

Макарова Л.М.

Національний університет кораблебудування імені адмірала Макарова

Латанська Л.О.

Національний університет кораблебудування імені адмірала Макарова

ПОРІВНЯЛЬНИЙ АНАЛІЗ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ МОВОЮ PYTHON З ВИКОРИСТАННЯМ DJANGO REST FRAMEWORK

Визначення розміру програмного забезпечення на ранніх стадіях роботи над проектом є поширеною задачею в галузі інженерії програмного забезпечення. Використовуючи такі існуючі моделі як, наприклад, COCOMO, COCOMO II, ISBSG, COSMIC та отриманий розмір можна прогнозувати трудомісткість, вартість, тривалість проекту. Значну частину розроблюваних проектів становлять веб-застосунки. Однією з найбільш популярних мов веб-розробок є мова Python. Для неї існує ряд фреймворків, найбільш повним та розповсюдженим з яких є Django Rest Framework.

Метою дослідження є визначення найбільш якісної та достовірної нелінійної регресійної моделі для прогнозування розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework.

Для отримання нелінійних регресійних моделей та інтервалів прогнозування застосовано методіку побудови нелінійних регресійних моделей на основі одновимірного нормалізуючого перетворення.

Для задачі аналізу побудовано дві нелінійні регресійні моделі разом з інтервалами прогнозування з використанням десятичного логарифму у якості нормалізуючого перетворення. В якості предикторів розглядалися такі метрики, як: кількість класів та кількість методів. Для однофакторної моделі предиктором слугувала кількість класів, для двофакторної – кількість класів та кількість методів у проекті. Також в роботі для порівняння була використана однофакторна нелінійна регресійна модель, побудована для веб-застосунків, які розроблені з використанням мови Java.

Отримані результати дозволяють зробити наступні висновки. При збільшенні кількості предикторів, що використовувалися для побудови нелінійних регресійних моделей для прогнозування розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework, зростає якість моделі, а саме покращилися значення критеріїв якості та зменшилася ширина інтервалу прогнозування. При спробі застосувати нелінійну регресійну модель, побудовану для веб-застосунків, які розроблені з використанням мови Java, до емпіричних даних із даного дослідження, отримано низькі показники якості та більшу ширину інтервалу прогнозування. Це може слугувати підтвердженням необхідності побудови регресійних моделей для конкретної мови програмування та фреймворку.

Ключові слова: оцінювання розміру програмного забезпечення, нелінійна регресійна модель, нормалізуюче перетворення, розробка програмного забезпечення, веб-застосунок, Python, Django Rest Framework.

Постановка проблеми. Визначення розміру програмного забезпечення на ранніх стадіях розробки проекту є важливою задачею, адже використовуючи отриманий розмір та такі моделі, як COCOMO, COCOMO II, ISBSG, COSMIC, можна виконати прогнозування вартості, тривалості та трудомісткості розробки [1–4]. Це призводить до більш якісного планування роботи над проектом, кращого управління ризиками проекту. Хоч сьогодні й існує певна кількість моделей для оцінювання розміру програмного забезпечення

для конкретних мов програмування та фреймворків, в більшості своїй ці моделі втрачають свою якість при застосуванні для іншої мови програмування, зокрема навіть при застосуванні моделі для іншого фреймворку.

Аналіз останніх досліджень і публікацій. Питання побудови багатфакторних регресійних моделей для оцінювання розміру інформаційних систем для різних мов програмування, наприклад, Java та PHP, були розглянуті, зокрема, у роботах [5, 6]. Моделі в цих роботах будувалися на основі об'єктно-

орієнтованих метрик програмного забезпечення, які можна отримати з діаграми класів при ранньому проектуванні, наприклад, на кількості класів, кількості відносин між класами, середній кількості атрибутів, в той час як залежною величиною виступала кількість рядків коду. Одним із критеріїв, що дозволяє використовувати лінійний регресійний аналіз, є підпорядкування розподілу залежної змінної або залишків регресії нормальному закону, що на практиці буває рідко, це і зумовлює необхідність використання методів нелінійного регресійного аналізу.

На сьогодні існує ряд робіт, присвячених побудові нелінійних регресійних моделей для оцінювання розміру веб-застосунків, що створюються такими мовами, як, PHP з різними фреймворками [6–8], Java [9], Python [10]. У цих роботах також використовуються метрики діаграми класів, але у різних комбінаціях, що дозволяє умовно поділити ці роботи за використаними метриками. Втім порівняння різних моделей з однаковими факторами між собою та моделей з різними факторами не проводилось.

Відсутність досліджень щодо впливу кількості факторів нелінійної регресійної моделі на її якість та на ширину інтервалу прогнозування для мови Python та фреймворку Django Rest Framework є підставою для проведення дослідження.

Мета роботи – визначення найбільш якісної та достовірної нелінійної регресійної моделі для прогнозування розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework.

В задачі аналізу будуть розглядатися дві нелінійні регресійні моделі разом з інтервалами прогнозування з використанням десятичного логарифму у якості нормалізуючого перетворення. В якості предикторів для двофакторної моделі будуть використовуватися такі метрики як: кількість класів та кількість методів у проекті. Для однофакторної моделі в якості предиктора приймається кількість класів.

Двофакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework, була побудована в роботі [10]. Однофакторна нелінійна регресійна модель та інтервали прогнозування обох моделей будуються в рамках даного дослідження на емпіричних даних з [10].

Для досягнення поставленої мети роботи необхідно виконати наступні завдання:

– Провести дослідження існуючих моделей для оцінювання розміру веб-застосунків, розроблених, зокрема, з використанням мови Python.

– Виконати побудову однофакторної нелінійної регресійної моделі для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework, для емпіричних даних на основі нормалізуючого перетворення.

– Виконати побудову інтервалів прогнозування обох моделей для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework.

– Виконати порівняння побудованих регресійних моделей для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework та однофакторної нелінійної регресійної моделі, побудованої для веб-застосунків, які розроблені з використанням мови Java.

Виклад основного матеріалу дослідження. За умови гаусівського розподілу початкових даних в якості регресійної моделі можна взяти лінійну регресійну модель з наступним загальним виглядом:

$$Y = \hat{Y} + \varepsilon = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon, \quad (1)$$

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

В тому випадку, коли початкові дані не є нормальними, необхідно будувати нелінійну регресійну модель. Одним з кращих за результативністю підходів до побудови нелінійних регресійних моделей є використання нормалізуючих перетворень. На відміну від лінеаризуючих перетворень чи методів підбору, нормалізуючі перетворення можна підібрати в більшості випадків.

Згідно [11] ітераційний алгоритм для побудови нелінійної регресійної моделі з використанням нормалізуючих перетворень наступний:

– Застосувати нормалізуюче перетворення до емпіричних даних та вилучити викиди.

– Для нормалізованих даних без викидів побудувати лінійну регресійну модель.

– Перевірити залишки лінійної регресії на відповідність нормальному закону розподілу та відкинути значення з максимальним відхиленням у разі невідповідності.

– Перейти від лінійної регресійної моделі до нелінійної, використавши зворотне нормалізуюче перетворення.

– Визначити інтервал прогнозування та відкинути значення, що виходять за границі інтервалу, якщо такі є.

Нормалізуюче перетворення негаусівського випадкового вектору $P = \{Y, X_1, X_2, \dots, X_k\}^T$ у гау-

сівський випадковий вектор $T = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ у загальному вигляді задається як:

$$T = \psi(P), \quad (2)$$

а зворотне перетворення до (2) – як:

$$P = \psi^{-1}(T), \quad (3)$$

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

Таким чином нелінійна регресійна модель, що побудована з використанням нормалізуючого перетворення, має вигляд:

$$Y = \psi_Y^{-1}(\hat{Z}_Y + \varepsilon) = \psi_Y^{-1}(b_0 + b_1 Z_1 + b_2 Z_2 + \dots + b_k Z_k + \varepsilon), \quad (4)$$

де ψ_Y – перша компонента вектору ψ перетворення (2).

Негаусівський розподіл даних підтверджують значення квадрату відстані Махаланобіса (SMD), які можна визначити як:

$$d_i^2 = (X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}), \quad (5)$$

де X_i – i -та точка багатовимірних даних вектору X ; \bar{X} – вектор вибіркового середнього; S_N – вибіркова коваріаційна матриця:

$$S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})^T, \quad (6)$$

де N – кількість елементів у вибірці.

Негаусівський розподіл багатовимірних емпіричних даних також підтверджує проведення тесту, який передбачає отримання оцінки багатовимірного ексцесу β_2 [12], яку можна розрахувати як:

$$\hat{\beta}_2 = \frac{1}{N} \sum_{i=1}^N \{(X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X})\}^2. \quad (7)$$

Після побудови нелінійної регресійної моделі можна визначити її інтервал прогнозування як це робиться у [11]:

$$\psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, \nu} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (z_X^+)^T [(Z_X^+)^T Z_X^+]^{-1} (z_X^+) \right\}^{1/2} \right). \quad (8)$$

Тут ψ_Y^{-1} – взаємозворотне перетворення, $t_{\alpha/2, \nu}$ – квантіль t -розподілу Стьюдента з рівнем значимості $\alpha/2$ та ν ступенями вільності, $\nu = N - k - 1$, k – кількість незалежних змінних, $z_X^+ = \{Z_1 - \bar{Z}_1, Z_2 - \bar{Z}_2, Z_3 - \bar{Z}_3\}^T$, $S_{Z_Y}^2 = \frac{1}{\nu} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2$, $(z_X^+)^T z_X^+ = k \times k$ матриця:

$$(z_X^+)^T z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_2 Z_1} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_k Z_1} & S_{Z_k Z_2} & \dots & S_{Z_k Z_k} \end{pmatrix}, \quad (9)$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{q_i} - \bar{Z}_q][Z_{r_i} - \bar{Z}_r]$, $q, r = \overline{1, k}$.

Нелінійні регресійні моделі в межах цього дослідження будувались на основі емпіричних даних, зібраних на платформі GitHub: 71 проекту веб-застосунків, створених мовою Python з вико-

ристанням Django Rest Framework та наведених у роботі [10]. Подальше опрацювання отриманих емпіричних даних проводилось з використанням розширення до PyCharm Statistic [13, 14]. В результаті отримано наступні метрики програмного забезпечення: кількість рядків коду у тисячах KLOC (Y), кількість класів Classes (X_1), кількість методів проекту Methods (X_2).

У випадку двофакторної нелінійної регресійної моделі у якості предикторів використаємо наступні метрики: кількість класів Classes (X_1), кількість методів проекту Methods (X_2), а залежною величиною буде кількість рядків коду у тисячах KLOC (Y).

Негаусівський розподіл даних підтверджують отримані значення квадрату відстані Махаланобіса (SMD), знайдені за допомогою (5) та (6).

Також негаусівський розподіл даних підтверджується за допомогою багатовимірного ексцесу β_2 . Для нормального розподілу $\beta_2 = 15$, що визначається за формулою $\beta_2 = m(m+2)$, де m дорівнює 3 [12]. Натомість для вихідних даних отримано значення 29,61.

Пошук викидів у даних виконано за допомогою нормалізуючого перетворення на основі десятичного логарифму та розрахунку квадрату відстані Махаланобіса за формулою (5) та тестової статистики за формулою (6), як наведено у [15, 16]. З початкового набору даних лишилось 62 проекти веб-застосунків, створених мовою Python з використанням Django Rest Framework.

Детально побудова нелінійної двофакторної регресійної моделі була представлена в [10]. Далі за допомогою (8) та (9) будуємо інтервал прогнозування двофакторної нелінійної регресії. Оскільки для двох точок (38 та 66) вихідні значення Y виходять за розраховані межі інтервалу прогнозування, вважаємо ці точки викидами та видалимо їх із загальної вибірки даних. Таким чином, в остаточному скорегованому наборі даних маємо 60 точок даних.

Для цього скорегованого набору даних виконуємо всі дії для побудови нелінійної регресійної моделі та інтервалу прогнозування. Остаточна лінійна двофакторна регресійна модель має вигляд:

$$Z_Y = -1,6417 - 0,0090 Z_1 + 0,9902 Z_2 + \varepsilon.$$

Застосувавши взаємо-зворотне нормалізуюче перетворення, отримуємо двофакторну нелінійну регресійну модель для оцінювання розміру веб-застосунків, створених мовою Python з використанням Django Rest Framework, яка має вигляд:

$$Y = 10^{e-1,6417} X_1^{-0,0090} X_2^{0,9902}.$$

Після побудови інтервалу прогнозування двофакторної нелінійної регресії модель вважається побудованою, адже відсутні значення змінної Y , які виходять за границі інтервалу прогнозування.

Оскільки для мови програмування Python серед опублікованих матеріалів не було знайдено відповідної моделі для порівняння, було прийнято рішення побудувати однофакторну нелінійну регресійну модель, оскільки це найпростіший варіант оцінювання у разі об'єктно-орієнтованої методології розробки.

Однофакторна нелінійна регресійна модель будувалась за тим же алгоритмом, що й двофакторна, незалежною величиною була обрана кількість класів $Classes (X_1)$, відповідно залежною – кількість рядків коду у тисячах KLOC (Y).

На етапі попередньої обробки даних за допомогою квадрату відстані Махаланобіса було знайдено 11 викидів. Після побудови нелінійної регресійної моделі та інтервалу прогнозування було знайдено ще 6 точок, для яких вихідні значення Y виходять за розраховані границі інтервалу прогнозування. Ці точки також було видалено із загальної вибірки даних. Таким чином, в остаточному скорегованому наборі даних маємо 54 точки даних.

Для скоригованого набору даних будуюмо нелінійну регресійну модель та інтервал прогнозування. Остаточно лінійна однофакторна регресійна модель має вигляд:

$$Z_Y = -1,0401 + 0,8597Z_1 + \varepsilon.$$

Відповідно однофакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, створених мовою Python з використанням Django Rest Framework має вигляд:

$$Y = 10^{e-1,0401} X_1^{0,8597}.$$

Після побудови інтервалу прогнозування однофакторної нелінійної регресії відсутні значення змінної Y , які виходять за границі інтервалу прогнозування.

Також було виконано порівняння моделей для веб-застосунків, створених мовою Python з використанням Django Rest Framework, з опублікованою однофакторною нелінійною регресійною моделлю, побудованою для веб-застосунків, які розроблені з використанням мови Java [17]. Для порівняння якості нелінійних регресійних моделей були використані такі критерії: коефіцієнт детермінації R^2 , середня величина відносної похибки $MMRE$, рівень прогнозування $PRED(0,25)$, які наведено в таблиці 1.

Таблиця 1

Критерії якості нелінійних регресійних моделей

Критерії якості	Двофакторна (Python)	Однофакторна (Python)	Однофакторна (Java)
R^2	0,8520	0,7063	0,5686
MMRE	0,1966	0,3533	1,2285
$PRED(0,25)$	0,6833	0,4444	0,2254

Аналіз даних, наведених в табл. 1, показує, що двофакторна нелінійна регресійна модель має кращі значення всіх трьох критеріїв. Отримані результати дозволяють зробити висновок про збільшення якості моделі при включенні до неї додаткових предикторів. Однак значення $PRED(0,25)$ дещо нижче за необхідне значення (не менше, ніж 0,75), що свідчить про необхідність або збільшення кількості предикторів, або використання для побудови моделі інших нормалізуючих перетворень, зокрема, Бокса-Кокса або Джонсона.

Оцінки для однофакторної нелінійної регресійної моделі, яка була побудована для веб-застосунків, що розроблені з використанням мови Java, свідчать про низьку якість отриманої моделі та необхідність будувати модель саме під конкретну мову програмування та фреймворк для отримання прийнятного результату. Таким чином, дана модель може відображати суттєву різницю у проектах програмного забезпечення, яка в свою чергу унеможливує застосування цієї моделі до емпіричних даних даного дослідження.

В таблиці 2 наведено інтервали прогнозування побудованих нелінійних регресійних моделей, а також їх ширини (фрагмент даних). Рядки, які містять прочерк, не ввійшли до даних, які було використано для остаточної побудови тієї чи іншої моделі.

У таблиці 2 напівжирним виділено кращі значення ширини інтервалу прогнозування серед усіх трьох моделей. Більше всього рядків з меншою шириною інтервалу прогнозування спостерігається для двофакторної нелінійної регресійної моделі – 46 рядків, для однофакторної – 3 рядки, для однофакторної моделі з використанням мови Java такі рядки відсутні (розглядалися лише ті рядки, де присутні усі три моделі).

Аналіз даних, які можна вважати викидами показує, що в процесі побудови нелінійних регресійних моделей деякі рядки були визначені спільними викидами для двофакторної та однофакторної моделей, деякі рядки були визначені викидами лише для однофакторної моделі, а деякі рядки були визначені викидами лише для двофакторної моделі.

Інтервали прогнозування нелінійних регресійних моделей та їх ширини (фрагмент даних)

№	KLOC (Y)	Двофакторна (Python)			Однофакторна (Python)			Однофакторна (Java)		
		LB	UB	Length	LB	UB	Length	LB	UB	Length
1	2	3	4	5	6	7	8	9	10	11
1	10,07	7,51	20,16	12,66	1,90	10,80	8,90	0,85	93,77	92,92
2	2,71	1,97	5,20	3,23	0,85	4,85	4,00	0,28	28,89	28,61
3	2,96	1,67	4,40	2,73	1,58	8,99	7,41	0,68	69,69	69,01
4	1,20	1,11	2,93	1,82	0,50	2,87	2,37	0,13	14,26	14,14
5	9,47	5,05	13,36	8,31	3,65	20,98	17,33	2,19	238,63	236,44
6	2,09	1,40	3,69	2,28	0,83	4,74	3,91	0,27	28,25	27,98
7	10,00							0,40	44,27	43,87
8	2,08	1,56	4,11	2,55	1,26	7,13	5,87	0,48	50,73	50,24
9	1,98	0,96	2,55	1,59	1,12	6,38	5,26	0,41	43,44	43,03
10	0,30							0,01	0,96	0,95
11	0,36	0,21	0,56	0,35	0,15	0,90	0,75	0,02	2,90	2,88
12	0,57	0,30	0,82	0,52				0,16	19,97	19,81
13	28,76	25,13	68,73	43,60	7,47	44,19	36,72	5,66	743,05	737,39
14	13,39	5,69	15,23	9,54	5,72	33,41	27,69	4,09	470,51	466,41
15	4,98	2,19	5,76	3,57	1,83	10,40	8,57	0,84	85,75	84,91
16	0,93				0,41	2,38	1,97	0,09	11,07	10,98
17	3,39	1,90	5,00	3,10	1,42	8,07	6,65	0,58	59,55	58,97
18	4,50	2,79	7,62	4,83				3,87	394,73	390,86
19	4,04	2,35	6,19	3,84	1,78	10,10	8,32	0,81	81,99	81,19
20	0,53	0,35	0,93	0,58	0,32	1,87	1,55	0,06	8,17	8,11

Висновки. У роботі було побудовано двофакторну та однофакторну нелінійні регресійні моделі для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework, також було виконано порівняння цих моделей між собою та з опублікованою однофакторною нелінійною регресійною моделлю, побудованою для веб-застосунків, які розроблені з використанням мови Java.

Отримані результати показують, що зі збільшенням предикторів, на основі яких будується нелінійна

регресійна модель, якість моделі підвищується, а ширина інтервалу прогнозування – зменшується.

За результатами порівняння нелінійних регресійних моделей бачимо, що є необхідність будувати модель для конкретної мови програмування та фреймворку.

В подальшому для отримання більш якісних нелінійних регресійних моделей планується збільшувати кількість предикторів у моделі та застосовувати інші нормалізуючі перетворення, зокрема, Бокса-Кокса або Джонсона.

Список літератури:

1. Boehm B., Abts C., Chulani S. Software development cost estimation approaches – A survey. *Annals of Software Engineering*. 2000. Vol. 10. P. 177–205.
2. Boehm B.W., Abts C., Brown A.W. et al. Software cost estimation with COCOMO II. Upper Saddle River, NJ: Prentice Hall PTR, 2000. 506 p.
3. Abran A. Data collection and industry standards: the ISBSG repository. *Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers*. IEEE, 2015. P. 161–184, DOI: 10.1002/9781118959312.ch8
4. Marco L.D., Ferrucci F., Gravino C. Approximate COSMIC Size to Early Estimate Web Application Development Effort. *Software Engineering and Advanced Applications (SEAA). 39th EUROMICRO Conference*. 2013. P. 349–356.
5. Prykhodko N.V., Prykhodko S.B. The non-linear regression model to estimate the software size of open-source Java-based systems. *Радіоелектроніка, інформатика, управління*. Запоріжжя: ЗНТУ, 2018. № 3(46). С. 158–166.
6. Приходько С.Б., Приходько Н.В., Фаріонова Т.А., Ворона М.В. Трьохфакторна нелінійна регресійна модель для оцінювання розміру PHP-застосунків із відкритим кодом. *Вчені записки Таврійського національного університету імені В.І. Вернадського, Технічні науки*. Київ, 2020. № 1(70) Ч. 1. С. 124–131.
7. Приходько С.Б., Приходько Н.В., Ворона М.В., Беловол І.О. Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*. Вінниця: ВНТУ, 2021. № 1(50). С. 115–121.

8. Латанська Л.О., Макарова Л.М., Кольцов А.В., Давлатова Д.Х. Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням PHP фреймворку Symfony. *Вісник Хмельницького національного університету. Серія: «Технічні науки»*. 2022. № 6, Т. 1. С. 119–124. DOI: <https://www.doi.org/10.31891/2307-5732-2022-315-6-119-124>
9. Prykhodko S.B., Pukhalevych A.V., Prykhodko K.S., Makarova L.M. Nonlinear regression models for estimating the duration of software development in Java for PC based on the 2021 ISBSG data. *Radio Electronics, Computer Science, Control*. 2022. № 3(62). P. 144–154. DOI 10.15588/1607-3274-2022-3-14
10. Макарова Л.М., Латанська Л.О., Давлатова Д.Х., Кольцов А.В. Двофакторна нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються мовою Python з використанням Django Rest Framework. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Київ: ТНУ, 2022. Том 33 (72) № 6. С. 104–108. DOI <https://doi.org/10.32782/2663-5941/2022.6/18>
11. Prykhodko S., Prykhodko N. Mathematical modeling of non-Gaussian dependent random variables by nonlinear regression models based on the multivariate normalizing transformations. *Mathematical Modeling and Simulation of Systems: 15th International Scientific-practical Conference MODS'2020*. Chernihiv, Ukraine, June 29 – July 01, 2020, selected papers. Springer, Cham, 2021, P. 166–174. (Advances in Intelligent Systems and Computing, Vol. 1265). DOI: 10.1007/978-3-030-58124-4_16
12. Mardia K.V. Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhya: The Indian Journal of Statistics, Series B* (1960–2002), 1974. Vol. 36. Issue 2. P. 115–128.
13. Plugin Statistic for PyCharm. URL: <https://plugins.jetbrains.com/plugin/4509-statistic> (дата звернення: 24.10.2023 р.).
14. Інструкція по початку роботи з PyCharm. URL: <https://py-charm.blogspot.com/2017/09/blog-post.html> (дата звернення: 14.10.2023 р.).
15. Prykhodko S., Prykhodko N., Makarova L., Pugachenko K. Detecting outliers in multivariate non-Gaussian data on the basis of normalizing transformations, in: *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Kyiv, Ukraine, 2017. P. 846–849. <https://doi.org/10.1109/UKRCON.2017.8100366>
16. Prykhodko S., Prykhodko N., Makarova L., Pukhalevych A. Outlier Detection in Non-Linear Regression Analysis Based on the Normalizing Transformations. *Proceedings of the 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), IEEE*. (Lviv-Slavske, 2020). P. 407–410.
17. Макарова Л.М., Приходько Н.В., Кудін О.О. Побудова нелінійної регресійної моделі для оцінювання розміру веб-додатків, реалізованих мовою Java. *Вісник Херсонського національного технічного університету*. Херсон, 2019. № 2(69). С. 145–153.

Makarova L.M., Latanska L.O. COMPARATIVE ANALYSIS OF NONLINEAR REGRESSION MODELS TO PREDICT THE SIZE OF WEB APPLICATIONS CREATED IN PYTHON USING THE DJANGO REST FRAMEWORK

Determining the size of the software in the early stages of project work is a common task in the field of software engineering. Using existing models such as, for example, COCOMO, COCOMO II, ISBSG, COSMIC and the size you can predict the effort, cost, duration of the project. Much of the developed projects are web-applications. Python is one of the most popular web development languages. There are several of frameworks exist for its, the most complete and widespread of which is Django Rest Framework.

The purpose of the study is to identify the highest quality and reliable nonlinear regression model for predicting the size of web- applications created by Python using Django Rest Framework.

To obtain nonlinear regression models and prediction intervals, a technique for building nonlinear regression models based univariate normalizing transformation was used.

For the analysis problem two nonlinear regression models with the prediction intervals using a decimal logarithm as a normalizing transformation were build. There was considered as predictors of such metrics: the number of classes and the number of methods. The number of classes served as a predictor for a single-factor model, the number of classes and the number of methods in the project served as a predictors for a two-factor model. A single-factor nonlinear regression model, built for web-applications, which was designed using the Java, was also used in comparison.

The results obtained allow you to draw the following conclusions. With the increasing number of predictors, used to construct nonlinear regression models to predict the size of web- applications created by Python using Django Rest Framework, the quality of the model is increased, namely the quality criteria became better and the width of the prediction interval is decreased. When trying to apply a nonlinear regression model built for web- applications, which are developed using the Java, to empirical data from this study, low quality and greater width of the prediction interval was obtained. This is a confirmation of the need to build regression models for each programming language and framework.

Key words: estimation of software size, nonlinear regression model, normalizing transformation, software development, web application, Python, Django Rest Framework.