

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ
АДМІРАЛА МАКАРОВА
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Кваліфікаційна наукова праця
на правах рукопису

ОРЕХОВ ОЛЕКСАНДР СТАНІСЛАВОВИЧ

УДК 004.412:519.25

ДИСЕРТАЦІЯ

**МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК
КОДУ JAVA-ЗАСТОСУНКІВ ДЛЯ ОЦІНЮВАННЯ ЇХ РОЗМІРУ**

Спеціальність 122 – Комп’ютерні науки
Галузь знань 12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії (PhD)

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

 О. С. Орехов

Науковий керівник Фаріонова Тетяна Анатоліївна, кандидат технічних наук,
доцент

Миколаїв – 2026

АНОТАЦІЯ

Орехов О. С. Математичні моделі для обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії (PhD) за спеціальністю 122 «Комп'ютерні науки» (галузь 12 – Інформаційні технології). – Національний університет кораблебудування імені адмірала Макарова, Міністерство освіти і науки України, Миколаїв, 2026.

Дисертаційна робота присвячена вирішенню науково-практичного завдання підвищення достовірності обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру шляхом побудови відповідних математичних моделей, а саме нелінійних регресійних моделей, та створенню інструментарію інформаційної технології (IT) для обробки інформації з метрик коду, необхідних для оцінювання параметра розміру на ранніх стадіях розробки програмних застосунків за метриками UML-діаграми класів на основі загальної вибірки JAVA-застосунків та вибірки Data Science (DS) і Machine Learning (ML) JAVA-застосунків.

Актуальність дисертаційної роботи полягає у тому, що достовірна обробка інформації з метрик коду JAVA-застосунків, а саме їх розміру, на ранніх стадіях проєктування безпосередньо впливає на прогнозування трудомісткості розробки програмних продуктів. Це дозволяє ефективно вирішувати завдання планування ресурсів, бюджету, часу на створення програмних проєктів. Однак оцінювання розміру програмного забезпечення (ПЗ) на ранніх стадіях проєктування ускладнюється тим, що на початкових етапах розробки така інформація обмежена та її важко виміряти, наслідком чого є низька достовірність отриманих оцінок. Підвищення достовірності оцінювання розміру ПЗ досягається завдяки використанню відповідних математичних моделей, які враховують фактори середовища – мову програмування JAVA та особливості

проєкту, зокрема застосування технологій DS та ML із цією мовою програмування. У зв'язку з цим існує потреба у підвищенні достовірності обробки інформації для оцінювання розміру ПЗ на ранніх етапах розробки шляхом використання метрик програмного коду, які доступні з концептуальної моделі проєкту (UML-діаграми класів).

Метою дослідження є підвищення достовірності обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків на ранніх стадіях проєктування програмного забезпечення за рахунок побудови нелінійних регресійних моделей на основі наборів метрик коду проєктів з відкритим кодом.

Робочою науковою гіпотезою дисертаційного дослідження є твердження, що підвищення достовірності обробки інформації з метрик коду для оцінювання розміру (кількості рядків коду) JAVA-застосунків на ранніх етапах розробки ПЗ досягається шляхом використання негаусівських нелінійних регресійних моделей, які дозволяють описувати зазначений розмір як випадкову величину та врахувати відмінність розподілу даних від нормального.

Для побудови нелінійних регресійних моделей пропонується використовувати підхід на основі застосування багатовимірних нормалізуючих перетворень, які дозволяють врахувати кореляцію між факторами та наблизити розподіл до нормального, що підвищує ймовірність достовірності обробки інформації з метрик коду JAVA-застосунків для оцінювання параметра їх розміру.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих математичних моделей для обробки інформації з метрик коду для оцінювання розміру загальних та DS і ML JAVA-застосунків, із використанням відповідних вибірок проєктів з відкритим кодом;
- сформувати навчальні та тестові вибірки з метрик коду JAVA-застосунків та обґрунтувати вибір незалежних факторів для побудови нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання їх розміру;

- дослідити отримані метрики на відсутність мультиколінеарності за критерієм VIFs, з урахуванням ітеративної обробки багатовимірних даних;
- обрати перетворення для нормалізації шестивимірних векторів характеристик метрик коду JAVA-застосунків на основі аналізу існуючих нормалізуючих перетворень та методів оцінювання їх параметрів;
- удосконалити чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із використанням п'ятивимірного нормалізуючого перетворення Бокса-Кокса на основі загальної вибірки метрик коду JAVA-застосунків з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- побудувати п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із застосуванням шестивимірного нормалізуючого перетворення Бокса-Кокса на основі загальної вибірки метрик коду JAVA-застосунків з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- побудувати п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із застосуванням шестивимірного нормалізуючого перетворення Бокса-Кокса на основі вибірки метрик коду DS та ML JAVA-проєктів з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- розробити інформаційну технологію (SaaS програмне рішення) та методику для обробки інформації з метрик коду, отриманих з UML-діаграми класів, для оцінювання параметра розміру JAVA-застосунків із використанням побудованих п'ятифакторних нелінійних регресійних моделей.

Наукова новизна отриманих результатів полягає у наступному:

- 1) Удосконалено чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків залежно від загальної кількості класів та інтерфейсів (CLASS), загальної кількості унікальних викликів методів у класах (RFC), середнього значення кількості зв'язків між класами (aCBO) та середнього

значення кількості видимих (public та protected) методів на клас (aVMQ) на основі п'ятивимірного перетворення Бокса-Кокса, що дозволило підвищити достовірність оцінювання параметра розміру таких застосунків як негаусівської випадкової величини у порівнянні з існуючими регресійними моделями. Модель, що побудована, у порівнянні з іншими моделями має більший відсоток прогнозованих значень, менше значення середньої величини відносної похибки та менші ширини інтервалу передбачення нелінійної регресії.

2) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків на основі шестивимірного перетворення Бокса-Кокса, в якій, на відміну від існуючих моделей, було розділено метрику загальної кількості класів та інтерфейсів (CLASS) на дві окремі метрики, а саме кількість класів (CLS) та кількість інтерфейсів (INFC), що дозволило підвищити достовірність обробки інформації з метрик коду для оцінювання параметра розміру в порівнянні з існуючими математичними моделями. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має кращі показники якості R^2 , $MMRE$ та $PRED(0, 25)$ та менші ширини інтервалу передбачення та довірчого інтервалу нелінійної регресії.

3) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків на основі шестивимірного перетворення Бокса-Кокса, в якій, на відміну від існуючих моделей, було розділено метрику загальної кількості класів та інтерфейсів (CLASS) на дві окремі метрики коду кількості класів (CLS) та інтерфейсів (INFC) та враховано особливості розробки цієї категорії застосунків, що дозволило підвищити достовірність обробки інформації з метрик коду для оцінювання параметра розміру у порівнянні із використанням одновимірного та п'ятивимірного перетворень Бокса-Кокса. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має більший відсоток прогнозованих значень, менше значення середньої величини відносної

похибки та менші ширини інтервалу передбачення та довірчого інтервалу нелінійної регресії.

4) Отримали подальший розвиток рівняння для визначення границь інтервалів передбачення п'ятифакторної нелінійної регресії для загальних JAVA-застосунків та DS і ML JAVA-застосунків на основі шестивимірних нормалізуючих перетворень Бокса-Кокса. Це у порівнянні з використанням п'ятивимірних перетворень Бокса-Кокса дозволяє зменшити ширини інтервалів передбачення нелінійних регресій для оцінювання параметра розміру відповідних програмних застосунків і тим самим підвищити достовірність оцінювання зазначеного розміру як залежної випадкової величини.

Практичне значення дисертаційної роботи полягає у розробці інструментарію ІТ для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, які можна отримати з концептуальної моделі даних на ранніх етапах розробки програмного проєкту, розрахунку довірчих інтервалів та інтервалів передбачення із застосуванням відповідних п'ятифакторних нелінійних регресійних моделей на основі загальної вибірки JAVA-застосунків, вибірки DS та ML JAVA-застосунків. Для цього мовою програмування Kotlin для отриманих нелінійних регресійних математичних моделей створено ПЗ, що розповсюджується за моделлю software-as-a-service (SaaS) та має відповідний програмний інтерфейс на основі протоколу HTTP.

У вступі дисертації розкрито сутність та значущість науково-практичного завдання обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, обґрунтовано необхідність проведення дослідження за обраною темою. Загальну характеристику дисертації подано в такій послідовності: обґрунтування вибору та актуальність теми дослідження; зв'язок роботи з науковими програмами, планами, темами; мета і завдання дослідження; наукова новизна і практична цінність одержаних результатів; особистий внесок здобувача; апробація результатів дисертації та публікації.

У першому розділі дисертації проведено аналіз існуючих методів та моделей для обробки інформації з метрик коду для оцінювання розміру

JAVA-застосунків та обґрунтовано необхідність проведення дослідження відповідно до поставленої мети.

У другому розділі дисертації на основі огляду та аналізу існуючих нормалізуючих перетворень, які використовуються для побудови нелінійних регресійних моделей, було обґрунтовано вибір перетворення для нормалізації шестивимірних векторів метрик програмного коду загальних JAVA-застосунків, DS та ML JAVA-застосунків.

У третьому розділі дисертації побудовано чотирьох- та п'ятифакторні нелінійні регресійні моделі, їх довірчі інтервали та інтервали передбачення для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із застосуванням багатовимірних нормалізуючих перетворень на основі загальної вибірки JAVA-застосунків, вибірки DS і ML JAVA-застосунків. Проведено порівняння якості обробки інформації з метрик коду для оцінювання розміру (кількості рядків коду) JAVA-застосунків із використанням розроблених нелінійних регресійних моделей.

У четвертому розділі дисертації запропоновано інструментарій ІТ для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків на початкових етапах проектування програмних застосунків за метриками діаграми класів. Для цього розроблено SaaS-рішення мовою програмування Kotlin та засобами фреймворку Spring Boot, яке пропонує API для обробки інформації з метрик коду для оцінювання KLOC JAVA-застосунків із застосуванням отриманих нелінійних регресійних моделей.

Ключові слова: трудомісткість розробки ПЗ, вихідний код, LOC-оцінка, Java, machine learning, математична модель, регресійна модель, нелінійна регресія, негаусівські дані, нормалізуюче перетворення, перетворення Бокса-Кокса, довірчий інтервал, інтервал передбачення, інформаційна технологія, програмне забезпечення.

Список публікацій здобувача за темою дисертації:

1) в яких опубліковані основні наукові результати дисертації:

1. **Орехов, О. С., & Фаріонова, Т. А. (2024).** Аналіз сучасного стану методів оцінювання трудомісткості розробки програмного забезпечення. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 1(494), 102–111. DOI: [https://doi.org/10.15589/znp2024.1\(494\).15](https://doi.org/10.15589/znp2024.1(494).15) (Реєстр наукових фахових видань України, категорія «Б»).

2. **Орехов О. С., & Фаріонова Т. А. (2024).** Математичні моделі для оцінювання розміру JAVA-застосунків. *Вісник Херсонського національного технічного університету*, 2(89), 196-203. DOI: <https://doi.org/10.35546/kntu2078-4481.2024.2.28> (Реєстр наукових фахових видань України, категорія «Б»).

3. **Орехов, О. С. (2024).** Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру JAVA-застосунків на ранніх стадіях розробки. У В. В. Вичужанін (ред.), *Розвитки інформаційно-керуючих систем та технологій* (с. 360–379). Liha-Pres. <https://doi.org/10.36059/978-966-397-422-4> (ISBN 978-966-397-422-4, одноосібний розділ в колективній монографії)

4. **Орехов, О. С., & Фаріонова, Т. А. (2025).** Математична модель для обробки інформації з метрик коду Data Science та Machine Learning JAVA-застосунків для оцінювання їх розміру. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 2(500), 275–285. DOI: [https://doi.org/10.15589/znp2025.2\(500\).37](https://doi.org/10.15589/znp2025.2(500).37) (Реєстр наукових фахових видань України, категорія «Б»).

5. **Орехов, О. С., & Фаріонова, Т. А. (2025).** П'ятифакторна математична модель для обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру. *Вісник Херсонського національного технічного університету*, 4(95,ч.2), 139–146. DOI: <https://doi.org/10.35546/kntu2078-4481.2025.4.2.18> (Реєстр наукових фахових видань України, категорія «Б»).

2) які засвідчують апробацію матеріалів дисертації:

6. Ворона, М. В., & Орехов, О. С. (2023). Методи оцінки трудомісткості розробки програмного забезпечення для agile проєктів. *Матеріали XIV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (20-21 вересня 2023). Миколаїв: НУК імені адмірала Макарова. 438-441.

7. Фаріонова Т., & Орехов О. (2023). Аналіз успішності розробки проєктів програмного забезпечення. *Матеріали міжнародної науково-практичної конференції "Інтелектуальні інформаційні системи в управлінні проєктами та програмами"* (12-15 вересня 2023) Харків: ХНУРЕ. 210–213. DOI: <https://doi.org/10.30837/IISRRM.2023.09>

8. Орехов, О. С., & Фаріонова, Т. А. (2023). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Data Science та machine learning проєктів, які розробляються за допомогою мови програмування Java. *Матеріали IV всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2023)* (30-31 жовтня 2023 р.). Миколаїв: НУК імені адмірала Макарова. 45–47.

9. Орехов, О. С., & Фаріонова, Т. А. (2024). Порівняльний аналіз нелінійних регресійних моделей оцінювання кількості рядків коду JAVA-застосунків. *Матеріали XV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (26-27 вересня 2024). Миколаїв: НУК імені адмірала Макарова. 563 – 566.

10. Oriekhov, O., Farionova, T., Chernova, L., Chernova, L., & Vorona, M. (2024). Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications. *5th International Workshop on IT Project Management, Bratislava, Slovakia, May 22, 2024. CEUR Workshop Proceedings, Vol. 3709, pp. 54-66.* DOI: https://doi.org/10.23939/IW_itpm2024.054 (ISSN 1613-0073. Стаття включена до міжнародних наукометричних баз Scopus та DBLP).

11. Oriekhov, O., Farionova, T., & Chernova, L. (2024), Three-factor nonlinear regression model of estimating the size of JAVA-software. *In Proceedings*

of the International Conference on Information Control Systems and Technologies (ICST-2024) (September 23-25). CEUR-WS. 506-518.
<https://ceur-ws.org/Vol-3790/paper44.pdf>

12. **Oriekhov, O.**, & Farionova, T. (2024). Early KLOC estimation of JAVA-applications by the three-factor nonlinear regression model. In V. Vychuzhanin (Ed.), *Materials of the XI International Scientific and Practical Conference "Information Control Systems and Technologies 2024" (ICST-2024) (23-25 September)*. Odesa: Helvetica Publishing House. pp. 310 – 312.
<https://icst-conf.com/2024.pdf>

13. **Орехов, О. С.**, & Фаріонова, Т. А. (2024). Оцінювання розміру JAVA-застосунків за трьохфакторною нелінійною регресійною моделлю. *Матеріали V всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2024) (30-31 жовтня 2024 р.)*. Миколаїв: НУК імені адмірала Макарова. 39–42.

14. **Oriekhov, O.**, Farionova, T., Chernova, L., & Vorona, M. (2025). A Five-Factor Nonlinear Regression Model for JAVA Applications Size Estimation. 6th International Workshop on IT Project Management, Kyiv, Ukraine, May 22, 2025. CEUR Workshop Proceedings, Vol. 4023, pp. 1-10.
<https://ceur-ws.org/Vol-4023/paper1.pdf> (ISSN 1613-0073. Стаття включена до міжнародних наукометричних баз Scopus та DBLP).

15. **Орехов, О. С.**, & Фаріонова, Т. А. (2025). Сучасні інформаційні технології для обробки метрик коду на ранніх етапах проектування програмного забезпечення. *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці" (25-26 вересня 2025)*. Миколаїв: НУК імені адмірала Макарова. 588 - 590.

16. **Орехов, О. С.**, & Ворона, М. В. (2025). Проект інформаційної технології для обробки інформації з метрик коду JAVA-застосунків. *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці" (25-26 вересня 2025)*. Миколаїв: НУК імені адмірала Макарова. 604 - 607.

17. **Орехов, О. С., & Фаріонова, Т. А.** (2025). Механізми зменшення мультиколінеарності між незалежними факторами при побудові нелінійної регресійної моделі для оцінювання JAVA-застосунків. *Матеріали VI всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2025)* (16-17 листопада 2025). Миколаїв: НУК імені адмірала Макарова. 40–42.

3) які додатково відображають наукові результати дисертації:

18. **Орехов, О. С., & Фаріонова, Т. А.** (2025). Комп'ютерна програма «Сервіс оцінювання інформації з метрик програмних застосунків» (SoftwareMetricsEstimator) [Свідоцтво про реєстрацію авторського права № 135110 від 15.04.2025]. У *Бюлетень з авторського права та суміжних прав*, № 89, с. 139.

https://nipo.gov.ua/wp-content/uploads/2025/05/BULETEN_Avt_Pravo_-89-2025.zip

19. **Oriekhov, O., & Farionova, T.** (2025). Information technology for estimating the size of software projects. In I. Linde (Ed.), *Innovative technologies for project and program management* (pp. 192-201). Ryha (Latvia): European University Press. <https://mmp-conf.org/documents/archive/monography2025.pdf> (ISBN 978-9984-891-34-7, розділ в колективній монографії)

ABSTRACT

Oriekhov O. S. Mathematical models for information processing of JAVA application code metrics for their size estimation. – Manuscript of the qualification scientific work.

Thesis for the degree of philosophy doctor (PhD) in specialty 122 "Computer Science" (field of knowledge 12 – Information Technology). – Admiral Makarov National University of Shipbuilding, Ministry of Education and Science of Ukraine, Mykolaiv, 2026.

The dissertation is devoted to solving the scientific and practical problem of improving the reliability of processing information from JAVA application code metrics for their software size estimation by constructing appropriate mathematical models, namely nonlinear regression models, and creating information technology (IT) tools for processing information from code metrics necessary for the software size parameter estimation in the early stages of software application development based on UML class diagram metrics using a general sample of JAVA applications and a sample of Data Science (DS) and Machine Learning (ML) JAVA applications.

The relevance of the dissertation lies in the fact that reliable information processing from JAVA application code metrics, namely their size, in the early stages of software design directly affects the prediction of the software development efforts estimation. This allows for effective planning of resources, budget, and time for the creation of software projects. However, software size estimation in the early stages of design is complicated by the fact that in the initial stages of development, such information is limited and difficult to measure, resulting in a series of unreliable estimates. The reliability of software size estimation is improved by using appropriate mathematical models that take into account environmental factors, such as the JAVA programming language and project characteristics, in particular the use of DS and ML technologies with this programming language. In this regard, there is a necessity to improve the reliability of software size estimation in the early stages of

development by using software code metrics that are available in the conceptual model of the project (UML-class diagram).

The dissertation aim is to improve the reliability of information processing from code metrics for estimating the size of JAVA applications in the early stages of software development by constructing nonlinear regression models based on sets of code metrics from open source projects.

The working scientific hypothesis of the dissertation research is the assertion that improving the reliability of information processing from code metrics for software size estimation (number of lines of code) of JAVA applications in the early stages of software development is achieved through the use of non-Gaussian nonlinear regression models, which allow the specified size to be described as a random variable and take into account the difference between the data distribution and the normal distribution.

To construct nonlinear regression models, it is proposed to use an approach based on the usage of multivariate normalizing transformations, which allow to take into account correlations between the factors and approximate the distribution to normal, which increases the probability of reliable processing of information from JAVA application code metrics for the size parameter estimation.

To achieve the aim, it is necessary to solve following tasks:

- to conduct an analysis of existing mathematical models for information processing of code metrics for JAVA applications size estimation of general JAVA applications, DS and ML JAVA applications using appropriate samples of open-source projects;
- to form training and testing samples from JAVA applications code metrics, and justify the selection of independent factors for building nonlinear regression models for processing information from code metrics to estimate their size;
- to examine the obtained metrics for multicollinearity using the VIFs criterion, taking into account iterative processing of multidimensional data;

- to select transformations for normalising six-dimensional vectors of JAVA application code metric characteristics based on the analysis of existing normalising transformations and methods for evaluating their parameters;
- to improve the four-factor nonlinear regression model for code metrics information processing for JAVA applications size estimation using the five-dimensional Box-Cox normalizing transformation based on a general sample of code metrics from open-source JAVA application that can be obtainable from the conceptual model of the project;
- to build a five-factor nonlinear regression model for processing code metric information to estimate the size of JAVA applications using a six-dimensional Box-Cox normalising transformation based on a general sample of code metrics from open-source JAVA applications, which can be obtained from the conceptual model of the project;
- to build a five-factor nonlinear regression model for processing code metrics information to estimate the size of JAVA applications using the six-dimensional Box-Cox normalizing transformation based on a sample of DS and ML from open-source JAVA applications, which can be obtained from the conceptual model of the project;
- to develop information technology (SaaS software solution) and methodology for processing information from code metrics obtained from UML class diagrams to estimate the JAVA applications size parameter by using the constructed five-factor nonlinear regression models.

The scientific novelty of the obtained results is as follows.

- 1) A four-factor nonlinear regression model has been improved for code metrics information processing to estimate the size parameter of JAVA applications depending on the total number of classes and interfaces (CLASS), the total number of unique method calls in classes (RFC), the average number of links between classes (aCBO), and the average number of visible (public and protected) methods per class (aVMQ) based on the five-dimensional Box-Cox transformation, which made it possible to increase the reliability of estimating the size parameter of such

applications as a non-Gaussian random variable compared to existing regression models. Compared to other models, the constructed model has a higher percentage of predicted values, a lower mean relative error, and smaller prediction interval widths of the nonlinear regression.

2) For the first time, a five-factor nonlinear regression model was constructed for code metrics information processing to estimate the size parameter of JAVA applications based on the six-variate Box-Cox transformation in which, comparing to existing models, the metric of the total number of classes and interfaces (CLASS) was split into two separate code metrics: the number of classes (CLS) and the number of interfaces (INFC) for the first time. This made it possible to increase the reliability of the code metrics information processing for the software size estimation compared to existing mathematical models. The constructed model has better quality indicators R^2 , $MMRE$ and $PRED(0, 25)$ and smaller prediction interval widths and confidence intervals of the nonlinear regression compared to other regression models.

3) For the first time, a five-factor nonlinear regression model was constructed for code metrics information processing to estimate the size parameter of DS and ML JAVA applications based on the six-variate Box-Cox transformation, in which, comparing to existing models, the metric of the total number of classes and interfaces (CLASS) was split into two separate code metrics: the number of classes (CLS) and the number of interfaces (INFC) and took into account the specifics of developing this category of applications for the first time. This made it possible to increase the reliability of processing information from code metrics for the software size estimation compared to using one-dimensional and five-dimensional Box-Cox transformations. The constructed model has a higher percentage of predicted values, a lower mean relative error, and smaller prediction and confidence interval widths for nonlinear regression in comparison to other regression models.

4) Equations for determining the prediction interval limits of five-factor nonlinear regressions for general JAVA applications and Data Science and Machine Learning JAVA applications based on six-dimensional Box-Cox normalising

transformations further developed. This allows us to reduce the width of the prediction intervals of nonlinear regressions for the size parameter estimation of the corresponding software applications compared to the usage of five-factor Box-Cox transformations and thus increase the reliability of specified size estimation as a dependent random variable.

The practical significance of the obtained results is as follows. The IT tool development for code metrics information processing to estimate the size parameter of JAVA applications based on metrics that can be obtained from conceptual data models at the early stages of software project development, calculating confidence intervals and prediction intervals using appropriate five-factor nonlinear regression models based on a general sample of JAVA applications and a sample of DS and ML JAVA applications. For this purpose, software was developed in the Kotlin programming language that uses the constructed nonlinear regression mathematical models. The software is distributed under the software-as-a-service (SaaS) model and offers a software interface based on the HTTP protocol.

The introduction of the dissertation reveals the essence and significance of the scientific and practical task of code metrics information processing for JAVA applications size estimation, and justifies the necessity of the research on the chosen topic. The general characteristics of the dissertation are presented in the following sequence: justification and relevance of the chosen topic; the connection of the work with scientific programs, plans, and topics; the purpose and objectives of the research; the scientific novelty and practical significance of the obtained results; the personal contribution of the applicant; the approbation of the dissertation results and publications.

In Section 1, the analysis of existing methods and models for code metric information processing to estimate the size parameter of JAVA application code lines and justifies the necessary for research on the chosen topic.

In Section 2, existing reciprocal singlevariate and multivariate normalizing transformations used to construct nonlinear regression models are reviewed and

analysed, the choice of transformation for normalising six-variate vectors of metrics of general JAVA applications, DS and ML JAVA applications was justified.

In Section 3, four- and five-factor nonlinear regression models, their confidence intervals and prediction intervals are constructed for processing information from code metrics to estimate the size of JAVA applications using multivariate Box-Cox transformations based on a general sample of JAVA applications and a sample of DS and ML JAVA applications. A comparison was made of the accuracy of processing information from code metrics for evaluating the size (number of lines of code) of JAVA applications using the constructed nonlinear regression models.

In Section 4, an IT tool is proposed for processing code metrics information to estimate the size of JAVA applications at the initial stages of software application design using class diagram metrics. For this purpose, a SaaS solution was developed in the Kotlin programming language and using the Spring Boot framework, which offers an API for code metrics information processing to estimate KLOC of JAVA applications using the obtained nonlinear regression models.

Keywords: software development effort, source code, LOC-estimation, Java, machine learning, mathematical model, regression model, nonlinear regression, non-Gaussian data, normalizing transformation, Box-Cox transformation, confidence interval, prediction interval, information technology, software.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	21
ВСТУП.....	22
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І МОДЕЛЕЙ ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК JAVA-ЗАСТОСУНКІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ДОСЛІДЖЕННЯ ЗА ОБРАНОЮ ТЕМОЮ.....	34
1.1 Аналіз існуючих методів та моделей для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків.....	34
1.2 Формування набору даних для обробки інформації з метрик коду загальної вибірki JAVA-застосунків.....	40
1.3 Формування набору даних для обробки інформації з метрик Data Science та Machine Learning JAVA-застосунків.....	49
1.4 Обґрунтування необхідності проведення досліджень за обраною темою....	54
1.5 Висновки за розділом 1 і постановка завдань дослідження.....	58
РОЗДІЛ 2. ВИБІР НОРМАЛІЗУЮЧОГО ПЕРЕТВОРЕННЯ ДЛЯ ПОБУДОВИ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК КОДУ JAVA-ЗАСТОСУНКІВ.....	61
2.1 Методика вибору нормалізуючого перетворення.....	61
2.2 Одновимірні взаємозворотні нормалізуючі перетворення.....	63
2.3 Багатовимірні взаємозворотні нормалізуючі перетворення.....	69
2.4 Вибір нормалізуючого перетворення для обробки інформації з багатовимірних даних метрик коду загальних JAVA-застосунків.....	72
2.5 Вибір нормалізуючого перетворення для обробки інформації з багатовимірних даних метрик коду Data Science та Machine Learning JAVA-застосунків.....	78
2.6 Висновки за розділом 2.....	84

РОЗДІЛ 3. ПОБУДОВА ЧОТИРЬОХ- ТА П'ЯТИФАКТОРНИХ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК КОДУ JAVA-ЗАСТОСУНКІВ.....	86
3.1 Загальний алгоритм побудови нелінійних регресійних моделей на основі взаємозворотних нормалізуючих перетворень.....	86
3.2 Побудова чотирьохфакторної нелінійної регресійної моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та її інтервалів передбачення на основі п'ятимірного нормалізуючого перетворення Бокса-Кокса.....	90
3.3 Побудова п'ятифакторної нелінійної регресійної моделі та інтервалів її передбачення обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків на основі шестивимірного нормалізуючого перетворення Бокса-Кокса.....	94
3.4 Побудова п'ятифакторної нелінійної регресійної моделі та інтервалів її передбачення обробки інформації з метрик коду для оцінювання розміру Data Science та Machine Learning JAVA-застосунків на основі шестивимірного нормалізуючого перетворення Бокса-Кокса.....	100
3.5 Порівняння результатів достовірності обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та Data Science та Machine Learning JAVA-застосунків з існуючими нелінійними регресійними моделями.....	109
3.6 Порівняння інтервалів передбачення за побудованими нелінійними регресійними моделями обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та Data Science та Machine Learning JAVA-застосунків.....	119
3.7 Висновки за розділом 3.....	122
РОЗДІЛ 4. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК JAVA-ЗАСТОСУНКІВ.....	125

4.1 Створення інформаційної технології для обробки інформації з метрик JAVA-застосунків для оцінювання параметра розміру на основі вибірки проєктів з відкритим кодом.....	125
4.2 Методика обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків за допомогою п'ятифакторних нелінійних регресійних моделей.....	141
4.3 Висновки за розділом 4.....	148
ВИСНОВКИ.....	151
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	155
Додаток А – Аналіз успішності розробки проєктів програмного забезпечення.....	168
Додаток Б – Навчальний та тестовий набори даних загальних JAVA-застосунків з відкритим програмним кодом.....	173
Додаток В – Навчальний та тестовий набори даних Data Science та Machine Learning JAVA-застосунків з відкритим програмним кодом	187
Додаток Г – Акти впровадження результатів дисертаційної роботи.....	190
Додаток Д – Акт впровадження результатів дисертаційної роботи в навчальний процес.....	192
Додаток Е – Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації.....	193
Додаток Є – Інструкція користувача інформаційної технології «Сервіс для обробки інформації з метрик програмних застосунків».....	197
Додаток Ж – Основні модулі вихідного коду інформаційної технології «Сервіс для обробки інформації з метрик програмних застосунків».....	211

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВВ – випадкова величина;

ІТ – інформаційна технологія;

ММП – метод максимальної правдоподібності;

ПЗ – програмне забезпечення;

CBO (Coupling between objects) – зв'язки між об'єктами;

CLS (Classes) – фактична кількість класів;

DS (Data Science) – наука про дані;

INFC (Interfaces) – фактична кількість інтерфейсів;

KLOC (Kilo Lines Of Code) – кількість тисяч рядків коду;

LOC (Lines Of Code) – кількість рядків коду;

ML (Machine Learning) – машинне навчання;

MMRE (Mean Magnitude of Relative Error) – середня величина відносної помилки;

MRE (Magnitude of Relative Error) – середня величина відносної помилки;

PRED (Percentage of Prediction) – відсоток прогнозованих результатів;

RFC (Response for a Class) – загальна кількість унікальний викликів методів у класі;

SaaS (Software as a Service) – програма як послуга;

TFQ (Total Fields Quantity) – загальна кількість полів;

VMQ (Visible Methods Quantity) – загальна кількість public та protected методів;

UML (Unified Modeling Language) – уніфікована мова моделювання.

ВСТУП

Обґрунтування вибору теми дослідження. Робота присвячена вирішенню науково-практичного завдання підвищення достовірності обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру шляхом побудови відповідних математичних моделей, а саме нелінійних регресійних моделей, та створенню інструментарію інформаційної технології (ІТ) для обробки інформації з метрик коду, необхідних для оцінювання параметра розміру на ранніх стадіях розробки програмних застосунків за метриками UML-діаграми класів на основі загальної вибірки JAVA-застосунків та вибірки DS і ML JAVA-застосунків із подальшим застосуванням у параметричних моделях розрахунку трудомісткості розробки програмного забезпечення (ПЗ) [1, 2].

Оцінка трудомісткості розробки ПЗ є одним з ключових показників для бізнесу під час формуванні бюджету та плануванні часу розробки. Ця задача є важливою для ІТ-компаній, оскільки дозволяє спрогнозувати кількісні показники ресурсів, які необхідно виділити на створення програмного проєкту, що, своєю чергою, сприяє підвищенню ефективності процесу розробки. Вибір ефективних методів та якісних моделей оцінювання трудомісткості створення ПЗ залежно від методології розробки дозволить підвищити достовірність обробки інформації щодо планування як окремих етапів розробки, так і всього проєкту в цілому [3-5].

Успішність проєкту залежить від достовірного прогнозування трудомісткості розробки ПЗ. Детальний аналіз сучасного стану успішності розробки ПЗ та вплив якісного прогнозування трудомісткості, вартості та часу наведено в додатку А дисертаційної роботи.

Незважаючи на стрімкий розвиток галузі інформаційних технологій, дослідження успішності завершення проєктів ПЗ демонструють наявність проблем, пов'язаних з достовірністю прогнозування трудомісткості розробки, яка залежить від якості прогнозування розміру програмних застосунків. Одним із ефективних методів оцінювання розміру програмних застосунків є побудова

ймовірнісних математичних моделей. Аналіз сучасного стану ІТ-галузі та останніх публікацій показав, що для отримання достовірних оцінок трудомісткості та вартості розробки ПЗ також необхідно враховувати сучасні технологічні аспекти як розробки, так і управління програмними проєктами, а також фактори середовища [6-11].

Мова програмування JAVA є однією з найпоширеніших на сучасному ринку розробки ПЗ, яка використовується в багатьох сферах, таких як створення веб-застосунків, мобільних застосунків, ігор, систем штучного інтелекту, а також у наукових обчисленнях. Мова JAVA має багато переваг, серед яких можна виокремити такі: кросплатформеність, об'єктно-орієнтованість, безпека, велика кількість бібліотек та фреймворків [12]. Згідно з результатами досліджень та індексами популярності мов програмування, такими як IEEE, TIOBE, JAVA постійно утримується на передових позиціях, конкуруючи з такими відомими мовами, як C, C++, Python, JavaScript та іншими [13-14]. JAVA була розроблена саме як об'єктно-орієнтована мова програмування, яка не просто підтримує, а й фундаментально базується на парадигмах ООП. На відміну від гібридних мов, які з'явилися раніше, JAVA з самого початку проєктувалася з вимогою, щоб розробники мислили категоріями класів та об'єктів. Вона надає прямі синтаксичні конструкції для реалізації всіх ключових принципів об'єктно-орієнтованого підходу: інкапсуляції, успадкування і поліморфізму [15]. Саме така будова цієї мови програмування забезпечує тісну сумісність між візуальними конструкціями UML (наприклад, діаграмами класів), що дає змогу здійснити легкий перехід від архітектурного проєктування до програмної реалізації [16-19].

Створення сучасних програмних продуктів неможливо уявити без застосування технологій DS та ML, оскільки ринок розробки потребує впровадження цих технологій для отримання конкурентної переваги і пропозиції кращих програмних рішень та сервісів для користувачів. Капіталізація та активне зростання ринку Machine Learning свідчать про перспективність розвитку цього напрямку індустрії ІТ. Розвиток великих мовних

моделей у 2022 та 2023 роках спричинив справжню революцію в IT-індустрії [20]. Це зумовило активну інтеграцію технологій DS та ML в інфраструктуру різних галузей, таких як охорона здоров'я, освіта, державний сектор, фінанси та економіка, електронна торгівля, аерокосмічна галузь, біоінформатика тощо [21]. Тому оцінювання трудомісткості розробки DS та ML JAVA-застосунків є важливою складовою управління програмними проектами.

Підсумовуючи, можна зробити висновок, що обробка інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків на ранніх етапах розробки з метою подальшого використання в параметричних моделях оцінювання трудомісткості є важливим науково-практичним завданням.

Коректне оцінювання параметра розміру сприяє підвищенню достовірності оцінювання трудомісткості та вартості розробки JAVA-застосунків, що дозволяє більш ефективно вирішувати такі завдання:

- прогнозування реальних витрат ресурсів та бюджету на розробку програмних проєктів на ранніх етапах розробки з урахуванням особливостей мови програмування JAVA, що забезпечує раціональне планування та ефективне використання ресурсів;
- оптимізацію розподілу та використання людських ресурсів, з урахуванням прогнозованої трудомісткості проєктів, що сприяє підвищенню продуктивності команди та рівномірному розподілу роботи між її членами залежно від компетенції та завантаженості;
- встановлення зв'язку між розміром програмних застосунків та трудомісткістю їх розробки, що дозволяє оцінити вплив архітектурних рішень на ефективність процесу розробки;
- забезпечення контролю над виконанням проєкту та моніторингу ходу виконання робіт, що дає змогу вчасно коригувати процес розробки у разі відхилень від планових показників;
- оцінювання параметра розміру ПЗ за окремими напрямками та категоріями з метою отримання достовірної інформації для подальшої оцінки трудомісткості з урахуванням фактора середовища розробки.

Для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків з відкритим кодом побудовано як лінійні [22, 23], так і нелінійні [24-32] регресійні рівняння та моделі залежно від різної варіації метрик коду та мов програмування. Так, у [22, 23] запропоновано трьохфакторні лінійні регресійні рівняння для оцінювання розміру JAVA-застосунків з відкритим кодом та інформаційних систем. Але, як відомо, під час побудови лінійних регресійних моделей необхідним є виконання певних умов, зокрема гаусівського розподілу багатовимірних даних інформації з метрик коду JAVA-застосунків, нормального розподілу похибок регресії, гомоскедастичності даних, що можливо лише в поодиноких випадках. У роботах [24-32] запропоновано нелінійні регресійні моделі для обробки інформації з різною варіативністю метрик коду як факторів регресії та наборів даних, які були використані для побудови таких моделей. Робота [25] присвячена побудові однофакторній нелінійної регресійної моделі для оцінювання JAVA веб-застосунків. В роботі [24] розроблено трьохфакторну нелінійну регресійну модель для оцінювання розміру інформаційних систем на базі набору даних з робіт [22, 23]. У дослідженні [26] представлено чотирьохфакторну нелінійну регресійну модель, яка побудована на основі п'ятивимірного перетворення Джонсона з сім'ї SB. Наукові розробки [26, 27] доводять, що зі збільшенням кількості факторів нелінійних регресійних моделей зменшується ширина інтервалів передбачення. Це дає змогу досягти кращої точності оцінювання параметра розміру, коли необхідно кількісно оцінити ризики і підготувати відповідні стратегії для різних сценаріїв. Крім того, розмір навчальних вибірок факторів моделей безпосередньо впливає на достовірність оцінювання, оскільки початкові дані можуть мати недостатній рівень репрезентативності генеральної сукупності [33, 34].

Отже розробка та вдосконалення нелінійних регресійних моделей обробки інформації з метрик коду для підвищення точності та достовірності оцінювання параметра розміру (у рядках коду) як загальних, так і DS та ML JAVA-застосунків є актуальним завданням, вирішення якого досягається

шляхом побудови відповідних нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень із використанням методів математичної статистики, нелінійного регресійного аналізу та методів перехресної перевірки.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційну роботу виконано у Національному університеті кораблебудування імені адмірала Макарова (НУК) відповідно до планів НДР з ініціативної теми «Математичні моделі обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків на ранніх етапах розробки програмних проєктів» (державний реєстраційний номер: 0125U002663). У зазначеній НДР здобувач був відповідальним виконавцем.

Мета і завдання дослідження. Метою дослідження є підвищення достовірності обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків на ранніх стадіях проєктування програмного забезпечення за рахунок побудови нелінійних регресійних моделей на основі наборів метрик коду проєктів з відкритим кодом.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих математичних моделей для обробки інформації з метрик коду для оцінювання розміру загальних та DS і ML JAVA-застосунків, із використанням відповідних вибірок проєктів з відкритим кодом;
- сформувати навчальні та тестові вибірки з метрик коду JAVA-застосунків та обґрунтувати вибір незалежних факторів для побудови нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання їх розміру;
- дослідити отримані метрики на відсутність мультиколінеарності за критерієм VIFs, з урахуванням ітеративної обробки багатовимірних даних;
- обрати перетворення для нормалізації шестивимірних векторів характеристик метрик коду JAVA-застосунків на основі аналізу існуючих нормалізуючих перетворень та методів оцінювання їх параметрів;

- удосконалити чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із використанням п'ятивимірного нормалізуючого перетворення Бокса-Кокса на основі загальної вибірки метрик коду JAVA-застосунків з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- побудувати п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із застосуванням шестивимірного нормалізуючого перетворення Бокса-Кокса на основі загальної вибірки метрик коду JAVA-застосунків з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- побудувати п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків із застосуванням шестивимірного нормалізуючого перетворення Бокса-Кокса на основі вибірки метрик коду DS та ML JAVA-проєктів з відкритим кодом, які можна отримати з концептуальної моделі проєкту;
- розробити інформаційну технологію (SaaS програмне рішення) та методику для обробки інформації з метрик коду, отриманих з UML-діаграми класів, для оцінювання параметра розміру JAVA-застосунків із використанням побудованих п'ятифакторних нелінійних регресійних моделей.

Об'єктом дослідження є процес обробки інформації з метрик коду JAVA-застосунків для оцінювання параметра їх розміру.

Предметом дослідження є нелінійні регресійні моделі для обробки інформації з метрик коду JAVA-застосунків для оцінювання параметра їх розміру.

Методи дослідження. Для досягнення поставленої мети та завдань дослідження використано такі методи: пошуковий, системний аналіз, синтез, порівняння, абдукція, історичний і логічний методи, методи теорії ймовірності, математичної статистики, багатовимірного статистичного аналізу, лінійного та нелінійного регресійного аналізу. Методи пошуку, порівняння, ідеалізації, історичний та логічний методи використано в процесі аналізу існуючих

регресійних моделей та рівнянь для обробки інформації з метрик коду JAVA-застосунків для оцінювання параметра розміру. Методи теорії ймовірності, математичної статистики та багатовимірного статистичного аналізу застосовано для оцінювання характеристик випадкових величин, перевірки статистичних гіпотез про нормальність розподілу одновимірних та багатовимірних емпіричних даних, оцінювання параметрів нормалізуючих перетворень. Методи лінійного та нелінійного регресійного аналізу застосовано для побудови нелінійних регресійних моделей та рівнянь, визначення границь довірчого інтервалу та інтервалу передбачення регресійних моделей оцінювання параметра розміру JAVA-застосунків.

Наукова новизна отриманих результатів полягає у наступному.

1) Удосконалено чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків залежно від загальної кількості класів та інтерфейсів (CLASS), загальної кількості унікальних викликів методів у класах (RFC), середнього значення кількості зв'язків між класами (aCBO) та середнього значення кількості видимих (public та protected) методів на клас (aVMQ) на основі п'ятивимірного перетворення Бокса-Кокса, що дозволило підвищити достовірність оцінювання параметра розміру таких застосунків як негаусівської випадкової величини у порівнянні з існуючими регресійними моделями. Модель, що побудована, у порівнянні з іншими моделями має більший відсоток прогнозованих значень, менше значення середньої величини відносної похибки та менші ширини інтервалу передбачення нелінійної регресії.

2) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків на основі шестивимірного перетворення Бокса-Кокса, в якій, на відміну від існуючих моделей, було розділено метрику загальної кількості класів та інтерфейсів (CLASS) на дві окремі метрики, а саме кількість класів (CLS) та кількість інтерфейсів (INFC), що дозволило підвищити достовірність обробки інформації з метрик коду для оцінювання параметра розміру в

порівнянні з існуючими математичними моделями. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має кращі показники якості R^2 , $MMRE$ та $PRED(0, 25)$ та менші ширини інтервалу передбачення та довірчого інтервалу нелінійної регресії.

3) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків на основі шестивимірного перетворення Бокса-Кокса, в якій, на відміну від існуючих моделей, було розділено метрику загальної кількості класів та інтерфейсів (CLASS) на дві окремі метрики коду кількості класів (CLS) та інтерфейсів (INFC) та враховано особливості розробки цієї категорії застосунків, що дозволило підвищити достовірність обробки інформації з метрик коду для оцінювання параметра розміру у порівнянні із використанням одновимірного та п'ятивимірного перетворень Бокса-Кокса. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має більший відсоток прогнозованих значень, менше значення середньої величини відносної похибки та менші ширини інтервалу передбачення та довірчого інтервалу нелінійної регресії.

4) Отримали подальший розвиток рівняння для визначення границь інтервалів передбачення п'ятифакторної нелінійної регресії для загальних JAVA-застосунків та DS і ML JAVA-застосунків на основі шестивимірних нормалізуючих перетворень Бокса-Кокса. Це у порівнянні з використанням п'ятивимірних перетворень Бокса-Кокса дозволяє зменшити ширини інтервалів передбачення нелінійних регресій для оцінювання параметра розміру відповідних програмних застосунків і тим самим підвищити достовірність оцінювання зазначеного розміру як залежної випадкової величини.

Практичне значення дисертаційної роботи полягає у розробці інструментарію ІТ для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, які можна отримати з концептуальної моделі даних на ранніх етапах розробки програмного проєкту, розрахунку довірчих інтервалів та інтервалів передбачення із застосуванням відповідних

п'ятифакторних нелінійних регресійних моделей на основі загальної вибірки JAVA-застосунків, вибірки DS та ML JAVA-застосунків. Для цього мовою програмування Kotlin для отриманих нелінійних регресійних математичних моделей створено ПЗ, що розповсюджується за моделлю software-as-a-service (SaaS) та має відповідний програмний інтерфейс на основі протоколу HTTP.

Результати дисертаційної роботи впроваджено в таких установах та організаціях: ТОВ «Н-ІКС ДЕЛІВЕРІ» та ФОП Поліщук Андрій Борисович. Застосування інструментарію інформаційної технології для оцінювання параметра розміру рядків коду JAVA-застосунків на основі вибірок з відкритим кодом загальних JAVA-застосунків, DS та ML JAVA-застосунків дозволило підвищити достовірність обробки інформації для оцінювання параметра розміру на ранніх стадіях розробки програмних проєктів за відповідними напрямками.

Результати дисертаційної роботи також застосовуються у навчальному процесі при викладанні студентам Навчально-наукового інституту комп'ютерних наук та управління проєктами (ННІКНУП) Національного університету кораблебудування імені адмірала Макарова таких дисциплін: «Керування проєктними ресурсами в інженерії програмного забезпечення» в ОП «Інженерія програмного забезпечення» для 2-го магістерського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»; «Основи програмної інженерії» в ОП «Інженерія програмного забезпечення» для 1-го бакалаврського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»; «Моделювання систем цифрового екологічного моніторингу» в ОП «Цифровий екологічний моніторинг» для 2-го магістерського рівня ВО, спеціальності E2 «Екологія», галузі знань E «Природничі науки, математика та статистика».

Акти про впровадження результатів дисертаційної роботи наведені у додатках Г та Д.

Особистий внесок здобувача. Дисертаційна робота є самостійно виконаною науковою працею. Усі наукові результати отримано здобувачем особисто. У працях, опублікованих у співавторстві, здобувачеві належать: збір, обробка та аналіз інформації з метрик програмного коду JAVA-застосунків на основі загальної вибірки та вибірки DS та ML JAVA-застосунків у всіх опублікованих матеріалах; побудова однофакторних [27], двофакторних [27], трьохфакторних [28-32], чотирьохфакторних [35-38] та п'ятифакторних [36-38] нелінійних регресійних моделей для обробки інформації з метрик коду JAVA-застосунків залежно від метрик з UML-діаграми класів, таких як загальної кількості класів, загальної кількості методів на клас, середньої кількості методів відносно суми класів та інтерфейсів, загальної кількості полів класів, середньої кількості полів відносно суми класів та інтерфейсів, загальної кількості зв'язків між об'єктами та середньої кількості зв'язків між об'єктами відносно суми класів та інтерфейсів, суми глибини дерева успадкування та середнього значення глибини дерева успадкування відносно суми класів та інтерфейсів, загальної кількості нащадків класів та середньої кількості нащадків відносно суми класів та інтерфейсів, загальної кількості унікальних методів класів та середньої кількості унікальних викликів методів відносно суми класів та інтерфейсів; створення інформаційної технології для оцінювання розміру JAVA-застосунків у кількості рядків коду із застосуванням побудованих багатфакторних нелінійних регресійних моделей [39, 40].

Апробація результатів досліджень. Основні положення і результати дисертаційної роботи доповідались на 10 науково-технічних конференціях, зокрема:

- XIV науково-технічній конференції “Інновації в суднобудуванні та океанотехніці” (Миколаїв: НУК імені адмірала Макарова, 20-21 вересня 2023).
- Міжнародній науково-практичній конференції "Інтелектуальні інформаційні системи в управлінні проєктами та програмами" (Харків: ХНУРЕ, 12-15 вересня 2023 р.).

- IV Всеукраїнській науково-практичній інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2023) (Миколаїв, 30-31 жовтня 2023 р.).
- V International Workshop on IT Project Management (ITPM-2024) (Bratislava, May 22 2024)
- XV науково-технічній. конференції "Інновації в суднобудуванні та океанотехніці" (Миколаїв, 26-27 вересня 2024 р.).
- XI International Scientific and Practical Conference "Information Control Systems and Technologies 2024" (ICST-2024) (Odesa, September 23-25 2024).
- V Всеукраїнській науково-практичній інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2024) (Миколаїв, 30-31 жовтня 2024 р.).
- VI International Workshop on IT Project Management (ITPM-2025) (Kyiv, May 22 2025)
- XVI Науково-технічній конференції "Інновації в суднобудуванні та океанотехніці" (Миколаїв, 25-26 вересня 2025).
- VI Всеукраїнській науково-практичній інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2025) (Миколаїв, 30-31 жовтня 2025 р.).

Публікації. Результати дисертаційного дослідження викладено у 19 публікаціях. Основні результати дисертації опубліковано у 4 статтях в наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України; 1 одноосібному розділі у колективній монографії. Ще 12 публікацій засвідчують апробацію матеріалів дисертації, з них 3 статті включено до міжнародної наукометричної бази Scopus. Дві публікації додатково відображають наукові результати дисертаційної роботи.

Структура і обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел з 107

найменувань і 8 додатків. Загальний обсяг дисертації становить 223 сторінок, обсяг основного тексту налічує 133 сторінок. Дисертаційна робота містить 5 рисунків, 44 таблиць. У додатках наведено набори емпіричних даних, документи про впровадження результатів роботи, інструкцію користувача, вихідний код програми, список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації.

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ І МОДЕЛЕЙ ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК JAVA-ЗАСТОСУНКІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ДОСЛІДЖЕННЯ ЗА ОБРАНОЮ ТЕМОЮ

У першому розділі наведено огляд, аналіз та порівняння існуючих математичних моделей обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, обґрунтування необхідності підвищення достовірності обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків на основі вибірок загальних JAVA-застосунків, DS та ML JAVA-застосунків шляхом побудови нелінійних регресійних моделей.

1.1 Аналіз існуючих методів та моделей для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків

Оцінка трудомісткості та вартості розробки ПЗ є важливою задачею для формування бюджету та планування ресурсів для розробки на ранніх етапах проєктування. Дослідження успішності проєктів ПЗ [3-5, 41, 42] підтверджують, що незважаючи на стрімкий розвиток ІТ-технологій і методологій розробки, прогнозування трудомісткості все ще має високий рівень невизначеності. Виявлено, що зі збільшенням розміру ПЗ зменшується частка успішно реалізованих проєктів. Отже коректне планування бюджету і ресурсів є надзвичайно важливим для підвищення ймовірності успішної реалізації будь-якого проєкту ПЗ, оскільки дає уявлення про те, чи можливо завершити розробку проєкту в межах очікуваного часу, бюджету та наявних ресурсів.

Сучасні методи оцінювання трудомісткості розробки ПЗ можна умовно поділити на три категорії: алгоритмічні, неалгоритмічні та методи машинного навчання. Оцінювання неалгоритмічними методами ґрунтується на думці експертів, які мають досвід подібних попередніх проєктів, тоді як алгоритмічні підходи використовують параметричну оцінку та побудовані на основі математичного апарату. Методи на основі штучного інтелекту (ШІ) та моделей

машинного навчання використовують вже існуючу інформацію щодо проєктів та ґрунтуються на певних наборах даних [6-8, 43].

Алгоритмічні (або параметричні) методи оцінювання базуються на застосуванні математичних моделей та алгоритмів для визначення параметрів програмного проєкту, таких як трудомісткість, час та вартість розробки. Ці методи можуть бути використані як на етапі планування проєкту, так і на етапі розробки ПЗ. Для оцінювання трудомісткості розробки ПЗ найбільш поширеними є моделі COCOMO та COCOMO II, SLIM, ISBSG, COSYSMO, оцінювання трудомісткості на основі Function Point Analysis (FPA) тощо [6-8, 44]. Залежно від моделі, для оцінки трудомісткості або вартості розробки використовується параметр розміру вихідного коду програмного проєкту, який виражається значенням кількості рядків коду (KLOC) або значенням функціональних точок (FP). Оцінки трудомісткості за кількістю рядків коду та функціональними точками мають свої переваги та недоліки [6-8, 44]. Функціональні точки дають змогу проводити оцінювання трудомісткості ПЗ незалежно від технологій. Цей підхід зосереджений на вимогах користувача, враховує обсяг вхідних/вихідних даних, складність обробки та взаємодію з іншими системами. Це робить FP більш незалежним від технології реалізації та сприяє ранньому оцінюванню трудомісткості на етапі специфікацій, але таким чином недостатньо враховуються технічні фактори, такі як мова програмування, фреймворк чи тип програмного проєкту, що призводить до втрат у точності порівняно з оцінюванням за параметром кількості рядків коду. Оцінка за кількістю рядків коду має прямий зв'язок з реалізацією та базується на концептуальній моделі даних у вигляді UML-діаграми класів, що забезпечує високу точність оцінювання для проєктів зі стабільними вимогами та деталізованим об'єктно-орієнтованим проєктуванням. Отже показник кількості рядків коду має перевагу в достовірності оцінювання параметра розміру над використанням функціональних точок, оскільки враховує технічні фактори створення майбутнього ПЗ.

Згідно з існуючим стандартом [45], обробка інформації представляє собою виконання однієї або кількох операцій, зокрема збирання, введення, записування, перетворення, зберігання тощо, які здійснюються в системі за допомогою технічних і програмних засобів. Обробка інформації в комп'ютерних науках – це послідовність операцій над вхідними даними, що трансформує їх у вихідні дані для подальшого ухвалення рішення [46]. Метою обробки інформації є перетворення даних на щось зрозуміле та корисне для людей або для інших систем. Достовірна обробка інформації з метрик коду JAVA-застосунків дає змогу отримати інструментарій оцінювання метрики розміру програмного проєкту для подальшого прогнозування трудомісткості, вартості та часу.

Процес побудови математичних моделей для оцінювання параметра розміру JAVA-застосунків складається з таких етапів обробки інформації з метрик коду JAVA-застосунків, а саме:

- збір інформації з метрик коду проєктів JAVA-застосунків з відкритим кодом;
- попередню обробку та структурування даних, що передбачає обчислення значень середніх метрик, перетворення метрик для подальшої нормалізації, поділ отриманого набору даних на навчальну та тестову вибірки;
- вибір незалежних факторів регресії та їхню перевірку на наявність мультиколінеарності;
- нормалізацію даних та ітеративне очищення навчальної вибірки від викидів;
- оцінювання параметрів лінійної регресійної моделі та застосування взаємозворотних нормалізуючих перетворень;
- перевірку результатів оцінювання параметра розміру за критеріями якості нелінійних регресійних моделей;
- застосування отриманої моделі в інформаційній технології для оцінювання параметра розміру JAVA-застосунків за вхідними параметрами метрик коду.

Отже етапи обробки інформації з метрик коду JAVA-застосунків є необхідною складовою процесу побудови відповідних математичних моделей для оцінювання параметра розміру ПЗ. Такі математичні моделі є інструментом перетворення інформації від значень початкових метрик коду з діаграми класів до значення параметра розміру, яке необхідне для використання в моделях оцінювання трудомісткості та вартості з метою ухвалення подальших управлінських рішень щодо розробки програмного проєкту.

Вирішенню проблеми підвищення достовірності обробки інформації з метрик коду ПЗ присвячені наукові праці як вітчизняних [24-26, 47-55], так і іноземних [22, 23] дослідників, зокрема окремому напрямку оцінювання розміру ПЗ на ранніх стадіях проєктування. У представлених роботах автори пропонують методи і моделі обробки інформації для оцінювання параметра розміру (кількості рядків коду) на ранніх стадіях проєктування програмних застосунків залежно від різних комбінацій наборів метрик з UML-діаграми класів, мови програмування та категорій ПЗ. Підвищення якості обробки інформації для оцінювання розміру ПЗ досягається завдяки використанню відповідних математичних моделей, зокрема нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень [25, 27, 47-54]. Застосування багатовимірних нормалізуючих перетворень дає змогу перейти від негаусівських до гаусівських даних із врахуванням кореляції між факторами. В роботі [27] доведено, що зі збільшенням кількості незалежних факторів, які представлені метриками коду в регресійних моделях, збільшується рівень достовірності оцінювання параметра розміру при обробці інформації за метриками коду програмних застосунків, зокрема JAVA-застосунків. Також це дає змогу отримати меншу ширину довірчого інтервалу та інтервалу передбачення у порівнянні з нелінійними регресійними моделями з меншою кількістю факторів [48, 49]. Застосування методів перехресної перевірки та використання вибірок достатнього розміру дає змогу досягти більшої стійкості регресійних моделей у порівнянні з регресійними моделями, які побудовані на

вибірках з відносно невеликим розміром та без перевірки якості моделі на тестових вибірках [33, 34, 57].

У роботі [22] запропоновано трьохфакторне лінійне регресійне рівняння для оцінювання кількості рядків коду JAVA-застосунків на основі методів множинного лінійного регресійного аналізу із використанням метрик загальної кількості класів (CLASS), загальної кількості зв'язків між класами та інтерфейсами (CBO) та загальної кількості атрибутів класів (TFQ) у вихідному коді. У роботі [23] удосконалено трьохфакторні лінійні регресійні рівняння для оцінювання кількості рядків коду для великих промислових інформаційних JAVA-систем на основі 16 рядків даних, JAVA-застосунків з відкритим кодом на основі 30 рядків даних та узагальнене лінійне регресійне рівняння на основі сукупного набору з 46 точок даних. Як метрики використано значення загальної кількості класів (CLASS), загальної кількості зв'язків між класами та інтерфейсами (CBO) та середнє значення кількості атрибутів на клас (aTFQ).

Робота [24] присвячена удосконаленню обробки інформації з метрик ПЗ для оцінювання кількості рядків коду для JAVA промислових інформаційних систем. У ній представлено трьохфакторну нелінійну регресійну модель на основі чотирьохвимірною нормалізуючого перетворення Джонсона для сімейства SB, за метриками програмного коду – кількості класів (CLASS), загальної кількості зв'язків між класами та інтерфейсами (CBO) та значення середньої кількості атрибутів на клас (aTFQ) із концептуальної моделі застосунку. Ця математична модель побудована із використанням невеликої вибірки 32 значень з набору даних з робіт [22, 23]. Вона має прийнятний рівень точності, але до її недоліків слід віднести відсутність перевірки отриманих результатів із застосуванням тестового набору даних та використання застарілого набору даних. Слід зазначити, що мова програмування JAVA зазнала суттєвих змін за останні 20 років, від моменту збирання отриманих метрик до сьогодення. Крім того, відбулись суттєві зміни у підходах, пов'язаних із технологічними особливостями розробки ПЗ. Тому трьохфакторна нелінійна регресійна модель [24] не може забезпечити необхідний рівень достовірності

обробки інформації для оцінювання параметра розміру для сучасних програмних проєктів, які розробляються за допомогою мови програмування JAVA.

У роботі [25] побудовано нелінійну регресійну модель для обробки інформації з оцінювання розміру веб-застосунків, які створені мовою програмування JAVA. Запропонована математична модель побудована із застосуванням одновимірного нормалізуючого перетворення Джонсона сімейства SB і використовує метрику кількості класів (CLASS) як незалежний фактор. Модель побудована із використанням 30 точок даних та обмежується категорією JAVA веб-застосунків для оцінювання параметра розміру, крім того, перевірку достовірності оцінювання не було проведено за допомогою тестової вибірки.

У роботі [26] розроблено чотирьохфакторну нелінійну регресійну модель обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків з відкритим кодом. Модель створено на основі багатовимірного нормалізуючого перетворення Джонсона сімейства SB. Для оцінювання параметра рядків коду використані метрики кількості класів (CLASS), кількості статичних методів (NOSM), загального значення згуртованості методів класу (LCOM – lack of cohesion of methods) та кількості унікальних викликів методу в класі (RFC). Цю модель побудовано із використанням 38 точок даних. До недоліків моделі слід віднести застосування метрики якості RFC та відсутність валідації результатів за допомогою тестового набору даних, для забезпечення перевірки достовірності обробки інформації з метрик коду для оцінювання розміру.

На основі емпіричних даних сучасних JAVA-застосунків з урахуванням технічних особливостей розробки, автором дисертації було побудовано одно-, двох-, трьохфакторні нелінійні регресійні моделі [27, 30]. Отримані однофакторні і двохфакторні нелінійні регресійні моделі не досягли необхідного рівня достовірності обробки інформації для оцінювання параметра розміру ПЗ відповідно до критеріїв якості нелінійних регресійних моделей, але підтверджують, що збільшення кількості факторів забезпечує підвищення

достовірності оцінювання. Трьохфакторна [30] модель покращує існуючу трьохфакторну нелінійну регресійну модель [26] та має прийнятний рівень достовірності обробки інформації для оцінювання параметра розміру, а також забезпечує перевірку якості побудованої моделі за допомогою тестової вибірки.

В роботах [28, 29] побудовано трьохфакторну нелінійну регресійну модель обробки інформації для оцінювання розміру DS та ML JAVA-застосунків на основі перетворення десяткового логарифма та метрик CLASS, TFQ та VMQ. Ця модель має прийнятний рівень достовірності обробки інформації для оцінювання параметра розміру відповідно до оцінок критеріїв якості, але до недоліків моделі слід віднести відсутність валідації результатів за допомогою тестової вибірки та розрахунку інтервалів передбачення і довірчих інтервалів.

Інші існуючі математичні моделі [47-54] не забезпечують достатнього рівня якості обробки інформації з метрик коду JAVA-застосунків, оскільки вони побудовані для зовсім інших мов програмування або категорій ПЗ.

Таким чином аналіз існуючих моделей та методів обробки інформації з метрик коду для оцінювання параметрів розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків показав, що існуючі лінійні регресійні рівняння та нелінійні регресійні моделі не завжди забезпечують достовірну обробку інформації з метрик коду для оцінювання параметра розміру цих категорій ПЗ. Отже, побудова математичних моделей обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків є актуальною науково-практичною задачею, яка потребує вирішення [58].

1.2 Формування набору даних для обробки інформації з метрик коду загальної вибірки JAVA-застосунків

Набір даних та його розмір мають ключове значення для побудови регресійних моделей, оскільки якість, обсяг і репрезентативність генеральної сукупності безпосередньо впливають на якість моделі та на достовірне відображення кореляції між змінними, мінімізують ризики викривлення оцінок

параметрів моделей, підвищують стійкість моделей до шуму та викидів. У цьому підрозділі представлено набір даних загальних JAVA-застосунків та описано механізми його перетворення для побудови багатofакторних нелінійних регресійних моделей.

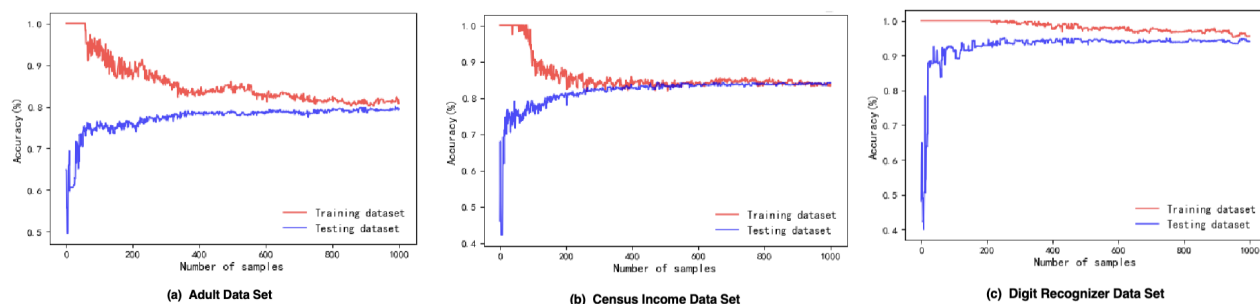


Рисунок 1.1 – Графіки оцінки точності прогнозування залежно від розмірів навчальної (Training) і тестової (Testing) вибірок [34]

Для побудови нелінійної регресійної моделі обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків та порівняння математичних моделей за критеріями якості були зібрані дані за метриками програмного коду 571 JAVA-застосунку з відкритим кодом, розташованих на платформі GitHub (<https://github.com>). За допомогою інструменту СК [59] було зібрано дані метрики: кількість рядків коду (KLOC), загальна кількість класів та інтерфейсів разом (CLASS) та їхній тип (TYPE), загальна кількість унікальних викликів методу в класі (RFC), загальне значення згуртованості методів класу (LCOM), загальна кількість статичних методів (NOSM), загальна кількість всіх методів (TMQ), загальна кількість видимих методів (VMQ), загальна кількість атрибутів класів (TFQ), загальна кількість видимих атрибутів класів (VFQ), абсолютне значення зв'язків між усіма класами та інтерфейсами (CBO), абсолютне значення глибини дерева успадкування (DIT), загальна кількість нащадків класів та інтерфейсів (NOC). Метрику CLASS було розділено на фактичну кількість класів (CLS) та фактичну кількість інтерфейсів (INFC).

Серед наведених метрик для побудови п'ятифакторної нелінійної регресійної моделі обрано метрики загальної кількості класів (CLS), загальної

кількості інтерфейсів (INFC), загальної кількості видимих методів (VMQ), загальної кількості атрибутів класів (TFQ), абсолютного значення зв'язків між усіма класами та інтерфейсами (CBO), оскільки ці метрики доступні із UML-діаграми класів і представляють всі необхідні характеристики, зокрема класи, інтерфейси, кількість полів, кількість методів та кількість зв'язків між класами та інтерфейсами. Для проведення подальшої обробки та нормалізації багатовимірних даних необхідно кожне значення кількості інтерфейсів на проєкт (INFC) збільшити на 1 із метою застосування одновимірних та багатовимірних нормалізуючих перетворень, оскільки існують проєкти JAVA-застосунків, де інтерфейси відсутні.

Отриманий набір даних розділено випадковим чином на навчальну і тестову вибірки з розмірами обсягом 286 та 285 рядків даних відповідно. Навчальний набір даних використано для побудови математичної моделі, а тестовий набір даних – для незалежного контролю якості обробки інформації з метрик коду для оцінювання параметра розміру. Ці набори даних наведено в таблиці 1.1 та в таблиці 1.2 у скороченому вигляді. Навчальний та тестовий набори даних інформації з метрик коду загальних JAVA-застосунків наведено в повному обсязі в додатку Б.

Таблиця 1.1 – Навчальний набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	2	3	4	5	6	7	8
1	mosaic.git	60,698	1334	170	6623	3657	8842
2	mp4parser.git	37,091	657	23	3852	2055	3300
3	closure-stylesheets.git	27,376	504	25	3214	1081	4197
4	animation-samples.git	4,47	132	10	482	290	779
5	shifu.git	61,27	765	20	4306	2936	4674
6	s2-geometry-library-java.git	53,506	635	61	4253	1414	4841
7	directory-server.git	119,761	1315	75	9372	3809	12676

Продовження табл.1.1

1	2	3	4	5	6	7	8
...	...	...	...	...	...	...	...
280	incubator-seatunnel.git	105,104	2455	210	9591	8376	16794
281	spring-data-couchbase.git	32,67	608	306	3631	1428	6197
282	initializr.git	29,705	658	46	3610	1107	3907
283	missinglink.git	2,331	89	15	263	56	389
284	clojure.git	46,79	391	526	4345	1850	3301
285	ec2-plugin.git	11,536	175	5	957	487	1365
286	leshan.git	56,448	960	141	5793	2220	8292

Таблиця 1.2 – Тестовий набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	2	3	4	5	6	7	8
1	subzero.git	3,531	126	3	269	170	511
2	remoting.git	33,008	540	49	2624	1377	2715
3	guice.git	79,577	3424	325	7403	2154	14121
4	aurora.git	49,435	1187	134	4459	2919	9478
5	otter.git	68,163	1155	141	6266	2816	5881
6	jackrabbit-oak.git	542,811	8229	618	46921	20819	58452
7	milo.git	127,563	1606	447	18077	11552	17550
...	...	...	...	...	...	...	...
279	maven-plugins.git	96,288	1478	53	7743	4043	7458
280	MVI-Dispatcher.git	2,512	68	13	323	141	344
281	warnings-plugin.git	7,908	222	0	856	484	1291
282	perforce-plugin.git	13,475	118	6	1233	460	699
283	directory-kerby.git	37,895	984	38	4911	1223	4567
284	maven.git	78,361	1306	349	9237	2996	9849
285	rocketmq-externals.git	9,956	236	17	1030	798	1461

Під час побудови багатфакторних нелінійних регресійних моделей виникає проблема мультиколінеарності між незалежними факторами в навчальній вибірці, якщо незалежні фактори мають високий рівень взаємозалежності (кореляції) між собою. Високий рівень кореляції між незалежними факторами спричиняє збільшення чутливості моделі до випадкових змін у даних, знижує стабільність моделі, призводить до спотворення оцінок ваг параметрів моделі та до зростання значень стандартних похибок [60]. Багатовимірні дані навчальної вибірки перевірено на наявність мультиколінеарності за коефіцієнтами впливу дисперсії (VIFs) між незалежними факторами. Для багатовимірних даних з k незалежними факторами X_i , $i = 1, 2, \dots, k$, коефіцієнти VIFs представлені діагональними елементами оберненої коваріаційної $k \times k$ матриці. Якщо значення коефіцієнта VIF перевищує 10, то це свідчить про наявність проблем із мультиколінеарністю, якщо коефіцієнт VIF наближається до 10, то виникає потенційна ймовірність появи проблем із мультиколінеарністю при побудові математичних моделей під час ітеративного очищення даних від викидів. Для факторі CLS, INFC, VMQ, TFQ та CBO значення коефіцієнтів VIFs дорівнюють 16,4, 2,3, 13,2, 10,8 та 20,9 відповідно, що свідчить про наявність значної мультиколінеарності між незалежними факторами. Для зменшення мультиколінеарності між факторами застосовано алгоритм, який викладено в [61, 62], в основу якого покладено підхід, який базується на ітеративному очищенні даних від викидів в багатовимірних даних із використанням нормалізуючого перетворення Бокса-Кокса. Внаслідок очищення даних від викидів було виявлено, що оцінки коефіцієнтів VIFs зростають під час поступового видалення викидів. Очищення даних від викидів було проведено за 32 ітерації, в таблиці 1.3 наведено значення коефіцієнтів VIFs для окремих ітерацій.

Таблиця 1.3 – Значення коефіцієнтів VIFs для багатовимірних даних CLS, INFC, VMQ, TFQ та CBO при ітеративному очищенні даних від викидів

Ітерація	CLS	INFC	VMQ	TFQ	CBO
1	16,4	2,3	13,2	10,8	20,9
5	16,9	3,5	14,0	10,8	22,4
10	17,5	3,6	14,6	12,1	29,0
15	18,6	4,6	16,0	13,2	32,0
20	18,7	4,6	16,1	13,2	32,3
25	19,6	4,6	16,2	13,3	32,2
32	25,6	4,5	18,5	15,6	36,4

Таблиця 1.3 демонструє, що вже початкова навчальна вибірка має суттєві проблеми з мультиколінеарністю для обраних факторів. Крім того, поетапне видалення викидів показало, що оцінки коефіцієнтів VIFs мають тенденцію до зростання на кожній наступній ітерації у порівнянні з попередньою. Остання ітерація показує значне підвищення оцінок коефіцієнтів VIFs у порівнянні з першою ітерацією, що дає підстави зробити висновок про те, що поступове очищення вибірки від викидів призводить до збільшення мультиколінеарності між незалежними факторами.

Для вирішення проблеми мультиколінеарності абсолютні значення метрик VMQ, TFQ та CBO замінено на їх середні значення відносно суми загальної кількості класів та інтерфейсів з програмного проєкту – $aVMQ$, $aTFQ$ та $aCBO$ відповідно [63]. Розрахунок середніх значень цих метрик виконано таким чином:

$$aVMQ = \frac{VMQ}{CLS+INFC} \quad (1.1)$$

де $aVMQ$ – середнє значення метрики VMQ проєкту.

$$aTFQ = \frac{TFQ}{CLS+INFC} \quad (1.2)$$

де $aTFQ$ – середнє значення метрики TFQ проєкту.

$$aCBO = \frac{CBO}{CLS+INFC} \quad (1.3)$$

де аСВО – середнє значення метрики СВО проєкту.

Для факторів CLS, INFC, аVMQ, аTFQ та аСВО розраховані значення оцінок коефіцієнтів VIFs дорівнюють 2,2, 2,3, 1,1, 1,2 та 1,1, що свідчить про відсутність мультиколінеарності між незалежними факторами. Відповідно до методів побудови нелінійних регресійних моделей [61, 62], багатовимірні дані аналогічно було очищено від викидів під час побудови нелінійної регресійної моделі на базі шестивимірного нормалізуючого перетворення Бокса-Кокса та перевірено на наявність мультиколінеарності між незалежними факторами. Результати перевірки показали, що при поступовому вилученні викидів – кореляція між незалежними факторами відсутня, оскільки значення коефіцієнтів VIFs не перевищують граничного значення 10. Зміну значень коефіцієнтів VIFs залежно від ітерації вилучення викидів наведено у таблиці 1.4.

Таблиця 1.4 – Значення коефіцієнтів VIFs для багатовимірних даних CLS, INFC, аVMQ, аTFQ та аСВО при ітеративному очищенні даних від викидів

Ітерація	CLS	INFC	аVMQ	аTFQ	аСВО
1	2,1	2,3	1,1	1,1	1,6
5	2,3	2,3	1,1	1,2	1,1
10	3,1	3,2	1,2	1,2	1,2
15	3,2	3,2	1,2	1,2	1,2
20	3,9	3,9	1,2	1,2	1,3
26	4,0	4,0	1,2	1,2	1,3

Отже вибір метрик CLS, INFC, аVMQ (середнє значення VMQ відносно суми класів та інтерфейсів), аTFQ (середнє значення TFQ відносно суми класів та інтерфейсів) та аСВО (середнє значення СВО відносно суми класів та інтерфейсів) для подальшої побудови математичних регресійних моделей є виправданим, оскільки на всіх ітераціях очищення даних від викидів оцінки VIF

для кожної метрики залишались в межах норми і не перевищили граничного значення.

Характеристики розподілу незалежної змінної загальної кількості класів (CLS) для навчальної та тестової вибірок наведено в таблиці 1.5.

Таблиця 1.5 – Характеристики розподілу CLS навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	1158,6993	1136,0386
Медіана	601	527
Середньоквадратичне відхилення	1626,2984	1610,2051
Асиметрія	2,9241	2,7383
Ексцес	9,9433	8,6873
Мінімальне значення	33	29
Максимальне значення	10610	9475

Характеристики розподілу незалежної змінної загальної кількості інтерфейсів (INFC) для навчальної та тестової вибірок наведено в табл. 1.6.

Таблиця 1.6 – Характеристики розподілу INFC навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	118,2168	103,2632
Медіана	46	35
Середньоквадратичне відхилення	220,0131	173,4958
Асиметрія	4,6118	3,2005
Ексцес	29,7762	13,0054
Мінімальне значення	0	0
Максимальне значення	2128	1245

Характеристики розподілу незалежної змінної середньої кількості методів на клас та інтерфейс (aVMQ) для навчальної та тестової вибірок наведено в таблиці 1.7.

Таблиця 1.7 – Характеристики розподілу aVMQ навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	5,1938	5,0614
Медіана	4,9861	4,6960
Середньоквадратичне відхилення	2,0967	2,0455
Асиметрія	2,4375	1,7512
Ексцес	12,5273	5,3314
Мінімальне значення	1,7053	1,5893
Максимальне значення	21,0549	16,9950

Характеристики розподілу незалежної змінної середньої кількості полів на загальну суму класів та інтерфейсів (aTFQ) для навчальної та тестової вибірок наведено в таблиці 1.8.

Таблиця 1.8 – Характеристики розподілу aTFQ навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	2,2761	2,3865
Медіана	2,1235	2,1725
Середньоквадратичне відхилення	0,9469	1,1341
Асиметрія	0,9167	1,8200
Ексцес	1,4855	6,6142
Мінімальне значення	0,1327	0,0171
Максимальне значення	6,3365	9,1600

Характеристики розподілу незалежної змінної середньої кількості зв'язків відносно суми класів та інтерфейсів (aCBO) для навчальної та тестової вибірок наведено в таблиці 1.9.

Таблиця 1.9 – Характеристики розподілу аСВО навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	5,9372	6,0169
Медіана	5,9611	5,8475
Середньоквадратичне відхилення	1,6280	1,4871
Асиметрія	0,0538	0,2684
Ексцес	0,9299	0,5075
Мінімальне значення	0,1184	2,4444
Максимальне значення	12,5902	11,7681

1.3 Формування набору даних для обробки інформації з метрик Data Science та Machine Learning JAVA-застосунків

Для побудови нелінійної регресійної моделі обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків та порівняння якості обробки інформації за математичними моделями для оцінювання розміру таких застосунків було зібрано дані з метрик програмного коду 112 JAVA-застосунків з відкритим кодом, розташованих на платформі GitHub (<https://github.com>). За допомогою інструменту СК [59] отримано такі метрики: кількість рядків коду (KLOC), загальна кількість класів та інтерфейсів (CLASS) та їхній тип (TYPE), загальна кількість унікальних викликів методу в класі (RFC), загальне значення згуртованості методів класу (LCOM), загальна кількість статичних методів (NOSM), загальна кількість видимих методів (VMQ), загальна кількість атрибутів класів (TFQ), абсолютне значення зв'язків між усіма класами та інтерфейсами (CBO) та абсолютне значення глибини дерева успадкування (DIT). Метрику CLASS було розділено на фактичну кількість класів (CLS) та фактичну кількість інтерфейсів (INFC).

Серед наведених метрик для побудови п'ятифакторної нелінійної регресійної моделі обрано метрики загальної кількості класів (CLS), загальної кількості інтерфейсів (INFC), загальної кількості видимих методів (VMQ),

загальної кількості атрибутів класів (TFQ), абсолютного значення зв'язків між усіма класами та інтерфейсами (CBO), оскільки ці метрики доступні із UML-діаграми класів на ранніх етапах проектування ПЗ. Для проведення обробки та нормалізації багатовимірних даних, згідно з методикою (див. пп. 1.2), кожне значення загальної кількості інтерфейсів проєкту (INFC) збільшено на 1 для подальшого використання одновимірних та багатовимірних нормалізуючих перетворень.

Отриманий набір даних розділено випадковим чином на навчальну і тестову вибірки обсягом 74 та 38 рядків даних відповідно. Навчальний набір даних використано для побудови математичної моделі, а тестовий набір даних – для незалежного контролю якості обробки інформації з метрик коду для оцінювання параметра розміру. Ці набори даних наведено в таблиці 1.10 та таблиці 1.11 у скороченому вигляді. Навчальний та тестовий набори даних інформації з метрик коду DS та ML JAVA-застосунків наведено у повному обсязі в додатку В.

Таблиця 1.10 – Навчальний набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	atlas	154,568	1757	89	11778	14096	12936
2	curator	49,161	1088	248	4666	5087	6648
3	hama	47,105	667	82	4196	4540	4538
4	helix	156,854	2282	144	11188	12677	15681
5	iceberg	258,993	3667	316	23662	27844	34060
6	incubator-celeborn	28,49	396	24	2379	2693	3159
...	...	...	...	...	...	...	...
70	pattern	3,146	128	2	421	454	654
71	graph-data-science	200,338	4536	1083	21631	24647	35804
72	nexus-repository-r	2,877	74	4	311	353	697
73	ml	4,216	134	5	473	502	727
74	systemds	207,825	2329	12	18941	21583	18056

Таблиця 1.11 – Тестовий набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	mastering-java-data-science	11,977	242	2	895	358	1635
2	bigtop	6,637	223	12	638	519	1114
3	java-data-mining-package	30,378	613	36	3870	1130	3705
4	h2o-2	119,202	1721	20	8546	4758	10532
5	MiA	3,847	117	5	252	137	706
6	mahout	97,394	1754	100	8195	3743	12033
...	...	...	...	...	...	...	...
34	BoostSRL	89,489	525	31	5971	3149	4490
35	DigitRecognizer	0,844	16	0	47	53	80
36	DeepLearning	1,083	11	0	60	71	50
37	onyx	10,392	207	11	814	410	790
38	opencv-object-detection	0,35	7	0	25	15	36

Отримані багатовимірні дані навчальної вибірки було перевірено на наявність мультиколінеарності за коефіцієнтами впливу дисперсії (VIFs) між незалежними факторами. Для факторів CLS, INFC, VMQ, TFQ і CBO розраховані VIF становили 21,6, 3,6, 12,8, 8,6 і 30,2 відповідно, що підтверджує наявність мультиколінеарності серед незалежних факторів.

Для зменшення мультиколінеарності між незалежними факторами абсолютні значення метрик VMQ, TFQ та CBO замінено на їхні середні значення відносно суми загальної кількості класів та інтерфейсів з програмного проєкту – aVMQ, aTFQ та aCBO відповідно. Розрахунок середніх значень цих метрик виконано за формулами 1.1, 1.2 та 1.3 [63].

Для факторів CLS, INFC, aVMQ, aTFQ і aCBO розраховані VIF становлять 2,38, 2,47, 1,33, 1,29 і 1,20 відповідно для першої ітерації видалення викидів із навчальної вибірки та 2,46, 2,57, 1,39, 1,31 і 1,23 відповідно для останньої ітерації, що вказує на відсутність мультиколінеарності між незалежними факторами. Отже вибір середніх значень метрик VMQ, TFQ і CBO

відносно загальної кількості класів та інтерфейсів для побудови нелінійної регресійної моделі є виправданим.

Характеристики розподілу незалежної змінної загальної кількості класів (CLS) для навчальної та тестової вибірок наведено в таблиці 1.12.

Таблиця 1.12 – Характеристики розподілу CLS навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	1258,9459	613,5789
Медіана	621	483,5
Середньоквадратичне відхилення	1784,2267	598,1681
Асиметрія	3,1534	0,7372
Ексцес	13,3710	-0,7370
Мінімальне значення	30	6
Максимальне значення	11230	1916

Характеристики розподілу незалежної змінної загальної кількості інтерфейсів (INFC) для навчальної та тестової вибірок наведено в табл. 1.13.

Таблиця 1.13 – Характеристики розподілу INFC навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	120,7568	45,6842
Медіана	47,5	27,5
Середньоквадратичне відхилення	227,6980	60,8863
Асиметрія	3,8247	1,9489
Ексцес	16,7620	3,7537
Мінімальне значення	0	0
Максимальне значення	1386	248

Характеристики розподілу незалежної змінної середньої кількості методів відносно суми класів та інтерфейсів (aVMQ) для навчальної та тестової вибірок наведено в таблиці 1.14.

Таблиця 1.14 – Характеристики розподілу aVMQ навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	5,4591	4,9109
Медіана	4,9873	4,9393
Середньоквадратичне відхилення	2,5518	2,0708
Асиметрія	2,1710	0,5921
Ексцес	7,3558	0,3341
Мінімальне значення	1,5893	1,6800
Максимальне значення	17,8145	10,7392

Характеристики розподілу незалежної змінної середньої кількості полів відносно суми класів та інтерфейсів (aTFQ) для навчальної та тестової вибірок наведено в таблиці 1.15.

Таблиця 1.15 – Характеристики розподілу aTFQ навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	2,5823	2,8159
Медіана	2,3632	2,4172
Середньоквадратичне відхилення	0,9606	1,5098
Асиметрія	0,4745	1,6196
Ексцес	-0,8561	3,9208
Мінімальне значення	0,9976	0,2600
Максимальне значення	4,5982	8,2032

Характеристики розподілу незалежної змінної середньої кількості зв'язків відносно суми класів та інтерфейсів (aCBO) для навчальної та тестової вибірок наведено в таблиці 1.16.

Таблиця 1.16 – Характеристики розподілу aCBO навчальної та тестової вибірок

Параметри розподілу вибірки	Навчальна	Тестова
Середнє	6,5483	6,5463
Медіана	6,5129	6,1507
Середньоквадратичне відхилення	1,5683	2,1054
Асиметрія	0,6959	1,0224
Ексцес	2,2037	1,2788
Мінімальне значення	2,8895	3,0714
Максимальне значення	12,5902	12,3634

1.4 Обґрунтування необхідності проведення досліджень за обраною темою

Вибір нелінійних регресійних моделей як інструменту для оцінювання розміру JAVA-застосунків, на відміну від альтернативних методів оцінювання, таких як нейронні мережі, нечітка логіка, методи експертних оцінок, методи аналогій, методи функціональних точок, байєсівські мережі довіри, метод опорних векторів тощо, зумовлений необхідністю забезпечення прозорості, стійкості та прогностичної здатності оцінювання параметра розміру [64-68]. Переваги вибору регресійних моделей полягають у такому:

- На відміну від методів штучного інтелекту, які функціонують за принципом «чорної скриньки», нелінійний регресійний аналіз дозволяє описати залежність параметра розміру від незалежних факторів у вигляді математичної формули. Такий підхід надає більш широкий доступ до моделі та дає змогу інтегрувати її в будь-які програмні рішення.

- Нелінійні регресійні моделі демонструють кращу стабільність на наборах даних середнього розміру. У той час як методи на основі штучного

інтелекту вимагають значних масивів даних для запобігання перенавчанню, регресійні моделі дозволяють отримувати статистично значущі оцінки без необхідності використання надвеликих наборів даних.

- Регресійний аналіз дозволяє отримати оцінки довірчого інтервалу та інтервалу передбачення, що є перевагою над методами аналогій, функціональних точок, нечіткої логіки тощо. Це надає можливість менеджерам проєктів оцінити ризики за різних сценаріях розвитку процесу розробки.

- При побудові регресійних моделей з емпіричного набору даних вилучають аномальні або помилкові значення (так звані викиди), що забезпечує підвищення точності та стійкості отриманих математичних моделей і зменшення похибки оцінювання.

Для обґрунтування необхідності дослідження проведемо порівняльний аналіз достовірності обробки інформації для оцінювання розміру JAVA-застосунків для існуючих моделей [58].

Щоб оцінити якість прогнозування за регресійними моделями доцільно використати такі критерії, як коефіцієнт детермінації R^2 , значення середньої величини відносної похибки $MMRE$ [33] та значення відсотка передбачення для рівня відносної похибки 0,25 – $PRED(0, 25)$ [33].

Коефіцієнт детермінації R^2 визначається як

$$R^2 = 1 - \frac{SSE}{SST}, \quad (1.4)$$

де SSE – сума квадратів залишків (the residual sum of squares) або сума квадратів помилок (the error sum of squares), $SSE = \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$; SS – загальна сума квадратів (the total sum of squares), $SST = \sum_{i=1}^N (Y_i - \bar{Y})^2$, та $\bar{Y}$ – математичне сподівання.

Для врахування впливу незалежних факторів нелінійної регресії на точність прогнозування моделі зазвичай використовується скориговане значення

коефіцієнту множинної детермінації R_{adj}^2 [67]. R_{adj}^2 визначається наступним чином

$$R_{adj}^2 = 1 - \left(1 - R^2\right) \frac{N-1}{N-k-1}, \quad (1.5)$$

де N – розмір вибірки, k – кількість незалежних факторів моделі.

Критерій $MMRE$ визначається як

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i, \quad (1.6)$$

де MRE_i – значення величини відносної похибки для i -го рядку даних випадкової величини

$$MRE_i = \left| \frac{(Y_i - \hat{Y}_i)}{Y_i} \right|. \quad (1.7)$$

Значення відсотка передбачення для рівня відносної похибки 0,25 $PRED$ розраховується як

$$PRED(0, 25) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{if } MRE_i \leq 0,25 \\ 0 & \text{otherwise} \end{cases}. \quad (1.8)$$

Якість моделі вважається задовільною, коли показники $R^2 \geq 0,75$, $R_{adj}^2 \geq 0,75$, $MMRE \leq 0,25$ та $PRED(0, 25) \geq 0,75$.

Проведемо порівняння якості обробки інформації для оцінювання параметра розміру загальних JAVA-застосунків за критеріями R^2 , $MMRE$ та $PRED(0, 25)$ для трьохфакторної нелінійної регресійної моделі [30].

Інші моделі [22-27] було відкинуто через застарілі дані, обмеження нормалізуючих перетворень або відмінність технологічних факторів розробки JAVA-застосунків та недостатню достовірність обробки інформації для оцінювання параметра розміру.

Оцінки критеріїв якості математичної моделі [30] обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків становлять $R^2 = 0,9073$, $MMRE = 0,1645$, $PRED(0, 25) = 0,7692$ для навчальної

вибірки та $R^2 = 0,9016$, $MMRE = 0,1617$, $PRED(0,25) = 0,8175$ для тестової вибірки. Слід зазначити, що модель [30] має достатньо високі оцінки якості оцінювання параметра розміру, але недоліком цієї моделі є те, що її побудовано за допомогою метрики RFC, яку неможливо достовірно визначити на ранньому етапі проєктування програмної системи. Це безпосередньо впливає на подальшу достовірність обробки інформації з метрик коду для оцінювання параметра розміру та трудомісткості на ранніх етапах проєктування програмних застосунків [69].

Оскільки в дисертаційній роботі окремим завданням виноситься побудова математичних моделей обробки інформації для оцінювання розміру JAVA-застосунків, в яких враховано технологічні особливості розробки та прикладні аспекти застосування, а саме DS та ML застосунків, необхідно провести порівняння існуючих моделей щодо їх якості обробки інформації з метрик коду ПЗ.

Для аналізу достовірності обробки інформації за метриками коду з оцінювання параметра розміру DS та ML JAVA-застосунків порівняємо трьохфакторну нелінійну регресійну модель на базі десятичного логарифма [28] та трьохфакторну нелінійну регресійну модель на основі перетворення Джонсона сім'ї SB [30] за критеріями якості R^2 , $MMRE$ та $PRED(0,25)$.

Таблиця 1.17 – Порівняння якості математичних моделей обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків.

Вид моделі	Навчальна вибірка			Тестова вибірка		
	R^2	MMRE	PRED (0,25)	R^2	MMRE	PRED (0,25)
Трьохфакторна на основі перетворення Джонсона SB [30]	0,8908	0,1758	0,7568	0,9269	0,2025	0,6579
Трьохфакторна на основі перетворення десятичного логарифма для DS та ML [28]	0,9226	0,1853	0,7703	0,9347	0,2073	0,6316

Моделі [28, 30] показали прийнятні значення показників якості оцінювання за критеріями R^2 та $MMRE$ для навчального та тестового наборів даних, але значення порогового рівня критерію $PRED(0, 25)$ не було досягнуто для чотирьохфакторної моделі для навчальної вибірки та для всіх моделей для тестової вибірки. Слід зазначити, що при побудові моделі [30] було використано метрику RFC, яку неможливо достовірно визначити на ранньому етапі проєктування програмної системи із UML-діаграми класів. Це безпосередньо впливає на подальшу достовірність обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків, і як наслідок – прогнозування трудомісткості програмного проєкту в цілому [69].

З огляду на це, на базі проведеного аналізу можна зробити висновок, що існуючі нелінійні регресійні моделі не забезпечують достатньої якості обробки інформації для оцінювання параметра розміру як для загальних JAVA-застосунків, так і для DS та ML JAVA-застосунків. Отже, удосконалення чотирьохфакторних та побудова п'ятифакторних нелінійних регресійних моделей для оцінювання інформації з метрик коду загальних JAVA-застосунків, DS та ML JAVA-застосунків за метриками коду, які доступні з UML-діаграми класів є актуальною науково-практичною задачею та подальші дослідження є необхідними.

1.5. Висновки за розділом 1 і постановка завдань дослідження

У першому розділі наведено аналіз математичних моделей для обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків та огляд методів і моделей для оцінювання трудомісткості розробки, які використовують параметр розміру у вигляді кількості рядків коду ПЗ. Здійснено обґрунтування необхідності проведення дослідження за обраною темою.

1) Визначено, що параметричні моделі оцінювання трудомісткості, такі як COCOMO II та COSYSMO, дають змогу надати кількісну оцінку трудомісткості, базуючись на метриці програмного коду, а саме його розміру, та враховують технологічні фактори середовища розробки ПЗ (мову програмування, прикладну сферу застосування тощо).

2) Проведено огляд та аналіз існуючих лінійних регресійних рівнянь та нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків. Виявлено, що існуючі лінійні регресійні рівняння та нелінійні регресійні моделі не завжди забезпечують достовірну обробку інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків, оскільки вони побудовані на вибірках невеликого розміру та їх достовірність оцінювання не було перевірено на тестових вибірках.

3) Зібрано набори даних інформації з метрик коду загальних JAVA-застосунків, що складається з 571 багатовимірних векторів характеристик, DS та ML JAVA-застосунків, що складається з 112 багатовимірних векторів характеристик для перевірки достовірності обробки інформації для оцінювання параметра розміру за допомогою існуючих математичних моделей та подальшої побудови нелінійних регресійних математичних моделей. Отриманий набір даних загальних JAVA-застосунків розділено на навчальну вибірку розміром 286 точок даних та тестову вибірку розміром 285 точок даних, та набір даних DS та ML JAVA-застосунків розділено на навчальну вибірку розміром 74 точки даних та тестову вибірку розміром 38 точок даних.

4) Проведено порівняння існуючих математичних моделей обробки інформації для оцінювання параметра розміру на вибірках даних за метриками загальних JAVA-застосунків, DS та ML JAVA-застосунків із використанням критеріїв якості R^2 , $MMRE$ та $PRED(0, 25)$. Показано, що існуючі нелінійні регресійні моделі обробки інформації з метрик коду для оцінювання кількості рядків коду загальних JAVA-застосунків, DS та ML JAVA-застосунків не

забезпечують достатнього рівня якості обробки інформації для оцінювання параметра розміру на навчальній та тестовій вибірках.

5) Обрано незалежні фактори з метрик коду ПЗ для побудови багатофакторних нелінійних регресійних моделей та проведено їх перевірку на мультиколінеарність. Запропоновано використовувати відносну характеристику середнього значення певних метрик до загальної суми класів та інтерфейсів проєкту, що дає змогу розв'язати проблему мультиколінеарності між незалежними факторами.

Результати дослідження поточного розділу автором опубліковано в [9-11, 27-32, 63].

РОЗДІЛ 2. ВИБІР НОРМАЛІЗУЮЧОГО ПЕРЕТВОРЕННЯ ДЛЯ ПОБУДОВИ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК КОДУ JAVA-ЗАСТОСУНКІВ

У другому розділі проведено дослідження існуючих взаємозворотних нормалізуючих перетворень, які використовують при побудові нелінійних регресійних моделей обробки інформації. Обрано перетворення для нормалізації шестивимірних негаусівських даних інформації з метрик JAVA-застосунків з метою їх подальшого використання під час побудови нелінійних регресійних моделей.

2.1 Методика вибору нормалізуючого перетворення

Вибір нормалізуючого перетворення – це необхідний крок для подальшого застосування під час побудови нелінійних регресійних моделей, що ґрунтується на припущенні про гаусівський розподіл емпіричних даних, застосовується до повної нормалізованої вибірки та включає ітеративну перевірку й виключення викидів за допомогою квадрата відстані Махаланобіса.

Нормалізація при обробці та аналізі даних забезпечує узгодженість масштабу числових ознак, що покращує продуктивність алгоритмів математичного моделювання та допомагає уникнути ситуацій, коли точки даних з найбільшими (або найменшими) абсолютними значеннями непропорційно впливають на результати моделювання. Нормалізуючі перетворення можна розділити на дві основні групи: одновимірні та багатовимірні. Багатовимірні перетворення частіше застосовуються для нормалізації даних, що представлені вектором, елементами якого є параметри майбутньої моделі. Реалізація багатовимірних перетворень є більш складним завданням, але такий підхід дозволяє зберегти цілісність структури даних, врахувати багатовимірний розподіл та кореляцію між змінними [70, 71]. Найчастіше для побудови

нелінійних регресійних моделей використовують такі нормалізуючі перетворення, як Бокса-Кокса та сімейства Джонсона.

Нормалізація даних та вилучення викидів дозволяє зменшити вплив аномальних значень та уникнути впливу шуму викидів на якість регресійних моделей і забезпечити об'єктивність, точність та стабільність оцінки перетворення відповідно до тесту Мардія [72].

Перевірка багатовимірних даних на відповідність нормальному розподілу здійснюється за допомогою критерію Мардія [72], що ґрунтується на відповідності нормальному закону розподілу параметрів багатовимірної асиметрії ($\beta_{1,k}$) та ексцесу ($\beta_{2,k}$)

$$\beta_{1,k} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left[(X_i - \bar{X})^T S_N^{-1} (X_j - \bar{X}) \right]^3, \quad (2.1)$$

$$\beta_{2,k} = \frac{1}{N} \sum_{i=1}^N \left[(X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}) \right]^2, \quad (2.2)$$

де X – k -розмірний вектор випадкової величини, $X = (X_1, X_2, \dots, X_k)$, а S_N — зміщена вибіркова коваріаційна матриця багатовимірної ВВ X , визначена як

$$S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})^T, \quad (2.3)$$

де $\bar{X}$ – це вектор вибірових середніх, $\bar{X} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_k)^T$.

За тестом Мардія очікувана асиметрія багатовимірних даних дорівнює 0, очікуваний ексцес становить $k(k + 2)$ для багатовимірного нормального закону розподілу.

Для $\beta_{1,k}$ застосовується тестова статистика

$$\frac{N}{6} \beta_{1,k} \leq \chi^2_{v, \alpha}, \quad (2.4)$$

яка має наближений розподіл χ^2 з $v = k(k + 1)(k + 2)/6$ ступенями вільності та рівнем значущості $\alpha = 0,005$.

Для $\beta_{2,k}$ у якості тестової статистики використовується $1-\alpha$ квантиль $\mathcal{N}$ нормального закону розподілу з математичним сподіванням $\mu = k(k + 2)$ та дисперсією $\sigma^2 = 8k(k + 2)/N$

$$\beta_{2,k} \leq \mathcal{N}_{1-\alpha}(\mu, \sigma^2). \quad (2.5)$$

У дисертаційній роботі з метою вибору нормалізуючих перетворень для побудови моделей обробки інформації з оцінювання параметра розміру загальних JAVA-застосунків було сформовано навчальну вибірку з 286 векторів, елементами яких є значення факторів CLS, INFC, aVMQ, aTFQ, aCBO, отриманих із UML-діаграми класів, та для оцінювання параметра розміру DS та ML JAVA-застосунків було сформовано навчальну вибірку з 74 векторів з аналогічними метриками.

2.2 Одновимірні взаємозворотні нормалізуючі перетворення

Одновимірні перетворення використовують для нормалізації параметрів моделі без урахування їх взаємозв'язків з іншими змінними набору даних. Це дозволяє зменшити асиметрію розподілу, стабілізувати дисперсію та наблизити розподіл емпіричних даних до нормального. Одновимірні нормалізуючі перетворення ефективні у випадках, коли змінні аналізують незалежно одна від одної або мають слабкий взаємозв'язок. Однак у ситуаціях з високочорельованими даними їх використання може бути обмеженим і, як наслідок, потребує застосування багатовимірних нормалізуючих перетворень. Найпоширенішими одновимірними взаємозворотними нормалізуючими перетвореннями є:

- перетворення десяткового логарифма, яке дозволяє ефективно зменшити розкид даних з великою варіантністю;
- степеневі трансформації, які стабілізують дисперсію;
- перетворення Бокса-Кокса, його окремий випадок – перетворення Чена-Джонсона;

- перетворення сімейства Джонсона.

2.2.1 Перетворення десяткового логарифма

Перетворення даних за допомогою десяткового логарифма є однією з найпростіших і найпоширеніших технік нормалізації, яку використовують для корекції розподілу змінних із вираженою правосторонньою асиметрією. Це перетворення дозволяє згладити вплив екстремальних значень, які можуть викривляти результати аналізу, роблячи їх менш чутливими до аномальних викидів [73, 74]. Перетворення десяткового логарифма представлене наступною формулою:

$$Z = \log_{10}(X), \quad (2.6)$$

де X – негаусівська ВВ та Z – гаусівська ВВ.

Зворотне перетворення має вигляд:

$$Z = 10^Z. \quad (2.7)$$

Одним із ключових обмежень цього методу є неможливість застосування до від’ємних та нульових значень.

2.2.2 Одновимірне перетворення Бокса-Кокса

Перетворення Бокса-Кокса, на відміну від десяткового логарифма, є більш гнучкішим та дає змогу визначити оптимальний ступінь трансформації для конкретного набору даних шляхом введення параметра λ , який підбирається таким чином, щоб мінімізувати відхилення розподілу емпіричних даних від нормального. Перетворення Бокса-Кокса ефективно працює для даних із додатною варіацією, які мають сильну правосторонню асиметрію або гетероскедастичність [75, 76]. Перетворення Бокса-Кокса має вигляд

$$Z = \begin{cases} (X^\lambda - 1)/\lambda, & \text{якщо } \lambda \neq 0 \\ \ln(X), & \text{якщо } \lambda = 0 \end{cases} \quad (2.8)$$

Зворотне перетворення Бокса-Кокса має такий вигляд

$$X = \begin{cases} \sqrt[\lambda]{(\lambda \cdot Z + 1)}, & \text{якщо } \lambda \neq 0 \\ e^X, & \text{якщо } \lambda = 0 \end{cases} \quad (2.9)$$

де λ – параметри перетворення Бокса-Кокса, X – негаусівська ВВ та Z – гаусівська ВВ.

Визначення оптимального параметра перетворення λ є критично важливим кроком, який має максимально наблизити розподіл перетвореної негаусівської величини якомога ближче до гаусівського розподілу. Оцінки параметрів нормалізуючого перетворення (2.8) найчастіше знаходяться за методом максимальної правдоподібності (ММП) [70]:

$$\hat{\lambda} = \operatorname{argmax} l(\lambda), \quad (2.10)$$

де $l(\lambda)$ – логарифмічна функція правдоподібності. Для перетворення Бокса-Кокса (2.8) логарифмічна функція правдоподібності для багатовимірних даних має вигляд

$$l(\lambda) = C - \frac{N}{2} \ln \sum_{i=1}^N \frac{(x(\lambda)_i - \overline{x(\lambda)})^2}{N} + (\lambda - 1) \sum_{i=1}^N \ln(x_i), \quad (2.11)$$

де C – константа, N – розмір вибірки, $\overline{x(\lambda)} = \frac{1}{N} \sum_{i=1}^N x(\lambda)_i$ – математичне сподівання [77].

Одним із суттєвих обмежень використання цього перетворення є те, що його застосовують лише для додатних значень та вимагає попередньої трансформації даних за наявності нульових або від’ємних спостережень. У разі нульових або від’ємних значень в емпіричних даних рекомендується використовувати окремий варіант перетворення Бокса-Кокса або перетворення Со-Джонсона [78]

2.2.3 Одновимірне перетворення Джонсона

Сімейства взаємозворотних нормалізуючих перетворень Джонсона, представлене в 1949 році, є потужним інструментом нормалізації даних, що дозволяє гнучко коригувати розподіл характеристик та привести їх значення до

нормального розподілу. Основна перевага перетворень Джонсона полягає в тому, що вони можуть адаптивно трансформувати широкий спектр розподілів без необхідності жорстких припущень про форму даних. У загальному випадку перетворення Джонсона мають вигляд [79]:

$$z = \gamma + \eta h(x, \varphi, \lambda); \eta > 0; -\infty < \gamma < \infty; \lambda > 0; -\infty < \varphi < \infty, \quad (2.12)$$

де h – функція сімейства розподілу Джонсона; $\gamma, \eta, \lambda, \varphi$ – параметри перетворення Джонсона, де γ та η – параметри форми, λ – параметр масштабу, φ – параметр положення, який характеризує зсув розподілу у напрямку осі абсцис.

Функція h представляє одну із трьох сімей розподілу Джонсона:

$$1) \text{ для сім'ї SL: } h(x, \varphi, \lambda) = \ln\left(\frac{x-\varphi}{\lambda}\right), \text{ де } x > \varphi;$$

$$2) \text{ для сім'ї SB: } h(x, \varphi, \lambda) = \ln\left(\frac{x-\varphi}{\lambda+\varphi-x}\right), \text{ де } \varphi < x < \varphi + \lambda;$$

$$3) \text{ для сім'ї SU: } h(x, \varphi, \lambda) = \operatorname{Arsh}\left(\frac{x-\varphi}{\lambda}\right), \text{ де } \varphi < x < \varphi + \lambda,$$

$$\text{де } \operatorname{Arsh}\left(\frac{x-\varphi}{\lambda}\right) = \ln\left[\frac{x-\varphi}{\lambda} + \sqrt{\left(\frac{x-\varphi}{\lambda}\right)^2 + 1}\right].$$

Якщо ввести позначення $t = \frac{(x-\varphi)}{\lambda}$, то функція h для сімейств розподілів Джонсона має вигляд

$$1) \text{ для сім'ї SL: } h = \ln(t), \text{ де } x > \varphi;$$

$$2) \text{ для сім'ї SB: } h(x, \varphi, \lambda) = \ln\left(\frac{t}{1-t}\right), \text{ де } \varphi < x < \varphi + \lambda;$$

$$3) \text{ для сім'ї SU: } h(x, \varphi, \lambda) = \operatorname{Arsh}(t), \text{ де } -\infty < x < \infty.$$

Тоді зворотне перетворення до (2.12) визначається наступним чином

$$x = \varphi + \lambda h^{-1}(z, \gamma, \varphi); \eta > 0; -\infty < \gamma < \infty; \lambda > 0; -\infty < \varphi < \infty, \quad (2.13)$$

де значення функції h^{-1} приймає наступні значення

$$1) \text{ для сім'ї SL: } h^{-1} = e^{\zeta};$$

$$2) \text{ для сім'ї SB: } h^{-1} = \frac{1}{(1+e^{-\zeta})};$$

$$3) \text{ для сім'ї SU: } h^{-1} = \frac{(e^{\zeta}-e^{-\zeta})}{2};$$

$$\text{де } \zeta = \frac{(z-\gamma)}{\eta}.$$

Сім'ї розподілів Джонсона відрізняються багатостатністю форм і у площині квадрата асиметрії A^2 та ексцесу ε займають значні області. На рис.2.1 представлені області комбінацій A^2 і ε для різних розподілів Джонсона. Ця діаграма дозволяє підібрати сім'ю розподілів Джонсона за значеннями оцінок асиметрії та ексцесу вибіркового розподілу [80].

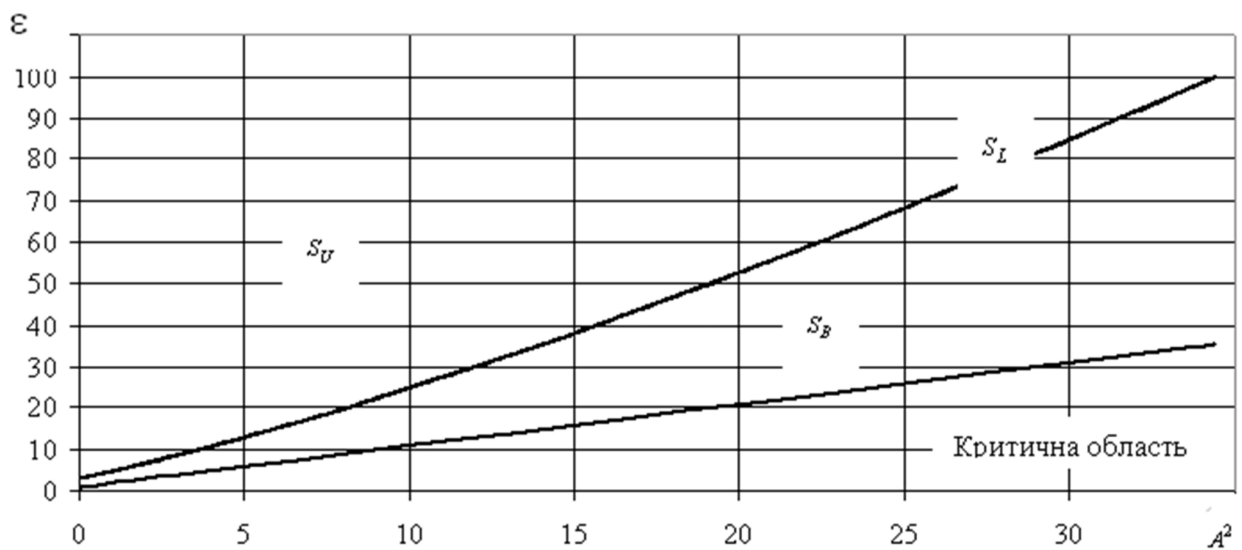


Рисунок 2.1 – Комбінації A^2 і ε для різних розподілів Джонсона [80]

Лінія верхньої границі критичної області задається залежністю

$$\varepsilon = A^2 + 1. \quad (2.14)$$

Якщо комбінація A^2 і ε розташована біля лінії S_L , то вибіркового розподілу можна апроксимувати розподілом із сім'ї S_L (логарифмічно нормальним розподілом). Розподіл зі значеннями A^2 і ε , які лежать вище лінії S_L , апроксимується розподілом із сім'ї S_U , а які лежать нижче лінії S_L (до лінії критичної області) – розподілом із сім'ї S_B . Якщо комбінація A^2 і ε потрапляє в критичну область, то вибіркового розподілу не можна апроксимувати розподілом із сімей Джонсона [80].

Оцінки параметрів перетворення Джонсона визначають за ММП. Один із варіантів визначення оцінок параметрів перетворення є застосування методу мінімізації квадратів асиметрії, ексцесу, середнього та дисперсії

$$\hat{\theta} = \operatorname{argmin} \left\{ A^2 + (\varepsilon - 3)^2 + \bar{Z}^2 + (S_Z^2 - 1)^2 \right\} \quad (2.15)$$

де $\hat{\theta}$ – вектор оцінок параметрів перетворення Джонсона $\hat{\theta} = \{\hat{\gamma}, \hat{\eta}, \hat{\lambda}, \hat{\varphi}\}$, A – асиметрія нормалізованих даних, ε – ексцес нормалізованих даних, $\bar{Z} = \frac{1}{N} \sum_{i=1}^N Z_i$ – математичне сподівання; $S_Z^2 = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z})^2$ – дисперсія, Z_i – i -те значення нормально розподіленої випадкової величини Z розміром N .

Метод максимізації логарифмічної функції правдоподібності є ефективним способом визначення оцінок параметрів перетворення Джонсона. Він має меншу статистичну нестабільність вибірових асиметрії та ексцесу, характерних для метрик програмного коду на відміну від методу заснованому на мінімізації статистичних моментів.

$$\hat{\theta} = \operatorname{argmax} l(\theta), \quad (2.16)$$

де $l(\theta)$ – логарифмічна функція правдоподібності. Для перетворення Джонсона сім'ї SB логарифмічна функція правдоподібності [80] має такий вигляд

$$l(\theta) = N \ln(\eta, \lambda) - \frac{N \ln(2\pi)}{2} - \sum_{i=1}^N \ln(x_i - \varphi) - \sum_{i=1}^N \ln(\varphi + \lambda - x_i) - \frac{1}{2} \sum_{i=1}^N \left[\gamma + \eta \ln \left(\frac{x_i - \varphi}{\varphi + \lambda - x_i} \right) \right]^2. \quad (2.17)$$

Для перетворення Джонсона сім'ї SU логарифмічна функція правдоподібності [80] задається як

$$l(\theta) = N \ln(\eta) - \frac{N \ln(2\pi)}{2} - \frac{1}{2} \sum_{i=1}^N \ln \left[(x_i - \varphi)^2 + \lambda^2 \right] - \frac{1}{2} \sum_{i=1}^N \left[\gamma + \eta \operatorname{Arsh} \left(\frac{x_i - \varphi}{\lambda} \right) \right]^2. \quad (2.18)$$

Надалі для застосування функції (2.17) при побудові моделі замість x_i слід підставляти або Y_i , або X_{ji} .

При оцінювання параметрів $\gamma, \eta, \lambda, \phi$ необхідно врахувати відповідні обмеження, що накладені на ці параметри для обраної сім'ї розподілів Джонсона.

2.3 Багатовимірні взаємозворотні нормалізуючі перетворення

Багатовимірні перетворення застосовуються для одночасної нормалізації взаємопов'язаних змінних багатовимірного набору даних. На відміну від одновимірних нормалізуючих перетворень, які обробляють кожен змінну окремо, багатовимірні трансформації враховують взаємозв'язки між змінними даних та зберігають інформацію про їх спільну варіативність. До взаємозворотних багатовимірних нормалізуючих перетворень належать перетворення Бокса-Кокса та перетворення Джонсона.

Нехай для перетворення негаусівського випадкового вектора $P = \{X_1, X_2, \dots, X_k\}^T$, розміром k , на гаусівський випадковий вектор $T = \{Z_1, Z_2, \dots, Z_k\}^T$ існує взаємозворотнє нормалізуюче перетворення ψ , тоді:

$$T = \psi(P), \quad (2.19)$$

де k – кількість факторів. Обернене перетворення до (2.19) має вигляд

$$P = \psi^{-1}(T), \quad (2.20)$$

де ψ – вектор взаємозворотних функцій нормалізуючого перетворення, $\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$.

У випадку використання багатовимірного перетворення Бокса-Кокса компоненти вектора змінних T багатовимірного нормалізуючого перетворення ψ визначаються наступним чином

$$Z_j = x(\lambda_j) = \begin{cases} (X_j^{\lambda_j} - 1)/\lambda_j, & \text{якщо } \lambda_j \neq 0 \\ \ln(X_j), & \text{якщо } \lambda_j = 0 \end{cases}, \quad (2.21)$$

де, $j = 1, 2, \dots, k$; X_j – випадкова негаусівська величина, яка нормалізується; Z_j – нормалізована гаусівська величина; λ_j – параметр перетворення Бокса-Кокса.

Для знаходження оптимальних оцінок вектора параметрів перетворення θ застосовується ММП. Вибір цього параметричного підходу обумовлений тим, що, на відміну від методів, що засновані на статистиках вищих порядків (зокрема, методу максимізації поліномів), ММП є найбільш ефективним та обґрунтованим для випадку, коли необхідно привести дані до нормального закону розподілу [81].

Для перетворення Бокса-Кокса логарифмічна функція правдоподібності [82] для багатовимірних даних має такий вигляд

$$\hat{\theta} = \arg_{\theta} \max l(P, \theta), \quad (2.22)$$

де $l(P, \theta)$ – логарифмічна функція правдоподібності; θ – вектор параметрів перетворення, $\theta = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$; k – розмірність багатовимірних даних. Для перетворення Бокса-Кокса логарифмічна функція правдоподібності (2.22) для багатовимірних даних задається як

$$l(X, \theta) = \sum_{j=1}^k (\lambda_j - 1) \sum_{i=1}^N \ln(X_{ji}) - \frac{N}{2} \ln[\det(S_N)], \quad (2.23)$$

де N – розмір вибірки, S_N – вибіркова коваріаційна матриця для компонентів вектора Z

$$S_N = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z})(Z_i - \bar{Z})^T \quad (2.24)$$

де Z_i – гаусівський випадковий вектор, $Z_i = \{Z_{1i}, Z_{2i}, \dots, Z_{ki}\}^T$; $\bar{Z}$ – вектор вибірових середніх, $\bar{Z} = \{\bar{Z}_1, \bar{Z}_2, \dots, \bar{Z}_k\}$.

У разі використання перетворення Джонсона сім'ї SB компоненти вектора T визначають так

$$Z_j = \gamma_j + \eta_j \ln\left(\frac{X_j - \varphi_j}{\lambda_j + \varphi_j - X_j}\right), \varphi_j < X_j < \varphi_j + \lambda_j, \quad (2.25)$$

де Z_j – стандартна гаусівська змінна, $Z_j \sim N(0, 1)$; $\gamma_j, \eta_j, \lambda_j, \varphi_j$ – параметри перетворення Джонсона сім'ї SB для j -ї змінної багатовимірної вибірки, $\eta_j > 0$; $-\infty < \gamma_j < \infty$; $\lambda_j > 0$; $-\infty < \varphi_j < \infty$, $j = 1, 2, \dots, k$. Залежна змінна Z_Y визначають аналогічно (2.19) з тією лише відмінністю, що замість $Z_j, X_j, \gamma_j, \eta_j, \lambda_j, \varphi_j$ слід підставити відповідно $Z_Y, X_Y, \gamma_Y, \eta_Y, \lambda_Y, \varphi_Y$.

Оцінювання параметрів багатовимірних перетворень Бокса-Кокса та Джонсона сімей SU, SB та SL можна здійснити за ММП шляхом мінімізації універсальної функції квадратів асиметрії, ексцесу, середнього та дисперсії (2.15). Для багатовимірних даних ця функція має вигляд:

$$\hat{\theta} = \operatorname{argmin} \left\{ A^2 + (\varepsilon - 3)^2 + \frac{1}{k} \sum_{j=1}^k \overline{Z_j^2} + \frac{1}{k} \sum_{i=1}^k (S_{Z_j}^2 - 1)^2 \right\} \quad (2.26)$$

де A – багатовимірна асиметрія, ε – багатовимірний ексцес, середнє значення математичних сподівань $\overline{Z} = \frac{1}{N} \sum_{i=1}^N Z_i$, k – розмірність багатовимірної вибірки, середнє значення дисперсій $S_{Z_j}^2 = \frac{1}{N} \sum_{i=1}^N (Z_{ji} - \overline{Z_j})^2$.

Іншим, не менш ефективним способом для оцінювання параметрів багатовимірного перетворення Джонсона є максимізація логарифмічних функцій правдоподібності (2.22). Для оцінки параметрів багатовимірного перетворення Джонсона сім'ї SB логарифмічна функція правдоподібності має вигляд

$$\begin{aligned} l(\theta) = & N \ln(\eta_Y, \lambda_Y) - N \sum_{j=1}^k \ln(n_j \lambda_j) - \frac{N(k+1)\ln(2\pi)}{2} - \\ & - \frac{N}{2} \ln[\det(S_T)] - \sum_{i=1}^N \ln[(Y_i - \varphi_Y)(\varphi_Y + \lambda_Y - Y_i)] - \\ & - \sum_{j=1}^k \sum_{i=1}^N \ln[(X_{ji} - \varphi_j)(\varphi_j + \lambda_j - X_{ji})] - \frac{1}{2} \sum_{i=1}^N T_i^T S_T^{-1} T_i, \end{aligned} \quad (2.27)$$

де k – розмірність вибірки; $\theta = \{\gamma_Y, \eta_Y, \varphi_Y, \lambda_Y, \gamma_1, \eta_1, \varphi_1, \lambda_1, \dots, \gamma_k, \eta_k, \varphi_k, \lambda_k\}$;

S_T – вибіркова коваріаційна матриця для компонентів вектора T

$$S_T = \frac{1}{N} \sum_{i=1}^N T_i T_i^T. \quad (2.28)$$

В (2.28) враховано те, що математичні сподівання компонентів вектора T для перетворення Джонсона дорівнюють нулю.

2.4 Вибір нормалізуючого перетворення для обробки інформації з багатовимірних даних метрик коду загальних JAVA-застосунків

Проведемо вибір нормалізуючого перетворення для шестивимірних даних навчальної вибірки розміром 286 метрик коду проєктів з відкритим кодом загальних JAVA-застосунків, а KLOC (Y), CLS (X_1), INFC (X_2), aVMQ(X_3), aTFQ (X_4) та aCBO (X_5). Початкові значення метрик навчальної вибірки загальних JAVA-застосунків наведені у додатку Б. Проведемо нормалізацію навчальної вибірки із застосуванням одновимірних та багатовимірних нормалізуючих перетворень. Але перед застосуванням нормалізуючих перетворень перевіримо шестивимірні дані навчальної вибірки на відповідність нормальному розподілу із застосуванням багатовимірного тесту Мардія, який заснований на багатовимірній асиметрії та ексцесі. Згідно з цим тестом виявлено, що розподіл шестивимірних даних CLS, INFC, aVMQ, aTFQ, aCBO і KLOC навчальної вибірки не є гаусівським, оскільки оцінка багатовимірної асиметрії $N\beta_1/6 = 5134,45$ перевищує значення квантиля хі-квадрат (χ^2) 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а оцінка багатовимірного ексцесу $\beta_2 = 219,63$ перевищує значення квантиля розподілу Гауса, яке дорівнює 50,77 для математичного сподівання 48 і дисперсії 1,16. Отже перехід до побудови нелінійної регресії є виправданим.

Наступним кроком, застосуємо до навчальної вибірки даних з додатку Б перетворення десятичного логарифма (2.6). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.1. Відповідно до тесту Мардія, виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 1679,74, перевищує значення квантиля розподілу χ^2 - квадрат, що становить 86,99 для 56 ступенів вільності та рівня значущості 0,005, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 123,48, перевищує значення квантиля нормального розподілу, яке становить 50,77 для математичного сподівання 48 і дисперсії 1,16 та рівня значущості $\alpha = 0,005$.

Таблиця 2.1 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням перетворенням десятичного логарифма

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	1,4776	2,7493	1,6243	0,6869	0,3174	0,7518
Медіана	1,4734	2,7789	1,6721	0,6978	0,3270	0,7753
Середньоквадратичне відхилення	0,5760	0,5407	0,6638	0,1545	0,1972	0,1629
Асиметрія	0,0182	-0,0391	-0,1412	0,2983	-1,0630	-4,3874
Ексцес	-0,4530	-0,3825	-0,2641	0,9130	4,4043	40,1665
Мінімальне значення	0,2003	1,5185	0	0,2318	-0,8770	-0,9266
Максимальне значення	2,8449	4,0257	3,3282	1,3234	0,8018	1,1000

До даних навчальної вибірки (див. додаток Б) застосовано одновимірне перетворення Бокса-Кокса (2.8) із використанням ММП для оцінювання параметрів перетворення (2.11). Характеристики нормалізованої багатовимірної

вибірки наведено в таблиці 2.2. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 224,99, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 68,84, перевищує значення квантиля нормального розподілу, яке становить 50,77 для математичного сподівання 48 і дисперсії 1,16 та рівня значущості $\alpha = 0,005$.

Таблиця 2.2 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням одновимірного перетворення Бокса-Кокса

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	3,3656	6,5946	4,0403	1,3260	0,9265	5,0367
Медіана	3,3609	6,6581	4,1198	1,3534	0,8964	5,0573
Середньоквадратичне відхилення	1,3015	1,3460	1,7376	0,2497	0,6008	1,6758
Асиметрія	0,0012	-0,0021	-0,0054	-0,0147	0,0524	0,0766
Ексцес	-0,4547	-0,3852	-0,2860	0,5577	0,6810	0,9236
Мінімальне значення	0,4606	3,5731	0	0,5036	-1,3265	-0,8712
Максимальне значення	6,4329	9,8207	8,7818	2,2203	2,8786	11,9356

До навчальної вибірки (див. додаток Б) застосовано шестивимірне перетворення Бокса-Кокса (2.21) із використанням ММП для оцінювання параметрів перетворення (2.23). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.3. Відповідно до тесту Мардія, виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу,

оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 219,20, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості 0,005, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 69,01, перевищує значення квантиля нормального розподілу, яке становить 50,77 для математичного сподівання 48 і дисперсії 1,16 та рівня значущості $\alpha = 0,005$.

Таблиця 2.3 – Характеристики розподілу нормалізованої загальної багатовимірної вибірки із використанням шестивимірного перетворення Бокса-Кокса

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	3,4308	6,8799	4,3575	1,2476	0,8852	4,7841
Медіана	3,4173	6,9372	4,3985	1,2749	0,8671	4,8130
Середньоквадратичне відхилення	1,3458	1,4579	1,9686	0,2202	0,5659	1,5552
Асиметрія	0,0312	0,0357	0,1228	-0,1222	-0,1068	0,0170
Ексцес	-0,4512	-0,3843	-0,2595	0,5200	0,8636	0,9466
Мінімальне значення	0,4617	3,6534	0	0,4932	-1,4271	-0,8985
Максимальне значення	6,6426	10,4282	10,0447	1,9972	2,6387	11,0651

Для нормалізації багатовимірних даних з метрик коду загальних JAVA-застосунків навчальної вибірки із використанням перетворень Джонсона обрано сім'ю SB. Вибір цієї сім'ї перетворення зумовлений тим, що значення квадрата асиметрії для змінних KLOC, CLS, INFC, aVMQ, aTFQ та aCBO дорівнюють 10,27, 8,55, 21,27, 5,94, 0,84, та 0,003 відповідно, а значення ексцесу дорівнюють 12,13, 9,94, 29,78, 12,53, 1,49 та 0,92 відповідно. Як можна побачити (рис. 2.1), комбінації значень оцінок квадрата асиметрії та ексцесу для

змінних KLOC, CLS, INFC, aVMQ aTFQ та aCBO потрапляють в область сім'ї SB.

Характеристики нормалізованої багатовимірної вибірки після застосування до навчальної вибірки (див. додаток Б) одновимірного перетворення Джонсона сім'ї SB (2.12) із використанням ММП для оцінювання параметрів перетворення (2.17) наведено в таблиці 2.4. Відповідно до тесту Мардія, виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1N/6$, яка дорівнює 1000,79, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 117,13, перевищує значення квантиля нормального розподілу, яке становить 50,77 для математичного сподівання 48 і дисперсії 1,16 та рівня значущості $\alpha = 0,005$.

Таблиця 2.4 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням одновимірного перетворення Джонсона сім'ї SB

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	-1,3995	-1,3325	-1,6820	-0,7586	-0,5532	-0,4255
Медіана	-1,4330	-1,3343	-1,6644	-0,7595	-0,5716	-0,4175
Середньоквадратичне відхилення	0,5009	0,4825	0,5423	0,1637	0,2065	0,1799
Асиметрія	0,5207	0,4211	0,2138	1,7950	0,6532	-0,5482
Ексцес	0,5121	0,4617	0,3071	10,2724	2,4612	6,0273
Мінімальне значення	-2,3829	-2,3376	-2,8865	-1,1423	-1,2512	-1,4995
Максимальне значення	0,4444	0,4442	0,4442	0,4498	0,4626	0,4536

Далі до навчальної вибірки (див. додаток Б) застосовано шестивимірне перетворення Джонсона сім'ї SB (2.25) із використанням ММП для оцінювання параметрів перетворення (2.27). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.5. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1N/6$, яка дорівнює 1116,92, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 127,65, перевищує значення квантиля нормального розподілу, яке становить 50,77 для математичного сподівання 48 і дисперсії 1,16 та рівня значущості $\alpha = 0,005$.

Таблиця 2.5 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням шестивимірного перетворення Джонсона сім'ї SB

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	-0,0886	0,0209	0,0192	-0,0001	0,0096	0,0030
Медіана	-0,0918	0,0209	0,0195	-0,0001	0,0092	0,0036
Середньоквадратичне відхилення	0,0572	0,0093	0,0069	0,0010	0,0068	0,0103
Асиметрія	0,4563	0,4174	0,1292	1,5873	0,0439	-1,8058
Ексцес	0,4634	0,4603	0,3338	8,7997	3,0596	15,9325
Мінімальне значення	-0,2058	0,0014	0,0028	-0,0026	-0,0238	-0,0803
Максимальне значення	0,1189	0,0553	0,0459	0,0074	0,0392	0,0484

Отже, було виявлено, що жодне з нормалізуючих перетворень не змогло нормалізувати багатовимірні дані інформації з метрик коду загальних

JAVA-застосунків, але нормалізуючі перетворення змогли наблизити розподіл даних до нормального. Найкращим нормалізуючим перетворенням, відповідно до тесту Мардія, виявилось шестивимірне перетворення Бокса-Кокса з оцінкою параметрів за допомогою ММП шляхом максимізації логарифмічної функції правдоподібності. Для подальшої обробки інформації з метрик коду шляхом побудови відповідних багатofакторних нелінійних регресійних моделей обрано шестивимірне перетворення Бокса-Кокса. Важливо зазначити, що обране багатовимірне перетворення враховує кореляцію між змінними багатовимірних даних, що дозволяє краще моделювати складні залежності між ознаками.

2.5 Вибір нормалізуючого перетворення для обробки інформації з багатовимірних даних метрик коду Data Science та Machine Learning JAVA-застосунків

Здійснено вибір нормалізуючого перетворення для шестивимірних даних навчальної вибірки розміром 74 метрики коду проєктів з відкритим кодом DS та ML JAVA-застосунків, а саме KLOC (Y), CLS (X_1), INFC (X_2), aVMQ(X_3), aTFQ (X_4) та aCBO (X_5). Початкові значення метрик навчальної вибірки DS та ML JAVA-застосунків наведені у додатку В. Проведено нормалізацію навчальної вибірки із застосуванням одновимірних та багатовимірних нормалізуючих перетворень. Але перед застосуванням нормалізуючих перетворень перевірено шестивимірні дані навчальної вибірки на відповідність нормальному розподілу із застосуванням багатовимірного тесту Мардія, який заснований на багатовимірній асиметрії та ексцесі. Згідно з цим тестом виявлено, що розподіл шестивимірних даних CLS, INFC, aVMQ, aTFQ, aCBO і KLOC навчальної вибірки не є гаусівським, оскільки оцінка багатовимірної асиметрії $N\beta_1/6 = 718,94$ перевищує значення квантиля $\chi^2 = 86,99$ для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а оцінка багатовимірного

ексцесу $\beta_2 = 112,31$ перевищує значення квантиля розподілу Гауса, яке дорівнює 51,89 для математичного сподівання 48 і дисперсії 2,28. Отже, перехід до побудови нелінійної регресії є виправданим.

Наступним кроком, до навчальної вибірки даних з додатку В застосовано перетворення десяткового логарифма (2.6). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.6. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 633,74, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 95,25, перевищує значення квантиля нормального розподілу, яке становить 51,89 для математичного сподівання 48 і дисперсії 2,28 та рівня значущості $\alpha = 0,005$.

Таблиця 2.6 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням перетворенням десяткового логарифма

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	1,5736	2,7468	1,5440	0,6996	0,3817	0,8037
Медіана	1,6753	2,7931	1,6852	0,6979	0,3735	0,8138
Середньоквадратичне відхилення	0,6796	0,5985	0,7741	0,1776	0,1654	0,1061
Асиметрія	-0,3161	-0,1594	-0,2724	0,2676	-0,1339	-0,4460
Ексцес	-0,8968	-0,6126	-0,5451	1,0960	-0,6891	1,4450
Мінімальне значення	0,2172	1,4771	0	0,2012	-0,0011	0,4608
Максимальне значення	2,7954	4,0504	3,1421	1,2508	0,6626	1,1000

До даних навчальної вибірки (див. додаток В) застосовано одновимірне перетворення Бокса-Кокса (2.8) із використанням ММП для оцінювання параметрів перетворення (2.11). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.7. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 587,59, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 92,80, перевищує значення квантиля нормального розподілу, яке становить 51,89 для математичного сподівання 48 і дисперсії 2,28 та рівня значущості $\alpha = 0,005$.

Таблиця 2.7 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням одновимірного перетворення Бокса-Кокса

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	4,6395	7,5421	4,1465	1,4092	0,9595	2,8418
Медіана	4,7985	7,6236	4,4314	1,4164	0,9240	2,8657
Середньоквадратичне відхилення	2,2951	1,9032	2,2440	0,3145	0,4395	0,5307
Асиметрія	-0,0587	-0,0160	-0,0252	-0,0194	-0,0173	0,0365
Ексцес	-0,9739	-0,6171	-0,5465	1,0511	-0,8042	1,2599
Мінімальне значення	0,5141	3,7171	0	0,4465	-0,0024	1,3410
Максимальне значення	9,3408	11,9704	9,3094	2,3061	1,7349	4,5539

До навчальної вибірки (див. додаток В) застосовано шестивимірне перетворення Бокса-Кокса (2.21) із використанням ММП для оцінювання параметрів перетворення (2.23). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.8. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 625,43, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 95,13, перевищує значення квантиля нормального розподілу, яке становить 51,89 для математичного сподівання 48 і дисперсії 2,28 та рівня значущості $\alpha = 0,005$.

Таблиця 2.8 – Характеристики розподілу нормалізованої загальної багатовимірної вибірки із використанням шестивимірного перетворення Бокса-Кокса

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
Середнє	3,9822	7,1577	4,0307	1,7268	0,9475	2,2101
Медіана	4,1970	7,2492	4,3255	1,7148	0,9146	2,2351
Середньоквадратичне відхилення	1,8143	1,7318	2,1509	0,4678	0,4306	0,3401
Асиметрія	-0,2147	-0,0578	-0,0700	0,4109	-0,0332	-0,2378
Ексцес	-0,9460	-0,6216	-0,5587	1,2117	-0,7906	1,2757
Мінімальне значення	0,5056	3,6214	0	0,4720	-0,0024	1,1711
Максимальне значення	7,4186	11,1112	8,8860	3,2389	1,7032	3,2222

Для нормалізації багатовимірних даних з метрик коду DS та ML JAVA-застосунків навчальної вибірки із використанням перетворень Джонсона

обрано сім'ю SB. Вибір цієї сім'ї перетворення зумовлений тим, що значення квадрата асиметрії для змінних KLOC, CLS, INFC, aVMQ, aTFQ та aCBO дорівнюють 2,06, 3,15, 3,82, 2,17, 0,47 та 0,70 відповідно, а значення ексцесу дорівнюють 5,20, 13,37, 16,76, 7,36, 0,86 та 2,20 відповідно. Як можна побачити (рис. 2.1), комбінації значень оцінок квадрата асиметрії та ексцесу для змінних KLOC, CLS, INFC, aVMQ, aTFQ та aCBO розташовані в області SB.

Характеристики нормалізованої багатовимірної вибірки після застосування до навчальної вибірки (див. додаток В) одновимірного перетворення Джонсона сім'ї SB (2.12) із використанням ММП для оцінювання параметрів перетворення (2.17) наведено в таблиці 2.9. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6$, яка дорівнює 689,37, перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу B_2 , яка становить 104,50, перевищує значення квантиля нормального розподілу, яке становить 51,89 для математичного сподівання 48 і дисперсії 2,28 та рівня значущості $\alpha = 0,005$.

Таблиця 2.9 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням одновимірного перетворення Джонсона сім'ї SB

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
1	2	3	4	5	6	6
Середнє	-0,9796	-1,0825	-1,5163	-0,1208	0,1958	0,3580
Медіана	-0,8906	-1,0688	-1,3389	-0,1457	0,1436	0,3597
Середньоквадратичне відхилення	1,0245	0,8671	1,1609	0,3748	0,4452	0,2659

Продовження табл.2.9

1	2	3	4	5	6	7
Асиметрія	-0,0098	0,3617	-0,0828	1,2557	0,0933	0,6503
Ексцес	-0,3670	0,5701	0,3556	4,7801	-0,5876	3,3643
Мінімальне значення	-2,9961	-2,7752	-3,9292	-1,0538	-0,7975	-0,3529
Максимальне значення	1,8096	1,8192	1,8151	1,5373	1,0760	1,4489

Далі до навчальної вибірки (див. додаток В) застосовано шестивимірне перетворення Джонсона сім'ї SB (2.25) із використанням ММП для оцінювання параметрів перетворення (2.27). Характеристики нормалізованої багатовимірної вибірки наведено в таблиці 2.10. Відповідно до тесту Мардія виявлено, що нормалізовані дані відхиляються від багатовимірного нормального розподілу, оскільки значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6 = 739,77$ перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, а тестова статистика розподілу цієї вибірки для багатовимірного ексцесу $B_2 = 109,50$ перевищує значення квантиля нормального розподілу, яке становить 51,89 для математичного сподівання 48 і дисперсії 2,28 та рівня значущості $\alpha = 0,005$.

Таблиця 2.10 – Характеристики розподілу нормалізованої навчальної шестивимірної вибірки із використанням шестивимірного перетворення Джонсона сім'ї SB

Параметри розподілу вибірки	KLOC	CLS	INFC	aVMQ	aTFQ	aCBO
1	2	3	4	5	6	7
Середнє	-0,1026	0,0022	0,0020	0,0059	0,0001	0,0002
Медіана	-0,0990	0,0022	0,0022	0,0054	0	0,0002
Середньоквадратичне відхилення	0,0537	0,0010	0,0009	0,0051	0,0003	0,0003

Продовження табл.2.10

1	2	3	4	5	6	7
Асиметрія	0,0956	0,3682	0,1859	1,8949	0,6912	1,2885
Ексцес	-0,3704	0,5756	0,2912	7,7882	-0,2254	6,0053
Мінімальне значення	-0,2014	0,0002	0,0004	-0,0049	-0,0004	-0,0004
Максимальне значення	0,0480	0,0056	0,0047	0,0312	0,0008	0,0014

Отже, було виявлено, що жодне з нормалізуючих перетворень не змогло повністю привести багатовимірні дані інформації з метрик коду DS та ML JAVA-застосунків до гаусівського закону, але ці перетворення змогли суттєво наблизити розподіл даних до нормального. Кращим нормалізуючим перетворенням, відповідно до тесту Мардія, виявилось одновимірне перетворення Бокса-Кокса з оцінкою параметрів за допомогою ММП шляхом максимізації логарифмічної функції правдоподібності, але шестивимірне нормалізуюче перетворення Бокса-Кокса має дуже наближені значення оцінок багатовимірної асиметрії $N\beta_1/6$ та ексцесу β_2 до одновимірного перетворення Бокса-Кокса. Для подальшої обробки інформації з метрик коду шляхом побудови відповідних п'ятифакторних нелінійних регресійних моделей обрано одновимірне перетворення Бокса-Кокса, через найменші оцінки багатовимірних асиметрії та ексцесу, і багатовимірне перетворення Бокса-Кокса, оскільки воно враховує кореляцію між змінними багатовимірних даних, що дозволяє краще моделювати складні залежності між ознаками.

2.6 Висновки за розділом 2

У другому розділі розглянуто одновимірні перетворення десяткового логарифма, Бокса-Кокса, Джонсона, багатовимірні перетворення Бокса-Кокса та Джонсона, методи оцінювання їх параметрів, обрано перетворення для нормалізації шестивимірних характеристик інформації із метрик коду загальних, DS та ML JAVA-застосунків.

1) Проведено аналіз найбільш поширених взаємозворотних нормалізуючих перетворень. Показано, що перевагами багатовимірних перетворень є те, що вони дають змогу врахувати кореляцію між змінними багатовимірних даних.

2) Здійснено одноетапну нормалізацію навчальних вибірок інформації з метрик коду загальних JAVA-застосунків, DS та ML JAVA-застосунків за допомогою розглянутих перетворень, та проведено оцінку ефективності нормалізації характеристик за допомогою тесту Мардія для пошуку найкращого нормалізуючого перетворення. Виявлено, що жодне із нормалізуючих перетворень не дає змоги наблизити розподіл багатовимірних даних до нормального, відповідно до цього тесту, за одну ітерацію нормалізації навчальних вибірок.

3) Здійснено вибір перетворення для нормалізації шестивимірних векторів характеристик інформації з метрик коду загальних JAVA-застосунків. У якості нормалізуючого перетворення обрано багатовимірне перетворення Бокса-Кокса, за допомогою якого краще здійснюється нормалізація багатовимірних даних з метрик коду загальних JAVA-застосунків у порівнянні з одновимірними перетвореннями десятичного логарифма, Бокса-Кокса, Джонсона сім'ї SB та багатовимірним перетворенням Джонсона сім'ї SB.

4) Здійснено вибір перетворення для нормалізації шестивимірних векторів характеристик інформації з метрик коду DS та ML JAVA-застосунків. Як нормалізуюче перетворення обрано одновимірне перетворення Бокса-Кокса, яке показало кращу нормалізацію багатовимірних даних з метрик коду DS та ML JAVA-застосунків у порівнянні з одновимірними перетвореннями десятичного логарифма, Джонсона сім'ї SB, багатовимірними перетвореннями Бокса-Кокса та Джонсона сім'ї SB; обрано шестивимірне перетворення Бокса-Кокса, оскільки воно враховує кореляцію між змінними багатовимірних даних.

Результати досліджень цього розділу опубліковано у працях [27-32].

РОЗДІЛ 3. ПОБУДОВА ЧОТИРЬОХ- ТА П'ЯТИФАКТОРНИХ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК КОДУ JAVA-ЗАСТОСУНКІВ

У третьому розділі побудовано чотирьох- та п'ятифакторні нелінійні регресійні моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків, їхні інтервали передбачення та довірчі інтервали на основі п'ятивимірного та шестивимірного перетворень Бокса-Кокса. Здійснено порівняння результатів обробки інформації розміру JAVA-застосунків за отриманими нелінійними регресійними моделями. Побудова чотирьох- та п'ятифакторних нелінійних регресійних моделей відбувалася за допомогою методики, яка ґрунтується на методах математичної статистики та регресійного аналізу. Для визначення оцінок якості прогнозування розміру (кількості рядків коду) JAVA-застосунків використано такі критерії, як коефіцієнти детермінації R^2 та R_{adj}^2 , значення середньої величини відносної похибки $MMRE$ [33] та значення відсотка передбачення для рівня відносної похибки 0,25 – $PRED(0, 25)$ [33].

3.1 Загальний алгоритм побудови нелінійних регресійних моделей на основі взаємозворотних нормалізуючих перетворень

Для побудови нелінійних регресійних моделей було обрано методи, які ґрунтуються на застосуванні взаємозворотних нормалізуючих перетворень з ітеративним видаленням викидів [61, 62]. Використання нормалізуючих перетворень дає змогу перейти до побудови лінійних регресійних моделей на основі нормалізованих даних із подальшим їх зворотним перетворенням на нелінійні регресійні моделі. Відповідно до цих методів загальний алгоритм побудови нелінійних регресійних моделей складається з шести етапів.

На **першому етапі** негаусівські емпіричні дані, які представлені випадковими векторами $P_i = \{Y, X_1, X_2, \dots, X_k\}^T$, перетворюють у гаусівські випадкові вектори $T_i = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ за допомогою визначеного нормалізуючого перетворення

$$T_i = \psi(P_i), \quad (3.1)$$

де k – кількість незалежних факторів; $i = 1, 2, \dots, N$, N – розмір вибірки емпіричних даних. Тоді обернене перетворення до (3.1) має такий вигляд

$$P_i = \psi^{-1}(T_i), \quad (3.2)$$

де ψ – вектор взаємозворотних функцій нормалізуючого перетворення, $\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$.

На **другому етапі** виконується перевірка багатовимірних нормалізованих даних на відповідність нормальному розподілу за допомогою критерію Мардія [72], залежно від параметрів багатовимірної асиметрії ($\beta_{1,k}$) (2.1) та ексцесу ($\beta_{2,k}$) (2.2). Визначення та вилучення викидів із багатовимірних нормалізованих даних здійснюється за допомогою методу на основі квадрата відстані Махаланобіса [83]. Значення квадрата відстані Махаланобіса – це елементи головної діагоналі матриці d^2 , розміром $N \times N$.

$$d^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (3.3)$$

де Z – ВВ з нормальним розподілом, S_N – зміщена вибіркова коваріаційна матриця. Суть цього методу полягає в тому, що значення d_i^2 , $i = 1, 2, \dots, N$, які перевищують значення квантиля розподілу χ^2 для рівня значущості α , вважають викидами і виключаються із вибірки. Зазвичай використовується значення $\alpha = 0,005$ для цього методу. Зміщена вибіркова коваріаційна матриця S_N має такий вигляд:

$$S_N = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z})(Z_i - \bar{Z})^T, \quad (3.4)$$

де Z_i – гаусівський випадковий вектор, $Z_i = \{Z_1, Z_2, \dots, Z_k\}^T$; $\bar{Z}$ – вектор вибірових середніх, $\bar{Z} = \{\bar{Z}_1, \bar{Z}_2, \dots, \bar{Z}_k\}$.

Дані, для яких значення квадрата відстані Махаланобіса більше, ніж квантиль розподілу χ^2 , вважають багатовимірними викидами та відкидають. У такому разі відбувається повернення до першого етапу. Якщо ж викидів немає, здійснюється перехід до наступного етапу.

На **третьому етапі** для нормалізованих даних за перетворенням (3.1), виконується побудова лінійної регресійної моделі [85], яка має вигляд

$$Z_y = \hat{Z}_y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \dots + \hat{b}_k Z_k + \varepsilon, \quad (3.5)$$

де ε – ВВ розподілена за нормальним законом, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, k – кількість незалежних факторів регресії, $\hat{b}_0, \hat{b}_1, \hat{b}_2, \dots, \hat{b}_k$ – оцінки параметрів лінійної регресійної моделі, які знаходяться за методом найменших квадратів.

На **четвертому етапі** залишки регресії ε отриманої лінійної регресійної моделі (3.5) перевіряються на відповідність нормальному закону розподілу. Якщо розподіл залишків лінійної регресійної моделі для нормалізованих даних не є нормальним, тоді необхідно відкинути точку даних, для якої модуль залишку є найбільшим, і повернутись до першого кроку. Перевірка нормальності розподілу залишків регресії здійснюється за критерієм Пірсона χ^2 [84].

На **п'ятому етапі** після побудови моделі лінійної регресії (3.5) для нормалізованих даних, застосовується зворотне перетворення (3.2). В результаті отримуємо нелінійну регресійну модель [85] з k незалежними факторами, яка має такий вигляд

$$Y = \Psi_Y^{-1}(\hat{Z}_Y + \varepsilon) = \Psi_Y^{-1}(\hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \dots + \hat{b}_k Z_k + \varepsilon), \quad (3.6)$$

де Ψ_Y^{-1} – функція зворотного нормалізуючого перетворення оцінки лінійної регресії.

На **шостому етапі** розраховуються верхня та нижня границі інтервалів передбачення та проводиться пошук точок, значення яких виходять за межі цих інтервалів. Якщо такі точки знайдені, то вони вважаються викидами і вилучаються із навчальної вибірки.

Границі інтервалів передбачення нелінійної регресійної моделі визначаються за формулою

$$\hat{Y}_{PI} = \Psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (Z_X^+)^T S_Z^{-1} (Z_X^+) \right\}^{1/2} \right). \quad (3.7)$$

Границі довірного інтервалу визначаються за формулою

$$\hat{Y}_{CI} = \Psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ \frac{1}{N} + (Z_X^+)^T S_Z^{-1} (Z_X^+) \right\}^{1/2} \right), \quad (3.8)$$

де $t_{\alpha/2, v}$ – квантиль t-розподілу Стюдента з $v = N - k - 1$ ступенями

вільності та $\alpha/2$ – рівнем значущості; $S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N \left(Z_{Y_i} - \hat{Z}_{Y_i} \right)^2$; Z_X^+ – вектор

центральных факторів (регресорів), який містить значення

$\left\{ Z_{1_i} - \bar{Z}_1, Z_{2_i} - \bar{Z}_2, \dots, Z_{k_i} - \bar{Z}_k \right\}$; S_Z – $k \times k$ матриця

$$S_Z = \begin{bmatrix} S_{Z_q} & S_{Z_r} \end{bmatrix}, \quad (3.9)$$

де $S_{X_q} S_{X_r} = \sum_{n=1}^N \left(Z_{q_i} - \bar{Z}_q \right) \left(Z_{r_i} - \bar{Z}_r \right)$, $q = 1, 2, \dots, k$ та $r = 1, 2, \dots, k$.

3.2 Побудова чотирьохфакторної нелінійної регресійної моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та її інтервалів передбачення на основі п'ятивимірного нормалізуючого перетворення Бокса-Кокса

В розділі 1 дисертаційної роботи була розглянута чотирьохфакторна нелінійна регресійна модель [26] для обробки інформації із метрик коду для оцінювання параметра розміру загальних JAVA-застосунків. Однак на етапі аналізу виявлено, що ця модель побудована на вибірці обмеженого розміру, відсутня валідація моделі за допомогою тестового набору даних, а навчальна вибірка є застарілою для оцінювання параметра кількості рядків коду сучасних JAVA-застосунків. Оцінки якості R^2 , $MMRE$ та $PRED(0, 25)$ для моделі [26] не дали значущих результатів [27, 29]. Тому у дисертаційній роботі було прийнято рішення щодо вдосконалення моделі [26] шляхом розширення обсягу початкових даних, використання тестової вибірки, набору інших метрик коду та застосування нормалізуючого перетворення Бокса-Кокса для побудови нової моделі.

Побудуємо чотирьохфакторну нелінійну регресійну модель на основі навчальної вибірки даних (див. додаток Б) за метриками коду CLASS (X_1), RFC (X_2), аCBO (X_3) та аVMQ (X_4) [35]. Для вдосконалення існуючої чотирьохфакторної нелінійної регресійної моделі [26] було прийнято рішення метрику NOSM замінити на метрику аVMQ, оскільки ця метрика краще представляє визначені методи в проєкті на відміну від метрики NOSM, яка враховує тільки статичні методи. Метрику LCOM замінено на метрику аCBO, яка характеризує загальну зв'язність між компонентами програмного проєкту. Слід зазначити, що метрика LCOM вимірює зв'язність методів та атрибутів в класі, але на етапі проєктування ПЗ цю інформацію немає можливості отримати з UML-діаграми класів [69].

Вибір параметрів для удосконалення чотирьохфакторної нелінійної регресійної моделі для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків проведено з урахуванням впливу мультиколінеарності. Для факторів X_1 , X_2 , X_3 та X_4 значення коефіцієнтів VIFs дорівнюють 6,3, 6,5, 1,3 та 1,2 відповідно, що свідчить про відсутність мультиколінеарності між факторами регресійної моделі на початковому етапі обробки даних у порівнянні з абсолютними значеннями VIFs для метрик CLASS, RFC, CBO та VMQ, які мають значення 14,2, 15,9, 21,9 та 17,4 відповідно.

Багатовимірні дані з метрик коду загальних JAVA-застосунків перевірено на відповідність нормальному закону розподілу за критерієм Мардія для рівня значущості $\alpha = 0,005$. Результати перевірки показали, що розподіл п'ятивимірних даних є негаусівським, оскільки тестова статистика для багатовимірної асиметрії $N \cdot \beta_1 / 6 = 12707,58$ перевищує значення 60,27 квантиля розподілу χ^2 для 35 ступенів вільності, та значення багатовимірного ексцесу $\beta_2 = 399,58$ більше за значення квантиля розподілу Гауса, яке становить 37,55, де $\mu = 35$ та $\sigma^2 = 0,98$. Отже, перехід до побудови нелінійної регресійної моделі є виправданим.

Негаусівські дані навчальної вибірки перетворено в гаусівські шляхом застосування п'ятивимірного нормалізуючого перетворення Бокса-Кокса. В процесі нормалізації даних визначено та ітеративно вилучено із навчальної вибірки 25 проєктів за критерієм квадрата відстані Махаланобіса. Для останньої ітерації п'ятивимірні нормалізовані дані перевірено за тестом Мардія на відповідність нормальному закону розподілу для рівня значущості $\alpha = 0,005$. Результати перевірки показали, що розподіл багатовимірних даних Z_1, Z_2, Z_3, Z_4 та Z_Y є нормальним, оскільки тестова статистика для багатовимірної асиметрії $N \cdot \beta_1 / 6 = 56,36$ не перевищує значення 60,27 квантиля розподілу χ^2 для 35

ступенів вільності, а значення багатовимірного ексцесу $\beta_2 = 35,29$ менше за значення квантиля розподілу Гауса, яке становить 37,70 для $\mu = 35$ та $\sigma^2 = 1,1$. Оцінки параметрів п'ятивимірного нормалізуючого перетворення Бокса-Кокса (2.21) отримано за допомогою ММП шляхом максимізації логарифмічної функції правдоподібності (2.23). Значення параметрів перетворення становлять $\hat{\lambda}_Y = -0,0502$, $\hat{\lambda}_{X_1} = 0,0147$, $\hat{\lambda}_{X_2} = -0,0286$, $\hat{\lambda}_{X_3} = 0,6506$ та $\hat{\lambda}_{X_4} = -0,02$ для залежного фактора Y та незалежних факторів X_1, X_2, X_3 та X_4 .

Для нормалізованих даних побудовано лінійну регресійну модель (3.5) з кількістю факторів $k = 4$. Оцінки параметрів чотирьохфакторної лінійної регресійної моделі на основі нормалізованих даних, отримані за методом найменших квадратів, становлять $\hat{b}_0 = -4,7268$, $\hat{b}_1 = 0,2389$, $\hat{b}_2 = 0,7564$, $\hat{b}_3 = -0,0899$ та $\hat{b}_4 = 0,4151$. Отримана чотирьохфакторна лінійна регресійна модель має наступний вигляд:

$$\begin{aligned} \hat{Z}_{Y_{4x}} + \varepsilon = & \hat{b}_0 + \hat{b}_1 \psi_1(X_1) + \hat{b}_2 \psi_2(X_2) + \\ & + \hat{b}_3 \psi_3(X_3) + \hat{b}_4 \psi_4(X_4) + \varepsilon, \end{aligned} \quad (3.10)$$

де $\psi_j(X_j) = (X_j^{\hat{\lambda}_{X_j}} - 1) / \hat{\lambda}_{X_j}$, – функція вектора п'ятивимірного нормалізуючого перетворення Бокса-Кокса із значенням відповідної оцінки параметра перетворення для незалежного фактора X_j ($j = 1, \dots, 4$).

Для побудованої чотирьохфакторної лінійної регресійної моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків (3.10) залишки регресії ε перевірено на відповідність нормальному закону розподілу за критерієм Пірсона χ^2 для рівня значущості $\alpha = 0,01$. В результаті ітераційного процесу побудови математичної моделі викиди не було виявлено.

Після застосування зворотного нормалізуючого перетворення (3.2) до отриманої чотирьохфакторної лінійної регресійної моделі обробки інформації з

метрик коду для оцінювання розміру загальних JAVA-застосунків (3.10) удосконалена чотирьохфакторна нелінійна регресійна модель має такий вигляд

$$Y_{4x} = \psi_Y^{-1}(\hat{Z}_{Y_{4x}} + \varepsilon) = \left[\hat{\lambda}_Y \left(\hat{Z}_{Y_{4x}} + \varepsilon \right) + 1 \right]^{\frac{1}{\hat{\lambda}_Y}}. \quad (3.11)$$

Для удосконаленої чотирьохфакторної нелінійної регресійної моделі (3.11) у дисертаційному дослідженні визначено рівняння верхньої та нижньої границь інтервалів передбачення та довірчих інтервалів для рівня значущості $\alpha = 0,01$, які мають такий вигляд:

$$\begin{aligned} \hat{Y}_{PI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{4x}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ 1 + \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \\ \hat{Y}_{CI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{4x}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \end{aligned} \quad (3.12)$$

де $\hat{Z}_{Y_{4x}}$ – оцінка параметра розміру для нормалізованих даних за (3.10).

На цьому етапі виявлено 6 викидів, які ітеративно вилучено із нормалізованої навчальної вибірки. Для останньої ітерації значення нормалізованих вибірових середніх $\bar{Z}_1, \bar{Z}_2, \bar{Z}_3$ та $\bar{Z}_4$ становлять 6,7411, 7,7479, 3,3751 та 1,5306 відповідно. Квантиль t -розподілу Стюдента $t_{\alpha/2, v} = 2,5946$ для рівня значущості $\alpha = 0,01$ та 250 ступенів вільності; $S_{Z_Y} = 0,1378$; $N = 255$. Обернена коваріаційна матриця (3.9) має такий вигляд

$$S_Z^{-1} = \begin{vmatrix} 0,0758 & -0,1071 & 0,0195 & 0,0430 \\ -0,1071 & 0,1556 & -0,0285 & -0,0645 \\ 0,0195 & -0,0285 & 0,0121 & 0,0058 \\ 0,0430 & -0,0645 & 0,0058 & 0,0706 \end{vmatrix}$$

Побудову нелінійної регресійної моделі (3.11) виконано за 32 ітерації, з багатовимірних нормалізованих даних вилучено 31 викид.

Отриману вдосконалену чотирьохфакторну нелінійну регресійну модель (3.11) для обробки інформації з метрик коду для оцінювання розміру загальних

JAVA-застосунків перевірено за допомогою критеріїв якості регресійних моделей R^2 , $MMRE$ та $PRED(0, 25)$ (див. табл. 3.1).

Таблиця 3.1 – Критерії якості вдосконаленої чотирьохфакторної нелінійної регресійної моделі.

	R^2	$MMRE$	$PRED(0, 25)$
Навчальна вибірка	0,9407	0,1311	0,8745
Тестова вибірка	0,8981	0,1536	0,8211

Наведені в таблиці 3.1 результати свідчать про високий рівень достовірності обробки інформації для оцінювання параметра розміру загальних JAVA-застосунків за допомогою отриманої чотирьохфакторної моделі (3.11).

3.3 Побудова п'ятифакторної нелінійної регресійної моделі та інтервалів її передбачення обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків на основі шестивимірною нормалізуючого перетворення Бокса-Кокса

Незважаючи на високий рівень достовірності обробки інформації для оцінювання параметра розміру загальних JAVA-застосунків за допомогою удосконаленої чотирьохфакторної нелінійної регресійної моделі (3.11), вона має певний недолік. Під час її побудови використано метрику RFC, яку неможливо достовірно визначити на ранньому етапі проєктування програмної системи, що безпосередньо впливає на подальшу достовірність обробки інформації для оцінювання розміру та трудомісткості на ранніх етапах проєктування програмних застосунків. Недоліком цієї метрики є те, що необхідно мати інформацію про унікальні виклики всіх залежних методів незалежно від модифікатора доступу до методу (public, protected, private). Однак UML-діаграма класів не надає інформацію про внутрішні виклики методів,

послідовність методів та їх алгоритми, яка необхідна для розрахунку метрики RFC. Необхідні дані для метрики RFC можна отримати, використовуючи ще й UML-діаграму послідовності. Це значно підвищує деталізацію архітектури ПЗ та час переходу від стадії планування до стадії розробки [69].

Проведено дослідження і висунуто гіпотезу, що існує інший набір параметрів (метрик коду, інформацію про які можна отримати тільки на ранніх стадіях проєктування з UML-діаграми класів) для побудови нелінійної регресійної моделі, яка дасть змогу підвищити достовірність обробки інформації з метрик коду для оцінювання параметра розміру із застосуванням багатовимірних нормалізуючих перетворень. Запропоновано побудувати п'ятифакторну нелінійну регресійну модель обробки інформації з метрик коду, що допоможе отримати кращі показники якості в порівнянні з чотирьохфакторною моделлю та зменшити відповідні інтервали прогнозування та передбачення.

Для побудови п'ятифакторної нелінійної регресійної моделі обробки інформації з метрик коду для оцінювання розміру KLOC (Y) обрано фактори CLS (X_1), INFC (X_2), aVMQ (X_3), aTFQ (X_4) та aCBO (X_5). Доцільність такого вибору обґрунтовано в дисертаційному дослідженні в пп. 1.2.

Багатовимірні дані з метрик коду JAVA-застосунків з відкритим кодом перевірено на відповідність нормальному закону розподілу за критерієм Мардія [72] для рівня значущості $\alpha = 0,005$ у розділі 2.3. Результати перевірки показали, що розподіл шестивимірних даних Y , X_1 , X_2 , X_3 , X_4 та X_5 не є гаусівським, а отже, перехід до побудови нелінійної регресійної моделі є виправданим.

Негаусівські дані навчальної вибірки (див. додаток Б) перетворено на гаусівські із застосуванням шестивимірного нормалізуючого перетворення Бокса-Кокса. При перевірці нормалізованих багатовимірних даних за тестом Мардія для 56 ступенів вільності та рівня значущості $\alpha = 0,005$ було виявлено, що розподіл даних не є нормальним. Далі всі значення навчальної вибірки

перевірено відповідно до алгоритму, що реалізує тест Мардія, та квадрата відстані Махаланобіса (3.3) з метою вилучення викидів. У результаті виявлено 23 викиди, які наведено в таблиці 3.2.

Таблиця 3.2 – Ітеративне вилучення викидів з навчальної вибірки проєктів за тестом Мардія та квадратом відстані Махаланобіса.

Ітерація	$\beta_1 N/6$	χ^2	β_2	$\mathcal{N}_{1-\alpha}(\mu, \sigma^2)$	σ^2	Ідентифікатор викиду
1	219,20	86,99	69,02	50,77	1,16	134
2	210,16	86,99	65,77	50,78	1,16	284
3	210,01	86,99	63,32	50,78	1,16	11
4	179,00	86,99	61,11	50,78	1,16	243
5	144,57	86,99	58,10	50,78	1,17	101
6	123,06	86,99	55,77	50,78	1,17	84
7	126,74	86,99	53,57	50,79	1,17	260
8	134,77	86,99	51,76	50,79	1,17	54
9	137,36	86,99	50,91	50,79	1,18	223
10	132,43	86,99	50,67	50,79	1,18	252
11	130,07	86,99	50,36	50,80	1,18	182
12	127,02	86,99	49,88	50,80	1,18	225
13	127,36	86,99	49,63	50,80	1,18	88
14	120,40	86,99	49,36	50,81	1,19	270
15	117,52	86,99	49,06	50,81	1,19	53
16	116,19	86,99	48,76	50,81	1,19	71
17	110,78	86,99	48,50	50,81	1,19	236
18	108,66	86,99	48,26	50,82	1,19	173
19	103,11	86,99	48,12	50,82	1,20	187
20	99,74	86,99	48,06	50,82	1,20	168
21	95,18	86,99	47,93	50,82	1,20	241
22	89,24	86,99	47,78	50,83	1,20	40
25	86,89	86,99	47,65	50,83	1,21	126

Далі за методом ММП (2.23) були визначені оцінки параметрів багатовимірного нормалізуючого перетворення Бокса-Кокса (2.21). На останній ітерації, для нормалізації залежного фактора Y та незалежних факторів X_1, X_2, X_3, X_4 та X_5 , ці оцінки мають такі значення:

$$\hat{\lambda}_Y = 1,6297 \cdot 10^{-2}, \hat{\lambda}_{X_1} = 4,3651 \cdot 10^{-2}, \hat{\lambda}_{X_2} = 0,1028,$$

$$\hat{\lambda}_{X_3} = -0,2212, \hat{\lambda}_{X_4} = 0,3545, \hat{\lambda}_{X_5} = 0,8947.$$

Слід зазначити, що величина багатовимірної асиметрії $\beta_1 N/6$, яка дорівнює 81,82, не перевищує значення квантиля розподілу χ^2 , що становить 86,99 для 56 ступенів вільності та рівня значущості 0,005. Величина тестової статистики розподілу вибірки для багатовимірного ексцесу $\beta_2 = 47,60$ не перевищує значення квантиля нормального розподілу, яке становить 50,84 для $\mu = 48$ і $\sigma^2 = 1,21$ та рівня значущості $\alpha = 0,005$. Це свідчить про нормальну природу розподілу нормалізованих даних, з яких були вилучені викиди. За одну ітерацію вилучався один викид, відповідно до наведеної методики (див. п.3.1) побудови нелінійних регресійних моделей.

Далі для нормалізованих даних побудовано лінійну регресійну модель (3.5) з кількістю факторів $k = 5$. Оцінки параметрів п'ятифакторної лінійної регресійної моделі на основі нормалізованих даних, отримані за методом найменших квадратів, становлять $\hat{b}_0 = -4,1177$, $\hat{b}_1 = 0,8097$, $\hat{b}_2 = -4,5997 \cdot 10^{-3}$, $\hat{b}_3 = 1,1754$, $\hat{b}_4 = 0,2523$ та $\hat{b}_5 = -1,6086 \cdot 10^{-2}$.

П'ятифакторна лінійна регресійна модель для нормалізованих даних на основі загальної вибірки (general) JAVA-застосунків має такий вигляд:

$$\begin{aligned} \hat{Z}_{Y_{gen}} + \varepsilon = & \hat{b}_0 + \hat{b}_1 \psi_1(X_1) + \hat{b}_2 \psi_2(X_2 + 1) + \\ & + \hat{b}_3 \psi_3(X_3) + \hat{b}_4 \psi_4(X_4) + \hat{b}_5 \psi_5(X_5) + \varepsilon, \end{aligned} \quad (3.13)$$

де $\psi_j(X_j) = (X_j^{\hat{\lambda}_{X_j}} - 1)/\hat{\lambda}_{X_j}$ – нормалізуюче перетворення Бокса-Кокса із значенням оцінок перетворення для незалежного фактору X_j ($j = 1, \dots, 5$).

Для побудованої лінійної регресійної моделі (3.13) залишки регресії ε перевірено на відповідність нормальному закону розподілу за критерієм Пірсона χ^2 для рівня значущості $\alpha = 0,01$. На цьому етапі викиди не виявлено, для останньої ітерації значення критерію Пірсона становить 10,79, що менше за $\chi^2_{кр} = 15,09$ для 5 ступенів вільності (для кількості інтервалів залишків регресії 8, яку розраховано за формулою Старджеса [80]).

Після застосування зворотного нормалізуючого перетворення (3.2) до лінійної регресійної моделі (3.13), п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків набуває такого вигляду

$$Y_{gen} = \psi_Y^{-1}(\hat{Z}_{Y_{gen}} + \varepsilon) = \left[\hat{\lambda}_Y \left(\hat{Z}_{Y_{gen}} + \varepsilon \right) + 1 \right]^{\frac{1}{\hat{\lambda}_Y}}. \quad (3.14)$$

Для отриманої нелінійної регресійної моделі (3.14) визначено верхню та нижню границі інтервалів передбачення для рівня значущості $\alpha = 0,01$ (3.8) та виявлено 2 багатовимірні точки-викиди, які ітеративно вилучено із нормалізованої навчальної вибірки та мають ідентифікатори 126 та 222.

Інтервал передбачення та довірчий інтервал нелінійної регресійної моделі Y_{gen} (3.14) мають такий вигляд

$$\begin{aligned} \hat{Y}_{PI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{gen}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ 1 + \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \\ \hat{Y}_{CI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{gen}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \end{aligned} \quad (3.15)$$

де $\hat{Z}_{Y_{gen}}$ – оцінка параметра розміру для нормалізованих даних за (3.13).

Для останньої ітерації значення нормалізованих вибірових середніх становлять: $\bar{Z}_1 = 7,4116$, $\bar{Z}_2 = 4,7788$, $\bar{Z}_3 = 1,3130$, $\bar{Z}_4 = 0,8822$, $\bar{Z}_5 = 4,4300$. Квантиль t -розподілу Стюдента $t_{\alpha/2,v} = 2,5952$ для рівня значущості $\alpha = 0,01$ та 255 ступенів вільності; $\hat{S}_{Z_y} = 0,1716$, $N = 261$.

Обернена коваріаційна матриця (3.9) має такий вигляд

$$\hat{S}_Z^{-1} = \begin{vmatrix} 8,0315 \cdot 10^{-3} & -5,2507 \cdot 10^{-3} & -2,5413 \cdot 10^{-3} & -1,0127 \cdot 10^{-3} & -5,4032 \cdot 10^{-4} \\ -5,2507 \cdot 10^{-3} & 4,2867 \cdot 10^{-3} & 3,4181 \cdot 10^{-4} & 1,2176 \cdot 10^{-3} & 3,2418 \cdot 10^{-4} \\ -2,5413 \cdot 10^{-3} & 3,4181 \cdot 10^{-4} & 1,0058 \cdot 10^{-1} & -1,2670 \cdot 10^{-2} & -5,2167 \cdot 10^{-3} \\ -1,0127 \cdot 10^{-3} & 1,2176 \cdot 10^{-3} & -1,2670 \cdot 10^{-2} & 1,9085 \cdot 10^{-2} & -2,0527 \cdot 10^{-3} \\ -5,4032 \cdot 10^{-4} & 3,2418 \cdot 10^{-4} & -5,2167 \cdot 10^{-3} & -2,0527 \cdot 10^{-3} & 3,3910 \cdot 10^{-3} \end{vmatrix}$$

Для отримання менших за шириною інтервалів передбачення використано квантиль t -розподілу Стюдента $t_{\alpha/2,v} = 1,9693$ з рівнем значущості $\alpha = 0,05$ та 255 ступенями вільності.

П'ятифакторна нелінійна регресійна модель (3.14) побудована за 26 ітерацій. З багатовимірних нормалізованих даних було вилучено 25 багатовимірних викидів, з яких 23 викиди вилучено за квадратом відстані Махаланобіса і 2 були за межами інтервалів передбачення. Побудована п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання KLOC загальних JAVA-застосунків має такі обмеження значень незалежних факторів: $X_1 \in [33; 10610]$, $X_2 \in [0; 1212]$, $X_3 \in [2, 3391; 12, 1402]$, $X_4 \in [0, 5385; 5, 6370]$ та $X_5 \in [2, 1046; 10, 4337]$. На останній ітерації для факторів CLS, INFC, aVMQ, aTFQ і aCBO розраховано значення VIFs, що становлять 3,95, 3,96, 1,22, 1,23 і 1,26 відповідно. Отримані значення VIFs показують відсутність мультиколінеарності між незалежними змінними моделі. На останньому етапі модель (3.14) було оцінено за критеріями якості нелінійних регресійних моделей, які наведені в таблиці 3.3.

Таблиця 3.3 – Значення критеріїв якості для моделі (3.14)

	R^2	R^2_{adj}	$MMRE$	$PRED(0, 25)$
Навчальна вибірка	0,9647	0,9641	0,1305	0,8927
Тестова вибірка	0,9193	-	0,1742	0,7825

Наведені результати показують високий рівень якості побудованої п'ятифакторної нелінійної регресійної моделі для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків.

3.4 Побудова п'ятифакторної нелінійної регресійної моделі та інтервалів її передбачення обробки інформації з метрик коду для оцінювання розміру Data Science та Machine Learning JAVA-застосунків на основі шестивимірного нормалізуючого перетворення Бокса-Кокса

Для побудови п'ятифакторної нелінійної регресійної моделі на основі шестивимірного перетворення Бокса-Кокса обробки інформації з метрик коду для оцінювання розміру KLOC (Y) DS та ML JAVA-застосунків обрано фактори CLS (X_1), INFC (X_2), aVMQ (X_3), aTFQ (X_4) та aCBO (X_5). Доцільність такого вибору обґрунтовано в дисертаційному дослідженні в підрозділі 1.3. Вихідними даними для побудови моделі є навчальна вибірка із значеннями незалежних факторів, які наведено в додатку В дисертації.

Багатовимірні дані з метрик коду DS та ML JAVA-застосунків з відкритим кодом перевірено на відповідність нормальному закону розподілу за критерієм Мардія [72] для рівня значущості $\alpha = 0,005$ у підрозділі 2.4. Результати перевірки показали, що розподіл шестивимірних даних Y , X_1 , X_2 , X_3 , X_4 та X_5 не є гаусівським, а отже, перехід до побудови нелінійної регресійної моделі є виправданим.

Негаусівські дані навчальної вибірки (див. додаток В) перетворено на гаусівські із застосуванням шестивимірною нормалізуючого перетворення Бокса-Кокса. Під час перевірки нормалізованих багатовимірних даних за тестом Мардія для 56 ступенів вільності та рівня значущості $\alpha = 0,005$ було виявлено, що розподіл даних не є нормальним. Далі, з метою видалення викидів, всі значення навчальної вибірки були перевірені відповідно до алгоритму, що реалізує тест Мардія та квадрата відстані Махаланобіса (3.3). Для першої ітерації оцінка багатовимірної асиметрії $N\beta_1/6 = 625,42$ перевищує значення квантиля $\chi^2_{\text{кр}} = 86,99$ для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, оцінка багатовимірною ексцесу $\beta_2 = 95,13$ перевищує значення квантиля розподілу Гауса $\mathcal{N}_{1-\alpha}(\mu, \sigma^2) = 51,89$ для $\mu = 48$ та $\sigma^2 = 2,28$. На цій ітерації видалено точку під номером 37 як багатовимірний викид, оскільки значення квадрата відстані Махаланобіса становило 62,60, що перевищило значення квантиля розподілу $\chi^2_{\text{кр}} = 14,86$ для рівня значущості $\alpha = 0,005$. На 2-й та 3-й ітераціях видалено точки під номерами 68 та 44, оскільки значення квадрата відстані Махаланобіса становили 23,77 та 20,03 відповідно, що також перевищило значення квантиля розподілу $\chi^2_{\text{кр}} = 14,86$.

Оцінки параметрів багатовимірною нормалізуючого перетворення (2.21) залежного фактора Y та незалежних факторів X_1, X_2, X_3, X_4 та X_5 мають такі значення на останній ітерації:

$$\begin{aligned}\hat{\lambda}_Y &= 0,1042, \hat{\lambda}_{X_1} = 9,1517 \cdot 10^{-2}, \hat{\lambda}_{X_2} = 0,1038, \\ \hat{\lambda}_{X_3} &= 0,5131, \hat{\lambda}_{X_4} = -7,1949 \cdot 10^{-3}, \hat{\lambda}_{X_5} = 0,1688.\end{aligned}$$

Оцінки параметрів перетворення отримано за ММП для багатовимірних даних шляхом максимізації багатовимірною логарифмічною функцією правдоподібності (2.23). Для останньої ітерації значення тестової статистики розподілу для багатовимірною асиметрії $B_1 N/6 = 71,6$ не перевищує значення

квантиля розподілу $\chi^2_{\text{кр}} = 86,99$ для 56 ступенів вільності та рівня значущості 0,005. Тестова статистика розподілу цієї вибірки для багатовимірного ексцесу $B_2 = 50,23$ не перевищує значення квантиля розподілу Гауса $\mathcal{N}_{1-\alpha}(\mu, \sigma^2) = 52,01$ для $\mu = 48$ та $\sigma^2 = 2,43$ і рівня значущості $\alpha = 0,005$. Це свідчить про нормальну природу розподілу нормалізованих даних без викидів. Слід зауважити, що на кожній ітерації вилучали по одному викиду.

Далі для нормалізованих даних побудовано лінійну регресійну модель (3.5) з кількістю незалежних факторів $k = 5$. Оцінки параметрів п'ятифакторної лінійної регресійної моделі на основі нормалізованих даних для останньої ітерації, отримані за методом найменших квадратів, становлять $\hat{b}_0 = -4,1016$, $\hat{b}_1 = 0,7929$, $\hat{b}_2 = 2,7031 \cdot 10^{-3}$, $\hat{b}_3 = 0,5088$, $\hat{b}_4 = 0,4588$ та $\hat{b}_5 = -3,2583 \cdot 10^{-2}$.

П'ятифакторна лінійна регресійна модель для нормалізованих даних DS та ML JAVA-застосунків має такий вигляд:

$$\begin{aligned} \hat{Z}_{Y_{DSML}} + \varepsilon = & \hat{b}_0 + \hat{b}_1 \psi_1(X_1) + \hat{b}_2 \psi_2(X_2 + 1) + \\ & + \hat{b}_3 \psi_3(X_3) + \hat{b}_4 \psi_4(X_4) + \hat{b}_5 \psi_5(X_5) + \varepsilon \end{aligned} \quad (3.16)$$

де $\psi_j(X_j) = (X_j^{\hat{\lambda}_{X_j}} - 1) / \hat{\lambda}_{X_j}$ – функція нормалізуючого багатовимірного перетворення Бокса-Кокса ψ із значенням оцінки перетворення $\hat{\lambda}_{X_j}$ для незалежного фактору X_j ($j = 1, \dots, 5$).

Для побудованої п'ятифакторної лінійної регресійної моделі залишки регресії ε перевірено на відповідність нормальному закону розподілу за критерієм Пірсона χ^2 . На цьому етапі виявлено і вилучено 5 проєктів на ітераціях 4, 6, 7, 8 та 9 під номерами 16, 26, 23, 40 та 34. Відповідно значення критеріїв Пірсона $\chi^2_{16} = 13,52$, $\chi^2_{26} = 12,43$, $\chi^2_{23} = 12,24$, $\chi^2_{40} = 13,39$ та

$\chi^2_{34} = 10,73$ перевищують порогове допустиме значення $\chi^2_{кр} = 9,49$ для рівня значущості $\alpha = 0,05$ та 4 ступенів вільності.

Після застосування зворотного нормалізуючого перетворення (3.2) до лінійної регресійної моделі (3.16), п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків на базі шестивимірною перетворення Бокса-Кокса має такий вигляд

$$Y_{DSML} = \Psi_Y^{-1}(\hat{Z}_{Y_{DSML}} + \varepsilon) = \left[\hat{\lambda}_Y \left(\hat{Z}_{Y_{DSML}} + \varepsilon \right) + 1 \right]^{\frac{1}{\hat{\lambda}_Y}}. \quad (3.17)$$

Для отриманої нелінійної регресійної моделі (3.17) визначено верхню та нижню границі інтервалів передбачення (3.7) та довірчих інтервалів (3.8) для рівня значущості $\alpha = 0,05$ та виявлено 1 багатовимірний викид з ідентифікатором 8, який був вилучений із нормалізованої навчальної вибірки.

Інтервал передбачення та довірчий інтервал нелінійної регресійної моделі Y_{DSML} (3.17) мають такий вигляд

$$\begin{aligned} \hat{Y}_{PI} &= \Psi_Y^{-1} \left(\hat{Z}_{Y_{DSML}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ 1 + \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \\ \hat{Y}_{CI} &= \Psi_Y^{-1} \left(\hat{Z}_{Y_{DSML}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ \frac{1}{N} + \left(Z_X^+ \right)^T \hat{S}_Z^{-1} \left(Z_X^+ \right) \right\}^{1/2} \right), \end{aligned} \quad (3.18)$$

де $\hat{Z}_{Y_{DSML}}$ – оцінка параметра розміру для нормалізованих даних за математичною моделлю (3.16).

Для останньої ітерації значення нормалізованих вибірових середніх становлять: $\bar{Z}_1 = 8,6944$, $\bar{Z}_2 = 4,4852$, $\bar{Z}_3 = 2,5272$, $\bar{Z}_4 = 0,8753$, $\bar{Z}_5 = 2,1896$. Квантиль t -розподілу Стюдента становить $t_{\alpha/2, v} = 2,001$ для рівня значущості $\alpha = 0,05$ та 59 ступенів вільності; $\hat{S}_{Z_Y} = 0,2233$; $N = 65$.

Обернена коваріаційна матриця (3.9) має вигляд

$$S_Z^{-1} = \begin{vmatrix} 1,4491 \cdot 10^{-2} & -1,1869 \cdot 10^{-2} & -2,5827 \cdot 10^{-3} & -8,3694 \cdot 10^{-3} & -8,2195 \cdot 10^{-3} \\ -1,1869 \cdot 10^{-2} & 1,2722 \cdot 10^{-2} & -1,3732 \cdot 10^{-3} & 1,0439 \cdot 10^{-2} & 4,1455 \cdot 10^{-3} \\ -2,5827 \cdot 10^{-3} & -1,3732 \cdot 10^{-3} & 3,4815 \cdot 10^{-2} & -2,6659 \cdot 10^{-2} & -1,6817 \cdot 10^{-2} \\ -8,3694 \cdot 10^{-3} & 1,0439 \cdot 10^{-2} & -2,6659 \cdot 10^{-2} & 1,429 \cdot 10^{-1} & -1,4449 \cdot 10^{-2} \\ -8,2195 \cdot 10^{-3} & 4,1455 \cdot 10^{-3} & -1,6817 \cdot 10^{-2} & -1,4449 \cdot 10^{-2} & 1,8016 \cdot 10^{-1} \end{vmatrix}$$

П'ятифакторна нелінійна регресійна модель (3.17) побудована за 10 ітерацій. З багатовимірних нормалізованих даних було вилючено 9 багатовимірних викидів, з яких 3 викиди вилучено за квадратом відстані Махаланобіса, 5 – через невідповідність розподілу залишків нормальному закону і 1 викид опинився за межами інтервалу передбачення. Побудована п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання KLOC DS та ML JAVA-застосунків на основі шестивимірного перетворення Бокса-Кокса має такі обмеження діапазонів значень незалежних факторів: $X_1 \in [30; 11230]$, $X_2 \in [1; 1084]$, $X_3 \in [2, 1569; 13, 1114]$, $X_4 \in [1, 0077; 4, 5982]$ та $X_5 \in [2, 8895; 12, 5902]$. На останній ітерації для факторів CLS, INFC, aVMQ, aTFQ і aCBO розраховано значення VIFs, що становлять 2,5, 2,6, 1,4, 1,3 і 1,2 відповідно. Отримані значення VIFs показують відсутність мультиколінеарності між незалежними змінними моделі. На останньому етапі, модель (3.17) було оцінено за критеріями якості нелінійних регресійних моделей, які наведені в таблиці 3.4.

Таблиця 3.4 – Значення критеріїв якості для моделі (3.17)

	R^2	R_{adj}^2	$MMRE$	$PRED(0, 25)$
Навчальна вибірка	0,9901	0,9893	0,1278	0,8462
Тестова вибірка	0,9456	-	0,1798	0,7631

Наведені результати показують високий рівень якості отриманої п'ятифакторної нелінійної регресійної моделі для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків на базі шестивимірною перетворення Бокса-Кокса (3.17).

Побудуємо п'ятифакторну нелінійну регресійну модель на основі одновимірною перетворення Бокса-Кокса для обробки інформації з метрик коду для оцінювання розміру KLOC (Y) DS та ML JAVA-застосунків за факторами CLS (X_1), INFC (X_2), aVMQ (X_3), aTFQ (X_4) та aCBO (X_5). Доцільність вибору цього нормалізуючого перетворення наведено в пп. 2.4 дисертаційного дослідження. Вихідними даними для побудови моделі є навчальна вибірка із значеннями незалежних факторів, наведеними в додатку В.

Негаусівські дані навчальної вибірки (див. додаток В) перетворено на гаусівські із застосуванням одновимірною нормалізуючого перетворення Бокса-Кокса. Під час перевірки нормалізованих багатовимірних даних за тестом Мардія для 56 ступенів вільності та рівня значущості $\alpha = 0,005$ було виявлено, що розподіл даних не є нормальним. Далі з метою вилучення викидів, всі значення навчальної вибірки були перевірені відповідно до алгоритму, що реалізує тест Мардія та квадрата відстані Махаланобіса (3.3). Для першої ітерації оцінка багатовимірної асиметрії $N\beta_1/6 = 587,59$ перевищує значення квантиля $\chi^2_{кр} = 86,99$ для 56 ступенів вільності та рівня значущості $\alpha = 0,005$, оцінка багатовимірною ексцесу $\beta_2 = 92,80$ перевищує значення квантиля розподілу Гауса $\mathcal{N}_{1-\alpha}(\mu, \sigma^2) = 51,89$ для $\mu = 48$ та $\sigma^2 = 2,28$. На ітераціях 1, 2, 3 видалено точки під номерами 37, 68 та 52, як багатовимірні викиди, оскільки значення квадрата відстані Махаланобіса становлять 62,14, 26,31 та 21,12 відповідно, що перевищує значення квантиля розподілу $\chi^2_{кр} = 14,86$ для рівня значущості $\alpha = 0,005$ та чотирьох ступенів вільності.

Оцінки параметрів одновимірного нормалізуючого перетворення (2.8) залежного фактора Y та незалежних факторів X_1, X_2, X_3, X_4 та X_5 мають такі значення на останній ітерації:

$$\hat{\lambda}_Y = 8,3856 \cdot 10^{-2}, \hat{\lambda}_{X_1} = 4,7835 \cdot 10^{-2}, \hat{\lambda}_{X_2} = 7,5329 \cdot 10^{-2},$$

$$\hat{\lambda}_{X_3} = -0,3310, \hat{\lambda}_{X_4} = 6,8548 \cdot 10^{-3} \text{ та } \hat{\lambda}_{X_5} = 0,2649.$$

Оцінки параметрів перетворення отримано за ММП для кожного незалежного фактора окремо шляхом максимізації одновимірної логарифмічної функції правдоподібності (2.11). Для останньої ітерації значення тестової статистики розподілу для багатовимірної асиметрії $B_1 N/6 = 60,84$ не перевищує значення квантиля розподілу $\chi^2_{\text{кр}} = 86,99$ для 56 ступенів вільності та рівня значущості 0,005. Тестова статистика розподілу цієї вибірки для багатовимірного ексцесу $B_2 = 50,14$ не перевищує значення квантиля розподілу Гауса $\mathcal{N}_{1-\alpha}(\mu, \sigma^2) = 52,01$ для $\mu = 48$ та $\sigma^2 = 2,43$ і рівня значущості $\alpha = 0,005$. Це свідчить про нормальну природу розподілу нормалізованих даних без викидів. Слід зауважити, що на кожній ітерації вилучали по одному викиду.

Далі для нормалізованих даних побудовано лінійну регресійну модель (3.5) з кількістю незалежних факторів $k = 5$. Оцінки параметрів п'ятифакторної лінійної регресійної моделі на основі нормалізованих даних для останньої ітерації, отримані за методом найменших квадратів, становлять $\hat{b}_0 = -5,8182$, $\hat{b}_1 = 1,0670$, $\hat{b}_2 = -5,3790 \cdot 10^{-2}$, $\hat{b}_3 = 1,7255$, $\hat{b}_4 = 0,3226$ та $\hat{b}_5 = -3,6842 \cdot 10^{-2}$.

П'ятифакторна лінійна регресійна модель для нормалізованих даних DS та ML JAVA-застосунків має такий вигляд:

$$\begin{aligned} \hat{Z}_{Y_{DSML}^*} + \varepsilon = & \hat{b}_0 + \hat{b}_1 \psi_1(X_1) + \hat{b}_2 \psi_2(X_2 + 1) + \\ & + \hat{b}_3 \psi_3(X_3) + \hat{b}_4 \psi_4(X_4) + \hat{b}_5 \psi_5(X_5) + \varepsilon, \end{aligned} \quad (3.19)$$

де $\psi_j(X_j) = (X_j^{\hat{\lambda}_{X_j}} - 1) / \hat{\lambda}_{X_j}$ – одновимірне нормалізуюче перетворення

Бокса-Кокса із значенням оцінки перетворення $\hat{\lambda}_{X_j}$ для незалежного фактора X_j ($j = 1, \dots, 5$).

Для побудованої п'ятифакторної лінійної регресійної моделі залишки регресії ε перевірено на відповідність нормальному закону розподілу за критерієм Пірсона χ^2 . На цьому етапі виявлено та вилучено 1 проєкт під номером 16 як викид, оскільки значення критерію Пірсона $\chi^2 = 14,90$ перевищує порогове допустиме значення $\chi_{кр}^2 = 9,49$ для рівня значущості $\alpha = 0,05$ та 4 ступенів вільності.

Після застосування зворотного нормалізуючого перетворення (3.2) до лінійної регресійної моделі (3.19) п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків на базі одновимірного перетворення Бокса-Кокса має такий вигляд

$$Y_{DSML^*} = \psi_Y^{-1}(\hat{Z}_{Y_{DSML}^*} + \varepsilon) = \left[\hat{\lambda}_Y \left(\hat{Z}_{Y_{DSML}^*} + \varepsilon \right) + 1 \right]^{\frac{1}{\hat{\lambda}_Y}}. \quad (3.20)$$

Для отриманої нелінійної регресійної моделі (3.20) визначено верхню та нижню границі інтервалів передбачення (3.7) та довірчих інтервалів (3.8) для рівня значущості $\alpha = 0,05$ та виявлено 5 багатовимірних викидів з ідентифікаторами 33, 8, 50, 26 та 23, які вилучено із нормалізованої навчальної вибірки.

Інтервал передбачення та довірчий інтервал нелінійної регресійної моделі Y_{DSML^*} (3.20) мають такий вигляд

$$\begin{aligned}\hat{Y}_{PI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{DSML*}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ 1 + \frac{1}{N} + (Z_X^+)^T \hat{S}_Z^{-1} (Z_X^+) \right\}^{1/2} \right), \\ \hat{Y}_{CI} &= \psi_Y^{-1} \left(\hat{Z}_{Y_{DSML*}} \pm \hat{t}_{\alpha/2, v} \hat{S}_{Z_Y} \left\{ \frac{1}{N} + (Z_X^+)^T \hat{S}_Z^{-1} (Z_X^+) \right\}^{1/2} \right),\end{aligned}\quad (3.21)$$

де $\hat{Z}_{Y_{DSML*}}$ – оцінка параметра розміру для нормалізованих даних за математичною моделлю (3.19).

Для останньої ітерації значення нормалізованих вибірових середніх становлять: $\bar{Z}_1 = 7,5106$, $\bar{Z}_2 = 4,2338$, $\bar{Z}_3 = 1,2343$, $\bar{Z}_4 = 0,8549$ та $\bar{Z}_5 = 2,3915$. Квантиль t -розподілу Стюдента становить $t_{\alpha/2, v} = 2,001$ для рівня значущості $\alpha = 0,05$ та 59 ступенів вільності; $S_{Z_Y} = 0,2217$. Обернена коваріаційна матриця (3.9) має вигляд

$$S_Z^{-1} = \begin{vmatrix} 2,5684 \cdot 10^{-2} - 1,8316 \cdot 10^{-2} & 8,5506 \cdot 10^{-3} - 1,6957 \cdot 10^{-2} - 1,3398 \cdot 10^{-2} \\ -1,8316 \cdot 10^{-2} & 1,6713 \cdot 10^{-2} - 1,8148 \cdot 10^{-2} & 1,5017 \cdot 10^{-2} & 6,3511 \cdot 10^{-3} \\ 8,5506 \cdot 10^{-3} - 1,8148 \cdot 10^{-2} & 6,3048 \cdot 10^{-1} - 1,3722 \cdot 10^{-1} - 2,9557 \cdot 10^{-2} \\ -1,6957 \cdot 10^{-2} & 1,5017 \cdot 10^{-2} - 1,3722 \cdot 10^{-1} & 1,4567 \cdot 10^{-1} - 1,1398 \cdot 10^{-2} \\ -1,3398 \cdot 10^{-2} & 6,3511 \cdot 10^{-3} - 2,9557 \cdot 10^{-2} - 1,1398 \cdot 10^{-2} & 1,2407 \cdot 10^{-1} \end{vmatrix}$$

П'ятифакторна нелінійна регресійна модель (3.20) побудована за 10 ітерацій. З нормалізованих даних було вилучено 9 багатовимірних викидів, з яких 3 викиди вилучено за квадратом відстані Махаланобіса, 1 через невідповідність розподілу залишків нормальному закону, а 5 викидів опинилися за межами інтервалу передбачення. Побудована п'ятифакторна нелінійна регресійна модель для обробки інформації з метрик коду для оцінювання KLOC DS та ML JAVA-застосунків на основі одновимірного перетворення Бокса-Кокса має такі обмеження значень незалежних факторів:

$X_1 \in [30; 11230]$, $X_2 \in [1; 1084]$, $X_3 \in [2, 6081; 10, 9542]$, $X_4 \in [0, 9975; 4, 5982]$ та $X_5 \in [2, 8895; 12, 5902]$. На останній ітерації для факторів CLS, INFC, aVMQ, aTFQ і aCBO розраховано значення VIF, що становлять 2,4, 2,4, 1,2, 1,3 і 1,1 відповідно. Отримані значення VIFs показують відсутність мультиколінеарності між незалежними змінними моделі. На останньому етапі, модель (3.20) було оцінено за критеріями якості нелінійних регресійних моделей, які наведені в таблиці 3.5.

Таблиця 3.5 – Значення критеріїв якості для моделі (3.20)

	R^2	$MMRE$	$PRED(0, 25)$
Навчальна вибірка	0,9800	0,1318	0,8462
Тестова вибірка	0,9421	0,2222	0,6579

Наведені результати показують високий рівень якості побудованої п'ятифакторної нелінійної регресійної моделі для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків на базі одновимірного перетворення Бокса-Кокса (3.20).

3.5 Порівняння результатів достовірності обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та Data Science та Machine Learning JAVA-застосунків з існуючими нелінійними регресійними моделями

Проведено порівняльний аналіз для оцінювання точності обробки інформації з метрик коду JAVA-застосунків із використанням побудованих п'ятифакторних нелінійних регресійних моделей з вже існуючими раніше побудованими нелінійними регресійними моделями із використанням критеріїв якості регресійних моделей R^2 (1.4), $MMRE$ (1.6) та $PRED(0, 25)$ (1.8) на навчальній і на тестовій вибірках відповідних категорій JAVA-застосунків.

Для порівняння точності обробки інформації з метрик коду для оцінювання параметра розміру для загальної вибірки JAVA-застосунків обрано трьохфакторну нелінійну регресійну модель на базі багатовимірного перетворення Джонсона сім'ї SB за метриками CLASS (X_1), RFC (X_2) та aVMQ (X_3) [30], чотирьохфакторну нелінійну регресійну модель (3.11) та чотирьохфакторну нелінійну регресійну модель на базі багатовимірного перетворення Бокса-Кокса за метриками CLASS (X_1), aVMQ (X_2), aTFQ (X_3) і aCBO (X_4) [36].

Для порівняння точності обробки інформації з метрик коду для оцінювання параметра розміру для DS та ML вибірки JAVA-застосунків було обрано вищезазначені нелінійні регресійні моделі для загальних JAVA-застосунків, а також трьохфакторну нелінійну регресійну модель на базі перетворення десяткового логарифма за метриками CLASS (X_1), TFQ (X_2) та VMQ (X_3) [28], та чотирьохфакторну нелінійну регресійну модель на базі багатовимірного перетворення Бокса-Кокса за метриками CLASS (X_1), aVMQ (X_2), aTFQ (X_3) і aCBO (X_4) [38].

Оцінки показників точності нелінійних регресійних моделей для навчальної та тестової вибірок загальних JAVA-застосунків наведено в таблицях 3.6 та 3.7 відповідно.

Таблиця 3.6 – Оцінки критерії якості нелінійних регресійних моделей на базі навчальної вибірки.

№	Нелінійна регресійна модель	Навчальна вибірка		
		R^2	MMRE	PRED
1	2	3	4	5
1	Трьохфакторна на базі перетворення Джонсона [30]	0,9179	0,1560	0,7867

Продовження табл.3.6

1	2	3	4	5
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	0,9067	0,1526	0,8181
3	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, загальної кількості класів та середніх значень метрик [36]	0,9150	0,1717	0,8146
4	Побудована п'ятифакторна нелінійна регресійна модель (3.14)	0,9294	0,1540	0,8462

Таблиця 3.7 – Оцінки критерії якості нелінійних регресійних моделей на базі тестової вибірки.

№	Нелінійна регресійна модель	Тестова вибірка		
		R^2	MMRE	PRED
1	Трьохфакторна на базі перетворення джонсона [30]	0,9094	0,1583	0,8315
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	0,8981	0,1536	0,8211
3	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, загальної кількості класів та середніх значень метрик [36]	0,9075	0,1854	0,7439
4	Побудована п'ятифакторна нелінійна регресійна модель (3.14)	0,9193	0,1742	0,7825

Окрім оцінки точності прогнозування за критеріями якості R^2 , $MMRE$ та $PRED(0, 25)$, проведено перевірку доцільності переходу від чотирьохфакторної нелінійної регресійної моделі до п'ятифакторної шляхом розрахунку скоригованого коефіцієнта детермінації R_{adj}^2 (1.5). Розрахунок проведено на очищених навчальних вибірках, що дає змогу оцінити значення параметра розміру без спотворень, викликаних викидами. Для чотирьохфакторної нелінійної регресійної моделі [36], отриманої на попередніх етапах

дослідження, це значення становить $R_{adj}^2 = 0,9511$, а для побудованої п'ятифакторної нелінійної регресійної моделі це значення становить $R_{adj}^2 = 0,9641$, що виправдовує перехід до п'ятифакторної нелінійної регресійної моделі.

Оцінки показників точності нелінійних регресійних моделей для DS та ML JAVA-застосунків наведені в таблиці 3.8 для навчальної вибірки та в таблиці 3.9 для тестової вибірки.

Таблиця 3.8 – Оцінки критерії якості нелінійних регресійних моделей на базі навчальної вибірки DS та ML JAVA-застосунків.

№	Нелінійна регресійна модель	Навчальна вибірка		
		R ²	MMRE	PRED
1	Трьохфакторна на базі перетворення Джонсона [30]	0,8907	0,1758	0,7568
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	0,8441	0,1781	0,7162
3	Побудована трьохфакторна нелінійна регресійна модель для оцінювання DS та ML JAVA-застосунків [28]	0,9226	0,1853	0,7703
4	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, для оцінювання DS та ML JAVA-застосунків [38]	0,9198	0,1776	0,7972
5	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для загальних JAVA-застосунків (3.14)	0,8884	0,1884	0,7297
6	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для DS та ML JAVA-застосунків (3.17)	0,9227	0,1721	0,7568
7	Побудована п'ятифакторна нелінійна регресійна модель на основі одновимірних перетворень Бокса-Кокса для DS та ML JAVA-застосунків (3.20)	0,8919	0,1789	0,7432

Таблиця 3.9 – Оцінки критерії якості нелінійних регресійних моделей на базі тестової вибірки DS та ML JAVA-застосунків.

№	Нелінійна регресійна модель	Тестова вибірка		
		R ²	MMRE	PRED
1	Трьохфакторна на базі перетворення Джонсона [30]	0,9269	0,2025	0,6579
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	0,9209	0,2097	0,6316
3	Побудована трьохфакторна нелінійна регресійна модель для оцінювання DS та ML JAVA-застосунків [28]	0,9347	0,2073	0,6316
4	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, для оцінювання DS та ML JAVA-застосунків [38]	0,9306	0,1927	0,7105
5	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для загальних JAVA-застосунків (3.14)	0,9335	0,1614	0,7894
6	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для DS та ML JAVA-застосунків (3.17)	0,9456	0,1798	0,7631
7	Побудована п'ятифакторна нелінійна регресійна модель на основі одновимірних перетворень Бокса-Кокса для DS та ML JAVA-застосунків (3.20)	0,9421	0,2222	0,6579

Для перевірки доцільності переходу від чотирьохфакторної нелінійної регресійної моделі до п'ятифакторної розраховано скоригований коефіцієнт детермінації R_{adj}^2 (1.5) на очищених навчальних вибірках. Це дозволяє оцінити значення параметра розміру без спотворень, викликаних викидами. Для чотирьохфакторної нелінійної регресійної моделі [38], отриманої на попередніх етапах дослідження, це значення становить $R_{adj}^2 = 0,9671$, а для побудованої

п'ятифакторної нелінійної регресійної моделі це значення становить $R_{adj}^2 = 0,9893$, що виправдовує перехід до п'ятифакторної нелінійної регресійної моделі.

Методом експертного оцінювання проведено вибір альтернатив за критеріями якості отриманих та існуючих нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків. Для цього застосовано підхід, заснований на рангових оцінках, де найвищий бал відповідає найкращому значенню відповідного критерію якості [86, 87]. Інтегральна оцінка якості моделі для кожної альтернативи ($Cr(a_i)$) визначається за формулою

$$Cr(a_i) = \sum_{j=1}^M E_j, \quad (3.22)$$

де a_i – альтернативи, що розглядаються, E_j – експертна оцінка j -го критерію, $M = 6$ – кількість критеріїв (див. табл. 3.10).

Таблиця 3.10 – Визначення критеріїв для пошуку найкращих альтернатив

Навчальна вибірка			Тестова вибірка		
R^2	MMRE	PRED	R^2	MMRE	PRED
E_1	E_2	E_3	E_4	E_5	E_6

Найкраща альтернатива визначається за формулою

$$a^\circ = \arg \max Cr(a_i), i = 1, 2, \dots, N, \quad (3.23)$$

де N – кількість альтернатив.

Для пошуку найкращої альтернативи використано нелінійні регресійні моделі для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, які наведені в таблиці 3.11.

Таблиця 3.11 – Альтернативи для вибору моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків

№	Нелінійна регресійна модель	Альтернатива
1	Трьохфакторна на базі перетворення Джонсона [30]	a_1
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	a_2
3	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, загальної кількості класів та середніх значень метрик [36]	a_3
4	Побудована п'ятифакторна нелінійна регресійна модель (3.14)	a_4

Наступним кроком побудовано матрицю експертних оцінок відповідних альтернатив для вибору моделі обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків. Слід зазначити, що альтернативи a_1 та a_2 мають обмеження (див. пп. 1.1, 3.2), тому їх виключено з порівняння. У таблиці 3.12 наведено результати вибору найкращої моделі.

Таблиця 3.12 – Експертні оцінки для множини альтернатив (для загальних JAVA-застосунків)

	E1	E2	E3	E4	E5	E6	$Cr(a_i)$
a_3	1	1	1	1	1	1	6
a_4	2	2	2	2	2	2	12

Таким чином, $a_4 > a_3$, тобто найкращою моделлю для обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків з

точки зору якості є п'ятифакторна нелінійна регресійна модель (3.14) на основі шестивимірного нормалізуючого перетворення Бокса-Кокса (a_4).

Для пошуку найкращої альтернативи з моделей для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків розглянуто нелінійні регресійні моделі, які наведено в таблиці 3.13.

Таблиця 3.13 – Альтернативи для вибору моделі обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків

№	Нелінійна регресійна модель	Альтернатива
1	Трьохфакторна на базі перетворення Джонсона [30]	a_1
2	Чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса та з метрикою RFC (3.11)	a_2
3	Побудована трьохфакторна нелінійна регресійна модель для оцінювання DS та ML JAVA-застосунків [28]	a_3
4	чотирьохфакторна на основі багатовимірного перетворення Бокса-Кокса, для оцінювання DS та ML JAVA-застосунків [38]	a_4
5	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для загальних JAVA-застосунків (3.14)	a_5
6	Побудована п'ятифакторна нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса для DS та ML JAVA-застосунків (3.17)	a_6
7	Побудована п'ятифакторна нелінійна регресійна модель на основі одновимірних перетворень Бокса-Кокса для DS та ML JAVA-застосунків (3.20)	a_7

Побудовано матрицю експертних оцінок відповідних альтернатив для вибору моделі обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків. Альтернативи a_1 та a_2 мають обмеження (див. пп. 1.1, 3.2), тому їх виключено з порівняння. У таблиці 3.14 наведено результати вибору найкращої моделі.

Таблиця 3.14 – Експертні оцінки для множини альтернатив (для DS та ML JAVA-застосунків)

	E1	E2	E3	E4	E5	E6	$Cr(a_i)$
a_3	4	2	4	3	2	1	16
a_4	3	4	5	1	3	3	19
a_5	1	1	1	2	5	5	15
a_6	5	5	3	5	4	4	26
a_7	2	3	1	4	1	2	13

Маємо $a_6 > a_4 > a_3 > a_5 > a_7$, тобто найкращою моделлю для обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків з точки зору якості є п'ятифакторна нелінійна регресійна модель (3.17) на основі шестивимірного нормалізуючого перетворення Бокса-Кокса (a_6).

Можна зробити висновок, що розділення метрики CLASS на окремі метрики, що відображають загальну кількість класів (CLS) та інтерфейсів (INFC), дало змогу побудувати моделі з найкращими показниками якості (R^2 , $MMRE$ та $PRED(0,25)$), як для оцінювання параметра розміру загальних JAVA-застосунків, так і для DS та ML JAVA-застосунків.

Серед обраних до порівняння DS та ML нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання параметра розміру п'ятифакторна нелінійна регресійна модель (3.17) має найкращі показники якості (табл. 3.8, 3.9 та 3.14). Виявлено, що

- нелінійна регресійна модель на основі шестивимірного перетворення Бокса-Кокса (3.17) показала вищі оцінки якості для оцінювання параметра розміру DS та ML JAVA-застосунків, ніж п'ятифакторна нелінійна

регресійна модель за такими ж факторами, але на основі одновимірного перетворення Бокса-Кокса (3.20), незважаючи на те, що одновимірне перетворення Бокса-Кокса показало кращі початкові оцінки нормалізації відповідно до тесту Мардія;

- серед всіх нелінійних регресійних моделей, взятих до порівняння, тільки математична модель (3.17) досягла порогових значень для всіх критеріїв якості;

- доцільність побудови математичної моделі обробки інформації для оцінювання розміру ПЗ саме з урахуванням технологічних особливостей та прикладних аспектів застосування DS та ML в JAVA-застосунках є виправданою, оскільки таблиці 3.8, 3.9, 3.14 показали, що математична модель для DS та ML JAVA-застосунків має кращі показники якості у порівнянні із запропонованими моделями. Тому саме для підвищення достовірності оцінювання параметра розміру DS та ML JAVA-застосунків доцільно використовувати побудовану математичну модель, що враховує відповідні технологічні фактори середовища та специфіку розробки ПЗ за цим напрямом.

- п'ятифакторна нелінійна регресійна модель для обробки інформації для оцінювання розміру загальних JAVA-застосунків (3.14) показала високі оцінки критеріїв якості для категорії DS та ML JAVA-застосунків. Слід зазначити, що не всі оцінки якості цієї моделі досягли порогових значень у порівнянні з математичною моделлю для DS та ML JAVA-застосунків (3.17). Це підтверджує необхідність побудови нелінійних регресійних моделей для окремих категорій JAVA-застосунків для досягнення вищого рівня достовірності оцінювання, але, за відсутності таких моделей, п'ятифакторна нелінійна регресійна модель (3.14) може бути використана для обробки інформації з метрик коду для оцінювання параметра розміру.

3.6 Порівняння інтервалів передбачення за побудованими нелінійними регресійними моделями обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків та Data Science та Machine Learning JAVA-застосунків

Крім показників критеріїв якості моделей, таких як R^2 , $MMRE$ та $PRED(0,25)$, на достовірність обробки інформації з метрик коду для оцінювання розміру і трудомісткості в умовах невизначеності впливає ширина інтервалу передбачення, яка враховує не тільки середні оцінки параметра розміру, а і довірчу імовірність знаходження оцінки в інтервальному діапазоні. Тому чим менша ширина інтервалу передбачення, тим вища точність прогнозування, що є підтвердженням високої якості моделі та низької дисперсії помилок.

У таблиці 3.15 наведено порівняльну характеристику ширин інтервалів передбачення між чотирьохфакторними нелінійними регресійними моделями (3.11)[36] та п'ятифакторною нелінійною регресійною моделлю (3.14) на основі вибірки загальних JAVA-застосунків.

Таблиця 3.15 – Порівняння ширин інтервалів передбачення між чотирьохфакторними та п'ятифакторною нелінійними регресійними моделями для загальних JAVA-застосунків

		Нелінійна регресійна модель					
		чотирьохфакторна (3.11)		чотирьохфакторна [36]		п'ятифакторна (3.14)	
1	2	3	4	5	6	7	8
№	Y	LB	UB	LB	UB	LB	UB
394	9,568	5,907	17,110	4,527	14,929	4,774	14,918
244	9,606	4,754	13,603	4,973	16,390	5,291	16,489
157	9,680	6,843	19,983	5,772	18,998	6,047	18,790
	...	...	...	...	...	...	...

Продовження табл.3.15

1	2	3	4	5	6	7	8
496	22,829	14,178	43,124	11,589	38,062	11,285	34,784
335	23,024	15,757	48,187	12,754	41,788	14,668	44,869
482	23,315	16,027	49,044	14,650	47,971	15,607	47,644
	...	...	...	...	...	...	...
202	44,250	27,582	87,656	25,806	84,694	26,952	81,870
328	45,803	28,878	91,489	28,715	93,712	30,762	92,758
382	45,965	24,106	75,604	22,626	73,925	23,766	71,993
	...	...	...	...	...	...	...
117	102,344	46,647	152,112	45,484	148,098	50,606	151,301
37	104,263	58,100	192,220	60,993	198,613	66,785	199,160
548	104,937	61,364	203,754	64,255	209,194	66,114	196,942
	...	...	...	...	...	...	...
104	628,639	280,980	1034,813	335,130	1086,247	343,867	998,575
182	699,674	205,629	739,136	189,812	615,465	204,840	599,127
430	777,036	269,018	987,471	382,835	1241,003	365,069	1060,118

Виявлено, що модель (3.14) має на 12,9% вузький інтервал, ніж чотирьохфакторна нелінійна регресійна модель (3.11) [35], та на 9,6% вузький інтервал, ніж чотирьохфакторна нелінійна регресійна модель [36-38]. Таким чином, можна зробити висновок, що п'ятифакторна нелінійна регресійна модель (3.14) залежно від метрик CLS, INFC, aVMQ, aTFQ та aCBO на основі шестивимірного перетворення Бокса-Кокса є найкращою за ширинами інтервалів передбачення для більшої кількості даних у порівнянні з чотирьохфакторними нелінійними регресійними моделями (3.11) та [36].

Кращі показники обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків за нелінійною регресійною моделлю (3.14) на основі шестивимірного перетворення Бокса-Кокса можна пояснити використанням кращої багатовимірної нормалізації, яку перевірено за відомим критерієм на основі багатовимірних ексцесу та асиметрії, а також застосуванням розділення метрики CLASS на дві окремі – CLS та INFC.

Порівняно ширини інтервалів передбачення між чотирьохфакторною нелінійною регресійною моделлю [38] та п'ятифакторною нелінійною

регресійною моделлю (3.17) для DS та ML JAVA-застосунків за формулою (3.22). Значення інтервалів передбачення для запропонованих нелінійних регресійних моделей наведено в таблиці 3.16.

Таблиця 3.16 – Порівняння ширин інтервалів передбачення між чотирьохфакторною та п'ятифакторною нелінійними регресійними моделями для DS та ML JAVA-застосунків

№	Y	Нелінійна регресійна модель					
		чотирьохфакторна [38]			п'ятифакторна (3.17)		
		$\hat{Y}$	LB	UB	$\hat{Y}$	LB	UB
39	12,985	15,808	4,977	47,936	15,319	4,424	45,997
29	14,469	15,995	4,959	49,196	15,785	4,512	47,783
31	15,737	17,066	5,349	51,965	17,207	5,024	51,231
	...						
30	29,704	41,221	13,336	121,902	40,213	12,991	110,496
9	32,999	37,890	12,279	111,881	36,594	11,804	100,677
60	35,976	34,333	10,978	102,639	34,155	10,846	95,151
							
20	51,23	68,920	23,096	197,296	65,751	22,879	170,179
27	58,864	48,623	15,574	145,127	51,953	17,094	140,684
49	61,27	57,330	18,867	166,891	58,598	19,844	155,038
	...						
25	98,284	96,165	32,109	276,211	97,024	34,827	244,847
11	105,104	138,088	46,304	395,072	129,069	47,303	320,224
41	107,002	111,718	37,289	320,995	107,187	38,622	269,654
	...						
71	200,338	235,492	79,994	665,720	201,832	75,785	490,882
23	204,252	151,878	52,126	425,264	151,865	57,756	365,490
74	207,825	219,775	75,967	611,338	227,347	83,096	565,307
	...						
40	362,876	280,940	97,055	781,887	276,705	108,849	647,577
26	458,003	379,489	131,633	1052,187	324,226	128,024	756,424
66	624,258	641,877	225,517	1758,667	617,291	259,461	1366,842

Серед запропонованих моделей, модель (3.17) має на 18,5% вужчий інтервал, ніж чотирьохфакторна нелінійна регресійна модель [38]. Таким чином,

можна зробити висновок, що п'ятифакторна нелінійна регресійна модель (3.17) залежно від метрик CLS, INFC, aVMQ, aTFQ та aCBO на основі шестивимірного перетворення Бокса-Кокса є найкращою за ширинами інтервалів передбачення для більшої кількості даних DS та ML JAVA-застосунків у порівнянні з чотирьохфакторною нелінійною регресійною моделлю [38].

Кращі показники обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків за нелінійною регресійною моделлю (3.17) на основі шестивимірного перетворення Бокса-Кокса можна пояснити використанням кращої багатовимірної нормалізації, яку перевірено за відомим критерієм на основі багатовимірних ексцесу та асиметрії, а також застосуванням розділення метрики CLASS на дві окремі – CLS та INFC.

3.7 Висновки за розділом 3

У третьому розділі побудовано чотирьохфакторну та п'ятифакторні нелінійні регресійні моделі обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків з відкритим кодом на основі багатовимірних нормалізуючих перетворень Бокса-Кокса. Для побудованих математичних моделей проведено порівняння якості обробки інформації для оцінювання параметра розміру за критеріями R^2 , R^2_{adj} , $MMRE$ та $PRED(0, 25)$ з метрик загальних JAVA-застосунків, DS та ML JAVA-застосунків з відкритим кодом, за даними навчальної та тестової вибірок. Розраховано інтервали передбачення та довірчі інтервали для відповідних нелінійних регресійних моделей, зроблено їх порівняльний аналіз.

1) Удосконалено чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання розміру (KLOC) загальних JAVA-застосунків із використанням метрик CLASS, RFC, aCBO та aVMQ на основі п'ятивимірного нормалізуючого перетворення Бокса-Кокса. Надійність

моделі підтверджено за допомогою збалансованого поділу набору даних загальних JAVA-застосунків на навчальну та тестову вибірки розміром 286 та 285 точок даних. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має кращі значення критеріїв якості обробки інформації з метрик коду для оцінювання параметра розміру, а саме R^2 , $MMRE$ та $PRED(0, 25)$.

2) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру KLOC загальних JAVA-застосунків на базі метрик CLS, INFC, aVMQ, aTFQ та aCBO з використанням шестивимірною перетворення Бокса-Кокса. Заміна метрики CLASS на окремі метрики CLS та INFC порівняно з існуючими нелінійними регресійними моделями, та збалансований поділ навчальної та тестової вибірок обсягом 286 та 285 спостережень забезпечують високу надійність перевірки отриманої математичної моделі та достовірність оцінювання KLOC, що підтверджують значення оцінок критеріїв якості R^2 , $MMRE$ та $PRED(0, 25)$. Критерій R_{adj}^2 для п'ятифакторної моделі на базі метрик CLS, INFC, aVMQ, aTFQ та aCBO має вище значення, ніж для чотирьохфакторної моделі, що виправдовує доцільність її побудови.

3) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру KLOC DS та ML JAVA-застосунків на базі метрик CLS, INFC, aVMQ, aTFQ та aCBO із використанням шестивимірною перетворення Бокса-Кокса. Отримана п'ятифакторна нелінійна регресійна модель, у порівнянні з існуючими нелінійними регресійними моделями, дозволяє врахувати технологічні фактори середовища та специфіку розробки ПЗ за цим напрямом. Модель показала кращий рівень достовірності обробки інформації з оцінювання параметра розміру за критеріями якості нелінійних регресійних моделей R^2 , $MMRE$ та $PRED(0, 25)$ у порівнянні з існуючими математичними моделями і отриманою п'ятифакторною нелінійною регресійною моделлю на основі вибірки загальних JAVA-застосунків.

4) Отримали подальший розвиток рівняння для визначення границь інтервалу передбачення та довірчого інтервалу п'ятифакторної нелінійної регресійної моделі (3.14) для загальних JAVA-застосунків та п'ятифакторної нелінійної регресійної моделі (3.17) DS та ML JAVA-застосунків. Виявлено, що ширина інтервалів передбачення для побудованих п'ятифакторних нелінійних регресійних моделей вужча у порівнянні з чотирьохфакторними нелінійними регресійними моделями. Це дає змогу підвищити достовірність обробки інформації для оцінювання параметра розміру як залежної випадкової величини.

Результати дослідження поточного розділу автором опубліковано в роботах [28-30, 35-40].

РОЗДІЛ 4. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК JAVA-ЗАСТОСУНКІВ

У четвертому розділі запропоновано інформаційну технологію (ІТ) для обробки інформації з метрик коду для оцінювання параметра розміру (KLOC) JAVA-застосунків на основі вибірок з відкритим кодом загальних JAVA-застосунків, DS та ML JAVA-застосунків. Розроблено інструментарій ІТ для обробки інформації для оцінювання розміру JAVA-застосунків на етапах аналізу і проєктування із використанням побудованих нелінійних регресійних моделей (3.14) та (3.17) за п'ятьма метриками діаграми класів (CLS, INFC, aVMQ, aTFQ та aCBO). Створено відповідне ПЗ мовою програмування Kotlin на базі фреймворку Spring Boot. ПЗ розповсюджується за моделлю Software as a Service (SaaS), доступ до якого надається за допомогою ключів доступу. Розроблено методику оцінювання розміру JAVA-застосунків, яка розрахована на використання цього ПЗ.

4.1 Створення інформаційної технології для обробки інформації з метрик JAVA-застосунків для оцінювання параметра розміру на основі вибірки проєктів з відкритим кодом

Термін «інформаційна технологія» (ІТ) має багато визначень залежно від періоду розвитку цієї галузі, появи нових методів та підходів для створення високоефективних рішень, які задовольняють потреби сучасного суспільства в обробці й аналізі інформації. Гарольд Дж. Левітт і Томас Л. Віслер в роботі [88] одні з перших ввели термін «інформаційна технологія» і визначили її як нову область, що змінить бізнес-процеси та управління завдяки автоматизації й обробці інформації. Вони виділили такі ключові характеристики та особливості ІТ [88]:

- застосування методів для швидкої обробки великої кількості інформації за допомогою високошвидкісних комп'ютерів;

- автоматизація обліку, обробку даних і процесу прийняття рішень, що відкриває нові можливості для розвитку бізнесу;
- трансформація традиційних бізнес-моделей шляхом впровадження комп'ютеризації робочих процесів, що прискорює роботу з інформацією та підвищує точність аналізу даних;
- оптимізація управлінських рішень через використання аналітичних даних, що підтримує реагування на зміни ринкових умов та збільшує конкурентоспроможність.

Це дослідження окреслило межі впливу ІТ на бізнес та управлінські процеси, що створило основу подальшого розвитку інформаційних систем і технологій як невід'ємної частини сучасної підприємницької діяльності. Відповідно до Оксфордського словника фізичних термінів [89], ІТ – це використання комп'ютерів і телекомунікаційного обладнання (з відповідною мікроелектронікою) для надсилання, отримання, зберігання та маніпулювання даними. Дані можуть бути текстовими, числовими, аудіо чи відео, або будь-якою їх комбінацією.

В Україні термін «Інформаційна технологія», відповідно до ДСТУ 2226-93 [90], визначається як сукупність методів, засобів і процесів для автоматизації обробки інформації в автоматизованих системах, головною метою яких є ефективне використання комп'ютерної техніки та ПЗ для роботи з інформаційними ресурсами, що включає автоматичне збирання, обробку, зберігання, передавання та захист даних. ДСТУ 2481-94 [91] вводить термін «інтелектуальні інформаційні технології», яких охоплює використання методів штучного інтелекту, експертних систем, інструментів аналізу і моделювання даних для підвищення ефективності ухвалення рішень в автоматизованих системах для розв'язання складних задач у науці, медицині, бізнесі й промисловості.

ІТ для обробки інформації з метрик коду JAVA-застосунків – це програмне рішення, яке надає інструментарій ІТ для оцінювання параметра розміру KLOC на основі інформації з метрик програмного коду, які можна

отримати із концептуальної моделі даних проєкту ПЗ у вигляді UML-діаграми класів на початкових етапах розробки.

Метою створення ІТ є автоматизація розрахунків для обробки інформації з метрик коду JAVA-застосунків, що дозволить підвищити якість та достовірність оцінювання параметра їх розміру на ранніх етапах розробки ПЗ шляхом застосування п'ятифакторних нелінійних регресійних моделей, які наведені в розділі 3 дисертації.

Інформаційна технологія [39, 40, 92, 93] для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків призначена для розв'язання таких:

- оцінювання середнього розміру JAVA-застосунків на ранніх стадіях розробки за допомогою п'ятифакторних нелінійних регресійних моделей (3.14) та (3.17) за метриками UML-діаграми класів: CLS, INFC, aVMQ, aTFQ та aCBO;
- оцінювання границь довірчого інтервалу середнього розміру JAVA-застосунків залежно від п'яти факторів CLS, INFC, aVMQ, aTFQ та aCBO для заданої довірчої ймовірності із використанням запропонованих моделей;
- оцінювання границь інтервалу передбачення середнього розміру JAVA-застосунків залежно від п'яти факторів CLS, INFC, aVMQ, aTFQ та aCBO для заданої довірчої ймовірності з використанням запропонованих моделей;
- реалізація програмного інтерфейсу для інтеграції ПЗ ІТ у сторонні програмні системи та комплекси.

Математичне забезпечення інформаційної технології

До математичного забезпечення ІТ для обробки інформації з метою оцінювання розміру JAVA-застосунків належать відповідні математичні моделі, побудовані на базі вибірок проєктів з відкритим кодом:

- п'ятифакторна нелінійна регресійна модель для оцінювання параметра розміру загальних JAVA-застосунків (3.14) на основі шестивимірною нормалізуючого перетворення Бокса-Кокса залежно від параметрів CLS, INFC, aVMQ, aTFQ та aCBO;

- п'ятифакторна нелінійна регресійна модель для оцінювання параметра розміру DS та ML JAVA-застосунків (3.17) на основі шестивимірного нормалізуючого перетворення Бокса-Кокса залежно від параметрів CLS, INFC, aVMQ, aTFQ та aCBO;
- рівняння нижньої та верхньої границь довірчого інтервалу та інтервалу передбачення (3.15) нелінійної регресії для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків;
- рівняння нижньої та верхньої границь довірчого інтервалу та інтервалу передбачення (3.18) нелінійної регресії для обробки інформації з метрик коду для оцінювання розміру DS та ML JAVA-застосунків;

Моделювання варіантів використання.

Користувачами комп'ютерної програми (ІТ) є інженери з розробки програмного забезпечення, DevOps-фахівці, науковці, менеджери проєктів та ін.

Для ІТ визначено дві 2 дійові особи – користувач та адміністратор. Визначені варіанти використання та дійові особи наведено в таблиці 4.1.

Таблиця 4.1 – Опис варіантів використання

Актори	Назва	Опис
1	2	3
Користувач, адміністратор	Отримати інформацію про математичні моделі	Користувач отримує список доступних математичних моделей, які можуть бути використані для обчислень, або інформацію про математичну модель за її ідентифікатором. До переліку параметрів входять – мова програмування, категорія математичної моделі або ідентифікатор математичної моделі в системі.

Продовження табл.4.1

1	2	3
Користувач, адміністратор	Обчислити значення залежного фактору за обраною математичною моделлю	Користувач вводить значення незалежних факторів математичної моделі, рівні значущості α для обчислення інтервалів передбачення та довірчих інтервалів. Система повертає результати обчислення.
Адміністратор	Додати нову математичну модель до системи	Адміністратор формує запит на додавання математичної моделі з інформацією, що включає її опис, параметри, перелік незалежних факторів, параметри обчислення інтервалів та обмеження математичної моделі.
Адміністратор	Редагувати наявну математичну модель	Адміністратор формує запит на редагування математичної моделі, що включає її опис, параметри, перелік незалежних факторів, параметри обчислення інтервалів та обмеження математичної моделі.
Адміністратор	Видалити наявну математичну модель із системи	Адміністратор формує запит на видалення математичної моделі вказавши її ідентифікатор в системі.
Адміністратор	Керування ключами доступу	Адміністратор переглядає, створює, редагує та видаляє ключі доступу до системи.

На рис. 4.1 представлена діаграма варіантів використання комп'ютерної програми «Сервіс для обробки інформації з метрик програмних застосунків».



Рисунок 4.1 – Діаграма варіантів використання комп’ютерної програми «Сервіс для обробки інформації з метрик програмних застосунків»

Специфікація варіантів використання.

1. Отримати інформацію про математичні моделі.

Короткий опис: Користувач отримує список доступних математичних моделей, які можуть бути використані для обчислень, або інформацію про математичну модель за її ідентифікатором. До переліку параметрів входять – мова програмування, категорія математичної моделі або ідентифікатор математичної моделі в системі.

Основні дійові особи: користувач, адміністратор.

Інші учасники прецеденту: відсутні.

Зв’язки з іншими варіантами використання: відсутні.

Передумови: користувач має чинний ключ доступу.

Основний потік:

- користувач надсилає запит на отримання інформації про математичні моделі;
- система перевіряє параметри запиту і виконує його;
- система формує список доступних математичних моделей у форматі JSON або JSON-об'єкт окремої моделі, якщо запит був за унікальним ідентифікатором моделі.

Постумови: відсутні.

Альтернативні потоки:

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо математична модель відсутня, система повертає інформацію користувачу про помилку (HTTP 404 Not Found);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

2. Обчислити значення залежного фактора за обраною математичною моделлю.

Короткий опис: Користувач вводить значення незалежних факторів математичної моделі, рівні значущості α для обчислення інтервалів передбачення та довірчих інтервалів. Система повертає результати обчислення.

Основні дійові особи: користувач, адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Передумови:

- користувач має чинний ключ доступу;
- обрано математична модель для обчислення.

Основний потік:

- користувач вибирає ідентифікатор математичної моделі для запиту;
- користувач надсилає вхідні дані для обчислення залежного фактору;

- система перевіряє параметри запиту і виконує обчислення залежного фактору, а також інтервалів передбачення та довірчих інтервалів;
- система формує результат у форматі JSON.

Постумови: відсутні.

Альтернативні потоки:

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

3. Додати нову математичну модель до системи

Короткий опис: Адміністратор формує запит на додавання математичної моделі з інформацією, що містить її опис, параметри, перелік незалежних факторів та параметри обчислення інтервалів.

Основні дійові особи: Адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Передумови: адміністратор має чинний ключ доступу.

Основний потік:

- адміністратор надсилає запит з інформацією про нову математичну модель;
- система перевіряє коректність вхідних даних;
- система додає модель до списку доступних моделей.

Постумови: відсутні.

Альтернативні потоки:

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

4. Редагувати наявну математичну модель

Короткий опис: Адміністратор формує запит на редагування математичної моделі, що містить її опис, параметри, перелік незалежних факторів та параметри обчислення інтервалів.

Основні дійові особи: Адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Передумови:

- адміністратор має чинний ключ доступу;
- математична модель була раніше додана до системи та має унікальний ідентифікатор.

Основний потік:

- адміністратор надсилає запит з оновленою інформацією про нову математичну модель, вказавши її ідентифікатор;
- система перевіряє коректність вхідних даних;
- система зберігає зміни існуючої математичної моделі.

Постумови: відсутні.

Альтернативні потоки:

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо математична модель відсутня, система повертає інформацію користувачу про помилку (HTTP 404 Not Found);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

5. Видалити наявну математичну модель із системи

Короткий опис: Адміністратор формує запит на видалення математичної моделі вказавши її ідентифікатор в системі.

Основні дійові особи: Адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Передумови:

- адміністратор має чинний ключ доступу;
- математична модель була раніше додано до системи та має унікальний ідентифікатор.

Основний потік:

- адміністратор надсилає запит на видалення математичної моделі за її ідентифікатором;
- система перевіряє коректність вхідних даних;
- система видаляє математичну модель.

Постумови: відсутні.**Альтернативні потоки:**

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо математична модель відсутня, система повертає інформацію користувачу про помилку (HTTP 404 Not Found);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

6. Керування ключами доступу.

Короткий опис: Адміністратор переглядає, створює, редагує та видаляє ключі доступу до системи.

Основні дійові особи: Адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Передумови: адміністратор має чинний ключ доступу;

Основний потік:

- адміністратор надсилає запит на перегляд, створення, редагування або видалення ключа доступу;
- система перевіряє коректність вхідних даних.

Постумови: відсутні.

Альтернативні потоки:

- якщо запит або вхідні дані некоректні, система повертає інформацію користувачу про помилку (HTTP 400 Bad Request);
- якщо ключ доступу відсутній, система повертає інформацію користувачу про помилку (HTTP 404 Not Found);
- якщо користувач не авторизований, система повертає інформацію користувачу про помилку (HTTP 401 Unauthorized).

Інформаційне забезпечення інформаційної технології.

До інформаційного забезпечення ІТ для обробки інформації з метрик JAVA-застосунків входить база даних для збереження параметрів нелінійних регресійних моделей для оцінювання розміру програмних застосунків та параметрів оцінки границь довірчих інтервалів та інтервалів їх передбачення.

Логічна модель даних математичної моделі з урахуванням різних мов програмування та категорій програмних застосунків має такий перелік атрибутів:

- унікальний ідентифікатор запису – первинний ключ;
- мова програмування – рядковий тип даних;
- категорія – рядковий тип даних;
- назва математичної моделі – рядковий тип даних;
- опис математичної моделі – рядковий тип даних;
- кількість незалежних факторів моделі – числовий тип даних;
- перелік незалежних факторів моделі – рядковий тип даних;
- назва залежного фактора моделі – рядковий тип даних;
- час та дата публікації моделі;
- посилання на першоджерело;
- параметри математичної моделі – рядковий тип даних;
- інформація щодо точності та випробувань математичної моделі – рядковий тип даних.

Для доступу до бази даних використовується система управління базами даних PostgreSQL. Вона має такі переваги [94, 95]:

- безкоштовно розповсюджується за ліцензією PostgreSQL License, що робить її доступною для використання в освітніх проєктах і дає змогу знизити витрати на ПЗ;
- відзначається своєю надійністю та продуктивністю;
- має розширені можливості для підтримки різноманітних даних, зокрема JSON;
- забезпечує гнучкість та можливості масштабування для малих та великих проєктів.

На наступному етапі побудовано фізичну модель бази даних для обробки інформації з метрик JAVA-застосунків для оцінювання параметра розміру програмних застосунків із урахуванням логічної моделі бази даних та обраної СУБД PostgreSQL. Фізична модель має таку структуру:

- guid – VARCHAR PRIMARY KEY;
- programming_language – VARCHAR NOT NULL;
- category – VARCHAR NOT NULL;
- name – VARCHAR NOT NULL;
- description – TEXT NOT NULL;
- factors_quantity – INT NOT NULL
- factors_list – TEXT NOT NULL;
- factor_estimating VARCHAR NOT NULL,
- model_json – TEXT NOT NULL;
- accuracy_json TEXT;
- restrictions_json TEXT;
- published_at – TIMESTAMP;
- published_reference – TEXT;
- created_at – TIMESTAMP NOT NULL;
- updated_at – TIMESTAMP NOT NULL.

Програмне забезпечення інформаційної технології

ПЗ інформаційної технології для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків є веб-застосунком на базі Spring Framework [96], зокрема його варіанту Spring Boot [97].

Для веб-застосунку обрано модель розповсюдження software as a service (SaaS) [98, 99]. Її перевагами є те, що вона надає легкий доступ до обчислення розміру JAVA-застосунків; забезпечує швидке оновлення, обслуговування та вдосконалення веб-застосунку без втручання з боку користувачів та дозволяє отримати миттєвий доступ до нових математичних моделей, програмних функцій, покращень безпеки і виправлень помилок; пропонує API для інтеграції обчислення оцінювання KLOC в програмний інструментарій клієнтів та існуючі системи інших сервісів. Такий підхід забезпечує високу сумісність і гнучкість та надає змогу організаціям і компаніям використовувати інструменти обчислення як частину власного IT-середовища, що сприяє підвищенню ефективності аналітичної оцінки розробки проєктів ПЗ.

В дисертаційній роботі розроблено веб-застосунок, як програмний засіб IT для підвищення достовірності обробки інформації з метрик коду для оцінювання KLOC JAVA-застосунків на ранніх стадіях розробки за метриками діаграми класів із використанням побудованих п'ятифакторних нелінійних регресійних моделей.

Як мову програмування для розробки IT обрано мову програмування Kotlin з фреймворком Spring Framework для побудови мережевої взаємодії. Kotlin є високорівневою об'єктно-орієнтованою мовою програмування, яка відрізняється своєю швидкодією та продуктивністю, є повністю сумісною з мовою програмування JAVA. Використання Spring Framework має такі переваги:

- широкий набір інструментів для реалізації мережевого функціоналу та підтримка REST API;
- швидкий запуск та автоматичне конфігурування, що зменшує обсяг ручного налаштування для різних компонентів мережевої взаємодії;

- інтеграція з базами даних, яка спрощує взаємодію з різними СУБД та автоматична конфігурація моделей бази даних в Kotlin-кодi;
- підтримка для роботи з JSON-даними;
- безкоштовна модель розповсюдження;
- контроль доступу до ресурсів із підтримкою таких стандартів, як OAuth2 і JWT (JSON Web Tokens), що особливо важливо для таких мережових застосунків як SaaS, де безпека даних має критичне значення.

До веб-застосунку для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків доступ надається через ключі доступу. Веб-застосунок не зберігає стан або проміжні результати обчислення розміру JAVA-застосунків [100].

Для SaaS веб-застосунку побудовано UML-діаграму класів (рис. 4.2), на якій представлено основні сутності системи, їх зв'язки та взаємодія між ними.

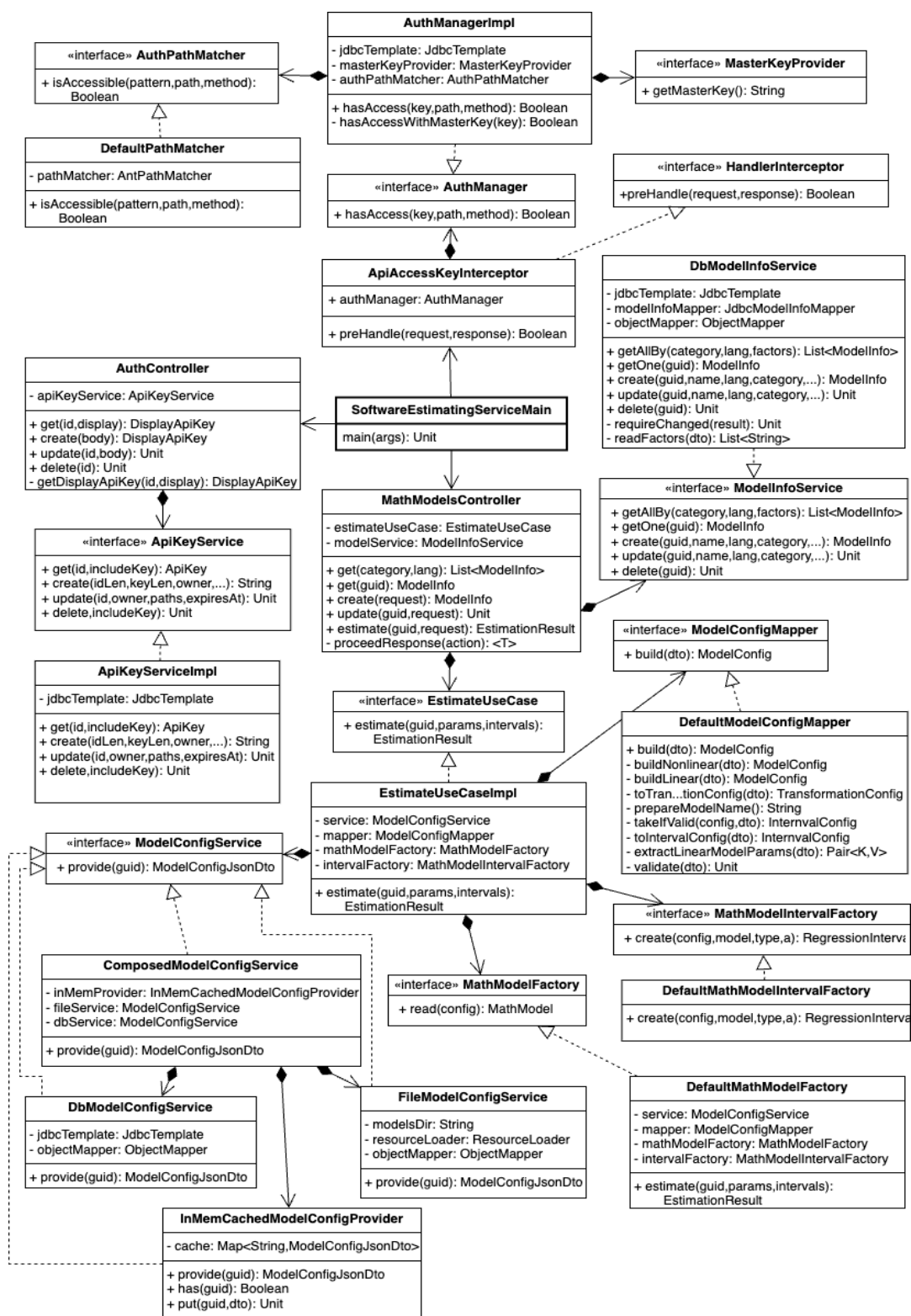


Рисунок 4.2 – Діаграма класів SaaS веб-застосунку для обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру

Технічне забезпечення інформаційної технології.

Вимоги до апаратного середовища виконання:

Процесор: мінімальна тактова частота 1 ГГц, рекомендована тактова частота 2 ГГц або вище.

Оперативна пам'ять: мінімальний обсяг 512 МВ, рекомендований 1 ГБ та вище.

Дисковий простір: від 500 МБ вільного місця.

Наявність підключення до мережі Інтернет.

Апаратне та програмне забезпечення технічного засобу користувача має забезпечувати можливість здійснення запитів за протоколом HTTPS (підтримка SSL/TLS) через мережу Інтернет [101, 102].

Програмне забезпечення, необхідне для функціонування інформаційної технології.

Вимоги до операційної системи [103]:

Windows: Windows 10, Windows 11, Windows Server 2016, 2019, 2022;

Linux: дистрибутиви, що підтримують glibc 2.17+ (наприклад, Ubuntu 18.04+, CentOS 7+, RHEL 7+);

Mac OS: macOS 10.15 (Catalina) і новіші.

Вимоги до середовища виконання [103]:

Java Runtime Environment 11 або вище;

СУБД PostgreSQL версії 14 або вище;

OpenSSL версії 1.1.1 або вище.

Методичне забезпечення інформаційної технології.

Методичним забезпеченням ІТ веб-застосунку для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків є методика використання прикладного програмного інтерфейсу (API) до веб-застосунку, яка наведена в пп. 4.2 дисертаційної роботи. Користувачам надається документація з описом вхідних і вихідних параметрів, методів прикладного програмного інтерфейсу (API) та алгоритму доступу до обчислювальних ресурсів у вигляді інструкції користувача – Додаток Є.

Впровадження інформаційної технології.

ІТ для обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків було впроваджено у виробничі процеси ТОВ «Н-ІКС ДЕЛІВЕРІ» та ФОП Поліщук Андрій Борисович (акти впровадження наведені у додатку Г). Впровадження запропонованої ІТ для оцінювання розміру JAVA-застосунків дало змогу підвищити достовірність обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків на ранніх стадіях проектування ПЗ.

4.2 Методика обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків за допомогою п'ятифакторних нелінійних регресійних моделей

Методика обробки інформації з метрик коду для оцінювання розміру JAVA-застосунків розрахована на використання відповідного ПЗ, програмних рішень на стороні клієнта, наявність ключів доступу до обчислювальних потужностей веб-застосунку для оцінювання інформації з метрик JAVA-застосунків, розробленого у межах цієї роботи та доступу до мережі Інтернет.

Доступ до ІТ здійснюється за веб-адресою та протоколом HTTP [104, 105].

Для забезпечення безпеки доступу до сервісу всі запити потребують аутентифікації та авторизації. Авторизовані запити до ІТ здійснюються за допомогою ключа доступу, який додається до кожного HTTP-запиту в секції заголовка. Приклад заголовка авторизації наведено у лістингу нижче:

X-Api-Key: <Api Key> ,

де «X-Api-Key» – ім'я параметра заголовка та «<Api Key>» – значення ключа доступу.

Обробка інформації з метрик коду JAVA-застосунків для оцінювання розміру складається з двох етапів. На першому етапі необхідно отримати список відповідних математичних моделей та обрати модель з їх незалежними

параметрами за її ідентифікатором. На другому етапі необхідно передати значення незалежних параметрів для обчислення залежного параметра математичної моделі, інтервалу передбачення та довірчого інтервалу. Перший етап є необов'язковим, оскільки, за наявності відомого ідентифікатора моделі, можна почати одразу з другого етапу – оцінки інформації з метрик.

Опис першого етапу.

На першому етапі користувач отримує список відповідних математичних моделей з їх незалежними параметрами для обробки інформації з метрик коду для оцінювання KLOC JAVA-застосунків. Для цього необхідно виконати GET HTTP-запит [106] до списку всіх наявних математичних моделей вказавши категорію для оцінювання параметра розміру – `category=software-size` та параметру мови програмування JAVA – `lang=JAVA`. Шлях веб запиту має наступне значення `<Base Url>/api/v1/math-models`. Приклад запиту із застосуванням утиліти CURL [107] не наведено в Лістингу 4.1. Основне призначення CURL – це здійснення HTTP/HTTPS-запитів і отримання відповідей без необхідності використовувати браузер.

Лістинг 4.1. CURL команда HTTP-запиту на отримання списку математичних моделей оцінки розміру JAVA-застосунків

```
curl -X GET \
<Base Url>/api/v1/math-models?category=software-size&lang=JAVA \
-H "X-API-Key: <Api Key>" \
-H "Accept: application/json"
```

Результатом обробки запиту IT буде список математичних моделей для обробки інформації з метрик коду для оцінювання KLOC JAVA-застосунків у форматі JSON (Лістинг 4.2).

Лістинг 4.2. Результат обробки HTTP-запиту на отримання списку математичних моделей оцінювання інформації з метрик коду JAVA-застосунків:

```
[
  {
    "guid": "esti-kloc-5x-cls-infc-p-1-avmq-atfq-acbo",
    "name": "Five-factor nonlinear regression model for KLOC
estimation of Java software",
    "lang": "JAVA",
    "description": "Five-factor nonlinear regression model for
KLOC estimation of Java software depends on factors total classes,
total interfaces, average VMQ, average TFQ, average CBO on basis
of Box-Cox transformation",
    "category": "software-size",
    "factorsTotal": 5,
    "factors": [
      "cls_class",
      "ifc_class",
      "avg_visibleMethodsQty",
      "avg_totalFieldsQty",
      "avg_cbo"
    ],
    "factorEstimating": "kloc",
    "publishedAt": null,
    "publishedReference": null,
    "createdAt": "2025-03-17",
    "updatedAt": "2025-03-17",
    "accuracy": [
      {
        "rsq": 0,9647,
        "mmre": 0,1305,
        "preds": {
          "0.25": 0,8927
        },
        "sampleSize": 261,
        "sample": "training-optimized"
      },
      {
        "rsq": 0,9294,
        "mmre": 0,1540,
        "preds": {
          "0.25": 0,8462
        },
        "sampleSize": 286,
        "sample": "training-raw"
      },
      {
        "rsq": 0,9193,
        "mmre": 0,1742,
        "preds": {
          "0.25": 0,7825
        },
        "sampleSize": 285,
        "sample": "testing"
      }
    ]
  },
]
```

```

    "restrictions": null
  }
]

```

Кожна структура даних опису математичної моделі є елементом масиву та містить такі поля:

- "guid" – унікальний ідентифікатор моделі;
- "name" – назва математичної моделі;
- "category" – категорія математичної моделі;
- "lang" – мова програмування, для якої було розроблено математичну модель;
- "factorsTotal" – загальна кількість незалежних факторів моделі;
- "factors" – список незалежних факторів математичної моделі;
- "factorEstimating" – залежний фактор математичної моделі;
- "createdAt" – дата та час створення математичної моделі в базі даних;
- "updatedAt" – дата та час останнього оновлення математичної моделі в базі даних;
- "accuracy" – оцінки критеріїв якості нелінійних регресійних моделей, таких як R^2 , $MMRE$ та $PRED(0, 25)$ та характеристики наборів даних, на яких ці оцінки були отримані;

"restrictions" – поле містить перелік обмежень математичної моделі та список допустимих значень параметра рівня значущості α для інтервалів передбачення ("alphaIntervalPrediction"), довірчих інтервалів ("alphaIntervalTrust"), обмеження діапазонів значень незалежних факторів моделі.

Для структури даних опису математичної моделі поля "guid", "name", "lang", "description", "category", "factorsTotal", "factors", "factorEstimating", "createdAt", "updatedAt", "accuracy" є обов'язковими.

Якщо ідентифікатор та найменування незалежних факторів існуючої математичної моделі відомі, то перший крок можна пропустити.

Опис другого етапу.

Необхідно виконати PUT HTTP-запит [106] для оцінювання кількості рядків коду, вказавши в шляху запиту ідентифікатор **guid** відповідної математичної моделі – `<BaseUrl>/api/v1/math-models/<guid>/estimate`. (Наприклад: `<BaseUrl>/api/v1/math-models/esti-kloc-5x-cls-ifc-p-1-avmq-atfq-acbo/estimate`.), а в блоці тіла запиту (Body) [106] вказати значення незалежних факторів математичної моделі, значення рівня значущості α для довірчого інтервалу та інтервалу передбачення. Якщо параметри рівня значущості α не вказані, то обчислення інтервалу передбачення та довірчого інтервалу відбуватись не буде. Приклад запиту обчислення значення оцінки розміру JAVA-застосунку наведено в Лістингу 4.3.

Лістинг 4.3. Приклад запиту обчислення значення розміру JAVA застосунку за допомогою математичної моделі з ідентифікатором `<guid>`:

```
curl --request PUT \
  --url '<BaseUrl>/api/v1/math-models/<guid>/estimate' \
  --header 'Content-Type: application/json' \
  --header 'X-API-Key: <Api Key>' \
  --data '{
    "factors": {
      "cls_class": 382,
      "ifc_class": 46,
      "avg_visibleMethodsQty": 6.28805606,
      "avg_totalFieldsQty": 2.606557377,
      "avg_cbo": 6.1264637
    },
    "interval_trust": [
      "0.01"
    ],
    "interval_prediction": [
      "0.01"
    ]
  }'
```

У запиті, "factors" – значення незалежних факторів моделі у форматі ключ-значення, ключ – рядкове значення існуючого незалежного фактору

моделі та значення – числова величина незалежного фактора, список незалежних факторів моделі отримується за допомогою запиту з Лістингу 4.1 або по ідентифікатору моделі відповідно до інструкції; "interval_trust" та "interval_prediction" – перелік рівнів значущості α , для яких необхідно обчислити значення довірчих інтервалів ("interval_trust") та інтервалів передбачення ("interval_prediction").

Якщо значення введено коректно, то веб-застосунок поверне код 200 для HTTP-запиту, а результати обчислення буде наведено в блоці BODY відповіді на HTTP-запит у форматі JSON (Лістинг 4.4).

Лістинг 4.4. Результат обробки HTTP-запиту обробки інформації за допомогою п'ятифакторної математичної моделі (3.14) за ідентифікатором <guid>:

```
{
  "factors": {
    "cls_class": 382.0,
    "ifc_class": 46.0,
    "avg_visibleMethodsQty": 6.28805606,
    "avg_totalFieldsQty": 2.606557377,
    "avg_cbo": 6.1264637
  },
  "result": 25.988340996260465,
  "resultName": "kLoc",
  "intervalsTrust": [
    {
      "a": "0.01",
      "lb": 24.872436807468482,
      "ub": 27.153661007798398
    }
  ],
  "intervalsPrediction": [
    {
      "a": "0.01",
      "lb": 16.992521079341,
      "ub": 39.657978950490325
    }
  ],
  "message": null
}
```

У відповіді, "factors" – значення незалежних факторів моделі у форматі ключ-значення відповідно до Лістингу 4.3; "result" – результати оцінювання

за допомогою математичної моделі `<guid>` та значень вхідних параметрів `"factors"`; `"resultName"` – назва залежного фактора, оцінка якого отримана в полі `"result"`; `"intervalsTrust"` та `"intervalsPrediction"` – значення довірчих інтервалів та інтервалів передбачення відповідно, для яких `"a"` – рівень значущості α , `"lb"` – значення нижньої границі інтервалу, `"ub"` – значення верхньої границі інтервалу; `"message"` – поле, яке може містити інформацію щодо результатів оцінювання або попередження.

ІТ повертає HTTP статус-коди відповідно до результатів обробки запиту та специфікації протоколу HTTP [105]:

200 OK – Запит успішно виконано;

400 Bad Request – Некоректні параметри запиту (відсутні вхідні параметри або в неправильному форматі);

401 Unauthorized – Відсутній або неправильний ключ доступу;

403 Forbidden – Відсутні права доступу;

404 Not Found – Об'єкт не знайдено;

500 Internal Server Error – Внутрішня помилка сервера.

На UML-діаграмі діяльності (рис. 4.3) представлено деталізацію сценаріїв варіантів використання “Отримати інформацію про математичні моделі” та “Обчислити значення залежного фактора за обраною математичною моделлю”.



Рисунок 4.3 – Діаграма діяльності основних варіантів використання.

Основні модулі вихідного коду ІТ «Сервіс для обробки інформації з метрик програмних застосунків» наведені в додатку Ж

4.3 Висновки за розділом 4

В четвертому розділі дисертації створено ІТ обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, із застосуванням

п'ятифакторних нелінійних регресійних моделей (3.14), (3.17) на основі шестивимірних нормалізуючих перетворень Бокса-Кокса за метриками CLS, INFC, aVMQ, aTFQ та aCBO.

1) Створена інформаційна технологія для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків за п'ятьма метриками діаграми класів (CLS, INFC, aVMQ, aTFQ та aCBO) на базі шестивимірних нормалізуючих перетворень Бокса-Кокса дає змогу підвищити достовірність оцінювання розміру JAVA-застосунків на ранніх стадіях розробки за відповідними напрямками.

2) Розроблено методику процесу обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків, яка розрахована на використання відповідного інструментарію ІТ – розробленого веб-застосунку.

3) Інформаційну технологію для обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків було впроваджено в виробничі процеси ТОВ «Н-ІКС ДЕЛІВЕРІ» та ФОП Поліщук Андрій Борисович. Впровадження запропонованої ІТ дало змогу підвищити якість оцінювання параметра розміру JAVA-застосунків на ранніх етапах розробки та оптимізувати бюджети проєктів для подальшої створення ПЗ.

4) Математичне забезпечення та інструментарій інформаційної технології для оцінювання інформації з метрик коду JAVA-застосунків використовуються під час викладання студентам Навчально-наукового інституту комп'ютерних наук та управління проєктами Національного університету кораблебудування імені адмірала Макарова таких дисциплін: «Керування проєктними ресурсами в інженерії програмного забезпечення» в ОП «Інженерія програмного забезпечення» для 2-го магістерського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»; «Основи програмної інженерії» в ОП «Інженерія програмного забезпечення» для 1-го бакалаврського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»; «Моделювання систем цифрового екологічного моніторингу» в ОП

«Цифровий екологічний моніторинг» для 2-го магістерського рівня ВО, спеціальності Е2 «Екологія», галузі знань Е «Природничі науки, математика та статистика». Акти впровадження результатів дисертаційної роботи наведені в додатках Г та Д.

Результати досліджень поточного розділу були опубліковані автором в роботах [37-40, 92, 93].

ВИСНОВКИ

В дисертаційній роботі вирішено важливе науково-практичне завдання підвищення достовірності обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків на ранніх стадіях розробки шляхом побудови відповідних п'ятифакторних нелінійних регресійних моделей із використанням шестивимірних нормалізуючих перетворень Бокса-Кокса за метриками UML-діаграми класів CLS, INFC, aVMQ, aTFQ та aCBO. Для практичної реалізації математичних моделей обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків створено інструментарій інформаційної технології.

1) Проведено огляд та аналіз існуючих лінійних регресійних рівнянь та нелінійних регресійних моделей обробки інформації з метрик коду для оцінювання параметра розміру JAVA-застосунків. Виявлено, що більшість існуючих регресійних рівнянь та моделей побудовані на застарілих даних або вибірках невеликого розміру без належної валідації на незалежних даних. Результати перевірки нелінійних регресійних моделей на навчальній та тестовій вибірках за критеріями R^2 , $MMRE$ та $PRED(0, 25)$ показали їхню непридатність для обробки інформації для оцінювання розміру сучасних JAVA-застосунків (зокрема таких, що враховують технологічні особливості розробки DS та ML застосунків) через низький рівень достовірності або обмеженість в охопленні актуального діапазону даних. Таким чином, обґрунтовано необхідність розробки відповідних математичних моделей для підвищення достовірності обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків.

2) Для підвищення достовірності обробки інформації з метрик коду для оцінювання параметра розміру загальних JAVA-застосунків зібрано набір даних обсягом 571 векторів характеристик, який було використано для удосконалення чотирьохфакторної нелінійної регресійної моделі та побудови

п'ятифакторної нелінійної регресійної моделі. Отриманий набір даних загальних JAVA-застосунків розділено на навчальну та тестову вибірки розміром 286 та 285 точок даних відповідно.

3) Для підвищення достовірності обробки інформації з метрик коду для оцінювання параметра розміру DS та ML JAVA-застосунків зібрано набір даних розміром 112 векторів характеристик, який було використано для побудови п'ятифакторної нелінійної регресійної моделі. Отриманий набір даних DS та ML JAVA-застосунків розділено на навчальну та тестову вибірки розмірами 74 та 38 точок даних відповідно.

4) Проведено обґрунтування вибору незалежних факторів серед метрик коду JAVA-застосунків для побудови відповідних п'ятифакторних нелінійних регресійних моделей з урахуванням мультиколінеарності. Як незалежні фактори обрано метрики CLS, INFC, aVMQ, aTFQ та aCBO.

5) Здійснено вибір перетворення для нормалізації шестивимірних векторів характеристик інформації з метрик коду загальних JAVA-застосунків для побудови п'ятифакторної нелінійної регресійної моделі. У якості нормалізуючого перетворення обрано багатовимірне перетворення Бокса-Кокса, за допомогою якого краще здійснюється нормалізація багатовимірних даних з метрик коду загальних JAVA-застосунків у порівнянні з одновимірними перетвореннями десяткового логарифма, Бокса-Кокса, Джонсона сім'ї SB та багатовимірним перетворенням Джонсона сім'ї SB.

6) Здійснено вибір перетворення для нормалізації шестивимірних векторів характеристик інформації з метрик коду DS та ML JAVA-застосунків для побудови п'ятифакторної нелінійної регресійної моделі. У якості нормалізуючого перетворення обрано шестивимірне перетворення Бокса-Кокса, оскільки воно показало кращу нормалізацію багатовимірних даних з метрик коду DS та ML JAVA-застосунків у порівнянні з одновимірним перетворенням Бокса-Кокса.

7) Удосконалено чотирьохфакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру KLOC

загальних JAVA-застосунків із використанням метрик CLASS, RFC, аCBO та аVMQ на основі п'ятивимірного нормалізуючого перетворення Бокса-Кокса. Надійність моделі підтверджено за допомогою збалансованого поділу набору даних загальних JAVA-застосунків на навчальну та тестову вибірки обсягом 286 та 285 точок даних. Модель, що побудовано, в порівнянні з іншими регресійними моделями, має кращі показники критеріїв якості обробки інформації R^2 , $MMRE$ та $PRED(0, 25)$ з метрик коду для оцінювання параметра розміру.

8) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру KLOC загальних JAVA-застосунків на базі метрик CLS, INFC, аVMQ, аTFQ та аCBO із використанням шестивимірного перетворення Бокса-Кокса. Заміна метрики CLASS на окремі метрики CLS та INFC, у порівнянні з існуючими нелінійними регресійними моделями, та збалансований поділ навчальної та тестової вибірок обсягом 286 та 285 спостережень забезпечують кращу надійність перевірки отриманої математичної моделі та достовірність оцінювання KLOC за критеріями якості R^2 , $MMRE$ та $PRED(0, 25)$.

9) Вперше побудовано п'ятифакторну нелінійну регресійну модель для обробки інформації з метрик коду для оцінювання параметра розміру KLOC DS та ML JAVA-застосунків на базі метрик CLS, INFC, аVMQ, аTFQ та аCBO із використанням шестивимірного перетворення Бокса-Кокса. Отримана п'ятифакторна нелінійна регресійна модель, у порівнянні з існуючими нелінійними регресійними моделями, дає змогу врахувати технологічні фактори середовища та специфіку розробки ПЗ за цим напрямом. Модель показала кращий рівень достовірності обробки інформації для оцінювання параметра розміру за критеріями якості R^2 , $MMRE$ та $PRED(0, 25)$ у порівнянні з існуючими математичними моделями і отриманою п'ятифакторною нелінійною регресійною моделлю на основі вибірки загальних JAVA-застосунків.

10) Отримали подальший розвиток рівняння для визначення границь інтервалу передбачення та довірчого інтервалу п'ятифакторної нелінійної регресійної моделі для загальних JAVA-застосунків та п'ятифакторної нелінійної регресійної моделі DS та ML JAVA-застосунків. Виявлено, що ширина інтервалів передбачення для побудованих п'ятифакторних нелінійних регресійних моделей вужча у порівнянні з іншими нелінійними регресійними моделями. Це дало змогу підвищити достовірність обробки інформації для оцінювання параметра розміру як залежної випадкової величини.

11) Створена інформаційна технологія та розроблено методику для обробки інформації з метрик коду для оцінювання розміру загальних JAVA-застосунків, DS та ML JAVA-застосунків за п'ятьма метриками діаграми класів (CLS, INFC, aVMQ, aTFQ та aCBO) на базі шестивимірних нормалізуючих перетворень Бокса-Кокса.

12) Результати досліджень впроваджено в виробничі процеси ТОВ «Н-ІКС ДЕЛІВЕРІ» та ФОП Поліщук Андрій Борисович, що підтверджується актами, які наведені у додатку Г. Результати дисертаційної роботи впроваджено у навчальний процес Навчально-наукового інституту комп'ютерних наук та управління проєктами Національного університету кораблебудування імені адмірала Макарова під час викладання дисциплін «Керування проєктними ресурсами в інженерії програмного забезпечення», «Основи програмної інженерії» та «Моделювання систем цифрового екологічного моніторингу», що підтверджується актом, який наведений у додатку Д.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D., & Steece, B. (2000). *Software cost estimation with COCOMO II*. Prentice Hall.
- [2] Jorgensen, M., & Shepperd, M. (2007). A systematic review of software development effort estimation studies. *IEEE Transactions on Software Engineering*, 33(1), 33–53. <https://doi.org/10.1109/TSE.2007.256943>
- [3] McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press.
- [4] Johnson, J., & Mulder, H. (2021, січень). *Endless modernization: How infinite flow keeps software fresh*. ResearchGate. https://www.researchgate.net/publication/348849361_Endless_Modernization_How_Infinite_Flow_Keeps_Software_Fresh
- [5] The Standish Group. (2015). *Chaos report 2015*. https://standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- [6] Munialo, S. W., & Muketha, G. M. (2016). A review of Agile Software effort estimation methods. *International Journal of Computer Applications Technology and Research*, 5(9), 612–618. <https://doi.org/10.7753/ijcatr0509.1009>
- [7] Iqbal, S. Z. (2017). Comparative analysis of common software cost estimation modeling techniques. *Mathematical Modelling and Applications*, 2(3), 33–39. <https://doi.org/10.11648/j.mma.20170203.12>
- [8] Shekhar, S., & Kumar, U. (2016). Review of various software cost estimation techniques. *International Journal of Computer Applications*, 141 (11), 31–34. <https://doi.org/10.5120/ijca2016909867>
- [9] Ворона, М. В., & Орехов, О. С. (2023). Методи оцінки трудомісткості розробки програмного забезпечення для agile проєктів. *Матеріали XIV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (20-21 вересня 2023). Миколаїв: НУК імені адмірала Макарова. 438-441.

- [10] Фаріонова Т., & Орехов О. (2023). Аналіз успішності розробки проєктів програмного забезпечення. *Матеріали міжнародної науково-практичної конференції "Інтелектуальні інформаційні системи в управлінні проєктами та програмами"* (12-15 вересня 2023) Харків: ХНУРЕ. 210–213. <https://doi.org/10.30837/IISRRM.2023.09>
- [11] Фаріонова Т. А., Орехов О. С. (2024), Аналіз сучасного стану методів оцінювання трудомісткості розробки програмного забезпечення. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 1(494), 102–111. [https://doi.org/10.15589/znp2024.1\(494\).15](https://doi.org/10.15589/znp2024.1(494).15)
- [12] Oracle. (2025, November 10). Java. Retrieved from <https://www.oracle.com/java/>
- [13] Cass, S. (2024, August 22). Top programming languages 2024. IEEE Spectrum. <https://spectrum.ieee.org/top-programming-languages-2024>
- [14] Jansen, P. (2025, November 11). TIOBE index of programming languages. TIOBE. <https://www.tiobe.com/tiobe-index/>
- [15] Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-oriented analysis and design with applications* (3rd ed.). Addison-Wesley Professional.
- [16] Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language user guide* (2nd ed.). Addison-Wesley Professional.
- [17] Bloch, J. (2017). *Effective Java* (3rd ed.). Addison-Wesley Professional.
- [18] Lindholm, T., Yellin, F., Bracha, G., & Buckley, A. (Eds.). (2024). *The Java Virtual Machine Specification, Java SE 22 Edition*. Oracle. <https://docs.oracle.com/javase/specs/jvms/se22/jvms22.pdf>
- [19] Gherardi, L., Brugali, D., & Comotti, D. (2012). A Java vs. C++ performance evaluation: A 3D modeling benchmark. In N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, & O. von Stryk (Eds.), *Simulation, modeling, and programming for autonomous robots (SIMPAN 2012)*. Vol. 7628. pp. 325–336. Springer. https://doi.org/10.1007/978-3-642-34327-8_17

- [20] Fortune Business Insights. (2024). Machine learning (ML) market size, share & COVID-19 impact analysis, by component (solution, and services), by application (enterprise resource planning, supply chain management, customer relationship management, and others), by enterprise size (large enterprises, and small & medium enterprises), by end-user (BFSI, healthcare, retail, manufacturing, IT, automotive, and others), and regional forecast, 2024-2032. <https://www.fortunebusinessinsights.com/machine-learning-market-102226>
- [21] Chinthamu, N., & Karukuri, M. (2023). Data science and applications. *Journal of Data Science and Intelligent Systems*, 1(2), 8–15. <https://doi.org/10.47852/bonviewJDSIS3202837>
- [22] Tan, H. B. K., Zhao, Y., & Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models. In *Proceedings of the 28th International Conference on Software Engineering* (pp. 321–330). Association for Computing Machinery. <https://doi.org/10.1145/1134285.1134331>
- [23] Tan, H. B. K., Zhao, Y., & Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. *ACM Transactions of Software Engineering and Methodology*, 19(2), 1-37. <https://doi.org/10.1145/1571629.1571630>
- [24] Приходько, Н. В., & Приходько, С. Б. (2018). Нелінійна регресійна модель для оцінювання розміру програмного забезпечення промислових інформаційних систем на Java. *Моделювання та інформаційні технології*, 85, 81–88. http://nbuv.gov.ua/UJRN/Mtit_2018_85_14
- [25] Макарова, Л. М., Приходько, Н. В., & Кудін, О. О. (2019). Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою Java. *Вісник Херсонського національного технічного університету*, 69(2), 145–153. <http://eir.nuos.edu.ua/handle/123456789/4443>
- [26] Приходько, С. Б., Приходько, Н. В., & Смикодуб, Т. Г. (2020). Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру JAVA-застосунків з відкритим кодом. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*, 31(2-1), 157–162. <https://doi.org/10.32838/2663-5941/2020.2-1/25>

- [27] **Орєхов, О. С., & Фаріонова, Т. А.** (2024). Математичні моделі для оцінювання розміру JAVA-застосунків. *Вісник Херсонського національного технічного університету*, 89(2), 196–203. <https://doi.org/10.35546/kntu2078-4481.2024.2.28>
- [28] **Oriekhov, O., Farionova, T., Chernova, L., Chernova, L., & Vorona, M.** (2024). Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications. In *Proceedings of V International Workshop on IT Project Management, Bratislava (ITPM-2024)*, Slovakia, May 22, 2024. CEUR Workshop Proceedings, Vol. 3709, pp. 54-66. https://doi.org/10.23939/IW_itpm2024.054
- [29] **Орєхов, О. С., & Фаріонова, Т. А.** (2023). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Data Science та machine learning проєктів, які розробляються за допомогою мови програмування Java. *Матеріали IV всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2023)* (30-31 жовтня 2023 р.). Миколаїв: НУК імені адмірала Макарова. 45–47.
- [30] **Oriekhov, O., Farionova, T., & Chernova, L.** (2024), Three-factor nonlinear regression model of estimating the size of JAVA-software. In *Proceedings of the International Conference on Information Control Systems and Technologies (ICST-2024)*, September 23-25, 2024. CEUR Workshop Proceedings. pp. 506-518. <https://ceur-ws.org/Vol-3790/paper44.pdf>
- [31] **Oriekhov, O., & Farionova, T.** (2024). Early KLOC estimation of JAVA-applications by the three-factor nonlinear regression model. In V. Vychuzhanin (Ed.), *Materials of the XI International Scientific and Practical Conference "Information Control Systems and Technologies 2024" (ICST-2024)* (23-25 September). Odesa: Helvetica Publishing House. pp. 310 – 312. <https://icst-conf.com/2024.pdf>
- [32] **Орєхов, О. С., & Фаріонова, Т. А.** (2024). Оцінювання розміру JAVA-застосунків за трьохфакторною нелінійною регресійною моделлю. *Матеріали V всеукраїнської науково-практичної інтернет конференції*

"Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2024) (30-31 жовтня 2024 р.). Миколаїв: НУК імені адмірала Макарова. 39–42.

[33] Port, D., & Korte, M. (2008). Comparative studies of the model evaluation criteria MMRE and PRED in software cost estimation research. In *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 51–60). Association for Computing Machinery. <https://doi.org/10.1145/1414004.1414015>

[34] Jia, J., & Qiu, W. (2020). Research on an ensemble classification algorithm based on Differential Privacy. *IEEE Access*, 8, 93499–93513. <https://doi.org/10.1109/ACCESS.2020.2995058>

[35] Орєхов, О. С. (2024). Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру JAVA-застосунків на ранніх стадіях розробки. У В. В. Вичужанін (Ред.), *Розвитки інформаційно-керуючих систем та технологій* (с. 360–379). Liha-Pres. <https://doi.org/10.36059/978-966-397-422-4>

[36] Oriekhov, O., Farionova, T., Chernova, L., & Vorona, M. (2025). A Five-Factor Nonlinear Regression Model for JAVA Applications Size Estimation. In *Proceedings of VI International Workshop on IT Project Management*, Kyiv, Ukraine, May 22, 2025. *CEUR Workshop Proceedings*, Vol. 4023, pp. 1-10. <https://ceur-ws.org/Vol-4023/paper1.pdf>

[37] Орєхов, О. С., & Фаріонова, Т. А. (2025). П'ятифакторна математична модель для обробки інформації з метрик коду JAVA-застосунків для оцінювання їх розміру. *Вісник Херсонського національного технічного університету*, 4(95, ч.2), 139–146. DOI: <https://doi.org/10.35546/kntu2078-4481.2025.4.2.18>

[38] Орєхов, О. С., & Фаріонова, Т. А. (2025). Математична модель для обробки інформації з метрик коду Data Science та Machine Learning JAVA-застосунків для оцінювання їх розміру. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 500(2), 275–285. [https://doi.org/10.15589/znp2025.2\(500\).37](https://doi.org/10.15589/znp2025.2(500).37)

[39] Орєхов, О. С., & Фаріонова, Т. А. (2025). Комп'ютерна програма «Сервіс оцінювання інформації з метрик програмних застосунків»

(SoftwareMetricsEstimator) [Свідоцтво про реєстрацію авторського права № 135110 від 15.04.2025]. *Бюлетень з авторського права та суміжних прав*, 89, 139.

https://nipo.gov.ua/wp-content/uploads/2025/05/BULETEN_Avt_Pravo_-89-2025.zip

[40] **Oriekhov, O.**, & Farionova, T. (2025). Information technology for estimating the size of software projects. In I. Linde (Ed.), *Innovative technologies for project and program management* (pp. 192–201). European University Press. <https://mmp-conf.org/documents/archive/monography2025.pdf>

[41] Cisco. (2017, 23 травня). Cisco survey reveals close to three-fourths of IoT projects are failing. Cisco Newsroom. <https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2017/m05/cisco-survey-reveals-close-to-three-fourths-of-iot-projects-are-failing.html>

[42] Гладка, М. В. (2021). *Моделі та методи мультиагентного розподілу трудових ресурсів в ІТ проектах в умовах невизначеності* [Дисертація кандидата технічних наук, Київський національний університет імені Тараса Шевченка].

[43] Liubchenko, V. (2024). Machine learning techniques for predicting software code properties using design metrics. *CEUR Workshop Proceedings*, 3675, 29–38. <https://ceur-ws.org/Vol-3675/paper3.pdf>

[44] Valerdi, R., Boehm, B. W., & Reifer, D. J. (2003). COSYSMO: A constructive systems engineering cost model coming of age. *Systems Engineering*, 13(1), 70–82.

[45] Закон України. (1994). *Про захист інформації в інформаційно-телекомунікаційних системах* (№ 80/94-ВР). Верховна Рада України. <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80>

[46] Butterfield, A., & Ngondi, G. E. (Eds.). (2016). *A dictionary of computer science* (7th ed.). Oxford University Press. <https://doi.org/10.1093/acref/9780199688975.001.0001>

[47] Приходько, Н. В., & Приходько, С. Б. (2018). Нелінійна регресійна модель для оцінювання розміру програмного забезпечення інформаційних систем на

базі VB. *Інформаційні технології та комп'ютерна інженерія*, (3), 37–42.
http://nbuv.gov.ua/UJRN/Itki_2018_3_8

[48] Макарова, Л. М., Латанська, Л. О., Нікітін, О. В., & Нікітіна, О. Ю. (2022). Математичні моделі для оцінювання розміру програмного забезпечення для крос-платформної розробки мобільних застосунків із використанням платформи Xamarin. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*, 33(1), 150–156.
<https://doi.org/10.32838/2663-5941/2022.1/23>

[49] Латанська, Л. О., Макарова, Л. М., & Нікітіна, О. Ю. (2022). Математичне моделювання в задачах оцінювання розміру прикладного програмного забезпечення з відкритим кодом на C#. *Computer Science and Applied Mathematics*, (1), 66–74. <https://doi.org/10.26661/2413-6549-2022-1-08>

[50] 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*, 60(1), 85–92.
<https://doi.org/10.15588/1607-3274-2024-1-8>

[51] Prykhodko, S., Shutko, I., & Prykhodko, A. (2021). Early LOC estimation of web apps created using Yii framework by nonlinear regression models. *WSEAS Transactions on Computers*, 20, 321–328. <https://doi.org/10.37394/23205.2021.20.35>

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

[53] Приходько, С. Б., & Ворона, М. В. (2021). Оцінювання розміру РНР-застосунків з відкритим кодом за нелінійними регресійними моделями з різними факторами. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 484(1), 92–98.
[https://doi.org/10.15589/znp2021.1\(484\).13](https://doi.org/10.15589/znp2021.1(484).13)

- [54] Prykhodko, S. B., Shutko, I. S., & Prykhodko, A. S. (2022). A nonlinear regression model to estimate the size of web apps created using the CakePHP framework. *Radio Electronics, Computer Science, Control*, 59(4), 129–139. <https://doi.org/10.15588/1607-3274-2021-4-12>
- [55] Приходько, А. С., & Малахов, Є. В. (2025). Удосконалена математична модель оцінювання якості вебдодатків з відкритим кодом. Центральнотураїнський науковий вісник. Технічні науки, 11(42, ч. 2), 30–37. [https://doi.org/10.32515/2664-262X.2025.11\(42\).2.30-37](https://doi.org/10.32515/2664-262X.2025.11(42).2.30-37)
- [56] Zhou, Y., Yang, Y., Xu, B., Leung, H., & Zhou, X. (2014). Source code size estimation approaches for object-oriented systems from UML class diagrams: A comparative study. *Information and Software Technology*, 56(2), 220–237. <https://doi.org/10.1016/j.infsof.2013.09.003>
- [57] Morin, K., & Davis, J. L. (2017). Cross-validation: What is it and how is it used in regression? *Communications in Statistics - Theory and Methods*, 46(11), 5238–5251. <https://doi.org/10.1080/03610926.2015.1099672>
- [58] Орехов, О. С., & Фаріонова, Т. А. (2024). Порівняльний аналіз нелінійних регресійних моделей оцінювання кількості рядків коду JAVA-застосунків. *Матеріали XV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (26-27 вересня 2024). Миколаїв: НУК імені адмірала Макарова. 563 – 566.
- [59] Aniche, M. (2015). Java code metrics calculator (CK) [Software]. GitHub. <https://github.com/mauricioaniche/ck>
- [60] Olkin, I., & Sampson, A. R. (2001). Multivariate analysis: Overview. In N. J. Smelser & P. B. Baltes (Eds.), *International encyclopedia of the social & behavioral sciences* (1st ed., pp. 10240–10247). Elsevier.
- [61] Prykhodko, S., & Prykhodko, N. (2021). Mathematical modeling of non-Gaussian dependent random variables by nonlinear regression models based on the multivariate normalizing transformations. In S. Shkarlet, V. Morozov, & A. Palagin (Eds.), *Mathematical modeling and simulation of systems (MODS'2020)* (pp. 166–174). Springer, Cham. https://doi.org/10.1007/978-3-030-58124-4_16

- [62] Prykhodko, S., Prykhodko, N., Makarova, L., & Pukhalevych, A. (2020). Outlier detection in non-linear regression analysis based on the normalizing transformations. In *Proceedings of 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (с. 407–410). IEEE. <https://doi.org/10.1109/TCSET49122.2020.235464>
- [63] Орехов, О. С., & Фаріонова, Т. А. (2025). Механізми зменшення мультиколінеарності між незалежними факторами при побудові нелінійної регресійної моделі для оцінювання JAVA-застосунків. У *Матеріали VI всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2025)* (с. 40 - 42). Національний університет кораблебудування імені адмірала Макарова. <https://itconf.nuos.edu.ua/2025/publications/approaches-to-reducing-multicollinearity-between-independent-factors-in-nonlinear-regression-modelling-for-java-application-assessment/>
- [64] Javdani Gandomani, T., Koh, T. W., & Binhamid, A. (2014). A Case Study Research on Software Cost Estimation Using Experts'. Estimates, Wideband Delphi, and Planning Poker Technique. *International Journal of Software Engineering and its Applications*, 8 (11), 173-182. <http://dx.doi.org/10.14257/ijseia.2014.8.11.16>
- [65] Ziauddin, Z., Kamal, S., Khan, S., & Nasir, J.A. (2013). A fuzzy logic based software cost estimation model. *International Journal of Software Engineering and its Applications*. 7 (2), 7–18.
- [66] Nassif, A. B., Azzeh, M., Capretz, L. F., & Ho, D. (2016). Neural network models for software development effort estimation: A comparative study. *Neural Computing and Applications*, 27 (8), 2369–2381. <https://doi.org/10.1007/s00521-015-2127-1>
- [67] Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (3rd ed.). John Wiley & Sons. <https://doi.org/10.1002/9781118625590>
- [68] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

- [69] Subramanyam, R., & Krishnan, M. (2003). Empirical analysis of CK metrics for object-oriented design complexity: Implications for software defects. *IEEE Transactions on Software Engineering*, 29(4), 297–310. <https://doi.org/10.1109/TSE.2003.1191795>
- [70] Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis* (6th ed.). Pearson Prentice Hall.
- [71] Slifker, J. F., & Shapiro, S. S. (1980). The Johnson system: Selection and parameter estimation. *Technometrics*, 22(2), 239–246. <https://doi.org/10.2307/1268463>
- [72] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3), 519–530. <https://doi.org/10.1093/biomet/57.3.519>
- [73] Bagenal, M. (1955). A note on the relations of certain parameters following a logarithmic transformation. *Journal of the Marine Biological Association of the United Kingdom*, 34(2), 289–296.
- [74] Benoit, K. (2011). *Linear regression models with logarithmic transformations*. London School of Economics, London, 22(1), 23–36.
- [75] Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2), 211–252. <https://doi.org/10.1111/j.2517-6161.1964.tb00553.x>
- [76] Sakia, R. M. (1992). The Box-Cox transformation technique: A review. *The Statistician*, 41(2), 169–178. <https://doi.org/10.2307/2348250>
- [77] Osborne, J. (2010). Improving your data transformations: Applying the Box-Cox transformation. *Practical Assessment, Research, and Evaluation*, 15(1), 12. <https://doi.org/10.7275/qbpc-gk17>.
- [78] Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4), 954–959. <https://doi.org/10.1093/biomet/87.4.954>
- [79] Приходько, С. Б. (2005). *Методичні вказівки до виконання лабораторних робіт з дисципліни “Обробка експериментальних даних на ЕОМ”*. Національний університет кораблебудування імені адмірала Макарова.

- [80] Приходько, С. Б., Макарова, Л. М., & Пугаченко, К. С. (2018). *Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Обробка експериментальних даних на комп'ютері"*. Національний університет кораблебудування імені адмірала Макарова.
- [81] Zabolotnii, S. V., Cherpynoha, A. V., Chorniy, A. M., & Honcharov, A. V. (2020). Comparative analysis of polynomial maximization and maximum likelihood estimates for data with exponential power distribution. *Visnyk NTUU KPI Serii A - Radiotekhnika Radioaparobuduvannia*, 82, 44–51. <https://doi.org/10.20535/RADAP.2020.82.44-51>
- [82] Lindsey, C., & Sheather, S. (2010). Power transformation via multivariate Box–Cox. *The Stata Journal*, 10(1), 69–81. <https://doi.org/10.1177/1536867X1001000108>
- [83] Rousseeuw, P. J., & Van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411), 633–639. <https://doi.org/10.1080/01621459.1990.10474920>
- [84] Bock, D. E., Velleman, P. F., & De Veaux, R. D. (2007). *Stats: Modeling the world* (2nd ed.). Pearson Addison Wesley.
- [85] Bates, D. M., & Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. John Wiley & Sons. <https://doi.org/10.1002/9780470316757>
- [86] Triantaphyllou, E. (2000). *Multi-criteria decision making methods: A comparative study*. Kluwer Academic Publishers. <https://doi.org/10.1007/978-1-4757-3157-6>
- [87] Tzeng, G.-H., & Huang, J.-J. (2011). *Multiple attribute decision making: Methods and applications*. CRC Press. <https://doi.org/10.1201/b11032>
- [88] Leavitt, H. J., & Whisler, T. L. (1958). Management in the 1980's. *Harvard Business Review*, 36(6), 41–48. <https://hbr.org/1958/11/management-in-the-1980s>
- [89] Rennie, R., & Law, J. (Eds.). (2019). *A dictionary of physics (8th ed.)*. Oxford University Press. <https://archive.org/details/a-dictionary-of-physics/page/n523/mode/2up>

- [90] Держстандарт України. (1994). Автоматизовані системи. Терміни та визначення (ДСТУ 2226-93). С. 92.
- [91] Держстандарт України. (1994). Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення (ДСТУ 2481-94). С. 72.
- [92] **Орєхов, О. С., & Фаріонова, Т. А.** (2025). Сучасні інформаційні технології для обробки метрик коду на ранніх етапах проєктування програмного забезпечення. *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (25-26 вересня 2025). Миколаїв: НУК імені адмірала Макарова. 588 - 590.
- [93] **Орєхов, О. С., & Ворона, М. В.** (2025). Проєкт інформаційної технології для обробки інформації з метрик коду JAVA-застосунків. У *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (с. 604–607). Національний університет кораблебудування імені адмірала Макарова.
- [94] Stonebraker, M., & Rowe, L. A. (1986). The design of POSTGRES. *SIGMOD '86: Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, 340–355. <https://doi.org/10.1145/16856.16888>
- [95] PostgreSQL Global Development Group. (2025). *PostgreSQL (Version 16)* [Software]. <https://www.postgresql.org/>
- [96] Johnson, R., Hoeller, J., Arendsen, A., Risberg, T., & Sampaleanu, C. (2007). *Professional Java development with the Spring Framework*. Wiley.
- [97] Pivotal Software. (2023). *Spring Boot (Version 2.7.18)* [Software]. <https://spring.io/projects/spring-boot>
- [98] Golding, T. (2024). *Building multi-tenant SaaS architectures*. O'Reilly Media.
- [99] Turner, M., Budgen, D., & Brereton, P. (2003). Turning software into a service. *IEEE Computer*, 36(10), 38–44. <https://doi.org/10.1109/MC.2003.1236470>
- [100] Pahl C., Jamshidi P. Microservices: A Systematic Mapping Study. *Proceedings of the 6th International Conference on Cloud Computing and Services Science*

(CLOSER 2016). Rome, Italy, 2016. P. 137–146. DOI: <https://doi.org/10.5220/0005748601370146>.

[101] Freier, A., Karlton, P., & Kocher, P. (2011). *The secure sockets layer (SSL) protocol version 3.0* (RFC 6101). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc6101>

[102] Rescorla, E. (2018). *The transport layer security (TLS) protocol version 1.3* (RFC 8446). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc8446>

[103] Oracle. (n.d.). Oracle JDK 8 and JRE 8 Certified System Configurations. Retrieved November 12, 2025, from <https://www.oracle.com/java/technologies/javase/products-doc-jdk8-jre8-certconfig.html>

[104] Berners-Lee, T., Fielding, R., & Masinter, L. (2005). *Uniform Resource Identifier (URI): Generic syntax* (RFC 3986). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc3986>

[105] Fielding, R., Nottingham, M., & Reschke, J. (2022). *HTTP semantics* (RFC 9110). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc9110>

[106] Fielding, R., & Reschke, J. (2014). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and content* (RFC 7231). Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc7231>

[107] Stenberg, D. (n.d.). Everything curl. Retrieved November 12, 2025, from <https://curl.se/book/>

ДОДАТОК А - АНАЛІЗ УСПІШНОСТІ РОЗРОБКИ ПРОЄКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Успішна реалізація проєкту з розроблення програмного забезпечення (ПЗ) є основною метою будь-якої компанії незалежно від того – сервісна чи продуктова. На результативність будь-якого проєкту впливає чимало факторів: розмір проєкту, фінансове планування та правильне оцінювання бюджету на розроблення проєкту, технічна кваліфікація та м'які навички кожного члена команди, здатність колективу злагоджено працювати, методологія розроблення, ефективне планування кожного етапу роботи тощо. Тому актуальним питанням є аналітичне оцінювання умов успішності ІТ-проєктів.

Попит на більшу функційність, надійність і продуктивність викликав підвищення вимог як до самого продукту, так і загалом до організації процесів ІТ-компаній. Щоб залишатися конкурентоспроможними, розробники ПЗ мають постачати програмні продукти вчасно, у межах бюджету та узгодженого рівня якості. Багато проєктів зазнають невдачі через проблеми з плануванням ресурсів (вартості, термінів розроблення тощо) та формулюванням вимог.

Дослідження результативності розроблення проєктів ПЗ упродовж 1994–2020 рр., проведене *Standish Group* [1], показало, що частка успішно реалізованих проєктів відтворює помірну позитивну динаміку. Згідно із дослідженням [1] такі проєкти не мали жодних проблем за такими показниками, як вчасна реалізація (*OnTime*), у межах бюджету (*OnBudget*), відповідність вимогам (*OnTarget*), досягнення цілей (*OnGoal*), досвід взаємодії замовника з виконавцем (*Customer Satisfaction*). Дослідження [1] показало, що частка успішно реалізованих ІТ-проєктів збільшилась з 16% до 35%. Обсяг провальних проєктів зменшився з 31% до 19%. Це пов'язано з поширенням використання *Agile*-методологій розроблення програмних проєктів (особливо малих і середніх) та доводить їх ефективність перед класичними методологіями.

Але необхідно зазначити, що відсоток проєктів, які вийшли на ринок, але мали певні проблеми за будь-яким із показників (не вклалися в терміни,

перевищили бюджет або не мали 100% функціоналу відповідно до вимог тощо) має незначну тенденцію до зменшення, приблизно 4% [1]. На рис. А.1 наведена динаміка успішності реалізації проєктів розроблення ПЗ упродовж 1994–2020 рр., де *Successful* – результативні проєкти, *Challenged* – проєкти з певними проблемами під час реалізації, *Failed* – проєкти, що не здійснилися.

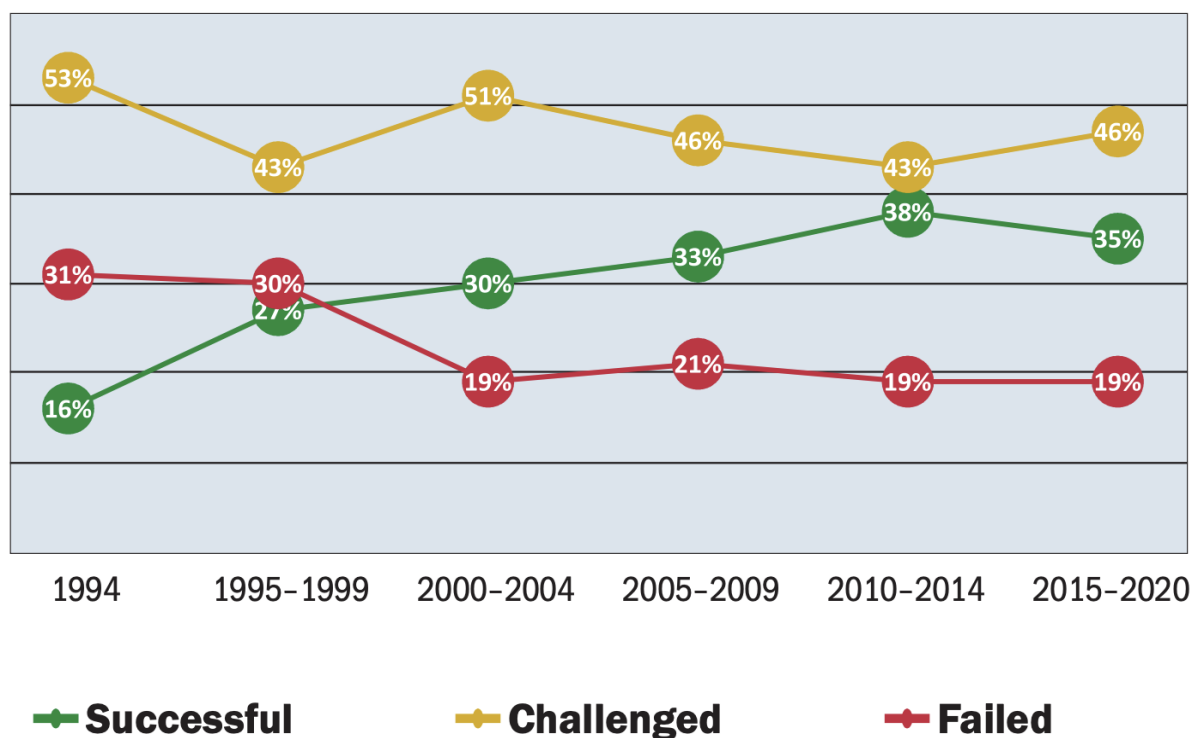


Рисунок А.1 – Динаміка успішності реалізації програмних проєктів протягом 1994– 2020 рр. (за інформацією The Standish Group) [3]

Компанія *CISCO* 2017 р. провела дослідження успішності реалізації проєктів IoT (*Internet-of-Things*), яке показало, що близько 60% з них зазнають невдач на етапі розроблення концепту й лише 26% випущених IoT-проєктів завершуються вчасно, вкладаються в бюджет та мають заявлену функційність [2]. Інше дослідження *The Standish Group* доводить: що більший і складніший проєкт, то вища вірогідність його невдачі або повного невиконання. На рис. А.2 подано статистику успішності реалізації IoT-проєктів, яка демонструє, що 61% проєктів малого розміру (*Small*) результативні та 7% зазнали невдач, а великі (*Large*) й дуже великі (*Grand*) проєкти мають набагато меншу статистику

успішної реалізації – 11% і 6% відповідно, і частка нездійснених проєктів становить 30% і 43% відповідно [3].

	SUCCESSFUL	CHALLENGED	FAILED	TOTAL
Grand	6%	51%	43%	100%
Large	11%	59%	30%	100%
Medium	12%	62%	26%	100%
Moderate	24%	64%	12%	100%
Small	61%	32%	7%	100%

Рисунок А.2 – Статистика успіху ІТ-проєктів залежно від його розміру [3]

Дослідження реалізації ІТ-проєктів підтверджують: що більший проєкт, то вища вірогідність невдачі. Подана в роботі [4] статистика успішності проєктів створення ПЗ залежно від кількості рядків коду або функціональних точок показала, що 92% проєктів з розміром 1000 рядків були успішно реалізовані раніше або вчасно, тоді як для проєктів з приблизним розміром 100 000 рядків коду результативність становить тільки 62%. А для великих проєктів 10 млн рядків коду, 14% були реалізовані вчасно, 21% реалізовані із запізненням і 65% не виконані (скасовані).

За інформацією *The Standish Group* [1] можна зробити висновок, що для успішного ІТ-проєкту вирішальну роль відіграють високий рівень професійних навичок членів команди та кваліфіковані, досвідчені спонсори або замовники проєкту. *Agile*-методологія добре співіснує із зазначеними нетехнічними

факторами. Це дає змогу краще підлаштовувати етапи розроблення ПЗ під вимоги, особливо це актуально для великих проєктів, які працюють тривалий час. *Agile*-методологія ґрунтується на парадигмах гнучкості, неточності, обмеженості даних і змінах напряму розроблення, що робить її якісно кращою за традиційні методології з фіксованими вимогами. Ефективність *Agile* підтверджує дослідження [1], де зазначено, що успішність реалізації *Agile*-проєктів становить 48% проти 20% для каскадної методології розроблення. Ще одним фактором успішності є використання парадигми безкінечної модернізації програмного проєкту [1]. За такого підходу команда підлаштовується або збирається відповідно до бюджету й дохідності проєкту, робота виконується в режимі безкінечного покращення та нарощування функціоналу з умовними релізами версій. Упровадження зазначеного підходу ефективно для успішно запущених проєктів, які прибуткові й більш орієнтовані на продуктиві компанії. Успішність таких проєктів становить близько 80%.

У роботі проаналізований сучасний стан успішності розроблення проєктів ПЗ. Динаміка показує поступове збільшення статистики ефективної реалізації, що пов'язано з використанням нових технологій і підходів. Визначено, що тенденція вірогідності успіху залежить від розміру проєктів і методології розроблення. Доведено, що *Agile*-проєкти мають більший шанс на успіх, ніж проєкти, що розробляються із застосуванням традиційних методологій розроблення ПЗ. Крім того, зазначено, що підхід безкінечної модернізації, на основі *Agile*-методології забезпечує значно більшу ймовірність успіху, але такий підхід орієнтований на продуктиві компанії та проєкти, що вже здатні приносити прибуток.

Література

1. Johnson, J., Mulder, H. The Standish Group International Inc, Endless Modernization: How infinite flow keeps software fresh, [Online]. 2021.

https://www.researchgate.net/publication/348849361_Endless_Modernization_How_Infinite_Flow_Keeps_Software_Fresh

2. Cisco Systems Inc, Cisco Survey Reveals Close to Three-Fourths of IoT Projects Are Failing, [Online]. 2017.
https://investor.cisco.com/files/doc_news/Cisco-Survey-Reveals-Close-to-Three-Fourths-of-IoT-Projects-Are-Failing-2017.pdf

3. The Standish Group International Inc, CHAOS Report. 2015. [Online].
<https://cdn1-public.infotech.com/agile/CHAOSReport2015-Final.pdf>

4. S. McConnell. Software Estimation: Demystifying the Black Art, Redmond, Washington: Microsoft Press. 2006. 352 p.

ДОДАТОК Б - НАВЧАЛЬНИЙ ТА ТЕСТОВИЙ НАБОРИ ДАНИХ ЗАГАЛЬНИХ JAVA-ЗАСТОСУНКІВ З ВІДКРИТИМ ПРОГРАМНИМ КОДОМ

Таблиця Б.1 – Навчальний набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO	RFC
1	2	3	4	5	6	7	8	9
1	mosaic.git	60,698	1334	170	6623	3657	8842	14844
2	mp4parser.git	37,091	657	23	3852	2055	3300	6813
3	closure-stylesheets.git	27,376	504	25	3214	1081	4197	5344
4	animation-samples.git	4,47	132	10	482	290	779	1227
5	shifu.git	61,27	765	20	4306	2936	4674	13793
6	s2-geometry-library-java.git	53,506	635	61	4253	1414	4841	9915
7	directory-server.git	119,761	1315	75	9372	3809	12676	22688
8	lemminx.git	60,333	840	72	5946	1690	6329	11398
9	dyno.git	25,108	735	46	3173	783	2685	4404
10	spring-data-relational.git	56,74	1451	260	6735	2648	9609	11678
11	jpa-spec.git	1,586	47	66	661	15	110	45
12	Java-Machine-Learning.git	3,539	66	15	407	153	307	643
13	open-location-code.git	2,25	52	15	224	145	330	533
14	cleartk.git	35,976	908	65	2810	1533	5530	8506
15	subversion-plugin.git	12,322	182	6	866	376	1468	2712
16	staash.git	12,686	314	56	1385	656	1540	2339
17	LiTr.git	7,628	176	12	737	640	1264	1471
18	logging-log4j2.git	145,041	2739	282	15251	6310	16308	29782
19	tapestry-5.git	130,368	3600	635	15263	7072	17575	23825
20	cv4j.git	7,301	136	4	441	268	606	751
21	spring-cloud-task.git	14,604	400	13	1607	780	2792	3502
22	celeborn.git	50,439	733	46	4559	2944	5435	12071
23	entity-fishing.git	29,591	255	1	2080	879	1783	4843
24	fineract.git	314,102	6105	631	22167	25719	43102	63807
25	commons-math.git	125,143	1712	134	10537	3349	8558	14624
26	jkube.git	58,842	1455	66	6209	2897	9562	24431
27	incubator-hugegraph.git	111,809	1217	72	9025	3291	10150	21555
28	copybara.git	61,517	1097	73	6059	3137	8480	15938
29	flume.git	73,591	1109	87	5108	3334	5995	13796
30	SimianArmy.git	16,056	267	38	1545	684	1783	4413
31	git-client-plugin.git	20,821	167	15	1805	690	1704	4297
32	spring-batch.git	91,63	2338	135	10841	4497	14375	19655
33	vertx-sql-client.git	61,773	880	58	5897	4078	6628	12289
34	toolbox.git	43,157	684	41	4030	1436	5787	12236

1	2	3	4	5	6	7	8	9
35	styx.git	30,819	448	91	3343	2010	4686	11633
36	hawkbit.git	54,747	1002	177	6901	2821	9754	15498
37	stratos.git	104,263	1785	57	9461	5850	10076	22356
38	james-project.git	354,945	7263	1139	43397	21680	58063	106636
39	firebase-android-sdk.git	196,389	3437	295	19093	8762	23108	43261
40	live-transcribe-speech-engine.git	2,668	35	10	190	177	254	599
41	baritone.git	21,897	385	83	2129	1014	3073	5384
42	parquet-mr.git	110,781	1929	95	9680	4183	12664	19985
43	servicecomb-toolkit.git	12,745	630	105	1724	501	4216	3709
44	incubator-nemo.git	32,999	812	62	3060	1699	6060	9465
45	gora.git	47,188	585	15	4831	1993	4524	9687
46	databus.git	124,341	1387	129	9890	5934	8335	19728
47	github-branch-source-plugin.git	16,098	208	1	1189	402	1825	2806
48	point-of-sale-android-sdk.git	2,278	61	4	194	172	300	449
49	spectator.git	41,279	640	49	4822	1477	3751	8283
50	docker-java.git	29,139	754	187	5025	1857	5616	6710
51	governator.git	19,818	822	61	2119	862	3271	4049
52	paho.mqtt.android.git	6,206	121	9	548	358	635	1212
53	dependency-check-plugin.git	2,329	52	0	322	177	289	469
54	dagli.git	137,599	2367	605	16302	4735	37418	22950
55	depan.git	38,585	1256	104	4642	2038	6634	7928
56	ambry.git	208,967	1926	173	12247	11832	18417	41539
57	flink-cdc.git	107,807	1674	105	9471	5930	14245	30573
58	commons-vfs.git	32,593	605	43	3747	1246	3976	7673
59	maxwell.git	17,546	297	11	1559	948	1893	4077
60	datafu.git	20,506	338	13	1474	870	2049	4063
61	robolectric.git	204,818	3397	357	25524	8189	20999	37027
62	incubator-hivemall.git	76,739	1022	34	5340	2254	7004	10592
63	MPAndroidChart.git	21,111	276	30	2289	810	2706	5726
64	artifact-manager-s3-plugin.git	2,566	41	1	231	127	385	788
65	mssql-jdbc.git	81,532	627	21	5489	3111	3822	11870
66	jenkins-test-harness.git	12,76	325	18	1252	418	1672	2815
67	startup-os.git	13,342	211	45	993	537	1366	3851
68	auto.git	30,957	979	130	3519	799	4583	7538
69	spring-restdocs.git	13,745	396	35	1953	534	2608	4590
70	light-4j.git	66,114	1073	88	5792	3777	5210	15464
71	Conjecture.git	4,595	63	8	742	165	304	839
72	commons-pool.git	16,202	206	16	1517	493	933	2490
73	jetcache.git	18,011	389	57	1750	725	2690	4250
74	epoxy.git	48,651	601	34	7709	1551	4598	5792
75	nailgun.git	2,953	58	6	291	191	199	598
76	docker-client.git	15,458	392	10	2637	330	2700	3304

1	2	3	4	5	6	7	8	9
77	metacat.git	27,579	590	70	2638	1782	5212	6736
78	incubator-retired-edgent.git	20,371	423	66	2179	863	2554	4594
79	gps-measurement-tools.git	5,661	119	4	349	468	522	978
80	spring-session.git	24,255	738	15	2775	934	4230	6043
81	bundletool.git	52,103	798	51	6479	1781	7981	22124
82	data-transfer-project.git	28,439	724	66	2833	2269	5483	9317
83	realm-java.git	110,167	1820	92	9807	3683	8274	15506
84	security-certification-resources.git	16,583	313	2	640	1996	1413	1874
85	ActionBarSherlock.git	26,463	382	45	2685	1113	2616	4855
86	spring-webflow.git	39,638	882	87	5347	1444	5270	7739
87	truth.git	28,655	393	25	3634	458	3018	5167
88	account-provisioning-for-google-apps.git	2,326	34	2	205	196	170	418
89	stetho.git	19,224	513	74	1530	1035	2572	3092
90	android-samples.git	8,386	176	2	694	679	1436	2272
91	spring-data-gemfire.git	73,165	1636	113	8754	2689	10623	20069
92	JZVideo.git	10,993	170	9	1017	503	1110	2410
93	commons-collections.git	65,525	903	48	7756	1261	4761	10901
94	flo.git	6,792	165	28	710	283	1076	2046
95	spring-cloud-dataflow.git	78,968	1770	119	8609	4747	13136	20992
96	performance-plugin.git	11,461	175	1	1033	504	1080	2657
97	activemq-artemis.git	553,976	6350	485	51148	17898	44580	112225
98	rocketmq.git	180,416	2312	156	16842	8180	16249	44126
99	tsml-java.git	290,27	1894	157	22467	9431	14621	45729
100	knox.git	122,87	1987	211	11023	5041	11626	26940
101	hop.git	5,877	97	2	1205	476	353	668
102	credentials-plugin.git	13,011	266	24	1155	292	2174	2879
103	team-explorer-everywhere.git	432,684	10610	537	48541	19841	47163	87466
104	drill.git	628,639	7684	652	54521	22853	65416	102112
105	eureka.git	32,877	671	71	3185	1756	4391	7637
106	jd-core.git	44,137	543	55	4735	1433	4723	5268
107	servicecomb-pack.git	22,179	584	58	2565	1564	3642	5962
108	tsunami-security-scanner-plugins.git	18,541	581	8	2132	1491	5066	12088
109	docker-plugin.git	11,295	191	3	1020	442	1341	2403
110	spring-data-solr.git	21,501	398	76	3049	695	2979	4511
111	cadence-java-client.git	46,356	650	170	4814	1693	5952	10244
112	office-365-connector-plugin.git	3,152	62	0	399	146	454	868
113	spring-cloud-commons.git	25,298	826	67	2727	1110	4979	5672
114	opennlp.git	64,504	1049	182	4765	2243	6746	10769
115	struts.git	195,866	3029	304	19635	7497	16231	33032
116	astyanax.git	50,928	944	131	6317	2239	6470	10444
117	avro.git	102,344	1541	73	8597	3362	10307	18251

1	2	3	4	5	6	7	8	9
118	titus-control-plane.git	116,955	2543	299	12735	7830	19666	40835
119	ozone.git	332,233	4207	295	28229	16921	40710	98110
120	spring-data-commons.git	59,986	1646	495	7677	2528	9803	14043
121	spring-cloud-consul.git	6,576	241	10	784	496	1860	2019
122	gitlab-plugin.git	12,52	298	16	1727	688	2178	3353
123	JCTools.git	37,082	751	37	3258	2873	3626	4986
124	iceberg.git	258,993	3667	316	23662	10382	34060	68958
125	iceberg.git	96,998	1521	105	8494	3474	13445	23812
126	avro-util.git	117,435	1169	56	6727	3420	7745	16088
127	data-science-in-action.git	9,741	245	1	656	429	1231	1905
128	json-in-db.git	17,656	296	5	1219	1336	1143	3289
129	archiva.git	87,254	974	167	9687	3572	8581	20840
130	spring-integration-samples.git	7,441	318	40	901	407	1730	2184
131	retrofit.git	26,665	985	109	2559	730	4515	4274
132	corrosion.git	7,195	106	4	434	487	887	2290
133	incubator-myriad.git	8,056	194	6	1018	575	1414	2237
134	qcloud-im-server-sdk-java.git	27,357	704	0	4915	2928	2435	1004
135	shenyu.git	125,017	3120	254	16654	7636	19977	41001
136	apollo.git	12,209	364	81	1671	716	2633	4002
137	configuration-as-code-plugin.git	8,501	260	10	904	339	1925	3199
138	spring-kafka.git	60,715	940	92	4693	2600	7711	12497
139	coral.git	18,136	300	6	1875	545	2512	4390
140	jna-gmp.git	1,848	33	1	126	56	110	284
141	windowbuilder.git	357,247	7191	330	30513	10603	41419	71942
142	gitiles.git	12,633	208	7	946	543	1540	3239
143	helix.git	156,854	2282	144	11188	5713	15681	39140
144	mockito.git	51,447	1813	261	6247	1707	6928	10676
145	gerrit-trigger-plugin.git	25,481	312	7	2213	861	2800	5368
146	tez.git	170,234	2175	122	11906	7703	18534	34739
147	spring-cloud-skipper.git	14,401	382	23	1573	889	2703	4267
148	mr4c.git	22,585	484	56	2989	1492	3117	5169
149	pdfbox.git	114,968	1423	52	10132	5187	9783	24152
150	felix-dev.git	368,15	5452	750	27770	12863	28400	62900
151	hadoop-hdfs.git	110,409	968	54	6705	3541	7556	18953
152	servicecomb-saga-actuator.git	6,295	242	40	720	545	1525	1801
153	ignite-3.git	356,209	6129	958	34083	15292	47226	86273
154	seldon-server.git	29,704	601	60	3229	2016	3206	5889
155	netty-incubator-codec-quic.git	15,213	265	21	1507	584	1361	2971
156	shiro.git	36,524	900	119	4681	1405	5123	7945
157	iep.git	9,68	209	17	1087	412	1020	2384
158	spring-cloud-deployer.git	33,173	453	15	3136	1183	3089	7625
159	Cognitive-Face-Android.git	7,227	160	1	588	299	741	1153

1	2	3	4	5	6	7	8	9
160	spring-security.git	205,514	4959	336	22704	9222	40094	56154
161	myanmar-tools.git	2,013	34	0	116	83	119	325
162	airpal.git	3,952	150	15	432	363	1076	1095
163	patchfield.git	1,769	36	0	163	75	145	340
164	Availability-Monitor-for-Kafka.git	2,614	55	6	177	243	368	713
165	shardingsphere-elasticjob-cloud.git	10,72	342	20	1218	659	2246	3837
166	texlipse.git	29,787	491	14	2595	1181	3348	6080
167	kapua.git	51,203	1035	274	6219	2417	6898	9768
168	microprofile-open-api.git	6,89	114	39	1267	173	909	1401
169	brooklin.git	33,909	390	46	2864	1693	2613	8167
170	hover.git	6,176	175	15	609	319	912	1194
171	rest.li.git	266,734	4722	425	24008	10196	32627	53994
172	activemq.git	420,031	7216	303	40682	14711	41580	98501
173	jsonp.git	9,17	133	24	1245	186	406	1529
174	spring-data-rest.git	19,589	597	124	2246	1209	4647	5322
175	elasticsearch-learning-to-rank.git	15,737	280	18	1520	628	2286	3624
176	tink.git	82,161	1409	56	7479	3378	10902	20951
177	Foundry.git	160,264	2162	286	17224	2993	14937	27288
178	FBReaderJ.git	75,101	1489	104	6735	3482	7158	12971
179	malmo.git	19,13	264	14	1214	771	1960	3875
180	tink.git	83,741	1487	58	7726	3325	10991	20786
181	junit-plugin.git	10,067	152	4	957	310	1209	2193
182	accumulo.git	699,674	5313	262	32597	12465	34891	76687
183	warnings-ng-plugin.git	24,874	656	12	2885	1301	4139	6449
184	FML.git	22,645	417	35	2080	998	2319	4899
185	pipeline-maven-plugin.git	13,74	249	10	1138	368	1694	3426
186	tv-samples.git	11,873	354	17	1072	784	2190	3018
187	mug.git	30,021	458	45	4491	322	2280	6510
188	sofa-common-tools.git	14,469	187	14	1488	433	808	2302
189	maven-surefire.git	63,753	1729	81	7118	2125	6843	13715
190	spring-boot-starter.git	4,426	159	16	589	203	790	904
191	vertx-mongo-client.git	11,788	111	2	1172	243	828	2141
192	alluxio.git	222,656	3343	240	20901	12404	24063	54495
193	metron.git	115,252	2579	156	11201	4551	12111	24730
194	shardingsphere-elasticjob.git	14,684	515	59	1689	1099	3041	5150
195	eclipse-collections.git	203,684	3071	293	33655	3265	23602	38513
196	commons-codec.git	18,135	179	8	1729	555	731	2345
197	undertow.git	161,85	2956	175	13353	6079	16167	29885
198	AirMapView.git	2,849	67	14	432	107	402	554
199	spring-loaded.git	43,158	1021	117	4007	1607	2395	6192
200	googletv-android-samples.git	8,615	212	11	685	419	864	1553
201	kubernetes-plugin.git	14,604	273	8	1509	650	2238	4528

1	2	3	4	5	6	7	8	9
202	commons-io.git	44,25	624	38	5267	1266	2543	7803
203	httpcomponents-core.git	69,726	1122	145	7377	2578	7736	12720
204	playwright-java.git	38,422	600	44	4781	1289	3798	7106
205	nebula.git	202,954	3009	163	18096	8323	17259	36790
206	spring-data-redis.git	102,327	1364	258	15599	2546	11406	21934
207	incubator-samoa.git	29,296	451	58	3147	2086	3101	4974
208	google-authenticator.git	3,935	54	7	303	173	237	490
209	spring-cloud-zookeeper.git	4,795	185	10	556	328	1404	1310
210	karaf.git	112,129	1735	191	8819	4720	10363	25678
211	appcenter-sdk-android.git	42,737	889	41	3420	1300	4843	10049
212	openhab-core.git	146,535	2369	368	13735	8851	21471	35744
213	belvedere.git	4,386	94	16	375	188	573	943
214	Java-WebSocket.git	12,801	245	16	1307	327	928	2205
215	graphql-java.git	60,66	1053	162	6517	2569	9771	13379
216	io2014-codelabs.git	9,306	181	9	821	568	984	2170
217	blueocean-plugin.git	40,095	891	41	3980	1599	7015	10124
218	gson.git	35,583	1081	21	2906	1031	3256	4915
219	spring-javaformat.git	7,609	237	13	668	289	873	1885
220	conscrypt.git	56,465	885	33	4999	1743	3172	9107
221	connectivity-samples.git	6,253	148	6	510	334	685	1299
222	mltk.git	19,314	206	10	1288	404	1313	2587
223	commons-dbcp.git	29,575	174	8	3832	601	923	4925
224	ViewPagerIndicator.git	2,947	62	4	261	128	359	661
225	java-photoslibrary.git	56,763	313	58	5239	1369	3572	6438
226	parseq.git	28,724	586	102	3532	1017	4248	6495
227	incubator-tuweni.git	44	491	55	6455	835	3556	9297
228	AutoDispose.git	5,399	116	17	703	137	945	1003
229	Quicksql.git	247,661	3564	391	23629	6638	28369	44270
230	J2EEFAST.git	21,385	389	62	2131	1495	3090	5149
231	okbuck.git	7,06	180	7	810	511	1222	3219
232	guice.git	11,721	430	21	1228	347	1604	2512
233	adt-tools-base.git	601,362	9109	752	41781	26781	59676	70503
234	jnosql.git	46,868	1167	189	7009	1628	7402	12070
235	jmeter.git	127,435	1590	127	11503	7567	9464	31079
236	easy-query.git	147,232	2215	1115	22212	5019	27052	25628
237	ProjectOxford-Apps-MimickerAlarm.git	7,197	152	21	626	396	955	1764
238	velocity-engine.git	25,161	477	47	3003	985	2462	4234
239	djl-serving.git	12,676	173	5	812	537	1200	3373
240	helios.git	37,442	800	50	3363	2263	5051	11228
241	tyrus.git	66,253	1951	51	4855	1794	7226	11029
242	java-design-patterns.git	33,207	1564	164	4146	1791	5700	8474

1	2	3	4	5	6	7	8	9
243	servlet-spec.git	2,101	33	43	621	146	9	318
244	media-samples.git	9,606	194	10	779	555	947	1695
245	spring-cloud-netflix.git	7,522	202	4	1115	431	1568	1790
246	spring-statemachine.git	51,261	1552	112	5550	1620	10277	11123
247	javamail.git	66,455	896	32	4741	1927	2065	10134
248	tajo.git	208,686	2562	115	17812	7156	20679	39191
249	shipkit.git	8,582	320	20	998	491	1428	2557
250	FBReader-Android-2.git	68,632	1370	103	5652	3357	6612	11875
251	bookkeeper.git	252,017	3833	429	19210	10216	24382	51598
252	mybatis.git	8,703	118	1	467	258	1100	2309
253	spring-data-neo4j.git	59,438	1238	379	6295	2453	10320	13349
254	openmeetings.git	60,294	1187	78	6155	4493	9128	16464
255	servo.git	12,823	263	28	1397	698	1674	3097
256	hadoop-mapreduce.git	188,878	2723	97	13193	7970	19863	37156
257	Data Science with Java.git	2,804	87	7	308	129	482	801
258	doris-flink-connector.git	23,704	365	12	2001	1639	2570	6357
259	pygradle.git	3,875	110	22	506	230	679	1343
260	spring-data-examples.git	50,626	4476	2128	11262	6816	17244	5205
261	jimfs.git	16,398	211	7	1535	373	1066	2918
262	metrics.git	31,007	627	43	2965	1481	3102	5852
263	weblogic-monitoring-exporter.git	8,96	214	19	1200	610	936	2167
264	pattern.git	3,146	128	2	421	131	654	908
265	EVCache.git	15,012	140	21	1286	664	1096	3147
266	ibatis-2.git	20,319	306	36	2559	727	1467	3520
267	spring-cloud-gateway.git	37,068	1037	68	4358	1459	7562	10992
268	j2cl.git	216,612	4590	898	30928	4343	17253	27711
269	exposure-notifications-android.git	27,074	535	23	2838	1994	5822	8703
270	generator.git	72,79	627	26	4458	976	4356	9959
271	vorto.git	36,576	868	108	4730	1975	6243	10233
272	input-samples.git	6,969	184	14	596	420	1155	1794
273	artifactory-plugin.git	29,891	648	8	3439	1908	4577	6167
274	commons-compress.git	66,695	739	38	5444	2934	3877	10450
275	stapler.git	19,553	529	53	2061	723	2570	3932
276	opentelemetry-plugin.git	11,135	227	15	1206	595	1922	3262
277	conductor.git	47,052	632	32	5487	1988	4956	13347
278	tienchin-video.git	561,263	8980	1212	71302	24007	67760	127607
279	junit4.git	32,915	1428	42	3922	941	5501	4821
280	incubator-seatunnel.git	105,104	2455	210	9591	8376	16794	31496
281	spring-data-couchbase.git	32,67	608	306	3631	1428	6197	7690
282	initializr.git	29,705	658	46	3610	1107	3907	7887
283	missinglink.git	2,331	89	15	263	56	389	918
284	clojure.git	46,79	391	526	4345	1850	3301	5307

1	2	3	4	5	6	7	8	9
285	ec2-plugin.git	11,536	175	5	957	487	1365	2731
286	leshan.git	56,448	960	141	5793	2220	8292	11924

Таблиця Б.2 – Тестовий набір даних

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO	RFC
1	2	3	4	5	6	7	8	9
1	subzero.git	3,531	126	3	269	170	511	1060
2	remoting.git	33,008	540	49	2624	1377	2715	5713
3	guice.git	79,577	3424	325	7403	2154	14121	15212
4	aurora.git	49,435	1187	134	4459	2919	9478	16361
5	otter.git	68,163	1155	141	6266	2816	5881	12394
6	jackrabbit-oak.git	542,811	8229	618	46921	20819	58452	117847
7	milo.git	127,563	1606	447	18077	11552	17550	21446
8	curator.git	1,201	44	1	109	24	110	243
9	jd-gui.git	19,309	360	47	1592	693	1847	4275
10	javacpp.git	26,575	205	7	2931	401	1259	3502
11	syncope.git	173,122	4102	533	19130	9764	34153	50667
12	cxfr.git	589,653	9231	830	52987	20156	66403	130272
13	grizzly.git	146,392	2399	192	12175	5066	12623	26604
14	azure-maven-plugins.git	20,258	445	68	2349	1784	3820	8004
15	dubbo.git	214,358	3824	624	23441	9356	23128	50881
16	sentry.git	214,048	2353	77	14102	5208	13528	29350
17	plc4x.git	432,207	7312	383	53091	20128	63677	89947
18	TextAnalyzer.git	6,885	157	15	711	320	497	1363
19	derby.git	480,09	3447	353	35659	18736	21268	70063
20	ratls.git	59,569	898	193	5913	2572	7868	17374
21	gdata-java-client.git	98,698	1805	90	13028	4627	9608	17941
22	uReplicator.git	14,569	173	7	976	953	1285	3901
23	mybatis-dynamic-sql.git	22,242	539	56	3111	1045	3577	4770
24	spring-vault.git	27,032	593	69	3115	1126	4455	7590
25	aCute.git	4,42	80	0	216	286	595	1521
26	repairnator.git	25,226	410	18	2269	1223	3220	8477
27	openjpa.git	388,615	5384	441	47606	17618	34988	68470
28	datasketches-memory.git	10,023	93	7	1159	209	820	1945
29	rabbitmq-tutorials.git	1,343	43	0	97	52	212	403
30	karyon.git	8,002	257	22	881	409	1470	1636
31	morpheus-core.git	50,934	487	76	4520	1092	4025	10508
32	oryx.git	14,469	392	22	1447	599	2834	5141
33	atlas.git	154,568	1757	89	11778	6956	12936	29339
34	scissors.git	1,525	32	3	148	101	211	325
35	eclipse-collections-kata.git	7,91	142	0	1301	227	1074	2102

1	2	3	4	5	6	7	8	9
36	keywhiz.git	22,772	502	6	3108	1530	4268	7757
37	androidtv-sample-inputs.git	9,226	111	10	697	499	869	1813
38	jacoco.git	34,351	833	52	3536	1156	3650	5891
39	code-coverage-api-plugin.git	10,33	190	4	1088	470	1440	3377
40	venice.git	288,418	3650	346	25106	12606	28940	67065
41	android-arscblamer.git	3,591	57	2	430	236	345	1000
42	azure-devops-intellij.git	45,803	961	63	4734	2829	5676	11549
43	git-plugin.git	29,922	499	8	2766	989	4377	6528
44	Mallet.git	70,037	766	64	5571	3717	5671	11175
45	tiles.git	12,14	268	26	1207	343	1606	2800
46	servicecomb-java-chassis.git	100,997	2855	311	12511	5433	15858	23960
47	hollow.git	81,042	1545	119	7147	3602	8494	14278
48	camunda-bpm-platform.git	545,665	9475	1245	72913	25257	67395	120511
49	acceptance-test-harness.git	23,024	663	20	2485	1120	4022	6914
50	build-failure-analyzer-plugin.git	9,856	147	1	843	380	1203	2311
51	google-maps-services-java.git	8,521	213	12	828	621	943	1276
52	iceberg.git	46,397	744	54	3661	1586	6319	9943
53	plugin-installation-manager-tool.git	4,782	62	2	479	196	319	1091
54	spring-ldap.git	37,059	877	74	4280	1500	3638	7443
55	spring-integration-extensions.git	18,026	452	7	1799	992	2912	4479
56	calcite.git	308,851	4272	649	29573	8115	37936	58386
57	shardingsphere-example.git	2,974	82	14	223	104	621	838
58	pig.git	242,66	2821	67	17539	6338	21897	39103
59	commons-beanutils.git	29,718	369	16	2976	599	1473	4242
60	codebuff.git	119,496	2124	101	12826	3867	12775	17922
61	ranger.git	254,155	2279	66	16347	11275	16684	44596
62	JSAT.git	87,712	1104	71	10252	2827	8468	16976
63	rabbitmq-java-client.git	31,912	616	77	3470	1159	3267	6777
64	moa.git	107,002	1408	108	8329	6012	8270	17094
65	paho.mqtt.java.git	32,562	343	39	2979	1929	2112	5949
66	HydraLab.git	29,476	488	97	2264	2376	3483	7925
67	dropbox-sdk-java.git	453,75	4980	6	24059	8344	29151	47983
68	spring-data-mongodb.git	133,452	2631	438	15867	4951	17751	25212
69	ethereumj.git	74,224	1165	65	7538	2976	7148	14957
70	physical-web.git	7,481	146	10	660	508	747	1673
71	vsminecraft.git	236,04	1459	280	14780	5240	13580	25899
72	jenkins.git	181,241	3869	174	17195	5253	24665	40098
73	cruise-control.git	54,324	748	39	4493	4480	5935	14950
74	datasketches-java.git	80,728	558	26	6282	1540	4518	11730
75	hama.git	47,105	667	82	4196	1891	4538	9356
76	android-testdpc.git	24,022	351	40	1512	1689	2718	5352
77	sens-parent.git	3,273	102	25	421	335	792	821

1	2	3	4	5	6	7	8	9
78	logging-log4jl.git	31,679	511	27	3384	1508	2233	5746
79	eclipse.jdt.ls.git	98,162	990	45	5133	2664	8893	19964
80	mlkit.git	9,181	168	6	581	621	1316	2419
81	code.quarkus.io.git	2,184	46	8	239	143	379	982
82	caliper.git	15,347	467	44	1697	705	2460	3394
83	struts-examples.git	3,85	140	10	725	296	418	708
84	guacamole-client.git	26,739	1001	115	4009	1777	5000	5543
85	identity-credential.git	27,756	260	18	1818	1050	2448	5222
86	unomi.git	41,877	789	76	4333	2312	5653	10268
87	angel.git	204,252	1736	145	14443	5019	14196	23616
88	hashicorp-vault-plugin.git	5,233	123	9	543	297	1023	1525
89	apex-core.git	78,373	1253	116	4906	3240	7049	12887
90	ant.git	137,266	1793	109	13477	6888	8845	25973
91	android-test.git	77,155	2022	123	7515	3548	11372	15596
92	soda-for-java.git	25,724	87	13	1115	916	658	2548
93	xtext-xtend.git	448,896	1894	103	20508	5475	15484	28601
94	spring-cloud-cli.git	2,331	69	0	226	94	315	511
95	wycheproof.git	9,076	69	0	705	111	812	1594
96	Metamorphosis.git	45,965	804	69	4066	1830	4102	8818
97	kafka-streams-machine-learning-examples.git	196,02	112	0	178	286	370	513
98	zeno.git	15,151	343	11	1694	492	1672	2901
99	dr-elephant.git	19,825	342	9	1384	1539	1946	4088
100	systemds.git	339,938	3664	72	31993	13658	30498	67276
101	spring-cloud-function.git	26,314	682	19	2722	766	3866	5981
102	mantis.git	95,971	2271	168	8836	5605	13224	22149
103	jira-steps-plugin.git	6,677	197	1	608	313	1324	1697
104	jifa.git	19,997	390	68	1484	1300	2471	4481
105	mahout.git	95,439	1727	90	8306	3597	11144	20416
106	commons-text.git	24,823	264	14	2380	477	869	3320
107	spring-cloud-open-service-broker.git	15,948	449	24	2173	728	3141	4320
108	jmx_exporter.git	9,568	167	18	902	352	1367	2596
109	spring-integration.git	232,063	4493	318	23112	9296	33149	52380
110	lettuce.git	90,715	1503	326	15267	3131	11600	20073
111	commons-lang.git	86,713	979	89	8706	1697	3232	10462
112	snippets-android.git	6,626	319	2	592	120	1043	1538
113	denominator.git	18,454	561	27	1926	898	3264	5098
114	android-play-games-in-motion.git	3,03	43	0	302	234	277	648
115	commons-csv.git	7,641	59	1	720	165	273	1240
116	Raigad.git	9,235	177	13	1156	874	998	1834
117	blockly-android.git	26,135	465	22	2211	1464	2870	5350
118	trident-ml.git	4,549	92	8	495	192	524	756

1	2	3	4	5	6	7	8	9
119	u2020.git	4,806	123	11	486	342	834	1073
120	active-directory-plugin.git	3,685	52	2	246	161	428	879
121	incubator-celeborn.git	28,49	396	24	2379	1706	3159	7048
122	NullAway.git	18,869	463	39	2021	744	2161	3401
123	DL-Learner.git	88,46	914	101	7454	3667	8617	18741
124	role-strategy-plugin.git	5,181	88	1	425	144	730	1152
125	shinyproxy.git	1,846	36	0	170	99	288	498
126	maven-mvnd.git	15,861	297	20	1170	635	1557	4145
127	dexmaker.git	9,913	214	19	816	251	719	1940
128	eventmesh.git	63,823	1467	134	6553	3921	8794	17874
129	testing-samples.git	1,809	83	1	239	148	468	684
130	talkback.git	73,288	1046	136	6103	4127	7691	13650
131	openvsx.git	24,435	315	18	3396	1518	3272	6203
132	curator.git	49,161	1088	248	4666	1565	6648	11140
133	spring-ws.git	33,72	848	97	4244	1364	5234	8368
134	wildwebdeveloper.git	8,808	236	10	622	730	1293	2973
135	cargotracker.git	3,999	104	11	540	370	844	910
136	spring-cloud-vault.git	7,97	254	12	956	520	1980	2288
137	zuul.git	20,082	421	44	2248	1217	3039	4995
138	aries.git	144,337	2556	405	13174	5844	13485	29946
139	pgql-lang.git	14,228	186	20	1634	461	1360	1810
140	parquet-java.git	117,021	2003	97	10274	4371	12961	21093
141	jira-plugin.git	9,794	194	5	985	517	1593	2592
142	spring-cloud-stream.git	43,904	945	54	3749	1410	6743	9686
143	k-NN.git	5,976	132	2	632	279	951	1686
144	xtext-eclipse.git	777,036	6571	768	56793	27005	49463	82682
145	mapstruct.git	94,115	3394	1084	14152	5802	17205	15861
146	closure-templates.git	124,203	2127	175	12407	4288	19126	29262
147	workflow-cps-plugin.git	26,661	662	17	2426	899	3891	5700
148	cayenne.git	229,744	4379	306	25661	9842	27130	51356
149	spring-authorization-server.git	31,174	542	16	2760	1453	5893	11402
150	gcm.git	6,768	119	2	462	368	693	1591
151	build-monitor-plugin.git	5,359	254	7	876	371	1333	1986
152	ceylon-ide-eclipse.git	129,07	2032	50	8589	3511	16501	28238
153	steady.git	94,342	1179	110	7861	4038	9656	20826
154	spring-shell.git	46,74	1092	150	4406	2203	5395	8638
155	piranha.git	4,337	98	6	396	148	322	573
156	flink-statefun.git	23,471	674	80	2747	1347	4289	5857
157	incubator-uniffle.git	59,884	926	44	4551	3603	7412	16742
158	tabris.git	29,772	527	56	4167	1145	3539	6127
159	spring-cloud-config.git	25,568	606	23	2561	1007	4165	6025
160	eclemma.git	8,741	241	14	969	554	1464	2076

1	2	3	4	5	6	7	8	9
161	spring-data-cassandra.git	54,114	1297	237	7156	2337	8755	12008
162	jib.git	34,823	651	46	3712	1905	4825	11298
163	gradle-plugin.git	3,974	109	9	421	255	691	1115
164	heroic.git	41,364	985	289	4836	2513	9174	10852
165	commons-imaging.git	37,644	667	26	2708	2428	3318	5291
166	submarine.git	112,035	615	74	6822	2386	4913	10340
167	spring-cloud-connectors.git	7,713	305	15	1058	345	1602	1951
168	facebook-java-business-sdk.git	596,287	4358	8	74200	9825	25872	63202
169	pignlproc.git	1,937	30	0	132	72	175	396
170	error-prone.git	121,757	4033	189	14680	3829	24202	35728
171	hk2.git	108,464	2557	554	10127	4495	13609	17647
172	quickstart-android.git	6,665	229	3	510	295	1203	2197
173	tomcat80.git	311,698	3486	303	28126	12565	16412	49782
174	spring-cloud-kubernetes.git	29,329	1021	15	3099	1793	7677	10964
175	incubator-paimon.git	153,871	2650	284	14437	6618	21837	40974
176	httpcomponents-client.git	60,461	969	77	5904	2132	7237	11423
177	grobid.git	58,864	524	19	3695	2397	2825	9378
178	fresco.git	67,449	1204	128	6796	3828	9215	12970
179	spring-amqp.git	61,18	916	89	5597	2331	6874	12894
180	rabbitmq-perf-test.git	14,095	198	29	1175	678	1186	3213
181	giraph.git	78,35	1778	331	9174	3382	12241	16270
182	elk.git	249,782	1488	140	13364	7467	11397	28422
183	inlong.git	256,154	4469	406	21235	21736	35223	65846
184	analysis-model.git	20,43	576	5	2516	1369	3041	5996
185	commons-configuration.git	46,795	576	67	5754	1406	4062	10700
186	phoenix.git	335,79	3411	168	23863	10193	26011	60944
187	re2j.git	8,447	89	3	454	413	375	1101
188	ribbon.git	21,788	648	62	2256	1045	3115	4668
189	butterknife.git	7,712	258	14	558	368	1081	1459
190	flink-kubernetes-operator.git	27,69	508	27	2089	1418	4608	9413
191	LiquidFunPaint.git	2,146	38	0	169	201	199	546
192	distributedlog.git	69,449	1138	117	5402	2866	6579	13633
193	tm4e.git	18,736	407	55	1779	1007	2568	4195
194	spring-batch-extensions.git	3,251	90	7	355	200	556	1024
195	rocketmq-dashboard.git	7,984	149	16	930	419	1151	2337
196	spring-hateoas.git	23,315	688	74	2973	1141	4557	6482
197	FirebaseUI-Android.git	13,829	273	16	1293	738	2923	4011
198	oozie.git	172,651	2441	51	13044	5762	14003	38518
199	spring-data-gremlin.git	5,162	157	35	625	578	1347	1412
200	start.spring.io.git	5,452	249	3	713	197	1439	2248
201	migrations.git	5,179	85	12	488	215	360	1224
202	zeppelin.git	193,873	2134	47	12689	4706	12123	28962

1	2	3	4	5	6	7	8	9
203	MOE.git	11,059	295	35	1246	707	2422	3291
204	ytk-learn.git	20,453	227	11	1067	995	1271	2714
205	incubator-baremaps.git	17,332	403	21	2084	1069	2570	4664
206	graphicsfuzz.git	43,878	826	34	4069	1374	5905	10649
207	Hystrix.git	62,01	1960	45	5376	1590	6708	9879
208	mina-sshd.git	118,018	1737	317	12186	4506	11565	27846
209	dagger.git	8,017	398	7	699	333	1288	1389
210	spring-graphql.git	22,829	554	125	2527	868	4789	6405
211	maven-hpi-plugin.git	3,364	56	1	164	145	401	1161
212	jersey.git	248,996	7396	334	25207	7999	36896	46200
213	ChatKit.git	4,68	82	26	529	304	623	925
214	p4-plugin.git	16,422	264	4	1467	622	2622	3460
215	jackrabbit.git	334,847	3952	430	26667	11429	28124	55924
216	spring-cloud-bus.git	2,071	80	4	252	116	514	471
217	Priam.git	12,056	274	32	1503	800	1725	3043
218	spring.git	5,083	146	20	591	207	812	1150
219	double-espresso.git	12,234	379	22	1106	436	1617	2656
220	tfs-plugin.git	12,888	287	9	1416	553	1707	2702
221	sqoop.git	69,624	871	45	5873	2725	6544	15078
222	hubot-steps-plugin.git	1,645	37	1	95	117	284	465
223	linkis.git	100,169	2168	431	13663	6366	12675	23754
224	datumbox-framework.git	21,604	392	18	1617	409	2364	4655
225	calcite-avatica.git	42,799	617	60	4343	1368	3797	6637
226	stream-applications.git	27,959	904	22	2902	1648	5892	8558
227	views-widgets-samples.git	7,334	146	6	579	428	612	1208
228	rabbitmq-jms-client.git	17,617	281	25	1785	810	1830	3942
229	cassandra-java-driver.git	113,397	1993	474	13610	5303	18574	30330
230	StringPacks.git	2,249	66	3	177	83	202	333
231	screenshot-tests-for-android.git	3,31	67	5	319	132	361	778
232	ApplicationInsights-Android.git	8,419	128	4	1031	361	409	1484
233	maven-plugin.git	15,348	281	10	1330	557	2237	3573
234	freemarker-scripting.git	1,303	29	9	146	53	207	363
235	tomee.git	460,011	8570	929	44607	16455	51121	82416
236	firebase-admin-java.git	69,433	1481	66	6824	2092	6754	12814
237	allocation-instrumenter.git	1,459	33	2	99	56	106	208
238	spring-data-ldap.git	1,892	60	17	255	115	536	505
239	email-ext-plugin.git	11,048	226	3	1059	332	1764	2515
240	spring-cloud-stream-binder-rabbit.git	5,592	87	1	415	180	610	1158
241	spring-data-elasticsearch.git	51,23	974	124	6321	2424	7372	9638
242	metro-jax-ws.git	169,078	3388	351	17418	7266	19786	31320
243	assertj-android.git	14,334	350	0	2752	6	2115	3363
244	mina.git	28,779	466	47	2577	884	2389	4054

1	2	3	4	5	6	7	8	9
245	samza.git	110,291	2071	246	9715	6402	15860	27234
246	google-api-java-client-samples.git	7,853	218	7	586	414	1337	3118
247	microprofile-conference.git	2,854	78	2	379	135	473	915
248	spring-cloud-openfeign.git	13,688	406	140	1569	467	3283	3023
249	CassJMeter.git	2,445	50	1	303	193	294	732
250	stickers.git	1,811	29	1	87	138	209	444
251	jadx.git	114,472	2412	152	9393	4254	11433	27095
252	bitbucket-branch-source-plugin.git	18,034	302	19	1622	632	2473	3359
253	smile.git	98,284	1157	143	9615	2944	7026	13860
254	dingtalk-plugin.git	2,037	56	1	146	174	306	559
255	ffwd.git	78,03	471	85	4598	1630	3949	6966
256	android-classyshark.git	8,714	198	14	734	464	672	1816
257	android-maps-utils.git	13,329	194	18	1214	526	1431	2853
258	flink-ml.git	41,079	903	131	3703	1935	8419	12111
259	pyramid.git	47,598	605	28	3408	1749	4188	11122
260	dart2java.git	17,578	568	154	3268	893	2949	2149
261	botbuilder-java.git	90,832	1743	110	9954	6687	8154	17249
262	eagle.git	104,937	2107	206	10134	6259	12189	23892
263	EasyML.git	29,343	627	50	2849	1177	3012	6346
264	EdXposed.git	71,794	854	66	6423	3774	5268	10596
265	activemq-apollo.git	27,918	565	41	2879	1505	2365	4289
266	datahub-gma.git	20,01	336	14	1934	639	3368	5969
267	sofa-registry.git	95,228	1638	213	10402	4429	10678	21378
268	microprofile-reactive-streams-operators.git	5,392	152	47	955	80	1102	1399
269	tomcat.git	363,664	4367	361	31781	14262	21471	59339
270	walt.git	5,83	115	4	438	372	655	1374
271	crate.git	438,265	7218	402	42594	16031	56759	101175
272	deltaspikes.git	48,45	1688	312	5722	2364	9835	12127
273	cdep.git	11,178	227	1	839	417	1452	2034
274	camel-kafka-connector.git	28,525	872	17	2815	7411	4513	5396
275	spring-data-keyvalue.git	4,522	122	22	733	202	786	996
276	spring-data-jpa.git	34,706	656	115	4639	1198	5636	6951
277	suro.git	29,129	613	28	2273	1207	3024	4883
278	spring-cloud-stream-binder-kafka.git	15,121	268	12	1128	543	2312	3480
279	maven-plugins.git	96,288	1478	53	7743	4043	7458	22047
280	MVI-Dispatcher.git	2,512	68	13	323	141	344	532
281	warnings-plugin.git	7,908	222	0	856	484	1291	2145
282	perforce-plugin.git	13,475	118	6	1233	460	699	2420
283	directory-kerby.git	37,895	984	38	4911	1223	4567	7977
284	maven.git	78,361	1306	349	9237	2996	9849	18221
285	rocketmq-externals.git	9,956	236	17	1030	798	1461	2544

**ДОДАТОК В - НАВЧАЛЬНИЙ ТА ТЕСТОВИЙ НАБОРИ ДАНИХ DATA
SCIENCE TA MACHINE LEARNING JAVA-ЗАСТОСУНКІВ З
ВІДКРИТИМ ПРОГРАМНИМ КОДОМ**

Таблиця В.1 – Навчальна вибірка інформації з метрик Data Science та Machine Learning JAVA-застосунків

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	2	3	4	5	6	7	8
1	atlas	154,568	1757	89	11778	14096	12936
2	curator	49,161	1088	248	4666	5087	6648
3	hama	47,105	667	82	4196	4540	4538
4	helix	156,854	2282	144	11188	12677	15681
5	iceberg	258,993	3667	316	23662	27844	34060
6	incubator-celeborn	28,49	396	24	2379	2693	3159
7	incubator-hivemall	76,739	1022	34	5340	5927	7004
8	incubator-myriad	8,056	194	6	1018	1120	1414
9	incubator-nemo	32,999	812	62	3060	3635	6060
10	incubator-retired-edgent	20,371	423	66	2179	2436	2554
11	incubator-seatunnel	105,104	2455	210	9591	11094	16794
12	inlong	256,154	4469	406	21235	24284	35223
13	mahout	95,439	1727	90	8306	9312	11144
14	opennlp	64,504	1049	182	4765	5471	6746
15	sqoop	69,624	871	45	5873	6645	6544
16	submarine	112,035	615	74	6822	7316	4913
17	mlkit	9,181	168	6	581	758	1316
18	datahub-gma	20,01	336	14	1934	2214	3368
19	dr-elephant	19,825	342	9	1384	1747	1946
20	spring-data-elasticsearch	51,23	974	124	6321	6734	7372
21	Data Science with Java	2,804	87	7	308	322	482
22	data-science-in-action	9,741	245	1	656	701	1231
23	angel	204,252	1736	145	14443	15628	14196
24	alluxio	222,656	3343	240	20901	23123	24063
25	smile	98,284	1157	143	9615	10045	7026
26	Alink	458,003	6488	1386	33458	35701	51272
27	grobid	58,864	524	19	3695	3942	2825
28	EasyML	29,343	627	50	2849	2975	3012
29	oryx	14,469	392	22	1447	1628	2834
30	seldon-server	29,704	601	60	3229	3470	3206
31	elasticsearch-learning-to-rank	15,737	280	18	1520	1627	2286
32	spatial-framework-for-hadoop	5,754	148	0	386	424	1250
33	tribuo	288,533	1554	278	24020	25939	18037
34	datumbox-framework	21,604	392	18	1617	1778	2364
35	Mallet	70,037	766	64	5571	6253	5671
36	cv4j	7,301	136	4	441	515	606

1	2	3	4	5	6	7	8
37	kafka-streams-machine-learning-examples	196,02	112	0	178	172	370
38	JSAT	87,712	1104	71	10252	10733	8468
39	ABAGAIL	12,985	293	31	1405	1486	1903
40	h2o-3	362,876	5155	187	23160	25960	31767
41	moa	107,002	1408	108	8329	9035	8270
42	codebuff	119,496	2124	101	12826	13675	12775
43	trident-ml	4,549	92	8	495	520	524
44	Conjecture	4,595	63	8	742	764	304
45	dagli	137,599	2367	605	16302	17556	37418
46	ytk-learn	20,453	227	11	1067	1222	1271
47	k-NN	5,976	132	2	632	680	951
48	flink-ml	41,079	903	131	3703	4045	8419
49	shifu	61,27	765	20	4306	5078	4674
50	antimalwareapp	1,649	35	0	88	107	228
51	entity-fishing	29,591	255	1	2080	2234	1783
52	FakeImageDetection	2,161	44	7	110	154	278
53	TextAnalyzer	6,885	157	15	711	793	497
54	pyramid	47,598	605	28	3408	4423	4188
55	pignlproc	1,937	30	0	132	137	175
56	DL-Learner	88,46	914	101	7454	8589	8617
57	tsml-java	290,27	1894	157	22467	23833	14621
58	Java-Machine-Learning	3,539	66	15	407	409	307
59	mltk	19,314	206	10	1288	1304	1313
60	cleartk	35,976	908	65	2810	3066	5530
61	Foundry	160,264	2162	286	17224	17542	14937
62	hudi	230,858	3341	181	19467	22962	31783
63	morpheus-core	50,934	487	76	4520	4826	4025
64	systemds	339,938	3664	72	31993	36808	30498
65	shinyproxy	1,846	36	0	170	185	288
66	vespa	624,258	11230	689	61030	74386	73651
67	djl-serving	12,676	173	5	812	1016	1200
68	serving	224,262	535	112	11526	12481	5496
69	toolbox	43,157	684	41	4030	4208	5787
70	pattern	3,146	128	2	421	454	654
71	graph-data-science	200,338	4536	1083	21631	24647	35804
72	nexus-repository-r	2,877	74	4	311	353	697
73	ml	4,216	134	5	473	502	727
74	systemds	207,825	2329	12	18941	21583	18056

Таблиця В.2 – Тестовий вибірка інформації з метрик Data Science та Machine Learning JAVA-застосунків

№	Репозиторій	KLOC	CLS	INFC	VMQ	TFQ	CBO
1	2	3	4	5	6	7	8
1	mastering-java-data-science	11,977	242	2	895	358	1635

1	2	3	4	5	6	7	8
2	bigtop	6,637	223	12	638	519	1114
3	java-data-mining-package	30,378	613	36	3870	1130	3705
4	h2o-2	119,202	1721	20	8546	4758	10532
5	MiA	3,847	117	5	252	137	706
6	mahout	97,394	1754	100	8195	3743	12033
7	OpenRefine	72,46	1353	44	6943	3298	9007
8	tablesaw	48,446	684	82	6116	1811	4634
9	odd-platform	23,603	680	227	3009	1580	8067
10	DataCleaner	106,729	1916	248	10020	5135	15432
11	datavines	31,339	826	116	3502	2376	4822
12	CQL	101,763	1082	60	8150	9368	13630
13	rumble	50,978	730	6	4915	2250	6711
14	blockchain2graph	3,035	49	8	372	235	325
15	djl	83,615	1326	79	9312	2908	9682
16	serve	18,747	237	14	1818	901	1456
17	modeldb	54,516	442	34	4317	1774	5885
18	conquary	44,793	1397	155	5376	3105	14237
19	eskimo	30,099	571	41	2835	1193	3777
20	jvector	18,01	234	31	1812	581	1726
21	AndroidTensorFlowMachineLearningExample	0,339	12	1	32	32	41
22	qupath	158,8	1575	118	10989	5771	14118
23	MLKit	0,484	50	0	84	13	241
24	jenetics	40,895	844	106	5429	1418	5828
25	Android-TensorFlow-Lite-Example	0,355	13	1	27	24	43
26	TonY	12,534	174	24	1020	867	1238
27	hms-ml-demo	43,232	803	37	2602	3036	4611
28	product-category-predict	0,424	14	0	36	41	110
29	ml-commons	109,627	1517	39	8323	5956	16445
30	car-OCR	1,498	21	2	128	73	110
31	aibolit	112,751	1151	42	6492	3797	6666
32	k-means-mapreduce	0,394	6	0	35	16	37
33	spark-transformers	4,328	173	4	410	155	1303
34	BoostSRL	89,489	525	31	5971	3149	4490
35	DigitRecognizer	0,844	16	0	47	53	80
36	DeepLearning	1,083	11	0	60	71	50
37	onyx	10,392	207	11	814	410	790
38	opencv-object-detection	0,35	7	0	25	15	36

ДОДАТОК Г - АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ



ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ "Н-ІКС
ДЕЛІВЕРІ"
LIMITED LIABILITY COMPANY N-IX DELIVERY

82660, Львівська обл., Сколівський район, смт. Славське, вул. Привокзальна, 23
т/факс +380322332504, +380322295929, email: contact@n-ix.com, https://www.n-ix.com/

Вих. №1-1/01-09/2025 від 01.09.2025 р.

АКТ

впровадження результатів дисертаційної роботи на здобуття наукового ступеня
доктора філософії (PhD) в галузі знань 12 «Інформаційні технології»
за спеціальністю 122 «Комп'ютерні науки»
Орехова Олександра Станіславовича

**«Математичні моделі та інформаційна технологія для обробки інформації з метрик
програмних застосунків, які розроблені за допомогою мови програмування JAVA»**

Я, що нижче підписався, директор ТОВ "Н-ІКС ДЕЛІВЕРІ" Мироненко Вадим Вікторович, цим Актом засвідчую, що результати дисертаційної роботи Орехова Олександра Станіславовича, а саме: інструментарій інформаційної технології для оцінювання параметру розміру рядків коду JAVA-застосунків на основі вибірок з відкритим кодом загальних JAVA-застосунків, та Data Science та Machine Learning JAVA-застосунків "Сервіс оцінювання інформації з метрик програмних застосунків" було впроваджено в процеси управління проектами з розробки програмного забезпечення мовою програмування JAVA.

Впровадження, зазначеного вище результату, дозволило підвищити достовірність обробки інформації для оцінювання параметру розміру вищезазначених JAVA-застосунків на ранніх стадіях їх розробки.

Директор ТОВ "Н-ІКС
ДЕЛІВЕРІ"



Мироненко В. В.

ЗАТВЕРДЖУЮ
ФОП Поліщук Андрій
Борисович

«31» жовтня 2025 р.

АКТ

впровадження результатів дисертаційної роботи на здобуття наукового ступеня
доктора філософії (PhD) в галузі знань 12 «Інформаційні технології»
за спеціальністю 122 «Комп'ютерні науки»
Орехова Олександра Станіславовича
«Математичні моделі та інформаційна технологія для обробки інформації з
метрик програмних застосунків, які розроблені за допомогою мови
програмування JAVA»

Я, Поліщук Андрій Борисович, цим Актом засвідчую, що результати дисертації Орехова Олександра Станіславовича, а саме: інструментарій інформаційної технології для оцінювання параметру розміру рядків коду загальних JAVA-застосунків та Data Science і Machine Learning JAVA-застосунків “Сервіс оцінювання інформації з метрик програмних застосунків” було впроваджено в процес розробки проєктів програмного забезпечення для збору та обробки інформації від туристичних операторів, які були розроблені мовою програмування JAVA.

Впровадження, зазначеного вище результату, дозволило підвищити достовірність обробки інформації для оцінювання параметру розміру вищезазначених JAVA-застосунків на ранніх стадіях їх розробки.



Поліщук Андрій Борисович

ДОДАТОК Д - АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ В НАВЧАЛЬНИЙ ПРОЦЕС



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
 ІМЕНІ АДМІРАЛА МАКАРОВА**
 54007, м. Миколаїв, пр. Героїв України 9, тел./факс +38(0512) 42-42-80, e-mail: university@nuos.edu.ua

«ЗАТВЕРДЖУЮ»

Ректор Національного університету
 кораблебудування ім. адмірала Макарова

проф. Євген ТРУШЛЯКОВ
 _____ 2025 р.



впровадження результатів дисертаційної роботи на здобуття наукового ступеня
 доктора філософії (PhD) в галузі знань 12 «Інформаційні технології»
 за спеціальністю 122 «Комп'ютерні науки»
 Орехова Олександра Станіславовича

Результати дисертаційної роботи Орехов О. С. дійсно використовуються при викладанні студентам Навчально-наукового інституту комп'ютерних наук та управління проектами (ННІКНУП) наступних дисциплін:

- «Керування проєктними ресурсам в інженерії програмного забезпечення» в ОПІ «Інженерія програмного забезпечення» для 2-го магістерського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»;
- «Основи програмної інженерії» в ОПІ «Інженерія програмного забезпечення» для 1-го бакалаврського рівня ВО, спеціальності 121 (F2) «Інженерія програмного забезпечення», галузі знань 12 (F) «Інформаційні технології»;
- «Моделювання систем цифрового екологічного моніторингу» в ОПІ «Цифровий екологічний моніторинг» для 2-го магістерського рівня ВО, спеціальності E2 «Екологія», галузі знань E «Природничі науки, математика та статистика».

Проректор з НІПР НУК
 к.т.н., проф. НУК

Сергій СЛОБОДЯН

ДОДАТОК Е - СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Список публікацій здобувача за темою дисертації:

1) в яких опубліковані основні наукові результати дисертації:

1. **Орехов, О. С., & Фаріонова, Т. А.** (2024). Аналіз сучасного стану методів оцінювання трудомісткості розробки програмного забезпечення. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 1(494), 102–111. DOI: [https://doi.org/10.15589/znp2024.1\(494\).15](https://doi.org/10.15589/znp2024.1(494).15) (Реєстр наукових фахових видань України, категорія «Б»).

2. **Орехов О. С., & Фаріонова Т. А.** (2024). Математичні моделі для оцінювання розміру JAVA-застосунків. *Вісник Херсонського національного технічного університету*, 2(89), 196-203. DOI: <https://doi.org/10.35546/kntu2078-4481.2024.2.28> (Реєстр наукових фахових видань України, категорія «Б»).

3. **Орехов, О. С.** (2024). Чотирьохфакторна нелінійна регресійна модель для оцінювання розміру JAVA-застосунків на ранніх стадіях розробки. У В. В. Вичужанін (ред.), *Розвитки інформаційно-керуючих систем та технологій* (с. 360–379). Liha-Pres. <https://doi.org/10.36059/978-966-397-422-4> (ISBN 978-966-397-422-4, одноосібний розділ в колективній монографії)

4. **Орехов, О. С., & Фаріонова, Т. А.** (2025). Математична модель для обробки інформації з метрик коду Data Science та Machine Learning JAVA-застосунків для оцінювання їх розміру. *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*, 2(500), 275–285. DOI: [https://doi.org/10.15589/znp2025.2\(500\).37](https://doi.org/10.15589/znp2025.2(500).37) (Реєстр наукових фахових видань України, категорія «Б»).

5. **Орехов, О. С., & Фаріонова, Т. А.** (2025). П'ятифакторна математична модель для обробки інформації з метрик коду JAVA-застосунків для оцінювання

їх розміру. *Вісник Херсонського національного технічного університету*, 4(94), 139–146. DOI: <https://doi.org/10.35546/kntu2078-4481.2025.4.2.18> (Реєстр наукових фахових видань України, категорія «Б»).

2) які засвідчують апробацію матеріалів дисертації:

6. Ворона, М. В., & Орєхов, О. С. (2023). Методи оцінки трудомісткості розробки програмного забезпечення для agile проєктів. *Матеріали XIV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (20-21 вересня 2023). Миколаїв: НУК імені адмірала Макарова. 438-441.

7. Фаріонова Т., & Орєхов О. (2023). Аналіз успішності розробки проєктів програмного забезпечення. *Матеріали міжнародної науково-практичної конференції "Інтелектуальні інформаційні системи в управлінні проєктами та програмами"* (12-15 вересня 2023) Харків: ХНУРЕ. 210–213. DOI: <https://doi.org/10.30837/IISRRM.2023.09>

8. Орєхов, О. С., & Фаріонова, Т. А. (2023). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Data Science та machine learning проєктів, які розробляються за допомогою мови програмування Java. *Матеріали IV всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2023)* (30-31 жовтня 2023 р.). Миколаїв: НУК імені адмірала Макарова. 45–47.

9. Орєхов, О. С., & Фаріонова, Т. А. (2024). Порівняльний аналіз нелінійних регресійних моделей оцінювання кількості рядків коду JAVA-застосунків. *Матеріали XV науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (26-27 вересня 2024). Миколаїв: НУК імені адмірала Макарова. 563 – 566.

10. Oriekhov, O., Farionova, T., Chernova, L., Chernova, L., & Vorona, M. (2024). Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications. *5th International Workshop on IT Project Management*, Bratislava, Slovakia, May 22, 2024. CEUR Workshop Proceedings, Vol. 3709, pp. 54-66. DOI: https://doi.org/10.23939/IW_itpm2024.054 (ISSN

1613-0073. Стаття включена до міжнародних наукометричних баз Scopus та DBLP).

11. **Oriekhov, O.**, Farionova, T., & Chernova, L. (2024), Three-factor nonlinear regression model of estimating the size of JAVA-software. *In Proceedings of the International Conference on Information Control Systems and Technologies (ICST-2024)* (September 23-25). CEUR-WS. 506-518. <https://ceur-ws.org/Vol-3790/paper44.pdf>

12. **Oriekhov, O.**, & Farionova, T. (2024). Early KLOC estimation of JAVA-applications by the three-factor nonlinear regression model. In V. Vychuzhanin (Ed.), *Materials of the XI International Scientific and Practical Conference "Information Control Systems and Technologies 2024" (ICST-2024) (23-25 September)*. Odesa: Helvetica Publishing House. pp. 310 – 312. <https://icst-conf.com/2024.pdf>

13. **Орехов, О. С.**, & Фаріонова, Т. А. (2024). Оцінювання розміру JAVA-застосунків за трьохфакторною нелінійною регресійною моделлю. *Матеріали V всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2024) (30-31 жовтня 2024 р.)*. Миколаїв: НУК імені адмірала Макарова. 39–42.

14. **Oriekhov, O.**, Farionova, T., Chernova, L., & Vorona, M. (2025). A Five-Factor Nonlinear Regression Model for JAVA Applications Size Estimation. *6th International Workshop on IT Project Management*, Kyiv, Ukraine, May 22, 2025. CEUR Workshop Proceedings, Vol. 4023, pp. 1-10. <https://ceur-ws.org/Vol-4023/paper1.pdf> (ISSN 1613-0073. Стаття включена до міжнародних наукометричних баз Scopus та DBLP).

15. **Орехов, О. С.**, & Фаріонова, Т. А. (2025). Сучасні інформаційні технології для обробки метрик коду на ранніх етапах проектування програмного забезпечення. *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці" (25-26 вересня 2025)*. Миколаїв: НУК імені адмірала Макарова. 588 - 590.

16. **Орєхов, О. С., & Ворона, М. В.** (2025). Проєкт інформаційної технології для обробки інформації з метрик коду JAVA-застосунків. *Матеріали XVI науково-технічної конференції "Інновації в суднобудуванні та океанотехніці"* (25-26 вересня 2025). Миколаїв: НУК імені адмірала Макарова. 604 - 607.

17. **Орєхов, О. С., & Фаріонова, Т. А.** (2025). Механізми зменшення мультиколінеарності між незалежними факторами при побудові нелінійної регресійної моделі для оцінювання JAVA-застосунків. *Матеріали VI всеукраїнської науково-практичної інтернет конференції "Інформаційні технології: моделі, алгоритми, системи" (ITMAS – 2025)* (16-17 листопада 2025). Миколаїв: НУК імені адмірала Макарова. 40–42.

3) які додатково відображають наукові результати дисертації:

18. **Орєхов, О. С., & Фаріонова, Т. А.** (2025). Комп'ютерна програма «Сервіс оцінювання інформації з метрик програмних застосунків» (SoftwareMetricsEstimator) [Свідоцтво про реєстрацію авторського права № 135110 від 15.04.2025]. У *Бюлетень з авторського права та суміжних прав*, № 89, с. 139.
https://nipo.gov.ua/wp-content/uploads/2025/05/BULETEN_Avt_Pravo_-89-2025.zip

19. **Oriekhov, O., & Farionova, T.** (2025). Information technology for estimating the size of software projects. In I. Linde (Ed.), *Innovative technologies for project and program management* (pp. 192-201).. Ryha (Latvia): European University Press. <https://mmp-conf.org/documents/archive/monography2025.pdf> (ISBN 978-9984-891-34-7, розділ в колективній монографії)

ДОДАТОК Є - ІНСТРУКЦІЯ КОРИСТУВАЧА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ “СЕРВІС ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК ПРОГРАМНИХ ЗАСТОСУНКІВ”

1 Загальні положення

Ця інструкція описує порядок виконання HTTP-запитів до бекенд-сервера комп'ютерної програми “Сервіс оцінювання інформації з метрик програмних застосунків” (далі - Сервіс), включаючи методи запитів, аутентифікацію, авторизацію, обробку помилок та приклади використання.

Обчислення інформації з метрик JAVA-застосунків розраховане наявність активного підключення до мережі Інтернет, використання відповідного програмного забезпечення та програмних рішень на стороні клієнта та наявності ключів доступу до обчислювальних потужностей веб-застосунку.

2 Базова URL-адреса

Доступ до програмного забезпечення “Сервіс оцінювання інформації з метрик програмних застосунків” здійснюється за веб адресою розташування серверу програмного забезпечення. Веб адреса доступу до Сервісу надається при укладанні договору про надання послуг з оцінювання інформації з метрик програмних застосунків. Доступ до Сервісу здійснюється за протоколом HTTP. Базова URL-адреса позначаються як `<Base Url>` в цій інструкції.

3 Аутентифікація та авторизація

Для забезпечення безпеки доступу до сервісу всі запити потребують аутентифікації та авторизації. Авторизовані запити до Сервісу здійснюються за допомогою ключа доступу, який додається до кожного HTTP запиту в секції заголовку. Приклад заголовку авторизації наведено у лістингу нижче:

`X-API-Key: <Api Key>,`

Де “X-API-Key” - ім'я параметра заголовку та “<Api Key>” - значення ключа доступу.

Ключ доступу “<Api Key>” надається при укладанні договору про надання послуг з оцінювання інформації з метрик програмних застосунків.

4 HTTP Запити

Цей розділ містить опис основних HTTP-запитів, які надає Сервіс для отримання переліку доступних математичних моделей, отримання інформації про математичну модель за її ідентифікатором та виконання обчислень інформації з метрик програмних застосунків за допомогою математичної моделі за її ідентифікатором.

Кожен запит має визначений набір доступних HTTP методів, а також набір параметрів, які впливають на його виконання. Для успішного використання API необхідно дотримуватись вимог щодо аутентифікації та авторизації, вхідних даних та форматування запитів. Усі відповіді повертаються у стандартизованому форматі у вигляді JSON, що спрощує їхню обробку та інтеграцію в інші системи. Для кожного веб запиту рекомендовано додавати заголовок "Accept: application/json".

Запити повинні відповідати стандартам HTTP-протоколу, які детально описані в стандарті “RFC 9110: HTTP Semantics”, розташованому за адресою “<https://datatracker.ietf.org/doc/html/rfc9110>”.

Приклади запитів до Сервісу надаються у вигляді запитів командного рядку утиліти cURL “<https://curl.se/>”.

4.1 HTTP запит отримання списку математичних моделей.

Для отримання списку відповідних математичних моделей необхідно виконати HTTP-запит за шляхом доступу до списку всіх наявних математичних моделей вказавши параметри запиту.

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models.

HTTP-Метод: GET.

Вимоги до HTTP заголовку - ключ доступу “<Api Key>”.

Вимоги до параметрів запиту:

Параметр `category` є обов'язковим. Список доступних значень параметра `category`: "software-size".

Параметр `lang` не є обов'язковим. Список доступних значень параметра `lang`: "JAVA".

Лістинг Є.4.1.1. CURL команда HTTP-запиту на отримання списку математичних моделей оцінки розміру JAVA-застосунків:

```
curl -X GET \
<Base Url>/api/v1/math-models?category=software-size&lang=JAVA \
-H "X-Api-Key: <Api Key>" \
-H "Accept: application/json"
```

Результатом обробки запиту Сервісом буде список математичних моделей оцінювання інформації з програмного коду JAVA-застосунків у форматі JSON.

Лістинг Є.4.1.2. Результат обробки HTTP-запиту на отримання списку математичних моделей оцінювання інформації з метрик коду JAVA-застосунків:

```
[
  {
    "guid": "DB-kLoc-x3-class-rfc-avg_cls_visibleMethodsQty",
    "name": "Three-factor nonlinear regression model for KLOC estimation of Java software",
    "lang": "JAVA",
    "description": "Three-factor nonlinear regression model for KLOC estimation of Java software depends on factors total classes, total rfc and average quantity of class methods on basis of Box-Cox transformation",
    "category": "software-size",
    "factorsTotal": 3,
    "factors": [
      "class",
      "rfc",
      "avg_cls_visibleMethodsQty"
    ],
    "factorEstimating": "kloc",
    "createdAt": "2025-01-28",
    "updatedAt": "2025-01-28",
    "accuracy": [
      {
        "rsq": 0.89,
        "mmre": 0.22,
        "preds": {
          "0.25": 0.76
        },
        "sampleSize": 200,
        "sample": "training"
      }
    ]
  }
]
```

```

    }
  ],
  "restrictions": {
    "alphaIntervalTrust": [0.01,0.05,0.005,0.001,0.02,0.1],
    "alphaIntervalPrediction":
[0.01,0.05,0.005,0.001,0.02,0.1],
    "factors": [
      {
        "min": 1.0,
        "max": 99999.0,
        "factor": "class"
      }
    ]
  }
}
]

```

де кожна структура даних опису математичної моделі є елементом масиву та містить наступні поля:

"guid" - унікальний ідентифікатор моделі;

"category" - категорія математичної моделі;

"lang" - мова програмування для якої було розроблено математичну модель;

"factorsTotal" - загальна кількість незалежних факторів моделі;

"factors" - список незалежних факторів математичної моделі,

"factorEstimating" - незалежний фактор математичної моделі;

"createdAt" - дата та час створення математичної моделі в базі даних;

"updatedAt" - дата та час останнього оновлення математичної моделі в базі даних;

"accuracy" - оцінки критеріїв якості нелінійних регресійних моделей, таких як коефіцієнт детермінації, MMRE, PRED та характеристики наборів даних, на яких ці оцінки були отримані;

"restrictions" - поле містить перелік обмежень математичної моделі та включає список допустимих значень параметра рівня значущості α для інтервалів передбачення ("alphaIntervalPrediction"), довірчих інтервалів ("alphaIntervalTrust"), та обмеження діапазонів значень незалежних факторів моделі.

Для структури даних опису математичної моделі, поля "guid", "name", "lang", "description", "category", "factorsTotal", "factors", "factorEstimating", "createdAt", "updatedAt", "accuracy" є обов'язковими.

4.2 HTTP запит отримання математичної моделі за ідентифікатором.

Для отримання математичної моделі необхідно виконати HTTP-запит з наступними параметрами:

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models/<guid>.

HTTP-Метод: GET.

Вимоги до HTTP заголовку - ключ доступу "<Api Key>".

Вимоги до шляху запиту:

Фрагмент <guid> є обов'язковим та містить унікальний ідентифікатор існуючої математичної моделі (Щоб отримати список наявних математичних моделей див. пункт 4.1). Якщо фрагмент шляху <guid> посилається на неіснуючу математичну модель, результат запиту поверне помилку з HTTP кодом 404.

Лістинг Є.4.2.1. CURL команда HTTP-запиту на отримання списку математичних моделей оцінки розміру JAVA-застосунків:

```
curl -X GET \
<Base Url>/api/v1/math-models/<guid> \
-H "X-API-Key: <Api Key>" \
-H "Accept: application/json"
```

Результатом обробки запиту Сервісом буде відповідь у форматі JSON опису математичної моделі.

Лістинг Є.4.2.2. Результат обробки HTTP-запиту на отримання списку математичних моделей оцінювання інформації з метрик коду JAVA-застосунків:

```
{
  "guid": "DB-kLoc-x3-class-rfc-avg_cls_visibleMethodsQty",
  "name": "Three-factor nonlinear regression model for KLOC
estimation of Java software",
  "lang": "JAVA",
```

```

    "description": "Three-factor nonlinear regression model for
KLOC estimation of Java software depends on factors total
classes, total rfc and average quantity of class methods on
basis of Box-Cox transformation",
    "category": "software-size",
    "factorsTotal": 3,
    "factors": [
        "class",
        "rfc",
        "avg_cls_visibleMethodsQty"
    ],
    "factorEstimating": "kloc",
    "createdAt": "2025-01-28",
    "updatedAt": "2025-01-28",
    "accuracy": [
        {
            "rsq": 0.89,
            "mmre": 0.22,
            "preds": {
                "0.25": 0.76
            },
            "sampleSize": 200,
            "sample": "training"
        }
    ],
    "restrictions": {
        "alphaIntervalTrust": [0.01,0.05,0.005,0.001,0.02,0.1],
        "alphaIntervalPrediction":
[0.01,0.05,0.005,0.001,0.02,0.1],
        "factors": [
            {
                "min": 1.0,
                "max": 99999.0,
                "factor": "class"
            }
        ]
    }
}

```

Опис HTTP-відповіді наведено для лістингу Є.4.1.2.

4.3 HTTP запит обчислення інформації за допомогою математичної моделі за ідентифікатором.

Для отримання математичної моделі необхідно виконати HTTP-запит з наступними параметрами:

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models/<guid>/estimate.

HTTP-Метод: PUT.

Вимоги до HTTP заголовку - ключ доступу “<Api Key>”.

Вимоги до шляху запиту:

Фрагмент <guid> є обов’язковим та містить унікальний ідентифікатор існуючої математичної моделі (Щоб отримати список наявних математичних моделей див. пункт 4.1). Якщо фрагмент шляху <guid> посилається на неіснуючу математичну модель, результат запиту поверне помилку з HTTP кодом 404.

Вимоги до тіла (BODY) HTTP-запиту.

Тіла (BODY) HTTP-запиту є обов’язковим та відформатованим як JSON об’єкт. Формат JSON об’єкту HTTP-запиту наведено в Лістингу Є.4.3.1.

Лістинг Є.4.3.1. Лістинг тіла (BODY) HTTP-запиту

```
{
  "factors": {
    "class": 427.0,
    "rfc": 4855.0,
    "avg_cls_visibleMethodsQty": 6.60994
  },
  "interval_trust": [ "0.05", "0.01", "0.005" ],
  "interval_prediction": [ "0.05", "0.01", "0.005" ]
}
```

де "factors" - значення незалежних факторів моделі в у форматі ключ-значення, ключ - строкове значення існуючого незалежного фактору моделі та значення - числова величина незалежного фактору, список незалежних факторів моделі отримується за допомогою запитів 4.1 або 4.2; "interval_trust" та "interval_prediction" - перелік рівнів значущості α , для яких необхідно обчислити значення інтервалів передбачення ("interval_prediction") та довірчих інтервалів ("interval_trust").

Поля "interval_prediction" та "interval_trust" для структури 4.3.1 є обов’язковими.

Лістинг Є.4.3.2. CURL команда HTTP-запиту на обчислення інформації з метрик за допомогою математичної моделі:

```
curl --request PUT \
  --url '<Base Url>/api/v1/math-models/<guid>/estimate?=' \
  --header 'Content-Type: application/json' \
```

```
--header 'X-Api-Key: <Api Key>' \
--data '{
  "factors": {
    "class": 427.0,
    "rfc": 4855.0,
    "avg_cls_visibleMethodsQty": 6.60994
  },
  "interval_trust": [ "0.05" ],
  "interval_prediction": [ "0.05" ]
}'
```

Результатом обробки запиту Сервісом буде відповідь у форматі JSON опису математичної моделі.

Лістинг Є.4.3.3. Результат обробки HTTP-запиту обчислення інформації за допомогою математичної моделі за ідентифікатором <guid>:

```
{
  "factors": {
    "class": 427.0,
    "rfc": 4855.0,
    "avg_cls_visibleMethodsQty": 6.60994
  },
  "result": 24.60237315863393,
  "resultName": "kLoc",
  "intervalsTrust": [
    {
      "a": "0.05",
      "lb": 23.753700033855907,
      "ub": 25.48172611655633
    }
  ],
  "intervalsPrediction": [
    {
      "a": "0.05",
      "lb": 16.988965122020964,
      "ub": 35.68365288192322
    }
  ],
  "message": null
}
```

де "factors" - значення незалежних факторів моделі в у форматі ключ-значення відповідно до Лістингу Є.4.3.1; "result" - результати оцінювання за допомогою математичної моделі <guid> та значень вхідних параметрів "factors"; "resultName" - назва залежного фактору, оцінка якого отримана в полі "result"; "intervalsTrust" та "intervalsPrediction" - значення довірчих інтервалів та інтервалів передбачення, для яких "a" - рівень

значущості α , "lb" - значення нижньої границі інтервалу, "ub" - значення верхньої границі інтервалу; "message" - поле строки, яке може містити інформацію щодо результатів оцінювання та попередження.

4.4 HTTP запит додавання математичної моделі до Сервісу.

Для збереження нової математичної моделі в Сервісі необхідно виконати HTTP-запит з наступними параметрами:

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models.

HTTP-Метод: POST.

Вимоги до HTTP заголовку - ключ доступу "<Api Key>".

Вимоги до шляху запиту: відсутні.

Вимоги до тіла (BODY) HTTP-запиту.

Тіло (BODY) HTTP-запиту є обов'язковим та відформатованим як JSON об'єкт. Формат JSON об'єкту HTTP-запиту наведено в Лістингу Є.4.4.1.

Лістинг Є.4.4.1. Лістинг тіла (BODY) HTTP-запиту

```
{
  "guid": "<guid>",
  "name": "<Name>",
  "lang": "java",
  "category": "<Description>",
  "description": "<Description>",
  "factorEstimating": "kLoc",
  "publishedAt": "2022-01-01",
  "publishedReference": "http://localhost:8080/...",
  "model": {
    "components": [
      {
        "name": null,
        "type": "linear",
        "params": {
          "b0": 9.99,
          "class": 9.99,
          "rfc": 9.99,
          "visibleMethodsQty": 9.99
        }
      },
      {
        "name": "kLoc",
        "type": "box-cox",
        "params": {
```

```

        "l": -0.011409013604228004,
        "scale": 1.0
    },
    ...],
    "interval_config": {
        "means": [0.00,0.00,0.00],
        "sx_inv_matrix": [
            0.00,0.00,0.00,
            0.00,0.00,0.00,
            0.00,0.00,0.00
        ],
        "sy": 0.00,
        "degreesOfFreedom": 0.00
    },
    "type": "parametric",
    "target": "kLoc",
    "factors": ["class","rfc","visibleMethodsQty"],
    "factorsCount": 3
},
"restrictions": {
    "alphaIntervalTrust": [0.01, ...],
    "alphaIntervalPrediction": [0.05, ...],
    "factors": [
        {
            "min": 1.0,
            "max": 99999.0,
            "factor": "class"
        }
    ]
}
}

```

де "guid" - значення ідентифікатора моделі, може бути відсутнім, система не повинна містити в базі даних модель з таким ідентифікатором; "name" - назва математичної моделі, обов'язкове поле; "lang" - значення мови програмування, "java" - для мови програмування Java, обов'язкове поле; "category" - категорія математичної моделі, для оцінювання параметра розміру - "software-size", обов'язкове поле; "description" - текстовий опис математичної моделі, поле не обов'язкове; "publishedAt" - дата та час публікації моделі, поле не обов'язкове; "publishedReference" - посилання на джерело опублікованої моделі, поле не обов'язкове; "model" - JSON об'єкт з інформацією про параметри і тип математичної моделі, поле обов'язкове;

"restrictions" - JSON об'єкт з інформацією про обмеження математичної моделі, поле не обов'язкове.

Для JSON об'єкту "model", поле "type" - тип математичної моделі, де "parametric" - для нелінійної регресійної моделі та "linear" - для лінійної регресійної моделі; поле "target" - містить назву залежного фактору математичної моделі; поле "factors" - перелік факторів (незалежних змінних) математичної моделі, кількістю "k", масив рядків; "factorsCount" - загальні кількість факторів математичної моделі, кількість елементів масив повинен відповідати значенню цього поля - "k"; поле "components" - масив JSON об'єктів, кожен з яких містить інформацію про фактори та параметри математичної моделі. JSON об'єкт параметра моделі містить обов'язкові поля: назва фактору "name" або може бути відсутнє, тип фактору "type", який може мати значення: "linear" - параметри лінійної математичної моделі, "box-cox" - тип нормалізуючого перетворення Бокса-Кокса, "log10" - перетворення десятичного логарифма, "johnson-sb", "johnson-sl", "johnson-su" - перетворення Джонсона SB, SL та SU відповідно; "params" - JSON об'єк з динамічною кількістю полів рядкового типу ключ-значення, який містить параметри нормалізуючих перетворень - для перетворення Бокса-Кокса: "l", "scale", для перетворення Джонсона "n", "y", "l", "f" або параметри лінійної регресії (для типу "linear") - "b0", "<factor1>", "<factor2>", ..., "<factorK>"; поле "interval_config" - JSON об'єкт з інформацією параметрів інтервалів передбачення та довірчих інтервалів, та складається з наступних полів: "means" - масив числових значень розміром "k" із значеннями середніх незалежних факторів моделі. "sx_inv_matrix" - одновимірний числовий масив розміром "k*k", який представляє матрицю добутку векторів різниці середніх незалежних факторів регресії, "sy" - середньоквадратичне відхилення залежного фактору математичної моделі, "degreesOfFreedom" - числове значення степенів вільності.

Для JSON об'єкту "restrictions", поле "alphaIntervalTrust" - містить числовий масив допустимих значень рівня значущості α для обчислення довірчих інтервалів математичної моделі; поле "alphaIntervalPrediction" - містить числовий масив допустимих значень рівня значущості α для обчислення інтервалів передбачення математичної моделі; "factors" - містить масив JSON об'єктів з інформацією про діапазон допустимих значень незалежних факторів математичної моделі та складається з полів: "min" - мінімальне допустиме значення, "max" - максимальне допустиме значення та "factor" - назва фактора математичної моделі.

Лістинг Є.4.4.2. CURL команда HTTP-запиту на збереження математичної моделі для обчислення інформації з метрик програмних застосунків:

```
curl --request POST \
      --url '<Base Url>/api/v1/math-models' \
      --header 'Content-Type: application/json' \
      --header 'X-API-Key: <Api Key>' \
      --data '<JSON об'єкт відповідно до лістингу Є.4.4.1>'
```

Результатом обробки запиту Сервісом буде відповідь у форматі JSON інформації про збережену математичну модель в системи, відповідно до лістингу Є.4.2.2.

4.5 HTTP запит оновлення існуючої математичної моделі в Сервіс.

Для редагування математичної моделі в Сервісі необхідно виконати HTTP-запит з наступними параметрами:

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models/<guid>.

HTTP-Метод: PUT.

Вимоги до HTTP заголовку - ключ доступу "<Api Key>".

Вимоги до шляху запиту: Фрагмент <guid> є обов'язковим та містить унікальний ідентифікатор існуючої математичної моделі (Щоб отримати список наявних математичних моделей див. пункт 4.1). Якщо фрагмент шляху <guid>

посилається на неіснуючу математичну модель, результат запиту поверне помилку з HTTP кодом 404.

Вимоги до тіла (BODY) HTTP-запиту.

Тіло (BODY) HTTP-запиту є обов'язковим та відформатованим як JSON об'єкт. Формат JSON об'єкту HTTP-запиту наведено в Лістингу Є.4.4.1. На відміну від вимог до тіла запиту 4.4., всі поля тіла (BODY) JSON об'єкту HTTP-запиту не є обов'язковими. До полів які відсутні зміни не будуть застосовані.

Лістинг Є.4.5.1. CURL команда HTTP-запиту на редагування математичної моделі для обчислення інформації з метрик програмних застосунків:

```
curl --request PUT \
      --url '<Base Url>/api/v1/math-models/<guid>' \
      --header 'Content-Type: application/json' \
      --header 'X-API-Key: <Api Key>' \
      --data '<JSON об'єкт відповідно до лістингу Є.4.4.1>'
```

Результатом обробки запиту Сервісом буде відповідь відповідно до HTTP кодів результатів виконання.

4.6 HTTP запит на видалення існуючої математичної моделі із Сервісу.

Для видалення математичної моделі із Сервісу необхідно виконати HTTP-запит з наступними параметрами:

Шлях веб запиту має наступне значення:

<Base Url>/api/v1/math-models/<guid>.

HTTP-Метод: DELETE.

Вимоги до HTTP заголовку - ключ доступу “<Api Key>”.

Вимоги до шляху запиту: Фрагмент <guid> є обов'язковим та містить унікальний ідентифікатор існуючої математичної моделі (Щоб отримати список

наявних математичних моделей див. пункт 4.1). Якщо фрагмент шляху <guid> посилається на неіснуючу математичну модель, результат запиту поверне помилку з HTTP кодом 404.

Вимоги до тіла (BODY) HTTP-запиту відсутні.

Лістинг Є.4.6.1. CURL команда HTTP-запиту на видалення математичної моделі:

```
curl --request DELETE \
      --url '<Base Url>/api/v1/math-models/<guid>' \
      --header 'Content-Type: application/json' \
      --header 'X-API-Key: <Api Key>'
```

Результатом обробки запиту Сервісом буде відповідь відповідно до HTTP кодів результатів виконання.

5 Обробка помилок

Сервіс повертає HTTP статус-коди відповідно до результатів обробки запиту:

200 OK – Запит успішно виконано;

400 Bad Request – Невірні параметри запиту (підсутні вхідні параметри або в невірному форматі);

401 Unauthorized – Відсутній або неправильний ключ доступу;

403 Forbidden – Відсутні права доступу;

404 Not Found – Об'єкт не знайдено;

500 Internal Server Error – Внутрішня помилка сервера.

6 Допоміжні ресурси

1. RFC 9110: HTTP Semantics: <https://datatracker.ietf.org/doc/html/rfc9110>
2. cURL: <https://curl.se/>
3. The JavaScript Object Notation (JSON) Data Interchange Format: <https://datatracker.ietf.org/doc/html/rfc8259>

ДОДАТОК Ж - ОСНОВНІ МОДУЛІ ВИХІДНОГО КОДУ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ “СЕРВІС ДЛЯ ОБРОБКИ ІНФОРМАЦІЇ З МЕТРИК ПРОГРАМНИХ ЗАСТОСУНКІВ”

1. Файл CreateMathModelEntityDto.kt

```
package com.softestimator.service.dto
import com.softestimator.service.models.modelinfo.*
import java.util.*
class CreateMathModelEntityDto(
    val guid: String? = null,
    val name: String,
    val lang: String,
    val category: String,
    val description: String? = null,
    val publishedAt: Date? = null,
    val publishedReference: String? = null,
    val model: ModelConfigJsonDto,
    val accuracy: List<ModelAccuracy>? = null,
    val restrictions: Restrictions? = null,
)
```

2. Файл UpdateMathModelEntityDto.kt

```
package com.softestimator.service.dto
import java.util.*
import com.softestimator.service.models.modelinfo.*
class UpdateMathModelEntityDto(
    val guid: String? = null,
    val name: String? = null,
    val lang: String? = null,
    val category: String? = null,
    val description: String? = null,
    val publishedAt: Date? = null,
    val publishedReference: String? = null,
    val model: ModelConfigJsonDto? = null,
    val accuracy: List<ModelAccuracy>? = null,
    val restrictions: Restrictions? = null,
)
```

3. Файл ModelCofigJsonDto.kt

```
package com.softestimator.service.dto
import com.fasterxml.jackson.annotation.JsonProperty
data class ModelConfigJsonDto(
    @field:JsonProperty("components")
    val components: List<ComponentsItemDto>? = null,
    @field:JsonProperty("interval_config")
    val intervalConfig: IntervalConfigDto? = null,
    @field:JsonProperty("type")
    val type: String? = null,
    @field:JsonProperty("target")
    val target: String? = null,
    @field:JsonProperty("factors")
    val factors: List<String>? = null,
    @field:JsonProperty("factorsCount")
    val factorsCount: Int? = null
)
```

4. Файл IntervalConfigDto.kt

```
package com.softestimator.service.dto
import com.fasterxml.jackson.annotation.JsonProperty
```

```
data class IntervalConfigDto(
    @field:JsonProperty("means")
    val means: List<Double>? = null,
    @field:JsonProperty("sx_inv_matrix")
    val sxInvMatrix: List<Double>? = null,
    @field:JsonProperty("sy")
    val sy: Double? = null,
    @field:JsonProperty("degreesOfFreedom")
    val degreesOfFreedom: Double? = null
)
```

5. Файл EstimationResult.kt

```
package com.softestimator.service.models
data class EstimationResult(
    val factors: Map<String, Double>,
    val result: Double,
    val resultName: String,
    val intervalsTrust: List<IntervalResult> = emptyList(),
    val intervalsPrediction: List<IntervalResult> = emptyList(),
    val message: String? = null,
)
```

6. Файл EstimatingRequestBody.kt

```
package com.softestimator.service.models.api
import com.fasterxml.jackson.annotation.JsonProperty
data class EstimatingRequestBody(
    @field:JsonProperty("factors")
    val factors: Map<String, Double>,
    @field:JsonProperty("interval_prediction")
    val intervalPrediction: List<String>?,
    @field:JsonProperty("interval_trust")
    val intervalTrust: List<String>?,
)
```

7. Файл IntervalResult.kt

```
package com.softestimator.service.models
data class IntervalResult(
    val a: String,
    val lb: Double,
    val ub: Double,
)
```

8. Файл Restrictions.kt

```
package com.softestimator.service.models.modelinfo
data class Restrictions(
    val alphaIntervalTrust: List<Double>?,
    val alphaIntervalPrediction: List<Double>?,
    val factors: List<FactorsItem>?
)
```

9. Файл ModelInfo.kt

```
package com.softestimator.service.models.modelinfo
import java.util.Date
data class ModelInfo(
    val guid: String,
    val name: String,
    val lang: String,
    val description: String?,
    val category: String,
    val factorsTotal: Int,
    val factors: List<String>,
    val factorEstimating: String,
```

```

        val publishedAt: Date?,
        val publishedReference: String?,
        val createdAt: Date,
        val updatedAt: Date,
        val accuracy: List<ModelAccuracy>?,
        val restrictions: Restrictions?,
    )

```

10. Файл FactorsItem.kt

```

package com.softestimator.service.models.modelinfo
data class FactorsItem(
    val min: Double,
    val max: Double,
    val factor: String
)

```

11. Файл ModelAccuracy.kt

```

package com.softestimator.service.models.modelinfo
data class ModelAccuracy(
    val rsq: Double?,
    val mmre: Double?,
    val preds: Map<String, Double?>?,
    val sampleSize: Int?,
    val sample: String?,
)

```

12. Файл DbContracts.kt

```

package com.softestimator.service.db
sealed interface DbContracts {
    val tableName: String
    val columns: List<String>
    data object MathModels : DbContracts {
        override val tableName: String get() = "math_models"
        const val GUID = "guid"
        const val PROGRAMMING_LANGUAGE = "programming_language"
        const val CATEGORY = "category"
        const val NAME = "name"
        const val DESCRIPTION = "description"
        const val FACTORS_QUANTITY = "factors_quantity"
        const val FACTORS_LIST = "factors_list"
        const val FACTOR_ESTIMATING = "factor_estimating"
        const val PUBLISHED_AT = "published_at"
        const val PUBLISHED_REFERENCE = "published_reference"
        const val MODEL_JSON = "model_json"
        const val ACCURACY_JSON = "accuracy_json"
        const val RESTRICTIONS_JSON = "restrictions_json"
        const val CREATED_AT = "created_at"
        const val UPDATED_AT = "updated_at"
        override val columns: List<String> =
            listOf(GUID, PROGRAMMING_LANGUAGE, CATEGORY, NAME, DESCRIPTION, FACTORS_QUANTITY, FACTORS_LIST, FACTOR_ESTIMATING, PUBLISHED_AT, PUBLISHED_REFERENCE, MODEL_JSON, ACCURACY_JSON, RESTRICTIONS_JSON, UPDATED_AT, CREATED_AT,)
    }
    data object ApiKey : DbContracts {
        //NDA
    }
}

```

13. Файл JdbcModelInfoMapper.kt

```

package com.softestimator.service.db.mappers
import com.fasterxml.jackson.core.type.TypeReference
import com.fasterxml.jackson.databind.ObjectMapper

```

```

import com.softestimator.service.db.DbContracts
import com.softestimator.service.models.modelinfo.*
import com.softestimator.service.utils.safe
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.jdbc.core.RowMapper
import org.springframework.stereotype.Component
import java.sql.ResultSet
@Component
class JdbcModelInfoMapper : RowMapper<ModelInfo> {
    @Autowired
    lateinit var objectMapper: ObjectMapper
    override fun mapRow(rs: ResultSet, rowNum: Int): ModelInfo? =
        DbContracts.MathModels.run{
            val guid = rs.getString(GUID)
            val name = rs.getString(NAME)
            val lang = rs.getString(PROGRAMMING_LANGUAGE)
            val description = rs.getString(DESCRIPTION)
            val softwareCategory = rs.getString(CATEGORY)
            val factorsTotal = rs.getInt(FACTORS_QUANTITY)
            val factors = rs.getString(FACTORS_LIST).split(",")
                .takeIf { it.isNotEmpty() } && factorsTotal > 0 && factorsTotal ==
it.size }
            ?: return null
            val factorEstimating = rs.getString(FACTOR_ESTIMATING)
            val publishedAt = rs.getDate(PUBLISHED_AT)
            val publishedReference = rs.getString(PUBLISHED_REFERENCE)
            val createdAt = rs.getDate(CREATED_AT)
            val updatedAt = rs.getDate(UPDATED_AT)
            val accuracyJson = rs.getString(ACCURACY_JSON)
            val restrictionsJson = rs.getString(RESTRICTIONS_JSON)
            val accuracy = if (accuracyJson.isNullOrBlank()) null else safe {
                objectMapper.readValue(accuracyJson, object :
TypeReference<List<ModelAccuracy?>>() {}))
                ?.filterNotNull()
            }
            val restriction = if (restrictionsJson.isNullOrBlank()) null else safe {
                objectMapper.readValue(restrictionsJson, Restrictions::class.java)
            }
            return
ModelInfo(guid,name,lang,description,softwareCategory,factorsTotal,factors,factorE
stimating,publishedAt,publishedReference,createdAt,updatedAt,accuracy,restriction)
        }
}

```

14. Файл DefaultTransformationFactory.kt

```

package com.softestimator.service.regressions.transformations
import com.softestimator.service.regressions.*
import org.oriekhov.da.proc.utils.transformations.*
import org.springframework.stereotype.Component
@Component
class DefaultTransformationFactory : TransformationFactory {
    override fun build(config: TransformationConfig): BiTransformation {
        val param = config.param
        val type = config.type
        return when (type) {
            TransformationType.LOG10 -> Log10Transformation()
            TransformationType.BOX_COX ->
BoxCoxTransformation(param[TransformationType.BoxCoxParams.1]!!)
            TransformationType.YEN_JOHNSON ->
YenJohnsonTransformation(param[TransformationType.YenJohnsonParams.1]!!)
            TransformationType.JOHNSON_SB -> TransformationType.JohnsonParams.run
{

```

```

        JohnsonTransformation.SB(param[n]!!, param[y]!!, param[l]!!,
param[f]!!)
    }
    TransformationType.JOHNSON_SU -> TransformationType.JohnsonParams.run
{
    JohnsonTransformation.SU(param[n]!!, param[y]!!, param[l]!!,
param[f]!!,)
}
    TransformationType.JOHNSON_SL -> TransformationType.JohnsonParams.run
{
    JohnsonTransformation.SL(param[n]!!, param[y]!!, param[l]!!,
param[f]!!,)
}
}
}
}
}
}
}

```

15. Файл TransformationFactory.kt

```

package com.softestimator.service.regressions.transformations
import com.softestimator.service.regressions.TransformationConfig
import org.oriekhov.da.proc.utils.transformations.BiTransformation
interface TransformationFactory {
    fun build(config: TransformationConfig): BiTransformation
}

```

16. Файл DefaultModelConfigMapper.kt

```

package com.softestimator.service.regressions.mapper
import com.softestimator.service.dto.*
import com.softestimator.service.regressions.*
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.stereotype.Component
import kotlin.math.roundToInt
import kotlin.math.sqrt
@Component
class DefaultModelConfigMapper @Autowired constructor() : ModelConfigMapper {
    override fun build(dto: ModelConfigJsonDto): ModelConfig = when (dto.type) {
        ModelConfig.LinearModelConfig.TYPE -> buildLinear(dto)
        ModelConfig.ParametricNonlinearModelConfig.TYPE -> buildNonlinear(dto)
        else -> throw IllegalStateException("Model building is not available for
this model type: ${dto.type}")
    }

    private fun buildNonlinear(dto: ModelConfigJsonDto):
ModelConfig.ParametricNonlinearModelConfig {
        validate(dto)
        val (b0, bVector) = extractLinearModelParams(dto)
        val factors = dto.factors?.filterNotNull().orEmpty()
        val target = requireNotNull(dto.target) { "Dependent variable is missing"
    }

        val components = dto.components?.filterNotNull().orEmpty()
        val yTransformConfig = components.first { it.name == target
    }.toTransformationConfig()
        val xTransformationsConfig = factors.map { f ->
            val config = components.firstOrNull { it.name == f }
            ?.toTransformationConfig()
            requireNotNull(config) { "Transformation config is missing for factor:
'$f'" }
            f to config
        }.toMap()
        return ModelConfig.ParametricNonlinearModelConfig(
            name = dto.prepareModelName(),
            yParam = target,
            xParams = factors,
            intervalConfig = dto.intervalConfig?.toIntervalConfig()
    )
    }
}

```

```

        ?.takeIfValid(dto),
        b0 = b0,
        bVector = bVector,
        yTransformationConfig = yTransformConfig,
        xTransformationsConfig = xTransformationsConfig,
    )
}
private fun ComponentsItemDto.toTransformationConfig(): TransformationConfig {
    val type = requireNotNull(TransformationType.findByNameId(type.orEmpty()))
{
    "Unknown transformation type: '${type}' for component: '$name' with
params '$params'"
    }
    val requiredParams = type.getTransformationParams()
    val transformationParams = requiredParams.map { p ->
        val value = requireNotNull(params?.get(p)?.takeIf { it.isFinite() }) {
            "Transformation param is null or infinite for param '$p' " +
            "of config name='$name' and type='$type'. \n total params:
'$params'"
        }
        p to value
    }.toMap()
    return TransformationConfig(
        type = type,
        param = transformationParams
    )
}

private fun buildLinear(dto: ModelConfigJsonDto):
ModelConfig.LinearModelConfig {
    validate(dto)
    val (b0, bVector) = extractLinearModelParams(dto)
    val factors = dto.factors?.filterNotNull().orEmpty()
    return ModelConfig.LinearModelConfig(
        name = dto.prepareModelName(),
        yParam = dto.target.orEmpty(),
        xParams = factors,
        intervalConfig =
dto.intervalConfig?.toIntervalConfig()?.takeIfValid(dto),
        b0 = b0,
        bVector = bVector
    )
}
private fun ModelConfigJsonDto.prepareModelName() = buildString {
    append(target)
    append("f_${type}")
    append(factors.orEmpty().joinToString(","))
    append("")
}

private fun IntervalConfig.takeIfValid(dto: ModelConfigJsonDto):
IntervalConfig? {
    val factorsCount = dto.factors?.size ?: 0
    return takeIf {
        factorsCount == it.means.size
        && factorsCount == it.sxInv.size
        && factorsCount == it.sxInv[0].size
    }
}
private fun IntervalConfigDto.toIntervalConfig(): IntervalConfig? {
    val cfg = this
    val sxInvFlat = cfg.sxInvMatrix.orEmpty()
    val sxDimension = sqrt(cfg.sxInvMatrix?.size?.toDouble() ?:
0.0).roundToInt()
    val sxDimSquare = Math.pow(sxDimension.toDouble(), 2.0).roundToInt()
    val sxInv = if (sxInvFlat.isNotEmpty() && sxInvFlat.size == sxDimSquare)

```

```

        (0 until sxDimension).map {
            val from = it * sxDimension
            val to = from + sxDimension
            sxInvFlat.subList(from, to)
        }
    } else emptyList()
    val means = cfg.means.orEmpty()
    val degreesOfFreedom = cfg.degreesOfFreedom?.takeIf { it > 0.0 } ?: 0.0
    return if (sxInvFlat.isNotEmpty()
        && sxDimSquare == sxInvFlat.size
        && means.size == sxDimension
        && degreesOfFreedom > 0.0
        && cfg.sy != null
    ) {
        IntervalConfig(
            sxInv = sxInv,
            means = means,
            sy = cfg.sy,
            degreesOfFreedom = degreesOfFreedom
        )
    } else null
}

private fun extractLinearModelParams(dto: ModelConfigJsonDto): Pair<Double,
Map<String, Double>> {
    val linearModelConfig = dto.components.orEmpty().filterNotNull().first {
        it.type == ModelConfig.LinearModelConfig.TYPE
    }
    val b0Key = "b0"
    val b0 = requireNotNull(linearModelConfig.params?.get(b0Key))
    val factors = dto.factors?.filterNotNull().orEmpty()
    val bVector = linearModelConfig.params.orEmpty()
        .filterKeys { factors.contains(it) }
    assert(bVector.keys.size == factors.size)
    return b0 to bVector
}

private fun validate(dto: ModelConfigJsonDto) {
    dto.apply {
        requireNotNull(target) { "Dependent variable is missing" }
        require(!factors.isNullOrEmpty()) { "Model should has factors" }
        factors.forEachIndexed { index, it ->
            requireNotNull(it) { "param at index:'$index' can't be null " }
            require(it.isNotBlank()) { "param at index:'$index' can't be
blank" }
        }
    }
}

```

17. Файл ModelConfigMapper.kt

```

package com.softestimator.service.regressions.mapper
import com.softestimator.service.dto.ModelConfigJsonDto
import com.softestimator.service.regressions.*
interface ModelConfigMapper {
    fun build(dto: ModelConfigJsonDto): ModelConfig
}

```

18. Файл ModelConfig.kt

```

package com.softestimator.service.regressions
sealed interface ModelConfig {
    val name: String
    val yParam: String
    val xParams: List<String>
    val intervalConfig: IntervalConfig?
}

```

```

data class LinearModelConfig(
    override val name: String,
    override val yParam: String,
    override val xParams: List<String>,
    override val intervalConfig: IntervalConfig?,
    val b0: Double,
    val bVector: Map<String, Double>,
) : ModelConfig {
    companion object {
        const val TYPE = "linear"
    }
}

data class ParametricNonlinearModelConfig(
    override val name: String,
    override val yParam: String,
    override val xParams: List<String>,
    override val intervalConfig: IntervalConfig?,
    val yTransformationConfig: TransformationConfig,
    val xTransformationsConfig: Map<String, TransformationConfig>,
    val b0: Double,
    val bVector: Map<String, Double>,
) : ModelConfig {
    companion object {
        const val TYPE = "parametric"
    }
}

data class TransformationConfig(
    val type: TransformationType,
    val param: Map<String, Double>
)

data class IntervalConfig(
    val sxInv: List<List<Double>>,
    val means: List<Double>,
    val sy: Double,
    val degreesOfFreedom: Double,
)

```

19. Файл TransformationType.kt

```

@file:Suppress("ConstPropertyName")
package com.softestimator.service.regressions
enum class TransformationType(
    val nameId: String
) {
    LOG10("log10"),
    BOX_COX("box-cox"),
    YEN_JOHNSON("yen-johnson"),
    JOHNSON_SB("johnson-sb"),
    JOHNSON_SU("johnson-su"),
    JOHNSON_SL("johnson-sl");
    companion object {
        fun findByNameId(nameId: String) = entries.firstOrNull { it.nameId ==
nameId }
    }
    object BoxCoxParams {
        const val l = "l"
    }
    object YenJohnsonParams {
        const val l = "l"
    }
    object JohnsonParams {
        const val n = "n"
        const val y = "y"
    }
}

```

```

        const val l = "l"
        const val f = "f"
    }
}
fun TransformationType.getTransformationParams(): List<String> = when (this) {
    TransformationType.LOG10 -> emptyList()
    TransformationType.BOX_COX -> listOf(TransformationType.BoxCoxParams.l)
                                TransformationType.YEN_JOHNSON ->
listOf(TransformationType.YenJohnsonParams.l)
                                TransformationType.JOHNSON_SB,      TransformationType.JOHNSON_SU,
TransformationType.JOHNSON_SL ->
    TransformationType.JohnsonParams.run { listOf(n, y, l, f) }
}

```

20. Файл DefaultMathModelFactory.kt

```

package com.softestimator.service.regressions.factories
import com.softestimator.service.regressions.ModelConfig
import com.softestimator.service.regressions.toLinearRegression
import com.softestimator.service.regressions.transformations.TransformationFactory
import org.oriekhov.da.proc.utils.models.*
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.stereotype.Component
@Component
class DefaultMathModelFactory @Autowired constructor(
    private val transformationFactory: TransformationFactory
) : MathModelFactory {
    override fun read(config: ModelConfig): MathModel = when (config) {
        is ModelConfig.LinearModelConfig -> LinearRegressionModel(
            linearRegression = config.toLinearRegression()
        )
        is ModelConfig.ParametricNonlinearModelConfig ->
NonlinearParametricRegressionMathModel(
            linearRegression = config.toLinearRegression(),
            yTransform =
transformationFactory.build(config.yTransformationConfig),
            xTransforms = config.xParams.map {
                val tConfig = requireNotNull(config.xTransformationsConfig[it]) {
                    "Transformation config is not found for $it"
                }
                val transformation = transformationFactory.build(tConfig)
                transformation
            }
        )
    }
}

```

21. Файл MathModelFactory.kt

```

package com.softestimator.service.regressions.factories
import com.softestimator.service.regressions.ModelConfig
import org.oriekhov.da.proc.utils.models.MathModel
interface MathModelFactory {
    fun read(config: ModelConfig): MathModel
}

```

22. Файл MathModelIntervalFactory.kt

```

package com.softestimator.service.regressions.factories
import com.softestimator.service.regressions.IntervalConfig
import org.oriekhov.da.proc.utils.*
import org.oriekhov.da.proc.utils.models.*
import org.springframework.stereotype.Component
interface MathModelIntervalFactory {
    fun create(intervalConfig: IntervalConfig, mathModel: MathModel, type:
IntervalType, a: Double): RegressionInterval?
}

```

```

}
@Component
class DefaultMathModelIntervalFactory : MathModelIntervalFactory {
    override fun create(intervalConfig: IntervalConfig, mathModel: MathModel, type:
IntervalType, a: Double): RegressionInterval? {
        val linRegressionInterval = LinearRegressionInterval(
            type = type,
            degreesOfFreedom = intervalConfig.degreesOfFreedom,
            listSXXInv = intervalConfig.sxInv.flatten(),
            xMeans = intervalConfig.means.toDoubleArray(),
            sy = intervalConfig.sy,
            a = a
        )
        return when (mathModel) {
            is LinearRegressionModel -> linRegressionInterval
                                     is NonlinearParametricRegressionMathModel ->
TransformedRegressionInterval(
            interval = linRegressionInterval,
            yBiTransformation = mathModel.yTransform,
            xTransformations = mathModel.xTransforms
        )
            else -> null
        }
    }
}

```

23. Файл ModelConfigExt.kt

```

package com.softestimator.service.regressions
import org.oriekhov.stats.proc.LinearRegression
fun ModelConfig.LinearModelConfig.toLinearRegression(): LinearRegression =
    LinearRegression(b0, xParams.map { bVector[it]!! }.toDoubleArray())
fun ModelConfig.ParametricNonlinearModelConfig.toLinearRegression():
LinearRegression =
    LinearRegression(b0, xParams.map { bVector[it]!! }.toDoubleArray(),)

```

24. Файл MathModelsController.kt

```

package com.softestimator.service.controllers
import com.softestimator.service.AppConstants
import com.softestimator.service.dto.*
import com.softestimator.service.models.EstimationResult
import com.softestimator.service.models.api.EstimatingRequestBody
import com.softestimator.service.models.modelinfo.ModelInfo
import com.softestimator.service.services.modelinfo.ModelInfoService
import com.softestimator.service.usecase.EstimateUseCase
import com.softestimator.service.utils.*
import kotlinx.coroutines.runBlocking
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.dao DuplicateKeyException
import org.springframework.data.crossstore.ChangeSetPersister.NotFoundException
import org.springframework.http.HttpStatus
import org.springframework.web.bind.annotation.*
import org.springframework.web.server.ResponseStatusException
@RestController
class MathModelsController {
    private val logger = createLoggerForClass()
    @Autowired
    lateinit var estimateUseCase: EstimateUseCase
    @Autowired
    lateinit var modelService: ModelInfoService
    @GetMapping("${AppConstants.MAIN_PATH}math-models")
    fun get(
        @RequestParam(name = "category", required = true) category: String,
        @RequestParam("lang") lang: String? = null,
    )

```

```

): List<ModelInfo> = proceedResponse {
    require(category.isCategoryValid()) { "Category param is required. It
can't be empty or contains spaces" }
    modelService.getAllBy(category, lang)
}
@GetMapping("${AppConstants.MAIN_PATH}math-models/{guid}")
fun get(
    @PathVariable("guid") guid: String,
): ModelInfo = proceedResponse {
    modelService.getOne(guid)!!
}
@PostMapping("${AppConstants.MAIN_PATH}math-models")
fun create(
    @RequestBody createRequest: CreateMathModelEntityDto
): ModelInfo = proceedResponse {
    createRequest.run {
        modelService.create(guid, name, lang, category, description,
publishedAt, publishedReference, model, accuracy, restrictions)
    }
}
@PutMapping("${AppConstants.MAIN_PATH}math-models/{guid}")
fun update(
    @PathVariable("guid") guid: String,
    @RequestBody updateRequest: UpdateMathModelEntityDto
) = proceedResponse {
    updateRequest.run {
        modelService.update(guid, name, lang, category, description,
publishedAt, publishedReference, model, accuracy, restrictions)
    }
}.unit()
@DeleteMapping("${AppConstants.MAIN_PATH}math-models/{guid}")
fun delete(
    @PathVariable("guid") guid: String,
) = proceedResponse {
    modelService.delete(guid = guid)
}.unit()
@PutMapping("${AppConstants.MAIN_PATH}math-models/{guid}/estimate")
fun estimate(
    @PathVariable("guid") guid: String,
    @RequestBody body: EstimatingRequestBody,
): EstimationResult = runBlocking {
    proceedResponse {
        estimateUseCase.estimate(
            guid,
            body.factors,
            body.intervalTrust.orEmpty(),
            body.intervalPrediction.orEmpty()
        )
    }
}
}
private inline fun <T> proceedResponse(action: () -> T): T {
    try {
        return action()
    } catch (ex: DuplicateKeyException) {
        logger.warn(ex.message, ex)
        throw ResponseStatusException(HttpStatus.BAD_REQUEST, "", ex)
    } catch (ex: IllegalArgumentException) {
        logger.warn(ex.message, ex)
        throw ResponseStatusException(HttpStatus.BAD_REQUEST, ex.message, ex)
    } catch (ex: NotFoundException) {
        logger.warn(ex.message, ex)
        throw ResponseStatusException(HttpStatus.NOT_FOUND, ex.message, ex)
    } catch (ex: NullPointerException) {
        logger.warn(ex.message, ex)
    }
}

```

```

        throw ResponseStatusException(HttpStatus.NOT_FOUND, ex.message, ex)
    }
}
private fun Any.unit() = Unit
}

```

25. Файл EstimateUseCase.kt

```

package com.softestimator.service.usecase
import com.softestimator.service.models.*
import com.softestimator.service.services.mathmodels.ModelConfigService
import com.softestimator.service.regressions.IntervalConfig
import com.softestimator.service.regressions.factories.*
import com.softestimator.service.regressions.mapper.ModelConfigMapper
import org.oriekhov.da.proc.utils.*
import org.oriekhov.da.proc.utils.models.MathModel
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.stereotype.Component
interface EstimateUseCase {
    suspend fun estimate(
        guid: String,
        params: Map<String, Double>,
        intervalsTrust: List<String> = emptyList(),
        intervalsPrediction: List<String> = emptyList(),
    ): EstimationResult
}
@Component
class EstimateUseCaseImpl : EstimateUseCase {
    @Autowired
    lateinit var service: ModelConfigService
    @Autowired
    lateinit var mapper: ModelConfigMapper
    @Autowired
    lateinit var mathModelFactory: MathModelFactory
    @Autowired
    lateinit var intervalFactory: MathModelIntervalFactory
    override suspend fun estimate(
        guid: String,
        params: Map<String, Double>,
        intervalsTrust: List<String>,
        intervalsPrediction: List<String>
    ): EstimationResult {
        val dtoConfig = service.provide(guid)
        val config = mapper.build(dtoConfig)
        val mathModel = mathModelFactory.read(config)
        val xVector = config.xParams.map {
            requireNotNull(params[it]) { "Param '$it' is not available in input
params: '$params'" }
        }.toDoubleArray()
        val result = mathModel.calculate(xVector)
        val calculatedIntervalsTrust = config
            .takeIf { intervalsTrust.isNotEmpty() }
            ?.intervalConfig
            ?.let { calcIntervals(mathModel, result, xVector, it, intervalsTrust,
IntervalType.Trust) }
        val calculatedIntervalsPredict = config
            .takeIf { intervalsPrediction.isNotEmpty() }
            ?.intervalConfig
            ?.let { calcIntervals(mathModel, result, xVector, it,
intervalsPrediction, IntervalType.Forecast) }
        return EstimationResult(
            factors = params,
            resultName = config.yParam,
            result = result,

```

```

        intervalsTrust = calculatedIntervalsTrust ?: emptyList(),
        intervalsPrediction = calculatedIntervalsPredict ?: emptyList(),
        message = if (intervalsTrust.isNotEmpty() &&
intervalsPrediction.isNotEmpty() && config.intervalConfig == null)
            "interval calculation is not available for this model: $guid"
        else null,
    )
}
private fun calcIntervals(
    mathModel: MathModel,
    y: Double,
    x: DoubleArray,
    intervalConfig: IntervalConfig,
    intervals: List<String>,
    type: IntervalType,
): List<IntervalResult> {
    return intervals.mapNotNull { a ->
        val aVal = a.toDoubleOrNull()
            ?.takeIf { it in 0.0..1.0 }
            ?: return@mapNotNull null
        val interval = intervalFactory.create(
            intervalConfig = intervalConfig,
            mathModel = mathModel,
            type = type,
            a = aVal,
        ) ?: return@mapNotNull null
        val lb = interval.calc(y, x, RegressionBound.LOWER)
        val ub = interval.calc(y, x, RegressionBound.UPPER)
        IntervalResult(
            a = a, lb = lb, ub = ub
        )
    }
}
}

```