Введение в искусственный интеллект. Машинное обучение

Лекция 3. Регрессия и оценка качества

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

МаТИС

1 марта 2019г.

План лекции

1. Регрессия
2. Оценка качества

Постановка задачи и допущения

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$

Линейная регрессия

Постановка задачи и допущения

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$, где $\theta = (\theta_0, \theta_1, ..., \theta_n)^T \in \mathbb{R}^{n+1}$ — параметры модели.

Линейная регрессия

Постановка задачи и допущения

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$, где $\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T \in \mathbb{R}^{n+1}$ — параметры модели.
- Удобно писать в векторном виде

$$a(x) = \theta^T \cdot x,$$

где $x = (x_0, x_1, \ldots, x_n)^T$ и $x_0 = 1$.

Постановка задачи и допущения

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$, где $\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T \in \mathbb{R}^{n+1}$ — параметры модели.
- Удобно писать в векторном виде

$$a(x) = \theta^T \cdot x,$$

где $x = (x_0, x_1, \ldots, x_n)^T$ и $x_0 = 1$.

Метод наименьших квадратов

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) = \sum\limits_i (\theta^T \cdot x^{(i)} - y_i)^2$ — функция потерь

Линейная регрессия

Постановка задачи и допущения

- $X = \mathbb{R}^n$, $Y = \mathbb{R}$
- $a(x) = f_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$, где $\theta = (\theta_0, \theta_1, \ldots, \theta_n)^T \in \mathbb{R}^{n+1}$ — параметры модели.
- Удобно писать в векторном виде

$$a(x) = \theta^T \cdot x,$$

где $x = (x_0, x_1, \ldots, x_n)^T$ и $x_0 = 1$.

Метод наименьших квадратов

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) = \sum\limits_i (\theta^T \cdot x^{(i)} - y_i)^2$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min\limits_\theta (L(\theta, X_{train}))$

Аналитическое решение

Теорема

Решением задачи $\arg\min\limits_{\theta}(\sum\limits_{i=1}^{\ell}(\theta^T \cdot x^{(i)} - y_i)^2)$ является $\hat{\theta} = (X^T X)^{-1} \cdot X^T \cdot y$, где $X_{i,j} = x_j^{(i)}$, $y = (y_1, ..., y_\ell)$.

Доказательство

Запишем задачу в векторном виде $||X\theta - y||^2 \to \min\limits_{\theta}$. Необходимое условие минимума в матричном виде имеет вид:

$$\frac{\partial}{\partial\theta}||X\theta - y||^2 = \frac{\partial}{\partial\theta}\Big((X\theta - y)^T \cdot (X\theta - y)\Big) = 2X^T(X\theta - y) = 0,$$

откуда получаем $X^T X\theta = X^T y$, из чего и следует требуемое.

Леммы

Определение

Пусть $\theta = (\theta_1, ..., \theta_n)$ — вектор столбец, а $z = z(\theta_1, ..., \theta_n)$. Тогда определим

$$\frac{\partial z}{\partial \theta} := (\frac{\partial z}{\partial \theta_1}, ..., \frac{\partial z}{\partial \theta_n})^T$$

Лемма 1

$$\frac{\partial}{\partial x} x^T a = a$$

Лемма 2

$$\frac{\partial}{\partial x} x^T A x = (A + A^T) x$$

Вероятностная интерпретация

Модель шума

$$y(x_i) = f_\theta(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

Метод максимума правдоподобия

$$L(\varepsilon_1, \ldots, \varepsilon_n|\theta) = \prod_i \frac{1}{\sigma_i\sqrt{2\pi}} exp(-\frac{\varepsilon_i^2}{2\sigma_i^2}) \to \max_\theta$$

$$-L(\varepsilon_1, \ldots, \varepsilon_n|\theta) = const(\theta) + \frac{1}{2}\sum_i \frac{1}{\sigma_i^2}(f_\theta(x_i) - y_i)^2 \to \min_\theta$$

Преимущества и недостатки линейной регрессии

Преимущества и недостатки линейной регрессии

Преимущества

- Простой алгоритм, вычислительно не сложный
- Линейная регрессия хорошо интерпретируемая модель
- Несмотря на свою простоту может описывать довольно сложные зависимости (например, полиномиальные)

Преимущества и недостатки линейной регрессии

Преимущества

- Простой алгоритм, вычислительно не сложный
- Линейная регрессия хорошо интерпретируемая модель
- Несмотря на свою простоту может описывать довольно сложные зависимости (например, полиномиальные)

Недостатки

- Алгоритм предполагает, что все признаки числовые
- Алгоритм предполагает, что данные распределены нормально, что не всегда так
- Алгоритм сильно чувствителен к выбросам

Гребневая регрессия

Гребневая регрессия

L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \frac{\alpha}{2}\sum_{i=0}^{n}\theta_i^2 = \sum_i(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum_{i=0}^{n}\theta_i^2$ — функция потерь

Гребневая регрессия

L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2 = \sum\limits_{i}(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min\limits_{\theta}(L(\theta, X_{train}))$

Гребневая регрессия

L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2 = \sum\limits_{i}(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min\limits_{\theta}(L(\theta, X_{train}))$

Теорема

Решением задачи $\arg\min\limits_{\theta}(\sum\limits_{i=1}^{\ell}(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2)$ является

$\hat{\theta} = (X^TX + \alpha I_n)^{-1} \cdot X^T \cdot y$, где $X_{i,j} = x_j^{(i)}$, $y = (y_1, \ldots, y_\ell)$, I_n — единичная матрица.

Гребневая регрессия

L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2 = \sum\limits_{i}(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min\limits_{\theta}(L(\theta, X_{train}))$

Теорема

Решением задачи $\arg\min\limits_{\theta}(\sum\limits_{i=1}^{\ell}(\theta^T \cdot x^{(i)} - y_i)^2 + \frac{\alpha}{2}\sum\limits_{i=0}^{n}\theta_i^2)$ является

$\hat{\theta} = (X^T X + \alpha I_n)^{-1} \cdot X^T \cdot y$, где $X_{i,j} = x_j^{(i)}$, $y = (y_1, \ldots, y_\ell)$, I_n — единичная матрица.

Доказательство

Аналогично, что и требовалось доказать.

LASSO

L1-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \alpha \sum\limits_{i=0}^{n} |\theta_i| = \sum\limits_{i} (\theta^T \cdot x^{(i)} - y_i)^2 + \alpha \sum\limits_{i=0}^{n} |\theta_i|$ — функция потерь

LASSO

L1-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + \alpha \sum\limits_{i=0}^{n} |\theta_i| = \sum\limits_{i} (\theta^T \cdot x^{(i)} - y_i)^2 + \alpha \sum\limits_{i=0}^{n} |\theta_i|$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min\limits_{\theta} (L(\theta, X_{train}))$

Свойства

- Эта регуляризация обеспечивает отбор признаков

Elastic Net

L1-регуляризация и L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + r\alpha \sum_{i=0}^{n} |\theta_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} \theta_i^2 = \sum_i (\theta^T \cdot x^{(i)} - y_i)^2 + r\alpha \sum_{i=0}^{n} |\theta_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} \theta_i^2$ — функция потерь

L1-регуляризация и L2-регуляризация

- $L(\theta, X_{train}) = MSE(\theta, X_{train}) + r\alpha \sum_{i=0}^{n} |\theta_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} \theta_i^2 =$
  $\sum_i (\theta^T \cdot x^{(i)} - y_i)^2 + r\alpha \sum_{i=0}^{n} |\theta_i| + (1-r)\frac{\alpha}{2} \sum_{i=0}^{n} \theta_i^2$ — функция потерь
- Задача найти $\hat{\theta} = \arg\min_{\theta}(L(\theta, X_{train}))$

Свойства

- Нет аналитического решения

Заключение

- Линейная регрессия — простая, хорошо интерпретируемая модель, не устойчивая к выбросам
- Имеет наглядную вероятностную интерпретацию
- Регуляризация — отличный способ борьбы с переобучением и шумом в данных

Классификация ответов бинарного классификатора

- Задача классификации на 2 класса: $X \to Y, Y = \{+1, -1\}$
- Алгоритм классификации $a(x_i) = y_i$
- Класс с меткой "+1" называется "**positive**"
- Класс с меткой "-1" называется "**negative**"

Классификация ответов бинарного классификатора

- Задача классификации на 2 класса: $X \to Y$, $Y = \{+1, -1\}$
- Алгоритм классификации $a(x_i) = y_i$
- Класс с меткой "+1" называется "**positive**"
- Класс с меткой "-1" называется "**negative**"

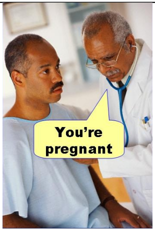
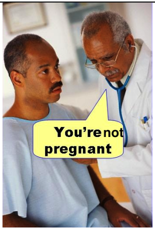
	Выход алгоритма	Правильный ответ
TP (True Positive)	$a(x_i) = +1$	$y_i = +1$
TN (True Negative)	$a(x_i) = -1$	$y_i = -1$
FP (False Positive)	$a(x_i) = +1$	$y_i = -1$
FN (False Negative)	$a(x_i) = -1$	$y_i = +1$

Матрица ошибок

Более наглядно эти соотношения можно изобразить с помощью **матрицы ошибок (confusion matrix)**

		Правильный ответ	
		$y = +1$	$y = -1$
Выход алгоритма	$a(x) = +1$	True Positive	False Positive (Ошибка 1 рода)
	$a(x) = -1$	False Negative (Ошибка 2 рода)	True Negative

Матрица ошибок

<table>
<tr><th></th><th>$y = +1$</th><th>$y = -1$</th></tr>
<tr><td>$a(x) = +1$</td><td>
</td><td>
</td></tr>
<tr><td>$a(x) = -1$</td><td>
</td><td>
</td></tr>
</table>

Простейшая метрика качества

- Простейшая метрика качества - это доля правильных ответов на тесте (контрольной выборке)
- По-английски - **Accuracy**

Формула Accuracy

$$Accuracy = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

Простейшая метрика качества

- Простейшая метрика качества - это доля правильных ответов на тесте (контрольной выборке)
- По-английски - **Accuracy**

Формула Accuracy

$$Accuracy = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) = y_i] = \frac{TP+TN}{TP+FP+TN+FN}$$

Недостаток

- Не учитывается дисбаланс классов
- Не учитывается цена ошибки на объектах разных классов

Метрики по положительному отклику алгоритма

Рассмотрим метрики, которые основаны на подсчёте доли положительных ответов алгоритма.

Доля ложных положительных классификаций

Также известно как False Positive Rate, или **FPR**.

$$FPR(a, X^\ell) = \frac{\sum_{i=1}^{\ell}[y_i{=}-1][a(x_i){=}+1]}{\sum_{i=1}^{\ell}[y_i{=}-1]}$$

Метрики по положительному отклику алгоритма

Рассмотрим метрики, которые основаны на подсчёте доли положительных ответов алгоритма.

Доля ложных положительных классификаций

Также известно как False Positive Rate, или **FPR**.

$FPR(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = -1][a(x_i) = +1]}{\sum_{i=1}^{\ell} [y_i = -1]}$

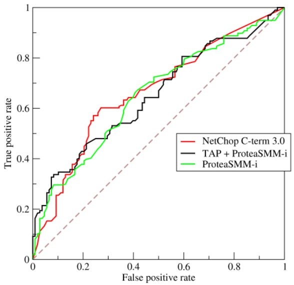
Доля верных положительных классификаций

Также известно как True Positive Rate, или **TPR**.

$TPR(a, X^\ell) = \frac{\sum_{i=1}^{\ell} [y_i = +1][a(x_i) = +1]}{\sum_{i=1}^{\ell} [y_i = +1]}$

Замечание. Обратите внимание на разные знаменатели!

Наиболее известна как рабочая характеристика приёмника, или Receiver Operating Characteristic (**ROC-кривая**), в который мы смотрим на компромисс между уровнем ложной тревоги и долей верного отклика.



По оси X откладывается FPR, по оси Y - TPR[1].
Замечание. На данной кривой никак не учитываются пропуски.

[1] https://wikipedia.org

AUROC

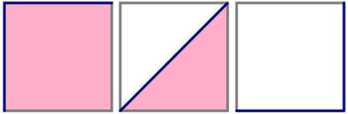
Чем больше для каждого значения ошибки FPR значение правильного предсказания TPR, тем лучше работает классификатор.
Т.о., площадь под кривой (Area Under Curve, AUC / AUROC) необходимо максимизировать.

[2] https://dyakonov.org

AUROC

Чем больше для каждого значения ошибки FPR значение правильного предсказания TPR, тем лучше работает классификатор.
Т.о., площадь под кривой (Area Under Curve, AUC / AUROC) необходимо максимизировать.

Наглядны ROC-кривые для наилучшего (AUC=1), случайного (AUC=0.5) и наихудшего (AUC=0) алгоритма[2].



[2] https://dyakonov.org

Задача

Предположим, что алгоритм бинарной классификации $a(x_i)$ принимает решение о присвоении класса на основе некоторого скалярного значения $g_\theta(x_i) \in \mathbb{R}$, где θ - набор параметров модели, а $g_\theta(x_i)$ - дискриминантная функция.

Задача

- Хотим построить ROC-кривую, т.е. найти точки $\{(FPR_i, TPR_i)\}_{i=1}^{\ell}$
- Подсчитать площадь под кривой - AUROC

Задача

Предположим, что алгоритм бинарной классификации $a(x_i)$ принимает решение о присвоении класса на основе некоторого скалярного значения $g_\theta(x_i) \in \mathbb{R}$, где θ - набор параметров модели, а $g_\theta(x_i)$ - дискриминантная функция.

Задача

- Хотим построить ROC-кривую, т.е. найти точки $\{(FPR_i, TPR_i)\}_{i=1}^{\ell}$
- Подсчитать площадь под кривой - AUROC

Подсчитаем количество правильных ответов разного типа:

- $\ell_+ = \sum_{i=1}^{\ell} [y(x_i) = +1]$
- $\ell_- = \sum_{i=1}^{\ell} [y(x_i) = -1]$ (понятно, что $\ell = \ell_+ + \ell_-$)

Упорядочим обучающую выборку X^ℓ по убыванию значений $g_\theta(x_i)$.
Тогда формула для $AUROC = \frac{1}{\ell_-} \sum_{i=1}^{\ell} [y_i = -1] TPR_i$.

Решение задачи

Алгоритм

Первую точку ставим в начало координат: $(FPR_0, TPR_0) = (0, 0), AUROC = 0$.

Решение задачи

Алгоритм

Первую точку ставим в начало координат: $(FPR_0, TPR_0) = (0, 0), AUROC = 0$.

Цикл по упорядоченной выборке $i = 1 \dots \ell$

Если $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{\ell_-}, TPR_{i-1})$ (двигаемся по оси X)
- $AUROC = AUROC + \frac{1}{\ell_-} TPR_i$

Решение задачи

Алгоритм

Первую точку ставим в начало координат: $(FPR_0, TPR_0) = (0, 0), AUROC = 0$.

Цикл по упорядоченной выборке $i = 1 \dots \ell$

Если $y_i = -1$:

- $(FPR_i, TPR_i) = (FPR_{i-1} + \frac{1}{\ell_-}, TPR_{i-1})$ (двигаемся по оси X)
- $AUROC = AUROC + \frac{1}{\ell_-} TPR_i$

Если $y_i = +1$:

- $(FPR_i, TPR_i) = (FPR_{i-1}, TPR_{i-1} + \frac{1}{\ell_+})$ (двигаемся по оси Y)

Другие важные метрики 1

В задачах информационного поиска

- Точность, или $Precision = \frac{TP}{TP+FP}$ (доля релевантных объектов среди найденных)
- Полнота, или $Recall = \frac{TP}{TP+FN}$ (доля найденных объектов среди релевантных)

Другие важные метрики 1

В задачах информационного поиска

- Точность, или $Precision = \frac{TP}{TP+FP}$ (доля релевантных объектов среди найденных)
- Полнота, или $Recall = \frac{TP}{TP+FN}$ (доля найденных объектов среди релевантных)

Как применяются

- **Точность**: позволяет следить, чтобы было мало ложных тревог; но при этом ничего не говорит о пропусках (высока цена ложной тревоги, а цена пропуска - низкая).
- **Полнота**: позволяет следить, чтобы было мало пропусков; но при этом ничего не говорит о ложных тревогах (высока цена пропуска, а цена ложной тревоги - низкая).

Замечание. Зачастую задача состоит в оптимизации одной метрики при фиксации другой.

Другие важные метрики 2

В задачах медицинской диагностики

- Чувствительность, или $Sensitivity = \frac{TP}{TP+FN}$ (доля верных положительных диагнозов)
- Специфичность, или $Specificity = \frac{TN}{TN+FP}$ (доля верных отрицательных диагнозов)

Другие важные метрики 2

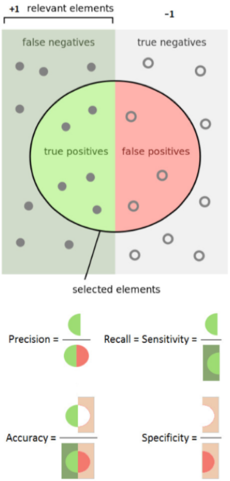
В задачах медицинской диагностики

- Чувствительность, или $Sensitivity = \frac{TP}{TP+FN}$ (доля верных положительных диагнозов)
- Специфичность, или $Specificity = \frac{TN}{TN+FP}$ (доля верных отрицательных диагнозов)

Как применяются

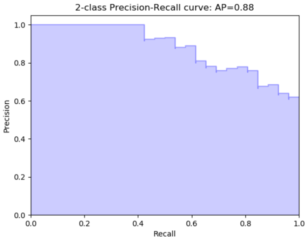
- **Чувствительность**: максимизируем количество верных положительных диагнозов, но не учитываем ложные диагнозы (стоимость лечения низкая, а цена пропуска - высокая).
- **Специфичность**: максимизируем количество верных отрицательных диагнозов, но не учитываем пропуски диагноза (стоимость лечения высокая, а цена пропуска - низкая).

Иллюстрация метрик



Агрегированные метрики над Precision-Recall

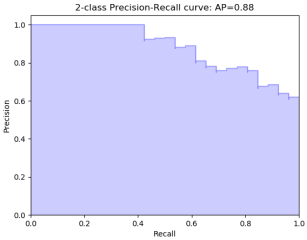
Можно построить кривую Точность-Полнота (PR-кривая) по аналогии с ROC-кривой:



Замечание. Обратите внимание, что в данном случае кривая не обязательно монотонна!

Агрегированные метрики над Precision-Recall

Можно построить кривую Точность-Полнота (PR-кривая) по аналогии с ROC-кривой:



Замечание. Обратите внимание, что в данном случае кривая не обязательно монотонна!

AUPRC

- Аналогично AUROC, можно вычислить площадь под PR-кривой - AUPRC
- Другое название - Average Precision (с некоторым допущениями на способ интегрирования): чем больше, тем лучше

Многоклассовая классификация

Для каждого класса $c \in Y$ обозначим через TP_c, FP_c и FN_c верные положительные, ложные положительные и ложные отрицательные ответы. Тогда:

Точность и полнота с макроусреднением

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Не чувствительно к ошибкам на маленьких классах

Многоклассовая классификация

Для каждого класса $c \in Y$ обозначим через TP_c, FP_c и FN_c верные положительные, ложные положительные и ложные отрицательные ответы. Тогда:

Точность и полнота с макроусреднением

- $Precision = \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}$
- $Recall = \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}$
- Не чувствительно к ошибкам на маленьких классах

Точность и полнота с микроусреднением

- $Precision = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FP_c}$
- $Recall = \frac{1}{|Y|} \sum_c \frac{TP_c}{TP_c + FN_c}$
- Чувствительно к ошибкам на маленьких классах

Резюме по оценкам качества классификации

- Точность и полнота подходят для задач информационного поиска, когда доля объектов релевантного класса мала

Резюме по оценкам качества классификации

- Точность и полнота подходят для задач информационного поиска, когда доля объектов релевантного класса мала
- Чувствительность и специфичность подходят для задач с несбалансированными классами (как, например, в медицине)

Резюме по оценкам качества классификации

- Точность и полнота подходят для задач информационного поиска, когда доля объектов релевантного класса мала
- Чувствительность и специфичность подходят для задач с несбалансированными классами (как, например, в медицине)
- AUROC подходит для оценки качества при нефиксированном соотношении цены ошибок

Резюме по оценкам качества классификации

- Точность и полнота подходят для задач информационного поиска, когда доля объектов релевантного класса мала
- Чувствительность и специфичность подходят для задач с несбалансированными классами (как, например, в медицине)
- AUROC подходит для оценки качества при нефиксированном соотношении цены ошибок
- Ещё одна агрегированная оценка качества - F-мера:
  $F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
  - Это *гармоническое среднее*, которое стремится к нулю когда хотя бы одно из значений стремится к нулю