

ЗАПОБІГАННЯ ОВЕРФІТІНГУ ПРИ ПОБУДОВІ НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ РАНЬОГО ОЦІНЮВАННЯ МЕТРИКИ LOC ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ НА KOTLIN

Приходько С.Б., д.т.н., проф.;¹ Кольцов А.В.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹ sergiy.prykhodko@nuos.edu.ua; ² andrew.koltsv@gmail.com

Анотація. В роботі на основі перетворення Вох-Сох побудована трьохфакторна нелінійна регресійна модель для оцінювання кількості строк коду застосунків на Kotlin та виконано валідацію побудованої моделі з метою запобігання оверфітінгу.

Ключові слова: оверфітінг; оцінка кількості строк коду програмного забезпечення; Kotlin; нелінійна регресійна модель.

Вступна частина. У математичному моделюванні та машинному навчанні термін оверфітінг (overfitting) характеризує ситуацію, коли модель занадто точно відповідає своїм навчальним даним (training data) та не може робити якісні прогнози на основі будь-яких інших наборів тестових даних (test data), які не використовувалися при її побудові (навчанні). Оверфітінг можна визначити як певну помилку моделювання, яка виникає, коли модель занадто тісно узгоджена з обмеженим набором точок даних. Як наслідок, модель корисна лише для свого початкового набору даних, а не для будь-яких інших наборів даних.

Як відомо [1], проблема оверфітінгу стосується і регресійних моделей. Для запобігання оверфітінгу при побудові моделі існують різні методи, які включають, наприклад, перехресну перевірку (cross-validation), під час якої дані, що використовуються для навчання моделі, поділяються на розділи (partitions), а модель запускається для кожного з них. Потім загальна оцінка помилки осереднюється. Інші методи включають ансамблювання (ensembling), для якого прогнози об'єднуються принаймні з двох окремих моделей. Втім, серед зазначених методів відносно простим у застосуванні є такий, що базується на розбитті набору даних на два: для тренування та для тестування. Це в свою чергу вимагає зібрати більше даних перед побудовою моделі та правильним чином їх розділити. Зазвичай, подібний поділ здійснюється у співвідношенні 80/20, або 70/30, або 60/40. Важливо враховувати, що в контексті задачі побудови нелінійної регресійної моделі після поділу даних у тренувальному наборі даних має залишитись така кількість точок даних, що представляє собою велику вибірку ($N > 30$).

Визначення трудомісткості розробки програмного забезпечення (ПЗ) є актуальним питанням для менеджерів проектів. Цю величину можна оцінити використовуючи таку математичну модель як, наприклад, СОСОМО II, яка в свою чергу вимагає інформацію про розмір ПЗ у строках коду, яка в більшості випадків невідома на ранніх етапах роботи над проектом. Для мови Java відома нелінійна регресійна модель для оцінювання розміру ПЗ [2], але не дивлячись на суттєву схожість мов Java та Kotlin існує потреба побудови нелінійних регресійних моделей під конкретну мову програмування.

Метою роботи є побудова нелінійної регресійної моделі для раннього оцінювання метрики LOC застосунків з відкритим кодом на Kotlin із запобіганням оверфітінгу.

Основна частина. Побудова нелінійної регресійної моделі на першому етапі вимагає пошуку емпіричних даних. Серед розміщених на сайті GitHub застосунків мовою Kotlin,

було знайдено 79 проектів. Для отримання метрик ПЗ було використано інструмент CodeMR (<https://plugins.jetbrains.com/plugin/10811-codemr>). Для побудови нелінійної регресійної моделі використовувались метрики діаграми класів такі, як: розмір застосунку Y у тисячах строк коду, кількість класів X_1 , метрику WMC X_2 та DIT X_3 . Три фактори було обрано через наступне: можливість отримання цих метрик на ранній стадії розробки з діаграми класів, відсутність мультиколінеарності між трьома факторами. Для визначення викидів у даних використовували квадрат відстані Махаланобіса. Для побудови та тестування моделі початковий набір даних з 79 проектів було поділено на дві частини: тренувальну (54 проекти) та тестувальну (25 проектів) з поділом у співвідношенні приблизно 60/40.

Нелінійну регресійну модель було побудовано з використанням чотиривимірного нормалізуючого перетворення Вох-Сох. Отримана нелінійна регресійна модель для тренувального набору даних, має наступні оцінки параметрів $\hat{\lambda}_Y, \hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3$: -0,161328, -0,159392, -0,239815, -1,253682, -6,1084, 1,7898, 0,84037 та 0,04182 відповідно. Оцінка стандартного квадратичного відхилення $\hat{\sigma}_\epsilon$ дорівнює 0,1421. Якість побудованої моделі було оцінено за допомогою розрахунку множинного коефіцієнту детермінації R^2 , середньої величини відносної похибки $MMRE$ та відсотку прогнозованих результатів $PRED(0,25)$. Значення R^2 , $MMRE$ і $PRED(0,25)$ для створеної моделі дорівнюють відповідно 0,9235, 0,1458 і 0,8235, що свідчить про прийнятну якість моделі [3].

Застосувавши отриману модель до тестового (валідаційного) набору даних отримуємо наступні значення R^2 , $MMRE$ і $PRED(0,25)$: 0,9818, 0,1871 та 0,7600 відповідно. Отримані оцінки якості нелінійної регресійної моделі при застосуванні до тестового (валідаційного) набору даних дозволяють зробити висновок про високу якість побудованої моделі.

Висновки. В роботі було побудовано нелінійну регресійну модель регресійної моделі для раннього оцінювання метрики LOC застосунків відкритим кодом мовою Kotlin з уникненням оверфітінгу. Результат застосування отриманої нелінійної регресійної моделі показує задовільні оцінки якості, а також показує доцільність застосування декількох наборів даних для подолання проблеми оверфітінгу. В подальшому планується проводити тестування моделі на більш об'ємних вибірках даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Frost, J. (2020). *Regression Analysis: An Intuitive Guide for Using and Interpreting Linear Models*. Statistics By Jim Publishing.
- [2] Prykhodko, S. B., Prykhodko, N. V., & Smykodub, T. G. (2020). Four-factor non-linear regression model to estimate the size of open source Java-based applications. *Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences*, 1(2), 157–162. <https://doi.org/10.32838/2663-5941/2020.2-1/25>.
- [3] Prykhodko, S. B., Prykhodko, N. V., & Koltsov, A. V. (2024). A nonlinear regression model for early LOC estimation of open-source Kotlin-based applications. *Radio Electronics, Computer Science, Control*, (1), 85. <https://doi.org/10.15588/1607-3274-2024-1-8>.

Prykhodko S.B, Koltsov A.V.

Overfitting prevention when building a nonlinear regression model for early LOC estimation of open-source Kotlin-based applications

Abstract. In work a nonlinear regression model with three predictors for estimating LOC of applications in Kotlin has been built. Addition validation to prevent overfitting was conducted.

Keywords: overfitting; LOC estimation; Kotlin; nonlinear regression model.