

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

_____ 2021 р.
«__» _____

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему: Математична модель для оцінювання розміру динамічних web-застосунків, що створюються на РНР, та розробка програмного забезпечення для її реалізації

Виконав: студентка 6151м групи
_____ Біла В.В.
(підпис, ПІБ)

Керівник роботи:
_____ ДОЦЕНТ, К.Т.Н., ДОЦЕНТ
(посада, науковий ступень вчене звання)

_____ Устенко І.В.
(підпис, ПІБ)

Миколаїв – 2021 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)

«__» _____ 2021 р.

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

Студенту Білій Вікторії Василівні

(Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель для оцінювання розміру динамічних web-застосунків, що створюються на PHP, та розробка програмного забезпечення для її реалізації

Керівник роботи Устенко Ірина Валеріївна, к.т.н., доцент

Затверджені наказом ректора № 1239уч від «13» _____ 10 _____ 2021 р.

2. Термін подання роботи: 06.12.2021 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

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

5. Перелік презентаційних матеріалів: 1.Актуальність теми, 2. Мета і завдання дослідження, 3.Об'єкт і предмет дослідження, 4. Методи дослідження, 5. Наукова новизна одержаних результатів, 6. Практичне значення одержаних результатів, 7. Апробація результатів досліджень та публікації, 8. Емпіричний та нормалізований розподіл вибірки X, 9. Емпіричний та нормалізований розподіл вибірки Y, 10. Нелінійна регресійна модель для оцінювання розміру динамічних web-додатків на PHP, 11. Лінійне рівняння регресії, довірчий інтервал та інтервал прогнозування, 12.Нелінійне рівняння регресії, довірчий інтервал та інтервал прогнозування, 13. Метрики якості, 14.Діаграма варіантів використання, 15. Діаграма діяльності для варіанту використання «Побудова та аналіз нелінійної регресії», 16. Концептуальна модель даних, 17. Діаграма класів, 18. Діаграма послідовності

для варіанту використання «Введення емпіричних даних», 19. Інтерфейс користувача, 20. Висновки.

6. Консультанти розділів роботи

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

7. Дата видачі завдання 13.10.2021 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	18.10.2021	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	19.10.2021	НС*
3.	Підготовка розділу (ів) з результатів власних досліджень	22.10.2021	НС*
4.	Підготовка розділу з проекту програмного забезпечення	16.11.2021	
5.	Підготовка організаційно-економічного розділу	19.11.2021	
6.	Підготовка розділу з охорони праці	22.11.2021	
7.	Підготовка розділу з охорони навколишнього середовища	24.11.2021	
8.	Підготовка розділу Висновки	25.11.2021	
9.	Оформлення списку використаних джерел та додатків	29.11.2021	
10.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	06.12.2021	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
11.	Підготовка презентації та доповіді	09.12.2021	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	10.12.2021	
13.	Подання на кафедру ПЗАС електронних версій 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	20.12.2021	

* - за результатами наукового стажування (НС), яке було з 01.09.2021 до 10.10.2021 р.

Студент

(підпис)

Біла В.В.

(ПІБ)

Керівник роботи

(підпис)

Устенко І.В.

(ПІБ)

РЕФЕРАТ

Біла Вікторія Василівна

Математична модель для оцінювання розміру динамічних web-застосунків, що створюються на РНР, та розробка програмного забезпечення для її реалізації

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2021 р.

Обсяг роботи: 109 стор., 12 таблиць, 15 рисунків, 25 використаних джерел, 5 додатків.

Актуальність теми: Отримання ефективної системи оцінювання кількості рядків коду в даний час є важливим завданням, що вимагає удосконалення існуючих методів. Адже саме ефективність оцінювання розміру програм може стати відправною точкою для успіху або невдачі проекту на ранньому етапі розробки.

Таким чином, побудова нелінійної регресійної моделі із використанням нормалізуючого перетворення для збільшення точності прогнозування кількості рядків динамічних web-додатків на РНР і створення на її основі інформаційної технології переробки інформації є актуальною.

Мета і завдання дослідження. Мета кваліфікаційної роботи полягає в підвищенні достовірності оцінювання кількості строк коду динамічних web-додатків на РНР. Для досягнення зазначеної мети поставлено наступні задачі: аналіз та порівняння наявних методів та моделей для оцінювання розміру ПЗ; обґрунтування необхідності розробки та побудова нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на РНР.

Об'єктом досліджень є процес оцінювання кількості строк коду динамічних web-додатків на РНР.

Предметом досліджень є математична модель оцінювання кількості рядків коду динамічних web-додатків на РНР.

Методи дослідження. При розв'язанні сформульованих завдань будуть використовуватися методи обробки статистичних даних, математичного моделювання і порівняння.

Наукова новизна отриманих результатів полягає в наступному: удосконалена математична модель оцінювання кількості рядків коду динамічних web-додатків на РНР за рахунок нормалізації десятковим логарифмом, що дозволяє підвищити достовірність оцінювання розміру динамічних web-додатків на РНР порівняно з лінійними моделями.

Практичне значення одержаних результатів. ПЗ для оцінювання кількості рядків коду динамічних web-додатків на РНР, яке розроблено в рамках кваліфікаційної роботи, дозволить автоматизувати та скоротити час відповідних розрахунків.

Апробація результатів досліджень: Основні положення кваліфікаційної роботи були оприлюднені на XIII Міжнародній науково-практичній конференції «Free and open source software» (FOSS-2021) (м. Харків, 16-18 листопада 2021).

Ключові слова: ОЦІНЮВАННЯ РОЗМІРУ, ДИНАМІЧНІ WEB-ДОДАТКИ, МОВА ПРОГРАМУВАННЯ РНР, МАТЕМАТИЧНА МОДЕЛЬ

ABSTRACT

BilaViktoriia

A mathematical model for estimating the size of dynamic web-applications created in PHP and developing the software for its implementation

Master's Degree in Master's Degree in Specialty 121 - Software Engineering. Admiral Makarov National University of Shipbuilding. Mykolaiv, 2021

The work contains: 109 pages, 12 tables, 15 figures, 25 references, 5 appendices.

Relevance of the topic: Obtaining an effective system for estimating the number of lines of code is currently an important task that requires improvement of existing methods. After all, it is the effectiveness of estimating the size of programs that can be the starting point for the success or failure of a project at an early stage of development.

Thus, building a nonlinear regression model using normalization transformation to increase the accuracy of predicting the number of rows of dynamic web-applications in PHP and creating on its basis information technology information processing is relevant.

The purpose and objectives of the study. The purpose of the qualification work is to increase the reliability of estimating the number of terms of the code of dynamic web-applications in PHP. To achieve the intended goal, the following tasks are set: analysis and comparison of existing methods and models for estimating the size of software; substantiation of the need to develop and build a nonlinear regression model for estimating the number of code strings of dynamic web-applications in PHP.

The object of research is the process of estimating the number of terms of dynamic web-applications in PHP.

The subject of research is a mathematical model for estimating the number of lines of code of dynamic web-applications in PHP.

Research methods. Methods of statistical data processing, mathematical modeling and comparison will be used to solve the formulated problems.

The scientific novelty of the obtained results is the following: improved mathematical model for estimating the number of lines of dynamic web-applications code in PHP due to normalization by decimal logarithm, which allowed to increase the reliability of estimating the size of dynamic web-applications in PHP compared to linear models.

The practical value of the results. Software for estimating the number of lines of code of dynamic web-applications in PHP, which was developed as part of the qualification work, will automate and reduce the time of the corresponding calculations.

Approbation of research results: The main provisions of the qualification work were published at the XIII International Scientific and Practical Conference "Free and open source software" (FOSS-2021) (Kharkov, November 16-18, 2021).

Keywords: SIZE ESTIMATION, DYNAMIC WEB-APPLICATIONS, PHP PROGRAMMING LANGUAGE, MATHEMATICAL MODEL

ЗМІСТ

ВСТУП	8
1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ І МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
1.1 Динамічні сайти на php: сутність, переваги та недоліки	12
1.2 Метрики програмного забезпечення	16
1.3 Огляд існуючих методів і моделей оцінювання розміру програмного забезпечення	18
1.4 Показники якості регресійної моделі для оцінювання розміру програмного забезпечення	20
1.5 Обґрунтування необхідності проведення досліджень	22
2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ ЇЇ РЕАЛІЗАЦІЇ	24
2.1 Використання нормалізуючих перетворень при побудові нелінійних регресійних моделей та визначення факторів, які впливають на оцінювання розміру програмного забезпечення	24
2.2 Побудова нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на PHP	26
2.3 Постановка задачі на розробку програми для оцінювання розміру динамічних web-додатків на PHP	35
3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP	37
3.1 Ескізний проект програмного забезпечення для оцінювання розміру динамічних web-додатків на PHP	37
3.1.1 Діаграма варіантів використання програмного забезпечення	37
3.1.2 Специфікації варіантів використання ПЗ для оцінювання розміру динамічних web-додатків на PHP	38

3.1.3 Діаграма діяльності ПЗ для оцінювання розміру динамічних web-додатків на РНР	42
3.1.4 Концептуальна модель даних ПЗ для оцінювання розміру динамічних web-додатків на РНР	45
3.1.5 Інтерфейс користувача ПЗ для оцінювання розміру динамічних web-додатків на РНР.....	46
3.2 Технічний проект ПЗ для оцінювання розміру динамічних web-додатків на РНР	48
3.2.1 Статична модель ПЗ для оцінювання розміру динамічних web-додатків на РНР	48
3.2.2 Динамічна модель ПЗ для оцінювання розміру динамічних web-додатків на РНР	50
3.3 Робочий проект програмного забезпечення для оцінювання розміру динамічних web-додатків на РНР	52
3.3.1 Обґрунтування вибору мови програмування та засобів розробки програмного забезпечення	52
3.3.2 Діаграма компонентів ПЗ для оцінювання розміру динамічних web-додатків на РНР	53
3.3.3 Кодування та тестування ПЗ для оцінювання розміру динамічних web-додатків на РНР	55
3.3.4 Проведення випробування ПЗ оцінювання розміру динамічних web-додатків на РНР	57
4 РЕЗУЛЬТАТИ РОЗРОБКИ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР	59
5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	61
5.1 Вступ	61

5.2 Розрахунок витрат на створення й експлуатацію програми для оцінювання розміру динамічних web-додатків на РНР.....	61
5.3 Економічна ефективність розробки і впровадження програми для оцінювання розміру динамічних web-додатків на РНР.....	64
5.4 Висновки	65
6 ОХОРОНА ПРАЦІ	66
6.1 Вступна частина	66
6.2 Аналіз шкідливих та небезпечних факторів при роботі з комп'ютером у офісі комп'ютерної фірми	67
6.3 Розрахунок системи штучного освітлення в офісі комп'ютерної фірми ..	71
6.4 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів у офісі комп'ютерної фірми	73
7 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	76
7.1 Основні положення охорони навколишнього середовища	76
7.2 Загальна характеристика забруднення атмосферного повітря	77
7.3 Забруднення атмосферного повітря автотранспортом	78
7.4 Заходи щодо зменшення забруднення атмосферного повітря	81
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР	90
ДОДАТОК Б – ТЕКСТ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР	95
ДОДАТОК В – ОПИС ПРОГРАМИ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР	102
ДОДАТОК Г – ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР.....	104
ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР	107

ВСТУП

Актуальність теми Процес оцінювання розміру програмного забезпечення (ПЗ) є одним з основним факторів, які впливають на успішність виконання проектів з розробки ПЗ [1]. Достовірне оцінювання розміру проектів з розробки програмного забезпечення дозволяє оцінити їх собівартість та тривалість. Це дасть можливість успішно завершувати розробку проектів, тоді як порушення запланованих витрат та тривалості проекту приведе до фінансових втрат з боку замовника та з боку розробника.

Все більша кількість компаній прагне замовити розробку програмного забезпечення. Це пов'язано з прагненням оптимізувати процеси складання звітності, управління ресурсами і проведення різних операцій. І великі корпорації, і порівняно невеликі, ще молоді компанії, які тільки беруть курс на розширення, замислюються про необхідність автоматизувати як окремі бізнес-процеси, так і роботу взагалі.

В наш час існує досить поширена проблема в галузі інформаційних технологій: не існує єдиного вимірювання, набору метрик та показників для оцінки розмірів програмного забезпечення. Існує багато методів для оцінки розміру програми, деякі з них базуються на кількості рядків коду, інші методи обчислюють розмір з функціональних, технічних або інших аспектів. Але більшість з них стають непридатними до використання через відсутність даних, ресурсів або експертних навичок в цій галузі.

Актуальність проблеми отримання ефективної системи оцінювання кількості рядків коду в даний час є важливим завданням, що вимагає удосконалення існуючих методів. Адже саме ефективність оцінки розміру програм може стати відправною точкою для успіху або невдачі проекту на ранньому етапі розробки.

Для підвищення достовірності оцінювання кількості рядків динамічних web-додатків на PHP, необхідно побудувати нелінійну регресійну модель

оцінювання кількості рядків динамічних web-додатків на PHP із використанням нормалізуючого перетворення у вигляді десяткового логарифму, яке дозволить звужити довірчий інтервал та інтервал прогнозування нелінійної регресії в порівнянні з лінійною у припущенні про нормальність вихідних даних [2].

Таким чином, побудова нелінійної регресійної моделі із використанням нормалізуючого перетворення для підвищення достовірності оцінювання кількості рядків динамічних web-додатків на PHP і створення на її основі інформаційної технології переробки інформації є актуальною та має практичну цінність [3, 4].

Мета і завдання дослідження. Мета кваліфікаційної роботи полягає в підвищенні достовірності оцінювання кількості строк коду динамічних web-додатків на PHP.

Мета роботи визначає наступні завдання дослідження:

- аналіз та порівняння наявних методів та моделей для оцінювання розміру ПЗ;
- обґрунтування необхідності розробки нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP;
- отримання емпіричних даних для побудови нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP;
- нормалізація вихідних даних з використанням нормалізуючого перетворення десятковий логарифм;
- перед обробка вихідних даних;
- побудова лінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP, довірчого інтервалу та інтервалу прогнозування для даних без викидів;
- побудова нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP, довірчого інтервалу та інтервалу прогнозування для емпіричних даних через зворотнє перетворення;

- побудова лінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на РНР, довірчого інтервалу та інтервалу прогнозування для емпіричних даних;
- порівняння лінійної та нелінійної регресійних моделей оцінювання кількості строк коду динамічних web-додатків на РНР;
- розробка ПЗ для прогнозування кількості строк коду динамічних web-додатків на РНР.

Об’єктом досліджень є процес оцінювання кількості рядків коду динамічних web-додатків на РНР.

Предметом досліджень є нелінійна модель регресії оцінювання кількості рядків коду динамічних web-додатків на РНР.

Методи дослідження. При розв’язанні сформульованих завдань будуть використовуватися методи обробки статистичних даних, математичного моделювання і порівняння.

Наукова новизна отриманих результатів полягає в наступному: удосконалена математична модель оцінювання кількості рядків коду динамічних web-додатків на РНР за рахунок нормалізації десятковим логарифмом, що дозволяє підвищити достовірність оцінювання розміру динамічних web-додатків на РНР порівняно з лінійними моделями.

Практичне значення одержаних результатів. ПЗ для оцінювання кількості рядків коду динамічних web-додатків на РНР, яке розроблено в рамках кваліфікаційної роботи, дозволить автоматизувати та скоротити час відповідних розрахунків.

Особистий внесок здобувача. Усі результати кваліфікаційної роботи, що виносяться на захист, отримані автором особисто. У тезах доповіді [5], написаних у співавторстві, здобувачу належить аналіз методів для побудови нелінійних регресійних моделей.

Апробація результатів досліджень. Основні положення кваліфікаційної роботи були оприлюднені на XIII Міжнародній науково-

практичній конференції «Free and open source software» (FOSS-2021) (м. Харків, 16-18 листопада 2021).

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

1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ І МОДЕЛЕЙ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 ДИНАМІЧНІ САЙТИ НА PHP: СУТНІСТЬ, ПЕРЕВАГИ ТА НЕДОЛІКИ

Сутність, переваги та недоліки динамічних сайтів

Динамічний сайт – це сайт, який складається зі змінюваних веб-сторінок. Вихідний код таких веб-сторінок, зазвичай, генерується під час обробки HTML файлу інтерпретатором будь-якої мови програмування. Найбільш вживаними мовами програмування на стороні сервера являються: Java, PHP, Ruby.

Коли клієнт запитує динамічну веб-сторінку, запускається цілий ланцюг операцій [6]:

- Формується запит до веб-серверу на отримання веб-сторінки. Веб-сервер знаходить веб-сторінку та передає її інтерпретатору PHP.

- Після отримання веб-сторінки інтерпретатор виконує програму, яка міститься в HTML-документі, взаємодіючи при цьому, якщо необхідно, з іншими елементами обчислювальної системи.

- Виконавши всі необхідні дії, інтерпретатор передає веб-серверу отриманий код HTML-документа.

- Отримавши згенерований код HTML-документа, веб-сервер відправляє його клієнту.

До переваг динамічних сайтів відносяться [6]:

- Відносно недорога підтримка сайту.

Однією з головних особливостей динамічних сайтів є поділ дизайну (шаблону), системи обробки інформації ("движка") і змісту (контенту). Тобто веб-майстру досить внести будь-яку зміну в одному місці, щоб вона

проявилася на всіх сторінках. Також на сайт можна відносно швидко "приміряти" різні варіанти дизайну.

Відділення контенту від движка і дизайну дозволяє залучити до наповнення і підтримки сайту людину, не знайому з інтернет-технологіями, але яка добре розбирається в області діяльності фірми. Це забезпечує коректне і максимально повне утримання веб-проекту.

- Широкі функціональні можливості.

Для користувачів це можливість реєстрації і аутентифікації, зберігання персональної інформації, форум, розсилка, пошук по сайту, форми замовлень, створення інтернет-магазину і багато інших можливостей.

- Підтримка бізнес-процесів.

Сайт може стати прекрасним інструментом для працівників компанії. Так, наприклад, за допомогою веб-сервера є можливість організувати обмін інформацією між філіями, розташованими далеко (наприклад, в різних містах).

Крім того, динамічні сайти допускається інтегрувати з різними системами, що вже діють усередині організації. Так, наприклад, можна створити прайс-лист, який буде автоматично генеруватися та інформація для якого буде отримуватися з програми складського обліку товарів.

Недоліками динамічних сайтів являються [6]:

- Необхідність використання додаткових програмних засобів для створення динамічного сайту.

При використанні статичного сайту, серверу залишається тільки показати сторінки, так як всі вони вже готові, а на динамічному сайті вносяться на сторінки зміни і тому необхідно розробляти певні програми.

Залежно від складності сайту, трудомісткість та вартість розробки таких програм може дуже сильно змінюватись. Зараз існує безліч готових рішень для створення сайту, у тому числі безкоштовних. У мережі можна знайти не один десяток усіляких безкоштовних скриптів, які дозволять створити на сайті форум, дошку оголошень, клуб знайомств, магазин тощо.

Але якщо потрібно щось специфічне, то тут не обійтися без додаткових розробок.

– Підвищення вимог до характеристик серверних систем.

Через необхідність внесення змін у сторінки зростають вимоги до потужності серверів. Ця проблема особливо проявляється на сайтах, які мають велику відвідуваність. Пришвидшити роботу сайту в цьому випадку може оптимізація коду.

Отже, вартість послуг за змістом такого сайту набагато вища, ніж статичного. Хоча зараз, навіть багато безкоштовних хостингів підтримують можливість створення динамічних сайтів, не кажучи вже про платні, де все необхідне входить до стандартного набору послуг.

– Третім недоліком, що також впливає з першого, є складність великих структурних змін сайту. У цьому випадку все пов'язане з програмним забезпеченням, яке використовується на сайті.

Незважаючи на ці недоліки, динамічних сайтів у мережі стає все більше і більше, мабуть, описані раніше переваги перекривають всі недоліки. Тепер давайте розглянемо, що ж на сьогоднішній момент є в мережі, які можливості може отримати людина, яка бажає створити сайт.

PHP – мова для розробки WEB-додатків

Автором мови PHP є Расмус Лердорф, який прагнув розробити набір інструментів спрощення процесу створення динамічних веб-сторінок. Сучасна PHP – це мова загального призначення, проте її частіше всього використовують в якості серверного інструменту для генерації HTML-коду, який потім інтерпретується веб-браузером.

PHP є мовою з відкритим кодом, яку розвивають програмісти-ентузіасти з усього світу. Вона має простий синтаксис, частково схожий на Java та C++. Це проект, який постійно розвивається.

На сьогоднішній день переважна більшість сайтів, сервісів та програм, а також такі популярні платформи як Joomla, Drupal, WordPress, 1С-Bitrix та UMI.CMS написані саме на PHP-мові.

Мова PHP має низку незаперечних переваг [7]:

- Висока швидкість роботи та, відповідно, загальна продуктивність ресурсів.

- Бюджетність, економічність.

Знайти фахівця не важко, ціна за написання програми на php не велика.

- Проста для освоєння, має простий синтаксис.

- Має гарну сумісність сумісність та легко переноситься.

Програми добре працюють на різних платформах.

- Набір тексту коду та його редагування можна здійснювати у будь-якому текстовому чи html-редакторі.

- Гнучкість

PHP – дуже гнучка мова. Майже всі задачі на PHP можна розв'язати різними способами.

- Багатозадачність та широкі можливості створення будь-яких веб-додатків, блогів, гостьових книг, інтернет-магазинів, сайтів, робота з редиректами, заголовками, pdf-документами, базами даних, електронною поштою та інш.

- Постійний розвиток

Постійно виконуються оновлення, які значно розширили можливості мови та збільшили продуктивність.

- Документування

PHP добре документована.

- Велике ком'юніті.

Спільнота PHP-розробників більше, ніж у Python, тому й вибір бібліотек для веб-розробки ширший. У ком'юніті простіше знайти потрібне керівництво чи отримати відповідь на запитання.

1.2 Метрики програмного забезпечення

Найпростішими і найефективнішими в реалізації є системи управління на базі вимірюваних характеристик об'єкта. Це також справедливо і для програмних проектів. Оскільки процес створення програмного забезпечення досить складний та багатогранний, то навіть незначні недоліки системи вимірювання неминуче відіб'ються на якості управління. Основу будь-якої системи вимірювання складають окремі показники (метрики). До найбільш актуальних метрик програмних проектів відносяться[8]:

– Метрики розміру

Найбільш поширеною метрикою вихідного коду програми є кількість рядків коду (SLOC).

Для метрики SLOC є багато похідних:

- кількість порожніх рядків (BLOC);
- кількість рядків з коментарями(CLOC);
- кількість рядків з вихідним кодом і коментарями (Lines with Both Code and Comments, C & SLOC);
- середнє число рядків для функцій (методів);
- середнє число рядків для модулів;
- середнє число рядків для класів.

Також поряд із перерахованими застосовуються інші показники, що відображають кількісну оцінку окремих складових проекту: число модулів, функцій тощо. Однак крім метрик, що підраховують число елементів проекту, до метрик розміру слід віднести також функціональні точки та інші метрики, які підраховуються на основі вимог користувача, специфікацій, прецедентів і т.д. У більшості випадків головне призначення цієї групи метрик полягає в оцінці обсягу робіт (трудомісткості, вартості, тривалості та інших).

– Метрики складності [9, 10]

Інша поширена група метрик програмних проектів – показники, що характеризують їх складність. Ці метрики застосовуються частіше для розробленого програмного забезпечення, але також можуть примінятися і на ранніх стадіях роботи при здійсненні проектування.

Метрики складності призначені, в основному, для знаходження тих ділянок програми, які потенційно можуть містити помилки та являються місцями з підвищеними ризиками на всіх етапах розробки програмного забезпечення.

Найбільш поширеною метрикою складності є цикломатична складність. Дана метрика призначена для оцінювання складності потоку керування програми і обчислюється на основі орієнтованого графа, де обчислювальні оператори або вирази представляються в вигляді вузлів, а передача управління між вузлами – у вигляді дуг.

Обчислення цієї метрики при розробці програмного забезпечення дозволяє своєчасно визначити найбільш складні ділянки програми, які мають високі ризики, і спробувати усунути ризики через внесення змін.

Для метрики цикломатичної складності існує багато похідних метрик.

– Об'єктно-орієнтовані метрики

В зв'язку з застосуванням об'єктно-орієнтованого програмування зв'явилася необхідність в появі аналогічних метрик. Так як основним елементом об'єктно-орієнтованого програмування є класи, які містять методи та атрибути, то саме класи, методи та атрибути використовуються в об'єктно-орієнтованих метриках. Основними об'єктно-орієнтованими метриками є:

- загальне число класів;
- загальне число методів;
- загальне число атрибутів;
- середнє число методів на клас;
- середнє число атрибутів на клас.

Розмір програмного проекту – це одна з основних величин, через яку обчислюються такі важливі показники як собівартість та тривалість розробки. Зазвичай розмір програми не підпорядковується нормальному закону розподілу і залежить від ряду метрик програмного забезпечення, як наприклад, кількість класів, кількість методів, кількість атрибутів та інш.

1.3 Огляд існуючих методів і моделей оцінювання розміру програмного забезпечення

Основними причинами для оцінювання розміру програмного забезпечення є:

- визначення якості існуючого ПЗ або його розробки/експлуатації;
- прогнозування якості ПЗ або його розробки/експлуатації у майбутньому;
- поліпшення якості ПЗ;
- оцінювання вартості майбутніх проектів;
- оцінювання продуктивності застосування нових засобів і методів;
- прогноз майбутніх потреб у персоналі;
- передбачення і скорочення майбутніх потреб у технічному обслуговуванні.

Розглянемо одну з класифікацій методів та моделей для оцінювання розміру програмних проектів: методи, що базуються на експертному оцінюванні; методи, які використовують функціональні точки; моделі з параметрами; регресійні та комбіновані моделі[3, 11-13].

Методи, що базуються на експертних оцінках. Ці методи широко використовують для оцінювання різних параметрів розробки програмних проектів, в тому числі і розміру. Методи базуються на оцінках кваліфікованих фахівців (експертів), досить прості в реалізації. Проблемами методів є суб'єктивність думок, важкість визначення кваліфікації, можливість упередженості експертів. Різновидністю експертного оцінювання

є метод оцінювання по аналогії, в якому експерти використовують інформацію по попередньо виконаним роботам.

Методи, які базуються на функціональних точках. Так як використання функціональних точок ґрунтується на визначенні вимог, то оцінювання розміру майбутнього проекту може бути виконано на ранніх стадіях його розробки. Для обчислення кількості функціональних точок визначають число входів, виходів, файлів даних, запитів і зовнішніх інтерфейсів, а також коефіцієнтів коригування складності.

На практиці використовують спеціальні коефіцієнти для переходу від кількості строк коду до числа функціональних точок. Ці коефіцієнти залежать від мови програмування та являються досить приблизними.

Існують різні думки щодо оцінювання розміру ПЗ в строках коду чи функціональних точках. В кожній думці є свої прихильники та опоненти. Та, незважаючи на критичні зауваження, обидва підходи дають менеджеру корисну інформацію, що дозволяє вже на початкових етапах отримати оцінки проекту з достовірністю, що відповідає початковому етапу проектування.

Параметричні моделі. Параметрична модель являється математичним співвідношенням, яке містить змінні параметри. Керівники програмних проектів використовують параметричні моделі для оцінювання розміру, тривалості, собівартості проектів, чисельності персоналу та вартості.

Основною перевагою таких моделей є те, що вони об'єктивні, відтворювані, відкалібровані та прості у використанні, хоча калібрування за попереднім досвідом може бути недоліком при застосуванні до істотно іншого проекту.

Регресійні моделі. Регресійна модель будується на емпіричних даних. Обираються незалежні та залежні змінні, вид функціональної залежності та знаходяться параметри моделі. Знаходження параметрів регресійної моделі іноді називають навчанням моделі. Розрізняють лінійні та нелінійні регресійні моделі, однофакторні та багатфакторні.

Недоліки регресійного аналізу: моделі, що мають дуже малу складність, можуть виявитися неточними, а моделі, що мають надмірну складність, можуть виявитися перенавченими.

Під час вирішення завдань регресійного аналізу постають такі питання:

- при побудові регресій як вибрати тип функціональної залежності?
- який розподіл випадкової змінної?
- якою цільовою функцією оцінити якість апроксимації?
- яким чином знайти параметри моделі, який має бути алгоритм оптимізації параметрів?

Регресійні моделі активно використовуються для оцінювання розміру програмного забезпечення [3-4]. Параметри регресійних моделей повинні бути відкалібровані в залежності від мови програмування, платформи, технології розробки.

Комбіновані моделі. Є комбінацією кількох моделей для збільшення точності оцінювання.

При використанні точної та надійної методології для оцінювання розміру можна зробити висновок про якість програмного забезпечення та навіть усього програмного процесу і, при необхідності, вжити подальших заходів.

На жаль, багато з відомих методів та моделей оцінювання розміру ПЗ є непридатними до використання, бо потребують багато недоступних даних або експертних навичок. Отже, існує потреба у вдосконаленні існуючих методів та моделей відповідно до наявних ресурсів і як результат розробці та впровадженні ПЗ для автоматизації цього процесу.

1.4 Показники якості регресійної моделі для оцінювання розміру програмного забезпечення

Для перевірки якості як лінійних так і нелінійних регресійних моделей часто використовують наступні показники: коефіцієнт детермінації R^2 , суму

квадратів відхилень S_Y , середню величину відносної похибки $MMRE$ та рівень прогнозування $PRED(l)$ [14, 15].

Розглянемо зміст кожного показника.

Коефіцієнт детермінації R^2

Розрахункова формула для R^2 представлена в (1.1).

$$R^2 = 1 - (\sum_1^n (y_i - \hat{y}_i)^2 / \sum_1^n (y_i - \bar{y})^2), \quad (1.1)$$

де y_i – вихідна величина;

$\hat{y}_i$ – розрахована величина;

$\bar{y}$ – середня величина y_i ;

n – кількість значень вихідних даних.

Цей коефіцієнт показує частку варіації результативної ознаки, що є під впливом досліджуваних чинників, тобто визначає, яка частка варіації ознаки Y врахована в моделі та обумовлена впливом на неї факторів.

Чим ближче R^2 до одиниці, тим вища якість моделі.

Сума квадратів відхилень S_Y

Розрахункова формула для S_Y представлена в (1.2).

$$S_Y = \sum_1^n (y_i - \hat{y}_i)^2 \quad (1.2)$$

Сума квадратів відхилень помилки S_Y характеризує розсіювання залежної змінної Y щодо лінії регресії.

Середня величина відносної похибки $MMRE$

Розрахункова формула для $MMRE$ представлена в (1.3).

$$MMRE = \frac{1}{n} \cdot \sum_1^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (1.3)$$

Прийнятне значенням $MMRE \leq 0,25$.

Рівень прогнозування $PRED(l)$

Розрахункова формула для $PRED(l)$ представлена в (1.4).

$$PRED(l) = \frac{k}{n}, \quad (1.4)$$

де k – кількість величин $\left| \frac{y_i - \hat{y}_i}{y_i} \right| \leq l$ ($i = \overline{1, n}$).

Часто використовується $l = 0,25$. Прийнятне значення $PRED(0,25) \geq 0,75$.

1.5 Обґрунтування необхідності проведення досліджень

Емпіричні дані, що використовуються при оцінюванні розміру програмних проектів, зазвичай не являються нормально розподіленими та зв'язки між ними не завжди є лінійними. Це приводить до невисокої точності оцінювання та ставить під сумнів достовірність отриманих результатів. Тому використання лінійної теорії для вихідних даних є невиправданим.

Щоб позбавитися цієї проблеми, перед побудовою математичної моделі виконують нормалізацію даних з допомогою одного з нормалізуючих перетворень. Найбільш вживаними нормалізуючими перетвореннями являються наступні: десятковий логарифм, натуральний логарифм, перетворення Вох-Сох, перетворення Джонсона та інші.

Для нормалізованих даних можемо побудувати лінійну регресію, довірчий інтервал та інтервал передбачення відомим способом. Для повернення до емпіричних даних. Для повернення від нормалізованих до емпіричних даних необхідно застосувати зворотне перетворення. В

результаті отримаємо нелінійне рівняння, прогнозний інтервал та інтервал довіри для емпіричних даних.

Довірчий інтервал лінійного рівняння регресії можна побудувати за наступною формулою [16]:

$$y = \hat{y} \pm t_{(\alpha/2, n-2)} \cdot S \cdot \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (1.5)$$

де $\hat{y}$ – значення y , розраховане за рівнянням регресії;

$t_{(\alpha/2, n-2)}$ – квантіль t -розподілу Стюдента;

α – рівень значущості;

n – кількість значень вихідних даних;

$$S = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}.$$

Інтервал прогнозування лінійного рівняння регресії можна побудувати за наступною формулою [17]:

$$y = \hat{y} \pm t_{(\alpha/2, n-2)} \cdot S \cdot \sqrt{1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (1.6)$$

Тим самим ми отримаємо більш точну математичну модель для оцінювання кількості рядків динамічних web-додатків на РНР.

2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ ЇЇ РЕАЛІЗАЦІЇ

2.1 Використання нормалізуючих перетворень при побудові нелінійних регресійних моделей та визначення факторів, які впливають на оцінювання розміру програмного забезпечення

При розв'язуванні деяких задач в програмній інженерії використовуються нелінійні регресійні моделі. До таких задач відносяться задачі прогнозування розміру програмного забезпечення на ранніх стадіях його розробки, задачі прогнозування трудомісткості, тривалості розробки та інші. Відомо, що для використання лінійної регресійної теорії існує ряд припущень. Основними з них є нормальність розподілу емпіричних даних та лінійність зв'язків між ними. Дані, які застосовуються для побудови регресійних моделей, не завжди задовольняють цим припущенням. Тому виникає необхідність враховувати нелінійність зв'язків між вихідними даними та використовувати нормалізацію.

Для побудови нелінійних регресійних моделей можна застосовувати методи простого перебору, лінеарізуючих та нормалізуючих перетворень [18].

Метод простого перебору полягає дослідженні різних видів рівнянь регресії, з яких потім, використовуючи певні критерії, обирається найкраще наближення. Основний недолік цього методу – великі затрати часу за рахунок великої кількості обчислень.

Метод лінеарізуючих перетворень полягає в застосуванні до даних, що аналізуються, такого перетворення, яке б дозволило представити існуючу залежність у вигляді лінійної функції. Підібрати необхідне перетворення не завжди вдається.

Метод нормалізуючих перетворень полягає в застосуванні до вихідних даних такого перетворення, яке дає змогу привести вихідні ненормалізовані дані до нормально розподілених. Для отриманих даних будується лінійна регресійна модель, визначаються довірчі інтервали та інтервали передбачення, а потім, з використанням зворотнього перетворення, виконується перехід до нелінійної регресії, її довірчих інтервалів та інтервалів передбачення.

Вибір нормалізуючого перетворення залежить від вихідних даних та впливає на якість отриманих результатів. Короткий огляд найбільш відомих з них можна знайти в публікації [19]. До самих простих нормалізуючих перетворень, які використовуються в програмній інженерії, відноситься логарифмічне. Досить часто використовується нормалізуюче перетворення Бобса-Кокса. Вказане перетворення має один параметр, рекомендації по знаходженню оптимального значення якого наведені в [19]. Кращий результат можна отримати, застосовуючи перетворення Джонсона, яке в загальному випадку має 4 параметри та три сім'ї функцій. Рекомендації по вибору сім'ї та співвідношення для обчислення параметрів наведені в [16].

Використання методу нормалізуючих перетворень дає змогу побудувати нелінійну регресійну модель, якість якої залежить від вихідних даних та обраного перетворення.

Для побудови нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на РНР та визначення довірчих інтервалів та інтервалів прогнозування будемо використовувати нормалізуючи перетворення у вигляді десяткового логарифму. Нелінійна регресійна модель оцінювання розміру динамічних web-додатків на РНР буде будуватися на основі наступних факторів:

- кількість класів;
- кількість рядків програмного коду;

2.2 Побудова нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на PHP

Для побудови нелінійної регресійної моделі були обрані наступні фактори:

- кількість класів (X);
- кількість рядків коду (Y).

Розмір програмного забезпечення буде розраховуватися як функція від одного фактору:

$$Y = f(X). \quad (2.1)$$

Всього зібрані дані про 33 проекти. 3 з них були вилучені як викиди

В таблиці 2.1 наведені вихідні дані без викидів для побудови нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на PHP.

Таблиця 2.1 – Вихідні дані для побудови нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на PHP

№ додатку	Y (кількість рядків коду)	X(кількість класів)
1	2	3
1	1386	7
2	1335	8
3	1462	8
4	1404	13
5	2701	13
6	3293	14
7	2836	18
8	2684	23
9	5065	27
10	2646	30
11	8884	33
12	4758	33
13	4415	35
14	7018	35

Продовження табл. 2.1

1	2	3
15	6475	35
16	8866	46
17	3844	49
18	8301	50
19	5452	54
20	7570	67
21	8544	68
22	12758	74
23	11259	82
24	14812	93
25	9344	95
26	34655	101
27	13157	105
28	24701	123
29	24036	157
30	65812	253

Вихідні дані Y , X не нормально розподілені. Це було перевірено шляхом підрахунку χ^2 критичного. Це означає, що для побудови нелінійної регресійної моделі необхідно виконати нормалізацію вихідних даних [20].

Нормалізацію емпіричних даних було виконано за допомогою десяткового логарифму. В результаті було отримано нормалізовані величини Z_Y та Z_X . Емпіричний розподіл для вибірки Y представлено на рис. 2.1.а, а її нормалізований розподіл представлений на рис. 2.1.б.

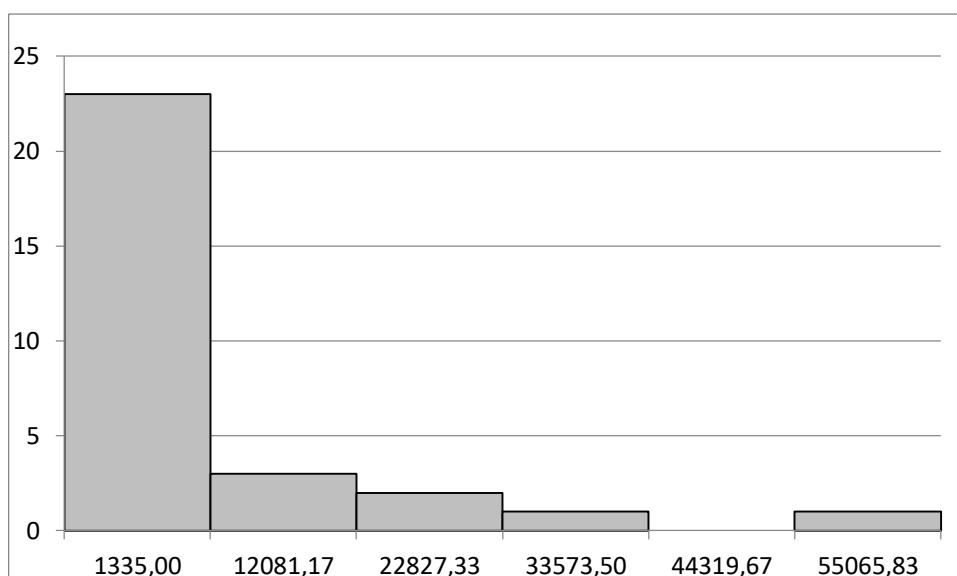


Рисунок 2.1.а – Емпіричний розподіл для вибірки Y

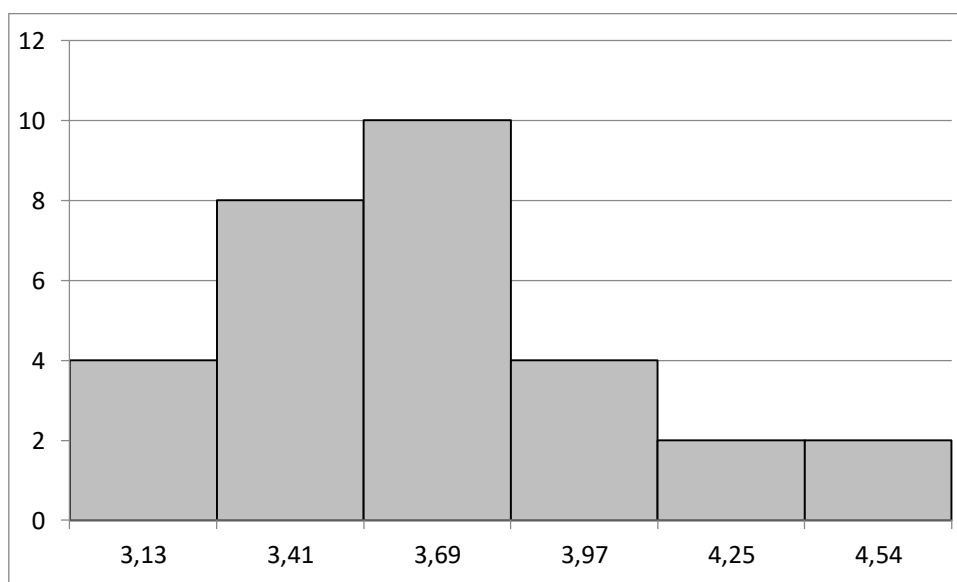


Рисунок 2.1.б – Нормалізований розподіл для вибірки Y

Емпіричний розподіл для вибірки X представлено на рис. 2.2.а, а її нормалізований розподіл представлений на рис. 2.2.б.

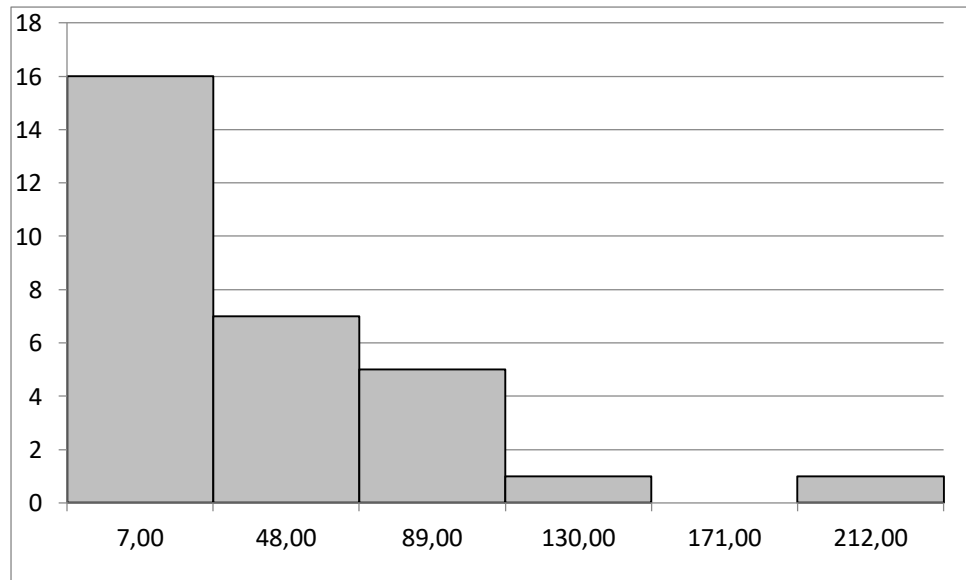


Рисунок 2.2.а – Емпіричний розподіл для вибірки X

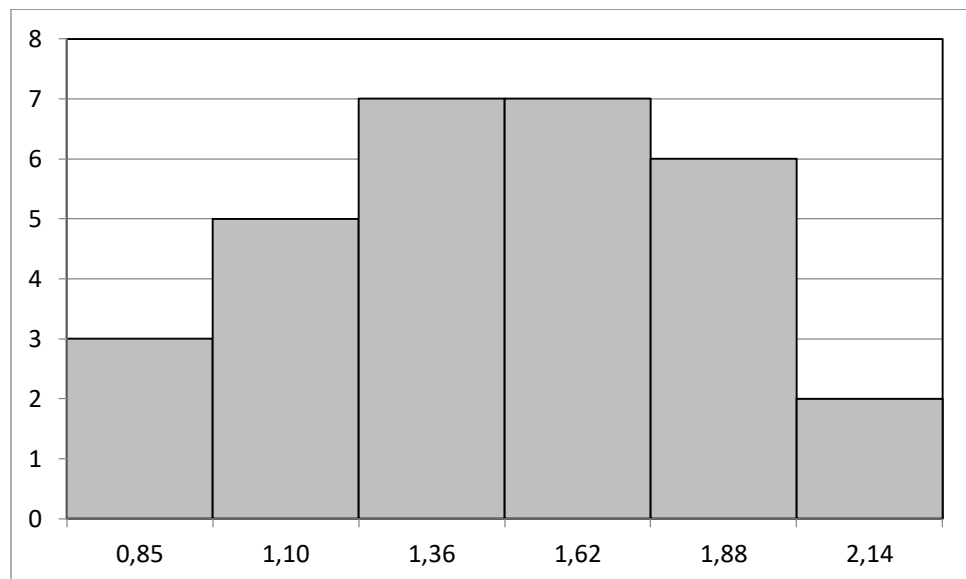


Рисунок 2.2.б – Нормалізований розподіл для вибірки X

Нелінійна регресійна модель будується на основі лінійної регресійної моделі. Лінійне регресійна модель з двома параметрами виглядає наступним чином [21]:

$$Z_{\hat{Y}} = b_0 + b_1 Z_x + \varepsilon. \quad (2.2)$$

Для знаходження параметрів використовували метод найменших квадратів. Отримали:

$$b_1 = 0,9922; b_0 = 2,2095.$$

Звідси

$$Z_{\hat{Y}} = 2,2095 + 0,9922Z_x + \varepsilon.$$

Для отриманої лінійної регресії обчислили залишки ($\varepsilon_i = \hat{y}_i - y_i$) та з допомогою критерію χ^2 перевірили, що вони мають нормальний розподіл.

Довірчий інтервал для побудованої лінійної моделі обчислюємо за формулою (2.5).

Інтервал передбачення для побудованої лінійної моделі обчислюємо за формулою (2.6).

Після цього з допомогою формул зворотного перетворення переходимо до вихідних даних [17]. Формули зворотного перетворення (2.3-2.5) застосовуємо до функції $Z_{\hat{Y}}$ та верхньої (Z_{max}) і нижньої (Z_{min}) границь довірчого інтервалу та інтервалу передбачення. Отримали вихідні значення залежної змінної і верхню та нижню границі довірчого інтервалу та інтервалу передбачення для нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на РНР (Y_{max} та Y_{min} відповідно):

$$\hat{Y} = 10^{Z_{\hat{Y}}} \quad (2.3)$$

$$Y_{max} = 10^{Z_{y_{max}}} \quad (2.4)$$

$$Y_{min} = 10^{Z_{y_{min}}} \quad (2.5)$$

Таким чином нелінійна регресійна модель має вигляд:

$$\hat{Y} = 10^{b_0 + \varepsilon} * X^{b_1},$$

а саме:

$$\hat{Y} = 10^{2,2095 + \varepsilon} * X^{0,9922}.$$

Також у роботі була побудована лінійна регресійна модель у припущенні про нормальність вихідних даних. Вона має вигляд:

$$\hat{Y} = 226.6996X - 2900.8222 + \varepsilon.$$

Для цієї моделі також пораховані довірчий інтервал та інтервал передбачення.

В табл. 2.2 знаходяться довірчі інтервали для лінійної регресії у припущенні про нормальність емпіричних даних та для нелінійної регресії, отриманої за допомогою нормалізації десятковим логарифмом.

Таблиця 2.2 – Довірчі інтервали для лінійної регресії у припущенні про нормальність емпіричних даних та для нелінійної регресії, отриманої за допомогою нормалізації десятковим логарифмом

№ п/п	Ненормалізовані дані			Нормалізовані дані		
	Нижня границя	Верхня границя	Довжина інтервалу	Нижня границя	Верхня границя	Довжина інтервалу
1	2	3	4	5	6	7
1	-4138,7375	1510,8879	5649,6254	804,6979	1549,7964	745,0985
2	-3109,3827	934,9325	4044,3152	936,8586	1735,0354	798,1768
3	-3109,3827	934,9325	4044,3152	936,8586	1735,0354	798,1768
4	-1978,6425	2071,1886	4049,8311	1622,1298	2626,0220	1003,8922
5	-1978,6425	2071,1886	4049,8311	1622,1298	2626,0220	1003,8922
6	-1752,6088	2298,5542	4051,1630	1762,5561	2799,6639	1037,1078
7	-848,9851	3208,5276	4057,5127	2330,3220	3486,6712	1156,3492
8	278,9398	4347,5990	4068,6592	3045,3920	4339,3601	1293,9681
9	1179,2463	5260,8897	4081,6434	3615,1453	5024,8539	1409,7087
10	1852,5878	5947,7461	4095,1583	4038,5720	5543,9472	1505,3752
11	2523,4215	6637,1101	4113,6886	4457,4356	6068,7378	1611,3022
12	2523,4215	6637,1101	4113,6886	4457,4356	6068,7378	1611,3022
13	2968,5937	7098,7365	4130,1428	4733,7814	6422,1863	1688,4049
14	2968,5937	7098,7365	4130,1428	4733,7814	6422,1863	1688,4049
15	2968,5937	7098,7365	4130,1428	4733,7814	6422,1863	1688,4049
16	5326,7810	9727,9412	4401,1602	6208,3291	8422,3924	2214,0633
17	5874,4493	10540,4708	4666,0215	6597,2762	8984,4590	2387,1828
18	6025,6219	10842,6975	4817,0756	6725,7403	9173,3259	2447,5856
19	6085,4934	12596,4230	6510,9296	7233,9536	9936,0732	2702,1196
20	9912,5643	14663,5427	4750,9785	8830,5141	12488,1509	3657,6368

Продовження табл. 2.2

1	2	3	4	5	6	7
21	10206,1354	14823,3709	4617,2355	8950,2652	12688,6550	3738,3898
22	11745,8541	16004,0479	4258,1938	9660,8397	13902,9383	4242,0986
23	13625,2872	17751,8089	4126,5217	10589,1591	15549,8225	4960,6635
24	16147,0806	20217,4076	4070,3271	11835,1323	17860,5586	6025,4263
25	16603,1042	20668,1826	4065,0784	12058,3237	18285,9566	6227,6329
26	17969,0959	22022,5866	4053,4907	12722,3155	19571,1829	6848,8674
27	18878,5860	22926,6936	4048,1076	13160,5934	20435,2149	7274,6215
28	22965,7682	27000,6983	4034,9301	15094,5759	24388,6781	9294,1023
29	30677,6534	34704,3885	4026,7350	18609,3059	32107,5087	13498,2028
30	52443,1164	56465,2560	4022,1396	27882,3849	55233,2238	27350,8389

В табл. 2.3 знаходяться інтервали прогнозування для лінійної регресії у припущенні про нормальність емпіричних даних та для нелінійної регресії, отриманої за допомогою нормалізації десятковим логарифмом.

Таблиця 2.3 – Інтервали прогнозування для лінійної регресії у припущенні про нормальність емпіричних даних та для нелінійної регресії, отриманої за допомогою нормалізації десятковим логарифмом

№ п/п	Ненормалізовані дані			Нормалізовані дані		
	Нижня границя	Верхня границя	Довжина інтервалу	Нижня границя	Верхня границя	Довжина інтервалу
1	2	3	4	5	6	7
1	-12832,5739	10204,7244	23037,2983	459,2388	2715,6197	2256,3809
2	-12593,4633	10419,0130	23012,4762	527,9958	3078,5903	2550,5945
3	-12593,4633	10419,0130	23012,4762	527,9958	3078,5903	2550,5945
4	-11401,3939	11493,9400	22895,3340	873,0464	4879,1776	4006,1311
5	-11401,3939	11493,9400	22895,3340	873,0464	4879,1776	4006,1311
6	-11163,6819	11709,6273	22873,3092	942,1141	5237,7567	4295,6426
7	-10215,1958	12574,7383	22789,9341	1218,1002	6670,2774	5452,1772

Продовження табл. 2.3

1	2	3	4	5	6	7
8	-9034,9507	13661,4896	22696,4404	1561,9699	8460,5036	6898,5337
9	-8095,0907	14535,2267	22630,3175	1835,9254	9894,5072	8058,5818
10	-7392,7476	15193,0815	22585,8291	2040,6695	10971,7080	8931,0385
11	-6692,6064	15853,1381	22545,7445	2244,7794	12050,6308	9805,8514
12	-6692,6064	15853,1381	22545,7445	2244,7794	12050,6308	9805,8514
13	-6227,0747	16294,4049	22521,4796	2380,4988	12770,9478	10390,4490
14	-6227,0747	16294,4049	22521,4796	2380,4988	12770,9478	10390,4490
15	-6227,0747	16294,4049	22521,4796	2380,4988	12770,9478	10390,4490
16	-3684,3665	18739,0887	22423,4552	3121,9704	16748,7119	13626,7414
17	-2996,1366	19411,0567	22407,1933	3322,7661	17838,4379	14515,6718
18	-2767,2275	19635,5469	22402,7744	3389,5668	18202,1513	14812,5845
19	-1854,0996	20536,0160	22390,1156	3656,1256	19659,3608	16003,2352
20	1085,8126	23490,2944	22404,4818	4515,6422	24421,0653	19905,4231
21	1310,2026	23719,3037	22409,1012	4581,3475	24788,9571	20207,6096
22	2651,2886	25098,6134	22447,3248	4974,4072	27001,0181	22026,6109
23	4425,4607	26951,6355	22526,1749	5495,4753	29962,7483	24467,2730
24	6839,2299	29525,2583	22686,0284	6206,6553	34057,3246	27850,6692
25	7274,9386	29996,3482	22721,4096	6335,3347	34804,4725	28469,1378
26	8576,3163	31415,3662	22839,0499	6720,2661	37050,7298	30330,4637
27	9439,1610	32366,1186	22926,9576	6975,9887	38552,1775	31576,1888
28	13276,3452	36690,1213	23413,7761	8118,3555	45346,2222	37227,8667
29	20334,7649	45047,2770	24712,5121	10242,5480	58334,9426	48092,3946
30	39225,9667	69682,4058	30456,4391	16052,8987	95934,9481	79882,0493

Аналіз інтервалів показує, що вони вужчі для переважної кількості нормалізованих даних. Також для нормалізованих даних границі інтервалів не містять від'ємних значень.

Для лінійної регресії у припущенні про нормальність емпіричних даних та для нелінійної регресії, отриманої за допомогою нормалізації десятковим

логарифмом, були обчислені наступні характеристики якості: R^2 , $MMRE$ та $PRED(0,25)$ [22, 23]. Результати обчислень наведені в табл. 2.4.

Таблиця 2.4 – Результати порівняння характеристик якості: R^2 , $MMRE$ та $PRED(25)$ лінійної регресії у припущенні про нормальність емпіричних даних та нелінійної регресії, отриманої за допомогою нормалізації десятиковим логарифмом

Модель	Коефіцієнт детермінації, R^2	Середня величина відносної похибки, $MMRE$	Рівень передбачення, $PRED(25\%)$
Лінійна	0,8488	0,5821	0,3333
Нелінійна	0,7546	0,2842	0,6333

Із таблиці видно, що R^2 має прийнятні значення для лінійної та нелінійної регресій, а значення $MMRE$ та $PRED(0,25)$ значно кращі для нелінійної регресії.

На рис. 2.3 зображені лінійна регресія, довірчий інтервал та інтервал передбачення для ненормалізованих даних.

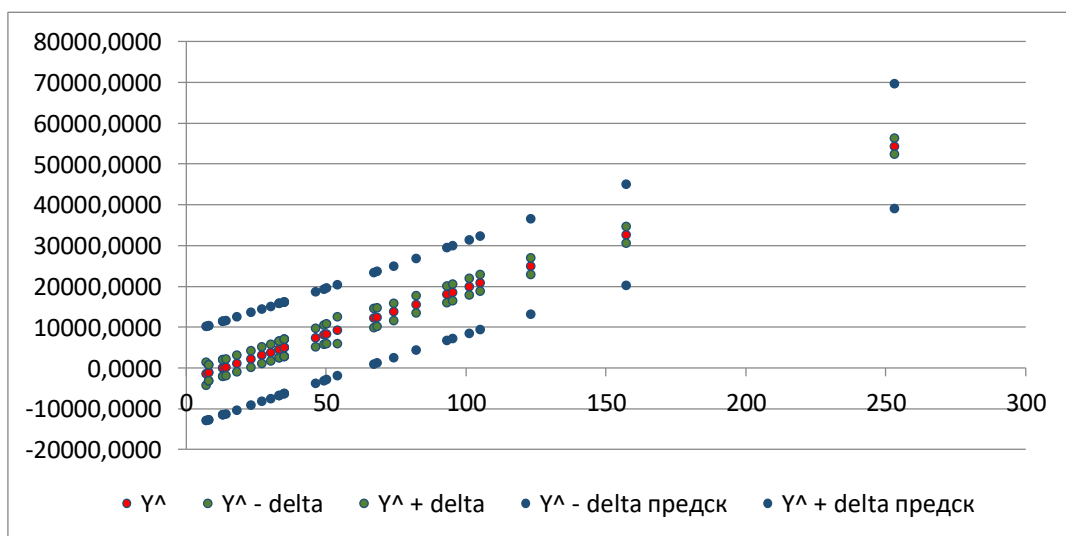


Рисунок 2.3 – Лінійна регресія, довірчий інтервал та інтервал передбачення для ненормалізованих даних

На рис. 2.4 зображені нелінійна регресія, довірчий інтервал та інтервал передбачення для даних, нормалізованих за допомогою десятичного логарифму.

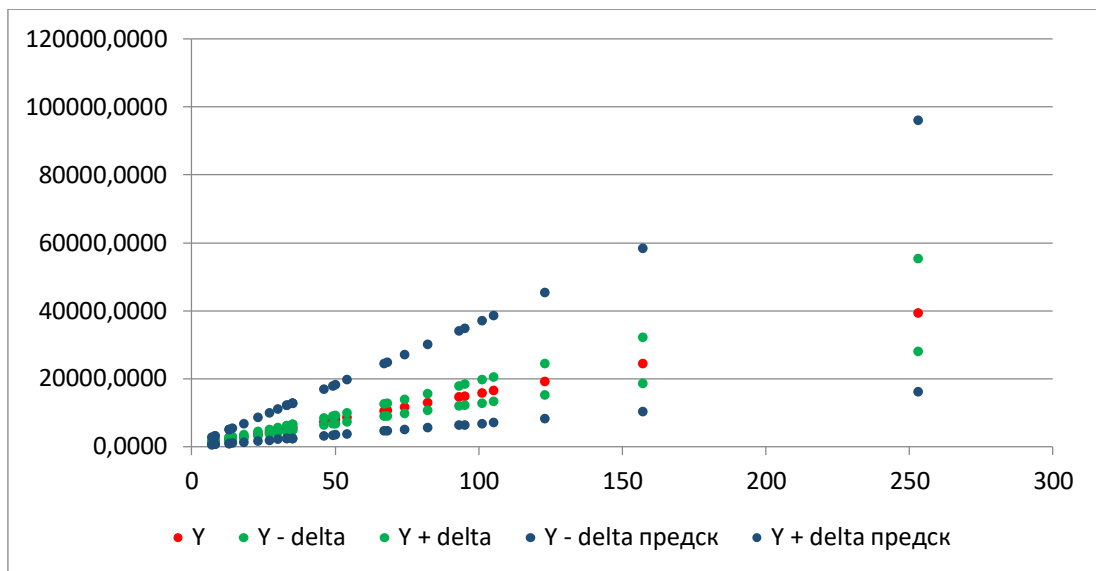


Рисунок 2.4 – Нелінійна регресія, довірчий інтервал та інтервал передбачення для даних, нормалізованих за допомогою десятичного логарифму

Враховуючи все вищеперераховане можна зробити висновок, що для вибраних вихідних даних результат прогнозування розміру динамічних web-додатків на РНР з допомогою нелінійної регресії (нормалізуюче перетворення десятичний логарифм) більш ефективний, ніж для ненормалізованих даних.

2.3 Постановка задачі на розробку програми для оцінювання розміру динамічних web-додатків на РНР

Побудувавши нелінійну регресійну модель для оцінювання розміру динамічних web-додатків на РНР та дослідивши її характеристики, сформулюємо постановку задачі: розробити програмний проект для

реалізації побудованої нелінійної регресійної моделі оцінювання розміру динамічних web-додатків на РНР.

Вимоги до складу виконуваних функцій

В розроблюваному програмному забезпеченні повинні бути реалізовані наступні функції:

- 1) введення емпіричних даних;
- 2) нормалізація емпіричних даних з використанням нормалізуючого перетворення десятиковий логарифм;
- 3) побудова нелінійної регресійної моделі та аналіз результатів;
- 4) побудова довірчих та прогнозних інтервалів для нелінійної регресійної моделі;
- 5) прогнозування розміру динамічних web-додатків на РНР.

Для побудови нелінійної регресійної моделі необхідно використати 30 динамічних web-додатків на РНР, які не містять викидів.

Детальні вимоги до програми наведені у додатку А.

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP

Розробка проекту програмного забезпечення включає ескізне, технічне та робоче проектування. Проект був розроблений з використанням об'єктно-орієнтованої методології та мови моделювання UML.

3.1 Ескізний проект програмного забезпечення для оцінювання розміру динамічних web-додатків на PHP

Ескізний проект представлений моделлю програмного забезпечення, що містить діаграму варіантів використання, специфікації варіантів використання, діаграми діяльності, що використовуються для формалізації представлення сценарію використання, проект користувальницького інтерфейсу, а також інформаційну модель (концептуальну модель даних предметної галузі).

3.1.1 Діаграма варіантів використання програмного забезпечення

Для правильного розуміння і проектування програмної системи необхідно визначити, що вона буде робити, іншими словами, необхідно розробити функціональну модель. Одним із перших кроків при визначенні функціональності є розробка проекції, яка називається функціональною моделлю.

Розглянемо моделювання вимог до програмної системи з допомогою діаграми варіантів використання. Діаграма варіантів використання — це діаграма UML за допомогою якої в графічному вигляді можна відобразити вимоги до системи, що розробляється.

Перед побудовою діаграми, в результаті аналізу предметної галузі, виявили акторів та варіанти використання системи, що проектується. Актора в даній системі назвали «Користувач». Через нього буде виконуватися вся робота з системою. Варіантами використання програмного забезпечення будуть наступні: «Введення емпіричних даних», «Побудова та аналіз нелінійної регресії», «Побудова довірчих та прогнозних інтервалів», «Прогнозування розміру», «Нормалізація емпіричних даних».

Діаграма варіантів використання програмного забезпечення для оцінювання розміру динамічних web-додатків на РНР представлена на рисунку 3.1.

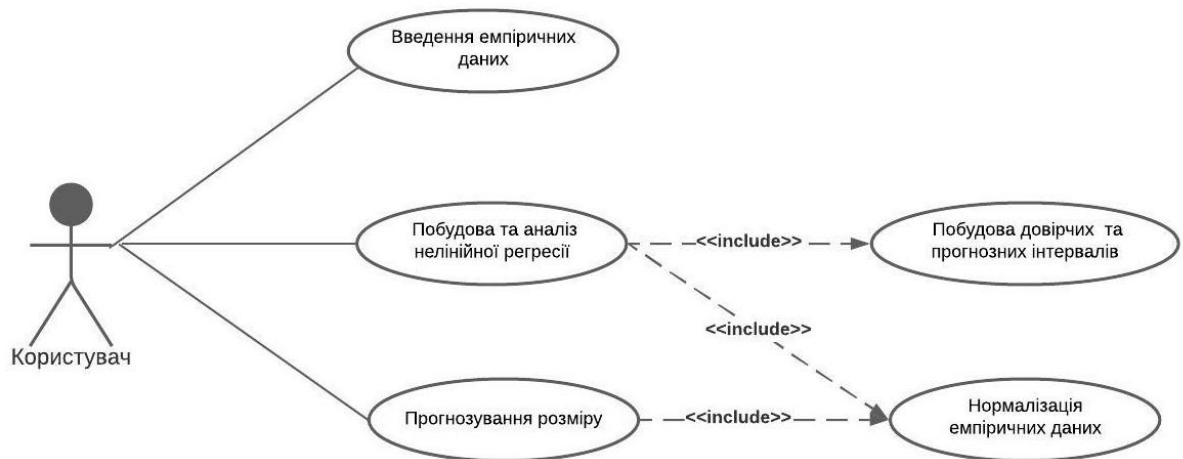


Рисунок 3.1 – Діаграма варіантів використання програмного забезпечення для оцінювання розміру динамічних web-додатків на РНР

Розглянемо специфікації варіантів використання програмного забезпечення для оцінювання розміру динамічних web-додатків на РНР

3.1.2 Специфікації варіантів використання ПЗ для оцінювання розміру динамічних web-додатків на РНР

Специфікація прецеденту «Введення емпіричних даних» представлена в табл. 3.1.

Таблиця 3.1 – Специфікація прецеденту «Введення емпіричних даних»

Назва елементу	Опис елементу
Короткий опис	Вводяться емпіричні дані з файлу
Передумови	Відсутні
Основний потік	Після натискання на кнопку «Введення емпіричних даних» обирається файл з даними, вводяться дані та перевіряється нормальність їх розподілу
Альтернативні потоки	У разі невірної форми файлу видається повідомлення про помилку і програма завершує свою роботу
Постумови	Відсутні

В таблиці 3.2 представлена специфікація прецеденту «Нормалізація емпіричних даних».

Таблиця 3.2 – Специфікація прецеденту «Нормалізація емпіричних даних»

Назва елементу	Опис елементу
Короткий опис	Виконується нормалізація введених емпіричних даних
Передумови	Успішне виконання прецеденту «Введення емпіричних даних»
Основний потік	Нормалізація емпіричних даних з використанням нормалізуючого перетворення у вигляді десяткового логарифму
Альтернативні потоки	Відсутні
Постумови	Відсутні

В таблиці 3.3 представлена специфікація прецеденту «Побудова та аналіз нелінійної регресії».

Таблиця 3.3 – Специфікація прецеденту «Побудова та аналіз нелінійної регресії»

Назва елементу	Опис елементу
Короткий опис	Будується нелінійна регресія та розраховуються показники її якості
Передумови	Успішне виконання прецеденту «Введення емпіричних даних»
Основний потік	Виконується нормалізація емпіричних даних. Обчислюються параметри лінійної регресії та знаходяться її розрахункові значення. З використанням зворотного перетворення будується нелінійна регресія та знаходяться її розрахункові значення. Знаходяться коефіцієнт детермінації, середня величина відносної похибки та рівень прогнозування для нелінійної регресії. Показники виводяться на екран. Обчислюються довірчі та прогнозні інтервали нелінійної регресії.
Альтернативні потоки	Відсутні
Постумови	Відсутні

В таблиці 3.4 представлена специфікація прецеденту «Побудова довірчих та прогнозних інтервалів».

Таблиця 3.4 – Специфікація прецеденту «Побудова довірчих та прогнозних інтервалів»

Назва елементу	Опис елементу
Короткий опис	Обчислюються довірчі та прогнозні інтервали регресії
Передумови	Успішне виконання прецеденту «Введення емпіричних даних». Побудова регресії.
Основний потік	Обчислюються та виводяться на екран довірчі та прогнозні інтервали регресії
Альтернативні потоки	Відсутній
Постумови	Відсутні

В таблиці 3.5 представлена специфікація прецеденту «Прогнозування розміру».

Таблиця 3.5 – Специфікація прецеденту «Прогнозування розміру»

Назва елементу	Опис елементу
Короткий опис	Виконується прогнозування розміру динамічного web-додатку на РНР
Передумови	Побудова моделі регресії
Основний потік	Введення даних для прогнозування, обчислення прогнозу та представлення результату на екрані
Альтернативні потоки	Відсутній
Постумови	Відсутні

3.1.3 Діаграма діяльності ПЗ для оцінювання розміру динамічних web-додатків на РНР

Діаграми діяльності (Activity Diagrams) представляють алгоритми деяких процесів системи.

Діаграми діяльності представляють переходи потоку керування від однієї діяльності до іншої. Таким чином, можна сказати, що Activity Diagrams являються різновидом діаграми станів, в якій всі або більша частина станів є деякими діяльностями, а всі або більша частина переходів спрацьовують при завершенні певної діяльності та дозволяють перейти до виконання наступної. Діаграму діяльності можемо приєднати до будь-якого елементу моделі, що має динамічну поведінку, тому логічніше говорити не «діаграма діяльності», а «діаграми діяльностей» – у множині.

На рисунку 3.2 зображена діаграма діяльності для варіанту використання «Побудова та аналіз нелінійної регресії» ПЗ для оцінювання розміру динамічних web-додатків на РНР.

На рисунку 3.3 зображена діаграма діяльності для варіанту використання «Побудова довірчих та прогнозних інтервалів» ПЗ для оцінювання розміру динамічних web-додатків на РНР.



Рисунок 3.2 – Діаграма діяльності для варіанту використання «Побудова та аналіз нелінійної регресії» ПЗ для оцінювання розміру динамічних web-додатків на РНР



Рисунок 3.3 – Діаграма діяльності для варіанту використання «Побудова довірчих та прогнозних інтервалів» ПЗ для оцінювання розміру динамічних web-додатків на РНР

3.1.4 Концептуальна модель даних ПЗ для оцінювання розміру динамічних web-додатків на PHP

При побудові моделі предметної галузі насамперед необхідно виявити абстракції, які існують в реальному світі та належать до предметної галузі, що моделюється. На цьому етапі розробляється концептуальна модель даних. Ця модель відображає семантику предметної галузі як сукупність понять (сутностей), їх характеристик (атрибутів) та зв'язків (асоціативних відносин між сутностями).

Для моделювання предметної галузі на концептуальному рівні можна застосовувати діаграму "сутність-зв'язок" (також ERD або ER-діаграму).

Метою ER-діаграм є показати зв'язки та відношення між елементами, тому вони відображають лише реляційну структуру. Дані мають бути чітко розбиті на поля, стовпці та рядки, інакше немає сенсу застосовувати ER-діаграми. Застосування ER-моделей для інтеграції з існуючою базою даних – непросте завдання через різницю в архітектурах.

Концептуальна модель даних програми для оцінювання розміру динамічних web-додатків на PHP представлена у вигляді ER-діаграми на рисунку 3.4.

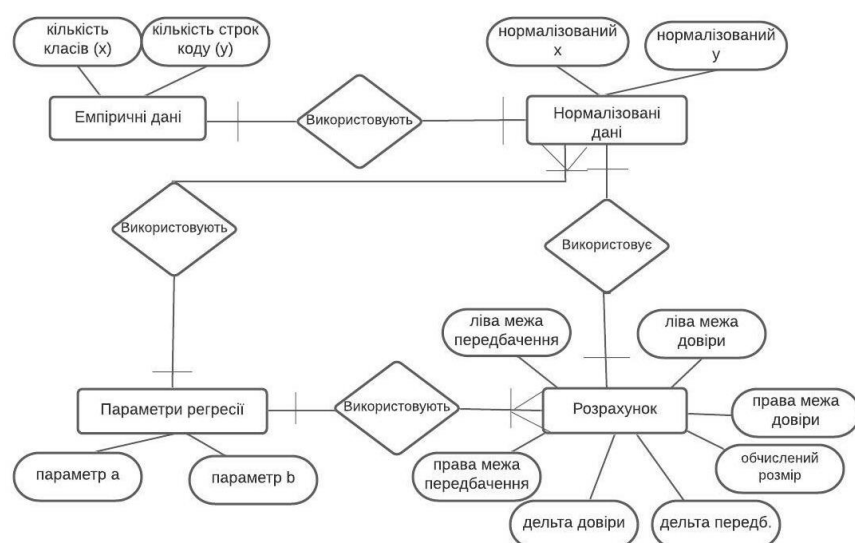


Рисунок 3.4 – Концептуальна модель даних програмного забезпечення для оцінювання розміру динамічних web-додатків на PHP

3.1.5 Інтерфейс користувача ПЗ для оцінювання розміру динамічних web-додатків на РНР

Інтерфейс – це «мова спілкування», яку розуміють обидва об'єкти, які взаємодіють один з одним з метою вирішити певне питання. Якщо програма – це помічник, то інтерфейс – це спосіб спілкуватися (взаємодіяти) з нею.

Він визначає взаємодію людини з розробленими програмами (додатками), що працюють під її управлінням. Більшість додатків використовують графічний інтерфейс користувача. При цьому для екранного відображення введення та виведення команд користувача та даних використовуються вікна. Елементи управління програмою зображуються графічно всередині вікон (у вигляді меню, кнопок, полів введення та ін.). Вибір користувачем одного з елементів може бути зроблений за допомогою миші, клавіатури або дотику до екрана (якщо сенсорний екран). Програма може робити виведення результатів обробки даних як тексту, гіпертексту, таблиць, діаграм, відео та інших. Стандартисткість графічних елементів управління полегшує процес освоєння нового програмного забезпечення.

Враховуючи склад виконуваних функцій, був розроблений проект інтерфейсу. На рисунку 3.5 представлено ескіз головного вікна програми.

Як бачимо, користувачу доступні такі функції:

- обрати файл;
- розрахувати;
- вивести інтервали;
- прогнозувати;
- зберегти;
- вийти з програми.

The image shows a software interface with a light gray background. At the top, there are four buttons: "Выбрать файл", "Рассчитать", "Вывести интервалы", and "Прогнозировать". Below these buttons, there are three input fields: "R^2=", "MMRE=", and "PRED(0,25)=". In the center of the interface is a large rectangular area divided into two equal vertical panels. At the bottom, there are two buttons: "Сохранить" and "Выйти".

Рисунок 3.5 – Проект головного вікна програмного забезпечення для оцінювання розміру динамічних web-додатків на PHP

3.2 Технічний проект ПЗ для оцінювання розміру динамічних web-додатків на РНР

В рамках технічного проекту були розроблені статична та динамічна моделі системи.

3.2.1 Статична модель ПЗ для оцінювання розміру динамічних web-додатків на РНР

В залежності від цілей застосування діаграм класів, існують різні точки зору на їх побудову:

- Концептуальна точка зору – з допомогою діаграми класів описується модель предметної галузі. В цьому випадку діаграма містить тільки класи прикладних об'єктів.

- Точка зору специфікації – діаграму класів використовують при проектуванні інформаційних систем.

- Точка зору реалізації – на діаграмі представляють класи з програмного коду.

Діаграма класів є статичною структурною діаграмою, що описує структуру системи та містить класи системи, їх атрибути, методи та залежності між класами.

На діаграмі класів можуть бути присутні класи, об'єкти, інтерфейси, кооперації та зв'язки між ними.

На рисунку 3.6 зображено діаграму класів ПЗ для оцінювання розміру динамічних web-додатків на РНР.

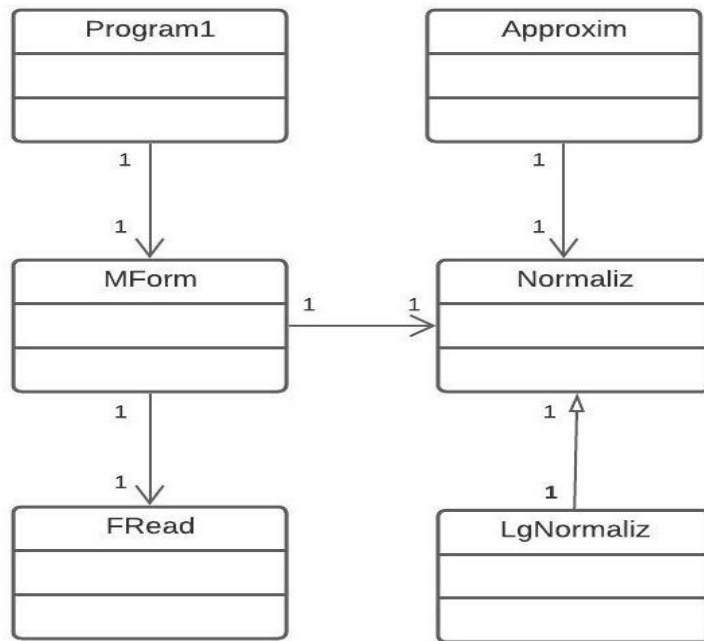


Рисунок 3.6 – Діаграма класів ПЗ для оцінювання розміру динамічних web-додатків на PHP

В програмному забезпеченні визначено 6 класів. Розглянемо призначення кожного з них:

Клас Approxim

Клас призначений для апроксимації функції.

Містить методи для отримання коефіцієнтів регресії.

Клас Program1

Клас призначений для запуску програми

Містить метод для запуску програми.

Клас MForm1

Клас призначений для створення графічного інтерфейсу програми.

Містить методи для обчислення та відображення інтервалів та метрик якості.

Клас FRead

Клас призначений для введення інформації з файлу.

Містить методи для отримання масивів кількості класів, розмірів.

Клас IgNormalizer

Клас призначений для нормалізації даних десятковим логарифмом і для виконання зворотного перетворення.

Містить методи для нормалізації даних і для виконання зворотного перетворення.

Клас Normaliz

Клас, що відповідає за основні розрахунки

Містить методи для визначення довірчого та прогнозного інтервалів, показників якості

3.2.2 Динамічна модель ПЗ для оцінювання розміру динамічних web-додатків на РНР

Взаємодію в розроблюваній системі можна моделювати за допомогою діаграм послідовності і комунікацій.

З допомогою діаграм послідовності виконується уточнення прецедентів, тобто більш детально описується логіка сценаріїв використання. Діаграми послідовності містять об'єкти, які задіяні в рамках сценарію та повідомлення, які передаються між об'єктами. Також на діаграмах іноді представляються результати, що повертаються.

З допомогою діаграми послідовності представляють взаємодію елементів моделі як послідовність повідомлень та відповідних подій на лініях життя об'єктів.

Діаграма послідовності варіанту використання «Введення емпіричних даних» представлена на рисунку 3.7.

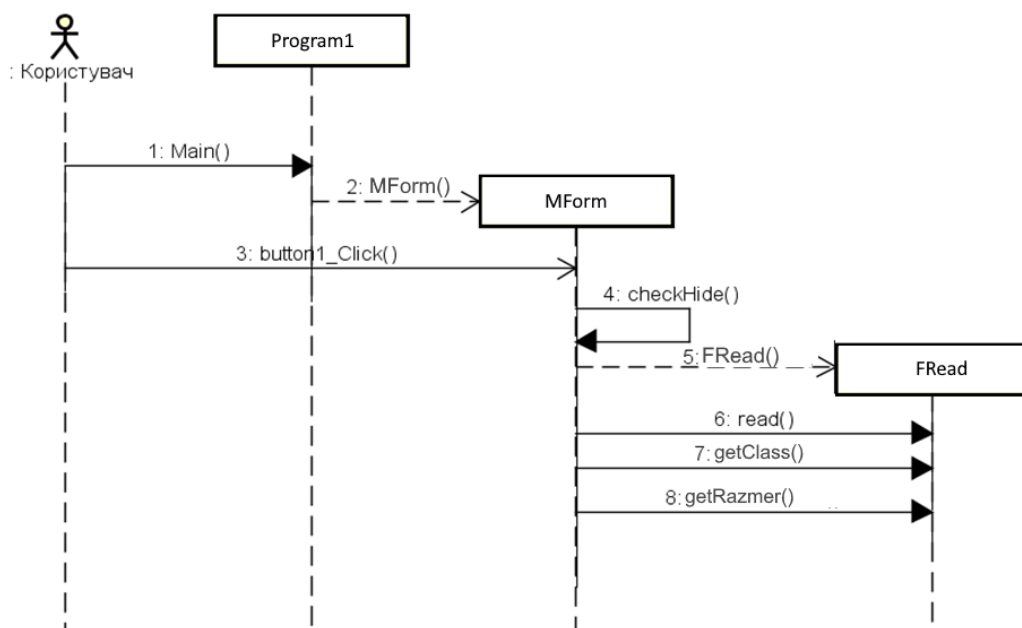


Рисунок 3.7 – Діаграма послідовності варіанту використання «Введення емпіричних даних»

На діаграмі послідовності об'єкти переважно представляють екземпляри класу чи сутності, які мають поведінку. У якості об'єктів можуть виступати користувачі, які ініціюють взаємодію, класи, що мають поведінку в розроблюваній системі, або програмні компоненти, а іноді й системи в цілому.

3.3 Робочий проект програмного забезпечення для оцінювання розміру динамічних web-додатків на РНР

На виході з етапу робочого проектування повинні бути у наявності програмна документація та програмний продукт.

Основними видами робіт, які виконуються на стадії робочого проекту

В робочому проекті зазвичай виконуються наступні роботи:

- кодування та тестування;
- налагодження;
- узгодження та затвердження порядку проведення випробувань програмного забезпечення;
- розробка програмних документів;
- випробування програмного забезпечення.

3.3.1 Обґрунтування вибору мови програмування та засобів розробки програмного забезпечення

У якості мови програмування була обрана мова об'єктно-орієнтована мова C#. Розроблена мова C# в компанії Microsoft. Ця мова програмування має потужну підтримку та визнана міжнародною спільнотою [24].

Мова C# інтенсивно розвивається, з кожною версією включаючи все нові можливості.

Мова C# є мовою компонентного програмування, тобто вона надає можливість повторного використання розроблених раніше компонентів. Це являється однією з головних переваг C#. При цьому компоненти, що створюються компілятором, є документованими і крім коду містять інформацію, яка описує компоненти. Тому компоненти можуть виконуватися на різних платформах.

Розглянемо переваги C# [24, 25]:

- C# створювалася та розвивається разом з каркасом Framework .Net і враховує всі можливості FCL та CLR.
- C# – об'єктно-орієнтована мова програмування. В ній і вбудовані типи представлені класами.
- В C# реалізовані спадкування та універсалізація.
- C# має спільний з C/C++ синтаксис та схожі оператори та перейняла їх найкращі елементи. Тому перехід від C/C++ до C# досить простий.
- Це обумовлене тим, що в C# хоч і допускаються, проте не рекламуються покажчики, адресація, розіменування, адресна арифметика, які являються досить небезпечними.
- Через каркас Framework.Net програмісти C# мають для роботи з віртуальною машиною такі ж самі переваги, що й програмісти Java. При цьому код, який отримується, навіть більш ефективний. Підвищення ефективності пов'язане з тим, що середовище CLR є компілятором проміжної мови, а віртуальна Java-машина – інтерпретатором байт-коду.
- Бібліотека каркасу надає можливість легко реалізовувати різні типи додатків, Web-служб, інші види компонентів, легко зберігати та отримувати інформацію з бази даних.
- Надійний та ефективний код C# є одним із головних факторів, що забезпечують успіх C#.

Виділення та об'єднання найкращих ідей сучасних мов програмування робить мову C# не просто сумою їх переваг, а мовою програмування нового покоління.

3.3.2 Діаграма компонентів ПЗ для оцінювання розміру динамічних web-додатків на PHP

Діаграма компонентів призначена для визначення програмних компонентів розроблюваної системи, в якості яких можуть бути вихідні, бінарні та виконувані коди, і визначити існуючі між ними зв'язки.

З допомогою діаграм компонентів можна представляти:

- специфікацію вихідного коду програмного забезпечення;
- специфікацію виконуваного варіанта програмного забезпечення.

З допомогою діаграми компонентів здійснюється перехід від логічного уявлення розроблюваного програмного забезпечення до фізичного у вигляді програмних компонентів. Причому одна частина програмних компонентів може існувати тільки на стадії компіляції, а інша – на стадії виконання. Діаграма компонентів може містити компоненти, класи, інтерфейси та зв'язки між ними. Компонент представляє фізичну складову розроблюваної системи. Компоненти, які є файлами з вихідним кодом класів, бібліотеками, модулями тощо повинні мати узгоджений набір інтерфейсів.

На рисунку 3.8 представлено діаграму компонентів програмного забезпечення для оцінювання розміру динамічних web-додатків на PHP.

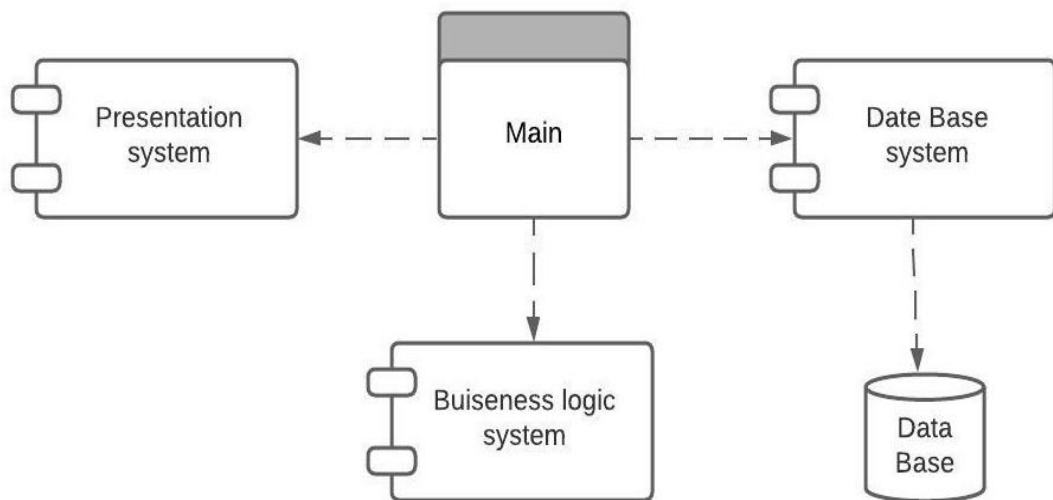


Рисунок 3.8 – Діаграма компонентів ПЗ для оцінювання розміру динамічних web-додатків на PHP

3.3.3 Кодування та тестування ПЗ для оцінювання розміру динамічних web-додатків на РНР

В якості мови програмування була обрана мова С# як сучасна об'єктно-орієнтована мова програмування. Текст програми представлений в додатку Б.

Основними методиками тестування є функціональне тестування та тестування по специфікації. Так як тестування по специфікації більш складне та вимагає більше часу, то в роботі проводилося функціональне тестування.

Розглянемо більш детально функціональне тестування, точніше його техніки (методи) [N]:

- класи еквівалентності;
- граничні значення;
- таблиця прийняття рішень;
- попарне тестування;
- діаграма переходів станів;
- варіанти використання (Use Case Testing).

Розглянемо техніку Use Case Testing.

Use case – це сценарії, що описують те, як actor (зазвичай людина, але може бути й інша система) користується системою для досягнення певної мети. Варіанти використання описуються з погляду користувача, а не системи. Внутрішні роботи з підтримки працездатності системи не є частиною варіанта використання.

Хоча б один тест-кейс повинен перевіряти основний сценарій і хоча б по одному кейсу має припадати на альтернативні сценарії.

При тестуванні об'єктно-орієнтованого ПЗ найменшим тестованим елементом є клас.

Для спрощення розробки тестів використовуються елементи Use Case, що є частиною моделі вимог. Кожен елемент Use Case задає сценарій, який дозволяє виявляти помилки при взаємодії користувача з системою. Для

підтвердження правильності може приводитися звичайне тестування «чорного ящика».

Як приклад розглянемо два тести для Use Case Testing.

Тест 1 – Перевірити неможливість ввести файл з даними неправильного формату

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
- Відкривається вікно в якому користувач може вибрати шлях до файлу.
- Користувач обирає файл з даними, що мають неправильний формат.
- Біля кнопки з'являється назва файлу.
- На екрані з'являється повідомлення про невідповідність формату даних.

Тест 2 – Перевірити можливості додавання даних до програми

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
- Відкривається вікно в якому користувач може вибрати шлях до файлу.
- Користувач обирає файл з даними, що мають правильний формат.
- Біля кнопки з'являється назва файлу.
- Програма продовжує роботу.

Аналогічно в роботі були протестовані всі функції системи. Помилки не було виявлено.

3.3.4 Проведення випробування ПЗ оцінювання розміру динамічних web-додатків на РНР

Випробування є одним із найважливіших елементів управління якістю програмного забезпечення. При випробуванні експериментально визначаються кількісні та якісні показники програмного забезпечення під час його роботи у реальному чи модельованому середовищі. На основі отриманих показників необхідно зробити висновок про можливість використовувати розроблене програмне забезпечення за призначенням. При негативному висновку програмне забезпечення підлягає доопрацюванню.

Перед випробуванням проводиться налагодження програми. Тестування являється основним його методом. Завдання тестування – виявлення помилок, завдання налагодження – виявлення та усунення помилок. Проте не можливо зупинитися на налагодженні ПЗ, навіть у випадку впевненості в усуненні всіх помилок. Налагодження та випробування переслідують різні цілі. Навіть якщо налагодження виконано повністю, розроблене програмне забезпечення може бути непридатним для застосування за призначенням. Замінити випробування не може і перевірка роботи програми на окремих прикладах. Тому що на інших прикладах програма може бути непрацездатною. А якщо спробувати перевірити програму на всіх можливих вхідних даних, які передбачаються функціональністю, то отримаємо теж саме випробування.

Програма і методика випробувань розробленого програмного забезпечення приводиться в Додатку Г.

Тести «чорної скриньки» пишуться із використанням специфікації тестового програмного забезпечення. У результаті опрацювання специфікації до програмного забезпечення, що підлягає тестуванню, було отримано ряд тестів «чорної скриньки». Всі тести пройшли успішно. Випробування показало відповідність розробленого програмного забезпечення вимогам, які сформульовані в технічному завданні.

Випробування функції «Вивести інтервали» представлено на рисунку 3.9. При натисканні на кнопку «Показати інтервали» на екрані з'являються розраховані довірчий інтервал та інтервал прогнозування.

The screenshot shows a software interface with four buttons at the top: «Выбрать файл», «Рассчитать», «Удалить интервалы», and «Прогнозировать». Below the buttons, three metrics are displayed: $R^2 = 0.7546$, $MMRE = 0.2842$, and $PRED(0,25) = 0.6333$. A table with two columns, «Доверительный» and «Прогнозирования», displays numerical data for five rows. The table has a vertical scrollbar on the right.

Доверительный		Прогнозирования	
804.6979	1549.7964	459.2388	2715.6197
936.8586	1735.0354	527.9958	3078.5903
936.8586	1735.0354	527.9958	3078.5903
1622.1298	2626.0220	873.0464	4879.1776
1622.1298	2626.0220	873.0464	4879.1776

Рисунок 3.9 – Довірчі інтервали та інтервали прогнозування

При цьому кнопка змінить назву на «Видалити інтервали». Результат відповідає очікуваному.

Випробування функції «Прогнозувати» представлено на рисунку 3.9.

.Для виконання прогнозування розміру необхідно натиснути на кнопку «Прогнозувати» (рис. 3.10).

The screenshot shows a software interface with four buttons at the top: «Выбрать файл», «Рассчитать», «Вывести интервалы», and «Прогнозировать». Below the buttons, there are input fields for «Классов» (value: 28) and «Строк кода» (value: 4412.1400), with an «Ok» button. A table with two columns, «Доверительный» and «Прогнозирования», displays numerical data for one row.

Доверительный		Прогнозирования	
3747.9296	5194.0621	1893.7807	10279.4266

Рисунок 3.10 – Результат прогнозування розміру

Результат прогнозування розміру програмного забезпечення та його довірчий і прогнозний інтервали.

4 РЕЗУЛЬТАТИ РОЗРОБКИ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP

Мета, яка ставилася на початку кваліфікаційної роботи, була повністю досягнута. Для досягнення мети були вирішені наступні завдання:

- виконано аналіз та порівняння наявних методів та моделей для оцінювання розміру ПЗ;
- обґрунтовано необхідність удосконалення нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP;
- зібрано емпіричні дані для побудови нелінійної регресійної моделі оцінювання кількості строк коду динамічних web-додатків на PHP;
- виконано нормалізацію вихідних даних з використанням нормалізуючого перетворення десятковий логарифм;
- виконано перед обробка вихідних даних;
- побудовано лінійну регресійну модель оцінювання кількості строк коду динамічних web-додатків на PHP, довірчі інтервали та інтервали прогнозування для даних без викидів;
- побудовано нелінійну регресійну модель оцінювання кількості строк коду динамічних web-додатків на PHP, довірчі інтервали та інтервали прогнозування для емпіричних даних через зворотне перетворення;
- побудовано лінійну регресійну модель оцінювання кількості строк коду динамічних web-додатків на PHP, довірчі інтервали та інтервали прогнозування для емпіричних даних;
- виконано порівняння лінійної та нелінійної регресійних моделей оцінювання кількості строк коду динамічних web-додатків на PHP;
- розроблено ПЗ для прогнозування кількості строк коду динамічних web-додатків на PHP.

Також була розроблена наступна документація:

- технічне завдання;
- текст програми;

- програма і методика випробувань;
- опис програми;
- інструкція користувача.

Для роботи ПЗ необхідна наявність ПЕОМ з наступними мінімальними характеристиками:

- оперативна пам'ять 512 МБ;
- дисковий простір – не менше 512 МБ;
- екран – 1280 x 1024 пікселів.

Було виконано тестування та випробування програмного забезпечення, яке показало працездатність ПЗ та відповідність його вимогам, сформульованим в технічному завданні.

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Вступ

Дана робота присвячена розробці програми для оцінювання розміру для динамічних web-додатків на РНР. Її можна використовувати для прогнозування обсягу та вартості робіт в комп'ютерних фірмах.

Ефективність програмних систем визначається ступенем їх позитивного впливу на підприємство, що управляється, або процес в результаті використання засобів обчислювальної техніки, програмного забезпечення, зміни інформаційних потоків та організаційної структури управління.

Створення програмного забезпечення 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 забезпечення характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

5.2 Розрахунок витрат на створення й експлуатацію програми для оцінювання розміру динамічних web-додатків на РНР

Витрати на розробку системи складаються з витрат:

- на заробітну платню розробника;
- на амортизацію ЕОМ, на якій виконується розробка;
- на експлуатацію цієї ЕОМ;
- на засоби розробки;

- на матеріали і комплектуючі.

Розробка програмного забезпечення виконується програмістом, місячна заробітна плата якого складає 10000 грн.

Вартість сучасного ПК становить 25000 грн.

При вартості кіловат-години електроенергії 1.68 грн, розраховується вартість розробки програми.

Витрати на допоміжні матеріали наведені в табл. 5.1.

Таблиця 5.1 – Витрати на допоміжні матеріали

№	Пункти витрат	Сума, грн
1	Флеш-накопичувач даних	320
2	Папір для принтера	90
3	Картридж та тонер для принтера	240
	Разом	650

Вартість ПЗ знаходиться за формулою:

$$C_{\text{пр}} = (Z_{\text{зп}} + Z_{\text{сз}} + Z_{\text{зг}} + Z_{\text{е}}) * T + Z_{\text{м}} \quad (5.1)$$

де T – тривалість розробки, міс.

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

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

$Z_{\text{зг}}$ – загальногосподарські витрати (10% від основної заробітної плати), грн.

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

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

Z_e при споживанні потужності 0.5 кВт, тривалості роботи на місяць, рівної $22 \cdot 8 = 176$ годин, вартості кіловат-години електроенергії 1.68 грн становить:

$$Z_e = 176 \cdot 1.68 \cdot 0.5 = 113.52 \text{ грн}$$

Витрати на розробку ПЗ наведені в табл. 5.2.

Таблиця 5.2 – Витрати на розробку програмного забезпечення

№	Найменування витрат	Одиниця	Кількість
1	Тривалість розробки (Т)	Міс.	3
2	Основна і додаткова заробітна плата ($Z_{зп}$)	Грн.	10000.00
3	Відрахування на соціальні заходи ($Z_{сз}$)	Грн.	1200.00
4	Загальногосподарські витрати ($Z_{зг}$)	Грн.	900.00
5	Витрати на допоміжні матеріали (Z_m)	Грн.	660.00
6	Витрати на електроенергію (Z_e)	Грн.	113.52

За формулою 5.1 розрахуємо вартість програми:

$$C_{пр} = (10000 + 1200 + 900 + 113.52) \cdot 3 + 660 = 37300.56 \text{ грн}$$

Амортизаційні відрахування складають 25% балансової вартості на рік:

$$A_{об} = 15000 \cdot 0.25 = 3750 \text{ грн}$$

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

$$B_m = 15000 \cdot 0.05 = 750 \text{ грн}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_m = 264.5 \cdot T_z \quad (5.2)$$

T_z – це середнє місячне навантаження устаткування (близько 6 годин), 264.5 – середня кількість робочих днів на рік.

Таким чином, річний обсяг роботи ПК складає:

$$\Phi_m = 264.5 \cdot 6 = 1587 \text{ годин}$$

Витрати на електроенергію Z_e при 1587 годинах роботи устаткування на рік становлять:

$$Z_e = 1587 * 1.29 * 0.5 = 1023.62 \text{ грн}$$

Експлуатаційні витрати на ПК на рік складуть:

$$Z_{зр} = 3750 + 750 + 1023.62 = 5523.62 \text{ грн}$$

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

$$Z_p = 37300.56 + 5523.62 = 42824.18 \text{ грн}$$

5.3 Економічна ефективність розробки і впровадження програми для оцінювання розміру динамічних web-додатків на РНР

Основним показником економічної ефективності програмного забезпечення є зменшення трудових витрат та часу на розпізнавання зображень автомобільних номерів, а також ефективне розподілення трудових ресурсів на основі отриманих результатів. Ефективний розподіл ресурсів дає змогу зменшити час тестування, і, відповідно, грошові витрати на роботу спеціалістів відділу забезпечення якості ПЗ.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 1.5 працівника (за експертною оцінкою фахівців підприємства).

Заробітна плата 1.5 працівника в рік складає:

$$Z = 10000 * 12 * 1.5 = 180000 \text{ грн}$$

Річний економічний ефект розраховується за формулою:

$$E_{\text{рік}} = \Delta C_n - E_n * k \quad (5.3)$$

де ΔC_n – вивільнені кошти після впровадження продукту (180000 грн) мінус експлуатаційні витрати (42824.18 грн) = 137175.82 грн;

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації 0.25);

k – одноразові витрати на впровадження продукту (37300.56 грн).

$$E_{\text{рік}} = 137175.82 - 0.25 * 37300.56 = 127850.68 \text{ грн}$$

Термін окупності ПЗ розраховується за формулою:

$$T = k / \Delta C_n = 37300.56 / 137175.82 = 0.27 \text{ року}$$

Отже, термін окупності ПЗ становить приблизно 4 місяці.

5.4 Висновки

У цьому розділі були здійснені розрахунки витрат, економічної ефективності та терміну окупності ПЗ для оцінювання розміру динамічних web-додатків на РНР. Виходячи з розрахунків, впровадження даного програмного забезпечення окупить себе протягом 4 місяців. Таким чином, його розробка є економічно обґрунтованою.

6 ОХОРОНА ПРАЦІ

6.1 Вступна частина

Рівень безпеки людини з розвитком цивілізації постійно зростає. Розвиток науки і техніки, в цілому збільшуючи безпеку життєдіяльності людини, призвів до появи цілого ряду нових проблем. Для дотримання безпеки праці під час виробничого процесу, використовуючи певне обладнання, слід дотримуватись правил техніки безпеки.

Основу правової системи щодо охорони праці та техніки безпеки складають положення, які відповідають Конституції України. Статті 43, 45, 46, 49, 50, 53, 56 і 64 Конституції України гарантують право громадян України на працю, відпочинок, охорону здоров'я, медичну допомогу та страхування, а також у випадку повної, часткової або тимчасової втрати працездатності, втрати годувальника, у старості та в інших випадках.

Законодавчі документи та положення з охорони праці затверджені і видані в різний час Верховною Радою України, Кабінетом Міністрів України, Державним Комітетом України з нагляду за охороною праці. Законодавство про охорону праці складається із Закону України "Про охорону праці", Кодексу законів про працю та інших нормативних актів.

Закон України "Про охорону праці" визначає основні положення щодо реалізації конституційного права громадян на охорону їх життя та здоров'я в процесі трудової діяльності, регулює за участю відповідних державних органів відносини між власником підприємства, установи і організації або уповноваженим органом і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні.

Окремої уваги заслуговує діяльність органів державного нагляду громадського контролю за дотриманням законодавства про охорону праці (розділ 7 і 8 Закону).

Організаційну систему заходів щодо охорони праці і техніки безпеки складають такі заходи, як раціональне компонування цехів, правильне розміщення устаткування і робочих місць, вибір найбільш безпечних способів ведення технологічних процесів з обліком новітніх науково-технічних досягнень. Крім вище переліченого організаційні заходи включають в себе управління охороною праці на підприємстві.

Об'єктом управління в системі управління охороною праці є діяльність функціональних служб і структурних підрозділів конкретних керівників та інженерно-технічних робітників підприємств з метою забезпечення безпечних, нешкідливих та сприятливих умов праці на робочих місцях, виробничих ланках, у цехах і на підприємстві в цілому.

6.2 Аналіз шкідливих та небезпечних факторів при роботі з комп'ютером у офісі комп'ютерної фірми

Усе різноманіття законодавчих актів, заходів і засобів, включених у поняття охорони праці, спрямовано на створення таких умов праці, при яких виключається вплив на працюючих небезпечних і шкідливих виробничих чинників. Небезпечний виробничий чинник – чинник, вплив якого на працюючого в певних умовах призводить до травми або раптового різкого погіршення здоров'я. Шкідливий виробничий чинник – чинник, вплив якого на працюючого в певних умовах призводить до захворювання або зниження працездатності. У залежності від рівня і тривалості впливу шкідливий виробничий чинник може стати небезпечним.

Питання, що відносяться до забезпечення охорони праці при роботі за комп'ютером, регулюються правилами, затвердженими Держнаглядохоронпраці від 10 лютого 1999 р. №21. Ці правила внесені в

«Закон про охорону праці». Відповідальність за недотриманням вимог законодавства до умов праці несе роботодавець, що покладає ці функції на службу охорони праці організації або на притягнутого на договірних засадах фахівця з охорони праці.

При роботі на персональному комп'ютері оператори стикаються з впливом таких небезпечних фізичних і психологічних чинників як – підвищена температура зовнішнього середовища, недостатня освітленість робочої зони, відсутність або нестача природного світла, електричний струм, статична електрика, розумова перенапруга, перенапруга очей, монотонність праці, емоційні перевантаження.

Велику небезпеку становлять дисплеї з електронно-променевими трубками (ЕПТ) – джерело шкідливих випромінювань, що небезпечно впливають на здоров'я операторів.

При роботі монітора виникають два типи випромінювань: електростатичне і електромагнітне. Навколо електростатично зарядженого монітора підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками відхиляючої системи, що розташовуються поряд цокольної частини ЕПТ. Результати медичних досліджень показують, що такий електризований пил може викликати запалення шкіри, а електромагнітне випромінювання може призводити до зниження загальної продуктивності оператора.

Розглянемо шкідливі та небезпечні фактори при роботі людини на ЕОМ.

Недостатня освітленість робочого місця.

Розглядаючи такий фактор безпеки, як недостатня освітленість, можна сказати, що від того наскільки правильно буде організована освітленість, залежатиме не тільки подальша робота, але й здоров'я людини. Втома органів зору залежить від ступеня напруженості процесів, що супроводжують зорове сприйняття. При роботі оператора ЕОМ одним з основних навантажень на організм є навантаження на зір, тому дуже важливо

приділити особливу увагу освітленості робочого місця.

У залежності від джерела світла виробниче освітлення може бути трьох видів: природним, штучним, сполученим.

Правильна освітленість робочого місця сприяє високій працездатності користувача ЕОМ і зводить до мінімуму шкідливий вплив на органи зору. Освітленість робочого місця, що рекомендується для користувача ЕОМ, складає 400 – 750 лк.

Вплив електромагнітного й електростатичного випромінювань.

При роботі ЕОМ доводиться зіштовхуватись з електростатичним й електромагнітним випромінюваннями. Навколо електростатично зарядженого монітору підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками. Результати медичних досліджень показують, що електризований пил викликає запалення шкіри, а електромагнітне випромінювання приводить до порушення біологічних процесів (порушення сну, зниження кров'яного тиску, уповільнення серцевої діяльності).

Вплив електричного струму

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпечності.

Людина відчуває дратівну дію змінного струму промислової частоти силою 0,6–1,5 мА і постійного струму 5–7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Сила невідпускаючого змінного струму промислової частоти знаходиться в межах 6–20 мА. Постійний струм, який не викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, становить 15–80 мА і більше.

Вплив шуму.

Шум являє собою безладне сполучення звуків різної інтенсивності і частоти. Шум викликає труднощі з розпізнанням колірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5–12% продуктивність праці. Тривалий вплив шуму з рівнем звукового тиску 90дБ знижує продуктивність праці на 30–60 % .

Робота ЕОМ може супроводжуватися шумами, викликаними деякими рухливими пристроями усередині комп'ютера, а також підключеними до нього. До таких пристроїв можна віднести вентилятори, пристрої читання дисків, принтери. Виникаючий шум, у залежності від інтенсивності, може привести до підвищення нервової напруги і зниження працездатності.

Несприятливі умови мікроклімату.

Мікрокліматичні параметри (температура, відносна вологість і швидкість руху повітря) впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки.

При підвищеній температурі і вологості в приміщенні відбувається перегрівання тіла, що погіршує самопочуття і працездатність. Варто враховувати, що ЕОМ, які знаходяться в приміщенні, сприяють збільшенню температури.

Для комфортної роботи приймається температура 23 (± 2)°C та 18 (± 2)°C в теплий і холодний час року відповідно, при відносній вологості 55 (± 5)%.

Нераціональне розташування устаткування і неправильна організація робочого місця.

Робоче місце – це частина простору, у якому користувач ПК здійснює трудову діяльність і проводить велику частину робочого часу. Важливу роль грає планування робочого місця, яке повинне задовольняти вимогам зручності виконання робіт і економії енергії і часу користувача, зручності обслуговування ЕОМ, дотримання правил охорони праці. При плануванні

робочого місця необхідно враховувати зони досяжності при розташуванні моніторів, клавіатури, маніпулятора типу «миша». Ці зони встановлюються на підставі антропологічних даних людського тіла.

6.3 Розрахунок системи штучного освітлення в офісі комп'ютерної фірми

До сучасного виробничого освітлення пред'являються високі вимоги як гігієнічного, так і техніко-економічного характеру. Правильно спроектоване і виконане освітлення забезпечує не тільки високий рівень працездатності, але й має позитивний психологічний вплив на працюючих, сприяє підвищенню продуктивності праці.

До систем освітлення висувають наступні вимоги:

- відповідність рівня освітленості робочих місць характеру виконуваної зорової роботи;
- досить рівномірний розподіл яскравості на робочих поверхнях і в навколишньому просторі;
- сталість освітленості в часі;
- оптимальна спрямованість випромінюваного освітлювальними приладами світлового потоку;
- довговічність, економічність, електробезпечність та пожежобезпечність, естетичність, зручність і простота експлуатації.

У тих випадках, коли одного природного освітлення в приміщенні недостатньо, створюють поєднане освітлення. При цьому додаткове штучне освітлення застосовують не тільки в темний, але й у світлий час доби (в залежності від виду виконуваної роботи).

По конструктивному виконанні штучне освітлення може бути загальними і місцевим. При загальному освітленні всі робочі місця одержують освітлення від загальної освітлювальної установки. Комбіноване освітлення поряд із загальним включає місцеве освітлення, яке зосереджує

світловий потік безпосередньо на робочих місцях. Застосування тільки місцевого освітлення неприпустимо, тому що при такому застосуванні виникає необхідність часткої адаптації зору, створюються глибокі і різкі тіні та багато інших несприятливих чинників. Для штучного освітлення приміщень використовують люмінесцентні лампи. Використання люмінесцентних ламп пов'язано з тим, що в них наявна висока світлова віддача і тривалий термін служби.

Розрахунок штучного освітлення будемо виконувати методом коефіцієнта використання світлового потоку.

Розрахуємо потрібну кількість світильників для приміщення з комп'ютерами при загальному рівномірному освітленні. Маємо наступні дані: довжина приміщення $A = 7$ м, ширина приміщення $B = 5$ м, висота приміщення $H = 3$ м. Висота робочої поверхні $h_p = 1$ м. Стеля обладнається світильниками з люмінесцентними лампами денного світла ЛД40. Коефіцієнт відбиття стелі $S_p = 70\%$, стін $S_c = 50\%$, робочої поверхні $S_r = 10\%$. Напруга мережі 220В.

Відстань від стелі до робочої поверхні: $H_0 = H - h_p = 3 - 1 = 2$ м;

Відстань від стелі до світильника: $h_c = 0,2 * H_0 = 0,2 * 2 = 0,4$ м;

Висота підвісу світильника над освітлюваною поверхнею:

$h = H_0 - h_c = 2 - 0,4 = 1,6$ м;

Висота підвісу світильника над підлогою: $H_p = h + h_p = 1,6 + 1 = 2,6$ м;

Для досягнення найбільшої рівномірності освітлення приймаємо відношення $L/h = 1,5$. Тоді відстань між центром і світильником: $L = 1,5 * h = 1,5 * 1,6 = 2,4$ м;

Потрібна кількість світильників: $N = S / L^2 = (7 * 5) / 2,4^2 = 6,076$ світильників;

Отже, потрібна кількість світильників дорівнює шести (6 світильників (2 ряди по 3 світильники));

Індекс приміщення:

$i = (A * B) / (h * (A + B)) = (7 * 5) / (1,6(7 + 5)) = 1,82$; (найближче значення в

таблиці 2);

По таблиці коефіцієнтів використання світлового потоку при $i=2$, $S_{п}=70\%$, $S_{с}=50\%$, $S_{р}=10\%$ для світильника типу «ЛД» коефіцієнт використання світлового потоку $\eta=0,59$.

Світловий потік однієї лампи:

$$\Phi = (E_{\min} * S * K_z * Z) / (N * \eta) = (100 * 35 * 1,5 * 1,1) / (6 * 0,59) = 1631 \text{ лм};$$

де K_z – коефіцієнт запасу, який враховує запилення світильників і знос джерел світла в процесі експлуатації; для приміщень, освітлюваних люмінесцентними лампами $K_z=1,5$ за умови чищення світильників не рідше двох разів на рік;

Z – коефіцієнт нерівномірності освітлення ($Z=1,1$);

По знайденому значенню світлового потоку кожної лампи з таблиці, при напрузі мережі 220В, визначаємо її потужність, що має світловий потік 2340 лм, найбільш близькій до розрахункового. При цьому фактична освітленість E_f буде дорівнює:

$$E_f = E_{\min} (\Phi_l / \Phi_p) = 100 * (2340 / 1631) = 143 \text{ лк};$$

$$\text{Загальна потужність світильників: } P_{\text{заг}} = P_l * N = 40 * 6 = 240 \text{ Вт.}$$

6.4 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів у офісі комп'ютерної фірми

Відповідно до «Закону про охорону праці» роботодавець зобов'язаний забезпечити:

- безпеку працівників при експлуатації устаткування;
- застосовування засобів індивідуального захисту працівників;
- відповідні вимоги охорони праці, умови праці на кожному робочому місці;
- дотримання режиму праці і відпочинку працівників;
- навчання безпечним методам і прийомам виконання робіт;
- інструктаж з охорони праці;

- організацію контролю за станом умов праці на робочих місцях;
- проведення атестації робочих місць за умовами праці;
- інформування працівників про умови й охорону праці на робочих місцях, про існуючий ризик ушкодження здоров'я і компенсації при ушкодженні та засоби індивідуального захисту.

Також існують певні вимоги, дотримання яких допоможе зменшити вплив небезпечних і шкідливих чинників. Розглянемо ці вимоги більш конкретно.

Вимоги до освітлення.

У приміщеннях, де експлуатуються комп'ютери, штучне освітлення повинне бути загальним і рівномірним. Однак якщо співробітники переважно працюють з документами, то допускається застосувати комбіноване освітлення. Освітленість поверхні столу в зоні розміщення робочого документа повинна становити 300-500 лк. Джерела освітлення варто встановлювати таким чином, щоб вони не осліплювали, при цьому яскравість світних поверхонь (вікна, світильники та ін.), які розташовуються в полі зору, повинна бути не більш 200 кд/м².

Вимоги до кондиціонування.

У виробничих приміщеннях, у яких установлені комп'ютери, мікроклімат повинен відповідати наступним санітарним нормам:

- температура повітря в теплий період року – не більш 23-25 градусів С, у холодний – 22-24 градуси С;
- відносна вологість повітря – 40-60%;

Для підвищення вологості повітря в приміщеннях варто застосовувати зволожувачі повітря, щодня заправляти їх дистильованою або кип'яченою водою.

Вимоги до розміщення, оснащення й організації робочих місць.

Відстань між робочими столами з моніторами (у напрямку тилу поверхні одного монітора та екрана іншого) повинне бути не менш 2м, а між бічними поверхнями моніторів – не менш 1,2м.

Бажає, щоб висоту робочої поверхні столу можна було регулювати в межах 680-800 мм, а при відсутності такої можливості вона повинна дорівнювати 725 мм.

Екран монітора повинен знаходитись від очей користувача на оптимальній відстані 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків.

Вимоги до безпеки під час експлуатації й обслуговування ЕОМ.

У приміщенні з комп'ютерами повинно проводитися щоденне вологе прибирання. Приміщення повинні оснащатися аптечкою першої допомоги та вуглекислотними вогнегасниками.

Вимоги до режиму праці і відпочинку.

Режими праці і відпочинку при роботі на комп'ютерах залежать від виду і категорії трудової діяльності.

При восьмигодинній робочій зміні і роботі на комп'ютері регламентовані перерви варто встановлювати:

- для I категорії робіт – через 2 години від початку робочої зміни і через 2 години після обідньої перерви тривалістю по 15 хв.;
- для II категорії робіт – через 2 години від початку робочої зміни і через 1,5-2 години після обідньої перерви тривалістю по 15 хв. або через кожну годину роботи тривалістю по 10 хв.

7 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

7.1 Основні положення охорони навколишнього середовища

Охорона навколишнього середовища – це комплекс заходів, призначених для обмеження негативного прояву людської діяльності на живу та не живу природу.

Загальне керування в області охорони навколишнього природного середовища здійснюють Кабінет Міністрів України і місцеві виконавчі і розпорядницькі органи. Спеціально уповноваженими державними органами в області охорони навколишнього середовища і використання природних ресурсів є Міністерство охорони навколишнього природного середовища, його органи на місцях.

Головними нормативно-правовими актами, що регулюють основи організації охорони навколишнього природного середовища, є Закони України: «Про охорону навколишньої природного середовища» від 25 червня 1991 р., «Про охорону атмосферного повітря» від 16 жовтня 1992 р., «Про природно-заповідний фонд України» від 16 червня 1992 р., «Про тваринний світ» від 3 березня 1993 р., «Про карантин рослин» від 30 червня 1993 р. і деякі інші.

Основу правової охорони навколишньої природного середовища складають норми земельного, водного, лісового законодавства, законодавства про надра і т.п.

Охорона навколишнього природного середовища складається з:

- правової охорони, що формулює наукові екологічні принципи у вигляді юридичних законів, обов’язкових для виконання;
- матеріального стимулювання природоохоронної діяльності, що прагне зробити її економічно вигідною для підприємств;

– інженерної охорони, що розробляє природоохоронну і ресурсозбережну технологію та техніку.

Метою охорони навколишнього середовища є: поліпшення загального екологічного стану за допомогою обмеження викидів в атмосферу; зберігання природних комплексів за допомогою створення заповідників та національних парків; зберігання певних видів за допомогою обмеження ловлі риби та полювання; обмеження несанкціонованого викиду сміття.

Величезну тривогу в світі викликає забруднення атмосфери шкідливими газами. Забруднення атмосфери призводить до збільшення озонових «дир» та активізації розвитку парникового ефекту на планеті. Перше явище спричинило зниження захисної дії озонового шару від сонячного ультрафіолетового випромінювання й, як наслідок, – масові захворювання людей і тварин. Парниковий ефект призводить до потепління клімату, танення льодовиків, значного глобального підвищення рівня Світового океану, до змін режиму утворення циклонів і буревіїв, порушення функціонування, навіть деградації екосистем окремих районів суші.

У період промислової й науково-технічної революцій збільшився об'єм викидів в атмосферу газів й аерозолів техногенного походження. Найбільш активними з погляду хімічної взаємодії з компонентами атмосфери й біосфери є сполуки сірки, азоту, фосфору, галогенів, фенолів і формальдегідів. За приблизними даними, щорічно в атмосферу надходять сотні мільйонів тонн оксидів сірки, азоту, галогенопохідних й інших сполук. Тому проблема охорони атмосферного повітря, на теперішній час, дуже важлива. Необхідно дотримуватися заходів щодо зменшення забруднення атмосферного повітря.

7.2 Загальна характеристика забруднення атмосферного повітря

Проблема чистоти атмосфери не нова. Вона виникла разом з появою промисловості й транспорту, що працюють на вугіллі, на нафті. Протягом

практично двох сторіч задимлення повітря носило місцевий характер. Однак швидкий ріст промисловості й транспорту в XX столітті привів до різкого збільшення об'ємів і токсичності викидів. Ці викиди вже не можуть бути «розчинені» в атмосфері до нешкідливих для природного середовища й людини концентрацій. Джерела забруднення атмосфери мають природне й штучне походження(рис.7.1).



Рисунок 7.1 – Джерела забруднення атмосфери

Також джерелами забруднення атмосферного повітря в Україні є енергетика, металургія, вугільна, машинобудівна, хімічна промисловість, сільськогосподарське виробництво, комунально-побутове господарство та транспорт.

7.3 Забруднення атмосферного повітря автотранспортом

Транспорт сьогодні є одним з найбільших джерел забруднення атмосфери. Вихлопні гази двигунів внутрішнього згорання містять велику кількість токсичних сполук: чадного газу, бензопірену, альдегіду, оксиду нітрогену, карбону, вуглеводнів, свинцю. У світі нараховується декілька мільйонів автомобілів, які спалюють велику кількість нафтопродуктів,

істотно забруднюючи атмосферне повітря. У великих містах України шкідливі викиди в повітря внаслідок роботи автотранспорту перевищують 50% загальної кількості.

Встановлено, що найбільша кількість забруднюючих речовин надходить в атмосферу при розгоні автомобіля, при русі з малою швидкістю, при гальмуванні та під час роботи двигуна на холостому ходу.

Кожен легковий автомобіль щорічно виділяє в повітря в середньому 800кг окисі вуглецю, 220кг вуглеводнів й 40кг оксидів азоту.

Становище погіршується тим, що автомобільні викиди концентруються в приземному шарі повітря – саме в зоні нашого дихання. До того ж вітчизняні автомобілі екологічно набагато «брудніші» від багатьох західних моделей: вони витрачають більше палива на 100км шляху, отож дужче забруднюють повітря. Двигуни автомобілів часто бувають погано відрегульованими, тому в їхніх вихлопних газах міститься багато СО, сажі тощо.

Навіть при переході на новий вид автомобільного палива з більш високим октановим числом, що сприяє зниженню вмісту окису вуглецю у вихлопних газах, окисли азоту й свинець залишаються активними компонентами викидів. Різко підвищений вміст свинцю й цинку в ґрунті й рослинності уздовж транспортних магістралей.

Автомобільному транспорту, як джерелу забруднення повітряного середовища, властивий ряд відмінних ознак:

По-перше, чисельність автомобілів у великих містах швидко збільшується та безупинно росте загальний викид шкідливих продуктів в атмосферу.

По-друге, на відміну від промислових джерел забруднення, прив'язаних до певних площадок, ізольованих від житлової забудови санітарно-захисними зонами, автомобіль – джерело забруднення, яке постійно рухається, негативний вплив якого поширюється на житлові райони, місця відпочинку й т.п.

По-третє, автомобільний викид поширюється на рівні подиху людини, і його розсіювання в умовах міської забудови ускладнене.

І, нарешті, сучасні можливості зниження токсичності вихлопних газів ще не в змозі забезпечити бажаний ступінь чистоти повітряного басейну. Автомобілі внаслідок конструктивних недосконалостей і дефектів експлуатації викидають у повітряний басейн більше 200 хімічних сполук. Забруднення атмосферного повітря автотранспортом у США, Японії й інших розвинених країнах вийшло на перше місце, випередивши промисловість і теплоелектростанції.

Для роботи автомобілей підприємства необхідне різне паливо таке, як бензин, метан, пропан, кожне з яких наносить шкоду атмосферному повітрю. Метан являє собою парниковий газ та утворює сполуки, включаючи газові гідрати. Якщо ступінь дії вуглекислого газу на клімат умовно прийняти за одиницю, то парникова активність метану становить 23 одиниці. В таблиці 7.1 наведені викиди шкідливих речовин за їздовий цикл.

Таблиця 7.1 – Викид шкідливих речовин за їздовий цикл

Викид шкідливих речовин за їздовий цикл			
	Бензин	Метан	Пропан
При «холодному» пуску двигуна			
CO	32,8	26,2	17,4
CH	8,6	5	5,7
NO	9,8	8,2	5,2
CH+NO	18,3	13,2	10,9
При «гарячому» пуску двигуна			
CO	20,6	27,2	19,8
CH	6,4	5,2	8,1
NO	8,2	6,8	4,6
CH+NO	14,6	12	12,7

7.4 Заходи щодо зменшення забруднення атмосферного повітря

Для захисту повітряного басейну від негативного антропогенного впливу у вигляді забруднення його шкідливими речовинами використовують наступні методи захисту повітряного середовища від шкідливих викидів:

- архітектурно-планувальні заходи;
- інженерно-організаційні заходи;
- заходи екологізації виробництв;
- техніко-технологічні заходи очистки викидів;
- організація санітарно-захисних зон.

Коротко розглянемо зазначені заходи.

Архітектурно-планувальні заходи пов'язані з правильним взаємним розміщенням джерел викидів і житлової забудови з урахуванням напрямку вітру, облаштуванням навколо промислових підприємств зелених зон тощо.

Інженерно-організаційні заходи спрямовані на зниження інтенсивності руху автотранспорту. Будівництво об'їзних і окружних доріг навколо міст і населених пунктів, спорудження різнорівневих розв'язок на перехрестях доріг, збільшення висоти димових труб для кращого розсіювання пилогазових викидів в атмосфері.

Екологізація виробництв, а саме, впровадження безвідходних та маловідходних технологій дає змогу значно знизити рівень забруднення атмосфери. Найперспективнішими напрямками є перехід підприємств теплоенергетики з твердого палива на природний газ; використання вторинних енергоресурсів у вигляді гарячої води і гарячих газів.

Техніко-технологічні заходи очистки викидів. Існують різні методи очистки викидів від твердих, рідких і газоподібних домішок. На основі цих методів розроблено багато пристроїв та приладів, комплексне їхнє використання забезпечує високоефективне очищення пилогазових викидів.

Для очищення газів від твердих і рідких часток використовують технології сухої інерційної очистки газів, мокрої очистки газів, фільтрації, електростатичного осадження.

Сухі пиловловлювачі (осаджувальні камери, інерційні пиловловлювачі, циклони) призначені для грубого механічного очищення викидів від великих і важких часток пилу, Принцип роботи – осідання частинок під дією відцентрових сил і сили земного тяжіння.

Мокрі пиловловлювачі потребують подання води і працюють за принципом осадження частинок пилу на поверхні крапель під дією сил інерції та броунівського руху.

Фільтри належать до високоефективних типів апаратів сухої очистки газів. Вони здатні затримувати тонкодисперсні частинки пилу до 0,05 мкм. В основі роботи фільтрів усіх видів є пропускання запиленого повітря через пористі середовища. При цьому частинки пилу, завислі у газі, під дією броунівської дифузії, ефекту дотику, інерційних, електростатичних та гравітаційних сил осідають у пористому середовищі.

Електрофільтри є досконалими приладами для очистки газів від пилу. Принцип роботи всіх типів електрофільтрів базується на ударній іонізації пилогазового потоку і осіданні пилу на осаджувальних і коронуючих електродах. Забруднені гази, які надходять в електрофільтр, завжди є частково іонізованими за рахунок різних зовнішніх факторів, тому вони можуть проводити струм, потрапляючи у простір між двома електродами. Осаджені частинки пилу під дією сили тяжіння потрапляють у пилозбірник.

Для очистки газів від токсичних газо- і пароподібних компонентів використовують методи абсорбції, адсорбції, термічні і каталітичні.

Абсорбційний метод побудований на поглинанні речовин із суміші газів рідинами з утворенням розчинів. Рідини, які використовують для поглинання газоподібних домішок, називають абсорбентами.

Адсорбційний метод дає змогу поглинати газоподібні домішки активними поверхнями твердих речовин. Фізична основа процесу адсорбції –

здатність деяких твердих тіл з ультрамікроскопічною структурою (адсорбентів) вибірково виділяти та концентрувати на своїй поверхні окремі компоненти газової пароповітряної суміші або розчину.

Термічні методи знешкодження газоподібних сполук ґрунтуються на нейтралізації промислових і вентиляційних газів у результаті високотемпературного доспалювання. Воно може здійснюватися як за рахунок змішування цих викидів з повітрям без додаткового використання палива, так і з додаванням палива, а також як з утилізацією тепла, так і без нього.

Каталітичний метод полягає в нейтралізації шкідливих речовин, які містяться у виробничих газах, у результаті їхньої взаємодії з компонентами цього ж газу або спеціальними добавками під впливом каталізатора. На поверхні каталізатора в результаті його взаємодії з компонентами викидів утворюються проміжні сполуки, які вступають у подальші хімічні перетворення з відновленням первинного хімічного складу каталізатора та зв'язуванням шкідливих речовин у нешкідливі сполуки.

Організація санітарно-захисних зон. Санітарно-захисна зона – це смуга, яка відділяє джерело промислового забруднення від житлових або громадських будівель для захисту населення від впливу шкідливих чинників виробництва (викиди пилу або інших видів забруднення середовища).

Ширину санітарно-захисних зон встановлюють залежно від класу виробництва, ступеня шкідливості й кількості виділених в атмосферу речовин і приймають рівною від 50 до 1000 м. Наприклад, для цементних заводів продуктивністю більше 150 тис. т цементу в рік (І клас виробництва) ширина санітарно-захисної зони – 1000 м.

Санітарно-захисна зона повинна бути упоряджена й озеленена газостійкими породами дерев і чагарників, наприклад, акацією білою, тополею канадською, ялиною колючою, шовковицею, кленом гостролистим і т.д. Про ефективність озеленення свідчать наступні дані: хвоя одного гектара ялинового лісу вловлює 32 т пилу, листя букового лісу – 68 т. На відстані 500

м від підприємства при відсутності озеленення забруднення повітря SO_2 , H_2S , і NO_2 у два рази нижче, ніж біля джерела забруднення, а при наявності озеленення нижче у три-чотири рази.

З огляду на виняткову актуальність охорони атмосферного повітря від забруднення відпрацьованими газами автомобілів, першочерговою проблемою є створення екологічно «чистих» видів транспорту. Весь час ведеться активний пошук більш «чистого» палива, ніж бензин. У якості його заміниці розглядаються екологічно чисте газове паливо, метиловий спирт (метанол), малотоксичний аміак й ідеальне паливо – водень. Тривають інтенсивні розробки по заміні карбюраторного двигуна на більш екологічні типи – дизельний, паровий, газотурбінний й ін.

Також виробляються автомобілі на електричних двигунах. Тривають роботи зі створення ідеального з погляду екологічних вимог виду транспорту – автомобіля на сонячних елементах.

ВИСНОВКИ

Кваліфікаційну роботу присвячено удосконаленню регресійної моделі оцінювання розміру динамічних web-додатків на РНР та розробці програми для його реалізації.

Під час виконання кваліфікаційної роботи поставленні задачі, які було виконано у повному обсязі:

- виконано аналіз існуючих методів та моделей оцінювання розміру ПЗ;
- обґрунтовано необхідність удосконалення математичної моделі для оцінювання розміру ПЗ динамічних web-додатків на РНР;
- досліджено різні джерела з відкритим вихідним кодом та визначено динамічні сайти на РНР, які можуть бути використані для удосконалення моделі;
- побудовано діаграми класів та отримано необхідні метрики з кожного проекту;
- перевірено дані на викиди;
- виконано нормалізацію отриманих емпіричних даних, використовуючи перетворення у вигляді десяткового логарифму;
- побудовано лінійне рівняння регресії, довірчий інтервал та інтервал прогнозування для нормалізованих даних;
- виконано перехід від лінійної регресії до нелінійної та побудувати рівняння регресії, довірчий інтервал та інтервал прогнозування для вихідних даних;
- побудовано лінійне рівняння регресії, довірчий інтервал і інтервал прогнозування для вихідних даних без нормалізації в припущенні про нормальність вихідних даних;
- виконано порівняння отриманих лінійної та нелінійної моделей;

– розроблено ПЗ для оцінювання розміру динамічних web-додатків на PHP.

ПЗ для оцінювання розміру динамічних web-додатків на PHP, яке розроблено в рамках кваліфікаційної роботи, дозволить автоматизувати та скоротити час відповідних розрахунків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Титов, А.И. Выбор метрики размера проекта в модели оценки трудоемкости разработки программ [Текст] / А.И.Титов // Интеллектуальные технологии на транспорте. –2016. – Вып. 5. – С. 31-38.
2. Malathi, S. et al. Analysis of size metrics and effort performance criterion in software cost estimation [Текст] / S. Malathi et al. // Indian Journal of Computer Science and Engineering. –2012. – Volume 2, Issue 1. – P. 24-31.
3. Приходько, Н. В. Нелінійна регресійна модель для оцінювання розміру програмного забезпечення інформаційних систем на базі VB [Текст] / Н. В. Приходько, С. Б. Приходько // Інформаційні технології та компютерна інженерія. –2018. – № 3. – С. 37-42.
4. Приходько, С. Б. Оцінювання розміру РНР-застосунків з відкритим кодом за нелінійними регресійними моделями з різними факторами / С. Б. Приходько, М. В. Ворона // Зб. наук. пр. НУК. – Миколаїв : НУК, 2021. – № 1 (484). – С. 92-98.
5. Латанська Л.О. Використання нормалізуючих перетворень при побудові нелінійних регресійних моделей / Л.О. Латанська, О.С. Каіров, В.В. Біла // «Free and Open Source Software»: Матеріали XIII-ої Міжнародної науково-практичної конференції (16-18 листопада 2021 р.). – Харків: Харківський національний університет будівництва та архітектури, 2021. – с.20-21.
6. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. – СПб.: Питер, 2016. – 768 с.
7. Котеров, Д. В. РНР 7 [Текст] / Д. В. Котеров, И. В. Симдянов. – СПб.: БХВ-Петербург, 2016. –1088 с.
8. Новичков А., Шамрай А., Черников А. Метрики кода и их практическая реализация в Subversion и ClearCase. Часть 1 – метрики. [Электронный ресурс]. – Режим доступа: <http://gdznet.com/userhelps/metriki-koda-i-ix-prakticheskaya-realizaciya-v-subversion-i-clearcase-chast-1-metriki.html>

9. Ледовских, И. Метрики сложности кода [Электронный ресурс]. – Режим доступа: https://www.ispras.ru/preprints/docs/prep_25_2013.pdf
10. Briand L.C. Resource Estimation in Software Engineering [Text] / L.C. Briand, I. Wieczorek, // Encyclopedia of Software Engineering. John Wiley & Sons, Inc., New Jersey (2002).
11. Tan H. B. K. Estimating LOC for information systems from their conceptual data models [Text] / H. B. K. Tan, Y. Zhao, H. Zhang // in Proceedings of the 28th International Conference on Software Engineering (ICSE '06), May 20-28, 2006, Shanghai, China, pp. 321-330.
12. Boehm, B.W. Software engineering economics [Текст] / B.W. Boehm. – Prentice-Hall, 1981. – 320 p.
13. Johnson, Kim. Software cost estimation – Metrics and models [Текст] / Johnson Kim. – University of Calgary, 2001. – 115 p.
14. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников [Текст]/ А.И. Кобзарь. – М.: ФИЗМАТЛИТ, 2006. – 816с
15. Магнус, Я.Р. Эконометрика. Начальный курс: Учеб. – 6-е изд., перераб. и доп. [Текст] / Я.Р. Магнус, П.К. Катыхев, А.А. Пересецкий. – М.: Дело, 2004. – 576 с.
16. Приходько С.Б. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни «Обробка експериментальних даних на комп'ютері» / С.Б. Приходько Л.М. Макарова, К.С. Приходько. – Миколаїв: НУК, 2018. – 76с.
17. Дрейпер, Н. Прикладной регрессионный анализ: В 2-х кн. Кн. 2 / Пер. с англ. – 2-е изд., перераб. и доп. [Текст] / Н. Дрейпер, Г. Смит – М.: Финансы и статистика, 1987. – 351 с.
18. Приходько С.Б. Построение нелинейной регрессионной модели времени восстановления работоспособности устройств терминальной сети / С.Б. Приходько Л.М. Макарова // Проблемы информационных технологий. – Херсон: ХНТУ, 2014. – № 1(015). – С.97-102.

19. Hossain M.Z. The Use of Box-Cox Transformation Technique in Economic and Statistical Analyses. / M.Z. Hossain // Journal of Emerging Trends in Economics and Management Sciences. – Scholarlink Research Institute Journals, 2011. – № 2(1). – С.32-39.
20. Регресійний аналіз [Електронний ресурс]. – Режим доступу: https://pidruchniki.com/17280924/ekonomika/regresiynyi_analiz – Назва з екрану
21. Chatterjee, Samprit. Handbook of Regression Analysis [Text] / Samprit Chatterjee, Jeffrey S. Somonoff. Wiley, 2012. – 240 p.
22. Кобзарь, А.И. Прикладная математическая статистика. Для инженеров и научных работников [Текст]/А.И. Кобзарь. – М.: ФИЗМАТЛИТ, 2006. –816 с.
23. Айвазян С.А. Прикладная статистика. Основы эконометрики: Учебник для вузов: В 2 т. 2-е изд., испр. – Т.1: Теория вероятностей и прикладная статистика [Текст] / С.А. Айвазян, В.С. Мхитарян – М.: ЮНИТИ-ДАНА, 2001. – 656 с.
24. Фаронов В.В. Программирование в С#: учеб. пособие / В.В. Фаронов. – 7-е изд., перераб. – М.: Нолидж, 2006. – 412с.
25. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер, 2007. – 432 с.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP

Вступ

Назва програми – «Математична модель для оцінювання розміру динамічних web-застосунків, що створюються на PHP, та розробка програмного забезпечення для її реалізації».

1. Підстави для розробки

Документом, на підставі якого ведеться розробка програмного продукту, є завдання на кваліфікаційну роботу «Математична модель для оцінювання розміру динамічних web-застосунків, що створюються на PHP, та розробка програмного забезпечення для її реалізації».

2. Призначення розробки

2.1. Функціональне призначення

Даний програмний продукт дозволить автоматизувати та скоротити час відповідних розрахунків.

2.2. Експлуатаційне призначення

Програмний продукт буде використовуватись для оцінювання розміру динамічних web-додатків на PHP та оптимізації коду і побудови інтервалу прогнозування та довірчого інтервалу на основі десяткового логарифму.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик

3.1.1. Вимоги до складу функцій, що виконуються

Програма повинна забезпечувати можливість виконання наведених нижче функцій:

- 1) введення емпіричних даних;
- 2) нормалізація емпіричних даних з використанням нормалізуючого перетворення десятковий логарифм;
- 3) побудова нелінійної регресійної моделі та аналіз результатів;
- 4) побудова довірчих та прогнозних інтервалів для нелінійної регресійної моделі;
- 5) прогнозування розміру динамічних web-додатків на РНР.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вхідні дані мають бути організовані у текстових файлах. Внесення вхідних даних виконується шляхом вибору шляху до відповідного текстового файлу у спеціальному вікні.

Вихідні дані (довірчі інтервали, інтервали передбачення та метрики якості) повинні виводитися в відповідних місцях на створеній програмою формі.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійності функціонування програми

Надійне (стійке) функціонування програми має бути забезпечено надійним (стійким) функціонуванням операційної системи.

3.2.2 Відмова виконання програмного продукту через некоректні дії користувача

Відмова програми можлива внаслідок некоректних дій користувача при взаємодії з операційною системою. Щоб уникнути виникнення відмов

програми за вказаної вище причини необхідно забезпечити стабільне функціонування операційної системи.

3.4 Умови експлуатації

Кліматичні умови експлуатації програмного продукту відповідають умовам експлуатації апаратної частини персонального комп'ютера.

3.5 Вимоги до інформаційної та програмної сумісності

Системні програмні засоби (не вільно поширювані), використовувані програмою, повинні бути представлені ліцензійною версією ОС Windows.

3.6 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм відсутні.

3.7 Вимоги до маркування та упаковки

3.7.1 Вимоги до маркування

Вимоги до маркування відсутні.

3.7.2 Вимоги до упаковки

Вимоги до упаковки відсутні.

3.8 Вимоги до транспортування та зберігання

Вимоги до транспортування програми відсутні.

4 Вимоги до програмної документації

Склад програмної документації:

- технічне завдання;
- текст програми;
- опис програми;
- програма і методика випробувань;

– інструкція користувача.

5 Техніко-економічні показники

Економічна ефективність впровадження програмного забезпечення розрахована у розділі 5. Згідно з отриманими результатами впровадження даного ПЗ є доцільним.

6 Стадії та етапи розробки програмного забезпечення

Стадії та етапи розробки програми оцінювання розміру динамічних web-додатків на РНР, зазначені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи робіт	Вміст робіт	Контрольні точки
1	2	3	4
Технічне завдання	Обумовлення необхідності розробки	Постановка задачі. Збір початкових матеріалів.	З 03.09.2021 до 07.09.2021
	Розробка та затвердження технічного завдання	Визначення вимог до ПЗ. Визначення вимог до ПЗ. Обговорення та затвердження технічного завдання	З 08.09.2021 до 15.09.2021
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис алгоритму	З 16.09.2021 до 10.10.2021
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту	З 11.10.2021 до 14.10.2021

Продовження таблиці А.1

1	2	3	4
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних, розробка алгоритму рішення задачі, розробка структури ПЗ	З 15.10.2021 до 05.11.2021
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту	З 06.11.2021 до 10.11.2021
Робочий проект	Розробка ПЗ	Програмування та налагодження програми	З 11.11.2021 до 21.11.2021
	Розробка програмної документації	Розробка програмної документації відповідно до вимог ГОСТ 19.101-77	З 22.11.2021 до 01.12.2021
	Випробування програми	Розробка, узгодження та затвердження програми і методики випробувань, коригування програми і програмної документації за результатами випробувань	З 02.12.2021 до 05.12.2021

7 Порядок контролю та приймання

Контроль здійснюється замовником на кожній стадії розробки ПЗ. Приймання здійснюється замовником за програмою та методикою випробувань. Випробування проводиться в зазначений термін.

Хід проведення приймально-здавальних випробувань документується за допомогою протоколу проведення випробувань. На підставі протоколу проведення випробувань виконавач сумісно з замовником підписують акт приймання-здачі програми в експлуатацію.

ДОДАТОК Б**ТЕКСТ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ
ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP**

Program.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using System.Windows.Forms;
```

```
namespace WindowsFormsApp1
```

```
{
```

```
    static class Program
```

```
    {
```

```
        /// <summary>
```

```
        /// Главная точка входа для приложения.
```

```
        /// </summary>
```

```
        [STAThread]
```

```
        static void Main()
```

```
        {
```

```
            Application.EnableVisualStyles();
```

```
            Application.SetCompatibleTextRenderingDefault(false);
```

```
            Application.Run(new MForm());
```

```
        }
```

```
    }
```

```
}
```

FRead.cs

```
using System;  
using System.Collections.Generic;
```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Diagnostics;

namespace WindowsFormsApp1
{
    class FRead
    {
        private double[] x;
        private double[] y;
        private string path;

        public void read()
        {
            using (var reader = new StreamReader(path))
            {
                string row;
                int i = 0;

                var lineCount = File.ReadLines(path).Count();

                x = new double[lineCount];
                y = new double[lineCount];

                while ((row = reader.ReadLine()) != null)
                {
                    try
                    {
                        x[i] = Double.Parse(row.Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries)[0]);

```

```

    }
    catch (Exception e)
    {
        throw new Exception("Ошибка в формате
        количества классов");
    }

    if (x[i] <= 0)
    {
        throw new Exception("Количество классов меньше
        или равно 0");
    }

    try
    {
        y[i] = Double.Parse(row.Split(new[] { ' ' },
        StringSplitOptions.RemoveEmptyEntries)[1].Trim
        Start());
    }
    catch (Exception e)
    {
        throw new Exception("Ошибка в формате
        строк кода");
    }

    if (y[i] <= 0)
    {
        throw new Exception("Значение строк кода
        меньше или равно 0");
    }

    i++;

```

```

        }

    }

}

public double[] GetClass()
{
    return x;
}

public double[] GetLines()
{
    return y;
}

public FRead(string path)
{
    this.path = path;
    this.read();
}
}

}

LgNormaliz.cs
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApp1
{

```

```

abstract class LgNormaliz
{
    public const double t = 2.3534;
    private double S;
    private double ZxAverage;
    private double sum;
    protected int length;
    protected double[] x;
    protected double[] y;
    protected double[] Yoproximated;
    protected double[] xn;
    protected double[] yn;
    protected double[] delta;
    protected double[] yMinusDelta;
    protected double[] yPlusDelta;
    protected double[] deltaPrognozirovaniya;
    protected double[] yMinusDeltaPrognozirovaniya;
    protected double[] yPlusDeltaPrognozirovaniya;
    protected double[] Y;
    protected double[] YMinusDelta;
    protected double[] YPlusDelta;
    protected double[] YMinusDeltaPrognozirovaniya;
    protected double[] YPlusDeltaPrognozirovaniya;
    private double a;
    private double b;
    protected double rSquare;
    protected double MMRE;
    protected double PRED;

    public abstract void Normaliz();
    public abstract void Reverse();

```

```
protected void countOproximation()
```

```
{
    Approxim approxim = new Approxim(xn, yn);
    a = approxim.getA();
    b = approxim.getB();
    for (int i = 0; i < length; i++)
    {
        Yoproximated[i] = a * xn[i] + b;
    }
}
```

```
protected void countDeltaPrognozirovaniya()
```

```
{
    for (int i = 0; i < length; i++)
    {
        deltaPrognozirovaniya[i] = t * S * Math.Sqrt( 1 +
            (double)1 / (double)length + (xn[i] - ZxAverage) * (xn[i]
            - ZxAverage) / sum);
    }
}
```

```
protected void countIntervalPrognozirovaniya()
```

```
{
    for (int i = 0; i < length; i++)
    {
        yMinusDeltaPrognozirovaniya[i] =
            Yoproximated[i] - deltaPrognozirovaniya[i];
        yPlusDeltaPrognozirovaniya[i] = Yoproximated[i]
            + deltaPrognozirovaniya[i];
    }
}
```

```
protected void countDeltaDoveria()
```

```

{
    double sumZx = 0;

    for(int i = 0; i < length; i++)
    {
        sumZx += xn[i];
    }

    ZxAverage = sumZx / length;

    //count sum (Zx - Zxcp)^2
    sum = 0;
    for (int i = 0; i < length; i++)
    {
        sum += (xn[i] - ZxAverage) * (xn[i] - ZxAverage);
    }

    for (int i = 0; i < length; i++)
    {
        delta[i] = t * S * Math.Sqrt( (double)1 /
            (double)length + xn[i] - ZxAverage) * (xn[i] -
            ZxAverage) / sum );
    }
}

```

ДОДАТОК В

ОПИС ПРОГРАМИ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: ПЗ оцінювання розміру динамічних web-додатків на PHP «P_Size».

Позначення: «P_Size».

1.2 Програмне забезпечення, необхідне для функціонування програми

Програмне забезпечення «P_Size» є крос-платформним і основною необхідною вимогою для функціонування виробу є наявність .NET Framework (4.5 і вище версії).

1.3 Мови програмування

ПЗ «P_Size» реалізовано на мові високого рівня C#. Модулі, реалізовані на інших мовах програмування, не використовуються.

2 Функціональне призначення

2.1 Класи вирішуваних завдань і призначення програми

ПЗ «P_Size» повинно забезпечувати можливість виконання нижче перерахованих функцій:

- 1) введення емпіричних даних;
- 2) нормалізація емпіричних даних з використанням нормалізуючого перетворення десятковий логарифм;
- 3) побудова нелінійної регресійної моделі та аналіз результатів;

4) побудова довірчих та прогнозних інтервалів для нелінійної регресійної моделі;

5) прогнозування розміру динамічних web-додатків на PHP.

2.2 Функціональні обмеження

Файли повинні бути формату EXE.

3 Використовувані технічні засоби

ПЗ «P_Size» призначене для використання на персональних комп'ютерах, що працюють під управлінням наступних операційних систем:
ОС MS Windows XP/7/8/8.1/10

Для ПЗ «P_Size» пред'являються такі апаратні вимоги до ПЕОМ:

- оперативна пам'ять 512 МБ;
- дисковий простір – не менше 512 МБ вільного місця на диску;
- роздільна здатність екрану – 1280 x 1024 пікселів.

4 Виклик і завантаження

Запуск ПЗ «P_Size» в графічному режимі здійснюється двома способами:

- 1) за допомогою файлу запуску: «P_Size»;
- 2) з використанням командного рядку: P_Size.exe.

5 Вхідні і вихідні дані

Вхідні дані програми(строки коду та класи) повинні знаходитись в окремому текстовому файлі у два стовбці. Вихідні дані програми мають відображатися на екрані користувача.

ДОДАТОК Г

ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА РНР

1 Об'єкт випробувань

Об'єктом випробувань є ПЗ для оцінювання розміру динамічних web-додатків на РНР.

2 Мета випробувань

Перевірка функціональних можливостей програмного забезпечення на відповідність вимогам технічного завдання.

3 Вимоги до програми

В ході тестування програмного продукту необхідно перевірити, що розроблене ПЗ реалізує наступні функції:

- 1) введення емпіричних даних;
- 2) нормалізація емпіричних даних з використанням нормалізуючого перетворення десятковий логарифм;
- 3) побудова нелінійної регресійної моделі та аналіз результатів;
- 4) побудова довірчих та прогнозних інтервалів для нелінійної регресійної моделі;
- 5) прогнозування розміру динамічних web-додатків на РНР.

4 Вимоги до програмної документації

Склад програмної документації, що надається на випробування:

- технічне завдання;
- текст програми;
- опис програми;

- програма і методика випробувань;
- інструкція користувача.

5 Засоби і порядок випробувань

Випробування проводилось на платформі Windows 7 32bit.

Порядок випробувань:

- встановити і запустити програму;
- перевірити можливості додавання даних до програми;
- перевірити можливість продивитися довірчі інтервали та інтервали передбачення;
- перевірити можливість сховати інтервали;
- перевірити можливість побудови графіків довірчих інтервалів та інтервалів передбачення;
- перевірити можливість розрахувати метрики якості;
- перевірити неможливість ввести файл з даними неправильного формату.

6 Методика випробувань

Тест 1 – Перевірити можливості додавання даних до програми

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
 - Відкривається вікно в якому користувач може вибрати шлях до файлу.
 - Користувач обирає файл з даними, що мають правильний формат.
 - Біля кнопки з'являється назва файлу.
 - Програма продовжує роботу.
- Очікуваний результат: після обрання файлу він додається.

Тест 2 – Перевірити неможливість ввести файл з даними неправильного формату

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
- Відкривається вікно в якому користувач може вибрати шлях до файлу.
- Користувач обирає файл з даними, що мають неправильний формат.

Очікуваний результат: На екрані з'являється повідомлення про невідповідність формату даних.

Тест 3: Перевірити можливість виведення інтервалів

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
- Користувач натискає кнопку «Розрахувати».
- Користувач натискає кнопку «Вивести інтервали».

Очікуваний результат: після натискання кнопки «Вивести інтервали» на головній формі додається таблиця, яка містить ці інтервали та кнопка «Вивести інтервали» змінюється на кнопку «Вилучити інтервали»

Тест 4: перевірити можливість вилучити інтервали

Сценарій:

- Користувач натискає кнопку «Вибрати файл».
- Користувач натискає кнопку «Розрахувати».
- Користувач натискає кнопку «Вивести інтервали». На головній формі додається таблиця, яка містить ці інтервали та кнопка «Вивести інтервали» змінюється на кнопку «Вилучити інтервали»

- Користувач натискає кнопку «Вилучити інтервали».

Очікуваний результат: таблиця з інтервалами зникає, а кнопка «Вилучити інтервали» змінює текст на «Вивести інтервали».

ДОДАТОК Д

ІНСТРУКЦІЯ КОРИСТУВАЧА ПЗ ОЦІНЮВАННЯ РОЗМІРУ ДИНАМІЧНИХ WEB-ДОДАТКІВ НА PHP

1. Для того, щоб запустити програму, потрібно запустити файл «P_Size.exe». При запуску програми з'являється головна форма програми (рис. Д.1).

Рисунок Д.1 – Головна форма

2. Для внесення даних необхідно натиснути на кнопку «Вибрати файл». Потім необхідно у вікні, що відкриється, знайти файл з даними та вибрати його. У випадку, якщо в файлі невірного формату дані по кількості класів, то з'явиться повідомлення «Помилка в форматі кількості класів» (рис. Д.2).

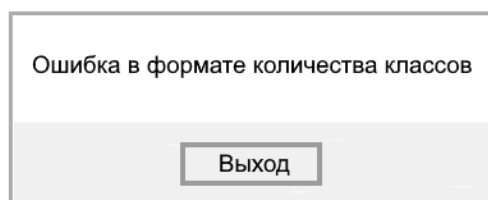


Рисунок Д.2 – Повідомлення про помилку в форматі кількості класів

У випадку, якщо в файлі невірною формату дані по кількості строк коду, то з'явиться повідомлення «Помилка в форматі строк коду» (рис. Д.3).

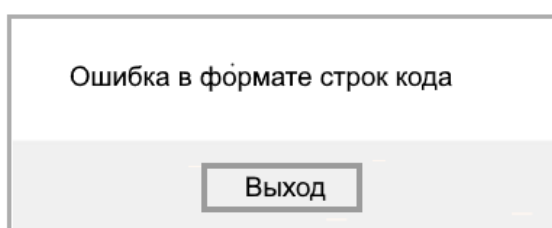


Рисунок Д.3 – Повідомлення про помилку в форматі кількості строк коду

У випадку помилок у форматі, необхідно натиснути на кнопку «Вихід» та обрати інший файл або вийти з програми.

3. Якщо інформація в файлі вірна, то необхідно натиснути на кнопку «Розрахувати».

Відбудеться побудова моделі і розрахунок показників якості. Показники якості з'являться на екрані (рис. Д.4).

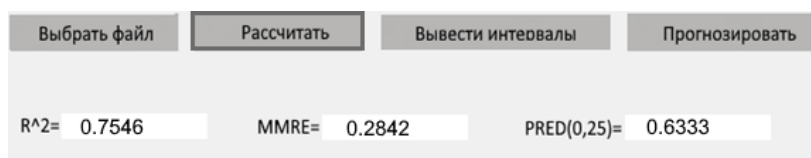


Рисунок Д.4 – Показники якості

6. Для розрахунку та виведення довірчих інтервалів та інтервалів прогнозування необхідно натиснути на кнопку «Вивести інтервали» (рис. Д.5).

Выбрать файл Рассчитать Удалить интервалы Прогнозировать

$R^2 =$ 0.7546 $MMRE =$ 0.2842 $PRED(0,25) =$ 0.6333

Доверительный		Прогнозирования	
804.6979	1549.7964	459.2388	2715.6197
936.8586	1735.0354	527.9958	3078.5903
936.8586	1735.0354	527.9958	3078.5903
1622.1298	2626.0220	873.0464	4879.1776
1622.1298	2626.0220	873.0464	4879.1776

Рисунок Д.5 – Довірчі інтервали та інтервали прогнозування

При цьому кнопка змінить назву на «Видалити інтервали»

7. Для того щоб видалити інтервали з форми, необхідно натиснути на кнопку «Видалити інтервали».

8. Для виконання прогнозування розміру необхідно натиснути на кнопку «Прогнозувати» (рис. Д.6).

Выбрать файл Рассчитать Вывести интервалы Прогнозировать

Классов 28 Ok

Строк кода 4412.1400

Доверительный		Прогнозирования	
3747.9296	5194.0621	1893.7807	10279.4266

Рисунок Д.6 – Результат прогнозування, довірчий інтервал та інтервал прогнозування

9. Для завершення роботи програми необхідно натиснути на кнопку «Вийти».