

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інженерії
програмного забезпечення
_____ Євген ДАВИДЕНКО
«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
СИСТЕМА АНАЛІЗУ ТА ПРОГНОЗУВАННЯ НАВЧАЛЬНОЇ
ПРОДУКТИВНОСТІ НА ОСНОВІ АКАДЕМІЧНИХ БАЛІВ

Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувач

_____ Олександр ШЛЯХТУН
«___» _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Гліб ГОРБАНЬ
«___» _____ 2024 р.

Чорноморський національний університет імені Петра Могили

(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії
програмного забезпечення
_____ Євген ДАВИДЕНКО
«___» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Шляхтуна Олександра Сергійовича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Система аналізу та прогнозування навчальної продуктивності на основі академічних балів

Затверджена наказом ЧНУ ім. Петра Могили від « 04 » 09 2024 р.
№ 220

2. Строк представлення кваліфікаційної роботи « ___ » _____ 2024 р.

3. Очікуваний результат роботи та початкові дані якщо такі потрібні:

Вхідні дані до роботи – функціональні та нефункціональні вимоги до системи аналізу та прогнозування навчальної продуктивності на основі академічних балів, дані про здобувачів у форматі Excel-файлу. Результат – функціонуючий застосунок

4. Перелік питань, що підлягають розробці:

- аналіз предметної області та існуючих аналогів;
- специфікація функціональних та нефункціональних вимог;
- проєктування та моделювання системи;
- реалізація, тестування та відлагодження застосунка.

5. Перелік графічних матеріалів:

Презентація

Керівник роботи

Особистий підпис

Гліб ГОРБАНЬ

Власне ім'я ПРІЗВИЩЕ

Здобувач

Особистий підпис

Олександр ШЛЯХТУН

Власне ім'я ПРІЗВИЩЕ

Дата видачі завдання « ____ » _____ 20 ____ р

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Система аналізу та прогнозування навчальної продуктивності на основі академічних балів

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	10.09.2024 р.	11.09.2024 р.	виконано
2.	Огляд літератури за темою роботи	13.09.2024 р.	19.09.2024 р.	виконано
3.	Складання календарного плану КМР	20.03.2024 р.	21.03.2024 р.	виконано
4.	Аналіз предметної області	24.09.2024 р.	26.09.2024 р.	виконано
5.	Розробка проектних рішень	28.09.2024 р.	30.09.2024 р.	виконано
6.	Моделювання та конструювання ПЗ	01.10.2024 р.	06.10.2024 р.	виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування	11.10.2024 р.	07.11.2024 р.	виконано
8.	Оформлення КМР та презентації	10.11.2024 р.	20.11.2024 р.	виконано
9.	Попередній захист	28.11.2024 р.	28.11.2024 р.	виконано
10.	Завершення оформлення КМР та презентації	01.12.2024 р.	14.12.2024 р.	виконано
11.	Відгук керівника КМР	14.12.2024 р.	16.12.2024 р.	виконано
12.	Рецензування	16.12.2024 р.	18.12.2024 р.	виконано
13.	Захист кваліфікаційної роботи	19.12.2024 р.	20.12.2024 р.	виконано

Розробив здобувач Шляхтун О. С.

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 2024 р.

Керівник роботи канд. техн. наук, доцент Горбань Г. В.

(посада, прізвище, ім'я, по батькові)

(підпис)

«__» _____ 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи магістра

«Система аналізу та прогнозування навчальної продуктивності на основі академічних балів»

Здобувач 608м гр.: Шляхтун Олександр Сергійович

Керівник: канд. техн. наук, доцент Горбань Г. В.

Дана кваліфікаційна магістерська робота присвячена розробці системи аналізу та прогнозування навчальної продуктивності на основі академічних балів з використанням методів машинного навчання. Робота адресована викладачам і експертам у галузі вищої освіти, зацікавленим в покращенні процесів оцінювання здобувачів.

Об'єктом є процес оцінювання успішності здобувачів вищої освіти.

Предметом є методи та засоби розробки інформаційної системи для моніторингу рейтингових балів та прогнозування подальшої успішності.

Метою дослідження є оптимізація процесу обліку та обробки даних про результати навчання, що дозволяє виявити закономірності в успішності здобувачів вищої освіти та прогнозувати їх подальші результати. Методологія включає аналіз предметної області, розробку моделей даних та процесів, специфікацію вимог, вибір алгоритмів обробки та прогнозування даних, а також реалізацію програмного застосунку для моніторингу рейтингових балів.

Виконано такі завдання:

- а) проаналізовано предметну сферу;
- б) розроблено модель даних;
- в) розроблено модель процесів у системі;
- г) здійснено специфікацію вимог;
- д) обрано алгоритми обробки та прогнозування даних;
- е) реалізовано застосунок для моніторингу рейтингових балів.

Ключові слова: аналіз даних, навчальна продуктивність, прогнозування, машинне навчання, кластеризація, PCA, Python, обробка даних, візуалізація.

Сторінок – 79. Рисуноків – 10. Таблиць – 2. Посилань – 36. Додатків – 1.

ABSTRACT

for the master's qualification work

“System for Analysis and Forecasting of Educational Productivity Based on Academic Grades”

Student of group 608m: Shliakhtun Oleksandr Serhiiiovych

Supervisor: PhD, Associate Professor Horban H. V.

This qualification master's thesis is devoted to the development of a system for analyzing and forecasting academic performance based on academic scores using machine learning methods. The work is addressed to teachers and experts in the field of higher education interested in improving the processes of assessing applicants.

The object is the process of assessing the success of higher education applicants.

The subject is methods and means of developing an information system for monitoring rating scores and predicting further success.

The purpose of the study is to optimize the process of accounting and processing data on learning outcomes, which allows identifying patterns in the success of higher education applicants and predicting their further results. The methodology includes analysis of the subject area, development of data and process models, specification of requirements, selection of data processing and forecasting algorithms, as well as implementation of a software application for monitoring rating scores.

The following tasks were performed:

- a) the subject area was analyzed;
- b) a data model was developed;
- c) a model of processes in the system was developed;
- d) requirements specification was carried out;
- e) data processing and forecasting algorithms were selected;
- f) an application for monitoring rating scores was implemented.

Keywords: data analysis, learning performance, forecasting, machine learning, clustering, PCA, Python, data processing, visualization.

Pages – 79. Figures – 10. Tables – 2. References – 36. Appendices – 1.

ЗМІСТ

ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО СИСТЕМИ	10
1.1 Значення аналізу навчальної продуктивності у вищій освіті.....	10
1.2 Опис і аналіз об'єкта дослідження	11
1.3 Огляд існуючих рішень і сучасних інформаційних технологій	13
1.4 Методи машинного навчання для аналізу освітніх даних.....	17
Висновки до розділу 1.....	20
2 МОДЕЛЮВАННЯ СИСТЕМИ ТА ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
2.1 Аналіз сучасних методів і моделей для вирішення поставлених завдань.....	22
2.2 Моделювання об'єкту і предмету дослідження	26
2.3 Формування специфікації вимог до програмного забезпечення.....	30
2.4 Математичне забезпечення системи.....	40
2.5 Алгоритми роботи системи.....	42
2.6 Інструментальні засоби та технології для реалізації.....	45
Висновки до розділу 2.....	47
3. РОЗРОБКА АРХІТЕКТУРИ ТА СТРУКТУРНИЙ ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	49
3.1 Розробка архітектури програмного забезпечення.....	49
3.2 Моделювання системи за допомогою UML-діаграм	52
3.3 Розробка алгоритмів виконання ключових функцій системи	56
3.4 Опис інтерфейсу користувача	59
Висновки до розділу 3.....	62
4 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	63
4.1 Аналіз розподілу оцінок здобувачів	63
4.2 Візуалізація оцінок здобувачів за предметами.....	65
4.3 Кластеризація здобувачів на основі оцінок.....	68
Висновки до розділу 4.....	71

	8
ВИСНОВКИ	73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	75
ДОДАТОК А Текст програмної реалізації основних функцій	78

ВСТУП

Застосування інформаційних технологій в сфері вищої освіти дозволило докорінно змінити підхід до здійснення навчального процесу та оцінювання досягнень здобувачів вищої освіти. Застосування аналізу та прогнозування навчальної продуктивності допомагає виявити закономірності та слабкі місця у навчальному процесі, що дозволяє реагувати на них. Це сприяє підвищенню якості навчання та досягненню кращих результатів. Оптимізація навчального процесу на основі аналізу академічних даних дозволяє ефективніше використовувати ресурси освітніх закладів, зменшуючи витрати часу, фінансів та зусиль.

Об'єкт – процес оцінювання успішності здобувачів вищої освіти.

Предметом є методи та засоби розробки інформаційної системи для моніторингу рейтингових балів та прогнозування подальшої успішності.

Мета полягає у оптимізації процесу обліку та обробки даних про результати навчання за рахунок розробки інформаційної системи.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- а) проаналізувати предметну сферу;
- б) розробити модель даних;
- в) розробити модель процесів у системі;
- г) здійснити специфікацію вимог;
- д) обрати алгоритми обробки та прогнозування даних;
- е) реалізувати застосунок для моніторингу рейтингових балів.

Розробка системи аналізу та прогнозування навчальної продуктивності має велике значення для покращення якості освіти та досягнення успіху як для здобувачів, так і для закладів освіти.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИМОГ ДО СИСТЕМИ

У першому розділі кваліфікаційної роботи проводиться системний аналіз предметної області, яка стосується оцінювання успішності здобувачів та використання академічних балів для прогнозування їхньої навчальної продуктивності. Сучасні інформаційні технології та методи машинного навчання надають широкі можливості для автоматизації процесів моніторингу і аналізу академічних досягнень.

У цьому розділі буде проведено детальний огляд сучасних систем для моніторингу навчальної продуктивності, аналіз функціональних і структурних особливостей процесу оцінювання здобувачів, а також описано найбільш ефективні підходи до прогнозування навчальних результатів. Цей аналіз є основою для подальшого формулювання вимог до системи та вибору інструментів для її реалізації.

1.1 Значення аналізу навчальної продуктивності у вищій освіті

Аналіз навчальної продуктивності здобувачів є одним із ключових аспектів управління освітнім процесом у закладах вищої освіти [1]. Сучасні тенденції у сфері освіти вимагають від навчальних закладів не тільки забезпечення високого рівня викладання, а й ефективного моніторингу навчальних результатів здобувачів [2]. Це дозволяє виявляти сильні та слабкі сторони навчального процесу, а також приймати своєчасні рішення щодо його вдосконалення.

Застосування інформаційних технологій у навчальному процесі відкриває нові можливості для збору та аналізу даних про навчальні досягнення здобувачів. Зокрема, автоматизовані системи дозволяють накопичувати значні обсяги інформації про академічні бали здобувачів [3], їхню участь у навчальному процесі та інші показники. Аналіз цих даних на основі методів машинного навчання [4] допомагає виявляти закономірності, які традиційними методами залишаються непоміченими. Такі інструменти дозволяють прогнозувати

подальші результати здобувачів, що особливо корисно для своєчасного виявлення ризиків відставання у навчанні або для підтримки обдарованих здобувачів.

Крім того, аналіз навчальної продуктивності є необхідним для підвищення якості освітніх програм і їх адаптації до індивідуальних потреб здобувачів [5]. Своєчасна оцінка результатів навчання дозволяє керівникам навчальних закладів ухвалювати обґрунтовані рішення щодо зміни методів викладання або впровадження нових навчальних матеріалів. Це, у свою чергу, сприяє оптимізації використання ресурсів закладів освіти і зменшує витрати на коригування навчальних планів.

Таким чином, значення аналізу навчальної продуктивності здобувачів не можна недооцінювати, адже він є важливим інструментом підвищення ефективності навчального процесу та досягнення високих результатів у сфері вищої освіти.

1.2 Опис і аналіз об'єкта дослідження

Об'єктом дослідження є процес оцінювання успішності здобувачів вищої освіти, який безпосередньо впливає на якість освіти, результативність навчання та рівень академічної підготовки здобувачів [1]. Оцінювання успішності здобувачів у закладах вищої освіти є багатограним процесом, що включає не лише встановлення академічних результатів, але й аналіз прогресу та впровадження заходів для підвищення продуктивності навчання [2]. Для цього важливо розглянути структурні та функціональні аспекти цього процесу, а також методи збору та обробки даних.

Оцінювання успішності здобувачів здійснюється за допомогою різних видів контролю: поточного, проміжного та підсумкового. Поточний контроль включає регулярні завдання, лабораторні роботи, тестування та інші форми оцінювання, які надають інформацію про рівень засвоєння матеріалу протягом навчального семестру [3]. Проміжний контроль зазвичай включає модульні або

тематичні тести, які допомагають відстежувати прогрес здобувачів у певних блоках навчального курсу. Підсумковий контроль охоплює екзамени та захист курсових і дипломних робіт, що є завершальними етапами оцінювання.

Значну роль у процесі оцінювання відіграє автоматизований збір та обробка даних. У сучасних освітніх закладах часто використовуються спеціалізовані інформаційні системи, такі як Moodle, Blackboard, Canvas та інші [4], які дозволяють зберігати, систематизувати та аналізувати академічні бали здобувачів. Ці системи надають можливість викладачам оперативно отримувати дані про успішність здобувачів, а також налаштовувати різні види оцінювання відповідно до потреб конкретної освітньої програми.

Ключовим аспектом об'єкта дослідження є **прогнозування успішності здобувачів**. На основі зібраних даних про поточні та минулі результати можна прогнозувати майбутні досягнення [5]. Прогнозування навчальної продуктивності є важливим інструментом для прийняття обґрунтованих рішень у навчальному процесі. Зокрема, результати такого аналізу можуть бути використані для виявлення здобувачів, які відстають у навчанні, і вчасного надання їм додаткової підтримки або індивідуальних консультацій.

Академічні бали є одним із головних індикаторів для оцінювання продуктивності навчання [6]. Цей показник є числовим відображенням рівня знань здобувача за конкретний період навчання або з конкретного предмету. Водночас, важливо розуміти, що бали самі по собі можуть бути недостатніми для глибокого аналізу. Тому доцільно враховувати додаткові фактори, такі як активність здобувача в навчальних процесах, його відвідуваність занять, участь у групових проектах та інші дані, що можуть впливати на загальну успішність [7]. Ці фактори дозволяють сформувати більш повну картину про потенційні ризики або можливості для підвищення результативності.

Процес оцінювання успішності також тісно пов'язаний із системою зворотного зв'язку між викладачами і здобувачами [8]. Використовуючи аналіз результатів поточних оцінок, викладачі можуть коригувати свої методи

викладання або змінювати підходи до оцінювання знань здобувачів. Це дозволяє адаптувати навчальний процес під індивідуальні потреби здобувачів, сприяючи їх більш ефективному навчанню.

Оцінювання успішності здобувачів є не тільки інструментом визначення рівня знань, але й засобом для виявлення потенційних проблем в організації навчального процесу. Наприклад, аналіз середніх балів або тенденцій до зниження результатів у конкретних групах здобувачів може свідчити про необхідність перегляду навчальної програми або застосування додаткових методик для покращення засвоєння матеріалу [9].

Окремо варто зазначити важливість інтеграції сучасних технологій в освітній процес для забезпечення більш ефективного моніторингу навчальних досягнень. Використання алгоритмів машинного навчання для аналізу академічних даних [10] надає можливість не тільки швидко обробляти великі обсяги інформації, але й робити точні прогнози щодо майбутньої успішності здобувачів. Це сприяє вчасному виявленню здобувачів, які можуть зіткнутися з труднощами в навчанні, що дозволяє надавати їм індивідуальну допомогу.

Таким чином, процес оцінювання успішності здобувачів є комплексним явищем, що включає як традиційні методи збору та аналізу даних, так і сучасні технології для прогнозування та вдосконалення навчального процесу. Важливим є використання академічних балів та додаткових факторів для побудови ефективної системи моніторингу успішності, яка буде сприяти підвищенню якості освіти та забезпечувати здобувачам умови для максимального розкриття їхнього потенціалу.

1.3 Огляд існуючих рішень і сучасних інформаційних технологій

У процесі аналізу навчальної продуктивності здобувачів важливим елементом є використання сучасних інформаційних систем для збору та аналізу академічних даних [11]. Такі системи дозволяють автоматизувати процеси

оцінювання та прогнозування навчальних досягнень, що сприяє підвищенню якості освіти [1, 2].

Платформа Moodle

Moodle є однією з найбільш популярних систем управління навчанням (Learning Management Systems, LMS), що широко використовується у вищих навчальних закладах [12]. Це платформа з відкритим кодом, яка забезпечує комплексне управління навчальним процесом, включаючи оцінювання успішності здобувачів. Основні можливості системи:

- збирання академічних балів за завдання, тести та інші форми оцінювання;
- створення інтерактивних звітів і відстеження навчальних досягнень у режимі реального часу;
- інтеграція з додатковими модулями для аналізу даних;
- простий інтерфейс для викладачів та адміністраторів.

Однак, Moodle має деякі обмеження в частині прогнозування навчальних результатів. Без інтеграції з додатковими аналітичними інструментами система надає лише статичні звіти на основі минулих даних, а не прогнозування майбутньої успішності [13].

Платформа Blackboard

Blackboard є ще однією популярною комерційною LMS, що пропонує більше можливостей для аналітики. Вона дозволяє:

- збір даних про успішність здобувачів з різних курсів і створювати детальні звіти;
- інтеграцію з аналітичними платформами для обробки великих обсягів даних;
- використання вбудованих інструментів для виявлення трендів у навчальних результатах;
- підтримку глибокої аналітики студентських даних із можливістю прогнозування навчальних результатів.

Blackboard має потужніші інструменти для аналізу і прогнозування порівняно з Moodle, однак вона є комерційним рішенням і потребує значних фінансових вкладень на впровадження і підтримку [14].

Платформа Canvas

Canvas є хмарною платформою для управління навчальним процесом, яка широко використовується у вищих навчальних закладах. Вона забезпечує:

- збір даних про успішність здобувачів з різних навчальних курсів;
- інтеграцію з аналітичними інструментами для побудови прогнозів;
- налаштування індивідуальних панелей для викладачів і здобувачів, що дозволяє в режимі реального часу відстежувати академічні результати;
- можливість застосування API для інтеграції з іншими платформами.

Canvas також має обмеження у прогнозуванні навчальних результатів, хоча й підтримує інші аналітичні системи через API [15].

Alteryx

Alteryx – це інструмент для обробки та аналізу даних, який застосовується для виявлення закономірностей у великих обсягах освітніх даних. Основні можливості:

- інтеграція з існуючими освітніми платформами для збору даних;
- побудова моделей машинного навчання для прогнозування успішності здобувачів;
- виявлення аномалій у навчальних результатах для своєчасного втручання.

Alteryx відрізняється можливістю гнучкого аналізу даних і прогнозування, однак це рішення потребує більш складної технічної підтримки і не призначене для інтеграції в навчальні процеси «з коробки» [16].

IBM Watson Analytics

IBM Watson Analytics є потужною аналітичною платформою з підтримкою штучного інтелекту. Вона дозволяє:

- використовувати машинне навчання для аналізу освітніх даних;

– робити прогнози на основі поточних результатів здобувачів та їхньої взаємодії з навчальними матеріалами;

– генерувати звіти та рекомендації для викладачів щодо покращення успішності.

IBM Watson пропонує широкий спектр аналітичних інструментів, однак його впровадження є досить складним і вимагає значних ресурсів [17].

Порівняння систем

Для більш наочного порівняння можливостей різних платформ, доцільно використати таблицю.

Таблиця 1.1 – Порівняння різних платформ

Платформа	Збір даних про успішність	Прогнозування	Інтеграція з аналітичними інструментами	Доступність	Складність впровадження
Moodle	Так	Ні	Частково	Відкрита	Низька
Blackboard	Так	Так	Так	Комерційна	Висока
Canvas	Так	Ні	Так	Комерційна	Середня
Alteryx	Частково	Так	Так	Комерційна	Висока
IBM Watson	Так	Так	Так	Комерційна	Дуже висока

На основі проведеного аналізу можна зробити висновок, що існуючі платформи для управління навчальним процесом, такі як Moodle, Canvas та Blackboard, забезпечують високий рівень автоматизації оцінювання та збирання даних про успішність здобувачів [18]. Проте лише деякі з них (наприклад, Blackboard) надають можливості для прогнозування результатів на основі зібраних даних.

Для більш точного прогнозування навчальної продуктивності необхідне використання спеціалізованих аналітичних інструментів, таких як Alteryx та IBM

Watson Analytics, які дозволяють застосовувати алгоритми машинного навчання для обробки даних та побудови прогнозів [19]. Однак, їх впровадження вимагає додаткових ресурсів та технічної підтримки.

1.4 Методи машинного навчання для аналізу освітніх даних

Використання машинного навчання у сфері освіти набуває дедалі більшої популярності, оскільки воно дозволяє автоматизувати процеси збору, аналізу та інтерпретації даних [20]. Серед основних підходів до застосування алгоритмів машинного навчання для прогнозування навчальної продуктивності можна виділити:

Класифікація: Класифікаційні алгоритми, такі як дерева рішень, наївний баєсівський класифікатор і метод опорних векторів (SVM), можуть використовуватися для прогнозування ймовірності успішності здобувачів на основі їхніх попередніх академічних досягнень [21]. Ці методи дозволяють класифікувати здобувачів на групи за рівнем успішності, що дає можливість вчасно вжити заходів для покращення результатів. Наприклад, класифікація може допомогти виявити групу здобувачів, які мають ризик неуспішності, та вжити необхідних заходів для їх підтримки [22].

Регресія: Регресійні моделі, такі як лінійна регресія та поліноміальна регресія, застосовуються для прогнозування кількісних показників навчальної продуктивності [23]. Наприклад, регресія може бути використана для передбачення фінальних оцінок здобувачів на основі їхніх попередніх балів. Регресійні моделі також можуть використовуватися для вивчення впливу різних факторів, таких як відвідуваність, участь у заняттях, та інші особисті характеристики, на успішність здобувачів [24].

Аналіз часових рядів: Цей метод дозволяє аналізувати дані, які змінюються з часом, і використовувати їх для прогнозування майбутніх показників [25]. Моделі, такі як ARIMA (авторегресійна інтегрована модель ковзної середньої), можуть бути використані для прогнозування академічної

успішності на основі історичних даних. Аналіз часових рядів може бути особливо корисним для виявлення трендів у навчальній продуктивності протягом семестру або навчального року.

Огляд найбільш ефективних моделей

У контексті аналізу освітніх даних особливо популярні наступні моделі машинного навчання:

Лінійна регресія: Один з найпростіших і найзрозуміліших методів, який використовується для прогнозування залежної змінної (наприклад, оцінок) на основі незалежних змінних (наприклад, кількість годин навчання, відвідуваність тощо) [26]. Ця модель дозволяє швидко оцінити вплив різних факторів на результати навчання, що робить її корисною для початкового аналізу.

Дерева рішень: Цей алгоритм розбиває дані на підгрупи на основі певних критеріїв. Дерева рішень можуть бути використані для класифікації (наприклад, визначення успішності здобувача) або регресії (прогнозування оцінок) [27]. Вони мають просту інтерпретацію і можуть виявляти складні взаємозв'язки між змінними. Важливою перевагою дерев рішень є їхня здатність обробляти як числові, так і категоріальні дані, що робить їх універсальними у застосуванні.

Нейронні мережі: Більш складні моделі, які особливо ефективні для роботи з великими обсягами даних [28]. Нейронні мережі здатні виявляти складні патерни та зв'язки в даних, що робить їх потужним інструментом для прогнозування навчальної продуктивності. Одним із видів нейронних мереж, які часто використовуються в освітньому контексті, є згорткові нейронні мережі (CNN), які зазвичай застосовуються для аналізу зображень, але також можуть бути адаптовані для обробки інших типів даних.

Метод випадкового лісу: Це ансамблевий метод, який використовує кілька дерев рішень для підвищення точності прогнозів [29]. Він добре справляється з проблемами переобучення і є надійним інструментом для аналізу великих наборів даних. Метод випадкового лісу може бути особливо корисним у контексті

освіти, оскільки він може працювати з різноманітними типами даних і надає важливу інформацію про важливість різних змінних у прогнозуванні успішності.

Глибоке навчання: Останнім часом методи глибокого навчання, такі як рекуррентні нейронні мережі (RNN) [30] і трансформери, набирають популярності в аналізі освітніх даних. Ці моделі можуть обробляти послідовні дані, що дозволяє їм враховувати контекст та час у прогнозуванні. Наприклад, RNN можуть бути використані для аналізу навчальних звичок здобувачів та їх впливу на успішність.

Переваги та виклики застосування машинного навчання в освіті

Переваги:

Автоматизація: Машинне навчання дозволяє автоматизувати процес аналізу даних, що заощаджує час і ресурси [31]. Завдяки алгоритмам, що працюють у фоновому режимі, викладачі можуть зосередитися на навчанні, а не на обробці даних.

Персоналізація навчання: Моделі можуть адаптуватися до потреб кожного здобувача, пропонуючи індивідуальні рекомендації [32]. Це особливо важливо в умовах, коли здобувачі мають різний рівень підготовки та потребують різних підходів до навчання.

Прогнозування результатів: Використання алгоритмів машинного навчання допомагає прогнозувати результати навчання, що дозволяє вчасно вжити заходів для покращення успішності [33]. Наприклад, система може попереджати викладачів про здобувачів, які ризикують неуспішно скласти курс, що надає можливість провести додаткові консультації.

Виклики:

Якість даних: Для досягнення точних прогнозів потрібні якісні дані [34]. Неповні або неточні дані можуть негативно вплинути на результати. Це може бути особливо проблематичним у випадках, коли дані збираються з різних джерел.

Складність моделей: Деякі моделі можуть бути складними для інтерпретації, що ускладнює прийняття рішень на основі їхніх результатів [35]. Наприклад, результати нейронних мереж можуть бути важко зрозумілими, що може призводити до нерозуміння викладачами та адміністрацією навчальних закладів.

Етичні питання: Використання даних про здобувачів піднімає питання конфіденційності та етики, які необхідно враховувати під час розробки та впровадження систем [36]. Важливо забезпечити захист особистої інформації здобувачів та використовувати дані відповідно до етичних норм.

Вартість впровадження: Впровадження машинного навчання в освітні процеси може вимагати значних фінансових та людських ресурсів. Заклади освіти повинні бути готові інвестувати в навчання персоналу та розвиток технологій.

Застосування методів машинного навчання в аналізі освітніх даних відкриває нові можливості для прогнозування навчальної продуктивності. Вибір відповідних моделей та алгоритмів, а також усунення викликів можуть суттєво покращити результати навчання та підвищити ефективність освітнього процесу. Подальше дослідження в цій сфері дозволить розробити нові методи та підходи, що забезпечать ще більшу точність та ефективність прогнозування, що в свою чергу позитивно вплине на якість вищої освіти.

Висновки до розділу 1

У результаті виконаного аналізу було досліджено актуальність і значення аналізу навчальної продуктивності у вищій освіті. Встановлено, що моніторинг та оцінка успішності здобувачів є важливими елементами управління освітнім процесом, які впливають на якість навчання та прийняття рішень щодо його вдосконалення. Застосування інформаційних технологій дозволяє автоматизувати ці процеси, забезпечуючи збір, обробку та аналіз даних про академічні результати студентів.

Аналіз об'єкта дослідження показав, що процес оцінювання успішності є багатограним явищем, яке включає різні види контролю: поточний, проміжний та підсумковий. У сучасній освіті широко застосовуються автоматизовані системи управління навчальним процесом, такі як Moodle, Blackboard та Canvas. Вони забезпечують ефективне зберігання та обробку академічних даних, однак їхні можливості у прогнозуванні навчальної продуктивності залишаються обмеженими без інтеграції зі спеціалізованими аналітичними інструментами.

Розглянуто сучасні рішення для аналізу освітніх даних, зокрема платформи Alteryx та IBM Watson Analytics. Встановлено, що ці системи мають потужні засоби для обробки великих обсягів даних і прогнозування майбутніх результатів, однак їх впровадження потребує значних технічних ресурсів і підтримки.

Окремо досліджено методи машинного навчання для аналізу освітніх даних, серед яких кластеризація, метод головних компонент (PCA) та регресійний аналіз. Ці підходи дозволяють отримати нові інсайти з даних, виявляти закономірності та тенденції, а також підвищувати ефективність прийняття рішень у навчальному процесі.

На основі проведеного аналізу було сформульовано висновок про необхідність розробки програмного забезпечення, яке поєднуватиме функціонал для збору, аналізу та прогнозування навчальної продуктивності здобувачів. Така система має використовувати сучасні методи машинного навчання та забезпечувати зручний інтерфейс для викладачів і адміністраторів. Результати дослідження стануть основою для подальших етапів роботи, зокрема для розробки концепції системи та її реалізації.

2 МОДЕЛЮВАННЯ СИСТЕМИ ТА ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі буде розглянуто процес проєктування інформаційної системи, що базується на застосуванні методів машинного навчання для аналізу академічних даних. Особливу увагу буде приділено формалізації основних функцій системи, побудові математичних моделей, а також використанню CASE-засобів для візуалізації та проєктування. Кожен етап проєктування сприятиме створенню системи, що відповідатиме вимогам та очікуванням користувачів.

2.1 Аналіз сучасних методів і моделей для вирішення поставлених завдань

Для ефективного аналізу та прогнозування навчальної продуктивності здобувачів важливо обрати такі методи, які забезпечують високу точність результатів, враховуючи специфіку освітніх даних. Дані, які використовуються в процесі оцінювання, зазвичай включають академічні бали, інформацію про відвідуваність, виконання завдань та інші параметри, що впливають на загальну успішність. Серед методів, що використовуються в освітній аналітиці, можна виділити кластеризацію, регресійний аналіз, метод головних компонент (РСА), нейронні мережі та технології обробки великих даних.

Кластеризація є потужним методом для групування студентів за схожими характеристиками. Основна ідея полягає у тому, щоб об'єднати здобувачів у кластери на основі таких параметрів, як академічні бали, відвідуваність або прогрес у виконанні завдань.

Найбільш популярним алгоритмом є метод К-середніх, який визначає центри кластерів і розподіляє дані на основі мінімізації відстаней між точками даних і центрами кластерів. Цей метод простий у реалізації та дозволяє ефективно обробляти невеликі та середні обсяги даних [1].

Однак, одним із обмежень кластеризації є необхідність заздалегідь визначати кількість кластерів. Крім того, алгоритми кластеризації менш

ефективні для обробки складних багатовимірних даних, що може вплинути на точність результатів.

РСА є широко використовуваним методом для зменшення розмірності даних. Це особливо важливо для освітніх систем, де дані можуть включати десятки змінних, таких як результати з різних дисциплін, активність у курсах, участь у проєктах тощо. РСА дозволяє зменшити кількість змінних, зберігаючи найбільш значущі компоненти, які впливають на результати навчання [2].

Цей метод корисний для візуалізації даних, що робить його незамінним у процесі підготовки даних для подальшого аналізу. Основним недоліком РСА є те, що він може втратити частину інформації, яка не вписується в основні компоненти, але все ще може бути важливою для аналізу.

Регресійний аналіз використовується для виявлення залежностей між змінними та побудови прогнозів. У контексті освітньої аналітики регресія дозволяє передбачати майбутні результати здобувачів на основі їхніх попередніх досягнень.

Лінійна регресія є одним із найпростіших і найбільш інтерпретованих методів. Вона дозволяє знайти лінійні зв'язки між змінними, наприклад, як оцінки з математики впливають на успішність у програмуванні [3]. Для складніших залежностей використовуються методи поліноміальної або логістичної регресії.

Основні переваги регресійного аналізу – це простота реалізації та висока інтерпретованість. Проте методи регресії є менш ефективними у випадках, коли дані містять значну кількість шуму або нелінійні залежності.

Нейронні мережі є основою сучасних методів глибокого навчання. Вони здатні обробляти великі обсяги даних та враховувати складні взаємозв'язки між змінними. У контексті освітньої аналітики нейронні мережі можуть використовуватися для прогнозування успішності здобувачів, виявлення проблемних тем або навіть персоналізації навчального процесу [4].

Зокрема, рекурентні нейронні мережі (RNN) корисні для аналізу послідовних даних, таких як прогрес студента протягом семестру, а згорткові нейронні мережі (CNN) – для виявлення складних взаємозв'язків у багатовимірних даних. Попри свою потужність, нейронні мережі потребують значних обчислювальних ресурсів, що обмежує їх використання у закладах освіти із обмеженим бюджетом.

Освітні системи часто працюють із великими обсягами даних, які потребують масштабованих технологій для їхньої обробки. Використання платформ, таких як Apache Hadoop або Spark, дозволяє обробляти дані паралельно, забезпечуючи швидкість і ефективність [5].

Ці технології можуть застосовуватися для аналізу даних студентів у великих університетах, де кількість записів про успішність сягає мільйонів. Однак складність налаштування таких систем і необхідність технічної експертизи є суттєвими недоліками.

Порівняння методів

Для оцінки кожного методу доцільно порівняти їх за основними критеріями:

Таблиця 2.1 – Порівняння методів машинного навчання

Метод	Основна мета	Переваги	Недоліки
Кластеризація	Групування студентів	Простота реалізації	Необхідність задавати кількість кластерів
РСА	Зменшення розмірності	Збереження ключових змінних	Втрата частини інформації
Регресійний аналіз	Прогнозування	Висока інтерпретованість	Вразливість до шуму
Нейронні мережі	Аналіз складних даних	Висока точність	Високі обчислювальні витрати
Обробка великих даних	Масштабована аналітика	Швидкість роботи	Складність налаштування

Аналіз методів показав, що найефективніше для освітніх даних є комбінація кластеризації, PCA та регресійного аналізу. Вибір конкретного методу залежить від обсягу даних, складності зв'язків між змінними та потреб у точності результатів.

Вибір технологій та інструментів для розробки програмного забезпечення

Поряд із вибором алгоритмів машинного навчання, важливим є підбір технологій і інструментів для розробки програмного забезпечення. Вибір технологій має забезпечити легкість інтеграції, гнучкість, а також можливість подальшого масштабування системи.

Основними технологіями, що будуть використані в розробці, є:

– Python: Завдяки багатій бібліотеці для машинного навчання (наприклад, TensorFlow, Keras, Scikit-learn) Python є однією з найпопулярніших мов програмування в сфері даних. Ця мова забезпечує простоту написання коду і відмінну підтримку для роботи з великими обсягами даних.

– SQL: Використання SQL для управління базами даних дозволить ефективно зберігати, обробляти та отримувати академічні дані. SQL є стандартним мовним засобом для роботи з реляційними базами даних, що дозволяє швидко формувати запити до великих наборів даних.

– CASE-засоби: Використання CASE-засобів, таких як BPWin, ARIS або Lucidchart, дозволить створювати візуальні моделі процесів та даних. Це забезпечить наочність проектування і допоможе у спілкуванні між членами команди та зацікавленими сторонами.

CASE-засоби для моделювання інформаційних систем

CASE-засоби (Computer-Aided Software Engineering) є важливими інструментами для моделювання інформаційних систем, оскільки вони забезпечують можливість візуалізації, документування і автоматизації етапів розробки. Використання таких засобів дозволяє зменшити ризик помилок і підвищити ефективність проектування системи.

В рамках проектування системи для аналізу та прогнозування навчальної продуктивності будуть використовуватися такі нотації:

– IDEF3: Дана нотація призначена для моделювання процесів, що допомагає візуалізувати послідовність дій у системі та взаємозв'язки між ними. Використання IDEF3 сприяє створенню зрозумілих моделей, які легко інтерпретуються.

– DFD (Data Flow Diagram): Діаграми потоків даних дозволяють візуалізувати, як дані переміщуються через систему, і які процеси їх обробляють. DFD допоможе зрозуміти, як академічні дані будуть вводитися в систему, оброблятися і використовуватися для прогнозування.

Таким чином, вибір методів та підходів для проектування системи є критично важливим етапом, що закладає основи для подальшої розробки. Обґрунтоване використання алгоритмів машинного навчання, технологій та CASE-засобів дозволить створити ефективну та надійну систему аналізу та прогнозування навчальної продуктивності, яка відповідатиме вимогам сучасних освітніх установ.

2.2 Моделювання об'єкту і предмету дослідження

Моделювання об'єкту та предмету дослідження є ключовим етапом у розробці програмного забезпечення, що забезпечує чітке визначення компонентів системи, їхніх взаємозв'язків і способів взаємодії. У контексті аналізу та прогнозування навчальної продуктивності основною метою моделювання є створення функціональної структури, яка відображає всі аспекти процесу оцінювання та прогнозування успішності здобувачів.

Загальний підхід до моделювання

Моделювання виконується на основі використання таких інструментів:

- **Функціональні моделі** для опису основних процесів у системі.
- **Інформаційні моделі** для визначення потоків даних і взаємозв'язків між елементами системи.

– **Графічні нотації** (IDEF0, DFD, UML), які забезпечують візуалізацію моделей і спрощують розуміння системи.

Ці моделі дозволяють формалізувати об'єкт та предмет дослідження, розділити систему на логічні блоки та визначити їхню функціональність.

Контекстна діаграма

Контекстна діаграма (рис.2.1) є найвищим рівнем абстракції функціональної моделі об'єкта дослідження. Вона відображає систему як єдине ціле, яке взаємодіє із зовнішніми сутностями через потоки даних.



Рисунок 2.1 – Контекстна діаграма IDEF0

У даній контекстній діаграмі система аналізу та прогнозування навчальної продуктивності взаємодіє з такими зовнішніми сутностями:

Викладачі – завантажують дані про оцінки студентів, отримують результати аналізу та прогнозування.

Студенти – надають свої навчальні дані, отримують рекомендації та звіти.

Освітні платформи (наприклад, Moodle) – постачають структуровані дані про оцінки студентів.

Адміністратори – забезпечують управління системою, контролюють коректність даних і функціональність.

Система отримує вхідні дані, виконує їх обробку, аналіз і прогнозування, а також надає результати у вигляді звітів і візуалізацій.

Інформаційна модель предмету дослідження

Інформаційна модель описує основні потоки даних у системі. На діаграмі потоків даних (DFD) зображено основні процеси та взаємодії системи для аналізу та прогнозування навчальної продуктивності (рис 2.2).



Рисунок 2.2 – Data Flow Diagram 0 рівня

Основні компоненти діаграми:

Зовнішні сутності:

- *Excel File*: Джерело даних, яке містить академічні бали та дані про здобувачів.
- *Moodle System*: Система управління навчанням, що надає навчальну статистику.
- *Викладачі*: Користувачі системи, які отримують результати аналізу та прогнозування.

Система приймає інформацію про здобувачів та їх навчальну продуктивність, результатом діяльності системи є Статистика та прогнозування, доступ до яких надається Викладачам.

На діаграмі потоків даних (DFD) зображено основні процеси та взаємодії системи для аналізу та прогнозування навчальної продуктивності (рис. 2.2).

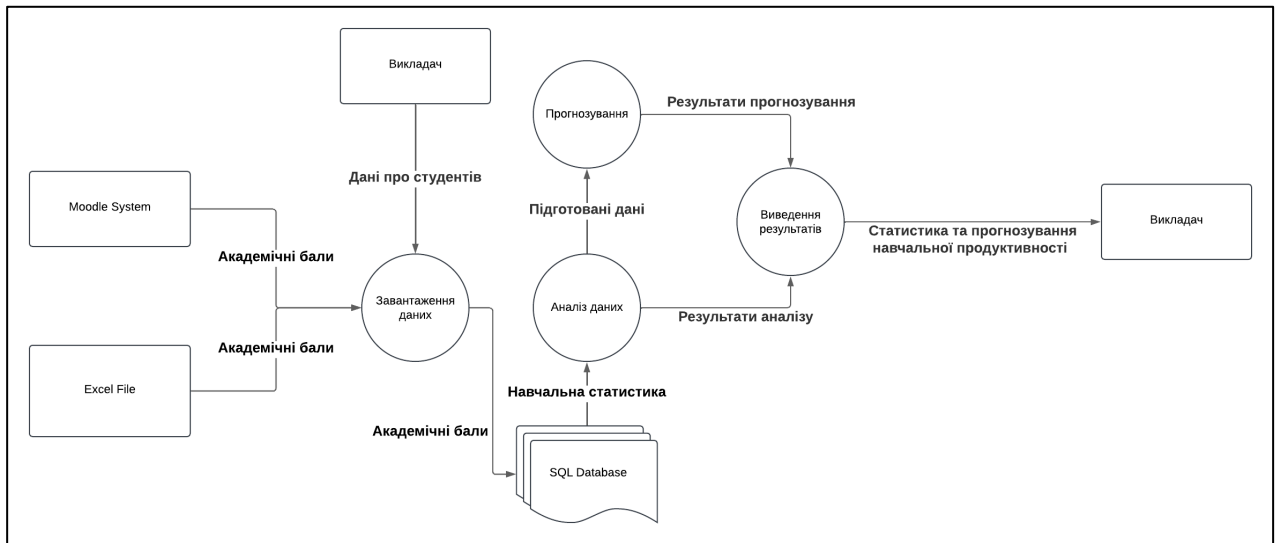


Рисунок 3.2 – Data Flow Diagram 1 рівня

Основні компоненти цієї діаграми:

Зовнішні сутності:

- *Excel File*: Джерело даних, яке містить академічні бали та дані про здобувачів.
- *Moodle System*: Система управління навчанням, що надає навчальну статистику.
- *Викладачі*: Користувачі системи, які отримують результати аналізу та прогнозування.

Процеси:

Завантаження даних:

Вхідні дані: Академічні бали, Дані про здобувачів.

Вихідні дані: Підготовані дані для збереження.

Аналіз даних:

Вхідні дані: Навчальна статистика.

Вихідні дані: Результати аналізу, Підготовані дані.

Прогнозування:

Вхідні дані: Підготовані дані.

Вихідні дані: Результати прогнозування.

Виведення результатів:

Вхідні дані: Результати прогнозування, Результати аналізу.

Вихідні дані: Статистика та прогнозування навчальної продуктивності.

Сховище даних:

SQL Database: місце, де зберігаються підготовлені дані, результати аналізу та прогнозування. Ця база даних слугує центральним сховищем для всіх даних, які використовуються в процесах.

Потоки даних:

Академічні бали: передаються від Excel File та Moodle System у процес «Завантаження даних».

Дані про здобувачів: передаються від викладачів у процес «Завантаження даних».

Навчальна статистика: передається від SQL Database у процес «Аналіз даних».

Підготовлені дані: вихідні дані з процесу «Аналіз даних», які використовуються у «Прогнозуванні».

Результати прогнозування: вихідні дані з процесу «Прогнозування», які передаються у «Виведення результатів».

Результати аналізу: вихідні дані з процесу «Аналіз даних», передаються у «Виведення результатів».

Статистика та прогнозування навчальної продуктивності: передається від процесу «Аналіз даних» та процесу «Прогнозування» до викладачів.

2.3 Формування специфікації вимог до програмного забезпечення

Розробка системи аналізу та прогнозування навчальної продуктивності вимагає чіткого формулювання вимог, які дозволять створити ефективний інструмент для освітніх установ. Специфікація вимог (англ. Software

Requirements Specification – SRS) містить опис функціональних та нефункціональних аспектів системи, а також системних вимог, які забезпечать її ефективність і безперебійну роботу.

1) Призначення та межі проєкту:

1.1) Призначення системи:

Розробка програмного забезпечення має на меті створення інтерактивної системи аналізу та прогнозування навчальної продуктивності на основі академічних балів. Ця система дозволить викладачам та адміністраторам навчальних закладів ефективно моніторити досягнення студентів, виявляти тенденції в навчанні, а також формувати рекомендації для підвищення успішності. Вона також забезпечить студентам зрозуміле представлення їхніх оцінок та прогресу.

1.2) Погодження:

У програмній документації будуть зазначені всі погодження, що стосуються функціональних можливостей системи, включаючи:

- визначення ролей користувачів (викладачі, студенти, адміністрація);
- перелік ключових функцій, які повинні бути реалізовані (введення та обробка оцінок, формування звітів, прогнозування результатів);
- принципи взаємодії між модулями системи, а також з зовнішніми сервісами, якщо такі передбачаються.

1.3) Межі проєкту ПЗ:

Проєкт не включає:

- інтеграцію з зовнішніми освітніми платформами чи базами даних.
- розробку мобільних додатків; проєкт зосереджений на веб-інтерфейсі.
- функціональність, пов'язану з фінансовим обліком або управлінням стипендіями.
- Обробку даних, що не стосуються навчальної продуктивності (наприклад, особисті дані, не пов'язані з навчанням).

2) Загальний опис

2.1) Сфера застосування:

Система призначена для використання в загальноосвітніх навчальних закладах, вищих навчальних закладах (університетах, коледжах) та професійно-технічних навчальних закладах. Вона забезпечує викладачів, студентів та адміністраторів інструментами для аналізу навчальних досягнень, що сприяє підвищенню ефективності навчального процесу.

2.2) Характеристики користувачів:

Користувачами системи будуть:

- викладачі: які ввели оцінки, можуть переглядати статистику успішності своїх студентів та отримувати рекомендації щодо поліпшення навчального процесу.

- студенти: які можуть отримувати доступ до своїх оцінок, звітів про навчальні досягнення та рекомендацій для покращення результатів.

- адміністрація: яка буде контролювати загальну продуктивність студентів, а також мати доступ до аналітичних звітів для прийняття управлінських рішень.

2.3) Загальна структура і склад системи:

Система складається з кількох основних модулів:

- модуль вводу даних: для введення академічних балів та особистих даних студентів;

- аналітичний модуль: для обробки даних і генерації звітів щодо навчальної продуктивності;

- модуль візуалізації: для графічного представлення результатів, включаючи графіки та діаграми.

2.4) Загальні обмеження:

- система не повинна перевищувати максимальної кількості одночасних користувачів, що можуть входити в систему (наприклад, 200 одночасних сесій);

- всі дані повинні зберігатися у базі даних у форматі, що відповідає стандартам безпеки, з регулярним резервним копіюванням;

– система повинна мати здатність обробляти дані не більше ніж за 24 години після їх введення.

3) Функції системи

3.1) Введення даних:

– функція вводу оцінок: Викладачі можуть вводити оцінки за предметами для кожного студента через інтуїтивно зрозумілий інтерфейс. Наприклад, система підтримує імпорт даних з Excel для зручності;

– управління інформацією про студентів: Додання, редагування та видалення профілів студентів, включаючи особисту інформацію (ім'я, прізвище, група).

3.2) Аналіз даних:

– генерація звітів: автоматичне створення звітів про успішність студентів за семестр, включаючи середній бал, розподіл оцінок;

– прогнозування успішності: використання алгоритмів машинного навчання для оцінки ймовірності успішності студентів на основі їхніх попередніх результатів та відвідуваності.

3.3) Візуалізація інформації:

– графіки та діаграми: візуалізація даних у формі графіків, що показують динаміку успішності студентів, порівняння між групами та предметами;

– особисті кабінети: студенти можуть візуалізувати свої досягнення, отримувати графіки зростання оцінок з часом.

3.4) Зворотний зв'язок:

– оцінка викладачів: студенти можуть залишати анонімні відгуки про викладачів та курс, що допоможе в покращенні навчального процесу;

– система рекомендацій: на основі результатів студентів, система буде пропонувати індивідуальні курси або додаткові матеріали для покращення знань.

3.5) Адміністративні функції:

– моніторинг активності: адміністратори можуть стежити за активністю користувачів, а також за якістю введених даних;

– налаштування прав доступу: Визначення рівнів доступу для різних користувачів (викладачі, студенти, адміністратори).

4) Вимоги до інформаційного забезпечення

4.1) Програмне забезпечення:

– операційна система: система повинна бути сумісною з Windows, Linux та macOS, щоб забезпечити доступність для всіх користувачів;

– бази даних: використання реляційних баз даних, таких як MySQL або PostgreSQL, для зберігання даних про студентів, оцінки та статистику;

– фреймворки: рекомендується використовувати фреймворки для розробки веб-додатків, такі як Django (Python) або Laravel (PHP), що забезпечать швидкість і безпеку.

4.2) Апаратне забезпечення:

– сервер: високопродуктивний сервер з процесором не менше ніж 8 ядер, 16 ГБ оперативної пам'яті та SSD-накопичувачами для швидкого доступу до даних;

– клієнтські пристрої: комп'ютери, ноутбуки або планшети з мінімум 4 ГБ оперативної пам'яті і сучасними веб-браузерами (Chrome, Firefox, Safari).

4.3) Мережеві вимоги:

– широкосмуговий Інтернет: стабільне з'єднання з Інтернетом зі швидкістю не менше 10 Мбіт/с для ефективної роботи системи;

– безпека мережі: Використання VPN для захисту даних під час передачі та реалізація шифрування (HTTPS) для веб-додатку.

4.4) Вимоги до даних:

– структура даних: структуровані дані з чітким визначенням полів (наприклад, ім'я студента, предмет, оцінка) для спрощення аналітики;

– регулярне резервне копіювання: автоматизоване резервне копіювання даних не рідше ніж раз на добу для запобігання втратам.

4.5) Технічна підтримка:

- доступність технічної підтримки: користувачі повинні мати можливість звертатися за технічною допомогою через електронну пошту, чат або телефон;
- документація: надання повної документації для користувачів, включаючи посібники з використання, налаштування та усунення неполадок.

5) Вимоги до технічного забезпечення

5.1) Серверне обладнання:

- процесор: мультипроцесорна система з підтримкою не менше ніж 8 ядер (наприклад, Intel Xeon або AMD Ryzen) для обробки великої кількості запитів одночасно;
- оперативна пам'ять: Мінімум 32 ГБ RAM для забезпечення стабільної роботи сервера під навантаженням;
- накопичувач: SSD-накопичувачі ємністю від 1 ТБ з технологією RAID для підвищення надійності та швидкості доступу до даних.

5.2) Клієнтське обладнання:

- персональні комп'ютери: з мінімальними характеристиками: процесор не нижче ніж Intel Core i3, 8 ГБ оперативної пам'яті, 256 ГБ SSD;
- мобільні пристрої: ноутбуки або планшети з можливістю запуску сучасних веб-браузерів для забезпечення доступу до системи з будь-якого місця.

5.3) Мережева інфраструктура:

- маршрутизатори та комутатори: високошвидкісні маршрутизатори з підтримкою Gigabit Ethernet для забезпечення швидкого та стабільного з'єднання в локальній мережі;
- бездротові точки доступу: Wi-Fi 6 точки доступу для забезпечення стабільного бездротового з'єднання для всіх користувачів у навчальному закладі.

5.4) Системи безпеки:

- антивірусне програмне забезпечення: встановлення антивірусних рішень на сервері та клієнтських пристроях для запобігання вірусним атакам;
- міжмережеві екрани (файрволи): впровадження програмних та апаратних файрволів для захисту системи від несанкціонованого доступу.

5.5) Енергетичне забезпечення:

- джерела безперебійного живлення (ДБЖ): установка ДБЖ для запобігання втратам даних у випадку збоїв у електромережі.
- системи охолодження: системи кондиціонування для підтримки оптимальної температури в серверному приміщенні.

б) Вимоги до програмного забезпечення

6.1) Основне програмне забезпечення:

- операційні системи: підтримка Windows Server та Linux для серверної частини. Клієнтські ПК повинні працювати на Windows 10 або новішій версії, macOS або Linux.
- веб-сервер: використання Apache або Nginx для хостингу веб-додатків, з налаштуванням для оптимізації швидкості завантаження.

6.2) Розробка додатків:

- мови програмування: використання Python або PHP для серверного програмування та JavaScript для фронтенду, що дозволить створити динамічні та інтерактивні веб-інтерфейси;
- фреймворки: рекомендовані фреймворки: Django (Python) для швидкої розробки та Laravel (PHP) для зручності управління базами даних.

6.3) Бази даних:

- системи управління базами даних: використання реляційних СУБД, таких як MySQL або PostgreSQL для зберігання інформації про студентів, курси та оцінки;
- безпека даних: реалізація механізмів шифрування для захисту конфіденційної інформації, таких як паролі та особисті дані.

6.4) Інтерфейс користувача:

- UX/UI дизайн: створення простого та інтуїтивно зрозумілого інтерфейсу, який забезпечить легкий доступ до функцій системи. Наприклад, використання адаптивного дизайну для мобільних пристроїв;

– мультимовність: Підтримка кількох мов (наприклад, українська та англійська) для зручності користувачів.

6.5) Інтеграції:

– API: Розробка RESTful API для інтеграції з іншими системами, такими як електронні бібліотеки або платформи для дистанційного навчання;

– системи звітності: Інтеграція з BI-інструментами, такими як Tableau або Power BI, для аналізу даних та візуалізації результатів.

6.6) Тестування та підтримка:

– автоматизоване тестування: впровадження юніт-тестів та інтеграційних тестів для забезпечення стабільності програмних рішень;

– документація: Надання повної технічної документації для розробників та користувачів, включаючи API-документацію та FAQ.

7) Вимоги до зовнішніх інтерфейсів

7.1) Інтерфейс користувача (UI):

– дизайн: інтерфейс повинен бути простим, зрозумілим та естетично привабливим, з використанням сучасних UX/UI принципів. Наприклад, використання кольорової палітри, що відповідає бренду навчального закладу, та зручних шрифтів;

– адаптивність: інтерфейс має бути адаптивним, щоб забезпечити коректне відображення на мобільних пристроях, планшетах і десктопах.

7.2) Зовнішні API:

– взаємодія з іншими системами: інтерфейс повинен підтримувати інтеграцію з іншими освітніми платформами через API (наприклад, API для платформи Zoom для онлайн-занять);

– стандарти: використання RESTful API з JSON для обміну даними та чітко визначеними кінцевими точками (endpoints) для основних функцій, таких як реєстрація студентів, отримання списку курсів тощо.

7.3) Системи аутентифікації:

- OAuth 2.0: реалізація механізму аутентифікації через OAuth 2.0 для забезпечення безпечного доступу до системи з можливістю входу через популярні сервіси (наприклад, Google або Facebook);
- двофакторна аутентифікація: підтримка двофакторної аутентифікації для підвищення рівня безпеки при доступі до акаунтів користувачів.

7.4) Інтеграція з платіжними системами:

- онлайн-платежі: реалізація інтеграції з платіжними шлюзами, такими як Stripe або PayPal, для забезпечення можливості онлайн-оплати за навчання чи інші послуги;
- безпека транзакцій: використання шифрування для захисту даних користувачів під час фінансових операцій.

7.5) Доступність:

- WCAG: дотримання стандартів веб-доступності WCAG (Web Content Accessibility Guidelines) для забезпечення можливості використання системи людьми з обмеженими можливостями;
- тестування доступності: регулярне тестування інтерфейсу на доступність за допомогою автоматизованих інструментів (наприклад, WAVE) та реальних користувачів.

7.6) Документація API:

- опис API: забезпечення докладної документації для розробників, включаючи приклади запитів та відповідей, щоб спростити інтеграцію з іншими системами;
- API Explorer: розробка інструменту API Explorer для тестування запитів у режимі реального часу, що допоможе розробникам швидше адаптуватися до використання API.

8) Властивості програмного забезпечення

8.1) Функціональність:

- основні можливості: програмне забезпечення повинно забезпечити можливість управління курсами, реєстрацію студентів, ведення електронного

журналу та створення звітів. Наприклад, викладач повинен мати можливість завантажувати матеріали курсу та призначати домашні завдання;

– інтерфейс API: забезпечення доступу до функцій системи через API, що дозволяє інтеграції з іншими зовнішніми системами.

8.2) Надійність:

– стійкість до збоїв: програмне забезпечення повинно мати механізми резервного копіювання та відновлення даних, щоб зменшити ризик втрати інформації у випадку збоїв;

– тестування: регулярне проведення системного тестування для виявлення та усунення помилок перед випуском нових версій.

8.3) Портативність:

– багатоплатформність: програмне забезпечення повинно працювати на різних операційних системах (Windows, Linux, macOS) без значних змін у кодї;

– мобільні платформи: забезпечення доступу до системи через мобільні додатки, що підтримують Android та iOS.

8.4) Виконуваність:

– час реакції: веб-додаток повинен забезпечувати швидке завантаження сторінок (не більше 2 секунд), що підвищує задоволеність користувачів;

– оптимізація швидкості: використання кешування та оптимізації запитів до бази даних для покращення загальної продуктивності системи.

8.5) Зручність у використанні:

– інтуїтивний інтерфейс: інтерфейс повинен бути простим та зрозумілим, що дозволяє користувачам легко знаходити потрібні функції без додаткових інструкцій;

– підтримка: надання користувачам доступу до навчальних матеріалів та FAQ для швидкого освоєння системи.

8.6) Безпека:

– захист даних: реалізація шифрування даних користувачів та механізмів захисту від SQL-ін'єкцій та крос-сайтових скриптів (XSS);

– регулярне оновлення: підтримка актуальності програмного забезпечення шляхом регулярних оновлень безпеки та патчів.

8.7) Підтримка та супровід:

– документація: наявність детальної документації для користувачів та розробників, яка охоплює всі аспекти функціонування системи.

– технічна підтримка: забезпечення служби підтримки для користувачів, щоб оперативно вирішувати питання та проблеми.

2.4 Математичне забезпечення системи

Математичне забезпечення є критичним елементом для створення ефективної системи аналізу та прогнозування навчальної продуктивності здобувачів. У цьому підрозділі буде представлено основні математичні моделі та алгоритми, що застосовуються для обробки даних і побудови прогнозів на основі академічних балів. Для вирішення завдань системи використовуються методи машинного навчання, які дозволяють аналізувати великі обсяги даних і виявляти закономірності в навчальній продуктивності.

Для прогнозування майбутньої успішності здобувачів може бути використана **лінійна регресія**. Це один з базових підходів, що дозволяє оцінити залежність майбутніх результатів здобувача від попередніх академічних показників. Формула лінійної регресії виглядає наступним чином:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (1)$$

де:

- y – прогнозований результат (майбутній академічний бал),
- x_1, x_2, \dots, x_n – незалежні змінні (поточні академічні результати здобувача),
- $\beta_0, \beta_1, \dots, \beta_n$ – коефіцієнти регресії.

Ця модель допомагає передбачити майбутні результати на основі минулих показників, таких як бали за контрольні роботи, тести та інші завдання.

Окрім лінійної регресії, доцільно застосувати **дерева рішень**, що є більш складною моделлю для прогнозування, яка дозволяє враховувати різні варіанти

розвитку подій і взаємозв'язки між ними. Древа рішень можуть враховувати нелінійні залежності між різними змінними та надавати більш точні прогнози.

Основний процес роботи з деревами рішень включає:

1. Розбиття даних на підгрупи на основі певних критеріїв (наприклад, успішність на попередніх етапах навчання).
2. Кожне рішення в дереві представляє вибір, який залежить від значень змінних.
3. На кожному етапі модель вибирає шлях, що найбільше відповідає історичним даним здобувача.

Нейронні мережі – це один із найпотужніших методів машинного навчання, що дозволяє будувати прогнози на основі великого обсягу даних із високою точністю. Для аналізу академічної продуктивності може бути використана багат шарова нейронна мережа, яка здатна розпізнавати складні взаємозв'язки між даними.

Типова нейронна мережа складається з:

- **вхідного шару** – отримує дані (поточні бали здобувачів),
- **прихованих шарів** – обробляють дані за допомогою функцій активації,
- **вихідного шару** – видає прогнозований результат.

Функція активації, що використовується у нейронних мережах, допомагає моделі краще враховувати нелінійні залежності між змінними, що є корисним для прогнозування навчальних результатів.

Метрики оцінки якості моделей

Для оцінки якості прогнозів, створених за допомогою зазначених моделей, використовуються наступні метрики:

1. Середньоквадратична помилка (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^{\circ})^2 \quad (2)$$

де y_i – реальний результат, а y_i° – прогнозований результат.

2. R-квадрат (коефіцієнт детермінації):

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

де SS_{res} – сума квадратів залишків, а SS_{tot} – загальна сума квадратів. Ця метрика показує, наскільки добре модель описує дані.

3. Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^o| \quad (4)$$

Ця метрика оцінює середню абсолютну помилку між реальними та прогнозованими значеннями, надаючи уявлення про точність моделі.

Алгоритм обробки даних

Алгоритм обробки даних складається з наступних кроків:

1. **Завантаження даних** – отримання даних із зовнішніх джерел (Excel або Moodle).

2. **Передобробка даних** – перевірка наявних даних на коректність, видалення помилкових записів та нормалізація.

3. **Навчання моделі** – застосування лінійної регресії, дерев рішень або нейронних мереж для аналізу попередніх результатів.

4. **Прогнозування** – побудова прогнозів на основі отриманих моделей.

5. **Оцінка якості** – використання метрик для перевірки точності прогнозів.

Ці алгоритми та методи допоможуть створити ефективну систему аналізу й прогнозування навчальної продуктивності здобувачів.

У цьому підрозділі було представлено математичні моделі, які можуть бути застосовані для аналізу даних і прогнозування успішності здобувачів. Лінійна регресія, дерева рішень та нейронні мережі є основними інструментами, що забезпечують точне прогнозування на основі минулих результатів здобувачів. Метрики, такі як MSE та R-квадрат, дозволяють оцінювати якість моделей та вдосконалювати алгоритми для досягнення кращих результатів.

2.5 Алгоритми роботи системи

Алгоритми є основою функціонування інформаційної системи, що забезпечує автоматизацію процесів збору, обробки, аналізу та прогнозування навчальної продуктивності. У цій інформаційній системі алгоритми виконують

ключові функції, включаючи збір даних, їх обробку, аналіз, прогнозування, генерацію звітів, а також забезпечення зворотного зв'язку. У цьому підрозділі ми розглянемо основні алгоритми роботи системи, які будуть реалізовані в розроблюваній інформаційній системі.

Збір даних є першим кроком у процесі аналізу навчальної продуктивності. Алгоритм збору даних передбачає кілька етапів, які включають різні джерела та методи:

- Ідентифікація джерел даних: Це можуть бути внутрішні системи навчального закладу, такі як електронні журнали успішності, системи управління навчанням (LMS), а також зовнішні джерела, такі як анкетування здобувачів.

- Автоматизація збору даних: Реалізація механізму для автоматичного збору даних із зазначених джерел. Це може включати API для інтеграції з LMS або безпосередній доступ до бази даних.

- Перевірка якості даних: Важливим етапом є перевірка зібраних даних на коректність і повноту. На цьому етапі можуть використовуватися алгоритми виявлення аномалій, які виявляють помилки або пропуски у даних.

- Класифікація даних: Визначення типу даних (числові, категоріальні) для подальшої обробки. Це необхідно для вибору відповідних алгоритмів аналізу.

- Зберігання даних: Зібрані дані зберігаються в базі даних, що дозволяє швидко їх обробляти та аналізувати.

На етапі аналізу даних використовуються різноманітні алгоритми машинного навчання, які допомагають виявити закономірності в навчальних даних. Алгоритм аналізу даних може бути розбитий на такі ключові етапи:

- Обробка даних: Перед подачею даних до алгоритмів машинного навчання важливо виконати їх обробку. Це може включати нормалізацію, видалення дублікатів та обробку пропущених значень.

- Вибір моделі: На цьому етапі визначається, які алгоритми машинного навчання будуть використовуватися для аналізу даних. Можна застосувати:

- лінійну регресію для прогнозування успішності на основі історичних даних;
- дерева рішень для прийняття рішень на основі певних умов;
- нейронні мережі для більш складних задач, коли існує значна кількість вхідних даних з нелінійними залежностями.
- Навчання моделей: Використовуючи підготовлені дані, моделі навчаються для виявлення залежностей і закономірностей.
- Валідація моделей: Після навчання моделей важливо оцінити їх ефективність, використовуючи метрики, такі як середня абсолютна похибка (MAE) або R^2 .
- Аналіз результатів: Вивчення результатів, отриманих від моделей, з метою виявлення сильних і слабких сторін.

Після завершення етапу аналізу наступним кроком є прогнозування навчальної продуктивності. Алгоритм прогнозування включає в себе:

- Використання навченої моделі: Застосування раніше навчених моделей для прогнозування успішності здобувачів на основі нових вхідних даних.
- Аналіз прогнозів: Оцінка точності прогнозів, порівняння їх із фактичними результатами успішності. Для цього можуть використовуватися графіки, які ілюструють, як прогнози співвідносяться з реальними даними.
- Корекція прогнозів: У разі виявлення суттєвих розбіжностей між прогнозами та фактичними даними можуть бути внесені корективи в параметри моделей або самі моделі.
- Складання рекомендацій: На основі отриманих прогнозів формуються рекомендації для здобувачів і викладачів.
- Зворотний зв'язок: Збір відгуків про точність прогнозів для подальшого удосконалення моделей.

Останнім етапом роботи системи є генерація звітів, які надають корисну інформацію про результати аналізу та прогнозування. Алгоритм генерації звітів складається з:

Формування звітів: Генерація звітів на основі отриманих результатів, які містять аналіз успішності здобувачів та рекомендації.

Візуалізація даних: Використання графіків і діаграм для наочного представлення даних. Візуалізація допомагає швидше зрозуміти інформацію, яка міститься в звітах.

– Розповсюдження звітів: Налагодження механізму автоматичної розсилки звітів для здобувачів, викладачів та адміністрації університету.

– Зворотній зв'язок: Отримання відгуків від користувачів системи про корисність звітів.

– Аналіз ефективності: Оцінка того, як звіти впливають на навчальний процес.

Розроблені алгоритми для збору, аналізу, прогнозування та генерації звітів формують основну архітектуру інформаційної системи. Чітка структура алгоритмів і їх логічна послідовність забезпечують ефективність і точність системи. Застосування машинного навчання дозволяє адаптуватися до змінюваних умов навчального процесу, надаючи цінні інсайти для поліпшення навчальних результатів здобувачів.

2.6 Інструментальні засоби та технології для реалізації

Для розробки інформаційної системи для аналізу та прогнозування навчальної продуктивності обрано такі середовища:

– Jupyter Notebook: Це інтерактивне середовище дозволяє зручно писати і виконувати код на Python, що ідеально підходить для прототипування алгоритмів машинного навчання та обробки даних. Jupyter надає можливість візуалізувати результати обробки даних у реальному часі.

– Visual Studio Code (VSCode): Обрано це середовище розробки через його потужні інструменти для налагодження, підтримки плагінів та інтеграції з системами контролю версій. VSCode також має розширення для Python, що спрощує написання та тестування коду.

У розробці інформаційної системи вирішено використовувати наступні мови програмування та технології:

– Python: Ця мова обрана через свою популярність у галузі даних і машинного навчання. Python має багатий набір бібліотек, таких як Pandas, NumPy, scikit-learn та TensorFlow, які дозволяють ефективно обробляти дані та створювати моделі машинного навчання.

– SQL: Використання мови SQL для взаємодії з реляційними базами даних є критично важливим для збору, зберігання та обробки навчальних даних. SQL надає можливість здійснювати запити до бази даних, що є важливим елементом в аналізі академічних балів.

Для моделювання та візуалізації процесів системи обрано:

– BPWin: Цей інструмент використовується для моделювання бізнес-процесів, що дозволяє створювати діаграми, які ілюструють робочі процеси та послідовності дій у системі. Використання BPWin забезпечить точність моделювання і полегшить розуміння складних процесів.

Однією з основних вимог до інформаційної системи є можливість інтеграції з існуючими системами. Це дозволяє використовувати наявні дані та функціональність без необхідності повної реорганізації. Для інтеграції було обрано:

– RESTful API: Цей архітектурний стиль дозволяє системам взаємодіяти між собою через стандартні HTTP-запити, що забезпечує простий доступ до функціональності інших систем та можливість обміну даними.

На завершальному етапі розробки інформаційної системи важливо провести тестування та верифікацію програмного забезпечення. Це включає:

– Юніт-тестування: Обрано проведення юніт-тестування для перевірки окремих модулів програми на коректність роботи. Інструменти, такі як pytest для Python, можуть бути використані для автоматизації цього процесу.

- Інтеграційне тестування: Буде проведено інтеграційне тестування для перевірки взаємодії між різними модулями системи, щоб забезпечити їх правильну роботу в комплексі.
- Системне тестування: Оцінка системи в цілому для перевірки, чи відповідає вона вимогам специфікації.
- Тестування зручності використання: Обрано провести оцінку інтерфейсу системи з точки зору користувача для виявлення можливих проблем у навігації та функціональності.
- Випробування на продуктивність: Оцінка швидкості та надійності системи під навантаженням.

Вибір інструментів і технологій для розробки інформаційної системи є критично важливим для успішної реалізації проекту. Поєднання потужних мов програмування, таких як Python і SQL, з візуальними CASE-засобами для моделювання, як BPWin, дозволяє забезпечити ефективність проектування та реалізації. Інтеграція з іншими системами і проведення тестування гарантують, що кінцевий продукт буде якісним, надійним і відповідатиме вимогам користувачів.

Висновки до розділу 2

У другому розділі було проведено всебічне моделювання предметної області та визначено ключові аспекти, що формують основу для створення ефективного програмного забезпечення. Розглянуто сучасні методи і моделі для аналізу навчальної продуктивності, що дозволило виявити найдоцільніші підходи для вирішення поставлених завдань.

На етапі вибору методів та підходів було обрано сучасні алгоритми машинного навчання, які найкраще підходять для аналізу академічних даних. Ці методи забезпечують точність прогнозування навчальної продуктивності, що є ключовим для вирішення поставлених завдань.

Аналіз методів, таких як кластеризація, метод головних компонент (РСА), регресійний аналіз та нейронні мережі, показав їх ефективність для обробки освітніх даних і прогнозування результатів. На основі проведеного аналізу обґрунтовано використання комбінації кластеризації та регресійного аналізу для забезпечення високої точності та обчислювальної ефективності.

У підрозділі моделювання об'єкту і предмету дослідження було створено функціональні та інформаційні моделі, що деталізують процеси обробки, аналізу та прогнозування даних. Контекстна діаграма, побудована в рамках моделювання, забезпечила чітке розуміння взаємодії системи із зовнішніми сутностями, такими як викладачі, студенти та освітні платформи.

У рамках математичного забезпечення системи була формалізована основна функціональність, що дозволяє здійснювати аналіз академічних балів та прогнозування навчальної продуктивності. Обрані алгоритми, такі як лінійна регресія, дерева рішень і нейронні мережі, забезпечують гнучкість і точність у реалізації прогнозів.

На основі проведених досліджень було сформовано специфікацію вимог до програмного забезпечення, яка визначає його функціональні можливості, архітектуру, вимоги до даних і зовнішніх інтерфейсів. Ця специфікація забезпечує базу для подальшого проєктування та реалізації системи.

Окрему увагу було приділено математичному забезпеченню системи, яке включає алгоритми аналізу даних, побудови прогнозів та кластеризації. Визначено основні алгоритми роботи системи, які забезпечують її функціональність та інтеграцію з іншими інструментами.

Проведено аналіз інструментальних засобів та технологій, що будуть використані для реалізації програмного забезпечення. Обґрунтовано вибір Python як основної мови програмування та рекомендованих бібліотек для роботи з даними, машинного навчання та візуалізації.

3. РОЗРОБКА АРХІТЕКТУРИ ТА СТРУКТУРНИЙ ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі детально розглянуто процес проектування програмного забезпечення для аналізу та прогнозування навчальної продуктивності здобувачів вищої освіти. Описано архітектуру системи, UML-діаграми, що ілюструють структуру й логіку роботи компонентів, а також вибрані технології для реалізації. Розробка програмного забезпечення базується на принципах модульності, масштабованості та зручності використання, що забезпечує адаптацію до потреб користувачів.

3.1 Розробка архітектури програмного забезпечення

Розробка архітектури програмного забезпечення є важливим етапом у створенні системи для аналізу та прогнозування навчальної продуктивності. Архітектура забезпечує основу для функціонування всіх компонентів системи, визначає їхню взаємодію та задає принципи розробки.

Загальна концепція архітектури

Розроблювана система має багаторівневу архітектуру, що забезпечує модульність, гнучкість та масштабованість. Основними рівнями системи є:

1. Рівень даних

Цей рівень відповідає за зберігання та управління даними. Він забезпечує:

- інтеграцію з джерелами даних (наприклад, Moodle або Excel);
- збереження даних у реляційній базі даних (наприклад, SQLite або PostgreSQL);
- можливість виконання CRUD-операцій (створення, читання, оновлення, видалення).

2. Логічний рівень (рівень обробки)

Логічний рівень містить основні алгоритми обробки даних та логіку бізнес-процесів. Основні функції цього рівня:

- очищення даних та перевірка на відповідність;

- обчислення статистичних характеристик;
- побудова прогнозів за допомогою методів машинного навчання;
- формування аналітичних звітів.

3. Презентаційний рівень

Цей рівень відповідає за взаємодію користувачів із системою. Інтерфейс реалізовано у вигляді інтерактивного середовища Jupyter Notebook, яке дозволяє виконувати аналіз, будувати графіки, переглядати результати та експортувати їх у різні формати.

Основні компоненти архітектури

Система складається з кількох компонентів, які забезпечують її функціональність:

1. DataManager

- Завантажує дані із зовнішніх джерел (CSV, Excel, Moodle).
- Забезпечує очищення та збереження даних.
- Підтримує структуру таблиць бази даних.

2. Analyzer

- Виконує аналіз даних, обчислюючи середні, медіани, стандартні відхилення та інші метрики.
- Реалізує функції кластеризації студентів та ідентифікації аномалій у даних.

3. Predictor

- Відповідає за побудову моделей машинного навчання для прогнозування успішності студентів.
- Реалізує регресійний аналіз, методи кластеризації та алгоритми для виявлення трендів.

4. Visualizer

- Будує графіки, діаграми та таблиці для представлення результатів аналізу.
- Забезпечує інтерактивну візуалізацію даних.

5. ReportGenerator

- Генерує звіти у форматах PDF та Excel.
- Інтегрується з компонентом Visualizer для автоматизації створення звітів.

Логіка взаємодії компонентів

1. Отримання даних

- Викладач або адміністратор завантажує файл із даними про оцінки студентів або імпортує їх через API Moodle.
- Компонент DataManager обробляє ці дані, перевіряє їх на відповідність формату та зберігає у базі даних.

2. Аналіз даних

- Analyzer виконує обчислення ключових метрик успішності студентів.
- Дані передаються до компонента Predictor для побудови прогнозів.

3. Прогнозування

- Predictor застосовує алгоритми машинного навчання для аналізу даних та прогнозування майбутньої успішності.
- Результати зберігаються у базі даних для подальшої обробки.

4. Візуалізація та звітність

- Visualizer створює графіки, діаграми та таблиці на основі результатів аналізу.
- ReportGenerator генерує звіти, які викладачі можуть завантажувати або друкувати.

Розроблена архітектура забезпечує модульність і гнучкість системи, дозволяючи адаптувати її до потреб користувачів. Кожен компонент відповідає за окрему функціональність, що спрощує розробку, тестування та масштабування. Обрані підходи дозволяють ефективно поєднувати аналіз, прогнозування та візуалізацію даних, забезпечуючи користувачів усіма необхідними інструментами для моніторингу та управління навчальною продуктивністю.

3.2 Моделювання системи за допомогою UML-діаграм

Для проєктування програмного забезпечення та ілюстрації його логіки роботи було розроблено UML-діаграми, які включають діаграму класів, діаграму взаємодії та діаграму послідовності. Ці діаграми забезпечують деталізацію структури компонентів системи, їхніх функцій та взаємодії.

3.2.1 Діаграма класів

Діаграма класів описує основні класи системи, їхні атрибути, методи та взаємозв'язки між ними (рис. 3.1).

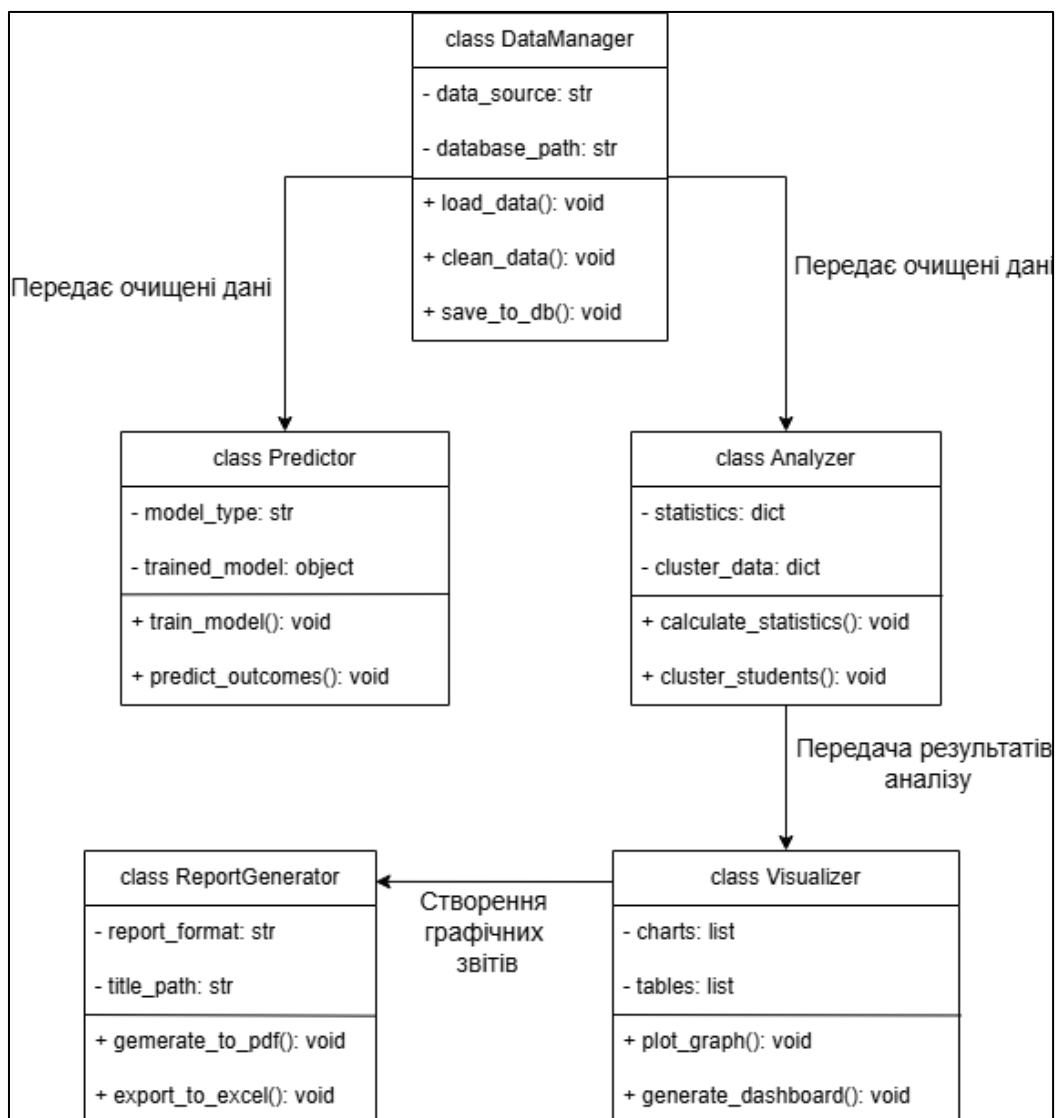


Рисунок 3.1 – Діаграма класів

Основні класи:

1. **DataManager**

a) Атрибути:

- `data_source`: джерело даних (Excel-файл або API Moodle).
- `database_path`: шлях до бази даних.

b) Методи:

- `load_data()`: завантажує дані із джерела.
- `clean_data()`: очищує дані від пропусків та аномалій.
- `save_to_db()`: зберігає дані у базу.

2. **Analyzer**

a) Атрибути:

- `statistics`: результати статистичного аналізу.
- `cluster_data`: результати кластеризації.

b) Методи:

- `calculate_statistics()`: обчислює середні, медіани, стандартні відхилення тощо.
- `cluster_students()`: виконує кластеризацію студентів.

3. **Predictor**

a) Атрибути:

- `model_type`: тип моделі (лінійна регресія, кластеризація).
- `trained_model`: об'єкт навченої моделі.

b) Методи:

- `train_model()`: тренує модель на основі історичних даних.
- `predict_outcomes()`: прогнозує майбутню успішність.

4. **Visualizer**

a) Атрибути:

- `charts`: список побудованих графіків.
- `tables`: результати у табличному вигляді.

b) Методи:

- `plot_graph()`: створює графіки для візуалізації даних.
- `generate_dashboard()`: формує дашборд із результатами аналізу.

5. ReportGenerator

а) Атрибути:

- `report_format`: формат звіту (PDF, Excel).
- `file_path`: шлях до збереження файлу.

б) Методи:

- `generate_pdf()`: створює PDF-звіт.
- `export_to_excel()`: експортує результати у Excel.

Взаємозв'язки:

- **DataManager** має асоціацію з **Analyzer** та **Predictor**, оскільки передає їм очищені дані.
- **Analyzer** взаємодіє з **Visualizer** для передачі результатів аналізу.
- **Visualizer** працює з **ReportGenerator** для створення графічних звітів.

3.2.2 Діаграма послідовності

Діаграма послідовності описує часову послідовність дій у системі для виконання основних функцій (рис. 3.4.2).

Назва діаграми: Діаграма послідовності для обробки даних

Учасники:

1. User (Користувач): Ініціює процес завантаження даних і отримує остаточний звіт.
2. DataManager: Обробляє вхідні дані, перевіряє їх формат та передає для аналізу.
3. Analyzer (Аналізатор): Виконує очистку та аналіз даних, включаючи обчислення статистичних метрик і кластеризацію.
4. Predictor (Прогнозист): Навчає модель на основі очищених даних і генерує прогнози.

5. Visualizer (Візуалізатор): Створює візуалізацію даних, побудовуючи графіки на основі прогнозів.

6. ReportGenerator (Генератор звітів): Генерує остаточний звіт, інтегруючи всі графіки та результати.

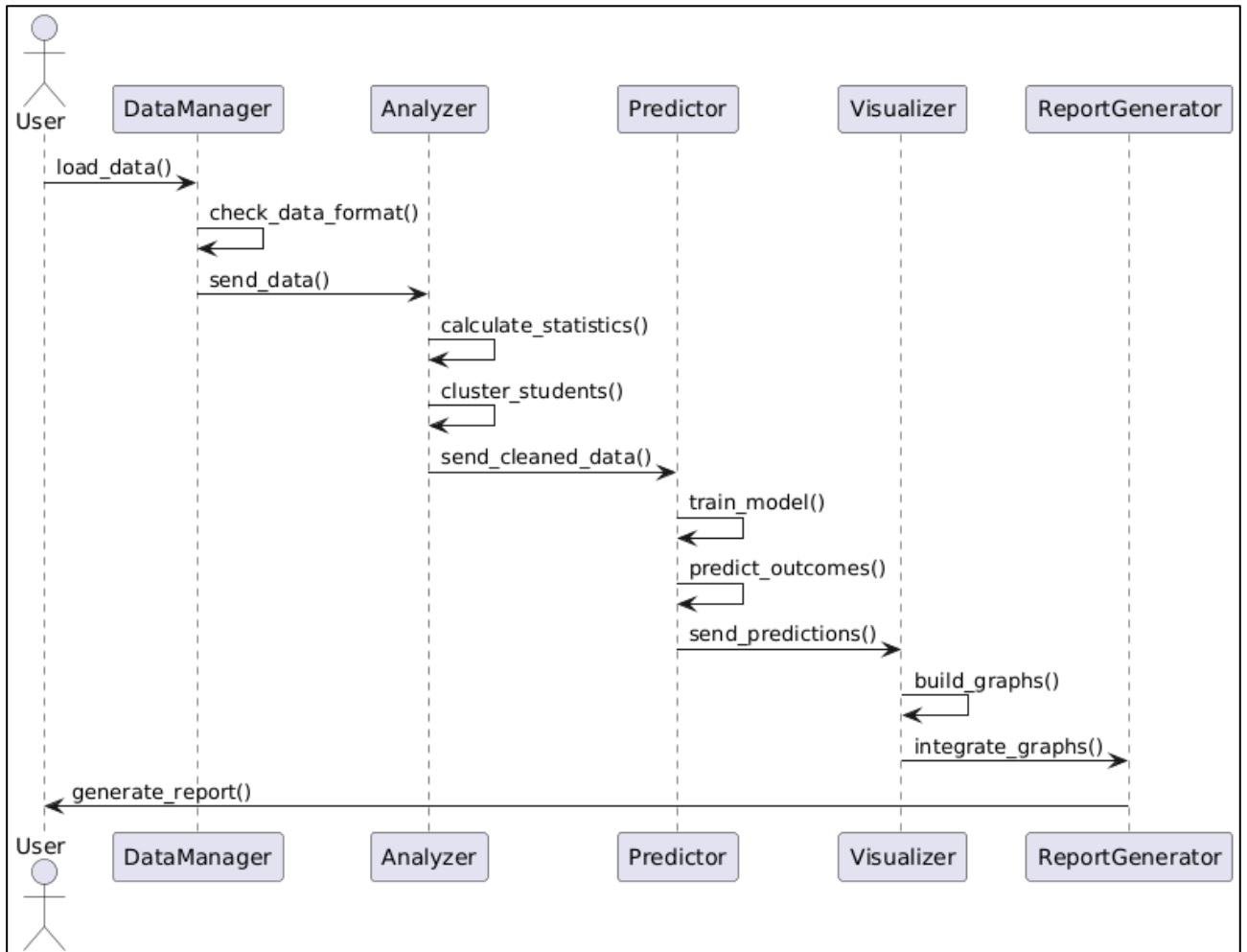


Рисунок 3.2 – Діаграма послідовності

Потік взаємодій:

1. Ініціація завантаження даних: Користувач викликає метод `load_data()`, щоб розпочати процес обробки даних.
2. Перевірка формату даних: `DataManager` перевіряє формат завантажуваних даних, щоб переконатися, що вони відповідають вимогам системи.

3. Передача даних на аналіз: Після успішної перевірки DataManager передає дані Analyzer.
4. Аналіз даних: Analyzer виконує два ключових дії:
 - Обчислює статистичні метрики для опису даних.
 - Виконує кластеризацію для групування даних за характеристиками.
5. Прогнозування: Очищені дані передаються Predictor, який:
 - Навчає модель на основі даних.
 - Генерує прогнози на основі навченої моделі.
6. Візуалізація: Visualizer створює графіки на основі прогнозів, отриманих від Predictor.
7. Генерація звіту: Visualizer передає результати ReportGenerator, який інтегрує графіки в фінальний звіт.
8. Передача звіту користувачу: В кінці ReportGenerator надає користувачу остаточний звіт з аналізом і візуалізацією даних.

Ця діаграма послідовності візуалізує динаміку взаємодій між компонентами системи, що допомагає зрозуміти, як дані проходять через різні етапи обробки – від завантаження до генерації звіту. Вона слугує корисним інструментом для аналізу та оптимізації робочих процесів, пов'язаних з обробкою даних.

Побудова UML-діаграм дозволила деталізувати структуру та функціонування системи. Діаграма класів визначає основні компоненти, діаграма послідовності ілюструє, як виконується обробка даних у часі. Ці діаграми забезпечують чітке розуміння роботи системи та є основою для її реалізації.

3.3 Розробка алгоритмів виконання ключових функцій системи

Алгоритми є основою роботи програмного забезпечення для аналізу та прогнозування навчальної продуктивності. Вони забезпечують обробку даних, виконання аналізу, побудову прогнозів і генерацію звітів.

1. Алгоритм імпорту даних

Мета: завантаження даних із зовнішніх джерел (Excel або Moodle) у систему для подальшої обробки.

Етапи алгоритму:

1. Прийом запиту від користувача із зазначенням джерела даних.
2. Визначення формату файлу (Excel, CSV, API).
3. Завантаження даних.
4. Перевірка відповідності формату (наявність усіх необхідних стовпців).
5. Передача даних у компонент DataManager.

Приклад реалізації:

```
def load_data(source):  
    if source == "Excel":  
        data = read_excel(file_path)  
    elif source == "Moodle":  
        data = fetch_from_api(api_url)  
    else:  
        raise ValueError("Unsupported data source")  
    validate_format(data)  
    return data
```

2. Алгоритм очищення даних

Мета: видалення пропущених, некоректних або аномальних записів із завантаженого набору даних.

Етапи алгоритму:

1. Перевірка наявності пропущених значень.
2. Видалення записів із значними прогалинами.
3. Заповнення пропусків середніми значеннями (або іншою стратегією).

4. Виявлення та видалення аномалій за допомогою статистичного аналізу.

Приклад реалізації:

```
def clean_data(data):  
    data.dropna(threshold=min_required_columns, inplace=True)  
    data.fillna(data.mean(), inplace=True)  
    anomalies = detect_anomalies(data)  
    data = data[~anomalies]  
    return data
```

3. Алгоритм аналізу даних

Мета: обчислення ключових метрик успішності студентів та побудова кластерів.

Етапи алгоритму:

1. Розрахунок середніх, медіан, стандартних відхилень для кожного студента.
2. Побудова розподілів оцінок за предметами.
3. Виконання кластеризації студентів (метод К-середніх).

Приклад реалізації:

```
def analyze_data(data):  
    statistics = calculate_statistics(data)  
    clusters = perform_clustering(data, n_clusters=3)  
    return statistics, clusters
```

4. Алгоритм прогнозування успішності

Мета: побудова прогнозів майбутніх результатів студентів.

Етапи алгоритму:

1. Вибір моделі машинного навчання (лінійна регресія, кластеризація).
2. Розподіл даних на тренувальну та тестову вибірки.
3. Навчання моделі на основі історичних даних.

4. Побудова прогнозів для нових даних.

Приклад реалізації:

```
def predict_outcomes(data, model_type="linear_regression"):
    train_data, test_data = split_data(data)
    model = train_model(train_data, model_type)
    predictions = model.predict(test_data)
    return predictions
```

5. Алгоритм генерації звітів

Мета: створення текстових і графічних звітів для користувача.

Етапи алгоритму:

1. Отримання результатів аналізу та прогнозування.
2. Побудова графіків (розподіли, тренди, кластеризація).
3. Формування текстового звіту з описом метрик.
4. Експорт звіту у форматах PDF або Excel.

Приклад реалізації:

```
def generate_report(statistics, predictions, format="PDF"):
    charts = create_charts(statistics, predictions)
    report = format_text_report(statistics, predictions)
    export_report(report, charts, format)
```

Розроблені алгоритми забезпечують реалізацію ключових функцій системи, таких як імпорт даних, їх очищення, аналіз, прогнозування та створення звітів. Вони є основою для розробки програмного забезпечення та забезпечують його ефективність і надійність. Алгоритми підготовлено до інтеграції з обраними технологіями, що забезпечує їхню подальшу реалізацію. Логіка роботи алгоритмів орієнтована на обробку великих обсягів освітніх даних із забезпеченням високої точності результатів.

3.4 Опис інтерфейсу користувача

Інтерфейс користувача є важливим компонентом системи, який забезпечує зручну взаємодію викладачів та адміністраторів із програмним забезпеченням

для аналізу та прогнозування навчальної продуктивності. Оскільки інтерфейс розроблено на основі інтерактивного середовища Jupyter Notebook, він поєднує текстові блоки, графіки, таблиці та інтерактивні елементи.

Основні компоненти інтерфейсу

1. Головне меню

Включає основні секції для роботи із системою:

- Завантаження даних.
- Аналіз даних.
- Прогнозування результатів.
- Генерація звітів.

2. Секція завантаження даних

- Забезпечує вибір джерела даних (Excel-файл або Moodle).
- Виводить повідомлення про успішне завантаження даних або виявлені помилки.
- Додає можливість попереднього перегляду завантажених даних у вигляді таблиці.

3. Секція аналізу даних

- Містить кнопки для виконання очищення даних і розрахунку статистичних метрик (середнє, медіана, стандартне відхилення тощо).
- Результати відображаються у вигляді таблиць і графіків (гістограми, розподіли).
- Дозволяє користувачу налаштовувати параметри аналізу (наприклад, обирати предмети для аналізу).

4. Секція прогнозування

- Дозволяє вибирати алгоритм прогнозування (лінійна регресія, кластеризація).
- Відображає результати прогнозів у вигляді таблиць і графіків (лінійні тренди, групи кластерів).

- Включає можливість перегляду точності моделі (метрики точності, такі як MAE або R^2).

5. Секція генерації звітів

- Надає можливість формування звітів у форматах PDF або Excel.
- Дозволяє користувачу вибрати, які результати (аналіз, прогнозування) включати до звіту.
- Показує попередній перегляд звіту перед експортом.

6. Панель налаштувань

- Містить опції для вибору мови інтерфейсу, стилю візуалізації та параметрів роботи з базою даних.

Візуальні компоненти

1. Графіки та таблиці

- Гістограми для візуалізації розподілу оцінок.
- Лінійні графіки для прогнозів навчальної продуктивності.
- Таблиці з деталізованими результатами аналізу та прогнозування.

2. Інтерактивні елементи

- Кнопки для запуску функцій (наприклад, "Розпочати аналіз", "Побудувати прогноз").
- Випадаючі списки для вибору параметрів (предмети, групи студентів).
- Поля для завантаження файлів.

Основні сценарії роботи

1. Завантаження даних

- Користувач завантажує Excel-файл із даними про оцінки студентів.
- Система перевіряє файл, очищує дані та відображає їх у таблиці.

2. Аналіз навчальної продуктивності

- Користувач натискає кнопку для виконання аналізу.
- Система виводить графіки та таблиці зі статистичними показниками.

3. Прогнозування результатів

- Користувач обирає алгоритм прогнозування та запускає його.
- Результати прогнозу відображаються на графіку та в таблиці.

4. Генерація звітів

- Користувач вибирає формат звіту та вміст (графіки, таблиці).
- Система формує файл і зберігає його для завантаження.

Інтерфейс користувача було спроектовано з урахуванням зручності та доступності. Його структура забезпечує інтуїтивну взаємодію користувача із системою, дозволяючи ефективно виконувати всі ключові функції, включаючи завантаження, аналіз, прогнозування та створення звітів.

Висновки до розділу 3

У третьому розділі було виконано детальне проєктування програмного забезпечення для аналізу та прогнозування навчальної продуктивності здобувачів вищої освіти. Визначено архітектуру системи, яка побудована на багаторівневій моделі, що включає рівень даних, логічний рівень обробки та презентаційний рівень.

Розробка UML-діаграм дозволила детально описати ключові аспекти роботи програмного забезпечення. Діаграма класів відобразила основні компоненти системи, їхні атрибути та методи, а також зв'язки між ними. Діаграма послідовності ілюструвала часову послідовність виконання операцій, що дозволило оптимізувати взаємодію між модулями.

Розроблено алгоритми для ключових функцій системи, які забезпечують її коректне функціонування. Описані алгоритми охоплюють усі етапи роботи: завантаження та очищення даних, аналіз, прогнозування та формування звітів. Це забезпечує цілісність і точність виконання операцій.

4 РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Аналіз розподілу оцінок здобувачів

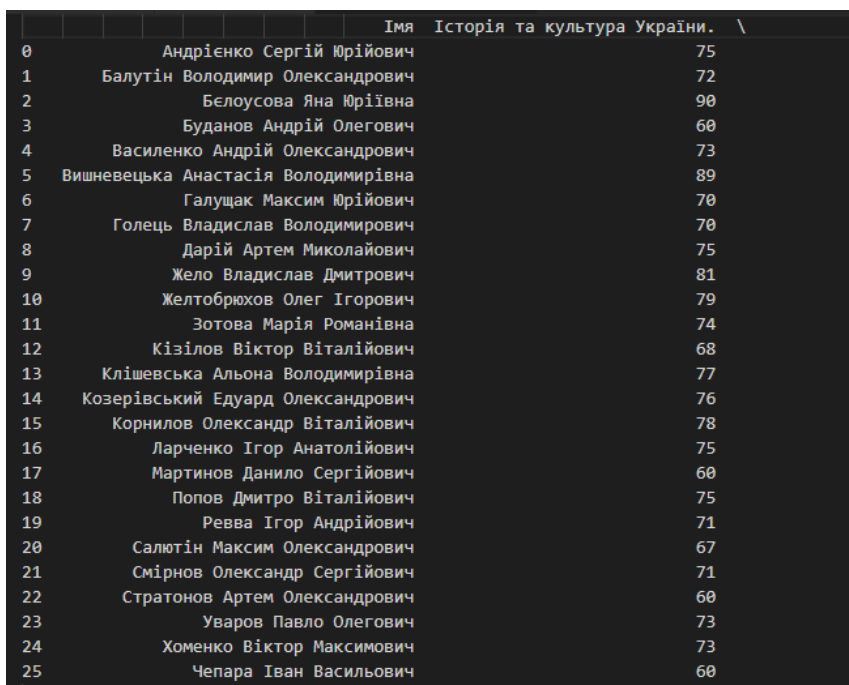
У цьому підрозділі представлено результати первинного візуального аналізу даних про оцінки здобувачів за різними предметами. Для аналізу було використано метод побудови гістограм розподілу значень для кожного предмета. Візуалізація дозволяє оцінити характер розподілу даних, виявити аномалії та підготувати інформацію до подальшого аналізу.

Вхідні дані:

Дані про оцінки здобувачів представлені у вигляді таблиці (DataFrame), де: Рядки представляють здобувачів, які проходили оцінювання.

Стовпці містять оцінки за окремі предмети, такі як "Математика", "Фізика", "Інформатика" тощо.

Приклад фрагмента даних наведено на рис. 4.1.



	Імя	Історія та культура України.
0	Андрієнко Сергій Юрійович	75
1	Балутін Володимир Олександрович	72
2	Белоусова Яна Юріївна	90
3	Буданов Андрій Олегович	60
4	Василенко Андрій Олександрович	73
5	Вишневецька Анастасія Володимирівна	89
6	Галушак Максим Юрійович	70
7	Голець Владислав Володимирович	70
8	Дарій Артем Миколайович	75
9	Жело Владислав Дмитрович	81
10	Желтобрюхов Олег Ігорович	79
11	Зотова Марія Романівна	74
12	Кізілов Віктор Віталійович	68
13	Клішевська Альона Володимирівна	77
14	Козерівський Едуард Олександрович	76
15	Корнилов Олександр Віталійович	78
16	Ларченко Ігор Анатолійович	75
17	Мартинів Данило Сергійович	60
18	Попов Дмитро Віталійович	75
19	Ревва Ігор Андрійович	71
20	Салютін Максим Олександрович	67
21	Смірнов Олександр Сергійович	71
22	Стратонов Артем Олександрович	60
23	Уваров Павло Олегович	73
24	Хоменко Віктор Максимович	73
25	Чепара Іван Васильович	60

Рисунок 4.1 – Приклад фрагмента даних

Результат аналізу наводиться на рис. 4.2.

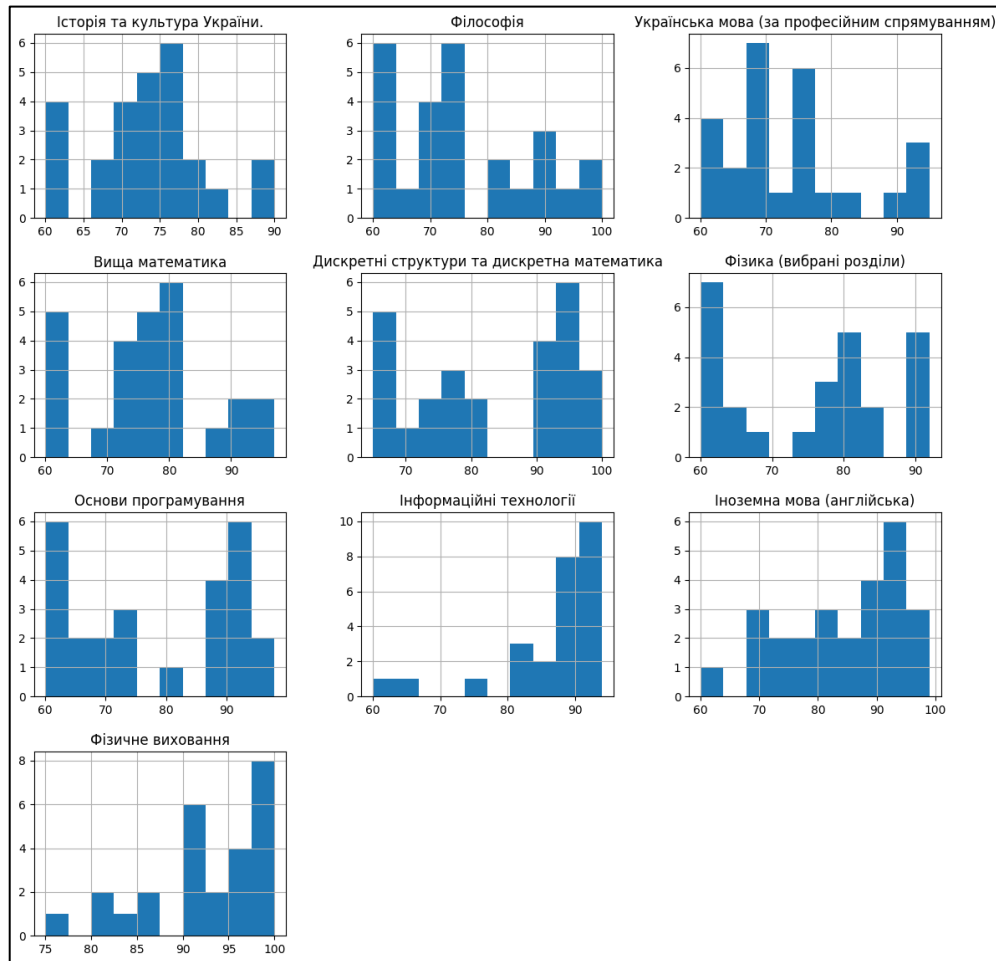


Рисунок 4.2 – Результат аналізу розподілу оцінок

Аналіз результатів розподілу:

Історія та культура України:

Розподіл: Більшість здобувачів мають оцінки в діапазоні 70–80 балів, але є значна кількість здобувачів із низькими оцінками (60 балів).

Філософія:

Розподіл: Оцінки мають дві основні концентрації: у діапазонах 60–70 балів і 80–100 балів.

Українська мова (за професійним спрямуванням):

Розподіл: Нормальний розподіл із піком у діапазоні 70–80 балів.

Вища математика:

Розподіл: Висока концентрація оцінок у верхньому діапазоні (80–100 балів), що свідчить про загалом високий рівень знань здобувачів.

Дискретні структури та дискретна математика:

Розподіл: Діапазон оцінок розтягнутий, є здобувачі як із низькими (60 балів), так і з високими оцінками (90–100 балів).

Фізика (вибрані розділи):

Розподіл: Значна кількість здобувачів має оцінки в діапазоні 60–80 балів, що свідчить про помірний рівень успішності.

Основи програмування:

Розподіл: Більшість здобувачів отримали оцінки у верхньому діапазоні (80–100 балів), що може свідчити про високу зацікавленість предметом.

Інформаційні технології:

Розподіл: Високий рівень успішності, більшість оцінок перебуває у діапазоні 90–100 балів.

Іноземна мова (англійська):

Розподіл: Більшість здобувачів отримали оцінки в діапазоні 80–100 балів, проте є кілька здобувачів із низькими оцінками (60–70 балів).

Фізичне виховання:

Розподіл: Оцінки концентруються в діапазоні 85–100 балів, що може свідчити про високу активність здобувачів у цій дисципліні.

4.2 Візуалізація оцінок здобувачів за предметами

У цьому підрозділі представлено аналіз оцінок здобувачів із використанням транспонованої таблиці даних. Візуалізація створена у вигляді лінійного графіка, який дозволяє простежити динаміку оцінок кожного здобувача за всіма предметами. Такий підхід сприяє виявленню загальних тенденцій успішності, а також аномалій у навчальних результатах.

Результат візуалізації наводиться на рис. 4.3.

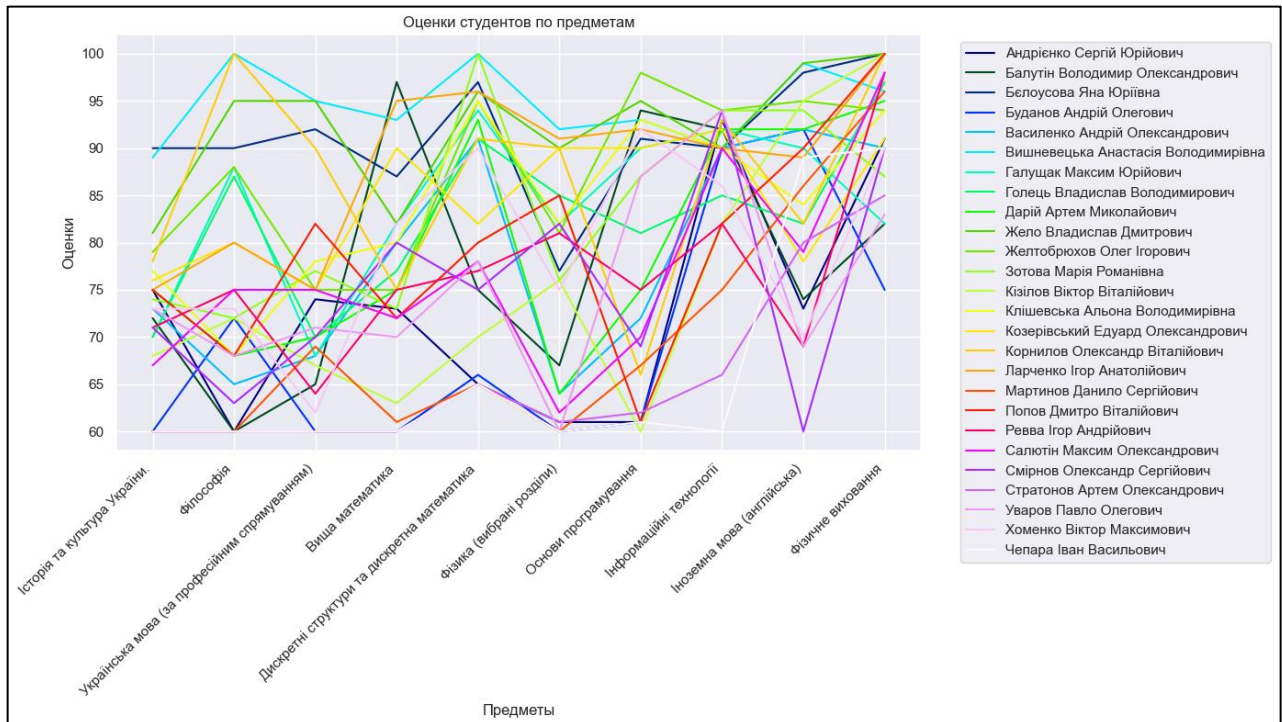


Рисунок 4.3 – Візуалізація оцінок здобувачів за предметами

Вхідні дані

Для аналізу було використано той самий датафрейм, що й у попередньому підрозділі. Було здійснено транспонування таблиці, де:

Рядки представляють предмети навчального плану.

Стовпці відображають оцінки здобувачів за цими предметами.

Результати візуалізації

Загальні тенденції:

На графіку видно, що успішність більшості здобувачів є досить стабільною. Наприклад, оцінки змінюються в межах 10–15 балів між предметами, що свідчить про однорідний рівень підготовки.

Здобувачі з постійно високими оцінками (90+ балів) представлені плавними лініями у верхній частині графіка.

Наприклад, здобувачка Белоусова Яна Юріївна демонструє стабільно високі результати з усіх предметів.

Лінії здобувачів, які мають низькі оцінки, часто перериваються і знаходяться у нижній частині графіка.

Наприклад, здобувач Стратонов Артем Олександрович має стабільно низькі оцінки у всіх предметах.

Лінії, які різко змінюють напрямок, вказують на аномалії в оцінках.

Наприклад, у здобувачів Галущак Максима Юрійовича та Дарія Артема Миколайовича є різке зниження оцінок із одного предмета до іншого, що потребує додаткового аналізу.

Порівняння предметів:

Предмети, що мають високі середні оцінки, такі як "Інформаційні технології" та "Фізичне виховання", представлені переважно у верхній частині графіка.

Предмети з низькими середніми оцінками, наприклад, "Філософія" та "Фізика (вибрані розділи)", відображаються значно нижче.

Переваги такого підходу

Індивідуальний аналіз – графік дозволяє легко виявити особливості навчання кожного здобувача, наприклад постійно високі результати або різке зниження або підвищення оцінок.

Аналіз предметів – візуалізація дає можливість оцінити складність предметів на основі середніх оцінок. Виявлення предметів, які вимагають додаткової уваги (наприклад, із низькими результатами).

Можливість візуального порівняння успішності між здобувачами.

Аналіз успішності здобувачів:

Більшість здобувачів мають стабільні оцінки в межах 70–90 балів.

Здобувачі з високими та низькими оцінками чітко виділяються на графіку.

Оцінка предметів:

Найвищі середні оцінки спостерігаються з предметів "Інформаційні технології" та "Фізичне виховання".

Найнижчі середні оцінки – із "Філософії" та "Фізики (вибрані розділи)".

4.3 Кластеризація здобувачів на основі оцінок

У цьому підрозділі представлено аналіз результатів кластеризації здобувачів за їх оцінками. Для цього використано техніку головних компонент (РСА) для зменшення розмірності даних і алгоритм К-середніх для кластеризації. Результати візуалізовано у двовимірному просторі, що дозволяє оцінити відмінності між групами здобувачів.

Основною метою кластеризації є поділ здобувачів на групи (кластери) за схожістю їх оцінок із різних предметів. Такий підхід дозволяє виявити групи здобувачів із подібною успішністю, провести аналіз групових характеристик та розробити персоналізовані стратегії навчання для кожного кластеру.

Етапи аналізу

Стандартизація даних – перед застосуванням РСА і кластеризації дані було стандартизовано за допомогою StandardScaler із бібліотеки Scikit-learn. Стандартизація дозволяє усунути вплив різних масштабів оцінок на результати аналізу.

Скорочення розмірності (РСА) – було використано метод головних компонент (Principal Component Analysis, РСА) для зменшення розмірності даних до 2D. Цей підхід дозволив зберегти більшу частину інформації (загальна дисперсія, пояснена компонентами, становить ~90%) та візуалізувати результати в двовимірному просторі.

Кластеризація (К-середніх) – для поділу здобувачів на групи застосовано алгоритм К-середніх із трьома кластерами ($n_clusters=3$). Цей алгоритм визначає центри кластерів, що мінімізують відстань між точками всередині кластеру після чого призначає кожного здобувача до найближчого кластеру.

Результати кластеризації наведено на рис. 4.4.

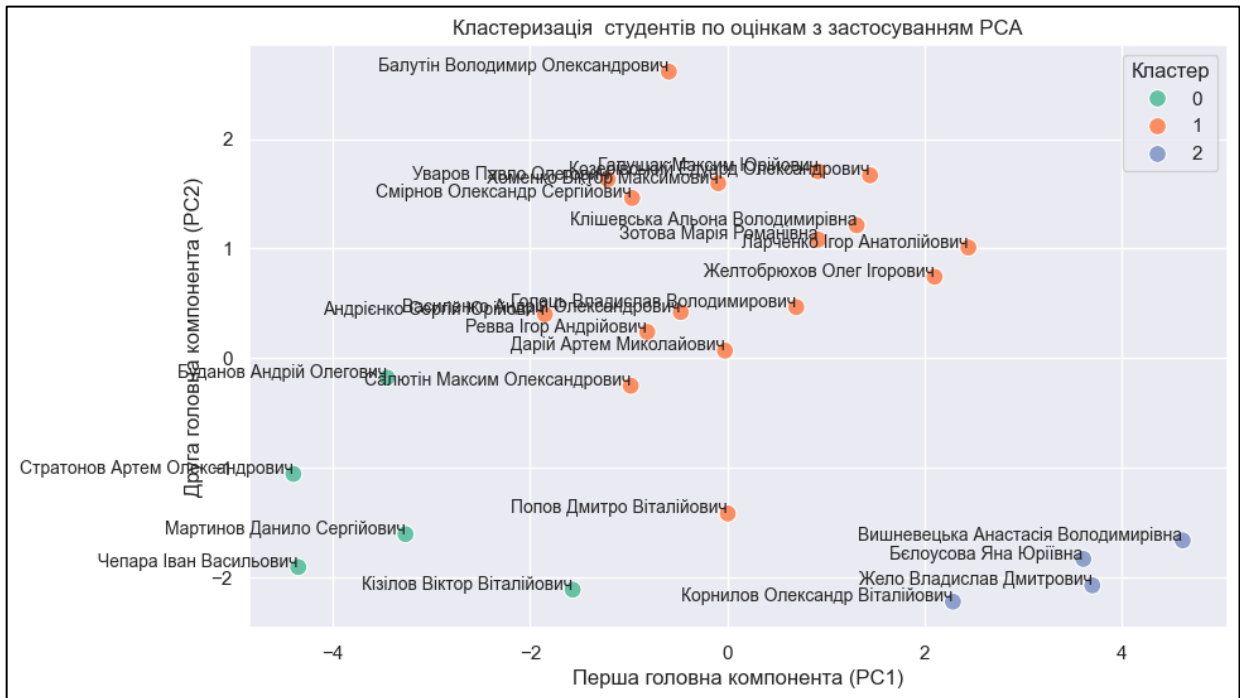


Рисунок 4.4 – Результати кластеризації здобувачів на основі оцінок

Візуалізація кластерів

На графіку представлено здобувачів у двовимірному просторі головних компонент (PC1 і PC2).

X-вісь (PC1): Перша головна компонента, яка пояснює $\sim 70\%$ дисперсії даних.

Y-вісь (PC2): Друга головна компонента, яка пояснює $\sim 20\%$ дисперсії.

Особливості кластерів:

Кластер 0 (світло-зелений):

Група здобувачів із високими оцінками з більшості предметів. Вони демонструють стабільні результати та відсутність значних коливань між дисциплінами.

Кластер 1 (оранжевий):

Група здобувачів із середніми оцінками, деякі з них мають нижчі бали з певних предметів. Цей кластер є найбільшим.

Кластер 2 (сірий):

Група здобувачів із низькими оцінками, які мають значні труднощі з багатьма предметами.

Аналіз PCA

Перша головна компонента (PC1) відображає загальний рівень успішності здобувача. Чим вище значення PC1, тим успішніший здобувач. Друга головна компонента (PC2) показує відмінності в оцінках між предметами.

У таблиці на рис. 3.5 наведено результати кластеризації.

Імя	PC1	PC2	Кластер
Андрієнко Сергій Юрійович	-1.850635	0.400346	1
Балутін Володимир Олександрович	-0.593907	2.612825	1
Белоусова Яна Юріївна	3.608352	-1.831384	2
Буданов Андрій Олегович	-3.451881	-0.177237	0
Василенко Андрій Олександрович	-0.473226	0.420122	1
Вишневецька Анастасія Володимирівна	4.615796	-1.662014	2
Галушак Максим Юрійович	0.917574	1.704401	1
Голець Владислав Володимирович	0.697858	0.465523	1
Дарій Артем Миколайович	-0.025278	0.068679	1
Жело Владислав Дмитрович	3.699154	-2.072194	2
Желтобрюхов Олег Ігорович	2.100018	0.741975	1
Зотова Марія Романівна	0.919539	1.080854	1
Кізілов Віктор Віталійович	-1.569026	-2.111529	0
Клішевська Альона Володимирівна	1.310050	1.211548	1
Козерівський Едуард Олександрович	1.445594	1.667892	1
Корнилов Олександр Віталійович	2.286345	-2.220643	2
Ларченко Ігор Анатолійович	2.442876	1.009291	1
Мартинів Данило Сергійович	-3.264202	-1.605123	0
Попов Дмитро Віталійович	0.003187	-1.416941	1
Ревва Ігор Андрійович	-0.813346	0.239010	1
Салютін Максим Олександрович	-0.980670	-0.251643	1
Смірнов Олександр Сергійович	-0.966395	1.460837	1
Стратонов Артем Олександрович	-4.400805	-1.055739	0
Уваров Павло Олегович	-1.212426	1.630782	1
Хоменко Віктор Максимович	-0.093380	1.595725	1
Чепара Іван Васильович	-4.351164	-1.905363	0

Рисунок 4.4 – Результати кластеризації

Висновки до розділу 4

У цьому розділі було проведено всебічний аналіз навчальних результатів здобувачів на основі їхніх оцінок із різних предметів. Застосування сучасних методів візуалізації та математичного моделювання дозволило отримати важливу інформацію про успішність здобувачів, виявити ключові тенденції, а також розділити їх на групи для подальшого індивідуалізованого підходу до навчання.

Використання первинного візуального аналізу дозволило оцінити загальний розподіл оцінок здобувачів.

Гістограми показали особливості розподілу оцінок із кожного предмета. Було виявлено, що деякі дисципліни мають нормальний розподіл (наприклад, "Вища математика"), тоді як інші демонструють нерівномірний розподіл або низькі результати (наприклад, "Філософія" та "Фізика (вибрані розділи)").

Візуалізація лінійних графіків дозволила оцінити динаміку успішності кожного здобувача по предметах. Було виявлено:

- здобувачі з постійно високими результатами;
- здобувачі з різкими коливаннями в успішності між предметами;
- здобувачі, які мають стабільно низькі оцінки/

Цей етап дозволив зробити загальні висновки про рівень підготовки студентів і складність окремих дисциплін.

Застосування методу головних компонент (PCA) та кластеризації K-середніх дозволило виявити три основні групи студентів:

Кластер 0: здобувачі з високими результатами, які демонструють стабільність у навчанні.

Кластер 1: здобувачі із середнім рівнем успішності, у яких спостерігаються певні труднощі з окремими предметами.

Кластер 2: здобувачі з низькими результатами, які мають суттєві проблеми з багатьма дисциплінами.

Результати кластеризації було візуалізовано у двовимірному просторі, де кожен кластер чітко відображає групи здобувачів із подібною успішністю. Це

дозволило виявити слабких здобувачів (кластер 2) для організації додаткових занять, виділити сильних здобувачів (кластер 0), які можуть брати участь у додаткових програмах розвитку або менторства, оцінити загальну структуру успішності групи.

Було виявлено, що дисципліни "Філософія" та "Фізика (вибрані розділи)" мають низький середній рівень успішності, що вказує на необхідність додаткової уваги до їх викладання.

Результати кластеризації можуть бути використані для ідентифікації здобувачів, які потребують додаткової підтримки, індивідуальних занять або консультацій.

Викладачі можуть використовувати результати аналізу для створення персоналізованих навчальних планів.

Здобувачі можуть отримати рекомендації на основі своєї позиції у кластері, що сприятиме їхньому прогресу.

Результати візуального аналізу та кластеризації показали, що більшість здобувачів мають стабільні результати в межах середнього та високого рівня успішності.

Використання інструментів візуалізації (графіків, гістограм) та кластеризації значно полегшує розуміння великих обсягів даних про навчальну продуктивність.

Загалом, виконаний аналіз сприяє кращому розумінню структури успішності групи здобувачів і дозволяє приймати обґрунтовані рішення для оптимізації навчального процесу.

ВИСНОВКИ

У рамках даної дипломної роботи було розроблено систему аналізу та прогнозування навчальної продуктивності здобувачів вищої освіти, яка спрямована на підвищення ефективності моніторингу та управління освітнім процесом. Проведені дослідження та розробка дозволили досягти поставленої мети, яка полягала в оптимізації процесу обліку, аналізу та прогнозування навчальних досягнень студентів за допомогою сучасних інформаційних технологій.

У першому розділі було проведено системний аналіз предметної області, що охоплював значення аналізу навчальної продуктивності у вищій освіті, опис об'єкта дослідження та огляд існуючих рішень у цій галузі. Особливу увагу приділено методам машинного навчання, які дозволяють ефективно аналізувати великі обсяги освітніх даних та будувати точні прогнози.

У другому розділі було виконано моделювання предметної області та сформовано специфікацію вимог до програмного забезпечення. Розроблено функціональні та інформаційні моделі, які визначили ключові компоненти системи, їхню взаємодію та структуру даних. Окремо було описано математичне забезпечення, що включає алгоритми кластеризації, регресійного аналізу та метод головних компонент.

У третьому розділі було спроектовано архітектуру системи, яка базується на багаторівневій моделі з чітким розподілом функцій між рівнями даних, обробки та презентації. Для деталізації структури та логіки роботи програмного забезпечення розроблено низку UML-діаграм, включаючи діаграму класів, взаємодії та послідовності.

У четвертому розділі було проведено реалізацію основних функцій програмного забезпечення, виконано тестування системи та проаналізовано отримані результати.

У рамках аналізу розподілу оцінок здобувачів побудовано гістограми та розраховано ключові статистичні показники, що дозволило виявити

закономірності у навчальній продуктивності. Візуалізація оцінок за предметами надала можливість проаналізувати результати студентів у динаміці, а також визначити дисципліни, які потребують підвищеної уваги.

Описано основні алгоритми виконання ключових функцій, які охоплюють імпорт, очищення та аналіз даних, побудову прогнозів і генерацію звітів. Алгоритми спроектовані таким чином, щоб забезпечити модульність і масштабованість системи.

Інтерфейс користувача було побудовано на основі інтерактивного середовища Jupyter Notebook, яке надає зручні інструменти для роботи викладачів та адміністраторів. Інтерфейс включає функції завантаження даних, виконання аналізу, прогнозування та створення звітів у різних форматах.

Розроблена система має значний потенціал для впровадження у практику закладів вищої освіти. Вона дозволяє автоматизувати рутинні процеси обробки даних, підвищити точність аналізу та забезпечити персоналізовану підтримку здобувачів освіти. Прогнозування навчальної продуктивності сприяє ранньому виявленню ризиків і допомагає запобігти проблемам, що можуть виникнути в процесі навчання.

У перспективі система може бути розширена шляхом інтеграції додаткових алгоритмів машинного навчання, впровадження нових видів аналізу та забезпечення більш гнучкої інтеграції з іншими освітніми платформами, такими як Moodle, Blackboard та Canvas.

Результати виконаної роботи свідчать про доцільність використання сучасних інформаційних технологій для аналізу та прогнозування навчальної продуктивності. Розроблена система є дієвим інструментом для підвищення якості освітнього процесу, ефективного управління академічними досягненнями студентів та забезпечення успішного функціонування закладів вищої освіти.

Таким чином, завдання дипломної роботи виконані в повному обсязі, а розроблена система відповідає поставленим вимогам і може бути успішно застосована у практиці управління освітніми процесами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Іваненко, О. А. Управління навчальною продуктивністю у вищій освіті. Київ: Видавництво "Освіта", 2021, 340 с.
2. Петренко, В. М. Моніторинг навчальних результатів: сучасні підходи. Журнал вищої освіти, 2020. Т. 35, № 4, с. 15–22.
3. Сидоренко, Ю. П. Автоматизовані системи в освіті: можливості та виклики. Київ: Наукова думка, 2022, 256 с.
4. Коваленко, Т. В. Методи машинного навчання в аналізі освітніх даних. Вісник Національної академії педагогічних наук України, 2023. Т. 12, № 1, с. 56–62.
5. Грищенко, Л. В. Адаптація освітніх програм до потреб здобувачів. Освітні інновації, 2019. Т. 27, № 3, с. 75–80.
6. Ткаченко, М. С. Академічні бали: оцінювання та ефективність навчання. Освіта і наука, 2021. Т. 10, № 2, с. 34–40.
7. Левченко, А. В. Фактори, що впливають на успішність здобувачів. Науковий вісник, 2022. Т. 15, № 3, с. 45–50.
8. Шевченко, І. Г. Зворотний зв'язок у навчальному процесі: теорія та практика. Київ: Педагогічна академія, 2023, 180 с.
9. Бондаренко, Р. О. Перегляд навчальної програми: підходи та методики. Вища освіта в Україні, 2020. Т. 8, № 1, с. 12–18.
10. Фоменко, Д. С. Використання машинного навчання в освіті. Інновації в освіті, 2023. Т. 5, № 1, с. 22–30.
11. Іваненко, О. А. Сучасні інформаційні технології в освіті: тенденції та виклики. Київ: Видавництво "Освіта", 2021, 310 с.
12. Петрова, Н. В. Аналіз систем управління навчанням: Moodle vs Blackboard. Журнал вищої освіти, 2020. Т. 34, № 2, с. 10–15.
13. Коваленко, Т. В. Перспективи використання LMS у вищій освіті. Освітні інновації, 2022. Т. 29, № 3, с. 40–45.

14. Гончаренко, М. С. Blackboard: потужний інструмент для аналітики. Науковий вісник, 2021. Т. 11, № 1, с. 25–30.
15. Левченко, А. В. Canvas: нові можливості для управління навчанням. Вища освіта в Україні, 2023. Т. 9, № 4, с. 55–60.
16. Ткаченко, М. С. Alteryx у вищій освіті: аналіз даних для прийняття рішень. Освіта і наука, 2020. Т. 10, № 3, с. 30–36.
17. Шевченко, І. Г. Використання штучного інтелекту в освітніх аналітиках: IBM Watson. Київ: Педагогічна академія, 2022, 270 с.
18. Федоренко, Р. О. Автоматизація навчального процесу: переваги та недоліки. Вісник Національної академії педагогічних наук України, 2023. Т. 12, № 5, с. 65–70.
19. Грищенко, Л. В. Інструменти прогнозування в освіті: виклики та можливості. Журнал освітніх технологій, 2021. Т. 8, № 2, с. 18–25.
20. Baker, R. S., & Yacef, K. The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining, 2009. Т. 1, № 1, с. 3–17.
21. Ranjan, A., & Sinha, S. Application of machine learning algorithms in educational data mining. International Journal of Educational Management, 2017. Т. 31, № 6, с. 1075–1092.
22. Paramitsiou, Z., & Economides, A. A. Learning analytics and educational data mining in practice: A systematic literature review of the last decade. Educational Technology & Society, 2014. Т. 17, № 4, с. 49–64.
23. Alam, F., & Noor, R. Predicting student success using machine learning techniques. Proceedings of the 2nd International Conference on Advanced Data and Information Engineering (ICADIE 2015), 2015, с. 1–6.
24. Kotsiantis, S. B. Use of machine learning techniques for educational purposes: A review. International Journal of Knowledge Management Studies, 2007. Т. 1, № 1, с. 1–18.

25. Hastie, T., Tibshirani, R., & Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009, 764 с.
26. Field, A. Discovering Statistics Using IBM SPSS Statistics. SAGE Publications Ltd, 2013, 952 с.
27. Breiman, L. Random Forests. Machine Learning, 2001. Т. 45, № 1, с. 5–32.
28. LeCun, Y., Bengio, Y., & Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998. Т. 86, № 11, с. 2278–2324.
29. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016, 800 с.
30. Hochreiter, S., & Schmidhuber, J. Long Short-Term Memory. Neural Computation, 1997. Т. 9, № 8, с. 1735–1780.
31. Siemens, G. Learning Analytics: The Emergence of a New Science. In Learning, Education, and Training, 2013, 200 с.
32. Kizilcec, R. F., Piech, C., & Schneider, E. F. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. Proceedings of the Third International Conference on Learning Analytics and Knowledge, 2013. с. 170–179.
33. Kumar, V., & Singh, S. Predictive models for student performance in educational data mining. International Journal of Computer Applications, 2015. Т. 116, № 11, с. 27–31.
34. Zhang, D., & Ma, Z. Data quality in data mining: A comprehensive review. Data Mining and Knowledge Discovery, 2019. Т. 33, № 3, с. 726–753.
35. Doshi-Velez, F., & Kim, B. Towards a rigorous science of interpretable machine learning. Proceedings of the 31st Conference on Neural Information Processing Systems, 2017, 100 с.
36. Zou, J. Y., & Schiebinger, L. AI can be sexist and racist – it’s time to make it fair. Nature, 2018. Т. 559, № 7714, с. 324–326.

ДОДАТОК А

Текст програмної реалізації основних функцій

```
# Основна функція для обробки багатосторінкових Excel-файлів
def excel_to_dataframe(file, study_year_start, study_year_end):
    groups = [sheet_name for sheet_name in file.sheet_names if re.match(r'^\d+m?$', sheet_name)]
    data = []

    for group in groups:
        df = file.parse(group)
        specialty = detect_specialty(group)
        course = detect_course(group)

        metadata_all = collect_subject_metadata(df)
        student_data_all = collect_student_data(df)
        additional_grades_all = collect_additional_grades(df)

        for semester in range(0, len(metadata_all)):
            metadata = metadata_all[semester]
            student_data = student_data_all[semester]
            additional_grades = additional_grades_all[semester]

            merged_dataset = pd.merge(student_data, metadata.transpose(), on='subject', how='inner')
            merged_dataset = pd.merge(merged_dataset, additional_grades, on='student_name',
            how='left')
            merged_dataset["study_year_start"] = study_year_start
            merged_dataset["study_year_end"] = study_year_end
            merged_dataset["semester"] = semester + 1
            merged_dataset["specialty"] = specialty
            merged_dataset["course"] = course
            data.append(merged_dataset)

    return pd.concat(data, axis=0)

# Функція для передбачення майбутніх оцінок
def predict_future_grades(data, target_subject):
    X = data.drop(columns=['student_name', target_subject])
    y = data[target_subject]

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

    model = LinearRegression()
    model.fit(X_train, y_train)

    predictions = model.predict(X_test)

    mae = mean_absolute_error(y_test, predictions)
```

```
r2 = r2_score(y_test, predictions)

model_performance = {
    'MAE': mae,
    'R2': r2
}

print(f'MAE: {mae:.2f}, R2: {r2:.2f}')
return predictions, model_performance

# Функція кластеризації з застосуванням PCA
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df)

pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)

kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(data_scaled)

pca_df = pd.DataFrame(data_pca, columns=['PC1', 'PC2'], index=df.index)
pca_df['Кластер'] = clusters

plt.figure(figsize=(10, 6))
sns.scatterplot(x=pca_df['PC1'], y=pca_df['PC2'], hue=pca_df['Кластер'], palette='Set2', s=100)
plt.title("Кластеризація студентів по оцінкам з застосуванням PCA")
plt.xlabel("Перша головна компонента (PC1)")
plt.ylabel("Друга головна компонента (PC2)")
plt.grid(True)

for i, txt in enumerate(pca_df.index):
    plt.annotate(txt, (pca_df['PC1'][i], pca_df['PC2'][i]), fontsize=10, ha='right')

plt.legend(title='Кластер')
plt.show()
```