

## ГЛАВА 8.

### Метрики качества

*Если не можешь что-то измерить,  
то не сможешь это улучшить.*

Лорд Кельвин

Построить какой-нибудь алгоритм машинного обучения не так сложно, например, при прогнозе спроса на товары интернет-магазина можно считать, что на следующей неделе каждого вида товара будет куплено ровно столько, сколько и на этой. После того, как алгоритм построен, возникает естественный вопрос: можно ли его улучшить? Для этого надо уметь измерять качество (performance) работы алгоритма, чтобы было понятно, что улучшать.

Простая эвристика «завтра будет также, как вчера» очень эффективна. Продвинутое методы машинного обучения часто улучшают её лишь не намного.

#### Качество работы алгоритма

Здесь и далее считаем, что есть некоторая выборка  $x_1, \dots, x_m$  с известными (м.б. не на этапе обучения алгоритма) значениями целевого признака (метками)

$$y_1 = y(x_1), \dots, y_m = y(x_m).$$

В данном случае не важно обучающая это выборка или контрольная – мы на ней будем измерять качество ответов алгоритма  $a(x)$ . Как отмечается в [главе «Контроль»](#) качество лучше оценивать на контрольной выборке, но чтобы его минимизировать на обучающей, его всё равно надо уметь вычислять. Считаем, что на этой выборке алгоритм выдал ответы

$$a_1 = a(x_1), \dots, a_m = a(x_m).$$

Ясно, что надо придумать функцию, которая измеряет схожесть полученных ответов и истинных. Есть два вида таких функций:

- **функционалы качества** (измеряют описанную схожесть),
- **функции ошибки** (измеряют различие между правильным ответом и полученным).

Из семантики понятно, что функционал качества максимизируют при обучении (настройке параметров алгоритма), а функцию ошибки, наоборот, минимизируют. В английском варианте есть общее понятие **метрики качества (metrics)**, которое объединяет перечисленные выше понятия. В русском языке термин «метрика» и так очень нагружен, но тоже применяется в смысле оценки качества.

Ниже перечислим стандартные метрики качества, часто будем указывать оптимальный алгоритм в классе константных, поскольку

- его часто можно найти аналитически,
- с помощью него иллюстрируются важные особенности функции ошибки,
- хорошее решение может строиться из таких константных решений (например, случайный лес – это сумма решающих деревьев, каждое из которых является кусочно-константной функцией).

Отметим ещё одну особенность употребления терминов. Под качеством алгоритма нам бы хотелось понимать качество его работы (в среднем или худшем случае) при его эксплуатации. Поскольку мы часто лишены возможности быстро и эффективно оценить работу алгоритма на данных, которых, быть может, ещё и нет – они появятся в будущем, под качеством понимают **качество ответа** на конкретной выборке. Именно поэтому все рассматриваемые функционалы качества и функции ошибки имеют вид:

$$L((y_1, \dots, y_m), (a_1, \dots, a_m)).$$

Очень часто, но далеко не всегда, функция ошибки на выборке представляется в виде сумм ошибок на отдельных объектах:

$$\frac{1}{m} \sum_{i=1}^m L(y_i, a_i)$$

(здесь для обозначения ошибки на выборке и на отдельном объекте мы используем одно и то же обозначение  $L$ , что не должно вызвать путаницы).

## Раздел 8.1. Функции ошибки в задачах регрессии

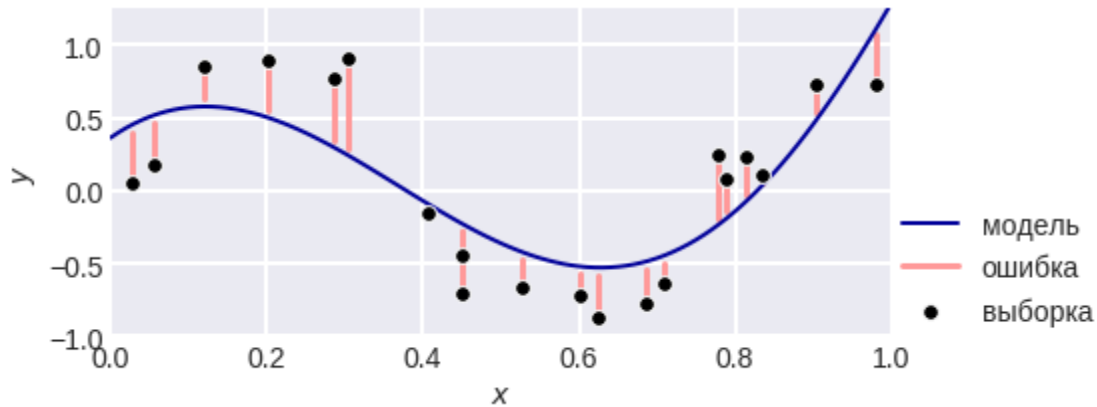


Рис. 8.1. Иллюстрация задачи регрессии.

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

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

Полезно помнить (см. главу «Средние»), что оптимальный константный алгоритм с точки зрения этой функции ошибки:

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

т.е. решением задачи  $\text{MAE} \rightarrow \min$  в классе констант является медиана, см. рис. 8.2.

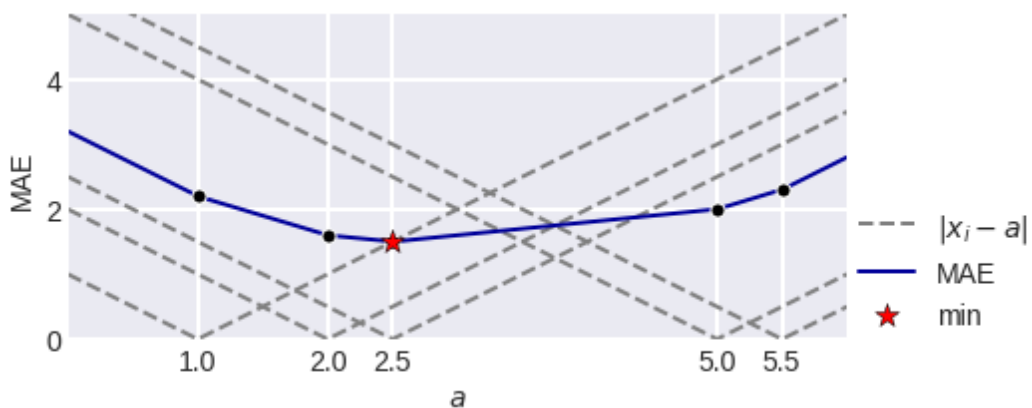
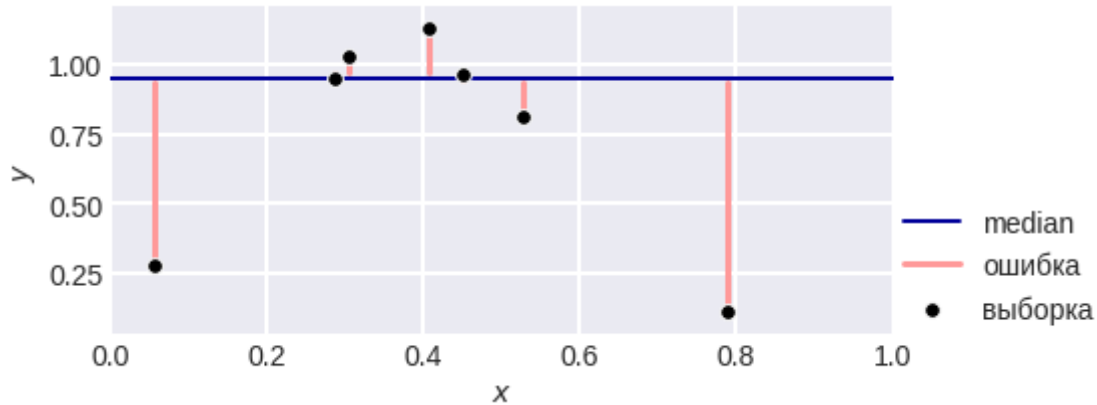


Рис. 8.2. График MAE от константного ответа  $a$  в модельной задаче.

В задачах регрессии с МАЕ при ансамблировании часто вместо усреднения нескольких алгоритмов берут их медиану – это, как правило, повышает качество. Если в такой задаче целевой признак целочисленный, то округление ответа часто не ухудшает качество решения.



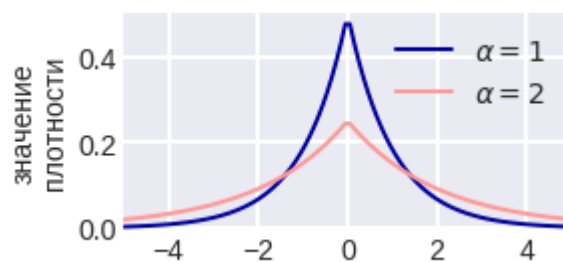
**Рис. 8.3. Решение в классе константных алгоритмов.**

Приведём теоретическое обоснование разумности использования МАЕ как функции ошибки. Пусть целевой признак описывается моделью

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

с точностью до шума  $\varepsilon \sim \text{laplace}(0, \alpha)$ , которое распределено по Лапласу, см. рис. 8.4. Для оценки параметров  $w$  модели  $a_w(x)$  выписываем правдоподобие модели

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



**Рис. 8.4. Плотность распределения Лапласа.**

Применение метода максимального правдоподобия (ММП) соответствует максимизации:

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

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

или (отбрасывая не влияющую на решение константу)

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

Таким образом, максимизация правдоподобия эквивалентна минимизации МАЕ. Заметим, что если бы в каждой точке была бы своя шибка:

$$y_i = a_w(x_i) + \varepsilon_i, \varepsilon_i \sim \text{laplace}(0, \alpha_i),$$

то мы бы получили **взвешенный вариант МАЕ (weighted MAE)**:

$$\sum_{i=1}^m \alpha_i |y_i - a_w(x_i)| \rightarrow \min ,$$

проведите соответствующие выкладки. Отметим, что мы не делали никаких предположений о природе модели, только о распределении ошибок.

Вот почему рекомендуют строить т.н. «Residual Plots» – важно знать распределение ошибок!

**Средний квадрат отклонения (MSE – Mean Squared Error):**

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

или корень из этой ошибки: **RMSE – Root Mean Squared Error** или **RMSD – Root Mean Square Deviation**

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

Также часто используют т.н. **коэффициент детерминации (R<sup>2</sup>)**:

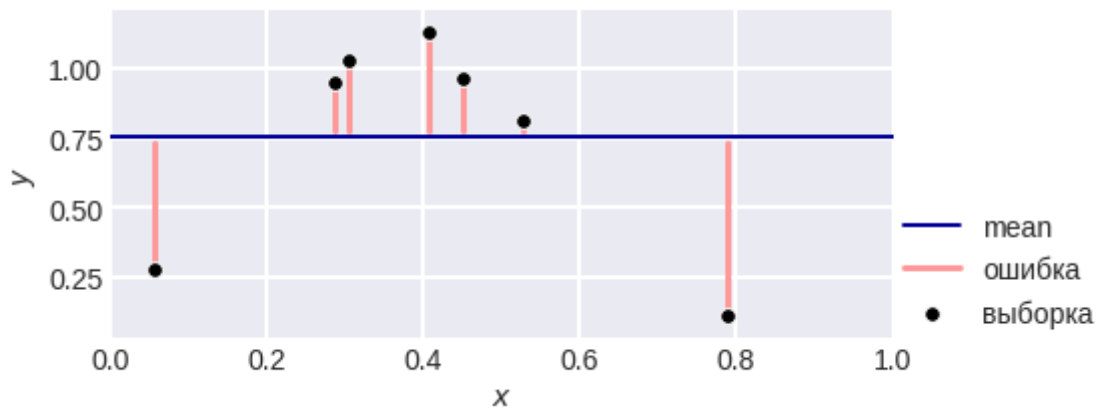
$$R^2 = 1 - \frac{\sum_{i=1}^m |a_i - y_i|^2}{\sum_{i=1}^m |\bar{y} - y_i|^2}, \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$$

Понятно, что с точки зрения оптимизации эти функции эквивалентны. MSE совсем простая функция, но её

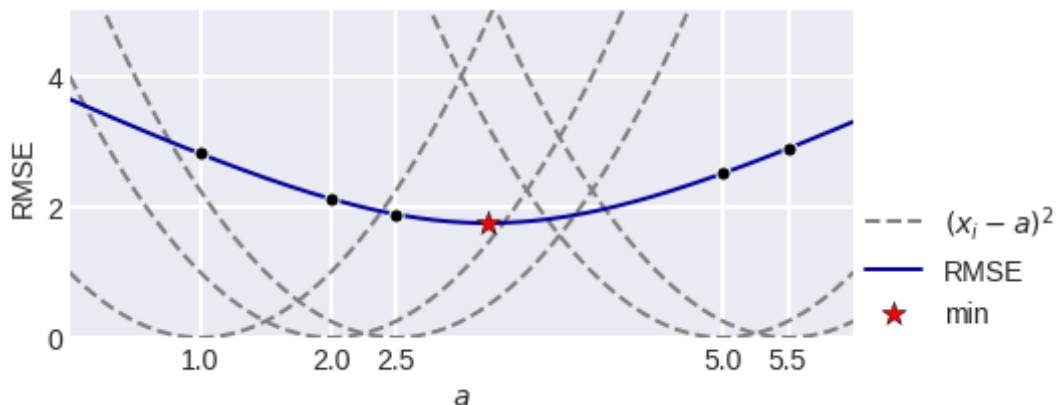
Хорошо или плохо, если RMSE=27?

значения сложно интерпретировать, поэтому в RMSE значение приводится в масштаб изменений целевого вектора, но сложности интерпретации остаются. Оптимальное решение в задаче минимизации MSE и RMSE в классе констант – среднее арифметическое

$$\bar{y} = \arg \min \frac{1}{m} \sum_{i=1}^m |a - y_i|^2,$$



**Рис. 8.5. Решение в классе константных алгоритмов.**



**Рис. 8.6. График RMSE от константного ответа \$a\$ в модельной задаче.**

поэтому в коэффициенте детерминации происходит нормировка: MSE-ошибка алгоритма делится на MSE-ошибку оптимального константного алгоритма. Коэффициент детерминации, который по смыслу является функционалом качества, принимает максимальное значение 1 в случае абсолютно точного ответа, значение 0 для оптимального константного ответа, но может быть и меньше нуля, если решение хуже такого константного. Кстати, коэффициент детерминации широко используется в статистике, и в общем случае определяется так:

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

т.е. единица минус доля дисперсии случайной ошибки модели (или условной по факторам дисперсии зависимой переменной) в дисперсии зависимой переменной.

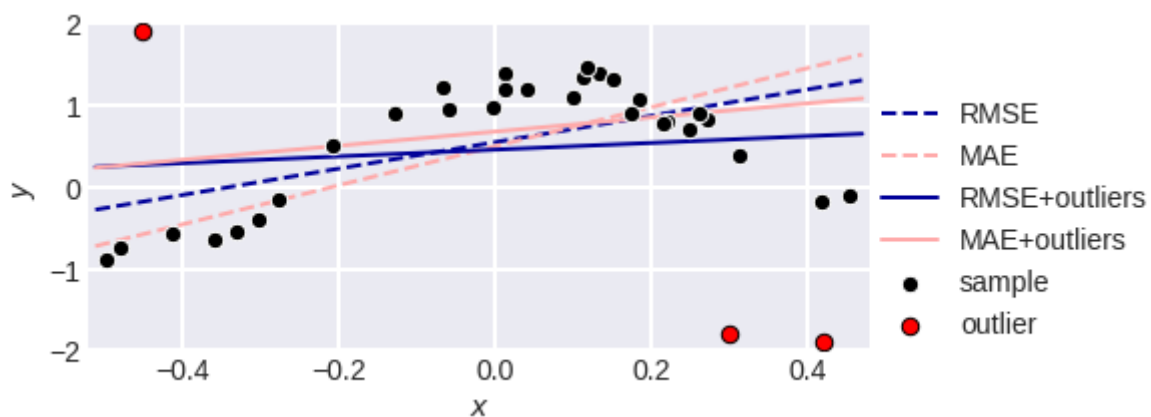
Заметим, что не смотря на то, что минимумы MSE, RMSE и максимум  $R^2$  достигаются в одной точке, с точки зрения практической оптимизации, например методом градиентного спуска, эти функции не эквивалентны, поскольку у них разные производные, например

Хороший вопрос: какую функцию «выгоднее» оптимизировать на практике?

$$\frac{\partial \text{MSE}}{\partial a} = \frac{2}{m} \sum_{i=1}^m (a - y_i), \quad \frac{\partial \text{RMSE}}{\partial a} = \frac{1}{m \text{RMSE}} \sum_{i=1}^m (a - y_i),$$

т.е. различаются в  $2/\text{RMSE}$  раз. Отметим, что модуль производной MAE не стремится к нулю при приближении к точке минимума MAE (в отличие от MSE).

Мы уже знаем, что среднее арифметическое не устойчиво к выбросам, в отличие от медианы. Это же справедливо в общем случае: если мы настраиваем параметры нашего алгоритма минимизируя (R)MSE, то они существенно зависят от выбросов. Для MAE такого не наблюдается. На рис. 8.7 показано, как будет выглядеть линейная модель, если её настраивать на (R)MSE / MAE на «хорошей» выборке и выборке с выбросами. При добавлении выбросов в выборку коэффициенты прямой сильнее изменятся, если минимизировать (R)MSE.



**Рис. 8.7. Линейная регрессия в зависимости от минимизируемой ошибки и наличия выбросов.**

Большинство алгоритмов регрессии явно или неявно ориентированы именно на минимизацию (R)MSE, например, линейная (гребневая) регрессия или случайный лес. При ансамблировании в задачах регрессии с (R)MSE лучше использовать «обычное» усреднение базовых алгоритмов (с помощью среднего арифметического). Отметим также, что в отличие от MSE и RMSE,  $R^2$  несимметричная функция:

$$R^2((y_i)_{i=1}^m, (a_i)_{i=1}^m) \neq R^2((a_i)_{i=1}^m, (y_i)_{i=1}^m).$$

Теоретическое обоснование (R)MSE подобно MAE, только теперь полагаем, что

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

с точностью до нормально распределённого шума  $\varepsilon \sim \text{norm}(0, \sigma^2)$ , рис. 8.8. Для оценки параметров  $w$  модели  $a_w(x)$  выписываем правдоподобие модели

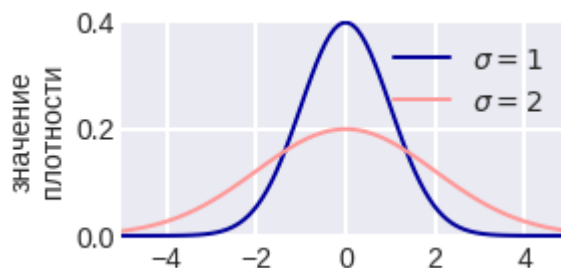
$$p(y | x, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y - a_w(x))^2}{2\sigma^2}\right]$$

и применяем метод максимального правдоподобия:

$$\begin{aligned} \log L(w) &= \log \prod_{i=1}^m p(y_i | x_i, w) = \\ &= \sum_{i=1}^m \left[ -\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_i - a_w(x_i))^2}{2\sigma^2} \right] \rightarrow \max \end{aligned}$$

Откуда получаем задачу минимизации MSE:

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



**Рис. 8.8.** Плотность нормального распределения.



Аналогично взвешенному МАЕ здесь можно вывести **взвешенную среднеквадратичную ошибку (weighted MSE)**.

Есть ещё одно интересное (и не вероятностное!) обоснование среднеквадратичной ошибки. Пусть функция ошибки представима в виде  $l(y, a) = g(y - a)$ , что вполне логично: зависит от отклонения истинного значения от нашего ответа. Что ещё логично?

- 1)  $g(0) = 0$  – если ответ верный, то ошибка нулевая,
- 2)  $|z_1| \leq |z_2| \Rightarrow g(z_1) \leq g(z_2)$  – чем больше отклонение, тем больше ошибка, можно также сразу предположить, что функция  $g$  чётная,
- 3) функция  $g$  достаточно гладкая и представима в виде ряда Маклорена:

$$g(z) = g(0) + g'(0)z + \frac{g''(0)}{2}z^2 + o(z^2)$$

Но тогда, пренебрегая последнем остаточным членом в этом ряде

$$\begin{aligned} l(y, a) = g(y - a) &\approx \underbrace{g(0)}_{=0(1)} + \underbrace{g'(0)(y - a)}_{=0(2)} + \frac{g''(0)}{2}(y - a)^2 = \\ &= \underbrace{C}_{>0} (y - a)^2 \end{aligned}$$

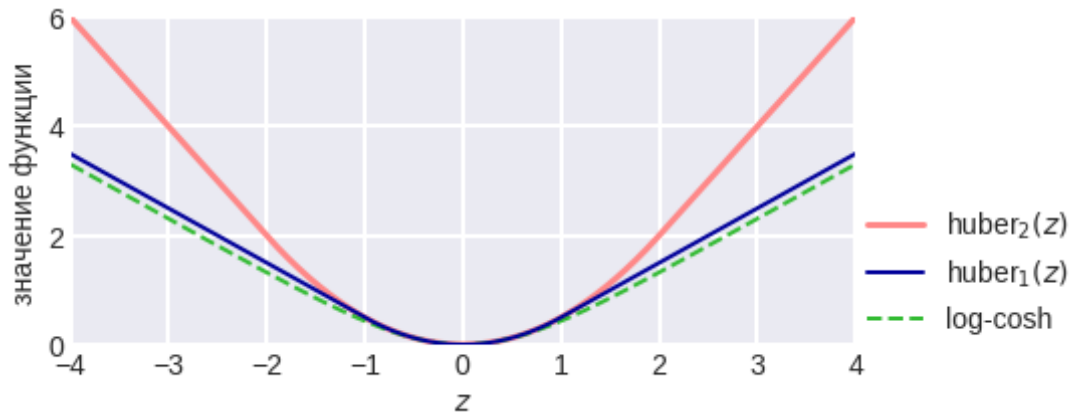
Здесь мы воспользовались нашими предположениями (1) и (2). Таким образом, с точностью до константы, когда значения  $y$  и  $a$  близки (формула Маклорена верна ведь в малой окрестности нуля и мы пренебрегли остаточным членом), разумно использовать средний квадрат отклонения. Именно это делается в **функции ошибки Хьюбера (Huber loss)**:

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

В окрестности нуля её график является параболой (см. рис. 8.9), а при больших значениях аргумента она линейная (что делает её «устойчивой к выбросам», аналогично МАЕ). Отметим, что эта функция ошибки имеет уже параметр –  $\delta$ , выбор которого тоже должен быть обоснован (он как раз и задаёт границу

перехода MSE ошибки в MAE). Непараметрическая функция, которая похоже на функцию Хьюбера, но используется существенно реже –

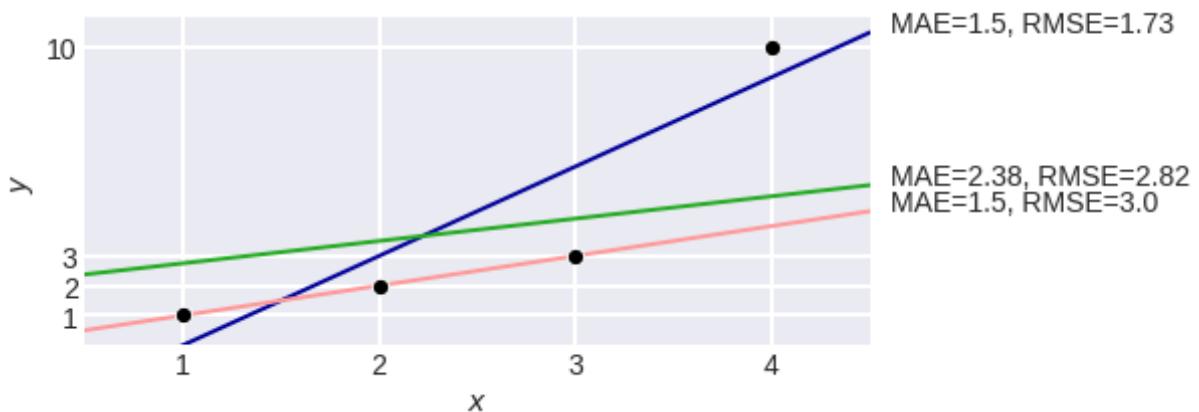
$$\text{logcosh} = \log\left(\frac{\exp(z) + \exp(-z)}{2}\right).$$



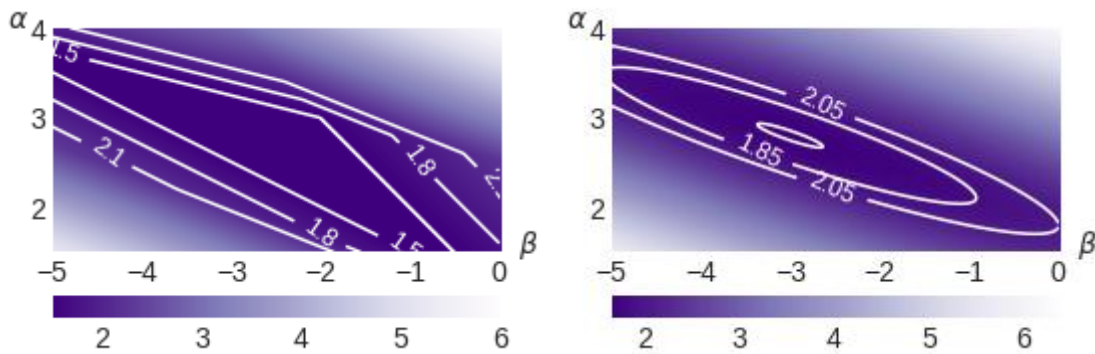
**Рис. 8.9. Функция ошибок Хьюбера и log-cosh.**

Для сравнения функций MAE и RMSE посмотрим ещё на следующую простую модельную задачу, см. рис. 8.10. На рис. 8.11 цветом показаны значения функций ошибок в зависимости от значения двух параметров прямой. Заметим, что минимальное значение MAE – 1.5 и оно достигается для всех пар параметров внутри треугольника с меткой «1.5» на рис. 8.11 (слева), минимальное значение RMSE достигается, естественно, в одной точке (т.к. MSE – строго выпуклая функция). На рис. 8.10 видно, что минимальное значение MAE может быть у совершенно разных решений (при точной настройке на трёх объектах и при настройке на выброс).

Не противоречит ли это нечувствительности MAE к выбросам?



**Рис. 8.10. Модельная задача регрессии.**



**Рис. 8.11. Линии уровней функции ошибок MAE (слева) и RMSE (справа) в модельной задаче.**

Обобщения MAE и RMSE можно записать формулой

$$\sqrt[p]{\frac{1}{m} \sum_{i=1}^m w_i |\varphi(a_i) - \varphi(y_i)|^p}.$$

Наличие функций  $\varphi$  (как правило, используется логарифм) не должно смущать: преобразовав с помощью этой функции целевой вектор мы переходим к задаче с функцией ошибки без  $\varphi$ . Осталось не забыть после настройки модели и получения ответов, подвергнуть их обратному преобразованию:

$$\varphi^{-1}(a(x)).$$

Такие функции деформации  $\varphi$  используют, чтобы «привести регрессионные метки в более равномерную шкалу», см. главу «Визуализация» (например, чтобы ошибки на объектах с большими значениями меток не сильно портили решение). Значения  $w_i$ , по сути, являются весами объектов. Некоторые методы обучения позволяют **в явном виде задать веса объектов** (при настройке модели). В методах, основанных на сэмплинге (взятии подвыборок обучающей выборки), можно **проводить сэмплинг с вероятностями включения объектов в подвыборку, пропорциональными указанным весам**.

Способ сэмплинга – это тоже параметр настройки модели!

Чуть сложнее с нетривиальными значениями  $p$ . Скажем, при  $p=3$  можно сначала обучить алгоритм для функции ошибки при  $p=2$ , т.е. учитывая предыдущие замечания, модель настраивается на RMSE, быть может, с деформированным целевым признаком и весами объектов. После настройки

можно в окрестности полученных параметров модели поискать меньшее значение исходной функции ошибки.

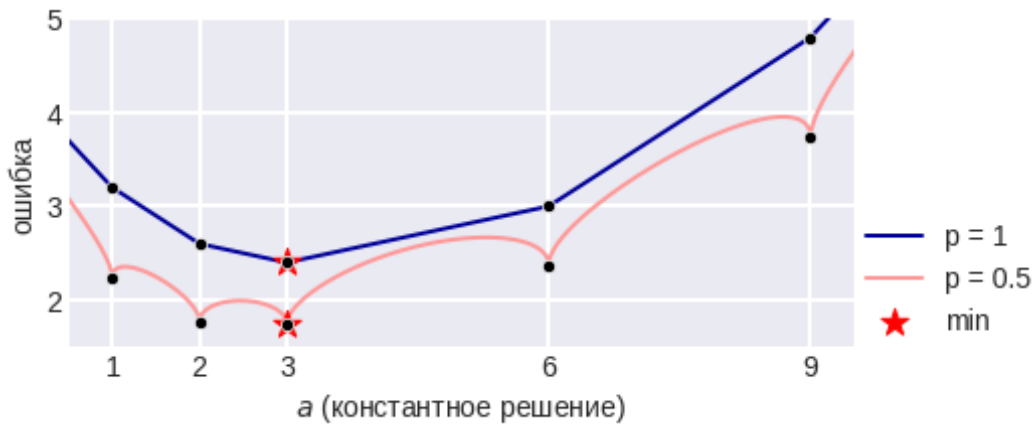


Рис. 8.12. Обобщённая ошибка константных решений для меток 1, 2, 3, 6, 9.

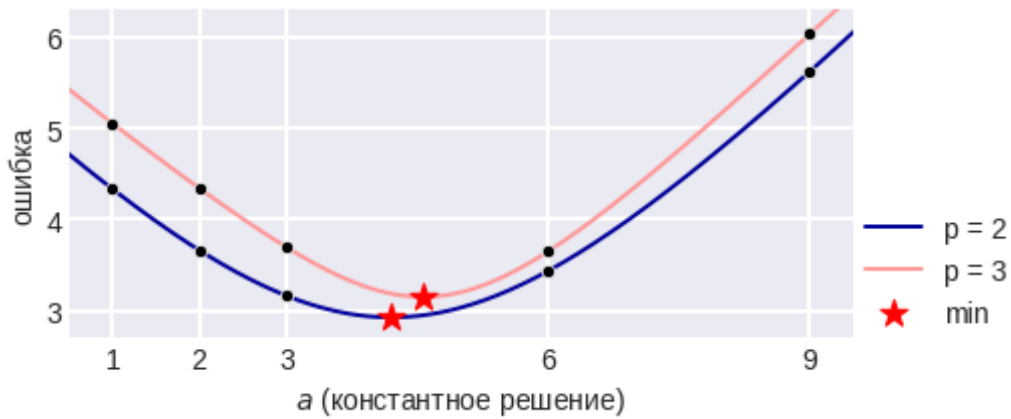


Рис. 8.13. Обобщённая ошибка константных решений для меток 1, 2, 3, 6, 9.

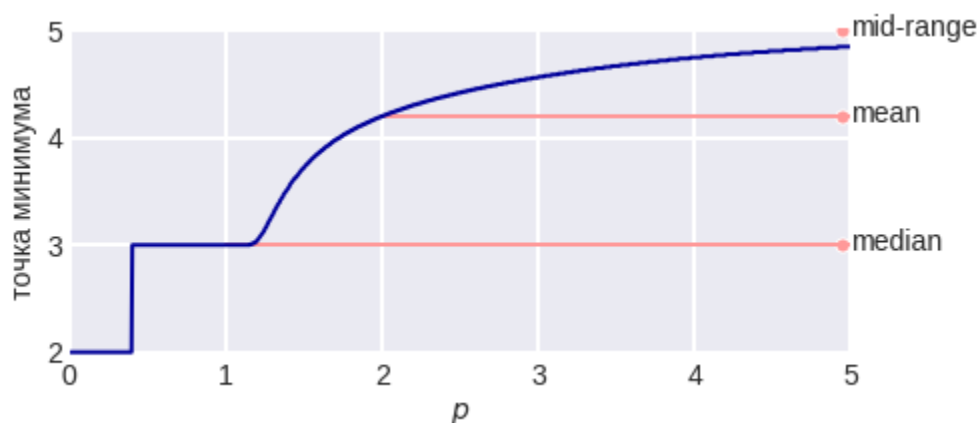


Рис. 8.14. Зависимость оптимального константного решения от степени.

Рассмотрим, например, модельную задачу с целевыми метками 1, 2, 3, 6, 9. На рис. 8.12-13 показана ошибка разных константных алгоритмов для разных  $p$ . Оптимальные константные решения при  $p=1$  и  $p=0.5$  совпадают.

При  $1 \leq p \leq 2$  оптимальное константное решение смещается от медианы к среднему, а при увеличении  $p$  – к mid-range (среднему арифметическому максимума и минимума), см. рис. 8.14.

Следующие функции пытаются измерить ошибку в процентах.

**Средний процент отклонения (MAPE – Mean Absolute Percent Error):**

$$\text{MAPE} = 100\% \cdot \frac{1}{m} \sum_{i=1}^m \frac{|y_i - a_i|}{|y_i|}$$

**Симметричный средний процент отклонения (SMAPE – Symmetric Mean Absolute Percentage Error):**

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

Эти функции ошибок выглядят очень логично: модуль разницы между ответом алгоритма и истинным значением мы делим на их среднее (или истинное значение). Будем считать, что значения целевого признака и ответы алгоритма положительные. При вычислении SMAPE можно считать, что они неотрицательные:

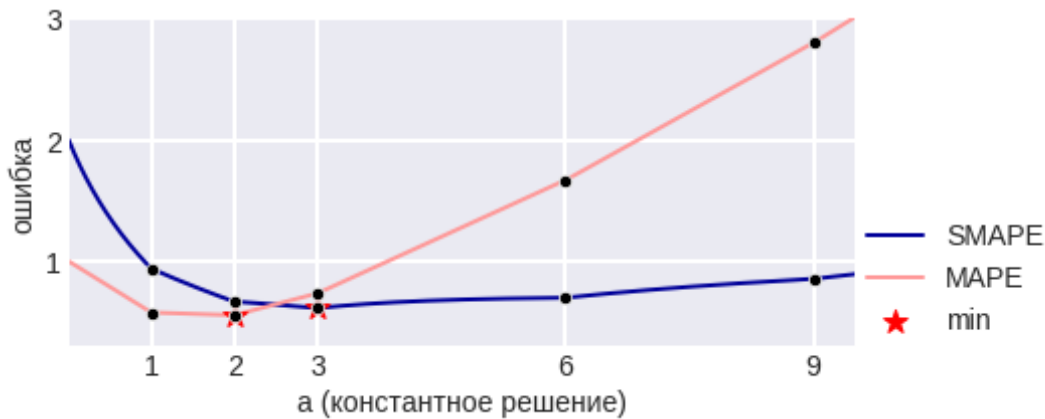
- если  $y_i = a_i = 0$ , тогда неопределённость вида  $0/0$  следует считать 0.
- $a_i > y_i = 0$  или  $y_i > a_i = 0$ , тогда получаем, что ошибка (неугадывания этого значения) составляет 200%.

Вторая ситуация не кажется логичной и именно она вызывает проблемы в ситуациях, когда целевые значения часто равны нулю и мы используем SMAPE: при настройке алгоритма мы сосредотачиваемся на «угадывании нулей» (а не настраиваемся на ненулевые целевые значения).

Менеджеры часто просят посчитать ошибку в процентах, обосновывая это её интуитивностью. Почти все они впадают в ступор при вопросе, сколько процентов составляет 2 от 0?

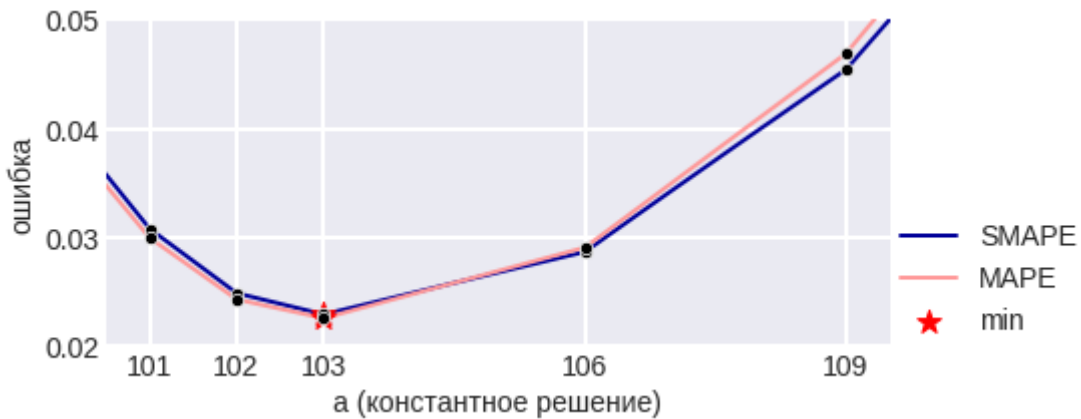
Функции SMAPE и MAPE часто используют при прогнозировании временных рядов, особенно финансовых, где как раз разумно измерять ошибку в процентах. Например, рассмотри две ситуации: вместо цены за акцию 1 у.е. мы

предсказали 2 и вместо 101 – 102. В первом случае мы ошиблись почти в два раза, а во втором «почти угадали», тем не менее MAE-ошибка в обоих случаях равна 1, а вот (S)MAPE сильно различается.



**Рис. 8.15. SMAPE и MAPE-ошибки константных решений для меток 1, 2, 3, 6, 9.**

Ни рис. 8.15 показано качество константных решений для модельной выборки с целевыми значениями 1, 2, 3, 6, 9. Но такая ситуация не очень практическая: целевые значения редко так сильно отличаются друг от друга (в 9 раз!), поэтому на рис. 8.16 показана более реалистичная ситуация.



**Рис. 8.16. SMAPE и MAPE-ошибки константных решений для меток 101, 102, 103, 106, 109.**

Заметим, что на рис. 8.16 оптимальное решение – медиана. Действительно, на практике при работе с (S)MAPE медиана часто полезна, например, для усреднения алгоритмов в ансамбле.

Отметим также, что хотя MAPE и SMAPE похожи, с точки зрения практической минимизации между ними большое отличие, первую функцию можно переписать в таком виде:

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

здесь существенно то, что веса  $w_i$  определяются постановкой задачи и не зависят от ответов алгоритма (их можно вычислить по обучающей выборке и производить обучение, учитывая их значения), т.е. MAPE – это просто весовой MAE.

Есть и другие способы перевода модуля отклонения в проценты, например

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

но эта функция отличается от MAE константным множителем (и с точки зрения минимизации они эквивалентны).

Есть целый класс функций ошибок, которые основаны на сравнении ошибки с некоторым бенчмарком. Пусть  $a'_i$  – метки, полученные бенчмарком.

**MRAE – Mean Relative Absolute Error:**

$$\text{MRAE} = \frac{1}{m} \sum_{i=1}^m \frac{|y_i - a_i|}{|y_i - a'_i|}$$

Это среднее отношений ошибок на разных объектах. Можно также поделить суммарные ошибки:

$$\text{REL\_MAE} = \frac{\sum_{i=1}^m |y_i - a_i|}{\sum_{i=1}^m |y_i - a'_i|},$$

или посчитать, в каком проценте случаев алгоритм лучше бенчмарка – это делает функционал качества **Percent Better:**

$$\text{PB(MAE)} = \frac{1}{m} \sum_{i=1}^m I[|y_i - a_i| < |y_i - a'_i|]$$

В качестве бенчмарка можно брать

- простой константный алгоритм (часто используют нулевой, сравните MAPE и MRAE),
- предыдущую версию Вашего алгоритма (чтобы оценить, насколько улучшилось решения),
- естественный алгоритм (в задачах прогнозирования рядов самый естественный и простой прогноз: значение функции в следующий момент времени такое же, как в предыдущий).

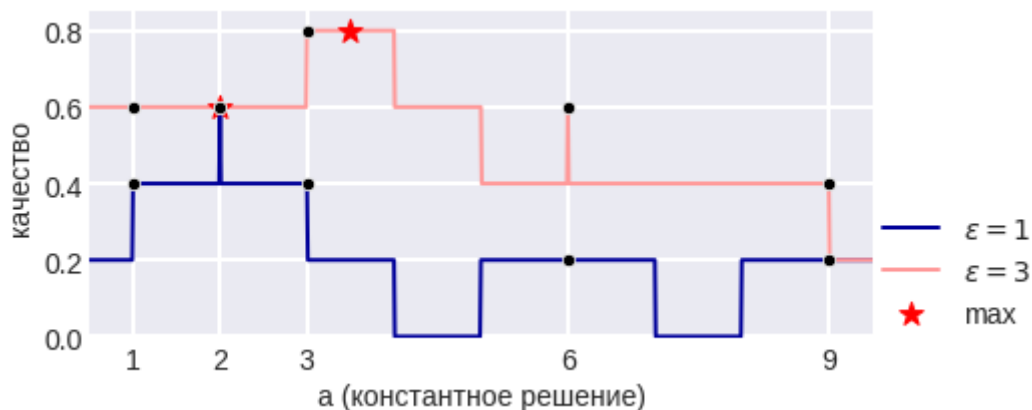
**Mean Absolute Scaled Error** иллюстрирует последнее замечание:

$$\text{MASE} = \frac{1}{\frac{m}{m-1} \sum_{i=2}^m |y_i - y_{i-1}|} \sum_{i=1}^m |y_i - a_i|$$

Здесь везде модуль разности между значением, полученным алгоритмом, и истинным можно заменить на модуль в некоторой степени. Ещё один функционал качества (уже не функция ошибки): процент случаев, когда ответ алгоритма верен с некоторой заранее заданной точностью

$$eB = \frac{1}{m} \sum_{i=1}^m I[|y_i - a_i| \leq \varepsilon].$$

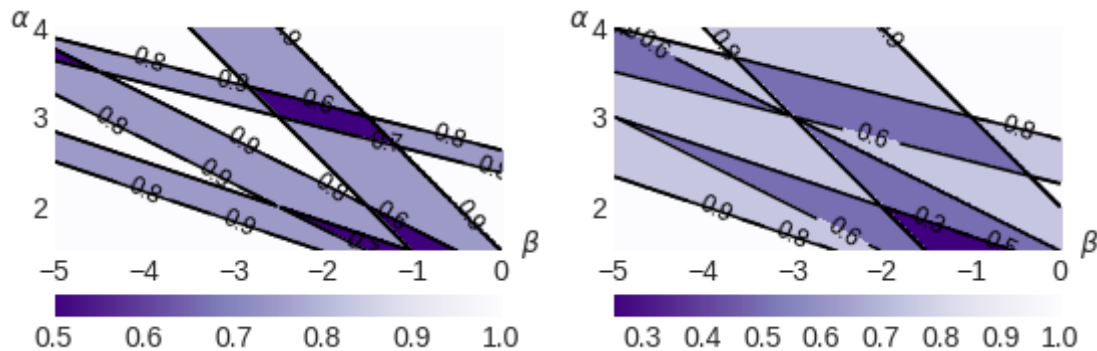
Оптимальное константное значение для этого функционала мы фактически научились строить в [главе «Средние»](#). График этого функционала в зависимости от константного алгоритма показан на рис. 18.17, по сути это график парzenовской оценки плотности выборки (с точностью до нормировки).



**Рис. 8.17. eB-качество константных решений для меток 1,2, 3, 6, 9.**



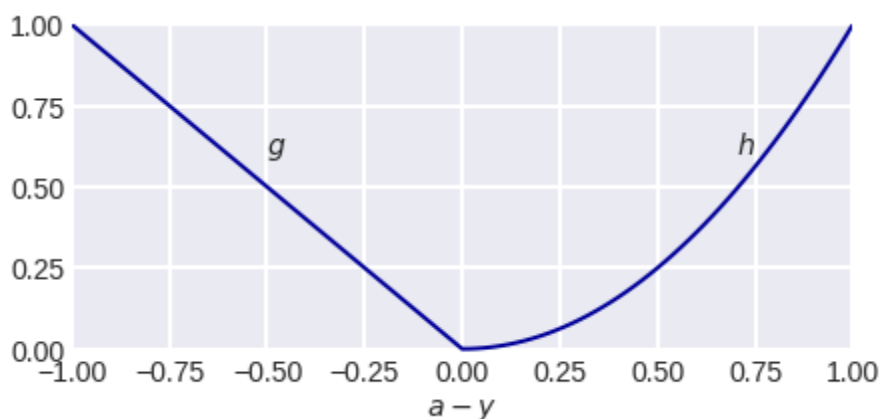
На рис. 8.18 показаны значения точности в модельной задаче рис. 8.10 в зависимости от параметров линейной регрессии. Естественно, здесь уже качество не выпукло по параметрам.



**Рис. 8.18.** Линии уровней функции  $\epsilon$ -качества для  $\epsilon = 0.5$  (слева) и  $\epsilon = 1$  (справа) в модельной задаче.

Стоит обратить внимание, что не всегда функции ошибки симметричны. На практике часто ошибки в разные стороны (завышение и занижение) неравнозначны. Например, при прогнозировании спроса на товар: если наш алгоритм занижает число продаж, то мы рискуем, что товар быстро раскупят и будут недовольные покупатели. Если же завышает, то мы не сможем продать весь товар, он останется на складе (в дальнейшем его придётся продавать по сниженным ценам или возвращать поставщику). С точки зрения денежных потерь, «завысить прогноз на 10» и «занизить на 10» имеют разную стоимость. В таких случаях используют несимметричные функции ошибок:

$$\frac{1}{m} \sum_{i=1}^m \begin{cases} g(|y_i - a_i|), & y_i < a_i, \\ h(|y_i - a_i|), & y_i \geq a_i, \end{cases}$$

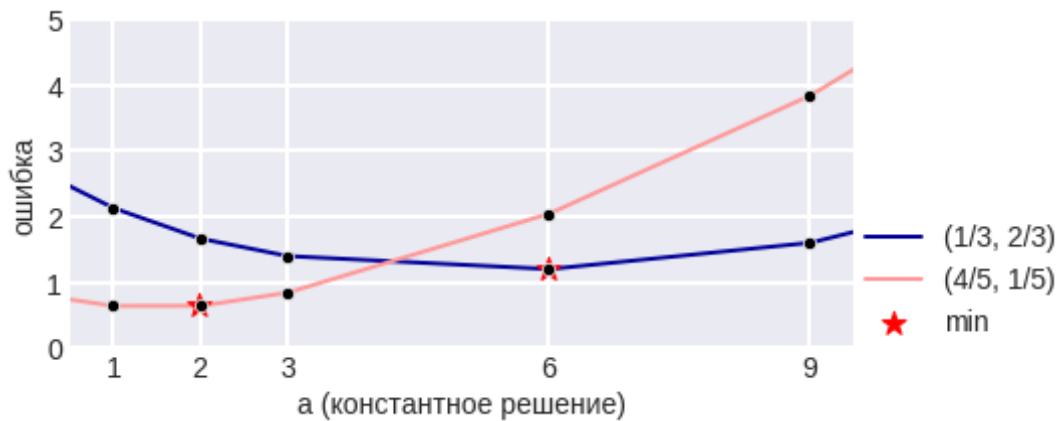


**Рис. 8.19.** Пример несимметричной функции ошибки.

На рис. 8.20 показана ошибка константного алгоритма для функции

$$\frac{1}{m} \sum_{i=1}^m \begin{cases} k_1 |y_i - a_i|, & y_i < a_i, \\ k_2 |y_i - a_i|, & y_i \geq a_i, \end{cases}$$

При  $k_1 = k_2$  оптимальной константой была бы медиана. При  $k_1 < k_2$  мы больше штрафует завышение и оптимальная константа смещается влево, при  $k_1 > k_2$  больше штрафует за занижение и оптимальная константа смещается вправо, но она всегда остаётся представителем выборки.



**Рис. 8.20. Несимметричные ошибки константных решений для меток 1, 2, 3, 6, 9.**

На самом деле, рассмотренная нами функция ошибки – т.н. ошибка **квантильной регрессии (Quantile regression)**, её минимизирует  $\tau$ -квантиль,  $\tau = k_2 / (k_1 + k_2)$ .

### Нестандартное применение функций ошибок

С помощью функций ошибок можно также генерировать признаки в некоторых задачах машинного обучения. Например, в задаче классификации сигналов сигнал часто описывают с помощью признаков. В такое описание могут включать статистические характеристики сигнала (среднее, дисперсию), описание некоторой модели, которая представляет сигнал (например, коэффициенты в разложении Фурье) и т.п. Можно также оценивать, насколько сигнал «предсказуем». Например, мы строим модель сигнала до некоторого момента времени и сравниваем прогноз модели с истиной после этого момента. Ошибка прогноза может оказаться неплохим признаком.

## Реализация функций ошибок в scikit-learn

Ниже представлены функции, реализованные в библиотеке scikit-learn, а также их прямые реализации (используя функции numpy). Из функций ошибок, которые мы в явном виде не упомянули, там реализована **Median Absolute Error**:

$$\text{MedAE} = \text{median}(|a_1 - y_1|, \dots, |a_m - y_m|).$$

```

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)))

```

## Литература

- Курс «How to Win a Data Science Competition: Learn from Top Kagglers» // <https://ru.coursera.org/learn/competitive-data-science>
- Стрижов В.В. Функция ошибки в задачах восстановления регрессии // Заводская лаборатория, 2013, 79(5): 65-73. <http://strijov.com/papers/Strijov2012ErrorFn.pdf>

Это часть главы книги автора, подготовленная для поста в блоге «Анализ малых данных»  
<https://dyakonov.org/>

Все замечания, предложения, найденные неточности и ошибки пишите в комментариях к  
основному посту, там же найдёте ссылки на другие материалы.

<https://dyakonov.org/2018/10/23/функции-ошибок-в-задачах-регрессии/>

Спасибо за внимание к этой работе!