

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

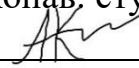
«Допущений до захисту»
Завідувач кафедри

«__» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему: Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, та розробка програми для її реалізації

Виконав: студент 6151м групи


Кольцов А.В.
(підпис, ПІБ)

Керівник роботи:

доцент, к.ф.-м.н., доцент
(посада, науковий ступень вчене звання)


Латанська Л.О.
(підпис, ПІБ)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ проф. С.Б.Приходько
 (підпис)

«10» жовтня 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту Кольцову Андрію В'ячеславовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: «Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, та розробка програми для її реалізації».

Керівник роботи Латанська Л. О.

Затверджені наказом ректора № 798 уч від «28» _____ 09 _____ 2022 р.

2. Термін подання роботи: 14.11.2022 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
 Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.); Огляд літератури та обґрунтування необхідності проведення досліджень за обраною темою; Викладення результатів власних досліджень з висвітленням того нового, що пропонується; Проект програмного забезпечення; Організаційно-економічний розділ; Розділи з охорони праці та охорони навколишнього середовища; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

5. Перелік презентаційних матеріалів 1.Тема 2. Актуальність теми, 3. Мета і завдання дослідження, 4. Об'єкт та предмет дослідження, 5. Методи дослідження, 6. Наукова новизна одержаних результатів, 7. Практичне значення одержаних результатів, 8. Апробація результатів досліджень та публікації, 9. Пошук даних з проектів web-застосунків, що використовують фреймворк Symfony, 10. Перевірка багатовимірних даних на нормальність, 11. Вилучення викидів, 12. Побудова нелінійної регресійної моделі, 13. Побудова лінійної регресійної моделі, 14. Побудова довірчих інтервалів та інтервалів передбачення регресійних моделей, 15. Порівняння якості лінійної та нелінійної регресійних моделей, 16. Порівняння нелінійної регресійної моделі з моделями для фреймворків CakePHP та Yii, 17. Діаграма варіантів використання, 18. Діаграма діяльності, 19. Діаграма класів, 20. Діаграма послідовності, _____

21. Інтерфейс користувача, 22-23. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 10.10.2022 р. _____

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	11.10.2022	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	12.10.2022	НС*
3.	Підготовка розділу (ів) з результатів власних досліджень	13.10.2022	НС*
4.	Підготовка розділу з проекту програмного забезпечення	03.11.2022	
5.	Підготовка організаційно-економічного розділу	06.11.2022	
6.	Підготовка розділу з охорони праці	09.11.2022	
7.	Підготовка розділу з охорони навколишнього середовища	11.11.2022	
8.	Підготовка розділу Висновки	12.11.2022	
9.	Оформлення списку використаних джерел та додатків	13.11.2022	
10.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	14.11.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
11.	Підготовка презентації та доповіді	19.11.2022	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	20.11.2022	
13.	Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	28.11.2022	

* - за результатами наукового стажування (НС), яке було з 01.09.2022 по 09.10.2022 р.

Студент



(підпис)

Кольцов. А. В.

(ПБ)

Керівник роботи



(підпис)

Латанська Л. О.

(ПБ)

РЕФЕРАТ

Кольцов Андрій В'ячеславович

«Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, та розробка програми для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2022 р.

Обсяг роботи: 94 стор., 19 табл., 7 рис., 27 використаних джерел, 5 додатків

Актуальність теми: полягає в необхідності більш точного оцінювання розміру web-застосунків, які створюються з використання фреймворку Symfony через відсутність побудованих моделей для цього фреймворку.

Мета дослідження. Метою роботи підвищення достовірності оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, за допомогою нелінійної регресійної моделі.

Об'єкт дослідження: процес оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Предмет дослідження: нелінійні регресійні моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Методи дослідження: методи математичної статистики та теорії ймовірностей, регресійного аналізу, математичного програмування, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів: удосконалено нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, на основі одновимірною нормалізуючого перетворення десяткового логарифму і викидів регресії. Ця нелінійна регресійна модель в порівнянні з існуючими нелінійними регресійними моделями для PHP фреймворків CakePHP та Yii має більше значення множинного коефіцієнту

детермінації R^2 , менше значення середньої величини відносної похибки $MMRE$ та більше значення відсотку прогнозованих результатів $PRED(0.25)$.

Практичне значення одержаних результатів. Виконано розробку програми, яка реалізує побудовану нелінійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Розроблена програма дозволяє виконати оцінювання розміру таких застосунків на початкових стадіях роботи над проектом.

Апробація результатів роботи. Основні положення й результати кваліфікаційної роботи опубліковані на III Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи» (м. Миколаїв, 26-28 жовтня 2022 року).

Публікації: Основні положення й результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах конференції.

Ключові слова: оцінювання кількості строк коду, нелінійні регресійні моделі, нормалізуючі перетворення, фреймворк Symfony.

ABSTRACT

Koltsov Andrii Vyacheslavovych

«A nonlinear regression model for estimating the size of web applications created using
Symfony framework and developing the software for its implementation»

Qualification work for obtaining a master's degree in specialty 121 – "Software
Engineering". Admiral Makarov National University of Shipbuilding. Mykolaiv, 2022.

Volume: 94 p., 19 tables, 7 figures, 27 references, 5 appendices.

Topic Relevance: is the issue of more precise estimation of the size of web applications developed using the Symfony framework due to the lack of built regression models for this framework.

Research goal. The goal is to increase the confidence of estimation the size of web application developed using the Symfony framework using a nonlinear regression model.

Object of research: the process of estimating the size of web application developed using the Symfony framework.

Subject of research: the nonlinear regression models for estimating the size of web applications developed using the Symfony framework.

Methods of research: methods of mathematical statistics and probability theory, regression analysis, mathematical programming, object-oriented programming.

Scientific contribution: the nonlinear regression model for estimating the size of web applications created using the Symfony framework, based on a one-dimensional normalizing transformation of the decimal logarithm and regression outliers, was improved. This non-linear regression model, compared to the existing non-linear regression models for PHP frameworks CakePHP and Yii, has a higher value of the multiple coefficient of determination R^2 , a lower value of the average value of the relative error $MMRE$ and a higher value of the percentage of predicted results $PRED(0.25)$.

Practical value of obtained results. The program that implements build nonlinear regression model for estimating the size of web application developed using the Symfony frameworks was developed. The program allows to estimate the size of those web application in early project stages.

Approbation of the thesis results. The 3rd All-Ukrainian scientific-practical Internet conference "Information technologies: models, algorithms, systems (ITMAS 2022)" (Mykolaiv, October 26-28, 2022).

Publications. The main provisions and results of the qualification work are published in 1 scientific work - theses of the conference.

Keywords: LOC estimation, non-linear regression models, normalizing transformations, Symfony framework.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ	14
1.1 Аналіз існуючих моделей для оцінювання розміру веб-застосунків	14
1.2 Обґрунтування необхідності проведення досліджень	19
1.3 Висновки до розділу 1	20
2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY.....	22
2.1 Перевірка даних для побудови нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, на наявність мультиколінеарності та викидів	22
2.2 Побудова нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.	29
2.3 Висновки до розділу 2	36
3 РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY	38
3.1 Постановка задачі на розробку проекту програмного забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony	38
3.2 Розробка ескізного проекту програмного забезпечення оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Symfony	39
3.2.1 Побудова моделі варіантів використання програмного застосунку «Symfony LOC estimation»	39
3.2.2 Специфікації варіантів використання програмного застосунку «Symfony LOC estimation».....	41
3.2.3 Моделювання сценарію виконання програмного застосунку «Symfony LOC estimation» за допомогою діаграми діяльності	45
3.3 Розробка технічного проекту програмного забезпечення оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Symfony	47
3.3.1 Статична модель програмного застосунку «Symfony LOC estimation» у вигляді діаграми класів	47
3.3.2 Специфікації класів програмного застосунку «Symfony LOC estimation»	48
3.3.3 Динамічна модель програмного застосунку «Symfony LOC estimation» у вигляді діаграм послідовностей	48
3.4 Розробка робочого проекту програмного забезпечення оцінювання розміру розробки веб-застосунків, що створюються з використанням фреймворку Symfony	50
3.4.1 Кодування програмного застосунку «Symfony LOC estimation»	50

3.4.2	Тестування програмного застосунку «Symfony LOC estimation»	51
3.4.3	Випробування програмного застосунку «Symfony LOC estimation»	55
3.5	Висновки до розділу 3	56
4	РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY.....	58
4.1	Розрахунок витрат на створення й експлуатацію програмного забезпечення	58
4.2	Економічна ефективність розробки та впровадження програмного забезпечення	60
4.3	Висновки до розділу 4	61
5	ОХОРОНА ПРАЦІ	62
5.1	Аналіз шкідливих та небезпечних факторів в офісі фірми	63
5.2	Заходи щодо зменшення впливу шкідливих факторів роботи за персональним комп'ютером.....	66
6	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	68
6.1	Забруднення навколишнього середовища	69
6.2	Розробка заходів щодо зменшення забруднення	71
	ВИСНОВКИ	73
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
	ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	79
	ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ	83
	ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА.....	86
	ДОДАТОК Г – ТЕКСТ ПРОГРАМИ	90
	ДОДАТОК Д – ОПИС ПРОГРАМИ	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПК – Персональний комп'ютер
COCOMO – COConstructive COst Model
NOC – Number Of Classes
DIT – Depth of Inheritance Tree
KLOC – Thousand lines of code
LB – Lower Bound
UP – Upper Bound
LOC – Lines Of Code
MMRE – Mean Magnitude of Relative Error
MRE – Magnitude of Relative Error
PHP – Hypertext Preprocessor
PRED – Percentage of prediction
SMD – the Square Mahalanobis Distance
VIF – Variance Influence Factor

ВСТУП

Актуальність теми. Оцінювання трудомісткості розробки проектів програмного забезпечення є одним з важливих питань у галузі інженерії програмного забезпечення та тісно пов'язано з оцінювання розміру проектів на початкових стадіях розробки. Проекти розробки програмного забезпечення мають досить серйозні проблеми з успішністю: так лише 30-40% проектів завершуються в строк з належними показниками якості та в межах встановленого бюджету [1, 2]. Значна частина проектів програмного забезпечення припадає на web-застосунки. Мова PHP сьогодні застосовується приблизно в 71,4% сайтів і більша частина сайтів використовує саме фреймворки мови PHP, найпопулярнішими з яких є Laravel, Symfony, Codeigniter, Yii, CakePHP [3-5].

В якості фреймворку для проведення дослідження обрано фреймворк Symfony, що за популярністю займає друге місце в опитуваннях сайту sitepoint за 2015 рік та в опитуванні компанії JetBrains за 2021 [4, 5]. Іншою причиною такого вибору є відсутність моделей для оцінювання розміру web-застосунків, які б були побудовані саме для цього фреймворку.

В якості підходів до прогнозування розміру програмних проектів використовуються методи, основані на порівнянні з аналогами, експертне оцінювання, алгоритмічні методи та математико-статичні методи та моделі.

Для оцінювання розміру у строках коду інформаційних систем з відкритим кодом на PHP можна використати лінійні регресійні рівняння [6, 7], або нелінійні регресійні моделі [8-13]. Моделі [6-11] для побудови вимагають дані діаграми класів: загальну кількість класів, загальну кількість зв'язків та середню кількість атрибутів на клас, а моделі [12-13] в якості факторів використовують такі метрики як: кількість класів; сума середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти; середня кількість методів. Для фреймворку Laravel побудовано нелінійну регресійну модель з використанням в якості предикторів метрик як в [12-13]. Подібні моделі

побудовано й для фреймворків CakePHP, Yii з дещо відмінним набором метрик [15, 16].

Питання щодо можливості використання лінійної регресійної моделі для оцінювання часто суперечить одному з припущень про можливість використання такої моделі – виконанню умови нормальності розподілу помилок, що в свою чергу призводить до необхідності використання нелінійних регресійних моделей. Вищенаведене ставить питання про необхідність побудови нелінійної моделі для оцінювання кількості строк коду (у тисячах) web-застосунків, що створюються з використанням фреймворку Symfony.

Сталою практикою для нормалізації даних для побудови моделей є використання десяткового логарифму, як це зроблено для моделей COCOMO, ISBSG, COCOMO II, тому для побудови моделі для оцінювання розміру веб-застосунків, розроблених з використанням фреймворку Symfony, використовуватимемо нормалізуюче перетворення десяткового логарифму.

Мета та завдання дослідження. Метою роботи є підвищення достовірності оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, за допомогою нелінійної регресійної моделі.

Досягнення мети передбачає виконання наступних завдань:

- Провести дослідження існуючих моделей для оцінювання розміру web-застосунків.
- Виконати збір даних, для яких буде побудовано нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.
- Виконати удосконалення нелінійної регресійної моделі для web-застосунків, що створюються з використанням фреймворку Symfony, на основі нормалізуючого перетворення.
- Розробити проект програмного забезпечення для реалізації розрахунків за побудованою моделлю для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Об'єкт дослідження: процес оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Предмет дослідження: нелінійні регресійні моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Методи дослідження: методи математичної статистики та теорії ймовірностей, регресійного аналізу, математичного програмування, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів: удосконалено нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, на основі одновимірного нормалізуючого перетворення десяткового логарифму і викидів регресії. Ця нелінійна регресійна модель в порівнянні з існуючими нелінійними регресійними моделями для PHP фреймворків CakePHP та Yii має більше значення множинного коефіцієнту детермінації R^2 , менше значення середньої величини відносної похибки *MMRE* та більше значення відсотку прогнозованих результатів *PRED*(0.25).

Практичне значення одержаних результатів. Виконано розробку програми, яка реалізує побудовану нелінійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Розроблена програма дозволяє виконати оцінювання розміру таких застосунків на початкових стадіях роботи над проектом.

Особистий внесок здобувача. Кваліфікаційна робота є самостійно виконаною науковою працею. Усі наукові результати отримано автором особисто. У праці [17], яка була опублікована у співавторстві, автору належить нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, на основі нормалізуючого перетворення десяткового логарифму.

Апробація результатів роботи. Основні положення й результати кваліфікаційної роботи опубліковані на III Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи» (м. Миколаїв, 26-28 жовтня 2022 року).

Публікації: Основні положення й результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах конференції.

Ключові слова: оцінювання кількості строк коду, нелінійні регресійні моделі, нормалізуючі перетворення, фреймворк Symfony.

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

1.1 Аналіз існуючих моделей для оцінювання розміру веб-застосунків

Оцінювання трудомісткості розробки проектів програмного забезпечення є важливою проблемою у галузі інженерії програмного забезпечення. Існує багато підходів до проведення такого оцінювання, зокрема часто застосовуються методи лінійного та нелінійного регресійного аналізу з відповідною побудовою регресійних моделей та рівнянь. Такі відомі моделі як COCOMO, COCOMO II, ISBSG дозволяють виконувати оцінювання вартості, тривалості та трудомісткості проектів програмного забезпечення та будуються на основі розміру програмного забезпечення (у строках коду), як одного з основних факторів, який на початкових стадіях роботи над проектом програмного забезпечення невідомий. Також ці моделі не враховують особливості мов програмування.

В роботах [6, 7] Hee Beng Kuan Tan, Yuan Zhao та Hongyu Zhang створили багатофакторні лінійні регресійні моделі для оцінювання розміру інформаційних систем, в тому числі інформаційних систем з відкритим кодом на мові РНР. В якості залежної величини розглядали кількість строк коду, а в якості факторів для побудови моделей розглядали метрики концептуальної моделі даних – кількість класів, кількість відносин між класами у концептуальній моделі, середня кількість атрибутів. Втім, застосування лінійного регресійного аналізу вимагає виконання певних умов: залишки або залежна змінна мають бути нормально розподіленими, що на практиці зустрічається рідко. Подібні складнощі з використанням лінійного регресійного аналізу призводять до необхідності використання методів нелінійного регресійного аналізу.

На практиці отримання якісних нелінійних регресійних моделей та рівнянь, визначення довірчих інтервалів та інтервалів прогнозування нелінійної регресії

можливе завдяки використанню нормалізуючих перетворень. Для інформаційних систем, що розробляються з використанням мови PHP, можна умовно виділити дві групи нелінійних регресійних моделей за метриками, що використовувались для побудови цих моделей. В роботах [8-11] будувались нелінійні регресійні рівняння з використанням таких же метрик діаграми класів, що і в роботах [6, 7]. В роботах [12, 13] використовували дещо інший набір метрик: кількість класів; сума середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти; середня кількість методів. Для нормалізації даних у вищенаведених роботах використовували різні набори нормалізуючих перетворень: десятковий логарифм, одновимірне та багатовимірне нормалізуюче перетворення Бокса-Кокса, одновимірне та багатовимірне нормалізуюче перетворення сімейства Джонсона S_b . Отримані у роботах оцінки дозволяють зробити висновок про більшу якість отриманих моделей при використанні саме багатовимірних нормалізуючих перетворень, зокрема перетворення сімейства Джонсона S_b , а також про менші ширини інтервалів прогнозування нелінійної регресії.

У роботах [15, 16] розглядали популярні фреймворки мови PHP – CakePHP та Yii. Автори висунули припущення про те, що розмір web-застосунків, що створюються з використанням фреймворків, може залежати і від інших метрик, аніж метрики у роботах [6-14], тому були використані такі метрики: кількість класів, середня кількість методів на кожен клас, глибина дерева наслідування. В якості нормалізуючого перетворення в цих роботах використовували десятковий логарифм, одновимірне та багатовимірне перетворення Бокса-Кокса. Найкращі оцінки нелінійної регресійної моделі отримано для багатовимірного перетворення Бокса-Кокса, як і менші ширини інтервалів прогнозування нелінійної регресії.

Відома також нелінійна регресійна модель для фреймворку Laravel [14], побудована з використанням таких метрик, як і у [12, 13]. Найкращі оцінки R^2 , $MMRE$ і $PRED(0,25)$ цієї моделі отримано для багатовимірного нормалізуючого перетворення Джонсона сімейства S_b . Таблицю з порівнянням оцінок для побудованих нелінійних регресійних моделей для PHP фреймворків наведено нижче.

Таблиця 1.1 – Порівняння оцінок нелінійних регресійних моделей для РНР фреймворків

Оцінка	Багатовимірне нормалізуюче перетворення		
	Вох-Сох		Johnson S_b
	CakePHP	Yii	Laravel
R^2	0.9963	0.9249	0.993
MMRE	0.0425	0.1439	0.107
PRED(0.25)	1.0	0.8684	0.929

Порівняння оцінок, отриманих для фреймворків, може слугувати підтвердженням доцільності використання метрик, як у [15, 16] для побудови нелінійних регресійних моделей.

Сталою практикою для нормалізації даних для побудови моделей є використання десяткового логарифму, як це зроблено для моделей COCOMO, ISBSG, COCOMO II, тому для побудови моделі для оцінювання розміру web-застосунків, розроблених з використанням фреймворку Symfony, скористаємось нормалізуючим перетворенням десяткового логарифму. В свою чергу в якості метрик для побудови моделей будемо використовувати кількість класів, середню кількість методів на кожен клас, глибину дерева наслідування.

У випадку нормальності початкових даних в якості регресійної моделі можна взяти лінійну регресійну модель з наступним загальним виглядом:

$$Y = \hat{Y} + \varepsilon = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon, \quad (1.1)$$

де $\hat{Y}$ – результат передбачення за рівнянням лінійної регресії, b_i ($i = \overline{0, k}$) – параметри лінійного рівняння регресії, X_i ($i = \overline{1, k}$) – значення незалежної змінної, ε – гаусівська випадкова величина, k – кількість факторів у моделі.

При підпорядкуванні початкових даних нормальному розподілу визначити довірчий інтервал лінійної регресії для k незалежних змінних можна таким чином

$$\hat{Y}_i \pm t_{\alpha/2, v} S_Y \left\{ \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.2)$$

а інтервал передбачення лінійної регресії як

$$\hat{Y}_i \pm t_{\alpha/2, v} S_Y \left\{ 1 + \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.3)$$

де $S_Y^2 = \frac{1}{v} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$; $v = N - k - 1$; $(\mathbf{X}^+)^T \mathbf{X}^+ - k \times k$ матриця

$$(\mathbf{X}^+)^T \mathbf{X}^+ = \begin{pmatrix} S_{X_1 X_1} & S_{X_1 X_2} & \dots & S_{X_1 X_k} \\ S_{X_1 X_2} & S_{X_2 X_2} & \dots & S_{X_2 X_k} \\ \dots & \dots & \dots & \dots \\ S_{X_1 X_k} & S_{X_2 X_k} & \dots & S_{X_k X_k} \end{pmatrix}, \quad (1.4)$$

де $S_{X_q X_r} = \sum_{i=1}^N [X_{qi} - \bar{X}_q][X_{ri} - \bar{X}_r]$, $q, r = \overline{1, k}$.

У випадку, коли початкові дані є негаусівськими, виникає потреба будувати нелінійну регресійну модель. Для побудови нелінійної моделі можна використовувати метод перебору, методи лінеаризуючих перетворень, нормалізуючі перетворення. Використання методу перебору вимагає значних часових витрат на підбір моделі та не надає гарантій того, що вона буде найкращою. Складнощі з використанням лінеаризуючих перетворень полягають у пошуці цих перетворень. Саме тому доцільніше всього використовувати нормалізуючі перетворення, які вдається підібрати майже завжди.

Суть методу з використанням нормалізуючих перетворень наступна. Виконується нормалізація емпіричних даних, проводиться вилучення викидів. За нормалізованими даними без викидів будується лінійна регресійна модель. Останнім кроком відбувається перехід від лінійної регресійної моделі до нелінійної за зворотнім нормалізуючим перетворенням [18].

Нормалізуюче перетворення негаусівського випадкового вектору $P = \{Y, X_1, X_2, \dots, X_k\}^T$ у гаусівський випадковий вектор $T = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$, задається як

$$T = \psi(P), \quad (1.5)$$

а зворотнє перетворення до (1.5) як

$$P = \psi^{-1}(T), \quad (1.6)$$

тут ψ – вектор, $\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$.

Таким чином нелінійна регресійна модель, що побудована з використанням нормалізуючого перетворення [19], має вигляд

$$Y = \psi_Y^{-1}(\hat{Z}_Y + \varepsilon) = \psi_Y^{-1}(b_0 + b_1 Z_1 + b_2 Z_2 + \dots + b_k Z_k + \varepsilon), \quad (1.7)$$

де ψ_Y – перша компонента вектору ψ перетворення (1.5).

В свою чергу довірчий інтервал нелінійної регресії для k незалежних змінних можна знайти таким чином

$$\psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right), \quad (1.8)$$

а інтервал передбачення нелінійної регресії як

$$\psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.9)$$

1.2 Обґрунтування необхідності проведення досліджень

Згідно даних сайту w3techs мова PHP використовується в приблизно 71,4% усіх сайтів, а згідно опитування компанії JetBrains, проведеного у 2021 році, лише 7% опитаних не використовуються жодний з PHP фреймворків [3, 4]. Використання фреймворків мови PHP надає можливість пришвидшити процес розробки за рахунок наявності готових рішень, великих можливостей для масштабування майбутніх застосунків, а також використання провідних практик. За даними опитування, проведеного сайтом sitepoint в 2015 році, до популярних фреймворків можна віднести: Laravel, Symfony, Codeigniter, Yii, CakePHP; опитування, проведене компанією JetBrains в 2021 році, підтверджує дані, отримані в 2015 році [4, 5].

Для оцінювання розміру інформаційних систем з відкритим кодом на PHP існують багатофакторні лінійні регресійні рівняння [6, 7], побудовані з використанням таких метрик, як кількість класів, кількість відносин між класами у концептуальній моделі, середня кількість атрибутів. З цими ж метриками було побудовано нелінійні регресійні моделі, що в якості нормалізуючого перетворення використовували багатовимірне перетворення Джонсона сімейства S_b [8-11]. Аналогічно до цих моделей в роботах [12, 13] будувались нелінійні регресійні моделі з відмінністю лише у використаних метриках, це були кількість класів; сума середньої кількості класів, на які впливає даний клас, і середньої кількості класів, з яких даний клас отримує ефекти; середня кількість методів.

З огляду на можливу залежність якості нелінійних регресійних моделей для оцінювання розміру для проектів web-застосунків, що використовують фреймворки мови PHP, від інших метрик, у [15, 16] автори запропонували використати кількість класів, середню кількість методів у класі, глибину дерева наслідування як предиктори, та в якості нормалізуючого перетворення використали перетворення Бокса-Кокса. Порівняння результатів побудови цих моделей з моделлю для фреймворку Laravel з набором метрик як у [12, 13] показало прийнятність такого підходу до побудови нелінійних регресійних моделей.

В якості фреймворку для проведення дослідження обрано фреймворк Symfony як один з найбільш затребуваних [4, 5]. Іншою причиною такого вибору є відсутність моделей, які б були побудовані саме для цього фреймворку.

Для побудови моделі для оцінювання розміру веб-застосунків, розроблених з використанням фреймворку Symfony, скористаємось нормалізуючим перетворення десятичного логарифму.

Зважаючи на все вищевкладене, бачимо доцільність проведення дослідження за обраною темою.

1.3 Висновки до розділу 1

В даному розділі було виконано огляд існуючих методів та моделей для виконання оцінювання розміру web-застосунків, що розробляються мовою PHP чи з використанням фреймворків мови PHP, проведено обґрунтування доцільності виконання дослідження та побудови нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Symfony.

Також в розділі:

- Визначено, що для оцінювання розміру програмних застосунків, що розробляються мовою PHP, використовуються як лінійні, так і нелінійні регресійні моделі.

- Визначено, що починають з'являтися більш точні нелінійні регресійні моделі, що спрямовані саме на оцінювання веб-застосунків, що створюються з використанням фреймворків мови PHP.

- Визначено, що існуючі нелінійні регресійні моделі для інформаційних систем на PHP потенційно можуть погано оцінювати розмір веб-застосунків, що створюються з використанням фреймворків мови PHP.

- Визначено залежність якості побудованих нелінійних регресійних моделей, рівнянь, довірчих інтервалів та інтервалів прогнозування від виду нормалізуючого перетворення.

– Визначено, що застосування багатовимірних нормалізуючих перетворень дає кращі результати, аніж застосування одновимірних перетворень.

Саме тому доцільним є проведення досліджень на обрану тему.

2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY

2.1 Перевірка даних для побудови нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, на наявність мультиколінеарності та викидів

Для побудови нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, було зібрано дані з метрик програмного забезпечення 50 web-застосунків, які було розроблено з використанням фреймворку Symfony, та які було розміщено на сайті GitHub (<https://github.com/>). За допомогою інструменту PhpMetrics було отримано наступні метрики діаграми класів: кількість строк коду KLOC (Y), кількість класів NOC (X_1), середня кількість методів на кожен клас Methods by Class (X_2), глибина дерева наслідування DIT (X_3). Отримані метрики наведено у таблиці 2.1.

Таблиця 2.1 – Початковий набір даних разом з розрахованою відстанню Махаланобіса (SMD)

№	KLOC (Y)	NOC (X_1)	Methods by Class (X_2)	DIT (X_3)	SMD
1	2	3	4	5	6
1	0,487	7	4,73	1,67	3,4127
2	0,512	8	3,50	1,38	1,5902
3	0,305	8	14,00	1,30	39,0471
4	0,408	8	3,50	1,38	1,5688
5	0,807	9	4,00	1,50	1,6274
6	0,506	11	2,27	1,27	2,7631
7	0,514	13	2,00	1,36	2,1645
8	0,812	14	4,86	1,47	1,3247
9	0,716	14	2,00	1,14	4,4710
10	1,037	15	4,27	1,33	1,0237
11	0,703	15	2,47	1,64	2,3041

Продовження табл. 2.1

1	2	3	4	5	6
12	1,152	17	2,94	1,44	0,9983
13	0,411	18	1,17	1,45	2,7239
14	1,630	18	4,28	1,41	1,1342
15	0,823	19	3,63	1,41	0,6762
16	1,813	21	3,43	1,33	1,4557
17	2,427	22	5,45	1,38	2,7999
18	0,837	23	2,87	1,48	0,9228
19	1,004	23	1,22	1,40	2,2219
20	0,990	23	3,30	1,74	3,1699
21	0,813	23	2,65	1,30	1,4578
22	2,193	24	6,21	1,38	2,2929
23	0,740	25	2,52	1,56	1,7723
24	0,799	25	2,16	1,21	2,7983
25	2,165	26	4,73	1,25	1,4198
26	1,056	26	3,08	1,22	1,7512
27	2,191	27	3,26	1,20	2,3205
28	0,974	27	1,52	2,21	17,1271
29	3,100	31	4,52	1,35	3,1277
30	2,289	32	3,88	1,75	3,5281
31	2,500	35	4,97	1,41	0,6138
32	2,387	36	4,31	0,95	5,2766
33	1,844	41	3,56	1,46	1,1383
34	1,803	41	3,51	1,63	2,5222
35	2,404	43	3,14	1,54	0,8127
36	3,046	44	3,98	1,67	2,7318
37	2,702	45	3,84	1,28	0,6274
38	5,571	46	6,22	1,23	16,0593
39	2,729	46	3,80	1,35	0,4194
40	2,190	46	2,87	1,23	1,8387
41	3,669	49	4,08	1,23	2,0866
42	2,436	50	3,20	1,35	1,3665
43	1,682	52	2,10	1,32	5,2234
44	4,107	61	3,18	1,35	2,2335
45	3,392	64	2,52	1,40	2,5943
46	3,034	67	3,24	1,31	4,9717
47	5,730	71	4,03	1,73	11,5027
48	4,047	83	2,77	1,24	7,8523
49	5,230	88	3,14	1,26	6,8327
50	5,993	91	4,86	1,32	8,3003

В таблиці 2.1 наведено результати розрахунку квадрату відстані Махаланобіса (SMD). Значення SMD для кожної з точок багатовимірних даних можна визначити як

$$d_i^2 = (X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}), \quad (2.1)$$

де X_i – i -та точка багатовимірних даних негаусівського вектору X ; $\bar{X}$ – вектор вибірових середніх; S_N – вибіркова коваріаційна матриця. В нашому випадку розглядається 4 компонентний вектор, N – кількість елементів у вибірці.

$$S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X}) (X_i - \bar{X})^T. \quad (2.2)$$

Підтвердженням негаусівського багатовимірного розподілу метрик є отримані значення SMD для проектів 3, 28, 38, 47, які дорівнюють 39,0471, 17,1271, 16,0593 та 11,5027 відповідно, що більше критичного значення квантіля розподілу χ^2 для рівня значимості 0,05, яке в свою чергу дорівнює 9,488.

Негаусівський розподіл багатовимірних даних також підтверджує оцінка багатовимірного ексцесу β_2 [20], яку можна визначити наступним чином

$$\hat{\beta}_2 = \frac{1}{N} \sum_{i=1}^N \{(X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X})\}^2. \quad (2.3)$$

Якщо дані підпорядковуються багатовимірному нормальному розподілу, то $\hat{\beta}_2 = 24$, що можна легко визначити як $\hat{\beta}_2 = m(m + 2)$ ($m = 4$ для трьохфакторної моделі). Для обраних метрик оцінка $\hat{\beta}_2$ дорівнює 53,001.

Побудова багатофакторної регресійної моделі пов'язана з перевіркою майбутніх факторів (предикторів) на предмет наявності мультиколінеарності.

Мультиколінеарність – негативне явище множинного регресійного аналізу, що унеможливорює виконання оцінювання окремого впливу кожного фактору на

залежну змінну. Тобто, при наявності мультиколінеарності модель має лінійну, або близьку до неї, залежність між факторами (незалежними змінними).

Для того, щоб впевнитись у відсутності мультиколінеарності майбутніх факторів, визначимо коефіцієнти впливу дисперсії (Variance Inflation Factors, VIFs). В множинній лінійній регресійній моделі з k -предикторами X_i , ($i = \overline{1, k}$), VIFs – це елементи, що знаходяться на головній діагоналі в оберненій коваріаційній матриці розміру $k \times k$ k -предикторів. Зазвичай значення VIFs, які більше за 10 вважають сигналом проблем з мультиколінеарністю. Значення VIFs від 1 до 5 свідчать про відсутність мультиколінеарності [21].

Для обраних предикторів NOC, Methods by Class, DIT, наведених у таблиці 2.1, отримано значення VIFs 1,027, 1,030 та 1,036 відповідно. Всі значення менші за 5, що показує відсутність мультиколінеарності в даних.

Розрахунок значень квадрату відстані Махаланобіса SMD та багатовимірною ексцесу β_2 показали, що багатовимірні дані є негаусівськими, що в свою чергу вимагає для вилучення викидів застосування методу, що базується на нормалізуючих перетвореннях та квадраті відстані Махаланобіса SMD [22, 23]. Для окремо взятої точки нормалізованих даних i ($i = \overline{1, N}$) застосовується наступна формула

$$d_i^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (2.4)$$

де Z_i – i -та точка багатовимірних даних нормалізованого вектору Z ; $\bar{Z}$ – вектор вибірових середніх; S_N – вибіркова коваріаційна матриця для нормалізованих даних. В нашому випадку розглядається 4 компонентний вектор.

$$S_N = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z}) (Z_i - \bar{Z})^T. \quad (2.5)$$

Для отримання вектору Z необхідно скористатись (1.5) та нормалізуючим перетворенням десяткового логарифму. Тоді перетворення (1.5) має наступний вигляд $\psi = \{\lg Y, \lg X_1, \lg X_2, \lg X_3\}^T$.

Для багатовимірних гаусівських даних розподіл величин SMD подібний до розподілу χ^2 , тобто для великих вибірок $N - m$ (принаймні більше 25) відстань d_i^2 має значення, що є близькими до незалежної випадкової величини $\chi_{m,\alpha}^2$, де $\chi_{m,\alpha}^2$ – квантіль розподілу χ^2 з рівнем значимості α [24].

Також для багатовимірних гаусівських даних значення статистики T_{S_i} поводитьсь подібно F -розподілу з відповідним значенням квантілю $F_{m,N-m,\alpha}$

$$T_{S_i} = \frac{N(N-m)d_i^2}{m(N^2-1)}. \quad (2.6)$$

Багатовимірним викидом вважається рядок або зі значенням SMD, яке перевищує значення квантілю розподілу χ^2 , або розрахована статистика T_{S_i} , більша за значення $F_{m,N-m,\alpha}$. Ітераційний процес проводиться до повного усунення викидів.

Для вилучення викидів з початкового набору даних на рівні значимості 0,05 знадобилось 4 ітерації. На кожній з ітерацій було розраховано квадрат відстані Махаланобіса SMD та значення статистики T_{S_i} , які вказали на одні й ті ж самі набори даних, які слід вважати викидами. Отримані значення квадрату відстані Махаланобіса на кожній з чотирьох ітерацій наведено в таблиці 2.2. Нагадаємо, що критичне значення квантіля розподілу χ^2 для рівня значимості 0,05 дорівнює 9,488.

Таблиця 2.2 – Процес вилучення викидів

№	SMD_1	SMD_2	SMD_3	SMD_4	№	SMD_1	SMD_2	SMD_3	SMD_4
1	2	3	4	5	6	7	8	9	10
1	3,4127	7,7323	7,6980	7,5313	26	1,7512	3,1670	3,9045	3,8043
2	1,5902	3,9358	3,7972	3,7293	27	2,3205	3,6090	4,2705	4,1800
3	39,0471	-	-	-	28	17,1271	-	-	-
4	1,5688	4,6726	4,6527	4,5439	29	3,1277	2,7795	2,7501	2,7670

Продовження табл. 2.2

1	2	3	4	5	6	7	8	9	10
5	1,6274	5,3960	5,4178	5,2934	30	3,5281	5,0408	5,8593	5,7731
6	2,7631	3,7561	4,2149	4,8502	31	0,6138	1,5906	1,8991	2,1584
7	2,1645	3,2426	3,8876	5,1421	32	5,2766	11,2752	-	-
8	1,3247	4,4422	5,1292	5,2234	33	1,1383	2,1097	2,1471	2,0794
9	4,4710	6,9518	8,4887	9,2917	34	2,5222	3,8844	3,9513	3,8824
10	1,0237	1,8875	2,1139	2,1298	35	0,8127	1,7202	1,9937	2,1111
11	2,3041	3,4962	4,6029	5,5254	36	2,7318	3,9461	4,3002	4,1960
12	0,9983	2,1595	2,8266	3,0906	37	0,6274	1,0597	1,3664	1,3661
13	2,7239	9,4773	11,7517	-	38	16,0593	5,6217	6,1639	7,1594
14	1,1342	2,3609	2,2981	2,2460	39	0,4194	0,7646	0,8244	0,8035
15	0,6762	2,9647	3,3020	3,2094	40	1,8387	1,6553	1,9579	1,9597
16	1,4557	3,1057	3,5674	3,5174	41	2,0866	1,9524	2,3086	2,3541
17	2,7999	3,6073	3,4898	3,7843	42	1,3665	1,1032	1,0912	1,0703
18	0,9228	2,6103	2,4958	2,6340	43	5,2234	3,6635	3,7137	4,3851
19	2,2219	14,1708	-	-	44	2,2335	2,4786	2,7399	2,7266
20	3,1699	4,8409	5,1762	5,2370	45	2,5943	3,2756	3,9830	4,3936
21	1,4578	2,7832	2,9120	3,0255	46	4,9717	2,6756	2,7654	2,6860
22	2,2929	3,4776	4,2118	5,0300	47	11,5027	8,4630	9,3651	9,1615
23	1,7723	5,0426	4,8893	5,3047	48	7,8523	3,6562	3,8374	3,8321
24	2,7983	4,1766	4,4902	4,9204	49	6,8327	3,6542	3,7842	3,6980
25	1,4198	2,0179	2,5287	2,7668	50	8,3003	4,5457	5,0805	5,4259

Загалом з початкового набору даних 50 проектів web-застосунків, що створюються з використанням фреймворку Symfony, було вилучено 5 проектів: проект 3 та 28 на першій ітерації, проект 19 та 32 на другій, проект 13 на третій ітерації. Для цих проектів розраховане значення SMD та статистики T_{S_i} перевищували відповідні критичні значення квантілю розподілу χ^2 та значення квантілю F -розподілу для рівня значимості 0,05.

Для подальшої побудови нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, буде використано нормалізований набір даних з 45 проектів без викидів.

Одним з критеріїв, що слугує для теоретичного обґрунтування використання лінійних регресійних моделей, є нормальність розподілу залишків ε . Зазвичай для перевірки нульової гіпотези щодо нормальності розподілу випадкової величини

використовують критерій Пірсона, який доцільно застосовувати для великих вибірок ($n > 30$) [25].

Значення критерія Пірсона χ^2 розраховують як

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j}, \quad (2.7)$$

де m – кількість підінтервалів (часток, кліток), на які розбивається інтервал $[x_{min}, x_{max}]$; x_{min}, x_{max} – відповідно мінімальне і максимальне значення випадкової величини X ; n_j – абсолютна частота в j -му підінтервалі (кількість значень випадкової величини, які попадають у j -ий підінтервал); p_j – ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал.

Існує декілька підходів для визначення кількості підінтервалів, на які можна розбити підінтервал $[x_{min}, x_{max}]$:

- Формула Старджеса: $m = \log_2 n + 1 = 3,3 \lg n + 1$.
- Формула Брукса і Карузера: $m = 5 \lg n$.

Ймовірність того, що значення випадкової величини X попадають у j -ий підінтервал може бути визначена як

$$p_j = \int_{x_{j-1}}^{x_j} f(x) dx, \quad (2.8)$$

де x_{j-1} та x_j – ліва і права границі j -го підінтервалу, $f(x)$ – функція щільності ймовірності.

Функція щільності ймовірності нормального розподілу має вигляд

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} * e^{-\frac{(x-m_x)^2}{2\sigma_x^2}}, \quad (2.9)$$

де m_x – математичне сподівання випадкової величини, σ_x – середьоквадратичне відхилення випадкової величини.

В свою чергу кількість ступенів вільності ν можна визначити як $\nu = m - k - 1$, де k – кількість параметрів, від яких залежить закон розподілу. Для нормального розподілу $k = 2$.

Після цього в залежності від рівня значимості α та кількості ступенів вільності ν за таблицю верхніх 100α %-вих точок розподілу χ^2 знаходять значення $\chi_{кр}^2$. Якщо $\chi^2 \leq \chi_{кр}^2$, то з ймовірністю $1 - \alpha$ можна прийняти гіпотезу про те, що закон розподілу результатів спостережень є нормальним, якщо $\chi^2 > \chi_{кр}^2$ – цю гіпотезу потрібно відкинути.

2.2 Побудова нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Для побудови нелінійної регресійної моделі для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, використаємо дані без викидів. Для отриманих нормалізованих даних без викидів побудуємо лінійну багатofакторну регресійна модель, яка має наступний вигляд

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \quad (2.10)$$

де $Z_Y = \lg Y$; $Z_i = \lg X_i, i = \overline{1, 3}$.

Оцінки параметрів побудованої моделі $\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3$ дорівнюють -1,6378, 0,9664, 0,9630 та -0,4627 відповідно.

Виконаємо перевірку нульової гіпотези щодо нормальності розподілу залишків ε лінійної регресії для нормалізованих даних. Гаусівський розподіл похибок може слугувати теоретичним обґрунтування можливості застосування лінійних регресійних моделей.

Для 45 проектів web-застосунків, що створюються з використанням фреймворку Symfony, отримане значення критерія Пірсона χ^2 для лінійної регресії для нормалізованих даних, визначене за формулою (2.7), дорівнює 6,934. В свою чергу для 3 ступенів вільності та на рівні значимості 0,05 критичне значення χ^2 дорівнює 7,814. Так як $\chi^2 < \chi_{кр}^2$, ми приймаємо нульову гіпотезу щодо нормальності розподілу залишків ε на рівні значимості 0,05.

Значення оцінки математичного сподівання залишків ε лінійної регресії для нормалізованих даних дорівнює $-3,02e^{-14}$, а значення оцінки середньоквадратичного дорівнює 0,091.

За лінійним регресійним рівнянням (2.10) та взаємо-зворотнім нормалізуючим перетворенням отримуємо нелінійну регресійну модель:

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}. \quad (2.11)$$

Виконавши підстановку визначених параметрів в (2.11) отримуємо нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Отриману модель було перевірено за допомогою оцінки множинного коефіцієнту детермінації R^2 , середньої величини відносної похибки $MMRE$ та відсотку прогнозованих результатів $PRED(0,25)$, що дорівнюють 0,900, 0,187 та 0,756 відповідно, та свідчать про задовільну якість побудованої моделі. Нагадаємо, що допустимі значення $MMRE$ і $PRED(0,25)$ складають не більше 0,25 і не менше 0,75 відповідно, а значення R^2 більше 0,75 [26, 27].

Для визначення довірчих інтервалів та інтервалів передбачення нелінійної регресії застосована у формулах (1.8) та (1.9) матриця S_Z має вигляд

$$S_Z = \begin{pmatrix} 3,775 & 0,067 & -0,084 \\ 0,067 & 0,674 & 0,044 \\ -0,084 & 0,044 & 0,099 \end{pmatrix}$$

За формулою (1.9) було визначено інтервали передбачення нелінійної регресії. Усі дані знаходяться в межах інтервалів передбачення нелінійної регресії.

Для проведення порівняння побудуємо багатофакторну лінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Модель має наступний вигляд:

$$Y = \hat{Y} + \varepsilon = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon. \quad (2.12)$$

Для побудованої моделі отримано наступні оцінки параметрів: $\hat{b}_0 = -1,3253$, $\hat{b}_1 = 0,0590$, $\hat{b}_2 = 0,0511$, $\hat{b}_3 = -0,3588$.

Виконаємо перевірку нульової гіпотези щодо нормальності розподілу залишків ε лінійної регресії для ненормалізованих даних.

Для 45 проектів web-застосунків, що створюються з використанням фреймворку Symfony, отримане значення критерія Пірсона χ^2 для лінійної регресії для ненормалізованих даних, визначене за формулою (2.7), дорівнює 3,519. В свою чергу для 2 ступенів вільності та на рівні значимості 0,05 критичне значення χ^2 дорівнює 7,814. Так як $\chi^2 < \chi^2_{кр}$, ми приймаємо нульову гіпотезу щодо нормальності розподілу залишків ε на рівні значимості 0,05.

Значення оцінки математичного сподівання залишків ε лінійної регресії для ненормалізованих даних дорівнює $-1,61e^{-14}$, а значення оцінки середньоквадратичного дорівнює 0,497.

Отриманий результат може слугувати підставою для застосування багатофакторної лінійної регресійної моделі, однак розраховані оцінки середньої величини відносної похибки *MMRE* та відсотку прогнозованих результатів *PRED(0,25)*, що дорівнюють 0,273 та 0,578 та свідчать про незадовільну якість побудованої моделі.

Для визначення довірчих інтервалів та інтервалів передбачення лінійної регресії застосована у формулах (1.2) та (1.3) матриця S_X має вигляд

$$S_X = \begin{pmatrix} 21368,311 & 5,409 & -20,520 \\ 5,409 & 46,693 & 0,772 \\ -20,520 & 0,772 & 1,091 \end{pmatrix}$$

В таблиці 2.3 наведено довірчі інтервали лінійної та нелінійної багатофакторних регресійних моделей та їх ширини. Довірчі інтервали лінійної та нелінійної регресії було визначено за формулами (1.2) та (1.8) відповідно.

Таблиця 2.3 – Порівняння ширин довірчих інтервалів лінійної та нелінійної регресії

№	KLOC (Y)	Межі довірчих інтервалів				Ширина інтервалу	
		Лінійна регресія		Нелінійна регресія		Лінійна регресія	Нелінійна регресія
		LB	UB	LB	UB		
1	2	3	4	5	6	7	8
1	0,487	0,6145	1,3874	0,4393	0,6437	0,7729	0,2044
2	0,512	0,2593	0,7622	0,4285	0,5707	0,5029	0,1422
3	0,408	0,2593	0,7622	0,4285	0,5707	0,5029	0,1422
4	0,807	0,5314	1,0532	0,5276	0,6968	0,5218	0,1692
5	0,506	-0,2597	0,4078	0,3937	0,5391	0,6675	0,1453
6	0,514	-0,3141	0,3469	0,3954	0,5452	0,6610	0,1498
7	0,812	1,2614	1,8487	0,9961	1,2851	0,5873	0,2890
8	0,716	-0,2598	0,5685	0,4464	0,6563	0,8283	0,2099
9	1,037	1,0950	1,6069	0,9974	1,2539	0,5120	0,2565
10	0,703	-0,0845	0,6518	0,5100	0,7039	0,7363	0,1939
11	1,152	0,4952	0,9510	0,7740	0,9319	0,4558	0,1579
12	1,630	1,2798	1,7295	1,1827	1,4316	0,4497	0,2489
13	0,823	1,0253	1,4115	1,0822	1,2647	0,3862	0,1825
14	1,813	1,0564	1,4615	1,1588	1,3562	0,4051	0,1974
15	2,427	2,0282	2,7177	1,7646	2,2976	0,6895	0,5330
16	0,837	0,7998	1,2518	1,0003	1,2038	0,4520	0,2035
17	0,990	0,7710	1,5507	0,9956	1,3625	0,7797	0,3669
18	0,813	0,7260	1,2211	0,9783	1,1901	0,4951	0,2118
19	2,193	2,4491	3,3404	2,1217	2,9070	0,8913	0,7852
20	0,740	0,6341	1,2244	0,8989	1,1669	0,5903	0,2680
21	0,799	0,5340	1,1932	0,8586	1,1484	0,6592	0,2898
22	2,165	1,9716	2,5751	1,9011	2,4569	0,6036	0,5558
23	1,056	1,1486	1,6669	1,3010	1,6070	0,5183	0,3059
24	2,191	1,3023	1,8368	1,4264	1,7860	0,5345	0,3596
25	3,100	2,1967	2,6454	2,1499	2,6042	0,4487	0,4543
26	2,289	1,6070	2,3863	1,5925	2,1911	0,7793	0,5987
27	2,500	2,6105	3,1389	2,5606	3,1882	0,5284	0,6276
28	1,844	2,2838	2,6403	2,1843	2,5840	0,3564	0,3996
29	1,803	2,0791	2,6699	1,9558	2,5361	0,5909	0,5803
30	2,404	2,0848	2,5719	1,9244	2,4034	0,4871	0,4790

Продовження табл. 2.3

1	2	3	4	5	6	7	8
31	3,046	2,4521	3,1216	2,3032	3,0748	0,6695	0,7716
32	2,702	2,7006	3,1224	2,6965	3,2746	0,4217	0,5781
33	5,571	3,7664	4,7388	4,0902	5,9170	0,9725	1,8268
34	2,729	2,7401	3,1082	2,6907	3,1944	0,3682	0,5038
35	2,190	2,2178	2,7287	2,0887	2,6122	0,5109	0,5236
36	3,669	3,0323	3,5538	3,0904	3,9279	0,5215	0,8375
37	2,436	2,6389	3,0444	2,4621	2,9457	0,4055	0,4835
38	1,682	2,0791	2,6932	1,6229	2,1868	0,6141	0,5639
39	4,107	3,2258	3,7350	2,9247	3,5982	0,5092	0,6734
40	3,392	2,9741	3,6038	2,3366	3,0527	0,6297	0,7160
41	3,034	3,5904	4,1712	3,2789	4,1013	0,5809	0,8224
42	5,730	3,9027	4,8690	3,4908	5,0715	0,9663	1,5807
43	4,047	4,1865	5,0151	3,4408	4,5997	0,8286	1,1589
44	5,230	4,6572	5,5134	4,1012	5,4202	0,8562	1,3190
45	5,993	5,6767	6,6322	6,2092	8,4862	0,9555	2,2771

З таблиці 2.3 бачимо, що ширини довірчих інтервалів при застосуванні нелінійної регресії на основі одновимірного перетворення десяткового логарифму менші за відповідні ширини інтервалів лінійної регресії для 27 рядків: 1-24, 26, 30, 38.

В таблиці 2.4 наведено інтервали передбачення лінійної та нелінійної багатофакторних регресійних моделей та їх ширини. Інтервали передбачення лінійної та нелінійної регресії було визначено за формулами (1.3) та (1.9) відповідно.

Таблиця 2.4 – Порівняння ширин інтервалів передбачення лінійної та нелінійної регресії

№	KLOC (Y)	Межі інтервалів передбачення				Ширина інтервалу	
		Лінійна регресія		Нелінійна регресія		Лінійна регресія	Нелінійна регресія
		LB	UB	LB	UB		
1	2	3	4	5	6	7	8
1	0,487	-0,1192	2,1212	0,3282	0,8617	2,2404	0,5335
2	0,512	-0,5703	1,5918	0,3104	0,7879	2,1621	0,4775
3	0,408	-0,5703	1,5918	0,3104	0,7879	2,1621	0,4775
4	0,807	-0,2910	1,8756	0,3810	0,9648	2,1666	0,5838
5	0,506	-1,0291	1,1772	0,2879	0,7373	2,2062	0,4494

Продовження табл. 2.4

1	2	3	4	5	6	7	8
6	0,514	-1,0857	1,1185	0,2898	0,7440	2,2043	0,4542
7	0,812	0,4634	2,6467	0,7134	1,7944	2,1833	1,0810
8	0,716	-0,9757	1,2844	0,3338	0,8776	2,2601	0,5438
9	1,037	0,2688	2,4331	0,7076	1,7676	2,1643	1,0600
10	0,703	-0,8304	1,3977	0,3739	0,9602	2,2280	0,5863
11	1,152	-0,3527	1,7989	0,5400	1,3357	2,1517	0,7958
12	1,630	0,4295	2,5798	0,8268	2,0476	2,1504	1,2208
13	0,823	0,1494	2,2874	0,7459	1,8348	2,1380	1,0889
14	1,813	0,1882	2,3297	0,7992	1,9664	2,1415	1,1672
15	2,427	1,2665	3,4795	1,2679	3,1974	2,2130	1,9295
16	0,837	-0,0497	2,1012	0,6977	1,7258	2,1509	1,0281
17	0,990	0,0395	2,2822	0,7278	1,8638	2,2427	1,1360
18	0,813	-0,1066	2,0537	0,6853	1,6990	2,1603	1,0137
19	2,193	1,7527	4,0367	1,5516	3,9751	2,2839	2,4235
20	0,740	-0,1628	2,0213	0,6452	1,6256	2,1841	0,9804
21	0,799	-0,2382	1,9655	0,6228	1,5832	2,2037	0,9604
22	2,165	1,1795	3,3672	1,3624	3,4283	2,1877	2,0660
23	1,056	0,3248	2,4906	0,9168	2,2805	2,1658	1,3638
24	2,191	0,4847	2,6544	1,0104	2,5215	2,1697	1,5111
25	3,100	1,3460	3,4962	1,5035	3,7239	2,1502	2,2204
26	2,289	0,8754	3,1180	1,1662	2,9920	2,2426	1,8258
27	2,500	1,7906	3,9588	1,8099	4,5107	2,1682	2,7008
28	1,844	1,3956	3,5285	1,5131	3,7302	2,1328	2,2170
29	1,803	1,2824	3,4666	1,4033	3,5347	2,1843	2,1313
30	2,404	1,2491	3,4076	1,3618	3,3964	2,1585	2,0346
31	3,046	1,6835	3,8903	1,6696	4,2417	2,2068	2,5721
32	2,702	1,8391	3,9839	1,8876	4,6778	2,1447	2,7902
33	5,571	3,0942	5,4110	3,0436	7,9516	2,3168	4,9079
34	2,729	1,8568	3,9916	1,8666	4,6047	2,1348	2,7381
35	2,190	1,3913	3,5553	1,4788	3,6895	2,1640	2,2107
36	3,669	2,2098	4,3763	2,2012	5,5144	2,1665	3,3132
37	2,436	1,7708	3,9124	1,7134	4,2330	2,1416	2,5196
38	1,682	1,2909	3,4815	1,1802	3,0072	2,1907	1,8270
39	4,107	2,3986	4,5622	2,0578	5,1141	2,1636	3,0564
40	3,392	2,1914	4,3865	1,6810	4,2432	2,1951	2,5622
41	3,034	2,7900	4,9716	2,3216	5,7924	2,1816	3,4708
42	5,730	3,2287	5,5429	2,6010	6,8064	2,3142	4,2054
43	4,047	3,4707	5,7309	2,4954	6,3423	2,2602	3,8469
44	5,230	3,9501	6,2206	2,9625	7,5035	2,2705	4,5409
45	5,993	4,9996	7,3093	4,5370	11,6139	2,3097	7,0769

З таблиці 2.4 бачимо, що ширини інтервалів передбачення при застосуванні нелінійної регресії на основі одновимірного перетворення десяткового логарифму менші за відповідні ширини інтервалів лінійної регресії для 27 рядків: 1-18, 20-24, 26, 29-30, 38.

Як для довірчих інтервалів лінійної регресії, так і для інтервалів передбачення спостерігаємо наявність від’ємних значень, що не є допустимим.

Порівнюючи отриману лінійну багатofакторну регресійну модель з нелінійною, ми бачимо суттєве покращення оцінок якості моделі до того рівня, який слугує теоретичним обґрунтуванням можливості застосування нелінійної регресійної моделі. Так R^2 , $MMRE$ та $PRED(0,25)$ для лінійної моделі дорівнюють 0,899, 0,273 та 0,578 відповідно. Відповідні оцінки для нелінійної моделі – 0,900, 0,187 та 0,756. Бачимо покращення показників якості оцінок $MMRE$ та $PRED(0,25)$ яке є суттєвим.

Іншим показником оцінювання якості моделі, який може претендувати на незалежність такого оцінювання, – є порівняння моделей. В роботах [15, 16] було побудовано нелінійні регресійні моделі для оцінювання розміру web-застосунків, що створюються з використанням PHP фреймворків CakePHP та Yii відповідно, в яких використовується такий же набір метрик, як і в даній роботі: кількість класів, середня кількість методів на кожен клас, глибина дерева наслідування. Для емпіричних даних з даної роботи були застосовані моделі, побудовані для фреймворку CakePHP [15] та Yii [16]. У таблиці 2.5 наведено порівняння отриманих оцінок.

Таблиця 2.5 – Порівняння оцінок моделей

Оцінки якості моделі	Фреймворк		
	Symfony	CakePHP	Yii
R^2	0,9	0,723	0,753
$MMRE$	0,187	0,307	0,222
$PRED(0,25)$	0,756	0,533	0,600

Порівняння моделей показує, що для отримання якісних оцінок моделей та якісної моделі в цілому можна покладатись лише на модель, побудовану під конкретний фреймворк.

2.3 Висновки до розділу 2

В даному розділі було побудовано нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony на основі нормалізуючого перетворення десятичного логарифму.

– Виконано пошук web-застосунків, що створюються з використанням фреймворку Symfony з переліку проектів, розміщених на сайті GitHub. Збір метрик проектів програмного забезпечення проводився за допомогою інструменту PhpMetrics.

– Виконано перевірку початкових даних на нормальність за допомогою розрахунку квадрату відстані Махаланобіса та багатовимірного ексцесу. Визначено, що багатовимірні дані є негаусівськими.

– Виконано перевірку на наявність мультиколінеарності між майбутніми предикторами багатфакторних регресійних моделей за допомогою розрахунку коефіцієнтів впливу дисперсії VIFs. Мультиколінеарність відсутня.

– Виконано вилучення викидів за допомогою методу, що базується на нормалізуючих перетвореннях та розрахунку квадрату відстані Махаланобіса. Всього вилучено 5 проектів програмного забезпечення.

– Побудовано багатфакторну нелінійну регресійну з використанням нормалізуючого перетворення десятичного логарифма.

– Побудовано багатфакторну лінійну регресійну модель з метою порівняння з побудованою нелінійною регресійною моделлю.

– Виконано порівняння побудованих моделей. Оцінки якості нелінійної регресійної моделі R^2 , $MMRE$ та $PRED(0,25)$ дорівнюють 0,9, 0,187 та 0,756. Оцінки якості лінійної регресійної моделі R^2 , $MMRE$ та $PRED(0,25)$ дорівнюють

0,899, 0,273 та 0,578 відповідно. Спостерігається суттєве покращення показника оцінки *MMRE* та оцінки *PRED(0,25)* для нелінійної регресійної моделі.

– Виконано обчислення довірчих інтервалів та інтервалі прогнозування лінійної та нелінійної регресій. Визначено, що для 60% даних отримано звуження довірчих інтервалів та інтервалів прогнозування у випадку використання нелінійної регресійної моделі у порівнянні з лінійною.

– Виконано порівняння оцінок якості побудованої в межах цієї роботи нелінійної регресійної моделі з оцінками якості моделей для фреймворків CakePHP та Yii, якщо застосувати їх до даних по web-застосункам, що створюються з використанням фреймворку Symfony. Порівняння оцінок підкреслило необхідність побудови моделей під конкретний фреймворк.

З РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY

3.1 Постановка задачі на розробку проекту програмного забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony

Основна ціль розробки програмного забезпечення в межах роботи – забезпечити кінцевого користувача простим та зрозумілим у користуванні програмним засобом для можливості оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Важливим є також питання щодо можливості виконання програми у різних операційних системах (Windows, Linux, Mac OS). Для того, щоб досягти поставленої цілі необхідно обрати підходящі програмні засоби – мову програмування, додаткові бібліотеки, тощо. Такому переліку вимог відповідає мова Python (<https://www.python.org/>).

Python – мова програмування високого рівня з динамічною строгою типізацією з відкритим кодом. Основними перевагами вибору Python для розробки програмного забезпечення в межах даної роботи є:

- Можливість вільного використання в комерційному середовищі.
- Сумісність з модулями інших мов програмування, за рахунок чого можливе застосування вже готових рішень.
- Наявність бібліотек для NumPy та SciPy для виконання математичних обчислень.
- Сумісність з популярною бібліотекою Qt для створення графічного інтерфейсу користувача.
- Наявність багатифункціональних середовищ розробки (IDE) та текстових редакторів, в тому числі й безкоштовних, для написання програмного коду мовою Python.

Підсумовуючи вищенаведене, бачимо, що Python є доцільним вибором для розробки в межах кваліфікаційної роботи.

Розробка програми для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, буде проводитись згідно технічного завдання, наведеного у додатку А. Розробка програми передбачає розробку ескізного, технічного та робочого проектів.

3.2 Розробка ескізного проекту програмного забезпечення оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony

В рамках ескізного проекту необхідно виконати моделювання функціональності програми на концептуальному рівні, для цього використовуватимемо діаграму варіантів використання та діаграму діяльності.

3.2.1 Побудова моделі варіантів використання програмного застосунку «Symfony LOC estimation»

Діаграма варіантів використання (або прецедентів) – один з способів побудови абстрактної моделі концепції розроблюваного програмного забезпечення без відображення будь-яких деталей реалізації. Діаграми варіантів використання часто зрозумілі навіть людям, що не займаються розробкою, адже такі діаграми містять функціональні вимоги, які записані формальною мовою відповідно до предметної галузі.

Користувачем програми буде один користувач, отже діаграма варіантів використання буде мати лише одного актора

Виявлені варіанти використання наведено у таблиці 3.1.

Таблиця 3.1 – Виявлені варіанти використання

Основні актори	Найменування	Формулювання
Користувач	Ввести кількість класів	Вводить кількість класів web-застосунку
Користувач	Ввести середню кількість методів на кожен клас	Вводить середню кількість методів на кожен клас web-застосунку
Користувач	Ввести глибину дерева наслідування	Вводить глибину дерева наслідування web-застосунку
Користувач	Ввести довірчу ймовірність	Вводить довірчу ймовірність для подальшого оцінювання інтервалу передбачення та довірчого інтервалу
Користувач	Виконати оцінювання розміру	Отримує розмір web-застосунку, інтервал передбачення, довірчий інтервал
Користувач	Виконати оцінювання інтервалу передбачення	Отримує інтервал передбачення web-застосунку
Користувач	Виконати оцінювання довірчого інтервалу	Отримує довірчий інтервал web-застосунку.

Діаграму варіантів використання наведено на рисунку 3.1.

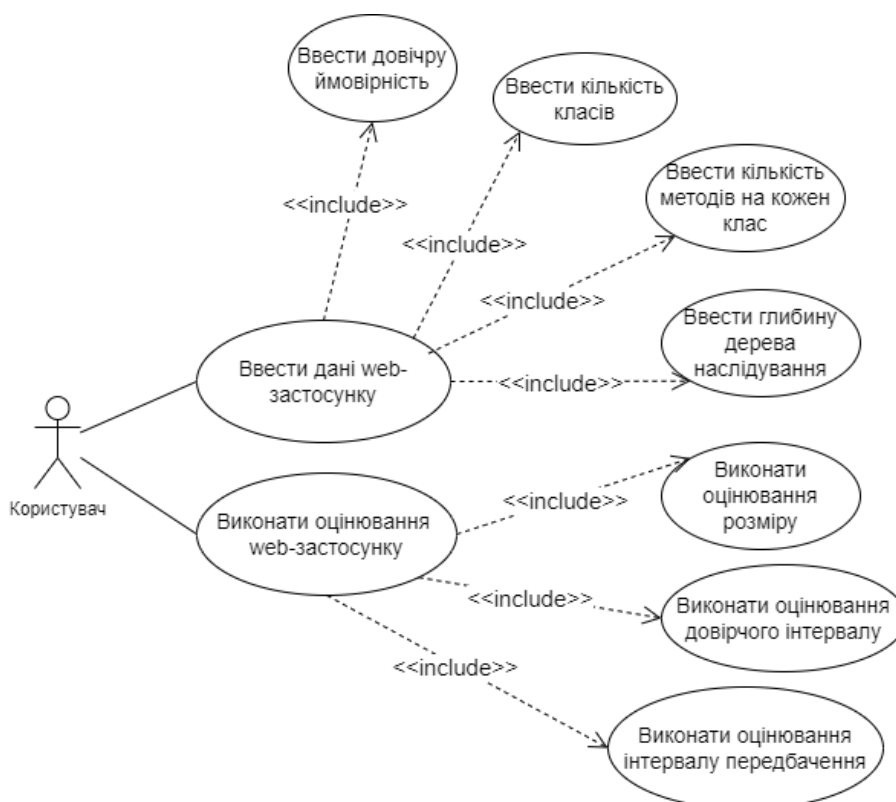


Рисунок 3.1 – Діаграма варіантів використання

Побудована діаграма варіантів використання є базисом для наступного моделювання та проведення розробки в межах ескізного, технічного та робочого проектів.

3.2.2 Специфікації варіантів використання програмного застосунку «Symfony LOC estimation»

Наведемо детальну специфікацію варіантів використання

1. Найменування: Ввести кількість класів web-застосунку

Актори: користувач

Зв'язок з іншими варіантами використання: включається варіантом використання «Ввести дані web-застосунку».

Короткий опис: варіант використання «Ввести кількість класів» передбачає введення кількості класів оцінюваного web-застосунку у відповідне поле графічного інтерфейсу.

Основний потік подій: прецедент починає виконання при взаємодії з відповідним полем графічного інтерфейсу користувача.

Базовий потік: Введення кількості класів web-застосунку:

- 1) Користувач заповнює поле для вказання кількості класів web-застосунку.

Спеціальні вимоги:

- 1) Значення кількості класів має бути цілим додатнім числом.

Аналогічним чином можна описати специфікації для варіантів використання «Ввести кількість методів на кожен клас», «Ввести глибину дерева наслідування», «Ввести довірчу ймовірність», з тою відмінністю, що значення для зазначених вище варіантів використання – дійсні додатні числа.

2. Найменування: Ввести дані web-застосунку

Актори: користувач

Зв'язок з іншими варіантами використання: включає варіанти використання «Ввести кількість класів web-застосунку», «Ввести глибину дерева наслідування web-застосунку», «Ввести глибину дерева наслідування web-застосунку», «Ввести довірчу ймовірність».

Короткий опис: варіант використання «Ввести дані web-застосунку» покликаний об'єднати чотири більш прості та логічно-зв'язні операції в одну за допомогою заповнення відповідних полів графічного інтерфейсу.

– Для варіанту використання «Ввести кількість класів» – відповідна кількість класів.

– Для варіанту використання «Ввести кількість методів на кожен клас» – середня кількість методів на кожен клас web-застосунку.

– Для варіанту використання «Ввести глибину дерева наслідування», – середня глибина дерева наслідування web-застосунку.

– Для варіанту використання «Ввести довірчу ймовірність» – значення довірчої ймовірності для оцінювання інтервалу передбачення та довірчого інтервалу.

Основний потік подій: прецедент починає виконання при взаємодії з відповідними полями графічного інтерфейсу користувача.

Базовий потік: Введення даних web-застосунку:

- 1) Користувач заповнює поле для вказання кількості класів web-застосунку.
- 2) Користувач заповнює поле для вказання середньої кількості методів на кожен клас web-застосунку.
- 3) Користувач заповнює поле для вказання середньої глибини дерева наслідування web-застосунку.
- 4) Користувач заповнює поле для вказання довірчої ймовірності.

Спеціальні вимоги:

- 1) Значення кількості класів має бути цілим додатнім числом.
- 2) Значення середньої кількості методів, глибини дерева наслідування, довірчої ймовірності та точного розміру web-застосунку має бути дійсним додатнім числом.

3. *Найменування:* Виконати оцінювання розміру

Актори: користувач

Зв'язок з іншими варіантами використання: включається варіантом використання «Виконати оцінювання web-застосунку».

Короткий опис: варіант використання «Виконати оцінювання розміру» покликаний виконати розрахунки, пов'язані з визначенням розміру web-застосунку. Результатом виконання варіанту використання буде відповідне значення на екрані графічного інтерфейсу програми.

Основний потік подій: прецедент починає виконання після заповнення полів для введення даних з метрик програмного забезпечення, довірчої ймовірності за допомогою графічного інтерфейсу користувача та ініціювання оцінювання за допомогою відповідної кнопки.

Базовий потік: Виконати оцінювання розміру web-застосунку:

1) Користувач вказує дані трьох метрик програмного забезпечення та довірчу ймовірність для оцінювання розміру web-застосунку.

2) Система відображає визначену оцінку розміру web-застосунку.

Спеціальні вимоги: відсутні.

Альтернативні потоки: відсутні.

4. *Найменування:* Виконати оцінювання довірчого інтервалу

Актори: користувач

Зв'язок з іншими варіантами використання: включається варіантом використання «Виконати оцінювання web-застосунку».

Короткий опис: варіант використання «Виконати оцінювання довірчого інтервалу» покликаний виконати розрахунки, пов'язані з визначенням меж довірчого інтервалу отриманої оцінки розміру web-застосунку. Результатом виконання варіанту використання буде відповідне значення на екрані графічного інтерфейсу програми.

Основний потік подій: прецедент починає виконання після заповнення полів для введення даних з метрик програмного забезпечення, довірчої ймовірності за

допомогою графічного інтерфейсу користувача та ініціювання оцінювання за допомогою відповідної кнопки.

Базовий потік: Виконати оцінювання довірчого інтервалу web-застосунку:

1) Користувач вказує дані трьох метрик програмного забезпечення та довірчу ймовірність для оцінювання розміру web-застосунку.

2) Система відображає визначені межі довірчого інтервалу розміру web-застосунку.

Спеціальні вимоги: відсутні.

Альтернативні потоки: відсутні.

Аналогічно можна описати поведінку для варіантів використання «Виконати оцінювання інтервалу передбачення».

5. Найменування: Виконати оцінювання web-застосунку

Актори: користувач

Зв'язок з іншими варіантами використання: включає варіанти використання «Виконати оцінювання розміру», «Виконати оцінювання довірчого інтервалу», «Виконати оцінювання інтервалу передбачення».

Короткий опис: варіант використання «Виконати оцінювання розміру web-застосунку» покликаний виконати розрахунки, пов'язані з визначенням розміру web-застосунку, визначенням довірчого інтервалу та інтервалу передбачення. Результатом виконання варіанту використання будуть відповідні значення на екрані графічного інтерфейсу програми.

Основний потік подій: прецедент починає виконання після заповнення полів для введення даних з метрик програмного забезпечення, довірчої ймовірності за допомогою графічного інтерфейсу користувача.

Базовий потік: Виконати оцінювання розміру web-застосунку:

1) Користувач вказує дані трьох метрик програмного забезпечення та довірчу ймовірність для оцінювання розміру web-застосунку.

2) Система відображає визначену оцінку розміру web-застосунку, інтервал передбачення та довірчий інтервал.

Спеціальні вимоги: відсутні.

Альтернативні потоки: відсутні.

3.2.3 Моделювання сценарію виконання програмного застосунку «Symfony LOC estimation» за допомогою діаграми діяльності

Призначення діаграми діяльності – моделювання процесу виконання операцій. Діаграма діяльності схожа на діаграму станів, але її відмінність в тому, що на переходах між станами не вказується сигнатура події, а також відрізняється семантика самих станів. Моделювання поведінки за допомогою діаграми діяльності необхідне для відображення алгоритму виконання операції. Кожна окремо взята діяльність на діаграмі – сукупність обчислень або дій, що виконуються автоматом. Діаграма відображає процес виконання певної операції з акцентом на результаті діяльності на кожному з етапів.

Елементами діаграми є:

- Стан – певна ситуація, в якій опинився об’єкт в результаті виконання вхідної дії. Стан має хоча б один вихідний перехід.
- Перехід між станами – передача управління в результаті виконання вихідного стану до подальшого.

За допомогою діаграми діяльності змоделюємо процес виконання операцій під час роботи програми для оцінювання розміру web-застосунку. Діаграму діяльності наведено на рисунку 3.2.

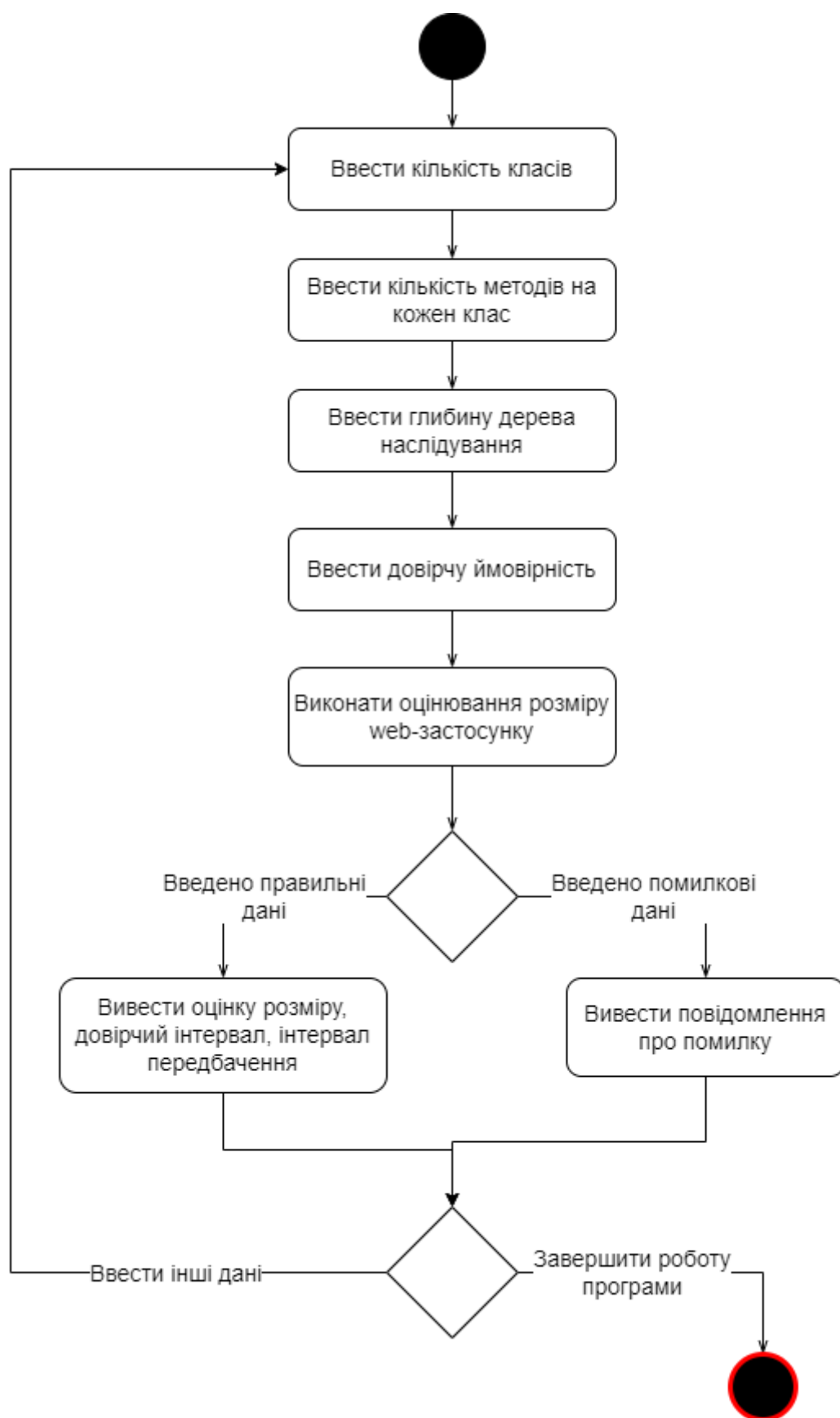


Рисунок 3.2 – Діаграма діяльності

Зображений на діаграмі діяльності потік подій дозволяє отримати краще концептуальне представлення роботи програми та проміжних результатів.

3.3 Розробка технічного проекту програмного забезпечення оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony

3.3.1 Статична модель програмного застосунку «Symfony LOC estimation» у вигляді діаграми класів

Один з способів подання статичної моделі програмного забезпечення – є моделювання відношень між класами за допомогою діаграми класів. Ця діаграма може містити пакети, інтерфейси, відношення між класами та класи. Діаграма класів також описує внутрішню структуру класів, що моделюються.

Статичну структуру програми з переліком класів та зв'язків між ними показана на рисунку 3.3.

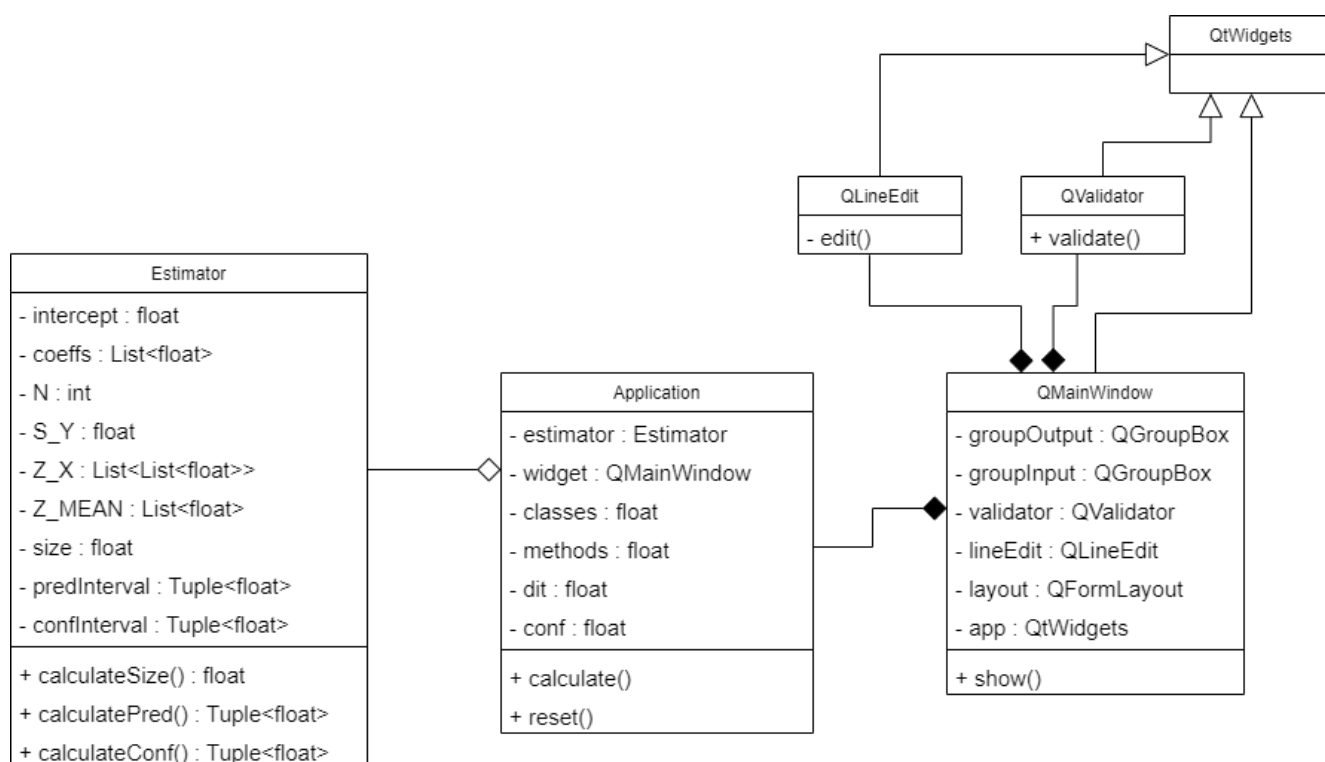


Рисунок 3.3 – Діаграма класів

Отримані результати при виконанні статичного моделювання будуть використані в подальшому для моделювання та при безпосередньому написанні програмного коду.

3.3.2 Специфікації класів програмного застосунку «Symfony LOC estimation»

Коротку специфікацію класів можна представити наступним чином:

- Клас QMainWindow – клас, що представляє собою головний екран програми.

- Клас QWidget – базовий клас для усіх компонентів користувацького інтерфейсу в бібліотеці Qt.

- Клас QLineEdit – клас, що дозволяє виконувати операції з введення та виведення даних до користувацького інтерфейсу.

- Клас QValidator – клас, що дозволяє проводити валідацію даних в реальному часі, не дозволяючи вводити дані, що не передбачені для певного поля графічного інтерфейсу.

- Клас Application – головний клас програми, містить введені користувачем дані та клас, що відповідає за виконання розрахунків.

- Клас Estimator – клас, що відповідає за оцінювання web-застосунку: оцінювання розміру web-застосунку, визначення довірчого інтервалу та інтервалу передбачення.

3.3.3 Динамічна модель програмного застосунку «Symfony LOC estimation» у вигляді діаграм послідовностей

Моделювання динамічної поведінки програмного забезпечення здійснюється за допомогою діаграми послідовності – діаграми, на якій відображають взаємодію об'єктів, яка впорядкована за часом. Діаграма послідовності не містить статичні

зв'язки об'єктів, вона містить лише послідовність виконуваних операцій. Діаграма складається з:

- об'єктів, тобто, дійових осіб, що беруть участь в виконанні операції; об'єкти можуть бути анонімними;
- фокусу управління – фокус управління передбачає передачу управління об'єкту для виконання операції;
- повідомлень, якими обмінюються об'єкти.

Діаграму послідовності для варіанту використання «Ввести дані web-застосунку» наведено на рисунку 3.4:

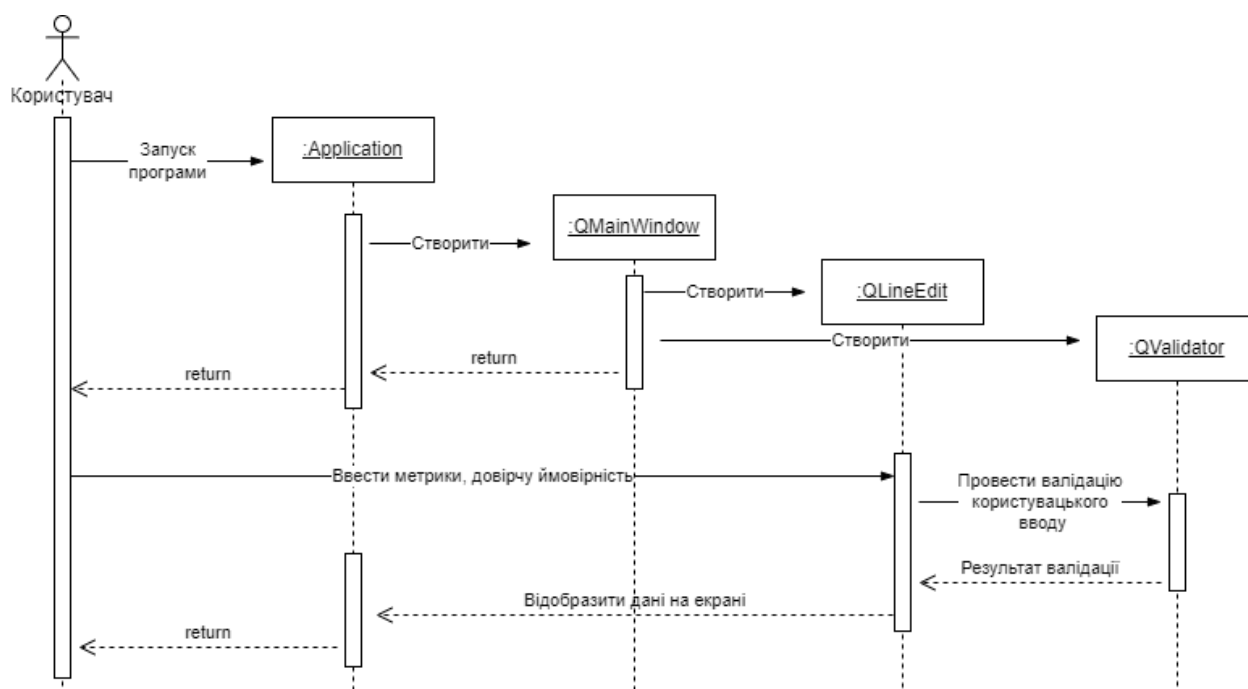


Рисунок 3.4 – Діаграма послідовності для варіанту використання «Ввести дані web-застосунку»

Діаграму послідовності для варіанту використання «Виконати оцінювання розміру web-застосунку» наведено на рисунку 3.5:

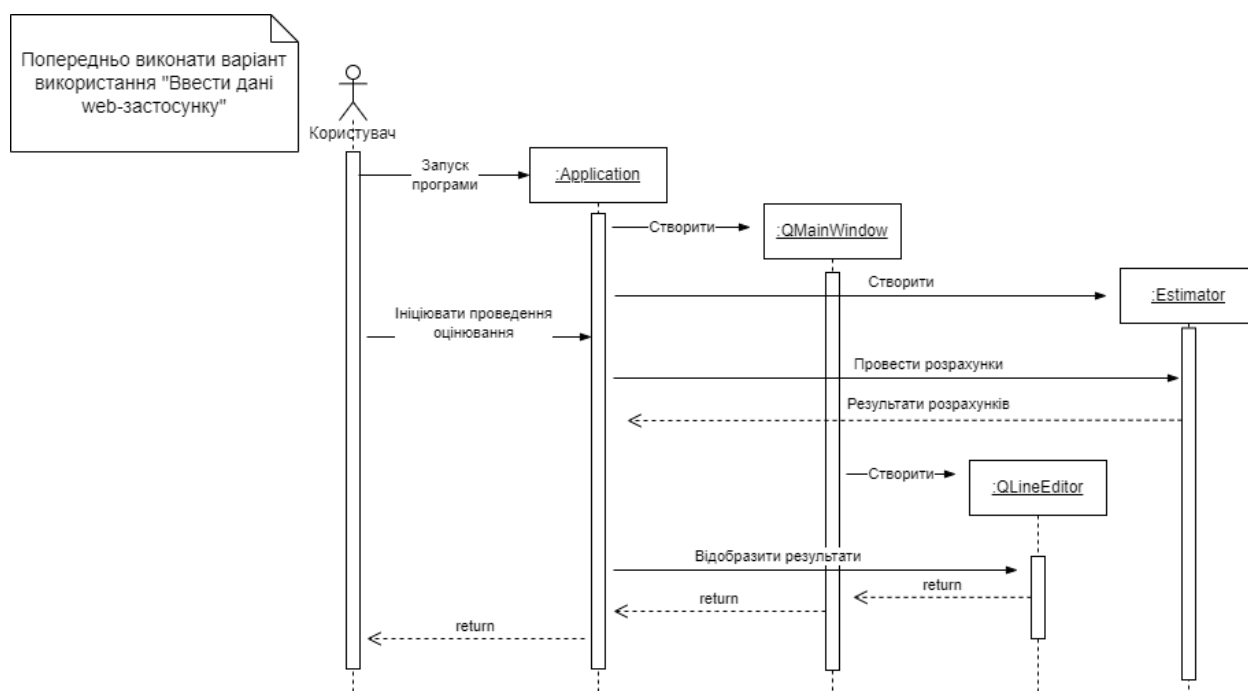


Рисунок 3.5 – Діаграма послідовності для варіанту використання «Виконати оцінювання розміру web-застосунку»

Отримані діаграми послідовностей допомагають визначити алгоритми виконання операцій для обраних варіантів використання з вказанням об'єктів-учасників та операцій, що полегшує роботу під час їх безпосередньої їх реалізації.

3.4 Розробка робочого проекту програмного забезпечення оцінювання розміру розробки web-застосунків, що створюються з використанням фреймворку Symfony

3.4.1 Кодування програмного застосунку «Symfony LOC estimation»

Однією задачею, що виконується в межах робочого проекту є написання коду. Згідно описаних вище причин було обрано мову програмування Python та безпосередньо виконано кодування. Нижче наведено програмний код класу, який відповідає за виконання розрахунків:

```

class Estimator:
    intercept = -1.63782126445733
    coeffs = np.array([0.96641578624015, 0.96301298247654, -0.46272880601894])
    N = 45
    S_Y = 0.0090855323014
    ZX = np.array([
        [0.27127997, -0.04335353, 0.25063032],
        [-0.04335353, 1.53383441, -0.71641416],
        [0.25063032, -0.71641416, 10.63646880]])
    Z_MEAN = [1.4497370, 0.5378432, 0.1425785]

    def __init__(self, vector, conf):
        self.metrics = vector
        self.confLevel = conf
        self.zx = np.log10(vector) - self.Z_MEAN
        self.se_z = np.matmul(np.matmul(self.zx.T, self.ZX), self.zx)

    def calculateSize(self):
        value = (10 ** self.intercept) * (self.metrics[0] ** self.coeffs[0]) * (
            self.metrics[1] ** self.coeffs[1]) * (self.metrics[2] **
self.coeffs[2])
        self.size = value
        return value

    def calculateConf(self):
        conf_val = (sp.stats.t.ppf(1 - (1 - self.confLevel) / 2,
                                self.N - 4) * math.sqrt(self.S_Y * (1.0 /
self.N + self.se_z)))
        conf_interval = (10 ** (math.log10(self.size) - conf_val),
                        10 ** (math.log10(self.size) + conf_val))
        return conf_interval

    def calculatePred(self):
        pred_val = (sp.stats.t.ppf(1 - (1 - self.confLevel) / 2, self.N - 4)
                    * math.sqrt(self.S_Y * (1 + 1.0 / self.N + self.se_z)))
        pred_interval = (10 ** (math.log10(self.size) - pred_val),
                        10 ** (math.log10(self.size) + pred_val))
        return pred_interval

```

Повний текст програми наведено в додатку Г.

3.4.2 Тестування програмного застосунку «Symfony LOC estimation»

Після завершення написання програмного коду було виконано тестування, розроблено супровідну програмну документацію, виконано випробування програмного забезпечення.

Для розробки тестів в межах даної роботи використаємо метод еквівалентного розбиття (класів еквівалентності).

Виконаємо тестування варіанту використання «Ввести кількість класів» за допомогою еквівалентного розбиття. Класи еквівалентності наведено у таблиці 3.2.

Таблиця 3.2 – Класи еквівалентності для тестування варіанту використання «Ввести кількість класів»

<i>Вхідні данні</i>	<i>Правильні класи</i>	<i>Неправильні класи</i>
Кількість класів	Введено натуральне число (1)	Введено дійсне число (2). Введено від'ємне число (3). Введено не число (4) Поле пуста(5).

Результати виконання тестування з використанням правильних класів еквівалентності наведено у таблиці 3.3.

Таблиця 3.3 – Тести правильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
1	1	32	Можливість виконати оцінювання	Тест пройдено

Результати виконання тестування з використанням неправильних класів еквівалентності наведено у таблиці 3.4.

Таблиця 3.4 – Тести неправильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
2	2	2,5	Повідомлення про недопустиме значення кількості класів	Тест пройдено
3	3	-2,5	Повідомлення про недопустиме значення кількості класів	Тест пройдено
4	4	2fia!i!»	Повідомлення про недопустиме значення кількості класів	Тест пройдено
5	5	Відсутні	Повідомлення про необхідність заповнити поле	Тест пройдено

Класи еквівалентності для варіанту використання «Ввести кількість методів на кожен клас» наведено у таблиці 3.5.

Таблиця 3.5 – Класи еквівалентності для тестування варіанту використання «Ввести кількість методів на кожен клас»

<i>Вхідні данні</i>	<i>Правильні класи</i>	<i>Неправильні класи</i>
Середня кількість методів на кожен клас (глибина дерева наслідування)	Введено дійсне додатне число (1)	Введено від'ємне число (2). Введено не число (3) Поле пусте(4).

Результати виконання тестування з використанням правильних класів еквівалентності наведено у таблиці 3.6.

Таблиця 3.6 – Тести правильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
1	1	7,65	Можливість виконати оцінювання	Тест пройдено

Результати виконання тестування з використанням неправильних класів еквівалентності наведено у таблиці 3.7.

Таблиця 3.7 – Тести неправильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
2	2	-2,5	Повідомлення про недопустиме значення кількості методів	Тест пройдено
3	3	2fia!	Повідомлення про недопустиме значення кількості методів	Тест пройдено
4	4	Відсутні	Повідомлення про необхідність заповнити поле	Тест пройдено

Тестування варіанту використання «Ввести глибину дерева наслідування» також вимагає введення дійсного додатного числа та виконується аналогічно варіанту використання «Ввести кількість методів на кожен клас».

Виконаємо тестування варіанту використання «Ввести довірчу ймовірність» за допомогою еквівалентного розбиття. Класи еквівалентності наведено у таблиці 3.8.

Таблиця 3.8 – Класи еквівалентності для тестування варіанту використання «Ввести довірчу ймовірність»

<i>Вхідні дані</i>	<i>Правильні класи</i>	<i>Неправильні класи</i>
Довірча ймовірність	Введено дійсне додатне число в межах від нуля до одиниці включно (1)	Введено від’ємне число (2). Введено не число (3) Поле пусте(4).

Результати виконання тестування з використанням правильних класів еквівалентності наведено у таблиці 3.9.

Таблиця 3.9 – Тести правильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
1	1	0,95	Можливість виконати оцінювання	Тест пройдено

Результати виконання тестування з використанням неправильних класів еквівалентності наведено у таблиці 3.10.

Таблиця 3.10 – Тести неправильних класів еквівалентності

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
2	2	-2,5	Повідомлення про недопустиме значення довірчої ймовірності	Тест пройдено
3	3	2fia!	Повідомлення про недопустиме значення довірчої ймовірності	Тест пройдено
4	4	Відсутні	Повідомлення про необхідність заповнити поле	Тест пройдено

Результати тестування дозволяють зробити висновок про належну роботу програми.

3.4.3 Випробування програмного застосунку «Symfony LOC estimation»

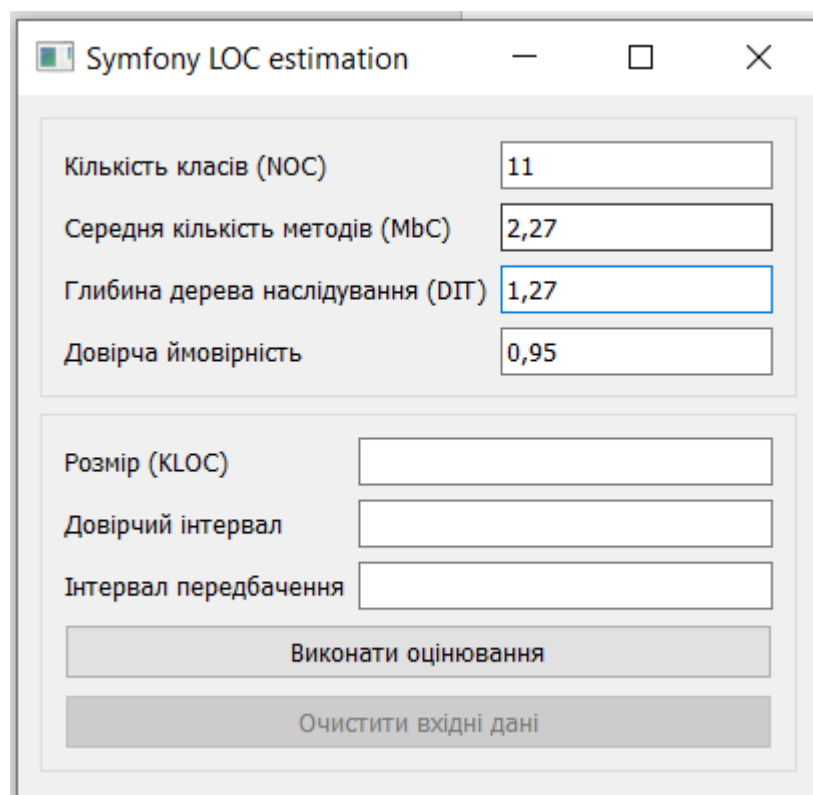
Згідно документу «Програма та методика випробувань», який наведено у Додатку Б, було проведено випробування розробленого програмного застосунку.

Перелік протестованих функцій:

- Введення кількості класів web-застосунку.
- Введення середньої кількості методів на клас web-застосунку.
- Введення глибини дерева наслідування web-застосунку.
- Отримання оцінки розміру web-застосунку, оцінок довірчого інтервалу, інтервалу передбачення web-застосунку.

Програмне забезпечення виконує всі функції належним чином.

Результат роботи програми для оцінювання розміру web-застосунку, що створюється з використанням фреймворку Symfony, наведено на рисунку 3.6 та 3.7.



Кількість класів (NOC)	11
Середня кількість методів (Mbc)	2,27
Глибина дерева наслідування (DIT)	1,27
Довірча ймовірність	0,95
Розмір (KLOC)	
Довірчий інтервал	
Інтервал передбачення	
Виконати оцінювання	
Очистити вхідні дані	

Рисунок 3.6 – Вікно програми до виконання оцінювання

The screenshot shows a window titled "Symfony LOC estimation". It contains two sections of input fields. The top section has four fields: "Кількість класів (NOC)" with value "11", "Середня кількість методів (MbC)" with value "2,27", "Глибина дерева наслідування (DIT)" with value "1,27", and "Довірча ймовірність" with value "0,95". The bottom section has three fields: "Розмір (KLOC)" with value "0.4607", "Довірчий інтервал" with value "0.3937 - 0.5391", and "Інтервал передбачення" with value "0.2879 - 0.7373". At the bottom of the window are two buttons: "Виконати оцінювання" and "Очистити вхідні дані".

Parameter	Value
Кількість класів (NOC)	11
Середня кількість методів (MbC)	2,27
Глибина дерева наслідування (DIT)	1,27
Довірча ймовірність	0,95
Розмір (KLOC)	0.4607
Довірчий інтервал	0.3937 - 0.5391
Інтервал передбачення	0.2879 - 0.7373

Рисунок 3.7 – Результат оцінювання розміру web-застосунку, що створюється з використанням фреймворку Symfony

Отримані результати дозволяють зробити висновок про відповідність розробленої програми висунутим вимогам.

3.5 Висновки до розділу 3

В даному розділі було розроблено проект програмного забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

Також в розділі було:

– Визначено перелік необхідних програмних засобів для проведення розробки в межах даної роботи з обґрунтуванням доцільності їх використання.

– Розроблено ескізний проект програмного забезпечення в рамках якого, визначено варіанти використання, створено діаграму варіантів використання з відповідними детальними специфікаціями варіантів використання.

– Розроблено технічний проект програмного забезпечення в рамках якого, виконано моделювання статичної структури програми за допомогою діаграми класів з супровідною специфікацією класів, наведено алгоритми виконання основних операцій за допомогою діаграм послідовностей.

– Розроблено робочий проект програмного забезпечення в рамках якого, виконано кодування програмного забезпечення, виконано тестування програми за допомогою техніки еквівалентного розбиття, виконано випробовування програми.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ WEB-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ SYMFONY

Вступ

Розробка програмного забезпечення передбачає витрати на купівлю техніки, наймання та підготовку кадрів, витрати на підготовчі роботи та розробку безпосередньо. Визначити економію від використання програмного забезпечення можна зважаючи на витрати при його розробці. Відношення економії до витрат характеризує економічну ефективність. Нашою метою є визначити ключові економічні показники, які слід урахувати, для розроблюваного програмного забезпечення та на їх основі проаналізувати доцільність такої розробки.

4.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення

Можна виділити наступний перелік витрат, що складають витрати на розробку:

- Витрати на заробітну плату.
- Витрати на амортизацію ОЕМ для виконання розробки.
- Витрати на експлуатацію цієї ЕОМ.
- Витрати на засоби розробки.
- Витрати на матеріали та комплектуючі.

Розробку буде виконувати програміст з заробітною платою 10000грн. на місяць.

Вартість сучасного та підходящого для виконання роботи ПК (чи ноутбука) становить приблизно 15000 грн.

Вартість кіловат-години електроенергії становить 1,68 грн.

Також слід врахувати інші можливі витрати, які наведено у таблиці 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

№	Пункти витрат	Сума, грн.
1	Папір для принтера	200
2	Картридж та тонер для принтера	300
3	Використання флеш-накопичувача	120
4	Непередбачувані витрати	400
	Разом	1020

Тоді вартість розробки програмного забезпечення можна розрахувати як

$$C_{\text{пр}} = (Z_{\text{зп}} + Z_{\text{зс}} + Z_{\text{зг}} + Z_{\text{е}}) * T + Z_{\text{м}},$$

де T – тривалість розробки у місяцях

$Z_{\text{зп}}$ – основна та додаткова заробітна плата обслуговуючого персоналу, грн.

$Z_{\text{зс}}$ – відрахування на соціальні заходи (15% від основної та додаткової заробітної плати), грн.

$Z_{\text{е}}$ – витрати на електроенергію, грн.

$Z_{\text{м}}$ – витрати на допоміжні матеріали, грн.

$Z_{\text{е}}$ при витратах в 0,5 кВт з тривалістю роботи на місяць $22 * 8 = 176$ годин та вартості електроенергії 1,68грн становить

$$Z_{\text{е}} = 176 * 1,68 * 0,5 = 147,84 \text{ грн.} \quad (4.1)$$

Деталізовані витрати на розробку програмного забезпечення наведено у таблиці 4.2

Таблиця 4.2 – Витрати на розробку програмного забезпечення

№	Пункти витрат	Одиниця	Кількість
1	Тривалість розробки (T)	Міс.	2
2	Основна та додаткова заробітна плата ($Z_{\text{зп}}$)	Грн.	10000
3	Відрахування на соціальні витрати ($Z_{\text{зс}}$)	Грн.	1500
4	Загальногосподарські витрати ($Z_{\text{зг}}$)	Грн.	1000
5	Витрати на допоміжні матеріали ($Z_{\text{м}}$)	Грн.	1020
6	Витрати на електроенергію $*(Z_{\text{е}})$	Грн.	147,84

За формулою (4.1) виконаємо розрахунок вартості програми:

$$C_{\text{пр}} = (10000 + 1500 + 1000 + 147.84) * 2 + 1020 = 26315,68 \text{ грн.}$$

Витрати на амортизацію становлять 25% від балансової вартості на рік:

$$A_{\text{об}} = 15000 * 0,25 = 3750 \text{ грн.}$$

Для підприємств річні витрати на основні та допоміжні матеріали складають 5% від вартості основного устаткування:

$$B_{\text{м}} = 15000 * 0,05 = 750 \text{ грн.}$$

Річний обсяг робіт ПК у годинах можна визначити як

$$\Phi_{\text{м}} = 264,5 * T_{\text{з}}, \quad (4.2)$$

де $T_{\text{з}}$ – середнє навантаження на устаткування в місяць (6 годин), 264,5 – середня кількість робочих днів на рік.

Отже, річний обсяг роботи ПК складає:

$$\Phi_{\text{м}} = 264,5 * 6 = 1578 \text{ годин}$$

Витрати на електроенергію $З_{\text{е}}$ при 1578 годинах роботи устаткування на рік становлять:

$$З_{\text{е}} = 1578 * 1,68 = 2666,16$$

Експлуатаційні витрати на ПК на рік становлять:

$$З_{\text{зр}} = 3750 + 750 + 2666,16 = 7166,16$$

Таким чином витрати на розробку та експлуатацію програмного забезпечення в перший рік становлять:

$$З_{\text{р}} = 26315,68 + 7166,16 = 33481,84$$

4.2 Економічна ефективність розробки та впровадження програмного забезпечення

Показник наявності економічної ефективності від використання програмного забезпечення щодо оцінювання розміру програмного забезпечення – покращення часових витрат та трудомісткості проведення такого оцінювання.

За експертною оцінкою фахівця підприємства, впровадження програмного забезпечення дозволить вивільнити 1 працівника.

Заробітна плата одного працівника на рік складає:

$$З = 10500 * 12 * 1 = 126000 \text{ грн.}$$

Отримати річний економічний ефект можна за допомогою формули:

$$E_{\text{рік}} = \Delta C_n - E_n * k, \quad (4.3)$$

де ΔE_n – кошти, що буде вивільнено при впровадженні програмного забезпечення (126000) без експлуатаційних витрат (33481,84), тобто, 92518,16 грн.;

E_n – коефіцієнт ефективності (такий же як й коефіцієнт амортизації – 0,25)

k – одноразові витрати на впровадження програмного забезпечення (26251,68 грн.).

$$E_{\text{рік}} = 126000 - 0,25 * 26351,68 = 119412,08$$

Термін, за який програмне забезпечення окупиться наступний:

$$T = \frac{k}{\Delta C_n} = \frac{26351,68}{92518,16} = 0,28 \text{ року}$$

Отже термін, за який програмне забезпечення окупиться – приблизно 4 місяці.

4.3 Висновки до розділу 4

В розділі було виконано розрахунок витрат, економічної ефективності та терміну, за який програмне забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, окупиться. За результатами розрахунків було визначено, що це займе 4 місяці. Отже, розробка програмного забезпечення є доцільною з економічної точки зору.

5 ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Роботодавець – власник підприємства, установи, організації або уповноважений ним орган, незалежно від форм власності, виду діяльності, господарювання, і фізична особа, яка використовує найману працю.

Працівник – особа, яка працює на підприємстві, в організації, установі та виконує обов'язки або функції згідно з трудовим договором (контрактом).

Метою охорони праці є забезпечення безпечних умов праці, тобто, таких умов, за яких є відсутнім виникнення травматизму на робочому місці, захворювань, аварійних ситуацій. Охорона праці здійснюється через реалізацію практичних та наукових задач. Практичні задачі – розробка заходів щодо досягнення вищеповисаних цілей. Наукові задачі – отримані нових знань, що засновані на аналізі технологічних процесів.

Основне завдання охорони праці – покращення умов праці. До завдань охорони праці також відносять:

- 1) Знаходження оптимальних співвідношень між різними факторами виробничого середовища.
- 2) Впровадження норм гранично допустимих рівнів виробничих факторів, визначення ступеня шкідливості і небезпеки праці.
- 3) Розробка та планування заходів щодо поліпшення умов праці.
- 4) Організація безпечного виконання робіт працівниками.
- 5) Впровадження технічних засобів і заходів щодо боротьби з травматизмом і профзахворюваннями.
- 6) Розробка методів оцінки соціальної та економічної ефективності, заходів з удосконалення умов і охорони праці.

Правовою основою законодавства щодо охорони праці є Конституція України, Закони України: «Про охорону праці», «Про охорону здоров'я», «Про пожежну безпеку», «Про використання ядерної енергії та радіаційний захист», «Про забезпечення санітарного та епідеміологічного благополуччя населення», а також Кодекс законів про працю України.

Основоположним законодавчим документом в галузі охорони праці є Закон України «Про охорону праці», дія якого поширюється на всі підприємства, установи і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють, а також залучені до праці на цих підприємствах.

5.1 Аналіз шкідливих та небезпечних факторів в офісі фірми

Шкідливі виробничі фактори – фактори середовища і трудового процесу, які можуть викликати професійну патологію, тимчасове або стійке зниження працездатності, підвищити частоту соматичних та інфекційних захворювань, призвести до порушення здоров'я потомства.

При роботі за комп'ютерною технікою працівник може зазнавати багатьох видів шкідливих та небезпечних виробничих факторів. Умовно їх можна поділити на фактори виробничого середовища та психофізіологічні фактори.

До факторів виробничого середовища можна віднести:

- незадовільний мікроклімат (температура, вологість, вентиляція повітря, інфрачервоне або ультрафіолетове випромінювання) в приміщенні;
- барометричний тиск;
- постійні електричні поля і випромінювання;
- небезпечне іонізуюче випромінювання;
- високий рівень промислових шумів та вібрацій (місцевий або загальний);
- недостатнє природне або технічне освітлення в робочих приміщеннях.

До психофізіологічних факторів відноситься:

- емоційна напруга (обумовлена, наприклад, надмірним навантаженням на центральну нервову систему, органи чуттів);

- динамічні й статичні перевантаження;
- вимушене положення тіла при виконанні різноманітних виробничих операцій; надмірний і тривалий тиск різних предметів на кінцівки та інші частини тіла,
- перевантаження окремих систем організму; недостатня рухова активність; надмірно швидкий темп роботи.

Робота на персональному комп'ютері супроводжується постійною і значною напругою функцій зорового аналізатора. Розлад органів зору різко збільшується при роботі понад чотири годин на день .

Нервово-емоційне напруження при роботі на ПК виникає внаслідок дефіциту часу, великого об'єму і щільності інформації, особливостей діалогового режиму спілкування людини і ПК, відповідальності за безпомилковість інформації. Тривала робота з дисплеєм, особливо в діалоговому режимі, може призвести до нервово-емоційного перенапруження, порушення сну, погіршення стану, зниження концентрації уваги та працездатності, хронічного головного болю, підвищеної збудливості нервової системи, депресії.

Підвищені статичні і динамічні навантаження у користувачів ПК призводять до скарг на болі в спині, шийному відділі хребта і руках. З усіх нездужань, обумовлених роботою на комп'ютерах, частіше зустрічаються ті, які пов'язані з використанням клавіатури. У період виконання операцій введення даних кількість дрібних стереотипних рухів кистей і пальців рук за зміну може перевищити 60 тис., що відповідно до гігієнічної класифікації праці відноситься до категорії шкідливих і небезпечних. Більшість працюючих рано чи пізно починають пред'являти скарги на болі в шії і спині. Ці нездужання накопичуються поступово і отримали назву «синдром тривалих статичних навантажень» (СТСН).

Іншою причиною виникнення СТСН може бути тривале перебування в положенні «сидячи», яке призводить до сильного перенапруження м'язів спини і ніг. Основною причиною перенапруги м'язів спини і ніг є нераціональна висота робочої поверхні столу і сидіння, відсутність опорної спинки і підлокітників, незручне розміщення монітора, клавіатури та документів, відсутність підставки для ніг.

Ще одним небезпечним фактором може стати недостатність освітлення, що приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. неправильний напрям світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть привести до нещасного випадку або профзахворювань, тому такий важливий правильний розрахунок освітленості.

Тривала і інтенсивна робота на комп'ютері може стати джерелом важких професійних та звичайних захворювань таких як:

- **Неправильна постава** – це призводить до подальшого розвитку викривлення хребта сколіозу, лордозу, кіфозу, а як результат головні болі, болі в області шиї і всього хребта, болі в області таза. Неправильно положення ніг може привести до артриту (запалення суглобів), артрозу (деформації).

- **Тунельний синдром** – найвідоміше захворювання людей, що працюють за комп'ютером, або ж синдром зап'ястного каналу. Тунельний синдром проявляється після кількох годин напруженої роботи за комп'ютером.

- **Геморой** – хвороба аналогічна варикозному розширенню вен. Причиною хвороби може бути застій крові у венах.

- **Розвиток короткозорості** – через те, що екран монітора за контрастом вище, ніж навколишні об'єкти, розвивається короткозорість.

- **Порушення фокусування** – наслідком напруженої роботи за монітором є порушення фокусування, яке може бути викликане перенапруженням очних м'язів.

- **Сухість очей** – при погляді на джерело світла око починає менше моргати через рефлекс. Виникає «обсушування» рогівки ока, яке призводить до болі в очах.

5.2 Заходи щодо зменшення впливу шкідливих факторів роботи за персональним комп'ютером

Працюючи за комп'ютером, рекомендується дотримуватися правил тривалості роботи, правильної постави, розміру шрифтів та зображень, вимог до приміщення тощо.

Принципи правильної роботи за комп'ютером:

- у робочому приміщенні (кімнаті), де встановлені комп'ютери, щодня потрібно виконувати вологе прибирання;
- приміщення, у якому знаходяться комп'ютери, потрібно провітрювати щогодини;
- після кожного часу роботи рекомендується робити десяти хвилинну перерву, яку зручно суміщати з провітрюванням. За будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин. Під час перерви не варто читати або дивитися телевізор;
- необхідно постійно слідкувати за станом екрану монітора: він має бути чистим, без плям та пилу;
- потрібно слідкувати за поставою: ноги повинні твердо стояти на підлозі чи на спеціальній підставці; стегна повинні бути розташовані під прямим кутом до тулуба, а гомілки – під прямим кутом до стегон; сидіти потрібно прямо або злегка нахилившись вперед; відстань від очей до екрану монітора – не менше 55-60 см; центр екрану має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;
- щоб попередити „синдром сухого ока”, потрібно моргати кожні 3-5 секунд;
- не використовувати звичайний телевізор замість монітору;
- у процесі роботи рекомендується періодично (приблизно раз на 20-30 хвилин) переводити погляд з екрану на найбільш віддалений предмет у кімнаті, а ще краще – на віддалений об'єкт за вікном;

- якщо з'явилося відчуття втоми, напруження, сонливості, тяжкості в очах, потрібно припинити роботу та хоча б трохи відпочити.

Рекомендовані умови для роботи за комп'ютером:

- твердий стілець; край стільця не повинен тиснути на судини під колінами;
- відстань до монітора повинна бути 50-70см;
- перерву в сидячій роботі, 15-20 хвилин кожні 1-2 години;
- правильне освітлення робочого місця. при слабкому світлі очі напружуються і болять;
- потрібно закривати очі для відпочинку. час від часу відводите очі вбік, щоб дати відпочити своєму зору.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Навколишнє середовище – система пов’язаних між собою природних та антропогенних явищ та об’єктів, вплив та використання яких дозволяють працювати, вести побут, відпочивати, тощо. Навколишнє середовище включає в себе соціальні, природні, штучно створені (фізичні, хімічні, біологічні) фактори. Важливим викликом сьогодення є необхідність зберегти природу та запобігти негативного впливу господарської діяльності людини на природне середовище, адже економічна діяльність в будь-якому прояві викликає забруднення навколишнього середовища.

Екологія – наука, що вивчає взаємовідносини між живими організмами та сферою їх перебування. Збільшення та розвиток виробництва стрімко посилює негативний вплив на навколишнє середовище, адже все більша кількість різних речовин, які часто є токсичними та не притаманними середовищу, потрапляють в нього, зумовлюючи негативні наслідки.

Джерела забруднення навколишнього середовища можуть досить легко переміщуватись на великі відстані. У атмосфері небезпечні речовини розповсюджуються повітряними течіями. Інтенсивність такого розповсюдження залежить від напрямку переміщення повітряних мас, погодних умов, тощо. Потрапивши у воду, небезпечні речовини переміщуються від одної водойми до іншої, поширюючи свій вплив на флору та фауну.

Охорону навколишнього середовища можна організувати різними засобами: економічними, правовими, науково-технічними, санітарно-гігієнічними, біологічними, тощо.

Загалом, охорона навколишнього середовища має вирішити дві основні задачі:

- Запобіжну – забезпечити чистоту екосистеми.
- Організаційну – забезпечити раціональне використання природних ресурсів.

Головною проблемою використання природних ресурсів є їх вичерпність та обмеженість, як це дуже гарно видно на прикладі корисних копалин, що накладає певні обмеження на економічний зріст. Вичерпність та обмеженість природних ресурсів викликає конкуренцію за їх видобуток, користування, тощо, так як нині природні ресурси слугують для задоволення різних потреб.

Раціональний підхід до природокористування передбачає розробку та реалізацію заходів щодо раціонального використання та відтворення природних ресурсів, відповідальну взаємодію людини та природи. Для цього необхідно:

- Виконувати оптимальний розподіл ресурсів для різних сфер господарської діяльності.
- Використовувати технології, що спрямовані на економне використання ресурсів.
- Проводити заходи для поповнення природних ресурсів.

Забезпечення людства чистим повітрям, водними ресурсами, чистими ґрунтовими покровами для можливості забезпечення продуктами харчування є важливим питанням, що пов'язане з охороною природи, та дозволить підтримувати високий рівень життя населення довгі роки.

6.1 Забруднення навколишнього середовища

В офісі можна виділити наступні джерела забруднення:

- Твердопаливні котли.
- Дизельні електростанції.
- Побутове сміття.
- Електромагнітне випромінювання.

Зазвичай у офісах знаходиться велика кількість одиниць комп'ютерної техніки, стаціонарних телефонів, серверів, оргтехніки, тощо. Вплив планового чи аварійного відключення електроенергії призводить до зупинки робочого процесу, тому існує практика щодо розміщення електростанцій на території підприємств,

зокрема дизельних електростанцій. Така станція може працювати як в штатному режимі, так і в аварійному.

Як результат роботи дизельної електростанції отримуємо продукти згоряння дизельного палива та шумове забруднення.

В холодну пору року для обігріву та опалення приміщень використовують твердопаливні котли. Для опалення використовують вугілля, мазут, дерев'яні пелети. Пелети хоч мають менше шкідливих викидів в атмосферу, ніж вугілля чи мазут, але все одно не можуть вважатись екологічно чистими.

В таблиці 6.1 наведено порівняльну характеристику згоряння різного палива.

Таблиця 6.1 – Рівні викидів забруднюючих речовин в атмосферу при спалюванні різних видів палива

Вид палива	Викиди забруднюючих речовин в атмосферне повітря без систем очищення, тонн на 1 тис. тонн нат. палива				
	CO ₂	NO ₂	SO ₂	Тверді частинки(пил неорг.)	РАЗОМ
Природний газ	1,18	3,52	0,00	0,00	4,70
Древні брикети, пелети	4,68	9,31	0,28	4,11	17,69
Деревина дров'яна	4,9	9,4	0,3	4,3	18,9
Тирса деревна	5,0	9,6	0,5	5,0	20,0
Древні відходи, обрізки	5,2	9,9	0,4	5,2	20,7
Швидкозростаюча деревина	4,8	9,5	0,0	8,4	22,7
Тріска, сучки, кора	5,6	11,4	0,8	13,4	31,3
Мазут	5,20	5,20	35,30	0,30	45,90
Брикет торф'яний	8,04	26,81	3,00	13,02	50,87
Кам'яне вугілля	9,58	63,56	9,20	65,32	147,66

Побутові відходи – це залишки речовин і предметів, які утворюються в результаті побутової та господарської діяльності людини і які не можуть бути використані на місці утворення, а їх накопичення і зберігання порушують санітарний стан навколишнього середовища.

Побутові відходи можуть бути:

- Рідкі – нечистоти, помий, стічні води
- Тверді – сміття (побутові відходи), кухонні відходи, зокрема папір, картон, скло, пластмаса, продукти харчування.

Серед основних генераторів електромагнітного випромінювання можна виділити комп'ютерну техніку. Більша частина техніки працює протягом робочого дня, тобто, порядку 8 годин, частина не вимикається взагалі (сервери, тощо). В більшості своїй, люди стикаються з випромінювання від комп'ютерних процесорів та моніторів. Також джерелом електромагнітного випромінювання є мікрохвильові печі, яка можуть бути на кухнях в офісі. Серед негативних наслідків впливу електромагнітного випромінювання на організм людини можна виділити головний біль, порушення сну, перевтому.

6.2 Розробка заходів щодо зменшення забруднення

Повністю виключити забруднення навколишнього середовища вкрай складно, тому необхідно виробити заходи щодо зменшення такого впливу. Можна виокремити наступний перелік заходів:

– Використовувати природний газ для опалення та обігріву приміщень. Такими чином вдасться зменшити вплив шкідливих викидів, адже порівняння, проведене у таблиці 6.1 показало, що для природного газу показники є найнижчими. Але слід зважати на те, що використання природного газу може не бути оптимальним з точки зору економічної ефективності.

– Раціоналізувати процес позбавлення від побутових відходів. Відправляти вторинну сировину на переробку, а деякі побутові відходи, що передбачають безпечну утилізацію шляхом спалювання, – на спалення для отримання електроенергії. Зменшити кількість пластмас, що використовуються для упаковки на користь такої тари та упаковки, що передбачає повторне використання.

– Зменшити рівень електромагнітного впливу та убезпечити себе від такого впливу. Цього можна досягти за рахунок своєчасного вимикання не потрібних для

подальшої роботи електроприладів та використання перерв під час роботи з комп'ютером та іншими електроприладами.

ВИСНОВКИ

За результатами кваліфікаційної роботи було вирішено наукове завдання щодо підвищення достовірності оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, за рахунок побудови нелінійної регресійної моделі.

В процесі роботи було досягнуто наступних результатів:

– Досліджено існуючі регресійні моделі для оцінювання розміру web-застосунків, серед яких є багатофакторні лінійні регресійні моделі, нелінійні регресійні моделі з різними наборами метрик для побудови, нелінійні регресійні моделі для web-застосунків, що використовують фреймворки мови PHP, зокрема Laravel, CakePHP та Yii. Обґрунтовано необхідність побудови регресійної моделі для фреймворку Symfony.

– Виконано пошук проектів програмного забезпечення, що є web-застосунками, та використовують фреймворк Symfony. Для пошуку проектів було використано GitHub, а для збору метрик інструмент PhpMetrics. Всього було 50 таких проектів, з яких було зібрано наступні метрики: кількість класів, середню кількість методів на кожен клас, глибину дерева наслідування.

– Виконано попередній аналіз даних на наявність викидів. Початкові дані є негаусівськими, тому для усунення викидів було застосовано метод, що заснований на нормалізуючих перетвореннях та розрахунку квадрату відстані Махаланобіса.

– Побудовано багатофакторну нелінійну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Отримана нелінійна модель має задовільні оцінки якості R^2 , $MMRE$ та $PRED(0,25)$, які дорівнюють 0,900, 0,187 та 0,756 відповідно. Це може слугувати підставою для подальшого застосування моделі.

– Побудовано лінійну багатофакторну регресійну модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, з метою її порівняння з нелінійною моделлю. Отримана модель має незадовільні

оцінки якості *MMRE* та *PRED(0,25)* у порівнянні з нелінійною регресійною моделлю.

– Визначено довірчі інтервали та інтервали передбачення лінійної та нелінійної регресійних моделей. Отримані результати показують, що для 60% даних отримано менші ширини як довірчих інтервалів, так й інтервалів передбачення, що є суттєвим.

– Виконано порівняння побудованої нелінійної регресійної моделі з моделями для фреймворків CakePHP та Yii. Порівняння моделей показує, що для отримання якісних оцінок моделей та якісної моделі в цілому, можна покладатись лише на модель побудовану під конкретний фреймворк.

– Виконано розробку проекту програмного забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Розроблена програма успішно пройшла тестування та повністю відповідає вимогам.

– Проведено розрахунок економічної ефективності від розробки та впровадження програмного забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony. Розрахунки показали доцільність застосування такої програми.

– Розроблено розділи з охорони праці та охорони навколишнього середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Arcidiacono G. Comparative research about high failure rate of IT projects. PM World Journal. February 2017. Vol. No VI, Issue II.
2. Dan S. N. Success Factors That Influence Agile Software Development Project Success. American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS). 2016. Vol. 17, No 1, P. 172–222.
3. Usage statistics of PHP for websites [Electronic resource] // Режим доступу : <https://w3techs.com/technologies/details/pl-php>. (дата звернення: 10.09.2022).
4. The Best PHP Framework for 2015: SitePoint Survey Results [Electronic resource] // Режим доступу : <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>. (дата звернення: 10.09.2022).
5. PHP developer ecosystem survey [Electronic resource] // Режим доступу : <https://www.jetbrains.com/lp/devecosystem-2021/php/>. (дата звернення: 10.09.2022).
6. H. B. K. Tan, Y. Zhao, H. Zhang. Estimating LOC for information systems from their conceptual data models. *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*. May 20-28, 2006. Shanghai, China, 2016. P. 321–330.
7. H. B. K. Tan, Y. Zhao, H. Zhang. Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*. 2009. Vol. 19, Issue 2, Article No. 4.
8. Prykhodko S. B., Prykhodko N. V., Makarova L. M. Building the Non-Linear Regression Equations on the Basis of Multivariate Normalizing Transformations. *Proceedings of the 2018 IEEE First International Conference on SYSTEM ANALYSIS & INTELLIGENT COMPUTING (SAIC)* : October 08-12, 2018. Kyiv, Ukraine, 2018. P. 48–52.
9. Prykhodko S. B., Prykhodko N. V., Makarova L. M. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis. *Proceedings of International Conference “Advanced Computer Information*

Technologies” (ACIT-2018) : CEUR Workshop Proceedings, Ceske Budejovice, CZECH REPUBLIC. Ceske Budejovice, 2019. P. 199–202.

10. Приходько С.Б., Приходько Н.В. Нелінійне регресійне рівняння для оцінювання розміру програмного забезпечення інформаційних систем з відкритим кодом на PHP. *Вісник східноукраїнського національного університету імені Володимира Даля*. 2018. № 6 (247). С. 135–139.

11. Prykhodko S. B., Prykhodko N. V., Smykodub T. G., Spinov A. V. Constructing the non-linear regression equation to estimate the software size of open source PHP-based information systems. *Проблемы информационных технологий*. 2018. № 1 (023). С. 118–125.

12. Приходько С. Б., Приходько Н. В., Фаріонова Т. А., Ворона М. В. Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php застосунків з відкритим кодом. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки*. 2020. № 4 (56). С. 124–131.

13. Vorona M. V. Comparison of the evaluation results of the size of open source php-based apps by nonlinear regression models. *German International Journal of Modern Science*. 2021. №6. P 43–47.

14. Приходько С. Б., Приходько Н. В., Ворона М. В., Бєловол І. О. Нелінійна регресійна модель для оцінювання розміру Web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*. 2021. Т. 50, № 1. С. 115–121.

15. Prykhodko S. B., Shutko I. S., Prykhodko A. S. A nonlinear regression model to estimate the size of web apps created using the cakephp framework. *Radio Electronics Computer Science Control*. 2021. Vol 4. P. 129–139.

16. Prykhodko S. B., Shutko I. S., Prykhodko A. S. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models. *WSEAS TRANSACTIONS on COMPUTERS*. 2021. Vol. 20. P. 321–328.

17. Латанська Л. О., Кольцов А. В. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ СТРОК КОДУ WEB-ЗАСТОСУНКІВ, ЩО

СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ PHP ФРЕЙМВОРКУ SYMFONY. *Інформаційні технології: моделі, алгоритми, системи* : зб. тез III всеукраїнської науково-практичної інтернет конференції, 26-28 жовтня. 2022 р. С 8–10.

18. Montgomery D.C., Peck E. A., Vining G. G. Introduction to Linear Regression Analysis : Wiley, 2021. 702 p.

19. Prykhodko S. B., Prykhodko N. V. Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*. 2018. T.40. № 6. С. 99–108.

20. Mardia K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*. 1970. Vol. 57, Issue 3. P. 519–530.

21. Frost J. Regression Analysis: An intuitive guide for using and interpreting linear models.by Example. Statistics by Jim Publishing, 2020. 355 p.

22. Prykhodko S. B., Prykhodko N. V., Makarova L. M., Pugachenko K. M. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations. *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)* : «Celebrating 25 Years of IEEE Ukraine Section». Kyiv, Ukraine, May 29-June 2. Kyiv, 2017. P. 846–849.

23. Prykhodko S. B., Prykhodko N. V., Makarova L. M. Pukhalevych A. V. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data. *Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* : Lviv-Slavske, Ukraine, February 20–24. Lviv-Slavske, 2018. P. 962–965.

24. Arkes J. Regression Analysis: A practical introduction. Taylor & Francis Group. 2019. 342 p.

25. Приходько С. Б., Макарова Л. М., Пугаченко К. С. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Обробка експериментальних даних на комп'ютері". Миколаїв : НУК, 2018. 76 с.

26. Foss T. A., Stensrud E., Kitchenham B., Myrtveit I. Simulation study of the model evaluation criterion MMRE. *IEEE Transactions on software engineering*. 2003. Vol. 11 (29). P. 985–995.

27. Prykhodko S. B., Prykhodko N. V. Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations. Mathematical Modeling and Simulation of Systems (MODS'2020). MODS 2020. Advances in Intelligent Systems and Computing. 2020. Vol. 1265. P. 166–174.

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Найменування програми: «Symfony LOC estimation».

1 Підстави для розробки

Підставою для розробки є наказ на затвердження тем кваліфікаційних робіт магістрів.

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням програми є надання користувачу можливості оцінити розмір web-застосунків, що створюються з використанням фреймворку Symfony.

2.2 Експлуатаційне призначення

Програма призначена для експлуатації на персональних комп'ютерах користувачів для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony.

3 Вимоги до програми або програмного продукту

3.1 Вимоги для функціональних характеристик

3.1.1 Вимоги до переліку виконуваних функцій

Розроблена програма має надавати можливість виконувати наступні функції:

- Оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру web-застосунків, що створюються з використанням фреймворку Symfony.
- Оцінювання інтервалу передбачення значення розміру web-застосунків, що

створюються з використанням фреймворку Symfony.

3.1.2 Вимоги для організації вхідних та вихідних даних

Введення вхідних даних відбувається за рахунок заповнення полів графічного інтерфейсу користувача.

Результати виводяться у відповідні поля графічного інтерфейсу користувача.

Вхідними даними для програми є такі метрики програмного забезпечення як кількість класів (NOC), кількість методів на кожен клас (MbC), глибина дерева наслідування (DIT). Також для виконання оцінювання необхідно вказати довірчу ймовірність.

Кількість класів web-застосунку – це ціле число, яке більше за одиницю.

Середня кількість методів на кожен клас, глибина дерева наслідування та довірча ймовірність, фактичний розмір – дійсні додатні числа. Довірча ймовірність приймає значення не більші за одиницю.

Вихідними даними є отримана оцінка розміру web-застосунку у тисячах рядків коду, оцінки довірчого інтервалу та інтервалу передбачення, які буде відображено у відповідних полях графічного інтерфейсу програми.

3.2 Вимоги до надійності

Надійне (стійке) функціонування програмного забезпечення повинно бути забезпечено виконанням сукупності організаційно-технічних заходів, виконанням валідації вхідних даних згідно вимог та відображення повідомлень про виявлення некоректних даних з закликом ввести коректні.

3.3 Умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

3.4 Вимоги до технічної та програмної сумісності

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

3.5 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм не висуваються.

3.6 Вимоги до маркування та упаковки

Вимоги до маркування та упаковки не висуваються.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4 Вимоги до програмної документації

Склад програмної документації має включати в себе:

- 1) Технічне завдання;
- 2) Програму та методику випробувань;
- 3) Інструкцію користувача;
- 4) Опис програмного забезпечення.
- 5) Текст програми

5 Техніко-економічні показники

Економічна ефективність від впровадження програмного забезпечення була розрахована у розділі 4. Виходячи з одержаних результатів, впровадження даного програмного забезпечення є доцільним, приблизний термін окупності – 4 місяці.

6 Стадії та етапи розробки програмного забезпечення

Стадії та етапи виконання розробки програмного забезпечення можна представити наступним чином:

Таблиця А.1 – Стадії та етапи розробки

Стадії розробки	Назва етапу	Термін виконання
Технічне завдання	Розробка технічного завдання	30.09.2022
	Затвердження технічного завдання	10.10.2022
Ескізний проект	Розробка ескізного проекту	25.10.2022
	Затвердження ескізного проекту	01.11.2022
Технічний проект	Розробка технічного проекту	15.11.2022
	Затвердження технічного проекту	30.11.2022
Робочий проект	Розробка програми	01.12.2022
	Розробка програмної документації	14.12.2022
	Випробування програми	20.12.2022

7 Порядок контролю та прийомки

Приймально-здавальні роботи мають проводитись під наглядом Замовника або уповноваженої ним особи.

Приймально-здавальні випробування повинні проводитись згідно документу «Програма та методика випробувань», який узгоджується між стороною розробника та стороною замовника.

Хід проведення приймально-здавальних робіт Замовник та Виконавець документують в «Протокол проведення випробувань».

На основі Протоколу проведення випробувань Виконавець та Замовник підписують «Акт прийому-здачі програми у експлуатацію».

ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

1 Об'єкт випробувань

Об'єктом дослідження є програмне забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony

2 Мета випробувань

Мета випробувань – перевірка працездатності програмного застосунку, перевірка відповідності фактичних характеристик програмного застосунку тим вимогам, які було викладено у технічному завданні.

3 Вимоги до застосунку

Під час виконання випробувань перевіряються функціональні характеристики (вимоги), що були викладені в технічному завданні.

Розроблена програма має надавати можливість виконувати наступні функції:

- Оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру web-застосунків, що створюються з використанням фреймворку Symfony.
- Оцінювання інтервалу передбачення значення розміру web-застосунків, що створюються з використанням фреймворку Symfony.

4 Вимоги до програмної документації

Програмна документація включає в себе наступні документи:

- 1) технічне завдання;
- 2) програму та методику випробувань;
- 3) інструкцію користувача;
- 4) опис програмного забезпечення.
- 5) текст програми

5 Засоби і порядок випробувань

5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувались під час випробувань:

- Lenovo IdeaPad 330-15IKBR.
- Intel(R) Core(TM) i3-7020U CPU 2.30GHz 2.30 GHz.
- Оперативна пам'ять: 8 Гбайт.
- Накопичувач даних: SSD 256 Гбайт.

5.2 Програмні засоби, що використовувались під час випробувань

Операційна система Windows 10 Pro 21H2 19044.2006

5.3 Порядок проведення випробувань

Порядок проведення випробувань – перевірка відповідності функціональних характеристик програмного застосунку, перевірка вимог до програмної документації.

6 Методи випробувань

6.1 Методика проведення перевірки вимог до програмної документації

За відсутності конкретних вимог до оформлення та подання програмної документації, розроблені програмні документи перевіряються представником замовника візуально. Уповноважена особа перевіряє відповідність розроблених програмних документів з тими, що були вказані у пункті «Склад програмної документації, що пропонується на випробування».

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» технічного завдання.

Перевірку можна вважати успішно завершеною у випадку відповідності виконуваних функцій технічному завданню.

Порядок випробувань:

- Перевірка роботи застосунку на моніторах з різною розподільною здатністю.
- Перевірка відображення шрифтів.
- Перевірка кожної форми графічного інтерфейсу на можливість введення даних.
- Перевірка кожної форми графічного інтерфейсу на введення невалідних даних.
- Перевірка можливості виконання оцінювання розміру, визначення довірчого інтервалу та інтервалу передбачення для web-застосунків, що створюються з використанням фреймворку Symfony.

7 Результати випробувань

За результатами випробувань було підтверджено відповідність функціональних можливостей висунутим вимогам до програмного забезпечення.

8 Відомості про відмови, збої та аварійні ситуації

Під час проведення випробувань відмов, збоїв та аварійних ситуацій помічено не було

9 Відомості про коригування параметрів об'єкта випробувань та технічної документації

Коригування параметрів об'єкта випробувань та технічної документації в процесі виконання випробувань не проводилось

ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА

Вступ

Інструкція користувача призначена для ознайомлення користувача з правилами та порядком виконання операцій, які забезпечуються роботою програмного забезпечення.

Ця програма призначена для оцінювання розміру та визначення інтервальних оцінок розміру у тисячах строк коду web-застосунків, що створюються з використанням фреймворку Symfony. Програма є вільним програмним забезпеченням з графічним інтерфейсом користувача з можливістю вводити дані та переглядати результати.

Загальні відомості про програму

Найменування програми: «Symfony LOC estimation».

Використані мови програмування

Програма написана на мові Python, для розробки графічного інтерфейсу користувача було використано сумісну з Python бібліотеку Qt.

Призначення програми

Програма призначена для оцінювання розміру (в тисячах рядків коду) веб-застосунків, що створюються з використанням фреймворку Symfony на основі таких метрик як кількість класів (NOC, Number of Classes), середня кількість методів на клас (MBC, Methods by Class), глибина дерева наслідування (DIT, Depth of Inheritance Tree).

Функціональні можливості програми

- Оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру web-застосунку, що створюється з використанням фреймворку Symfony.
- Оцінювання інтервалу передбачення значення розміру web-застосунку, що створюється з використанням фреймворку Symfony.

Обмеження області застосування програми

Область застосування програми обмежена галуззю розробки програмного забезпечення.

Умови, необхідні для використання програми

Спеціальні умови для використання програми відсутні

Вимоги до технічної сумісності та програмного середовища

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

Загальний вигляд стартового вікна програми

Загальний вигляд стартового вікна програми «Symfony LOC estimation» наведено на рисунку В.1.

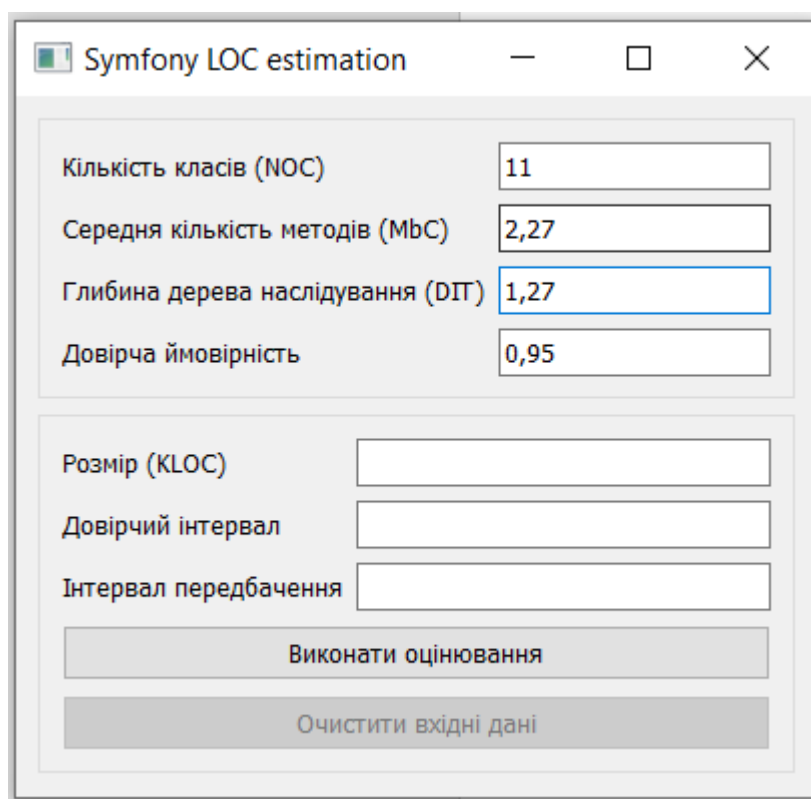
Стартове вікно містить наступний перелік полів для введення даних:

- Поле «Кількість класів (NOC)» дозволяє вказати кількість класів оцінюваного веб-застосунку. В якості кількості класів можна ввести лише невід'ємне число, більше одиниці.

- Поле «Середня кількість методів (MbC)» дозволяє вказати середню методів у класах оцінюваного веб-застосунку. В якості середньої кількості класів можна ввести дійсне невід'ємне число, більше одиниці.

- Поле «Глибина дерева наслідування (DIT)» дозволяє вказати середнє значення глибини дерева наслідування у класах оцінюваного веб-застосунку. В якості глибини дерева наслідування можна ввести дійсне невід'ємне число.

- Поле «Довірча ймовірність» для вказання довірчої ймовірності моделювання. В якості довірчої ймовірності можна ввести дійсне невід'ємне число до одиниці включно.



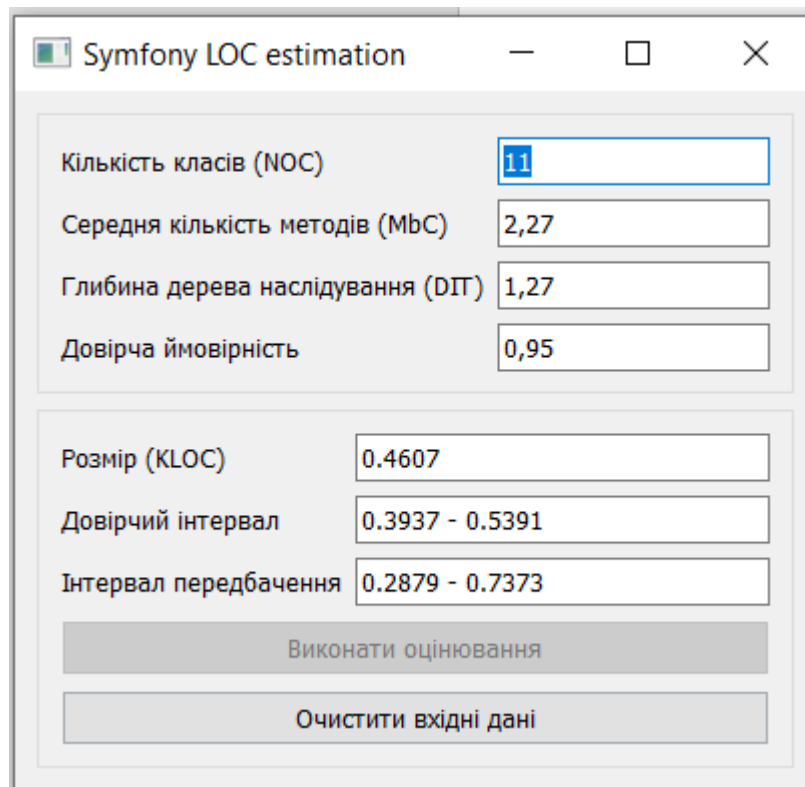
Кількість класів (NOC)	11
Середня кількість методів (MbC)	2,27
Глибина дерева наслідування (DIT)	1,27
Довірча ймовірність	0,95
Розмір (KLOC)	
Довірчий інтервал	
Інтервал передбачення	
Виконати оцінювання	
Очистити вхідні дані	

Рисунок В.1 – Вигляд вікна програми до виконання оцінювання

Результати оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Symfony

Після введення вхідних даних та ініціювання процедури виконання обчислень за допомогою відповідної кнопки, програмне забезпечення виконає розрахунок відповідних оцінок. Результати оцінювання розміру веб-застосунку (його розмір у тисячах строк коду, довірчі інтервали та інтервали прогнозування нелінійної регресії), будуть показані в полях в нижній частині екрану.

Результат виконання оцінювання можна побачити на рисунку В.2.



The screenshot shows a window titled "Symfony LOC estimation". It contains two sections of input and output fields. The first section has four input fields: "Кількість класів (NOC)" with value 11, "Середня кількість методів (MbC)" with value 2,27, "Глибина дерева наслідування (DIT)" with value 1,27, and "Довірча ймовірність" with value 0,95. The second section has three output fields: "Розмір (KLOC)" with value 0.4607, "Довірчий інтервал" with value 0.3937 - 0.5391, and "Інтервал передбачення" with value 0.2879 - 0.7373. At the bottom, there are two buttons: "Виконати оцінювання" and "Очистити вхідні дані".

Field	Value
Кількість класів (NOC)	11
Середня кількість методів (MbC)	2,27
Глибина дерева наслідування (DIT)	1,27
Довірча ймовірність	0,95
Розмір (KLOC)	0.4607
Довірчий інтервал	0.3937 - 0.5391
Інтервал передбачення	0.2879 - 0.7373

Рисунок В.2 – Результат оцінювання веб-застосунку, що створюється з використанням фреймворку Symfony

Для виконання оцінювання іншого проекту слід скористатись кнопкою для очищення попередніх вхідних даних.

ДОДАТОК Г – ТЕКСТ ПРОГРАМИ

```

import math
import sys

import numpy as np
import scipy as sp
from PySide2 import QtGui, QtWidgets

class Window(QtWidgets.QMainWindow):
    def __init__(self):
        super().__init__()
        input_group_box = QtWidgets.QGroupBox(self)
        result_group_box = QtWidgets.QGroupBox(self)

        self.class_edit = QtWidgets.QLineEdit("0", self)
        self.methods_edit = QtWidgets.QLineEdit("0", self)
        self.dit_edit = QtWidgets.QLineEdit("0", self)
        self.confidence_edit = QtWidgets.QLineEdit("0,95", self)

        self.class_edit.setValidator(QtGui.QIntValidator(self))
        self.methods_edit.setValidator(QtGui.QDoubleValidator(self))
        self.dit_edit.setValidator(QtGui.QDoubleValidator(self))
        self.confidence_edit.setValidator(
            QtGui.QDoubleValidator(0.0, 1.0, 2, parent=self))

        central_widget = QtWidgets.QWidget()
        self.setCentralWidget(central_widget)
        layout = QtWidgets.QVBoxLayout(central_widget)
        layout.addWidget(input_group_box)
        layout.addWidget(result_group_box)

        layout = QtWidgets.QFormLayout(input_group_box)
        layout.addRow("Кількість класів (NOC)", self.class_edit)
        layout.addRow("Середня кількість методів (MbC)", self.methods_edit)
        layout.addRow("Глибина дерева наслідування (DIT)", self.dit_edit)
        layout.addRow("Довірча ймовірність", self.confidence_edit)

        self.kloc = QtWidgets.QLineEdit(self)
        self.kloc.setReadOnly(True)
        self.kloc.setMinimumWidth(150)
        self.confidence_interval = QtWidgets.QLineEdit(self)
        self.confidence_interval.setReadOnly(True)
        self.confidence_interval.setMinimumWidth(150)
        self.prediciton_interval = QtWidgets.QLineEdit(self)
        self.prediciton_interval.setReadOnly(True)
        self.prediciton_interval.setMinimumWidth(150)

        layout = QtWidgets.QFormLayout(result_group_box)
        layout.addRow("Розмір (KLOC)", self.kloc)
        layout.addRow("Довірчий інтервал", self.confidence_interval)
        layout.addRow("Інтервал передбачення", self.prediciton_interval)

        self.calcButton = QtWidgets.QPushButton(self)
        self.calcButton.setText("Виконати оцінювання")
        self.calcButton.clicked.connect(self.calculate)

        self.resetButton = QtWidgets.QPushButton(self)

```

```

self.resetButton.setText("Очистити вхідні дані")
self.resetButton.clicked.connect(self.reset)

layout.addRow(self.calcButton)
layout.addRow(self.resetButton)

self.resetButton.setEnabled(False)

def calculate(self):
    vector = []
    for edit in [self.class_edit, self.methods_edit, self.dit_edit]:
        if edit.text() and edit.hasAcceptableInput():
            vector.append(float(edit.text().replace(",", ".")))
        else:
            return

    conf_level = float(self.confidence_edit.text().replace(",", "."))
    estimator = Estimator(vector, conf_level)

    size = estimator.calculateSize()
    pred_interval = estimator.calculatePred()
    conf_interval = estimator.calculateConf()

    self.kloc.setText('%0.4f' % size)
    self.predicton_interval.setText('%0.4f - %0.4f' % pred_interval)
    self.confidence_interval.setText('%0.4f - %0.4f' % conf_interval)

    self.calcButton.setEnabled(False)
    self.resetButton.setEnabled(True)

def reset(self):
    self.kloc.setText('')
    self.predicton_interval.setText('')
    self.confidence_interval.setText('')
    self.class_edit.setText('')
    self.methods_edit.setText('')
    self.dit_edit.setText('')

    self.calcButton.setEnabled(True)
    self.resetButton.setEnabled(False)

class Estimator:
    intercept = -1.63782126445733
    coeffs = np.array([0.96641578624015, 0.96301298247654, -0.46272880601894])
    N = 45
    S_Y = 0.0090855323014
    ZX = np.array([
        [0.27127997, -0.04335353, 0.25063032],
        [-0.04335353, 1.53383441, -0.71641416],
        [0.25063032, -0.71641416, 10.63646880]])
    Z_MEAN = [1.4497370, 0.5378432, 0.1425785]

    def __init__(self, vector, conf):
        self.metrics = vector
        self.confLevel = conf
        self.zx = np.log10(vector) - self.Z_MEAN
        self.se_z = np.matmul(np.matmul(self.zx.T, self.ZX), self.zx)

    def calculateSize(self):
        value = (10 ** self.intercept) * (self.metrics[0] ** self.coeffs[0]) * (
            self.metrics[1] ** self.coeffs[1]) * (self.metrics[2] **
self.coeffs[2])

```

```

        self.size = value
        return value

    def calculateConf(self):
        conf_val = (sp.stats.t.ppf(1 - (1 - self.confLevel) / 2,
                                   self.N - 4) * math.sqrt(self.S_Y * (1.0 /
self.N + self.se_z)))
        conf_interval = (10 ** (math.log10(self.size) - conf_val),
                          10 ** (math.log10(self.size) + conf_val))
        return conf_interval

    def calculatePred(self):
        pred_val = (sp.stats.t.ppf(1 - (1 - self.confLevel) / 2, self.N - 4)
                    * math.sqrt(self.S_Y * (1 + 1.0 / self.N + self.se_z)))
        pred_interval = (10 ** (math.log10(self.size) - pred_val),
                          10 ** (math.log10(self.size) + pred_val))
        return pred_interval

if __name__ == "__main__":
    app = QtWidgets.QApplication([])
    widget = Window()
    widget.setWindowTitle("Symfony LOC estimation")
    widget.resize(400, 300)
    widget.show()
    sys.exit(app.exec_())

```

ДОДАТОК Д – ОПИС ПРОГРАМИ

1 Загальні відомості

Найменування: «Symfony LOC estimation».

1.2 Програмне забезпечення, необхідне для функціонування програми

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

1.3 Мови програмування

Програмне забезпечення для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony, розроблене з використанням мови програмування Python та використанням бібліотек NumPy та SciPy для виконання математичних обчислень. Для створення графічного інтерфейсу користувача було використано сумісну з Python бібліотеку Qt.

2 Функціональне призначення

Функціональним призначенням програми є надання користувачу можливості оцінити розмір web-застосунків, що створюються з використанням фреймворку Symfony.

Функції програмного застосунку, які доступні користувачу:

- Оцінювання розміру web-застосунків, що створюються з використанням фреймворку Symfony у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру web-застосунків, що створюються з використанням фреймворку Symfony.
- Оцінювання інтервалу передбачення значення розміру web-застосунків, що створюються з використанням фреймворку Symfony.

2.1 Функціональні обмеження

Програмне забезпечення не вимагає використання зовнішніх ресурсів, тому функціональні обмеження відсутні.

3 Технічні засоби, що використовуються

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

4 Виклик та завантаження

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

Запуск програмного застосунку відбувається за рахунок натискання на ярлик програмного застосунку.

5 Вхідні дані

Вхідними даними для програми є кількість класів, середня кількість методів на кожен класів, глибина дерева наслідування, довірча ймовірність.

Кількість класів web-застосунку – це ціле число, яке більше за одиницю.

Значення середньої кількості методів, глибини дерева наслідування web-застосунку мають бути дійсними додатніми числами.

Довірча ймовірність вказує на відсотки та є дійсним числом від нуля до одиниці включно.

6 Вихідні дані

Вихідними даними є отримані оцінки розміру у тисячах рядків коду, оцінки довірчого інтервалу та інтервалу передбачення. Отримані значення буде відображено у відповідних полях графічного інтерфейсу програми.