

Лекция 3. Предварительная обработка данных

Тема: Очистка, нормализация, пропуски, дискретизация, отбор признаков

1. Введение

Одним из важнейших этапов любого проекта по анализу данных является **предварительная обработка данных (Data Preprocessing)**.

Как известно, *качество анализа напрямую зависит от качества данных*.

Даже самые современные модели машинного обучения не смогут показать хорошие результаты, если исходные данные содержат ошибки, пропуски или нерелевантные признаки.

По оценкам специалистов, **до 70–80% времени работы аналитика уходит на подготовку и очистку данных**, и лишь 20–30% — на построение моделей и интерпретацию результатов.

Именно поэтому предварительная обработка рассматривается как **ключевой элемент процесса интеллектуального анализа данных (Data Mining)**.

2. Цель и значение предварительной обработки

Основная цель предварительной обработки — подготовить данные к анализу и моделированию, обеспечив их **достоверность, согласованность и пригодность для алгоритмов**.

Основные задачи предварительной обработки:

1. Удаление ошибок, дубликатов и шумов;
2. Обработка пропущенных значений;
3. Приведение данных к единой шкале (нормализация);
4. Кодирование категориальных признаков;
5. Удаление нерелевантных и избыточных признаков;
6. Снижение размерности данных для ускорения анализа.

Хорошо подготовленные данные повышают точность моделей, сокращают время обучения и делают результаты интерпретируемыми.

3. Этапы предварительной обработки данных

Процесс включает несколько последовательных шагов:

1. **Очистка данных (Data Cleaning)**
2. **Преобразование данных (Data Transformation)**
3. **Интеграция данных (Data Integration)**
4. **Сокращение данных (Data Reduction)**

В рамках данной лекции мы подробно рассмотрим ключевые процессы — очистку, нормализацию, обработку пропусков, дискретизацию и отбор признаков.

4. Очистка данных (Data Cleaning)

Очистка данных — это процесс выявления и устранения ошибок, пропусков, дубликатов и несоответствий в данных.

Главная цель — сделать данные достоверными и согласованными.

Основные типы проблем в данных:

- Пропущенные значения (missing values);
- Неверные или противоречивые данные;
- Дублирование записей;
- Шум (ошибочные или случайные значения);
- Разнородные форматы (например, дата в разных форматах).

Методы очистки:

- удаление дубликатов;
- корректировка ошибочных значений;
- фильтрация выбросов (outliers);
- проверка диапазонов и логических условий;
- консолидация данных из разных источников.

Пример:

Если в наборе данных о клиентах встречаются записи с возрастом 300 лет, такие значения необходимо исправить или удалить как очевидную ошибку.

5. Обработка пропусков в данных

Пропуски (missing values) — частое явление в реальных данных. Они могут возникать из-за ошибок сбора, неполных анкет, технических сбоев и других причин.

Неправильная работа с пропусками может привести к искажению результатов анализа.

Способы обработки пропусков:

1. **Удаление записей с пропусками**
 - подходит, если доля пропусков мала (менее 5%).
 - Пример: `df.dropna()` в Python.
2. **Заполнение средними значениями (Mean/Median/Mode Imputation)**
 - числовые признаки можно заменить средним или медианой, категориальные — модой.
3. **Интерполяция или прогнозирование пропусков**
 - заполнение с помощью статистических или машинных моделей.
4. **Использование специальных категорий ("unknown")**
 - применяется для категориальных признаков.

Пример:

Если в данных о доходе клиента есть пропуск, можно заменить его на средний доход группы с аналогичными характеристиками.

6. Нормализация данных

Нормализация (или стандартизация) — это процесс приведения числовых признаков к единому масштабу, чтобы все признаки вносили одинаковый вклад в обучение модели.

Без нормализации признаки с большими числовыми диапазонами могут «доминировать» над другими и исказить результаты анализа.

Основные методы нормализации:

1. **Min-Max нормализация:**

Преобразует данные в диапазон $[0, 1]$:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$
2. **Z-нормализация (стандартизация):**

Приведение данных к нулевому среднему и единичному стандартному отклонению:

$$Z = \frac{X - \mu}{\sigma}$$
3. **Нормализация по вектору (L2-норма):**

Используется в текстовом и пространственном анализе.

Пример:

Если один признак измеряется в тысячах, а другой — в единицах, модель может «предпочесть» первый. Нормализация решает эту проблему.

7. Дискретизация данных

Дискретизация — процесс преобразования непрерывных данных в категориальные (дискретные) интервалы.

Это полезно для упрощения анализа и повышения интерпретируемости моделей.

Методы дискретизации:

1. **Равные интервалы (Equal-width):** диапазон значений делится на равные отрезки.
2. **Равная частота (Equal-frequency):** в каждый интервал попадает одинаковое количество элементов.
3. **Методы на основе кластеризации:** интервалы формируются по естественным кластерам данных.
4. **Эвристические методы:** определение границ на основе доменной логики.

Пример:

Возраст (непрерывный признак) можно дискретизировать как:

- 18–25 лет → «молодые»
 - 26–40 лет → «взрослые»
 - 41–65 лет → «зрелые»
 - 65+ → «пожилые»
-

8. Отбор признаков (Feature Selection)

Отбор признаков — это процесс выбора наиболее информативных переменных, которые действительно влияют на результат.

Он помогает:

- снизить размерность данных;
- повысить точность модели;
- ускорить обучение;
- улучшить интерпретацию результатов.

Основные методы отбора признаков:

1. **Фильтрационные методы (Filter methods):**

– анализ статистических характеристик признаков (корреляция, χ^2 -тест, ANOVA).

2. **Методы обёртки (Wrapper methods):**

– пошаговый перебор признаков с использованием модели (например, RFE — Recursive Feature Elimination).

3. **Встроенные методы (Embedded methods):**

– встроены в алгоритмы (например, Lasso, Random Forest Feature Importance).

Пример:

Если модель прогнозирует успех маркетинговой кампании, признаки «возраст» и «доход» могут быть важны, а «номер клиента» — нет и должен быть исключён.

9. Инструменты и библиотеки для предварительной обработки

Современные инструменты позволяют автоматизировать многие этапы обработки данных:

- **Python:** pandas, NumPy, scikit-learn, Feature-engine
- **R:** dplyr, tidyr, caret
- **Платформы:** RapidMiner, KNIME, Power BI, Tableau Prep

Пример (Python):

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

10. Заключение

Предварительная обработка данных — это фундамент качественного анализа.

Без очистки, нормализации и отбора признаков невозможно построить надёжную и точную модель.

Этот этап требует внимательности, понимания контекста и знания инструментов.

Как говорил Питер Норк, исследователь из Google:

«Больше данных не заменит плохие данные — важнее качество, а не количество».

Таким образом, грамотная подготовка данных — это не техническая рутина, а стратегическая основа успешного анализа и принятия решений.

Список литературы

1. Хэн, Дж., Камбер, М., Пей, Дж. *Интеллектуальный анализ данных: концепции и методы*. — М.: Вильямс, 2019.
2. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. — O'Reilly, 2022.
3. Witten, I. H., Frank, E., Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. — Morgan Kaufmann, 2017.
4. Bishop, C. M. *Pattern Recognition and Machine Learning*. — Springer, 2016.
5. Aggarwal, C. C. *Data Mining: The Textbook*. — Springer, 2015.