Занятие 3. Линейные методы регрессии.

Елена Кантонистова

elena.kantonistova@yandex.ru

ВШЭ, 2020

МЕТРИКИ КАЧЕСТВА И ФУНКЦИОНАЛЫ ОШИБКИ
В ЗАДАЧАХ РЕГРЕССИИ

МЕТРИКИ КАЧЕСТВА И ФУНКЦИИ ОШИБКИ

- **Функционал (функция) ошибки –** функция, которую минимизируют в процессе обучения модели для нахождения неизвестных параметров (весов).
- **Метрика качества –** функция, которую используют для оценки качества построенной (уже обученной) модели.

МЕТРИКИ КАЧЕСТВА И ФУНКЦИИ ОШИБКИ

- **Функционал (функция) ошибки –** функция, которую минимизируют в процессе обучения модели для нахождения неизвестных параметров (весов).
- **Метрика качества –** функция, которую используют для оценки качества построенной (уже обученной) модели.

Иногда одна и та же функция может использоваться и для обучения модели (функция ошибки), и для оценки качества модели (метрика качества).

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

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

Обучение линейной регрессии - минимизация среднеквадратичной ошибки:

$$\frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2 \to \min_{w}$$

СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE (MEAN SQUARED ERROR)

Среднеквадратичное отклонение:

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

Среднеквадратичное отклонение: MSE (Mean Squared Error)

Среднеквадратичное отклонение:

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

Плюсы:

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

СРЕДНЕКВАДРАТИЧНОЕ ОТКЛОНЕНИЕ: MSE

Среднеквадратичное отклонение:

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

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)
- Тяжело понять, насколько хорошо данная модель решает задачу, так как MSE не ограничена сверху.

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 (в отличие от MSE)**

Минусы:

- Тяжело понять, насколько хорошо данная модель решает задачу, так как RMSE не ограничена сверху.

Коэффициент детерминации (R^2)

Коэффициент детерминации:

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

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

Коэффициент детерминации (R^2)

Коэффициент детерминации:

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

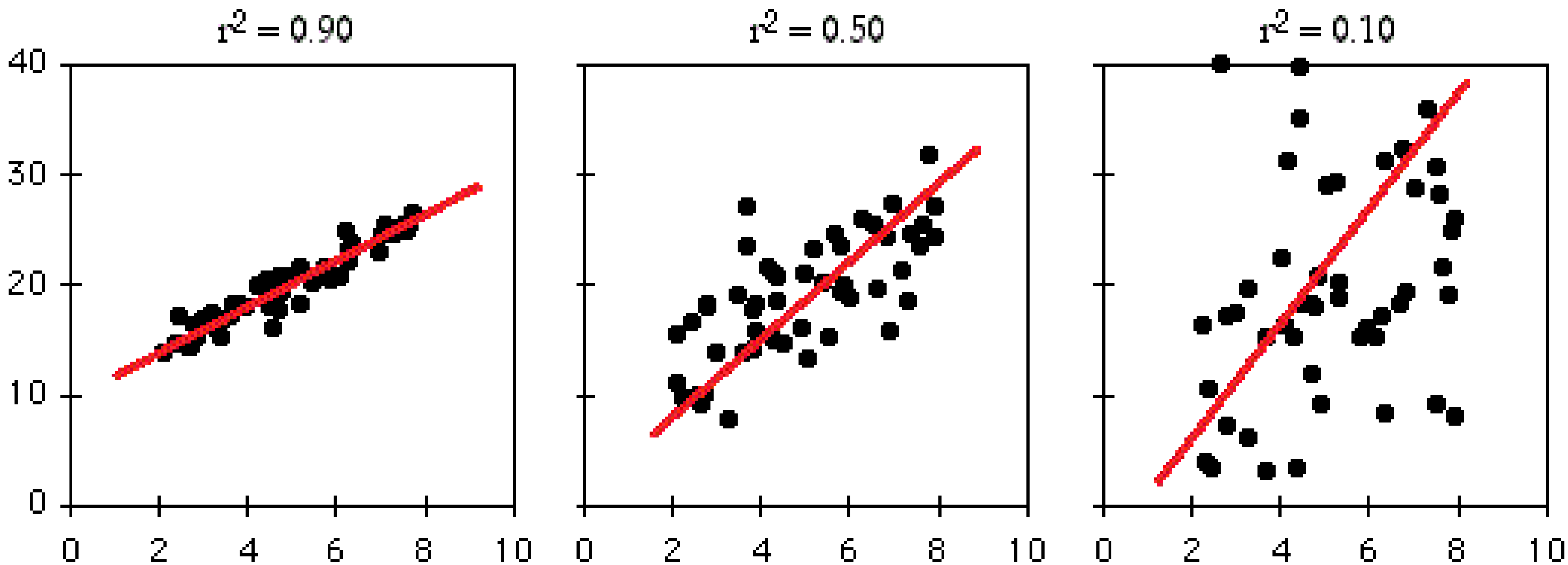
где $\overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i$.

Коэффициент детерминации <u>объясняет долю дисперсии, объясняемую целевой переменной</u>.

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

КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ (R^2)

$\mathbf{0}$ (плохое качество) $\leq \boldsymbol{R^2} \leq \mathbf{1}$ (хорошее качество)



MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

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

MAE (MEAN ABSOLUTE ERROR)

Средняя абсолютная ошибка:

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

Плюсы:

- Менее чувствителен к выбросам, чем MSE

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:

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

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

MAPE

MAPE – Mean Absolute Percentage Error:

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

MAPE измеряет относительную ошибку.

MAPE

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

Плюсы:

- Ограничена: $0 \leq MAPE \leq 1$
- Хорошо интерпретируема: например, MAPE=0.16 означает, что ошибка модели в среднем составляет 16% от фактических значений.

MAPE

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

Плюсы:

- Ограничена: $0 \le MAPE \le 1$
- Хорошо интерпретируема: например, MAPE=0.16 означает, что ошибка модели в среднем составляет 16% от фактических значений.

Минусы:

- По-разному относится к недо- и перепрогнозу. Например, если правильный ответ $y = 10$, а прогноз $a(x) = 20$, то ошибка $\frac{|10-20|}{|10|} = \mathbf{1}$, а если ответ $y = 30$, то ошибка $\frac{|30-20|}{|30|} = \frac{1}{3} \approx \mathbf{0.33}$.

SMAPE

SMAPE – *Symmetric Mean Absolute Percentage Error* (симметричный вариант MAPE):

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

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

SMAPE

SMAPE – *Symmetric Mean Absolute Percentage Error* (симметричный вариант MAPE):

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

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

<u>Проверим:</u>

Пусть правильный ответ $\boldsymbol{y} = \mathbf{10}$, а прогноз $\boldsymbol{a}(\boldsymbol{x}) = \mathbf{20}$, то ошибка $\frac{|10-20|}{|10+20|/2} = \frac{2}{3} \approx \mathbf{0.67}$, а если ответ $\boldsymbol{y} = \mathbf{30}$, то ошибка $\frac{|30-20|}{|30+20|/2} = \frac{2}{5} = \mathbf{0.4}$.

SMAPE

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

<u>Проверим:</u>

Пусть правильный ответ $\boldsymbol{y = 10}$, а прогноз $\boldsymbol{a(x) = 20}$, то ошибка $\frac{|10-20|}{|10+20|/2} = \frac{2}{3} \approx \mathbf{0.67}$, а если ответ $\boldsymbol{y = 30}$, то ошибка $\frac{|30-20|}{|30+20|/2} = \frac{2}{5} = \mathbf{0.4}$.

Ошибки стали меньше отличаться друг от друга, но всё-таки не равны.

SMAPE

SMAPE – попытка сделать симметричным прогноз (то есть дать одинаковую ошибку для недо- и перепрогноза).

“Сейчас уже в среде прогнозистов сложилось более-менее устойчивое понимание, что SMAPE не является хорошей ошибкой. Тут дело не только в завышении прогнозов, но ещё и в том, что наличие прогноза в знаменателе позволяет манипулировать результатами оценки.” (см. источник)

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: $R(w) = \left||w|\right|_1 = \sum_{i=1}^{d} |w_i|$

Регуляризация

- **Регуляризация штрафует за слишком большую норму весов.**

Наиболее используемые регуляризаторы:

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

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.

ПОЛЕЗНОЕ СВОЙСТВО L1-РЕГУЛЯРИЗАЦИИ

Все ли признаки в задаче нужны?

ПОЛЕЗНОЕ СВОЙСТВО L1-РЕГУЛЯРИЗАЦИИ

Все ли признаки в задаче нужны?

- Некоторые признаки могут не иметь отношения к задаче, т.е. они не нужны.
- Если есть ограничения на скорость получения предсказаний, то чем меньше признаков, тем быстрее
- Если признаков больше, чем объектов, то решение задачи будет неоднозначным.

Полезное свойство L1-регуляризации

Все ли признаки в задаче нужны?

- Некоторые признаки могут не иметь отношения к задаче, т.е. они не нужны.
- Если есть ограничения на скорость получения предсказаний, то чем меньше признаков, тем быстрее
- Если признаков больше, чем объектов, то решение задачи будет неоднозначным.

Поэтому в таких случаях надо делать отбор признаков, то есть убирать некоторые признаки.

ПОЛЕЗНОЕ СВОЙСТВО L1-РЕГУЛЯРИЗАЦИИ

Свойство модели, обученной с помощью минимизации функционала с добавлением L1-регуляризации:

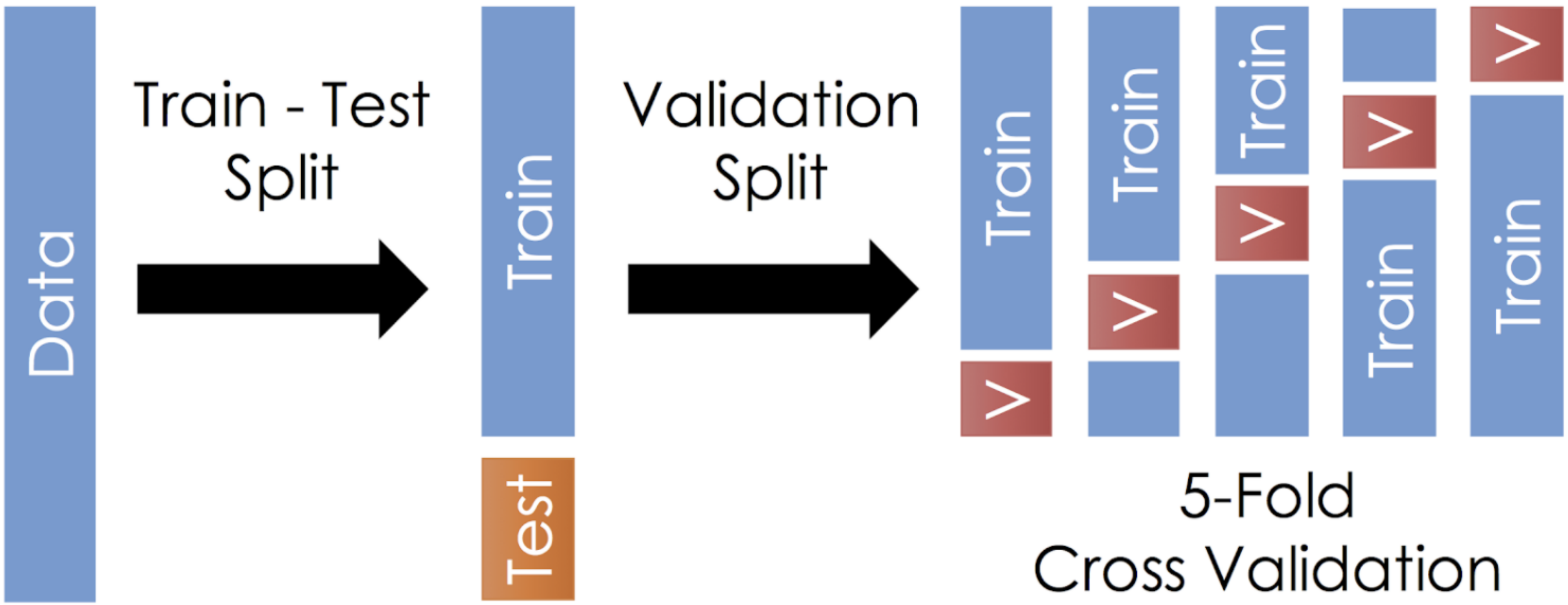
В результате обучения такой модели происходит зануление некоторых весов, то есть отбор признаков.

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: $C_1, C_2, \dots, C_m$.

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)*.

ONE-HOT ENCODING В PYTHON

Один из способов сделать One-hot кодирование в Python – применить функцию get_dummies из библиотеки pandas.

Пример:

data = pd.get_dummies(data, columns=['City_Category'], dtype=int)

Пояснение:

Столбец 'City_Category’ содержит категориальные данные, поэтому кодируем его. В результате применения кодирования вместо столбца 'City_Category’ в таблице data появятся закодированные с помощью One-hot кодировки столбцы.

СЧЁТЧИКИ

Счётчик (*mean target encoding*) – это вероятность получить значение целевой переменной для данного значения категориального признака.

СЧЁТЧИКИ (ПРИМЕР)

	feature	target
0	Moscow	0
1	Moscow	1
2	Moscow	1
3	Moscow	0
4	Moscow	0
5	Tver	1
6	Tver	1
7	Tver	1
8	Tver	0
9	Klin	0
10	Klin	0
11	Tver	1

СЧЁТЧИКИ (ПРИМЕР)

	feature	target
0	Moscow	0
1	Moscow	1
2	Moscow	1
3	Moscow	0
4	Moscow	0
5	Tver	1
6	Tver	1
7	Tver	1
8	Tver	0
9	Klin	0
10	Klin	0
11	Tver	1

	feature	feature_mean	target
0	Moscow	0.4	0
1	Moscow	0.4	1
2	Moscow	0.4	1
3	Moscow	0.4	0
4	Moscow	0.4	0
5	Tver	0.8	1
6	Tver	0.8	1
7	Tver	0.8	1
8	Tver	0.8	0
9	Klin	0.0	0
10	Klin	0.0	0
11	Tver	0.8	1

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

В случае бинарной классификации счётчики можно задать формулой:

$$Likelihood = \frac{Goods}{Goods + Bads} = mean(target),$$

где *Goods* – число единиц в столбце target,

Bads – число нулей в столбце target.

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