

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматики
Кафедра комп'ютерних наук

Бакалаврська кваліфікаційна робота на тему:

ПРОГРАМНИЙ МОДУЛЬ ПРОГНОЗУВАННЯ УСПІШНОСТІ СТУДЕНТІВ
НА ОСНОВІ РЕЗУЛЬТАТІВ НАЦІОНАЛЬНОГО
МУЛЬТИПРЕДМЕТНОГО ТЕСТУ

Виконав: студент 4 курсу, групи 4КН-216
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

Наумов О. О.
(прізвище та ініціали)

Керівник: к.ф.-м.н., доцент каф. КН

Шевчук О. Ф.
(прізвище та ініціали)

« 04 » червня 2025 р.

Рецензент: к.т.н., доцент каф. САІТ

Горячев Г. В
(прізвище та ініціали)

« 05 » червня 2025 р.

Допущено до захисту
Зав. кафедри КН Григорієв А. І.
« 06 » червня 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А. А.
« 05 » травня 2025 р.

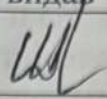
ЗАВДАННЯ

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Наумову Олегу Олександровичу

1. Тема роботи: Програмний модуль прогнозування успішності студентів на основі результатів національного мультипредметного тесту
Керівник роботи: Шевчук Олександр Федорович, к.ф.-м.н., доц.
Затверджені наказом вищого навчального закладу « 20 » 03 2025 року № 91
2. Термін подання студентом роботи 30 травня 2025 року
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; мінімальна кількість результатів тестування – 400; мінімальна кількість показників для прогнозування успішності студентів – 4; можливість проведення візуалізації та статистичної оцінки вхідних даних; тип інтерфейсу користувача інтуїтивно-зрозумілий.
4. Зміст текстової частини (перелік питань, які потрібно розробити): провести аналіз предметної області та сучасних програмних засобів прогнозування академічної успішності здобувачів освіти; спроектувати та програмно реалізувати модуль прогнозування успішності студентів; виконати тестування програмного модуля та розробити інструкцію користувача.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): UML-діаграма удосконаленого алгоритму прогнозування успішності студентів; структура програмного модуля; діаграма компонентів програмного модуля прогнозування успішності здобувачів вищої освіти; діаграма класів програмного модуля; діаграма послідовностей прогнозування успішності здобувачів вищої освіти; результати тестування розробленого програмного модуля.

6. Консультанти розділів роботи

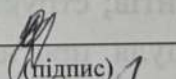
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Шевчук О.Ф., к.ф.-м.н, доцент каф. КН		

7. Дата видачі завдання 05 травня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області та сучасних програмних засобів прогнозування академічної успішності студентів	05.05.25	06.05.25	
2	Огляд існуючих методів прогнозування успішності здобувачів вищої освіти	04.05.25	11.05.25	
3	Проектування програмного модуля прогнозування успішності студентів	12.05.25	15.05.25	
4	Програмна реалізація модуля прогнозування успішності студентів на основі результатів національного мультипредметного тесту	16.05.25	26.05.25	
5	Оформлення матеріалів до захисту БКР		04.06.25	

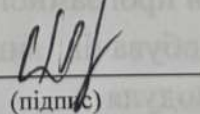
Студент


(підпис)

Наумов О.О.

(прізвище та ініціали)

Керівник роботи


(підпис)

Шевчук О.Ф.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 76 сторінок формату А4, на яких є 22 рисунки, 6 таблиць, список використаних джерел містить 22 найменування.

Бакалаврська кваліфікаційна робота присвячена розробці програмного модуля прогнозування академічної успішності здобувачів вищої освіти з урахуванням результатів національного мультипредметного тесту (НМТ). Проведено аналіз чинників, що впливають на успішність студентів, а також сучасних програмних рішень у сфері освітньої аналітики. Розроблено програмний модуль, архітектура якого забезпечує гнучкість, модульність та адаптованість до національного контексту. Модуль реалізовано мовою Python із застосуванням бібліотек машинного навчання та засобів візуалізації. Результати тестування підтвердили ефективність розробленого рішення, його розширений функціонал у порівнянні з наявними аналогами, а також можливість локального використання. Система підтримує індивідуалізоване прогнозування та може бути інтегрована в освітні процеси для підвищення ефективності академічної підтримки здобувачів освіти.

Ключові слова: програмний модуль, прогнозування успішності, національний мультипредметний тест, машинне навчання, моделювання даних, Python.

ABSTRACT

The bachelor's qualification thesis comprises 76 A4 pages, including 22 figures, 6 tables, and a list of 22 references.

The thesis is devoted to the development of a software module for predicting the academic performance of higher education students based on the results of the National Multi-Subject Test (NMT). The study includes an analysis of factors influencing student success, as well as an overview of existing software solutions in the field of educational analytics.

A software module was developed, featuring an architecture that ensures flexibility, modularity, and adaptability to the national context. The module is implemented in Python using machine learning and data visualization libraries.

Testing results confirmed the effectiveness of the proposed solution, its extended functionality compared to existing analogs, and its capability for local deployment. The system supports individualized prediction and can be integrated into educational processes to enhance academic support for students.

Keywords: software module, academic performance prediction, National Multi-Subject Test, machine learning, data modeling, Python.

ЗМІСТ

ВСТУП	4
1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ	6
1.1 Аналіз предметної області та основних факторів, що впливають на успішність здобувачів вищої освіти	6
1.2 Порівняльний аналіз сучасних програмних засобів прогнозування академічної успішності здобувачів освіти	8
1.3 Огляд існуючих методів прогнозування успішності здобувачів вищої освіти	17
1.4 Постановка задачі	21
1.5 Висновок до розділу 1	22
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ СТУДЕНТІВ	23
2.1 Розробка удосконаленого алгоритму роботи програмного модуля.....	23
2.2 Розробка архітектури програмного модуля та UML-моделюван ня процесів	27
2.3 Висновок до розділу 2	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ СТУДЕНТІВ	32
3.1 Обґрунтування вибору мови програмування	32
3.2 Обґрунтування вибору середовища розробки.....	36
3.3 Підготовка навчального датасету та статистичний аналіз вхідних даних.....	38
3.4 Розробка функціоналу програмного модуля	44
3.5 Тестування програмного модуля	47
3.6 Висновок до розділу 3	51
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55

Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи	58
Додаток Б (обов'язковий) Лістинг програми	59
Додаток В (обов'язковий) Ілюстративна частина.....	69
Додаток Г (довідниковий) Інструкція користувача	75

Вступ

Актуальність досліджень. Освіта є одним із ключових чинників розвитку сучасного суспільства, а якість навчального процесу значною мірою визначає майбутні перспективи молоді. В умовах зростаючих вимог до фахівців та необхідності адаптації до швидких змін ринку праці особливої важливості набувають механізми оцінювання та прогнозування успішності здобувачів вищої освіти. Одним із інструментів для визначення рівня знань вступників є національний мультипредметний тест (НМТ), результати якого впливають на подальшу академічну кар'єру здобувачів вищої освіти.

Проте самі по собі бали НМТ не дають вичерпної відповіді на питання про майбутню успішність здобувачів вищої освіти у закладах вищої освіти. Важливим завданням є розробка методів, що дозволяють на основі отриманих даних передбачати подальші академічні результати та допомагати у коригуванні освітньої траєкторії. Використання методів аналізу даних та машинного навчання відкриває нові можливості для прогнозування успішності здобувачів вищої освіти, дозволяючи не лише оцінювати їхні перспективи, а й розробляти рекомендації щодо підтримки їхнього навчання.

Актуальність дослідження зумовлена необхідністю створення ефективних інструментів прогнозування, які можуть використовуватися як в освітніх установах, так і на рівні державної освітньої політики. Розробка програмного модуля для прогнозування успішності здобувачів вищої освіти на основі результатів НМТ сприятиме оптимізації навчального процесу, індивідуалізації підходів до навчання та підвищенню загального рівня підготовки фахівців.

Метою дослідження є розширення функціональних можливостей програмного модуля прогнозування академічної успішності студентів за рахунок використання даних національного мультипредметного тесту.

Об'єктом дослідження є процеси прогнозування академічної успішності здобувачів вищої освіти.

Предметом дослідження є алгоритми та програмні засоби прогнозування академічної успішності здобувачів вищої освіти.

Задачі дослідження:

1. Аналіз предметної області та основних факторів впливу на показники успішності здобувачів вищої освіти.
2. Порівняльний аналіз програмних засобів прогнозування академічної успішності здобувачів освіти.
3. Проектування програмного модуля прогнозування академічної успішності здобувачів вищої освіти.
4. Програмна реалізація програмного модуля прогнозування академічної успішності здобувачів вищої освіти
5. Тестування програмного модуля прогнозування успішності студентів на основі результатів національного мультипредметного тесту.

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

1.1 Аналіз предметної області та основних факторів, що впливають на успішність здобувачів вищої освіти

У зв'язку з постійними змінами в освітній сфері, зокрема у вимогах до абітурієнтів та методах оцінки їхньої підготовленості, необхідність створення ефективних інструментів для прогнозування академічної успішності здобувачів вищої освіти стає дедалі актуальнішою.

Прогнозування рівня успішності дає змогу не лише отримати точну картину потенціалу здобувачів, а й ефективно планувати ресурси для їхньої підтримки. Зокрема, це дозволяє більш обґрунтовано розподіляти академічні групи, виявляти потенційні труднощі при вивченні певних дисциплін та своєчасно впроваджувати додаткові заходи підтримки, такі як додаткові курси для покращення результатів навчання або, навпаки, поглиблювати знання за допомогою наукових гуртків. Цей підхід також дає можливість більш точно індивідуалізувати навчальні траєкторії, що в результаті може підвищити загальний рівень підготовки здобувачів та оптимізувати навчальний процес [1].

Аналіз факторів, що впливають на успішність здобувачів вищої освіти, є важливою складовою для розробки точних моделей прогнозування. У наукових дослідженнях було виявлено кілька основних факторів, які суттєво впливають на академічні досягнення студентів. Серед них важливими є соціально-економічний статус, рівень мотивації, підготовка в школі, психоемоційний стан, а також навчальне середовище [1-6].

Одним із найбільш обговорюваних факторів є соціально-економічний статус здобувачів [3, 4]. Дослідження показують, що студенти з родин з високим соціально-економічним статусом мають більше можливостей для отримання додаткової освіти, доступу до репетиторів, а також використання інших ресурсів для покращення академічних результатів. Наприклад, у роботі [4] показано, що

студенти з сімей з вищим доходом, як правило, мають більш високі результати в навчанні завдяки кращому доступу до ресурсів освіти. Крім того, мотивація здобувачів вищої освіти та їхня здатність до самоорганізації також визначають рівень успішності. Високий рівень мотивації сприяє кращим академічним досягненням, оскільки студенти більш активно залучаються до навчального процесу і готові до подолання труднощів [5, 6].

Важливим фактором є підготовка здобувачів у школі. Відомо, що якість освіти, отриманої в школі, безпосередньо впливає на академічні результати вищих навчальних закладів. За результатами досліджень [7, 8], шкільні успіхи, зокрема середній бал атестату, є вагомим індикатором подальшої успішності в навчанні. Водночас психоемоційний стан здобувачів відіграє важливу роль, оскільки стрес, депресія чи інші психічні розлади можуть знижувати рівень концентрації і впливати на якість виконання навчальних завдань [9].

Окрім цих загальних факторів, існують показники, які можуть бути виміряні та використані безпосередньо в прогностичних моделях. Одним з таких є результат національного мультипредметного тесту (НМТ). НМТ є важливим етапом для вступу до вищих навчальних закладів, і його результат є одним з основних показників, що характеризує рівень знань здобувача вищої освіти на момент вступу [7]. Оскільки НМТ складається з чотирьох предметів, три з яких є обов'язковими (українська мова, математика, історія України), його результати дають загальне уявлення про академічну підготовленість студента.

Іншим вимірюваним фактором є середній бал атестату про повну загальну середню освіту. Середній бал атестату є важливим індикатором успішності здобувача під час навчання в школі. Він відображає загальний рівень знань і старання здобувача, що може впливати на його здатність до успішного навчання у вищих навчальних закладах.

Не менш важливим фактором є місцевість проживання здобувача. За результатами досліджень, студенти, що проживають у великих містах, як правило, мають кращі умови для отримання освіти, більший доступ до додаткових навчальних ресурсів, таких як репетитори та освітні курси, ніж студенти з

сільських районів [10]. Цей фактор також важливо враховувати при створенні моделей прогнозування, оскільки здобувачі з різних регіонів можуть мати різні стартові умови для успішного навчання [11].

Інші фактори, такі як стать, вік, тип навчального закладу, який здобувач закінчив, а також участь у позакласних заходах, також можуть впливати на успішність здобувачів вищої освіти. Проте всі ці фактори потребують детальнішого вивчення і аналізу для того, щоб визначити їхній конкретний вплив на академічні досягнення.

Отже, для побудови прогнозної моделі успішності здобувачів вищої освіти необхідно враховувати не лише загальні фактори, як-то соціально-економічний статус або мотивація, але й конкретні вимірювані показники, що відображають рівень академічної підготовленості здобувачів, такі як результат НМТ, середній бал атестату, місцевість проживання тощо. Тільки комплексний підхід до аналізу цих факторів дозволить створити ефективну та точну модель для прогнозування успішності здобувачів вищої освіти.

1.2 Порівняльний аналіз сучасних програмних засобів прогнозування академічної успішності здобувачів освіти

У сучасних умовах цифровізації освіти зростає потреба у використанні програмних засобів для прогнозування академічної успішності здобувачів освіти. Такі системи допомагають навчальним закладам аналізувати потенціал студентів, визначати ризики академічної неуспішності та впроваджувати своєчасні коригувальні заходи. Проведемо порівняльний аналіз найвідоміших освітніх платформ та програмних рішень, що використовуються для прогнозування успішності здобувачів освіти, їх функціональних можливостей та підходів до моделювання освітніх результатів.

Civitas Learning [12] – це потужна платформа, орієнтована на аналіз і прогнозування успішності студентів у вищих навчальних закладах (рис. 1.1). Вона використовує технології Big Data та машинного навчання для допомоги

університетам та коледжам в управлінні успішністю студентів, покращенні результатів навчання та зниженні рівня відсіву студентів. Система дозволяє навчальним закладам отримувати точніші прогнози щодо успішності студентів і надавати персоналізовану підтримку для підвищення академічних досягнень.

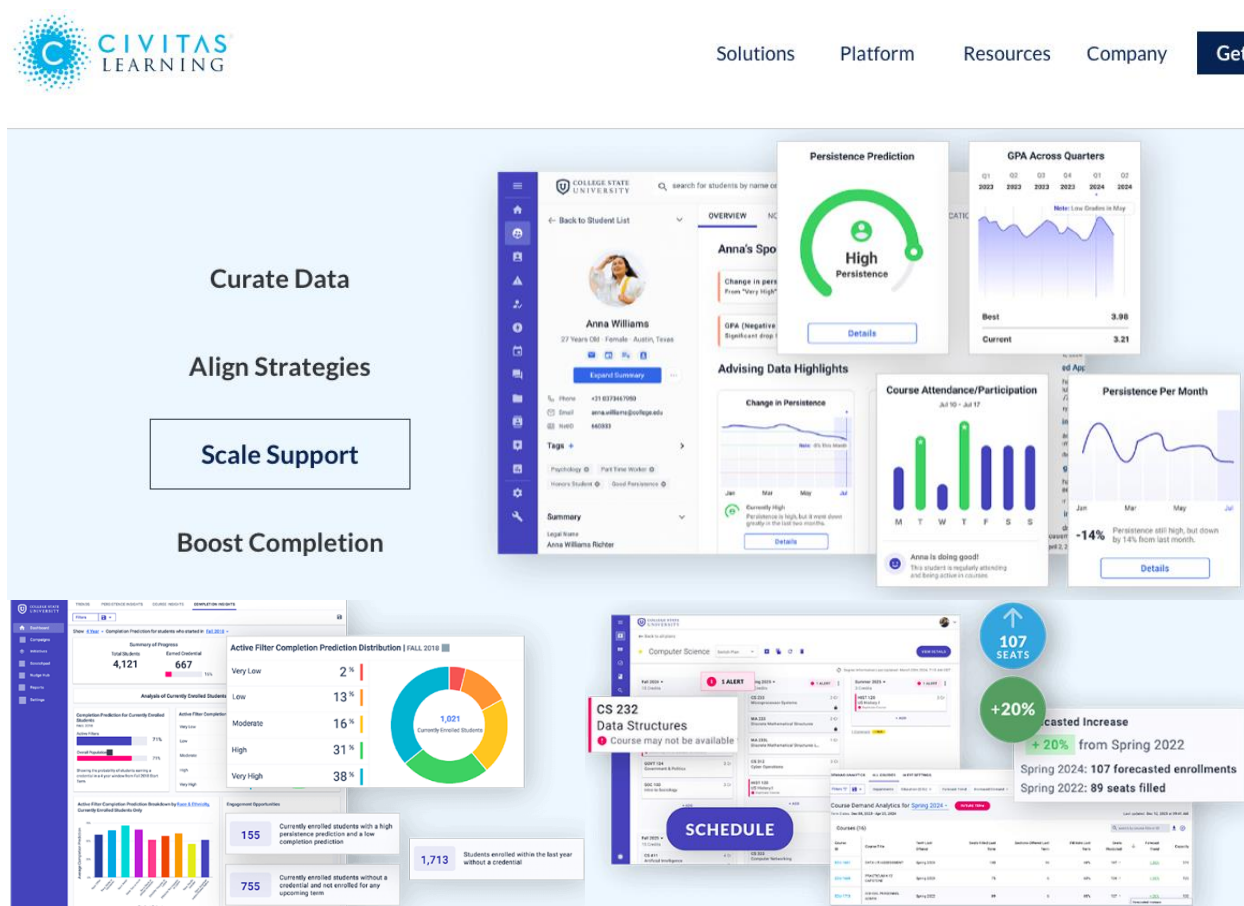


Рисунок 1.1 – Екранна форма платформи Civitas Learning

Наведемо основні можливості платформи Civitas Learning:

1. Аналіз великих даних (Big Data). Платформа використовує технологію Big Data для обробки великих обсягів інформації про студентів. Ці дані включають академічні результати, відвідуваність, участь у додаткових заходах та соціальних мережах університету, а також інші фактори, що впливають на успішність. Платформа дозволяє виявити патерни та залежності, які можуть бути використані для прогнозування результатів навчання.

2. Прогнозування академічної успішності. Однією з основних функцій Civitas Learning є здатність прогнозувати майбутні академічні результати

студентів. Платформа аналізує дані про студентів (їхні оцінки, поведінкові фактори, участь у навчальних заходах) і на основі цього створює прогнози щодо ймовірності того, що студент досягне успіху в певних дисциплінах або в навчанні загалом.

3. Система раннього попередження. Платформа також надає можливість створювати системи раннього попередження для виявлення студентів, які можуть бути на шляху до відсіву або низьких академічних результатів. При цьому використовуються такі індикатори, як низька відвідуваність, низькі оцінки на початкових етапах навчання, відсутність взаємодії з навчальними матеріалами тощо.

4. Індивідуальні рекомендації для студентів. Civitas Learning використовує алгоритми для надання персоналізованих рекомендацій студентам. На основі їхнього навчального досвіду та профілю система може поради курси, програми або навчальні стратегії, які найбільше відповідають їхнім потребам і можливостям.

5. Покращення взаємодії між студентами та викладачами. Платформа надає інструменти для покращення взаємодії між студентами, викладачами та адміністрацією. Наприклад, за допомогою Civitas Learning студенти можуть отримувати відгуки про свої досягнення та отримувати поради щодо подальшого розвитку. Викладачі можуть отримати інформацію про те, де студенти мають труднощі, що дозволяє оперативно коригувати методику викладання.

6. Підвищення рівня утримання студентів. Однією з головних цілей використання Civitas Learning є покращення рівня утримання студентів у навчальних закладах. Платформа дозволяє виявити тих студентів, які можуть залишити навчальний заклад через низькі результати або інші проблеми. Використовуючи прогнозні моделі та дані, можна впровадити програми підтримки для цих студентів, такі як менторські програми, консультації з кар'єрним розвитком або психо-соціальну допомогу.

7. Аналіз ефективності навчальних стратегій. Civitas Learning також може використовуватися для оцінки ефективності різних навчальних стратегій та

програм. Завдяки аналітиці можна визначити, які курси, методи навчання або ресурси найкраще підходять для студентів з різними потребами та якості їхніх результатів. Це дозволяє університетам удосконалювати свої навчальні плани, враховуючи успішність студентів.

Blackboard Analytics [13] – це потужна система аналітики для вищих навчальних закладів, яка забезпечує аналіз даних про студентів, дозволяючи адміністраціям, викладачам та іншому персоналу приймати обґрунтовані рішення щодо поліпшення навчального процесу (рис. 1.2). Blackboard Analytics використовує методи машинного навчання та штучного інтелекту для прогнозування успішності студентів, покращення взаємодії з ними та оптимізації управління навчальним процесом.

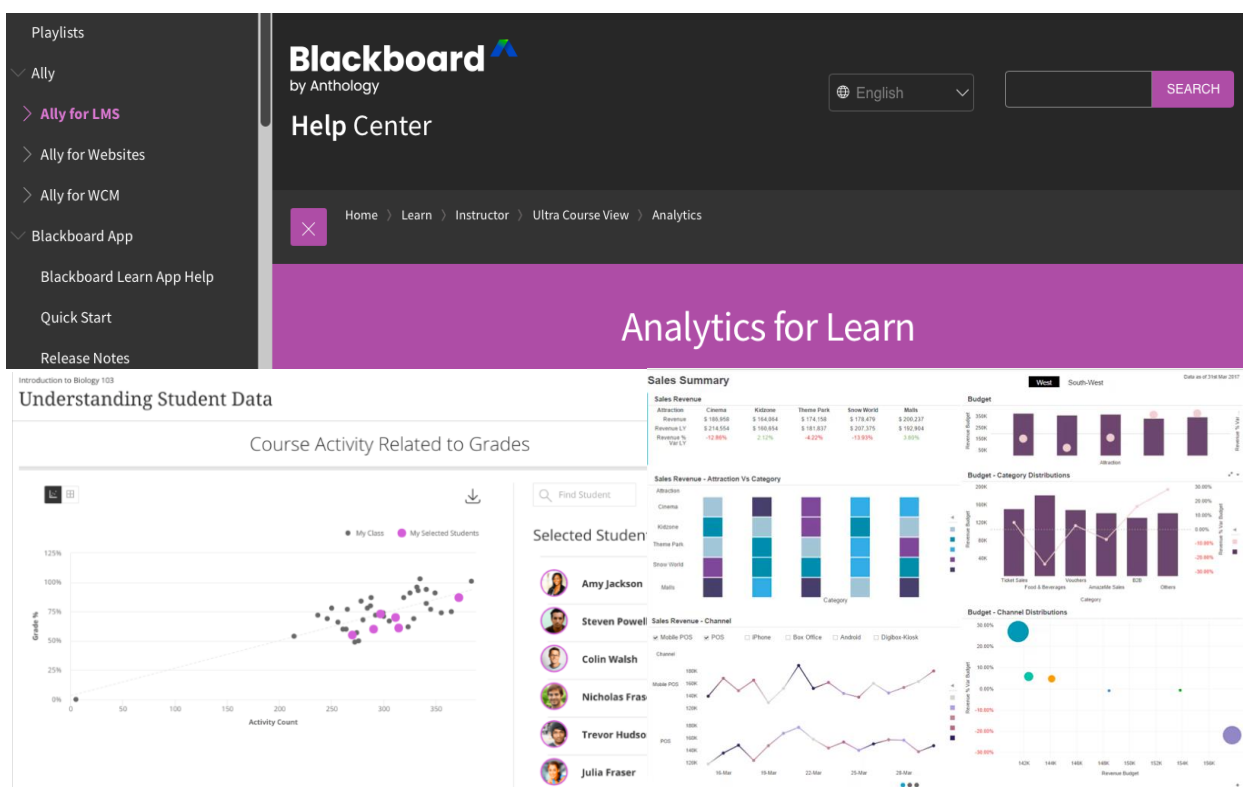


Рисунок 1.2 – Екранна форма платформи Blackboard Analytics

Розглянемо основні можливості Blackboard Analytics [13]:

1. Аналіз даних активності студентів. Blackboard Analytics дозволяє збирати та аналізувати різноманітні дані про студентів, включаючи їхні оцінки, активність у навчальних системах, участь у групових проектах, відвідуваність занять та

взаємодію з навчальними матеріалами. Це дає змогу створити детальну картину про кожного студента, виявити потенційні проблеми в навчанні та своєчасно надавати необхідну підтримку.

2. Прогнозування успішності студентів. Однією з основних функцій Blackboard Analytics є прогнозування академічної успішності студентів. Використовуючи дані про їхню діяльність, систему оцінок і взаємодію з навчальними матеріалами, платформа створює прогнози щодо ймовірності того, що студент отримає низькі оцінки або може мати проблеми з певними предметами. Ці прогнози дозволяють викладачам і адміністрації своєчасно вжити заходів для надання підтримки тим студентам, які ризикують відстати.

3. Раннє виявлення ризиків. Blackboard Analytics використовує систему раннього попередження для виявлення студентів, які можуть мати проблеми в навчанні, такі як низька активність на платформі, низькі оцінки або відсутність взаємодії з курсами, що дозволяє швидко реагувати на потенційні проблеми, надаючи студентам додаткову підтримку.

4. Моніторинг ефективності викладання. Платформа також дозволяє навчальним закладам оцінювати ефективність методів викладання. Blackboard Analytics аналізує, як зміни в методиках викладання або навчальних матеріалах впливають на успішність студентів.

5. Індивідуалізація навчання. Blackboard Analytics сприяє персоналізації навчального процесу, допомагаючи визначити, які ресурси або підходи до навчання будуть найефективнішими для кожного студента. Залежно від результатів аналізу, система може рекомендувати студентам додаткові ресурси, тренінги або навчальні курси для покращення їхніх результатів. Це дозволяє створити індивідуальні навчальні траєкторії, що оптимізують процес навчання.

6. Підвищення взаємодії між студентами та викладачами. Blackboard Analytics дозволяє викладачам краще розуміти прогрес своїх студентів, виявляти слабкі місця в навчанні та надавати індивідуальну допомогу. Платформа також дозволяє студентам отримувати зворотний зв'язок щодо їхніх досягнень, що стимулює їх до покращення результатів. Крім того, система може допомогти

викладачам краще організовувати навчальні заняття, виходячи з того, як студенти взаємодіють із навчальним матеріалом.

7. Прогнозування й управління утриманням студентів. Однією з головних цілей Blackboard Analytics є підвищення рівня утримання студентів у навчальному закладі. Використовуючи інструменти для прогнозування ймовірності відрахування студентів, система допомагає виявити тих, хто має ризик залишити університет або не завершити своє навчання. Це дозволяє вчасно організувати програми підтримки, менторство або інші заходи, спрямовані на збереження студентів і покращення їхньої успішності.

8. Інтеграція з іншими системами. Blackboard Analytics інтегрується з іншими навчальними системами, такими як системи управління навчанням (LMS), онлайн-курси тощо.

Panorama Education [14] – це аналітична платформа, орієнтована на освіту, яка використовує дані для покращення академічних результатів, моніторингу успішності та розвитку студентів (рис. 1.3).

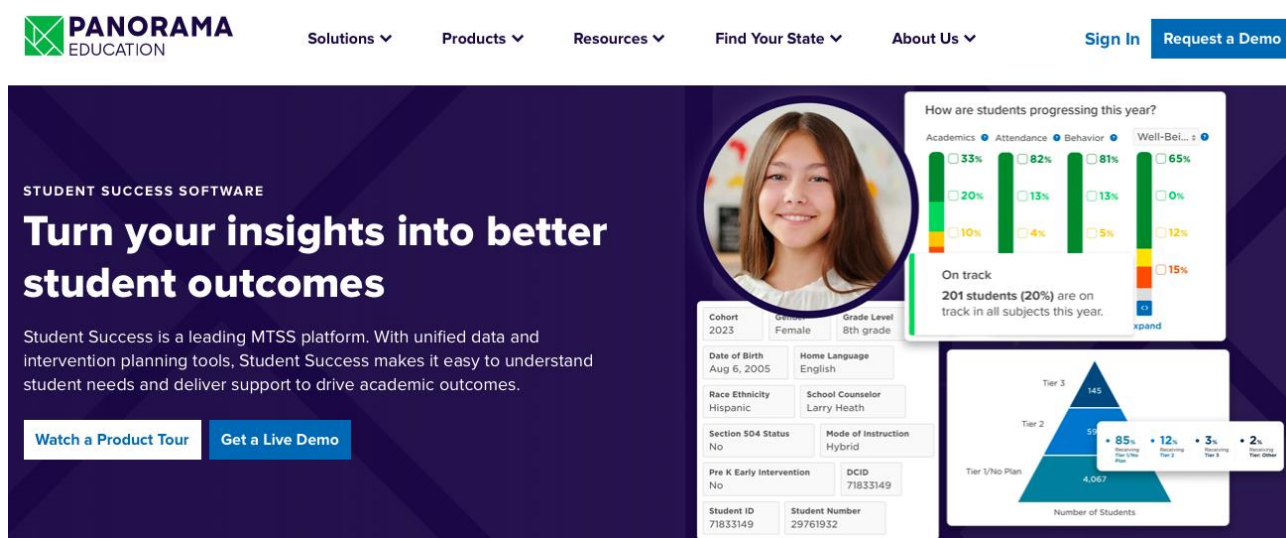


Рисунок 1.3 – Екранна форма платформи Panorama Education

Платформа надає інструменти для збору, аналізу та інтерпретації даних, що дозволяють адміністраціям шкіл, вчителям і іншим освітнім працівникам розуміти потреби учнів і приймати обґрунтовані рішення щодо підвищення рівня

їхньої успішності. Основна мета Panorama Education – створення персоналізованого підходу до кожного студента і покращення освітнього процесу [15]. Основними можливостями Panorama Education є:

1. Збір та аналіз даних про студентів. Обробка даних включає оцінки студентів, їхню відвідуваність, поведінкові показники, соціально-емоційні аспекти та участь у додаткових програмах. Зібрані дані дозволяють створити детальну картину успішності та потреб кожного студента, відстежувати зміни в прогресі студентів у часі та швидко виявляти потенційні проблеми.

2. Прогнозування академічних результатів. Платформа використовує прогностичні моделі для оцінки ймовірності академічних результатів студентів. Це дає змогу адміністраціям навчальних закладів і викладачам прогнозувати успіх або можливі труднощі, з якими студенти можуть зіткнутися в майбутньому. Використовуючи ці прогнози, можна заздалегідь надавати підтримку тим студентам, які потребують додаткової уваги.

3. Інструменти для моніторингу соціально-емоційного розвитку. Однією з унікальних особливостей Panorama Education є можливість відстежувати соціально-емоційний розвиток учнів. Платформа дозволяє збирати відгуки від учнів, вчителів і батьків щодо емоційного стану, соціальних навичок і загального благополуччя студентів, що дозволяє не лише відслідковувати академічний прогрес, але й визначати потенційні соціально-емоційні проблеми, які можуть вплинути на успішність студента.

4. Персоналізовані звіти та рекомендації. Платформа генерує детальні звіти, які дозволяють розуміти потреби кожного студента. Звіти містять персоналізовані рекомендації щодо покращення успішності учнів, включаючи варіанти додаткових занять, підтримки чи адаптаційних стратегій.

5. Інструменти для підвищення утримання студентів. Panorama Education має вбудовані інструменти для виявлення студентів, які можуть зіткнутися з труднощами або мати високий ризик залишити навчальний заклад. Використовуючи дані про їхню успішність, участь у навчальних заходах і поведінку, система прогнозує ризики відсіву і дає змогу своєчасно втручатися,

пропонуючи додаткову підтримку. Це допомагає підвищити рівень утримання студентів і знизити відсів.

6. Аналітика для адміністрації закладів освіти. Panorama Education надає адміністраціям шкіл та вищих навчальних закладів глибокий аналіз академічної та соціально-емоційної ситуації в закладі в цілому. Використовуючи аналітичні звіти, адміністрація може оцінювати ефективність різних програм, визначати тренди в успішності учнів, аналізувати вплив певних змін на загальну продуктивність і приймати стратегічні рішення для покращення освітнього процесу.

7. Моніторинг прогресу та адаптація планів. Завдяки можливості відстежувати прогрес студентів у реальному часі, Panorama Education дозволяє вчителям і адміністраціям адаптувати навчальні плани і стратегії на основі отриманих даних. Це дозволяє оперативно змінювати підходи до навчання залежно від того, як студенти реагують на нові методи або матеріали, а також коригувати навчальні траєкторії, якщо це необхідно.

У таблиці 1.1 наведено порівняльний аналіз основних характеристик та функціональних можливостей розглянутих освітніх платформ.

Таблиця 1.1 – Порівняльний аналіз функціональних можливостей розглянутих освітніх платформ

Функціональна можливість	Civitas Learning	Blackboard Analytics	Panorama Education
1	2	3	4
Прогнозування академічної успішності	Використовує прогностичні моделі на основі поведінки студентів, оцінок та академічних досягнень	Прогнозування успішності на основі активності студентів, оцінок та інших показники	Прогнозує академічні результати з врахування емоційного та соціального стану студентів
Інтеграція з даними вступних іспитів	Не враховує дані вступних випробувань	Не враховує дані вступних випробувань	Не враховує дані вступних випробувань
Аналіз поведінкових даних	Враховує взаємодію студентів з навчальними матеріалами, відвідуваність та інші фактори	Аналізує активність студентів в LMS (відвідуваність, участь у навчальних активностях)	Моніторить поведінку студентів у соціальних та навчальних контекстах

Продовження табл. 1.1

1	2	3	4
Соціально-емоційна аналітика	Обмежена, акцент на академічних даних	Мінімальна, фокусується переважно на академічних аспектах	Глибокий аналіз соціально-емоційного стану студентів та його вплив на успішність
Індивідуалізація прогнозів для студентів	Індивідуальні прогнози успішності, що враховують історію академічних результатів та поведінки	Персоналізація через аналіз даних LMS для передбачення ризиків та можливих проблем у навчанні	Персоналізовані прогнози з урахуванням академічних та соціальних факторів
Прогнозування ризиків відсіву	Прогнозує ризик відставання студентів, виявляючи проблеми на ранніх етапах	Прогнозує ризики відсіву на основі даних LMS та активності студентів	Прогнозує не тільки академічний, а й соціально-емоційний ризик відсіву
Визначення студентів, які потребують додаткової підтримки	Визначення проблемних студентів на основі прогнозів успішності та участі в навчальному процесі	Визначення студентів, що мають ризик відсіву на основі аналізу їх активності в LMS	Визначення студентів, які мають труднощі як в академічній сфері так і у соціальному та емоційному контекстах
Раннє виявлення проблем у навчанні	Прогностичні моделі раннього виявлення студентів, що потребують підтримки	Раннє виявлення студентів із високим ризиком відсіву через аналіз даних LMS	Виявляє студентів, що мають емоційні чи соціальні проблеми
Прогнозування з урахуванням результатів вступних іспитів	Не враховує дані вступних випробувань	Не враховує дані вступних випробувань	Не враховує дані вступних випробувань
Гнучкість у налаштуванні прогнозних моделей	Висока гнучкість, можливість налаштування моделей під конкретні потреби закладу	Висока гнучкість, дозволяє налаштувати прогностичні моделі відповідно до закладу освіти	Гнучкість на середньому рівні, з можливістю адаптації під соціально-емоційні аспекти

Узагальнюючи дані наведені в табл. 1.1, можна зробити висновок, що кожна з розглянутих платформ забезпечує потужні інструменти для прогнозування академічної успішності студентів. Вони використовують різноманітні джерела даних, такі як академічні показники, взаємодія студентів з навчальним контентом, соціально-емоційні аспекти та інші чинники, що дозволяють індивідуалізувати підхід до підтримки студентів та своєчасно виявляти потенційні проблеми. Однак усі ці платформи не враховують в прогнозуванні успішності результати вступних випробувань, що є важливим етапом оцінки підготовленості абітурієнтів.

Залучення результатів вступних іспитів (наприклад національного мультипредметного тесту (НМТ), що проводиться в Україні), до аналітичних

моделей цих платформ значно розширить їхні функціональні можливості. Така інтеграція дозволить забезпечити більш точні прогнози успішності студентів, оскільки вступні результати є важливим індикатором рівня академічної підготовленості абітурієнтів і можуть значною мірою впливати на їх подальші досягнення в навчанні. Включення цього фактору в прогностичні моделі дозволить створити комплексніші та точніші системи підтримки студентів, зменшити ризики відсіву і покращити загальну ефективність навчального процесу.

1.3 Огляд існуючих методів прогнозування успішності здобувачів вищої освіти

Прогнозування успішності здобувачів вищої освіти є складною задачею, що вимагає врахування багатьох факторів, зокрема академічних, соціально-економічних та поведінкових. Існуючі методи прогнозування можна умовно поділити на три основні групи: статистичні методи, методи машинного навчання та експертні системи [16]. У цьому розділі розглянемо кожен з них, а також їх застосування у сфері освіти.

1. Статистичні методи прогнозування. Статистичні методи є основою для аналізу освітніх даних і широко застосовуються у дослідженнях прогнозування академічної успішності. Вони дозволяють встановлювати кореляції між різними факторами, оцінювати тренди та будувати регресійні моделі. Основні статистичні методи, що застосовуються у цій сфері:

- лінійна та множинна регресія. Використовується для моделювання залежності між залежною змінною (успішністю здобувачів вищої освіти) та однією чи кількома незалежними змінними (результати НМТ, середній бал атестату, соціально-економічний статус тощо). Регресійний аналіз допомагає визначити, які фактори мають найбільший вплив на прогнозовану успішність;

- кореляційний аналіз. Дозволяє встановити зв'язки між вхідними факторами (наприклад, оцінками НМТ, середнім балом атестату) та показниками

успішності під час навчання. Наприклад, дослідження показують, що результати стандартизованих тестів часто корелюють із подальшою академічною успішністю;

- дискримінантний аналіз. Метод, що використовується для класифікації здобувачів вищої освіти за рівнем прогнозованої успішності. Він допомагає визначити, до якої категорії може належати студент: успішний, середній чи такий, що перебуває у зоні ризику відрахування;

- логістична регресія. Використовується для прогнозування ймовірності певної події, наприклад, чи зможе здобувач завершити навчання або ж ризикує відрахуванням;

Отже, загалом можна відзначити, що статистичні методи мають перевагу в простоті реалізації та інтерпретації, однак вони менш ефективні у випадках, коли дані мають складні нелінійні залежності.

2. Методи машинного навчання. Методи машинного навчання значно розширюють можливості прогнозування академічної успішності, оскільки вони можуть обробляти великі обсяги даних та враховувати складні залежності між змінними:

- дерева рішень (Decision Trees). Метод, який поділяє дані на групи за допомогою логічних умов, створюючи структуру у вигляді дерева. Наприклад, можна побудувати дерево рішень, яке прогнозує ймовірність успішності на основі результатів НМТ, середнього балу атестату та інших факторів;

- метод опорних векторів (SVM – Support Vector Machine). Використовується для класифікації студентів за рівнем успішності, створюючи гіперплощину, що розділяє їх на відповідні групи;

- нейронні мережі (Artificial Neural Networks, ANN). Потужний метод для складного нелінійного прогнозування, який імітує роботу людського мозку. Нейронні мережі можуть навчатися на великих масивах даних і визначати найважливіші фактори, що впливають на успішність. Вони ефективно використовуються в адаптивних системах навчання;

- методи ансамблевого навчання (Random Forest, Gradient Boosting).

Поєднують кілька моделей для покращення точності прогнозування. Наприклад, Random Forest використовує багато дерев рішень для покращення точності прогнозів;

- кластеризація (K-Means, DBSCAN). Використовується для групування студентів за подібними характеристиками, наприклад, для виявлення студентів з потенційними труднощами у навчанні.

Методи машинного навчання є ефективними для прогнозування, однак вони потребують великих обсягів навчальних даних та складніші в інтерпретації, ніж статистичні методи.

3. Експертні системи – це програмні комплекси, що використовують бази знань і правила для імітації рішень експертів у певній галузі. У контексті прогнозування успішності здобувачів освіти вони можуть використовувати:

- продукційні системи правил. Визначають логіку прогнозування на основі експертних знань. Наприклад, "Якщо результат НМТ високий і середній бал атестату > 9 , то висока ймовірність успішного навчання";

- Байєсові мережі. Використовують теорію ймовірностей для аналізу залежностей між факторами. Наприклад, можна оцінювати ризик відрахування на основі ймовірнісної моделі, що враховує всі можливі варіанти;

- системи, що ґрунтуються на штучному інтелекті та експертних рекомендаціях. Такі системи можуть інтегрувати як статистичні методи, так і машинне навчання, доповнюючи їх експертними оцінками для покращення точності прогнозування.

Експертні системи можуть бути корисними для адаптації до конкретних освітніх умов, однак їх розробка є складною та потребує значної кількості знань від фахівців.

Загалом можна відзначити, що кожен з розглянутих методів має свої переваги та обмеження. А тому, з метою вибору найоптимальнішого підходу до прогнозування рівня успішності здобувачів вищої освіти, у таблиці 1.2 наведено узагальнене порівняння основних методів з урахуванням мети та задач дослідження.

Таблиця 1.2 – Порівняльна таблиця основних методів прогнозування успішності здобувачів вищої освіти

Критерій	Статистичні методи	Методи машинного навчання	Експертні системи
1	2	3	4
Складність реалізації	Низька – прості у впровадженні та інтерпретації	Висока – потребують налаштування та великих обсягів даних	Висока – вимагають створення правил та бази знань
Точність прогнозів	Середня – працюють добре для лінійних залежностей	Висока – здатні знаходити складні залежності	Висока – за умови правильної експертної бази
Гнучкість	Обмежена – ефективні лише для простих моделей	Висока – можуть адаптуватися до різних сценаріїв	Висока – можуть бути налаштовані для конкретних ситуацій
Обсяг даних для навчання	Малий – достатньо невеликих вибірок	Великий – потребують великої кількості навчальних даних	Середній – базуються на експертних опитуваннях
Інтерпретованість	Висока – результати легко пояснюються	Низька – складно пояснити, чому модель дала певний прогноз	Висока – рішення базуються на зрозумілих правилах
Можливість обробки соціальних факторів	Обмежена – переважно числові показники	Висока – можуть враховувати широке коло факторів	Висока – дозволяють вбудовувати експертні знання
Можливість врахування НМТ	Так – можуть використовувати бали НМТ та середній бал атестату	Так – можуть будувати складні залежності між НМТ, середнім балом, місцем проживання тощо	Так – можуть включати правила, що враховують НМТ
Можливість адаптації	Низька – модель потребує перебудови при зміні вхідних факторів	Висока – навчаються на нових даних	Висока – можна змінювати правила прогнозування
Відповідність задачам дослідження	Так, як базовий метод аналізу даних	Так, але лише у випадку великого обсягу вхідних даних	Частково, як додатковий метод для адаптації прогнозів

Аналізуючи дані неведені в табл. 1.2 можна відзначити, що статистичні методи найкраще підходять для аналізу кореляцій та трендів за умови малочисельності вибірки, хоча й мають обмежену здатність до роботи з нелінійними залежностями. Методи машинного навчання дозволяють створювати складні прогностичні моделі з високою точністю, але потребують значних обсягів

даних і обчислювальних ресурсів. Їхнє використання обґрунтоване лише при створенні комплексних моделей прогнозування, що враховують як поточні показники навчання здобувачів освіти так і їхні попередні результати вступних випробувань. Експертні системи можуть враховувати людський досвід та знання, але при цьому їхня розробка та впровадження є досить ресурсомісткою.

1.4 Постановка задачі

На основі проведеного аналізу предметної області для розробки програмного модуля прогнозування успішності здобувачів вищої освіти застосуємо математичну модель у вигляді:

$$Y = f(X), \quad (1.1)$$

де: Y – вектор вихідних даних, що представляє результати прогнозування успішності здобувача освіти з окремих навчальних дисциплін 1 курсу закладу вищої освіти та його загальної рейтингової оцінки;

$X(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$ – вектор вхідних даних, де:

x_1 – загальний конкурсний бал абітурієнта;

x_2 – результат НМТ з математики;

x_3 – результат НМТ з української мови;

x_4 – результат НМТ з історії України;

x_5 – результат НМТ з вибіркової дисципліни;

x_6 – середній бал атестату про здобуття повної загальної середньої освіти;

x_7 – показник, що визначає стать абітурієнта;

x_8 – місцевість проживання здобувача (місто чи село).

Таким чином, розширення функціональних можливостей модуля прогнозування успішності здобувачів вищої освіти за рахунок врахування результатів національного мультипредметного тесту сприятиме вдосконаленню прогностичних моделей та підвищенню точності оцінки.

1.5 Висновок до розділу 1

У першому розділі проведено аналіз предметної області прогнозування академічної успішності здобувачів освіти на основі результатів національного мультипредметного тесту. Розглянуто основні фактори, що впливають на успішність навчання, зокрема вплив вступних випробувань, середнього балу атестата та соціально-економічних умов. На основі порівняльного аналізу сучасних програмних засобів прогнозування академічної успішності визначено їхні переваги та недоліки. Встановлено, що більшість існуючих рішень зосереджуються на аналізі успішності під час навчання, проте не враховують результати вступних випробувань як ключовий фактор прогнозування. На основі проведеного аналізу сформульовано постановку задачі та окреслено напрямки розширення функціональних можливостей програмного модуля, зокрема через інтеграцію прогнозування успішності на основі результатів НМТ, середнього балу атестата та інших факторів.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ СТУДЕНТІВ

2.1 Розробка удосконаленого алгоритму роботи програмного модуля

Програмний модуль прогнозування успішності здобувачів вищої освіти базується на аналізі вхідних даних, зокрема результатів національного мультипредметного тесту (НМТ), середнього балу атестату, місцевості проживання та інших чинників, що можуть впливати на академічну успішність. Його основна мета – забезпечити ефективний інструмент для освітніх установ, що дозволяє прогнозувати потенційну успішність здобувачів, оптимізувати навчальні процеси та своєчасно виявляти необхідність додаткової підтримки.

Таким чином програмний модуль має забезпечувати такі основні функціональні вимоги:

1. Збір та обробка вхідних даних – отримання даних від абітурієнтів або освітніх установ, включаючи результати НМТ, середній бал атестату, регіон проживання та інші фактори.

2. Попередня обробка даних – нормалізація, видалення пропущених значень та приведення до єдиного формату для подальшого аналізу.

3. Аналіз взаємозв'язків між показниками – виявлення кореляцій між успішністю у навчанні та вхідними параметрами.

4. Прогнозування успішності – застосування статистичних методів, машинного навчання або експертних систем для визначення ймовірності успішного навчання у першому семестрі або році.

5. Генерація аналітичних звітів – надання візуалізації результатів прогнозування у вигляді графіків, таблиць та аналітичних висновків.

6. Інтерактивний інтерфейс – реалізація веб-або десктопного застосунку, що дозволяє взаємодіяти з системою у зручному форматі.

7. Можливість інтеграції – підтримка експорту даних у форматах CSV, Excel або JSON для подальшого використання у навчальних аналітичних системах.

Окрім основних функцій, програмний модуль має відповідати наступним вимогам: продуктивність (швидка обробка даних навіть при великій кількості запитів); масштабованість (можливість роботи з розширеними наборами даних та адаптація до нових змін у вступних випробуваннях); безпека (захист персональних даних відповідно до вимог законодавства про захист інформації); доступність (можливість роботи на різних пристроях та підтримка адаптивного інтерфейсу; зручність використання (інтуїтивний дизайн, що дозволяє швидко освоїти роботу з системою без спеціальної підготовки) [17].

Також відзначимо, що програмний модуль буде розроблятися для українських освітніх установ з урахуванням специфіки НМТ. Дані для аналізу надаються у вигляді масиву історичних показників успішності студентів попередніх років. При цьому система не буде приймати управлінські рішення, а лише надавати рекомендації на основі прогнозів.

У даному випадку доступних даних може бути недостатньо для використання складних моделей машинного навчання, таких як градієнтний бустинг чи нейронні мережі. Тому доцільно обмежитися лінійними моделями, які є більш інтерпретованими, менш вибагливими до обсягу даних і дозволяють робити прогнози навіть на основі невеликої вибірки. Переваги лінійної регресії у даному випадку: простота реалізації та обчислювальна ефективність; можливість інтерпретації впливу кожного фактора на успішність; стійкість до обмеженої кількості даних, що є критичним у разі недостатнього набору історичних записів; відсутність необхідності складного налаштування гіперпараметрів.

Удосконалений алгоритм прогнозування успішності здобувачів освіти що наведено на рис. 2.1 складається з таких основних етапів:

- Збір та обробка вхідних даних. Отримання результатів НМТ, середнього балу атестату, інформації про місцевість проживання. Попередня обробка даних: перевірка на пропуски, нормалізація значень, кодування категоріальних змінних (якщо необхідно).

- Формування математичної моделі. Побудова моделі лінійної регресії у вигляді рівняння:

$$Y = \beta_0 + \beta_1 \cdot x_1 + \beta_1 \cdot x_1 + \dots + \beta_n \cdot x_n + \varepsilon, \quad (2.1)$$

де Y – прогнозована успішність, x_i – значення вхідних факторів, β_i – коефіцієнти регресії, ε – випадкова похибка.

Навчання моделі на доступних даних із використанням методу найменших квадратів.

- Оцінка якості моделі. Використання коефіцієнта детермінації R^2 для оцінки якості прогнозу. Перевірка залишків моделі на нормальність розподілу та відсутність автокореляції.

- Формування прогнозу. Застосування побудованої моделі для розрахунку ймовірної успішності здобувачів освіти. Інтерпретація отриманих результатів, визначення факторів, що мають найбільший вплив на прогноз.

- Візуалізація та представлення результатів. Побудова графіків залежностей успішності від основних факторів. Формування аналітичного звіту з висновками та рекомендаціями.

Напрями удосконалень:

- Додавання нових ознак. Якщо аналіз помилок показує, що важливих факторів не враховано (наприклад, відсутність даних про тип навчального закладу), нові ознаки додаються у вхідні дані.

- Оптимізація гіперпараметрів. Виконується перебір значень параметрів моделі (вибір методів регуляризації або зміна вагових коефіцієнтів).

- Зміна підходу до обробки даних. Якщо виявлено викиди або необхідність трансформації змінних (наприклад, нормалізація або категоризація), застосовуються відповідні методи.

- Оцінка покращень. Якщо зміни дали покращений прогноз, модель використовується. Якщо покращити прогноз не вдалося, обирається найкраща з можливих моделей.

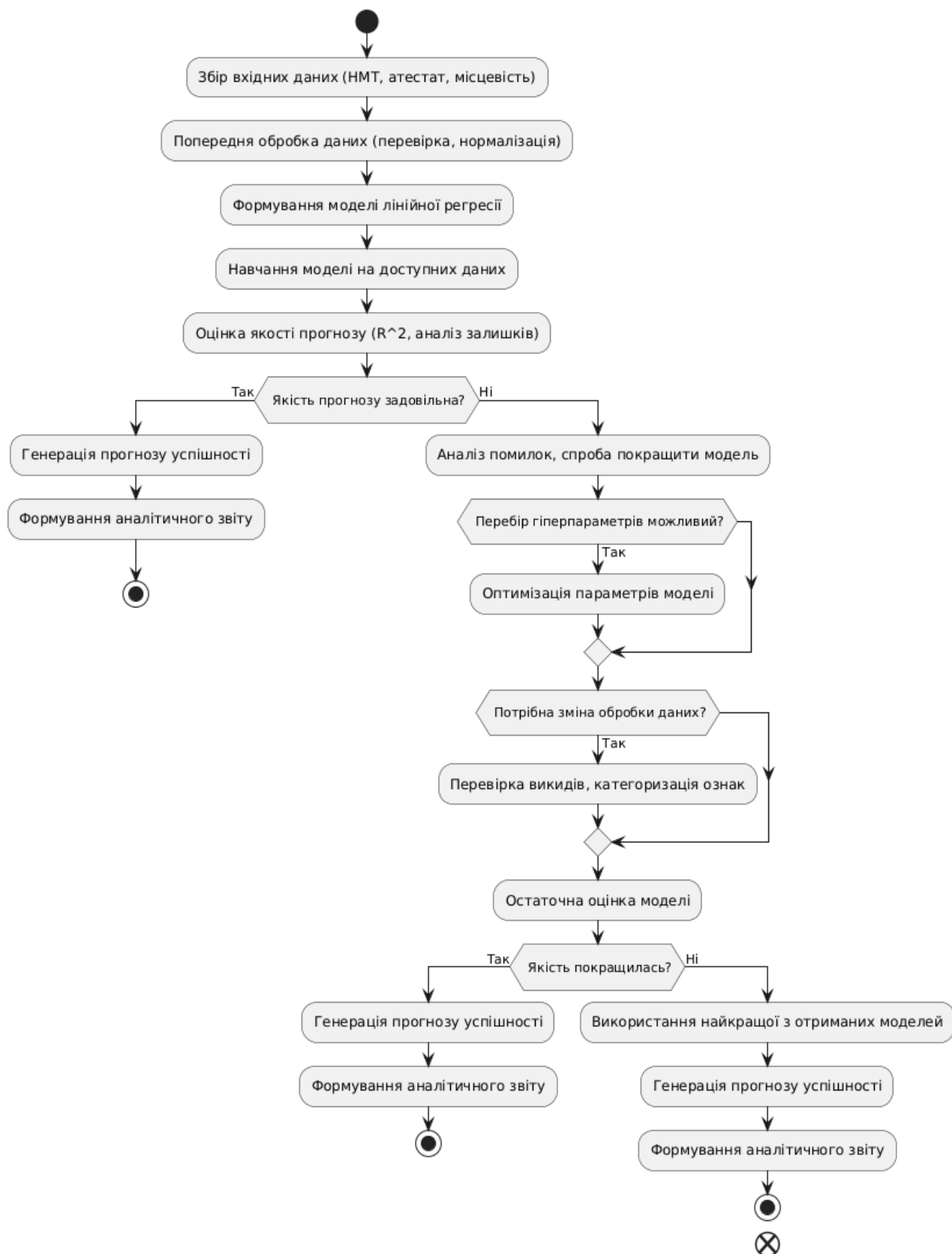


Рисунок 2.1 – UML-діаграма удосконаленого алгоритму прогнозування успішності здобувачів освіти

2.2 Розробка архітектури програмного модуля та UML-моделювання процесів

Програмний модуль прогнозування успішності здобувачів освіти базується на клієнт-серверній архітектурі. Це дозволяє забезпечити гнучкість, масштабованість та можливість інтеграції з іншими освітніми системами.

В рамках проектування програмного модуля прогнозування успішності здобувачів освіти було розроблено основні UML-діаграми, які відображають структуру та функціональність системи. Це дозволяє наочно представити взаємозв'язки між компонентами, класами та їхніми методами, а також продемонструвати алгоритм взаємодії різних частин модуля [17].

1. Діаграма компонентів (рис. 2.2). Діаграма компонентів використовується для відображення логічної архітектури програмного модуля. Вона показує, як основні компоненти системи взаємодіють між собою, які модулі існують та які залежності між ними.

Діаграма містить три основні пакети:

- Клієнтська частина – відповідає за введення вихідних даних користувачем та відображення прогнозу. Вона взаємодіє з API-запитами, які передають дані на сервер.

- Серверна частина – приймає API-запити, обробляє їх у контролері, передає дані у модуль прогнозування, отримує результати прогнозу та зберігає їх у базі даних.

- Аналітика – включає модель прогнозування на основі лінійної регресії, яка використовується для обчислення успішності.

Клієнтська частина відправляє API-запити до серверної частини, яка обробляє інформацію та звертається до бази даних. Далі дані передаються у модуль аналізу, де відбувається обчислення прогнозу. Результати повертаються користувачу через інтерфейс.

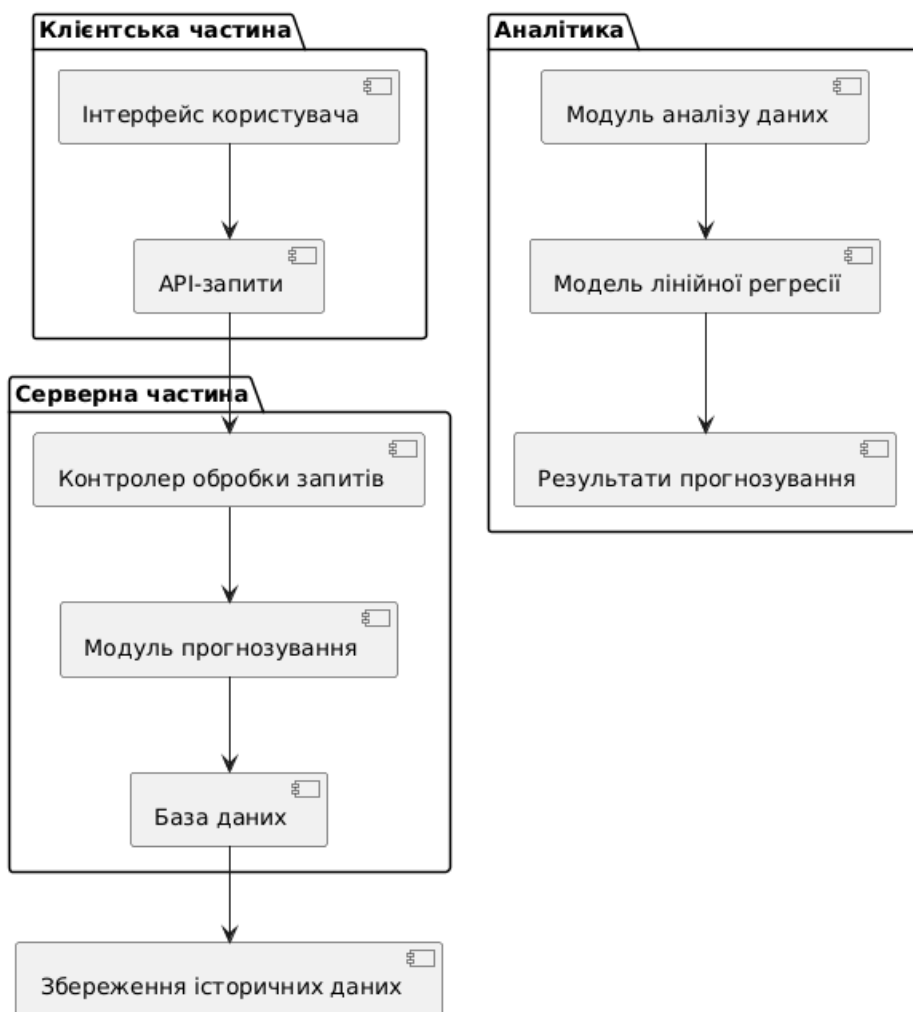


Рисунок 2.2 – Діаграма компонентів програмного модуля прогнозування успішності здобувачів вищої освіти

2. Діаграма класів (рис. 2.3). Діаграма класів зображує структуру об'єктно-орієнтованої моделі програмного модуля. Вона описує основні класи, їхні атрибути, методи та взаємозв'язки.

Наведемо опис діаграми класів.

- Клас Student (Здобувач освіти) – містить атрибути, що характеризують здобувача:
 - ID – унікальний ідентифікатор,
 - Name – ім'я,
 - nmt_score – результат НМТ,
 - highschool_gra – середній бал атестату,
 - region – місце проживання.

- Метод `predictPerformance()` викликає модель прогнозування для розрахунку успішності.
- Клас `PredictionModel` (Модель прогнозування) – реалізує методи навчання та прогнозування:
 - `trainModel(data: DataSet)` – тренує модель на історичних даних,
 - `predict(input: Student): float` – прогнозує успішність для нового студента.
- Клас `Database` (База даних) – містить методи роботи з інформацією:
 - `storeStudentData(student: Student)` – зберігає інформацію про студента,
 - `retrieveData(): DataSet` – отримує історичні дані для навчання моделі.

Інтерпретація. Об'єкти класу `Student` взаємодіють із `PredictionModel` для отримання прогнозу. `PredictionModel`, у свою чергу, звертається до `Database` для отримання історичних даних, необхідних для навчання.

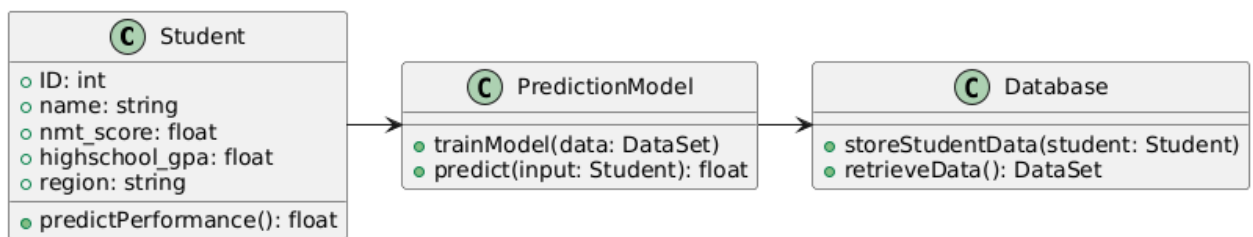


Рисунок 2.3 – Діаграма класів програмного модуля прогнозування успішності здобувачів вищої освіти

3. Діаграма послідовності (рис. 2.4). Діаграма послідовності відображає порядок взаємодії між користувачем та основними елементами програмного модуля. Вона демонструє послідовність виклику операцій у системі.

Опис діаграми:

- Користувач вводить вихідні дані – заповнює форму із результатами НМТ, середнім балом атестату та місцем проживання.
- Інтерфейс передає запит серверу – дані надсилаються через API.
- Сервер обробляє запит та звертається до бази даних – перевіряє, чи є історичні дані, та отримує їх.

- Сервер передає дані у модуль прогнозування – модель прогнозування використовує лінійну регресію для обчислення успішності.
- Сервер повертає результат інтерфейсу – відображається прогнозована успішність здобувача.

Інтерпретація. Діаграма показує, як дані проходять через всі рівні архітектури – від користувача до модуля прогнозування та назад. Це допомагає зрозуміти процес обробки даних у системі.

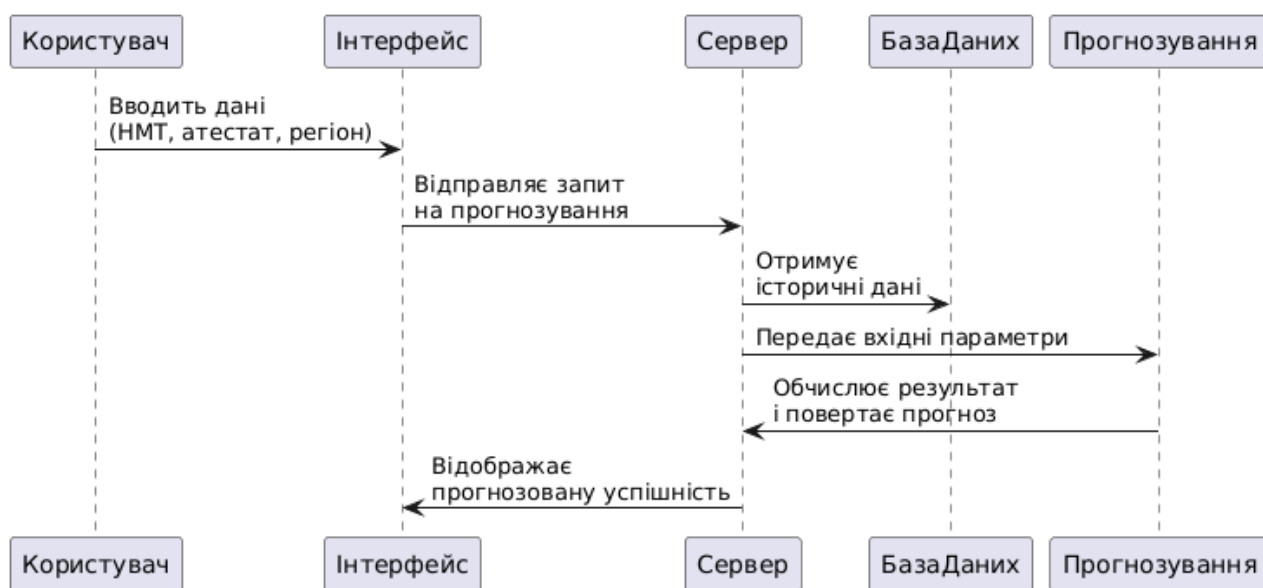


Рисунок 2.4 – Діаграма послідовностей прогнозування успішності здобувачів вищої освіти

Представлені UML-діаграми відображають ключові аспекти проектування програмного модуля. Вони показують взаємодію між компонентами, структурну організацію даних та послідовність виконання операцій. Це дозволяє ефективно розробляти та масштабувати систему, а також інтегрувати її в освітні платформи.

2.3 Висновок до розділу 2

У другому розділі роботи здійснено комплексне проектування програмного модуля прогнозування успішності здобувачів вищої освіти на основі результатів національного мультипредметного тесту. Розроблено алгоритм, який забезпечує

послідовність виконання операцій у системі, включаючи збір, обробку та аналіз вхідних даних, навчання моделі та формування прогнозу академічної успішності. Алгоритм став основою для побудови структури програмного модуля, визначаючи основні компоненти системи та їхні взаємозв'язки. Представлено діаграму послідовностей, яка моделює ключові сценарії функціонування системи, зокрема взаємодію між користувачем і модулями прогнозування, обробки даних та аналітики. Діаграма послідовностей дозволила візуалізувати порядок виконання дій і визначити логічні точки інтеграції між компонентами. Розроблено також діаграму класів, яка структурно описує основні елементи системи, їхні властивості, методи та взаємозв'язки. Вона забезпечила об'єктно-орієнтовану основу програмного модуля, що сприяє його подальшому розвитку та розширенню. Таким чином, у розділі проведено повний цикл проєктування, починаючи з розробки алгоритму прогнозування та закінчуючи деталізованою архітектурою програмного модуля. Отримані результати забезпечують логічну цілісність, модульність та відповідність вимогам, що створює основу для подальшої реалізації та впровадження системи прогнозування.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ СТУДЕНТІВ

3.1 Обґрунтування вибору мови програмування

Для розробки програмного модуля прогнозування успішності здобувачів вищої освіти на основі результатів національного мультипредметного тесту необхідно обрати таку мову програмування, яка забезпечить ефективну роботу з даними, побудову та навчання моделей прогнозування, інтеграцію з іншими компонентами програмного забезпечення, а також матиме розвинену екосистему бібліотек для аналізу даних та машинного навчання [18].

На сучасному етапі розвитку інформаційних технологій для вирішення задач подібного типу найбільш оптимальними є мови програмування Python, R та Julia.

Python є однією з найбільш популярних мов для задач обробки даних і машинного навчання. Вона має велику кількість бібліотек, таких як `scikit-learn`, `pandas`, `numpy`, `matplotlib`, які забезпечують усі необхідні інструменти для підготовки даних, побудови моделей, їх валідації та візуалізації результатів. Python має простий синтаксис, що пришвидшує розробку, та активну спільноту підтримки [19].

R є спеціалізованою мовою для статистичних обчислень та аналізу даних. Вона володіє потужним набором бібліотек для статистики, регресійного аналізу та візуалізації даних. Особливо добре R підходить для проведення глибокого статистичного аналізу. Проте її інтеграція в серверні рішення або створення веб-інтерфейсів є дещо складнішою порівняно з Python.

Julia – відносно нова мова програмування, орієнтована на високопродуктивні наукові обчислення. Вона забезпечує високу швидкість виконання коду, близьку до мови C. Julia має сучасний синтаксис і підтримує машинне навчання через бібліотеки типу `Flux.jl`. Однак на даному етапі екосистема Julia має менше доступних готових рішень для типових задач

машинного навчання, ніж Python або R.

У табл. 3.1 наведено узагальнений аналіз та порівняння розглянутих мов програмування.

Таблиця 3.1 – Порівняльний аналіз основних мов програмування, що відповідають поставленим задачам дослідження

Критерій	Мова програмування		
	Python	R	Julia
Призначення	Універсальна, акцент на ML та аналіз даних	Статистичний аналіз даних	Наукові обчислення, ML
Наявність бібліотек	Дуже висока (pandas, sklearn, TensorFlow)	Висока (caret, random Forest, ggplot2)	Помірна (Flux.jl, MLJ.jl)
Простота навчання	Висока	Середня	Середня
Візуалізація даних	Висока (matplotlib, seaborn)	Дуже висока (ggplot2)	Середня
Інтеграція у веб-сервіси	Легка (Flask, FastAPI)	Складна	Середня
Продуктивність	Середня	Середня	Висока
Спільнота підтримки	Дуже велика	Велика	Порівняно невелика
Зрілість екосистеми	Дуже висока	Висока	Помірна

Враховуючи поставлену задачу – побудову модуля прогнозування на основі обмеженої кількості даних (результати НМТ, середній бал атестату, місцевість проживання тощо) – ключовими критеріями вибору є наявність готових бібліотек для регресійного аналізу, простота інтеграції в прикладні системи та можливість швидкої побудови прототипу.

Мова програмування Python найкраще відповідає цим вимогам. Вона дозволяє швидко організувати весь життєвий цикл роботи над проектом – від обробки вхідних даних до побудови і оцінювання моделі прогнозування. Бібліотеки scikit-learn і statsmodels надають потужний інструментарій для

створення моделей лінійної регресії, що актуально з огляду на припущення про невеликий обсяг вхідних даних. Крім того, для візуалізації результатів можна ефективно використати бібліотеки `matplotlib` та `seaborn`.

Розглянемо детальніше основні бібліотеки та інструменти, що можуть бути використані під час розробки програмного модуля:

1. Бібліотека `pandas` є виступатиме стандартом для роботи з табличними даними у Python. Вона забезпечує гнучкі структури даних (об'єкти `DataFrame`, `Series`) і широкий набір інструментів для завантаження, очищення, перетворення та аналізу даних. `Pandas` дозволяє легко виконувати агрегацію, фільтрацію, злиття та інші операції над даними, що є критично важливим етапом перед побудовою моделей машинного навчання. У межах програмного модуля використовуватимуться такі основні компоненти:

- `pandas.DataFrame` для представлення даних у вигляді таблиць;
- функції `read_csv()` для завантаження даних з файлів;
- методи `dropna()`, `fillna()`, `astype()`, `merge()` для очищення і попередньої обробки даних;
- `describe()`, `corr()` для статистичного аналізу вхідних ознак.

2. `Scikit-learn` це одна з найпотужніших бібліотек Python для реалізації класичних алгоритмів машинного навчання, таких як регресія, класифікація, кластеризація та зниження розмірності даних. У межах даного проєкту бібліотека використовується для побудови моделей лінійної регресії, їхнього навчання та оцінювання якості прогнозування. Бібліотека також надає інструменти для розділення даних на тренувальні та тестові вибірки, стандартизації ознак і налаштування моделей. У програмному модулі будуть застосовані:

- `LinearRegression` із модуля `sklearn.linear_model` для побудови моделі лінійної регресії;
- `train_test_split` із модуля `sklearn.model_selection` для поділу даних на тренувальну та тестову вибірки;
- `mean_squared_error`, `r2_score` із модуля `sklearn.metrics` для оцінки точності моделі.

3. Statsmodels спеціалізується на побудові статистичних моделей, особливо для регресійного аналізу. На відміну від scikit-learn, ця бібліотека надає більше можливостей для детального аналізу результатів моделювання (наприклад, оцінки коефіцієнтів моделі, р-значення, довірчі інтервали), що важливо для інтерпретації результатів прогнозування в академічному контексті. Використовуватимуться:

- OLS (Ordinary Least Squares) з модуля statsmodels.api для побудови регресійної моделі;
- функція .summary() для отримання детальної статистики моделі (коефіцієнти, р-значення, довірчі інтервали).

4. Matplotlib є базовою бібліотекою для створення графіків і візуалізації даних у Python. Вона дозволяє будувати різноманітні типи діаграм – гістограми, графіки залежностей, коробчасті діаграми тощо. У межах проекту Matplotlib використовується для візуалізації розподілу даних, результатів прогнозування та оцінки якості моделі.

У програмному модулі будуть застосовані:

- matplotlib.pyplot для побудови графіків залежностей;
- функції plot(), scatter(), hist(), title(), xlabel(), ylabel() для візуалізації даних та результатів прогнозу.

5. Seaborn побудована на основі Matplotlib та дозволяє створювати більш складні й інформативні графіки з мінімальними зусиллями. Особливо корисна для побудови теплових карт кореляційних матриць, парних графіків ознак і порівняльних діаграм. Використовується для кращої ілюстрації залежностей між факторами (наприклад, балами НМТ, середнім балом атестата, місцевістю) та рівнем академічної успішності. Основні інструменти:

- seaborn.heatmap() для побудови кореляційної матриці факторів;
- seaborn.pairplot() для візуалізації парних залежностей між ознаками;
- seaborn.regplot() для побудови графіків регресії.

У табл. 3.2 структуровано наведено розглянуті бібліотеки та їхні ключові функції та інструменти, що можуть бути використані під час програмної реалізації

модуля прогнозування успішності здобувачів вищої освіти на основі НМТ.

Таблиця 3.2 – Основні бібліотеки, функції та інструменти аналізу

Бібліотека	Основне призначення	Ключові функції та інструменти
pandas	Обробка табличних даних	DataFrame, read_csv(), dropna(), fillna(), astype(), merge(), describe(), corr()
scikit-learn	Створення моделей машинного навчання і їх оцінка	LinearRegression, train_test_split, mean_squared_error, r2_score
statsmodels	Статистичне моделювання і детальний аналіз регресії	OLS, .summary()
matplotlib	Побудова базових графіків	plot(), scatter(), hist(), title(), xlabel(), ylabel()
seaborn	Розширена візуалізація даних	heatmap(), pairplot(), regplot()

3.2 Обґрунтування вибору середовища розробки

У процесі розробки програмного модуля прогнозування успішності здобувачів вищої освіти важливим етапом є вибір відповідного середовища розробки (IDE), яке забезпечує зручність написання, налагодження, тестування та розгортання програмного коду. Для цього було розглянуто три найбільш поширені середовища для Python-розробки: Jupyter Notebook, PyCharm та Visual Studio Code, кожне з яких має свої переваги та обмеження [20, 21].

Jupyter Notebook надає інтерактивне середовище, яке ідеально підходить для аналітики, візуалізації даних та експериментів із машинним навчанням. Воно дозволяє поєднувати код, вивід результатів і текстові пояснення в одному документі, що зручно для поетапного аналізу даних. Проте, Jupyter менш зручний для структурованої розробки великих програмних модулів і має обмежену підтримку відлагодження коду.

PyCharm є потужним середовищем, орієнтованим на повноцінну розробку Python-проектів. Воно має вбудовану підтримку систем контролю версій, тестування, аналізу коду, а також інструменти для роботи з базами даних.

PyCharm забезпечує високу продуктивність та ефективну організацію великих проєктів, однак вимагає більше ресурсів системи й може бути перевантаженим для початкових експериментів з даними.

Visual Studio Code – це легке, модульне середовище, яке дозволяє працювати з різними мовами програмування завдяки своїм розширенням. Воно підтримує інтеграцію з Git, термінал, дебагер і різні інструменти машинного навчання. Порівняно з PyCharm, VS Code має гнучкіше налаштування, але може потребувати додаткової конфігурації для повноцінної роботи з Python та ML-бібліотеками.

У табл. 3.3 наведено узагальнене порівняння основних середовищ розробки з урахуванням поставлених задач дослідження.

Таблиця 3.3 – Порівняльний аналіз основних середовищ розробки

Критерій	Jupyter Notebook	PyCharm	Visual Studio Code
Зручність аналізу даних	Висока – інтерактивність, миттєвий вивід	Середня – потребує більше налаштувань	Середня – залежить від встановлених плагінів
Підтримка машинного навчання	Висока – широко використовується в Data Science	Висока – має інтеграцію з ML-бібліотеками	Середня – потребує додаткових розширень
Зручність візуалізації результатів	Висока – вбудований рендер графіків	Середня – зовнішнє вікно, або плагіни	Середня – можливість через розширення
Організація проєкту	Низька – неструктурований код	Висока – підтримка структури проєкту	Висока – підтримка директорій та модулів
Вбудовані інструменти налагодження	Мінімальні	Потужні – дебагер, профайлер	Хороші – інтегрований дебагер
Вимоги до ресурсів	Низькі	Високі – особливо для великих проєктів	Низькі – працює швидко навіть на слабких ПК
Доступність та налаштування	Відкрите, проста установка через Anaconda або pip	Безкоштовна версія доступна, потребує установки	Безкоштовне, легко конфігурується

На основі проведеного у табл. 3.3 порівняння, основними критеріями якого були зручність аналізу даних, підтримка машинного навчання, зручність

візуалізації результатів, а також доступність і швидкість роботи, було прийнято рішення використовувати Jupyter Notebook на початковому етапі розробки та тестування моделі, а для структурування остаточного модуля – PyCharm Community Edition як основне середовище реалізації. Такий комбінований підхід дозволяє ефективно поєднати переваги обох середовищ: швидку аналітику та повноцінну реалізацію програмного модуля.

3.3 Підготовка навчального датасету та статистичний аналіз вхідних даних

Основними вхідними даними для побудови моделі прогнозування академічної успішності здобувачів вищої освіти виступають результати Національного мультипредметного тесту (НМТ). Ці дані є ключовим елементом вступної кампанії та містяться у Єдиній державній електронній базі з питань освіти (ЄДБО) – офіційній інформаційній системі, що забезпечує збір, обробку та зберігання освітніх даних на державному рівні. Результати НМТ є стандартизованими, що забезпечує об'єктивність у порівнянні здобувачів та дозволяє використовувати їх як достовірну основу для подальшого прогнозного аналізу. Саме ці показники, доповнені іншими чинниками (середній бал атестата, стать, тип місцевості, наявність пільг), використовуються як вхідні змінні в моделі прогнозування майбутньої академічної успішності студентів.

У межах вступної кампанії формування конкурсного бала (КБ) здійснюється за спеціальною формулою, яка включає результати національного мультипредметного тесту з трьох предметів; вагові коефіцієнти та можливі додаткові бали за досягнення або пільгові умови (рис. 3.1). На основі розрахованого КБ системою ЄДБО формується загальний рейтинговий список, на основі якого здійснюється зарахування абітурієнтів до відповідної спеціальності ЗВО (рис. 3.2). Таким чином, конкурсний бал акумулює найважливіші освітні характеристики вступника, що визначають його стартову академічну підготовку.

СЛІДКА

ЕЛЕКТРОННА

БАЗА ЗАЯВ НА ВСТУП

#ВСТУП-2024

Конкурсні пропозиції

Порівняння конкурсних пропозицій

Статистика

Попередні роки

Новини

Запитання-відповіді

ВСТУПНА КАМПАНІЯ 2024

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

БАКАЛАВР (основа вступу - повна загальна середня освіта)

Спеціальність

Назва пропозиції

Код конкурсної пропозиції

Освітня програма

Факультет

Форма навчання

Курс зарахування

Вид пропозиції

Термін навчання

Термін подачі заяв

Ліцензійний обсяг

Макс. обсяг держзамовлення

122 КОМП'ЮТЕРНІ НАУКИ

122 Комп'ютерні науки (Комп'ютерні науки)

1329464

Комп'ютерні науки

Факультет інтелектуальних інформаційних технологій та автоматизації

Денна

1 курс

Відкрита, з пріоритетом

01.09.2024 – 30.06.2028 (3р 10м)

19.07.2024 – 31.07.2024

109

51

КОНКУРСНІ ПРЕДМЕТИ

1 0.3 НМТ Українська мова ≥ 100

2 0.5 НМТ Математика ≥ 100

3 0.2 НМТ Історія України ≥ 100

0.2 НМТ Українська література В ≥ 100

0.3 НМТ Іноземна мова В ≥ 100

0.2 НМТ Біологія В ≥ 100

0.2 НМТ Географія В ≥ 100

0.4 НМТ Фізика В ≥ 100

0.2 НМТ Хімія В ≥ 100

5 0.3 ЗНО Українська мова ≥ 100

6 0.5 ЗНО Математика ≥ 100

0.3 ЗНО Іноземна мова В ≥ 100

0.2 ЗНО Історія України В ≥ 100

СТАТИСТИКА ЗАЯВ

470 Подано заяв

150.9 Сер. бал

41 Рекомендовано на загальних підставах

159.3 Мін. бал рекомендованих

458 Допущено до конкурсу

112.8 Мін. бал

44 Зараховано на бюджет

140.8 Мін. бал зарахованих на бюджет

356 Заяв на бюджет

195.3 Макс. бал

64 Зараховано на контракт

112.8 Мін. бал зарахованих на контракт

Інші конкурсні пропозиції закладу

УВАГА! Конкурсний бал (КБ) розраховується за такою формулою (пункт 2 розділу IX Порядку прийому на навчання для здобуття вищої освіти в 2024 році): $КБ = (K1 \times P1 + K2 \times P2 + K3 \times P3) / (K1 + K2 + K3) + ОУ$

Остаточко КБ множиться на регіональний (РК) та галузевий (ГК) коефіцієнти шляхом його множення на їх добуток

Рисунок 3.1 – Екранна форма системи ЄДБО

КОНКУРСНІ ЗАЯВИ									
Пошук за прізвищем									
#	Статус	ПР	ПВМ	ВЗ	КБ	Показники – Коефіцієнти – Предмети			
1	до наказу (бюджет)	1	✓	✓	163,846	+46.800 = 156 x 0.3	Українська мова		
					+81.500 = 163 x 0.5	Математика			
					+34.000 = 170 x 0.2	Історія України			
					+50.700 = 169 x 0.3	Іноземна мова			
						Квота 2			
2	до наказу (бюджет)	1	✓	✓	155,615	+49.500 = 165 x 0.3	Українська мова		
					+72.500 = 145 x 0.5	Математика			
					+33.800 = 169 x 0.2	Історія України			
					+46.500 = 155 x 0.3	Іноземна мова			
						Квота 1			
3	до наказу (бюджет)	1	✓	✓	155,077	+47.700 = 159 x 0.3	Українська мова		
					+74.000 = 148 x 0.5	Математика			
					+34.600 = 173 x 0.2	Історія України			
					+45.300 = 151 x 0.3	Іноземна мова			
						Квота 1			
4	до наказу (бюджет)	1	✓	✓	146,154	+45.000 = 150 x 0.3	Українська мова		
					+70.000 = 140 x 0.5	Математика			
					+30.000 = 150 x 0.2	Історія України			
					+45.000 = 150 x 0.3	Іноземна мова			
						Квота 1			
5	до наказу (бюджет)	2	✓	✓	142,923	+42.600 = 142 x 0.3	Українська мова		
					+72.500 = 145 x 0.5	Математика			
					+27.200 = 136 x 0.2	Історія України			
					+43.500 = 145 x 0.3	Іноземна мова			
						Квота 1			
6	до наказу (бюджет)	1	✓	✓	140,769	+42.900 = 143 x 0.3	Українська мова		
					+67.000 = 134 x 0.5	Математика			
					+30.200 = 151 x 0.2	Історія України			

Рисунок 3.2 – Формування рейтингового списку абітурієнтів системою ЄДБО

Враховуючи, що результати НМТ є закритою інформацією і не надаються у вільному доступі в повному обсязі, для розробки програмного модуля

прогнозування академічної успішності студентів було створено тестовий навчальний датасет, який імітує типову структуру даних вступників (рис. 3.3). Такий підхід дозволив побудувати функціональний прототип системи та протестувати працездатність обраної моделі прогнозування. Набір даних включає як основні показники вступної кампанії: результати НМТ з української мови, математики, історії України, конкурсний бал, наявність пільги так і додаткові показники якими є середній бал атестата, стать абітурієнта та тип місцевості (місто або село) навчання абітурієнта (рис. 3.4). Дані були згенеровані з урахуванням реалістичних статистичних розподілів, характерних для результатів реальних вступників, що дозволило забезпечити наближення до поведінки справжніх освітніх даних.

```
np.random.seed(42)
n = 500

nmt_ukrainian = np.random.normal(155, 25, n)
nmt_ukrainian = np.clip(nmt_ukrainian, 100, 200)
nmt_ukrainian[nmt_ukrainian == 200] = np.random.randint(180, 200, size=(nmt_ukrainian == 200).sum())

nmt_math = np.random.normal(162, 20, n)
nmt_math = np.clip(nmt_math, 100, 200)
nmt_math[nmt_math == 200] = np.random.randint(180, 200, size=(nmt_math == 200).sum())

nmt_history = np.random.normal(150, 25, n)
nmt_history = np.clip(nmt_history, 100, 200)
nmt_history[nmt_history == 200] = np.random.randint(150, 200, size=(nmt_history == 200).sum())
nmt_history[nmt_history == 100] = np.random.randint(100, 140, size=(nmt_history == 100).sum())

gender = np.random.choice(['чоловіча', 'жіноча'], n)
avg_certificate = np.random.uniform(6.0, 12.0, n)
locality = np.random.choice([0, 1], n) # 0 – міська, 1 – сільська
benefit = np.random.choice([0, 1], n, p=[0.85, 0.15])

K1, K2, K3 = 0.3, 0.5, 0.2
OU = np.random.uniform(0, 10, n)

KB = (K1 * nmt_ukrainian + K2 * nmt_math + K3 * nmt_history) / (K1 + K2 + K3) + OU
KB = np.clip(KB, 100, 200)

entrants_df = pd.DataFrame({
    'id': np.arange(1, n + 1),
    'nmt_ukrainian': np.round(nmt_ukrainian, 1),
    'nmt_math': np.round(nmt_math, 1),
```

Рисунок 3.3 – Фрагмент програмного коду генерації
тестового навчального датасету

	id	nmt_ukrainian	nmt_math	nmt_history	gender	avg_certificate	locality	benefit	contest_score	avg_sem1	avg_sem2	higher_math	programming
0	1	167.4	166.7	171.8	жіноча	9.59	0	0	173.33	78.5	87.4	88.6	76.4
1	2	151.5	167.5	169.2	жіноча	7.78	0	0	167.92	69.9	77.0	52.4	84.1
2	3	171.2	147.5	182.0	чоловіча	7.80	1	1	165.54	66.1	76.3	77.5	64.1
3	4	193.1	151.3	158.7	чоловіча	10.46	1	0	166.87	88.9	75.2	75.7	62.3
4	5	149.1	172.6	166.0	чоловіча	6.29	1	0	169.95	78.6	68.2	76.5	63.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	496	168.5	151.4	145.7	чоловіча	10.68	0	0	158.14	73.0	70.9	54.7	86.7
496	497	129.1	174.4	165.4	жіноча	8.75	0	0	166.09	83.6	63.4	75.3	51.0
497	498	150.2	184.6	167.5	жіноча	8.39	0	0	173.57	63.6	76.2	83.8	69.4
498	499	133.1	183.0	127.3	чоловіча	7.82	1	0	165.94	71.1	60.7	61.0	66.6
499	500	120.4	160.1	176.4	жіноча	6.39	0	1	155.19	75.6	59.1	71.0	50.7

500 rows x 13 columns

Рисунок 3.4 – Структура тестового навчального датасету

Окремо до цього датасету було додано змодельовані результати академічної успішності студентів у першому році навчання, що включають середній бал за I та II семестри, а також оцінки з базових навчальних дисциплін: вища математика, алгоритмізація та програмування. Формування цих значень відбувалося з урахуванням ймовірнісної залежності від рівня результатів НМТ – абітурієнти з вищими вступними балами частіше демонстрували кращі академічні результати, однак у даних також збережено роль випадкових коливань та впливу додаткових факторів.

Тестовий датасет слугує основою для подальшої побудови, навчання та валідації моделі прогнозування, а також дозволяє здійснювати початковий аналіз залежностей між характеристиками вступника та його потенційною успішністю у навчанні.

Також на етапі попередньої обробки даних було здійснено розширений статистичний аналіз синтетичного навчального датасету, метою якого було виявлення структурних закономірностей та потенційних залежностей між вхідними ознаками абітурієнтів та їхньою подальшою академічною успішністю. Аналіз розпочато з описової статистики (рис. 3.5), яка дозволила оцінити основні числові характеристики кожної змінної: середнє значення, медіану, мінімальні та максимальні значення, стандартне відхилення та квартилі. Ці параметри забезпечили загальне уявлення про розподіл даних, симетрію та можливу наявність викидів або значних відхилень.

	HMT_Українська мова	HMT_Математика	HMT_Історія	Конкурсний бал
count	500.0	500.0	500.0	500.0
mean	154.0	161.0	151.0	162.0
std	22.0	19.0	23.0	13.0
min	100.0	100.0	100.0	109.0
25%	137.0	148.0	135.0	154.0
50%	155.0	162.0	152.0	162.0
75%	171.0	175.0	168.0	171.0
max	199.0	200.0	199.0	200.0

Рисунок 3.5 – Описова статистика вхідних даних навчального датасету

Подальшим кроком була побудова візуалізацій – зокрема, гістограм розподілу результатів НМТ з української мови, математики та історії України (рис. 3.6) а також загального конкурсного балу (рис. 3.7). Завдяки цим графікам стало можливим візуально оцінити щільність балів, виявити скупчення значень у певних інтервалах та проаналізувати, чи відповідає розподіл змінних очікуваній нормі. На основі побудованих графіків можна стверджувати, що результати розподілені досить наближено до нормального, хоча спостерігалися окремі крайні значення, що було передбачено під час генерації.

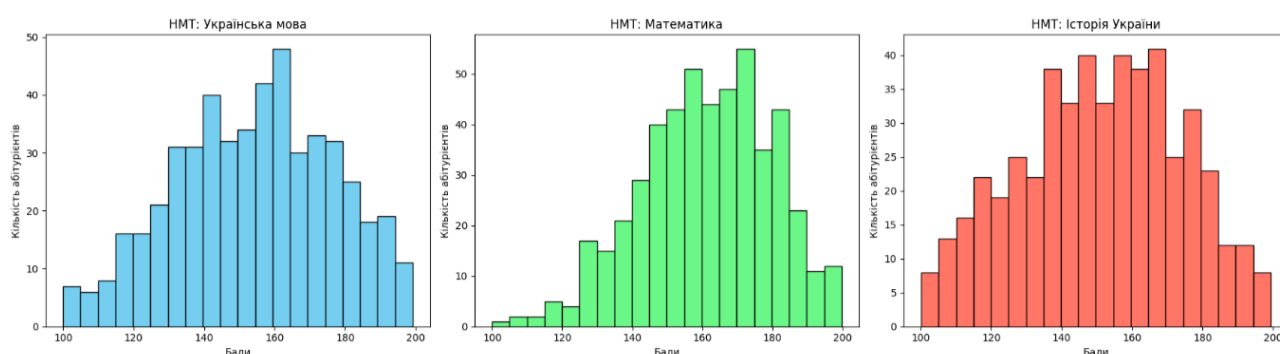


Рисунок 3.6 – Гістограми розподілу результатів НМТ з української мови, математики та історії України

Для глибшого розуміння взаємозв'язків між змінними був проведений кореляційний аналіз із використанням коефіцієнта Пірсона ().

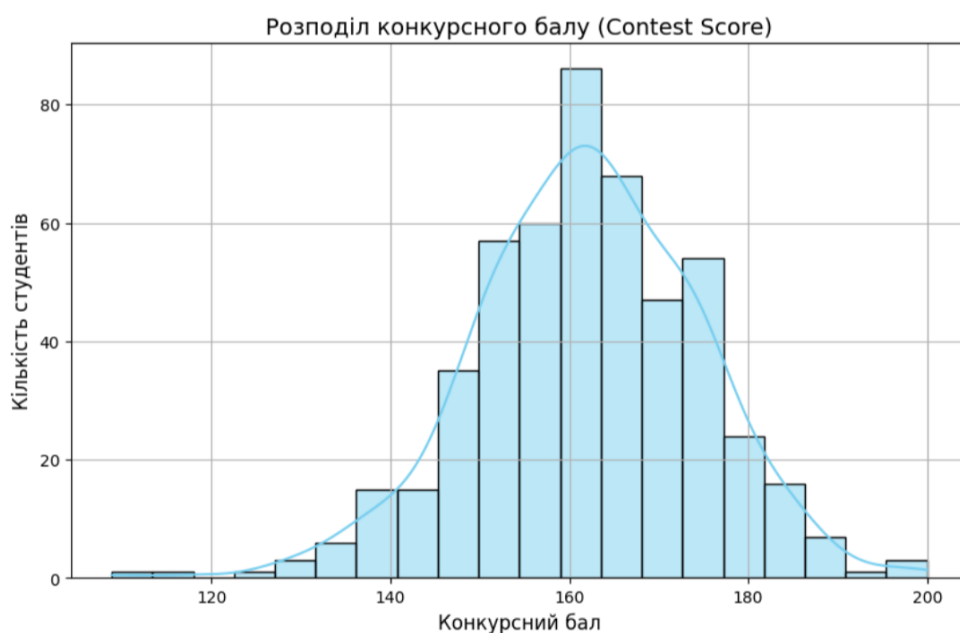


Рисунок 3.7 – Гістограма розподілу конкурсного балу абітурієнтів

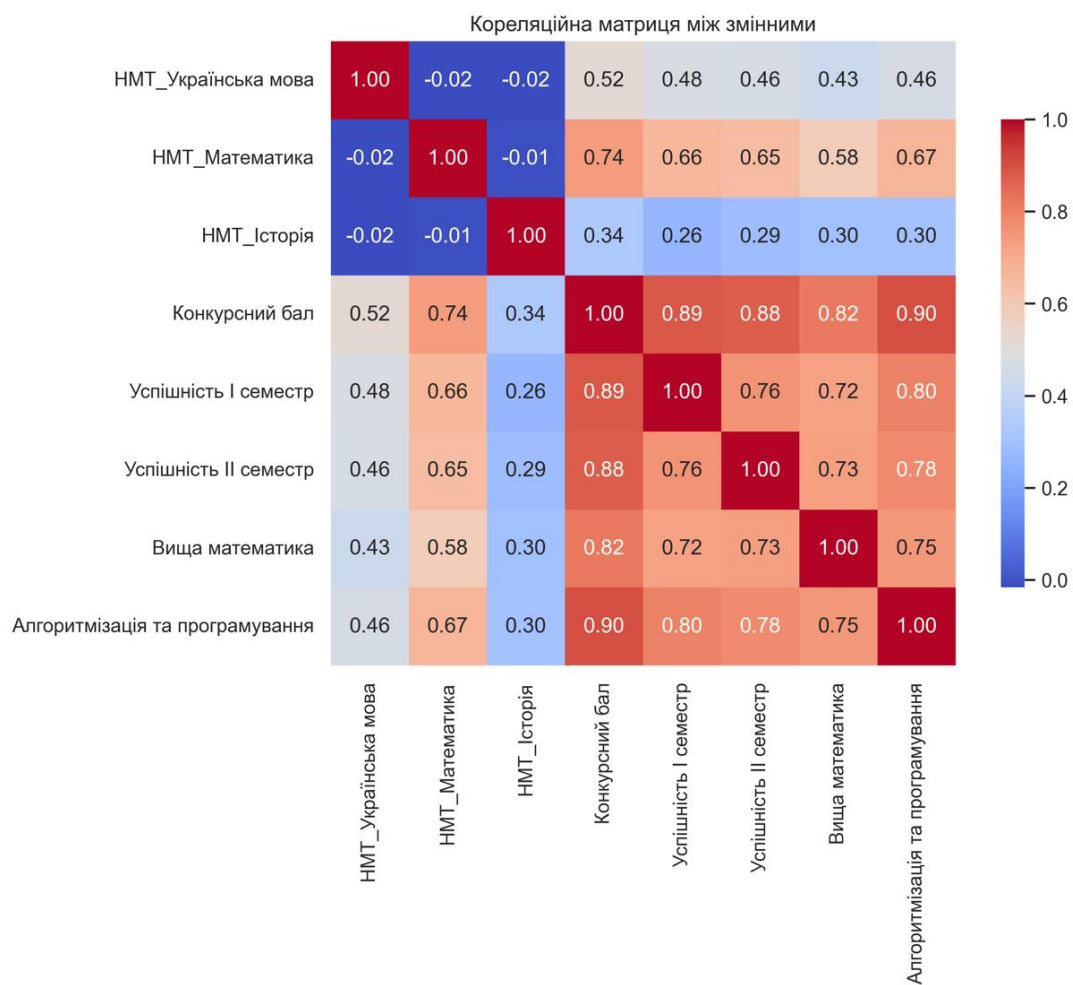


Рисунок 3.8 – Кореляційна матриця взаємозв'язку між вхідними даними та результатами навчання у ЗВО

Також було здійснено візуальний аналіз розподілу даних за допомогою діаграм розмаху (boxplot), що дозволило виявити викиди та аномальні значення в кожному з полів. Це особливо важливо на етапі підготовки датасету до моделювання, оскільки дозволяє зменшити вплив шумових даних і покращити якість прогнозної моделі.

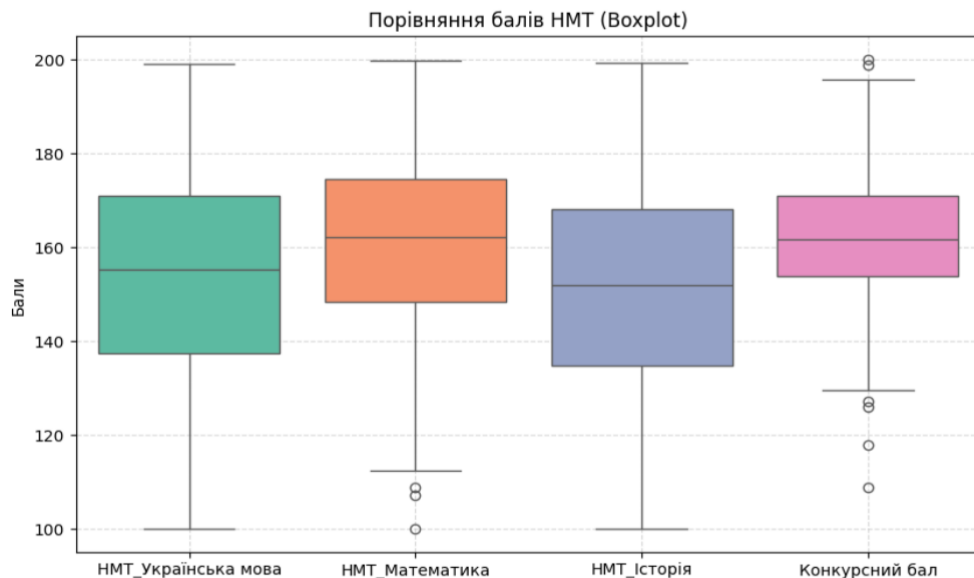


Рисунок 3.9 – Діаграми розмаху вхідних даних НМТ

3.4 Розробка функціоналу програмного модуля

На основі проведеного в другому розділі моделювання архітектури програмного модуля, зокрема розроблених UML-діаграм – діаграми класів, компонентів та послідовностей – було здійснено безпосередню програмну реалізацію ключових функціональних компонентів системи прогнозування успішності здобувачів вищої освіти. Ключовими компонентами реалізації є класи Student, PredictionModel та Database, кожен із яких відповідає за окремий аспект функціональності системи.

Центральним об'єктом у модулі є клас Student, який відповідає за представлення вхідних даних абітурієнта. До його полів належать унікальний ідентифікатор (ID), прізвище та ім'я, результати НМТ з української мови, математики та історії, середній бал атестату, стать, тип місцевості (місто/село) та

наявність пільги. Метод `get_features()` формує вектор ознак, який використовується як вхідні дані для прогнозовної моделі на основі заданих характеристик студента. Фрагмент програмного коду реалізації класу `Student` наведено на рис. 3.10.

```
# == Kлас Student ==
class Student:
    def __init__(self, ID, name, nmt_ukr, nmt_math, nmt_hist, gpa, kb, gender, rural, privilege):
        self.ID = ID
        self.name = name
        self.nmt_ukr = nmt_ukr
        self.nmt_math = nmt_math
        self.nmt_hist = nmt_hist
        self.gpa = gpa
        self.kb = kb
        self.gender = gender
        self.rural = rural
        self.privilege = privilege

    def get_features(self):
        return [
            self.nmt_ukr,
            self.nmt_math,
            self.nmt_hist,
            self.kb,
            self.gpa,
```

Рисунок 3.10 – Фрагмент програмного коду реалізації класу `Student`

Клас `PredictionModel` відповідає за обробку навчального процесу та прогнозування. Він містить два основні методи:

- `trainModel(data: DataFrame)` – виконує підготовку та навчання мультивихідної регресійної моделі (на базі `MultiOutputRegressor` з `LinearRegression`) на основі переданого датасету. Модель одночасно прогнозує чотири показники успішності: середній бал за I семестр, середній бал за II семестр, оцінку з вищої математики та оцінку з дисципліни "Алгоритмізація та програмування". Крім того, у методі реалізовано розділення на тренувальні й тестові дані, обчислення метрик точності для кожного з результатів, що дозволяє оцінити якість моделі.

- `predict(input: Student)` – приймає об'єкт `Student`, формує з нього вектор ознак і виконує прогнозування результатів навчання. У результаті повертається масив з чотирма значеннями прогнозованих оцінок. Фрагмент програмного коду реалізації даного класу наведено на рис. 3.11.

```
# == Клас PredictionModel ==
class PredictionModel:
    def __init__(self):
        self.model = MultiOutputRegressor(LinearRegression())

    def trainModel(self, dataset):
        features = ['nmt_ukrainian', 'nmt_math', 'nmt_history', 'kb', 'gpa', 'gender', 'rural', 'privilege']
        targets = ['avg_sem1', 'avg_sem2', 'math_score', 'prog_score']
        X = dataset[features]
        y = dataset[targets]

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

        self.model.fit(X_train, y_train)
        predictions = self.model.predict(X_test)

        print("Оцінка моделі R²:")
        for i, col in enumerate(targets):
            r2 = r2_score(y_test[col], predictions[:, i])
            print(f"{col}: R² = {r2:.2f}")

    def predict(self, student: Student):
        input_data = np.array(student.get_features()).reshape(1, -1)
        return self.model.predict(input_data)[0]
```

Рисунок 3.11 – Фрагмент програмного коду реалізації класу PredictionModel

Для роботи з джерелом даних реалізовано клас Database. Він виконує завантаження навчального датасету з CSV-файлу через метод retrieveData(), при цьому відбувається перетворення категоріальних змінних на числові (наприклад, стать або наявність пільг). За потреби цей клас можна доповнити методом storeStudentData() для збереження нових або прогнозованих даних до бази.

Функціонально програмний модуль реалізовано у вигляді окремого Python-скрипту з модульною структурою, де:

- виконується імпорт необхідних бібліотек (pandas, numpy, scikit-learn, matplotlib);
- реалізовано функції попередньої обробки даних (нормалізація, фільтрація, індексація);
- застосовано алгоритм лінійної регресії для навчання моделі;
- реалізовано інтерфейс взаємодії через командний рядок або інтерактивне середовище (Jupyter Notebook), що дозволяє вводити нові значення або працювати з тестовими даними.

Окрему увагу було приділено модулю оцінювання якості моделі, що включає розрахунок коефіцієнта детермінації (R^2), візуалізацію залишків та графіки порівняння фактичних і прогнозованих значень. Таким чином, реалізація

функціоналу забезпечує повний цикл обробки даних – від введення вхідних характеристик до генерації прогнозу та формування аналітичного звіту.

3.5 Тестування програмного модуля

Тестування програмного модуля є основним етапом, що дозволяє переконатися в правильності реалізації функціоналу, достовірності прогнозів та відповідності поведінки модуля очікуваним результатам. Процес тестування охоплює як функціональні перевірки роботи окремих класів, так і оцінку ефективності побудованої моделі прогнозування на тестових даних.

На першому етапі було проведено модульне тестування основних класів, таких як PredictionModel, Student та Database. Було перевірено правильність обробки вхідних даних; коректність збереження та вибірки даних у класі Database; стабільність функцій навчання та прогнозування в PredictionModel.

Для тестування використовувався стандартний інструментарій Python – модуль unittest. Приклад тестового класу для перевірки функціональності методу predict наведено на рис. 3.12

```
import unittest
from model import PredictionModel, Student
import pandas as pd

class TestPredictionModel(unittest.TestCase):
    def setUp(self):
        self.model = PredictionModel()
        df = pd.read_csv("training_data.csv")
        self.model.trainModel(df)

    def test_prediction_output(self):
        student = Student(ID=1, name="Тест", nmt_score=160, highschool_gpa=9.5,
                           region="micro", gender="жіноча", privilege=0)
        result = self.model.predict(student)
        self.assertIsInstance(result, dict)
        self.assertTrue("semester_1" in result and "math" in result)

if __name__ == '__main__':
    unittest.main()
```

Рисунок 3.12 – Фрагмент програмного коду перевірки функціональності методу predict

У ході тестування особливу увагу було приділено перевірці якості прогнозування результатів навчання. Після успішного модульного тестування базових компонентів системи, тестування було зосереджено на моделі прогнозування, зокрема її здатності узагальнювати навчальні дані та робити точні передбачення на нових прикладах. Для цього було сформовано навчальну і тестову вибірки з попередньо підготовленого датасету. На основі вхідних ознак, серед яких результати НМТ з трьох предметів, конкурсний бал, стать абітурієнта, регіон походження та середній бал атестату, було проведено навчання лінійної регресійної моделі.

Після навчання моделі здійснювалося прогнозування успішності за чотирма навчальними показниками: оцінками за I та II семестри, а також результатами з дисциплін "Вища математика" та "Алгоритмізація та програмування". Отримані результати були зіставлені з реальними значеннями на тестовій вибірці для обчислення метрик якості – середньої абсолютної похибки (MAE), середньоквадратичної похибки (MSE) та коефіцієнта детермінації (R^2) (рис. 3.13).

```
display(results_df.round(2))
```

	Цільова змінна	MSE	MAE	R^2
0	Успішність I семестр	12.02	2.15	0.77
1	Успішність II семестр	10.07	2.11	0.78
2	Вища математика	28.81	3.48	0.52
3	Алгоритмізація та програмування	2.73	1.29	0.93

Рисунок 3.13 – Результат розрахунку основних метрик побудованих моделей

Оцінювання моделі показало задовільну якість прогнозу, а для кращого представлення точності моделі були побудовані графіки розсіювання з регресійними лініями, які демонструють залежності між конкурсним балом і результатами навчання (рис. 3.14). Окремо був також реалізований прогноз на основі показників НМТ для окремих студентів з розрахунком довірчих інтервалів. Це дозволило оцінити не лише очікуване значення прогнозу, а й можливі межі

коливань оцінки, враховуючи похибки моделі. Для зручності результати візуалізувалися у вигляді горизонтальних гістограм із зазначенням довірчих інтервалів (рис. 3.15).

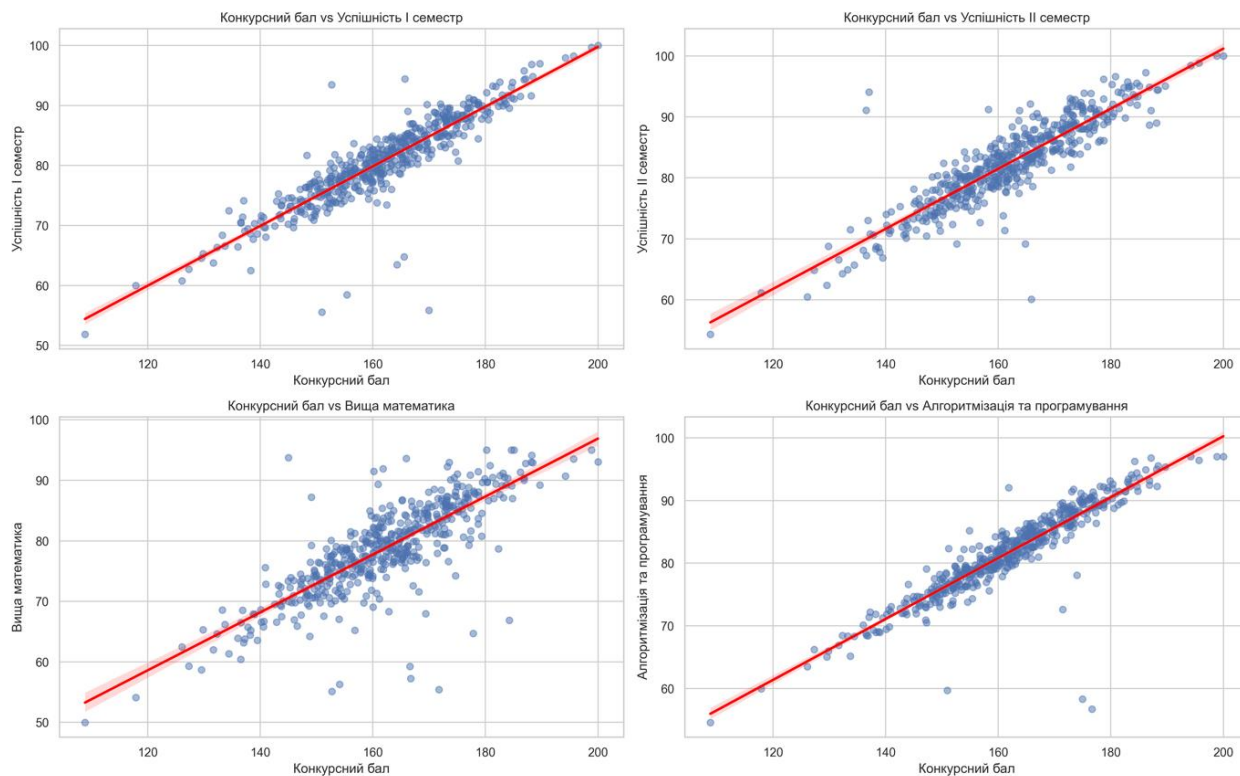


Рисунок 3.14 – Графіки розсіювання значень прогнозних значень побудованих моделей

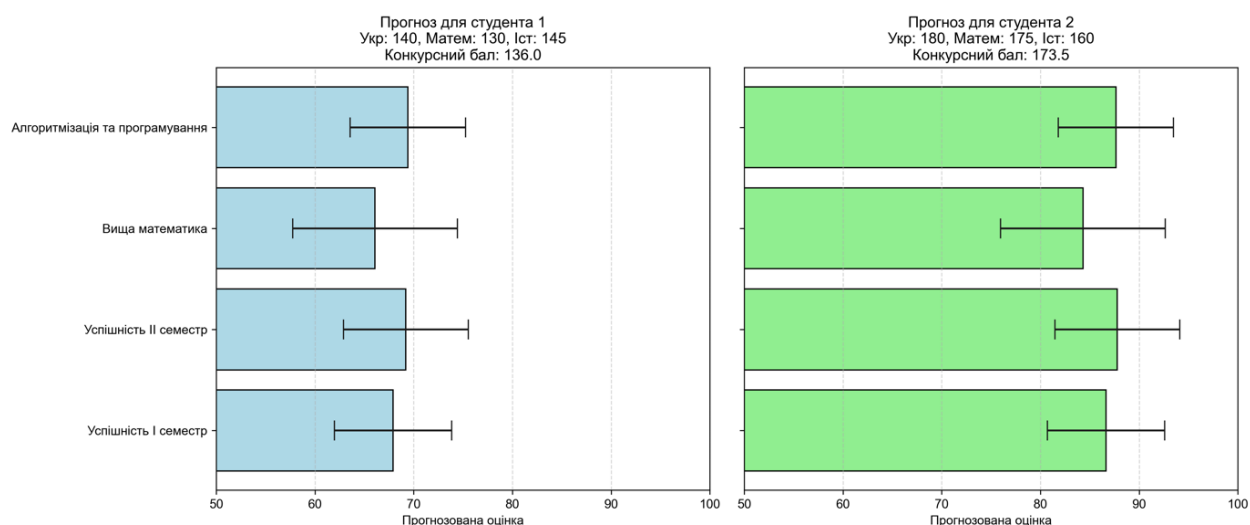


Рисунок 3.15 – Результат точкового та інтервального прогнозування успішності окремих студентів на основі показників НМТ

Таким чином, проведення тестування модуля підтвердило функціональну коректність роботи системи та статистичну обґрунтованість її прогнозів, що в цілому забезпечує надійність розробленого інструменту прогнозування академічної успішності студентів.

Також можна відзначити, що більшість наявних систем або інтегрованих рішень, які застосовуються для оцінки та моніторингу академічної успішності, зосереджені переважно на описовій аналітиці – вони надають статистичні звіти, індикатори відвідуваності та середні бали без глибокого прогнозування майбутніх результатів або персоналізації висновків. У таблиці 3.4 наведено порівняльний аналіз функціональних можливостей розробленого програмного модуля відносно популярних освітніх аналітичних платформ Civitas Learning, Blackboard Analytics та Panorama Education.

Таблиця 3.4 – Порівняння функціоналу розробленого програмного модуля з конкурентними системами

Функціональна можливість	Розроблений модуль	Civitas Learning	Blackboard Analytics	Panorama Education
1	2	3	4	5
Прогнозування академічної успішності	Реалізовано	Реалізовано	Обмежене, здебільшого описове	Фокусується на емоційному кліматі
Аналіз НМТ та конкурсного балу	Враховуються зовнішні оцінки	Не підтримується	Не підтримується	Не підтримується
Індивідуалізовані прогнози для студентів	Реалізовано	Реалізовано	Обмежене в інтерфейсі аналітики	Обмежене
Інтеграція з базами даних ЗВО / ЄДЕБО	Можлива при доопрацюванні	Не підтримується	Не підтримується	Не підтримується
Візуалізація результатів прогнозу	Реалізовано	Реалізовано	Обмежена деталізація	Обмежена деталізація
Оцінка впливу додаткових факторів	Реалізовано	Обмежено	Не підтримується	Реалізовано
Відкритість архітектури / розширюваність	Можна додавати нові моделі та змінні	Закриті системи	Закриті системи	Закриті системи

Продовження табл. 3.4

1	2	3	4	5
Аналіз соціально-емоційного клімату	Можливий при доопрацюванні	Обмежено	Не підтримується	Реалізовано
Автоматичне оновлення на основі нових даних	Можливе при інтеграції з БД	Реалізовано	Реалізовано	Реалізовано

Аналізуючи дані табл. 3.4 можна відзначити, що розроблений програмний модуль відрізняється своєю прогностичною спрямованістю, яка реалізується через інтеграцію алгоритмів машинного навчання, зокрема лінійної регресії, яка дозволяє на основі вхідних даних (результатів НМТ, конкурсного балу, середнього балу атестата та інших показників) формувати персоналізований прогноз академічної успішності. Це забезпечує більш гнучке та адаптивне рішення, яке може бути корисним не лише для адміністрації навчального закладу, а й для самих здобувачів освіти.

Підсумовуючи можна відзначити, що врахування освітньої специфіки України шляхом інтеграції результатів зовнішнього незалежного оцінювання (НМТ), впровадження елементів розширеної статистичної візуалізації та можливостей індивідуалізованого прогнозування академічної успішності забезпечили перевагу функціональних можливостей розробленого програмного модуля над існуючими аналітичними платформами. Зокрема, модуль реалізує на 20% більше функціональних критеріїв, ніж найближчий конкурент – Civitas Learning, який охоплює лише 7 відповідних параметрів (табл. 3.4).

3.6 Висновок до розділу 3

У третьому розділі розглянуто програмну реалізацію модуля прогнозування академічної успішності здобувачів вищої освіти, що включала вибір мови програмування та інструментів розробки, створення функціональних компонентів та проведення тестування. Вибір мови програмування Python забезпечив

ефективну реалізацію завдяки наявності потужних бібліотек для роботи з даними, машинного навчання та візуалізації. Розроблений модуль враховує освітню специфіку України через використання результатів національного мультипредметного тесту (НМТ) як одного з ключових факторів прогнозування. Проведене тестування підтвердило коректність функціонування програмного забезпечення та його здатність точно прогнозувати академічні результати студентів. У порівнянні з конкурентними аналітичними платформами, розроблений модуль має на 20% ширший функціонал, адаптований до національних освітніх умов, і може бути розгорнутий локально без додаткових витрат, що робить його доступним і практичним рішенням для закладів вищої освіти.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи досягнуто поставленої мети – розширення функціональних можливостей процесу прогнозування академічної успішності здобувачів вищої освіти шляхом урахування результатів національного мультипредметного тесту (НМТ), а саме:

1. Проведено аналіз предметної області та виявлено ключові фактори, що впливають на успішність студентів, серед яких: результати вступних випробувань (НМТ), середній бал атестата, а також інші соціально-економічні умови. Встановлено, що більшість існуючих моделей акцентуються переважно на академічних показниках, отриманих під час навчання, і при цьому не враховують дані вступної кампанії як важливі прогностичні індикатори.

2. Здійснено порівняльний аналіз сучасних програмних рішень у сфері освітньої аналітики. Виявлено їхні основні переваги та недоліки. На цій основі обґрунтовано необхідність розширення функціональних можливостей модуля, який буде адаптований до національного контексту та міститиме ширший спектр функцій прогнозування.

3. Реалізовано проєктування програмного модуля, що включало створення алгоритму прогнозування, розробку архітектури системи, побудову діаграми послідовностей та діаграми класів. Запропонована архітектура забезпечує логічну цілісність, модульність та гнучкість для майбутнього розширення функціоналу.

4. Здійснено програмну реалізацію модуля з використанням мови Python та сучасних бібліотек для аналізу даних, машинного навчання та візуалізації. Розроблений модуль забезпечує обробку вхідних даних, формування прогнозу, відображення результатів у зручному для користувача форматі та враховує освітню специфіку України.

5. Проведено тестування програмного модуля, що підтвердило коректність його функціонування та здатність прогнозувати академічну успішність на основі результатів НМТ. Порівняння з аналогічними рішеннями показало, що запропонований модуль має на 20% ширший функціонал у порівнянні з

найближчим конкурентом за рахунок врахування освітньої специфіки України шляхом інтеграції результатів зовнішнього незалежного оцінювання (НМТ), впровадження елементів розширеної статистичної візуалізації та можливостей індивідуалізованого прогнозування академічної успішності.

Таким чином, всі поставлені перед бакалаврською кваліфікаційною роботою задачі реалізовано в повному обсязі та мету досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дужич Н. В., Мялюк О. П., Марущак М. І. Аналіз чинників, які впливають на академічну успішність. *Медична освіта*. 2023. № 3. С. 54–61. DOI: <https://doi.org/10.11603/m.2414-5998.2023.3.14050>.
2. Козьменко О. І. Аналіз впливу різноманітних чинників на успіх студентів в університеті. *Науковий часопис НПУ імені М. П. Драгоманова. Серія 5. Педагогічні науки: реалії та перспективи*. 2018. Вип. 63(59). С. 82–85. URL: <https://www.chasopys.ps.npu.kiev.ua/archive/63-2018/21.pdf>.
3. Marks G. N., Pokropek A. Family income effects on mathematics achievement: their relative magnitude and causal pathways. *Oxford Review of Education*. 2019. Vol. 45, No. 6. P. 769–785. DOI: <https://doi.org/10.1080/03054985.2019.1620717>.
4. Dahl G. B., Lochner L. The impact of family income on child achievement [Електронний ресурс]. *National Bureau of Economic Research*. 2005. (NBER Working Paper No. 11279). Режим доступу: <http://www.nber.org/papers/w11279>.
5. Shogren K. A. et al. Exploring the Relationship Between Self-Determination and Academic Outcomes. *Remedial and Special Education*. 2025. 0(0). DOI: <https://doi.org/10.1177/07419325251322908>.
6. Alsaeed A. H. et al. Student perspectives on using the self-determined learning model of instruction to enhance outcomes. *Inclusion*. 2024. Vol. 12, No. 4. P. 259–274. DOI: <https://doi.org/10.1352/2326-6988-12.4.259>.
7. Шевчук О. Ф. Прогностична валідність двомодельної системи конкурсного відбору студентів економічного напрямку. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців*. 2021. Вип. 62. С. 307–318. DOI: <https://doi.org/10.31652/2412-1142-2021-62-307-318>.
8. Шевчук О. Ф. Прогностична валідність конкурсного бала студентів-першокурсників економічного напрямку. *Економіка, фінанси, менеджмент: актуальні питання науки і практики*. 2018. № 7(35). С. 65–78.

9. Barbayannis G. et al. Academic Stress and Mental Well-Being in College Students: Correlations, Affected Groups, and COVID-19. *Frontiers in Psychology*. 2022. Vol. 13. Article 886344. DOI: <https://doi.org/10.3389/fpsyg.2022.886344>.

10. Fu W., Hashim A. T. M. The Impact of Urban and Rural Education Gap on Student Achievement Differences. *Journal of Contemporary Educational Research*. 2024. Vol. 8, No. 9. P. 145–150. DOI: <https://doi.org/10.26689/jcer.v8i9.8388>.

11. Шевчук О. Ф. Вивчення впливу сільського коефіцієнта на прогностичну валідність конкурсного бала студентів-першокурсників. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців*. 2018. Вип. 52. С. 439–443.

12. *Civitas Learning* [Електронний ресурс]. URL: <https://www.civitaslearning.com> (дата звернення: 05.05.2025).

13. *Analytics for Learn | Blackboard Help* [Електронний ресурс]. URL: https://help.blackboard.com/Learn/Instructor/Ultra/Performance/Analytics_for_Learn (дата звернення: 05.05.2025).

14. *Student Behavior Management Software | Panorama Education* [Електронний ресурс]. URL: <https://www.panoramaed.com/products/behavior> (дата звернення: 05.05.2025).

15. *Panorama Education*. The “Big 5” of Behavior Data for PBIS Teams: Cheatsheet [Електронний ресурс]. URL: <https://go.panoramaed.com/hubfs/Marketing%20Content/PBIS%20Big%205.pdf> (дата звернення: 12.05.2025).

16. Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання. Вінниця : ВНТУ, 2024. 258 с.

17. Савчук Т. О., Ольшанська О. В. Технології комп’ютерного проектування : електронний навчальний посібник комбінованого (локального та мережевого) використання. Вінниця : ВНТУ, 2024. 125 с.

18. *Top-10 мов програмування, які будуть актуальні в 2025 // IT Jobs at SoftServe – Start tech career now* [Електронний ресурс]. URL:

<https://career.softserveinc.com/uk-ua/stories/top-10-programming-languages-to-learn>
(дата звернення: 21.05.2025).

19. *Python.org*. Welcome to Python.org [Електронний ресурс]. URL: <https://www.python.org> (дата звернення: 21.05.2025).

20. *Популярні середовища розробки мов програмування – UA5.org* [Електронний ресурс]. URL: <https://ua5.org/osnprog/1600-populyarni-seredovyshha-rozrobky-mov-programuvannya.html> (дата звернення: 21.05.2025).

21. *Середовище розробки Python: що для себе обере професіонал – FoxmindEd* [Електронний ресурс]. URL: <https://foxminded.ua/seredovyshche-rozrobky-python/> (дата звернення: 21.05.2025).

22. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» / уклад.: Яровий А. А., Сілагін О. В., Богач І. В. Вінниця : ВНТУ, 2023. [Електронний ресурс].

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Програмний модуль прогнозування успішності студентів на основі результатів національного мультипредметного тесту

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра комп'ютерних наук
(кафедра, факультет, навчальна група)

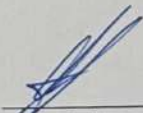
Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 0,81 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

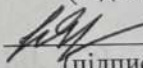
- ✓ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Яровий А.А., зав. каф. КН
(прізвище, ініціали, посада)


(підпис)

Колесницький О.К., проф. каф КН
(прізвище, ініціали, посада)


(підпис)

Особа, відповідальна за перевірку 
(підпис)

Озеранський В.С.
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Шевчук О.Ф.
(прізвище, ініціали, посада)

Здобувач 
(підпис)

Наумов О.О.
(прізвище, ініціали)

Додаток Б (обов'язковий)**Лістинг програми**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import unittest
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.multioutput import MultiOutputRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression
from model import PredictionModel, Student

import openpyxl
import sqlite3
from sqlalchemy import create_engine
from flask import Flask, request, render_template
import os
import json

np.random.seed(42)
n = 500

nmt_ukrainian = np.random.normal(155, 25, n)
nmt_ukrainian = np.clip(nmt_ukrainian, 100, 200)
nmt_ukrainian[nmt_ukrainian == 200] = np.random.randint(180, 200,
size=(nmt_ukrainian == 200).sum())

nmt_math = np.random.normal(162, 20, n)
nmt_math = np.clip(nmt_math, 100, 200)
nmt_math[nmt_math == 200] = np.random.randint(180, 200, size=(nmt_math ==
200).sum())

nmt_history = np.random.normal(150, 25, n)
nmt_history = np.clip(nmt_history, 100, 200)
```

```

nmt_history[nmt_history == 200] = np.random.randint(150, 200, size=(nmt_history ==
200).sum())
nmt_history[nmt_history == 100] = np.random.randint(100, 140, size=(nmt_history ==
100).sum())

```

```

gender = np.random.choice(['чоловіча', 'жіноча'], n)
avg_certificate = np.random.uniform(6.0, 12.0, n)
locality = np.random.choice([0, 1], n) # 0 — міська, 1 — сільська
benefit = np.random.choice([0, 1], n, p=[0.85, 0.15])

```

```

K1, K2, K3 = 0.3, 0.5, 0.2
OU = np.random.uniform(0, 10, n)

```

```

KB = (K1 * nmt_ukrainian + K2 * nmt_math + K3 * nmt_history) / (K1 + K2 + K3) +
OU
KB = np.clip(KB, 100, 200)

```

```

entrants_df = pd.DataFrame({
    'id': np.arange(1, n + 1),
    'nmt_ukrainian': np.round(nmt_ukrainian, 1),
    'nmt_math': np.round(nmt_math, 1),
    'nmt_history': np.round(nmt_history, 1),
    'gender': gender,
    'avg_certificate': np.round(avg_certificate, 2),
    'locality': locality,
    'benefit': benefit,
    'contest_score': np.round(KB, 2)
})

```

```

def generate_achievement_scores(n):
    scores = []

    high = int(n * np.random.uniform(0.05, 0.08))
    high_scores = np.random.uniform(90, 98, high)
    scores.extend(high_scores)

    mid_high = int(n * np.random.uniform(0.15, 0.18))
    mid_high_scores = np.random.uniform(82, 89, mid_high)
    scores.extend(mid_high_scores)

```

```

mid = int(n * np.random.uniform(0.25, 0.28))
mid_scores = np.random.uniform(75, 81, mid)
scores.extend(mid_scores)

remaining = n - len(scores)
low_scores = np.random.uniform(50, 74, remaining)
scores.extend(low_scores)

np.random.shuffle(scores)
return np.round(scores[:n], 1)

academic_df = pd.DataFrame({
    'id': np.arange(1, n + 1),
    'avg_sem1': generate_achievement_scores(n),
    'avg_sem2': generate_achievement_scores(n),
    'higher_math': generate_achievement_scores(n),
    'programming': generate_achievement_scores(n)
})

full_dataset = pd.merge(entrants_df, academic_df, on='id')
full_dataset.head(500)

correlation_matrix = df.corr(numeric_only=True)

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
            square=True, cbar_kws={"shrink": .75})
plt.title('Кореляційна матриця між змінними')
plt.tight_layout()
plt.savefig("correlation_matrix.png", dpi=300, bbox_inches='tight')
plt.show()

fig, axs = plt.subplots(1, 3, figsize=(18, 5))

axs[0].hist(nmt_ukrainian, bins=20, color='skyblue', edgecolor='black')
axs[0].set_title('НМТ: Українська мова')
axs[0].set_xlabel('Бали')
axs[0].set_ylabel('Кількість абітурієнтів')

```

```

axs[1].hist(nmt_math, bins=20, color='lightgreen', edgecolor='black')
axs[1].set_title('НМТ: Математика')
axs[1].set_xlabel('Бали')
axs[1].set_ylabel('Кількість абітурієнтів')

```

```

axs[2].hist(nmt_history, bins=20, color='salmon', edgecolor='black')
axs[2].set_title('НМТ: Історія України')
axs[2].set_xlabel('Бали')
axs[2].set_ylabel('Кількість абітурієнтів')

```

```

plt.tight_layout()
plt.show()

```

```

plt.figure(figsize=(10, 6))
sns.histplot(full_dataset['contest_score'], kde=True, bins=20, color='skyblue')
plt.title('Розподіл конкурсного балу (Contest Score)', fontsize=14)
plt.xlabel('Конкурсний бал', fontsize=12)
plt.ylabel('Кількість студентів', fontsize=12)
plt.grid(True)
plt.show()

```

```

print("Описова статистика:")
print(df.describe().round(2))

```

```

plt.figure(figsize=(10, 6))
sns.boxplot(data=df, palette='Set2')
plt.title('Порівняння балів НМТ (Boxplot)')
plt.ylabel('Бали')
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()

```

```

plt.figure(figsize=(12, 6))
for column, color in zip(df.columns, ['skyblue', 'lightgreen', 'salmon']):
    sns.histplot(df[column], kde=True, bins=30, color=color, label=column,
stat="density", alpha=0.5)

```

```

plt.title('Розподіл балів НМТ з щільністю')
plt.xlabel('Бали')

```

```

plt.ylabel('ЩІЛЬНІСТЬ')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()

# === Клас Student ===
class Student:
    def __init__(self, ID, name, nmt_ukr, nmt_math, nmt_hist, gpa, kb, gender, rural,
privilege):
        self.ID = ID
        self.name = name
        self.nmt_ukr = nmt_ukr
        self.nmt_math = nmt_math
        self.nmt_hist = nmt_hist
        self.gpa = gpa
        self.kb = kb
        self.gender = gender
        self.rural = rural
        self.privilege = privilege

    def get_features(self):
        return [
            self.nmt_ukr,
            self.nmt_math,
            self.nmt_hist,
            self.kb,
            self.gpa,
            int(self.gender == 'жіноча'),
            int(self.rural == 'так'),
            int(self.privilege == 'так')
        ]

# === Клас PredictionModel ===
class PredictionModel:
    def __init__(self):
        self.model = MultiOutputRegressor(LinearRegression())

    def trainModel(self, dataset):

```

```

features = ['nmt_ukrainian', 'nmt_math', 'nmt_history', 'kb', 'gpa', 'gender', 'rural',
'privilege']
targets = ['avg_sem1', 'avg_sem2', 'math_score', 'prog_score']
X = dataset[features]
y = dataset[targets]

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

self.model.fit(X_train, y_train)
predictions = self.model.predict(X_test)

print("Оцінка моделі R²:")
for i, col in enumerate(targets):
    r2 = r2_score(y_test[col], predictions[:, i])
    print(f"{col}: R² = {r2:.2f}")

def predict(self, student: Student):
    input_data = np.array(student.get_features()).reshape(1, -1)
    return self.model.predict(input_data)[0]

# === Клас Database ===
class Database:
    def __init__(self, filepath):
        self.filepath = filepath

    def retrieveData(self):
        df = pd.read_csv(self.filepath)
        df['gender'] = df['gender'].map({'чоловіча': 0, 'жіноча': 1})
        df['rural'] = df['rural'].map({'ні': 0, 'так': 1})
        df['privilege'] = df['privilege'].map({'ні': 0, 'так': 1})
        return df

# === Використання ===
if __name__ == "__main__":
    db = Database("synthetic_student_data.csv")
    df = db.retrieveData()

    model = PredictionModel()

```

```

model.trainModel(df)

new_student = Student(
    ID=501,
    name="Іван Петренко",
    nmt_ukr=182,
    nmt_math=176,
    nmt_hist=170,
    gpa=11.2,
    kb=185.4,
    gender='чоловіча',
    rural='ні',
    privilege='ні'
)

predictions = model.predict(new_student)
labels = ['Семестр 1', 'Семестр 2', 'Вища математика', 'Алгоритмізація і
програмування']
print("\nПрогнозовані результати успішності:")
for label, score in zip(labels, predictions):
    print(f"{label}: {score:.1f}")

X_train, X_test, y_train, y_test = train_test_split(features, df[targets], test_size=0.2,
random_state=42)

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

y_pred = model.predict(X_test)

metrics = {
    'Цільова змінна': [],
    'MSE': [],
    'MAE': [],
    'R²': []
}

for i, col in enumerate(targets):
    metrics['Цільова змінна'].append(col)

```



```

metrics['MSE'].append(mean_squared_error(y_test.iloc[:, i], y_pred[:, i]))
metrics['MAE'].append(mean_absolute_error(y_test.iloc[:, i], y_pred[:, i]))
metrics['R²'].append(r2_score(y_test.iloc[:, i], y_pred[:, i]))

results_df = pd.DataFrame(metrics)
print("\n Метрики якості лінійної регресії:\n")
print(results_df.round(2).to_string(index=False))

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

for i, col in enumerate(targets):
    mse = mean_squared_error(y_test.iloc[:, i], y_pred[:, i])
    mae = mean_absolute_error(y_test.iloc[:, i], y_pred[:, i])
    r2 = r2_score(y_test.iloc[:, i], y_pred[:, i])
    print(f"{col}: MSE = {mse:.2f}, MAE = {mae:.2f}, R² = {r2:.2f}")

def calculate_total_score(ukr, math, hist):
    return 0.3 * ukr + 0.5 * math + 0.2 * hist

student1 = {
    'НМТ_Українська мова': 140,
    'НМТ_Математика': 130,
    'НМТ_Історія': 145
}
student1['Конкурсний бал'] = calculate_total_score(
    student1['НМТ_Українська мова'],
    student1['НМТ_Математика'],
    student1['НМТ_Історія']
)

student2 = {
    'НМТ_Українська мова': 180,
    'НМТ_Математика': 175,
    'НМТ_Історія': 160
}
student2['Конкурсний бал'] = calculate_total_score(

```

```

    student2['НМТ_Українська мова'],
    student2['НМТ_Математика'],
    student2['НМТ_Історія']
)

features = df[[
    'НМТ_Українська мова',
    'НМТ_Математика',
    'НМТ_Історія',
    'Конкурсний бал'
]]
targets = [
    'Успішність I семестр',
    'Успішність II семестр',
    'Вища математика',
    'Алгоритмізація та програмування'
]

X_train, X_test, y_train, y_test = train_test_split(features, df[targets], test_size=0.2,
random_state=42)

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

new_data = pd.DataFrame([student1, student2])
predictions = model.predict(new_data)

y_train_pred = model.predict(X_train)
residuals = y_train - y_train_pred
mse = np.mean(residuals**2, axis=0)
n = len(X_train)
conf_interval = 1.96 * np.sqrt(mse * (1 + 1/n))

fig, axs = plt.subplots(1, 2, figsize=(14, 6))

for i, (student, pred) in enumerate(zip([student1, student2], predictions)):
    axs[i].barh(targets, pred, xerr=conf_interval, color='lightgreen' if i == 1 else
'lightblue',
                edgecolor='black', capsize=8)

```

```
axs[i].set_xlim(50, 100)
axs[i].set_title(f'Прогноз для студента {i+1}\n'
                 f'Укр: {student['НМТ_Українська мова']}, Матем: '
                 f'{student['НМТ_Математика']}, Іст: {student['НМТ_Історія']}\n'
                 f'Конкурсний бал: {student['Конкурсний бал']:.1f}')
axs[i].set_xlabel("Прогнозована оцінка")
axs[i].grid(axis='x', linestyle='--', alpha=0.5)

plt.tight_layout()
plt.savefig("student_comparison_predictions.png", dpi=300)
plt.show()
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

Програмний модуль прогнозування успішності студентів на основі
результатів національного мультипредметного тесту

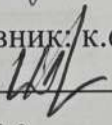
Виконав: студент 4 курсу, групи 4КН-216
спеціальності 122 – Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

 Наумов О. О.

(прізвище та ініціали)

Керівник: к.ф.-м.н., доцент каф. КН

 Шевчук О. Ф.

(прізвище та ініціали)

« 03 » червня 2025 р.

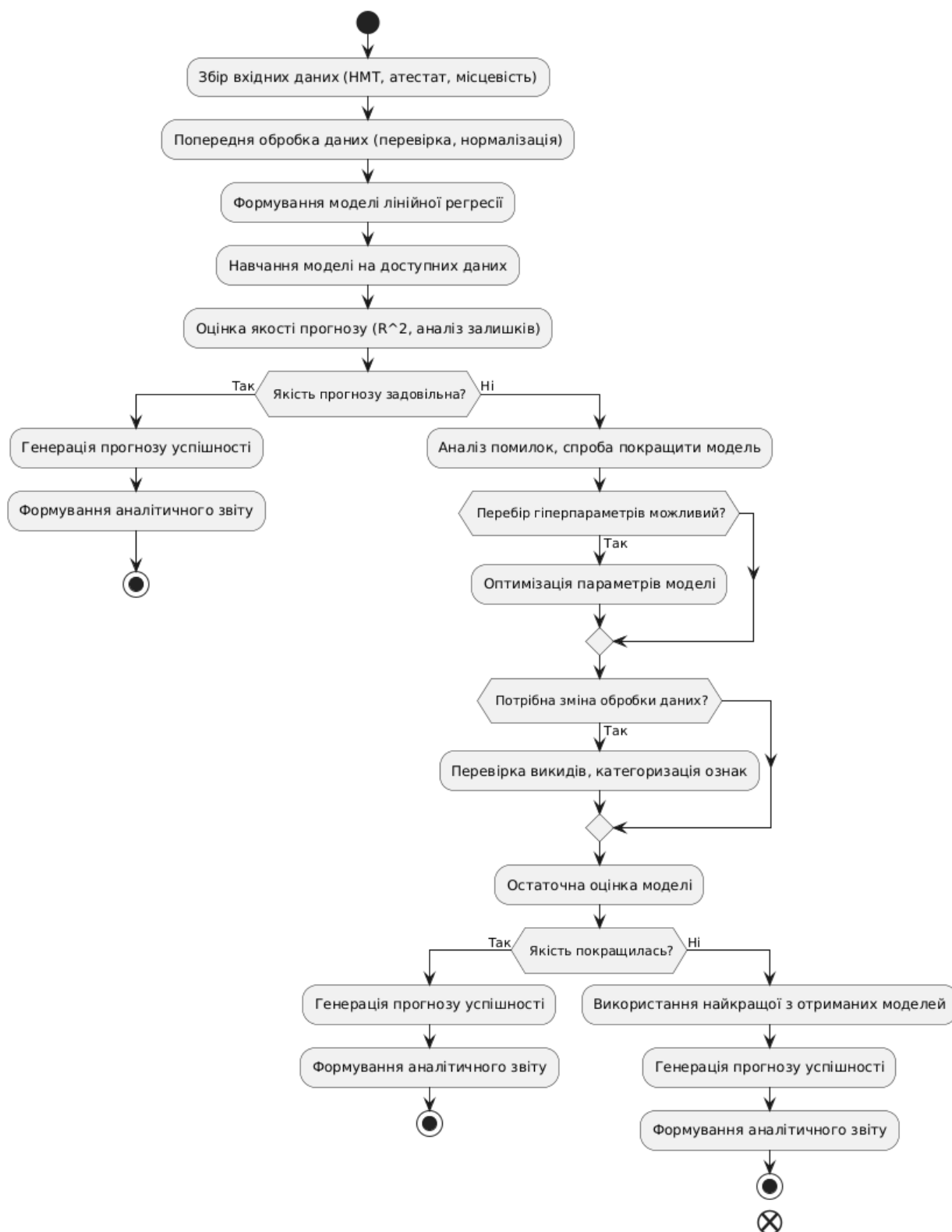


Рисунок В.1 – UML-діаграма удосконаленого алгоритму прогнозування успішності здобувачів освіти

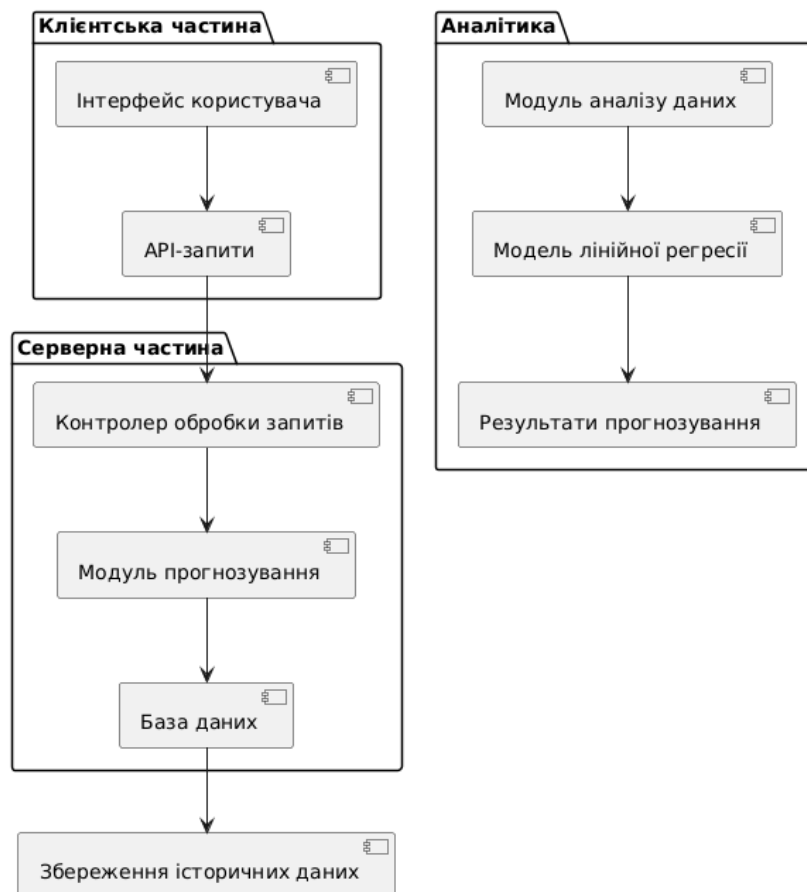


Рисунок В.2 – Діаграма компонентів програмного модуля прогнозування успішності здобувачів вищої освіти

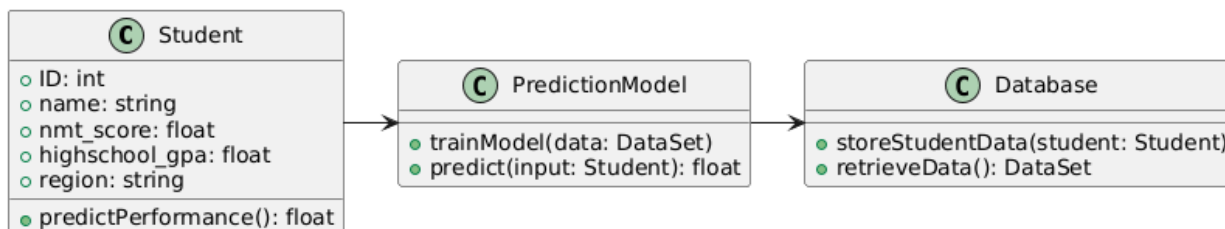


Рисунок В.3 – Діаграма класів програмного модуля прогнозування успішності здобувачів вищої освіти

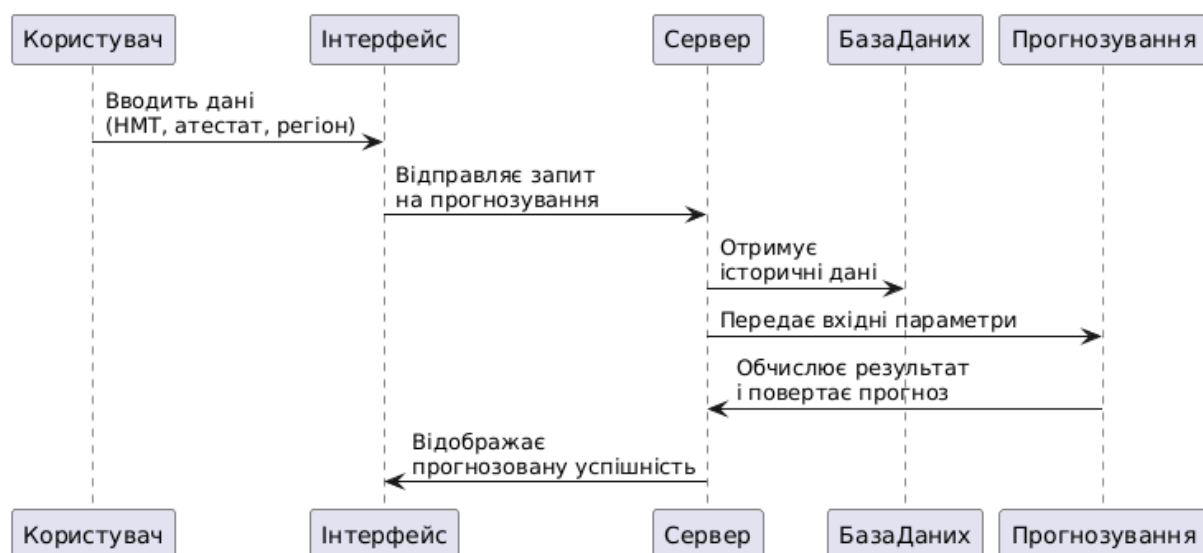


Рисунок В.4 – Діаграма послідовностей прогнозування успішності здобувачів вищої освіти

	HMT_Українська мова	HMT_Математика	HMT_Історія	Конкурсний бал
count	500.0	500.0	500.0	500.0
mean	154.0	161.0	151.0	162.0
std	22.0	19.0	23.0	13.0
min	100.0	100.0	100.0	109.0
25%	137.0	148.0	135.0	154.0
50%	155.0	162.0	152.0	162.0
75%	171.0	175.0	168.0	171.0
max	199.0	200.0	199.0	200.0

Рисунок В.5 – Описова статистика вхідних даних навчального датасету

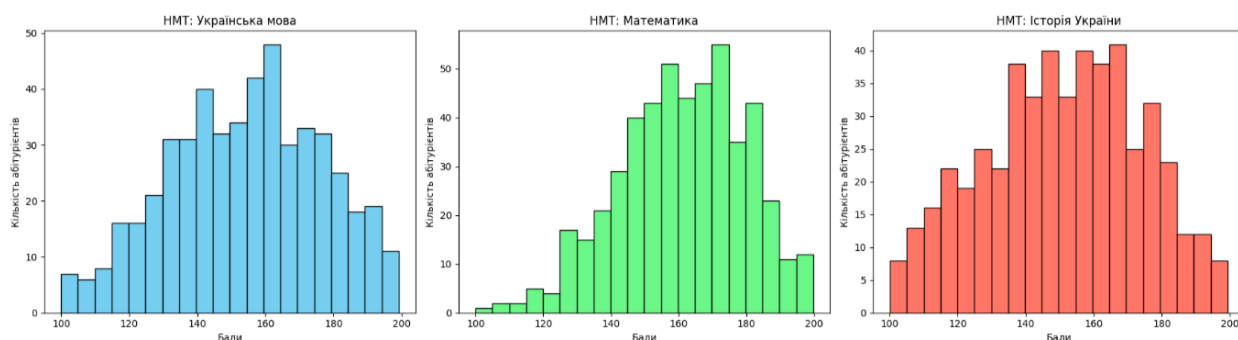


Рисунок В.6 – Гістограми розподілу результатів НМТ з української мови, математики та історії України

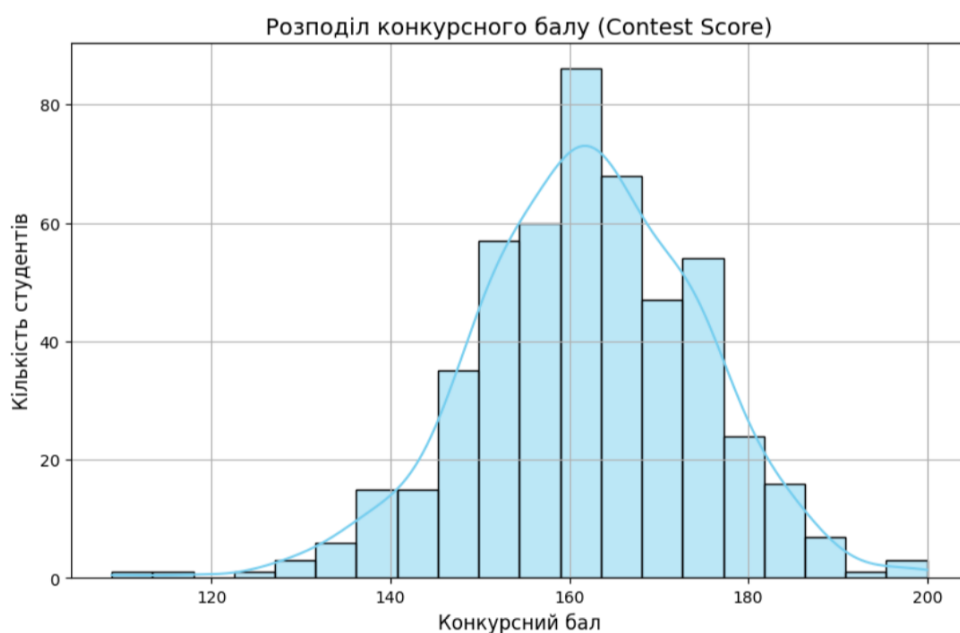


Рисунок В.7 – Гістограма розподілу конкурсного балу абітурієнтів

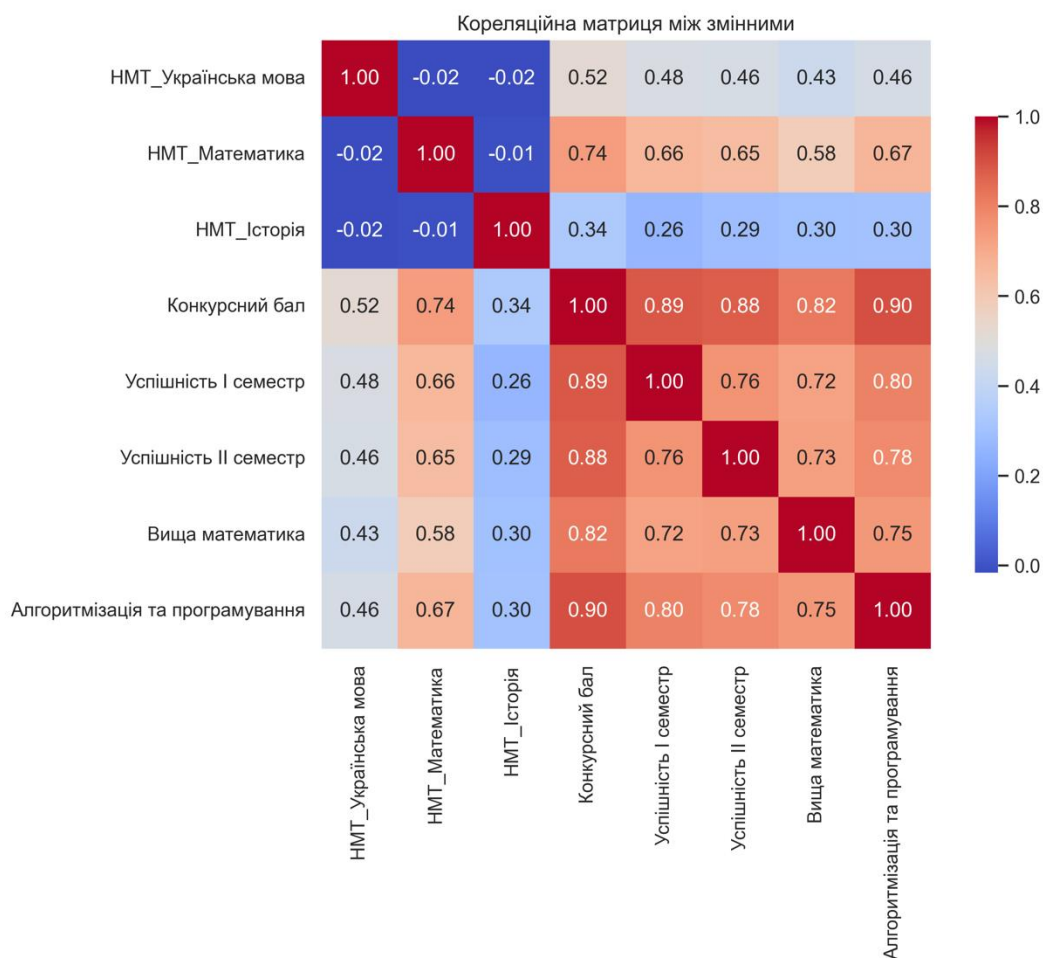


Рисунок В.8 – Кореляційна матриця взаємозв'язку між вхідними даними та результатами навчання у ЗВО

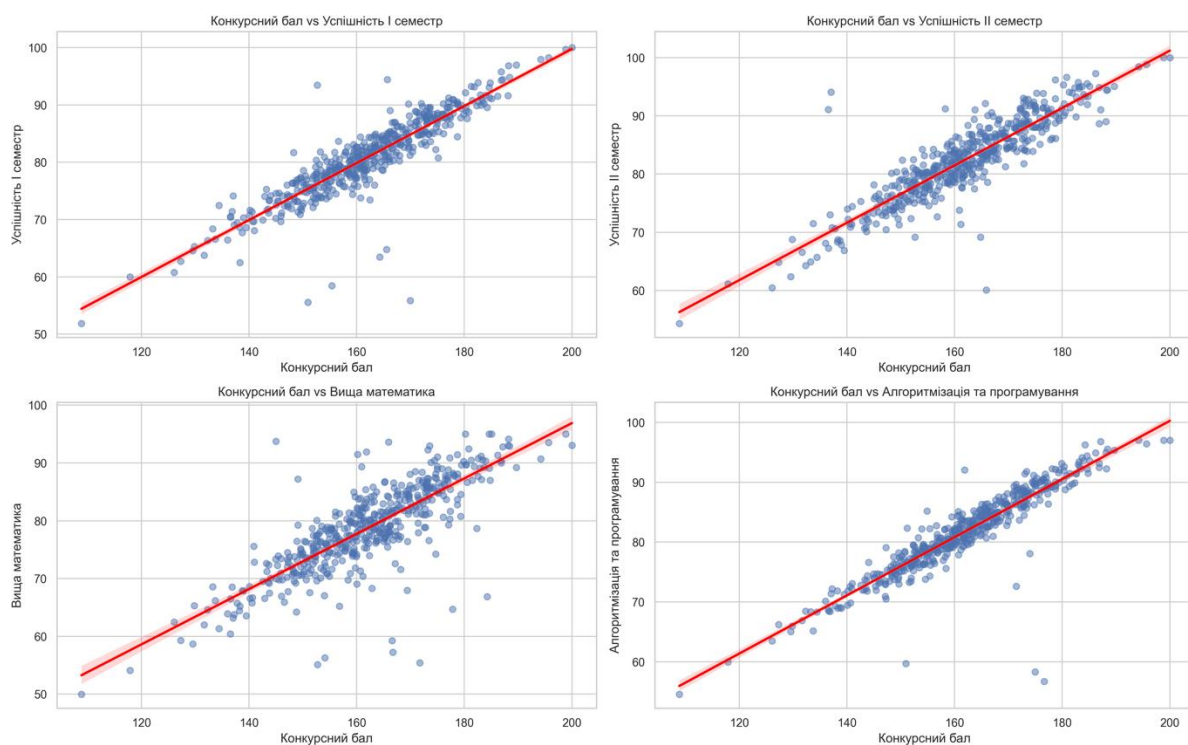


Рисунок В.9 – Графіки розсіювання значень прогностичних значень побудованих моделей

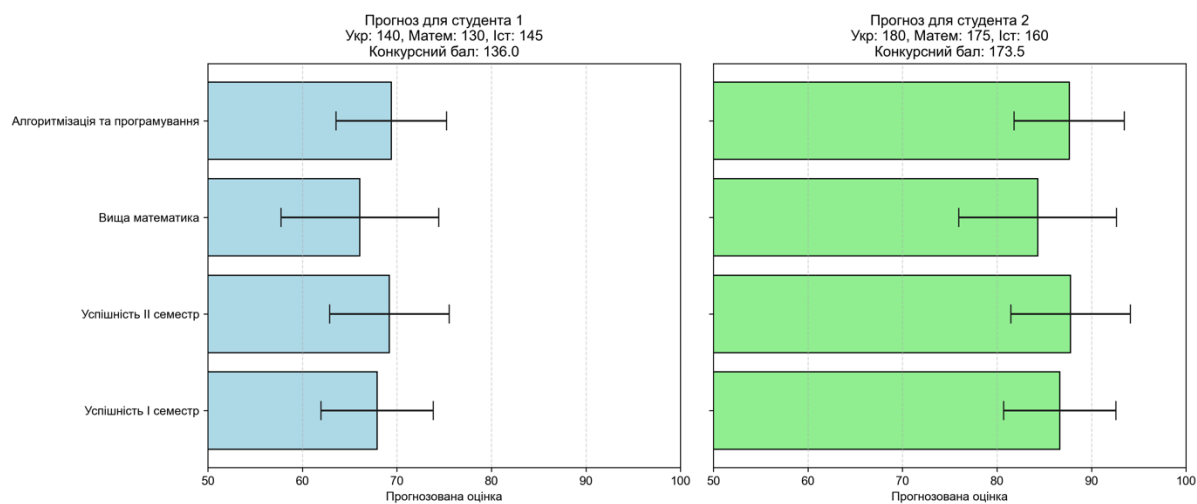


Рисунок В.10 – Результат точкового та інтегрального прогнозування успішності окремих студентів на основі показників НМТ

Додаток Г (довідниковий)

Інструкція користувача

1. Вимоги до системи. Перед запуском модуля переконайтеся, що на комп'ютері встановлено Python 3.9+ та необхідні бібліотеки:

- pandas
- numpy
- statsmodels
- scikit-learn
- matplotlib
- seaborn
- tkinter

2. Запуск модуля. Скачайте та розпакуйте архів з проєктом. Запустіть головний файл.

3. Завантаження вхідних даних. Файл з даними повинен бути у форматі CSV. Обов'язкові колонки: nmt_math, nmt_ukr, nmt_history – результати НМТ, certificate_score – середній бал атестата, region, urbanity, – інші соціально-економічні фактори

4. Попередня обробка даних. Після завантаження дані проходять очищення: пропущені значення, кодування категоріальних змінних, масштабування. Якщо дані некоректні, з'явиться відповідне повідомлення.

5. Формування прогнозу. Натисніть кнопку "Сформувати прогноз". Модуль обробить дані через навчений класифікатор. Результат прогнозу – очікуваний рівень академічної успішності відобразиться графічно у вікні або таблиці.

6. Візуалізація результатів. Автоматично генеруються графіки: розподіл вхідних даних та прогнозованого рівня успішності, кореляційна матриця факторів, важливість ознак.

7. Збереження результатів. Натисніть "Зберегти результати", щоб експортувати прогноз у __.csv. Файл буде містити початкові дані та значення прогнозу.

8. Завершення роботи. Закрийте вікно або натисніть "Вийти" для завершення сесії. Усі тимчасові файли буде автоматично очищено (опціонально).

Типові помилки:

Помилка	Причина	Рішення
FileNotFoundError	Не знайдено CSV-файл	Перевірте шлях до файлу
ValueError: could not convert	Некоректні типи даних	Очистіть/відформатуйте вхідний CSV
ModuleNotFoundError	Не встановлено бібліотеки	Виконайте <code>pip install</code> для потрібних пакетів