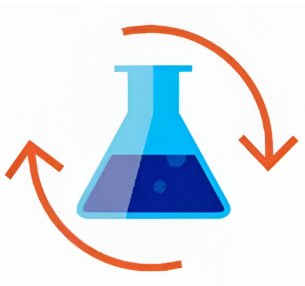


Преоброцессинг и контроль качества моделей

Виктор Китов

victorkitov.github.io

Курс поддержан
фондом
'Интеллект'



Победитель
конкурса VK среди
курсов по IT



Препроцессинг данных

Препроцессинг данных включает:

- заполнение пропусков
- детекцию и обработку выбросов
- кодирование признаков
- преобразование признаков
- отбор признаков / снижение размерности
- преобразование отклика

Перед работой с данными:

- визуализировать зависимость y от самых значимых признаков
(график, scatter-plot с раскраской)
- визуализировать распределение в первых 2х, 3х главных компонентах
(с раскраской)

Заполнение пропусков

- Самый простой способ - удалить объекты с пропусками.
 - потеряем много информации, если таких много
- Заполнение:
 - для вещественных:
 - средним/медианой
 - аномально большим или малым значением (для деревьев решений)
 - для категориальных
 - самой частой категорией
 - новой категорией "пропуск"
- Более точно - предсказывать пропущенные признаки, используя располагаемые.

Пример использования

```
# среднее для числовых признаков (для медианы - 'median')
num_imputer = SimpleImputer(strategy='mean')
# мода для категориальных признаков
cat_imputer = SimpleImputer(strategy='most_frequent')

preprocessor = ColumnTransformer(
    transformers=[
        ('num', num_imputer, NUM_COLS),
        ('cat', cat_imputer, CAT_COLS)
    ],
    remainder='passthrough'
    # остальные пр-ки без изменений
)

X_train_imputed = pd.DataFrame(
    preprocessor.fit_transform(X_train),
    columns=NUM_COLS + CAT_COLS)
```

[Ссылка на полный код.](#)

Другие преобразования

Детекция выбросов:

- Строить распределения каждого признака, удалять объекты с аномальными значениями.
- Либо по правилу непринадлежности значения $(\mu + k\sigma, \mu + k\sigma)$, $k \sim 4$.

Преобразование отклика:

- преобразуем $y \rightarrow g(y)$ и обучаем модель на $\{x_n, g(y_n)\}_n$
- применяем модель, прогноз $g^{-1}(\hat{y})$
- популярно: $g(y) = \ln y$
- для поиска $g(\cdot)$:
 - строим scatter-plot зависимости y от значимых признаков.
 - либо подбираем $g(\cdot)$, чтобы $g(y) \sim \mathcal{N}(0, 1)$

Содержание

- 1 Кодирование признаков
- 2 Генерация признаков
- 3 Представление текстов
- 4 Переобучение и недообучение
- 5 Оценка качества на тесте
- 6 Оценка качества регрессии

Признаки

- Можно использовать вещественные признаки и бинарные.
- Вещественные можно дискретизовать
 - Пример: возраст, зарплата. По-разному обрабатываем каждый сегмент.
- Как представить категориальные признаки?

Признаки

- Можно использовать вещественные признаки и бинарные.
- Вещественные можно дискретизовать
 - Пример: возраст, зарплата. По-разному обрабатываем каждый сегмент.
- Как представить категориальные признаки?
 - номером категории (плохо)
 - счетчиком встречаемости категории
 - в виде бинарных (one-hot encoding)
 - в виде вещественных (mean value encoding, cyclic encoding)

One-hot кодирование (one-hot encoding)

Row Number	Direction		Row Number	Direction_N	Direction_S	Direction_W	Direction_E	Direction_NW
1	North	→	1	1	0	0	0	0
2	North-West		2	0	0	0	0	1
3	South		3	0	1	0	0	0
4	East		4	0	0	0	1	0
5	North-West		5	0	0	0	0	1
	North-West			0	0	0	0	1
	East			0	0	0	1	0
	South			0	1	0	0	0

One-hot кодирование (one-hot encoding)

Row Number	Direction		Row Number	Direction_N	Direction_S	Direction_W	Direction_E	Direction_NW
1	North	→	1	1	0	0	0	0
2	North-West		2	0	0	0	0	1
3	South		3	0	1	0	0	0
4	East		4	0	0	0	1	0
5	North-West		5	0	0	0	0	1
	North-West			0	0	0	0	1
	East			0	0	0	1	0
	South			0	1	0	0	0

Какие могут быть проблемы у этого метода?

Пример использования

```
preprocessor = ColumnTransformer(
    transformers=[
        ( 'cat' , OneHotEncoder(handle_unknown='ignore' ,
                                sparse_output=False) ,
          CAT_COLS)
    ],
    remainder='passthrough'
    # остальные пр-ки без изменений
)

# Обучение и применение для train
X_train_trans = preprocessor.fit_transform(X_train)

# Только применение для test
X_test_trans = preprocessor.transform(X_test)
```

Ссылка на полный код (там же - другие виды кодирования).

Кодирование средним (mean encoding)

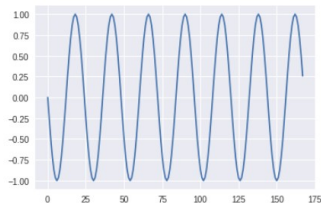
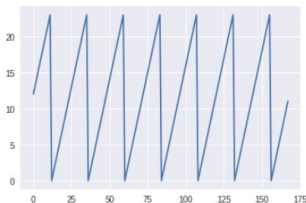
- можно делать по вещественному признаку
- если делаем по y , то на отдельной выборке!

id	job	job_mean	target
1	Doctor	0,50	1
2	Doctor	0,50	0
3	Doctor	0,50	1
4	Doctor	0,50	0
5	Teacher	1	1
6	Teacher	1	1
7	Engineer	0,50	0
8	Engineer	0,50	1
9	Waiter	1	1
10	Driver	0	0

Циклическое кодирование (cyclic encoding)

- Некоторые признаки принимают циклические значения
 - номер месяца, день месяца, час дня
 - для часа 0,1,2,...23. Но $23 \approx 0$!
- Используем циклическое кодирование:

$$x \rightarrow \left[\sin \left(\frac{2\pi \cdot x}{\max(x)} \right); \cos \left(\frac{2\pi \cdot x}{\max(x)} \right) \right]$$



Первая компонента при кодировании часа в сутках.

Содержание

- 1 Кодирование признаков
- 2 Генерация признаков**
- 3 Представление текстов
- 4 Переобучение и недообучение
- 5 Оценка качества на тесте
- 6 Оценка качества регрессии

Генерация признаков

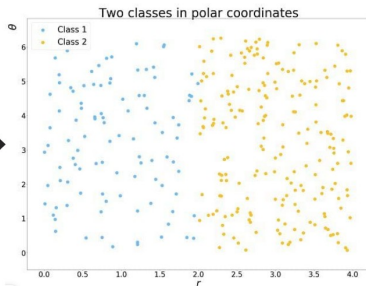
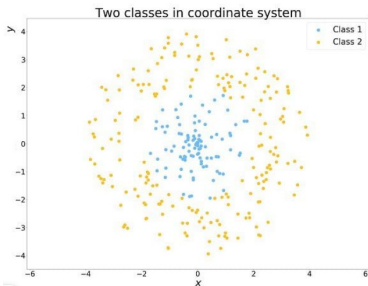
- Генерация признаков (feature engineering) - создание трансформаций из уже известных признаков.
- Хорошая новые признаки существенно повышают качество!
 - см. kaggle соревнования
- Хорошие признаки м.быть свои для каждой модели:
 - линейные трансформации для метрических методов
 - метрические признаки для линейных моделей

Пример генерации признаков

- Трансформация признаков¹:

$$(x, y) \rightarrow (r, \rho)$$

$$r = \sqrt{x^2 + y^2}, \quad \rho = \arctan \frac{y}{x}$$



¹Источник.

Популярные трансформации признаков

Популярные преобразования признаков.

$\phi_k(x)$	примеры
$(x^i)^2, \sqrt{x^i}, \ln x^i$	учитываем нелинейное влияние расстояния до метро на стоимость квартиры
$\mathbb{I}\{x^i \in [a, b]\}$	принадлежит ли клиент определенному возрасту? (совершеннолетний, но не пенсионер)
$x^i \mathbb{I}[x^i \leq a], x^i \mathbb{I}[x^i > a]$	учесть изменения влияния x^i при $x^i > a$
$(x^i)(x^j)$	длина x ширина участка = площадь
x^i/x^j	стоимость квартиры/метраж = стоимость одного метра
$F_{x^i}(x^i)$	приводим признак к равномерному распределению ($F(\cdot)$ - ф-ция распределения)

Популярные трансформации признаков

Использование метрических признаков (метод перестаёт быть линейным, нужна численная оптимизация).

$\phi_k(x)$	примеры
$\langle x, z \rangle / (\ x\ \ z\)$	угол между объектом и репрезентативным объектом z
$\ x - z\ ^2$	расстояние от объекта до репрезентативного объекта z (чаще используют близость)

Содержание

- 1 Кодирование признаков
- 2 Генерация признаков
- 3 Представление текстов**
- 4 Переобучение и недообучение
- 5 Оценка качества на тесте
- 6 Оценка качества регрессии

Токены в текстах

Требуется представить текст вектором $\in \mathbb{R}^D$. Пусть D - $\#$ уникальных слов.

- $x_w = \mathbb{I}[w \text{ встретился в документе}]$
- $x_w = TF_w = \# [w \text{ встретился в документе}]$
- $x_w = TF_w IDF_w, IDF_w = \frac{N}{N_w}$
 - N - $\#$ документов
 - N_w - $\#$ документов, содержащих w хотя бы раз.

Формирование словаря уникальных слов

- в простейшем случае: все уникальные слова языка
- можно ограничить словами предметной области
- можно в разных формах или нормализованной
 - единственное число, именительный падеж, начальная форма глагола.
- убрать слишком частые слова и слишком редкие
 - с редкими - осторожно, многие информативны!
- убрать неинформативные "стоп-слова" из словаря
 - а, но, если, конечно, зато, или, ...

Токены в текстах

- можно добавить биграммы/триграммы:
 - мне фильм не понравился -> 'мне фильм', 'фильм не', 'не понравился'.
 - мне фильм не понравился -> 'мне фильм не', 'фильм не понравился'.
- Но биграм/триграм много.

Токены в текстах

- можно добавить биграммы/триграммы:
 - мне фильм не понравился -> 'мне фильм', 'фильм не', 'не понравился'.
 - мне фильм не понравился -> 'мне фильм не', 'фильм не понравился'.
- Но биграм/триграм много.
- Поэтому можно добавить только коллокации (неслучайно часто встречающиеся слова)
 - Запустили линейную регрессию. Линейная регрессия показала точность... -> 'линейная регрессия'

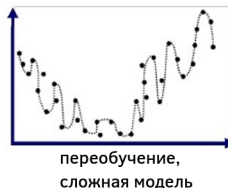
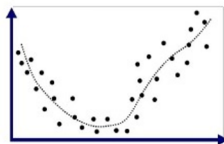
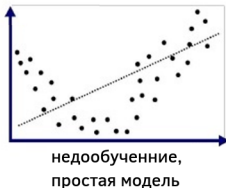
$$\frac{p(w_1 w_2)}{p(w_1)p(w_2)} > threshold$$

Содержание

- 1 Кодирование признаков
- 2 Генерация признаков
- 3 Представление текстов
- 4 Переобучение и недообучение**
- 5 Оценка качества на тесте
- 6 Оценка качества регрессии

Проблемы недообучения и переобучения

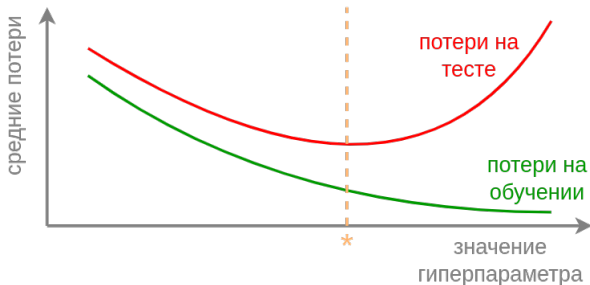
- Недообучение: модель слишком простая для реальных данных.
 - не улавливает тонких закономерностей
- Переобучение: модель слишком сложная для реальных данных.
 - настраивается на шум в измерениях



Вид переобучения

Зависимость потерь от гиперпараметра

- например, K в K -NN, λ в регрессии.



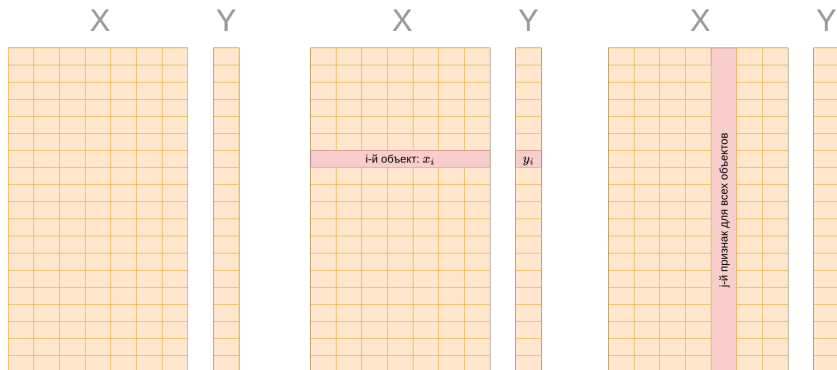
- до *: недообучение
- после *: переобучение

Содержание

- 1 Кодирование признаков
- 2 Генерация признаков
- 3 Представление текстов
- 4 Переобучение и недообучение
- 5 **Оценка качества на тесте**
 - Отдельная валидационная выборка
 - Кросс-валидация
- 6 Оценка качества регрессии

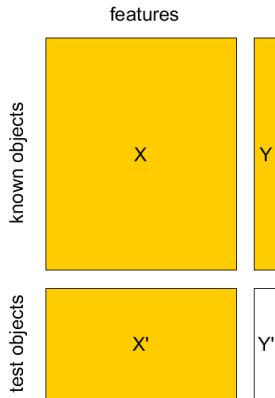
Обучающая выборка

Обучающая выборка (training set): $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$, задаётся матрицей объекты-признаки (design matrix) $X \in \mathbb{R}^{N \times D}$, и вектором откликов (targets) $Y = [y_1, \dots, y_M]^T$.



Обучающая и тестовая выборка

- Обучающая выборка $X, Y: (\mathbf{x}_1, y_1), \dots (\mathbf{x}_M, y_M)$
- Тестовая выборка $X', Y': (\mathbf{x}'_1, y'_1), \dots (\mathbf{x}'_K, y'_K)$



Критерий оптимизации параметров модели

- Необходимо минимизировать **теоретический риск**:

$$\int \int \mathcal{L}(f_w(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \rightarrow \min_w$$

²Предполагаем что объекты независимы и одинаково распределены.

Критерий оптимизации параметров модели

- Необходимо минимизировать **теоретический риск**:

$$\int \int \mathcal{L}(f_w(\mathbf{x}), y) p(\mathbf{x}, y) d\mathbf{x} dy \rightarrow \min_w$$

- Но мы можем минимизировать только **эмпирический риск**²:

$$L(w|X, Y) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_w(\mathbf{x}_n), y_n)$$

- Параметры находим из условия:

$$\hat{w} = \arg \min_w L(w|X, Y)$$

²Предполагаем что объекты независимы и одинаково распределены.

Эмпирический риск на тестовой выборке

- Как связаны $L(\hat{w}|X, Y)$ и $L(\hat{w}|X', Y')$?

Эмпирический риск на тестовой выборке

- Как связаны $L(\hat{w}|X, Y)$ и $L(\hat{w}|X', Y')$?
- В типичной ситуации

$$L(\hat{w}|X, Y) < L(\hat{w}|X', Y')$$

- Эффект растет с ростом переобучения.
- Как получить реалистичную оценку $L(\hat{w}|X', Y')$?

Эмпирический риск на тестовой выборке

- Как связаны $L(\hat{w}|X, Y)$ и $L(\hat{w}|X', Y')$?
- В типичной ситуации

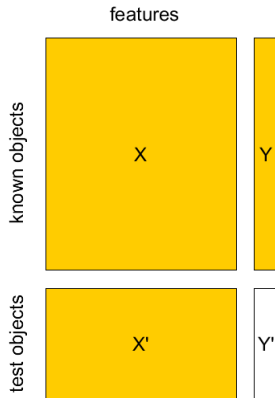
$$L(\hat{w}|X, Y) < L(\hat{w}|X', Y')$$

- Эффект растет с ростом переобучения.
- Как получить реалистичную оценку $L(\hat{w}|X', Y')$?
 - на отдельной *валидационной выборке* (hold-out)
 - кросс-проверка, кросс-валидация (cross-validation)
 - скользящий контроль (leave-one-out)

- 5 Оценка качества на тесте
 - Отдельная валидационная выборка
 - Кросс-валидация

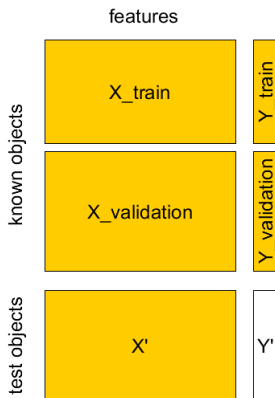
Отдельная валидационная выборка

- Обучающая выборка $X, Y: (\mathbf{x}_1, y_1), \dots (\mathbf{x}_M, y_M)$
- Тестовая выборка $X', Y': (\mathbf{x}'_1, y'_1), \dots (\mathbf{x}'_K, y'_K)$



Отдельная валидационная выборка

Разделим обучающую выборку на ту, где будем обучать модель и оценивать случайно:



Комментарии

- I_{train}, I_{val} - индексы объектов обучающей/валидационной выборки.
- Настройка параметров по

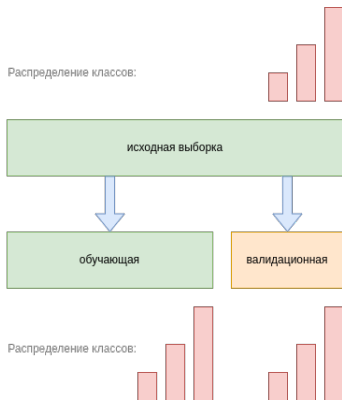
$$L(w|X_{train}, Y_{train}) = \frac{1}{|I_{train}|} \sum_{i \in I_{train}} \mathcal{L}(f_w(\mathbf{x}_i), y_i) \rightarrow \min_w$$

- Оценка модели:

$$L(w|X_{val}, Y_{val}) = \frac{1}{|I_{val}|} \sum_{i \in I_{val}} \mathcal{L}(f_w(\mathbf{x}_i), y_i)$$

Стратификация

Стратификация (stratification) - разбиение с сохранением априорного распределения классов / дискретного признака.



- 5 Оценка качества на тесте
 - Отдельная валидационная выборка
 - Кросс-валидация

Пример: 4х блоковая кросс-валидация

X	Y
1	1
2	2
3	3
4	4

Разделим обучающую выборку на K частей (блоков) ($K = 4$).

- перед разбиением важно перемешать объекты
- используется предположение независимости объектов

Пример: 4х блоковая кросс-валидация

X	Y
1	1
2	2
3	3
4	4

Блоки 1,2,3 для обучения, а блок 4 - для прогнозов.

Пример: 4х блоковая кросс-валидация

X	Y
1	1
2	2
3	3
4	4

Блоки 1,2,4 для обучения, а блок 2 - для прогнозов.

Пример: 4х блоковая кросс-валидация

X	Y
1	1
2	2
3	3
4	4

Блоки 1,3,4 для обучения, а блок 2 - для прогнозов.

Пример: 4х блоковая кросс-валидация

X	Y
1	1
2	2
3	3
4	4

Блоки 2,3,4 для обучения, а блок 1 - для прогнозов.

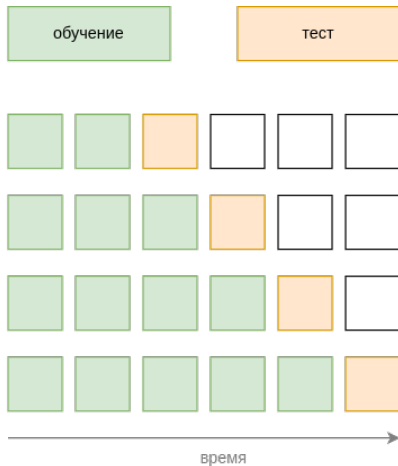
Варианты применения

- Частный случай - скользящий контроль (leave-one-out):
 $K = N^3$.
- Обычно $K \in [3, 10]$.
- ShuffleSplit: усреднение качества на валидации при K -кратном случайном разбиении на обучение и валидацию.
- После подбора гиперпараметров мы можем
 - настраивать модель с наилучшими гиперпараметрами на всех данных
 - либо строить прогноз как усреднение по уже настроенным K моделям.
 (медленнее, зато получаем \hat{y} и std. отклонение!)
- точность 2х подходов нужно сверять эмпирически

³Для какого изученного метода его можно посчитать быстро?

Временной ряд

- Для временного ряда качество оценивается по прогнозам только вперёд без перемешивания:



Содержание

- 1 Кодирование признаков
- 2 Генерация признаков
- 3 Представление текстов
- 4 Переобучение и недообучение
- 5 Оценка качества на тесте
- 6 Оценка качества регрессии**

MSE, RMSE

Средне-квадратичная ошибка (mean squared error, MSE):

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^N (\hat{y}(x_n) - y_n)^2$$

Корень из средне-квадратичной ошибки (root mean squared error, RMSE)

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}(x_n) - y_n)^2}$$

RMSE лучше MSE, т.к. измеряет ошибку в той же размерности, что и y .

Коэффициент детерминации

- Коэффициент детерминации R^2 :

$$R^2 = 1 - \frac{\frac{1}{N} \sum_{n=1}^N (\hat{y}(x_n) - y_n)^2}{\frac{1}{N} \sum_{n=1}^N (y_n - \bar{y})^2}$$

- R^2 принимает значения
 - от $-\infty$ (худший прогноз)
 - до 1 (идеальный прогноз)
 - 0 - качество прогноза равно константе

MSE

- Средняя абсолютная ошибка (mean absolute error, MAE):

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |\hat{y}(x_n) - y_n|$$

- Настройка по MAE устойчивее к выбросам, чем MSE.

MSE

- Средняя абсолютная ошибка (mean absolute error, MAE):

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^N |\hat{y}(x_n) - y_n|$$

- Настройка по MAE устойчивее к выбросам, чем MSE.
- Средняя процентная ошибка
(mean absolute percentage error, MAPE):

$$\text{MAPE} = \frac{1}{N} \sum_{n=1}^N \frac{|\hat{y}(x_n) - y_n|}{|y_n|}, \quad \text{MAPE}' = \frac{1}{N} \sum_{n=1}^N \frac{|\hat{y}(x_n) - y_n|}{\max\{|y_n|, a\}}$$

- $a > 0$ - малая константа, чтобы не делить на ноль.

WAPE

- Взвешенная средняя процентная ошибка (weighted average percentage error, WAPE):

$$\text{WAPE} = \frac{1}{N} \frac{\sum_{n=1}^N |\hat{y}(x_n) - y_n|}{\sum_{n=1}^N |y_n|}$$

- Это микроусреднение по откликам, а не макроусреднение, как в MAPE.
- Полезна при прогнозировании спроса на товары, когда существует много наблюдений, когда товар вообще не покупался.

Доля плохих прогнозов

- Доля плохих прогнозов:

$$\text{BadFreq} = \frac{1}{N} \sum_{n=1}^N \mathbb{I}\{|\hat{y}(x_n) - y_n| > h\}$$

- Доля плохих относительных прогнозов:

$$\text{RelBadFreq} = \frac{1}{N} \sum_{n=1}^N \mathbb{I}\left\{\frac{|\hat{y}(x_n) - y_n|}{|y_n|} > h\right\}$$