# Three-factor nonlinear regression model of estimating the size of java-software

Oleksandr Oriekhov[1,*] , Tetyana Farionova[1,*] and Liubava Chernova[1]

[1] *Makarov National University of Shipbuilding, Ukraine, Heroes avenue, 9, Mykolaiv, 54007, Ukraine*

## Abstract

This paper proposes a three-factor nonlinear regression model for Java-software size estimation. Java is one of the most widely used programming languages in the world and is actively used in the development of various software projects. The software size estimation plays a key planning role at the early stages of software project planning for the successful implementation of software development projects. The software size estimation is valuable information that is used to predict software development effort estimation using parametric models such as COCOMO, COCOMO II, COSYSMO, etc. The aim of the study is to build the three-factor nonlinear regression model for early KLOC size estimation of Java-software on the basis of multivariate Johnson SB family normalizing transformations to increase the reliability and accuracy of the software size estimation at the early stage of software project planning using UML class diagram metrics. The object of the study is the process of size estimation for open-source Java-software. The subject of the study is the nonlinear regression models to estimate the software size. To achieve this goal we collected software code metrics information from 571 open-source Java software projects. The obtained dataset is split up in training and testing samples accordingly Cross-Validation technique. The regression model building is based on an iterative method that includes bijective multivariate normalizing transformation, multivariable outliers detection, confidence and prediction interval building. The obtained nonlinear regression model is compared to the existing models by the regression models quality criteria such as the determination coefficient, mean magnitude of relative error (MMRE) and the percentage of prediction of the relative error level 0.25 (PRED(0.25)). The comparison results confirm increasing the accuracy of the software size estimation using the obtained regression model on both training and testing samples. The prospects for further research may include increasing the samples of Java-software code metrics, usage of different software code metrics, extending the set of independent factors and using different normalizing transformations for early KLOC estimation.

## Keywords

Software size estimation, nonlinear regression model, normalizing transformation, Java, Cross-Validation

## 1. Introduction

The software size estimation plays a key planning role at the early stages of software project planning for the successful implementation of software development projects. The software size can be represented as functional points or number of code lines (KLOC - kilo lines of code). Both variants have their own advantages, drawbacks and restrictions. One of the crucial advantages of KLOC usage is in taking into account such important parameter as environmental factors which include programming languages or software category [1] and KLOC parameter is widely used in software development effort estimation parametric models such as COCOMO, COCOMO II, SLIM [2], COSYSMO [3], etc. Software development effort estimation is one of the significant indicators of budget, resources and duration planning of any project for software development business. Reliable estimates of software development effort provide valuable information for project

[1] *ICST-2024: Information Control Systems & Technologies, September , 23 – 25, 2024, Odesa, Ukraine*

oleksandr.oriekhov@nuos.edu.ua (O. Oriekhov); tetyana.farionova@nuos.edu.ua (T. Farionova); liubava.chernova@nuos.edu.ua (L. Chernova)

0000-0002-0001-0140 (O. Oriekhov); 0000-0003-3384-4712 (T. Farionova); 0000-0001-7846-9034 (L. Chernova)

planning, and it helps to take into account risks, to recognize planning gaps and to increase the efficiency of the development process [4].

Java programming language is one of the most widely used in software development [5]. Java software varies from utility and micro web-service applications to huge information or automotive or operation system development. Java is object oriented programming (OOP) language and it executes in a separated environment - Java Virtual Machine that makes it a universal instrument for any purpose [6]. Java as an OOP language has compatibility with UML software projection therefore they both have a shared set of software code metrics such as quantity of classes, total quantity of methods and fields or depending on access modifiers, inheritance, association or complexity metrics.

The CHAOS report 2015 [7] of The Standish group research confirms that large and grand projects have much higher failure rates in comparison with small, moderate or medium projects. Software development of most of the large projects is based on traditional project management methodologies like software development life cycle and waterfall, obviously it makes most of the parametric models more suitable for software development effort estimation than agile methodologies. Otherwise, company or team specific calibration of the parametric models that can work with uncertainty, such as COCOMO II and could provide reliable estimates for the projects that follow Agile methodology. It's expected the large projects have a bigger code base on final stages of development because of the complexity of integrating with highly distributed and different data and flows.

Since Java is one of the most demanded programming languages, the size estimation is an important task of the software project management life cycle and the share of failures or problems in software development is still high, lead us to confirmation the calculating requirement of reliable and accurate estimates of the size in code lines requires appropriate models for Java software to provide valuable information for project planning at the early stages.

The aim of the study is to build the three-factor nonlinear regression model for early KLOC estimation of Java-software on the basis of multivariate Johnson family normalizing transformations to increase the reliability and accuracy of the Java-software size estimation at the early stage of software project planning using UML class diagram metrics. The obtained math model should achieve higher quality criteria values in comparison with the existing math models of Java-software size estimation.

The object of the study is the process of size estimation for open-source Java-software.

The subject of the study is the nonlinear regression models to estimate the software size.

## 2. Review of the literature

Nowadays, Java-software size estimation is achieved with linear and nonlinear regression equations and models for general Java-software or categories such as web-service, informational-systems, etc on the metrics basis of the conceptual data model derived from a UML class diagram. The papers [8,9] propose linear regression equations on the basis of 3 independent factors. The Java KLOC estimation in [8] is based on total amount of classes (CLASS), coupling between objects (CBO) and total quantity of fields per class (TFQ) and the [9] is based on CLASS, CBO and average value of TFQ per class and proposes better variant of the equation in comparison with [8]. Typically, the software code metrics have non-Gaussian distribution. Therefore it has restrictions in the ability to apply linear models for KLOC estimation. One of the theoretical requirements indicates that the regression residuals $\varepsilon$ should have Gaussian distribution for the usage of linear regression models.

The existing nonlinear one-factor and multifactor regression models [10, 11, 12, 13] propose Java-software size estimation depending on different variants of software code metrics. A good benefit of the models is a proposed prediction intervals for possibility to make expected, pessimistic and optimistic KLOC estimates. The model [10] is based on dataset from researches [8] and [9], and it can not guarantee robustness and accuracy of KLOC estimation due to Java language evolution

and changing in modern software development practices that makes significant impact on software metrics previously and nowadays. The [10, 11, 12] regression models are based on restricted samples of Java applications that also have an impact on reliability and robustness of the nonlinear regression models. The latest research [13] proves the models [10, 11, 12] are restricted in estimation of code lines on different samples of code metrics from Java-software therefore the accurate result is not guaranteed. The [13] research offers one-factors and two-factor nonlinear regression models which were built using Cross-Validation technique. The regression models quality criteria such as the coefficient of determination R2, a mean magnitude of relative error MMRE and percentage of prediction for magnitude of relative error (MRE) level 0.25 PRED(0.25) reveal good quality level of the obtained two-factor model but the values of some quality criteria thresholds are not exceeded. Thus, study [14] confirms that with increasing sample sizes, the values of the quality criteria for the accuracy of predicting mathematical models for training and testing samples approach each other, which indicates the benefits of using large samples to build mathematical models in combination with Cross-Validation technique and adding more factors to nonlinear regression models allows to achieve higher accuracy accordingly quality criteria [13].

The review and analysis of the literature has shown that the linear equations and nonlinear regression models exist for estimating Java-software code size and confirms the necessity of increasing the robustness, reliability and accuracy of Java-software size estimation on early stages of project planning.

## 3. Problem statement

To achieve the aim of the research it is necessary to build the three-factor nonlinear regression model on the basis of multivariate normalizing transformation using the original training and testing non-Gaussian four-dimension samples of Java-software metrics to build and verify the regression model. The training and testing sample includes the thousand lines of code (KLOC) Y, the total number of classes $X_1$, total quantity of unique methods invocations in the code (RFC - response for class) $X_2$ and average value of public and protected methods per class (VMQ - visual methods quantity) $X_3$ from code metrics information of Java open-source software. The code metrics information of projects does not intersect between training and testing samples. Suppose there are bijective multivariate normalizing transformation of non-Gaussian random vector $P = \{Y, X_1, X_2, \dots, X_k\}^T$ into Gaussian random vector $T = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$ is given by

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

where $k$ is number of factors (regressors or independent variables) and the inverse transformation of (1) is given by

$$P = \psi^{-1}(T), \tag{2}$$

where $\psi$ is a vector of bijective normalizing transformation functions, $\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$.

On the bases of the transformations (1) and the required nonlinear regression model has the form $Y = F(Y, X_1, X_2, X_3, \varepsilon)$, where $\varepsilon$ is a Gaussian random variable that defines residuals, $F$ is three-factor nonlinear regression models and $Y$ is the dependent variable KLOC.

## 4. Materials, research methods and three-factor nonlinear regression model constructing

### 4.1. Three-factor nonlinear regression model constructing

The authors collected code metrics dataset of 571 open-source Java software hosted on the GitHub platform (https://github.com) using the CK tool (https://github.com/mauricioaniche/ck). The

following metrics were acquired: the actual total of Java-software size measured in number of thousand lines of code (KLOC), total quantity of classes (CLASS) $X_1$, total number of unique method calls in classes (RFC) $X_2$ and average value of visible methods per class of the software (aVMQ) $X_3$. CLASS, RFC, aVMQ, metrics can be obtained at an early stage of software planning from the conceptual model. The obtained dataset was randomly divided into training and testing samples with sizes of 286 and 285 rows of Java-software project data, respectively. The distribution of metrics relative to KLOC is shown in Figure 1 for training dataset and in Figure 2 for testing dataset.

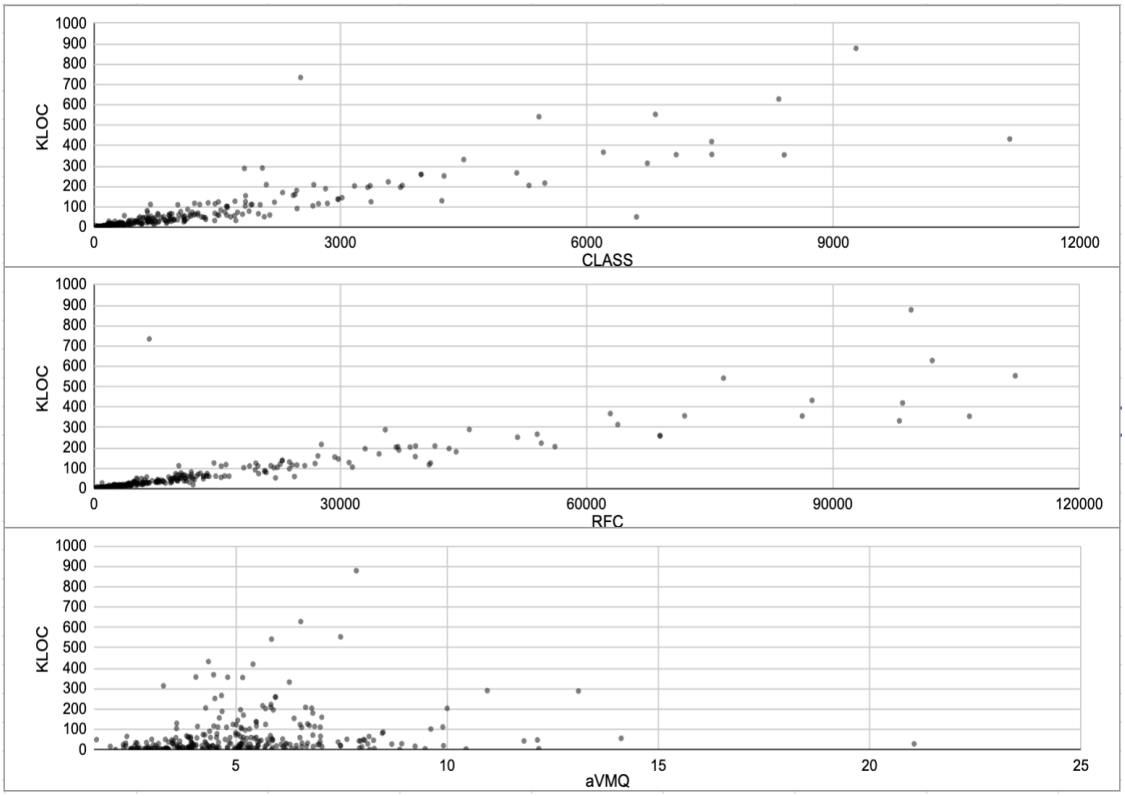


Figure 1: Training sample distribution relative to KLOC.

The variance inflation factors (VIFs) technique is used to check the predictors multicollinearity level to avoid a high correlation between regression factors and sensitivity of the model to random changes in the data.

For a multiple regression model with $k$ factors $X_i$ , $i = 1,2,\ldots,$ k, the VIFs are represented by the diagonal elements of the inverse correlation $k \times k$ matrix. If the value of the VIF coefficient exceeds 10 (the threshold value), a high level of multicollinearity exists between the independent variables [15].

For factors $X_1$, $X_2$, and $X_3$, the VIFs are equal to 6.0889, 6.0890 and 1.0 respectively, thus it indicates the absence of multicollinearity between the factors of the regression model.

The nonlinear regression models building technique is based on statistical analysis methods [16, 17] and it is based on detecting and discarding outliers in nonlinear regression analysis of non-Gaussian data and includes bijective normalizing transformations, Mahalanobis distance outliers detection, regression residuals distribution verification and prediction interval detection.

According to the technique it is recommended to detect and discard only one outlier in an iterative way once it is detected. In case if an outlier is detected, the technique starts from the first step using the modified sample without the detected outlier from the previous iteration. Otherwise the nonlinear regression model is successfully constructed. The first step of the technique is based on normalizing transformation.

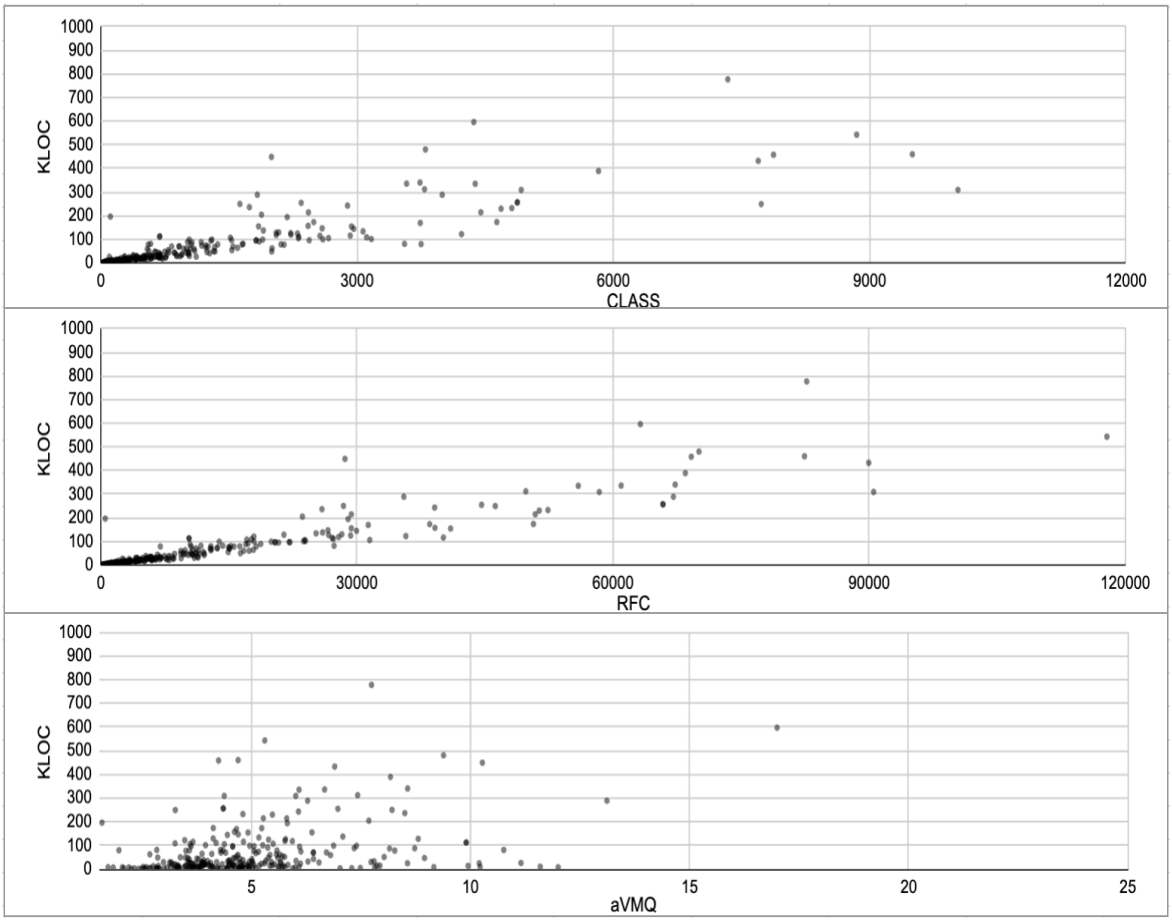

Figure 2: Testing sample distribution relative to KLOC.

In the first step for normalizing the multivariate non-Gaussian data of the training sample, we use the Johnson univariate and multivariate transformation for the SB family. It suggests a rich set of parameters for accurate calibration of the transformation. In our case the Johnson normalizing translation of SB family is given by [18]:

$$Z = \hat{\gamma} + \hat{\eta}\, ln\left(\frac{X - \hat{\varphi}}{\hat{\varphi} + \hat{\lambda} - X}\right), \tag{3}$$

where γ, η, φ та λ – parameters of Johnson SB family normalizing transformation, $\varphi < X < \varphi + \lambda$, $\eta > 0$, $\lambda > 0$; $X$ - non-Gaussian random variable which $X$ equals to $Y$, $X_1$, $X_2$, $X_3$, respectively.

And the inverse transformation is given by

$$X = \hat{\varphi} + \hat{\lambda}\left[1 + e^{-(Z-\hat{\gamma})/\hat{\eta}}\right]^{-1}, \tag{4}$$

where Z is a Gaussian random variable which $Z$ equals to $Z_Y$, $Z_{X1}$, $Z_{X2}$, $Z_{X3}$, respectively.

Parameters of Johnson univariate and multivariate transformations (3) are estimated by the finding solution of the equation [19] for computer programming task given by

$$\hat{\theta} = argmin\{A^2 + (\mathcal{E} - 3)^2 + \overline{Z^2} + (S_Z^2 - 1)^2\}, \tag{5}$$

where $\theta$ is a vector of estimators, $\theta = \{\gamma, \eta, \varphi, \lambda\}$; $A$ is skewness of a random variable, $\mathcal{E}$ is kurtosis of a random variable; $\bar{Z} = \frac{1}{N}\sum_{i=1}^{N} Z_i$; $S_Z^2 = \frac{1}{N}\sum_{i=1}^{N}(Z_i - \bar{Z})^2$; $Z_i$ - $i$-th value of random variable $Z$ of the sample size $N$. For the multidimensional data, we use measurement of multivariate skewness $(\beta 1, k)$ and kurtosis $(\beta 2, k)$ of the sample proposed in a Mardia [20] multivariate normality test.

Estimators for the parameters of multivariate transformation (3) vector $\hat{\theta}$ for metrics *Y* are: $\hat{\gamma}_Y = -4.061130$, $\hat{\eta}_Y = -0.501291$, $\hat{\varphi}_Y = -0.608187$, $\hat{\lambda}_Y = 82315.076609$; for $X_1$: $\hat{\gamma}_{X_1} = -45.701128$, $\hat{\eta}_{X_1} = 48.673051$, $\hat{\varphi}_{X_1} = -32.735283$, $\hat{\lambda}_{X_1} = 163.504695 \cdot 10^{12}$; for $X_2$: $\hat{\gamma}_{X_2} = -41.803051$, $\hat{\eta}_{X_2} = 47.042155$, $\hat{\varphi}_{X_2} = -110.321073$, $\hat{\lambda}_{X_2} = 627.451423 \cdot 10\text{^}14$; for $X_3$: $\hat{\gamma}_{X_3} = -36.520911$, $\hat{\eta}_{X_3} = 37.582881$, $\hat{\varphi}_{X_3} = -0.841250$, $\hat{\lambda}_{X_3} = 952.137743 \cdot 10^9$ for the latest iteration.

The second step includes outlier detection using squared Mahalanobis distance (MD) technique in the sample of normalized data. The squared Mahalanobis distance are elements on the main diagonal of the $d^2$ matrix of size *N x N*. The MD formula is given by

$$d^2 = (Z_i - \bar{\mathrm{Z}})^T S_N^{-1} (Z_i - \bar{\mathrm{Z}}) \tag{6}$$

and $S_N$ is given by

$$S_N = \frac{1}{N} \sum_{i=1}^{N} (Z_i - \bar{\mathrm{Z}})(Z_i - \bar{\mathrm{Z}})^T, \tag{7}$$

where *Z* is a normally distributed random variable $Z_i = \left(Z_{1_i}, Z_{2_i}, \dots, Z_{k_i}\right)^T$ , $S_N$ is a biased sample variance matrix and $Z$ is a means vector of independent variable of the sample, $\bar{\mathrm{Z}} = (\overline{\mathrm{Z}_1}, \overline{\mathrm{Z}_2}, \dots, \overline{\mathrm{Z}_\mathrm{k}})^T$.

The elements of the main diagonal $d_i^2, i = 1,2,\dots N$ matrix are detected as outliers if the values exceed the threshold value of the Chi-Square $\chi^2$ distribution quantile for the significant level - $\alpha$. For the training sample 17 data rows were detected as outliers because their $d_i^2$ values greater then threshold value 18.55 of the Chi-Square $\chi^2$ for the significant level $\alpha = 0.005$ and 20 degrees of freedom.

The linear regression model is built in the third step on the basis of the normalized multi dimensional sample is given by

$$Z_y = \hat{Z}_y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \tag{8}$$

where $\varepsilon$ is Gaussian random variable, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$; $\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3$ - estimators for parameters of the linear regression model (6).

The estimators are calculated by the least square method: $\hat{b}_0 = -23.784722$, $\hat{b}_1 = -0.005724$, $\hat{b}_2 = -0.005062$, $\hat{b}_3 = -0.008718$ .

In the fourth step the normality distribution of the linear regression residuals are tested with Pearson Chi-Square criteria for significant level $\alpha = 0.01$.

The actual values of the Chi-Square for the residuals are 116.5 and 19.24 for the 1th and 8th iterations respectively and the values are bigger than quantile 16.81 of the Chi-Square distribution for 6 degrees of freedom.

The 2 data rows with the highest values of residuals are removed from the training sample iteratively.

In the fifth step, the nonlinear regression models is built by applying inverse transformation to (2) to the linear regression models (8):

$$Y = \psi_Y^{-1}(\hat{Z}_y + \varepsilon) = \psi_Y^{-1}(\hat{b}_0 + \hat{b}_1 \psi_{X_1}(X_1) + \hat{b}_2 \psi_{X_2}(X_2) + \hat{b}_3 \psi_{X_3}(X_3) + \varepsilon), \tag{9}$$

where $\psi^{-1}$ is inverse Johnson SB transformation (4).

In the sixth step we constructed prediction interval $\hat{Y}_{PI}$ of the nonlinear regression model (8). The prediction interval is based on linear regression model (6) and it's given by

$$\hat{Y}_{PI} = \psi_Y^{-1}\left(\hat{Z}_Y \pm t_{\alpha/2,v} S_{Z_Y} \left\{1 + \frac{1}{N} + (Z_X^+)^T S_Z^{-1} (Z_X^+)\right\}^{1/2}\right), \tag{10}$$

where $t_{\alpha/2,v}$ is a quantile of T-Student distribution with $v = N - k - 1$ degrees of freedom and $\alpha/2$ significant level; $S_{Z_Y}^2 = \frac{1}{v}\sum_{i=1}^{N}\left(Z_{Y_i} - \hat{Z}_{Y_i}\right)^2$; $Z_X^+$ is a vector of central moments of independent variables of the sample which is given by $\left\{Z_{1_i} - \underline{Z}_1, Z_{2_i} - \underline{Z}_2, \ldots, Z_{k_i} - \underline{Z}_k\right\}$; $S_Z$ is *k x k* matrix

$$S_Z = \left[S_{Z_q} S_{Z_r}\right], \tag{11}$$

where $S_{Z_q} S_{Z_r} = \sum_{n-1}^{N}\left(Z_{q_i} - \bar{Z}_q\right)\left(Z_{r_i} - \bar{Z}_r\right)$, $q,r = 1,2, \ldots,k$.

For the training sample, 1 outlier is detected and discarded. For the obtained prediction interval, the values of normalized sample means $\bar{Z}_1$, $\bar{Z}_2$ and $\bar{Z}_3$ are $-1321.94070$, $-1447.061944$ and $-1008.068369$, respectively. The $t_{\alpha/2,v} = 2.594580$ for significant level $\alpha = 0.01$ and 264 degrees of freedom; $S_{Z_Y} = 0.089394$. The inverse matrix of (11) is

$$S_Z^{-1} = \begin{bmatrix} 2.39061 \cdot 10^{-5} & -2.28155 \cdot 10^{-5} & 2.1578 \cdot 10^{-5} \\ -2.28155 \cdot 10^{-5} & 2.29553 \cdot 10^{-5} & -2.28762 \cdot 10^{-5} \\ 2.1578 \cdot 10^{-5} & -2.28762 \cdot 10^{-5} & 5.60723 \cdot 10^{-5} \end{bmatrix},$$

The three-factor nonlinear regression model (9) is built in 19 iterations with discarding of 18 outliers from the training sample.

The Gaussian distribution of the four-dimension training data set ($N = 268$) is verified by a multivariate normality test proposed by Mardia [20] at the latest iteration. The test is based on measurement of multivariate skewness ($\beta 1, \mathrm{k}$) and kurtosis($\beta 2, \mathrm{k}$) of the sample.

Accordingly the test, distribution of four-dimension data $X_1$ (CLASS), $X_2$ (RFC), $X_3$ (aVMQ) and $Y$ (KLOC) is Gaussian, because multivariate skewness estimate $N\beta 1/6 = 39.85$ is not exceeded Chi-Square quantile value $40.00$ for $20$ degrees of freedom and significant level $\alpha$ =0.005 and the estimate of multivariate kurtosis $\beta 2 = 23.73$ does not exceed the value Gaussian distribution quantile which is equal to 26.18 for mean 24 and standard deviation 0.85.

## 4.2. Quality criteria of the obtained nonlinear regression model

The obtained nonlinear regression model (7) is tested with regression models quality criteria [21] such as the coefficient of determination $R^2$, a mean magnitude of relative error MMRE and percentage of prediction for magnitude of relative error (MRE) level 0.25 $PRED(0.25)$. The MMRE criterion is defined as

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i, \tag{12}$$

where *N* - sample size and $MRE_i$ is the value of the magnitude of relative error for the *i*-th datapoint of the sample.

$$MRE_i = \left|\frac{(Y_i - \widehat{Y}_i)}{Y_i}\right|. \tag{13}$$

The calculation of prediction percentage (PRED) for the magnitude of relative error level 0.25 is given by

$$PRED(0.25) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases} 1 \; if \; MRE_i \leq 0.25 \\ 0 \qquad otherwise \end{cases}. \tag{14}$$

The acceptable conditions for the quality criteria is $MMRE \leq 0.25$ and $PRED(0.25) \geq 0.75$ for the measurement of the regression models accuracy of prediction results. The coefficient of determination ($R^2$) value is acceptable if it is more or equals to 0.75 [21].

The obtained three-factor nonlinear regression model is tested with the quality criteria to assess the predictive reliability and accuracy on the training and the testing samples and compared with

existing one-factor and two-factor nonlinear regression models from the paper [13]. The estimates of $R^2$, $MMRE$ and $PRED(0.25)$ are devoted in Table 1.

Table 1
Comparison of nonlinear regression models by quality criteria

| Nonlinear regression model | Training sample | | | Testing sample | | |
|---|---|---|---|---|---|---|
| | $R^2$ | MMRE | PRED (0.25) | $R^2$ | MMRE | PRED (0.25) |
| One-factor on the basis of decimal logarithm transformation [13] | 0.7266 | 0.3522 | 0.4720 | 0.7170 | 0.3350 | 0.4632 |
| One-factor on the basis of Box-Cox transformation [13] | 0.7291 | 0.3514 | 0.4790 | 0.7159 | 0.3354 | 0.4780 |
| One-factor on the basis of Johnson SB transformation [13] | 0.6825 | 0.3593 | 0.4790 | 0.6616 | 0.3417 | 0.4702 |
| Two-factor on basis of Box-Cox transformation [13] | 0.8002 | 0.2332 | 0.6853 | 0.8981 | 0.1964 | 0.7158 |
| Three-factor on basis of Johnson SB transformation (9) | 0.9073 | 0.1645 | 0.7692 | 0.9016 | 0.1617 | 0.8175 |

To compare prediction quality and accuracy of the obtained regression model (9) we build prediction intervals using the training sample for the one-factor nonlinear regression model on the basis of Johnson SB transformation [13]. Similar to the obtained three-factor nonlinear regression model (9), the one-factor nonlinear model is built using the same techniques [16, 17] that is based on statistical analysis of multidimensional data and multivariate reciprocal transformations. The estimators for the Johnson SB transformation parameters of the one-factor model are $\hat{\gamma}_Y = 4311.992036$, $\hat{\gamma}_X = 3555.710081$, $\hat{\eta}_Y = 30479.194612$, $\hat{\eta}_X = 29712.602140$, $\hat{\varphi}_Y = -0.345327$, $\hat{\varphi}_X = -3.150988$, $\hat{\lambda}_Y = 2.471319 \cdot 10^{14}$, $\hat{\lambda}_X = 22965.385684$ and the estimators of the linear regression are $\hat{b}_0 = -799519.0395$, $\hat{b}_1 = 0.999920$. For the obtained prediction interval, the values of normalized sample means $\bar{Z}_X = -104.702321 \cdot 10^3$. The $t_{\alpha/2,\nu} = 2.594161$ for significant level $\alpha = 0.01$ and 270 degrees of freedom; $S_{Z_Y} = 10577.32936$. The inverse matrix of (11) consists of one element $S_Z^{-1} = 2.46245 \cdot 10^{-12}$. We have randomly chosen 30 of 286 data rows from the training sample to compare actual ($Y$) and estimated ($\hat{Y}$) KLOC values by the regression models and their lower bound (LB) and upper bound (UB) prediction intervals (Table 2).

Table 2
The prediction results and prediction intervals of nonlinear regressions

| | | The one-factor nonlinear regression model on the basis on Johnson SB | | | The three-factor nonlinear regression model on the basis on Johnson SB | | |
|---|---|---|---|---|---|---|---|
| # | $Y$ | $\hat{Y}$ | LB | UB | $\hat{Y}$ | LB | UB |
| 1 | 26.463 | 20.262 | 8.016 | 50.444 | 24.675 | 15.284 | 39.613 |
| 2 | 47.052 | 31.592 | 12.614 | 78.360 | 59.386 | 37.014 | 95.045 |
| 3 | 25.108 | 37.233 | 14.902 | 92.267 | 25.381 | 15.694 | 40.821 |
| 4 | 32.877 | 35.348 | 14.138 | 87.621 | 33.067 | 20.574 | 52.923 |
| 5 | 19.818 | 42.180 | 16.908 | 104.470 | 19.756 | 12.139 | 31.922 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 40.095 | 44.568 | 17.876 | 110.361 | 42.683 | 26.618 | 68.221 |
| 7 | 208.967 | 103.964 | 41.908 | 257.163 | 155.162 | 97.054 | 247.739 |
| 8 | 33.909 | 20.691 | 8.190 | 51.499 | 32.801 | 20.357 | 52.625 |
| 9 | 137.599 | 152.221 | 61.383 | 376.737 | 136.863 | 85.637 | 218.433 |
| 10 | 110.167 | 94.084 | 37.915 | 232.712 | 85.814 | 53.683 | 136.928 |
| 11 | 160.264 | 122.820 | 303.858 | 49.522 | 154.151 | 96.579 | 245.725 |
| 12 | 204.818 | 198.788 | 80.143 | 492.324 | 220.936 | 138.397 | 352.277 |
| 13 | 25.298 | 42.667 | 17.105 | 105.671 | 26.406 | 16.352 | 42.416 |
| 14 | 6.576 | 11.900 | 4.620 | 29.852 | 8.207 | 4.929 | 13.426 |
| 15 | 78.968 | 92.879 | 37.428 | 229.731 | 92.129 | 57.698 | 146.857 |
| 16 | 59.438 | 78.795 | 31.734 | 194.899 | 62.564 | 39.097 | 99.884 |
| 17 | 102.327 | 79.052 | 31.837 | 195.532 | 133.556 | 83.444 | 213.469 |
| 18 | 43.158 | 54.689 | 21.977 | 135.340 | 33.918 | 21.045 | 54.440 |
| 19 | 13.745 | 20.453 | 8.094 | 50.912 | 19.967 | 12.334 | 32.100 |
| 20 | 205.514 | 301.683 | 121.507 | 748.272 | 250.745 | 157.173 | 399.541 |
| 21 | 17.546 | 14.607 | 5.720 | 36.516 | 16.844 | 10.362 | 27.154 |
| 22 | 47.105 | 35.686 | 14.275 | 88.454 | 42.570 | 26.551 | 68.030 |
| 23 | 258.993 | 213.091 | 85.900 | 527.859 | 283.491 | 177.688 | 451.734 |
| 24 | 76.739 | 50.644 | 20.338 | 125.354 | 51.318 | 32.057 | 81.925 |
| 25 | 64.504 | 59.304 | 23.846 | 146.737 | 48.487 | 30.259 | 77.468 |
| 26 | 222.656 | 188.312 | 75.925 | 466.304 | 236.744 | 148.473 | 377.040 |
| 27 | 29.704 | 31.448 | 12.556 | 78.005 | 29.517 | 18.339 | 47.285 |
| 28 | 15.737 | 14.132 | 5.527 | 35.347 | 15.719 | 9.658 | 25.358 |
| 29 | 137.599 | 152.221 | 61.383 | 376.737 | 136.863 | 85.637 | 218.433 |
| 30 | 61.270 | 37.426 | 14.981 | 92.744 | 51.895 | 32.348 | 83.025 |
| N | … | … | … | … | … | … | … |

Prediction intervals of the one-factor and three-factor nonlinear regression models are compared by given formula

$$DIFF = \left(1 - \frac{\sum_{i=1}^{N}[UB_{3x} - LB_{3x}]}{\sum_{i=1}^{N}[UB_{1x} - LB_{1x}]}\right) \cdot 100\%, \quad (15)$$

where $UB_{3x}$ , $LB_{3x}$ - $i$-th prediction intervals of the three-factor nonlinear regression model (9) and $UB_{1x}$ , $LB_{1x}$ - $i$-th prediction intervals of the one-factor nonlinear regression model. The $DIFF$ value indicates the three-factor nonlinear regression interval is $52.78\%$ smaller than the one-factor nonlinear regression interval on the basis of the training sample.

To make visualization of the comparison of the one-factor nonlinear model and the three-factor nonlinear model accuracy we sorted all 571 data rows in ascending order by actual KLOC values of the multidimensional dataset with KLOC estimates and prediction intervals of the target models and first 400 records were chosen by x-axis and 150 KLOC max value was chosen by y-axis due to scaling. The KLOC estimates and prediction intervals were smoothed by moving averages technique on a basis of 5 values regarding the centric value of them. The obtained charts are demonstrated on Figure 3.

The graphic representation displays that the estimates of the three-factor nonlinear model (9) are closer to the actual KLOC values than the estimates of the one-factor nonlinear model.

The prediction interval width is smaller in comparison with the interval of the one-factor nonlinear model on the basis of the same normalizing transformation.

The values of the quality criteria indicate good model accuracy, the high values of for the testing sample confirms the model is not overfitted that confirms robustness and reliability. $R^2$, $MMRE$ and $PRED(0.25)$ values of training and testing samples are close to each other thus confirm the samples have a high level of representativity of the population of the open-source Java-software [13].

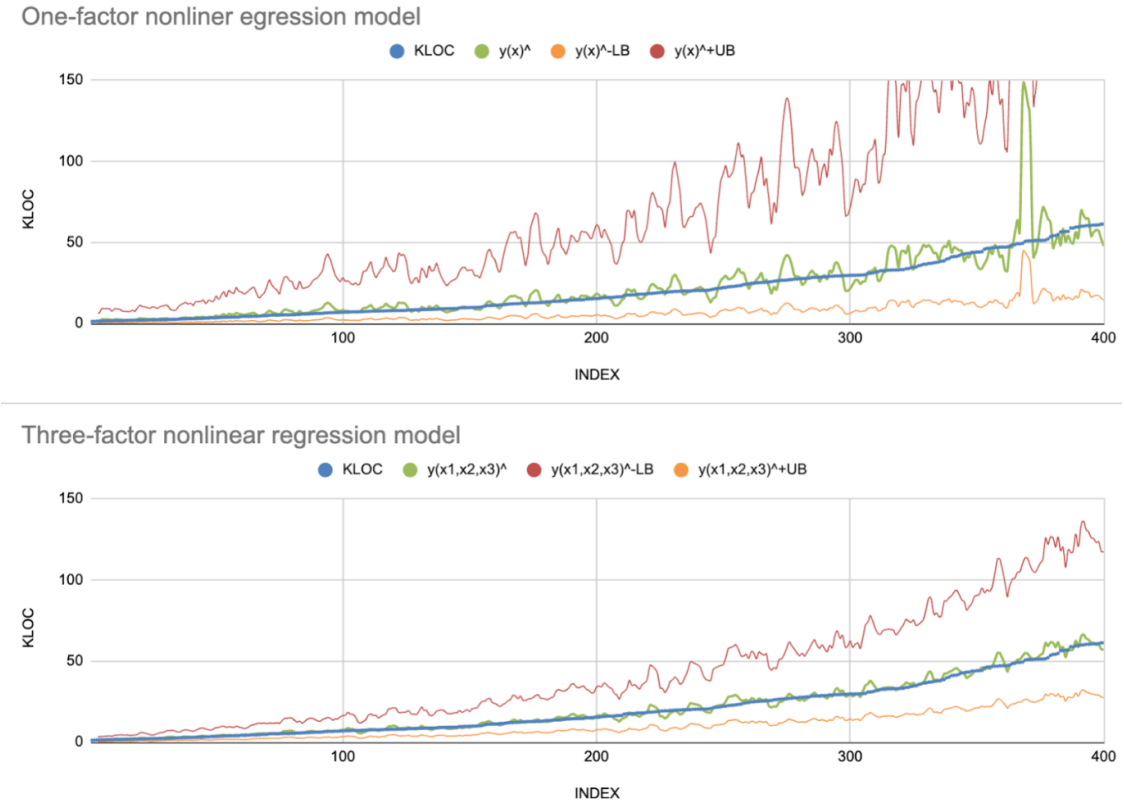


Figure 3: Testing sample distribution relative to KLOC.

Table 2 and (15) prove the three-factor nonlinear regression model (9) has higher level of prediction accuracy and smaller prediction intervals in comparison with the existing model [13] with the same normalizing transformation (3).

## 5. Discussion

We obtained the three-factor nonlinear regression model for early estimation of KLOC in Java-software using appropriate techniques for constructing non-linear regression models on the basis of multivariate normalizing Johnson SB family transformation. The technique is chosen due to the non-Gaussian distribution of four-dimensional data of the code metrics according to Mardia criteria and the regression residuals according to the Chi-Square test.

Table 1 and models comparison in [13] confirms that the results of this research demonstrate the effectiveness and accuracy of the proposed three-factor nonlinear regression model for early KLOC estimation in comparison with existing linear and non-linear regression models by quality criteria $R^2$, $MMRE$ and $PRED(0.25)$. The criteria for the model are exceeded expectation $R^2 \geq 0.75$, $MMRE \leq 0.25$ and $PRED(0.25) \geq 0.75$ for both training and testing samples. The values are $R^2 = 0.9073$, $MMRE = 0.1645$, $PRED(0.25)0.7692$ for training sample and $R^2 = 0.9016$,

$MMRE = 0.1617$, $PRED(0.25) = 0.8175$ for the testing sample. Cross-Validation technique increases guarantees of robustness of the model. Moreover, the width of prediction interval (10) of the model (9) is 52.78% smaller (Table 2 and Figure 3) than the interval of one-factor nonlinear regression model which allows to increase accuracy of KLOC estimation taking in account positive and negative cases.

The advantages of the proposed model (9) includes the possibility of early KLOC estimation of Java-software using three code metrics such as total quantity of classes (CLASS), total number of unique method calls in classes (RFC) and average value of visible methods per class (aVMQ) that could be obtained on the early stages of the project planning from UML class diagrams. The disadvantages of the proposed model (9) include the following restrictions for the predictors: the interval for CLASS is from 25 to 11147, the interval for RFC is from 45 to 117847 and the interval for aVMQ is from 1.5893 to 85.92. Despites on the usage of the big sample of code metrics, the proposed model is built only on a basis of open-source Java-software projects from GitHub.

However, the model achieved high quality level according to quality criteria [21] and Table 2 confirms that the KLOC estimates by the three factor model (9) is closer to the actual KLOC values than the one-factor model, there are more directions for further research. The model (9) could be improved by adding more metrics to reveal characteristics of inheritance relation between classes and quantity of visible fields of the classes. But the adding of more predictors could cause multicollinearity problems that lead us to converting or avoiding usage of the predictors.

The obtained results confirm that the constructed three-factor nonlinear model (9) improves confidence of KLOC estimation of the Java-software.

## 6. Conclusion

The obtained three-factor nonlinear regression model and prediction intervals solves important problem of Java-software size estimation at the early stages of software development project planning using UML class diagram metrics: total quantity of classes (CLASS), total quantity of responses for class (RFC) and average value of public and protected methods per class (aVMQ).

The scientific novelty of the obtained results is that the three-factor nonlinear regression model is improved in comparison with existing models and equation for KLOC estimation of Java-software; Multivariate Johnson SB family normalizing transformation is firstly used for model construction using CLASS, RFC and aVMQ multidimensional dataset; the three-factor nonlinear regression model is firstly built using large sample size ($N = 286$) and the robustness and reliability verified by using the testing sample with the similar size. The model, in comparison with other nonlinear regression models, has higher value of coefficient of determination $R^2$, a lower value of the mean relative error $MMRE$, higher value of the percentage of prediction of the relative error level $PRED(0.25)$ for both training and testing samples and the prediction interval is smaller in comparison with one-factor nonlinear regression models. The obtainer quality criteria values confirm the training and the testing samples are close to each other thus verifies the samples have a high level of representativity of the population of the open-source Java-software

The practical significance of the obtained results allows us to recommend the built model for use in practice. The proposed model is implemented as a software product that can be used by project managers for Java software development effort estimation at early stages of project planning to reduce risks and cost.

Prospects for further research may include extending a set of independent factors, the usage of other multivariate normalizing transformations and extending wider data sets to build a nonlinear regression model.

## 7. Acknowledgements

The authors would like to express their sincere gratitude to the Doctor of Technical Sciences, Professor Sergiy B. Prykhodko, NUOS, Mykolaiv, Ukraine for his support in applying the

mathematical apparatus for the research and to PhD. Maurício Aniche, Assistant Professor in Software Engineering at TU Delft, for providing valuable Java code static analysis tools CK.

## References

[1] A. Trendowicz, R. Jeffery, Software Project Effort Estimation, Foundations and Best Practice Guidelines for Success, in: Springer International Publishing, 2014. doi: https://doi.org/10.1007/978-3-319-03629-8

[2] S. W. Munialo, A Review of Agile Software Effort Estimation Methods, International Journal of Computer Applications Technology and Research. Association of Technology and Science 5 (2016) 612–618. doi:10.7753/IJCATR0509.1009

[3] R. Valerdi, B. W. Boehm, D. J. Reifer, COSYSMO: A constructive systems engineering cost model coming of age, in: Proc. INCOSE Int. Symp., volume. 13 of No. 1, Wiley, Hoboken, NJ, USA, 2003, pp. 70–82.

[4] S. McConnel, Software Estimation: Demystifying the Black Art, Microsoft Press, Redmond, Washington, USA, 2006.

[5] TIOBE, TIOBE Index, 2024. URL: https://www.tiobe.com/tiobe-index/.

[6] Oracle, Java, 2024. URL: https://www.oracle.com/my/java/.

[7] The Standish Group, Chaos report 2015, 2015. URL: https://standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.

[8] H. B. K. Tan, Y. Zhao, H. Zhang, Estimating LOC for information systems from their conceptual data models, in: Proceedings - International Conference on Software Engineering, 2006, pp. 321-330. doi:10.1145/1134285.1134331.

[9] H. B. K. Tan, Y. Zhao, H., H. Zhang, Conceptual Data Model-Based Software Size Estimation for Information Systems, ACM Transactions of Software Engineering and Methodology 19 (2009). doi:10.1145/1571629.1571630.

[10] N. V. Prykhodko, S.B. Prykhodko, A nonlinear regression model for estimation of the size of Java enterprise information systems software, Modeling and Information Technologies 85 (2018) 81-88.

[11] L. M. Makarova, N.V. Prykhodko, O. O. Kudin, Constructing the non-linear regression model for size estimation of web-applications implemented in Java, Herald (Kherson National Technical University), 69 (2019) 145-153.

[12] S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, Four-factor non-linear regression model to estimate the size of open source Java-based applications, Scientific Notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences, 70 (2020) 157-162. doi:10.32838/2663-5941/2020.2-1/25

[13] O. S. Oriekhov, T. A. Farionova, Mathematical models for the size estimating of JAVA applications, Visnyk of Kherson National Technical University 89 2 (2024) 196-203. doi: 10.35546/kntu2078-4481.2024.2.28.

[14] J. Jia, W. Qiu, Research on an Ensemble Classification Algorithm Based on Differential Privacy, in: IEEE Access 99 (2020). DOI:10.1109/ACCESS.2020.2995058

[15] I. Olkin, A. R. Sampson, Multivariate Analysis: Overview, in N. J. Smelser, P. B. Baltes, International encyclopedia of social & behavioral sciences (eds.) 1st edn., Elsevier, Pergamon, 2001, pp. 10240–10247.

[16] S. Prykhodko, N. Prykhodko, Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations, in: S. Shkarlet, A. Morozov, A. Palagin, Mathematical Modeling and Simulation of Systems (MODS'2020). Advances in Intelligent Systems and Computing, volume 1265 of MODS, 2021, pp. 166-174. doi:10.1007/978-3-030-58124-4_16

[17] S. Prykhodko, N. Prykhodko, L. Makarova and A. Pukhalevych, Outlier Detection in Non-Linear Regression Analysis Based on the Normalizing Transformations, in: 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and

Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2020, pp. 407-410. doi:10.1109/TCSET49122.2020.235464.

[18] P. M. Stanfield, J. R. Wilson, G. A. Mirka, N. F. Glasscock, J. P. Psihogios, J. R. Davis, Multivariate input modeling with Johnson distributions, in: Proceedings of the 28th Winter simulation conference WSC'96, Coronado, CA, USA, December 8-11, 1996, ed. S. Andradóttir, K.J. Healy, D.H. Withers and B.L. Nelson, IEEE Computer Society Washington, DC, USA, pp. 1457-1464.

[19] S. B. Prykhodko, Simulation of gaussian random variables using Johnson SU transform, Informatics and mathematical methods in simulation 1 5 (2015) 92-97.

[20] K. V. Mardia, Measures of multivariate skewness and kurtosis with applications, Biometrika 57 (1970) 519–530. doi:10.1093/biomet/57.3.519.

[21] D. Port, M. Korte, Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research, in: Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, New York, USA, 2008, pp. 51–60. doi:10.1145/1414004.1414015