4. ПОПЕРЕДНЯ ОБРОБКА ДАНИХ

4.1 Етапи попередньої обробки даних

Застосування методів інтелектуального аналізу даних є багатоетапною процедурою, у якій одним з найважливіших підготовчих етапів є попередня обробка даних.

Передобробка – приведення даних у прийнятний для подальшого використання вигляд. У процесі передобробки дані приводяться у відповідність вимогам, що визначаються специфікою розв'язуваної задачі.

Від якості попередньої обробки залежить можливість одержання якісних результатів наступного процесу аналізу даних. На передобробку може витрачатись до 80% усього часу, відведеного на проект.

Попередня підготовка даних включає:

- очищення;
- оптимізацію даних.

Очищення даних у статистичній вибірці проводиться для факторів, що знижують якість даних та заважають роботі алгоритмів. Воно включає:

– обробку дублікатів, протиріч та фіктивних значень;
– відновлення та заповнення пропущених значень;
– згладжування, пригнічення шуму;
– редагування аномальних значень.

У процесі очищення відновлюються порушення структури, повноти та цілісності даних, перетворюються некоректні формати.

Оптимізація даних як елемент передобробки дозволяє адаптувати дані під конкретну задачу та підвищує ефективність їх аналізу.

Оптимізація включає:

– перетворення даних (масштабування);
– зниження розмірності;
– виявлення та виключення незначних ознак;
– відбір екземплярів.

4.2 Проблеми «брудних» даних

Дані, що мають пошкодження (неточні, неповні, дубльовані, суперечливі, зашумлені), називають «**брудними**».

Джерелами «брудних» даних можуть бути пошкоджені інструменти збору даних, проблеми під час збору та введення вихідних даних, проблеми в каналах передачі даних, обмеження технологій передачі даних тощо.

Проблеми даних поділяють на дві групи, викликані:

– інтеграцією різних джерел даних (Multi-Source Problems);

– проблемами єдиного джерела даних (Single-Source Problems).

Кожна з цих груп може бути поділена на дві самостійні групи, які визначаються:

– недосконалістю схем баз даних, що інтегруються (Schema Level), зокрема, погана архітектура тощо;

– недосконалістю на рівні елементів даних (Instance Level): записів, об'єктів і т.п. – помилки введення даних єдиного джерела, перекриття та суперечливість даних різних джерел.

Найпоширеніші види брудних даних:

– пропущені значення, неповні дані;

– дублікати даних;

– помилкові дані,

– шуми й викиди.

1. Пропущені значення (Missing Values) можуть виникати з ряду причин, таких як:

– дані не були зібрані (наприклад, при анкетуванні схований вік);

– деякі атрибути не можуть бути застосовні для деяких об’єктів (наприклад, атрибут «річний дохід» неможливо застосувати до дитини);

– несправність вимірювального обладнання тощо.

Відсутні значення часто відображаються як значення поза діапазоном можливих значень атрибута (наприклад, від’ємні числа для додатних значень атрибута або нульові числа для ненульових значень атрибута, пропуски або дефіси для категоріальних змінних).

2. Дублювання даних. Набір даних може включати дублікати – записи з однаковими значеннями всіх атрибутів. Наявність дублікатів у наборі даних може бути способом підвищення значимості деяких записів. Однак у більшості випадків продубльовані дані є результатом помилок при підготовці даних.

3. Неточні дані.

3.1 Помилки у даних. Похибки та неточності в даних можуть бути обумовлені свідомими діями людей або обмеженнями комп'ютерних систем обробки даних.

Інколи значення категоріального атрибуту записані неповністю або помилково, що дозволяє припустити існування додаткових значень атрибуту.

Числові атрибути можуть містити помилки, які при візуальному дослідженні можуть виглядати як аномальні викиди.

3.2 Шуми й викиди. Викиди – різко відмінні об'єкти або спостереження в наборі даних. Викиди можуть являти собою окремі спостереження або бути об'єднаними в якісь групи.

Під час підготовки даних корисною є їх візуалізація у вигляді гістограм розподілу значень категоріальних атрибутів, графіків значень та діаграм розсіювання числових атрибутів. Візуалізація дозволяє ідентифікувати викиди, наприклад рік 9999 або вагу –1 кг, які можуть бути або помилками у даних або невідомими закономірностями.

Різні методи Data Mining мають різну чутливість до викидів, що необхідно враховувати при виборі методу аналізу даних. Завдання аналітика не тільки виявити викиди, але й оцінити ступінь їх впливу на результати подальшого аналізу. Якщо викиди є інформативною частиною аналізованого набору даних, використовують робастні методи й процедури. Поширена практика проведення двоетапного аналізу – з викидами та без них – і порівняння отриманих результатів.

У табл. 4.1 наведено деякі приклади "брудних" даних.

Таблиця 4.1 – Приклади "брудних" даних

Місце проблеми	Проблема	«Брудні» дані	Причини
Атрибут	Неприпустимі значення	дата народження = 30.13.70	Значення за межами діапазону
Атрибут	Орфографічні помилки	місто = Харьків	Орфографічні помилки
Атрибут	Об'єднані значення	ім'я = Іван 12.07.70 Київ	Декілька значень в атрибуті
Запис	Порушення залежності атрибутів	вік = 22 дата народження = 12.02.70	Вік = (поточна дата – дата народження)
Тип запису	Дублікати записів	співробітник 1 = (ім'я = Іван, SSN = 123) співробітник 2 = (ім'я = Іван, SSN = 123)	Дублікати записів
Тип запису	Протиречні записи	співробітник 1 = (ім'я = Іван, SSN = 123) співробітник 1 = (ім'я = Іван, SSN = 321)	Один і той ж співробітник з різними SSN
Тип запису	Порушення унікальності	співробітник 1 = (ім'я = Іван, SSN = 123) співробітник 2 = (ім'я = Гліб, SSN = 123)	неунікальний SNN
Джерело	Невірні посилання	співробітник = (ім'я = Іван, відділ = 789)	Відділ 789 існує, але вказаний невірно

4.3 Очищення даних

Очищення даних (data cleaning, data cleansing, scrubbing) передбачає виявлення та видалення помилок і невідповідностей у даних з метою поліпшення їх якості.

Для вирішення проблем очищення даних в одному джерелі, у тому числі перед його інтеграцією з іншими джерелами даних, реалізують такі етапи.

1. Вилучення значень з атрибутів довільної форми (розбиття атрибутів, Extracting values from free-form attributes (attribute split)) передбачає виділення значущої частини ряд-

кового значення атрибуту, яке складається з декількох слів поспіль (наприклад, адреса або повне ім'я людини).

2. Валідація та корекція (Validation and correction). Даний етап передбачає пошук помилок, введення даних та їх виправлення найбільш автоматичним способом (наприклад, автоматична перевірка правопису або відома залежність атрибутів «дата народження» – «вік»).

3. Стандартизація (Standardization) передбачає приведення всіх даних до єдиного універсального формату (формат написання дати та часу, розмір регістру написання строкових значень, уніфікація абревіатур тощо).

4. Усунення пропущених даних. Можливі варіанти вирішення проблеми пропущених даних:

– виключити об'єкти із пропущеними значеннями з обробки;

– розрахувати нові значення для пропущених даних;

– ігнорувати пропущені значення в процесі аналізу;

– замінити пропущені значення на можливі.

Розглянемо ці методи детальніше:

А. Видалення пропущених значень. Найпростіша стратегія обробки пропущених даних – видалення записів, що містять пропущене значення (видалення відповідних стовпців (ознак) або рядків (об'єктів) з набору даних).

Переваги та недоліки:

+ добре впливає на точність моделі;

+ видалення конкретного рядка або стовпця без конкретної інформації краще, оскільки він не має великої ваги;

– втрата інформації та даних, цінних для класифікатора;

– є ризик видалити занадто багато зразків, що зробить якісний аналіз неможливим;

– працює погано, якщо відсоток відсутніх значень високий.

Видалення рядків з пропущеними значеннями може позбавити модель важливих даних. Альтернативою є відновлення пропущених значень шляхом заповнення їх деякими значеннями.

Б. Заповнення пропущених значень. Використовують такі види інтерполяції для оцінки пропущених даних:

– значення з іншого випадково вибраного запису;

– середнє значення або медіана всього стовпця ознак (для числових даних);

– постійне осмислене значення, відмінне від усіх інших значень, наприклад, 0 або найчастіше значення в стовпці (може використовуватися для категоріальних ознак);

– значення, що оцінюється прогнозною моделлю.

Значення, застосоване для заповнення у навчальному наборі даних, має застосовуватись і на нових даних у майбутньому.

Переваги:

+ гарний метод, якщо кількість даних невелика;

+ дозволяє попередити видалення значущих об'єктів або ознак.

В. Прогнозування пропущених значень. Використовуючи атрибути, у яких немає відсутніх значень, можна заповнити пропуски у стовпці того атрибуту, де вони є. Метод заснований на припущенні про те, що близьким об'єктам у просторі ознак відповідають схожі мітки.

Для прогнозування пропущених значень може використовуватися модель регресії або класифікації залежно від характеру даних, що мають пропущене значення (категоріальні або неперервні ознаки).

Переваги та недоліки:

+ дає кращий результат, ніж попередні методи;

+ враховує коваріацію між стовпцем з пропущеним значенням та іншими стовпцями;

– дає тільки наближення до реальних значень.

Перед корегуванням пропущених даних варто оцінити важливість атрибутів, які відсутні в даних, і причини їх відсутності. Більшість методів інтелектуального аналізу роблять припущення про відсутність значення через невідомість. Однак, можливо, є причини, за якими значення атрибута відсутнє, або було прийнято рішення з певних причин не проводити заміри конкретного атрибута, і цей факт може мати значення для отримання додаткової інформації про екземпляр.

5. Усунення дублювання записів. Однією з проблем при інтеграції різних джерел даних є дублювання записів. Усунення дублікатів виконується після переважної більшості чисток і передбачає ідентифікацію подібних у певному сенсі записів та їх злиття з об'єднанням атрибутів. Вирішення цього завдання за наявності у дублюючих записів первинного ключа просте. Якщо ж однозначно ідентифікуючої ознаки немає, завдання значно ускладнюється, вимагаючи застосування спеціальних підходів порівняння (близькості у певному сенсі) між собою записів.

6. Збалансованість даних. Інколи певне значення атрибуту класу у наборі даних зустрічається набагато частіше, ніж інші. За таких умов точність класифікації буде високою, якщо більшість об'єктів (або всі) буде віднесено до найпоширенішого класу, навіть, помилково. Однак, якщо представники меншого класу є більш значущими (наприклад, діагностування хворих на туберкульоз або СНІД у протилежність до здорових людей), то висока якість класифікації до найпоширенішого класу (і здорових і хворих віднесено до здорових) може мати відчутно шкідливі наслідки. Отже, на етапі підготовки даних до інтелектуального аналізу необхідно перевіряти їх збалансованість.

Очевидно, що результати інтелектуального аналізу на основі брудних даних не можуть вважатися надійними, але завжди необхідно зважено підходити до вибору між ігноруванням брудних даних і вартістю й/або часом, необхідним для їхнього очищення.

4.4. Оптимізація простору ознак

4.4.1 Масштабування ознак

Масштабування ознак (*feature scaling*) є ключовим етапом у передобробці даних. Більшість алгоритмів аналізу даних веде себе набагато краще, якщо ознаки подаються у однаковій

шкалі і мають однакові масштаби.

Найпоширенішим способом масштабування даних є їх нормування (стандартизація). Нормування дозволяє центрувати стовпець певної ознаки до нульового середнього значення з одиничним середньоквадратичним відхиленням за формулою:

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j},$$

де $x_j^{(i)}$ та $\tilde{x}_j^{(i)}$ – це початкове та стандартизоване значення j-ї ознаки i-го зразка;

μ_j та σ_j – вибіркове середнє та стандартне відхилення поточного стовпця ознак.

4.4.2 Відбір ознак

Сучасні масиви даних, до яких застосовуються методи аналізу даних, можуть характеризуватись великою кількістю ознак, що формують ознаковий простір великої розмірності. Тому актуальним є завдання зниження розмірності такого простору до розмірності, що дозволяє без зайвих труднощів здійснювати обробку даних та/або їх візуалізацію. Вирішення цієї задачі називають оптимізацією ознакового простору або пошуком значущих ознак (англ. Feature Selection, іноді – Feature Engineering).

Існують наступні підходи до зниження розмірності вихідного ознакового простору:

– трансформація ознакового простору, зокрема, методом головних компонентів (МГК, англ. Principal Component Analysis). Недоліком таких методів є спотворення початкового ознакового простору під час його трансформації, що може призвести до зниження якості результатів аналізу даних;

– вибір найбільш інформативних, корисних ознак та виключення з розгляду неінформативних ознак без трансформації вихідного простору.

4.5 Попередня обробка даних в R

4.5.1 Обробка пропущених значень

У R є кілька функцій, призначених для виявлення пропущених значень.

Функція `is.na()` дозволяє перевірити дані про наявність пропущених значень:

```
y <- c(1, 2, 3, NA)
is.na(y)
c(FALSE, FALSE, FALSE, TRUE).
```

Вона повертає об'єкт такої ж розмірності, що й аналізований, де `TRUE` відповідають пропущеним значенням і `FALSE` – якщо значення не було пропущено.

Оскільки `TRUE` маркується 1, а `FALSE` – 0, то застосування функції `sum()` до результату перевірки дасть кількість пропусків:

```
sum(is.na(y))
[1] 1
```

Якщо досліджується датафрейм, то застосування функції `colSums()` до результатів `is.na()` дає кількість пропусків у кожному стовпці датафрейму:

```
colSums(is.na(df))
```

Наявність пропущених значень призводить до того, що числові функції на таких наборах даних не можуть видавати коректний результат. У більшості числових функцій є параметр `na.rm=TRUE`, який ігнорує пропущені значення у процесі обчислень:

```
sum(y)
[1] NA
sum(y, na.rm=TRUE)
[1] 6
```

Для видалення тих рядків датафрейму, які містять пропущені значення (хоча б в одній з ознак), використовується функція `na.omit()`:

```
na.omit(df)
```

Якщо ж пропущені значення розсіяні по всій таблиці даних, видалення рядків знищить помітну частину даних. За таких умов краще заповнювати пропуски певними значеннями, наприклад, середнім значенням або медіаною за відповідною ознакою. Для цього зручно використовувати пакет `Hmisc` (потребує попереднього одноразового інсталювання та підключення у кожній сесії):

```
    library(Hmisc)
    # заміна пропусків у стовпці feature1 датафрейму
df середнім значенням цього стовпця
    df$feature1<-impute(df$feature1, mean)
    # заміна медіаною
    df$feature1<-impute(df$feature1, median)
    # заміна заданим значенням
    df$feature1<-impute(df$feature1, 50)
```

4.5.2 Стандартизація даних

Функція `scale()` стандартизує заданий стовпець матриці або таблиці даних так, щоб його середнє арифметичне дорівнювало нулю, а стандартне відхилення – одиниці.

```
    newdf <- scale(df$feature1)
```

Для перетворення стовпця так, щоб його середнє арифметичне та стандартне відхилення набули заданих значень, використовується такий код:

```
    newdf <- scale(newdf)*SD + M
```

де M та SD – потрібні значення середнього арифметичного та стандартного відхилення відповідно.

Застосування `scale()` до стовпців із нечисловими даними викликає повідомлення про помилку. Щоб стандартизувати окремий стовпець, а не всю матрицю або таблицю даних цілком, потрібно використовувати наступний код:

```
    newdf <- transform(df$feature1=scale(df$feature1)*SD + M)
```