НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету (Директор ННІ)	**Завідувач кафедри**
інформаційних технологій (назва факультету (ННІ))	комп'ютерних наук (назва кафедри)
_____ Ігор БОЛБОТ (підпис) (ім'я ПРІЗВИЩЕ)	_____ Белла ГОЛУБ (підпис) (ім'я ПРІЗВИЩЕ)
"___" _____ 2025 р.	"___" _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Застосування машинного навчання та глибоких нейронних мереж для аналізу фармацевтичних даних

Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма Інформаційні управляючі системи і технології
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к.т.н., доцент _____ Белла ГОЛУБ
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент _____ Олексій ТКАЧЕНКО
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Виконав _____ Владислав ВОЛОДЧЕНКО
(підпис) ((ім'я ПРІЗВИЩЕ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____
к.т.н., доцент _____ Белла ГОЛУБ
(науковий ступінь, вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)
"_01_" ___листопада____2024 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Володченку Владиславу Олександровичу
(прізвище, ім'я, по батькові)

Спеціальність ___122 «Комп'ютерні науки»_____

Освітня програма __Інформаційні управляючі системи і технології_____

Орієнтація освітньої програми _освітньо-професійна_____

Тема магістерської кваліфікаційної роботи Застосування машинного навчання та глибоких нейронних мереж для аналізу фармацевтичних даних

затверджена наказом ректора НУБіП України від "01" листопада 2024 р. №1964 «С»

Термін подання завершеної роботи на кафедру 01.12.2025

Вихідні дані до магістерської кваліфікаційної роботи дані про фармацевтичні препарати та їх властивості, набори даних для тренування моделей машинного навчання, програмне забезпечення для розробки, база даних для збереження інформації.

Перелік питань, що підлягають дослідженню:

1. Аналіз предметної області та визначення факторів, що впливають на показники лікарських засобів та виявлення побочних ефектів.
2. Дослідження можливостей використання алгоритмів машинного навчання та глибоких нейронних мереж для прогнозування властивостей препаратів та виявлення побочних ефектів.
3. Оцінка точності та формування висновків по результатам роботи моделей, аналіз практичної придатності розробленої системи.

Дата видачі завдання "01" листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Олексій ТКАЧЕНКО
(підпис) (ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Владислав ВОЛОДЧЕНКО
(ім'я ПРІЗВИЩЕ)

Зміст

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ML - Machine Learning (Машинне навчання);

DL - Deep Learning (Глибоке навчання);

ADR - Adverse Drug Reaction (Побічні реакції на ліки);

KPI - ключові показники ефективності;

SSAS (SQL Server Analysis Services) - служби аналізу SQL серверу;

SSIS (SQL Server Integration Services) – служби інтергації даних SQL;

OLAP (Online Analytical Processing) - технологія багатовимірного аналітичного опрацювання даних.

ВСТУП

Актуальність дослідження. Сучасна фармацевтична галузь генерує великі обсяги цифрових даних, в процесі лабораторних досліджень та при застосуванні лікарських засобів на практиці. З кожним роком такі дані стають більш складними для аналізу звичайними статистичними методами. Це створює потребу у використанні нових технологій, здатних виявляти закономірності, які не можуть бути виявлені традиційними методами аналізу фармацевтичних даних.

Машинне навчання та глибокі нейронні мережі є одними з ключових інструментів сучасної аналітики, які демонструють високу швидкість та точність у вирішенні задач класифікації, прогнозування та оцінки ризиків у сфері медицини. Використання таких методів дозволяє автоматизувати аналіз фармацевтичних даних, підвищувати точність оцінювання властивостей лікарських засобів, прискорювати процеси навчання та оцінки моделей та зменшувати ймовірність помилкових рішень. Таким чином, технології штучного інтелекту стають все більше важливими в задачах, де потрібно автоматизувати монотонну та складну роботу, щоб виключити вплив людського фарктору на результат.

Магістерська робота спрямована на дослідження можливостей застосування алгоритмів машинного навчання та глибинних нейронних мереж для аналізу фармацевтичних даних, розробку архітектури програмного рішення для побудови моделей та оцінку їхньої працездатності на реальних даних. Робота передбачає порівняння результатів класичних моделей машинного навчання з результатами глибокої нейронної мережі по ключовим показникам моделей, для об'єктивного порівняння моделі будуть мати однаковий набір навчальних даних, що допоможе зрозуміти як вони працюють в рівних початкових умовах.

Отримані результати дослідження можуть бути використані для підвищення якості аналітики у фармацевтичній сфері. Значення машинного

навчання у цій сфері продовжує зростати, що визначає актуальність даної роботи та її наукову цінність.

У рамках даної роботи під ефективністю лікарського засобу розуміється узагальнений показник, що відображає очікувану результативність препарату при його клінічному застосуванні без виникнення серйозних побочних ефектів, сформований на основі фармакологічних характеристик, активних речовин, історичних результатів попередніх моделей прогнозування та інших даних. Тобто ефективність у цьому дослідженні не є конкретною клінічною величиною, а є прогнозованою оцінкою, яка формується моделями машинного навчання з метою визначення потенційної успішності препарату відносно інших представників своєї групи. Ця величина не замінює реальні клінічні дані, але служить аналітичним показником для допомоги аналізування роботи системи.

Об’єктом дослідження є фармацевтичні дані, що формуються у процесі дослідження та практичного використання лікарських засобів, а також дані про потенційні побочні ефекти від препаратів.

Предметом дослідження є алгоритми машинного навчання та методи глибинних нейронних мереж, принципи їх побудови, навчання та оптимізації, а також способи їхнього застосування для обробки, аналізу та моделювання фармацевтичних даних з метою прогнозування властивостей препаратів, виявлення прихованих закономірностей.

Метою дослідження є підвищення точності аналізу фармацевтичних даних шляхом розробки, дослідження та оцінки моделей машинного навчання і глибинних нейронних мереж для прогнозування властивостей фармацевтичних препаратів, оцінки ризиків їх застосування та формування аналітичних висновків.

Для досягнення мети у роботі необхідно вирішити такі завдання:

1. Провести теоретичний аналіз сучасних підходів і наукових праць, присвячених застосуванню машинного навчання та глибинного навчання у медицині та фармацевтиці, виділити ключові переваги та обмеження цих методів.

2. Виконати збір, відбір та попередню обробку фармацевтичних даних, включаючи нормалізацію, підготовку ключових ознак, формування навчальних та тестових вибірок, а також забезпечення коректності структури даних для використання у моделях.

3. Реалізувати класичні алгоритми машинного навчання (логістична регресія, дерева рішень, випадковий ліс) та побудувати глибоку нейронну мережу для порівняння результатів їх роботи на однакових вхідних даних.

4. Виконати навчання моделей та провести дослідження щодо їх показників, визначивши ключові метрики якості, такі як accuracy, precision, recall, F1-score.

5. Оцінити практичну придатність розроблених моделей і системи, визначити можливостей впровадження в процес роботи аналітичних підрозділів фармацевтичних компаній, наукових лабораторій та дослідницьких центрів.

Новизна роботи полягає у розробці та дослідженні підходу до аналізу фармацевтичних даних із застосуванням методів машинного навчання та глибинних нейронних мереж, що дає змогу для прогнозування властивостей лікарських засобів та виявлення можливих побочних ефектів. У роботі запропоновано методику побудови моделей, що враховує специфіку фармацевтичних даних.

На відміну від традиційних підходів, де використовуються виключно класичні статистичні методи або поодинокі алгоритми машинного навчання, у роботі реалізовано порівняльний аналіз декількох моделей із різною структурою, включаючи глибоку нейронну мережу, що надало можливість визначити оптимальні методи для обробки фармацевтичних даних. Показано,

що використання глибинного навчання в поєднанні з правильно сформованою архітектурою та попередньою підготовкою даних дозволяє досягти кращої стабільності результатів порівняно з класичними методами.

Запропонована методика формує основу для побудови прогнозних моделей, які можуть адаптуватися до нових наборів фармацевтичних даних без значного погіршення результатів, а також демонструє можливості застосування алгоритмів машинного навчання як інструмента підтримки для фармацевтичної аналітики, що розширює науково-методичну базу в цій галузі.

Пояснювальна записка складається з вступу, 4 основних розділів, висновки, списку використаних джерел у кількості 21 найменувань та додатків до роботи, основний текст викладений на 70 сторінках.

У магістерській роботі представлено методологічні підходи та практичні результати створення системи для оцінювання ефективності лікарських засобів, а також прогнозування можливих побічних ефектів з використанням методів машинного навчання та глибокої нейронної мережі. Робота складається з чотирьох розділів, кожен з яких описує окремий етап дослідження.

У першому розділі роботи проведено аналіз фармацевтичної галузі та особливостей обробки медичних і фармакологічних даних. Розглянуто традиційні методи аналізу, їхні основні обмеження та недоліки. Також розглянуті сучасні підходи, які використовують моделі машинного навчання.

У другому розділі здійснено моделювання програмної системи, що реалізує процес аналізу фармацевтичних даних. Представлено об’єктне й функціональне моделювання у вигляді UML-діаграм, які описують роботу системи, взаємодію між компонентами та структуру її класів.

У третьому розділі детально описано процес розробки системи та принципи реалізації функціональних компонентів. Наведено структуру бази даних і сховища даних, а також описано процедури їх заповнення та обробки інформації. Особливу увагу приділено модулю машинного навчання, у якому розглянуто використані алгоритми, процес тренування моделей та методи

оцінювання їхньої якості. Крім того, представлено модуль прогнозування властивостей лікарських засобів, який забезпечує аналітичні можливості системи.

У четвертому розділі наведено результати проведеного дослідження. Описано вимоги до апаратного та програмного забезпечення, необхідного для коректної роботи системи. Проведено детальний аналіз отриманих результатів роботи моделей машинного навчання по ключовим показникам. Розглянуто результати процесу прогнозування властивостей лікарських засобів та виявлення можливих побочних ефектів.

У висновках підсумовано результати магістерської роботи, сформовані основні результати, яких вдалось досягнути в дослідженні. Також було оцінено запропоновані методи та визначено перспективи подальшого розвитку системи.

РОЗДІЛ 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз фармацевтичної галузі

Фармацевтична галузь є однією з найбільш наукоємних та даноорієнтованих сфер сучасної економіки. У процесі створення, дослідження, виробництва та контролю якості лікарських засобів формується значна кількість різноманітних даних, які охоплюють широкі категорії інформації - від ранніх доклінічних експериментів до постмаркетингового нагляду за безпекою препаратів. Внаслідок впровадження цифрових технологій, автоматизації виробничих процесів, розвитку лабораторних інформаційних систем, електронної медичної документації обсяг таких даних постійно зростає, що актуалізує потребу в застосуванні сучасних аналітичних методів та інтелектуальних алгоритмів обробки інформації.

Важливою особливістю фармацевтичної сфери є багатовекторність походження даних. Науково-дослідні лабораторії формують величезні масиви первинних експериментальних даних, що стосуються молекулярних властивостей та параметрів фармакологічних субстанцій. Доклінічні етапи досліджень пов’язані з біологічними експериментами, які включають дані щодо токсичності, фармакокінетики, біосумісності та впливу на різні біологічні системи. Значні обсяги інформації генеруються на стадії клінічних випробувань, де оцінюються властивості та безпечність лікарських засобів залежно від дозування, характеристик пацієнтів та клінічних умов застосування.

Таким чином, фармацевтична галузь формує складні, різнорідні та багатовимірні набори даних, які характеризуються великим обсягом, високою швидкістю формування та високим рівнем варіативності параметрів. Така інформація є надзвичайно цінною для аналізу, прогнозування та оптимізації як наукових, так і виробничо-технологічних процесів. Проте ручна обробка та класичні статистичні методи аналізу часто не забезпечують достатнього рівня

точності та гнучкості для роботи з такими даними, що зумовлює необхідність використання сучасних методів машинного навчання та глибинних нейронних мереж. Ці підходи здатні працювати з великою кількістю ознак, автоматично виявляти приховані закономірності, моделювати складні нелінійні залежності та формувати прогностичні висновки за мінімального втручання експерта.

Зростання ролі цифрових даних у фармацевтичній індустрії визначає нові вимоги до систем аналітики, підходів обробки інформації та побудови моделей прогнозування. Створення інтелектуальних інструментів аналізу стає необхідною умовою підвищення якості досліджень, скорочення термінів розробки лікарських засобів, зменшення фінансових витрат на проведення експериментів та забезпечення високої якості фармацевтичної продукції. Саме тому використання машинного навчання та глибоких нейронних мереж у цій сфері є не лише перспективним науковим напрямом, але й практично необхідним елементом розвитку фармацевтичної науки та індустрії.

1.2. Традиційні підходи до аналізу фармацевтичних даних та їх обмеження

Упродовж багатьох років аналіз фармацевтичних даних базувався переважно на методах класичної статистики та експертної інтерпретації результатів. Такі підходи включали застосування кореляційного аналізу, дисперсійного аналізу, регресійних моделей. Вони дозволяли встановлювати зв’язки між параметрами лікарських засобів, оцінювати вплив окремих факторів на властивості препаратів та виникнення побочних ефектів. Завдяки цим методам була сформована основа для стандартизації аналітичних процедур, що використовуються у фармацевтичній промисловості та клінічній фармакології.

Однак зі збільшенням складності та обсягів фармацевтичних даних традиційні статистичні методи почали демонструвати низку істотних обмежень. Більшість класичних підходів розраховані на роботу з невеликими за обсягом вибірками, відносно низькою кількістю ознак та лінійними

залежностями між параметрами. У сучасних реальних умовах фармацевтичні дані містять сотні та тисячі характеристик, що описують властивості препаратів, умови їх виробництва, параметри пацієнтів та результати терапевтичних застосувань. Це призводить до ускладнення аналізу, а традиційні методи часто не здатні відтворити реальні закономірності, що існують у таких наборах даних.

Ще одним суттєвим недоліком класичних методів є їх слабка здатність до моделювання складних нелінійних взаємозв'язків. У фармацевтиці такі взаємозв'язки зустрічаються дуже часто: вплив дозування, взаємодія активних компонентів, сумісність речовин, вплив зовнішніх чинників виробничого середовища тощо можуть бути нелінійними, комбінаторними або прихованими. Для таких умов традиційні статистичні моделі часто надають некоректні або нестабільні результати, що знижує їх прогностичну цінність та обмежує можливість практичного використання.

Крім того, традиційні статистичні методи значною мірою залежать від глибокого залучення експертів доменної галузі. Кінцева якість таких методів часто визначається тим, наскільки дослідник зможе сформувати правильну гіпотезу, здійснити коректний підбір ознак та провести аналіз з урахуванням специфіки процесів. Це підсилює суб'єктивність аналітичних рішень і знижує можливість автоматизації процесу обробки даних на великих масштабах.

З огляду на перераховані обмеження, фармацевтична галузь потребує більш гнучких та масштабованих інструментів аналізу, які можуть працювати з великими обсягами інформації, виявляти складні закономірності, адаптуватися до різних типів даних та формувати точні прогностичні висновки. Саме з цих причин зростає зацікавленість у впровадженні методів машинного навчання та глибинних нейронних мереж, що відкриває можливість переходу від класичної статистики до інтелектуальних систем аналітики.

1.3. Сучасні підходи до аналізу фармацевтичних даних

На сучасному етапі розвитку фармацевтичної галузі методи машинного навчання (Machine Learning) та глибинних нейронних мереж (Deep Learning) починають відігравати ключову роль у побудові інтелектуальних систем аналізу та прогнозування. На відміну від традиційних статистичних методів, алгоритми машинного навчання здатні автоматично адаптуватися до даних, виявляти складні багатовимірні залежності та будувати моделі, які забезпечують високу точність прогнозів навіть у випадках, коли структури даних є нерівномірними, неповними або містять нелінійні закономірності. Це робить ML та DL одним із найперспективніших інструментів у фармацевтичних дослідженнях та фармацевтичній аналітиці.

Методи машинного навчання активно використовуються для вирішення широкого спектру задач у фармацевтиці, включаючи прогнозування фармакологічних властивостей субстанцій, виявлення потенційної токсичності, класифікацію лікарських препаратів, оцінку ризиків взаємодії між діючими речовинами та моделювання успішності лікування. Алгоритми ML, такі як логістична регресія, дерева рішень, ансамблеві моделі (випадковий ліс, градієнтний бустинг) демонструють високу точність при роботі з табличними даними середньої складності та можуть забезпечувати пояснюваність моделей, що важливо для середовища фармацевтики.

Глибокі нейронні мережі, у свою чергу, дають змогу розв’язувати ще складніші задачі, пов’язані з високою розмірністю ознак та необхідністю обробки даних різної природи від текстових описів до результатів біологічних тестів. Глибинне навчання забезпечує можливість автоматичного вилучення ознак, побудову глибинних абстракцій представлення об’єктів та моделювання складних нелінійних залежностей, що робить його гарним допоміжним інструментом для розв’язання дослідницьких завдань.

Розвиток сучасних фреймворків (TensorFlow, PyTorch, Scikit-learn) та покращення апаратного забезпечення обчислень суттєво прискорив процес навчання складних моделей та зробив їх використання доступним у

практичних дослідницьких та виробничих середовищах. Це сприяло переходу від експериментального застосування ML та DL до впровадження системи інтелектуальної аналітики у повноцінні бізнес-процеси фармацевтичних компаній.

Таким чином, застосування машинного навчання та глибинних нейронних мереж відкриває нові можливості для підвищення точності аналітичних висновків, автоматизації досліджень та підвищення рівня обґрунтованості фармацевтичних рішень. Це визначає актуальність подальшого вивчення ML та DL підходів та їх адаптації до специфіки фармацевтичних даних у даній магістерській роботі.

1.4. Аналіз існуючих рішень та досліджень

Рішення MEDICASCY створено з метою прогнозування побічних ефектів та режимів дії малих молекул, використовуючи тільки хімічну структуру ліків без потреби у великій кількості експериментальних даних.

Перевага цього рішення - мінімальні вхідні дані (лише структура молекули) й можливість застосування на ранніх етапах розробки препарату. Проте обмеження лежать у тому, що модель орієнтована на малі молекули та специфічні задачі (побічні ефекти, прогнозування росту клітин), і не охоплює повний спектр клінічних даних або виробничих процесів.

Рішення DeepDR являє собою бібліотеку глибинного навчання, орієнтовану на прогнозування відповіді на засіб на основі профілів мутацій та експресії генів в контексті ракових клітин чи пухлин.

Такий підхід має великий потенціал для персоналізованої медицини. Але слід врахувати, що дані - переважно для ракових моделей клітин, а не обов’язково для усіх фармацевтичних препаратів чи виробничих даних. Узагальнення на клінічні дані чи інші типи препаратів може бути складним.

Рішення ChemicalX - це бібліотека глибинного навчання для задач прогнозування взаємодій між двома препаратами, включно з побічними ефектами поліпрагмазії.

Таке рішення особливо актуальне для фармацевтичної практики, коли розглядаються комбіновані терапії та ризики побічних ефектів при одночасному застосуванні кількох препаратів. Основним обмеженням є те, що така система більше орієнтована на дослідницькі набори даних і комбінаторні сценарії, а не обов'язково на інтеграцію у виробничі процеси великих фармацевтичних компаній чи на практичну оцінку для кожного нового лікарського засобу.

Для зручного порівняння наявних рішень, певна інформація занесена у таблицю 1.1.

Таблиця 1.1

Таблиця порівняння сучасних рішень у сфері прогнозування побічних ефектів лікарських засобів

Рішення	**Завдання**	**Переваги**	**Обмеження**	**Доступність**
MEDICASCY	Прогнозування побічних ефектів	Мінімум вхідних даних, рання стадія застосування	Обмежена кількість типів даних, вузлька спеціалізованість	Комерційна інтеграція обмежена
DeepDR	Прогнозування відповіді на препарат	Висока точність у контрольованих дослідженнях	Дані орієнтовані на ракові клітини	Інтеграція можлива, але потребує технічного ресурсу
ChemicalX	Прогнозування взаємодій препаратів	Орієнтація на реальну проблему застосування багатьох ліків	Преважно дослідницький, потрібні великі набори даних	Необмежена, присутній відкритий код

Аналіз наявних рішень показує, що інструменти машинного та глибинного навчання для прогнозування побічних ефектів і успішного застосування лікарських засобів вже існують і мають підтверджену працездатність. Проте багато з них мають обмеження щодо загальної

застосовності, даних, масштабування або інтеграції у фармацевтичні процеси. Це створює можливість для нового дослідження - розробка системи, яка адаптована до фармацевтичних даних у дослідницькій сфері, проведення порівняння класичних моделей машинного навчання та глибоких нейронних мереж, зробити візуалізацію результатів і тим самим підвищити практичну придатність рішення.

1.5. Постановка задачі

На основі проведеного аналізу предметної області, розгляду сучасних підходів та оцінки існуючих наукових рішень у сфері застосування машинного навчання у фармацевтичній аналітиці, можна сформувати наукову проблему, що полягає у необхідності підвищення точності та водночас простоти аналітичних рішень при роботі з фармацевтичними даними. Існуючі підходи не завжди забезпечують оптимальний рівень достовірності моделей для прийняття рішень, а процес побудови моделей часто потребує адаптацій, оптимізації та порівняння різних алгоритмів для визначення найкращого методу прогнозування.

У межах даної магістерської роботи ставиться задача розробити підхід та програмний комплекс для аналізу фармацевтичних даних із застосуванням методів машинного навчання та глибинних нейронних мереж, а також провести порівняльну оцінку їхніх показників при вирішенні задач прогнозування, пов’язаних із фармацевтичною діяльністю.

Для досягнення поставленої мети необхідно виконати такі завдання:

- сформувати вибірку фармацевтичних даних, провести попередню підготовку, нормалізацію та очистку датасету;
- обрати та реалізувати набір алгоритмів машинного навчання (класичних та глибинних) для вирішення задачі прогнозування;
- здійснити навчання моделей та оцінити їхні показники на основі стандартних аналітичних метрик;
- провести порівняльний аналіз отриманих результатів, визначити переваги та недоліки моделей різного типу.

Реалізація поставленої задачі дозволить підвищити якість аналітичної обробки фармацевтичних даних, оптимізувати процес вибору моделі машинного навчання та сприятиме впровадженню інтелектуальних методів аналізу у фармацевтичну практику.

РОЗДІЛ 2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1. Об'єктне та функціональне моделювання

Функціональне моделювання є важливим етапом аналізу та проєктування програмних систем, оскільки дозволяє формалізувати логіку роботи майбутнього програмного продукту, визначити основні ролі користувачів, їх взаємодію із системою та ключові сценарії обробки даних. У контексті розробки системи для аналізу фармацевтичних даних із використанням методів машинного навчання та глибоких нейронних мереж, функціональне моделювання дає можливість відобразити повний цикл роботи - від внесення вихідних даних до формування прогнозів та отримання аналітичних звітів.

Для опису функціональних можливостей системи використовуються UML-діаграми, які дозволяють структурувати та візуально представити взаємозв’язки між користувачами, компонентами та процесами. Зокрема, у даному розділі буде описано діаграму прецедентів, яка відображає взаємодію зовнішніх акторів із системою та перелік основних функціональних можливостей. Така діаграма дозволяє визначити основні ролі (фармацевт-дослідник, аналітик, IT-інженер) та окреслити ключові операції, які вони виконують у процесі роботи з фармацевтичними даними.

Окрім того, на наступних етапах моделювання будуть сформовані діаграми послідовності, які деталізують поведінку системи та порядок взаємодії між її компонентами в межах окремих сценаріїв. Створення таких діаграм дозволяє отримати більш детальне уявлення про логічну структуру операцій і є підготовчим етапом до проєктування архітектури програмного забезпечення.

Таким чином, функціональне моделювання забезпечує основу для подальшої формалізації логіки роботи системи, створення діаграм та переходу до наступних етапів проєктування та реалізації магістерського дослідження.

2.1.1. Діаграма прецедентів

Діаграма прецедентів дозволяє формалізувати взаємодію основних категорій користувачів із системою аналізу фармацевтичних даних, визначити межі відповідальності кожного актора та описати ключові сценарії застосування системи. Такий підхід забезпечує чітке бачення того, які функціональні можливості надає розроблена система, хто саме ініціює певні процеси, які дані використовуються на вході та які результати формуються на виході див. Рис. 2.1.1.1.

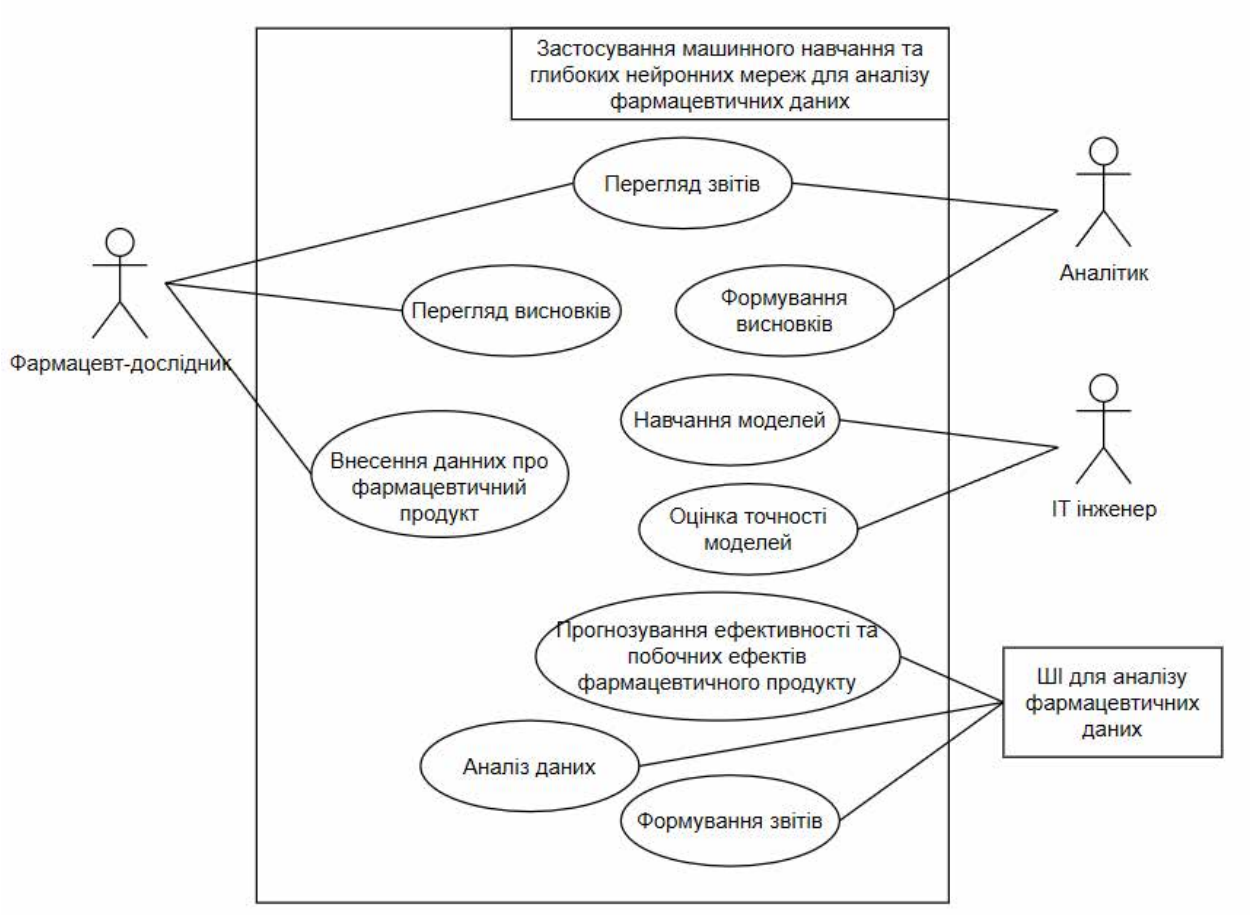

Рис. 2.1.1.1 Діаграма прецедентів

У системі виділено три основні актори: фармацевт-дослідник, аналітик та IT-інженер. Фармацевт-дослідник виконує внесення або оновлення даних про фармацевтичні продукти, а також здійснює перегляд результатів аналізу та сформованих звітів. Він є користувачем, який взаємодіє з системою на рівні прикладної фармацевтичної практики та результатів для подальшої роботи з лікарськими засобами. Аналітик опрацьовує результати прогнозування та

звіти, формує висновки щодо оцінки ефективності та потенційних ризиків лікарського засобу. Таким чином, він виконує інтерпретаційну та експертну функцію у загальному процесі обробки даних.

ІТ-інженер забезпечує технічне функціонування системи та роботу алгоритмів машинного навчання. Він проводить навчання моделей, оцінює якість отриманих результатів та контролює оновлення або реєстрацію нових версій моделей для забезпечення максимальної точності прогнозування. Внутрішня аналітична частина системи реалізує процеси обробки даних за допомогою алгоритмів машинного навчання та глибоких нейронних мереж: здійснює аналіз, отримує прогностичні оцінки та формує звіти з результатами.

Далі представлений детальний опис основних прецедентів системи:

Прецедент «Внесення даних про фармацевтичний продукт»

1.1 Передумови:

- Користувач має роль «Фармацевт-дослідник».
- Користувач авторизований у системі.
- Дані, які необхідно внести, є у готовому вигляді (структуровані або заповнювані через форму).

1.2 Тригер:

- Користувач ініціює дію додавання або редагування продукту через інтерфейс системи.

1.3 Головний потік:

1. Система відображає форму введення/редагування інформації по продукту.
2. Користувач вводить дані щодо складу, опису, фармакологічної групи, потенційних ефектів.
3. Система виконує первинну валідацію формату та обов’язкових полів (А1).

4. Дані зберігаються у базі та стають доступними для аналізу моделями ML.
5. Система повідомляє користувача про успішне збереження.

1.4 Альтернативні потоки:

- А1: Якщо обов’язкові поля не заповнено — система відображає помилку, користувач повертається до редактора.

Прецедент «Навчання моделей машинного навчання»

1.1 Передумови:

- Користувач має роль «IT інженер».
- База даних вже містить достатній обсяг даних.

1.2 Тригер:

- Інженер запускає процес тренування моделі через панель управління.

1.3 Головний потік:

1. Система отримує дані з бази.
2. Виконується підготовка/нормалізація даних (А1).
3. Система тренує модель ML або DL.
4. Виконується оцінка якості моделі за метриками (accuracy, precision, recall, F1-score).
5. Система зберігає модель та її параметри.

1.4 Альтернативні потоки:

- А1: Виникла помилка у структурі даних – система пропонує нормалізацію даних.
- Якщо метрики нижче порогу - система пропонує інженеру повторну конфігурацію.

Прецедент «Прогнозування ефективності та побочних ефектів лікарського засобу»

1.1 Передумови:

- Модель вже навчена та доступна у системі.

1.2 Тригер:

- Будь-який користувач ініціює прогноз (аналітик або фармацевт).

1.3 Головний потік:

1. Система отримує вхідні дані по обраному фармацевтичному продукту.
2. Модель виконує прогнозування ефективності та ймовірності виникнення побочних ефектів.
3. Система формує результати аналізу.

1.4 Альтернативні потоки:

- Якщо дані не повні - система повертає інформативне повідомлення про необхідність доповнення.

Таким чином, діаграма прецедентів описує загальну логіку взаємодії між користувачами та системою, демонструючи ключові функціональні сценарії, що забезпечують повний цикл обробки фармацевтичних даних - від внесення первинної інформації до отримання аналітичних висновків. Побудова такого представлення є важливим етапом моделювання, оскільки спрощує перехід до деталізації процесів, створення діаграм послідовності та подальшої розробки архітектури програмного забезпечення у наступних розділах.

2.1.2. Діаграма послідовності

Побудована діаграма послідовності демонструє покрокову взаємодію між користувачем, системою, базою даних та модулем машинного навчання під час виконання ключових операцій у системі аналізу фармацевтичних даних.

На початковому етапі користувач ініціює додавання нового фармацевтичного препарату. Система отримує введені дані та здійснює перевірку коректності полів. У випадку успішної валідації запис надсилається до бази даних, яка підтверджує збереження інформації. Таким чином формується основний набір знань системи, який надалі може бути використаний для навчання моделей. Даний процес відображено на відповідній діаграмі послідовності, що ілюструє обмін повідомленнями між користувачем та серверами даних див. Рис. 2.1.2.1.

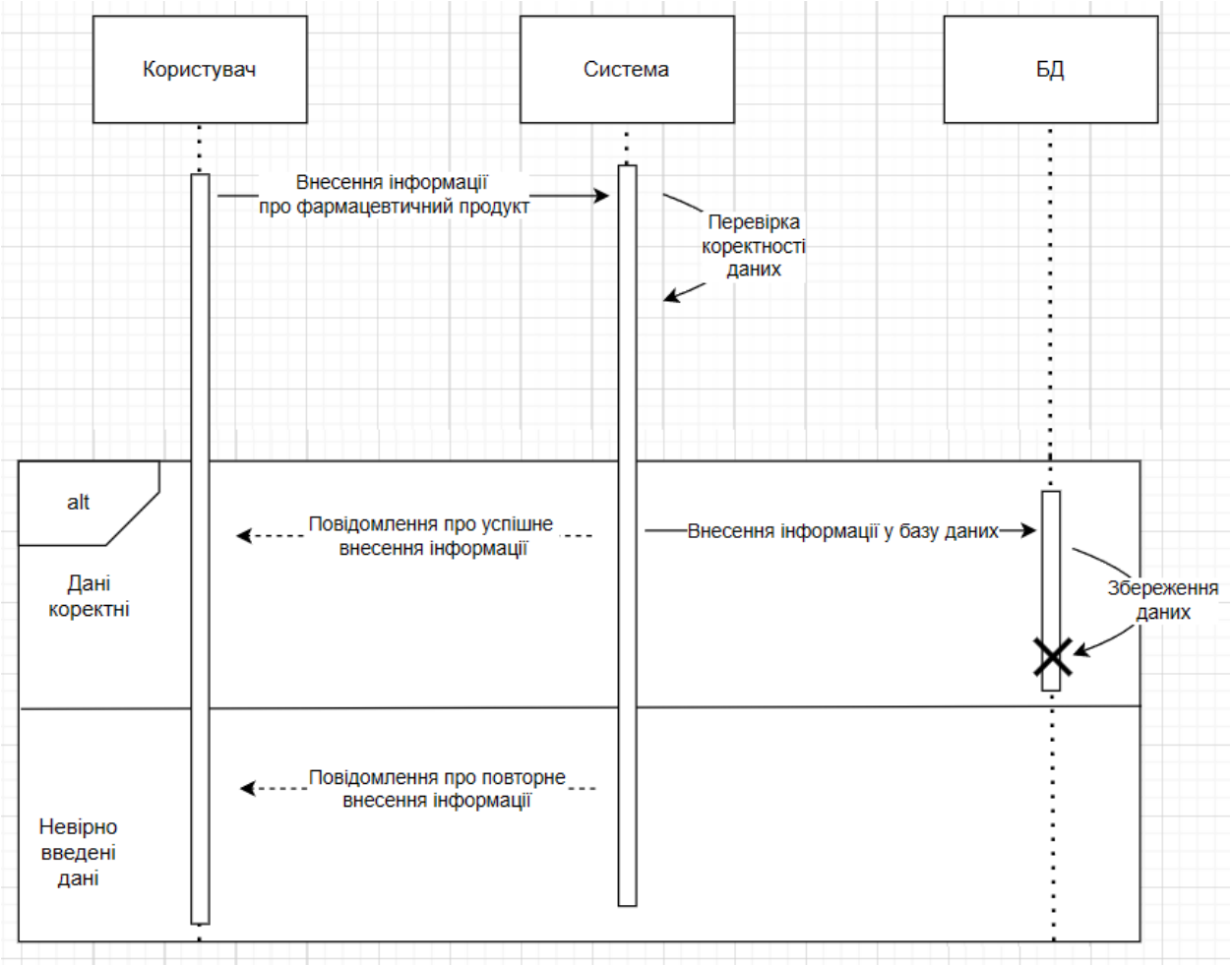

Рис. 2.1.2.1 Діаграмі послідовності «Внесення даних про фармацевтичний продукт»

Другий сценарій відображає етап навчання моделі на вже накопичених даних у базі. Після ініціації операції користувачем система формує запит до бази даних, обирає відповідні вибірки, передає їх до модуля машинного навчання, де відбувається процес тренування моделі. У разі успішного

завершення тренування результати повертаються назад системі та зберігаються в базі даних для подальшого аналізу та вибору оптимального алгоритму в майбутньому. Даний процес демонструє динаміку взаємодії між аналітичними модулями та дозволяє відстежити як саме формується історія моделей див. Рис. 2.1.2.2.

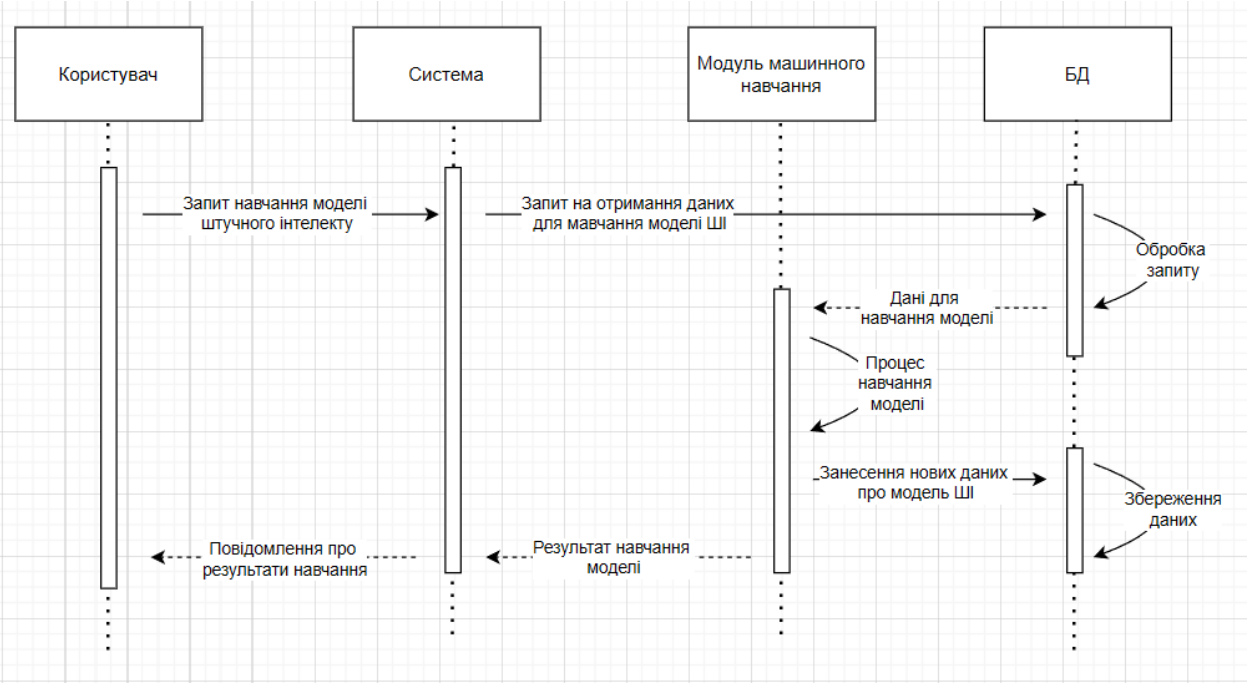

Рис. 2.1.2.2 Діаграмі послідовності «Навчання моделі штучного інтелекту»

Третій ключовий сценарій стосується процесу прогнозування, що є основною практичною цінністю системи. Після того як користувач обирає конкретний продукт і модель, система надсилає запит до бази даних, отримує необхідні вхідні дані та передає їх моделі машинного навчання для формування прогнозу. Результат прогнозу повертається системі, яка відображає його користувачу та зберігає отриманий аналітичний результат у базі даних. Таким чином, цей сценарій демонструє механізм прийняття рішень на основі попередньо навченої моделі та забезпечує можливість накопичення історії прогнозів для подальшого порівняння див. Рис. 2.1.2.3.

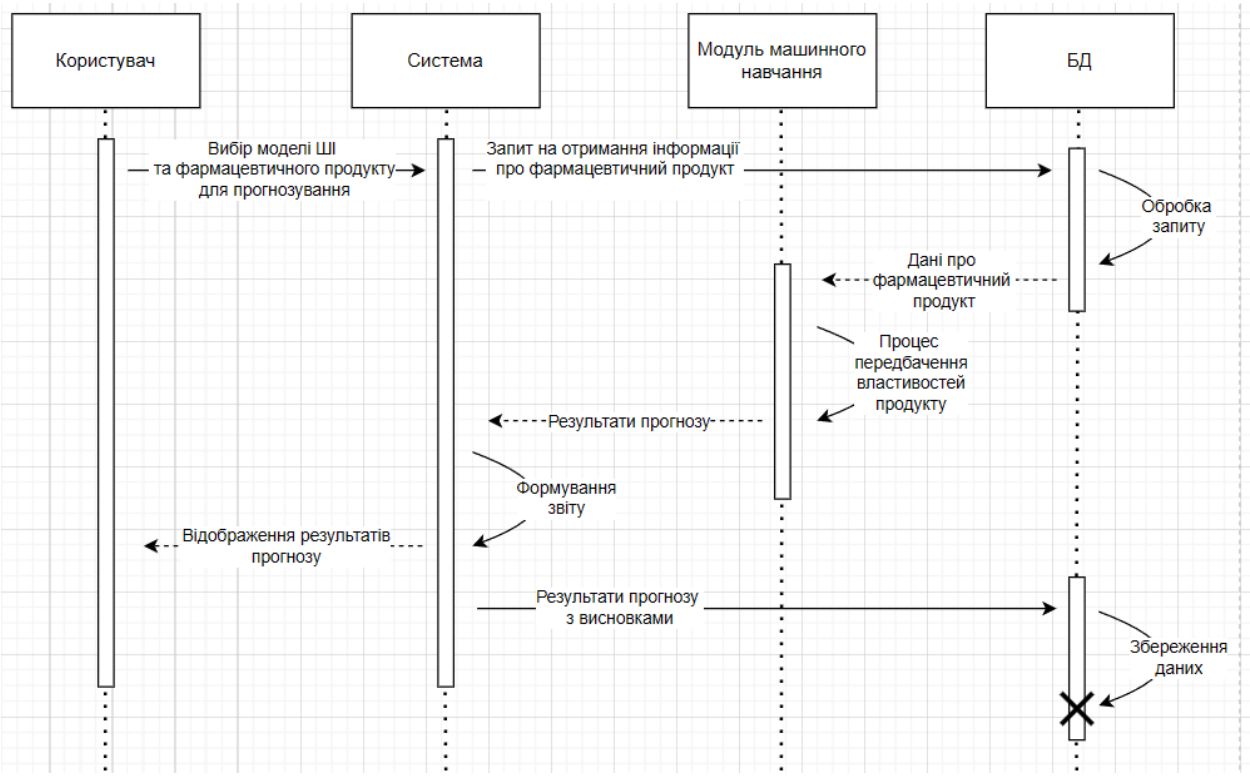

Рис. 2.1.2.3 Діаграмі послідовності «Прогнозування властивостей фармацевтичного продукту»

Описані три сценарії визначають повний цикл функціонування системи - від внесення даних до застосування моделей машинного навчання. На діаграмі послідовності можна чітко відобразити логічні переходи, етапи обміну даними та взаємодію між компонентами, що дозволяє у подальших розділах перейти до проєктування архітектури та реалізації компонентів системи.

2.1.3. Діаграма класів

Діаграма класів відображає об’єктно-орієнтований підхід до моделювання системи, визначає основні сутності, їхні атрибути та взаємозв’язки між ними. Цей тип моделювання дозволяє формалізувати структуру системи, окреслити логічні групи даних та компоненти, які беруть участь у процесі збору, обробки, аналізу та збереження фармацевтичної інформації. Діаграма класів є основою для подальшого проєктування бази даних, програмних модулів та архітектури системи див. Рис. 2.1.3.1.

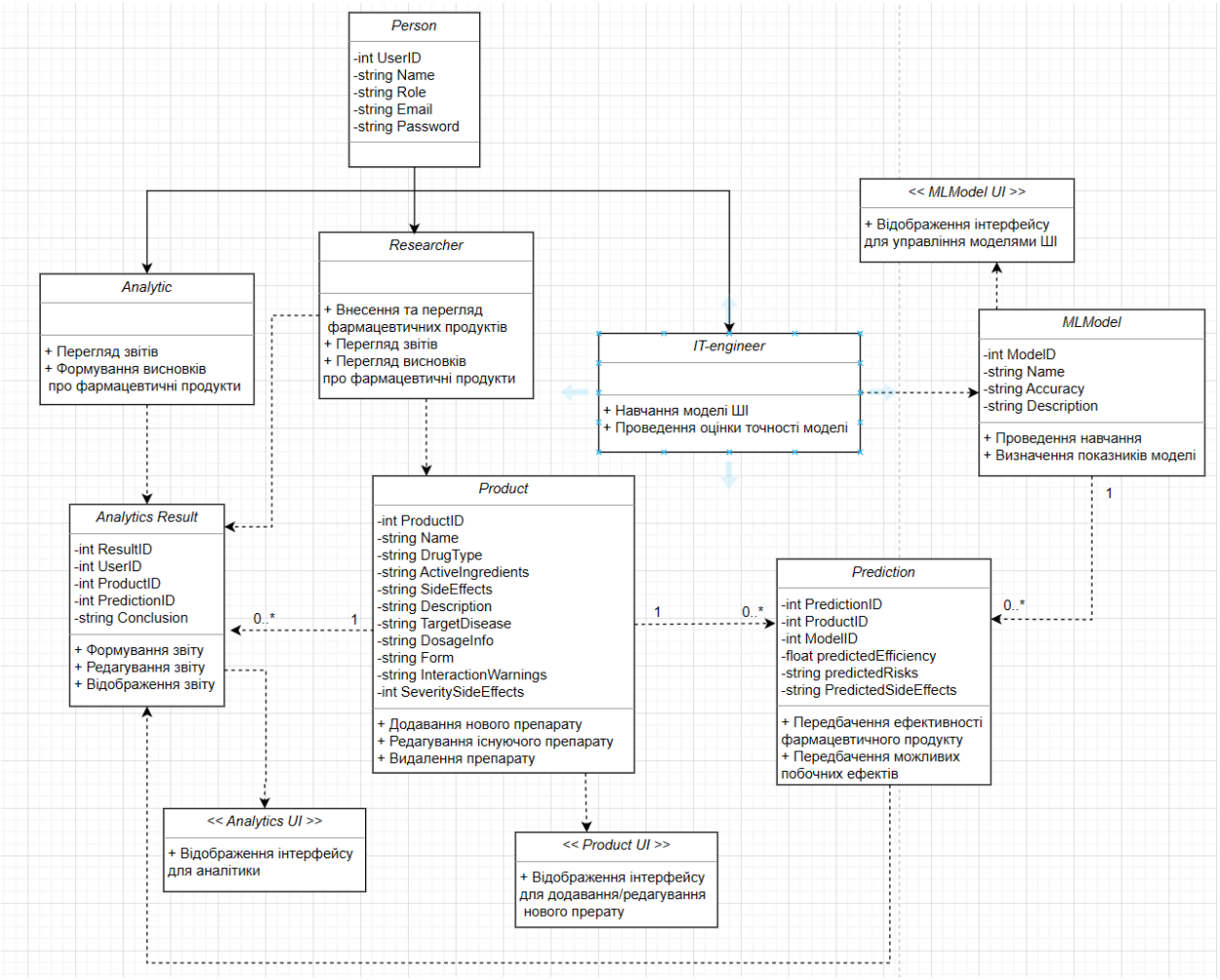

Рис. 2.1.3.1 Діаграма класів

У розробленій системі виділено наступні основні класи:

Person - базовий клас користувача системи. Атрибути: UserID, Name, Surname, Role, Email, Password. Взаємозв’язки: виступає батьківським класом для Researcher, Analytic та IT-engineer, від яких успадковуються загальні атрибути користувача.

Researcher - клас користувача, що відповідає за внесення даних про фармацевтичні продукти. Методи: внесення нових продуктів, перегляд продуктів та перегляд висновків щодо препаратів. Взаємозв’язки: взаємодіє з класом Product та може ініціювати створення AnalyticsResult.

Analytic - клас користувача, що здійснює аналіз прогнозів та формування висновків. Методи: перегляд звітів та формування висновків щодо фармацевтичних продуктів. Взаємозв’язки: взаємодіє з результатами моделей (Prediction) та створює записи AnalyticsResult.

IT-engineer - клас користувача, що виконує навчання та оцінку точності моделей машинного навчання. Методи: навчання моделі та проведення оцінки. Взаємозв’язки: взаємодіє з класом MLModel, оновлює та керує його параметрами.

Product - клас фармацевтичного продукту. Атрибути: ProductID, Name, DrugType, ActiveIngredients, SideEffects, Description. Методи: додавання нового препарату, редагування продукту та видалення. Взаємозв’язки: один продукт може мати декілька прогнозів (Prediction) та аналітичних результатів (AnalyticsResult).

MLModel - клас, що представляє модель машинного навчання. Атрибути: ModelID, Name, Accuracy, Description. Методи: проведення навчання та визначення показників моделі. Взаємозв’язки: модель використовується для формування прогнозів (Prediction).

Prediction - клас, що містить прогноз, сформований на основі MLModel. Атрибути: PredictionID, ProductID, ModelID, PredictedEfficiency, PredictedRisks. Методи: прогнозування ефективності та можливих побічних ефектів. Взаємозв’язки: один продукт може мати кілька прогнозів; кожний прогноз формується конкретною MLModel.

AnalyticsResult - клас сформованих аналітичних висновків. Атрибути: ResultID, UserID, ProductID, PredictionID, Conclusion. Методи: формування звіту, редагування та відображення звіту. Взаємозв’язки: пов’язаний із користувачем (Person), продуктом (Product) та прогнозом (Prediction).

Діаграма класів дозволяє наочно представити структуру системи, визначити об’єктні залежності та забезпечує основу для побудови бази даних та реалізації логіки обробки даних у програмному забезпеченні. Використання такого підходу забезпечує зрозумілу ієрархію сутностей, спрощує підтримку системи та інтеграцію нових функціональних модулів у майбутньому.

РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ

3.1. Архітектура розробленої системи

Архітектура розробленої системи передбачає модульну побудову з чітким поділом на логічні підсистеми, які взаємодіють між собою через механізми передачі даних та інтерфейси. Така структура забезпечує масштабованість, можливість незалежного розвитку окремих компонентів, адаптацію під нові моделі машинного навчання, а також спрощує підтримку та модернізацію системи у майбутньому.

Система складається з декількох основних підсистем: підсистеми зберігання даних, підсистеми машинного навчання та моделювання, підсистеми прогнозування та аналітичної підсистеми. Централізованим джерелом даних є база даних PharmacyData, де зберігається інформація про фармацевтичні продукти, результати тренування моделей, сформовані прогнози та аналітичні висновки. Підсистема машинного навчання використовує ці дані для тренування моделей, збереження їх параметрів, а також формування прогнозів для лікарських засобів на основі даних.

Підсистема прогнозування відповідає за обчислення прогнозних показників ефективності та ризиків виникнення побочних ефектів для кожного лікарського засобу, використовуючи результати навчання моделей. Аналітична підсистема забезпечує формування висновків на основі отриманих результатів прогнозування та надає можливість експертам аналізувати отримані дані, редагувати аналітичні висновки, проводити порівняння між моделями та продуктами. Взаємодія користувачів з системою реалізується через інтерфейси, розроблені для різних ролей користувачів: дослідник, аналітик, IT-інженер. Кожен тип користувачів має доступ лише до тих функцій, які відповідають його ролі у системі.

Загалом, архітектура системи забезпечує цілісну інтеграцію процесів роботи з фармацевтичними даними - від зберігання первинної інформації та навчання моделей до побудови прогнозів та формування аналітичних

висновків для підтримки прийняття рішень у фармацевтичній сфері. Такий підхід дозволяє підвищити точність прогнозування та автоматизувати роботу з великими масивами фармацевтичних даних.

3.2. Модуль зберігання фармацевтичних даних

3.2.1. Створення бази даних

База даних відповідає за організацію структурованого зберігання інформації про користувачів, ролі, фармацевтичні продукти, моделі машинного навчання, результати прогнозування та аналітичні висновки. Вона є важливою частиною всієї системи, оскільки забезпечує доступність даних для всіх інших програмних компонентів див. Рис. 3.2.1.1.

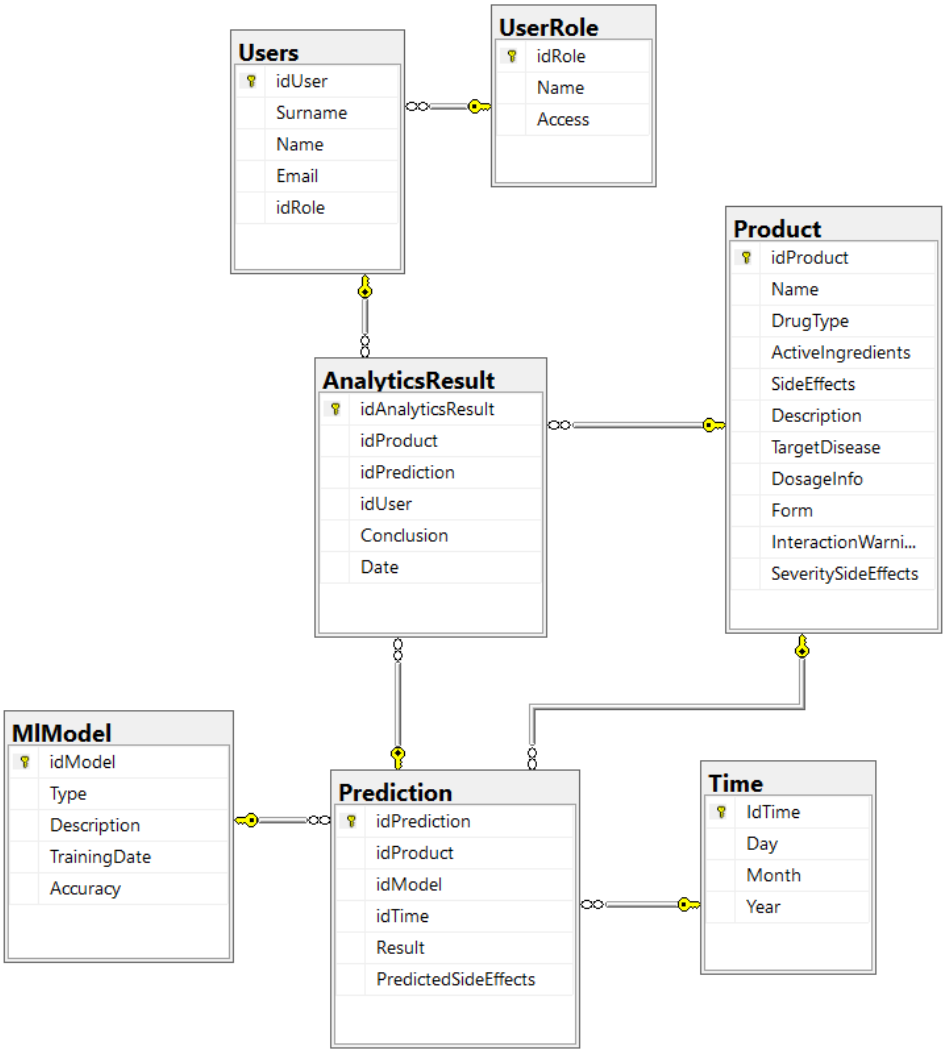

Рис. 3.2.1.1 Схема бази даних для зберігання інформації

Таблиця **UserRole** містить інформацію про ролі користувачів, які можуть використовувати систему.

- idRole - унікальний ідентифікатор ролі.
- Name - назва ролі (аналітик, дослідник, інженер).
- Access - рівень доступу та дозволені операції для ролі.

Таблиця **Users** містить інформацію про користувачів.

- idUser - унікальний ідентифікатор користувача.
- Surname, Name - ПІБ користувача.
- Email - контактна пошта.
- idRole - роль користувача у системі (зв’язок на UserRole).

Таблиця **MlModel** містить інформацію про моделі машинного навчання, які використовуються для прогнозування властивостей фармацевтичних препаратів.

- idModel - унікальний ідентифікатор моделі машинного навчання.
- Type - тип моделі.
- Description - опис моделі або її призначення.
- TrainingDate - дата навчання моделі.
- Accuracy - показник точності моделі після тренування.

Таблиця **Product** містить детальну інформацію про фармацевтичні препарати.

- idProduct - унікальний ідентифікатор лікарського препарату.
- Name - назва препарату.
- DrugType - тип препарату або фармакологічна група.
- ActiveIngredients - діючі речовини.
- SideEffects - відомі побічні ефекти.
- Description - загальний опис препарату.
- TargetDisease – захворювання, яке лікує препарат.
- DosageInfo - інформація про дозування.
- Form - форма препарату (таблетки, капсули і тд.).

- InteractionWarnings - попередження щодо взаємодій з іншими препаратами.
- SeveritySideEffects - рівень важкості побічних ефектів.

Таблиця **Prediction** містить інформацію про результат передбачення властивостей препаратів.

- idPrediction - унікальний ідентифікатор прогнозу.
- idProduct - препарат, для якого зроблено прогноз.
- idModel - модель, яка зробила прогноз.
- Result - прогнозована ефективність препарату.
- PredictedSideEffects - прогнозовані можливі побічні ефекти.
- Date - дата формування прогнозу.

Таблиця **AnalyticsResult** містить висновки аналітика щодо результатів прогнозування властивостей препарату.

- idAnalyticsResult - унікальний ідентифікатор результату.
- idProduct – препарат, для якого формується висновок.
- idPrediction – ідентифікатор прогнозу.
- idUser - користувач, який сформував висновок.
- Conclusion - аналітичний висновок за результатами аналізу.
- Date - дата формування висновку.

Усі результати прогнозування та аналітики зберігаються з відповідними часовими мітками, що створює додаткові можливості для порівняння, а також відстеження тенденції зміни ефективності продуктів та показники моделей.

Таким чином, база даних необхідна для правильного функціонування алгоритмів машинного навчання, процесів моделювання та подальшого прийняття рішень у фармацевтичних дослідженнях.

3.2.2. Створення сховища даних

Сховище даних необхідне для зберігання великих обсягів інформації, структуризує її та забезпечує можливість аналізу. Сховище даних необхідне

для підтримки аналітики та прийняття рішень. Воно зберігає історичну інформацію, що дозволяє відстежувати зміни показників і виявляти закономірності у великих наборах даних.

У контексті фармацевтичної галузі сховище даних дозволяє об’єднати інформацію про лікарські засоби, результати роботи моделей машинного навчання та часові показники, що забезпечує повноцінну базу для дослідження властивостей препаратів і виявлення закономірностей.

Для побудови багатовимірного куба було використано середовище розробки Visual Studio із застосуванням компонента SSAS. Для створення кубу спочатку потрібно підключити джерело даних. Саме це джерело використовувалося як основа для завантаження необхідної інформації до системи. На рисунках 3.2.2.1–3.2.2.2 наведено приклад процесу налаштування з’єднання з джерелом даних, де обиралося попередньо створене сховище.

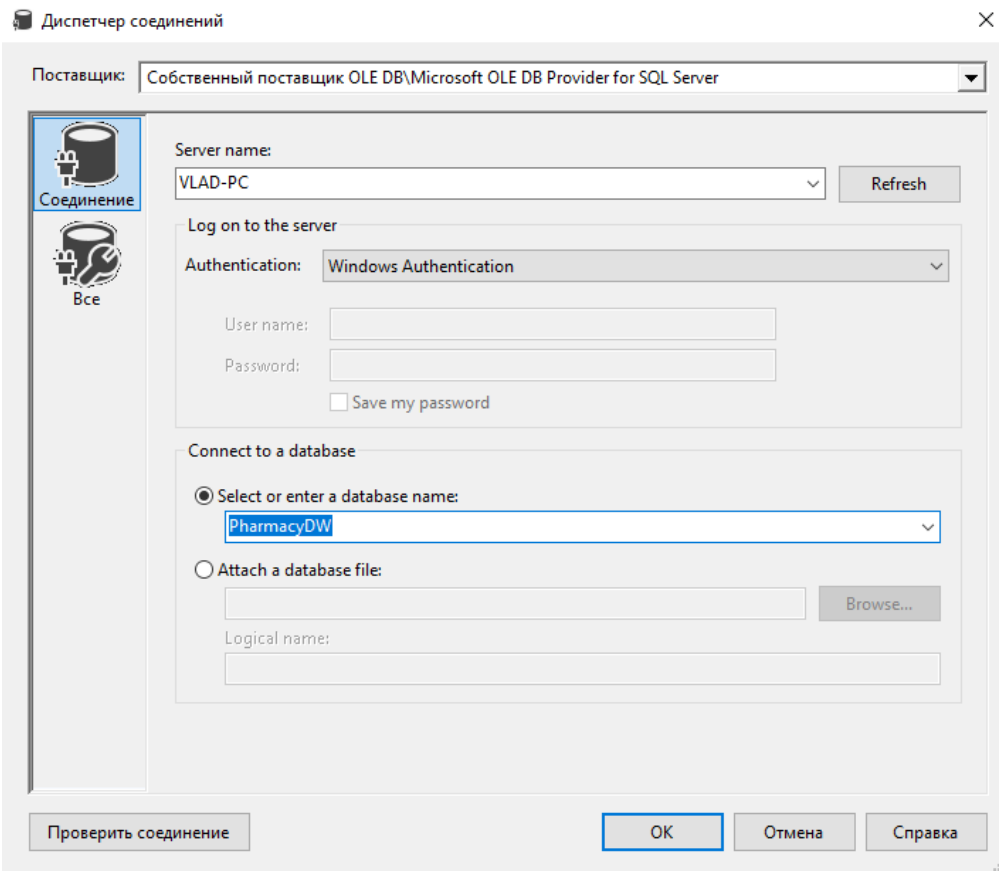

Рис. 3.2.2.1 Процес налаштування підключення до джерела даних

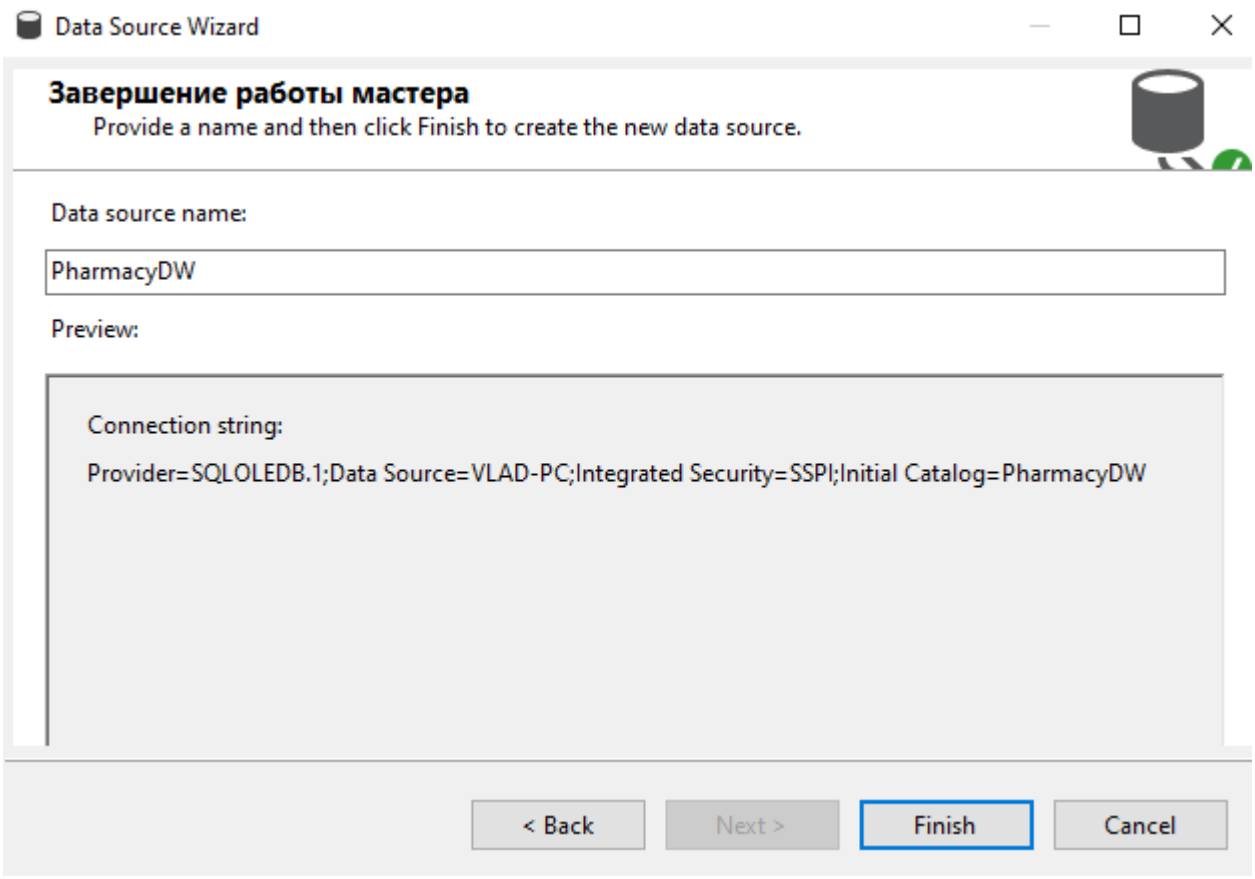

Рис. 3.2.2.2 Завершення процесу підключення джерела даних

Наступним кроком у процесі розгортання куба є створення представлення джерела даних. Це забезпечує зручне відображення структури бази даних і використовується як основа для подальшого формування вимірів і самого кубу.

Основне призначення цього представлення джерела даних полягає у створенні гнучкої моделі даних, яка дозволяє виконувати зміни, що не впливають на початкові таблиці. Наприклад, можна перейменовувати поля, створювати обчислювані стовпці, формувати віртуальні зв’язки або об’єднувати таблиці, що надає нові можливості для аналізу наявної інформації.

Процес налаштування представлення джерела даних та визначення необхідних зв’язків між таблицями показано на рисунках 3.2.2.3–3.2.2.4.

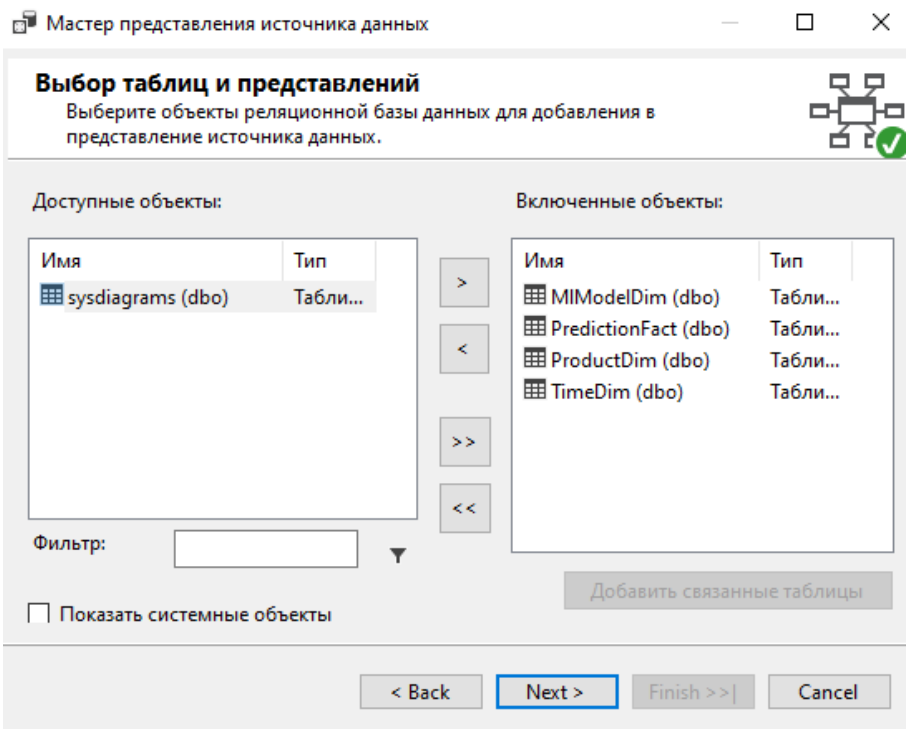

Рис. 3.2.2.3 Вибір таблиць для представлення джерела даних

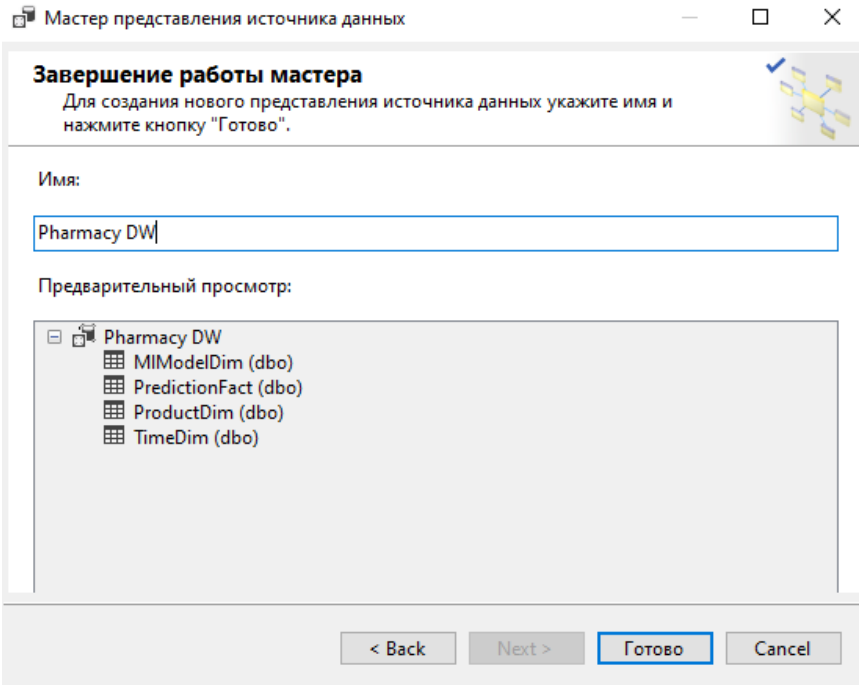

Рис. 3.2.2.4 Підтвердження створення

Завершальним етапом у формуванні кубу є його обробка. У результаті обробки здійснюється побудова структур даних, що дозволяє здійснювати подальший аналіз, формувати звіти та виконувати запити без безпосереднього звернення до реляційної бази даних див. Рис. 3.2.2.5.

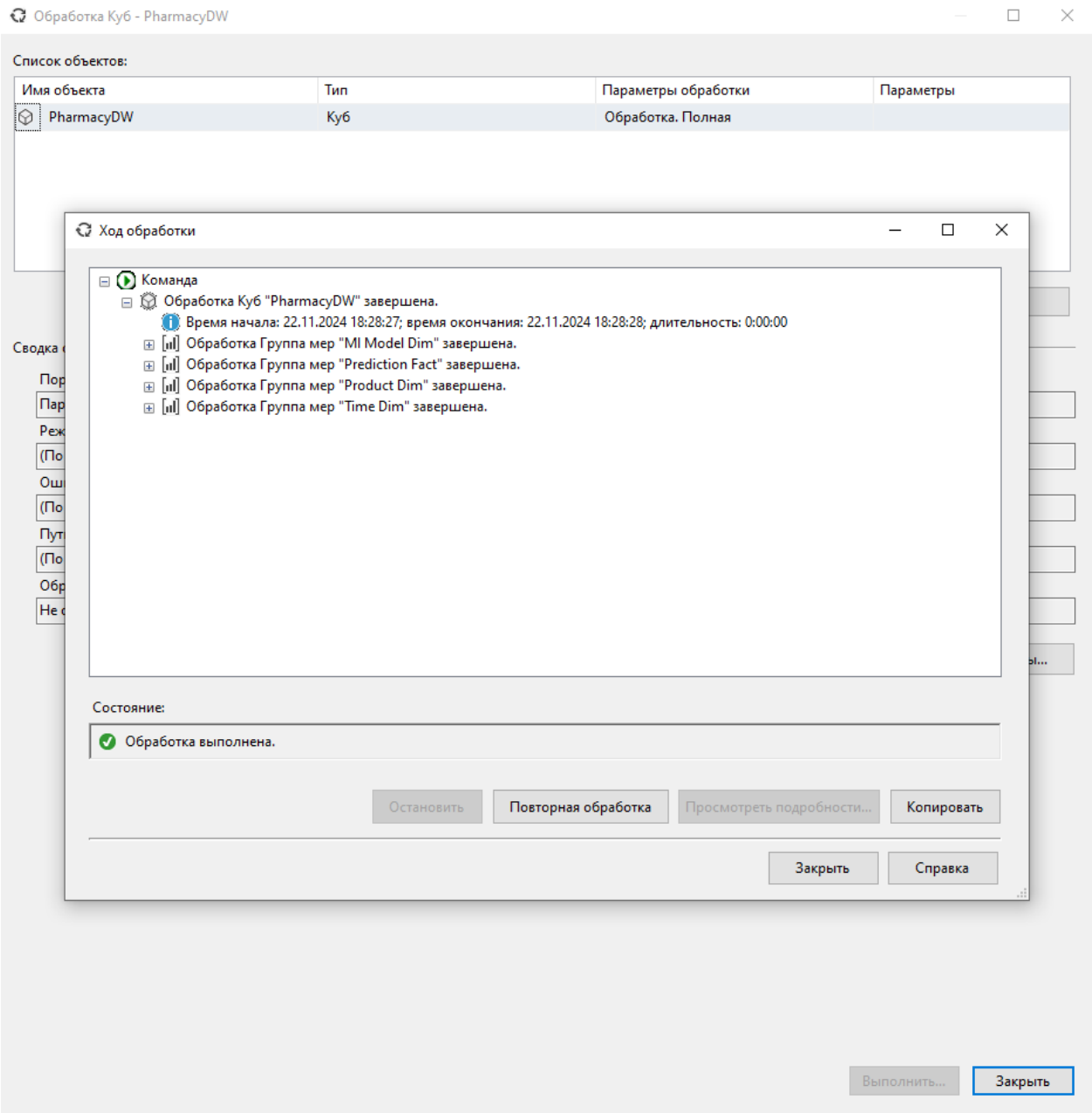

Рис. 3.2.2.5. Обробка кубу

У створеній системі сховище даних побудовано за класичною зірковою схемою, де центральна таблиця фактів пов’язана з кількома вимірами. Така архітектура спрощує виконання аналітичних запитів, дозволяє швидко обчислювати узагальнення та забезпечує таку ж стабільну роботу при збільшенні обсягів даних див. Рис. 3.2.2.6.

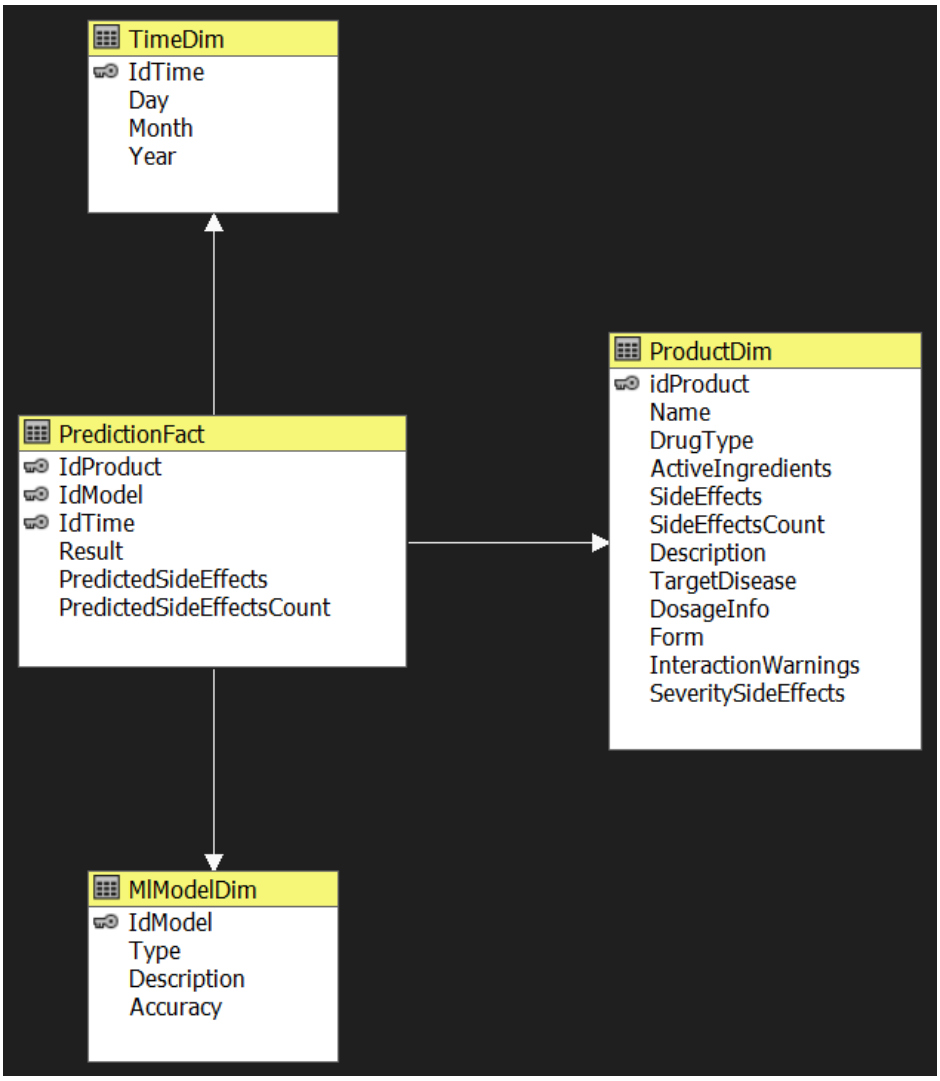

Рис. 3.2.2.6. Сформований куб

Таблиця **MlModelDim** зберігає інформацію про моделі машинного навчання, які використовуються для прогнозування ефективності та побічних ефектів лікарських засобів. Поля таблиці містять ідентифікатор моделі (IdModel), тип алгоритму (Type), додатковий опис (Description) та показник точності (Accuracy), який характеризує якість навчання моделі. Завдяки цим даним аналітична система може порівнювати результати різних моделей і відстежувати, яка з них демонструє кращі прогностичні властивості.

Таблиця **TimeDim** призначена для зберігання часових вимірів, що дозволяє аналізувати динаміку прогнозів у часі. Вона містить поля IdTime, Day, Month та Year, які забезпечують можливість побудови звітів за днями, місяцями чи роками. Завдяки цьому можна простежити, як змінюються показники препаратів або точність моделей протягом певного періоду.

Таблиця **ProductDim** містить інформацію про лікарські препарати. Тут містяться дані про назву препарату (Name), його тип (DrugType), діючі речовини (ActiveIngredients), опис (Description), захворювання яке лікує препарат (TargetDisease), форму (Form), дозування (DosageInfo), а також попередження щодо взаємодій (InteractionWarnings). Додаткові поля, такі як SideEffects, SideEffectsCount і SeveritySideEffects, містять дані про побічні ефекти та їхню інтенсивність, що дає змогу проводити детальні аналітичні дослідження ризиків лікарських засобів.

Таблиця **PredictionFact** є головною таблицею, у ній зберігаються фактичні результати прогнозів, отримані від моделей машинного навчання. Поля IdProduct, IdModel і IdTime визначають контекст кожного прогнозу - для якого препарату, якою моделлю та коли він був виконаний. Поле Result відображає прогнозований рівень ефективності препарату, тоді як PredictedSideEffects та PredictedSideEffectsCount містять текстовий опис і кількісну оцінку можливих побічних ефектів.

3.2.3. Заповнення сховища даних

Передача даних до сховища була реалізована за допомогою SQL Server Integration Services (SSIS). Вона дозволяє автоматизувати процеси перенесення інформації, гарантуючи її узгодженість, цілісність, а також високу швидкість обробки.

Ключовим елементом у межах SSIS є механізм Data Flow, який виконує основні операції з обробки та перенесення даних. Саме він відповідає за наповнення таблиць вимірів і фактів у сховищі, використовуючи інформацію, отриману з бази даних. Завдяки цьому забезпечується перехід від транзакційної структури до аналітичної моделі, що використовується для подальшого аналізу даних.

Загальна структура потоків даних і приклади їх реалізації для заповнення відповідних таблиць наведені на рисунках 3.2.3.1 - 3.2.3.2.

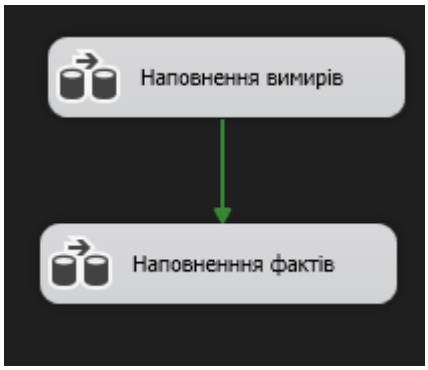

Рис.3.2.3.1 Загальна структура потоків

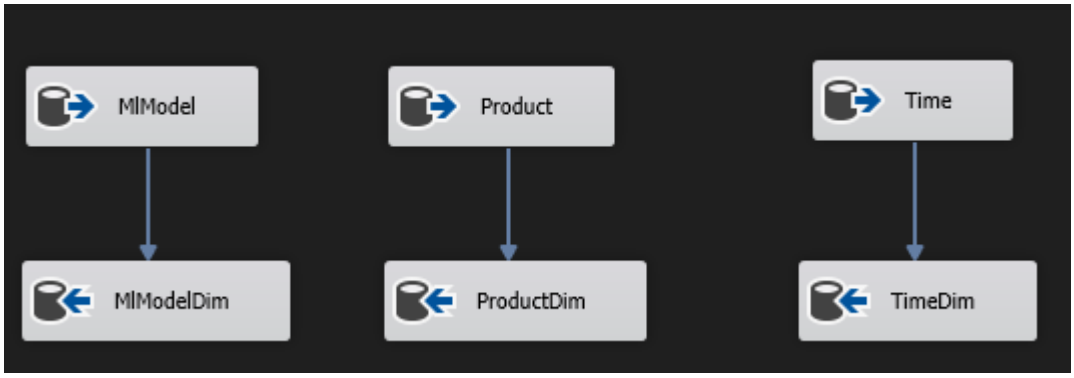

Рис.3.2.3.2 Структура потоків наповнення вимірів

Для кожної таблиці потрібно становити співставлення стовпців, щоб дані коректно завантажились у сховище даних. У разі розбіжностей у типах даних або форматах значень можуть застосовуватись додаткові зміни у структурі наприклад, зміна типу даних, об’єднання кількох полів, перейменування або створення додаткових значень, які будуть вираховуватись. Це дозволяє стандартизувати структуру даних і підготувати їх до подальшого використання рис. 3.2.3.3.

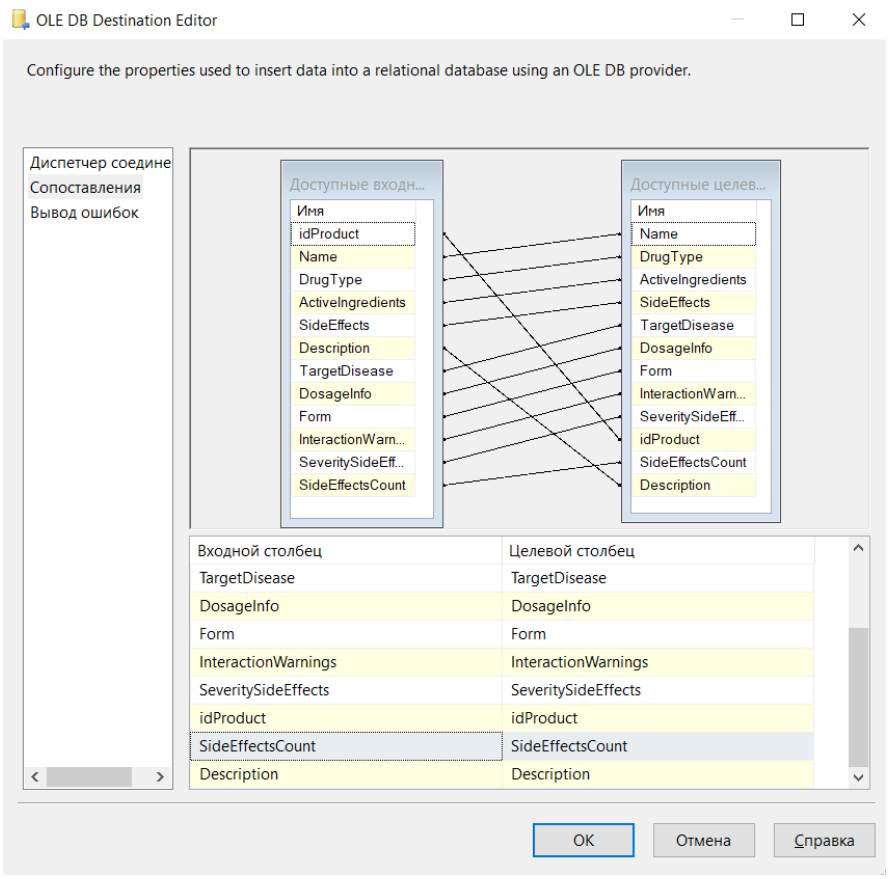

Рис.3.2.3.3 Співставлення стовпців у таблицях

Наступним кроком є заповнення таблиці фактів, яке має схожий алгоритм з наповненням таблиць вимірів, проте виконується на основі вже підготовлених і завантажених даних. Під час формування фактів система використовує зв’язки між вимірами, які було створено раніше, для забезпечення коректного зіставлення ключів і цілісності даних див. Рис. 3.2.3.4 – 3.2.3.5.

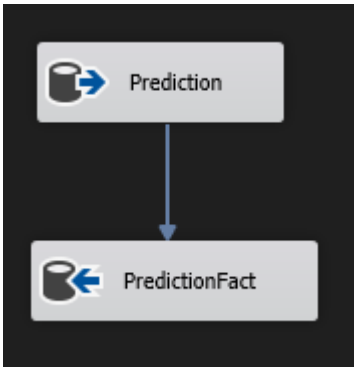

Рис.3.2.3.4 Вигляд потоку заповнення фактів

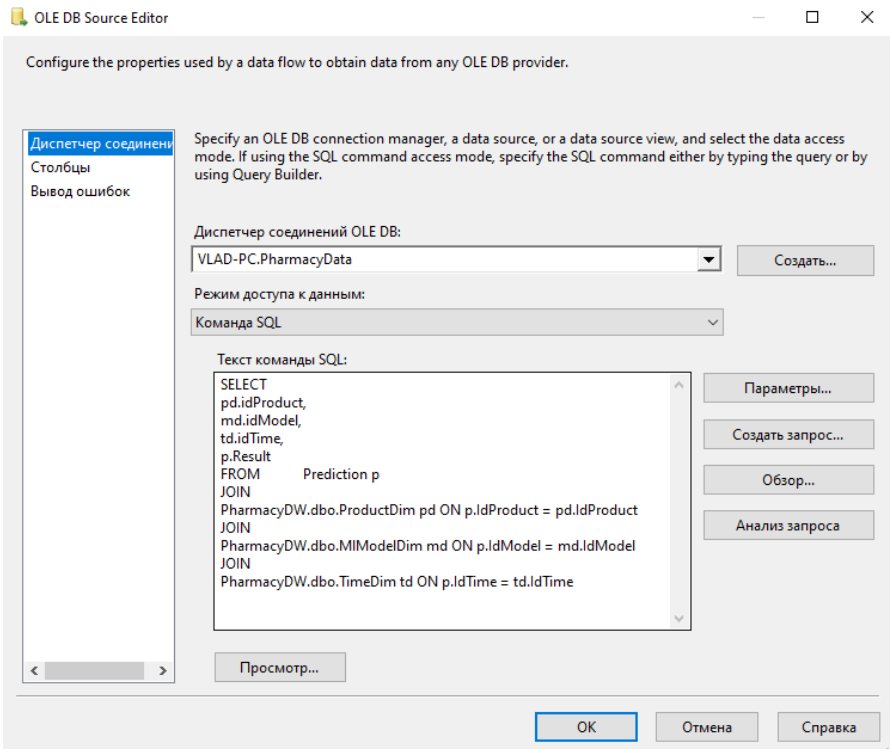

Рис.3.2.3.5 Вибірка для заповнення фактів

В результаті виконання дані завантажуються з бази даних у сховище даних. Завдяки цьому інформація набуває узгодженого вигляду, що дає змогу використовувати її для подальшого аналізу див. Рис. 3.2.3.6.

Рис.3.2.3.6 Успішне завантаження інформації до сховища даних

3.3. Модуль машинного навчання

Підсистема моделювання та машинного навчання є ключовим компонентом системи, що забезпечує інтелектуальну обробку фармацевтичних даних та формування прогнозних висновків щодо ефективності лікарських засобів і ймовірності виникнення побічних ефектів. Вона включає набір алгоритмів машинного навчання та глибинну нейронну

мережу, які аналізують відомі характеристики препаратів та формують прогноз на основі математичних закономірностей, виділених із даних.

3.3.1. Алгоритми машинного навчання

У роботі використовуються декілька різних алгоритмів машинного навчання, що дозволяють порівняти їх точність та придатність для задач фармацевтичної аналітики. Кожен з алгоритмів має різні підходи до побудови прогнозів, різні вимоги до структури даних та різні властивості узагальнення.

Логістична регресія є базовим лінійним алгоритмом класифікації, який перетворює вихідний результат у значення ймовірності приналежності об'єкта до певної категорії. У фармацевтичному аналізі вона може бути використана для прогнозування ймовірності ефективності препарату або настання побічної реакції на основі заданих ознак. Перевагою алгоритму є хороша інтерпретованість результатів та швидкість навчання, в свою чергу недоліком є обмежена здатність виявляти складні нелінійні залежності.

Дерева рішень будують послідовну структуру із вузлів, в яких здійснюється розподіл даних за правилами вибору ознак. Результатом є дерево умов, кожне з яких веде до кінцевого класу. Даний алгоритм добре підходить для фармацевтики, оскільки дозволяє зрозуміти логіку прийняття рішення, а саме які параметри продукту найбільше впливають на ефективність або виникнення побочних ефектів препарату. Слабка сторона цього алгоритпу полягає в тому, що дерева можуть легко перенавчатися, особливо на невеликій кількості даних.

Випадковий ліс складається з набору незалежних дерев рішень, результати яких комбінуються шляхом голосування. Такий підхід дозволяє значно зменшити ризик перенавчання, оскільки кожне дерево вивчає дані по-своєму на випадкових вибірках. Алгоритм добре масштабується, має високу точність і використовується в задачах, де дані мають складні залежності та змішані типи ознак. У даній роботі Random Forest використовується як один з базових алгоритмів.

Глибока нейронна мережа застосовує багатошарову структуру штучних нейронів, які здатні виділяти ієрархію особливостей у даних. На відміну від класичних моделей, нейронна мережа не потребує ручного створення правил, вона самостійно навчається виявляти приховані залежності, що недоступні для традиційних методів. Завдяки цьому вона здатна обробляти складні фармацевтичні дані, де ефективність препарату часто визначається взаємодією кількох факторів одночасно. У дослідженні нейронна мережа показала найвищу точність прогнозування.

3.3.2. Процес тренування моделей

Процес тренування моделей машинного навчання та глибинних нейронних мереж є повторюваним та необхідним процесом, щоб можна було з більшою точністю прогнозувати властивості фармацевтичних продуктів.

Основна мета полягає у виявленні складних математичних залежностей між ознаками лікарських препаратів та кінцевими результатами, серед яких ймовірність виникнення побічних ефектів та ризики небажаних взаємодій. Сформована модель має не просто запам'ятовувати історичні дані, але й узагальнювати вивчені закономірності для надійного прогнозування на раніше невідомих даних, що є ключовим для підтримки прийняття рішень у розробці нових препаратів.

На цьому етапі з бази даних виділяються структуровані записи, що описують лікарські засоби, їхні властивості та інші дані. Критично важливим кроком є інженерія ознак, зокрема, перетворення текстової інформації (наприклад діюча речовина, доза препарату, можливі побочні ефекти і тд.) у числові вектори, зрозумілі для алгоритмів. Для цього можуть застосовуватися методи як "мішок слів" (Bag of words), так і складніші техніки, зокрема векторні представлення слів (Word Embeddings), які зберігають семантичні зв'язки між термінами.

Для об'єктивної оцінки здатності моделі до узагальнення, весь доступний датасет випадковим чином розбивається на три незалежні підмножини.

Тренувальна вибірка, найбільша за обсягом, безпосередньо використовується для корекції внутрішніх параметрів моделі шляхом мінімізації функції втрат.

Валідаційна вибірка необхідна для проміжного контролю та вибору оптимальних гіперпараметрів моделі, вона допомагає визначити, коли модель починає перенавчатися на тренувальних даних, і зупинити процес вчасно.

Тестова вибірка в свою чергу містить незнайомі дані і використовується лише один раз після завершення всього циклу навчання для фінальної перевірки моделі.

Цей етап визначає, як саме модель буде вчитися. Для класичних алгоритмів машинного навчання (наприклад, дерева рішень, логістична регресія) проводиться налаштування їхніх гіперпараметрів, наприклад глибина дерева.

У випадку глибинних нейронних мереж завдання ускладнюється, необхідно визначити саму архітектуру мережі, кількість та розмір прихованих шарів, також визначити швидкість навчання. Пошук найкращої конфігурації часто автоматизується за допомогою таких методів, як сітковий пошук або випадковий пошук.

Власне навчання - це циклічний процес, де модель поступово покращує свою точність. На кожному етапі модель прогнозує значення для тренувальних даних, після чого обчислюється функція втрат, яка кількісно оцінює величину помилки прогнозу.

Для мінімізації цієї помилки використовується механізм зворотного поширення помилки, а саме градієнт функції втрат поширюється від вихідного шару до вхідного, що дозволяє обчислити, наскільки кожен параметр моделі зробив свій внесок у загальну помилку. Далі, за допомогою алгоритму оптимізації (наприклад, градієнтного спуску), ваги моделі оновлюються в напрямку, протилежному градієнту, що веде до поступового зменшення помилки.

Однією з найважливіших проблем у машинному навчанні є перенавчання моделі, коли вона демонструє ідеальні результати на тренувальних даних, але погано працює на нових прикладах. Для боротьби з цим використовується комплекс підходів.

Для запобігання перенавчанню також використовується метод перехресної перевірки, яка забезпечує більш надійну оцінку якості моделі, в потрібний момент вона зупиняє тренування, коли продуктивність на вибірці перестає покращуватися, що в свою чергу допомагає уникнути запам'ятовуванню непотрібного шуму.

При навчанні нейронних мереж також застосовується випадкове "вимкнення" частини нейронів під час тренування, що змушує нейронну мережу вчитися за допомогою узагальнених ознак. Також дієвим інструментом проти перенавчання є регуляризація, вона працює за рахунок додавання спеціального штрафного балу до функції втрат моделі, яке зростає пропорційно до величини її вагових коефіцієнтів. Це змушує алгоритм навчання не лише мінімізувати похибку прогнозу, але й підтримувати значення вагових коефіцієнтів якнайменшими. У результаті модель стає менш схильною до запам'ятовування шуму в тренувальних даних і, як наслідок, краще узагальнює закономірності на нових, невідомих даних.

Після завершення процесу тренування та налаштування всі моделі проходять остаточну перевірку на незалежній тестовій вибірці. Після чого за допомогою метрик для оцінювання моделей, визначається якісні показники роботи цих моделей.

3.3.3. Метрики оцінювання моделей машинного навчання

Для порівняння показників алгоритмів машинного навчання використовуються стандартні метрики класифікації, що дозволяють вимірювати точність передбачень на тестових вибірках. Найчастіше у фармацевтичній аналітиці важливо не лише визначити загальну точність моделі, але й зрозуміти, наскільки коректно вона класифікує позитивні та

негативні випадки, оскільки помилкове визначення ефективності або побочних ефектів препарату може призвести до значних ризиків.

Основними метриками, що використовуються у даній роботі, є:

Accuracy, ця метрика відображає загальну точність класифікації, вона є однією з найбільш базових метрик оцінювання моделей машинного навчання. Вона показує, яку частку всіх передбачень модель класифікувала правильно, тобто скільки разів прогноз збігся з реальною категорією. У формулі вона розраховується як відношення суми правильних позитивних та негативних передбачень до загальної кількості прикладів:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \quad (3.1)$$

де TP (True Positive) - це кількість випадків, коли модель правильно визначила позитивний результат, TN (True Negative) - коли модель правильно виявила негативний результат, FP (False Positive) - помилкові позитивні передбачення, а FN (False Negative) - пропущені реальні позитивні випадки.

Accuracy надає швидке уявлення про точність моделі, однак її застосування має обмеження у ситуаціях із незбалансованими даними, коли одна категорія переважає над іншою. У таких випадках навіть модель, що завжди прогнозує найпоширенішу категорію, може демонструвати високий показник, хоча насправді не вміє виявляти менш чисельні, але критично важливі випадки.

У фармацевтичній аналітиці це означає, що показник Accuracy може не відображати реальної здатності моделі ідентифікувати ризикові або небезпечні препарати, тому його зазвичай використовують лише як базову орієнтовну метрику.

Precision характеризує точність передбачень моделі для позитивного класу, тобто визначає, яка частка передбачених позитивних випадків справді є такими. Формула має вигляд:

$$\text{Precision} = \frac{TP}{TP+FP} \ , \qquad (3.2)$$

де TP - це правильні позитивні передбачення, а FP - помилкові позитивні спрацьовування.

Високий показник Precision означає, що модель рідко помиляється, класифікуючи негативний випадок як позитивний, що особливо важливо у фармацевтичному контексті, де помилкове виявлення ризикового препарату може призвести до зайвого відхилення показників ліків або невиправданих витрат на додаткові дослідження.

Низька точність позитивного передбачення свідчить про те, що модель часто дає помилкові сигнали, і її прогнози слід додатково перевіряти. Precision добре відображає надійність позитивних прогнозів і є критичною метрикою у випадках, коли важливо мінімізувати кількість хибних позитивних результатів.

Recall або чутливість вимірює здатність моделі виявляти всі реальні позитивні випадки. Його обчислюють за формулою:

$$\text{Recall} = \frac{TP}{TP+FN} \ , \qquad (3.3)$$

де TP - правильно передбачені позитивні випадки, а FN - пропущені реальні позитивні приклади.

У фармацевтичній аналітиці висока повнота означає, що модель здатна виявляти більшість ризикових або небезпечних препаратів, що критично для забезпечення безпеки пацієнтів та запобігання серйозним побічним ефектам.

Низький Recall свідчить про те, що модель часто пропускає важливі випадки, що може призвести до невиявлення потенційно небезпечних ліків. Ця метрика є ключовою у завданнях, де критично важливо мінімізувати пропуск реальних позитивних прикладів, навіть якщо це означає деяке збільшення кількості помилкових спрацьовувань.

F1-Score є гармонічним середнім між Precision та Recall і дозволяє оцінити роботу моделі з урахуванням одночасного контролю над помилковими позитивними та пропущеними позитивними випадками. Формула розрахунку виглядає так:

$$\text{F1 Score} = 2 * \frac{\text{Precision} * Recall}{\text{Precision} + Recall} \quad (3.4)$$

Високий F1-Score свідчить про те, що модель добре збалансована, вона не лише рідко помиляється у визначенні позитивних прикладів, але й виявляє більшість реальних позитивних випадків.

У фармацевтичній практиці ця метрика особливо цінна, оскільки дозволяє оцінити модель комплексно, з точки зору безпеки та ефективності препаратів. F1-Score часто використовується як основна метрика у ситуаціях із незбалансованими даними, коли необхідно враховувати і точність, і повноту передбачень одночасно, оскільки кожне хибне передбачення або пропущений випадок може мати серйозні наслідки для пацієнтів та фармацевтичних досліджень.

3.4. Модуль прогнозування властивостей лікарських засобів

Прогнозування властивостей лікарських засобів є центральним функціональним компонентом роботи, оскільки саме вона забезпечує практичне застосування результатів машинного навчання та глибинних нейронних мереж для визначення властивостей препаратів, можливих побічних реакцій та потенційної корисності певного лікарського засобу в клінічному застосуванні. Основною задачею цієї підсистеми є використання навченої моделі для формування об’єктивних прогнозів на підставі нових вхідних даних.

Для початку роботи дані передаються у модель машинного навчання або нейронну мережу. Модель повертає результат прогнозу у числовій формі,

що надалі інтерпретується як оцінка ефективності, а також ризику побічних ефектів або інших характеристик.

Кожен прогнозований результат зберігається у таблиці Prediction, що дозволяє проводити історичний аналіз та порівняння результатів між різними моделями. Додатково, система передбачає можливість використання декількох моделей паралельно, що дозволяє порівняти їхню якість і обрати найбільш оптимальні методи прогнозування залежно від конкретного типу препарату, структури його діючих речовин або характеру побічних ефектів.

Завдяки цьому система стає здатною до аналізу фармацевтичних даних, а процес попередньої оцінки властивостей лікарських засобів переходить від суб'єктивного експертного підходу до формального математичного моделювання на підставі накопичених історичних даних і статистично підтверджених моделей. Це дозволяє підвищити рівень об'єктивності оцінки фармацевтичної продукції, прискорити процес прийняття рішень та підсилити доказовість вибору оптимальних препаратів у практичних медичних задачах.

Прогнозована ефективність препарату є комплексною характеристикою, що визначає здатність лікарського засобу впливати на симптоми хвороби при без виникнення серйозних побочних ефектів. Для її оцінки важливо враховувати кілька ключових аспектів, оскільки жоден з них окремо не дає змоги оцінити цю величину.

Діюча речовина препарату визначає, чи здатен препарат впливати на захворювання. Оцінка впливу діючої речовини базується на клінічних даних, результатах досліджень та наукових публікаціях.

Дозування препарату також є дуже важливим показником, адже дуже діюча речовина може бути недостатньо впливовою, якщо доза не відповідає оптимальному діапазону. Таким чином, цей показник є важливим для передбачення реальної ефективності лікарського засобу в конкретного пацієнта.

Також форма препарату, яка впливає на швидкість дії активної речовини. Наприклад, капсули або ін’єкційні форми можуть діяти швидше та забезпечувати більш стабільну концентрацію у крові, ніж таблетки.

Важливо враховувати і цільове захворювання, не всі препарати однаково дієві для всіх захворювань. Цей показник важливий для точного прогнозу ефективності, оскільки навіть сильна діюча речовина може бути неефективною, якщо вона не орієнтована на конкретний патологічний механізм.

Швидкість дії препарату є також важливим показником, який потрібно прахувати. Для багатьох захворювань важлива не лише кінцева дія препарату, але і наскільки швидко настає ефект від нього. Швидкодіючі препарати дозволяють контролювати симптоми та покращувати стан пацієнта, що особливо важливо в гострих або прогресуючих захворюваннях.

Окремо слід враховувати побічні ефекти, які можуть суттєво знижувати загальну користь препарату. Таким чином, препарат із високою терапевтичною активністю, але значними побічними ефектами, отримає менший загальний показник ефективності.

Об’єднання всіх цих параметрів у єдину метрику дозволяє отримати збалансовану оцінку Result, яка відображає не лише фармакологічну активність препарату, а і його безпеку, форму, специфічність для захворювання та швидкість дії. Такий комплексний підхід забезпечує більш точне прогнозування ефекту і дозволяє приймати обґрунтовані рішення щодо вибору лікарських засобів.

$$\boldsymbol{Result} = (w_1 * AI + w_2 * D + w_3 * F + w_4 * TD + w_5 * AS) * (1 - SE) \quad , (3.5)$$

де

- AI – діюча речовина;
- D - дозування препарату;
- F - лікарська форма;
- TD – цільова хвороба (симптоми);
- AS - швидкість настання ефекту;

- SE - можливі побічні ефекти, що знижує загальну ефективність;
- w_1-w_5 - вагові коефіцієнти, що визначають значимість кожного параметра.

Розглянемо також приклад прогнозування можливих побічних ефектів лікарського засобу та оцінки його ефективності. До системи надходить новий фармацевтичний продукт з інформацією про діючу речовину, фармакологічний тип та опис механізму дії. Після збереження таких даних у сховищі система обирає навчений алгоритм машинного навчання, який було завчасно навчено на історичних фармацевтичних даних, де кожен препарат має відомі результати клінічних досліджень та реальних медичних спостережень.

На основі структурованих атрибутів препарату модель виконує побудову векторного опису препарату - це представлення характеристик у числовому форматі, з яким можуть працювати моделі машинного навчання та нейронні мережі. Далі цей вектор подається на вхід моделі, яка на основі знань, отриманих під час тренування, робить оцінку ймовірності виникнення певних побічних ефектів (наприклад головний біль, нудота, алергічні реакції) та дає прогноз ефективності щодо очікуваної дії препарату без значних побочних ефектів.

Наприклад, модель може повернути результат у такому вигляді: ефективність 82%, ризик головного болю 21%, ризик нудоти 12%, ризик підвищеної температури 5%. Ці значення зберігаються, після чого можуть бути використані аналітиком для формування висновку. Аналітик, переглядаючи прогноз, може визначити, що даний препарат має високий потенціал ефективності та низьку ймовірність серйозних побічних реакцій, що робить його доцільним для подальших досліджень.

РОЗДІЛ 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1. Вимоги до апаратного та програмного забезпечення

Реалізація розробленої системи для прогнозування властивостей лікарських засобів, виявлення можливих побочних ефектів та формування аналітичних звітів потребує визначення чітких апаратних і програмних вимог. Оскільки система включає роботу із значними обсягами даних, побудову моделей, роботу сховища даних та OLAP-аналіз, важливо забезпечити достатній рівень обчислювальних ресурсів і відповідне програмне забезпечення для стабільної і коректної роботи всіх підсистем.

Апаратні вимоги

Для роботи всієї системи використовується персональний комп’ютер, який виконує функції робочої станції для розробки, моделювання та тестування. Основні вимоги формуються на основі обсягів даних, кількості моделей машинного навчання, навантаження на процесор та потреб у пам’яті під час обробки даних.

Мінімальна рекомендована конфігурація апаратного забезпечення

- Процесор потрібен щонеменше 4-ядерний CPU (не нижче Intel Core i5 або AMD Ryzen 5). Багатоядерність необхідна для паралельної обробки даних, навчання моделей та для виконання прогнозування властивостей та можливих побочних ефектів фармацевтичних препаратів.
- Для користування системою кількість оперативної пам’яті повинно бути від 8 ГБ. Цей обсяг є достатнім для роботи SQL Server, модулів SSIS та SSAS, а також для виконання навчання моделей у Python.
- Для комфортної роботи бажано мати накопичувач SSD від 256 ГБ. SSD значно пришвидшує роботу SQL-запитів, завантаження великих наборів даних та доступ до проектів.
- Графічний адаптер не є критично важливим, але базова підтримка GPU прискорює роботу деяких моделей нейронних мереж.

- Операційна система Windows 10 або 11 (64-bit), необхідна для коректної роботи Microsoft SQL Server, Visual Studio та загалом всіх компонентів системи.

Програмні вимоги

Для реалізації функціоналу збереження, обробки даних, навчання моделей, прогнозування властивостей фармацевтичних продуктів та їх можливих побочних ефектів, були застосовані такі компоненти:

- **Microsoft SQL Server** (версія 2017 або новіша). Використовується для створення бази даних операційного рівня, таблиць продуктів, моделей, прогнозів та аналітичних результатів.
- **SQL Server Integration Services (SSIS).** Застосовується для реалізації завантаження даних із системи у сховище даних, трансформації, очищення та підготовки до аналітичної обробки.
- **SQL Server Analysis Services (SSAS).** Використовується для проєктування багатовимірної моделі, створення вимірів, фактів та ключових показників ефективності (KPI). SSAS забезпечує побудову OLAP-куба, що дозволяє здійснювати гнучкий і швидкий аналітичний перегляд даних.
- **Visual Studio (версія 2019 або новіша).** Застосовується для розробки проєктів SSIS та SSAS, а також для налаштування KPI, вимірів і процесу відтворення даних у OLAP-кубі.

Python версії 3.10 або вище з основними бібліотеками

- NumPy - для числових обчислень;
- Pandas - для обробки табличних даних;
- scikit-learn - для побудови моделей машинного навчання;
- Matplotlib - для побудови графіків та візуалізацій результатів аналізу;
- SciPy - для обчислення відстаней між кластерами та статистичної обробки.

Середовище розробки PyCharm або Visual Studio Code, що дозволяє виконувати покроковий аналіз і створення візуалізацій.

4.2. Хід виконання дослідження

У ході виконання дослідження спочатку було сформовано базу даних, яка включає інформацію про лікарські засоби, їх активні речовини, типи, побічні ефекти, що лікує продукт, форму препаратів та інші характеристики. Для подальшої роботи з цими даними, їх було структуровано до єдиного вигляду. Після формування набору даних створено підсистему зберігання, у якій інформація розподіляється між таблицями згідно з логічною структурою. Це забезпечило зручний доступ до даних для аналітичних операцій та навчання моделей.

Наступним етапом стала підготовка даних для моделювання, яка включала нормалізацію числових ознак, кодування текстових полів (наприклад, типів препаратів або побічних ефектів) та формування вибірок для навчання й тестування. На основі цих даних було проведено навчання кількох моделей машинного навчання, а саме логістична регресія, дерева рішень, випадковий ліс та нейронна мережа. Метою було оцінити, яка з моделей найкраще прогнозує ефективність лікарських засобів і ймовірність виникнення побічних ефектів.

Після тренування моделей проведено їх оцінювання за допомогою метрик точності, таких як Accuracy, Precision, Recall та F1 Score. На цьому етапі було виявлено, що моделі з глибшими та складнішими архітектурами (наприклад, випадковий ліс та нейронна мережа) демонструють кращі результати на великих обсягах даних, тоді як простіші алгоритми, такі як дерево рішень та логістична регресія, показують гарні показники лише в простих задачах.

Далі було реалізовано механізм прогнозування, який дає змогу користувачеві вибирати фармацевтичний продукт і отримувати передбачення ефективності та ризику побічних ефектів на основі навченої моделі.

Результати зберігаються у таблиці Prediction та можуть бути переглянуті аналітиком або фармацевтом через відповідний інтерфейс.

Фінальним етапом стала аналітична обробка результатів - порівняння прогнозів із наявними даними, аналіз отриманих висновків та формування підсумкових звітів у підсистемі аналітики. Ці звіти дозволяють експертам інтерпретувати роботу моделей, а також формувати висновки про потенційну ефективність лікарських засобів і ризики побічних реакцій.

У результаті виконаного дослідження сформовано цілісну архітектуру інтелектуальної системи для аналізу фармацевтичних даних. Реалізовані алгоритми машинного навчання забезпечили можливість прогнозування ефективності лікарських засобів і ймовірності виникнення побічних ефектів. Отримані результати підтверджують працездатність побудованої системи, її здатність до узагальнення нових даних та потенційну користь для подальшого застосування у фармацевтичній аналітиці й підтримці прийняття рішень.

4.3. Аналіз результатів роботи

Після розроблення системи аналітики фармацевтичних даних було проведено тестування та порівняльний аналіз показників різних алгоритмів машинного навчання, таких як логістичної регресії, дерева рішень, випадкового лісу та глибокої нейронної мережі. Основною метою було оцінити здатність моделей прогнозувати ефективність лікарських засобів та виявленню побочних ефектів на основі структурованих даних.

Для аналізу використовувалася вибірка з приблизно 500 препаратів, серед яких містилася інформація про форму препарату, активні речовини, фармакологічну групу, дозування, рівень можливих побічних реакцій та інші дані. Моделі навчалися на 80% даних, а їх тестування проводилося на решті 20%.

4.3.1. Порівняльна характеристика моделей

Під час навчання моделі вивчали закономірності на одному й тому самому наборі даних, що дозволило порівняти їх показники, коли вони були в одинакових початкових умовах. Результати тестування наведено у таблиці 4.1.

Таблиця 4.1

Таблиця порівняння показників моделей штучного інтелекту

Модель	Accuracy	Precision	Recall	F1-score
Логістична регресія	0.74	0.70	0.68	0.69
Дерева рішень	0.77	0.75	0.74	0.74
Випадковий ліс	0.8	0.78	0.76	0.77
Глибока нейронна мережа	0.84	0.81	0.80	0.80

Аналізуючи результати, можна помітити, що глибока нейронна мережа показала найвищу точність серед усіх моделей 84%. Такий результат пояснюється її здатністю виявляти нелінійні взаємозв'язки між характеристиками лікарських засобів, зокрема між активними речовинами, формою препарату та вираженістю побічних ефектів. Завдяки багатошаровій структурі нейронна мережа краще узагальнює інформацію навіть у випадках, коли дані містять приховані залежності або неоднорідність.

Випадковий ліс виявився найкращим серед **базових** моделей машинного навчання з точністю 80%. Цей алгоритм поєднує кілька дерев рішень, що дозволяє зменшити ризик перенавчання і покращити стабільність прогнозів. Він добре працює з табличними фармацевтичними даними, оскільки здатний автоматично визначати важливість ознак, зокрема тих, які впливають на ефективність препарату та побочні ефекти.

4.3.2. Аналіз стабільності моделей

Окрім точності прогнозування, важливим аспектом оцінювання алгоритмів машинного навчання є стабільність їхньої роботи. Під

стабільністю у даному контексті розуміється здатність моделі демонструвати подібні результати при повторному навчанні на різних вибірках даних, що мають однаковий розподіл, або при незначних змінах у структурі даних. Цей показник є критично важливим для фармацевтичних систем, де результати аналізу можуть впливати на прийняття рішень щодо безпеки лікарських засобів.

Для оцінки стабільності кожна модель була повторно декілька разів на різних випадкових підвибірках із початкового набору даних. Після цього обчислювалося середнє відхилення для показників Accuracy та F1-score. Результати наведено у таблиці 4.2.

Таблиця 4.2

Показники стабільності роботи моделей машинного навчання

Модель	**Середня Accuracy**	**Середнє відхилення**	**Середній F1-score**	**Середнє відхилення**
Логістична регресія	0.72	0.018	0.67	0.021
Дерево рішень	0.76	0.042	0.73	0.047
Випадковий ліс	**0.80**	**0.015**	**0.77**	**0.017**
Глибока нейронна мережа	**0.82**	**0.028**	**0.78**	**0.031**

Як видно з результатів в таблиці, найвищу стабільність продемонстрував алгоритм «Випадковий ліс», для якого стандартне відхилення показників точності становило 0.015, а для F1-score 0.017. Це свідчить про те, що модель практично не змінює свої результати при повторному навчанні, що зумовлено ансамблевою природою методу. Завдяки використанню множини незалежних дерев рішень із різними наборами ознак і зразків, випадковий ліс компенсує коливання, пов’язані з вибором конкретних

даних. Саме тому його прогнозування залишається найбільш надійним серед розглянутих моделей.

Глибока нейронна мережа показала трохи вищі результати точності, але водночас має меншу стабільність (Середнє відхилення 0.028 для Accuracy). Це пов’язано з тим, що її параметри (ваги) залежать від початкової ініціалізації та швидкості навчання. Навіть невеликі зміни в навчальному наборі або в послідовності подачі даних можуть призвести до відмінностей у кінцевих результатах. Тим не менш, коливання залишаються в межах допустимих значень, що свідчить про загальну узгодженість нейронної моделі.

Для додаткового дослідження стабільності моделей машинного навчання було проведено аналіз зміни точності (Accuracy) протягом десяти ітерацій навчання для кожної моделі. Результати відображено на рис. 4.3.2.1.

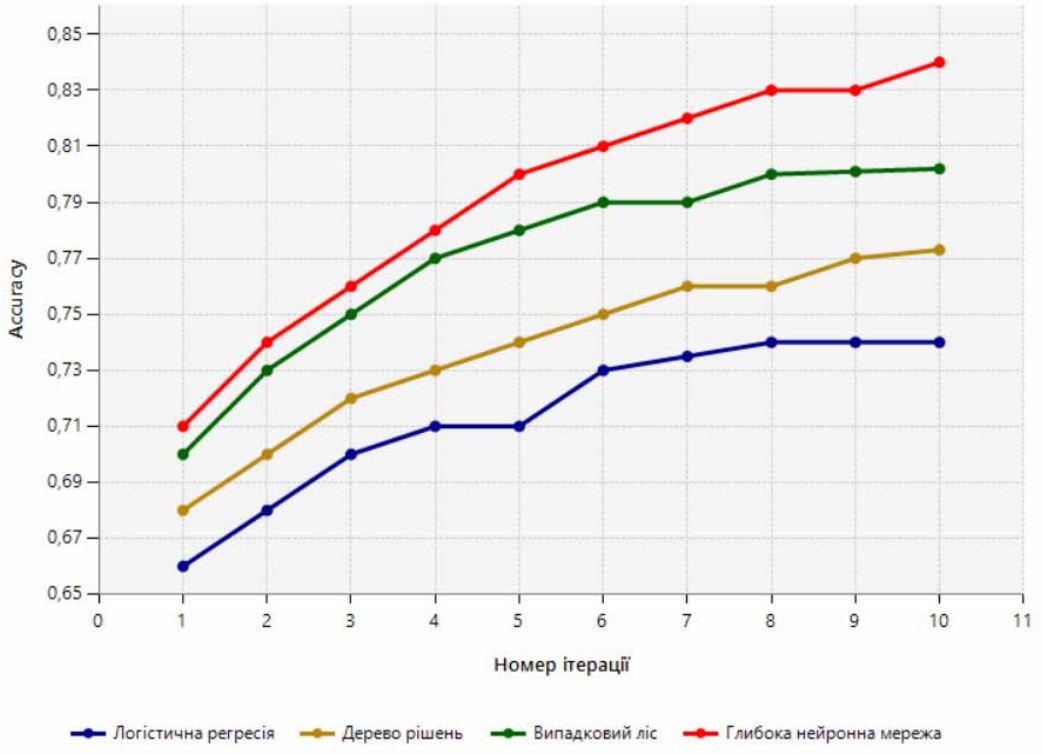

Рис. 4.3.2.1. Графік зміни точності моделей

На початкових етапах тренування усі моделі демонстрували поступове зростання точності, що свідчить про активне засвоєння закономірностей у навчальних даних. Логістична регресія, яка є найпростішою моделлю серед розглянутих, показала стабільне, але помірне покращення точності від 66% до

74%. Це свідчить про її надійність, проте модель досягає плато вже після сьомої ітерації, що означає обмежену здатність до подальшого вдосконалення при збільшенні обсягу тренувальних даних.

Найкращі результати стабільності спостерігаються у глибокої нейронної мережі. Точність моделі зросла з 71% до 84%, причому темпи покращення залишалися сталими протягом усього процесу навчання. Це свідчить про високу адаптивність до складних фармацевтичних даних та навчання навіть на пізніх етапах. Водночас плавне зростання без різких стрибків вказує на збалансовану архітектуру мережі та правильне її налаштування.

Таким чином, проведений аналіз показав, що всі моделі характеризуються задовільною стабільністю, проте найкраще співвідношення між швидкістю навчання, точністю та стійкістю до перенавчання продемонстрували випадковий ліс і глибока нейронна мережа. Перша модель вирізняється збалансованістю, а друга найвищим рівнем показників при тривалому навчанні.

4.3.3. Аналіз прогнозування властивостей фармецевтинних продуктів

На заключному етапі дослідження було проведено оцінку підсистеми прогнозування, яка визначає можливі побічні ефекти лікарських засобів на основі історичних даних, складу препарату, типу діючих речовин та інших даних. Метою цього аналізу було перевірити, наскільки коректно система виявляє нові або потенційно пропущені побічні ефекти.

Для проведення експерименту було сформовано вибірку з п’яти основних груп лікарських засобів: жарознижувальні, противірусні, знеболюючі, антибактеріальні та антиалергічні препарати. Дані для кожної групи вже були в базі даних, де для кожного фіксувалась кількість відомих побічних ефектів до застосування алгоритмів прогнозування та кількість додатково виявлених побічних реакцій після обробки системою див. Рис. 4.3.3.1.

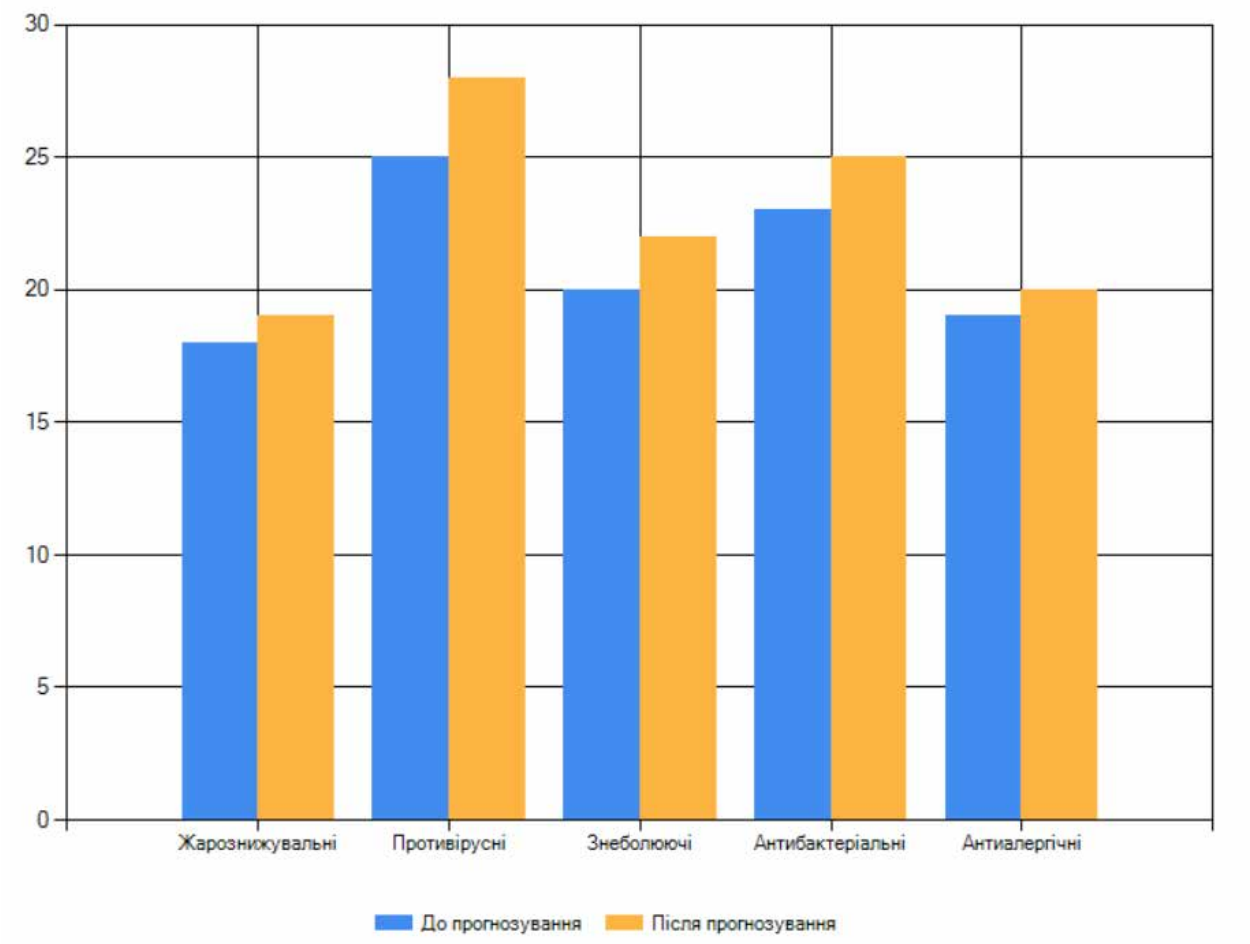

Рис. 4.3.3.1 Результати наявних та прогнозованих побічних ефектів

Результати дослідження показали, що для всіх груп препаратів спостерігається збільшення кількості виявлених побічних ефектів у середньому на 5-10%. Це свідчить про здатність системи машинного навчання виявляти закономірності, які не були очевидними при традиційних підходах до фармакологічного аналізу. Зокрема, найбільше зростання кількості побічних ефектів спостерігалося у противірусних засобів (приблизно 10%), що може бути пов’язано з більш складною фармакодинамікою цих препаратів. Найменше підвищення спостерігалося у жарознижувальних та антиалергічних препаратів (5%), що є очікуваним, адже вони мають добре досліджений профіль безпеки.

У результаті проведеного аналізу можна зробити висновок, що розроблена система машинного навчання демонструє високу працездатність у задачі прогнозування можливих побічних ефектів лікарських засобів. Навіть незначне, але стабільне збільшення кількості виявлених побічних реакцій

свідчить про те, що алгоритми здатні виявляти приховані закономірності у фармакологічних даних, які не були враховані під час первинних клінічних досліджень. Такий підхід може суттєво підвищити якість фармаконагляду, сприяти ранньому виявленню потенційно небезпечних взаємодій між компонентами препаратів та знизити ризики для пацієнтів.

4.3.4. Аналіз результатів кластеризації та KPI

Для оцінювання ефективності системи було обрано KPI, що відображають точність (Accuracy) різних моделей машинного навчання, які застосовувалися для прогнозування ефективності лікарських засобів і виявлення побічних ефектів. Для цільового значення було обране вище за середнє значення точності наявних моделей, а саме 82%, яке вважається прийнятним рівнем точності для задач із реальними фармацевтичними даними, що містять шум, неоднорідність і неповноту інформації. Таке значення дає змогу об’єктивно оцінити якість моделей, не вимагаючи нереалістично високої точності понад 90%, яка зазвичай недосяжна без перенавчання на великих обсягах точних даних див. Рис. 4.3.4.1.

Отобразить структуру	Значение	Цель	Состояние
Точність моделі випадковий ліс	0,8	0,82	
Точність моделі дерево рішень	0,77	0,82	
Точність моделі логістичної регресії	0,74	0,82	
Точність моделі нейронна мережа	0,84	0,82	

Рис. 4.3.4.1 KPI точності моделей

Логістична регресія продемонструвала стабільні результати при помірному рівні точності. Вона добре працює з лінійними залежностями між характеристиками лікарських засобів, проте її обмеження виявляються при аналізі складних багатовимірних взаємозв’язків. Незважаючи на відставання від цільового показника, модель залишається корисною як базова, оскільки забезпечує швидке навчання та точні результати на простих даних.

Модель дерева рішень забезпечила покращення точності завдяки здатності працювати з нелінійними зв’язками між ознаками. Вона краще враховує комбінації діючих речовин і типи препаратів, що впливають на

прогноз ефективності. Незважаючи на те, що модель ще не досягла заданої цілі, її результати свідчать про високий потенціал для використання у попередньому аналізі даних перед застосуванням складніших алгоритмів.

Алгоритм випадковий ліс показав найвищу точність серед класичних методів машинного навчання. Завдяки ансамблевому підходу, який поєднує велику кількість незалежних дерев рішень, він зменшує ризик перенавчання і забезпечує збалансовану роботу на даних із різною структурою. Досягнутий рівень точності майже дорівнює цільовому, що свідчить про успішне узагальнення закономірностей у фармацевтичних даних.

Глибока нейронна мережа продемонструвала найкращі результати серед усіх моделей. Її здатність до самоорганізації та виявлення складних нелінійних зв'язків дала змогу досягнути точності, вищої за заплановане цільове значення. Перевищення порогового рівня на 2% свідчить про те, що цей підхід найкраще адаптується до складних фармацевтичних закономірностей, у тому числі взаємозалежностей між активними речовинами, дозуванням препарату, побічними ефектами та іншими даними.

Також були створені ключові показники ефективності для оцінки середнього рівня ефективності різних типів лікарських засобів на основі даних, отриманих після аналітичного оброблення в системі машинного навчання. В якості цільового значення середньої ефективності було прийнято 75%, що також є трохи вищим за середнє значення серед всіх препаратів див. Рис. 4.3.4.2.

Отобразить структуру	Значение	Цель	Состояние	Тренд
Середня ефективність препаратів антиалергічного типу	0,76	0,75		↗
Середня ефективність препаратів антибактеріального типу	0,74	0,74		→
Середня ефективність препаратів жарознижувального типу	0,79	0,75		↑
Середня ефективність препаратів знеболюючого типу	0,72	0,74		↘
Середня ефективність препаратів противірусного типу	0,68	0,71		↓

Рис. 4.3.4.2 KPI рівня прогнозованої ефективності для різних груп препаратів

Данний KPI показує, як змінились показники ефективності препаратів, порівнюючи із значеннями за попередній рік, що дозволяє аналізувати як

змінюється якість фармацевтичних препаратів, їх властивості та можливі побочні ефекти.

Знеболюючі препарати показали ефективність, трохи гіршу чим минулого року. Незначне відставання може бути пов’язане з різницею у складі активних речовин.

Жарознижувальні засоби продемонстрували гарний приріст ефективності. Це пояснюється тим, що такі препарати мають добре вивчені механізми дії, стандартизовані дозування та однорідні характеристики.

Противірусні препарати показали спад показнику ефективності. Це може бути пов’язано з високою мінливістю вірусів, складністю прогнозування їх дії та обмеженістю достовірних даних.

Антибактеріальні засоби мають такий самий рівень, як і попереднього року. Це свідчить про стабільність препаратів цієї групи.

Антиалергічні препарати покращили показник ефективності, що свідчить про високу стабільність цієї групи препаратів. Це можна пояснити великою кількістю доступних даних, типовими механізмами дії та добре структурованими фармакологічними характеристиками, які легко піддаються аналізу моделлю машинного навчання.

Переходячи до кластеризації, у контексті фармацевтичного аналізу вона дозволяє виявити групи лікарських засобів із подібними властивостями, а саме рівнем ефективності або ризиком виникнення побочних ефектів. Це особливо корисно для виявлення потенційно небезпечних груп препаратів, які потребують додаткового аналізу.

Для визначення оптимальної кількості кластерів було застосовано метод «ліктя». На графіку спостерігається, що доцільно розподілити дані саме на 3 кластери див. Рис. 4.3.4.3-4.3.4.4.

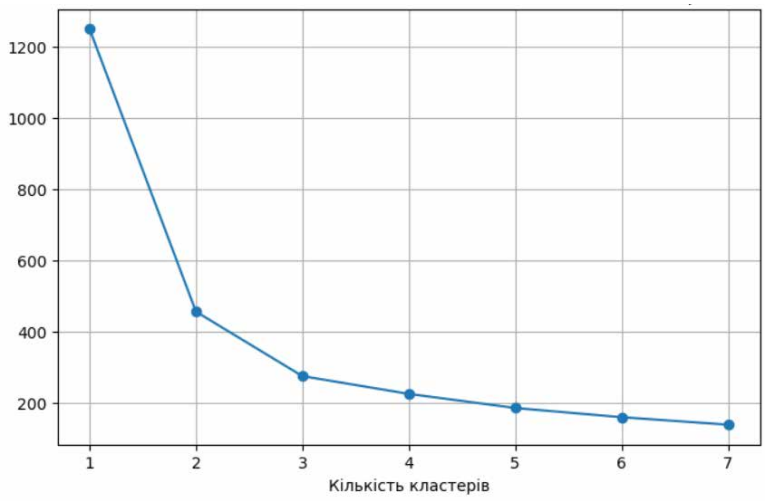

Рис. 4.3.4.3 Результат методу ліктя

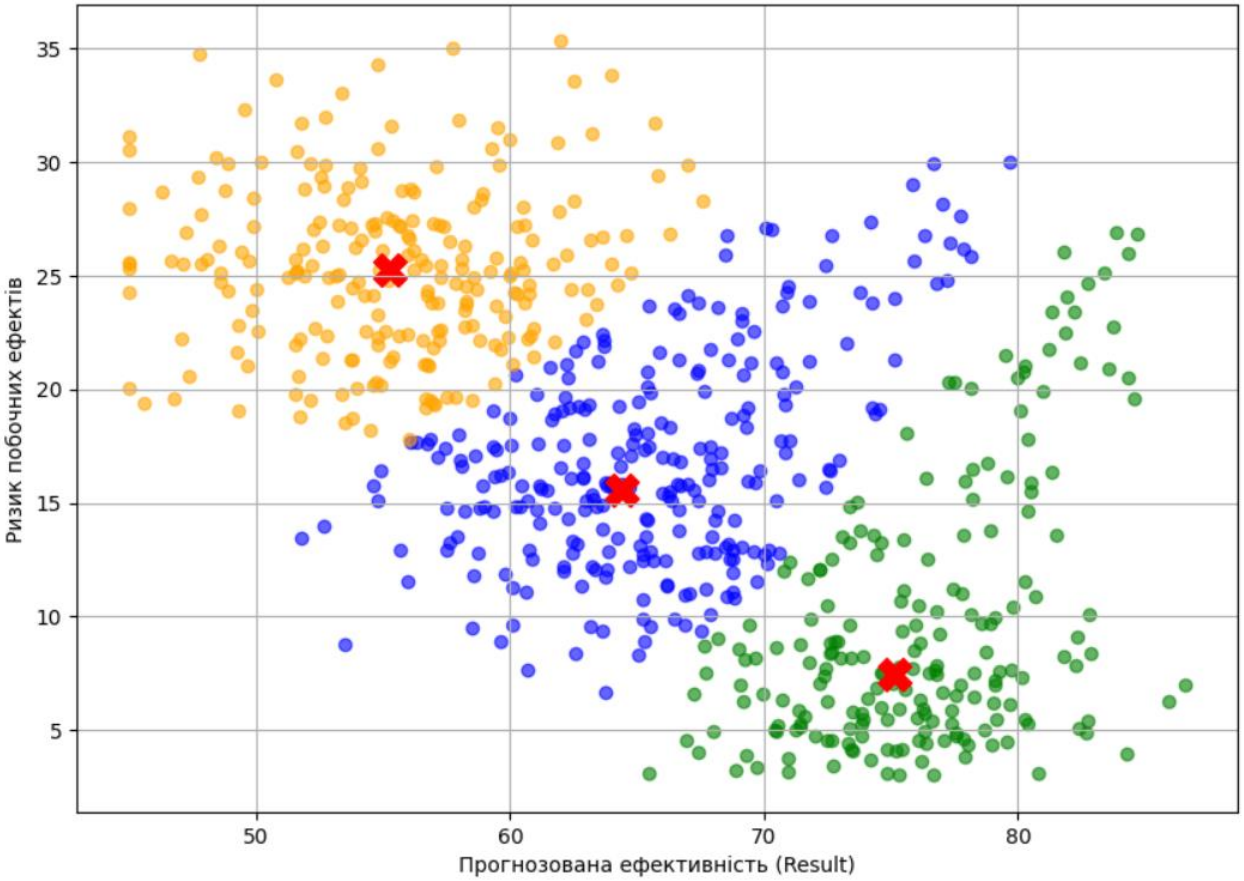

Рис. 4.3.4.4 Кластерний аналіз

Як видно на рисунку, такий підхід дозволяє формувати групи препаратів, виявляти потенційно небезпечні або засоби з низькою ефективністю. Як показано на рисунку, фармацевтичні препарати поділились на 3 групи з схожими показниками.

Зелений кластер об’єднує препарати з високою ефективністю 70-85% та низьким рівнем побічних ефектів близько 7%. Це, як правило, добре перевірені засоби з тривалим досвідом використання.

Синій кластер включає препарати із середньою ефективністю 60-70% і помірним ризиком побічних ефектів 10-20%. До цієї групи можуть входити препарати нового покоління або ті, що мають складніші механізми дії.

Помаранчевий кластер характеризується нижчою ефективністю 50-60% і підвищеним ризиком побічних ефектів 20-30%. Це потенційно ризикована група препаратів, які можуть вимагати додаткового аналізу або вдосконалення формули.

ВИСНОВКИ

У магістерській роботі було досліджено процеси збору, обробки та аналітичного використання фармацевтичних даних із застосуванням сучасних технологій машинного навчання та моделей штучного інтелекту. У результаті виконання поставлених завдань було розроблено систему, що дозволяє виконувати прогнозування ефективності лікарських засобів та виявляти потенційні побічні ефекти на основі зібраних даних.

Під час роботи було створено сховище даних, що включає таблиці вимірів та фактів, необхідні для глибинного аналізу лікарських засобів та їхніх характеристик, а також аналізу показників моделей машинного навчання.

Особлива увага була приділена побудові та навчанням моделей машинного навчання**,** включаючи логістичну регресію, дерево рішень, випадковий ліс та штучну нейронну мережу. Кожна модель була навчена на попередньо оброблених фармацевтичних даних, що дозволило виконати порівняльний аналіз їх точності. Результати продемонстрували, що найвищий рівень точності забезпечила модель нейронної мережі 84%. Решта моделей показали точність на рівні 74%-80%, що вказує на можливіть їхнього подальшого покращення для збільшення показників, що в свою чергу покращить точність прогнозування.

У ході дослідження було розроблено підсистему прогнозування можливих побічних ефектів лікарських засобів, ключовою функцією якої стало виявлення можливих побочних ефектів у фармацевтичних продуктів на основі комплексного аналізу історичних даних, складу препаратів, діючих речовин та інших даних. У результаті прогнозування система змогла збільшити кількість виявлених побічних ефектів у середньому на 5-10%, що підтверджує її здатність знаходити приховані закономірності, які не фіксуються традиційними методами аналізу. Найбільше зростання спостерігалося серед противірусних препаратів, що є логічним з огляду на їх

складну структуру, а також з високою мінливістю та складністю вірусних захворювань.

Важливим результатом стало впровадження системи KPI у середовищі Microsoft SSAS**,** яка дозволила сформувати індикатори для контролю якості моделей машинного навчання, а також порівняння прогнозованих оцінок лікарських засобів. Зокрема, були визначені KPI точності для кожної моделі машинного навчання та KPI ефективності різних груп препаратів з можливістю порівняння з минулими роками. Такий підхід забезпечує відображення досягнутих результатів та дозволяє здійснювати порівняння роботи системи в проміжку певного часу.

У підсумку можна стверджувати, що всі поставлені в роботі задачі були повністю виконані. Створена система є комплексною технологічною платформою, яка може бути використана як для наукових досліджень, так і для практичних потреб фармацевтичних компаній або лабораторій. Вона забезпечує підвищення точності аналізу властивостей лікарських засобів, прогнозування можливих ризиків побочних ефектів та формує основу для подальшого розширення функціональних можливостей.

Таким чином, виконана магістерська робота має як теоретичну, так і практичну цінність. Вона підтверджує важливість застосування машинного навчання та технологій роботи з даними у сучасній фармацевтичній галузі та демонструє перспективність розвитку інтелектуальних систем для підвищення якості аналізу лікарських засобів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко О.М., Бублій Т.Д., Перспективи використання штучного інтелекту в медичній сфері, 2024р. [Електронний ресурс]. URL: https://doi.org/10.31718/2077-1096.24.3.137 (дата звернення: 18.09.2025).

2. А.А. Висоцький, О.О. Суріков, С.В. Василюк-Зайцева, Розвиток штучного інтелекту в сучасній медицині, 2023р. [Електронний ресурс]. URL: https://doi.org/10.32471/umj.1680-3051.154.241221 (дата звернення: 11.08.2025).

3. T. O. Nazirova, O. B. Kostenko, Нейрономережева інформаційна технологія опрацювання медичних даних, 2018р. [Електронний ресурс]. URL: https://doi.org/10.15421/40280828 (дата звернення: 12.09.2025).

4. Попова І.А., Демченко Н.В., Переваги та особливості застосування штучного інтелекту в медичній та фармацевтичній системах, 2024р. [Електронний ресурс]. URL: https://dspace.nuph.edu.ua/bitstream/123456789/32837/1/114-121.pdf (дата звернення: 22.09.2025).

5. Шуляк Н.С., Ружицька О.Б, Побічна дія ліків, 116с, 2024р. [Електронний ресурс]. URL: https://libvmi.volyn.ua/sites/default/files/2025-02/%D0%9F%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA%20%D0%B4%D0%BB%D1%8F%20%D1%81%D0%B0%D0%BC.%D1%80%D0%BE%D0%B1.%20%D0%9F%D0%BE%D0%B1%D1%96%D1%87%D0%BD%D0%B0%20%D0%B4%D1%96%D1%8F%20%D0%BB%D1%96%D0%BA%D1%96%D0%B2.pdf (дата звернення: 03.07.2025).

6. Лисенко М. М., Пронькін О. В. Застосування технологій машинного навчання для встановлення медичного діагнозу, 2024р. [Електронний ресурс]. URL: https://doi.org/10.31673/2412-9070.2024.051397 (дата звернення: 03.09.2025).

7. SQL Server Analysis Services (SSAS) Multidimensional, 2025р. [Електронний ресурс]. URL: https://docs.dataedo.com/docs/documenting-

technology/supported-databases/analysis-services-multidimensional/ (дата звернення: 09.09.2025).

8. Лизогуб В.Г., Богдан Т.В., Шараєва М.Л., Крайдашенко О.В., Волошина О.О., Побічні дії лікарських засобів, 137с, 2013р. [Електронний ресурс]. URL: https://nmuofficial.com/files/kaf87/posibnsk.pdf (дата звернення: 12.08.2025).

9. SQL Server Tutorial, 2025р. [Електронний ресурс]. URL: https://www.sqlservertutorial.net/ (дата звернення: 11.06.2025).

10. Visual Studio Code documentation, 2025р. [Електронний ресурс]. URL: https://code.visualstudio.com/docs (дата звернення: 14.07.2025).

11. Python 3.14.0 documentation, 2025р. [Електронний ресурс]. URL: https://scikit-learn.org/stable/ (дата звернення: 21.07.2025).

12. Machine Learning Tutorial, 2025р. [Електронний ресурс]. URL: https://www.geeksforgeeks.org/machine-learning/machine-learning/ (дата звернення: 20.07.2025).

13. scikit-learn Machine Learning in Python, 2025р. [Електронний ресурс]. URL: https://docs.python.org/3/ (дата звернення: 18.07.2025).

14. The Unified Modeling Language, 2025р. [Електронний ресурс]. URL: https://www.uml-diagrams.org/#google_vignette (дата звернення: 15.07.2025).

15. Use Case Diagram - Unified Modeling Language (UML), 2025р. [Електронний ресурс]. URL: https://www.geeksforgeeks.org/system-design/use-case-diagram/ (дата звернення: 16.07.2025).

16. UML Class Diagram, 2025р. [Електронний ресурс]. URL: https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-class-diagrams/ (дата звернення: 20.07.2025).

17. Sequence Diagrams - Unified Modeling Language (UML), 2025р. [Електронний ресурс]. URL: https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/ (дата звернення: 27.07.2025).

18. Neural Networks, 2025р. [Електронний ресурс]. URL: https://docs.pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html (дата звернення: 19.07.2025).

19. BISM SSAS Tabular Documentation Tool, 2021р. [Електронний ресурс]. URL: https://docs.dataedo.com/docs/documenting-technology/supported-databases/analysis-services-multidimensional/ (дата звернення: 17.07.2025).

20. Deep Learning (Neural Networks), 2025р. [Електронний ресурс]. URL: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html (дата звернення: 23.07.2025).

21. Neural Network, 2025р. [Електронний ресурс]. URL: https://docs.logicaldoc.com/en/artificial-intelligence/models/neural-network (дата звернення: 23.07.2025).

ДОДАТКИ

Код створення бази даних

```
create database PharmacyData;

create table UserRole(
idRole int primary key,
Name varchar(30) not null,
Access varchar(30) not null
)

create table Users(
idUser int Primary key,
Surname varchar(35) not null,
Name varchar(25) not null,
Email varchar(50) not null,
idRole int not null,
Foreign key(idRole) references UserRole(idRole)
)

create table MlModel(
idModel int Primary key,
Type varchar(30) not null,
Description varchar(150) not null,
TrainingDate date not null,
Accuracy float not null
)

create table Product(
idProduct int Primary key,
Name varchar(40) not null,
DrugType varchar(35) not null,
ActiveIngredients varchar(80) not null,
SideEffects varchar(45) not null,
Description varchar(150) not null
TargetDisease varchar(60) not null;
DosageInfo varchar(60) not null;
Form varchar(30) not null;
InteractionWarnings varchar(150) null;
SeveritySideEffects int null;
)

create table Prediction(
idPrediction int primary key,
idProduct int not null,
idModel int not null,
Result float not null,
PredictedSideEffects text;
Date date not null,
Foreign Key(idProduct) references Product(idProduct),
Foreign Key(idModel) references MlModel(idModel)
)
```

```
create table AnalyticsResult(
idAnalyticsResult int Primary key,
idProduct int not null,
idPrediction int not null,
idUser int not null,
Conclusion varchar(150) not null,
Date date not null,
Foreign Key(idProduct) references Product(idProduct),
Foreign Key(idPrediction) references Prediction(idPrediction),
Foreign Key(idUser) references Users(idUser)
)
```

Код створення сховища даних

```
create database PharmacyDW;

create table MlModelDim(
IdModel int Primary key,
Type varchar(30) not null,
Description varchar(150) not null,
Accuracy float not null
)
create table TimeDim(
IdTime int Primary key,
Day INT not null,
Month INT not null,
Year INT not null
)

create table ProductDim(
idProduct int Primary key,
Name varchar(40),
DrugType varchar(35),
ActiveIngredients varchar(80),
SideEffects varchar(100),
SideEffectsCount int,
Description varchar(150),
TargetDisease varchar(60),
DosageInfo float,
Form varchar(30),
InteractionWarnings varchar(150),
SeveritySideEffects int,
)

CREATE TABLE PredictionFact (
IdProduct int not null,
IdModel int not null,
IdTime int not null,
Result float not null,
```

```
PredictedSideEffects text,
PredictedSideEffectsCount int,
PRIMARY KEY (IdProduct, IdModel, IdTime),
FOREIGN KEY (IdProduct) REFERENCES ProductDim(IdProduct),
FOREIGN KEY (IdModel) REFERENCES MlModelDim(IdModel),
FOREIGN KEY (IdTime) REFERENCES TimeDim(IdTime)
)
```

Частина коду функціонального додатку

```
class PharmaceuticalAnalysisSystem:

    def prepare_features(self, df):
        try:
            feature_columns = [
                'DrugType', 'Form', 'SeveritySideEffects',
                'ActiveIngredientsCount', 'SideEffectsCount'
            ]

            X = df[feature_columns]
            y = df['IsEffective']

            X_scaled = self.scaler.fit_transform(X)

            self.logger.info("Підготовка ознак завершена")
            return X_scaled, y, feature_columns

        except Exception as e:
            self.logger.error(f"Помилка підготовки ознак: {str(e)}")
            return None, None, None

    def train_models(self, X, y):
        try:
            X_train, X_test, y_train, y_test = train_test_split(
                X, y, test_size=0.2, random_state=42, stratify=y
            )

            models = {
                'LogisticRegression': LogisticRegression(random_state=42),
                'DecisionTree': DecisionTreeClassifier(random_state=42),
                'RandomForest': RandomForestClassifier(n_estimators=100, random_state=42)
            }

            results = {}
            for name, model in models.items():
                model.fit(X_train, y_train)
                y_pred = model.predict(X_test)

                accuracy = accuracy_score(y_test, y_pred)
                precision = precision_score(y_test, y_pred)
                recall = recall_score(y_test, y_pred)
                f1 = f1_score(y_test, y_pred)

                results[name] = {
                    'model': model,
                    'accuracy': accuracy,
                    'precision': precision,
                    'recall': recall,
                    'f1_score': f1
                }

                self.logger.info(f"Модель {name} навчена. Accuracy: {accuracy:.4f}")

            self.models.update(results)
            return results, X_test, y_test
```

```
        except Exception as e:
            self.logger.error(f"Помилка навчання моделей: {str(e)}")
            return None, None, None

    def build_neural_network(self, X, y):
        try:
            X_train, X_test, y_train, y_test = train_test_split(
                X, y, test_size=0.2, random_state=42, stratify=y
            )

            model = keras.Sequential([
                layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
                layers.Dropout(0.3),
                layers.Dense(32, activation='relu'),
                layers.Dropout(0.3),
                layers.Dense(16, activation='relu'),
                layers.Dense(1, activation='sigmoid')
            ])

            model.compile(
                optimizer='adam',
                loss='binary_crossentropy',
                metrics=['accuracy']
            )

            history = model.fit(
                X_train, y_train,
                epochs=50,
                batch_size=32,
                validation_split=0.2,
                verbose=0
            )

            y_pred_proba = model.predict(X_test)
            y_pred = (y_pred_proba > 0.5).astype(int).flatten()

            accuracy = accuracy_score(y_test, y_pred)
            precision = precision_score(y_test, y_pred)
            recall = recall_score(y_test, y_pred)
            f1 = f1_score(y_test, y_pred)

            nn_results = {
                'model': model,
                'accuracy': accuracy,
                'precision': precision,
                'recall': recall,
                'f1_score': f1,
                'history': history
            }

            self.models['NeuralNetwork'] = nn_results
            self.logger.info(f"Нейронна мережа навчена. Accuracy: {accuracy:.4f}")

            return nn_results

        except Exception as e:
            self.logger.error(f"Помилка навчання нейронної мережі: {str(e)}")
            return None
```

```
    def compare_models(self):
        try:
            comparison_data = []
            for model_name, model_info in self.models.items():
                if model_name != 'NeuralNetwork' or 'accuracy' in model_info:
                    comparison_data.append({
                        'Model': model_name,
                        'Accuracy': model_info['accuracy'],
                        'Precision': model_info['precision'],
                        'Recall': model_info['recall'],
                        'F1-Score': model_info['f1_score']
                    })

            comparison_df = pd.DataFrame(comparison_data)
            self.logger.info("Порівняння моделей завершено")
            return comparison_df

        except Exception as e:
            self.logger.error(f"Помилка порівняння моделей: {str(e)}")
            return None

    def predict_drug_properties(self, drug_data):
        try:
            processed_data = self.preprocess_single_drug(drug_data)

            if processed_data is None:
                return None

            predictions = {}
            for model_name, model_info in self.models.items():
                if model_name == 'NeuralNetwork':
                    model = model_info['model']
                    prediction_proba = model.predict(processed_data.reshape(1, -1))
                    prediction = (prediction_proba > 0.5).astype(int)[0][0]
                    confidence = float(prediction_proba[0][0] if prediction == 1 else 1 -
prediction_proba[0][0])
                else:
                    model = model_info['model']
                    prediction = model.predict(processed_data.reshape(1, -1))[0]
                    prediction_proba = model.predict_proba(processed_data.reshape(1, -1))
                    confidence = float(max(prediction_proba[0]))

                predictions[model_name] = {
                    'prediction': prediction,
                    'confidence': confidence,
                    'is_effective': bool(prediction)
                }

            return predictions

        except Exception as e:
            self.logger.error(f"Помилка прогнозування: {str(e)}")
            return None

    def preprocess_single_drug(self, drug_data):
        try:
            drug_df = pd.DataFrame([drug_data])

            drug_df['ActiveIngredientsCount'] = drug_df['ActiveIngredients'].apply(
```

```
                lambda x: len(str(x).split(',')) if pd.notna(x) else 0
            )

            drug_df['SideEffectsCount'] = drug_df['SideEffects'].apply(
                lambda x: len(str(x).split(',')) if pd.notna(x) and str(x) != 'Немає' else 0
            )

            categorical_columns = ['DrugType', 'Form', 'SeveritySideEffects']
            for col in categorical_columns:
                if col in drug_df.columns and col in self.label_encoders:
                    try:
                        drug_df[col] =
self.label_encoders[col].transform([str(drug_data.get(col, 'Невідомо'))])[0]
                    except ValueError:
                        drug_df[col] = 0

            feature_columns = ['DrugType', 'Form', 'SeveritySideEffects',
'ActiveIngredientsCount', 'SideEffectsCount']
            features = drug_df[feature_columns].values.flatten()

            features_scaled = self.scaler.transform([features])

            return features_scaled[0]

        except Exception as e:
            self.logger.error(f"Помилка обробки даних препарату: {str(e)}")
            return None

    def calculate_drug_efficiency(self, drug_data):
        try:
            weights = {
                'active_ingredient': 0.3,
                'dosage': 0.2,
                'form': 0.15,
                'target_disease': 0.2,
                'action_speed': 0.15
            }

            scores = {
                'active_ingredient':
self.estimate_active_ingredient_score(drug_data.get('ActiveIngredients', '')),
                'dosage': self.estimate_dosage_score(drug_data.get('DosageInfo', '')),
                'form': self.estimate_form_score(drug_data.get('Form', '')),
                'target_disease':
self.estimate_target_disease_score(drug_data.get('TargetDisease', '')),
                'action_speed': self.estimate_action_speed_score(drug_data.get('Form', ''))
            }

            side_effects_score = self.estimate_side_effects_score(drug_data.get('SideEffects',
''))

            base_efficiency = sum(scores[param] * weights[param] for param in scores)
            final_efficiency = base_efficiency * (1 - side_effects_score)

            return {
                'final_efficiency': final_efficiency,
                'base_efficiency': base_efficiency,
                'side_effects_score': side_effects_score,
                'parameter_scores': scores
```

```
        }

    except Exception as e:
        self.logger.error(f"Помилка розрахунку ефективності: {str(e)}")
        return None

def estimate_active_ingredient_score(self, ingredients):
    if not ingredients:
        return 0.5
    count = len(str(ingredients).split(','))
    return min(count * 0.1, 1.0)

def estimate_dosage_score(self, dosage):
    if not dosage:
        return 0.5
    return 0.7 if 'стандарт' in str(dosage).lower() else 0.5

def estimate_form_score(self, form):
    form_scores = {
        'таблетки': 0.8,
        'капсули': 0.9,
        'ін'єкції': 0.95,
                 'сироп': 0.7,
        'мазь': 0.6
    }
    return form_scores.get(str(form).lower(), 0.5)

def estimate_target_disease_score(self, disease):
    if not disease:
        return 0.5
    return 0.8 if 'вірус' in str(disease).lower() else 0.6

def estimate_action_speed_score(self, form):
    speed_scores = {
        'ін'
    єкції
    ': 0.9,
    'таблетки': 0.7,
    'капсули': 0.8,
    'сироп': 0.6,
    'мазь': 0.5
    }
    return speed_scores.get(str(form).lower(), 0.5)

def estimate_side_effects_score(self, side_effects):
    if not side_effects or str(side_effects) == 'Немає':
        return 0.1

    effects_count = len(str(side_effects).split(','))
    return min(effects_count * 0.05, 0.5)

def generate_analytics_report(self, df_with_clusters, comparison_df):
    try:
        report = {
            'timestamp': datetime.now(),
            'total_drugs_analyzed': len(df_with_clusters),
            'models_comparison': comparison_df.to_dict('records'),
            'clustering_summary': df_with_clusters['Cluster'].value_counts().to_dict(),
            'average_efficiency': df_with_clusters['PredictedEfficiency'].mean(),
```

```
                'efficiency_by_drug_type':
df_with_clusters.groupby('DrugType')['PredictedEfficiency'].mean().to_dict(),
                'side_effects_analysis': {
                    'average_side_effects_count': df_with_clusters['SideEffectsCount'].mean(),
                    'drugs_with_high_side_effects':
len(df_with_clusters[df_with_clusters['SideEffectsCount'] > 3])
                }
            }

            self.logger.info("Аналітичний звіт сформовано")
            return report

        except Exception as e:
            self.logger.error(f"Помилка генерації звіту: {str(e)}")
            return None

    def visualize_results(self, df_with_clusters, comparison_df):
        try:
            fig, axes = plt.subplots(2, 2, figsize=(15, 12))

            models_plot = comparison_df.set_index('Model')[['Accuracy', 'Precision', 'Recall',
'F1-Score']]
            models_plot.plot(kind='bar', ax=axes[0, 0], title='Порівняння моделей машинного
навчання')
            axes[0, 0].tick_params(axis='x', rotation=45)

            scatter = axes[0, 1].scatter(
                df_with_clusters['ActiveIngredientsCount'],
                df_with_clusters['PredictedEfficiency'],
                c=df_with_clusters['Cluster'],
                cmap='viridis',
                alpha=0.6
            )
            axes[0, 1].set_xlabel('Кількість активних інгредієнтів')
            axes[0, 1].set_ylabel('Ефективність')
            axes[0, 1].set_title('Кластеризація препаратів')
            plt.colorbar(scatter, ax=axes[0, 1])

            axes[1, 0].hist(df_with_clusters['PredictedEfficiency'], bins=20, alpha=0.7,
color='skyblue')
            axes[1, 0].set_xlabel('Ефективність')
            axes[1, 0].set_ylabel('Кількість препаратів')
            axes[1, 0].set_title('Розподіл ефективності препаратів')

            axes[1, 1].scatter(
                df_with_clusters['SideEffectsCount'],
                df_with_clusters['PredictedEfficiency'],
                alpha=0.6,
                color='red'
            )
            axes[1, 1].set_xlabel('Кількість побічних ефектів')
            axes[1, 1].set_ylabel('Ефективність')
            axes[1, 1].set_title('Залежність ефективності від побічних ефектів')

            plt.tight_layout()
            plt.savefig('pharmaceutical_analysis_results.png', dpi=300, bbox_inches='tight')
            plt.show()

            self.logger.info("Візуалізація результатів завершена")
```

```
    except Exception as e:
        self.logger.error(f"Помилка візуалізації: {str(e)}")

def save_models(self):
    try:
        for model_name, model_info in self.models.items():
            if model_name == 'NeuralNetwork':
                model_info['model'].save(f'{model_name}_model.h5')
            else:
                joblib.dump(model_info['model'], f'{model_name}_model.pkl')

        joblib.dump(self.scaler, 'scaler.pkl')
        joblib.dump(self.label_encoders, 'label_encoders.pkl')

        self.logger.info("Моделі та об'єкти збережено")
        return True

    except Exception as e:
        self.logger.error(f"Помилка збереження моделей: {str(e)}")
        return False

def load_models(self):
    try:
        model_files = {
            'LogisticRegression': 'LogisticRegression_model.pkl',
            'DecisionTree': 'DecisionTree_model.pkl',
            'RandomForest': 'RandomForest_model.pkl'
        }

        for model_name, filename in model_files.items():
            self.models[model_name] = {
                'model': joblib.load(filename),
                'accuracy': 0.0,
                'precision': 0.0,
                'recall': 0.0,
                'f1_score': 0.0
            }

        try:
            nn_model = keras.models.load_model('NeuralNetwork_model.h5')
            self.models['NeuralNetwork'] = {
                'model': nn_model,
                'accuracy': 0.0
            }
        except:
            self.logger.warning("Нейронну мережу не знайдено")

        self.scaler = joblib.load('scaler.pkl')
        self.label_encoders = joblib.load('label_encoders.pkl')

        self.logger.info("Моделі та об'єкти завантажено")
        return True

    except Exception as e:
        self.logger.error(f"Помилка завантаження моделей: {str(e)}")
        return False
```