УДК 004.412:519.237.5

МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ РАННЬОГО ОЦІНЮВАННЯ КІЛЬКОСТІ РЯДКІВ КОДУ В ОСТАННІХ ВЕРСІЯХ UNITY ЗАСТОСУНКІВ НА C#

Приходько С.Б., д.т.н., проф.[1]; Дроздов Д.С.[2]
[1,2] Національний університет кораблебудування імені адмірала Макарова
[1,2] Україна, Миколаїв
[1]sergiy.prykhodko@nuos.edu.ua; [2]251186@nuos.edu.ua

Анотація. *В роботі на основі перетворення у вигляді десяткового логарифму побудовано три однофакторні нелінійні регресійні моделі для раннього оцінювання кількості рядків коду останніх версій Unity-застосунків на C# за кількістю класів попередніх версій.*

Ключові слова: *оцінювання кількості рядків коду; Unity; C#; нелінійна регресійна модель; відстань Махаланобіса.*

Вступна частина. Оцінювання розміру програмних застосунків є важливою задачею на ранній стадії їх розробки, оскільки ця інформація використовується для прогнозування трудомісткості створення програмного забезпечення (ПЗ) за допомогою математичних моделей, наприклад, таких як COCOMO II.

На теперішній час відомі нелінійні регресійні моделі для оцінювання кількості рядків коду останніх випусків програмних застосунків [1-3]. Але ці моделі не розраховані для використання саме для Unity-застосунків на C#. А значна частина ігрових та інтерактивних застосунків зараз створюється на рушії Unity із використанням мови C#. Тому виникає потреба у побудові відповідних моделей для раннього оцінювання кількості рядків коду наступних версій Unity-застосунків на C#.

Метою дослідження є побудова математичних моделей для раннього оцінювання кількості рядків коду останніх версій Unity-застосунків на C# за кількістю класів їх попередніх версій.

Основна частина. Розглянемо побудову моделей для оцінювання кількості рядків коду Y останніх версій застосунків в залежності від кількості класів X попередніх версій. Для цього Unity-проєкти з відкритим кодом відбиралися на сайті GitHub за допомогою GitHub API за запитом «Unity game language:C# stars:>10». Далі залишались репозиторії розміром до 5 ГБ, що мають структуру Unity-проєкту (каталоги Assets і ProjectSettings та файл ProjectVersion.txt у каталозі ProjectSettings) і щонайменше 20 файлів на C#. Метрики отримано за допомогою власного аналізатора на C#, що використовує бібліотеку Roslyn Microsoft.CodeAnalysis.CSharp: кількість класів визначалась як кількість оголошень класів у синтаксичному дереві, а кількість рядків коду – як кількість фізичних рядків у C#-файлах. Службові та згенеровані каталоги (.git, .vs, Library, Logs, Temp, obj, Build, Builds, UserSettings) не аналізувались.

Для кожного з 98 застосунків розглядались версії V0, V20, V40, V60, V80 та V100 (V0 – перший коміт, V100 – остання версія), а точками даних були пари сусідніх версій одного застосунку (485 пар). Пари V0 → V20 було виключено, оскільки V0 відповідає першому коміту і такі пари мають значно більший розкид приросту розміру. Залишилось 388 пар.

Дані нормалізовано десятковим логарифмом. Для виявлення двовимірних викидів використано тест на основі квадрату відстані Махаланобіса (MD^2) для нормалізованих даних [4] із значенням тестової статистики 10,60 (квантиль розподілу хі-квадрат для рівня значущості 0,005 та двох ступенів свободи): на кожному кроці вилучалась точка з найбільшим значенням

MD^2, після чого вектор середніх та коваріаційна матриця перераховувались. Після вилучення 16 викидів залишилось 372 точки, які були розподілені за застосунками на навчальну (298 точок, 80%) та тестову (74 точки) вибірки так, щоб версії одного застосунку не потрапляли в обидві вибірки. Для побудови нелінійної регресійної моделі ми використовували відповідний метод [4], який базується на нормалізуючих перетвореннях та визначенні наявності викидів у регресійні моделі.

У разі застосування нормалізуючого перетворення у вигляді десяткового логарифму лінійна регресійна модель для нормалізованих даних має вигляд

$$\lg Y = a + b \lg X + \varepsilon,$$

де ε – гаусівська випадкова величина з нульовим математичним сподіванням; a та b – оцінки параметрів, отримані методом найменших квадратів. Відповідна нелінійна регресійна модель може бути представлена у формі [1]

$$\hat{Y} = 10^{a+\varepsilon} X^b.$$

Оскільки одна модель для всієї вибірки даних не забезпечувала прийнятної якості, а при її побудові за методом з [4] були викиди, в роботі отримано три нелінійні регресійні моделі. Модель 1 (M1) будувалась за навчальною вибіркою. При цьому, згідно з [4], на кожному кроці перевірялись наявність викидів за MD^2 для нормалізованих даних, нормальність розподілу ε та потрапляння точок у 95% інтервал прогнозування нелінійної регресії. У разі порушення будь-якої умови вилучалась одна точка, після чого модель перебудовувалась. Точки, вилучені при побудові моделі M1, утворили вибірку для моделі 2 (M2), а вилучені з M2 – для моделі 3 (M3). Оцінки параметрів і показники якості отриманих моделей наведено в табл. 1.

Таблиця 1 – Оцінки параметрів і показники якості моделей

Модель	**n**	**a**	**b**	**R^2**	**MMRE (навч.)**	**PRED(0,25) (навч.)**	**MMRE (тест)**	**PRED(0,25) (тест)**
M1	140	1,7969	1,1090	0,9903	0,1146	0,9143	0,2353	0,7879
M2	62	2,0885	1,0703	0,9878	0,1141	0,9839	0,1331	1,0000
M3	53	1,7095	1,0378	0,9856	0,1268	0,8302	0,1049	0,9200

Створені моделі було перевірено за показниками R^2, MMRE та PRED(0,25). Як відомо, допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Для нової версії застосунку модель обирається за метриками попередньої версії: перша з моделей M1, M2, M3, для якої значення MD^2 для нормалізованих даних не перевищує 10,60, а кількість рядків коду лежить у межах інтервалу прогнозування. На тестовій вибірці модель обрано для 67 з 74 точок, при цьому MMRE = 0,173 і PRED(0,25) = 0,866, тоді як одна модель M1 для всіх тестових точок дає MMRE = 0,511 і PRED(0,25) = 0,365. Це свідчить про добру якість побудованих моделей та доцільність їх використання для раннього оцінювання кількості рядків коду останніх версій Unity-застосунків на C#.

Висновки. В роботі удосконалено математичну модель для раннього оцінювання кількості рядків коду останніх версій програмних застосунку в залежності від кількості класів їх попередніх версій, що складається з двох нелінійних регресійних моделей, побудованих на основі десяткового логарифму. Удосконалення здійснено за рахунок уточнення оцінок параметрів відповідних регресійних моделей та додаткового застосування третьої моделі. Це дозволило підвищити достовірність раннього оцінювання кількості рядків коду останніх версій Unity-засто-

сунків на C#. В подальшому планується підвищити достовірність раннього оцінювання кількості рядків коду останніх версій Unity-застосунків на C#, для яких не може бути застосована розроблена математична модель. Це може бути зроблено за рахунок побудови додаткових регресійних моделей.

Використання інструментів ШІ. Інструменти штучного інтелекту (Claude, Anthropic) використовувались для перевірки розрахунків у Microsoft Excel, підготовки допоміжних скриптів для ітераційного вилучення викидів та підготовки чернетки тексту тез. Усі результати перевірено та відредаговано авторами. Наукові положення, висновки та результати є власним інтелектуальним внеском авторів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Гунченко, Ю., Мещеряков, В., Рачинська, А., & Приходько, С. (2025). Регресійні моделі для раннього оцінювання кількості рядків коду різних випусків застосунків з відкритим кодом. *Прикладні питання математичного моделювання, 2* (8), 86-92. https://doi.org/10.32782/mathematical-modelling/2025-8-2-9

[2] Гунченко, Ю., Мещеряков, В., Приходько, С., & Рачинська, А. (2025). Раннє оцінювання кількості рядків коду останніх випусків застосунків з відкритим кодом за метриками їх перших випусків. *Information Technology: Computer Science, Software Engineering and Cyber Security*, *4*, 72-78. https://doi.org/10.32782/IT/2025-4-9

[3] Gunchenko, Y. O., Meshcheryakov, V. I., Prykhodko, S. B., & Rachinska, A. L. (2026). Estimating the number of lines of code of the latest releases of open-source applications using nonlinear regression models. *Applied Aspects of Information Technology, 1* (9), 122–133. https://doi.org/10.15276/aait.09.2026.09

[4] Prykhodko, S., & Prykhodko, N. (2022). A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations. Selected Papers of the VIII International Scientific Conference “Information Technology and Implementation" (IT&I-2021). Workshop Proceedings (IT&I-WS-2021), Kyiv, Ukraine, December 1-3, 2021. *CEUR Workshop Proceedings, 3179*, 156-166. http://ceur-ws.org/Vol-3179/Paper_15.pdf

Prykhodko S.B., Drozdov D.S.

Mathematical models for early estimation of the number of lines of code in the latest versions of Unity applications in C#

Abstract. *In the work, based on the decimal logarithm transformation and the squared Mahalanobis distance, three one-factor nonlinear regression models have been built for early estimation of the number of lines of code of the latest versions of Unity applications in C# based on the number of classes of the previous versions.*

Keywords: *LOC estimation; Unity; C#; nonlinear regression model; Mahalanobis distance.*