Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Автоматизований аналіз якості даних
на основі штучного інтелекту
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-1
Володимир КІРЮШИН
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник доц. Олександра ВІТЬКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кірюшину Володимиру Артемовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизований аналіз якості даних на основі штучного інтелекту

затверджена наказом університету від 6 квітня 20 26 р. № 269Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 20 травня 20 26 р.

3. Вихідні дані до роботи наукові статті та публікації з області автоматизованого дослідницького аналізу даних та інтелектуальної оцінки якості даних; дослідження, присвячені алгоритмам виявлення багатовимірних аномалій (Isolation Forest), методам ідентифікації шуму в мітках та застосуванню автоенкодерів для моделювання розподілів; роботи, що описують підходи до семантичної інтерпретації результатів аналізу за допомогою локальних великих мовних моделей та методів їх квантування; документація бібліотек обробки даних (Pandas, Polars) та машинного навчання (Scikit-learn); відкриті набори даних «Titanic» та «Zomato» як ресурси для проведення оцінки ефективності розробленої системи.

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі та постановка задачі

2) Теоретичні дослідження методів аналізу якості даних

3) Розробка та практична реалізація інтелектуальної системи аналізу якості табличних даних

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	06.04.2026	виконано
2	Визначення проблеми	09.04.2026	виконано
3	Аналіз теоретичних матеріалів	12.04.2026	виконано
4	Проведення теоретичних досліджень	18.04.2026	виконано
5	Реалізація програмного застосунку	29.04.2026	виконано
6	Проведення тестування застосунку	02.05.2026	виконано
7	Оформлення пояснювальної записки	04.05.2026	виконано
8	Захист	20.05.2026	виконано

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Олександра ВІТЬКО
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 134 с., 21 рис., 3 табл., 2 дод., 41 джерело.

АВТОМАТИЗОВАНИЙ ДОСЛІДНИЦЬКИЙ АНАЛІЗ, ВИЯВЛЕННЯ АНОМАЛІЙ, ОЦІНЮВАННЯ ЯКОСТІ ДАНИХ, ТАБЛИЧНІ ДАНІ, ШТУЧНИЙ ІНТЕЛЕКТ, AEDA, AUTOENCODER, CONFIDENT LEARNING, EDA, ISOLATION FOREST, LLM, ON-DEVICE AI.

Об’єктом дослідження є процес автоматизованого дослідницького аналізу та оцінювання якості табличних наборів даних.

Предметом дослідження є алгоритми машинного та глибокого навчання, статистичні методи виявлення проблем якості даних (аномалій, дублікатів, відсутніх значень), а також застосування локальних великих мовних моделей для генерації семантичних пояснень.

Метою роботи є розробка програмної системи для автоматизованого комплексного аналізу якості даних, яка інтегрує передові методи штучного інтелекту для діагностики табличних наборів даних та формування аналітичних звітів.

Методами дослідження є описова статистика, методи ансамблевого та глибокого навчання, алгоритми на основі локально-чутливого хешування, а також методи квантування великих мовних моделей.

Результати роботи можуть бути використані для автоматизації процесів підготовки даних у машинному навчанні та підвищення продуктивності роботи Data Science фахівців, зокрема для комплексного автоматизованого аналізу якості табличних наборів даних. Сфера застосування включає конвеєри машинного навчання, корпоративні системи управління даними та інтелектуальні платформи аналітики.

ABSTRACT

Master’s thesis contains: 134 pp., 21 fig., 3 tabl., 2 ann., 41 references.

AEDA, ANOMALY DETECTION, ARTIFICIAL INTELLIGENCE, AUTOENCODER, AUTOMATED EXPLORATORY ANALYSIS, CONFIDENT LEARNING, DATA QUALITY ASSESSMENT, EDA, ISOLATION FOREST, LLM, ON-DEVICE AI, TABULAR DATA.

The object of this study is the process of automated research analysis and evaluation of the quality of tabular datasets.

The subject of this study includes machine learning and deep learning algorithms, statistical methods for detecting data quality issues (anomalies, duplicates, missing values), as well as the application of local large language models for generating semantic explanations.

The aim of this work is to develop a software system for automated comprehensive data quality analysis that integrates state-of-the-art artificial intelligence methods for diagnosing tabular datasets and generating analytical reports.

The research methods include descriptive statistics, ensemble and deep learning methods, algorithms based on locality-sensitive hashing, as well as methods for quantizing large language models.

The results of this work can be used to automate data preparation processes in machine learning and to improve the productivity of data science specialists, particularly for comprehensive automated analysis of the quality of tabular datasets. The scope of application includes machine learning pipelines, enterprise data management systems, and intelligent analytics platforms.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 10
Вступ 11
1 Аналіз предметної галузі та постановка задачі 13
1.1 Роль якості даних у сучасних інформаційних системах 13
1.1.1 Значення якості даних у процесах аналізу даних та машинного навчання 13
1.1.2 Основні характеристики якості даних у інформаційних системах 14
1.1.3 Вплив якості даних на результати аналітичних моделей 15
1.2 Проблеми підготовки та аналізу даних у задачах Data Science 16
1.2.1 Етапи підготовки даних у типовому ML-pipeline 16
1.2.2 Типові проблеми реальних наборів даних 18
1.2.3 Складність ручного дослідницького аналізу даних 20
1.3 Типи проблем якості даних у табличних наборах даних 21
1.3.1 Пропущені значення та їх вплив на аналіз даних 22
1.3.2 Дублікати та нечіткі дублікати записів 23
1.3.3 Аномалії та викиди у багатовимірних даних 24
1.3.4 Невідповідність значень та порушення структурної цілісності даних 25
1.3.5 Шум у мітках у задачах навчання з учителем 25
1.4 Сучасні інструменти автоматизованого дослідницького аналізу даних 26
1.4.1 Огляд існуючих систем автоматизованого аналізу даних 27
1.4.2 Функціональні можливості сучасних інструментів профілювання даних 28
1.4.3 Порівняльний аналіз можливостей існуючих рішень 29
1.5 Обмеження існуючих підходів до аналізу якості даних 31

1.5.1 Недостатня глибина статистичного та алгоритмічного аналізу 31
1.5.2 Відсутність інтеграції методів машинного навчання 32
1.5.3 Обмежені можливості інтерпретації результатів аналізу 33
1.5.4 Проблеми масштабованості та автоматизації 34
1.6 Формування вимог до системи автоматизованого аналізу якості даних 35
1.6.1 Основні сценарії використання системи 35
1.6.2 Функціональні вимоги до системи аналізу даних 37
1.6.3 Нефункціональні вимоги до програмної системи 38
1.7 Постановка задачі 39
2 Теоретичні дослідження методів аналізу якості даних 40
2.1 Теоретичні основи дослідницького аналізу даних 40
2.1.1 Поняття та цілі EDA 41
2.1.2 Основні етапи дослідницького аналізу даних 43
2.1.3 Роль автоматизації у сучасних системах аналізу даних 45
2.2 Методи виявлення проблем якості даних 46
2.2.1 Методи виявлення пропущених значень та неповних даних 46
2.2.2 Методи виявлення дублікатів та схожих записів 48
2.3 Методи виявлення аномалій у даних 51
2.3.1 Статистичні підходи до виявлення аномалій 51
2.3.2 Алгоритми машинного навчання для пошуку аномалій 52
2.3.3 Алгоритм Isolation Forest у задачах виявлення викидів 53
2.4 Методи виявлення помилок цільової функції 55
2.4.1 Проблема шуму у мітках у задачах навчання з учителем 55
2.4.2 Підхід Confident Learning для виявлення помилкових міток 56
2.4.3 Використання ансамблів моделей для оцінювання достовірності міток 58
2.5 Методи аналізу структурної узгодженості даних 59
2.5.1 Поняття функціональних залежностей у табличних даних 59

2.5.2 Методи виявлення порушень цілісності даних 60
2.6 Методи моделювання розподілів даних 61
2.6.1 Підходи до аналізу багатовимірних розподілів 61
2.6.2 Використання автоенкодерів для виявлення аномалій 62
2.7 Оцінювання загальної якості набору даних 64
2.7.1 Моделі оцінювання якості даних 64
2.7.2 Стандарт ISO 25012 для оцінювання якості даних 65
2.7.3 Формування інтегральної оцінки якості набору даних 66
2.8 Використання великих мовних моделей для інтерпретації результатів аналізу 67
2.8.1 Принципи роботи великих мовних моделей 67
2.8.2 Можливості використання LLM у задачах аналізу даних 69
2.8.3 Формування текстових пояснень результатів аналізу 70
2.9 Концепція рішення для автоматизованого аналізу якості даних 71
3 Розробка та практична реалізація інтелектуальної системи аналізу якості табличних даних 74
3.1 Організація експериментального дослідження 74
3.1.1 Мета та задачі практичних досліджень 74
3.1.2 Характеристики використаних наборів даних 75
3.1.3 Сценарії використання системи 76
3.2 Архітектура застосунку та pipeline обробки даних 78
3.2.1 Загальна архітектура системи 78
3.2.2 Організація виконання аналізу 80
3.3 Дослідження шару обробки даних 85
3.3.1 Реалізація dual-engine підходу 85
3.3.2 Порівняльний аналіз продуктивності рушіїв 86
3.3.3 Уніфікація доступу до даних 88
3.4 Базові компоненти аналізу 90
3.4.1 Компонент аналізу структури даних 90
3.4.2 Компонент аналізу пропущених значень 91

3.4.3 Компонент виявлення точних дублікатів 92
3.5 Компоненти виявлення аномалій та якості даних 93
3.5.1 Виявлення подібних записів 93
3.5.2 Виявлення аномалій у числових даних 95
3.5.3 Виявлення аномалій у категоріальних даних 96
3.5.4 Виявлення шуму у мітках 96
3.5.5 Аналіз логічної узгодженості даних 98
3.5.6 Аналіз розподілу даних 99
3.5.7 Комлексна оцінка якості даних та формування звіту 99
3.6 Інтеграція великих мовних моделей 101
3.6.1 Архітектура LLM-інтеграції 101
3.6.2 Ролі LLM у системі 102
3.6.3 Prompt Engineering 104
3.6.4 Детерміністична стратегія у виявленні шуму міток 105
3.7 Генерація звітів та інтерфейс користувача 106
3.7.1 Створення фінальних звітів 106
3.7.2 Десктопний користувацький інтерфейс 110
3.7.3 Розповсюдження системи 112
3.8 Дослідження продуктивності та масштабованості 113
3.9 Узагальнення результатів практичних досліджень 117
3.9.1 Порівняння з існуючими інструментами автоматизованого EDA 117
3.9.2 Переваги запропонованого підходу 118
3.9.3 Обмеження розробленої системи 119
3.9.4 Напрями подальшого розвитку системи 121
Висновки 123
Перелік джерел посилання 125
Додаток А Реалізації просунутих компонентів аналізу 130
Додаток Б Відомість кваліфікаційної роботи 134

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AEDA – Automated Exploratory Data Analysis – автоматизований дослідницький аналіз даних, назва програмного застосунку;

EDA – Exploratory Data Analysis – дослідницький аналіз даних;

LLM – Large Language Model – велика мовна модель;

MAR – Missing At Random – випадкова відсутність;

MCAR – Missing Completely At Random – повністю випадкова відсутність;

MNAR – Missing Not At Random – невипадкова відсутність.

ВСТУП

Стрімкий розвиток інформаційних технологій та систем, що базуються на даних, значно підвищив важливість аналізу даних. Машинне навчання, штучний інтелект та аналітичні системи значною мірою залежать від якості даних, що використовуються під час навчання та оцінювання моделей [1]. Однак на практиці реальні набори даних часто містять пропущені значення, невідповідності, дублікати, аномалії та інші проблеми з якістю даних, які можуть негативно вплинути на надійність аналітичних результатів.

Тому значна частина роботи в галузі науки про дані присвячена дослідницькому аналізу даних та підготовці наборів даних. Ці процеси включають виявлення аномалій, оцінку статистичних властивостей даних та забезпечення цілісності й узгодженості структур наборів даних. Незважаючи на важливість цього етапу, багато перевірок якості даних досі виконуються вручну або за допомогою інструментів, що надають лише описову статистику та базові візуалізації, що обмежує глибину аналізу та знижує відтворюваність.

У цьому контексті важливим завданням є розробка автоматизованих систем для комплексного аналізу якості даних. Такі системи повинні поєднувати статистичні методи, алгоритми машинного навчання та сучасні обчислювальні інструменти для виявлення складних закономірностей, аномалій та невідповідностей у наборах даних. Крім того, результати такого аналізу повинні бути представлені у чіткій та зрозумілій формі для забезпечення ефективного прийняття рішень під час фази підготовки даних.

Ця робота присвячена розробці програмної системи для автоматизованого дослідницького аналізу даних та оцінки якості наборів даних під назвою AEDA. Запропонована система реалізує модульну архітектуру, що поєднує методи статистичного аналізу, методи машинного навчання для виявлення аномалій та механізми автоматизованої звітності.

Система виконує низку аналітичних процедур, зокрема виявлення дублікатів, ідентифікацію виняткових значень, аналіз відсутніх значень, оцінку реляційної узгодженості та моделювання розподілів даних.

Важливою особливістю розробленої системи є формування детальних аналітичних звітів, які узагальнюють виявлені проблеми та надають зрозумілі пояснення результатів аналізу. Інтеграція сучасних технологій штучного інтелекту дозволяє системі представляти складні аналітичні висновки у структурованій та зрозумілій для користувачів формі.

Розробка таких автоматизованих аналітичних інструментів сприяє підвищенню ефективності дослідницького аналізу даних та надійності процесів машинного навчання, забезпечуючи комплексну оцінку якості наборів даних ще до етапу розробки моделі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Роль якості даних у сучасних інформаційних системах

1.1.1 Значення якості даних у процесах аналізу даних та машинного навчання

У сучасних інформаційних системах дані стали одним із найцінніших ресурсів для прийняття рішень, прогнозування та автоматизації. Аналітичні платформи, моделі машинного навчання та системи штучного інтелекту значною мірою залежать від наявності надійних і узгоджених наборів даних. Ефективність цих систем безпосередньо залежить від якості даних, що використовуються на перших етапах життєвого циклу моделей – навчання, валідації та інференції.

Високоякісні дані гарантують, що аналітичні моделі будуть здатні виявляти значущі закономірності та взаємозв’язки в наборі даних. Коли набори даних є повними, узгодженими та репрезентативними щодо базових процесів, алгоритми машинного навчання можуть навчатися точним прогнозним моделям і надавати надійні висновки. І навпаки, низька якість даних може призвести до упереджених моделей, неправильних прогнозів та ненадійних аналітичних результатів. Ця проблема стає особливо критичною у сферах, де рішення, засновані на аналізі даних, мають значні практичні наслідки. Наприклад, медицина чи фінанси.

У контексті пайплайнів машинного навчання якість даних впливає на кілька етапів життєвого циклу розробки моделі. Під час фази дослідницького аналізу даних аналітики оцінюють властивості набору даних, виявляють невідповідності та визначають, чи підходять наявні дані для моделювання. Якщо на цьому етапі якісь значні проблеми з якістю даних залишаються невиявленими, вони можуть поширитися на весь

подальший робочий процес і негативно вплинути на кінцеву продуктивність системи.

Отже, забезпечення високої якості даних є основною передумовою для розробки надійних аналітичних систем. Для забезпечення надійних процесів прийняття рішень на основі даних необхідні ефективні механізми виявлення невідповідностей, аномалій та структурних проблем у наборах даних.

1.1.2 Основні характеристики якості даних у інформаційних системах

Якість даних – це комплексне поняття, яке описує ступінь придатності даних для їхнього передбаченого використання в конкретному контексті. В інформаційних системах та аналітичних середовищах для оцінки якості наборів даних зазвичай використовують кілька характеристик.

Однією з основних характеристик є повнота, яка означає наявність усіх необхідних значень даних у наборі. Відсутні значення або неповні записи можуть суттєво знизити ефективність аналітичних моделей та ускладнити інтерпретацію результатів. Іншою важливою характеристикою є узгодженість, яка описує відсутність суперечностей у наборі даних. Значення даних повинні відповідати заздалегідь визначеним правилам, обмеженням та взаємозв’язкам між атрибутами, відповідно до предметної галузі.

Точність – це ще один ключовий аспект якості даних. Точні дані правильно відображають реальні об’єкти або процеси, які вони представляють. Неточні або помилкові дані можуть призвести до оманливих аналітичних висновків та знизити надійність моделей машинного навчання. Окремо варто виділити унікальність, що є важливою властивістю, яка гарантує відсутність дубльованих записів, що представляють один і той самий об’єкт.

Інші характеристики включають валідність, яка вказує на те, що значення даних відповідають очікуваним форматам або доменам, та цілісність, яка описує збереження логічних взаємозв'язків між різними атрибутами набору даних. У складних наборах даних порушення цих характеристик можуть проявлятися у різних формах, таких як невідповідність значень атрибутів, структурні аномалії або суперечливі записи.

Наявність недоліків у будь-якому з цих аспектів може істотно вплинути на результати аналізу даних та навчання моделей. Отже, для забезпечення надійності аналітичних процесів необхідна комплексна оцінка низки характеристик якості даних.

1.1.3 Вплив якості даних на результати аналітичних моделей

Проблеми з якістю даних можуть істотно вплинути на продуктивність та надійність систем аналітики та машинного навчання. Оскільки більшість сучасних алгоритмів базується на статистичних закономірностях у наборах даних, будь-які спотворення в даних можуть призвести до некоректної поведінки моделі або оманливих прогнозів.

Одним із поширених наслідків низької якості даних є виникнення систематичної похибки в аналітичних моделях. Наприклад, неповні або незбалансовані набори даних можуть призвести до того, що моделі навчатимуться закономірностям, які неточно відображають базовий розподіл даних. Це може призвести до систематичних помилок при застосуванні моделі до нових даних, що модель не бачила під час тренування.

Іншою важливою проблемою є наявність виняткових значень та аномальних спостережень. Хоча деякі аномалії можуть відображати значущі рідкісні події, багато виняткових значень виникають через помилки вимірювання, помилки введення даних або невідповідності в процесах

збору даних. Якщо такі значення не виявляються та не обробляються належним чином, вони можуть суттєво спотворити статистичні оцінки та негативно вплинути на навчання моделі.

Дублікати записів є ще однією формою проблеми якості даних, яка може вплинути на результати аналізу. Коли в наборі даних з'являються дублікати, вони можуть штучно підвищити важливість певних даних, що може призвести до упередженого формування параметрів та неправильної поведінки моделі. При цьому варто зазначити, що дублікати можуть виникати як природно, так і як наслідок некоректного збору даних. Саме тому всі ці та інші проблеми не можна повністю віддавати на автоматичне опрацювання, хоча, звичайно, залежно від предметної галузі та доменних знань.

Шум у значеннях цільової функції також є серйозною проблемою при виконанні завдань контрольованого навчання. Неправильні або суперечливі мітки можуть призвести до того, що моделі машинного навчання засвоять неточні взаємозв'язки між ознаками та цільовими змінними. Це часто призводить до зниження точності моделі та погіршення її здатності до узагальнення.

З цих причин систематичне виявлення та аналіз проблем якості даних є важливими етапами у підготовці наборів даних для аналітичних завдань. Автоматизовані методи виявлення аномалій, дублікатів, структурних невідповідностей та помилок у мітках можуть істотно підвищити надійність та ефективність робочих процесів аналізу даних.

1.2 Проблеми підготовки та аналізу даних у задачах Data Science

1.2.1 Етапи підготовки даних у типовому ML-pipeline

Типовий ML-pipeline являє собою структуровану послідовність процесів, які перетворюють необроблені дані у форму, придатну для

аналітичного моделювання [2]. На початковому етапі дані отримують із різних джерел, серед яких можуть бути бази даних, файли, API або стрімінгові системи. Цей етап тісно пов’язаний із завантаженням та інтеграцією даних, під час яких неоднорідні дані з різних джерел необхідно об’єднати в єдиний набір даних.

Після збору даних важливим кроком є профілювання даних (рисунок 1.1), яке передбачає аналіз структури, розподілу та статистичних властивостей набору даних. Цей етап дозволяє виявити аномалії, невідповідності та потенційні проблеми з якістю, які можуть вплинути на подальший аналіз. Після профілювання виконується очищення та перетворення даних. Ці процеси включають обробку пропущених значень, виправлення помилок, нормалізацію форматів та перетворення даних у відповідні представлення для завдань моделювання.

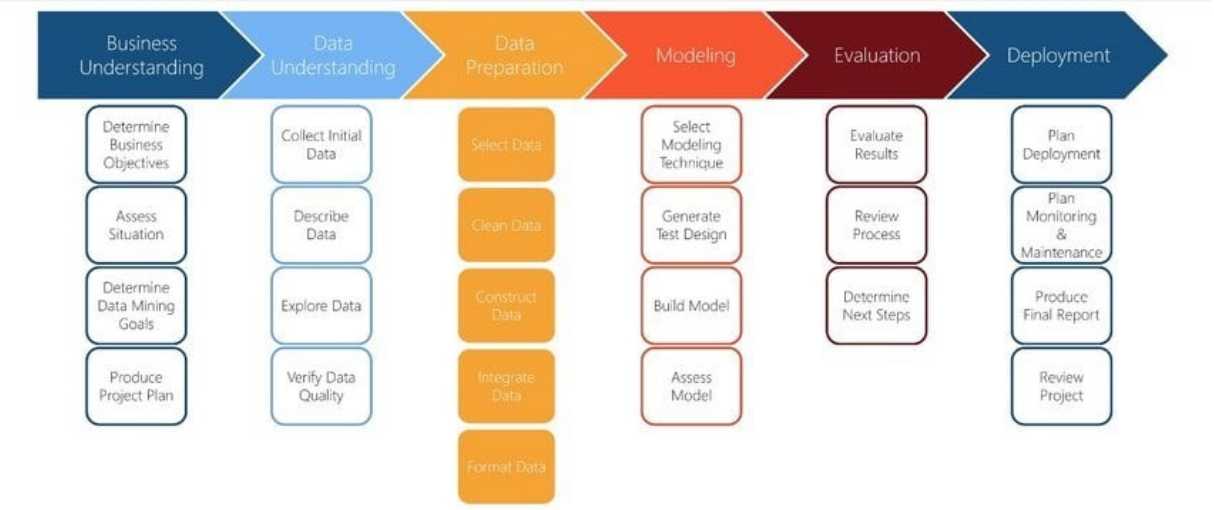


Рисунок 1.1 – Типова діаграма потоку даних CRISP-DM

Іншим важливим етапом є feature engineering, під час якого необроблені атрибути перетворюються на змістовні ознаки, що покращують ефективність моделей машинного навчання. Це може включати масштабування, кодування категоріальних змінних, агрегування даних або генерування нових ознак на основі знань у відповідній галузі. Згодом

підготовлений набір даних поділяється на підмножини для навчання, валідації та тестування, що забезпечує можливість належної оцінки моделі.

Дослідницький аналіз даних інтегрований у весь робочий процес і є важливим етапом для розуміння характеристик даних, виявлення прихованих закономірностей та прийняття рішень щодо попередньої обробки. Він надає інформацію, яка впливає як на розробку ознак, так і на вибір моделі.

Загалом, етап підготовки даних є ітеративним і значною мірою залежить від конкретних характеристик набору даних та аналітичного завдання. Відсутність універсального стандартизованого робочого процесу означає, що фахівці повинні динамічно адаптувати ці етапи, часто покладаючись на досвід та експерименти для досягнення оптимальних результатів.

1.2.2 Типові проблеми реальних наборів даних

У реальності, набори даних рідко бувають «чистими», структурованими або відразу придатними для аналітичної обробки. Натомість вони часто містять цілу низку проблем із якістю, що ускладнює їх використання в задачах машинного навчання та аналізу даних. Однією з найпоширеніших проблем є наявність пропущених або неповних даних, що може бути спричинено помилками під час збору даних, збоями в роботі систем або невідповідностями між різними джерелами даних. Такі проблеми зменшують обсяг корисної інформації та можуть спричинити упередженість аналітичних результатів.

Ще однією значною проблемою є невідповідність даних. Коли набори даних збираються з різних джерел, часто виникають розбіжності у форматах, правилах іменування та представленні даних. Ці невідповідності ускладнюють інтеграцію даних і можуть призвести до їх неправильного тлумачення, якщо не вирішити це належним чином. Крім того, набори даних

часто містять недійсні або помилкові значення, зокрема випадкові користувацькі помилки, неправильні вимірювання або логічно суперечливі записи.

Дублювання даних також є поширеною проблемою, особливо у масштабних системах, де записи можуть повторюватися у різних джерелах даних. Дубльовані записи можуть спотворювати статистичні розподіли та призводити до упередження у моделях. Крім того, набори даних можуть містити шум у вигляді випадкових помилок або нерелевантної інформації, що зменшує відношення сигнал/шум та негативно впливає на ефективність моделі.

Важливим фактором якості даних є їх неоднорідність. Дані можуть відрізнятися не лише за форматом, а й за структурою: від структурованих табличних даних до напівструктурованих та неструктурованих форматів. Ця різноманітність ускладнює попередню обробку та вимагає застосування гнучких і адаптивних методів для перетворення та аналізу даних (рисунок 1.2).

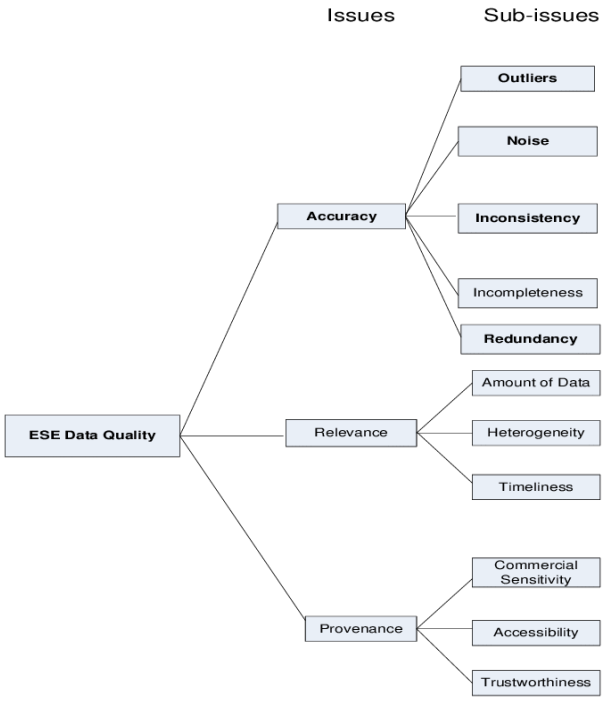


Рисунок 1.2 – Типові проблеми якості даних

Всі вищезазначені виклики свідчать про те, що набори даних реального світу за своєю суттю є недосконалими, а їх підготовка вимагає ретельного аналізу та численних етапів обробки. Без системних підходів до виявлення та вирішення таких проблем надійність подальших аналітичних процесів залишається обмеженою.

1.2.3 Складність ручного дослідницького аналізу даних

Дослідницький аналіз даних є фундаментальним етапом у галузі науки про дані, спрямованим на розуміння властивостей наборів даних, виявлення аномалій та пошук прихованих закономірностей. Саме він відповідає за вирішення деяких із проблем, перелічених раніше. Незважаючи на свою важливість, цей процес залишається надзвичайно трудомістким, особливо при виконанні вручну. Аналітики змушені послідовно перевіряти дані, обчислювати статистичні зведені показники, створювати візуалізації та інтерпретувати результати, часто повторюючи ці кроки кілька разів протягом усього процесу аналізу.

Емпіричні дослідження та галузеві звіти вказують на те, що значна частина загальних зусиль у проєктах з Data Science присвячена підготовці даних та дослідницькому аналізу. За загальними оцінками, до 60–80% загального часу в аналітичних робочих процесах витрачається на збір, очищення та попередню обробку даних, тоді як лише менша частина виділяється на розробку та оцінку моделей (рисунок 1.3) [3]. Цей дисбаланс підкреслює складність та затратність роботи з реальними даними. В свою чергу ефективність дослідницького аналізу часто залежить від досвіду та інтуїції аналітика, що призводить до варіативності результатів та знижує відтворюваність. До того ж ручний аналіз погано масштабується зі збільшенням обсягів даних, що ускладнює застосування послідовних аналітичних процедур до великих або складних наборів даних.

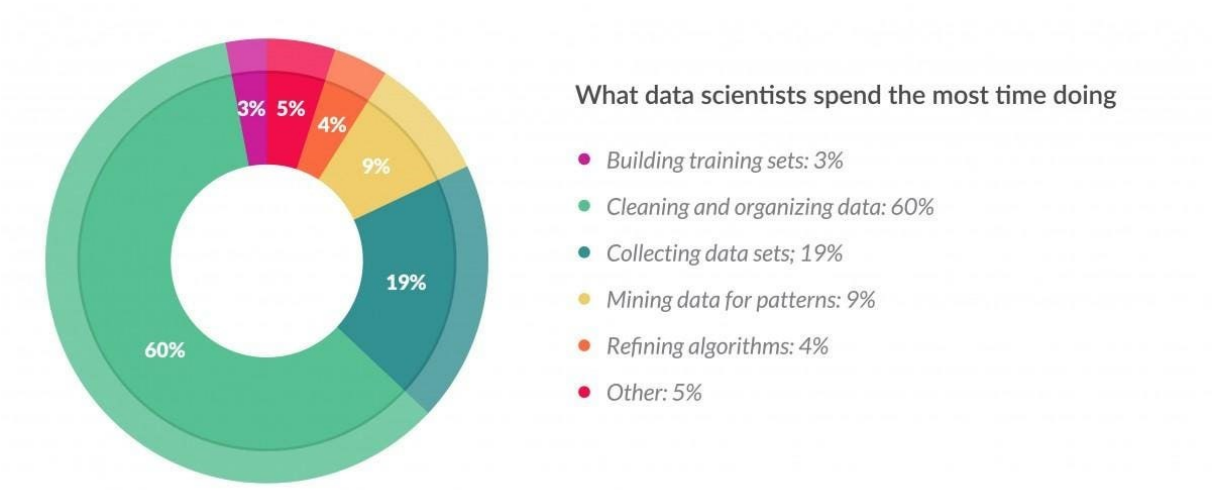


Рисунок 1.3 – Розподіл найбільш затратних завдань data-science фахівця із опитування журналу Forbes

Ще одним обмеженням ручного дослідницького аналізу є його обмежена здатність виявляти складні закономірності та взаємодії в даних. Хоча базові статистичні методи та візуалізації можуть виявити прості взаємозв'язки, більш складні аномалії або залежності можуть залишитися непоміченими без використання передових аналітичних методів.

Ці фактори вказують на необхідність створення автоматизованих систем, які можуть скоротити час, необхідний для дослідницького аналізу даних, забезпечити узгодженість результатів та дозволити виявляти складні проблеми з якістю даних. Розробка таких систем є важливим напрямком у підвищенні ефективності та надійності сучасних робочих процесів у галузі науки про дані.

1.3 Типи проблем якості даних у табличних наборах даних

Проблеми якості даних у табличних наборах даних становлять серйозну проблему в сучасних робочих процесах аналізу даних та машинного навчання. Ці проблеми виникають через недоліки у зборі, інтеграції, зберіганні та обробці даних. На практиці набори даних рідко

відповідають ідеальним умовам і часто містять численні супутні дефекти, зокрема пропущені значення, дублювання, невідповідності, аномалії та помилки маркування. Такі проблеми можуть суттєво знизити надійність аналітичних моделей та ускладнити процес отримання значущих висновків.

Виявлення та систематичний аналіз цих проблем є необхідними для забезпечення коректності систем, що базуються на даних. Кожен тип проблеми якості даних має свої особливості, причини та вплив на аналітичні процеси, що вимагає використання спеціалізованих методів для їх виявлення та усунення.

1.3.1 Пропущені значення та їх вплив на аналіз даних

Пропущені значення є однією з найпоширеніших і найважливіших проблем у табличних наборах даних. Вони виникають, коли дані за певними атрибутами не записані, втрачені або навмисно пропущені під час збору даних чи процесів їх інтеграції. Така неповнота може бути спричинена різними причинами, зокрема системними помилками, особливостями предметної галузі, помилками вимірювання або невідповідностями між різними джерелами даних.

Наявність таких аномалій у певних випадках може суттєво вплинути на статистичний аналіз та моделі машинного навчання. Зі статистичної точки зору, неповні дані зменшують ефективний розмір вибірки, що може знизити стійкість та надійність оцінок. У багатьох випадках відсутні дані призводять до упереджених результатів, особливо коли відсутність значень не є випадковою, а залежить від факторів, що лежать в основі набору даних.

У задачах машинного навчання відсутні значення ускладнюють навчання моделі, оскільки багато алгоритмів вимагають повних числових представлень. Як наслідок, необхідно застосовувати додаткові етапи попередньої обробки, такі як imputation, видалення таких записів або використання спеціалізованих моделей. Однак ці підходи створюють

додаткову невизначеність, оскільки, наприклад, вставлені значення можуть не точно відображати справжній розподіл даних.

Тому виявлення та правильна обробка відсутніх значень є важливими етапами попередньої обробки даних, що безпосередньо впливають на валідність та надійність подальших аналітичних процесів.

1.3.2 Дублікати та нечіткі дублікати записів

Дублікати записів виникають, коли один і той самий об’єкт реального світу представлений у наборі даних кілька разів. Ці дублікати можуть бути точними копіями або наближеними представленнями з незначними відмінностями, які часто називають «near-duplicates». Такі проблеми часто виникають під час інтеграції даних із різних джерел, міграції даних або при багаторазовому введенні даних.

Наявність дублікатів даних може суттєво спотворити результати аналізу. Зі статистичної точки зору, дублікати штучно збільшують частоту певних спостережень, що призводить до упереджених оцінок та асиметричних розподілів. У системах машинного навчання дубльовані записи можуть призвести до надмірного пристосування моделей до конкретних шаблонів, що знижує їхню здатність до узагальнення.

Нечіткі дублікати становлять ще складнішу проблему, оскільки їх не завжди легко ідентифікувати за допомогою простих порівнянь на основі рівності. Відмінності у форматуванні, типографічні розбіжності або часткові невідповідності даних можуть призвести до того, що кілька записів будуть представляти один і той самий об’єкт, не будучи розпізнаними як дублікати. Виявлення таких випадків часто вимагає використання методів наближеного зіставлення та алгоритмів пошуку схожості.

Вирішення проблеми дублювання є важливим для збереження цілісності даних, зменшення надмірності та забезпечення того, щоб

результати аналізу точно відображали явища предметної галузі, представлені в наборі даних.

1.3.3 Аномалії та викиди у багатовимірних даних

Аномалії та випадіння – це такі записи у даних, які суттєво відхиляються від загального розподілу набору даних. У багатовимірних даних такі відхилення можуть бути непомітними при перевірці окремих ознак, але стають очевидними під час аналізу комбінацій декількох атрибутів.

Випадіння можуть виникати через помилки вимірювання, пошкодження даних або їх неправильне введення, але в деяких випадках вони можуть відображати рідкісні, значущі події. Ця неоднозначність робить їх виявлення та інтерпретацію складними завданнями. З статистичної точки зору, випади можуть мати непропорційний вплив на узагальнювальні показники, такі як середнє значення та дисперсія, що призводить до спотвореного відображення набору даних.

У машинному навчанні наявність аномалій може негативно вплинути на навчання моделі, вводячи шум та оманливі закономірності. Моделі можуть або надмірно пристосовуватися до аномальних спостережень, або не бути спроможними належним чином узагальнювати через спотворені розподіли даних. Водночас у певних сферах застосування, таких як виявлення шахрайства або діагностика несправностей, аномалії становлять основний інтерес, що ще більше ускладнює розрізнення між шумом та цінною інформацією.

Виявлення винятків у таких даних вимагає більш досконалих методів порівняно з одновимірним аналізом, оскільки передбачає ідентифікацію спостережень, що є нетиповими стосовно спільного розподілу декількох змінних. Це робить виявлення аномалій обчислювально складним, але важливим компонентом аналізу якості даних.

1.3.4 Невідповідність значень та порушення структурної цілісності даних

Несумісність у табличних наборах даних виникає, коли дані порушують заздалегідь визначені правила, обмеження або логічні зв’язки між атрибутами. Ці проблеми можуть проявлятися у вигляді суперечливих значень, неправильних форматів або порушень обмежень, характерних для конкретної області.

Структурна цілісність стосується правильності взаємозв’язків у наборі даних, включаючи залежності між атрибутами та дотримання обмежень схеми. Порушення структурної цілісності можуть виникати, коли пов’язані поля містять несумісні значення або коли порушуються референтні взаємозв’язки між записами. Такі невідповідності часто виникають у системах, де дані збираються з неоднорідних джерел без механізмів перевірки. Відмінності у правилах форматування, відсутність стандартизованих схем та людські помилки під час введення даних сприяють виникненню цих проблем.

Наявність несумісних даних ускладнює аналітичну обробку та може призвести до неправильної інтерпретації. Наприклад, розбіжності в категоріальних значеннях або порушення логічних правил можуть внести неоднозначність у набір даних, що ускладнить ефективне застосування методів статистики або машинного навчання. Відповідно, забезпечення узгодженості та структурної цілісності є важливою вимогою для надійної роботи із даними.

1.3.5 Шум у мітках у задачах навчання з учителем

Під «шумом міток» розуміють наявність неправильних, суперечливих або неоднозначних міток у наборах даних, що використовуються для навчання з учителем [4]. Ця проблема особливо поширена у великих

наборах даних, отриманих за допомогою автоматизованих методів збору, таких як веб-скрейпінг або краудсорсингові платформи, де помилок важко уникнути.

Вплив шуму міток на моделі машинного навчання може бути значним. Неправильні мітки вводять оманливі навчальні шаблони, що змушують моделі вивчати неточні взаємозв'язки між вхідними ознаками та цільовими змінними. Це часто призводить до зниження точності моделі, поганої генералізації та підвищеної чутливості до перенавчання.

Шум міток може виникати з різних джерел, включаючи помилки анотації, неоднозначні зразки даних або невідповідності в інструкціях з мітки. У деяких випадках один і той самий екземпляр даних може бути по-різному позначений різними анотаторами, що призводить до невизначеності в істинній відповіді.

Вирішення проблеми шуму міток вимагає використання спеціалізованих технік, таких як надійні алгоритми навчання, методи виявлення шуму та процедури очищення даних. Виявлення та виправлення неправильно позначених екземплярів є важливим для покращення продуктивності моделі та забезпечення надійності систем навчання з наставником.

Підсумовуючи, шум міток є критичним аспектом якості даних, який безпосередньо впливає на ефективність моделей машинного навчання та підкреслює важливість точних та послідовних процесів анотації даних.

1.4 Сучасні інструменти автоматизованого дослідницького аналізу даних

Зростаючий попит на ефективну обробку та аналіз даних спричинив розробку численних інструментів, спрямованих на автоматизацію дослідницького аналізу даних (EDA). Ці інструменти призначені для спрощення початкових етапів роботи з наборами даних шляхом надання

статистичних зведень, візуалізацій та базових можливостей профілювання даних. Їхня головна мета – зменшити обсяг ручної роботи, необхідної для розуміння структури даних та виявлення потенційних проблем перед застосуванням методів аналізу або машинного навчання.

Незважаючи на наявність таких інструментів, рівень автоматизації та глибина аналізу значно різняться у різних рішеннях. Деякі інструменти зосереджуються переважно на створенні описових звітів, тоді як інші намагаються включити більш просунуті аналітичні функції. Проте комплексний та повністю автоматизований підхід до аналізу якості даних залишається складним завданням, особливо при роботі зі складними наборами даних з великою кількістю вимірів.

1.4.1 Огляд існуючих систем автоматизованого аналізу даних

Для підтримки автоматизованого дослідницького аналізу даних було розроблено широкий спектр програмних рішень. Серед найбільш поширених інструментів – бібліотеки та фреймворки, інтегровані в популярні екосистеми науки про дані. Наприклад, такі інструменти, як pandas-profiling (нині ydata-profiling) [5], Sweetviz [6] та Dataprep [7], забезпечують автоматичне формування звітів, що містять описову статистику, кореляції та візуалізації для табличних наборів даних.

Окрім автономних бібліотек, багато платформ даних та хмарних сервісів пропонують вбудовані можливості профілювання даних. Ці системи дозволяють користувачам аналізувати набори даних у інтегрованих середовищах, часто надаючи інтерактивні інформаційні панелі та інструменти візуалізації. Такі платформи зазвичай використовуються в корпоративних середовищах, де аналіз даних є частиною більш широкого робочого процесу з обробки даних.

Більш просунуті рішення розширюють базове профілювання, включаючи перевірку на відповідність правилам та перевірку якості даних.

Ці системи можуть дозволяти користувачам визначати обмеження, перевіряти дані на відповідність заздалегідь визначеним правилам та контролювати якість даних у часі. Однак ці підходи часто вимагають ручної конфігурації та знань у відповідній галузі, що обмежує їх доступність для менш досвідчених користувачів.

Загалом, існуючі системи надають цінну підтримку для початкового дослідження даних, але, як правило, зосереджуються на описовому аналізі, а не на всебічній діагностиці якості даних.

1.4.2 Функціональні можливості сучасних інструментів профілювання даних

Сучасні інструменти профілювання даних зазвичай пропонують набір основних функцій, спрямованих на узагальнення характеристик набору даних та виявлення основних проблем. Однією з основних можливостей є обчислення описової статистики, включаючи такі показники, як середнє значення, медіана, стандартне відхилення та властивості розподілу для числових ознак, а також аналіз частоти для категоріальних змінних.

Ще однією важливою функцією є виявлення пропущених значень та основних невідповідностей у даних. Інструменти часто надають інформацію про частку відсутніх даних, виявляють стовпці з неповними записами та візуалізують закономірності пропусків.

Крім того, багато систем включають кореляційний аналіз, який допомагає виявляти взаємозв’язки між змінними та виявляти надлишкові ознаки.

Візуалізація є ключовим компонентом сучасних інструментів профілювання. Автоматичне створення гістограм, діаграм розсіювання, кореляційних матриць та інших візуальних представлень дозволяє користувачам швидко зрозуміти структуру та розподіл даних. Ці візуалізації

сприяють інтуїтивній інтерпретації та полегшують виявлення потенційних аномалій або нерегулярних закономірностей.

Деякі вдосконалені інструменти також надають обмежені можливості виявлення аномалій, наприклад, виявлення екстремальних значень або нетипових розподілів. Однак ці методи зазвичай базуються на простих статистичних порогових значеннях і можуть не виявляти складні багатовимірні аномалії. Крім того, у більшості інструментів відсутні механізми виявлення більш складних проблем із якістю даних, таких як шум у мітках, реляційні невідповідності або записи, що майже дублюються.

Таким чином, хоча сучасні інструменти профілювання значно спрощують початкові етапи аналізу даних, їхня функціональність часто обмежується поверхневим вивченням наборів даних.

1.4.3 Порівняльний аналіз можливостей існуючих рішень

Аналізуючи сучасний ринок інструментів для автоматизованого аналізу якості даних, можна виявити низку спільних переваг та обмежень. Однією з головних переваг цих інструментів є їхня здатність швидко формувати вичерпні зведення даних із мінімальним втручанням користувача. Це значно скорочує час, необхідний для початкового розуміння даних, та створює зручну відправну точку для подальшого аналізу. Водночас такі інструменти, як правило, добре інтегруються в існуючі робочі процеси Data Science, що спрощує їх практичне застосування. Також вони забезпечують уніфіковане представлення базових характеристик даних, що підвищує відтворюваність початкових етапів аналізу.

Однак більшість існуючих рішень має низку обмежень, що звужують їхню застосовність у більш складних сценаріях. У таблиці 1.1 продемонстровано порівняння наявних інструментів із потенційним рішенням.

Таблиця 1.1 – Порівняння наявних рішень із запропонованим підходом

Інструмент	Описова статисти-ка	Виявлен-ня аномалій	Інтеграція ML	Виявлен-ня шуму міток	LLM-пояснен ня	Оцінка якості даних
pandas-profiling	+	-	-	-	-	-
Sweetviz	+	-	-	-	-	-
DataPrep	+	Частково	-	-	-	-
D-Tale	+	-	-	-	-	-
Great Expectations	Частково	-	Частково	-	-	Частково
AEDA (потенційно)	+	+	+	+	+	+

По-перше, існуючі рішення зосереджуються переважно на описовій статистиці та візуалізації, не виконуючи більш глибоких аналітичних завдань. Як наслідок, багато важливих проблем якості даних, таких як багатовимірні аномалії, структурні невідповідності та шум міток, залишаються невиявленими.

По-друге, більшість інструментів покладається на заздалегідь визначені статистичні методи та не використовує передові техніки машинного навчання. Це обмежує їхню здатність виявляти складні закономірності та взаємозв’язки в даних. Крім того, відсутність адаптивних або навчальних компонентів знижує їхню ефективність при роботі з різноманітними та неоднорідними наборами даних.

Ще одним істотним обмеженням є відсутність можливості інтерпретації даних, що виходить за межі базової візуалізації. Хоча інструменти надають графічні зображення та зведені показники, вони часто не пропонують змістовних пояснень виявлених проблем. Це ускладнює користувачам розуміння глибинних причин проблем із якістю даних та вжиття відповідних коригувальних заходів.

Нарешті, багато систем не розробляються з урахуванням можливості розширення. Їхня архітектура часто не підтримує легку інтеграцію нових

аналітичних методів або налаштування існуючих функцій, що обмежує їхню здатність адаптуватися до мінливих вимог.

Отже, можна зробити висновок, що хоча існуючі автоматизовані інструменти EDA надають корисні функції для початкового дослідження даних, вони не повністю вирішують завдання комплексного аналізу якості даних. Це створює потребу в більш просунутих системах, які поєднують статистичні методи, техніки машинного навчання та інтерпретовану звітність у єдиній структурі.

1.5 Обмеження існуючих підходів до аналізу якості даних

Незважаючи на наявність різноманітних інструментів для попереднього аналізу даних та їх профілювання, існуючі підходи до оцінки якості даних мають низку суттєвих обмежень. Ці обмеження знижують їхню ефективність у реальних робочих процесах у галузі науки про дані, особливо під час роботи зі складними, багатовимірними та неоднорідними наборами даних. Більшість сучасних рішень орієнтована на базовий описовий аналіз і не забезпечує комплексної структури для виявлення та інтерпретації різноманітних проблем, пов’язаних із якістю даних.

1.5.1 Недостатня глибина статистичного та алгоритмічного аналізу

Одним із основних обмежень існуючих інструментів аналізу даних є відносно низький рівень статистичної та алгоритмічної обробки, який вони забезпечують. Більшість систем зосереджується на обчисленні базових описових статистичних показників, таких як міри центральної тенденції, показники розкиду та прості характеристики розподілу. Хоча ці показники є корисними для початкового огляду даних, вони не дають змоги виявляти складні закономірності або приховані аномалії в наборі даних.

Зокрема, багато інструментів покладаються на однофакторний або парний аналіз, який розглядає ознаки незалежно або досліджує лише прості взаємозв'язки між змінними. Такі підходи є недостатніми для виявлення багатофакторних аномалій, де невідповідності стають очевидними лише при розгляді комбінацій декількох атрибутів. Як наслідок, важливі проблеми якості даних можуть залишатися невиявленими, особливо у високорозмірних наборах даних.

Крім того, виявлення аномалій часто базується на простих методах, що використовують порогові значення, такі як виявлення значень поза заздалегідь визначеними діапазонами або статистичними інтервалами. Цим підходам бракує адаптивності, і вони не є стійкими до варіацій у розподілі даних. Вони також не враховують нелінійні взаємозв’язки та складні залежності між ознаками, що є типовими для реальних наборів даних.

Загалом, обмежена глибина статистичного та алгоритмічного аналізу в існуючих інструментах обмежує їхню здатність забезпечувати всебічну оцінку якості даних, особливо в ситуаціях, що вимагають застосування передових аналітичних методів.

1.5.2 Відсутність інтеграції методів машинного навчання

Ще одним істотним обмеженням сучасних підходів є мінімальне використання методів машинного навчання в аналізі якості даних. Хоча машинне навчання демонструє високу ефективність у виявленні складних закономірностей, аномалій та невідповідностей у даних, більшість існуючих інструментів не включає такі методи у свої аналітичні процеси.

Натомість багато систем покладаються переважно на чіткі правила та заздалегідь визначені статистичні процедури. Хоча ці методи відносно прості у впровадженні та інтерпретації, вони є недостатніми для обробки складних або нелінійних структур даних. Як наслідок, їхня здатність виявляти ледь помітні аномалії, приховані залежності або нерегулярні

закономірності, які неможливо виявити за допомогою традиційних статистичних методів, є обмеженою.

Відсутність інтеграції машинного навчання також впливає на здатність системи адаптуватися до різних типів наборів даних. Підходи, засновані на правилах, зазвичай вимагають ручної конфігурації та галузевих знань, що знижує їхню гнучкість і масштабованість. На відміну від цього, методи машинного навчання можуть автоматично вивчати закономірності на основі даних, що робить їх більш придатними для аналізу різноманітних і мінливих наборів даних.

Крім того, існуючі інструменти рідко вирішують складні проблеми якості даних, такі як шум міток або ймовірнісні невідповідності, для виявлення яких потрібні спеціалізовані алгоритми. Відсутність таких можливостей ще більше обмежує ефективність сучасних рішень у сучасних додатках науки про дані.

1.5.3 Обмежені можливості інтерпретації результатів аналізу

Можливість правильно інтерпретувати дані залишається серйозною проблемою в існуючих інструментах аналізу даних. Хоча багато систем генерують візуалізації та зведені статистичні дані, вони часто не надають чітких пояснень щодо виявлених проблем якості даних. Як наслідок, користувачам доводиться вручну інтерпретувати діаграми, таблиці та числові результати, що може призвести до суб’єктивних висновків та збільшення навантаження на експертів.

Відсутність структурованих та описових пояснень ускладнює користувачам розуміння основних причин виявлених проблем. Наприклад, інструмент може вказувати на наявність виняткових значень або відсутніх значень, але не пояснює їхнього потенційного походження, значення або впливу на подальший аналіз. Ця відсутність контекстуальної інформації знижує практичну корисність результатів аналізу.

Більше того, інтерпретованість є особливо важливою для користувачів, які не володіють глибокими знаннями в галузі науки про дані. У таких випадках неможливість чітко донести аналітичні висновки стає перешкодою для ефективного прийняття рішень. Навіть досвідчені користувачі можуть стикатися з труднощами при роботі з великими та складними наборами даних, де ручна інтерпретація результатів стає дедалі складнішою.

Отже, обмежене «розуміння» даних у існуючих рішеннях підкреслює необхідність більш просунутих підходів, які не лише виявляють проблеми з якістю даних, а й надають змістовні та доступні пояснення їхніх наслідків.

1.5.4 Проблеми масштабованості та автоматизації

Нарешті, варто підкреслити проблеми масштабованості та автоматизації існуючих підходів до аналізу якості даних. Багато інструментів призначені для роботи з відносно невеликими або середніми наборами даних і можуть не працювати ефективно при застосуванні до великих масивів даних. Зі збільшенням обсягу та розмірності даних зростає обчислювальна складність, що призводить до подовження часу обробки та збільшення споживання ресурсів.

Окрім обмежень у продуктивності, багато систем вимагають значного ручного втручання в процес аналізу. Користувачам часто доводиться налаштовувати параметри, визначати правила валідації та інтерпретувати проміжні результати, що знижує рівень автоматизації та збільшує загальні витрати зусиль. Така залежність від ручного втручання обмежує відтворюваність аналізу та ускладнює інтеграцію таких інструментів у автоматизовані конвеєри обробки даних.

Проблемою також є відсутність уніфікованих робочих процесів для комплексної оцінки якості даних. Існуючі інструменти зазвичай вирішують окремі аспекти аналізу даних, такі як візуалізація або базове профілювання,

не надаючи комплексного рішення, що охоплює всі етапи оцінки якості даних. Як наслідок, користувачам доводиться поєднувати кілька інструментів та підходів, що збільшує складність системи та знижує ефективність.

1.6 Формування вимог до системи автоматизованого аналізу якості даних

Зростаюча складність сучасних систем, що базуються на даних, зумовлює необхідність розробки спеціалізованих інструментів для автоматизованої оцінки якості даних. Як зазначалося в попередніх розділах, реальні набори даних характеризуються різними типами невідповідностей, аномалій та структурних проблем, тоді як існуючі підходи до дослідницького аналізу даних залишаються або недостатньо комплексними, або значною мірою залежними від ручного втручання людини.

У цьому контексті формулювання системних вимог є вирішальним етапом у розробці автоматизованого рішення. Ці вимоги повинні відображати як практичні потреби кінцевих користувачів, так і технічні виклики, пов’язані з обробкою гетерогенних наборів даних. Чітко визначений набір вимог гарантує, що система здатна виконувати всебічний аналіз даних, надавати зрозумілі результати та ефективно працювати в реальних умовах.

1.6.1 Основні сценарії використання системи

Система автоматизованого аналізу якості даних призначена для використання в різноманітних сценаріях, пов’язаних з обробкою даних, аналізом та машинним навчанням. Один із основних напрямів використання передбачає попередню оцінку наборів даних перед розробкою аналітичних моделей. У цьому сценарії система виступає діагностичним інструментом,

що дозволяє користувачам виявляти потенційні проблеми, такі як відсутні значення, аномалії, дублікати та невідповідності, ще до навчання моделі.

Іншим важливим сценарієм використання є інтеграція системи в конвеєри машинного навчання (рисунок 1.4).

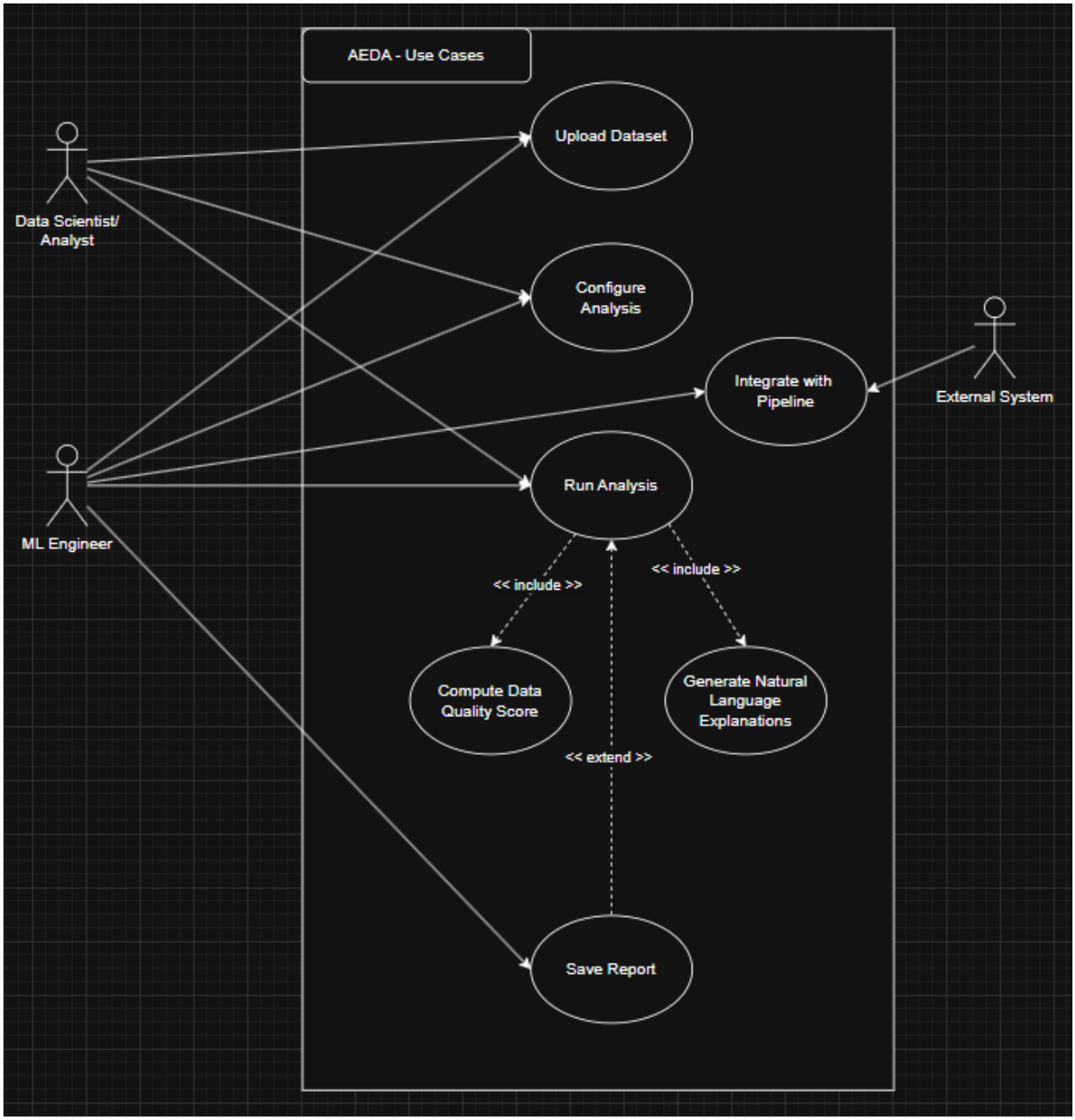


Рисунок 1.4 – Use Case діаграма застосунку AEDA

У таких випадках система працює як компонент попередньої обробки, що автоматично оцінює якість набору даних і надає стандартизовані звіти.

Це дозволяє науковцям, що працюють з даними, включити перевірки якості в відтворювані робочі процеси та зменшити ризик поширення проблем, пов’язаних з даними, на пізніші етапи розробки моделі.

Система також може застосовуватися в середовищах, де користувачі мають обмежений досвід аналізу даних. Для таких користувачів наявність автоматизованих звітів та зрозумілих пояснень є надзвичайно важливою, оскільки це дозволяє їм розуміти характеристики наборів даних без необхідності глибоких технічних знань. У цьому контексті система діє як інтелектуальний помічник, який перетворює складні аналітичні результати у зрозумілу для людини форму.

1.6.2 Функціональні вимоги до системи аналізу даних

Функціональні вимоги до системи визначають набір можливостей, які необхідно реалізувати для проведення комплексного аналізу якості даних. По-перше, система повинна підтримувати імпорт та обробку наборів даних у загальноприйнятих табличних форматах, таких як CSV, JSON, Parquet, XLSX, забезпечуючи сумісність із реальними джерелами даних. Вона повинна бути здатна обробляти набори даних різного розміру та структури, включаючи як невеликі, так і великі масиви даних.

Основною вимогою є здатність виконувати автоматизований дослідницький аналіз даних без необхідності ручної конфігурації. Система повинна аналізувати структуру наборів даних, ідентифікувати типи даних та обчислювати статистичні характеристики, необхідні для подальшого аналізу. Вона повинна реалізовувати механізми виявлення відсутніх значень, дублікатів та майже дублікатів записів, а також аномалій як у числових, так і в категоріальних даних.

Ще однією важливою вимогою є інтеграція саме сучасних аналітичних методів. Система повинна підтримувати підходи на основі машинного навчання для виявлення складних закономірностей, таких як

багатовимірні випадіння та шум міток. Крім того, вона повинна надавати механізми для оцінки реляційної узгодженості між атрибутами та виявлення порушень структурних обмежень у наборі даних.

1.6.3 Нефункціональні вимоги до програмної системи

До нефункціональних вимог, по-перше, можна віднести ефективність, оскільки система повинна бути здатна обробляти різні за величиною набори даних у «розумні» терміни. Масштабованість є ще однією критичною вимогою. Система повинна бути здатна обробляти зростаючі обсяги даних без значного зниження продуктивності. Це включає здатність обробляти набори даних, що перевищують межі обчислень в оперативній пам’яті, шляхом застосування ефективних стратегій обробки даних.

Зручність використання відіграє важливу роль, особливо для користувачів, які можуть не мати великого досвіду в галузі. Інтерфейс системи та згенеровані звіти повинні бути інтуїтивно зрозумілими та легкими для інтерпретації, що дозволяє користувачам швидко зрозуміти результати аналізу. Презентація результатів повинна бути чіткою, структурованою та підкріпленою відповідними візуалізаціями або текстовими поясненнями.

Відтворюваність також є важливою вимогою до аналітичних систем. Система повинна гарантувати, що повторний аналіз одного й того ж набору даних дасть однакові результати. Цього можна досягти за допомогою стандартизованих конфігурацій.

Ще одним важливим аспектом є надійність, тобто система повинна коректно працювати в різних умовах та галузях і обробляти можливі помилки. Для забезпечення стабільної роботи системи необхідні надійні механізми обробки помилок та перевірки даних.

Нарешті, слід враховувати розширюваність та зручність обслуговування. Архітектура системи повинна відповідати принципам

модульної побудови, що забезпечує легке оновлення, інтеграцію нових компонентів та довгострокове обслуговування.

Ці характеристики є необхідними для адаптації системи до мінливих вимог та технологічних досягнень у сфері аналізу даних.

1.7 Постановка задачі

Проведений аналіз предметної галузі показав, що якість даних відіграє вирішальну роль у забезпеченні надійності та ефективності сучасних систем, що базуються на даних. Водночас реальні вибірки зазвичай мають широкий спектр проблем, зокрема відсутність значень, дублювання, аномалії, невідповідності та шум у мітках. Ці проблеми значно ускладнюють процес аналізу даних і негативно впливають на ефективність моделей машинного навчання.

Дослідження існуючих підходів та інструментів для автоматизованого дослідницького аналізу даних показало, що, незважаючи на їхню корисність, вони не забезпечують комплексного рішення для оцінки якості даних. Більшість доступних систем обмежується базовим статистичним аналізом та візуалізацією, не має інтеграції з передовими методами машинного навчання.

У цьому контексті існує потреба у розробці автоматизованої системи, здатної виконувати комплексний аналіз якості даних для табличних наборів даних. Така система повинна поєднувати статистичні методи та підходи машинного навчання для виявлення різних типів проблем якості даних, включаючи багатовимірні аномалії, дублікати та майже дублікати записів, структурні невідповідності та помилки маркування.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ ЯКОСТІ ДАНИХ

2.1 Теоретичні основи дослідницького аналізу даних

Дослідницький аналіз даних (EDA) – це фундаментальний етап у процесі аналізу даних, спрямований на розуміння структури, властивостей та прихованих закономірностей набору даних до застосування формальних методів моделювання. Ця концепція спочатку була запроваджена як підхід, орієнтований на виявлення значущих висновків за допомогою статистичних зведень та візуальних представлень, а не виключно на основі заздалегідь сформульованих гіпотез [8].

У контексті сучасної науки про дані EDA слугує важливим механізмом для виявлення проблем із якістю даних, перевірки припущень та визначення подальших етапів попередньої обробки даних і розробки моделей. На відміну від статичних перевірок якості, які перевіряють заздалегідь сформульовані гіпотези, EDA за своєю суттю є відкритим та ітеративним процесом, що дозволяє аналітикам виявляти несподівані взаємозв’язки та аномалії в даних.

Основні цілі дослідницького аналізу даних включають:

– розуміння структури даних: виявлення типів даних, розподілів та взаємозв'язків між змінними;

– виявлення проблем із якістю даних: таких як відсутні значення, дублікати, невідповідності та аномалії;

– оцінка статистичних властивостей: центральна тенденція, дисперсія та форми розподілу;

– виявлення закономірностей та залежностей: як лінійних, так і нелінійних взаємозв’язків між атрибутами;

– підтримка прийняття рішень на етапі попередньої обробки: таких, як вибір ознак, трансформація та стратегія очищення.

EDA відіграє важливу роль у забезпеченні придатності наборів даних для подальшої аналітичної обробки. Недостатнє вивчення даних на цьому етапі може призвести до неправильних припущень, що, у свою чергу, спричинить неоптимальні або оманливі результати в моделях машинного та глибокого навчання. В автоматизованих системах оцінки якості даних, таких як розроблена бібліотека AEDA, EDA виходить за межі ручної перевірки та включає систематичні й відтворювані процедури, що поєднують статистичні методи та алгоритмічний аналіз. Таке перетворення забезпечує масштабовану та послідовну оцінку наборів даних у різних галузях.

2.1.1 Поняття та цілі EDA

З формальної точки зору набір даних можна представити у вигляді матриці $D \in R^{n \times m}$, де n позначає кількість спостережень, а m – кількість атрибутів. Мета EDA полягає у дослідженні статистичних та структурних властивостей матриці D, зокрема граничних розподілів $P(X_j)$ та спільних розподілів $P(X_i, X_j)$, та потенційних залежностей між змінними. Такий аналіз дозволяє виявити відхилення від очікуваних закономірностей, що може вказувати на проблеми з якістю даних або на явища, характерні для конкретної галузі.

Центральним аспектом EDA є оцінка описової статистики, що характеризує набір даних. Для числової ознаки X ключовими властивостями є математичне очікування та дисперсія, які визначаються відповідно як:

$$\mu = E[X] = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{2.1}$$

$$\sigma^2 = \mathrm{Var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2, \tag{2.2}$$

де μ – математичне очікування;

σ^2 – дисперсія;

x_i – і-те значення ознаки x;

n – загальна кількість спостережень (обсяг вибірки).

Ці показники дають уявлення про центральну тенденцію та розкид даних, створюючи основу для подальшого аналізу, такого, як нормалізація або виявлення аномалій. Окрім числових характеристик, категоріальні змінні досліджуються за допомогою частотних розподілів та показників, заснованих на ентропії, що дозволяє оцінити різноманітність та дисбаланс у даних.

EDA також слугує важливим механізмом для виявлення проблем із якістю даних, включаючи відсутні значення, невідповідності та структурні порушення. Наявність відсутніх значень у ознаці X_j можна кількісно оцінити за допомогою коефіцієнта відсутності:

$$\mathrm{r_{missing}(X_j)} = \frac{\sum_{i=1}^{n} \mathrm{I(x_{ij}=\emptyset)}}{\mathrm{n}}, \tag{2.3}$$

де I – індикаторна функція.

Високі значення цього коефіцієнта можуть свідчити про неповноту процесів збору даних або про систематичні похибки, що можуть суттєво вплинути на результати подальшого аналізу.

У контексті сучасної науки про дані EDA виходить за межі ручної перевірки і стає невід’ємною частиною автоматизованих систем оцінки якості даних. Її цілі більше не обмежуються описовим розумінням, а також включають систематичне виявлення проблем, які можуть поставити під загрозу валідність аналітичних моделей. Отже, EDA виступає мостом між збором необроблених даних і розробкою надійних моделей і гарантує, що наступні етапи ґрунтуються на точних і перевірених даних.

2.1.2 Основні етапи дослідницького аналізу даних

Попередній аналіз даних можна розглядати як структурований, але ітеративний процес, спрямований на поступове виявлення властивостей та проблем набору даних. Хоча на практиці аналітики часто переходять від одного етапу до іншого, можна виділити кілька концептуальних етапів, які разом утворюють цілісний аналітичний робочий процес.

Зазвичай процес починається з вивчення загальної структури набору даних. На цьому етапі увага зосереджується на розумінні розмірності даних, кількості спостережень та атрибутів, а також типів наявних змінних. Це включає розрізнення між числовими, категоріальними та часовими ознаками, а також виявлення невідповідностей у представленні даних. Структурний аналіз забезпечує необхідну основу, оскільки помилки в типах даних або визначеннях схем можуть поширюватися на наступні етапи та ставити під загрозу достовірність аналізу.

Після структурного огляду аналіз переходить до оцінки якості даних. Цей етап стосується виявлення недоліків у наборі даних, які можуть вплинути на його придатність для використання. Типові проблеми включають відсутні значення, дубльовані записи та недійсні або суперечливі записи. Виявлення таких проблем є критично важливим, оскільки вони можуть спричинити упередженість, знизити статистичну потужність або призвести до неправильних висновків, якщо їх не усунути. У багатьох реальних сценаріях цей етап вимагає ретельного розбору, оскільки певні невідповідності можуть відображати реальні характеристики конкретної галузі, а не помилки.

Наступний етап передбачає статистичну характеристику набору даних. Тут основна увага приділяється узагальненню розподільчих властивостей окремих ознак. Показники центральної тенденції та мінливості використовуються для опису типових значень та розкиду, тоді як аналіз розподілу допомагає виявити асиметрію, важкі хвости або інші

нерегулярні закономірності. Цей етап дозволяє аналітику отримати уявлення про загальну поведінку даних та визначити, чи необхідні перетворення або процедури нормалізації.

Важливим розширенням цього аналізу є дослідження взаємозв'язків між змінними. Розуміння того, як атрибути взаємодіють між собою, є необхідним для виявлення залежностей, надлишковості та потенційних причинно-наслідкових структур. Лінійні взаємозв'язки часто можна виявити за допомогою кореляційного аналізу, тоді як для більш складних взаємодій можуть знадобитися просунуті статистичні методи або методи машинного навчання. Виявлення таких відношень відіграє вирішальну роль у створенні нових ознак та зменшенні розмірності.

Іншим ключовим компонентом дослідницького аналізу даних є виявлення аномалій. Це спостереження, що значно відхиляються від загальних закономірностей, присутніх у наборі даних. Аномалії можуть виникати через помилки у зборі даних, шум або справді рідкісні події. Їх виявлення важливе не лише для очищення даних, а й для розуміння виняткових випадків, що можуть містити цінну інформацію. У складних наборах даних аномалії часто проявляються лише в контексті декількох змінних, що робить їх виявлення нетривіальним завданням.

Останній етап дослідницького процесу передбачає інтерпретацію та узагальнення результатів. Інформація, отримана під час аналізу, має бути зібрана у цілісну форму, яка може сприяти прийняттю рішень. Це включає складання ключових висновків, висвітлення потенційних проблем та надання пояснень щодо спостережуваних закономірностей. У сучасних системах цей етап часто підтримується автоматизованими інструментами звітності, які генерують структуровані результати та описові пояснення, тим самим покращуючи доступність та зручність використання аналізу. Водночас ефективність цього етапу значною мірою залежить від рівня системи, зокрема її здатності формувати контекстно узгоджені та семантично зрозумілі висновки.

2.1.3 Роль автоматизації у сучасних системах аналізу даних

Швидке поширення додатків, що базуються на даних, значно збільшило обсяг і складність наборів даних, через що ручний дослідницький аналіз даних стає дедалі менш доцільним. Як наслідок, автоматизація стала центральним компонентом сучасних систем аналізу даних, що забезпечує ефективну, послідовну та масштабовану оцінку даних.

Системи автоматизованого дослідницького аналізу даних призначені для формалізації аналітичного процесу шляхом його перетворення на послідовність заздалегідь визначених обчислювальних кроків. Замість того, щоб покладатися на ситуативні рішення, прийняті аналітиками, ці системи застосовують стандартизовані процедури для оцінки різних аспектів набору даних. Цей перехід від ручного до автоматизованого аналізу зменшує суб’єктивність та гарантує, що подібні набори даних обробляються стандартизовано.

По мірі зростання розмірів наборів даних, ручна перевірка стає складнішою як з точки зору часу, так і ресурсів. Автоматизовані системи вирішують цю проблему, використовуючи оптимізовані методи обробки даних та ефективні алгоритми. У багатьох випадках вони включають такі стратегії, як вибірка піднаборів даних або паралельна обробка, щоб підтримувати продуктивність без втрати глибини аналізу.

Ще однією ключовою перевагою автоматизації є відтворюваність. У ручних робочих процесах результати часто залежать від досвіду та інтуїції аналітика, що може призвести до варіативності результатів. Натомість автоматизовані системи дають детерміновані результати при застосуванні до одного й того ж набору даних за однакових налаштувань. Ця властивість є особливо важливою в наукових та промислових проєктах, де послідовність та відтворюваність результатів є надзвичайно важливими.

Автоматизація також дозволяє проводити більш комплексний аналіз якості даних. Інтегруючи численні аналітичні методи в єдину систему,

автоматизовані системи можуть одночасно оцінювати різні аспекти якості даних, включаючи повноту, послідовність, унікальність та властивості розподілу. Такий цілісний підхід дозволяє виявляти складні проблеми, які можуть бути непомітними при ізольованих перевірках.

На додаток до класичних статистичних методів, сучасні автоматизовані системи все частіше використовують методи машинного навчання для розширення своїх аналітичних можливостей. Ці методи дозволяють виявляти тонкі закономірності та аномалії, які неможливо виявити за допомогою традиційних підходів. Як результат, автоматизовані системи здатні надавати глибше розуміння структури та якості даних.

Особливо важливим досягненням є інтеграція технологій обробки природної мови для інтерпретації аналітичних результатів. Генеруючи текстові пояснення виявлених закономірностей та проблем, автоматизовані системи заповнюють прогалину між технічним аналізом та людським розумінням.

У контексті розробленої системи AEDA автоматизація реалізується за допомогою модульної архітектури, яка організовує аналіз у вигляді потоку «компонент якості». Така конструкція не тільки покращує масштабованість і розширюваність, але й гарантує, що аналіз залишається структурованим і відтворюваним. Крім того, інтеграція методів машинного навчання та мовних моделей дозволяє системі надавати як кількісні, так і якісні оцінки, тим самим підвищуючи її практичну цінність у реальних робочих процесах аналізу даних.

2.2 Методи виявлення проблем якості даних

2.2.1 Методи виявлення пропущених значень та неповних даних

Відсутні значення є однією з найпоширеніших і найважливіших причин погіршення якості даних у реальних наборах даних. Їхня наявність

зменшує обсяг доступної інформації та ускладнює статистичний аналіз. Відповідно, виявлення та аналіз відсутніх значень є фундаментальною складовою попереднього аналізу даних.

З формальної точки зору, відсутність даних можна розглядати як властивість набору даних, в якому певні записи не визначені або відсутні. Масштаб цієї проблеми зазвичай кількісно оцінюється на рівні окремих ознак за допомогою коефіцієнта відсутності даних, який відображає частку спостережень, що не мають дійсних значень. Цей показник дає пряме уявлення про повноту набору даних і дозволяє виявити атрибути, які можуть потребувати особливого поводження або виключення.

Окрім простої кількісної оцінки, важливо проаналізувати структуру та механізми, що лежать в основі відсутніх даних. У статистичній теорії відсутність даних зазвичай класифікують на три типи: повністю випадкова відсутність (MCAR), випадкова відсутність (MAR) та невипадкова відсутність (MNAR) [9]. MNAR представляє найскладніший сценарій, де відсутність даних залежить від самого неспостережуваного значення. Розрізнення цих механізмів є надзвичайно важливим, оскільки безпосередньо впливає на вибір відповідних стратегій обробки.

Деякі методи виявлення виходять за межі підрахунку відсутніх записів і включають аналіз закономірностей для виявлення кореляцій між відсутністю даних та іншими змінними. Наприклад, спільна відсутність значень у декількох ознаках може вказувати на системні проблеми у процесах збору або інтеграції даних. Для виявлення таких закономірностей часто використовуються методи візуалізації, такі як матриці відсутності даних або теплові карти, хоча в автоматизованих системах ці закономірності зазвичай узагальнюються за допомогою статистичних показників.

Крім того, важливо враховувати розподіл відсутніх значень по підмножинах даних. Наприклад, якщо відсутні значення, зосереджені в певних групах або діапазонах, це може свідчити про приховані упередження або ефекти сегментації. Автоматизовані системи аналізу вирішують цю

проблему шляхом обчислення показників відсутності даних у різних розділах набору даних, що дозволяє отримати більш детальну інформацію.

У контексті автоматизованого аналізу якості даних, виявлення відсутніх значень зазвичай інтегровано в ширший конвеєр, який не тільки кількісно оцінює їх наявність, але й аналізує їхній потенційний вплив на подальші завдання. Це включає виявлення критичних ознак із високим рівнем відсутності даних, ранжування атрибутів за їхньою повнотою та надання рекомендацій щодо подальшої обробки. Такі систематичні підходи гарантують, що неповні дані не тільки виявляються, а й правильно контекстуалізуються в рамках загального аналізу.

2.2.2 Методи виявлення дублікатів та схожих записів

Дублікати записів становлять ще одну значну категорію проблем із якістю даних, що виникають, коли один і той самий об’єкт реального світу представлений у наборі даних кілька разів. Ці дублікати можуть бути точними копіями або наближеними представленнями з незначними відмінностями, які часто називають «майже-дублікатами». Обидва типи дублювання можуть спотворювати статистичні розподіли, призводити до упередженості аналітичних результатів та негативно впливати на ефективність моделей машинного навчання.

Виявлення точних дублікатів концептуально є простим і ґрунтується на ідентифікації рядків, що є ідентичними за всіма атрибутами.

Однак реальні набори даних часто містять близькі дублікати, які дещо відрізняються через невідповідності у форматуванні, друкарські помилки або користувацькі помилки при введенні даних. Виявлення таких записів вимагає більш складних підходів, заснованих на мірах схожості. Ці методи спрямовані на кількісну оцінку ступеня схожості між записами шляхом порівняння значень їх атрибутів за допомогою метрик відстані або схожості.

Для текстових атрибутів схожість часто оцінюється за допомогою підходів на основі токенів, таких як перекриття n-грам або схожість за Джакардом, яка вимірює співвідношення спільних елементів між двома наборами [10]. У випадку числових атрибутів схожість можна оцінити на основі відносних відмінностей або нормалізованих відстаней. При роботі із записами із багатьма атрибутами ці окремі ознаки зазвичай об’єднуються в сукупний бал, що відображає загальну схожість між записами.

Для забезпечення масштабованості, особливо для великих наборів даних, зазвичай застосовуються наближені методи, такі як хешування з урахуванням локальності (рисунок 2.1).

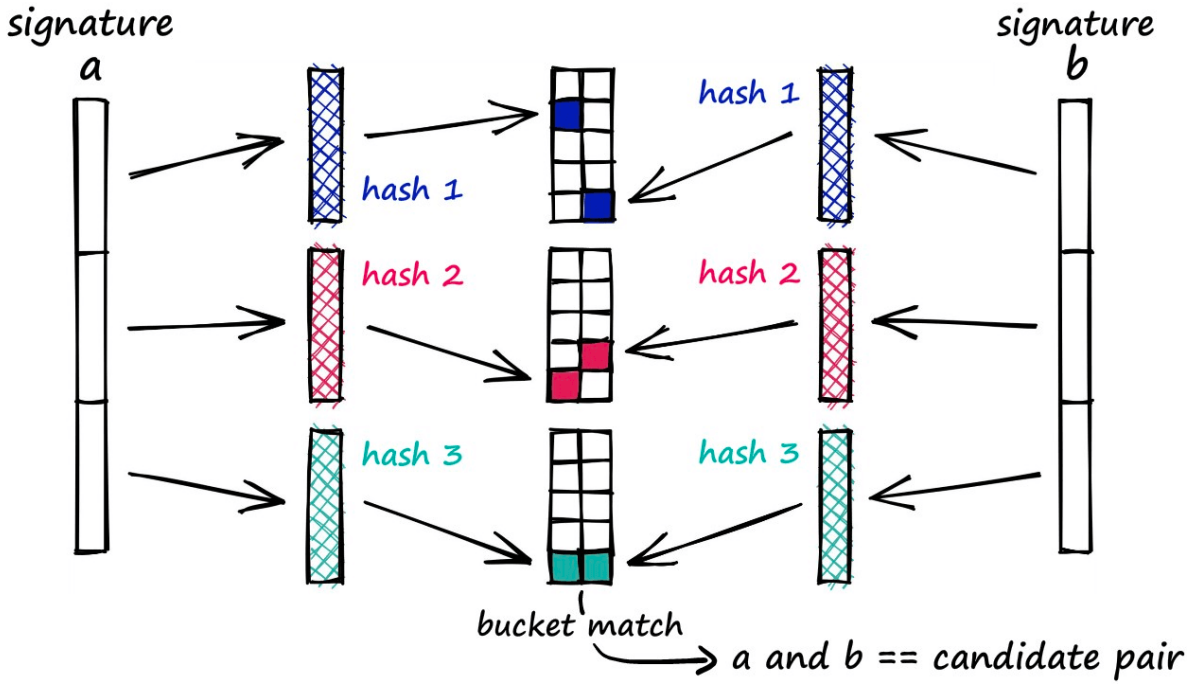


Рисунок 2.1 – Принцип роботи алгоритму LSH

В автоматизованих системах аналізу даних виявлення дублікатів зазвичай інтегровано з іншими компонентами оцінки якості. Система може обчислювати зведені показники, такі як частка записів із дублюванням, визначати найпоширеніші шаблони та надавати репрезентативні приклади.

Цей комплексний підхід дозволяє користувачам зрозуміти як масштаб, так і характер дублювання в наборі даних.

2.2.3 Методи статистичного аналізу структури даних

Статистичний аналіз структури даних складає основу для розуміння властивостей та організації набору даних. На відміну від простих описових узагальнень, структурний аналіз зосереджується на виявленні закономірностей, відхилень та взаємозв'язків, що визначають базовий розподіл даних.

Одним із ключових аспектів цього аналізу є вивчення розподілів ознак. Для числових змінних це передбачає оцінку таких властивостей, як дисперсія та тенденція. Аналіз розподілу допомагає виявити асиметрію, «важкі хвости» та мультимодальність, що може вказувати на приховану гетерогенність або наявність субпопуляцій у даних. Для категоріальних змінних використовуються частотні розподіли для оцінки збалансованості та різноманітності значень, які можуть виявити такі проблеми, як дисбаланс класів або рідкісні категорії.

Іншим важливим компонентом є аналіз кардинальності ознак. Атрибути з високою кардинальністю, особливо ті, що мають значення, близькі до кількості спостережень, часто є ідентифікаторами або квазі-ідентифікаторами. Хоча такі ознаки можуть бути корисними для індексації, вони зазвичай не містять прогнозної інформації і можуть потребувати особливого підходу під час аналізу. І навпаки, ознаки з низькою кардинальністю можуть представляти категоріальні змінні з обмеженим набором можливих значень, що може потребувати, наприклад, кодування у машинному навчанні.

Аналіз взаємозв'язків між змінними також сприяє розумінню структури даних. Кореляційний аналіз зазвичай використовується для виявлення лінійних залежностей між числовими ознаками, тоді як міри

асоціації застосовуються до категоріальних змінних. Ці відношення можуть виявити надмірність, мультиколінеарність або функціональні залежності, які можуть вплинути на ефективність моделі.

Окрім парних взаємозв'язків, структурний аналіз може передбачати виявлення залежностей та закономірностей вищого порядку. Наприклад, групи змінних можуть демонструвати спільну поведінку, яка не є очевидною при розгляді окремо. Виявлення таких закономірностей часто вимагає застосування передових статистичних методів або методів машинного навчання, здатних фіксувати складні взаємодії.

Виявлення винятків також є частиною структурного аналізу, оскільки екстремальні значення можуть суттєво впливати на статистичні властивості та спотворювати інтерпретації. Виявлення та аналіз цих значень допомагають гарантувати, що набір даних точно відображає базові явища і не має домінування нетипових спостережень.

Врешті, в автоматизованих системах структурний аналіз реалізується за допомогою поєднання описових метрик, аналізу розподілу та виявлення залежностей.

2.3 Методи виявлення аномалій у даних

2.3.1 Статистичні підходи до виявлення аномалій

Статистичні методи є найдавнішими та найбільш зрозумілими підходами до виявлення аномалій [11]. Вони ґрунтуються на припущенні, що нормальні точки даних відповідають базовому розподілу, тоді як аномалії відповідають подіям з низькою ймовірністю.

Поширена стратегія полягає в аналізі розкиду значень у межах певної ознаки та виявленні спостережень, що виходять за межі очікуваних діапазонів. Для цього широко використовуються методи, засновані на стандартному відхиленні або міжквартильних розмахах, оскільки вони

надають прості та ефективні критерії для виявлення екстремальних значень. Такі методи є обчислювально ефективними та легко інтерпретованими, що робить їх придатними для початкової оцінки якості даних.

Однак статистичні підходи мають кілька обмежень. Зазвичай вони припускають незалежність між ознаками та призначені переважно для однофакторного аналізу. Як наслідок, вони можуть не виявити аномалій, що виникають через складні взаємодії між декількома змінними. Крім того, їхня ефективність залежить від обґрунтованості припущень щодо розподілу, які часто порушуються в реальних наборах даних, що характеризуються асиметрією, мультимодальністю або «важкими хвостами».

Незважаючи на ці обмеження, статистичні методи залишаються цінною основою в системах автоматизованого аналізу даних. У контексті програмного застосунку, що розробляється, такі підходи можуть використовуватися для швидкого попереднього скринінгу наборів даних, надаючи базові зведення про потенційні аномалії перед застосуванням більш просунутих методів.

2.3.2 Алгоритми машинного навчання для пошуку аномалій

Щоб подолати обмеження суто статистичних методів, сучасні підходи все частіше спираються на алгоритми машинного навчання, здатні моделювати складні та багатовимірні розподіли даних. Ці методи не вимагають суворих припущень щодо вихідних даних і краще підходять для виявлення ледь помітних та багатовимірних аномалій.

Методи виявлення аномалій на основі машинного навчання можна умовно розділити на контрольовані, напівконтрольовані та неконтрольовані підходи. На практиці неконтрольовані та напівконтрольовані методи частіше використовуються в аналізі якості даних. Ці підходи зосереджуються на вивченні структури нормальних даних та виявленні відхилень від цього вивченого представлення.

Методи на основі кластеризації виявляють аномалії як точки, що не належать до жодного щільного кластера або розташовані далеко від центроїдів кластерів.

Методи на основі щільності ідентифікують аномалії як спостереження, розташовані в областях з низькою щільністю простору ознак.

Ключовою перевагою підходів машинного навчання є їхня здатність фіксувати складні взаємозв'язки між змінними. Це робить їх особливо ефективними у високорозмірних наборах даних, де аномалії можуть бути невиявними за допомогою аналізу окремих ознак. Однак ці методи часто накладають вищі обчислювальні витрати та знижену інтерпретованість порівняно зі статистичними техніками.

У системі AEDA алгоритми машинного навчання слугують розширеним рівнем аналізу, що доповнює простіші статистичні перевірки. Поєднуючи кілька підходів, система здатна виявляти ширший спектр аномалій – від очевидних винятків до ледь помітних відхилень, вбудованих у багатовимірні структури.

2.3.3 Алгоритм Isolation Forest у задачах виявлення викидів

Серед сучасних алгоритмів виявлення аномалій Isolation Forest зарекомендував себе як особливо ефективний метод [12]. Його концептуальна основа відрізняється від традиційних підходів, оскільки базується на ідеї ізоляції аномалій, а не на явному моделюванні нормальних даних.

Алгоритм працює шляхом побудови ансамблю випадкових дерев рішень, де кожне дерево рекурсивно розбиває дані, випадково вибираючи ознаку та значення розбиття. Спостереження, які легше ізолювати, тобто для яких потрібно менше кроків розбиття, з більшою ймовірністю є аномаліями (рисунок 2.2). Ця властивість дозволяє алгоритму присвоювати

кожній точці даних оцінку аномальності на основі середньої довжини шляху по всьому ансамблю.

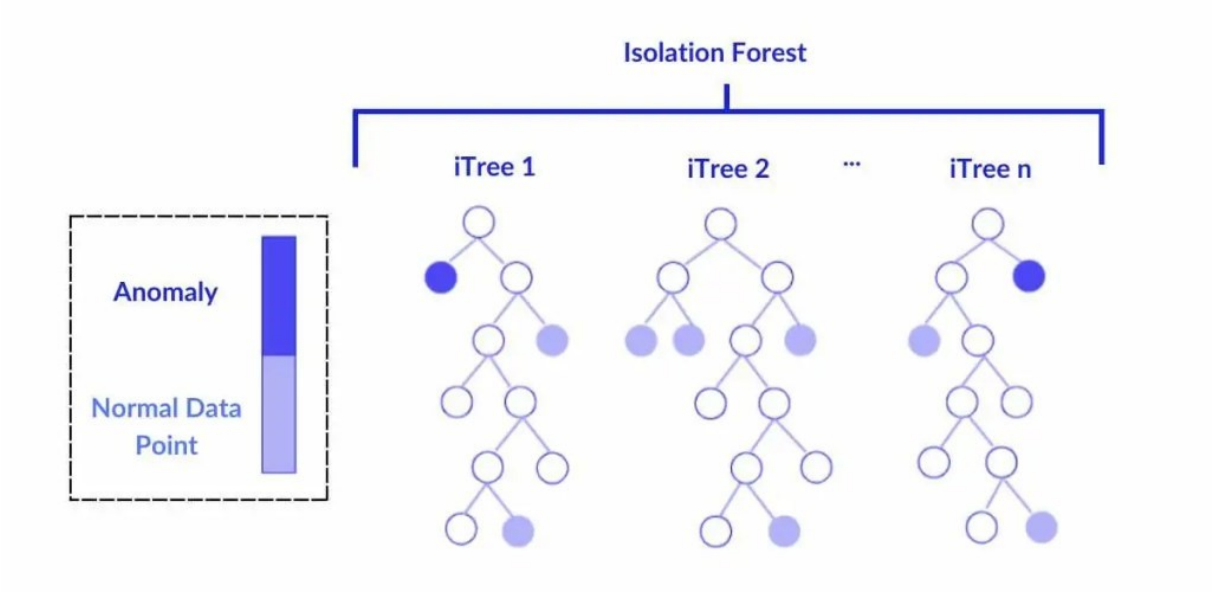


Рисунок 2.2 – Алгоритм Isolation Forest

Важливою перевагою Isolation Forest є його обчислювальна ефективність. Оскільки він спирається на випадкове розбиття, а не на обчислення відстані чи щільності, він добре масштабується до великих наборів даних та високорозмірних просторів ознак. Крім того, він не вимагає припущень щодо розподілу даних, що робить його надійним у різноманітних реальних сценаріях.

На відміну від одновимірних статистичних методів, цей алгоритм неявним чином фіксує взаємодії між ознаками через рекурсивний поділ. Це робить його особливо придатним для виявлення складних аномалій, що виникають внаслідок комбінацій значень атрибутів, що є дуже важливим кроком у автоматизації процесу, адже фактично пристосовує алгоритм до будь-якої предметної галузі.

Однак цей метод має свої обмеження. Його ефективність може залежати від вибору параметрів, таких як кількість дерев та розмір підвибірки.

У бібліотеці AEDA, яка розробляється в рамках цієї роботи, Isolation Forest є основним компонентом для виявлення аномалій. Його баланс між ефективністю та результативністю робить його придатним для автоматизованого аналізу якості даних, де масштабованість та універсальність є критичними вимогами. Використання цього алгоритму в системі виправдане його здатністю виявляти як прості, так і складні аномалії, не покладаючись на сильні припущення щодо даних, тим самим доповнюючи інші аналітичні методи, розглянуті в цьому розділі.

2.4 Методи виявлення помилок цільової функції

2.4.1 Проблема шуму у мітках у задачах навчання з учителем

У задачах навчання з учителем якість моделі напряму залежить від якості цільової функції. Мітки слугують еталоном, який направляє процес навчання, і будь-які невідповідності чи помилки в цих мітках безпосередньо впливають на здатність моделі до узагальнення. У реальних наборах даних шум є поширеною проблемою, що виникає внаслідок людських помилок, неоднозначності в інструкціях щодо анотації тощо.

Шум міток може проявлятися у різних формах, включаючи випадкове неправильне позначення та систематичні помилки. Випадковий шум виникає, коли мітки присвоюються неправильно без будь-якої послідовної закономірності, тоді як систематичний шум відображає упередження або повторювані неточності в процесі маркування. Останнє є особливо проблематичним, оскільки може вводити оманливі закономірності, які моделі вивчають і закріплюють.

Саме тому з точки зору якості даних виявлення помилок у мітках є надважливим завданням. На відміну від інших типів проблем з даними, помилки міток часто не можна легко помітити, і для їх виявлення потрібні непрямі методи. Це робить їх виявлення більш складним і вимагає використання статистичних підходів та підходів на основі машинного навчання.

У контексті автоматизованих систем аналізу даних, таких як AEDA, виявлення шуму міток розглядається як аналітичне завдання вищого рівня, що базується на раніше виявлених шаблонах та оцінках на основі моделей. Його включення відображає необхідність оцінювати не лише структуру даних, а й надійність їхніх семантичних анотацій.

2.4.2 Підхід Confident Learning для виявлення помилкових міток

Одним із найвідоміших сучасних підходів до виявлення помилок у мітках є методологія «Confident Learning» [13]. Цей метод ґрунтується на ідеї, що невідповідності між прогнозованими ймовірностями та спостережуваними мітками можна використовувати для оцінки ймовірності неправильного маркування.

Основний принцип Confident Learning полягає в моделюванні спільного розподілу між мітками з шумом та справжніми мітками шляхом використання прогнозованих ймовірностей класів, отриманих від навченого класифікатора. Замість того, щоб припускати, що спостережувані мітки є правильними, метод оцінює, наскільки впевнено модель відносить зразок до певного класу. Якщо зразок позначений як один клас, але постійно отримує високі оцінки ймовірності для іншого класу, він вважається кандидатом на помилку мітки Такий підхід дозволяє систематично виявляти приховані помилки розмітки навіть у складних багатокласових наборах даних без необхідності ручної перевірки (рисунок 2.3).

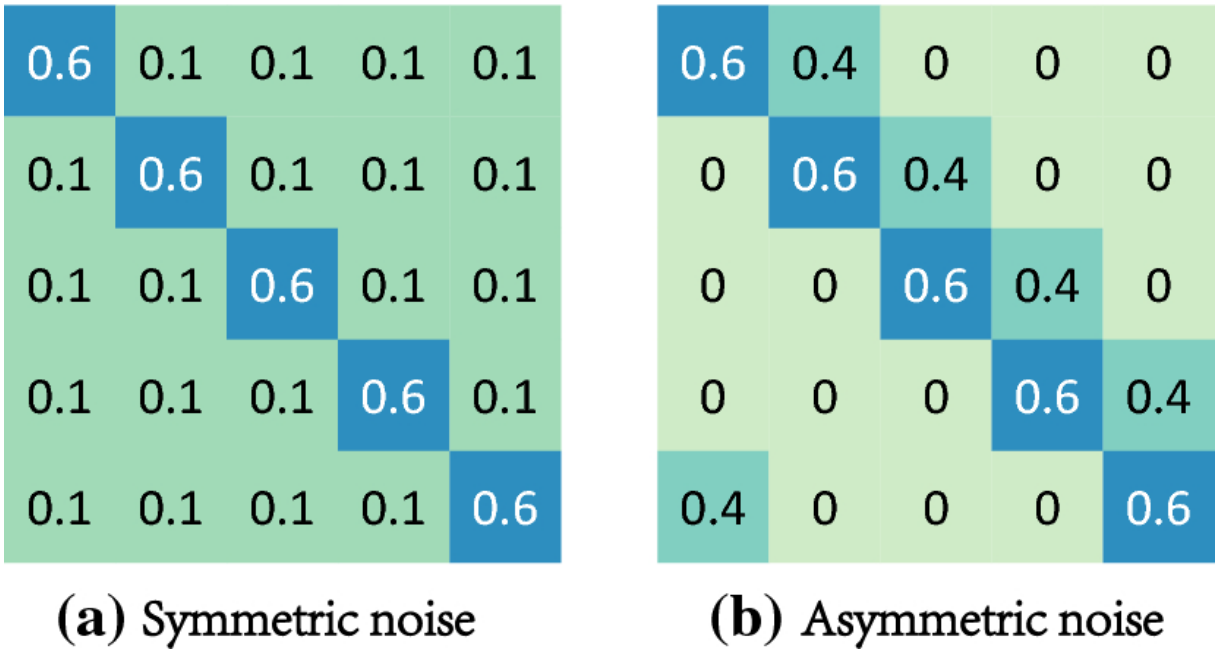


Рисунок 2.3 – Приклад матриці Label Noise

Цей підхід вводить концепцію механізму фільтрації на основі впевненості, де зразкам присвоюються ранги відповідно до ймовірності їх помилкового маркування. Вибираючи екземпляри з найбільшою розбіжністю між прогнозованими та спостережуваними мітками, метод ефективно ізолює потенційно неправильні анотації.

Ключовою перевагою Confident Learning є те, що він не вимагає доступу до чистих міток («ground truth»). Натомість він покладається на здатність моделі апроксимувати справжній розподіл міток. Це робить його особливо придатним для реальних сценаріїв, де ідеально розмічені набори даних є рідкістю.

Однак ефективність підходу залежить від якості базового класифікатора. Якщо сама модель має значне упередження або недостатньо навчена, її прогнози можуть не надавати надійних сигналів для виявлення помилок міток. Тому ретельний вибір та валідація моделі є важливими факторами при застосуванні цього методу. Тому бібліотека AEDA, що розробляється, може спиратися на Confident Learning як на принциповий механізм для виявлення підозрілих міток.

2.4.3 Використання ансамблів моделей для оцінювання достовірності міток

Альтернативний та доповнюючий підхід до виявлення помилок у мітках передбачає використання ансамблевих методів [14]. Замість того, щоб покладатися на одну модель, ця стратегія використовує декілька моделей, навчених на одному й тому ж наборі даних, для оцінки узгодженості прогнозів.

Основне припущення полягає в тому, що правильно позначені зразки дадуть узгоджені прогнози в різних моделях, тоді як неправильно позначені екземпляри призведуть до розбіжностей. Аналізуючи рівень узгодженості всередині ансамблю, можна оцінити надійність кожної мітки.

Підходи на основі ансамблю можна реалізувати за допомогою моделей з різною архітектурою, конфігураціями навчання або підмножинами даних. Ця різноманітність є важливою, оскільки вона зменшує ймовірність того, що всі моделі зроблять однаковий неправильний прогноз. Ступінь розбіжностей між моделями можна потім кількісно оцінити та використовувати як показник потенційного шуму міток.

Однією з переваг цього підходу є його стійкість до індивідуальних похибок моделей. Оскільки остаточна оцінка базується на агрегованих прогнозах, помилки, допущені однією моделлю, рідше домінують у результаті. Крім того, ансамблеві методи можуть враховувати різні перспективи щодо даних, покращуючи загальне виявлення неоднозначних або граничних випадків.

Однак використання ансамблів створює додаткове обчислювальне навантаження, оскільки необхідно навчати та оцінювати декілька моделей. Це може обмежувати їхню застосовність у масштабних датасетах або середовищах з обмеженими ресурсами. Крім того, хоча розбіжність є корисним сигналом, вона не завжди означає неправильне маркування, оскільки може також відображати властиву даним неоднозначність.

У рамках AEDA аналіз на основі ансамблів можна розглядати як розширення оцінки якості на основі моделей. Він доповнює Confident Learning, надаючи додатковий рівень валідації на основі консенсусу. Разом ці методи сприяють більш надійному виявленню проблем, пов'язаних з мітками, які інакше важко виявити суто статистичними засобами.

2.5 Методи аналізу структурної узгодженості даних

2.5.1 Поняття функціональних залежностей у табличних даних

Структурна узгодженість набору даних відображає ступінь, у якому взаємозв'язки між його атрибутами відповідають чітко визначеним і стабільним закономірностям. Одним із фундаментальних понять, що використовуються для формалізації таких взаємозв'язків у табличних даних, є поняття функціональних залежностей. Функціональна залежність описує обмеження між двома наборами атрибутів, за якого значення одного набору однозначно визначає значення іншого [15].

На практиці залежність у формі $X \rightarrow Y$ означає, що для будь-яких двох записів у наборі даних, якщо вони мають однакові значення атрибуту X, вони також повинні мати однакові значення атрибуту Y. Ця концепція широко використовується в теорії баз даних для забезпечення нормалізації та усунення надмірності, але вона також відіграє важливу роль в аналізі якості даних.

Наявність дійсних функціональних залежностей відображає внутрішню узгодженість у наборі даних, тоді як їх порушення вказує на потенційні проблеми з якістю даних. Наприклад, якщо атрибут, що представляє поштовий індекс, повинен визначати конкретне місто, будь-яке відхилення від цього зв'язку може сигналізувати про помилки під час введення або інтеграції даних (таблиця 2.1).

Таблиця 2.1 – Приклад порушення функціональної залежності

Поштовий індекс	Місто	Порушення?
61000	Харків	Ні
61000	Харків	Ні
61000	Київ	Так (порушення FD)

У дослідницькому аналізі даних функціональні залежності не завжди відомі заздалегідь, і їх необхідно виводити з даних. Цей процес передбачає пошук комбінацій атрибутів, які демонструють детерміновані або майже детерміновані зв’язки. У реальних наборах даних строгі залежності часто замінюються наближеними, де взаємозв’язок зберігається для більшості, але не для всіх записів.

У системі AEDA концепція функціональних залежностей забезпечує теоретичну основу для аналізу структурних взаємозв’язків між атрибутами. Виявляючи стабільні закономірності та відхилення від них, система може виявляти невідповідності, які не є очевидними при простому статистичному аналізі.

2.5.2 Методи виявлення порушень цілісності даних

Цілісність даних означає правильність і достовірність даних стосовно заздалегідь визначених або виведених обмежень. Порушення цілісності виникають, коли дані не відповідають цим обмеженням, що призводить до невідповідностей, які можуть поставити під сумнів результати аналізу.

Один із видів обмежень цілісності – це обмеження домену, за яким значення атрибутів повинні знаходитися в певному діапазоні або наборі. Порушення відбуваються, коли значення виходять за межі цих очікуваних меж, що вказує на можливі помилки або неправильні тлумачення. Інший вид – це обмеження унікальності, які вимагають, щоб певні атрибути або

комбінації атрибутів однозначно ідентифікували записи. Наявність дублікатів або суперечливих записів безпосередньо порушує ці обмеження.

Сучасні методи виявлення порушень цілісності базуються як на підходах, що ґрунтуються на правилах, так і на підходах, що базуються на даних. Методи на основі правил застосовують заздалегідь визначені обмеження, похідні від знань про домен або визначень схем. Хоча вони є ефективними, вони вимагають попередньої специфікації та можуть не охоплювати всі відповідні шаблони. З іншого боку, підходи на основі даних виводять обмеження безпосередньо з даних шляхом виявлення закономірностей та відхилень від них. Ці методи є більш гнучкими та краще підходять для дослідницького аналізу, особливо коли знання про домен є обмеженими.

2.6 Методи моделювання розподілів даних

2.6.1 Підходи до аналізу багатовимірних розподілів

Моделювання розподілів є важливим у виявленні відхилень та розумінні загальної структури набору даних. Хоча однофакторний аналіз дає уявлення про окремі ознаки, реальні набори даних за своєю суттю є багатовимірними, і багато важливих закономірностей виявляються лише при розгляді відношень між змінними.

Аналіз багатовимірного розподілу має на меті виявити спільну поведінку декількох атрибутів. Класичні підходи ґрунтуються на припущеннях щодо форми розподілу, таких як багатовимірна нормальність, і використовують статистичні показники для опису залежностей та мінливості. Ці методи є ефективними, коли дані відповідають відносно простим структурам, але їхня ефективність, як правило, знижується із збільшенням розмірності та ускладненням розподілів.

Ключовим викликом у багатовимірному аналізі є так зване «прокляття розмірності», коли об’єм простору ознак зростає експоненційно з кількістю вимірів [16]. Як наслідок, дані стають розрідженими, а традиційні методи, засновані на відстані або щільності, втрачають свою дискримінаційну здатність. Це обмеження знижує ефективність класичних статистичних методів у високорозмірних умовах.

Щоб вирішити ці проблеми, сучасні підходи дедалі частіше спираються на методи, що базуються на даних, які навчаються базовій структурі безпосередньо з даних. Ці методи не вимагають жорстких припущень щодо розподілу і краще підходять для виявлення складних, нелінійних взаємозв’язків між атрибутами. У контексті автоматизованого аналізу якості даних такі підходи забезпечують більш надійну основу для виявлення тонких невідповідностей та нерегулярностей.

Тому у застосунку AEDA багатовимірне моделювання розподілу має слугувати розширенням попереднього статистичного аналізу, дозволяючи системі перейти від незалежної оцінки ознак до цілісного розуміння структури набору даних.

2.6.2 Використання автоенкодерів для виявлення аномалій

Автоенкодери – це клас моделей нейронних мереж, спеціально розроблених для навчання компактним представленням даних шляхом кодування вхідних даних у простір з меншою розмірністю та їх подальшої реконструкції [17].

Їх застосування для виявлення аномалій ґрунтується на припущенні, що модель, навчена на типових даних, навчиться точно відтворювати нормальні закономірності (рисунок 2.4). При цьому вона не зможе відтворити аномальні спостереження.

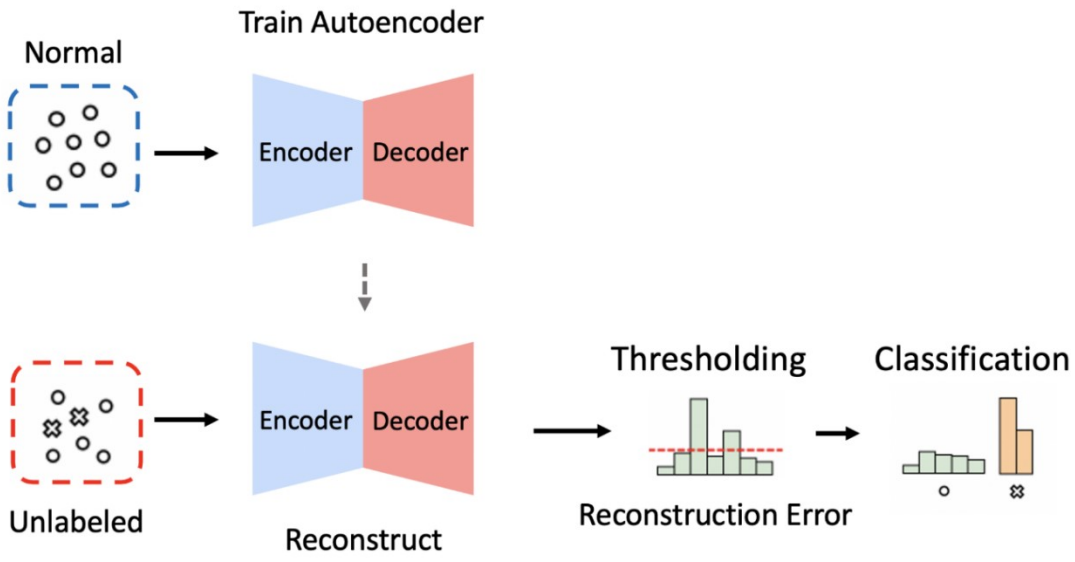


Рисунок 2.4 – Процес виявлення аномалій за допомогою автоенкодерів

Ключова перевага автоенкодерів полягає в їхній здатності моделювати складні нелінійні взаємозв'язки у високорозмірних даних. На відміну від традиційних статистичних методів, вони не покладаються на явні припущення щодо розподілу і можуть адаптуватися до широкого спектру типів та структур даних.

Ще однією важливою характеристикою автоенкодерів є їхня гнучкість. Регулюючи архітектуру та процедуру навчання, можна контролювати рівень абстракції у навченому представленні. Це дозволяє моделі фіксувати як глобальні закономірності, так і дрібні структури, залежно від вимог аналізу. Однак ефективність методів на основі автоенкодерів залежить від якості та репрезентативності навчальних даних. Якщо навчальний набір містить значну кількість аномалій, модель може навчитися відтворювати їх як нормальні закономірності.

Підсумовуючи, в автоматизованому додатку автоенкодери можна розглядати як розширений аналітичний компонент, що доповнює статистичні та деревоподібні методи. Їхнє включення дає змогу виявляти складні аномалії, які неможливо ідентифікувати за допомогою простіших технік.

2.7 Оцінювання загальної якості набору даних

2.7.1 Моделі оцінювання якості даних

Оцінка якості даних є одним із фінальних етапів у циклі аналізу даних. Вона спрямована на надання систематизованої та всебічної характеристики придатності набору даних для подальшого використання. На відміну від окремих перевірок окремих проблем, таких як відсутні значення або аномалії, загальна оцінка якості вимагає об’єднання численних вимірів у єдину аналітичну міру.

Моделі якості даних слугують формальними структурами, що визначають, як вимірюються, інтерпретуються та агрегуються різні аспекти даних. Зазвичай ці моделі розкладають якість даних на набір вимірів, кожен з яких відображає конкретну властивість набору даних. До загальновизнаних вимірів належать повнота, узгодженість, точність, унікальність та валідність. Кожен із цих вимірів охоплює окремий тип потенційної проблеми, а разом вони дають цілісне уявлення про надійність даних.

Ключова проблема при розробці таких моделей полягає в неоднорідності наборів даних та сфер застосування. Різні сценарії використання висувають різні вимоги до якості даних, і вимір, який є критичним в одному контексті, може бути менш релевантним в іншому. Як результат, сучасні моделі якості даних часто мають гнучку та розширювану структуру, що дозволяє обирати та зважувати виміри на основі конкретних аналітичних цілей.

З методологічної точки зору, кожен вимір зазвичай пов’язаний з набором вимірюваних показників, отриманих з набору даних. Ці показники обчислюються за допомогою статистичного аналізу, валідації на основі правил або методів машинного навчання, як обговорювалося в попередніх

розділах. Отримані значення забезпечують кількісну основу для оцінки окремих аспектів якості даних.

Говорячи про автоматизовані системи, моделі оцінки якості даних відповідають за організацію результатів. Структуруючи різноманітні метрики в цілісну систему, програма здатна перетворювати необроблені аналітичні результати на змістовні оцінки, що підтримують процес прийняття рішень. Цей структурований підхід також полегшує порівняння між наборами даних і забезпечує послідовний моніторинг якості з плином часу.

2.7.2 Стандарт ISO 25012 для оцінювання якості даних

Широко визнаною базовою моделлю для оцінки якості даних є стандарт ISO/IEC 25012, який пропонує комплексну модель оцінки якості даних у різних контекстах [18]. Цей стандарт визначає якість даних як багатовимірне поняття та розрізняє властивості даних та характеристики, що залежать від системи.

До властивостей даних належать такі атрибути, як точність, повнота, узгодженість та достовірність. З іншого боку, якість даних, що залежить від системи, стосується того, як дані управляються та використовуються в конкретному середовищі, охоплюючи такі характеристики, як доступність, переносимість та можливість відновлення.

Модель ISO 25012 визначає набір характеристик якості, які в сукупності описують загальний стан набору даних. Ці характеристики не призначені для оцінки окремо; навпаки, вони утворюють взаємопов’язану систему, в якій недоліки в одному вимірі можуть впливати на інші. Наприклад, неповні дані можуть призвести до зниження точності, тоді як невідповідності можуть підірвати достовірність. Цей стандарт лише запроваджує словник та концептуальну структуру, полегшує комунікацію

між зацікавленими сторонами та підтримує розробку сумісних систем оцінки якості даних.

Відповідно, у застосунку, що розробляється, варто використовувати стандарт ISO 25012, який має слугувати концептуальною основою для організації показників якості та їх узгодження з усталеними аспектами якості даних.

2.7.3 Формування інтегральної оцінки якості набору даних

Заключний етап оцінки якості даних передбачає об’єднання окремих показників якості в єдиний комплексний показник, який узагальнює загальний стан набору даних. Цей процес перетворює багатовимірну оцінку на компактне представлення, яке можна використовувати для порівняння, моніторингу та прийняття рішень.

Формування комплексного показника якості вимагає ретельного врахування кількох факторів. По-перше, окремі метрики необхідно нормалізувати для забезпечення порівнянності, оскільки вони можуть бути визначені за різними шкалами та представляти різні типи вимірювань. По-друге, кожному виміру якості необхідно присвоїти відповідні ваги, щоб відобразити їх відносну важливість у даному контексті.

Сам процес агрегації зазвичай передбачає об’єднання нормалізованих показників за допомогою зваженої суми або подібної функції. Хоча такий підхід забезпечує просте та зрозуміле рішення, він також створює потенційні обмеження. Зокрема, агрегація може приховати критичні проблеми, якщо високі показники в одних вимірах компенсують серйозні недоліки в інших. Тому важливо доповнити інтегральний бал детальним розбиттям на окремі компоненти.

Завдяки інтеграції різноманітних аналітичних методів у єдину систему оцінювання формування інтегрального показника якості є кульмінацією теоретичних підходів, розглянутих у цьому розділі. Воно

встановлює прямий зв’язок між окремими перевірками якості даних та комплексною оцінкою, тим самим створюючи міцну основу для практичної реалізації, описаної в наступному розділі.

2.8 Використання великих мовних моделей для інтерпретації результатів аналізу

2.8.1 Принципи роботи великих мовних моделей

Великі мовні моделі, LLM, є значним проривом у сфері обробки природної мови. Вони використовують архітектури глибокого навчання – насамперед трансформери та механізми самоуваги – для генерації логічного та контекстуально релевантного тексту на основі засвоєних ймовірнісних розподілів [19].

Однак розгортання найсучасніших LLM зазвичай вимагає потужної обчислювальної інфраструктури, зокрема графічних процесорів із високою пропускною здатністю, через величезну кількість параметрів та значні вимоги до пам’яті. Щоб подолати розрив між теоретичними можливостями моделей та практичними обмеженнями споживчого обладнання, у сучасних реалізаціях дедалі частіше застосовуються передові методи квантування [20].

Квантування математично знижує точність числових значень, що представляють ваги моделі – наприклад, стискаючи стандартні числа з плаваючою комою до 4-бітних представлень – без непропорційного погіршення основних лінгвістичних та міркувальних можливостей моделі.

Такий підхід дозволяє суттєво зменшити обсяг пам’яті, необхідний для зберігання моделі. Водночас це відкриває можливість використання LLM на локальних пристроях без залучення дорогих обчислювальних ресурсів (рисунок 2.5).

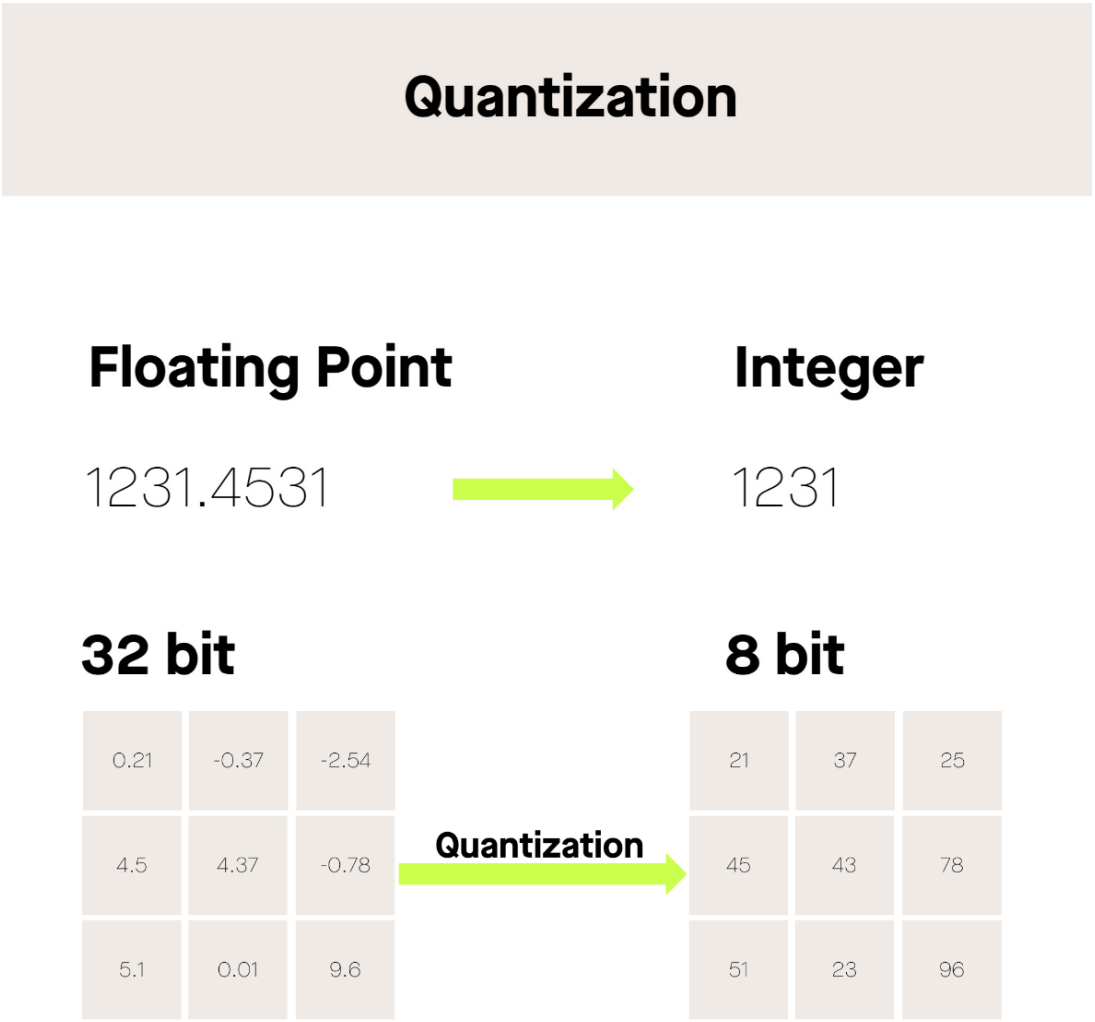


Рисунок 2.5 – Приклад квантизації у LLM

Такі формати, як GGUF, сприяють цьому агресивному стисненню, різко зменшуючи обсяг пам’яті з декількох гігабайтів до лише частки цього розміру. З теоретичної та архітектурної точки зору, ця оптимізація дозволяє розгортати складні генеративні моделі безпосередньо на центральних процесорах у локальних обчислювальних середовищах.

У контексті запропонованої системи автоматизованого дослідницького аналізу даних використання цих квантованих моделей гарантує, що розширена семантична інтерпретація може виконуватися повністю локально, задовольняючи суворі апаратні обмеження та зберігаючи операційну незалежність програмного забезпечення.

2.8.2 Можливості використання LLM у задачах аналізу даних

Головною метою EDA є перетворення необроблених, неструктурованих табличних даних на практичні висновки; проте результати роботи сучасних статистичних алгоритмів та детекторів аномалій на основі машинного навчання часто складаються з абстрактних числових показників, багатовимірних оцінок відхилень та складних параметрів розподілу, що в свою чергу, потребує втручання експерта для їх інтерпретації. Великі мовні моделі пропонують надзвичайно ефективний підхід до подолання цього «вузького місця», виступаючи в ролі інтелектуального семантичного шару між базовим обчислювальним механізмом та кінцевим користувачем.

Опрацьовуючи структуровані числові вхідні дані та результати алгоритмів, LLM можуть синтезувати розрізнені, високотехнічні дані в цілісні, зрозумілі для людини описи. Ця здатність дозволяє аналітичній системі автоматично генерувати вичерпні підсумки щодо готовності наборів даних, виявляти критичні структурні недоліки та надавати інтуїтивні пояснення природною мовою, чому конкретні дані позначаються, наприклад, як багатовимірні винятки або близькі дублікати.

Крім того, інтеграція LLM у робочі процеси аналізу даних порушує важливі питання щодо конфіденційності та безпеки даних. Набори даних часто містять конфіденційну, власницьку або особисту інформацію, що робить передачу таких даних стороннім хмарним інтерфейсам прикладного програмування дуже ризикованою і часто юридично недопустимою.

Використовуючи локально виконувані квантовані LLM, аналітична платформа гарантує, що вся обробка даних, статистичне вилучення та семантичне генерування тексту залишаються суворо в межах безпечного офлайн-середовища користувача. Такий локалізований підхід не тільки забезпечує відповідність сучасним стандартам захисту даних, але й

повністю усуває залежність від зовнішніх мережевих сервісів, що підвищує надійність аналітичної системи.

2.8.3 Формування текстових пояснень результатів аналізу

Автоматичне формування надійних текстових пояснень у рамках пайплайна з обробки даних вимагає ретельного проєктування промптів, щоб гарантувати, що мовна модель генерує фактично точні, детерміновано обмежені та надзвичайно релевантні відповіді. Без відповідних обмежень генеративні моделі за своєю природою схильні до «галюцинацій» – явищ, при яких модель вигадує неіснуючі статистичні показники або фабрикує елементи докладу, яких немає у вихідних даних [21].

Щоб зменшити цей системний ризик, промпти мають бути ретельно розроблені, щоб прив’язати вихідні дані моделі до точних числових характеристик, показників відхилення та статистичних контекстів, визначених під час попереднього алгоритмічного аналізу. Інструкційні підказки мають чітко наказувати моделям покладатися виключно на надані значення та суворо забороняти генерування сторонніх елементів форматування, таких як марковані списки, фрагменти коду або загальні приклади.

Крім того, запровадження суворих обмежень щодо токенів та структурних обмежень – таких як встановлення максимальної довжини у два речення та конкретного ліміту кількості слів – гарантує, що згенеровані пояснення залишаються лаконічними, чітко сфокусованими та стилістично відповідними для структурованих аналітичних звітів.

Поза межами генеративних обмежень теоретична основа просунутої аналітичної системи отримує значну користь від гібридної стратегії пояснення, яка поєднує ймовірнісний характер LLM із детермінованими логічними структурами.

Наприклад, у сценаріях, де базовий ансамбль машинного навчання надає достатні та однозначні діагностичні дані – такі як рівні шуму за класами або показники впевненості прогнозів у системах впевненого навчання – система може повністю оминути LLM і побудувати фактичні пояснення безпосередньо на основі внутрішніх метрик моделі.

Таким чином, мовна модель стратегічно використовується лише у сценаріях, що вимагають семантичної інтерпретації складних відхилень з багатьма ознаками, оптимізуючи тим самим обчислювальні витрати та максимізуючи цілісність кінцевого аналітичного результату.

2.9 Концепція рішення для автоматизованого аналізу якості даних

Проведений у цьому розділі аналіз теоретичних підходів до дослідницького аналізу даних, методів виявлення проблем якості та сучасних інструментів їх реалізації дає змогу сформулювати узагальнену концепцію інтелектуальної програмної системи автоматизованого аналізу якості табличних даних. На відміну від існуючих рішень, що переважно обмежуються описовою статистикою та базовим профілюванням, запропонований підхід передбачає інтеграцію статистичних методів, алгоритмів машинного навчання, моделей глибинного навчання та засобів семантичної інтерпретації в межах єдиного аналітичного середовища, що дозволяє розглядати аналіз якості даних як комплексну задачу штучного інтелекту.

Запропоноване рішення доцільно реалізувати у вигляді модульної багаторівневої архітектури, яка забезпечує розділення функціональності та масштабованість системи. Базовий рівень відповідає за уніфікований доступ до даних, їх завантаження, нормалізацію та первинне профілювання, включаючи аналіз структури, типів даних і базових статистичних характеристик. На наступному рівні реалізуються спеціалізовані інтелектуальні модулі виявлення проблем якості даних, зокрема підсистема

аналізу пропущених значень, модулі виявлення точних і нечітких дублікатів, компонент перевірки структурної узгодженості на основі функціональних залежностей, а також модулі виявлення аномалій, що поєднують класичні статистичні підходи з алгоритмами машинного та глибинного навчання, включаючи ансамблеві методи та автоенкодери для моделювання багатовимірних розподілів.

Окремим важливим компонентом системи має бути підсистема аналізу достовірності міток у задачах навчання з учителем, яка базується на ансамблевих підходах і ймовірнісному оцінюванні коректності цільових значень. Це дозволить вирішувати одну з найскладніших задач забезпечення якості даних – виявлення шуму міток, що традиційно не охоплюється більшістю існуючих інструментів автоматизованого аналізу.

Результати роботи всіх аналітичних модулів інтегруються в межах єдиного механізму оцінювання якості даних, побудованого відповідно до принципів стандарту ISO/IEC 25012. Такий підхід передбачає формалізацію характеристик якості даних у вигляді обчислюваних метрик та формування інтегрального показника якості, що дозволяє виконувати узагальнену, кількісно визначену та відтворювану оцінку наборів даних, а також забезпечує можливість їх порівняння в різних аналітичних сценаріях.

З метою підвищення інтерпретованості результатів аналізу запропоновано інтеграцію компонента на основі великих мовних моделей, який забезпечує автоматичну генерацію структурованих текстових пояснень, узагальнення виявлених проблем та формування рекомендацій щодо їх усунення. Це дозволяє трансформувати результати обчислювального аналізу у зрозумілу для користувача форму та реалізувати когнітивний рівень взаємодії з системою.

Узагальнюючи, запропонована система повинна функціонувати як єдиний автоматизований pipeline, що охоплює всі ключові етапи аналізу якості даних – від профілювання до формування інтерпретованих аналітичних звітів, забезпечуючи при цьому масштабованість,

розширюваність та мінімізацію ручного втручання. Така інтеграція методів різної природи дозволяє подолати обмеження існуючих підходів і забезпечити комплексний, глибокий та інтелектуально орієнтований аналіз якості табличних наборів даних.

Наукова новизна отриманих результатів полягає у формуванні цілісного підходу до автоматизованого аналізу якості даних як задачі штучного інтелекту, в межах якого поєднуються методи різної природи – статистичні, алгоритмічні та мовні. Пропонується покращити методи виявлення складних проблем якості даних за рахунок використання алгоритмів машинного та глибинного навчання, що дозволить виявляти багатовимірні аномалії, шум міток та приховані залежності у даних, які не визначаються традиційними статистичними підходами. Крім того, запропоновано інтеграцію великих мовних моделей як інтелектуального компонента інтерпретації результатів, що забезпечує автоматичне формування семантично узгоджених пояснень і рекомендацій. Сукупність цих рішень забезпечує перехід від фрагментованого описового аналізу до побудови інтелектуальної системи діагностики якості даних, здатної підтримувати прийняття рішень у процесах підготовки даних для задач машинного навчання.

3 РОЗРОБКА ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ЯКОСТІ ТАБЛИЧНИХ ДАНИХ

3.1 Організація експериментального дослідження

3.1.1 Мета та задачі практичних досліджень

Головною метою практичної частини цього дослідження є розробка і оцінка ефективності запропонованої системи автоматизованого EDA у виявленні та кількісній оцінці проблем якості даних у табличних наборах. Система розроблена для роботи у вигляді єдиного конвеєра, що об’єднує статистичний аналіз, діагностику на основі машинного навчання та формальні методики оцінки якості. Отже, практичне дослідження має на меті не лише продемонструвати функціональну коректність окремих компонентів, а й оцінити узгодженість та повноту системи як інтегрованої аналітичної платформи [22].

Центральним фокусом у дослідженні є порівняння автоматизованої оцінки якості даних з традиційними підходами до аналізу. Зокрема, дослідження зосереджується на визначенні того, чи здатна запропонована система систематично виявляти ключові категорії проблем з даними, включаючи відсутні значення, дублікати записів, аномалії, шум та порушення обмежень логічної узгодженості, без необхідності ручного втручання [23]. Ця мета безпосередньо стосується обмежень існуючих інструментів, які зазвичай надають описову статистику, але не мають глибших діагностичних можливостей, що відповідають робочим процесам машинного навчання.

Окрім коректності результатів, дослідження оцінює ефективність автоматизованого EDA з точки зору інтерпретованості та зручності використання. Це включає оцінку того, чи надають згенеровані результати, включаючи структуровані метрики та описові пояснення, достатнього

розуміння для фахівця, щоб зрозуміти стан набору даних та прийняти обґрунтовані рішення щодо попередньої обробки та розробки моделі. Особлива увага приділяється ролі автоматизованих пояснень, включаючи ті, що генеруються за допомогою інтегрованих мовних моделей, у покращенні інтерпретованості складних аналітичних результатів.

Також буде проведена оцінка адаптивності системи до різних сценаріїв даних. Застосунок розроблений для роботи в різних умовах, включаючи набори даних різного розміру, структури та рівня складності.

Нарешті, дослідження має на меті підтвердити придатність системи для інтеграції в реальні робочі процеси з обробки даних. Це передбачає вивчення її поведінки в різних режимах використання – як бібліотека Python, так і взаємодія через графічний інтерфейс користувача. Мета полягає в тому, щоб продемонструвати, що система може ефективно функціонувати в цих контекстах, зберігаючи при цьому послідовність аналітичних результатів та відтворюваність.

3.1.2 Характеристики використаних наборів даних

Експериментальна оцінка запропонованої системи проводиться з використанням декількох наборів даних, що відрізняються за структурою, розміром та передбачуваною метою аналізу. Така різноманітність є необхідною для того, щоб система була випробувана в широкому діапазоні реалістичних умов, а її поведінку можна було узагальнити.

Перша категорія наборів даних складається зі структурованих табличних даних, які зазвичай використовуються в задачах машинного навчання. Ці набори даних, як правило, містять поєднання числових та категоріальних ознак і можуть включати визначену цільову змінну. Типовим прикладом, використаним у цьому дослідженні, є набір даних «Titanic», який містить 891 запис та 12 ознак. Цей набір даних широко визнаний еталоном у машинному навчанні і, як відомо, містить кілька

властивих йому проблем якості даних, зокрема відсутні значення, категоріальні невідповідності тощо. Його включення дозволяє перевірити здатність системи виявляти добре відомі проблеми.

Для оцінки масштабованості та характеристик продуктивності до дослідження включено великий набір даних «Zomato» обсягом приблизно 80 мегабайтів. Цей набір даних слугує стрес-тестом для конвеєра попередньої обробки та аналізу системи, зокрема стосовно конструкції з двома двигунами, яка динамічно обирає між Pandas та Polars залежно від розміру даних. Використання великого набору даних дозволяє оцінити ефективність використання пам'яті, час виконання та ефективність реалізованих стратегій оптимізації, таких як вибірка, адаптивні пороги та обмеження на рівні компонентів.

Також буде використано мінімальний набір даних, що містить лише дві ознаки, який використовується для вивчення поведінки системи у нестандартних умовах. Такі набори даних важливі для виявлення потенційних вузьких місць в компонентах, що базуються на багатовимірних взаємозв’язках, таких як виявлення викидів та моделювання розподілу. Включаючи цей сценарій, дослідження оцінює, чи може система коректно обробляти обмежені простори ознак, не спотворюючи фінальний звіт.

Отже, вибір наборів даних не є довільним, а чітко узгоджений із цілями оцінки дослідження. Кожен набір даних спрямований на конкретний аспект продуктивності системи, включаючи правильність аналізу, стійкість до характеристик даних та масштабованість за умови збільшеного обчислювального навантаження. Цей структурований підхід забезпечує всебічну перевірку системи.

3.1.3 Сценарії використання системи

Розроблена система призначена для підтримки декількох режимів роботи, кожен з яких відповідає окремому сценарію використання. Тому

практичне дослідження включає оцінку цих сценаріїв, щоб продемонструвати гнучкість та застосовність системи в різних контекстах.

Один із основних сценаріїв передбачає базовий дослідницький аналіз без використання цільової змінної або розширеної діагностики на основі машинного навчання. У цьому режимі система виконує структурний аналіз, виявлення пропущених значень, ідентифікацію дублікатів та інші базові перевірки. Цей сценарій відображає початковий етап розуміння даних, де метою є отримання загального уявлення про якість та структуру набору даних [24].

Більш просунутий сценарій передбачає використання визначеної цільової змінної, що дозволяє активувати компоненти, орієнтовані на машинне навчання, такі як виявлення шуму міток. У цьому контексті система використовує методи контрольованого навчання та ймовірнісне моделювання для виявлення невідповідностей між шаблонами ознак та призначеними значеннями цільової функції.

Інший важливий сценарій використання передбачає обробку великомасштабних наборів даних. У цьому випадку система автоматично адаптує свою внутрішню поведінку, вибираючи відповідний механізм обробки даних та застосовуючи стратегії оптимізації, такі як підвибірка та обчислювальні обмеження. Це демонструє здатність системи підтримувати аналітичну функціональність під час роботи в умовах обмежених ресурсів, що є критично важливим для практичного розгортання.

Окрім аналітичних сценаріїв, система оцінюється в різних режимах розповсюдження та взаємодії. Як бібліотека Python, вона надає програмний інтерфейс, який можна інтегрувати в існуючі конвеєри даних та автоматизовані робочі процеси. Як десктоп-застосунок, реалізований за допомогою графічного інтерфейсу користувача, вона підтримує інтерактивне використання фахівцями, яким може не знадобитися безпосередня взаємодія на рівні коду.

3.2 Архітектура застосунку та pipeline обробки даних

3.2.1 Загальна архітектура системи

Розроблена система побудована у вигляді багаторівневого конвеєра, призначеного для забезпечення чіткого розмежування функцій, розширюваності та контрольованого потоку даних протягом усього процесу аналізу. Архітектурний дизайн відповідає п’ятирівневій моделі, де кожен рівень інкапсулює конкретну функцію в рамках загального життєвого циклу виконання аналізу (рисунок 3.1).



Рисунок 3.1 – Архітектура системи AEDA

На найвищому рівні користувацького інтерфейсу надаються кілька точок входу для взаємодії з системою, включаючи програмний API та графічний інтерфейс для настільних систем [25]. Цей рівень навмисно відокремлений від внутрішньої логіки системи, що дозволяє повторно використовувати той самий аналітичний конвеєр у різних режимах роботи без змін основного коду.

Під інтерфейсом рівень оркестрування відповідає за координацію виконання аналітичного конвеєра. Ця функціональність реалізована через центральний клас аналізатора, який виступає фасадом для всієї системи. Він ініціалізує контекст аналізу, налаштовує параметри виконання та забезпечує детермінований порядок виконання компонентів. Рівень оркестрування гарантує, що всі компоненти працюють у спільному середовищі, зберігаючи суворий контроль над залежностями та потоком виконання.

Основна аналітична логіка знаходиться на рівні компонентів, який складається з набору модульних одиниць аналізу. Кожен компонент відповідає за виявлення певного класу проблем якості даних або обчислення конкретного набору метрик. Модульна структура дозволяє незалежно розробляти та тестувати компоненти, підтримуючи їх інтеграцію в єдиний конвеєр. Важливо, що компоненти не взаємодіють безпосередньо між собою, що запобігає тісному зв’язку та спрощує підтримку системи.

Рівень компонентів спирається на рівень сервісів, який надає спільну інфраструктуру, необхідну для декількох компонентів. Сюди входять механізми попередньої обробки даних, такі як завантаження та трансформація наборів даних, а також допоміжні сервіси, наприклад, інтерфейс мовної моделі, що використовується для генерації пояснень природною мовою. Завдяки централізації цих функцій система уникає дублювання логіки та забезпечує узгодженість між компонентами [26].

Нарешті, вихідний рівень відповідає за перетворення результатів аналізу у формат, придатний для використання користувачем. Це включає генерацію повністю самостійного HTML-звіту, збагаченого візуалізаціями

та пояснювальним контентом. Вихідний рівень агрегує структуровані результати з усіх компонентів і представляє їх у послідовній та зрозумілій формі, завершуючи конвеєр.

Ця багаторівнева архітектура забезпечує чітке розділення між введенням даних, аналізом, інтерпретацією та презентацією. В результаті система досягає високого рівня модульності та розширюваності, що дозволяє впроваджувати нові аналітичні компоненти або формати виводу без порушення існуючої функціональності.

3.2.2 Організація виконання аналізу

Виконання процесу аналізу централізовано керується оркестратором, який об’єднує весь робочий процес в єдину точку входу. Ця архітектура відповідає паттерну «фасад», забезпечуючи спрощений інтерфейс для запуску складної багатоетапної обробки та приховуючи від користувача деталі внутрішньої реалізації.

Аналіз починається з ініціалізації об’єкта конфігурації, який визначає параметри виконання. Ця конфігурація дозволяє здійснювати детальний контроль над поведінкою системи, включаючи ввімкнення або вимкнення конкретних компонентів та налаштування параметрів аналізу. Наявність централізованого механізму конфігурації гарантує, що виконання є гнучким і відтворюваним, оскільки одна й та сама конфігурація може бути використана повторно для отримання ідентичних результатів.

Після встановлення конфігурації система створює контекст аналізу, який слугує спільним станом у всьому конвеєрі. Цей контекст інкапсулює набір даних і забезпечує зберігання проміжних артефактів та вихідних даних компонентів. Усі подальші операції виконуються в цьому контексті, що гарантує узгодженість та простежуваність результатів [27].

Потім оркестратор приступає до виконання компонентів аналізу у заздалегідь визначеній послідовності (рисунок 3.2).

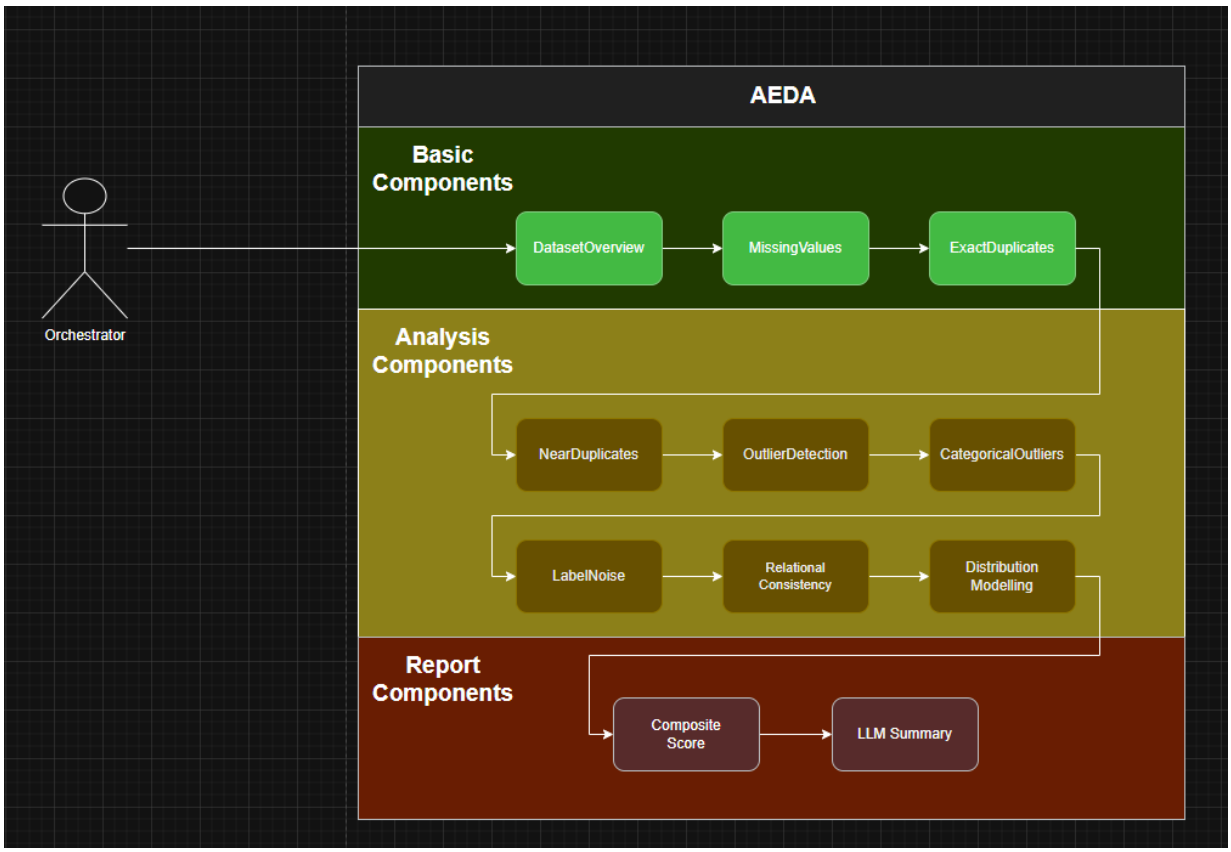


Рисунок 3.2 – Послідовність виконання компонент у AEDA

Ця послідовність не є довільною, а визначається логічними залежностями між компонентами. Наприклад, компоненти, що потребують знання типів даних або категоризації ознак, покладаються на попередній аналіз, виконаний раніше в конвеєрі. Завдяки дотриманню фіксованого порядку виконання система гарантує, що всі необхідні вхідні дані будуть доступні в потрібний момент, усуваючи ризик помилок залежності під час виконання.

Кожен компонент виконує свій аналіз незалежно, отримуючи доступ до набору даних та будь-яких раніше обчислених артефактів із спільного контексту. Після завершення компонент зберігає свої результати та публікує відповідні артефакти для використання на наступних етапах [28]. Такий підхід гарантує, що обчислення виконуються лише один раз для кожного компонента, а результати можуть ефективно повторно використовуватися на наступних етапах конвеєра.

Важливим аспектом моделі виконання є її детермінізм. Система дотримується фіксованого порядку операцій і використовує контрольовану випадковість там, де це необхідно, наприклад, при вибірці або ініціалізації моделі. Це гарантує, що повторні виконання з однаковою конфігурацією та набором даних дають ідентичні результати, що є важливим для експериментальної валідації та відтворюваності.

3.2.3 Механізм взаємодії компонентів

Ключовою архітектурною особливістю системи є механізм взаємодії між компонентами аналізу. Замість прямого зв’язку або тісно пов’язаних інтерфейсів система використовує модель спільного сховища артефактів, яка забезпечує опосередкований обмін даними у контрольований та розширюваний спосіб.

Спільне сховище артефактів реалізовано як централізований словник у контексті аналізу. Кожен компонент може публікувати проміжні результати у вигляді іменованих артефактів, які потім стають доступними для всіх наступних компонентів у конвеєрі. Ці артефакти можуть включати похідні метадані, статистичні зведення, результати моделювання або проміжні обчислення, що мають значення для подальшого аналізу.

Така конструкція надає кілька важливих переваг. По-перше, вона усуває необхідність явних залежностей між компонентами, дозволяючи кожному компоненту залишатися автономним. Компонентам не потрібно знати внутрішню структуру або деталі реалізації інших компонентів; вони покладаються лише на наявність конкретних ключів артефактів та спільний інтерфейс. Це значно зменшує зв’язаність і покращує зручність обслуговування.

По-друге, підхід на основі артефактів дозволяє ефективно повторно використовувати обчислені результати. Наприклад, коли компонент ідентифікує числові та категоріальні ознаки, ця інформація може бути

повторно використана кількома наступними компонентами без повторного обчислення. Це не тільки покращує продуктивність, але й забезпечує узгодженість між аналізами.

По-третє, механізм підтримує розширюваність. Нові компоненти можна вводити в конвеєр із мінімальними зусиллями на інтеграцію, за умови дотримання встановлених конвенцій щодо публікації та споживання артефактів. Це відповідає принципу «open/closed», що дозволяє системі розвиватися без необхідності модифікації існуючих компонентів.

Окрім спільних артефактів, система підтримує окрему структуру для зберігання узагальнених результатів кожного компонента. Ці узагальнення накопичуються протягом виконання і згодом використовуються для генерації звітів та інтерпретації на високому рівні. Таке розділення між необробленими артефактами та зведеними результатами гарантує, що як детальна, так і агрегована інформація зберігаються без строгої зв’язності.

Загалом, механізм взаємодії компонентів є архітектурним рішенням, що надає пріоритет модульності, масштабованості та чіткості потоку даних. Завдяки роз'єднанню компонентів та забезпеченню контрольованого обміну інформацією система досягає балансу між незалежністю та інтеграцією, що є необхідним для складних аналітичних конвеєрів.

3.2.4 Система конфігурації та відтворюваність

Система містить централізований механізм конфігурації, який регулює всі аспекти виконання аналізу – від поведінки компонентів до алгоритмічних порогових значень. Ця конфігурація реалізована за допомогою поєднання об’єкта конфігурації, доступного для користувача, та набору внутрішньо визначених констант, що визначають параметри за замовчуванням і налаштування.

Використання спеціального об’єкта конфігурації дозволяє користувачам налаштовувати процес аналізу без зміни базового коду. Такі

параметри, як активація компонентів, специфікація цільових змінних та додаткові функції, можуть бути визначені декларативно, що забезпечує гнучкість, зберігаючи при цьому чітке розділення між конфігурацією та логікою. Цей підхід є особливо важливим в експериментальних умовах, де можуть тестуватися та порівнюватися численні конфігурації.

На глибшому рівні система спирається на набір попередньо визначених констант, що керують поведінкою алгоритмів. Ці константи включають порогові значення для виявлення винятків, обмеження вибірки, гіперпараметри моделі та часові обмеження для обчислювально інтенсивних операцій. Централізуючи ці значення, система забезпечує узгодженість між компонентами та спрощує процес налаштування й оптимізації.

Відтворюваність є фундаментальною вимогою до проєктування системи і досягається за допомогою декількох скоординованих механізмів. По-перше, всі джерела випадковості, такі як процедури вибірки та ініціалізація моделей машинного навчання, контролюються за допомогою фіксованих випадкових значень. Це гарантує, що стохастичні процеси дають однакові результати при багаторазовому виконанні.

По-друге, детермінований порядок виконання компонентів гарантує, що послідовність операцій залишається послідовною.

По-третє, підхід, керований конфігурацією, дозволяє фіксувати та повторно використовувати точні умови аналізу. Зберігаючи параметри конфігурації разом із набором даних, ми можемо точно відтворити середовище аналізу, що є критично важливим для валідації, порівняння та дебагінгу.

Поєднання централізованої конфігурації, контрольованої випадковості та детермінованого виконання робить відтворюваність невід’ємною властивістю системи, а не додатковим елементом. Це особливо важливо в контексті оцінки якості даних, де послідовні результати необхідні для надійного прийняття рішень та наукової строгості.

3.3 Дослідження шару обробки даних

3.3.1 Реалізація dual-engine підходу

Одним із фундаментальних проектних рішень у розробленій системі є впровадження підходу до обробки даних на основі двох рушіїв, що дозволяє динамічно обирати між різними методами обробки даних залежно від характеристик набору даних. Цей механізм запроваджено для усунення неминучого компромісу між простотою інтеграції та обчислювальною продуктивністю, який виникає під час роботи з табличними даними в робочих процесах з аналізу даних на базі Python.

Система підтримує два популярні двигуни обробки даних, а саме Pandas і Polars, кожен із яких має свої переваги. Pandas слугує стандартною та найпоширенішою бібліотекою для обробки табличних даних, забезпечуючи широку сумісність із загальною екосистемою Python, включаючи бібліотеки машинного навчання та статистики. Натомість Polars розроблено для високопродуктивної обробки даних, використовуючи columnar модель пам’яті на базі Apache Arrow та надаючи можливості багатопотокового виконання [29]. Хоча Polars пропонує значні переваги у продуктивності для великих наборів даних, його інтеграція з подальшими аналітичними інструментами залишається менш досконалою порівняно з Pandas [30].

Щоб узгодити ці відмінності, система впроваджує рівень абстракції у вигляді спеціального класу набору даних. Цей клас інкапсулює базове представлення даних і надає уніфікований інтерфейс для доступу до набору даних незалежно від обраного двигуна. Вибір двигуна обробки здійснюється автоматично під час ініціалізації набору даних на основі розміру вхідних даних. Зокрема, коли набір даних перевищує заздалегідь визначений поріг, система обирає Polars, щоб скористатися його характеристиками продуктивності, тоді як менші набори даних

обробляються за допомогою Pandas для забезпечення максимальної сумісності та простоти.

Цей механізм адаптивного вибору доповнюється можливістю перевизначення, що дозволяє користувачам явно вказати бажаний двигун. Така гнучкість важлива в сценаріях, де специфічні для домену міркування або вимоги до налагодження вимагають ручного контролю над бекендом обробки. Система підтримує примусовий вибір двигуна, гарантуючи, що в разі потреби перевагу над автоматизованим рішенням матимуть уподобання користувача.

З точки зору реалізації абстракція набору даних надає єдину точку доступу для аналітичних компонентів через стандартизований метод, яка повертає набір даних у форматі, сумісному з Pandas. Коли Polars використовується як базовий двигун, ця властивість виконує прозоре перетворення у Pandas DataFrame. Ця конструкція гарантує, що всі аналітичні компоненти працюють на єдиній структурі даних, усуваючи необхідність підтримувати окремі реалізації для різних бекендів. Як результат, складність, пов’язана з підтримкою декількох двигунів, обмежується рівнем попередньої обробки, зберігаючи простоту та зручність обслуговування на рівні компонентів.

Таким чином, підхід з двома двигунами є практичним компромісом між оптимізацією продуктивності та сумісністю з екосистемою. Завдяки абстракції базового представлення даних та автоматизації вибору двигуна система досягає адаптивності до різних обсягів даних, зберігаючи при цьому єдиний інтерфейс для подальшої обробки.

3.3.2 Порівняльний аналіз продуктивності рушіїв

Впровадження архітектури з двома двигунами обумовлено істотними відмінностями в характеристиках продуктивності між Pandas і Polars, особливо під час обробки великих наборів даних. Pandas працює з рядковою

організацією пам’яті та виконує операції переважно в однопотоковому режимі, що обмежує його масштабованість із збільшенням розміру набору даних. Незважаючи на високу гнучкість та хорошу інтеграцію з екосистемою Python, Pandas може стати вузьким місцем у робочих процесах, що вимагають інтенсивного використання даних, особливо під час виконання складних перетворень або агрегацій.

На відміну від цього, Polars побудовано на стовпчастій моделі пам'яті та використовує векторизовані операції у поєднанні з багатопотоковим виконанням. Така конструкція дозволяє йому ефективно обробляти великі набори даних, мінімізуючи навантаження на пам’ять та максимізуючи використання процесора. Емпіричні спостереження показують, що Polars може досягти підвищення продуктивності від декількох разів і швидше порівняно з Pandas, особливо для таких операцій, як фільтрування та агрегація над великими наборами даних.

Система використовує ці характеристики, делегуючи завдання початкового завантаження даних та попередньої обробки Polars, коли розмір набору даних перевищує визначений поріг. Це включає розбір файлів та базові перетворення, які значно виграють від оптимізованої моделі виконання Polars. Однак, оскільки багато компонентів аналітики, що працюють на нижньому рівні, покладаються на бібліотеки, які очікують DataFrames від Pandas або масиви NumPy, перед початком аналізу вводиться етап перетворення. Хоча це перетворення спричиняє додаткові накладні витрати, воно виконується лише один раз і компенсується приростом продуктивності, досягнутим під час початкової фази обробки даних.

Важливим аспектом у цій конструкції є ефективність використання пам’яті. Колонкове представлення, яке використовує Polars, зменшує фрагментацію пам’яті та забезпечує ефективніше зберігання однорідних типів даних. Це особливо вигідно для широких наборів даних із великою кількістю стовпців, де Pandas може спричиняти значні накладні витрати через свої внутрішні структури даних. Як результат, використання Polars не

лише покращує швидкість виконання, а й підвищує здатність системи обробляти більші набори даних в умовах обмеженої пам'яті.

Незважаючи на ці переваги, система свідомо уникає використання Polars як єдиного механізму обробки. Основна причина такого рішення полягає в необхідності сумісності із зовнішніми бібліотеками, що використовуються на пізніших етапах конвеєра, зокрема з фреймворками машинного навчання та іншими інструментами. Зберігаючи Pandas як загальний інтерфейс для всіх аналітичних компонентів, система забезпечує безперебійну інтеграцію з цими бібліотеками, водночас отримуючи переваги від оптимізації продуктивності, яку надає Polars під час попередньої обробки.

Отже, порівняльний аналіз двох рушіїв підкреслює важливість вибору відповідної стратегії обробки даних на основі характеристик набору даних. Конструкція з двома двигунами дозволяє системі збалансувати продуктивність і сумісність, забезпечуючи ефективну роботу в широкому діапазоні сценаріїв обробки даних.

3.3.3 Уніфікація доступу до даних

Для забезпечення узгодженого аналізу у всіх компонентах система реалізує уніфікований механізм доступу до даних, який абстрагує базові деталі зберігання та обробки даних. Цей механізм реалізовано за допомогою рівня абстракції наборів даних, який надає стандартизований інтерфейс для отримання даних та взаємодії з ними.

В основі цього підходу лежить використання єдиної властивості, яка надає доступ до набору даних у форматі, сумісному з Pandas. Незалежно від того, чи дані були спочатку завантажені за допомогою Pandas чи Polars, усі аналітичні компоненти взаємодіють з набором даних через цей уніфікований інтерфейс. Такий вибір архітектури усуває необхідність у умовній логіці всередині компонентів і гарантує, що всі аналізи

виконуються на основі єдиної структури даних. Як результат, складність обробки різних представлень даних ізольована в шарі попередньої обробки, що дозволяє решті системи залишатися незалежною від базового механізму.

Окрім доступу до даних, система виконує автоматичне виведення схеми для категоризації стовпців на основі їхніх типів даних. Цей процес розрізняє числові, категоріальні та часові ознаки за допомогою механізмів перевірки типів, що надаються базовими бібліотеками обробки даних. Виведена схема відіграє вирішальну роль у керуванні поведінкою компонентів глибшого рівня, оскільки до різних категорій ознак застосовуються різні типи аналізу.

Виведені групування стовпців використовуються не тільки локально в шарі попередньої обробки, але й поширюються на інші компоненти через механізм спільних артефактів. Наприклад, компоненти, відповідальні за виявлення винятків, покладаються на ідентифікацію числових ознак, тоді як виявлення категоріальних аномалій залежить від набору категоріальних стовпців.

Іншим важливим аспектом уніфікованого рівня доступу до даних є його підтримка декількох форматів вхідних даних. Система здатна приймати набори даних з різних типів файлів, включаючи CSV, Parquet, JSON, ORC та XLSX. Коли певні формати не підтримуються обраним механізмом обробки на рівні ядра, система застосовує стратегії резервного варіанту для забезпечення успішного завантаження даних. Наприклад, формати, які не обробляються безпосередньо Polars, спочатку завантажуються за допомогою Pandas, а потім перетворюються за потреби. Такий підхід гарантує надійність завантаження даних, зберігаючи при цьому гнучкість у роботі з різноманітними джерелами даних.

Поєднання уніфікованого доступу до даних, автоматичного визначення схеми та гнучкої підтримки форматів утворює цілісний рівень попередньої обробки, який лежить в основі всього аналітичного конвеєра. Завдяки стандартизації способу представлення даних та доступу до них

система гарантує, що всі наступні компоненти працюють на основі узгоджених припущень, тим самим підвищуючи надійність та спрощуючи загальну архітектуру.

3.4 Базові компоненти аналізу

3.4.1 Компонент аналізу структури даних

Початковий етап аналітичного конвеєра має давати розуміння структури набору даних, що реалізується за допомогою компонента огляду набору даних. Цей компонент слугує фундаментальною основою для всіх наступних етапів аналізу, оскільки він витягує та формалізує основні метадані, необхідні для компонентів, що працюють на наступних етапах.

Основною відповідальністю цього компонента є ідентифікація та класифікація типів ознак. За допомогою вбудованих механізмів визначення типів кожен стовпець класифікується як числовий, категоріальний або datetime. Ця класифікація не є тільки описовою, а й напряму впливає на виконання компонентів, що знаходяться далі по конвеєру, таких як виявлення винятків та аналіз категоріальних аномалій. Щоб забезпечити цю взаємодію, компонент публікує ідентифіковані групи ознак у спільному сховищі артефактів, роблячи їх доступними по всьому конвеєру. Цей механізм гарантує, що всі компоненти покладаються на єдине, узгоджене визначення схеми.

Виконується додатковий рівень аналізу для оцінки кардинальності стовпців. Для кожної ознаки обчислюється кількість унікальних значень та відповідний коефіцієнт унікальності. На основі цих показників система ідентифікує потенційні ідентифікатори, які визначаються як ознаки з майже унікальними значеннями у всьому наборі даних. Реалізація поєднує як відносні, так і абсолютні порогові значення, щоб уникнути помилок у невеликих наборах даних, що гарантує виявлення лише значущих

ідентифікаторів. Компонент також обчислює використання пам’яті як на рівні стовпців, так і на рівні набору даних.

Для числових ознак компонент обчислює розширений набір описової статистики, включаючи розподіли на додачу до стандартних показників, таких як середнє значення та стандартне відхилення.

Загалом, компонент огляду набору даних створює структуроване представлення набору даних, яке повторно використовується у всьому конвеєрі. Його реалізація відображає принцип проєктування, що полягає у одноразовому обчисленні спільних метаданих та їх розподілі по системі, тим самим підвищуючи ефективність та забезпечуючи узгодженість аналізу.

3.4.2 Компонент аналізу пропущених значень

Аналіз пропущених значень реалізовано у вигляді спеціалізованого компонента, який оцінює повноту кожної ознаки в наборі даних. Цей компонент працює безпосередньо з уніфікованим представленням набору даних і застосовує векторний підхід для виявлення відсутніх записів, забезпечуючи як обчислювальну ефективність, так і коректність для різних типів даних.

В основі реалізації лежить використання єдиного механізму виявлення відсутніх значень, який послідовно обробляє порожні значення незалежно від їх базового представлення. Компонент обчислює кількість відсутніх значень для кожного стовпця та виводить відповідний коефіцієнт відносно загальної кількості рядків. Цей коефіцієнт слугує основним показником для оцінки повноти даних і використовується для виявлення ознак, які можуть потребувати попередньої обробки або «імпутації».

Реалізація також безперешкодно інтегрується з представницьким рівнем, надаючи як структуровані метрики, так і зрозумілі для людини зведення. Це подвійне представлення дозволяє використовувати результати програмно, залишаючи їх доступними для користувачів через згенерований

звіт. Замість того, щоб показувати користувачеві повні дані у фінальному зведенні, компонент виводить лише найбільш критичні випадки, які можуть більше за інші вплинути на якість набору даних.

Отже, компонент аналізу відсутніх значень забезпечує точний та ефективний механізм оцінки повноти даних. Його конструкція поєднує обчислювальну простоту з практичною значущістю, гарантуючи, що проблеми з відсутніми даними чітко ідентифікуються в рамках ширшого аналітичного робочого процесу.

3.4.3 Компонент виявлення точних дублікатів

Виявлення точних дублікатів записів реалізовано як окремий компонент, який ідентифікує рядки з ідентичними значеннями за всіма ознаками. Цей тип надмірності є поширеною проблемою в реальних наборах даних і може суттєво спотворити результати статистичного аналізу та машинного навчання, якщо її не вирішити належним чином.

Реалізація базується на оптимізованому механізмі виявлення дублікатів, який позначає всі випадки дублювання рядків. Такий підхід гарантує, що буде виявлено весь обсяг дублювання, включаючи всі екземпляри в кожній групі дублікатів. Для забезпечення ефективного групування та підрахунку компонент застосовує стратегію хешування, яка генерує унікальний ідентифікатор для кожного рядка на основі його вмісту. Це дозволяє системі об’єднувати ідентичні рядки та обчислювати кількість груп дублікатів без виконання ресурсомістких парних порівнянь.

Ключовим аспектом компонента є кількісна оцінка дублювання за допомогою нормалізованого коефіцієнта, що відображає частку рядків, які належать до груп дублікатів. Цей показник забезпечує чітку та зрозумілу оцінку надмірності в наборі даних. Для полегшення практичної інтерпретації реалізація класифікує цей коефіцієнт за дискретними рівнями ризику на основі заздалегідь визначених порогових значень. Ці порогові

значення обрано так, щоб відобразити потенційний вплив дублювання на подальші завдання, зокрема в контексті машинного навчання.

Компонент явно враховує наслідки дублювання даних для навчання та оцінки моделі. Дубльовані записи можуть штучно завищувати видиму частоту певних подій, що призводить до упередженої поведінки моделі, тому компонент виходить за межі простого виявлення та надає змістовний контекст для виявлених проблем.

Реалізація точного виявлення дублікатів демонструє баланс між обчислювальною ефективністю та глибиною аналізу. Поєднуючи ефективні алгоритми з інтерпретацією, що відповідає домену, компонент сприяє більш всебічному розумінню якості даних у системі.

3.5 Компоненти виявлення аномалій та якості даних

3.5.1 Виявлення подібних записів

Виявлення близьких дублікатів реалізовано у вигляді спеціалізованого компонента, призначеного для ідентифікації семантично схожих спостережень, які неможливо виявити за допомогою точного зіставлення. На відміну від точних дублікатів, такі записи зазвичай виникають через невідповідності при введенні даних, варіації форматування або часткове дублювання під час процесів інтеграції даних. Їхня наявність спричиняє незначні похибки у розподілі даних і може призвести до надмірної представленості певних закономірностей у моделях машинного навчання.

Реалізація базується на техніці хешування з урахуванням локальності MinHash, яка дозволяє ефективно оцінювати схожість між багатовимірними записами без необхідності вичерпних парних порівнянь. Кожен рядок набору даних перетворюється на набір токенів за допомогою структурованої стратегії токенізації, що включає як n-грами на рівні

символів, так і префікси, специфічні для стовпців. Така конструкція гарантує, що ідентичні значення в різних стовпцях не будуть помилково розглядатися як схожі, зберігаючи семантичну структуру даних [31].

Процедура MinHash генерує компактні сигнатури для кожного запису, які згодом індексуються. Цей підхід значно зменшує простір пошуку шляхом групування записів у «кошики» кандидатів, де для подальшого порівняння розглядаються лише ті, що мають схожі сегменти сигнатур. Ймовірнісний характер цього методу гарантує, що записи з високим ступенем схожості зберігаються з високою ймовірністю, тоді як несхожі пари ефективно відкидаються.

Потім пари-кандидати проходять етап верифікації, під час якого обчислюється уточнений показник схожості. Цей показник обчислюється як зважена сукупність схожостей за стовпцями з урахуванням як числових, так і категоріальних ознак. Числові значення порівнюються за допомогою нормалізованих різниць, тоді як категоріальні значення оцінюються за допомогою мір схожості на основі токенів. Цей гібридний підхід дозволяє компоненту обробляти неоднорідні набори даних.

Для впорядкування виявлених взаємозв’язків реалізація застосовує механізм кластеризації на основі структури даних роз’єднаних множин, групуючи всі транзитивно пов’язані пари в кластери записів, що майже дублюються. Це забезпечує більш змістовне представлення шаблонів дублювання порівняно з ізольованими парними збігами.

З огляду на обчислювальну складність аналізу на основі схожості, було впроваджено кілька стратегій оптимізації. До них належать обмеження кількості оброблених рядків за допомогою вибірки, обмеження кількості пар-кандидатів та дотримання меж часу виконання. Ці заходи гарантують, що компонент залишається керованим навіть для досить великих наборів даних.

Результат роботи цього компонента включає кількісні показники, що описують поширеність близьких дублікатів, а також репрезентативні

приклади, які ілюструють виявлені подібності. Ці результати передаються до наступних компонентів.

3.5.2 Виявлення аномалій у числових даних

Виявлення аномалій у числових даних здійснюється за допомогою модельного підходу, що враховує багатовимірні взаємозв’язки між ознаками. Цей компонент усуває обмеження традиційних одновимірних методів, виявляючи спостереження, які відхиляються від сукупного розподілу набору даних, а не від розподілів окремих ознак.

Реалізація базується на алгоритмі Isolation Forest, який будує ансамбль випадкових дерев рішень для ізоляції окремих точок даних. Основна ідея цього методу полягає в тому, що аномальні спостереження легше відокремити від решти даних, а отже, вони вимагають менше кроків розділення. Ця властивість відображається у довжині шляху кожного спостереження в ансамблі, яка згодом перетворюється на оцінку аномалії [32].

Перед навчанням моделі компонент виконує етапи попередньої обробки, що включають вибір релевантних числових ознак, імпутацію відсутніх значень та стандартизацію шкал ознак. Стовпці з високою кардинальністю, що нагадують ідентифікатори, виключаються з аналізу, щоб запобігти спотворенню поведінки моделі. Ці рішення щодо попередньої обробки є критичними для забезпечення того, щоб процес виявлення аномалій зосереджувався на значущих варіаціях у даних.

Для підвищення інтерпретованості компонент інтегрує механізм атрибуції ознак, який пояснює внесок окремих ознак у кожну виявлену аномалію [33]. Це досягається завдяки застосуванню специфічної для моделі техніки пояснення, яка розкладає оцінку аномалії на внески на рівні ознак. У випадках, коли цей метод не можна застосувати через

обчислювальні або структурні обмеження, використовується запасна стратегія, заснована на відхиленні від середніх значень ознак.

Результат надає як сукупні показники, такі як частка виявлених винятків, так і детальні приклади, що висвітлюють найважливіші аномалії.

3.5.3 Виявлення аномалій у категоріальних даних

Аномалії в категоріальних даних виявляються за допомогою механізму на основі частоти, який ідентифікує рідкісні або несподівані значення в межах дискретних областей ознак. На відміну від числових винятків, які визначаються через відстань або відхилення, категоріальні аномалії характеризуються низькою частотою появи порівняно із загальним розподілом.

Реалізація обчислює частотні розподіли для кожної категоріальної ознаки та виявляє значення, що не досягають порогу. Цей поріг динамічно коригується залежно від розміру набору даних, що гарантує, що процес виявлення залишається ефективним як для малих, так і для великих наборів даних. Враховуючи як абсолютні, так і відносні фактори, система уникає надмірної чутливості в малих наборах даних і зберігає здатність виявляти справді рідкісні події у великих наборах.

Результат роботи компонента включає як зведені статистичні дані, що описують поширеність рідкісних значень, так і ілюстративні приклади, які надають контекст для інтерпретації.

3.5.4 Виявлення шуму у мітках

Виявлення неправильно розмічених екземплярів у наборах даних для навчання з учителем реалізовано за допомогою принципової ймовірнісної моделі, яка розрізняє властиву даним складність та фактичні помилки

маркування. Цей компонент спеціально розроблено тільки для сценаріїв, де присутня цільова змінна.

Реалізація відбувається за багатоетапним процесом, який починається з оцінки ймовірностей класів за допомогою ансамблю моделей машинного навчання. Щоб забезпечити об’єктивність прогнозів, система використовує стратегію перехресної валідації, за якої кожне спостереження оцінюється моделями, що не навчалися на ньому.

Ансамбль поєднує декілька класифікаторів; їхні результати агрегуються для отримання надійної оцінки ймовірностей належності до класів. Для підвищення надійності цих ймовірностей застосовуються методи калібрування, що гарантують точне відображення впевненості моделі.

На основі ймовірностей компонент будує спільне представлення спостережуваних та прогнозованих міток, яке слугує основою для виявлення невідповідностей. Рядкам, для яких прогнозована мітка суттєво відрізняється від спостережуваної, присвоюється оцінка шуму, що відображає ймовірність помилкового маркування. Потім застосовується поріг для класифікації сутностей як потенційно зашумлених.

Окрім виявлення, компонент оцінює структуру шуму міток, аналізуючи переходи між справжніми та спостережуваними мітками. Це дозволяє системі розрізняти різні типи шуму, такі як рівномірний шум, що однаково впливає на всі класи, або шум, що залежить від класу та зосереджений у конкретних категоріях.

Щоб забезпечити надійність результатів, реалізація включає процедуру статистичної валідації на основі тестування перестановок. Порівнюючи спостережуваний рівень шуму з базовим рівнем, отриманим із випадково переставлених міток, система оцінює, чи є виявлений шум статистично значущим або чи можна його віднести до випадкових коливань.

З огляду на обчислювальну складність цього процесу, компонент включає адаптивні стратегії для великих наборів даних, такі як підвибірка

та спрощені конфігурації моделей. Ці коригування зберігають цілісність аналізу, забезпечуючи при цьому прийнятний час виконання.

Результат роботи компонента надає вичерпну характеристику якості міток, включаючи частку потенційно неправильно позначених екземплярів, розподіл шуму між класами та репрезентативні приклади. Ця інформація є необхідною для розуміння надійності завдань контрольованого навчання та для керівництва роботою з очищення даних.

3.5.5 Аналіз логічної узгодженості даних

Аналіз реляційної узгодженості спрямований на перевірку логічної цілісності набору даних шляхом виявлення порушень обмежень та структурних невідповідностей. Цей компонент виходить за межі статистичного аналізу, оскільки враховує принципи теорії баз даних та перевірки даних.

Реалізація підтримує різні типи обмежень, зокрема унікальність, домени значень, шаблони формату та взаємозв’язки між стовпцями.

Ключовою особливістю компонента є виявлення функціональних залежностей між ознаками. Це досягається за допомогою групового аналізу, який оцінює, чи значення одного набору стовпців однозначно визначають значення іншого стовпця. Сила цих залежностей кількісно оцінюється за допомогою міри достовірності, що дозволяє системі виявляти як точні, так і наближені взаємозв’язки.

Для узагальнення результатів компонент обчислює показник узгодженості, який відображає загальний рівень дотримання виявлених обмежень. Цей показник виводиться з тяжкості та частоти виявлених порушень, що відображає нормалізовану міру цілісності даних.

Результат включає як кількісну оцінку узгодженості, так і детальні описи виявлених порушень. Ця інформація надає цінний огляд структурних проблем у наборі даних.

3.5.6 Аналіз розподілу даних

Останній компонент цієї групи спрямований на моделювання загального розподілу числових даних з метою виявлення складних нелінійних аномалій, які можуть не бути виявлені традиційними методами. Це досягається за допомогою нейронної мережі-автоенкодера, яка навчається стислому представленню набору даних і відтворює вихідні дані на основі цього представлення.

Реалізація визначає симетричну нейронну архітектуру з шаром «вузького місця», який змушує модель фіксувати найвиразніші закономірності в даних. Під час навчання модель мінімізує помилку реконструкції, ефективно засвоюючи базову структуру набору даних. Спостереження, що значно відхиляються від цієї структури, дають вищі помилки реконструкції і тому ідентифікуються як аномалії.

Важливим аспектом реалізації є інтерпретованість результатів. Для кожної виявленої аномалії компонент обчислює помилки реконструкції для кожної ознаки, що дозволяє ідентифікувати ознаки, які найбільше сприяють відхиленню. Це забезпечує рівень розуміння, порівнянний з методами атрибуції ознак, що використовуються в інших компонентах.

Для обробки великих наборів даних компонент обирає підмножину даних під час навчання та обробляє дані партіями під час інференції.

Захоплюючи складні взаємодії між ознаками, цей компонент доповнює інші методи виявлення аномалій та сприяє всебічній оцінці якості даних.

3.5.7 Комлексна оцінка якості даних та формування звіту

На фінальному етапі аналітичного конвеєра результати роботи всіх попередніх компонентів об’єднуються в єдину оцінку якості набору даних, що забезпечує як кількісну оцінку, так і зрозумілу звітність. Ця

функціональність реалізована у вигляді механізму комплексної оцінки, який інтегрує різнорідні метрики в єдину цілісну систему, забезпечуючи цілісне уявлення про якість даних без необхідності ручного об’єднання даних користувачем.

Реалізація базується на принципі зваженого об’єднання, де кожна раніше обчислена метрика впливає на загальний показник якості відповідно до своєї відносної важливості. Ці метрики включають показники, отримані в результаті аналізу відсутніх значень, виявлення дублікатів, виявлення аномалій, оцінки шуму міток та перевірки узгодженості. Кожна метрика спочатку нормалізується для забезпечення порівнянності, після чого вона включається до кінцевого показника з використанням заздалегідь визначених ваг. Вибір ваг відображає практичний вплив різних типів проблем з даними на подальші аналітичні завдання, особливо в контексті машинного навчання [34].

Важливим аспектом цього компонента є перетворення необроблених метрик у зрозумілі виміри якості. Замість того, щоб подавати одне агреговане значення окремо, система подає результати у концептуальних категоріях, таких як повнота, узгодженість, унікальність та валідність. Це структуроване представлення дозволяє користувачам зрозуміти не лише загальний рівень якості, а й конкретні фактори, що сприяють потенційному погіршенню.

Функціональність звітності реалізована як окремий, але тісно інтегрований шар, який перетворює агреговані результати на самостійний HTML-документ. Цей звіт поєднує текстові пояснення, табличні резюме та візуальні елементи, щоб представити аналіз у структурованому та доступному форматі. Процес генерації базується на системі шаблонів, яка динамічно включає вихідні дані компонентів, забезпечуючи те, щоб звіт точно відображав результати аналізу.

З точки зору реалізації особлива увага приділяється узгодженості та відтворюваності результатів звіту. Оскільки всі метрики походять із

спільного контексту аналізу та обробляються за допомогою детермінованої логіки агрегації, згенеровані звіти залишаються дуже близькими за змістом при повторному виконанні.

Узагальнюючи, механізм комплексної оцінки та звітності слугує інтеграційним шаром системи, заповнюючи прогалину між детальними аналітичними обчисленнями та орієнтованою на користувача інтерпретацією.

3.6 Інтеграція великих мовних моделей

3.6.1 Архітектура LLM-інтеграції

Інтеграція великих мовних моделей у систему AEDA реалізована у вигляді окремого сервісного рівня, чітко відокремленого від аналітичного конвеєра. Це архітектурне рішення гарантує, що основні компоненти аналізу даних залишаються повністю детермінованими, тоді як мовна модель працює як допоміжний модуль, відповідальний виключно за завдання інтерпретації. Таке розділення функціональних областей дозволяє системі зберегти відтворюваність і прозорість, водночас користуючись перевагами можливостей генерації природної мови.

На рівні реалізації функціональність LLM інкапсульована в сервіс, незалежний від провайдера, розроблений з використанням шаблону Singleton. Це гарантує, що модель інстанціюється лише один раз протягом життєвого циклу виконання і згодом повторно використовується у всіх компонентах, які потребують текстових пояснень.

Ключовим аспектом архітектури є підтримка декількох провайдерів бекенду з пріоритетною стратегією резервного переходу. Система намагається завантажити локально збережену квантовану модель у форматі GGUF, а якщо вона недоступна, переходить до реалізації на базі HuggingFace [35]. За відсутності обох систем, наприклад якщо застосунок

увімкнено локально, логіка коректно вимикає функціональність LLM, не впливаючи на аналітичний конвеєр. Ця конструкція гарантує надійність та адаптивність у різних сценаріях розгортання.

Вибір моделі базувався на суворих практичних обмеженнях, зокрема на вимозі виконання лише на процесорі та обмеженому використанні пам’яті. Ранні експерименти з більшими моделями показали, що вони створюють надмірне обчислювальне навантаження без пропорційного підвищення якості пояснень для цільового випадку використання. Як результат, остаточна реалізація використовує компактну модель з оптимізованими інструкціями та агресивною квантизацією, що дозволяє виконувати її в межах приблизно 350 МБ оперативної пам’яті. Незважаючи на невеликий розмір, модель демонструє достатню здатність генерувати лаконічні пояснення, що враховують контекст.

Основним принципом проєктування, що лежить в основі інтеграції, є збереження конфіденційності. Усі операції з LLM виконуються локально, і вміст набору даних не передається назовні. Це гарантує, що систему можна безпечно застосовувати в середовищах, що містять конфіденційні дані. Модель автоматично завантажується та кешується при першому запиті, що мінімізує втручання користувача та спрощує процес налаштування.

3.6.2 Ролі LLM у системі

Роль великої мовної моделі в AEDA навмисно обмежена завданнями інтерпретації та узагальнення. Вона не бере участі в жодному етапі обробки даних, статистичних обчислень або прийняття рішень, пов’язаних із виявленням аномалій. Це обмеження має вирішальне значення для забезпечення надійності та відтворюваності системи, оскільки всі аналітичні результати повинні залишатися незалежними від ймовірнісного генерування тексту.

По-перше, застосування LLM полягає у наданні пояснень щодо окремих результатів, виявлених аналітичними компонентами. До них належать, наприклад, аномальні точки даних, записи, що є близькими дублікатами, або підозрілі категоріальні значення. Пояснення генеруються для обмеженої кількості репрезентативних прикладів, щоб збалансувати інформативність та обчислювальну ефективність. Кожне пояснення є навмисно стислим, зазвичай обмеженим одним або двома реченнями, що гарантує, що згенерований текст покращує інтерпретованість, не перевантажуючи користувача.

Друга задача полягає у генеруванні підсумків на рівні компонентів. Після завершення виконання кожен аналітичний компонент створює структуровані результати, що описують виявлені проблеми та обчислені метрики. LLM перетворює ці результати на короткі описові тексти, які надають контекст знахідкам та підкреслюють їх практичне значення. Цей рівень слугує мостом між необробленими числовими показниками та людським розумінням, сприяючи швидшій інтерпретації результатів.

Третя роль – це генерування підсумку на рівні набору даних, що синтезує результати всіх компонентів. Цей глобальний опис надає загальну оцінку якості даних, підкреслюючи найкритичніші проблеми та їхній потенційний вплив на подальші завдання машинного навчання. Агрегуючи інформацію за різними аналітичними вимірами, LLM створює послідовну та високорівневу інтерпретацію, яка в іншому випадку вимагала б ручної роботи.

Щоб контролювати обчислювальні витрати та запобігти надмірному генеруванню, встановлюються суворі обмеження на кількість оброблених прикладів та максимальну довжину токенів у кожній відповіді. Ці обмеження гарантують, що інтеграція залишається ефективною та передбачуваною навіть для великих наборів даних.

3.6.3 Prompt Engineering

Ефективність інтеграції LLM значною мірою залежить від ретельно розроблених підказок, які спрямовують модель на створення точних, лаконічних і контекстуально обґрунтованих пояснень. Розробка підказок вважається критично важливим компонентом системи, оскільки неправильно сформульовані підказки можуть призвести до надмірно великих, нерелевантних або вигаданих відповідей.

При створенні підказок система дотримується чотирьох ключових принципів. По-перше, встановлюються суворі обмеження щодо довжини, щоб відповіді залишалися лаконічними та цілеспрямованими. Модель отримує чітку інструкцію генерувати короткі пояснення, зазвичай обмежені заздалегідь визначеною кількістю речень або слів. Це запобігає зайвій багатослівності та забезпечує узгодженість між різними компонентами.

По-друге, підказки ґрунтуються на фактичних даних та обчислених метриках. Замість того, щоб покладатися на абстрактні описи, модель отримує конкретні значення, такі як внесок ознак, показники аномалій або підрахунки частоти. Таке ґрунтування значно зменшує ймовірність галюцинацій та гарантує, що згенеровані пояснення залишаються пов’язаними з спостережуваними доказами.

По-третє, підказки явно розроблені для запобігання спекулятивним або необґрунтованим міркуванням. Модель налаштована уникати припущень, що виходять за межі наданих даних, і суворо зосереджуватися на інтерпретації наданої інформації. Це особливо важливо в контексті аналізу якості даних, де неправильні пояснення можуть ввести користувачів в оману.

По-четверте, підказки побудовані так, щоб уникнути генерації коду або надмірно технічних описів. Мета виводу LLM – надати інтуїтивні, зрозумілі для людини пояснення, а не повторювати деталі реалізації.

Процес розробки підказок був ітеративним і передбачав численні вдосконалення на основі емпіричної оцінки виводів моделі. Початкові версії підказок, як правило, давали надто розлогі або загальні відповіді. Завдяки послідовним вдосконаленням підказки були оптимізовані для досягнення балансу між стислістю, конкретністю та інтерпретованістю.

3.6.4 Детерміністична стратегія у виявленні шуму міток

На відміну від інших компонентів, що спираються на пояснення, згенеровані LLM, компонент виявлення шуму міток використовує повністю детерміновану стратегію пояснення. Таке проєктне рішення обумовлене надзвичайною важливістю надійності та відтворюваності в контексті керованого навчання, де неправильне тлумачення шуму міток може призвести до помилкових рішень щодо очищення даних.

Пояснення шуму міток базується на результатах базової платформи Confident Learning та прогнозах ансамблевого класифікатора. Для кожного підозрілого зразка система аналізує прогнозовані ймовірності класів, порівнюючи ймовірність, присвоєну поточній мітці, з ймовірностями альтернативних класів. Значна розбіжність між цими ймовірностями вказує на високу ймовірність неправильного маркування.

Окрім аналізу окремих зразків, система використовує оцінену матрицю переходу шуму для виявлення систематичних патернів помилкового маркування між класами. Це дозволяє не тільки виявляти окремі зашумлені екземпляри, але й характеризувати ширші структури шуму в наборі даних, такі як класово-умовна плутанина.

Детерміноване пояснення включає в себе кілька сигналів, зокрема найімовірніший альтернативний клас, впевненість у прогнозі та загальний рівень шуму для відповідного класу. Поєднуючи ці фактори, система генерує пояснення, які є як інтерпретованими, так і обґрунтованими статистичними даними.

Рішення уникнути використання LLM у цьому компоненті базується на кількох міркуваннях. По-перше, детерміновані пояснення є повністю відтворюваними, що забезпечує узгодженість при повторних запусках. По-друге, вони усувають ризик галюцинацій або оманливих інтерпретацій. По-третє, вони забезпечують пряму простежуваність до вихідних даних базової моделі, що є важливим для налагодження та валідації.

3.7 Генерація звітів та інтерфейс користувача

3.7.1 Створення фінальних звітів

Останнім етапом у конвеєрі AEDA є перетворення результатів аналізу на комплексний, самодостатній звіт у форматі HTML. Цей компонент слугує основним інтерфейсом між системою та кінцевим користувачем, об’єднуючи результати роботи всіх аналітичних модулів у структурований та зрозумілий формат. Генератор звітів реалізовано як автономний модуль, призначений для роботи незалежно від середовища виконання та створення переносного артефакту, який можна переглянути в будь-якому сучасному веб-браузері без додаткових залежностей. Важливою особливістю звіту є поєднання візуальних елементів, статистичних узагальнень та текстових пояснень, що забезпечує багаторівневе представлення результатів аналізу. Такий підхід дозволяє користувачеві не лише отримати доступ до числових показників, але й зрозуміти їхній зміст та взаємозв’язки. Крім того, структура звіту організована таким чином, щоб підтримувати послідовний перехід від загального огляду до детального аналізу окремих аспектів якості даних.

Створений звіт має ієрархічну структуру, що віддзеркалює конвеєр виконання. Він починається з загального огляду, за яким йдуть детальні розділи, що відповідають кожному аналітичному компоненту, і завершується інтегрованою оцінкою якості даних (рисунок 3.3).

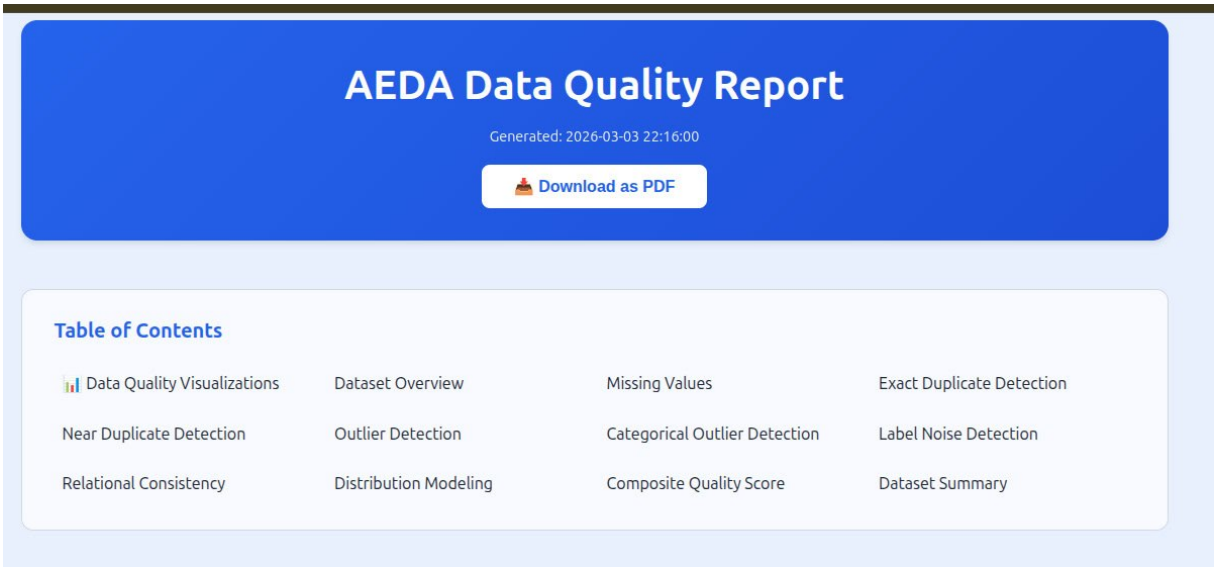


Рисунок 3.3 – Структура HTML-звіту

Така організація забезпечує користувачам можливість логічно та інтуїтивно переходити від загальних висновків до конкретних результатів. Динамічно генерований зміст ще більше підвищує зручність використання, надаючи прямий доступ до окремих розділів. Це особливо важливо при роботі з великими наборами даних, де кількість аналітичної інформації може бути значною. Крім того, навігаційна структура звіту сприяє швидкому виявленню критичних проблем якості даних без необхідності послідовного перегляду всього документа. Такий підхід зменшує когнітивне навантаження на користувача та підвищує ефективність аналізу. В результаті звіт виступає не лише як засіб представлення результатів, а як інструмент взаємодії з ними.

Усі необхідні ресурси, включаючи таблиці стилів, логіку JavaScript та візуалізації, вбудовані безпосередньо у файл HTML. Це усуває зовнішні залежності та гарантує, що звіт можна легко поширювати, архівувати або експортувати без втрати функціональності. Реалізація досягає цього шляхом вбудовування повного інтерфейсного шару, включаючи інтерактивні елементи на базі легкої бібліотеки D3 [36].

Візуалізація відіграє вирішальну роль у покращенні інтерпретованості. Звіт містить різні типи діаграм, такі як гістограми, зведення розподілів та порівняльні візуалізації статистики ознак (рисунок 3.4).



Рисунок 3.4 – Основі візуалізації у звіті

Ці візуальні елементи генеруються динамічно на основі результатів аналітичних компонентів і безперешкодно інтегруються з текстовими поясненнями. Особлива увага приділяється контекстно-залежному форматуванню метрик, де числові значення подаються як в абсолютних, так і в відносних величинах для підвищення зрозумілості. Додатково реалізовано механізми виділення критичних значень та потенційних проблем за допомогою візуальних індикаторів (рисунок 3.5).

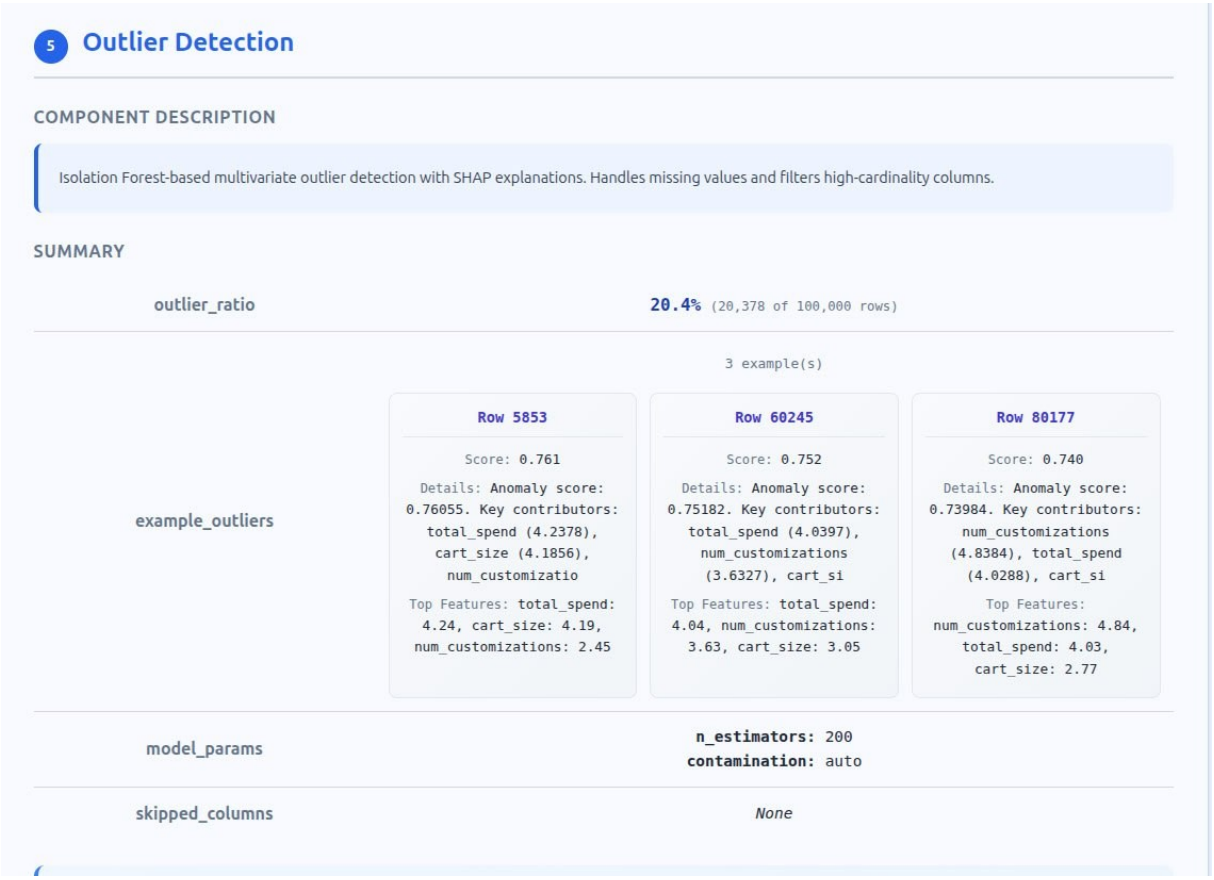


Рисунок 3.5 – Результати роботи компонента

Іншим важливим аспектом дизайну звіту є представлення висновків на основі прикладів. Для компонентів, що виявляють конкретні проблемні записи, такі як випадіння або майже дублікати, звіт містить структуровані «картки прикладів», які виділяють ключові атрибути та надають пояснювальний контекст. Ці картки поєднують табличні дані з інтерпретаційним текстом, що дозволяє користувачам швидко зрозуміти суть виявлених проблем.

Поєднання інтерактивних візуалізацій та контекстуальних пояснень забезпечує високий рівень сприйняття результатів дослідницького аналізу. Така структура звіту дозволяє фахівцеві з аналізу даних не витрачати час на ручне зіставлення розрізнених статистичних показників, а відразу переходити до обґрунтованого прийняття рішень щодо стратегії попередньої обробки та очищення вибірки (рисунок 3.6).

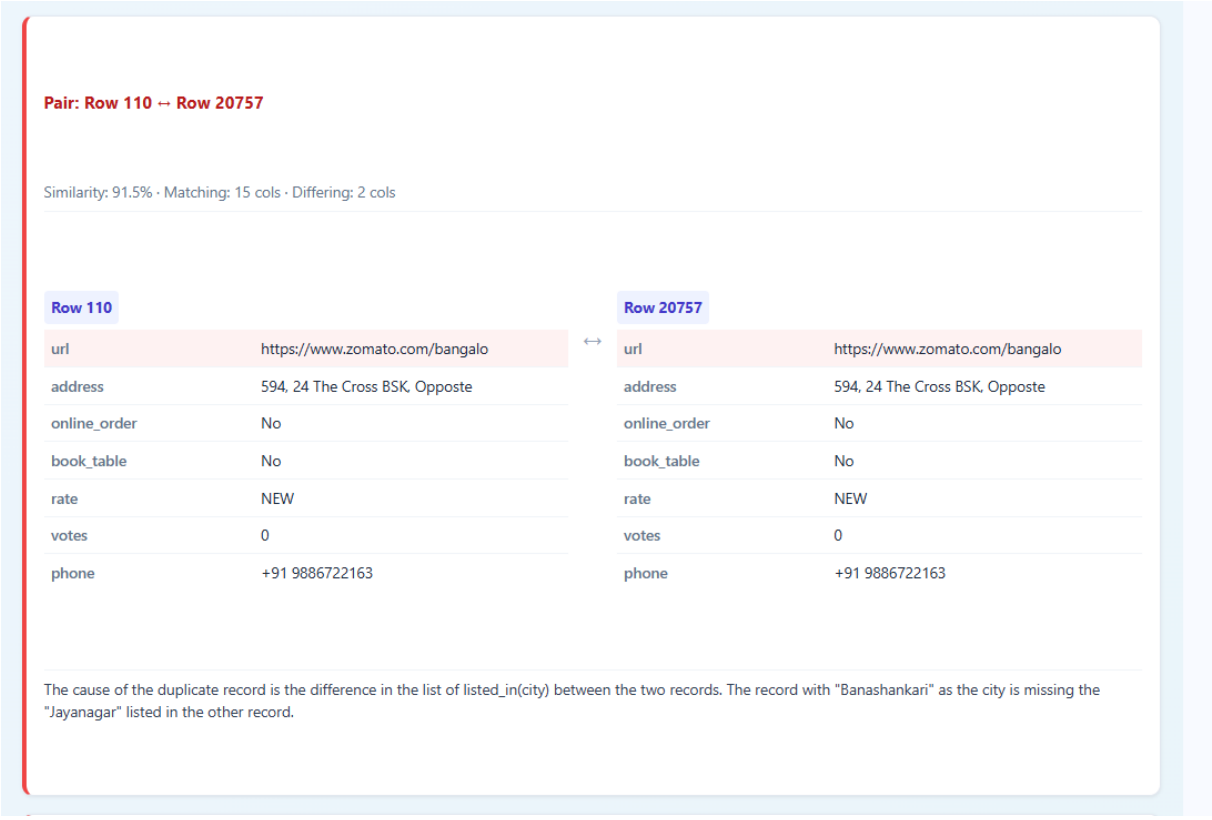


Рисунок 3.6 – Пояснення максимально схожих прикладів на основі LLM

Звіт також підтримує експорт у форматі PDF за допомогою вбудованих функцій браузера. Це досягається за допомогою спеціально оптимізованих стилів для друку, які забезпечують коректне відображення структури та візуальних елементів у статичному форматі. Як результат, система підтримує як інтерактивний аналіз, так і використання статичної документації.

3.7.2 Десктопний користувацький інтерфейс

Окрім програмного інтерфейсу, AEDA надає графічний інтерфейс користувача для комп’ютерів, розроблений для того, щоб зробити систему доступною для користувачів без досвіду програмування. Графічний інтерфейс реалізовано з використанням легкого кросплатформного фреймворку, який було обрано завдяки його мінімальній кількості

залежностей та сумісності з інструментами для створення виконуваних пакетів (рисунок 3.7).

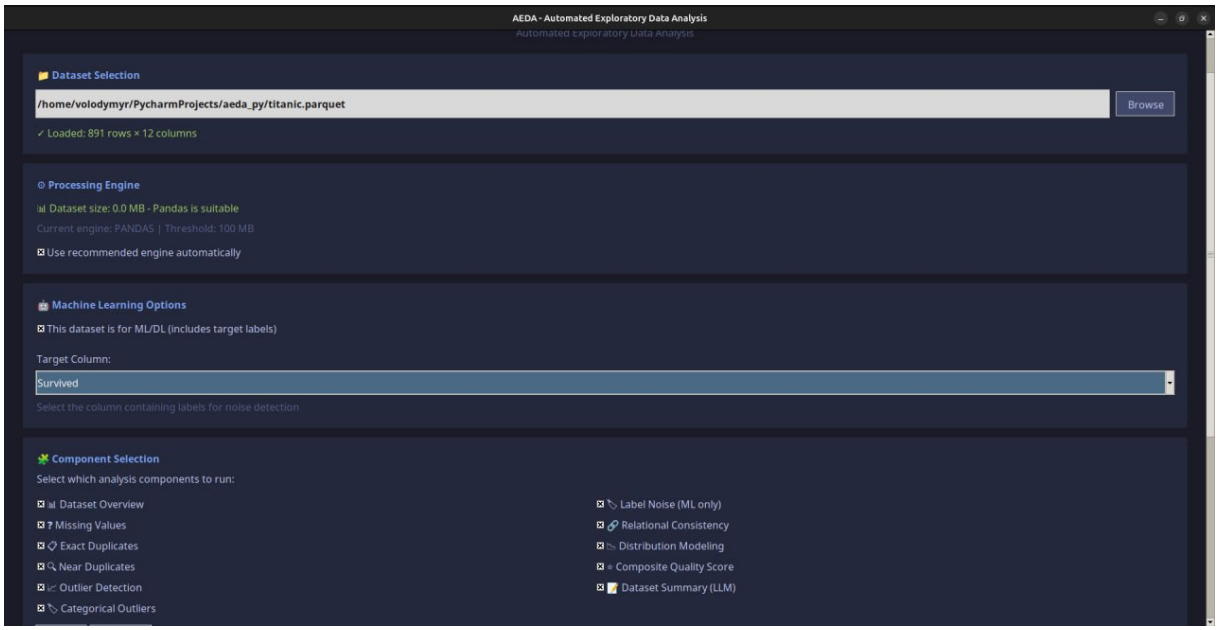


Рисунок 3.7 – Головна сторінка десктоп-застосунку

Інтерфейс побудований навколо набору логічно згрупованих панелей налаштувань, які супроводжують користувача протягом процесу аналізу, що також показані на рисунку 3.7. Ці панелі включають вибір набору даних, налаштування двигуна, параметри машинного навчання, вибір компонентів та контроль виконання. Така організація відображає базову архітектуру системи, виставляючи на показ ключові параметри конфігурації та зберігаючи простоту.

Макет використовує структуру на основі карток, де кожен функціональний блок візуально відокремлений, що дозволяє користувачам зосередитися на конкретних аспектах конфігурації без відволікання.

З точки зору реалізації, графічний інтерфейс працює асинхронно, щоб підтримувати швидкість реагування під час тривалих аналізів. Виконання конвеєра аналізу делегується фоновому потоку, тоді як головний потік

залишається відповідальним за обробку взаємодій користувача та оновлення інтерфейсу.

Додатковою функцією інтерфейсу є можливість переривати поточний аналіз. Це реалізовано за допомогою механізму кооперативного скасування, де довготривалі компоненти періодично перевіряють наявність сигналу завершення та плавно зупиняють виконання за запитом. Ця можливість є особливо важливою під час роботи з великими наборами даних або обчислювально інтенсивними компонентами (рисунок 3.8).



Рисунок 3.8 – Процес виконання компонентів аналізу

Загалом, графічний інтерфейс користувача забезпечує практичний та зручний для користувача вхід до системи, доповнюючи програмний API та забезпечуючи ширше застосування серед різних груп користувачів.

3.7.3 Розповсюдження системи

AEDA розроблена для підтримки декількох режимів розповсюдження, кожен з яких орієнтований на певну категорію користувачів та сценарії використання.

Перший режим – це бібліотека Python, яка надає доступ до повного функціоналу системи через програмний інтерфейс. Цей режим призначений для аналітиків даних та розробників, які бажають інтегрувати автоматизований аналіз якості даних у свої робочі процеси або пайплайни.

Бібліотеку можна встановити за допомогою інструменту управління пакетами pip. Вона забезпечує високий рівень гнучкості та розширюваності.

Другий режим – це десктоп-застосунок, який пропонує графічний інтерфейс для інтерактивного використання. Цей режим орієнтований на користувачів, які віддають перевагу візуальному робочому процесу або не мають досвіду програмування. Він дозволяє легко виконувати аналіз із мінімальною настройкою, що робить систему доступною для ширшої аудиторії.

Для підтримки безперервної розробки та розгортання реалізовано автоматизований конвеєр із використанням хмарної системи GitHub Actions. Кожне оновлення основної кодової бази запускає автоматичне тестування та пакування, після чого відбувається розгортання у репозиторій пакетів. Таким чином, користувачі завжди мають доступ до останньої стабільної версії системи, зберігаючи при цьому надійність та узгодженість.

3.8 Дослідження продуктивності та масштабованості

3.8.1 Оптимізації компонентів

Ключовим практичним завданням, яке вирішувалось під час розробки AEDA, було забезпечення ефективності та оперативності системи при роботі з наборами даних різного розміру та складності. Замість того, щоб покладатися на єдину глобальну стратегію оптимізації, система використовує деталізований підхід, за якого кожен аналітичний компонент реалізує власні механізми масштабованості. Така конструкція ґрунтується на спостереженні, що різні алгоритми мають свої характерні обчислювальні «вузькі місця» і тому потребують індивідуальних методів оптимізації.

Для обчислювально інтенсивних компонентів, таких як виявлення близьких дублікатів та аналіз шуму міток, основна стратегія передбачає адаптивну вибірку. Замість обробки всього набору даних, ці компоненти

працюють з репрезентативними підмножинами, коли набір даних перевищує заздалегідь визначені порогові значення. Це значно зменшує обчислювальну складність, зберігаючи при цьому статистичні властивості, необхідні для надійного аналізу. Процес вибірки ретельно розроблений для збереження балансу класів та характеристик розподілу, особливо у сценаріях, що передбачають навчання з наставником.

Виконання з обмеженням за часом – це ще один важливий механізм, що використовується для забезпечення надійності. Компоненти, що базуються на ітеративних або комбінаторних процедурах, таких як виявлення функціональної залежності або пошук схожості, обмежуються явними параметрами тайм-ауту. Якщо обчислення перевищують виділений час, компонент коректно завершується та повертає часткові результати. Це запобігає тому, щоб система перестала відповідати на великі або суперечливі набори даних.

Окрім вибірки та часових обмежень, у відповідних випадках застосовуються оптимізації, специфічні для алгоритмів. Наприклад, при виявленні близьких дублікатів кількість пар-кандидатів, що розглядаються для детального порівняння, обмежена, а попередня фільтрація використовується для відкидання явно несхожих записів. При виявленні аномалій кількість оцінювачів та вибіркових точок даних динамічно коригується на основі розміру набору даних. Аналогічно, компонент моделювання розподілу на основі автоенкодера використовує раннє зупинення, щоб запобігти непотрібним ітераціям навчання після досягнення збіжності. Завдяки такому багаторівневому підходу система гнучко адаптує свої обчислювальні можливості до наявних ресурсів та характеристик набору даних, забезпечуючи стабільно високу продуктивність без втрати глибини аналітичного дослідження, що є дуже важливим при роботі із будь-яким набором даних. Для наочності список проблем та покращень описано у таблиці 3.1.

Таблиця 3.1 – Основні оптимізації застосовані в роботі

Компонент	Проблема	Покращення
Near-Duplicate Detection	Квадратична кількість порівнянь на великих наборах даних	MinHash та LSH для наближеного пошуку кандидатів замість повного перебору
Outlier Detection	Повільне навчання на великих датасетах	Зменшення n_estimators до 100; обмеження max_samples до 10 000 рядків; пояснення лише для топ-500 аномалій (OUTLIER_MAX_EXPLAIN_ROWS)
Label Noise Detection	Ансамбль з 4 класифікаторів використовує > 14 ГБ RAM	Особлива вибірка до 10 000 рядків, для великих датасетів: скорочений ансамбль (2 замість 4 класифікаторів), 3 фолди замість 5, 5 перестановок замість 20
Relational Consistency	Повторні обчислення та сповільнення	Використання groupby_cache для зберігання результатів групування, які повторно використовуються для різних залежних стовпців
Categorical Outlier Detection	Стовпці близькі до унікальних (наприклад, імена) генерують хибні аномалії	Автоматичне пропускання стовпців з кардинальністю >80%; адаптивні пороги частоти залежно від розміру датасету
LLM Service	Великі вхідні/вихідні дані від LLM – повільна генерація та нерелевантні відповіді	Обмеження пояснюваних прикладів до 5; максимум 256 токенів на відповідь; мінімізація контексту в промптах; строгі інструкції щодо довжини (2 речення)

Сукупний ефект цих стратегій – це система, яка адаптує свої обчислювальні можливості до наявних ресурсів та характеристик набору даних, що забезпечує баланс між продуктивністю та глибиною аналізу.

3.8.2 Оптимізація LLM

Впровадження великих мовних моделей також тягне за собою додаткові вимоги до продуктивності, зокрема через обчислювальні витрати на інференцію. Щоб вирішити цю проблему, застосовують кілька стратегій оптимізації, які гарантують, що використання пояснень на основі великих мовних моделей не призведе до істотного зниження загальної продуктивності системи.

Однією з основних оптимізацій є застосування агресивної квантизації моделі. Зменшуючи точність ваг моделі до представлення з низькою кількістю бітів, система досягає суттєвого скорочення використання пам’яті та часу інференції, що робить можливим запуск моделі на доволі простому процесорі. Це є ключовою вимогою для збереження портативності та доступності системи.

Іншою важливою стратегією є використання паттерна Singleton для управління моделлю. Модель мови завантажується лише один раз і повторно використовується у всіх компонентах, що усуває накладні витрати на повторну ініціалізацію. Це особливо корисно в сценаріях, де кілька компонентів послідовно запитують пояснення.

Для додаткового контролю обчислювальних витрат встановлюються суворі обмеження на кількість генерованих пояснень та їхню довжину. До LLM передається лише невелика підмножина репрезентативних прикладів з кожного компонента, а максимальна кількість токенів на відповідь обмежується. Це гарантує, що загальне навантаження на інференцію залишається обмеженим, незалежно від розміру набору даних.

Нарешті, застосовуються техніки мінімізації запитів, щоб зменшити розмір вхідних даних, що надаються моделі. Включаючи лише найважливішу інформацію та уникаючи зайвого контексту, система зменшує як затримку, так і ризик отримання нерелевантних або надмірно розгорнутих результатів. Це сприяє не лише покращенню продуктивності, а й загальній якості та узгодженості генерованих пояснень.

3.9 Узагальнення результатів практичних досліджень

3.9.1 Порівняння з існуючими інструментами автоматизованого EDA

Проведені практичні дослідження показують, що запропонована система AEDA надає унікальні можливості порівняно з існуючими інструментами автоматизованого дослідницького аналізу даних. Хоча широко використовувані рішення зосереджуються переважно на описовій статистиці та візуалізації, їхня функціональність залишається обмеженою поверхневим профілюванням наборів даних. Натомість AEDA розширює сферу автоматизованого аналізу в бік діагностики, орієнтованої на машинне навчання, що дозволяє виявляти проблеми, які безпосередньо впливають на ефективність моделі.

Ключовим фактором, що відрізняє систему, є глибина аналітичного охоплення. Традиційні інструменти зазвичай надають інформацію про розподіли, кореляції та відсутні значення, але не включають багатовимірне виявлення аномалій, оцінку шуму міток або навчені представлення розподілів даних. Як наслідок, вони не призначені для виявлення складних проблем якості даних, таких як неправильно позначені екземпляри або тонкі порушення розподілу. Запропонована система усуває це обмеження шляхом інтеграції декількох передових аналітичних компонентів, кожен з яких націлений на конкретний аспект якості даних.

Ще одна важлива відмінність полягає у формалізації оцінки якості даних. Існуючі інструменти зазвичай надають необроблені метрики, не об’єднуючи їх у єдину систему. Натомість AEDA втілює стандарт ISO 25012 у практику, перетворюючи абстрактні виміри якості на вимірювані показники та об’єднуючи їх у комплексний бал. Це надає користувачам структуровану та зрозумілу оцінку готовності набору даних до завдань машинного навчання.

Крім того, інтеграція пояснень природною мовою є значним кроком вперед порівняно з існуючими підходами. Тоді як традиційні інструменти повністю покладаються на візуалізації та числові зведення, AEDA доповнює ці результати автоматично згенерованими текстовими інтерпретаціями. Це зменшує зусилля, що треба докласти для розуміння складних метрик, та робить систему доступнішою для користувачів з різним рівнем експертизи [37].

Нарешті, AEDA демонструє вищий рівень автоматизації та адаптивності. Вона підтримує численні сценарії використання, включаючи програмну інтеграцію та графічний інтерфейс, зберігаючи при цьому сталу поведінку. Ця гнучкість значною мірою відсутня в існуючих інструментах, які зазвичай обмежуються одним режимом роботи.

3.9.2 Переваги запропонованого підходу

Запропонований підхід має низку практичних переваг, які в сукупності забезпечують його ефективність як комплексної системи аналізу якості даних. Однією з найважливіших переваг є інтеграція різнорідних аналітичних методів у єдиний робочий процес. Завдяки поєднанню класичних статистичних методів, моделей машинного навчання та формальних моделей оцінки якості система здатна виявляти широке коло проблем якості даних, для вирішення яких інакше знадобилося б кілька незалежних інструментів [38].

Наступною важливою перевагою є використання локальних великих мовних моделей для забезпечення інтерпретованості. На відміну від хмарних рішень, система гарантує, що вся обробка даних та генерація пояснень відбуваються локально, що забезпечує конфіденційність даних та дозволяє використовувати систему в обмежених середовищах. Водночас контрольоване використання великих мовних моделей підвищує зручність використання системи, перетворюючи складні аналітичні результати на стислі та змістовні описи.

Система також робить сильний акцент на масштабованості та адаптивності. Завдяки поєднанню обробки даних на основі двох рушіїв, стратегій оптимізації для кожного компонента та обмеженого використання ресурсів, AEDA підтримує стабільну продуктивність для наборів даних різного розміру. Це робить її придатною як для дослідницького аналізу невеликих наборів даних, так і для більш вимогливих реальних сценаріїв.

Використання фіксованих випадкових початкових значень, детермінованого порядку виконання та централізованої конфігурації гарантує, що результати аналізу залишаються послідовними при повторних запусках. Це особливо важливо в наукових та промислових контекстах, де відтворюваність є фундаментальною вимогою.

Крім того, модульна архітектура системи підтримує розширюваність. Нові аналітичні компоненти можна інтегрувати без модифікації існуючого коду, що дозволяє системі розвиватися з часом. Це відповідає сучасним принципам розробки програмного забезпечення та забезпечує легку довгострокову підтримку [39].

3.9.3 Обмеження розробленої системи

Незважаючи на свої переваги, запропонована система має низку обмежень, про які слід пам’ятати. Одним із головних обмежень є обчислювальні витрати, пов’язані з певними складними аналітичними

компонентами. Хоча використовується низка стратегій оптимізації, такі методи, як виявлення шуму міток та моделювання розподілу, залишаються ресурсозатратними, особливо для дуже великих наборів даних.

Інше обмеження пов’язане з використанням евристичних порогових значень та заздалегідь визначених конфігурацій. Хоча ці параметри ретельно підбираються та перевіряються на практиці, вони можуть виявитися неоптимальними для всіх можливих наборів даних або галузей. У деяких випадках знання, специфічні для певної галузі, можуть ще більше підвищити точність та релевантність виявлених проблем.

Використання компактних локальних мовних моделей також призводить до компромісів щодо якості пояснень. Хоча обрані моделі забезпечують прийнятну продуктивність в межах обмежених ресурсів, вони можуть іноді давати надто загальні або менш точні інтерпретації порівняно з більшими моделями. Це обмеження є невід’ємною частиною вимоги щодо локального виконання та вказує на потенційну сферу для майбутнього вдосконалення [40].

З архітектурної точки зору, поточна реалізація зосереджена переважно на табличних даних. Розширення системи для підтримки інших типів даних, таких як часові ряди або простий текст, значно розширило б її застосовність, але вимагало б істотних модифікацій існуючих компонентів.

Майбутні роботи можуть вирішувати ці обмеження у кількох напрямках. Одне з потенційних вдосконалень передбачає інтеграцію більш просунутих стратегій вибору моделей, що дозволить системі динамічно адаптувати аналітичні методи на основі характеристик набору даних. Іншим напрямком є вдосконалення інтеграції LLM за допомогою кращої оптимізації промптів або гібридних підходів, що поєднують локальні та опціональні хмарні моделі. Крім того, перспективним є напрям автоматичного навчання системи на основі попередніх аналізів для підвищення точності та релевантності результатів у майбутніх сценаріях використання.

3.9.4 Напрями подальшого розвитку системи

Відповідно до визначених недоліків можна виділити кілька напрямів подальшого розвитку, спрямованих на підвищення продуктивності, масштабованості та функціонального охоплення.

Поточна реалізація базується на виконанні квантованих моделей на CPU, що забезпечує доступність, але обмежує швидкість інференції. Впровадження прискорення на базі CUDA для моделей, сумісних з GGUF, значно зменшило б затримку та дозволило б використовувати більш потужні архітектури без шкоди для швидкості реагування.

Інший важливий напрямок стосується паралелізації аналітичних компонентів. Наразі компоненти виконуються послідовно, що спрощує оркестрування, але не дозволяє повною мірою використовувати наявні обчислювальні ресурси. Перехід до моделі виконання на основі спрямованого ациклічного графа дозволив би здійснювати паралельну обробку незалежних компонентів, тим самим скорочуючи загальний час виконання та покращуючи масштабованість на багатоядерних системах.

Особливо перспективним розширенням може бути підтримка потокового аналізу даних. Поточна архітектура передбачає, що весь набір даних можна завантажити в пам’ять, що стає обмеженням для наборів даних, які значно перевищують обсяг доступної оперативної пам’яті. Впровадження механізмів інкрементальної обробки дозволило б системі працювати в потоковому режимі, обробляючи дані частинами при збереженні агрегованих статистичних даних та проміжних моделей. Це значно розширило б можливості застосування системи до великомасштабних промислових наборів даних.

Подальшу розширюваність можна досягти шляхом розробки набору інструментів для створення користувацьких компонентів. Така архітектура на основі плагінів дозволила б користувачам визначати та інтегрувати власні аналітичні компоненти без модифікації основної системи. Це не

тільки підвищило б гнучкість, а й сприяло б розширенню можливостей системи силами спільноти [41].

Інтеграція з фреймворками розподіленої обробки даних є ще одним важливим напрямком. Завдяки підтримці таких систем, як Apache Spark або Ray, платформа зможе масштабуватися горизонтально та обробляти великі набори даних у розподілених середовищах. Це дозволить використовувати її в інфраструктурах даних корпоративного рівня, де обробки на одній машині недостатньо.

Нарешті, інтеграція більших і потужніших мовних моделей дає можливість ще більше підвищити якість генерованих пояснень. Хоча поточна система надає пріоритет легким моделям для локального виконання, середовища, оснащені ресурсами GPU, могли б отримати вигоду від моделей у діапазоні від 3 до 7 мільярдів параметрів. Такі моделі, ймовірно, даватимуть більш точні, контекстно-орієнтовані та нюансовані інтерпретації для складних аналітичних висновків.

Отже, всі ці напрями окреслюють чіткий шлях для еволюції системи від надійного автономного інструменту до масштабованої, розширюваної та високопродуктивної платформи для автоматизованого аналізу якості даних.

ВИСНОВКИ

У проведеному дослідженні було проаналізовано обмеження сучасних інструментів дослідницького аналізу даних, зокрема недостатній рівень автоматизації, поверхневий характер аналізу та відсутність інтелектуальних механізмів інтерпретації результатів. Отримані результати демонструють, що подолання цих обмежень можливе за рахунок застосування підходів штучного інтелекту, зокрема інтеграції методів машинного навчання, глибинного навчання та мовних моделей у межах єдиної модульної системи аналізу даних.

Головним результатом роботи є розробка інтелектуальної системи AEDA, яка реалізує підхід до автоматизованого аналізу якості даних як до задачі, що вирішується засобами штучного інтелекту. Система поєднує класичні статистичні методи з алгоритмами машинного навчання, такими як моделі виявлення аномалій та автоенкодери, що дозволяє виявляти складні, нелінійні та багатовимірні закономірності у даних. На відміну від традиційних інструментів, орієнтованих переважно на описову статистику, запропоноване рішення реалізує інтелектуальну діагностику якості даних, включаючи виявлення шуму міток, складних аномалій, неоднорідностей розподілу та структурних невідповідностей.

Важливим аспектом роботи є інтеграція методів штучного інтелекту для підвищення інтерпретованості результатів аналізу. Зокрема, використання великих мовних моделей дозволяє автоматично формувати семантичні пояснення, узагальнення та рекомендації, що трансформує результати обчислювального аналізу у зрозумілу для користувача форму. Таким чином, система реалізує не лише аналітичну, але й когнітивну функцію підтримки прийняття рішень.Ключовий науковий внесок роботи полягає у формалізації оцінки якості даних шляхом інтеграції стандарту ISO 25012. Перетворення абстрактних вимірів якості на обчислювані метрики, у поєднанні з введенням комплексного показника якості та оцінки

достовірності, забезпечує принципову та відтворювану структуру для оцінки готовності наборів даних. Цей підхід заповнює прогалину між теоретичними моделями якості та практичним аналізом даних, дозволяючи проводити послідовне порівняння між наборами даних та підтримуючи процеси прийняття рішень на основі даних.

З практичної точки зору, система впроваджує кілька інженерних інновацій, що покращують зручність використання та застосовність. До них належать архітектура обробки даних з двома двигунами, стратегії адаптивної масштабованості на рівні компонентів та автономний механізм звітності у форматі HTML, що дозволяє інтуїтивно досліджувати результати.

Результати експериментів та впровадження підтверджують, що запропонована система досягає високого рівня автоматизації, зберігаючи при цьому глибину аналізу. Модульна архітектура підтримує розширюваність, дозволяючи інтегрувати нові компоненти без впливу на існуючу функціональність. Водночас акцент на відтворюваності гарантує, що результати залишаються послідовними та перевіреними під час повторних аналізів.

Узагальнюючи, результати цієї роботи демонструють, що автоматизований аналіз якості даних можна значно покращити завдяки інтеграції машинного навчання, формальних моделей якості та контрольованого генерування природної мови. Запропонована система AEDA є кроком у напрямку інтелектуальних, інтерпретованих та масштабованих інструментів аналізу даних, здатних підтримувати сучасні робочі процеси, що базуються на даних, та підвищувати надійність систем машинного навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pei J., Han J., Kamber M. Data Mining: Concepts and Techniques. Waltham: Morgan Kaufmann, 2011. 744 p.

2. Wirth R., Hipp J. CRISP-DM: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining.* London, 2000. 29-39 p.

3. Dasu T., Johnson T. Exploratory Data Mining and Data Cleaning. Hoboken: John Wiley & Sons, 2003. 203 p.

4. Frénay B., Verleysen M. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems.* 2013. Vol. 25, no. 5. P. 845-869. URL: https://doi.org/10.1109/tnnls.2013.2292894 (date of access: 15.03.2026)

5. GitHub – Data-Centric-AI-Community/fg-data-profiling: 1 Line of code data quality profiling & exploratory data analysis for Pandas and Spark DataFrames. *GitHub.* URL: https://github.com/ydataai/ydata-profiling (date of access: 16.03.2026).

6. GitHub – Visualize and compare datasets, target values and associations, with one line of code. *GitHub.* URL: https://github.com/fbdesignpro/sweetviz (date of access: 20.03.2026).

7. DataPrep.EDA: Task-Centric Exploratory Data Analysis for Statistical Modeling in Python / J. Peng et al. *SIGMOD/PODS '21: International Conference on Management of Data*, Virtual Event China. New York, NY, USA, 2021. URL: https://doi.org/10.1145/3448016.3457330 (date of access: 25.03.2026).

8. Tukey J. W. Exploratory Data Analysis. Reading: Addison-Wesley, 1977. 688 p.

9. Little R. J. A., Rubin D. B. Statistical Analysis with Missing Data. 3rd ed. Hoboken: John Wiley & Sons, 2019. 464 p.

10. Christen P. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Berlin: Springer, 2012. 270 p.

11. Chandola V., Banerjee A., Kumar V. Anomaly detection. *ACM Computing Surveys.* 2009. Vol. 41, no. 3. P. 1–58. URL: https://doi.org/10.1145/1541880.1541882 (date of access: 01.04.2026).

12. Liu F. T., Ting K. M., Zhou Z.-H. Isolation Forest. 2008 *Eighth IEEE International Conference on Data Mining (ICDM)*, Pisa, Italy, 15–19 December 2008. URL: https://doi.org/10.1109/icdm.2008.17 (date of access: 01.04.2026).

13. Northcutt C., Jiang L., Chuang I. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research.* 2021. Vol. 70. P. 1373–1411. URL: https://doi.org/10.1613/jair.1.12125 (date of access: 03.04.2026).

14. Dietterich T. G. Ensemble methods in machine learning. *Multiple Classifier Systems.* 2000. P. 1–15.

15. Papenbrock T. et al. Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms. *Proceedings of the VLDB Endowment*. 2015. Vol. 8, no. 10. P. 1082–1093.

16. Bellman R. Dynamic Programming. Princeton: Princeton University Press, 1957. 342 p.

17. Sakurada M., Yairi T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. *the MLSDA 2014 2nd Workshop*, Gold Coast, Australia QLD, Australia, 2 December 2014. New York, New York, USA, 2014. URL: https://doi.org/10.1145/2689746.2689747 (date of access: 04.04.2026).

18. ISO/IEC 25012:2008. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model. Geneva: International Organization for Standardization, 2008. 16 p.

19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *Neural Information Processing Systems*, 30, P. 5998–6008.

20. LLM.int8(): 8-Bit Matrix Multiplication for Transformers at Scale / Y. Belkada et al. *Advances in Neural Information Processing Systems 35*, New Orleans, Louisiana, USA, 28 November – 9 December 2022. San Diego, California, USA, 2022. P. 30318–30332. URL: https://doi.org/10.52202/068431-2198 (date of access: 04.04.2026).

21. Survey of Hallucination in Natural Language Generation / Z. Ji et al. *ACM Computing Surveys*. 2022. URL: https://doi.org/10.1145/3571730 (date of access: 05.04.2026).

22. Terziyan V., Vitko O. Context-Aware Machine Learning for Smart Manufacturing. *Procedia Computer Science*. 2025. Vol. 253. P. 25-36. DOI: https://doi.org/10.1016/j.procs.2025.01.066 (date of access: 05.04.2026).

23. Вітько О. В. Методологія інтелектуалізації інформаційних технологій обробки даних у гетерогенних системах // Матеріали міжнар. наук.-практ. конф. – Харків : ХНУРЕ, 2023.

24. Terziyan V., Vitko O., Terziyan O. A Conceptual Design of Industrial Asset Maintenance System by Autonomous Agents Enhanced with ChatGPT. *Intelligent and Sustainable Manufacturing.* – 2025. – Vol. 1.

25. The Python Language Reference. *Python documentation.* URL: https://docs.python.org/3.12/reference/index.html (date of access: 07.04.2026).

26. Array programming with NumPy / C. R. Harris et al. *Nature*. 2020. Vol. 585, no. 7825. P. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2 (date of access: 08.04.2026).

27. SciPy 1.0: fundamental algorithms for scientific computing in Python / P. Virtanen et al. *Nature Methods*. 2020. Vol. 17, no. 3. P. 261–272. URL: https://doi.org/10.1038/s41592-019-0686-2 (date of access: 10.04.2026).

28. Hunter J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 2007. Vol. 9, no. 3. P. 90–95. URL: https://doi.org/10.1109/mcse.2007.55 (date of access: 11.04.2026).

29. McKinney W. Data Structures for Statistical Computing in Python. *Python in Science Conference*, Austin, Texas. 2010. URL: https://doi.org/10.25080/majora-92bf1922-00a (date of access: 12.04.2026).

30. An Empirical Study on the Energy Usage and Performance of Pandas and Polars Data Analysis Python Libraries / F. Nahrstedt et al. EASE 2024: *28th International Conference on Evaluation and Assessment in Software Engineering*, Salerno Italy. New York, NY, USA, 2024. URL: https://doi.org/10.1145/3661167.3661203 (date of access: 12.04.2026).

31. Terziyan V., Vitko O. Adaptive Generalized Mean and Social Distance Metric for Smart Manufacturing Tasks // Procedia Computer Science. 2025. DOI: https://doi.org/10.1016/j.procs.2025.01.077 (date of access: 12.04.2026).

32. Nelli F. Machine Learning with scikit-learn. *Python Data Analytics*. Berkeley, CA, 2023. P. 259–287. URL: https://doi.org/10.1007/978-1-4842-9532-8_8 (date of access: 15.04.2026).

33. Wallace E., Gardner M., Singh S. Interpreting Predictions of NLP Models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Online. Stroudsburg, PA, USA, 2020. URL: https://doi.org/10.18653/v1/2020.emnlp-tutorials.3 (date of access: 16.04.2026).

34. Batini C., Scannapieco M. Data Quality: Concepts, Methodologies and Techniques. Springer Berlin Heidelberg. 2006. P. 504 URL: https://doi.org/10.1007/3-540-33173-5 (date of access: 16.04.2026).

35. GitHub – ggml-org/llama.cpp: LLM inference in C/C++. GitHub. URL: https://github.com/ggerganov/llama.cpp (date of access: 17.04.2026).

36. Bostock M., Ogievetsky V., Heer J. D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*. 2011. Vol. 17, no. 12. P.

2301–2309. URL: https://doi.org/10.1109/tvcg.2011.185 (date of access: 19.04.2026).

37. Data driven: profiting from your most important business asset. *Choice Reviews Online*. 2009. Vol. 46, no. 06. P. 46–3345. URL: https://doi.org/10.5860/choice.46-3345 (date of access: 20.04.2026).

38. Cai L., Zhu Y. The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. *Data Science Journal*. 2015. Vol. 14. P. 2. URL: https://doi.org/10.5334/dsj-2015-002 (date of access: 20.04.2026).

39. Ge M., Dohnal V. Quality Management in Big Data. *Informatics*. 2018. Vol. 5, no. 2. P. 19. URL: https://doi.org/10.3390/informatics5020019 (date of access: 22.04.2026).

40. Data Management Challenges in Production Machine Learning / N. Polyzotis et al. *SIGMOD/PODS'17: International Conference on Management of Data*, Chicago Illinois USA. New York, NY, USA, 2017. URL: https://doi.org/10.1145/3035918.3054782 (date of access: 23.04.2026).

41. Teoh T. T., Rong Z. Artificial Intelligence with Python. *Singapore: Springer*. Singapore, 2022. URL: https://doi.org/10.1007/978-981-16-8615-3 (date of access: 24.04.2026).