

В.Є. Величко,

кандидат фізико-математичних наук, доктор педагогічних наук, професор,
ДВНЗ «Донбаський державний педагогічний університет»
<https://orcid.org/0000-0001-9752-0907>

О.С. Ганієв

здобувач магістерського рівня вищої освіти
ДВНЗ «Донбаський державний педагогічний університет»

С.С. Жадан

здобувач магістерського рівня вищої освіти
ДВНЗ «Донбаський державний педагогічний університет»
<https://orcid.org/0000-0002-0683-2196>

ШТУЧНИЙ ІНТЕЛЕКТ ЯК ІНСТРУМЕНТ АНАЛІЗУ І КЛАСИФІКАЦІЇ ЗАДАЧ З ПРОГРАМУВАННЯ В ОСВІТНЬОМУ ПРОЦЕСІ

У статті розглядається використання штучного інтелекту як інструменту для аналізу та класифікації задач з програмування в освітньому процесі. Запропоновано підхід до автоматизації процесу класифікації задач, що враховує їхню складність, тематику та тип. Особливу увагу приділено перевагам застосування AI-технологій для підвищення ефективності навчання програмуванню, зокрема адаптації задач до рівня знань учнів. Наведено практичні результати реалізації системи класифікації задач, які демонструють її високу точність і перспективність. У статті також окреслено можливості подальшого розвитку цього напрямку, такі як створення адаптивних платформ та рекомендаційних систем. Висновки статті підкреслюють значущість інтеграції штучного інтелекту у сучасний освітній процес.

Ключові слова: програмування, шкільний курс інформатики, класифікація задач, штучний інтелект.

V.Ye. Velychko, O.S. Ganiev, S.S. Zhadan

ARTIFICIAL INTELLIGENCE AS A TOOL FOR ANALYZING AND CLASSIFYING PROGRAMMING PROBLEMS IN THE EDUCATIONAL PROCESS

The article considers the use of artificial intelligence as a tool for analyzing and classifying programming problems in the educational process. An approach to automating the problem classification process is proposed, which takes into account their complexity, subject matter, and type. Particular attention is paid to the advantages of using AI technologies to increase the efficiency of programming training, in particular, adapting

problems to the level of students' knowledge. Practical results of implementing the problem classification system are presented, which demonstrate its high accuracy and prospects. The article also outlines the possibilities for further development of this area, such as the creation of adaptive platforms and recommender systems. The conclusions of the article emphasize the importance of integrating artificial intelligence into the modern educational process.

Keywords: programming, school computer science course, problem classification, artificial intelligence.

Постановка проблеми в загальному вигляді. У сучасному світі технології розвиваються швидкими темпами, і програмування стає ключовою професійною компетенцією в багатьох сферах. Проте навчання програмуванню часто стикається з рядом викликів, через різний рівень підготовки учнів, об'єктивні та суб'єктивні складнощі з розумінням алгоритмічного мислення, недостатність персоналізованого підходу до навчання. Генеративні нейронні мережі, які частіше називають “Штучний інтелект” (ШІ) пропонує нові підходи для вирішення цих проблем, роблячи процес навчання більш гнучким, ефективним і доступним.

Навчання програмуванню є невід'ємною частиною шкільного курсу інформатики, головною задачею якого є формування алгоритмічного мислення, логіки та навичок вирішення проблем, розв'язування задач за допомогою комп'ютерної техніки та інформаційних технологій. Однак викладачі та учні стикаються з низкою труднощів, пов'язаних із вибором і розумінням задач, які відповідають різним рівням підготовки. У цьому контексті класифікація задач стає важливим інструментом для підвищення ефективності навчального процесу.

З одного боку ШІ є помічником в навчальній діяльності учня, а з іншого може слугувати помічником викладача з питання класифікації задач, що використовуються під час вивчення програмування. З цієї точки зору є багато проблем, які необхідно розв'язати. Метою статті є дослідження можливості використання штучного інтелекту для аналізу та класифікації задач з програмування, оцінка його потенціалу у вдосконаленні освітнього процесу, визначення ключових переваги, викликів та перспектив впровадження таких технологій у навчанні програмуванню.

Виклад основного матеріалу. Системи на базі ШІ здатні аналізувати прогрес учня, виявляти його слабкі сторони та пропонувати індивідуальні завдання для закріплення знань. Особливо це важливо в умовах класів із різнорівневими групами, цифровою нерівністю та психологічними особливостями провадження навчальної діяльності учнів. Окрім того, ШІ може автоматично перевіряти правильність виконання завдань, аналізувати ефективність алгоритмів та надавати конструктивні рекомендації. Це знижує навантаження на викладачів і дає учням миттєвий зворотний зв'язок, а не тоді, коли викладач має на це час. Популярні інтерактивні навчальні платформи з елементами ШІ, такі як чат-боти чи віртуальні асистенти, намагаються допомогти учням краще розуміти алгоритми через покрокове пояснення задач і аналіз помилок. Не менш важливим моментом є

можливість інтегрувати гейміфікацію в процес навчання. ШІ може задіяти механізми гейміфікації для підвищення мотивації учнів, створюючи адаптивні ігри чи завдання, засновані на рівні їх знань. Велике значення має самоосвітня діяльність. Платформи з елементами ШІ стають у нагоді в цьому виді діяльності, особливо це важливо для регіонів із обмеженим доступом до кваліфікованих викладачів. В глобальному значенні, інтеграція ШІ в освітній процес відповідає сучасним тенденціям цифровізації та розвитку STEM-освіти. Це допомагає не лише навчити учнів основам програмування, а й підготувати їх до майбутніх професійних викликів, пов'язаних з автоматизацією та використанням технологій ШІ.

Хоча використання ШІ має значний потенціал, все ще залишається ціла низка проблем. Перш за все це етичне використання даних учнів. Платформи на яких є курси з програмування збирають дані користувачів. Ці дані можуть бути використані не за призначенням, а саме, надсилання рекламних повідомлень, повідомлень шкідливого і неетичного характеру тощо. Контролювати цей процес надзвичайно складно, а тому необхідно обирати ті платформи, що вже мають добру репутацію, та створили власний імідж, який постійно підтримується.

Не менш важливим питанням є етичне використання згенерованого контенту, коли з'являються відповіді на питання, розв'язки задач, опис вирішення проблем тощо. Необхідно на законодавчому рівні вирішити це питання і позначати контент, що був згенерований нейронною мережею. Є випадки, що згенерована нерелевантна інформація призводила до негативних наслідків в навчанні.

Не обговорюючи якість створеного, ми спостерігаємо всебічну довіру, до інформації, яку повідомив штучний інтелект. Треба розуміти, що на сьогодні "штучний інтелект" є комерційним терміном, терміном, що потрібен для продажу послуг доступу до нейронної мережі. Жодних нових ідей, окрім вже існуючих, на яких мережа "навчалась" згенеровано не буде. Тому, необхідно критично відноситись до отриманого контенту і використовувати його після всебічного аналізу.

Платформи, що використовують генеративні нейронні мережі створюються із залученням великих ресурсів. А тому власники бажають не тільки повернути вкладені кошти, а і отримати прибуток. Рекламні компанії стосовно платформ зі штучним інтелектом теж підпадають під категорію великобюджетних. І все це відбувається для того, що можна було отримати прибуток від створених платформ. Це треба розуміти і критично відноситись до пропозиції щодо безкоштовного використання. Одна справа, що цей ресурс пропонує велика міжнародна корпорація, зовсім інша справа, коли ресурс пропонує для використання майже нікому не відомий стартап.

Не менш важливим є і кваліфікація тих, хто використовує генеративні нейронні мережі. Не зважаючи на значні успіхи, що пов'язані з впровадженням великої мовної моделі [1, 2] широке застосування ШІ в освітній діяльності поки не спостерігається. Практикуючі вчителі ще не мають практичного досвіду застосування ШІ, а поодинокі епізодичні випадки є, скоріш за все, виключенням з твердження про те, що нейронні мережі вже вплинули на сучасну освіту. Процес навчання практикуючих вчителів відбувається, існують курси підвищення

кваліфікації, є платформи де можна отримати інформацію про штучний інтелект та його можливості. Необхідно констатувати, що масового використання ШІ в освіті ще не відбувається. А тому є соціальний запит щодо підготовки майбутніх вчителів та практикуючих до застосування елементів штучного інтелекту в освітній діяльності. Попри це, інвестиції в розробку і впровадження ШІ у навчання програмуванню є важливим кроком для покращення якості освіти та підвищення її ефективності.

У шкільних класах часто навчаються учні з різним рівнем знань, досвіду та здібностей в програмуванні. Чітка класифікація задач дозволяє підбирати завдання, що відповідають поточному рівню учня, що відповідає адаптивному навчанню. Забезпечувати поступовий перехід від простих задач до складніших, навіть в одній темі, що є корисним для засвоєння матеріалу. Окрім того класифікація задач надає можливість уникати ситуацій, коли учень відчуває перевантаження або, навпаки, брак задач, завдань, проблем тощо. Не менш важливим фактором є вивчення навчального матеріалу блоками. Під час використання методу проблемного навчання нова задача буде говорити про те, що переходимо до нового, ще не вивченого блоку навчального матеріалу.

Класифікація задач не тільки за тематикою а і за рівнем складності надає викладачу інструмент для поточного та підсумкового оцінювання учня. Якщо перейти від складності задачі до кількості балів за її розв'язок, то ми отримуємо допоміжний інструмент достатньо об'єктивного оцінювання рівня знань та вмінь учня. Окрім того, його можна використовувати для виявлення більш здібних учнів, додаткова підготовка яких буде результативною при участі у олімпіадах зі спортивного програмування.

Для тих учнів, що вивчають програмування самостійно класифіковані задачі є важливим ресурсом який можна використовувати для побудови освітньої траєкторії. Коли ми починаємо щось вивчати і відразу потрапляємо на складну задачу, то відсутність швидкого успіху знижує нашу мотивацію. В деяких випадках, учні взагалі можуть відмовитись від вивчення програмуванню, вирішивши що воно є занадто складним для них. Починаючи з простих задач підтримується мотивація на достатньому для самоосвітньої діяльності рівні, і з підвищенням складності задач, підвищується і мотивація до самостійного вивчення програмування.

Попри очевидні переваги, використання класифікованих задач у шкільному курсі інформатики, існує і декілька проблем, що потребують вирішення. По-перше, це брак систематизованих ресурсів. Не всі навчальні матеріали мають чітку класифікацію і підходи до класифікації теж різні. По-друге, це відома проблема недостатньої підготовки викладачів у напрямі програмування. Не всі вчителі мають навички і знання для викладання змістової лінії алгоритмізація та програмування, і відповідно, щоб правильно оцінювати складність задач. По-третє, це доволі обмежений час на вивчення цієї змістової лінії, а враховуючи вищевказане, є випадки скорочення часу на вивчення програмування.

Розглянемо можливий варіант реалізації теоретичних міркувань. На початковому етапі нам потрібно зібрати якомога більше різноманітних задач. Джерелами задач можуть бути шкільні підручники, онлайн-платформи вивчення

програмування (наприклад, Codeforces, LeetCode, Codewars, HackerRank, AtCoder тощо), масові онлайн-курси відповідної тематики (наприклад, Coursera, edX), спільноти та форуми (наприклад, Stack Overflow, Reddit), тематичні платформи для конкретних мов програмування, платформи для проведення олімпіад та змагань (наприклад, Eolymp, Algotester), сайти із задачами з програмування.

Наступним етапом є підготовка до автоматизації класифікування задач. Необхідно занести кожну задачу в електронну таблицю (наприклад, у Google Sheets або Excel, за наявності). Для кожної задачі додайте колонки:

- Текст задачі;
- Категорія задачі (наприклад, “цикли”, “масиви”, “рядки”, “об’єкти” тощо);
- Рівень складності (наприклад, “легкий”, “середній”, “складний”);
- Тип задачі (алгоритмічна, практична, творчо-аналітична тощо).

Наступним етапом є інтеграція процесу передачі тексту умови задачі до боту ChatGPT та запису отриманої відповіді до таблиці. Для цього етапу можна використовувати Zapier або Make (Integromat).

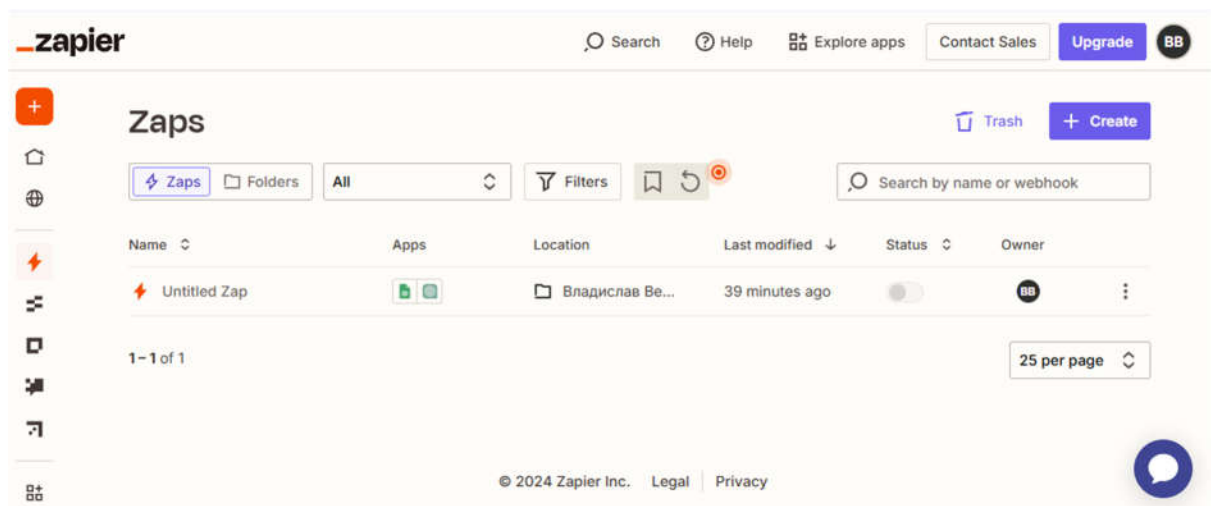


Рис.1. Інтеграція ChatGPT та Google Sheets

Виявляється, що компанія Google та її партнери поіклувалась про своїх клієнтів розробивши декілька розширень, які надають можливість користуватись ChatGPT, безпосередньо з хмарного сервісу. Для прикладу, розширення SheetGPT (див. Рис.2), яке після встановлення та включення надає можливість використання нової функції GPT в такому форматі

=GPT(prompt; model; temperature; maxTokens; cache),

де:

prompt - підказка або запитання до GPT;

model - [необов'язково] Яку модель OpenAI GPT використовувати. Може бути одним із таких: «gpt-4o-mini» (за замовчуванням), «gpt4-turbo», «gpt-4o», «gpt-4», «gpt-3.5-turbo». Див. <https://beta.openai.com/docs/models> для опису кожного.

temperature - [необов'язково] Рівень випадковості або «креативності» згенерованого тексту як значення від 0 до 1, де 0 = відсутність креативності, 1 =

максимальна креативність. Щоб отримати докладнішу інформацію, див. <https://beta.openai.com/docs/api-reference/completions/create#completions/create-temperature>.

maxTokens - визначає максимальну кількість tokenів (слів, частин слів або символів), що можуть бути у відповіді моделі

cache - [необов'язково] Чи кешувати цей результат для повторного використання для всіх відповідних ідентичних підказок. За замовчуванням значення true;

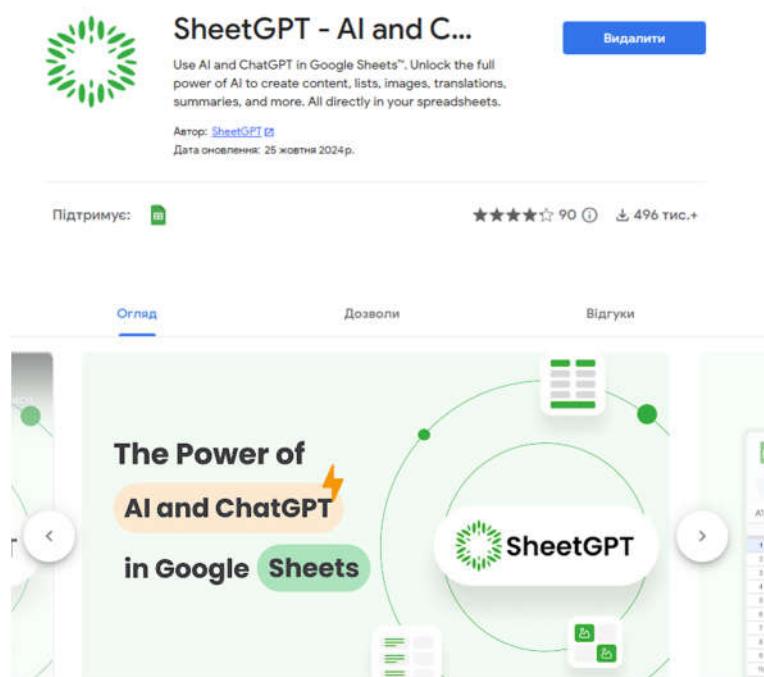


Рис.2. Розширення для Google Sheet

Для використання розширення SheetGPT ми створили таблицю наступної структури: № з/п; Умова задачі; Категорія задачі; Рівень складності; Тип задачі. Не обмежуючи відповідь від GPT через параметр maxTokens заповнили формулами відповідні клітинки таблиці, а саме:

```
=gpt("Визначи одним словом категорію задачі такої умови "&B2& " визначивши категорію задачі серед)")
```

```
=gpt("Визначи рівень складності задачі такої умови "&B2&"Наприкінці відповіді дай висновок окремим рядком такого змісту 'Рівень складності:' і далі число від 1 - легка, до 5 - важка")
```

```
=gpt("Визначи тип задачі такої умови "&B2)
```

Так як GPT дуже любить “поговорити”, то отримали такі відповіді для невеликої кількості задач (див. Рис. 3). Необхідно зазначити, що при одних і тих самих параметрах час від часу можемо отримати відповідь різного змісту.

D2					
=gpt("Визначи рівень складності задачі такої умови: "6035"Наприклад відповіді дай висновок окремим рядком такого змісту "Рівень складності: 1" і дати число від 1 - легка, до 5 - важка")					
	A	B	C	D	E
1	N з/n	Умова задачі	Категорія задачі	Рівень складності	Тип задачі
2				Задача "Надрукувати 'Hello World!'" є базовою задачею, яка часто використовується для початкового ознайомлення з програмуванням. Фраза вимагає знання лише основних концепцій, таких як синтаксис мови програмування, використання команди виводу та, в деяких випадках, розуміння середовища розробки. Оскільки ця задача є дуже простою і зрозумілою, її можна віднести до легкого рівня складності. Рівень складності: 1	Задачу "Надрукувати 'Hello World!'" можна віднести до базової, яка часто використовується для ознайомлення з синтаксисом мови програмування та основними методами виведення інформації на екран. Це може бути також і для навчання. Тип задачі: **Задача на виведення інформації**
3		Дана послідовність чисел, що закінчується числом 0. Визначити серед цієї послідовності найбільше число	Задача є "пошуку".	Задача полягає в обробці послідовності чисел, що закінчується на 0, та у визначенні найбільшого з цих чисел. Ця задача вимагає простих навичок програмування, зокрема вміння працювати з циклами та умовами. Загальний підхід включає зчитування чисел, порівняння їх для знаходження максимального значення, і завершення обробки при досягненні числа 0. Враховуючи вищезазначене, я б оцінив рівень складності цієї задачі на: Рівень складності: 2	Задача, яку ви описали, є задачею на пошуку. Такі задачі можна віднести до обробки послідовностей або масивів алгоритмічне мислення. Додатково, ця задача на пошук, де пот найбільше число в заданій послідовності до появи спеціального випадку числа 0, є сигналом про закінчення вводу.
4		Написати програму, що додає два числа в формі нормальних дробів	Алгебра.	Для задачі "Написати програму, що додає два числа в формі нормальних дробів" можна оцінити складність за кількома критеріями: 1. **Знання основ програмування** - Необхідно знати основи програмування, такі як змінні, операції і умовні конструкції. 2. **Розуміння дробів** - Потрібно знати, як працювати з дробами, зокрема як їх додавати, враховувати спільний знаменник, скорочувати дроб. 3. **Валідність даних** - Потрібно враховувати всі можливі випадки (наприклад, нуль в знаменнику). 4. **Алгоритмічні навички** - Вимагається написати алгоритм, який коректно виконує додавання дробів. Враховуючи ці аспекти, задачу можна вважати середньою, оскільки необхідно поєднати базові знання програмування з математичними операціями. Рівень складності: 3	Задача, яку ви описали, відноситься до категорії задач програм арифметичних операцій з дробами. Конкретно, ця задача є ре операції - додавання, але в контексті роботи з дробами (які мають вигляд і чисельника та знаменника). Основні аспекти задачі: 1. **Тип даних** - Нормальні дробі (чисельник і знаменник). 2. **Операція** - Додавання двох дробів. 3. **Алгоритм** - Зазначай для додавання дробів потрібно при знаменника, а потім виконати операцію над чисельниками. Отже, ця задача є класичним прикладом задачі на обробку мв (дробів) та виконання арифметичних операцій над ними.
5		Знайти найбільше ціле кратне 5 чотирьох чисел	Оптимізація.	Задача "Знайти найбільше ціле кратне чотирьох чисел" є цікавою математичною задачею, яка потребує знання про кратність та найбільше спільне кратне (НСК). Щоб знайти найбільше кратне для чотирьох чисел, потрібно виконати ряд обчислень, в тому числі знаходження НСК для всіх чотирьох чисел, що може вимагати вміння працювати з розкладом чисел на прості множники. Враховуючи ці фактори, можна оцінити цю задачу як задачу середньої складності. Рівень складності: 3	Задача, в якій потрібно знайти найбільше ціле кратне чотирьох чисел, потрібно виконати ряд обчислень, в тому числі знаходження НСК для всіх чотирьох чисел, що може вимагати вміння працювати з розкладом чисел на прості множники. Враховуючи ці фактори, можна оцінити цю задачу як задачу середньої складності. Таким чином, тип задачі можна визначити як задачу на обчислення спільного кратного для кількох чисел.

Рис.3. Таблиця в Google Sheets з результатами запиту до GPT

Проведені нами дії надали можливість створити простий і зрозумілий алгоритм для класифікації задач з програмування для шкільної інформатики. Наразі триває підбір задач та покращення запитів до GPT з метою покращення взаємодії.

Висновки. Застосування штучного інтелекту для аналізу і класифікації задач з програмування значно підвищує ефективність освітнього процесу. Автоматизація таких завдань дозволяє викладачам більше зосередитися на розробці навчальних стратегій, залишаючи рутинну класифікацію алгоритмам. Використання ШІ допомагає створювати індивідуалізовані навчальні траєкторії для учнів, забезпечуючи відповідність задач рівню знань кожного студента. Це сприяє більш глибокому розумінню основ програмування і розвитку алгоритмічного мислення. Запровадження штучного інтелекту в освітній процес відповідає сучасним тенденціям цифровізації, роблячи навчання програмуванню інноваційним та інтерактивним. Система класифікації задач, основана на ШІ, демонструє високу точність при розпізнаванні категорій задач і визначенні рівня їх складності. Це доводить перспективність подальшого розвитку таких інструментів для автоматизації освітніх процесів. Використання штучного інтелекту в аналізі задач відкриває нові можливості для освітніх досліджень, таких як: створення адаптивних навчальних платформ, розробка рекомендаційних систем для вибору задач та інтеграція AI в шкільні й університетські курси інформатики.

Таким чином, впровадження штучного інтелекту у процес навчання програмуванню є перспективним напрямом, що дозволяє значно покращити якість освіти, адаптуючи навчальні матеріали до потреб кожного учня і сприяючи розвитку сучасних технологій в освітній сфері.

Список використаних джерел

1. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing*, Vol.4, Issue. 2, 100211. <https://doi.org/10.1016/j.hcc.2024.100211>
2. Murugesan, S. and Cherukuri, A.K. (2023). The Rise of Generative Artificial Intelligence and Its Impact on Education: The Promises and Perils. *Computer*, 56(5), pp.116–121. <https://doi.org/10.1109/mc.2023.3253292>
3. Величко В.Є., Ананьєв М.С., Іванюк С.В., Шеремет М.М. Електронне навчання у процесі вивчення програмування: Інформатика та методика її навчання. *Збірник наукових праць фізико-математичного факультету ДДПУ*, 2023, 13: 54-61. URL: <https://doi.org/10.31865/2413-26672415-3079132023295330>