

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

_____ проф. Приходько С.Б.

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему: Трьохфакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python, та розробка програми для її реалізації

Виконав: студент групи 6151мз

_____ Левицький І.О.

(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н, доцент.

(посада, науковий ступень вчене звання)

_____ Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2024 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)

« 14 » жовтня 2024 р.

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

Студенту Левицькому Ігорю Олександровичу _____
(Прізвище, ім'я, по батькові)

1. Тема роботи: Трьохфакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python, та розробка програми для її реалізації. _____

Керівник роботи доцент кафедри ПЗАС, к.т.н, доцент Макарова Л.М. _____

Затверджені наказом ректора № 1070-УЧ від « 11 » 10 2024 р.

2. Термін подання роботи: 09.12.2024 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; Розділ з розробки 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 Титульний слайд, Мета та завдання дослідження, Об'єкт та предмет дослідження, Наукова новизна та Практичне значення одержаних результатів, Апробація результатів досліджень та Публікації, Аналіз існуючих моделей та методів, що застосовуються для оцінювання розміру програмного забезпечення, Обґрунтування актуальності дослідження, Побудова нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python, Постановка задачі на розробку програми для оцінювання розміру веб-застосунків, що створюються мовою Python, Діаграма варіантів використання, Технічний проєкт програмного забезпечення, Результати розробки, Висновки.

6. Консультанти розділів роботи

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

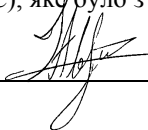
7. Дата видачі завдання 14.10.2024 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	16.10.2024	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	18.10.2024	НС*
3.	Підготовка розділу (ів) за результатами власних досліджень	21.10.2024	НС*
4.	Підготовка розділу з розробки програмного забезпечення	27.11.2024	
5.	Підготовка організаційно-економічного розділу	29.11.2024	
6.	Підготовка розділу з охорони праці	02.12.2024	
7.	Підготовка розділу з охорони навколишнього середовища	04.12.2024	
8.	Підготовка розділу ВИСНОВКИ	05.12.2024	
9.	Оформлення списку використаних джерел та додатків	06.12.2024	
10.	Подання на кафедрі 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.)	09.12.2024	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
11.	Підготовка презентації та доповіді	13.12.2024	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	16.12.2024	
13.	Подання на кафедрі 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	19.12.2024	

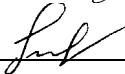
* - за результатами наукового стажування (НС), яке було з 02.09.2024 по 13.10.2024 р.

Студент



Левицький І.О.
(ПІБ)

Керівник роботи



Макарова Л.М.
(ПІБ)

РЕФЕРАТ

Левицький Ігор Олександрович

«Трьохфакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python, 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, 2024 р.

Обсяг роботи: 90 стор., 10 табл., 7 рис., 26 використаних джерел, 5 додатків.

Актуальність теми: Сьогодні багато компаній стикаються з необхідністю розробки або впровадження веб-застосунків для запуску продуктів чи оптимізації своїх бізнес-процесів. Для створення веб-застосунків на боці сервера використовуються різноманітні технології та мови програмування, зокрема і мова Python. Отже, побудова нелінійної регресійної моделі із застосуванням нормалізуючих перетворень є актуальним завданням, що сприяє підвищенню достовірності оцінювання розміру веб-застосунків, створених на Python, і має практичне значення.

Мета та завдання дослідження: підвищення достовірності оцінювання розміру веб-застосунків, що створюються мовою Python, та розробка програми для її реалізації. Завдання: проаналізувати вже існуючі моделі оцінювання розміру ПЗ, відібрати проекти веб-застосунків мовою Python з відкритим вихідним кодом, які будуть використовуватися для побудови нелінійної регресійної моделі, побудувати трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python, розробити програму, яка буде реалізовувати побудовану модель.

Об'єкт дослідження: процес оцінювання розміру веб-застосунків, що створюються мовою Python.

Предмет дослідження: нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python.

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

Наукова новизна: удосконалено трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python, за рахунок застосування нормалізуючого перетворення на основі десятичного логарифму і викидів регресії, що дозволить підвищити достовірність оцінювання розміру веб-застосунків, що створюються мовою Python, в порівнянні з існуючими моделями.

Практичне значення одержаних результатів: розробка програми для оцінювання розміру веб-застосунків, що створюються мовою Python, на основі побудованої моделі.

Апробація результатів дослідження: основні положення і результати досліджень, викладені у кваліфікаційній роботі, пройшли апробацію на VII Всеукраїнській науково-практичній інтернет-конференції молодих вчених та студентів «Сучасні інформаційні системи та технології» (м. Херсон, 29.11.2024).

Публікації: основні результати кваліфікаційної роботи викладено у 1 науковій праці – матеріалах конференції.

Ключові слова: програмне забезпечення, оцінювання розміру, нелінійна регресійна модель, веб-застосунок, Python.

ABSTRACT

Levytskyi Ihor Oleksandrovyh

«A three-factor nonlinear regression model to size estimation of web applications created in Python and software development for its implementation»

Qualification work for obtaining a master's degree in the speciality 121 «Software Engineering». Admiral Makarov National University of Shipbuilding. Mykolaiv, 2024.

Volume of work: 90 pages, 10 tables, 7 figures, 26 references, 5 appendices.

Relevance of the topic: Today, many companies are faced with the need to develop or implement web applications to launch products or optimize their business processes. Various technologies and programming languages are used to create server-side web applications, including Python. Therefore, building a nonlinear regression model using normalizing transformations is an urgent task that contributes to increasing the reliability of size estimating of web applications created in Python and is of practical importance.

Purpose and objectives of the study: high reliability of size estimating of web applications created in Python, and developing an application for its implementation. Task: to analyze existing models of software size estimation, to select projects of open source Python web applications that will be used to build a nonlinear regression model, to build a three-factor nonlinear regression model to size estimating of web applications created in Python, to develop a software that will implement the built model.

The object of the study: the process of size estimating of web applications created in Python.

Subject of the study: a nonlinear regression model to size estimating of web applications created in Python.

Methods of the study: methods of probability theory, mathematical statistics, mathematical modeling, regression analysis, object-oriented programming.

Scientific novelty: improved three-factor nonlinear regression model for size estimating of web applications created in Python by using normalizing transformation based on decimal logarithm and regression outliers, which will increase the reliability of size estimating of web applications created in Python in comparison with existing models.

Practical significance of the obtained results: development of a software for size estimating of web applications created in Python, based on the built model.

Testing of the research results: the main provisions and research results set forth in the qualification work were tested at the VII All-Ukrainian Scientific and Practical Internet Conference of Young Scientists and Students "Modern Information Systems and Technologies" (Kherson, 29.11.2024).

Publications: the main results of qualification work are set forth in 1 scientific work – conference materials.

Keywords: software, size estimation, nonlinear regression model, web application, Python.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ТА МЕТОДІВ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	14
1.1 Метрики, що застосовуються для оцінювання розміру програмного забезпечення	14
1.2 Моделі та методи, що застосовуються для оцінювання розміру програмного забезпечення	15
1.3 Обґрунтування актуальності дослідження по обраній темі	17
2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON ...	24
2.1 Методи, що застосовуються для побудови нелінійних регресійних моделей для оцінювання розміру програмного забезпечення	24
2.2 Методи, що застосовуються для передобробки даних	27
2.3 Побудова трьохфакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python	29
3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ- ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON	36
3.1 Постановка задачі на розробку програми для оцінювання розміру веб- застосунків, що створюються мовою Python	36
3.2 Розробка ескізного проекту програми для оцінювання розміру веб- застосунків, що створюються мовою Python	38
3.3 Розробка технічного проекту програми для оцінювання розміру веб- застосунків, що створюються мовою Python	43
Діаграму станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python, представлено на рисунку 3.4	43
3.4 Розробка робочого проекту програми для оцінювання розміру веб- застосунків, що створюються мовою Python	44
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON ...	48

4.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python	48
4.2 Економічна ефективність розробки та впровадження ПЗ	51
5 ОХОРОНА ПРАЦІ	54
5.1 Загальні вимоги з охорони праці при роботі на комп'ютері	54
5.2 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів	56
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	61
6.1 Вплив людини на навколишнє середовище	61
6.2 Розробка заходів щодо зменшення забруднення навколишнього середовища	64
ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON	71
ДОДАТОК Б – КОД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON ...	77
ДОДАТОК В – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON	83
ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON	86
ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МНК – метод найменших квадратів

ПЗ – програмне забезпечення

LOC – a number of lines of code

MMRE – a mean magnitude of relative error

MRE – a magnitude of relative error

PRED – percentage of prediction

SMD –squared Mahalanobis distance

WEBMO – Web Application Maintenance Model

WO – Web Objects

UML – Unified Modeling Language

ВСТУП

Актуальність теми. Існуючі на сьогодні мови програмування можна розділити на більш універсальні, які підходять для рішення різноманітних задач, наприклад, C#, Java, та на ті, які підходять для рішення більш специфічних задач, наприклад, HTML, JavaScript. Кожна мова має свою специфіку, свої переваги та недоліки. До першої групи можна віднести і мову програмування Python.

Мова програмування Python – це багатофункціональна мова програмування, яка стрімко розвивається, легко адаптується до масштабування та вирізняється зручним і логічним синтаксисом. Вона отримала широку підтримку в спільноті, має великий набір фреймворків і бібліотек для вирішення найрізноманітніших завдань. Мова Python широко використовують у багатьох провідних компаніях [1].

За даними компанії TIOBE, яка спеціалізується на оцінці та відстеженні якості програмного забезпечення, та її індексу спільноти програмування TIOBE, який є показником популярності мов програмування, мова Python посідає перше місце, стрімко увірвавшись у 2019 році в трійку лідерів. Мова програмування Python вже 5 раз ставала мовою року: у 2007, 2010, 2018, 2020 та 2021 роках, отримавши нагороду, яка присуджується мові програмування з найвищим зростанням рейтингів за один рік. Завдяки підтримці штучного інтелекту та інтелектуального аналізу даних, а також великому набору бібліотек та простоті навчання, мова програмування Python продовжує отримати своїх прихильників [2, 3].

Сьогодні багато компаній стикаються з необхідністю розробки або впровадження веб-застосунків для запуску продуктів чи оптимізації своїх бізнес-процесів. Веб-застосунок – це програмне забезпечення, яке працює у веб-

браузері. Для роботи веб-застосунку не потрібна установка на комп'ютер, тож доступ до нього можна отримати з будь-якого пристрою з веб-браузером, за умови підключення до мережі Інтернет. Найпоширеніші функції веб-сайтів, такі як кошики для покупок, пошук і фільтрація товарів, обмін миттєвими повідомленнями та стрічки новин соціальних мереж, за своєю структурою є веб-застосунками. Вони надають доступ до складних функціональних можливостей без необхідності встановлення чи налаштування програмного забезпечення [4]. Для створення веб-застосунків на боці сервера використовуються різноманітні технології та мови програмування, зокрема і мова Python.

Задача розробки ефективної системи оцінювання розміру веб-застосунків є важливою і потребує вдосконалення наявних моделей. Існують різні моделі оцінювання, побудовані для різних мов програмування та фреймворків, зокрема: створених мовою Java [5], веб-застосунків на PHP [6, 7]. Однак, оскільки ці моделі залежать від особливостей конкретної мови програмування, результати їх застосування для інших мов можуть давати недостовірні результати.

Для обраної мови програмування (Python) та типу програмних проєктів (веб-застосунки) також існує двохфакторна модель для оцінювання розміру, але вона побудована для конкретного фреймворку – Django Rest Framework [8]. До того ж, врахування додаткового фактору в моделі дозволить отримати більш якісну модель.

Отже, побудова нелінійної регресійної моделі із застосуванням нормалізуючих перетворень є актуальним завданням, що сприяє підвищенню достовірності оцінювання розміру веб-застосунків, створених на Python, і має практичне значення.

Мета і завдання дослідження. Метою роботи є підвищення достовірності оцінювання розміру веб-застосунків, що створюються мовою Python, та розробка програми для її реалізації.

Завдання, які потрібно вирішити, для досягнення мети:

- визначити метрики, необхідні для оцінювання розміру ПЗ;
- проаналізувати вже існуючі моделі оцінювання розміру ПЗ;
- обґрунтувати актуальності дослідження по обраній темі;
- проаналізувати методи, що застосовуються для побудови нелінійних регресійних моделей для оцінювання розміру ПЗ;
- проаналізувати методи, що застосовуються для передобробки даних;
- відібрати проекти веб-застосунків мовою Python з відкритим вихідним, які будуть використовуватися для побудови нелінійної регресійної моделі;
- побудувати трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python;
- перевірити якість побудованої нелінійної регресійної моделі;
- розробити програму, яка буде реалізовувати побудовану модель.

Об'єктом досліджень є процес оцінювання розміру веб-застосунків, що створюються мовою Python.

Предметом досліджень є нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python.

Методи дослідження. Для вирішення поставлених завдань були використані методи теорії ймовірностей, математичної статистики, математичного моделювання, регресійного аналізу, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python, за рахунок застосування нормалізуючого перетворення на основі десяткового логарифму і викидів регресії, що дозволить підвищити достовірність оцінювання розміру веб-застосунків, що створюються мовою Python, в порівнянні з існуючими моделями.

Практичне значення одержаних результатів полягає у розробці програми для оцінювання розміру веб-застосунків, що створюються мовою Python, на основі побудованої моделі.

Особистий внесок здобувача. Кваліфікаційна робота є самостійно виконаною працею. Усі результати, викладені у роботі, отримані автором особисто. У праці, яка опублікована у співавторстві з керівником роботи [9], здобувачеві належить побудова трьохфакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python.

Апробація результатів роботи. Основні положення і результати досліджень, викладені у кваліфікаційній роботі, пройшли апробацію на VII Всеукраїнській науково-практичній інтернет-конференції молодих вчених та студентів «Сучасні інформаційні системи та технології» (м. Херсон, 29.11.2024).

Публікації. Основні результати кваліфікаційної роботи викладено у 1 науковій праці – матеріалах конференції.

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ТА МЕТОДІВ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Метрики, що застосовуються для оцінювання розміру програмного забезпечення

Метрики ПЗ – це міра, яка дозволяє отримати числові значення деякої властивості ПЗ або його специфікацій. Використання метрик дає змогу проаналізувати складність розробки проєкту, оцінити обсяг роботи, стилістику програмування та зусилля, витрачені кожним розробником для реалізації конкретних рішень [10].

Однією з основних груп метрик є метрики розміру ПЗ. Вони зосереджені на визначенні кількісних характеристик, пов'язаних із розміром програми, і вирізняються своєю простотою. Серед найвідоміших метрик цього типу – кількість операторів програми, кількість рядків вихідного коду (LOC) та метрики Холстеда. Ці метрики орієнтовані на аналіз вихідного тексту, тому їх можна застосовувати для оцінки складності проміжних етапів розробки.

Розмірно-орієнтовані метрики забезпечують пряме вимірювання програмного продукту та процесу його створення, зазвичай базуючись на LOC-оцінках. Однак є й такі метрики, які замість розміру оцінюють функціональність або корисність продукту, що дозволяє отримати більш загальне уявлення про якість та продуктивність. Найбільше поширення в практиці створення ПЗ отримали розмірно-орієнтовані метрики.

В компаніях, які зайняті розробкою ПЗ, зазвичай фіксують та аналізують такі показники розробки програмного проєкту:

- тривалість виконання проєкту;
- загальні трудовитрати (в людино-місяцях, людино-годинах);

- витрати на розробку;
- кількість членів команди;
- обсяг вихідного коду (у тисячах рядків, KLOC);
- обсяг програмної документації;
- кількість помилок, виявлених протягом першого року експлуатації.

Ці дані слугують основою для обчислення метрик, що оцінюють продуктивність команди та якість проєкту. Проте такі метрики часто є дискусійними, особливо показник LOC, який залежить від особливостей мови програмування [11].

У сфері веб-розробки зростає популярність концептуальних моделей, оскільки вони дають змогу вносити зміни на будь-якому етапі розробки. Через швидкий розвиток технологій створення єдиної стандартизованої моделі оцінювання розміру ПЗ є складним завданням. Різні компанії застосовують різні методи оцінювання, адаптуючи їх до характеру й масштабу своїх проєктів.

1.2 Моделі та методи, що застосовуються для оцінювання розміру програмного забезпечення

Модель – це спрощене відображення реального об'єкта, системи або процесу, створене для аналізу, дослідження, опису чи імітації. Вона представляє основні характеристики або властивості об'єкта, опускаючи менш важливі деталі, щоб полегшити розуміння або роботу з ним [12].

Починаючи з кінця 1970-х років проводяться численні дослідження, присвячені розробці ПЗ та оцінюванню вартості цієї діяльності. Тема залишається актуальною і сьогодні, що підтверджується значною кількістю наукових робіт як зарубіжних [13–15], так і українських авторів [5–8].

Існує ряд методів та моделей для оцінювання розміру ПЗ, які можуть адаптуватися до різних типів проєктів, включаючи веб-застосунки.

Експертне оцінювання ґрунтується на знаннях і досвіді фахівців, які дають оцінку розміру проєкту, спираючись на свої професійні навички.

Метод нечіткої логіки використовує функції та їхній обсяг для визначення розміру ПЗ.

Нейронні мережі застосовуються для аналізу й оцінювання розміру програмного забезпечення за допомогою штучного інтелекту та технологій машинного навчання.

Метод функціональних точок оцінює кількість функцій у логічній моделі ПЗ. Розмір програми визначається кількістю базових елементарних процесів у проєкті.

Регресійні моделі передбачають використання математичних залежностей для прогнозування розміру ПЗ. За своїм видом регресійні моделі поділяються на лінійні та нелінійні в залежності від функцій, за допомогою яких вони побудовані.

Для побудови нелінійної регресійної моделі можна застосувати метод нормалізуючих перетворень, які дозволяють перевести випадкові величини з негаусівського розподілу до гаусівського. Це спрощує побудову лінійних рівнянь регресії, розрахунок довірчих інтервалів та інтервалів прогнозування для нормалізованих даних, а також забезпечує точний перехід до вихідних значень через зворотне перетворення. Метод нормалізуючих перетворень уникає проблем, які виникають у таких методах, як лінеаризація або простий перебір.

Комбіновані підходи поєднують кілька методів для підвищення точності оцінювання розміру ПЗ.

Щодо валідації таких підходів, більшість досліджень базується на аналізі даних із завершених проєктів. Крім того, більшість досліджень стосується

промислового контексту. У контексті оцінювання розміру веб-застосунків, спочатку використовувалися загальні моделі, створені для оцінювання розміру ПЗ.

Одним із перших, хто запропонував метод для веб-застосунків, був Рейфер, який розробив метрику WO та модель WEBMO [16, 17]. Метрика WO базується на підрахунку веб-об'єктів у застосунку. Під веб-об'єктами розуміються структурні елементи веб-застосунку, які впливають на складність його розробки та обслуговування: HTML-сторінки, сценарії (скрипти), динамічні елементи (форми, інтерактивні частини тощо), графіка, бази даних. Ідея метрики полягає у вимірюванні кількості та складності таких об'єктів, що дозволяє оцінити зусилля, необхідні для розробки та підтримки веб-застосунку.

Модель WEBMO призначена для оцінювання трудомісткості та витрат на обслуговування веб-застосунків. Вона враховує специфічні аспекти веб-розробки та підтримки. Метрика WO і модель WEBMO широко використовуються в дослідженнях і практиці для порівняння веб-застосунків за їхньою складністю та розміром. Ці інструменти були одними з перших спеціалізованих підходів до оцінювання веб-застосунків і заклали основу для подальших досліджень у цій сфері.

Однак наразі не існує універсальної моделі, яка була б здатна адекватно оцінити розмір веб-застосунку.

1.3 Обґрунтування актуальності дослідження по обраній темі

Для обґрунтування актуальності дослідження по обраній темі виконаємо оцінювання розміру веб-застосунків, що створюються мовою Python.

Для цього використаємо модель, яка приведена в роботі [8]. В якості вихідних даних візьмемо метрики 55 проєктів веб-застосунків, що створюються мовою Python, які знаходяться у відкритому доступі в репозитарії GitHub [18].

Для того, щоб можливо було виконати побудову моделі, потрібно, щоб співпали метрики, які будуть використовуватися для побудови моделі. Тому будемо використовувати такі метрики: Y – розмір веб-застосунку, KLOC, X_1 – кількість класів, X_2 – кількість методів.

Метрики були зібрані за допомогою розширення Statistic до PyCharm [19]. Кількість класів та кількість методів було підраховано за допомогою скрипта. Фрагмент даних наведено у таблиці 1.1.

Таблиця 1.1 – Фрагмент вихідних даних для розрахунку

№ проєкту	Y	X_1	X_2
1	6,886	41	376
2	16,858	134	569
3	143,924	1429	4547
4	1,341	17	51
5	1,697	40	105
6	13,227	250	861
7	1,879	29	106
8	1,517	33	78
9	2,679	39	165
10	0,937	29	62
11	6,771	66	315
12	10,201	36	595
13	2,077	21	104
14	38,07	473	1180
15	21,39	746	1337
16	3,15	37	177
17	121,007	1413	3039
18	5,852	242	234
19	10,355	36	291
20	15,222	141	562

Нагадаємо, двофакторне нелінійне рівняння регресії, представлене в роботі [8]:

$$\hat{Y} = 10^{-1,6796} X_1^{0,0534} X_2^{0,9571}. \quad (1.1)$$

Для перевірки якості та порівняння моделей використаємо такі параметри якості: коефіцієнт детермінації R^2 , середнє значення відносної похибки $MMRE$, Рівень прогнозування $PRED(0,25)$.

Формулу для розрахунку коефіцієнту детермінації R^2 візьмемо згідно з [20]:

$$R^2 = 1 - \left(\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \right), \quad (1.2)$$

де y_i – емпіричне значення y ,

$\hat{y}$ – розрахункове значення y ,

$\bar{y}$ – середнє значення y ,

n – кількість значень y у вибірці.

Значення $\bar{y}$ розраховується за формулою:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Формулу для розрахунку середнього значення відносної похибки $MMRE$ візьмемо згідно з [20]:

$$MMRE = \frac{1}{n} \sum_{i=1}^N MRE_i, \quad (1.3)$$

де MRE – величина відносної похибки.

Значення MRE розраховується за формулою [20]:

$$MRE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|.$$

Формулу для розрахунку рівня прогнозування $PRED(0,25)$ візьмемо згідно з [20]:

$$Pred(0,25) = \frac{k}{n}, \quad (1.4)$$

де k – кількість значень з $MRE \leq 0,25$,

n – кількість значень у вибірці.

Параметри якості рівняння, побудованого за (1.1), розраховані за формулами (1.2) – (1.4), наступні:

$$R^2=0,86, MMRE=0,26, PRED(0,25)=0,58.$$

За отриманими результатами можна зробити такий висновок: за параметром якості R^2 побудована модель відповідає необхідним умовам ($R^2 > 0,75$), за параметром якості $MMRE$ побудована модель майже відповідає необхідним вимогам ($MMRE < 0,25$), за параметром якості $PRED(0,25)$ побудована модель не відповідає необхідним вимогам ($PRED(0,25) > 0,75$).

Оскільки побудоване рівняння регресії двофакторне, потрібно пересвідчитися у відсутності мультиколінеарності незалежних змінних X_1 та X_2 .

Перевірити наявність чи відсутність мультиколінеарності можна за значеннями діагональних елементів оберненої коваріаційної матриці з k незалежних змінних, які називають коефіцієнтами впливу дисперсії (VIFs). Якщо значення VIFs знаходяться в інтервалі $[1, 5]$ – мультиколінеарність між незалежними змінними відсутня.

Коваріаційная матриця для незалежних змінних X_1 та X_2 вихідних даних наступна:

$$\begin{pmatrix} 1,0000 & 0,9318 \\ 0,9318 & 1,0000 \end{pmatrix},$$

а обернена коваріаційная матриця, відповідно, наступна:

$$\begin{pmatrix} 7,5954 & -7,0778 \\ -7,0778 & 7,5954 \end{pmatrix}.$$

Оскільки діагональні елементи оберненої коваріаційної матриці більші за 5, незалежні змінні X_1 та X_2 мають проблеми з мультиколінеарністю, а отже, використовувати приведену модель для оцінювання розміру веб-застосунків, що створюються мовою Python, недоцільно.

Щоб позбутися проблеми мультиколінеарності, замінемо метрику X_2 – кількість методів, на середню кількість методів на клас. Представимо отримані дані у вигляді таблиці 1.2 (в таблиці наведено фрагмент даних, які відповідають даним з таблиці 1.1).

Таблиця 1.2 – Фрагмент перетворених вихідних даних для розрахунку

№ проекту	Y	X_1	X_2
1	2	3	4
1	6,886	41	9,17
2	16,858	134	4,25
3	143,924	1429	3,18
4	1,341	17	3,00
5	1,697	40	2,63
6	13,227	250	3,44
7	1,879	29	3,66
8	1,517	33	2,36

Продовження таблиці 1.2

1	2	3	4
9	2,679	39	4,23
10	0,937	29	2,14
11	6,771	66	4,77
12	10,201	36	16,53
13	2,077	21	4,95
14	38,07	473	2,49
15	21,39	746	1,79
16	3,15	37	4,78
17	121,007	1413	2,15
18	5,852	242	0,97
19	10,355	36	8,08
20	15,222	141	3,99

Перевіримо перетворені дані на мультиколінеарність. Коваріаційная матриця для перетворених незалежних змінних X_1 та X_2 наступна:

$$\begin{pmatrix} 1,0000 & -0,2092 \\ -0,2092 & 1,0000 \end{pmatrix},$$

а обернена коваріаційная матриця, відповідно, наступна:

$$\begin{pmatrix} 1,0458 & 0,2188 \\ 0,2188 & 1,0458 \end{pmatrix}.$$

Як видно, діагональні елементи оберненої коваріаційної матриці знаходяться в діапазоні $[1; 5]$, отже, незалежні змінні X_1 та X_2 не мають проблеми з мультиколінеарністю. Але для використання таких незалежних змінних потрібно будувати нову нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python.

Тобто можна зробити висновок про актуальність подальшого дослідження по обраній темі та необхідність побудови нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python, з використанням нормалізуючого перетворення та врахуванням викидів у даних, що дозволить підвищити достовірність оцінювання розміру веб-застосунків, що створюються мовою Python, в порівнянні з існуючими моделями.

2 ПОБУДОВА НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

2.1 Методи, що застосовуються для побудови нелінійних регресійних моделей для оцінювання розміру програмного забезпечення

Дані, за якими проводиться оцінювання розміру програмного забезпечення, зазвичай, не є нормально розподіленими. Тому для побудови нелінійної регресійної моделі потрібно застосувати нормалізуюче перетворення, яке приведе такі дані до нормального закону розподілу.

Найпростішим для реалізації у даному випадку є логарифмічне перетворення, де кожна змінна замінюється значенням її логарифму. Оберемо логарифм з основою 10, тоді згідно з [20], нормалізуюче перетворення буде мати вигляд:

$$z = \lg(x). \quad (2.1)$$

Перетворення, зворотне до (2.1), буде мати вигляд [20]:

$$x = 10^z. \quad (2.2)$$

Тоді у разі нормально розподілених даних, отриманих за допомогою нормалізуючого перетворення (2.1), можна побудувати багатофакторне лінійне рівняння регресії, яке згідно з [20] буде мати вигляд:

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k, \quad (2.3)$$

де: $\hat{Y}$ – розрахункове значення залежної змінної;

$b_0, b_1, b_2, \dots, b_k$ – коефіцієнти лінійного рівняння регресії;

$X_1, X_2, \dots, X_k$ – значення незалежних змінних.

В такому випадку багатофакторна лінійна регресійна модель згідно з [20] буде мати вигляд:

$$Z_y = \hat{Z}_y + \varepsilon = b_0 + b_1 Z_{X_1} + b_2 Z_{X_2} + \dots + b_k Z_{X_k} + \varepsilon, \quad (2.4)$$

де ε – випадкова похибка, розподіл якої є нормальним.

Щоб знайти коефіцієнти лінійного рівняння регресії (2.3) $b_0, b_1, b_2, \dots, b_k$ можна застосувати метод найменших квадратів (МНК) [21]:

$$b_i = \frac{\sum (Z_{x_i} - \bar{Z}_{x_i})(Z_y - \bar{Z}_y)}{\sum (Z_{x_i} - \bar{Z}_{x_i})^2}, \quad (2.5)$$

$$b_0 = \bar{Z}_y - b_1 \bar{Z}_{x_1} - b_2 \bar{Z}_{x_2} - \dots - b_k \bar{Z}_{x_k}, \quad (2.6)$$

де i приймає значення від 1 до k .

Багатофакторне нелінійне рівняння регресії можна побудувати з використанням багатофакторного лінійного рівняння регресії (2.3) та зворотного нормалізуючого перетворення (2.2), яке, згідно з [20], буде мати вигляд:

$$Y = 10^{b_0} X_1^{b_1} X_2^{b_2} \cdot \dots \cdot X_k^{b_k}. \quad (2.7)$$

Тоді багатофакторну нелінійну регресійну модель можна побудувати з використанням багатофакторної лінійної регресійної моделі (2.4) та

зворотного нормалізуючого перетворення (2.2), яка, згідно з [20], буде мати вигляд:

$$Y = 10^{\varepsilon + b_0} X_1^{b_1} X_2^{b_2} \cdot \dots \cdot X_k^{b_k}. \quad (2.8)$$

Довірчий інтервал та інтервал передбачення лінійної регресії можна побудувати за наступними формулами, які будуть мати вигляд відповідно [20]:

$$\hat{Z}_y \pm t_{\alpha/2, \nu} S_{Z_y} \left\{ \frac{1}{n} + (Z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (Z_X^+) \right\}^{1/2}, \quad (2.9)$$

$$\hat{Z}_y \pm t_{\alpha/2, \nu} S_{Z_y} \left\{ 1 + \frac{1}{n} + (Z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (Z_X^+) \right\}^{1/2}, \quad (2.10)$$

де: $t_{\alpha/2, \nu}$ – квантіль t -розподілу Стюдента з рівнем значущості $\alpha/2$ та ν ступіннями вільності,

$$\nu = n - k - 1;$$

n – кількість елементів у вибірці,

$$S_{Z_y} = \left\{ \frac{1}{\nu} \sum_{i=1}^n (Z_{Y_i} - \hat{Z}_{Y_i})^2 \right\}^{1/2},$$

$(Z_X^+)^T Z_X^+$ – матриця $k \times k$:

$$(Z_X^+)^T Z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix}, \quad (2.11)$$

$$\text{де: } S_{Z_q Z_r} = \sum_{i=1}^n [Z_{qi} - \bar{Z}_q][Z_{ri} - \bar{Z}_r], q, r = 1, 2, \dots, k.$$

Довірчий інтервал та інтервал передбачення нелінійної регресії можна побудувати за допомогою формул (2.9) – (2.11) та зворотного нормалізуючого перетворення (2.2).

За допомогою описаних вище формул можливо побудувати нелінійну регресійну модель для оцінювання розміру програмного забезпечення, зокрема, і для оцінювання розміру веб-застосунків, що створюються мовою Python.

2.2 Методи, що застосовуються для передобробки даних

Оскільки дані для побудови нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python, є багатовимірними, потрібно перевірити їх на відповідність багатовимірному нормальному закону розподілу та на багатовимірні викиди.

Перевірку на відповідність багатовимірному нормальному закону розподілу можна виконати за допомогою багатовимірного ексцесу β_2 . Перевірку на багатовимірні викиди можна виконати за допомогою квадрату відстані Махаланобіса SMD.

Багатовимірний ексцес β_2 буде мати вигляд [22]:

$$\beta_2 = \frac{1}{n} \sum_{i=1}^n \{(Z_i - \bar{Z})^T S_n^{-1} (Z_i - \bar{Z})\}^2, \quad (2.12)$$

де: Z_i – i -ий компонент даних,

$\bar{Z}$ – вектор вибірових середніх,

n – кількість елементів у вибірці,

S_n – коваріаційна матриця, яка буде мати вигляд:

$$S_n = \frac{1}{n} \sum_{i=1}^n (Z_i - \bar{Z}) (Z_i - \bar{Z})^T. \quad (2.13)$$

Значення β_2 для багатовимірного нормального розподілу буде мати вигляд [22]:

$$\beta_2 = m(m+2), \quad (2.14)$$

де m – кількість параметрів у наборі даних.

Якщо значення багатовимірного ексцесу β_2 , яке розраховане за формулою (2.12), не перевищує значення, яке розраховане за формулою (2.14), багатовимірні вихідні дані відповідають багатовимірному нормальному закону розподілу.

Квадрат відстані Махаланобіса SMD буде мати вигляд [23, 24]:

$$d_i^2 = (Z_i - \bar{Z})^T S_n^{-1} (Z_i - \bar{Z}), \quad (2.15)$$

де: Z_i – i -ий компонент даних,

$\bar{Z}$ – вектор вибірових середніх,

n – кількість елементів у вибірці,

S_n – коваріаційна матриця, яка буде мати вигляд відповідно до (2.13).

Далі отримані значення квадрату відстані Махаланобіса SMD можна порівняти з критерієм Пірсона χ^2 . Дані, для яких значення квадрату відстані Махаланобіса SMD будуть більші, ніж квантиль розподілу χ^2 , є викидами. Ці значення потрібно видалити з вибірки та повторити розрахунок квадрату відстані Махаланобіса SMD, поки всі отримані значення не будуть меншими, ніж квантиль розподілу χ^2 .

У випадку одновимірних даних їх перевірку на нормальність можна виконати за допомогою критерію згоди Пірсона (критерій χ^2), який буде мати вигляд [20]:

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j}, \quad (2.16)$$

де: m – кількість підінтервалів для інтервалу $[x_{min}, x_{max}]$,

n_j – кількість значень випадкової величини, які потрапляють у j -ий підінтервал,

p_j – ймовірність того, що значення випадкової величини потрапляють у j -ий підінтервал,

n – кількість елементів у вибірці,

x_{min} та x_{max} – мінімальне та максимальне значення випадкової величини відповідні.

Кількість підінтервалів для інтервалу $[x_{min}, x_{max}]$ можна обчислити за формулою Старджеса, яка буде мати вигляд [20]:

$$m = \log_2 n + 1 = 3,3 \lg n + 1.$$

2.3 Побудова трьохфакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python

Як вже було показано в підрозділі 1.3, використання нелінійної регресійної моделі, яка приведена в роботі [8], неможливо для оцінювання розміру веб-застосунків, що створюються мовою Python, в тому числі і за

рахунок мультиколінеарності між незалежними змінними X_1 – кількістю класів та X_2 – кількістю методів.

Для усуненні проблеми з мультиколінеарністю була виконана заміна метрики X_2 на середню кількість методів на клас. Подальшу побудову нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python, було виконано по цим заміненним даним.

Перед побудовою нелінійної регресійної моделі було виконано перевірку на відповідність багатовимірному нормальному закону розподілу за допомогою багатовимірного ексцесу β_2 , розрахованого за формулами (2.12) та (2.13): $\beta_2=49,74$. В той же час значення β_2 для багатовимірного нормального розподілу згідно з формулою (2.14) повинно становити $\beta_2=15$ при $m=3$. Оскільки розрахункове значення багатовимірного ексцесу β_2 , яке розраховане за формулами (2.12) та (2.13), перевищує значення, яке розраховане за формулою (2.14), багатовимірні вихідні дані не відповідають багатовимірному нормальному закону розподілу.

Для подальшої побудови моделі потрібно виконати нормалізацію даних за формулою (2.1).

Далі потрібно виконати перевірку на багатовимірні викиди за допомогою квадрату відстані Махаланобіса SMD згідно з формулою (2.15) з використанням (2.13).

Нормалізовані дані, а також квадрат відстані Махаланобіса SMD представимо у вигляді таблиці 2.1 (в таблиці наведено фрагмент даних, які відповідають даним з таблиці 1.2).

Таблиця 2.1 – Фрагмент нормалізованих даних та SMD

№ проекту	Z_y	Z_{x1}	Z_{x2}	SMD
1	0,8380	1,6128	0,9624	3,73
2	1,2268	2,1271	0,6280	2,24
3	2,1581	3,1550	0,5027	8,33
4	0,1274	1,2304	0,4771	1,89
5	0,2297	1,6021	0,4191	1,05
6	1,1215	2,3979	0,5371	2,88
7	0,2739	1,4624	0,5629	0,65
8	0,1810	1,5185	0,3736	0,73
9	0,4280	1,5911	0,6264	1,13
10	-0,0283	1,4624	0,3300	2,26
11	0,8307	1,8195	0,6788	0,62
12	1,0086	1,5563	1,2182	8,98
13	0,3174	1,3222	0,6948	0,92
14	1,5806	2,6749	0,3970	4,51
15	1,3302	2,8727	0,2534	4,52
16	0,4983	1,5682	0,6798	0,90
17	2,0828	3,1501	0,3326	10,32
18	0,7673	2,3838	-0,0146	4,38
19	1,0152	1,5563	0,9076	5,44
20	1,1825	2,1492	0,6005	1,55

Обчислимо критерій Пірсона χ^2 для $m=3$ та $\alpha=0,005$: $\chi^2=12,84$. Отже, виходячи із аналізу отриманих даних SMD, викидів у нормалізованих даних немає.

Далі будуємо двохфакторне рівняння лінійної регресії згідно з (2.3). Коефіцієнти лінійного рівняння регресії (2.3) $b_0, b_1, b_2, \dots, b_k$ знаходимо за МНК згідно з (2.5) та (2.6): $b_0=-1,7257, b_1=1,0137, b_2=1,0480$.

Перевіримо випадкову похибку ε , чи її розподіл є нормальним за допомогою критерію згоди Пірсона χ^2 , згідно з (2.16). Розрахункове $\chi^2=2,45$. Значення квантилю χ^2 при $\alpha=0,05$: $\chi^2=9,48$. Отже, виходячи із аналізу отриманих даних, розподіл випадкової похибки ε є нормальним.

Двохфакторну лінійну регресійну модель будуємо згідно з (2.4):

$$Z_y = -1,7257 + 1,0137Z_{x_1} + 1,0480Z_{x_2} + \varepsilon.$$

Далі будуємо двохфакторне рівняння нелінійної регресії згідно з (2.7) з урахуванням зворотного нормалізуючого перетворення (2.2) та двохфакторну нелінійну регресійну модель за рівнянням (2.8) також з урахуванням зворотного нормалізуючого перетворення (2.2). Двохфакторна нелінійна регресійна модель буде мати вигляд:

$$Y = 10^{\varepsilon - 1,7257} X_1^{1,0137} X_2^{1,0480}.$$

Виконаємо побудову довірчого інтервалу та інтервалу передбачення для лінійної регресії за допомогою формул (2.9) – (2.11), і далі виконаємо побудову довірчого інтервалу та інтервалу передбачення для нелінійної регресії, використовуючи побудовані раніше інтервали для лінійної регресії та зворотне нормалізуюче перетворення (2.2).

Відомо, що у випадку, коли вихідні дані виходять за межі інтервалу передбачення, вони вважаються викидами і повинні бути видалені з вибірки, що використовується для побудови нелінійної регресійної моделі. Для вихідних даних, що розглядаються, є одна така точка даних, яку ми видаляємо із подальшої обробки.

Оскільки на першій ітерації був знайдений викид, повторюємо весь алгоритм спочатку. На другій ітерації також була знайдена одна точка даних, яка знаходиться за межами інтервалу прогнозування нелінійної регресії, яка також була видалена із подальшої обробки.

Таким чином, фінальна двохфакторна нелінійна регресійна модель була побудована з використанням вихідних даних метрик 53 проєктів веб-застосунків, що створюються мовою Python, та має такий вигляд:

$$Y = 10^{\varepsilon - 1,6705} X_1^{0,9862} X_2^{1,0345}. \quad (2.17)$$

Параметри якості рівняння, побудованого за (2.17), розраховані за формулами (1.2) – (1.4), наступні:

$$R^2=0,85, MMRE=0,22, PRED(0,25)=0,64.$$

Порівняємо ці дані з параметрами якості рівняння, побудованого за (1.1), та наведеними в підрозділі 1.1. Параметр якості R^2 залишився майже незмінним, параметр якості $MMRE$ зменшився на 0,04 та став менше за 0,25, що говорить про покращення побудованої моделі та її відповідність вимогам якості, параметр якості $PRED(0,25)$ підвищився на 0,06, однак все ще не відповідає необхідним вимогам якості ($PRED(0,25)>0,75$).

Побудуємо трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python, додавши до двохфакторної моделі третю незалежну змінну X_3 – кількість файлів у проєкті.

Додавання цього фактора у багатфакторну регресійну модель на початкових етапах розробки можна обґрунтувати наступними аргументами:

- прогноз за аналогією: на основі наявного досвіду аналогічних виконаних проєктів можна оцінити кількість файлів, навіть на ранніх етапах розробки;
- зв'язок із складністю проєкту: кількість файлів корелює з масштабом і складністю проєкту, що можна передбачити на основі вимог до програмного забезпечення;
- попереднє планування: на етапі проектування вже може бути визначена орієнтовна кількість файлів для кожного модуля.

Це дозволяє отримати більш достовірний прогноз та відслідковувати хід розробки, починаючи з ранніх етапів розробки програми.

Як і в попередньому випадку, було виконано перевірку на відповідність багатовимірному нормальному закону розподілу за допомогою багатовимірного

ексцесу β_2 , розрахованого за формулами (2.12) та (2.13): $\beta_2=80,06$. В той же час значення β_2 для багатовимірного нормального розподілу згідно з формулою (2.14) повинно становити $\beta_2=24$ при $m=4$. Оскільки розрахункове значення багатовимірного ексцесу β_2 , яке розраховане за формулами (2.12) та (2.13), перевищує значення, яке розраховане за формулою (2.14), багатовимірні вихідні дані не відповідають багатовимірному нормальному закону розподілу.

Для подальшої побудови моделі була виконана нормалізація даних за формулою (2.1). Далі була виконана перевірка на багатовимірні викиди за допомогою квадрату відстані Махаланобіса SMD згідно з формулою (2.15) з використанням (2.13). Критерій Пірсона χ^2 для $m=4$ та $\alpha=0,005$: $\chi^2=14,86$. Максимальне значення SMD= 11,26. Отже, виходячи із аналізу отриманих даних SMD, викидів у нормалізованих даних немає.

Далі було побудовано трьохфакторне рівняння лінійної регресії згідно з (2.3). Коефіцієнти лінійного рівняння регресії (2.3) $b_0, b_1, b_2, \dots, b_k$ були знайдені за МНК згідно з (2.5) та (2.6). Випадкова похибка ε була перевірена на відповідність нормальному закону розподілу за допомогою критерію згоди Пірсона χ^2 , згідно з (2.16). Трьохфакторна лінійна регресійна модель була побудована згідно з (2.4).

Далі було побудовано трьохфакторне рівняння нелінійної регресії згідно з (2.7) з урахуванням зворотного нормалізуючого перетворення (2.2) та трьохфакторна нелінійна регресійна модель за рівнянням (2.8) також з урахуванням зворотного нормалізуючого перетворення (2.2).

Також були побудовані довірчий інтервал та інтервал передбачення для лінійної регресії за допомогою формул (2.9) – (2.11) і довірчий інтервал та інтервал передбачення для нелінійної регресії, використовуючи побудовані раніше інтервали для лінійної регресії та зворотне нормалізуюче перетворення (2.2). Для вихідних даних, що розглядалися для побудови трьохфакторної

моделі, була знайдена одна точка даних, яка була видалена із подальшої обробки.

Цей алгоритм був виконаний спочатку для вихідних даних метрик 54 проєктів. На другій ітерації викидів знайдено не було, тож була побудована трьохфакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python, яка має такий вигляд:

$$Y = 10^{\varepsilon - 1,7069} X_1^{0,8034} X_2^{0,7262} X_3^{0,2880}. \quad (2.18)$$

Параметри якості рівняння, побудованого за (2.18), розраховані за формулами (1.2) – (1.4), наступні:

$$R^2=0,92, MMRE=0,19, PRED(0,25)=0,74.$$

Порівняємо ці дані з параметрами якості рівняння, побудованого за (2.17). Параметр якості R^2 підвищився на 0,07, параметр якості $MMRE$ зменшився на 0,03, параметр якості $PRED(0,25)$ підвищився на 0,1, та майже приблизився до значення, яке відповідає необхідним вимогам якості ($PRED(0,25)>0,75$).

Проаналізувавши всі отримані дані, можна зробити такі висновки.

1. Побудована трьохфакторна нелінійна регресійна модель, відповідає необхідним вимогам якості та може бути використана для оцінювання розміру веб-застосунків, що створюються мовою Python.

2. Включення в модель третього фактору покращило параметри якості моделі в порівнянні з двохфакторною нелінійною регресійною моделлю.

3. Подальше покращення якості моделі можливо за рахунок використання інших нормалізуючих перетворень.

3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

3.1 Постановка задачі на розробку програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Наступним етапом кваліфікаційної роботи є розробка проекту програми для оцінювання розміру веб-застосунків, що створюються мовою Python, використовуючи побудовану раніше трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python.

Виходячи з аналізу існуючих моделей, що застосовуються для оцінювання розміру ПЗ, було сформульовано постановку задачі на розробку програми для оцінювання розміру веб-застосунків, що створюються мовою Python. Програма, що розробляється, повинна задовольняти таким вимогам.

Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання наведених нижче функцій:

- Введення даних про параметри регресійної моделі (значення відповідних коефіцієнтів рівняння регресії b_0 , b_1 , b_2 , b_3).
- Введення довірчої ймовірності.
- Введення даних про веб-застосунок, для якого потрібно виконати розрахунок (значення метрик X_1 , X_2 , X_3).
- Оцінювання розміру веб-застосунку, що створюється мовою Python.
- Збереження результатів оцінювання розміру веб-застосунку в базу даних.

Вимоги до організації вхідних та вихідних даних

Вимоги до організації вхідних даних

Вхідні дані вносяться у відповідні поля форми згідно з допустимим типом даних (числовий).

Вхідними даними є: параметри регресійної моделі b_0 , b_1 , b_2 , b_3 , значення довірчої ймовірності, значення метрик X_1 , X_2 , X_3 .

Вимоги до організації вихідних даних

Вихідні дані виводяться у відповідні поля форм та зберігаються в базі даних.

Вихідними даними є прогнозне значення розміру веб-застосунку, що створюється мовою Python.

Вимоги до складу та параметрів технічних засобів

Для роботи з програмним продуктом необхідний наступний склад технічних засобів з мінімальними параметрами:

- CPU: Core i3;
- RAM: 1 Гб;
- HDD/SDD 500 Мб;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

Вимоги до інформаційної та програмної сумісності

Для використання даної програми потрібно мати операційну систему Windows 10 або вище.

Детальна постановка задачі сформульована в технічному завданні, яке наведене у Додатку А.

3.2 Розробка ескізного проєкту програми для оцінювання розміру веб-застосунків, що створюються мовою Python

При розробці ескізного проєкту було виявлено акторів програми для оцінювання розміру веб-застосунків, що створюються мовою Python, сформульовано варіанти використання програми, розроблено специфікації варіантів використання програми, розроблено сценарії варіантів використання та прототип користувацького інтерфейсу програми.

Програма, що розробляється, передбачає наявність одного актора – користувача програми – який буде вводити дані про параметри регресійної моделі, довірчу ймовірність, дані про веб-застосунок, для якого потрібно виконати розрахунок, оцінювання розміру веб-застосунку, що створюється мовою Python, збереження результатів оцінювання розміру веб-застосунку в базу даних.

Варіанти використання програми для оцінювання розміру веб-застосунків, що створюються мовою Python, наведені у таблиці 3.1.

Таблиця 3.1 – Варіанти використання програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Прецедент	Короткий опис
1	2
Введення даних про параметри регресійної моделі	Призначений для внесення параметрів розробленої моделі
Введення довірчої ймовірності	Призначений для внесення значення довірчої ймовірності

Продовження таблиці 3.1

1	2
Введення даних про веб-застосунок	Призначений для внесення трудомісткості розроблюваного застосунку для оцінювання його тривалості
Оцінювання розміру веб-застосунку	Призначений для оцінювання тривалості розробки в залежності від заданої трудомісткості
Збереження результатів оцінювання	Призначений для збереження в файлі результатів оцінювання

Діаграма варіантів використання програми для оцінювання розміру веб-застосунків, що створюються мовою Python, наведена на рисунку 3.1.



Рисунок 3.1 – Діаграма варіантів використання програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Сценарії варіантів використання програми для оцінювання розміру веб-застосунків, що створюються мовою Python, представлені у вигляді діаграми діяльності та зображені на рисунку 3.2.

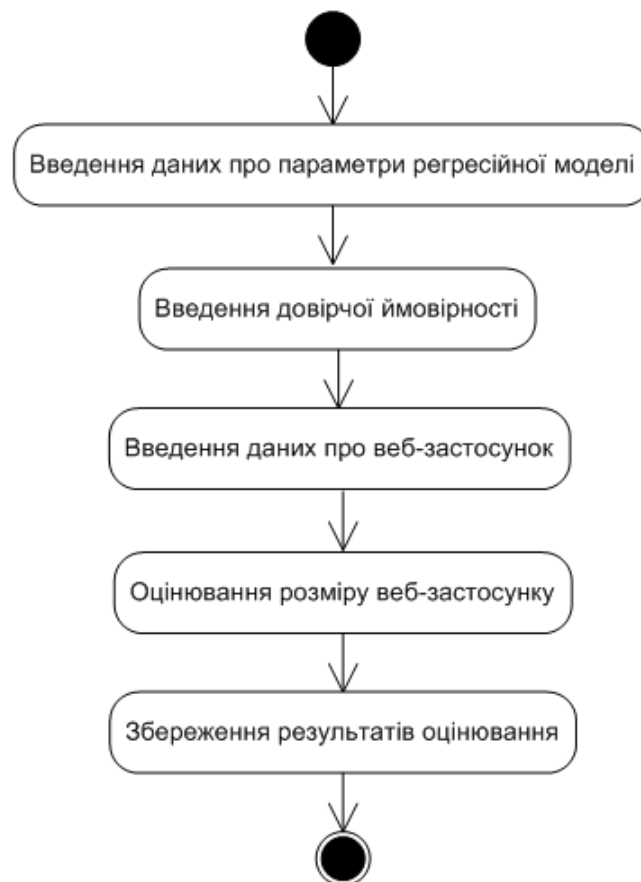


Рисунок 3.2 – Діаграма діяльності програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Зовнішня специфікація програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Функції, які виконує програма:

- Вводить у відповідні поля форми значення параметрів регресійної моделі – коефіцієнтів рівняння регресії b_0 , b_1 , b_2 , b_3 .

- Вводить у відповідне поле форми значення довірчої ймовірності, з якою потрібно виконати розрахунок.

- Вводить у відповідні поля форми значення даних про веб-застосунок, для якого потрібно виконати розрахунок – значення метрик X_1 , X_2 , X_3 .

- Виконує оцінювання розміру веб-застосунку, що створюється мовою Python, яке відповідає введенням параметрам регресійної моделі.

- Зберігає результати оцінювання розміру веб-застосунку в базу даних.

Вхідні дані

- параметри регресійної моделі b_0 , b_1 , b_2 , b_3 ,

- значення довірчої ймовірності,

- значення метрик X_1 , X_2 , X_3 .

Вихідні дані

- прогнозне значення розміру веб-застосунку, що створюється мовою Python.

Зовнішні ефекти: відкриття графічного вікна з відповідними полями для вводу даних, управління виводом даних, вихід з програми.

Діаграма станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python, зображена на рисунку 3.3.

Цифрами позначені такі стани 0 – початковий стан програми, 1 – головне вікно програми, 2 – введення параметрів регресійної моделі, 3 – введення довірчої ймовірності, 4 – введення даних про веб-застосунок, для якого потрібно виконати оцінювання, 5 – оцінювання розміру веб-застосунку, 6 – кінцевий стан програми.

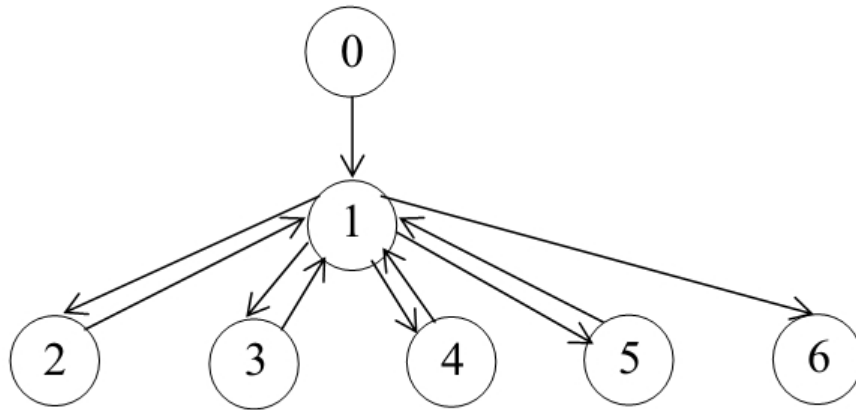


Рисунок 3.3 – Діаграма станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Далі розробимо прототип користувацького інтерфейсу програми для оцінювання розміру веб-застосунків, що створюються мовою Python, який представлено на рисунку 3.4.

Рисунок 3.4 – Прототип вікна введення даних про параметри регресійної моделі

3.3 Розробка технічного проекту програми для оцінювання розміру веб-застосунків, що створюються мовою Python

При розробці технічного проекту було розроблено діаграму станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python, та логічну модель даних ПЗ.

Діаграму станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python, представлено на рисунку 3.4.



Рисунок 3.4 – Діаграма станів програми для оцінювання розміру веб-застосунків, що створюються мовою Python

Логічна модель даних програми для оцінювання розміру веб-застосунків, що створюються мовою Python, буде містити одну таблицю з інформацією такої структури, як наведено у таблиці 3.2.

Таблиця 3.2 – Опис структури таблиці бази даних

Ім'я поля	Тип	Ключ
id	int	PK
data	timestamp	
name	varchar	
b0	double	
b1	double	
b2	double	
b3	double	
alfa	double	
x1	double	
x2	double	
x3	double	
size	double	

Після розробки технічного проекту можна приступати до наступного етапу проектування – розробки робочого проекту.

3.4 Розробка робочого проекту програми для оцінювання розміру веб-застосунків, що створюються мовою Python

При розробці робочого проекту було виконано вибір мови та засобів розробки, кодування, тестування та випробування програми для оцінювання розміру веб-застосунків, що створюються мовою Python, а також розробка усієї необхідної програмної документації.

Вибір засобів розробки програми

Програма була розроблена мовою програмування C# з використанням середовища розробки MS Visual Studio. Код програми наведений у додатку Б.

Тестування розробленої програми було проведено методом еквівалентного розбиття. Для перевірки було використано дані з таблиці 2.1.

При тестуванні використовувалися такі типи класів еквівалентності:

– коректні значення: значення, які є припустимими для системи згідно з її специфікацією;

– некоректні значення: значення, які повинні бути відхилені системою.

Були протестовані всі функції програми. Наведено приклад тестування функції «Введення даних про параметри регресійної моделі». Класи еквівалентності вхідних даних для функції «Введення даних про параметри регресійної моделі» представлені в таблиці 3.3.

Таблиця 3.3 – Класи еквівалентності вхідних даних для функції «Введення даних про параметри регресійної моделі»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Параметри регресійної моделі	1 Всі параметри регресійної моделі задані вірно	2 Одне чи більше даних задані невірно
		3 Відсутність одного чи декількох даних

В таблиці 3.4 наведено тести з правильних класів еквівалентності для функції «Введення даних про параметри регресійної моделі».

Таблиця 3.4 – Тести з правильних класів еквівалентності для функції «Введення даних про параметри регресійної моделі»

№ покритого класу еквівалентності	Вхідні дані	Результат
1	Всі параметри регресійної моделі задані вірно	Результат вірний

В таблиці 3.5 наведено тести з неправильних класів еквівалентності для функції «Введення даних про параметри регресійної моделі».

Таблиця 3.5 – Тести з неправильних класів еквівалентності для функції «Введення даних про параметри регресійної моделі»

№ покритого класу еквівалентності	Вхідні дані	Результат
2	Одне чи більше даних задані невірно	Програма виводить повідомлення про помилку. Результат підтверджено
3	Відсутність одного чи декількох даних	Програма виводить повідомлення про помилку. Результат підтверджено

По результатам тестування можна зробити висновок про те, що програма працює згідно вимог, сформульованих у технічному завданні.

Зовнішній вигляд основного вікна програми для оцінювання розміру веб-застосунків, що створюються мовою Python, та розрахунків, які виконані у програмі, представлено на рисунках 3.5 – 3.6.

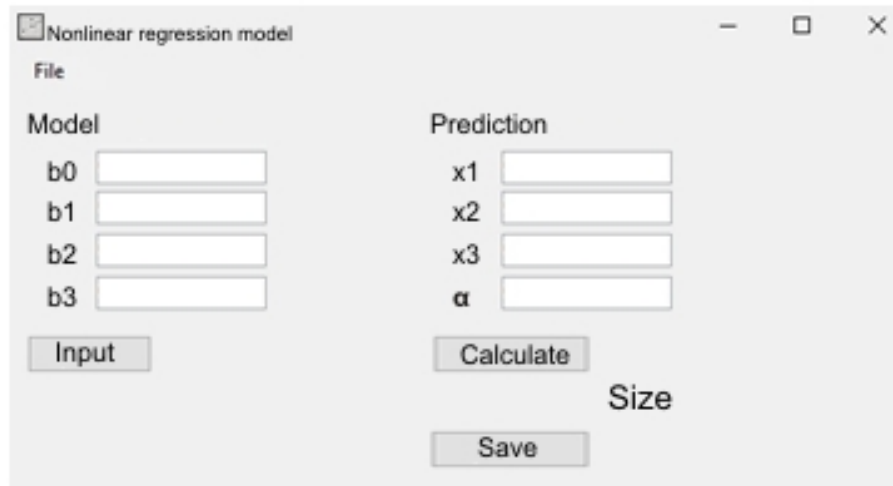


Рисунок 3.5 – Зовнішній вигляд основного вікна програми для оцінювання розміру веб-застосунків, що створюються мовою Python

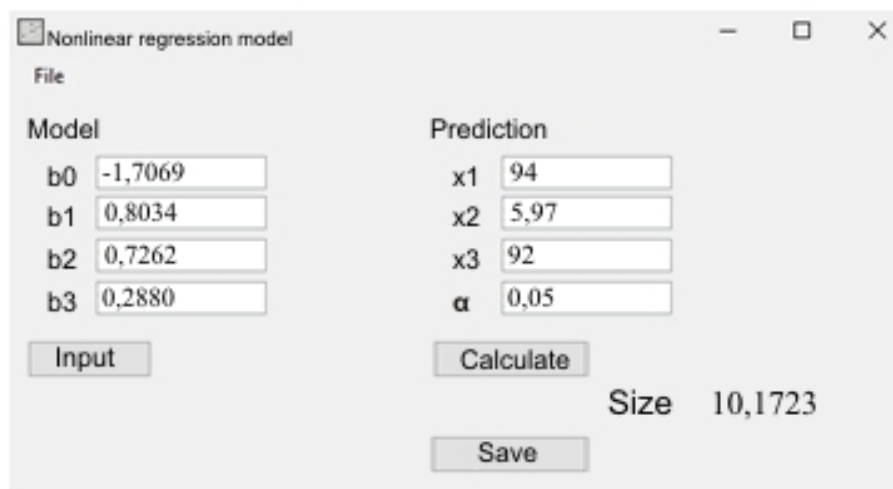


Рисунок 3.6 – Розрахунок оцінювання розміру веб-застосунків, що створюються мовою Python

Випробування програми для оцінювання розміру веб-застосунків, що створюються мовою Python, виконано у відповідності з Програмою та методикою випробувань, яка представлена у Додатку Д.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

Дана кваліфікаційна робота присвячена розробці програми для оцінювання розміру веб-застосунків, що створюються мовою Python.

Створення програмного продукту вимагає одноразових витрат на розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування ПЗ визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного продукту характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

У даному підрозділі розраховується собівартість розробки програмного продукту, а також розрахунки економічної ефективності та строку окупності.

4.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python

Витрати на розробку ПЗ складаються з витрат на заробітну платню розробника; на амортизацію комп'ютера, на якому виконується розробка; на експлуатацію комп'ютера; на засоби розробки; на матеріали і комплектуючі.

Розробка програмного продукту виконується програмістом, місячна заробітна плата якого складає 25000 грн. Вартість комп'ютера складає 40000 грн. Витрати на допоміжні матеріали наведені в таблиці 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

№	Пункти витрат	Сума, грн.
1	Доступ до мережі Internet	350
2	Папір А4	220
3	Заправка картриджа до принтера	150
4	Непередбачувані витрати	250
	Разом	970

Вартість розробки ПЗ знаходиться за формулою:

$$K_{\text{пр}} = (B_{\text{зп}} + B_{\text{есв}} + B_{\text{зг}} + B_{\text{е}}) * T + B_{\text{дм}}, \quad (4.1)$$

де: $B_{\text{зп}}$ – витрати на оплату праці, встановлені зарплатнею працівника, грн.,

$B_{\text{есв}}$ – єдиний соціальний внесок, що складає 38% від витрат на заробітну плату, грн.,

$B_{\text{зг}}$ – загальногосподарські витрати, що складають 10% від витрат на оплату праці без урахування податкового навантаження, грн.;

$B_{\text{е}}$ – витрати за електроенергію, грн.;

T – тривалість розробки, міс.;

$B_{\text{дм}}$ – витрати на допоміжні матеріали, грн.

Розрахунок вартості розробки ПЗ виконується при вартості кіловат-години електроенергії 4,32 грн. За умови споживання потужності 0,75 кВт та тривалості розробки на місяць, а саме $22 \times 8 = 176$ годин розрахуємо значення $З_e$:

$$B_e = 176 \cdot 0,75 \cdot 4,32 = 570,24 \text{ грн.}$$

Витрати на розробку ПЗ наведені в таблиці 4.2.

Таблиця 4.2 – Витрати на розробку ПЗ

№	Найменування витрат	Одиниця	Кількість
1	Тривалість розробки	міс.	1
2	Основна і додаткова заробітна плата	грн.	30000
3	Відрахування в соціальні фонди	грн.	11400
4	Загальногосподарські витрати	грн.	3000
5	Витрати на допоміжні матеріали	грн.	970
6	Витрати на електроенергію	грн.	570,24

Оскільки за методом експертної оцінки на основі розробки аналогічних систем визначено, що загальна тривалість розробки Т складає 1 місяць, а кількість необхідних спеціалістів дорівнює 1 людині, то вартість розробки системи розрахуємо за формулою (4.1) і вона складає:

$$K_{\text{пр}} = (30000 + 11400 + 3000 + 570,24) \cdot 1 + 970 = 45940,24 \text{ грн.}$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{\text{уст}} = 40000 \cdot 0,6 = 24000 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються у розмірі 5% вартості основного устаткування:

$$B_m = 40000 \cdot 0,05 = 2000 \text{ грн.}$$

Оскільки в середньому устаткування споживає 0,1 кВт електроенергії і навантажується близько 6,5 годин на день, а в році маємо 256 робочих днів, можемо розрахувати річне споживання електроенергії устаткуванням:

$$B_e = 256 \cdot 6,5 \cdot 4,32 \cdot 0,1 = 718,85 \text{ грн.}$$

Експлуатаційні витрати на устаткування на рік складуть:

$$B_{уст} = 24000 + 2000 + 718,85 = 26718,85 \text{ грн.}$$

Отже, у перший рік витрати на створення й експлуатацію програмного продукту складуть:

$$B_p = 45940,24 + 26718,85 = 72659,09 \text{ грн.}$$

4.2 Економічна ефективність розробки та впровадження ПЗ

Основним показником економічної ефективності системи є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти: високі витрати на складування паперових документів (сейфи, шафи, папки й ін.); підвищені витрати на канцтовари; витрати, пов'язані з роботою виявлення раніше допущених помилок

До числа основних факторів, що визначають приріст прибутку в зв'язку з упровадженням програмного забезпечення, відносяться: підвищення продуктивності праці; вивільнення робочого часу.

Крім того, не піддається прямій грошовій оцінці підвищення оперативності керування, якості одержуваних результатів, поліпшення організації праці і т.д. Обов'язковою умовою визначення економічної ефективності програмного забезпечення є порівнянність усіх показників у часі, за цінами й іншими нормами, використання для визначення показників, за змістом і колом елементів витрат.

Визначення прямої економічної ефективності ґрунтується на тому, що впровадження ПЗ вивільняє 0,6 працівників (за експертною оцінкою фахівців підприємства).

Зарплата 0,6 працівника в рік складає:

$$(25000,00 \cdot 12) \cdot 0,6 = 180000,00 \text{ грн.}$$

Річний економічний ефект розраховується за наступною формулою:

$$E_{\text{год}} = \Delta C_n - E_n \cdot k, \quad (4.2)$$

де ΔC_n – вивільнені кошти після впровадження ПЗ (180000,00 грн) мінус експлуатаційні витрати (26718,85) = 153281,15 грн.;

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації (0,6));

k – одноразові витрати на впровадження продукту (45940,24).

$$E_{\text{год}} = 153281,15 - 0,6 * 45940,24 = 125717,01 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = k/\Delta C_n = 45940,24/153281,15 \approx 0,3 \text{ (року)}$$

Отже, строк окупності ПЗ складає приблизно 3,5 місяців.

5 ОХОРОНА ПРАЦІ

Охорона праці є важливим кластером та соціально-економічним напрямом розвитку країни. Головне завдання – це система технічних, економічних, правових, санітарних заходів, спрямованих на забезпечення належних умов праці. Працюючи, людина взаємодіє з виробничим середовищем, як соціальним явищем, з елементами технічного та природного походження. З охороною праці тісно пов'язане питання пожежної безпеки. стан навколишнього повітряного середовища приміщень різного типу, їх цільове освітлення, вентиляційні системи, вібрація, електромагнітно-радіоактивні випромінювання, шум. Метою є дослідження трудового процесу з позиції забезпечити здоров'я та життя людини, зведення до мінімуму ймовірність нещасних випадків. Всі норми пов'язані з охороною праці закріплені в Конституції України і в Законі України «Про охорону праці» [25] та мають законодавчий характер.

5.1 Загальні вимоги з охорони праці при роботі на комп'ютері

При роботі на комп'ютері працівник зобов'язаний виконувати роботу, яка визначена його посадовою інструкцією та дотримуватись правил внутрішнього трудового розпорядку праці і відпочинку в залежності від тривалості, виду і категорії трудової діяльності. Також необхідно застосовувати засоби індивідуального або колективного захисту до вимог охорони праці в конкретній ситуації.

Необхідно навчитися методам надання першої допомоги потерпілим на робочому місці, а також знати як надати першу допомогу постраждалим від електричного струму і при інших нещасних випадках.

Окремо пройти інструктаж по протипожежним заходам та вміти застосовувати первинні засоби пожежогасіння.

Загальні вимоги з охорони праці при роботі на комп'ютері

1. Для роботи на комп'ютері допускаються особи, що проходять спеціальний інструктаж, навчання та стажування на робочому місці, перевірку знань вимог охорони праці, дотримання норм, які мають I групу з електробезпеки.

2. Небезпечні та шкідливі фактори при експлуатації комп'ютера, які можуть діяти на людину:

- надлишок електромагнітних випромінювань;
- великий рівень накопиченої статичної електрики;
- забруднення повітря;
- фізичне перевантаження в статичному положенні;
- перенавантаження на зір;
- мала освітленість місця роботи.

3. Бажано, щоби основні елементи корпусу , клавіатура та інші блоки і пристрої мали матову поверхню, яка не створюватиме відблиски, а приміщення, де розміщуються робочі місця з комп'ютерами, повинні бути обладнані захисним заземленням відповідно до технічних вимог з експлуатації. Робочі місця з комп'ютерами повинні розміщуватися таким чином, щоб відстань від екрана одного до тилу іншого було не менше 2м, а відстань між бічними поверхнями не менше 1,2 м. Робочі столи слід розміщувати таким чином, щоб природне світло падало переважно ліворуч.

4. Вікна в приміщеннях, де використовуються комп'ютери, повинні бути обладнані регульованими елементами типу жалюзі та інші.

5. Освітлення в приміщеннях, де розташовані комп'ютери, повинне здійснюватися системою загального рівномірного освітлення. У виробничих та адміністративних приміщеннях, у разі переважної роботи з документами, слід застосовувати системи комбінованого освітлення (до загального освітлення додатково встановлюються світильники місцевого освітлення, призначені для освітлення зони розташування документів).

6. Екран монітора повинен знаходитися від очей користувача на відстані 600 – 700 мм, але не ближче 500 мм.

7. У приміщеннях, обладнаних комп'ютерами, проводиться щоденне вологе прибирання і систематичне провітрювання.

5.2 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів

Розглянемо детальніше чинники, котрі призводять до проблем із зором при роботі за комп'ютером.

Основні чинники, які призводять до виникнення проблем із зоровим апаратом:

- недостатнє освітлення робочого місця;
- високий контраст між монітором та оточуючим середовищем.
- відблиски на моніторі;
- мала дистанція між користувачем на монітором;
- мала частота кліпання повіками.

Для покращення освітлення робочого місця зазвичай використовують одну настільну лампу. Але одне джерело світла забезпечує нерівномірне освітлення робочого місця та часто створює відблиски на екрані монітору. Щоб забезпечити рівномірне освітлення також використовують два когерентні

джерела світла. Але приведений метод, дозволяє тільки частково зменшити контраст між працюючим екраном та оточуючим середовищем і є не економічним.

Принцип дії запропонованого методу досить простий і полягає в розміщенні 4 напрямлених перпендикулярно до користувача джерел світла на корпусі монітору для тилового підсвічування. В якості джерел світла були використані світлодіодні стрічки. Колір випромінювання був обраний спектрально-оптимальний – білий «теплий» (2700–3200 K). А залежно від моделі LED, мають кут розсіювання від 30 до 270 градусів, що дозволяє використовувати їх як для акцентного підсвічування, так і для загального освітлення. Варто також зауважити, що світлодіоди не несуть ніякої загрози навколишньому середовищу.

Проведено експеримент з фотоапаратом та опитування серед програмістів. За допомогою цифрового фотоапарата було створено серію фотознімків у «ручному» режимі, тобто без автоматичного налаштування яскравості, витримки та балансу білого, з тиловою підсвіткою та без неї. Дане рішення дозволяє помітно зменшити контраст, різкий перепад яскравості; завдяки використанню LED-стрічки – отримати мініатюрні габарити, низьку собівартість, високу тривалість роботи та безпечність джерел освітлення.

Вимоги безпеки під час роботи з комп'ютером

Щодня перед початком роботи оператор повинен:

- оглянути своє робоче місце;
- про виявлення ознак пошкодження обладнання інформувати свого безпосереднього керівника;
- відрегулювати освітленість на робочому місці, переконатися в відсутності відблисків на екрані комп'ютера, відсутності зустрічного світла;
- перевірити правильність підключення обладнання комп'ютера до

електромережі;

- очистити екран комп'ютера від пилу та інших забруднень;
- перевірити правильність організації робочого місця й за необхідності провести відповідні корегування.

Оператор під час роботи зобов'язаний:

- виконувати тільки ту роботу, яку йому було доручено;
- підтримувати порядок і чистоту на робочому місці;
- тримати відкритими всі вентиляційні отвори обладнання;
- коректно закрити всі активні завдання у разі припинення роботи з комп'ютером;

- негайно відключити комп'ютер від електричної мережі у разі виникнення аварійної ситуації.

У ході виконання робіт оператор комп'ютера повинен:

- витримувати відстань від очей до екрана комп'ютером в межах 60-70 см;
- дотримуватися змінного режиму праці та відпочинку, регламентованих перерв у роботі;
- для розробників програм – тривалістю 15 хвилин через кожен годину роботи;
- для інших категорій працівників – тривалістю 15 хвилин через кожні дві години роботи;
- для операторів комп'ютерного набору – тривалістю 10 хвилин, після кожної години роботи.

Під час регламентованих перерв рекомендується виконувати комплекси вправ для очей, рук, хребта, поліпшення мозкового кровообігу тощо. Про виявлення несправності обладнання або інших факторів, які створюють загрозу для життя або здоров'я працівників, необхідно негайно інформувати свого

безпосереднього керівника.

Не допускається:

- виконання ремонту та налагодження комп'ютерної техніки безпосередньо на робочому місці оператора;
- зберігання біля комп'ютера паперу, носіїв інформації, запасних блоків, деталей тощо, якщо вони не використовуються для поточної роботи;
- відключення захисних пристроїв, самочинні зміни в конструкції комп'ютера;
- використання комп'ютерів, на екранах яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані тощо;
- доторкання до задньої панелі системного блоку при включеному живленні;
- вимикання живлення під час виконання активного завдання;
- попадання вологи на поверхню системного блоку, монітора, клавіатури, дисководів, принтерів та інших пристроїв;
- приймання напоїв та їжі на робочому місці.

Після закінчення роботи з використанням необхідно дотримуватися такої послідовності вимикання обладнання:

- закрити всі активні завдання, використавши опцію "Завершення роботи" у меню "Пуск", вимкнути живлення системного блоку;
- вимкнути живлення всіх комп'ютерів;
- вимкнути блок аварійного живлення (за наявності);
- відключити комп'ютер від електромережі, при цьому забороняється тягнути штепсельну вилку за дріт.

У випадку виникнення аварійної ситуації оператор зобов'язаний:

- у всіх випадках виявлення пошкодження проводів електричного живлення, несправності заземлення та інших пошкодженнях

електрообладнання, виникненні запаху гарі, диму – негайно вимкнути електричне живлення і повідомити про аварійну ситуацію свого безпосереднього керівника й чергового електрика;

- при попаданні людини під електричну напругу негайно звільнити її від дії струму шляхом вимкнення електричного живлення, до прибуття лікаря надати потерпілому долікарську медичну допомогу;

- при будь-яких випадках порушень роботи технічного обладнання або програмного забезпечення негайно викликати представника технічної служби з питань експлуатації обчислювальної техніки;

- у випадку виникнення різі в очах, різкого погіршення зору, виникнення головного болю, больових відчуттів у пальцях та кистях рук, посилення серцебиття – негайно припинити роботу з використанням комп'ютера, повідомити про те, що сталося, свого безпосереднього керівника й звернутися до медичної установи;

- при загорянні обладнання негайно відключити його від електромережі;

- про загорання повідомити свого безпосереднього керівника, оперативного чергового, пожежну службу; ужити заходів щодо ліквідації вогню за допомогою вуглекислотного або порошкового вогнегасника.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вплив людини на навколишнє середовище

Охорона навколишнього середовища являє собою форму відносин між суспільством і природою. Вона здійснюється різними засобами: економічними, правовими, науково-технічними, санітарно-гігієнічними, біологічними та іншими [26].

В загальному випадку проблема охорони навколишнього середовища зводиться до вирішення двох завдань:

- організації раціонального природокористування;
- забезпечення чистоти природних (екологічних) систем. При здійсненні різних видів економічної діяльності суб'єкти господарювання використовують різноманітні природні ресурси: землю, воду, корисні копалини тощо. Проте ресурси ці обмежені. Обмеженість природних ресурсів була і залишається головною і дуже жорсткою умовою, що накладається на розвиток економіки і відповідно зростання суспільного добробуту.

Наслідком обмеженості природних ресурсів є конкуренція за їх застосування, тобто суперництво між альтернативними цілями використання ресурсів. Адже майже всі ресурси можуть використовуватися для задоволення найрізноманітніших потреб. Наприклад, нафта може служити сировиною для одержання палива, виробництва синтетичних волокон, пластмас, лакофарбових виробів, побутової хімії тощо. І всі ці альтернативні цілі конкурують за використання сирової нафти, обсяги якої, як відомо, обмежені.

Раціональне природокористування означає розробку та здійснення концепції і конкретних заходів щодо раціонального використання і відтворення природних ресурсів, гармонічну взаємодію суспільства і природи, людини і

навколишнього природного середовища.

Завдання організації раціонального природокористування вирішується шляхом:

- оптимального розподілу ресурсів між різними господарськими цілями;
- використання технологій, що зберігають ресурси;
- проведення заходів щодо поповнення природних ресурсів.

Іншим, не менш важливим, завданням охорони навколишнього середовища є забезпечення чистоти природних екологічних систем, тобто водного середовища, повітряного басейну, ґрунтових покривів тощо, з тим, щоб забезпечити населення екологічно чистими продуктами харчування, водою, повітрям і, в остаточному підсумку, зберегти високий рівень здоров'я населення та його активного довголіття.

Економічна діяльність у всіх її проявах здійснює забруднення навколишнього середовища. У процесі цієї діяльності забруднюються і стають дефіцитними ресурси повітря, води, територій, що здавалися нескінченними. Нині рівень забруднення досяг загрозливих розмірів, набувши по суті кризового характеру.

Основними принципами охорони навколишнього природного середовища є:

- а) пріоритетність вимог екологічної безпеки, обов'язковість дотримання екологічних стандартів, нормативів та лімітів використання природних ресурсів при здійсненні господарської, управлінської та іншої діяльності;
- б) гарантування екологічно безпечного середовища для життя і здоров'я людей;
- в) запобіжний характер заходів щодо охорони навколишнього природного середовища;
- г) екологізація матеріального виробництва на основі комплексності

рішень у питаннях охорони навколишнього природного середовища, використання та відтворення відновлюваних природних ресурсів, широкого впровадження новітніх технологій;

д) збереження просторової та видової різноманітності і цілісності природних об'єктів і комплексів;

е) науково обґрунтоване узгодження екологічних, економічних та соціальних інтересів суспільства на основі поєднання міждисциплінарних знань екологічних, соціальних, природничих і технічних наук та прогнозування стану навколишнього природного середовища;

є) обов'язковість екологічної експертизи;

ж) гласність і демократизм при прийнятті рішень, реалізація яких впливає на стан навколишнього природного середовища, формування у населення екологічного світогляду;

з) науково обґрунтоване нормування впливу господарської та іншої діяльності на навколишнє природне середовище;

и) безоплатність загального та платність спеціального використання природних ресурсів для господарської діяльності;

і) стягнення збору за забруднення навколишнього природного середовища та погіршення якості природних ресурсів, компенсація шкоди, заподіяної порушенням законодавства про охорону навколишнього природного середовища;

ї) вирішення питань охорони навколишнього природного середовища та використання природних ресурсів з урахуванням ступеня антропогенної змінності територій, сукупної дії факторів, що негативно впливають на екологічну обстановку;

й) поєднання заходів стимулювання і відповідальності у справі охорони навколишнього середовища;

к) вирішення проблем охорони навколишнього природного середовища на

основі широкого міждержавного співробітництва.

Природні ресурси України є власністю народу України, який має право на володіння, використання та розпорядження природними багатствами

6.2 Розробка заходів щодо зменшення забруднення навколишнього середовища

Використанні батарейки збирають у спеціальний контейнер, який потім здається до відповідних інстанцій, які займаються переробкою батарейок. Адже використані батарейки містять у собі низку небезпечних токсичних складових, як свинець, марганець, літій і цинк. Якщо їх викидати на звичайні звалища, а не переробляти, то токсини починають отруювати воду і ґрунт, а також те, що поблизу росте. Зрештою, це, так чи інакше, потрапляє в людський організм.

Комп'ютери та оргтехніка, як відомо, далеко не вічні – і, як тільки вони вичерпають свій фізичний ресурс, їх потрібно утилізувати. Для чого потрібна утилізація оргтехніки? Справа в тому, що офісна техніка має в своєму складі як матеріали на основі фенолформальдегіда і полівінілхлориду, так і майже всі метали з періодичної таблиці Менделєєва. В процесі роботи комп'ютерів, принтерів, сканерів та іншого обладнання дані компоненти не представляють ніякої небезпеки для здоров'я людини. Зовсім інша справа – коли відпрацьовуваний свій термін виріб відправляється на міське звалище.

Завдяки впливу вологи, метали, що містяться в електронних компонентах (миш'як, кадмій, цинк і свинець) переходять в розчинні сполуки, які представляють собою сильні отрути. Нагальною екологічною проблемою є і утилізація пластиків, які містять в собі хлорні сполуки, а також ароматичні вуглеводні.

Крім того, утилізація оргтехніки та комп'ютерів – це обов'язкова процедура для підприємств і організацій різної форми власності. Неналежне виконання цієї процедури призводить до податкової та адміністративної відповідальності.

Завдяки комплексному підходу до проблеми утилізації оргтехніки кількість не переробляються відходів зводиться до мінімуму, а цінні матеріали і компоненти, такі як чорні і кольорові метали, люмінофор, рідкісні метали вилучаються і повертаються у виробництво. Золото і срібло міститься в техніці в малих кількостях, але після переробки на заводі воно здається в Державний фонд.

За фактом виконання утилізації оргтехніки і комп'ютерів оформляється акт на списання основних засобів підприємства.

Таким чином, правильно утилізувати оргтехніку потрібно як по природоохоронним, так і з економічних міркувань.

ВИСНОВКИ

Кваліфікаційна робота була присвячена вирішенню актуального завдання – підвищенню достовірності оцінювання розміру веб-застосунків, що створюються мовою Python, та розробці програми для її реалізації.

Удосконалено трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python, за рахунок застосування нормалізуючого перетворення на основі десяткового логарифму і викидів регресії, що дозволило підвищити достовірність оцінювання розміру веб-застосунків, що створюються мовою Python, в порівнянні з існуючими моделями.

В результаті роботи було вирішено такі завдання:

- визначено метрики, необхідні для оцінювання розміру ПЗ;
- проаналізовано вже існуючі моделі оцінювання розміру ПЗ;
- обґрунтовано актуальності дослідження по обраній темі;
- проаналізовано методи, що застосовуються для побудови нелінійних регресійних моделей для оцінювання розміру ПЗ;
- проаналізовано методи, що застосовуються для передобробки даних;
- відібрано проекти веб-застосунків мовою Python з відкритим вихідним кодом, які були використані для побудови нелінійної регресійної моделі;
- побудовано трьохфакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, що створюються мовою Python;
- перевірено якість побудованої нелінійної регресійної моделі;
- розроблено програму, яка реалізувала побудовану модель.

Було проведено тестування та випробування програми згідно з методикою випробувань ПЗ. Також була розроблена уся необхідна програмна документація.

В кваліфікаційній роботі також були виконані спеціальні розділи з розрахунку економічної ефективності від розробки і впровадження програмного забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python, охорони праці, охорони навколишнього середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The official site of the Python Programming Language. URL: <https://www.python.org/> (дата звернення: 14.09.2024 р.).
2. TIOBE Index for December 2023. URL: <https://www.tiobe.com/tiobe-index/> (дата звернення: 14.09.2024 р.).
3. The Python Programming Language. URL: <https://www.tiobe.com/tiobe-index/python/> (дата звернення: 14.09.2024 р.).
4. What is a Web Application? URL: https://aws.amazon.com/what-is/web-application/?nc1=h_ls (дата звернення: 14.09.2024 р.).
5. Prykhodko N.V., Prykhodko S.B. The non-linear regression model to estimate the software size of open-source Java-based systems. *Радіоелектроніка, інформатика, управління*. Запоріжжя: ЗНТУ, 2018. № 3(46). С. 158–166.
6. Приходько С.Б., Приходько Н.В., Ворона М.В., Беловол І.О. Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*. Вінниця: ВНТУ, 2021. № 1(50). С. 115–121.
7. Приходько С.Б., Приходько Н.В., Фаріонова Т.А., Ворона М.В. Трьохфакторна нелінійна регресійна модель для оцінювання розміру РНР-застосунків із відкритим кодом. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Київ, 2020. № 1(70) Ч.1. С. 124–131.
8. Макарова Л.М., Латанська Л.О., Давлатова Д.Х., Кольцов А.В. Двофакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework. *Вчені записки Таврійського національного університету імені В.І.*

Вернадського. Серія: Технічні науки. Київ: ТНУ, 2022. Том 33 (72) № 6. С.104-108. DOI <https://doi.org/10.32782/2663-5941/2022.6/18>

9. Левицький І.О., Макарова Л.М. Трьохфакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python. *Матеріали VII Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених «Сучасні інформаційні системи та технології»* (29 листопада 2024 року, м. Херсон, м. Хмельницький). За ред. А.А. Григорової. Херсон: Книжкове видавництво ФОП Вишемирський В.С., 2024. С. 186-187.

10. Що таке метрики? Базові метрики, які повинен знати кожен. URL: <https://qagroup.com.ua/publications/what-is-metrics/> (дата звернення: 25.09.2024 р.).

11. Tan H.B.K., Zhao Y., Zhang H. Estimating LOC for information systems from their conceptual data models. *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*. Shanghai, 2006. P. 321-330.

12. Модель (значення). URL: [https://uk.wikipedia.org/wiki/Модель_\(значення\)](https://uk.wikipedia.org/wiki/Модель_(значення)) (дата звернення: 25.09.2024 р.).

13. Kitchenham B., Mendes E. Why Comparative Effort Prediction Studies may be Invalid. *International Conference on Predictor Models in Software Engineering (PROMISE)*. New York, 2009. P. 1-5.

14. Boehm B., Abts C. and Chulani S. Software development cost estimation approaches – A survey. *Annals of Software Engineering*. 2000. Vol. 10. P. 177-205.

15. Kemerer C.F. An empirical validation of software cost estimation models. *Communications of the ACM*. 1987. Vol. 30, № 5. P. 416-429.

16. Reifer D.J. Estimating Web Development Costs : There Are Differences. *CROSSTALK, The Journal of Defense Software Engineering*. 2002. P. 13-17.

17. Reifer D.J. Web development: estimating quick-to-market software. *IEEE Software*. 2000. Vol. 17, №6. P. 57-64.

18. GitHub. URL: <https://github.com> (дата звернення: 04.10.2024 р.).
19. Plugin Statistic for PyCharm. URL: <https://plugins.jetbrains.com/plugin/4509-statistic> (дата звернення: 04.10.2024 р.).
20. Приходько С.Б., Макарова Л.М., Приходько Н.В., Пухалевич А.В. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни "Емпіричні методи програмної інженерії". Миколаїв: НУК, 2023. 56 с.
21. Chatterjee S., Hadi A.S. Regression analysis by example. Fifth Edition. New Jersey: A John Wiley & sons, Inc., Publication, 2012. 403 p.
22. Mardia K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*. 57. 1970. P. 519–530. DOI: 10.1093/biomet/57.3.519
23. Aggarwal C.C. Outlier analysis. 2nd ed. New York : Springer International Publishing AG, 2017. 481 p.
24. Prykhodko S., Prykhodko N., Makarova L., Pukhalevych A. 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, 2018. P. 962-965.a
25. Про охорону праці: Закон України від 15.11.2024 № 2694-XII. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (дата звернення: 10.11.2024 р.).
26. Про охорону навколишнього природного середовища: Закон України від 15.11.2024 № 1264-XII. URL: <https://zakon.rada.gov.ua/laws/show/1264-12#Text> (дата звернення: 12.11.2024 р.).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

Вступ

Повна назва розроблюваного проекту: Програмне забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python.

Коротка назва: програма.

Галузь застосування: програма призначена для застосування в компаніях та командах розробників, що займаються створенням веб-застосунків мовою Python.

1 Підстави для розробки

Підставою для розробки є завдання на кваліфікаційну роботу, видане кафедрою ПЗАС НУК ім. адмірала Макарова.

2 Призначення розробки

2.1 Функціональне призначення програми

Дана програма призначена для оцінювання розміру веб-застосунків, що створюються мовою Python.

2.2 Експлуатаційне призначення програми

Програма може використовуватися для оцінювання розміру веб-застосунків, що створюються мовою Python в компаніях та командах розробників, що займаються розробкою відповідного ПЗ.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання наведених нижче функцій:

- Введення даних про параметри регресійної моделі (значення відповідних коефіцієнтів рівняння регресії).
- Введення довірчої ймовірності.
- Введення даних про веб-застосунок, для якого потрібно виконати розрахунок (значення метрик X_1 , X_2 , X_3).
- Оцінювання розміру веб-застосунку, що створюється мовою Python.
- Збереження результатів оцінювання розміру веб-застосунку в базу даних.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вимоги до організації вхідних даних

Вхідні дані вносяться у відповідні поля форми згідно з допустимим типом даних (числовий).

Вхідними даними є: параметри регресійної моделі b_0 , b_1 , b_2 , b_3 , значення довірчої ймовірності, значення метрик X_1 , X_2 , X_3 .

Вимоги до організації вихідних даних

Вихідні дані виводяться у відповідні поля форм та зберігаються в базі даних.

Вихідними даними є прогнозне значення розміру веб-застосунку, що створюється мовою Python.

3.2 Вимоги до надійності

Вимоги до забезпечення надійного функціонування ПЗ має бути забезпеченим виконанням сукупності організаційно-технічних заходів, перелік яких наведений нижче:

- організацією безперебійного живлення ПК або сервера;
- безперебійним каналом зв'язку.

Час відновлення після відмови

Час відновлення після відмови, викликаній проблемами з електроживленням або каналом зв'язку, не фатальним збоєм (не крахом) операційної системи, не перевищує 3 хвилин, які необхідні для перезавантаження ПК або відновлення каналу зв'язку.

3.3 Вимоги до умов експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, пропонованим до технічних засобів у частині умов їхньої експлуатації.

3.3.2 Вимоги до видів обслуговування

Програмний продукт не вимагає проведення будь-яких видів обслуговування.

3.3.3 Вимоги до чисельності та кваліфікації персоналу

З програмою працює один користувач. Необхідний рівень підготовки користувачів: кінцевий користувач повинен мати практичні навички роботи з комп'ютером та розумітися на предметній області.

3.4 Вимоги до складу та параметрів технічних засобів

Для роботи з програмним продуктом необхідний наступний склад технічних засобів з мінімальними параметрами:

- CPU: Core i3;
- RAM: 1 Гб;
- HDD/SDD 500 Мб;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

3.5 Вимоги до інформаційної та програмної сумісності

Для використання даної програми потрібно мати операційну систему Windows 10 або вище.

3.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- код програми;
- опис програми;
- інструкцію користувача;
- програму та методику випробувань ПЗ.

5 Техніко-економічні показники

Економічна ефективність впровадження ПЗ розрахована в розділі 4. Згідно з отриманими результатами впровадження даного ПЗ є доцільним.

6 Стадії та етапи розробки

Стадії та етапи розробки програми для оцінювання розміру веб-застосунків, що створюються мовою Python, представлені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи розробки	Термін виконання
1. Технічне завдання	Розробка та затвердження технічного завдання	21.09.2024
2. Ескізний проект	Розробка ескізного проекту	18.10.2024
	Затвердження ескізного проекту	21.10.2024
3. Технічний проект	Розробка технічного проекту	03.11.2024
	Затвердження технічного проекту	06.11.2024
4. Робочий проект	Розробка програми	16.11.2024
	Розробка програмної документації	26.11.2024
	Випробування програми	01.12.2024

7 Порядок контролю та приймання

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Приймання проводяться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі

знаходження помилок підчас прийому програмного виробу складається акт про знайдені помилки, який підписуються представниками замовника і розробника і затверджуються керівником організації – замовника та організації – розробника. Розробник повинен протягом не більше ніж двох тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

ДОДАТОК Б – КОД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

Form.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Regression {
    public partial class Form : System.Windows.Forms.Form
    {
        private Reader reader;
        private NonlinearRegression nonlinearRegression;
        public Form()
        {
            InitializeComponent();
        }
        private void openToolStripMenuItem_Click(object sender,
            EventArgs e)
        {
            Stream stream = null;
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.InitialDirectory = "c:\\\\";
            openFileDialog.FilterIndex = 1;
            openFileDialog.RestoreDirectory = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                nonlinearRegression = new
                    NonlinearRegression(new);
            }
        }
    }
}
```

```

        classes_numeric.Enabled = true;
        calculate_button.Enabled = true;
    }
    return;
}
}
catch (Exception err)
{
    clearPrediction();
    MessageBox.Show(err.Message);
}
}
}

private void calculate_button_Click(object sender,
EventArgs e)
{
    int classes = (int)classes_numeric.Value;
    if(classes <= 0)
    {
        MessageBox.Show("The number of x must be
greater than 0");
        clearPrediction();
        return;
    }
    lines_text.Text = getFormattedLinesOfCode(classes);
}

private string getFormattedLinesOfCode(double classes)
{
    return
getRoundedNumber(nonlinearRegression.getYCalculated(class
es));
}

private string getRoundedNumber(double number)
{
    return Math.Round(number, 2).ToString();
}

private void clearPrediction()
{
    lines_text.Text = "";
}

```

```

    }
}

```

Reader.cs

```

using System;
using System.IO;

namespace Regression
{
    class Reader {
        private string[] header;
        private double[][] data;
        private int nLines;
        private int nColumns;
        public Reader(Stream stream)
        {
            string line;
            StreamReader streamReader = new StreamReader(stream);
            line = streamReader.ReadLine();
            header = line.Split(',');
            nColumns = header.Length;
            nLines = 0;
            while ((line = streamReader.ReadLine()) != null)
            {
                if (line.Length > 0) nLines++;
            }

            data = new double[nLines][];
            streamReader.BaseStream.Seek(0, 0);
            streamReader.ReadLine();
            for (int i = 0; i < nLines; i++)
            {
                data[i] = new double[nColumns];
                line = streamReader.ReadLine();
                string[] dataPair = line.Split(',');
                for (int j = 0; j < nColumns; j++)
                    data[i][j] = double.Parse(dataPair[j]);
            }

            streamReader.Close();
        }
    }
}

```

```

    public int get_nLines()
    {
        return nLines;
    }

    public int get_nColumns()
    {
        return nColumns;
    }

    public double[][] get_Data() => data;
    public string[] get_Header()
    {
        return header;
    }
}

```

lgNormalizer.cs

```

using System;
namespace Regression
{
    static class lgNormalizer
    {
        public static double reverse(double numb)
        {
            return Math.Round(Math.Exp(numb));
        }

        public static double[][] reverseData(double[][] data)
        {
            double[][] reversedData = new double[data.Length][];
            for (int i = 0; i < data.Length; i++)
            {
                reversedData[i] = new double[2];
                reversedData[i][0] = reverse(data[i][0]);
                reversedData[i][1] = reverse(data[i][1]);
            }
            return reversedData;
        }
    }
}

```

```

    }
}

```

NonlinearRegression.cs

```

using System;
namespace Regression
{
    class NonlinearRegression : RegressionModel
    {
        public NonlinearRegression() : base(new
        Data(LgNormalizer.reverseData(_Regression.getData().getData())
        ))

        public override double[][] getRegression()
        {
            double[][] regression = new double[data.getN()][];
            for (int i = 0; i < data.getN(); i++)
            {
                regression[i] = new double[2];
                regression[i][0] = data.getX(i);
                regression[i][1] =
                getYCalculated(regression[i][0]);
            }
            return regression;
        }

        public override double getYCalculated(double x)
        {
            return Math.Pow(x1, getB1()) * Math.Pow(x2, getB2())
            * Math.Pow(x3, getB3()) * Math.Exp(getB0());
        }

        public override double getB0()
        {
            return Regression.getB0();
        }

        public override double getB1()
        {
            return Regression.getB1();
        }
    }
}

```

```
public override double getB2()
{
    return Regression.getB2();
}

public override double getB3()
{
    return Regression.getB3();
}

}
```

ДОДАТОК В – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

1 Загальні відомості

Повна назва розроблюваного проекту: Програмне забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python.

Коротка назва: програма.

Галузь застосування: програма призначена для застосування в компаніях та командах розробників, що займаються створенням веб-застосунків мовою Python.

2 Функціональне призначення

Дана програма призначена для оцінювання розміру веб-застосунків, що створюються мовою Python.

Програма повинна забезпечувати можливість виконання наведених нижче функцій:

- Введення даних про параметри регресійної моделі (значення відповідних коефіцієнтів рівняння регресії).
- Введення довірчої ймовірності.
- Введення даних про веб-застосунок, для якого потрібно виконати розрахунок (значення метрик X_1 , X_2 , X_3).
- Оцінювання розміру веб-застосунку, що створюється мовою Python.
- Збереження результатів оцінювання розміру веб-застосунку в базу даних.

3 Експлуатаційне призначення

Програма може використовуватися для оцінювання розміру веб-застосунків, що створюються мовою Python в компаніях та командах розробників, що займаються розробкою відповідного ПЗ.

4 Технічні засоби, що використовуються

Для роботи з програмним продуктом необхідний наступний склад технічних засобів з мінімальними параметрами:

- CPU: Core i3;
- RAM: 1 Гб;
- HDD/SDD 500 Мб;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

5 Програмні засоби, що використовуються

Для використання даної програми потрібно мати операційну систему Windows 10 або вище.

6 Вхідні дані

Вхідні дані вносяться у відповідні поля форми згідно з допустимим типом даних (числовий).

Вхідними даними є: параметри регресійної моделі b_0 , b_1 , b_2 , b_3 , значення довірчої ймовірності, значення метрик X_1 , X_2 , X_3 .

7 Вихідні дані

Вихідні дані виводяться у відповідні поля форм та зберігаються в базі даних.

Вихідними даними є прогнозне значення розміру веб-застосунку, що створюється мовою Python.

ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

Програма для оцінювання розміру веб-застосунків, що створюються мовою Python, встановлюється безпосередньо на комп'ютер користувача. Програма запускається або за допомогою ярлика, який знаходиться на робочому столі, або за допомогою вибору відповідного пункту меню «Пуск» – «Програми». Програма має один режим використання.

Після запуску програми з'являється головне вікно програми, яке має дві умовних частини: ліва частина вікна призначена для введення даних про параметри регресійної моделі та має заголовок "Model", права частина вікна призначена для введення даних про веб-застосунок та має заголовок "Prediction". Головне вікно програми для оцінювання розміру веб-застосунків, що створюються мовою Python, представлено на рисунку Г.1.

Nonlinear regression model

File

Model

b0

b1

b2

b3

Input

Prediction

x1

x2

x3

α

Calculate

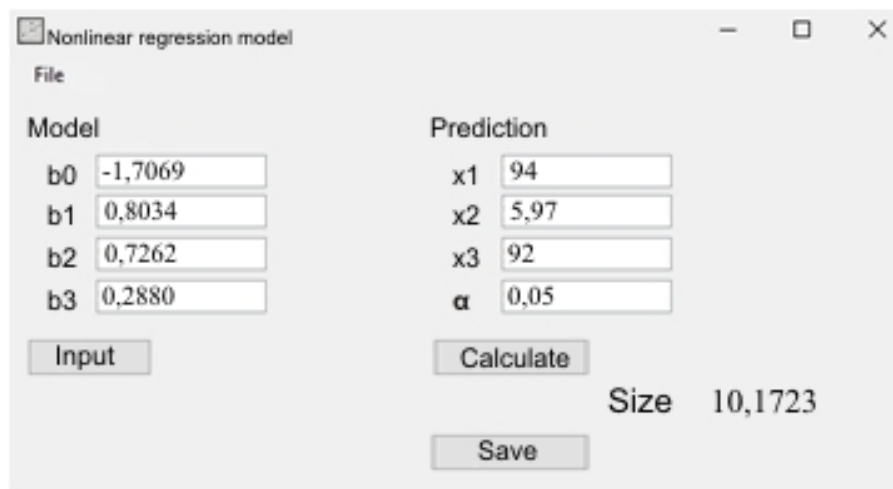
Size

Save

Рисунок Г.1 – Головне вікно програми для оцінювання розміру веб-застосунків,
що створюються мовою Python

Для початку роботи треба ввести дані про параметри регресійної моделі b_0 , b_1 , b_2 , b_3 та натиснути кнопку "Input".

Для того щоб виконати оцінювання розміру веб-застосунку, що створюються мовою Python, потрібно ввести дані про метрики веб-застосунку та значення довірчої ймовірності і натиснути кнопку "Calculate". Розраховане значення розміру веб-застосунку з'явиться у відповідному полі "Size". Зовнішній вигляд вікна з розрахунками представлено на рисунку Г.2.



The screenshot shows a window titled "Nonlinear regression model" with a menu bar containing "File". The window is divided into two main sections: "Model" and "Prediction".

Model		Prediction	
b0	-1,7069	x1	94
b1	0,8034	x2	5,97
b2	0,7262	x3	92
b3	0,2880	α	0,05

Below the input fields are three buttons: "Input", "Calculate", and "Save". The "Calculate" button is highlighted. To the right of the "Calculate" button, the text "Size 10,1723" is displayed.

Рисунок Г.2 – Зовнішній вигляд вікна з розрахунками

Для збереження результатів оцінювання розміру веб-застосунку в базу даних треба натиснути кнопку "Save".

ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ- ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON

1 Об'єкт випробовування

Об'єкт випробувань: Програмне забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python.

2 Мета випробовування

Перевірка функціональних можливостей програмного забезпечення для оцінювання розміру веб-застосунків, що створюються мовою Python, на відповідність вимогам технічного завдання.

3 Вимоги до програми

При проведенні випробувань, функціональні характеристики програмного забезпечення підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

Для проведення випробувань підлягають перевірці наступні функції:

- Введення даних про параметри регресійної моделі (значення відповідних коефіцієнтів рівняння регресії).
- Введення довірчої ймовірності.
- Введення даних про веб-застосунок, для якого потрібно виконати розрахунок (значення метрик X_1 , X_2 , X_3).
- Оцінювання розміру веб-застосунку, що створюється мовою Python.
- Збереження результатів оцінювання розміру веб-застосунку в базу даних.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- текст програми;
- опис програми;
- інструкцію користувача;
- програму та методику випробування ПЗ.

5 Засоби та порядок випробувань

Для роботи із програмою необхідний наступний склад технічних засобів з мінімальними параметрами:

- CPU: Core i3;
- RAM: 1 Гб;
- HDD/SDD 500 Мб;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

Програмні засоби, що повинні використовуватися під час випробувань

Для використання даної програми потрібно мати операційну систему Windows 10 або вище.

Випробування програмного забезпечення повинні виконуватися в наступному порядку:

- виконуються інструкції до кожної функції;
- робиться аналіз отриманих результатів і встановлюється відповідність програмного продукту вимогам і системним документам.

При виявленні помилок у роботі програмного продукту складається їх перелік і обговорюється термін їх виправлення розробником. Після цього

замовник проводить повторне тестування в повному обсязі (можливе використання нових чи додаткових тестів).

6 Методи випробувань

Випробування будемо проводити за стратегією «чорного ящика». Всі функції програми будуть проходити випробовування. В таблиці Д.1 наведено перелік випробувань, що пропонуються до виконання:

Таблиця Д.1 – Методика випробувань

№	Дія	Очікуваний результат
1	Введення даних про параметри регресійної моделі (b_0, b_1, b_2, b_3)	Відображення результату введення даних про параметри регресійної моделі
2	Введення довірчої ймовірності	Відображення результату введення довірчої ймовірності
3	Введення даних про веб-застосунок, для якого потрібно виконати розрахунок (значення метрик X_1, X_2, X_3)	Відображення результату введення даних про веб-застосунок
4	Оцінювання розміру веб-застосунку	Відображення результату оцінювання розміру веб-застосунку (прогнозне значення розміру)
5	Збереження результатів оцінювання розміру веб-застосунку в базу даних	Запис в базу даних вхідних даних та результату оцінювання