Системы искусственного интеллекта

Лекция

Ошибки и функционалы качества

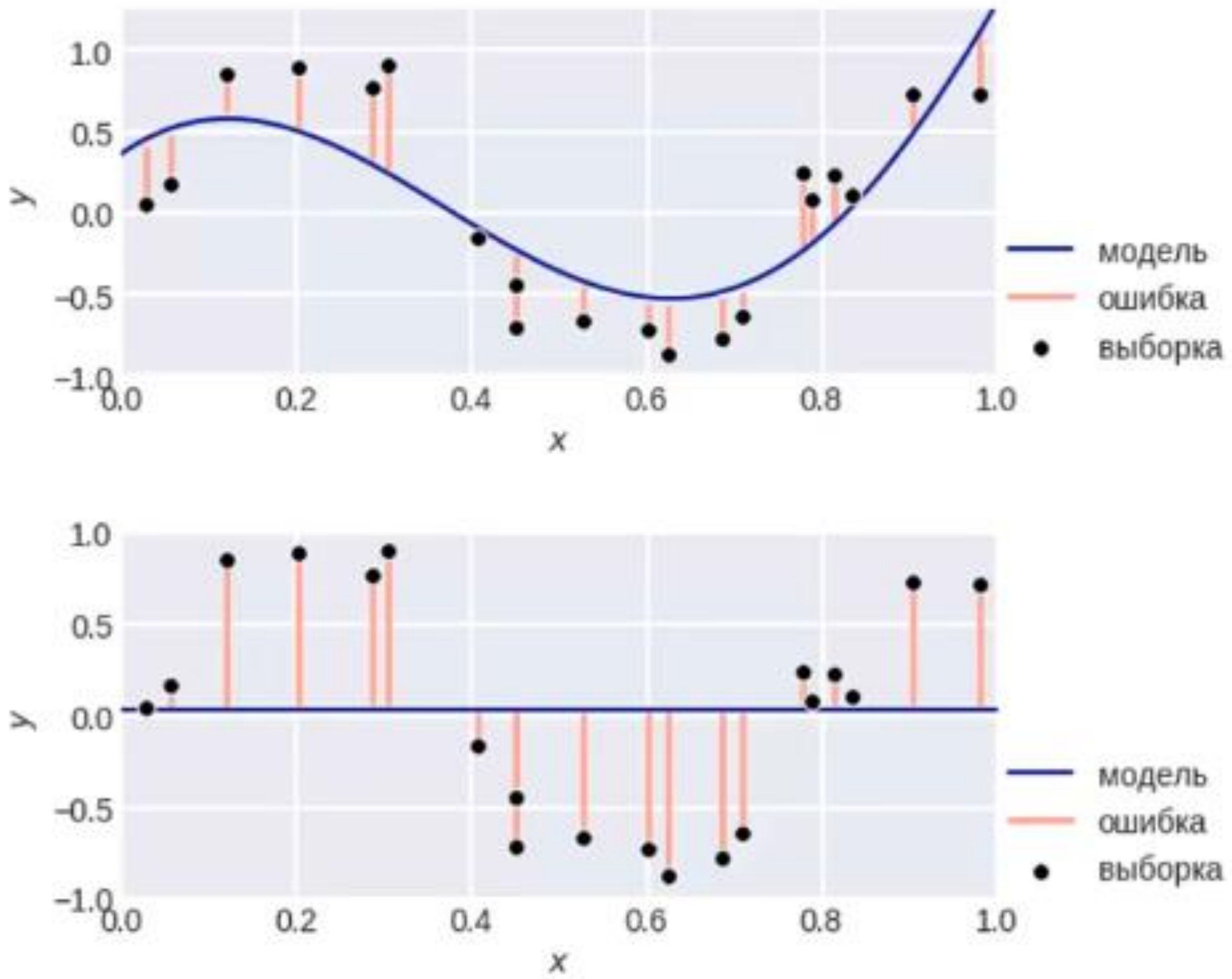
Запорожцев Иван Федорович
zaporozhtsev.if.work@gmail.com

Определение качества для задач…

- Задача регрессии
- Задача бинарной классификации
- Задача нечёткой бинарной классификации
- Задача классификации на несколько классов

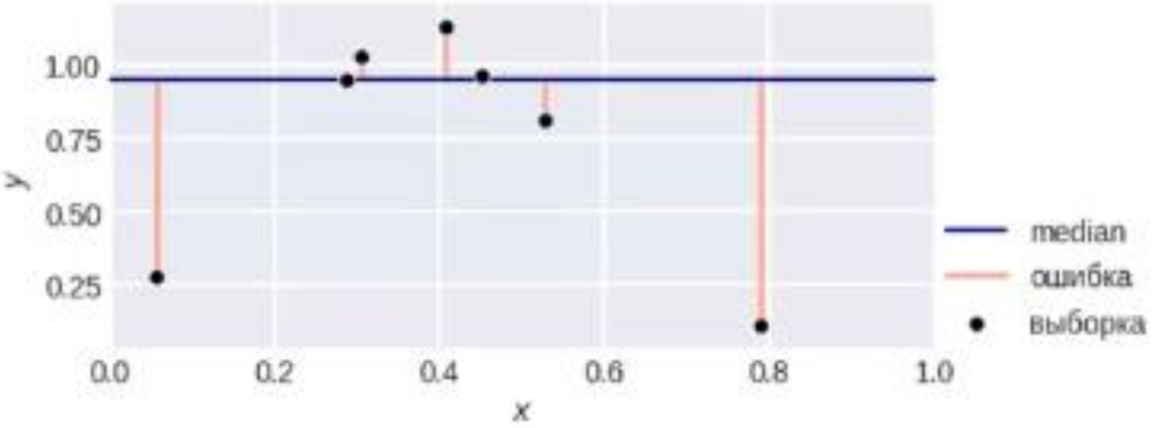
Определение качества для задач…

Задача регрессии



Средний модуль отклонения

Mean Absolute Error (MAE), Mean Absolute Deviation (MAD)



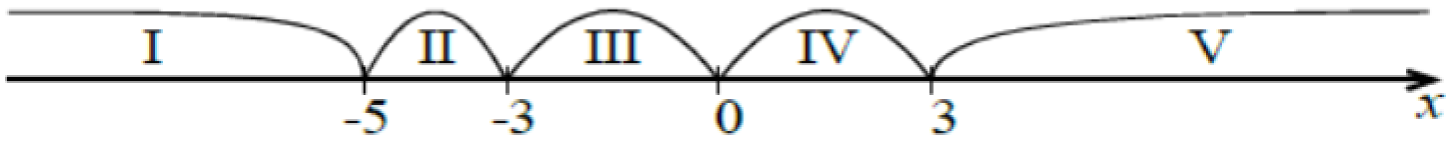
$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m} | a_i - y_i |$$

$$\sum_{i=1}^{m} | a - y_i | \to \min \qquad a = \mathrm{median}(\{y_i\}_{i=1}^{m})$$

$$\sum_{i=1}^{m} |a - y_i| \to \min \qquad a = \text{median}(\{y_i\}_{i=1}^{m})$$

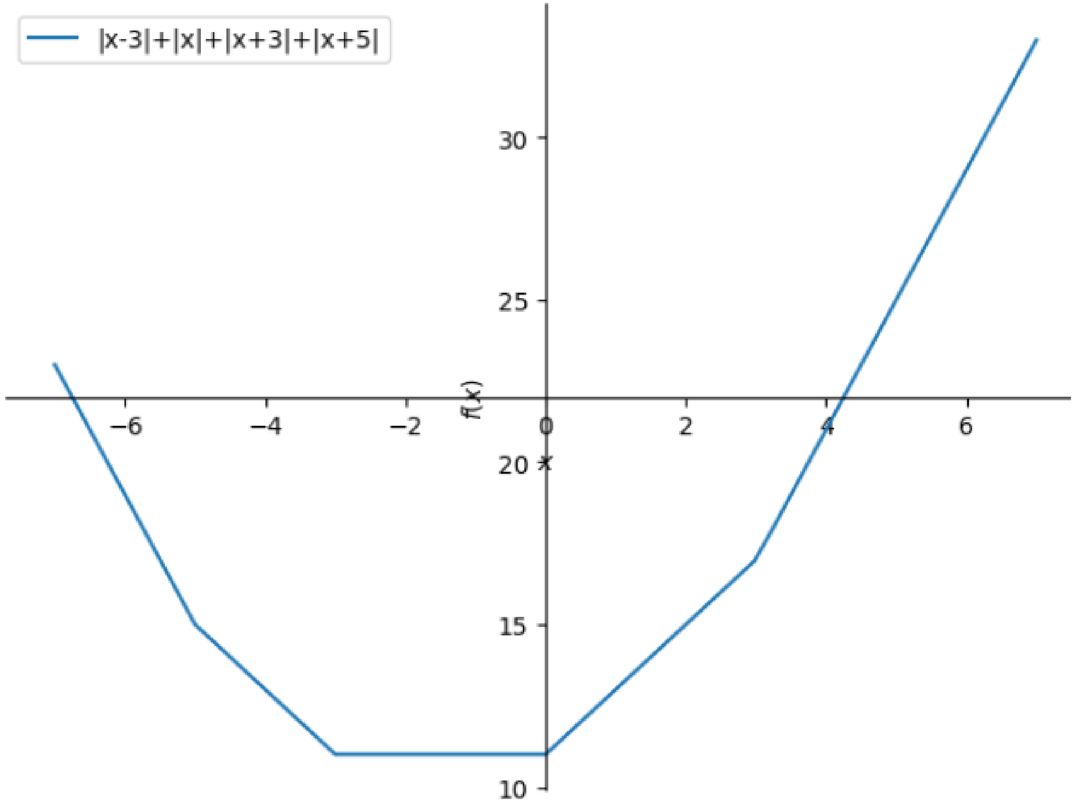
$$f(x) = |x-3| + |x| + |x+3| + |x+5|$$

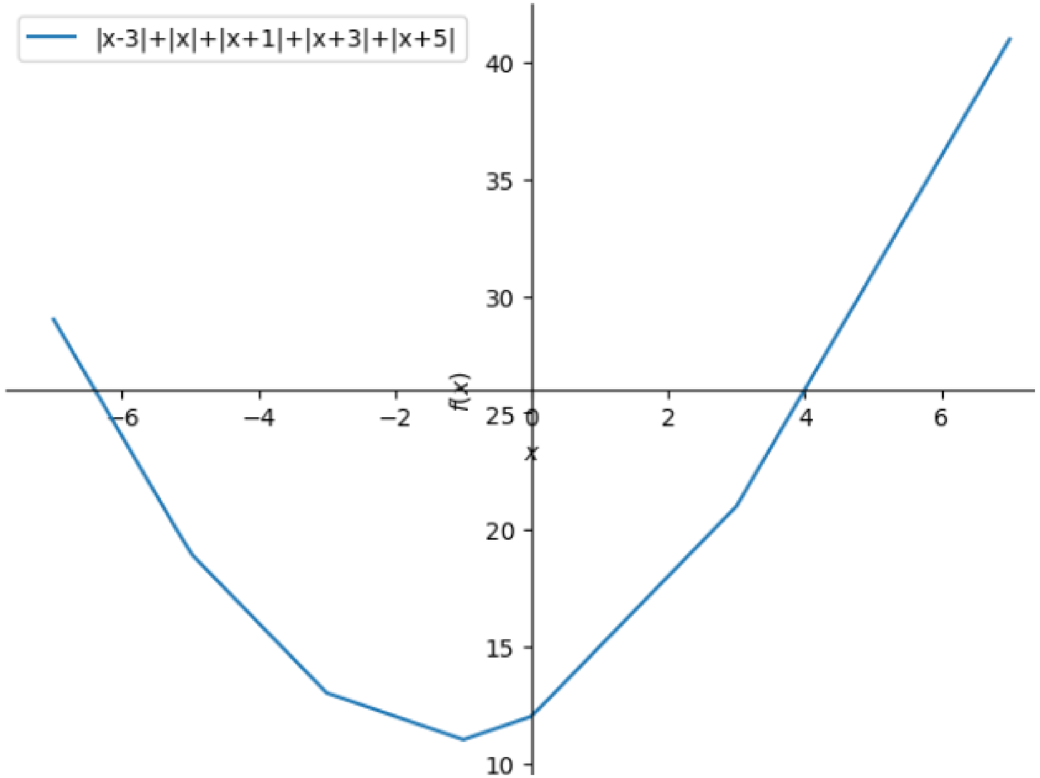
Как видим, функция зависит от четырёх модулей. Нанесём на числовую ось точки, в которых выражения под модулем обращаются в ноль.



	I	II	III	IV	V
	$(-\infty;-5)$	$[-5;-3)$	$[-3;0)$	$[0;3)$	$[3;+\infty)$
$x-3$	–	–	–	–	+
x	–	–	–	+	+
$x+3$	–	–	+	+	+
$x+5$	–	+	+	+	+
$f(x)$	$3-x-x-x-$ $-3-x-5$	$-x+3-x-x-$ $-3+x+5$	$-x+3-x+x+$ $+3+x+5$	$-x+3+x+x+$ $+3+x+5$	$x-3+x+x+$ $+3+x+5$

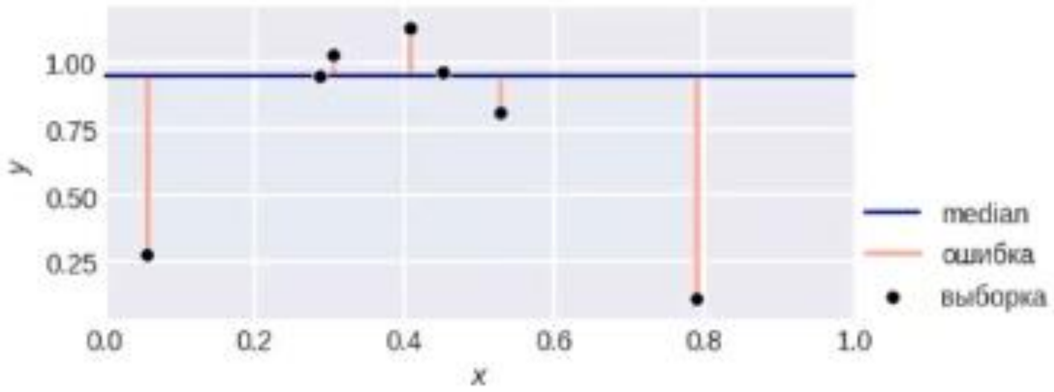
$$\sum_{i=1}^{m} | a - y_i | \to \min \qquad a = \mathrm{median}(\{y_i\}_{i=1}^{m})$$





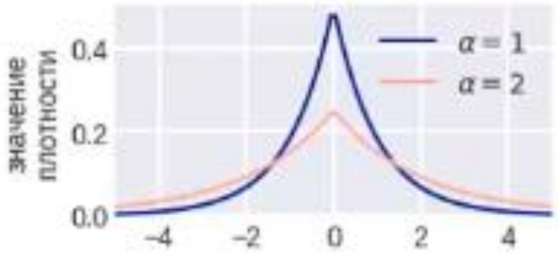
Средний модуль отклонения

Mean Absolute Error (MAE), Mean Absolute Deviation (MAD)



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

Если ошибка распределена по Лапласу

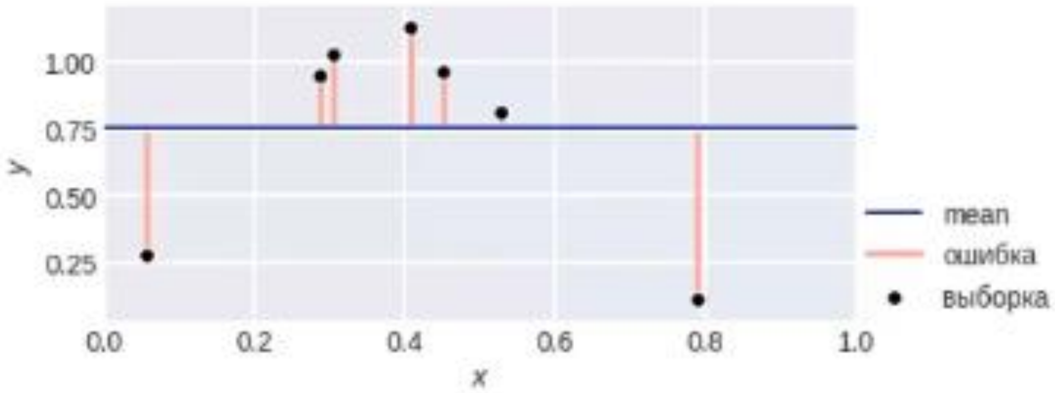


$$\frac{1}{m}\sum_{i=1}^{m}|a - y_i| \to \min$$

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

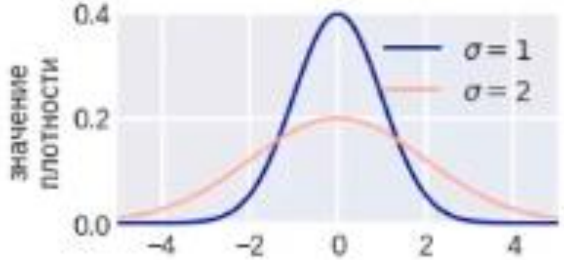
Средний квадрат отклонения

Mean Squared Error (MSE)



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

Если ошибка нормально распределена



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

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

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

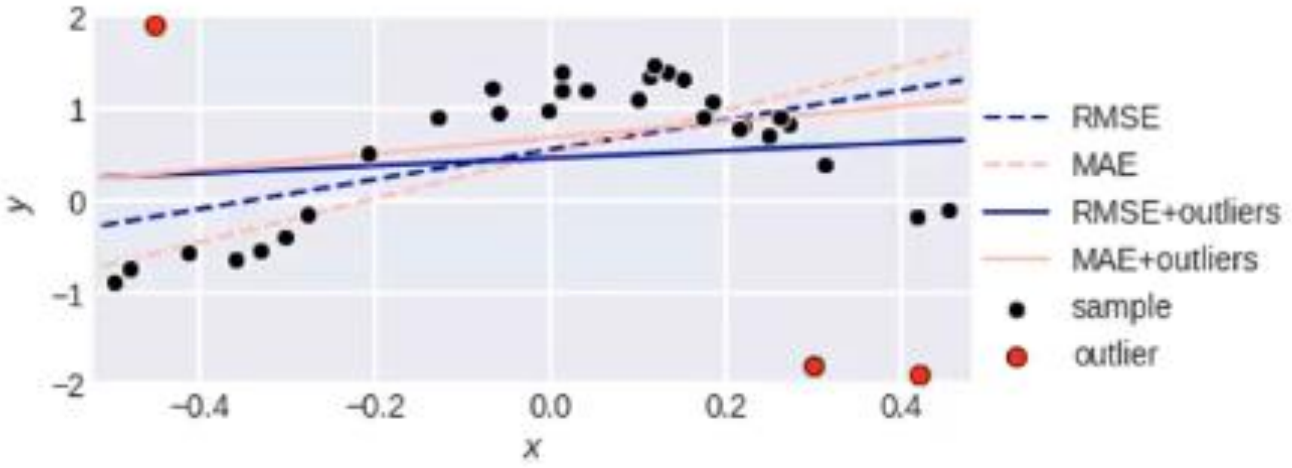
Нормированная версия: коэффициент детерминации R^2

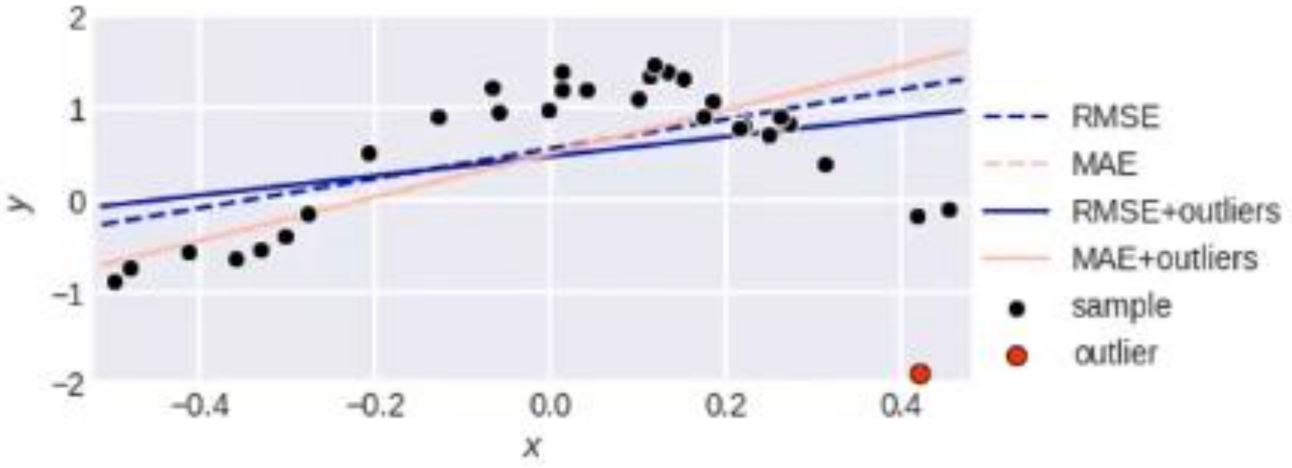
(Coefficient of Determination)

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

Различия MSE и MAE

Устойчивость к выбросам...





Symmetric mean absolute percentage error

SMAPE or sMAPE

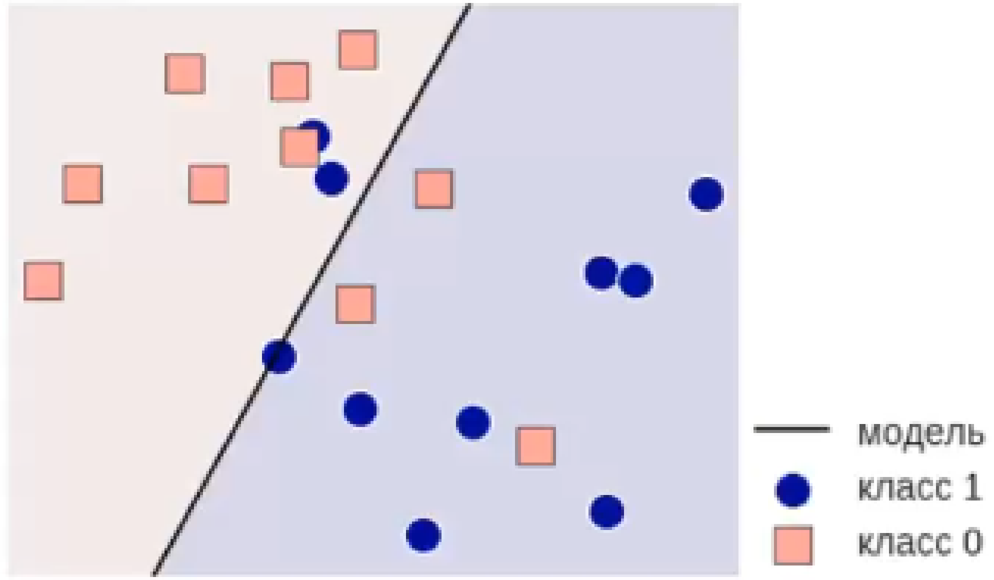
$$\text{SMAPE} = \frac{2}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{y_i + a_i} = 100\% \cdot \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - a_i|}{(y_i + a_i)/2}$$

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

Метрики в регрессии: минутка кода

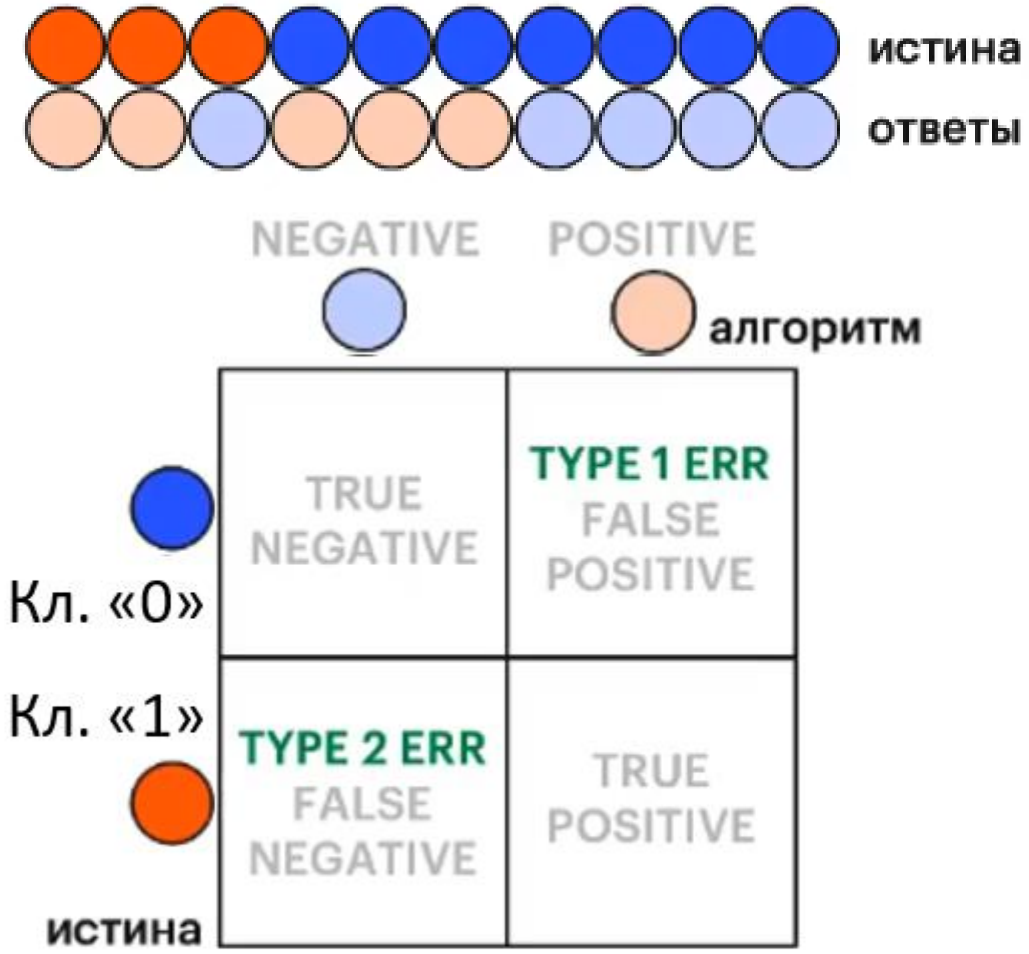
```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import median_absolute_error
from sklearn.metrics import explained_variance_score
# R^2
print (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))
# MSLpLE
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)))
```

Задача бинарной классификации



Сначала – чёткая классификация

«Confusion Matrix» в задаче классификации с двумя классами




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

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

Как запомнить названия ошибок

НЕТ эффекта, но его увидели!

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

ЕСТЬ эффект, но его НЕ увидели!

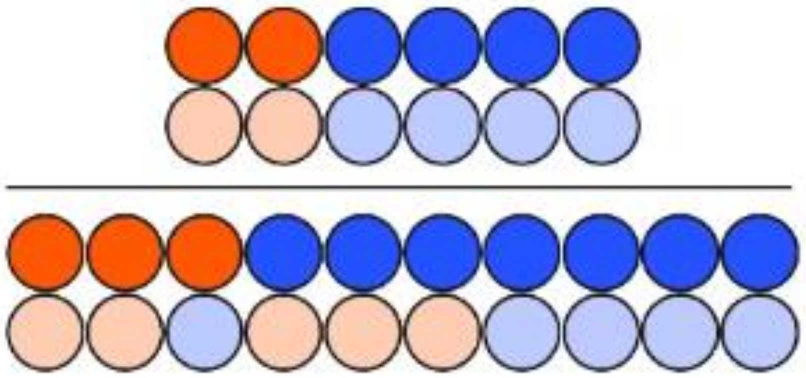
2 рода – учил, но не сдал
(= не знает по мнению экзаменатора)

истина \ алгоритм	NEGATIVE	POSITIVE
Кл. «0»	TRUE NEGATIVE	TYPE 1 ERR FALSE POSITIVE
Кл. «1»	TYPE 2 ERR FALSE NEGATIVE	TRUE POSITIVE

Точность

Accuracy

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



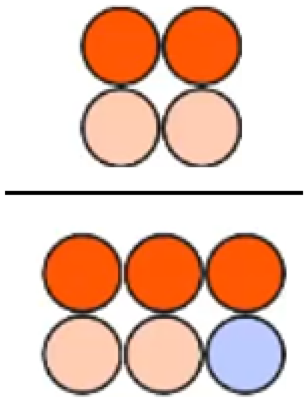
$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP}$$

Полнота

Sensitivity, True Positive Rate, Recall, Hit Rate

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

Какой процент объектов положительного класса мы правильно классифицировали



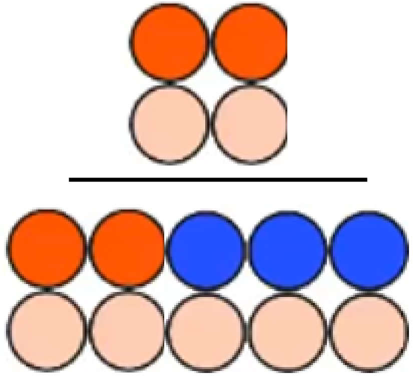
$$TPR = R = \frac{TP}{TP + FN}$$

Точность

Precision, Positive Predictive Value

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

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



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

Специфичность

Specificity, True Negative Rate

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

$$\mathrm{TNR} = \mathrm{Specificity} = R_0 = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Процент правильно классифицированных объектов негативного класса
«полнота для негативного класса»!

False Positive Rate

FPR, fall-out, false alarm rate

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

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

Доля объектов негативного класса, которых мы ошибочно отнесли к положительному

F_1 score

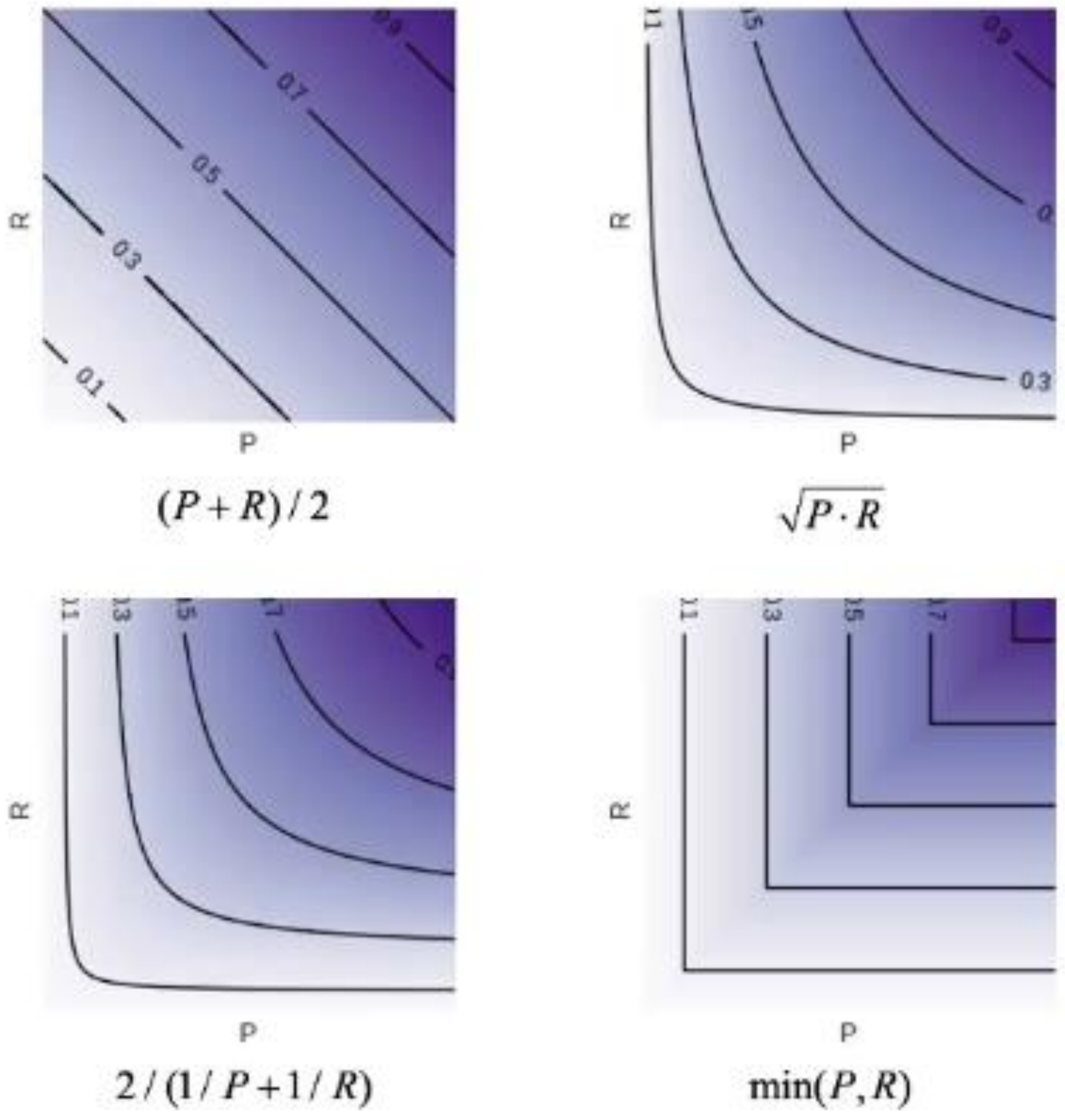
$$\frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2}{\frac{1}{TP/(TP+FP)}+\frac{1}{TP/(TP+FN)}} = \frac{2TP}{2TP+FP+FN}$$

F_β score

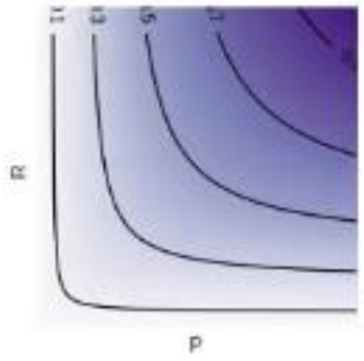
$$F_\beta = \frac{1}{\frac{\alpha}{P}+\frac{1-\alpha}{R}} = \frac{1}{\alpha}\frac{P\cdot R}{R+\left(\frac{1}{\alpha}-1\right)P} = (1+\beta^2)\frac{P\cdot R}{R+\beta^2 P}$$

$$\beta^2 = \left(\frac{1}{\alpha}-1\right)$$

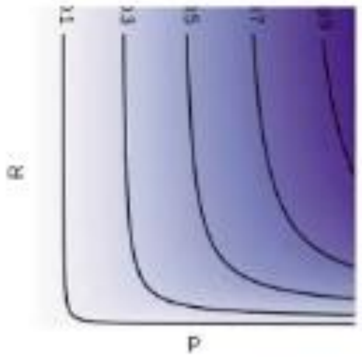
Почему используется F-мера



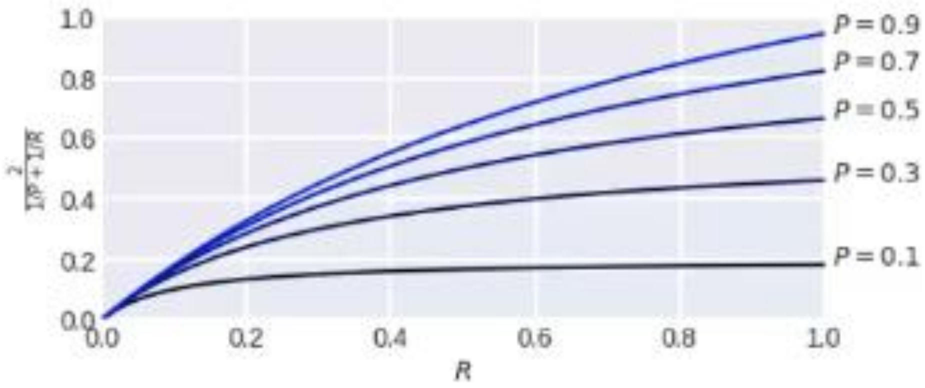
Почему используется F-мера



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



$1/(0.9/P+0.1/R)$



Можно сколь угодно улучшать один из показателей (R), если второй не увеличивается (P), то качество ограничено

Сбалансированная точность

Balanced Accuracy

Среднее арифметическое чувствительности и специфичности

$$\mathrm{BA} = \frac{R_1 + R_0}{2} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right)$$

Если классы примерно равномощны...

$$\mathrm{TP}+\mathrm{FN} \approx \mathrm{TN}+\mathrm{FP} \approx m/2$$

Минутка кода

	score
cohen_kappa_score	0.24
accurancy_score	0.81
matthews_corrcoef	0.26
f1_score	0.34
roc_auc_score	0.67
balanced_accuracy_score	0.67

```
from sklearn.metrics import classification_report
print (classification_report(y_test, a_test)) # нужен print

              precision    recall  f1-score   support

         0.0       0.94      0.84      0.89     16199
         1.0       0.26      0.50      0.34      1801

   micro avg       0.81      0.81      0.81     18000
   macro avg       0.60      0.67      0.61     18000
weighted avg       0.87      0.81      0.83     18000

from sklearn.metrics import cohen_kappa_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import f1_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import balanced_accuracy_score
```

Задача нечёткой 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

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

BCE, Binary Cross Entropy

Bernoulli distribution

$$f(k;p) = \begin{cases} p & \text{if } k = 1, \\ q = 1-p & \text{if } k = 0. \end{cases}$$

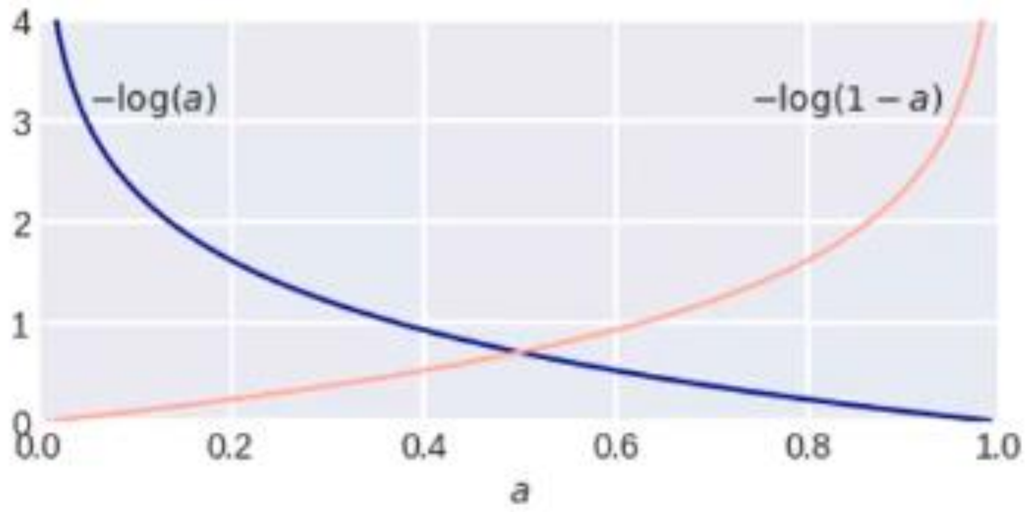
This can also be expressed as

$$f(k;p) = p^k(1-p)^{1-k} \quad \text{for } k \in \{0,1\}$$

Раздельная форма понятнее...

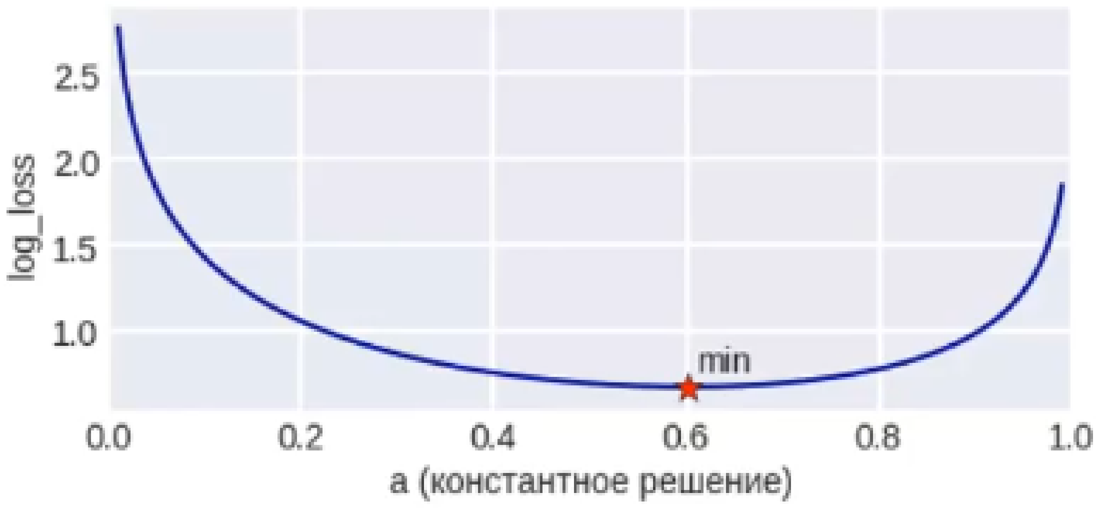
$$-\begin{cases} \log a_i, & y_i = 1, \\ \log(1-a_i), & y_i = 0. \end{cases}$$

Нельзя ошибаться!



Log Loss

Оптимальная константа для конечной выборки



$$-\frac{1}{m}\sum_{i=1}^{m}(y_i \log a + (1-y_i)\log(1-a)) \to \min_a$$

$$-\frac{m_1}{m}\log a - \frac{m_0}{m}\log(1-a) \to \min_a$$

$$\frac{\partial \ln loss}{\partial a} = 0 \underset{m,a>0,a\neq 1}{\Leftrightarrow} -\frac{m_1}{ma} + \frac{m_0}{m(1-a)}\bigg|_{a(x,w)=const\,(x,w)} = 0 \underset{m,a>0,a\neq 1}{\Leftrightarrow} a = \frac{m_1}{m}$$

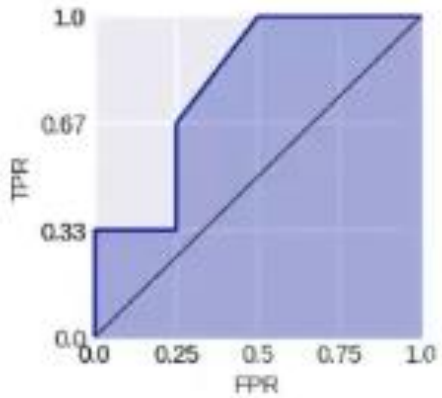
ROC и AUC ROC

ROC = receiver operating characteristic

Функционал зависит не от конкретных значений, а от их порядка

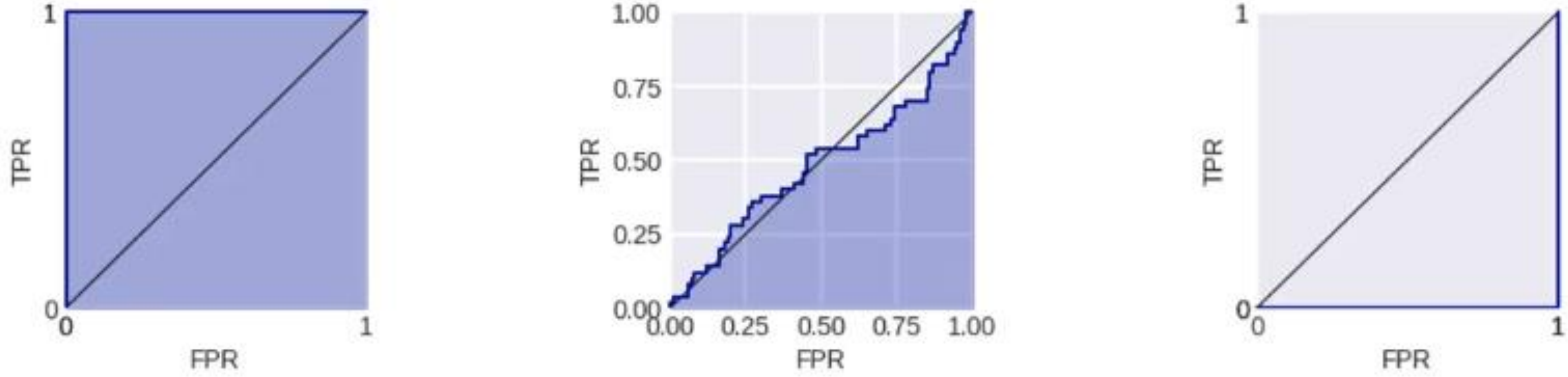
	оценка	класс
0	0.5	0
1	0.1	0
2	0.2	0
3	0.6	1
4	0.2	1
5	0.3	1
6	0.0	0

	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
0	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

AUC = area under the curve

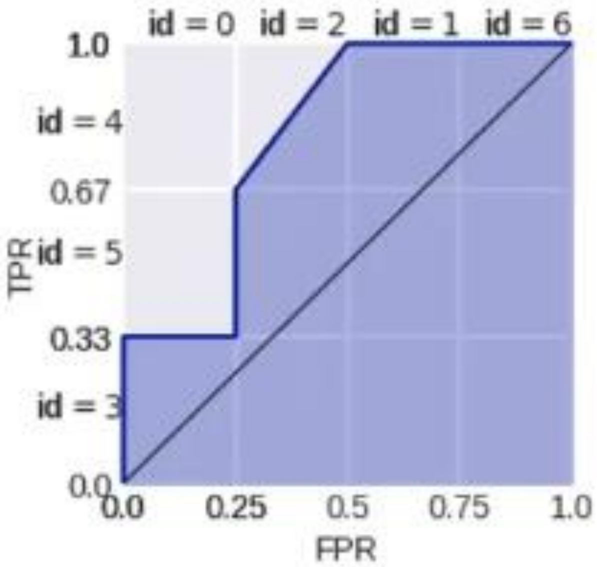
```
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(y_test, a)
plt.plot(fpr, tpr, lw=3, c='#000099')
```

Смысл AUC

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

Это сложно объяснить заказчику!

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



	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
0	0.0	0	0

Чем хороша эта запись?

Что неправильно
(требует пояснения) в формуле?

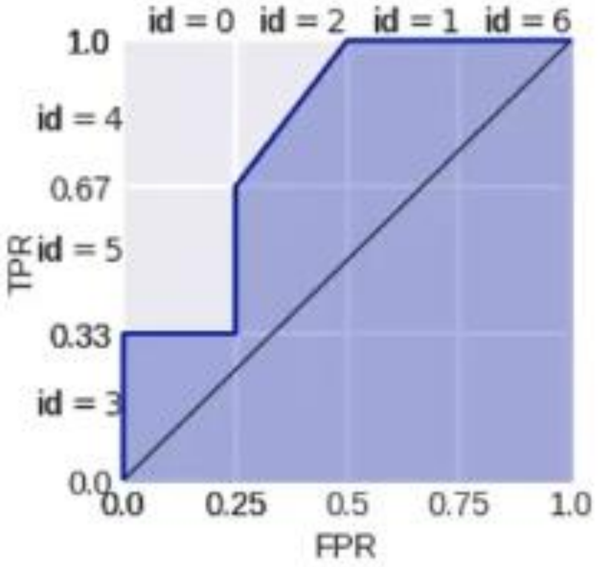
Смысл AUC

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

Это сложно объяснить заказчику!

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

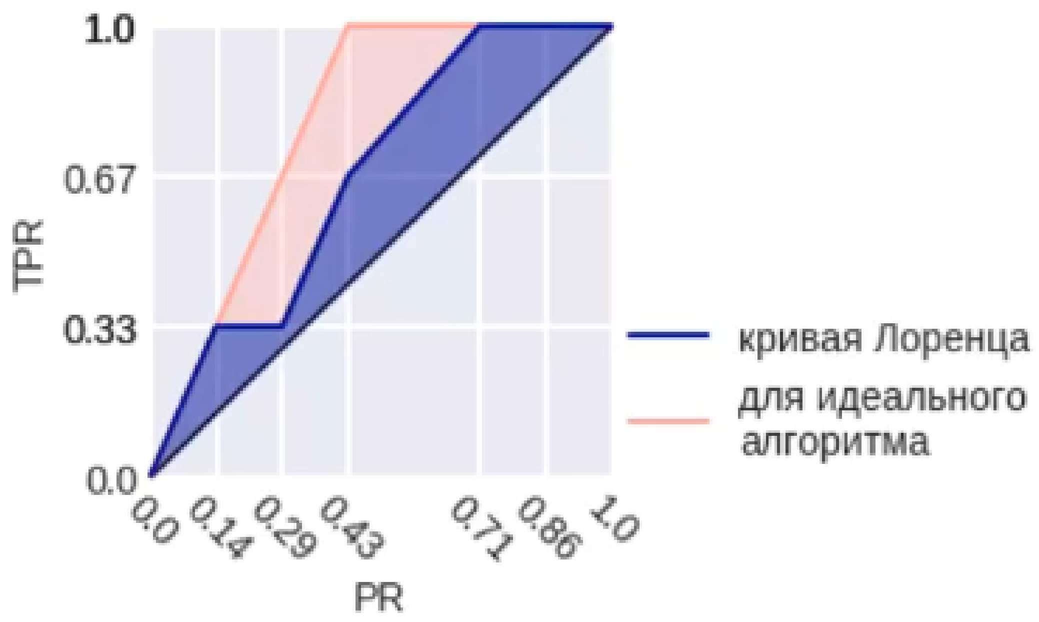
$$I[a_i < a_j] = \begin{cases} 1, & a_i < a_j, \\ 1/2, & a_i = a_j, \\ 0, & a_i > a_j. \end{cases}$$



	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
0	0.0	0	0

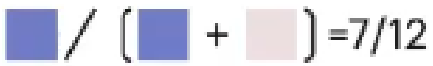
GINI в машинном обучении

Кривая Лоренца (или CAP – Cumulative Accuracy Profile Curve)



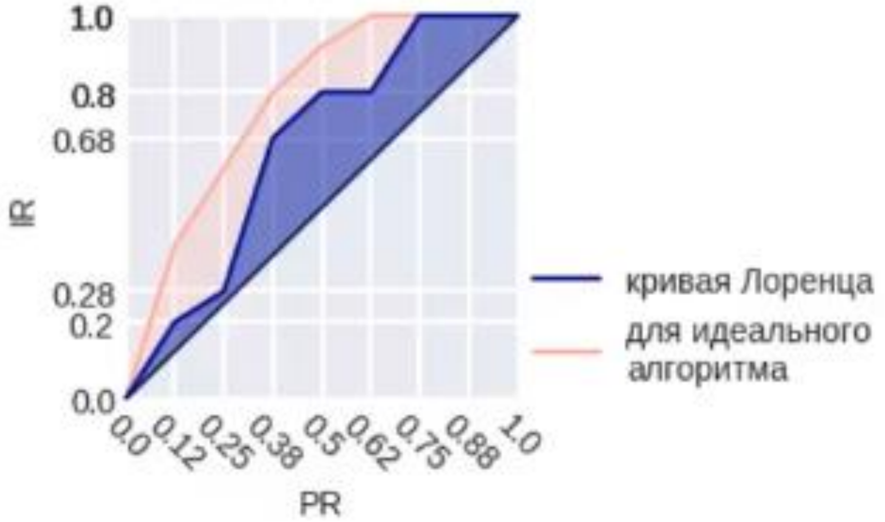
PR = Positive Rate – процент объектов, которые при определённом выборе порога, отнесены к классу 1

Коэффициент Джини – отношение площадей



$$\mathrm{gini} = 2\,\mathrm{AUCROC} - 1$$

GINI в задаче регрессии



Суммы страховых случаев:

5, 2, 10, 3, 0, 5, 0, 0

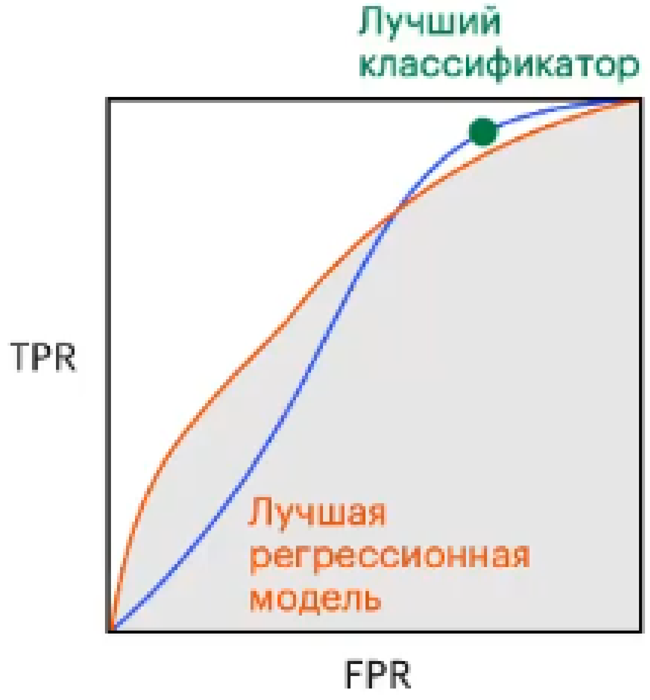
Так упорядочил алгоритм

Идеальный алгоритм:
10, 5, 5, 3, 2, 0, 0, 0

$$\text{gini} \approx 0.57$$

AUC ROC

+	в задачах, где важен порядок
+	учитывает разную мощность классов (не зависит от пропорций)
+	не важны значения, важен порядок
+	можно использовать для оценки признаков
–	«завышает» качество
–	оценивает не конкретный классификатор, а регрессию
–	сложно объяснить заказчику
–	не путать классификацию и регрессию



Ещё примеры кривых… «полнота-точность»

	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
0	0.0	0	0

Порог =0.25
Точность =
Полнота =

Ещё примеры кривых… «полнота-точность»

	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
0	0.0	0	0

Порог =0.25
Точность = 2/3
Полнота = 2/3

Ещё примеры кривых... «полнота-точность»

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

Порог =0.4
Точность =
Полнота =

Ещё примеры кривых… «полнота-точность»

	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	0
2	0.2	0	0
4	0.2	1	0
1	0.1	0	0
0	0.0	0	0

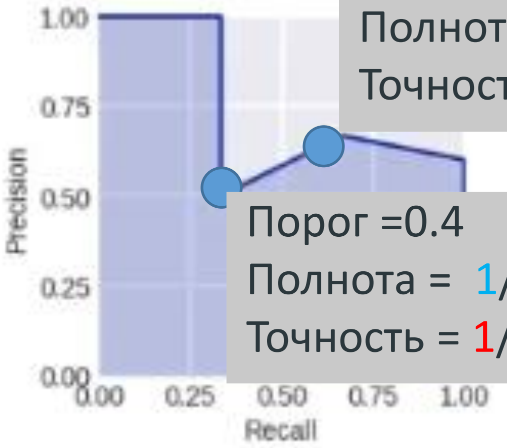
Порог =0.4
Точность = 1/2
Полнота = 1/3

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
0	0.0	0	0



Порог =0.25
Полнота = 2/3
Точность = 2/3

Порог =0.4
Полнота = 1/3
Точность = 1/2

```
from sklearn.metrics import precision_recall_curve
precision, recall, thresholds = precision_recall_curve(y_test, a)
plt.plot(recall, precision)
# [illegible]
from sklearn.metrics import auc
auc(recall, precision)
# [illegible]
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

Многомерный AUC: минутка кода

```
from sklearn.metrics import roc_auc_score

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

roc_auc_score(y, a, average='micro')
# [illegible]
roc_auc_score(y.ravel(), a.ravel())

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

roc_auc_score(y, a, average='samples')
# [illegible]
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	1	0
1	0	1	1
2	0	1	0
3	1	0	0

Матрица ответов

	class 1	class 2	class 3
0	0.7	0.6	0.5
1	0.3	0.4	0.6
2	0.5	0.9	0.2
3	0.4	0.5	0.1

macro	weighted	micro	samples
0.806	0.75	0.833	0.875

class 1	0.750
class 2	0.667
class 3	1.000

P_1	0.50
P_2	0.75
P_3	0.25

x_1	1.0
x_2	1.0
x_3	1.0
x_4	0.5

Многоклассовая задача: Log Loss

Cross-entropy

Естественное обобщение логистической ошибки

$a_{ij} \in [0,1]$

$$\text{logloss} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij}\log a_{ij}$$

тонкость: лучше для непересекающихся классов,

$\{a_{ij}\}_{j=1}^{l}$ ~ распределение

В задаче с пересекающимися классами (multi-label)

$$\text{logloss}_{\text{multi-label}} = -\frac{1}{l}\frac{1}{m}\sum_{j=1}^{l}\sum_{i=1}^{m}(y_{ij}\log a_{ij} + (1-y_{ij})\log(1-a_{ij}))$$

$a_{ij} \in [0,1]$ макроусреднение logloss по классам

Сбалансированная точность «Balanced accuracy»

Макро-усреднение полноты

Сбалансированная точность (accuracy) не есть усреднение точностей (precision)

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} R_j = \frac{1}{l}\sum_{j=1}^{l} \frac{\sum_{i=1}^{m} I[y(x_i)_{[j]} = 1] I[a(x_i)_{[j]} = 1]}{\sum_{i=1}^{m} I[y(x_i)_{[j]} = 1]}$$

```
from sklearn.metrics import balanced_accuracy_score
```

Другие (неэквивалентные) определения:

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} \min[P_j, R_j]$$

$$\mathrm{BA} = \frac{1}{l}\sum_{j=1}^{l} \min[\mathrm{sens}_j, \mathrm{spec}_j]$$