

МЕГАФОН



# МЕТРИКИ

ВОЛОДКИНА ЕКАТЕРИНА

If you can not  
measure it,  
you can not  
improve it.



- Lord Kelvin



# Зачем нужны метрики качества?

Для оценки качества работы модели

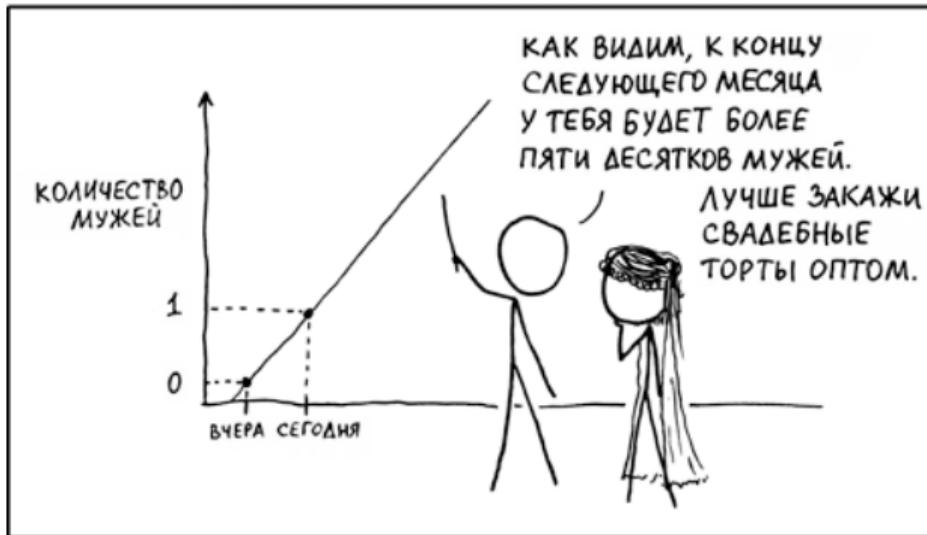
Для сравнения моделей

Для интерпретации результатов



# Метрики регрессии

МОЁ ХОББИ: ЭКСТРАПОЛИРОВАТЬ





## Постановка задачи

$X$  - множество **объектов**;

$Y \in \mathbb{R}$  - множество **ответов**;

$\{x_1, \dots, x_\ell\} \subset X$  - обучающая **выборка**

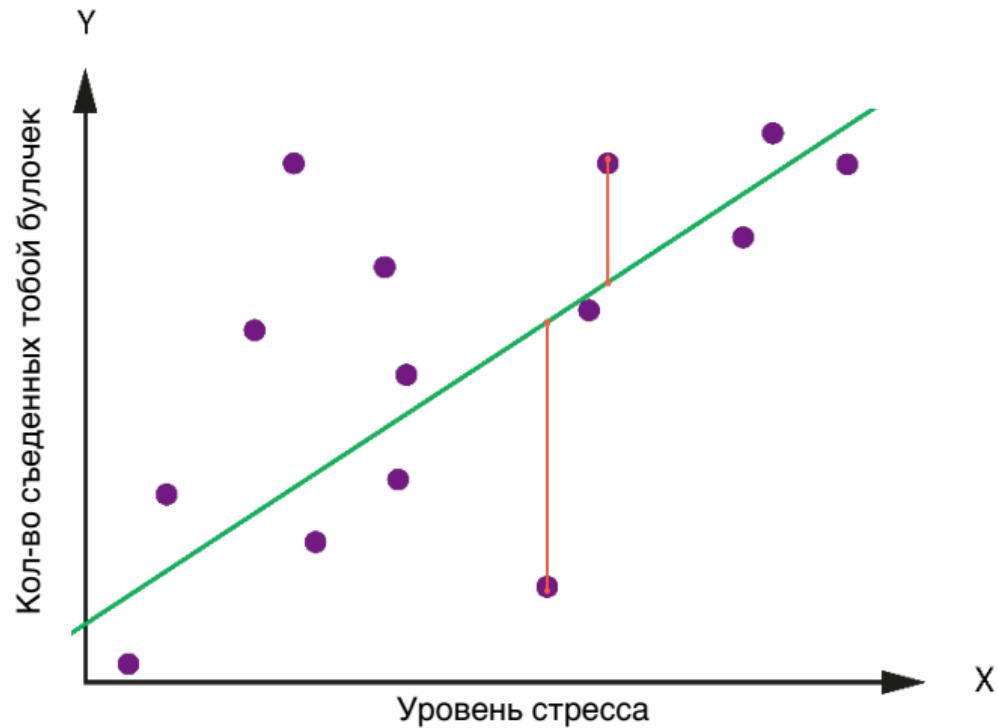
$y_i = y(x), i = 1, \dots, \ell$  - известные **ответы**

$a : X \rightarrow Y$  - **алгоритм**, решающий функцию (decision function),  
приближающую  $y$  на всем мн-же  $X$

$a_i = a(x), i = 1, \dots, \ell$  - **ответы** нашего алгоритма (**предсказанное значение**)



## Остатки



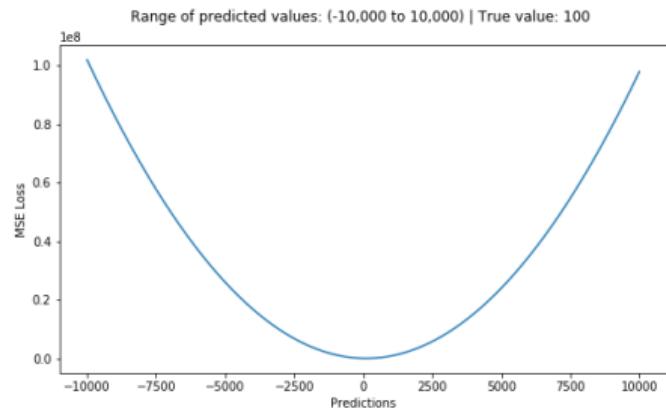


# MSE (Mean Squared Error)

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

C = target mean

- Дифференцируемая
- Чувствительна к выбросам
- Сложно интерпретировать





# RMSE (Root Mean Squared Error)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} = \sqrt{\text{MSE}}$$

C = target mean

- Дифференцируемая
- Чувствительна к выбросам
- Интерпретация: стандартное отклонение ответа

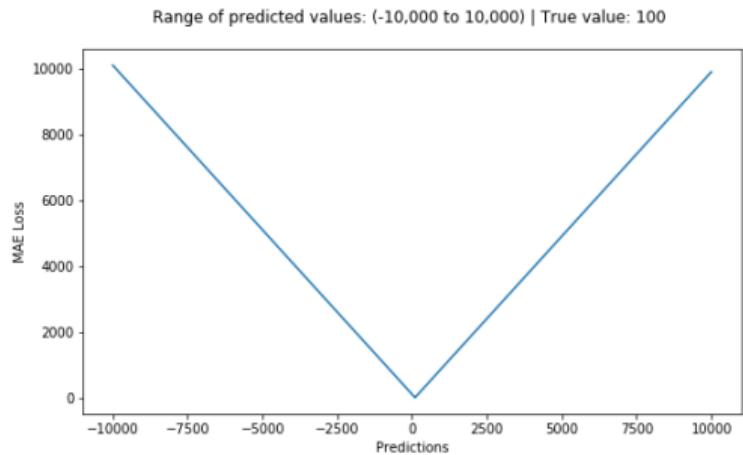


# MAE (Mean Absolute Error)

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

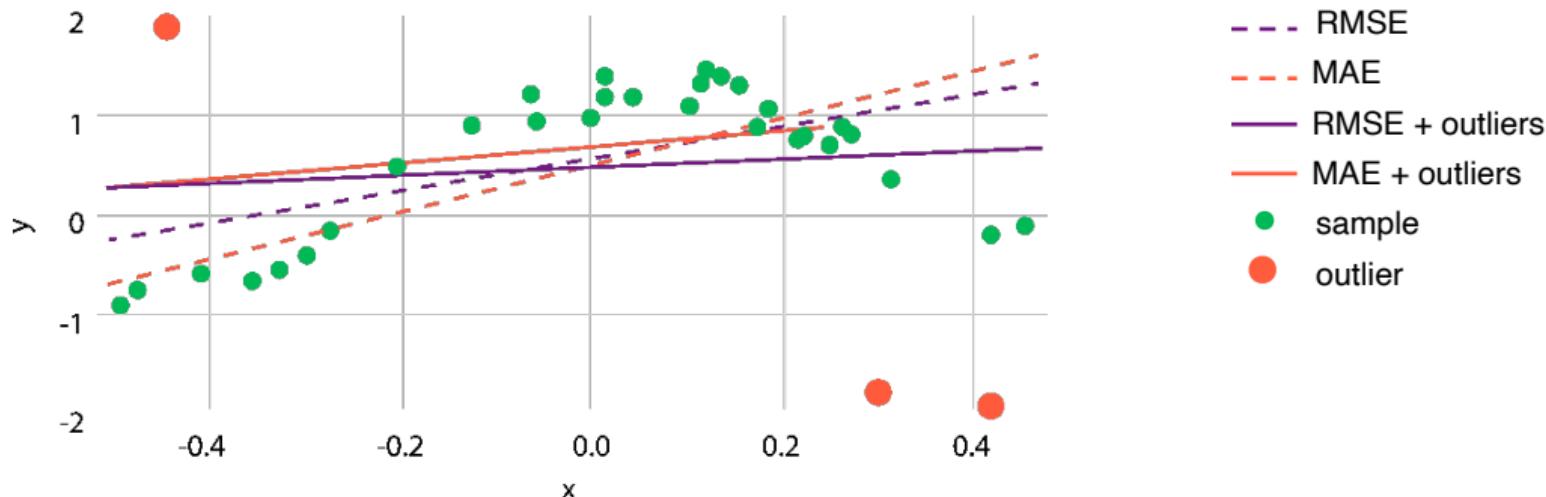
C = target median

- Единицы измерения как у таргета
- Сложно интерпретировать
- Нечувствителен к выбросам
- Не дифференцируемая



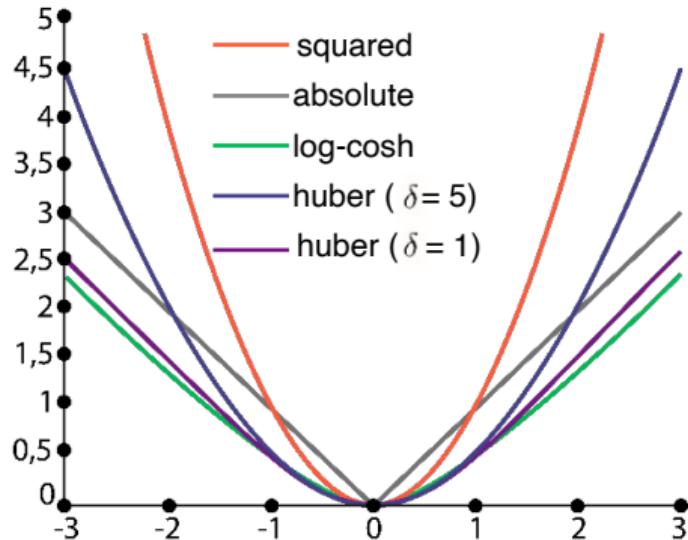


# Чувствительность к выбросам





# Huber и log-cosh



$$\text{huber } (\delta = 5) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

$$\text{log-cosh} = \log(\cosh(h(\mathbf{x}_i) - y_i))$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

- Сглаживают МАЕ
- Дифференцируемые.
- Можно использовать как loss.
- Нечувствительны к выбросам
- Сложно интерпретировать



# MSPE MAPE

## Mean Squared Percent Error

$$\text{MSPE} = \frac{100\%}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2$$

C = weighted target mean

## Mean Absolute Percent Error

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

C = weighted target median

- Присваивают больший вес абсолютно более маленьким объектам => смещены.
- Нечувствительны к выбросам.
- Хорошо интерпретируются: относительный прирост.



# RMSLE (Root Mean Squared Logarithmic Error)

$$\begin{aligned}\text{RMSLE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2} = \\ &= RMSE(\log(y_i + 1), \log(\hat{y}_i + 1)) = \\ &= \sqrt{MSE(\log(y_i + 1), \log(\hat{y}_i + 1))}\end{aligned}$$

C = exp(target mean)

- RMSLE = RMSE in log space
- RMSLE иногда противопоставляют MAPE, так как она менее смещена по отношению маленьким объектам



## $R^2$ (коэффициент детерминации)

$$R^2 = 1 - \frac{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} = 1 - \frac{MSE}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

- Дифференцируемый
- Чувствителен к выбросам
- Хорошо интерпретируется: насколько наша модель лучше, чем константное решение



## Выводы

Аномальные значения - лишь неожиданные значения, которые нужно учитывать?

Используем MSE, RMSE или  $R^2$  для интерпретации результатов.

Аномальные значения - это выбросы?

Вычищаем их или используем MAE или MAPE для интерпретации результатов.



# Метрики классификации





## Постановка задачи

$X$  - множество **объектов**;

$Y \in \mathbb{R}$  - множество **ответов**;

$\{x_1, \dots, x_\ell\} \subset X$  - **обучающая выборка**

$y_i = y(x), i = 1, \dots, \ell$  - известные **ответы**

$a : X \rightarrow Y$  - **алгоритм**, решающий функцию (decision function),  
приближающую  $y$  на всем мн-же  $X$

$a_i = a(x), i = 1, \dots, \ell$  - **ответы** нашего алгоритма (**метка класса или вероятность**)



# Accuracy (доля правильных ответов)

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

$C$  = самый популярный класс



KFC = 10

Dog = 90

—

Accuracy = ???



# Accuracy (доля правильных ответов)

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

$C$  = самый популярный класс



KFC = 10

Dog = 90

—

Accuracy = 0.9!

Чувствителен к  
дисбалансу  
классов!



# Confusion Matrix

		Actual Values	
		1	0
Predicted Values	1	 TRUE POSITIVE	 FALSE POSITIVE
	0	 FALSE NEGATIVE TYPE 2 ERROR	 TRUE NEGATIVE

$H_0$ : человек не ждет ребенка  
 $H_a$ : ох как ждет :)



# Precision и Recall

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Точность** - доля беременных среди всех предсказанных моделью беременных

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

**Полнота** - доля предсказанных моделью беременных среди всех беременных

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



# Precision и Recall

Predicted Values

		Actual Values	
		Positive (1)	Negative (0)
Positive (1)	TP	FP	
	FN	TN	
Negative (0)			

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

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

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

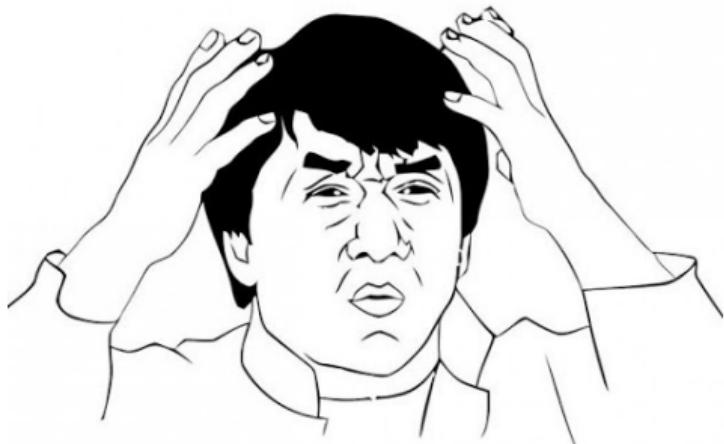
$$FPR = \frac{FP}{FP + TN}$$

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

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



# Precision и Recall



$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

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

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



# Precision vs Recall





# Пример: Спам



spam

not spam

sent to spam folder

true  
positives

false  
positives



sent to inbox

false  
negatives

true  
negatives





## Пример: Болезнь



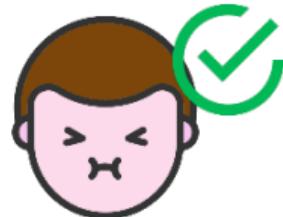
sick

healthy

diagnosed sick

true  
positives

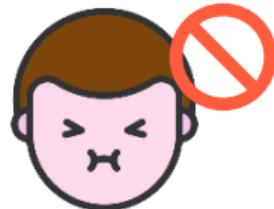
false  
positives



diagnosed healthy

false  
negatives

true  
negatives





## Пример: Спам



sent to spam folder

spam

not spam

sent to inbox

false  
negatives



false  
positives





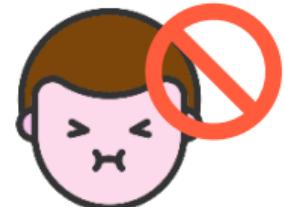
## Пример: Болезнь



sick

diagnosed sick

diagnosed healthy



false  
negatives



healthy

false  
positives

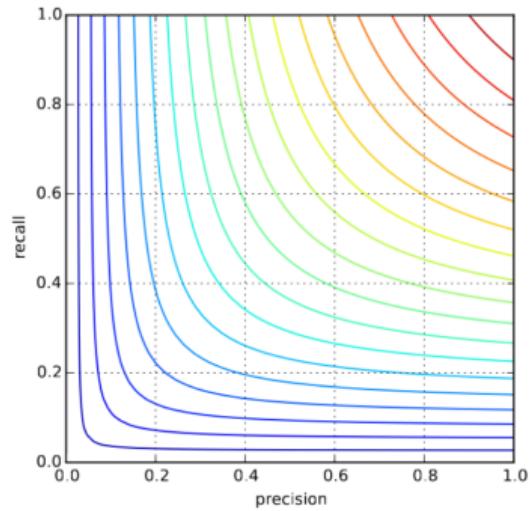


# F-measure

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{true positive}}{(1 + \beta^2) \cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}$$

$$F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



Стремится к нулю, когда хотя бы один из аргументов близок к нулю.  
 $\beta$  - определяет важность recall по сравнения с precision.



# Classification report w sklearn

```
from sklearn.metrics import classification_report  
  
print(classification_report(y_test,y_hat_test))
```

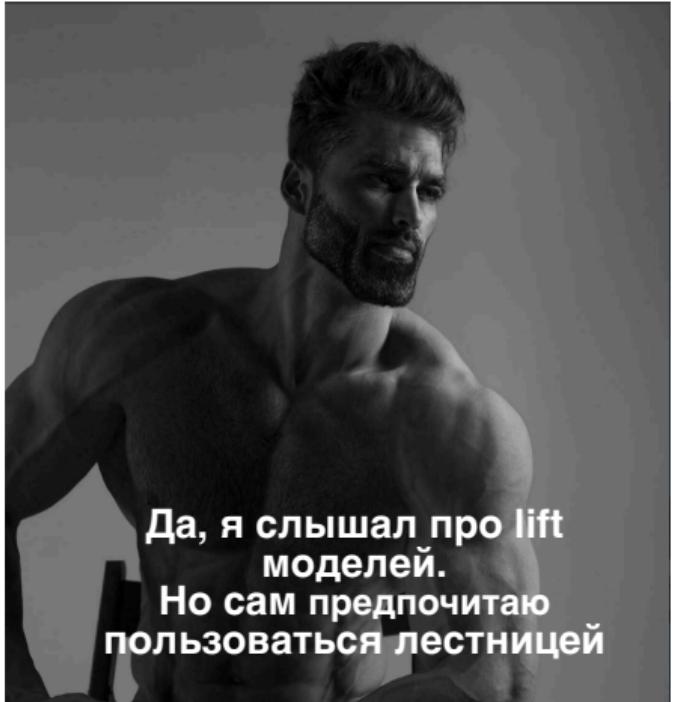
	precision	recall	f1-score	support
0	0.74	0.84	0.79	12733
1	0.96	0.92	0.94	48532
micro avg	0.91	0.91	0.91	61265
macro avg	0.85	0.88	0.86	61265
weighted avg	0.91	0.91	0.91	61265



# Lift (прирост концентрации)

$$\text{lift} = \frac{\text{precision}}{(TP + FN)/\ell}$$

Во сколько раз модель определяет точнее чем случайный выбор.

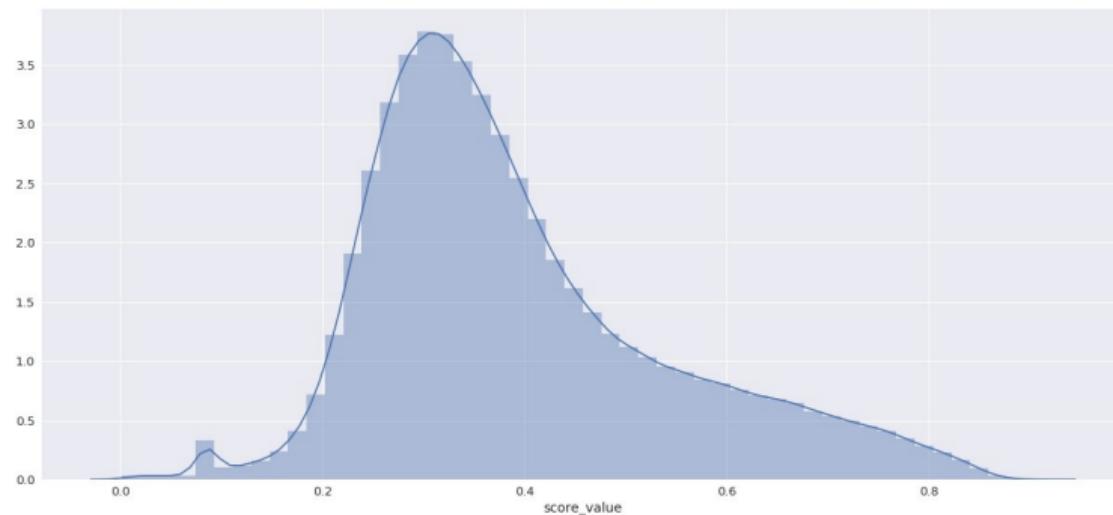




## Soft target

$a : X \rightarrow Y'$ ,  $Y' \in (0, 1)$  - алгоритм предсказывает значение от 0 до 1  
(например, вероятность принадлежности к положительному классу)

Распределение предсказания на тестовом множестве

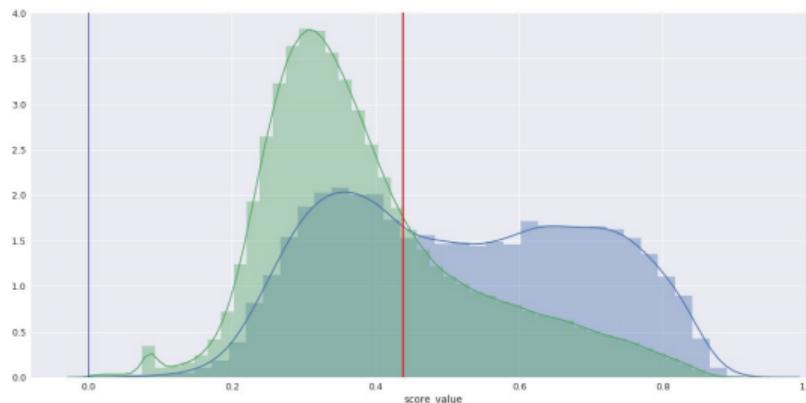




# Поквантильная таблица

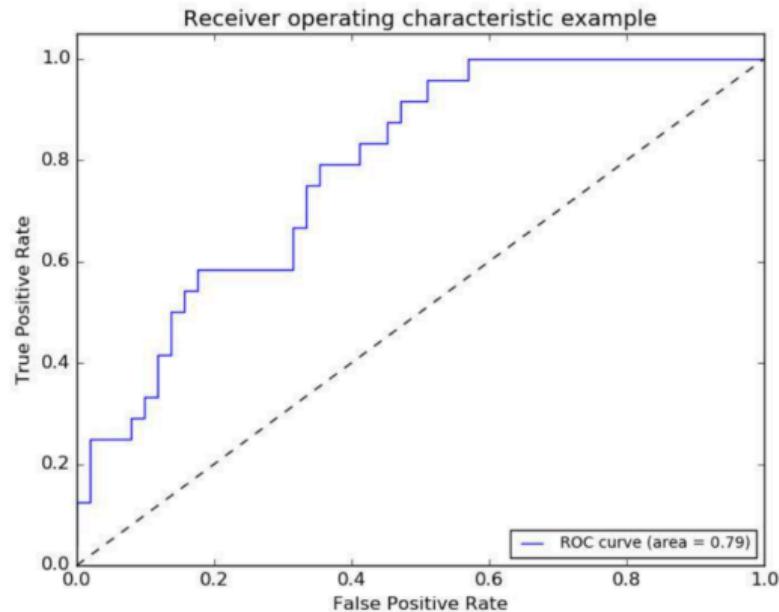
top 1 top 5 top 10 top 30 top 50 top 70 top 100

	top 1	top 5	top 10	top 30	top 50	top 70	top 100
lift	2.78	2.52	2.49	2.13	1.67	1.33	1.0
precision	88.10	79.70	78.80	67.30	52.90	42.00	31.7
recall	2.80	12.50	24.80	63.70	83.50	92.70	100.0
count	59.00	295.00	593.00	1784.00	2975.00	4158.00	5950.0
q	0.99	0.95	0.90	0.90	0.50	0.30	0.0



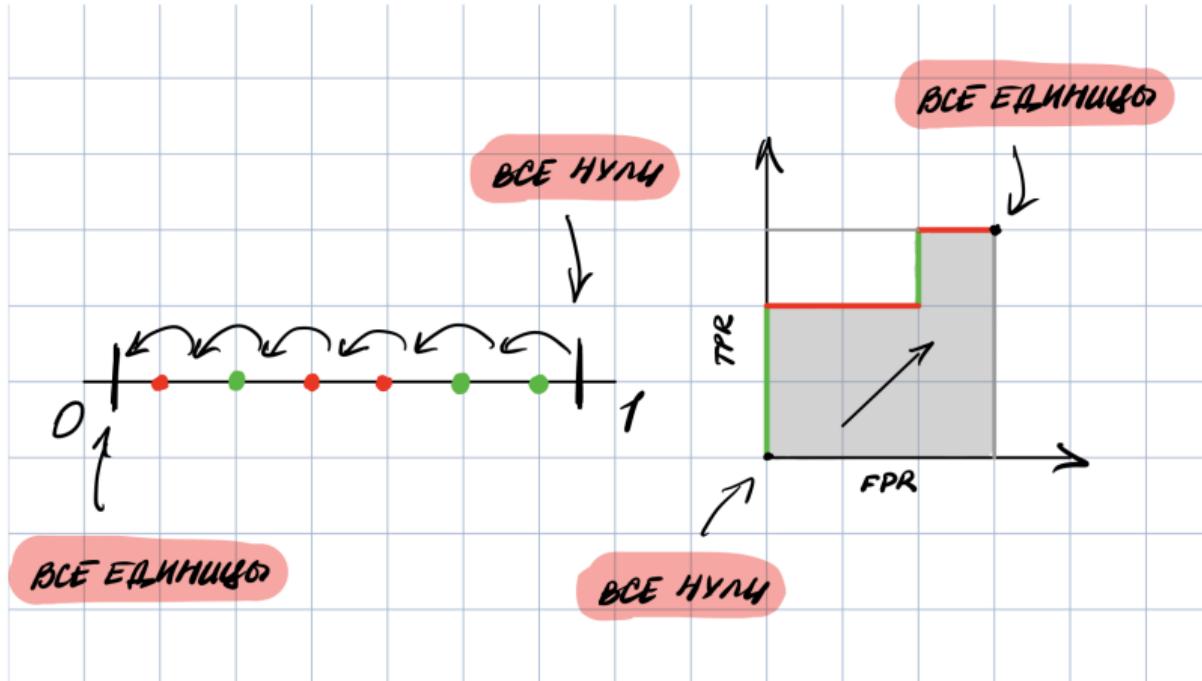


# ROC AUC (ROC Area Under Curve)





# ROC AUC



$$\text{ROC-AUC} = 7 / 9$$



# ROC AUC

\*только пары с разными метками

$$\text{AUC} = \frac{\# \text{ correctly ordered pairs}}{\text{total number of pairs}} = 1 - \frac{\# \text{ incorrectly ordered pairs}}{\text{total number of pairs}}$$
$$= 1 - 2 / 9 = 7 / 9$$



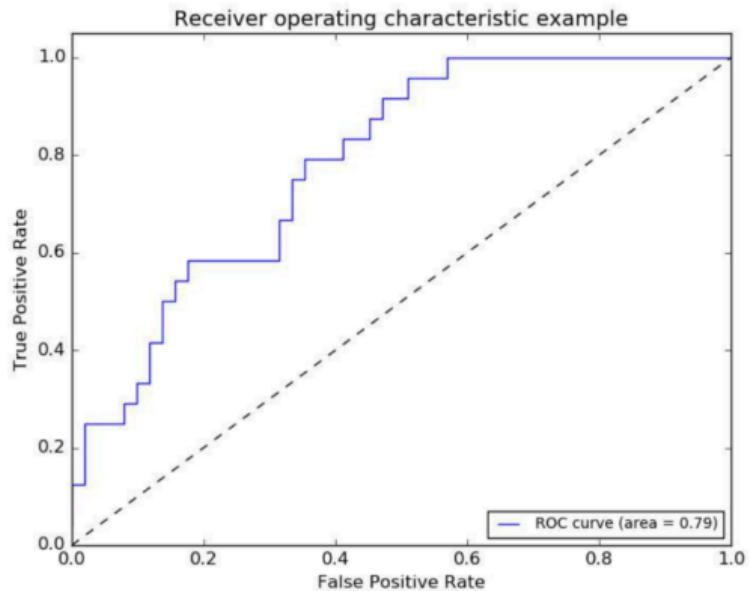
Можно сделать вывод, что метрика roc auc нечувствительна на выборках с несбалансированными классами.



# Индекс Джини

$$GINI = 2 * ROC\text{-}AUC - 1$$

По сути это площадь между ROC-кривой и диагональю соединяющей точки  $(0,0)$  и  $(1, 1)$



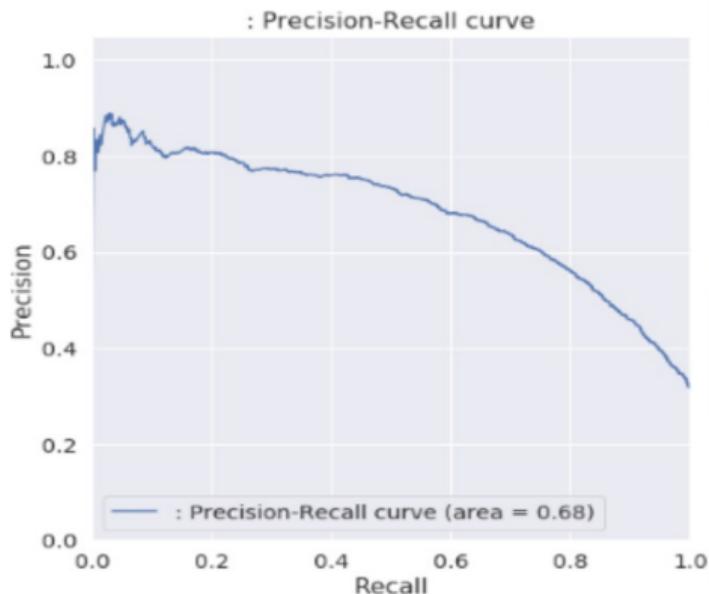


# PR кривая

$$\text{AUC-PR} = \frac{1}{\ell_+} \sum_{k=1}^{\ell} [y_k = 1] \text{precision}@k$$

Между ROC- и PR-кривой имеется тесная связь:

если ROC-кривая одного алгоритма лежит полностью над ROC-кривой другого алгоритма, то и PR-кривая одного лежит над PR- кривой другого.





# Метрика vs функция потерь

---

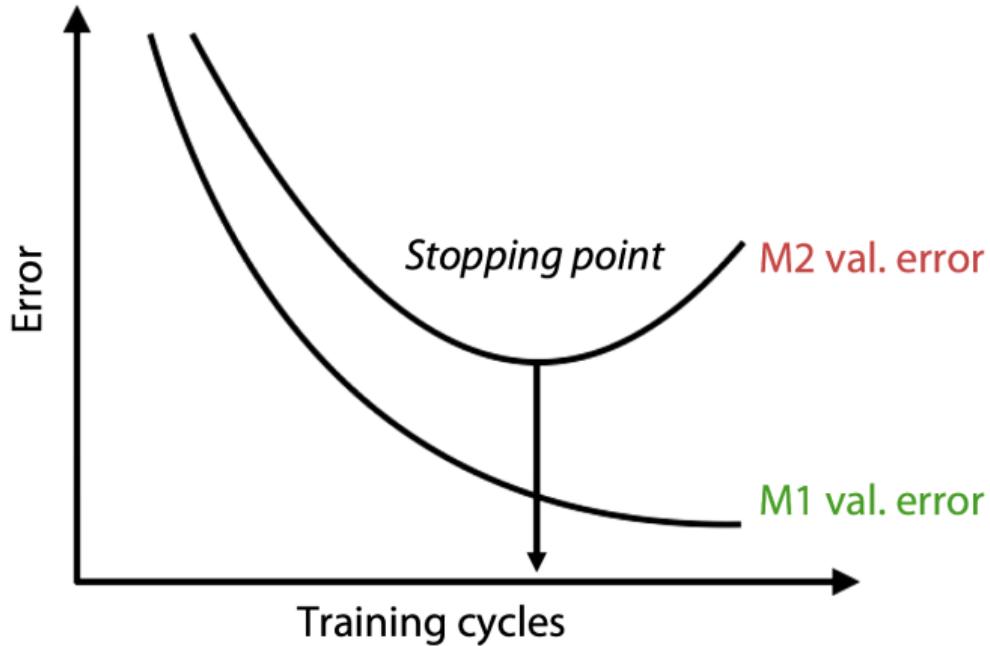
Метрика –  
это то, что мы хотим  
оптимизировать.

---

Функция потерь (loss) -  
это то, что оптимизирует  
модель.



# Оптимизация



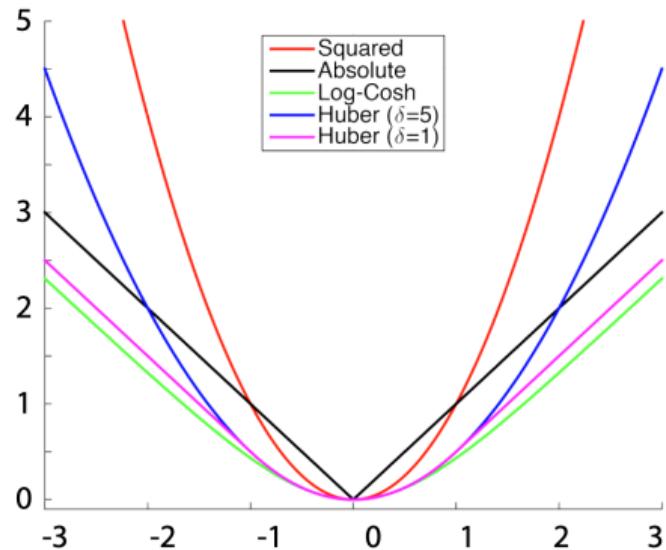


## RMSE, MSE, R-squared

$$\text{RMSE} = \sqrt{\text{MSE}} \quad R^2 = 1 - \frac{\text{MSE}}{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}$$

Оптимизируя MSE, мы оптимизируем RMSE и  $R^2$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$



Если MAE не встроен в модель,  
оптимизируем Log-Cosh или Huber



## MSPE (MAPE) as weighted MSE (MAE)

$$\text{MSPE} = \frac{100\%}{N} \sum_{i=1}^N \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2 \quad \left| \begin{array}{l} w_i = \frac{1/y_i^2}{\sum_{i=1}^N 1/y_i^2} \end{array} \right.$$

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad \left| \begin{array}{l} w_i = \frac{1/y_i}{\sum_{i=1}^N 1/y_i} \end{array} \right.$$

- Если предусмотрено, передать веса в `sample_weights`, либо перераспределить объекты (`df.sample(weights)`) train согласно весам
- Оптимизировать MSE(MAE)



# RMSLE

$$\begin{aligned}\text{RMSLE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2} = \\ &= \sqrt{MSE(\log(y_i + 1), \log(\hat{y}_i + 1))}\end{aligned}$$

- Логарифмируем таргет  $y = \log(y+1)$
- Обучаем модель на MSE
- Трансформируем предсказания  $y_{\text{pred}} = \exp(y_{\text{pred}}) - 1$



# LogLoss

$$\text{LogLoss} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

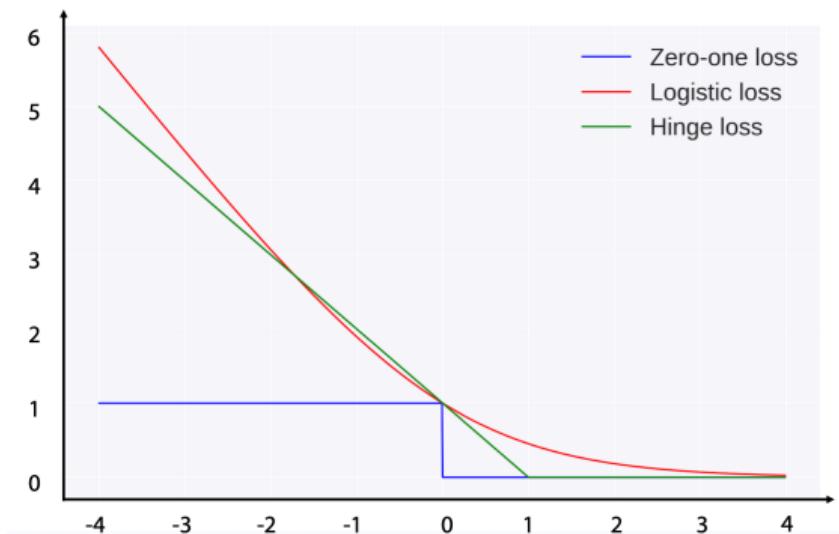
- Дифференцируем!
- Позволяет корректно предсказывать вероятности
- Есть почти во всех методах



# Accuracy

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N [\hat{y}_i = y_i]$$

- Используем другой loss,  
подбираем порог по accuracy





## ROC-AUC

- **ROC-AUC** - это вероятность того, что для пары объектов разных классов вероятность положительного класса будет выше.
- Так как это попарное сравнение, можно использовать Pairwise подход расчета loss.
- В качестве loss можно брать logloss.

$$\text{Loss} = -\frac{1}{N_0 N_2} \sum_{j:y_j=1}^{N_1} \sum_{i:y_i=0}^{N_0} \log(\text{prob}(\hat{y}_j - \hat{y}_i))$$



## Оптимизация. Итоги

MSE, LogLoss

Дифференцируемы и есть во множестве библиотек

MAE, MSPE, MAPE, RMSLE, Accuracy, ROC-AUC...

Не дифференцируемы. Оптимизируем, используя другие метрики

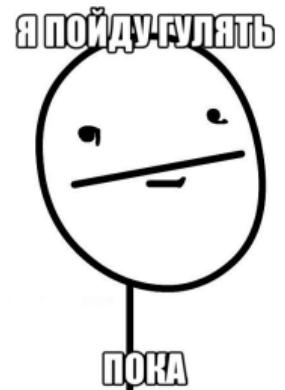
Что-то свое

Пишем свой loss и выводим градиенты



## Итоги

- Чем отличается метрика от функции потерь (loss)
- Метрики и функции потерь для регрессии и классификации
- Какие функции потерь нужно оптимизировать, чтобы оптимизировать конкретные метрики





ВОПРОСЫ?