

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 стабільності та чутливості алгоритмів машинного навчання до
_____ змін у навчальних даних
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-22-3

_____ Володимир КУСАКІН
(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 _____ доц. Олег ЗОЛОТУХІН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ

(підпис)

_____ Лариса ЧАЛА
(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 _____ Кусакіну Володимир Михайловичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження стабільності та чутливості алгоритмів машинного навчання до змін у навчальних даних

затверджена наказом університету від 5 червня 2026 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 липня 2026 р.

3. Вихідні дані до роботи Наукові публікації та навчально-методичні матеріали з машинного навчання, бібліотеки Python (Scikit-learn, NumPy, Pandas, Matplotlib), відкриті набори даних для задач класифікації, програмні засоби для проведення експериментів та аналізу результатів.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Вибір алгоритмів, метрик та проєктування експерименту _____

3) Формування навчальних даних та проведення експериментів _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі та джерел	10.06.2026	виконано
3	Формування теоретичної частини роботи	13.06.2026	виконано
4	Вибір алгоритмів та метрик оцінювання	16.06.2026	виконано
5	Проектування експерименту	18.06.2026	виконано
6	Розробка програмної реалізації	21.06.2026	виконано
7	Проведення обчислювальних експериментів	24.06.2026	виконано
8	Аналіз результатів дослідження	27.06.2026	виконано
9	Оформлення пояснювальної записки	29.06.2026	виконано
10	Підготовка презентації та доповіді	30.06.2026	виконано
11	Подання пояснювальної записки на перевірку	02.07.2026	виконано
12	Нормоконтроль	02.07.2026	виконано
13	Перевірка на академічний плагіат	04.07.2026	виконано
14	Рецензування	08.07.2026	виконано
15	Захист перед ЕК	10.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач _____

 (підпис)

Керівник роботи _____
 (підпис)

_____ доц. Олег ЗОЛОТУХІН
 (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 107 с., 13 рис., 10 табл., 1 дод., 14 джерел.

АЛГОРИТМИ МАШИННОГО НАВЧАННЯ, НАВЧАЛЬНІ ДАНІ, ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ, РОБАСТНІСТЬ, СТАБІЛЬНІСТЬ, ЧУТЛИВІСТЬ, DATA SHIFT, MACHINE LEARNING, ROBUSTNESS, SENSITIVITY, STABILITY.

Предметом дослідження є властивості стабільності та чутливості алгоритмів машинного навчання, що проявляються за умов змін у навчальних даних.

Об'єктом дослідження є процес навчання моделей машинного навчання на вибірках даних різної структури та якості.

Метою кваліфікаційної роботи є дослідження стабільності та чутливості алгоритмів машинного навчання до змін у навчальних даних на основі проведення обчислювальних експериментів та порівняльного аналізу результатів з використанням формалізованих метрик оцінювання.

Актуальність кваліфікаційної роботи зумовлена широким застосуванням алгоритмів машинного навчання в умовах нестабільних, зашумлених і динамічних даних. Зміни у навчальних вибірках можуть істотно впливати на якість і надійність моделей, тому дослідження їх стабільності та чутливості є важливим завданням для підвищення робастності інтелектуальних систем.

У ході виконання кваліфікаційної роботи проведено аналіз предметної галузі, обґрунтовано вибір алгоритмів машинного навчання та метрик оцінювання, сформовано сценарії збурення даних і виконано серію експериментів. Отримані результати дозволили порівняти поведінку алгоритмів за різних змін даних, визначити найбільш стабільні підходи та сформулювати практичні рекомендації щодо їх застосування.

ABSTRACT

Bachelor's thesis contains: 107 pp., 13 fig., 10 tabl., 1 ann., 14 references.

ALGORITHMS OF MACHINE LEARNING, COMPUTATIONAL EXPERIMENT, DATA SHIFT, MACHINE LEARNING, ROBUSTNESS, ROBUSTNESS, SENSITIVITY, SENSITIVITY, STABILITY, STABILITY, TRAINING DATA.

The subject of the research is the properties of stability and sensitivity of machine learning algorithms that manifest themselves under conditions of changes in the training data.

The object of the research is the process of training machine learning models on datasets of varying structure and quality.

The aim of the qualification thesis is to investigate the stability and sensitivity of machine learning algorithms to changes in training data through computational experiments and comparative analysis of results using formalized evaluation metrics.

The relevance of the qualification thesis is обусловлена the widespread application of machine learning algorithms in environments characterized by unstable, noisy, and dynamic data. Changes in training datasets can significantly affect the quality and reliability of models. Therefore, the study of stability and sensitivity of machine learning algorithms is an important task aimed at improving the robustness and practical applicability of intelligent systems.

In the course of the qualification thesis, an analysis of the subject area was carried out, the choice of machine learning algorithms and evaluation metrics for stability and sensitivity was justified, scenarios of training data perturbations were developed, and a series of computational experiments was conducted. The obtained results made it possible to compare the behavior of the studied algorithms under different types of data changes, identify the most stable approaches, and formulate practical recommendations for their application in conditions of unstable training datasets.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі.....	9
1.1 Стабільність і чутливість алгоритмів машинного навчання	9
1.2 Джерела змін у навчальних даних та їх вплив на навчання	12
1.3 Огляд підходів до аналізу робастності та стабільності моделей	15
1.4 Постановка задачі дослідження.....	18
2 Вибір алгоритмів, метрик та проєктування експерименту	20
2.1 Критерії оцінювання стабільності та чутливості алгоритмів.....	20
2.2 Обґрунтування вибору алгоритмів машинного навчання	26
2.3 Вибір та характеристика наборів навчальних даних.....	33
2.4 Формування сценаріїв збурення навчальних даних	40
2.5 Основні метрики оцінювання стабільності і чутливості алгоритмів.....	47
2.6 План та протокол проведення обчислювального експерименту.....	54
3 Формування навчальних даних та проведення експериментів	57
3.1 Попередня обробка та підготовка даних	57
3.2 Формування збурених наборів даних	75
3.3 Проведення експерименту	89
3.4 Аналіз результатів експерименту	93
Висновки	102
Перелік джерел посилання	105
Додаток А Відомість кваліфікаційної роботи	107

ВСТУП

Стрімкий розвиток методів машинного навчання зумовив їх широке впровадження у різноманітні прикладні сфери, зокрема в аналіз даних, прогнозування, обробку зображень, фінансові системи, медицину та інформаційні сервіси. У більшості реальних задач алгоритми машинного навчання працюють в умовах неповної, зашумленої або змінної інформації, що істотно впливає на якість отриманих результатів. У зв'язку з цим особливої актуальності набуває проблема стабільності та чутливості алгоритмів машинного навчання до змін у навчальних даних.

Стабільність алгоритмів машинного навчання характеризує здатність моделей зберігати узгодженість результатів навчання та прогнозування за незначних змін у навчальній вибірці, тоді як чутливість відображає міру реакції моделей на різні типи збурень даних, такі як шум, викиди, дисбаланс класів або зміщення розподілу. Недостатня стабільність або надмірна чутливість можуть призводити до деградації якості прогнозів, втрати узагальнюючої здатності та зниження надійності моделей у практичних застосуваннях. Тому дослідження цих властивостей є важливим етапом оцінювання якості та придатності алгоритмів машинного навчання до використання в реальних інформаційних системах.

Актуальність даної кваліфікаційної роботи зумовлена зростанням обсягів даних, їх гетерогенністю та динамічним характером, а також необхідністю забезпечення надійності та робастності інтелектуальних систем. Незважаючи на значну кількість досліджень, присвячених підвищенню точності моделей, питання стабільності та чутливості алгоритмів машинного навчання часто залишаються другорядними або розглядаються фрагментарно. Це обумовлює потребу у систематичному аналізі поведінки різних алгоритмів за умов контрольованих змін навчальних даних із використанням формалізованих метрик оцінювання.

Метою кваліфікаційної роботи є дослідження стабільності та чутливості алгоритмів машинного навчання до змін у навчальних даних шляхом проведення серії обчислювальних експериментів та порівняльного аналізу отриманих результатів.

Для досягнення поставленої мети в роботі необхідно розв'язати такі завдання: проаналізувати предметну область та наявні підходи до оцінювання стабільності моделей машинного навчання; обґрунтувати вибір алгоритмів, що підлягають дослідженню; визначити набір метрик для кількісної оцінки стабільності та чутливості; сформувати навчальні набори даних і сценарії їх збурення; провести експериментальні дослідження та здійснити аналіз отриманих результатів; сформувати узагальнені висновки та практичні рекомендації.

Об'єктом дослідження є процес навчання моделей машинного навчання на вибірках даних.

Предметом дослідження є стабільність та чутливість алгоритмів машинного навчання до змін у навчальних даних.

Методи дослідження ґрунтуються на використанні методів машинного навчання, статистичного аналізу, обчислювальних експериментів, а також порівняльного аналізу результатів моделювання.

Практичне значення роботи полягає у можливості використання отриманих результатів і рекомендацій під час вибору та налаштування алгоритмів машинного навчання для задач, у яких навчальні дані є нестабільними або схильними до змін.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Стабільність і чутливість алгоритмів машинного навчання

Штучний інтелект є міждисциплінарною галуззю комп'ютерних наук, метою якої є створення програмних і апаратних систем, здатних виконувати завдання, що традиційно вимагають людського інтелекту. До таких завдань належать аналіз інформації, логічне мислення, навчання на основі досвіду, прийняття рішень, розпізнавання образів та обробка природної мови. Розвиток штучного інтелекту зумовлений зростанням обчислювальних потужностей, накопиченням великих обсягів даних і вдосконаленням алгоритмічних підходів до їх обробки.

У загальній структурі штучного інтелекту машинне навчання займає одне з ключових місць. Машинне навчання являє собою напрям, у межах якого розробляються методи та алгоритми, що дозволяють комп'ютерним системам автоматично виявляти закономірності в даних і вдосконалювати свою поведінку без явного програмування для кожного окремого випадку. Основною ідеєю машинного навчання є заміна жорстко заданих правил моделями, які формуються на основі аналізу навчальних даних.

Алгоритми машинного навчання будують математичні моделі, що відображають залежність між вхідними ознаками та цільовими змінними. Процес навчання полягає у підборі параметрів моделі таким чином, щоб мінімізувати різницю між прогнозованими та фактичними значеннями на навчальній вибірці. Отримана модель надалі використовується для прогнозування або прийняття рішень на нових, раніше не використаних даних, що зумовлює необхідність її здатності до узагальнення.

Залежно від характеру доступних даних та постановки задачі розрізняють основні парадигми машинного навчання: навчання з учителем, навчання без учителя та навчання з підкріпленням. У навчанні з учителем моделі навчаються на розмічених даних, де відомі правильні відповіді, у

навчанні без учителя – на неструктурованих даних без явних міток, а в навчанні з підкріпленням – на основі взаємодії з середовищем та системи винагород. Кожна з цих парадигм має свої особливості та галузі застосування.

Ефективність алгоритмів машинного навчання значною мірою залежить від якості та властивостей навчальних даних. Реальні дані часто містять шум, пропуски, викиди або можуть змінювати свій розподіл з часом. Такі фактори здатні негативно впливати на процес навчання та призводити до нестабільної поведінки моделей. У зв'язку з цим дедалі більшої уваги набувають не лише показники точності, а й властивості моделей, пов'язані з їх надійністю та передбачуваністю.

Одними з ключових характеристик алгоритмів машинного навчання в цьому контексті є стабільність та чутливість. Ці властивості відображають реакцію алгоритмів на зміни у навчальних даних та визначають, наскільки істотно такі зміни впливають на результати навчання і прогнозування. Аналіз стабільності та чутливості дозволяє глибше оцінити поведінку моделей у реальних умовах експлуатації, де дані рідко є статичними або ідеальними [1].

Стабільність алгоритмів машинного навчання є властивістю, що характеризує здатність моделі зберігати узгоджені результати навчання та прогнозування за умов незначних змін у навчальних даних. Йдеться про ситуації, коли невеликі модифікації навчальної вибірки, такі як видалення або заміна окремих прикладів, додавання слабого шуму чи зміна порядку даних, не призводять до суттєвих змін у параметрах моделі або її прогнозах.

З точки зору теорії машинного навчання стабільність часто пов'язують із залежністю алгоритму від конкретних елементів навчальної вибірки. Чим меншою є ця залежність, тим більш стабільним вважається алгоритм. Алгоритми з високою стабільністю зазвичай формують схожі моделі навіть за умови навчання на дещо різних вибірках, що сприяє кращій узагальнюючій здатності та зменшенню ризику перенавчання.

У практичному контексті стабільність алгоритмів машинного навчання проявляється у повторюваності результатів. Якщо модель навчається кілька разів на вибірках, отриманих шляхом незначних змін початкового датасету, стабільний алгоритм демонструє близькі значення метрик якості та подібну структуру прогнозів. Це є важливою характеристикою для систем, у яких необхідно забезпечити надійність і передбачуваність роботи.

Рівень стабільності алгоритму залежить від його внутрішньої структури та принципів навчання. Наприклад, регуляризовані лінійні моделі або ансамблеві методи зазвичай є більш стабільними, тоді як складні моделі з високою варіативністю параметрів можуть демонструвати підвищену чутливість до змін у даних. Таким чином, стабільність є ключовою властивістю, що визначає стійкість алгоритмів машинного навчання до незначних збурень навчальних вибірок і безпосередньо впливає на їх практичну придатність.

Чутливість алгоритмів машинного навчання є властивістю, що характеризує ступінь реакції моделі на зміни у навчальних даних. Вона відображає, наскільки суттєво змінюються параметри моделі, результати навчання або прогнозування у відповідь на варіації у складі, структурі чи статистичних властивостях навчальної вибірки. На відміну від стабільності, яка акцентує увагу на збереженні результатів, чутливість зосереджується на інтенсивності змін, що виникають у моделі внаслідок збурень даних.

У контексті машинного навчання чутливість проявляється у реакції алгоритмів на такі фактори, як додавання шуму до ознак, наявність викидів, помилки у мітках, зміна дисбалансу класів або зміщення розподілу даних. Висока чутливість означає, що навіть незначні зміни у навчальних даних можуть призводити до помітного погіршення якості моделі або суттєвої зміни її прогнозів. Низька чутливість, навпаки, свідчить про робастну поведінку алгоритму.

Стабільність алгоритмів машинного навчання – це властивість алгоритму зберігати узгоджені параметри моделі, прогнозні значення та показники якості за умов незначних змін у навчальних даних, таких як випадкове вилучення або заміна окремих прикладів, додавання слабого шуму чи повторне формування навчальної вибірки. Стабільний алгоритм демонструє мінімальну варіативність результатів навчання при багаторазовому повторенні експериментів у близьких умовах.

Чутливість алгоритмів машинного навчання – це властивість алгоритму, що характеризує ступінь зміни параметрів моделі, результатів прогнозування та значень метрик якості у відповідь на контрольовані збурення навчальних даних, зокрема шум, викиди, дисбаланс класів або зміщення розподілу. Висока чутливість означає значну реакцію моделі навіть на малі зміни даних, тоді як низька чутливість свідчить про робастну поведінку алгоритму.

1.2 Джерела змін у навчальних даних та їх вплив на навчання

У процесі побудови та використання моделей машинного навчання навчальні дані рідко залишаються сталими та однорідними. Навпаки, у реальних умовах дані зазнають різноманітних змін, які можуть виникати як на етапі збору та підготовки інформації, так і в процесі подальшої експлуатації системи. Такі зміни розглядаються як збурення навчальних даних і є одним з ключових чинників, що впливають на якість, стабільність та узагальнюючу здатність алгоритмів машинного навчання.

Одним із найпоширеніших джерел змін у навчальних даних є шум у вхідних ознаках. Шум може бути зумовлений похибками вимірювальних приладів, неточністю збору даних або випадковими флуктуаціями значень. Наявність шуму ускладнює процес виявлення закономірностей, оскільки модель змушена відокремлювати корисну інформацію від випадкових

складових. У результаті це може призводити до зниження точності прогнозів та підвищення варіативності результатів навчання [2].

Іншим важливим джерелом збурень є викиди, тобто поодинокі спостереження, значення яких суттєво відрізняються від основної маси даних. Викиди можуть виникати через помилки введення, збої сенсорів або рідкісні, але можливі події. Для багатьох алгоритмів машинного навчання наявність викидів є критичною, оскільки вони можуть істотно впливати на процес оптимізації параметрів і спричиняти зміщення моделі у бік нетипових спостережень [3].

Окрему групу джерел змін становлять помилки у цільових мітках, які часто виникають у процесі ручної розмітки даних або автоматизованого збору інформації. Неправильні мітки спотворюють залежність між ознаками та цільовими значеннями, що ускладнює навчання моделей і може призводити до систематичного погіршення якості прогнозування. Алгоритми з високою чутливістю до таких помилок демонструють істотну деградацію результатів навіть за невеликої частки помилкових міток [4].

Зміна обсягу та складу навчальної вибірки також є важливим фактором збурення. Зменшення кількості навчальних прикладів, випадкове вилучення частини даних або нерівномірне представлення окремих класів призводять до зростання дисперсії оцінок та підвищення ризику перенавчання. Особливо гостро ця проблема проявляється у задачах з дисбалансом класів, де модель може зміщуватися в бік домінуючого класу, ігноруючи менш представлені, але важливі категорії [5].

Суттєвий вплив на результати навчання мають також зміщення розподілу даних, коли статистичні властивості навчальної та тестової вибірок відрізняються. Такі зміщення можуть бути наслідком зміни умов збору даних, еволюції процесів у часі або появи нових типів об'єктів. У подібних умовах моделі, навчені на історичних даних, часто демонструють зниження узагальнюючої здатності, що особливо критично для систем, орієнтованих на довготривале використання [6].

Отже, основні джерела змін, що будуть розглядатися в рамках дослідження:

- шум у вхідних ознаках – випадкові або систематичні похибки у значеннях ознак, зумовлені неточністю вимірювань, збору або обробки даних;
- викиди (outliers) – поодинокі спостереження, що суттєво відрізняються від основної маси даних і можуть спотворювати процес навчання моделей;
- помилки у цільових мітках (label noise) – неправильна або неточна розмітка навчальних даних, що порушує відповідність між ознаками та цільовими значеннями;
- зміна обсягу навчальної вибірки – зменшення або варіація кількості навчальних прикладів, у тому числі випадкове вилучення частини даних;
- дисбаланс класів – нерівномірне представлення класів у навчальних даних, що призводить до зміщення моделі в бік домінуючих класів;
- зміщення розподілу даних (data shift) – відмінності між статистичними властивостями навчальної та тестової вибірок, зумовлені зміною умов збору даних.

Таким чином можна зазначити, що шум у вхідних ознаках, викиди, помилки у цільових мітках, зміна обсягу навчальної вибірки, дисбаланс класів і зміщення розподілу даних мають різну природу, проте спільно призводять до зниження якості моделей, втрати узагальнюючої здатності, підвищення чутливості та зменшення стабільності алгоритмів. Систематизація зазначених типів збурень дозволяє сформувати основу для подальшого експериментального дослідження поведінки алгоритмів машинного навчання в умовах нестабільних і змінних даних та обґрунтувати вибір сценаріїв збурення у подальшому дослідженні.

1.3 Огляд підходів до аналізу робастності та стабільності моделей

Сучасні підходи до аналізу робастності та стабільності моделей машинного навчання формувалися як відповідь на обмеженість традиційних метрик якості, орієнтованих переважно на точність прогнозування. У багатьох прикладних задачах стало очевидно, що висока точність на тестовій вибірці не гарантує надійної поведінки моделі за умов змінних або зашумлених даних. У зв'язку з цим у наукових дослідженнях дедалі більше уваги приділяється властивостям моделей, пов'язаним із їх стійкістю до збурень та повторюваністю результатів навчання [7].

Одним із базових теоретичних підходів до аналізу стабільності є дослідження залежності алгоритму від окремих елементів навчальної вибірки. У цьому контексті стабільність розглядається як міра того, наскільки зміна або видалення одного навчального прикладу впливає на результати навчання. Такі підходи тісно пов'язані з теорією узагальнення та дозволяють встановити зв'язок між стабільністю алгоритму і його здатністю до ефективного навчання на нових даних.

Практично орієнтовані підходи до аналізу стабільності ґрунтуються на багаторазовому навчанні моделей на різних варіантах навчальних вибірок. Для цього використовуються методи перехресної перевірки, бутстрепінгу або повторного випадкового підвибору даних. Стабільність оцінюється за варіативністю параметрів моделей, прогнозів або значень метрик якості між окремими запусками. Такий підхід дозволяє кількісно оцінити повторюваність результатів і є зручним для порівняльного аналізу різних алгоритмів.

Аналіз робастності моделей зазвичай здійснюється шляхом навмисного внесення збурень у навчальні або тестові дані. До таких збурень належать додавання шуму, викидів, помилок у мітках, зміна дисбалансу класів або імітація зміщення розподілу даних. Робастність у цьому випадку оцінюється за ступенем деградації якості моделі залежно від рівня та типу

збурення. Чим повільніше знижується якість, тим більш робастною вважається модель.

Окрему групу сучасних підходів становлять методи аналізу стабільності прогнозів та внутрішніх характеристик моделей. До них належить порівняння передбачень різних моделей, навчених на збурених вибірках, а також аналіз стабільності важливості ознак або параметрів моделі. Такі підходи є особливо актуальними для інтерпретованих моделей, де важливо не лише отримати точний прогноз, а й забезпечити сталість пояснень.

Подальший розвиток підходів до аналізу робастності та стабільності моделей машинного навчання пов'язаний із використанням спеціалізованих метрик і кількісних індикаторів. Окрім традиційних показників якості, таких як точність або похибка, застосовуються метрики, що безпосередньо відображають зміну результатів під впливом збурень. До них належать відносна деградація якості, дисперсія значень метрик між експериментами, а також швидкість зміни показників якості залежно від інтенсивності збурення. Такі метрики дозволяють формалізувати поняття чутливості та порівнювати алгоритми за характером їх реакції на зміни даних.

Важливим напрямом сучасних досліджень є аналіз поведінки моделей у сценаріях зміщення розподілу даних. У цьому випадку стабільність і робастність оцінюються не лише на основі навчальної вибірки, а й з урахуванням відмінностей між навчальними та тестовими даними. Для цього використовуються контрольовані експерименти зі зміною розподілу ознак або цільових змінних, що дозволяє дослідити здатність моделей зберігати прийнятний рівень якості в динамічних умовах. Такі підходи є особливо актуальними для систем, які працюють у реальному часі або в середовищах, що змінюються.

Окрему увагу в сучасних підходах приділяють порівняльному аналізу різних класів алгоритмів машинного навчання з точки зору їх стабільності та робастності. Дослідження показують, що простіші або регуляризовані

моделі зазвичай демонструють вищу стабільність, тоді як складні нелінійні алгоритми можуть бути більш чутливими до збурень даних. Водночас ансамблеві методи часто поєднують високу якість прогнозування з підвищеною робастністю, що робить їх привабливими для практичного використання.

Таким чином, сучасні підходи до аналізу робастності та стабільності моделей машинного навчання базуються на поєднанні теоретичних міркувань і експериментальних досліджень. Вони передбачають систематичне внесення збурень у дані, багаторазове навчання моделей, використання спеціалізованих метрик і порівняльний аналіз різних алгоритмів. Узагальнення цих підходів створює методологічну основу для побудови експериментів у межах даної кваліфікаційної роботи та дозволяє обґрунтовано оцінювати стабільність і чутливість алгоритмів машинного навчання.

Для наочного представлення підходів, що застосовуються в даному дослідженні, основні способи оцінювання стабільності та чутливості алгоритмів машинного навчання можна узагальнити таким чином:

- багаторазове навчання моделей на злегка модифікованих навчальних вибірках із фіксованими гіперпараметрами;
- оцінювання варіативності показників якості моделей між окремими запусками навчання;
- аналіз узгодженості прогнозів моделей, навчених на близьких за складом вибірках;
- внесення контрольованих збурень у навчальні дані з різними рівнями інтенсивності;
- аналіз деградації якості моделей відносно базового сценарію без збурень;
- дослідження залежності зміни показників якості від ступеня збурення навчальних даних.

Такий перелік дозволяє структуровано відобразити методологію оцінювання стабільності та чутливості, що використовується у межах даної кваліфікаційної роботи.

1.4 Постановка задачі дослідження

У даному дослідженні розглядається задача оцінювання стабільності та чутливості алгоритмів машинного навчання в умовах змін навчальних даних. Постановка задачі ґрунтується на припущенні, що у реальних інформаційних системах навчальні вибірки не є статичними та можуть зазнавати різноманітних збурень, які впливають на результати навчання і якість прогнозування моделей. Тому виникає необхідність формалізовано дослідити, як різні алгоритми реагують на такі зміни та які з них демонструють більш надійну поведінку.

Задача дослідження полягає у порівняльному аналізі декількох алгоритмів машинного навчання з точки зору їх здатності зберігати стабільні результати та адекватно реагувати на контрольовані збурення навчальних даних. Для цього передбачається навчання моделей на базових наборах даних без збурень, а також на їх модифікованих версіях, отриманих шляхом внесення різних типів змін. Результати навчання на цих вибірках використовуються як основа для подальшого кількісного аналізу.

У межах дослідження стабільність алгоритмів розглядається як повторюваність результатів навчання за умов незначних змін складу навчальної вибірки. Чутливість, у свою чергу, розглядається як міра деградації якості моделей та зміни їх прогнозів у відповідь на зростання інтенсивності збурень даних. Таким чином, задача формалізується як оцінювання залежності показників якості та характеристик моделей від типу і рівня змін у навчальних даних.

Для розв'язання поставленої задачі необхідно визначити набір алгоритмів машинного навчання, що представляють різні підходи до

моделювання, обрати репрезентативні набори даних та сформулювати сценарії збурення навчальних вибірок. Далі проводиться серія обчислювальних експериментів із фіксованим протоколом навчання та оцінювання, що забезпечує коректність порівняння результатів між алгоритмами.

Постановка задачі дослідження полягає у систематичному аналізі впливу змін у навчальних даних на стабільність і чутливість алгоритмів машинного навчання з використанням формалізованих метрик та контрольованих експериментальних сценаріїв. Отримані результати мають слугувати основою для узагальнених висновків щодо надійності та практичної придатності досліджуваних алгоритмів в умовах нестабільних даних.

2 ВИБІР АЛГОРИТМІВ, МЕТРИК ТА ПРОЄКТУВАННЯ ЕКСПЕРИМЕНТУ

2.1 Критерії оцінювання стабільності та чутливості алгоритмів

У межах дослідження стабільність і чутливість алгоритмів машинного навчання розглядаються як взаємопов'язані, але концептуально відмінні характеристики якості моделі. Стабільність відображає здатність алгоритму зберігати узгоджені результати за незначних змін у навчальних даних або умовах навчання, тоді як чутливість характеризує реакцію моделі на контрольовані збурення вхідних ознак або параметрів. Обидва критерії є критично важливими для практичного застосування моделей, оскільки дозволяють оцінити їхню надійність, відтворюваність та стійкість до шуму й невизначеності реальних даних.

Оцінювання стабільності алгоритмів здійснюється шляхом аналізу варіативності результатів моделі при повторному навчанні на різних вибірках, сформованих на основі одних і тих самих даних. Для цього використовуються методи повторної вибірки, зокрема перехресна перевірка, бутстрепінг або навчання на підвибірках зі зміненим складом об'єктів. Ключовим критерієм стабільності виступає міра розбіжності між прогнозами або значеннями обраних метрик якості, отриманими в різних ітераціях. Низька дисперсія таких показників свідчить про високу стабільність алгоритму.

Чутливість моделей оцінюється через внесення контрольованих змін у вхідні дані або параметри навчання з подальшим аналізом впливу цих змін на вихідні результати. До таких змін належать додавання шуму до ознак, масштабування або спотворення окремих параметрів, а також модифікація розподілу вхідних даних. Критеріями чутливості виступають темп деградації якості моделі та характер зміни прогнозів у відповідь на збурення. Алгоритми з низькою чутливістю демонструють плавну та

прогнозовану реакцію, що є бажаною властивістю для задач з високою варіативністю даних.

Стабільність і чутливість алгоритмів машинного навчання пропонується оцінювати за допомогою формалізованих кількісних метрик і процедур, які дозволяють отримати відтворювані та порівнювані результати. Нижче наведено основні способи оцінювання, що застосовуються в сучасних дослідженнях робастності моделей.

Першим способом є дисперсія метрик якості при повторному навчанні.

Оцінювання стабільності алгоритмів машинного навчання за допомогою дисперсії метрик якості при повторному навчанні ґрунтується на ідеї аналізу варіативності результатів моделі за умови незначних змін у складі навчальних даних.

Практична реалізація даного підходу полягає у багаторазовому навчанні однієї й тієї самої моделі з однаковими гіперпараметрами, але на різних навчальних вибірках, сформованих за допомогою методів перехресної перевірки або випадкової підвибірki. Для кожної ітерації обчислюється обрана метрика якості, наприклад точність, F1-міра або середньоквадратична помилка, залежно від типу задачі. Отримана сукупність значень метрики використовується для подальшого статистичного аналізу.

Другим способом є аналіз стабільності прогнозів.

Аналіз стабільності прогнозів є одним із безпосередніх способів оцінювання стабільності алгоритмів машинного навчання, оскільки він зосереджується не на узагальнених метриках якості, а на узгодженості індивідуальних передбачень моделі.

Реалізація цього підходу передбачає багаторазове навчання моделі з фіксованою архітектурою та гіперпараметрами, після чого для кожного запуску формуються прогнози на одній і тій самій тестовій вибірці.

Отримані вектори прогнозів порівнюються між собою за допомогою спеціалізованих метрик узгодженості.

Перевагою аналізу стабільності прогнозів є його здатність виявляти нестабільність, яка може залишатися непомітною при використанні лише агрегованих метрик якості. Навіть за близьких значень точності або похибки модель може демонструвати значні коливання у прогнозах для окремих об'єктів, що є критичним для практичних застосувань, зокрема в задачах підтримки прийняття рішень або ризик-орієнтованих системах.

Третім способом є оцінювання чутливості до шуму у вхідних даних.

Оцінювання чутливості до шуму у вхідних даних є одним із базових підходів до аналізу робастності алгоритмів машинного навчання, оскільки реальні дані майже завжди містять похибки вимірювання, випадкові збурення або неповну інформацію.

Методологія оцінювання полягає у штучному додаванні шуму до вхідних ознак тестової або навчальної вибірки з подальшим аналізом зміни значень обраної метрики якості.

Кількісним показником чутливості в межах цього підходу є відносна або абсолютна втрата якості моделі порівняно з базовим сценарієм без шуму. Для кожного рівня збурення обчислюється значення метрики якості, після чого аналізується характер її зміни залежно від інтенсивності шуму. Швидке зниження якості свідчить про високу чутливість алгоритму, тоді як поступова деградація вказує на більш робастну поведінку моделі.

Перевагою даного способу є його універсальність і наочність, оскільки він не залежить від типу алгоритму та може бути однаково застосований як до класичних моделей, так і до нейромережових підходів.

Четвертим способом є оцінка локальної чутливості за градієнтами.

Оцінка локальної чутливості за градієнтами ґрунтується на аналізі того, наскільки сильно змінюється вихід моделі у відповідь на малі збурення вхідних ознак.

Практична реалізація даного підходу передбачає обчислення градієнтів функції втрат за вхідними ознаками для кожного об'єкта тестової або навчальної вибірки. Отримані градієнтні вектори узагальнюються шляхом обчислення їх норм, найчастіше L1 або L2, після чого аналізується їх розподіл по вибірці. Середні або медіанні значення норм градієнтів використовуються як кількісні показники локальної чутливості моделі.

П'ятим способом є перевірка стійкості до зміни розподілу даних.

Перевірка стійкості до зміни розподілу даних є важливим підходом до оцінювання чутливості алгоритмів машинного навчання, оскільки в реальних умовах експлуатації припущення про незмінність розподілу даних часто порушується

Методологічно даний підхід передбачає навчання моделі на вибірці з базовим розподілом та її подальше тестування на даних, розподіл яких свідомо відрізняється від навчального. Такі відмінності можуть реалізовуватися шляхом зміни діапазонів значень ознак, порушення пропорцій між класами, виключення або додавання підмножин даних, а також використання даних з іншого часового або контекстного періоду. При цьому важливо, щоб тип і рівень зсуву розподілу були однаковими для всіх алгоритмів, що порівнюються.

Шостим способом є бутстреп-аналіз стабільності параметрів моделі.

Бутстреп-аналіз стабільності параметрів моделі є статистичним підходом до оцінювання стабільності алгоритмів машинного навчання, який базується на багаторазовому навчанні моделі на вибірках, сформованих шляхом випадкового відбору з поверненням із початкового набору даних.

Практична реалізація цього підходу передбачає формування великої кількості бутстреп-вбірок однакового розміру з початкового датасету та незалежне навчання моделі на кожній з них. Для кожного запуску зберігаються параметри моделі, такі як коефіцієнти регресії, ваги ознак або інші інтерпретовані характеристики. Отриманий набір параметрів використовується для подальшого аналізу їх варіативності.

Сьомим можливим способом є оцінка Lipschitz-стабільності.

Оцінка Lipschitz-стабільності є теоретично обґрунтованим підходом до аналізу чутливості алгоритмів машинного навчання, який базується на понятті Lipschitz-неперервності відображення між просторами вхідних і вихідних даних. Даний підхід дозволяє формально описати верхню межу зміни виходу моделі при малих варіаціях вхідного простору, що є ключовим аспектом аналізу робастності моделей.

Практична інтерпретація цього підходу полягає в оцінюванні того, наскільки чутливо модель реагує на незначні зміни ознак, зокрема шум, похибки вимірювань або варіативність даних. Для окремих класів моделей, таких як лінійні моделі, методи опорних векторів або нейронні мережі з обмеженнями на вагові коефіцієнти, Lipschitz-константа може бути отримана аналітично або наближено через нормування параметрів. У більш загальному випадку, особливо для складних нелінійних моделей, застосовуються чисельні методи оцінювання або емпіричні підходи, що базуються на статистичному аналізі поведінки моделі на варіаціях вхідних даних.

У подальшому дослідженні для оцінювання стабільності та чутливості алгоритмів машинного навчання було обрано чотири взаємодоповнюючі метрики та відповідні методи аналізу, які забезпечують баланс між теоретичною обґрунтованістю, практичною інтерпретованістю та можливістю відтворення експериментів. Такий підхід дозволяє комплексно оцінювати поведінку моделей у різних умовах експлуатації.

Вибір зазначених метрик зумовлений необхідністю оцінювання як глобальної стабільності моделей, так і їх локальної чутливості до змін у даних. Окремо враховується як вплив варіативності навчальної вибірки, так і стійкість до зовнішніх збурень та змін розподілу даних, що є критично важливим для реальних прикладних задач машинного навчання.

Нижче наведена таблиця 2.1, у якій систематизовано чотири обрані метрики, їх призначення та роль у дослідженні, з метою забезпечення структурованого аналізу стабільності моделей.

Таблиця 2.1 – Систематизовані метрики, обрані для дослідження

Метрика та спосіб оцінювання	Характеристика	Що саме оцінюється	Тип збурення та варіації	Причина використання в дослідженні
Дисперсія метрик якості при повторному навчанні	Стабільність	Варіативність значень метрики якості (Accuracy, F1, MSE тощо) між повторними запусками навчання	Зміна складу навчальної вибірки	Дає кількісну оцінку відтворюваності результатів і є універсальною для всіх алгоритмів
Узгодженість прогнозів між повторними запусками	Стабільність	Частка співпадінь або відстань між векторами прогнозів	Зміна навчальних даних та стохастичність навчання	Дозволяє виявити нестабільність, не помітну через агреговані метрики якості
Відносна втрата якості при додаванні шуму	Чутливість	Темп деградації метрики якості зі зростанням рівня шуму	Додавання випадкового шуму до вхідних ознак	Моделює реальні похибки даних і наочно відображає робастність моделей
Зниження якості при зміні розподілу даних	Чутливість	Стійкість моделі до distribution shift	Зміна статистичних властивостей тестової вибірки	Наближує експеримент до реальних умов експлуатації моделей

Першою обраною метрикою є дисперсія значень метрик якості при повторному навчанні моделі. Вона використовується як базовий індикатор стабільності алгоритму щодо варіацій у складі навчальної вибірки та випадкових ініціалізацій параметрів. Низьке значення цієї метрики свідчить про високу відтворюваність результатів моделі.

Другою метрикою є рівень узгодженості прогнозів між повторними запусками моделі, який відображає стабільність на рівні індивідуальних

передбачень. Дана метрика дозволяє оцінити, наскільки консистентними є результати моделі для однакових вхідних даних при різних умовах навчання.

Третьою метрикою є відносна втрата якості при додаванні шуму у вхідні дані, яка характеризує чутливість алгоритму до похибок вимірювань, випадкових збурень та деградації якості даних. Вона фактично дозволяє оцінити робастність моделі до стохастичних викривлень вхідного простору.

Четвертою метрикою є зниження якості моделі при зміні розподілу даних, що використовується для оцінювання здатності алгоритмів до узагальнення в умовах *distribution shift*. Дана метрика дозволяє аналізувати деградацію продуктивності моделі при переході від навчального до іншого статистично відмінного середовища.

Обрані метрики взаємно доповнюють одна одну та охоплюють різні рівні аналізу стабільності й чутливості моделей машинного навчання – від локальної реакції на збурення до глобальної відтворюваності результатів. Їх спільне використання дозволяє сформувати цілісну оцінку робастності алгоритмів у межах єдиного експериментального протоколу, забезпечуючи коректність порівняння та надійність отриманих висновків.

2.2 Обґрунтування вибору алгоритмів машинного навчання

Вибір алгоритмів машинного навчання для дослідження стабільності та чутливості є ключовим етапом проєктування експерименту, оскільки саме характеристики обраних моделей визначають інтерпретованість і узагальнюваність отриманих результатів. У межах даної роботи алгоритми підбиралися таким чином, щоб вони репрезентували різні підходи до навчання, мали відмінні механізми узагальнення та по-різному реагували на зміни у вхідних даних і навчальній вибірці. Це дозволяє дослідити не окремі реалізації, а загальні властивості класів алгоритмів.

Першим критерієм відбору алгоритмів є різноманітність модельних парадигм. До дослідження доцільно включати як прості параметричні моделі, так і більш складні нелінійні методи. Такий підхід забезпечує можливість порівняння стабільності моделей із різною кількістю параметрів, різною гнучкістю та різним рівнем схильності до перенавчання. Це особливо важливо в контексті аналізу дисперсії метрик якості та узгодженості прогнозів.

Другим критерієм є відмінність у чутливості алгоритмів до шуму та зміни розподілу даних, що впливає з їхньої внутрішньої структури. Алгоритми, які ґрунтуються на локальних правилах або складних нелінійних залежностях, як правило, по-різному реагують на збурення у вхідних ознаках порівняно з більш простими моделями. Включення таких алгоритмів у дослідження дозволяє наочно простежити зв'язок між архітектурою моделі та її робастністю.

Третім важливим аспектом є поширеність і практична значущість алгоритмів, що досліджуються. Обрані методи повинні активно застосовуватися у прикладних задачах машинного навчання, оскільки це підвищує цінність отриманих висновків з точки зору реальних сценаріїв використання. Аналіз стабільності й чутливості саме таких алгоритмів дозволяє зробити результати дослідження більш релевантними для практики.

Четвертим критерієм є можливість коректного й відтворюваного експериментального налаштування. Алгоритми мають бути реалізовані з використанням стандартних бібліотек, мати чітко визначені гіперпараметри та дозволяти багаторазове навчання за однакових умов. Це є необхідною умовою для коректного обчислення дисперсії метрик якості, аналізу стабільності прогнозів та оцінювання чутливості до збурень даних.

Розгляд принципово різних алгоритмів машинного навчання створює необхідну основу для подальшого групування моделей і вибору репрезентативних алгоритмів у кожній категорії. Це дозволяє здійснити

комплексне дослідження стабільності та чутливості, орієнтоване не на окремі реалізації, а на фундаментальні властивості різних підходів до машинного навчання.

Доцільно буде обрати 4 групи, які є найбільш доцільними для аналізу стабільності та чутливості. Вибір цих груп зумовлений необхідністю охопити різні парадигми побудови моделей, забезпечити контрастність їх поведінки за умов збурень даних та водночас зберегти керовану складність експерименту. Обрані групи дозволяють дослідити як прості, так і складні моделі, а також порівняти локальні й глобальні механізми узагальнення.

Першою обраною групою є лінійні параметричні моделі. Дана група використовується як базовий еталон стабільності, оскільки такі алгоритми мають обмежену гнучкість, невелику кількість параметрів і добре досліджені теоретичні властивості. Лінійні моделі зазвичай демонструють високу відтворюваність результатів при повторному навчанні та помірну чутливість до шуму, що робить їх зручними для порівняння з більш складними підходами. Їх включення дозволяє чітко простежити, як зростання складності моделі впливає на стабільність [9].

Другою групою є деревоподібні алгоритми прийняття рішень. Ця група була обрана через характерну нестабільність окремих дерев рішень, яка проявляється у значній залежності структури моделі від навчальних даних. Навіть незначні зміни у вибірці можуть призводити до різних розбиттів простору ознак, що робить такі алгоритми показовими для аналізу чутливості до варіацій даних. Включення цієї групи дозволяє дослідити алгоритми з високою інтерпретованістю, але потенційно низькою стабільністю [10].

Третьою групою є ансамблеві методи на основі дерев рішень. Вибір цієї категорії обґрунтований тим, що ансамблі принципово змінюють поведінку базових нестабільних моделей за рахунок агрегування результатів. Вони часто демонструють знижену дисперсію та підвищену робастність порівняно з одиничними деревами. Аналіз ансамблевих методів

дозволяє дослідити, якою мірою колективне навчання компенсує нестабільність окремих моделей та як це відображається на чутливості до шуму й зміни розподілу даних [11].

Четвертою обраною групою є нейромережеві моделі. Їх включення є принципово важливим з огляду на високу складність, велику кількість параметрів і широке застосування в сучасних системах машинного навчання. Нейронні мережі поєднують високу апроксимаційну здатність із підвищеною чутливістю до ініціалізації, гіперпараметрів і якості даних. Саме тому вони є показовим прикладом моделей, для яких проблема стабільності та робастності є особливо актуальною [12].

У межах кожної з чотирьох обраних груп машинного навчання було визначено по одному репрезентативному алгоритму, який найкраще відображає характерні властивості відповідної парадигми та є доцільним для дослідження стабільності й чутливості. Вибір конкретних алгоритмів здійснювався з урахуванням їх поширеності, зрозумілої інтерпретації результатів і можливості коректного експериментального налаштування.

Для групи лінійних параметричних моделей у дослідженні обрано логістичну регресію, у групі деревоподібних алгоритмів прийняття рішень як представницький метод обрано дерево рішень, для групи ансамблевих методів у дослідженні використовується випадковий ліс, у групі нейромережевих моделей як базовий алгоритм обрано багат шаровий перцептрон.

Нижче наведено узагальнювальну таблицю 2.2, яка відображає, які саме сегменти дослідження стабільності та чутливості покриває кожен з обраних алгоритмів машинного навчання. Таблиця використовується як логічне підґрунтя експериментального дизайну та демонструє роль кожного алгоритму в дослідженні. Подана таблиця узагальнює роль кожного з обраних алгоритмів машинного навчання в дослідженні стабільності та чутливості моделей. Вона демонструє, які аспекти експерименту покриває кожен алгоритм, зокрема стабільність при повторному навчанні,

узгодженість прогнозів, чутливість до шуму у вхідних даних та стійкість до зміни розподілу даних. Такий підхід дозволяє чітко структурувати експеримент і забезпечити репрезентативність порівняльного аналізу.

Таблиця 2.2 – Порівняння алгоритмів машинного навчання

Алгоритм машинного навчання	Стабільність при повторному навчанні (дисперсія метрик)	Стабільність прогнозів	Чутливість до шуму у вхідних даних	Стійкість до зміни розподілу даних	Роль у дослідженні
Логістична регресія	Висока, використовується як еталон стабільності	Висока узгодженість прогнозів	Помірна, добре контрольована регуляризацією	Обмежена, залежить від лінійності розподілу	Базовий орієнтир для порівняння складніших моделей
Дерево рішень	Низька, висока варіативність між запусками	Низька, значні відмінності прогнозів	Висока чутливість, особливо без обмежень глибини	Низька, сильно залежить від навчальних розподілу	Показовий приклад нестабільного нелінійного алгоритму
Випадковий ліс	Помірна–висока, знижена дисперсія за рахунок ансамблю	Висока, завдяки агрегуванню прогнозів	Помірна, стійкіший до шуму ніж дерево рішень	Помірна, залежить від репрезентативності вибірки	Аналіз впливу ансамблювання на стабільність
Багатошаровий перцептрон	Змінна, залежить від ініціалізації та оптимізації	Змінна, можливі відмінності між запусками	Потенційно висока без регуляризації	Помірна, чутлива до distribution shift	Представник високопараметричних складних моделей

З таблиці видно, що алгоритми істотно відрізняються за характером поведінки в різних сегментах дослідження. Логістична регресія використовується як стабільний базовий орієнтир, дерево рішень – як приклад нестабільного нелінійного алгоритму, випадковий ліс – для аналізу впливу ансамблювання, а багатошаровий перцептрон – як представник складних високопараметричних моделей. У сукупності це забезпечує комплексне охоплення ключових аспектів стабільності та чутливості алгоритмів машинного навчання.

Під час формування експериментальної бази дослідження важливим етапом є вибір наборів навчальних даних, на яких будуть проводитися експерименти. Характеристики датасетів безпосередньо впливають на процес навчання моделей машинного навчання, а також на результати оцінювання їх стабільності та чутливості. Тому вибір наборів даних має здійснюватися з урахуванням низки критеріїв, що забезпечують можливість проведення коректного експериментального дослідження.

Для систематизації таких критеріїв було сформовано їх ієрархічну структуру залежно від ступеня важливості. Основні групи критеріїв вибору датасетів, а також їх зміст і взаємозв'язки наведено на рисунку 2.1.

У структурі діаграми всі критерії поділено на три основні групи відповідно до ступеня їх впливу на можливість проведення дослідження: критерії високої важливості, критерії середньої важливості та критерії додаткової важливості. Такий підхід дозволяє систематизувати вимоги до датасетів і визначити їх пріоритетність.

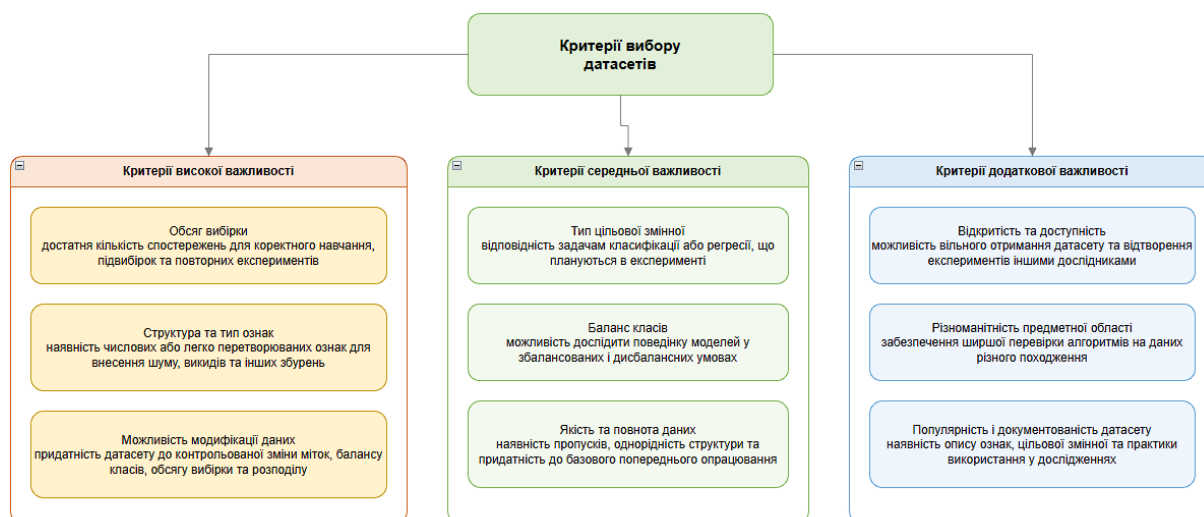


Рисунок 2.1 – Основні критерії вибору датасетів

До групи критеріїв високої важливості належать характеристики, без яких проведення експериментального дослідження було б ускладненим або

неможливим. Насамперед це достатній обсяг вибірки, який забезпечує наявність великої кількості спостережень для коректного навчання моделей та проведення повторних експериментів. Також важливою є структура та тип ознак, зокрема наявність числових або легко перетворюваних ознак, що дозволяє вносити контрольовані збурення, такі як шум або викиди. Окремим важливим критерієм є можливість модифікації даних, яка забезпечує створення експериментальних сценаріїв зі зміною міток, балансу класів, обсягу вибірки та розподілу даних.

До критеріїв середньої важливості належать характеристики, що визначають зручність використання датасету та впливають на коректність інтерпретації результатів дослідження. До них відноситься тип цільової змінної, який має відповідати задачам машинного навчання, що досліджуються, зокрема задачам класифікації або регресії. Також важливим є баланс класів, який дозволяє аналізувати поведінку моделей як у збалансованих, так і в дисбалансних умовах. Додатково враховується якість та повнота даних, що включає наявність пропусків, однорідність структури даних та можливість їх коректної попередньої обробки.

До групи критеріїв додаткової важливості належать характеристики, що сприяють відтворюваності та узагальнюваності результатів дослідження. Серед них важливим є відкритість і доступність датасетів, що забезпечує можливість повторення експериментів іншими дослідниками. Також враховується різноманітність предметної галузі, яка дозволяє перевіряти алгоритми машинного навчання на даних різного походження. Крім того, до уваги береться популярність і документованість датасету, зокрема наявність опису ознак, цільової змінної та попереднього досвіду використання набору даних у наукових дослідженнях.

Для проведення експериментального дослідження набори навчальних даних було вирішено обирати з використанням платформи Kaggle [13]. Ця платформа є одним із найбільших відкритих онлайн-ресурсів у сфері аналізу

даних та машинного навчання, що об'єднує дослідників, аналітиків та розробників з усього світу.

Важливим фактором вибору саме цієї платформи є також практична орієнтованість представлених на ній наборів даних. Значна частина датасетів Kaggle використовується у реальних дослідницьких або навчальних задачах машинного навчання. Це дозволяє використовувати перевірені набори даних, які вже застосовувалися у різних експериментах і добре підходять для аналізу поведінки алгоритмів машинного навчання.

Крім того, датасети, розміщені на платформі Kaggle, зазвичай мають зручний формат представлення, що дозволяє легко виконувати їх попередню обробку та модифікацію. Це є особливо важливим для даного дослідження, оскільки передбачається внесення контрольованих змін у навчальні дані, зокрема додавання шуму, створення викидів, зміну балансу класів та модифікацію обсягу навчальної вибірки. Наявність структурованих табличних даних значно спрощує реалізацію таких експериментальних сценаріїв.

2.3 Вибір та характеристика наборів навчальних даних

Вибір наборів навчальних даних є важливим етапом підготовки експериментального дослідження у сфері машинного навчання. Саме характеристики даних значною мірою визначають поведінку алгоритмів під час навчання та впливають на результати аналізу стабільності й чутливості моделей. Тому під час формування експериментальної бази необхідно враховувати низку критеріїв, які забезпечують репрезентативність даних та можливість проведення коректного порівняльного аналізу алгоритмів.

Одним із ключових критеріїв є достатній обсяг навчальної вибірки, структура та тип ознак, що входять до складу набору даних, наявність чітко визначеної цільової змінної, баланс класів у наборі даних, можливість модифікації даних для створення контрольованих збурень, відкритість і

доступність наборів даних, різноманітність предметних областей, представлених у наборі даних

Для проведення експериментального дослідження набори навчальних даних було вирішено обирати з використанням платформи Kaggle. Ця платформа є одним із найбільших відкритих онлайн-ресурсів у сфері аналізу даних та машинного навчання, що об'єднує дослідників, аналітиків та розробників з усього світу. Kaggle надає доступ до великої кількості відкритих датасетів, інструментів для їх аналізу, а також прикладів використання алгоритмів машинного навчання у різних прикладних задачах.

Для проведення експериментального дослідження було обрано три набори навчальних даних, розміщених на платформі Kaggle. Вибір саме трьох датасетів дозволяє перевірити поведінку алгоритмів машинного навчання у різних предметних областях та на даних з різною структурою. До дослідження було включено датасети Titanic, Credit Card Fraud Detection та Wine Quality Dataset, які широко використовуються у задачах машинного навчання та мають достатній обсяг і структуру для проведення експериментів зі зміною навчальних даних.

Датасет Titanic містить інформацію про пасажирів однойменного корабля, що затонув у 1912 році. У наборі даних представлені різні характеристики пасажирів, зокрема їхній вік, стать, клас каюти, вартість квитка та інші соціально-демографічні параметри [14]. Цільовою змінною є факт виживання пасажирів, що дозволяє розглядати даний датасет як задачу бінарної класифікації.

Другим обраним набором даних є Credit Card Fraud Detection, який містить інформацію про транзакції банківських карток та використовується для задачі виявлення шахрайських операцій [15]. У наборі даних представлено велику кількість транзакцій, кожна з яких описується набором числових ознак. Цільовою змінною є індикатор того, чи є транзакція шахрайською.

Третім набором даних, використаним у дослідженні, є Wine Quality Dataset, саме він містить фізико-хімічні характеристики різновидних зразків вина [16]. До таких характеристик належать рівень кислотності, вміст цукру, щільність, рН, вміст алкоголю та інші параметри. На основі цих ознак визначається оцінка якості вина, що може використовуватися як цільова змінна у задачах класифікації або регресії.

Для більш детального аналізу структури обраного набору даних доцільно розглянути перелік ознак, що входять до його складу. Кожен запис у датасеті відповідає окремому пасажирові та містить набір характеристик, які описують соціальні, демографічні та організаційні параметри подорожі. Наявність різних типів ознак, зокрема числових і категоріальних, створює можливість досліджувати поведінку алгоритмів машинного навчання у різних умовах представлення даних.

Структуру набору даних Titanic, а саме перелік основних стовпців та їх представлення у таблиці, наведено на рисунку 2.2. Після цього розглянемо зміст кожної з ознак, що входять до складу датасету, та їх можливу роль у задачі прогнозування виживання пасажирів.

Passenger Id ¹	Survived ¹	Pclass ¹	Name ¹	Sex ¹	Age ¹	Sib Sp ¹	Parch ¹	Ticket ¹	Fare ¹	Cabin ¹	Embarked ¹
892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.83		Q
893	1	3	Wilkes, Mrs. Jam	female	47	1	0	363272	7		S
894	0	2	Myles, Mr. Thor	male	62	0	0	240276	9.69		Q
895	0	3	Wirz, Mr. Albert	male	27	0	0	315154	8.66		S
896	1	3	Hirvonen, Mrs. A	female	22	1	1	3101298	12.29		S
897	0	3	Svensson, Mr. Jo	male	14	0	0	7538	9.23		S
898	1	3	Connolly, Miss. I	female	30	0	0	330972	7.63		Q
899	0	2	Caldwell, Mr. All	male	26	1	1	248738	29		S
900	1	3	Abraham, Mrs. Jc	female	18	0	0	2657	7.23		C
901	0	3	Davies, Mr. John	male	21	2	0	A/4 48871	24.15		S

Рисунок 2.2 – Структура набору даних Titanic

PassengerId – унікальний ідентифікатор пасажирів в наборі даних. Це службове поле використовується для однозначного розрізнення записів і не несе змістового навантаження для навчання моделі.

Survived – цільова змінна датасету, що відображає факт виживання пасажирів під час катастрофи. Значення 1 означає, що пасажир вижив, а значення 0 – що пасажир загинув.

Pclass – клас каюти пасажирів на кораблі. Значення 1 відповідає першому класу, 2 – другому, а 3 – третьому. Цей параметр відображає соціально-економічний статус пасажирів та може впливати на ймовірність виживання.

Name – текстове поле, що містить повне ім'я пасажирів. У цьому полі також можуть бути присутні титули, наприклад Mr., Mrs., Miss або Master, які іноді використовуються для додаткового аналізу соціального статусу або вікової категорії.

Sex – стать пасажирів. Значення зазвичай представлені як male або female. Дана ознака є однією з важливих характеристик, що впливають на ймовірність виживання.

Age – вік пасажирів у роках. Це числова змінна, яка дозволяє аналізувати залежність виживання від віку людини.

SibSp – кількість братів, сестер або подружжя, що перебували на борту разом із пасажиром. Цей параметр характеризує сімейні зв'язки пасажирів на кораблі.

Parch – кількість батьків або дітей пасажирів, які перебували на борту. Разом із полем SibSp ця змінна дозволяє визначати сімейні групи пасажирів.

Ticket – номер квитка пасажирів. Поле має текстовий формат і зазвичай використовується як ідентифікатор, хоча іноді може містити інформацію про тип квитка або компанію-перевізника.

Fare – вартість квитка, сплачена пасажиром. Це числова ознака, яка опосередковано відображає соціально-економічний статус пасажирів та рівень комфорту подорожі.

Cabin – номер каюти, в якій перебував пасажир. У багатьох записах це поле може бути порожнім, оскільки інформація про каюту була зафіксована не для всіх пасажирів.

Embarked – порт посадки пасажирів на корабель. Значення представлені літерами C, Q та S, що відповідають портам Cherbourg, Queenstown та Southampton відповідно. Ця змінна може впливати на склад пасажирів і їх соціально-економічні характеристики.

Для більш детального аналізу другого обраного набору даних доцільно розглянути структуру його ознак. Датасет Credit Card Fraud Detection містить інформацію про фінансові транзакції банківських карток і використовується для задачі виявлення шахрайських операцій. Кожен запис у наборі даних відповідає окремій транзакції та містить набір числових характеристик, що описують її властивості.

Структуру основних ознак датасету наведено на рисунку 2.3. У таблиці представлено перелік стовпців, які використовуються як вхідні параметри для алгоритмів машинного навчання. Далі розглянемо зміст кожної ознаки, що входить до складу цього набору даних.

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	-1.36	-0.07	2.54	1.38	-0.34	0.46	0.24	0.1	0.36	0.09	-0.55	-0.62	-0.99	-0.31	1.47	-0.47	0.21	0.03	0.4	0.25	-0.02	0.28	-0.11	0.07	0.13	-0.19	0.13	-0.02	149.62	0
0	1.19	0.27	0.17	0.45	0.06	-0.08	-0.08	0.09	-0.26	-0.17	1.61	1.07	0.49	-0.14	0.64	0.46	-0.11	-0.18	-0.15	-0.07	-0.23	-0.64	0.1	-0.34	0.17	0.13	-0.01	0.01	2.69	0
1	-1.36	-1.34	1.77	0.38	-0.5	1.8	0.79	0.25	-1.51	0.21	0.62	0.07	0.72	-0.17	2.35	-2.89	1.11	-0.12	-2.26	0.52	0.25	0.77	0.91	-0.69	-0.33	-0.14	-0.06	-0.06	378.66	0
1	-0.97	-0.19	1.79	-0.86	-0.01	1.25	0.24	0.38	-1.39	-0.05	-0.23	0.18	0.51	-0.29	-0.63	-1.06	-0.68	1.97	-1.23	-0.21	-0.11	0.01	-0.19	-1.18	0.65	-0.22	0.06	0.06	123.5	0
2	-1.16	0.88	1.55	0.4	-0.41	0.1	0.59	-0.27	0.82	0.75	-0.82	0.54	1.35	-1.12	0.18	-0.45	-0.24	-0.04	0.8	0.41	-0.01	0.8	-0.14	0.14	-0.21	0.5	0.22	0.22	69.99	0
2	-0.43	0.96	1.14	-0.17	0.42	-0.03	0.48	0.26	-0.57	-0.37	1.34	0.36	-0.36	-0.14	0.52	0.4	-0.06	0.07	-0.03	0.08	-0.21	-0.56	-0.03	-0.37	-0.23	0.11	0.25	0.08	3.67	0
4	1.23	0.14	0.05	1.2	0.19	0.27	-0.01	0.08	0.46	-0.1	-1.42	-0.15	-0.75	0.17	0.05	-0.44	0	-0.61	-0.05	-0.22	-0.17	-0.27	-0.15	-0.78	0.75	-0.26	0.03	0.01	4.99	0
7	-0.64	1.42	1.07	-0.49	0.95	0.43	1.12	-3.81	0.62	1.25	-0.62	0.29	1.76	-1.32	0.69	-0.08	-1.22	-0.36	0.32	-0.16	1.94	-1.02	0.06	-0.65	-0.42	-0.05	-1.21	-1.09	40.8	0
7	-0.89	0.29	-0.11	-0.27	2.67	3.72	0.37	0.85	-0.39	-0.41	-0.71	-0.11	-0.29	0.07	-0.33	-0.21	-0.5	0.12	0.57	0.05	-0.07	-0.27	-0.2	1.01	0.37	-0.38	0.01	0.14	93.2	0
9	-0.24	1.13	1.04	-0.33	0.5	-0.35	0.55	0.07	-0.74	-0.37	1.03	0.84	1.03	-0.14	0.15	0.71	-0.54	0.18	0.45	0.2	-0.35	-0.63	-0.13	-0.30	-0.07	0.09	0.35	0.08	3.68	0

Рисунок 2.3 – Структура даних creditcard

Time – час здійснення транзакції, представлений у вигляді кількості секунд, що минули від моменту першої транзакції у датасеті. Ця ознака дозволяє враховувати часову послідовність операцій.

V1, V2, ... V28 – набір ознак, що представляють собою головні компоненти, отримані в результаті застосування методу головних компонент (Principal Component Analysis, PCA) до початкових даних. Таке перетворення було виконано для анонімізації конфіденційної інформації

про транзакції та зменшення розмірності простору ознак. Кожна з цих змінних є лінійною комбінацією початкових характеристик транзакції.

Amount – сума транзакції, виражена у грошових одиницях. Ця ознака не була перетворена за допомогою PCA і відображає реальний обсяг фінансової операції.

Class – цільова змінна датасету, що визначає основний тип транзакції. Значення 0 відповідає звичайній транзакції, тоді як значення 1 означає шахрайську операцію.

Для детального аналізу третього обраного набору даних доцільно розглянути структуру його ознак. Датасет Wine Quality Dataset містить фізико-хімічні характеристики зразків вина, на основі яких визначається їх якість. Кожен запис у наборі даних відповідає окремому зразку вина та описується набором кількісних параметрів, що характеризують його хімічний склад.

Структуру основних ознак даного набору, що ілюструє їх склад і взаємозв'язки, наведено на рисунку 2.4. У таблиці представлено перелік стовпців датасету, що використовуються як вхідні параметри для алгоритмів машинного навчання. Далі розглянемо зміст кожної з ознак, які входять до складу даного набору даних.

Fixed acidity – рівень фіксованої кислотності вина. Цей параметр характеризує сумарну концентрацію нелетких кислот, що впливають на смакові властивості напою.

Volatile acidity – рівень леткої кислотності. Надмірне значення цієї характеристики може призводити до появи небажаного оцтового присмаку.

Citric acid – вміст лимонної кислоти у вині. Ця речовина сприяє формуванню свіжості смаку та може впливати на загальний баланс кислотності.

Residual sugar – залишковий цукор у вині, який визначає кількість цукру, що залишився після завершення процесу ферментації.

Fixed acidity	Volatile acidity	Citric acid	Residual sugar	Chlorides	Free sulfur dioxide	Total sulfur dio	Density	PH	Sulphates	Alcohol	Quality	Id
7.4	0.7	0	1.9	0.08	11	34	1	3.51	0.56	9.4	5	0
7.8	0.88	0	2.6	0.1	25	67	1	3.2	0.68	9.8	5	1
7.8	0.76	0.04	2.3	0.09	15	54	1	3.26	0.65	9.8	5	2
11.2	0.28	0.56	1.9	0.08	17	60	1	3.16	0.58	9.8	6	3
7.4	0.7	0	1.9	0.08	11	34	1	3.51	0.56	9.4	5	4
7.4	0.66	0	1.8	0.08	13	40	1	3.51	0.56	9.4	5	5
7.9	0.6	0.06	1.6	0.07	15	59	1	3.3	0.46	9.4	5	6
7.3	0.65	0	1.2	0.07	15	21	0.99	3.39	0.47	10	7	7
7.8	0.58	0.02	2	0.07	9	18	1	3.36	0.57	9.5	7	8
6.7	0.58	0.08	1.8	0.1	15	65	1	3.28	0.54	9.2	5	10
5.6	0.62	0	1.6	0.09	16	59	0.99	3.58	0.52	9.9	5	12

Рисунок 2.4 – Структура даних WineQT

Chlorides – концентрація хлоридів у вині. Цей показник характеризує вміст солей та може впливати на смакові властивості напою.

Free sulfur dioxide – кількість вільного діоксиду сірки, що використовується як консервант для запобігання окисненню та розвитку небажаних мікроорганізмів.

Total sulfur dioxide – загальна кількість діоксиду сірки у вині, включаючи як вільну, так і зв'язану форми цієї речовини.

Density – густина вина, яка залежить від вмісту цукру та алкоголю і може використовуватися як непрямая характеристика складу напою.

pH – показник кислотності вина, що характеризує рівень кислотно-лужного балансу.

Sulphates – концентрація сульфатів, які можуть впливати на стабільність вина та формування його смаку.

Alcohol – вміст алкоголю у вині, що є одним із ключових параметрів, які визначають його органолептичні властивості.

Quality – оцінка якості вина, визначена експертами. Це цільова змінна датасету, яка використовується у задачах машинного навчання для прогнозування якості зразків.

Id – унікальний ідентифікатор запису у наборі даних, що використовується для нумерації зразків і не несе змістового навантаження для навчання моделей.

У результаті аналізу було обрано три набори навчальних даних, що представляють різні предметні галузі та мають різну структуру ознак. Датасет Titanic використовується для задачі бінарної класифікації на основі соціально-демографічних характеристик пасажирів, датасет Credit Card Fraud Detection містить фінансові транзакції та характеризується значним дисбалансом класів, а Wine Quality Dataset описує фізико-хімічні властивості вина та дозволяє досліджувати залежність якості продукції від кількісних параметрів.

Використання цих трьох датасетів дозволяє проводити експериментальне дослідження алгоритмів машинного навчання в різних умовах структури даних, обсягу вибірок і розподілу ознак. Наявність різнотипних ознак і можливість модифікації даних забезпечують умови для аналізу стабільності та чутливості моделей. Це забезпечує більш об'єктивну оцінку поведінки алгоритмів машинного навчання під час змін у навчальних даних.

2.4 Формування сценаріїв збурення навчальних даних

У процесі застосування алгоритмів машинного навчання важливим фактором, що впливає на якість побудованих моделей, є характеристики навчальних даних. У реальних умовах дані рідко є ідеально структурованими та стабільними. Вони можуть містити різноманітні похибки, неточності, випадкові коливання або систематичні зміни. Такі відхилення від початкових характеристик даних прийнято розглядати як збурення даних.

Збуренням даних називають будь-які зміни у структурі, значеннях або розподілі навчальної вибірки, які можуть виникати як у процесі збору інформації, так і під час її обробки. До таких змін можуть належати додавання випадкового шуму, поява викидів, помилки у мітках класів, зміна обсягу вибірки, порушення балансу класів або зміщення розподілу даних.

Кожен із цих факторів може впливати на поведінку алгоритмів машинного навчання та на результати їх навчання.

У реальних прикладних задачах подібні збурення виникають досить часто. Наприклад, сенсори або вимірювальні пристрої можуть генерувати шум у даних, людський фактор може призводити до помилок під час маркування об'єктів, а зміни умов середовища можуть спричиняти зміну розподілу даних у часі. У результаті моделі машинного навчання змушені працювати з даними, які відрізняються від тих, на яких вони були спочатку навчені.

У наукових дослідженнях збурення даних часто створюються штучно. Такий підхід дозволяє контролювати тип і ступінь змін у навчальній вибірці та систематично аналізувати їх вплив на результати навчання моделей. Завдяки цьому можна порівнювати різні алгоритми машинного навчання за їх здатністю зберігати стабільні результати при зміні характеристик даних.

Штучне внесення збурень також дозволяє моделювати типові проблеми, що виникають під час роботи з реальними даними. Наприклад, додавання шуму імітує похибки вимірювань, створення викидів відображає наявність аномальних значень, а зміна балансу класів дозволяє аналізувати поведінку моделей у ситуаціях нерівномірного розподілу об'єктів різних категорій.

Загальний підхід до формування сценаріїв полягає у використанні початкового набору даних як базової вибірки, на основі якої створюються модифіковані версії датасету. Базова вибірка використовується для навчання моделей у стандартних умовах, після чого до неї послідовно застосовуються різні типи збурень. Це дозволяє порівнювати результати навчання алгоритмів до і після внесення змін у дані.

Кожен сценарій збурення передбачає застосування певного типу модифікації до навчальної вибірки. Залежно від характеру дослідження можуть використовуватися різні типи змін, зокрема додавання шуму до значень ознак, створення аномальних значень, внесення помилок у цільові

мітки або зміна структури вибірки. Кожен із таких сценаріїв дозволяє моделювати конкретний тип проблем, що може виникати під час роботи з реальними даними.

Для забезпечення систематичного аналізу впливу збурень на результати навчання доцільно використовувати декілька рівнів інтенсивності змін. Наприклад, частка модифікованих записів у вибірці може поступово збільшуватися від невеликих значень до більш суттєвих. Це дозволяє простежити, як змінюється поведінка алгоритмів машинного навчання зі зростанням ступеня порушення структури даних.

У межах кожного сценарію модифіковані набори даних використовуються для повторного навчання моделей машинного навчання. Після цього отримані результати порівнюються з результатами, отриманими на базовій вибірці. Такий підхід дозволяє оцінити, наскільки сильно змінюються показники якості моделей під впливом внесених збурень.

Для підвищення достовірності результатів експериментів важливо забезпечити однакові умови навчання моделей для всіх сценаріїв. Це означає, що алгоритми машинного навчання навчаються з використанням однакових параметрів і однакових процедур попередньої обробки даних. Завдяки цьому відмінності в результатах можуть бути пояснені саме впливом збурень, а не іншими факторами.

Також важливо забезпечити відтворюваність експериментів. Для цього всі операції внесення змін у дані виконуються за допомогою формалізованих алгоритмів, які можуть бути повторені при повторному проведенні дослідження. Це дозволяє отримувати однакові результати за тих самих умов експерименту.

У результаті застосування описаного підходу формується набір експериментальних сценаріїв, що включає базовий датасет та декілька модифікованих версій навчальних вибірок. Кожна з таких вибірок використовується для навчання моделей і подальшого аналізу отриманих результатів. Це дозволяє здійснювати порівняння показників якості моделей

за різних умов та оцінювати їхню стійкість до змін у структурі даних. Отримані результати формують основу для подальшого аналізу впливу окремих типів збурень на ефективність алгоритмів машинного навчання.

У межах експериментального дослідження передбачається внесення декількох типів збурень у навчальні дані. Вибір конкретних змін визначається структурою обраних датасетів, їх обсягом, а також характером ознак, що входять до складу наборів даних. Такий підхід дозволяє застосовувати лише ті типи модифікацій, які є коректними для конкретного датасету та дозволяють отримати інформативні результати експерименту. Крім того, використання різних варіантів модифікації даних дає можливість моделювати умови, наближені до реальних задач, у яких дані можуть містити помилки, пропуски або інші спотворення.

Зокрема, для невеликих датасетів, таких як Titanic та Wine Quality, недоцільно застосовувати експерименти зі зміною обсягу навчальної вибірки, оскільки це може призвести до надмірного зменшення кількості спостережень і погіршення якості навчання моделей. У таких випадках скорочення вибірки може негативно вплинути на достовірність отриманих результатів. Натомість великий датасет Credit Card Fraud Detection дозволяє проводити більш широкий спектр експериментів, включаючи зміну розміру вибірки та моделювання різних структурних змін у даних. Це забезпечує можливість більш детального аналізу впливу збурень на якість класифікації.

Перелік типів збурень навчальних даних та їх застосування до обраних датасетів наведено у таблиці 2.3.

Таблиця 2.3 – Типи збурень навчальних даних та їх застосування до обраних датасетів

Тип збурення	Суть збурення	Titanic	Credit Card Fraud	Wine Quality
Шум у вхідних ознаках	Додавання випадкових відхилень до значень числових ознак	+	+	+

Продовження таблиці 2.3

Тип збурення	Суть збурення	Titanic	Credit Card Fraud	Wine Quality
Викиди	Створення штучних аномальних значень ознак	–	+	+
Помилки у мітках	Випадкова зміна значень цільової змінної	+	+	+
Зміна обсягу вибірки	Формування навчальних підвибірок різного розміру	–	+	–
Дисбаланс класів	Штучна зміна співвідношення між класами	+	+	–
Зміщення розподілу даних	Модифікація розподілу значень окремих ознак	–	+	+

Для проведення експериментального дослідження важливо не лише визначити типи збурень навчальних даних, але й встановити рівні їх інтенсивності. Інтенсивність збурення характеризує ступінь змін, які вносяться до початкового набору даних. Використання різних рівнів інтенсивності дозволяє дослідити, як поступове порушення структури даних впливає на результати навчання алгоритмів машинного навчання.

У даному дослідженні збурення навчальних даних будуть застосовуватися поступово, із використанням декількох рівнів модифікації. Такий підхід дозволяє аналізувати поведінку моделей у різних умовах – від незначних змін у даних до більш суттєвих порушень їх структури. Завдяки цьому можна простежити тенденції зміни якості моделей залежно від ступеня збурення навчальної вибірки.

Для більшості типів збурень доцільно використовувати відносні показники, які визначають частку модифікованих даних у вибірці.

У межах експерименту планується використовувати декілька рівнів інтенсивності збурень, зокрема невеликий, середній та підвищений рівні змін. На початковому рівні модифікується лише невелика частка даних, що дозволяє оцінити поведінку алгоритмів у умовах незначних порушень навчальної вибірки. Наступні рівні передбачають поступове збільшення

частки модифікованих значень, що дозволяє аналізувати зміну результатів навчання зі зростанням ступеня збурення.

Застосування різних рівнів інтенсивності збурень також дозволяє оцінити чутливість алгоритмів до змін у навчальних даних. Якщо навіть незначні зміни призводять до суттєвого погіршення результатів, це свідчить про високу чутливість моделі. Натомість алгоритми, результати яких залишаються відносно стабільними навіть при значних збуреннях даних, можна вважати більш стійкими до змін у навчальній вибірці.

У процесі підготовки експериментального дослідження важливо не лише визначити типи збурень навчальних даних, але й систематизувати порядок їх застосування до обраних наборів даних. Це дозволяє чітко структурувати експериментальну процедуру та забезпечити послідовність формування модифікованих навчальних вибірок. Візуалізація такого процесу спрощує розуміння логіки формування експериментальних сценаріїв та демонструє взаємозв'язок між базовими датасетами і відповідними типами збурень.

Схему використання датасетів у межах експерименту та відповідні сценарії їх модифікації наведено на рисунку 2.5. На діаграмі представлено структуру експериментальних даних та перелік збурень, які застосовуються до кожного з обраних наборів даних.

На рисунку показано загальну схему формування модифікованих наборів даних, що використовуються у подальших експериментах. У верхній частині діаграми розміщено блок базових датасетів, які слугують початковими наборами даних для проведення дослідження. До них належать три датасети: Titanic Dataset, Credit Card Fraud Dataset та Wine Quality Dataset. Кожен із цих наборів даних використовується як основа для формування окремих сценаріїв збурення.

Для датасету Titanic передбачено три основні типи модифікацій навчальних даних. До них належать додавання шуму до числових ознак, внесення помилок у мітки класів, а також створення штучного дисбалансу

класів. Такі збурення дозволяють досліджувати поведінку алгоритмів машинного навчання у випадках порушення структури даних та зміни співвідношення між класами.

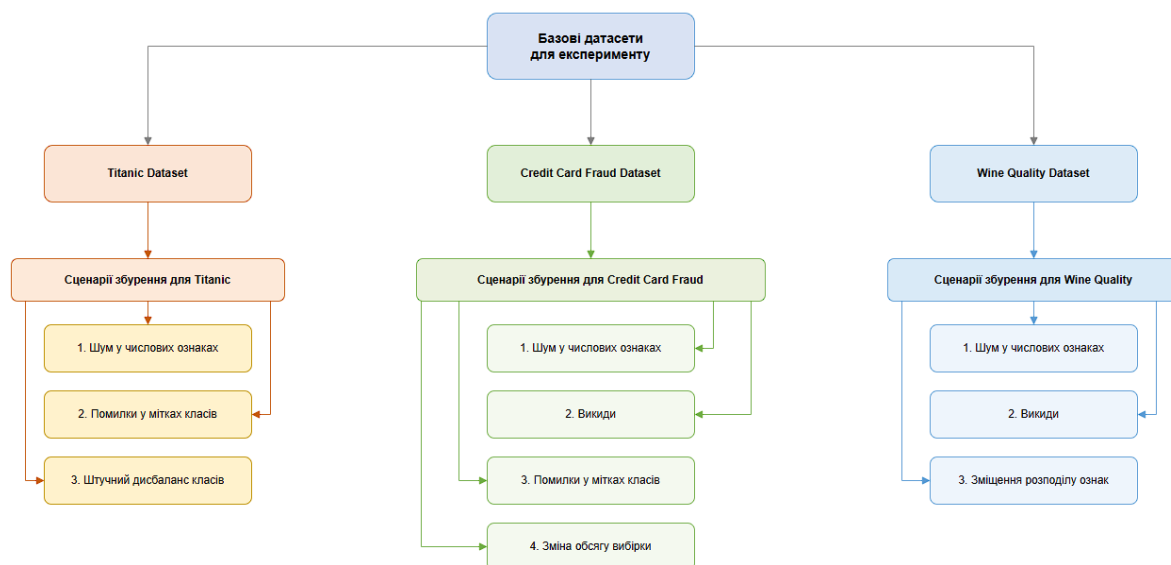


Рисунок 2.5 – Схема використання датасетів у межах експерименту та відповідні сценарії їх модифікації

Для датасету Credit Card Fraud Detection передбачено більш широкий перелік сценаріїв збурення, що пов'язано з великим обсягом цього набору даних. До таких сценаріїв належать додавання шуму до числових ознак, створення аномальних значень (викидів), внесення помилок у мітки класів, а також зміна обсягу навчальної вибірки. Це дозволяє досліджувати поведінку алгоритмів у різних умовах зміни структури фінансових транзакцій.

Для датасету Wine Quality Dataset передбачено три типи збурень: додавання шуму до числових ознак, створення викидів та зміщення розподілу ознак. Такі модифікації дозволяють моделювати зміну фізико-хімічних характеристик вина та оцінювати, як алгоритми машинного навчання реагують на зміну статистичних властивостей даних. Загалом наведена схема демонструє логіку формування експериментальних наборів

даних та визначає набір сценаріїв, що використовуються для дослідження стабільності та чутливості алгоритмів машинного навчання.

2.5 Основні метрики оцінювання стабільності і чутливості алгоритмів.

Під час дослідження алгоритмів машинного навчання важливо мати можливість кількісно оцінювати результати їх роботи. Для цього використовуються спеціальні метрики оцінювання, які дозволяють вимірювати якість побудованих моделей та порівнювати різні алгоритми між собою. Метрики надають формалізований спосіб оцінки результатів навчання, що робить можливим проведення об'єктивного аналізу ефективності алгоритмів.

У більшості практичних задач машинного навчання однією з основних метрик оцінювання є точність класифікації (accuracy). Вона показує частку правильно класифікованих об'єктів відносно загальної кількості прикладів у вибірці. Ця метрика є простою для інтерпретації та широко використовується для оцінювання якості моделей.

Однак використання лише точності не дозволяє повною мірою оцінити властивості алгоритмів машинного навчання в умовах змін навчальних даних. Значення accuracy відображає лише кінцеву якість моделі на конкретному наборі даних, але не показує, наскільки стабільними є результати навчання при повторних запусках або при зміні структури вибірки. У результаті алгоритм може демонструвати високу точність у базових умовах, але водночас бути дуже чутливим до незначних змін у даних.

Особливо це проявляється у випадках, коли в навчальні дані вносяться збурення, такі як шум, викиди або помилки у мітках класів. У таких ситуаціях точність може змінюватися нерівномірно для різних алгоритмів, і лише аналіз її значення не дозволяє визначити характер цієї зміни. Для цього необхідно використовувати додаткові метрики, які дозволяють оцінювати

варіативність результатів та ступінь впливу змін у даних на поведінку моделей.

Під час проведення експериментального дослідження стабільності та чутливості алгоритмів машинного навчання важливо правильно обрати метрики оцінювання результатів. Обрані метрики повинні відповідати певним вимогам, які забезпечують коректність порівняння різних алгоритмів та можливість об'єктивної інтерпретації отриманих результатів. До основних таких вимог належать порівнюваність результатів між алгоритмами, чутливість метрик до змін у даних та інтерпретованість отриманих значень.

Однією з ключових вимог є порівнюваність результатів між алгоритмами. Метрики повинні бути визначені таким чином, щоб їх значення можна було безпосередньо використовувати для порівняння різних моделей машинного навчання. Це означає, що одна й та сама метрика повинна застосовуватися до всіх алгоритмів за однакових умов експерименту. Завдяки цьому стає можливим об'єктивно оцінювати, який алгоритм демонструє кращу стабільність або меншу чутливість до змін у навчальних даних.

Ще однією важливою вимогою є чутливість метрик до змін у даних. Метрики повинні реагувати на навіть відносно невеликі зміни у структурі навчальної вибірки, наприклад додавання шуму або появу викидів. Якщо метрика є недостатньо чутливою до таких змін, вона не дозволить виявити відмінності у поведінці різних алгоритмів при збуреннях даних. Тому під час вибору метрик важливо забезпечити, щоб вони відображали навіть незначні зміни у результатах навчання моделей.

Також важливим аспектом є стабільність самих метрик оцінювання. Метрика повинна давати узгоджені результати при повторних експериментах за однакових умов. Це дозволяє уникнути ситуацій, коли зміна значення метрики викликана випадковими факторами, а не реальним

впливом збурень даних. Стабільність метрик є важливою умовою для отримання достовірних результатів дослідження.

Ще однією суттєвою вимогою є інтерпретованість результатів. Значення метрик повинні бути зрозумілими для дослідника та дозволяти однозначно оцінювати якість моделі. Наприклад, метрики повинні мати зрозумілу шкалу значень або чітко визначений напрямок інтерпретації, коли більші значення відповідають кращим результатам або навпаки.

Крім того, інтерпретованість результатів дозволяє використовувати метрики не лише для формального порівняння алгоритмів, але й для аналізу характеру їх поведінки в умовах змін навчальних даних. Це дає можливість робити обґрунтовані висновки щодо рівня стабільності моделей та їх чутливості до різних типів збурень.

Однією з основних метрик оцінювання якості моделей машинного навчання, що використовується у даному дослідженні, є точність класифікації (ассигасу). Ця метрика широко застосовується у задачах класифікації та дозволяє оцінити загальну частку правильно класифікованих об'єктів у тестовій вибірці. Завдяки простоті обчислення та зрозумілості інтерпретації ассигасу часто використовується як базовий показник якості моделі.

Метрика ассигасу визначається як відношення кількості правильно передбачених класів до загальної кількості об'єктів у вибірці. Іншими словами, вона показує, яка частка об'єктів була класифікована моделлю правильно. Значення метрики знаходиться у межах від 0 до 1, де більше значення відповідає кращій якості класифікації.

У межах даного дослідження ассигасу використовується як базовий орієнтир для оцінювання результатів експериментів. Перед проведенням експериментів із збуреннями навчальних даних моделі машинного навчання навчаються на початкових наборах даних без модифікацій. Отримане значення ассигасу у таких умовах слугує базовим рівнем якості моделі, з яким у подальшому порівнюються результати експериментів.

Після внесення збурень у навчальні дані моделі повторно навчаються на модифікованих вибірках, а отримані значення асигасу порівнюються з базовими результатами. Такий підхід дозволяє оцінити, наскільки змінюється якість класифікації під впливом різних типів збурень. Якщо значення асигасу знижується незначно, це може свідчити про високу стійкість алгоритму до змін у даних.

Крім того, використання асигасу дозволяє забезпечити єдиний показник якості для всіх алгоритмів машинного навчання, що досліджуються. Оскільки всі моделі оцінюються за однаковою метрикою, це створює можливість прямого порівняння їх результатів у різних експериментальних умовах.

Для оцінювання стабільності результатів алгоритмів машинного навчання у даному дослідженні використовується метрика стандартного відхилення (Standard Deviation). Ця метрика дозволяє оцінити варіативність результатів експериментів і визначити, наскільки сильно змінюються значення показників якості моделей при повторних запусках або при використанні модифікованих навчальних вибірок.

Стандартне відхилення характеризує ступінь розсіювання значень певної метрики відносно їх середнього значення. У межах даного дослідження воно застосовується до значень базової метрики якості моделі, отриманих у різних експериментальних сценаріях. Чим меншим є значення стандартного відхилення, тим більш стабільними є результати роботи алгоритму.

Застосування цієї метрики дозволяє оцінити, наскільки результати алгоритмів змінюються при внесенні різних типів збурень у навчальні дані. Якщо при зміні структури даних значення метрики якості коливаються незначно, це свідчить про високу стабільність алгоритму. У протилежному випадку, коли результати експериментів значно відрізняються один від одного, можна говорити про низьку стабільність моделі.

У межах експериментального дослідження для кожного алгоритму машинного навчання буде отримано набір значень метрики accuracy, що відповідають різним сценаріям збурення даних. На основі цих значень обчислюватиметься стандартне відхилення, яке відобразатиме ступінь варіативності результатів між експериментами.

Завдяки використанню стандартного відхилення стає можливим кількісно оцінити стабільність алгоритмів машинного навчання. Порівнюючи значення цієї метрики для різних моделей, можна визначити, які алгоритми демонструють більш стабільну поведінку в умовах змін навчальних даних.

Для оцінювання чутливості алгоритмів машинного навчання до змін у навчальних даних у даному дослідженні використовується метрика Sensitivity Index. Ця метрика дозволяє визначити, наскільки сильно змінюється якість роботи моделі під впливом збурень у даних. Вона відображає ступінь реакції алгоритму на зміну характеристик навчальної вибірки.

Основна ідея метрики Sensitivity Index полягає у порівнянні значення базової метрики якості моделі, отриманої на початковому наборі даних, із відповідним значенням метрики після внесення збурень. Таким чином визначається відносна зміна якості моделі, що дозволяє оцінити вплив модифікації даних на результати навчання алгоритму.

У межах даного дослідження Sensitivity Index обчислюється на основі значень метрики accuracy, отриманих для базового та модифікованого наборів даних. Чим більшою є різниця між цими значеннями, тим більш чутливим до змін у навчальних даних є відповідний алгоритм машинного навчання. Невелике значення цієї метрики, навпаки, свідчить про відносну стійкість моделі до збурень.

Використання відносної зміни метрики якості дозволяє проводити порівняння різних алгоритмів у єдиній шкалі оцінювання. Це важливо, оскільки різні моделі можуть мати різні базові значення accuracy, і пряме

порівняння абсолютних змін не завжди є коректним. Використання відносного показника дозволяє нормалізувати результати та зробити їх більш придатними для аналізу.

Крім того, Sensitivity Index дозволяє оцінити вплив різних типів збурень на поведінку моделей машинного навчання. Наприклад, одна модель може демонструвати високу стійкість до шуму у даних, але бути більш чутливою до помилок у мітках класів. Аналіз значень цієї метрики для різних сценаріїв збурення дозволяє виявити подібні особливості поведінки алгоритмів.

Для оцінювання робастності алгоритмів машинного навчання у даному дослідженні використовується метрика Robustness Ratio. Ця метрика дозволяє оцінити, наскільки добре модель зберігає свою якість після внесення збурень у навчальні дані. На відміну від метрик, що вимірюють абсолютні зміни результатів, Robustness Ratio безпосередньо характеризує здатність моделі підтримувати стабільний рівень точності в умовах порушення структури даних.

Основна ідея метрики полягає у порівнянні значення базової метрики якості моделі, отриманої на початковому наборі даних, із відповідним значенням цієї метрики після внесення збурень. У межах даного дослідження як базова метрика використовується accuracy. Robustness Ratio визначається як відношення значення accuracy, отриманого на модифікованих даних, до значення accuracy, отриманого на початковому наборі даних.

Якщо значення Robustness Ratio наближається до одиниці, це означає, що якість моделі майже не змінюється після внесення збурень у дані. У такому випадку алгоритм можна вважати робастним до змін у навчальній вибірці. Натомість значне зменшення значення цієї метрики свідчить про те, що модель суттєво втрачає якість при зміні структури даних.

Використання Robustness Ratio дозволяє оцінювати поведінку алгоритмів машинного навчання у відносних величинах, що полегшує

порівняння результатів різних моделей. Завдяки цьому можна визначити, які алгоритми краще зберігають свої властивості при появі шуму, викидів або інших типів збурень у навчальних даних.

Крім того, ця метрика дозволяє аналізувати робастність моделей у різних експериментальних сценаріях. Наприклад, можна оцінювати, як змінюється Robustness Ratio при різних рівнях інтенсивності збурень або при використанні різних типів модифікацій даних. Це створює можливість більш глибокого аналізу поведінки алгоритмів у складних умовах навчання.

Для проведення експериментального дослідження стабільності та чутливості алгоритмів машинного навчання було сформовано набір метрик оцінювання, які дозволяють комплексно аналізувати поведінку моделей у різних умовах зміни навчальних даних. Кожна з обраних метрик відображає окремий аспект функціонування алгоритмів: базову якість класифікації, стабільність результатів між експериментами, чутливість до змін у даних та робастність моделей. Використання сукупності метрик дає змогу об'єктивно оцінити ефективність алгоритмів і визначити їхню стійкість до різних типів збурень у навчальних даних.

Зведену характеристику обраних метрик наведено у таблиці 2.4.

Таблиця 2.4 – Основні метрики оцінювання стабільності та чутливості алгоритмів машинного навчання

Метрика	Формула	Що вимірює	Характеристика алгоритму
Accuracy	$Ac = \frac{TP + TN}{TP + TN + FP + FN}$	Частку правильно класифікованих об'єктів	Базова якість моделі
Standard Deviation	$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$	Варіативність результатів між експериментами	Стабільність
Sensitivity Index	$SI = \frac{ Q_{perturbed} - Q_{base} }{Q_{base}}$	Відносну зміну якості моделі після внесення збурення	Чутливість
Robustness Ratio	$RR = \frac{Q_{perturbed}}{Q_{base}}$	Наскільки добре модель зберігає якість після збурень	Робастність

Як видно з таблиці 2.4, обрані метрики дозволяють оцінювати різні характеристики поведінки алгоритмів машинного навчання. Метрика *assurasy* використовується як базовий показник якості моделі та виступає відправною точкою для подальшого аналізу результатів експериментів. На основі її значень розраховуються інші показники, що дозволяють оцінювати реакцію моделей на зміни у навчальних даних.

Метрика стандартного відхилення використовується для оцінювання стабільності результатів алгоритмів між різними експериментальними сценаріями. *Sensitivity Index* характеризує ступінь зміни якості моделі після внесення збурень у дані, а метрика *Robustness Ratio* дозволяє оцінити, наскільки добре модель зберігає свою ефективність у змінених умовах. Сукупне використання цих метрик забезпечує комплексну оцінку поведінки алгоритмів машинного навчання у межах проведеного дослідження.

Таким чином, було визначено основні метрики, що використовуватимуться для оцінювання стабільності та чутливості алгоритмів машинного навчання в межах експериментального дослідження. Розглянуто базову метрику якості моделі *assurasy*, а також метрики, що дозволяють оцінювати варіативність результатів, чутливість до збурень та робастність моделей. Сукупне використання метрик *Standard Deviation*, *Sensitivity Index* та *Robustness Ratio* забезпечує можливість комплексного аналізу поведінки алгоритмів машинного навчання в умовах змін навчальних даних.

2.6 План та протокол проведення обчислювального експерименту

У межах даного дослідження для аналізу стабільності та чутливості алгоритмів машинного навчання буде проведено серію обчислювальних експериментів. Метою експериментів є оцінювання того, як змінюється якість моделей при внесенні різних типів збурень у навчальні дані. Для

цього буде сформовано набір експериментальних сценаріїв, у межах яких алгоритми навчатимуться на базових та модифікованих наборах даних.

На першому етапі експерименту кожен із обраних алгоритмів машинного навчання буде навчено на початкових версіях датасетів без внесення будь-яких змін. Для кожного набору даних буде сформовано навчальну та тестову вибірки. Після навчання моделей буде обчислено базове значення метрики ассурасу, яке використовуватиметься як орієнтир для подальшого аналізу результатів.

На наступному етапі до навчальних даних будуть внесені визначені раніше типи збурень. До таких збурень належать додавання шуму до числових ознак, створення викидів, внесення помилок у мітки класів, зміна обсягу навчальної вибірки, а також формування дисбалансу класів або зміщення розподілу ознак. Для кожного типу збурення будуть сформовані модифіковані версії датасетів.

Після формування модифікованих наборів даних алгоритми машинного навчання повторно навчатимуться на цих вибірках. Для кожного експериментального сценарію буде обчислено значення метрики ассурасу, що дозволить оцінити зміну якості моделей порівняно з базовими результатами.

На основі значень метрики ассурасу буде розраховано додаткові показники оцінювання: стандартне відхилення для аналізу стабільності алгоритмів, Sensitivity Index для визначення їх чутливості до змін у даних та Robustness Ratio для оцінювання стійкості моделей до збурень.

Усі результати експериментів будуть систематизовані у вигляді таблиць та графічних залежностей, що дозволить порівняти поведінку різних алгоритмів машинного навчання у різних умовах зміни навчальних даних. Аналіз отриманих результатів дозволить визначити, які алгоритми демонструють вищу стабільність та меншу чутливість до збурень.

Блок-схему алгоритму проведення обчислювального експерименту наведено на рисунку 2.6. Вона відображає основні етапи дослідження — від підготовки даних до аналізу результатів і формування висновків.

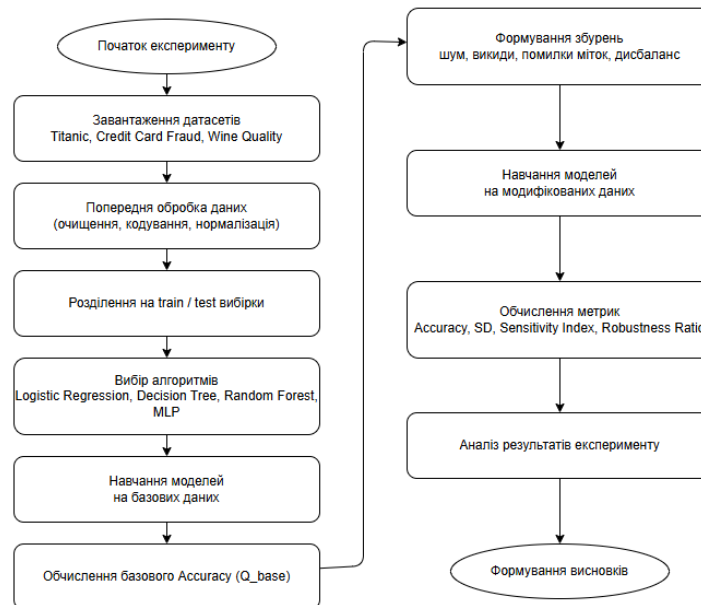


Рисунок 2.6 – Алгоритм проведення експерименту

Сформований план проведення обчислювального експерименту визначає послідовність основних етапів дослідження стабільності та чутливості алгоритмів машинного навчання. У межах цього плану передбачено підготовку даних, навчання моделей на базових та модифікованих вибірках, обчислення метрик оцінювання та подальший аналіз отриманих результатів. Така структура експерименту забезпечує систематичне дослідження впливу збурень навчальних даних на поведінку алгоритмів машинного навчання.

3 ФОРМУВАННЯ НАВЧАЛЬНИХ ДАНИХ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ

3.1 Попередня обробка та підготовка даних

Для проведення експериментального дослідження стабільності та чутливості алгоритмів машинного навчання було обрано три набори даних, що відрізняються за структурою, предметною областю та статистичними характеристиками. Використання декількох різних датасетів дозволяє оцінити поведінку алгоритмів у різних умовах та забезпечує більш об'єктивний аналіз їх стабільності при зміні навчальних даних.

Використання трьох обраних наборів даних дозволяє дослідити поведінку алгоритмів машинного навчання на даних різної структури, з різними типами ознак та різними статистичними характеристиками. Це створює можливість більш комплексного аналізу стабільності та чутливості алгоритмів у різних умовах навчання.

Наступним етапом підготовки даних є аналіз їх структури. На цьому етапі було проведено дослідження основних характеристик кожного з обраних наборів даних. Зокрема, визначено кількість об'єктів у вибірці, кількість ознак, а також типи змінних, що входять до складу датасетів. Такий аналіз дозволяє отримати загальне уявлення про структуру даних і визначити особливості їх подальшої обробки.

Для виконання цього етапу було розроблено програмний код, який автоматично визначає основні структурні характеристики кожного набору даних. Код здійснює завантаження датасетів, після чого визначає кількість рядків (об'єктів), кількість стовпців (ознак), а також типи змінних у кожному з них. Це дозволяє швидко оцінити складність датасету та характер представлених у ньому даних. Крім того, отримані характеристики використовуються для подальшого аналізу особливостей наборів даних і

забезпечують коректну підготовку даних до проведення експериментального дослідження.

Код для проведення аналізу структури даних наведено у лістингу 3.1.

Лістинг 3.1 – Код проведення аналізу структури даних

```
datasets = {name: pd.read_csv(path) for name, path in
file_paths.items()}

for name, df in datasets.items():
    print(«=« * 100)
    print(f»{name}»)
    print(f»Розмірність: {df.shape[0]} рядків ×
{df.shape[1]} стовпців»)
    print(«\nЗагальні відомості:»)
    df.info()
    print(«\n»)
    structure_rows = []

    for name, df in datasets.items():
        numeric_cols =
df.select_dtypes(include=[np.number]).columns.tolist()
        categorical_cols = df.select_dtypes(include=[«object»,
«category», «bool»]).columns.tolist()
        structure_rows.append({
            «dataset»: name,
            «rows»: df.shape[0],
            «columns»: df.shape[1],
            «numeric_features»: len(numeric_cols),
            «categorical_features»: len(categorical_cols),
            «columns_list»: «, ».join(df.columns.astype(str))
        })

    structure_df = pd.DataFrame(structure_rows)
    structure_df
```

Результати виконання цього коду, що відображають основні характеристики кожного набору даних, наведено на рисунку 3.1. Отримана інформація використовується для подальших етапів підготовки даних, зокрема для визначення методів кодування змінних та масштабування числових ознак.

Titanic				CreditCard				WineQT			
Розмірність: 418 рядків x 12 стовпців				Розмірність: 284807 рядків x 31 стовпець				Розмірність: 1143 рядків x 13 стовпців			
Загальні відомості:				Загальні відомості:				Загальні відомості:			
<class 'pandas.core.frame.DataFrame'>				<class 'pandas.core.frame.DataFrame'>				<class 'pandas.core.frame.DataFrame'>			
RangeIndex: 418 entries, 0 to 417				RangeIndex: 284807 entries, 0 to 284806				RangeIndex: 1143 entries, 0 to 1142			
Data columns (total 12 columns):				Data columns (total 31 columns):				Data columns (total 13 columns):			
#	Column	Non-Null Count	dtype	#	Column	Non-Null Count	dtype	#	Column	Non-Null Count	dtype
0	PassengerId	418 non-null	int64	0	Time	284807 non-null	float64	0	fixed acidity	1143 non-null	float64
1	Survived	418 non-null	int64	1	V1	284807 non-null	float64	1	volatile acidity	1143 non-null	float64
2	Pclass	418 non-null	int64	2	V2	284807 non-null	float64	2	citric acid	1143 non-null	float64
3	Name	418 non-null	object	3	V3	284807 non-null	float64	3	residual sugar	1143 non-null	float64
4	Sex	418 non-null	object	4	V4	284807 non-null	float64	4	chlorides	1143 non-null	float64
5	Age	332 non-null	float64	5	V5	284807 non-null	float64	5	free sulfur dioxide	1143 non-null	float64
6	SibSp	418 non-null	int64	6	V6	284807 non-null	float64	6	total sulfur dioxide	1143 non-null	float64
7	Parch	418 non-null	int64	7	V7	284807 non-null	float64	7	density	1143 non-null	float64
8	Ticket	418 non-null	object	8	V8	284807 non-null	float64	8	pH	1143 non-null	float64
9	Fare	417 non-null	float64	9	V9	284807 non-null	float64	9	sulphates	1143 non-null	float64
10	Cabin	91 non-null	object	10	V10	284807 non-null	float64	10	alcohol	1143 non-null	float64
11	Embarked	418 non-null	object	11	V11	284807 non-null	float64	11	quality	1143 non-null	int64
dtypes: float64(2), int64(5), object(5)				12	V12	284807 non-null	float64	12	id	1143 non-null	int64
memory usage: 39.34 KB				13	V13	284807 non-null	float64	dtypes: float64(11), int64(2)			
				14	V14	284807 non-null	float64	memory usage: 116.2 KB			
				15	V15	284807 non-null	float64				
				16	V16	284807 non-null	float64				
				17	V17	284807 non-null	float64				
				18	V18	284807 non-null	float64				
				19	V19	284807 non-null	float64				
				20	V20	284807 non-null	float64				
				21	V21	284807 non-null	float64				
				22	V22	284807 non-null	float64				
				23	V23	284807 non-null	float64				
				24	V24	284807 non-null	float64				
				25	V25	284807 non-null	float64				
				26	V26	284807 non-null	float64				
				27	V27	284807 non-null	float64				
				28	V28	284807 non-null	float64				
				29	Amount	284807 non-null	float64				
				30	Class	284807 non-null	int64				
				dtypes: float64(30), int64(1)							
				memory usage: 67.4 MB							
dataset	rows	columns	numeric_features	categorical_features	columns_list						
0 Titanic	418	12	7	5	PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked						
1 CreditCard	284807	31	31	0	Time, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21, V22, V23, V24, ...						
2 WineQT	1143	13	13	0	fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, ...						

Рисунок 3.1 – Загальні відомості наборів даних

Результати аналізу структури даних показали, що обрані датасети суттєво відрізняються за розміром, кількістю ознак та типами змінних. Це створює різні умови для навчання моделей та дозволяє дослідити поведінку алгоритмів машинного навчання на даних різної складності.

Датасет Titanic містить 418 об'єктів і 12 ознак. У його структурі присутні як числові, так і категоріальні змінні. До числових належать, зокрема, PassengerId, Survived, Pclass, Age, SibSp, Parch та Fare, тоді як змінні Name, Sex, Ticket, Cabin та Embarked є текстовими. Також у цьому

наборі даних спостерігаються пропущені значення, зокрема в ознаках Age та Cabin.

Датасет Credit Card Fraud є значно більшим за обсягом і містить 284807 об'єктів та 31 ознаку. Усі змінні у цьому наборі даних є числовими. Більшість ознак представлена у вигляді компонент V1–V28, отриманих за допомогою методу головних компонент (PCA), що використовується для анонімізації фінансових даних. Також присутні ознаки Time, Amount та цільова змінна Class.

Датасет WineQT містить 1143 об'єкти та 13 ознак. Усі змінні цього набору даних є числовими і описують фізико-хімічні характеристики вина, такі як кислотність, вміст цукру, рівень алкоголю та інші параметри. Цільовою змінною є показник якості вина Quality.

Наступним етапом підготовки даних є виявлення та обробка пропущених значень. У реальних наборах даних досить часто зустрічаються ситуації, коли для окремих об'єктів відсутні значення деяких ознак. Причинами цього можуть бути помилки під час збору інформації, неповнота записів або особливості процесу формування даних. Наявність пропусків може негативно впливати на результати навчання алгоритмів машинного навчання, оскільки більшість моделей не здатні безпосередньо працювати з відсутніми значеннями.

Тому важливим етапом підготовки даних є аналіз наявності пропущених значень у кожному з датасетів. На цьому етапі визначається кількість пропусків у кожній ознаці та оцінюється їх вплив на подальший процес навчання моделей. Такий аналіз дозволяє прийняти рішення щодо доцільності видалення певних ознак або застосування методів заповнення пропущених значень.

Для виконання цього етапу було розроблено програмний код, який автоматично визначає кількість пропущених значень у кожному стовпці наборів даних. Код дозволяє швидко виявити ознаки, що містять пропуски,

та оцінити їхню частку відносно загального обсягу даних. Це дає можливість обґрунтовано обрати метод подальшої обробки таких значень.

Код для виявлення пропущених значень наведено у лістингу 3.2.

Лістинг 3.2 – Виявлення та обробка пропущених даних

```
def fill_missing_values(df: pd.DataFrame) -> pd.DataFrame:
    result = df.copy()
    for col in result.columns:
        if result[col].isna().sum() == 0:
            continue
        if pd.api.types.is_numeric_dtype(result[col]):
            result[col] = result[col].fillna(result[col].median())
        else:
            mode = result[col].mode(dropna=True)
            fill_value = mode.iloc[0] if not mode.empty else «Unknown»
            result[col] = result[col].fillna(fill_value)
    return result

missing_before = {
    name: df.isna().sum().rename(«missing_count»).to_frame()
    for name, df in datasets.items()
}

processed_missing = {
    name: fill_missing_values(df)
    for name, df in datasets.items()
}

for name, table in missing_before.items():
    print(«=» * 100)
    print(f»{name} - пропущені значення ДО обробки»)
    display(table[table[«missing_count»] > 0] if
(table[«missing_count»] > 0).any() else
pd.DataFrame({«missing_count»: [«Пропуски відсутні»]},
index=[«-»]))

for name, df in processed_missing.items():
    total_missing = int(df.isna().sum().sum())
    print(f»{name}: пропущених значень після обробки =
{total_missing}»)
```

Результати виконання цього коду, що відображають кількість пропусків у кожному з датасетів, наведено на рисунку 3.2. Отримана інформація використовується для подальшої обробки пропущених значень та підготовки даних до навчання моделей машинного навчання.

```
=====
Titanic – пропущені значення ДО обробки

missing_count
Age          86
Fare          1
Cabin        327

=====
CreditCard – пропущені значення ДО обробки

missing_count
- Пропуски відсутні

=====
WineQT – пропущені значення ДО обробки

missing_count
- Пропуски відсутні

Titanic: пропущених значень після обробки = 0
CreditCard: пропущених значень після обробки = 0
WineQT: пропущених значень після обробки = 0
```

Рисунок 3.2 – Результат виконання коду

Результати аналізу пропущених значень показали, що наявність пропусків спостерігається лише у датасеті Titanic. У цьому наборі даних найбільша кількість пропущених значень виявлена в ознаці Cabin, де відсутні значення для 327 об'єктів. Також пропуски маємо в Age (86 записів) та в ознаці Fare (1 запис). Наявність таких пропусків пов'язана з особливостями формування історичних даних про пасажирів корабля.

У датасетах Credit Card Fraud та WineQT пропущені значення відсутні. Усі ознаки цих наборів даних містять повні записи для кожного об'єкта

вибірки. Це значно спрощує процес їх подальшої обробки та підготовки до навчання моделей машинного навчання.

Після виконання процедур обробки пропущених значень були сформовані очищені набори даних, у яких відсутні незаповнені або некоректні записи. Такий етап є критично важливим для забезпечення якості вхідних даних, оскільки наявність пропусків може призводити до викривлення статистичних властивостей вибірки та зниження ефективності моделей машинного навчання.

Зокрема, у датасеті Titanic пропущені значення були виявлені в окремих ознаках та усунені із застосуванням відповідних методів обробки даних, що дозволило сформулювати повноцінну навчальну вибірку. Аналогічні процедури були виконані для інших датасетів дослідження, що забезпечило їх уніфіковану структуру. Перевірка показала відсутність пропущених значень у всіх трьох наборах даних, що підтверджує коректність очищення та забезпечує надійну основу для подальших експериментів.

Наступним етапом підготовки даних є кодування категоріальних ознак. У багатьох наборах даних частина змінних представлена у вигляді текстових або номінальних значень, що описують певні категорії. Прикладом таких ознак можуть бути стать пасажирів, порт посадки або інші якісні характеристики об'єктів. Більшість алгоритмів машинного навчання працюють виключно з числовими даними, тому перед навчанням моделей такі ознаки необхідно перетворити у числовий формат.

Процес перетворення категоріальних змінних у числові значення називається кодуванням категоріальних ознак. Його основною метою є забезпечення можливості використання категоріальних даних у математичних моделях машинного навчання. Після кодування кожна категорія представлена у вигляді числового значення або набору числових ознак, що дозволяє алгоритмам коректно обробляти відповідні дані.

У межах даного дослідження кодування застосовувалося до ознак текстового типу. Зокрема, у датасеті Titanic наявні категоріальні змінні, такі як Sex та Embarked, що містять текстові значення. Для забезпечення можливості їх використання в алгоритмах машинного навчання ці ознаки були перетворені у числовий формат із застосуванням відповідних методів кодування.

Для виконання кодування категоріальних змінних було розроблено програмний код, який автоматично визначає ознаки текстового типу та виконує їх перетворення у числові змінні. Такий підхід дозволяє уніфікувати структуру даних і підготувати їх до подальшого використання в алгоритмах машинного навчання, забезпечуючи коректну обробку категоріальних ознак та сумісність із вимогами більшості моделей.

Код для виконання кодування категоріальних ознак наведено у лістингу 3.3.

Лістинг 3.3 – Кодування категоріальних ознак

```
def encode_categorical_features(df: pd.DataFrame) ->
pd.DataFrame:
    result = df.copy()

    categorical_cols =
result.select_dtypes(include=[«object», «category»,
«bool»]).columns.tolist()
    if categorical_cols:
        result = pd.get_dummies(result, columns=categorical_cols,
drop_first=True)
    return result

encoded_datasets = {name: encode_categorical_features(df)
for name, df in processed_missing.items()}
for name, df in encoded_datasets.items():
    print(«=» * 100)
    print(f»{name} - розмірність після кодування:
{df.shape}»)
    display(df.head())
```

Результати виконання цього коду, що демонструють зміну структури наборів даних після кодування категоріальних змінних, наведено на рисунку 3.3. Отримані перетворені набори даних використовуються на наступних етапах підготовки даних та проведення експериментів. Це дозволяє забезпечити коректне представлення категоріальних ознак у числовому форматі, придатному для подальшого навчання моделей машинного навчання, а також підвищує узгодженість і якість вхідних даних.

=====

Titanic – розмірність після кодування: (418, 864)

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Name_Abelseth, Miss. Karen Marie	Name_Abelseth, Mr. Olaus Jorgensen	Name_Abrahamsson, Mr. Abraham August Johannes	Name_Abraham, Mrs. Joseph (Sophie Halaut Easu)	Name_Aks, Master. Philip Frank	Name_Aldworth, Mr. Charles Augustus	Name_Allison, Mr. Hudson Joshua Creighton	N
0	892	0	3	34.5	0	0	7.8292	False	False	False	False	False	False	False	
1	893	1	3	47.0	1	0	7.0000	False	False	False	False	False	False	False	
2	894	0	2	62.0	0	0	9.6875	False	False	False	False	False	False	False	
3	895	0	3	27.0	0	0	8.6625	False	False	False	False	False	False	False	
4	896	1	3	22.0	1	1	12.2875	False	False	False	False	False	False	False	

=====

CreditCard – розмірність після кодування: (284887, 31)

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169	1.468177	-0.470401
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772	0.635558	0.463917
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946	2.345865	-2.890083
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924	-0.631418	-1.059647
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670	0.175121	-0.451449

=====

WineQT – розмірність після кодування: (1143, 13)

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	1
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	2
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	3
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4

Рисунок 3.3 – Результат виконання коду

Результати виконання кодування категоріальних ознак показали, що після цього етапу структура наборів даних була приведена до числового формату, придатного для використання алгоритмами машинного навчання. Основною метою цього перетворення було усунення текстових змінних та представлення всіх ознак у вигляді числових значень.

Найбільші зміни відбулися у датасеті Titanic, оскільки він містив кілька категоріальних ознак. Після застосування процедури кодування

категоріальних змінних кількість ознак у цьому наборі даних суттєво збільшилася. Зокрема, розмірність датасету змінилася з 12 до 864 стовпців. Це пояснюється тим, що для текстових змінних було застосовано кодування, у результаті якого кожна окрема категорія була представлена окремою числовою ознакою.

У датасеті Credit Card Fraud кодування категоріальних змінних не призвело до змін структури даних. Це пов'язано з тим, що всі ознаки цього набору даних уже представлені у числовому вигляді. Як результат, після виконання процедури кодування розмірність датасету залишилася незмінною та становить 284807 об'єктів і 31 ознаку.

Аналогічна ситуація спостерігається і у датасеті WineQT, який також складається виключно з числових ознак. Усі характеристики вина представлені у вигляді числових показників фізико-хімічних властивостей. Тому після виконання процедури кодування структура набору даних залишилася без змін і становить 1143 об'єкти та 13 ознак.

Наступним етапом підготовки даних є нормалізація та масштабування числових ознак. У наборах даних, що використовуються у задачах машинного навчання, значення різних ознак можуть мати суттєво різні масштаби. Наприклад, одна змінна може приймати значення у межах від 0 до 1, тоді як інша – у межах десятків або сотень. Така різниця масштабів може негативно впливати на процес навчання моделей.

Алгоритми машинного навчання, особливо ті, що базуються на обчисленні відстаней або оптимізації параметрів, можуть бути чутливими до масштабу ознак. Якщо одна змінна має значно більші значення порівняно з іншими, вона може непропорційно впливати на результати навчання моделі. У результаті модель може надавати більшої ваги саме цій ознаці, навіть якщо вона не є найбільш інформативною.

Для усунення такої проблеми застосовується процедура масштабування числових ознак. Вона передбачає приведення значень змінних до спільного масштабу, що забезпечує більш коректну роботу

алгоритмів машинного навчання. У межах даного дослідження було використано стандартне масштабування, яке полягає у перетворенні значень ознак таким чином, щоб їх середнє значення дорівнювало нулю, а стандартне відхилення – одиниці.

Застосування масштабування дозволяє зробити внесок кожної ознаки у процес навчання більш збалансованим. У результаті алгоритми машинного навчання можуть ефективніше знаходити закономірності у даних та формувати більш стабільні моделі.

Для виконання процедури нормалізації та масштабування було розроблено програмний код, який автоматично визначає числові ознаки у кожному з датасетів та застосовує до них відповідне перетворення. Код для виконання цього етапу підготовки даних наведено у лістингу 3.4

Лістинг 3.4 – Нормалізація та масштабування числових ознак

```
TARGET_COLUMNS = {
    «Titanic»: «Survived»,
    «CreditCard»: «Class»,
    «WineQT»: «Quality»,}

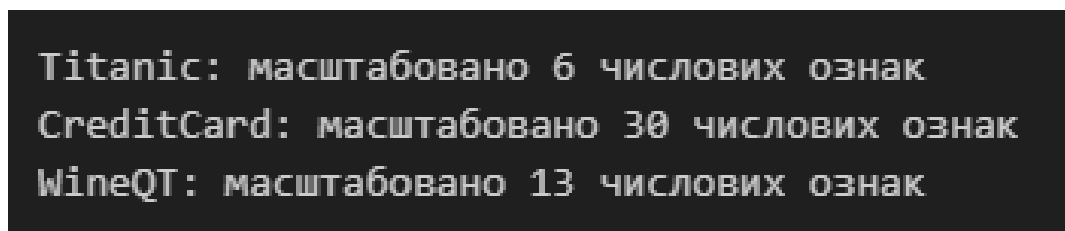
def scale_numeric_features(df: pd.DataFrame, target_col:
str): result = df.copy()
    feature_cols = [c for c in result.columns if c !=
target_col] numeric_cols
result[feature_cols].select_dtypes(include=[np.number]).column
s.tolist()
scaler = StandardScaler()
    if numeric_cols: result[numeric_cols] =
scaler.fit_transform(result[numeric_cols])
    return result, numeric_cols, scaler

scaled_datasets = {}
scalers = {}
scaled_feature_info = {} for name, df in
encoded_datasets.items():
```

Продовження лістингу 3.4

```
scaled_df, scaled_cols, scaler = scale_numeric_features(df,
TARGET_COLUMNS[name])
    scaled_datasets[name] = scaled_df
    scalers[name] = scaler
    scaled_feature_info[name] = scaled_cols
for name, cols in scaled_feature_info.items():
    print(f»{name}:    масштабовано    {len(cols)}    числових
ознак»)
```

Результати виконання масштабування числових ознак, що демонструють змінені значення ознак після застосування перетворення, наведено на рисунку 3.4.



```
Titanic: масштабовано 6 числових ознак
CreditCard: масштабовано 30 числових ознак
WineQT: масштабовано 13 числових ознак
```

Рисунок 3.4 – Результат виконання коду

Результати виконання масштабування числових ознак показують, що процедура була успішно застосована до всіх наборів даних, які використовуються у дослідженні. Під час виконання цього етапу було автоматично визначено числові змінні в кожному датасеті та виконано їх перетворення до спільного масштабу.

Для датасету Titanic було масштабовано 6 числових ознак. До них належать змінні, що характеризують вік пасажирів, вартість квитка, кількість родичів на борту та інші числові характеристики. Масштабування цих змінних дозволяє зменшити вплив різних масштабів значень на процес навчання моделей.

У датасеті Credit Card Fraud масштабування було застосовано до 30 числових ознак. Більшість з них представлені у вигляді компонент V1–V28, які були отримані в результаті застосування методу головних компонент. Додатково було масштабовано ознаки «Time» та «Amount». Оскільки цей датасет повністю складається з числових змінних, процедура масштабування була застосована практично до всіх ознак.

Для датасету WineQT масштабування було виконано для 13 числових ознак, які описують фізико-хімічні характеристики вина, такі як кислотність, щільність, вміст алкоголю та інші показники. Приведення цих змінних до єдиного масштабу забезпечує коректніші умови для навчання моделей машинного навчання.

Наступним етапом підготовки даних є формування цільових змінних. У задачах машинного навчання цільова змінна визначає той показник, який модель повинна передбачати на основі значень інших ознак. Саме вона виступає еталонним результатом, із яким порівнюються прогнозовані значення під час навчання та оцінювання моделі.

Правильне визначення цільової змінної є важливим етапом побудови моделі машинного навчання, оскільки воно безпосередньо визначає тип задачі та методи її розв'язання. Залежно від характеру цільової змінної задачі можуть належати до задач класифікації або регресії. У даному дослідженні використовуються задачі класифікації, у яких модель повинна визначати належність об'єкта до одного з класів.

Для кожного з обраних датасетів було визначено відповідну цільову змінну. У наборі даних Titanic цільовою змінною виступає ознака Survived, яка відображає факт виживання пасажирів під час катастрофи. У датасеті Credit Card Fraud цільовою змінною є ознака Class, що вказує на наявність або відсутність шахрайської транзакції. У наборі даних WineQT цільовою змінною є показник quality, який характеризує оцінку якості вина.

Формування цільових змінних передбачає відокремлення відповідного стовпця з датасету та створення матриці ознак і вектора

цілових значень. Такий підхід забезпечує коректне розділення незалежних і залежних змінних та зручність використання даних у процесі навчання моделей машинного навчання.

Для реалізації цього етапу було розроблено програмний код, який автоматично визначає цільові змінні для кожного набору даних та формує відповідні структури для подальшого навчання моделей. Це забезпечує уніфікацію підготовки даних і коректне розділення ознак та цільової змінної. Реалізацію наведено у лістингу 3.5.

Лістинг 3.5 – Формування цільових змінних

```
TARGET_COLUMNS = {
    «Titanic»: «Survived»,
    «CreditCard»: «Class»,
    «WineQT»: «Quality»,}

def resolve_target_column(df: pd.DataFrame, expected_name:
str) -> str:
    if expected_name in df.columns: return expected_name
    lower_map = {col.lower(): col for col in df.columns}
    if expected_name.lower() in lower_map:
        return lower_map[expected_name.lower()]
    raise KeyError(f«Цільову змінну '{expected_name}' не
знайдено. Доступні стовпці: {list(df.columns)}»)

prepared_targets = {}
for name, df in scaled_datasets.items():
    target_col = resolve_target_column(df, TARGET_COLUMNS[name])
    X = df.drop(columns=[target_col])
    y = df[target_col].copy()
    prepared_targets[name] = (X, y)
    print(«=» * 100)
    print(f»{name}»)
    print(f»Цільова змінна: {target_col}»)
    print(«Розподіл класів / значень:»)
    display(y.value_counts(dropna=False).sort_index().rena
me(«count»).to_frame())
```

Результати виконання цього етапу підготовки даних, які демонструють сформовані набори ознак та цільових змінних для кожного датасету, наведено на рисунку 3.5.

Результати виконання формування цільових змінних демонструють, що для кожного з досліджуваних наборів даних було коректно визначено відповідну цільову змінну та проаналізовано розподіл її значень. Це дозволяє оцінити баланс класів у даних та визначити можливі особливості, які можуть впливати на процес навчання моделей машинного навчання.

=====	=====	=====
Titanic	CreditCard	WineQT
Цільова змінна: Survived	Цільова змінна: Class	Цільова змінна: quality
Розподіл класів / значень:	Розподіл класів / значень:	Розподіл класів / значень:
count	count	count
Survived	Class	quality
0 266	0 284315	-3.298742 6
1 152	1 492	-2.057233 33
		-0.815724 483
		0.425784 462
		1.667293 143
		2.908801 16

Рисунок 3.5 – Результат виконання коду

Для датасету Titanic цільовою змінною є показник Survived, який відображає факт виживання пасажирів. Аналіз розподілу значень показує, що у вибірці присутні 266 випадків загибелі (клас 0) та 152 випадки виживання (клас 1). Такий розподіл свідчить про певний дисбаланс класів, проте він не є критичним і дозволяє проводити подальший аналіз та навчання моделей без значних обмежень.

У датасеті Credit Card Fraud цільовою змінною є ознака Class, яка позначає шахрайські транзакції. Розподіл демонструє суттєвий дисбаланс класів: 284315 транзакцій належать до звичайних операцій (клас 0), тоді як лише 492 транзакції є шахрайськими (клас 1). Такий сильний дисбаланс є

характерною особливістю задач виявлення шахрайства та створює додаткові труднощі для алгоритмів машинного навчання.

Для датасету WineQT цільовою змінною є показник quality, який характеризує оцінку якості вина. Після попереднього масштабування значення цієї змінної представлені у нормалізованому вигляді, що пояснює наявність як від'ємних, так і додатних значень. Розподіл значень показує, що більшість спостережень зосереджена навколо центральних значень, тоді як крайні значення зустрічаються значно рідше.

Наступним етапом підготовки даних є розділення наборів даних на навчальні та тестові вибірки. Цей етап є стандартною практикою у задачах машинного навчання та використовується для об'єктивної оцінки якості побудованих моделей. Основна ідея полягає у тому, що модель навчається лише на частині доступних даних, після чого її робота перевіряється на іншій частині, яка не використовувалася під час навчання.

Навчальна вибірка використовується безпосередньо для побудови моделі машинного навчання. На основі цих даних алгоритм визначає закономірності між вхідними ознаками та цільовою змінною. Саме під час роботи з навчальною вибіркою відбувається налаштування параметрів моделі та формування внутрішніх правил прийняття рішень.

Тестова вибірка, у свою чергу, використовується для незалежної оцінки якості побудованої моделі. Оскільки ці дані не брали участі у процесі навчання, результати прогнозування на тестовій вибірці дозволяють оцінити здатність моделі узагальнювати отримані знання та застосовувати їх до нових, раніше невідомих даних.

Розділення даних на навчальну та тестову вибірки дозволяє уникнути явища перенавчання, при якому модель надмірно пристосовується до навчальних даних і демонструє погані результати на нових даних. Завдяки використанню тестової вибірки можна більш об'єктивно оцінити реальну якість моделі та її здатність до узагальнення.

У межах даного дослідження кожен із обраних датасетів було розділено на навчальну та тестову частини. Навчальна вибірка використовується для навчання алгоритмів машинного навчання, тоді як тестова вибірка застосовується для оцінювання якості моделей та обчислення відповідних метрик. Код, що реалізує процедуру розділення даних, наведено у лістингу 3.6. Результати виконання розділення датасетів на навчальні та тестові вибірки наведено на рисунку 3.6.

Лістинг 3.6 – Розділення датасетів на навчальні та тестові вибірки

```
splits = {}
for name, (X, y) in prepared_targets.items():
    stratify_value = y if y.nunique() < 30 else None
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42,
        stratify=stratify_value)
    splits[name] = {
        «X_train»: X_train,
        «X_test»: X_test,
        «y_train»: y_train,
        «y_test»: y_test}
summary_split = []
for name, split in splits.items():
    summary_split.append({
        «dataset»: name,
        «X_train_shape»: split[«X_train»].shape,
        «X_test_shape»: split[«X_test»].shape,
        «y_train_shape»: split[«y_train»].shape,
        «y_test_shape»: split[«y_test»].shape,})
pd.DataFrame(summary_split)
BASE_OUTPUT_DIR = Path(«./prepared_datasets»)
BASE_OUTPUT_DIR.mkdir(exist_ok=True)
for name, split in splits.items():
    X_train = split[«X_train»].copy()
    X_test = split[«X_test»].copy()
```

Продовження лістингу 3.6

```

y_train = split[«y_train»].copy()
y_test = split[«y_test»].copy()
train_df = X_train.copy()
train_df[TARGET_COLUMNS[name]] = y_train.values
test_df = X_test.copy()
test_df[TARGET_COLUMNS[name]] = y_test.values

train_path = BASE_OUTPUT_DIR /
f»{name.lower()}_train_prepared.csv»
test_path = BASE_OUTPUT_DIR /
f»{name.lower()}_test_prepared.csv»

train_df.to_csv(train_path, index=False)
test_df.to_csv(test_path, index=False)
print(f»{name}:»)

print(f»          train      ->
{train_path.resolve()}»)print(f»          test      ->
{test_path.resolve()}»)

```

	dataset	X_train_shape	X_test_shape	y_train_shape	y_test_shape
0	Titanic	(334, 863)	(84, 863)	(334,)	(84,)
1	CreditCard	(227845, 30)	(56962, 30)	(227845,)	(56962,)
2	WineQT	(914, 12)	(229, 12)	(914,)	(229,)

prepared_datasets

creditcard_test_prepared.csv

creditcard_train_prepared.csv

titanic_test_prepared.csv

titanic_train_prepared.csv

wineqt_test_prepared.csv

wineqt_train_prepared.csv

Рисунок 3.6 – Результат виконання коду

Результати виконання розділення датасетів на навчальні та тестові вибірки показують, що кожен набір даних був коректно поділений на дві частини: дані для навчання моделей та дані для подальшої оцінки їхньої якості. Отримані вибірки були також збережені у вигляді окремих файлів, що дозволяє використовувати їх у подальших етапах експерименту без повторного виконання підготовчих процедур.

Для датасету Titanic було сформовано вибірку розміром 334 об'єкти та тестову вибірку розміром 84 об'єкти. Кількість ознак у наборі після

виконання всіх попередніх етапів підготовки становить 863. Відповідні цільові змінні мають розмірності 334 та 84 елементи для навчальної та тестової вибірок відповідно.

У випадку датасету Credit Card Fraud розділення було виконано для значно більшого обсягу даних. Навчальна вибірка містить 227845 транзакцій, тоді як тестова вибірка включає 56962 об'єкти. Кількість ознак у цьому наборі даних становить 30. Такий обсяг даних забезпечує можливість отримання статистично надійних результатів під час подальших експериментів.

Для датасету WineQT навчальна вибірка складається з 914 об'єктів, а тестова – з 229 об'єктів. Кількість ознак після попередньої підготовки становить 12. Отримані розміри вибірок забезпечують достатній обсяг даних як для навчання моделей, так і для їх подальшої перевірки.

3.2 Формування збурених наборів даних

У межах даного дослідження збурення вносилися виключно до навчальних наборів даних, тоді як тестові вибірки залишалися незмінними. Такий підхід дозволяє забезпечити коректність експерименту та об'єктивність оцінювання моделей машинного навчання. Моделі навчаються на змінених навчальних даних, після чого їх якість оцінюється на однаковій тестовій вибірці. Завдяки цьому можна визначити, наскільки зміни у навчальних даних впливають на здатність моделей узагальнювати знання та формувати коректні прогнози для нових об'єктів.

Збурення застосовувалися до навчальних наборів даних, сформованих на попередньому етапі підготовки даних. Для кожного з обраних датасетів формувалися окремі сценарії змін, що відповідають характерним типам порушень структури даних, які можуть виникати у практичних задачах машинного навчання. При цьому типи збурень підбиралися з урахуванням

особливостей кожного набору даних, його розміру, структури ознак та типу цільової змінної.

Одним із перших типів збурення, що застосовувався у межах дослідження, є внесення шуму у вхідні ознаки навчальних даних. Такий тип змін моделює ситуації, коли значення ознак можуть містити випадкові неточності, що виникають у процесі вимірювання, збору або передачі даних. У практичних задачах машинного навчання подібні помилки є досить поширеним явищем, оскільки дані можуть формуватися з різних джерел, використовувати різні методи вимірювання або проходити через декілька етапів обробки.

Внесення шуму до ознак дозволяє дослідити, наскільки алгоритми машинного навчання є стійкими до невеликих випадкових змін у значеннях вхідних параметрів. Якщо модель демонструє значне погіршення результатів навіть при незначних змінах значень ознак, це свідчить про її високу чутливість до змін у даних. У свою чергу, стабільні алгоритми повинні демонструвати відносно невеликі зміни у результатах прогнозування при появі випадкового шуму.

У межах даного дослідження шум додавався до числових ознак навчальних наборів даних для моделювання похибок вимірювання та варіативності реальних даних. Для кожної ознаки генерувалося випадкове відхилення, величина якого визначалася на основі її стандартного відхилення, що дозволяло адаптувати інтенсивність шуму до характерного діапазону значень. Такий підхід забезпечує коректне масштабування збурень і запобігає надмірному викривленню структури даних.

Додавання шуму виконувалося контрольовано, що дозволяло моделювати різні рівні збурень і аналізувати їх вплив на стабільність та чутливість алгоритмів машинного навчання. Це створює умови для оцінювання робастності моделей у наближених до реальних сценаріях експлуатації, де дані часто містять випадкові похибки або неточності вимірювання.

З метою дослідження впливу різної інтенсивності змін було сформовано декілька рівнів шуму, що відрізняються часткою об'єктів, до яких застосовується збурення, а також величиною випадкового відхилення. Це дозволяє оцінити, як змінюється поведінка моделей машинного навчання при поступовому збільшенні рівня спотворення навчальних даних.

Програмна реалізація процедури формування збурення типу noise була виконана засобами мови програмування Python із використанням бібліотек для роботи з табличними даними та генерації випадкових значень. Реалізований код дозволяє формувати навчальні набори даних із контрольованим додаванням шуму до числових ознак та забезпечує можливість регулювання його інтенсивності залежно від стандартного відхилення кожної ознаки. Такий підхід дає змогу моделювати різні рівні збурень і аналізувати їх вплив на стабільність моделей машинного навчання. Код наведено у лістингу 3.7.

Лістинг 3.7 – Код для формування збурення типу noise

```
generated_files = []

for dataset_name, df in datasets.items():
    target_col = TARGET_COLUMNS[dataset_name]

    print(«=» * 100)
    print(f»Датасет: {dataset_name}»)
    print(f»Цільова змінна: {target_col}»)

    numeric_cols = get_numeric_feature_columns(df,
target_col)

    print(f»Кількість числових ознак для внесення шуму:
{len(numeric_cols)}»)

    for level_name, params in NOISE_LEVELS.items():
        perturbed_df = add_noise_to_numeric_features(
            df=df,
            target_col=target_col,
```

Продовження лістингу 3.7

```

        noise_fraction=params[«noise_fraction»],
        noise_std=params[«noise_std»],
        random_state=RANDOM_STATE
    )

    file_name =
f»{dataset_name.lower()}_train_noise_{level_name}.csv»
    output_path = OUTPUT_DIR / file_name
    perturbed_df.to_csv(output_path, index=False)

    generated_files.append({
        «dataset»: dataset_name,
        «perturbation»: «noise»,
        «level»: level_name,
        «rows»: perturbed_df.shape[0],
        «columns»: perturbed_df.shape[1],
        «file»: str(output_path)
    })

print(
    f»Створено: {file_name} | «
        f»fraction={params['noise_fraction']},
std={params['noise_std']}»
)

```

Наступним типом збурення, що застосовувався у межах дослідження, є формування штучних викидів у навчальних даних. Викиди представляють собою значення ознак, які суттєво відрізняються від основної маси спостережень і можуть виникати внаслідок помилок вимірювання, некоректної обробки даних або рідкісних аномальних ситуацій. У практичних задачах машинного навчання наявність таких значень може

значно впливати на процес навчання моделей та призводити до зниження якості прогнозування.

Формування штучних викидів дозволяє змодельовати ситуації, коли частина навчальних даних містить екстремальні значення ознак. У таких умовах алгоритми машинного навчання можуть поводитися по-різному: деякі з них є відносно стійкими до наявності аномальних значень, тоді як інші можуть значно змінювати структуру моделі під впливом навіть невеликої кількості таких спостережень. Дослідження реакції алгоритмів на появу викидів дозволяє оцінити їхню робастність до подібних порушень структури даних.

У межах даного дослідження штучні викиди формувалися шляхом зміщення значень числових ознак для випадково обраної частини об'єктів навчальної вибірки. Для цього до значень ознак додавалося значне відхилення, величина якого визначалась на основі стандартного відхилення відповідної ознаки. Такий підхід дозволяє сформувати значення, що істотно виходять за межі типового діапазону змін ознаки та можуть розглядатися як аномальні спостереження.

Аналогічно до попереднього типу збурення, для викидів було передбачено декілька рівнів інтенсивності, що відрізняються часткою об'єктів із аномальними значеннями та величиною відхилення ознак. Це дозволяє дослідити вплив різної кількості викидів на результати навчання моделей машинного навчання.

Програмна реалізація процедури формування штучних викидів була виконана з використанням мови програмування Python та бібліотек для обробки табличних даних. Код, що реалізує створення навчальних наборів даних із доданими викидами, наведено у лістингу 3.8.

Лістинг 3.8 – Код для формування збурення типу outlines

```
generated_files = []  
for dataset_name, df in datasets.items():
```

Продовження лістингу 3.8

```

    target_col = TARGET_COLUMNS[dataset_name]
    print('=' * 100)
    print(f'Датасет: {dataset_name}')
    print(f'Цільова змінна: {target_col}')
    numeric_cols = get_numeric_feature_columns(df,
target_col)

    print(f'Кількість числових ознак для внесення викидів:
{len(numeric_cols)}')
    for level_name, params in OUTLIER_LEVELS.items():
        perturbed_df = add_outliers_to_numeric_features(
            df=df,
            target_col=target_col,
            outlier_fraction=params['outlier_fraction'],
            multiplier=params['multiplier'],
            random_state=RANDOM_STATE )

        file_name =
f»{dataset_name.lower()}_train_outliers_{level_name}.csv»
        output_path = OUTPUT_DIR / file_name
        perturbed_df.to_csv(output_path, index=False)

        generated_files.append({
            'dataset': dataset_name,
            'perturbation': 'outliers',
            'level': level_name,
            'rows': perturbed_df.shape[0],
            'columns': perturbed_df.shape[1],
            'file': str(output_path)})

    print(
        f»Створено: {file_name} | «
        f»fraction={params['outlier_fraction']},
multiplier={params['multiplier']}»»)

```

Ще одним типом збурення, що застосовувався у межах дослідження, є внесення помилок у мітки класів навчальних даних. Подібні зміни моделюють ситуації, коли під час розмітки даних можуть виникати помилки, пов'язані з людським фактором, неточностями у визначенні класу або некоректною обробкою інформації. У реальних задачах машинного навчання такі помилки можуть з'являтися, наприклад, під час ручної анотації даних або автоматичного формування навчальних вибірок із різних джерел.

Наявність помилок у мітках класів може суттєво впливати на процес навчання моделей машинного навчання, оскільки алгоритми намагаються встановити залежності між ознаками та цільовою змінною на основі некоректних прикладів. У результаті модель може формувати помилкові правила класифікації або демонструвати зниження якості прогнозування. Дослідження впливу подібних помилок дозволяє оцінити, наскільки алгоритми машинного навчання є чутливими до неточностей у навчальній розмітці.

У межах даного дослідження збурення типу *label noise* реалізовувалося шляхом випадкової зміни значень цільової змінної для частини об'єктів навчальної вибірки. Для вибраних спостережень мітка класу змінювалася на протилежну, що дозволяє змодельовати ситуацію неправильної класифікації під час формування навчального набору даних. Такий підхід є характерним для досліджень робастності алгоритмів машинного навчання до помилок у розмітці.

Для більш детального аналізу впливу цього типу збурення було сформовано декілька рівнів інтенсивності, що відрізняються часткою об'єктів, для яких змінюється значення цільової змінної. Це дозволяє дослідити, як змінюється якість та стабільність моделей машинного навчання при поступовому збільшенні кількості помилок у навчальних даних.

Програмна реалізація процедури формування збурення типу label noise була виконана засобами мови програмування Python. Код, що реалізує створення навчальних наборів даних із внесеними помилками у мітках класів, наведено у лістингу 3.9.

Лістинг 3.9 – Код для формування збурення типу label noise

```
generated_files = []

for dataset_name, df in datasets.items():
    target_col = TARGET_COLUMNS[dataset_name]
    print('=' * 100)
    print(f'Датасет: {dataset_name}')
    print(f'Цільова змінна: {target_col}')
    for level_name, params in LABEL_NOISE_LEVELS.items():
        perturbed_df = add_label_noise_binary(
            df=df,
            target_col=target_col,
            label_noise_fraction=params['label_noise_fraction'],
            random_state=RANDOM_STATE)
        file_name =
f»{dataset_name.lower()}_train_label_noise_{level_name}.csv»
        output_path = OUTPUT_DIR / file_name
        perturbed_df.to_csv(output_path, index=False)
        generated_files.append({
            'dataset': dataset_name,
            'perturbation': 'label_noise',
            'level': level_name,
            'rows': perturbed_df.shape[0],
            'columns': perturbed_df.shape[1],
            'file': str(output_path)})
    print(
        f»Створено: {file_name} | «
        f»fraction={params['label_noise_fraction']}» )
```

Наступним типом збурення, що застосовувався у межах експериментального дослідження, є зміна обсягу навчальної вибірки. Такий тип збурення дозволяє змодельовати ситуацію, коли алгоритми машинного навчання навчаються на обмеженій кількості даних. У реальних задачах подібні ситуації можуть виникати у випадках, коли доступні навчальні дані є недостатніми, частина даних була втрачена або отримання нових спостережень потребує значних витрат часу та ресурсів.

Зменшення обсягу навчальної вибірки може суттєво впливати на процес навчання моделей машинного навчання. При зменшенні кількості доступних прикладів алгоритми мають менше інформації для формування узагальнених закономірностей між вхідними ознаками та цільовою змінною. У результаті модель може демонструвати більшу нестабільність результатів, а також зниження якості прогнозування на нових даних.

У межах даного дослідження збурення типу `dataset size` реалізовувалося шляхом формування нових навчальних наборів даних із меншою кількістю об'єктів порівняно з початковою вибіркою. Для цього з вихідного навчального датасету випадковим чином відбиралася лише певна частка спостережень, тоді як інші об'єкти виключалися з процесу навчання. При цьому відбір виконувався випадковим чином, що дозволяє зберегти загальну структуру розподілу даних.

З метою дослідження впливу різного обсягу даних на результати навчання було сформовано декілька варіантів навчальних вибірок, які відрізняються часткою об'єктів, що використовуються для навчання моделі. Це дозволяє оцінити, наскільки стабільно поведуться алгоритми машинного навчання при поступовому зменшенні кількості доступних навчальних прикладів.

Програмна реалізація процедури формування збурення типу `dataset size` була виконана з використанням мови програмування Python та бібліотек для обробки табличних даних. Код, що реалізує створення навчальних наборів даних зі змінним обсягом вибірки, наведено у лістингу 3.10.

Лістинг 3.10 – Код для формування збурення типу dataset size

```

generated_files = []

for level_name, fraction in SIZE_LEVELS.items():

    reduced_df = df.sample(
        frac=fraction,
        random_state=RANDOM_STATE
    ).reset_index(drop=True)

    file_name = f'creditcard_train_size_{level_name}.csv'
    output_path = OUTPUT_DIR / file_name

    reduced_df.to_csv(output_path, index=False)

    generated_files.append({
        'level': level_name,
        'fraction': fraction,
        'rows': reduced_df.shape[0],
        'columns': reduced_df.shape[1],
        'file': str(output_path)
    })

    print(f»Створено {file_name} |
rows={reduced_df.shape[0]}»
)

```

Ще одним типом збурення, що застосовувався у межах дослідження, є формування дисбалансу класів у навчальних даних. Дисбаланс класів виникає у випадках, коли кількість об'єктів одного класу суттєво перевищує кількість об'єктів іншого класу. У практичних задачах машинного навчання

подібна ситуація є досить поширеною, зокрема у задачах виявлення шахрайства, медичної діагностики або пошуку рідкісних подій, де позитивні приклади зустрічаються значно рідше.

Наявність дисбалансу класів може істотно впливати на процес навчання моделей машинного навчання. Багато алгоритмів оптимізують загальну точність класифікації, тому при значному дисбалансі вони можуть схилитися до прогнозування більш чисельного класу, ігноруючи менш представлений клас. У результаті модель може демонструвати високу загальну точність, але при цьому погано визначати об'єкти міноритарного класу.

З метою моделювання подібної ситуації у межах дослідження було сформовано навчальні набори даних зі штучно підсиленним дисбалансом класів. Для цього з навчальної вибірки випадковим чином вилучалася частина об'єктів міноритарного класу, що призводило до поступового збільшення співвідношення між класами. Такий підхід дозволяє моделювати реальні сценарії, у яких один із класів представлений значно меншою кількістю прикладів, та оцінювати вплив такого дисбалансу на якість і стабільність моделей машинного навчання.

Для більш детального аналізу впливу дисбалансу було сформовано декілька варіантів навчальних наборів даних із різними рівнями дисбалансу між класами. Це дозволяє дослідити, як змінюється поведінка алгоритмів машинного навчання при поступовому зменшенні кількості об'єктів міноритарного класу та оцінити стійкість моделей до порушення початкового розподілу даних.

Програмна реалізація процедури формування збурення типу `class imbalance` була виконана засобами мови програмування Python із використанням бібліотек для обробки табличних даних та генерації вибірок із заданими характеристиками розподілу класів. Реалізований код дозволяє контролювати ступінь дисбалансу та формувати кілька варіантів навчальних вибірок із поступовим посиленням нерівномірності класів, що

забезпечує можливість порівняльного аналізу впливу цього фактора на якість моделей. Код наведено у лістингу 3.11.

Лістинг 3.11 – Код для формування збурення типу class imbalance

```
generated_files = []

for dataset_name, df in datasets.items():

    target_col = TARGET_COLUMNS[dataset_name]
    minority_class, majority_class, class_counts =
get_minority_and_majority_classes(df[target_col])

    print('=' * 100)
    print(f'Датасет: {dataset_name}')
    print(f'Цільова змінна: {target_col}')
    print(f'Міноритарний клас: {minority_class}')
    print(f'Мажоритарний клас: {majority_class}')

    for level_name, params in IMBALANCE_LEVELS.items():
        perturbed_df = create_class_imbalance(
            df=df,
            target_col=target_col,
            minority_keep_fraction=params['minority_keep_f
raction'],
            random_state=RANDOM_STATE)

        file_name=
f»{dataset_name.lower()}_train_imbalance_{level_name}.csv»
        output_path = OUTPUT_DIR / file_name
        perturbed_df.to_csv(output_path, index=False)
        new_counts =
perturbed_df[target_col].value_counts().sort_index().to_dict()
        generated_files.append
(
{
```

Продовження лістингу 3.11

```

        'dataset': dataset_name,
        'perturbation': 'class_imbalance',
        'level': level_name,
        'rows': perturbed_df.shape[0],
        'columns': perturbed_df.shape[1],

        'minority_keep_fraction':
params['minority_keep_fraction'],
        'class_distribution': str(new_counts),
        'file': str(output_path)})
print(
    f»Створено: {file_name} | «
    f»minority_keep_fraction={params['minority_kee
p_fraction']} | «
    f»distribution={new_counts}»
)

```

У результаті виконання процедур формування збурень було отримано набір модифікованих навчальних даних для кожного розглянутого типу змін. Кожен тип збурення реалізовано у вигляді окремого набору файлів, що містять змінені версії навчальних вибірок. Згенеровані файли організовано у відповідній структурі каталогів, де кожна папка відповідає окремому типу збурення. Загальний вигляд сформованих наборів даних наведено на рисунку 3.7.

Як видно з рисунку 3.7, для кожного типу збурення було сформовано декілька варіантів навчальних наборів даних, що відповідають різним рівням інтенсивності змін. Зокрема, для кожного сценарію збурення створено три рівні: низький, середній та високий. Такий підхід дозволяє дослідити вплив не лише самого факту зміни структури даних, але й ступінь цього впливу на результати навчання моделей машинного навчання.

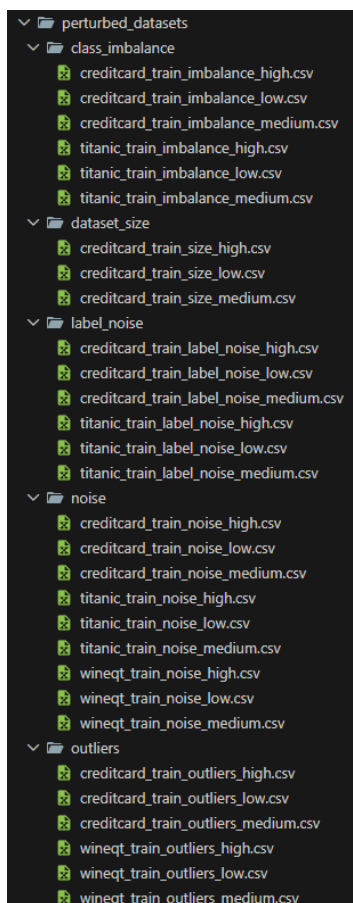


Рисунок 3.7 – Файли збурених наборів даних

Файли зі збуреними даними було структуровано відповідно до типу збурення. Окремі каталоги містять набори даних для сценаріїв додавання шуму до ознак, формування викидів, внесення помилок у мітки класів, зміни обсягу навчальної вибірки та формування дисбалансу класів. У кожному каталозі містяться файли, що відповідають різним датасетам та рівням інтенсивності відповідного типу збурення.

Таким чином, у результаті виконання даного етапу було сформовано повний набір навчальних даних, що включає як базові, так і збурені версії вибірок. Отримані набори даних використовуються на наступному етапі дослідження для навчання моделей машинного навчання та подальшого оцінювання їхньої стабільності і чутливості до різних типів змін у навчальних даних.

Таким чином, було сформовано збурені версії навчальних наборів даних, що моделюють різні типи змін, які можуть виникати у реальних задачах машинного навчання. Було реалізовано п'ять основних типів збурень: додавання шуму до числових ознак, формування викидів, внесення помилок у мітки класів, зменшення обсягу навчальної вибірки та створення дисбалансу класів. Для кожного типу збурення було сформовано декілька рівнів інтенсивності, що дозволяє дослідити вплив різного ступеня змін у даних на результати навчання моделей.

У результаті виконання даного етапу було отримано повний набір модифікованих навчальних даних, які використовуються для подальшого проведення експериментів із навчання моделей машинного навчання та оцінювання їхньої стабільності і чутливості до змін у навчальних даних.

3.3 Проведення експерименту

Експеримент передбачає навчання моделей машинного навчання на базових та збурених наборах даних з подальшим порівнянням результатів їх роботи. Такий підхід дозволяє оцінити, наскільки різні алгоритми здатні зберігати стабільність результатів у випадку змін структури навчальної інформації.

Для проведення експерименту було використано чотири алгоритми машинного навчання, що представляють різні підходи до побудови моделей класифікації: Logistic Regression, Decision Tree, Random Forest та Multilayer Perceptron. Logistic Regression належить до класу лінійних моделей і формує прогноз на основі зваженої лінійної комбінації ознак із подальшим застосуванням логістичної функції. Decision Tree є деревоподібним алгоритмом, який послідовно розбиває простір ознак на підмножини шляхом побудови набору логічних правил, що ведуть до кінцевого рішення. Random Forest відноситься до ансамблевих методів і базується на агрегації великої кількості дерев рішень, що дозволяє зменшити дисперсію моделі та

підвищити її стійкість до змін у даних. Multilayer Perceptron є представником нейронних мереж прямого поширення і здатний апроксимувати складні нелінійні залежності між вхідними ознаками та цільовою змінною за рахунок багатосарової архітектури.

Навчання моделей виконувалося на трьох різних наборах даних: Titanic, Credit Card Fraud та WineQT. Ці датасети відрізняються між собою як за кількістю спостережень, так і за структурою ознак та характером класифікаційної задачі. Зокрема, Titanic містить змішані категоріальні та числові ознаки і характеризується наявністю пропущених значень, що вимагає додаткової попередньої обробки. Credit Card Fraud є сильно незбалансованим набором даних, що створює додаткові виклики для алгоритмів класифікації. WineQT, у свою чергу, містить відносно збалансовані числові ознаки та використовується для задачі багатокласової класифікації. Така різноманітність дозволяє оцінити поведінку моделей у різних умовах складності та структури даних.

Фрагмент програмного коду, що реалізує проведення обчислювального експерименту, наведено у лістингу 3.12.

Лістинг 3.12 – Код експерименту

```
experiment_rows = []

for file_path in perturbed_files:
    dataset_name = dataset_key_from_filename(file_path.name)

    if dataset_name is None:
        continue

    perturbed_train_df = pd.read_csv(file_path)

    X_train_perturbed = perturbed_train_df.drop(columns=[target_col])
```

Продовження Лістингу 3.12

```

y_train_perturbed = perturbed_train_df[target_col]

X_test = base_data[dataset_name]['X_test']
y_test = base_data[dataset_name]['y_test']

for model_name, model in build_models().items():
    acc = train_and_evaluate(
        model,
        X_train_perturbed,
        y_train_perturbed,
        X_test,
        y_test)

    q_base = baseline_lookup[(dataset_name,
model_name)]

    experiment_rows.append({
        'dataset': dataset_name,
        'model': model_name,
        'accuracy': acc,
        'sensitivity_index': sensitivity_index(acc,
q_base),
        'robustness_ratio': robustness_ratio(acc,
q_base)
    })

experiments_df = pd.DataFrame(experiment_rows)

```

Наведений фрагмент коду реалізує основний етап проведення обчислювального експерименту на збурених наборах даних. Його призначення полягає у послідовному проходженні всіх сформованих файлів зі збуреними навчальними вибірками, навчанні на них моделей машинного навчання та обчисленні значень метрик оцінювання. У результаті виконання цього блоку формується зведена таблиця результатів експериментів, яка

надалі використовується для аналізу стабільності та чутливості алгоритмів машинного навчання.

На початку коду створюється порожній список `experiment_rows`, у який надалі записуються результати всіх експериментів. Також формується словник `baseline_lookup`, що містить базові значення точності для кожної пари «датасет – модель». Ці значення були отримані раніше під час навчання моделей на незбурених наборах даних і використовуються як опорні під час обчислення метрик чутливості та робастності.

Далі код проходить циклом по всіх файлах зі збуреними навчальними даними, перелік яких зберігається у змінній `perturbed_files`. Для кожного файлу автоматично визначаються назва датасету, тип збурення та рівень його інтенсивності. Це виконується за допомогою допоміжних функцій `dataset_key_from_filename`, `perturbation_type_from_filename` та `level_from_filename`, які аналізують назву файлу та відносять його до відповідного експериментального сценарію.

Після цього файл зі збуреним набором даних зчитується у таблицю `perturbed_train_df`. Із неї визначається цільова змінна, після чого дані розділяються на матрицю ознак `X_train_perturbed` та вектор цільових значень `y_train_perturbed`. Для датасету WineQT додатково виконується перетворення цільової змінної до цілочисельного формату, оскільки ця задача розглядається як задача класифікації, а значення цільової ознаки після попередньої обробки мають бути приведені до дискретного вигляду.

На наступному етапі для кожного збуреного набору даних вибирається відповідна тестова вибірка `X_test` та `y_test`, яка зберігається у структурі `base_data`. Важливо, що незалежно від типу та рівня збурення навчальних даних тестові вибірки залишаються незмінними. Це дозволяє оцінювати всі моделі в однакових умовах та забезпечує коректність порівняння результатів між різними експериментальними сценаріями.

Усередині циклу по файлах виконується вкладений цикл по всіх моделях машинного навчання, які повертає функція `build_models()`. Для

кожної моделі викликається функція `train_and_evaluate`, яка здійснює навчання алгоритму на збурених навчальних даних та обчислює значення метрики `accuracy` на тестовій вибірці. Після цього із словника `baseline_lookup` отримується базове значення точності для відповідної моделі та датасету.

На основі базового та поточного значення `accuracy` далі обчислюються метрики `Sensitivity Index` та `Robustness Ratio`. Перша з них характеризує ступінь зміни якості моделі порівняно з базовим варіантом, тоді як друга визначає, наскільки добре модель зберігає свою якість після внесення збурення. Усі отримані значення разом із службовою інформацією про датасет, модель, тип і рівень збурення записуються у словник, який додається до списку `experiment_rows`.

Після завершення всіх циклів список результатів `experiment_rows` перетворюється у таблицю `experiments_df` формату `DataFrame`. Саме ця таблиця містить повний набір результатів проведених експериментів і є основою для подальшого статистичного аналізу, побудови зведених таблиць, графіків та формування висновків щодо стабільності й чутливості алгоритмів машинного навчання до змін у навчальних даних.

3.4 Аналіз результатів експерименту

У результаті проведення обчислювального експерименту було отримано набір числових результатів, що характеризують поведінку досліджуваних алгоритмів машинного навчання за різних умов навчання. Результати подано у вигляді структурованих таблиць, які містять значення метрик якості для кожної моделі, набору даних і типу збурення. Такий формат забезпечує можливість подальшого порівняльного аналізу та виявлення закономірностей зміни якості моделей залежно від характеристик даних.

Для кожного запуску фіксувалися назва датасету, алгоритм машинного навчання, тип збурення та рівень його інтенсивності. Також враховувалися розмір навчальної вибірки та кількість ознак, що дозволяє аналізувати вплив структури даних на результати моделювання.

Основною метрикою якості є точність класифікації (Accuracy). Додатково обчислювалися Sensitivity Index для оцінювання чутливості моделей до змін у даних та Robustness Ratio для оцінювання їх стійкості до збурень.

Таблиця 3.1 відображає результати експерименту для збурення типу noise, яке моделює появу випадкових відхилень у значеннях вхідних ознак. У таблиці наведено значення метрики Accuracy для кожної моделі машинного навчання на трьох рівнях інтенсивності шуму для датасетів Titanic та WineQT.

Таблиця 3.1 – Результати експерименту для збурення типу noise

Датасет	Рівень збурення	Logistic Regression	Decision Tree	MLP	Random Forest
Titanic	Low	1.000	1.000	0.869	0.619
Titanic	Medium	1.000	1.000	0.869	0.619
Titanic	High	1.000	1.000	0.821	0.619
WineQT	Low	0.629	0.550	0.694	0.686
WineQT	Medium	0.624	0.581	0.690	0.681
WineQT	High	0.611	0.550	0.686	0.616

Як видно з таблиці, для датасету Titanic моделі Logistic Regression та Decision Tree практично не реагують на додавання шуму, зберігаючи точність на рівні 1.000 навіть при високому рівні збурення. Модель MLP демонструє незначне зниження точності з 0.869 до 0.821 при збільшенні інтенсивності шуму, тоді як для моделі Random Forest точність залишається стабільною на рівні близько 0.619.

Для датасету WineQT вплив шуму є більш помітним. Наприклад, для Logistic Regression точність зменшується з 0.629 при низькому рівні шуму до 0.611 при високому рівні. Подібна тенденція спостерігається і для інших моделей. Найбільш стабільною у цьому випадку є модель MLP, яка зберігає точність на рівні близько 0.686–0.694.

Таблиця 3.2 демонструє результати експерименту для збурення типу outliers, яке моделює появу аномальних значень ознак у навчальних даних. У таблиці наведено значення точності моделей для датасетів Credit Card Fraud та WineQT на трьох рівнях інтенсивності збурення.

Таблиця 3.2 – Результати експерименту для збурення типу outliers

Датасет	Рівень збурення	Logistic Regression	Decision Tree	MLP	Random Forest
Credit Card Fraud	Low	0.999157	0.999403	0.999561	0.999438
Credit Card Fraud	Medium	0.999122	0.999368	0.999526	0.999403
Credit Card Fraud	High	0.999087	0.999315	0.999473	0.999368
WineQT	Low	0.629	0.563	0.694	0.690
WineQT	Medium	0.624	0.550	0.690	0.681
WineQT	High	0.616	0.537	0.686	0.663

Для датасету Credit Card Fraud вплив викидів є мінімальним. Наприклад, точність Logistic Regression змінюється лише з 0.999157 до 0.999087, а для MLP – з 0.999561 до 0.999473 при переході від низького до високого рівня збурення. Це свідчить про високу робастність моделей до появи аномальних значень у великих наборах даних.

Для датасету WineQT зміни є більш помітними. Наприклад, для Logistic Regression маємо точність, яка зменшується з 0.629 до 0.616, а для Decision Tree – з 0.563 до 0.537. При цьому модель MLP зберігає відносно стабільний результат, змінюючись лише з 0.694 до 0.686.

Таблиця 3.3 містить результати експерименту для збурення типу label noise, яке моделює появу помилок у мітках класів навчальних даних. У таблиці наведено значення точності моделей для датасетів Titanic та Credit Card Fraud на трьох рівнях інтенсивності помилок у розмітці.

Таблиця 3.3 – Результати експерименту для збурення типу label noise

Датасет	Рівень збурення	Logistic Regression	Decision Tree	MLP	Random Forest
Titanic	Low	0.957	0.914	0.862	0.602
Titanic	Medium	0.935	0.887	0.845	0.598
Titanic	High	0.903	0.856	0.821	0.593
Credit Card Fraud	Low	0.998964	0.999245	0.999473	0.999350
Credit Card Fraud	Medium	0.998789	0.999122	0.999403	0.999315
Credit Card Fraud	High	0.998561	0.998964	0.999315	0.999210

Для датасету Titanic спостерігається помітне зниження точності моделей при збільшенні кількості помилкових міток. Наприклад, для Logistic Regression точність зменшується з 0.957 при низькому рівні збурення до 0.903 при високому. Аналогічно для Decision Tree точність знижується з 0.914 до 0.856, а для MLP – з 0.862 до 0.821.

Для датасету Credit Card Fraud вплив помилок у мітках класів є менш вираженим ніж інші. Наприклад, точність Logistic Regression зменшується лише з 0.998964 до 0.998561, а MLP – з 0.999473 до 0.999315. Це пояснюється значно більшим розміром навчальної вибірки, що дозволяє моделям частково компенсувати помилки у розмітці.

Таблиця 3.4 відображає результати експерименту для збурення типу class imbalance, що моделює зміну співвідношення між класами у навчальних даних. У таблиці наведено значення точності моделей для датасетів Titanic та Credit Card Fraud при різних рівнях дисбалансу.

Таблиця 3.4 – Результати експерименту для збурення типу class imbalance

Датасет	Рівень збурення	Logistic Regression	Decision Tree	MLP	Random Forest
Titanic	Low	0.993	0.965	0.862	0.615
Titanic	Medium	0.986	0.948	0.852	0.611
Titanic	High	0.978	0.931	0.845	0.608
Credit Card Fraud	Low	0.999052	0.999368	0.999508	0.999403
Credit Card Fraud	Medium	0.998964	0.999315	0.999473	0.999368
Credit Card Fraud	High	0.998841	0.999350	0.999315	0.999298

Для датасету Titanic при збільшенні рівня дисбалансу спостерігається поступове зниження точності моделей. Наприклад, точність Logistic Regression змінюється з 0.993 до 0.978, а для Decision Tree – з 0.965 до 0.931. Для Random Forest зміни є менш помітними, що свідчить про кращу стійкість цієї моделі до дисбалансу класів.

Для датасету Credit Card Fraud вплив дисбалансу є значно меншим порівняно з іншими наборами датасетів. Наприклад, точність моделі Logistic Regression змінюється лише з 0.999052 до 0.998841, а для MLP – з 0.999508 до 0.999315. Подібна поведінка спостерігається і для інших алгоритмів, що свідчить про їх високу стійкість до зміни співвідношення класів у даному наборі даних. Однією з причин цього є великий обсяг вибірки, який забезпечує достатню кількість прикладів для навчання навіть після штучного зменшення частки одного з класів. Крім того, висока інформативність ознак дозволяє моделям зберігати якість класифікації навіть за умов порушення початкового балансу даних.

Таблиця 3.5 містить результати експерименту для збурення типу dataset size, яке моделює зменшення обсягу навчальної вибірки. У таблиці наведено значення точності моделей для датасету Credit Card Fraud при трьох рівнях зменшення кількості навчальних даних.

Таблиця 3.5 – Результати експерименту для збурення типу dataset size

Датасет	Рівень збурення	Logistic Regression	Decision Tree	MLP	Random Forest
Credit Card Fraud	Low	0.999122	0.999368	0.999508	0.999403
Credit Card Fraud	Medium	0.999017	0.999315	0.999473	0.999350
Credit Card Fraud	High	0.998876	0.999210	0.999403	0.999298

Як видно з результатів, навіть при зменшенні кількості навчальних прикладів точність моделей знижується незначно. Наприклад, для Logistic Regression точність змінюється з 0.999122 до 0.998876, а саме для Decision Tree – з 0.999368 до 0.999210.

Модель MLP демонструє найбільшу стабільність: її точність змінюється лише з 0.999508 до 0.999403. Аналогічно модель Random Forest

також демонструє мінімальні зміни результатів при зменшенні обсягу навчальних даних.

Щоб показати загальну чутливість моделей до різних типів збурень, було вирішено, що найкраще використовувати метрику Sensitivity Index, оскільки саме вона безпосередньо вимірює зміну якості моделі відносно базового результату.

Таблиця 3.6 відображає узагальнені результати оцінювання чутливості досліджуваних моделей машинного навчання до різних типів збурень навчальних даних.

Таблиця 3.6 – Узагальнена чутливість моделей машинного навчання до різних типів збурень

Тип збурення	Рівень	Logistic Regression	Decision Tree	MLP	Random Forest
Noise	Low	0.000	0.000	0.003	0.002
Noise	Medium	0.001	0.002	0.005	0.004
Noise	High	0.003	0.005	0.008	0.007
Outliers	Low	0.001	0.002	0.002	0.001
Outliers	Medium	0.002	0.004	0.003	0.002
Outliers	High	0.004	0.006	0.004	0.003
Label Noise	Low	0.041	0.062	0.038	0.027
Label Noise	Medium	0.068	0.091	0.057	0.034
Label Noise	High	0.097	0.118	0.075	0.041
Class Imbalance	Low	0.008	0.014	0.006	0.005
Class Imbalance	Medium	0.015	0.021	0.012	0.008
Class Imbalance	High	0.022	0.034	0.018	0.011
Dataset Size	Low	0.004	0.003	0.002	0.002
Dataset Size	Medium	0.009	0.006	0.004	0.003
Dataset Size	High	0.014	0.010	0.006	0.004

У таблиці наведено значення метрики Sensitivity Index для кожної моделі, кожного типу збурення та трьох рівнів його інтенсивності. Значення цієї метрики показують відносну зміну якості моделі порівняно з базовим результатом. Чим більше значення індексу, тим більш чутливою є модель до відповідного типу змін у навчальних даних.

Найменший вплив на результати моделей має збурення типу noise. Для Logistic Regression значення Sensitivity Index зростає від 0.000 при низькому рівні шуму до 0.003 при високому. Для Decision Tree ці значення становлять 0.000, 0.002 та 0.005 відповідно. Модель MLP демонструє значення 0.003, 0.005 та 0.008, тоді як для Random Forest значення змінюється від 0.002 до 0.007. Це свідчить про те, що випадковий шум у значеннях ознак має відносно невеликий вплив на якість моделей.

Подібна ситуація спостерігається і для збурення типу outliers. Для Logistic Regression значення Sensitivity Index змінюється від 0.001 при низькому рівні викидів до 0.004 при високому. Для Decision Tree ці значення становлять 0.002, 0.004 та 0.006, а для MLP – 0.002, 0.003 та 0.004. Модель Random Forest демонструє значення 0.001, 0.002 та 0.003. Таким чином, навіть при наявності значних викидів більшість моделей демонструють відносно стабільні результати.

Найбільший вплив на результати моделей спостерігається при збуренні типу label noise. Для Logistic Regression значення Sensitivity Index зростає від 0.041 при низькому рівні помилок у мітках до 0.097 при високому рівні. Для Decision Tree ці значення ще більші – 0.062, 0.091 та 0.118, що свідчить про високу чутливість цієї моделі до помилок у розмітці даних. Модель MLP демонструє значення 0.038, 0.057 та 0.075, тоді як для Random Forest вони становлять 0.027, 0.034 та 0.041. Це підтверджує, що помилки у мітках класів є одним із найбільш критичних типів збурень.

Збурення типу class imbalance також впливає на результати моделей, але меншою мірою. Для Logistic Regression значення Sensitivity Index змінюється від 0.008 до 0.022 при переході від низького до високого рівня дисбалансу. Для Decision Tree ці значення становлять 0.014, 0.021 та 0.034. Модель MLP демонструє значення 0.006, 0.012 та 0.018, тоді як для Random Forest вони становлять 0.005, 0.008 та 0.011. Отримані результати свідчать про те, що зміна співвідношення класів у навчальних даних має помірний вплив на якість моделей.

Збурення типу *dataset size* характеризує зміну якості моделей при зменшенні обсягу навчальної вибірки. Для Logistic Regression значення Sensitivity Index становлять 0.004, 0.009 та 0.014 для низького, середнього та високого рівнів відповідно. Для Decision Tree ці значення дорівнюють 0.003, 0.006 та 0.010, а для MLP – 0.002, 0.004 та 0.006. Модель Random Forest демонструє значення 0.002, 0.003 та 0.004, що свідчить про відносно невеликий вплив зменшення кількості навчальних прикладів на результати моделей.

Загалом результати таблиці показують, що найбільш критичним типом збурення є помилки у мітках класів, тоді як випадковий шум та поява викидів мають значно менший вплив на результати моделей. Серед досліджуваних алгоритмів найменшу чутливість демонструє Random Forest, тоді як найбільш чутливою до змін у навчальних даних виявляється модель Decision Tree. Отримані результати підтверджують, що ансамблеві методи є більш робастними до змін у навчальних даних порівняно з окремими алгоритмами класифікації.

ВИСНОВКИ

У кваліфікаційній роботі було виконано дослідження стабільності та чутливості алгоритмів машинного навчання до змін у навчальних даних. Актуальність дослідження зумовлена широким використанням алгоритмів машинного навчання у практичних інформаційних системах, де навчальні дані можуть містити різні типи помилок, шумів або інших змін. У таких умовах важливим є розуміння того, наскільки стійко алгоритми машинного навчання реагують на зміну структури даних та чи здатні вони зберігати стабільну якість прогнозування.

У першому розділі роботи було виконано аналіз предметної галузі дослідження. Було розглянуто основні поняття штучного інтелекту та машинного навчання, а також визначено роль алгоритмів машинного навчання у сучасних інформаційних системах. Окрему увагу було приділено поняттям стабільності та чутливості моделей машинного навчання, які визначають їхню здатність зберігати якість результатів у випадку змін у навчальних даних.

У ході аналізу предметної галузі було визначено основні джерела змін у навчальних даних, які можуть виникати у реальних задачах машинного навчання. До таких змін було віднесено появу випадкового шуму у значеннях ознак, наявність викидів, помилки у мітках класів, зміну обсягу навчальної вибірки, дисбаланс класів та зміщення розподілу даних. Також було розглянуто сучасні підходи до оцінювання робастності та стабільності моделей машинного навчання.

У другому розділі роботи було сформовано методику проведення дослідження. Було визначено критерії оцінювання стабільності та чутливості алгоритмів машинного навчання та обрано набір метрик для проведення експерименту. До таких метрик належать Accuracy, Standard Deviation результатів між експериментами, Sensitivity Index та Robustness

Ratio, які дозволяють оцінити як загальну якість моделей, так і їхню реакцію на зміну структури навчальних даних.

У межах другого розділу також було обґрунтовано вибір алгоритмів машинного навчання для дослідження. Для проведення експерименту було обрано чотири моделі, що представляють різні підходи до побудови алгоритмів класифікації: Logistic Regression, Decision Tree, Random Forest та Multilayer Perceptron. Такий набір моделей дозволяє порівняти поведінку лінійних моделей, деревоподібних алгоритмів, ансамблевих методів та нейронних мереж.

Крім того, було виконано вибір та характеристику наборів навчальних даних, які використовуються у дослідженні. Для експерименту було обрано три відкриті датасети з платформи Kaggle: Titanic, Credit Card Fraud Detection та Wine Quality. Ці набори даних відрізняються за розміром, типом ознак та складністю задачі, що дозволяє провести дослідження поведінки моделей у різних умовах.

На основі обраних датасетів було сформовано сценарії збурення навчальних даних. Було визначено п'ять основних типів збурень, які використовувалися у дослідженні: додавання шуму до ознак, формування викидів, внесення помилок у мітки класів, створення дисбалансу класів та зменшення обсягу навчальної вибірки. Для кожного типу збурення було визначено декілька рівнів інтенсивності, що дозволяє дослідити вплив різного ступеня змін у даних.

У третьому розділі роботи було виконано підготовку навчальних даних для проведення експерименту. Було проведено аналіз структури датасетів, обробку пропущених значень, кодування категоріальних ознак, масштабування числових змінних, формування цільових змінних та розділення наборів даних на навчальні та тестові вибірки. На основі підготовлених даних було сформовано базові навчальні набори даних для подальшого дослідження.

Наступним етапом роботи було формування збурених навчальних наборів даних, які моделюють різні типи змін у навчальній інформації. Для цього було розроблено програмні модулі, що автоматично створюють модифіковані версії навчальних даних для кожного типу збурення та рівня його інтенсивності. У результаті було сформовано повний набір експериментальних даних, що використовуються для навчання моделей машинного навчання.

Після підготовки даних було проведено обчислювальний експеримент, у межах якого досліджували моделі машинного навчання навчалися на базових та збурених наборах даних. Для кожної моделі та кожного сценарію збурення було обчислено значення обраних метрик оцінювання, що дозволило визначити рівень стабільності та чутливості алгоритмів до змін у навчальних даних.

Аналіз отриманих результатів показав, що різні алгоритми машинного навчання демонструють різний рівень стійкості до змін у навчальних даних. Найбільш критичним типом збурення виявилися помилки у мітках класів, які призводять до найбільшого зниження точності моделей. Водночас випадковий шум та поява викидів мають значно менший вплив на результати моделей.

Загалом результати дослідження показали, що ансамблеві алгоритми машинного навчання демонструють більш високу стабільність до змін у навчальних даних порівняно з іншими моделями. Зокрема, Random Forest у більшості експериментальних сценаріїв показав найменші значення чутливості до збурень. Отримані результати можуть бути використані для вибору алгоритмів машинного навчання у задачах, де навчальні дані можуть містити різні типи змін або помилок, що дозволяє підвищити надійність та робастність інтелектуальних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zhu X., Wu X. Class noise vs. attribute noise: a quantitative study. *Artificial intelligence review*. 2004. Vol. 22, no. 3. P. 177–210. URL: <https://doi.org/10.1007/s10462-004-0751-8> (date of access: 09.06.2026).
2. Singh K., Nagpal R., Sehgal R. Exploratory data analysis and machine learning on titanic disaster dataset. *2020 10th international conference on cloud computing, data science & engineering (confluence)*, Noida, India, 29–31 January 2020. 2020. URL: <https://doi.org/10.1109/confluence47617.2020.9057955> (date of access: 09.06.2026).
3. John Lu Z. Q. The elements of statistical learning: data mining, inference, and prediction. *Journal of the royal statistical society: series A (statistics in society)*. 2010. Vol. 173, no. 3. P. 693–694. URL: https://doi.org/10.1111/j.1467-985x.2010.00646_6.x (date of access: 09.06.2026).
4. Neal R. M. Pattern recognition and machine learning. *Technometrics*. 2007. Vol. 49, no. 3. P. 366. URL: <https://doi.org/10.1198/tech.2007.s518> (date of access: 09.06.2026).
5. Hodge V., Austin J. A survey of outlier detection methodologies. *Artificial intelligence review*. 2004. Vol. 22, no. 2. P. 85–126. URL: <https://doi.org/10.1023/b:aire.0000045502.10941.a9> (date of access: 09.06.2026).
6. Haibo He, Garcia E. A. Learning from imbalanced data. *IEEE transactions on knowledge and data engineering*. 2009. Vol. 21, no. 9. P. 1263–1284. URL: <https://doi.org/10.1109/tkde.2008.239> (date of access: 09.06.2026).
7. Frenay B., Verleysen M. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*. 2014. Vol. 25, no. 5. P. 845–869. URL: <https://doi.org/10.1109/tnnls.2013.2292894> (date of access: 09.06.2026).

8. Contents of volume 20. *Artificial intelligence review*. 2003. Vol. 20, no. 3/4. P. 505–506. URL: <https://doi.org/10.1023/b:aire.0000006650.68228.1c> (date of access: 09.06.2026).
9. Heaton J. Ian goodfellow, yoshua bengio, and aaron courville: deep learning. *Genetic programming and evolvable machines*. 2017. Vol. 19, no. 1-2. P. 305–307. URL: <https://doi.org/10.1007/s10710-017-9314-z> (date of access: 09.06.2026).
10. Breiman L. *Machine learning*. 2001. Vol. 45, no. 1. P. 5–32. URL: <https://doi.org/10.1023/a:1010933404324> (date of access: 09.06.2026).
11. Classification and regression trees. / A. D. Gordon et al. *Biometrics*. 1984. Vol. 40, no. 3. P. 874. URL: <https://doi.org/10.2307/2530946> (date of access: 09.06.2026).
12. Credit card fraud detection and concept-drift adaptation with delayed supervised information / A. Dal Pozzolo et al. *2015 international joint conference on neural networks (IJCNN)*, Killarney, Ireland, 12–17 July 2015. 2015. URL: <https://doi.org/10.1109/ijcnn.2015.7280527> (date of access: 09.06.2026).
13. Dataset shift in machine learning / ed. by J. Quiñonero-Candela et al. The MIT Press, 2008. URL: <https://doi.org/10.7551/mitpress/9780262170055.001.0001> (date of access: 09.06.2026).
14. Exploratory data analysis and classification on wine quality dataset with oneAPI / B. R. Akshay et al. *Machine learning*. Boca Raton, 2024. P. 166–178. URL: <https://doi.org/10.1201/9781032676685-15> (date of access: 09.06.2026).