МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Кваліфікаційна наукова
праця на правах рукопису

ПИЛИПЕНКО ВЛАДИСЛАВ ІГОРОВИЧ

УДК 004.85 + 004.4 + 004.62

ДИСЕРТАЦІЯ

МОДЕЛІ ТА МЕТОДИ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ОСВІТИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

Дисертація на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ В.І. Пилипенко

Науковий керівник Стаценко Володимир Володимирович, доктор технічних наук, професор

Київ – 2026

АНОТАЦІЯ

Пилипенко В.І. Моделі та методи прогнозування успішності здобувачів освіти на основі машинного навчання. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 – «Комп'ютерні науки» (12 – Інформаційні технології). Київський національний університет технологій та дизайну, Київ, 2026.

Дисертаційна робота присвячена вирішенню актуальної науково-практичної задачі покращення точності прогнозування академічної успішності студентів за допомогою моделей машинного навчання на основі даних взаємодії з відеонавчальним контентом у системі управління навчанням Moodle.

Зміст анотації. Дисертаційне дослідження присвячене розв'язанню актуального науково-практичного завдання підвищення точності прогнозування академічної успішності здобувачів освіти із використанням методів машинного навчання в системах управління навчанням, що розглядаються як інструмент раннього виявлення ризиків академічної неуспішності. Основна частина роботи складається зі вступу, чотирьох розділів та висновків.

У вступі розглянуто актуальність теми дисертаційного дослідження, сформульовано мету, завдання дослідження, необхідні для її досягнення, виділено наукову новизну отриманих результатів та їх практичну цінність.

У **першому розділі** дисертаційної роботи виконано аналітичний огляд сучасних наукових підходів та інформаційних технологій у сфері прогнозування академічної успішності здобувачів освіти в системах дистанційного навчання. Досліджено особливості структури, організації та накопичення даних у системах управління навчанням, зокрема LMS Moodle, а також визначено основні категорії освітніх даних, що можуть бути використані для задач освітньої аналітики та прогнозування. Проаналізовано сучасні методи обробки великих даних, підходи до попередньої

обробки та формування ознакового простору, а також методи інтелектуального аналізу даних і машинного навчання, що застосовуються для прогнозування академічної успішності студентів. Виконано систематизацію сучасних досліджень у даній предметній області, розглянуто переваги та обмеження існуючих моделей прогнозування, а також проаналізовано типи даних, що використовуються у відповідних дослідженнях. За результатами аналізу встановлено, що більшість існуючих підходів ґрунтуються переважно на використанні традиційних освітніх показників, тоді як поведінкові характеристики взаємодії студентів із навчальним контентом, зокрема відеоматеріалами, залишаються недостатньо дослідженими. Це обумовлює актуальність використання поведінкової освітньої аналітики та методів машинного навчання для підвищення точності прогнозування академічної успішності студентів. На основі проведеного аналізу сформульовано мету, задачі, об’єкт та предмет дисертаційного дослідження, а також обґрунтовано доцільність розроблення ансамблевих моделей прогнозування на основі інтеграції освітніх та поведінкових даних.

У **другому розділі** дисертаційної роботи досліджено сучасні методи та алгоритми машинного навчання, що використовуються для розв’язання задач класифікації та прогнозування академічної успішності здобувачів освіти. Проведено порівняльний аналіз моделей на основі логістичної регресії (Logistic Regression), наївного баєсівського класифікатора (Naive Bayes), методу опорних векторів (Support Vector Machines), випадкового лісу (Random Forest) та нейронних мереж (Neural Network). Розглянуто принципи побудови моделей класифікації, особливості їх застосування в задачах освітньої аналітики, а також методи оцінювання ефективності моделей машинного навчання. Для забезпечення коректності порівняльного аналізу всі моделі були навчено та протестовано на однаковому наборі даних із використанням єдиної стратегії формування тренувальної та тестової вибірок. Проведено експериментальне дослідження ефективності алгоритмів класифікації на

основі показників навчальної активності студентів, зокрема оцінок та відвідуваності занять. Оцінювання моделей виконувалося із застосуванням метрик Accuracy, Sensitivity, Specificity, F1-score та ROC-AUC, що дозволило комплексно оцінити якість класифікації та узагальнювальну здатність моделей. За результатами порівняльного аналізу встановлено, що алгоритми випадкового лісу та нейронних мереж забезпечують найкращі результати прогнозування академічної успішності студентів та характеризуються найбільш ефективним балансом між точністю класифікації й здатністю до узагальнення. Водночас виявлено обмежену здатність моделей до розпізнавання студентів групи ризику, що обумовлює необхідність подальшого вдосконалення моделей шляхом розширення ознакового простору, оптимізації параметрів та використання ансамблевих підходів машинного навчання. Отримані результати стали основою для подальшого дослідження поведінкових характеристик взаємодії студентів із навчальним контентом та розроблення ансамблевої стекінгової моделі прогнозування академічної успішності.

У **третьому розділі**, досліджено вплив поведінкових характеристик взаємодії студентів із навчальними відеоматеріалами на якість прогнозування академічної успішності. Для побудови моделей прогнозування використано інтегрований набір ознак, що включає показники відвідуваності занять, академічних оцінок та поведінкової активності під час перегляду навчальних відеоматеріалів. Дані про взаємодію користувачів із відеоматеріалами отримано з електронного журналу, бази даних LMS Moodle та розробленого плагіну VideoPlayer, інтегрованого в систему управління навчанням університету. На основі отриманих подій взаємодії сформовано додатковий ознаковий простір, який включає показники переглядів, тривалості перегляду, пауз, перемотувань та повторних переглядів відеоконтенту. У розділі проведено побудову та дослідження моделей машинного навчання для прогнозування академічної успішності студентів із використанням розширеного набору поведінкових та освітніх ознак. Результати експериментального дослідження

показали, що використання поведінкової освітньої аналітики дозволяє суттєво підвищити якість прогнозування. Зокрема, моделі на основі випадкового лісу та нейронних мереж продемонстрували найвищі показники точності – 87.1% та 85.3% відповідно, що перевищує результати логістичної регресії та наївного баєсівського класифікатора в середньому на 8.5%. Встановлено, що додавання поведінкових ознак взаємодії з відеонавчальним контентом забезпечує приріст загальної точності прогнозування приблизно на 10%, підвищення збалансованої точності на 15%, а також збільшення значення ROC-AUC на 14%, що підтверджує високу інформативність поведінкових характеристик у задачах освітньої аналітики. Для підвищення точності прогнозування та зменшення похибки узагальнення у роботі запропоновано дворівневу ансамблеву стекінгову модель, побудовану на основі комбінування базових моделей різної природи, зокрема логістичної регресії, наївного баєсівського класифікатора та випадкового лісу. Як мета-модель використано алгоритм градієнтного бустингу, що забезпечує інтеграцію результатів базових класифікаторів та компенсацію їхніх похибок. Проведений порівняльний аналіз показав, що запропонована стекінгова модель забезпечує найкращі результати прогнозування серед усіх досліджених підходів, досягаючи загальної точності 90.2%, чутливості 97.5%, збалансованої точності 85% та ROC-AUC 92.6%, що свідчить про високу узагальнювальну здатність та ефективність ансамблевого підходу в задачах прогнозування академічної успішності здобувачів.

У **четвертому розділі** представлено функціональну та програмну реалізацію системи збору й аналізу поведінкової освітньої аналітики, а також інтеграцію розробленого плагіну відеоплеєра в систему управління навчанням Moodle. Реалізований програмний модуль забезпечує автоматизований збір показників взаємодії користувачів із навчальними відеоматеріалами, зокрема даних про тривалість перегляду, паузи, перемотування та повторні перегляди, які використовуються для побудови моделей прогнозування академічної успішності

студентів. Наведено принципи організації зберігання та обробки освітніх і поведінкових даних у межах реалізованої системи прогнозування, в якій апробовано запропоновані методи та моделі машинного навчання. Додатково представлено програмний застосунок для формування аналітичної звітності та вибірок щодо академічної успішності здобувачів освіти, який забезпечує підтримку процесів моніторингу навчальної діяльності та аналізу результатів прогнозування.

Основні наукові результати дисертації опубліковано у 23 працях, зокрема: десять статей – у наукових фахових періодичних виданнях України; одна стаття – у наукових періодичних виданнях іншої держави (Scopus, Q3); дванадцять публікацій – у матеріалах міжнародних та всеукраїнських наукових, науково-технічних, науково-практичних конференцій (три із них входять до наукометричної бази Scopus);

Ключові слова: прогнозування академічної успішності, машинне навчання, освітня аналітика, LMS Moodle, відеоаналітика, ансамблеві моделі, стекінг, градієнтний бустинг, випадковий ліс, логістична регресія, нейронні мережі.

ABSTRACT

Pylypenko V.I. Models and methods for predicting student success based on machine learning. – Qualification scientific work in the form of a manuscript. Dissertation for the degree of Doctor of Philosophy in the specialty 122 – "Computer Science" (12 – Information Technologies). Kyiv National University of Technologies and Design, Kyiv, 2026.

The dissertation is devoted to solving the current scientific and practical problem of improving the accuracy of predicting the academic success of students using machine learning models based on data from interaction with video educational content in the Moodle learning management system.

Contents of the abstract. The dissertation research is devoted to solving the current scientific and practical problem of increasing the accuracy of predicting the academic success of education seekers using machine learning methods in learning management systems, which are considered as a tool for early detection of risks of academic failure. The main part of the work consists of an introduction, four sections and conclusions. The introduction considers the relevance of the topic of the dissertation research, formulates the goal and objectives of the research necessary to achieve it, and highlights the scientific novelty of the results obtained and their practical value.

The first section of the dissertation provides an analytical review of modern scientific approaches and information technologies in the field of predicting the academic success of students in distance learning systems. The features of the structure, organization and accumulation of data in learning management systems, in particular LMS Moodle, are studied, and the main categories of educational data that can be used for educational analytics and forecasting tasks are identified. Modern methods of big data processing, approaches to pre-processing and feature space formation, as well as methods of data mining and machine learning used to predict students' academic success are analyzed. Modern research in this subject area is systematized, the advantages and limitations of existing forecasting models

are considered, and the types of data used in relevant studies are analyzed. The analysis results show that most existing approaches are based mainly on the use of traditional educational indicators, while the behavioral characteristics of students' interaction with educational content, in particular video materials, remain insufficiently studied. This determines the relevance of using behavioral educational analytics and machine learning methods to increase the accuracy of predicting students' academic success. Based on the analysis, the goal, objectives, object and subject of the dissertation research are formulated, and the feasibility of developing ensemble forecasting models based on the integration of educational and behavioral data is substantiated.

The second section of the dissertation explores modern machine learning methods and algorithms used to solve classification problems and predict academic success of students. A comparative analysis of models based on Logistic Regression, Naive Bayes classifier, Support Vector Machines, Random Forest, and Neural Networks is conducted. The principles of constructing classification models, the features of their application in educational analytics tasks, and methods for assessing the effectiveness of machine learning models are considered. To ensure the correctness of the comparative analysis, all models were trained and tested on the same data set using a single strategy for forming training and test samples. An experimental study of the effectiveness of classification algorithms based on indicators of student academic activity, in particular grades and class attendance, is conducted. The models were evaluated using the metrics Accuracy, Sensitivity, Specificity, F1-score and ROC-AUC, which allowed for a comprehensive assessment of the classification quality and generalization ability of the models. The results of the comparative analysis showed that the random forest and neural network algorithms provide the best results in predicting students' academic performance and are characterized by the most effective balance between classification accuracy and generalization ability. At the same time, the limited ability of the models to recognize students at risk was revealed, which necessitates further improvement of the models by expanding the feature space, optimizing

parameters and using ensemble machine learning approaches. The results obtained became the basis for further research into the behavioral characteristics of students' interaction with educational content and the development of an ensemble stacking model for predicting academic performance.

In the third section, the influence of behavioral characteristics of students' interaction with educational video materials on the quality of predicting academic success is investigated. To build prediction models, an integrated set of features was used, which includes indicators of class attendance, academic grades, and behavioral activity while watching educational video content. Data on user interaction with video materials were obtained from the electronic journal, the LMS Moodle database, and the developed VideoPlayer plugin integrated into the university's learning management system. Based on the obtained interaction events, an additional feature space was formed, which includes indicators of views, viewing duration, pauses, rewinds, and repeated viewings of video content. The section builds and studies machine learning models for predicting students' academic success using an expanded set of behavioral and educational features. The results of the experimental study showed that the use of behavioral educational analytics allows significantly improving the quality of prediction. In particular, models based on random forest and neural networks demonstrated the highest accuracy rates – 87.1% and 85.3%, respectively, which exceeds the results of logistic regression and naive Bayesian classifier by an average of 8.5%. It was found that adding behavioral features of interaction with educational video content provides an increase in the overall prediction accuracy by approximately 10%, an increase in balanced accuracy by 15%, as well as an increase in the ROC-AUC value by 14%, which confirms the high informativeness of behavioral characteristics in educational analytics tasks. To increase the prediction accuracy and reduce the generalization error, the paper proposes a two-level ensemble stacking model built on the basis of a combination of basic models of different nature, in particular, logistic regression, naive Bayesian classifier and random forest. The gradient boosting algorithm was used as a meta-model, which provides integration of the

results of the basic classifiers and compensation of their errors. The comparative analysis showed that the proposed stacking model provides the best prediction results among all the studied approaches, achieving an overall accuracy of 90.2%, sensitivity of 97.5%, balanced accuracy of 85%, and ROC-AUC of 92.6%, which indicates a high generalization ability and effectiveness of the ensemble approach in the tasks of predicting students' academic performance.

The fourth section presents the functional and software implementation of the system for collecting and analyzing behavioral educational analytics, as well as the integration of the developed video player plugin into the Moodle learning management system. The implemented software module provides automated collection of indicators of user interaction with educational video content, in particular data on viewing duration, pauses, rewinding and repeated viewings, which are used to build models for predicting students' academic performance. The principles of organizing the storage and processing of educational and behavioral data within the framework of the implemented forecasting system are presented, in which the proposed methods and models of machine learning are tested. Additionally, a software application is presented for generating analytical reporting and samples on the academic performance of education seekers, which provides support for the processes of monitoring educational activities and analyzing forecasting results.

The main scientific results of the dissertation were published in 23 works, including: ten articles in scientific professional periodicals of Ukraine; one article in scientific periodicals of another state (Scopus, Q3); twelve publications in the materials of international and all-Ukrainian scientific, scientific and technical, scientific and practical conferences (three of them are included in the scientometric database Scopus);

Keywords: academic performance prediction, machine learning, educational analytics, LMS Moodle, video analytics, ensemble models, stacking, gradient boosting, random forest, logistic regression, neural networks.

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, у яких опубліковані основні наукові результати дисертації:

Публікації у зарубіжних та наукових фахових виданнях України, внесених до міжнародних наукометричних баз:

1) Пилипенко В., & Стаценко В. (2023). ПРОГНОЗУВАННЯ АКТИВНОСТІ КОРИСТУВАЧІВ ПЛАТФОРМИ MOODLE НА БАЗІ МЕТОДІВ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical Sciences, 323(4), 257–261. DOI: 10.31891/2307-5732-2023-323-4-257-261
2) Пилипенко В., & Стаценко В. (2024). ВИКОРИСТАННЯ ТЕСТУ СТЬЮДЕНТА ДЛЯ СТАТИСТИЧНОГО АНАЛІЗУ ДАНИХ ОПИТУВАННЯ КОРИСТУВАЧІВ MOODLE. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (1), 226–230. https://doi.org/10.31891/2219-9365-2024-77-29
3) Стаценко В., & Пилипенко В. (2024). ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ МОДЕЛІ ПРОГНОЗУВАННЯ УСПІШНОСТІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical Sciences, 331(1), 271-276. https://doi.org/10.31891/2307-5732-2024-331-41
4) Пилипенко В., Стаценко В. (2024). ДОСЛІДЖЕННЯ ТОЧНОСТІ МЕТОДІВ МАШИННОГО НАВЧАННЯ ПРИ ПРОГНОЗУВАННІ УСПІШНОСТІ ЗДОБУВАЧІВ. Herald of Khmelnytskyi National University. Technical Sciences, 335(3(1), 349-356. https://doi.org/10.31891/2307-5732-2024-335-3-47
5) Пилипенко, В., & Стаценко, В. (2024). ВИКОРИСТАННЯ ДВОРІВНЕВОГО МЕТОДУ СТЕКОВОГО АНСАМБЛЮ ДЛЯ ПОКРАЩЕННЯ ТОЧНОСТІ ПРОГНОЗУВАННЯ УСПІШНОСТІ. Наука і техніка сьогодні, 9 (37), 763-774. https://doi.org/10.52058/2786-6025-2024-9(37)-763-774
6) Пилипенко, В., Скідан, В., & Волівач, А. (2024). АНАЛІЗ ОПИТУВАННЯ ЩОДО ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ. Herald of Khmelnytskyi National

University. Technical Sciences, 345(6(2), 108-112. https://doi.org/10.31891/2307-5732-2024-345-6-16

7) Pylypenko, V., Statsenko, V., Bila, T., & Statsenko, D. (2024). Determining the influence of data on working with video materials on the accuracy of student success prediction models. Eastern-European Journal of Enterprise Technologies, 5(4 (131), 52–62. https://doi.org/10.15587/1729-4061.2024.313333
8) Пилипенко, В., & Стаценко, В. (2025). ПЛАГІН ДЛЯ ЗБОРУ ДАНИХ ВЗАЄМОДІЇ КОРИСТУВАЧІВ MOODLE З ВІДЕО МАТЕРІАЛАМИ. Наука і техніка сьогодні, 1(42), 1318-1330. https://doi.org/10.52058/2786-6025-2025-1(42)-1318-1330
9) Пилипенко В. (2025). Прогнозування високого рівня академічної успішності здобувачів з використанням машинного навчання. Наука і техніка сьогодні, 8(49), 1634-1649. https://doi.org/10.52058/2786-6025-2025-8(49)-1634-1649
10) PYLYPENKO, V. (2025). THE EFFECT OF TRAINING SAMPLE SIZE ON THE STABILITY OF CLASSIFICATION MODELS. Technologies and Engineering, 26(6), 32-44. https://doi.org/10.30857/2786-5371.2025.6.3
11) PYLYPENKO, V. (2026). IMPACT OF STACKING ENSEMBLE DEPTH ON GENERALIZATION ABILITY OF ACADEMIC PERFORMANCE PREDICTION MODELS. Technologies and Engineering, 27(1), 72-79. https://doi.org/10.30857/2786-5371.2026.1.7

2. Опубліковані наукові праці апробаційного характеру:

12) Стаценко, В. В., & Пилипенко, В. І. (2023). Оцінка ефективності моделі прогнозування активності користувачів Moodle методами машинного навчання. VII Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2023», 2023, с. 28-29. https://test.knutd.edu.ua/bitstream/123456789/25953/1/MSIE_2023_P028-029.pdf
13) Statsenko, V. V., Pavlenko, V. M., & Pylypenko, V. I. (2023). Choise problem in learning management systems, Digital transformation and technologies for the

sustainable development all branches of modern education, science and practice, MANS w Łomży, 125-129.

14) Pavlenko, V., Ponomarenko, I., Morhulets, O., Fedorchenko, A., Chorna, O., & Pylypenko, V. (2023, October). Creating Educational Products With Using Data Science and Digital Marketing. In 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek) (pp. 1-4). IEEE.

15) Volodymyr Pavlenko, Ihor Ponomarenko, Oksana Morhulets, Andrii Fedorchenko, Vladyslav Pylypenko: Use of Information Technologies and Marketing Tools for The Formation of An Educational Platform. ITTAP 2023: 702-708

16) Стаценко, В. В., & Пилипенко, В. І. (2024). АНАЛІЗ ПРОГНОЗНОЇ АНАЛІТИКИ ОСВІТНІХ РИЗИКІВ У СИСТЕМАХ УПРАВЛІННЯ НАВЧАННЯМ. НАПРЯМ № 1 ВОЄННА НАУКА. НАЦІОНАЛЬНА БЕЗПЕКА, 282-285.

17) Volodymyr Statsenko, Pylypenko Vladyslav, Skidan, Vladyslava, Volivach, Antonina. (2024). Investigation of the Accuracy of Machine Learning Methods in Prediction of Students Success. 1-4. 10.1109/KhPIWeek61434.2024.10877975.

18) Pylypenko Vladyslav, Statsenko Volodymyr: DEVELOPMENT OF A MOODLE PLUG-IN USING AJAX REQUEST FOR ASYNCHRONOUS DATA TRANSFER. Proceedings of the XXXIII International Scientific and Practical Conference. Seville, Spain. 2024. Pp. 7-14, URL: https://isg-konf.com/scientific-developments-of-young-scientists-to-improve-life/

19) Pylypenko Vladyslav, Statsenko Volodymyr: INCREASING THE ACCURACY OF PREDICTION OF STUDENT SUCCESS FOR A MODEL WITH A RANDOM FOREST ALGORITHM. Proceedings of the I International Scientific and Practical Conference. Boston, USA. 2024. Pp. 9-12, URL: https://isg-konf.com/innovative-scientific-research-theory-methodology-practice/

20) Pylypenko Vladyslav, Statsenko Volodymyr: STACKED ENSEMBLE MACHINE LEARNING ALGORITHM IN PREDICTION OF STUDENT SUCCESS. Proceedings

of the II International Scientific and Practical Conference. Copenhagen, Denmark. 2024. Pp. 8-11, URL: https://isg-konf.com/integration-of-science-and-practice-as-a-mechanism-of-effective-development/

21) Statsenko Volodymyr, Pylypenko Vladyslav: Development of a Moodle video player plug-in for user interaction analysis. VIII Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2024», 2024, с. 266-268.

22) Пилипенко, В. І., & Стаценко, В. В. (2025). Прогнозування академічної успішності здобувачів за допомогою методів машинного навчання. IV Міжнародна науково-практична інтернет конференція молодих учених та здобувачів «ЕЛЕКТРОМЕХАНІЧНІ, ІНФОРМАЦІЙНІ СИСТЕМИ ТА НАНОТЕХНОЛОГІЇ», 2025, с.134-135.

23) Пилипенко, В. І. (2026). Вплив розміру навчальної вибірки на стабільність та узагальнювальну здатність моделей класифікації. Збірник наукових праць IX Міжнародної науково-практичної конференції «Мехатронні системи: інновації та інжиніринг» – «MSIE-2026», (с. 287–290).

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

NB – Naive Bayes (Наївний Баєс);

RF – Random Forest (випадковий ліс);

NN – Neural networks (нейронні мережі);

LR – Logistic Regression (логістична регресія);

ML – Machine Learning (машинне навачання);

TP – True Positive (істинно позитивні випадки);

TN – True Negative (істинно негативні випадки);

FP – False Positive (хибно позитивні випадки);

FN – False Negative (хибно негативні випадки);

AUC – Area Under the ROC Curve (площа під кривою);

ROC – Receiver Operating Charachteristic (операційна характеристика класифікатора);

CSV – comma-separated values (значення, розділені комою);

LMS – Leaning Management System (система управління навчанням);

HTTP – hyper-text transfer protocol (протокол передачі гіпертексту);

JSON – javascript object notation (нотація об’єкта мови програмування Javascript);

ВСТУП

Актуальність теми. Системи управління навчанням відносяться до покоління систем ІКТ (інформаційно-комунікаційних технологій), які в основному інтегрують процес навчання, оцінювання, ведення журналів та комунікації для моніторингу, контролю та взаємодії зі здобувачами, щоб забезпечити їх інтелектуальними програмами та послугами. В останні роки прогнозування успішності стало одним із важливих напрямків в освітньому процесі. Це мотивувало на виконання багатьох досліджень з питань оцінки освітніх ризиків, прогнозуванні академічної успішності та прогнозній аналітиці. Прогнозування успішності, як одного з основних складників освітнього процесу, стає однією з найважливіших до кінця не вирішених проблем для закладів вищої освіти. Не кожна система управління навчанням має функціонал для виконання прогнозування успішності та оцінки освітніх ризиків, більшість дають лише загальну прогнозну аналітику, без можливості створення власних моделей для прогнозування та аналізу. На сьогоднішній день в сфері освіти застосовується широкий спектр систем управління навчанням – від адміністративних та моніторингових до систем підтримуючих повноцінне електронне навчання з процесом адміністрування. За сприяння буму в електронному навчанні, швидкому доступу до навчальних матеріалів, доступності з будь-якого місця без обмежень, матеріальній доступності та простоті використання системи управління навчанням активно використовуються в освітньому процесі. Системи розширили свої можливості в електронному навчанні та отримали змогу замінити або доповнити очне чи офлайн навчання. Дані системи контролюють і управляють процесом навчання в реальному (або в близькому до реального) часі, як правило, з циклами зворотного зв’язку, де результати процесу навчання подаються в звітах та статистичних зведеннях.

Популярність методів машинного навчання (ML) при прогнозуванні успішності здобувачів зросла значною мірою протягом останніх десятиліть. Завдяки сучасним

комп'ютерам та хмарним технологіям можна швидко обробляти великі масиви даних, що робить застосування складних ML-алгоритмів більш доступним. А прогрес у галузі алгоритмів машинного навчання, включаючи глибоке навчання, які часто перевершують традиційні статистичні методи у точності прогнозування ризиків, робить їх незамінними та пріорітетними у використанні. Методи машинного навчання все більше інтегруються в різні галузі, включаючи і освітній сектор. Це явище пояснюється кількома ключовими факторами: здатність обробляти великі обсяги даних і знаходити складні закономірності робить їх дуже точними в прогнозуванні, моделі можуть адаптуватися до нових даних і змінюватися з часом, що дозволяє їм залишатися актуальними, автоматизація процесу аналізу даних знижує потребу в ручній праці і підвищує ефективність, здатність виявляти складні та приховані закономірності в даних, які неможливо виявити за допомогою традиційних методів аналізу. Тому чим більший обсяг інформації ознак успішності про здобувача, тим більша точність прогнозування. Важливими ознаками при прогнозуванні є показники, що змінюються в динаміці освітнього процесу, а саме: активність в освітньому процесі, відвідуваність, ступінь взаємодії з навчальними матеріалами, виконання навчальних завдань. Більшість прогнозних моделей, як правило, використовують лише дані з журналів Moodle та значення балів (оцінок по дисциплінам, тестам і модулям) отриманих за попередні навчальні періоди, тому виникає певна межа точності прогнозування, оскільки дані про перегляд завдання (відкриття файлу чи документу) не містять повної інформації про фактичну роботу здобувача з документом. Дані про реальну взаємодію з навчальними матеріалами, такі як: тривалість перегляду, статус чи додивились до кінця відео, кількість зупинок та перемоток дадуть більше розуміння як здобувач насправді працював. Створення плагіна для системи Moodle, що дозволятиме збиратиме дані взаємодії здобувачів з навчальними матеріалами, дозволить отримати об’єктивні дані про роботу здобувачів з цими матеріалами та розширить набір даних для прогнозування. Важливим аспектом є виявлення проблеми з успішністю на ранній стадії навчання, щоб викладач та

адміністрація закладу мали змогу з цим ознайомитись. На теперішній час розробка та дослідження методів та засобів оцінювання і прогнозування освітніх ризиків здобувачів в системах управління навчанням є актуальним науковим завданням.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота відповідає науковому напряму кафедри «Комп'ютерних наук» Київського національного університету технологій та дизайну. Ініціативна тема: 0122U200947. Дата реєстрації: 02-10-2022. Дослідження Internet-технологій для побудови систем управління дистанційним навчанням.

Мета і задачі дослідження. Метою дисертаційного дослідження є розв'язок науково-практичної задачі прогнозування академічної успішності студентів шляхом дослідження сучасних моделей машинного навчання та розробки нової моделі, яка дозволить підвищити достовірність прогнозування успішності на основі розширеної інформації про дії здобувачів освіти під час роботи в електронних системах управління навчанням, їх взаємодію з навчальними матеріалами, зокрема, з відеоматеріалами.

Для досягнення поставленої мети необхідне розв'язання наступних задач:

1) Проаналізувати структуру та характер інформації, що зберігається в електронних системах управління навчанням.
2) Дослідити використання сучасних методів машинного навчання для задач прогнозування академічної успішності, визначити їх переваги, обмеження та можливі шляхи підвищення достовірності результатів.
3) Побудувати моделі машинного навчання на основі відомих алгоритмів, виконати їх навчання із використанням інформації про відвідування здобувачами освіти навчальних занять та отримані оцінки. Виконати порівняльний аналіз розроблених моделей.

4) Розробити для LMS Moodle програмний модуль збору інформації про взаємодію здобувачів освіти з навчальними відеоматеріалами.
5) Сформувати розширений перелік ознак, що можуть використовуватись для навчання моделей машинного навчання з урахуванням даних взаємодії здобувачів освіти з навчальними відеоматеріалами.
6) Провести навчання моделей машинного навчання, створених на основі відомих алгоритмів, із використанням даних з розширеним переліком ознак. Визначити вплив навчання моделей на основі даних з розширеним переліком ознак на достовірність прогнозування.
7) На основі отриманих результатів, розробити нову модель прогнозування академічної успішності, що передбачатиме можливість проведення навчання на основі розширеного переліку ознак.
8) Визначити достовірність прогнозів отриманих на основі розробленої моделі, порівняти їх з прогнозами моделей, отриманих у попередніх дослідженнях.
9) Розробити програмне забезпечення для візуалізації та аналізу результатів прогнозування.

Об’єктом досліджень є процес прогнозування академічної успішності студентів в електронних системах управління навчанням на основі даних про роботу здобувачів освіти та їх взаємодію з навчальним матеріалами.

Предметом досліджень є моделі та методи прогнозування успішності здобувачів освіти на основі машинного навчання.

Методи досліджень. Для розв’язання поставлених у дисертаційній роботі задач використані відомі алгоритми та методи машинного навчання для задач класифікації і прогнозування, методи статистичного аналізу даних, а також методи аналізу великих даних у системах освітньої аналітики. Для оцінювання достовірності прогнозів отриманих за допомогою розроблених моделей використано методи аналізу даних,

зокрема метрики класифікації (Accuracy, Precision, Recall, F1-score, ROC-AUC), а також методи оцінювання узагальнювальної здатності моделей. Для реалізації моделей та програмних компонентів системи використано методи об'єктно-орієнтованого програмування та сучасні програмні засоби реалізації алгоритмів машинного навчання.

Наукова новизна одержаних результатів.

1. Вперше розроблено 2-рівневу стекінгову модель прогнозування академічної успішності здобувачів освіти, яка відрізняється від відомих моделей використанням на першому рівні ансамблю алгоритмів логістичної регресії, випадкового лісу та нейронної мережі, а на другому рівні – алгоритму градієнтного бустингу, що забезпечує підвищення достовірності прогнозування.

2. Удосконалено метод розширення простору ознак для задач прогнозування академічної успішності, за рахунок включення характеристик взаємодії здобувачів освіти з навчальними відеоматеріалами в LMS, що дозволило підвищити інформативність навчальних вибірок та достовірність прогнозування.

3. Удосконалено методи збору та підготовки даних для задач прогнозування успішності здобувачів освіти, що передбачає автоматизоване отримання і накопичення даних про взаємодію користувачів з навчальними відеоматеріалами та їх інтеграцією у процес побудови моделей машинного навчання.

4. Набули подальшого розвитку методи аналізу освітніх даних, зокрема щодо оцінювання впливу поведінкових характеристик роботи користувачів електронного освітнього середовища на результати прогнозування академічної успішності.

Практичне значення одержаних результатів.

1. Розроблено програмний модуль для LMS Moodle, який забезпечує автоматизований збір, накопичення та обробку даних про взаємодію здобувачів освіти з навчальними відеоматеріалами.

2. Реалізовано програмні засоби підготовки даних та формування розширеного набору ознак для навчання моделей машинного навчання в задачах прогнозування успішності здобувачів освіти.
3. Розроблено та експериментально досліджено комплекс моделей машинного навчання для прогнозування академічної успішності, включаючи логістичну регресію, метод опорних векторів, Наївний Баєсівський класифікатор, випадковий ліс, нейронну мережу та запропоновану 2-рівневу стекінгову модель.
4. Встановлено, що використання розробленої 2-рівневої стекінгової моделі забезпечує підвищення точності прогнозування на 14,3 % порівняно з моделлю логістичної регресії, на 4,56 % порівняно з нейронною мережею та на 2,29 % порівняно з моделлю випадкового лісу.
5. Розроблено програмне забезпечення для візуалізації результатів прогнозування та формування аналітичних звітів у форматах PDF та XLS, яке може використовуватися як компонент інтелектуальних інформаційних систем підтримки прийняття рішень у закладах освіти.
6. Результати роботи впроваджені в освітній процес Київського національного університету технологій та дизайну та Хмельницького національного університету.
7. Результати дослідження можуть бути використані при створенні та модернізації інформаційних систем освітньої аналітики, систем підтримки прийняття рішень, а також інтелектуальних сервісів моніторингу освітнього процесу.

Запропоновані методи і моделі впроваджені у навчальний процес Київського національного університету технологій та дизайну при викладанні дисциплін: «Проєктування інтерфейсу користувача» та «Комп’ютерні технології та програмування» для отримання даних відеоаналітики при визначенні приросту точності моделей прогнозування успішності. Додаток В (підтверджено актом впровадження від 23 грудня 2024 року).

Запропоновані методи і моделі впроваджені у навчальний процес кафедри Хмельницького національного університету технологій. Впровадження полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів так і в циклах лабораторних робіт для отримання відеоаналітики при визначенні приросту точності моделей прогнозування успішності. Зокрема при викладанні дисципліни «Інтелектуальний аналіз даних»: Додаток В (підтверджено актом впровадження від 26 грудня 2025 року).

Особистий внесок здобувача. Основний зміст роботи, всі теоретичні та практичні результати, висновки і дослідження, які представлено до захисту, одержані автором особисто. Наукові публікації опубліковані самостійно. У наукових працях, опублікованих у співавторстві, автору належать: розроблення програмних засобів для аналізу та класифікації освітніх даних, побудова та дослідження моделей прогнозування академічної успішності студентів, формування та аналіз ознакового простору на основі поведінкових характеристик взаємодії з навчальним контентом, дослідження впливу поведінкових показників на результати прогнозування, розроблення програмних компонентів для отримання додаткових ознак, а також проведення експериментальних досліджень, аналіз та інтерпретація отриманих результатів.

Апробація роботи. Основні теоретичні положення та практичні результати дисертаційної роботи доповідалися і обговорювались на таких конференціях: VII Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE – 2023» «Оцінка ефективності моделі прогнозування активності користувачів Moodle методами машинного навчання», Київ, 23 листопада 2023 рік, XXXIII International Scientific and Practical Conference «DEVELOPMENT OF A MOODLE PLUG-IN USING AJAX REQUEST FOR ASYNCHRONOUS DATA TRANSFER», Seville, Spain. 2024, I International Scientific and Practical Conference «INCREASING THE ACCURACY OF PREDICTION OF STUDENT SUCCESS FOR A

MODEL WITH A RANDOM FOREST ALGORITHM» Boston, USA. 2024, IEEE 5th KhPI Week on Advanced Technology «Investigation of the accuracy of machine learning methods in prediction of students success» – 2024, 7 – 10 жовтня, VIII Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2024» «Development of a Moodle video player plug-in for user interaction analysis», Київ, 23 листопада 2024 рік. IV Міжнародна науково-практична інтернет конференція молодих учених та здобувачів «ЕЛЕКТРОМЕХАНІЧНІ, ІНФОРМАЦІЙНІ СИСТЕМИ ТА НАНОТЕХНОЛОГІЇ» Прогнозування академічної успішності здобувачів за допомогою методів машинного навчання, 2025, с.134-135. IX Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2026» Вплив розміру навчальної вибірки на стабільність та узагальнювальну здатність моделей класифікації, (с. 287–290).

Публікації. У 11 наукових публікаціях повністю відображені основні результати дисертації, з них: 10 статті у наукових фахових виданнях України; 1 публікація у наукових виданнях, які входять до міжнародних наукометричних баз (з них 1 стаття у науковому періодичному виданні іншої держави); 12 тез доповідей та матеріали конференцій.

Структура та обсяг роботи. Повний обсяг дисертації становить 209 сторінок, з яких 163 сторінки основного тексту. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (116 найменувань) та 3 додатків. Робота містить 48 рисунків, 27 таблиць та 3 додатки.

РОЗДІЛ 1. ВИКОРИСТАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ У ЗАДАЧАХ ПРОГНОЗУВАННЯ УСПІШНОСТІ

В першому розділі проведено аналітичне опрацювання широкого спектру інформаційних джерел з проблемної області прогнозування успішності здобувачів в системах управління навчанням. Розглянуто особливості даних в системах управління навчанням та способи їх отримання. Проаналізовано роль та впливи сучасних інформаційних технологій на процеси формування успішності. Проведено аналіз методів обробки великих даних. Проведено оцінювання актуальних інформаційних технологій сформованих на методологічній основі та засобах машинного навчання, їх потенціал у процесах прогнозування успішності. Було розглянуто також недоліки та переваги їх використання на практиці. За результатами огляду сформовано задачі дисертаційного дослідження. Результати розділу опубліковано у працях автора **[1, 2, 3, 4].**

1.1 Особливості структури даних та їх зберігання в системі управління навчанням Moodle

Освіта є ключовим фактором, який визначає майбутні можливості та кар'єрний шлях здобувачів. Разом із поширенням комп'ютерних технологій електронне навчання поступово стає альтернативою традиційним формам освіти **[5]**. Цей перехід від класичної аудиторної методики до онлайн-платформ і цифрових ресурсів не лише забезпечує гнучкість і доступність освіти, але й дозволяє інтегрувати інноваційні методи навчання, такі як інтерактивні курси, аудіо і відео лекції. Завдяки системам управління навчанням (LMS) отримано численні переваги: можливість доступу до освітніх матеріалів з будь-якої точки світу, адаптацію навчального процесу під індивідуальні потреби здобувачів та використання сучасних технологій для моніторингу і оцінювання **[6]**. Однією з таких систем є Moodle, вона має високий рівень визнання в багатьох закладах освіти, безкоштовна і з відкритим кодом, має широкий спектр активних курсів, доступних багатьма мовами **[7]**. Враховуючи

зростаючий обсяг даних, що генеруються в онлайн-навчанні, здатність аналізувати ці дані і використовувати їх для прогнозування академічних досягнень дозволяє своєчасно виявляти здобувачів, які потребують додаткової підтримки. Оцінка і прогнозування успішності виступає ключовим інструментом для аналізу ефективності освітнього процесу та досягнень здобувачів та є складним процесом, оскільки залежить від багатьох факторів, зокрема: відвідуваності, залученості та активності в освітньому процесі, роботі з навчальними матеріалами та виконанні завдань **[8]**. У сучасних освітніх середовищах системи управління навчанням виконують не лише роль інструментів для організації навчального процесу, але й виступають центральними сховищами освітніх даних. Дані про успішність дуже різноманітні та об'ємні, це вимагає додаткових прогностичних та аналітичних інструментів для їх обробки. До них належать оцінки за виконані види робіт, відвідування занять, активність та залученість до навчального процесу, виконання тестових та модульних контролів, завантаження виконаних робіт тощо. З точки зору зберігання даних Moodle демонструє структуровану, масштабовану та надійну архітектуру. Moodle використовує реляційні бази даних (MySQL, PostgreSQL, MariaDB) для централізованого зберігання всієї інформації платформи. Основними структурними компонентами бази є таблиці, які охоплюють усі аспекти функціонування системи: mdl_user – зберігає інформацію про користувачів;

mdl_course – містить дані про курси, їх опис, структуру та пов’язані ресурси;

mdl_assign – зберігає завдання та інформацію про їх виконання;

mdl_grade_grades – зберігає конкретні оцінки користувачів для кожного grade_item

mdl_logstore_standard_log – веде журнал подій та активностей користувачів;

mdl_enrol – містить відомості про реєстрацію на курси.

Загальний вигляд структури основних таблиць бази даних описаних вище, представлено на рис.1.1.

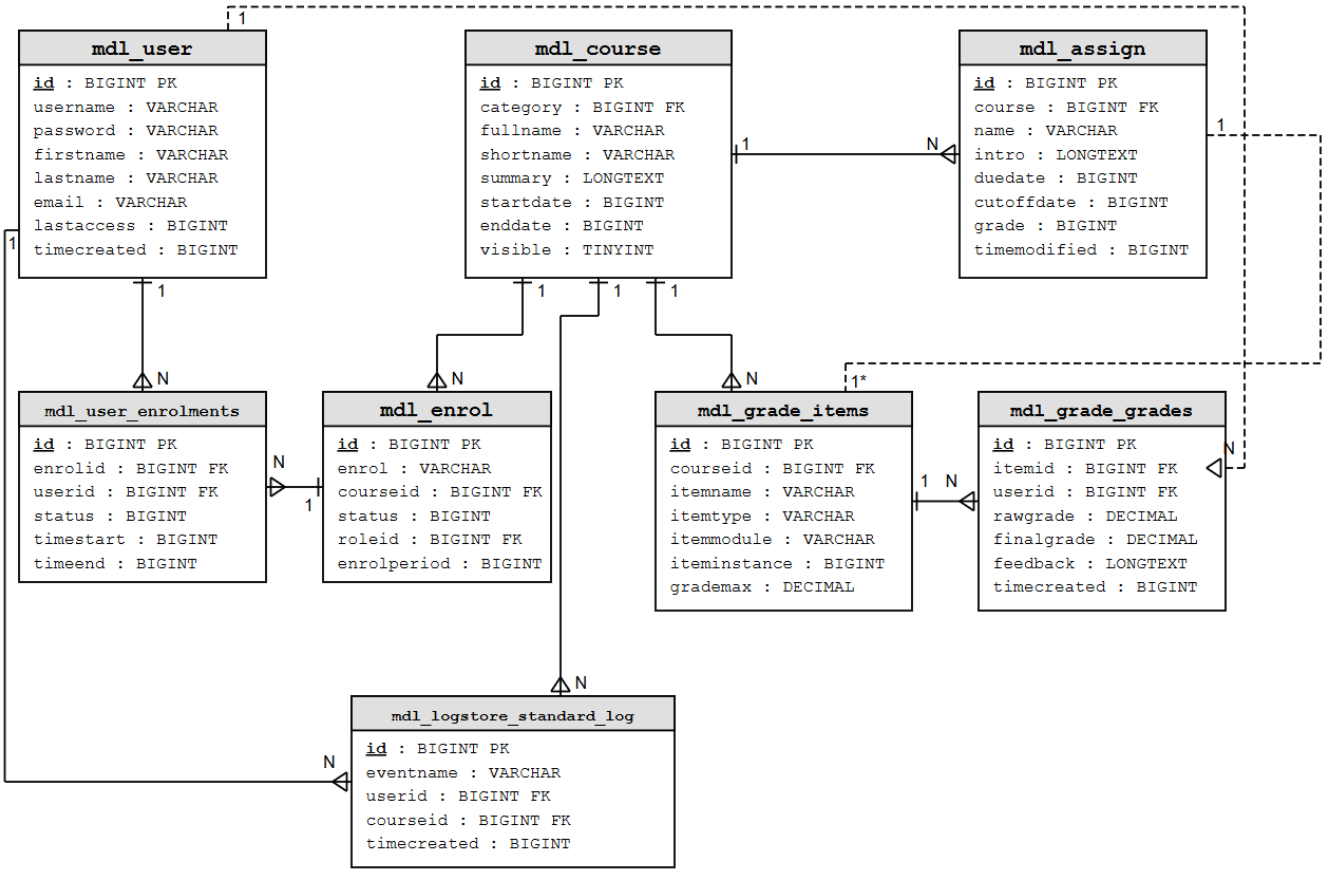


Рис. 1.1 - Структура основних таблиць бази даних

Таблиця mdl_user є однією з основних і містить інформацію про користувачів. Вона зберігає як основні, так і додаткові дані профілю користувача. Опис основних полів, які зазвичай є в таблиці mdl_user (може відрізнятися залежно від версії Moodle та наявних плагінів) представлено на рис.1.2.

Columns: Add Remove Up Down

#	Name	Datatype	Length...	Unsign...	Allow N...	Zerofill	Default	Comm...	Collation	Expression
1	id	BIGINT	10	☐	☐	☐	AUTO...			
2	auth	VARCHAR	20	☐	☐	☐	'manu...		utf8mb4_unico...	
3	confirmed	TINYINT	1	☐	☐	☐	'0'			
4	policyagreed	TINYINT	1	☐	☐	☐	'0'			
5	deleted	TINYINT	1	☐	☐	☐	'0'			
6	suspended	TINYINT	1	☐	☐	☐	'0'			
7	mnethostid	BIGINT	10	☐	☐	☐	'0'			
8	username	VARCHAR	100	☐	☐	☐	''		utf8mb4_unico...	
9	password	VARCHAR	255	☐	☐	☐	''		utf8mb4_unico...	
10	idnumber	VARCHAR	255	☐	☐	☐	''		utf8mb4_unico...	
11	firstname	VARCHAR	100	☐	☐	☐	''		utf8mb4_unico...	
12	lastname	VARCHAR	100	☐	☐	☐	''		utf8mb4_unico...	
13	email	VARCHAR	100	☐	☐	☐	''		utf8mb4_unico...	
14	emailstop	TINYINT	1	☐	☐	☐	'0'			
15	phone1	VARCHAR	20	☐	☐	☐	''		utf8mb4_unico...	
16	phone2	VARCHAR	20	☐	☐	☐	''		utf8mb4_unico...	
17	institution	VARCHAR	255	☐	☐	☐	''		utf8mb4_unico...	
18	department	VARCHAR	255	☐	☐	☐	''		utf8mb4_unico...	
19	address	VARCHAR	255	☐	☐	☐	''		utf8mb4_unico...	
20	city	VARCHAR	120	☐	☐	☐	''		utf8mb4_unico...	
21	country	VARCHAR	2	☐	☐	☐	''		utf8mb4_unico...	
22	lang	VARCHAR	30	☐	☐	☐	'en'		utf8mb4_unico...	
23	calendartype	VARCHAR	30	☐	☐	☐	'grego...		utf8mb4_unico...	
24	theme	VARCHAR	50	☐	☐	☐	''		utf8mb4_unico...	
25	timezone	VARCHAR	100	☐	☐	☐	'99'		utf8mb4_unico...	
26	firstaccess	BIGINT	10	☐	☐	☐	'0'			
27	lastaccess	BIGINT	10	☐	☐	☐	'0'			
28	lastlogin	BIGINT	10	☐	☐	☐	'0'			

Рис. 1.2 – Інформація про користувачів із таблиці mdl_user

В таблиці mdl_course зберігається вся базова інформація про курси, які створюються в системі. Кожен запис у цій таблиці відповідає одному курсу, який бачать користувачі на сайті, загальний вигляд представлено на рис. 1.3. Це можуть бути навчальні дисципліни, тренінги, семінари чи будь-які інші освітні модулі. Унікальний числовий ідентифікатор курсу id використовується в інших таблицях для встановлення зв’язків (наприклад, з темами, модулями, користувачами тощо). Зовнішній ідентифікатор курсу idnumber, може бути використаний для інтеграції з іншими системами (наприклад, обліковими або академічними).

Columns: Add Remove Up Down

#	Name	Datatype	Length...	Unsign...	Allow N...	Zerofill	Default	Comm...	Collation	Expression
1	**id**	**BIGINT**	**10**	☐	☐	☐	**AUTO...**			
2	courseid	BIGINT	10	☐	☑	☐	NULL			
3	categoryid	BIGINT	10	☐	☑	☐	NULL			
4	itemname	VARCHAR	255	☐	☑	☐	NULL		utf8mb4_unico...	
5	itemtype	VARCHAR	30	☐	☐	☐	''		utf8mb4_unico...	
6	itemmodule	VARCHAR	30	☐	☑	☐	NULL		utf8mb4_unico...	
7	iteminstance	BIGINT	10	☐	☑	☐	NULL			
8	itemnumber	BIGINT	10	☐	☑	☐	NULL			
9	iteminfo	LONGTEXT		☐	☑	☐	NULL		utf8mb4_unico...	
10	idnumber	VARCHAR	255	☐	☑	☐	NULL		utf8mb4_unico...	
11	calculation	LONGTEXT		☐	☑	☐	NULL		utf8mb4_unico...	
12	gradetype	SMALLINT	4	☐	☐	☐	'1'			
13	grademax	DECIMAL	10,5	☐	☐	☐	'100.00...			
14	grademin	DECIMAL	10,5	☐	☐	☐	'0.00000'			
15	scaleid	BIGINT	10	☐	☑	☐	NULL			
16	outcomeid	BIGINT	10	☐	☑	☐	NULL			
17	gradepass	DECIMAL	10,5	☐	☐	☐	'0.00000'			
18	multfactor	DECIMAL	10,5	☐	☐	☐	'1.00000'			
19	plusfactor	DECIMAL	10,5	☐	☐	☐	'0.00000'			
20	aggregatio...	DECIMAL	10,5	☐	☐	☐	'0.00000'			
21	aggregatio...	DECIMAL	10,5	☐	☐	☐	'0.00000'			
22	sortorder	BIGINT	10	☐	☐	☐	'0'			
23	display	BIGINT	10	☐	☐	☐	'0'			
24	decimals	TINYINT	1	☐	☑	☐	NULL			
25	hidden	BIGINT	10	☐	☐	☐	'0'			

Рис. 1.3 – Інформація прокурс із таблиці mdl_course

Таблиця mdl_enrol зберігає відомості про методи зарахування (реєстрації) користувачів на курси, представлена на рис. 1.4.

Columns: Add Remove Up Down

#	Name	Datatype	Length...	Unsign...	Allow N...	Zerofill	Default	Comm...	Collation
1	**id**	**BIGINT**	**10**	☐	☐	☐	**AUTO...**		
2	enrol	VARCHAR	20	☐	☐	☐	''		utf8mb4_unico...
3	status	BIGINT	10	☐	☐	☐	'0'		
4	courseid	BIGINT	10	☐	☐	☐	No def...		
5	sortorder	BIGINT	10	☐	☐	☐	'0'		
6	name	VARCHAR	255	☐	☑	☐	NULL		utf8mb4_unico...
7	enrolperiod	BIGINT	10	☐	☑	☐	'0'		
8	enrolstartdate	BIGINT	10	☐	☑	☐	'0'		
9	enrolenddate	BIGINT	10	☐	☑	☐	'0'		
10	expirynotify	TINYINT	1	☐	☑	☐	'0'		
11	expirythreshold	BIGINT	10	☐	☑	☐	'0'		
12	notifyall	TINYINT	1	☐	☑	☐	'0'		
13	password	VARCHAR	50	☐	☑	☐	NULL		utf8mb4_unico...
14	cost	VARCHAR	20	☐	☑	☐	NULL		utf8mb4_unico...
15	currency	VARCHAR	3	☐	☑	☐	NULL		utf8mb4_unico...
16	roleid	BIGINT	10	☐	☑	☐	'0'		

Рис. 1.4 – Інформація про користувачів з таблиці mdl_enrol

Архітектура бази даних Moodle базується на принципах нормалізації, що дозволяє уникати дублювання, забезпечити зв'язність даних і підтримку їхньої цілісності. У Moodle зберігаються як структуровані дані (таблиці оцінок, реєстрацій, тестів), так і напівструктуровані або неструктуровані (файли завдань, повідомлення в чатах, форумах тощо). Основні категорії даних що зберігаються і LMS Moodle представлено в табл.1.1 **[9]**.

Таблиця 1.1

Основні категорії даних що зберігаються в LMS Moodle

Категорія	Опис
Персональні дані	Ім'я, прізвище, електронна пошта, інформація автентифікації, роль у системі. Зберігається переважно в таблицях mdl_user та mdl_role_assignments.
Навчальні результати	Оцінки за завдання, тести, результати проходження тестів, підсумкова оцінка за курс. Зберігається переважно в таблицях mdl_grade_items, mdl_grade_grades, mdl_quiz_attempts, mdl_question_attempts, mdl_grade_items.
Активність користувачів	Відвідуваність курсів, занять, вхід до системи, перегляд сторінки, спроба тесту, завантаження файлу. Зберігається переважно в таблицях mdl_user, mdl_quiz_attempts, mdl_question_attempts, mdl_logstore_standard_log.
Комунікація	Пряма взаємодія між здобувачами, викладачами й адміністрацією. Важливо для моніторингу залученості, обговорень, активного навчання. Зберігається переважно в таблицях mdl_forum, mdl_forum_discussions, mdl_forum_posts mdl_chat, mdl_chat_messages, mdl_chat_users

Система зберігання в Moodle реалізована з урахуванням принципів безпеки: шифрування конфіденційної інформації, обмеження прав доступу, журналювання

змін, регулярне резервне копіювання даних **[10]**. Moodle підтримує автоматичне створення резервних копій курсів, що дозволяє зберігати повну історію змін та відновлювати дані при збоях. Системи управління навчанням, зокрема Moodle, виступають не лише платформами для проведення навчання, але й потужними інструментами збору, зберігання та аналізу освітніх даних. Правильно реалізована інфраструктура зберігання даних в LMS дозволяє забезпечити безпечне та централізоване зберігання навчальної інформації, інтегрувати дані з іншими системами для створення єдиного цифрового середовища. Таким чином, зберігання та обробка даних у LMS є не лише технічним аспектом, але й фундаментом для прийняття рішень у сфері цифрової трансформації освіти.

Якість даних є критично важливою для аналітичних моделей у сфері освіти. Наявність неповних, застарілих або неконкретних записів знижує ефективність алгоритмів прогнозування, тому необхідне застосування методів попереднього очищення та нормалізації даних **[11, 12]**. Видалення інформації, що не відповідає критеріям аналітики (наприклад: записи, що давно не оновлювались; неповні профілі здобувачів; або дані, не релевантні до поточних освітніх цілей), дозволяє суттєво підвищити точність і швидкість обробки. Окрім технічних аспектів, важливу роль відіграють етичні міркування, пов’язані із забезпеченням конфіденційності особистих даних здобувачів, оскільки зростаюча увага до захисту інформації зобов’язує освітні установи та LMS рішень дотримуватись правових і етичних стандартів при зборі, зберіганні та використанні персональних освітніх даних **[13]**. Це не лише гарантує безпеку даних, але й сприяє підвищенню довіри між здобувачами освіти та навчальними платформами. Ефективне прогнозування результатів навчання базується на якісно підготовлених, релевантних і етично оброблених даних. Враховуючи великі обсяги інформації в LMS, необхідно застосовувати методи, які поєднують швидкість і простоту з високою точністю та аналітичною цінністю. За допомогою моделей

машинного навчання можна виявити певні патерни в поведінці здобувачів, які можуть вказувати на ризик низьких результатів.

1.2. Аналіз методів обробки великих даних

На основі даних, зібраних у системах управління навчанням, методи обробки великих даних відіграють важливу роль у підвищенні якості освітньої аналітики. Такі методи включають широкий спектр технік – від зменшення розмірності даних до фільтрації шумів і виявлення релевантних патернів, що дозволяють підвищити інформативність навчальних даних та спростити подальший аналіз **[14]**. Застосування зазначених методів значно покращує як точність, так і швидкість обчислень при побудові моделей прогнозування успішності здобувачів на базі машинного навчання **[15]**. Водночас ефективність попередньої обробки безпосередньо залежить від якості вхідних даних. У випадках, коли початкові дані мають високий рівень повноти, актуальності та структурованості, вплив додаткової обробки на кінцеві результати може бути мінімальним, і загальний приріст у точності чи швидкості аналітики є незначним. Таким чином, попередня обробка даних є важливим етапом у побудові аналітичних моделей в LMS, особливо коли йдеться про гетерогенні або неструктуровані освітні дані **[16, 17]**. Проте її доцільність і ефективність мають оцінюватися з урахуванням характеристик вихідного інформаційного масиву. Тому у процесі побудови моделей машинного навчання для прогнозування академічної успішності здобувачів у системах управління навчанням (LMS), вибірка даних відіграє критичну роль. Вона полягає у відборі репрезентативної підмножини з повної (генеральної) сукупності даних з метою ефективного аналізу без потреби опрацювання всього масиву інформації, а це дозволяє скоротити обчислювальні витрати та пришвидшити процес розробки аналітичних рішень. Одним з ключових аспектів вибірки є забезпечення її репрезентативності, тобто точного відображення структури генеральної сукупності **[18]**. Бо невідповідність та не збалансованість вибірки реальній структурі може призвести до упереджених висновків, що особливо

критично у прогнозних моделях, де навіть незначне викривлення вхідних даних може спричинити падіння точності **[19]**. У випадку роботи з великими обсягами освітніх даних (наприклад, у національних LMS або масових відкритих онлайн-курсах), випадковість вибірки і її обсяг істотно впливають на ймовірність відображення ключових характеристик генеральної сукупності. Ймовірність потрапляння окремого елемента до вибірки можна описати формулою **[20]**:

$$P = \frac{n}{N}, (1.1)$$

де P – ймовірність включення елемента до вибірки;

N – загальна кількість елементів (здобувачів) у наборі даних;

n – кількість елементів у вибірці.

Відсотковий коефіцієнт варіації, %CV, є одиницею вимірювання варіації, і його можна вважати «відносним стандартним відхиленням», оскільки він визначається як стандартне відхилення, поділене на середнє значення, помножене на 100 відсотків **[21]**:

$$\%CV = 100\% * \frac{\sigma}{\mu}, (1.2)$$

де $\%CV$ – відсотковий коефіцієнт варіації;

σ – стандартне відхилення (показує, наскільки дані розкидані навколо середнього значення);

μ – середнє арифметичне значення (mean) генеральної сукупності.

Ця формула справедлива для нетрансформованих даних. Розрахунок %CV математично відрізнятиметься залежно від середнього значення та дисперсії перетворення. Якщо нетрансформований %CV використовується для логарифмічних нормальних даних, отриманий %CV буде надто малим і дасть надто оптимістичний,

але неправильний погляд на вимірювання. Фактично, відсотковий коефіцієнт варіації є статистичним показником, який використовується для оцінки відносної мінливості даних і який застосовують у контексті оцінки якості даних.

Визначення середнього значення генеральної сукупності, яке показує математичне очікування всіх можливих значень у популяції можна за наступною формулою **[22]**:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \text{ , (1.3)}$$

де x_i – окремі значення в генеральній сукупності;

N – загальна кількість елементів;

μ – математичне очікування або очікуване значення.

Основні типи вибірок даних у контексті машинного навчання представлено в табл. 1.2 **[23]**.

Таблиця 1.2

Основні типи вибірок для даних освітньої аналітики

Тип вибірки	**Опис**	**Переваги**
Random	Кожен здобувач має рівну ймовірність потрапити до вибірки	Простота, репрезентативність при великій N
Stratified	Розподіл здобувачів за ознаками (наприклад, успішність, факультет)	Забезпечує рівномірне представлення підгруп
Систематична	Вибір кожного k-го елемента після випадкового старту	Структурованість, проста реалізація

Кластерна	Вибір цілих груп (кластерів), наприклад, груп або курсів	Знижує витрати при великій кількості кластерів
Oversampling	Збільшення частки менш представлених класів (наприклад, здобувачів з низькою успішністю)	Бореться з дисбалансом класів
Bootstrap	Створення вибірок із заміною для оцінки статистичних параметрів	Покращує надійність оцінки

Вибірка має бути не лише репрезентативною, а й структурно відповідати особливостям досліджуваної сукупності. Існує кілька типів вибірок, які використовуються частіше для освітньої аналітики **[24]**.

- **Stratified-**вибірка полягає у поділі даних на окремі групи за релевантними критеріями, наприклад такими як курс, спеціальність чи рівень успішності **[25]**. Такий підхід забезпечує представлення усіх значущих підгруп та підвищує точність моделі, особливо у випадках дисбалансу класів.
- **Random-**вибірка ґрунтується на принципі рівної ймовірності потрапляння кожного елементу до підмножини **[26]**. Такий підхід є простим у реалізації та ефективним у випадках, коли дані не мають суттєвих внутрішніх відмінностей або класи розподілені більш-менш рівномірно.
- **Oversampling** застосовується для збалансування класів за рахунок штучного збільшення кількості прикладів рідкісного класу, наприклад, здобувачів із низькими показниками успішності **[27]**. Це дозволяє моделі краще розпізнавати такі випадки та зменшує ризик їх ігнорування в процесі навчання.

Додатково використовується Bootstrap-вибірка, що передбачає багаторазову випадкову вибірку з поверненням **[28]**. Вона є корисною для оцінювання стабільності

моделей, перевірки надійності результатів та формування ансамблевих моделей. Таким чином, у практиці навчальної аналітики найчастіше використовують саме ці типи вибірок, оскільки вони дозволяють ефективно працювати з нерівномірно розподіленими освітніми даними, забезпечують репрезентативність і сприяють підвищенню точності прогнозування. Вибір конкретної стратегії вибірки залежить від структури даних, наявності дисбалансу між класами та поставлених аналітичних завдань.

У процесі побудови моделей прогнозування успішності здобувачів на основі даних з систем управління навчанням (LMS) важливим етапом є трансформація даних. Вона полягає у зміні формату, структури або значень даних для забезпечення їхньої придатності до алгоритмів машинного навчання, та дозволяє усунути проблеми, пов’язані з різнорідністю, масштабністю та неповнотою освітніх даних, що безпосередньо впливає на точність та стабільність прогнозних моделей **[29]**. У прогнозному аналізі трансформація відіграє критичну роль – вона дозволяє привести ознаки до однакового масштабу, замінити категоріальні значення на числові, усунути пропущені дані, виявити викиди або зменшити розмірність. Завдяки цьому модель краще навчається та ефективніше виявляє закономірності, що сигналізують про можливу низьку або високу успішність здобувача Найпоширеніші методи трансформації даних, які використовуються на етапі підготовки освітніх даних до прогнозування, наведено в табл. 1.3 **[30]**.

Таблиця 1.3

Найпоширеніші методи трансформації даних

Вид трансформації	**Опис**	**Застосування у прогнозуванні**
Нормалізація	Масштабування значень до інтервалу [0, 1].	Для ознак, що мають різні діапазони (наприклад, час активності, кількість спроб).

Стандартизація	Масштабування до середнього значення 0 і стандартного відхилення 1.	Для моделювання з алгоритмами, чутливими до масштабу (логістична регресія, SVM).
Логарифмічне перетворення	Застосування натурального логарифма до значень.	Для зменшення впливу великих значень (наприклад, кількість заходів у LMS).
Степеневе перетворення	Використання степеневих функцій для стабілізації дисперсії.	Для ознак із високою варіативністю.
One-Hot Encoding	Перетворення категорій у двійкові змінні.	Для перетворення нечислових ознак: факультет, тип завдань, тощо.
Label Encoding	Присвоєння чисел категоріям за порядком.	Для порядкових ознак (наприклад, рівень знань: базовий, середній, високий).
Binning	Розбиття числових змінних на категорії.	Для зменшення впливу викидів або групування рівнів активності.
Поліноміальні функції	Створення нових ознак через взаємодію змінних.	Для врахування складних залежностей між ознаками.
Масштабування функцій	Регулювання масштабу окремих змінних.	Для усунення переваги окремих ознак над іншими.
PCA (аналіз головних компонент)	Зменшення кількості ознак без втрати ключової інформації.	Для боротьби з високою розмірністю, наприклад, при великій кількості поведінкових ознак.

При прогнозування успішності здобувачів якісна трансформація освітніх даних є важливою для побудови точних і стабільних моделей. Вибір конкретного виду трансформації залежить від типу даних, характеристик змінних та вимог алгоритму прогнозування. Її правильне застосування дозволяє зменшити похибки моделі, уникнути перенавчання та більш точно виявити здобувачів, які перебувають у зоні ризику академічної неуспішності.

1.3. Аналіз використання відомих методів машинного навчання для прогнозування успішності

Для виконання прогнозування успішності здобувачів в основному використовуються методи машинного навчання. Найпоширенішими алгоритмами класифікації є: логістична регресія (LR), дерево рішень (DT), наївний класифікатор Баєса (NB), метод опорних векторів (SVM), випадковий ліс (RF), нейронні мережі (NN) **[31]**. Досягнення високої точності в прогнозуванні успішності є складним і багатогранним завданням через велику кількість факторів, які на неї впливають. Якість даних є одним із головних факторів і фактично основою будь-якого успішного прогнозування. Першим при створенні моделі виникає питання збору даних для її навчання. Найбільш доступним способом їх отримання в реальних умовах є аналіз інформації яку збирають LMS, зокрема, Moodle.

Тим не менш, важливо розуміти, що точність прогнозування успішності залежить не тільки від алгоритмів та методів машинного навчання, але й від кількості та ваги параметрів включених до моделі.

У роботі **[32]** наведено результати досліджень прогнозування успішності здобувачів на базі підсумкових іспитових оцінок. Тобто фактор успішності являє собою набір здобувачем 60 і більше балів. Дані були взяті зі здобувачської інформаційної системи (SIS). У цих записах оцінки за проміжні іспити, оцінки за підсумкові іспити та проміжні оцінки по курсам зібрані з 1854 здобувачів. Всього було

використано три типи параметрів: оцінки за проміжні іспити, дані оцінок з кафедри та дані оцінок з факультету. Було розраховано та порівняно точність алгоритмів випадкового лісу (RF), найближчого сусіда (KNN), опорних векторів (SVM), логістичної регресії (LR), наївного Баєса (NB) та алгоритмів k-найближчого сусіда (kNN), яка склала: 74,6 %, 74,6 %, 73,5 %, 71,7 %, 71,3 % і 69,9 % відповідно. Показано, що алгоритм випадкового лісу зміг досягти найвищої точності класифікації 74,6 % . Значення розрахованої площі під кривою (AUC) для алгоритмів RF, NN, SVM, LR, NB і kNN становило 86 %, 86,3 %, 80,4 %, 82,6 %, 81 % і 81 % відповідно. Точність прогнозу оцінювали за допомогою десятикратної перехресної перевірки. Використання великої кількості алгоритмів для визначення кращого, є гарним рішенням. Але залишилися невирішеними питання, пов'язані з обмеженістю даних, із яких наявні лише оцінки за проміжні результати та іспити. Тому при прогнозуванні успішності на ранній стадії, коли ще немає оцінок, навряд вийде досягти значної точності.

У роботі **[33]** наведено результати досліджень прогнозування успішності здобувачів, виходячи з відвідуваності занять. В якості даних для тренування та навчання моделі було використано дані про оцінки та відвідуваність на лабораторних роботах, лекційних та практичних заняттях, які були взяті з LMS Moodle. Було розраховано та порівняно точність алгоритмів випадкового лісу (RF), опорних векторів (SVM) та логістичної регресії (LR), яка склала: 80 %, 79 % та 79 % відповідно. Значення розрахованої площі під кривою (AUC) для алгоритмів RF, SVM та LR становило 73 %, 66 % і 70 % відповідно. Показано, що алгоритм випадкового лісу зміг досягти найвищої точності класифікації 73 %. Відвідуваність є показником який прямо впливає на успішність, але не може бути єдиним достатнім фактором. Тому залишилися невирішеними питання, пов'язані з обмеженістю даних, оскільки навіть при 100 % відвідуваності здобувач може мати низькі бали і не успішно здати сесію. Варіантом подолання відповідних труднощів може бути додавання інших видів даних про роботу здобувачів.

Саме такий підхід використаний у роботі **[34]**, де в якості даних для тренування та навчання моделі було використано: оцінки за лекції, тести та лабораторні роботи і переглянуті відео. Для прогнозування успішності був використаний лише алгоритм випадкового лісу для побудови моделі. Показано, що отримана модель змогла передбачити неуспішність з точністю 96,3 %. В роботі використали 3-кратну перехресну перевірку, повторену 5 разів, а вже потім побудували модель випадкового лісу з відцентрованими та масштабованими даними. Відсутність побудованої ROC-кривої та розрахунку площі під кривою (AUC) не дає наочного розуміння загальної ефективності отриманої моделі. Залишилися невирішеними питання, пов'язані з параметром: кількості переглянутих відео, а саме як він розраховується. Чи дійсно є перевірка, що відео переглянуте до кінця, а не просто відкрите здобувачем. Модель показує високу точність, але не зрозуміло який буде результат прогнозування на ранній стадії. Також не вистачає розрахунків якості та ефективності моделі і порівняння точності її прогнозування з іншими алгоритмами машинного навчання.

У роботі **[35]** прогнозування успішності проводилося з наступними алгоритмами: випадковий ліс (RF), логістична регресія (LR), нейронні мережі (NN), градієнто розширені дерева рішень (XGBoost). Показано, що точність прогнозування склала: 90 %, 90 %, 87 % та 84 % відповідно. Порівняння показників оцінювання показує кращу продуктивність для алгоритмів нейронних мереж та дерева посилення градієнтного спуску порівняно з логістичною регресією та випадковим лісом. Гіпотеза дослідження полягає в тому, що успіх або неуспіх у навчанні можна передбачити, використовуючи дані активності здобувачів з журналів LMS Moodle. В якості даних для навчання та тренування моделі було використано: стать, кількість завантажених файлів, переглянуті файли, виконані модулі, кількість відвідувань платформи, активність за день та інші дані активності з журналів активності в LMS. Результати дослідження підтверджують гіпотезу про те, що успішність у навчанні можна передбачити за допомогою алгоритмів машинного навчання на даних, отриманих під час взаємодії здобувачів із платформами електронного навчання. Залишилися

невирішеними питання, пов'язані з тим що звіти про активність користувачів не мають даних про фактичну взаємодію користувача з платформамою електронного навчання. А тільки сам факт переходу на активність чи відкриття, наприклад перегляд лекції чи вправи. Варіантом подолання труднощів може бути додавання нових даних та перевірка результатів точності прогнозування на більшій кількості відповідних зразків.

У роботі **[36]** для прогнозування успішності використали модель, засновану на багаторівневій нейронній мережі Персептрона, яку було навчено прогнозувати успішність здобувачів у середовищі змішаного курсу навчання. Передбачуваний успіх здобувача заснований на чотирьох навчальних видах діяльності: спілкування електронною поштою, спільне створення контенту за допомогою wiki, взаємодія контенту, виміряна за переглядами файлів, і самооцінка за допомогою онлайн-тестів. Показано, що модель передбачила успішність здобувачів з рівнем правильної класифікації (CCR) з точністю 98,3 %. Отриману точність також підтверджує побудована ROC-крива, значення розрахованої площі під кривою (AUC) складає 98,9 %. Для комплексної оцінки та порівняння результатів прогнозування доцільно додати інші алгоритми класифікації. Необхідна перевірка отриманих результатів точності прогнозування на більшій кількості даних та інших алгоритмах, оскільки точність в такому випадку може змінитися.

У роботі **[37]** наведено результати досліджень прогнозування успішності здобувача, використовуючи академічні дані та записи із системи керування навчанням, які корелюють з успіхом або невдачею в проходженні курсу. Було використано шість алгоритмів (GBT, RF, DT, LR, NB, SVM) з навчанням моделей на трьох різних етапах дворічного курсу. Протестовано моделі на записах 394 здобувачів з 3 курсів. Показано, що випадковий ліс дав найкращі результати з 84,47 % за балом F1 у експериментах, за яким слідує Дерево рішень, яке отримало подібні результати. На відміну від попередніх досліджень, у цьому враховувалися дані з 3 курсів, що сприяє більш точному прогнозуванню моделі, здобувачів яким загрожуєнезадовільна

оцінка. Але залишилися невирішеним питання, пов'язане з достатністю використаних даних до точного прогнозування успішності на ранніх стадіях навчання.

У роботі **[38]** наведено результати досліджень розробки точного прогнозування результатів курсу здобувачів, чи вони його пройдуть чи взагалі не складуть. На відміну від попередніх досліджень, у цьому дослідженні враховувалися демографічні дані, оцінювання та дані про взаємодію учнів, щоб надати комплексні прогнози. Для розробки моделей прогнозування використано логістичну регресію та випадковий ліс. Точність моделей оцінювали на основі класифікації за чотирма класами (прогнозування чотирьох можливих результатів) і класифікації за двома класами (прогнозування проходження чи невдачі). Показано, що прості показники, такі як рівень активності здобувача в певний день, можуть бути такими ж ефективними, як і більш складні комбінації даних або особиста інформація для прогнозування успішності учня. Модель логістичної регресії досягла точності 72,1 % для класифікації за чотирма класами та 92,4 % для класифікації за 2 класами. Тоді як класифікатор випадкового лісу досяг точності 74,6 % для класифікації за чотирма класами та 95,7 % для класифікації за двома класами. Такий підхід для прогнозування дає розуміння результатів здобувачів курсу, пропонуючи цінну інформацію для покращення залучення здобувачів в навчальних онлайн середовищах.

У дослідженні **[39]** використовувався метод пакетного ансамблю з вісьмома техніками машинного навчання, включаючи KNN, RF, SVM, LR і NB, і три окремі топології ANN, щоб визначити здобувачів групи ризику, які не пройдуть курс. У дослідженні використовувалися два набори даних з двох різних курсів, щоб класифікувати здобувачів на одну з трьох категорій: хороші, задовільні або слабкі. Перший набір даних містив оцінки для групи з 52 здобувачів інженерних спеціальностей за один курс із платформи електронного навчання, а другий набір даних містив різні оцінки завдань для 486 здобувачів природничих наук, включаючи завдання, тести та іспити з платформи електронного навчання. Результати показали, що модель пакетування має найкращу продуктивність. Для першого набору даних

точність досягла 66,7%, тоді як модель пакетування другого набору даних досягла точності 88,2%.

У дослідженні **[40]** використовувалися оцінки за перші два тижні формувального оцінювання (вправи та домашнє завдання), щоб виявити здобувачів, які ризикували провалити випускний іспит початкового курсу програмування. Техніка ансамблю була використана для розгортання моделі прогнозування, яка використовувалася для ідентифікації здобувачів, які ризикують провалити іспит чи ні. Набір даних містив два предиктори для 289 здобувачів, які навчаються на курсі програмування, та використовувався для навчання та тестування запропонованої моделі. Запропонована модель класифікації досягнула 72,73% точності навчання та 59,64% точності тестування.

У дослідженні **[41]** класифікували успішність здобувачів наприкінці першого навчального року, щоб дізнатися про вплив критеріїв прийому на успішність здобувачів. Було використано набір даних 1445 здобувачів бакалаврату. У експерименті застосовувалися різні методики, включаючи RF, ансамбль дерев, DT, NB, LR і MLP. Алгоритм LR отримав найвищу точність 50,23% з платформою KNIME, а ANN отримав найвищу точність 51,9%. Крім того, результат показав, що існує слабкий зв’язок між критеріями вступу та академічною успішністю здобувача.

У дослідженні **[42]** проводили пошук найкращого способу прогнозування успішності здобувача шляхом застосування різних методів EDM на реальних даних 104 здобувачів, які були зібрані з інформаційної системи Universitas Islam Indonesia. Алгоритми класифікації Баєса (BN) і дерев рішень (DT), а також п’ять методів відбору ознак були використані для прогнозування здобувачів, які, швидше за все, покинуть навчання. У результаті застосування вибору ознак вони виявили, що точність моделі прогнозування була підвищена. Крім того, характеристики які включають відвідуваність і середній бал за перший семестр, мають найбільший вплив на успішність здобувача порівняно з іншими характеристиками, такими як особиста

інформація, інформація про сім'ю та характеристики підготовки до університету. За допомогою алгоритму Баєса була досягнута найвища точність 98,08%.

В проаналізованих дослідженнях використовують різні алгоритми машинного навчання та будують моделі які мають високу розраховану точність прогнозування успішності. Проте більшість використовує лише дані з журналів LMS Moodle та значення балів (оцінок по дисциплінам, тестам і модулям) отриманих за попередні навчальні періоди. Це дозволяє стверджувати, що доцільним є проведення дослідження впливу розширеного набору даних про роботу здобувачів з відеоматеріалами на точність моделей прогнозування успішності. Оскільки такі дані дозволять враховувати взаємодію з кожним відео для кожного здобувача. У свою чергу розширення набору даних для навчання пов'язане з додатковими витратами ресурсів на їх збір та обробку. І при цьому збільшується тривалість процесу навчання моделей. Результати такого дослідження дадуть можливість приймати обґрунтовані рішення щодо вибору наборів даних для тренування моделей машинного навчання.

1.4. Методи оцінювання і прогнозування успішності

Академічна успішність здобувачів у вищій освіті широко досліджується для вирішення проблем із недостатньою успішністю, підвищенням рівня відсіву з університетів та проблемами виконання навчальних планів **[43]**. Успішність здобувача означає ступінь досягнення короткострокових і довгострокових цілей в освіті **[44]**. Вимірювання успішності здобувачів відбувається з різних точок зору, починаючи від підсумкових оцінок здобувачів, середнього бала (GPA), відвідуваності, залученості в навчальний процес і закінчуючи майбутніми перспективами працевлаштування **[45]**. Прогнозування успішності дає змогу виявляти здобувачів із низькою успішністю, таким чином викладачам можуть допомагати та вирішувати потенційні проблеми на ранніх етапах навчального процесу. Це може бути консультування та опитування здобувачів, подальший моніторинг успішності, розбір навчальних матеріалів, впровадження додаткових інтелектуальних систем навчання

[46]. Проведене комплексне опитування показало, що приблизно 70% розглянутих робіт досліджували прогнозування успішності здобувачів за допомогою оцінок здобувачів і середніх балів, тоді як лише 10% досліджень перевіряли прогнозування досягнень здобувачів за допомогою результатів навчання **[47]**. Попередні огляди показали, що кумулятивний середній бал і оцінювання курсу є найбільш використовуваними предикторами успішності та успішності здобувачів **[48, 49]**. Кілька досліджень навіть використовували оцінки за наступний семестр як основний показник успішності здобувача **[50, 51]**. Але загальна академічна успішність здобувача не може в повній мірі оцінюватися лише оцінками. Найбільш часто використовуваними ознаками для прогнозування академічних досягнень у вищій освіті, як це зустрічається в досліджуваній літературі є: стать та середній бал, вони використовуються у більш ніж половині досліджень. За ними йде вік (40%) і знання мови (31%). Інші характеристики, такі як дохід, національність, сімейний стан, статус зайнятості та відвідуваність, використовуються менш ніж у 30% публікацій. Це питання слід вивчати в ширшому контексті, зокрема з використанням поточних результатів здобувачів та їх взаємодії з навчальним процесом. Недавнє дослідження рекомендує вивчити перспективу прогнозування досягнення результатів здобувачів, щоб зробити висновок про успішність здобувачів **[52]**. Загальний процес аналізу даних, який починається з попередньої обробки цільового набору даних, а потім виконання прогнозування представлено на рис. 1.5.

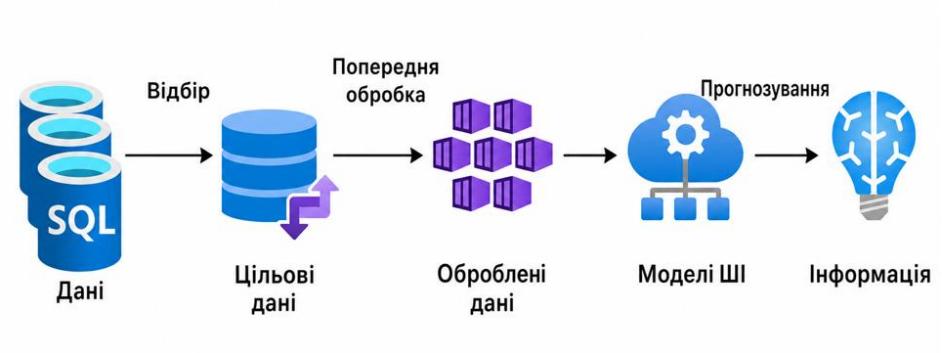


Рис. 1.5 – Процес аналізу даних та прогнозування

В цілому, методологія прогнозування успішності складається з наступних етапів **[53]**:

1. Збір освітніх даних.
2. Підготовка даних (попередня обробка, очищення, скорочення, перетворення).
3. Побудова моделі прогнозування (навчання, тренування, підбір класифікаторів).
4. Оцінка моделі класифікації (значень характеризуючих точність та якість).
5. Обробка та застосування отриманої інформації.

Інтелектуальні методи, які використовуються в аналітиці навчання для прогнозування досягнень учнів, як правило, класифікуються на контрольоване навчання, неконтрольоване навчання, інтелектуальний аналіз даних і статистичні підходи **[54]**. Кожна категорія містить велику кількість інтелектуальних алгоритмів, таких як штучні нейронні мережі, машина опорних векторів, K-найближчий сусід і випадковий ліс. Атрибути, які передбачають успішність здобувачів, широко досліджуються в літературі, що призводить до поєднання академічних (бали перед вступом і вступні кваліфікації) і неакадемічних факторів (емоційний інтелект і стійкість) **[55, 56, 57]**. Проте питання вибору найоптимальніших факторів, які впливають на досягнення результатів курсу та навчальної програми залишається відкритим. Більшість даних про навчання здобувачів отримується з систем управління навчанням, як правило, це загальні для всіх дані про оцінки, відвідування, активність здобувача, виконання робіт. Але для підвищення точності прогнозування моделей необхідно збільшувати кількість дани та їх різноманітність. Для системи управління навчанням Moodle, якою користується більшість навчальних закладів у світі, можливістю отримання більшої кількості даних є плагіни. Саме вони дозволяють створювати нові таблиці в базі даних та накопичувати додаткові інформацію, яку потім можна використати для прогнозування. Отримання нових та актуальних навчальних даних є одним із головних питань в контексті прогнозування успішності. При чому більшість попередніх досліджень страждали від проблем пов’язаних із

невеликим розміром набору даних для побудови моделі, недостатньою кількістю ознак та незбалансованістю самих даних.

1.5. Постановка задачі прогнозування академічної успішності здобувачів освіти

Системи управління навчанням стали одним із основних компонентів освітнього процесу в навчальних закладах. Їх використання дозволяє ефективно контролювати та автоматизувати процеси: адміністрування, управління контентом, навчальними курсами, спілкування з користувачами, створення окремих навчальних програм та інші. Багато навчальних закладів переходять на системи для управління навчанням і активно використовують їх у навчальних процесах. У дослідженнях інших авторів показано, що в начальному процесі у здобувачів виникають різні освітні ризики, передбачивши які, не рідко за допомогою даних із самих же LMS, можна значно покращити рівень навчальних здобутків здобувачів. Окільки якісний підхід до проведення навчального процесу, підтримка та допомога здобувачам є фундаментом успішного навчання. Тому це важливо і для аналізу навчального контенту, його покращення та вдосконалення. Водночас важливим напрямом розвитку сучасних освітніх технологій є аналіз навчальної поведінки здобувачів освіти та виявлення закономірностей, які впливають на їх академічну успішність. Особливого значення набуває дослідження взаємодії студентів із навчальним контентом, зокрема відеоматеріалами, оскільки такі дані дозволяють більш глибоко оцінити рівень залученості до навчального процесу та виявити потенційні ризики зниження академічної успішності. У цьому контексті актуальною є задача побудови моделей прогнозування академічної успішності студентів на основі інтеграції традиційних освітніх показників (відвідуваність занять, академічні оцінки) та поведінкових характеристик взаємодії з навчальним контентом. Особливого значення набувають ранні прогнози навчальної успішності, які дозволяють ідентифікувати студентів, що належать до групи ризику недосягнення порогового рівня академічних результатів. Це, у свою чергу, створює можливості для своєчасного педагогічного втручання,

надання індивідуальної підтримки, консультування та формування рекомендацій щодо підвищення ефективності навчання, що позитивно впливає на загальні результати освітнього процесу. Таким чином, виникає необхідність у дослідженні методів машинного навчання для прогнозування академічної успішності студентів на основі багатовимірних освітніх даних, а також у аналізі впливу поведінкових характеристик взаємодії з навчальним контентом на точність таких прогнозів.

1.6. Висновки до розділу 1

1) Проведено комплексний аналіз структури, особливостей накопичення та організації даних у системах управління навчанням, що дозволило визначити ключові джерела формування освітньої аналітики в цифровому навчальному середовищі.
2) Досліджено архітектуру бази даних LMS Moodle та встановлено групи даних, які характеризують навчальну активність користувачів, що створює основу для подальшого формування структурованого ознакового простору для задач машинного навчання.
3) Проаналізовано методи обробки великих даних, які використовуються в задачах освітньої аналітики, що дозволило визначити їх роль у підвищенні якості інтерпретації навчальних процесів та подальшій побудові моделей прогнозування успішності.
4) Досліджено підходи до попередньої обробки даних та виявлення інформативних патернів у навчальних наборах даних, що забезпечує підвищення якості ознакового представлення даних.
5) Проведено систематизацію сучасних наукових досліджень у сфері прогнозування академічної успішності з використанням методів машинного навчання, що дозволило визначити основні типи використовуваних даних та підходи до побудови прогностичних моделей.

6) Виконано аналіз результатів застосування моделей машинного навчання в задачах прогнозування успішності, що підтвердило ефективність використання даних підходів для задач класифікації.
7) Узагальнено підходи до побудови моделей прогнозування успішності, що включають етапи збору, очищення, трансформації даних та формування ознак.

РОЗДІЛ 2. МЕТОДИ ТА АЛГОРИТМИ МАШИННОГО НАВЧАННЯ

У другому розділі розглянуто методи і алгоритми машинного навчання та виконано їх порівняльний аналіз, а саме: логістичної регресії (Logistic Regression), класифікатора наївного Баєса (Naive Bayes), методу опорних векторів (Support Vector Machines), випадкового лісу (Random Forest) та нейронних мереж (MLPClassifier). Було розглянуто також недоліки та переваги їх використання на практиці. Визначено найбільш точні моделі по прогнозуванню успішності здобувачів на основі оцінок та відвідуваності, стратегію оцінювання ефективності моделей і виділення їх ознак. Результати розділу опубліковано у наукових працях **[58, 59, 60]**.

2.1. Моделі та алгоритми машинного навчання

Машинне навчання являє собою процес навчання комп'ютерної програми або алгоритму, спрямований на поступове поліпшення ефективності виконання певного завдання **[61]**. З наукової точки зору, машинне навчання можна аналізувати через призму теоретичного та математичного моделювання механізмів його функціонування. Однак на практиці цей процес включає розробку додатків та плагінів, що демонструють ітеративне покращення результатів. Класифікація машинного навчання може базуватися на різних підходах, але серед основних виділяють три: навчання з підкріпленням, навчання з учителем та навчання без учителя **[62]**.

Математична модель є формальним описом системи, що використовує математичні поняття і мову для представлення її основних властивостей та взаємозв’язків **[63]**. Процес створення такої моделі називається математичним моделюванням. Зазвичай, математичні моделі складаються з множини змінних, що

описують елементи системи, та відносин між ними, які визначаються через рівняння або функції, а змінні можуть бути різного типу. Незважаючи на те, що змінні відображають певні характеристики системи, сама модель є системою функцій, що описують взаємозв'язки між цими змінними. Відношення між змінними можуть бути виражені через алгебраїчні операції, диференціальні рівняння або інші математичні інструменти. Для класифікації математичних моделей за структурою використовують різні підходи, одним із них є поділ на статичні та динамічні моделі. Статичні моделі не враховують зміни системи з часом і припускають, що всі взаємозв'язки постійні. Динамічні ж моделі, навпаки, безпосередньо враховують часову змінність і відображають процеси, що змінюються в часі. Крім того, моделі можна класифікувати на детерміністичні та стохастичні. У детерміністичних моделях всі взаємозв'язки між змінними є чітко визначеними, що дозволяє передбачити результат з високою точністю. В стохастичних моделях хоча б одна змінна є випадковою, що вносить елемент невизначеності та ймовірнісні залежності в модель. В контексті машинного навчання статистична модель є математичним представленням, що базується на наборі статистичних припущень, які визначають генерацію вибіркових даних **[64]**. Вона описує процес генерації даних в ідеальних умовах і задається через математичні зв'язки між однією або кількома випадковими змінними та іншими невипадковими змінними. Зазвичай, статистичну модель представляють як пару (S, P), де S позначає множину можливих спостережень (вибірку), а P – множина ймовірнісних розподілів, що визначають ймовірність для кожного елемента з S. У цій моделі розподіл ймовірності, позначений як P, вибирається таким чином, щоб набір даних, отриманий у результаті генерації спостережень, наближався до істинного розподілу ймовірності. Водночас, модель не обов'язково повинна містити сам істинний розподіл ймовірностей у своєму наборі значень. Множина, як правило, є параметризованою: $P = \{P_\theta : \theta \in \Theta\}$ і визначається як P_θ для параметра θ, який належить множині параметрів Θ, що визначають модель. Параметризація, тобто набір можливих значень

параметрів, має властивість ін'єкційності, що означає: якщо $P\theta 1 = P\theta 2$, то $\theta 1 = \theta 2$. Математично статистичну модель можна представити так **[65]**:

$$M = (S, P) = (S, \{P_\theta : \theta \in \Theta\}), \qquad (2.1)$$

де:

S – множина можливих спостережень,

$P = \{P_\theta : \theta \in \Theta\}$ – множина ймовірнісних розподілів, параметризованих параметром θ, де θ належить параметричній множині Θ .

Множина Θ містить параметри, які визначають розподіли ймовірностей у статистичній моделі. Це гарантує ідентифікованість моделі, тобто можливість однозначно визначити параметри моделі на основі даних. У випадку, коли статистична модель має параметризацію, яку можна описати через множину Θ з кінцевим розміром, модель називається параметричною. Якщо ж множина параметрів Θ нескінченна, така модель є непараметричною.

Алгоритм машинного навчання – це набір математичних та статистичних процедур, які виконуються над даними з метою побудови моделі, здатної навчатися та робити прогнози або приймати рішення без явного програмування для кожного конкретного випадку **[66]**. Алгоритми машинного навчання можна розглядати як процес навчання цільової функції f, яка оптимально зв'язує вхідні змінні X з вихідною змінною Y **[67]**. Математично цільову функцію можна представити так:

$$Y = f(X), \quad (2.2)$$

де

X – вхідні змінні;

Y – вихідна змінна;

f – функція, яка відображає відношення між X і Y.

Оскільки точна форма функції f невідома, для її визначення використовуються різні алгоритми, які намагаються знайти таку функцію, що мінімізує відмінність між передбаченими значеннями Y і реальними спостереженнями в даних. Це досягається за допомогою методів навчання, таких як контрольоване або неконтрольоване навчання, залежно від наявності або відсутності міток у даних. У контексті мов програмування алгоритм визначається як набір інструкцій, які комп'ютер може виконати для розв'язання конкретних задач. Ефективні алгоритми здатні виконувати задачі з мінімальними витратами часу та ресурсів, що є критичним для високопродуктивних програмних систем.

2.2. Моделі класифікації

Класифікація є одним із напрямків машинного навчання, що присвячений вирішенню задачі поділу об'єктів (ситуацій) на класи за певними ознаками. У цьому контексті задана кінцева множина об'єктів, для яких відома їх належність до конкретних класів, що складає навчальну вибірку. Для інших об'єктів класова приналежність невідома, і необхідно розробити алгоритм, здатний здійснювати класифікацію будь-якого об'єкта з початкової множини. Класифікація об'єкта полягає в тому, щоб визначити номер або назву класу, до якого він належить. Результатом застосування алгоритму класифікації є вказівка на номер або найменування класу, до якого віднесено конкретний об'єкт. Серед алгоритмів класифікації, які користуються популярністю та використовуються в прикладних задачах можна виділити **[68]**:

- регресію (лінійну, логістичн);

- ймовірнісні класифікатори (Наївний баєсівський класифікатор);

- ансамблеві методи (випадковий ліс);

- нейронні мережі (MLC класифікатор).

Класифікація в машинному навчанні – це процес розподілу даних на визначені категорії або класи на основі певних характеристик або властивостей вхідних даних **[69]**. Основна мета класифікації – створення моделі, яка точно передбачає клас для нових, раніше невідомих даних, використовуючи навчальні вибірки. Існують різні типи класифікаційних задач в залежності від кількості можливих класів та природи вихідних даних. Одним із таких типів є бінарна класифікація, де дані можуть належати лише до двох класів **[70]**. Наприклад, успішна та не успішна здача навчальних дисциплін здобувачем на сесії. Іншим типом є мультикласова класифікація, де результат може бути одним з кількох можливих класів **[71]**. Наприклад, оцінка за дисципліну: відмінно, добре, задовільно, незадовільно. Також існує мульти-міткова класифікація, при якій вихідна змінна може належати до кількох класів одночасно **[72]**. Наприклад, здобувач може бути «відмінником» та мати оцінки «добре» за дисципліни. З точки зору часу навчання, класифікацію можна розділити на два основні типи.

- Перший тип – це **класифікація з пізнім навчанням**, де модель зберігає всю інформацію для тренування і використовує її для подальших прогнозів. Це дозволяє зменшити час навчання, але збільшує час на передбачення.
- Другий тип – це класифікація з раннім навчанням, коли модель обробляє дані під час їх отримання, що дозволяє швидше робити прогнози, але вимагає більше часу для навчання. Прикладом такого методу є дерево рішень чи штучні нейронні мережі**.**

Класифікаційні моделі базуються на спостережуваних даних та використовують різні підходи для побудови прогнозів. Результати класифікації часто є категоріями, які прив'язуються до кожного об'єкта в наборі даних, як це відбувається при

визначенні успішності здобувача (наприклад, категорії оцінок "відмінно", "добре", "задовільно", "незадовільно"). Існує два основні підходи до навчання в машинному навчанні: навчання з учителем та навчання без учителя. У методі навчання з учителем використовуються мічені дані для тренування моделі, що дозволяє побудувати алгоритм, який здатний точно класифікувати нові об'єкти. Класичним прикладом є задача класифікації успішності здобувача, де на основі історії успіхів здобувача визначається його майбутня успішність. Протилежним є метод без учителя**,** при якому не використовується мітка, і модель шукає структури або групи в даних, як це відбувається в кластеризації, де дані групуються на основі схожості. Для вирішення класифікаційних завдань використовуються: логістична регресія**,** наївний баєсів класифікатор**,** випадковий ліс**,** метод опорних векторів**,** нейронні мережі та інші **[73]**. Логістична регресія дозволяє оцінити ймовірність належності об'єкта до певного класу. Наївний баєсів класифікатор обчислює ймовірність належності до кожного класу на основі кожної ознаки окремо, а потім комбінує ці ймовірності для визначення остаточного класу. Це простий, але ефективний метод, особливо коли дані мають умовну незалежність між ознаками. Випадковий ліс є ансамблевим методом, що складається з багатьох дерев рішень, що знижує ймовірність помилок. Нейронні мережі здатні моделювати складні нелінійні залежності між ознаками та класами. Під час навчання мережі використовують алгоритм зворотного поширення помилки для коригування вагових коефіцієнтів нейронів, що дозволяє зменшити похибку в передбаченнях і покращити точність класифікації. Всі ці методи мають широке застосування при оцінці успішності здобувачів **[74]**. Де на основі попередніх результатів і характеристик здобувачів (такі як оцінки, відвідуваність, активність в модульному середовищі, робота з навчальними матеріалами) система може передбачити, чи здасть здобувач успішно сесію, або якому класу він належить: "високий рівень", "середній рівень" або "низький рівень" **[75]**.

2.3. Способи оцінки достовірності прогнозування за допомогою моделей машинного навчання

Основними критеріями ефективності та якості моделі були обрані показники: загальна точність, точність, збалансована точність, чутливість, специфічність, F1 Score, площа під кривою (AUC) та ROC-крива. Ці показники розраховуються на основі матриці помилок **[76]**. Загальна точність показує, який відсоток прикладів був правильно класифікований. Вона відноситься до частки правильних прогнозів, які включають істинно позитивні та істинно негативні. Вираз для визначення загальної точності можна записати у вигляді наступної формули:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN), \quad (2.3)$$

де TP (true positives) – кількість правильно передбачених позитивних класів;

TN (true negatives) – кількість правильно передбачених негативних класів;

FP (false positives) – кількість неправильно передбачених позитивних класів;

FN (false negatives) – кількість неправильно передбачених негативних класів.

Чутливість дозволяє визначити можливість моделі виявляти позитивні випадки. Вираз для визначення чутливості можна записати у вигляді наступної формули:

$$Sensitivity = TP/(TP + FN), \quad (2.4)$$

де TP (true positives) – кількість правильно передбачених позитивних класів;

FN (false negatives) – кількість неправильно передбачених негативних класів.

Специфічність дозволяє визначити можливість моделі виявляти негативні випадки. Вираз для визначення специфічності можна записати у вигляді наступної формули:

$$Specificity = TN/(TN + FP), \quad (2.5)$$

де TN (true negatives) – кількість правильно передбачених негативних класів;

FP (false positives) – кількість неправильно передбачених позитивних класів.

Точність визначає, яка частка передбачених позитивних результатів є дійсно позитивними. Вона показує, наскільки модель точна при прогнозуванні позитивних класів. Вираз для визначення точності можна записати у вигляді наступної формули:

$$\Pr ecision = TP/(TP + FP), \quad (2.6)$$

де TP (true positives) – кількість правильно передбачених позитивних класів;

FP (false positives) – кількість неправильно передбачених позитивних класів.

F1 Score дозволяє оцінити модель у випадках коли важливо одночасно мінімізувати хибні позитивні та хибні негативні прогнози. Вираз для визначення можна записати у вигляді наступної формули:

$$F1Score = (2 * \Pr ecision * Sensitivity)/(\Pr ecision + Sensitivity), \quad (2.7)$$

Збалансована точність дозволяє отримати оцінку загальної ефективності моделі бінарного класифікатора, враховуючи баланс між класами даних. Вираз для визначення збалансованої точності можна записати у вигляді наступної формули:

$$BalancedAccuracy = (Sensitivity + Specificity)/2, \quad (2.8)$$

Для оцінки загальної ефективності моделі незалежно від вибору порогового значення було використано параметр площі під кривою (AUC) **[77]**. Він обчислюється як площа під ROC-кривою, і приймає значення в діапазоні від 0 до 1. Чим більше значення до 1, тим краща якість моделі класифікації. Вираз для визначення AUC має наступний вигляд:

$$AUC = \sum_{n=1}^{\infty} (TPR(i+1) - TPR(i)) * (FPR(i) + FPR(i+1))/2, \quad (2.9)$$

де TPR(i) – чутливість (True Positive Rate) для і-го порогового значення;

FPR(i) – специфічність (1 – False Positive Rate) для і-го порогового значення.

Завдяки обчисленню площі під кривою, можна зрозуміти міру її «хорошості», чим далі крива від діагональної лінії, тим вона краща.

2.4. Побудова моделей прогнозування успішності здобувачів освіти на основі відомих алгоритмів

Для перевірки точності прогнозування моделей з обраними алгоритмами було виконання прогнозування успішності здобувачів відносно їх відвідуваності та оцінок. Дослідження проводилось на основі інформації з бази даних системи управління навчанням Moodle та електронного журналу університету. Дані про оцінки та відвідування здобувачів університету за перший та другий семестр 2021-2022 навчального року були експортовані у csv формат. Загальна кількість оброблених записів по оцінкам з дисциплін та відвідуваності здобувачів, експортованих у файли склала 68,670. Структура таблиці з оцінками по дисциплінам представлено у табл. 3, а структура таблиці з даними відвідуваності представлено у табл. 2.1.

Таблиця 2.1

Таблиця з оцінками по дисциплінам експортована з електронного журналу

id_student	date	id_sem	value	name	who_write	student_name	discipline_name	group_name
…	…	…	…	…	…	…	…	…
23163	22.02.2022	2	5	Звіт	522	#####	Алгоритми даних	#####
23171	23.02.2022	2	5	Звіт	522	#####	Алгоритми даних	#####
…	…	…	…	…	…	…	…	…

де id_student – унікальний ідентифікатор користувача;

date – дата запису даних;

id_sem – позначення семестру (1 або 2);

value – значення оцінки отриманої здобувачем за виконаний вид роботи;

name – назва виду роботи, яку виконував здобувач;

who_write – ідентифікатор викладача;

student_name – ім’я та прізвище користувача;

discipline_name – назва дисципліни;

group_name – назва групи.

Таблиця 2.2

Таблиця з даними про відвідуваність експортована з електронного журналу

id_student	date_write	id_sem	value	who_write	student_name	discipline_name	group_name	para_name
…	…	…	…	…	…	…	…	…
23163	22.02.2022	2	0	582	#####	Алгоритми даних	#####	Практ. заняття
23171	23.02.2022	2	1	582	#####	Алгоритми даних	#####	Практ. заняття
…	…	…	…	…	…	…	…	…

де id_student – унікальний ідентифікатор користувача;

date_write – дата запису даних;

id_sem – позначення семестру (1 або 2);

value – значення присутності (1-так, 0-ні);

who_write – ідентифікатор викладача;

student_name – ім’я та прізвище користувача;

discipline_name – назва дисципліни;

group_name – назва групи;

para_name – назва виду заняття (лекція, лабораторна, семінар).

Після цього дані по відвідуваності та оцінкам були розділені по семестрам. Для кожного здобувача по унікальному ідентифікатору «id_student» було виконано прив'язку оцінок та відвідуваності по кожній дисципліні. Для обробки даних з файлів було написано додаток з UI-інтерфейсом на WindowsForms, загальний вигляд представлено на рис. 2.1.

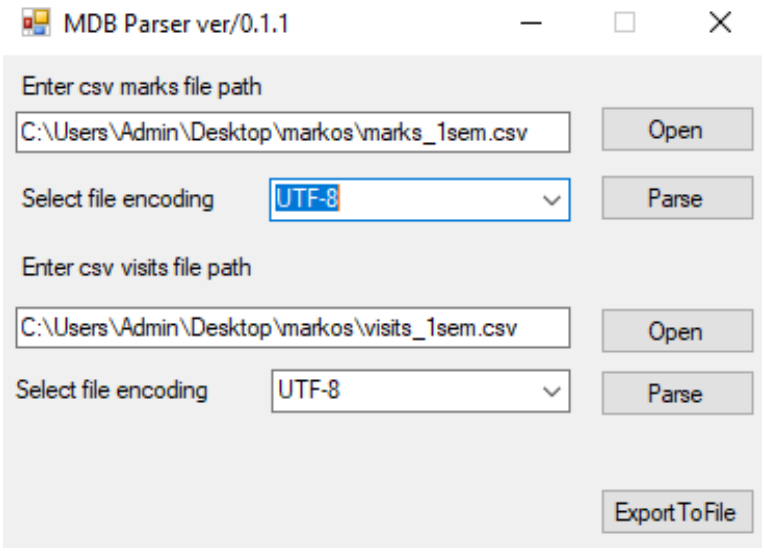


Рис. 2.1 – Додаток для обробки даних із таблиць

Додаток написаний на мові програмування C# у середовищі розробки Microsoft Visual Studio. Для обробки колонок таблиць використовувалася бібліотека CsvHelper.

Результуюча оцінка за кожен предмет вираховувалася шляхом сумування усіх отриманих балів за кожен вид виконаної роботи здобувачем. Перед виконанням навчання моделі вихідні дані були розділені на тренувальну та тестову вибірки для того, щоб перевірити, наскільки добре модель, навчена на тренувальній вибірці, може передбачати класи нових даних. Обсяг даних взятих для обробки складав 68670

вибірок користувачів, які були розподілені у відношенні 10225/58445. З яких тренувальна вибірка містила – 58445, а тестова – 10225. Ділення даних на тренувальну та тестову вибірки допомагає уникнути перенавчання (overfitting) моделі **[78]**. Оцінка та перевірка якості моделей здійснювалась на основі тестової вибірки. Моделі були побудовані за наступними ознаками (features): загальна відвідуваність, відвідуваність на лекціях, лабораторних та практичних заняттях і семінарах по кожній дисципліні. Оцінки в балах за кожен вид заняття також були враховані. Створення моделей для прогнозування виконано на мові програмування Python **[79]** із використанням бібліотеки Scikit-learn **[80]** в середовищі розробки PyCharm **[81]**. Основними критеріями ефективності моделі були обрані показники: точність, заблансована точність, чутливість, специфічність, AUC та ROC-крива. Ці показники розраховуються на основі так званої матриці помилок (confusion matrix) **[82]**.

2.4.1. Логістична регресія

Логістична регресія (Logistic Regression) – це статистичний метод машинного навчання, який використовується для моделювання й передбачення настання певних подій, зокрема для завдань класифікації, коли залежна змінна є бінарною (тобто приймає лише два значення) **[83]**. Вона широко застосовується в задачах класифікації, де необхідно передбачити належність об'єкта до одного з двох класів на основі набору незалежних змінних. І дозволяє оцінювати ймовірність належності до одного з класів на основі вхідних даних. В контексті прогнозування успішності здобувачів логістична регресія може бути застосована для прогнозування того, чи буде здобувач успішним у навчанні, на основі різних факторів, таких як оцінки, відвідуваність, результати виконання тестів, модульних контролів. Логістична регресія базується на логістичній функції (сигмоїді), яка трансформує лінійну комбінацію вхідних ознак у значення ймовірності, що варіюється від 0 до 1. Завдяки цій функції моделі логістичної регресії можуть передбачати ймовірність того, до якого класу належить об'єкт, враховуючи його характеристики. Математично це виражається наступною формулою:

$$P(Y = 1|X) = \frac{1}{1+e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots + \beta_p X_p)}}, \quad (2.10)$$

де:

$P(Y = 1|X)$ – імовірність того, що подія Y належить до класу 1 при заданих вхідних значеннях X;

e – базове число (~2.71828);

$\beta_0, \beta_1, \beta_2, \ldots, \beta_p$- параметри моделі, які оптимізують під час навчання;

$X_1, X_2, \ldots, X_p$- вхідні ознаки.

Параметри моделі $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ оптимізуються під час навчання, за допомогою оптимізаційних методів. Метою цього є знаходження значень параметрів на тренувальному наборі даних, які мінімізують помилку класифікації. Функція втрат визначає, наскільки ефективно модель прогнозує класи, і для логістичної регресії зазвичай використовується логарифмічна функція втрат. Під час навчання моделі параметри $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ знаходяться таким чином, щоб мінімізувати функцію втрат, що досягається шляхом використання різних методів оптимізації. За допомогою розрахованих показників, таких як точність, специфічність, збалансована точність та F1-міра можна оцінити ефективність моделі. Для перевірки здатності моделі до узагальнення на нових даних, її потрібно протестувати на тестовому наборі даних. Логістична регресія добре підходить для задач, де необхідно передбачити ймовірність належності об'єкта до певного класу, таких як прогнозування ймовірності рівня успішності, вірогідності відрахування здобувача, виконання освітньої складової на сесії **[84]**. У лістингу 2.1 представлено повний список параметрів для логістичної регресії (LogisticRegression), що використовувався при побудові моделі.

Лістинг 2.1. Повний список параметрів для LogisticRegression

*class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)*

Параметр penalty – задає значення штрафу, dual – формулювання оптимізаційної задачі, tol – порогове значення для зупинки процесу оптимізації, fit_intercept – визначає, чи слід додавати константу до функції прийняття рішень, intercept_scaling – масштабування константного члена, class_weight – збалансування ваги класів, random_state – значення для генератора випадкових чисел, solver – алгоритм для використання в задачі оптимізації, max_iter – максимальна кількість ітерацій, multi_class – визначає підхід до обробки з багатьма класами, verbose – для розв'язувачів liblinear та lbfgs, warm_start – повторне використання рішення, n_jobs – кількість ядер ЦП, l1_ratio – параметр змішування.

Отримана матриця помилок, для створеної моделі на базі логістичної регресії, представлена на рис. 2.2.

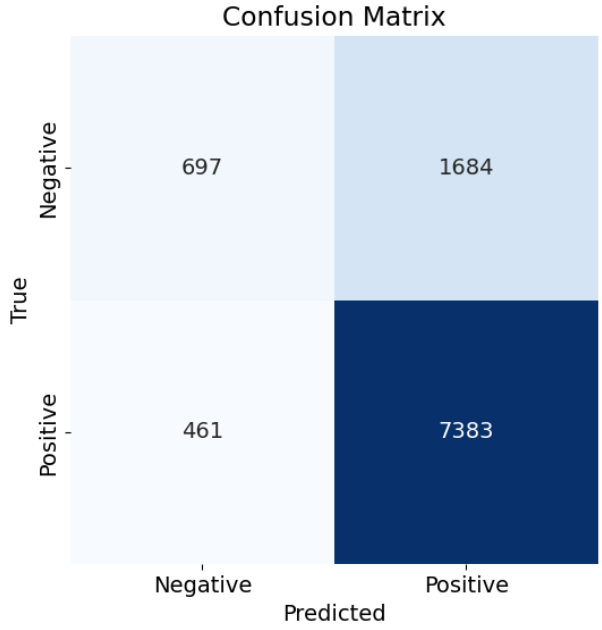


Рис. 2.2 – Матриця помилок для логістичної регресії

Виходячи з отриманої матриці проведено розрахунок значень характеризуючих загальну точність класифікації, результати яких наведені в табл. 2.3.

Таблиця 2.3

Розрахунки значень характеризуючих загальну точність

Метод	Точність	Чутливість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою (AUC)
Логістична регресія	0.79	0.941	0.292	0.873	0.616	0.7

Щоб наглядно оцінити здатність моделі до правильної класифікації, враховуючи різні значення порогового значення було побудовано ROC-криву, яка представлена на рис. 2.3.

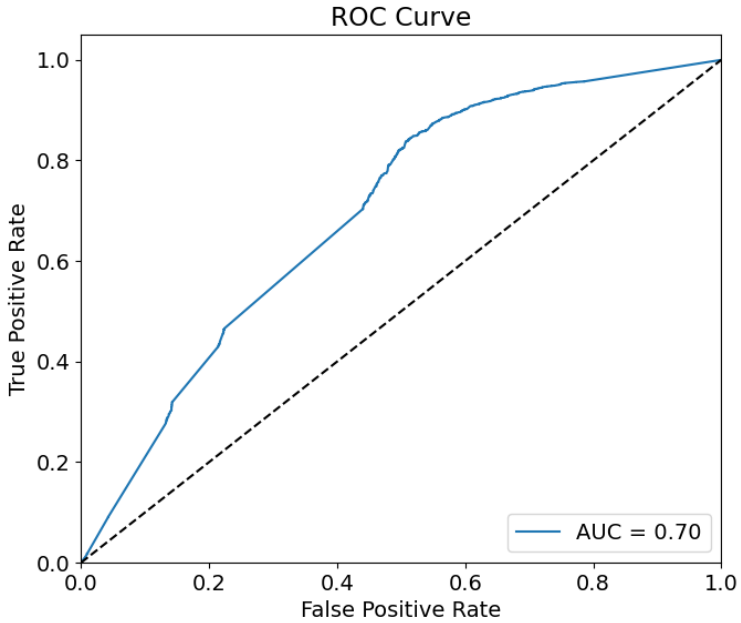


Рис. 2.3 – Графік ROC-кривої логістичної регресії

Для оцінки здатності моделі до навчання та узагальнення було побудовано криву навчання (learning curve) для моделі на базі логістичної регресії **[85]**. На рис. 2.4 відображено залежність точності моделі на тренувальній та тестовій вибірках від кількості навчальних прикладів.

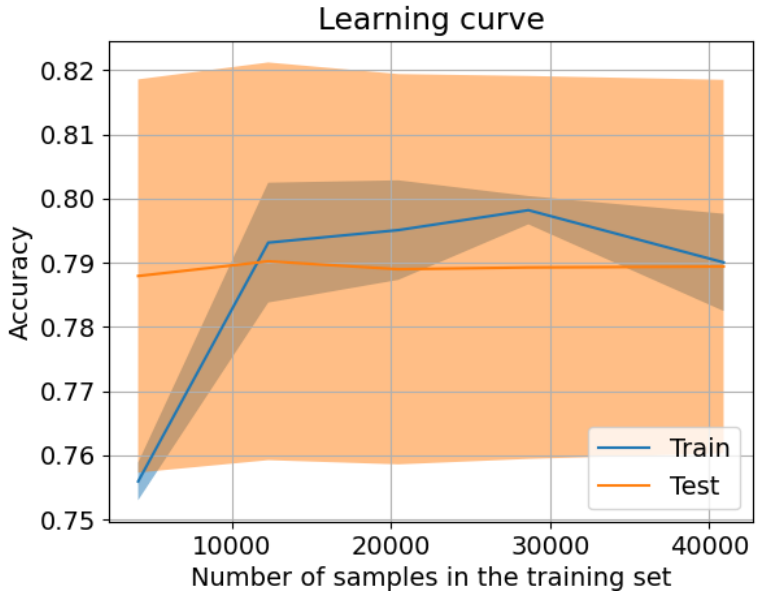


Рис. 2.4 – Графік кривої навчання моделі логістичної регресії

На графіку видно, що при малій кількості зразків (<5000) спостерігається значний розрив між тренувальною та тестовою точністю, причому точність на тренувальних даних нижча, ніж на тестових, що є нетиповим і може вказувати на особливості розподілу даних. У діапазоні 5000-10000 зразків, тренувальна точність швидко зростає, а тестова повільніше. Розрив між кривими зменшується. У діапазоні 10000-28000 зразків, тренувальна точність продовжує поступово зростати, досягаючи піку близько 0.80 при 29000 зразків. Тестова точність залишається відносно стабільною (близько 0.79). При кількості зразків, що перевищують 30000 обидві криві починають злегка знижуватися, що може вказувати на появу перенавчання або включення більш складних, зашумлених даних. Оптимальний розмір набору даних, при яких модель досягає найкращого балансу між тренувальною та тестовою точністю складає ~30000 зразків. При збільшенні кількості зразків тренувальна та тестова точність наближаються одна до одної, що свідчить про зменшення варіативності та

покращення генералізації моделі. Незважаючи на невелике падіння точності після 30000 зразків, графік не демонструє класичних ознак перенавчання (значне розходження кривих). Згідно графіка дана модель навряд зможе досягти точності вище 0.82 на наявних даних, незалежно від кількості зразків. Отриманий графік показує компроміс між розміром навчального набору та продуктивністю моделі.

Для перевірки наскільки добре ймовірності, які прогнозує класифікаційна модель, відповідають реальним частотам позитивних класів побудовано криву калібрування (Calibration Curve) **[86]**, представлена на рис. 2.5.

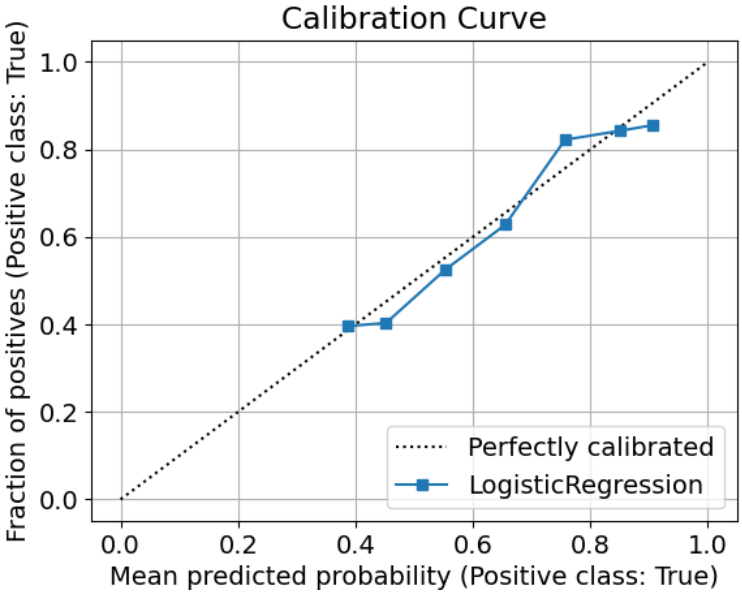


Рис. 2.5 – Графік кривої калібрування моделі (LR)

Пунктирна діагональна лінія на графіку представляє ідеальне калібрування ("Perfectly calibrated"), де передбачені ймовірності точно відповідають реальним частотам появи позитивного класу. Синя лінія з маркерами відображає фактичну калібрувальну криву моделі логістичної регресії. Ця крива досить близька до ідеальної діагоналі, що свідчить про хороше калібрування моделі. Можна відзначити, що у діапазоні ймовірностей 0.4-0.6 модель майже ідеально відкалібрована. А в області ймовірностей 0.6-0.7 спостерігається невелике відхилення від ідеальної лінії. Між 0.7-0.8 лінія проходить трохи вище діагоналі, що означає деяку недооцінку ймовірностей

моделлю (фактична частка позитивних результатів трохи вища за передбачену). У верхньому діапазоні (0.8-0.9) крива виходить на плато, що вказує на обмеження моделі в розрізненні дуже високих ймовірностей. Загалом, графік демонструє досить добре калібрування моделі. Логістична регресія вимагає мінімальних обчислювальних ресурсів, що робить її ефективною для роботи з великими даними. Та демонструє хорошу точність на невеликих та відносно простих наборах даних. Проте логістичну регресію не слід застосовувати, коли кількість ознак перевищує кількість спостережень, оскільки це може призвести до перенавчання. Тому як правило підходить лише для прогнозування дискретних результатів, коли залежна змінна має обмежений набір значень.

2.4.2. Метод опорних векторів

Метод опорних векторів (SVM) належить до алгоритму класифікаційного типу, який використовується для задач класифікації та регресії **[87]**. Він належить до сімейства лінійних класифікаторів і здатний розв'язувати проблеми навіть у складних, багатовимірних просторах. Однією з його ключових характеристик є те, що він здійснює пошук гіперплощини, яка максимізує відстань між точками різних класів, таким чином мінімізуючи помилки класифікації. У випадку лінійно роздільних класів цей метод дозволяє знайти оптимальну гіперплощину, яка розділяє класи з максимальною відстанню між найближчими точками кожного класу – так званими опорними векторами. Для нелінійних випадків використовуються ядрові функції, що дозволяють перетворювати дані у простір більшої розмірності, де лінійне розділення може бути виконано навіть при складних структурах даних. Для задачі класифікації лінійно роздільних даних SVM шукає гіперплощину, яка максимізує відстань до найближчих точок обох класів. У лінійному випадку метод опорних векторів намагається знайти гіперплощину, яка розділяє класи з максимальною маржею. Гіперплощина, яка розділяє два класи в просторі R^d, може бути виражена рівнянням **[88]**:

$$w^T x + b = 0, \quad (2.11)$$

де:

w – вектор ваг (параметри моделі);

x – вектор вхідних ознак (ознаки вхідного об'єкта);

b – зміщення або поріг;

T – транспонування вектора.

Функція маржі γ визначається як відстань між найближчими точками обох класів (опорними векторами) та гіперплощиною:

$$\gamma = \frac{2}{||w||} \quad (2.12)$$

де: $||w||$ – норма вектора ваг w.

Виходячи з описаного вище, задача оптимізації, яка стосується пошуку оптимальних значень параметрів для побудови гіперплощини, що розділяє класи виглядатиме так:

$$\min_{w,b} \frac{1}{2} \|w\|^2, \quad (2.13)$$

при умові: $y_i(w^T x_i + b) \geq 1, \quad \forall_i = 1, 2, \ldots, n$

де:

w – вектор ваг (параметри моделі), який визначає орієнтацію гіперплощини;

b – зміщення гіперплощини, яке контролює її розташування в просторі;

$y_i \in \{-1, +1\}$ – мітка класу для іі-го елемента;

x_i – вектор ознак іі-го елемента;

n – кількість елементів у навчальній вибірці.

У лістингу 2.2 представлено представлено повний список параметрів для методу опорних векторів (SVM), що використовувався при побудові моделі.

Лістинг 2.2. Повний список параметрів для SVM

class sklearn.svm.SVC(, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=True, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None)*

Параметр C – параметр регуляризації, kernel – визначає тип ядра, degree – ступінь поліноміальної функції, gamma – ядерний коефіцієнт, coef0 – незалежний термін у функції ядра, shrinking – вказує чи використовувати евристику скорочення, probability – вмикання оцінки ймовірності, tol – критерій допуску до зупинки, cache_size – розмір кешу ядра, class_weight – множники параметра C для кожного класу, verbose – увімкнення докладного виводу, max_iter – обмеження на ітерації, decision_function_shape – віибір прийняття рішень, break_ties – розрив зв’язків, random_state – генерація псевдовипадкових чисел.

Отримана матриця помилок, для створеної моделі на базі методу опорних векторів, представлена на рис. 2.6.

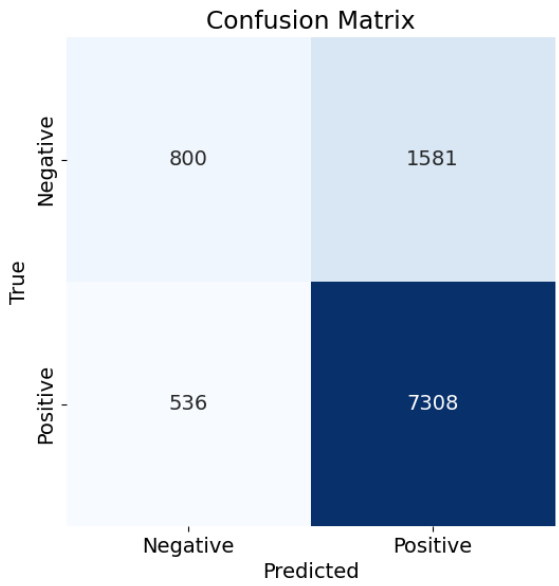


Рис. 2.6 – Матриця помилок для методу опорних векторів

Виходячи з отриманої матриці проведено розрахунок значень характеризуючих загальну точність класифікації, результати розрахунків наведені в табл. 2.4.

Таблиця 2.4

Розрахунки значень характеризуючих загальну точність

Метод	Точність	Чутливість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою
Метод опорних векторів	0.792	0.931	0.335	0.873	0.633	0.66

Щоб наглядно оцінити здатність моделі до правильної класифікації, враховуючи різні значення порогового значення було побудовано ROC-криву, яка представлена на рис. 2.7.

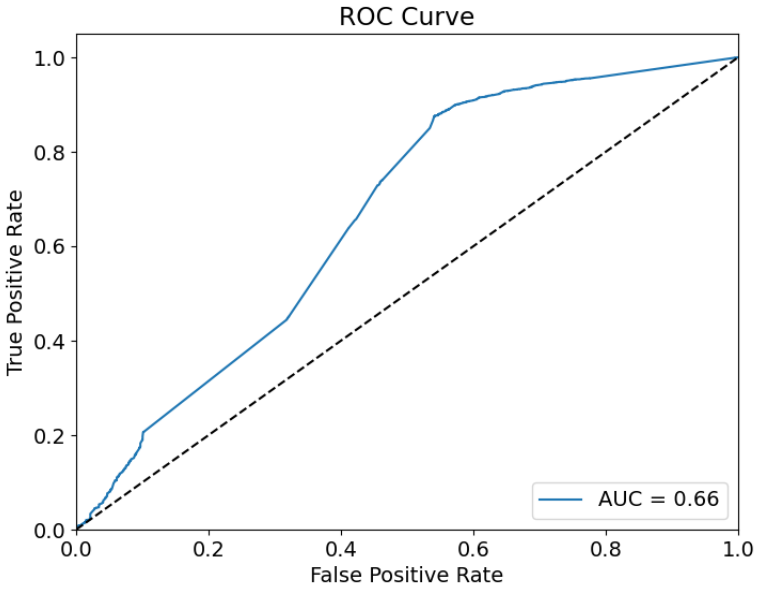


Рис. 2.7 – Графік ROC-кривої методу опорних векторів

Побудовану криву навчання (learning curve) для моделі на базі методу опорних векторів відображено на рис. 2.8.

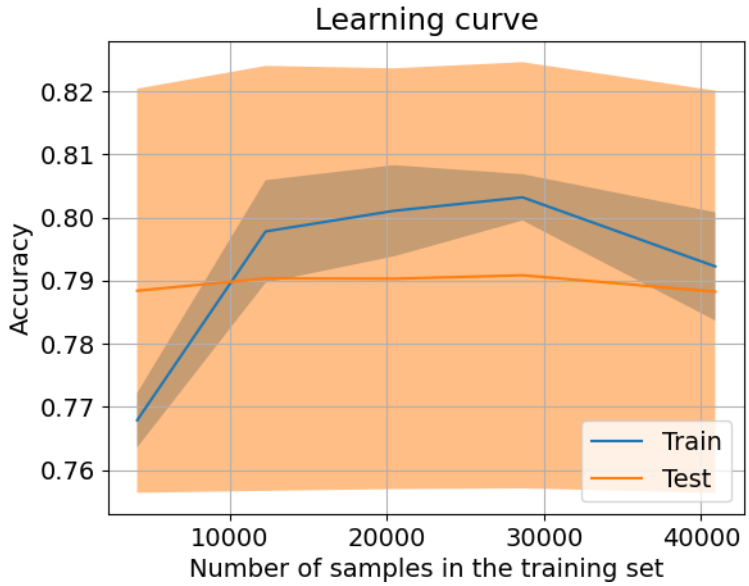


Рис. 2.8 – Графік кривої навчання моделі опорних векторів (SVM)

На графіку видно, що при малій кількості зразків (<4000) спостерігається розрив між тренувальною та тестовою точністю, причому точність на тренувальних даних

нижча, ніж на тестових, що є нетиповим і може вказувати на особливості розподілу даних. У діапазоні 3000-10000 зразків, тренувальна точність швидко зростає. Розрив між кривими зменшується. У діапазоні 10000-27000 зразків, тренувальна точність продовжує поступово зростати, досягаючи піку близько 0.803 при 28000 зразків. Тестова точність залишається відносно стабільною (близько 0.79). При кількості зразків, що перевищують 30000 тренувальна крива починає злегка знижуватися, що може вказувати на появу перенавчання або включення більш складних, зашумлених даних. Оптимальний розмір набору даних, при яких модель досягає найкращого балансу між тренувальною та тестовою точністю складає ~28000 зразків. При збільшенні кількості зразків тренувальна та тестова точність наближаються одна до одної, що свідчить про зменшення варіативності та покращення генералізації моделі. Незважаючи на невелике падіння точності після 28000 зразків, графік не демонструє класичних ознак перенавчання (значне розходження кривих). Згідно графіка дана модель навряд зможе досягти точності вище 0.81 на наявних даних, незалежно від кількості зразків. Отриманий графік показує компроміс між розміром навчального набору та продуктивністю моделі. Побудовану криву калібрування (Calibration Curve) для моделі на базі методу опорних векторів представлено на рис. 2.9.

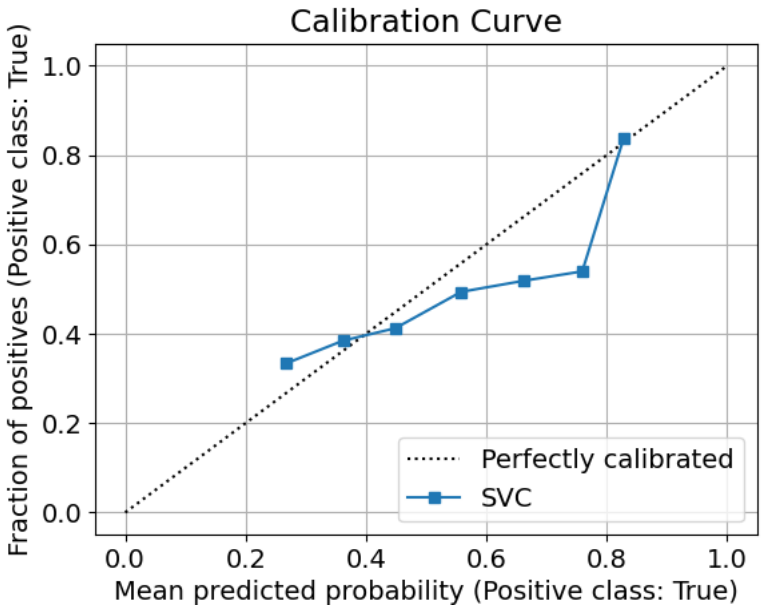


Рис. 2.9 – Графік кривої калібрування моделі (SVC)

На графіку видно, що крива в діапазоні 0.4-0.5 демонструє відхилення вгору, що вказує на певну недооцінку ймовірностей (фактична частка позитивних результатів трохи вища, ніж передбачає модель). У діапазоні 0.5-0.7 спостерігається плато, де крива вирівнюється горизонтально. Це проблематична ділянка, оскільки різні передбачені ймовірності (0.5, 0.55, 0.6, 0.65) дають приблизно однакову фактичну частку позитивних результатів (~0.55). Модель не розрізняє ці ймовірності належним чином. У діапазоні 0.7-0.85 відбувається різкий вертикальний стрибок, що свідчить про значну недокаліброваність. Моделі важко розрізняти ймовірності в цьому діапазоні. В результаті модель потребує додаткового калібрування, особливо у верхньому діапазоні ймовірностей, в у діапазоні середніх ймовірностей (0.5-0.7) модель не забезпечує адекватної диференціації, що зменшує її корисність для ранжування передбачень. Такий аналіз калібрувальної кривої є важливим кроком перед впровадженням моделі в робоче середовище, особливо для задач, де важлива не лише класифікація, але й точність імовірнісних оцінок. До переваг можна віднести хорошу здатність до узагальнення на нових, невідомих даних та можливість оптимізації загальної помилки на навчальних даних, навіть у випадках з невеликою кількістю помилкових класифікацій. До недоліків можна віднести часову складність алгоритму, яка може бути високою, особливо при великих обсягах даних, оскільки потребує обчислення парних відстаней між усіма точками в просторі ознак. Для великих наборів даних SVM може бути менш ефективним у порівнянні з іншими методами, такими як випадковий ліс чи нейронні мережі, через високу обчислювальну складність.

2.4.3. Випадковий ліс

Випадковий ліс (Random Forest) – це ансамблевий метод машинного навчання, що складається з множини дерев рішень **[89]**. Основна ідея полягає в тому, що безліч слабких класифікаторів (дерев рішень) можуть створити сильний класифікатор.

Кожне дерево в лісі будується за допомогою випадкової підвибірки з даних і випадкової підмножини ознак. Випадковий ліс використовує принцип агрегації, щоб комбінувати результати всіх дерев для отримання остаточного прогнозу. У випадковому лісі кожне дерево навчається на випадковій вибірці з оригінального набору даних, а також для кожного розгалуження дерева вибирається випадкова підмножина ознак. Для побудови кожного дерева випадковий ліс використовує техніку бутстрепу (bootstrap), щоб вибрати випадкову підвибірку з оригінальних даних, де кожне дерево навчання на своїй підвибірці даних. Якщо є T дерев, прогноз кожного дерева T_j для нового спостереження x є класом, який дерево передбачає **[90]**:

$$T_j(x) = y_j, \text{ (2.14)}$$

де:

$T_j(x)$ – прогноз, зроблений jj-м деревом для спостереження x;

y_j – клас або значення, яке передбачає j-е дерево для об'єкта x;

x – вхідні ознаки або характеристика спостереження;

j – індекс дерева в ансамблі дерев (де T – загальна кількість дерев).

У випадку класифікації, кожне дерево передбачає клас y_j, до якого належить спостереження x, а в кінцевому підсумку, за допомогою голосування (або іншого методу агрегації) формується фінальний прогноз. Математично процес голосування серед усіх дерев у випадковому лісі для визначення фінального класу $\hat{y}$ для нового спостереження x описується так **[91]**:

$$\hat{y} = arg \max_{y \in Y} \sum_{j=1}^{T} I(T_j(x) = y), \text{ (2.15)}$$

де:

$\hat{y}$ – фінальний прогноз, який є класом, до якого, ймовірно, належить спостереження x;

$y \in Y$ – ймовірні класи, до яких може належати спостереження xx, а Y – множина всіх можливих класів;

$I(T_j(x) = y)$ – індикаторна функція, яка дорівнює 1, якщо j-е дерево передбачає клас y для об'єкта x, і 0 в іншому випадку;

$T_j(x)$ – прогноз j-го дерева для об'єкта j, тобто клас або значення, яке це дерево передбачає для цього спостереження.

Основна ідея полягає в використанні великої кількості вирішальних дерев, кожне з яких окремо може дати не надто точні прогнози, але комбінуючи результати всіх дерев, можна досягти високої якості передбачень. У разі прогнозування успішності здобувачів, це може бути передбачення, чи складе здобувач іспит на "відмінно" чи "задовільно" або не складе взагалі. При цьому кожне дерево побудовано на основі різних підвибірок даних, що дозволяє зберігати різноманітність і знижувати ймовірність перенавчання. У результаті, якщо дерева побудовані на достатньо різноманітних даних, можна отримати точні передбачення щодо успішності здобувачів, які будуть здатні адаптуватися до нових, раніше невідомих даних. Загалом, використання випадкового лісу для прогнозування успішності здобувачів є ефективним завдяки здатності алгоритму працювати з великими наборами даних та враховувати багато змінних факторів, які можуть впливати на точність. У лістингу 2.3 представлено розділ даних на навчальний та тестовий набори, ініціалізацію та тренування для випадкового лісу (RandomForest) з використанням параметрів по замовчуванню.

Лістинг 2.3. Повний список параметрів для RandomForestClassifier

*class sklearn.ensemble.RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None, monotonic_cst=None)*

Параметр *n_estimators* – задає кількість дерев у лісі, *criterion* – визначає критерій (параметр залежить від дерева), *max_depth* – задає максимальну глибина дерева, *min_samples_split* – мінімальна кількість зразків, необхідних для розбиття внутрішнього вузла, *min_samples_leaf* – визначає мінімальну суму ваги для кожного листка, *min_weight_fraction_leaf* – мінімальна зважена частка загальної суми ваг (усіх вхідних вибірок), *max_features* – визначає кількість ознак, *max_leaf_nodes* – кількість листових вузлів, *min_impurity_decrease* – розщеплення вузла, *bootstrap* – визначає вибір вибірки, *oob_score* – оцінка набору навчальних даних, *n_jobs* – кількість завдань, *random_state* – контролює випадковість початкового завантаження зразків, verbose – відображає вихідну інформацію, *warm_start* – визначає, чи буде використовуватися попередньо навчена модель для ініціалізації та навчання нової моделі, *class_weight* – вирівнювання ваги класів, *ccp_alpha* – параметр обрізки, *max_samples* – максимальна кількість зразків, *monotonic_cst* – обмеження монотонності.

Отримана матриця помилок, для створеної моделі на базі випадкового лісу представлена на рис. 2.10.

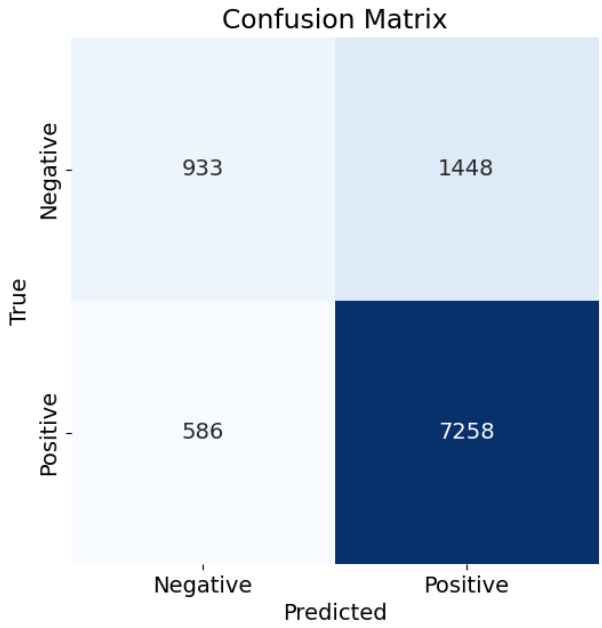


Рис. 2.10 – Матриця помилок для випадкового лісу

Виходячи з отриманої матриці проведено розрахунок значень характеризуючих загальну точність класифікації, результати розрахунків наведені в табл. 2.5.

Таблиця 2.5

Розрахунки значень характеризуючих загальну точність

Метод	Точність	Чутли-вість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою
Випадковий ліс	0.801	0.925	0.391	0.877	0.658	0.73

Щоб наглядно оцінити здатність моделі до правильної класифікації, було побудовано ROC-криву, яка представлена на рис. 2.11.

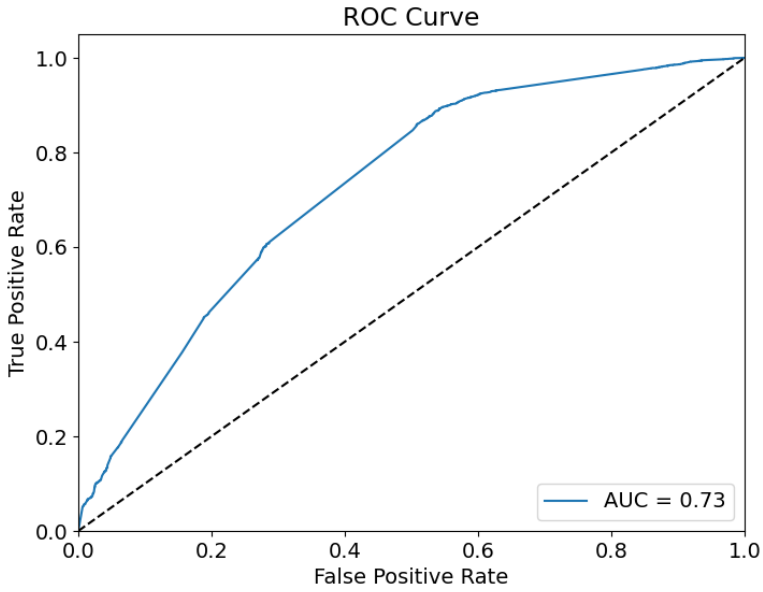


Рис. 2.11 – Графік ROC-кривої випадкового лісу

Побудовану криву навчання (learning curve) для моделі на базі випадкового лісу представлено на рис. 2.12.

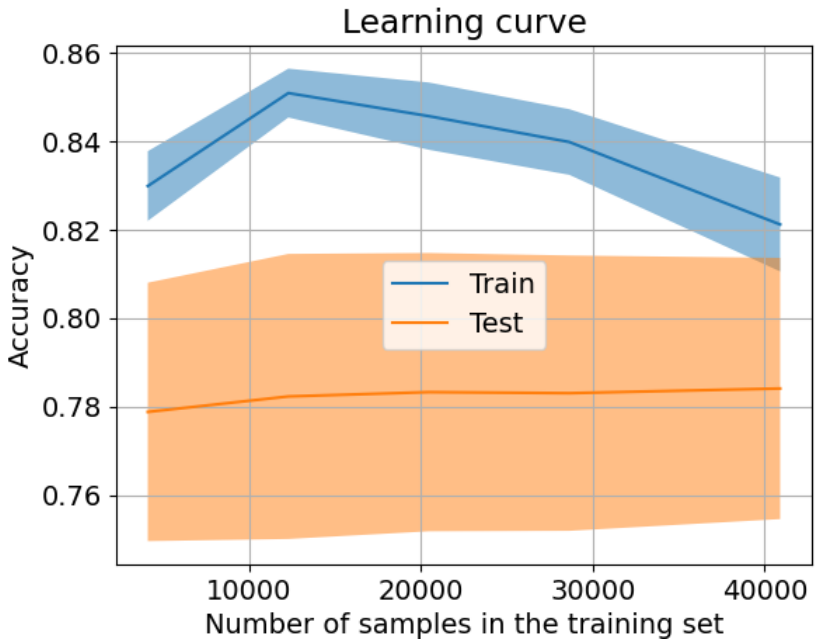


Рис. 2.12 – Графік кривої навчання моделі випадкового лісу (RF)

Тренувальна вісь починається зі значення ~0.83 ще при малій кількості зразків та досягає піку близько 0.85 при ~10000 зразків. Далі відбувається поступове зниження до ~0.82 при 40000 зразків. В даному проміжку графік демонструє

зменшення точності на тренувальних даних при збільшенні їх кількості. Тестова точність (помаранчева вісь) починається з ~0.78 при малій кількості зразків, а потім поступово і зростає до ~0.79 при ~20000 зразків. Далі вона залишається практично стабільною до 40000 зразків і демонструє значно менші коливання, ніж тренувальна точність. Суттєвий розрив між тренувальною та тестовою точністю присутній на всіх етапах навчання (приблизно 4-6%), що вказує на наявність перенавчання (overfitting). Тренувальна точність знижується з ростом кількості зразків, оскільки моделі стає складніше підлаштовуватися під більш різноманітні дані. Тестова точність майже стабільна, що свідчить про те, що додавання більшої кількості навчальних зразків не покращує генералізацію моделі суттєво. Для підвищення продуктивності моделі доцільно застосувати методи регуляризації або спростити модель, щоб зменшити розрив між тренувальною та тестовою точністю. Побудовану криву калібрування (Calibration Curve) для моделі на базі методу випадкового лісу представлено на рис. 2.13.

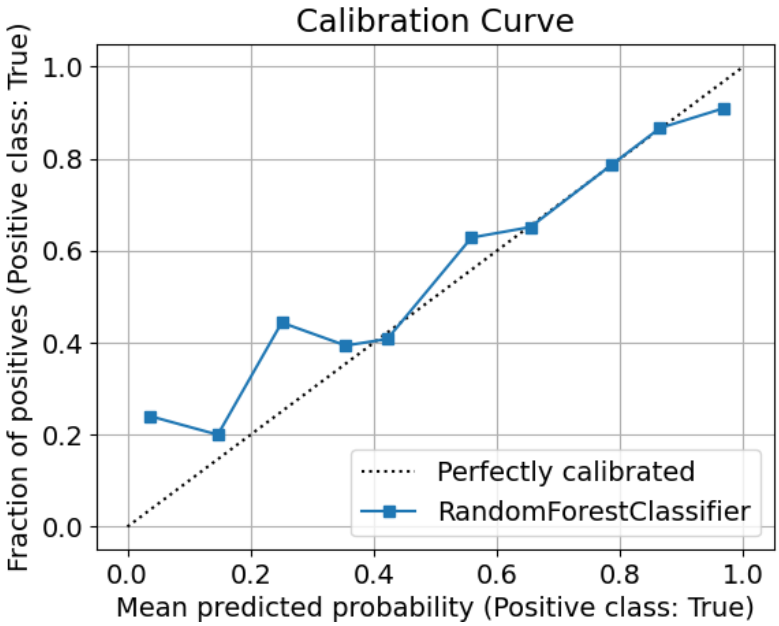


Рис. 2.13 – Графік кривої калібрування моделі (RF)

На графіку видно, що на низьких ймовірностях (0-0.2) модель переоцінює ймовірності. У діапазоні 0.2-0.4: калібрувальна крива різко піднімається і наближається до ідеальної лінії. У середньому діапазоні (0.4-0.6) спостерігаються

коливання навколо ідеальної лінії з невеликими відхиленнями в обидва боки. При ймовірностях 0.6-0.7: модель трохи переоцінює ймовірності, калібрувальна крива проходить нижче ідеальної діагоналі. У високому діапазоні (0.8-1.0) модель майже ідеально відкалібрована, лінія майже збігається з діагоналлю. Графік демонструє типові проблеми для Random Forest – схильність видавати або дуже високі, або дуже низькі ймовірності, з недостатньою диференціацією в середньому діапазоні. До переваг можна віднести: високу точність прогнозів завдяки використанню багатьох дерев рішень та стійкість до перенавчання, оскільки використовується ансамбль дерев, що дає змогу знизити ризик створення надмірно складних моделей, які погано генералізуються. Модель ефективно працює з великими та складними наборами даних, включаючи як числові, так і категоріальні змінні. Випадковий ліс може оцінювати важливість кожної ознаки, що допомагає зрозуміти, які фактори найбільше впливають на результат. До недоліків можна віднести чутливість до шуму, оскільки якщо дані містять багато шуму, результат може стати менш точним, оскільки деревам може бути важко знаходити корисні шаблони серед випадкових варіацій.

2.4.4. Наївний Баєс

Наївний Баєсівський класифікатор (Naive Bayes) – це статистичний метод класифікації, який ґрунтується на застосуванні теореми Баєса з припущенням, що всі ознаки (вхідні характеристики) є умовно незалежними одна від одної **[92]**. Це означає, що для кожної ознаки не враховуються взаємозв'язки з іншими ознаками. Такий підхід робить модель дуже простою, але водночас досить ефективною, особливо в умовах великих наборів даних. У більшості випадків для оцінки параметрів наївного баєсівського класифікатора використовують метод максимальної правдоподібності (Maximum Likelihood Estimation, MLE) **[93]**. Це означає, що в деяких випадках модель може працювати навіть без суворого застосування Баєсівської ймовірності, використовуючи лише параметри, оцінені методом максимального правдоподібності. Незважаючи на свою простоту та спрощення, Наївний Баєсівський класифікатор часто

перевершує інші більш складні методи, такі як нейронні мережі, в реальних практичних задачах. Однією з головних переваг наївного баєсівського класифікатора є невелика кількість даних, необхідних для тренування моделі, що робить його швидким і ефективним навіть при обмежених обсягах даних. Баєсівський класифікатор використовує ймовірнісну модель для прогнозування класу, до якого належить об'єкт. Згідно з теоремою Баєса, апостеріорну ймовірність можна обчислити через умовні ймовірності, математично це можна записати як **[94]**:

$$P(y\,|x) = \frac{P(y) * P\,(x\,|y)}{P(x)}, \quad (2.16)$$

де:

$P(y)$ – апріорна ймовірність класу y;

$P(y|x)$ – ймовірність спостереження x, якщо об'єкт належить до класу y;

$P(x)$ – нормалізаційний фактор, що враховує ймовірність спостереження x у всіх класах.

Для класифікації необхідно обчислити $P(y)$ – ймовірність того, що об'єкт належить класу y, а також $P(y|x)$ – ймовірність того, що ознака x і належить класу x. Ці параметри оцінюються в процесі навчання класифікатора, що також називається **тренуванням моделі**. Для тренування наївного баєсівського класифікатора використовується метод максимальної правдоподібності, де ймовірності класів і умовні ймовірності для кожної ознаки оцінюються на основі тренувальних даних.

Лістинг 2.4. Повний список параметрів для GaussianNB

class sklearn.naive_bayes.GaussianNB(, priors=None, var_smoothing=1e-09)*

Параметр *priors* – апріорні ймовірності класів, за замовчуванням=*None*. Якщо вказано, апріорні ймовірності не коригуються відповідно до даних. Параметр *var_smoothingfloat* – частина найбільшої дисперсії всіх ознак, яка додається до дисперсій для стабільності розрахунку, за замовчуванням=*1e-9*.

Отримана матриця помилок, для створеної моделі на базі Наївного Баєса представлена на рис. 2.14.

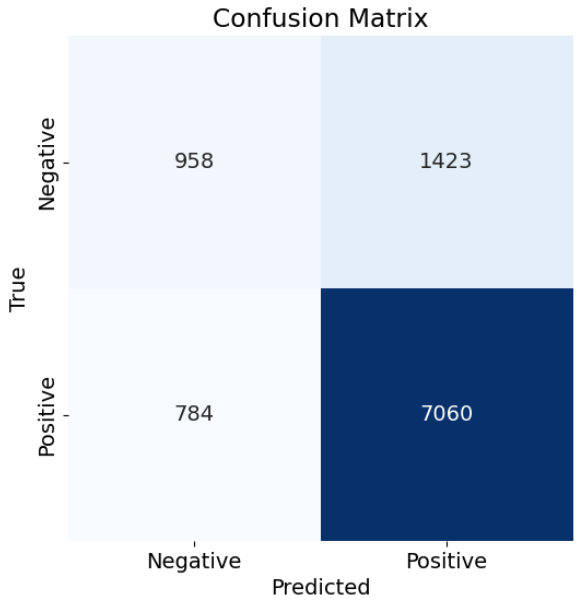


Рис. 2.14 – Матриця помилок для Наївного Баєса

Виходячи з отриманої матриці проведено розрахунок значень характеризуючих загальну точність класифікації, результати розрахунків наведені в табл. 2.6.

Таблиця 2.6

Розрахунки значень характеризуючих загальну точність

Метод	Точність	Чутли-вість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою (AUC)
Наївний Баєс	0.784	0.900	0.402	0.864	0.651	0.697

Щоб наглядно оцінити здатність моделі до правильної класифікації, було побудовано ROC-криву, яка представлена на рис. 2.15.

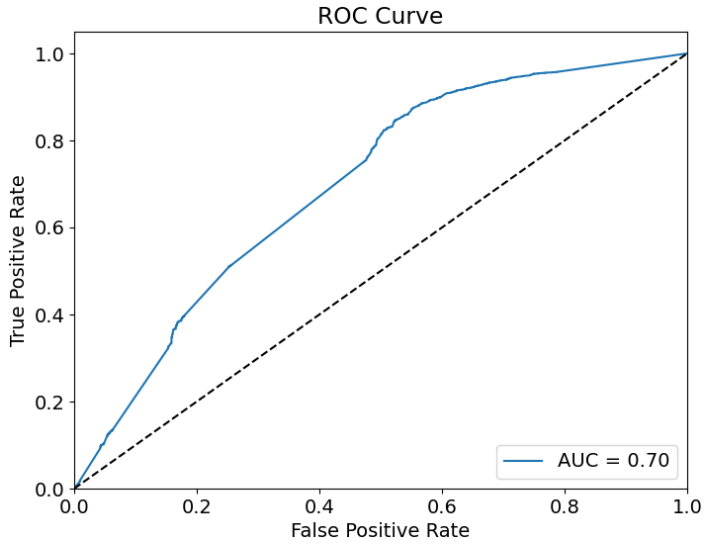


Рис. 2.15 – Графік ROC-кривої Наївного Баєса

Побудовану криву навчання (learning curve) для моделі на базі Наївного Баєса представлено на рис. 2.16.

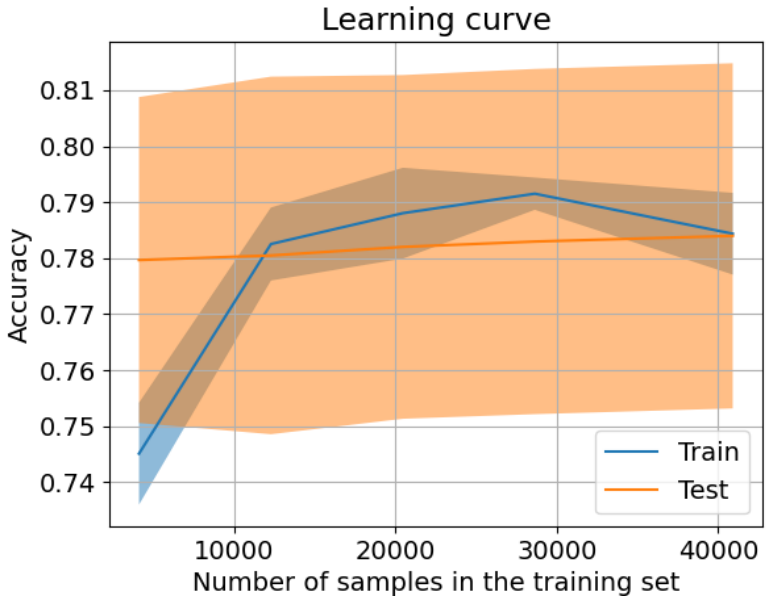


Рис. 2.16 – Графік кривої навчання моделі Наївного Баєса (NB)

На графіку видно, що при малій кількості зразків (< 2000) спостерігається розрив між тренувальною та тестовою точністю (~0.74 – 0.78), причому точність на тренувальних даних нижча, ніж на тестових, що є нетиповим і може вказувати на

особливості розподілу даних. У діапазоні 3000-11000 зразків, тренувальна точність швидко зростає до ~0.78. Розрив між кривими зменшується. У діапазоні 11000-27000 зразків, тренувальна точність продовжує поступово зростати, досягаючи піку близько 0.793 при 28000 зразків. Тестова точність є стабільною на рівні ~0.783. Оптимальний компроміс між тренувальною та тестовою точністю досягається при обсязі навчальної вибірки близько 28 000 зразків. Подальше збільшення даних супроводжується незначним зниженням тренувальної точності, що може бути пов'язано з підвищенням складності або зашумленості даних, без виражених ознак перенавчання. Зближення тренувальної та тестової кривих свідчить про зменшення варіативності моделі та покращення її здатності до узагальнення. На основі графіка можна зробити висновок, що досягнення точності понад 0.80 на наявних даних є малоймовірним. Побудовану криву калібрування (Calibration Curve) для моделі на базі Наївного Баєса представлено на рис. 2.17.

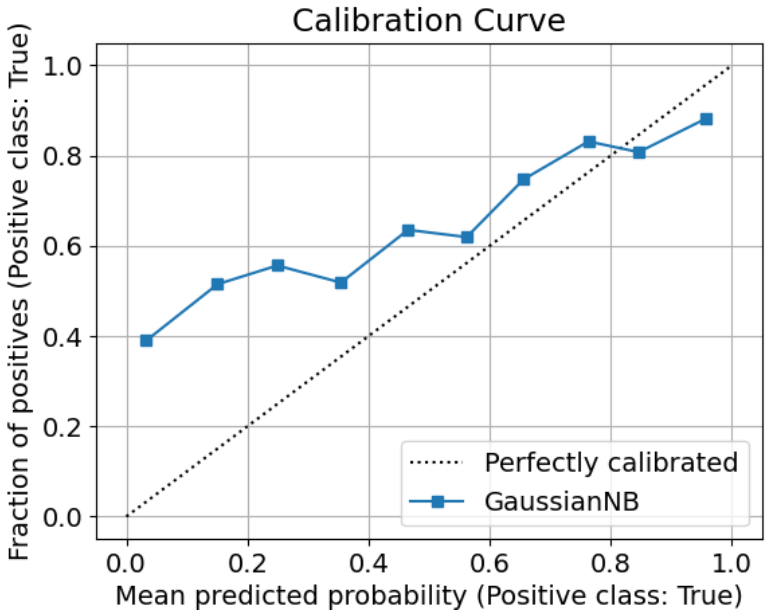


Рис. 2.17 – Графік кривої калібрування моделі (NB)

На графіку видно, що при значеннях (0.0-0.4) модель постійно недооцінює ймовірності, а калібрувальна крива знаходиться значно вище ідеальної діагоналі. При середній ймовірності (0.4-0.6) крива продовжує проходити вище діагоналі, хоча

розрив між фактичною та передбаченою ймовірністю тут менший. При значеннях (0.6-0.8) крива перетинає діагональ і наближається до ідеального калібрування. У випадку високої ймовірності (0.8-1.0) спостерігаються коливання, де спочатку крива проходить трохи нижче діагоналі (при ~0.8), а потім знову піднімається вище (при ~1.0). Модель на Gaussian Naive Bayes суттєво недооцінює ймовірності в нижньому діапазоні, що типово для цього алгоритму через його сильні припущення про незалежність ознак. На графіку видно що крива має кілька невеликих "сходинок" і коливань, що вказує на нестабільність калібрування в низьких діапазонах. При використанні моделі для прийняття рішень варто враховувати систематичну недооцінку ймовірностей. Зазвичай Naive Bayes видає некалібровані ймовірності через "наївне" припущення про незалежність ознак, яке рідко виконується в реальних даних. Мережі Баєса ефективно працюють з неповними даними, використовуючи умовні ймовірності для оцінки відсутніх значень на основі інших змінних. Побудовані моделі не мають надмірної складності, зберігаючи чітке уявлення про залежності між змінними, що робить їх корисними в реальних застосуваннях. До недоліків можна віднести умовну незалежність змінних, що може бути обмеженням для складних взаємозв'язків, а також складність роботи з неперервними змінними без додаткових апроксимацій.

2.4.5. Нейронні мережі (MLPClassifier)

Нейронні мережі – це клас моделей машинного навчання, що складається з великої кількості взаємопов'язаних "нейронів". Кожен нейрон у мережі виконує обчислення, приймаючи вхідні дані, застосовуючи до них ваги, додавання зміщення і обчислюючи результат через функцію активації **[95]**. Вони складаються з багатьох з'єднаних між собою нейронів, які працюють у кілька етапів, проходячи через вхідний, прихований та вихідний шари. Кожен нейрон виконує операції з вхідними даними і передає результат на наступний рівень. Це дозволяє нейронним мережам моделювати складні нелінійні залежності між вхідними ознаками і вихідними прогнозами, що є

критичним при прогнозуванні таких складних явищ, як успішність здобувачів. Математична модель нейронної мережі описує функціонування кожного нейрону і взаємодію між нейронами в шарах. Математична формула для одного нейрону в прихованому шарі має наступний вигляд **[96]**:

$$y = f(\sum_{i=1}^{n} w_i x_i + b), \quad (2.17)$$

де:

y – вихід нейрону;

x_i – вхідні сигнали (ознаки) для нейрону;

w_i – ваги, які визначають важливість кожного входу;

b – зміщення (bias), яке дозволяє моделі бути гнучкішою;

f – функція активації (наприклад: сигмоїда, ReLU тощо).

Основна ідея нейронних мереж у контексті прогнозування успішності здобувачів полягає в їх здатності вчитися на великих наборах даних, адаптуючи свої ваги та параметри для точного передбачення результату. Кожен вхідний параметр (наприклад, оцінки за попередні курси, рівень відвідуваності, участь у позанавчальних активностях) передається через нейронну мережу. Кожен шар мережі виконує певну трансформацію цих вхідних даних, і в результаті мережа генерує передбачення щодо ймовірності успішності здобувача (наприклад, чи здобувач отримає високий бал на екзамені, чи не складе його взагалі). Загальна формула для нейронної мережі, що включає кілька шарів, виглядає так **[97]**:

$$y_{out} = f(\sum_{i=1}^{n} w_i^{(k)} x_i + b^{(k)}, \quad (2.18)$$

де:

y_{out} – вихід нейронної мережі після проходження через кілька шарів;

x_i – ваги для кожного входу в шарі k;

$w_i^{(k)}$ – зміщення (bias) для шару k;

$b^{(k)}$ – відповідні ваги та зміщення для кожного шару;

f – функція активації для кожного шару.

Для кожного нейрону у шарі проводиться лінійне комбінування вхідних значень x_i, яке потім коригується через ваги $w_i^{(k)}$ і зміщення $b^{(k)}$, а результат передається через функцію активації f, щоб отримати вихід. Нейронна мережа складається з кількох шарів, де кожен шар виконує операції активації, що дозволяють моделювати складні взаємозв'язки між вхідними даними (наприклад, оцінки за певними предметами) і результатами успішності. Наприклад, для прогнозування успішності здобувача на основі його відвідуваності та академічних досягнень, вхідними даними можуть бути оцінки за попередні курси, частота відвідування лекцій і участь у семінарах, а вихід може бути ймовірністю того, що здобувач складе курс на високий бал. Під час навчання нейронна мережа коригує свої ваги таким чином, щоб мінімізувати різницю між передбаченням і фактичними результатами. Процес оптимізації може здійснюватися за допомогою алгоритмів, таких як градієнтний спуск, що дозволяє поступово знаходити оптимальні параметри для досягнення найкращих результатів. Нейронні мережі мають здатність до глибокого навчання, що дозволяє їм працювати з великими наборами даних, де важливо враховувати численні фактори, які можуть впливати на успішність здобувачів. Зокрема, мережа може враховувати складні взаємозв'язки між оцінками, соціальною активністю та іншими характеристиками, що мають вплив на кінцевий результат. Однією з основних переваг нейронних мереж є їх здатність до самооптимізації: з кожним новим прикладом даних мережа коригує свої параметри, таким чином покращуючи точність прогнозів **[98]**. У випадку прогнозування успішності здобувачів нейронні мережі можуть враховувати варіативність здобувачів у процесі навчання, наприклад, якщо здобувач покращив

свої оцінки в кінці курсу або змінив свою поведінку в класі. Завдяки своїй гнучкості, нейронні мережі можуть бути застосовані до широкого спектра задач у навчанні, зокрема для прогнозування успішності здобувачів, аналізу факторів, що впливають на академічні досягнення, а також для виявлення можливих проблем у навчальному процесі.

Лістинг 2.5. Повний список параметрів для MLPClassifier

*class sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(100,), activation ='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state= None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_ momentum=True, early_stopping=False, validation_fraction=0.1, beta_1= 0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000)*

Параметр *hidden_layer_sizes* – кількість нейронів у прихованих шарах, *activation* – функція активації в прихованих шарах, *solver* – алгоритм оптимізації, *alpha* – коефіцієнт L2-регуляризації, *batch_size* – розмір пакета для оновлення вагів, *learning_rate* – стратегія зміни швидкості навчання, *learning_rate_init* – початкова швидкість навчання, *power_t* – ступінь зменшення швидкості при 'invscaling', *max_iter* – максимальна кількість епох навчання, *shuffle* – чи перемішувати дані перед кожною ітерацією, *random_state* – початкове значення генератора випадкових чисел, *verbose* – вивід проміжної інформації в консоль, *warm_start* – продовжити навчання з попереднього стану, *momentum* – імпульс для градієнтного спуску, *nesterovs_ momentum* – використання імпульсу, *early_stopping* – зупинка навчання при відсутності покращення, *validation_fraction* – частка даних для валідації при early_stopping, *beta_1* – параметр експоненціального згладжування для моменту, *beta_2* – параметр згладжування для квадрату градієнта, *epsilon* – мале число для

уникнення ділення на нуль, *n_iter_no_change* – кількість ітерацій без покращення для зупинки, *max_fun* – максимальна кількість викликів функції втрат для 'lbfgs'.

Проведений порівняльний аналіз характеристик нейронної мережі для різної кількості нейронів у прихованому шарі (10, 50, 100, 150, 200) показав, що модель з 100 нейронами найкраще себе зарекомендувала. На основі комплексного дослідження 8 ключових метрик якості класифікації (Accuracy, F1-Score, Precision, Recall, Specificity, Balanced Accuracy, ROC AUC, PR AUC) було встановлено, що конфігурація зі 100 нейронами демонструє найкращий збалансований результат серед усіх досліджених варіантів. Модель з 100 нейронами досягла точності 79,4%, чутливості 93,5%, F1-міри 87,4% та площі під ROC-кривою 0,70. Ці показники представлені на рис.2.18 – 2.20, та перевищують результати як менших конфігурацій (10 та 50 нейронів), так і більших (150 та 200 нейронів), що підтверджує оптимальність обраної архітектури для поставленої задачі класифікації.

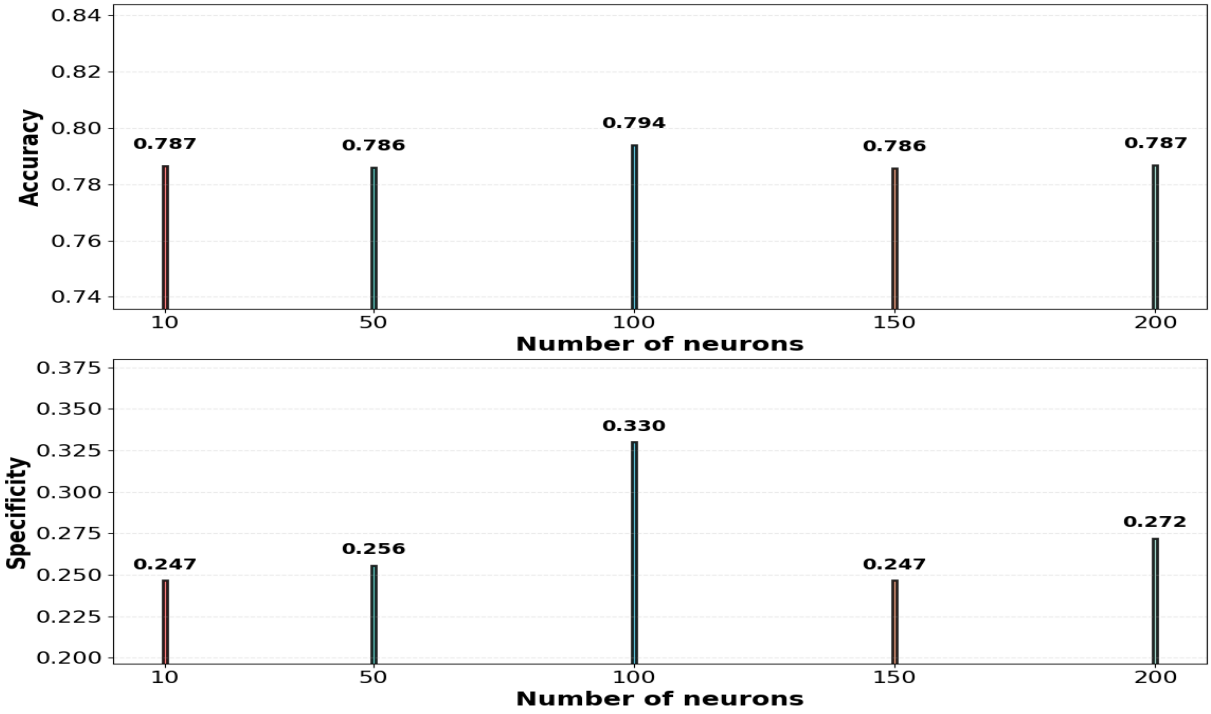


Рис. 2.18 – Порівняння Accuracy та Specificity для різної кількості нейронів

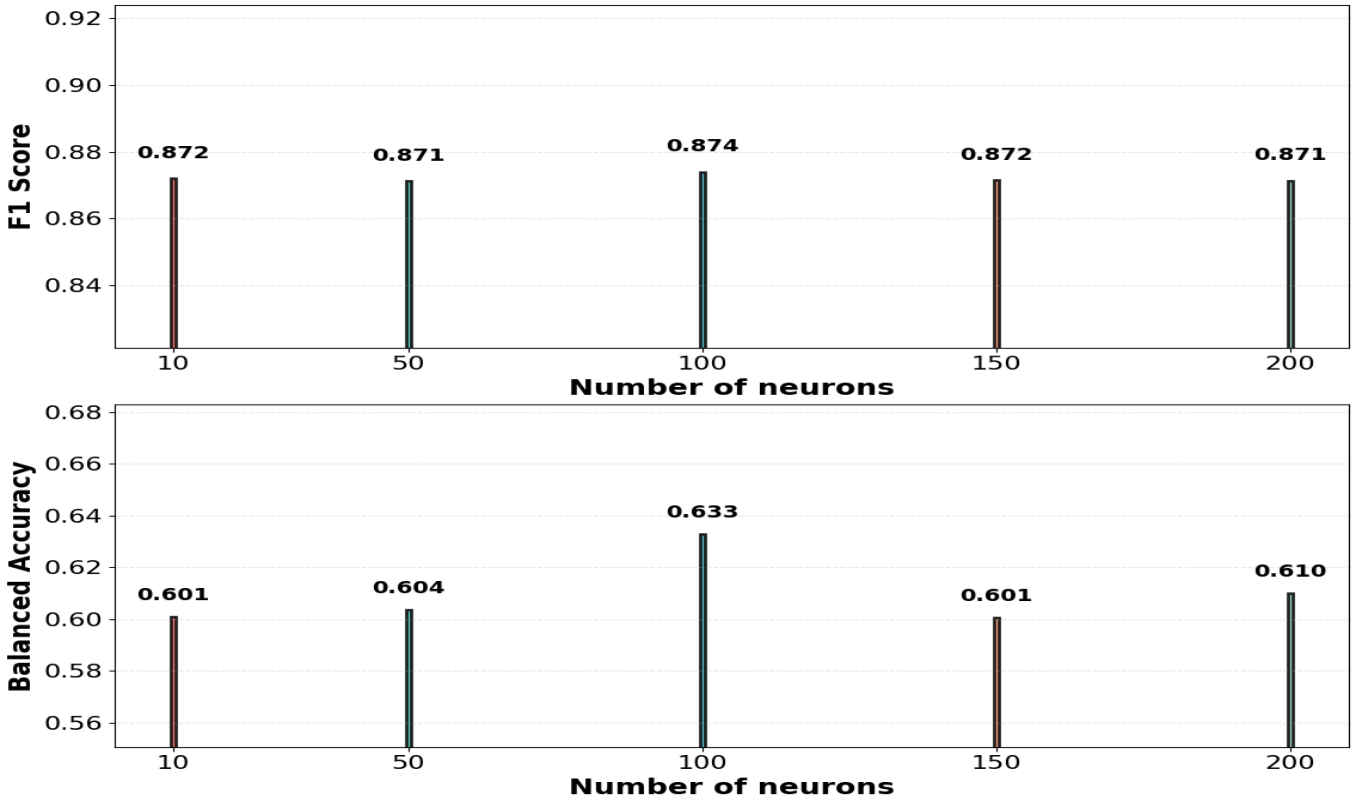


Рис. 2.19 – Порівняння F1 Score та Balanced Accuracy для різної кількості нейронів

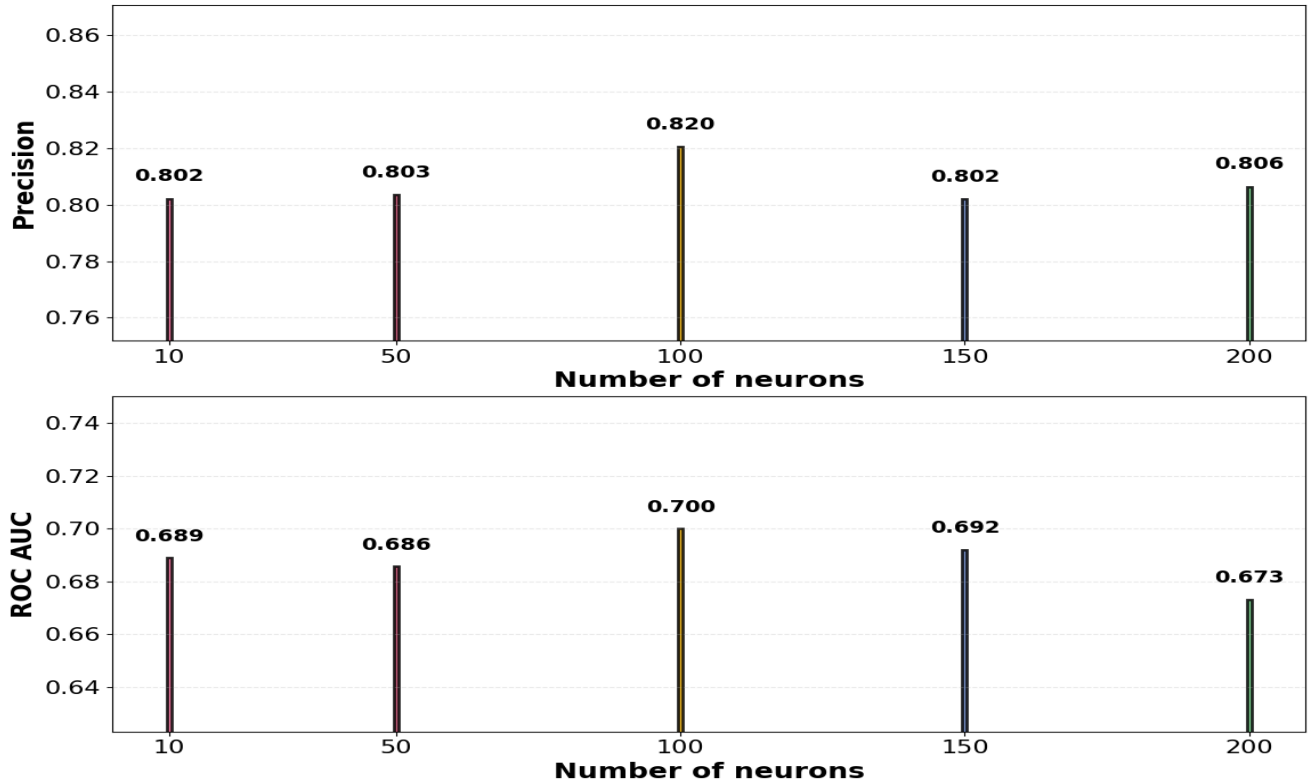


Рис. 2.20 – Порівняння Precision та AUC для різної кількості нейронів

Отримана матриця помилок, для створеної моделі на базі нейронних мереж (MLPClassifier) представлена на рис. 2.21.

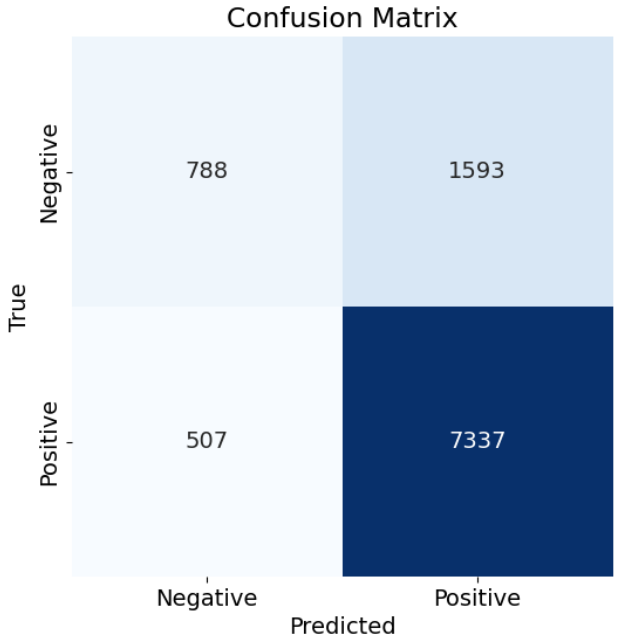


Рис. 2.21 – Матриця помилок для нейронних мереж.

Виходячи з отриманої матриці проведено розрахунок значень характеризуючих загальну точність класифікації, результати розрахунків наведені в табл. 2.19.

Таблиця 2.19

Розрахунки значень характеризуючих загальну точність

Метод	Точність	Чутли-вість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою (AUC)
Нейронна мережа	0.794	0.935	0.330	0.874	0.633	0.70

Щоб наглядно оцінити здатність моделі до правильної класифікації, було побудовано ROC-криву, яка представлена на рис. 2.22.

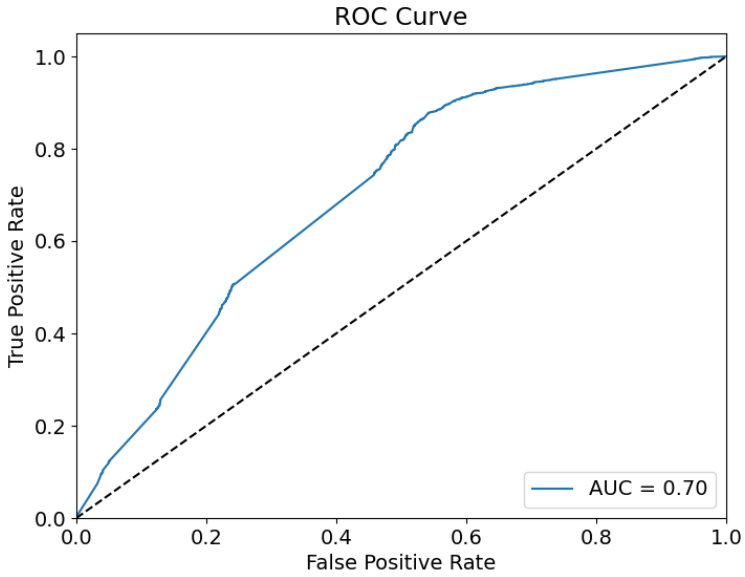


Рис. 2.22 – Графік ROC-кривої нейронних мереж(MLPClassifier).

Побудовану криву навчання (learning curve) для моделі на базі нейронних мереж представлено на рис. 2.23.

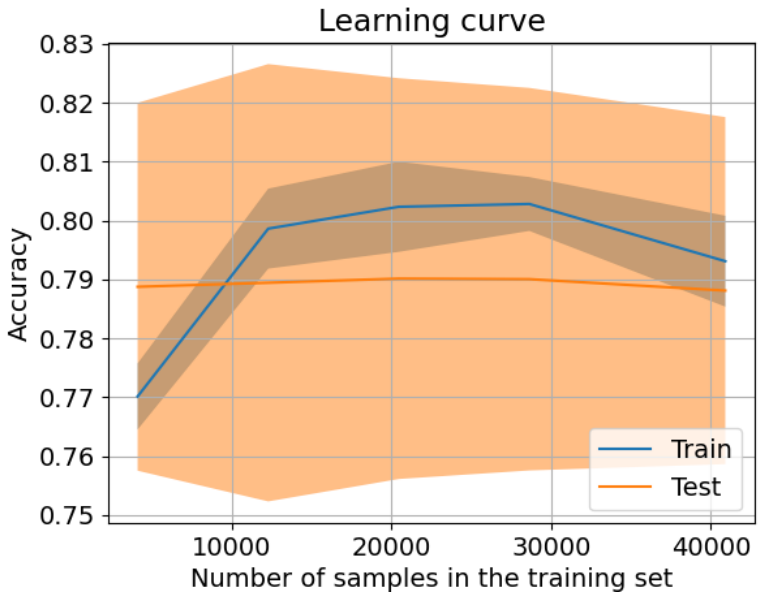


Рис. 2.23 – Графік кривої навчання моделі нейронних мереж (MLPClassifier).

На графіку видно, що при малій кількості зразків (< 3000) спостерігається розрив між тренувальною та тестовою точністю (~0.77-0.79), причому точність на тренувальних даних нижча, ніж на тестових, що є нетиповим і може вказувати на особливості розподілу даних. У діапазоні 2000-12000 зразків, тренувальна точність швидко зростає до ~0.80. Розрив між кривими зменшується. У діапазоні 12000-27000 зразків, тренувальна точність продовжує поступово зростати, досягаючи піку близько 0.803 при 28000 зразків. Тестова точність залишається відносно стабільною (близько 0.79). При кількості зразків, що перевищують 28000-29000 тренувальна крива починає злегка знижуватися, що може вказувати на появу перенавчання або включення більш складних, зашумлених даних. Оптимальний розмір набору даних, при яких модель досягає найкращого балансу між тренувальною та тестовою точністю складає ~28000 зразків. При збільшенні кількості зразків тренувальна та тестова точність наближаються одна до одної, що свідчить про зменшення варіативності та покращення генералізації моделі. Незважаючи на невелике падіння точності після 28000 зразків, графік не демонструє класичних ознак перенавчання (значне розходження кривих). Згідно графіка дана модель навряд зможе досягти точності вище 0.82 на наявних даних, незалежно від кількості зразків.

Побудовану криву калібрування (Calibration Curve) для моделі на базі нейронних мереж представлено на рис. 2.24.

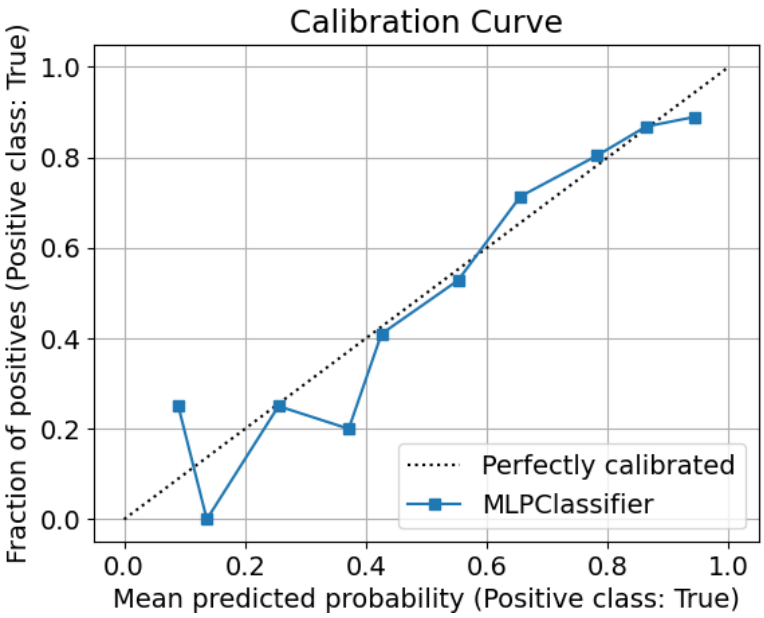


Рис. 2.24 – Графік кривої калібрування моделі нейронних мереж (MLPClassifier)

На графіку видно, що в діапазоні 0.1-0.15 спостерігається аномальне коливання вісі, коли модель передбачає ймовірність ~0.1, але фактична частка позитивних випадків становить ~0.24. У точці 0.2 відбувається різкий провал до нуля. Саме коли модель передбачає ймовірність ~0.2, фактична частка позитивних результатів близька до 0. У діапазоні 0.25-0.4 крива стрімко піднімається і наближається до ідеальної діагоналі. У точці 0.3 спостерігається майже ідеальне калібрування. У діапазоні 0.5-0.84 крива наближається до ідеальної діагоналі, демонструючи хороше калібрування моделі у високому діапазоні ймовірностей. Отримана модель є досить надійною, коли передбачає високі ймовірності позитивного класу. Передбачення з ймовірністю вище 0.6 можна впевнено використовувати для прийняття рішень. У високому діапазоні ймовірностей MLPClassifier показує краще калібрування, ніж багато інших класифікаторів таких як логістична регресія, Наївний Баєс та метод опрних векторів. Основні перевагами є можливість обробки великих обсягів даних, виконання складніих завдань при розпізнаванні складних шаблонів. Недоліком є схильність до перенавчання, особливо на малих наборах даних.

2.5. Аналіз результатів прогнозування успішності здобувачів освіти

Після реалізації усіх методів можна порівняти отримані результати прогнозування та визначити ті, що виявились найбільш точними. Усі п’ять методів були реалізовані на мові програмування Python із використанням бібліотеки Scikit-learn у середовищі розробки PyCharm. Навчання проходило на однаковому наборі даних, який був розділений на тренувальну та тестову вибірки. Для порівняльного оцінювання результатів класифікації було обрано наступні показники: точність, заблансована точність, чутливість, специфічність, площа під кривою (Area Under Curve, AUC), побудова ROC-кривої та графіків навчання і калібрування **[99]**.

Результат представлений в табл. 2.8 показав, що метод випадкового лісу має дещо кращі показники класифікації, ніж логістична регресія, Наївний Баєс та метод опорних векторів, хоча різниця і не значна.

Таблиця 2.8

Порівняння характеристик шести моделей класифікації

Метод	Точність	Чутли-вість	Специфіч-ність	F1-Score	Збалансована точність	Площа під кривою (AUC)
Логістична регресія	0.79	0.941	0.292	0.873	0.616	0.70
Метод опорних векторів	0.79	0.931	0.335	0.873	0.633	0.66
Випадковий ліс	0.80	0.925	0.391	0.877	0.658	0.73
Наївний Баєс	0.78	0.900	0.402	0.865	0.651	0.70
Нейронна мережа	0.79	0.935	0.331	0.874	0.633	0.70

Отримані результати дослідження показали, що всі розглянуті алгоритми демонструють близькі значення загальної точності в діапазоні 78-80%, що свідчить про схожу ефективність базових підходів машинного навчання на даному наборі ознак. Метод випадкового лісу продемонстрував найкращі комплексні показники серед усіх моделей, досягнувши найвищої загальної точності (80%), найвищого значення F1-Score (0.877), найкращої збалансованої точності (0.658) та найбільшої площі під ROC-кривою (AUC = 0.73), що підтверджує його перевагу як для загальної класифікації, так і для дискримінації між класами.

Водночас аналіз виявив спільну проблему для всіх моделей – низьку специфічність, значення якої коливаються від 0.292 до 0.402. Це означає, що моделі значно краще розпізнають студентів з високою успішністю (чутливість 90-94%), ніж студентів з низькими результатами. Найвищу специфічність продемонстрував Наївний Баєс (0.402), що робить його потенційно корисним у випадках, коли пріоритетним є виявлення студентів групи ризику. Логістична регресія показала найвищу чутливість (0.941), однак за рахунок найнижчої специфічності (0.292), що свідчить про її схильність класифікувати більшість випадків як позитивні. На основі отриманих результатів для подальшого прогнозування було обрано метод випадкового лісу та нейронні мережі як моделі з найкращим балансом між точністю та узагальнювальною здатністю. Отримані результати формують основу для практичного впровадження системи раннього виявлення студентів з ризиком низької академічної успішності, хоча подальша оптимізація моделей шляхом балансування класів та розширення набору ознак може суттєво покращити показники специфічності та загальної якості прогнозування.

2.5. Висновки до розділу 2

1. У межах розділу проаналізовано основні методи та алгоритми машинного навчання, що застосовуються для розв'язання задач класифікації, зокрема

логістичну регресію, метод опорних векторів, випадковий ліс, Наївний Баєс та нейронні мережі (MLPClassifier).

2. Усі реалізовані моделі класифікації були навчено та протестовано на однаковому наборі даних з використанням розподілу 80/20 для тренувальної та тестової вибірок, що забезпечує коректність та об'єктивність порівняльного аналізу результатів.
3. Порівняльний аналіз п’яти алгоритмів машинного навчання показав, що всі моделі демонструють близькі значення загальної точності в діапазоні 78-80%, що свідчить про схожу базову ефективність розглянутих методів на сформованому наборі ознак навчальної активності студентів.
4. Метод випадкового лісу (Random Forest) продемонстрував найкращі комплексні показники класифікації: найвищу загальну точність (80%), найвищий F1-Score (0.877), найкращу збалансовану точність (0.658) та найбільшу площу під ROC-кривою (AUC = 0.73), що обґрунтовує його вибір як основного алгоритму для прогнозування.
5. Усі досліджені моделі характеризуються високою чутливістю (90-94%) та низькою специфічністю (29-40%), що свідчить про їх здатність ефективно виявляти студентів з високою успішністю, але обмежену спроможність розпізнавати студентів групи ризику з низькими академічними результатами.
6. Для подальшого прогнозування академічної успішності обрано, як найбільш ефективні, алгоритми випадкового лісу та нейронні мережі із найкращим балансом між точністю класифікації та узагальнювальною здатністю.
7. Отримані результати формують основу для практичного впровадження прогнозування академічної успішності та потребують подальшого вдосконалення шляхом застосування методів балансування класів, оптимізації гіперпараметрів та розширення набору ознак для підвищення специфічності моделей.

РОЗДІЛ 3. ВИКОРИСТАННЯ ІНФОРМАЦІЇ ПРО РОБОТУ ЗДОБУВАЧІВ ОСВІТИ З ВІДЕОМАТЕРІАЛАМИ У ЗАДАЧАХ ПРОГНОЗУВАННЯ УСПІШНОСТІ

У третьому розділі на базі обраних методів машинного навчання створено моделі та виконано порівняльний аналіз точності їх прогнозування та ефективності із додаванням нового набору даних по взаємодії здобувачів із відеоматеріалами. Використано метод опорних векторів (SVM), логістичну регресію (Logistic Regression), класифікатор наївного Баєса (Naive Bayes), випадковий ліс (Random Forest) та нейронні мережі (Neural Network). Було розглянуто також недоліки та переваги їх використання на практиці. Визначено найбільш точні моделі по прогнозуванню успішності здобувачів, стратегію оцінювання моделей і виділення ознак. Проведено порівняння приросту точності прогнозування моделей та рівень впливу на модель такого виду даних. Результати розділу опубліковано у наукових працях **[100, 101, 104, 106, 107]**.

3.1. Формування наборів даних для навчання та тестування моделей

Набір даних для прогнозування взятий з бази даних Moodle та електронного журналу університету. У зборі даних приймали участь студенти Київського національного університету технологій та дизайну, які вивчали дисципліни: «Комп’ютерні технології та програмування» та «Проєктування інтерфейсу користувача». Всього у першому семестрі було 16 відео лекцій. Оцінки, відвідування здобувачів, а також дані про взаємодію з навчальними відеоматеріалами були експортовані у csv форматі. Для того щоб перевірити приріст точності прогнозування успішності за рахунок даних взаємодії з відео матеріалами, дані були розділені на два набори [**100**]. Із першого набору дані про взаємодії з відео матеріалами були вилучені, загальний вигляд представлено у табл. 3.1.

Таблиця 3.1

Фрагмент таблиці з даними для прогнозування першого набору

id	DiscMark	LectVisit	PractVisit	LabVisit	TotalVisit
…	…	…	…	…	…
2316	82	83	–1	83	83
2317	51	25	–1	33	29
…	…	…	…	…	…

де id – унікальний ідентифікатор користувача;

DiscMark – оцінка за дисципліну;

LectVisit – відсоток відвідуваності лекцій;

PractVisit – відсоток відвідуваності практичних занять;

LabVisit – відсоток відвідуваності лабораторних занять;

TotalVisit – загальний відсоток відвідуваності.

Другий набір містив усі дані про оцінки, відвідуваність та взаємодію здобувачів з навчальними відео матеріалами, загальний вигляд представлено в табл. 3.2.

Таблиця 3.2

Фрагмент таблиці з вхідними даними для прогнозування другого набору

id	Disc Mark	Lect Visit	Pract Visit	LabVisit	Total Visit	Duration	Play Count	Pause Count	Stop Count	Complete
…	…	…	…	…	…	…	…	…	…	…
2316	82	83	–1	83	83	27	1	1	0	1

2317	51	25	–1	33	29	0	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…

де id – унікальний ідентифікатор користувача;

DiscMark – оцінка за дисципліну;

LectVisit – відсоток відвідуваності лекцій;

PractVisit – відсоток відвідуваності практичних занять;

LabVisit – відсоток відвідуваності лабораторних занять;

TotalVisit – загальний відсоток відвідуваності;

Duration – тривалість перегляду відео (хвилин);

PlayCount – кількість натискань кнопки Play;

PauseCount – кількість натискань кнопки Pause;

StopCount – кількість натискань кнопки Stop;

Complete – статус завершення перегляду відео до кінця (1 – так, 0 – ні).

На рисунку 3.1 наведено ER-діаграму фрагмента реляційної бази даних Moodle LMS, використаної як джерело даних для прогнозування успішності. Діаграма відображає логічну структуру та зв’язки між основними сутностями системи. Таблиця mdl_user містить персональні дані здобувачів, mdl_course — інформацію про навчальні курси. Зарахування реалізовано через таблиці mdl_enrol і mdl_user_enrolments. Дані про завдання зберігаються в mdl_assign, елементи оцінювання – у mdl_grade_items, а результати оцінювання здобувачів – у mdl_grade_grades. Активність користувачів фіксується в журналі mdl_logstore_standard_log. Окремо виділено таблиці модуля відеоаналітики uvplayer, що містить метадані відеоматеріалів, та uvplayer_tracking, яка акумулює статистику взаємодії здобувачів з відеоконтентом. Зв’язки між таблицями реалізовано за

допомогою зовнішніх ключів із типом відношень «один-до-багатьох», що забезпечує референційну цілісність даних.Таблиця uvplayer містить метадані навчальних відеоматеріалів курсу, а таблиця uvplayer_tracking зберігає детальну статистику взаємодії здобувачів з відеоконтентом, що включає тривалість перегляду, кількість запусків, пауз та зупинок відео, статус завершення перегляду та швидкість відтворення. Зв'язки між таблицями реалізовано за допомогою зовнішніх ключів, що забезпечують референційну цілісність даних, а нотація «1:N» вказує на тип зв'язку «один-до-багатьох» між сутностями.

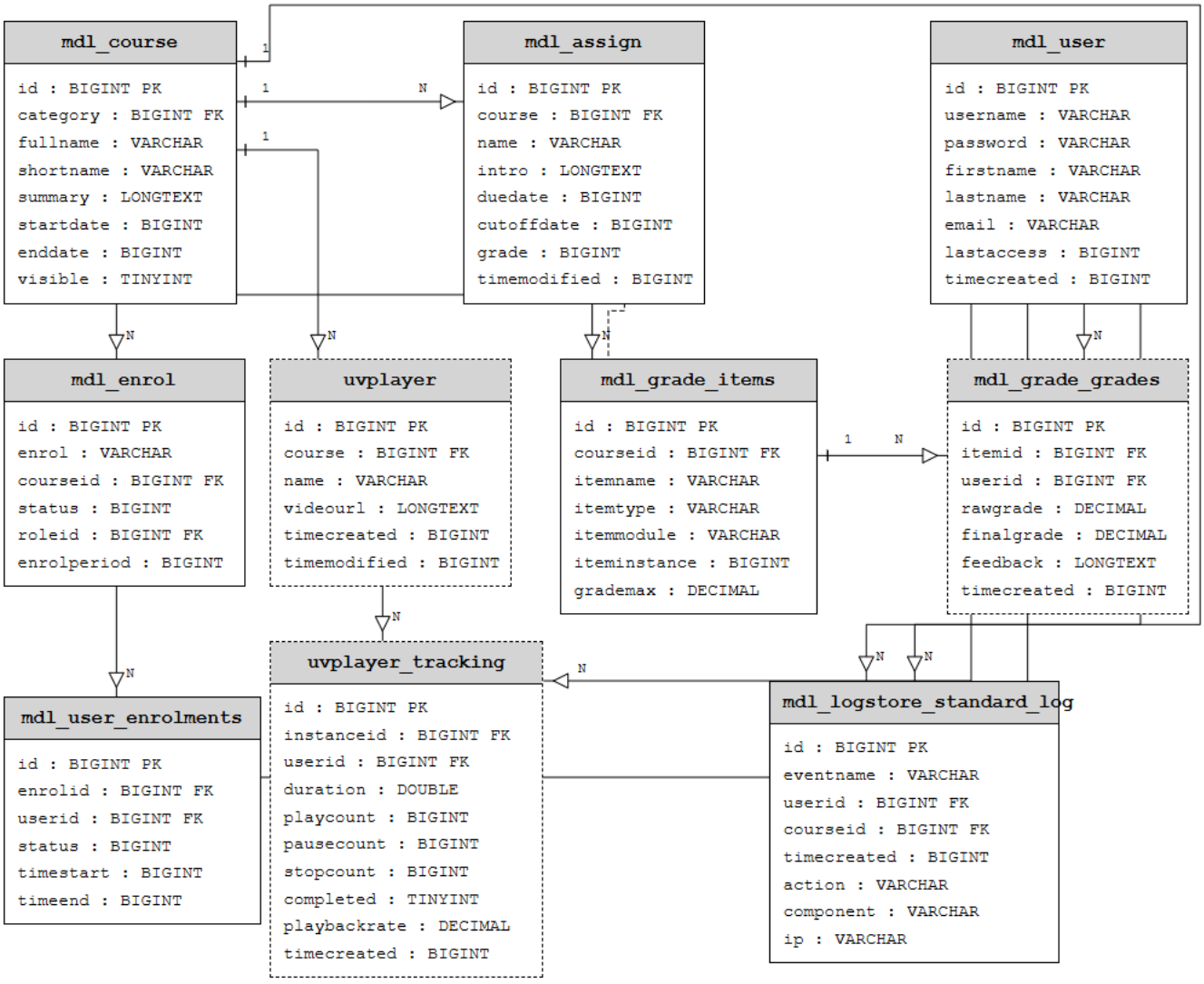


Рис. 3.1. – ER-діаграма фрагменту реляційної бази даних системи управління навчанням

Набір даних перед виконанням навчання моделей був розділений на тренувальну та тестову вибірки. Для того, щоб перевірити, наскільки добре модель, навчена на тренувальній вибірці, може передбачати класи нових даних. Обсяг даних взятих для обробки складав 2599 записів вибірок користувачів, які були розподілені у відношенні 520/2079. З яких тренувальна вибірка містила – 2079, а тестова – 520. Оцінка та перевірка якості моделей здійснювалась на основі тестової вибірки. У дослідженні використано вибірку обсягом близько 3000 записів навчальної активності студентів, що включають академічні показники, дані відвідуваності та поведінкові характеристики взаємодії з навчальними відеоматеріалами. Дані були розподілені на тренувальну та тестову вибірки у співвідношенні 80/20, що забезпечило коректне оцінювання узагальнювальної здатності моделей та мінімізувало вплив випадкових факторів під час навчання. У задачах машинного навчання для бінарної класифікації визначальним фактором є не лише абсолютний обсяг даних, а їх репрезентативність, інформативність ознакового простору та стабільність результатів на незалежній тестовій вибірці [**101**]. Використаний набір даних охоплює різні типи навчальної активності студентів, а отримані результати експериментів продемонстрували стабільність значень Accuracy, F1-score та ROC-AUC на тестовій вибірці без виражених ознак перенавчання, що підтверджує достатність використаного обсягу даних для навчання. Реалізацію, навчання та тестування моделей прогнозування здійснено в середовищі розробки PyCharm з використанням мови програмування Python. Для реалізації моделей використано бібліотеку scikit-learn. Для побудови графіків використано бібліотеки: seaborn та matplotlib **[102]**. Обробка табличних даних виконувалася з використанням бібліотеки pandas, а числових з використанням numpy **[103]**.

3.2. **Визначення достовірності прогнозів моделей навчених без даних про роботу здобувачів з відеоматеріалами**

На рисунку 3.2 представлено кореляційну матрицю, що демонструє взаємозв'язки між показниками відвідуваності навчальних занять та підсумковою оцінкою здобувачів (DiscMark). Матриця охоплює 5 ключових змінних: відвідуваність лекцій (LectVisit), практичних занять (PractVisit), лабораторних робіт (LabVisit), загальну відвідуваність (TotalVisit) та оцінку за дисципліну. Кольорова шкала візуалізації варіюється від синього кольору, що позначає негативні кореляції (-1.0), до червоного, що відповідає сильним позитивним зв'язкам (+1.0). Числові значення у кожній комірці матриці відображають коефіцієнт кореляції Пірсона між відповідними парами змінних. Дане представлення дозволяє оцінити вплив кожного типу занять на академічну успішність окремо від факторів взаємодії з електронними навчальними матеріалами, що є важливим для розуміння базового впливу традиційної форми навчання на результати здобувачів.

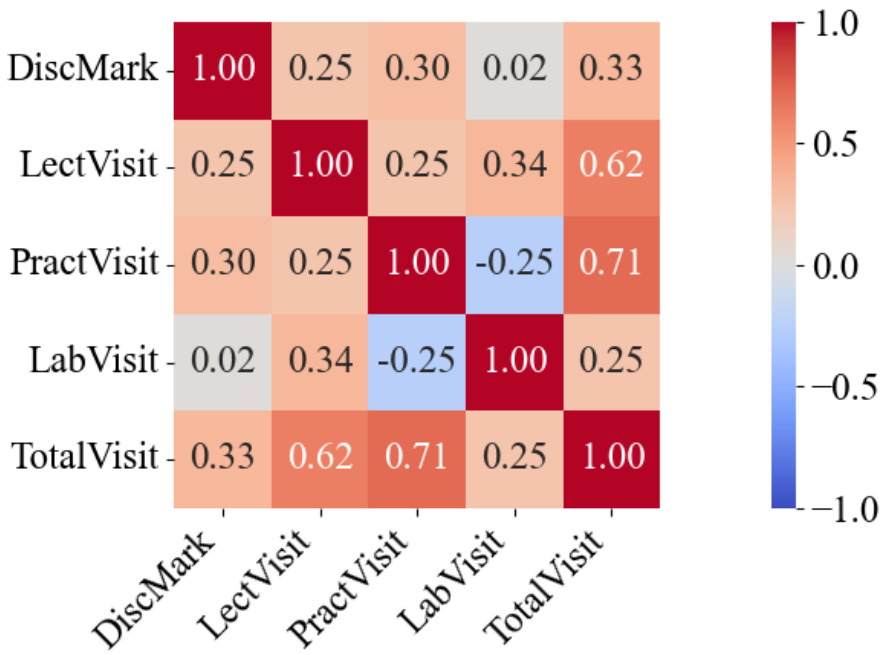
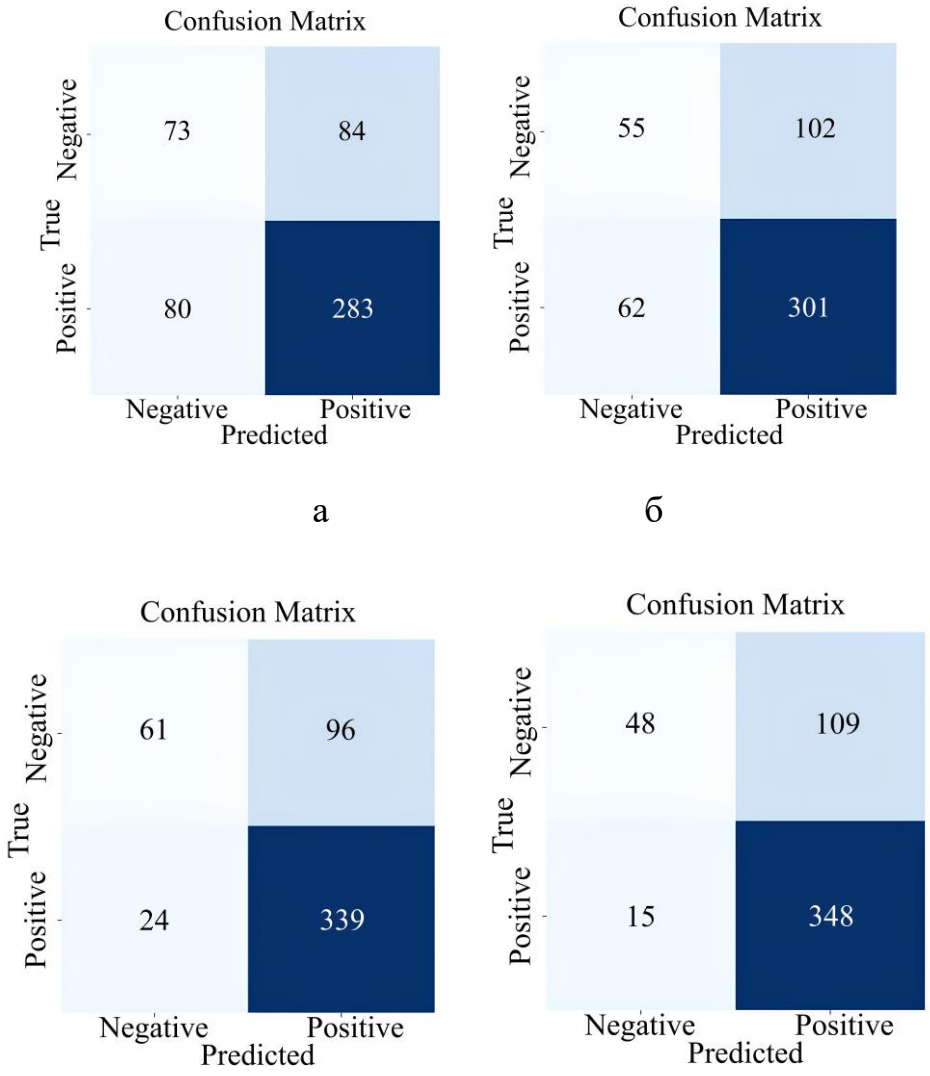


Рис. 3.2. – Кореляційна матриця ознак для першого набору даних

Результати аналізу показали, що загальна відвідуваність (TotalVisit = 0.33) та відвідування практичних занять (PractVisit = 0.30) мають найбільший вплив на оцінку

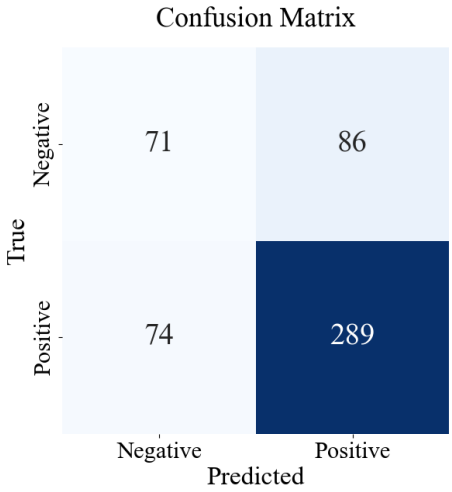
серед показників присутності. Водночас відвідуваність лабораторних робіт практично не корелює з підсумковою оцінкою (LabVisit = 0.02), що може свідчити про їх меншу вагу в системі оцінювання або високу однорідність відвідуваності серед здобувачів. Виявлена негативна кореляція між практичними та лабораторними заняттями (-0.25) вказує на існування різних стратегій навчання серед здобувачів.

Матриця помилок моделі дозволяє визначити, для скількох здобувачів прогнозування було виконано правильно. Зображення отриманих матриць помилок, моделей для першого набору даних, представлене на рис. 3.3.



а б

в г



д

Рис. 3.3. – Матриці помилок для створених моделей: а – Наївний Баєс; б – логістична регресія; в – випадковий ліс; г – нейронні мережі; д – метод опорних векторів.

Наївний Баєс демонструє найкращі показники по специфічності (найвищий показник True Negative – 73), але має відносно високі значення хибно позитивних (84) і хибно негативних (80) результатів, що вказує на можливі проблеми з точністю передбачення. Логістична регресія показує кращий результат у розпізнаванні позитивних класів (True Positive – 301), але при цьому має велику кількість хибно позитивних (102) результатів, що знижує її точність у передбаченні негативних класів. Випадковий ліс та нейронні мережі продемонстрували найкращі показники по чутливості, маючи найменшу кількість хибно негативних результатів (24 та 15) та найвищий True Positive (339 та 348), що свідчить про їх високу здатність правильно розпізнавати позитивні випадки. Метод опорних векторів демонструє високу здатність до розпізнавання позитивного класу (True Positive = 289), що є найвищим показником серед усіх комірок матриці, однак має значні труднощі з коректною класифікацією негативних випадків (True Negative = 71). Відносно високе значення хибно позитивних результатів (False Positive = 86) свідчить про схильність моделі до

надмірного прогнозування позитивного класу, що може бути спричинено дисбалансом класів у навчальній вибірці. Показник хибно негативних результатів (False Negative = 74) вказує на те, що частина позитивних випадків залишається невиявленою. Загалом модель демонструє помірну загальну точність (69.2%) з кращою продуктивністю для позитивного класу (Recall = 79.6%) порівняно з негативним (Specificity = 45.2%).

Випадковий ліс і нейронні мережі є найефективнішими в розпізнаванні позитивних класів з найменшою кількістю помилок на поточних даних. А наївний Баєс і логістична регресія мають більші похибки при передбаченні. Проте випадковий ліс має найнижчий серед розглянутих показник коректного розпізнавання негативних випадків (True Negative = 61), а значення специфічності становить лише 38.9%. Висока кількість хибно позитивних результатів (False Positive = 96) свідчить про виражену тенденцію моделі до надмірного прогнозування позитивного класу, тоді як низьке значення хибно негативних помилок (False Negative = 24) підтверджує ефективність у виявленні позитивних випадків. Загальна точність моделі становить 76.9%, що є вищим показником порівняно з попередньою моделлю, проте суттєвий дисбаланс між Recall та Specificity вказує на необхідність калібрування порогу класифікації для досягнення більш збалансованих результатів. У нейронної мережі найвище значення хибно позитивних результатів (False Positive = 109) свідчить про виражену схильність моделі класифікувати більшість випадків як позитивні. Загальна точність моделі становить 76.1%, проте низька специфічність робить модель менш придатною для задач, де важливим є коректне виявлення негативного класу.

Виходячи з отриманих матриць проведено розрахунок значень характеризуючих загальну точність класифікації та ефективність моделей, а саме: загальна точність, точність, збалансована точність, чутливість, специфічність, F1 Score, площа під кривою (AUC) та ROC-крива. Результати розрахунків наведені в табл. 3.3, 3.4.

Таблиця 3.3

Розрахунки значень характеризуючих загальну точність та ефективність

Алгоритм (класифікатор)	Загальна точність	Точність	Чутливість	Специфічність
Наївний Баєс	0.684	0.771	0.779	0.464
Логістична регресія	0.684	0.746	0.829	0.350
Випадковий ліс	0.769	0.779	0.933	0.388
Нейронні мережі	0.761	0.761	0.958	0.305
Опорних векторів	0.692	0.770	0.796	0.452

Таблиця 3.4

Розрахунки значень характеризуючих загальну точність та ефективність

Алгоритм (класифікатор)	Збалансована точність	Площа під кривою (AUC)	F1 Score
Наївний Баєс	0.622	0.607	0.775
Логістична регресія	0.589	0.627	0.785
Випадковий ліс	0.661	0.738	0.849
Нейронні мережі	0.632	0.681	0.848
Опорних векторів	0.668	0.663	0.783

Розрахунки показали, що алгоритм випадкового лісу краще виконує прогнозування успішності і має більшу точність 76,9 %. Отримане значення точності всього на 0,8 % більше ніж у алгоритму нейронних мереж. Проте порівняно з точністю наївного Баєса та логістичної регресії приріст в точності складає вже 8,5 %. Значення класів True і False отриманих при розрахунку Precision та F1 Score представлені в табл. 3.5.

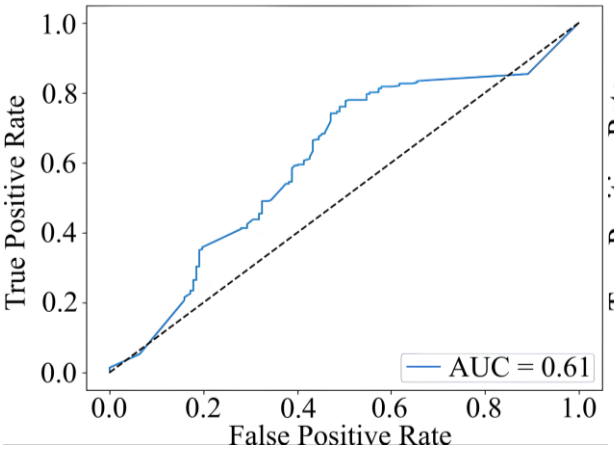
Таблиця 3.5

Значення класів True і False для Precision та F1 Score

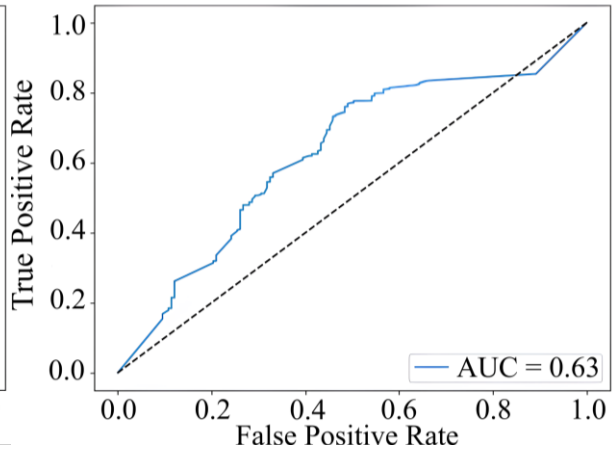
Алгоритм (класифікатор)	Precision	True	False	F1 Score	True	False
Наївний Баєс	0.771	0.77	0.48	0.775	0.78	0.47
Логістична регресія	0.746	0.75	0.47	0.785	0.79	0.40
Випадковий ліс	0.779	0.78	0.72	0.849	0.85	0.50
Нейронні мережі	0.761	0.76	0.76	0.848	0.85	0.44
Опорних векторів	0.770	0.77	0.49	0.783	0.78	0.47

Серед розглянутих алгоритмів класифікації найкращі результати показує випадковий ліс, який продемонстрував найвищу загальну точність (0.779) та збалансованість між класами. Цей алгоритм має високу точність для класу True (0.78) і суттєво кращу точність для класу False (0.72) порівняно з іншими моделями. Значення F1 Score для класу True (0.85) також є найвищим, що вказує на добру здатність правильно розпізнавати та не пропускати випадки цього класу. Хоча для класу False цей показник нижчий (0.50), що свідчить про деякі обмеження у розпізнаванні негативних випадків. Нейронні мережі показали високий рівень точності для обох класів (True і False по 0.76), демонструючи збалансованість та продуктивність. Проте значення F1 Score для класу False (0.44) вказує на те, що точність цих прогнозів все ще залишається проблемною. Логістична регресія та Наївний Баєс мали подібні результати з помірною загальною точністю (0.746 та 0.771 відповідно) і слабкою здатністю розпізнавати негативний клас (False), що підтверджується низькими значеннями точності та F1 Score для цього класу. По отриманим результатам можна зробити висновок, що випадковий ліс є найбільш придатним для застосування в умовах цієї задачі, оскільки він показує найкращий баланс між точністю для обох класів та загальною продуктивністю моделі. Цей

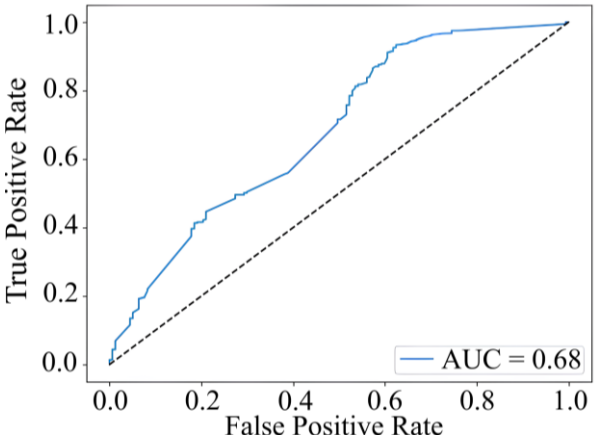
алгоритм забезпечує найбільш стабільні прогнози та мінімізує кількість хибно позитивних випадків, що робить його оптимальним вибором серед розглянутих методів. Щоб оцінити здатність моделей до правильної класифікації, враховуючи різні значення порогового значення було побудовано ROC-криві. Отримані графіки ROC-кривої моделей для першого сету даних, представлені на рис. 3.4.



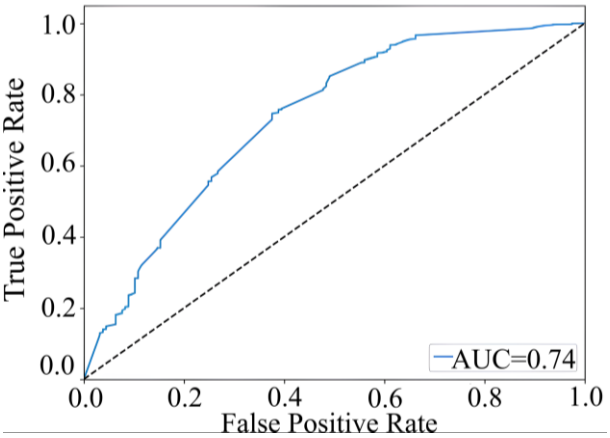
а



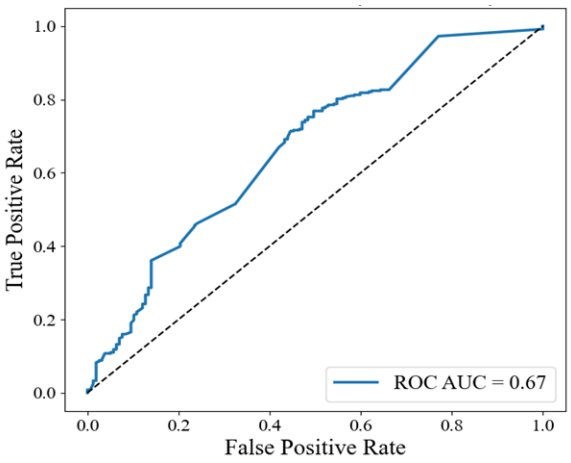
б



в



г



д

Рис. 3.4 – Побудовані графіки ROC-кривих для моделей: а – Наївний Баєс; б – логістична регресія; в – нейронні мережі; г – випадковий ліс; д – метод опорних векторів.

Побудовані ROC-криві для алгоритмів класифікації відображають різний рівень здатності моделей розрізняти між позитивними і негативними випадками. Випадковий ліс має найвищу площу під кривою 0.738, що свідчить про його хорошу загальну продуктивність у класифікації. Крива для випадкового лісу буде значно вище діагональної лінії, відзначаючи високу чутливість (0.933) і точність (0.779), а також досить високий F1 Score (0.849). Це свідчить про гарну здатність моделі виявляти позитивні випадки, хоча специфічність (0.388) залишається нижчою. Нейронні мережі також показують добрі результати з площею під кривою 0.681, що вказує на здатність до класифікації. Крива для нейронних мереж буде розташована вище діагональної лінії, з високою чутливістю (0.958) і точністю (0.761), що забезпечує високе значення F1 Score (0.848). Це показує високу ефективність у виявленні позитивних випадків, але з помірною специфічністю (0.305). Що може впливати на точність класифікації негативних випадків. Логістична регресія має площу під кривою (0.627), що є помірним показником ефективності моделі. Крива для логістичної регресії буде розташована трохи вище діагональної лінії, з високою чутливістю (0.829), помірною

специфічністю (0.350) та високим F1 Score (0.785). Це вказує на добру здатність моделі класифікувати позитивні випадки, але вона має певні труднощі з точністю класифікації негативних випадків. Наївний Баєс має найнижче значення площі під кривою 0.607, що вказує на помірну здатність розрізняти класи. Крива для наївного Баєса буде розташована ближче до діагональної лінії, відзначаючи добру точність (0.771) і чутливість (0.779), але з помірною специфічністю (0.464). Значення F1 Score для наївного Баєса (0.775) є найближчим до результатів логістичної регресії, що свідчить про відносно добре поєднання правильних прогнозів і зменшення помилок. У методу опорних векторів при досягненні 80% повноти виявлення позитивного класу (TPR = 0.8) модель допускає приблизно 40% хибно позитивних результатів (FPR = 0.4). Ступінчаста форма кривої у нижній частині графіка свідчить про обмежену кількість спостережень у тестовій вибірці. Згідно із загальноприйнятою інтерпретацією, значення AUC у діапазоні 0.6 – 0.7 вважається задовільним, але недостатнім для високоточної класифікації. Отримані результати вказують на необхідність вдосконалення моделі шляхом додавання нових ознак, оптимізації гіперпараметрів або застосування більш складних алгоритмів для підвищення якості прогнозування.

Випадковий ліс та нейронні мережі демонструють найкращі результати у розрізненні позитивних випадків. Логістична регресія також показує добру продуктивність, але з меншими значеннями AUC і специфічності, ніж випадковий ліс і нейронні мережі. Наївний Баєс має найнижчі результати, що свідчить про відносно низьку здатність моделі до класифікації в порівнянні з іншими методами.

3.3. Визначення достовірності прогнозів моделей навчених з даними про роботу здобувачів з відеоматеріалами

На рисунку 3.5 представлено кореляційну матрицю, яка відображає взаємозв'язки між показниками навчальної активності здобувачів та їх підсумковою оцінкою (DiscMark). Матриця включає 10 змінних: дані про відвідуваність різних типів занять (лекції, практики, лабораторні), показники взаємодії з відеоматеріалами (тривалість перегляду, кількість відтворень, пауз та зупинок), а також статус завершення курсу.

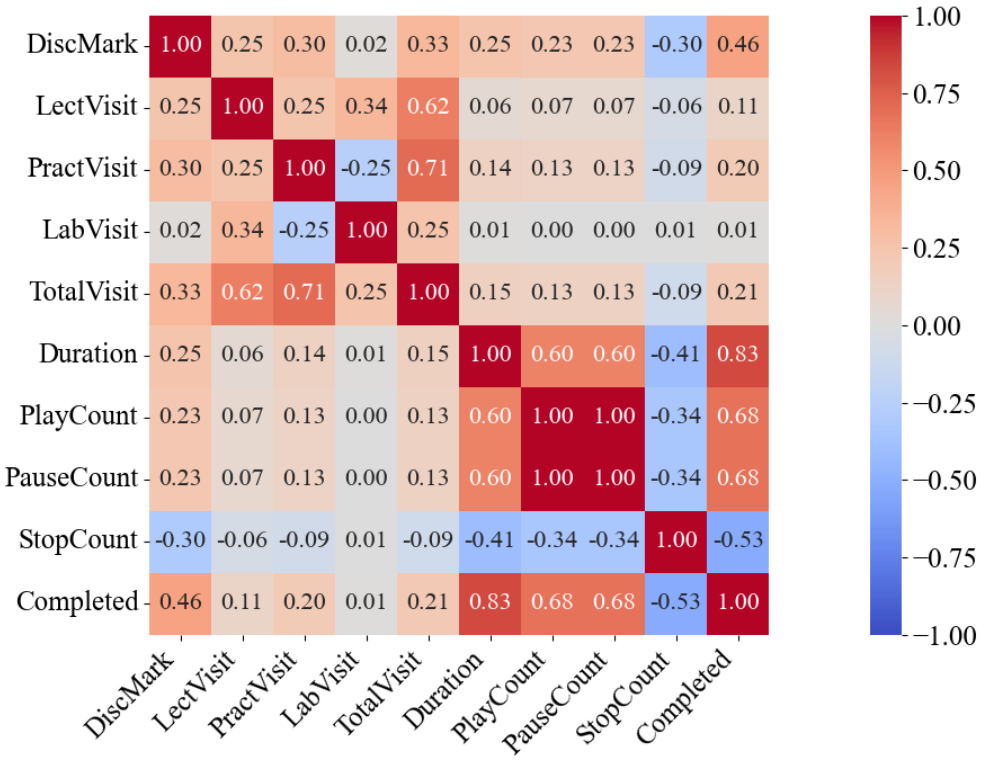


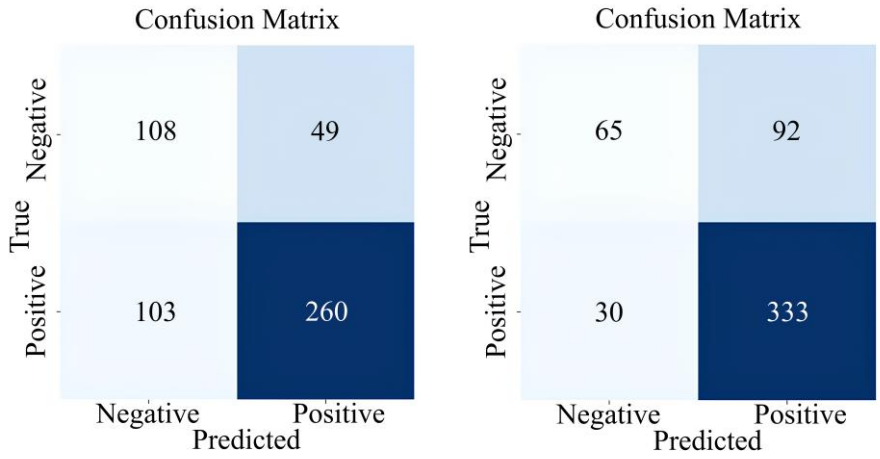
Рис. 3.5. – Матриця кореляцій ознак для другого набору даних

Кореляційний аналіз змінних засвідчує чітку структурованість показників навчальної активності та дозволяє виокремити змістовно різні кластери. Дискримінаційна оцінка (DiscMark) демонструє помірний негативний зв’язок із завершеністю перегляду відеоматеріалу в плеєрі (Completed, r = −0.46), що вказує на тенденцію зниження ймовірності повного завершення перегляду відео зі зростанням дискримінаційної оцінки, а також слабку негативну кореляцію з кількістю зупинок (StopCount, r = −0.30), що може свідчити про менш фрагментовану взаємодію з

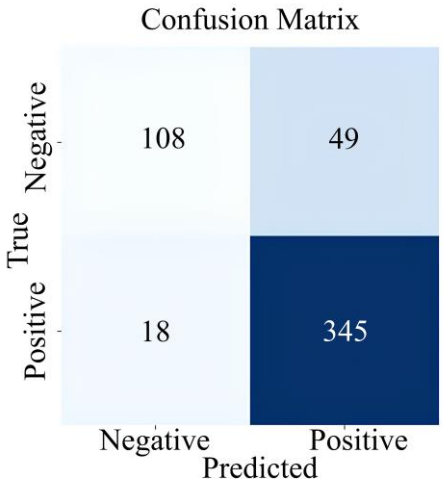
відеоматеріалом при вищих значеннях DiscMark, водночас кореляції з іншими змінними є близькими до нуля, що підтверджує її відносну незалежність. Показники відвідувань формують окремий блок. Відвідування лекцій (LectVisit) та практичних занять (PractVisit) мають сильні позитивні кореляції із загальною відвідуваністю (TotalVisit, r = 0.62 та r = 0.71 відповідно), що підтверджує їхній визначальний внесок у цей інтегральний показник, тоді як відвідування лабораторних занять (LabVisit) загалом не пов’язане з іншими змінними та лише слабко негативно корелює з PractVisit (r = −0.25), що може вказувати на частковий ефект заміщення між цими видами занять. Усі показники відвідувань демонструють слабкі або нульові кореляції з метриками відео-активності, що свідчить про незалежність очної навчальної активності від поведінки користувачів у відеоплеєрі. Натомість змінні відео-активності утворюють тісно пов’язаний кластер. Тривалість перегляду відеоматеріалу (Duration) має дуже сильний позитивний зв’язок із завершенням перегляду відео в плеєрі (Completed, r = 0.83), що вказує на прямий зв’язок між фактичною тривалістю взаємодії з відеоконтентом і доведенням перегляду до кінця. Крім того, Duration демонструє помірні позитивні кореляції з кількістю відтворень (PlayCount) та кількістю пауз (PauseCount) (обидві r = 0.60), а також помірний негативний зв’язок із кількістю зупинок (StopCount, r = −0.41). Кількість відтворень (PlayCount) і кількість пауз (PauseCount) характеризуються ідеальною позитивною кореляцією (r = 1.00), що вказує на лінійну залежність між цими змінними та можливу мультиколінеарність. Обидва показники мають сильні позитивні кореляції із завершенням перегляду відеоматеріалу в плеєрі (Completed, r = 0.68), що підтверджує роль активної взаємодії з відео у доведенні перегляду до кінця, водночас демонструючи помірні негативні кореляції з кількістю зупинок (StopCount, r = −0.34). Кількість зупинок (StopCount), своєю чергою, має помірний негативний зв’язок із завершенням перегляду відеоматеріалу в плеєрі (Completed, r = −0.53), а також із тривалістю перегляду (Duration, r = −0.41), що свідчить про зниження безперервності та ефективності взаємодії з відеоконтентом за умов частих переривань. Загалом кореляційна матриця

виявляє наявність двох відносно автономних груп змінних, показників відвідувань та показників відео-активності, між якими спостерігаються слабкі взаємозв’язки, тоді як завершення перегляду відеоматеріалу в плеєрі найбільшою мірою пов’язане з тривалістю та інтенсивністю взаємодії користувачів із відеоконтентом. Виявлена структурна незалежність між показниками відвідувань і метриками відео-активності свідчить про те, що дані перегляду відеоматеріалів відображають окремий вимір навчальної залученості, який не фіксується традиційними показниками очної участі. Це дозволяє розглядати показники відео-активності, як потенційно інформативні предиктори у моделях прогнозування навчальної успішності, здатні підвищувати їх прогностичну точність.

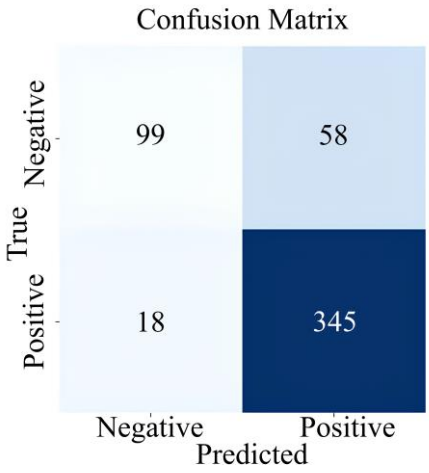
Зображення отриманих матриць помилок, моделей для другого набору даних, представлене на рис. 3.6.



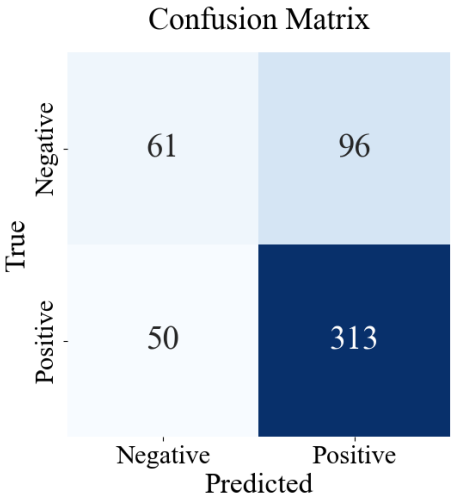
а б



в



г



д

Рис. 3.6 – Матриці помилок для створених моделей: а – Наївний Баєс; б – логістична регресія; в – випадковий ліс; г – нейронні мережі; д – метод опорних векторів.

Виходячи з отриманих матриць проведено розрахунок значень характеризуючих загальну точність класифікації, а саме: загальна точність, точність, збалансована точність, чутливість, специфічність, F1 Score, площа під кривою (AUC) та ROC-крива. Результати розрахунків наведені в табл. 3.6, 3.7

Таблиця 3.6

Розрахунки значень характеризуючих загальну точність та ефективність

Алгоритм (класифікатор)	Загальна точність	Точність	Чутливість	Специфічність
Наївний Баєс	0.707	0.841	0.716	0.687
Логістична регресія	0.765	0.783	0.917	0.414
Випадковий ліс	0.871	0.875	0.950	0.687
Нейронні мережі	0.853	0.856	0.950	0.630
Опорні вектори	0.719	0.765	0.862	0.388

Таблиця 3.7

Розрахунки значень характеризуючих загальну точність та ефективність

Алгоритм (класифікатор)	Збалансована точність	Площа під кривою (AUC)	F1 Score
Наївний Баєс	0.702	0.776	0.773
Логістична регресія	0.665	0.779	0.845
Випадковий ліс	0.819	0.875	0.911
Нейронні мережі	0.790	0.859	0.900
Опорні вектори	0.625	0.716	0.810

Розрахунки показали, що алгоритм випадкового лісу краще виконує прогнозування успішності на вхідних даних і має більшу точність 87,1 %. Проте порівняно з точністю наївного Баєса, логістичної регресії та методу опорних векторів приріст в точності складає вже 16,4 %, 10,6 % та 11%. Значення класів True і False отриманих при розрахунку Precision та F1 Score представлені в табл. 3.8.

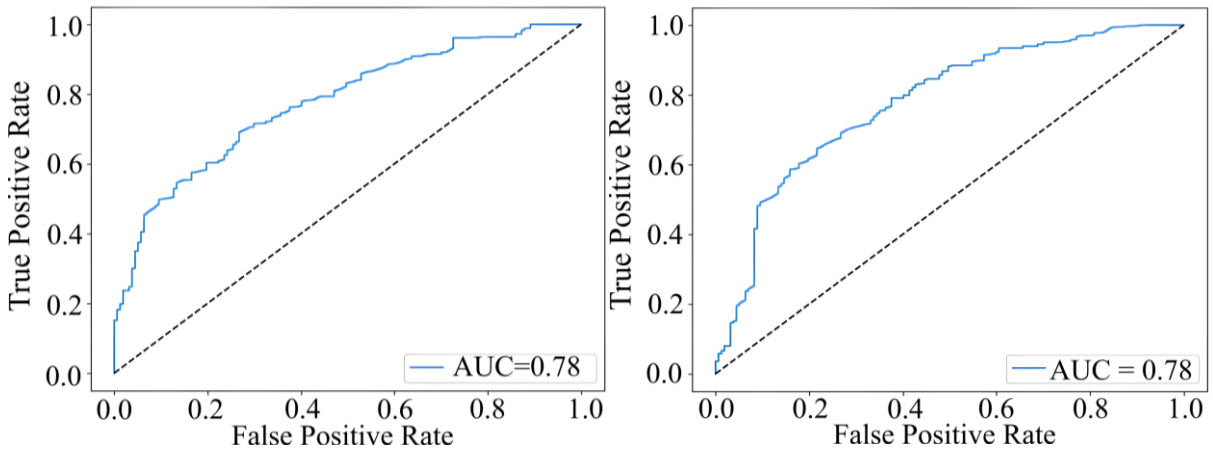
Таблиця 3.8

Значення класів True і False для Precision та F1 Score

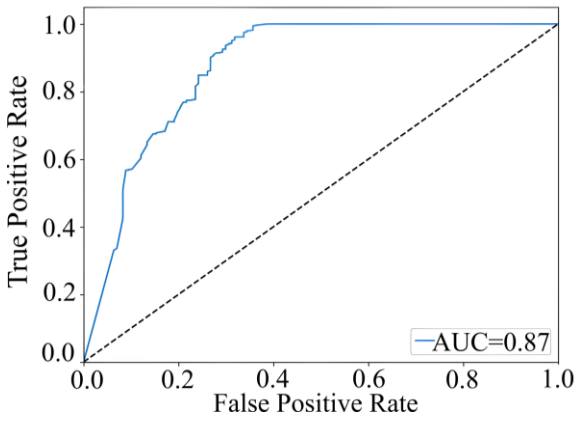
Алгоритм (класифікатор)	Precision	True	False	F1 Score	True	False
Наївний Баєс	0.841	0.84	0.51	0.773	0.77	0.59
Логістична регресія	0.783	0.78	0.68	0.845	0.85	0.52
Випадковий ліс	0.875	0.88	0.86	0.911	0.91	0.76
Нейронні мережі	0.856	0.86	0.85	0.900	0.90	0.72
Опорні вектори	0.765	0.77	0.55	0.810	0.81	0.46

Серед алгоритмів класифікації, найкращі результати продемонстрував випадковий ліс, який має найкращу загальну точність (0.875) і збалансованість між класами. Випадковий ліс показав високу точність для класу True (0.88) та досить високу точність для класу False (0.86). Це забезпечує хороший баланс між виявленням позитивних і негативних випадків. F1 Score для класу True (0.911) і класу False (0.91) є найвищими серед усіх розглянутих алгоритмів. Це свідчить про відмінну здатність моделі правильно класифікувати як позитивні, так і негативні випадки. Нейронні мережі теж показали високі результати з точністю для класу True (0.86) і класу False (0.85), але їх F1 Score для класу True (0.900) і False (0.90) трохи нижчий у порівнянні з випадковим лісом, хоча все ще високий. З цього можна зробити висновок, що нейронні мережі добре справляються з класифікацією обох класів, проте мають невелику відмінність у загальній продуктивності. Наївний Баєс і логістична регресія показали нижчі результати. Наївний Баєс мав точність 0.841, але високу False Positive Rate (0.51), що вказує на часті помилки в класифікації негативних випадків. Логістична регресія також має помірну точність (0.783), але її результати є менш збалансованими. Низьке значення F1 Score для класу False (0.52) свідчить про труднощі в точній класифікації негативних випадків. Метод опорних векторів демонструє специфічність 38.9% (TN = 61), що є недостатнім для надійного розпізнавання негативного класу. Високе значення FP = 96 свідчить про тенденцію до

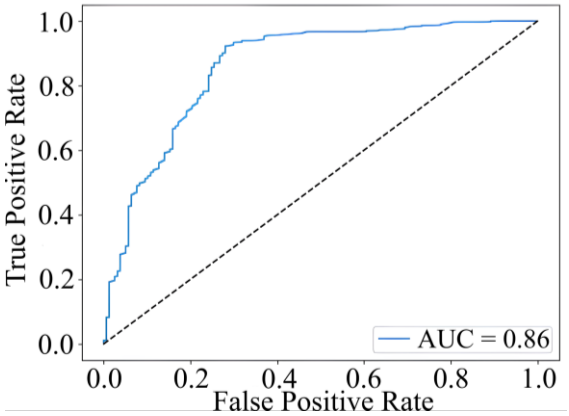
надмірного прогнозування позитивного класу, а FN = 50 вказує на втрату частини позитивних випадків. Загальна точність становить 71.9% –найнижчий показник серед розглянутих моделей. Випадковий ліс є найефективнішим серед розглянутих алгоритмів, досягаючи найвищої точності 87.1% та F1-score 91.1%. Модель забезпечує TP = 345 та TN = 108, демонструючи Recall = 95.0% та найвищу специфічність 68.8%. Низькі значення помилок (FN = 18, FP = 49) підтверджують збалансовану роботу з обома класами, що робить модель найбільш придатною для прогнозування академічної успішності. Нейронна мережа поступається випадковому лісу за специфічністю та загальною точністю, однак залишається ефективним алгоритмом для систем раннього виявлення здобувачів з ризиком низької успішності. Щоб оцінити здатність моделей до правильної класифікації, враховуючи різні значення порогового значення було побудовано ROC-криві, отримані графіки представлені на рис. 3.7.



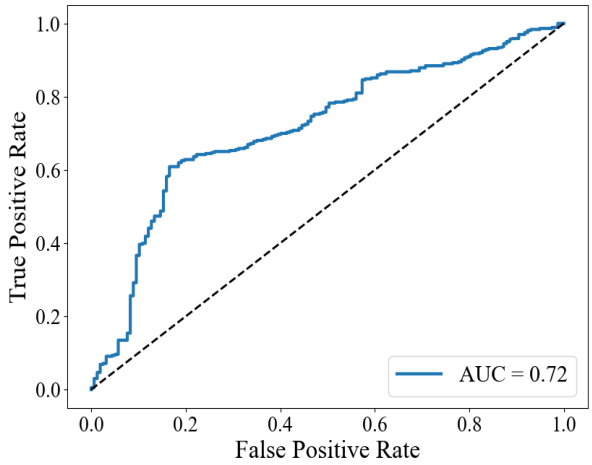
а б



в



г



д

Рис. 3.7 – Побудовані графіки ROC-кривих для моделей: а – Наївний Баєс; б – логістична регресія; в – випадковий ліс; г – нейронні мережі; д – метод опорних векторів.

Побудовані ROC-криві для розглянутих алгоритмів класифікації демонструють високий рівень ефективності в розрізненні між позитивними і негативними випадками. Випадковий ліс досягає найвищої площі під кривою (0.875), що вказує на його відмінну здатність класифікувати обидва класи. Крива для випадкового лісу розташовується значно вище діагональної лінії, відзначаючи високу точність (0.875) і чутливість (0.950). Високе значення F1 Score (0.911) свідчить про сильну ефективність у розпізнаванні як позитивних, так і негативних випадків. Нейронні мережі теж

демонструють високий рівень продуктивності з площею під кривою (0.859) показуючи добру здатність до класифікації. Крива для нейронних мереж розташована вище діагональної лінії, з високою чутливістю (0.950) і точністю (0.856), що забезпечує високий F1 Score (0.900). Це свідчить про ефективну роботу моделі в розпізнаванні позитивних випадків і відносно хорошу у розрізненні негативних. Логістична регресія має площу під кривою (0.779), значення є меншим ніж у випадкового лісу і нейронних мереж, але все ще показує добру здатність до класифікації. Крива для логістичної регресії буде розташована вище діагональної лінії, з високою чутливістю (0.917) і помірною специфічністю (0.414), а також F1 Score (0.845), що свідчить про добру збалансованість, особливо для позитивних випадків. Наївний Баєс має найнижче значення площі під кривою (0.776), що вказує на помірну здатність класифікації. Крива для наївного Баєса буде розташовуватися ближче до діагональної лінії, відзначаючи добру точність (0.841) і чутливість (0.716), але з менш високою специфічністю (0.687). F1 Score для наївного Баєса (0.773) є найнижчим серед розглянутих алгоритмів, що вказує на меншу ефективність у поєднанні правильних прогнозів і мінімізації помилок. Графік кривої метод опорних векторів показав, що забезпечує приблизно 60% повноти виявлення позитивного класу (TPR) при 20% хибно позитивних результатів (FPR), а при збільшенні FPR до 40% досягається близько 80% TPR. Крива розташована значно вище діагоналі випадкового класифікатора на всьому діапазоні значень, що підтверджує практичну цінність моделі. Більш плавна форма кривої порівняно з попереднім графіком вказує на більш стабільну поведінку моделі при зміні порогу класифікації. Отримане значення AUC свідчить про прийнятну якість прогнозування, однак для досягнення відмінних результатів (AUC > 0.8) рекомендується подальша оптимізація моделі або використання ансамблевих методів.

Випадковий ліс та нейронні мережі демонструють найкращу загальну продуктивність, з високими значеннями AUC та F1 Score, тоді як логістична регресія

також показує добрі результати, але з меншою ефективністю порівняно з випадковим лісом. Наївний Баєс та метод опорних векторів має найнижчі результати серед усіх моделей, особливо, у поєднанні точності і чутливості. Даний результат є досить доброю початковою точкою, але в процесі подальшого дослідження може знадобитися додаткове вдосконалення для підвищення даних показників. Випадковий ліс є ансамблевим методом, який зазвичай працює краще в тих випадках, коли взаємозв'язки між ознаками та вихідними класами більш складні, нелінійні або коли є багато ознак. Він може автоматично враховувати важливість ознак і робити кращі передбачення, ніж лінійні моделі на складних даних.

3.5. Аналіз впливу даних про роботу здобувачів освіти на достовірність прогнозування успішності

Для порівняння приросту точності прогнозування моделей шляхом додавання даних по взаємодії з відеоматеріалами було використано два набори даних. В першому наборі були тільки дані по відвідуваності та оцінкам. В другому наборі додались дані по взаємодії з навчальними відеоматеріалами. Для кожного з наборів було побудовано моделі для прогнозування успішності з алгоритмами: логістичної регресії, наївного Баєса, випадкового лісу та нейронних мереж (MLPClassifier). І розраховані характеристики, що відповідають за точність прогнозування. Для першого набору даних розрахунки представлені в табл. 3.3 і 3.4, для другого набору даних розрахунки представлені в табл. 3.6 і 3.7. Значення класів отриманих при розрахунку точності та F1 Score представлені в табл. 3.5 для першого набору даних, та в табл. 3.8 для другого набору даних. Побудовані кореляційні матриці ознак для першого та другого набору даних представлені на рис. 3.2 та рис. 3.5. Матриці помилок для створених моделей прогнозування по першому набору даних представлено на рис. 3.3, а побудовані графіки ROC-кривих представлено на рис. 3.4. Матриці помилок для створених моделей по другому набору даних представлено на рис. 3.6, графіки ROC-кривих представлено на рис. 3.7. Отримані результати підтвердили гіпотезу, що додаткові

дані підвищують точність прогнозування. Визначення отриманого приросту точності прогнозування успішності серед моделей між двома наборами даних представлено в табл. 3.9, 3.10.

Таблиця 3.9

Результат отриманого приросту точності прогнозування

Алгоритм (класифікатор)	Загальна точність	Точність	Чутливість	Специфічність
Наївний Баєс	+2.3 %	+7%	– 6.3 %	+22.3 %
Логістична регресія	+8.1 %	+3.7%	+8.8 %	+6.4 %
Випадковий ліс	+10.2 %	+9.6%	+1.7 %	+29.9 %
Нейронна мережа	+9.2 %	+9.5%	– 0.8 %	+32.5 %
Метод опорних векторів	+3.9 %	−0.6 %	+8.3 %	−14.2 %

Таблиця 3.10

Результат отриманого приросту точності прогнозування

Алгоритм (класифікатор)	Збалансована точність	Площа під кривою (AUC)	F1 Score
Наївний Баєс	+8 %	+16.9 %	– 0.002
Логістична регресія	+7.6 %	+15.2 %	+0.006
Випадковий ліс	+15.8 %	+13.7 %	+0.062
Нейронна мережа	+15.8 %	+17.8 %	+0.052
Метод опорних векторів	−6.4 %	+8 %	+0.027

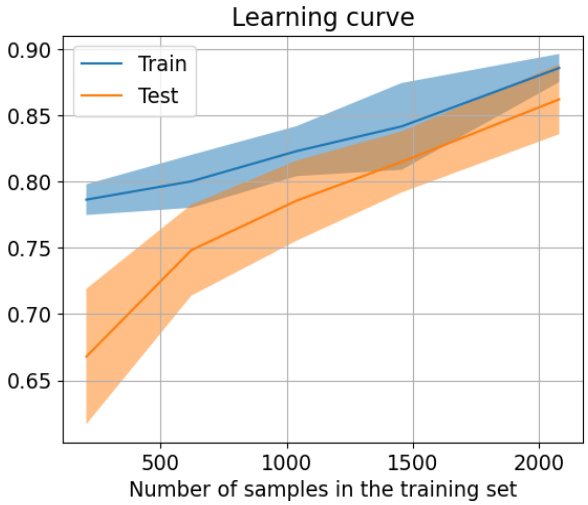
Найвищій приріст по точності з різницею в 1.8% показали моделі з алгоритмами: випадкового лісу – 87.1% та нейронних мереж – 85.3%. Приріст точності склав більше 10%, збалансована точність збільшилась на 15%, а загальна ефективність виражена площею під кривою (AUC) збільшилась на 14%. Дослідження показало, що моделі з алгоритмами випадкового лісу та нейронних мереж дають кращу точність прогнозування 87.1% та 85.3% на наявних даних в порівнянні з алгоритмами наївного Баєса та логістичної регресії точність яких склала 70.7% та 76.5% відповідно. Найменший приріст точності прогнозування 2.3 % у моделі з алгоритмом наївного Баєса. Порівняння приросту показників класів True і False для Precision та F1 Score між двома наборами даних, що представлені в таблицях 3.5 та 3.8 для кожного з методів класифікації представлено в табл. 3.11.

Таблиця 3.11

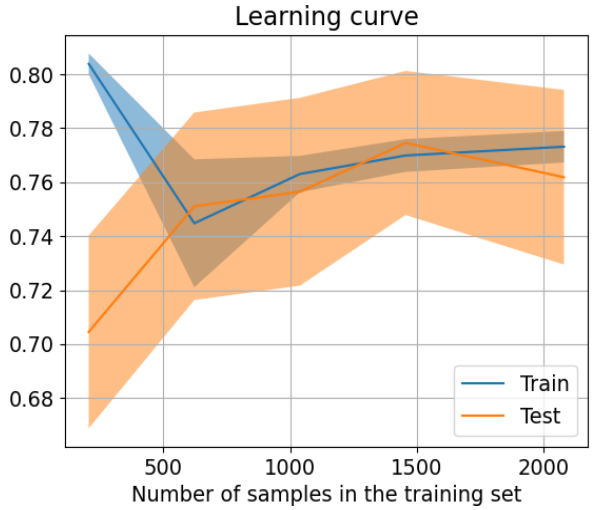
Результат отриманого приросту класів True і False для Precision та F1 Score

Алгоритм (класифікатор)	Приріст Точності True	Приріст Точності False	Приріст F1 Score True	Приріст F1 Score False	Приріст Точності True (%)	Приріст Точності False (%)
Наївний Баєс	+0.071	+0.030	-0.002	+0.110	+9.2%	+6.2%
Логістична регресія	+0.033	+0.210	+0.039	+0.390	+4.4%	+44.7%
Випадковий ліс	+0.095	+0.140	+0.062	+0.300	+12.2%	+19.4%
Нейронні мережі	+0.100	+0.110	+0.048	+0.280	+13.2%	+14.5%
Опорні вектори	-0.005	+0.027	+0.03	-0.01	0%	+12.2%

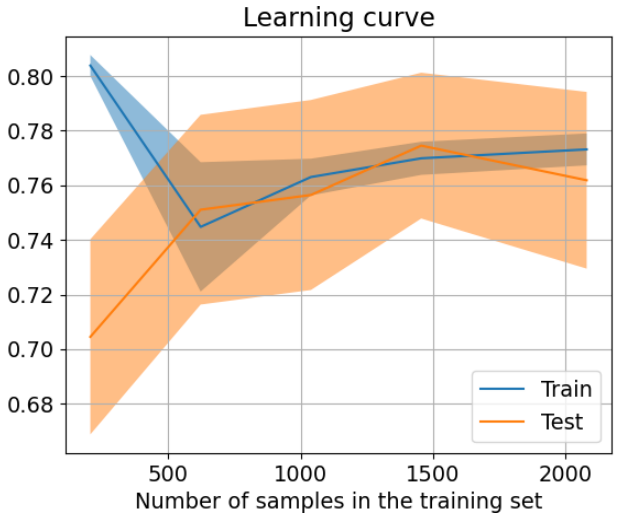
На основі аналізу приросту точності та F1 score можна зробити висновок, що найбільший приріст показали алгоритми випадкового лісу та нейронних мереж (MLPClassifier), що підтверджує їх високу ефективність для задач класифікації, особливо у випадках, коли потрібно правильно класифікувати обидва класи, зокрема клас False. випадковий ліс продемонстрував значний приріст точності для класу False (+19.4%) та F1 score для цього класу (+30%), що вказує на його здатність ефективно працювати з незначними класами. Нейронні мережі також показали хороший приріст точності та F1 score для обох класів, особливо для False. Логістична регресія та Наївний Баєс показали менші прирости, що робить їх менш ефективними порівняно з іншими методами. Для оцінки здатності моделей до навчання та узагальнення на другому наборі даних було побудовано криві навчання. На рис. 3.8 відображено залежність точності моделі на тренувальній та тестовій вибірках від кількості навчальних прикладів.


Learning curve
Train
Test
0.90
0.85
0.80
0.75
0.70
0.65
500
1000
1500
2000
Number of samples in the training set

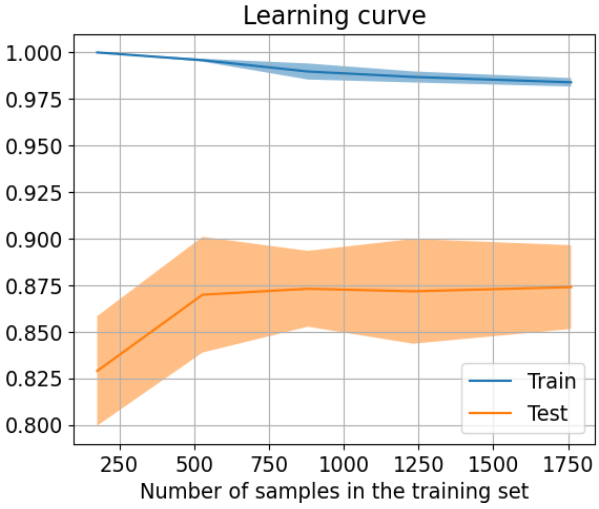
а


Learning curve
0.80
0.78
0.76
0.74
0.72
0.70
0.68
Train
Test
500
1000
1500
2000
Number of samples in the training set

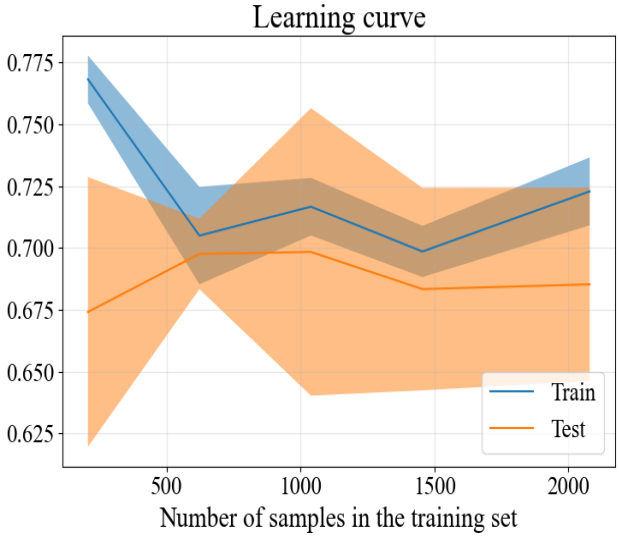
б


Learning curve
0.80
0.78
0.76
0.74
0.72
0.70
0.68
Train
Test
500
1000
1500
2000
Number of samples in the training set

в


Learning curve
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
Train
Test
250
500
750
1000
1250
1500
1750
Number of samples in the training set

г


Learning curve
0.775
0.750
0.725
0.700
0.675
0.650
0.625
Train
Test
500
1000
1500
2000
Number of samples in the training set

……Д

Рис. 3.8 – Побудовані графіки кривих навчання для другого набору даних: а – логістична регресія; б – Наївний Баєс; в – нейронні мережі; г – випадковий ліс; д – метод опорних векторів.

На основі проведеного аналізу видно, що логістична регресія та наївний Баєсівський класифікатор демонструють схожу поведінку як класичні лінійні моделі з низькою дисперсією та високою здатністю до узагальнення. В обох випадках початкова точність становить приблизно 0.7 і поступово зростає до 0.77-0.78 зі збільшенням розміру тренувального набору. Ключовим фактором є мінімальний розрив між тренувальною та тестовою точністю, що свідчить про відсутність перенавчання. Така стабільність пояснюється регуляризаційними властивостями відповідних моделей. Але обмежена складність обох моделей знижує їхню ефективність у задачах зі складною нелінійною структурою або сильно корельованими ознаками. Метод опорних векторів показує, що модель демонструє високу точність на тренувальній вибірці (~0.775), однак значно нижчу на тестовій (~0.68), що є ознакою перенавчання. Зі збільшенням кількості тренувальних зразків спостерігається характерна тенденція, що точність на тренувальних даних поступово знижується. Тоді як точність на тестових даних коливається, але загалом наближається до тренувальної. При максимальному обсязі вибірки (~2000 зразків) обидві криві сходяться в діапазоні 0.70–0.73, що свідчить про покращення узагальнювальної здатності моделі. Зближення кривих навчання та тестування при збільшенні обсягу даних підтверджує відсутність критичного перенавчання та свідчить про потенційну користь від подальшого розширення навчальної вибірки для підвищення стабільності та точності прогнозування. Нейронна мережа демонструє найвищу фінальну точність (0.87-0.90), починаючи з нижчих значень (0.65) через складність моделі та потребу в достатній кількості даних. Розрив між тренувальною та тестовою точністю (0.02-0.04) вказує на помірне, але прийнятне перенавчання. Випадковий ліс показує найшвидше початкове зростання точності (до 0.82 на малих вибірках) та досягає майже повної точності (0.97)

на тренувальному наборі. Однак значний розрив (~0.15) з тестовою точністю свідчить про істотне перенавчання через високу дисперсію моделі та тенденцію глибоких дерев до запам'ятовування даних. Результати підтверджують, що застосування нейронних мереж (MLPClassifier), випадкового лісу та додавання даних про взаємодію з відеоматеріалами позитивно впливає на точність прогнозування.

3.6. Розробка 2-рівневої стекінгової моделі та дослідження її характеристик

Однією з ключових переваг ансамблевих методів є можливість поєднання моделей різної природи, які по-різному апроксимують залежність між вхідними ознаками та цільовою змінною. У запропонованій архітектурі стекінгу використовуються три базові алгоритми: лінійна регресія (LR), нейронна мережа (NN) та випадковий ліс (RF). Кожна з моделей характеризується власним механізмом навчання та різною структурою похибок. Для перевірки гіпотези про підвищення точності прогнозування та зменшення похибки узагальнення у дослідженні за рахунок стекінгу моделей було побудовано дворівневу ансамблеву стекінгову модель.

Вибір саме дворівневої архітектури обумовлений результатами проведеного дослідження впливу глибини стекінгових ансамблів на узагальнювальну здатність моделей прогнозування академічної успішності [**104**]. У межах дослідження було проаналізовано ансамблі глибиною від одного до п’яти рівнів та встановлено, що збільшення кількості рівнів стекінгу не приводить до монотонного покращення точності прогнозування. На початкових рівнях ансамблювання спостерігається підвищення якості прогнозування за рахунок ефективного комбінування моделей різної природи та зменшення зміщення (bias), однак подальше збільшення глибини ансамблю супроводжується накопиченням похибок попередніх рівнів, зростанням дисперсії (variance) та перенавчанням метамоделей.

Результати дослідження показали, що ансамблі середньої глибини забезпечують найкращий баланс між узагальнювальною здатністю та складністю моделі. Зокрема,

дворівнева архітектура дозволяє ефективно агрегувати прогнози базових класифікаторів без суттєвого збільшення ризику перенавчання та обчислювальної складності. Враховуючи обсяг вибірки, структуру ознакового простору та необхідність забезпечення стабільних результатів на незалежних тестових даних, у роботі було обрано саме дворівневу стекінгову модель як оптимальну архітектуру для задачі прогнозування академічної успішності студентів.

Побудована дворівнева стекінгова модель, була розроблена шляхом комбінування трьох базових прогностичних моделей на базі: лінійної регресії (LR), нейронної мережі (NN) і випадкового лісу (RF), а для мета-навчання використано метод посилення градієнта (GB). Модель 1-го рівня включає три базових моделі з алгоритмами лінійної регресії, нейронної мережі і випадкового лісу. Та модель 2-го рівня (мета-модель), яка поєднує передбачення базових моделей. Базові моделі навчаються на одних і тих же навчальних даних, при цьому кожна модель генерує прогнози для навчальних даних. На основі прогнозів базових моделей навчається мета-модель, яка вчиться найкраще поєднати отримані прогнози для покращення точності прогнозування. Формула яка представляє математичне визначення стекінгу в ансамблевих моделях має наступний вигляд **[105]** :

$$f_x(x) = \sum_{i=1}^{n} a_i f_i(x), \quad (4.1)$$

де $f_x(x)$ – прогноз стекінгової моделі;

$f_i(x)$ – прогноз кожної базової моделі;

a_i – ваговий коефіцієнт (внесок кожної моделі у фінальний прогноз);

n – кількість базових моделей.

Перевагою логістичної регресії є висока інтерпретованість та стійкість до перенавчання, однак модель має обмежену здатність до опису складних нелінійних залежностей. Завдяки нелінійним перетворенням нейронна мережа здатна виявляти

складні приховані закономірності, однак може бути чутливою до шуму та перенавчання. Випадковий ліс ефективно виявляє локальні нелінійні залежності та взаємодії між ознаками, забезпечуючи високу стійкість до викидів і шуму. Основна мета полягає в оптимальному комбінуванні окремих моделей, щоб підвищити загальну прогностичну здатність ансамблю. Загальна структура дворівневої моделі ансамблю стекування представлена на рис. 3.9.

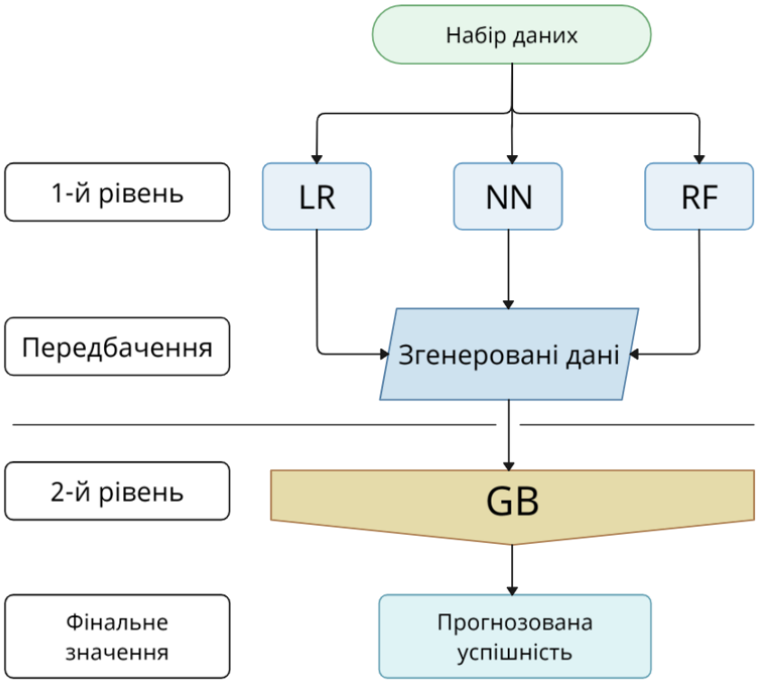


Рис. 3.9 – Структура 2-рівневої моделі ансамблю стекування

Дане рішення є лінійною комбінацією базових алгоритмів в єдину функцію передбачення, яка через вторинний процес навчання покращується, і в результаті покращує точність і стабільність передбачення **[106-107]**. На першому рівні використовуються K базових моделей:

$$f_1(x), f_2(x), \dots, f_K(x), \quad (4.2)$$

де $f_1(x)$ – LR(x) логістична регресія;

$f_2(x)$ – NN(x) MLC класифікатор нейронної мережі;

$f_3(x)$ – RF(x) випадковий ліс.

Кожна модель формує власний прогноз: $z_k = f_k(x), k = 1, \dots, K$ і тоді для кожного об'єкта утворюється вектор мета-ознак:

$$z(x) = \begin{matrix} f_1(x) \\ \dots \\ f_K(x) \end{matrix}, \quad (4.3)$$

На другому рівні використовується метамодель $\mathrm{g(z)} = \mathrm{GB(z)}$, яка отримує прогнози базових моделей як вхідні дані та формує фінальний прогноз:

$$\hat{y} = F(x) = g\big(f_1(x), f_2(x), \dots, f_K(x)\big), (4.4)$$

де:

$f_K(x)$ – базові моделі першого рівня;

$g()$ – метамодель;

$\hat{y}$ – фінальний прогноз ансамблю.

Кожна базова модель апроксимує різні аспекти досліджуваного процесу. Це призводить до того, що похибки моделей мають різну природу та різний розподіл. Для k-ї моделі похибка прогнозування визначається як:

$$e_k = y - f_k(x), (4.5)$$

де:

y – реальне значення цільової змінної;

$f_k(x)$ – прогноз базової моделі;

e_k – похибка моделі.

Ефективність ансамблю визначається не лише точністю окремих моделей, але й ступенем кореляції між їхніми похибками. Для оцінювання взаємозалежності похибок використовується коефіцієнт кореляції Corr(εi,εj). Якщо похибки двох моделей сильно корельовані то моделі припускаються подібних помилок, а комбінування таких прогнозів практично не покращує результат. У випадку слабкої кореляції або від'ємної кореляції ансамбль здатний компенсувати помилки окремих моделей, що

призводить до підвищення узагальнювальної здатності системи. На відміну від простого усереднення прогнозів, метамодель виконує адаптивне нелінійне комбінування результатів базових алгоритмів та враховує їхню локальну точність у різних областях простору ознак. Це досягається завдяки декореляції помилок та комбінуванню моделей, які по-різному реагують на структуру даних. Тому використання моделей різної природи в архітектурі стекінгу забезпечує підвищення точності прогнозування не лише за рахунок збільшення кількості моделей, а передусім завдяки комплементарності їхніх апроксимаційних властивостей та зменшенню кореляції похибок. Ансамбль стекування використовується саме для побудови сильної моделі, яка враховує передбачення інших підібраних алгоритмів моделювання для отримання високої точності прогнозування. Приклад створення базових моделей, генерація мета-ознак через крос-валідацію для тренувальних даних для створеної моделі представлено у програмному лістингу 3.1.

Програмний лістинг 3.1. Створення базових моделей, генерація мета-ознак через крос-валідацію для тренувальних даних, навчання базових та мета-моделі.

```
level1_models = {
    'LogisticRegression': LogisticRegression(
        max_iter=1000, class_weight='balanced', random_state=42
    ),
    'NaiveBayes': GaussianNB(),
    'SVM': SVC(kernel='rbf', probability=True, class_weight='balanced', random_state=42),
    'RandomForest': RandomForestClassifier(
        n_estimators=100, max_depth=10, class_weight='balanced',
        random_state=42, n_jobs=-1
```

```
    ),
    'NeuralNetwork': MLPClassifier(
        hidden_layer_sizes=(100, 50), max_iter=500,
        early_stopping=True, random_state=42
    )
}
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
meta_features_train = np.zeros((len(X_train_scaled), len(level1_models)))
meta_features_test = np.zeros((len(X_test_scaled), len(level1_models)))
for idx, (name, model) in enumerate(level1_models.items()):
    meta_features_train[:, idx] = cross_val_predict(
        model, X_train_scaled, y_train,
        cv=cv, method='predict_proba', n_jobs=-1
    )[:, 1]
    model.fit(X_train_scaled, y_train)
    meta_features_test[:, idx] = model.predict_proba(X_test_scaled)[:, 1]
meta_model = GradientBoostingClassifier(
    n_estimators=100, max_depth=3, learning_rate=0.1,
    random_state=42
)
meta_model.fit(meta_features_train, y_train)
```

```
y_pred_2level = meta_model.predict(meta_features_test)

y_proba_2level = meta_model.predict_proba(meta_features_test)[:, 1]
```

Основними критеріями для визначення ефективності та точності прогнозування моделі були обрані наступні показники: точність, збалансована точність, загальна точність, чутливість, специфічність, F1 Score, площа під кривою AUC та ROC-крива. Побудована матриця помилок стекінгової моделі прогнозування представлена на рис. 3.10.

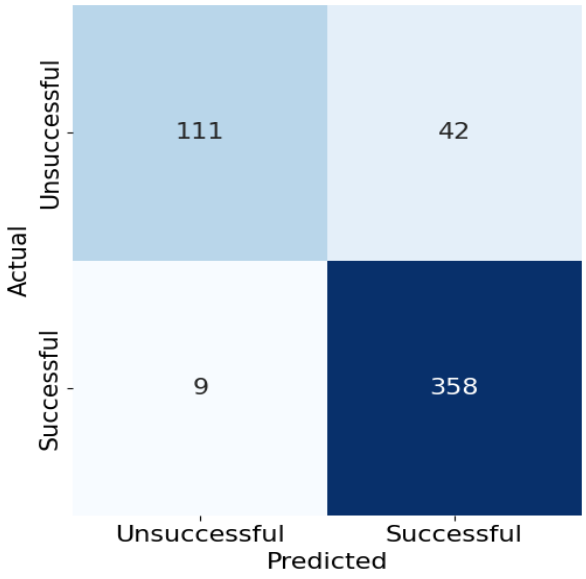


Рис. 3.10 – Матриця помилок для стекінгової моделі

На основі отриманої матриці розраховано величини, що характеризують загальну точність класифікації (прогнозування), а саме: точність, збалансовану точність, загальну точність, чутливість, специфічність, F1 Score та площу під кривою. Результати проведених розрахунків наведені в табл. 3.12.

Таблиця 3.12

Розрахунки значень характеризуючих загальну точність стекінгової моделі

Точність	Чутливість	Специфіч-ність	Збалансована точність	Загальна точність	F1 Score	Площа під кривою
0.895	0.975	0.725	0.850	0.902	0.898	0.926

Для того що оцінити здатність моделі до правильної класифікації візуально, була побудована ROC-крива, графік якої представлено на рис. 3.11.

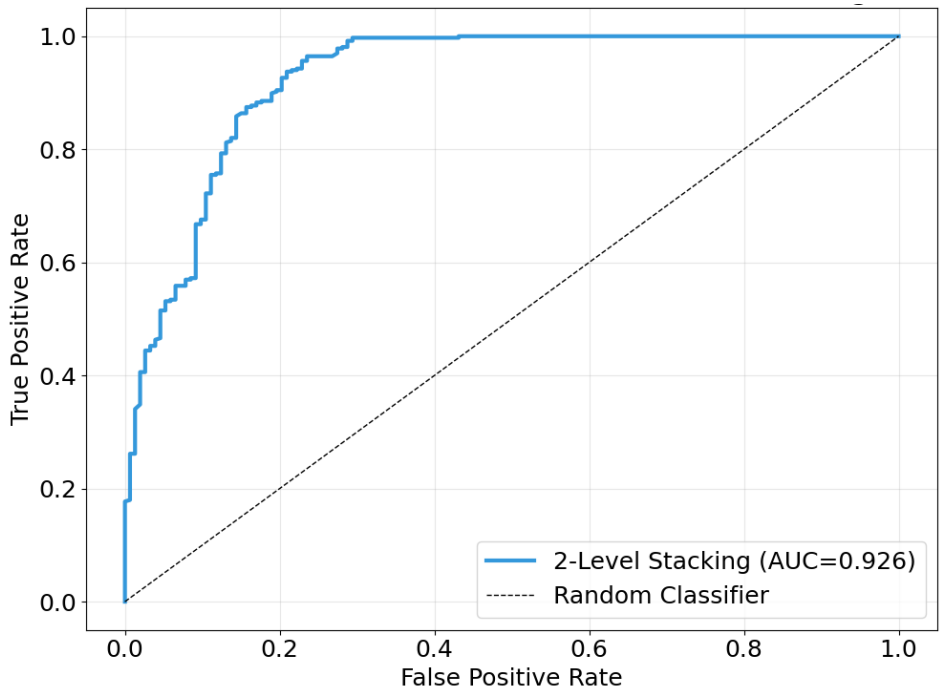


Рис. 3.11 – Графік ROC-кривої побудованої моделі

На графіку видно, що він має чітко визначену область, яка більше вигнута вгору та вліво. Чим більше і вище область вигнута вліво тим вище ефективність моделі. В даному випадку розраховане значення площі під кривою складає 92.6% і говорить про високу точність прогнозування моделі та високу дискримінаційну силу моделі. Згідно діапазону інтерпретації значення площі під кривою: 0.9 ≤ AUC отримане значення говорить про високу ефективність моделі **[108]**.

Порівняння приросту точності прогнозування дворівневої стекінгової моделі з результати моделей на базі випадкового лісу, логістичної регресії та нейронної мережі представлено в табл. 3.13, табл. 3.14, табл. 3.15. У всіх випадках прогнозування виконувалося на однаковому наборі даних.

Таблиця 3.13

Розрахунки значень приросту точності у порівнянні з логістичною регресією

Метрика	Базове значення	Стекінгова модель	Приріст
Точність	0.783	0.895	+14.30%
Чутливість	0.917	0.975	+6.32%
Специфічність	0.414	0.725	+75.12%
Збалансована точність	0.665	0.85	+27.82%
Загальна точність	0.765	0.902	+17.91%
F1 Score	0.845	0.898	+6.27%
Площа під кривою (AUC)	0.779	0.926	+18.87%

Таблиця 3.14

Розрахунки значень приросту точності у порівнянні з нейронною мережею

Метрика	Базове значення	Стекінгова модель	Приріст
Точність	0.856	0.895	+4.56%
Чутливість	0.95	0.975	+2.63%
Специфічність	0.63	0.725	+15.08%
Збалансована точність	0.79	0.85	+7.59%

Загальна точність	0.853	0.902	+5.74%
F1 Score	0.9	0.898	-0.22%
Площа під кривою (AUC)	0.859	0.926	+7.80%

Таблиця 3.15

Розрахунки значень приросту точності у порівнянні з випадковим лісом

Метрика	Базове значення	Стекінгова модель	Приріст
Точність	0.875	0.895	+2.29%
Чутливість	0.95	0.975	+2.63%
Специфічність	0.687	0.725	+5.53%
Збалансована точність	0.819	0.85	+3.79%
Загальна точність	0.871	0.902	+3.56%
F1 Score	0.911	0.898	-1.43%
Площа під кривою (AUC)	0.875	0.926	+5.83%

Дворівнева стекінгова модель продемонструвала суттєве покращення якості класифікації порівняно з базовими алгоритмами, що підтверджується узгодженим зростанням більшості показників ефективності. Загальна точність моделі досягла значення 0.902, перевищивши результати випадкового лісу (0.871), нейронної мережі (0.853) та логістичної регресії (0.765). Приріст точності на 3.6% відносно найкращої базової моделі та на 17.9% порівняно з логістичною регресією свідчить про кращу узагальнювальну здатність стекінгової архітектури та ефективність агрегування рішень різних алгоритмів. Чутливість стекінгової моделі становить 0.975, що перевищує значення, отримані для випадкового лісу та нейронної мережі (по 0.950) та

логістичної регресії (0.917). Це означає, що модель з високою ймовірністю правильно ідентифікує позитивний клас, та озволяє суттєво зменшити кількість хибнонегативних прогнозів. Значення специфічності є вищими за показники випадкового лісу (0.687), нейронної мережі (0.630) та логістичної регресії (0.414). Збільшення специфічності на 5.5%, 15.1% та 75.1% відповідно свідчить про покращену здатність моделі коректно розпізнавати негативний клас, тобто неуспішних здобувачів. Таким чином, стекінгова модель демонструє здатність зменшувати кількість хибнопозитивних прогнозів, що забезпечує більш надійне прийняття рішень. Приріст збалансованої точності на 3.8%, 7.6% та 27.8% відповідно підтверджує, що стекінгова модель забезпечує узгоджений баланс між виявленням позитивних і негативних випадків та не демонструє перекосу в бік одного з класів навіть за умов потенційної незбалансованості вибірки. Показник F1-score має незначне відставання від окремих базових моделей але це компенсується перевагами стекінгу за іншими показниками, зокрема чутливістю, специфічністю та збалансованою точністю. Це свідчить про стабільний компроміс між точністю та повнотою прогнозування. Приріст значення площі під кривою на 5.8%, 7.8% та 18.9% відповідно свідчить про високу дискримінаційну здатність моделі та її здатність надійно розрізняти класи в усьому діапазоні порогових значень, що є ключовим критерієм якості для задач бінарної класифікації. Результати експерименту дозволяють зробити висновок, що дворівнева стекінгова модель забезпечує більш повне та стійке відображення складної структури даних порівняно з окремими базовими алгоритмами.

3.7. Висновки до розділу 3

В ході виконання прогнозування успішності використовувалися наступні ознаки: відвідування, оцінки та взаємодія з навчальними відео матеріалами. Дані про взаємодію з відео матеріалами були отримані з електронного журналу, бази даних Moodel та розробленого плагіну VideoPlayer інтегрованого в систему управління

навчанням університету. Розрахунки показали, що алгоритми нейронних мереж (MLPClassifier) та випадкового лісу краще виконують прогнозування успішності і мають більшу точність у порівнянні з іншими. Порівняно з точністю наївного Баєса та логістичної регресії приріст в точності складає вже ~8,5%. Найвищій приріст по точності з різницею в 1.8% показали моделі з алгоритмами: випадкового лісу – 87.1% та нейронних мереж – 85.3%. Приріст загальної точності склав ~10%, збалансована точність збільшилася на 15%, а загальна ефективність виражена площею під кривою (AUC) збільшилась на 14%. Тоді як точність прогнозування моделей з алгоритмами наївного Баєса та логістичної регресії склала 70.7% та 76.5%, а приріст відповідно 2.3% та 8.1%. Отриманий результат свідчить про те, що додавання даних про взаємодію з відеоматеріалами добре впливає на підвищення точності прогнозування. Отримані результати експерименту підтверджують ефективність дворівневої стекінгової моделі для задачі прогнозування навчальної успішності студентів. Порівняльний аналіз показав, що стекінговий підхід забезпечує вищу загальну точність прогнозування 90.2% порівняно з усіма базовими класифікаторами, що свідчить про кращу узагальнювальну здатність моделі. Високе значення чутливості 97.5% вказує на здатність моделі надійно ідентифікувати успішних студентів та мінімізувати кількість хибнонегативних прогнозів, що є критично важливим у контексті освітньої аналітики. Одночасне зростання специфічності 72.5% свідчить про покращене розпізнавання неуспішних студентів і зменшення кількості хибнопозитивних рішень. Підвищення збалансованої точності до рівня 85% підтверджує, що модель забезпечує стійкий баланс між виявленням позитивного та негативного класів навіть за умов потенційної незбалансованості даних. Порівнянні значення F1-score 89.8% свідчать про збереження оптимального співвідношення між точністю та повнотою класифікації. Високе значення площі під ROC-кривою 92.6% підтверджує відмінну дискримінаційну здатність стекінгової моделі незалежно від вибору порогового значення. Загалом результати демонструють, що поєднання різних алгоритмів машинного навчання в межах стекінгової архітектури дозволяє отримати

більш надійну та стабільну модель порівняно з окремими базовими підходами. Це обґрунтовує доцільність використання запропонованої моделі в практичних системах прогнозування успішності здобувачів.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ ДЛЯ ЗБИРАННЯ ДАНИХ ПРО РОБОТУ З ВІДЕОМАТЕРІАЛАМИ ТА ФОРМУВАННЯ ЗВІТІВ

У розділі описано розроблений та інтегрований в Moodle плагін по збору даних взаємодії здобувачів із навчальними відеоматеріалами. Представлено основний функціонал, описано сценарії роботи плагіна з користувачем та адміністратором. Виконано розробку архітектури та апробовано результати роботи. Результати розділу опубліковано у наукових працях **[110, 111, 112, 113, 114, 115, 116]**.

4.1. Розробка програмних модулів для LMS Moodle

Плагін дозволяє відстежувати натискання кнопок управління плеєром, загальну тривалість перегляду та встановити, що був повний або частковий перегляд. Це може допомогти виявити частини матеріалу, які здобувачі можуть пропускати або з якими мають труднощі. Збір даних про кількість натискань кнопок "стоп", "пауза" та "початок відтворення" у продовж усієї сесії перегляду відео дозволяє детальніше аналізувати, як здобувачі взаємодіють з навчальними відеоматеріалами. Часте натискання на "паузу" може вказувати на те, що здобувачу потрібно більше часу для осмислення матеріалу або що відео є занадто швидким. Ще одним показником є загальна тривалість перегляду відео, вона вимірюється у хвилинах. Якщо здобувач переглядає відео лише частково або з великою перервою, це може вказувати на недостатню мотивацію або труднощі з матеріалом. Комбінуючи ці дані з іншими показниками, такими як оцінки за виконання завдань і тестів та відвідуваність занять можна прогнозувати академічний успіх і виявляти здобувачів, які можуть потребувати додаткової допомоги або підтримки в цифровому освітньому середовищі [**110-112**]. Процес встановлення плагіну в середовище Moodle представлений на рис. 4.1.

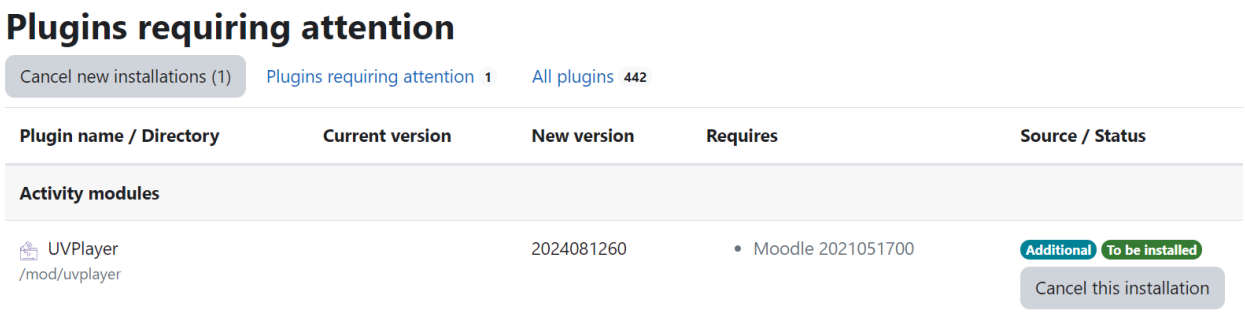


Рис. 4.1 – Встановлення плагіна в середовище Moodle

При розробці плагіну використовувалися мови програмування: PHP та JavaScript. На PHP була написана серверна логіка, а JavaScript використовувався для сторінки та обміну даними з сервером. Щоб уникнути затримок та перевантаження сторінки обмін даними має проходити асинхронно. Для цього було використано AJAX-запити, які дозволяють здійснювати асинхронні запити до сервера для отримання або надсилання даних без перезавантаження сторінки, що підвищує швидкість і зручність користувацького інтерфейсу плагіну [**113**]. Використання AJAX-запиту для відправки даних в ajax.php представлено в лістингу 4.1.

Лістинг 4.1. AJAX запит для відправки даних.

```
var xhr = new XMLHttpRequest();

var purl = "http://... /mod/uvplayer/ajax.php";

xhr.open("POST", purl, false);

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded;
charset=utf-8");

try { xhr.send(data); }

catch (e) { alert("Exception: " + e.message); }

xhr.onreadystatechange = function() {

   if (xhr.readyState === XMLHttpRequest.DONE){
```

```
if (xhr.status === 200){
    alert("Data success saved, status:" + xhr.status); }   }
```

Для збору даних про кількість натискань кнопок "стоп", "пауза" та "початок відтворення" у продовж всієї сесії перегляду використано подію onPlayerStateChange(event). Вона дозволяє виконувати обробку даних включаючи обробку кількості натискань при зміні стану кнопок. Використання події onPlayerStateChange для обробки натискань представлено в лістингу 4.2.

Лістинг 4.2. Використання події onPlayerStateChange(event).

```
var playCount = 0;
var pauseCount = 0;
var stopCount = 0;
function onPlayerStateChange(event) {
    if (event.data == YT.PlayerState.PLAYING) {
        playCount++;
        startTime = new Date().getTime();
    } else if (event.data == YT.PlayerState.PAUSED) {
        pauseCount++;
        if (startTime) {
            durationWatched += (new Date().getTime() – startTime) / 60000;
            startTime = null;
     } }
 else if (event.data == YT.PlayerState.ENDED) {
```

```
stopCount++;
completed = 1;
if (startTime) {
    durationWatched += (new Date().getTime() – startTime) / 60000;
} } }
```

Розроблений плагін має інтуїтивно зрозумілий інтерфейс налаштувань та має три основних поля для введення значень: назва відео, опис та посилання на відео з платформи YouTube [**114**]. Це значно спрощує процес конфігурації плагіна для викладача та адміністратора системи. Загальний вигляд інтерфейсу налаштувань плагіна представлено на рис. 4.2.

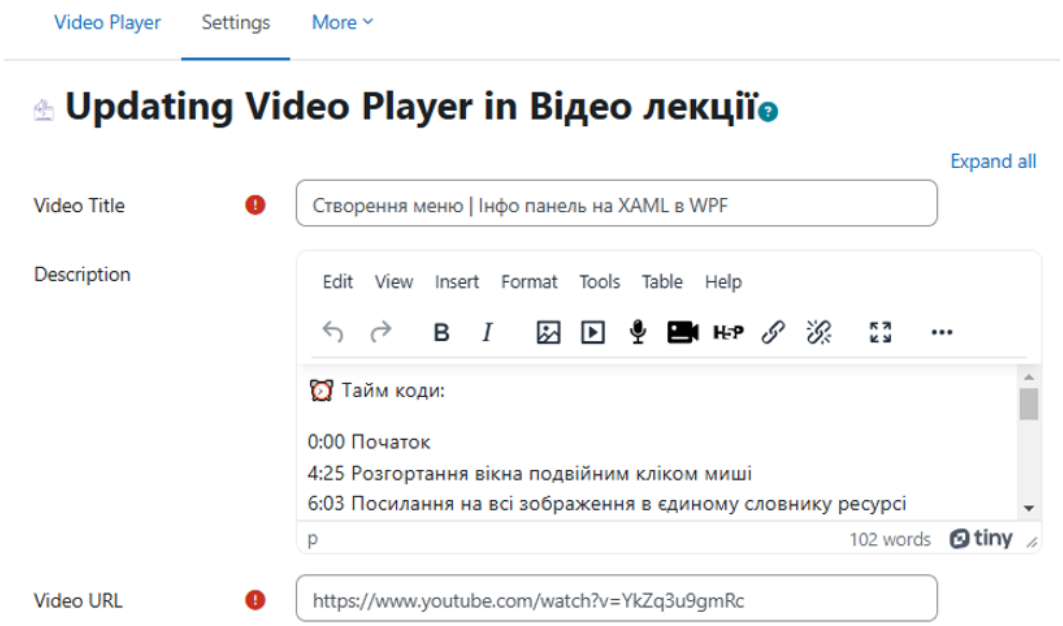


Рис. 4.2 – Налаштування плагіна для відтворення відео

Зрозумілий інтерфейс і простота в роботі, дозволяє легко інтегрувати даний плагін у навчальний процес і забезпечує легкість у подальшій експлуатації викладачами та здобувачами. Для того щоб додати відео на сторінку потрібно перейти

на загальну панель активностей та ресурсів, і вибрати VideoPlayer зі списку. Загальний вигляд панелі зі списком представлено на рис. 4.3.

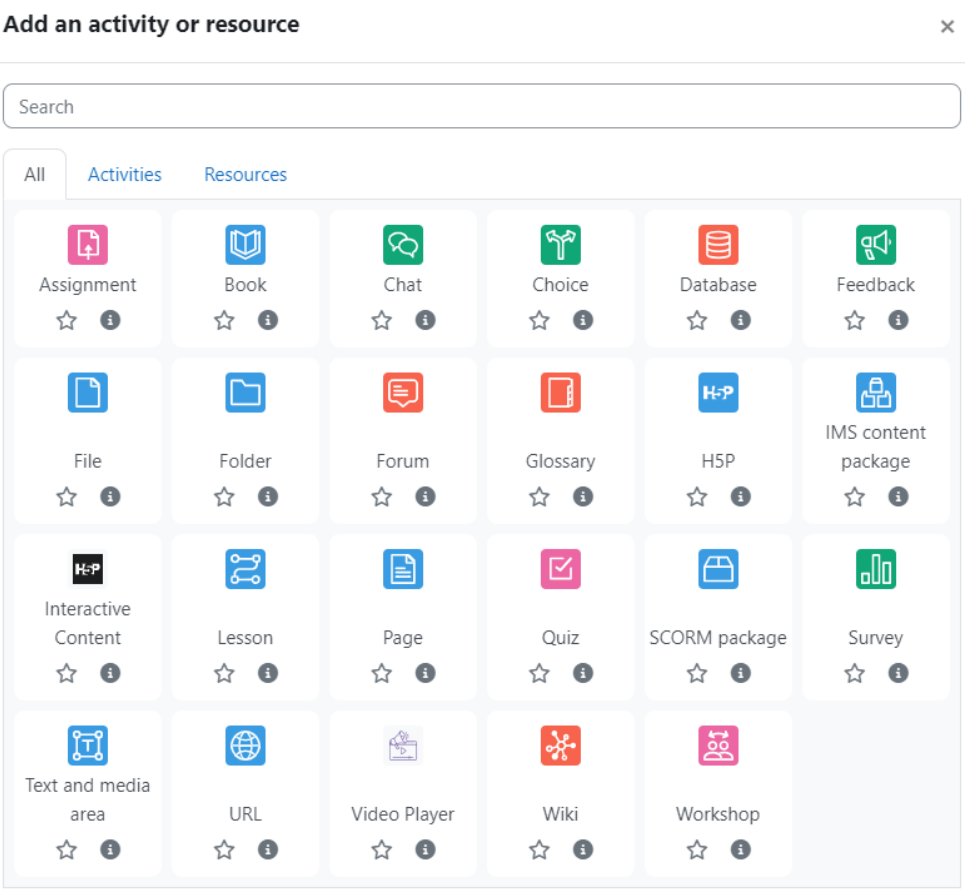


Рис. 4.3 – Вибір плагіна для вставки на сторінку

Плагін дозволяє користувачеві додати відео за посиланням, вказати назву відеолекції, а також задати детальний опис з часовими мітками для зручності перегляду. Додавання посилань до часових міток дозволяє швидко переходити до потрібної частини відео. Якщо при налаштуваннях дані будуть введено не вірно, тоді буде виведено відповідне повідомлення з попередженням. Після налаштувань на фінальній сторінці можна побачити назву, опис відео та відео плеєр фіксованого розміру з iFrame в середині. Загальний вигляд доданих навчальних відеолекцій на сторінці дисципліни в системі управління навчанням Moodle представлено на рис. 4.4.

Рис. 4.4 – Загальний вигляд доданих навчальних відеолекцій

В цьому списку кожен елемент є окремим відео плеєром, доданий викладачем. Кожен здобувач, що заходить на сторінку дисципліни має змогу подивитись будь-яке відео, при цьому кожен окремий відео плеєр збиратиме свою історію взаємодії з користувачем та зберігатиме до бази даних. Після вибору користувачем одного з плеєрів він переходить на сторінку перегляду навчального відео. При цьому, за необхідності, є можливість розвороту відео на весь екран. Загальний вигляд сторінки з налаштованим плагіном представлено на рис. 4.5.

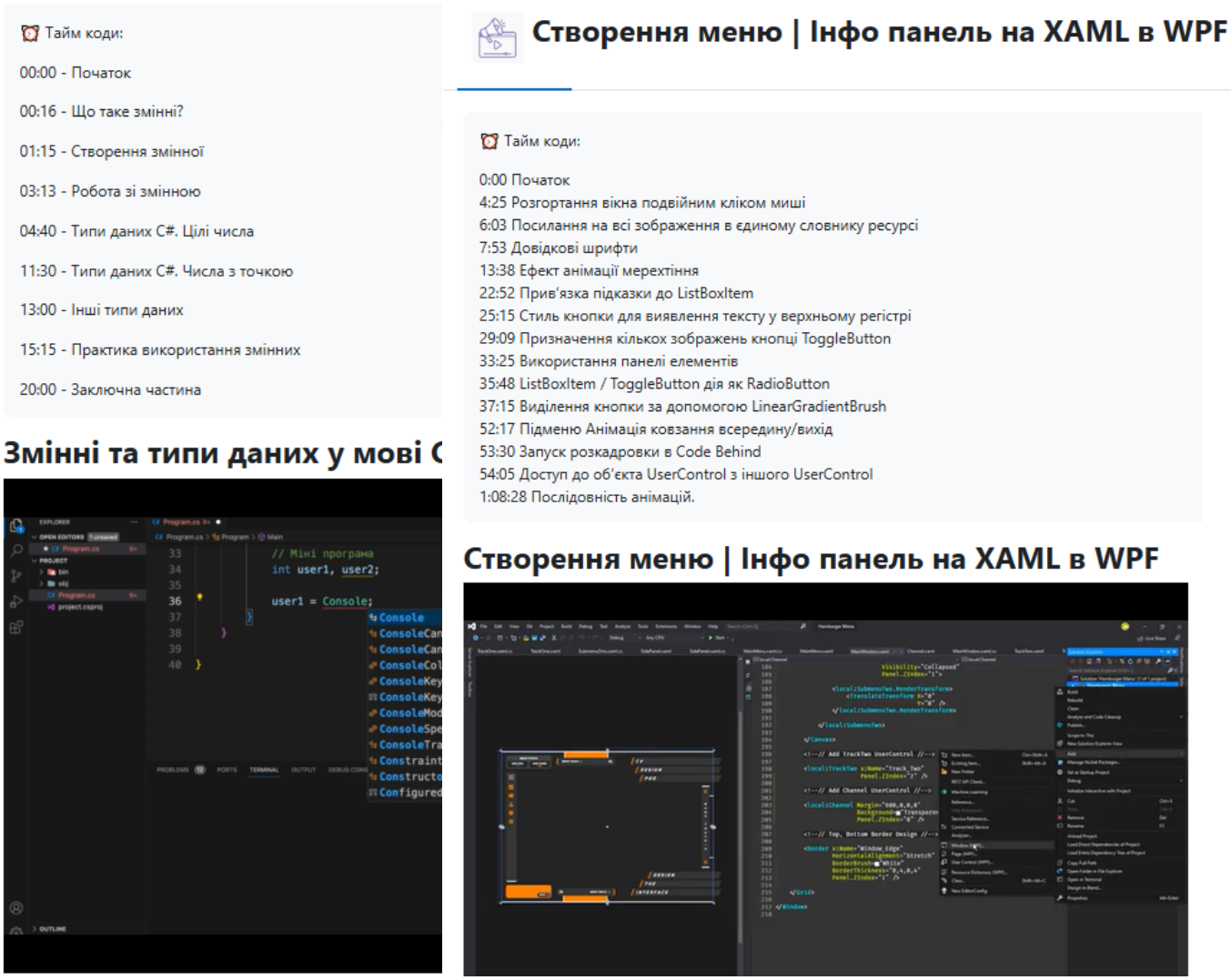


Рис. 4.5 – Загальний вигляд сторінки з налаштованим плагіном

В якості контейнера для плеєра використовується iFrame, який дозволяє вбудовувати одну веб-сторінку в іншу. Завдяки цьому, можна показати зовнішній контент, наприклад, відео з YouTube, документи та інші веб-ресурси без потреби зберігати їх локально [**115-116**]. Показ зовнішнього контенту без потреби зберігати його локально має низку переваг. Зокрема, дозволяє суттєво зекономити ресурси, зокрема місце на сервері, оскільки немає необхідності зберігати великі відеофайли. Крім того, зменшується навантаження на пропускну здатність серверу, адже контент завантажується напряму зі стороннього ресурсу, а саме YouTube. Це також забезпечує

автоматичне оновлення інформації: якщо контент змінюється на сторонньому сайті, ці зміни відображаються і у вбудованому вигляді, без потреби повторного завантаження або редагування вашої сторінки. Такий підхід значно спрощує підтримку сайту та забезпечує інтеграцію з якісними, вже готовими сервісами.

YouTube також надає IFrame Player API – це набір інструментів на JavaScript, який дозволяє програмно керувати відтворенням відео, вставленого через <iframe>. За допомогою API можна відтворювати або зупиняти відео, перемотувати його, реагувати на події (наприклад, коли відео почалося або завершилося), динамічно змінювати відео без перезавантаження сторінки. Прогрес бар та кнопки для управління відео має такий самий інтерфейс як і в YouTube та представлений на рис. 4.5.

Рис. 4.5 – Загальний вигляд інтерфейсу управління відео плеєром

Плагін починає зберігати дані про взаємодію з відео відразу після натискання кнопки початку відтворення відео і до моменту його закінчення. Якщо відео не було переглянуто до кінця дані всеодно будуть записані внутрішнім таймером, який детектує скільки часу користувач переглядав відео та які кнопки при цьому натискались.

4.2 Розробка архітектури плагіна відеоплеєра

Використання плагіна для Moodle, який зберігає дані взаємодії користувачів із відео в базу даних, має ряд суттєвих переваг та забезпечує високу зручність як для викладачів, так і для адміністраторів освітнього процесу. Насамперед, плагін дозволяє відстежувати поведінку здобувачів під час перегляду відео – зокрема, фіксує, які саме частини відео були переглянуті, які фрагменти пропущені, коли відбувалося зупинення або перемотування. Це надає можливість глибокого аналізу залученості здобувачів, що особливо важливо в дистанційному або змішаному навчанні. Збереження таких даних дозволяє створювати додаткові аналітичні звіти, виявляти проблемні місця в навчальному матеріалі та адаптувати подачу інформації відповідно до реальної активності користувачів. Наприклад, якщо більшість здобувачів зупиняються на певному моменті відео або часто перемотують його назад – це сигнал про складний для розуміння фрагмент, який варто пояснити детальніше. На рис. 4.6 представлена UML-діаграма класів, відображає структурну організацію плагіна UVPlayer, призначеного для інтеграції відеоконтенту з платформи YouTube у систему управління навчанням Moodle. Діаграма демонструє трирівневу архітектуру системи відстеження активності студентів при перегляді відеоматеріалів у середовищі Moodle. Архітектура системи складається з рівня даних, рівня прикладної логіки та рівня представлення з чітким розділенням відповідальності між компонентами.

Рівень даних представлено п'ятьма класами-сутностями, що відображають структуру реляційної бази даних. Центральний клас uvplayer_instance зберігає метадані відеоекземплярів, включаючи ідентифікатори, назву, опис активності, URL відеоресурсу YouTube та темпоральні атрибути для аудиту змін, а також надає методи для трансформації URL у формат вбудовування та валідації даних. Клас uvplayer_tracking відповідає за персистентність статистичних даних взаємодії користувачів з відеоконтентом через атрибути тривалості перегляду, кількості запусків, призупинень, зупинок та індикатора завершення перегляду, реалізуючи

аналітичні методи для обчислення відсотка переглянутого контенту та метрики залученості студента. Допоміжні сутності moodle_course, moodle_user та course_module забезпечують інтеграцію з існуючою архітектурою платформи Moodle, реалізуючи зв'язки між модулем відстеження відео, курсами та користувачами системи.

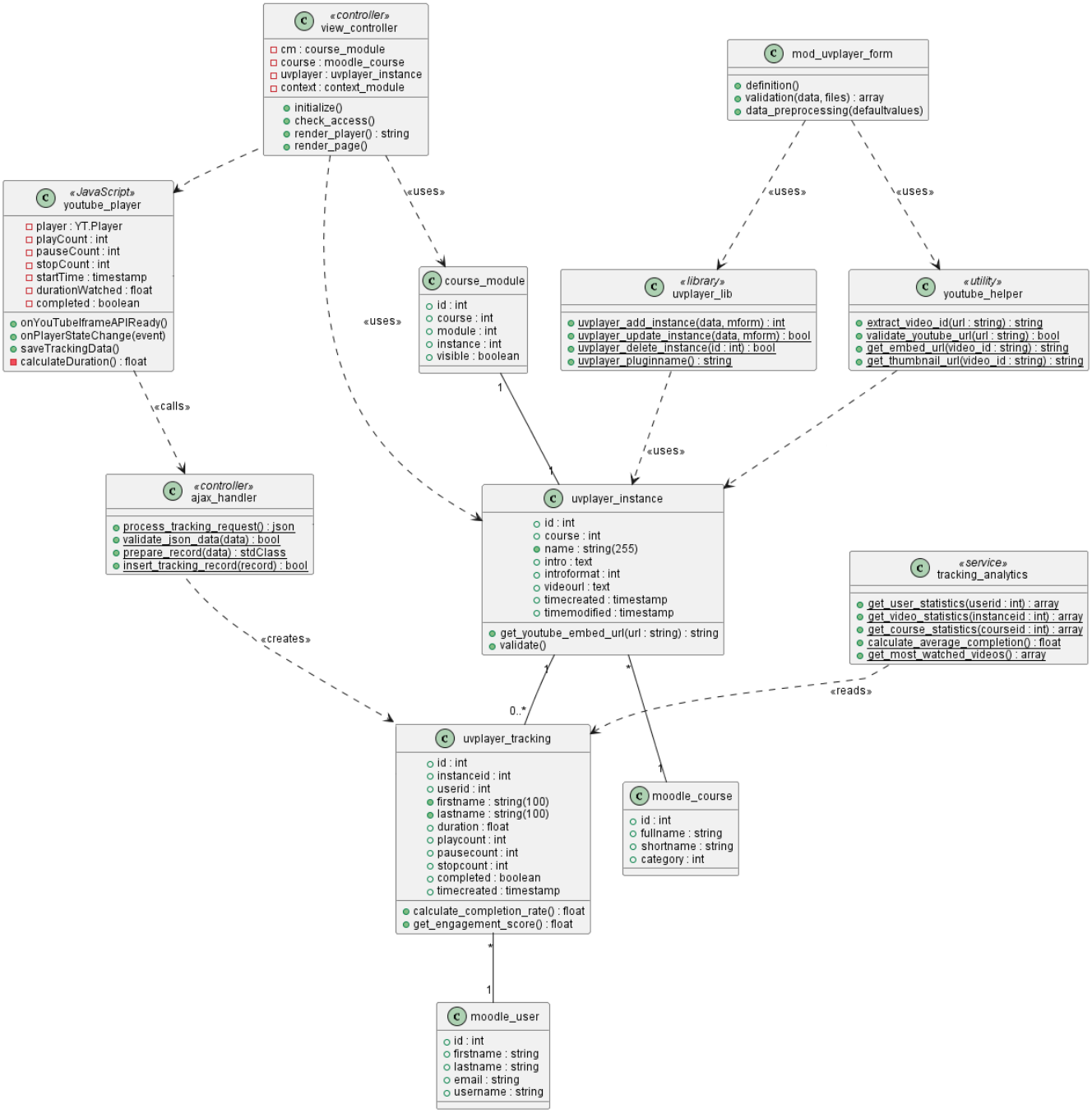


Рис. 4.6 – UML діаграма класів плагіна для Moodle

Рівень прикладної логіки реалізовано через статичний бібліотечний клас uvplayer_lib, що надає стандартний програмний інтерфейс для створення, модифікації та видалення екземплярів модуля згідно з архітектурою Moodle, клієнтський JavaScript клас youtube_player, що інкапсулює взаємодію з YouTube IFrame Player API та накопичує метрики відстеження у пам'яті браузера з подальшою асинхронною передачею на сервер, та серверний клас-контролер ajax_handler для обробки асинхронних запитів з валідацією JSON-даних та атомарним записом у базу даних. Контролер рівня представлення view_controller керує відображенням сторінки перегляду відео, виконуючи завантаження контексту запиту, верифікацію прав доступу користувача, генерацію HTML-коду вбудованого плеєра та формування повної сторінки через шаблонізатор, тоді як клас mod_uvplayer_form забезпечує адміністративний інтерфейс створення та редагування активності з серверною валідацією введених даних.

Рівень сервісів представлено аналітичним класом tracking_analytics для агрегації статистичних даних на рівні користувача, відеоекземпляра та курсу з розрахунком середніх показників та формуванням рейтингів популярності відеоматеріалів, а також допоміжним класом youtube_helper для роботи з URL відеоресурсів через методи екстракції ідентифікаторів за допомогою регулярних виразів, валідації коректності посилань та генерації URL для вбудовування і превью-зображень. Архітектурні зв'язки діаграми включають асоціації типу один-до-багатьох між uvplayer_instance та uvplayer_tracking, багато-до-одного між екземплярами та курсами, один-до-одного між course_module та uvplayer_instance, а також залежності використання між класами прикладної логіки та класами даних без створення персистентних зв'язків.

Діаграма демонструє застосування архітектурного патерну MVC для розділення моделей даних, контролерів та представлення, патерну Service Layer для виділення аналітичної та допоміжної функціональності, патерну Static Factory для управління

життєвим циклом екземплярів, та патерну Facade для спрощення інтерфейсу взаємодії зі стороннім API YouTube, що забезпечує низьку зв'язаність компонентів, високу згуртованість модулів та дотримання принципу єдиної відповідальності згідно з методологією об'єктно-орієнтованого проектування програмних систем. Плагін UVPlayer забезпечує гнучку інтеграцію відеоконтенту в освітній процес, підтримує моніторинг активності здобувачів та відповідає принципам масштабованості. Завдяки модульності архітектури та дотриманню принципів MVC система легко адаптується до змін середовища Moodle. Дані про взаємодію користувача зберігаються у внутрішній базі даних (БД) MariaDB середовища Moodle. Загальний вигляд таблиці із даними користувачів в БД представлено на рис. 37.

#	id	instanceid	userid	firstname	lastname	duration	playcount	pausecount	stopcount	completed	timecreated
1	1	0	2	[redacted]	[redacted]	1.0	2	3	1	1	1,723,549,360
2	2	0	2	[redacted]	[redacted]	1.0	2	3	1	1	1,723,549,405
3	3	0	2	[redacted]	[redacted]	1.0	2	3	1	1	1,723,549,409
4	4	0	2	[redacted]	[redacted]	3.0	2	3	10	1	1,723,549,861
5	5	0	2	[redacted]	[redacted]	5.0	2	3	10	1	1,723,550,034
6	6	9	2	[redacted]	[redacted]	0.35	2	1	1	1	1,723,550,445
7	7	9	2	[redacted]	[redacted]	0.22	2	1	1	1	1,723,554,204
8	8	4	4	[redacted]	[redacted]	0.09	2	1	1	1	1,723,554,257
9	9	4	4	[redacted]	[redacted]	0.18	2	1	1	1	1,723,554,397
10	10	4	2	[redacted]	[redacted]	0.7	6	5	1	1	1,723,554,791
11	11	6	2	[redacted]	[redacted]	1.06	7	6	1	1	1,723,555,011
12	12	7	2	[redacted]	[redacted]	20.55	1	0	1	1	1,723,556,443

Рис. 4.7 – Таблиця БД із даними взаємодії користувачів

Основними показниками взаємодії користувачів із відеоматеріалами, які збирає плагін для запису в БД, є кількість натискань кнопок: «паузи», «зупинки», «початку відтворення», тривалість перегляду та стан перегляду відео до кінця. Загальний вигляд таблиці з даними про взаємодію з конкретним навчальним відео представлено на рис. 4.8.

Columns: Add Remove Up Down

#	Name	Datatype	Length...	Unsign...	Allow N...	Zerofill	Default	Comm...	Collation	Expr...	Virtua...
1	id	BIGINT	10				AUTO...				
2	instanceid	BIGINT	10				'0'				
3	userid	BIGINT	10				'0'				
4	firstname	VARCHAR	100				NULL		utf8mb4_unico...		
5	lastname	VARCHAR	100				NULL		utf8mb4_unico...		
6	duration	DECIMAL	10,2				'0.00'				
7	playcount	BIGINT	10				'0'				
8	pausecount	BIGINT	10				'0'				
9	stopcount	BIGINT	10				'0'				
10	firstplay	TINYINT	1				'0'				
11	playbackrate	DECIMAL	10,2				'1.00'				
12	completed	TINYINT	1				'0'				
13	timecreated	BIGINT	10				'0'				

Рис. 4.8 – Таблиця БД із даними взаємодії з навчальним відео

Загальний вигляд таблиці з даними про відео матеріал доданий викладачем на курсі представлено на рис. 4.9.

Columns: Add Remove Up Down

#	Name	Datatype	Length...	Unsign...	Allow N...	Zerofill	Default	Comm...	Collation	Expr...	Virtua...
1	id	BIGINT	10				AUTO...				
2	course	BIGINT	10				No def...				
3	name	VARCHAR	255				''		utf8mb4_unico...		
4	intro	LONGTEXT					NULL		utf8mb4_unico...		
5	introformat	SMALLINT	4				No def...				
6	videourl	VARCHAR	255				''		utf8mb4_unico...		
7	timecreated	BIGINT	10				No def...				
8	timemodified	BIGINT	10				No def...				

Рис. 4.9 – Таблиця БД із даними про доданий відеоматеріал

Архітектура плагіна UVPlayer сформована з урахуванням технічних вимог платформи Moodle, принципів безпечної розробки, вимог до продуктивності та масштабованості, а також з урахуванням дидактичних потреб навчального процесу. Нижче наведено ключові чинники, що обумовили вибір архітектурного підходу.

Розробка здійснена з дотриманням інфраструктурних особливостей Moodle, що забезпечує сумісність, підтримуваність та легку інтеграцію плагіна в існуюче середовище, а саме:

- Успадкування moodleform_mod дозволяє використовувати типову систему форм Moodle для створення та редагування навчальних активностей.

- Використання глобального об’єкта $DB відповідає стандартному підходу до роботи з базою даних у Moodle і забезпечує автоматичний захист від SQL-ін’єкцій.
- Організація файлової структури (lib.php, mod_form.php, view.php) відповідає модульним принципам Moodle, що спрощує підтримку та розгортання плагіна.

Безпека є фундаментальним аспектом архітектури плагіна. Реалізовані рішення орієнтовані на запобігання несанкціонованому доступу та захист персональних даних, а саме:

- Окремі обробники AJAX-запитів: ajax.php та track.php (з повною авторизацією).
- Перевірка контексту та прав доступу реалізована через функцію require_capability() у захищених частинах коду.
- Валідація вхідних даних забезпечена за допомогою функцій PARAM_URL, PARAM_TEXT тощо, що є частиною вбудованої системи фільтрації Moodle.
- Використання Moodle Database API гарантує автоматичне екранування запитів і мінімізує ризики SQL-ін’єкцій.
- Механізм асинхронного трекінгу (через AJAX) забезпечує надсилання даних у режимі реального часу без переривання користувацького досвіду.

Архітектурна модель плагіна UVPlayer сформована з урахуванням стандартів платформи Moodle, принципів безпечного, масштабованого та підтримуваного програмного забезпечення, а також специфічних педагогічних вимог сучасного освітнього середовища. Інтеграція концептів модульності, орієнтації на зовнішні API, асинхронного трекінгу користувацької активності в реальному часі та адаптивності дозволяє системі ефективно вбудовувати відеоконтент у навчальний процес, забезпечуючи розширені можливості моніторингу та аналітики освітньої взаємодії. Діаграма станів відеоплеєра у модулі UVPlayer представлена на рис. 4.10. На етапі ініціалізації завантажується YouTube API та створюється екземпляр Player через конструктор YT.Player(). Після успішної ініціалізації система переходить до стану

Ready (готовий), в якому відеоплеєр повністю завантажений та готовий до відтворення, а лічильники статистики ініціалізовані (playCount = 0). З стану Ready можливий перехід до стану Playing (відтворення) при натисканні користувачем кнопки Play. У цьому стані генерується подія YTPlayerState.PLAYING, відбувається збільшення лічильника відтворень (playCount++), фіксується початковий час (startTime = now()) та запускається таймер перегляду. Зі стану Playing можливі чотири типи переходів: до стану Paused (пауза) при натисканні користувачем кнопки Pause, до стану Buffering (буферизація) при виникненні затримки завантаження даних, до стану Ended (завершено) при досягненні кінця відео, або повернення до стану Ready при зупинці відтворення. Стан Paused активується при призупиненні відтворення користувачем. У цьому стані генерується подія YTPlayerState.PAUSED, інкрементується лічильник пауз (pauseCount++), обчислюється тривалість перегляду з моменту останнього запуску (durationWatched += delta, де delta = now() - startTime) та запам'ятовується поточна позиція у відео. З цього стану можливі переходи назад до Playing (при відновленні відтворення) або до Ended (якщо пауза відбулась наприкінці відео). Стан Buffering є проміжним технічним станом, в який плеєр переходить при необхідності довантаження даних з мережі. Після завершення буферизації система автоматично повертається до попереднього стану (Playing або Paused залежно від контексту). Термінальний стан Ended досягається при завершенні відтворення відео до кінця. У цьому стані генерується подія YTPlayerState.ENDED, інкрементується лічильник завершень (stopCount++), встановлюється прапорець повного перегляду (completed = 1), акумулюється остання порція часу перегляду (durationWatched += delta) та автоматично викликається функція saveTrackingData() для збереження зібраної статистики у базі даних. З цього стану можливі два шляхи: відправка синхронного POST-запиту до ajax.php для збереження даних (при успішному збереженні повертається HTTP 200, при помилці – HTTP 500) або обробка події beforeunload при закритті вкладки користувачем, що також ініціює збереження даних через синхронний XMLHttpRequest. Окремою гілкою діаграми відображено обробку

події beforeunload (закриття вкладки браузера), яка може спрацювати з будь-якого активного стану. При її виникненні виконується фінальний підрахунок тривалості перегляду та гарантоване збереження статистики через синхронний AJAX-запит, що забезпечує цілісність даних навіть при несподіваному завершенні сесії користувачем.

Діаграма станів представлена на рис. 4.10 та демонструє повний життєвий цикл відеоплеєра від моменту завантаження до завершення роботи з обов'язковим збереженням аналітичних даних для подальшого аналізу навчальної активності здобувачів. Всі переходи між станами супроводжуються відповідними обчисленнями статистичних показників, що забезпечує точність та повноту збору даних про взаємодію користувача з навчальними відеоматеріалами. Представлена на діаграмі взаємодія демонструє цілісну архітектуру механізму відстеження перегляду відеоконтенту. Система забезпечує облік поведінкових подій користувача, передачу даних за умов нестабільного або перерваного сеансу, збереження статистики у базі даних Moodle, узгоджену взаємодію між клієнтським JavaScript-модулем, YouTube IFrame API та серверною логікою Moodle.

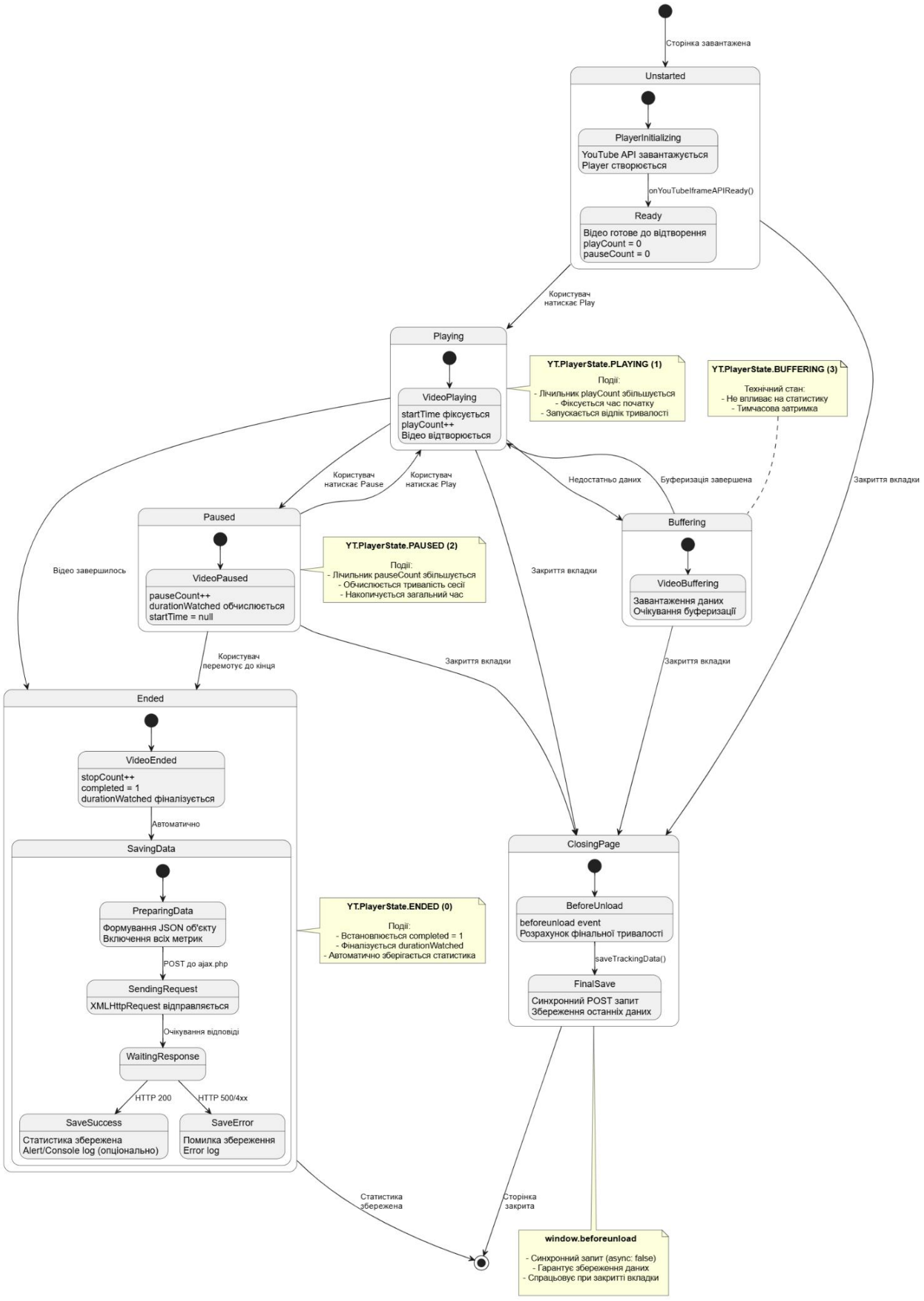

Сторінка завантажена
Unstarted
PlayerInitializing
YouTube API завантажується
Player створюється
onYouTubeIframeAPIReady()
Ready
Відео готове до відтворення
playCount = 0
pauseCount = 0
Користувач натискає Play
Playing
VideoPlaying
startTime фіксується
playCount++
Відео відтворюється
YT.PlayerState.PLAYING (1)
Події:
- Лічильник playCount збільшується
- Фіксується час початку
- Запускається відлік тривалості
YT.PlayerState.BUFFERING (3)
Технічний стан:
- Не впливає на статистику
- Тимчасова затримка
Користувач натискає Pause
Користувач натискає Play
Недостатньо даних
Буферизація завершена
Закриття вкладки
Paused
VideoPaused
pauseCount++
durationWatched обчислюється
startTime = null
YT.PlayerState.PAUSED (2)
Події:
- Лічильник pauseCount збільшується
- Обчислюється тривалість сесії
- Накопичується загальний час
Buffering
VideoBuffering
Завантаження даних
Очікування буферизації
Відео завершилось
Закриття вкладки
Користувач перемотує до кінця
Закриття вкладки
Закриття вкладки
Ended
VideoEnded
stopCount++
completed = 1
durationWatched фіналізується
Автоматично
SavingData
PreparingData
Формування JSON об'єкту
Включення всіх метрик
YT.PlayerState.ENDED (0)
Події:
- Встановлюється completed = 1
- Фіналізується durationWatched
- Автоматично зберігається статистика
POST до ajax.php
SendingRequest
XMLHttpRequest відправляється
Очікування відповіді
WaitingResponse
HTTP 200
HTTP 500/4xx
SaveSuccess
Статистика збережена
Alert/Console log (опціонально)
SaveError
Помилка збереження
Error log
ClosingPage
BeforeUnload
beforeunload event
Розрахунок фінальної тривалості
saveTrackingData()
FinalSave
Синхронний POST запит
Збереження останніх даних
Статистика збережена
Сторінка закрита
window.beforeunload
- Синхронний запит (async: false)
- Гарантує збереження даних
- Спрацьовує при закритті вкладки

Рис. 4.10 Діаграма послідовності роботи системи відстеження перегляду відео

Діаграма сутностей та зв’язків (рис. 4.11) відображає логічну модель даних, яка використовується модулем UVPlayer у навчальному середовищі Moodle. ER-діаграма демонструє, які таблиці беруть участь у зберіганні інформації, які атрибути вони містять, та як саме пов’язані між собою. Така структура забезпечує коректність зберігання даних, цілісність зв’язків, а також оптимальність виконання запитів і масштабованість системи.

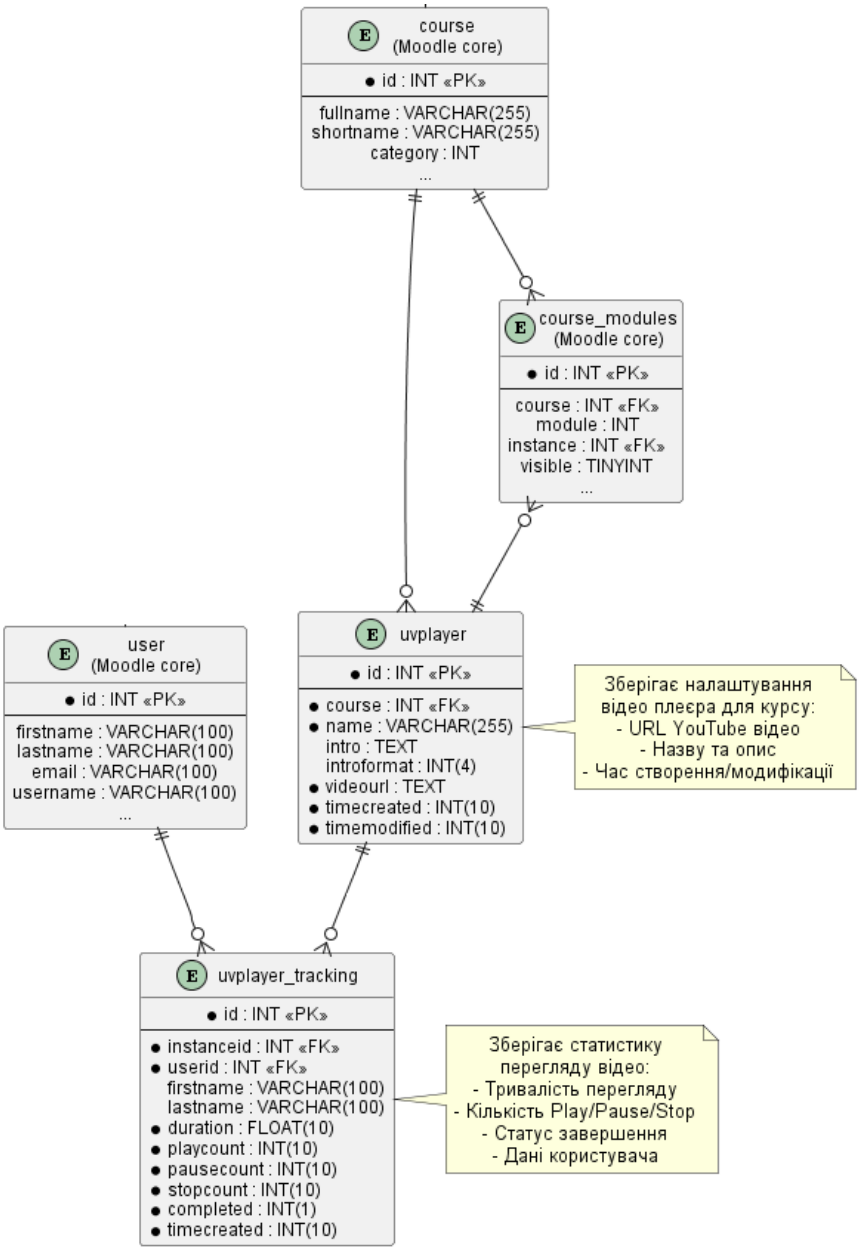


Рис. 4.11 Діаграма сутностей та зв’язків (ER) структури даних плагіна UVPlayer у середовищі Moodle

ER-модель демонструє структуру об’єктів бази даних, їх атрибути та взаємозв’язки, що забезпечують цілісність, узгодженість і ефективність зберігання даних про відеоресурси та інформацію про їх перегляди. Плагін UVPlayer інтегрується безпосередньо у стандартну архітектуру Moodle, тому використовує системні таблиці, зокрема course, user та course_modules, що функціонують як фундаментальні сутності загальної моделі даних. Таблиця course містить інформацію про навчальні курси, включаючи унікальний ідентифікатор, повну та коротку назви й належність до певної категорії. Таблиця user представляє зареєстрованих у системі користувачів – здобувачів, викладачів і адміністраторів – і включає їх особисті дані та облікову інформацію. Таблиця course_modules реалізує механізм прив’язки активностей до курсів і визначає тип модуля, інстанс конкретної активності та її видимість. Сутності, що належать безпосередньо плагіну, представлені таблицями uvplayer та uvplayer_tracking. Таблиця uvplayer зберігає конфігураційні дані кожного відеоекземпляра, зокрема назву, опис, формат опису, URL-адресу відеоматеріалу на платформі YouTube, а також часові мітки створення та модифікації. Це дозволяє формувати статичну структуру відеоресурсів у межах навчальних курсів. Таблиця uvplayer_tracking, у свою чергу, накопичує операційні дані, що характеризують перегляд відео здобувачами: тривалість відтворення, кількість взаємодій (play, pause, stop), індикатор завершеного перегляду, а також час створення запису. Значення імені та прізвища користувача дублюються у таблиці з метою оптимізації вибірки даних й уникнення необхідності частого приєднання таблиці user.

У структурі даних визначено три ключові типи зв’язків. Зв’язок між таблицями course та uvplayer є типу «один-до-багатьох» і забезпечує можливість розміщення декількох відеоресурсів у межах одного курсу. Аналогічний тип зв’язку встановлено між таблицями uvplayer та uvplayer_tracking, що дозволяє фіксувати потенційно

необмежену кількість переглядів одного відео різними користувачами. Таблиці user та uvplayer_tracking також пов’язані типом «один-до-багатьох», оскільки кожний користувач може створити кілька записів відстеження залежно від кількості переглядів різних відео або повторних переглядів одного й того ж ресурсу. Таблиця course_modules встановлює зв’язок із uvplayer типу «один-до-одного» через унікальну пару атрибутів (module, instance), що гарантує однозначне прикріплення певного відеоекземпляра до конкретного місця в курсі. Запити до бази даних, які виконуються під час використання модуля UVPlayer, включають вибірку параметрів курсу й відеоресурсу, з’єднання таблиць для виведення статистики переглядів та агрегаційні операції, наприклад обчислення середньої тривалості перегляду відео. Для підвищення продуктивності роботи системи застосовано індексацію ключових полів таблиці uvplayer_tracking, зокрема userid, instanceid та композитного індексу (instanceid, userid, timecreated). Це дозволяє значно прискорити виконання запитів навіть у разі значного накопичення статистичних даних. Забезпечення цілісності даних реалізується за допомогою зовнішніх ключів та правил каскадного видалення. Видалення курсу супроводжується автоматичним видаленням усіх відеоресурсів, пов’язаних із цим курсом, а видалення відео – видаленням усієї відповідної статистики переглядів. Такий підхід запобігає виникненню «висячих» посилань і підтримує узгодженість та структурну чистоту бази даних. Наведена діаграма відображає комплексну, логічно організовану модель даних, яка забезпечує управління відеоресурсами, збереження поведінкових даних користувачів та продуктивність роботи розробленого плагіна у середовищі Moodle.

Діаграми процесу перегляду відео представлені на (рис. 4.13 та рис. 4.14), і демонструють покроковий процес взаємодії здобувача з відеоресурсом у плагіні системи Moodle та відображають розподіл функціональної відповідальності між основними учасниками: користувачем, браузером, серверною частиною Moodle, JavaScript-обробником та модулем ajax.php. Така діаграма фіксує послідовність

операцій у межах асинхронного сценарію, описуючи поведінку системи від моменту ініціалізації перегляду до завершення сесії та збереження даних.

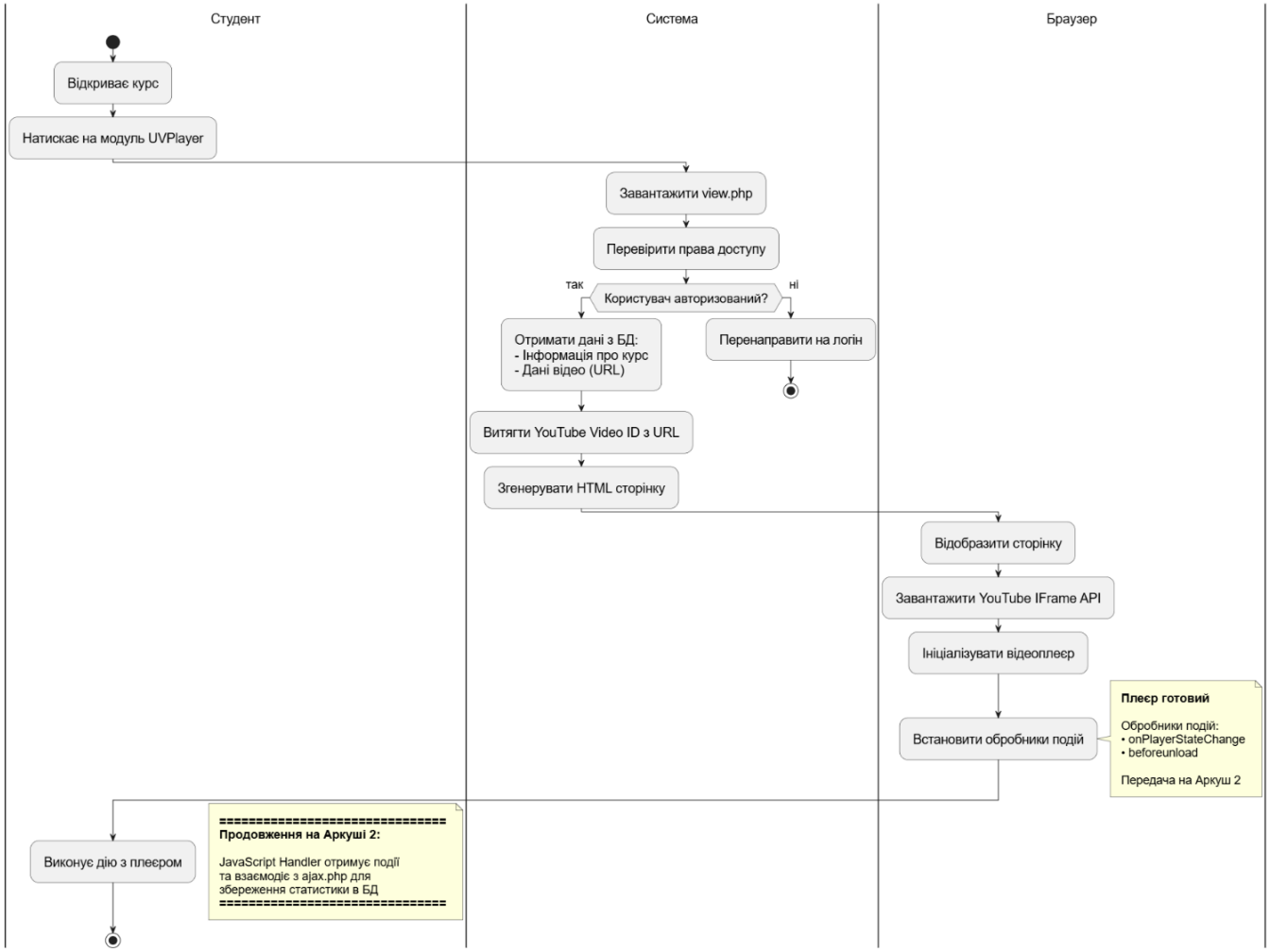


Рис. 4.13 – Діаграма діяльності процесу перегляду відео здобувачем (частина 1)

Перша частина діаграми (рис. 4.13) описує етап ініціалізації системи. Процес починається з відкриття користувачем навчального курсу та активації модуля UVPlayer. Серверна складова виконує автентифікацію користувача, здійснює отримання метаданих відеоматеріалу з бази даних, визначає ідентифікатор відео платформи YouTube (YouTube Video ID) та формує HTML-сторінку для відображення в браузері. У свою чергу веб-браузер завантажує YouTube IFrame API, ініціалізує

відеоплеєр та реєструє відповідні обробники подій, зокрема onPlayerStateChange і beforeunload.

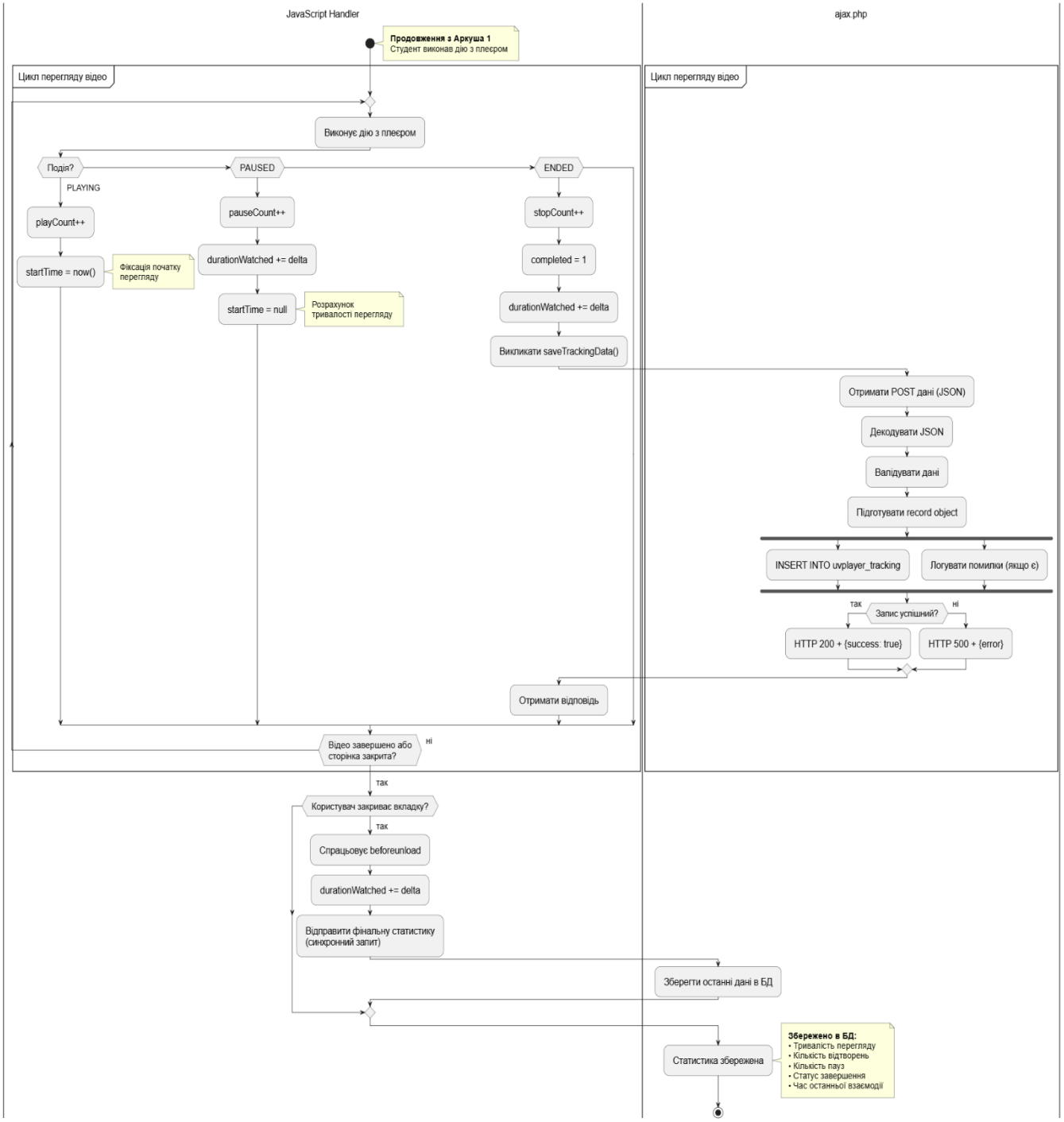


Рис. 4.14 – Діаграма діяльності процесу перегляду відео здобувачем (частина 2)

Друга частина діаграми (рис. 4.14) ілюструє механізм циклічної обробки подій, що виникають під час відтворення відео. Клієнтський модуль JavaScript Handler здійснює моніторинг станів плеєра та обробляє три основні типи подій: PLAYING – фіксація моменту початку відтворення та інкрементація лічильника playCount; PAUSED – інкрементація лічильника pauseCount і обчислення тривалості перегляду (durationWatched); ENDED – встановлення ознаки завершення перегляду (completed = 1) та ініціювання виклику функції saveTrackingData(). На рисунках 4.15 та 4.16 зображено діаграму послідовності взаємодії компонентів системи UVPlayer. Діаграма демонструє послідовність обміну повідомленнями між: здобувачем, браузером, серверним скриптом view.php, YouTube API, JavaScript Player Handler та базою даних Moodle.

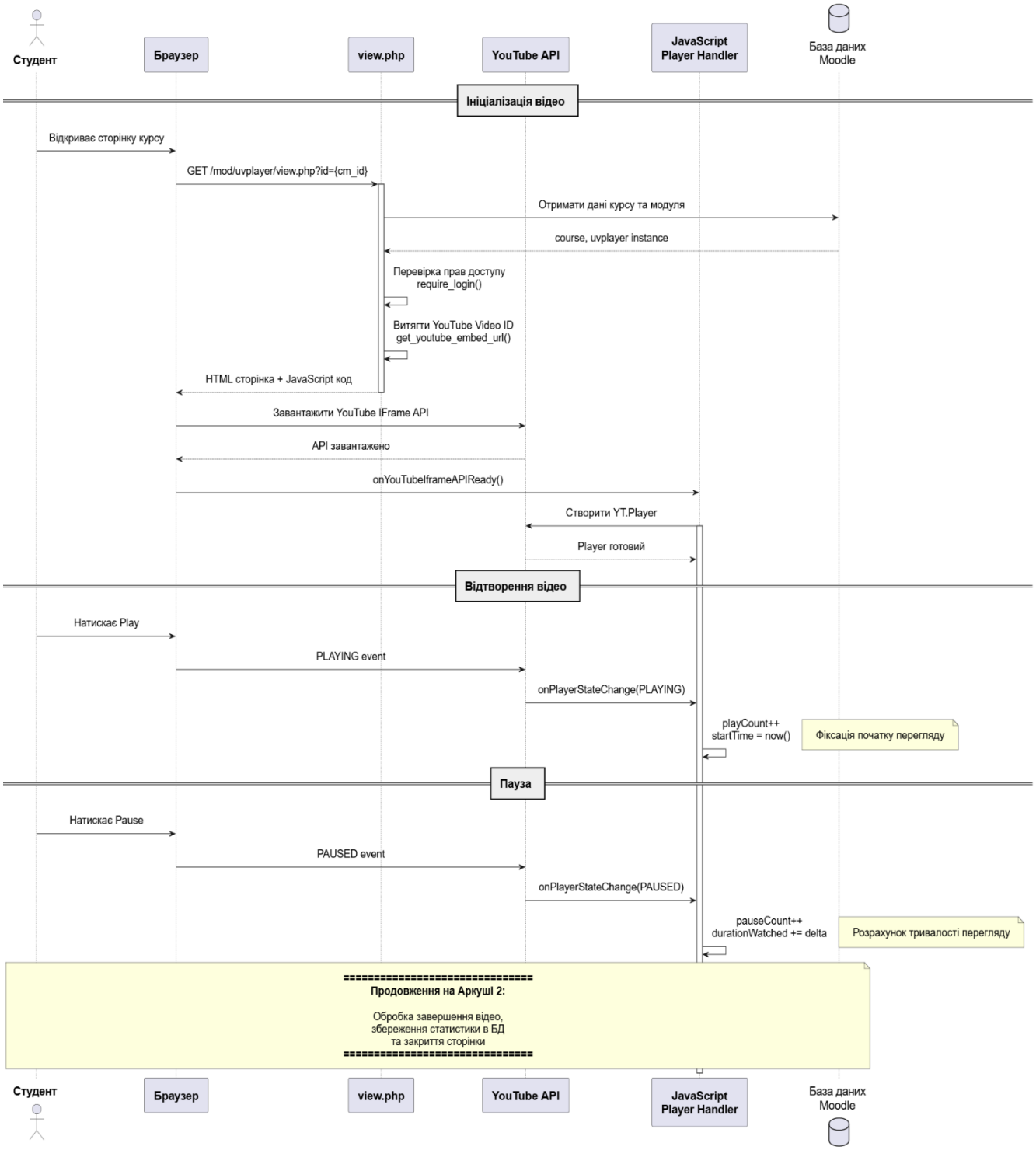


Рис. 4.15 – Діаграма процесу збереження даних в процесі перегляду відео (частина 1)

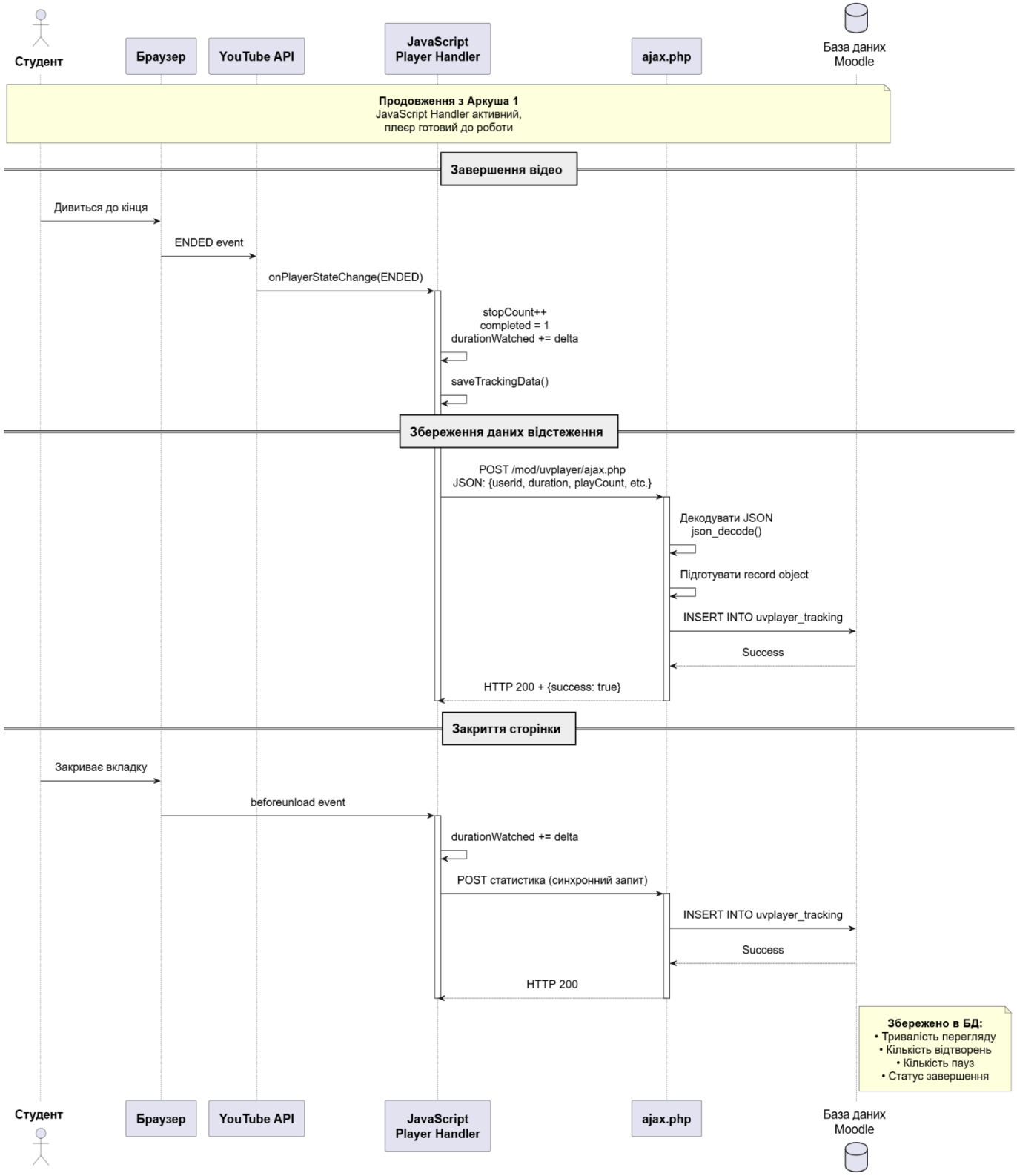


Рис. 4.16 – Діаграма процесу збереження даних в процесі перегляду відео (частина 2)

Друга частина (рис. 4.17) деталізує процеси збереження статистичних даних та завершення сесії перегляду. Коли здобувач дивиться відео до кінця, YouTube API генерує подію ENDED, яку перехоплює JavaScript Player Handler. Обробник виконує фінальні обчислення: інкрементує stopCount, встановлює прапорець завершення (completed = 1), оновлює загальну тривалість перегляду та викликає функцію saveTrackingData(). Ця функція формує POST-запит до серверного скрипта ajax.php, передаючи JSON-об'єкт зі статистичними даними (userid, duration, playCount, pauseCount, stopCount, completed). Серверна частина декодує JSON (json_decode()), валідує отримані дані, підготовує об'єкт запису та виконує операцію INSERT INTO uvplayer_tracking. У разі успішного збереження ajax.php повертає HTTP-відповідь зі статусом 200 та JSON-об'єктом {success: true}. Окремо діаграма демонструє обробку сценарію закриття вкладки браузера: подія beforeunload активує JavaScript Handler, який виконує фінальний підрахунок тривалості (durationWatched += delta) та відправляє синхронний POST-запит до ajax.php для збереження останніх даних у базі перед завершенням сесії. Використання синхронного запиту гарантує відправлення даних навіть у випадку швидкого закриття вкладки користувачем.

Результатом виконання повного циклу взаємодії є збереження у таблиці uvplayer_tracking комплексної інформації про сесію перегляду відеоматеріалу, що включає ідентифікатори користувача, курсу та відео, фактичну тривалість перегляду, кількісні показники взаємодії (відтворення, паузи, зупинки), статус завершення перегляду та часові мітки для подальшого аналізу навчальної активності здобувачів.

4.3. Розробка програмного забезпечення для візуалізації та розсилки звітів прогнозування успішності

Програмне забезпечення реалізовано на основі архітектурного шаблону Model–View–ViewModel (MVVM), який є одним із найбільш поширених підходів до розроблення настільних застосунків на платформі Windows Presentation Foundation (WPF). Використання даного шаблону забезпечує чітке розмежування між рівнем представлення даних, рівнем керування логікою взаємодії користувача із системою та рівнем бізнес-процесів. Така організація програмного коду підвищує його підтримуваність, тестованість та розширюваність, що є особливо важливим для науково-аналітичних програмних комплексів, функціональність яких може змінюватися або доповнюватися в процесі проведення досліджень. Архітектура програмної системи складається з чотирьох основних рівнів: представлення (View), логіки презентації (ViewModel), сервісного шару (Services) та моделей даних (Models). Рівень представлення реалізовано засобами XAML-розмітки та представлено головним вікном MainWindow, яке відповідає за відображення інтерфейсу користувача та забезпечує взаємодію із функціональними компонентами системи через механізм прив’язки даних (Data Binding). Логіка презентації інкапсульована у класі MainViewModel, який виступає центральним координатором усіх користувацьких сценаріїв та забезпечує зв’язок між графічним інтерфейсом і сервісним шаром. Рівень сервісів реалізує функціональні можливості системи, пов’язані з обробкою даних, машинним навчанням, побудовою візуалізацій та формуванням звітності. Рівень моделей забезпечує структуроване подання сутностей предметної області та використовується для обміну даними між окремими компонентами програмного комплексу. Запропонована архітектура відповідає фундаментальним принципам сучасної програмної інженерії, зокрема принципу єдиної відповідальності (Single Responsibility Principle) та принципу слабкого зв’язування (Loose Coupling). Кожен сервіс інкапсулює окрему функціональну

область і виконує визначений набір операцій. Керування станом користувацького інтерфейсу реалізовано на основі механізму ObservableProperty, що забезпечує автоматичне оновлення візуальних компонентів при зміні значень відповідних властивостей у ViewModel. Завдяки цьому в режимі реального часу оновлюються показники ефективності системи, статистичні метрики, індикатори виконання операцій, повідомлення журналу подій та службова інформація щодо поточного стану програмного комплексу. Реактивна модель оновлення даних дозволяє забезпечити високу швидкодію інтерфейсу та підвищити комфортність роботи користувача під час виконання тривалих обчислювальних процедур. Панель керування забезпечує доступ до всіх функціональних можливостей системи, тоді як область результатів організована у вигляді набору тематичних вкладок: «Дані», «Статистика», «Графіки», «Прогноз» та «Журнал». Така структура дозволяє користувачеві швидко перемикатися між різними режимами аналізу та отримувати необхідну інформацію у зручному для сприйняття вигляді. Сервісний рівень реалізує безпосередню взаємодію із зовнішніми джерелами даних та інструментами формування звітності. Для побудови графічних представлень результатів використовуються об’єкти типу PlotModel, які забезпечують створення інтерактивних наукових візуалізацій. Підсистема звітності підтримує експорт результатів аналізу у формати PDF та XLSX, що дозволяє інтегрувати результати дослідження у наукові публікації, звіти та документацію. Моделі даних представлені набором спеціалізованих класів, серед яких ReportData, ReportTable, ModelMetrics та класи конфігурації підключень. Вони забезпечують стандартизоване представлення результатів аналізу, параметрів навчання моделей та налаштувань системи, виступаючи проміжною ланкою між сервісами та компонентами інтерфейсу користувача. На рисунку 4.17 представлено загальний вигляд головного вікна програмного застосунку.

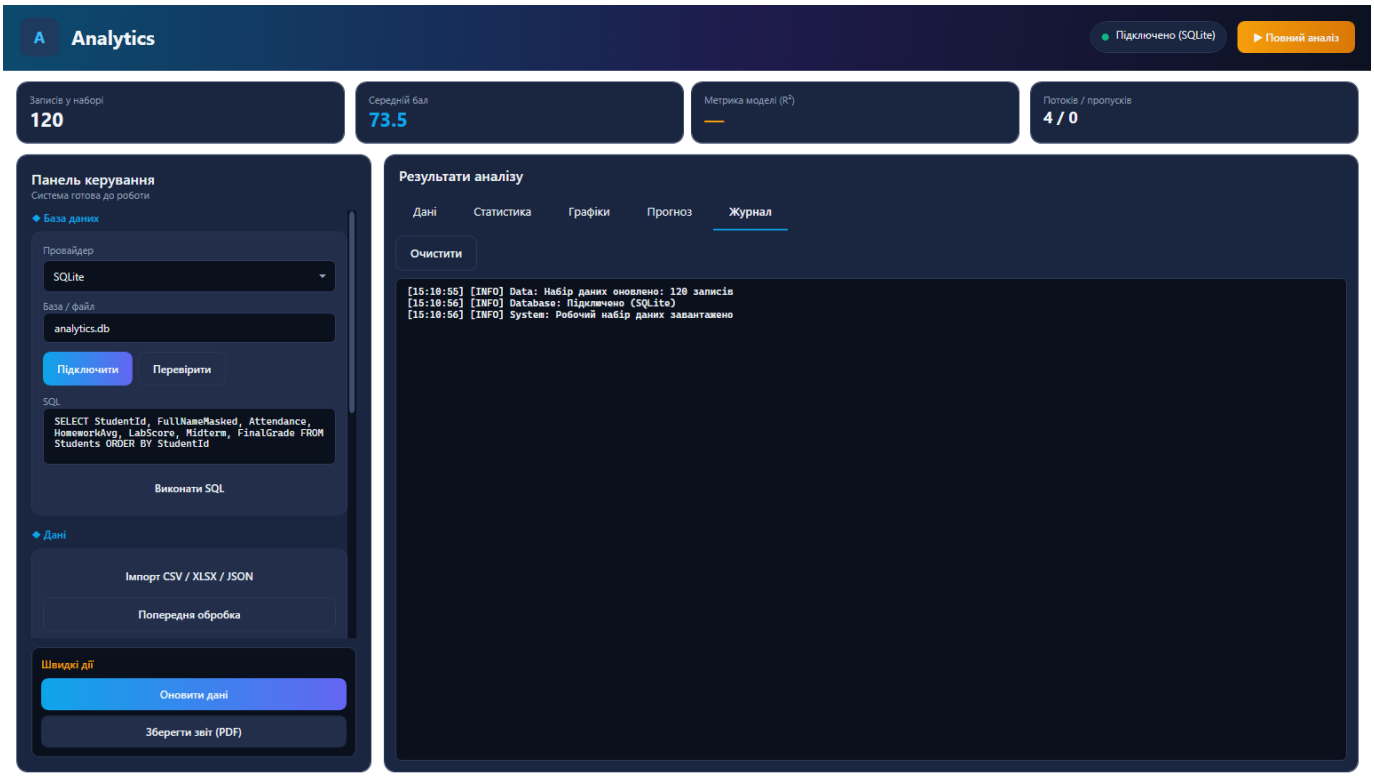


Рис. 4.17 – Загальний вигляд інтерфейсу користувача у застосунку Analytics

Головне вікно програмної системи структурно поділене на панель керування та панель відображення результатів. Панель керування містить елементи для підключення до бази даних, завантаження навчальних даних, виконання SQL-запитів, ініціалізації процесу навчання моделі, запуску прогнозування, побудови графічних візуалізацій та генерації звітів. Панель відображення забезпечує представлення даних у табличному, статистичному та графічному вигляді, що сприяє зручному аналізу результатів. Для забезпечення універсальності програмного рішення передбачено підтримку декількох типів систем керування базами даних. Вибір використовуваного джерела здійснюється за допомогою елемента керування типу ComboBox, який дозволяє користувачеві обрати необхідний провайдер доступу до даних. У поточній реалізації підтримуються локальні бази даних SQLite та серверні рішення на основі Microsoft SQL Server. Такий підхід забезпечує можливість використання програмного комплексу як у локальному режимі роботи, так і в середовищах корпоративного рівня. Інтерфейс взаємодії з базою даних реалізовано у вигляді окремого функціонального

блоку «База даних», розташованого на панелі керування головного вікна програмного комплексу і представлено на рисунку 4.18.

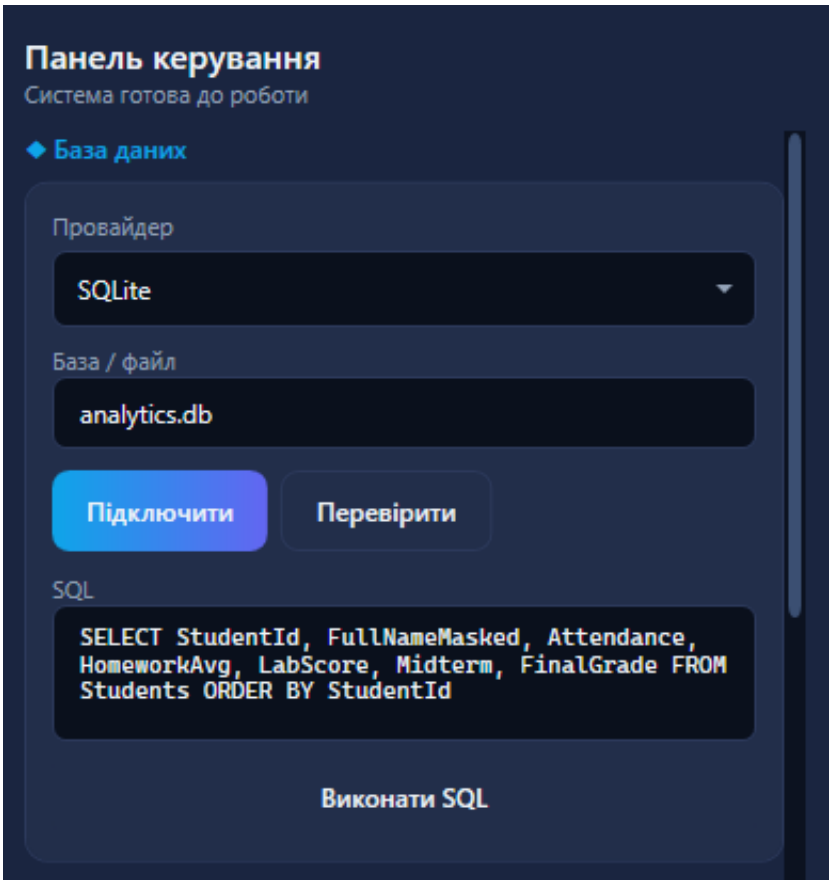


Рис. 4.18 – Інтерфейс взаємодії з базою даних

Після вибору типу джерела даних користувач задає параметри підключення у спеціалізованому полі. Для баз даних SQLite зазначається шлях до відповідного файлу бази даних, тоді як для SQL Server вказуються реквізити підключення, включаючи адресу сервера, назву бази даних та параметри автентифікації. Застосування єдиного інтерфейсу введення параметрів спрощує процес роботи користувача та забезпечує уніфікованість взаємодії з різними типами джерел інформації. Для встановлення з'єднання передбачено кнопку «Підключити», яка ініціює процедуру створення активної сесії взаємодії з базою даних. Додатково реалізовано функцію попередньої перевірки коректності налаштувань за допомогою кнопки «Перевірити». Виконання даної операції дозволяє здійснити тестове підключення до обраного джерела даних

без запуску повного циклу аналітичної обробки. У випадку успішної перевірки користувач отримує підтвердження доступності бази даних та коректності введених параметрів, що зменшує ризик виникнення помилок на наступних етапах роботи. Важливою складовою модуля є підсистема виконання SQL-запитів, яка забезпечує можливість безпосередньої взаємодії користувача зі структурою бази даних. Для цього під блоком налаштування підключення розміщено багаторядкове текстове поле, призначене для введення або редагування SQL-запитів. За замовчуванням у полі може бути розміщений типовий запит вибірки навчальних показників, що використовується під час демонстрації функціональних можливостей програмного комплексу. Виконання сформованого запиту здійснюється шляхом натискання кнопки «Виконати SQL», після чого результати автоматично передаються до підсистеми візуалізації та відображаються у відповідній вкладці області результатів. Для підвищення інформативності інтерфейсу реалізовано механізм оперативного контролю стану підключення. Після успішного встановлення з’єднання у верхній частині головного вікна відображається інформаційне повідомлення із зазначенням типу використовуваної системи керування базами даних, наприклад «Підключено (SQLite)» або «Підключено (SQL Server)». Одночасно змінюється стан візуального індикатора ConnectionStatus, який представлено у вигляді кольорового бейджа. Використання зеленого кольору індикатора сигналізує про готовність системи до виконання подальших операцій аналізу даних, включаючи імпорт інформації, навчання моделей машинного навчання, прогнозування та формування звітів. Додатково результати перевірки підключення та виконання SQL-запитів реєструються у журналі подій системи, що забезпечує можливість моніторингу виконаних операцій та спрощує процес діагностики можливих помилок. Такий підхід сприяє підвищенню надійності функціонування програмного комплексу та забезпечує контроль усіх етапів взаємодії користувача з джерелами даних. На рисунку 4.19 представлено загальний вигляд інтерфейсу із завантаженими даними.

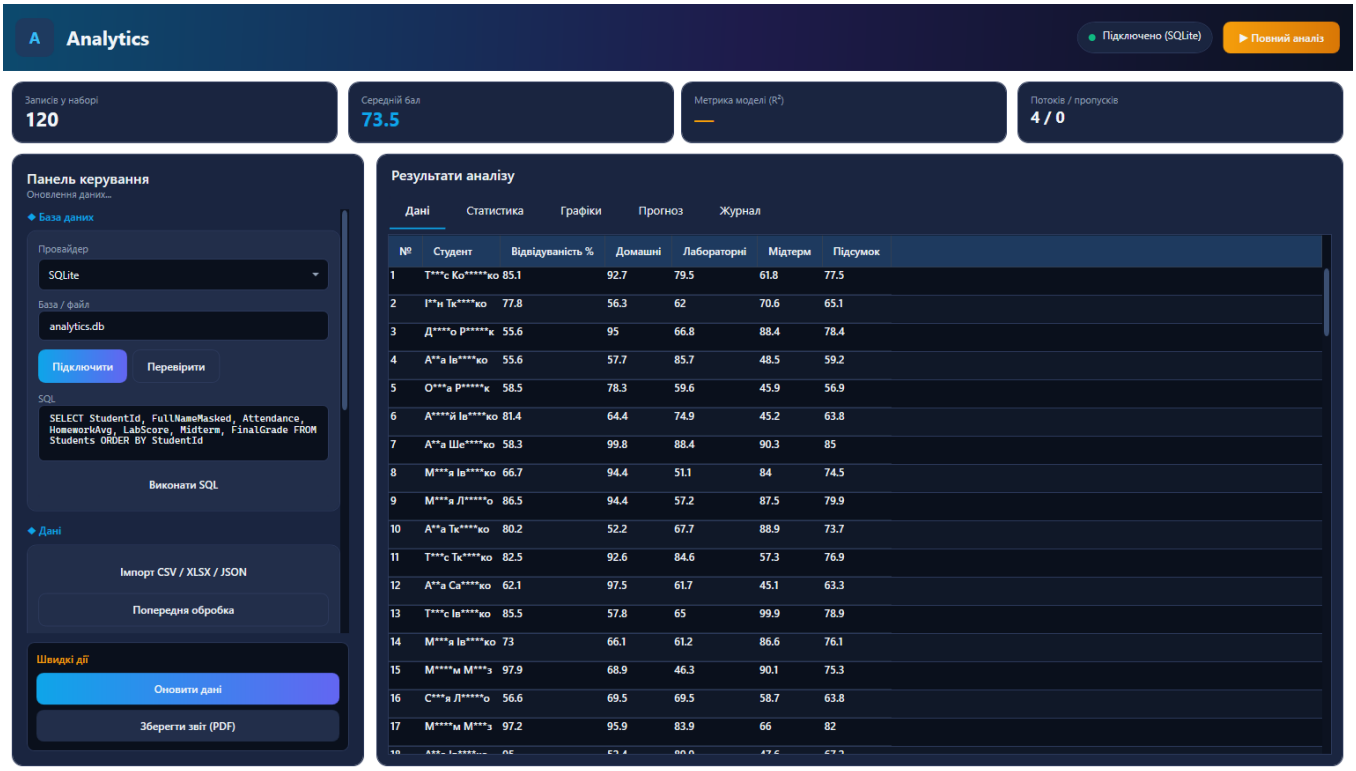


Рис. 4.19 – Загальний вигляд інтерфейсу застосунку із завантаженими даними

Підсистема звітності виконує роль інтеграційного компонента, що об’єднує результати роботи всіх функціональних модулів системи та перетворює їх у структурований документ, придатний для подальшого використання у науковій, освітній та управлінській діяльності. Підсистема звітності складається з трьох взаємопов’язаних компонентів: моделі даних звіту ReportData, модуля формування змістового наповнення DemoDataService.BuildAnalyticsReport та сервісу рендерингу ReportService. Такий підхід забезпечує розділення відповідальності між формуванням даних звіту, генерацією аналітичного змісту та безпосереднім створенням кінцевого документа. Використання окремої моделі звіту дозволяє стандартизувати структуру вихідних даних і спростити подальше розширення функціональних можливостей підсистеми. Модель ReportData виступає централізованим контейнером результатів аналізу та містить усі необхідні дані для формування звітної документації. До її складу

входять статистичні показники, результати прогнозування, параметри моделі машинного навчання, графічні матеріали, адміністративні таблиці та текстові аналітичні висновки. Завдяки використанню єдиної структури даних забезпечується узгодженість інформації між різними форматами представлення звітів. Формування змістової частини документа виконується модулем BuildAnalyticsReport, який автоматично генерує аналітичний звіт на основі результатів поточного сеансу роботи користувача. На відміну від традиційних шаблонних документів, зміст звіту формується динамічно відповідно до фактичних результатів виконаного аналізу. Це дозволяє забезпечити актуальність представленої інформації та мінімізувати необхідність ручного редагування документа після його створення. Структура сформованого звіту включає титульний блок та низку тематичних розділів, які відображають основні етапи виконаного аналізу. У розділі «Опис» наводиться загальна характеристика проведеного дослідження та його мета. Розділ «Набір даних» містить інформацію про джерела даних, кількість записів та основні характеристики вибірки. У розділі «Статистика» подаються результати статистичного аналізу та ключові показники досліджуваних даних. Розділ «Модель» містить відомості про використаний алгоритм машинного навчання, параметри навчання та метрики якості моделі. У розділі «Прогноз» відображаються результати прогнозування та оцінка точності отриманих прогнозів. Завершальними елементами документа є розділи «Ключові висновки» та «Рекомендації», які містять автоматично сформовані аналітичні узагальнення та практичні рекомендації щодо подальших дій. Особливе значення у структурі звіту мають адміністративні таблиці (AdminTables), які забезпечують детальне подання результатів аналізу у табличному вигляді. До складу цих таблиць входять зведені відомості щодо навчальних потоків, статистичний розподіл категорій успішності та порівняльні таблиці фактичних і прогнозованих значень. Наявність таких структурованих даних дозволяє здійснювати поглиблений аналіз результатів та використовувати звіт як основу для прийняття управлінських рішень.

Для підвищення наочності звітності до документа автоматично інтегруються графічні матеріали, сформовані підсистемою візуалізації. Зокрема, гістограма розподілу підсумкових балів експортується засобами сервісу ChartService у формат PNG та вбудовується до звіту як окремий графічний елемент. Завдяки цьому забезпечується повна відповідність між візуальними представленнями, доступними в інтерфейсі програмного комплексу, та графіками, що містяться у сформованій документації. Такий підхід підвищує інформативність звіту та спрощує сприйняття результатів аналізу кінцевими користувачами. Рендеринг документа здійснюється сервісом ReportService, який забезпечує формування звітів у декількох форматах представлення. Основним форматом є PDF, створення якого реалізовано за допомогою бібліотеки QuestPDF. Документ генерується відповідно до міжнародного стандарту формату аркуша A4 із використанням полів шириною 32 pt. Структура документа містить тематичні секції, текстові блоки, таблиці та графічні елементи, оформлені відповідно до вимог офіційної аналітичної документації. Для підвищення зручності навігації у нижній частині кожної сторінки автоматично виводиться дата формування документа, а в колонтитулі реалізовано наскрізну нумерацію сторінок у форматі «Analytics стор. N / M». Така організація документа відповідає сучасним вимогам до оформлення звітних матеріалів та забезпечує їх зручне використання у друкованому й електронному вигляді. Другим підтримуваним форматом є електронна таблиця Microsoft Excel. Структура Excel-звіту передбачає створення окремих аркушів для загального змісту документа та детальних аналітичних таблиць. Такий формат забезпечує можливість подальшого опрацювання результатів, виконання додаткових розрахунків та інтеграції звітних даних із зовнішніми інформаційними системами. Для забезпечення практичного використання результатів дослідження в програмному комплексі реалізовано декілька сценаріїв розповсюдження сформованих звітів. Перші два це локальне збереження документа за допомогою команд «Зберегти звіт (PDF)» або «Звіт PDF / Excel» та через кнопку «Відкрити теку звітів», яка автоматично запускає Провідник Windows та відкриває каталог зі

сформованими документами. Файл автоматично записується до каталогу звітів користувача, розташованого за шляхом %UserProfile%\Documents\Analytics\Reports. Для уніфікації процесу архівування використовується стандартизована схема найменування файлів, що включає дату та час створення документа у форматі AnalyticsReport_YYYYMMDD_HHmm. Такий підхід спрощує організацію архіву звітів та забезпечує можливість швидкого пошуку необхідних документів. Третій сценарій орієнтований на автоматизоване електронне розповсюдження результатів. Для його реалізації використовується метод SendEmailAsync, який забезпечує надсилання сформованих PDF-документів засобами протоколу SMTP. Конфігурація параметрів поштового сервера виконується відповідно до вимог організації та може включати використання корпоративних засобів автентифікації та захищених каналів передачі даних. Такий механізм дозволяє інтегрувати програмний комплекс у наявні процеси електронного документообігу та автоматизувати розсилання результатів аналітичної діяльності визначеним отримувачам. На рисунку 4.20 представлено фрагмент із сформованого звіту у форматі PDF.

Звіт з аналізу успішності — Analytics

Діапазон балів	Кількість	Частка %
0–49 (критично)	1	0.8
50–59 (незадовільно)	10	8.3
60–74 (добре)	50	41.7
75–84 (добре+)	50	41.7
85–100 (відмінно)	9	7.5

Слухачі, що потребують уваги (підсумок < 60)

№	Студент	Відвід. %	Домашні	Лаб.	Мідтерм	Підсумок
40	С***я Са****ко	55	55.8	51	40	46.5
48	С***я Ів****ко	62.2	50.6	68.4	44.7	56
60	М***я Л*****о	57.9	80	49.5	46.9	56.1
5	О***а Р*****к	58.5	78.3	59.6	45.9	56.9
34	А****й Ше****ко	89.3	52.3	48	51.3	57
69	Д****о Р*****к	93.9	54.5	52.1	47	58
73	С***я Са****ко	76	54.9	71.3	41.6	58.6
4	А**а Ів****ко	55.6	57.7	85.7	48.5	59.2
74	Ю**я Тк****ко	77	65.2	69.2	45.3	59.3
116	Т***с Са****ко	58.3	51.8	48.9	66.2	59.3
38	І**н М*****к	67.6	72.3	46	57.9	59.7

Висока успішність (підсумок ≥ 85)

№	Студент	Відвід. %	Підсумок
117	М***я Ко*****ко	86.6	91.3
67	І**н Ко*****ко	77.1	89.8
46	Ю**я Р*****к	86.9	89
101	С***я Р*****к	76.8	87.7
45	М***я Р*****к	75	87.6
77	С***я Са****ко	84.9	87.4
37	А**а М*****к	88.4	86.9
19	Д****о М*****к	78.9	85.5
7	А**а Ше****ко	58.3	85

Ключові висновки

- Слухачів із підсумком нижче 60 балів: 11 (9.2%).
- Висока успішність (≥ 85): 9 (7.5%).
- Найбільший вплив на результат: мідтермова оцінка та лабораторний цикл.
- Рекомендовано посилити супровід слухачів з відвідуваністю нижче 70%.
- Середній бал по всіх потоках: 73.5.

Рекомендації

Адміністрації доцільно використовувати зведені таблиці нижче для планування консультацій, перерозподілу навантаження кураторів та моніторингу академічної успішності по потоках.

Зведення по навчальних потоках

Навчальний потік	Слухачів	Сер. бал	Сер. відвід. %	Ризик <60	% ризику	Відмінники	% відмін.
Навчальний потік 1	19	72.8	75.5	2	10.5	2	10.5
Навчальний потік 2	22	70.2	79.9	3	13.6	1	4.5
Навчальний потік 3	39	73.6	79.2	5	12.8	4	10.3
Навчальний потік 4	40	75.4	78.2	1	2.5	2	5.0

Рис. 4.20 – Фрагмент сформованого звіту у форматі PDF

Отримані результати можуть бути використані деканатами для своєчасного виявлення здобувачів освіти, які входять до групи ризику щодо академічної неуспішності, та планування відповідних коригувальних заходів. На рівні факультету система може використовуватися для формування зведених звітів щодо успішності студентів за спеціальностями, курсами та академічними групами. Автоматично сформовані статистичні показники та прогнози дозволяють керівництву факультету оперативно оцінювати поточний стан освітнього процесу, аналізувати тенденції зміни успішності та визначати напрями підвищення якості підготовки здобувачів освіти. Особливої актуальності програмний комплекс набуває на рівні навчального відділу та адміністрації університету. У великих закладах вищої освіти аналіз результатів навчання тисяч студентів потребує обробки значних обсягів даних, які надходять від різних структурних підрозділів. Формування консолідованої звітності в ручному режимі може займати від декількох годин до декількох днів залежно від масштабів освітньої установи. Використання розробленого рішення дозволяє скоротити час підготовки таких матеріалів завдяки автоматизованому збору, обробці та узагальненню інформації.

4.5. Висновки до розділу 4

У четвертому розділі представлено розроблене програмне забезпечення для збору та аналізу поведінкових даних здобувачів освіти, реалізоване у вигляді плагіна відеоаналітики, інтегрованого до системи управління навчанням Moodle. Розроблений модуль забезпечує автоматизований моніторинг взаємодії студентів із навчальними відеоматеріалами та накопичення поведінкових показників, серед яких тривалість перегляду відео, кількість пауз, повторних переглядів, переходів між фрагментами контенту та показники відвідуваності навчальних занять. На основі зібраних даних сформовано ознаковий простір, який відображає особливості навчальної поведінки здобувачів освіти та може бути використаний для побудови моделей машинного

навчання прогнозування академічної успішності. Використання поведінкових характеристик як предикторів дозволяє враховувати не лише результати навчання, а й особливості взаємодії студентів з електронними освітніми матеріалами, що підвищує інформативність прогнозних моделей та розширює можливості освітньої аналітики. Своєчасне виявлення студентів групи ризику, дає можливість оперативного реагування на негативні тенденції у навчальному процесі та прийняття управлінських рішень на рівні кафедри, факультету й закладу вищої освіти. Розроблено настільний програмний комплекс для завантаження, обробки та аналізу отриманих даних навчання, прогнозування, візуалізації результатів та автоматизованого формування аналітичної звітності. Реалізовані засоби інтеграції з базами даних, генерації графічних представлень та створення звітів у форматах PDF і Excel забезпечують практичну придатність запропонованого рішення для використання в діяльності навчальних відділів та інших структурних підрозділів закладів освіти. Розроблене програмне забезпечення формує цілісну інформаційну платформу для збору, обробки та інтелектуального аналізу освітніх даних, створюючи основу для розвитку систем підтримки прийняття рішень та раннього виявлення негативних тенденцій у навчальному процесії.

ВИСНОВКИ

1. Визначено структуру інформації, що зберігаються в електронні системі управління навчанням Moodle. Встановлено зв’язки між інформацією про здобувачів освіти, їх діями, навчальними курсами та результатами навчання у LMS Moodle.
2. Досліджено використання сучасних методів машинного навчання для задач прогнозування академічної успішності, яке показало, що найпоширенішими алгоритмами класифікації є: логістична регресія, дерево рішень, наївний класифікатор Баєса, метод опорних векторів, випадковий ліс та нейронні мережі. Встановлено, що підвищення достовірності результатів можливе за рахунок використання під час навчання додаткових даних, що характеризують роботу здобувача з освітніми матеріалами, та створення комбінованих моделей машинного навчання.
3. Побудувано моделі машинного навчання на основі алгоритмів логістичної регресії, опорних векторів, випадкового лісу, наївного Баєсу, нейронної мережі. Найкраща загальна точність (80%) отримана моделлю на основі випадкового лісу, найкращий баланс між точністю, чутливістю та специфічністю продемонстрували моделі на основі випадкового лісу (точність – 0.8, чутливість – 0.925, специфічність – 0.391) та нейронної мережі (MLPClassifier) (точність – 0.79, чутливість– 0.935, специфічність – 0.331).
4. Для LMS Moodle розроблено програмний модуль збору інформації про взаємодію здобувачів освіти з навчальними відеоматеріалами, що дозволяє отримувати та зберігати інформацію про тривалість перегляду відео, кількість зупинок та пауз, перегляд відеоматеріалів в повному обсязі.
5. Перелік ознак з інформацією про результати навчання та відвідування занять доповнено 5-ма ознаками з даними від програмного модулю збору інформації про взаємодію здобувачів освіти з навчальними відеоматеріалами.

6. Набір даних із розширеним переліком ознак використано для навчання моделей створених на основі алгоритмів логістичної регресії, опорних векторів, випадкового лісу, наївного Баєсу та нейронної мережі. Встановлено, що включення до навчального набору інформації про взаємодію здобувачів освіти з відеоматеріалами мало найбільший вплив на достовірність прогнозування моделей на основі випадкового лісу (точність збільшилась на 9,6%, чутливість збільшилась на 1,7%, специфічність збільшилась на 29,9%) та нейронної мережі (точність збільшилась на 9,5%, чутливість зменшилась на 0,8%, специфічність збільшилась на 32,5%).
7. Розроблено 2-рівневу стекінгову модель, що на першому рівні включає моделі на основі алгоритмів лінійної регресії, випадкового лісу та нейронної мережі, а на другому рівні – на основі алгоритму посилення градієнту. Проведено навчання запропонованої 2-рівневої стекінгової моделі з використанням набору даних, що включає інформацію про взаємодію здобувачів освіти з навчальними відеоматеріалами.
8. Встановлено, що достовірність прогнозування результатів навчання за допомогою 2-рівневої стекінгової моделі є кращою за всі інші досліджені у роботі моделі. Зокрема, точність зросла на 14,3% порівняно з логістичною регресією, на 4,56% порівняно з нейронною мережею та на 2,29% порівняно з випадковим лісом.
9. Розроблено програмне забезпечення для створення звітів у PDF та XLS форматах, які містять узагальнену статистичну інформацію, результати прогнозування успішності та детальні таблиці з даними щодо дій здобувачів освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пилипенко В., & Стаценко В. (2023). ПРОГНОЗУВАННЯ АКТИВНОСТІ КОРИСТУВАЧІВ ПЛАТФОРМИ MOODLE НА БАЗІ МЕТОДІВ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical Sciences, 323(4), 257–261. https://www.doi.org/10.31891/2307-5732-2023-323-4-257-261
2. Пилипенко В., & Стаценко В. (2024). ВИКОРИСТАННЯ ТЕСТУ СТЬЮДЕНТА ДЛЯ СТАТИСТИЧНОГО АНАЛІЗУ ДАНИХ ОПИТУВАННЯ КОРИСТУВАЧІВ MOODLE. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (1), 226–230. https://doi.org/10.31891/2219-9365-2024-77-29
3. Стаценко, В. В., & Пилипенко, В. І. (2023). Оцінка ефективності моделі прогнозування активності користувачів Moodle методами машинного навчання. VII Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2023», 2023, с. 28-29.
4. Statsenko, V. V., Pavlenko, V. M., & Pylypenko, V. I. (2023). Choise problem in learning management systems, Digital transformation and technologies for the sustainable development all branches of modern education, science and practice, MANS w Łomży, 125-129.
5. Стаценко, В. В., & Пилипенко, В. І. (2024). АНАЛІЗ ПРОГНОЗНОЇ АНАЛІТИКИ ОСВІТНІХ РИЗИКІВ У СИСТЕМАХ УПРАВЛІННЯ НАВЧАННЯМ. НАПРЯМ № 1 ВОЄННА НАУКА. НАЦІОНАЛЬНА БЕЗПЕКА, 282-285.
6. Liu, M., & Yu, D. (2023). Towards intelligent E-learning systems. Education and Information Technologies, 28(7), 7845-7876.
7. Bradley, V. M. (2021). Learning Management System (LMS) use with online instruction. International Journal of Technology in Education, 4(1), 68-92.
8. Gamage, S. H., Ayres, J. R., & Behrend, M. B. (2022). A systematic review on trends in using Moodle for teaching and learning. International journal of STEM education, 9(1), 9.

9. Bognár, L., & Fauszt, T. (2022). Factors and conditions that affect the goodness of machine learning models for predicting the success of learning. Computers and Education: Artificial Intelligence, 3, 100100.
10. Aleksieva-Petrova, A., Chenchev, I., & Petrov, M. (2019). LMS data collection, processing and compliance with EU GDPR. In EDULEARN19 Proceedings (pp. 6494-6501). IATED.
11. Mikadze, G., Khachidze, M., Lomidze, I., & Tomadze, G. (2024). SECURITY ISSUES IN OPEN ACCESS SOFTWARE MOODLE-A CASE STUDY. Applied Mathematics, Informatics & Mechanics, 29(1).
12. Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M., Correia, L., & J. Tallón-Ballesteros, A. (2023, August). The impact of data normalization on the accuracy of machine learning algorithms: a comparative analysis. In International conference on soft computing models in industrial and environmental applications (pp. 344-353). Cham: Springer Nature Switzerland.
13. Rangineni, S., Bhanushali, A., Suryadevara, M., Venkata, S., & Peddireddy, K. (2023). A Review on enhancing data quality for optimal data analytics performance. International Journal of Computer Sciences and Engineering, 11(10), 51-58.
14. Alier, M., Casañ Guerrero, M. J., Amo, D., Severance, C., & Fonseca, D. (2021). Privacy and e-learning: A pending task. Sustainability, 13(16), 9206.
15. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. Big data analytics, 1, 1-22.
16. Izonin, I., Tkachenko, R., Shakhovska, N., Ilchyshyn, B., & Singh, K. K. (2022). A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain. Mathematics, 10(11), 1942.
17. Zhekova, M. (2023, May). A Process Model for Intelligent Analysis and Normalization of Academic and Educational Data. In International Conference on Information, Communication and Computing Technology (pp. 855-872). Singapore: Springer Nature Singapore.

18. Kumar, G., Basri, S., Imam, A. A., Khowaja, S. A., Capretz, L. F., & Balogun, A. O. (2021). Data harmonization for heterogeneous datasets: a systematic literature review. Applied Sciences, 11(17), 8275.
19. Petropoulos, F., & Siemsen, E. (2023). Forecast selection and representativeness. Management Science, 69(5), 2672-2690.
20. Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. Expert Systems with Applications, 41(2), 321-330.
21. Joarder, A. H., & Islam, A. M. M. (2022). Inclusion Probability in Simple Random Sampling by Hypergeometric Distribution. BL COLLEGE JOURNAL, 4(2), 79–89. https://doi.org/10.62106/blc2022v4i2e1
22. Canchola, J. A., Tang, S., Hemyari, P., Paxinos, E., & Marins, E. (2017). Correct use of percent coefficient of variation (% CV) formula for log-transformed data. MOJ Proteom. Bioinform, 6(4), 10-15406.
23. Whittle, P. (2012). Probability via expectation. Springer Science & Business Media.
24. Ghojogh, B., & Crowley, M. (2019). The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial. arXiv preprint arXiv:1905.12787.
25. Szeghalmy, S., & Fazekas, A. (2023). A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning. Sensors, 23(4), 2333.
26. Nguyen, T. D., Shih, M. H., Srivastava, D., Tirthapura, S., & Xu, B. (2021). Stratified random sampling from streaming and stored data. Distributed and Parallel Databases, 39, 665-710.
27. Saini, M., Jitendrakumar, B. R., & Kumar, A. (2022). Optimum estimator in simple random sampling using two auxiliary attributes with application in agriculture, fisheries and education sectors. MethodsX, 9, 101915.
28. Bej, S., Davtyan, N., Wolfien, M., Nassar, M., & Wolkenhauer, O. (2021). LoRAS: An oversampling approach for imbalanced datasets. Machine Learning, 110, 279-301.

29. Tsamardinos, I., Greasidou, E., & Borboudakis, G. (2018). Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. Machine learning, 107, 1895-1922.
30. Dien, T. T., Luu, S. H., Thanh-Hai, N., & Thai-Nghe, N. (2020). Deep learning with data transformation and factor analysis for student performance prediction. International Journal of Advanced Computer Science and Applications, 11(8).
31. Dhawas, P., Dhore, A., Bhagat, D., Pawar, R. D., Kukade, A., & Kalbande, K. (2024). Big data preprocessing, techniques, integration, transformation, normalisation, cleaning, discretization, and binning. In Big Data Analytics Techniques for Market Intelligence (pp. 159-182). IGI Global Scientific Publishing.
32. Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students' performance by means of machine learning: A review. Applied sciences, 10(3), 1042.
33. Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learning Environments, 9(1), 11.
34. Пилипенко, В., Стаценко, В. (2024). Дослідження точності методів машинного навчання при прогнозуванні успішності здобувачів. Herald of Khmelnytskyi National University. Technical sciences, 335(3 (1)), 349-356. DOI: https://doi.org/10.31891/2307-5732-2024-335-3-47
35. Ljubobratović, D., & Matetić, M. (2019). Using LMS activity logs to predict student failure with random forest algorithm. The Future of Information Sciences, 113.
36. Aleksandrova, Y. (2019). Predicting students performance in moodle platforms using machine learning algorithms. In Conferences of the department Informatics (No. 1, pp. 177-187). Publishing house Science and Economics Varna. F
37. Zacharis, N. Z. (2016). Predicting student academic performance in blended learning using artificial neural networks. International Journal of Artificial Intelligence and Applications, 7(5), 17-29.

38. Tamada, M. M., Giusti, R., & Netto, J. F. D. M. (2022). Predicting students at risk of dropout in technical course using LMS logs. Electronics, 11(3), 468. DOI: https://doi.org/10.3390/electronics11030468
39. Althibyani, H. A. (2024). Predicting student success in MOOCs: a comprehensive analysis using machine learning models. PeerJ Computer Science, 10, e2221. DOI: https://doi.org/10.7717/peerj-cs.2221
40. Injadat M., Moubayed A., Nassif A. B., and Shami A., Multi-split optimized bagging ensemblemodel selection for multi-class educational data mining, Applied Intelligence. (2020) 50, no. 12, 4506–4528,
41. Kumar Veerasamy A., D'Souza D., Apiola M. V., Laakso M. J., and Salakoski T., Using early assessment performance as early warning signs to identify at-risk students in programming courses, Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), 2020, Uppsala, Sweden, https://doi.org/10.1109/FIE44824.2020.9274277.
42. Adekitan A. I. and Noma-Osaghae E., Data mining approach to predicting the performance of first year student in a university using the admission requirements, Education and Information Technologies. (2019) 24, no. 2, 1527–1543, https://doi.org/10.1007/s10639-018-9839-7, 2-s2.0-85058040501.
43. Khasanah A. U., A comparative study to predict student's performance using educational data mining techniques, IOP Conference Series: Materials Science and Engineering. (2017) 215, 012036, https://doi.org/10.1088/1757-899X/215/1/012036, 2-s2.0-85028308730.
44. Daniel, B. (2015). B ig D ata and analytics in higher education: Opportunities and challenges. British journal of educational technology, 46(5), 904-920.
45. Zohair, A., & Mahmoud, L. (2019). Prediction of Student's performance by modelling small dataset size. International Journal of Educational Technology in Higher Education, 16(1), 1-18.

46. Yu-Wei, C., & David, C. (2015). Machine learning with R cookbook: Explore over 110 recipes to analyze data and build predictive models with the simple and easy-to-use R code. Birmingham: Packt Publishing.
47. Zhang, L., & Li, K. F. (2018, May). Education analytics: Challenges and approaches. In 2018 32nd international conference on advanced information networking and applications workshops (WAINA) (pp. 193-198). IEEE.
48. Hellas, A., Ihantola, P., Petersen, A., Ajanovski, V. V., Gutica, M., Hynninen, T., & Liao, S. N. (2018, July). Predicting academic performance: a systematic literature review. In Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education (pp. 175-199).
49. Shahiri, A. M., Husain, W., & Rashid, N. A. A. (2015). A review on predicting student's performance using data mining techniques. procedia computer science, 72, 414-422.
50. Tatar, A. E., & Düştegör, D. (2020). Prediction of academic performance at undergraduate graduation: Course grades or grade point average?. Applied sciences, 10(14), 4967.
51. Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., & Rangwala, H. (2016). Predicting student performance using personalized analytics. Computer, 49(4), 61-69.
52. Cui, Y.; Chen, F.; Shiri, A.; Fan, Y. Predictive analytic models of student success in higher education: A review of methodology. Inf. Learn. Sci. 2019, 120, 208–227.
53. Alyahyan, E., Düştegör, D. Predicting academic success in higher education: literature review and best practices. Int J Educ Technol High Educ 17, 3 (2020). https://doi.org/10.1186/s41239-020-0177-7
54. Alshanqiti, A.; Namoun, A. Predicting student performance and its influential factors using hybrid regression and multi-label classification. IEEE Access 2020, 8, 203827–203844.
55. Rastrollo-Guerrero, J.L.; Gómez-Pulido, J.A.; Durán-Domínguez, A. Analyzing and predicting students' performance by means of machine learning: A review. Appl. Sci. 2020, 10, 1042.

56. Mthimunye, K.; Daniels, F.M. Predictors of academic performance, success and retention amongst undergraduate nursing students: A systematic review. S. Afr. J. High. Educ. 2019, 33, 200–220.
57. Dixson, D.D.; Worrell, F.C.; Olszewski-Kubilius, P.; Subotnik, R.F. Beyond perceived ability: The contribution of psychosocial factors to academic performance. Ann. N. Y. Acad. Sci. 2016, 1377, 67–77.
58. Стаценко В., & Пилипенко В. (2024). ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ МОДЕЛІ ПРОГНОЗУВАННЯ УСПІШНОСТІ МЕТОДАМИ МАШИННОГО НАВЧАННЯ. Herald of Khmelnytskyi National University. Technical Sciences, 331(1), 271-276. https://doi.org/10.31891/2307-5732-2024-331-41
59. Пилипенко, В., Скідан, В., & Воліавач, А. (2024). АНАЛІЗ ОПИТУВАННЯ ЩОДО ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ. Herald of Khmelnytskyi National University. Technical Sciences, 345(6(2), 108-112. https://doi.org/10.31891/2307-5732-2024-345-6-16
60. Pavlenko, V., Ponomarenko, I., Morhulets, O., Fedorchenko, A., Chorna, O., & Pylypenko, V. (2023, October). Creating Educational Products With Using Data Science and Digital Marketing. In 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek) (pp. 1-4). IEEE.
61. Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9(1), 381-386.
62. Morales, E. F., & Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In Biosignal processing and classification using computational learning and intelligence (pp. 111-129). Academic Press.
63. Banerjee, S. (2021). Mathematical modeling: models, analysis and applications. Chapman and Hall/CRC.
64. Dangeti, P. (2017). Statistics for machine learning. Packt Publishing Ltd.

65. McCullagh, P. (2002). What is a statistical model?. The Annals of Statistics, 30(5), 1225-1310.
66. Ayodele, T. O. (2010). Types of machine learning algorithms. New advances in machine learning, 3(19-48), 5-1.
67. Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2016). Machine learning basics. Deep learning, 98-164.
68. Chen, K. (2023, September). Research on Popular Machine Learning Algorithms. In 2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 14-21). IEEE.
69. Soofi, A. A., & Awan, A. (2017). Classification techniques in machine learning: applications and issues. Journal of Basic & Applied Sciences, 13, 459-465.
70. Kumari, R., & Srivastava, S. K. (2017). Machine learning: A review on binary classification. International Journal of Computer Applications, 160(7).
71. Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756.
72. Endut, N., Hamzah, W. A. F. W., Ismail, I., Yusof, M. K., Baker, Y. A., & Yusoff, H. (2022). A systematic literature review on multi-label classification based on machine learning algorithms. TEM Journal, 11(2), 658.
73. Musa, A. B. (2024). Understanding Student Performance in Foundation Year: Insights from Logistic Regression, Naïve Bayes, and Random Forest Models. International Journal of Information and Education Technology, 14(12).
74. Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learning Environments, 9(1), 11.
75. Vijayalakshmi, V., & Venkatachalapathy, K. (2019). Comparison of predicting student's performance using machine learning algorithms. International Journal of Intelligent Systems and Applications, 11(12), 34.
76. Vujović, Ž. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599-606.

77. Li, J. (2024). Area under the ROC Curve has the most consistent evaluation for binary classification. PLoS One, 19(12), e0316019.
78. Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Overfitting, model tuning, and evaluation of prediction performance. In Multivariate statistical machine learning methods for genomic prediction (pp. 109-139). Cham: Springer International Publishing.
79. Thanaki, J. (2017). Python natural language processing. Packt Publishing Ltd.
80. Hackeling, G. (2017). Mastering Machine Learning with scikit-learn. Packt Publishing Ltd.
81. Van Horn II, B. M., & Nguyen, Q. (2023). Hands-on application development with PyCharm: Build applications like a Pro with the ultimate Python development tool. Packt Publishing Ltd.
82. Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. African Journal of Biomedical Research, 4023-4031.
83. D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, Hoboken, NJ: John Wiley & Sons, 2013
84. Albreiki, B., Zaki, N., & Alashwal, H. (2021). A systematic literature review of student'performance prediction using machine learning techniques. Education Sciences, 11(9), 552.
85. Hutter, M. (2021). Learning curve theory. arXiv preprint arXiv:2102.04074.
86. Sun, C., Tian, Y., Gao, L., Niu, Y., Zhang, T., Li, H., ... & Yu, J. (2019). Machine learning allows calibration models to predict trace element concentration in soils with generalized LIBS spectra. Scientific reports, 9(1), 11363.
87. Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In Machine learning (pp. 101-121). Academic Press.
88. Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification: thinking with examples for effective learning (pp. 207-235). Boston, MA: Springer US.

89. Reis, I., Baron, D., & Shahaf, S. (2018). Probabilistic random forest: A machine learning algorithm for noisy data sets. The Astronomical Journal, 157(1), 16.
90. Datta, J., & Ghosh, J. K. (2014). Bootstrap-an exploration. Statistical Methodology, 20, 63-72.
91. Liu, Y., Wang, Y., & Zhang, J. (2012, September). New machine learning algorithm: Random forest. In International conference on information computing and applications (pp. 246-252). Berlin, Heidelberg: Springer Berlin Heidelberg.
92. Yang, F. J. (2018, December). An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI) (pp. 301-306). IEEE.
93. Jiang, W., & Zhang, C. H. (2009). General maximum likelihood empirical Bayes estimation of normal means.
94. Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. Encyclopedia of machine learning, 15(1), 713-714.
95. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
96. Guliyev, N. J., & Ismailov, V. E. (2016). A single hidden layer feedforward network with only one neuron in the hidden layer can approximate any univariate function. Neural computation, 28(7), 1289-1304.
97. Cepowski, T., & Chorab, P. (2021). Determination of design formulas for container ships at the preliminary design stage using artificial neural network and multiple nonlinear regression. Ocean Engineering, 238, 109727.
98. Zarco, M., & Froese, T. (2018). Self-optimization in continuous-time recurrent neural networks. Frontiers in Robotics and AI, 5, 96.
99. Alsalem, M. A., Zaidan, A. A., Zaidan, B. B., Hashim, M., Albahri, O. S., Albahri, A. S, & Mohammed, K. I. (2018). Systematic review of an automated multiclass detection and classification system for acute Leukaemia in terms of evaluation and benchmarking, open challenges, issues and methodological aspects. Journal of medical systems, 42, 1-36.

100. Pylypenko, V., Statsenko, V., Bila, T., & Statsenko, D. (2024). Determining the influence of data on working with video materials on the accuracy of student success prediction models. Eastern-European Journal of Enterprise Technologies, 5(4 (131), 52–62. https://doi.org/10.15587/1729-4061.2024.313333

101. PYLYPENKO, V. (2025). THE EFFECT OF TRAINING SAMPLE SIZE ON THE STABILITY OF CLASSIFICATION MODELS. Technologies and Engineering, 26(6), 32-44. https://doi.org/10.30857/2786-5371.2025.6.3

102. Bisong, E. (2019). Matplotlib and seaborn. In Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners (pp. 151-165). Berkeley, CA: Apress.

103. Sapre, A., & Vartak, S. (2020). Scientific computing and data analysis using NumPy and Pandas. International Research Journal of Engineering and Technology, 7, 1334-1346.

104. PYLYPENKO, V. (2026). IMPACT OF STACKING ENSEMBLE DEPTH ON GENERALIZATION ABILITY OF ACADEMIC PERFORMANCE PREDICTION MODELS. Technologies and Engineering, 27(1), 72-79. https://doi.org/10.30857/2786-5371.2026.1.7

105. Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University-Computer and Information Sciences, 35(2), 757-774.

106. Pylypenko Vladyslav, Statsenko Volodymyr: STACKED ENSEMBLE MACHINE LEARNING ALGORITHM IN PREDICTION OF STUDENT SUCCESS. Proceedings of the II International Scientific and Practical Conference. Copenhagen, Denmark. 2024. Pp. 8-11, URL: https://isg-konf.com/integration-of-science-and-practice-as-a-mechanism-of-effective-development/

107. Pylypenko Vladyslav, Statsenko Volodymyr: INCREASING THE ACCURACY OF PREDICTION OF STUDENT SUCCESS FOR A MODEL WITH A RANDOM FOREST ALGORITHM. Proceedings of the I International Scientific and Practical

Conference. Boston, USA. 2024. Pp. 9-12, URL: https://isg-konf.com/innovative-scientific-research-theory-methodology-practice/

108. Пилипенко, В., & Стаценко, В. (2024). ВИКОРИСТАННЯ ДВОРІВНЕВОГО МЕТОДУ СТЕКОВОГО АНСАМБЛЮ ДЛЯ ПОКРАЩЕННЯ ТОЧНОСТІ ПРОГНОЗУВАННЯ УСПІШНОСТІ. Наука і техніка сьогодні, 9 (37), 763-774. https://doi.org/10.52058/2786-6025-2024-9(37)-763-774

109. Bowers, A. J., & Zhou, X. (2019). Receiver operating characteristic (ROC) area under the curve (AUC): A diagnostic measure for evaluating the accuracy of predictors of education outcomes. Journal of Education for Students Placed at Risk (JESPAR), 24(1), 20-46.

110. Volodymyr Statsenko, Pylypenko Vladyslav, Skidan, Vladyslava, Volivach, Antonina. (2024). Investigation of the Accuracy of Machine Learning Methods in Prediction of Students Success. 1-4. 10.1109/KhPIWeek61434.2024.10877975.

111. Пилипенко, В. І., & Стаценко, В. В. (2025). Прогнозування академічної успішності здобувачів за допомогою методів машинного навчання. IV Міжнародна науково-практична інтернет конференція молодих учених та здобувачів «ЕЛЕКТРОМЕХАНІЧНІ, ІНФОРМАЦІЙНІ СИСТЕМИ ТА НАНОТЕХНОЛОГІЇ», 2025, с.134-135.

112. Volodymyr Pavlenko, Ihor Ponomarenko, Oksana Morhulets, Andrii Fedorchenko, Vladyslav Pylypenko: Use of Information Technologies and Marketing Tools for The Formation of An Educational Platform. ITTAP 2023: 702-708

113. Pylypenko Vladyslav, Statsenko Volodymyr: DEVELOPMENT OF A MOODLE PLUG-IN USING AJAX REQUEST FOR ASYNCHRONOUS DATA TRANSFER. Proceedings of the XXXIII International Scientific and Practical Conference. Seville, Spain. 2024. Pp. 7-14, URL: https://isg-konf.com/scientific-developments-of-young-scientists-to-improve-life/

114. Statsenko Volodymyr, Pylypenko Vladyslav: Development of a Moodle video player plug-in for user interaction analysis. VIII Міжнародна науково-практична

конференція «Мехатронні системи: інновації та інжиніринг» – «MSIE-2024», 2024, с. 266-268.

115. Пилипенко В., & Стаценко, В. (2025). ПЛАГІН ДЛЯ ЗБОРУ ДАНИХ ВЗАЄМОДІЇ КОРИСТУВАЧІВ MOODLE З ВІДЕО МАТЕРІАЛАМИ. Наука і техніка сьогодні, 1(42), 1318-1330. https://doi.org/10.52058/2786-6025-2025-1(42)-1318-1330.

116. Пилипенко, В. І. (2026). Вплив розміру навчальної вибірки на стабільність та узагальнювальну здатність моделей класифікації. Збірник наукових праць IX Міжнародної науково-практичної конференції «Мехатронні системи: інновації та інжиніринг» – «MSIE-2026», (с. 287–290)

ДОДАТОК А. ФРАГМЕНТ ПРОГРАМНОГО КОДУ 2-РІВНЕВОЇ СТЕКІНГОВОЇ МОДЕЛІ ПРОГНОЗУВАННЯ УСПІШНОСТІ ЗДОБУВАЧІВ ОСВІТИ

```
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_predict
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.metrics import (
    classification_report, confusion_matrix, balanced_accuracy_score,
    roc_auc_score, roc_curve, f1_score, accuracy_score,
    precision_score, recall_score
)
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv('data.csv')
feature_cols = [
    'LectionVisitPercent',
    'PracticeVisitPercent',
```

```
    'LabVisitPercent',
    'TotalVisitPercent',
    'Duration',
    'PlayCount',
    'PauseCount',
    'StopCount',
    'Completed'
]
available_cols = data.columns.tolist()
missing_cols = [col for col in feature_cols if col not in available_cols]
if missing_cols:
    print(f"  відсутні колонки: {missing_cols}")
    feature_cols = [col for col in feature_cols if col in available_cols]
print(f"  Використано ознак: {len(feature_cols)}")
print(f"  Ознаки: {feature_cols}")
if 'DisciplineMark' in available_cols:
    marks = data['DisciplineMark']
else:
    raise ValueError("Не знайдено ")

X = data[feature_cols].copy()
```

```
valid_idx = X.dropna().index
X = X.loc[valid_idx].reset_index(drop=True)
marks = marks.loc[valid_idx].reset_index(drop=True)


print(f"  : {len(X)} записів")
print(f"   {marks.min():.0f} - {marks.max():.0f}")
threshold = thValue
y = (marks >= threshold).astype(int)
# Розділення на тренувальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
# Масштабування
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
level1_models = {
    'LogisticRegression': LogisticRegression(
        max_iter=1000, class_weight='balanced', random_state=42
    ),
    'NaiveBayes': GaussianNB(),
```

```
    'SVM': SVC(kernel='rbf', probability=True, class_weight='balanced', random_state=42),
    'RandomForest': RandomForestClassifier(
        n_estimators=100, max_depth=10, class_weight='balanced',
        random_state=42, n_jobs=-1
    ),
    'NeuralNetwork': MLPClassifier(
        hidden_layer_sizes=(100, 50), max_iter=500,
        early_stopping=True, random_state=42
    )
}
# Генерація мета-ознак через крос-валідацію
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
meta_features_train = np.zeros((len(X_train_scaled), len(level1_models)))
meta_features_test = np.zeros((len(X_test_scaled), len(level1_models)))

# Зберігаємо передбачення базових моделей для візуалізації
base_models_predictions = {}
base_models_probabilities = {}
base_models_confusion = {}
for idx, (name, model) in enumerate(level1_models.items()):
    print(f"  Навчання {name}...")
```

```
    # Крос-валідація для тренувальних мета-ознак
    meta_features_train[:, idx] = cross_val_predict(
        model, X_train_scaled, y_train,
        cv=cv, method='predict_proba', n_jobs=-1
    )[:, 1]

    # Навчання на повних даних для тестових мета-ознак
    model.fit(X_train_scaled, y_train)
    meta_features_test[:, idx] = model.predict_proba(X_test_scaled)[:, 1]

    y_pred_base = model.predict(X_test_scaled)
    y_proba_base = model.predict_proba(X_test_scaled)
    base_models_predictions[name] = y_pred_base
    base_models_probabilities[name] = y_proba_base[:, 1]
    base_models_confusion[name] = confusion_matrix(y_test, y_pred_base)

    ba = balanced_accuracy_score(y_test, y_pred_base)
    print(f"    → Balanced Accuracy: {ba:.4f}")

meta_model = GradientBoostingClassifier(
```

```
    n_estimators=100, max_depth=3, learning_rate=0.1,
    random_state=42
)
meta_model.fit(meta_features_train, y_train)
y_pred_2level = meta_model.predict(meta_features_test)
y_proba_2level = meta_model.predict_proba(meta_features_test)[:, 1]
print(classification_report(y_test, y_pred_2level, target_names=class_names))
```

ДОДАТОК Б. ФРАГМЕНТ ПРОГРАМНОГО КОДу ЗАСТОСУНКУ ДЛЯ ОБРОБКИ ДАНИХ ВІДВІДУВАНЬ ЗДОБУВАЧІВ ОСВІТИ

```
public class StudentVisitData
{
    public String studentID;
    public String groupName;

    public Dictionary<string, List<int>> _lectionVisits = new Dictionary<string,
List<int>>();
    public Dictionary<string, List<int>> _practiceVisits = new Dictionary<string,
List<int>>();
    public Dictionary<string, List<int>> _laboratoryVisits = new Dictionary<string,
List<int>>();

    public StudentVisitData(String stID, String grName)
    {
        studentID = stID;
        groupName = grName;
    }

    public string[] disciplineNames()
    {
        List<string> diciplineNames = new List<string>();

        diciplineNames.AddRange(_lectionVisits.Keys);
        diciplineNames.AddRange(_practiceVisits.Keys);
        diciplineNames.AddRange(_laboratoryVisits.Keys);
```

```
        return diciplineNames.ToArray();
    }

    public int CalcTotalVisitPercent(string disciplineName)
    {
        int visitLectCount = 0;
        int nonVisitLectCount = 0;
        int visitLectPercent = 0;

        var lectionVisits = CalcLectionVisitCount(disciplineName, out visitLectCount, out
nonVisitLectCount, out visitLectPercent);

        if (lectionVisits < 0) lectionVisits = 0;

        int visitPractCount = 0;
        int nonVisitPractCount = 0;
        int visitPractPercent = 0;

        var practiceVisits = CalcPracticeVisitCount(disciplineName, out visitPractCount, out
nonVisitPractCount, out visitPractPercent);

        if (practiceVisits < 0) practiceVisits = 0;

        int visitLabCount = 0;
        int nonVisitLabCount = 0;
        int visitLabPercent = 0;
```

```
        var laboratoryVisits = CalcLaboratoryVisitCount(disciplineName, out visitLabCount,
out nonVisitLabCount, out visitLabPercent);
        if (laboratoryVisits < 0) laboratoryVisits = 0;
        var totalVisitCount = lectionVisits + practiceVisits + laboratoryVisits;
        var totalVisits = visitLectCount + visitPractCount + visitLabCount;
        if (totalVisits == 0 || totalVisitCount == 0) return 0;
        return totalVisits * 100 / totalVisitCount;
    }

    public int CalcLectionVisitCount(string disciplineName, out int visits, out int nonVisits,
out int percent)
    {
        visits = 0;
        nonVisits = 0;
        percent = 0;
        List<int> lectionVisits = null;

        if (_lectionVisits.ContainsKey(disciplineName))
        {
            if (_lectionVisits.TryGetValue(disciplineName, out lectionVisits))
            {
                foreach(var visit in lectionVisits)
                {
                    if (visit == 1) visits = visits + 1;
                    else nonVisits = nonVisits + 1;
                }

                percent = visits * 100 / (visits + nonVisits);
```

```
            return lectionVisits.Count;
        }
    }
    return 0;
}

public int CalcPracticeVisitCount(string disciplineName, out int visits, out int nonVisits,
out int percent)
{
    visits = 0;
    nonVisits = 0;
    percent = 0;
    List<int> practiceVisits = null;

    if (_practiceVisits.ContainsKey(disciplineName))
    {
        if (_practiceVisits.TryGetValue(disciplineName, out practiceVisits))
        {
            foreach (var visit in practiceVisits)
            {
                if (visit == 1) visits = visits + 1;
                else nonVisits = nonVisits + 1;
            }

            percent = visits * 100 / (visits + nonVisits);
            return practiceVisits.Count;
        }
    }
```

```
    return 0;
  }

  public int CalcLaboratoryVisitCount(string disciplineName, out int visits, out int
nonVisits, out int percent)
  {
    visits = 0;
    nonVisits = 0;
    percent = 0;
    List<int> laboratoryVisits = null;

    if (_laboratoryVisits.ContainsKey(disciplineName))
    {
      if (_laboratoryVisits.TryGetValue(disciplineName, out laboratoryVisits))
      {
        foreach (var visit in laboratoryVisits)
        {
          if (visit == 1) visits = visits + 1;
          else nonVisits = nonVisits + 1;
        }

        percent = visits * 100 / (visits + nonVisits);

        return laboratoryVisits.Count;
      }
    }
    return 0;
  }
```

```
public void AddVisitData(string disciplineName, string paraType, int isVisit)
{
    List<int> visitValues = null;

    if (paraType == "") return;

    if (paraType == _lectionTagName)
    {
        if (_lectionVisits.ContainsKey(disciplineName))
        {
            if (_lectionVisits.TryGetValue(disciplineName, out visitValues))
                visitValues.Add(isVisit);
        }
        else
        {
            _lectionVisits.Add(disciplineName, new List<int>());
            if (_lectionVisits.TryGetValue(disciplineName, out visitValues))
                visitValues.Add(isVisit);
        }
        return;
    }

    if (paraType == _practiceTagName)
    {
        if (_practiceVisits.ContainsKey(disciplineName))
        {
            if (_practiceVisits.TryGetValue(disciplineName, out visitValues))
```

```
            visitValues.Add(isVisit);
        }
        else
        {
            _practiceVisits.Add(disciplineName, new List<int>());
            if (_practiceVisits.TryGetValue(disciplineName, out visitValues))
                visitValues.Add(isVisit);
        }
        return;
    }

    if (paraType == _laboratoryTagName)
    {
        if (_laboratoryVisits.ContainsKey(disciplineName))
        {
            if (_laboratoryVisits.TryGetValue(disciplineName, out visitValues))
                visitValues.Add(isVisit);
        }
        else
        {
            _laboratoryVisits.Add(disciplineName, new List<int>());
            if (_laboratoryVisits.TryGetValue(disciplineName, out visitValues))
                visitValues.Add(isVisit);
        }
        return;
    }
}
```

ДОДАТОК В. АКТИ ВПРОВАДЖЕННЯ

Затверджую
Проректор з наукової та міжнародної діяльності
Київського національного університету
технологій та дизайну
_____ Людмила ГАНУЩАК-ЄФІМЕНКО
«23» грудня 2024р.

АКТ

про впровадження в навчальний процес результатів

дисертаційної роботи

Пилипенка Владислава Ігоровича

Цей акт складено про те, що результати дисертаційної роботи Пилипенка Владислава Ігоровича «Методи та засоби оцінки освітніх ризиків в системах управління навчанням» впроваджено у навчальний процес кафедри «Інформаційних та комп'ютерних технологій» Київського національного університету технологій та дизайну.

Впровадження результатів дисертаційної роботи полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт. Зокрема при викладанні дисциплін «Проєктування інтерфейсу користувача» та «Комп'ютерні технології та програмування» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за напрямом 121 «Інженерія програмного забезпечення» використано такі результати:

- розроблено плагін відео плеєра для збору даних про взаємодію користувачів системи управління навчанням Moodle;
- проведено аналіз та визначення впливу даних про роботу з відеоматеріалами на точність моделей прогнозування успішності студентів;
- створено модель для прогнозування успішності студентів на ранній стадії.

Декан факультету МКТ,
д.т.н, професор — Борис ЗЛОТЕНКО

Завідувач кафедри ІКТ,
к.т.н, доцент — Владислава СКІДАН

Професор кафедри ІКТ,
д.т.н, професор — Олег НІКОНОВ

Затверджую
Проректор з наукової роботи
Хмельницького національного університету
_____ проф. Олег СИНЮК
«_26_» _____12_____ 2025р.

АКТ

про впровадження в навчальний процес результатів

дисертаційної роботи

Пилипенка Владислава Ігоровича

Цей акт складено про те, що результати дисертаційної роботи Пилипенка Владислава Ігоровича «Методи та програмні засоби підвищення точності прогнозування успішності здобувачів освіти на основі машинного навчання» впроваджено у навчальний процес кафедри комп'ютерних наук Хмельницького національного університету.

Впровадження результатів дисертаційної роботи полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт для отримання даних відеоаналітики при визначенні приросту точності моделей прогнозування успішності. Зокрема при викладанні дисципліни «Інтелектуальний аналіз даних» для студентів першого (бакалаврського) рівня вищої освіти, що навчаються за спеціальністю 122 «Комп'ютерні науки» використано такі результати:

- впроваджено плагін відео плеєра для збору відеоаналітики користувачів Moodle;
- проведено аналіз та визначення впливу даних про роботу з відеоматеріалами на точність моделей прогнозування успішності студентів;
- використано дворівневу модель для прогнозування успішності студентів.

Декан факультету інформаційних
технологій, д.т.н., професор
Тетяна ГОВОРУЩЕНКО

Завідувач кафедри комп'ютерних
наук, д.т.н., професор
Олександр БАРМАК

Професор кафедри комп'ютерних
наук, д.т.н., професор
Едуард МАНЗЮК