Факультет компьютерных наук | Машинное обучение | Москва 2026

Лекция 3
Предобработка данных и конструирование признаков

Преподаватель: Меликян Алиса Валерьевна, amelikyan@hse.ru
кандидат наук, доцент Департамента программной инженерии ФКН НИУ ВШЭ,
академический руководитель магистерской программы
«Искусственный интеллект и продуктовый подход в HR-менеджменте»

Предобработка данных

Предобработка данных (Preprocessing) приводит данные в корректный и удобный для модели вид, не добавляя новой информации:

- Очистка: удаление/импутация пропусков, удаление дубликатов, обработка выбросов;
- Преобразование формата данных (строки → числа, даты → datetime);
- Масштабирование числовых признаков;
- Кодирование категориальных признаков.

Конструирование признаков

Конструирование признаков (Feature Engineering) – по сравнению с предобработкой данных более творческий этап, который включает в себя создание, преобразование и отбор признаков из исходных данных для улучшения модели.

Например, из «даты» формируется признак «день недели» или «число дней с момента наступления события», из текста – количество ключевых слов, или создаются агрегаты и взаимодействия между признаками.

Проблемы в данных

Student ID	Student Name	Age	GPA	Classification
100122014	Joseph	21	3.5	Junior
100232015	Patrick	200	3.2	Sophomore
100122012	Seller	24	3.0	Senior
100342013	Roger	23	234	Senior
100942012	Davis	2.8	3.7	Sophomore
	Travis	23	3.4	Sr
100982015	Alex	27		Sophomore
100982013	Trevor	-22	4.0	Senior
AUC2016XC	Aman	30	3.5	Jr

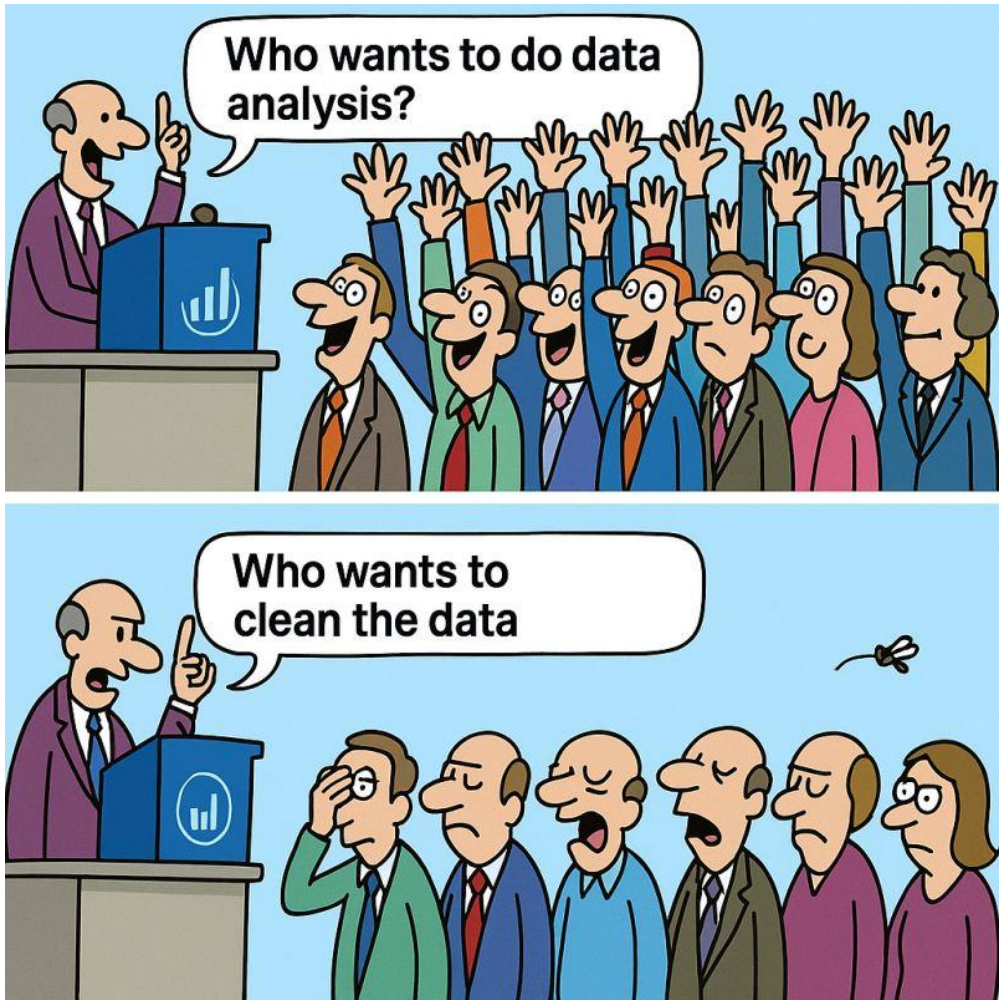
Missing Data

Inconsistent Data

Noisy Data

Очистка данных

Очистка данных (Data Cleaning) – важный этап предобработки. Заниматься им не всегда интересно, но необходимо. Чаще всего мы работаем с неидеальными данными: они могут содержать ошибки, быть неполными или некорректно подготовленными. Даже хорошо собранные датасеты нередко требуют дополнительной очистки для решения конкретных аналитических задач.



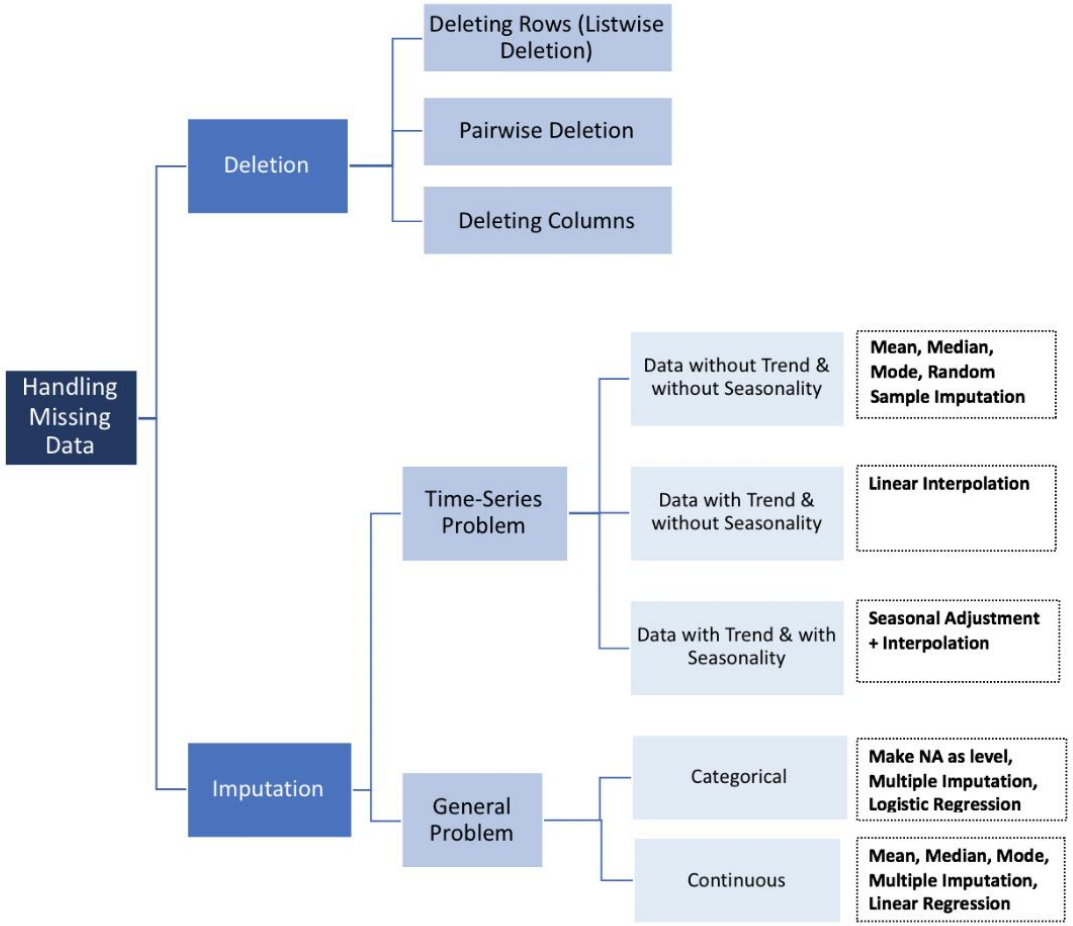
Пропуски

Почему появляются?

- Ошибки при сборе или переносе/записи данных
- Некачественная работа с данными
- Отсутствие информации о чем-либо и т.д.

Почему вредят?

- Сокращение числа значений для анализа
- Смещенность выборки и выборочных характеристик (выборочное среднее, выборочная дисперсия, …)
- Проблемы с визуализацией



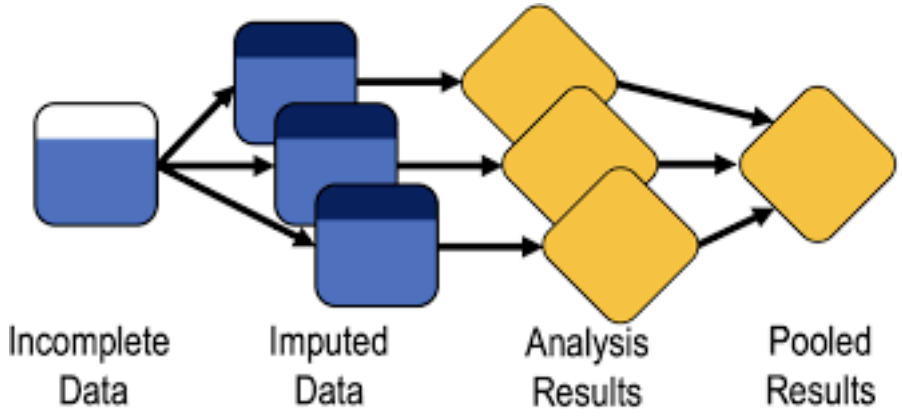
Способы работы с пропусками

1. Удаление пропусков, если их немного и потеря данных не критична.
2. Заполнение статистическими значениями (среднее, медиана, мода), можно по группам.
3. Заполнение фиксированным значением (например, 0, -1 или ”Unknown”).
4. Заполнение с помощью модели (KNN, линейная/логистическая регрессия).
5. Множественная иммутация (Multiple Imputation): создание нескольких заполненных вариантов данных и объединение результатов для повышения надежности.

Множественная импутация

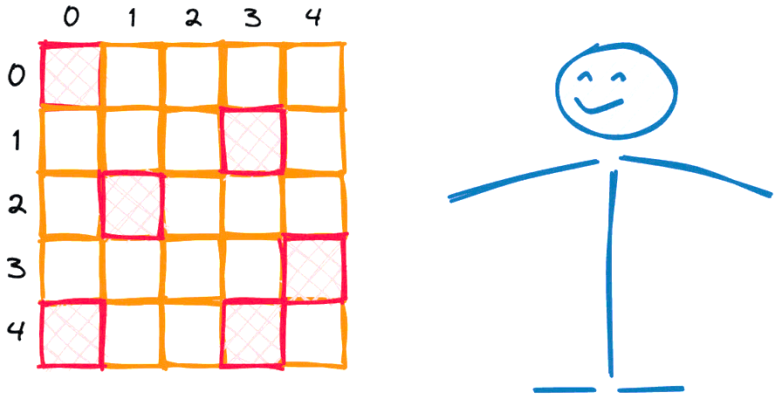
Вместо одного заполнения пропусков мы:

- Создаём несколько разных версий датасета.
- В каждой версии пропуски заполняются случайно, но осмысленно (на основе распределений и других признаков).
- Обучаем модель на каждом датасете.
- Объединяем результаты (усредняем коэффициенты, предсказания и т.д.).
- Так учитывается неопределённость, связанная с пропущенными данными.

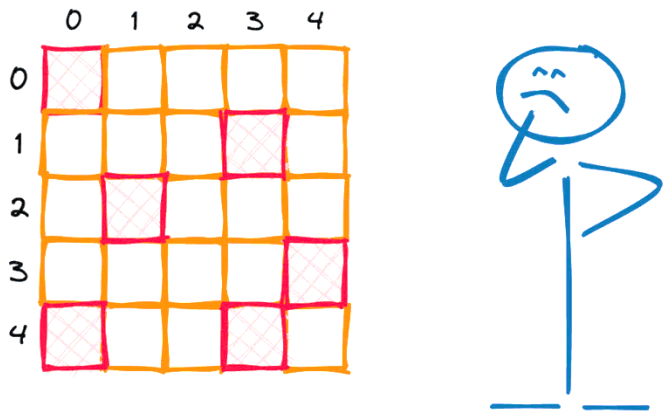


Способы работы с пропусками

Missing values, let's impute them quickly



I must understand WHY do I have missing values before imputing them



Дубликаты в данных

Дубликаты в данных - это объекты, которые полностью или частично повторяются в наборе данных. Они увеличивают размер данных без пользы, искажают результаты анализа и могут привести к неправильным выводам.

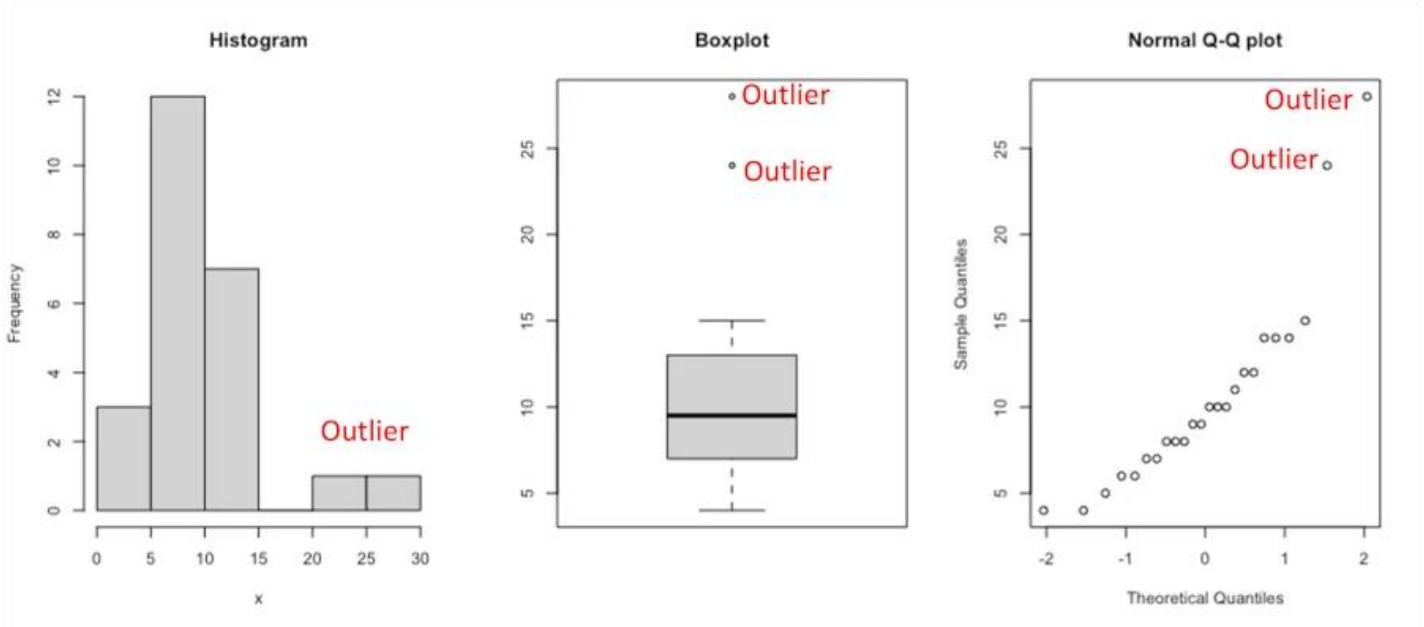
- Полные дубликаты: записи, у которых все поля совпадают.
- Частичные дубликаты: записи, совпадающие по определённым ключевым полям, но отличающиеся по другим.

Как обрабатывать дубликаты?

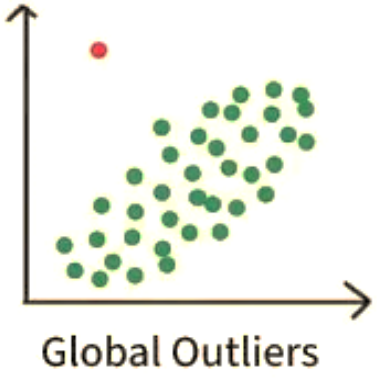
- Удалять: если дубликаты – ошибка.
- Объединять: если разные записи дополняют друг друга.
- Оставлять: если дубликаты обоснованы (например, повторные покупки одного клиента).

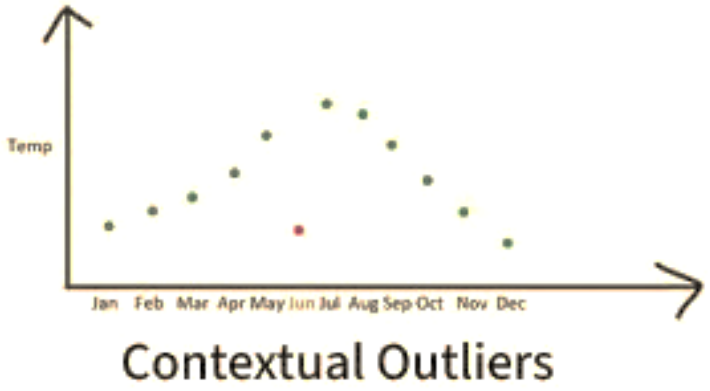
Выбросы

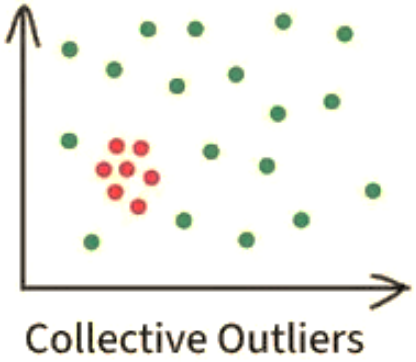
Выбросы – это значения признака, которые существенно отличаются от основной массы данных. Они могут быть как редкими, но валидными наблюдениями, так и ошибками измерений. Из-за них искажаются статистики (среднее, дисперсия) и ухудшается обучение ряда моделей (особенно линейных и distance-based).



Виды выбросов







- Глобальные выбросы (Global Outliers) – отдельные точки, которые сильно отличаются от всей совокупности данных и легко обнаруживаются.
- Контекстные выбросы (Contextual Outliers) – значения, являющиеся аномальными только в определённом контексте (например, температура в конкретный месяц).
- Коллективные выбросы (Collective Outliers) – группа точек, которые по отдельности могут выглядеть нормально, но вместе образуют аномальный кластер.

Обработка выбросов: тримминг

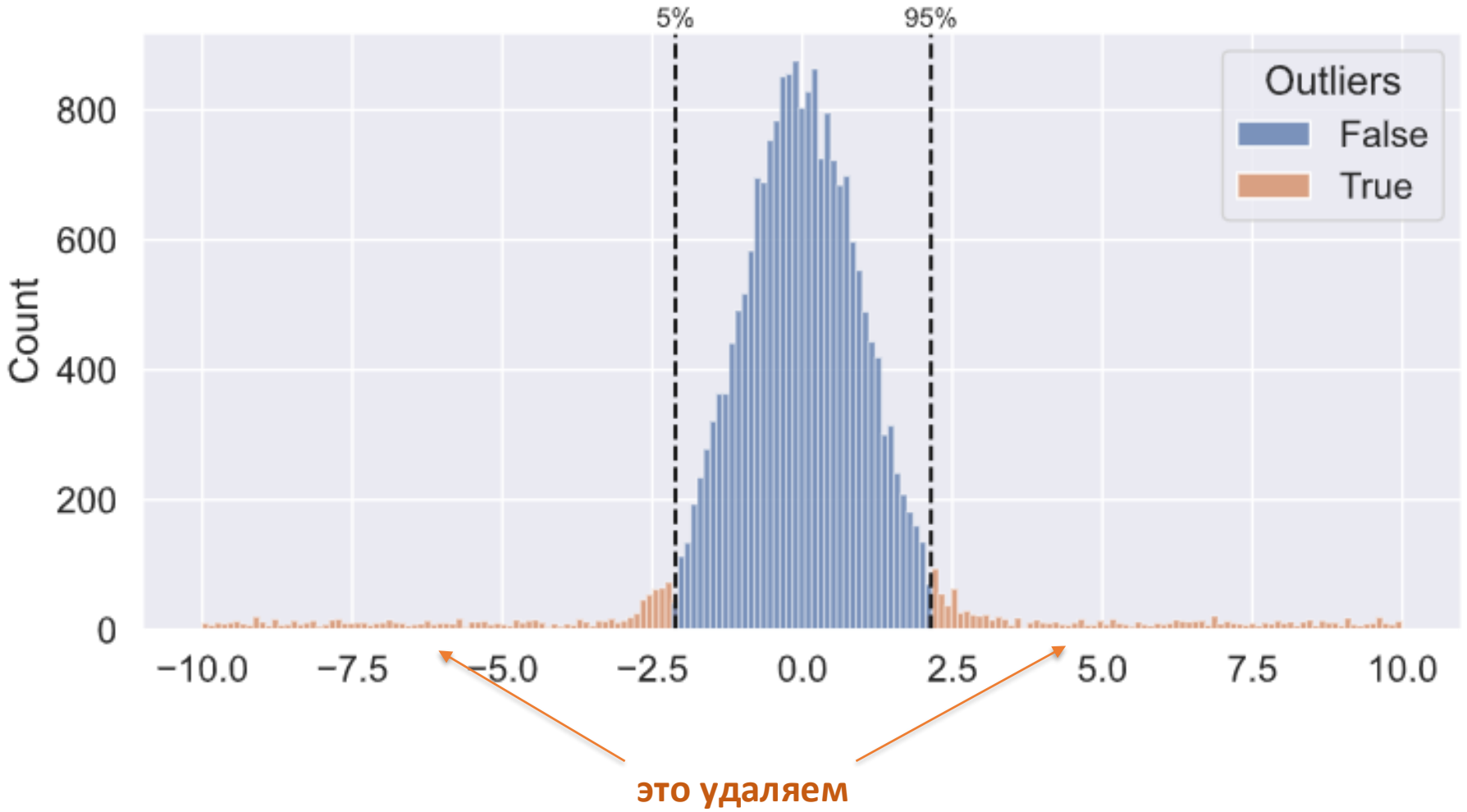
Тримминг (Trimming) – удаление из выборки крайних значений с одной или обеих сторон распределения. Часто удаляют фиксированный процент (например, по 5%) с нижнего и верхнего концов. Это простой и быстрый способ снизить влияние экстремальных значений. Однако есть ограничения:

- снижается размер выборки;
- несколько искажается исходное распределение;
- теряется информация об истинной вариации значений признака.

Нужно осторожно подбирать процент удаляемых данных, чтобы не нарушить их репрезентативность и не получить в результате смещенные оценки модели. Не рекомендуется применять если:

- выборка мала и важно каждое значение;
- выбросы – часть изучаемого явления;
- нет уверенности, что удаляемые значения не несут важной информации.

Обработка выбросов: тримминг



Обработка выбросов: клиппинг

Клиппинг (Clipping) – ограничение значений признака сверху и/или снизу некоторыми порогами. Пороги задаются заранее, могут быть основаны на физических ограничениях, бизнес-правилах, знаниях предметной области и пр.

Значения, выходящие за пределы допустимого диапазона, заменяются на соответствующие граничные значения. Размер выборки при этом не изменяется.

Клиппинг в HR

Клиппинг могут использовать в HR чтобы:

- средние значения не искажались;
- модели прогнозирования (attrition, time-to-hire) были устойчивее;
- KPI рекрутеров не занижались из-за единичных аномалий.

Он позволяет не учитывать кейсы, не отражающие типичный процесс.

Задача: проанализировать время закрытия вакансий (time-to-hire).

Данные (в днях): 12, 15, 18, 20, 22, 25, 30, 35, 40, 180

Значение 180 дней – выброс (редкий случай: заморозка вакансии, ошибка ввода, топ-позиция).

Бизнес решил, что минимальное значение – 10 дней, а максимальное – 60.

После клиппинга: 12, 15, 18, 20, 22, 25, 30, 35, 40, 60

180 заменили на 60, а не удалили строку.

Обработка выбросов: винзоризация

Винзоризация (Winsorization) – это версия клиппинга, основанная на процентилях.

Все значения ниже 5-го процентиля → значение 5-го процентиля.

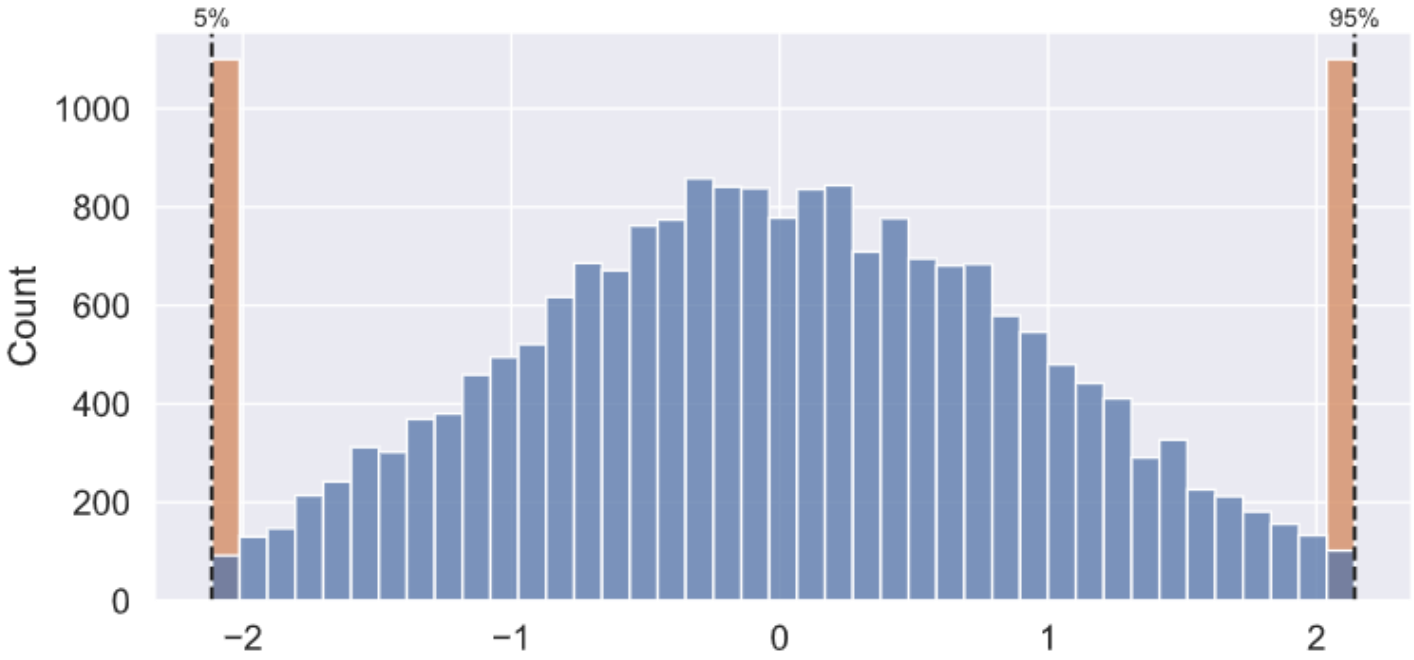
Все значения выше 95-го → значение 95-го

Исходные данные:

[1, 2, 3, 4, 5, 100]

Винзоризация (5–95%):

[2, 2, 3, 4, 5, 5]



Обработка выбросов: специальные модели

Выбросы можно обрабатывать не только предобработкой, но и выбором модели, которая к ним устойчива или на них ориентирована.

Робастные модели (устойчивые к выбросам) не пытаются найти выбросы, но их влияние на обучение минимально (Деревья и ансамбли, Random Forest – усреднение снижает эффект выбросов, Gradient Boosting.

Модели, которые **обнаруживают выбросы** (Anomaly Detection). Для них выбросы – объект интереса (Isolation Forest, DBSCAN).

Примеры конструирования признаков

1. **Создание признаков (Feature Creation)** путем модификации или агрегации существующих признаков или вычисления новых на основе комбинации значений нескольких признаков.
2. **Трансформация признаков (Feature Transformation)**: масштабирование, нелинейные преобразования, биннинг.
3. **Отбор признаков (Feature Selection)**, например, на основе корреляций или парных статистических тестов.

Создание признака: взаимосвязи признаков

Есть данные клиентов банка для предсказания их ухода по признакам: возраст, баланс, количество продуктов. Можно сгенерировать новый признак – соотношение баланса к возрасту (баланс/возраст), который поможет выявить зависимости и улучшить точность модели. Например, молодые клиенты с высоким балансом могут быть менее лояльными.

Для предсказания риска увольнения можно сгенерировать признаки:

- отношение стажа к возрасту;
- количество проектов на год стажа;
- средняя продолжительность работы на проекте;
- отношение отпуска к стажу;
- разница в возрасте сотрудника и начальника;
- совпадение пола сотрудника и начальника.

Создание признака: агрегация

Вычисляются статистические показатели, такие как среднее, сумма или медиана.

- **Средний доход клиента на основе его транзакций** может стать полезным признаком для предсказания его финансового поведения.
- **Средняя продолжительность работы сотрудника** в компании (например, по предыдущим позициям или отделам) может помочь спрогнозировать его лояльность и риск увольнения.
- **Количество пропущенных дней в месяц** важный показатель вовлечённости и удовлетворённости работой.
- **Среднее количество проектов, в которых участвует сотрудник** за последний год, отражает загруженность и опыт.
- **Максимальная оценка эффективности (performance score)** за последние полгода – показатель потенциала и качества работы.

Создание признака: полиномиальные признаки

Полиномиальные признаки создаются путём возведения признаков в степень или их комбинирования, помогая моделям улавливать нелинейные зависимости.

Взаимодействие между количеством проектов и пропущенными днями
Произведение этих признаков отражает степень нагрузки: сотрудник с большим числом проектов и частыми пропусками может быть подвержен стрессу и выгоранию.

Опыт работы и зарплата
Опыт работы и опыт в квадрате позволяют модели уловить, что зарплата растёт с опытом неравномерно – сначала быстро, затем замедляется.

Создание признака: комбинирование категориальных признаков

Категориальные признаки принимают дискретные значения, обозначающие различные категории или классы (например, «город», «тип недвижимости», «отдел в компании» и т.д.). Иногда их комбинация позволяет учесть взаимодействия между признаками, которые влияют на качество предсказаний.

1. Анализ текучести сотрудников

Комбинация категориальных признаков «Отдел» и «Уровень должности» может выявить, что именно junior в отделе продаж увольняются чаще, чем senior в том же отделе или junior в других отделах.

2. Оценка эффективности обучения

Комбинация признаков «Тип обучения» и «Должность» может помочь понять, какой формат обучения более эффективен для каждой должности. Например, менеджерам офлайн обучение подходит лучше, а инженерам – онлайн.

Создание признака: знания из предметной области

Иногда применяются **знания из предметной области (Domain Knowledge).** Это позволяет создавать фичи, специфичные для задачи. Например, в финансовом анализе полезен коэффициент текущей ликвидности, который учитывает соотношение активов и обязательств компании.

Это один из самых ценных этапов Feature Engineering. В HR-аналитике это особенно важно, потому что сырые данные о сотрудниках сами по себе редко напрямую отражают реальные процессы: мотивацию, выгорание, риск увольнения, потенциал роста. Именно экспертиза HR-специалистов позволяет превратить данные в информативные признаки.

На основе понимания реальных бизнес-процессов и поведения людей создаются новых признаки, которые:

- логически связаны с целевой переменной;
- не лежат на поверхности в исходных данных;
- отражают причины, а не только следствия.

Создание признака: знания из предметной области

Задача: предсказать вероятность увольнения сотрудника в ближайшие 3–6 месяцев.
Исходные данные: дата найма; зарплата; отдел; должность; дата последнего повышения; количество больничных; оценка performance; количество сверхурочных часов.

Новые признаки:
- стаж < 6 месяцев (высокий риск увольнения в начале работы);
- стаж > 3 лет без повышения.
- повышение за последний год;
- зарплата ниже медианы по роли;
- много переработок и падение performance;
- средняя текучесть в команде руководителя;
- карьерный потолок.

Трансформация числового признака: Масштабирование (нормализация)

Масштабирование (Scaling) – это один из важнейших шагов трансформации, особенно для алгоритмов, чувствительных к масштабу признаков (например, линейная регрессия, k-ближайших соседей, k-means). Суть масштабирования в том, чтобы привести все признаки к единому диапазону значений.

Нормализация (Min-Max Scaling): приводит значения признака в диапазон от 0 до 1.

Формула:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

x – исходное значение;
min(x) – минимальное значение;
max(x) – максимальное значение;
x′ – нормализованное значение.

Трансформация числового признака: Масштабирование (нормализация)

#	Emp	Age	Salary
1	Emp1	44	73000
2	Emp2	27	47000
3	Emp3	30	53000
4	Emp4	38	62000
5	Emp5	40	57000
6	Emp6	35	53000
7	Emp7	48	78000

Normalization

Age	Normalized Age	Salary	Normalized Salary
44	0.80952381	73000	0.838709677
27	0	47000	0
30	0.142857143	53000	0.193548387
38	0.523809524	62000	0.483870968
40	0.619047619	57000	0.322580645
35	0.380952381	53000	0.193548387
48	1	78000	1
	Range 0-1		**Range 0-1**

How to calculate Normalized value?
X = 35, min = 27, max = 48 for column Age.
$Xnorm(\text{for } 35) = \frac{35-27}{48-27} = 0.3809$

Трансформация числового признака: Масштабирование (стандартизация)

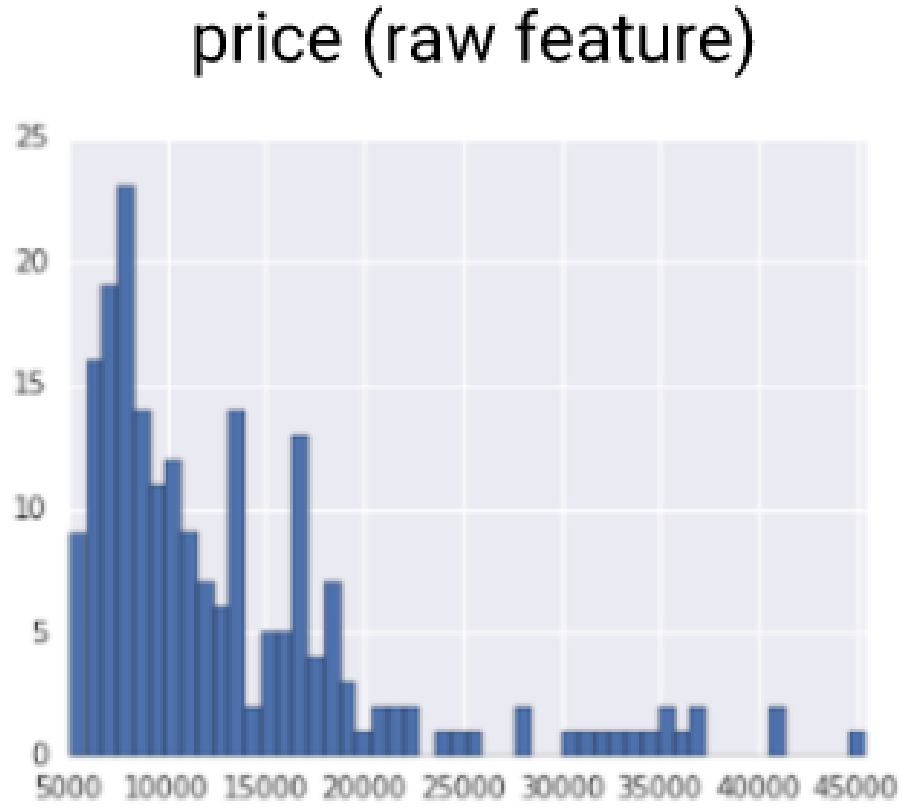
Стандартизация (Standardization): преобразует данные так, чтобы они имели нулевое среднее значение и стандартное отклонение, равное единице. Это часто используется в алгоритмах, которые предполагают нормальное распределение данных.

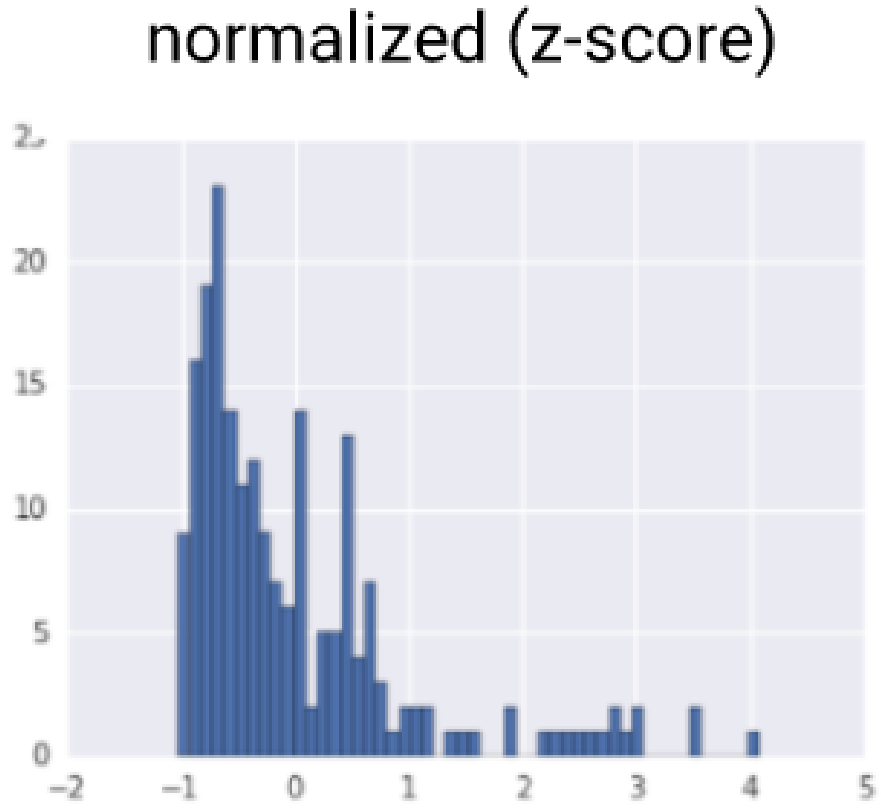
Формула:

$$x_{\text{std}} = \frac{x - \mu}{\sigma}$$

x – исходное значение;
μ – среднее значение;
σ – стандартное отклонение;
x_{std} – стандартизованное значение признака.

Трансформация числового признака: Масштабирование (стандартизация)





Трансформация числового признака: Масштабирование (робастное)

Робастное масштабирование (Robust Scaling) применяется для данных с выбросами, которые могли остаться после очистки. Минимизирует их влияние. Использует медиану и интерквартильный размах вместо среднего и стандартного отклонения.

Формула:

$$X_{\text{robust}} = \frac{X - X_{\text{median}}}{\text{IQR}}$$

X – исходное значение;
X_{median} – медианное значение;
IQR (межквартильный размах) – разница между 75-м и 25-м процентилями (Q3 − Q1);
X_{robust} — масштабированное значение признака.

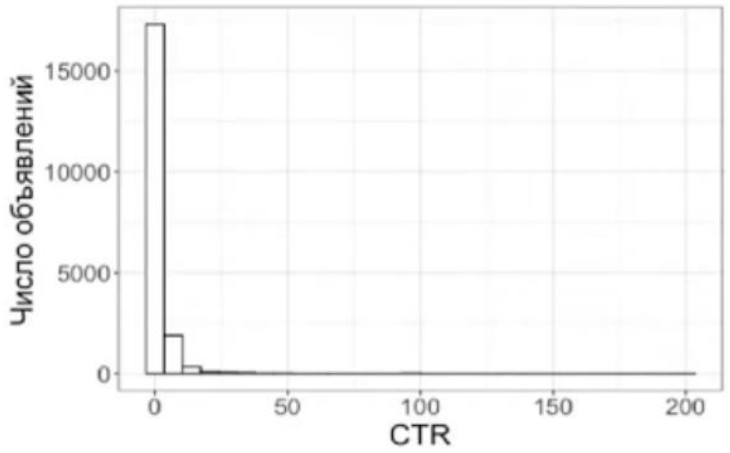
Трансформация числового признака: Логарифмирование

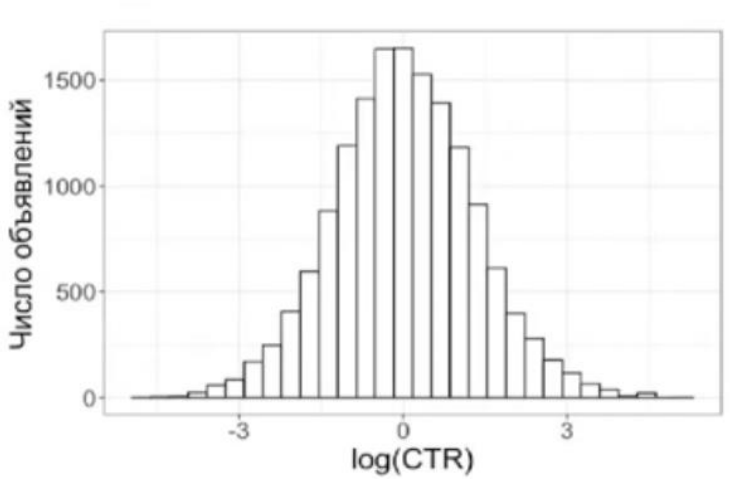
Логарифмическая трансформация: применяется для признаков с сильно скошенным распределением, где существует длинный хвост справа, чтобы сделать данные более симметричными.

Каждый элемент признака заменяется на log(x+1), где log может быть натуральным логарифмом (по основанию e) или логарифмом по другому основанию (например, по основанию 10).

Применяется, например, если есть данные о доходах, которые сильно варьируются (большинство людей имеют низкие доходы, но есть несколько с очень высокими доходами). Логарифмическое преобразование может помочь нормализовать данные.

Трансформация числового признака: Логарифмирование





X	log(x)
1	0
2	0.69
3	1.10
100	4.61
1000	6.91

$$\text{CTR} = \frac{\text{Количество кликов}}{\text{Количество показов}} \times 100\%$$

Трансформация числового признака: квадратный корень

Квадратный корень: используется для сглаживания скошенного распределения данных, особенно когда значения признаков варьируются в широком диапазоне. Это помогает уменьшить влияние больших значений на модель, но менее радикально, чем логарифмическая трансформация.

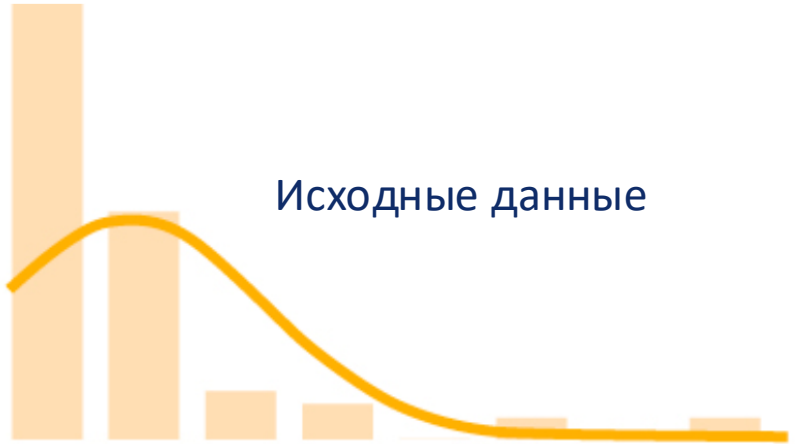
Каждый элемент признака заменяется на $\sqrt{x}$

При анализе текучести персонала можно использовать признак «количество больничных дней за год»: у большинства сотрудников он невелик (0–5 дней), но у небольшой группы может достигать 30–40 дней, что искажает модель. Применение преобразования позволяет уменьшить влияние таких больших значений, сохранив при этом различия между малыми значениями, которые наиболее информативны.

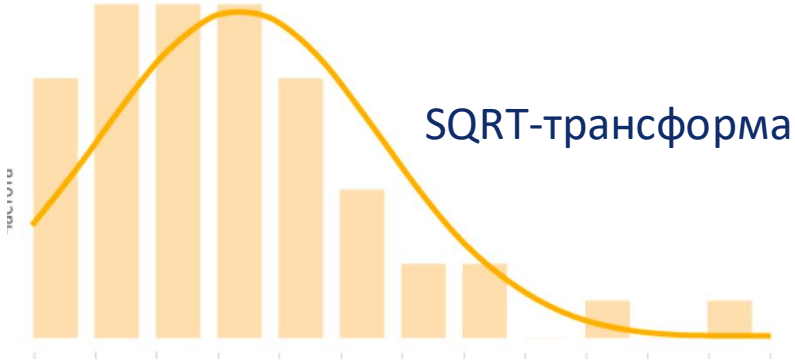
Log- или SQRT-трансформация

Log-трансформация

Исходные данные



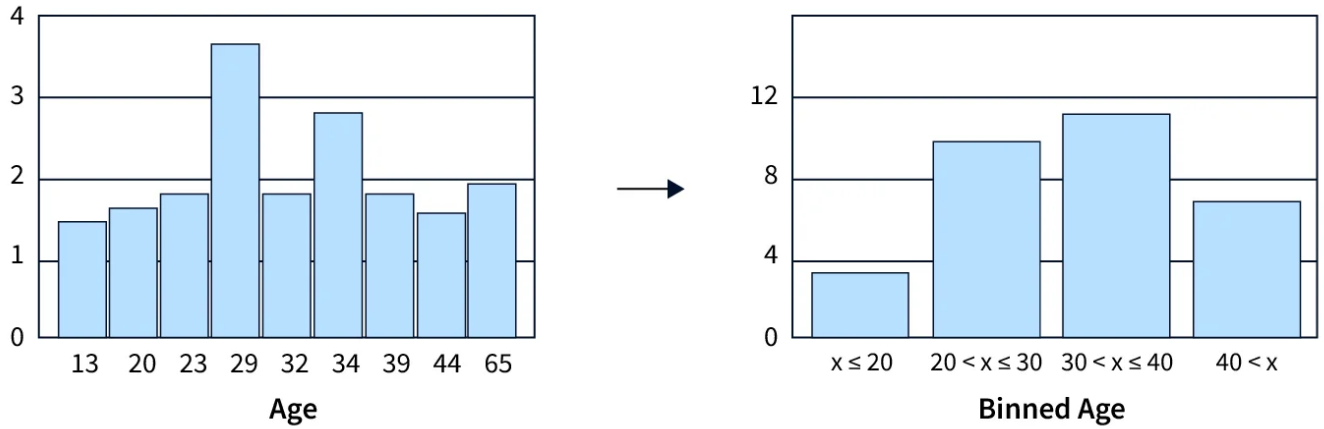
SQRT-трансформация



Трансформация числового признака: бининг

Бининг (Binning) разбивает числовые значения на несколько интервалов (бинов). Каждому значению присваивается номер или метка соответствующего интервала.

- Снижает сложность и шум в данных за счёт группировки похожих значений.
- Помогает моделям лучше выявлять закономерности, особенно для алгоритмов, которые плохо работают с непрерывными переменными.
- Выбросы попадают в отдельные бины, что облегчает их обработку.



Трансформация числового признака: бининг

- **Равномерный бининг** (Equal-width): делит весь диапазон значений на интервалы одинаковой ширины.
- **Равновероятный бининг** (Equal-frequency): делит данные так, чтобы в каждом бине было примерно одинаковое количество наблюдений.
- **Произвольный бининг**: интервалы задаются вручную или на основе бизнес-логики.

Исходный набор значений:
[1, 3, 7, 8, 12, 15, 18, 22, 27, 30]

Равномерный бининг с 3 бинами:

Бин 1: 1–10

Бин 2: 11–20

Бин 3: 21–30

Каждое значение заменяется номером бина.

Трансформация числового признака: бинаризация

Бинаризация – это преобразование числового признака в бинарный (0/1) по определённому правилу, чаще всего по порогу.

Возраст → признак "совершеннолетний":

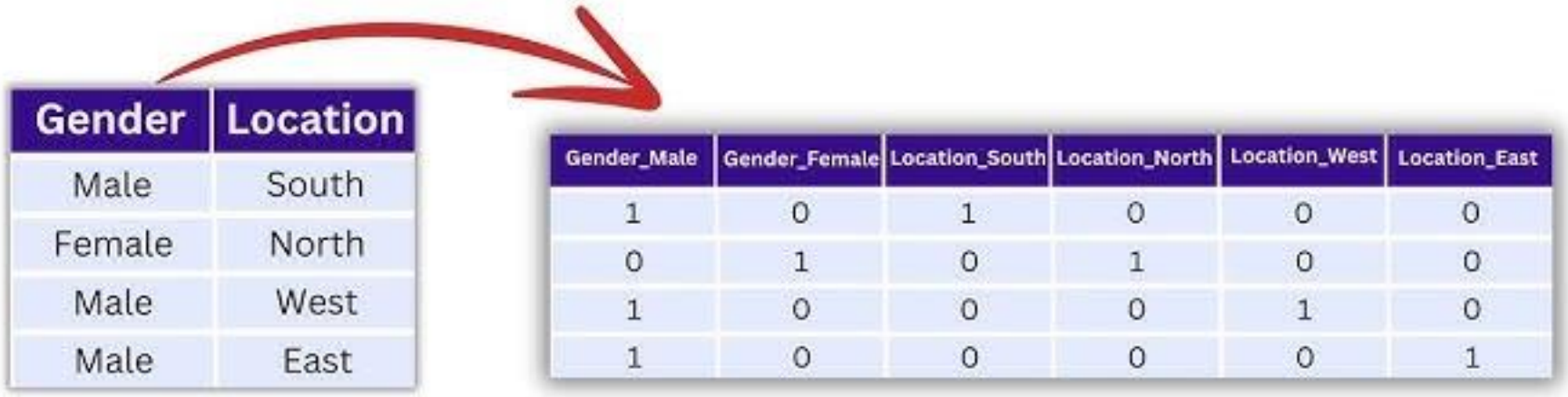
- 1, если возраст ≥ 18
- 0, если < 18

Варианты выбора порога:

- Фиксированный (экспертный): 100 000 → высокий доход;
  температура ≥ 37 → повышенная
- По среднему или медиане
- По квантилям

Трансформация категориального признака: One-Hot Encoding

One-Hot Encoding (OHE) преобразует категориальные признаки в бинарные, создавая новый столбец для каждой уникальной категории. Рекомендуется использовать, когда категориальные признаки не имеют порядка и количество категорий не слишком велико.



Gender	Location
Male	South
Female	North
Male	West
Male	East

Gender_Male	Gender_Female	Location_South	Location_North	Location_West	Location_East
1	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	0	0	0	0	1

Трансформация категориального признака: Label Encoding

Label Encoding присваивает каждой категории уникальное числовое значение. Присваивание чисел не несёт смысловой нагрузки.

Признак: название отдела

HR → 0

IT → 1

Sales → 2

Проблема: модель может посчитать, что: Sales > IT > HR, хотя никакого порядка между отделами нет.

Можно использовать когда порядок категорий не имеет смысла.

Трансформация категориального признака: Ordinal Encoding

Ordinal Encoding применяется для категорий, которые имеют естественный порядок. Каждой категории присваивается числовое значение в соответствии с их порядком. Например, «Высокий» = 0, «Средний» = 1, «Низкий» = 2.

Стоит использовать, когда порядок категорий важен для модели.

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

Трансформация категориального признака: Target Encoding

Target Encoding заменяет категории числовыми значениями, основанными на целевой переменной. Для каждой категории вычисляется среднее значение целевой переменной, и это значение используется как новый числовой признак. Используют когда категориальный признак имеет много уникальных значений, и поэтому классическое one-hot кодирование становится неэффективным.

workclass	target
State-gov	0
Self-emp-not-inc	1
Private	0
Private	0
Private	1

workclass	target mean
State-gov	0
Self-emp-not-inc	1
Private	1/3

workclass
0
1
1/3
1/3
1/3

Трансформация категориального признака: Frequency Encoding

Frequency Encoding преобразует категории в числовые значения, основанные на частоте их появления в данных. Например, если «HR-отдел» встречается в 50% записей, ему присваивается значение 0.5.

Стоит использовать, когда важно сохранить информацию о распределении категорий в данных и минимизировать размерность.

Height	Sex
173.1	Male
160.4	Female
178.5	Male
155.5	Female
163.7	Female

Frequency Encoding →

Height	Sex
173.1	0.4
160.4	0.6
178.5	0.4
155.5	0.6
163.7	0.6

Трансформация категориального признака: Rare Label Encoding

Rare Label Encoding объединяет редкие категории в одну. Категории, которые встречаются реже определенного порога (например, менее 1% данных), объединяются в одну новую категорию (например, «Другое»).

Стоит использовать, когда существует большое количество категорий и некоторые из них встречаются очень редко.

Задаём порог, например, минимум 20 наблюдений.

После кодирования все вакансии с числом кандидатом < 20 → группа Rare.

Отбор признаков

При отборе признаков учитываем следующее:

- взаимосвязь признаков с целевой переменной,
- минимизация мультиколлинеарности,
- улучшение интерпретируемости модели.

Это включает корреляционный анализ, статистические тесты, ранжирование признаков, регуляризацию.

Порядок проведения

Если выполняем предобработку, то :

- обучаем (fit) все шаги предобработки только на train;
- применяем (transform) те же самые шаги к train и к test.

! не делаем заново предобработку на test.

- Заполнение пропусков: считаем на train медиану, заполняем пропуски в test этой же медианой.
- Масштабирование: считаем среднее и стандартное отклонение на train, масштабируем данные в test с теми же параметрами
- Кодирование категорий: на train определяем набор категорий, на test применяем то же кодирование

Train → fit + transform

Test → transform only

Эту логику автоматически гарантирует Pipeline в sklearn.

Вопросы на размышление

- Вам дали датасет от бизнеса. С чего вы начнёте работу с данными и почему?
- Если в данных есть пропуски, вы всегда будете их удалять? Когда это плохая идея?
- Вам дали таблицу с зарплатами, но часть значений – строки с текстом "не указано". Что будете делать?
- В датасете есть колонка с возрастом, и там встречаются значения -5 и 300. Как вы это объясните и обработаете?
- Если один признак измеряется в тысячах, а другой – в долях, что может пойти не так?

Вопросы на размышление

- В данных есть признак «город» с 500 уникальными значениями. Будете ли вы делать one-hot encoding?
- Почему нельзя просто заменить категории на числа: A=1, B=2, C=3?
- Можно ли использовать среднее значение по всему датасету для заполнения пропусков?
- Модель показывает 99% accuracy. Какие вопросы вы зададите в первую очередь?
- После предобработки качество стало хуже. Это нормально? Что будете проверять?
- Что важнее: сложная модель или качественная предобработка и почему?

Вопросы на размышление

- Почему масштабирование нужно делать после train/test split?
- Какие ошибки в данных модель никогда не сможет «простить»?
- Что из предобработки вы автоматизировали бы в первую очередь?
- В данных есть поле «адрес». Как из него сделать полезные признаки?
- Что делать, если команда считает, что конструирование признаков – пустая трата времени? Как доказать обратное?

Спасибо за внимание!