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 перший (бакалаврський)

Дослідження методів автоматизації аналізу даних із використанням моделей штучного інтелекту
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-22-2

Владислав РОМАНОВ
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект

(повна назва освітньої програми)
Керівник ас. Дмитро МАЛЄЄВ
(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 перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Романову Владиславу Станіславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів автоматизації аналізу даних із використанням моделей штучного інтелекту

затверджена наказом університету від 5 червня 20 26 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 липня 20 26 р.

3. Вихідні дані до роботи Науково-технічні публікації з автоматизації аналізу даних та AutoML-систем; матеріали з методів машинного навчання і попередньої обробки даних; набори структурованих даних у форматах CSV та JSON; сучасні підходи до побудови інтелектуальних аналітичних систем

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі дослідження
2) Методи та технології автоматизації аналізу даних
3) Проектування та реалізація системи автоматизованого аналізу даних
4) Експериментальне дослідження та оцінювання роботи системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз сучасних підходів до автоматизації аналізу	12.06.2026	виконано
	даних		
3	Дослідження методів машинного навчання та	14.06.2026	виконано
	AutoML-систем		
4	Проектування архітектури системи автоматизованого	17.06.2026	виконано
	аналізу даних		
5	Розробка модулів обробки та аналізу даних	20.06.2026	виконано
6	Реалізація користувацького інтерфейсу системи	23.06.2026	виконано
7	Проведення експериментального дослідження роботи	25.06.2026	виконано
	системи		
8	Аналіз та оцінювання отриманих результатів	28.06.2026	виконано
9	Оформлення пояснювальної записки	30.06.2026	виконано
10	Підготовка до захисту кваліфікаційної роботи	05.07.2026	виконано
11	Захист перед ЕК	14.07.2026	виконано

Дата видачі завдання 8 червня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Дмитро МАЛЄЄВ
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 91 с., 14 рис., 9 табл., 1 дод., 21 джерело.

АВТОМАТИЗАЦІЯ АНАЛІЗУ ДАНИХ, АНАЛІТИЧНИЙ PIPELINE, ІНТЕЛЕКТУАЛЬНІ АНАЛІТИЧНІ СИСТЕМИ, МАШИННЕ НАВЧАННЯ, ОБРОБКА ДАНИХ, ШТУЧНИЙ ІНТЕЛЕКТ, AUTOML.

Об’єкт дослідження – процес автоматизації аналізу структурованих даних у інформаційних системах.

Предмет дослідження – методи автоматичного вибору моделей машинного навчання, попередньої обробки даних та побудови інтелектуальних аналітичних систем.

Мета роботи – дослідити та спроектувати систему автоматизованого аналізу даних із використанням моделей штучного інтелекту, здатну автоматично виконувати основні етапи аналітичного процесу.

Методи дослідження – методи машинного навчання, автоматизований вибір моделей, попередня обробка даних, аналіз ефективності моделей.

Робота присвячена дослідженню та розробці системи автоматизованого аналізу даних із використанням моделей штучного інтелекту. Розглянуто підходи до побудови AutoML-систем, методи попередньої обробки даних та автоматичного вибору моделей машинного навчання. Запропоновано архітектуру системи, що автоматизує основні етапи аналітичного процесу.

Проведено експериментальне дослідження роботи системи на різних наборах даних, яке підтвердило можливість автоматичного виконання аналізу без ручного налаштування більшості параметрів. Результати роботи можуть бути використані у системах бізнес-аналітики та підтримки прийняття рішень.

ABSTRACT

Bachelor’s thesis contains: 91 pp., 14 fig., 9 tabl., 1 ann., 21 references.

ANALYTICAL PIPELINE, ARTIFICIAL INTELLIGENCE, AUTOML, DATA ANALYSIS AUTOMATION, DATA PROCESSING, INTELLIGENT ANALYTICAL SYSTEMS, MACHINE LEARNING.

Object of research – he process of automating the analysis of structured data in information systems.

Subject of research – methods for automatic selection of machine learning models, data preprocessing, and the development of intelligent analytical systems.

Aim of the work – to investigate and design an automated data analysis system based on artificial intelligence models capable of automatically performing the main stages of the analytical process.

Research methods – machine learning methods, automated model selection techniques, data preprocessing methods, and model performance evaluation methods.

This work is devoted to the study and development of an automated data analysis system based on artificial intelligence models. Approaches to the construction of AutoML systems, data preprocessing methods, and automated machine learning model selection techniques are considered. An architecture of a system that automates the main stages of the analytical process is proposed.

An experimental study of the system was conducted using various datasets, confirming the feasibility of performing data analysis automatically without manual configuration of most parameters. The results of this work can be applied in business intelligence systems and decision support systems.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Аналіз предметної галузі та постановка задачі дослідження 10
1.1 Сутність аналізу даних та його автоматизації 10
1.2 Класи методів машинного навчання для аналізу даних 13
1.3 Сучасні підходи до AutoML та інтелектуальних аналітичних систем 16
1.4 Огляд існуючих інструментів і платформ автоматизації аналізу даних 19
1.5 Постановка задачі дослідження 22
2 Методи та моделі автоматизованого аналізу даних 25
2.1 Задача автоматизації аналізу даних в рамках програмної системи ... 25
2.2 Вибір моделей машинного навчання для автоматизованого аналізу даних 27
2.3 Методи автоматичного вибору моделей та налаштування параметрів 30
2.4 Методи підготовки даних у системі автоматизованого аналізу 33
2.5 Метрики оцінювання якості моделей 35
2.6 Запропонований алгоритм автоматизованого аналізу даних 38
3 Розробка та реалізація системи автоматизованого аналізу даних 41
3.1 Загальна архітектура програмної системи 41
3.2 Реалізація користувацького інтерфейсу 44
3.3 Модуль профілювання даних 51
3.4 Автоматичне визначення типу задачі 54
3.5 Автоматизована попередня обробка даних 55
3.6 Модуль автоматичного навчання моделей 57
3.7 Оцінка та ранжування моделей 59
3.8 Модуль візуалізації результатів 61

3.9 Генерація аналітичного звіту 62
4 Експериментальне дослідження та оцінка ефективності системи 64
4.1 Постановка експериментального дослідження 64
4.1.1 Мета, принципи та підхід до оцінювання 64
4.1.2 Експериментальне середовище та набори даних 65
4.1.3 Метрики оцінювання 67
4.2 Коректність роботи pipeline 68
4.2.1 Сценарій та методика перевірки 68
4.2.2 Результати та їх аналіз 70
4.3 Ефективність автоматизації та порівняння з базовими моделями 72
4.3.1 Методика порівняння 72
4.3.2 Результати та їх аналіз 73
4.4 Стабільність результатів 74
4.4.1 Методика оцінювання стабільності 74
4.4.2 Результати та їх аналіз 76
4.5 Оцінка роботи системи на різних типах задач 77
4.5.1 Результати для різних типів задач 77
4.5.2 Узагальнений аналіз 79
4.6 Обчислювальна ефективність системи 80
4.6.1 Методика вимірювання 80
4.6.2 Результати та аналіз 81
4.7 Аналіз використання системи неекспертним користувачем 83
Висновки 87
Перелік джерел посилання 89
Додаток А Відомість кваліфікаційної роботи 91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – програмний інтерфейс застосунку;

AutoML – Automated Machine Learning – автоматизоване машинне навчання;

CSV – Comma-Separated Values – формат табличних даних;

EDA – Exploratory Data Analysis – розвідувальний аналіз даних;

ETL – Extract, Transform, Load – процес вилучення, перетворення та завантаження даних;

JSON – JavaScript Object Notation – формат обміну даними;

KNN – K-Nearest Neighbors – метод k найближчих сусідів;

ML – Machine Learning – машинне навчання;

MSE – Mean Squared Error – середньоквадратична помилка;

REST – Representational State Transfer – архітектурний стиль вебсервісів;

UI – User Interface – користувацький інтерфейс;

UX – User Experience – користувацький досвід.

ВСТУП

На сьогоднішній день, дані стають ключовим ресурсом в прийнятті рішень в різних галузях, таких як медицина, техніка та інформаційні системи. Водночас методи штучного інтелекту набувають все більшого розповсюдження, проте через стрімке зростання обсягів даних та ускладнення інформаційних процесів, особливої актуальності проблема ефективного аналізу даних з використанням інтелектуальних технологій.

Традиційні підходи до аналізу даних дедалі частіше стають неефективними через значні витрати часу, потребу у висококваліфікованих спеціалістах та складність налаштування аналітичних моделей. В приведеному контексті, автоматизація процесів аналізу даних із використанням моделей штучного інтелекту виступає як перспективний напрям підвищення ефективності обробки інформації.

Методи штучного інтелекту характеризуються тим, що вони виявляють приховані закономірності у даних, дають можливість побудувати прогнозних моделей та автоматизувати процес прийняття рішень. Особливу роль відіграють підходи, що дозволяють мінімізувати участь спеціалістів в процесі побудови моделей, а саме автоматизоване машинне навчання, яке інтегрує вибір моделей налаштування їх гіперпараметрів та оцінювання якості результатів роботи моделей, поєднуючи їх в одну систему. Такі підходи підвищують доступність інструментів аналізу та розширюють можливості їх застосування в прикладних проєктах.

В рамках даної проблеми, підвищується необхідність розробки ефективних методів автоматизації аналізу даних, які поєднують в собі точність, адаптивність та зручність використання. Важливим аспектом також є розробка користувацького інтерфейсу, який дозволяє реалізувати повний цикл пояснювального аналізу, який включає в себе завантаження даних, їх попередньої оброки, побудову моделей та візуалізацію результатів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Перший розділ присвячено теоретичному обґрунтуванню автоматизації процесу аналізу даних з використанням методів штучного інтелекту. По-перше, розкривається сутність аналізу даних, який є першим, та ключовим етапом в прийнятті рішень. По-друге, необхідним є визначення підходів до його автоматизації в умовах зростання обсягів та складності даних, а саме розглядаються основні класи методів машинного навчання, що застосовуються для аналітичних задач, також сучасні концепції та інтелектуальних аналітичних систем і створюється огляд існуючих інструментів та платформ.

1.1 Сутність аналізу даних та його автоматизації

Аналіз даних є важливим та невід’ємним етапом розробки та роботи сучасних інформаційних систем. Він спрямований на виявлення прихованих закономірностей та структур, що є у даних для того щоб обґрунтовувати рішення в процесі підтримки прийняття рішень.

В цілому, аналіз даних охоплює сукупність методів та певних процедур, які дозволяють перетворювати сирі дані у корисну, змістову інформацію, яка має практичну цінність. У свою чергу, в умовах цифровізації суспільства і стрімкого зростання обсягів даних, традиційні підходи до їх обробки стають все дедалі менш ефективними, що обумовлює необхідність автоматизації [1].

Процес аналізу даних складається з декількох взаємозв’язаних етапів, на першому з яких здійснюється збір даних із різних джерел, які можуть відрізнятись за структурою, форматом і якістю. На наступному кроці відбувається попередня обробка даних, що включає в себе очищення від шуму, обробка пропущених значень та нормалізацію. Далі відбувається

аналітична обробка, яка передбачає застосування статистичних методів та підходів машинного навчання з метою виявлення прихованих закономірностей. Останнім, і не менш важливим етапом є інтерпретація результатів та їх представлення у зрозумілому та зручному виді для спеціаліста та моделі машинного навчання відповідно задля прийняття рішень.

Традиційний аналіз даних передбачає залучення експерта, який визначає структуру даних, обирає методи аналізу, та налаштовує параметри моделей задля коректного аналізу і, в результаті, інтерпретує результати.

Приведений підхід має ряд обмежень, серед яких висока трудоємність, залежність від висококваліфікованого спеціаліста, людського фактора та обмежена масштабованість. Також, варто зазначити, що якість і коректність отриманих результатів залежить від рівня кваліфікації аналітика, що ускладнює використання аналітичних інструментів в широкому колі задач.

Звідси, виникає потреба в автоматизації аналізу даних, що передбачає часткове або повне усунення участі людини у виконані окремих етапів процесу аналізу даних [2]. Серед переваг автоматизації можна визначити підвищення швидкості обробки даних, зменшення впливу суб’єктивних факторів та забезпечення відтворюваності результатів. Вона базується на використанні алгоритмічних підходів, що здатні самостійно виконувати певні фрагменти, або навіть повні етапи підготовки та аналізу даних, вибору моделей для подальшої роботи та їх оптимізацію.

Одним із найважливіших аспектів автоматизації є формалізація процесу аналізу даних, тобто представлення кожного етапу у вигляді чітко визначених процедур, які можуть бути програмно реалізовані.

Автоматизація аналізу даних тісно пов’язана з розвитком методів машинного навчання [1]. Такі методи забезпечують можливість побудови моделей без безпосереднього програмування, мають здатність навчатися на основі історичних даних та адаптуватися до нових умов, що призводить до

покращення ефективності із часом. Не менш важливу роль відіграють також методи оптимізації, що використовуються для налаштування параметрів моделей з метою досягнення найкращих результатів.

В рамках автоматизації процесу аналізу даних, ключовим є поняття конвеєра [2], який являє собою послідовність взаємопов’язаних операцій, що виконуються над даними.

Такий підхід дозволяє стандартизувати процес аналізу даних і забезпечити його відтворюваність та повторюваність, водночас однією з переваг є значне скорочення часу на виконання аналітичних задач.

Етап попередньої обробки даних включає надзвичайно важливий крок щодо забезпечення якості даних який прямим чином впливає на якість роботи системи машинного навчання та отримані результати. Сюди можна включити механізми виявлення аномалій, перевірка узгодженності даних та контролю їх повноти. Виконання цього етапу дозволяє мінімізувати влив помилок у даних на результати аналізу.

Також, поширеною практикою є включення автоматизованих систем в різні інформаційні системи. Зокрема, інтеграція передбачена в автоматизації аналізу даних, що забезпечує безперервний потік даних та їх обробку в режимі реального часу. У таких системах, аналіз даних стає частиною більш широкого процесу прийняття рішень та його реалізацію.

Процес автоматизації аналізу даних, хоч і мінімізує участь експерта, але не виключає повністю, змінюючи роль, яка полягає в постановці задач, інтерпретації результатів та контролі роботи системи. Таким чином можна підвищити ефективність аналізу даних, зосереджуючи експерта на стратегічних аспектах аналізу.

Отже, аналіз даних та його автоматизація визначають ефективність сучасних інформаційних систем. Автоматизація дозволяє подолати обмеження традиційних підходів, підвищуючи якість результатів та забезпечуючи масштабованість.

1.2 Класи методів машинного навчання для аналізу даних

Методи машинного навчання лежать в основі сучасного аналізу даних, завдяки їх можливостям знаходити приховані закономірності, будувати прогнози та приймати рішення на основі отриманих великих обсягів даних.

Загалом, машинне навчання визначаються як підхід до побудови моделей які здатні навчатися з даних без явного програмування правил. Приведений підхід досягається шляхом оптимізації певної цільової функції на основі навчальної вибірки [3], звідси модель може узагальнювати отримані знання на нові, раніше невідомі дані.

Класифікація методів машинного навчання базується на томі які дані подаються для вирішення певної задачі. Класичний поділ має три основні парадигми, навчання з учителем, навчання без учителя та навчання з підкріпленням. Кожен з них має власні особливості, переваги та недоліки, визначаючи сферу їх застосування.

Навчання з учителем передбачає побудову моделі яка має здатність встановлювати залежність між вхідними ознаками та цільовою змінною, використовуючи розмічені дані. Вони характерні тим, що мають об’єкт і відповідний до нього правильний результат, також позначається як мітка.

В рамках приведеної парадигми існує два класи задач – регресія та класифікація. У задачах класифікації модель відносить об’єкти до визначених класів, тоді як в задачі регресії передбачається прогнозувати неперервних значень

Серед найбільш поширених методів машинного навчання в рамках парадигми навчання із вчителем варто виділити лінійні моделі, дерева рішень, ансамблеві методи та нейронні мережі. Лінійним моделям характерна простота та інтерпретовність, проте їх застосовність дуже обмежена, адже лінійні залежності не характерні реальним даним.

У свою чергу, дерева рішень дозволяють моделювати складні нелінійні залежності та мають гарну інтерпретовну структуру. Ансамблеві методи, такі як випадковий ліс або градієнтний бустинг, поєднують декілька базових, більш простих моделей з метою підвищення точності та стійкості. Найбільш популярними і просунутими методами є нейронні мережі, які здатні апроксимувати складні функції та ефективно працювати з великими обсягами даних, проте вони потребують значних обчислювальних ресурсів та мають більш складний процес навчання та налаштування.

Друга парадигма це навчання без учителя, яке застосовується у випадках коли дані не містять міток і метою є виявлення внутрішньої структури або закономірностей у даних. Серед основних задач цього класу можна визначити кластеризацію, зменшення розмірності та виявлення аномалій.

Кластеризація, замість того щоб шукати приховані закономірності у розмічених навчальних даних, так як наприклад класифікація, вона групує об’єкти за подібністю їх характеристик, наприклад, під час сегментації клієнтів або є аналізі поведінки користувачів.

Методи зменшення розмірності використовуються для зменшення кількості ознак, при цьому максимально зберігаючи всю корисну інформацію, для того щоб підвищити ефективність аналізу або візуалізації даних.

Виявлення аномалій спрямоване на ідентифікацію об’єктів, що суттєво відрізняються від основної маси даних, і застосовується у задачах виявлення шахрайства або технічних збоїв.

Третя популярна парадигма навчання це навчання із підкріпленням, яке базується на взаємодії агента та середовища. Навчання відбувається шляхом отримання винагороди або штрафу за виконані дії, формуючи крок за кроком оптимальну поведінку.

Такі методи застосовуються в задачах управління, робототехніці та оптимізації складних процесів. Хоча навчання з підкріплення менш

поширене в класичних задачах аналізу даних, воно набуває все більшого значення в контексті автоматизованих систем.

Окрім зазначних популярних парадигм, існують інші, проміжні, а також комбіновані підходи, такі як напівконтрольоване навчання та самовнавчання. Напівконтрольоване поєднує невелику кількість розмічених даних з великою нерозмічених, що дозволяє підвищити ефективність моделей у випадках, коли розмічення наборів даних, або самих міток є занадто дорого, або ж трудоємно. Самонавчання базується на здатності моделі самостійно формувати начальні сигнали на основі структури даних, що особливо актуально у задачах обробки текстів та зображень.

З точки зору практичного застосування, в автоматизації аналізу даних необхідним є поділ методів за рівнем складності та вимогами до обчислювалиї ресурсів. Прості моделі, такі як лінійна регресія, наївний баєсівський класифікатор можуть бути швидко навчені та застосовані. У свою чергу, складні моделі – нейронні мережі, потребують значних обчислюваних ресурсів та наборів даних для забезпечення високої точності у задачах, де залежності у даних є складними.

Узагальнюючі здатності моделей, іншими словами здатність ефективно працювати з новими даними, є ще одним не менш важливим аспектом. З метою забезпечення високої точності моделей, необхідно проходити класичні етапи, серед яких розподілення на навчальну, тестову вибірки та набір даних для перехнесної валідації, з використанням метрик. Обрані метрики показують коли модель працює коректно, коли виникає перенавчання (у випадку, коли модель занадто складна для поточних даних), або недонавчання (коли модель занадто проста для поточних даних).

В рамках автоматизації аналізу даних, особливого значення набуває автоматичний вибір моделей та їх параметрів. Такий підхід передбачає використання алгоритмів, які здатні самостійно визначати оптимальну архітектуру моделі, її гіперпараметри, та набір ознак.

Важливим аспектом застосування методів машинного навчання є інтерпретованість моделей, яка визначає можливість пояснення отриманих результатів та факторів, що вплинули на формування прогнозу. У багатьох прикладних сферах, зокрема у фінансових та медичних інформаційних системах, необхідним є не лише досягнення високої точності моделей, але й забезпечення прозорості їх роботи.

Отже, різні класи методів машинного навчання формують широкий набір інструментів для аналізу даних [4], вибір яких визначається характеристиками даних, цілями дослідження та вимогами до рівня автоматизації аналітичного процесу. У зв’язку з цим актуальним стає питання автоматизації вибору моделей та їх параметрів [5], що реалізується в межах сучасних підходів AutoML та інтелектуальних аналітичних систем.

1.3 Сучасні підходи до AutoML та інтелектуальних аналітичних систем

У сучасних системах аналізу даних усе частіше використовується підхід автоматизованого машинного навчання AutoML [1], основна ідея якого полягає в частковій або повній автоматизації побудови моделей. У такому випадку система самостійно формує послідовність аналітичних операцій та підбирає відповідні алгоритми. Фактично AutoML перетворює складний аналітичний процес на керований алгоритмічний конвеєр [2] де більшість технічних рішень приймається автоматично.

У структурі AutoML ключову роль відіграє оптимізація гіперпараметрів моделей [6]. Саме ці параметри значною мірою визначають якість роботи алгоритмів машинного навчання. Традиційно їх підбір виконувався вручну або через простий перебір можливих значень, проте у сучасних системах застосовуються більш ефективні методи. Найпоширенішими є випадковий пошук, байєсівська оптимізація, а також різні еволюційні алгоритми [7]. Байєсівська оптимізація дозволяє поступово

уточнювати область пошуку параметрів і таким чином зменшувати кількість експериментів необхідних для знаходження оптимальної конфігурації.

Інший важливий аспект AutoML пов’язаний з автоматичним формуванням конвеєрів машинного навчання. Конвеєр являє собою послідовність операцій що виконуються над даними. У типовому випадку він включає трансформацію ознак відбір змінних навчання моделі та оцінювання результатів. У системах AutoML структура такого конвеєра не задається вручну [8]. Вона формується автоматично шляхом комбінування різних алгоритмічних блоків. Іноді система перевіряє десятки або навіть сотні можливих конфігурацій. Потім обирається та яка забезпечує найкращу якість моделювання.

Важливим напрямом розвитку AutoML є автоматизована інженерія ознак [3]. У багатьох задачах саме представлення даних визначає ефективність моделей. Тому сучасні системи використовують алгоритми що виконують трансформації змінних генерують нові ознаки та оцінюють їх інформативність. До таких операцій належать нормалізація даних, кодування категоріальних змінних, агрегування значень іноді також застосовується зменшення розмірності. У результаті дані поступово набувають більш зручної структури для подальшого моделювання.

Окремий напрям досліджень пов’язаний із автоматичним проєктуванням архітектур моделей. Найчастіше мова йде про нейронні мережі де необхідно визначати кількість шарів типи операцій та інші параметри структури. У таких задачах використовуються еволюційні алгоритми або методи підкріплення. Іноді застосовуються градієнтні підходи до оптимізації архітектури. Система поступово змінює структуру моделі оцінює її якість і зберігає найбільш ефективні варіанти.

У ширшому контексті AutoML є частиною інтелектуальних аналітичних систем. Такі системи зазвичай включають декілька взаємопов’язаних компонентів. Це модулі інтеграції джерел даних механізми керування аналітичними процесами сховища моделей а також

інструменти візуалізації результатів. Дані проходять через ці етапи поступово перетворюючись з сирого інформаційного потоку на аналітичні результати.

Сьогодні значна частина таких систем реалізується у хмарних обчислювальних середовищах.

Це дозволяє масштабувати обчислення та виконувати паралельне тестування великої кількості моделей. У певному сенсі такий процес нагадує лабораторію де одночасно працюють десятки експериментів. Кожен з них перевіряє свою конфігурацію алгоритму і поступово формується найбільш ефективне рішення.

Ще одним напрямом розвитку є використання метанавчання. У цьому підході система використовує результати попередніх задач для прискорення побудови моделей у нових умовах. Іншими словами накопичується досвід про те які алгоритми краще працюють для певних типів даних. Завдяки цьому пошук оптимальних моделей відбувається значно швидше.

Разом з тим автоматизація побудови моделей створює і нові виклики. Одним з них є проблема інтерпретованості результатів.

Складні ансамблеві моделі або глибокі нейронні мережі часто забезпечують високу точність але їх внутрішня логіка може бути важко пояснити [9]. Саме тому останніми роками активно розвиваються методи поясняваного машинного навчання. Вони дозволяють оцінювати внесок окремих ознак та аналізувати поведінку моделей.

Таким чином сучасні підходи AutoML формують основу інтелектуальних систем аналізу даних. Вони дозволяють автоматизувати значну частину процесу побудови моделей і зробити аналітичні технології більш доступними. У результаті аналітичні системи поступово переходять від ручного налаштування моделей до автоматизованих середовищ де алгоритми самі шукають найбільш ефективні рішення.

1.4 Огляд існуючих інструментів і платформ автоматизації аналізу даних

Практична реалізація автоматизованого аналізу даних сьогодні здебільшого здійснюється через спеціалізовані програмні інструменти та платформи які реалізують принципи AutoML. Такі системи зазвичай забезпечують автоматичну побудову моделей машинного навчання починаючи з обробки даних і завершуючи оцінюванням результатів. Більшість сучасних платформ працює за принципом аналітичного конвеєра де кожен етап обробки представлений окремим модулем.

Одним із найбільш відомих відкритих інструментів є Auto-sklearn. Ця бібліотека побудована на основі середовища scikit-learn і реалізує підхід метанавчання для автоматичного вибору алгоритмів. Auto-sklearn використовує попередньо накопичену інформацію про ефективність моделей на різних наборах даних [9]. Це дозволяє швидше знаходити відповідні алгоритми для нових задач. Крім того система застосовує байєсівську оптимізацію для підбору гіперпараметрів. Внутрішньо Auto-sklearn формує ансамбль моделей що дозволяє підвищити стабільність прогнозів.

Іншим інструментом автоматизації є TPOT. Ця система використовує генетичне програмування для автоматичного створення конвеєрів машинного навчання. У межах TPOT кожна конфігурація конвеєра розглядається як окрема особина у популяції. Конвеєр може включати різні етапи обробки даних трансформації ознак відбір змінних а також різні алгоритми машинного навчання. У процесі еволюції система виконує мутації та кросовер між конфігураціями. Після декількох поколінь залишаються лише ті конвеєри які демонструють найкращу якість моделювання.

Ще одним потужним інструментом є H2O AutoML. Ця платформа орієнтована на роботу з великими наборами даних та підтримує розподілені

обчислення. У H2O реалізовано автоматичний запуск декількох алгоритмів машинного навчання з подальшою оптимізацією їх параметрів. Система тестує різні моделі зокрема градієнтний бустинг випадковий ліс логістичну регресію а також глибокі нейронні мережі. Після завершення навчання формується ансамбль моделей який зазвичай демонструє найкращі результати прогнозування.

Окрім бібліотек програмування існують також аналітичні платформи з графічним інтерфейсом. Такі системи дозволяють будувати аналітичні процеси без безпосереднього написання коду. Однією з таких платформ є KNIME. У KNIME аналітичний процес представлений у вигляді графічного робочого потоку. Кожен елемент потоку виконує конкретну операцію наприклад читання даних трансформацію ознак навчання моделі або оцінювання результатів. Користувач може комбінувати різні модулі з’єднуючи їх між собою та формуючи складні аналітичні процеси.

Подібний підхід використовується у платформі RapidMiner. Вона також дозволяє будувати конвеєри аналізу даних через графічний інтерфейс. RapidMiner підтримує широкий набір алгоритмів машинного навчання включаючи методи класифікації регресії та кластеризації. Крім того система містить модулі автоматичного підбору моделей. У результаті користувач може отримати готову модель без глибокого занурення у технічні деталі алгоритмів.

Окрему категорію становлять хмарні платформи аналізу даних [10]. Вони забезпечують доступ до значних обчислювальних ресурсів і дозволяють масштабувати аналітичні задачі. Багато таких систем інтегруються з інфраструктурою зберігання даних та сервісами обробки великих масивів інформації. У таких платформах користувач зазвичай завантажує набір даних визначає тип задачі а далі система автоматично виконує підбір моделей.

У хмарних середовищах особливо важливою є можливість паралельного навчання моделей. Система може одночасно запускати

десятки експериментів з різними конфігураціями алгоритмів. Кожна модель оцінюється за заданими метриками якості. Після завершення обчислень формується рейтинг моделей де обирається найбільш ефективна конфігурація.

Важливою технічною особливістю сучасних платформ є інтеграція з системами обробки великих даних. Наприклад багато інструментів підтримують роботу з розподіленими обчислювальними фреймворками. Це дозволяє обробляти великі обсяги інформації які не можуть бути ефективно оброблені на одному комп'ютері. У таких системах дані розподіляються між кількома вузлами обчислювального кластера. Кожен вузол виконує частину обчислень після чого результати агрегуються.

Ще одним технічним аспектом є інтеграція платформ з мовами програмування що використовуються у сфері аналізу даних. Багато інструментів мають інтерфейси для Python або R. Це дозволяє поєднувати автоматизовані підходи з традиційним програмуванням. У практиці часто використовується гібридний підхід коли частина аналітичного процесу виконується автоматично а складні або специфічні етапи реалізуються вручну.

Попри значні можливості сучасних платформ автоматизації аналізу даних існують і певні обмеження. Зокрема автоматичні системи не завжди враховують специфіку предметної галузі або особливості структури даних. Іноді це призводить до вибору моделей які не є оптимальними з точки зору інтерпретованості або обчислювальної складності [11], [12]. Крім того деякі платформи мають обмеження щодо гнучкості конфігурації моделей.

Таким чином сучасні інструменти автоматизації аналізу даних формують технологічну основу інтелектуальних аналітичних систем. Вони дозволяють автоматично будувати моделі машинного навчання оптимізувати їх параметри та оцінювати результати. Завдяки цьому значно спрощується процес роботи з даними а аналітичні методи стають доступнішими для широкого кола користувачів. Разом з тим ефективне

використання таких платформ потребує розуміння їх технічних можливостей та обмежень.

1.5 Постановка задачі дослідження

У сучасних інформаційних системах аналіз даних відіграє важливу роль у підтримці прийняття рішень. Зі зростанням обсягів інформації та складності її структури традиційні підходи до аналітики поступово втрачають ефективність. Процес побудови моделей машинного навчання включає велику кількість етапів: підготовку даних трансформацію ознак вибір алгоритму налаштування параметрів та оцінювання результатів. У реальних задачах ці етапи часто виконуються багаторазово оскільки необхідно протестувати різні конфігурації моделей. У результаті процес аналізу даних стає трудомістким і потребує значних обчислювальних ресурсів.

Актуальність дослідження пов’язана з необхідністю підвищення ефективності аналітичних процесів. Сучасні організації працюють з великими масивами даних які надходять з різних джерел і швидко змінюються. Ручне налаштування моделей у таких умовах є складним і не завжди забезпечує оптимальні результати. Саме тому виникає потреба у програмних системах що здатні автоматично формувати моделі аналізу даних та адаптувати їх до нових наборів інформації. Автоматизація дозволяє скоротити час побудови аналітичних рішень і зробити методи машинного навчання більш доступними.

Для вирішення цієї задачі у сучасних дослідженнях застосовується декілька підходів. Найбільш поширеним є використання систем автоматизованого машинного навчання. У межах таких систем виконуються автоматичний підбір алгоритмів моделювання та оптимізація їх параметрів. Для цього застосовуються різні методи оптимізації зокрема випадковий пошук байєсівська оптимізація або еволюційні алгоритми. Вони дозволяють

досліджувати великий простір можливих конфігурацій моделей і знаходити ефективні рішення.

Іншим напрямом є використання аналітичних конвеєрів. У таких системах процес аналізу даних представлений як послідовність етапів обробки інформації. До них належать підготовка даних формування ознак навчання моделей та оцінювання результатів. Конвеєрна організація дозволяє стандартизувати аналітичний процес та автоматизувати його виконання у програмних системах.

Також у сучасних дослідженнях застосовуються методи метанавчання. У цьому підході система використовує інформацію про результати попередніх задач аналізу даних. Накопичений досвід дозволяє швидше визначати ефективні алгоритми для нових наборів даних та зменшує кількість необхідних експериментів.

Разом з тим існуючі підходи до автоматизації аналізу даних мають певні недоліки. Частина систем AutoML орієнтована лише на оптимізацію окремих моделей і не забезпечує повної автоматизації всього аналітичного процесу. У багатьох випадках користувач повинен вручну виконувати підготовку даних або визначати структуру аналітичного конвеєра. Крім того деякі платформи мають обмежену адаптивність до різних типів даних та підтримують лише певні алгоритми машинного навчання. Також існує проблема інтеграції різних етапів аналізу у межах єдиного програмного середовища оскільки для підготовки даних навчання моделей та оцінювання результатів часто використовуються різні інструменти.

У зв’язку з цим виникає необхідність розроблення програмної системи яка забезпечує комплексну автоматизацію аналізу даних. Така система повинна інтегрувати основні етапи аналітичного процесу у межах єдиного програмного середовища та забезпечувати автоматичний вибір моделей і параметрів.

Для реалізації цього підходу у межах дослідження необхідно розв’язати такі задачі:

– проаналізувати сучасні підходи до автоматизації аналізу даних та визначити їх основні обмеження;

– розробити архітектуру програмної системи автоматизованого аналізу даних;

– реалізувати механізм автоматичного формування аналітичного конвеєра обробки даних;

– забезпечити автоматичний вибір моделей машинного навчання та оптимізацію їх параметрів;

– реалізувати модуль оцінювання якості моделей та порівняння результатів.

2 МЕТОДИ ТА МОДЕЛІ АВТОМАТИЗОВАНОГО АНАЛІЗУ ДАНИХ

2.1 Задача автоматизації аналізу даних в рамках програмної системи

У сучасних інформаційних системах аналіз даних поступово переходить від ручного виконання до автоматизованих процедур, що реалізуються засобами програмного забезпечення. Це пов’язано з постійним зростанням обсягів даних, складністю їх структури та необхідністю швидкого отримання аналітичних результатів. У таких умовах традиційні підходи до аналізу даних, які передбачають значну участь аналітика на кожному етапі побудови моделей, стають малоефективними. Відповідно виникає потреба у створенні програмних систем, здатних автоматично виконувати значну частину аналітичних операцій.

Автоматизація аналізу даних у програмних системах передбачає реалізацію механізмів, які дозволяють системі самостійно виконувати послідовність операцій, пов’язаних із підготовкою даних, вибором методів аналізу, навчанням моделей та оцінюванням результатів. Такий підхід дозволяє суттєво зменшити залежність процесу аналізу від людського фактору та зробити використання аналітичних інструментів доступнішим для користувачів, які не мають глибоких знань у галузі машинного навчання.

У межах програмної системи аналіз даних розглядається як сукупність взаємопов’язаних процесів, які виконуються над певним набором даних [13]. Набір даних у цьому контексті являє собою структуровану множину записів, де кожен запис відповідає окремому об’єкту спостереження та містить набір характеристик, що описують цей об’єкт. Ці характеристики можуть мати різний тип: числовий, категоріальний або порядковий. У багатьох випадках набір даних також містить цільову змінну, яка визначає результат, що має бути спрогнозований або визначений у процесі аналізу.

Особливістю задач автоматизованого аналізу даних є необхідність роботи з наборами даних різної структури та якості. Дані, що використовуються у практичних задачах, часто містять пропущені значення, нерівномірний розподіл ознак або надлишкову інформацію. Крім того, різні набори даних можуть вимагати застосування різних алгоритмів аналізу. У зв’язку з цим програмна система повинна мати можливість адаптувати процес аналізу залежно від характеристик отриманих даних.

У програмних системах аналізу даних процес побудови моделей зазвичай організовується у вигляді послідовності операцій, що утворюють аналітичний конвеєр. Такий конвеєр включає декілька основних етапів: отримання даних, попередню обробку, вибір моделі, навчання алгоритму та оцінювання результатів. Кожен із цих етапів виконує певну функцію у процесі аналізу і може бути реалізований у вигляді окремого програмного модуля.

Першим етапом є отримання та первинний аналіз даних. На цьому етапі система отримує набір даних з джерела зберігання або зовнішнього середовища та визначає його основні характеристики [13]. Це дозволяє сформувати уявлення про структуру даних, типи змінних та їх розподіл. У деяких випадках уже на цьому етапі можуть бути виявлені проблеми, пов’язані з якістю даних, наприклад пропущені значення або аномальні записи.

Після цього виконується етап попередньої обробки даних. Його метою є підготовка даних до подальшого використання у моделях машинного навчання. На практиці це може включати очищення даних, перетворення ознак, приведення змінних до єдиного масштабу або видалення надлишкових характеристик. Без виконання цих операцій більшість алгоритмів машинного навчання не здатна забезпечити стабільні результати.

Наступним етапом є вибір методів аналізу даних. У програмних системах цей процес може виконуватися автоматично на основі

характеристик набору даних. Наприклад, якщо у даних присутня цільова змінна, система може використовувати алгоритми прогнозування або класифікації. Якщо ж цільова змінна відсутня, застосовуються методи, спрямовані на виявлення структури даних.

Після вибору алгоритму виконується навчання моделі. У процесі навчання модель аналізує структуру даних та формує залежності між ознаками, які дозволяють виконувати прогнозування або класифікацію нових об’єктів. У сучасних програмних системах цей етап часто супроводжується автоматичним налаштуванням параметрів алгоритму, що дозволяє підвищити точність отриманих результатів.

Завершальним етапом аналізу є оцінювання ефективності побудованої моделі. Для цього використовуються спеціальні критерії оцінювання, які дозволяють визначити, наскільки добре модель працює з даними. На основі результатів такого оцінювання система може приймати рішення щодо вибору найбільш ефективної моделі або повторного навчання з іншими параметрами.

Таким чином, задача автоматизації аналізу даних у програмних системах полягає у створенні механізму, який забезпечує виконання всіх основних етапів аналітичного процесу у автоматичному режимі. Реалізація такого механізму дозволяє значно спростити процес аналізу даних та зробити його більш ефективним [14]. Крім того, автоматизація аналітичних процедур створює передумови для використання методів машинного навчання у широкому спектрі прикладних задач, де ручне налаштування моделей було б надто складним або ресурсомістким.

2.2 Вибір моделей машинного навчання для автоматизованого аналізу даних

Реалізація автоматизованого аналізу даних у програмній системі передбачає наявність набору моделей, які можуть використовуватися

залежно від структури вхідного набору даних та типу задачі, що виникає під час обробки інформації. У цьому контексті важливим є не просто перелік алгоритмів, а формування такого набору моделей, який забезпечує достатню універсальність системи та водночас не ускладнює аналітичний конвеєр зайвими обчислювальними витратами. Якщо сказати простіше – система повинна мати кілька перевірених інструментів, а не десятки алгоритмів, половина з яких у реальній роботі взагалі не знадобиться.

Вибір моделей у межах даної роботи здійснюється за декількома практичними критеріями. По-перше, алгоритми повинні демонструвати стабільні результати на різних наборах даних без складних процедур налаштування. По-друге, їх параметри повинні бути придатними до автоматичної оптимізації, оскільки система не передбачає ручного втручання користувача [14]. По-третє, алгоритми повинні коректно працювати з наборами даних середньої розмірності, що характерно для більшості прикладних задач аналізу даних у програмних системах.

Для задач класифікації у системі використовується обмежений, але достатньо ефективний набір алгоритмів. Першою моделлю є логістична регресія, яка у цьому випадку виконує роль базового алгоритму. Її використання обумовлено високою швидкістю навчання та стабільністю роботи на наборах даних із відносно невеликою кількістю ознак. У межах автоматизованого аналізу така модель фактично виконує функцію початкової оцінки – своєрідного контрольного рівня, з яким надалі порівнюються результати більш складних алгоритмів.

Другу групу моделей становлять дерева рішень. Їх включення до системи пояснюється здатністю працювати з даними різної природи без складних процедур трансформації [15]. Для автоматизованих систем це важливо, оскільки структура даних може змінюватися від одного набору до іншого. Дерева рішень формують послідовність розділень даних за ознаками, і така структура моделі досить добре адаптується до різних типів задач. Разом з тим окреме дерево інколи поводиться трохи нестабільно,

якщо дані змінюються навіть незначно, тому у практичних реалізаціях його часто використовують у складі ансамблевих методів.

Саме з цієї причини до системи включено алгоритм випадкового лісу. Його використання дозволяє компенсувати нестабільність окремих дерев і отримати більш надійні результати класифікації. У рамках автоматизованого аналізу випадковий ліс виконує роль універсальної моделі, яка добре працює у ситуаціях, коли структура даних не є заздалегідь відомою. Іноді, до речі, саме ця модель дає найкращі результати навіть без складної оптимізації параметрів.

Окрему групу моделей становлять алгоритми градієнтного бустингу. Вони включені до системи з практичних міркувань – такі алгоритми часто демонструють високу точність на різних наборах даних [15]. У межах аналітичного конвеєра бустинг використовується як одна з моделей, що здатна враховувати складні взаємозв’язки між ознаками. При цьому алгоритм поступово формує послідовність моделей, кожна з яких коригує помилки попередніх. Такий підхід іноді працює дуже ефективно, хоча й потребує більше обчислювальних ресурсів.

Для задач регресії система використовує моделі, які здатні прогнозувати числові значення змінних. У цьому випадку основну роль відіграють лінійні моделі та ансамблеві алгоритми. Лінійні підходи застосовуються як базовий метод прогнозування, оскільки вони швидко навчаються та дозволяють отримати початкову оцінку залежностей у даних. Ансамблеві методи, у свою чергу, використовуються у тих випадках, коли залежності між ознаками мають більш складний характер і прості моделі не забезпечують достатньої точності.

У ситуаціях, коли набір даних не містить цільової змінної, система переходить до використання алгоритмів кластеризації. Основна функція таких алгоритмів полягає у виявленні структури даних та групуванні об’єктів за подібністю. У межах автоматизованого аналізу це дозволяє отримати початкове уявлення про структуру набору даних і, певною мірою,

навіть підказує подальший напрям аналізу. Інколи кластеризація працює як своєрідний дослідницький інструмент – вона ніби розкладає дані на декілька природних груп, і система вже працює з ними далі.

Таким чином, набір моделей, що використовується у системі автоматизованого аналізу даних, формується з алгоритмів, які демонструють стабільну роботу на різних типах наборів даних та допускають автоматичне налаштування параметрів. Кожен із розглянутих алгоритмів виконує конкретну функцію у межах аналітичного конвеєра: одні забезпечують швидку базову оцінку результатів, інші дозволяють підвищити точність прогнозування, а деякі – виявити структуру у даних, коли прогнозування взагалі неможливе. Саме така комбінація моделей створює основу для реалізації автоматизованого аналізу даних у програмній системі.

2.3 Методи автоматичного вибору моделей та налаштування параметрів

Однією з ключових проблем побудови систем автоматизованого аналізу даних є вибір конфігурації моделей машинного навчання без участі спеціаліста. У реальних аналітичних задачах ефективність моделі визначається не лише типом алгоритму, але і набором його параметрів. Значення цих параметрів можуть суттєво впливати на результат прогнозування, стабільність моделі та її здатність узагальнювати структуру даних. Тому процес побудови моделей у автоматизованих системах фактично перетворюється на задачу пошуку оптимальної конфігурації алгоритмів.

У класичних підходах підбір параметрів моделей здійснюється вручну або за допомогою стандартних процедур оптимізації. Найбільш поширеним підходом є повний перебір конфігурацій параметрів. У межах такого методу система формує всі можливі комбінації значень параметрів у заданому

діапазоні та послідовно перевіряє їх під час навчання моделі. Попри очевидну простоту, цей підхід має суттєвий недолік – кількість можливих конфігурацій швидко зростає разом із кількістю параметрів. У результаті обчислювальні витрати можуть ставати непропорційно великими, що обмежує практичне використання такого методу у системах автоматизованого аналізу.

Іншим підходом є випадковий пошук конфігурацій параметрів. У цьому випадку система формує випадкові комбінації параметрів у межах визначеного простору значень та оцінює ефективність моделей, що відповідають цим конфігураціям. На відміну від повного перебору, такий підхід дозволяє досліджувати простір параметрів більш економно, не перевіряючи всі можливі комбінації. У багатьох практичних задачах випадковий пошук демонструє результати, порівнянні з повним перебором, але при значно менших обчислювальних витратах.

Разом із тим використання стандартного випадкового пошуку має певні обмеження. У класичній реалізації всі моделі отримують однакову кількість експериментів та досліджують однаковий простір параметрів, незалежно від складності алгоритму або характеристик набору даних. Такий підхід не завжди є ефективним, оскільки різні алгоритми потребують різної кількості конфігурацій для отримання стабільного результату. Іноді система витрачає ресурси на перевірку конфігурацій, які з самого початку мають низьку ймовірність бути ефективними.

У межах даної роботи пропонується адаптивний підхід до автоматичного вибору моделей та налаштування параметрів. Основна ідея підходу полягає у тому, що процедура підбору параметрів враховує характеристики набору даних та складність алгоритму машинного навчання. Іншими словами, система не застосовує однакову процедуру оптимізації до всіх моделей, а формує конфігурації параметрів залежно від властивостей задачі.

На першому етапі система аналізує структуру набору даних. Визначаються такі характеристики, як кількість об’єктів, кількість ознак, типи змінних та наявність пропущених значень. Ці параметри дозволяють оцінити складність задачі та приблизно визначити, які моделі можуть бути більш ефективними. Наприклад, при невеликій кількості спостережень система обмежує використання складних моделей або зменшує кількість експериментів під час налаштування параметрів.

На другому етапі формується набір кандидатних моделей, що можуть бути використані для аналізу даних. Кожна модель отримує власний простір параметрів, який визначає допустимі значення внутрішніх налаштувань алгоритму. Простір параметрів формується таким чином, щоб охоплювати найбільш типові конфігурації алгоритмів, але водночас не створювати надмірної кількості можливих комбінацій.

На третьому етапі виконується процедура адаптивного випадкового пошуку параметрів. Для кожної моделі генерується набір випадкових конфігурацій параметрів, після чого виконується навчання моделей та оцінювання їх результатів. Кількість конфігурацій параметрів може відрізнятися для різних алгоритмів. Прості моделі перевіряються за меншою кількістю конфігурацій, тоді як більш складні алгоритми отримують більший простір експериментів.

Після завершення процесу навчання система порівнює результати моделей за обраними метриками якості. Для кожного алгоритму визначається найкраща конфігурація параметрів, після чого виконується порівняння моделей між собою. У результаті система обирає модель, що демонструє найкращі показники ефективності для даного набору даних.

Таким чином, запропонований підхід дозволяє поєднати автоматичний вибір моделей та налаштування їх параметрів у межах єдиного алгоритмічного процесу. Використання адаптивного механізму пошуку конфігурацій дозволяє зменшити обчислювальні витрати та підвищити ефективність автоматизованого аналізу даних. Система

фактично виконує ту ж саму роботу, яку у традиційному аналізі даних виконує спеціаліст з машинного навчання, але робить це алгоритмічно, послідовно і без зайвих експериментів.

2.4 Методи підготовки даних у системі автоматизованого аналізу

Ефективність роботи моделей машинного навчання у значній мірі залежить від якості вхідних даних. Навіть добре підібраний алгоритм може демонструвати нестабільні результати, якщо дані містять пропуски, некоректні значення або ознаки, що не несуть інформаційного навантаження. У системах автоматизованого аналізу даних проблема підготовки даних має особливе значення, оскільки більшість етапів обробки повинні виконуватися без ручного втручання користувача. Фактично, система повинна самостійно підготувати дані до подальшого використання моделями машинного навчання.

У межах даної роботи підготовка даних реалізується як послідовний конвеєр обробки, що виконує декілька етапів трансформації набору даних. Кожен етап спрямований на усунення певного типу проблем, які можуть виникати під час аналізу інформації. Такий підхід дозволяє забезпечити стабільну роботу моделей та зменшити залежність результатів від початкової структури даних.

Першим етапом конвеєра є аналіз структури набору даних. На цьому етапі система визначає типи змінних, кількість ознак, наявність пропущених значень та загальну структуру таблиці даних. Ця інформація використовується для визначення подальших процедур обробки. Наприклад, система повинна визначити, які змінні є числовими, які мають категоріальний характер, а які можуть бути інтерпретовані як ідентифікатори або службові поля. Такий попередній аналіз дозволяє уникнути некоректних трансформацій, коли, наприклад, текстові змінні випадково обробляються як числові.

Наступним етапом є обробка пропущених значень. У реальних наборах даних пропуски зустрічаються досить часто, і їх ігнорування може призвести до некоректної роботи моделей. У системі автоматизованого аналізу використовується декілька стратегій заповнення пропущених значень залежно від типу змінної. Для числових змінних застосовується заповнення статистичними характеристиками розподілу даних. Для категоріальних змінних використовується заміна пропущених значень найбільш поширеним значенням або введення окремої категорії, що позначає відсутність інформації. Подібна процедура дозволяє зберегти всі записи набору даних та уникнути втрати інформації.

Після обробки пропущених значень виконується перетворення типів змінних та кодування категоріальних ознак. Більшість алгоритмів машинного навчання працюють лише з числовими даними, тому категоріальні змінні повинні бути перетворені у відповідний формат. У системі використовується числове представлення категорій, що дозволяє інтегрувати такі змінні до процесу навчання моделей. Процедура кодування виконується автоматично для всіх змінних, які були ідентифіковані як категоріальні під час аналізу структури даних.

Далі виконується масштабування числових ознак. У багатьох алгоритмах машинного навчання різні масштаби значень змінних можуть впливати на результати навчання моделей. Наприклад, змінні з великим діапазоном значень можуть домінувати у процесі побудови моделі. Для усунення цієї проблеми у системі застосовується нормалізація або стандартизація числових ознак. Вибір конкретного методу масштабування залежить від характеристик даних та типу алгоритму, який використовується для аналізу.

Окремим етапом підготовки даних є відбір інформативних ознак. У реальних наборах даних часто присутні змінні, що мають низьку інформаційну цінність або дублюють інформацію, яка вже міститься в інших ознаках. Наявність таких змінних може ускладнювати навчання

моделей та знижувати якість прогнозування. У системі автоматизованого аналізу застосовується процедура автоматичного відбору ознак, яка дозволяє зменшити розмірність простору даних та залишити лише найбільш інформативні змінні. Такий підхід сприяє підвищенню стабільності моделей та зменшенню обчислювальних витрат.

Завершальним етапом конвеєра є формування фінального набору ознак, який передається до модуля навчання моделей машинного навчання. На цьому етапі всі трансформації даних вже виконані, структура набору даних є узгодженою, а змінні представлені у форматі, придатному для використання алгоритмами машинного навчання. Фактично, після цього етапу дані стають готовими до застосування моделей класифікації, регресії або кластеризації.

Таким чином, підготовка даних у системі автоматизованого аналізу реалізується у вигляді послідовного конвеєра трансформацій, що дозволяє автоматично усувати типові проблеми наборів даних та забезпечувати їх коректне використання у моделях машинного навчання. Використання такого підходу дозволяє значно спростити процес аналізу даних та зробити систему більш незалежною від ручного втручання аналітика. Іноді, власне, саме цей етап визначає успіх подальшого аналізу бо навіть найкраща модель нічого не зробить з погано підготовленими даними.

2.5 Метрики оцінювання якості моделей

Після побудови моделей машинного навчання необхідним етапом автоматизованого аналізу даних є оцінювання їх ефективності. У системах автоматичного аналізу цей етап виконує особливу роль, оскільки саме результати оцінювання визначають, яка модель буде обрана системою для подальшого використання. Іншими словами, механізм оцінювання фактично виконує функцію внутрішнього критерію прийняття рішень.

Система повинна мати чіткий спосіб визначити, яка модель працює краще, і зробити це без участі користувача.

У межах даної роботи оцінювання моделей здійснюється відповідно до типу задачі, що вирішується системою. Оскільки система може працювати з задачами класифікації, регресії та структурного аналізу даних, набір метрик оцінювання формується таким чином, щоб забезпечити коректне порівняння моделей у кожному з цих випадків. При цьому використовуються показники, що дозволяють отримати узагальнену оцінку якості моделі та водночас не створюють надмірної складності для автоматизованого аналізу.

Для задач класифікації у системі використовується метрика точності прогнозування, яка характеризує частку правильно класифікованих об’єктів серед усіх спостережень. Ця метрика дозволяє швидко оцінити загальну ефективність моделі та часто використовується як базовий показник якості класифікації. Разом із тим у деяких наборах даних розподіл класів може бути нерівномірним, і тоді використання лише точності може призводити до некоректних висновків.

З цієї причини система також враховує показники, що дозволяють більш детально оцінити результати класифікації. Зокрема аналізуються показники повноти та точності визначення класів, які характеризують здатність моделі коректно виявляти об’єкти певної категорії. Поєднання цих показників дозволяє отримати більш збалансовану оцінку ефективності моделі, особливо у випадках, коли класи розподілені нерівномірно.

Для задач регресії оцінювання моделей здійснюється за допомогою показників похибки прогнозування. Основною метрикою є середня абсолютна похибка, що відображає середню різницю між прогнозованими та фактичними значеннями змінної. Цей показник дозволяє оцінити точність прогнозування у зрозумілій інтерпретації, оскільки результат безпосередньо відображає величину відхилення прогнозу від реальних значень.

Додатково використовується показник середньоквадратичної похибки, який більш чутливий до значних відхилень прогнозів. Використання цього показника дозволяє виявити ситуації, коли модель демонструє прийнятну середню точність, але робить значні помилки для окремих спостережень. У системі автоматизованого аналізу обидва показники використовуються для порівняння результатів різних моделей та визначення найбільш ефективної конфігурації.

У випадку задач кластеризації оцінювання моделей здійснюється за допомогою показників, що характеризують якість формування груп об’єктів. Оскільки у таких задачах відсутня цільова змінна, оцінювання моделей ґрунтується на аналізі внутрішньої структури сформованих кластерів. Система оцінює ступінь компактності груп та відстань між ними, що дозволяє визначити, наскільки чітко алгоритм розділяє об’єкти набору даних.

У процесі автоматизованого аналізу даних оцінювання моделей виконується після кожного експерименту з конфігураціями параметрів. Для кожної моделі система зберігає значення відповідних метрик та формує узагальнений рейтинг результатів. Після завершення навчання всіх моделей виконується порівняння отриманих показників, і система обирає модель, що демонструє найкращі результати відповідно до визначених критеріїв.

Важливою особливістю системи є те, що вибір моделі здійснюється на основі єдиного механізму оцінювання. Це забезпечує узгодженість результатів та дозволяє коректно порівнювати різні алгоритми між собою. У результаті система не просто будує декілька моделей, а виконує їх систематичне порівняння, поступово визначаючи найбільш ефективний підхід для конкретного набору даних.

Таким чином, метрики оцінювання моделей відіграють ключову роль у процесі автоматизованого аналізу даних. Вони дозволяють системі об’єктивно оцінювати результати роботи алгоритмів та виконувати вибір моделі на основі формалізованих критеріїв. Саме завдяки цьому

автоматизований аналіз даних може виконуватися без прямого втручання спеціаліста, що є однією з основних цілей створення подібних програмних систем.

2.6 Запропонований алгоритм автоматизованого аналізу даних

Розглянуті у попередніх підрозділах методи машинного навчання, підходи до налаштування параметрів та процедури підготовки даних створюють основу для побудови системи автоматизованого аналізу даних. Проте окреме використання цих компонентів не дозволяє отримати завершений інструмент аналізу інформації. Для практичного застосування необхідно об’єднати їх у єдиний алгоритмічний процес, який буде виконувати всі етапи аналізу послідовно та без участі спеціаліста. Саме тому у межах даної роботи запропоновано алгоритм автоматизованого аналізу даних, що інтегрує процедури підготовки даних, вибору моделей та їх оптимізації.

Запропонований алгоритм базується на принципі поетапної обробки набору даних. Кожен етап виконує окрему функцію у межах аналітичного процесу та передає результати наступному етапу обробки. Така організація дозволяє реалізувати аналіз даних у вигляді програмного конвеєра, де всі операції виконуються у визначеній послідовності. У результаті система здатна виконувати аналіз наборів даних різної структури без попереднього налаштування.

Першим етапом алгоритму є аналіз структури вхідного набору даних. На цьому етапі система визначає основні характеристики таблиці даних: кількість спостережень, кількість змінних, типи ознак та наявність пропущених значень. Цей етап виконує роль початкової діагностики набору даних. Фактично система намагається зрозуміти, з яким типом інформації вона має справу, перш ніж переходити до наступних процедур обробки.

Після визначення структури даних запускається конвеєр підготовки даних. У межах цього етапу виконуються процедури очищення даних, обробки пропущених значень, кодування категоріальних змінних та масштабування числових ознак. Додатково виконується процедура відбору інформативних ознак, що дозволяє зменшити розмірність простору даних та підвищити стабільність подальшого навчання моделей. Результатом цього етапу є набір даних, підготовлений для використання алгоритмами машинного навчання.

Наступним етапом алгоритму є визначення типу аналітичної задачі. Система аналізує структуру набору даних та визначає, чи присутня у ньому цільова змінна. Якщо така змінна існує, система переходить до задач прогнозування, які можуть бути представлені у вигляді задач класифікації або регресії. Якщо цільова змінна відсутня, система застосовує алгоритми структурного аналізу даних, зокрема методи кластеризації.

Після визначення типу задачі формується набір моделей-кандидатів, які можуть бути використані для аналізу даних. Для кожної моделі визначається простір параметрів, що буде використовуватися під час оптимізації. При цьому система враховує складність алгоритму та характеристики набору даних. У деяких випадках складні моделі можуть отримувати ширший простір параметрів, тоді як для простіших алгоритмів використовується обмежена кількість конфігурацій.

Далі запускається процедура адаптивного підбору параметрів моделей. У межах цього процесу система генерує випадкові конфігурації параметрів для кожної моделі та виконує навчання моделей з використанням відповідних конфігурацій. Кількість експериментів для різних алгоритмів може відрізнятися залежно від їх складності. Такий підхід дозволяє більш ефективно використовувати обчислювальні ресурси та уникнути надмірного перебору параметрів.

Після завершення навчання моделей виконується етап оцінювання результатів. Для кожної моделі обчислюються відповідні метрики якості,

що визначають ефективність прогнозування або структурування даних. На основі отриманих значень система формує рейтинг моделей та визначає конфігурацію, що демонструє найкращі результати.

Завершальним етапом алгоритму є вибір фінальної моделі та формування результатів аналізу. Обрана модель використовується для генерації прогнозів або структурування даних, після чого результати передаються до користувацького інтерфейсу системи. Таким чином користувач отримує готові результати аналізу без необхідності виконувати складні процедури побудови моделей.

Запропонований алгоритм дозволяє інтегрувати основні етапи аналізу даних у межах єдиного програмного процесу. Його використання забезпечує автоматизацію ключових процедур машинного навчання та зменшує потребу у ручному налаштуванні моделей. Система фактично виконує повний цикл аналізу даних – від обробки вхідної інформації до побудови та вибору оптимальної моделі. Інколи, чесно кажучи, саме така автоматизація і є головною метою подібних систем: зробити складні аналітичні процедури доступними навіть для користувачів, які не мають глибоких знань у галузі машинного навчання.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗОВАНОГО АНАЛІЗУ ДАНИХ

У третьому розділі розглянуто розробку програмної системи автоматизованого аналізу табличних даних. Реалізація побудована як наскрізний пайплайн, що охоплює завантаження даних, їх профілювання, визначення типу задачі, попередню обробку, навчання моделей, оцінювання та формування звіту.

Система має модульну архітектуру, у якій кожен етап реалізований як окремий компонент із чітко визначеною функцією. Це забезпечує узгодженість роботи пайплайну та можливість подальшого розширення.

3.1 Загальна архітектура програмної системи

Розроблена програмна система реалізована як локальний застосунок для автоматизованого аналізу табличних даних. Основною ідеєю системи є побудова наскрізного пайплайну, який охоплює повний цикл роботи з набором даних: від завантаження файлу до формування узагальненого аналітичного звіту. Такий підхід дозволяє користувачу виконувати базовий аналіз даних, визначення типу задачі, попередню обробку, навчання моделей та отримання підсумкової інформації без необхідності самостійно налаштовувати кожен етап окремо.

Система працює у форматі локального вебзастосунку на основі Streamlit. Окремий серверний застосунок, зовнішній програмний інтерфейс або спеціалізований backend у межах реалізації не використовуються. Усі основні операції виконуються в одному процесі Python, а користувацький інтерфейс слугує зовнішнім рівнем взаємодії з аналітичним ядром системи. Такий варіант реалізації зменшує складність розгортання, не потребує окремої клієнт-серверної інфраструктури. Загальна архітектура системи представлена на рисунку 3.1.



Рисунок 3.1 – Загальна архітектура системи

Архітектура системи побудована за модульним принципом. Кожен основний етап пайплайну винесено в окремий логічний компонент, що відповідає за свою частину обробки. До таких компонентів належать модуль завантаження даних, модуль профілювання, модуль визначення типу задачі, модуль попередньої обробки, модуль вибору та навчання моделей, модуль оцінювання, модуль візуалізації та модуль генерації звіту. Користувацький інтерфейс взаємодіє з цими компонентами через центральний механізм запуску пайплайну, але самі аналітичні модулі не залежать від інтерфейсу.

Центральним елементом архітектури є оркестратор пайплайну. Він

відповідає не за виконання окремих аналітичних операцій, а за правильну послідовність їх виклику. Спочатку система завантажує табличний файл і перетворює його у внутрішнє представлення даних.

Далі виконується профілювання набору даних, після чого на основі отриманих метаданих визначається тип аналітичної задачі. Потім формується схема попередньої обробки, запускається навчання набору моделей, виконується їх порівняння та, за потреби, створюється HTML-звіт із графічними матеріалами. Завдяки такій організації кожен модуль залишається незалежним, а загальна логіка роботи системи зберігається в одному керуючому компоненті.

Передавання даних між етапами реалізовано через дві основні структури. Сам набір даних зберігається у вигляді таблиці pandas DataFrame, яка передається між модулями без зміни початкового змісту.

Друга структура містить метадані профілювання, тобто інформацію про типи колонок, кількість пропущених значень, унікальність, ролі ознак, кардинальність та інші характеристики.

Такі метадані використовуються наступними етапами для прийняття рішень щодо типу задачі, способів кодування ознак, обробки пропусків та побудови моделей. Такий підхід дозволяє уникнути дублювання логіки аналізу колонок у різних частинах системи.

Загальний потік даних у системі має послідовний характер. Потік обробки даних у системі. Користувач завантажує файл у форматі CSV або Excel, після чого система виконує його читання та базову перевірку. Далі формується профіль набору даних, визначається задача аналізу, будується preprocessing-пайплайн, запускається навчання моделей і формується таблиця порівняння.

Після цього дані можуть бути використані для побудови візуалізацій та генерації звіту. Отже, логіка системи може бути подана як ланцюг: файл, профілювання, визначення задачі, попередня обробка, моделювання, оцінювання, візуалізація та звіт.

Проміжні результати переважно зберігаються в оперативній пам’яті. Це стосується завантаженого набору даних, профілю колонок, параметрів задачі та таблиці моделей. Така організація спрощує реалізацію і відповідає локальному формату застосунку. Для зручності роботи інтерфейсу використовується механізм кешування завантажених даних та збереження стану виконання між оновленнями сторінки. Файлові артефакти створюються лише на етапі формування звіту, коли система генерує HTML-документ та пов’язані з ним графічні файли.

3.2 Реалізація користувацького інтерфейсу

Користувацький інтерфейс розробленої системи реалізовано у вигляді локального вебзастосунку з використанням бібліотеки Streamlit. Такий підхід дозволяє поєднати в одному середовищі як обчислювальну логіку, так і засоби взаємодії з користувачем без необхідності створення окремого клієнтського застосунку. Інтерфейс функціонує як тонкий шар над аналітичним ядром системи і виконує роль засобу ініціалізації та візуалізації процесу автоматизованого аналізу даних.

Логіка взаємодії з користувачем організована як послідовний сценарій, що відповідає етапам виконання пайплайну. На першому етапі користувач завантажує набір даних у форматі CSV або Excel – рисунок 3.2.

Після успішного завантаження система автоматично відображає базову інформацію про дані, зокрема розмірність таблиці та перші рядки, що дозволяє оцінити структуру набору перед подальшим аналізом – рисунок 3.3. Таким чином забезпечується початковий контроль якості вхідних даних без необхідності відкривати їх у сторонніх інструментах.

Після попереднього перегляду система формує припущення щодо цільової змінної та типу задачі. Це припущення використовується як значення за замовчуванням у відповідних елементах інтерфейсу, але користувач має можливість змінити його.

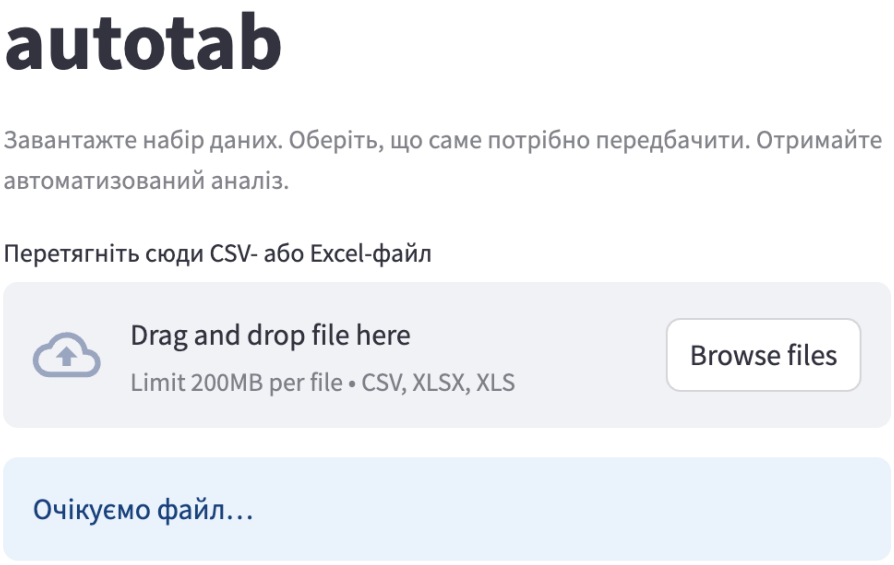
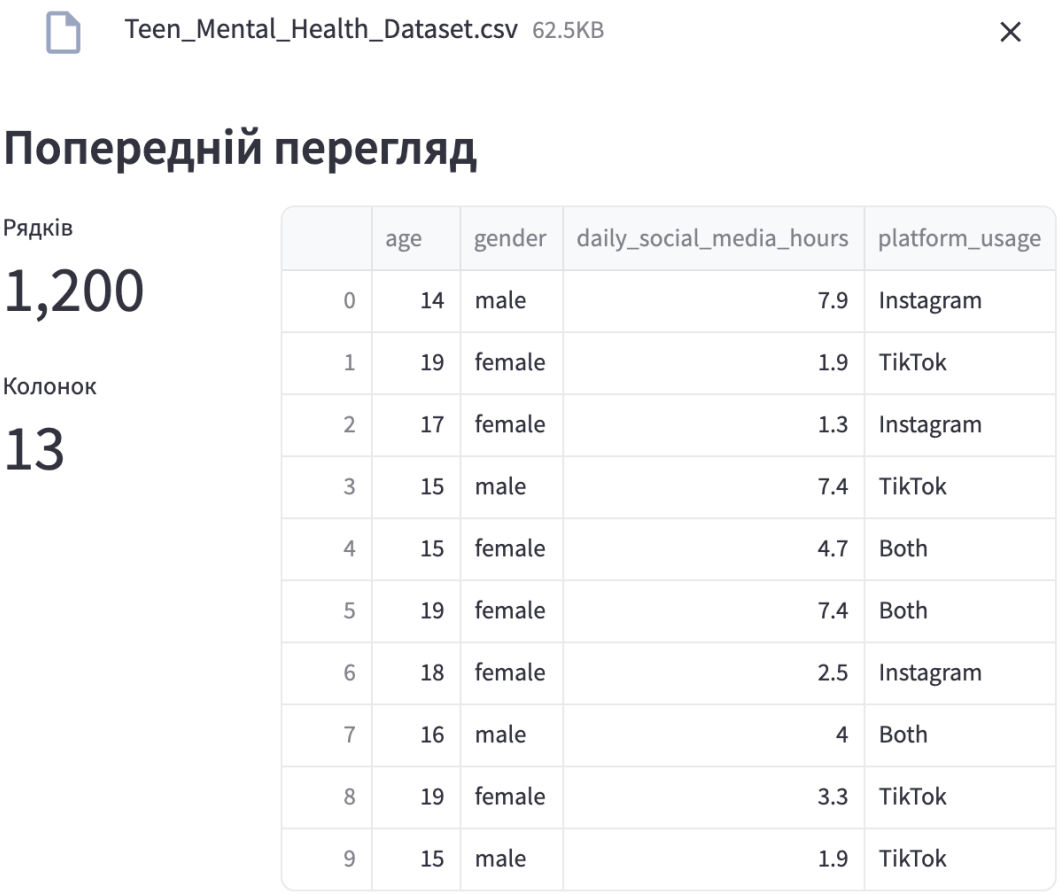


Рисунок 3.2 – Перший етап: завантажебння набору даних у відповідних форматах



	age	gender	daily_social_media_hours	platform_usage
0	14	male	7.9	Instagram
1	19	female	1.9	TikTok
2	17	female	1.3	Instagram
3	15	male	7.4	TikTok
4	15	female	4.7	Both
5	19	female	7.4	Both
6	18	female	2.5	Instagram
7	16	male	4	Both
8	19	female	3.3	TikTok
9	15	male	1.9	TikTok

Рисунок 3.3 – Відображення базових відомостей про завантажені дані

Для цього реалізовано елементи вибору, які дозволяють явно задати тип задачі та, за потреби, обрати іншу цільову змінну, що представлено на рисунку 3.4. Така організація забезпечує баланс між автоматизацією та контролем: у типовому випадку користувач може одразу запускати аналіз, але при необхідності має можливість втрутитися у процес.



Рисунок 3.4 – Рекомендація системи щодо типу задачі та цільової змінної; вибір типу задачі та цільової змінної

Запуск основного пайплайну здійснюється через окремий керуючий елемент інтерфейсу. Після його активації система ініціює виконання всіх етапів обробки даних у встановленій послідовності. Для відображення стану виконання використовується механізм поступового оновлення статусу, який інформує користувача про поточний етап обробки (рисунок 3.5).

Повідомлення про прогрес формуються у вигляді текстових описів і відображають ключові дії системи, що дозволяє зробити процес виконання більш прозорим і зрозумілим.

Результати роботи системи відображаються у вигляді кількох логічних блоків. До них належать таблиця моделей із їх характеристиками, візуалізації набору даних, а також детальний профіль колонок рисунки 3.6–3.11.

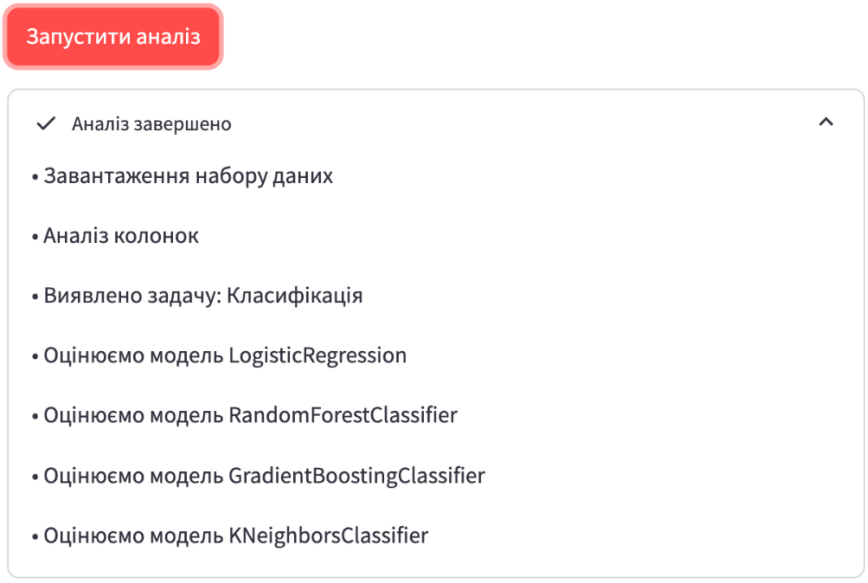


Рисунок 3.5 – Відображення про поточний етап обробки

Таблиця моделей дозволяє швидко оцінити доступні варіанти, тоді як графічні елементи сприяють інтерпретації структури даних. Профіль колонок подається у згорнутому вигляді, для уникникання перевантаження інтерфейсу та забезпечення доступ до детальної інформації при необхідності.

Однією з ключових особливостей інтерфейсу є мінімізація ручного налаштування. Більшість параметрів аналізу визначається автоматично на основі характеристик даних, а елементи керування мають попередньо встановлені значення. Це дозволяє використовувати систему без глибоких знань у галузі машинного навчання та скорочує час, необхідний для отримання результату. Водночас збережено базові можливості для ручного втручання, що підвищує гнучкість системи.

Для покращення зрозумілості інтерфейсу застосовано пояснювальні підписи та текстові підказки. Вони супроводжують основні елементи керування і пояснюють їх призначення, а також інформують користувача про зроблені системою припущення.

Результати

Задача	Найкраща модель	F1 (macro)
Класифік…	GradientB…	0.9828

Таблиця моделей

модель	accuracy	f1_macro	f1_weighted	roc_auc
GradientBoostingClassifier	0.9983 ± 0.0020	0.9828 ± 0.0212	0.9983 ± 0.0021	0.9954 ± 0.0076
LogisticRegression	0.9475 ± 0.0120	0.7135 ± 0.0570	0.9592 ± 0.0084	0.9827 ± 0.0113
RandomForestClassifier	0.9817 ± 0.0042	0.7065 ± 0.1129	0.9760 ± 0.0077	0.9996 ± 0.0009
KNeighborsClassifier	0.9733 ± 0.0020	0.5182 ± 0.0502	0.9622 ± 0.0040	0.8416 ± 0.0716

Рисунок 3.6 – Результат роботи системи: рекомендація кращого класифікатора та таблиця порівняння роботи моделей

Огляд набору даних

Баланс класів цілі

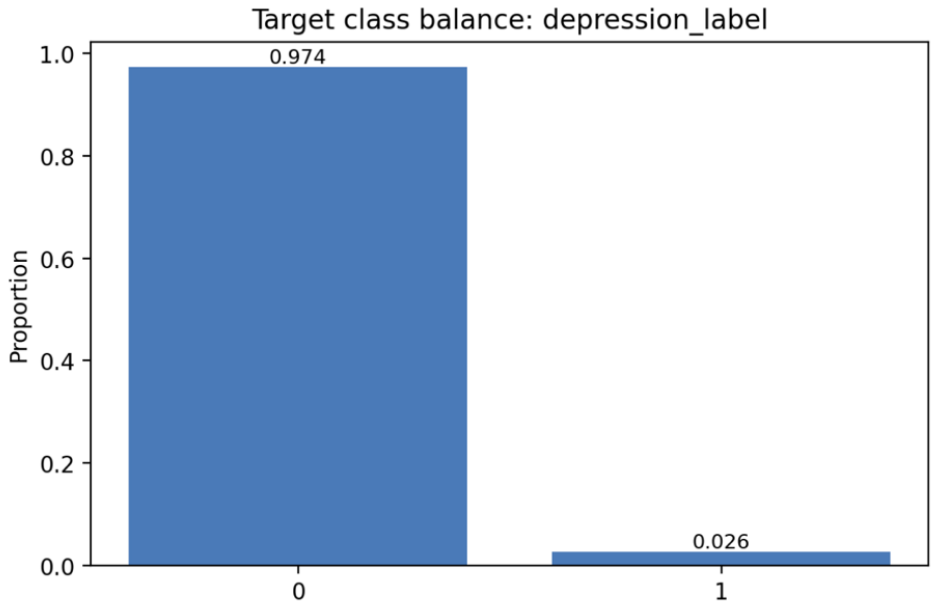


Рисунок 3.7 – Результат роботи системи: візуалізація балансу класів в цільовій змінній

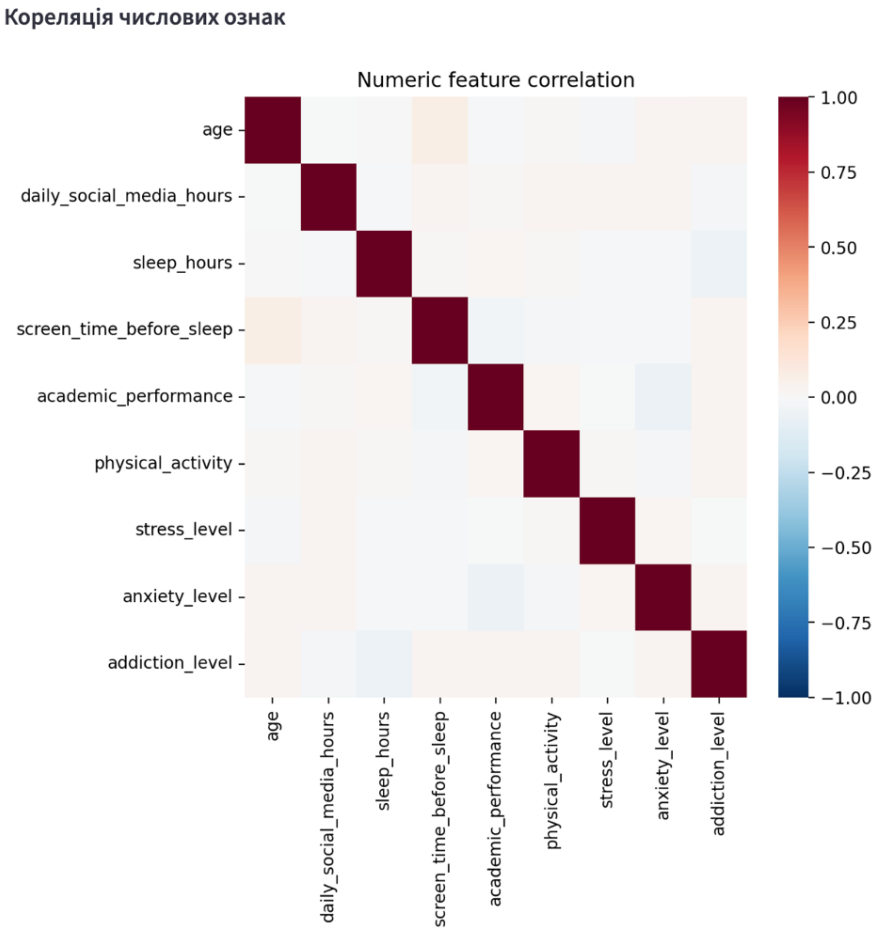


Рисунок 3.8 – Графік кореляції числових ознак

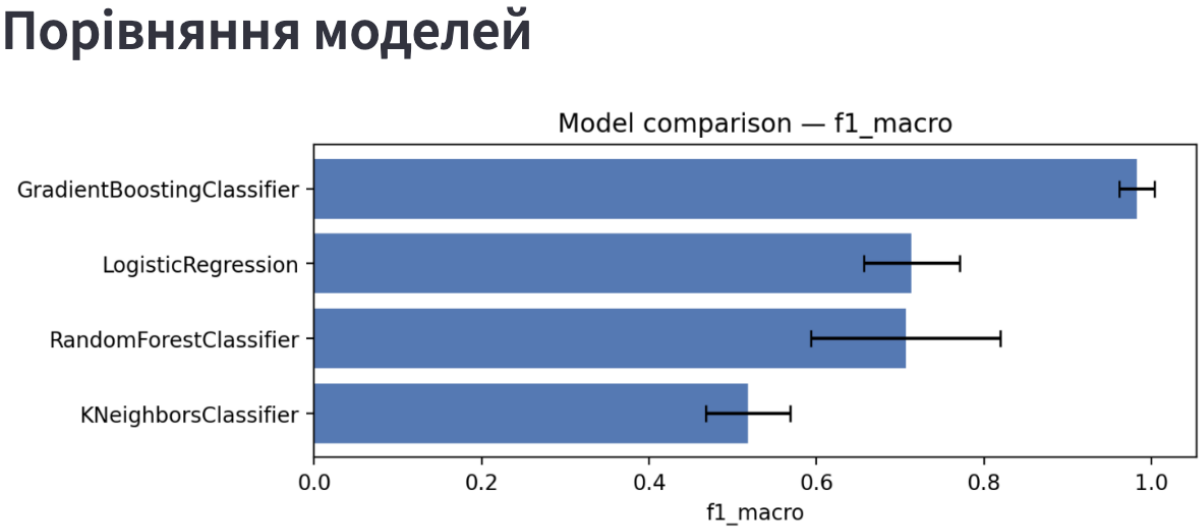


Рисунок 3.9 – Графік порівняння моделей по критерію f1 macro

Розподіли числових ознак

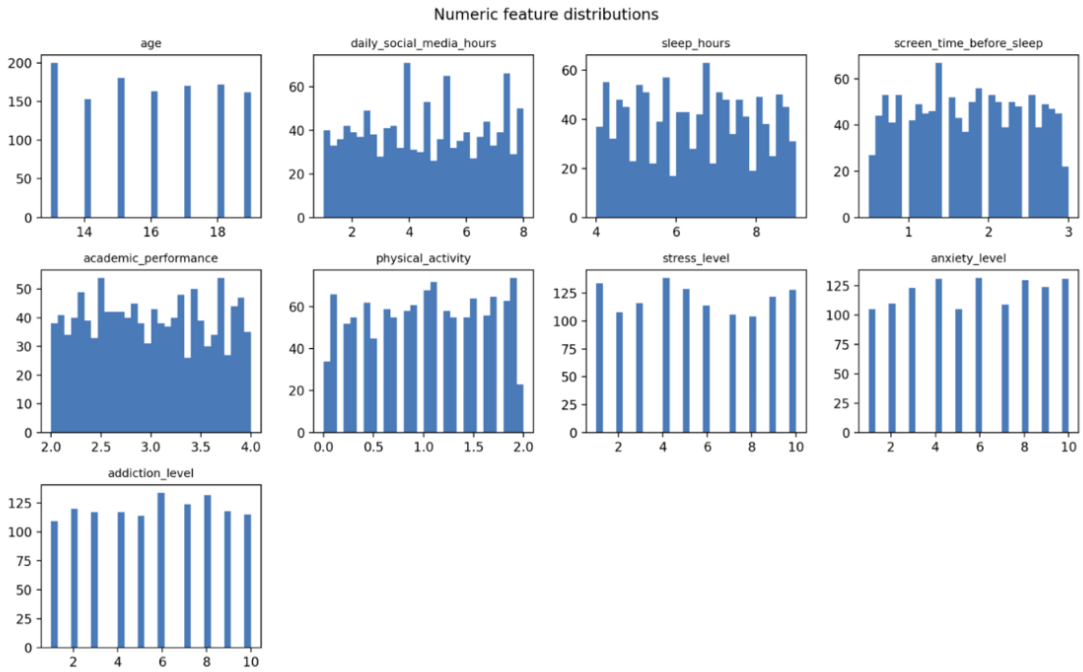


Рисунок 3.10 – Розподіл числових ознак

Профіль колонок

колонка	роль	тип	пропусків %	унікальних	потужність	кодування
daily_social_media_hours	ознака	float64	0.00	71	висока	числове
platform_usage	ознака	object	0.00	3	низька	one-hot
sleep_hours	ознака	float64	0.00	51	висока	числове
screen_time_before_sleep	ознака	float64	0.00	26	середня	числове
academic_performance	ознака	float64	0.00	201	висока	числове
physical_activity	ознака	float64	0.00	21	середня	числове
social_interaction_level	ознака	object	0.00	3	низька	one-hot
stress_level	ознака	int64	0.00	10	низька	числове
anxiety_level	ознака	int64	0.00	10	низька	числове
addiction_level	ознака	int64	0.00	10	низька	числове
depression_label	ціль	int64	0.00	2	бінарна	ціль

Рисунок 3.11 – Профіль колонок

Такий підхід дозволяє зменшити когнітивне навантаження та зробити взаємодію з системою більш інтуїтивною.

З технічної точки зору інтерфейс підтримує збереження стану між повторними оновленнями сторінки. Це дозволяє уникнути повторного виконання ресурсоємних операцій, зокрема завантаження та обробки даних, якщо вхідні параметри не змінювались.

Також передбачено можливість формування та завантаження звіту у вигляді окремого файлу, що розширює можливості використання системи поза межами інтерфейсу (рисунок 3.12).

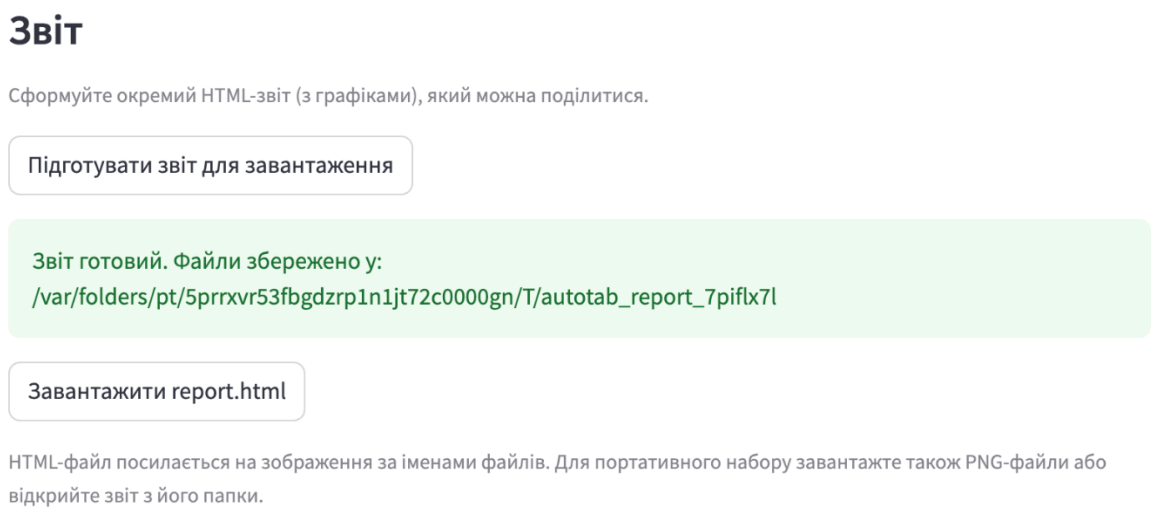


Рисунок 3.12 – Візуалізація формування звіту у форматі .html

Отже, реалізація користувацького інтерфейсу забезпечує цілісну інтеграцію з аналітичним ядром системи, підтримує послідовну логіку взаємодії та орієнтована на спрощення процесу аналізу даних без втрати функціональності.

3.3 Модуль профілювання даних

Модуль профілювання даних призначений для автоматичного аналізу

структури табличного набору даних та формування метаданих, які використовуються на всіх наступних етапах пайплайну. Його основна функція полягає не лише в описі даних, а у підготовці узгоджених характеристик, що безпосередньо впливають на вибір типу задачі, методів попередньої обробки та параметрів моделювання.

Вхідними даними модуля є табличний набір у вигляді DataFrame. На першому етапі виконується визначення типів колонок із подальшою бінарною класифікацією ознак на числові та нечислові. Такий поділ використовується як база для всіх наступних обчислень і рішень.

Для кожної колонки формується набір характеристик, що включає кількість пропущених значень, їх частку, кількість унікальних значень та базові статистичні показники. Для числових ознак додатково обчислюються показники розподілу, зокрема середнє значення, стандартне відхилення, квартилі та коефіцієнт асиметрії. На основі значення асиметрії визначається тип розподілу, що використовується надалі для вибору стратегії заповнення пропусків. Для категоріальних ознак визначаються найчастіші значення та частка рідкісних категорій, що дозволяє оцінити їх структуру та потенційні проблеми при кодуванні.

Окремим етапом є визначення ролі кожної колонки. На основі кількості унікальних значень, їх співвідношення до розміру вибірки та типу даних колонки автоматично класифікуються як ознаки, цільова змінна, ідентифікатори або константи. Такий підхід дозволяє ще на ранньому етапі виключити неінформативні або технічні поля з подальшого аналізу.

Модуль також виконує виявлення типових проблем у даних. До них належать наявність пропущених значень, висока кардинальність категоріальних змінних, значна асиметрія розподілу числових ознак та сильні кореляційні залежності між ними. Для кожної ознаки формується відповідний набір індикаторів, які використовуються як сигнали для прийняття рішень у наступних модулях. Зокрема, рівень кардинальності визначає спосіб кодування категоріальних змінних, а інформація про

розподіл – вибір методу імпутації.

Результатом роботи модуля є структурований набір метаданих, що містить як агреговану інформацію про набір даних, так і детальні характеристики кожної колонки. Процес формування метаданих представлений на рисунку 3.13. Ця структура передається до інших компонентів системи без змін і виступає єдиним джерелом інформації про властивості даних. Завдяки цьому забезпечується узгодженість рішень на різних етапах пайплайну та усувається необхідність повторного аналізу даних у кожному модулі окремо.

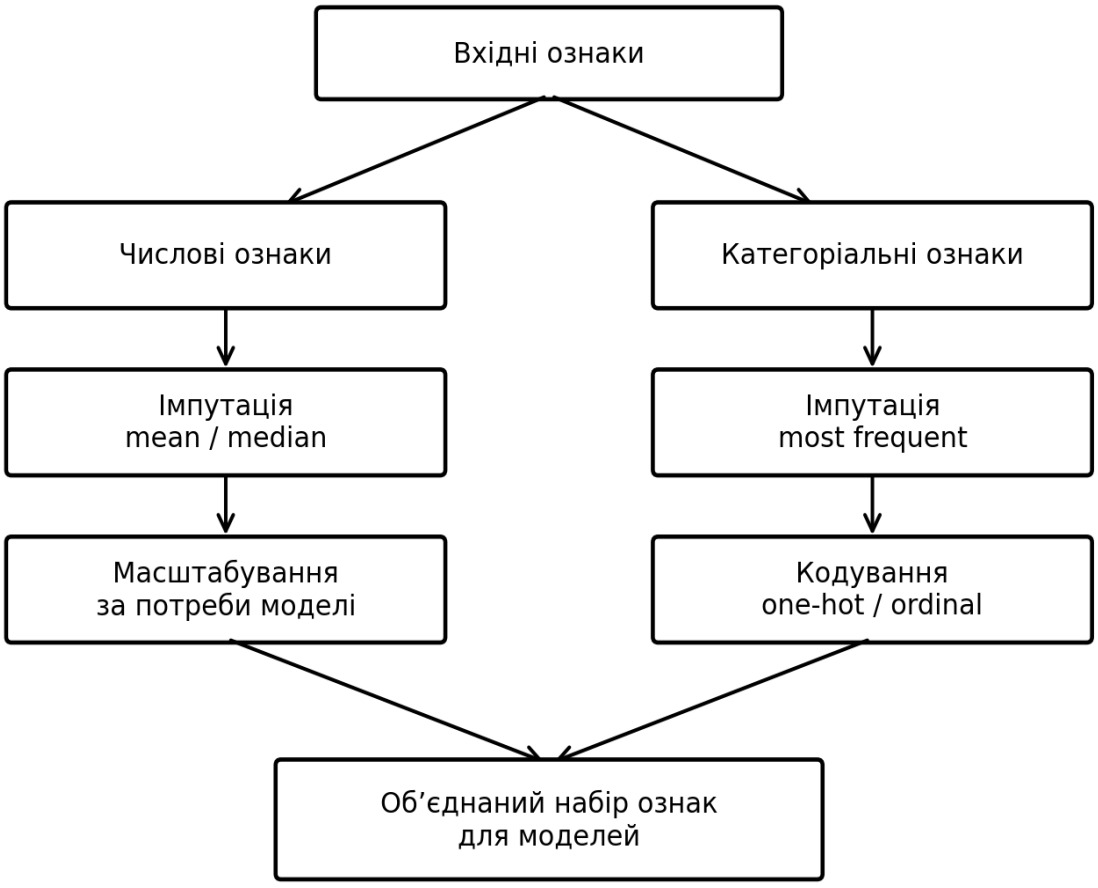


Рисунок 3.13 – Профілювання даних

Таким чином, модуль профілювання виконує роль базового аналітичного шару системи, який перетворює сирі дані у формалізоване

представлення їх характеристик, придатне для подальшої автоматизованої обробки.

3.4 Автоматичне визначення типу задачі

Модуль автоматичного визначення типу задачі призначений для класифікації вхідного набору даних за типом аналітичної задачі без участі користувача. Його функція полягає у виборі одного з трьох варіантів: класифікація, регресія або кластеризація. Результат цього етапу визначає подальшу конфігурацію пайплайну, зокрема вибір моделей, метрик оцінювання та структури обробки даних.

Визначення типу задачі реалізовано як окремий логічний компонент, який використовує як самі дані, так і метадані, отримані на етапі профілювання. Основними вхідними параметрами є наявність цільової змінної, її тип та кількість унікальних значень.

На першому кроці перевіряється наявність цільової змінної. Якщо вона відсутня, задача автоматично інтерпретується як кластеризація, оскільки в такому випадку відсутня змінна, яку необхідно передбачити. Це дозволяє системі коректно обробляти задачі без учителя.

У разі наявності цільової змінної визначальним фактором є її тип та кардинальність. Якщо змінна має числовий тип і характеризується великою кількістю унікальних значень, задача визначається як регресія. Такий підхід дозволяє відокремити безперервні цільові змінні від дискретних. Якщо ж кількість унікальних значень є обмеженою або змінна має категоріальний характер, задача класифікується як класифікація.

Для підвищення стабільності визначення використовується порогове значення кількості унікальних значень, яке дозволяє розрізняти регресійні та класифікаційні задачі у випадках числових змінних. Це запобігає некоректному визначенню типу задачі для змінних, що мають числове представлення, але фактично є категоріальними.

Окрім базової логіки визначення, система підтримує механізм попереднього припущення щодо типу задачі, який формується на етапі профілювання даних. Якщо таке припущення вже сформоване, воно використовується як пріоритетне, що дозволяє уникнути повторних обчислень та забезпечити узгодженість між модулями.

Результатом роботи модуля є символьне представлення типу задачі, яке передається до наступних компонентів системи. Це значення використовується для вибору набору моделей, конфігурації крос-валідації, визначення метрик оцінювання та налаштування візуалізації результатів.

Таким чином, модуль автоматичного визначення типу задачі забезпечує адаптивність системи до різних типів даних і дозволяє формувати відповідний сценарій аналізу без ручного втручання користувача.

3.5 Автоматизована попередня обробка даних

Модуль попередньої обробки даних реалізує автоматичне перетворення вхідного набору даних у формат, придатний для навчання моделей машинного навчання. Його основна функція полягає у формуванні узгодженої схеми обробки ознак на основі метаданих, отриманих на етапі профілювання.

Обробка даних виконується з урахуванням типу ознак, їх розподілу та ролі у наборі даних. На першому етапі виконується відбір ознак: колонки, що визначені як ідентифікатори або константи, автоматично виключаються з подальшого аналізу. Це дозволяє зменшити розмірність простору ознак та уникнути включення неінформативних змінних у модель.

Обробка пропущених значень здійснюється диференційовано залежно від типу даних. Для числових ознак використовується імпутація, яка враховує форму розподілу: для змінних із вираженою асиметрією застосовується медіана, тоді як для більш симетричних розподілів

використовується середнє значення. Такий підхід дозволяє зменшити вплив викидів на результати обробки. Для категоріальних ознак застосовується заповнення найбільш частим значенням, що зберігає структуру розподілу категорій.

Категоріальні ознаки перетворюються у числовий формат із використанням різних стратегій кодування залежно від їх кардинальності. Ознаки з невеликою кількістю унікальних значень кодуються за допомогою one-hot представлення, що дозволяє уникнути введення штучного порядку між категоріями. Для ознак із середньою або високою кардинальністю застосовується порядкове кодування, яке забезпечує компактніше представлення та зменшує розмірність простору ознак.

Числові ознаки додатково піддаються масштабуванню у випадках, коли це необхідно для коректної роботи моделей. Масштабування виконується не для всіх моделей, а лише для тих, які чутливі до масштабу ознак, наприклад лінійних моделей або алгоритмів на основі відстаней. Для моделей, що не залежать від масштабу, зокрема дерев рішень та ансамблевих методів, цей етап пропускається. Такий підхід дозволяє уникнути зайвих обчислень і зберегти інтерпретованість ознак.

Реалізація попередньої обробки базується на використанні композиційного підходу, де різні трансформації застосовуються до відповідних груп ознак. Схема попередньої обробки даних наведена на рисунку 3.14. Це дозволяє незалежно обробляти числові та категоріальні змінні в межах єдиного пайплайну. Усі перетворення об’єднуються у єдину структуру, яка інтегрується з моделлю на етапі навчання.

Важливою особливістю реалізації є те, що попередня обробка виконується у складі загального пайплайну навчання моделі. Власне всі трансформації навчаються лише на тренувальних підвибірках у процесі крос-валідації, а потім застосовуються до тестових даних. Такий підхід дозволяє уникнути витоку інформації та забезпечує коректність оцінювання моделей.

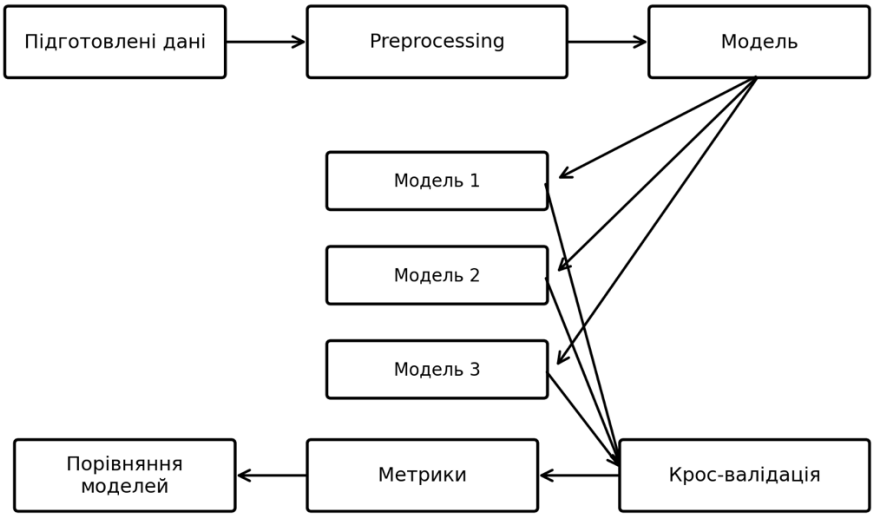


Рисунок 3.14 – Схема попепредньої обробки даних

Таким чином, модуль попередньої обробки забезпечує автоматичну побудову узгодженої та адаптивної схеми трансформації даних, яка враховує їх структуру та властивості і створює основу для ефективного навчання моделей без ручного налаштування.

3.6 Модуль автоматичного навчання моделей

Модуль автоматичного навчання моделей реалізує побудову та оцінювання набору алгоритмів машинного навчання в межах єдиного пайплайну. Його основна функція полягає у запуску декількох моделей на однакових даних з подальшим формуванням узгодженого набору результатів, придатного для порівняння.

Навчання організовано за принципом multi-model підходу, відповідно до якого для кожного типу задачі формується заздалегідь визначений набір кандидатів. Для задач класифікації використовуються як лінійні моделі, так і методи на основі дерев рішень та алгоритми, що базуються на відстанях. Для задач регресії застосовуються лінійні та ансамблеві методи. У випадку

кластеризації використовуються алгоритми, орієнтовані на розбиття даних без використання цільової змінної.

Кожна модель розглядається як незалежна одиниця, яка має уніфікований інтерфейс навчання та прогнозування. Це дозволяє організувати процес обробки у вигляді єдиного циклу, де для кожного алгоритму виконується однакова послідовність дій. Така уніфікація забезпечує простоту розширення системи, оскільки додавання нових моделей не потребує зміни загальної логіки пайплайну.

Перед навчанням кожної моделі формується відповідна схема попередньої обробки даних. Вона будується на основі характеристик набору даних та параметрів конкретного алгоритму. Зокрема, для моделей, чутливих до масштабу ознак, додається етап нормалізації, тоді як для інших моделей використовується спрощена схема обробки. Таким чином забезпечується адаптація preprocessing-пайплайну до вимог конкретної моделі без зміни структури даних.

Навчання моделей здійснюється з використанням крос-валідації, що дозволяє отримати більш стабільну оцінку якості порівняно з одноразовим розбиттям вибірки. Для задач класифікації використовується стратифікований підхід, який зберігає пропорції класів у підвибірках, тоді як для задач регресії застосовується стандартне випадкове розбиття. Такий вибір забезпечує коректність оцінювання для різних типів задач.

У випадку задач кластеризації, де відсутня цільова змінна, навчання виконується на всьому наборі даних без розбиття на підвибірки. Для алгоритмів, що потребують визначення кількості кластерів, використовується перебір можливих значень цього параметра з подальшим вибором найбільш доцільного варіанту за внутрішніми критеріями.

Інтеграція попередньої обробки та моделі виконується у вигляді єдиного пайплайну. Структура пайплайну навчання моделей представлена на рисунку 3.5. Це означає, що всі трансформації даних застосовуються безпосередньо в процесі навчання моделі. Такий підхід забезпечує коректне

застосування обробки лише до тренувальних даних у межах кожної ітерації крос-валідації та виключає витік інформації між підвибірками.

Результатом роботи модуля є структурований набір даних, що містить інформацію про кожну модель, включаючи її назву та обчислені характеристики. Цей набір використовується наступним модулем для ранжування моделей і формування підсумкового представлення результатів.

Таким чином, модуль автоматичного навчання забезпечує систематичну побудову та обробку набору моделей у єдиному форматі, що створює основу для подальшого порівняння та аналізу без необхідності ручного налаштування.

3.7 Оцінка та ранжування моделей

Модуль оцінки та ранжування моделей призначений для формування уніфікованого представлення результатів навчання та їх подальшого впорядкування відповідно до якості. Його функція полягає у приведенні результатів роботи різних алгоритмів до єдиного формату, що дозволяє виконувати їх пряме порівняння незалежно від типу задачі.

Оцінка моделей інтегрована з процесом навчання і виконується автоматично для кожного алгоритму. Для задач із учителем використовується підхід крос-валідації, у межах якого модель послідовно навчається на різних підвибірках даних. У результаті для кожної метрики формується набір значень, що відображає поведінку моделі на різних розбиттях даних. На основі цих значень обчислюються агреговані характеристики, зокрема середнє значення та варіація, які використовуються для подальшого порівняння моделей.

Набір метрик визначається залежно від типу задачі. Для задач класифікації застосовуються метрики, що враховують як загальну точність, так і якість класифікації по окремих класах. Для задач регресії

використовуються метрики, що оцінюють відхилення прогнозованих значень від реальних. У випадку кластеризації застосовуються внутрішні критерії, які дозволяють оцінити структуру сформованих кластерів без використання цільової змінної.

Результати оцінки кожної моделі зберігаються у структурованому вигляді, де для кожного алгоритму формується запис, що містить його ідентифікатор та набір обчислених метрик. Такий підхід дозволяє уніфікувати представлення результатів для різних типів задач і забезпечує можливість їх подальшої обробки без додаткових перетворень.

Ранжування моделей виконується шляхом впорядкування отриманого списку відповідно до обраної основної метрики. Для кожного типу задачі визначається одна ключова метрика, яка використовується як критерій сортування. Моделі, для яких не вдалося обчислити метрики через помилки під час навчання, автоматично відокремлюються та розміщуються у кінці списку. Це дозволяє зберегти інформацію про всі спроби навчання, не впливаючи на коректність ранжування.

Реалізація ранжування передбачає просту та ефективну процедуру сортування, яка застосовується лише до валідних результатів. Такий підхід забезпечує стабільність роботи системи навіть у випадках, коли окремі моделі не можуть бути коректно навчені на конкретному наборі даних.

Важливою особливістю модуля є уніфікація вихідного формату незалежно від типу задачі. Незважаючи на відмінності у метриках та способах їх обчислення, результати подаються у єдиній структурі, що дозволяє використовувати їх без змін у модулях візуалізації та генерації звітів.

Таким чином, модуль оцінки та ранжування забезпечує перетворення результатів навчання моделей у стандартизований вигляд та формує впорядкований список алгоритмів, що є основою для подальшого представлення результатів користувачу.

3.8 Модуль візуалізації результатів

Модуль візуалізації призначений для графічного представлення як вхідних даних, так і результатів роботи моделей у формі, придатній для швидкої інтерпретації. Його основна функція полягає у перетворенні числових та табличних характеристик у наочні графічні об’єкти, що відображають структуру даних і дозволяють порівнювати моделі між собою.

Реалізація модуля побудована як окремий компонент, незалежний від користувацького інтерфейсу. Це дозволяє використовувати одні й ті самі механізми як для інтерактивного відображення у вебзастосунку, так і для генерації графіків у складі звіту. Такий підхід забезпечує повторне використання коду та узгодженість візуального представлення результатів.

Функціонально візуалізація поділяється на два типи: аналіз структури даних та відображення результатів моделювання. Перший тип включає графіки, що характеризують сам набір даних. До них належать розподіли числових ознак, співвідношення класів у цільовій змінній, структура пропущених значень та кореляційні залежності між ознаками. Побудова таких графіків базується на метаданих, отриманих у модулі профілювання, що дозволяє уникнути повторного обчислення характеристик.

Другий тип візуалізацій пов’язаний із порівнянням моделей. Для цього використовується графічне представлення основної метрики для кожного алгоритму у вигляді стовпчикових діаграм. У випадку задач із учителем додатково відображається варіативність метрики, що отримується в процесі крос-валідації. Це дозволяє не лише порівнювати значення метрик, але й оцінювати стабільність моделей.

З технічної точки зору побудова графіків виконується із використанням бібліотек для наукової візуалізації. Графіки формуються як окремі об’єкти, що можуть бути або безпосередньо відображені в інтерфейсі, або збережені у вигляді файлів для подальшого використання.

Такий підхід дозволяє розділити логіку генерації графіків і їх відображення, що спрощує інтеграцію з іншими компонентами системи.

Інтеграція модуля з іншими частинами системи здійснюється через передачу даних у стандартизованому форматі. Для побудови візуалізацій структури даних використовуються вихідні дані та профіль, тоді як для порівняння моделей – результати оцінювання. Це забезпечує чітке розділення відповідальності між модулями та спрощує їх взаємодію.

Таким чином, модуль візуалізації виконує роль проміжного рівня між обчислювальною частиною системи та її представленням користувачу, забезпечуючи наочне відображення як характеристик даних, так і результатів роботи моделей у єдиному узгодженому форматі.

3.9 Генерація аналітичного звіту

Модуль генерації аналітичного звіту призначений для формування структурованого підсумкового представлення результатів роботи системи у вигляді окремого файлу. Його основна функція полягає у збиранні всіх отриманих на попередніх етапах даних та їх організації у зручному для перегляду форматі, що не потребує додаткового середовища виконання.

Реалізація модуля побудована як окремий компонент, який працює незалежно від користувацького інтерфейсу та викликається лише після завершення основного пайплайну. На вхід модуль отримує структуровані дані, зокрема метадані профілювання, результати оцінювання моделей, тип задачі та набір згенерованих графіків. Це дозволяє сформувати повний звіт без необхідності повторного виконання обчислень.

Формування звіту здійснюється у форматі HTML. Такий формат обрано з огляду на його універсальність, можливість поєднання текстових і графічних елементів та відсутність залежності від спеціалізованих інструментів для перегляду. Структура звіту визначається шаблоном, який містить заздалегідь задані секції та правила їх заповнення.

Зміст звіту включає узагальнену інформацію про набір даних, результати профілювання колонок, таблицю моделей із відповідними характеристиками та графічні матеріали. Табличні елементи формуються шляхом ітерації по структурованих даних, тоді як графіки інтегруються у вигляді окремих зображень, що були згенеровані на попередньому етапі. Такий підхід дозволяє чітко розділити процес створення графіків та їх використання у звіті.

Збереження звіту виконується у файловій системі у вигляді HTML-документа разом із пов’язаними графічними файлами. При цьому використовується відносна структура шляхів, що забезпечує коректне відображення зображень при відкритті звіту у браузері. У межах користувацького інтерфейсу передбачена можливість завантаження сформованого звіту, що дозволяє використовувати його поза межами системи.

Важливою особливістю реалізації є відокремлення логіки формування структури звіту від даних, що в ньому відображаються. Це досягається шляхом використання шаблонного підходу, який дозволяє змінювати вигляд звіту без впливу на інші компоненти системи. Така організація підвищує гнучкість та спрощує подальше розширення функціональності.

Разом з тим, реалізація має певні обмеження. Зокрема, зображення підключаються до звіту як окремі файли, що вимагає збереження всієї структури каталогу для коректного відображення. Крім того, звіт є статичним і не передбачає інтерактивної взаємодії з даними. Незважаючи на це, обраний підхід забезпечує простоту реалізації та достатній рівень функціональності для задач автоматизованого аналізу даних.

Таким чином, модуль генерації звіту завершує роботу системи, перетворюючи результати обробки даних у цілісний документ, придатний для подальшого використання та представлення.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

У четвертому розділі розглянуто розробку програмної системи автоматизованого аналізу табличних даних. Реалізація побудована як наскрізний пайплайн, що охоплює завантаження даних, їх профілювання, визначення типу задачі, попередню обробку, навчання моделей, оцінювання та формування звіту. Система має модульну архітектуру, у якій кожен етап реалізований як окремий компонент із чітко визначеною функцією для узгедженності роботи.

4.1 Постановка експериментального дослідження

4.1.1 Мета, принципи та підхід до оцінювання

Метою експериментального дослідження є оцінка ефективності реалізованої системи автоматизованого аналізу даних як цілісного програмного рішення. На відміну від класичних підходів, де аналізується якість окремих алгоритмів машинного навчання, у даній роботі система розглядається як інтегрований конвеєр обробки даних, що об’єднує етапи профілювання, попередньої обробки, автоматичного визначення типу задачі, навчання моделей та їх оцінювання.

Основне завдання полягає у перевірці здатності системи формувати коректні та практично корисні результати аналізу без участі користувача у налаштуванні параметрів. Такий підхід відображає реальний сценарій використання системи, орієнтованої на користувачів без глибоких знань у сфері аналізу даних.

Оцінювання здійснюється аналізом з позиції її зовнішньої поведінки без детального втручання у внутрішні механізми роботи окремих компонентів. Користувач надає вхідний набір даних і отримує результати

аналізу, а якість системи визначається на основі цих результатів та їх узгодженості з очікуваними характеристиками задачі.

У межах дослідження оцінювання проводиться за декількома взаємопов’язаними аспектами. По-перше, перевіряється коректність проходження основних етапів конвеєра та узгодженість між ними. По-друге, аналізується якість моделей, що обираються системою, у порівнянні з базовими підходами. По-третє, досліджується стабільність отриманих результатів за різних умов розбиття даних. Додатково оцінюється здатність системи працювати з різними типами задач, її обчислювальна ефективність, а також зручність використання для неекспертного користувача.

4.1.2 Експериментальне середовище та набори даних

Експериментальне дослідження проводилося на одній обчислювальній платформі з метою забезпечення відтворюваності результатів та виключення впливу апаратних відмінностей на отримані показники. Обчислення виконувалися на робочій станції з процесором Apple M1 Pro та обсягом оперативної пам’яті 16 ГБ у середовищі macOS на архітектурі Apple Silicon.

Програмна реалізація системи базується на мові Python версії 3.8.18. У процесі експериментів використовувалися стандартні бібліотеки для аналізу даних та машинного навчання, зокрема pandas для роботи з табличними даними, NumPy для чисельних обчислень, scikit-learn для реалізації моделей та процедур оцінювання, а також matplotlib і seaborn для побудови візуалізацій. Користувацький інтерфейс системи реалізовано із застосуванням бібліотеки Streamlit.

Для забезпечення стабільності та коректності експериментів застосовано фіксовані параметри виконання. Усі оцінки якості моделей проводилися з використанням п’ятикратної крос-валідації. Для відтворюваності результатів встановлено значення параметра random_state,

рівне 42. Виконання моделей здійснювалося в однопотоковому режимі, що дозволило уникнути нестабільності, пов’язаної з паралельними обчисленнями та конкуренцією потоків.

Для оцінювання роботи системи було використано шість наборів даних, що охоплюють різні типи задач та характеристики даних. Частина з них використовувалася під час розробки системи, інші були обрані як незалежні приклади для перевірки узагальнювальної здатності.

Набір даних Teen Mental Health використано для задачі бінарної класифікації. Він характеризується значним дисбалансом класів, де одна з категорій представлена значно меншою кількістю об’єктів, а також містить як числові, так і категоріальні ознаки. Це дозволяє перевірити стійкість системи до нерівномірного розподілу цільової змінної.

Набір Taxi Trip Pricing застосовано для задачі регресії. Особливістю цього набору є наявність пропущених значень у цільовій змінній, що потребує коректної обробки перед навчанням моделей. Також набір містить змішані типи ознак.

Набір Palmer Penguins використано для задачі кластеризації без явно заданої цільової змінної. Він має невеликий обсяг і включає рядки з повністю відсутніми значеннями, що дозволяє оцінити поведінку системи в умовах неповних даних.

Набір Iris застосовано як класичний приклад багатокласової класифікації. Він характеризується збалансованими класами та відсутністю пропущених значень, що робить його зручним для перевірки базової коректності роботи моделей.

Набір California Housing використано для задачі регресії на даних більшого масштабу. Усі ознаки є числовими, що дозволяє оцінити ефективність системи при роботі з великими наборами даних.

Набір UCI Adult використано для задачі бінарної класифікації з великою кількістю об’єктів і змішаними типами ознак. Цей набір

відображає більш реалістичні умови використання системи, включаючи наявність шуму та складні залежності між ознаками.

Застосування наборів даних різного типу та масштабу дозволяє комплексно оцінити поведінку системи в умовах, наближених до реальних задач аналізу даних.

4.1.3 Метрики оцінювання

Оцінювання результатів роботи системи здійснюється з урахуванням типу аналітичної задачі, що автоматично визначається pipeline. Оскільки система підтримує задачі класифікації, регресії та кластеризації, для кожного типу використовується набір відповідних метрик, що дозволяє коректно інтерпретувати якість отриманих результатів.

Для задач класифікації основною метрикою обрано F1-macro. Це зумовлено тим, що дана метрика однаково враховує якість класифікації для кожного класу незалежно від їх частоти, що є критично важливим у випадках дисбалансу даних. Така ситуація має місце, зокрема, у наборі Teen Mental Health, де одна з категорій представлена значно меншою кількістю спостережень. Додатково аналізуються значення accuracy, F1-weighted та ROC-AUC. Використання кількох метрик дозволяє отримати більш повну картину якості моделей, оскільки кожна з них відображає різні аспекти їхньої поведінки.

Для задач регресії основною метрикою використовується коефіцієнт детермінації R^2, який відображає частку варіації цільової змінної, що пояснюється моделлю. Додатково розглядаються середньоквадратична помилка (RMSE) та середня абсолютна помилка (MAE). Це дозволяє оцінити як відносну якість моделі, так і абсолютний рівень похибки прогнозу.

Для задач кластеризації використовується коефіцієнт силуету (Silhouette score) як основна метрика. Він оцінює ступінь

відокремленості кластерів та щільність об’єктів усередині них. Додатково застосовується індекс Девіса-Болдіна, який характеризує компактність кластерів і відстань між ними. Важливо зазначити, що ці метрики є внутрішніми, тобто не потребують наявності істинних міток класів і оцінюють структуру даних, а не відповідність зовнішнім категоріям.

Усі метрики для задач класифікації та регресії обчислюються з використанням п’ятикратної крос-валідації. Результати подаються у вигляді середнього значення та стандартного відхилення, що дозволяє оцінити не лише якість моделей, але й стабільність їхньої роботи на різних підмножинах даних.

Окремо слід враховувати, що моделі у даній системі використовують стандартні параметри без додаткового налаштування. У зв’язку з цим отримані значення метрик відображають ефективність саме автоматизованого підходу до аналізу даних, а не максимально можливу якість, яку можна досягти при ручному підборі гіперпараметрів. Це відповідає основній меті дослідження, яка полягає у перевірці практичної придатності системи в умовах мінімального втручання користувача.

Таким чином, обраний набір метрик забезпечує комплексне оцінювання результатів роботи системи з урахуванням специфіки кожного типу задач та дозволяє обґрунтовано порівнювати отримані результати між різними експериментами.

4.2 Коректність роботи pipeline

4.2.1 Сценарій та методика перевірки

Перевірка коректності роботи системи здійснюється в умовах повністю автоматизованого сценарію, що відповідає типовому способу її використання. Користувач завантажує табличний набір даних та запускає процес аналізу без додаткового налаштування параметрів або втручання у

внутрішню логіку системи. У такому режимі всі етапи обробки виконуються послідовно, починаючи з профілювання даних і завершуючи формуванням підсумкових результатів.

Методика перевірки базується на поетапному контролі вихідних даних кожного компонента pipeline. Для цього визначено набір критеріїв, відповідно до яких оцінюється коректність роботи кожного етапу. Особливу увагу приділено узгодженості між етапами, оскільки помилки на ранніх стадіях можуть призводити до некоректних результатів на подальших кроках.

Оцінювання проводиться для трьох репрезентативних наборів даних, що відповідають різним типам задач: класифікації, регресії та кластеризації. Такий підхід дозволяє перевірити універсальність реалізованого рішення.

Критерії перевірки сформульовано для кожного етапу конвеєра:

– на етапі завантаження перевіряється коректність зчитування даних та відповідність їх розмірності очікуваним характеристикам;

– на етапі профілювання оцінюється правильність визначення типів ознак, наявність статистичних характеристик та виявлення особливостей даних;

– на етапі визначення задачі перевіряється відповідність автоматично обраного типу задачі фактичній структурі даних;

– на етапі попередньої обробки контролюється відсутність пропущених значень після трансформації та коректність формування числового представлення ознак;

– на етапі навчання моделей перевіряється виконання всіх моделей, передбачених для відповідного типу задачі;

– на етапі оцінювання аналізується формування таблиці метрик та коректність ранжування моделей;

– на етапі формування звіту перевіряється генерація візуалізацій і підсумкового HTML-документа.

Окрім формальних критеріїв, додатково аналізуються нетривіальні випадки, зокрема наявність пропущених значень у цільовій змінній, сильний дисбаланс класів або відсутність цільової змінної для задач кластеризації. Це дозволяє оцінити поведінку системи в умовах, наближених до реальних задач аналізу даних.

Таким чином, запропонована методика забезпечує комплексну перевірку коректності функціонування pipeline як єдиної системи, що є необхідною умовою для подальшого аналізу якості моделей та ефективності автоматизації.

4.2.2 Результати та їх аналіз

Результати перевірки коректності роботи pipeline узагальнено для трьох репрезентативних наборів даних, що відповідають задачам класифікації, регресії та кластеризації. Оцінювання здійснювалося відповідно до визначених на попередньому етапі критеріїв. Коректність роботи етапів pipeline приведено в таблиці 4.1.

Таблиця 4.1 – Коректність роботи етапів pipeline

Етап	Teen Mental Health	Taxi Trip Pricing	Palmer Penguins
Завантаження даних	виконано (1200 × 13)	виконано (1000 × 11)	виконано (344 × 5)
Профілювання	коректно	коректно	коректно
Визначення задачі	класифікація (коректно)	регресія (коректно)	кластеризація (коректно)
Попередня обробка	17 ознак, без пропусків	18 ознак, без пропусків	7 ознак, без пропусків
Навчання моделей	4/4 моделей виконано	4/4 моделей виконано	3/3 моделей виконано
Оцінювання	сформовано таблицю метрик	сформовано таблицю метрик	сформовано таблицю метрик
Формування звіту	згенеровано	згенеровано	згенеровано

Отримані результати свідчать про коректну роботу всіх етапів pipeline незалежно від типу задачі та структури даних. Жодного випадку аварійного завершення або пропуску етапів не зафіксовано.

Окрему увагу було приділено аналізу специфічних ситуацій, що можуть впливати на коректність обробки даних.

Для набору Teen Mental Health характерним є значний дисбаланс класів. Система коректно виявила цю особливість на етапі профілювання, що відобразилося у подальшому використанні метрики F1-macro як основної для оцінювання моделей. Це свідчить про узгодженість між етапами pipeline.

У наборі Taxi Trip Pricing було виявлено пропущені значення у цільовій змінній. Система автоматично виключила відповідні рядки перед навчанням моделей, що є коректною поведінкою для задач регресії. Дана операція не призвела до порушення структури даних і була виконана до початку крос-валідації.

У наборі Palmer Penguins, який використовувався для задачі кластеризації, була відсутня явна цільова змінна. Система правильно визначила тип задачі як кластеризацію. Крім того, було виявлено рядок з повністю відсутніми значеннями, який був оброблений на етапі попередньої обробки шляхом імпутації. Після цього сформовано коректне числове представлення даних.

Загалом результати експерименту підтверджують, що pipeline працює як узгоджена система, де кожен етап передає коректні дані наступному. Всі перевірені сценарії виконуються без втручання користувача, що відповідає поставленій меті автоматизації аналізу даних.

Отримані результати створюють основу для подальшого аналізу якості моделей та ефективності автоматизованого підходу, що розглядається у наступних підрозділах.

4.3 Ефективність автоматизації та порівняння з базовими моделями

4.3.1 Методика порівняння

Для оцінки ефективності автоматизованого підходу проведено порівняння результатів роботи реалізованої системи з базовим підходом, який імітує дії користувача з початковим рівнем підготовки у сфері аналізу даних. Такий підхід дозволяє визначити, чи забезпечує автоматизація реальний виграш у якості результатів без додаткових зусиль з боку користувача.

Базовий підхід реалізовано у вигляді стандартного конвеєра на основі бібліотеки scikit-learn. Для попередньої обробки використовуються прості стратегії: числові ознаки заповнюються медіанними значеннями, категоріальні – найчастішими значеннями з подальшим кодуванням методом one-hot.

Для числових ознак також застосовується масштабування. На етапі навчання використовується одна модель зі стандартними параметрами: логістична регресія для задач класифікації та лінійна регресія для задач регресії.

Автоматизований підхід передбачає використання реалізованого pipeline, який самостійно визначає тип задачі, виконує попередню обробку та запускає декілька моделей відповідного типу з подальшим вибором найкращої за основною метрикою.

Порівняння здійснюється на однакових розбиттях даних із використанням п’ятикратної крос-валідації. Це забезпечує коректність порівняння та виключає вплив випадкових факторів.

Для задач класифікації основною метрикою є F1-macro, для задач регресії – коефіцієнт детермінації R^2.

4.3.2 Результати та їх аналіз

Результати порівняння автоматизованого підходу з базовими моделями наведено у таблиці 4.2.

Таблиця 4.2 – Порівняння автоматизованого pipeline з базовими моделями

Набір даних	Тип задачі	Базова модель	Базова метрика	Найкраща модель (pipeline)	Метрика pipeline	Приріст
Teen Mental Health	Класи-фікація	LogisticRegression	0.786 ± 0.020	GradientBoosting Classifier	0.983 ± 0.021	+0.197
Iris	Класи-фікація	LogisticRegression	0.953 ± 0.045	KNeighbors Classifier	0.973 ± 0.025	+0.020
Taxi Trip Pricing	регресія	LinearRegression	0.854 ± 0.036	RandomForest Regressor	0.932 ± 0.021	+0.078

Отримані результати демонструють, що автоматизований pipeline у всіх розглянутих випадках забезпечує вищу якість моделей порівняно з базовим підходом.

Найбільш суттєвий приріст спостерігається для набору Teen Mental Health. Це пояснюється тим, що базова модель логістичної регресії є лінійною і має обмежену здатність відображати складні залежності між ознаками. Крім того, вона є чутливою до дисбалансу класів. У той же час модель градієнтного бустингу, обрана автоматизованою системою, дозволяє враховувати нелінійні взаємозв’язки та демонструє значно кращі результати.

Разом з тим, настільки високе значення F1-macro для цього набору даних потребує обережної інтерпретації. Додатково було виконано перевірку на можливу наявність витоку цільової змінної, зокрема шляхом контролю включення цільової змінної до ознак та аналізу корельованих

атрибутів. Ознак витоку виявлено не було, що дозволяє вважати отриманий результат коректним у межах проведеного експерименту.

Для набору Iris різниця між підходами є незначною, через те що задача є відносно простою, дані збалансовані, а залежності між ознаками є близькими до лінійних. У таких умовах навіть базова модель демонструє високі результати, тому автоматизація не дає суттєвого виграшу, а лише незначно покращує якість за рахунок вибору альтернативної моделі.

У випадку регресії на наборі Taxi Trip Pricing спостерігається помірний приріст якості. Лінійна регресія не здатна повністю відобразити нелінійні залежності між ознаками, такими як відстань поїздки, час та додаткові фактори. Натомість модель Random Forest, обрана системою, краще адаптується до таких залежностей, що й призводить до покращення значення R^2.

Загалом результати експерименту підтверджують, що автоматизований підхід дозволяє отримувати моделі не гіршої, а в більшості випадків кращої якості порівняно з простими ручними рішеннями. Це свідчить про ефективність застосування автоматизації у задачах аналізу табличних даних, особливо у випадках складних або неоднорідних наборів даних.

4.4 Стабільність результатів

4.4.1 Методика оцінювання стабільності

Стабільність результатів є критичною характеристикою системи автоматизованого аналізу даних, оскільки кінцевий користувач очікує отримання узгоджених рекомендацій незалежно від випадкових факторів, пов’язаних із розбиттям вибірки. Нестабільність може проявлятися у значних коливаннях значень метрик або зміні порядку моделей у підсумковому ранжуванні при повторному запуску аналізу.

Для оцінювання стабільності роботи реалізованого pipeline застосовано два взаємодоповнюючих підходи.

Перший підхід базується на аналізі варіативності метрик якості в межах крос-валідації. Для кожної моделі обчислюється середнє значення основної метрики та стандартне відхилення за п’ятьма фолдами. Додатково розраховується коефіцієнт варіації, який визначається як відношення стандартного відхилення до середнього значення. Невелике значення цього показника свідчить про стабільність оцінки якості моделі на різних підмножинах даних.

Другий підхід передбачає аналіз стабільності ранжування моделей. Для цього pipeline запускається декілька разів із різними значеннями параметра random_state, що впливає на формування навчальних і тестових підвибірок. У даному дослідженні використано п’ять запусків із різними значеннями random_state. Для кожного запуску формується впорядкований список моделей за значенням основної метрики.

Отримані ранжування порівнюються між собою за допомогою коефіцієнта кореляції рангів Кендалла (Kendall tau), який дозволяє оцінити ступінь узгодженості порядку моделей між різними запусками. Значення коефіцієнта, близьке до одиниці, свідчить про високу стабільність ранжування, тоді як менші значення вказують на чутливість системи до випадкових змін у розбитті даних.

Експеримент проводиться для двох наборів даних: Teen Mental Health та Taxi Trip Pricing, які характеризуються різними властивостями. Перший набір має значний дисбаланс класів, що може впливати на варіативність метрик, тоді як другий є більш збалансованим, але містить складні залежності між ознаками.

Таким чином, запропонована методика дозволяє оцінити як стабільність абсолютних значень метрик, так і надійність вибору найкращої моделі, що є ключовим для практичного використання системи автоматизованого аналізу даних.

4.4.2 Результати та їх аналіз

Результати оцінювання стабільності наведено у таблицях 4.3 та 4.4. Перша таблиця відображає варіативність основної метрики для кожної моделі в межах крос-валідації, друга – узгодженість ранжування моделей при різних значеннях random_state.

Таблиця 4.3 – Варіативність метрик у крос-валідації

Набір даних	Модель	Середнє значення	Стандартне відхилення	Коефіцієнт варіації
Teen Mental Health	GradientBoostingClassifier	0.983	0.021	0.022
Teen Mental Health	LogisticRegression	0.714	0.057	0.080
Teen Mental Health	RandomForestClassifier	0.706	0.113	0.160
Teen Mental Health	KNeighborsClassifier	0.518	0.050	0.097
Taxi Trip Pricing	RandomForestRegressor	0.932	0.021	0.022
Taxi Trip Pricing	GradientBoostingRegressor	0.929	0.031	0.033
Taxi Trip Pricing	Ridge	0.854	0.036	0.043
Taxi Trip Pricing	LinearRegression	0.854	0.036	0.042

Отримані значення свідчать про різний рівень стабільності моделей. Для задачі регресії всі моделі демонструють низький коефіцієнт варіації, що вказує на стабільні оцінки якості. Для задачі класифікації спостерігається більша варіативність, особливо для моделі RandomForestClassifier, що пояснюється впливом дисбалансу класів на розбиття даних у фолдах.

Результати показують, що для набору Teen Mental Health ранжування моделей є повністю стабільним при зміні random_state. Усі запуски дають

однаковий порядок моделей, що свідчить про значну різницю у якості між ними.

Таблиця 4.4 – Узгодженість ранжування моделей

Набір даних	Середнє значення Kendall tau	Ідентичні ранжування
Teen Mental Health	1.00	10 / 10
Taxi Trip Pricing	0.67	2 / 10

Для набору Taxi Trip Pricing спостерігається менша стабільність ранжування. Це пояснюється тим, що значення метрик для найкращих моделей є дуже близькими, тому навіть незначні зміни у розбитті даних можуть призводити до зміни їхнього порядку.

Важливо зазначити, що в обох випадках найкраща модель залишається незмінною при всіх запусках. Таким чином, нестабільність спостерігається лише у середній частині ранжування і не впливає на основну рекомендацію системи.

Загалом результати експерименту підтверджують, що система забезпечує достатній рівень стабільності для практичного використання. Вибір найкращої моделі є надійним, навіть за наявності варіативності у другорядних позиціях.

4.5 Оцінка роботи системи на різних типах задач

4.5.1 Результати для різних типів задач

Для оцінювання універсальності розробленої системи було проведено аналіз її роботи на задачах трьох основних типів: класифікації, регресії та кластеризації. Основна увага приділялася перевірці здатності pipeline автоматично визначати тип задачі, обирати відповідний набір моделей,

застосовувати релевантні метрики оцінювання та формувати коректне ранжування моделей.

Результати експерименту наведено у таблиці 4.5.

Таблиця 4.5 – Результати роботи системи для різних типів задач

Набір даних	Визначений тип задачі	Кількість моделей	Основна метрика	Найкраща модель	Значення метрики
Teen Mental Health	класифікація	4	F1-macro	GradientBoosting Classifier	0.983 ± 0.021
Iris	класифікація	4	F1-macro	KNeighbors Classifier	0.973 ± 0.025
Taxi Trip Pricing	регресія	4	R^2	RandomForest Regressor	0.932 ± 0.021
California Housing	регресія	4	R^2	RandomForest Regressor	0.811 ± 0.008
Palmer Penguins	кластеризація	3	Silhouette	DBSCAN	0.536

Отримані результати свідчать про коректну роботу системи для всіх розглянутих типів задач. У кожному випадку pipeline правильно визначає тип задачі та використовує відповідний набір моделей і метрик.

Для задач класифікації система застосовує метрику F1-macro, що є обґрунтованим вибором у випадках дисбалансу класів. Для задач регресії використовується коефіцієнт детермінації R^2, який дозволяє оцінити якість прогнозування. Для задач кластеризації застосовується Silhouette score, що характеризує якість розбиття даних на кластери без використання еталонних міток.

Також підтверджено, що для кожного типу задачі система використовує відповідний набір моделей. Для задач класифікації та регресії це набір моделей із бібліотеки scikit-learn, тоді як для кластеризації застосовуються алгоритми, що не потребують цільової змінної.

Таким чином, результати демонструють здатність системи коректно адаптуватися до різних типів задач без необхідності ручного налаштування

з боку користувача, що є ключовою вимогою до автоматизованих систем аналізу даних.

4.5.2 Узагальнений аналіз

Отримані результати дозволяють сформулювати узагальнену оцінку роботи системи на різних типах задач.

По-перше, система демонструє коректну поведінку на рівні визначення типу задачі. У всіх розглянутих випадках тип задачі був визначений правильно, що забезпечило використання відповідного набору моделей та метрик. Це є критично важливим, оскільки помилка на цьому етапі призвела б до некоректного подальшого аналізу.

По-друге, спостерігається узгодженість між вибором моделей та характеристиками даних. Для більшості задач класифікації та регресії найкращі результати показують ансамблеві моделі, зокрема Random Forest та Gradient Boosting. Це пояснюється їх здатністю враховувати нелінійні залежності між ознаками. Водночас у простіших задачах, таких як Iris, система може обирати менш складні моделі, що демонструє відсутність жорсткої прив’язки до одного класу алгоритмів.

По-третє, для задач кластеризації система забезпечує коректне функціонування навіть за відсутності цільової змінної. Використання внутрішніх метрик, зокрема Silhouette score, дозволяє оцінити якість кластеризації без необхідності знання еталонних міток. Водночас слід враховувати, що такі метрики оцінюють лише геометричну структуру даних і не гарантують відповідності кластерів реальним категоріям.

Окремо варто відзначити, що результати для різних задач мають різний рівень чутливості до структури даних. Для задач регресії значення метрик змінюються більш плавно, тоді як для задач класифікації, особливо за наявності дисбалансу, можливі суттєві коливання якості моделей.

Таким чином, система демонструє універсальність та здатність адаптуватися до різних типів задач аналізу даних. Водночас результати підкреслюють важливість урахування специфіки даних при інтерпретації отриманих показників, особливо для задач кластеризації та класифікації з дисбалансом класів.

4.6 Обчислювальна ефективність системи

4.6.1 Методика вимірювання

Оцінка обчислювальної ефективності системи спрямована на визначення часу виконання pipeline та виявлення етапів, які найбільше впливають на загальну тривалість аналізу. Це є важливим з точки зору практичного використання системи, оскільки надмірний час обробки може обмежувати її застосування у реальних умовах.

Вимірювання часу виконання здійснювалося для всіх основних етапів pipeline: завантаження даних, профілювання, визначення типу задачі, попередньої обробки, навчання моделей та формування звіту. Для цього в реалізації було використано функцію time.perf_counter, яка дозволяє отримувати точні значення часу виконання окремих операцій.

Експеримент проводився для кількох наборів даних різного розміру та типу. Для малих та середніх наборів (Teen Mental Health, Taxi Trip Pricing, Palmer Penguins) вимірювання здійснювалося для повного циклу обробки, включаючи генерацію звіту. Для аналізу масштабованості використовувалися підмножини набору UCI Adult розміром 100, 1000, 10000 та 30000 рядків. У цьому випадку етап формування звіту було виключено, щоб оцінити саме обчислювальне навантаження.

Кожен експеримент повторювався декілька разів, після чого обчислювалося середнє значення часу виконання. Це дозволило зменшити

вплив випадкових коливань, пов’язаних із роботою операційної системи та фоновими процесами.

Таким чином, запропонована методика дозволяє оцінити як абсолютний час виконання системи, так і її поведінку при збільшенні обсягу даних, що є важливим для визначення меж практичного застосування розробленого рішення.

4.6.2 Результати та аналіз

Результати вимірювання обчислювальної ефективності наведено у таблицях 4.6 та 4.7. Перша таблиця відображає розподіл часу виконання між основними етапами pipeline для наборів даних малого та середнього розміру, друга – залежність загального часу виконання від кількості об’єктів.

Таблиця 4.6 – Розподіл часу виконання за етапами pipeline

Етап	Teen Mental Health (1200)	Taxi Trip Pricing (1000)	Palmer Penguins (344)
Завантаження	< 0.01 с	< 0.01 с	< 0.01 с
Профілювання	0.01 с	0.01 с	< 0.01 с
Визначення задачі	< 0.01 с	< 0.01 с	< 0.01 с
Навчання та оцінювання	1.50 с	3.03 с	0.10 с
Формування звіту	0.70 с	0.53 с	0.39 с
Загальний час	2.21 с	3.57 с	0.49 с

Отримані результати свідчать про те, що основна частина часу виконання припадає на етап навчання та оцінювання моделей. Для всіх наборів даних цей етап займає від 60% до 85% загального часу виконання. Інші етапи, зокрема профілювання та визначення задачі, виконуються

практично миттєво і не впливають суттєво на загальну продуктивність системи.

Цікавим є те, що для набору Taxi Trip Pricing час навчання є більшим, ніж для набору Teen Mental Health, незважаючи на схожий розмір даних. Це пояснюється використанням моделей регресії, зокрема Random Forest та Gradient Boosting, які є обчислювально більш затратними порівняно з відповідними моделями для класифікації.

Таблиця 4.7 – Масштабованість системи залежно від розміру даних

Кількість рядків	Загальний час	Час на один рядок
100	0.90 с	9.05 мс
1000	1.70 с	1.70 мс
10000	10.87 с	1.09 мс
30000	36.59 с	1.22 мс

Аналіз масштабованості показує, що система демонструє близьку до лінійної залежність часу виконання від кількості об’єктів. При збільшенні обсягу даних час на обробку одного рядка зменшується на початкових етапах, що пояснюється впливом постійних накладних витрат, але надалі стабілізується.

Таким чином, результати експерименту підтверджують, що система є достатньо ефективною для обробки наборів даних малого та середнього розміру в інтерактивному режимі. Основним обмеженням є етап навчання моделей, який визначає загальну тривалість аналізу.

Отримані результати дозволяють зробити висновок про придатність системи до практичного використання, а також визначити напрями для подальшої оптимізації, зокрема за рахунок скорочення кількості моделей або оптимізації їх параметрів.

4.7 Аналіз використання системи неекспертним користувачем

Для оцінки зручності використання системи було змодельовано сценарій взаємодії неекспертного користувача з інтерфейсом. Це такий користувач, який має досвід роботи з методами машинного навчання, але має базові навички роботи з табличними даними.

Сценарій передбачає виконання мінімальної кількості дій: відкриття інтерфейсу, завантаження набору даних та запуск аналізу. Користувач не змінює налаштування, не обирає моделі вручну та покладається на автоматичні рекомендації системи щодо вибору цільової змінної та типу задачі.

Для порівняння розглядається також сценарій використання системи експертом, з котнролем параметрів або безпосередня програмна робота.

Оцінювання проводиться за такими показниками як кількість дій, необхідних для отримання результату, час від початку взаємодії до отримання підсумкових результатів, необхідність втручання користувача у процес (override), узгодженість результатів із результатами, отриманими експертом.

Для проведення експерименту використано два набори даних: Teen Mental Health як приклад невеликого набору даних та UCI Adult як приклад набору більшого масштабу. Результати експерименту наведено у таблицях 4.8 та 4.9.

Таблиця 4.8 – Результати для набору Teen Mental Health

Користувач	Кількість дій	Час до результату	Використано рекомендації	Найкраща модель	Метрика
Неексперт	3	1.55 с	так	GradientBoosting Classifier	0.983 ± 0.021
Експерт	3	1.68 с	так	GradientBoosting Classifier	0.983 ± 0.021
Експерт (код)	–	1.50 с	–	GradientBoosting Classifier	0.983 ± 0.021

Таблиця 4.9 – Результати для набору UCI Adult

Користувач	Кількість дій	Час до результату	Використано рекомендації	Найкраща модель	Метрика
Неексперт	3	41.20 с	так	GradientBoosting Classifier	0.801 ± 0.004
Експерт	3	40.82 с	так	GradientBoosting Classifier	0.801 ± 0.004
Експерт (код)	–	41.19 с	–	GradientBoosting Classifier	0.801 ± 0.004

Отримані результати свідчать про те, що неекспертний користувач може отримати результати, ідентичні результатам експерта, виконуючи мінімальну кількість дій. Це підтверджує ефективність автоматизації та коректність роботи системи в умовах відсутності ручного налаштування.

У процесі проведення експерименту було виявлено проблему у механізмі автоматичного визначення цільової змінної. Початкова реалізація використовувала пошук підрядків, що призвело до помилкового вибору ознаки workclass як цільової через наявність підрядка «class». Це є прикладом потенційно небезпечної ситуації, коли система може сформувати некоректний результат без явного повідомлення про помилку.

Після виявлення дефекту алгоритм було модифіковано шляхом переходу до порівняння окремих токенів назв ознак, що дозволило уникнути помилкових збігів. Після внесення змін система коректно визначає цільову змінну для набору UCI Adult.

Таким чином, експеримент дозволив не лише оцінити зручність використання системи, але й виявити та усунути критичний дефект, що підвищило надійність її роботи.

Загалом результати підтверджують, що система є доступною для користувачів без спеціалізованих знань та забезпечує отримання коректних результатів за мінімальної взаємодії.

У межах даного розділу було проведено комплексне експериментальне дослідження реалізованої системи автоматизованого аналізу даних, що розглядається як цілісний pipeline обробки табличних

даних. Оцінювання здійснювалося за кількома взаємопов’язаними напрямами, що дозволило сформувати узагальнене уявлення про ефективність, надійність та практичну придатність системи.

Перш за все, було підтверджено коректність роботи всіх етапів pipeline. Система стабільно виконує завантаження даних, профілювання, автоматичне визначення типу задачі, попередню обробку, навчання моделей та формування результатів. Важливим є те, що коректна робота забезпечується навіть у складних умовах, таких як наявність пропущених значень, дисбаланс класів або відсутність цільової змінної.

Порівняння з базовими моделями показало, що автоматизований підхід у більшості випадків забезпечує покращення якості результатів. Це досягається за рахунок використання більш гнучких моделей та автоматичного вибору оптимального алгоритму. При цьому у простих задачах різниця між підходами є незначною, що підтверджує адекватність базових моделей у таких умовах.

Аналіз стабільності результатів показав, що система демонструє високий рівень надійності. Варіативність метрик у межах крос-валідації є помірною, а вибір найкращої моделі залишається стабільним навіть при зміні параметрів розбиття даних. Нестабільність спостерігається лише для моделей із близькими значеннями метрик і не впливає на кінцеву рекомендацію.

Дослідження роботи системи на різних типах задач підтвердило її універсальність. Pipeline коректно адаптується до задач класифікації, регресії та кластеризації, автоматично обираючи відповідні моделі та метрики. Це дозволяє використовувати систему для широкого спектра задач аналізу даних без додаткового налаштування.

Оцінка обчислювальної ефективності показала, що система є придатною для інтерактивного використання на наборах даних малого та середнього розміру. Основним обмежуючим фактором є етап навчання моделей, який визначає більшу частину часу виконання. При збільшенні

обсягу даних система демонструє близьку до лінійної залежність часу виконання.

Аналіз сценарію використання неекспертним користувачем підтвердив, що система є доступною та зрозумілою у використанні. Користувач може отримати коректні результати, виконуючи мінімальну кількість дій. Водночас експеримент дозволив виявити критичний дефект у механізмі визначення цільової змінної, що був усунений у процесі дослідження. Це підкреслює важливість експериментальної перевірки системи як цілісного продукту.

Разом з тим, у ході дослідження було визначено ряд обмежень. Зокрема, система не виконує підбір гіперпараметрів моделей, що може обмежувати максимальну якість результатів. Для задач кластеризації використовуються внутрішні метрики, які не завжди відображають реальну структуру даних. Крім того, механізм автоматичного визначення цільової змінної залежить від назв ознак і може потребувати подальшого вдосконалення.

Таким чином, результати проведеного дослідження свідчать про те, що розроблена система є ефективним інструментом автоматизації аналізу табличних даних. Вона забезпечує коректність роботи, достатній рівень якості моделей, стабільність результатів та зручність використання, що дозволяє розглядати її як основу для подальшого розвитку та вдосконалення.

ВИСНОВКИ

У межах виконаної кваліфікаційної роботи було вирішено науково-практичну задачу автоматизації аналізу табличних даних із використанням моделей штучного інтелекту у вигляді цілісної програмної системи. Отримані результати безпосередньо відповідають задачам, сформульованим у постановці задачі, та підтверджують досягнення поставленої мети.

У процесі дослідження було проаналізовано існуючі підходи до автоматизації аналізу даних, включаючи класичні методи машинного навчання, AutoML-платформи та сучасні інтелектуальні системи підтримки прийняття рішень. Встановлено, що наявні рішення або вимагають високого рівня експертності користувача, або мають обмежену гнучкість у роботі з різнорідними даними, що обґрунтовує доцільність розробки власного підходу.

Було розроблено метод автоматизованого аналізу даних, який реалізує повний аналітичний конвеєр: від завантаження та попередньої обробки даних до автоматичного вибору моделей, налаштування гіперпараметрів та формування аналітичних висновків. На відміну від класичних підходів, запропоноване рішення орієнтоване на мінімізацію участі користувача та забезпечує інтерпретованість отриманих результатів на рівні системних звітів.

У рамках практичної реалізації було спроєктовано та реалізовано програмну систему з модульною архітектурою, що включає підсистеми обробки даних, аналітичного ядра, автоматичного вибору моделей та генерації звітів. Архітектурні рішення забезпечують масштабованість, розширюваність та можливість інтеграції з іншими інформаційними системами.

Експериментальна перевірка системи проводилась на наборі різнорідних табличних датасетів, що дозволило оцінити її ефективність як

єдиного інтегрованого рішення. За результатами експериментів встановлено, що система забезпечує стабільну якість моделей, коректно адаптується до різних типів задач (класифікація, регресія) та демонструє узгодженість результатів при повторних запусках. Отримані метрики якості підтверджують доцільність використання автоматизованого підходу для задач аналізу даних без глибокого втручання експерта.

Таким чином, усі поставлені задачі були виконані: проведено аналіз предметної галузі, розроблено метод автоматизації аналізу даних, реалізовано програмну систему та проведено її експериментальну оцінку.

Практичне значення роботи полягає у можливості використання розробленої системи як інструменту для швидкого отримання аналітичних інсайтів у задачах бізнес-аналітики, дослідження даних та підтримки прийняття рішень. Запропонований підхід може бути використаний як основа для подальшого розвитку інтелектуальних систем автоматизованого аналізу даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. He X., Zhao K., Chu X. AutoML: a survey of the state-of-the-art. *Knowledge-Based systems*. 2021. Vol. 212. P. 106622. URL: https://doi.org/10.1016/j.knosys.2020.106622 (date of access: 08.07.2026).

2. Automated machine learning: past, present and future / M. Baratchi et al. *Artificial Intelligence Review*. 2024. Vol. 57, no. 5. URL: https://doi.org/10.1007/s10462-024-10726-1 (date of access: 08.07.2026).

3. A comprehensive survey on automated machine learning for recommendations / B. Chen et al. *ACM Transactions on Recommender Systems*. 2023. URL: https://doi.org/10.1145/3630104 (date of access: 08.07.2026).

4. AutoML: a systematic review on automated machine learning with neural architecture search / I. Salehin et al. *Journal of Information and Intelligence*. 2023. URL: https://doi.org/10.1016/j.jiixd.2023.10.002 (date of access: 08.07.2026).

5. Alcobaça E., de Carvalho A. C. P. L. F. A literature review on automated machine learning. *Artificial Intelligence Review*. 2025. Vol. 59, no. 1. URL: https://doi.org/10.1007/s10462-025-11397-2 (date of access: 08.07.2026).

6. Feurer M., Hutter F. Hyperparameter optimization. *Automated Machine Learning*. Cham, 2019. P. 3–33. URL: https://doi.org/10.1007/978-3-030-05318-5_1 (date of access: 08.07.2026).

7. Automated machine learning / ed. by F. Hutter, L. Kotthoff, J. Vanschoren. *Springer International Publishing*, 2019. URL: https://doi.org/10.1007/978-3-030-05318-5 (date of access: 08.07.2026).

8. Olson R. S., Moore J. H. TPOT: A tree-based pipeline optimization tool for automating machine learning. *Automated Machine Learning*. Cham, 2019. P. 151–160. URL: https://doi.org/10.1007/978-3-030-05318-5_8 (date of access: 08.07.2026).

9. Auto-sklearn: efficient and robust automated machine learning / M. Feurer et al. *Automated Machine Learning*. Cham, 2019. P. 113–134. URL: https://doi.org/10.1007/978-3-030-05318-5_6 (date of access: 08.07.2026).

10. Ramasubramanian K., Singh A. Scalable machine learning and related technologies. *Machine Learning Using R*. Berkeley, CA, 2016. P. 519–554. URL: https://doi.org/10.1007/978-1-4842-2334-5_9 (date of access: 08.07.2026).

11. Zhuhadar L. P., Lytras M. D. The application of AutoML techniques in diabetes diagnosis: current approaches, performance, and future directions. *Sustainability*. 2023. Vol. 15, no. 18. P. 13484. URL: https://doi.org/10.3390/su151813484 (date of access: 08.07.2026).

12. A survey of evaluating AutoML and automated feature engineering tools in modern data science / D. Dissanayake et al. *27th International Conference on Enterprise Information Systems*, Porto, Portugal, 4–6 April 2025. 2025. P. 218–225. URL: https://doi.org/10.5220/0013266700003929 (date of access: 08.07.2026).

13. Automatic design of machine learning via evolutionary computation: A survey / N. Li et al. *Applied Soft Computing*. 2023. P. 110412. URL: https://doi.org/10.1016/j.asoc.2023.110412 (date of access: 08.07.2026).

14. Assessing the use of AutoML for data-driven software engineering / F. Calefato et al. *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, New Orleans, LA, USA, 26–27 October 2023. 2023. URL: https://doi.org/10.1109/esem56168.2023.10304796 (date of access: 08.07.2026).

15. Data preprocessing using AutoML: a survey / A. Salhi et al. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, Las Vegas, NV, USA, 24–27 July 2023. 2023. URL: https://doi.org/10.1109/csce60160.2023.00265 (date of access: 08.07.2026).