файлів та не підтримує real-time аналіз.

Порівняльний аналіз показує, що жодна з існуючих систем не поєднує одночасно можливості Text-to-SQL, автоматичної візуалізації, підключення до real-time баз даних та доступності за ціною. Tableau та ThoughtSpot є надто дорогими, Metabase має обмежений інтелект вибору візуалізацій, Seek AI не візуалізує дані автоматично, а ChatGPT не працює з реальними базами даних. Запропонована система буде заповнювати цю нішу, пропонуючи повний цикл аналізу даних при мінімальних витратах, обмежених лише викликами LLM API.

Для реалізації такого AI-асистента буде використовуватись технологічний стек, що складається з безкоштовних рішень з відкритим вихідним кодом. Архітектура включає фронтенд на React для створення інтерактивного інтерфейсу, бекенд на Node.js для обробки запитів та виконання SQL-запитів до бази даних. Інтелектуальне ядро системи буде покладатися на локально розгорнуті великі мовні моделі, спеціалізовані для генерації SQL. Такий підхід дозволяє уникнути залежності від комерційних API та пов'язаних з ними витрат, забезпечуючи повний контроль над даними та інфраструктурою. Для рендерингу інтерактивних графіків на стороні клієнта будуть використовуватись бібліотеки, наприклад, Chart.js або ECharts, які також є вільним програмним забезпеченням [5].

Висновки. Проведений аналіз показав, що існуючі рішення для аналізу даних не пропонують комплексного підходу, який би поєднував гнучкість запитів природною мовою, автоматичний вибір доречної візуалізації та доступність для широкого кола користувачів. Запропоновано розроблення на базі на відкритих технологіях AI-асистента, який буде автоматизовувати повний цикл роботи з даними. Ключовою перевагою AI-асистента є не лише перетворення тексту в SQL-запит, але й подальший інтелектуальний аналіз результатів для автоматичного вибору оптимального типу візуалізації. Це істотно буде спрощувати роботу нетехнічних користувачів, а також робити аналітику даних більш доступною та дозволяти компаніям приймати обґрунтовані рішення без значних витрат на спеціалізоване програмне забезпечення чи навчання персоналу.

Список використаної літератури

[1] E. Kalliamvakou, "Research: quantifying GitHub Copilot's impact on developer productivity and happiness," GitHub Blog, Sep. 7, 2022 (Updated May 21, 2024). [Online]. Available: https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/ [Accessed: Oct. 19, 2025].

[2] T. Yu et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018, pp. 3911–3921. doi: 10.18653/v1/D18-1425.

[3] "Ask Data User Guide," Tableau Software. [Online]. Available: https://help.tableau.com/current/pro/desktop/en-us/ask_data.htm [Accessed: Oct. 19, 2025]

[4] "Metabase Documentation: Asking questions," Metabase. [Online]. Available: https://www.metabase.com/docs/latest/questions/start.html [Accessed: Oct. 19, 2025]

[5] "How to Architect LLM-Powered Web Apps," Medium. [Online]. Available: https://medium.com/@tommywilczek/how-to-architect-llm-powered-web-apps-99e02fa643e9 [Accessed: Oct. 19, 2025].

УДК 004.891:378.147

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА АЛГОРИТМІВ ДЛЯ ЗАДАЧІ ПРОГНОЗУВАННЯ АКАДЕМІЧНОЇ УСПІШНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ НА ОСНОВІ АНАЛІЗУ ДАНИХ ПРО ВІДВІДУВАНІСТЬ ТА ОЦІНКИ

Волощук М. А., Богатєнкова О. Є. (maksimvolosuk22@gmail.com, oleksandra.bohatienkova@mnau.edu.ua)

Миколаївський національний аграрний університет (Україна)

Проведено порівняльний аналіз двох алгоритмів ансамблевого машинного навчання для задачі прогнозування академічної успішності студентів: Random Forest та Gradient Boosting. Нами згенеровано штучний датасет з 1000 записів, що включає ознаки відвідуваність занять (0-100%),

оцінки за проміжні тести (0-100%) та домашні завдання (0-100%), а також бінарну цільову змінну – успішність (проходження/невдача). Розглянуто принципи функціонування обох алгоритмів та їх застосування до обробки згенерованих даних. Показано, що Random Forest забезпечує точність 84,50%, F1-score 67.37% та AUC 92,21%, а Gradient Boosting демонструє вищу точність 85,50%, F1-score 70,71% та AUC 92,58%. Обґрунтовано доцільність вибору Gradient Boosting для освітніх систем з обмеженими даними та запропоновано гібридні підходи для підвищення ефективності.

Прогнозування академічної успішності студентів є ключовим завданням у сфері нових інформаційних технологій в освіті, оскільки дозволяє автоматизувати процеси раннього виявлення ризиків відставання, оптимізації навчальних планів та персоналізованих рекомендацій у системах дистанційного навчання. Застосування методів машинного навчання у сфері освіти сьогодні активно розвивається не лише в академічних дослідженнях, а й у практичних рішеннях – від систем раннього попередження про ризик академічної неуспішності до адаптивних освітніх платформ. Подібні моделі дозволяють адміністраціям закладів освіти оперативно виявляти здобувачів вищої освіти, яким потрібна додаткова підтримка, а також оптимізувати розподіл навчального навантаження. Важливо, що ефективність таких систем безпосередньо залежить від правильного вибору алгоритму машинного навчання, який повинен не лише точно прогнозувати, але й добре узагальнювати дані за умов обмеженої вибірки, типової для освітніх середовищ.

Із розвитком машинного навчання з’явилися нові методи на основі ансамблевих алгоритмів, які значно підвищили точність прогнозів. Однак існує проблема вибору оптимального алгоритму залежно від характеристик даних (відвідуваність та оцінки) та специфіки застосування. Два домінуючі підходи – Random Forest та Gradient Boosting – базуються на різних принципах і мають відмінні переваги та обмеження, що потребує їх детального порівняльного аналізу.

Метою роботи є порівняльний аналіз алгоритмів Random Forest та Gradient Boosting для задачі прогнозування академічної успішності студентів на основі згенерованого датасету та визначення найбільш оптимального підходу. Для досягнення мети вирішувалися наступні завдання: 1) згенерувати штучний датасет, що відображає реальні освітні дані; 2) дослідити принципи функціонування Random Forest та їх застосування до обробки даних про студентів; 3) проаналізувати можливості Gradient Boosting у моделюванні залежностей між відвідуваністю, оцінками та успішністю; 4) порівняти ефективність обох алгоритмів; 5) обґрунтувати доцільність гібридних підходів.

Random Forest є ансамблевим методом, що базується на побудові множини дерев рішень, де кожне дерево навчається на випадковій підмножині даних та ознак. Це дозволяє зменшити перенавчання та підвищити стійкість моделі до шуму в даних. Завдяки випадковому вибору підмножин ознак кожне дерево у лісі формує унікальну структуру рішень, що дозволяє системі враховувати широкий спектр варіацій у навчальних даних. У результаті, навіть якщо окремі дерева помиляються, загальна ансамблева модель демонструє стабільні показники якості. Такий підхід добре підходить для задач з невеликим обсягом даних, де спостерігається нерівномірний розподіл оцінок або відвідуваності між студентами.

У контексті прогнозування академічної успішності алгоритм ефективно обробляє дані про відвідуваність, оцінки за проміжні тести та домашні завдання, виділяючи ключові патерни, такі як кореляція між відвідуваністю понад 80% та успішністю. Завдяки паралельній обробці дерев, Random Forest демонструє високу швидкість і масштабованість.

Gradient Boosting є послідовним ансамблевим методом, де кожне наступне дерево рішень корегує помилки попереднього, мінімізуючи функцію втрат за допомогою градієнтного спуску. Це дозволяє моделі фокусуватися на складних прикладах, таких як студенти з нестабільною відвідуваністю або коливаннями в оцінках. На відміну від Random Forest, який створює незалежні дерева, Gradient Boosting вибудовує послідовну структуру, де кожне наступне дерево «вчиться» на залишкових помилках попередніх. Це робить алгоритм особливо чутливим до складних закономірностей і слабких сигналів у даних, наприклад, коли незначні зміни у відвідуваності поєднуються з падінням результатів тестів. Завдяки цьому Gradient Boosting часто виявляє приховані взаємозв’язки між навчальними показниками, що недоступні для простіших моделей. У задачах прогнозування успішності Gradient Boosting ефективно враховує нелінійні залежності, досягаючи високої точності в задачах з обмеженими даними.

ами згенеровано штучний датасет з 1000 записів про студентів, що включає три основні ознаки:

- відвідуваність (0-100% – відсоток присутності на заняттях);
- оцінки за проміжні тести (0-100 балів);
- оцінки за домашні завдання (0-100 балів);
- цільова змінна – успішність (1 – проходження курсу, 0 – невдача).

Датасет створено з використанням логістичної функції для моделювання ймовірності успішності з додаванням випадкового шуму для реалістичності. Дані розділено на тренувальну (80%) та тестову (20%) вибірки. Моделі Random Forest та Gradient Boosting навчено з використанням бібліотеки scikit-learn у Python (n_estimators=100 для обох).

Отримані результати базуються на синтетично згенерованому датасеті, який відтворює статистичні властивості реальних освітніх даних, але не містить персональної інформації студентів.

Таблиця 1. Порівняння метрик моделей

Метрика	Random Forest	Gradient Boosting
Accuracy	0,8450	0,8550
F1-Score	0,6737	0,7071
AUC	0,9221	0,9258

Дослідження показало, що Gradient Boosting перевершує Random Forest за всіма метриками: точність на 1%, F1-score на 3,4%, AUC на 0,4%. Це свідчить про перевагу послідовного навчання у моделюванні складних нелінійних залежностей в освітніх даних. Хоча різниця у точності між моделями здається незначною (приблизно 1%), навіть така різниця може мати практичне значення при великій кількості студентів або у випадках, коли необхідно мінімізувати кількість помилкових прогнозів щодо неуспішності. Крім того, вищий показник F1-score у Gradient Boosting свідчить про краще співвідношення між виявленням успішних студентів і правильним визначенням тих, хто потребує підтримки. Це означає, що модель не лише точніше класифікує, але й робить це більш збалансовано, зменшуючи ризик «помилкових тривог». Високе значення AUC (>0.92) демонструє, що обидві моделі ефективно розділяють класи, проте Gradient Boosting краще справляється з випадками, де кореляція між ознаками складніша.

Найбільш ефективним підходом є комбінування алгоритмів: Random Forest для швидкої початкової класифікації та Gradient Boosting для уточнення прогнозів складних випадків.

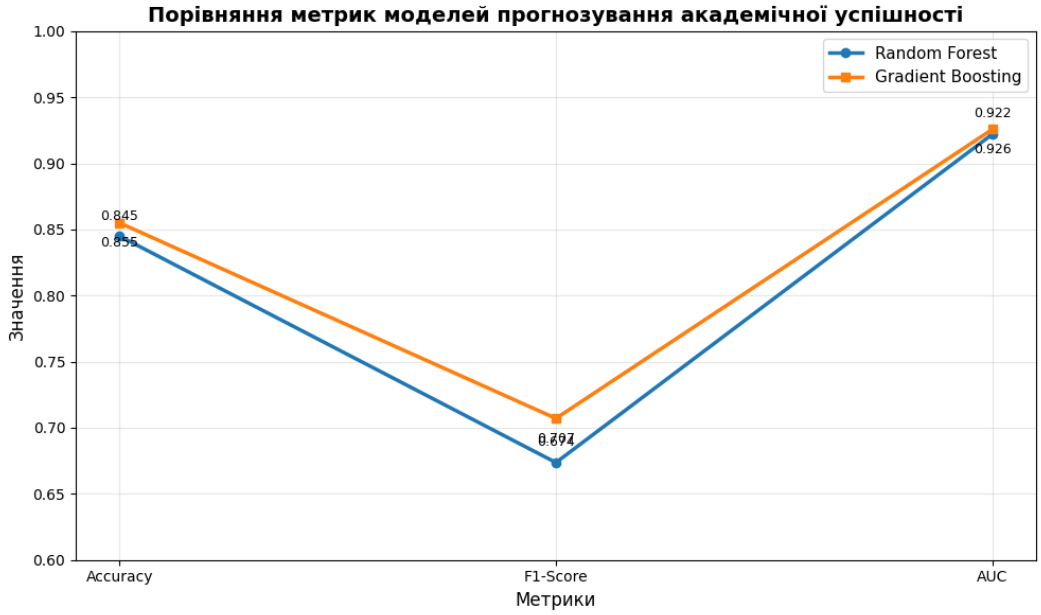

Рис. 1 - Порівняння точності моделей прогнозування академічної успішності

У результаті проведеного дослідження вирішено поставлені завдання та отримано наступні результати:

- згенеровано реалістичний датасет з 1000 записів, що відображає залежності між відвідуваністю, оцінками та успішністю студентів;
- встановлено, що Random Forest забезпечує точність 84,50%, F1-score 67,37% та AUC 92,21% завдяки стійкості до шуму та паралельній обробці;
- доведено, що Gradient Boosting демонструє вищу ефективність (85,50%, 70,71%, 92,58%) у моделюванні нелінійних залежностей освітніх даних;
- обґрунтовано, що оптимальний результат досягається за рахунок гібридного підходу, який поєднує швидкість Random Forest та точність Gradient Boosting.

Важливо зазначити, що всі результати отримані на синтетично згенерованому наборі даних, який моделює типові закономірності у навчальній діяльності студентів. Це дозволило безпечно дослідити властивості алгоритмів без залучення персональної інформації.

У подальших дослідженнях доцільно перевірити узагальнюваність цих висновків на реальних освітніх вибірках, а також оцінити вплив додаткових факторів, таких як тип навчальної дисципліни, формат занять або психологічні аспекти мотивації здобувачів освіти.

Список використаної літератури

[1] Performance Comparison of Xgboost and Random Forest for The Prediction of Student Academic Performance, European Journal of Computer Science and Information Technology, 2025. Дата звернення: 20 жовт. 2025. [Онлайн]. Доступно: https://eajournals.org/ejcsit/wp-content/uploads/sites/21/2025/02/Performance-Comparison.pdf

[2] A Comparative Analysis of Machine Learning Models for Predicting Student Performance, Review of Contemporary Philosophy, 2024. Дата звернення: 20 жовт. 2025. [Онлайн]. Доступно: https://reviewofconphil.com/index.php/journal/article/download/35/17/65

[3] XGBoost To Enhance Learner Performance Prediction, Procedia Computer Science, vol. 240, pp. 123-130, 2024. Дата звернення: 20 жовт. 2025. [Онлайн]. Доступно: https://www.sciencedirect.com/science/article/pii/S2666920X24000572

УДК 004.42

ГЕНЕРАЦІЯ ТЕСТІВ ТА ДОКУМЕНТАЦІЇ У ВЕБІНТЕГРОВАНОМУ СЕРЕДОВИЩІ РОЗРОБКИ З ВИКОРИСТАННЯМ OPENAI API

Ганжа А.С. Антоненко С.В. (hanzha_as@outlook.com)
Дніпровський національний університет імені Олеся Гончара (Україна)

В тезах розглядається застосування великих мовних моделей (LLM), зокрема OpenAI GPT, для автоматизованої генерації юніт-тестів і документації в контексті веб-IDE, реалізованої засобами Flask. Описано архітектуру системи, що забезпечує створення й редагування проєктів, інтеграцію з API мовної моделі, обробку текстових промптів та збереження згенерованих артефактів у структурі файлів. Особливу увагу приділено побудові промптів як ключовому чиннику, що визначає якість результатів генерації. Подано приклади реалізації, а також аналіз переваг і обмежень використання LLM у програмній інженерії. Робота орієнтована на дослідників і розробників, зацікавлених у впровадженні штучного інтелекту в сучасні інструменти розробки ПЗ.

У сучасній практиці розробки програмного забезпечення якість і супровідність коду значною мірою залежать від наявності модульних тестів та зрозумілої документації. Незважаючи на визнану важливість цих складових, їх написання залишається рутинним і часозатратним процесом, який розробники часто відкладають або виконують формально.

З огляду на швидкий розвиток великих мовних моделей (LLM), зокрема GPT-архітектури, з’явилася можливість частково автоматизувати створення тестів та документації до програмного коду. Проте інтеграція таких моделей у повноцінне середовище розробки вимагає вирішення