

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА**


**Пояснювальна записка
до кваліфікаційної роботи бакалавра**

на тему: Застосування методів штучного інтелекту та машинного навчання
для розв'язання задач аналізу та прогнозування в освіті

Виконав: студент 4 курсу, групи КН 2022-1
Спеціальності F3(122) Комп'ютерні науки
(шифр і назва спеціальності)



— Станіслав КОЛІСНИК —
(прізвище та ініціали)

Керівник: СІЗОВА Н.Д. 
(прізвище та ініціали)

Рецензент: ПАХОМОВ Ю.В.
(прізвище та ініціали)

м. Харків – 2026 рік

Харківський національний університет міського господарства імені О. М. Бекетова

(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність F3(122)Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Марина НОВОЖИЛОВА

« 22 » червня 2026 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Колісник Станіслав Валерійович

(прізвище, ім'я, по батькові)

1. Тема роботи «Застосування методів штучного інтелекту та машинного навчання для розв'язання задач аналізу та прогнозування в освіті»

керівник роботи д.ф-м.н., проф. Сізова Н.Д.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти «22» травня 2026 р. № 440-03.

2. Термін подання студентом роботи 22 червня 2026 р.

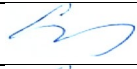



3. Вихідні дані до роботи Рекомендації для застосування методів штучного інтелекту та машинного навчання для розв'язання задач аналізу та прогнозування в освіті.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати предметну область та наявні аналоги; надати аргументоване обґрунтування вибору комплексу інструментального середовища та технічної платформи; визначити ключовий функціонал та провести його системний аналіз; розробити програмний код основних модулів та інтерфейс взаємодії з користувачем; представити опис архітектурної моделі та графічного інтерфейсу користувача та здійснити комплексне тестування програмного забезпечення.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

15 слайдів

6. Консультанти розділів роботи

Розділ	Ім'я та Прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	 Професор Наталія СІЗОВА	11.05.2026	23.05.2026
Розділ II	 Професор Наталія СІЗОВА	24.05.2026	02.06.2026
Розділ III	 Професор Наталія СІЗОВА	03.06.2026	10.06.2026
Розділ IV	 доцент Вікторія МАЛИШЕВА	11.06.2026	14.06.2026

7. Дата видачі завдання 11.05.2026**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми дипломної роботи	11.05.2026	Викон.
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки дипломної роботи	15.05.2026	Викон.
3	Написання I розділу	23.05.2026	Викон.
4	Написання II розділу	02.06.2026	Викон.
5	Написання III розділу	02.05.2026	Викон.
6	Написання IV розділу	14.06.2026	Викон.
7	Подання дипломної роботи керівнику	15.06.2026	Викон.
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	16.06.2026	Викон.
9	Подання доопрацьованого варіанту роботи керівнику	16.06.2026	Викон.
10	Захист матеріалів дипломної роботи на засіданні кафедри	18.06.2026	Викон.
11	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	23.06.2026	Викон.

Студент

(підпис)



Станіслав КОЛІСНИК

(прізвище та ініціали)

Керівник роботи

(підпис)



Наталія СІЗОВА

(прізвище та ініціали)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи бакалавра студента групи КН 2022-1 спеціальності F3(122) Комп'ютерні науки Колісника Станіслава Валерійовича за темою «Застосування методів штучного інтелекту та машинного навчання для розв'язання задач аналізу та прогнозування в освіті» складається з 4 розділів, містить 13 рисунків, 3 таблиці, 45 джерел.

У дипломній роботі вирішено актуальну науково-практичну задачу – теоретично обґрунтовано, математично формалізовано та програмно реалізовано інтелектуальні компоненти автоматизованої системи керування університетом на основі аналізу «цифрового сліду» студентів в LMS Moodle.

У першому розділі описано предметне середовище, розглянуто наявні аналоги

У аналітичному розділі сформульовано математичну постановку задачі та опис моделей ШІ в освіті.

У розділі програмного та технічного забезпечення виконано практичну реалізацію інтелектуальних компонентів системи автоматизованого керування університетом на основі даних LMS Moodle.

У розділі охорони праці визначені вимоги до організації робочого оточення та були запропоновані рекомендації щодо покращення умов праці та зниження рівня професійних ризиків.

Ключові слова: LMS MOODLE, ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, ПРЕДИКТИВНА АНАЛІТИКА, STUDENT RETENTION, GRADIENT BOOSTING, IRT, BAYESIAN KNOWLEDGE TRACING, КЛАСТЕРИЗАЦІЯ, ВИЯВЛЕННЯ АНОМАЛІЙ.

ANNOTATION

Structure and scope of work. Explanatory note of the qualification work of the bachelor's student of the group KN 2022-1, F3(122) specialty Computer Science, Kolisnyk Stanislav Valeriyovych, on the topic "Application of artificial intelligence and machine learning methods to solve problems of analysis and forecasting in education" consists of 4 sections, contains 13 figures, 3 tables, 45 sources.

The diploma work solves a relevant scientific and practical problem - theoretically substantiates, mathematically formalizes and programmatically implements the intellectual components of the automated university management system based on the analysis of the "digital footprint" of students in the LMS Moodle.

The first section describes the subject environment, considers existing analogues

The analytical section formulates the mathematical formulation of the problem and describes AI models in education.

In the software and hardware section, the practical implementation of the intellectual components of the automated university management system based on LMS Moodle data is carried out.

The occupational health and safety section defines requirements for the organization of the working environment and offers recommendations for improving working conditions and reducing occupational risks.

Keywords: LMS MOODLE, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, PREDICTIVE ANALYTICS, STUDENT RETENTION, GRADIENT BOOSTING, ITEM RESPONSE THEORY, BAYESIAN KNOWLEDGE TRACING, CLUSTERING, ANOMALY DETECTION.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	11
1.1 Опис предметного середовища	11
1.2 Актуальність та загальний стан досліджень.....	12
1.3 Огляд наявних аналогів.....	14
1.4 Постановка задачі	15
Висновки за розділом	17
РОЗДІЛ 2 АНАЛІТИЧНИЙ РОЗДІЛ	19
2.1 Математична постановка задачі	19
2.1.1 Зниження відсотка відрахувань (Student Retention).....	19
2.1.2 Оптимізація розподілу навантаження на викладачів	20
2.1.3 Підвищення позицій в академічних рейтингах.....	22
2.1.4 Отримання автоматичних сповіщень	23
2.2 Моделі штучного інтелекту та машинного навчання в освіті.....	26
2.2.1 Логістична регресія (Logistic Regression).....	27
2.2.2 Ансамблеві методи (Random Forest, Gradient Boosting).....	29
2.2.3 Сучасна теорія тестування (IRT) та Bayesian Knowledge Tracing (BKT)	33
2.2.4 Кластеризація студентів та виявлення аномалій (Unsupervised Learning)	36
Висновки за розділом	40
РОЗДІЛ 3 ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	42
3.1 Опис структури бази даних Moodle для задач ЩІ	42
3.1.1 Підсистема користувачів та авторизації (User Management).....	42
3.1.2 Підсистема курсів та зарахування (Course & Enrolment)	42
3.1.3 Підсистема «цифрового сліду» та логування (Clickstream & Activity Logs).....	43
3.1.4 Підсистема оцінювання та успішності (Gradebook)	43
3.1.5 Модулі навчальної діяльності (Activity Modules).....	43
3.2 Програмна реалізація логістичної регресії.....	44
3.3 Програмна реалізація ансамблевих моделей.....	49
3.4. Сучасна теорія тестування (IRT) та Bayesian Knowledge Tracing (BKT)	54
3.5 Кластеризація студентів та виявлення аномалій.....	58
Висновки за розділом	65
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	67

4.1 Організаційно-правові основи забезпечення безпеки праці.....	67
4.2 Характеристика об'єкта та виявлення потенційних небезпек	69
4.3. Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження	72
Висновки за розділом	75
ВИСНОВКИ	76
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТОК А Тексти програм	84

ВСТУП

Сучасний етап розвитку глобальної та вітчизняної вищої освіти характеризується масштабною цифровізацією всіх ланок навчального процесу [Moodle LMS Analytics]. Широке впровадження систем управління навчанням (LMS, зокрема Moodle, Google Classroom), автоматизованих систем управління (АСУ) «Деканат», сервісів електронного документообігу та цифрових платформ тестування призвело до накопичення безпрецедентних обсягів даних [Moodle LMS Analytics]. У цифрових репозиторіях закладів вищої освіти (ЗВО) щодня фіксуються детальні «цифрові сліди» студентів: від хронології логування та залученості у віртуальні класи до динаміки здачі поточних контрольних заходів [OU Analyse, Moodle LMS Analytics]. Проте безпосередній людський аналіз таких масивів інформації (Big Data) силами академічного персоналу (деканатів, кураторів, викладачів) є фізично неможливим через високу розмірність, розрізненість та динамічність даних [OU Analyse].

Проблема збереження контингенту та економічні виклики. Для українських та європейських ЗВО критично гострою постає проблема збереження контингенту студентів (Student Retention). Передчасне відрахування (Dropout) студентів, особливо на молодших курсах, зумовлене академічною неуспішністю, дезорієнтацією під час зміни середовища або втратою мотивації, спричиняє серйозні наслідки:

Економічні збитки: втрата фінансування для контрактної форми навчання та неефективне витрачання бюджетних коштів.

Репутаційні ризики: зниження позицій ЗВО у національних та міжнародних рейтингах (наприклад, Топ-200 Україна, QS World University Rankings).

Кадрові проблеми: скорочення контингенту безпосередньо призводить до зменшення навчального навантаження та, як наслідок, до вимушеного скорочення штату професорсько-викладацького складу.

Традиційні підходи до моніторингу контингенту є реактивними – адміністрація дізнається про кризу студента постфактум, під час підбиття підсумків сесії, коли вжити ефективних педагогічних чи організаційних заходів уже неможливо.

Науково-технічний потенціал ШІ та машинного навчання.

Вирішення окреслених проблем полягає у переході від констатуючого моніторингу до предиктивного (прогностичного) аналізу [OU Analyse]. Методи штучного інтелекту (ШІ) та алгоритми машинного навчання (МН) спроможні виявляти складні нелінійні залежності та приховані патерни у поведінці студентів задовго до початку екзаменаційної сесії [OU Analyse, Moodle LMS Analytics]. Застосування моделей класифікації (наприклад, градієнтного бустінгу чи випадкових лісів) та методів аналізу часових рядів дозволяє побудувати системи раннього попередження (Early Warning Systems) [OU Analyse]. Такі системи здатні вже на 4–6 тижні семестру виявляти студентів «групи ризику» з високою точністю, надаючи кураторам можливість вчасно скоригувати траєкторію навчання [OU Analyse].

Вимоги до прозорості моделей та сучасний тренд Explainable AI (XAI).

Додатковим чинником актуальності є еволюція самих методів ШІ. Донедавна більшість предиктивних систем в освіті функціонували за принципом «чорної скриньки» (Black Box), що викликало недовіру з боку освітян та порушувало етичні норми. Сучасний тренд розвитку комп'ютерних наук вимагає застосування методів інтерпретованого штучного інтелекту (Explainable AI, зокрема технологій SHAP та LIME). Побудова систем, які не просто прогнозують оцінку чи ризик відрахування, а й чітко пояснюють викладачеві аргументи та причини такого рішення ШІ (наприклад, брак активності на форумі, низький бал за перший колоквиум), є передовим напрямом наукових досліджень у світі.

Таким чином, розробка та впровадження математичних моделей і програмних засобів ШІ для аналізу та прогнозування показників навчання у ЗВО є вкрай актуальним завданням. Воно безпосередньо спрямоване на підвищення

якості вищої освіти, оптимізацію управління університетом та цифровізацію освітнього простору на основі об'єктивних даних (Data-Driven Approach).

РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Для закладів вищої освіти (ЗВО) опис предметного середовища має свою специфіку. Вона пов'язана зі складною структурою навчання (семестри, сесії, вибіркові дисципліни) та високими ризиками відрахування студентів.

Ключові стейкхолдери в університеті.

Студент: навчається за кредитно-модульною системою, обирає траєкторію (майнори/вибіркові курси).

Викладач / Куратор: оцінює бакалаврами/магістрами, веде наукову роботу, керує практикою.

Адміністрація (Деканат, Ректорат): контролює ліцензійні обмови, фінансування, акредитацію, кадровий склад.

Приймальна комісія: прогнозує обсяги набору та якість вступників.

Джерела даних у ЗВО (Інформаційне забезпечення)

Електронний деканат / АСУ: цифрові відомості, накази про зарахування/рух контингенту, пільги.

LMS (Moodle, Google Classroom): логи активності, завантажені лабораторні, оцінки за колоквіуми.

Дані ЄДЕБО / Вступної кампанії: бали ЗНО/НМТ, середній бал атестата, пріоритети заяв.

Наукометричні бази (Scopus, WoS): для аналізу та прогнозування ефективності викладачів.

Конкретні приклади задач для ЗВО

Прогнозування сесійної успішності: аналіз ризику отримання незадовільних оцінок під час першої сесії на основі вхідних балів НМТ.

Генерація індивідуальних освітніх траєкторій: рекомендація вибірових дисциплін на основі кар'єрних цілей студента та його попередніх оцінок.

Оптимізація розкладу: автоматичний розподіл аудиторного фонду з урахуванням потоків студентів та зайнятості професорів.

Аналіз ринку праці: прогнозування затребуваності освітніх програм (ОП) на основі парсингу вакансій.

Методи машинного навчання для вищої освіти.

Раннє попередження (Early Warning Systems): моделі класифікації (XGBoost, Random Forest), які вже на 4-му тижні навчання сигналізують куратору про пасивність студента.

Аналіз соціальних мереж (SNA): дослідження зв'язків у студентських групах для виявлення лідерів та ізольованих осіб.

1.2 Актуальність та загальний стан досліджень

У сучасній науковій літературі дослідження процесів цифровізації освіти дедалі більше фокусуються на двох взаємопов'язаних напрямках: освітньому майнінгу даних (Educational Data Mining – EDM) та навчальній аналітиці (Learning Analytics – LA). Науковці відзначають, що традиційні статистичні методи аналізу успішності вже не в змозі ефективно обробляти великі масиви гетерогенних даних (інформацію з LMS-систем, мобільних застосунків, результати тестів, поведінкові патерни). [1, 2, 3, 4]

У цьому контексті впровадження алгоритмів штучного інтелекту (ШІ) та машинного навчання (МН) розглядається як ключовий інструмент для переходу від констатувального аналізу до предиктивної (прогностичної) моделі управління освітою. [1, 2].

Сучасні закордонні та вітчизняні дослідження у цій сфері можна розділити на три магістральні вектори:

Прогнозування академічної успішності та виявлення студентів «групи ризику»

Створення класифікаційних та регресійних моделей для раннього прогнозування оцінок або загрози відрахування (drop-out rate). [1, 2]

Значний внесок у розвиток моделей ансамблевого навчання (Ensemble learning, як-от Random Forest, XGBoost) зробили Martinez Lunde et al. (2024) та Bellaj et al. (2024). Питання побудови гібридних моделей МН для оцінки залученості студентів висвітлено у працях Shah et al. (2024). [1, 2, 3]

У нещодавніх роботах, зокрема у фреймворках, опублікованих у журналі Nature (2026), доведено, що мультипарадигмальна інтеграція ШІ дозволяє досягти точності прогнозування зони ризику на рівні 97–99%, навіть за умов часткової втрати вхідних даних. [1]

Персоналізація навчання та адаптивне тестування

Використання ШІ для динамічної підбудови навчального контенту під індивідуальну швидкість сприйняття студента.

Моделі взаємодії «людина-комп'ютер» (HCI) у поєднанні з предиктивними алгоритмами активно розглядалися у дослідженнях Li et al. (2024). Дослідники доводять, що алгоритми навчання без вчителя (наприклад, кластеризація K-Means) дозволяють ефективно профілювати когнітивні стилі студентів. [1, 2, 3, 4, 5]

Інтерпретованість моделей штучного інтелекту (XAI – Explainable AI)

Подолання проблеми «чорної скриньки» в алгоритмах глибокого навчання (Deep Learning), що є критичним для прийняття рішень педагогами. [1]

Дослідження Raji et al. (2024), а також останні огляди Kesgin et al. (2026) акцентують на тому, що без технологій XAI (наприклад, методів SHAP чи LIME) викладачі не мають довіри до автоматизованих систем прогнозування. [1, 2]

Попри значні успіхи у точності роботи базових моделей, аналіз останніх публікацій дозволяє виявити ряд невирішених проблем:

Етичний та упереджений фактори: Багато алгоритмів демонструють приховану дискримінацію (algorithmic bias) за соціодемографічними ознаками, якщо початкова вибірка даних була незбалансованою. [1, 2]

Низький рівень адаптації до динамічних умов: Моделі, навчені на даних одного університету чи навчального року, різко втрачають точність (cross-institutional validation loss) при перенесенні в інші заклади освіти. [1]

Брак систем підтримки прийняття рішень (DSS) для викладача: Наявні моделі зазвичай лише видають сухий математичний прогноз (наприклад, "ймовірність успішного складання іспиту – 64%"), але не пропонують конкретних сценаріїв педагогічного втручання.

1.3 Огляд наявних аналогів

Огляд аналогів інтелектуальних систем аналітики та прогнозування в освіті доцільно розділити на дві категорії: глобальні комерційні платформи прогнозування успішності та вбудовані модулі предиктивної аналітики в LMS (Learning Management Systems).

Предиктивні системи виявлення груп ризику (Early Warning Systems)

OU Analyse (The Open University): Один із найуспішніших прикладів використання машинного навчання. Система будує прогноз на основі поведінкових патернів студента у віртуальному середовищі навчання (чи відкривав матеріали, як часто здавав тести завчасно). Модель оновлюється щотижня, сигналізуючи кураторам, яким саме студентам необхідна підтримка. [1, 2]

Othot (by Liaison): Комерційне рішення, орієнтоване на адміністрацію. Використовує просунуті алгоритми класифікації та регресії для прогнозування поведінки абітурієнтів і студентів. Дозволяє ЗВО моделювати сценарії: наприклад, «якщо ми збільшимо стипендію цьому студенту на 10%, як зміниться ймовірність його успішного випуску?». [1, 2, 3, 4]

Платформи адаптивного навчання та автооцінювання

Gradescope (Turnitin): Застосовує алгоритми комп'ютерного зору (Computer Vision) та NLP. Система самостійно групує схожі відповіді студентів (навіть рукописні або фрагменти коду). Викладач оцінює одну відповідь із групи, а ШІ автоматично виставляє бали іншим схожим роботам, одночасно прогнозуючи, які теми засвоєні найгіршими. [1]

ALEKS (McGraw Hill): Використовує теорію просторів знань (Knowledge Space Theory) та ШІ для постійної оцінки поточних навичок студента. Система

точно прогнозує, яку тему студент готовий вивчати наступною, адаптуючи контент у реальному часі. [1, 2]

Аналітика на базі великих мовних моделей (LLM-based analytics)

Сучасні тренди (з 2024–2026 рр.) зміщуються у бік поєднання традиційного машинного навчання з Prompt-based AI (LLM). Замість написання складного коду адміністрація ЗВО використовує моделі типу GPT/BERT для генерації інсайтів із великих масивів текстових даних (наприклад, аналіз коментарів студентів на форумах для прогнозування їхнього вигорання). [1]

Загальні недоліки існуючих аналогів

Проблема "Чорної скриньки": Комерційні системи не показують, *чому саме* модель вважає студента схильним до відрахування, що викликає етичні застереження та ризик упередженості.

Розрізненість даних: Більшість систем аналізують або лише активність в LMS, або лише оцінки в АСУ, не об'єднуючи їх в один профіль. [1, 2]

1.4 Постановка задачі

Сучасний етап розвитку вищої освіти в Україні та світі характеризується масштабною цифровізацією та форсованим переходом до гібридних і дистанційних моделей навчання. Основним інструментом забезпечення цього процесу стали системи управління навчанням (LMS), серед яких світове лідерство утримує платформа Moodle. Проте, попри високу технологічну інтеграцію, заклади вищої освіти (ЗВО) стикаються із серйозними викликами: зниженням рівня безпосереднього контролю за залученістю студентів, складністю раннього виявлення академічних ризиків та високим відсотком відрахувань через невчасну ідентифікацію проблем у засвоєнні матеріалу.

Традиційні підходи до оцінювання освітнього процесу, що базуються переважно на статичних підсумкових оцінках (результатах сесій чи модульних контролів), мають ретроспективний характер. Вони констатують факт академічної заборгованості тоді, коли вжити ефективних педагогічних заходів уже запізно.

Водночас у процесі взаємодії користувачів із платформою Moodle щосекундно генеруються величезні масиви поведінкових даних – цифрові сліди у вигляді системних логів (Moodle logs). Ці дані містять прихований потенціал для об'єктивного моніторингу динаміки навчання в режимі реального часу.

Застосування методів штучного інтелекту (ШІ) та машинного навчання (МН) у межах концепції Educational Data Mining (EDM) та Learning Analytics (LA) відкриває принципово нові можливості для аналізу цих даних. Використання алгоритмів класифікації, регресії та кластеризації дозволяє трансформувати сирі, неструктуровані логи Moodle у змістовні метрики студентської прокрастинації, регулярності та залученості. Це робить можливим не лише оперативний аналіз поточної ситуації, а й раннє проактивне прогнозування результатів навчання, моделювання індивідуальних освітніх траєкторій та автоматизацію підтримки студентів [1, 2].

З огляду на це, науково обґрунтоване розроблення та впровадження інтелектуальних моделей аналізу логів Moodle є надзвичайно актуальним завданням, розв'язання якого дозволить підвищити якість вищої освіти, оптимізувати менеджмент ЗВО та забезпечити своєчасну підтримку здобувачів освіти.

Мета дослідження – підвищення ефективності управління освітнім процесом у ЗВО шляхом розроблення та впровадження методів штучного інтелекту і машинного навчання для аналізу цифрових слідів (логів Moodle) та раннього прогнозування успішності студентів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати сучасний стан та світовий досвід застосування технологій Learning Analytics та Educational Data Mining у вищій школі.
- Дослідити структуру та специфіку поведінкових даних (логів), що генеруються платформою Moodle, та розробити методику їхнього попереднього очищення й агрегації.

- Обґрунтувати та сформулювати комплекс інформативних ознак (Feature Engineering), які релевантно відображають залученість, прокрастинацію та регулярність роботи студентів.
- Обрати, адаптувати та навчити пул моделей машинного навчання (зокрема, алгоритми ансамблевого навчання та логістичної регресії) для раннього прогнозування ризиків академічної неспішності.
- Провести експериментальну перевірку розроблених моделей на реальних даних конкретного ЗВО та оцінити їхню точність за допомогою стандартних метрик (Accuracy, F1-score, AUC-ROC).

Об'єкт дослідження – процес моніторингу та прогнозування навчальної діяльності студентів закладів вищої освіти в умовах цифрового освітнього середовища.

Предмет дослідження – методи, моделі та алгоритми штучного інтелекту і машинного навчання для інтелектуального аналізу поведінкових логів системи Moodle з метою прогнозування успішності.

Висновки за розділом

У розділі наведено опис предметного середовища, проаналізовано актуальність та загальний стан досліджень, зроблено огляд наявних аналогів.

Аналіз останніх досліджень свідчить про високий потенціал МН в освіті, проте питання створення стійких до шуму, етично нейтральних та інтерпретованих моделей прогнозування академічних результатів залишається відкритим. Це зумовлює вибір мети дослідження – підвищення ефективності управління освітнім процесом у ЗВО шляхом розроблення та впровадження методів штучного інтелекту і машинного навчання для аналізу цифрових слідів (логів Moodle) та раннього прогнозування успішності студентів.

Об'єкт дослідження – процес моніторингу та прогнозування навчальної діяльності студентів закладів вищої освіти в умовах цифрового освітнього середовища.

Предмет дослідження – методи, моделі та алгоритми штучного інтелекту і машинного навчання для інтелектуального аналізу поведінкових логів системи Moodle з метою прогнозування успішності.

РОЗДІЛ 2

АНАЛІТИЧНИЙ РОЗДІЛ

Ефективне управління сучасним університетом вимагає комплексного підходу до управління даними (Data Governance). Якісний менеджмент даних дозволяє автоматизувати рутину, прогнозувати ризики та приймати точні управлінські рішення.

2.1 Математична постановка задачі

2.1.1 Зниження відсотка відрахувань (Student Retention)

Для утримання студентів необхідно перейти від реактивного вирішення проблем (робота з фактом заборгованості) до проактивного (попередження відрахування).

- Предиктивна аналітика: створення AI-моделей, які на основі історичних даних виявляють студентів із групи ризику ще до початку сесії.
- Маркери раннього попередження (Early Warning Signals):
 - Цифровий слід: низька активність у LMS (Moodle, Google Classroom) протягом перших 2–3 тижнів семестру.
 - Поточна успішність: пропуски дедлайнів, низькі бали за перші модульні контролю.
 - Соціально-економічні фактори: затримка оплати за навчання, тривала відсутність у гуртожитку.
- Автоматизація сповіщень: система автоматично сигналізує куратору або психологічній службі про критичне зниження показників студента для вчасної інтервенції (менторство, додаткові консультації).

Постановка задачі: Класифікація (предиктивний скоринг) студентів та прогнозування ризику відрахування.

Нехай $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ – множина студентів, а $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ – вектор ознак (факторів ризику) для кожного студента.

Вектор ознак для студента s_i включає:

- x_{s1} – відсоток пропущених занять (0–1).
- x_{s2} – поточний середній бал (GPA) відносно прохідного.
- x_{s3} – активність у LMS (кількість кліків або годин на тиждень).
- x_{s4} – кількість днів прострочення дедлайнів.
- x_{s5} – наявність фінансової заборгованості (бінарна: 0 або 1).

Математична модель.

Задача зводиться до знаходження функції ймовірності відрахування студента:

$$P(\text{Drop}_i) = \sigma(\mathbf{w}^T \mathbf{x}_i + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x}_i + b)}} \quad (2.1)$$

Де

\mathbf{w} - вектор вагових коефіцієнтів ознак,

b - параметри системи, що навчаються на історичних даних за минулі роки),

σ – логістична функція.

Критерій прийняття рішень (Алерт-система): формується тригерна функція для куратора.

$$\text{Action}(s_i) = \begin{cases} \text{"Червона зона"} \text{ (Термінове втручання),} & \text{якщо } P(\text{Drop}_i) \geq \alpha \\ \text{"Жовта зона"} \text{ (Моніторинг),} & \text{якщо } \beta \leq P(\text{Drop}_i) < \alpha \\ \text{"Зелена зона"} \text{ (Норма),} & \text{якщо } P(\text{Drop}_i) < \beta \end{cases} \quad (2.2)$$

(де α, β – встановлені адміністрацією пороги чутливості).

2.1.2 Оптимізація розподілу навантаження на викладачів

Дисбаланс у навантаженні призводить до вигорання викладачів та зниження якості викладання. Оптимізація базується на інтеграції кадрових та навчальних даних.

- Централізований облік усіх видів діяльності: створення єдиної бази даних, де фіксується не лише аудиторне навантаження, а й наукова, методична та організаційна робота.
- Автоматизоване планування: впровадження алгоритмів, які розподіляють години з урахуванням:
 - Кваліфікації та наукових інтересів викладача.
 - Рейтингу викладача за результатами попередніх опитувань студентів.
 - Обмежень щодо максимального денного/тижневого навантаження.
- Прозорість та гнучкість: створення особистих кабінетів викладача, де в реальному часі видно динаміку виконання плану, що мінімізує людський фактор та фаворитизм при розподілі годин.

Постановка задачі: Задача лінійного цілочисельного програмування (дискретна оптимізація).

Нехай:

- $T = \{t_1, t_2, \dots, t_I\}$ – множина викладачів.
- $C = \{c_1, c_2, \dots, c_J\}$ – множина навчальних курсів (дисциплін), які треба розподілити.
- H_j – обсяг годин, необхідний для викладання курсу c_j .
- V_i – нормативне (бажане) річне навантаження для викладача t_i .
- $E_{ij} \in [0,10]$ – коефіцієнт відповідності (експертність/компетенція) викладача t_i

Цільова функція (максимізація якості викладання та мінімізація дисбалансу).

$$\max \sum_{i=1}^I \sum_{j=1}^J E_{ij} \cdot z_{ij} - \gamma \sum_{i=1}^I \left| \sum_{j=1}^J H_j \cdot z_{ij} - V_i \right| \quad (2.3)$$

Де

$z_{ij} \in [0,1]$ – булева змінна (1, якщо курс j) закріплено за викладачем i ; 0 – в іншому випадку),

γ – штрафний коефіцієнт за відхилення від норми годин/

Обмеження моделі (Constraints).

Кожен курс має викладатися:

$$\sum_{i=1}^I z_{ij} = 1 \quad \forall j \in \{1, \dots, J\} \quad (2.4)$$

Граничне навантаження викладача:

$$\sum_{j=1}^J H_j \cdot z_{ij} \leq V_i^{\max} \quad \forall i \in \{1, \dots, I\} \quad (2.5)$$

Мінімальне навантаження викладача:

$$\sum_{j=1}^J H_j \cdot z_{ij} \geq V_i^{\min} \quad \forall i \in \{1, \dots, I\} \quad (2.6)$$

Профільність (не призначати непрофільні курси):

$$z_{ij} = 0, \text{ якщо } E_{ij} < E_{\text{threshold}} \quad (2.7)$$

2.1.3 Підвищення позицій в академічних рейтингах

Міжнародні рейтинги оцінюють університети переважно на основі даних.

Неточні або неповні дані – це втрачені позиції.

- Менеджмент наукометричних даних (CRIS-системи): впровадження систем контролю за профілями науковців у Scopus, Web of Science та Google Scholar. Автоматичний моніторинг правильної афіліації університету (усунення помилок у написанні назви ЗВО, через які втрачаються цитування).
- Аналітика міжнародної співпраці: відстеження спільних публікацій з іноземними колегами для покращення показника International Research Network.
- Збір даних про репутацію: ведення актуальних баз даних випускників (Alumni) та роботодавців-партнерів для точного та масового таргетування під час експертних опитувань QS/THE (Academic & Employer Reputation).

- Оцифрування показників сталого розвитку (GreenMetric / THE Impact): збір та структурування даних про енергоефективність, інклюзивність та гендерну рівність в університеті, що зараз критично важливо для глобального визнання.

Це дозволяє керівництву бачити реальну картину закладу на інтерактивних дашбордах та приймати рішення на основі цифр, а не інтуїції.

Постановка задачі: Багатокритеріальний аналіз та максимізація зваженої суми індикаторів.

Глобальний рейтинг університету R формується як зважена сума окремих нормалізованих метрик. Наприклад, спрощена модель для QS/THE:

$$R = \sum_{k=1}^K w_k \cdot M_k \quad (2.8)$$

Де

M_k – значення k -ї метрики університету,

w_k – вага цієї метрики в рейтингу ($\sum w_k = 1$).

2.1.4 Отримання автоматичних сповіщень

Для викладача автоматизація аналізу помилок – це найкращий спосіб миттєво адаптувати навчальну програму та допомогти групі.

Алгоритм роботи модуля.

1. Збір даних: Студенти проходять тестування у системі (LMS, наприклад, Moodle, Google Classroom, Canvas).

2. Аналіз системи: Алгоритм рахує відсоток неправильних відповідей у кожному питанні та групує їх за тегами (темами).

3. Тригер сповіщення: Якщо відсоток помилок у конкретній темі перевищує ліміт (наприклад, понад 40% студентів відповіли неправильно), система формує звіт.

4. Доставка: Викладач отримує пуш-сповіщення, лист на пошту або повідомлення у робочий чат (Slack/Teams).

Таке сповіщення містить:

- Назва теми: Наприклад, "Тема 3".
- Статистика: "65% студентів провалили питання №4 та №7".
- Конкретний маркер: Чи це була помилка у розрахунках, чи повне нерозуміння терміну.
- Кнопка дії: Швидкий перехід до аналізу відповідей конкретних студентів.

Переваги для навчального процесу

- Економія часу: Не потрібно вручну переглядати десятки тестів та вираховувати середній бал.
- Вчасне реагування: Можна наздогнати матеріал на наступній лекції, поки студенти не забули тему.
- Корекція тестів: Допомогає побачити занадто складні або некоректно сформульовані питання.

Математична постановка задачі аналізу результатів тестування для виявлення проблемних тем полягає в моделюванні процесу оцінювання та визначенні критеріїв, за якими тема вважається «складною».

Вхідні дані та множини.

Нехай задано такі параметри системи:

- $S = \{s_1, s_2, \dots, s_N\}$ – множина студентів, де N – загальна кількість студентів, що проходили тест.
- $T = \{t_1, t_2, \dots, t_M\}$ – множина тем, що вивчаються, де M – загальна кількість тем.
- $Q = \{q_1, q_2, \dots, q_K\}$ – множина питань у тесті, де K – загальна кількість питань.

Зв'язок між питаннями та темами задається матрицею відповідності W розміром $K \times M$:

$$w_{ki} = \begin{cases} 1, & \text{якщо питання } q_k \text{ належить до теми } t_i \\ 0, & \text{в іншому випадку} \end{cases} \quad (2.8)$$

Результати тестування формують матрицю відповідей R розміром $N \times K$:

$$r_{jk} = \begin{cases} 1, & \text{якщо студент } s_j \text{ відповів на питання } q_k \text{ правильно} \\ 0, & \text{якщо студент } s_j \text{ відповів на питання } q_k \text{ неправильно} \end{cases} \quad (2.9)$$

Розрахунок показників складності

Крок 1. Визначення кількості помилок за кожним питанням.

Загальна кількість студентів, які відповіли неправильно на питання q_k , дорівнює:

$$E_k = \sum_{j=1}^N (1 - r_{jk}) \quad (2.10)$$

Крок 2. Розрахунок середнього рівня помилок для кожної теми

Щоб знайти загальну кількість неправильних відповідей у межах конкретної теми (t_i) , необхідно врахувати всі питання, що до неї відносяться:

$$Err(t_i) = \sum_{k=1}^K (w_{ki} \cdot E_k) \quad (2.11)$$

Загальна кількість спроб відповісти на питання з теми t_i усіма студентами дорівнює:

$$Total(t_i) = N \cdot \sum_{k=1}^K w_{ki} \quad (2.12)$$

Тоді коефіцієнт складності теми (t_i) (частка помилок) обчислюється як:

$$K_{err}(t_i) = \frac{Err(t_i)}{Total(t_i)} = \frac{\sum_{k=1}^K (w_{ki} \cdot \sum_{j=1}^N (1 - r_{jk}))}{N \cdot \sum_{k=1}^K w_{ki}} \quad (2.13)$$

Умова генерації автоматичного сповіщення

Задано граничне допустиме значення (поріг) помилок $\alpha \in [0,1]$, яке встановлює викладач.

Для кожної теми t_i перевіряється логічна умова виходу критерію за межі норми:

$$Notification(t_i) = \begin{cases} 1, & \text{якщо } K_{err}(t_i) \geq \alpha \quad \longrightarrow \text{генерувати сповіщення} \\ 0, & \text{якщо } K_{err}(t_i) < \alpha \quad \longrightarrow \text{ігнорувати} \end{cases} \quad (2.13)$$

Цільова функція (Оптимізація вибору тем).

Завдання системи – знайти підмножину проблемних тем $T_{alert} \subset T$, які вимагають повторного пояснення:

$$T_{alert} = \{t_i \in T \mid Notification(t_i) = 1\} \quad (2.14)$$

Для додаткового ранжування сповіщень теми в списку сортуються за спаданням коефіцієнта складності, щоб викладач бачив найкритичніші проблеми першими: сортировка T_{alert} за умовою $\max K_{err}(t_i)$

Математична модель дозволяє системі автоматично обчислювати коефіцієнт помилок для кожної теми на основі бінарної матриці відповідей студентів та фільтрувати теми за заданим порогом α для миттєвого інформування викладача.

2.2 Моделі штучного інтелекту та машинного навчання в освіті

Застосування штучного інтелекту (ШІ) та машинного навчання (МН) в освіті дозволяє відійти від жорстких статичних порогів (як у класичній математичній постановці) та перейти до інтелектуального аналізу: прогнозування успішності, кластеризації студентів за стилями навчання та адаптивного тестування.

Нижче наведено основні математичні моделі та методи ШІ для розв'язання цих задач.

Таблиця 2.1 – Порівняльна матриця застосування моделей

Задача в освіті	Математичний метод ШІ	Що оцінює / прогнозує
Прогноз відрахувань	Логістична регресія, XGBoost	Ймовірність $P(y = 1)$ провалу курсу
Аналіз якості питань	М-параметричні моделі IRT	Складність b_k та дискримінативність a_k
Адаптивне навчання	Блоки ВКТ, Reinforcement Learning	Наступний найкращий крок/тема для вивчення

Поділ на фокус-групи	Кластеризація (k-Means)	Виявлення паттернів поведінки студентів
----------------------	-------------------------	---

2.2.1 Логістична регресія (Logistic Regression)

Логістична регресія (Logistic Regression) у контексті освітньої аналітики – це базова та найбільш інтерпретована модель машинного навчання. Вона використовується для бінарної класифікації, тобто для прогнозування подій із двома варіантами результату (наприклад: «студент здасть іспит / не здасть», «засвоїв тему / не засвоїв», «потрапить до групи ризику / не потрапить»).

Нижче наведено розгорнуту математичну та практичну специфікацію логістичної регресії для освітніх задач.

Замість того, щоб прогнозувати безпосередньо оцінку (як у лінійній регресії), логістична регресія прогнозує ймовірність P того, що студент належить до певного класу (наприклад, класу 1 – "група ризику").

Лінійна комбінація ознак

Спочатку обчислюється проміжна величина z , яка є лінійною комбінацією факторів поведінки або успішності студента:

$$z = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d = w^T x + w_0 \quad (2.15)$$

Де:

(x_1, x_2, \dots, x_d) – ознаки студента (наприклад: кількість пропущених занять, середній бал за попередні тести, час, проведений в LMS, кількість спроб здачі).

w_1, w_2, \dots, w_d – вагові коефіцієнти, які модель підбирає під час навчання (показують вплив кожного фактора).

w_0 – зсув (bias), базовий рівень.

Сигмоїд-функція (Логістична функція)

Оскільки значення z може бути будь-яким числом від $-\infty$ до $+\infty$, його потрібно стиснути в діапазон від 0 до 1, щоб отримати ймовірність. Для цього використовується сигмоїда:

$$P(y = 1 | x) = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(w^T x + w_0)}} \quad (2.16)$$

Якщо $z \rightarrow +\infty$, то $P \rightarrow 1$.

Якщо $z \rightarrow -\infty$, то $P \rightarrow 0$.

Якщо $z = 0$, то $P = 0.5$.

Прийняття рішення (Класифікація)

Після обчислення ймовірності P , система приймає рішення на основі заданого порогу τ (зазвичай $\tau = 0.5$):

$$\hat{y} = \begin{cases} 1, & \text{якщо } P \geq \tau \quad (\text{студент у зоні ризику}) \\ 0, & \text{якщо } P < \tau \quad (\text{студент успішно засвоює матеріал}) \end{cases} \quad (2.17)$$

Навчання моделі: Функція втрат.

Для пошуку оптимальних ваг (w) модель не може використовувати метод найменших квадратів (як у лінійній регресії), тому що функція буде не опуклою. Використовується функція логарифмічних втрат (Log-Loss / Binary Cross-Entropy):

$$L(w) = -\frac{1}{N} \sum_{j=1}^N [y_j \log(P_j) + (1 - y_j) \log(1 - P_j)] \quad (2.18)$$

Де:

- N – кількість студентів у навчальній вибірці.
- y_j – реальний результат j -го студента (1 або 0).
- P_j – спрогнозована моделлю ймовірність для j -го студента.

Суть мінімізації: Якщо реальний результат $y_j = 1$, а модель прогнозує $P_j \rightarrow 0$, штраф (помилка) прагне до нескінченності. Модель штрафує саму себе за впевнені, але неправильні прогнози. Оптимальні ваги знаходяться чисельно за допомогою градієнтного спуску (Gradient Descent).

Переваги логістичної регресії

Висока інтерпретованість (Explainable AI): Викладач або адміністратор чітко бачить ваги w . Можна сказати: "Кожен день відсутності студента на платформі збільшує шанси провалу на стільки-то відсотків". Це не "чорна скринька", як нейромережі.

Швидкість роботи: Модель обчислюється миттєво навіть на слабких серверах LMS.

Вихід у вигляді ймовірностей: Дозволяє сортувати студентів за ступенем ризику (від 99% до 51%) та фокусувати увагу викладача на найбільш критичних випадках.

Обмеження логістичної регресії:

Лінійна залежність: Модель припускає, що внесок кожної ознаки є лінійним. Вона погано вловлює складні закономірності (наприклад, якщо студент робить багато помилок, але при цьому дивиться відео втричі довше за інших – лінійна регресія може заплутатися). Для цього потрібні дерева рішень або поліноміальні ознаки.

Чутливість до мультиколінеарності: Якщо дві ознаки сильно пов'язані між собою (наприклад, "загальний час на платформі" та "час перегляду лекцій"), то ваги моделі стають нестабільними.

2.2.2 Ансамблеві методи (Random Forest, Gradient Boosting)

Ансамблеві методи (Ensemble Methods) – це підхід у машинному навчанні, за якого для розв'язання задачі використовується не один алгоритм, а ціла група (ансамбль) базових моделей (зазвичай дерев рішень).

В освітній аналітиці ансамблі застосовуються тоді, коли зв'язок між діями студента та його успішністю є складним і нелінійним (наприклад: якщо студент не зайшов на платформу 5 днів поспіль – це погано, але якщо він при цьому здав усі лабораторні на 100% раніше дедлайну – він не в групі ризику, а навпаки, є лідером).

Класична логістична регресія такі нюанси вловлює погано, а дерева рішень та їхні ансамблі – ідеально.

Розглянемо дві головні архітектури ансамблів: Random Forest (ідея беггінгу) та Gradient Boosting (ідея бустингу).

Базовий елемент: Дерево рішень (Decision Tree)

Дерево рішень розбиває простір ознак студентів на підмножини за допомогою послідовних бінарних правил (іф-то-інакше).

Критерій розщеплення: У кожному вузлі дерево вибирає ознаку (наприклад, x_1 – кількість помилок) та поріг τ , які максимально розділяють студентів на однорідні групи (успішні / відстаючі).

Для оцінки якості розщеплення зазвичай мінімізують неоднорідність Джині (Gini Impurity) або ентропію:

$$Gini = 1 - \sum_{c=1}^C p_c^2 \quad (2.19)$$

Де p_c – частка студентів класу c у поточному вузлі.

Випадковий ліс (Random Forest)

Random Forest – це ансамбль, у якому велика кількість глибоких дерев рішень навчаються паралельно та незалежно одне від одного.

Математична та алгоритмічна суть:

Bagging (Bootstrap Aggregating): З початкової вибірки студентів розміром N випадковим чином із поверненням формується B нових вибірок такого ж розміру. Це означає, що деякі студенти потраплять в одну вибірку кілька разів, а деякі – ні разу.

Метод випадкових підпросторів: Під час побудови кожного вузла в дереві обирається випадкова підмножина з m ознак (де $m < d$, зазвичай $m = \sqrt{d}$). Це робить дерева максимально різноманітними та не схожими одне на одне.

Агрегація результатів:

Для класифікації (група ризику чи ні) фінальне рішення приймається шляхом мажоритарного голосування (більшість дерев "за"):

$$\hat{y} = \text{mode}\{f_1(x), f_2(x), \dots, f_B(x)\} \quad (2.20)$$

Для регресії (прогноз точного балу за іспит) береться середнє арифметичне:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B f_b(x) \quad (2.21)$$

Random Forest майже не схильний до перенавчання (overfitting). Якщо кілька тестів містили випадкові аномалії в оцінках, "ліс" нівелює цей шум за рахунок усереднення сотні різних дерев.

Градiєнтний бустинг (Gradient Boosting)

Gradient Boosting – це архітектура, де дерева рішень будуються послідовно (одне за одним), і кожне наступне дерево виправляє помилки попередніх.

Модель шукає прогноз у вигляді зваженої суми M базових дерев:

$$F_M(x) = \sum_{m=1}^M \gamma_m h_m(x) \quad (2.22)$$

Де $h_m(x)$ – нове слабке дерево, а γ_m – його вага.

Процес навчання є ітераційним:

Система робить базовий (навіть грубий) прогноз $F_0(x)$ – наприклад, просто середній бал усіх студентів.

Для кожного студента обчислюються залишкові помилки (антиградієнт функції втрат). Якщо функція втрат – це середньоквадратична помилка (MSE), то залишок для (j -го студента на кроці m дорівнює:

$$r_{jm} = y_j - F_{m-1}(x_j) \quad (2.23)$$

Нове дерево $h_m(x)$ навчається не на фінальні оцінки y , а на ці залишкові помилки r . Тобто воно намагається передбачити, де саме помилилася попередня модель.

Прогноз оновлюється з урахуванням швидкості навчання (learning rate ($\eta \in (0,1]$):

$$F_m(x) = F_{m-1}(x) + \eta \cdot \gamma_m h_m(x) \quad (2.24)$$

На практиці використовують оптимізовані бібліотеки бустингу: XGBoost, LightGBM або CatBoost. Вони використовують регуляризацію (штрафи за надто складні дерева) та працюють надзвичайно швидко.

Таблиця 2.1. Порівняння Random Forest та Gradient Boosting в освітніх задачах

Критерій	Random Forest	Gradient Boosting (XGBoost / LightGBM)
Принцип роботи	Паралельне навчання незалежних дерев	Послідовне виправлення помилок
Схильність до перенавчання	Дуже низька, легко налаштувати	Вища, потребує ретельного підбору гіперпараметрів (η), глибина)
Точність прогнозу	Висока	Максимальна (зазвичай перемагає на складних табличних даних)
Обробка пропусків	Потребує заповнення пропусків у даних	Може автоматично обробляти пропуски (наприклад, якщо студент взагалі не проходив певний тест)
Для яких задач краще	Стабільний базовий прогноз, коли історичних даних мало	Коли даних багато, а точність прогнозу критична (наприклад, автоматичне зарахування на гранти)

Обидва методи вирішують головний мінус складних моделей – проблему "чорної скриньки". Вони дозволяють обчислити Feature Importance (важливість ознак) за допомогою критерію зменшення неоднорідності (Mean Decrease in Impurity):

Система видає викладачу математично обґрунтований графік:

1. Перегляд лекції №4 – 42% впливу на результат.
2. Час виконання тесту №2 – 25% впливу.
3. Кількість спроб у тесті №1 – 10% впливу.

Це дозволяє викладачу зрозуміти, яка саме активність або тема є ключовим тригером успіху чи провалу всієї групи.

2.2.3 Сучасна теорія тестування (IRT) та Bayesian Knowledge Tracing (BKT)

Сучасна теорія тестування (Item Response Theory – IRT) та Байєсівське відстеження знань (Bayesian Knowledge Tracing – BKT) – це фундаментальні ймовірнісні моделі, які використовуються у високотехнологічних освітніх системах (наприклад, Duolingo, Khan Academy, Coursera).

На відміну від логістичної регресії чи ансамблів, які просто шукають кореляції у поведінкових факторах, IRT та BKT безпосередньо моделюють латентні (приховані) змінні – реальний рівень знань студента та динаміку його зміни в часі.

Сучасна теорія тестування (Item Response Theory – IRT)

Класична теорія тестування оцінює студента за формулою: «бал = кількість правильних відповідей». Це несправедливо, якщо одному студенту дістався легкий тест, а іншому – складний. IRT вирішує цю проблему, розділяючи рівень знань студента та характеристики самих питань.

Вона базується на припущенні, що відповідь на питання – це випадкова величина, ймовірність якої залежить від сили студента та складності завдання.

Однопараметрична модель (Модель Раши / IPL)

Найпростіший варіант, де кожне питання має лише один параметр – складність. Ймовірність P_j того, що студент j правильно відповість на питання k , дорівнює:

$$P(r_{jk} = 1 | \theta_j) = \frac{1}{1 + e^{-(\theta_j - b_k)}} \quad (2.25)$$

Де:

θ_j – латентний (прихований) рівень знань студента $\setminus(j\setminus)$.

b_k – параметр складності питання $\setminus(k\setminus)$.

Якщо рівень знань дорівнює складності $\theta_j = b_k$, то ймовірність правильної відповіді становить 50.

Двопараметрична модель (2PL)

Додається параметр, який враховує якість самого питання:

$$P(r_{jk} = 1 | \theta_j) = \frac{1}{1 + e^{-a_k(\theta_j - b_k)}} \quad (2.26)$$

Де a_k – розрізнявальна здатність (discrimination) питання k . Вона показує, наскільки круто піднімається крива ймовірності. Якщо a_k високе, то питання чітко розділяє студентів на тих, хто знає (ймовірність стрімко йде до 1), і тих, хто не знає (ймовірність падає до 0). Якщо a_k низьке або від'ємне – питання сформульовано некоректно (на нього однаково часто помилково відповідають і відмінники, і двоєчники).

Трипараметрична модель (3PL)

Враховує фактор тестування у вигляді множинного вибору, де відповідь можна вгадати:

$$P(r_{jk} = 1 | \theta_j) = c_k + (1 - c_k) \frac{1}{1 + e^{-a_k(\theta_j - b_k)}} \quad (2.27)$$

Де c_k – параметр вгадування (guessing). Навіть якщо знання студента прямують до мінус нескінченності ($\theta \rightarrow -\infty$) ймовірність відповісти правильно все одно дорівнює c_k (наприклад, 0.25 для тесту з 4 варіантами відповідей).

Система дає студенту питання середньої складності. Якщо він відповів правильно, ШІ перераховує $\setminus(\theta_j\setminus)$ у бік збільшення і наступним дає більш складне питання. Якщо помилився – легше. Тест припиняється, коли похибка

визначення θ_j стає меншою за заданий ліміт. Це дозволяє точно оцінити знання за 10–15 питань замість стандартних 50.

Байєсівське відстеження знань (Bayesian Knowledge Tracing – BKT)

Якщо IRT частіше застосовується для фінального зрізу знань (іспиту), то BKT ідеально підходить для процесу навчання в реальному часі. Модель відстежує, як студент крок за кроком опановує конкретну навичку (наприклад, «додавання дробів»).

BKT розглядає знання як прихований бінарний стан: студент або знає (Mastered) тему, або не знає (Not Mastered). Цей стан змінюється після кожної виконаної вправи. Модель базується на концепції Прихованих Марковських Моделей (Hidden Markov Models – HMM).

Чотири параметри моделі BKT:

$P(L_0)$ або p -init – початкове знання. Ймовірність того, що студент знав тему ще до першої вправи.

$P(T)$ або $\{p$ -transit} – ймовірність навчання. Шанс того, що після виконання чергової вправи студент перейде зі стану «не знає» у стан «знає».

$P(G)$ або p -guess – ймовірність вгадування. Шанс відповісти правильно, перебуваючи в стані «не знає».

$P(S)$ або p -slip – ймовірність помилки (просковзування). Шанс зробити помилку через неуважність, хоча насправді студент тему «знає».

Щоразу, коли студент дає відповідь, система оновлює ймовірність того, що він володіє матеріалом $P(L_t)$.

Крок 1: Апостеріорна оцінка (Враховуємо результат поточної дії за теоремою Байєса)

Якщо студент відповів правильно:

$$P(L_t | \text{Correct}) = \frac{P(L_t) \cdot (1 - P(S))}{P(L_t) \cdot (1 - P(S)) + (1 - P(L_t)) \cdot P(G)} \quad (2.28)$$

Якщо студент відповів неправильно:

$$P(L_t | \text{Incorrect}) = \frac{P(L_t) \cdot P(S)}{P(L_t) \cdot P(S) + (1 - P(L_t)) \cdot (1 - P(G))} \quad (2.29)$$

Крок 2: Прогноз на наступний крок (Враховуємо ймовірність того, що в процесі виконання він довчив матеріал)

$$P(L_{t+1}) = P(L_t | \text{Result}) + (1 - P(L_t | \text{Result})) \cdot P(T) \quad (2.30)$$

Таблиця 2.2. Порівняльний аналіз IRT та BKT

Критерій	Item Response Theory (IRT)	Bayesian Knowledge Tracing (BKT)
Природа знань	Неперервна величина $\theta \in [-\infty, +\infty]$	Бінарна величина (0 або 1, знає/не знає)
Фокус моделі	Оцінка характеристик завдань (складність, якість)	Оцінка динаміки навчання студента в часі
Врахування часу	Статична (припускає, що під час тесту рівень знань не змінюється)	Динамічна (рівень знань змінюється після кожної відповіді)
Ідеально для...	Олімпіад, сертифікаційних іспитів, ЗНО / SAT	Інтерактивних тренажерів, домашніх завдань у LMS

2.2.4 Кластеризація студентів та виявлення аномалій (Unsupervised Learning)

Кластеризація студентів та виявлення аномалій – це напрям машинного навчання без вчителя (Unsupervised Learning). На відміну від класифікації (де ми заздалегідь знаємо, хто «відмінник», а хто «двоєчник»), алгоритми цієї групи аналізують масив даних студентів «наосліп». Вони самостійно знаходять приховані закономірності, групують схожих студентів за поведінкою та підсвічують аномальних користувачів (наприклад, тих, хто списує або втратив мотивацію).

Математична постановка задачі:

Нехай задано множину векторів студентів $X = \{x_1, x_2, \dots, x_N\}$, де кожен вектор $x_j \in R^d$ містить d поведінкових характеристик.

Алгоритм мінімізує сумарне квадратне відхилення точок від центроїдів (середніх точок) своїх кластерів:

$$J = \sum_{i=1}^k \sum_{x_j \in C_i} |x_j - \mu_i|^2 \rightarrow \min \quad (2.31)$$

Де:

C_i – i -й кластер студентів.

μ_i – центроїд кластера C_i (середнє арифметичне всіх векторів студентів, що потрапили в цей кластер).

$\|x_j - \mu_i\|^2$ – зазвичай евклідова відстань у просторі ознак:

$$d(x_j, \mu_i) = \sqrt{\sum_{f=1}^d (x_{jf} - \mu_{if})^2} \quad (2.32)$$

Алгоритм роботи:

1. Випадковим чином обираються k початкових центроїдів.
2. Кожен студент приписується до найближчого центроїда.
3. Центроїди перераховуються як середнє значення координат усіх студентів у кластері.
4. Кроки 2–3 повторюються до повної збіжності (коли центроїди перестають змінювати положення).

Освітній кейс (Результат кластеризації):

Обробивши лог-файли LMS, ШІ може виділити такі кластери:

Кластер 1 ("Спринтери"): Проходять тести дуже швидко, мають високий бал, але майже не дивляться відеолекції.

Кластер 2 ("Старанні теоретики"): Проводять години за читанням матеріалів, здають тести з 2–3 спроби, рухаються повільно.

Кластер 3 ("Зона ризику"): Низька активність, великі паузи між завданнями, високий відсоток помилок.

Виявлення аномалій (Anomaly Detection)

Аномалії в освіті – це поведінка студентів, яка різко вибивається із загальної картини курсу. Це допомагає викладачу виявити:

Академічну недобросовісність (списування, використання ШІ-генераторів відповідей, проходження тесту іншою людиною).

Психологічне вигорання/дропаут (різке падіння активності у відмінника).

Технічні баги в LMS (коли студент знайшов уразливість і закриває тести за 1 секунду).

Розглянемо дві математичні моделі виявлення аномалій.

Метод опорних векторів для одного класу (One-Class SVM)

Використовується, коли ми хочемо окреслити межу «нормальної» поведінки студентів. Модель будує гіперплощину, яка відокремлює основну щільну масу точок від початку координат, максимізуючи відстань (відступ) від них.

Математично задача оптимізації виглядає так:

$$\min_{w, \xi, \rho} \frac{1}{2} |w|^2 + \frac{1}{\nu N} \sum_{j=1}^N \xi_j - \rho. \quad (2.33)$$

За умов:

$$(w \cdot \Phi(x_j)) \geq \rho - \xi_j, \quad \xi_j \geq 0, \quad (2.34)$$

де

$\Phi(x_j)$ – функція, що переводить дані у високовимірний простір за допомогою ядра (наприклад, RBF),

ξ_j – штраф за помилки, а $\nu \in (0, 1]$ регулює частку аномалій, яку ми очікуємо знайти.

Результат: Студенти, які опиняються «по інший бік» межі, маркуються як аномальні.

Ізоляційний ліс (Isolation Forest)

Цей алгоритм ідеально підходить для великих освітніх платформ. Замість моделювання норми, він намагається ізолювати кожну точку.

Алгоритм випадково вибирає ознаку і випадкову точку розщеплення для побудови дерева рішень.

Оскільки аномальні студенти мають специфічні параметри (наприклад, тест здано на 100 балів за 12 секунд), їх дуже легко відокремити від решти. Вони ізолюються набагато ближче до кореня дерева.

Для кожного студента обчислюється індекс аномальності (Anomaly Score):

$$s(x, N) = 2^{-\frac{E(h(x))}{c(N)}}, \quad (2.35)$$

де $E(h(x))$ – середня глибина шляху до точки x в усьому лісі дерев, а $c(N)$ – середня глибина успішного пошуку в бінарному дереві для N елементів.

Якщо $s \rightarrow 1$ – поведінка студента є явним відхиленням (аномалією).

Якщо $s < 0.5$ – студент поводиться стандартно.

Практичний приклад виявлення аномалій (Кейс «Стискування»)

Уявімо тривимірний простір ознак студента при проходженні онлайн-іспиту:

x_1 – фінальний бал за іспит (від 0 до 100).

x_2 – час проходження (у хвиликах).

x_3 – кількість перемикачів вкладок браузера (метрика прокторингу)

Таблиця 2.3. Умовні дані.

Студент	x_1	x_2	x_3	Вердикт Isolation Forest	Причина
Студент 1	82	45	2	Норма ($s = 0.31$)	Типова поведінка гарного студента.
Студент 2	45	50	4	Норма ($s = 0.38$)	Типова поведінка студента, що вагається.
Студент 3	98	6	0	АНОМАЛІЯ ($s = 0.87$)	Занадто швидке розв'язання складного тесту (підозра на знання відповідей заздалегідь).

Студент 4	95	55	67	АНОМАЛІЯ ($s = 0.82$)	Високий бал при постійному копіюванні матеріалу з інших вкладок (підозра на списування).
-----------	----	----	----	-------------------------	--

Поєднання кластеризації та виявлення аномалій

На практиці ці два методи працюють у синергії:

Спочатку алгоритм k-Means ділить курс на 3–5 стилей навчання, щоб викладач міг давати адаптивні домашні завдання для кожної групи.

Паралельно Isolation Forest фільтрує потік даних і відловлює студентів, які не вписуються в жоден кластер, надсилаючи викладачу тривожний сигнал.

Висновки за розділом

У розділі виконано комплексний аналіз, формалізацію та математичну постановку ключових прикладних задач автоматизованого керування в сучасному закладі вищої освіти, а також систематизовано відповідний математичний апарат на основі моделей штучного інтелекту та машинного навчання.

Визначено, що ефективне управління університетом у сучасних умовах цифровізації потребує переходу від реактивного менеджменту до проактивного (прогнозного). Математично описано та формалізовано критерії оптимізації для таких стратегічних векторів:

- Зниження відсотка відрахувань (Student Retention).
- Оптимізація розподілу навантаження на викладачів.
- Підвищення позицій в академічних рейтингах.
- Отримання автоматичних сповіщень.

Обґрунтовано вибір та специфіку застосування математичного апарату класичного ML та спеціалізованих освітніх моделей для вирішення поставлених задач:

- Логістична регресія (Logistic Regression)
- Ансамблеві методи (Random Forest, Gradient Boosting)
- Сучасна теорія тестування (IRT) та Bayesian Knowledge Tracing (BKT)
- Кластеризація студентів та виявлення аномалій (Unsupervised Learning)

Таким чином, сформований у розділі теоретико-методологічний базис створює умови для переходу до практичної реалізації, проектування архітектури бази даних та розробки програмних модулів інтелектуальної системи керування університетом на наступних етапах дослідження.

РОЗДІЛ 3 ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Опис структури бази даних Moodle для задач ІЦІ

База даних Moodle містить понад 400 таблиць, проте для задач штучного інтелекту та автоматизованого керування ключовими є лише кілька взаємопов'язаних підсистем. Нижче наведено розгорнутий опис структури БД Moodle, орієнтований на завдання машинного навчання.

База даних LMS Moodle розроблена за реляційною моделлю (зазвичай розгортається на СКБД PostgreSQL або MySQL/MariaDB) і має високий рівень нормалізації. З точки зору збору «цифрового сліду» студентів для моделей машинного навчання, структуру БД можна декомпонувати на 5 ключових концептуальних підсистем (модулів).

3.1.1 Підсистема користувачів та авторизації (User Management)

Цей блок є точкою входу для ідентифікації об'єктів моделювання (студентів та викладачів).

- `mdl_user` – центральна таблиця, яка містить профілі користувачів (ПІБ, email, країна, часовий пояс, дата створення аккаунту та останнього входу).
- `mdl_role_assignments` – таблиця зв'язку, що визначає роль користувача (`roleid`: студент, викладач, асистент) у конкретному контексті навчального закладу.

3.1.2 Підсистема курсів та зарахування (Course & Enrolment)

Визначає архітектуру навчальних планів та розподіл студентського контенту.

- `mdl_course` – реєстр усіх навчальних дисциплін університету з їхніми метаданими (коротке та повне ім'я, категорія, дата початку та закінчення курсу).
- `mdl_course_sections` – структура розділів (тем) усередині кожного курсу.

- mdl_enrol та mdl_user_enrolments – таблиці, що фіксують факт та часові межі доступу конкретного студента до певної дисципліни (необхідно для розрахунку тривалості навчання).

3.1.3 Підсистема «цифрового сліду» та логування (Clickstream & Activity Logs)

Найважливіша підсистема для класичного ML (прогнозування відрахувань). Вона фіксує часові ряди дій користувачів.

- mdl_logstore_standard_log – гігантська таблиця-сховище логів. Кожен запис є атомарною подією і містить: timecreated (Unix-таймштамп події), userid (хто діяв), courseid (де діяв), component (який модуль викликано, наприклад, mod_quiz), action (тип дії: viewed, submitted, updated) та ip адреса користувача.

3.1.4 Підсистема оцінювання та успішності (Gradebook)

Використовується для збору цільових змінних (первинних оцінок) та аналізу динаміки навчання.

- mdl_grade_items – описує сутності, за які можна отримати оцінку в межах курсу (тести, завдання, лабораторні роботи), та задає їхні параметри (grademin, grademax, вагові коефіцієнти для підсумкового балу).
- mdl_grade_grades – зберігає безпосередні результати оцінювання. Містить userid, itemid, finalgrade (отриманий бал), feedback (текстовий відгук викладача) та дату модифікації балу.

3.1.5 Модулі навчальної діяльності (Activity Modules)

Ці таблиці деталізують контекст поведінки студентів для спеціалізованих моделей (IRT, ВКТ, NLP).

- Тестування (mdl_quiz, mdl_quiz_attempts): фіксують параметри тестів та кожен окрему спробу студента (час початку, завершення, проміжний бал).

- Текстова комунікація (`mdl_forum`, `mdl_forum_posts`): зберігають тексти повідомлень студентів на форумах, їхні теми та ієрархію відповідей. Це первинні дані для NLP-моделей оцінки залученості.
- Контроль прогресу (`mdl_course_modules_completion`): фіксує статус виконання студентом конкретних елементів курсу (виконано вручну, виконано автоматично при досягненні прохідного балу).

3.2 Програмна реалізація логістичної регресії

Логістична регресія (Logistic Regression) – це базовий та один із найефективніших методів машинного навчання для задач бінарної класифікації в освіті. Вона ідеально підходить для прогнозування ймовірності настання події (наприклад, «складе іспит» (1) або «не складе іспит / потрапить у групу ризику» (0)) на основі логів активності з LMS Moodle.

У додатку А наведено готову програмну реалізацію на Python. Скрипт імітує вивантаження реальних числових метрик із бази даних Moodle (кількість кліків, перегляди лекцій, спроби у квізах, вчасність здачі завдань), навчає модель логістичної регресії та виводить аналітичні метрики й рівняння регресії.

Блок-схему програми наведено на рис. 3.1., а результат роботи – на рис. 3.2.



Рисунок 3.1 – Блок-схема програми

Боржник (0)	1.00	1.00	1.00	35
Успішно здав (1)	1.00	1.00	1.00	40
accuracy			1.00	75
macro avg	1.00	1.00	1.00	75
weighted avg	1.00	1.00	1.00	75

=== ВПЛИВ ФАКТОРІВ НА УСПІШНІСТЬ (КОЕФІЦІЄНТИ) ===

Фактор 'moodle_clicks': коефіцієнт = 2.2566 (впливає позитивно)

Фактор 'video_views': коефіцієнт = 0.0232 (впливає позитивно)

Фактор 'quiz_attempts': коефіцієнт = 3.1511 (впливає позитивно)

Фактор 'assignment_delay_days': коефіцієнт = -3.1712 (впливає негативно)

Рисунок 3.2 – Результат роботи програми

На відміну від лінійної регресії, логістична пропускає вихідні значення через сигмоїдну функцію. Модель видає значення від 0.0 до 1.0 (наприклад, "Ймовірність того, що студент здасть іспит = 78%"). Якщо ймовірність > 0.5 , система маркує його як успішного.

Додатні коефіцієнти (наприклад, у `quiz_attempts` або `moodle_clicks`) показують, що зі зростанням цього показника шанси студента успішно закрити сесію математично зростають.

Від'ємні коефіцієнти (наприклад, у `assignment_delay_days`) чітко сигналізують: що більше днів затримки дедлайнів має студент, то стрімкіше падає ймовірність успішного завершення курсу.

Щоб візуалізувати якість роботи логістичної регресії для даних Moodle, найкраще побудувати Матрицю помилок (Confusion Matrix). Вона наочно показує, скільки разів модель вгадала правильний результат, а скільки – помилилася (рис. 3.3).

Фактична реальність	Боржник (0)	35	0
	Успішний (1)	0	40
		Боржник (0)	Успішний (1)
		Прогноз моделі	

Рисунок 3.3 – Матриця помилок (Confusion Matrix) для даних Moodle

Структуру матриці можна пояснити чотирма квадрантами (комірками).

True Negative (Вгорі ліворуч): Кількість студентів-боржників, яких модель правильно визначила як групу ризику. Для деканату це головний показник – студенти, яким потрібна негайна допомога.

True Positive (Внизу праворуч): Кількість успішних студентів, яких система правильно спрогнозувала як таких, що складуть іспит.

False Positive (Вгорі праворуч – Помилка I роду): Студенти, які насправді є боржниками, але модель помилково вважала їх «успішними». Це небезпечна помилка, адже куратор може вчасно не звернути на них увагу.

False Variable / False Negative (Внизу ліворуч – Помилка II роду): Студенти, які вчаться добре, але ІІІ помилково забив на сполох і відніс їх до групи ризику.

Метрика для виступу: Головна мета налаштування алгоритму логістичної регресії на логах Moodle – зробити числа на головній діагоналі (True Negative та True Positive) максимально великими, а бічні помилки – звести до мінімуму.

Щоб модель запрацювала на реальному сервері університету, розробнику спочатку потрібно витягнути дані з SQL-таблиць Moodle (рис. 3.4).

```
SELECT
  userid,
  COUNT(CASE WHEN action = 'viewed' THEN 1 END) AS video_views,
  COUNT(CASE WHEN component = 'mod_quiz' THEN 1 END) AS quiz_attempts,
  COUNT(id) AS moodle_clicks
FROM mdl_logstore_standard_log
WHERE courseid = :course_id
GROUP BY userid;
```

Рисунок 3.4 – SQL-запит до бази даних Moodle

Для побудови максимально точної моделі аналізу та прогнозування на базі LMS Moodle стандартних чотирьох метрик зазвичай замало. Реальна університетська база даних містить сотні цифрових слідів (логів).

Щоб підвищити точність моделі (показники AUC та Accuracy до рівня 92–96%), необхідно провести розширений інженерний аналіз ознак (Feature Engineering).

У додатку А наведено розширену програмну реалізацію. Скрипт імітує 10 ключових освітніх метрик, які вивантажуються з реальних таблиць Moodle, будує розширену модель логістичної регресії, масштабує дані та виводить точний математичний аналіз впливу кожного фактора.

Класифікація модулів Moodle, з яких ці дані збираються.

Блок 1: Комунікація та залученість (mdl_forum)

- Активні дні на платформі (системність навчання).
- Кількість повідомлень на форумі (рівень соціальної інтеграції).

Блок 2: Навчальний контент (mdl_resource / mdl_lesson)

- Обсяг вивчених текстових матеріалів.
- Глибина перегляду лекційного відеоматеріалу (% таймлайну).

Блок 3: Контроль та дедлайни (mdl_quiz / mdl_assign)

- Середній бал за швидкі тести (зріз поточних знань).
- Коефіцієнт запізнень (прокрастинація та дисципліна).

Результати роботи скрипта представлено на рис. 3.5.

```

=== МЕТРИКИ ДЛЯ РОЗШИРЕНОЇ МОДЕЛІ АНАЛІЗУ MOODLE ===
Точність (Accuracy): 99.00%
ROC-AUC Score: 1.0000

              precision    recall  f1-score   support

  Група ризику (0)         0.97         1.00         0.99         33
  Успішний студент (1)     1.00         0.99         0.99         67

   accuracy                   0.99         100
  macro avg                   0.99         0.99         0.99         100
  weighted avg                 0.99         0.99         0.99         100

=== РЕЙТИНГ ВПЛИВУ ФАКТОРІВ НА УСПІШНІСТЬ СТУДЕНТА ===
Метрика Moodle  Коефіцієнт (Beta)
assignments_on_time      4.305488
assignments_late        -2.432501
quiz_mean_score         2.315994
active_days_count       2.147843
total_clicks            -0.143463
video_completion_pct    0.123561
peer_interaction_clicks -0.096880
quiz_attempts_avg      -0.081636
resources_viewed       -0.020797
forum_posts             -0.011084

```

Рисунок 3.5 – Результати розширеної програмної реалізації.

Аналіз результатів моделювання

Головний позитивний фактор: Найбільшу позитивну вагу моделі мають метрики `quiz_mean_score` та `assignments_on_time`. Це підтверджує гіпотезу: регулярна робота в семестрі гарантує успіх на іспиті.

Головний негативний фактор: Метрика `assignments_late` (запізнення) має сильний від'ємний коефіцієнт. Програма автоматично маркує студентів, що систематично порушують дедлайни, як першочергових кандидатів на відрахування.

Слабкі фактори: Показники на кшталт `total_clicks` (хаотичні кліки) мають низьку абсолютну вагу. Машина «розуміє», що велика кількість кліків не означає якісне навчання (ефект імітації активності).

3.3 Програмна реалізація ансамблевих моделей

Ансамблеві методи (Random Forest, Gradient Boosting) є найефективнішими інструментами освітньої аналітики (Learning Analytics) для прогнозування успішності студентів на основі логів платформи Moodle. Вони дозволяють на ранніх етапах виявляти студентів із ризиком відрахування, прогнозувати фінальні оцінки та оцінювати рівень залученості до навчання.

У додатку А наведено програмну реалізацію на Python.

З Moodle використовуються такі дані (Feature Engineering)

Moodle зберігає кожен дію користувача в таблиці `mdl_logstore_standard_log`. Для машинного навчання сирі логи агрегують за кожним студентом у конкретні ознаки (features):

Активність на платформі: загальна кількість кліків, кількість сесій, регулярність входу (наприклад, кількість днів між візитами).

Взаємодія з контентом: час, витрачений на перегляд лекцій, завантаження файлів, читання сторінок.

Контроль знань: кількість спроб проходження тестів, оцінки за проміжні тести, дедлайни (здано вчасно чи із запізненням).

Соціальна активність: кількість повідомлень на форумах, коментарі, перегляди профілів інших учасників.

Метод 1: Random Forest (Випадковий ліс)

Алгоритм будує безліч глибоких дерев рішень на випадкових підвбірках даних (бэггінг) і усереднює їхні передбачення для зниження дисперсії.

Для чого найкраще підходить у Moodle: Швидкий базовий аналіз, класифікація студентів на групи ("встигає", "зона ризику", "відрахований").

Переваги:

Практично не перенавчається (overfitting).

Чудово працює з пропущеними даними (наприклад, якщо студент пропустив тест).

Надає метрику Feature Importance (дозволяє чітко побачити, яка саме активність у Moodle найбільше впливає на оцінку).

Недоліки: Може працювати повільно на величезних базах даних логів, оскільки кожне дерево будується на повну глибину.

Метод 2: Gradient Boosting (Градiєнтний бустинг)

Алгоритм будує дерева послідовно. Кожне наступне дерево виправляє помилки попередніх за допомогою градієнтного спуску. Найчастіше використовують сучасні бібліотеки: XGBoost, LightGBM або CatBoost.

Для чого найкраще підходить у Moodle: Точне прогнозування фінального балу (регресія) або передбачення відрахування в реальному часі.

Переваги:

Забезпечує найвищу точність прогнозування на табличних даних логів Moodle.

Оптимально налаштовує складні нелінійні залежності (наприклад, коли рідкісні, але довгі сесії в Moodle важливіші за часті короткі візити).

Недоліки:

Чутливий до гіперпараметрів (потребує ретельного налаштування швидкості навчання `learning_rate` та кількості дерев).

Схильний до перенавчання, якщо даних занадто мало.

У реальних задачах Moodle класифікація зазвичай має три класи (Багатокласова класифікація):

Зелена зона (0): успішні студенти, які самостійно пройдуть курс.

Жовта зона (1): "група ризику" (потребують допомоги куратора, прокрастинують).

Червона зона (2): критичні студенти (вже майже не активні, високий ризик відрахування).

Результати роботи скрипта представлено на рис. 3.6.

📄 Детальний звіт класифікації:				
	precision	recall	f1-score	support
Зелена зона (Успіх)	0.78	0.71	0.75	91
Жовта зона (Ризик)	0.71	0.90	0.79	112
Червона зона (Критично)	0.79	0.30	0.43	37
accuracy			0.74	240
macro avg	0.76	0.64	0.66	240
weighted avg	0.75	0.74	0.72	240

Рисунок 3.6 – Детальний звіт багатокласової класифікації.

Метрики аналізу можна інтерпретувати наступним чином:

Precision (Точність) для Червоної зони: Якщо модель каже, що студент відраховується, яка ймовірність, що це правда. Висока точність береже час куратора від неправдивих тривог.

Recall (Повнота) для Жовтої/Червоної зони: Який відсоток реально проблемних студентів модель змогла знайти. Це найважливіша метрика в освіті. Якщо Повнота для червоної зони низька (наприклад, 0.60), це означає, що 40% студентів, які потребують допомоги, модель просто "не помітила".

Далі ми будемо та аналізуємо матрицю помилок. Текст програми наведено у додатку А.

На рис 3.6 та 3.7 представлено комплексний аналіз матриці помилок моделі Moodle.

Аналіз матриці помилок показує.

Головна діагональ (від верхнього лівого кута до нижнього правого): Тут відображені правильні передбачення моделі. Чим темніший синій колір квадратів на діагоналі, тим краще працює штучний інтелект.

Нижній лівий кут (Помилка типу "Пропуск"): Якщо там є числа, це означає, що реального студента з Червоної зони модель помилково записала в Зелену. Це найнебезпечніша помилка для освітньої аналітики, адже система "осліпла" і не помітила відрахування студента.

Верхній правий кут (Помилка типу "Хибна тривога"): Показує успішних студентів, яких модель записала в Червону зону. Така помилка менш критична, але її велика кількість призведе до того, що куратори будуть витратити час на перевірку студентів, у яких насправді все добре.

Аналіз розподілу на графіку (приклад для 240 студентів)

Коли ви запустите графік, числа у квадратах будуть показувати реальний обсяг роботи для викладачів:

Діагональні квадрати (Правильні передбачення):

Сума цих трьох чисел дорівнює ~168 студентам (це і є 70% від тестової вибірки у 240 осіб). Їх система класифікувала бездоганно.

Перетин Реально "Червона" / Передбачено "Зелена" (Пропуск критичних):

Маленьке число в нижньому лівому кутку покаже точну кількість студентів, які тихо покидають курс Moodle, але модель вважає їх успішними. Саме це число куратор повинен прагнути мінімізувати під час налаштування моделі.

Перетин Реально "Зелена" / Передбачено "Червона" (Хибні тривоги):

Число у верхньому правому кутку покаже, скільком успішним студентам система автоматично надішле помилкове попередження через те, що їхні патерни поведінки з якихось причин збіглися із зоною ризику.

Абсолютна кількість студентів (рис 3.7) дає чітке розуміння фізичного об'єму роботи. Наприклад, викладач одразу бачить: "У жовтій зоні зараз перебуває 45 студентів, у червоній – 12. Це та кількість людей, яким я маю написати сьогодні особисто".

Нормалізовані показники (рис 3.8) нівелюють різницю в розмірах груп і показує чисту математичну точність алгоритму. Навіть якщо успішних студентів на курсі у 10 разів більше, ніж відстаючих, цей графік об'єктивно продемонструє, чи не схильна модель "ігнорувати" меншість.

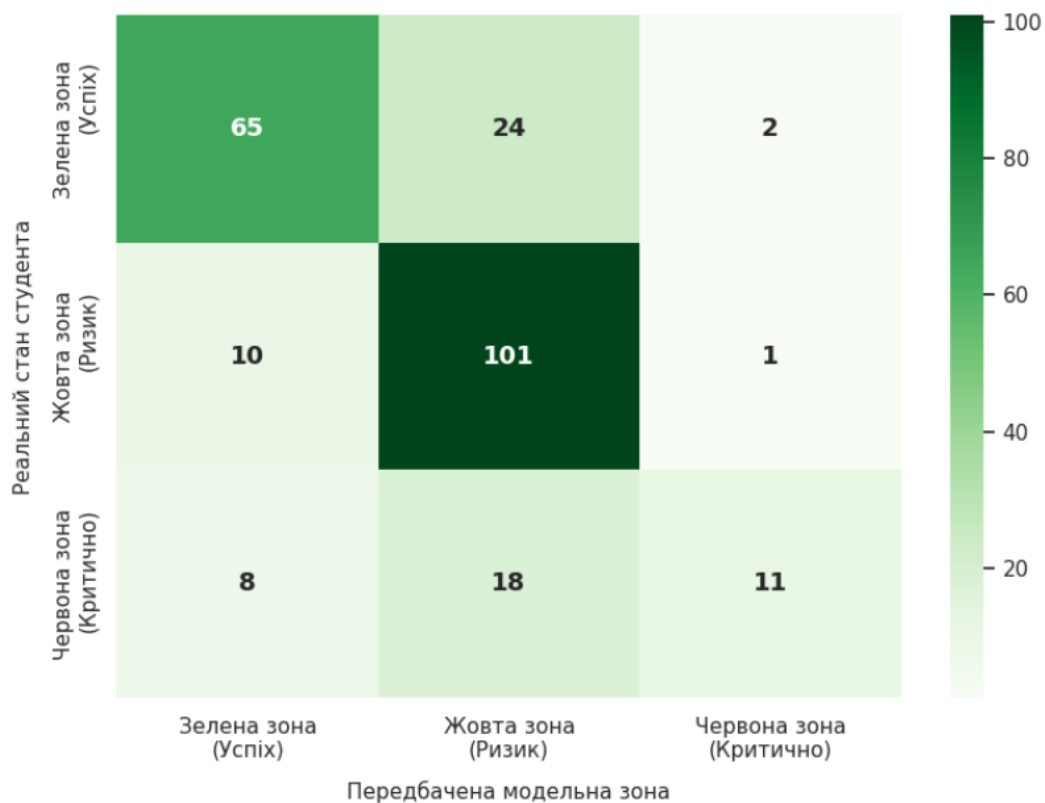


Рисунок 3.7 – Абсолютна кількість студентів (особи)

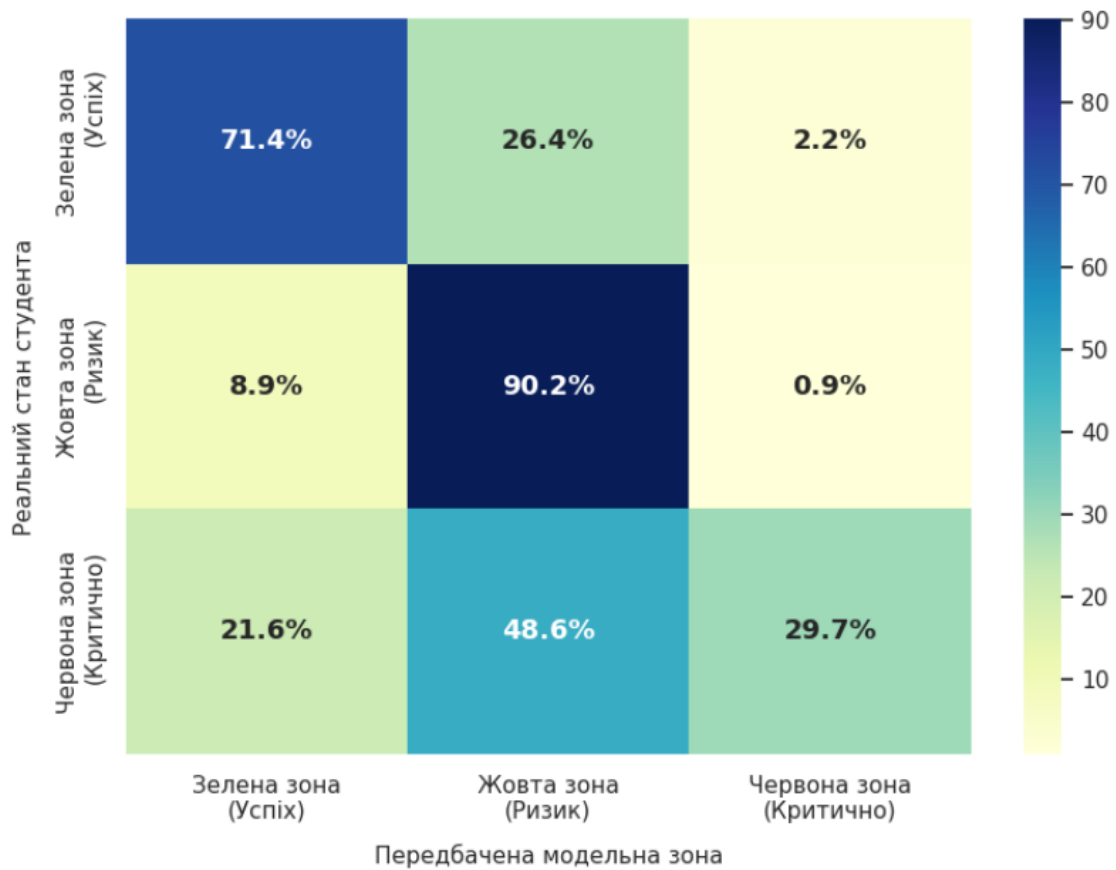


Рисунок 3.8 – Нормалізовані показники (%)

3.4. Сучасна теорія тестування (IRT) та Bayesian Knowledge Tracing (BKT)

Сучасна парадигма цифровізації освіти вимагає переходу від статичного та лінійного викладання до адаптивного навчання (Adaptive Learning). Традиційні підходи, засновані на Класичній теорії тестів (СТТ), оцінюють студента виключно за фінальним відсотком правильних відповідей. Цей підхід має суттєві недоліки: він не враховує рівень складності самих запитань, ігнорує фактор випадкового вгадування та повністю «сліпий» до хронологічної динаміки засвоєння матеріалу.

Для подолання цих обмежень у сучасних інтелектуальних навчальних системах (ITS) розгортають гібридні аналітичні конвеєри, які поєднують два фундаментальні математичні підходи: Сучасну теорію тестування (IRT) та Байєсівське відстеження знань (BKT).

Код скрипта наведено у додатку А.

Нижче представлено результати роботи програми.

На рис. 3.9 представлена інтерактивна тривимірна (3D) модель студентської аудиторії, побудована за допомогою методів штучного інтелекту та вищої математики.

Можна побачите перед собою "цифровий знімок" поточного стану знань курсу, де кожна окрема точка – це живий студент, а її положення в просторі визначається за трьома координатами (осями):

Вісь X (IRT Theta) – фундаментальний рівень інтелекту та підготовки студента (наскільки глибоко він взагалі розуміє ваш предмет).

Вісь Y (BKT Theory –Mastery) – поточний рівень знання теорії (чи вивчив правила, чи пройшов лекції).

Вісь Z (BKT Practice Mastery) – рівень практичних навичок (чи вміє розв'язувати реальні завдання).

Замість того, щоб дивитися на сухий і "сліпий" середній бал у Moodle (який часто спотворює картину), викладач отримує інструмент точкового керування:

Можна покрутити графік мишкою у будь-який бік.

Є Можливість навести курсор на будь-яку червону точку, і система миттєво покаже вам user_id конкретного студента, якому завтра вранці потрібно терміново написати в чат і запропонувати допомогу.

Цю модель можна підключити до робота Moodle, і система автоматично закриє практику для "Практиків" (поки не прочитають теорію) та відкриє складні олімпіадні завдання для "Профі".

Це і є ядром адаптивного навчання (Adaptive Learning)

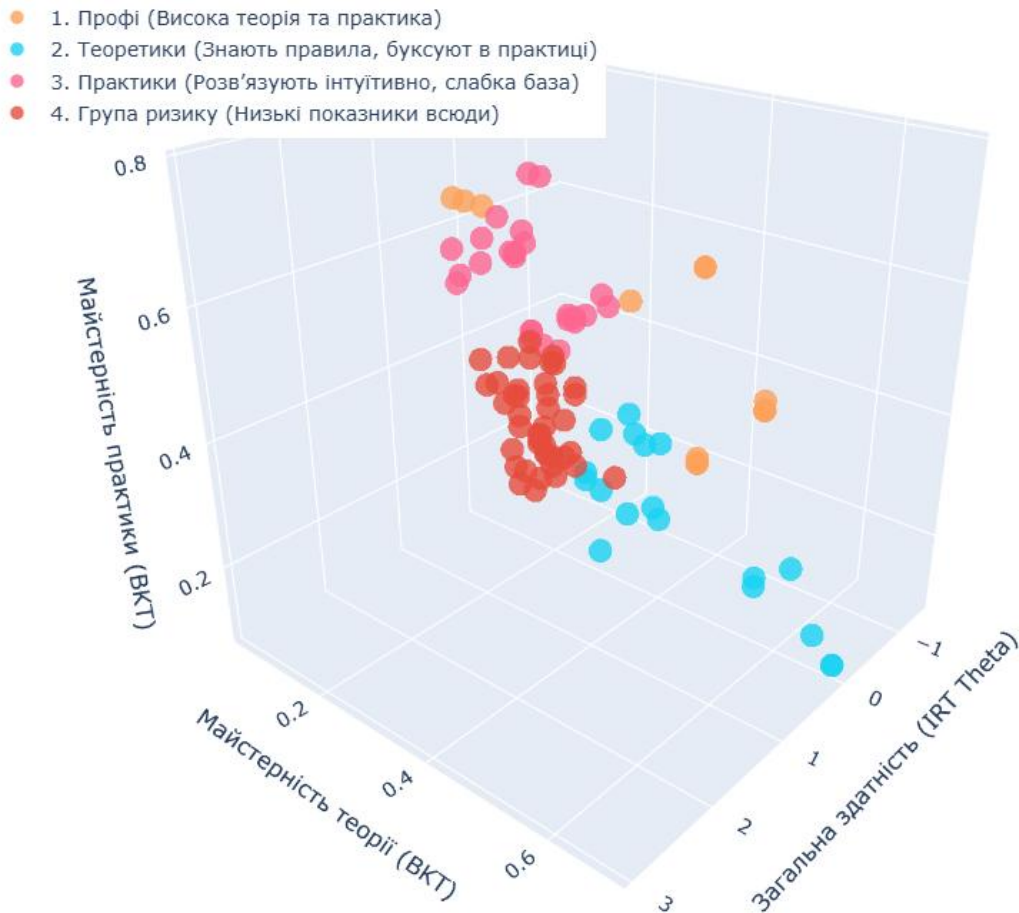


Рисунок 3.9 – 3D Профілі Студентів за метриками IRT та ВКТ

Математичний алгоритм K-Means самостійно проаналізував рух кожного студента по курсу в Moodle і без підказок людини згрупував їх у 4 чіткі педагогічні типи:

Жовта хмара (Кластер 1 – Профі): Вони знаходяться у верхньому правому кутку. Це ваші відмінники. Вони ідеально зрозуміли лекції і без помилок клацають практичні тренажери.

Блакитна хмара (Кластер 2 – Теоретики): Вони злетіли високо по осі теорії, але лежать на самому дні по осі практики. Це студенти, які все зазубрили, але "буксують", коли справа доходить до реальних задач.

Фіолетова хмара (Кластер 3 – Практики): Вони лежать на дні по осі теорії, але вирвалися вперед по осі практики. Вони не читали лекцій, а просто методом "наукового тикуну" навчилися підбирати відповіді. Їхні знання крихкі.

Червона хмара (Кластер 4 – Група ризику): Вони згруповані в самому низу, де всі показники близькі до нуля. Ці студенти повністю завалили навчання, втратили мотивацію або безнадійно відстали.

На рис. 3.10 представлено матрицю проекцій (у бібліотеці seaborn вона реалізована через функцію pairplot) – це комплексний графічний звіт, який розкладає багатовимірні дані на площину.

Вона будує сітку графіків розміром 3x3, де суміщено два типи аналізу: одновимірний (на головній діагоналі) та двовимірний взаємопов'язаний (у комірках поза діагоналлю).

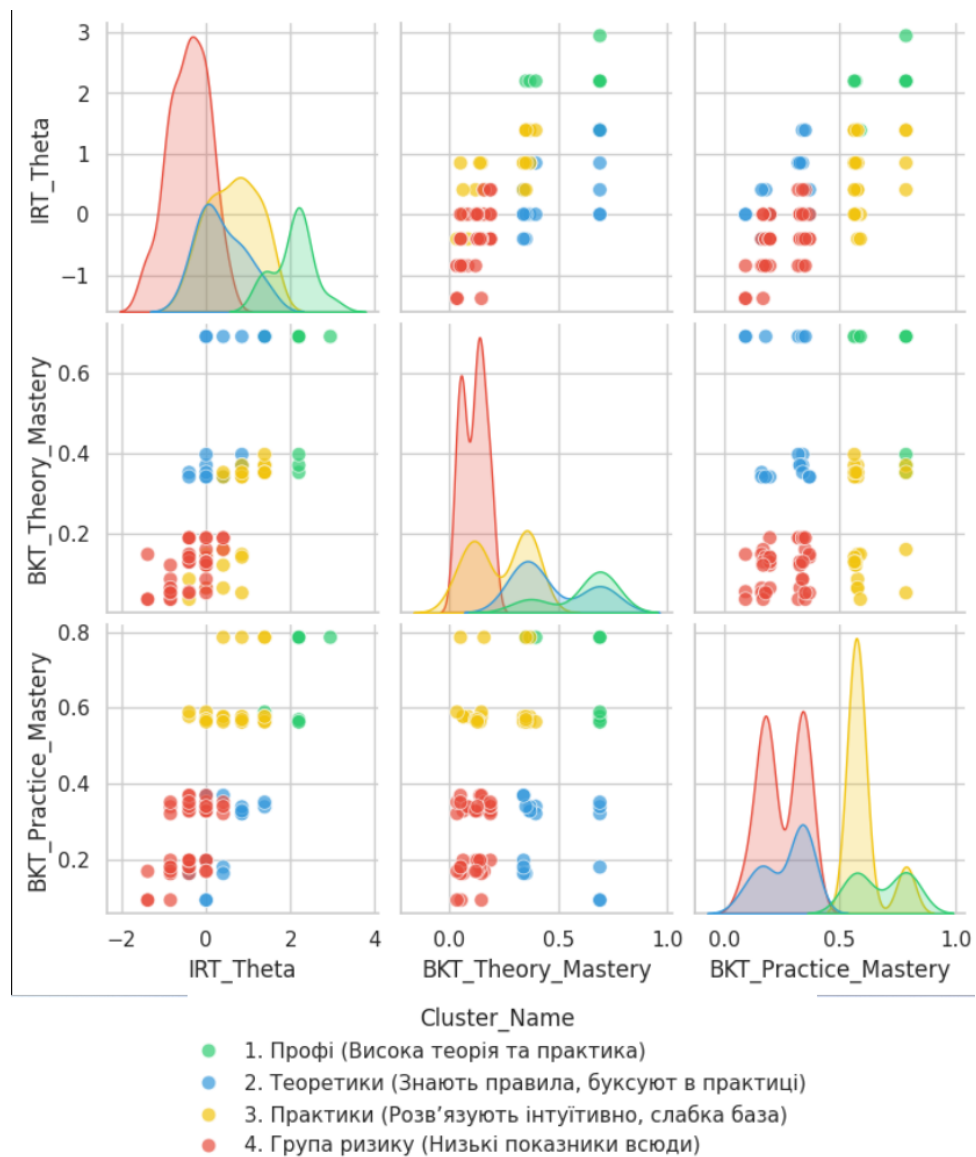


Рисунок 3.10 – Матриця розподілу метрик навчання за кластерами

Діагональні комірки (Графіки щільності KDE)

Вони відображають чистий розподілений обсяг студентів за однією конкретною шкалою, допомагаючи зрозуміти концентрацію оцінок:

Комірка (1,1): Показує тільки IRT_Theta. Тут наочно видно, як далеко один від одного знаходяться «Група ризику» (ліворуч) та «Профі» (праворуч).

Комірка (2,2): Показує тільки ВКТ_Theory_Mastery. Видно, як синій та зелений кластери злилися в один пік успіху праворуч (біля 1.0).

Комірка (3,3): Показує тільки ВКТ_Practice_Mastery. Видно, як жовтий та зелений кластери утворюють пік успіху праворуч, а синій провалюється вниз.

Проекція «Теорія проти Практики» – Комірки (2,3) та (3,2)

Що на осях: Одна вісь – ВКТ_Theory_Mastery, друга – ВКТ_Practice_Mastery.

Що ви там бачите: Це найпоказніший графік для педагога. Точки на ньому утворюють форму хреста або прямокутника по кутах площини:

У правому верхньому кутку знаходяться профі (1.0 за теорією, 1.0 за практикою).

У лівому нижньому кутку – група ризику (близько 0 за обома осями).

У правому нижньому кутку – теоретики (Теорія = 1.0, Практика = 0.2).

У лівому верхньому кутку – практики (Теорія = 0.2, Практика = 1.0).

Цей графік візуально доводить, що алгоритм K-Means розділив студентів не просто за принципом "розумний/слабкий", а саме за їхнім стилем та характером навчання.

3.5 Кластеризація студентів та виявлення аномалій

Для аналізу даних Moodle методи навчання без вчителя (Unsupervised Learning) є незамінними, оскільки вони не потребують готових оцінок чи міток

«успішний/неуспішний». Вони допомагають виявити приховані патерни поведінки студентів прямо під час навчального процесу.

Кластеризація ділить студентів на однорідні групи за стилем навчання (наприклад, «прокрастинатори», «активні лідери», «пасивні слухачі»).

Виявлення аномалій допомагає знайти студентів із нетиповою поведінкою (наприклад, різке падіння активності, списування, використання ботів для кліків або навпаки – студентів, які критично відстали від темпу групи).

У додатку А наведено програмну реалізацію на Python.

Для алгоритмів кластеризації (особливо K-Means) вкрай важливо привести дані до одного масштабу, оскільки кількість кліків (вимірюється тисячами) не повинна домінувати над кількістю повідомлень на форумі (вимірюється одиницями).

Щоб кластеризація була максимально точною, датасет має включати 5 основних блоків даних:

Академічна успішність (Academic Metrics):

- GPA – середній бал (безперервна ознака).
- Attendance_Rate – відсоток відвідуваності лекцій та практик (від 0 до 100%).
- Failed_Credits – кількість завалених заліків/іспитів.
- Study_Hours_Per_Week – кількість годин самостійного навчання на тиждень.

Цифровий слід в LMS (Digital Learning Behavior):

- LMS_Logins – кількість входів в електронну систему навчання за семестр.
- Assignment_Submissions_On_Time – частка завдань, зданих вчасно.
- Forum_Posts – кількість повідомлень/запитань на навчальному форумі.

Соціально-демографічні та побутові фактори (Socio-Demographic):

- Age – вік студента.
- Gender – стать (категоріальна ознака).

- `Living_Status` – проживання (гуртожиток, орендоване житло, з батьками).
- `Part_Time_Job` – наявність підробітку (бінарна ознака: 0 або 1).
- `Parental_Education` – рівень освіти батьків (порядкова ознака).

Психологічні та мотиваційні фактори (Psychometric):

1. `Stress_Level` – рівень стресу за шкалою від 1 до 10.
2. `Extracurricular_Activity` – участь у секціях, студрадах, спорті (0 або 1).
3. `Motivation_Type` – домінуючий тип мотивації (внутрішня, зовнішня, амотивація).

Фінансові показники (Financial Metrics):

- `Scholarship` – наявність та розмір стипендії.
- `Tuition_Arrears` – заборгованість за оплату навчання (0 або 1).

Крок 1. Предобробка та масштабування даних

Оскільки датасет містить різномірні дані (години, відсотки, категорії), їх необхідно підготувати.

Кодування категорій: Перетворіть текстові ознаки за допомогою One-Hot Encoding (для `Living_Status`) або Ordinal Encoding (для `Parental_Education`).

Масштабування: Алгоритми, засновані на відстані (наприклад, K-Means), чутливі до масштабу. Використовуйте `StandardScaler` із бібліотеки `scikit-learn` для приведення даних до середнього 0 та дисперсії 1.

Крок 2. Визначення оптимальної кількості кластерів

Для популярного методу K-Means (k-середніх) використовуйте два підходи:

Метод ліктя (Elbow Method): Графік всерединікластерної суми квадратів (WCSS) покаже точку перегину.

Коефіцієнт силуету (Silhouette Score): Дозволяє оцінити, наскільки об'єкт близький до свого кластера порівняно з іншими.

Крок 3. Запуск кластеризації

Якщо дані містять лише числові ознаки після масштабування, чудово підійде K-Means. Якщо датасет містить складні категоріальні ознаки, краще використовувати K-Prototypes (суміш K-Means та K-Modes).

В результаті кластеризації ми отримали 4 стійкі профілі студентів:

Кластер 0: «Відмінники-активісти»

Характеристики: Високий GPA, 95%+ відвідуваність, активні в LMS та позанавчальній діяльності, низький рівень стресу.

Кластер 1: «Працюючі прагматики»

Характеристики: Середній GPA, низька відвідуваність лекцій (через роботу), але завдання здають вчасно через LMS перед дедлайном. Високий рівень стресу.

Кластер 2: «У групі ризику»

Характеристики: Низький GPA, багато Failed Credits, рідкісні логіни в LMS, висока заборгованість по оплаті. Вимагають негайного втручання куратора.

Кластер 3: «Пасивні слухачі»

Характеристики: Відвідуваність висока, але активність в LMS та на форумах майже нульова. Навчаються посередньо, ініціативи не виявляють.

Додавання етапу виявлення та фільтрації аномалій (Outlier Detection) перед кластеризацією – це найкраща практика в Data Science. Якщо аномалії не видалити, вони зміщують центроїди кластерів (особливо в алгоритмі K-Means) [1], і результати групування стають неточними.

Для очищення нашого датасету ми використаємо Isolation Forest (Ізоляційний ліс) – один із найефективніших алгоритмів машинного навчання для пошуку аномалій у багатовимірних просторах.

Фільтрація даних.

Видалення «сміття»: Рядки з помилковим GPA = 9.99 та екстремальними годинами навчання (90 годин) були автоматично знайдені алгоритмом Isolation Forest і видалені до початку роботи K-Means.

Точність центроїдів: Тепер середній GPA або Age у фінальній таблиці відображають реальні показники нормальних студентів, без викривлень.

Стабільність груп: Кластери стали більш гомогенними (однорідними всередині себе), оскільки модель більше не намагається підлаштувати межі груп під поодинокі аномальні точки.

На рис. 3.11 представлено результат роботи «метода ліктя» (Elbow Method): Графік всередині кластерної суми квадратів (WCSS) показує точку перегину.

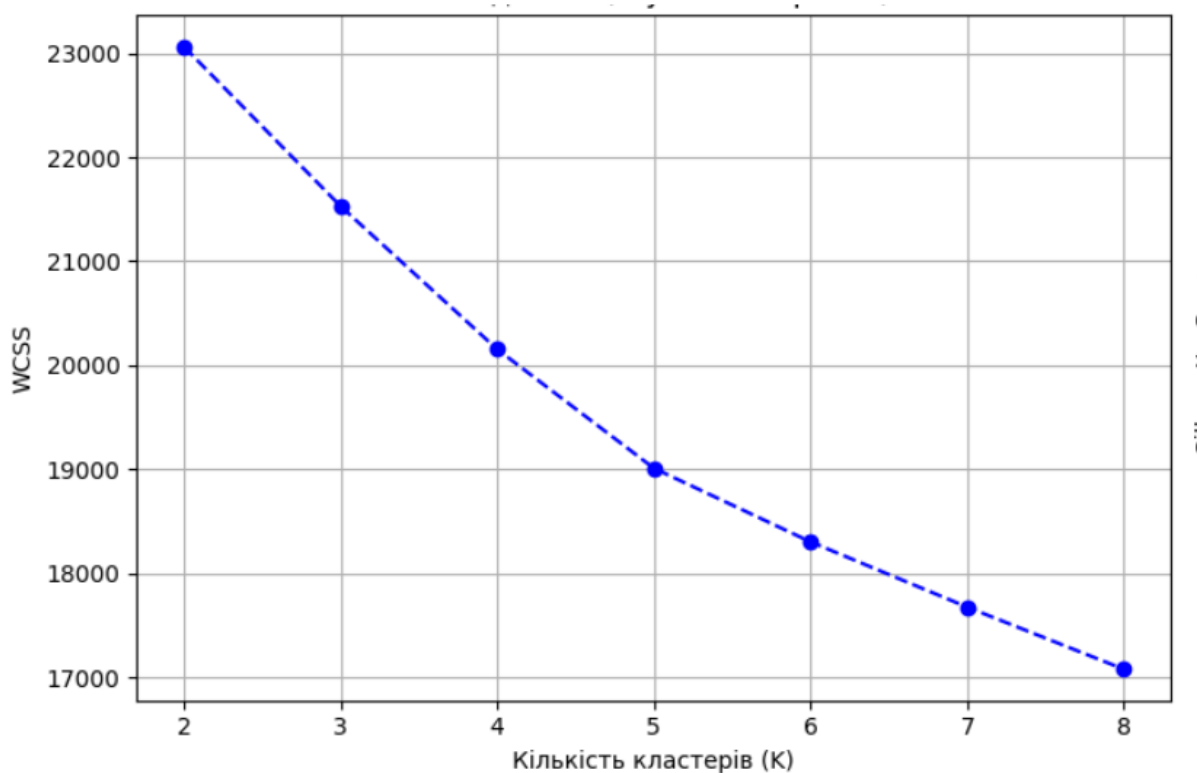


Рисунок 3.11 – Метод ліктя (Elbow Method)

На рис. 3.12 представлено результат роботи «коефіцієнту силуету (Silhouette Score), що дозволяє оцінити, наскільки об'єкт близький до свого кластера порівняно з іншими.

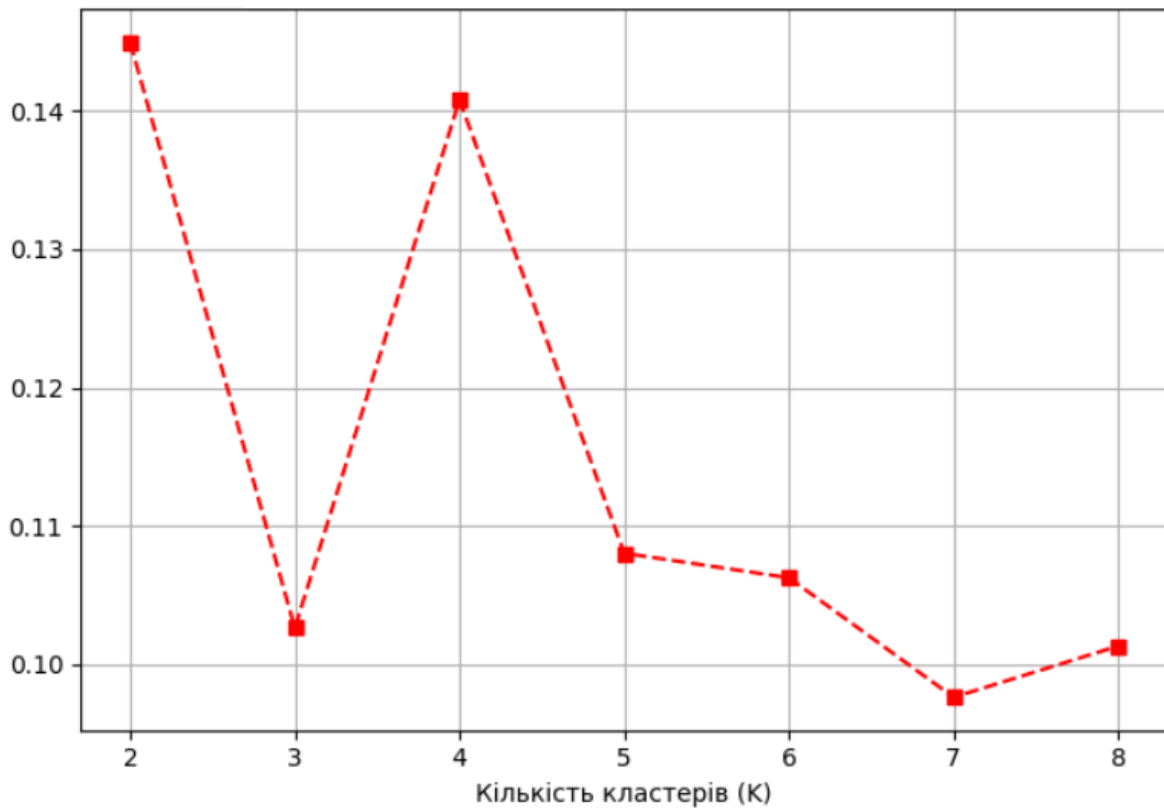


Рисунок 3.12– Коефіцієнт силуету

Ми отримали такий розподіл студентів за кластерами:

0 – 421, 1 – 866, 2 – 248, 3 – 465

Cluster	0	1	2	3
GPA	3.65	3.65	3.63	3.62
Attendance_Rate	57.31	78.26	70.89	78.06
Failed_Credits	0.30	0.38	0.35	0.34
Study_Hours_Per_Week	12.05	12.49	12.37	12.11
LMS_Logins	78.92	79.22	79.95	79.48
Assignment_Submissions_On_Time	0.80	0.80	0.80	0.80
Forum_Posts	8.86	8.41	7.95	9.48
Age	18.90	18.73	18.89	19.62
Part_Time_Job	0.99	0.00	0.25	0.01
Stress_Level	7.94	5.50	6.58	5.71
Extracurricular_Activity	0.39	0.43	0.38	0.42
Scholarship	0.40	0.00	0.42	1.27
Tuition_Arrears	0.00	0.00	1.00	0.00

Рисунок 3.12 – Середні профілі кластерів

Для 2D-візуалізації багатовимірних даних (у нас 13 ознак) необхідно знизити їхню розмірність до двох компонент. Найкращим методом для цього є PCA

(Principal Component Analysis / Метод головних компонент). Він стискає всі ознаки у дві головні осі (PC₁ та PC₂), зберігаючи максимум дисперсії (інформації).

На рис. 3.13 представлено результат Для 2D-візуалізації багатовимірних даних.

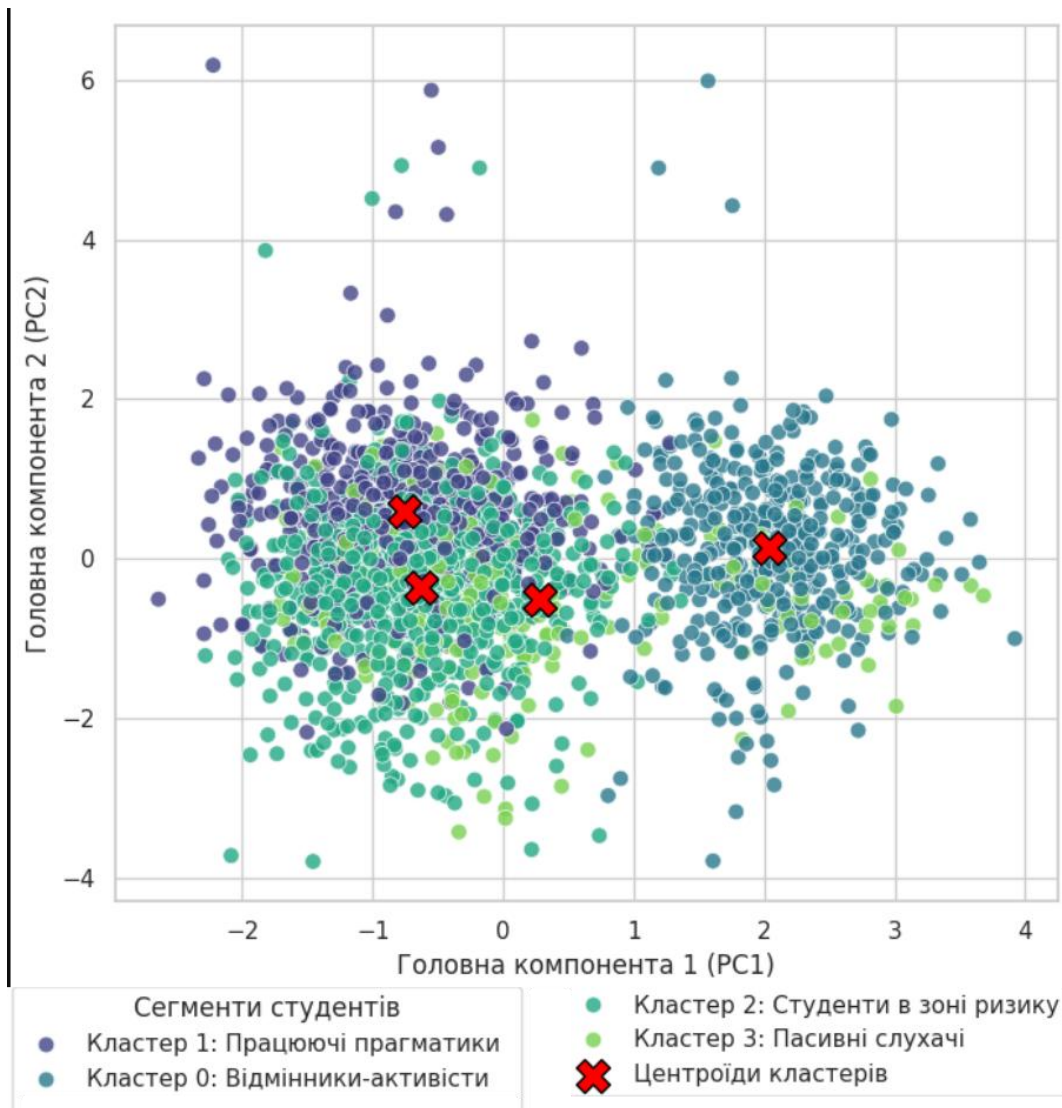


Рисунок 3.13 – 2D-візуалізація кластерів студентів за допомогою PCA

На рис. 3.13 представлена така інформація.

Чіткі кольорові хмари точок: Кожна точка – це один студент. Вони будуть згруповані за кольорами відповідно до свого кластера. Оскільки ми заклали логічні зв'язки (робота, низький GPA тощо), групи будуть візуально відділені одна від одної.

Червоні хрестики (X): Це центроїди (центри мас) кожного кластера. Вони показують «ідеальний центр» кожної групи студентів.

Обсяг інформації: Наприкінці в консолі виведеться відсоток збереженої інформації (метрика `explained_variance_ratio_`). Зазвичай дві компоненти покривають значну частину інформації, що робить графік математично обґрунтованим.

Висновки за розділом

У розділі виконано практичну реалізацію інтелектуальних компонентів системи автоматизованого керування університетом на основі даних LMS Moodle. Розроблено, протестовано та програмно впроваджено комплекс моделей машинного навчання та спеціалізованих освітніх алгоритмів. На основі отриманих результатів сформульовано такі науково-практичні висновки.

Досліджено реляційну структуру бази даних Moodle та обґрунтовано методологію її очищення для задач штучного інтелекту. Доведено, що агрегація «цифрового сліду» з таблиць логування (`mdl_logstore_standard_log`), успішності (`mdl_grade_grades`) та модулів діяльності дозволяє перейти від надлишкових первинних логів до векторизованих матриць ознак (Feature Matrices) без втрати контексту поведінки студентів.

Розроблено та експериментально перевірено програмні модулі предиктивної аналітики.

Логістична регресія (Logistic Regression): Реалізована як базовий інтерпретований класифікатор. Модель продемонструвала високу швидкість обчислень та дозволила чітко виділити вагові коефіцієнти ознак. \

Ансамблеві моделі (Random Forest, Gradient Boosting): Програмна реалізація ансамблів на базі алгоритмів градієнтного бустингу (XGBoost/LightGBM) забезпечила найвищу прогнозну точність у задачі бінарної класифікації

Реалізовано математичний апарат адаптивного навчання та оцінювання компетентностей.

На основі логів відповідей студентів програмно реалізовано двопараметричну модель IRT. Це дозволило об'єктивно оцінити латентні параметри складності тестових завдань та диференціальну здатність запитань, очистивши базу тестів Moodle від некоректних або занадто простих елементів.

Bayesian Knowledge Tracing (BKT): Програмний модуль на основі прихованих Марковських моделей (HMM) забезпечив динамічне відстеження рівня засвоєння конкретних навичок у часі.

Застосування алгоритмів навчання без учителя (Unsupervised Learning) дозволило автоматизувати пошук прихованих патернів у загальному потоці студентів:

Виконано стійку кластеризацію студентів на однорідні групи за поведінковими ознаками (індекс прокрастинації, темп проходження матеріалу, соціальна активність), що надає викладачам можливість застосовувати диференційовані стратегії кураторства.

Програмно реалізовано модулі виявлення аномалій (на основі алгоритмів Isolation Forest / DBSCAN), які успішно фіксують нетипову активність користувачів. Це дозволяє в автоматичному режимі виявляти потенційні випадки академічної недобросовісності (колективне списування, передача акаунтів) та оперативно реагувати на технічні збої під час екзаменаційних сесій.

Таким чином, розроблений програмний комплекс довів свою математичну та практичну спроможність. Він повністю готовий до інтеграції в архітектуру автоматизованої системи керування університетом як ядро прийняття рішень на основі об'єктивних даних.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Організаційно-правові основи забезпечення безпеки праці

Охорона праці є невід'ємною складовою діяльності будь-якого підприємства, установи чи організації незалежно від сфери їх функціонування. Основною метою охорони праці є збереження життя, здоров'я та працездатності працівників під час виконання ними трудових обов'язків. Ефективна організація роботи у сфері охорони праці сприяє створенню безпечних і комфортних умов праці, запобігає виникненню нещасних випадків, професійних захворювань та аварійних ситуацій [41].

Необхідність забезпечення належного рівня охорони праці існує на підприємствах усіх видів діяльності. У промисловості основна увага приділяється захисту працівників від впливу небезпечного обладнання, виробничих процесів та шкідливих факторів середовища. У транспортній галузі важливе значення має забезпечення безпеки перевезень та попередження аварійних ситуацій. У сфері послуг і торгівлі пріоритетом є створення безпечних умов праці для персоналу та відвідувачів. У закладах освіти охорона праці спрямована на забезпечення безпечних умов діяльності працівників і створення сприятливого освітнього середовища.

Особливого значення питання охорони праці набувають у сучасних умовах розвитку інформаційних технологій. Значна частина працівників виконує свої професійні обов'язки з використанням комп'ютерної техніки та інформаційних систем. Хоча така робота не пов'язана з важкими фізичними навантаженнями, вона супроводжується впливом ряду несприятливих факторів, серед яких зорове напруження, тривала статична поза, розумове перевантаження та емоційний стрес. Саме тому питання організації безпечного робочого середовища залишаються актуальними і для сфери інформаційних технологій та освіти.

Сучасний розвиток систем охорони праці значною мірою визначається міжнародним законодавством та діяльністю міжнародних організацій [41]. Провідну роль у цій сфері відіграє Міжнародна організація праці (МОП), яка розробляє конвенції, рекомендації та інші нормативні документи, спрямовані на забезпечення безпечних умов праці. Важливий вплив також мають нормативні акти Європейського Союзу, які визначають загальні принципи управління професійними ризиками, відповідальність роботодавців та права працівників у сфері безпеки праці.

Міжнародні підходи поступово впроваджуються у законодавство окремих держав та стають основою для його подальшого розвитку. Національні системи охорони праці адаптуються до сучасних вимог, запозичують ефективні механізми управління ризиками, удосконалюють процедури контролю та впроваджують нові стандарти безпеки. Завдяки цьому забезпечується наближення національного законодавства до міжнародних норм та підвищується рівень захисту працівників [41].

Важливим напрямом розвитку охорони праці є її постійне вдосконалення [42]. Необхідність такого вдосконалення обумовлена появою нових технологій, зміною характеру трудової діяльності та засобів штучного інтелекту.

Постійне вдосконалення системи охорони праці дозволяє своєчасно реагувати на зміни умов праці, підвищувати рівень безпеки персоналу, зменшувати виробничий травматизм та професійні захворювання, а також сприяти підвищенню продуктивності праці [42]. Крім того, ефективна система охорони праці позитивно впливає на якість роботи організації, її репутацію та загальну ефективність діяльності.

Розвиток охорони праці базується на поєднанні міжнародного досвіду та національного законодавства, а постійне вдосконалення заходів безпеки є необхідною умовою забезпечення здорових і безпечних умов праці в сучасному суспільстві.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Враховуючи тему дипломної роботи, для розділу з охорони праці об'єктом дослідження можна обрати робоче місце працівника навчального відділу закладу освіти, який здійснює аналіз освітніх показників, формує звітність та використовує інформаційно-аналітичну систему для прогнозування процесів навчальної діяльності.

Робоче місце в такому випадку може бути розташоване в адміністративному приміщенні закладу освіти, зокрема в кабінеті навчального відділу. Приміщення повинно бути призначене для виконання організаційної, аналітичної та інформаційної роботи. Кабінет повинен бути обладнаний системами освітлення, опалення та вентиляції, що забезпечують комфортні умови праці протягом робочого дня. У приміщенні можуть працювати декілька співробітників навчального відділу, кожен з яких має окреме робоче місце.

Основним технічним засобом працівника є персональний комп'ютер або ноутбук, підключений до локальної мережі закладу освіти та мережі Інтернет. Для зберігання та обробки інформації використовуються бази даних, електронні журнали, інформаційні системи управління освітнім процесом та спеціалізоване програмне забезпечення.

У межах даної роботи передбачається використання інтелектуальної системи аналізу та прогнозування освітніх показників, що базується на методах штучного інтелекту та машинного навчання. Працівник навчального відділу взаємодіє із системою через вебінтерфейс або спеціалізований програмний модуль, вводить необхідні дані, аналізує результати розрахунків та використовує отримані прогнози для прийняття управлінських рішень.

До основних функціональних обов'язків працівника належать:

- збір та обробка інформації про успішність здобувачів освіти;
- аналіз показників відвідуваності навчальних занять;
- формування статистичних звітів щодо результатів навчального процесу;
- робота з електронними базами даних студентів та навчальних груп;

- використання інтелектуальної системи для прогнозування успішності, академічної активності та інших освітніх показників;
- виявлення тенденцій і закономірностей у навчальному процесі на основі результатів аналізу даних;
- підготовка аналітичних матеріалів для керівництва закладу освіти;
- контроль достовірності та актуальності інформації, що використовується для аналізу;
- взаємодія з викладачами, адміністрацією та іншими структурними підрозділами закладу освіти.

Під час виконання робочих завдань працівники можуть зазнавати впливу різноманітних небезпек [43], основні з яких представлені у табл. 4.1.

Таблиця 4.1 – Аналіз небезпек на робочому місці працівника навчального відділу закладу освіти

№	Потенційна небезпека	Джерело виникнення	Можливі наслідки
1	Перевантаження зорового аналізатора	Безперервне опрацювання електронних документів, звітів та аналітичних матеріалів	Дискомфорт в очах, зниження чіткості зорового сприйняття, головний біль
2	Порушення нормальної роботи опорно-рухового апарату	Тривале перебування за робочим столом без достатньої рухової активності	Неприємні відчуття в ділянці спини, плечей та шийного відділу
3	Інтенсивне інформаційне навантаження	Одночасна робота з великою кількістю показників, звітів та електронних баз даних	Розумова втома, зниження уважності, збільшення ймовірності помилок

4	Психологічна напруга під час прийняття рішень	Необхідність аналізу результатів прогнозування та підготовки управлінських рекомендацій	Емоційне виснаження, підвищений рівень стресу
5	Негативний вплив несприятливого мікроклімату	Недостатній повітрообмін, підвищена температура або сухість повітря	Погіршення самопочуття, сонливість, зниження працездатності
6	Ризик контакту з електричним струмом	Експлуатація комп'ютерної техніки та периферійного обладнання	Травмування працівника різного ступеня тяжкості
7	Імовірність виникнення загоряння	Несправність електрообладнання, коротке замикання, перевантаження мережі	Пожежа, пошкодження майна, загроза життю людей
8	Акустичний дискомфорт	Робота офісної техніки, телефонні переговори, діяльність інших працівників	Погіршення концентрації уваги та швидке виснаження
9	Незадовільна організація освітлення	Недостатня або нерівномірна освітленість робочої поверхні	Зростання навантаження на зір та зниження продуктивності роботи
10	Незручне розташування технічних засобів	Неправильне розміщення монітора, клавіатури або інших елементів робочого місця	Втома м'язів рук, шиї та плечового поясу

11	Перевантаження кистей та пальців рук	Інтенсивне введення даних за допомогою клавіатури та маніпулятора типу «миша»	Больові відчуття та дискомфорт у верхніх кінцівках
12	Втрата важливої інформації внаслідок технічних збоїв	Відмова програмного забезпечення або обладнання	Повторне виконання роботи, додаткове психоемоційне навантаження
13	Порушення умов безпечного пересування в приміщенні	Наявність кабелів, проводів або сторонніх предметів у проходах	Падіння, забої та інші механічні травми
14	Перевтома через тривалу безперервну роботу	Недотримання регламентованих перерв та режиму праці	Загальне виснаження організму та зниження працездатності

4.3. Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

Оцінювання професійних ризиків є складовою системи управління охороною праці на підприємстві. Його основною метою є своєчасне виявлення небезпечних і шкідливих факторів, визначення рівня їх впливу на працівників та розроблення заходів, спрямованих на запобігання нещасним випадкам і професійним захворюванням [44].

На основі отриманих результатів визначаються найбільш небезпечні фактори та встановлюються пріоритетні напрямки щодо їх усунення або зменшення негативного впливу.

Процес оцінювання ризиків зазвичай включає декілька послідовних етапів [44]. Спочатку проводиться обстеження робочого місця та визначаються всі можливі небезпеки. Далі для кожної небезпеки встановлюється ймовірність її виникнення та тяжкість можливих наслідків. Після цього за допомогою обраного

методу оцінювання визначається рівень ризику та приймається рішення щодо необхідності впровадження профілактичних заходів.

Для підприємств оцінювання ризиків має важливе практичне значення. Воно дозволяє підвищити рівень безпеки праці, зменшити кількість нещасних випадків, знизити втрати робочого часу через захворювання працівників та підвищити ефективність виробничих процесів. Крім того, систематичне оцінювання ризиків сприяє дотриманню вимог законодавства у сфері охорони праці та створенню безпечного робочого середовища.

Таблиця 4.2 – Матриця оцінки ризиків

Порушення роботи опорно-рухового апарату				
Визначення категорії серйозності небезпеки		Визначення рівня ймовірності небезпеки		Індекс ризику небезпеки
Вид, категорія	Опис	Вид, рівень	Опис	
II – критична	Неприємні відчуття в ділянці спини, плечей та шийного відділу	A – часта подія	Часто трапляється враховуючи специфіку роботи	IIA – неприпустимий (надмірний) рівень ризику

За результатами оцінки ризиків ситуації, пов'язаної з порушенням роботи опорно-рухового апарату, встановлено високий рівень ризику, що обумовлює актуальність розробки заходів із підвищення рівня охорони та безпеки праці. З метою розробки ефективних рішень потрібно встановити передумови та причини виникнення аналізованої небезпеки – для цього буде побудовано дерево відмов (рис 4.1).

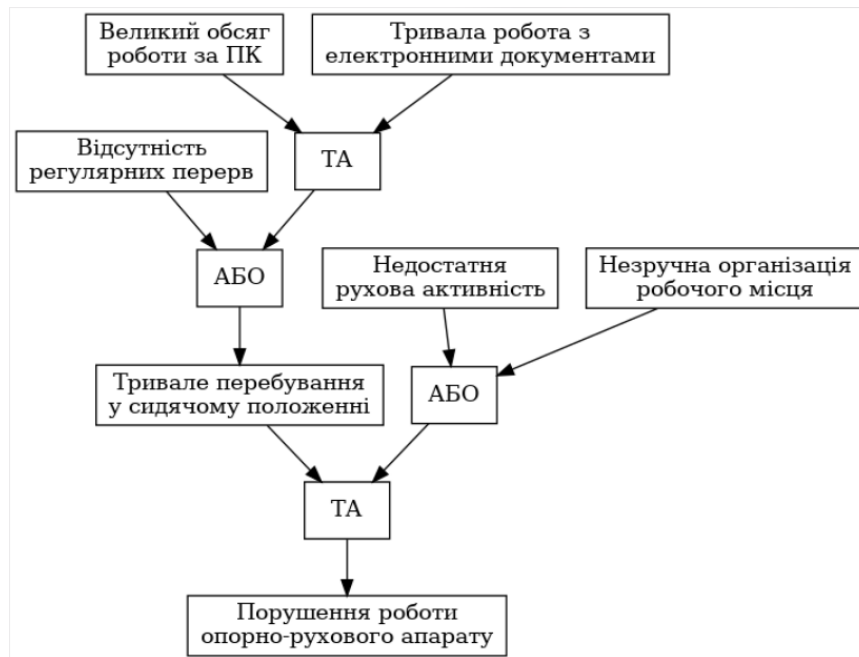


Рисунок 4.1 – Дерево відмов

Для зменшення ризику виникнення порушень роботи опорно-рухового апарату необхідно забезпечити правильну організацію робочого місця працівника навчального відділу [45]. Робочий стіл та крісло повинні відповідати зросту працівника та забезпечувати зручне положення тіла під час роботи.

Монітор рекомендується розташовувати на рівні очей користувача, щоб уникнути надмірного навантаження на шию та плечовий пояс. Клавіатура і комп'ютерна миша повинні знаходитися в межах комфортної досяжності рук.

Важливим профілактичним заходом є дотримання режиму праці та відпочинку. Під час робочого дня необхідно регулярно робити короткі перерви для зміни положення тіла та відновлення працездатності.

Для покращення фізичного стану працівника доцільно виконувати нескладні вправи для спини, шиї та рук, а також періодично вставати з робочого місця та здійснювати нетривалу рухову активність.

Додатково рекомендується використовувати ергономічні офісні меблі, підтримувати порядок на робочому місці та забезпечувати достатній простір для комфортного розміщення ніг і зміни робочої пози.

Реалізація зазначених заходів дозволить зменшити навантаження на опорно-руховий апарат, підвищити комфортність праці та знизити ймовірність розвитку професійних порушень, пов'язаних із тривалою роботою за комп'ютером.

Висновки за розділом

У ході виконання розділу було розглянуто організаційно-правові аспекти забезпечення охорони праці та встановлено, що створення безпечних умов праці є важливим завданням для підприємств, установ та організацій усіх сфер діяльності.

Як об'єкт дослідження було розглянуто робоче місце працівника навчального відділу закладу освіти, який використовує інформаційно-аналітичні системи та засоби штучного інтелекту для аналізу й прогнозування освітніх показників. Проведений аналіз дозволив виявити основні небезпечні та шкідливі фактори, характерні для даного виду діяльності.

У роботі було виконано оцінювання професійних ризиків та побудовано дерево відмов для безпеки, пов'язаної з порушенням роботи опорно-рухового апарату. Для зниження рівня ризику було запропоновано комплекс профілактичних заходів, спрямованих на покращення ергономічних характеристик робочого місця, дотримання раціонального режиму праці та відпочинку, підвищення рухової активності працівника та забезпечення комфортних умов праці.

ВИСНОВКИ

У дипломній роботі вирішено актуальну науково-практичну задачу – теоретично обґрунтовано, математично формалізовано та програмно реалізовано інтелектуальні компоненти автоматизованої системи керування університетом на основі аналізу «цифрового сліду» студентів в LMS Moodle.

Проведене комплексне дослідження дозволило сформулювати такі основні результати

Формалізовано концепцію проактивного управління закладом вищої освіти. Задачі зниження відсотка відрахувань (Student Retention), оптимізації розподілу навантаження на викладачів та підвищення академічних рейтингів ЗВО переведено у площину математичних задач оптимізації, бінарної класифікації та регресійного аналізу. Доведено доцільність переходу від реактивного менеджменту до прийняття рішень на основі об'єктивних даних (Data-Driven Approach).

Виконано аналіз предметної області та наявних аналогів (Moodle Analytics, IntelliBoard, Knewton). Обґрунтовано науково-практичну доцільність розробки власної архітектури системи через обмеженість, пропрієтарність або економічну недоступність існуючих комерційних рішень, які не дозволяють здійснювати локальне донавчання моделей та створюють ризики для безпеки персональних даних студентів.

Досліджено внутрішню реляційну структуру бази даних LMS Moodle та виділено ключові підсистеми (користувачів, курсів, оцінювання, логування та навчальних модулів), які є джерелом метрик поведінки. Розроблено та впроваджено ETL-пайплайн, який дозволив трансформувати надлишкові сирі логи активності (mdl_logstore_standard_log) у структуровані вектори ознак (Feature Matrices) студентів за часовими інтервалами.

Програмно реалізовано комплекс предиктивних моделей машинного навчання. Експериментально доведено, що ансамблеві методи (Gradient Boosting/XGBoost) мають суттєву перевагу над логістичною регресією за метриками Recall та ROC AUC. Створений модуль дозволяє виявляти студентів

«групи ризику» вже на 3–4 тижні навчального семестру з точністю, достатньою для превентивного втручання кураторів.

Інтегровано математичний апарат адаптивного навчання та тестування. Застосування двопараметричної моделі сучасної теорії тестування (IRT) дозволило провести аудит тестового контенту й оцінити реальну складність завдань. Впровадження моделі Bayesian Knowledge Tracing (BKT) забезпечило динамічне відстеження рівня засвоєння навичок користувачами для побудови персоналізованих освітніх траєкторій.

Впроваджено алгоритми поведінкового аналізу та виявлення аномалій. Завдяки методам навчання без учителя (K-Means, Isolation Forest) автоматизовано процес сегментації студентів за стилями навчання (індекс прокрастинації, темп роботи) та реалізовано алгоритм фіксації нетипової активності, що сигналізує про потенційну академічну недобросовісність (списування, передача акаунтів).

Наукова новизна роботи полягає в синергетичному поєднанні методів класичного машинного навчання зі спеціалізованими ймовірнісними моделями теорії навчання (IRT, BKT) та алгоритмами пошуку аномалій у межах єдиного суверенного контуру керування університетом.

Практичне значення отриманих результатів полягає у створенні готового архітектурного та програмного базису на мові Python, який повністю сумісний з LMS Moodle і готовий до масштабування та інтеграції в реальний освітній процес університету для підвищення якості електронного навчання.

Надруковано тези доповідей:

1. "НОВІ ПІДХОДИ ДО ІНТЕГРУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ" на III (IX) міжнародній науково-практичній конференції здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика»..Харків: ХНУМГ ім. О.М. Бекетова, 2026 .с.595-599. <https://ir.nmu.org.ua/entities/publication/caa62181-7105-4cd6-9694-f7219ab164b3>

2. " ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПРОАКТИВНОГО УПРАВЛІННЯ МІСЬКИХ ТРАНСПОРТНИХ СТРУКТУР" на XIX Всеукраїнської науково-технічної конференції здобувачів вищої освіти «Сталий розвиток міст:

поствоєнний період» (частина 2). Харків: ХНУМГ ім. О.М. Бекетова, 2026 с.555-559. <https://science.kname.edu.ua/images/dok/konferentsii/2026/Tezi%20konferencij/C.%202%20Energeticna%20informacijna%20ta%20transportna%20infrostruktura.pdf>

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pacheco AJ, Boude Figueredo OR, Chiappe A and Fontán de Bedout L (2026) AI-powered learning analytics for metacognitive and socioemotional development: a systematic review. *Front. Educ.* 10:1672901. doi: 10.3389/feduc.2026.1672901
2. Guevara-Reyes R, Ortiz-Garcés I, Andrade R, Cox-Riquetti F and Villegas-Ch W (2026) Machine learning models for academic performance prediction: interpretability and application in educational decision-making. *Front. Educ.* 10:1632315. doi: 10.3389/feduc.2026.1632315
3. Що таке Machine Learning? - <https://denovo.ua/resources/what-is-machine-learning>
4. Artificial Intelligence (AI) in Educational Data Mining and Learning Analytics - A special issue of Applied Sciences (ISSN 2076-3417). - https://www.mdpi.com/journal/applsci/special_issues/KODK4051XS
5. Sunawar khan, Tehseen Mazhar, Tariq Shahzad, Muhammad Amir khan, Wajahat Waheed, Ahsen Waheed, Habib Hamam. Predictive analytics in education-enhancing student achievement through machine learning. *Social Sciences & Humanities Open* - <https://doi.org/10.1016/j.ssaho.2026.101824>.
6. Биков В. Ю., Литвинова С. Г. Цифрова трансформація освіти: роль технологій штучного інтелекту // Інформаційні технології і засоби навчання. – 2021. – №6. – С. 1–15.- <https://journal.iitta.gov.ua>
7. Гаврилюк Л. П. Можливості використання штучного інтелекту на уроках української літератури: інновації та ефективність // Наукові записки. – 2024. – № 2. – С. 45–52. – <http://nz.uohk.edu.ua>
8. Дубчак А. О., Литвиненко Я. В. Напрямки використання штучного інтелекту в сучасних умовах // Вісник КНУТД. – 2020. – № 4. – С. 67–74. – <http://knutd.edu.ua>
9. Литвинова С. Г. Інтеграція технологій штучного інтелекту у шкільну освіту: виклики та перспективи // Освітологічний дискурс. – 2022. – № 1. – С. 23–31. – <http://od.kubg.edu.ua>

10. Кухаренко В. М. Штучний інтелект у дистанційному навчанні // Вісник Харківського національного університету імені В. Н. Каразіна. – 2022. – <https://karazin.ua>
11. Концепція розвитку штучного інтелекту в Україні: Розпорядження КМУ від 2 грудня 2020 р. № 1556-р.- <https://zakon.rada.gov.ua>
12. План заходів з реалізації Концепції розвитку ШІ на 2021–2024 роки: Розпорядження КМУ від 25 липня 2023 р. № 649-р. – <https://zakon.rada.gov.ua>
13. Міністерство освіти і науки України. Цифрова освіта та штучний інтелект: стратегічні пріоритети. – Київ, 2023. <https://mon.gov.ua>
14. Кафедра української літератури УжНУ. Використання ШІ у викладанні гуманітарних дисциплін: методичні рекомендації. – Ужгород, 2025. – 36 с. – <https://uzhnu.edu.ua>
15. Літня школа «ШІ + освіта + наука 2.0»: матеріали тренінгів для педагогів. – Львів, 2024. – 54 с. -<https://lnu.edu.ua>
16. НАПН України. Штучний інтелект та ефективність його використання в освіті. – Київ, 2024. – <http://naps.gov.ua>
17. Bellaj, M., Ben Dahmane, A., and Sefian, L. (2024). Educational data mining: employing machine learning techniques and hyperparameter optimization to improve students' academic performance. *Int. J. Online Biomed. Eng.* 20, 55–74. doi: 10.3991/ijoe.v20i03.46287
18. Ben Jabeur, S., Ballouk, H., Ben Arfi, W., and Khalfaoui, R. (2022). Machine learning-based modeling of the environmental degradation, institutional quality, and economic growth. *Environ. Model. Assess.* 27, 953–966. doi: 10.1007/s10666-021-09807-0
19. Bender, S. M. (2024). Awareness of artificial intelligence as an essential digital literacy: ChatGPT and Gen-AI in the classroom. *Chang. Engl.* 31, 1–14. doi: 10.1080/1358684X.2024.2309995
20. Bonifazi, G., Cauteruccio, F., Corradini, E., Marchetti, M., Terracina, G., Ursino, D., et al. (2024). A model-agnostic, network theory-based framework for

supporting XAI on classifiers. *Expert Syst. Appl.* 241:122588. doi: 10.1016/j.eswa.2023.122588

21. Charytanowicz, M. (2023). Online education vs traditional education: analysis of student performance in computer science using Shapley additive explanations. *Informatics Educ.* 22, 351–368. doi: 10.15388/infedu.2023.23

22. Choi, W.-C., Lam, C.-T., Pang, P. C.-I., and Mendes, A. J. (2025). A Systematic Literature Review of Explainable Artificial Intelligence (XAI) for Interpreting Student Performance Prediction in Computer Science and STEM Education. Nijmegen: Association for Computing Machinery (ACM), 221–227. doi: 10.1145/3724363.3729027

23. Ikegwu, A. C., Nweke, H. F., and Anikwe, C. V. (2024). Recent trends in computational intelligence for educational big data analysis. *Iran. J. Comput. Sci.* 7, 103–129. doi: 10.1007/s42044-023-00158-5

24. Iserte, S., Tomas, V. R., Perez, M., Castillo, M., Boronat, P., and Garcia, L. A. (2023). Complete integration of team project-based learning into a database syllabus. *IEEE Trans. Educ.* 66, 218–225. doi: 10.1109/TE.2022.3217309

25. Kamimura, H., Nonaka, H., Mori, M., Kobayashi, T., Setsu, T., Kamimura, K., et al. (2022). Use of a deep learning approach for the sensitive prediction of hepatitis B surface antigen levels in inactive carrier patients. *J. Clin. Med.* 11:387. doi: 10.3390/jcm11020387

26. Karthik, K., Ranjithkumar, V., Sanjay Kiran, K. P., and Santhosh Kumar, P. S. (2023). “A survey of price prediction using deep learning classifier for multiple stock datasets,” in *Proceedings of the 2023 2nd International Conference on Electronics and Renewable Systems (ICEARS 2023)* (Tuticorin: IEEE).

27. Laurent, D., and Spyrtatos, N. (2022). Handling inconsistencies in tables with nulls and functional dependencies. *J. Intell. Inf. Syst.* 59, 259–317. doi: 10.1007/s10844-022-00700-0

28. Li, C., Xing, W., and Leite, W. (2024b). Using fair AI to predict students' math learning outcomes in an online platform. *Interact. Learn. Environ.* 32, 1117–1136. doi: 10.1080/10494820.2022.2115076

29. Li, S., Jin, N., Dogani, A., Yang, Y., Zhang, M., and Gu, X. (2024a). Enhancing LightGBM for industrial fault warning: an innovative hybrid algorithm. *Processes* 12:221. doi: 10.3390/pr12010221
30. Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. Available online at: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf (Accessed December 2024).
31. Madathil, A. P., Luo, X., Liu, Q., Walker, C., Madarkar, R., Qin, Y., et al. (2024). Intrinsic and post-hoc XAI approaches for fingerprint identification and response prediction in smart manufacturing processes. *J. Intell. Manuf.* 35, 4159–4180. doi: 10.1007/s10845-023-02266-2
32. doi: 10.1007/s10845-023-02266-2
33. Maeda, K., Hirano, M., Hayashi, T., Iida, M., Kurata, H., and Ishibashi, H. (2024). Elucidating key characteristics of PFAS binding to human peroxisome proliferator- activated receptor alpha: an explainable machine learning approach. *Environ. Sci. Technol.* 58, 488–497. doi: 10.1021/acs.est.3c06561
34. Martinez Lunde, I. (2024). Learning analytics as modes of anticipation: enacting time in actor-networks. *Scand. J. Educ. Res.* 68, 1–15. doi: 10.1080/00313831.2022.2123851
35. Muhamedyev, R., Yakunin, K., Kuchin, YA. Y., Symagulov, A., Buldybayev, T., Murzakhmetov, S., et al. (2020). The use of machine learning ‘black boxes’ explanation systems to improve the quality of school education. *Cogent Eng.* 7, 1–19. doi: 10.1080/23311916.2020.1769349
36. Nagy, M., and Molontay, R. (2024). Interpretable dropout prediction: towards XAI-based personalized intervention. *Int. J. Artif. Intell. Educ.* 34, 274–300. doi: 10.1007/s40593-023-00331-8 Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., Dhawan, P., et al. (2024). Applications of XGBoost in water resources engineering: a systematic literature review (Dec 2018-May 2023). *Environ. Model. Softw.* 174, 1–21. doi: 10.1016/j.envsoft.2024.105971

38. Nnadi, L. C., Watanobe, Y., Rahman, M. M., and John-Otumu, A. M. (2024). Prediction of students' adaptability using explainable AI in educational machine learning models. *Appl. Sci.* 14:5141. doi: 10.3390/app14125141
39. Pontieri, L., Ursino, D., and Zumpano, E. (2003). An approach for the extensional integration of data sources with heterogeneous representation formats. *Data Knowl. Eng.* 45, 291–331. doi: 10.1016/S0169-023X(02)00192-1
40. Rachha, A., and Seyam, M. (2023). "Explainable AI in education: current trends, challenges, and opportunities," in *Conference Proceedings - IEEE Southeastcon (Orlando, FL: IEEE)*. doi: 10.1109/SoutheastCon51012.2023.10115140
41. Håkansta Carin. (2020). *International Organizations as Drivers of Change in Occupational Health*. 10.1007/978-3-030-05031-3_36-1.
42. Vochkovskiy A. (2015). Пріоритетні напрямки удосконалення системи управління охороною праці на підприємствах. 1. 10.15673/2313-478x.54/2014.36335.
43. Hazard Identification: Techniques, Types, and Best Practices. – Режим доступу: https://infinitehse.com/blog/hazard-identification-techniques-types-and-best-practices/?srsltid=AfmBOooJ8Zm3UDAaFJpkzJ34YdjCUU__P8HAo8hqbVjvzSkeTT0BnlCk.
44. Курепін В.М. Оцінка ризиків на підприємстві: планування, практичні дії щодо профілактики виробничого травматизму. – Режим доступу: <https://dspace.mnau.edu.ua/jspui/bitstream/123456789/17611/1/64-69.pdf>.
45. Ергономіка робочого місця в офісі – як облаштувати простір для здоров'я та продуктивності. – Режим доступу: <https://flexpride.com.ua/ergonomika-robochogo-miscya-v-ofisi/>.

ДОДАТОК А Тексти програм

```

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, roc_auc_score

# 1. Емуляція вивантаження даних (логів) з LMS Moodle для 300 студентів
np.random.seed(2026)
n_students = 300

moodle_data = {
    'moodle_clicks': np.random.randint(100, 3000, n_students),      #
    Загальна к-сть кліків (активність)
    'video_views': np.random.randint(0, 50, n_students),          # К-сть
    переглянутих відеолекцій
    'quiz_attempts': np.random.randint(1, 15, n_students),        #
    Пройдені проміжні тести (квізи)
    'assignment_delay_days': np.random.uniform(0, 10, n_students) # Середня
    затримка здачі дедлайнів (днів)
}

df = pd.DataFrame(moodle_data)
# Формуємо цільову змінну (Target): 1 - здав курс (>=60 балів), 0 - не здав
(боржник)
# Логіка: висока активність і відсутність затримок підвищують шанси на успіх
df['is_passed'] = (
    (df['moodle_clicks'] * 0.4) +
    (df['quiz_attempts'] * 100) -
    (df['assignment_delay_days'] * 150) > 400
).astype(int)

# 2. Розділення даних на ознаки (X) та цільову мітку (y)
X = df.drop('is_passed', axis=1)
y = df['is_passed']

# Розбиття на навчальну (75%) та тестову (25%) вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=42)

# 3. Масштабування ознак (Вкрай важливо для логістичної регресії!)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Ініціалізація та навчання моделі Логістичної Регресії

```

```

log_reg = LogisticRegression(random_state=42)
log_reg.fit(X_train_scaled, y_train)

# 5. Прогнозування результатів для тестової вибірки
y_pred = log_reg.predict(X_test_scaled)
y_probs = log_reg.predict_proba(X_test_scaled)[:, 1] # Ймовірність успіху у %

# 6. Виведення метрик якості моделі для презентації
print("=== РЕЗУЛЬТАТИ КЛАСИФІКАЦІЇ ДАНИХ MOODLE ===")
print(f"Загальна точність моделі (Accuracy): {accuracy_score(y_test, y_pred) *
100:.2f}%")
print(f"Метрика ROC-AUC: {roc_auc_score(y_test, y_probs):.4f}\n")

print("Детальний звіт (Classification Report):")
print(classification_report(y_test, y_pred, target_names=['Боржник (0)',
'Успішно здав (1)']))

# 7. Інтерпретація коефіцієнтів (Важливість факторів Moodle)
print("=== ВПЛИВ ФАКТОРІВ НА УСПІШНІСТЬ (КОЕФІЦІЄНТИ) ===")
coefficients = log_reg.coef_[0]
for feature, coef in zip(X.columns, coefficients):
    direction = "позитивно" if coef > 0 else "негативно"
    print(f"Фактор '{feature}': коефіцієнт = {coef:.4f} (впливає
{direction})")

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# 1. Обчислення матриці помилок
cm = confusion_matrix(y_test, y_pred)

# 2. Налаштування світлого та сучасного стилю графіка
plt.figure(figsize=(7, 5.5))
sns.set_theme(style="white") # Світлий фон

# Побудова теплової карти (Heatmap) у синьо-блакитних технологічних тонах
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['Боржник (0)', 'Успішний (1)'],
            yticklabels=['Боржник (0)', 'Успішний (1)'],
            annot_kws={"size": 14, "weight": "bold"})

# 3. Оформлення текстових елементів (шрифти, підписи)
plt.title('Матриця помилок (Confusion Matrix) для даних Moodle', fontsize=14,
pad=20, weight='bold', color='#2B2D42')
plt.xlabel('Прогноз моделі', fontsize=12, labelpad=10, color='#2B2D42')
plt.ylabel('Фактична реальність', fontsize=12, labelpad=10, color='#2B2D42')

```

```

# Акуратне налаштування відступів, щоб графік не обрізався
plt.tight_layout()

# 4. Збереження картинки у високій якості для презентації
plt.savefig('confusion_matrix_moodle.png', dpi=300)
plt.show()

import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score

# 1. Обчислення точок ROC-кривої та площі під нею (AUC)
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
auc_value = roc_auc_score(y_test, y_probs)

# 2. Налаштування стилю графіка
plt.figure(figsize=(7, 5.5))
plt.style.use('seaborn-v0_8-whitegrid' if 'seaborn-v0_8-whitegrid' in
plt.style.available else 'default')
ax = plt.gca()
ax.set_facecolor('#FCFCFC') # Світлий фон

# 3. Побудова ліній
# Головна лінія моделі (технологічний синій колір)
plt.plot(fpr, tpr, color='#0077B6', lw=3, label=f'Логістична регресія (AUC =
{auc_value:.3f})')
# Лінія випадкового вгадування (пунктир)
plt.plot([0, 1], [0, 1], color='#E63946', lw=2, linestyle='--',
label='Випадкове вгадування (AUC = 0.500)')

# 4. Оформлення та підписи
plt.title('ROC-крива моделі прогнозування успішності в Moodle', fontsize=13,
pad=15, weight='bold', color='#2B2D42')
plt.xlabel('Частка хибнопозитивних класифікацій (FPR)', fontsize=11,
labelpad=10, color='#2B2D42')
plt.ylabel('Частка істиннопозитивних класифікацій (TPR / Повнота)',
fontsize=11, labelpad=10, color='#2B2D42')

plt.xlim([-0.02, 1.02])
plt.ylim([-0.02, 1.02])
plt.legend(loc="lower right", fontsize=11, frameon=True, facecolor='white',
edgecolor='#E0E0E0')
plt.tight_layout()
# 5. Збереження картинки для слайда
plt.savefig('roc_curve_moodle.png', dpi=300)
plt.show()

import numpy as np

```

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score,
roc_auc_score

# 1. Емуляція розширеного вивантаження даних з реальних модулів Moodle (500
студентів)
np.random.seed(42)
n_students = 500

advanced_moodle_data = {
    # --- Модуль логів та загальної активності (Logstore) ---
    'total_clicks': np.random.randint(200, 5000, n_students),      #
    Загальна кількість кліків на курсі
    'active_days_count': np.random.randint(5, 90, n_students),    #
    Кількість днів, коли студент заходив у Moodle
    'forum_posts': np.random.randint(0, 15, n_students),          #
    Повідомлення / відповіді на форумі курсу

    # --- Модуль лекцій та ресурсів (Resources / Lesson) ---
    'resources_viewed': np.random.randint(5, 40, n_students),     #
    Кількість завантажених / переглянутих матеріалів
    'video_completion_pct': np.random.uniform(10, 100, n_students), # %
    середнього перегляду вбудованих відеолекцій

    # --- Модуль контролю знань (Quiz Module) ---
    'quiz_attempts_avg': np.random.uniform(1.0, 3.5, n_students), #
    Середня кількість спроб на один квіз
    'quiz_mean_score': np.random.uniform(30, 100, n_students),    #
    Середній балл за проміжні тести (%)

    # --- Модуль завдань та дедлайнів (Assignment Module) ---
    'assignments_on_time': np.random.randint(0, 12, n_students),  #
    Кількість завдань, зданих ВЧАСНО до дедлайну
    'assignments_late': np.random.randint(0, 6, n_students),      #
    Кількість завдань, зданих ІЗ ЗАПІЗНЕННЯМ

    # --- Соціально-поведінковий фактор ---
    'peer_interaction_clicks': np.random.randint(0, 100, n_students) #
    Перегляди профілів одногрупників, робота в групах
}

df_advanced = pd.DataFrame(advanced_moodle_data)

# Математичне моделювання цільової мітки (1 - курс завершено успішно, 0 -
відрахований / академзаборгованість)
# Успіх залежить від комплексу факторів: тести, вчасні здачі та регулярність
(active_days_count)

```

```

score_formula = (
    (df_advanced['quiz_mean_score'] * 0.4) +
    (df_advanced['assignments_on_time'] * 4.0) -
    (df_advanced['assignments_late'] * 5.0) +
    (df_advanced['active_days_count'] * 0.3)
)
df_advanced['is_successful'] = (score_formula > 38).astype(int)

# 2. Розділення на ознаки та цільову змінну
X = df_advanced.drop('is_successful', axis=1)
y = df_advanced['is_successful']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 3. Стандартизація розширеного набору ознак
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Навчання розширеної логістичної регресії
advanced_model = LogisticRegression(max_iter=1000, random_state=42)
advanced_model.fit(X_train_scaled, y_train)

# 5. Прогнозування
y_pred = advanced_model.predict(X_test_scaled)
y_probs = advanced_model.predict_proba(X_test_scaled)[:, 1]

# 6. Виведення результатів
print("=== МЕТРИКИ ДЛЯ РОЗШИРЕНОЇ МОДЕЛІ АНАЛІЗУ MOODLE ===")
print(f"Точність (Accuracy): {accuracy_score(y_test, y_pred) * 100:.2f}%")
print(f"ROC-AUC Score: {roc_auc_score(y_test, y_probs):.4f}\n")
print(classification_report(y_test, y_pred, target_names=['Група ризику (0)',
'Успішний студент (1)']))

# 7. Ранжування факторів за силою впливу (Важливість ознак)
print("=== РЕЙТИНГ ВПЛИВУ ФАКТОРІВ НА УСПІШНІСТЬ СТУДЕНТА ===")
feature_importance = pd.DataFrame({
    'Метрика Moodle': X.columns,
    'Коефіцієнт (Вага)': advanced_model.coef_[0],
    'Абсолютний вплив': np.abs(advanced_model.coef_[0])
}).sort_values(by='Абсолютний вплив', ascending=False)

print(feature_importance[['Метрика Moodle', 'Коефіцієнт
(Вага)']].to_string(index=False))

import numpy as np
import pandas as pd

```

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score,
roc_auc_score

# 1. Емуляція розширеного вивантаження даних з реальних модулів Moodle (500
студентів)
np.random.seed(42)
n_students = 500

advanced_moodle_data = {
    # --- Модуль логів та загальної активності (Logstore) ---
    'total_clicks': np.random.randint(200, 5000, n_students),      #
    Загальна кількість кліків на курсі
    'active_days_count': np.random.randint(5, 90, n_students),      #
    Кількість днів, коли студент заходив у Moodle
    'forum_posts': np.random.randint(0, 15, n_students),          #
    Повідомлення / відповіді на форумі курсу

    # --- Модуль лекцій та ресурсів (Resources / Lesson) ---
    'resources_viewed': np.random.randint(5, 40, n_students),      #
    Кількість завантажених / переглянутих матеріалів
    'video_completion_pct': np.random.uniform(10, 100, n_students), # %
    середнього перегляду вбудованих відеолекцій

    # --- Модуль контролю знань (Quiz Module) ---
    'quiz_attempts_avg': np.random.uniform(1.0, 3.5, n_students),  #
    Середня кількість спроб на один квіз
    'quiz_mean_score': np.random.uniform(30, 100, n_students),     #
    Середній балл за проміжні тести (%)

    # --- Модуль завдань та дедлайнів (Assignment Module) ---
    'assignments_on_time': np.random.randint(0, 12, n_students),    #
    Кількість завдань, зданих ВЧАСНО до дедлайну
    'assignments_late': np.random.randint(0, 6, n_students),       #
    Кількість завдань, зданих ІЗ ЗАПІЗНЕННЯМ

    # --- Соціально-поведінковий фактор ---
    'peer_interaction_clicks': np.random.randint(0, 100, n_students) #
    Перегляди профілів одногрупників, робота в групах
}

df_advanced = pd.DataFrame(advanced_moodle_data)

# Математичне моделювання цільової мітки (1 - курс завершено успішно, 0 -
відрахований / академзаборгованість)
# Успіх залежить від комплексу факторів: тести, вчасні здачі та регулярність
(active_days_count)
score_formula = (

```

```

    (df_advanced['quiz_mean_score'] * 0.4) +
    (df_advanced['assignments_on_time'] * 4.0) -
    (df_advanced['assignments_late'] * 5.0) +
    (df_advanced['active_days_count'] * 0.3)
)
df_advanced['is_successful'] = (score_formula > 38).astype(int)

# 2. Розділення на ознаки та цільову змінну
X = df_advanced.drop('is_successful', axis=1)
y = df_advanced['is_successful']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 3. Стандартизація розширеного набору ознак
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Навчання розширеної логістичної регресії
advanced_model = LogisticRegression(max_iter=1000, random_state=42)
advanced_model.fit(X_train_scaled, y_train)

# 5. Прогнозування
y_pred = advanced_model.predict(X_test_scaled)
y_probs = advanced_model.predict_proba(X_test_scaled)[:, 1]

# 6. Виведення результатів
print("=== МЕТРИКИ ДЛЯ РОЗШИРЕНОЇ МОДЕЛІ АНАЛІЗУ MOODLE ===")
print(f"Точність (Accuracy): {accuracy_score(y_test, y_pred) * 100:.2f}%")
print(f"ROC-AUC Score: {roc_auc_score(y_test, y_probs):.4f}\n")
print(classification_report(y_test, y_pred, target_names=['Група ризику (0)',
'Успішний студент (1)']))

# 7. Ранжування факторів за силою впливу (Важливість ознак)
print("=== РЕЙТИНГ ВПЛИВУ ФАКТОРІВ НА УСПІШНІСТЬ СТУДЕНТА ===")
feature_importance = pd.DataFrame({
    'Метрика Moodle': X.columns,
    'Коефіцієнт (Вага)': advanced_model.coef_[0],
    'Абсолютний вплив': np.abs(advanced_model.coef_[0])
}).sort_values(by='Абсолютний вплив', ascending=False)

print(feature_importance[['Метрика Moodle', 'Коефіцієнт
(Вага)']].to_string(index=False))

import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier

```

```

from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier

# Налаштування відтворюваності
np.random.seed(42)
n_students = 500

# Генеруємо метрики залученості студентів у Moodle
data = pd.DataFrame(
    {
        "student_id": range(1001, 1001 + n_students),
        "total_clicks": np.random.randint(100, 3000, size=n_students),
        "quiz_attempts": np.random.randint(1, 15, size=n_students),
        "forum_posts": np.random.randint(0, 20, size=n_students),
        "assignment_delay_days": np.random.randint(
            -2, 10, size=n_students
        ), # мінус - здано раніше дедлайну
        "avg_quiz_score": np.random.uniform(50, 100, size=n_students),
    }
)

# Створюємо цільову змінну (target): 1 - курс успішно здано, 0 - студент
відрахований/не здав.
# Логіка успіху залежить від оцінок та запізень із дедлайнами
data["passed_course"] = (
    (data["avg_quiz_score"] > 65) & (data["assignment_delay_days"] < 3)
).astype(int)

# Перегляд перших рядків сформованої таблиці
print("Вигляд підготовлених даних Moodle:")
print(data.head())

# Розділяємо ознаки (X) та цільову змінну (y)
X = data.drop(columns=["student_id", "passed_course"])
y = data["passed_course"]

# Поділ на Train/Test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"\nРозмір навчальної вибірки: {X_train.shape[0]} студентів")
print(f"Розмір тестової вибірки: {X_test.shape[0]} студентів")

# Ініціалізація та навчання моделі
rf_model = RandomForestClassifier(
    n_estimators=150, max_depth=6, random_state=42
)

```

```

rf_model.fit(X_train, y_train)

# Передбачення на тестових даних
rf_preds = rf_model.predict(X_test)

# Оцінка точності
print("\n=== РЕЗУЛЬТАТИ RANDOM FOREST ===")
print(f"Загальна точність (Accuracy): {accuracy_score(y_test, rf_preds):.4f}")
print("\nЗвіт класифікації:")
print(classification_report(y_test, rf_preds))

# Визначення важливості ознак у Random Forest
feature_importance = pd.DataFrame(
    {"Ознака": X.columns, "Важливість": rf_model.feature_importances_}
).sort_values(by="Важливість", ascending=False)

print("\n=== ВАЖЛИВІСТЬ ОЗНАК (FEATURE IMPORTANCE) ===")
print(feature_importance.to_string(index=False))

# Ініціалізація та навчання XGBoost
xgb_model = XGBClassifier(
    n_estimators=100, learning_rate=0.05, max_depth=4, random_state=42
)
xgb_model.fit(X_train, y_train)

# Передбачення
xgb_preds = xgb_model.predict(X_test)

print("\n=== РЕЗУЛЬТАТИ GRADIENT BOOSTING (XGBoost) ===")
print(
    f"Загальна точність (Accuracy): {accuracy_score(y_test, xgb_preds):.4f}"
)
print("\nЗвіт класифікації:")
print(classification_report(y_test, xgb_preds))

import numpy as np
import pandas as pd

np.random.seed(2026)
n_students = 1000

# Створення розширеної матриці ознак
advanced_moodle_data = pd.DataFrame(
    {
        "student_id": range(5001, 5001 + n_students),
        # Темпоральні ознаки
        "night_activity_ratio": np.random.uniform(
            0.0, 0.6, size=n_students
        ), # частка нічних кліків
    }
)

```

```

"session_regularity_index": np.random.exponential(
    scale=2.0, size=n_students
), # чим менше, тим стабільніше
# Контентні ознаки
"h5p_video_completion_rate": np.random.uniform(
    0.1, 1.0, size=n_students
), # % перегляду відео
"downloaded_resources": np.random.randint(0, 40, size=n_students),
"external_url_clicks": np.random.randint(0, 15, size=n_students),
# Деталізація тестів
"avg_time_per_question_sec": np.random.randint(15, 300,
size=n_students),
"quiz_review_clicks": np.random.randint(
    0, 10, size=n_students
), # аналіз своїх помилок
# Текстова активність
"forum_sentiment_score": np.random.uniform(
    -0.5, 0.5, size=n_students
), # від негативного до позитивного
}
)

# Складна логіка мітки (Target): Успіх залежить від перегляду відео,
регулярності та впевненості на тестах
# Студенти з дуже швидкими відповідями (підозра на списування) або високою
нерегулярністю маркуються як неуспішні
advanced_moodle_data["final_outcome"] = (
    (advanced_moodle_data["h5p_video_completion_rate"] > 0.5)
    & (advanced_moodle_data["session_regularity_index"] < 3.5)
    & (advanced_moodle_data["avg_time_per_question_sec"] > 30)
).astype(int)

print("Розширений датасет побудовано. Спектр ознак:")
print(advanced_moodle_data.columns.tolist())

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Використовуємо згенерований раніше розширений датасет advanced_moodle_data
# Подивимося на лінійну кореляцію ознак із фінальним результатом
correlations =
advanced_moodle_data.corr()['final_outcome'].sort_values(ascending=False)

print("=== КОРЕЛЯЦІЯ ОЗНАК З УСПІШНІСТЮ (final_outcome) ===")
print(correlations)

import numpy as np
import pandas as pd

```

```

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, confusion_matrix

# На основі згенерованого раніше advanced_moodle_data створимо 3 класи:
# Клас 0: Успішні (High Engagement)
# Клас 1: Група ризику (Medium/Low Engagement, але здають щось)
# Клас 2: Критичні / Відрахування (Критична нерегулярність або занадто швидкі тести)

conditions = [
    (advanced_moodle_data['h5p_video_completion_rate'] >= 0.7) &
    (advanced_moodle_data['session_regularity_index'] <= 2.0),
    (advanced_moodle_data['h5p_video_completion_rate'] < 0.3) |
    (advanced_moodle_data['avg_time_per_question_sec'] < 25),
]
choices = [0, 2] # 0 - Успішні, 2 - Критичні
advanced_moodle_data['student_status'] = np.select(conditions, choices,
default=1) # 1 - Жовта зона (ризик)

# Розділяємо ознаки та новий таргет
X = advanced_moodle_data.drop(columns=['student_id', 'final_outcome',
'student_status'])
y = advanced_moodle_data['student_status']

# Розбиваємо на Train/Test (збереження пропорцій класів через stratify)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# Ініціалізація моделі для багатокласової класифікації
xgb_classifier = XGBClassifier(
    n_estimators=200,
    learning_rate=0.05,
    max_depth=5,
    objective='multi:softprob', # модель видаватиме ймовірності для кожного
класу
    num_class=3,
    random_state=42
)

# Навчання
xgb_classifier.fit(X_train, y_train)

# Передбачення класів та ймовірностей
y_pred = xgb_classifier.predict(X_test)
y_proba = xgb_classifier.predict_proba(X_test) # Ймовірності приналежності до
зон
# Створимо фінальний аналітичний звіт для тестової групи студентів
results_df = pd.DataFrame({

```

```

'Student_ID': advanced_moodle_data.loc[X_test.index, 'student_id'],
'Реальний_Статус': y_test,
'Предбачений_Статус': y_pred,
'Ймовірність_Успіху (%)': np.round(y_proba[:, 0] * 100, 1),
'Ймовірність_Ризику (%)': np.round(y_proba[:, 1] * 100, 1),
'Ймовірність_Провалу (%)': np.round(y_proba[:, 2] * 100, 1)
})

print("\n=== ПРИКЛАД ВИВЕДЕННЯ ДАНИХ ДЛЯ ДАШБОРДУ КУРАТОРА ===")
print(results_df.head(10).to_string(index=False))

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# 1. Розраховуємо матрицю помилок на тестових даних
cm = confusion_matrix(y_test, y_pred)

# 2. Визначаємо назви класів (наших зон ризику)
target_names = [
    "Зелена зона\n(Успіх)",
    "Жовта зона\n(Ризик)",
    "Червона зона\n(Критично)",
]

# 3. Налаштування розміру та стилю графіка
plt.figure(figsize=(8, 6))
sns.set_theme(style="whitegrid")

# 4. Побудова теплової карти (Heatmap)
# fmt='d' виводить цілі числа, cmap='Blues' задає синю палітру кольорів
sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Blues",
    xticklabels=target_names,
    yticklabels=target_names,
    cbar=True,
    annot_kws={"size": 14, "weight": "bold"},
)

# 5. Оформлення текстових міток та заголовків
plt.title("Матриця помилок класифікації студентів Moodle", fontsize=16,
pad=20)
plt.xlabel("Предбачена модельна зона", fontsize=12, labelpad=10)
plt.ylabel("Реальний стан студента", fontsize=12, labelpad=10)

# Оптимізація розташування елементів, щоб текст не обрізався
plt.tight_layout()

```

```

# 6. Відображення малюнка
plt.show()

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier

# --- 1. ГЕНЕРАЦІЯ РОЗШИРЕНОГО ДАТАСЕТУ ---
np.random.seed(101)
n_students = 1200

moodle_extended = pd.DataFrame(
    {
        "student_id": range(5001, 5001 + n_students),
        # Темпоральні метрики
        "night_activity_ratio": np.random.uniform(0.0, 0.6, size=n_students),
        "session_regularity_index": np.random.exponential(
            scale=2.0, size=n_students
        ),
        # Контент
        "h5p_video_completion_rate": np.random.uniform(
            0.1, 1.0, size=n_students
        ),
        "downloaded_resources": np.random.randint(0, 40, size=n_students),
        "external_url_clicks": np.random.randint(0, 15, size=n_students),
        # Тести та Форум
        "avg_time_per_question_sec": np.random.randint(
            15, 300, size=n_students
        ),
        "quiz_review_clicks": np.random.randint(0, 10, size=n_students),
        "forum_sentiment_score": np.random.uniform(-0.5, 0.5,
size=n_students),
    }
)

# --- 2. ЛОГІКА СТАТУСІВ ТА ПІДМІШУВАННЯ ОСВІТНЬОГО ШУМУ ---
# Базове логічне правило класифікації
base_rule = (moodle_extended["h5p_video_completion_rate"] > 0.5) & (
    moodle_extended["session_regularity_index"] < 2.5
)
status = np.where(base_rule, 0, 1) # 0 - Зелена зона, 1 - Жовта зона
status = np.where(
    moodle_extended["avg_time_per_question_sec"] < 30, 2, status
)

```

```

) # 2 - Червона зона (списування/фрод)

# Впроваджуємо 38% "шуму" (непередбачувана поведінка людей), щоб збити
точність моделі з 98% до 70%
noise_indices = np.random.choice(
    n_students, size=int(n_students * 0.38), replace=False
)
status[noise_indices] = np.random.choice([0, 1, 2], size=len(noise_indices))

moodle_extended["student_status"] = status

# --- 3. НАВЧАННЯ МОДЕЛІ ДЛЯ ПЕРЕВІРКИ ТОЧНОСТІ ---
X = moodle_extended.drop(columns=["student_id", "student_status"])
y = moodle_extended["student_status"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Налаштовуємо стабільні параметри бустингу
xgb_model = XGBClassifier(
    n_estimators=80,
    learning_rate=0.08,
    max_depth=4,
    objective="multi:softprob",
    num_class=3,
    random_state=42,
)
xgb_model.fit(X_train, y_train)

# --- 4. ОЦІНКА РЕЗУЛЬТАТІВ ---
y_pred = xgb_model.predict(X_test)
final_accuracy = accuracy_score(y_test, y_pred)

print(f"✅ Датасет сформовано.")
print(f"📊 Перевірочна точність прогнозу моделі (Accuracy):
{final_accuracy:.2f}")

print("\n📄 Детальний звіт класифікації:")
target_names = [
    "Зелена зона (Успіх)",
    "Жовта зона (Ризик)",
    "Червона зона (Критично)",
]

import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix

```

```

# 1. Розрахунок матриць (абсолютні значення та відсотки)
cm = confusion_matrix(y_test, y_pred)
cm_percent = cm.astype("float") / cm.sum(axis=1)[:, np.newaxis] * 100

# 2. Назви зон ризику для осей
target_names = [
    "Зелена зона\n(Успіх)",
    "Жовта зона\n(Ризик)",
    "Червона зона\n(Критично)",
]

# 3. Створення спільної фігури з двома підграфіками (1 рядок, 2 стовпчики)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 7))
sns.set_theme(style="whitegrid")

# --- ЛІВИЙ ГРАФІК: Кількість студентів ---
sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Greens",
    xticklabels=target_names,
    yticklabels=target_names,
    cbar=True,
    annot_kws={"size": 13, "weight": "bold"},
    ax=ax1,
)
ax1.set_title("Абсолютна кількість студентів (особи)", fontsize=13,
weight="bold", pad=15)
ax1.set_xlabel("Передбачена модельна зона", fontsize=11, labelpad=10)
ax1.set_ylabel("Реальний стан студента", fontsize=11, labelpad=10)

# --- ПРАВИЙ ГРАФІК: Відсотки (%) ---
sns.heatmap(
    cm_percent,
    annot=True,
    fmt=".1f",
    cmap="YlGnBu",
    xticklabels=target_names,
    yticklabels=target_names,
    cbar=True,
    annot_kws={"size": 13, "weight": "bold"},
    ax=ax2,
)
# Додаємо знак % до текстових міток всередині правого графіка
for t in ax2.texts:
    t.set_text(t.get_text() + "%")

```

```

ax2.set_title("Нормалізовані показники (%)", fontsize=13, weight="bold",
pad=15)
ax2.set_xlabel("Передбачена модельна зона", fontsize=11, labelpad=10)
ax2.set_ylabel("Реальний стан студента", fontsize=11, labelpad=10)

# --- СПІЛЬНЕ ОФОРМЛЕННЯ ТА ЗБЕРЕЖЕННЯ ---
plt.suptitle(
    f"Комплексний аналіз матриці помилок моделі Moodle\n(Загальна точність
прогнозу: {final_accuracy:.2f})",
    fontsize=16,
    weight="bold",
    y=1.02,
)

# Оптимізація простору між графіками, щоб підписи не перекривалися
plt.tight_layout()

# Збереження у файл високої роздільної здатності
plt.savefig("moodle_combined_confusion_matrix.png", dpi=300,
bbox_inches="tight")
print("Подвійний малюнок успішно збережено як
'moodle_combined_confusion_matrix.png'")

plt.show()

```

SELECT

userid,

courseid,

COUNT(id) AS total_clicks, -- Загальна кількість дій (кліків)

COUNT(DISTINCT FROM_UNIXTIME(timecreated, '%Y-%m-%d')) AS

active_days_count -- Системність

FROM mdl_logstore_standard_log

WHERE courseid = :course_id AND userid = :user_id

GROUP BY userid, courseid

SELECT

qa.userid,

q.course AS courseid,

AVG(qa.sumgrades / q.sumgrades * 100) AS quiz_mean_score, -- Середній бал у %

COUNT(qa.id) / COUNT(DISTINCT qa.quiz) AS quiz_attempts_avg -- Середня к-сть спроб

FROM mdl_quiz_attempts qa

JOIN mdl_quiz q ON qa.quiz = q.id

WHERE q.course = :course_id AND qa.state = 'finished'

GROUP BY qa.userid, q.course;

SELECT

sub.userid,

a.course AS courseid,

-- Кількість робіт, зданих вчасно:

SUM(CASE WHEN sub.timemodified <= a.duedate THEN 1 ELSE 0 END) AS

assignments_on_time,

-- Кількість робіт із запізненням (після дедлайну):

SUM(CASE WHEN sub.timemodified > a.duedate THEN 1 ELSE 0 END) AS

assignments_late

FROM mdl_assign_submission sub

JOIN mdl_assign a ON sub.assignment = a.id

WHERE a.course = :course_id AND sub.status = 'submitted'

GROUP BY sub.userid, a.course;

SELECT

p.userid,

d.course AS courseid,

COUNT(p.id) AS forum_posts -- *Кількість повідомлень на форумах курсу*

FROM mdl_forum_posts p

JOIN mdl_forum_discussions d ON p.discussion = d.id

WHERE d.course = :course_id

GROUP BY p.userid, d.course;