ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.412:519.233.5 **doi: 10.32620/reks.2022.3.06**

Sergiy PRYKHODKO[1], Ivan SHUTKO[1], Andrii PRYKHODKO[2]

[1] ***Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine***
[2] ***GlobalLogic Inc., Mykolaiv, Ukraine***

EARLY SIZE ESTIMATION OF WEB APPS CREATED USING CODEIGNITER FRAMEWORK BY NONLINEAR REGRESSION MODELS

***Subject matter**: Early software size estimation is one of the project managers' significant problems in evaluating app development efforts because software size is the major determinant of software project effort. Function points (FPs) and lines of code (LOC) are most commonly used as measures of size in existing software effort estimation methods and models. As is known, both these metrics have their advantages and disadvantages when used for software effort estimation. Although the FPs-based measure has the advantage over the LOC in that it does not depend on the technologies used, however, the assessment of efforts requires considering such factors (environmental factors). Considering the above factors can be ensured by appropriate models for estimating the LOC-based effort. Nowadays, many Web apps are created using PHP frameworks making the app development faster. CodeIgniter is one such powerful framework. However, there are no regression models for estimating the software size of Web apps created using the CodeIgniter framework. This requires the construction of the appropriate models. **The task** of this paper is to develop a nonlinear regression model for estimating the software size (in KLOC, kilo lines of code) of Web apps created using the CodeIgniter framework. **Method**: We apply the technique for constructing nonlinear regression models based on the multivariate normalizing transformations and prediction intervals. **The result** is three nonlinear regression models with three predictors: the total number of classes, the average number of methods per class, and the DIT (Depth of Inheritance Tree) average per class. To build these models for estimating the size of Web apps created using the CodeIgniter framework, we used three well-known normalizing transformations: two univariate transformations (the decimal logarithm and the Box-Cox transformation) and the Box-Cox four-variate transformation. **Conclusions**. The nonlinear regression model constructed by the Box-Cox four-variate transformation has better size prediction results than other regression models based on the univariate transformations.*

Keywords: *software size estimation; web app; CodeIgniter; framework; nonlinear regression model; normalizing transformation; non-Gaussian data.*

Introduction

Early software size estimation is one of the project managers' significant problems in evaluating app development efforts [1, 2] including Web apps [3, 4]. According to [5], “Software size is the major determinant of software project effort.” Failed software size estimating is often the main contributor to failed effort estimates and, in consequence, failed projects.

Despite a large number of currently existing various methods and models for estimating the software size [6, 7] including object-oriented approach [8], research in this direction does not stop [9, 10]. This is primarily due to the low accuracy of estimating the size of the software in the early stages of its development. One way to solve this problem is to develop appropriate models for estimating the size of the software, which is developed in a specific programming language such as C++ [11], Java [12], PHP [13], Visual Basic [14] and for a specific type of app, including information systems [15], embedded software components [16], Web apps [17].

Function points (FPs) and lines of code (LOC) are most commonly used as measures of size in existing software effort estimation methods and models. As known [5], both of these metrics have their advantages and disadvantages when used for software effort estimation. Although the FPs-based measure has the advantage over the LOC in that it does not depend on the technologies used – in particular, the programming language, however, the assessment of efforts requires taking into account such factors (environmental factors). Taking into account the above factors can be ensured by appropriate models for estimating the LOC-based effort.

Nowadays many Web apps are created using PHP frameworks making the app development faster. CodeIgniter (https://www.codeigniter.com/) is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web apps. However, there are no regression models for estimating the software size of Web apps created using the CodeIgniter framework. There are some regression equations, both linear [14, 15] and nonlinear [13], for estimating the software size of information open-source PHP-based systems. Also, there are regression models for estimating the software size of Web apps created using some PHP frameworks such as CakePhp, Laravel, and Yii [18, 19]. This demands the construction of the models for early software size estimation of Web apps created using the CodeIgniter framework.

Although machine learning methods are becoming increasingly popular for software size estimation [20], methods based on nonlinear regression analysis have not yet reached their full potential [21, 22]. We suggest using the nonlinear regression models for estimating the size of Web apps created using the CodeIgniter framework because, firstly, there are two random variables, both a dependent variable (response) and an error term, in a regression model, and, secondly, the size (response) distribution is not Gaussian. We apply the technique for constructing nonlinear regression models based on the multivariate normalizing transformations and prediction intervals [23]. In this technique, prediction intervals of nonlinear regressions are used to detect the outliers in constructing a nonlinear regression model. Usually, the above process is iterative since we repeat building the model for new data after the outlier cutoff. If there are no outliers, the process of constructing the nonlinear regression model ends.

From the above, the following two questions are quite natural. The first question: is it possible to use existing nonlinear regression models [18, 19] for other PHP frameworks to estimate the software size of Web apps created using the CodeIgniter framework with a magnitude of relative error (MRE) of not more than 0.25? The second question: is it better to use the multivariate normalizing transformations in comparison to the univariate ones for building a nonlinear regression model to estimate the software size of Web apps created using the CodeIgniter framework? In this paper, we get answers to these questions. For these purposes, first, we apply existing nonlinear regression models [18, 19] for other PHP frameworks to estimate the software size of Web apps created using the CodeIgniter framework, second, we compare nonlinear regression models constructed by the well-known univariate and multivariate normalizing transformations for estimating the software size of Web apps created using the CodeIgniter framework.

1. Formulation of the problem

Suppose given the original sample as the four-dimensional non-Gaussian data set: actual software size in the thousand lines of code (KLOC) Y, the total number of classes X_1, the average number of methods per class X_2, the average of Depth of Inheritance Tree (DIT) per class X_3 in a class diagram from N Web apps. Suppose that there are a five-variate normalizing transformation of non-Gaussian random vector $\mathbf{P} = \{Y, X_1, X_2, X_3\}^T$ to Gaussian random vector $\mathbf{T} = \{Z_Y, Z_1, Z_2, Z_3\}^T$ given by

$$\mathbf{T} = \boldsymbol{\psi}(\mathbf{P}) \tag{1}$$

and the inverse transformation for (1)

$$\mathbf{P} = \boldsymbol{\psi}^{-1}(\mathbf{T}). \tag{2}$$

It is required to build the nonlinear regression model in the form $Y = Y(X_1, X_2, X_3, \varepsilon)$ based on the transformations (1) and (2).

The aim of this paper is to build the nonlinear regression model based on the transformations (1) and (2) for estimating the software size (in KLOC) of Web apps created using the CodeIgniter framework.

2. Problem Solution

To build a nonlinear regression model for estimating the size of Web apps created using the CodeIgniter framework, we collected data from 50 apps hosted on GitHub (https://github.com). We took 50 apps because it is generally accepted that the lower limit of a large sample is 30, and many models (for example, [11, 15]) were built for the data number from 30 to 50. We obtained the data set by the PhpMetrics tool (https://phpmetrics.org/) around the following variables: actual software size (in KLOC) Y, the total number of classes X_1, the average number of methods per class X_2, and the DIT average per class X_3. Table 1 contains that data set. As in [18, 19], we chose the above predictors X_1, X_2, and X_3 because these can be obtained from the class diagram.

We used values of variable Y and predictors from Table 1 to estimate the size of Web apps created using the CodeIgniter framework by existing nonlinear regression models [18, 19] for CakePhp and Yii frameworks. Although the PRED(0.25) value is 0.9 for 50 data rows from Table 1 if we use the model [18] for the CakePhp framework however five data rows (rows 11, 15, 23, 42, and 47) have magnitude relative error is greater than 0.958. The PRED(0.25) value is 0.06 for 50 data rows from Table 1 if we use the model [19] for the Yii framework. In other words, there are only three data rows (rows 15, 42, and 47) for which MRE is less than 0.25. The above indicates that it is not possible to use existing nonlinear regression models [18, 19] for all

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

data rows from Table 1 to estimate the size of Web apps created using the CodeIgniter framework with an MRE value of not more than 0.25. Hence, it is necessary to build a model for estimating the software size of Web apps created using the CodeIgniter framework.

Table 1

The data set and SMD values

No	Y	X_1	X_2	X_3	SMD	SMD_Z
1	42.068	142	11.66	1.33	3.50	3.94
2	37.94	132	10.89	1.29	2.18	1.76
3	39.073	138	10.80	1.31	0.56	0.40
4	40.487	143	10.76	1.33	0.49	1.24
5	38.994	140	10.72	1.31	0.80	0.24
6	119.424	274	12.70	1.17	18.72	11.64
7	39.269	146	10.38	1.30	3.15	2.22
8	41.85	142	11.23	1.30	0.23	0.17
9	127.696	330	11.52	1.19	10.77	7.16
10	155.740	420	11.33	1.21	**20.33**	11.17
11	2.280	32	0.31	1.71	10.66	10.48
12	38.912	141	10.60	1.31	1.13	0.40
13	38.326	134	10.85	1.31	0.48	0.94
14	39.699	139	10.81	1.31	0.49	0.68
15	2.265	31	0.19	1.75	7.95	10.00
16	32.390	97	13.31	1.26	3.05	2.66
17	45.540	177	9.91	1.38	7.85	5.91
18	33.368	101	13.45	1.25	2.76	4.83
19	37.754	134	10.79	1.31	0.66	0.49
20	28.333	83	13.41	1.24	4.45	4.76
21	40.908	140	11.04	1.33	0.75	2.79
22	39.276	144	10.44	1.33	0.68	0.33
23	2.361	35	0.23	1.77	8.40	10.53
24	**0.939**	**10**	**3.80**	**1.67**	9.39	**16.21**
25	38.431	147	10.43	1.38	6.01	5.87
26	54.052	187	11.41	1.29	2.17	10.21
27	41.347	150	10.49	1.34	1.06	1.25
28	38.654	132	11.10	1.29	1.33	1.20
29	39.817	152	10.33	1.32	1.87	2.13
30	28.653	83	13.40	1.29	6.58	9.44
31	41.236	150	10.49	1.30	2.34	1.33
32	32.778	100	13.10	1.26	2.43	2.34
33	38.867	135	11.01	1.31	0.26	0.43
34	29.738	94	12.84	1.24	4.02	5.64
35	39.404	146	10.44	1.33	0.86	0.20
36	33.373	97	13.75	1.26	4.76	4.18
37	38.763	134	11.01	1.31	0.25	0.75
38	44.253	155	11.25	1.31	0.91	2.30
39	34.230	106	12.86	1.31	6.15	5.87
40	39.205	148	10.26	1.31	2.63	1.71
41	39.191	142	10.62	1.35	1.57	2.46
42	2.097	30	0.20	1.75	7.98	7.55
43	128.355	341	11.38	1.20	9.80	6.16
44	39.611	143	10.65	1.31	0.98	0.31
45	39.565	142	10.89	1.31	0.57	0.39
46	38.184	135	10.79	1.30	1.38	0.89
47	2.347	34	0.18	1.77	8.36	9.11
48	41.800	154	10.49	1.33	1.17	0.56
49	31.365	98	12.89	1.25	2.69	3.40
50	32.836	101	13.19	1.26	2.44	3.39

To do this we checked the four-dimensional data from Table 1 for multivariate outliers. This is step 1 according to [23]. But before that, we tested the normality of multivariate data from Table 1 because well-known statistical methods (for example, multivariate outlier detection based on the squared Mahalanobis distance (SMD) [24]) are used to detect outliers in multivariate data under the assumption that the data is described by a Gaussian distribution [25, 26].

We applied a multivariate normality test proposed by Mardia [27, 28]. This test is based on measures of multivariate skewness β_1 and kurtosis β_2. According to the Mardia test [27], the distribution of four-dimensional data from Table I is not Gaussian since the test statistic for multivariate skewness $N\beta_1/6$ of this data, which equals 148.95, is greater than the quantile of the Chi-Square distribution, which is 40.00 for 20 degrees of freedom and 0.005 significance level. Analogically, the test statistic for multivariate kurtosis β_2, which equals 35.64, is greater than the value of the Gaussian distribution quantile, which is 29.05 for the mean of 24, the variance of 3.84, and 0.005 significance level.

Because we used the statistical technique [25] to detect multivariate outliers in the four-dimensional non-Gaussian data from Table 1 based on the multivariate normalizing transformations and the SMD for normalized data. To normalize the data from Table 1, we applied the four-variate Box-Cox transformation with components [24]

$$Z_j = x(\lambda_j) = \begin{cases} \left(X_j^{\lambda_j} - 1\right) / \lambda_j, & \text{if} \quad \lambda_j \neq 0; \\ \ln(X_j), & \text{if} \quad \lambda_j = 0. \end{cases} \tag{3}$$

Here Z_j is a Gaussian variable; λ_j is a parameter of the Box-Cox transformation, $j = 1,2,3$. The variable Z_Y is defined analogously (3) with the only difference that instead of Z_j, X_j, and λ_j should be put respectively Z_Y, Y, and λ_Y.

The parameter estimates of the four-variate Box-Cox transformation for the data from Table 1 are calculated by the maximum likelihood method according to [24] and are $\hat{\lambda}_Y = 0.0336755$, $\hat{\lambda}_1 = 0.1019055$, $\hat{\lambda}_2 = 1.047316$, $\hat{\lambda}_3 = 1.353001$.

Table 1 contains the SMD for normalized data (SMD_Z), which is transformed using the four-variate Box-Cox transformation. The SMD_Z values from Table 1 indicate that row 24 is a multivariate outlier in four-dimensional non-Gaussian data since the SMD_Z value for this data row is greater than the quantile of the Chi-Square distribution, which equals 14.86 for 4 degrees of freedom and 0.005 significance level. Note, for data without normalization, row 10 is the multivariate outlier since the SMD value for row 36 is the largest and greater than the above quantile. In Table 1, the above SMD and SMD_Z values are highlighted in bold.

We erased data row 24 as a multivariate outlier. The first iteration is completed. And we go to step 1 of the second iteration according to [23].

We checked the four-dimensional data from Table 1 (without row 24) for multivariate outliers. To do this, we normalized 49 data rows using the four-variate Box-Cox transformation with components, which are defined by (3). The parameter estimates of the four-variate Box-Cox transformation for 49 data rows from Table 1 (without row 24) are calculated by the maximum likelihood method according to [24] and are $\hat{\lambda}_Y = -0.0066662$, $\hat{\lambda}_1 = -0.00943335$, $\hat{\lambda}_2 = 1.1404485$, $\hat{\lambda}_3 = 1.4723061$. There is no multivariate outlier in four-dimensional non-Gaussian data from Table 1 (without row 24) since the SMD_Z values for 49 data rows are less than the quantile of the Chi-Square distribution, which equals 14.86 for 4 degrees of freedom and 0.005 significance level.

Next, we built a nonlinear regression model based on the four-variate Box-Cox transformation for 49 data rows. The nonlinear regression model with three predictors for estimating the size of Web apps created using the CodeIgniter framework is built based on the four-variate Box-Cox transformation for 49 data rows from Table 1 (without row 24) and has the form

$$Y = \left[\hat{\lambda}_Y\left(\hat{Z}_Y + \varepsilon\right) + 1\right]^{1/\hat{\lambda}_Y}, \qquad (4)$$

where ε is a Gaussian random variable, $\varepsilon \sim N\left(0, \sigma_\varepsilon^2\right)$, with the estimate $\hat{\sigma}_\varepsilon$ of 0.0272; $\hat{Z}_Y$ is a prediction result by the linear regression equation $\hat{Z}_Y = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3$ for normalized data, which are transformed by the four-variate Box-Cox transformation with components (3); $\hat{b}_0 = -2.7220$, $\hat{b}_1 = 1.207254$, $\hat{b}_2 = 0.058823$, $\hat{b}_3 = -0.628765$.

After constructing a model (4), we have to find the nonlinear regression prediction interval [23]

$$\psi_Y^{-1}\left(\hat{Z}_Y \pm t_{\alpha/2,\nu} S_{Z_Y}\left\{1 + \frac{1}{N} + \left(\mathbf{z}_X^+\right)^T \mathbf{S}_Z^{-1}\left(\mathbf{z}_X^+\right)\right\}^{1/2}\right), \qquad (5)$$

where ψ_Y is the transformation (3) for Y, $\psi_Y^{-1} = \left(\hat{\lambda}_Y Z_Y + 1\right)^{1/\hat{\lambda}_Y}$; $t_{\alpha/2,\nu}$ is a student's t-distribution quantile with $\alpha/2$ significance level and ν degrees of freedom; $\nu = N - k - 1$; k is a number of independent variables (in our case, k is 3); $\mathbf{z}_X^+$ is a vector with components $Z_{1_i} - \overline{Z}_1$, $Z_{2_i} - \overline{Z}_2$, $Z_{3_i} - \overline{Z}_3$ for i-row; $\overline{Z}_j = \frac{1}{N}\sum_{i=1}^{N} Z_{j_i}$, $S_{Z_Y}^2 = \frac{1}{\nu}\sum_{i=1}^{N}\left(Z_{Y_i} - \hat{Z}_{Y_i}\right)^2$, $j = 1,2,3$; $\mathbf{S}_Z$ is a 3×3 matrix

$$\mathbf{S}_Z = \begin{pmatrix} S_{Z_1Z_1} & S_{Z_1Z_2} & S_{Z_1Z_3} \\ S_{Z_1Z_2} & S_{Z_2Z_2} & S_{Z_2Z_3} \\ S_{Z_1Z_3} & S_{Z_2Z_3} & S_{Z_3Z_3} \end{pmatrix}. \qquad (6)$$

In (6) $S_{Z_qZ_r} = \sum_{i=1}^{N}\left[Z_{q_i} - \overline{Z}_q\right]\left[Z_{r_i} - \overline{Z}_r\right]$, $q, r = 1,2,3$.

For the data normalized by the four-variate Box-Cox transformation from 49 Web apps, the matrix (6) is the following:

$$\mathbf{S}_Z = \begin{pmatrix} 13.516 & 79.493 & -3.733 \\ 79.493 & 975.021 & -36.849 \\ -3.733 & -36.849 & 1.500 \end{pmatrix}.$$

Table 2 contains the values of lower (LB) and upper (UB) bounds of the nonlinear regression prediction interval calculated by (5) based on the four-variate Box-Cox transformation for 0.05 significance level in the second iteration for 49 data rows from Table 1 (without row 24). In Table 2, we denoted LB and UB in the second iteration as LB_2 and UB_2, respectively.

As we observe in Table 2, there are two values of *Y* for Web apps 26 and 30 (rows 26 and 30) that are out of the prediction intervals computed by (5) for a significance level of 0.05. Next, we erased data rows 26 and 30. In Table 2, we highlighted the data outliers in bold, and a dash (-) shows the exception of the relevant data. The second iteration is completed. And we go to step 1 of the third iteration according to [23].

We checked the four-dimensional data from Table 1 (without rows 24, 26, and 30) for multivariate outliers. To do this, we normalized 47 data rows using the four-variate Box-Cox transformation with components, which are defined by (3). The parameter estimates of the four-variate Box-Cox transformation for 47 data rows from Table 1 (without data rows 24, 26, and 30) are calculated by the maximum likelihood method according to [24] and are $\hat{\lambda}_Y = -0.036072$, $\hat{\lambda}_1 = -0.0806416$, $\hat{\lambda}_2 = 1.151465$, $\hat{\lambda}_3 = 1.613450$. There is no multivariate outlier in four-dimensional non-Gaussian data from Table 1 (without rows 24, 26, and 30) since the SMD_Z values for 47 data rows are less

 ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

than the quantile of the Chi-Square distribution, which equals 14.86 for 4 degrees of freedom and 0.005 significance level.

Table 2
LB and UB of nonlinear regression prediction intervals in various iterations

No	*Y*	The second iteration		The fourth iteration	
		LB_2	UB_2	LB_4	UB_4
1	42.068	39.778	44.942	40.497	44.446
2	37.94	35.253	39.715	35.845	39.222
3	39.073	36.370	40.903	36.984	40.422
4	40.487	37.247	41.901	37.874	41.409
5	38.994	36.743	41.327	37.374	40.852
6	119.424	106.122	119.813	109.708	120.752
7	39.269	37.764	42.553	38.457	42.101
8	41.85	39.297	44.191	40.019	43.749
9	127.696	117.525	133.171	120.059	132.588
10	155.74	151.331	171.934	152.640	169.040
11	2.28	2.183	2.482	2.190	2.393
12	38.912	36.677	41.262	37.310	40.790
13	38.326	35.277	39.673	35.848	39.175
14	39.699	36.713	41.287	37.339	40.810
15	**2.265**	2.028	2.296	-	-
16	32.39	30.757	34.698	30.929	33.880
17	45.54	42.837	48.527	43.540	47.892
18	33.368	32.893	37.106	33.174	36.348
19	37.754	35.098	39.475	35.665	38.977
20	28.333	26.129	29.567	26.018	28.548
21	40.908	37.183	41.854	37.802	41.351
22	39.276	36.565	41.120	37.175	40.631
23	2.361	2.303	2.611	2.332	2.542
25	38.431	35.948	40.696	36.509	40.110
26	**54.052**	**55.583**	**62.623**	-	-
27	41.347	38.232	43.030	38.883	42.534
28	38.654	35.899	40.408	36.503	39.919
29	39.817	38.895	43.756	39.592	43.296
30	**28.653**	**25.192**	**28.493**	-	-
31	41.236	39.362	44.325	40.104	43.888
32	32.778	31.321	35.314	31.548	34.541
33	38.867	36.075	40.563	36.669	40.070
34	29.738	28.831	32.604	28.954	31.755
35	39.404	37.164	41.796	37.793	41.312
36	33.373	31.929	36.079	32.122	35.248
37	38.763	35.759	40.208	36.341	39.711
38	44.253	43.317	48.755	44.174	48.348
39	34.23	31.644	35.802	31.928	35.060
40	39.205	37.729	42.483	38.408	42.025
41	39.191	35.948	40.492	36.518	39.964
42	2.097	1.952	2.211	1.952	2.125
43	128.355	119.862	135.810	122.127	134.874
44	39.611	37.452	42.128	38.110	41.663
45	39.565	37.906	42.626	38.573	42.162
46	38.184	35.654	40.129	36.254	39.644
47	2.347	2.221	2.518	2.245	2.447
48	41.8	39.738	44.708	40.445	44.236
49	31.365	30.235	34.118	30.427	33.324
50	32.836	31.938	36.013	32.191	35.252

Next, we built a nonlinear regression model (4) based on the four-variate Box-Cox transformation for 47 data rows. In this case in model (5) the estimate $\hat{\sigma}_\varepsilon$ equals 0.01997, $\hat{b}_0 = -3.38010$, $\hat{b}_1 = 1.562030$, $\hat{b}_2 = 0.0511177$, $\hat{b}_3 = -0.562295$.

After constructing a model (4), we have to find the nonlinear regression prediction interval using (5). For the data normalized by the four-variate Box-Cox transformation from 47 Web apps, the matrix (6) is the following:

$$\mathbf{S}_Z = \begin{pmatrix} 6.937 & 60.683 & -2.904 \\ 60.683 & 990.492 & -39.368 \\ -2.904 & -39.368 & 1.674 \end{pmatrix}.$$

There is one value of *Y* for Web app 15 (row 15) that is out of the prediction intervals computed by (5) for a significance level of 0.05. Next, we erased data row 15. The third iteration is completed. And we go to step 1 of the fourth iteration according to [23].

We checked the four-dimensional data from Table 1 (without rows 15, 24, 26, and 30) for multivariate outliers. To do this, we normalized 46 data rows using the four-variate Box-Cox transformation with components, which are defined by (3). The parameter estimates of the four-variate Box-Cox transformation for 46 data rows from Table 1 (without data rows 15, 24, 26, and 30) are calculated by the maximum likelihood method according to [23] and are $\hat{\lambda}_Y = -0.0483054$, $\hat{\lambda}_1 = -0.121859$, $\hat{\lambda}_2 = 1.18457$, $\hat{\lambda}_3 = 1.57566$. There is no multivariate outlier in four-dimensional non-Gaussian data from Table 1 (without rows 15, 24, 26, and 30) since the SMD_Z values for 46 data rows are less than the quantile of the Chi-Square distribution, which equals 14.86 for 4 degrees of freedom and 0.005 significance level.

Next, we built a nonlinear regression model (4) based on the four-variate Box-Cox transformation for 46 data rows. In this case in the model (5) the estimate $\hat{\sigma}_\varepsilon$ equals 0.01761, $\hat{b}_0 = -3.90201$, $\hat{b}_1 = 1.84301$, $\hat{b}_2 = 0.045971$, $\hat{b}_3 = -0.553901$.

After constructing a model (4), we calculated the nonlinear regression prediction interval for 46 data rows in the fourth iteration (see Table 2). For the data normalized by the four-variate Box-Cox transformation from 46 Web apps, the matrix (6) is the following:

$$\mathbf{S}_Z = \begin{pmatrix} 6.937 & 60.683 & -2.904 \\ 60.683 & 990.492 & -39.368 \\ -2.904 & -39.368 & 1.674 \end{pmatrix}.$$

In Table 2, we denoted LB and UB in the fourth iteration as LB_4 and UB_4, respectively. The LB_4 and UB_4

values indicate there are no values of Y for 46 data rows that are out of the prediction intervals computed by (5) for a significance level of 0.05. Because we completed the stages' iterations and constructed a nonlinear regression model (4) with 46 Web apps data.

Also, to estimate the size of Web apps created using the CodeIgniter framework, we built two nonlinear regression models with three predictors based on the univariate normalizing transformations (the decimal logarithm, and the Box-Cox transformation) for the same 46 Web apps data from Table 1.

The nonlinear regression model with three predictors based on the univariate Box-Cox transformation has the form (4) too, but with the only difference that parameters estimates are the following: $\hat{\lambda}_Y = 0.46420$, $\hat{\lambda}_1 = 0.343935$, $\hat{\lambda}_2 = 1.92580$, $\hat{\lambda}_3 = -6.95771$, $\hat{b}_0 = 9.02530$, $\hat{b}_1 = 1.11051$, $\hat{b}_2 = -0.00083422$, $\hat{b}_3 = -111.6498$, $\hat{\sigma}_\varepsilon = 0.27344$.

For the data normalized by the univariate Box-Cox transformation from 46 Web apps, the matrix (6) is the following:

$$\mathbf{S}_Z = \begin{pmatrix} 320.282 & 1035.442 & -0.79465 \\ 1035.442 & 16281.89 & -6.3029 \\ -0.79465 & -6.3029 & 0.003834 \end{pmatrix}.$$

The nonlinear regression model based on the decimal logarithm transformation has the form

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}, \tag{7}$$

where the estimators for parameters are: $\hat{b}_0 = -0.015841$, $\hat{b}_1 = 0.920457$, $\hat{b}_2 = 0.094150$, $\hat{b}_3 = -3.90152$. The estimate $\hat{\sigma}_\varepsilon$ is 0.02687.

For the data normalized by the decimal logarithm transformation from 46 Web apps, the matrix (6) is the following:

$$\mathbf{S}_Z = \begin{pmatrix} 2.354 & 3.890 & -0.336 \\ 3.890 & 10.663 & -0.834 \\ -0.336 & -0.834 & 0.0734 \end{pmatrix}.$$

To evaluate the prediction accuracy of the nonlinear regression models we applied the standard metrics R^2, MMRE, and PRED(0.25). MMRE and PRED(0.25) are accepted as standard evaluations of prediction results by regression models. These metrics are applied in software engineering too [29, 30]. The acceptable values of MMRE and PRED(0.25) are not more than 0.25 and not less than 0.75 respectively. The values of R^2, MMRE and PRED(0.25) are shown in Table 3 for models (4) for both the univariate and four-variate Box-Cox transformations, and model (7).

Table 3

The prediction accuracy metrics
of the nonlinear regression models

Metrics	univariate		bivariate
	Log10	Box-Cox	Box-Cox
R^2	0.9975	0.9946	0.9984
MMR_{min}	0.0031	0.0011	0.0002
MMR_{max}	0.9719	0.2657	0.0443
MMRE	0.0492	0.0452	0.0167
PRED(0.25)	0.8261	0.9783	1.0000

The values of these metrics are acceptable for all models. These values indicate good prediction accuracy of the nonlinear regression models (4) and (7) for estimating the size of Web apps created using the CodeIgniter framework. However, model (4) based on the four-variate Box-Cox transformation has the best R^2, MMRE and PRED(0.25) values.

Also, Table 3 contains minimum and maximum values of MRE denoted MMR_{min} and MMR_{max}, respectively. As we observe in Table 3, we have the smallest MMR_{max} value for model (4) based on the four-variate Box-Cox transformation. The above indicates the advantages of using model (4) based on the four-variate Box-Cox transformation for estimating the size of Web apps created using the CodeIgniter framework.

The advantage of using model (4) based on the four-variate Box-Cox transformation in comparison to other constructed models based on univariate transformations is also indicated by the width of the confidence and prediction intervals. We calculated the confidence intervals of regressions by (5) with the only difference that in the sum in curly brackets, there is not 1.

The widths of the confidence interval of nonlinear regression based on the Box-Cox four-variate transformation are less than for nonlinear regression based on the Box-Cox univariate transformation for all 46 data rows (with a difference from 28 up to 810%). Also, the widths of the confidence interval of nonlinear regression based on the Box-Cox four-variate transformation are less than for nonlinear regression based on the decimal logarithm univariate transformation for 41 (with the difference up to 398%) from 46 data rows (except rows 45, 46, and 48-50 with the difference up to 59%).

Approximately the same results are obtained for the prediction intervals of nonlinear regressions. The widths of the prediction interval of nonlinear regression based on the Box-Cox four-variate transformation are less than for nonlinear regression based on the Box-Cox univariate transformation for all 46 data rows (with a

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

difference from 13 up to 841 %). Also, the widths of the prediction interval of nonlinear regression based on the Box-Cox four-variate transformation are less than for nonlinear regression based on the decimal logarithm univariate transformation for 41 (with the difference up to 265 %) from 46 data rows (except rows 45, 46, and 48-50 with the difference up to 52 %).

The above indicates that it is better to use the multivariate normalizing transformations in comparison to the univariate ones for building a nonlinear regression model to estimate the software size of Web apps created using the CodeIgniter framework.

Conclusions

Early size estimation (in KLOC) of Web apps created using the CodeIgniter framework by nonlinear regression models with three predictors based on the normalizing transformations, both univariate and multivariate ones, is performed. The nonlinear regression model constructed using the four-variate Box-Cox transformation has better size prediction results compared to other regression ones based on the univariate transformations (the decimal logarithm and Box-Cox).

To construct nonlinear regression models with multiple predictors for estimating the software size of Web apps created using the CodeIgniter framework, it needs to apply multivariate normalizing transformations and outlier detection.

Prospects for further research may include the application of other multivariate normalizing transformations and data sets from at least 100 apps to construct nonlinear regression models for estimating the size of Web apps created using the specific frameworks.

Contribution of authors: advising on the process of formulating the problem, constructing the models, and analyzing the data – **Dr. Sergiy Prykhodko**; constructing the model based on the univariate transformations and carrying out the analysis – **Ivan Shutko**; constructing the model based on the four-variate transformation and carrying out the analysis – **Andrii Prykhodko**.

All the authors have read and agreed to the published version of the manuscript.

Acknowledgment. The authors are grateful to **Bogdan Kollar** for his assistance in data collection.

Literature

1. Software cost estimation with COCOMO II [Text] / B. W. Boehm, C. Abts, A. W. Brown, et al. – Upper Saddle River, NJ : Prentice Hall PTR, 2000. – 506 p.

2. Jorgensen, M. A systematic review of software development cost estimation studies [Text] / M. Jorgensen, M. Shepperd // IEEE Transactions on Software Engineering. – 2007. – Vol. 33, No. 1. – P. 33-53.

3. Ruhe, M. Cost estimation for Web applications [Text] / M. Ruhe, R. Jeffery, I. Wieczorek // Proceedings of the International Conference on Software Engineering, 2003. – P. 285–294.

4. Mendes, E. Web effort estimation [Text] / E. Mendes, N. Mosley, S. Counsell // In Web engineering, Emilia Mendes and Nile Mosley (Eds.). – Springer, 2006. –P. 29-73.

5. Trendowicz, A. Software project effort estimation: Foundations and best practice guidelines for success [Text] / A. Trendowicz, R. Jeffery. – Springer International Publishing, 2014. – 491 p. DOI: 10.1007/978-3-319-03629-8.

6. Laird, L. M. Software measurement and estimation. A practical approach. quantitative software engineering series [Text] / L. M. Laird, M. C. Brennan. – Wiley-IEEE Computer Society Press, 2006. – 379 p.

7. Dewi Sholiq, R.S. A comparative study of software development size estimation method: UCPabc vs Function Points [Text] / R. S. Dewi Sholiq, A. P. Subriadi // Procedia Computer Science, – 2017. – Vol. 124, – P. 470-477. DOI: 10.1016/j.procs.2017.12.179.

8. Zifen, Y. An improved software size estimation method based on object-oriented approach [Text] / Y. Zifen // Electrical & Electronics Engineering : IEEE Symposium EEESYM'12, Kuala Lumpur, Malaysia, 24-27 June, 2012 : proceedings. – IEEE, 2012. – P. 615-617. DOI: 10.1109/EEESym.2012.6258733.

9. Daud, M. Improving the accuracy of early software size estimation using analysis-to-design adjustment factors (ADAFs) [Text] / M. Daud, A. A. Malik // IEEE Access. – 2021. – Vol. 9. – P. 81986-81999. DOI: 10.1109/ACCESS.2021.3085752.

10. Efficiency improvement of function point-based software size estimation with deep learning model [Text] / K. Zhang, X. Wang, J. Ren, C. Liu // IEEE Access. – 2021. – Vol. 9. – P. 107124-107136. DOI: 10.1109/ACCESS.2020.2998581.

11. Kiewkanya, M. Constructing C++ software size estimation model from class diagram [Text] / M. Kiewkanya, S. Surak // Computer Science and Software Engineering : 13th International Joint Conference, Khon Kaen, Thailand, July 13-15, 2016 : proceedings. – IEEE, 2016. – P. 1-6. DOI: 10.1109/JCSSE.2016.7748880.

12. Kaczmarek, J. Size and effort estimation for applications written in Java [Text] / J. Kaczmarek, M. Kucharski // Information and Software Technology. – 2004. – Vol. 46, Iss. 9. – P. 589-601. DOI: 10.1016/j.infsof.2003.11.001.

13. Prykhodko, S. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis [Text] / S. Prykhodko, N. Prykhodko, L. Makarova // Proceedings of International Conference "Advanced Computer

Information Technologies" (ACIT-2018). CEUR Workshop Proceedings. – 2019. – Vol. 2300. – Ceske Budejovice, CZECH REPUBLIC. CEUR-WS.org – P. 199-202. ISSN 1613-0073.

14. Tan, H. B. K. Estimating LOC for information systems from their conceptual data models [Text] / H. B. K. Tan, Y. Zhao, H. Zhang // Software Engineering : the 28th International Conference (ICSE '06), Shanghai, China, May 20-28, 2006 : proceedings. – P. 321-330. DOI: 10.1145/1134285.1134331.

15. Tan, H. B. K. Conceptual data model-based software size estimation for information systems [Text] / H. B. K. Tan, Y. Zhao, H. Zhang // Transactions on Software Engineering and Methodology. – 2009. – Vol. 19, Issue 2. – October 2009. – Article No. 4. DOI: 10.1145/1571629.1571630.

16. CompSize: Automated size estimation of embedded software components [Text] / K. Lind, R. Heldal, T. Harutyunyan, T. Heimdahl // Proceedings from Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement. – Nara, Japan, 2011. – P. 86-95. DOI: 10.1109/IWSM-MENSURA.2011.49.

17. An approach to estimate the size of Web application using IFML User interface model [Text] / V. R. N. Neyveli, S. S. Sivakumar, D. Arunagiri, et al. // Proceedings from AICAI'19: Amity International Conference on Artificial Intelligence. Dubai, United Arab Emirates, 2019, – P. 292-295. DOI: 10.1109/AICAI. 2019.8701268.

18. Prykhodko, S. Estimating the Size of Web Apps Created Using the CakePHP Framework by Nonlinear Regression Models with Three Predictors [Text] / S. Prykhodko, A. Prykhodko, I. Shutko // 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), 2021. – P. 333-336. DOI: 10.1109/CSIT52700.2021.9648680.

19. Prykhodko, S. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models [Text] / S. Prykhodko, I. Shutko, A. Prykhodko // WSEAS Transactions on Computers. – 2021. – Vol. 20. – P. 321-328. DOI: 10.37394/23205. 2021.20.35.

20. Manisha. Early size estimation using machine learning [Text] / Manisha, R. Rishi // 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021. – P. 757-762.

21. Nassif, A. B. Software effort estimation from Use Case diagrams using nonlinear regression analysis [Text] / A. B. Nassif, M. AbuTalib, L. F. Capretz // Proceedings from CCECE'20: IEEE Canadian Conference on Electrical and Computer Engineering. London, ON, Canada, 2020. – P. 1-4. DOI: 10.1109/CCECE47787. 2020.9255712.

22. Sharma, A. The combined model for software development effort estimation using polynomial regression for heterogeneous projects [Text] / A. Sharma, N. Chaudhary // Radioelectronic and computer systems. – 2022. – Vol. 2. – P. 75-82. DOI: 10.32620/reks. 2022.2.06.

23. Prykhodko, S. Mathematical modeling of non-Gaussian dependent random variables by nonlinear regression models based on the multivariate normalizing transformations [Text] / S. Prykhodko, N. Prykhodko // 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.

24. Johnson, R.A. Applied multivariate statistical analysis [Text] / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.

25. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations [Text] / S. Prykhodko, N. Prykhodko, L. Makarova, et al. // Electrical and Computer Engineering : the 2017 IEEE First Ukraine Conference (UKRCON) «Celebrating 25 Years of IEEE Ukraine Section», Kyiv, Ukraine, May 29 – June 2, 2017 : proceedings. – Kyiv : IEEE, 2017. – P. 846-849. DOI: 10.1109/UKRCON.2017.8100366.

26. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data [Text] / S. Prykhodko, N. Prykhodko, L. Makarova, et al. // Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering : 14th International Conference (TCSET), Lviv-Slavske, Ukraine, February 20–24, 2018 : proceedings. – P. 962-965. DOI: 10.1109/TCSET.2018.8336353.

27. Mardia, K. V. Measures of multivariate skewness and kurtosis with applications [Text] / K. V. Mardia // Biometrika. – 1970. – Vol. 57. – P. 519–530. DOI: 10.1093/biomet/57.3.519.

28. Mardia, K. V. Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies [Text] / K. V. Mardia // Sankhya: The Indian Journal of Statistics, Series B (1960–2002). – 1974. – Vol. 36, Issue 2. – P. 115–128.

29. A simulation study of the model evaluation criterion MMRE [Text] / T. Foss, E. Stensrud, B. Kitchenham, et al. // IEEE Transactions on software engineering. – 2003. – Vol. 29, Iss. 11. – P. 985–995. DOI: 10.1109/TSE.2003.1245300.

30. Port, D. Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research [Text] / D. Port, M. Korte // Empirical Software Engineering and Measurement : the 2nd ACM-IEEE International Symposium ESEM, Kaiserslautern, Germany, October, 2008 : proceedings. – New York: ACM, 2008. – P. 51–60.

References

1. Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D., Steece, B. *Software cost estimation with COCOMO*

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

II, Upper Saddle River, NJ, Prentice Hall PTR, 2000. 506 p.

2. Jorgensen, M., Shepperd M. A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 2007, vol. 33, no. 1, pp. 33-53.

3. Ruhe, M., Jeffery, R., Wieczorek, I. Cost estimation for Web applications. *Proceedings of the International Conference on Software Engineering*, 2003, pp. 285–294.

4. Mendes, E., Mosley, N., Counsell, S. Web effort estimation, In *Web engineering*, Emilia Mendes and Nile Mosley (Eds.). Springer, 2006, pp. 29-73.

5. Trendowicz, A., Jeffery, R. *Software project effort estimation: Foundations and best practice guidelines for success*, Springer International Publishing, 2014. 491 p. DOI: 10.1007/978-3-319-03629-8.

6. Laird, L. M., Brennan, M. C. *Software measurement and estimation. A practical approach. quantitative software engineering series*, Wiley-IEEE Computer Society Press, 2006. 379 p.

7. Dewi Sholiq, R. S., Subriadi, A. P. A comparative study of software development size estimation method: UCPabc vs Function Points. *Procedia Computer Science*, 2017, vol. 124, pp. 470-477. DOI: 10.1016/j.procs.2017.12.179.

8. Zifen, Y. An improved software size estimation method based on object-oriented approach. *Proceedings of IEEE Symposium on Electrical & Electronics Engineering, EEESYM*'12, Kuala Lumpur, Malaysia, 24-27 June, 2012, IEEE, 2012, pp. 615-617. DOI: 10.1109/EEESym.2012.6258733.

9. Daud, M., Malik, A. A. Improving the accuracy of early software size estimation using analysis-to-design adjustment factors (ADAFs). *IEEE Access*, 2021, vol. 9, pp. 81986-81999. DOI: 10.1109/ACCESS. 2021.3085752.

10. Zhang, K., Wang, X., Ren, J., Liu, C. Efficiency improvement of function point-based software size estimation with deep learning model. *IEEE Access*, 2021, vol. 9, pp. 107124-107136. DOI: 10.1109/ACCESS.2020.2998581.

11. Kiewkanya, M., Surak, S. Constructing C++ software size estimation model from class diagram. *Proceedings of the 13th International Joint Conference on Computer Science and Software Engineering*, Khon Kaen, Thailand, July 13-15, 2016, IEEE, 2016, pp. 1-6. DOI: 10.1109/JCSSE.2016.7748880.

12. Kaczmarek, J., Kucharski, M. Size and effort estimation for applications written in Java. *Information and Software* Technology, 2004, vol. 46, iss. 9, pp. 589-601. DOI: 10.1016/j.infsof.2003.11.001.

13. Prykhodko, S., Prykhodko, N., Makarova, L. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis. *Proceedings of International Conference "Advanced Computer Information Technologies" (ACIT-2018). CEUR Workshop Proceedings*, 2019, vol. 2300, Ceske Budejovice, CZECH REPUBLIC. CEUR-WS.org, pp. 199-202. ISSN 1613-0073.

14. Tan, H. B. K., Zhao, Y., Zhang, H. Estimating LOC for information systems from their conceptual data models. *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*, Shanghai, China, May 20-28, 2006, pp. 321-330. DOI: 10.1145/1134285.1134331.

15. Tan, H. B. K., Zhao, Y., Zhang, H. Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*, 2009, vol. 19, iss. 2, October 2009, Article No. 4. DOI: 10.1145/1571629.1571630.

16. Lind, K., Heldal, R., Harutyunyan, T., Heimdahl, T. CompSize: Automated size estimation of embedded software components. *Proceedings from Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*. – Nara, Japan, 2011, pp. 86-95. DOI: 10.1109/IWSM-MENSURA.2011.49.

17. Neyveli, V. R. N., Sivakumar, S. S., Arunagiri, D., Arumugam, C. Veeramani, A. M. An approach to estimate the size of Web application using IFML User interface model. *Proceedings from AICAI'19: Amity International Conference on Artificial Intelligence*, Dubai, United Arab Emirates, 2019, pp. 292-295. DOI: 10.1109/AICAI.2019.8701268.

18. Prykhodko, S., Prykhodko, A., Shutko, I. Estimating the Size of Web Apps Created Using the CakePHP Framework by Nonlinear Regression Models with Three Predictors. *Proceedings of the 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, Lviv, Ukraine, 2021, pp. 333-336. DOI: 10.1109/CSIT52700. 2021.9648680.

19. Prykhodko, S., Shutko, I., Prykhodko, A. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models. *WSEAS Transactions on Computers*, 2021, vol. 20, pp. 321-328. DOI: 10.37394/23205.2021.20.35.

20. Manisha, Rishi, R. Early size estimation using machine learning. *Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 757-762.

21. Nassif, A. B., Abu Talib, M., Capretz, L. F. Software effort estimation from Use Case diagrams using nonlinear regression analysis. *Proceedings from CCECE'20: IEEE Canadian Conference on Electrical and Computer Engineering*, London, ON, Canada, 2020, pp. 1-4. DOI: 10.1109/CCECE47787. 2020.9255712.

22. Sharma, A., Chaudhary, N. The combined model for software development effort estimation using polynomial regression for heterogeneous projects. *Radioelectronic and computer systems*, 2022, vol. 2, pp. 75-82. DOI: 10.32620/reks.2022.2.06.

23. 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, pp. 166-174. (Advances in Intelligent Systems and Computing, vol. 1265). DOI: 10.1007/978-3-030-58124-4_16.

24. Johnson, R. A., Wichern, D. W. *Applied multivariate statistical analysis.* Pearson Prentice Hall, 2007. 800 p.

25. Prykhodko, S., Prykhodko, N., Makarova, L., Pugachenko, K. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations. *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) «Celebrating 25 Years of IEEE Ukraine Section»*, Kyiv, Ukraine, May 29 – June 2, 2017. Kyiv, IEEE, 2017, pp. 846-849. DOI: 10.1109/UKRCON.2017.8100366.

26. Prykhodko, S., Prykhodko, N., Makarova, L., Pukhalevych, A. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data. *Proceedings of the 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, February 20–24, 2018, IEEE, 2018, pp. 962-965. DOI: 10.1109/TCSET.2018.8336353.

27. Mardia, K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 1970, Vol. 57, pp. 519–530. DOI: 10.1093/biomet/57.3.519.

28. 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, pp. 115–128.

29. Foss, T., Stensrud, E., Kitchenham, B., Myrtveit, I. A simulation study of the model evaluation criterion MMRE. *IEEE Transactions on software engineering*, 2003, Vol. 29, Issue 11, pp. 985–995. DOI: 10.1109/TSE.2003.1245300.

30. Port, D., Korte, M. Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Kaiserslautern, Germany, October 2008, New York, ACM, 2008, pp. 51–60.

Надійшла до редакції 20.07.2022, розглянута на редколегії 25.08.2022

РАННЯ ОЦІНКА РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ CODEIGNITER, ЗА ДОПОМОГОЮ НЕЛІНІЙНИХ РЕГРЕСІЙНИХ МОДЕЛЕЙ

С. Б. Приходько, І. С. Шутко, А. С. Приходько

Тема: Раннє оцінювання розміру програмного забезпечення є однією з серйозних проблем керівників проектів при оцінці зусиль розробки додатків, оскільки розмір програмного забезпечення є основним фактором, що визначає зусилля з розробки програмного забезпечення. Функціональні точки (FP) і рядки коду (LOC) найчастіше використовуються як міра розміру в існуючих методах і моделях оцінювання трудомісткості програмного забезпечення. Як відомо, обидві ці метрики мають свої переваги та недоліки при використанні для оцінювання трудомісткості розробки програмного забезпечення. Хоча міра на основі FP має перевагу перед LOC в тому, що вона не залежить від використовуваних технологій, проте оцінювання трудомісткості вимагає врахування таких факторів (факторів навколишнього середовища). Врахування перерахованих вище факторів може бути забезпечена відповідними моделями оцінювання трудомісткості на основі LOC. В даний час багато веб-застосунків створюються з використанням PHP фреймворків, що прискорює розробку додатків. CodeIgniter – один із таких потужних фреймворків. Однак немає регресійних моделей для оцінювання розміру програмного забезпечення веб-застосунків, що створюються з використанням фреймворку CodeIgniter. Це потребує побудови відповідних моделей. **Завдання цієї статті** – побудувати модель нелінійної регресії для оцінювання розміру програмного забезпечення (у KLOC, тисячах рядків коду) веб-додатків, що створюються за допомогою фреймворку CodeIgniter. **Метод**: Ми застосовуємо метод побудови нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень та інтервалів прогнозування. **Результатом** є три нелінійні регресійні моделі з трьома предикторами: загальна кількість класів, середня кількість методів на клас та середнє значення DIT (дерева глибини спадкування) на клас. Щоб побудувати ці моделі для оцінювання розміру веб-застосунків, що створюються за допомогою фреймворку CodeIgniter, ми використовували три відомі нормалізуючі перетворення: два одновимірні перетворення (де-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

сятковий логарифм і перетворення Бокса-Кокса) і чотиривимірне перетворення Бокса-Кокса. **Висновки**. Модель нелінійної регресії, побудована за допомогою чотиривимірного перетворення Бокса-Кокса, має найкращі результати прогнозування розміру порівняно з іншими регресійними моделями, що ґрунтуються на одновимірних перетвореннях.

Ключові слова: оцінювання розміру програмного забезпечення; веб-застосунок; CodeIgniter; фреймворк; нелінійна регресійна модель; нормалізуюче перетворення; негаусівські дані.

Приходько Сергій Борисович – д-р техн. наук, проф., зав. каф. програмного забезпечення автоматизованих систем, Національний університет кораблебудування імені адмірала Макарова, Миколаїв, Україна.

Шутко Іван Сергійович – асп. каф. програмного забезпечення автоматизованих систем, Національний університет кораблебудування імені адмірала Макарова, Миколаїв, Україна.

Приходько Андрій Сергійович – магістр з інженерії програмного забезпечення, інженер-програміст, GlobalLogic Inc, Харків, Україна.

Sergiy Prykhodko – Dr. Sc. in Mathematical Modeling and Computational Methods, Professor, Head of Department of Software for Automated Systems, Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine,
e-mail: sergiy.prykhodko@nuos.edu.ua, ORCID: 0000-0002-2325-018X, Scopus Author ID: 55225622100,
ResearcherID: B-5544-2015.

Ivan Shutko – Ph.D. student of Department of Software for Automated Systems, Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine,
e-mail: ishutko@gmail.com, ORCID: 0000-0001-8584-113X, Scopus Author ID: 57456779600,
ResearcherID: GNM-9126-2022.

Andrii Prykhodko – Master in Software Engineering, Software Engineer, GlobalLogic Inc., Kharkiv, Ukraine,
e-mail: whiterandrek@gmail.com, ORCID: 0000-0002-7109-5508, Scopus Author ID: 57457099600,
ResearcherID: GNM-8058-2022.