Лекция 3
Линейные методы регрессии.

Кантонистова Е.О.

ВШЭ, 2019

ФУНКЦИОНАЛЫ ОШИБКИ В ЗАДАЧАХ РЕГРЕССИИ

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j.$$

Обучение линейной регрессии (минимизация среднеквадратичной ошибки):

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (\langle w, x_i \rangle - y_i)^2 \to \min_w$$

СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE (MEAN SQUARED ERROR)

Среднеквадратичное отклонение (среднекв. ошибка):

$$MSE(a, X) = \frac{1}{l}\sum_{i=1}^{l}(\boldsymbol{a(x_i) - y_i})^{\boldsymbol{2}}$$

Плюсы:

- Позволяет сравнивать модели
- Подходит для контроля качества во время обучения

Минусы:

- Плохо интерпретируется, т.к. не сохраняет единицы измерения (если целевая переменная – кг, то MSE измеряется в кг в квадрате)

RMSE (ROOT MEAN SQUARED ERROR)

Корень из среднеквадратичной ошибки:

$$RMSE(a, X) = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2}$$

Плюсы:

- Все плюсы MSE
- Сохраняет единицы измерения

Минусы:

- Не позволяет понять, насколько хорошо данная модель решает задачу (это минус и для MSE)

КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ (R^2)

Коэффициент детерминации:

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^{l}(a(x_i) - y_i)^2}{\sum_{i=1}^{l}(y_i - \overline{y})^2},$$

где $\overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i$.

Коэффициент детерминации объясняет долю дисперсии, объясняемую целевой переменной.

- Чем ближе R^2 к 1, тем лучше модель объясняет данные
- Чем ближе R^2 к 0, тем ближе модель к константному предсказанию

MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

$$MAE(a, X) = \frac{1}{l}\sum_{i=1}^{l} |\boldsymbol{a(x_i) - y_i}|$$

Плюсы:

- Менее чувствителен к выбросам, чем MSE

Минусы:

- MAE - не дифференцируемый функционал

MSLE (MEAN SQUARED LOGARITHMIC ERROR)

Среднеквадратичная логарифмическая ошибка:

$$MSLE(a, X) = \frac{1}{l}\sum_{i=1}^{l}\left(\log(a(x_i) + 1) - \log(y + 1)\right)^2$$

- Подходит для задач с неотрицательной целевой переменной ($y \geq 0$)
- Штрафует за отклонения в порядке величин
- Штрафует заниженные прогнозы сильнее, чем завышенные

MAPE, SMAPE

MAPE – mean absolute percentage error:

$$MAPE(a, X) = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - a(x_i)|}{|y_i|}$$

- Измеряет относительную ошибку
- Хорошо интерпретируема: например, MAPE=0.16 означает, что модель может ошибаться в среднем на 16%.

SMAPE – symmetric mean absolute percentage error (симметричный вариант MAPE):

$$SMAPE(a, X) = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - a(x_i)|}{(|y_i| + |a(x_i)|)/2}$$

КВАНТИЛЬНАЯ РЕГРЕССИЯ

Квантильная функция потерь:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \rho_\tau(y_i - a(x_i))$$

Здесь

$$\rho_\tau(z) = (\tau - 1)[z < 0]z + \tau[z \geqslant 0]z = (\tau - \frac{1}{2})z + \frac{1}{2}|z|$$

Параметр $\tau \in [0; 1]$.

- Чем больше τ, тем больше штрафуем за занижение прогноза.

МЕТОДЫ БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ

МЕТОД БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ: РЕГУЛЯРИЗАЦИЯ

Утверждение. Если в выборке есть линейно-зависимые признаки, то задача оптимизации $Q(w) \to min$ имеет бесконечное число решений.

- Большие значения параметров (весов) модели w – признак переобучения.

МЕТОД БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ: РЕГУЛЯРИЗАЦИЯ

Утверждение. Если в выборке есть линейно-зависимые признаки, то задача оптимизации $Q(w) \to min$ имеет бесконечное число решений.

- Большие значения параметров (весов) модели w – признак переобучения.

Решение проблемы – ***регуляризация***.

Регуляризованный функционал ошибки:

$$\boldsymbol{Q_{alpha}(w) = Q(w) + \alpha \cdot R(w)},$$

где $R(w)$ - регуляризатор.

РЕГУЛЯРИЗАЦИЯ

- Регуляризация штрафует за слишком большую норму весов

Наиболее используемые регуляризаторы:

- L_2-регуляризатор: $R(w) = \left||w|\right|_2 = \sum_{i=1}^{d} w_i^2$
- L_1-регуляризатор: $R(w) = \left||w|\right|_1 = \sum_{i=1}^{d} |w_i|$

Пример регуляризованного функционала:

$$Q(a(w), X) = \frac{1}{l}\sum_{i=1}^{l}((w, x_i) - y_i)^2 + \alpha \sum_{i=1}^{d} w_i^2 ,$$

где α – коэффициент регуляризации.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ МНК С L_2-РЕГУЛЯРИЗАТОРОМ

Задача оптимизации в матричном виде:

$$Q(w) = (y - Xw)^T(y - Xw) + \alpha w^T I w \to min \quad (*)$$

где I – единичная матрица.

Эта задача имеет аналитическое решение:

$$\boldsymbol{w = \left(X^T X + \alpha I\right)^{-1} X^T y}$$

- Матрица $X^T X + \alpha I$ всегда положительно определена, поэтому её можно обратить. Следовательно, задача (*) имеет единственное решение.

РАЗРЕЖЕННЫЕ МОДЕЛИ

Разреженные модели – модели, в которых часть весов равна 0.

Связь разреженных моделей с отбором признаков:

- Некоторые признаки могут не иметь отношения к задаче, т.е. они не нужны. Тогда при зануленении весов при этих признаках происходит *отбор признаков*.
- Если есть ограничения на скорость получения предсказаний, то чем меньше признаков, тем быстрее
- Если признаков больше, чем объектов, то решение задачи будет неоднозначным. Поэтому надо делать *отбор признаков*.

L_1-РЕГУЛЯРИЗАЦИЯ

Утверждение. В результате обучения модели с L_1-регуляризатором происходит зануление некоторых весов, т.е. отбор признаков.

Можно показать, что задачи

$$(1) \quad Q(w) + \alpha \left|\left|w\right|\right|_1 \to \min_w$$

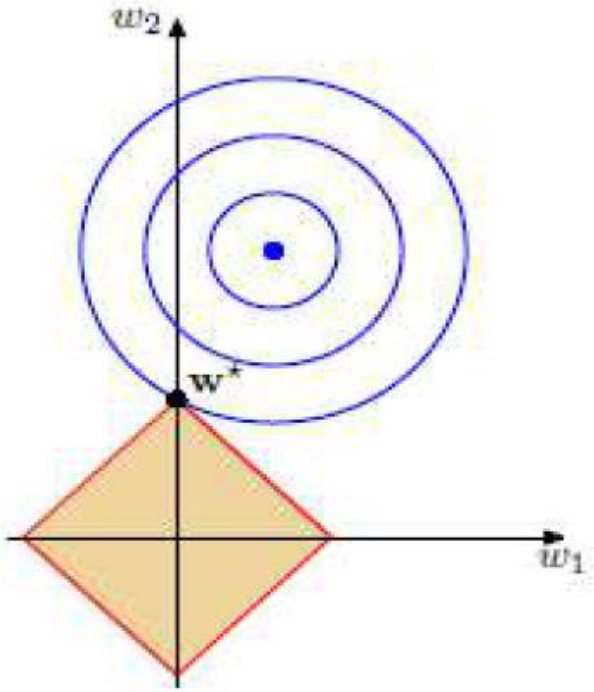
и

$$(2) \quad \begin{cases} Q(w) \to \min\limits_w \\ \left|\left|w\right|\right|_1 \leq C \end{cases}$$

эквивалентны.

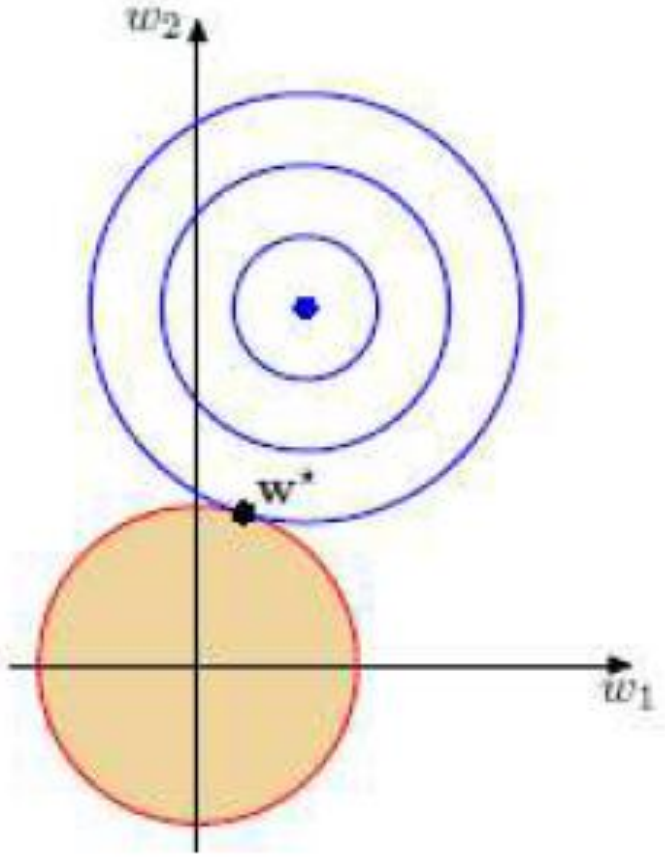
ОТБОР ПРИЗНАКОВ ПО L1-РЕГУЛЯРИЗАЦИИ

Нарисуем линии уровня $Q(w)$ и область $\left|\left|w\right|\right|_1 \leq C$:



Если признак незначимый, то соответствующий вес близок к 0. Отсюда получим, что в большинстве случаев решение нашей задачи попадает в вершину ромба, т.е. обнуляет незначимый признак.

L2-РЕГУЛЯРИЗАЦИЯ НЕ ОБНУЛЯЕТ ПРИЗНАКИ

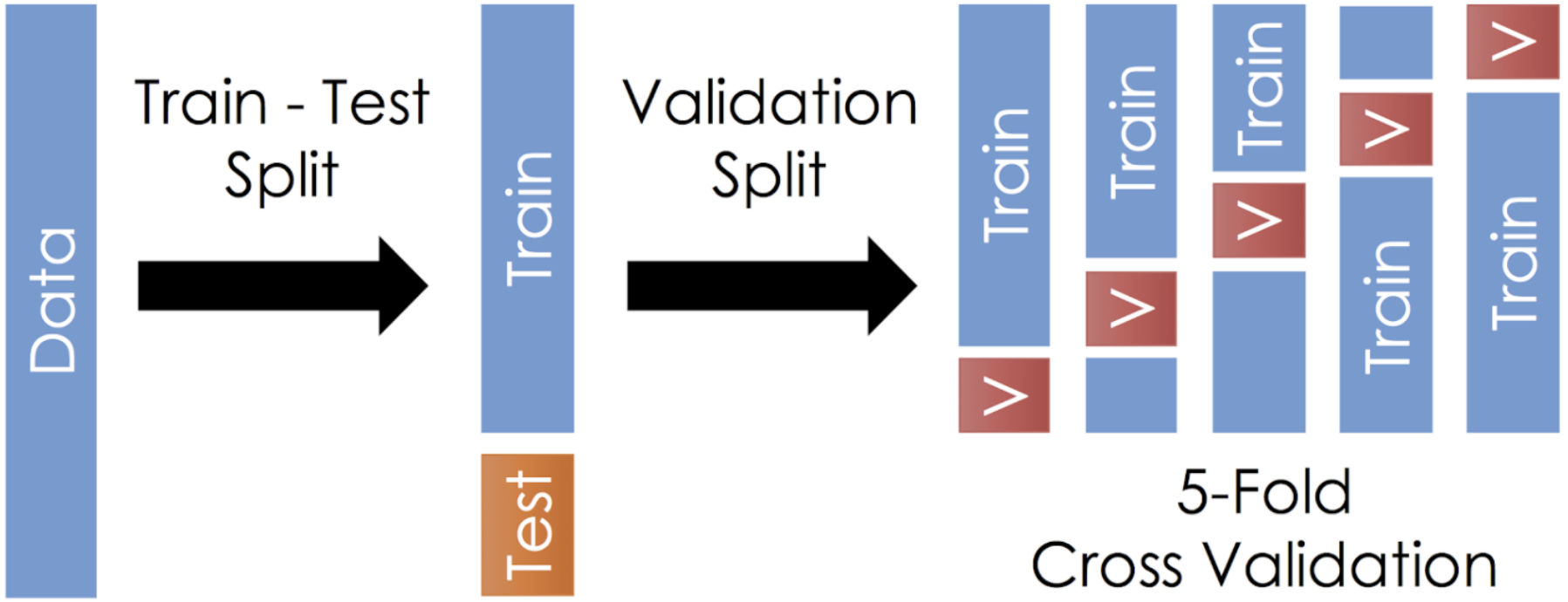


ГИПЕРПАРАМЕТРЫ МОДЕЛИ

- **Параметры модели –** величины, настраивающиеся по обучающей выборке (например, веса w в линейной регресии)
- **Гиперпараметры модели –** величины, контролирующие процесс обучения. Поэтому они не могут быть настроены по обучающей выборке (например, коэффициент регуляризации α).

Проблема: если подбирать гиперпараметры по кросс-валидации, то мы будем использовать отложенную (валидационную) выборку для поиска наилучших значений гиперпараметров. Т.е. отложенная выборка становится обучающей.

СХЕМА РАЗБИЕНИЯ ДАННЫХ ДЛЯ ПОДБОРА ПАРАМЕТРОВ И ГИПЕРПАРАМЕТРОВ МОДЕЛИ



РАБОТА С ПРИЗНАКАМИ

КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак $f_j(x)$ принимает m различных значений: $C_1, C_2, \ldots, C_m$.

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

КОДИРОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ: ONE-HOT ENCODING

- Предположим, категориальный признак $f_j(x)$ принимает m различных значений: $C_1, C_2, \ldots, C_m$.

Пример: еда может быть *горькой, сладкой, солёной или кислой* (4 возможных значения признака).

- Заменим категориальный признак на m бинарных признаков: $b_i(x) = [f_j(x) = C_i]$ (индикатор события).

Тогда One-Hot кодировка для нашего примера будет следующей:

горький = (1,0,0,0), *сладкий* = (0,1,0,0),

солёный = (0,0,1,0), *кислый* = (0,0,0,1).

ХЭШИРОВАНИЕ ПРИЗНАКОВ

- Возьмем некоторую функцию (hash-функция), которая переводит значения категориального признака в числа от *1* до *B*: $h: U \rightarrow \{1,2, \dots, B\}$.
- То есть для каждого объекта:

$$g_j(x) = \left[h\left(f(x)\right) = j\right], j = 1, \dots, B$$

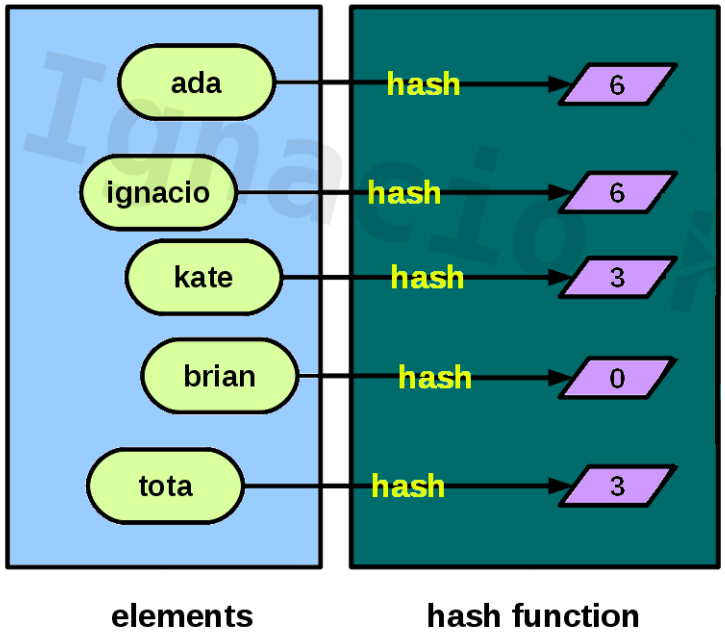
ХЭШИРОВАНИЕ ПРИЗНАКОВ

- Возьмем некоторую функцию (hash-функция), которая переводит значения категориального признака в числа от *1* до *B*: $h: U \rightarrow \{1,2,\dots,B\}$.
- То есть для каждого объекта:

$$g_j(x) = \left[h\left(f(x)\right) = j\right], j = 1,\dots,B$$

Идея: *хэширование группирует значения признака. Так как часто встречающихся значений немного, они редко попадают в одну группу при группировке.*

ХЭШИРОВАНИЕ ПРИЗНАКОВ



0	1	2	3	4	5	6
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	1	0	0	0
1	0	0	0	0	0	0
0	0	0	1	0	0	0

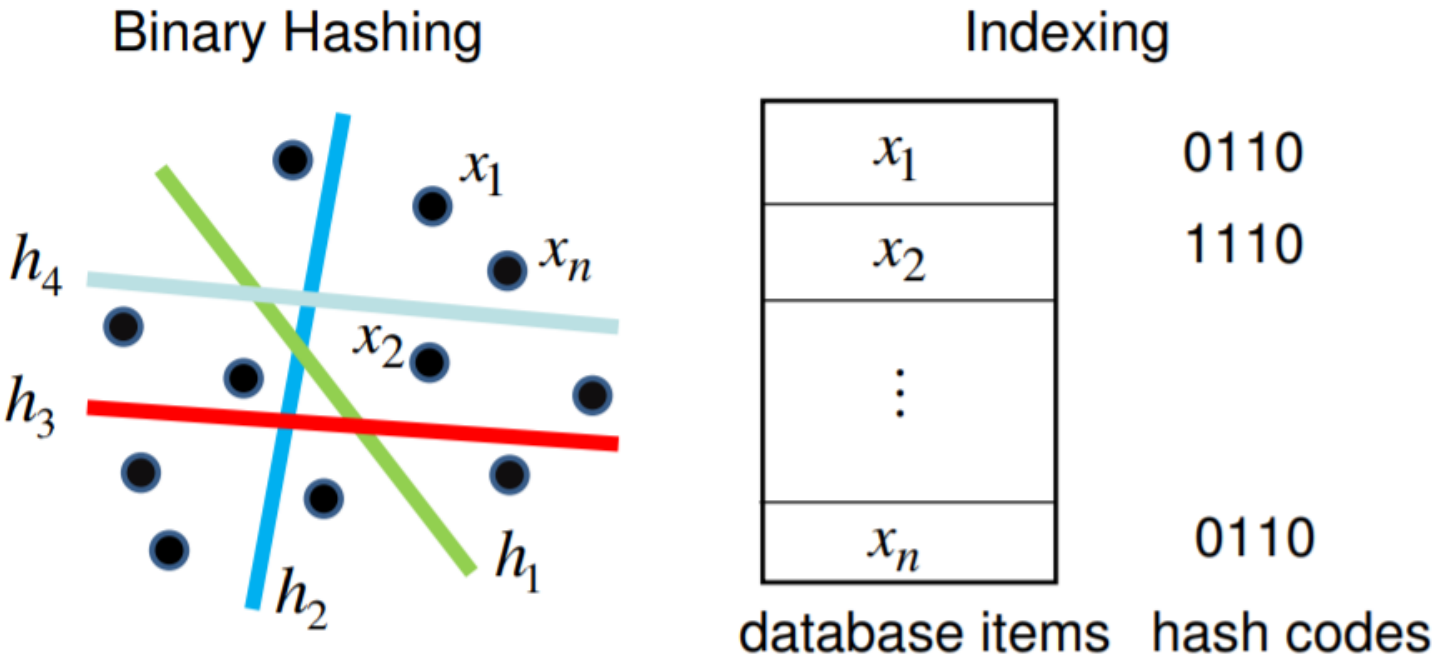
ХЭШИРОВАНИЕ ПРИЗНАКОВ

- Возьмем некоторую функцию (hash-функция), которая переводит значения категориального признака в числа от *1* до *B*: $h: U \rightarrow \{1,2,\dots,B\}$.
- То есть для каждого объекта:

$$g_j(x) = \left[h\left(f(x)\right) = j\right], j = 1,\dots,B$$

+ позволяет закодировать любое значение категориального признака (в том числе, то, которого не было в тренировочной выборке)

ХЭШИРОВАНИЕ ДЛЯ ПОНИЖЕНИЯ РАЗМЕРНОСТИ



ХЭШИРОВАНИЕ

- Хороший способ работать с категориальными данными, принимающими множество различных значений
- Хорошие результаты на практике
- Позволяет понизить размерность пространства признаков с незначительным снижением качества

Статья про хэширование:

https://arxiv.org/abs/1509.05472

СЧЁТЧИКИ

- Пусть целевая переменная y принимает значения от 1 до K.
- Закодируем категориальную переменную $f(x)$ следующим способом:

$$counts(u, X) = \sum_{(x,y)\in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], k = 1, \dots, K$$

СЧЁТЧИКИ: ПРИМЕР

city	target	0	1	2
Moscow	1	1/4	1/2	1/4
London	0	1/2	0	1/2
London	2	1/2	0	1/2
Kiev	1	1/2	1/2	0
Moscow	1	1/4	1/2	1/4
Moscow	0	1/4	1/2	1/4
Kiev	0	1/2	1/2	0
Moscow	2	1/4	1/2	1/4

СЧЁТЧИКИ

- Пусть целевая переменная y принимает значения от 1 до K.
- Закодируем категориальную переменную $f(x)$ следующим способом:

$$counts(u, X) = \sum_{(x,y)\in X} [f(x) = u]$$

$$successes_k(u, X) = \sum_{(x,y)\in X} [f(x) = u][y = k], k = 1, \dots, K$$

Тогда кодировка:

$$\boldsymbol{g_k(x, X) = \frac{successes_k(f(x), X) + c_k}{counts(f(x), X) + \sum_{i=1}^{K} c_i} \approx p(y = k|f(x))},$$

c_i - чтобы не было деления на 0.

СЧЁТЧИКИ: ОПАСНОСТЬ ПЕРЕОБУЧЕНИЯ

Вычисляя счётчики, мы закладываем в признаки информацию о целевой переменной и, тем самым, переобучаемся!

СЧЁТЧИКИ: КАК ВЫЧИСЛЯТЬ

- Можно вычислять счётчики так:

city	target
Moscow	1
London	0
London	2
Kiev	1
Moscow	1
Moscow	0
Kiev	0
Moscow	2

Вычисляем счетчики по этой части

Кодируем признак вычисленными счётчиками и обучаемся по этой части

СЧЁТЧИКИ: КАК ВЫЧИСЛЯТЬ

Более продвинутый способ (по кросс-валидации):

1) Разбиваем выборку

на m частей $X_1, \dots, X_m$

2) На каждой части X_i

значения признаков

вычисляются по

оставшимся частям:

$x \in X_i \Rightarrow g_k(x) = g_k(x, X \backslash X_i)$