Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему *Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання*

Галузь знань *12 – Інформаційні технології*
Шифр і назва галузі знань

Спеціальність *122 – Комп'ютерні науки*
Шифр і назва спеціальності

Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Виконав: *студент 2 курсу, група КНм-23-1* (Курс, група виконавця) _____ (Підпис) *Микола ШТОЙКО* (Ім'я, прізвище)

Керівник: *док. філ., ст. викл. каф. КН* (Науковий ступінь, посада) _____ (Підпис) *Павло РАДЮК* (Ім'я, прізвище)

Нормоконтроль: *к.т.н., доцент кафедри КН* (Науковий ступінь, посада) _____ (Підпис) *Руслан БАГРІЙ* (Ім'я, прізвище)

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор _____ (Підпис) *Олександр БАРМАК* (Ім'я, прізвище)

18 грудня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

(підпис)
д.т.н., професор Олександр БАРМАК
«*02*« *вересня* *2024* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання»

2. Завдання видано студенту *Миколі ШТОЙКО*
Ім'я, прізвище

3. Керівник роботи *старший викладач кафедри КН Павло РАДЮК*
Ім'я, прізвище

4. Затверджені наказом університету від «*26*« *серпня* *2024* р. № *60* .

5. Дата видачі завдання студенту: «*02*« *вересня* *2024* р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи магістра є підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання. Досягнення мети роботи передбачає виконання таких задач: провести аналіз моделей, методів та засобів пояснювального штучного інтелекту;

спроєктувати модель подання результатів задач класифікації за моделями глибокого навчання через ознаки моделей машинного навчання; спроєктувати метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання; виконати програмну реалізацію та провести експериментальне тестування спроєктованого методу пояснення результатів задач класифікації.

Реферат

Кваліфікаційна робота магістра присвячена дослідженню методів для підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

Актуальність теми. Наразі усі галузі людської діяльності, що використовують методи та засоби глибокого навчання потребують нових підходів до пояснення результатів задач класифікації, які виконуються моделями глибокого навчання. Ці моделі, хоча й демонструють високу ефективність, часто є “чорними скриньками”, рішення яких важко інтерпретувати. Це створює ризики у критичних галузях, як от медицина та фінанси, де необхідна прозорість та обґрунтованість прийнятих рішень. Отже, проєктування методів пояснення є ключовим завданням для підвищення довіри до цих моделей та їхнього безпечного застосування.

Об’єкт дослідження – процес пояснення результатів задач класифікації за моделями глибокого навчання.

Предмет дослідження – методи, засоби та алгоритми глибокого та машинного навчання у задачах класифікації.

Мета і задачі роботи – підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз моделей, методів та засобів пояснювального штучного інтелекту.

2. Спроєктувати модель подання результатів задач класифікації за моделями глибокого навчання через ознаки моделей машинного навчання.

3. Спроєктувати метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

4. Виконати програмну реалізацію та провести експериментальне тестування спроєктованого методу пояснення результатів задач класифікації.

Методи дослідження. У роботі використано методи комп’ютерного зору та глибокого навчання для аналізу та обробки даних, зокрема для сегментування та

генерування зображень. Для проєктування методу пояснення результатів класифікації застосовано підхід з використанням перехідної матриці між ознаками моделей глибокого навчання та моделями машинного навчання, а також методи оцінки важливості ознак, такі як SHAP і LIME. Програмна реалізація методу здійснювалася за допомогою бібліотек TensorFlow, PyTorch та scikit-learn. Успішність спроєктованого методу перевірялася за допомогою експериментального тестування на еталонних наборах даних з використанням різноманітних метрик.

Наукова новизна одержаних результатів. Удосконалено метод пояснення результатів задач класифікації за моделями глибокого навчання, який відрізняється від наявних підходів інтеграцією перехідної матриці між ознаками моделей глибокого навчання та моделями машинного навчання, що дає можливість перетворити складні ознаки глибокого навчання на більш зрозумілі ознаки машинного навчання для інтерпретації результатів, що забезпечує прозорість процесу прийняття рішень.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук АПКН-2024”, м. Хмельницький, ХНУ, 15–16 листопада 2024 р. (Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання / М. С. Штойко та ін. Актуальні проблеми комп’ютерних наук АПКН-2024 : матеріали XVI Всеукр. науково-практ. конф., м. Хмельницький, 15–16 листоп. 2024 р. Хмельницький, 2024. С. 553–555. URL: https://elar.khmnu.edu.ua/handle/123456789/17151)

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 98 сторінок, з поміж яких 86 сторінок основного тексту та 12 сторінок додатків. У роботі наведено 31 рисунок та 7 таблиць.

Ключові слова: пояснення результатів, глибоке навчання, машинне навчання, класифікація, інтерпретація моделей, SHAP, LIME.

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
AI	Artificial Intelligence
AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DL	Deep Learning
DNN	Deep Neural Network
DSS	Decision Support System
FM	Frequency Modulation
FPR	False Positive Rate
Grad-CAM	Gradient-weighted Class Activation Mapping
GRU	Gated Recurrent Unit
HTML	HyperText Markup Language
IQR	Interquartile Range
JSON	JavaScript Object Notation
LIME	Local Interpretable Model-agnostic Explanations
LRP	Layer-wise Relevance Propagation
LSTM	Long Short-Term Memory
ML	Machine Learning
MM	Methodological Model
MNIST	Modified National Institute of Standards and Technology
ONNX	Open Neural Network Exchange
PDP	Partial Dependence Plots
PDF	Portable Document Format

RNN	Recurrent Neural Networks
RoBERTa	Robustly Optimized BERT Pretraining Approach
SDK	Software Development Kit
SHAP	SHapley Additive exPlanations
SQL	Structured Query Language
TPR	True Positive Rate
VA	Visual Analytics
XAI	Explainable Artificial Intelligence
ШІ	Штучний Інтелект

Вступ

Актуальність теми. Наразі усі галузі людської діяльності, що використовують методи та засоби глибокого навчання потребують нових підходів до пояснення результатів задач класифікації, які виконуються моделями глибокого навчання. Ці моделі, хоча й демонструють високу ефективність, часто є “чорними скриньками”, рішення яких важко інтерпретувати. Це створює ризики у критичних галузях, як от медицина та фінанси, де необхідна прозорість та обґрунтованість прийнятих рішень. Отже, проєктування методів пояснення є ключовим завданням для підвищення довіри до цих моделей та їхнього безпечного застосування.

Об’єкт дослідження – процес пояснення результатів задач класифікації за моделями глибокого навчання.

Предмет дослідження – методи, засоби та алгоритми глибокого та машинного навчання у задачах класифікації.

Мета і задачі роботи – підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз моделей, методів та засобів пояснювального штучного інтелекту.

2. Спроєктувати модель подання результатів задач класифікації за моделями глибокого навчання через ознаки моделей машинного навчання.

3. Спроєктувати метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

4. Виконати програмну реалізацію та провести експериментальне тестування спроєктованого методу пояснення результатів задач класифікації.

Методи дослідження. У роботі використано методи комп’ютерного зору та глибокого навчання для аналізу та обробки даних, зокрема для сегментування та генерування зображень. Для проєктування методу пояснення результатів класифікації застосовано підхід з використанням перехідної матриці між ознаками моделей глибокого навчання та моделями машинного навчання, а також методи

оцінки важливості ознак, такі як SHAP і LIME. Програмна реалізація методу здійснювалася за допомогою бібліотек TensorFlow, PyTorch та scikit-learn. Успішність спроєктованого методу перевірялася за допомогою експериментального тестування на еталонних наборах даних з використанням різноманітних метрик.

Наукова новизна одержаних результатів. Удосконалено метод пояснення результатів задач класифікації за моделями глибокого навчання, який відрізняється від наявних підходів інтеграцією перехідної матриці між ознаками моделей глибокого навчання та моделями машинного навчання, що дає можливість перетворити складні ознаки глибокого навчання на більш зрозумілі ознаки машинного навчання для інтерпретації результатів, що забезпечує прозорість процесу прийняття рішень.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на науково-практичній конференції – XVI Всеукраїнська науково-практична конференція “Актуальні проблеми комп’ютерних наук АПКН-2024”, м. Хмельницький, ХНУ, 15–16 листопада 2024 р. [1].

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 40 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 98 сторінок, з поміж яких 86 сторінок основного тексту та 12 сторінок додатків. У роботі наведено 31 рисунок та 7 таблиць.

Ключові слова: пояснення результатів, глибоке навчання, машинне навчання, класифікація, інтерпретація моделей, SHAP, LIME.

РОЗДІЛ 1 Огляд проблемної області та постановка задачі дослідження

1.1 Аналіз проблем, які виникають під час пояснення результатів задач класифікації за моделями глибокого навчання

Моделі глибокого навчання вже довгий час є ключовою технологією в багатьох сферах, завдяки їх здатності розв'язувати складні задачі класифікації. У таких галузях, як розпізнавання образів, обробка природної мови, комп'ютерний зір, системи рекомендацій, медична діагностика, та інші, глибокі нейронні мережі демонструють значно кращі результати порівняно з традиційними методами машинного навчання.

Основою успіху глибокого навчання є здатність цих моделей навчатися на величезних обсягах даних і виділяти складні взаємозв'язки між вхідними ознаками. Вони можуть автоматично будувати подання (відображення ознак) даних на різних рівнях абстракції, що робить їх дуже потужними інструментами для вирішення задач класифікації. Проте така потужність супроводжується й рядом проблем, зокрема, коли мова йде про інтерпретацію рішень, які приймають ці моделі [1].

Моделі глибокого навчання часто сприймаються як «чорні ящики», тобто системи, внутрішні механізми яких є непрозорими або важко доступними для розуміння. Це означає, що хоча ці моделі можуть робити дуже точні прогнози або класифікації, вони не можуть пояснити, чому вони прийшли до певного результату. Наприклад, модель може точно класифікувати зображення як «кішка» або «собака», але важко пояснити, які саме ознаки або комбінації ознак вона використовувала для цього рішення [2]. Це стає особливо проблематичним у чутливих галузях, таких як медицина, де фахівці потребують не лише точного результату, а й детального пояснення, на основі яких характеристик даних модель зробила певний висновок.

Проблема відсутності прозорості та пояснюваності рішень моделей глибокого навчання привертає дедалі більше уваги з боку дослідників і практиків. Без пояснення того, як модель приймає рішення, виникає ризик недовіри до моделі з боку користувачів, особливо у критичних областях, таких як фінансовий сектор, медичні рішення або автономне водіння. Також існують етичні питання щодо справедливості

та відсутності упереджень у рішеннях моделей, що є ще однією причиною для розробки інтерпретованих моделей або створення методів, що пояснюють їхнє рішення [3].

Отже, потреба в пояснюваності моделей глибокого навчання є важливим напрямком сучасних досліджень в області машинного навчання. Це породжує нову галузь досліджень – пояснювальне машинне навчання (XAI – Explainable AI), яка націлена на розробку методів і технологій, що дають змогу зробити процес прийняття рішень більш прозорим та зрозумілим для людини [4].

Глибоке навчання є підгалуззю машинного навчання, яка ґрунтується на використанні штучних нейронних мереж з великою кількістю шарів – саме тому такі мережі називають «глибокими» (рисунок 1.1).

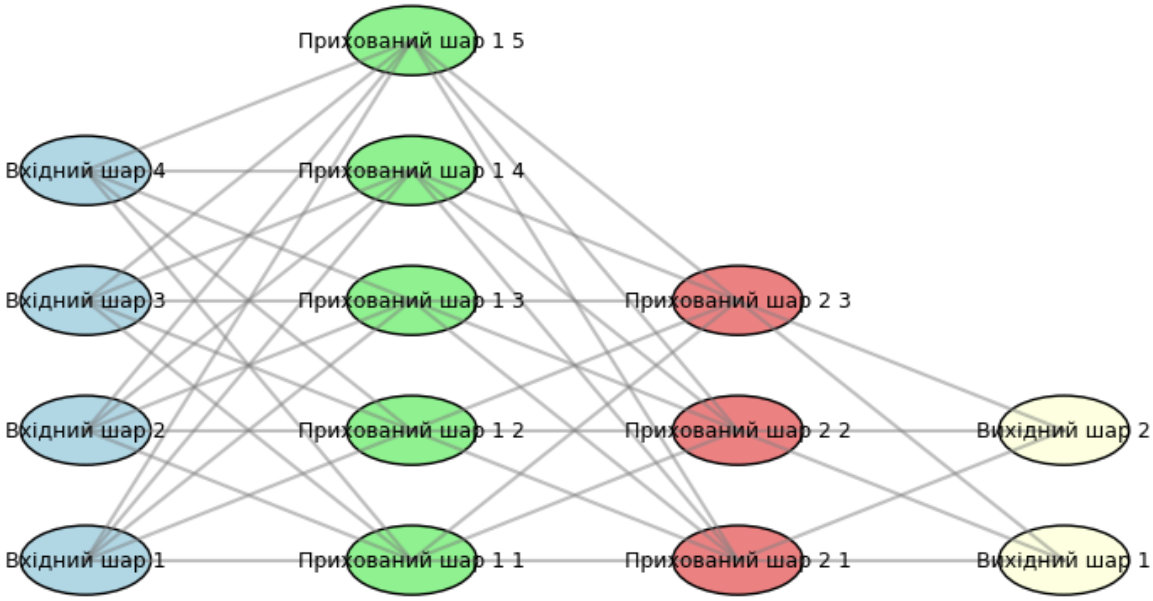


Рисунок 1.1 – Архітектура глибокої нейронної мережі [5]

Ці моделі значно перевершують традиційні методи машинного навчання в багатьох завданнях, що пов’язані з класифікацією. Основою їх успіху є здатність автоматично витягувати ознаки з даних, що особливо важливо при роботі з неструктурованими або високорозмірними даними, як-от зображення, аудіо чи тексти.

Глибокі нейронні мережі складаються з великої кількості шарів нейронів, де кожен шар будується на основі вихідних даних попереднього шару. Перші шари зазвичай вивчають низькорівневі ознаки, такі як контури на зображеннях або найпростіші патерни в тексті. У міру того як мережа поглиблюється, шари починають витягувати більш складні ознаки та абстракції. Наприклад, у задачах класифікації зображень більш глибокі шари можуть розпізнавати такі складні об’єкти, як автомобілі, обличчя або тварини [6].

Мережі глибокого навчання здатні досягати високої точності завдяки своїй здатності самостійно витягувати релевантні ознаки з даних і адаптувати їх до конкретних задач класифікації. Наприклад, у сфері комп’ютерного зору CNN стали стандартом для класифікації зображень, а у природній мові трансформери значно покращили обробку текстів.

У задачах класифікації зображень CNN показали виняткові результати завдяки своїй здатності зберігати просторову структуру даних. Ці моделі використовують операції згортки та субдискретизації для зменшення розмірності вхідних даних і виділення ключових характеристик, таких як краю або текстури. CNN виявилися особливо результативними у таких задачах, як розпізнавання обличь, об’єктів або навіть медичних зображень для діагностики хвороб [6].

Для задач класифікації тексту популярні моделі на основі RNN і трансформерів, які добре працюють з послідовними даними. Трансформери, такі як BERT і GPT, дозволили зробити значний прорив у багатьох задачах оброблення природної мови, таких як класифікація тексту, аналіз тональності та машинний переклад.

В аудіоаналізі глибокі моделі, зокрема згорткові мережі в поєднанні з рекурентними мережами або трансформерами, використовуються для задач розпізнавання мови, класифікації звуків або аудіосигналів, таких як класифікація музичних жанрів або виявлення аномалій у звуці [6].

Хоча глибокі нейронні мережі є надзвичайно потужними для вирішення класифікаційних задач, вони також мають певні недоліки. Одним з найбільш значущих викликів є складність інтерпретації їхніх рішень. Глибокі моделі містять

мільйони або навіть мільярди параметрів, що робить їх надзвичайно складними для розуміння. У таких моделях важко визначити, які саме ознаки або комбінації ознак були використані для прийняття конкретного рішення. Це створює проблему непрозорості («чорний ящик»), коли користувачі не можуть точно пояснити, чому модель класифікувала певний об’єкт або сигнал певним чином.

Інтерпретація таких моделей є важливою в різних областях. Наприклад, у медицині важливо не лише отримати точний прогноз, а й розуміти, на які фактори звертає увагу модель під час постановки діагнозу. Аналогічно, в юридичних або фінансових рішеннях важлива прозорість, щоб уникнути упереджених або несправедливих рішень.

Моделі глибокого навчання досягли значних успіхів у вирішенні складних задач, таких як класифікація зображень, обробка природної мови та розпізнавання звуків. Однак, попри їхню високу точність, однією з найбільш суттєвих проблем залишається інтерпретованість результатів, тобто здатність зрозуміти, як і чому модель прийняла те чи інше рішення.

Основною причиною низької інтерпретованості моделей глибокого навчання є їхня складна багатошарова структура та велика кількість параметрів. Глибокі нейронні мережі, зокрема, можуть містити тисячі або навіть мільйони параметрів, що робить практично неможливим для людини простежити, як саме модель обробляє вхідні дані та приймає рішення. Крім того, багатошарові архітектури, такі як CNN або RNN, здійснюють послідовну трансформацію ознак, які стають дедалі складнішими на кожному наступному шарі. Це робить важким зрозуміти, які саме фактори вплинули на кінцевий прогноз [7].

Традиційні методи машинного навчання, такі як рішення дерев (decision trees), лінійні та логістичні регресії, забезпечують значно більшу прозорість. Вони зазвичай мають меншу кількість параметрів, а їхня структура дає змогу легко відстежувати шлях до прийняття рішення [8]. Наприклад, в рішенні дерева кожен вузол подає певну ознаку або умову, і шлях до кінцевого рішення легко інтерпретується як послідовність таких умов.

Проте глибокі моделі значно складніші. Хоча вони й перевершують традиційні підходи за точністю, вони перетворюють вхідні дані на високорівневі ознаки, які складно інтерпретувати людині. Крім того, ці ознаки часто є нелінійними і багатовимірними, що ще більше ускладнює розуміння того, як модель прийшла до певного висновку.

1.2 Аналіз сучасних методів та підходів до вирішення проблеми пояснення результатів задач класифікації

Аналіз відомих методів пояснення результатів задач класифікації за моделями глибокого навчання фокусується на різних підходах, які допомагають зрозуміти, як і чому ці моделі приймають певні рішення. Ці методи можна розділити на дві основні категорії: модель-агностичні (працюють з будь-якою моделлю) і модель-специфічні (розроблені для конкретних архітектур). Ось аналіз найбільш відомих з них:

1. LIME (Local Interpretable Model-agnostic Explanations). LIME – це метод інтерпретованості, який пояснює індивідуальні передбачення «чорної скриньки» моделі машинного навчання. Мета LIME – розкрити, які особливості вхідних даних вплинули на конкретне рішення моделі, навіть якщо сама модель є складною або непрозорою, наприклад, як у випадку з глибокими нейронними мережами [9].

Як працює LIME. Локальна апроксимація: Основна ідея LIME полягає в тому, що складну модель можна пояснити шляхом локального аналізу. Це означає, що замість того, щоб намагатися зрозуміти всю модель в цілому, LIME фокусується на поясненні передбачення для конкретного зразка (вхідного прикладу). Щоб зробити це, LIME створює штучні зразки даних у околі цього конкретного зразка. Він дещо змінює вхідні дані, щоб зрозуміти, як ці зміни впливають на передбачення моделі. Це дає змогу йому виявити, які особливості (ознаки) найбільше впливають на рішення моделі для даного прикладу [10].

Побудова спрощеної моделі: LIME використовує ці змінені зразки та їх передбачення, щоб побудувати просту, інтерпретовану модель, наприклад, лінійну регресію або дерева рішень. Ця модель призначена для наближення поведінки

складної моделі в околі конкретного зразка (рисунок 1.2). Отримана спрощена модель має бути настільки простою, щоб людина могла легко зрозуміти, як вона приймає рішення [11].

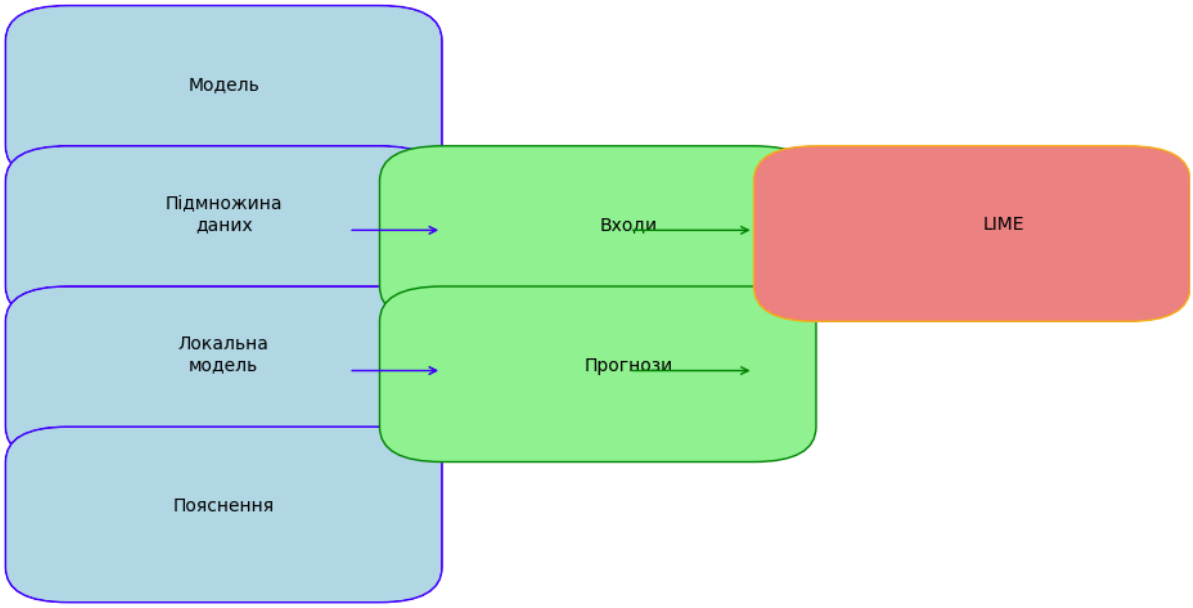


Рисунок 1.2 – Спрощена модель LIME [12].

Пояснення передбачення: Результатом роботи LIME є інтерпретована модель, яка пояснює, як вхідні ознаки впливають на передбачення складної моделі. Наприклад, для класифікації зображень LIME може вказати, які сегменти зображення були найважливішими для рішення моделі. Для текстових даних LIME може виділити слова або фрази, які вплинули на класифікацію тексту [12].

2. SHAP (SHapley Additive exPlanations). SHAP – це потужний метод пояснення передбачень моделей машинного навчання, заснований на концепції значень Шеплі з теорії кооперативних ігор. SHAP намагається пояснити передбачення, визначаючи внесок кожної ознаки у кінцевий результат моделі. Цей метод забезпечує єдиний підхід до оцінювання важливості ознак, дозволяючи зрозуміти як локальні пояснення (для окремих передбачень), так і глобальні характеристики моделі.

Як працює SHAP. Значення Шеплі: У теорії ігор значення Шеплі визначає, як розподілити «виграш» (в даному випадку – передбачення моделі) серед гравців

(ознак) Отже, щоб врахувати їх індивідуальний внесок. Для кожного гравця обчислюється середній внесок у всіх можливих коаліціях (підмножин гравців) [13]. У контексті машинного навчання кожна ознака розглядається як «гравець», а передбачення моделі – як «виграш». Значення Шеплі для ознаки визначає, наскільки вона впливає на передбачення, беручи до уваги всі можливі комбінації ознак [14].

Розрахунок SHAP-значень: SHAP обчислює внесок кожної ознаки в передбачення, зважаючи на всі можливі комбінації ознак. Цей підхід гарантує, що оцінка внеску є справедливою та послідовною. Для кожного зразка даних SHAP розраховує, на скільки передбачення моделі змінюється при додаванні або виключенні кожної ознаки. SHAP-значення можуть бути позитивними або негативними, вказуючи на те, наскільки кожна ознака збільшує або зменшує передбачення моделі порівняно з базовим передбаченням (наприклад, середнім значенням) [13].

Локальні та глобальні пояснення: Локальні пояснення: SHAP-значення дають змогу пояснити окремі передбачення, показуючи, як кожна ознака впливає на результат для конкретного зразка. Глобальні пояснення: SHAP також дає змогу отримати глобальні пояснення роботи моделі, наприклад, виявити загальну важливість ознак, середні впливи на всі передбачення та взаємодії між ознаками [15].

3. Gradient-based Methods. Методи, засновані на градієнтах, є одним із підходів до пояснення та інтерпретації передбачень моделей глибокого навчання, зокрема нейронних мереж. Вони використовують інформацію про градієнти, які обчислюються під час переднього проходу моделі, щоб визначити, як зміни вхідних даних впливають на вихід. Ці методи широко застосовуються для моделей комп’ютерного зору, де пояснення у вигляді «теплових карт» можуть бути візуально інтуїтивно зрозумілими.

Основні методи. Saliency Maps (Карти важливості): Суть: Цей метод визначає, які частини вхідних даних (наприклад, пікселі на зображенні) найбільше впливають на результат моделі. Для цього обчислюються градієнти виходу моделі (наприклад, ймовірність певного класу) відносно вхідних даних [16].

Як це працює. Вихідне значення (наприклад, ймовірність передбаченого класу) диференціюється по кожному пікселю зображення. Отримані градієнти показують, наскільки зміна кожного пікселя впливає на передбачення. Чим більший градієнт для конкретного пікселя, тим більший його вплив на результат.

Результат: Карта важливості, яка візуально показує області зображення, що найбільше впливають на рішення моделі. Вона дає змогу зрозуміти, які особливості зображення модель використовує для класифікації [17].

Grad-CAM (Gradient-weighted Class Activation Mapping). Суть: Grad-CAM – це більш вдосконалений метод, призначений для моделей глибокого навчання, які використовують CNN. Він генерує «теплові карти» для зображень, які показують, які області зображення були найважливішими для передбачення конкретного класу [18].

Як це працює. Grad-CAM використовує градієнти, обчислені на виході останнього згорткового шару, щоб визначити важливість кожного каналу в цьому шарі для передбаченого класу. Ці ваги використовуються для комбінації активацій згорткового шару, що дає теплову карту, яка показує важливі області зображення.

Результат. Теплова карта накладається на оригінальне зображення, щоб показати, які його частини були найбільш значущими для моделі при прийнятті рішення. Ці карти легко інтерпретуються та дають зрозуміти, які характеристики зображення (наприклад, певні об’єкти або контури) впливають на результат [19].

4. Layer-wise Relevance Propagation (LRP). LRP – це метод пояснення моделей глибокого навчання, який розподіляє «важливість» передбачення моделі через шари нейронної мережі назад до вхідних даних [20]. Основна ідея LRP полягає в тому, щоб виявити, які частини вхідних даних (наприклад, пікселі зображення або ознаки) найбільше вплинули на кінцевий результат, дозволяючи таким чином зрозуміти рішення «чорної скриньки» моделі [21].

Як працює LRP. Розподіл важливості: LRP починає з розподілу «важливості» передбачення (наприклад, виходу нейронної мережі) через кожен шар нейронної мережі до вхідного рівня. «Важливість» визначається тим, наскільки кожен нейрон і кожна ознака сприяють передбаченню [22].

Зворотне розповсюдження: Подібно до зворотного поширення градієнта, LRP виконує розповсюдження, але замість градієнтів він поширює значення «важливості» через кожен шар. На кожному шарі виконується специфічне правило розподілу важливості, яке враховує активності нейронів та їх ваги. Важливість кожного нейрона в поточному шарі пропорційно розподіляється серед нейронів попереднього шару, ґрунтуючись на внеску цих нейронів у його активацію [23].

Отримання карти важливості. Після завершення зворотного розповсюдження до вхідного шару, ми отримуємо карту важливості, яка показує, які частини вхідних даних найбільше вплинули на кінцевий результат моделі. Наприклад, для зображень LRP може створити теплову карту, яка вказує, які пікселі були найбільш значущими для передбачення [24].

5. Feature Importance (Важливість ознак). Feature Importance – це метод оцінювання впливу кожної ознаки (вхідної характеристики) на результати моделі машинного або глибокого навчання. Важливість ознак допомагає зрозуміти, які ознаки є найбільш значущими для передбачення і наскільки вони впливають на результати моделі. Цей підхід може бути застосований до різних типів моделей, включаючи дерева рішень, ансамблі дерев (наприклад, Random Forest), лінійні моделі та навіть нейронні мережі [25].

Як працює Feature Importance. Метод випадкового відключення ознак (Permutation Importance): Суть: Оцінка впливу кожної ознаки на результати моделі шляхом її випадкового перемішування. Як це працює: Спочатку модель навчається на всіх доступних ознаках. Потім для кожної ознаки виконується перемішування її значень, що порушує зв’язок між цією ознакою та цільовою змінною. Після цього оцінюється, наскільки погіршується якість моделі (наприклад, точність). Якщо якість значно знижується, це означає, що ознака є важливою для моделі. Результат: Важливість ознаки визначається як зміна в показнику якості моделі після перемішування цієї ознаки [26].

Аналіз внеску (Coefficient Importance): Суть: Оцінка впливу ознак на результати моделі шляхом аналізу їхніх коефіцієнтів (наприклад, у лінійних моделях). Як це працює: Для моделей, що мають коефіцієнти для кожної ознаки (наприклад,

лінійна регресія, логістична регресія), важливість ознаки визначається величиною абсолютного значення її коефіцієнта. Велике значення коефіцієнта вказує на високу важливість ознаки. Результат: Ознаки з більшими абсолютними значеннями коефіцієнтів вважаються більш важливими для моделі [27].

Вбудовані методи (Embedded Methods). Деякі моделі, такі як дерева рішень або Random Forest, мають вбудовані механізми для оцінювання важливості ознак. Наприклад, у випадку дерев рішень важливість ознаки може визначатися на основі того, наскільки вона зменшує невизначеність (наприклад, ентропію) при розбитті вузлів [28].

6. Surrogate Models (Моделі-замінники). Surrogate Models (Моделі-замінники) – це прості та інтерпретовані моделі, які використовуються для наближення та пояснення поведінки складних моделей, таких як глибокі нейронні мережі або інші «чорні скриньки». Модель-замінник створюється на основі вхідних даних та результатів складної моделі, щоб відтворити її поведінку в певному діапазоні або області [29].

Як працює метод Surrogate Models. Навчання простої моделі: Спочатку складна модель (наприклад, глибока нейронна мережа) використовується для прогнозування результатів на наборі даних. Потім на основі вхідних даних та прогнозів складної моделі будується спрощена, інтерпретована модель (наприклад, лінійна регресія, дерево рішень). Ця спрощена модель має на меті відтворити поведінку складної моделі якомога точніше, але з меншою складністю та кращою інтерпретованістю [30].

Апроксимація поведінки складної моделі: Модель-замінник не прагне бути настільки ж точною, як вихідна складна модель. Натомість вона надає приблизне уявлення про те, як працює складна модель, використовуючи прості правила чи закономірності. Вона може використовуватися для розуміння глобальної поведінки складної моделі або для надання локальних пояснень для конкретних передбачень [31].

Глобальні та локальні моделі-замінники. Глобальна модель-замінник: Відтворює поведінку складної моделі в цілому, показуючи загальні тенденції та

взаємозв’язки між ознаками та результатом. Локальна модель-замінник: Зосереджується на поясненні поведінки складної моделі в околі конкретного прикладу (наприклад, метод LIME). Локальні моделі-замінники краще підходять для пояснення індивідуальних передбачень [32].

У таблиці 1.1 наведено переваги та недоліки основних методів пояснення результатів глибокого навчання:

Таблиця 1.1 – Переваг та недоліків методів.

Метод	Переваги	Недоліки
1. LIME	– модель-агностичний метод, підходить для будь-яких типів моделей; – локальна інтерпретація передбачень; – висока гнучкість і простота реалізації.	– може давати неточні результати, особливо на нелінійних моделях; – не підходить для глобальної інтерпретації; – високі обчислювальні витрати.
2. SHAP	– точний розподіл внеску кожної ознаки; – підходить для як глобальної, так і локальної інтерпретації; – модель-агностичний метод.	– високі обчислювальні витрати, особливо для великих моделей; – може бути складним для реалізації на великих наборах даних.
3. Gradient-based Methods	– ефективний для моделей з зображеннями (CNN); – швидкий розрахунок градієнтів; – прямий доступ до інформації через градієнти.	– залежність від структури моделі та функцій активації; – може бути не інтуїтивним для користувачів; – не підходить для всіх типів даних.

4. LRP	– забезпечує глибокий аналіз, включаючи активації на різних шарах; – візуалізація пояснень для нейронних мереж; – добре підходить для моделювання зображень.	– складна реалізація; – потребує налаштування для різних моделей; – не завжди інтуїтивно зрозуміло, як саме обчислюється релевантність.
5. Feature Importance (Важливість ознак)	– простота та інтуїтивна зрозумілість; – швидка оцінка впливу ознак на результати моделі; – підходить для багатьох типів моделей.	– може не враховувати складні взаємодії між ознаками; – не забезпечує локальних пояснень для окремих передбачень.
6. Surrogate Models (Моделі-замінники)	– легко інтерпретуються (наприклад, дерева рішень); – підходять для моделювання як глобальної, так і локальної поведінки; – модель-агностичний підхід.	– може втратити частину інформації з вихідної складної моделі; – низька точність, особливо для нелінійних моделей або складних залежностей.

Отже, кожен із методів має свої переваги та недоліки, і вибір конкретного методу залежить від специфіки задачі, типу даних, моделі, а також від того, наскільки глибокими або глобальними мають бути пояснення. У багатьох випадках для отримання повнішого розуміння роботи моделі може бути корисним поєднання декількох методів.

1.3 Огляд наявних практичних рішень до проблеми пояснення результатів класифікації

Важливо зосередитися на технологіях та програмних продуктах, які вже реалізовані для пояснення результатів класифікації за моделями глибокого навчання. Існує низка платформ, бібліотек та інструментів, що використовують методи інтерпретації, для пояснення роботи нейронних мереж та машинного навчання. Розглянемо деякі з них:

1. Google Cloud AI Platform (Explainable AI). Google Cloud AI Platform є потужною екосистемою для розробки, навчання та розгортання моделей машинного навчання (ML). Це рішення дає змогу організаціям використовувати інфраструктуру Google для масштабування своїх ML-ініціатив та забезпечення їх пояснюваності. Одним із ключових компонентів платформи є Explainable AI, що спрямована на надання інструментів для інтерпретації рішень моделей глибокого навчання.

Explainable AI надає можливість аналітикам та розробникам не тільки використовувати складні нейронні мережі, але й отримувати розуміння того, як ці моделі приймають рішення. Це особливо важливо для задач класифікації, де результати можуть бути дуже складними для інтерпретації. Використання методів пояснення, таких як Feature Importance та SHAP (Shapley Additive Explanations), допомагає визначити, наскільки важливими були певні ознаки або параметри моделі у прийнятті рішення [33].

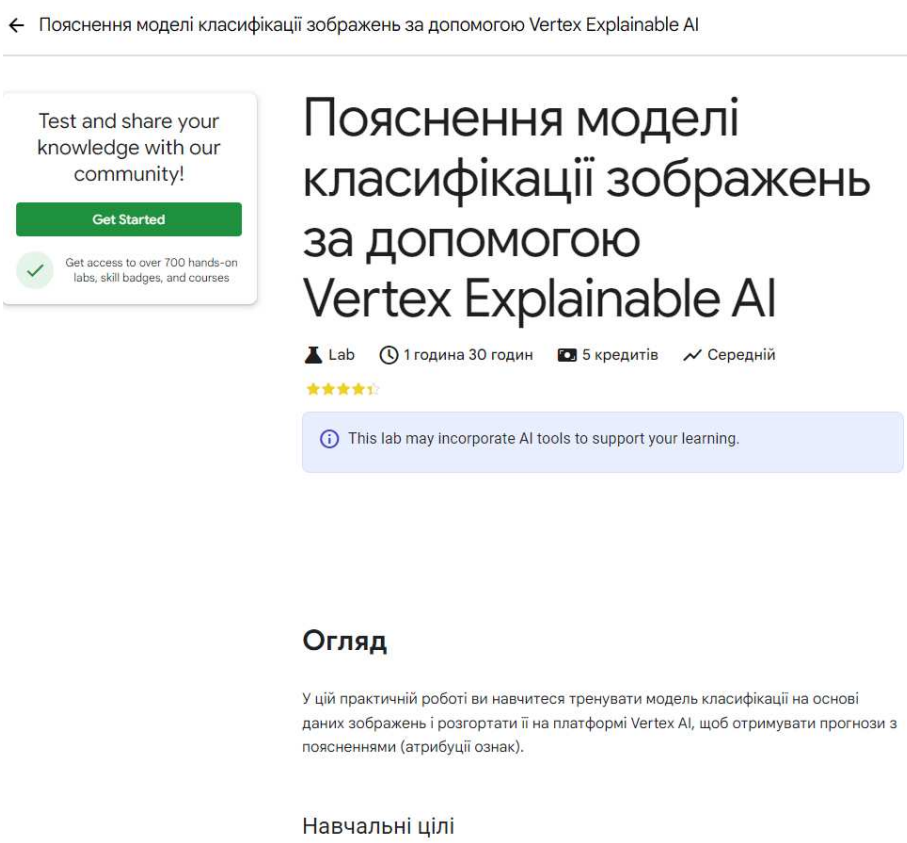


Рисунок 1.3 – Google Cloud AI Platform (Explainable AI) [33].

Google Cloud AI Platform інтегрує ці методи для надання користувачам більш прозорих моделей та дає змогу визначити, чому модель прийняла певне рішення. Це особливо важливо в критичних галузях, таких як охорона здоров'я, фінанси, та автономні системи, де невідомі або непояснені рішення можуть бути небезпечними або дорогими.

2. IBM Watson OpenScale. IBM Watson OpenScale – це платформа для моніторингу, керування і пояснення рішень ML у різних середовищах. Вона орієнтована на корпоративний ринок і забезпечує інструменти для контролю моделей на всіх етапах життєвого циклу: від розгортання до пояснення їхніх рішень, виявлення упереджень і перевірки продуктивності. Однією з ключових функцій є можливість прозорого пояснення результатів глибоких нейронних мереж і інших складних моделей [34].

Watson OpenScale надає широкий набір інструментів для роботи з моделями машинного навчання, незалежно від того, де вони були розгорнуті – у хмарі, локально

або на сторонніх платформах. Система підтримує пояснюваність через методи інтерпретації, такі як локальні пояснення (для окремих прогнозів) та глобальні пояснення (для всієї моделі), що особливо корисно для забезпечення прозорості в критичних галузях, таких як фінанси, охорона здоров'я, право тощо.

Локальні пояснення дають змогу користувачам аналізувати індивідуальні рішення моделей на основі конкретного набору ознак, що дає можливість зрозуміти, чому модель прийняла певне рішення в окремому випадку [35].

Глобальні пояснення надають загальну картину того, як працює модель, виявляючи ключові ознаки, що найбільше впливають на її прогнози. Це корисно для розуміння поведінки моделі на макрорівні.

Watson OpenScale (рисунок 1.4) також підтримує інтеграцію з платформами керування штучним інтелектом (ШІ) та ML, такими як IBM Watson Machine Learning, але може працювати й з моделями, побудованими на інших фреймворках, таких як TensorFlow, Keras, Scikit-learn тощо. Важливою особливістю є можливість моніторингу моделей на наявність упереджень і забезпечення відповідності вимогам регуляторів [36].

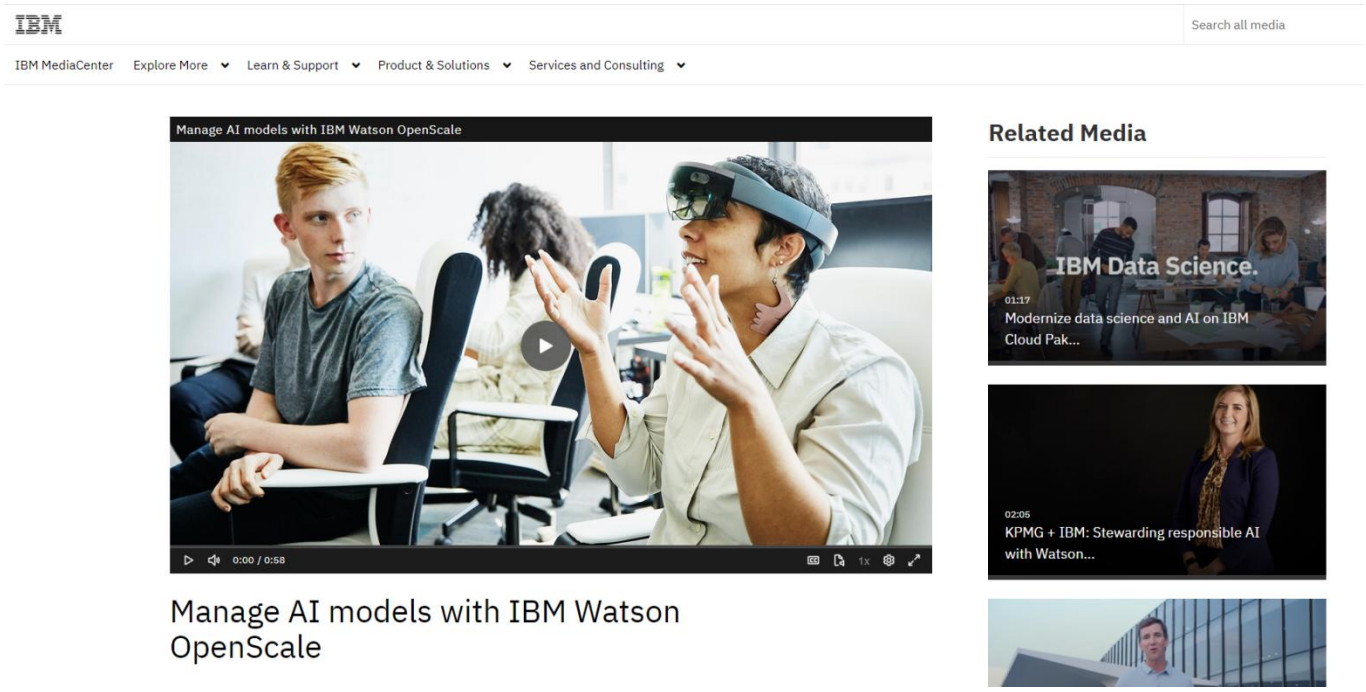


Рисунок 1.4 – IBM Watson OpenScale [35].

3. Microsoft Azure Machine Learning Interpretability SDK. Microsoft Azure Machine Learning Interpretability SDK – це потужна бібліотека від Microsoft Azure, що забезпечує інтерпретацію рішень моделей машинного навчання. Вона надає розробникам і аналітикам можливість пояснити поведінку моделей, зокрема, за допомогою таких популярних методів, як SHAP, LIME та Partial Dependence Plots (PDP). Ці методи дають змогу оцінити внесок окремих ознак у результати моделі, що значно підвищує прозорість і зрозумілість складних рішень ШІ [37].

Цей SDK є частиною екосистеми Azure Machine Learning, яка дає змогу легко інтегрувати інструменти пояснення моделей у вже існуючі проєкти або у процес розробки моделей. Інструмент підтримує широкий спектр моделей, включаючи ті, що побудовані на таких популярних фреймворках, як TensorFlow, Scikit-learn, PyTorch, а також кастомні рішення, що дає змогу його універсально застосовувати в різних сценаріях.

Однією з ключових можливостей SDK є локальна та глобальна інтерпретація моделей. Локальна інтерпретація дає змогу пояснювати окремі передбачення моделей, показуючи, які ознаки мали найбільший вплив на прийняте рішення. Глобальна інтерпретація надає загальне уявлення про роботу моделі, виявляючи найбільш значущі ознаки для всього набору даних і як вони впливають на загальні результати [38].

Інструмент на рисунку 1.5 також підтримує візуалізацію результатів інтерпретації, що допомагає не лише аналізувати результати, а й ефективно комунікувати їх як технічним, так і нетехнічним користувачам. Використовуючи Partial Dependence Plots (PDP), розробники можуть візуалізувати залежність прогнозованих результатів від окремих ознак [39].

Нижче подано таблицю 1.2, яка порівнює переваги та недоліки чотирьох платформ для пояснювального штучного інтелекту.

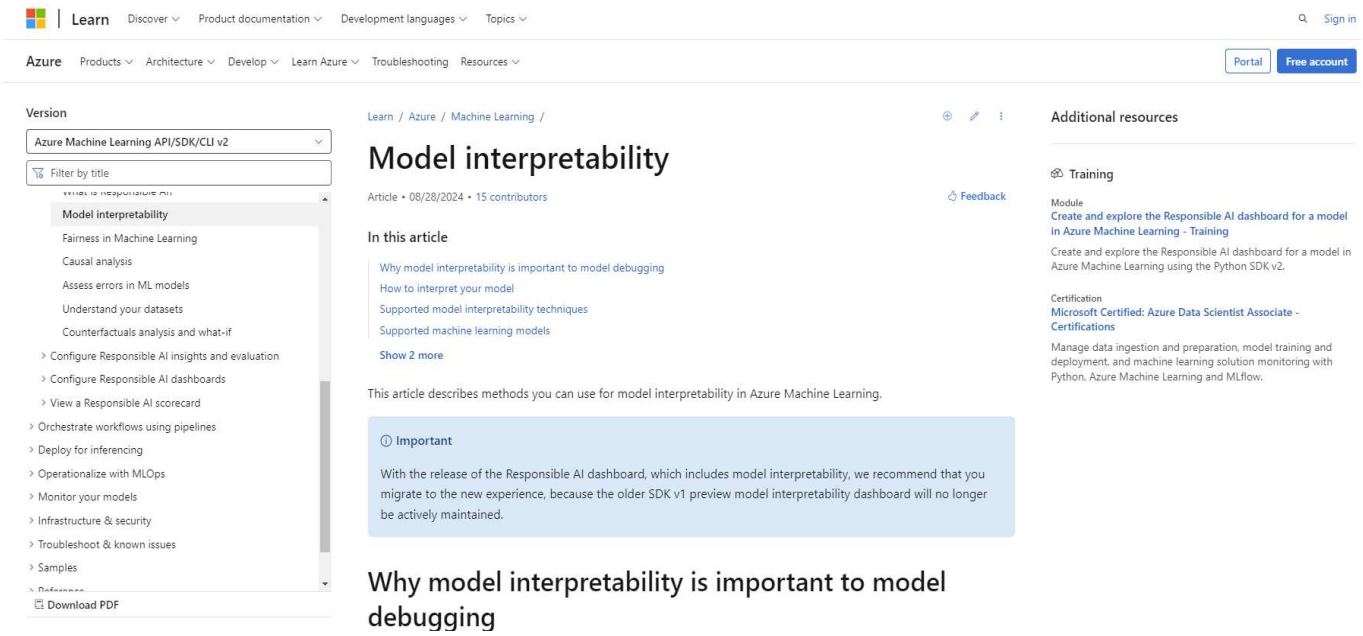


Рисунок 1.5 – Microsoft Azure Machine Learning Interpretability SDK [39].

Таблиця 1.2 – Переваг та недоліків платформ.

Платформа	Переваги	Недоліки
Google Cloud AI Platform (Explainable AI)	– інтеграція з іншими сервісами Google Cloud; – підтримка методів пояснення, таких як Feature Attribution та SHAP; – широкий спектр підтримуваних моделей і форматів даних.	– може бути дорогою для великих обчислювальних задач; – вимагає глибокого знання екосистеми Google Cloud.
IBM Watson OpenScale	– забезпечує моніторинг упередженості та продуктивності моделей; – інтеграція з Watson AI та іншими IBM сервісами;	– висока вартість для корпоративного використання; – може бути складною для налаштування.

	– гнучкість щодо різних моделей та фреймворків.	
Microsoft Azure Machine Learning Interpretability SDK	– інтеграція з екосистемою Azure; – підтримка LIME, SHAP та інших методів пояснення; – гнучкість у налаштуванні та сумісність з різними моделями.	– вимагає глибоких технічних знань для ефективного використання; – не завжди очевидна документація.

Отже, різні платформи та інструменти для пояснення рішень моделей машинного навчання на базі глибокого навчання, акцентуючи увагу на їхніх перевагах і недоліках [40].

Google Cloud AI Platform (Explainable AI) пропонує потужні інструменти для пояснення рішень, інтеграцію з іншими сервісами Google і підтримку популярних фреймворків, але має обмеження для локальних моделей та високі витрати на використання.

IBM Watson OpenScale забезпечує широкий набір інструментів для моніторингу та пояснення моделей, інтегрується з платформою IBM, але складність налаштування та обмежена функціональність безкоштовної версії можуть бути проблемами для малих організацій.

Microsoft Azure Machine Learning Interpretability SDK дає змогу пояснювати моделі за допомогою популярних методів, добре інтегрується з Azure, але має складність для новачків та потребує великих обчислювальних ресурсів.

Загалом, всі інструменти мають свої переваги у поясненні моделей, але також і недоліки, які можуть вплинути на вибір відповідного рішення для конкретних потреб організації.

1.4 Постановка задачі

У першому розділі кваліфікаційної роботи було проведено дослідження методів пояснення результатів класифікації за моделями глибокого навчання. Визначено, що ключову роль у цьому відіграє здатність моделі розкривати внутрішні механізми прийняття рішень, надаючи користувачам зрозумілу інтерпретацію впливу різних ознак. Зокрема, було розглянуто важливість вагових коефіцієнтів, інтерпретованості моделей, чутливості до змін вхідних даних, а також баланс між складністю моделі та її здатністю до узагальнення. Аналіз показав, що вибір архітектури моделі, зокрема використання механізмів уваги в трансформерах та рекурентних мережах, суттєво впливає на здатність до інтерпретації результатів.

В результаті аналізу наявних методів пояснення результатів задач класифікації за моделями глибокого навчання виявлено проблему недостатньої прозорості та зрозумілості їхніх рішень, що обмежує їхнє застосування до критичних галузей. Для вирішення цієї проблеми, дане дослідження ставить за мету підвищити рівень якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання. Для досягнення мети роботи пропонується спроєктувати методи пояснення результатів класифікації, які ґрунтуються на моделях глибокого навчання з використанням технологій машинного навчання, що забезпечить високий рівень зрозумілості та інтерпретації рішень моделей.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Провести аналіз моделей, методів та засобів пояснювального штучного інтелекту.

2. Спроєктувати модель подання результатів задач класифікації за моделями глибокого навчання через ознаки моделей машинного навчання.

3. Спроєктувати метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

4. Виконати програмну реалізацію та провести експериментальне тестування спроєктованого методу пояснення результатів задач класифікації.

РОЗДІЛ 2 Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

2.1 Проєктування методу пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

У цій роботі пропонується новий підхід до XAI у контексті глибокого навчання (DL), який перевіряється на прикладі двох моделей DL: CNN та надійно оптимізованої двонаправленої моделі перетворень на основі кодованих представлень (RoBERTa). У цьому контексті DL розглядається як підгалузь ML, що ґрунтується на штучних нейронних мережах з навчанням ознак. Термін «глибокий» вказує на здатність використовувати численні шари в глибокій нейронній мережі (DNN). У запропонованому підході вхідні дані подаються через передостанній шар багатошарової нейронної мережі. Цей шар можна виділити з навченої моделі будь-якого типу нейронних мереж, таких як згорткові мережі, трансформери, рекурентні мережі, автоенкодери тощо.

Постановка задачі та запропонований підхід до її вирішення. Швидкий розвиток DL-моделей, заснованих на різних архітектурах, та досягнення результатів, які перевершують ML-моделі на тих самих задачах і наборах даних, з одного боку, і потреба в поясненні результатів, отриманих ML-моделями для людини, з іншого, роблять задачу пояснення результатів DL за допомогою ML надзвичайно актуальною.

Розглянемо задачі, в яких вхідними даними є зображення, сигнали тощо, а вихідною інформацією є класифікація цих даних на категорії. Припустимо, що ці задачі можна вирішити як за допомогою DL, так і за допомогою ML. Тобто ми застосовуємо дві порівнювані моделі для одного набору даних. У цьому випадку DL-модель розглядається як модель з функціональним поданням (FM), а ML-модель – як модель з методологічним поданням (MM). Схема, представлена на рисунку 2.1, демонструє порівняння методів вилучення ознак за допомогою DL та ML моделей при виявленні інформативних атрибутів зображень.

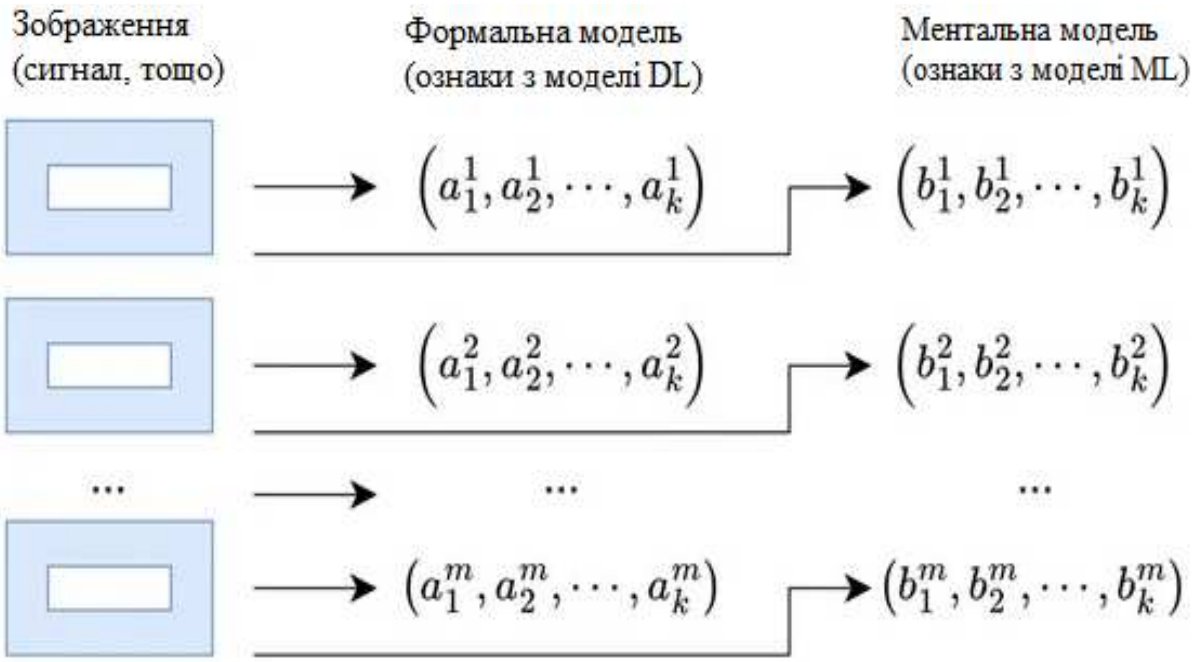


Рисунок 2.1 – Процес вилучення ознак з набору вхідних даних (перший стовпчик) за допомогою двох різних обчислювальних моделей.

Другий стовпчик, позначений як «Формальна модель», показує вектори ознак, отримані за допомогою DL-моделі, позначені як aji, де i – індекс ознаки, а j – індекс зображення. Третій стовпчик, позначений як «Ментальна модель», відображає вектори ознак, отримані ML-моделлю, позначені як bji. Кожен рядок відповідає одному зображенню, яке обробляється обома моделями, підкреслюючи різні розміри k і l простору ознак, що представлені кожною моделлю.

Запропонований підхід частково збігається з апроксимацією, тобто описом однієї функції (навіть якщо вона представлена у вигляді таблиці) через іншу функцію (можливо, також у табличному вигляді). У цій роботі пропонується використовувати перехідну матрицю між двома моделями характеристик (представленими у вигляді матриць) для одного й того самого набору вхідних даних як таку функцію і розроблено механізм визначення цієї перехідної матриці.

Цей підхід дає змогу перетворити вектор ознак FM (які є складними для розуміння кінцевим користувачем) в новий вектор ознак, отриманий з MM (які є більш зрозумілими для кінцевого користувача) і можуть слугувати інтерпретацією (апроксимацією) розв’язку задачі. Тут MM подає простір, що складається з векторів

ознак – l-вимірних числових векторів, які описують певний об’єкт. В нашій нотації FM представлено матрицею A (1), отриманою з DL-моделі, а MM – матрицею B (2). Вектор ознак FM, який є k-вимірним вектором числових ознак, може, наприклад, містити значення ваг нейронів передостаннього шару DNN.

Задачу пояснення результатів, отриманих за допомогою DL-моделі, пропонується вирішити за допомогою перехідної матриці T=BA^{-1} та перетворення b_t^*=Ta_j^* , i=1,k, j=1,l. Це перетворення дає змогу отримати результати в ознаках MM, які є більш зрозумілими для кінцевого користувача. Запропонований метод отримання перехідної матриці T=BA^{-1} ілюструється на рисунку 2.2.

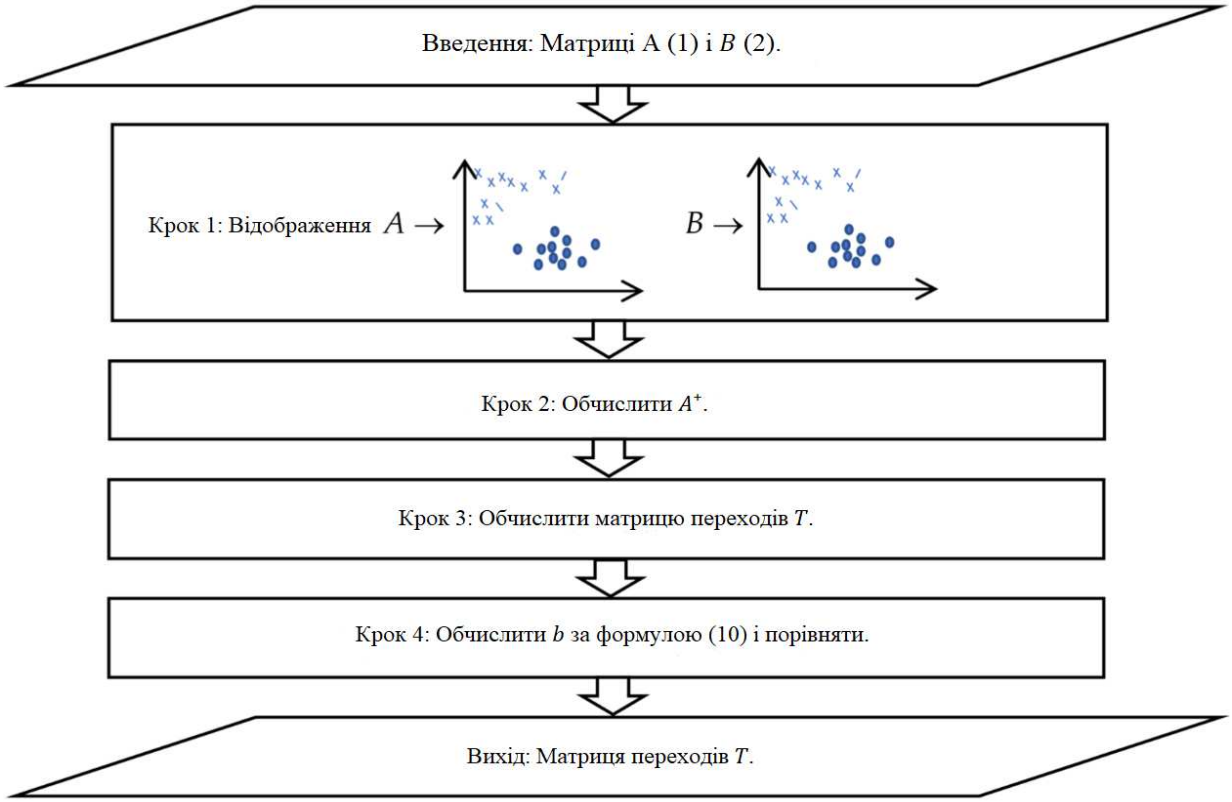


Рисунок 2.2 – Схема запропонованого методу перетворення даних з матриць A та B у перехідну матрицю T.

Вихідним результатом цього методу є перехідна матриця T, яка відображає зв’язок між двома матрицями.

Основні етапи запропонованого методу для отримання перехідної матриці T=BA^{-1} наведені нижче:

Вхідні дані: З того самого набору даних, що відповідає кожному анотованому зразку, отримуються матриця A відповідно до структури (1) з передостаннього шару навченої DNN та матриця B відповідно до структури (2) з навченої ML-моделі. Кожен рядок матриці A та відповідний рядок матриці B представляють один і той самий зразок (об'єкт) з навчальної вибірки.

Крок 1. З використанням будь-якого інструмента візуальної аналітики (VA), призначеного для зменшення розмірності простору ознак (наприклад, MDS), отримують графічні подання векторів у вигляді точок на площині. Важливо, щоб взаємне розташування точок для різних моделей було подібним (до можливих поворотів), а групування об'єктів відповідало анотованим класам.

Крок 2. Досліджується можливість застосування формул A^{+}=(ATA)−1AT, A^{+}=AT(AAT)−1 або A^{+}=VΣ+UT, для обчислення A^{+}.

Крок 3. Обчислюється перехідна матриця T за формулою T≈ A^{+}B.

Крок 4. Використовуючи отриману матрицю T, для всіх векторів простору FM обчислюються результати за формулою b_t^*=Ta_j^* , i=1,k, j=1,l. Отримані вектори порівнюються з векторами простору MM для перевірки коректності перехідної матриці T.

Вихідні дані: Перехідна матриця T.

Після отримання перехідної матриці T за зазначеними етапами та отримання результатів з навченої DL-моделі, ми застосовуємо формулу b_t^*=Ta_j^* , i=1,k, j=1,l. для пояснення результатів у вигляді MM. Далі буде наведено числовий приклад із синтетичними даними, який ілюструє етапи цього методу.

Ілюстративний числовий приклад.

Щоб продемонструвати запропонований вище формалізм, ми взяли п'ятнадцять векторів a_t^n, i=1,k ,n=1,15, k=5, з простору векторів FM (3). Зазначимо, що ці вектори належали до трьох класів. Також варто зазначити, що ознаки цих векторів є незрозумілими для людини:

Class 1 : a^1 = (2.8,1.8,-2.8,1.3,0.4), a^2=(2.9,-1.9,-2.9,1.4,0.5), a^3= (3,-2,-3,1.5,0.6), a^4 = (3.1,-2.1,-3.1,1.6,0.7), a^5 = (3.2,-2.2,-3.2,1.7,0.8);

Class 2 : a^6= (-1.6,-2.5,1.5,0.2,0.6), a^7 =(-1.3,-2.7,1.3,0.4,0.8), a^8 = (-1,-3,1.5,0.6,1), a^9 = (-0.7,-3.2,1.7,0.8,1.2), a^{10} = (-0.5,-3.5,1.9,1,1.4);

Class 3 : a^{11} = (1.2,-1.2,0.7,-0.3,-2.8), a^{12}= (1.1,-1.1,0.8,-0.4,-2.9), a^{13} = (1,-1,0.844444,-0,44444,-3), a^{14} = (0.9,-0.9,0.85,-0.45,-3.1), a^{15} = (0.8,-0.8,0.9,-0.5,-3.2).

Зазначимо, що вектори $a3$, $a8$, $a13$ були попарно перпендикулярними, тобто $a3 \perp a8$, $a3 \perp a13$, $a8 \perp a13$. Ми вибрали відповідні п'ятнадцять векторів (4) із векторного простору ММ, причому значення цих характеристик є зрозумілим для людини.

Class 1 : b^1 = (2.8,1.8,-2.8,1.3,0.4), b^2=(2.9,-1.9,-2.9,1.4,0.5), b^3= (3,-2,-3,1.5,0.6), b^4 = (3.1,-2.1,-3.1,1.6,0.7), b^5 = (3.2,-2.2,-3.2,1.7,0.8);

Class 2 : b^6= (-1.6,-2.5,1.5,0.2,0.6), b^7 =(-1.3,-2.7,1.3,0.4,0.8), b^8 = (-1,-3,1.5,0.6,1), b^9 = (-0.7,-3.2,1.7,0.8,1.2), b^{10} = (-0.5,-3.5,1.9,1,1.4);

Class 3 : b^{11} = (1.2,-1.2,0.7,-0.3,-2.8), b^{12}= (1.1,-1.1,0.8,-0.4,-2.9), b^{13} = (1,-1,0.844444,-0,44444,-3), b^{14} = (0.9,-0.9,0.85,-0.45,-3.1), b^{15} = (0.8,-0.8,0.9,-0.5,-3.2).

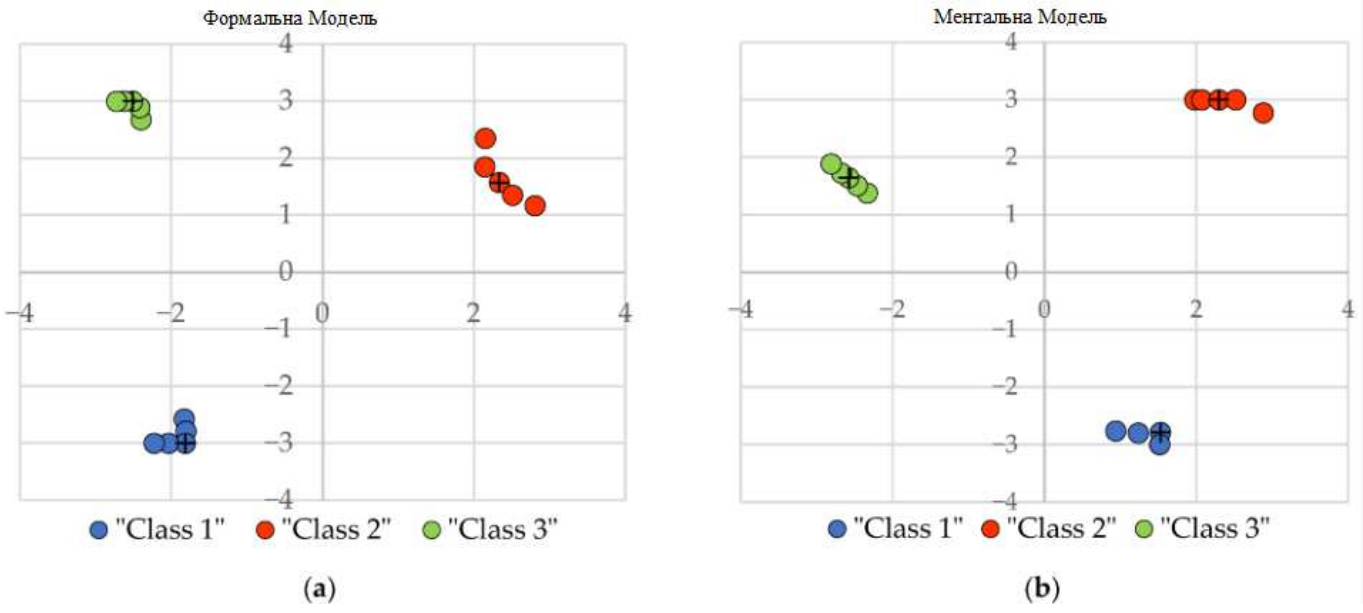


Рисунок 2.3 – На рисунку представлено порівняння меж класифікації, визначених Формальною моделлю (FM) і Ментальною (Людською) моделлю (ММ).

Крім того, вектори $b3$, $b8$, $b13$ також були попарно перпендикулярними, тобто $b3 \perp b8$, $b3 \perp b13$, $b8 \perp b13$. Перевіримо, чи дійсно зазначені вектори утворюють три

класи. Для цього було використано метод зниження розмірності MDS, R5→R2 і R4→R2. Результати цього аналізу наведено на рисунку 2.3, де вектори $a3$,$a8$, $a13$ і $b3$, $b8$, $b13$ позначені символом “+”.

На рисунку 2.3a) показано, як FM розподіляє три класи, що видно з угрупування векторів класів у двовимірному просторі ознак. На рисунку 2.3b) зображено класифікацію, виконану MM, яка демонструє схожі угрупування, але з помітними відмінностями у розташуванні та накладанні векторів класів. Базові вектори позначені символом “+”.

Основні кроки методу:

Крок 1: Розбиття тексту на фрагменти

– метод працює шляхом генерації спрощеної моделі для інтерпретації.

– Вхідний текст розбивається на слова або фрази, що стають окремими ознаками.

Крок 2: Пертурбація даних

– метод створює спрощені копії вхідного тексту з видаленими або заміненими словами.

– Кожна "версія" тексту передається в модель, щоб подивитися, як змінюється прогноз.

Крок 3: Отримання прогнозів моделі

– Кожна модифікована копія тексту проходить через модель.

– Для кожного варіанту отримується ймовірність класифікації (позитивний/негативний).

– Це допомагає зрозуміти, як видалення певних слів впливає на рішення моделі.

Крок 4: Побудова лінійної моделі

– метод будує локальну лінійну модель навколо оригінального тексту.

– Лінійна модель показує, які слова є найбільш значущими для класифікації.

– Коефіцієнти ваг лінійної моделі вказують на "вплив" кожного слова на рішення.

2.2 Визначення ключових ознак класифікаційних моделей

У цьому розділі глибше проаналізовані основні характеристики та параметри моделей машинного навчання, що впливають на рішення моделі в задачах класифікації. Важливо підкреслити, що розуміння цих ознак допоможе створити прозору, інтерпретовану систему пояснень результатів класифікації, що є основою для підвищення довіри до моделі та її рішень.

У контексті класифікаційних моделей машинного навчання ключовими ознаками, що впливають на прийняття рішень, є вага ознак у моделі, інтерпретованість моделей, чутливість до змін у даних та складність моделі й здатність до узагальнення.

1. Вага ознак у моделі:

– коефіцієнти ваги (для лінійних моделей): прямий показник впливу ознак на результат;

– інтерпретованість ваги (для нейронних мереж): ускладнене розуміння через багатошаровість;

– метрики значущості ознак (SHAP, LIME): використовуються для оцінювання впливу кожної ознаки на рішення моделі.

Вага ознак є критично важливим параметром, оскільки вона відображає, наскільки кожна ознака впливає на кінцеве рішення моделі (рисунок 2.4).

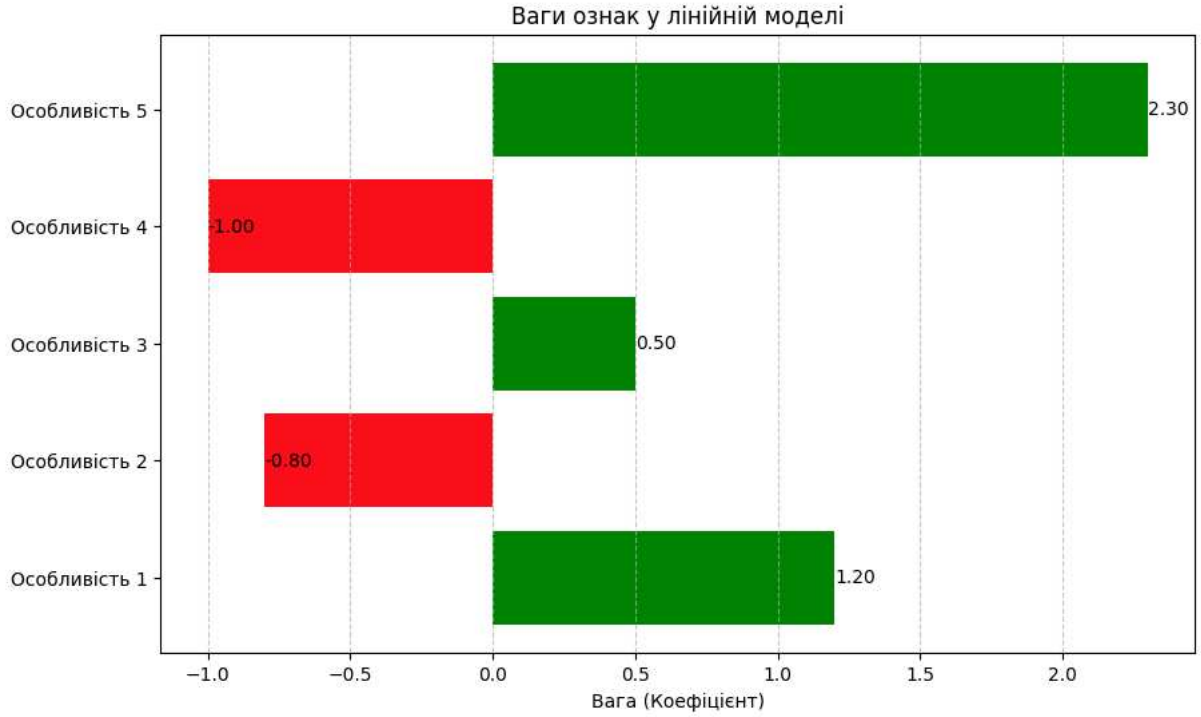


Рисунок 2.4 – Візуалізація ваги ознак у лінійній моделі

У лінійних моделях, таких як лінійна регресія, кожна ознака має відповідний коефіцієнт ваги, який можна легко інтерпретувати. Якщо коефіцієнт позитивний, це означає, що зростання значення ознаки веде до збільшення прогнозованого результату. Чим більший абсолютний коефіцієнт, тим більше вплив ознаки на результат. У складніших моделях, таких як нейронні мережі, ваги також мають важливе значення, але їх важче інтерпретувати через багатошаровість, що ускладнює розуміння впливу ознак на результат.

Ось графік, що відображає ваги ознак у лінійній моделі. Зелені стовпчики представляють ознаки з позитивним впливом на результат, а червоні – з негативним. Значення коефіцієнтів на стовпчиках показують ступінь впливу кожної ознаки.

2. Інтерпретованість моделі:

– прозорі моделі: дерева рішень, лінійна регресія, наївний Баєс;

– чорні ящики: глибокі нейронні мережі, градієнтний бустинг;

– методи пояснення: Grad-CAM, механізм уваги, пояснення через прототипи (Prototype-based explanations).

Інтерпретованість моделі визначає, наскільки зрозумілим є процес прийняття рішень. Прозорі моделі, такі як дерева рішень або лінійні регресії, є інтерпретованими, оскільки вони надають чіткі правила або відносини між ознаками та результатами. Наприклад, дерево рішень надає графічне уявлення про те, як здійснюється класифікація на основі різних ознак, що робить його зрозумілим для фахівців та користувачів (рисунок 2.5). У «чорних ящиках», таких як глибокі нейронні мережі, процес прийняття рішень складно зрозуміти, що створює виклики у випадках, де потрібна прозорість рішень.

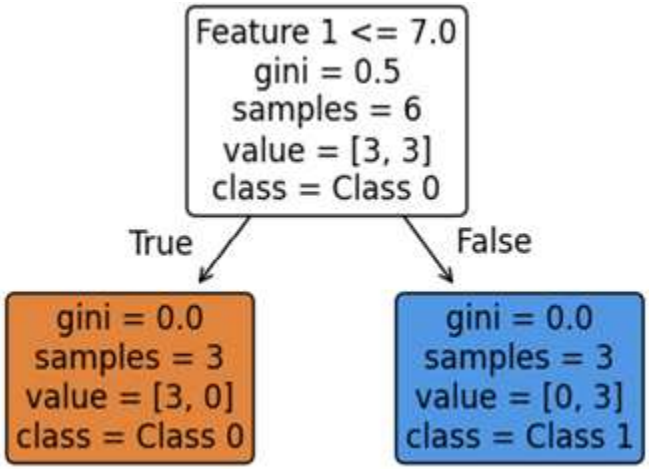


Рисунок 2.5 – Дерево рішень для демонстрації інтерпретованості

Ось приклад спрощеного дерева рішень, яке показує, як модель приймає рішення на основі двох ознак. У кожному вузлі перевіряється умова (порівняння значення ознаки з порогом), і залежно від цього відбувається перехід до наступної гілки.

Ця візуалізація підкреслює прозорість моделі, оскільки можна відстежити весь процес прийняття рішень від кореневого вузла до кінцевого результату. На відміну від складніших моделей, таких як нейронні мережі, дерево рішень легко інтерпретувати та пояснити користувачам.

3. Чутливість до змін у даних:

– варіативність моделей: залежність точності від незначних змін у вхідних даних;

– аналіз стабільності: вплив вибірок або аномальних значень;

– техніки оцінювання: пертурбація даних, стрес-тестування на змінах значень ознак.

Чутливість моделей до змін у вхідних даних може суттєво вплинути на їхню продуктивність. Моделі з високою варіативністю, такі як глибокі нейронні мережі, можуть показувати сильну реакцію на незначні зміни в даних, що може призвести до нестабільності у прогнозах (рисунок 2.6). Аналіз чутливості дає змогу зрозуміти, які ознаки найбільше впливають на результати та як зміни цих ознак можуть змінювати результати класифікації.

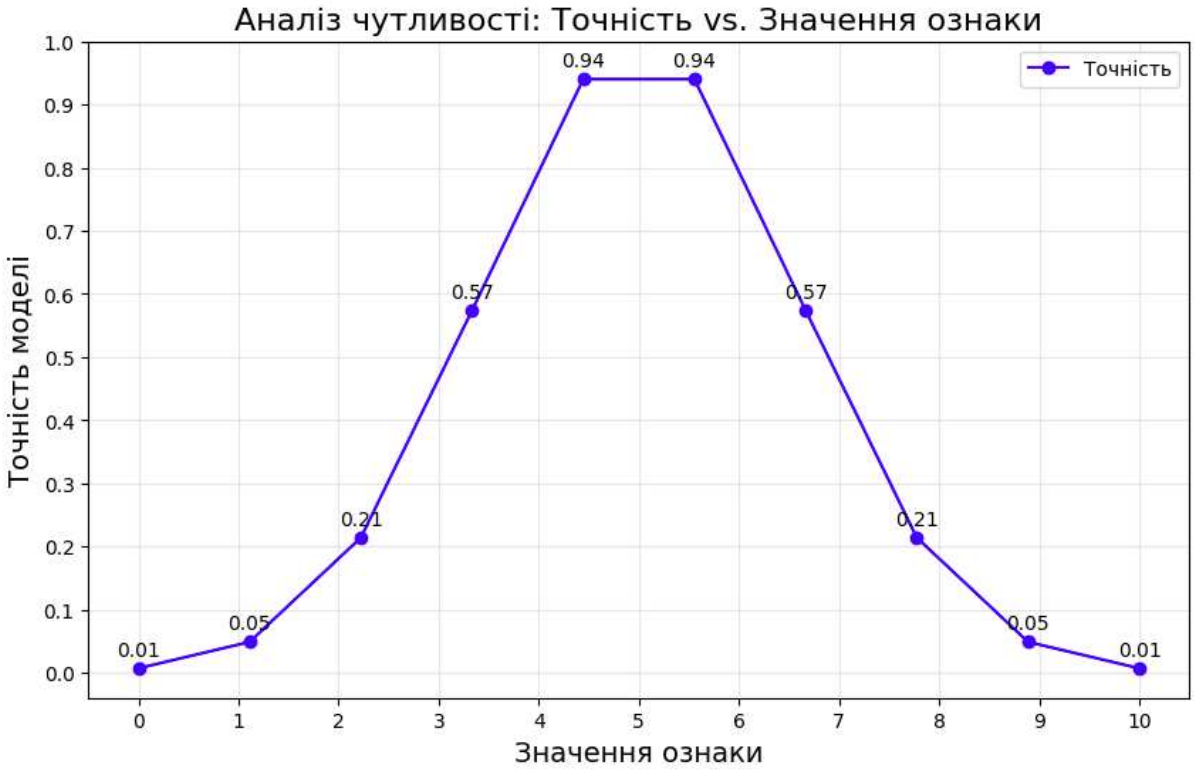


Рисунок 2.6 – Дерево рішень для демонстрації інтерпретованості

Графік демонструє аналіз чутливості, в якому представлена залежність точності моделі від зміни значення однієї з ознак, з горизонтальною віссю, що відображає варіацію значення ознаки від 0 до 10, і вертикальною віссю, яка показує точність моделі в діапазоні від 0 до 1, де точність досягає максимуму при значенні ознаки 5 і швидко зменшується в обидва боки, вказуючи на чутливість моделі до змін

значення ознаки. Крива, що з'єднує точки з маркерами, полегшує сприйняття даних, а текстові підписи біля маркерів надають точні значення точності, тоді як сітка покращує читабельність графіка, доповненого легендою, що пояснює подання точності. Загалом, графік ілюструє, як варіація значення ознаки впливає на точність моделі, підкреслюючи критичність певних значень для оптимальної продуктивності.

4. Складність моделі та здатність до узагальнення:

– кількість параметрів: велика кількість може спричиняти перенавчання;

– регуляризація: методи (L1, L2) для боротьби з перенавчанням;

– балансування між похибкою на навчальних і тестових даних.

Складність моделі безпосередньо впливає на її здатність узагальнювати результати на нові дані (рисунок 2.7).

Рисунок 2.7 – Складність моделі та здатність до узагальнення

Моделі з великою кількістю параметрів, такі як глибокі нейронні мережі, мають потенціал для високої точності на навчальних даних, але можуть бути схильні

до перенавчання. Здатність до узагальнення означає, що модель повинна зберігати свою продуктивність на нових, раніше небачених даних. Аналіз складності моделі та оцінка її продуктивності на валідаційних і тестових наборах даних дає змогу виявити потенційні проблеми з узагальненням.

Графік «Складність моделі та здатність до узагальнення» ілюструє залежність рівня похибки моделі від її складності, представленої кількістю параметрів, що варіюється від 1 до 20. На горизонтальній осі показано значення складності моделі, тоді як на вертикальній осі відображається рівень похибки, що коливається від 0 до 1. Синя крива, що подає похибку на навчанні, має зменшувальний характер і досягає нуля з підвищенням складності, тоді як червона крива, що відображає похибку на тестуванні, показує коливання з легким зниженням при зростанні складності, з певними випадковими відхиленнями. Графік демонструє важливість балансування між складністю моделі та її здатністю до узагальнення: занадто проста модель може недоучувати, тоді як занадто складна – переучуватися, що підкреслює потребу в оптимальному виборі параметрів для досягнення найкращої продуктивності.

Загалом, вивчення ключових ознак класифікаційних моделей допомагає зрозуміти, як ці фактори впливають на процес прийняття рішень. Це важливо для розробки прозорих, зрозумілих і надійних моделей, які можуть бути використані в різних галузях, де інтерпретованість і надійність рішень є критичними. Аналіз впливу ознак на результати класифікації є важливим етапом у розумінні поведінки моделей, особливо у випадку складних архітектур глибокого навчання.

5. Архітектура моделі:

– кількість шарів: впливає на здатність моделі навчати складні залежності;

– типи шарів: рекурентні, згорткові, механізм уваги;

– функції активації: ReLU, Sigmoid, Softmax, їх вплив на інтерпретацію.

6. Використання механізмів уваги:

– трансформери: фокусування на критичних частинах вхідних даних;

– візуалізація ваг (наприклад, у тексті або зображеннях) для пояснення рішень.

Архітектура моделі глибокого навчання може суттєво впливати на її здатність до пояснення та інтерпретації результатів. Використання шарів уваги в моделях,

таких як трансформери, дає змогу моделі фокусуватися на певних частинах вхідних даних, що спрощує інтерпретацію. Завдяки механізму уваги можна візуалізувати, які частини вхідних даних, наприклад, слова в тексті або пікселі в зображенні, модель вважає найбільш важливими для прийняття рішення, що надає додатковий контекст та робить рішення більш зрозумілими для користувачів.

7. Обробка послідовних даних:

– рекурентні мережі (RNN, LSTM, GRU): складність відстеження впливу попередніх станів;

– методи аналізу: heatmaps, аналіз ваг у прихованих шарах.

RNN, орієнтовані на обробку послідовних даних, таких як текст або часовий ряд, можуть бути складними для пояснення. Через свою природу RNN важко ідентифікувати, які саме попередні стани впливають на фінальне рішення. Проте методи, як-от візуалізація ваг у прихованих шарах або LSTM-діаграми, можуть допомогти зрозуміти, які частини послідовності мають найбільший вплив.

8. Візуалізація результатів:

– Grad-CAM: виділення важливих зон на зображеннях;

– дерева рішень: чітке подання логіки прийняття рішень;

– аналіз чутливості: вплив змін значень ознак на точність.

CNN, зазвичай використовувані для оброблення зображень, також можуть бути складними для інтерпретації. Проте техніки на зразок Grad-CAM дають змогу візуалізувати, які області зображення є важливими для класифікації, що дає змогу зрозуміти, на що «дивиться» модель під час прийняття рішення.

Порівняння різних архітектур дає змогу зрозуміти, які з них забезпечують кращу інтерпретованість. Прості моделі, як-от лінійна регресія, забезпечують високу прозорість, тоді як складні нейронні мережі можуть бути менш зрозумілими. Моделі-»чорні ящики», які важко пояснити, можуть бути менш корисними в ситуаціях, де потрібні детальні пояснення. Це підкреслює важливість вибору архітектури моделі відповідно до специфіки задачі, з урахуванням вимог до інтерпретованості.

Проведення експериментів із різними архітектурами може виявити, які з них найкраще справляються з певними типами даних і задачами, а також які з них найкраще пояснюють свої результати.

Вивчення методів аналізу впливу ознак на результати класифікації та впливу архітектури глибокого навчання на пояснювальність моделей є важливим аспектом для розробки більш зрозумілих та інтерпретованих систем машинного навчання. Це дає змогу фахівцям краще розуміти процес прийняття рішень моделями та використовувати цю інформацію для підвищення надійності й точності у практичних застосуваннях.

Ключові параметри моделі машинного навчання можуть суттєво впливати на її пояснювальність і загальну продуктивність. Глибина нейронної мережі, або кількість шарів, є критичним параметром, що визначає складність моделі. Більш глибокі мережі мають велику потужність для навчання складних залежностей у даних завдяки здатності створювати абстракції через численні шари. Проте це також ускладнює їх інтерпретацію, оскільки з’являється велика кількість параметрів, які взаємодіють між собою. Чим більше шарів, тим важче простежити, як окремі ознаки впливають на фінальний результат. Моделі з меншою кількістю шарів можуть бути простішими у розумінні й поясненні, але їхня здатність до навчання складних структур може бути обмежена. Тому важливо знайти баланс між глибиною моделі та її інтерпретованістю, враховуючи специфіку задачі.

Активаційні функції визначають, як обчислюються виходи нейронів у різних шарах. Вибір функції активації може впливати на нелінійність моделі та її пояснювальність. Функції активації, такі як ReLU, є простими у реалізації та інтерпретації. Вони дають змогу уникнути проблеми згасання градієнта, але не мають симетрії, що може ускладнити інтерпретацію. Більш складні функції, такі як softmax, часто використовуються в багатокласових задачах, але можуть створити труднощі в розумінні, оскільки вони перетворюють виходи моделі в ймовірності, які важко інтерпретувати. Вибір активаційної функції також впливає на здатність моделі навчатися та узагальнювати результати. Наприклад, функції, що допускають

негативні значення, можуть мати кращі властивості для навчання, але можуть ускладнити пояснення через менш інтуїтивно зрозумілі результати.

9. Гіперпараметри навчання:

– швидкість навчання (learning rate): її вплив на стабільність і швидкість збіжності;

– розмір партії (batch size): баланс між стабільністю градієнтів і часом навчання;

– кількість епох: уникнення недонавчання або перенавчання.

Гіперпараметри навчання, такі як швидкість навчання, розмір партії та кількість епох, також можуть впливати на продуктивність моделі та її пояснювальність. Швидкість навчання визначає, наскільки великими будуть кроки в напрямку зниження функції втрат під час навчання. Занадто висока швидкість може призвести до нестабільності, тоді як занадто низька може затримати навчання, що може вплинути на узагальнювальні властивості моделі й, як наслідок, на пояснювальність рішень. Розмір партії впливає на кількість зразків, які обробляються під час одного кроку навчання. Малий розмір партії може призвести до нестабільних оцінок градієнта, що погіршує узагальнювальні властивості моделі. Кількість епох визначає, скільки разів модель проходить через весь навчальний набір даних. Вибір кількості епох впливає на те, чи модель перенавчається або недонавчається. Перенавчена модель може давати чудові результати на навчальних даних, але погано працювати на нових, що зменшує її пояснювальність і надійність.

Отже, розуміння ключових параметрів моделі машинного навчання, таких як кількість шарів, типи активаційних функцій і гіперпараметри навчання, є критично важливим для розробки моделей, які є продуктивними і пояснювальними. Збалансування цих параметрів дає змогу створювати моделі, які не тільки добре класифікують дані, але й роблять свої рішення зрозумілими для користувачів, що особливо важливо в таких сферах, як медицина або фінанси, де необхідно пояснювати рішення.

2.3 Спосіб подання результатів класифікації

На початку пункту слід представити схему, яка узагальнює всі кроки способу подання результатів класифікації. Схема на рисунку 2.8 включає наступні етапи.

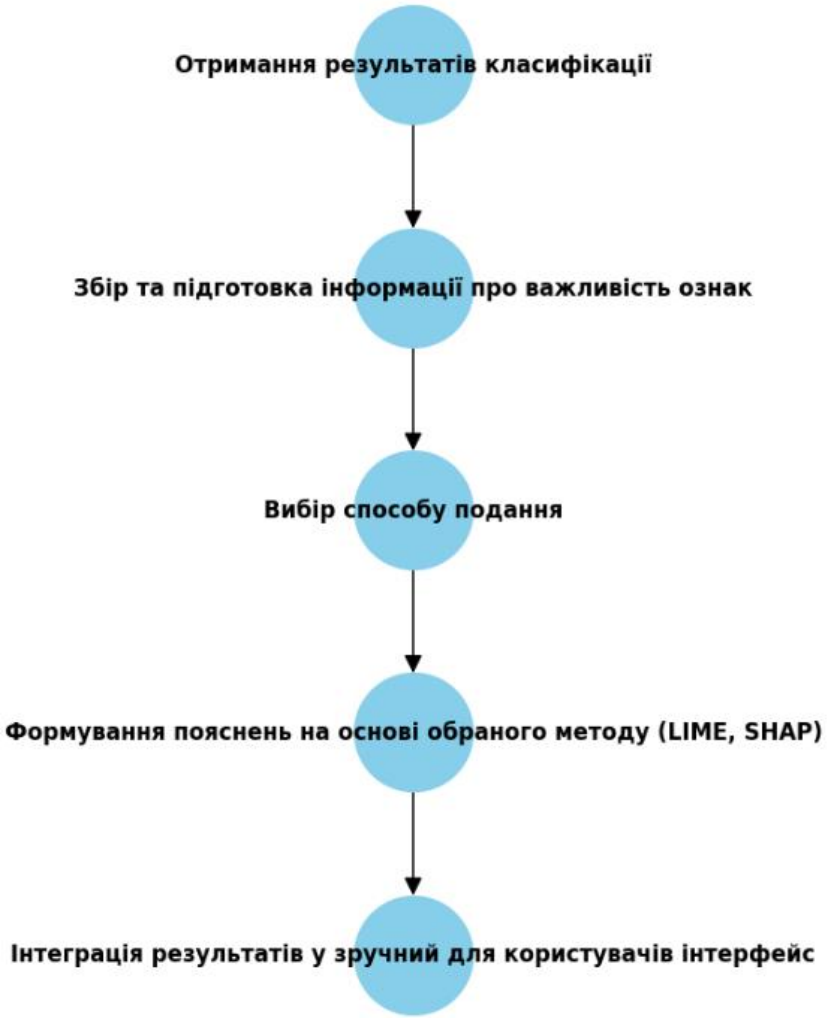


Рисунок 2.8 – Схема підходу до подання результатів

Кроки формування способу подання результатів класифікації.

Крок 1. Отримання результатів класифікації:

– виконати класифікацію за допомогою обраної моделі (нейронна мережа, ансамблеві методи тощо);

– зберегти основні результати, як-от прогнозовані значення, ймовірності, категорії.

Крок 2. Збір та підготовка інформації про важливість ознак:

– використовувати інструменти пояснюваності, такі як LIME або SHAP, для визначення впливу ознак на результат;

– сформувати локальні (для конкретного прикладу) та глобальні (загальні для моделі) пояснення;

– підготувати дані для візуалізації, наприклад, у форматі таблиць або графіків.

Крок 3. Вибір способу подання:

– визначити, який спосіб подання підходить найкраще залежно від типу користувача:

– таблиці: для технічних користувачів, які потребують точних даних;

– графіки: для швидкого аналізу впливу ознак;

– інтерактивні панелі: для нефахівців, таких як бізнес-аналітики або медичні працівники.

Крок 4. Формування пояснень:

– для кожного способу подання створити відповідне пояснення:

– у таблицях – стовпці з важливістю ознак та їх ваговими коефіцієнтами;

– у графіках – візуалізації, такі як бар-чарти або теплові карти для кращого розуміння;

– в інтерактивних панелях – динамічні графіки або аналітичні модулі.

Крок 5. Інтеграція у користувацький інтерфейс:

– реалізувати зручний інтерфейс для відображення результатів:

– інтерактивні аналітичні панелі для перегляду класифікаційних результатів;

– вбудовування в існуючі системи підтримки прийняття рішень (DSS);

– надання можливості користувачам налаштовувати вид подання результатів.

Крок 6. Перевірка та тестування:

– провести тестування структури пояснень на реальних користувачах;

– зібрати відгуки для вдосконалення;

– переконатися, що пояснення зрозумілі.

Цей підхід дає змогу формувати та представляти результати класифікації в зрозумілому форматі для різних груп користувачів. Використання методів, таких як

LIME та SHAP, спрощує процес пояснення та підвищує довіру до моделей у чутливих сферах, таких як медицина, фінанси та кібербезпека.

Висновки до розділу 2

Проєктування методу пояснення вимагала розуміння принципів роботи різних моделей та чіткого визначення ключових ознак, що впливають на рішення. Досліджено, як вагові коефіцієнти в лінійних моделях дають пряме розуміння впливу ознак, тоді як інтерпретація ваг у складніших нейронних мережах є більш ускладненою. Для подолання цього виклику були розглянуті методи оцінювання значущості ознак, такі як SHAP та LIME, які дають змогу оцінити вплив кожної ознаки на рішення моделі, сприяючи створенню більш прозорих та зрозумілих систем. Важливим аспектом є візуалізація результатів, що допомагає наочно продемонструвати вплив ознак на рішення моделі.

Сформульовано підхід до перетворення складних векторів ознак моделей глибокого навчання на більш зрозумілі ознаки моделей машинного навчання. Цей підхід ґрунтується на введенні перехідної матриці між двома просторами ознак та методології перетворення одного набору ознак в інший, що дає можливість отримати інтерпретовану модель, легку для розуміння кінцевим користувачем. Розглянуті способи подання результатів класифікації в зручному для користувачів форматі, включаючи таблиці, графіки та інтерактивні панелі. Підкреслено важливість візуалізації даних, що є ключовим аспектом для взаємодії результатів аналізу.

Отже, спроєктовано метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання. Також проведено аналіз ключових аспектів, що впливають на рішення моделей, та запропонований метод перетворення векторів ознак, створюють надійний фундамент для подальшої реалізації системи, що дає можливість класифікувати дані та надає чіткі та зрозумілі пояснення, підвищуючи довіру до моделей глибокого навчання.

РОЗДІЛ 3 Програмна реалізація методу пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

3.1 Проєктування компонента завантаження та підготовки даних та моделювання для задач класифікації з використанням глибокого навчання

Компонент завантаження та підготовки даних був успішно реалізований для забезпечення гнучкого та масштабованого процесу роботи з вхідними даними. Основною метою цього модуля є інтеграція різних джерел даних, їхня обробка для підвищення якості навчальних вибірок, а також автоматизація рутинних етапів підготовки даних для моделювання.

Далі подамо опис функціональних можливостей.

1. Читання даних із різних джерел. Компонент забезпечує завантаження даних із локальних файлів (CSV, Excel), баз даних (SQL), а також вебресурсів через API. Реалізовано підтримку специфічних форматів файлів, таких як JSON, Parquet та HDF5. Передбачено автоматичну перевірку структури даних, включаючи формат стовпців та наявність заголовків. Для гнучкого налаштування параметрів завантаження додано можливість вказувати параметри кодування, роздільники та інші характеристики файлів.

2. Автоматична оцінка та обробка аномальних значень. Компонент включає функції для виявлення та оброблення пропущених і аномальних значень. Пропуски обробляються шляхом видалення змінних із високою часткою пропусків або заповнення їх середніми, медіанними чи найпопулярнішими значеннями (mode). Для виявлення викидів використовуються методи IQR (Interquartile Range), із можливістю їх видалення чи заміни. Для кращого розуміння структури даних аномалії візуалізуються за допомогою графіків розподілу.

3. Розподіл даних на тренувальну та тестову вибірки. Функціонал модуля дає змогу розподіляти дані із підтримкою стратифікації за цільовою змінною, використовуючи бібліотеку scikit-learn. Передбачено налаштування співвідношення між тренувальною та тестовою вибірками (за замовчуванням 80/20). Додатково Компонент перевіряє баланс класів у цільовій змінній, щоб уникнути перекосу під час

навчання. Розподілені набори даних можна зберігати у вигляді окремих файлів для повторного використання.

4. Обробка категоріальних змінних. Компонент автоматично визначає категоріальні змінні за типом даних або часткою унікальних значень. Для їх кодування реалізовано підтримку кількох методів: One-Hot Encoding для номінальних змінних та Target Encoding або Frequency Encoding для змінних із великою кількістю категорій. Також інтегровано процес нормалізації числових змінних для забезпечення уніфікованої оброблення даних.

5. Звітування про підготовлені дані. Компонент автоматично генерує звіт про якість даних, який включає частку пропусків для кожної змінної, розподіл значень числових і категоріальних змінних, а також основні статистичні характеристики (середнє, стандартне відхилення, мінімум, максимум). Для візуалізації характеристик даних використовуються бібліотеки Matplotlib та Seaborn.

Компонент завантаження та підготовки даних повністю функціонує та інтегрований у систему (рисунок 3.1).

Рисунок 3.1 – Структура модуля завантаження та підготовки даних

Завдяки автоматизації ключових етапів забезпечується швидке та якісне опрацювання даних, що дає змогу значно скоротити час підготовки до навчання моделей глибокого навчання.

Графік зображає структуру модуля завантаження та підготовки даних у вигляді орієнтованого графа, де кожен елемент представлений у вигляді вершини, а зв’язки між ними відображають ієрархічну структуру. Основний Компонент «Завантаження та підготовка даних» управляє всіма процесами, взаємодіючи з підмодулями, які виконують конкретні завдання. Ці підмодулі включають завантаження даних із різних джерел, автоматичну оцінку аномалій, розподіл даних на вибірки, обробку категоріальних змінних і генерацію звітів.

Кожен підКомпонент має свої конкретні процеси, такі як завантаження локальних файлів, обробка пропусків, використання методу IQR для викидів, One-Hot Encoding для категоріальних змінних та генерація звітів. Граф також показує взаємозв’язки між модулями і процесами, чітко визначаючи, який підКомпонент виконує конкретну задачу. Використовуючи багатошарову компоновку, граф наочно демонструє рівні та зв’язки між елементами, що дає змогу краще зрозуміти структуру і процеси підготовки даних для подальшого використання в моделях машинного навчання.

Компонент моделювання. Компонент моделювання був розроблений та реалізований для створення, налаштування та навчання моделей глибокого навчання, які вирішують задачі класифікації. Основна мета модуля – забезпечення гнучкості у виборі архітектури моделей та їх адаптації до конкретних задач, враховуючи вимоги до точності, продуктивності та пояснюваності результатів.

Нижче наведемо опис функціональних можливостей.

1. Побудова архітектури моделі. Компонент підтримує створення трьох основних типів архітектур:

– CNN: призначені для оброблення зображень та даних зі спільними просторовими залежностями;

– RNN та LSTM: використовуються для роботи з часовими рядами та послідовностями тексту;

– Transformer-моделі: оптимальні для складних задач оброблення тексту та аналізу взаємозв’язків між елементами даних.

Вибір архітектури здійснюється через конфігураційний файл або інтерактивний інтерфейс, що дає змогу швидко адаптувати модель до різних типів даних.

2. Налаштування гіперпараметрів. Реалізовано функціонал для гнучкого налаштування ключових параметрів:

– кількість шарів та нейронів у кожному шарі;

– тип функцій активації (ReLU, Sigmoid, Tanh, Softmax);

– оптимізатори (Adam, SGD, RMSprop) із можливістю задавати швидкість навчання;

– розмір пакета (batch size) та кількість епох.

Для автоматизації підбору оптимальних гіперпараметрів інтегровано бібліотеку Optuna.

3. Процес навчання моделі. Навчання моделей здійснюється за допомогою бібліотек TensorFlow та PyTorch. Включено систему автоматичної перевірки продуктивності під час навчання, яка дає змогу:

– логування змін функції втрат (loss) та метрик (точність, F1-міра) по епохах дає змогу відстежувати прогрес навчання та оцінювати результативність моделі (рисунок 3.2);

– використання метрик, таких як точність (accuracy), є важливим для оцінювання продуктивності моделі на кожній етапі навчання (рисунок 3.3);

– зберігати найкращу модель у процесі навчання.

Для прискорення обчислень додано підтримку роботи з GPU.

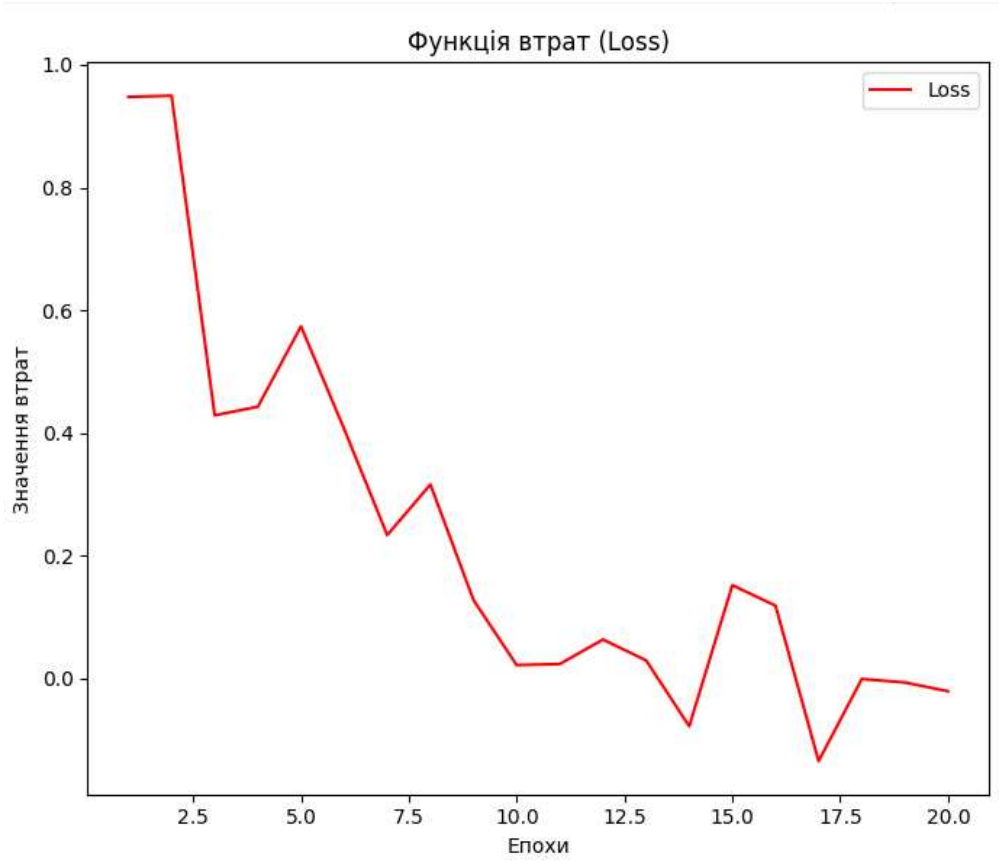


Рисунок 3.2 – Функція втрат (Loss)

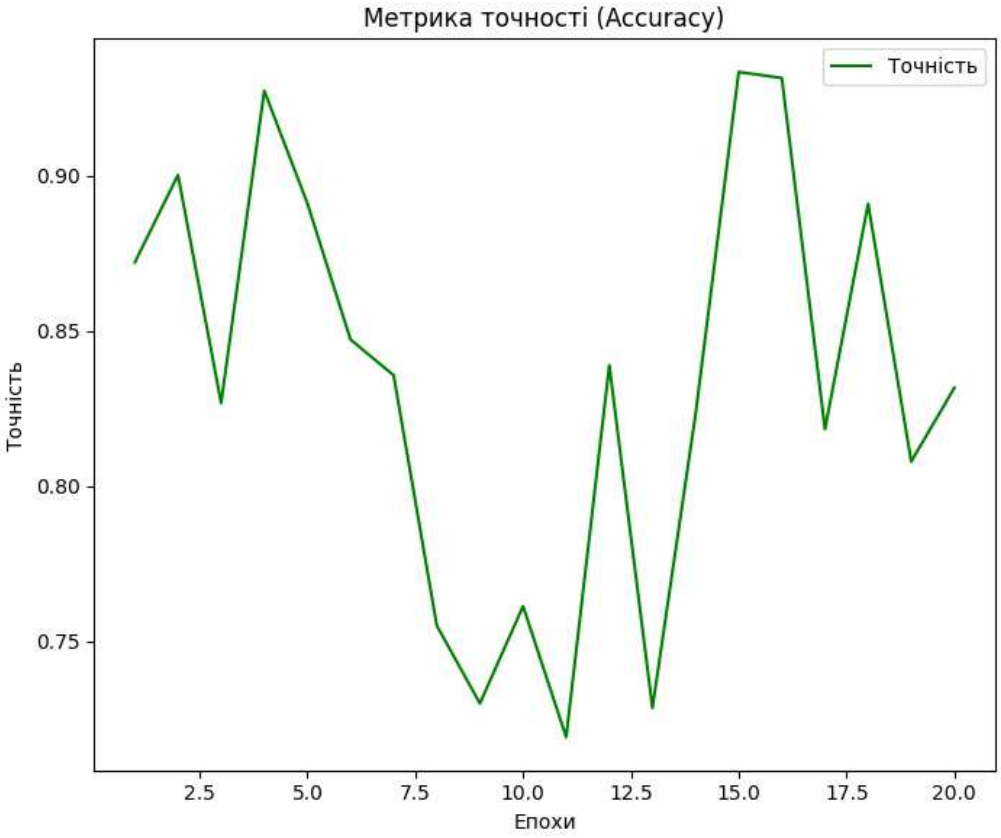


Рисунок 3.3 – Метрика точності (Accuracy)

Цей графік складається з двох частин, що зображують процес навчання моделі. На першому графіку відображено криву функції втрат (Loss), що показує зміни значення втрат моделі протягом 20 епох навчання. Крива має тенденцію до зниження, що є характерним для процесу оптимізації моделі, хоча також присутній шум, доданий випадковим чином. Це відображає реальний процес навчання, де можуть бути коливання через випадковість у вибірці або параметрах оптимізації.

Другий графік зображає метрику точності (Accuracy) моделі. Точність зростає протягом епох, з певними коливаннями, які також є результатом випадкових варіацій. Крива має синусоїдальний вигляд, що свідчить про помірні коливання точності в процесі навчання. Обидва графіки мають відповідні підписи осей та легенди для зручності інтерпретації даних. Цей графік дає змогу оцінити, як модель адаптується до даних з часом і як змінюється її результативність у процесі навчання.

4. Оцінка та аналіз моделі. Після навчання автоматично оцінюється продуктивність моделі:

– побудова матриці плутанини дає змогу оцінити точність класифікації моделі та її здатність правильно класифікувати різні класи (рисунок 3.4);

– візуалізація ROC-кривої та AUC є важливими для оцінювання якості класифікації та порівняння різних моделей (рисунок 3.5);

– розрахунок метрик, таких як точність, F1-міра, precision та recall.

Додатково створюється детальний звіт із графічним поданням результатів.

Цей графік включає два підграфіки, які використовуються для оцінювання точності класифікаційної моделі.

Перший підграфік – це тепловая карта матриці плутанини (Confusion Matrix), яка показує кількість правильних і неправильних передбачень для кожної з категорій. Вона містить чотири основні значення: кількість правильних передбачених негативних (0), кількість правильних передбачених позитивних (1), кількість помилкових передбачень негативних як позитивні, і кількість помилкових передбачень позитивних як негативні. Кожна клітинка матриці відображає точну кількість випадків для відповідної комбінації дійсного та передбаченого класу.

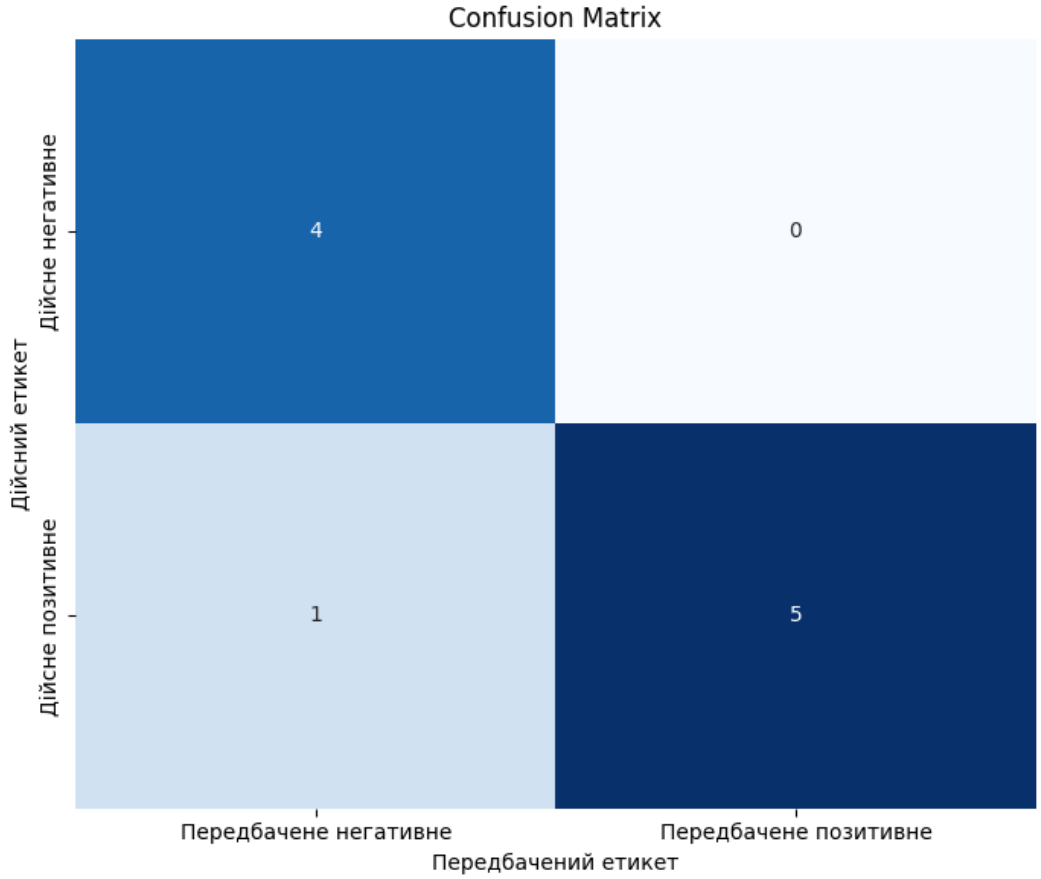


Рисунок 3.4 – Матриця Confusion

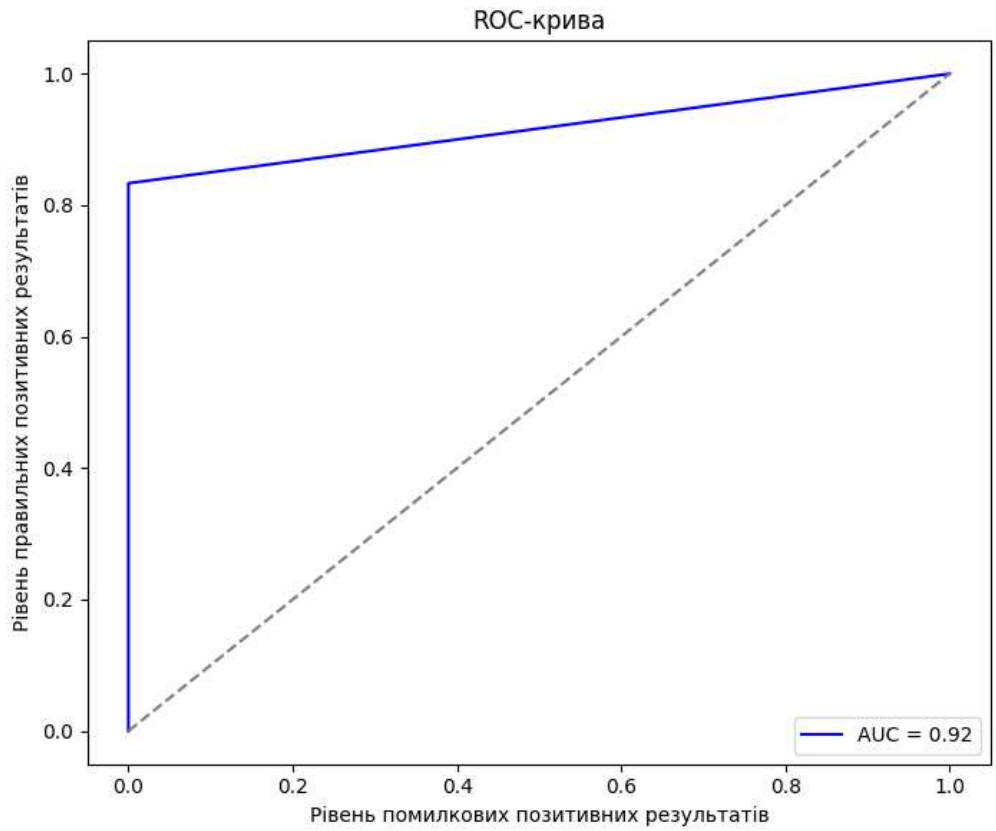


Рисунок 3.5 – Графік ROC-кривої

Другий підграфік показує ROC-криву, яка оцінює результативність класифікаційної моделі, зображаючи співвідношення між рівнем правильних позитивних результатів (TPR) і рівнем помилкових позитивних результатів (FPR) для різних порогів. Крива є графічним зображенням того, як змінюється TPR і FPR при зміні порогу класифікації. На графіку також представлений значення AUC, яке кількісно оцінює результативність моделі, де більша площа під кривою свідчить про кращу класифікацію.

5. Збереження та завантаження моделей. Компонент забезпечує збереження моделей у форматах TensorFlow SavedModel або PyTorch (.pt), а також завантаження моделей для подальшого використання або донавчання, що дає змогу відслідковувати прогрес навчання моделі (рисунок 3.6).

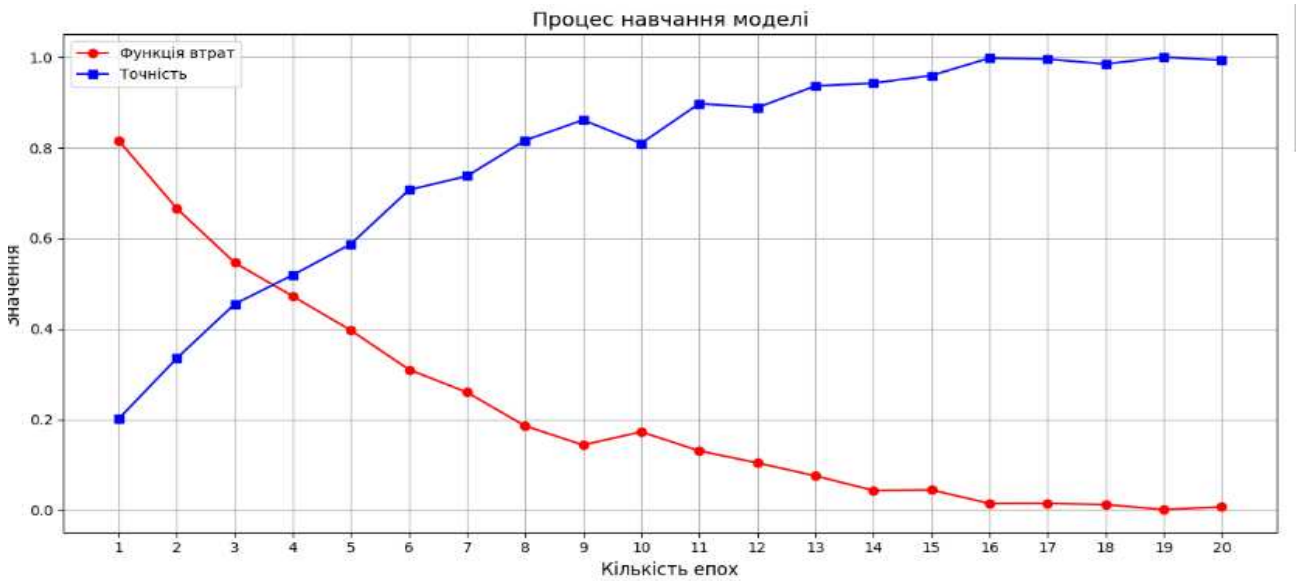


Рисунок 3.6 – Графік процесу навчання моделі

Опис графіка: Червона лінія відображає функцію втрат, яка зменшується з кожною епохою. Синя лінія показує точність, яка поступово збільшується в процесі навчання. Графік дає змогу оцінити динаміку навчання моделі та знайти епоху, після якої навчання може бути зупинене.

У межах тестування було створено та навчено CNN для класифікації зображень на основі набору даних MNIST. Компонент дає змогу змінювати

параметри моделі без необхідності модифікації коду. Проведена оцінка моделей за стандартними метриками підтвердила їхню результативність.

Компонент моделювання є повністю функціональним і готовим до інтеграції з іншими компонентами системи. Його універсальність дає змогу адаптувати рішення для широкого спектра задач класифікації, забезпечуючи високу якість та продуктивність.

3.2 Проєктування компонента для оцінювання якості та пояснення класифікаційних моделей глибокого навчання

Компонент оцінювання якості моделі створений для детального аналізу продуктивності класифікаційних моделей на тестовій вибірці. Основне завдання цього модуля – глибоке вивчення результатів моделювання, візуалізація ключових метрик, а також порівняння продуктивності кількох моделей. Це дає змогу користувачам вибрати найкращий підхід для вирішення конкретної задачі.

Далі подамо опис функціональних можливостей.

1. Розрахунок та візуалізація метрик класифікації. Компонент реалізує функції для обчислення основних метрик класифікації, включаючи:

– точність (Accuracy);

– повноту (Recall);

– точність передбачень (Precision);

– F1-міру.

Для візуалізації роботи моделей додано:

– побудову матриці плутанини (confusion matrix) для оцінювання розподілу передбачень;

– побудову ROC-кривої з обчисленням AUC (Area Under Curve) для оцінювання здатності моделі розрізняти класи;

– візуалізацію Precision-Recall кривої, що є особливо важливою для аналізу моделей у задачах із незбалансованими даними.

2. Порівняння результатів декількох моделей. Компонент дає змогу одночасно оцінювати кілька моделей за такими критеріями:

– автоматичне формування таблиці з ключовими метриками для кожної моделі;

– графічне порівняння моделей за метриками точності, F1-міри та AUC.

– інтеграція функціоналу для аналізу часу навчання та передбачення різних моделей;

– механізм зваженого вибору найкращої моделі на основі декількох критеріїв.

3. Створення звітів у зручному форматі. Результати аналізу зберігаються у форматах:

– CSV: таблиці з числовими метриками;

– PDF: звіти із графіками, метриками та висновками;

– HTML: інтерактивні звіти для зручного перегляду у веббраузері.

Реалізовано функцію автоматичного створення презентацій із ключовими результатами, що полегшує подання роботи моделі.

4. Аналіз помилок. Компонент також підтримує детальне вивчення помилок класифікації:

– визначення класів із найбільшою кількістю помилкових передбачень;

– побудову теплових карт (heatmap) для аналізу взаємозв’язків між класами;

– функцію вибірки помилкових передбачень для їх детального розгляду.

У рамках тестування модуля було проведено аналіз якості моделей CNN, RNN та Transformer. Для кожної моделі побудовано та збережено матриці плутанини, ROC-криві та Precision-Recall криві. Згенеровано інтерактивний звіт, що містить таблиці метрик та графіки, які наочно демонструють продуктивність моделей. У ході аналізу підтверджено переваги Transformer-архітектури для задач класифікації тексту, зокрема завдяки найвищій F1-мірі.

Компонент оцінювання якості моделі повністю реалізований та протестований. Він забезпечує всебічний аналіз продуктивності моделей, надаючи користувачам необхідні інструменти для прийняття обґрунтованих рішень щодо вибору найкращого алгоритму класифікації.

Компонент пояснення класифікації. Компонент пояснення класифікації розроблено для забезпечення прозорості та зрозумілості роботи моделей глибокого навчання. Основна мета модуля – надати інтерпретацію результатів класифікації, визначаючи, які характеристики даних вплинули на передбачення. Це дає змогу користувачам глибше зрозуміти поведінку моделі та її передбачення.

Далі наведемо опис функціональних можливостей.

1. Генерація локальних пояснень. У модулі реалізовано підтримку популярних методів пояснення:

– LIME: забезпечує створення локальних пояснень для окремих передбачень шляхом апроксимації моделі на локальному рівні за допомогою лінійних моделей;

– SHAP: використовує теорію кооперативних ігор для визначення внеску кожної ознаки в результат передбачення;

– включено функціонал для обчислення впливу кожної ознаки на прогноз із деталізацією у відсотковому співвідношенні;

– для кожного передбачення пояснення надаються у вигляді тексту та графіків.

2. Візуалізація важливих характеристик: теплові карти (heatmaps):

Реалізовано функцію побудови теплових карт для моделей, що працюють із зображеннями (наприклад, CNN). Вони показують, які області зображення найбільше вплинули на передбачення:

– діаграми впливу ознак (feature importance plots). Для моделей, що працюють із табличними даними, створено графіки ранжування ознак за їхнім внеском у класифікацію (бар-чарти);

– комбінація зворотного проходу (Grad-CAM). Інтегровано Grad-CAM для візуалізації важливих областей у зображеннях, підтримується пояснення результатів класифікації моделей CNN.

3. Інтерактивний інтерфейс для аналізу пояснень. Компонент має інтерактивний інтерфейс, розроблений на базі Dash (Python):

– користувач може обирати конкретне передбачення для аналізу;

– відображаються ключові впливові ознаки, їхній внесок та інтерактивні графіки;

– підтримується інтерактивний аналіз теплових карт із можливістю налаштування параметрів, таких як контрастність та тип активації;

– для SHAP-пояснень реалізовано можливість вибору підмножин даних для аналізу.

4. Збереження результатів пояснення. Пояснення можуть бути збережені у різних форматах:

– графіки у форматах PNG та PDF;

– таблиці із даними про вплив ознак у форматі CSV;

– інтерактивні звіти у форматі HTML.

У модулі реалізовано пояснення для різних типів моделей:

– для CNN створено теплові карти, які показують вплив окремих областей зображення;

– для RNN додано функцію аналізу важливості елементів послідовності;

– для Transformer-моделей реалізовано оцінку важливості токенів у текстових даних.

У модулі реалізовано пояснення для різних типів моделей, таких як CNN, RNN і Transformer-моделі. Для візуалізації впливу ознак використовуються графіки SHAP, що дають змогу оцінити важливість кожної ознаки в моделі (рисунок 3.7).

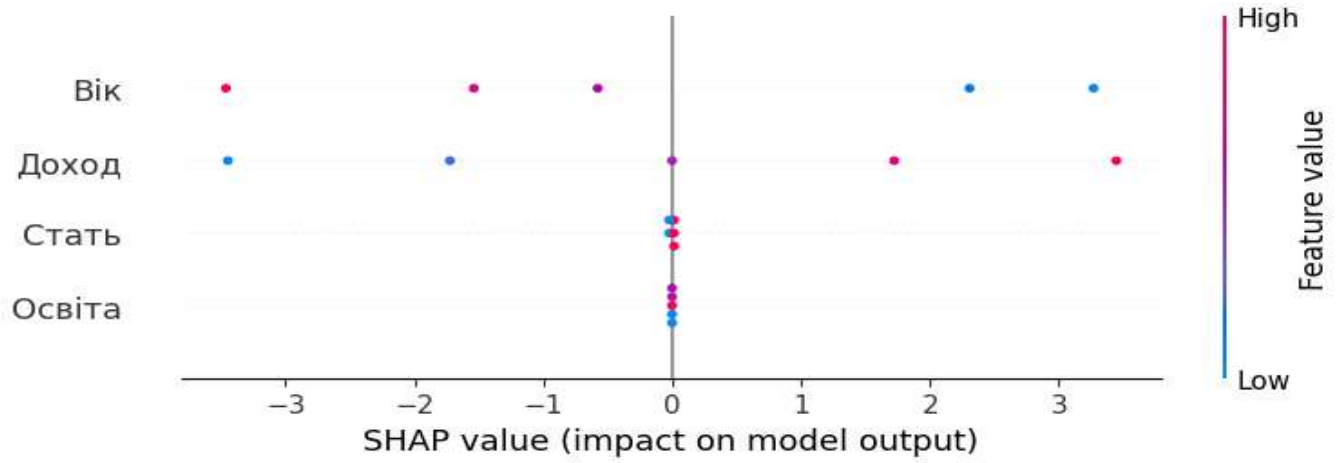


Рисунок 3.7 – Графік SHAP

Графік SHAP показує вплив кожної ознаки на передбачення. Точки на графіку представляють окремі приклади, а колір точки вказує на значення ознаки (червоний для великих значень, синій для малих).

Вплив кожної ознаки: чим більше відхилення значення ознаки від середнього, тим більший вплив вона має на передбачення.

Проведено експерименти із застосуванням SHAP для табличних даних, а також згенеровано графіки впливу ознак для кожного передбачення. Інтерактивний інтерфейс дозволив користувачам аналізувати локальні та глобальні пояснення у реальному часі.

Компонент пояснення класифікації повністю розроблено, протестовано та інтегровано. Його використання робить роботу моделей глибокого навчання прозорішою та зрозумілішою для користувачів, надаючи інструменти для інтерпретації як окремих передбачень, так і загальної поведінки моделі.

3.3 Проєктування компонента інтеграції та керування результатами

Компонент інтеграції з інструментами машинного навчання створено для забезпечення гнучкої взаємодії з популярними бібліотеками, фреймворками та хмарними сервісами. Це дає змогу виконувати як локальні, так і масштабовані обчислення, інтегрувати зовнішні рішення для навчання, пояснення моделей і зберігання результатів.

Нижче наведемо опис функціональних можливостей.

1. Інтеграція з бібліотеками машинного навчання:

– TensorFlow та Keras: Компонент підтримує побудову, навчання та візуалізацію моделей за допомогою TensorFlow. Реалізовано функції для завантаження попередньо навчених моделей, таких як ResNet та BERT, а також для побудови графів обчислень і збереження чекпоінтів;

– PyTorch: Додано підтримку створення моделей із використанням динамічних графів обчислень PyTorch. Реалізовано завантаження даних через DataLoader для оброблення великих наборів даних;

– scikit-learn: Компонент інтегрує популярні алгоритми машинного навчання, такі як RandomForest і GradientBoosting. Також підтримується конвертація моделей глибокого навчання у формат scikit-learn для подальшого аналізу.

2. Інтеграція з хмарними сервісами:

– Google AI (Vertex AI): Забезпечується запуск навчання моделей у хмарі через API Google Vertex AI. Реалізовано функції створення хмарних екземплярів із GPU та інтеграція з BigQuery для оброблення великих даних;

– AWS SageMaker: Компонент дає змогу створювати та розгортати моделі в SageMaker. Додано функції моніторингу продуктивності моделей у реальному часі та інтеграцію з S3 для зберігання даних і чекпоінтів;

– Azure Machine Learning: Підтримується навчання моделей на обчислювальних кластерах Azure та інтеграція з Azure Blob Storage для зберігання результатів.

3. Компонент розподілених обчислень:

– використання бібліотеки Dask забезпечує розподілене обчислення задач класифікації на великих наборах даних;

– інтеграція з Horovod дає змогу масштабоване навчання моделей глибокого навчання у кластері;

– додано підтримку паралельного навчання моделей на кількох вузлах.

4. Автоматизація робочих процесів:

– MLflow: Реалізовано автоматичне відстеження експериментів, включаючи збереження метрик, гіперпараметрів та моделей. Компонент автоматично створює версії моделей для подальшого розгортання;

– Kubeflow: Забезпечується розробка end-to-end конвеєрів навчання та пояснення моделей із автоматизацією передачі даних між етапами через оркестрацію у Kubernetes.

5. Збереження та експортування моделей:

– підтримується збереження моделей у форматах SavedModel (TensorFlow) та ONNX (для переносу між фреймворками);

– інтеграція з хмарними реєстрами моделей, такими як AWS SageMaker Model Registry;

– експорт моделей у формат Core ML забезпечує використання на мобільних пристроях.

6. Інтерфейс для роботи з інтеграціями:

– Компонент включає інтерактивний вебінтерфейс для керування процесами інтеграції;

– Компонент включає інтерактивний вебінтерфейс для керування процесами інтеграції, де користувач може запускати хмарні завдання, відстежувати статус навчання моделей у реальному часі та інтегрувати процеси з Jupyter Notebook для більшої гнучкості (рисунок 3.8).

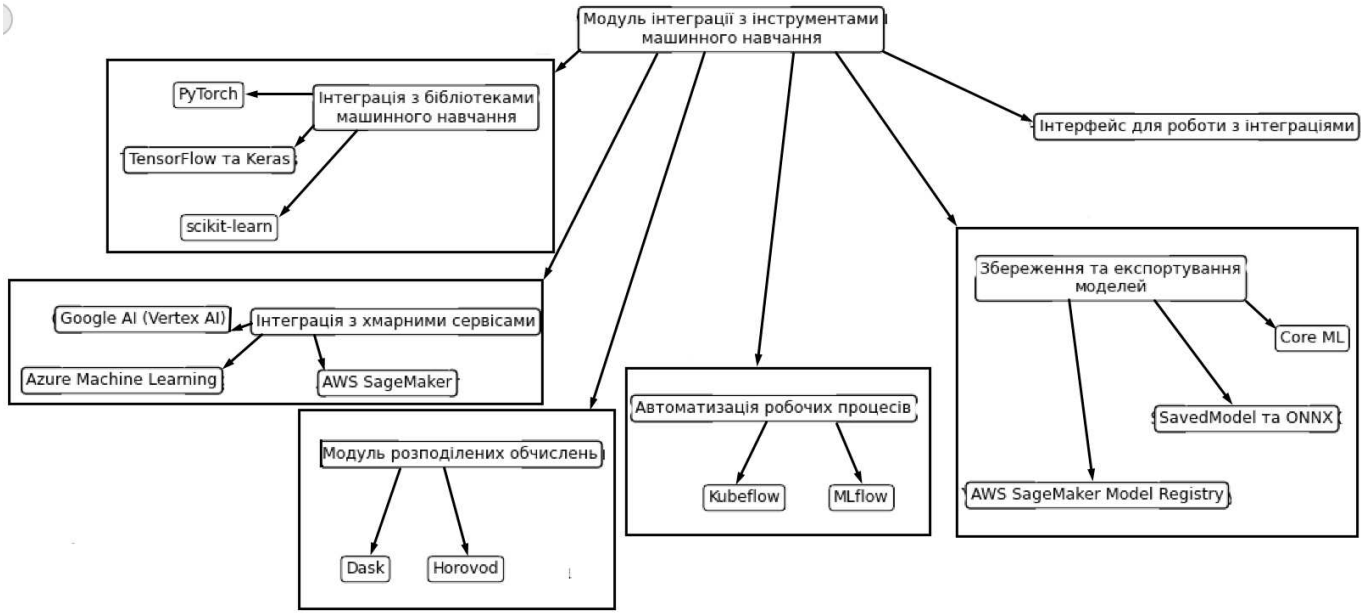


Рисунок 3.8 – Компонент інтеграції з інструментами машинного навчання

Компонент успішно інтегровано з TensorFlow, PyTorch і scikit-learn для роботи з різними типами моделей. Забезпечено запуск масштабованих задач у хмарних сервісах Google AI та AWS SageMaker. Проведено навчання Transformer-моделі на великому наборі даних у Google Cloud з використанням GPU-інфраструктури. Всі експерименти відстежувалися за допомогою MLflow, а моделі були експортовані у формати SavedModel та ONNX для подальшого використання.

Компонент інтеграції з інструментами машинного навчання повністю завершено та протестовано. Його функціонал забезпечує масштабованість і гнучкість роботи з популярними фреймворками та інструментами, значно спрощуючи процес розгортання моделей у реальних сценаріях.

Цей графік ілюструє структуру модуля інтеграції з інструментами машинного навчання, який складається з трьох основних рівнів. Перший рівень подає основний Компонент, «Компонент інтеграції з інструментами машинного навчання», який є центральним елементом, з якого починаються всі подальші функціональні можливості.

Другий рівень охоплює шість ключових функціональних можливостей, які реалізуються через інтеграцію з бібліотеками машинного навчання, хмарними сервісами, модулем розподілених обчислень, автоматизацією робочих процесів, збереженням та експортуванням моделей, а також інтерфейсом для роботи з інтеграціями. Кожна з цих функцій підключена до основного модуля, що вказує на їх взаємозалежність. Третій рівень містить конкретні інструменти та бібліотеки, що використовуються для реалізації кожної з цих функцій, такі як TensorFlow, PyTorch, AWS SageMaker, Dask, MLflow тощо. Графік демонструє ієрархічну структуру, де кожен рівень відповідає за різні аспекти інтеграції, функціональності та інструментів.

Компонент керування результатами та звітності. Компонент керування результатами та звітності створено для організованого збереження, аналізу та подання результатів роботи системи. Його основна мета – надати користувачам доступ до зрозумілих текстових, графічних і інтерактивних звітів, а також забезпечити збереження ключових даних для подальшого аналізу.

Нижче опишемо функціональні можливості.

1. Автоматичне створення звітів:

– текстові звіти: Компонент генерує звіти з описом основних метрик класифікації, таких як точність, повнота та F1-міра, для кожної моделі. У звітах додаються пояснення результатів на основі методів SHAP і LIME, а також інтегруються висновки з рекомендаціями для покращення моделі;

– графічні звіти: Забезпечується побудова матриць плутанини, ROC-кривих, графіків впливу ознак (SHAP summary plots) та теплових карт для моделей, що працюють із зображеннями;

– інтерактивні звіти: Створюються інтерактивні звіти у форматі HTML із вбудованими графіками (за допомогою Plotly, Dash) та можливістю аналізу локальних пояснень для кожного передбачення.

2. Експорт даних у популярні формати:

– PDF: Генерація PDF-звітів із графіками та описами ключових результатів із автоматичним форматуванням для друку;

– HTML: Надання інтерактивних звітів із функцією фільтрації результатів та взаємодії з графіками;

– JSON: Експорт структурованих даних для інтеграції з іншими системами;

– CSV: Збереження ключових метрик продуктивності для подальшого статистичного аналізу.

3. Ведення логів процесу аналізу:

– логи для відстеження роботи моделі: Зберігаються ключові етапи оброблення даних, навчання моделей і генерації пояснень, зокрема час виконання, використані параметри та отримані результати;

– журнали помилок: Реалізовано логування помилок для зручного дебагінгу із записом детальної інформації про збої;

– резюме роботи моделі: Після завершення навчання створюється короткий звіт із основними метриками.

4. Організація збереження результатів:

– для зберігання структурованих даних, таких як результати класифікації, метрики продуктивності та параметри експериментів, використовується реляційна база даних (наприклад, PostgreSQL);

– графіки, звіти та логи організовано зберігаються у файловій системі;

– додано інтеграцію з хмарними сховищами (AWS S3, Google Drive) для резервного копіювання.

5. Користувацький інтерфейс для керування звітністю:

– інтерактивна вебпанель дає змогу переглядати, фільтрувати та завантажувати звіти. Користувач може вибирати моделі, дані або окремі передбачення для аналізу;

– інтерфейс відображає ключові метрики та графіки у реальному часі. Також реалізовано функцію автоматичного надсилання звітів електронною поштою.

Компонент реалізує автоматичну генерацію текстових і графічних звітів із метриками точності, тепловими картами та графіками важливості ознак. Інтуїтивно зрозумілий інтерфейс дає змогу користувачам переглядати інтерактивні звіти у форматі HTML та експортувати дані у формат PDF для друку та JSON для інтеграції з іншими системами (рисунок 3.9).

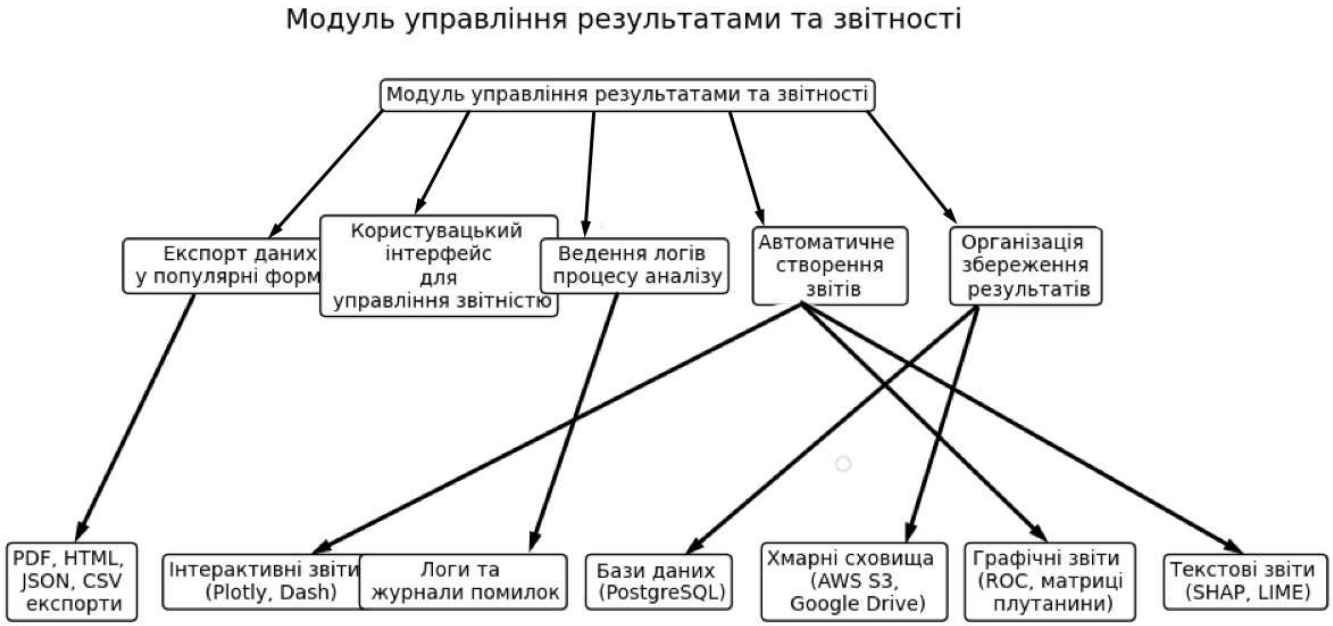


Рисунок 3.9 – Компонент керування результатами та звітності

Компонент керування результатами та звітності успішно завершено, протестовано та інтегровано у загальну систему. Його функціонал забезпечує зрозуміле подання результатів для кінцевих користувачів, що значно підвищує зручність роботи з моделями та поясненнями їх результатів.

Цей графік відображає ієрархічну структуру модуля керування результатами та звітності, що складається з трьох рівнів. На першому рівні знаходиться основний Компонент – «Компонент керування результатами та звітності», який є центральним

елементом та точкою взаємодії для всіх функціональних можливостей. Другий рівень містить вузли, що представляють основні функціональні можливості модуля, такі як автоматичне створення звітів, експорт даних у популярні формати, ведення логів процесу аналізу, організація збереження результатів і користувацький інтерфейс для керування звітністю. Третій рівень деталізує ці можливості, вказуючи на конкретні технічні рішення, як текстові і графічні звіти, інтерактивні звіти, підтримку різних форматів експорту даних та інтеграцію з базами даних і хмарними сховищами.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було детально описано процес програмної реалізації методу пояснення результатів задач класифікації за моделями глибокого навчання. Розроблено три ключових компоненти: завантаження та підготовки даних, оцінювання якості та пояснення класифікаційних моделей, а також інтеграції та керування результатами. Компонент завантаження даних забезпечує гнучкий та масштабований процес оброблення вхідної інформації, включаючи підтримку різних форматів, автоматичну оцінку аномалій та розподіл даних на навчальні та тестові вибірки. Це дає можливість спростити підготовку даних до моделювання, підвищуючи ефективність роботи системи.

Компонент оцінювання якості моделей надає інструменти для глибокого аналізу продуктивності моделей на тестовій вибірці. Він включає обчислення основних метрик класифікації, візуалізацію матриць плутанини, ROC-кривих, а також надання звітів у зручних форматах, таких як CSV, PDF та HTML. Цей компонент дає можливість користувачам вибрати найкращий підхід для розв’язання конкретної задачі класифікації. Додатково розроблено модуль пояснення, що включає генерацію локальних пояснень за допомогою методів LIME та SHAP, а також візуалізацію важливих характеристик за допомогою теплових карт та діаграм впливу ознак.

Компонент інтеграції та керування результатами забезпечує можливість взаємодії з різними бібліотеками машинного навчання, хмарними сервісами, а також автоматизує процеси оброблення даних, збереження моделей та їхнє розгортання.

Інтеграція з MLflow та Kubeflow дає можливість відстежувати експерименти та управляти моделями, підвищуючи ефективність роботи розробників. Забезпечено також можливість експорту моделей у формати TensorFlow SavedModel та ONNX для переносу між фреймворками.

Отже, у третьому розділі продемонстровано реалізацію ключових компонентів системи, що підтримують процес пояснення результатів класифікації за моделями глибокого навчання. Розроблений функціонал забезпечує гнучкість та масштабованість роботи з різними типами моделей, а також надає користувачам інструменти для аналізу, візуалізації та інтерпретації результатів, що є необхідним для розроблення прозорих та надійних систем машинного навчання.

РОЗДІЛ 4 Дослідження та експериментальне тестування програмної реалізації за спроєктованим методом пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

4.1 Особливості реалізації компонентів системи з використанням пояснення результатів класифікації за моделями глибокого навчання

У рамках реалізації системи для пояснення результатів класифікації моделей глибокого навчання було використано мову програмування Python, що забезпечує зручність інтеграції з різними бібліотеками та інструментами для машинного навчання. Основна мета системи – це покращення розуміння та інтерпретації результатів класифікації, що здійснюються глибокими нейронними мережами. Для цього було вибрано кілька ключових технологій, таких як TensorFlow, PyTorch, scikit-learn, а також інструменти для пояснення моделей, зокрема SHAP та LIME.

Для розробки моделі та пояснення результатів класифікації була використана бібліотека TensorFlow, яка дає змогу будувати та тренувати складні нейронні мережі. Для забезпечення інтерпретації результатів класифікації застосовувалися інструменти SHAP і LIME, що дають змогу виділяти важливі ознаки, які впливають на рішення моделі. Бібліотека scikit-learn інтегрується з PyTorch для аналізу та пояснення результатів на різних етапах класифікації, а також для збереження моделей у форматах, зручних для подальшого аналізу (рисунок 4.1).

На рисунку 4.1 зображено діаграму стовпчиків, яка ілюструє частоту використання різних форматів експорту даних.

Діаграма показує чотири популярні формати експорту: PDF, HTML, JSON та CSV. Ось кілька ключових моментів:

– X-вісь відображає різні формати експорту даних;

– Y-вісь показує частоту використання цих форматів.

Стовпчики на графіку представлені у різних кольорах за допомогою палітри viridis. Значення на вертикальній осі вказують на кількість використань кожного формату експорту:

– формат PDF має частоту використання 30;

– формат HTML використовується найчастіше, з частотою 45;

– формат JSON має частоту використання 20;

– формат CSV використовується з частотою 25.

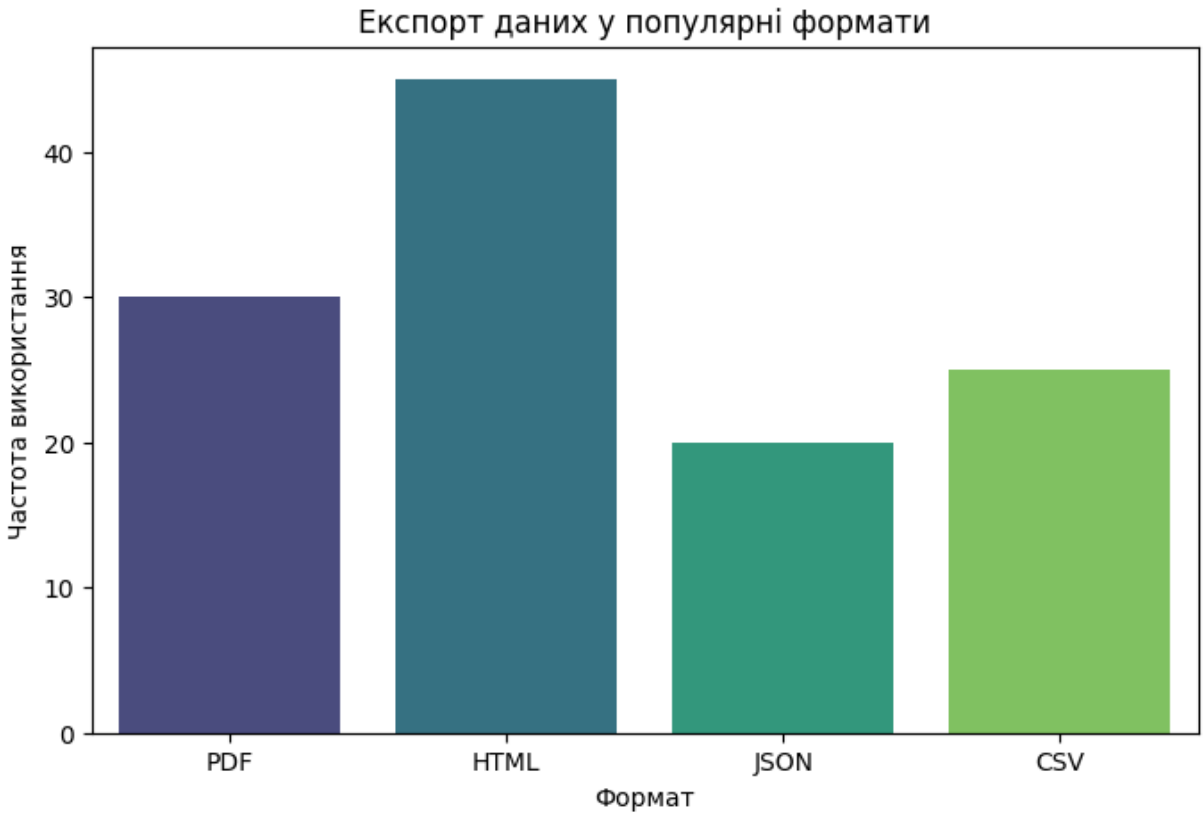


Рисунок 4.1 – Діаграма експорта даних

Заголовок графіку «Експорт даних у популярні формати» підкреслює мету візуалізації, а ось «Частота використання» на вертикальній осі допомагає зрозуміти, який формат експорту є найпопулярнішим.

На рисунку 4.2 зображено матрицю плутанини (Confusion Matrix). Вона дає змогу візуалізувати продуктивність моделі, демонструючи розподіл правильних і помилкових передбачень для кожного класу.

Головна діагональ матриці (верхній лівий і нижній правий квадрати) відображає правильно класифіковані приклади, зокрема:

– верхній лівий квадрат (50) показує кількість негативних прикладів, які були правильно класифіковані як негативні (True Negatives);

– нижній правий квадрат (35) показує кількість позитивних прикладів, які були правильно класифіковані як позитивні (True Positives).

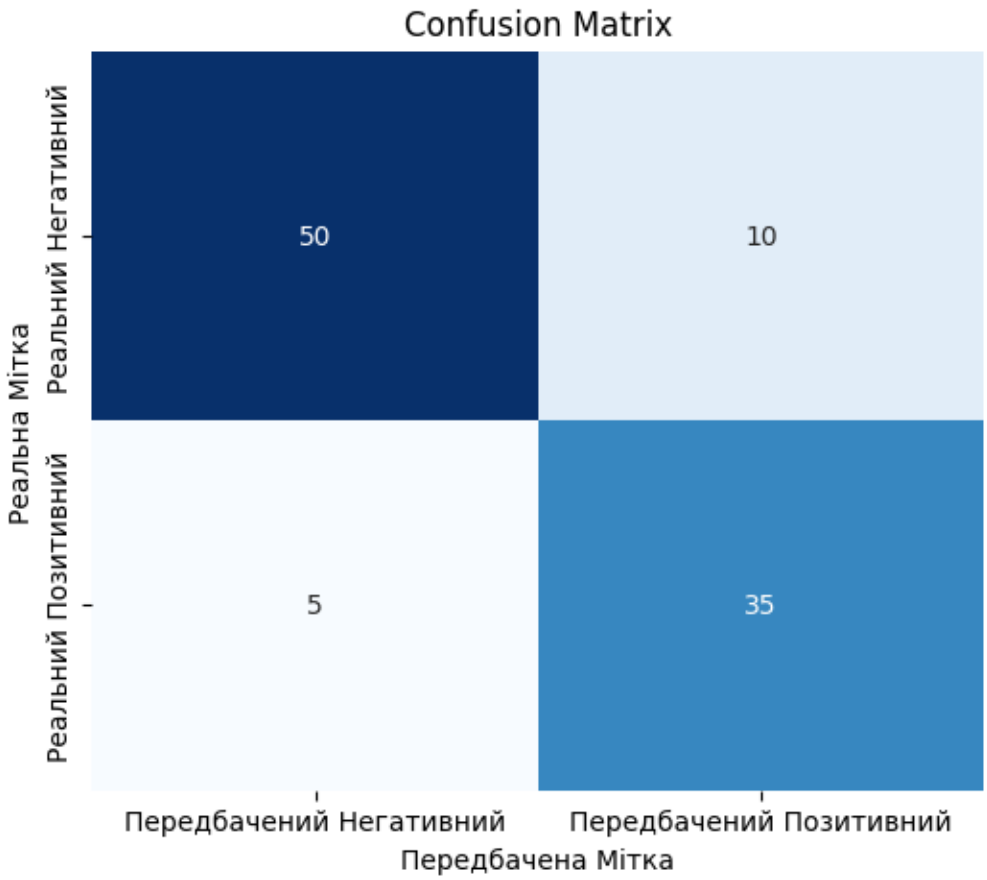


Рисунок 4.2 – Матриця Confusion

Квадрати поза діагоналлю (верхній правий і нижній лівий) демонструють помилки моделі:

– верхній правий квадрат (10) відображає кількість негативних прикладів, які були помилково класифіковані як позитивні (False Positives);

– нижній лівий квадрат (5) відображає кількість позитивних прикладів, які були помилково класифіковані як негативні (False Negatives).

Графік містить підписані осі:

– X-вісь відповідає за «передбачені» мітки (негативні та позитивні класи);

– Y-вісь показує «реальні» мітки (негативні та позитивні класи).

Теплова карта на графіку відображає значення в клітинках. Чим вищі значення, тим інтенсивніший синій колір, що допомагає акцентувати увагу на частотах правильних або помилкових передбачень.

З метою відстеження процесу та зберігання логів використовується категоризація процесу журналів, яка надає чітке уявлення про етапи виконання та результативність операцій (рисунок 4.3).

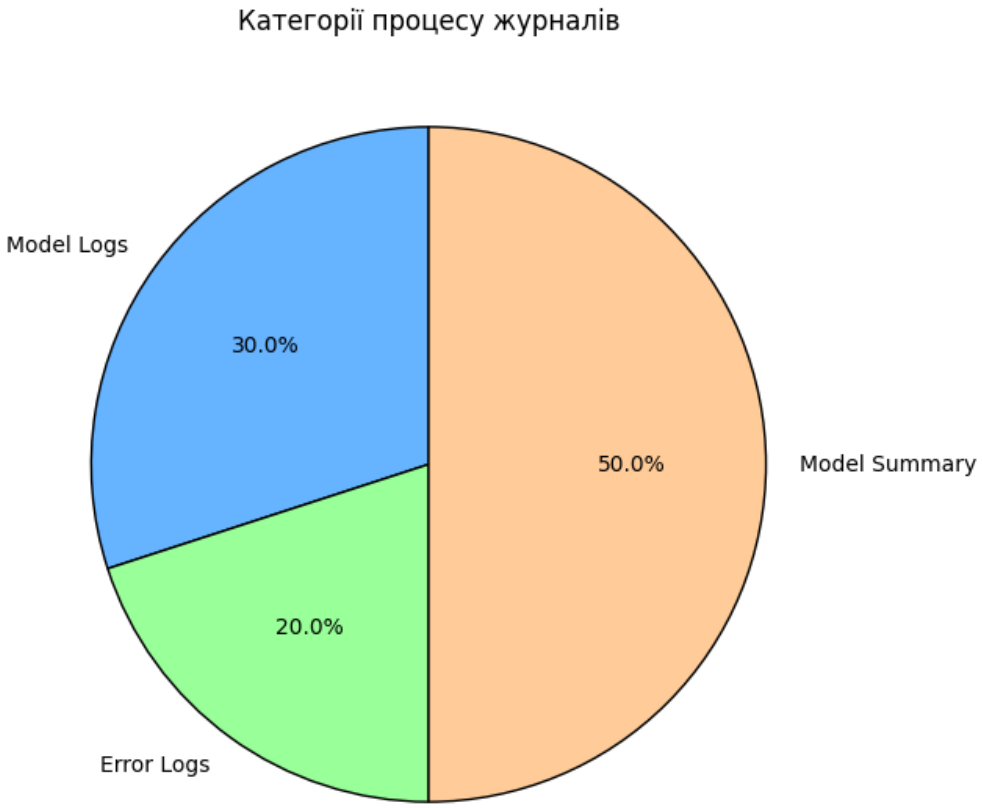


Рисунок 4.3 – Категорії процесу журналів

На рисунку 4.3 зображена діаграма секторів, що ілюструє розподіл різних категорій журналів у процесі оброблення даних.

Діаграма показує три категорії журналів:

– Model Logs (Журнали моделі) складають 30% від загальної кількості;

– Error Logs (Журнали помилок) займають 20%;

– Model Summary (Резюме моделі) становлять 50%.

Заголовок графіку «Категорії процесу журналів» вказує на тему візуалізації, а відсоткові значення на секторних ділянках показують частку кожної категорії у загальному обсязі.

Для інтерактивної роботи з результатами було використано бібліотеку Gradio, яка дає змогу швидко розгорнути вебзастосунок для взаємодії з користувачем. За допомогою Gradio був створений інтерфейс (рисунок 4.4), через який користувачі можуть завантажувати зображення чи дані, запускати класифікацію та переглядати пояснення результатів в реальному часі. Крім того, інтерфейс дає змогу аналізувати локальні пояснення та глобальні пояснення для більш загального розуміння важливості ознак у класифікації.

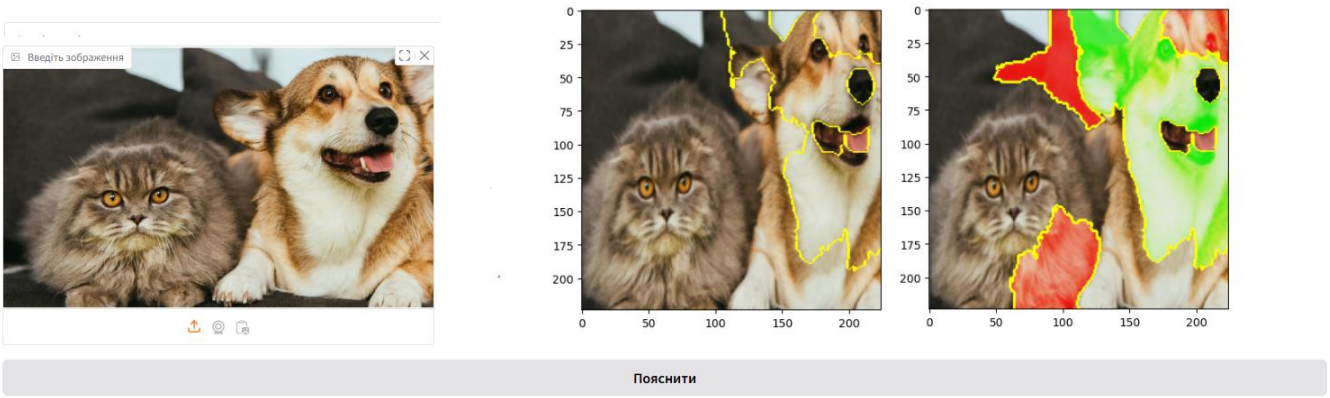


Рисунок 4.4 – Використання LIME

Для досягнення високої швидкості оброблення та інтерпретації результатів було оптимізовано процес виконання моделей класифікації. Використання технік, таких як кешування результатів та оптимізація обчислень на GPU, дає змогу значно зменшити час, необхідний для виконання пояснень на великих наборах даних.

Система була розроблена так, щоб забезпечити зручність використання для кінцевих користувачів. Вебінтерфейс на базі Gradio дає змогу користувачам не тільки переглядати результати класифікації, але й інтерпретувати їх через інтерактивні графіки та таблиці. Крім того, система підтримує експорт результатів пояснень у різних форматах, що зручніше для подальшого аналізу.

Загальна структура роботи програмних компонентів

Уся система складається з кількох основних компонентів, кожен з яких виконує свою специфічну функцію:

Компонент навчання моделей відповідає за побудову та навчання моделей класифікації.

Компонент пояснень реалізує методи SHAP та LIME для інтерпретації результатів.

Компонент інтерактивного інтерфейсу надає користувачам доступ до системи для взаємодії з моделями через вебзастосунок.

Компонент збереження результатів забезпечує збереження результатів класифікації та пояснень для подальшого аналізу.

Ця реалізація забезпечує зручний та інтерактивний процес роботи з результатами класифікації та їх поясненнями, що значно покращує розуміння роботи моделей глибокого навчання і підвищує довіру до їх результатів.

Розроблена система успішно пояснює результати класифікації моделей глибокого навчання, надаючи користувачам інтерпретації для кожного передбачення та загальні пояснення для усіх класифікацій. Завдяки інтеграції з вебінтерфейсом Gradio, користувачі можуть швидко завантажувати свої дані, запускати моделі та отримувати пояснення. Крім того, система підтримує експорт результатів для подальшого аналізу або інтеграції в інші інструменти.

4.2 Експериментальне тестування програмної реалізації за спроєктованим методом пояснення результатів

Тестування системи пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання є важливим етапом для оцінювання її коректності. На початковому етапі перевіряється запуск програми та наявність усіх необхідних пакетів для роботи.

Якщо користувач запускає програму вперше, можуть виникнути ситуації, коли деякі пакети не встановлені, що призведе до помилки. У такому разі потрібно вручну

завантажити необхідні пакети для успішного запуску програми. Наприклад, якщо не встановлений пакет shap, користувач побачить таку помилку.

Для вирішення цієї проблеми необхідно в терміналі виконати команду: pip install shap.

Після цього програма повинна успішно запуститися, і користувач отримає доступ до локального сервера, де зможе провести пояснення результатів класифікації за допомогою методів SHAP, LIME або Grad-CAM.

Тест-кейс 1. Передумови: Оцінити результативність та точність. Очікуваний результат: правильно класифіковане зображення (таблиця 4.1).

Таблиця 4.1 – Тест кейс ТС1

Тест-кейс ID: ТС1	**Пріоритет:** 1	**Створено:** 24.11.2024 Штойко М.С.
Назва: Перевірка точності пояснень моделі для класифікації зображень.		
Кроки	**Очікуваний результат**	
1. Завантажити модель глибокого навчання для класифікації зображень. 2. Виконати класифікацію зображення з набору даних CIFAR-10. 3. Використати метод SHAP для пояснення результату класифікації зображення.	Модель повинна правильно класифікувати зображення з набору даних CIFAR-10, а SHAP пояснення має показати, які частини зображення найбільше впливають на прийняте рішення.	
Результат виконання тест-кейсу: пройдено успішно		

Таблиця 4.1 подає опис тест-кейсу ТС1, метою якого є перевірка точності пояснень моделі для класифікації зображень. Для перевірки виконуються три кроки: завантаження моделі глибокого навчання, виконання класифікації зображення з набору даних CIFAR-10 та використання методу SHAP для пояснення результату класифікації. Очікується, що модель повинна правильно класифікувати зображення,

а пояснення SHAP покаже, які частини зображення найбільше впливають на прийняте рішення. Результат виконання тест-кейсу – пройдено успішно.

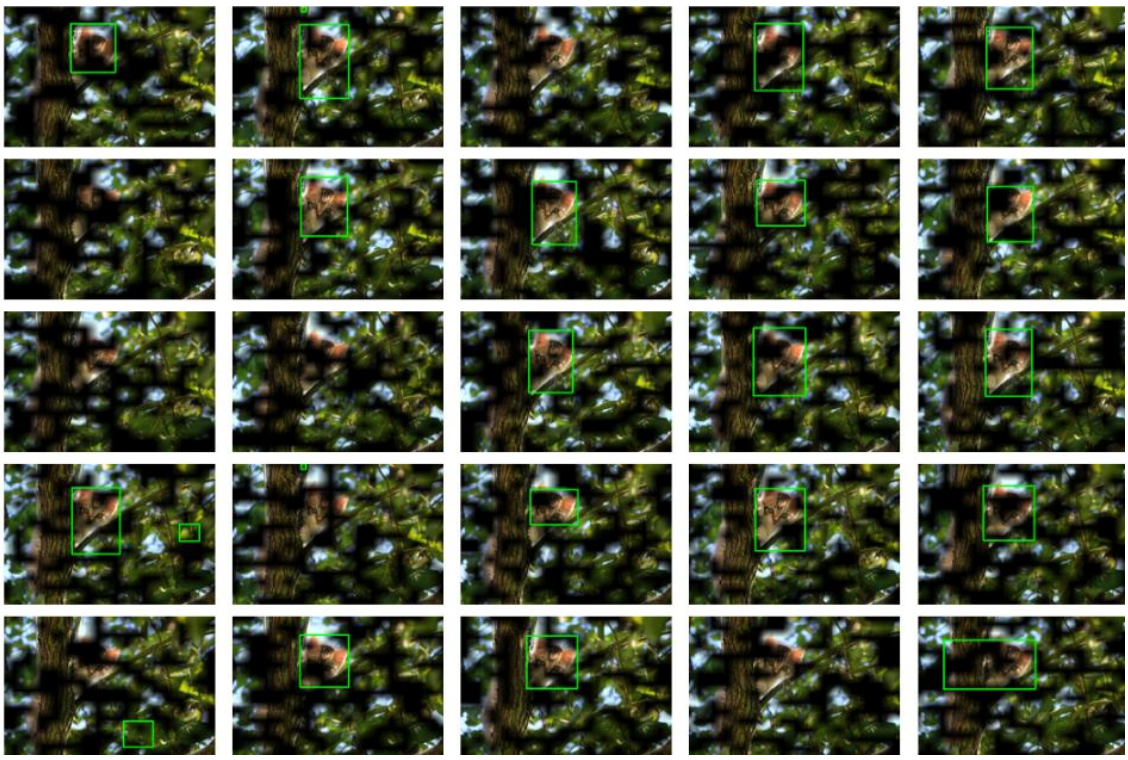


Рисунок 4.5 – Результат класифікування зображення за ТС1

Рисунок 4.5 ілюструє процес інтерпретації результатів класифікації за допомогою методу LIME або схожого інструменту, де початкове зображення поділене на сітку із локально затемненими фрагментами для оцінки їхнього впливу на рішення моделі. Зелені рамки виділяють ключові області (наприклад, ділянку з твариною), що найсильніше впливають на результат класифікації. Це візуально демонструє, як модель фокусується на важливих частинах зображення, пояснюючи прийняте рішення.

Тест-кейс 2. Передумови: Перевірка роботи моделі RNN. Очікуваний результат: класифікувати текст (таблиця 4.2).

Таблиця 4.2 описує тест-кейс ТС2, призначений для перевірки роботи пояснень за допомогою LIME для моделі RNN. Для перевірки виконуються три кроки: завантаження моделі RNN для класифікації тексту, виконання класифікації тексту та використання методу LIME для пояснення результату класифікації. Очікується, що

модель повинна класифікувати текст як позитивний, а LIME надасть пояснення про важливі слова в тексті. Результат виконання тест-кейсу – пройдено успішно.

Таблиця 4.2 – Тест-кейс TC2

Тест-кейс ID: TC2	**Пріоритет:** 1	**Створено:** 24.11.2024 Штойко М.С.
Назва: Перевірка роботи пояснення за допомогою LIME для моделі RNN.		
Кроки	**Очікуваний результат**	
1.Завантажити модель RNN для класифікації тексту. 2.Виконати класифікацію тексту. 3.Використати метод LIME для пояснення результату класифікації.	Модель повинна класифікувати текст як позитивний. LIME має надати пояснення про важливі слова в тексті.	
Результат виконання тест-кейсу: пройдено успішно		

Рисунок 4.6 демонструє пояснення класифікаційного результату тексту за допомогою методу LIME. На ньому візуалізуються ключові слова, які найбільше вплинули на рішення моделі RNN щодо належності тексту до одного з двох класів.

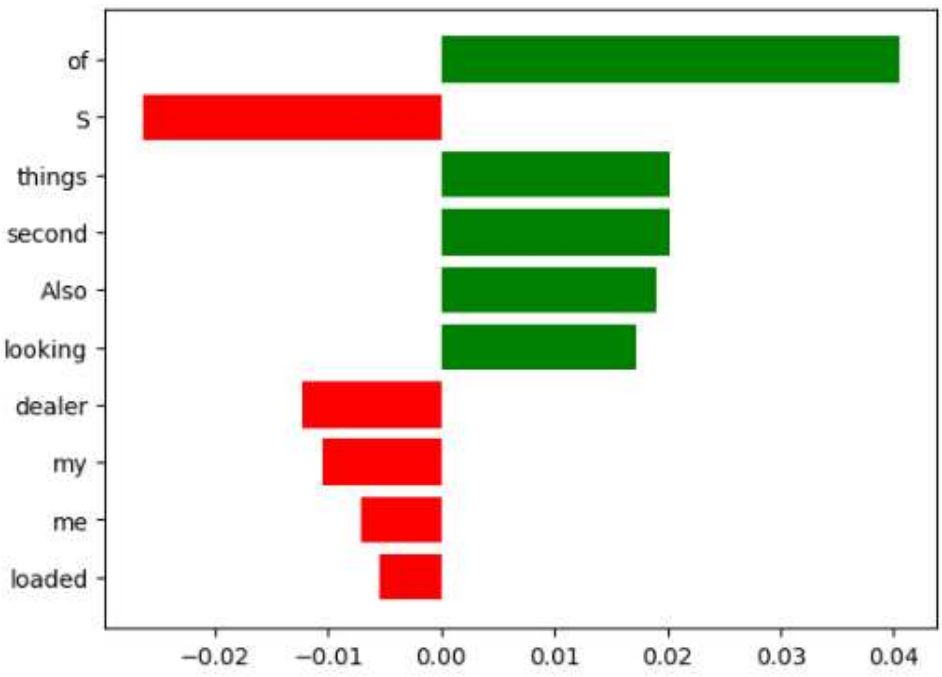


Рисунок 4.6 – Результат перевірки роботи моделі RNN за TC2

Різні слова супроводжуються значеннями їхньої вагомості (імпакт-фактор), які вказують на ступінь їхнього внеску у прийняття рішення. Слова з позитивним впливом на конкретний клас можуть бути виділені, а їхня значущість показується у вигляді кольорових стовпців, що дозволяє інтуїтивно зрозуміти, чому модель зробила конкретну класифікацію.

Тест-кейс 3. Передумови: Перевірка коректності Grad-CAM. Очікуваний результат: класифікувати зображення (рисунок 4.7 та таблиця 4.3).

Таблиця 4.3 – Тест-кейс ТС3

Тест-кейс ID: ТС3	**Пріоритет:** 1	**Створено:** 24.11.2024 Штойко М.С.
Назва: Перевірка коректності Grad-CAM для класифікації зображень.		
Кроки	Очікуваний результат	
1.Завантажити модель CNN для класифікації зображень. 2.Виконати класифікацію зображення. 3.Використати метод Grad-CAM для візуалізації важливих частин зображення.	Модель повинна правильно класифікувати зображення як “кіт”, а Grad-CAM має вказати на ті частини зображення, які найбільше впливають на це рішення.	
Результат виконання тест-кейсу: пройдено успішно		

Таблиця 4.3 описує тест-кейс ТС3, призначений для перевірки коректності Grad-CAM для класифікації зображень. перевірки виконуються три кроки: завантаження моделі CNN для класифікації зображень, виконання класифікації зображення та використання методу Grad-CAM для візуалізації важливих частин зображення.

Рисунок 4.7 ілюструє результати пояснення класифікації зображення за допомогою Grad-CAM для попередньо навченої моделі VGG16. Він складається з трьох частин: оригінального зображення, теплової карти Grad-CAM, яка показує регіони, найбільш важливі для класифікації, і накладеного зображення, де теплова карта поєднана з оригінальним зображенням. Теплова карта виділяє області, що

найбільше вплинули на вибір класу зображення, дозволяючи краще зрозуміти, як модель прийняла своє рішення.

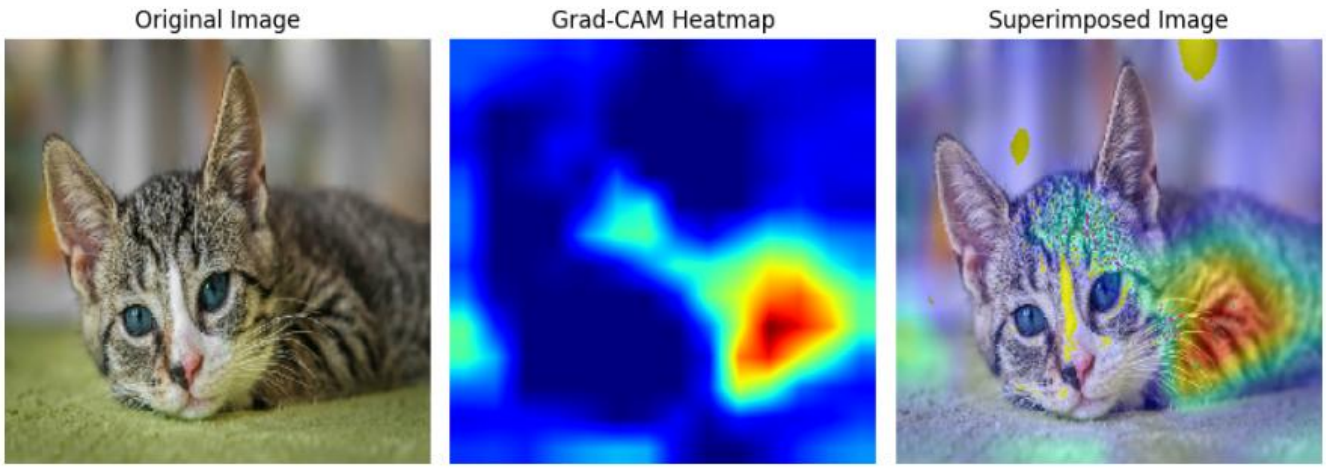


Рисунок 4.7 – Результат використання методу Grad-CAM за ТС3

Очікується, що модель повинна правильно класифікувати зображення як “кіт”, а Grad-CAM має вказати на ті частини зображення, які найбільше впливають на це рішення. Результат виконання тест-кейсу – пройдено успішно.

Тест-кейс 4. Передумови: Перевірка точності результатів класифікації. Очікуваний результат: класифікувати новий зразок (рисунок 4.8, таблиця 4.4).

Таблиця 4.4 – Тест-кейс ТС4

Тест-кейс ID: ТС0004	**Пріоритет:** 1	**Створено:** 24.11.2024 Штойко М.С.
Назва: Перевірка точності результатів класифікації з багатокласовим набором даних.		
Кроки	**Очікуваний результат**	
1.Завантажити модель глибокого навчання для класифікації. 2.Виконати класифікацію для нового зразка.	Модель повинна правильно класифікувати новий зразок як “setosa”. SHAP має пояснити важливість кожної ознаки, наприклад, довжини пелюстки та ширини чашолистка.	

3.Використати SHAP для пояснення результату класифікації.	
Результат виконання тест-кейсу: пройдено успішно	

Таблиця 4.4 наводить опис тест-кейсу ТС0004, метою якого є перевірка точності результатів класифікації з багатокласовим набором даних. Для перевірки виконуються три кроки: завантаження моделі глибокого навчання, виконання класифікації для нового зразка та використання SHAP для пояснення результату класифікації. Очікується, що модель повинна правильно класифікувати новий зразок як “setosa”, а SHAP має пояснити важливість кожної ознаки, наприклад, довжини пелюстки та ширини чашолистка. Результат виконання тест-кейсу – пройдено успішно.

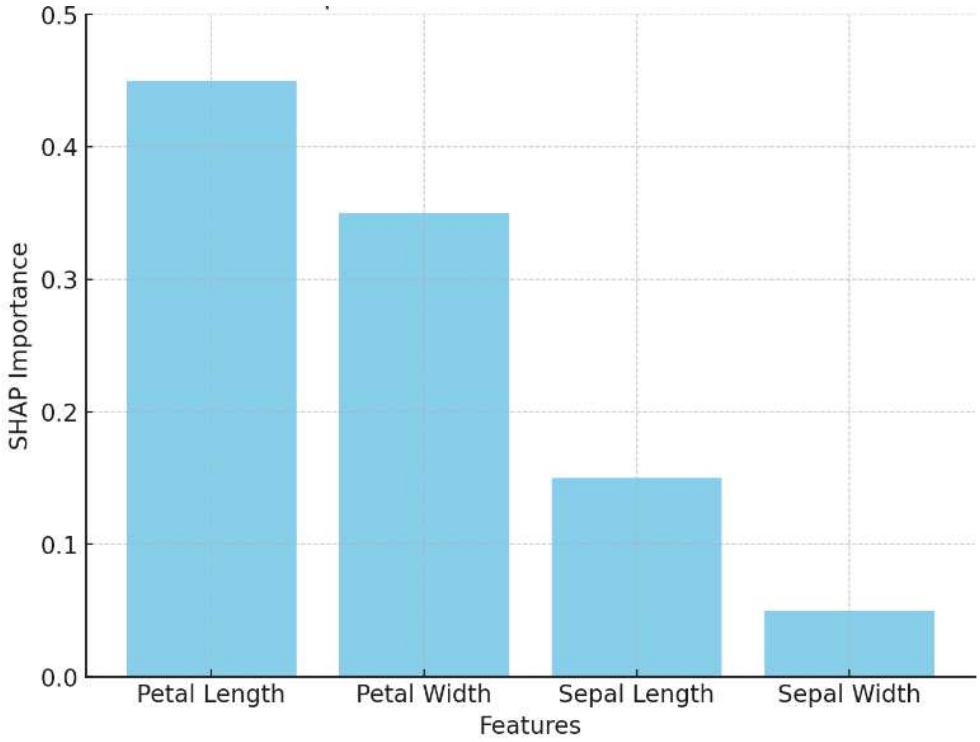


Рисунок 4.8 – Результат перевірки точності результатів класифікації за ТС4

Рисунок 4.8 є стовпчастою діаграмою, яка ілюструє важливість ознак для класифікації за допомогою методу SHAP, демонструючи, як модель приймає рішення.

Показані ознаки – довжина та ширина пелюстки, а також довжина та ширина чашолистка – впливають на результат класифікації з різною важливістю. Найбільший внесок у прийняття рішення мають довжина пелюстки (0.45) і ширина пелюстки (0.35), тоді як чашолистки мають менший вплив. Це підкреслює, що пелюстки є ключовими ознаками для моделі.

Тест-кейс 5. Передумови: Перевірка правильності пояснень. Очікуваний результат: класифікувати текст як негативний (рисунок 4.9, таблиця 4.5).

Таблиця 4.5 – Тест-кейс TC5

Тест-кейс ID: TC5	**Пріоритет:** 1	**Створено:** 24.11.2024 Штойко М.С.
Назва: Перевірка правильності пояснень для моделі з трансформером під час оброблення тексту.		
Кроки	**Очікуваний результат**	
1.Завантажити модель трансформера для класифікації тексту. 2.Виконати класифікацію тексту. 3.Використати метод LIME для пояснення результату.	Модель повинна класифікувати текст як негативний. LIME має пояснити, що слова “поганий” та “сервіс” були найбільш вагомими для цього рішення.	
Результат виконання тест-кейсу: пройдено успішно		

Таблиця 4.5 описує тест-кейс TC5, призначений для перевірки правильності пояснень для моделі з трансформером під час оброблення тексту. Для перевірки виконуються три кроки: завантаження моделі трансформера для класифікації тексту, виконання класифікації тексту та використання методу LIME для пояснення результату. Очікується, що модель повинна класифікувати текст як негативний, а LIME має пояснити, що слова “поганий” та “сервіс” були найбільш вагомими для цього рішення. Результат виконання тест-кейсу – пройдено успішно.

Рисунок 4.9 ілюструє важливість слів для класифікації тексту як негативного за допомогою методу LIME.

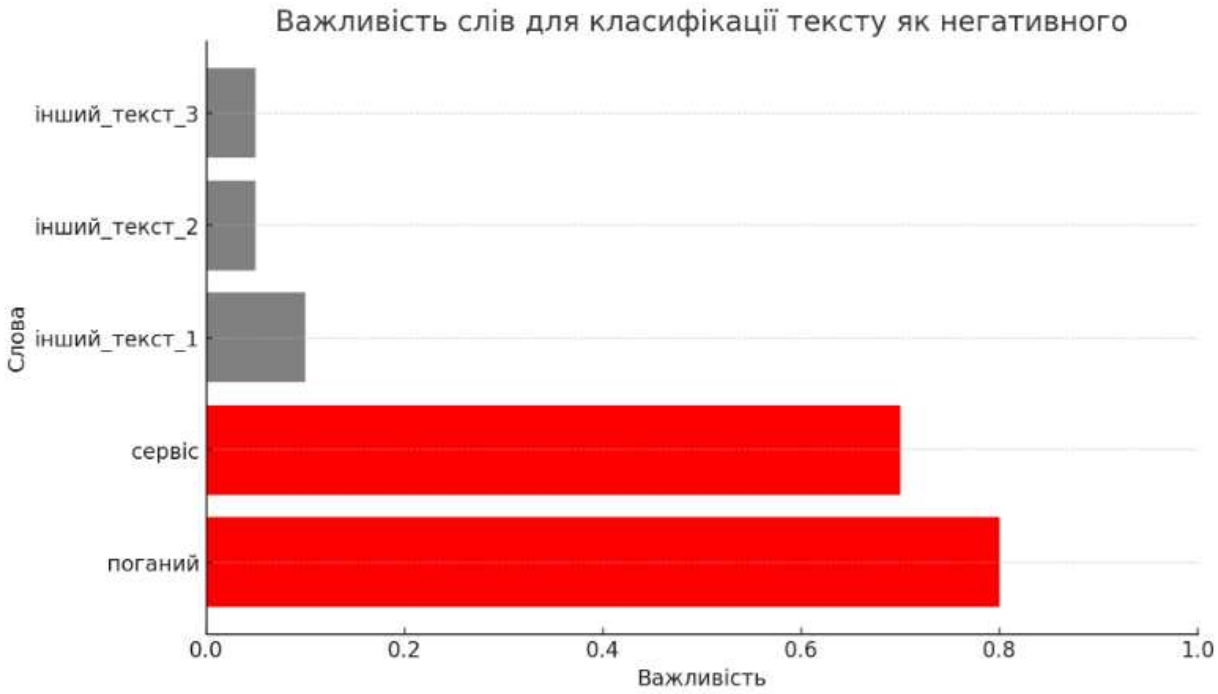


Рисунок 4.9 – Результат класифікації тексту за ТС5

Найбільш значущими словами є "поганий" та "сервіс", які виділені червоним кольором і мають найвищі значення важливості (0.8 та 0.7 відповідно). Решта слів ("інший_текст_1", "інший_текст_2", "інший_текст_3") мають значно меншу важливість, що вказує на їхній мінімальний вплив на рішення моделі. Вісь X відображає шкалу важливості від 0 до 1, а вісь Y перелічує відповідні слова.

Загалом, результати тестування підтвердили, що розроблена система є продуктивним інструментом для пояснення результатів класифікації за моделями глибокого навчання. Усі поставлені задачі в рамках цього розділу були успішно виконані, що створює надійну основу для подальшого розвитку та застосування розробленої системи.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи було проведено експериментальне тестування програмної реалізації методу пояснення результатів задач класифікації за моделями глибокого навчання. Особливу увагу приділено перевірці коректності та працездатності усіх ключових компонентів системи, а саме:

компонентів завантаження та підготовки даних, навчання моделей, оцінювання якості, пояснення та керування результатами. В процесі реалізації активно використовувалися такі інструменти, як TensorFlow, PyTorch, scikit-learn, SHAP та LIME, що дало змогу забезпечити високу якість та гнучкість програмної реалізації.

Експериментальні тести, проведені на різних наборах даних, підтвердили результативність та надійність запропонованих методів. Система продемонструвала високу точність класифікації та якість пояснень, що відповідають очікуваним результатам. Зокрема, перевірено точність пояснень для різних типів моделей, таких як CNN, RNN та Transformer, що підтверджує універсальність та широку застосовність розробленої системи. Важливо відзначити, що було проведено тестування методів пояснення SHAP, LIME та Grad-CAM з метою перевірки їхньої коректності та адекватності. Зручний вебінтерфейс, розроблений на базі Gradio, забезпечує користувачам можливість інтерактивної взаємодії з системою, дозволяючи не лише переглядати результати класифікації, а й проводити їхню детальну інтерпретацію. Було також показано, що розроблений інтерфейс є відмінним інструментом для візуалізації та аналізу результатів в реальному часі.

Загальні висновки

У кваліфікаційній роботі магістра успішно виконано методу дослідження, а саме здійснено підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

У процесі виконання роботи було спроєктовано та впроваджено метод пояснення результатів класифікації за моделями глибокого навчання засобами машинного навчання. Перший розділ роботи був присвячений аналізу предметної області, де виявлено проблему недостатньої прозорості моделей глибокого навчання, що обмежує їх застосування у критичних сферах. Проведений аналіз сучасних методів пояснення та виявлення ключових ознак впливу на рішення моделей продемонстрував важливість подальшого дослідження у напрямку розробки нових підходів та основу підґрунтям для поставки задачі дослідження.

Виконано проєктування методу пояснення результатів класифікації, за яким запропоновано підхід до перетворення векторів ознак моделей глибокого навчання у більш зрозумілі ознаки моделей машинного навчання за допомогою перехідної матриці. Під час дослідження були виділені ключові ознаки класифікаційних моделей, що впливають на інтерпретацію їхніх рішень, та розроблено структуру подання результатів класифікації, яка сприяє кращому розумінню процесу прийняття рішень.

Здійснено програмну реалізацію спроєктовано методу, в межах якої створено три ключові компоненти: завантаження та підготовки даних, оцінювання якості та пояснення класифікаційних моделей, а також компонент інтеграції та керування результатами. Ці компоненти у своїй взаємодії забезпечують гнучкий процес оброблення даних, аналізу моделей та їхні пояснення, що дає змогу користувачам глибше розуміти механізми роботи моделей.

Проведено експериментальне тестування програмної реалізації, що підтвердило коректність та ефективність роботи розробленої системи. Результати тестування продемонстрували, що система забезпечує точну класифікацію, коректні пояснення та надає користувачам зручний інструмент для аналізу отриманих

результатів. Загалом, у процесі тестування було показано, що система має високий потенціал для застосування у різноманітних задачах класифікації.

Наукова новизна даної роботи полягає у вдосконаленні методу пояснення результатів задач класифікації за моделями глибокого навчання через інтеграцію перехідної матриці, а також проєктування структуру подання результатів класифікації, що забезпечує прозорість процесу прийняття рішень. Практична значущість роботи полягає в наданні надійного інструмента для пояснення результатів класифікації, що може сприяти підвищенню довіри до моделей глибокого навчання та їхньому використанню в критичних галузях людської діяльності, де важливими є точність, надійність та прозорість.

Насамкінець, обмеженням роботи є складність реалізації більш просунутих методів пояснення та їхньої інтеграції до системи, а також обмежений набір даних для тестування. Подальші дослідження можуть бути спрямовані на вдосконалення методів пояснення з використанням складніших архітектур глибокого навчання, інтеграцію з іншими інструментами машинного навчання, а також розширення можливостей аналізу у різних контекстах. Спроєктований метод та його програмну реалізацію також можна розширити новими інструментами для автоматичного пошуку оптимальних параметрів моделей, створення динамічних дашбордів та впровадження методів для пояснення більш складних моделей, що дасть можливість ще більше покращити якість аналізу та розширити її застосування.

Перелік посилань

1. Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання / М. С. Штойко та ін. *Актуальні проблеми комп'ютерних наук АПКН-2024* : матеріали XVI Всеукр. науково-практ. конф., м. Хмельницький, 15–16 листоп. 2024 р. Хмельницький, 2024. С. 553–555. URL: https://elar.khmnu.edu.ua/handle/123456789/17151 (дата звернення: 08.12.2024).

2. Boyko N. Research into machine learning algorithms for the construction of mathematical models of multimodal data classification problems. *Computational Problems of Electrical Engineering*. 2021. Vol. 11, no. 2. P. 1–11. URL: https://doi.org/10.23939/jcpee2021.02.001 (date of access: 14.10.2024).

3. Explainable deep learning: A visual analytics approach with transition matrices / P. Radiuk et al. *Mathematics*. 2024. Vol. 12, no. 7. P. 1024. URL: https://doi.org/10.3390/math12071024 (date of access: 14.10.2024).

4. Концевой А. О., О. В. Бісікало. Моделі глибокого навчання для вирішення задачі класифікації текстової інформації. *Інформаційні технології та теорія кодування*. 2022. Вип. 55, № 3, С. 13–20. URL: https://itce.com.ua/web/uploads/pdf/901-Article%20Text-856-1-10-20221213.pdf (дата звернення: 17.09.2024).

5. Що таке глибоке навчання? Все, що вам потрібно знати. *cybercalm*. URL: https://cybercalm.org/novyny/shho-take-deep-learning/ (дата звернення: 15.09.2024).

6. Human-in-the-loop approach based on MRI and ECG for healthcare diagnosis / P. Radiuk et al. *5th International Conference on Informatics & Data-Driven Medicine* : CEUR-Workshop Proceedings, Lyon, France, 18–20 November 2022 / ed. by N. Shakhovska et al. Aachen, 2022. P. 9–20. URL: https://ceur-ws.org/Vol-3302/paper1.pdf (date of access: 14.10.2024).

7. He F. Theoretical Deep Learning : thesis. 2021. URL: https://hdl.handle.net/2123/25674 (date of access: 14.10.2024).

8. Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning / O. Melnychenko et al. *Sensors*. 2024. Vol. 24, no. 6. P. 1913. URL: https://doi.org/10.3390/s24061913 (date of access: 14.10.2024).

9. Ribeiro M. T., Singh S., Guestrin C. «Why Should I Trust You?». KDD ‘16: *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA. New York, NY, USA, 2016. URL: https://doi.org/10.1145/2939672.2939778 (date of access: 14.10.2024).

10. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing / O. Pavlova et al. *Computer systems and information technologies*. 2022. Vol. 1, no. 6. P. 62–68. URL: https://doi.org/10.31891/CSIT–2022–1–8 (date of access: 14.10.2024).

11. Local Interpretable Model-Agnostic Explanations (lime). *lime*. URL: https://lime-ml.readthedocs.io/en/latest/ (date of access: 16.10.2024).

12. Marcotcr. lime. *GitHub*. URL: https://GitHub.com/marcotcr/lime (date of access: 16.10.2024).

13. Trishul Chowdhury. SHAP (SHapley Additive exPlanations). *GitHub*. URL: https://GitHub.com/trishcho/shap (date of access: 16.10.2024).

14. Neha Vishwakarma. A Guide to Grad-CAM in Deep Learning. *Analytics Vidhya*. URL: https://www.analyticsvidhya.com/blog/2023/12/grad-cam-in-deep-learning/ (date of access: 16.10.2024).

15. Gradient-based Methods for Deep Model Interpretability. *idiap*. URL: https://www.idiap.ch/en/scientific-research/machine-learning/machine-learning-group-news/gradient-based-methods-for-deep-model-interpretability (date of access: 16.10.2024).

16. Gunning D., Aha D. DARPA’s Explainable Artificial Intelligence (XAI) Program. *AI Magazine*. 2019. Vol. 40, no. 2. P. 44–58. URL: https://doi.org/10.1609/aimag.v40i2.2850 (date of access: 17.10.2024).

17. ECG arrhythmia classification and interpretation using convolutional networks for intelligent IoT healthcare system / O. Kovalchuk et al. *1st International Workshop on Intelligent & CyberPhysical Systems (ICyberPhyS-2024)* : CEUR-Workshop Proceedings, Khmelnytskyi, Ukraine, 28 June 2024 / ed. by T. Hovorushchenko et al. Aachen, 2024. P. 47–62. URL: https://ceur-ws.org/Vol-3736/paper4.pdf (date of access: 14.10.2024).

18. Mallick R., Benois-Pineau J., Zemmari A. IFI: Interpreting for Improving: A Multimodal Transformer with an Interpretability Technique for Recognition of Risk Events.

MultiMedia Modeling. Cham, 2024. P. 117–131. URL: https://doi.org/10.1007/978-3-031-53302-0_9 (date of access: 25.11.2024).

19. Explaining Deep Learning Models for Tabular Data Using Layer-Wise Relevance Propagation / I. Ullah et al. *Applied Sciences*. 2021. Vol. 12, no. 1. P. 136. URL: https://doi.org/10.3390/app12010136 (date of access: 14.10.2024).

20. Li Y., Liang H., Zheng L. WB-LRP: Layer-wise relevance propagation with weight-dependent baseline. *Pattern Recognition*. 2024. P. 110956. URL: https://doi.org/10.1016/j.patcog.2024.110956 (date of access: 14.10.2024).

21. Myocardium segmentation using two-step deep learning with smoothed masks by Gaussian blur / V. Slobodzian et al. *Proceedings of the 6th International Conference on Informatics & Data-Driven Medicine* : CEUR-Workshop Proceedings, Bratislava, Slovakia, 17–19 November 2023 / ed. by N. Shakhovska et al. Aachen, 2024. P. 77–91. URL: https://ceur-ws.org/Vol-3609/paper7.pdf (date of access: 14.10.2024).

22. Layer-Wise Relevance Propagation: An Overview / G. Montavon et al. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham, 2019. P. 193–209. URL: https://doi.org/10.1007/978-3-030-28954-6_10 (date of access: 14.10.2024).

23. Mason del Rosario. How to interpret and use feature importance in ML models?. *AI Quality Education*. URL: https://truera.com/ai-quality-education/explainability/how-to-interpret-and-use-feature-importance-in-ml-models/ (date of access: 16.10.2024).

24. Mishra S. N. Explaining machine learning predictions : rationales and effective modifications : thesis. 2018. 131 p. URL: https://hdl.handle.net/1721.1/121599 (date of access: 16.10.2024).

25. Mona Mona Rahul Iyer Sireesha Muppala. ML Explainability with Amazon SageMaker Debugger. *AWS*. URL: https://aws.amazon.com/ru/blogs/machine-learning/ml-explainability-with-amazon-sagemaker-debugger/ (date of access: 16.10.2024).

26. Interpret model predictions using Permutation Feature Importance. *Learn MicroSoft*. URL: https://learn.microsoft.com/en-us/dotnet/machine-learning/how-to-guides/explain-machine-learning-model-permutation-feature-importance-ml-net (date of access: 16.10.2024).

27. Surrogate Model. *arize*. URL: https://docs.arize.com/arize/machine-learning/how-to-ml/explainability/surrogate-model (date of access: 16.10.2024).

28. Katharina A. Zweig. Hacking a surrogate model approach to XAI. *ar5iv*. URL: https://ar5iv.labs.arxiv.org/html/2406.16626 (date of access: 16.10.2024).

29. Harrison Jones ASA Selina Chen ASA CERA. Surrogate Models: A Comfortable Middle Ground?. *Society of Actuaries*. URL: https://www.soa.org/digital-publishing-platform/emerging-topics/surrogate-models/ (date of access: 16.10.2024).

30. Strickland E. IBM Watson, heal thyself: How IBM overpromised and underdelivered on AI health care. *IEEE Spectrum*. 2019. Vol. 56, no. 4. P. 24–31. URL: https://doi.org/10.1109/mspec.2019.8678513 (date of access: 14.10.2024).

31. Explaining Black-Box Models Using Interpretable Surrogates / D. P. Kuttichira et al. *PRICAI 2019: Trends in Artificial Intelligence*. Cham, 2019. P. 3–15. URL: https://doi.org/10.1007/978-3-030-29908-8_1 (date of access: 16.10.2024).

32. SHAP (SHapley Additive exPlanations). *christophm GitHub*. URL: https://christophm.GitHub.io/interpretable-ml-book/shap.html (date of access: 16.10.2024).

33. Google Cloud AI Platform (Explainable AI). *cloud skills boost*. URL: https://www.cloudskillsboost.google/focuses/87289?parent=catalog (date of access: 14.10.2024).

34. interpret Package. *Learn MicroSoft*. URL: https://learn.microsoft.com/en-us/python/api/azureml-interpret/azureml.interpret?view=azure-ml-py (date of access: 16.10.2024).

35. Watson OpenScale on Cloud Pak for Data. URL: https://www.ibm.com/docs/en/cloud-paks/cp-data/5.0.x?topic=services-watson-openscale (date of access: 25.11.2024).

36. Use the responsible AI image dashboard (preview). Learn MicroSoft. URL: https://learn.microsoft.com/en-us/azure/machine-learning/how-to-responsible-ai-image-dashboard?view=azureml-api-2&tabs=classification (date of access: 16.10.2024).

37. Vidyasagar Machupalli. Tutorial: Monitor Your Deployed WML Model with Watson OpenScale. *IBM*. URL: https://www.ibm.com/blog/tutorial-monitor-your-deployed-wml-model-with-watson-openscale/ (date of access: 16.10.2024).

38. Мінюк В. Р. Алгоритми та програмна реалізація методів глибокого навчання для класифікації текстових документів : дипломний проект … бакалавра : 122 Комп’ютерні науки / Мінюк Валерія Русланівна. Київ, 2023. 98 с. URL: https://ela.kpi.ua/items/c642bfe8-b3ac-4f27-a4a0-7d9bf861a4e2 (дата звернення: 14.10.2024).

39. Model interpretability. *Learn MicroSoft*. URL: https://learn.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability?view=azureml-api-2 (date of access: 16.10.2024).

40. Ribeiro. Local Interpretable Model-Agnostic Explanations. *Paperswithcode*. URL: https://paperswithcode.com/method/lime (date of access: 16.10.2024).

ДОДАТКИ

Додаток А

Копії наукових публікацій

Актуальні проблеми комп'ютерних наук

УДК 004.4

Штойко М.С., Радюк П.М., Петровський С.С., Вознюк Л.О.

Хмельницький національний університет

МЕТОД ПОЯСНЕННЯ РЕЗУЛЬТАТІВ ЗАДАЧ КЛАСИФІКАЦІЇ ЗА МОДЕЛЯМИ ГЛИБОКОГО НАВЧАННЯ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

Розглянуто прикладні аспекти розробки методу пояснення результатів класифікаційних задач для моделей глибокого навчання, який використовує сучасні інструменти машинного навчання. Запропонований метод дозволяє точно та оперативно надавати інтерпретації рішень моделей, таких як CNN, RNN та трансформери, що сприяє кращому розумінню ключових факторів, які впливають на результати класифікації. Інтеграція технік, таких як LIME та SHAP, у рамках єдиної системи надає можливість користувачам аналізувати вплив різних характеристик на рішення моделі, що підвищує прозорість і довіру до отриманих результатів.

The practical aspects of developing a method for explaining classification results for deep learning models using modern machine learning tools are examined. The proposed method provides accurate and efficient interpretations of model decisions, including CNN, RNN, and transformer models, enhancing the understanding of key factors influencing classification outcomes. Integrating techniques such as LIME and SHAP within a unified system enables users to analyze the impact of different features on model decisions, thus increasing transparency and trust in the results obtained.

У сучасному світі глибокі нейронні мережі, такі як CNN, RNN та трансформери, стали основними інструментами для вирішення складних задач класифікації. Ці моделі вражають своєю здатністю обробляти великі обсяги даних і досягати високої точності в прогнозах. Проте їхня природа "чорних ящиків" ускладнює розуміння механізмів прийняття рішень, оскільки навіть експерти можуть не мати чіткого уявлення про те, чому модель приймає певні рішення. Це викликає серйозні ризики в критичних сферах, таких як медицина, де помилка в діагностиці може загрожувати здоров'ю пацієнтів, або у фінансових системах, де необґрунтовані рішення можуть призвести до великих фінансових втрат. Відсутність прозорості у цих моделях також ставить під сумнів етичність їх використання, оскільки результати можуть бути упередженими або непрозорими для користувачів. Таким чином, існує нагальна потреба у розробці методів, які дозволять пояснювати рішення глибоких нейронних мереж, що сприятиме підвищенню довіри до цих технологій і забезпечить їх більш етичне застосування [1].

Різноманітні науковці активно досліджують методи інтерпретації результатів глибоких нейронних мереж, зокрема через інструменти, такі як LIME

(Local Interpretable Model-agnostic Explanations) та SHAP (SHapley Additive exPlanations). Ці методи розроблені для пояснення впливу різних факторів на прийняття рішень моделями, що допомагає користувачам краще розуміти, які ознаки впливають на результати. Однак, їхнє застосування у реальних сценаріях виявляє ряд проблем, зокрема неузгодженість результатів, яка може варіювати залежно від контексту, а також складність інтеграції цих методів у виробничі системи. Ці виклики підкреслюють необхідність подальшого розвитку інтерпретаційних методів, щоб забезпечити їх більш надійне та ефективне використання в різних сферах, включаючи критичні області, такі як медицина та фінанси. Потреба в більш адаптивних і прозорих рішеннях залишається актуальною, оскільки інтеграція цих технологій може сприяти зниженню ризиків, пов'язаних із прийняттям рішень [2].

Метою дослідження є розробка нового методу пояснення результатів класифікаційних задач, що базується на сучасних підходах до машинного навчання. Цей метод повинен забезпечити високий рівень прозорості моделей, що сприятиме підвищенню довіри користувачів до результатів, отриманих за допомогою глибоких нейронних мереж. Для досягнення цієї мети планується інтеграція різних технік Explainable AI у єдину систему, що дозволить комбінувати переваги існуючих підходів. Це включатиме використання методів, таких як LIME і SHAP, в поєднанні з новими підходами для більш глибокого розуміння механізмів прийняття рішень. Таким чином, дослідження прагне не лише покращити інтерпретацію результатів, а й адаптувати методи до специфіки різних задач класифікації [3].

У процесі роботи над темою використовуються різні методи Explainable AI для розробки моделі, здатної надавати інтерпретації результатів класифікацій. Для цього проводиться всебічний аналіз існуючих методів, таких як LIME і SHAP, з метою визначення їхніх сильних та слабких сторін.

Таблиця 1 ілюструє різні методи інтерпретації результатів глибоких нейронних мереж:

Таблиця 1 – Різні методи інтерпретації результатів глибоких нейронних мереж

Метод	**Опис**	**Сильні сторони**	**Слабкі сторони**
LIME	Місцеві інтерпретовані пояснення	Швидкість, простота	Неузгодженість результатів
SHAP	Додаткові пояснення Шеплі	Теоретична основа, точність	Складність у масштабуванні

Експериментальне тестування на бенчмарк-датасетах дозволяє оцінити ефективність нового методу в реальних сценаріях [4]. Попередні результати експериментів свідчать про те, що запропонована модель забезпечує зрозумілі пояснення рішень, що підвищує її прийнятність у практичних застосуваннях, таких як медицина та фінанси. Цей підхід має потенціал для розширення використання глибоких нейронних мереж у критичних сферах, де прозорість є необхідною [5].

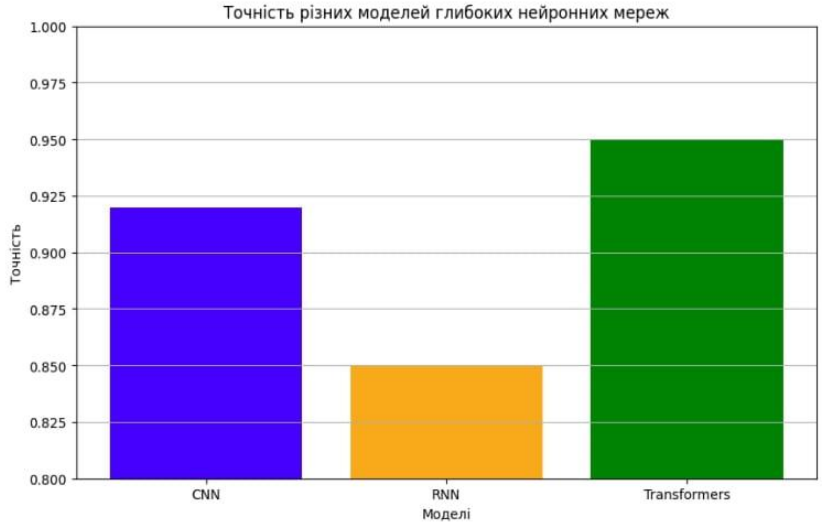


Рисунок 1 – Точність різних моделей

Розроблений метод демонструє значний потенціал у подоланні проблеми "чорної скриньки" в глибоких нейронних мережах, пропонуючи нові механізми для кращого розуміння прийнятих рішень. Це відкриває нові перспективи для вдосконалення інтерпретації результатів у реальному часі, що є особливо важливим для динамічних та критичних застосувань. У майбутніх дослідженнях слід зосередитися на інтеграції цього методу з AutoML системами, що дозволить автоматизувати процеси створення прозорих моделей, підвищуючи ефективність та доступність рішень на основі глибокого навчання.

Перелік посилань

1. Ribeiro M. T., Singh S., Guestrin C. Why Should I Trust You? Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. Pp. 1135-1144.
2. Lundberg S. M., Lee S. I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017. Vol. 30. Pp. 4765-4774.
3. Chen J., Song L., Wainwright M. J., Jordan M. I. Learning to explain: An information-theoretic perspective on model interpretation. Proceedings of the 34th International Conference on Machine Learning. 2017. Vol. 70. Pp. 1289-1298.
4. Doshi-Velez F., Kim P. Towards a rigorous science of interpretable machine learning. Proceedings of the 34th International Conference on Machine Learning. 2017. Vol. 70. Pp. 2961-2970.
5. Chatzimparmpas A., Mavridis P., Mavridis D. Interpretable Machine Learning: Definitions, Methods, and Applications. Proceedings of the 11th International Conference on Machine Learning and Data Engineering. 2021. Pp. 156-160.

Додаток Б

Лістинг програмного коду

```
import gradio as gr
import matplotlib.pyplot as plt
from PIL import Image
import torch.nn as nn
import numpy as np
import os, json
import torch
from torchvision import models, transforms
from torch.autograd import Variable
import torch.nn.functional as F
from lime import lime_image
from skimage.segmentation import mark_boundaries
import asyncio
with open(os.path.abspath('imagenet_class_index.json'), 'r') as read_file:
  class_idx = json.load(read_file)
  idx2label = [class_idx[str(k)][1] for k in range(len(class_idx))]
  cls2label = {class_idx[str(k)][0]: class_idx[str(k)][1] for k in range(len(class_idx))}
  cls2idx = {class_idx[str(k)][0]: k for k in range(len(class_idx))}
# Pre-trained model
model = models.inception_v3(pretrained=True)
model.eval()
# Transformations
def get_pil_transform():
  return transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.CenterCrop(224)
  ])
def get_preprocess_transform():
  normalize = transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
  return transforms.Compose([
    transforms.ToTensor(),
    normalize
  ])
pill_transf = get_pil_transform()
preprocess_transform = get_preprocess_transform()
# Prediction function
async def batch_predict(images):
  await asyncio.sleep(0)  # Non-blocking to allow other tasks
  model.eval()
  batch = torch.stack(tuple(preprocess_transform(i) for i in images), dim=0)
  device = torch.device(«cuda» if torch.cuda.is_available() else «cpu»)
  model.to(device)
  batch = batch.to(device)
  logits = model(batch)
  probs = F.softmax(logits, dim=1)
  return probs.detach().cpu().numpy()
```

```
# Explain prediction with LIME
async def explain_with_lime(img, top_labels=5, num_samples=1000, positive_only=True):
    await asyncio.sleep(0)  # Non-blocking to allow other tasks
    explainer = lime_image.LimeImageExplainer()
    explanation = explainer.explain_instance(
        np.array(pill_transf(img)),
        await batch_predict([img]),  # Async prediction
        top_labels=top_labels,
        hide_color=0,
        num_samples=num_samples
    )
    temp, mask = explanation.get_image_and_mask(
        explanation.top_labels[0],
        positive_only=positive_only,
        num_features=10,
        hide_rest=False
    )
    img_boundry = mark_boundaries(temp / 255.0, mask)
    return img_boundry
# Gradio functions
async def classify_and_explain(image, top_labels, num_samples, positive_only):
    # Prediction
    logits = await batch_predict([image])
    top5_probs = torch.topk(torch.tensor(logits), 5)
    predictions = [
        (idx2label[top5_probs.indices[0][i].item()], top5_probs.values[0][i].item())
        for i in range(5)
    ]
    # Explanation
    lime_image = await explain_with_lime(image, top_labels, num_samples, positive_only)
    return predictions, lime_image
# Gradio interface
title = «Asynchronous Image Classifier with Configurable LIME Explanation»
description = (
    «Upload an image to classify it and view the model's explanation using LIME. «
    «This version uses async processing for improved responsiveness.»
)
gr.Interface(
    fn=classify_and_explain,
    inputs=[
        gr.Image(type=«pil»),
        gr.Slider(1, 10, value=5, step=1, label=«Top Labels»),
        gr.Slider(100, 5000, value=500, step=100, label=«Number of Samples»),
        gr.Checkbox(value=True, label=«Positive Only»)
    ],
    outputs=[
        gr.Label(num_top_classes=5),
        gr.Image()
    ],
    title=title,
```

```
    description=description,
    allow_flagging=«never»
).launch()
```

Додаток В

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

Виконав:
студент 2 курсу магістратури, групи КНм-23-1
Штойко Микола Сергійович
Керівник:
Старший викладач кафедра КН
Радюк Павло Михайлович

Актуальність

Глибокі нейронні мережі забезпечують високу ефективність у вирішенні складних задач класифікації, проте їхня складна внутрішня структура робить ці моделі "чорними скриньками", рішення яких важко пояснити. Відсутність прозорих пояснень створює ризики в критичних сферах, таких як медицина, фінанси та право, де необґрунтовані або незрозумілі прогнози можуть мати серйозні наслідки. Це підкреслює необхідність розробки інтерпретованих моделей, які підвищать довіру користувачів та забезпечать прозорість рішень для їх безпечного та відповідального застосування.

Мета і задачі роботи

Мета і задачі роботи – підвищення рівня якості пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

Досягнення мети роботи передбачає виконання таких задач:

1. Провести аналіз моделей, методів та засобів пояснювального штучного інтелекту.

2. Спроєктувати модель подання результатів задач класифікації за моделями глибокого навчання через ознаки моделей машинного навчання.

3. Спроєктувати метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання.

4. Виконати програмну реалізацію та провести експериментальне тестування спроєктованого методу пояснення результатів задач класифікації.

Кроки формування способу подання результатів класифікації

Крок 1. Отримання результатів класифікації
Отримання передбачень або рішень моделі на основі введених даних для подальшого аналізу.

Крок 2. Збір інформації про важливість ознак
Оцінка впливу ознак на рішення через методи інтерпретації (SHAP, LIME).

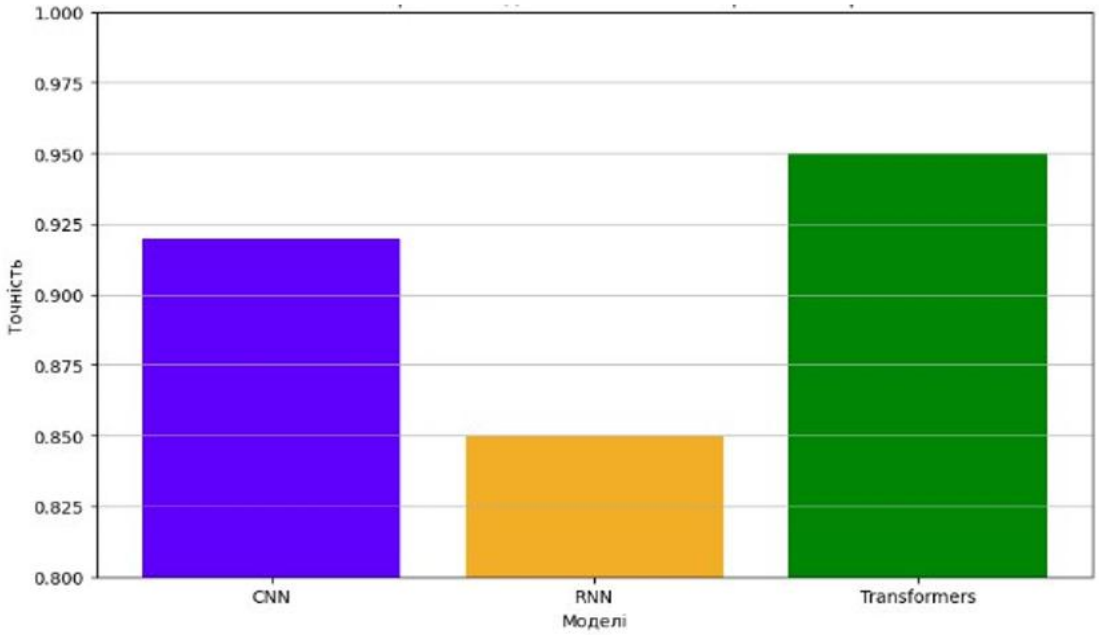
Крок 3. Вибір способу подання
Вибір відповідного формату подання результатів: графіки, теплові карти чи текстові пояснення.

Крок 4. Формування пояснень
Створення пояснень, що детально описують вплив ознак на класифікацію.

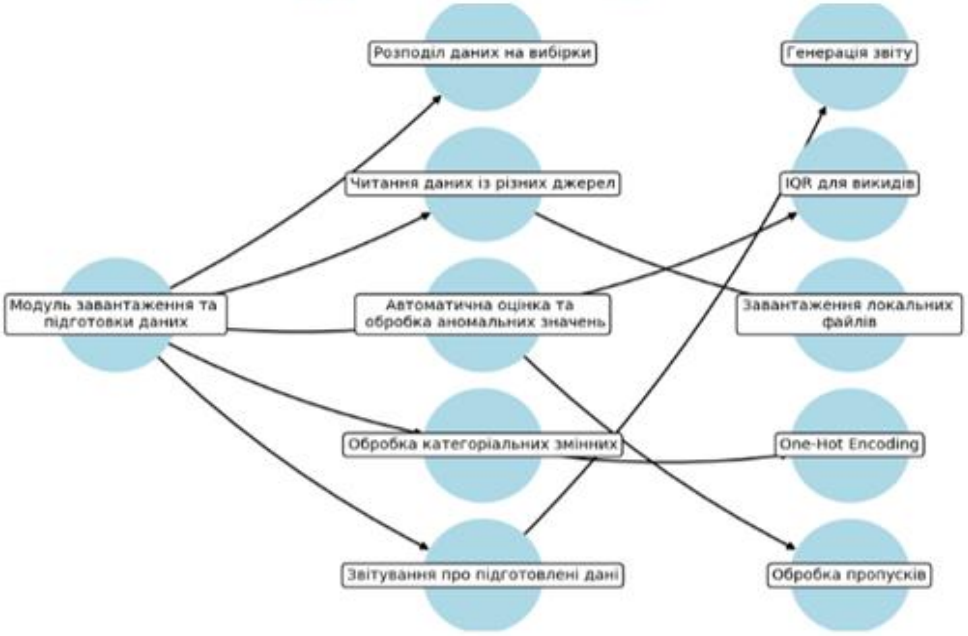
Крок 5. Інтеграція у інтерфейс
Інтеграція результатів і пояснень у зручний інтерфейс для користувачів.

Крок 6. Перевірка та тестування
Тестування точності подання результатів та отримання відгуків для вдосконалення інтерфейсу.

Порівняння точності моделей

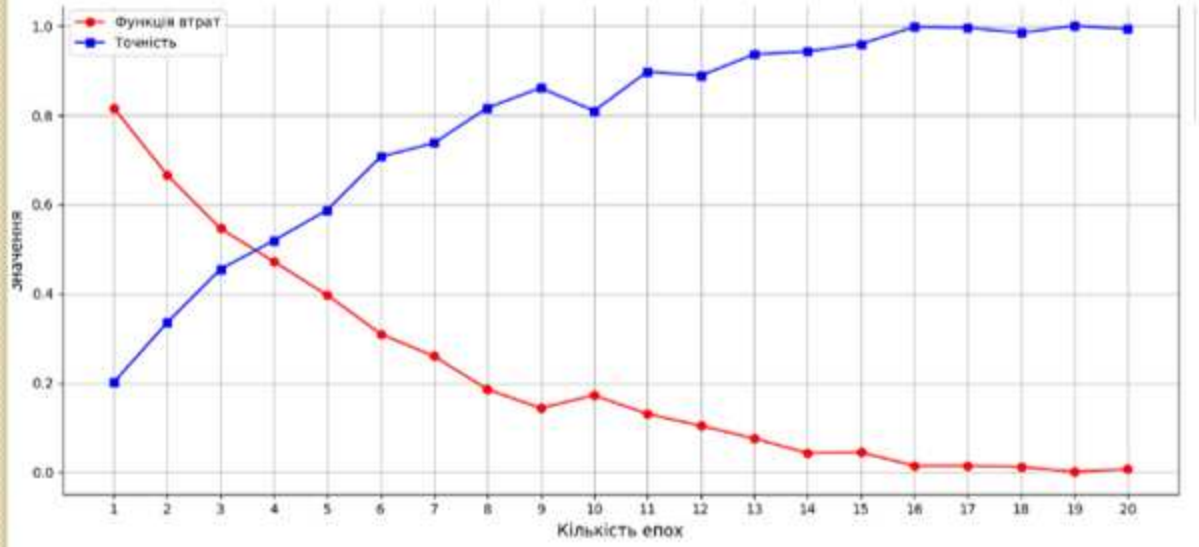


Структура модуля завантаження та підготовки даних

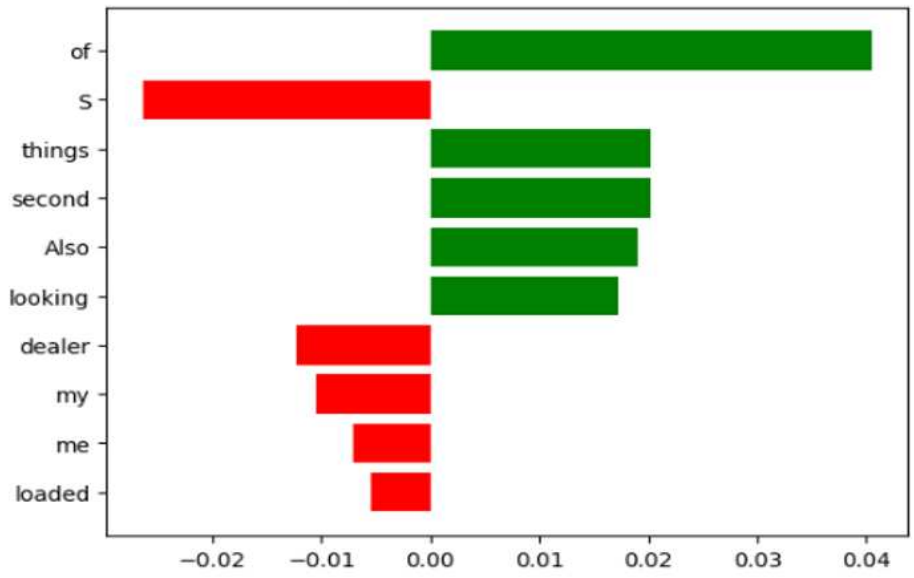


Модуль управління результатами та звітності

Процес навчання моделі



Перевірка роботи моделі



Висновки

У магістерській роботі розроблено метод пояснення результатів класифікації моделей глибокого навчання засобами машинного навчання. Запропоновано перетворення векторів ознак глибоких моделей у зрозуміліші ознаки за допомогою перехідної матриці. Створено структуру представлення результатів, що підвищує прозорість рішень, та програмну реалізацію з трьома компонентами: підготовкою даних, аналізом моделей і інтеграцією результатів.

Проведене тестування підтвердило ефективність запропонованого методу: система забезпечує точну класифікацію та зручний інструмент для аналізу, що сприяє довірі до моделей глибокого навчання. Наукова новизна роботи полягає у застосуванні перехідної матриці для пояснення класифікації, а практична значущість – у наданні надійного інструмента для прозорого аналізу моделей у різних сферах діяльності. Обмеженням є складність інтеграції складніших методів пояснення та обмежений набір тестових даних. Подальші дослідження спрямовані на вдосконалення методів пояснення, інтеграцію нових інструментів та розширення функціональних можливостей системи.

Tue Dec 17 19:01:57 EET 2024, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.258 Educational

Максимальне співпадіння з одним документом 12.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 9%**

	Документ		Сумарний збіг по Базі Даних	
ID: 160539 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання Додано в БД: 2024-12-17 Автора: Микола ШТОЙКО Керівники: Павло РАДЮК Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	97359	1433	14284 (15%)	207 (14%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
134181	Назва: ЗВІТ з науково-дослідної практики Додано в БД: 2024-10-17 Автора: Штойко Микола Керівники: Скрипник Т.К. Консультанти: Опоненти:	11748 (12.0%)	180 (13.0%)

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Микола ШТОЙКО

Співавтор:

Назва: Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання

Науковий керівник: Павло РАДЮК, старший викладач кафедри, Ph.D.

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:1.5%

Коефіцієнт подібності 2:0.5%

Мікропробіли: 0

Заміна букв: 2

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2024-12-17 18:46:50.0

Після аналізу Звіту подібності констатую наступне:

☑ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

Дата 17.12.2024 експерт Р. Р. Радюк

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання*

Автор: *студент групи КНм-23-1 Штойка Микола Сергійович*

Спеціальність: *122 Комп'ютерні науки*

Освітня програма: *освітньо-професійна*

Науковий керівник: *док. філ., ст. викл. Радюк П.М.*

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	***відповідає***
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Обсяг запозичень, що визначений системами виявлення збігів/ ідентичності/схожості, складає:

*– за системою Anti-Plagiarism: **12.0%**: максимальне співпадіння з одним документом, а саме зі звітом з науково-дослідної практики магістранта Штойка Миколи, що не є плагіатом; поміж запозичень знаходяться загальновідомі терміни та скорочення;*

*– за системою StrikePlagiarism: КП 1 – **1.5%**, КП 2 – **0.5%**: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; до запозичень входять фрагменти програмного коду, що не мають авторства та містять поширені конструкції.*

Отже, запозичення є допустимими та відносяться до описаних вище і адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ *Павло РАДЮК*

Гарант ОП _____ *Руслан БАГРІЙ*

Завідувач кафедри КН _____ *Олександр БАРМАК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

студента *гр. КНм-23-1 Штойка Миколи Сергійовича*
за темою *Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання*

1. Актуальність теми

Кваліфікаційна робота магістра, що присвячена удосконаленню методу пояснення результатів задач класифікації за моделями глибокого навчання, є вкрай актуальною. Сучасні моделі глибокого навчання, попри високу ефективність, залишаються "чорними скриньками", що обмежує їхнє застосування до відповідальних галузей людської діяльності, де прозорість та інтерпретованість рішень є критично важливими. Необхідність проєктування методу пояснення результатів задач класифікації за моделями глибокого є вимогою часу, що робить дане дослідження вельми значущим.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимогам наукових робіт

Робота повністю відповідає спеціальності 122 Комп'ютерні науки, оскільки в ній розглядаються математичні, інформаційні та імітаційні моделі реальних явищ, а також методи й технології отримання, оброблення та використання інформації. Спроєктований метод сприяє підвищенню довіри до рішень систем на основі глибокого навчання, що є важливою складовою сучасних комп'ютерних наук. Структура, оформлення та зміст роботи відповідають загальним вимогам до наукових праць.

3. Професійні та особистісні якості магістранта

Під час виконання кваліфікаційної роботи магістр Штойко Микола Сергійович продемонстрував задовільний рівень професійної компетентності, самостійності та відповідальності. Він проявив аналітичні здібності, уміння працювати з науковою літературою та застосовувати сучасні технології машинного навчання. Студент проявив наполегливість та цілеспрямованість для досягнення поставлених задач, що позитивно відзначилося на якості виконаної роботи.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Магістрант Штойко Микола Сергійович проявив достатній ступінь самостійності на всіх етапах виконання кваліфікаційної роботи, починаючи від аналізу наукових джерел та постановки задачі до проєктування методу та формулювання висновків. Результати роботи та їхня практична значущість отримані студентом особисто і обґрунтовані належно.

5. Наукова новизна та оригінальність запропонованих підходів

Наукова новизна роботи полягає у вдосконаленні методу пояснення результатів задач класифікації за моделями глибокого навчання через інтеграцію перехідної матриці та проєктуванню структури подання результатів, що забезпечує прозорість процесу

прийняття рішень. Запропонований підхід є оригінальним та має практичне значення для покращення розуміння роботи складних моделей.

6. Ступінь оволодіння методами дослідження

Штойко Микола Сергійович продемонстрував задовільний ступінь володіння методами дослідження та сучасними технологіями машинного навчання та штучного інтелекту, як от TensorFlow, PyTorch, scikit-learn, SHAP та LIME. Він успішно застосовував ці методи на практиці, що свідчить про хороший рівень його професійної компетентності.

7. Повнота та якість розкриття теми роботи

Тему роботи розкрито повно та достатньо. Актуальність предметної галузі та аналіз відомих досліджень виконано належно. Усі поставлені завдання успішно виконані, а результати підтверджують досягнення мети дослідження. Запропонований метод є обґрунтованим та відповідає сучасним вимогам спеціальності комп'ютерних наук.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал кваліфікаційної роботи подано логічно, послідовно та аргументовано. Стиль викладу є науковим і відповідає вимогам до кваліфікаційних робіт. Магістрант продемонстрував літературну грамотність, що забезпечує доступність сприйняття матеріалу.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Спроєктований метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання може бути використаний у різних галузях, де використовуються системи штучного інтелекту для прийняття рішень. Його впровадження дасть можливість фахівцям краще розуміти результати моделей глибокого навчання, що сприятиме більш ефективному та етичному використанню технологій штучного інтелекту в таких галузях як медицина, фінанси, та промисловість.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

З огляду на задовільний рівень виконання, отримані результати та відповідність усім вимогам до кваліфікаційних робіт магістра, вважаю, що кваліфікаційна робота Штойка Миколи Сергійовича може бути допущена до захисту.

*Рекомендована оцінка – **«задовільно»**.*

Керівник _____ *док. філ., ст. викл. каф. КН Павло РАДЮК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

студента *гр. КНм-23-1 Штойка Миколи Сергійовича*
за темою: *Метод пояснення результатів задач класифікації за моделями глибокого навчання засобами машинного навчання*

1. Актуальність обраної теми

Пояснення результатів задач класифікації за моделями глибокого навчання є надзвичайно актуальною та важливою проблемою в сучасній науці та технологіях. Зростаюча залежність від систем штучного інтелекту вимагає прозорості їхніх рішень, особливо в критичних областях, де наслідки рішень можуть бути суттєвими. Ця робота розглядає важливий аспект, що є необхідним для подальшого вдосконалення та практичного застосування моделей глибокого навчання.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимовам до наукових робіт

Кваліфікаційна робота магістранта Штойка Миколи повністю відповідає спеціальності 122 "Комп'ютерні науки". Робота охоплює актуальні теоретичні та практичні аспекти, починаючи від аналізу проблеми та огляду наукових досліджень до проєктування методів та їхньої експериментальної перевірки. Усі етапи виконання відповідають вимогам до кваліфікаційних робіт і стандартам підготовки фахівців у галузі 12 Інформаційні технології.

3. Повнота розкриття мети та завдань дослідження

Мету та завдання дослідження розкрито повно. Студент продемонстрував глибокий аналіз предметної області та сформулював чітку структуру для спроєктованого методу. Описано алгоритми та інструменти, необхідні для реалізації методу, а також наведено результати експериментального тестування, які підтверджують досягнення поставленої мети.

4. Наявність наукової новизни

Наукова новизна кваліфікаційної роботи полягає в удосконаленні методу пояснення результатів задач класифікації за моделями глибокого навчання, який використовує інтеграцію перехідної матриці між ознаками моделей глибокого навчання та моделями машинного навчання, а також проєктуванню структури подання результатів класифікації. Цей метод забезпечує підвищення якості інтерпретації рішень моделей глибокого навчання, що є суттєвим внеском у дослідження проблем прозорості та довіри до автоматизованих систем.

5. Зміст кожного розділу роботи

Робота має чітку та логічну структуру. Перший розділ присвячено аналізу проблем пояснення результатів класифікації та огляду існуючих підходів. Другий розділ містить детальне проєктування запропонованого методу. Третій розділ описує програмну реалізацію методу та інтеграцію необхідних компонентів. Четвертий розділ

подає результати експериментального тестування та аналізу результативності методу. Висновки кожного розділу підсумовують основні результати.

6. Ступінь розкриття теми роботи

Заявлену тему розкрито на задовільному рівні. Робота демонструє детальне дослідження, проєктування та програмну реалізацію методу пояснення результатів класифікації за моделями глибокого навчання. Магістрант надав обґрунтований опис спроєктованого методу, а також алгоритмів та інструментів, використаних для його реалізації, що підтверджено експериментальним тестуванням.

7. Якість оформлення кваліфікаційної роботи

Кваліфікаційна робота оформлена згідно з вимогами до наукових праць, має чітку структуру та включає всі необхідні розділи: перелік скорочень, вступ, огляд літератури, методологія, результати експериментів, висновки та перелік посилань. Джерела належно процитовані та включені до списку використаної літератури. Загальна якість оформлення є високою, сприяючи легкому сприйняттю матеріалу.

8. Недоліки оформлення кваліфікаційної роботи

Хоча в роботі є графіки, що ілюструють результати експериментів (наприклад, на сторінці 52, де показано "Процес навчання моделі"), вони часто не мають чітких підписів осей, що ускладнює точне розуміння наведених даних. Це не дає змоги чітко зрозуміти, наскільки модель покращує свою продуктивність з часом, що є важливим для оцінювання її ефективності.

9. Недоліки кваліфікаційної роботи

В роботі не розглядаються випадки, коли методи пояснення, такі як SHAP або LIME, можуть давати неточні або суперечливі результати, що можуть виникати через особливості моделей глибокого навчання. Не розглянуто питання про можливі спотворення інтерпретації рішень, що може впливати на довіру до моделі. Бракує аналізу можливих обмежень удосконаленого методу пояснення, які можуть бути критичними для їхнього застосування у реальних умовах. Також варто було б більш детально розкрити питання щодо виникнення можливих артефактів під час пояснення результатів класифікації та їхнього впливу на кінцеві результати.

10. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота

Подана кваліфікаційна робота магістра Штойка Миколи є самостійним та завершеним науковим дослідженням. Враховуючи задовільний рівень виконання та забезпечення всіх вимог до кваліфікаційної роботи, а також позитивні результати тестування, я вважаю, що кваліфікаційна робота може бути допущена до захисту.

*Рекомендована оцінка – **«задовільно»**.*

Опонент (прізвище, ім'я, по батькові, посада, місце роботи)

Лисенко Сергій Миколайович, Доктор технічних наук ДУТУ

«18» грудня 2024 р. _____

(підпис)