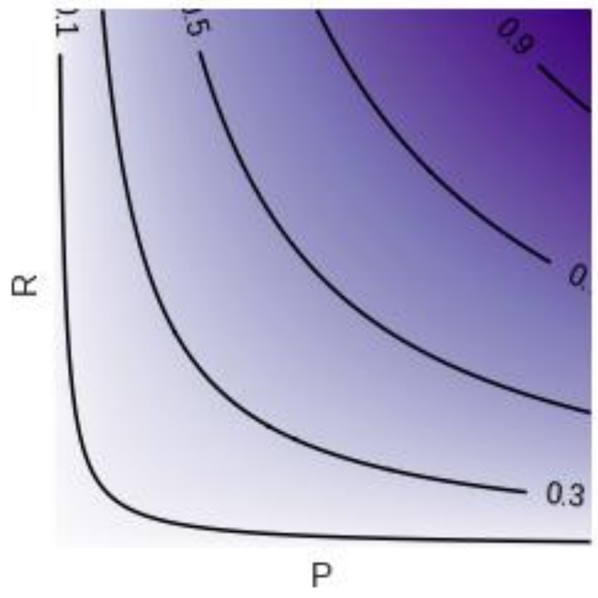
F_1 score

$$\frac{2}{\frac{1}{P}+\frac{1}{R}}=\frac{2}{\frac{1}{TP/(TP+FP)}+\frac{1}{TP/(TP+FN)}}=\frac{2TP}{2TP+FP+FN}$$

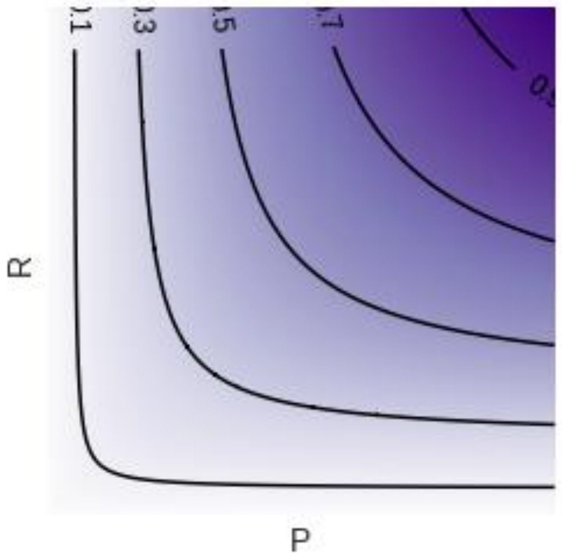
Почему используется F-мера



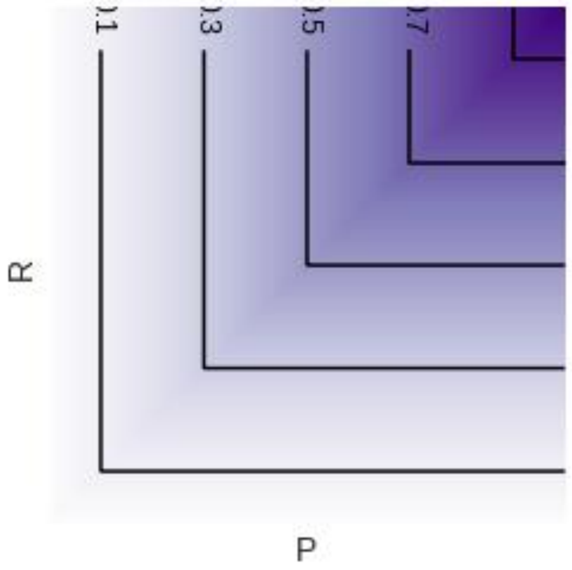
$(P+R)/2$



$\sqrt{P \cdot R}$



$2/(1/P+1/R)$



$\min(P, R)$

Задача бинарной классификации

Теперь выдаём оценку принадлежности к классу 1

$$y \in \{0, 1\}$$

$$a \in [0, 1]$$

кроме меток {0, 1} возможны промежуточные значения

Log Loss

В задаче классификации с двумя непересекающимися классами (0, 1), когда ответ вероятность (?) принадлежности к классу 1

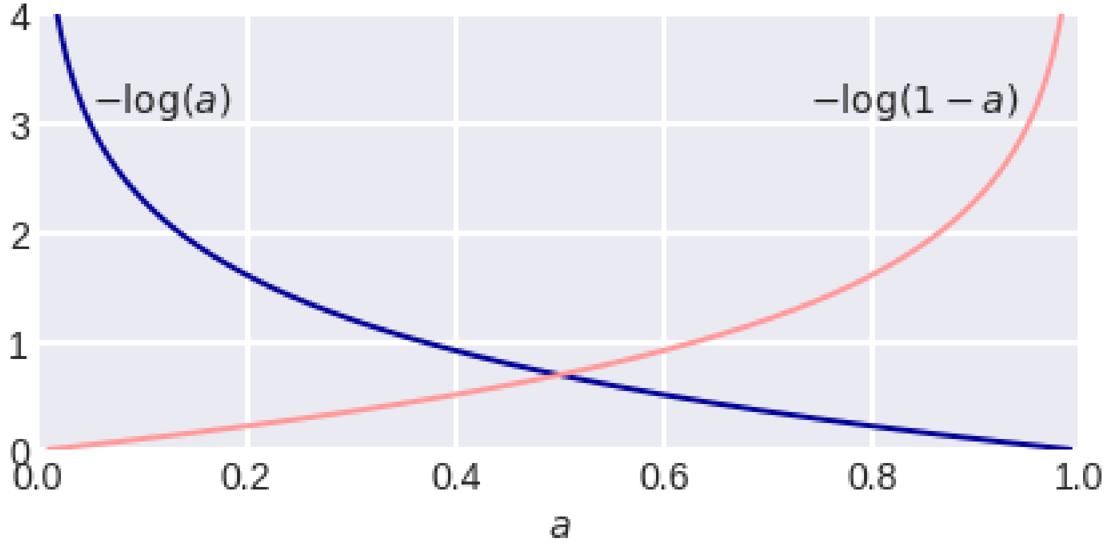
$$\mathrm{logloss} = -\frac{1}{m}\sum_{i=1}^{m}(y_i \log a_i + (1-y_i)\log(1-a_i))$$

На что похоже?

Раздельная форма понятнее…

$$-\begin{cases} \log a_i, & y_i = 1, \\ \log(1-a_i), & y_i = 0. \end{cases}$$

Нельзя ошибаться!



Откуда берётся Log Loss

Обучающая выборка ~ реализация обобщённой схемы Бернулли:

для x_i генерируем

$$y_i = \left[\begin{array}{ll} 1, & p_i, \\ 0, & 1-p_i. \end{array}\right.$$

Пусть наша модель генерирует эти вероятности!

$$a_i = a(x_i \mid w)$$

Правдоподобие:

$$p(y \mid X, w) = \prod_i p(y_i \mid x_i, w) = \prod_i a_i^{y_i}(1-a_i)^{1-y_i} \to \max$$

Откуда берётся Log Loss

Максимизация правдоподобия эквивалентна

$$\sum_i (-y_i \log a_i - (1 - y_i)\log(1 - a_i)) \to \min$$

Логична ровно настолько, насколько MSE в задаче регрессии (тоже выводится из ММП)

Названия

- **логистическая функция ошибки**
- **«логлосс»**
- **перекрёстная энтропия (кросс-энтропия)**

Интерпретация константного решения

Посчитаем матожидание ошибки –

у нас один (*i*-й) объект, который с вероятностью p принадлежит классу 1.

$$-p\log(a_i)-(1-p)\log(1-a_i)$$

Минимизируем это выражение:

$$\frac{p}{a_i}-\frac{1-p}{1-a_i}=0$$

$$a_i=p$$

О чудо! Так и должно быть, но не всегда бывает…

Вот почему используют log_loss!

Интерпретация константного решения

Если подставить оптимальное значение $a_i = p$ в

$$-p\log(a_i) - (1-p)\log(1-a_i)$$

получаем энтропию:

$$-p\log(p) - (1-p)\log(1-p)$$

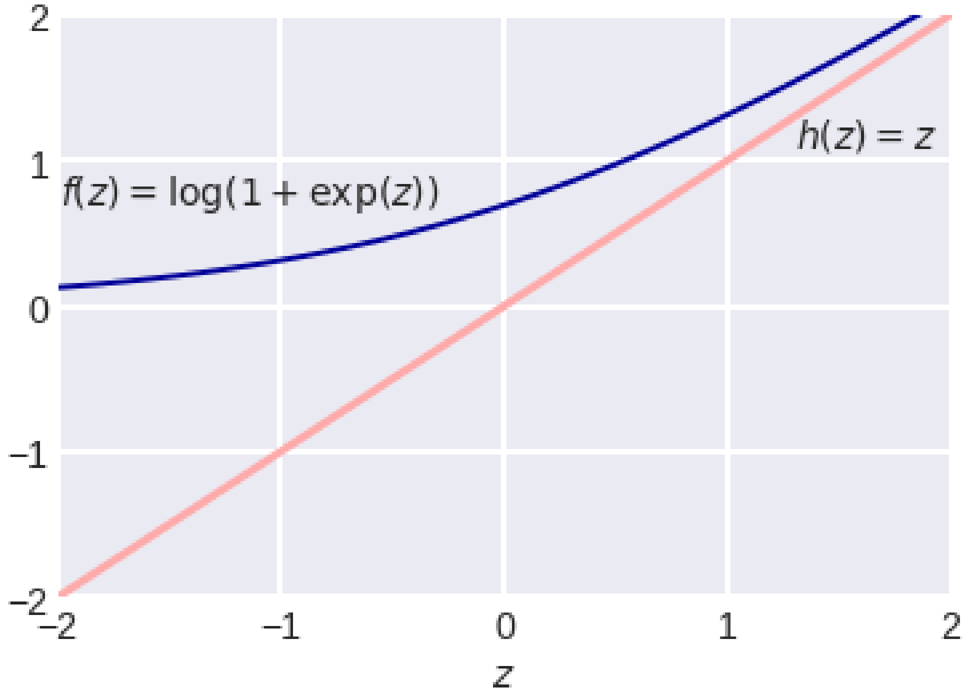
Вот почему используют энтропийный критерий расщепления!

он минимизирует logloss!

Другая форма функционала

Подставим выражение для сигмоиды, сделаем переобозначение: метки классов теперь –1 и +1, тогда

$$\mathrm{logloss}(a, y) = \log(1 + \exp(-y \cdot w^{\mathrm{T}} x))$$



LogReg

$$\sum_i \log(1 + \exp(-y_i \cdot w^{\mathrm{T}} x_i)) \to \min$$

SVM – Hinge Loss

$$\sum_i \max[1 - y_i w^{\mathrm{T}} x, 0] + \alpha w^{\mathrm{T}} w \to \min$$

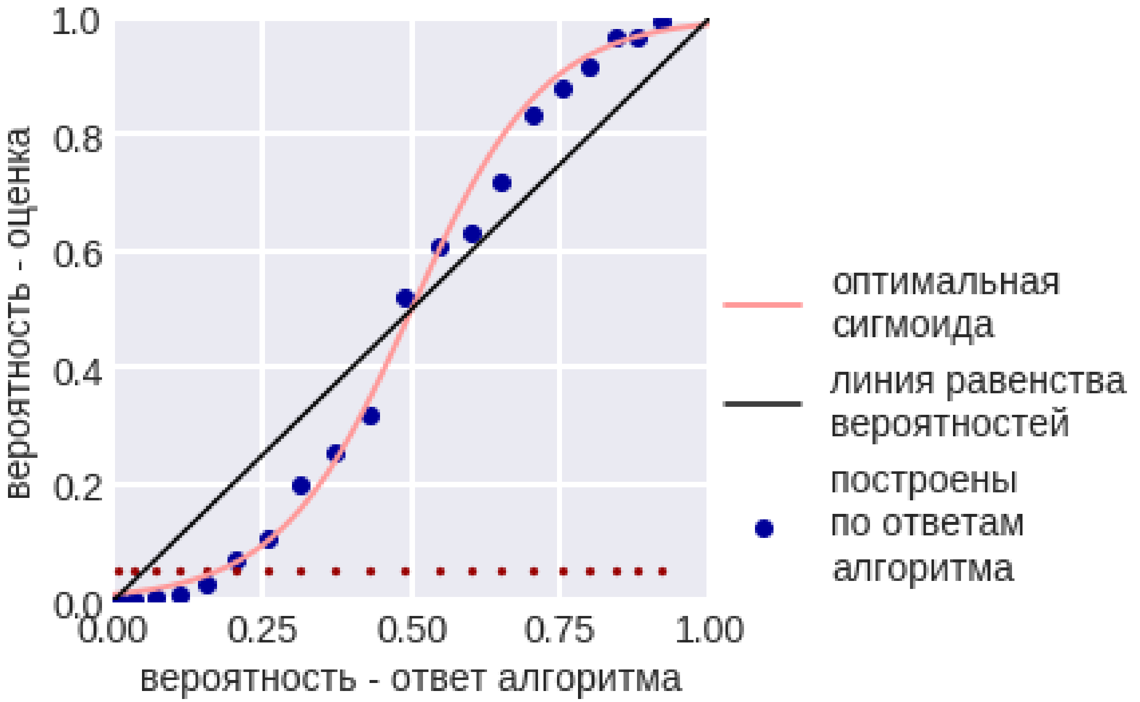
RVM

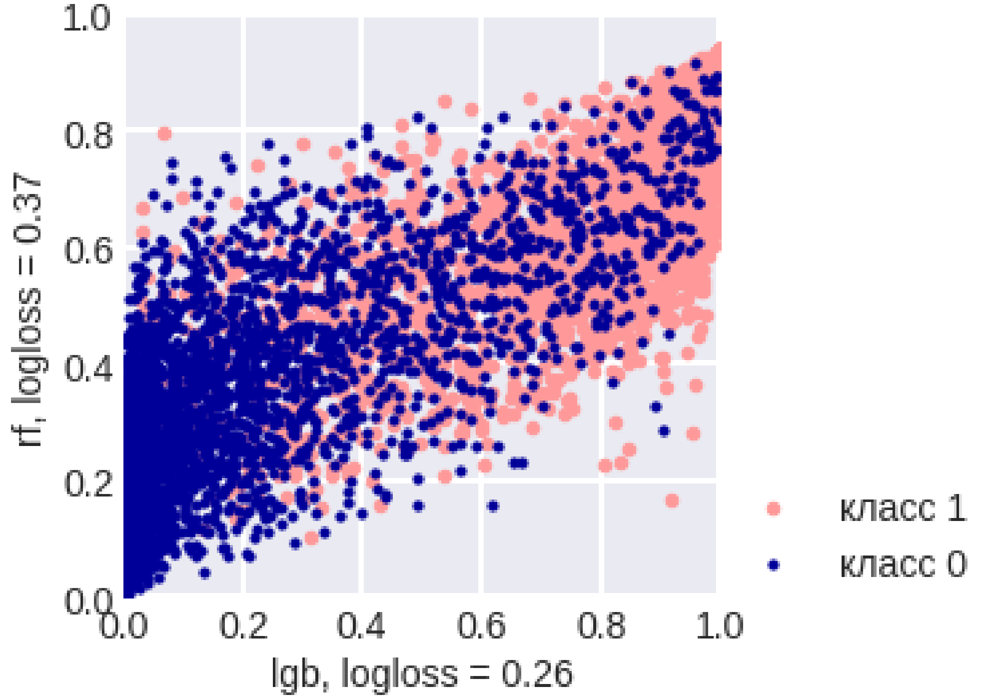
$$\sum_i \log(1 + \exp(-y_i w^{\mathrm{T}} x)) + w^{\mathrm{T}} \mathrm{diag}(\alpha) w \to \min$$

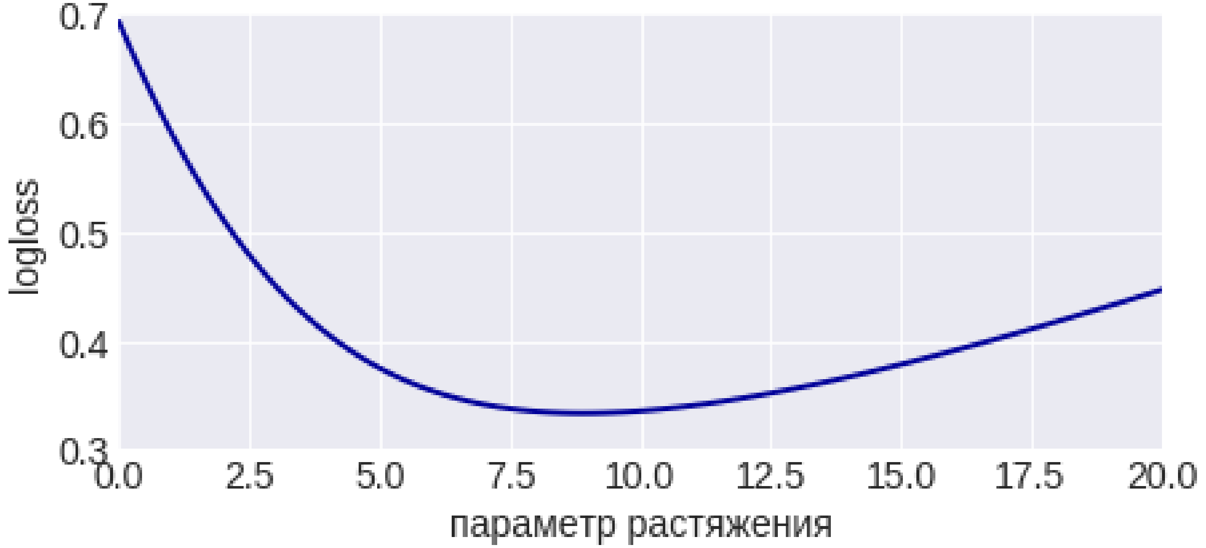
Настройка на Logloss – методы калибровки

калибровка Платта (Platt calibration)
– для SVM

$$a(x) = \mathrm{sigmoid}(\alpha \cdot r(x) + \beta)$$





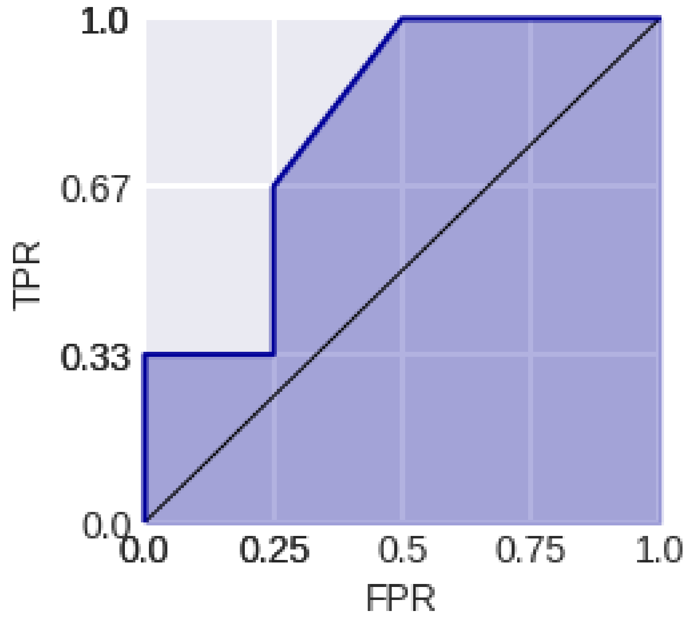


ROC и AUC ROC

ROC = receiver operating characteristic
Функционал зависит не от конкретных значений, а от их порядка

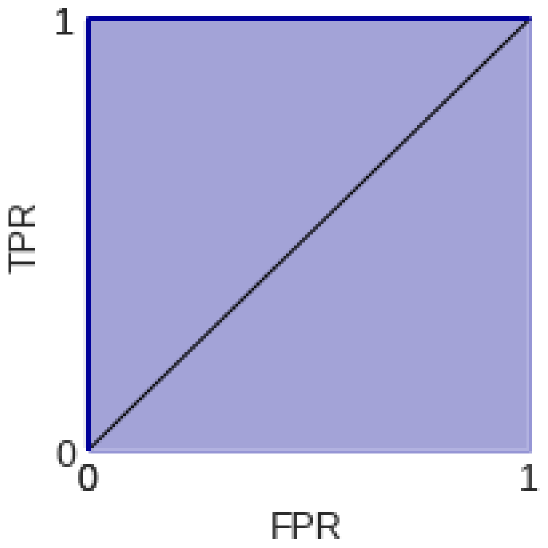
	оценка	класс
0	0.5	0
1	0.1	0
2	0.2	0
3	0.6	1
4	0.2	1
5	0.3	1
6	0.0	0

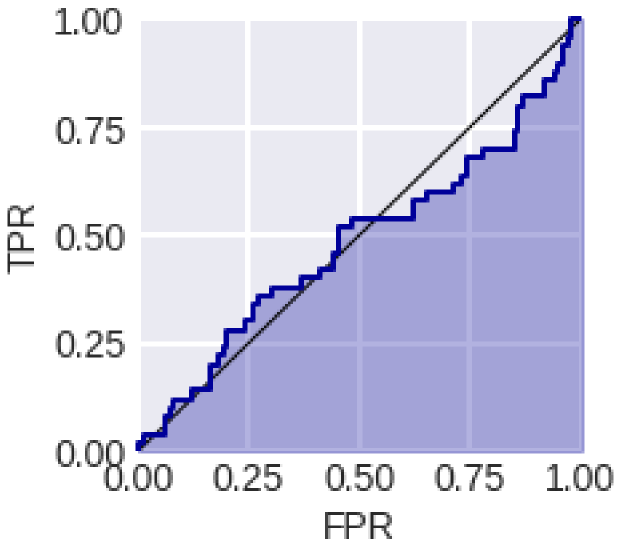
	оценка	класс	ответ
3	0.6	1	1
0	0.5	0	1
5	0.3	1	1
2	0.2	0	0
4	0.2	1	0
1	0.1	0	0
6	0.0	0	0

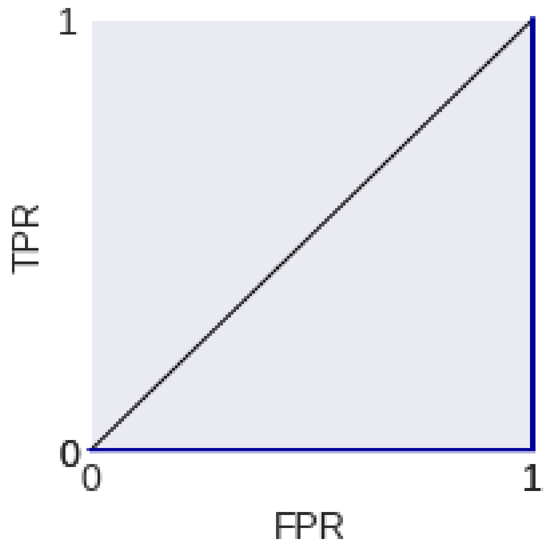



```
df['ответ'] = (df['оценка'] > 0.25).astype(int)
df.sort_values('оценка', ascending=False)
```

ROC и AUC ROC







наилучший (AUC=1), случайный (AUC~0.5) и наихудший (AUC=0) алгоритм

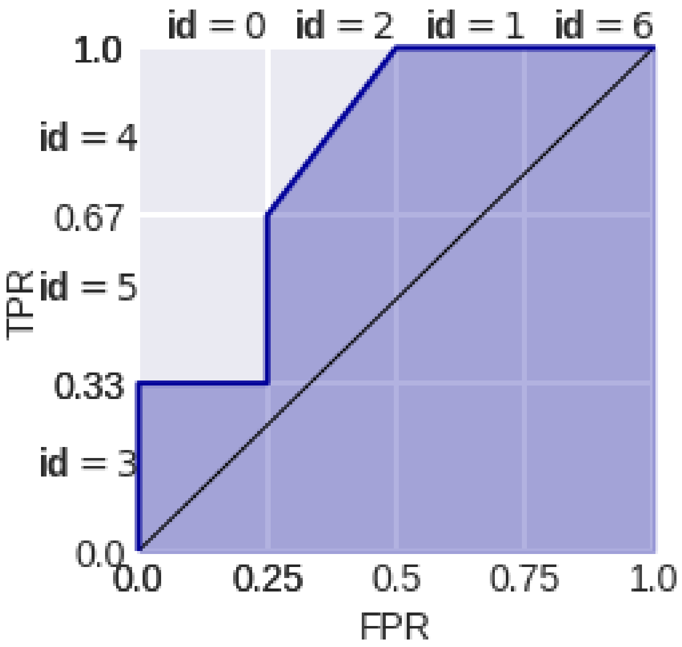
```
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(y_test, a)
plt.plot(fpr, tpr, lw=3, c='#000099')
```

Смысл AUC

AUC ~ число правильно отсортированных пар
(на рис. «кирпичики»)

Это сложно объяснить заказчику!

$$AUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} I[y_i < y_j] I[a_i < a_j]}{\sum_{i=1}^{m}\sum_{j=1}^{m} I[y_i < y_j]}$$



	оценка	класс	ответ
3	0.6	1	1
0	0.5	0	1
5	0.3	1	1
2	0.2	0	0
4	0.2	1	0
1	0.1	0	0
6	0.0	0	0

Смысл AUC

$$AUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} I[y_i < y_j] I[a_i < a_j]}{\sum_{i=1}^{m}\sum_{j=1}^{m} I[y_i < y_j]}$$

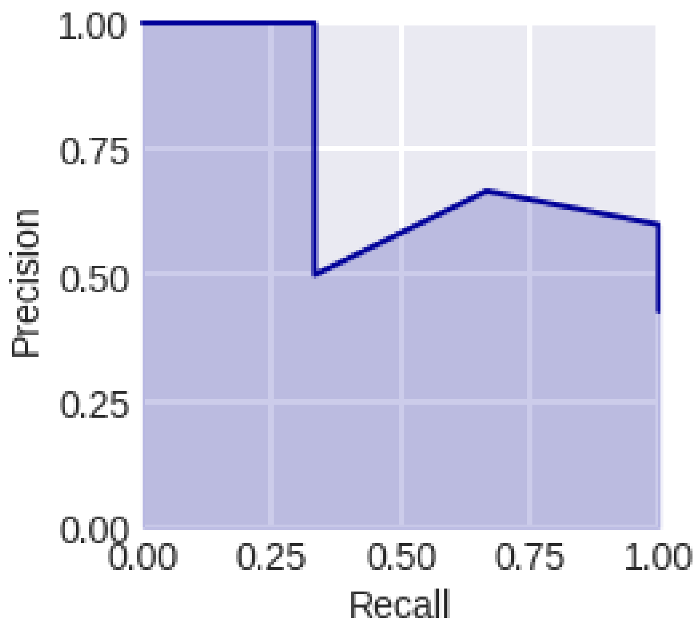
$$I[a_i < a_j] = \begin{cases} 1, & a_i < a_j, \\ 1/2, & a_i = a_j, \\ 0, & a_i > a_j. \end{cases}$$

Ещё примеры кривых… «полнота-точность»

Площадь под кривой.. «Average Precision» (есть и другой смысл)

	оценка	класс
0	0.5	0
1	0.1	0
2	0.2	0
3	0.6	1
4	0.2	1
5	0.3	1
6	0.0	0

	оценка	класс	ответ
3	0.6	1	1
0	0.5	0	1
5	0.3	1	1
2	0.2	0	0
4	0.2	1	0
1	0.1	0	0
6	0.0	0	0




```
from sklearn.metrics import precision_recall_curve
precision, recall, thresholds = precision_recall_curve(y_test, a)
plt.plot(recall, precision)
# вычисление площади методом трапеций
from sklearn.metrics import auc
auc(recall, precision)
# или готовую функцию использовать
from sklearn.metrics import average_precision_score
```

Многоклассовая задача «Multi-label»

матрица классификаций

$$\| y_{ij} \|_{m \times l}$$

	class 1	class 2	class 3
0	1	0	0
1	0	1	0
2	0	0	1
3	1	1	0

матрица ответов

$$\| a_{ij} \|_{m \times l}$$

	class 1	class 2	class 3
0	0.75	0.00	0.25
1	0.00	0.50	0.25
2	0.25	1.00	0.25
3	0.00	0.25	0.75

По сути, надо сравнить матрицы на похожесть

микро-подход	можно сравнивать матрицы как векторы
макро-подход	можно сравнивать столбцы матриц
по объектам	можно сравнивать строки матриц и усреднять

Многоклассовый AUCROC: Макро-усреднение

$$\mathrm{AUC} = \frac{1}{l}\sum_{j=1}^{l}\mathrm{AUC}_j$$

AUC_j **– значение функционала в задаче бинарной классификации**

«j-й класс / не j-й класс»

$$\{(x_i, I[y(x_i)_{[j]} = 1])\}_{i=1}^{m}$$

Многоклассовый AUCROC: Весовое макро-усреднение

$$\mathrm{AUC} = \frac{\sum_{j=1}^{l} P_j\,\mathrm{AUC}_j}{\sum_{j=1}^{l} P_j}$$

P_j **– вероятность j-го класса**

(процент «1» в столбце матрицы классификации)

Многоклассовый AUCROC: Микро-усреднение

значение функционала в задаче

$$\{((x_i, j), I[y(x_i)_{[j]} = 1])\}_{i=1, j=1}^{m, l}$$

«вытягиваем матрицу ответов в вектор»

Многоклассовый AUCROC: Усреднение по объектам

$$\mathrm{AUC} = \frac{1}{m}\sum_{i=1}^{m}\mathrm{AUC}'_i$$

AUC'_i **– значение функционала в задаче**

$$\{((x_i, j), I[y(x_i)_{[j]} = 1])\}_{j=1}^{l}$$

«решение задачи по строкам»

Многомерный AUC: минутка кода

```
from sklearn.metrics import roc_auc_score

roc_auc_score(y, a, average='macro')
# эквивалентно:
auc_pclass = [roc_auc_score(y[:,i], a[:,i]) for i in range(l)]
auc_pclass, mean(auc_pclass)

roc_auc_score(y, a, average='micro')
# эквивалентно:
roc_auc_score(y.ravel(), a.ravel())

roc_auc_score(y, a, average='weighted')
# эквивалентно:
w = y.sum(axis=0)
sum(np.array(auc_pclass) * w) / sum(w)

roc_auc_score(y, a, average='samples')
# эквивалентно:
auc_pinstance = [roc_auc_score(y[i,:], a[i,:]) for i in
range(m)]
auc_pinstance, mean(auc_pinstance)
```

Многомерный AUC ROC

матрица классификаций

	class 1	class 2	class 3
0	1	0	0
1	0	1	0
2	0	0	1
3	1	1	0

матрица ответов

	class 1	class 2	class 3
0	0.75	0.00	0.25
1	0.00	0.50	0.25
2	0.25	1.00	0.25
3	0.00	0.25	0.75

macro	micro	weighted	samples
0.49	0.53	0.52	0.56

	class 0	class 1	class 2
AUC_per_class	0.62	0.5	0.33
P_per_class	0.50	0.5	0.25

	class 0	class 1	class 2	class 3
AUC_per_instance	1.0	1.0	0.25	0.0

Литература

Стрижов В.В. Функция ошибки в задачах восстановления регрессии // Заводская лаборатория, 2013, 79(5): 65-73.
http://strijov.com/papers/Strijov2012ErrorFn.pdf

«How to Win a Data Science Competition: Learn from Top Kagglers»
https://ru.coursera.org/learn/competitive-data-science

конспект этих лекций
https://dyakonov.org/2018/10/23/функции-ошибок-в-задачах-регрессии/

Литература

Jeffrey M Girard «Inter-observer reliability» //
https://github.com/jmgirard/mReliability/wiki

Функционалы качества бинарной классификации
https://dyakonov.org/2019/05/31/функционалы-качества-в-задаче-бинарн/

Литература

Tom Fawcett An introduction to ROC analysis // Pattern Recognition Letters V.27 № 8, 2006, P. 861-874.
https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf

Интерактивная ROC-кривая
http://www.navan.name/roc/

Логистическая функция ошибки
https://dyakonov.org/2018/03/12/логистическая-функция-ошибки/

Кривые в машинном обучении
https://dyakonov.org/2019/08/29/кривые-в-машинном-обучении/

Калибровки
https://dyakonov.org/2020/03/27/проблема-калибровки-уверенности/