UDC 004.412:519.25
DOI: 10.15587/1729-4061.2026.363077

DEVELOPMENT OF A MATHEMATICAL MODEL FOR SIZE ESTIMATION OF JAVA SPRING WEB APPLICATIONS USING REGRESSION ANALYSIS

Lidiia Makarova
Corresponding author
Candidate of Technical Sciences, Associate Professor*
E-mail: lidiia.makarova@nuos.edu.ua
ORCID: https://orcid.org/0000-0003-2903-3001
Liudmyla Latanska
Candidate of Physical and Mathematical Sciences, Associate Professor*
ORCID: https://orcid.org/0000-0001-6473-7624
Andrii Pukhalevych
Candidate of Technical Sciences, Associate Professor*
ORCID: https://orcid.org/0000-0002-8827-3251
Vladimir Kairov
Candidate of Technical Sciences, Associate Professor*
ORCID: https://orcid.org/0000-0003-0502-7942
Maksym Dzhurynksyi*
ORCID: https://orcid.org/0009-0002-0685-9038
*Department of Software of Automated Systems
Admiral Makarov National University of Shipbuilding
Heroiv Ukrainy ave., 9, Mykolaiv, Ukraine, 54007

This study investigates the process to estimate the size of web applications developed in Java (country of origin – USA) using the Spring framework (country of origin – USA).

The task addressed is to improve reliability in estimating the size of the corresponding web applications. Estimating the size of web applications developed in Java using the Spring framework is an important task in software engineering. This makes it possible to assess the required time frame for implementing the functionality and to plan a project budget.

As a result of the work, a mathematical model was built for estimating the size of web applications in Java and Spring based on the normalizing transformation of the decimal logarithm; a corresponding program was developed to automate calculations. Quality parameters of the constructed model are as follows: R2 = 0.9173, MMRE = 0.1511, PRED(0.25) = 0.7931.

That has made it possible to improve the reliability of estimating the size of such web applications, namely, to reduce the value of average relative error and increase the level of prediction, as well as to narrow the widths of the confidence and prediction intervals.

A data set consisting of 36 projects was collected; its preprocessing was performed, which included checking for multivariate normality of the distribution and normalization by logarithmization. An appropriate nonlinear regression model was constructed. To improve the reliability of the model, an algorithm was applied that includes iterative removal of outliers using the square of the Mahalanobis distance and Fisher's criterion. A program was developed based on the constructed nonlinear regression model; the results were analyzed.

Analysis of the quality of the constructed model reveals its adequacy and applicability for solving tasks of estimating the size of the corresponding software

Keywords: size estimation, normalization, nonlinear regression, outlier, Java, Spring, software

Received 16.03.2026
Received in revised form 27.04.2026
Accepted date 28.05.2026
Published date 30.06.2026

How to Cite: *Makarova, L., Latanska, L., Pukhalevych, A., Kairov, V., Dzhurynskyi, M. (2026). Development of a mathematical model for size estimation of Java Spring web applications using regression analysis. Eastern-European Journal of Enterprise Technologies, 3 (2 (141)), 67–74. https://doi.org/10.15587/1729-4061.2026.363077*

1. Introduction

The Java programming language is a common general-purpose language. It is used together with the Spring framework to develop a variety of software: web services, games, desktop applications. Java is one of the most popular programming languages in the world due to its simple syntax, flexibility, security, portability, and scalability [1].

At the early stages of development, the customer and the developer must agree on the scope of work, deadlines, as well as budget, but doing so without reliable quantitative estimates of the code is difficult. That is why estimating the size of Java web applications using the Spring framework is an important task in software engineering. Solving this task makes it possible to effectively plan the costs of developing and further supporting such web applications, plan the project budget, and determine the necessary timeframe for implementing the functionality.

Existing models for estimating the size of software do not take into account the features of the Spring framework. Their use may lead to low reliability of the estimates obtained. Therefore, conducting scientific research on estimating the size of Java web applications using the Spring framework is a relevant and necessary task in modern software engineering.

The practical result of such research is the opportunity for developers and project managers to obtain a quantitative estimate of the size of a web application at an early stage. This allows for realistic budget planning and development deadlines, as well as reasonable team composition.

In addition, under current development conditions, there is a need to automate the processing of code metrics because manual analysis of large data sets and building dependences are not only time-consuming but also prone to errors due to the human factor.

Thus, given the prevalence of the Spring framework for developing web applications in Java and the lack of spe-

cialized models for estimating the size of such applications, scientific research into this area is becoming increasingly relevant.

2. Literature review and problem statement

The authors of [2] derived linear regression equations for estimating the size of information systems software. The dependent variable was the number of lines of code in thousands, and the independent variables were the following characteristics of the conceptual data model: the total number of classes, the total number of relationships, and the average number of attributes per class. However, the number of lines of code may also depend on other factors and their number, which was considered, in particular, in [3]. This work considers the construction and validation of models for estimating the size of software projects based on four metrics of the analysis class diagram using stepwise multiple linear regression. The authors used the following metrics, adjusted for the analysis-to-design adjustment factor (ADAF): the number of classes, the number of attributes, the number of methods, and the number of relationships. In addition, non-functional requirements may also affect the size of the software. In [4], the influence of non-functional requirements related to security on the early estimation of the size of software systems was investigated by constructing a linear regression model using an industrial dataset.

A common drawback of papers [2–4] is the use of linear regression. However, for the correct application of linear regression analysis, the data must meet a number of statistical requirements:

– model variables must have a distribution close to normal;

– dependent and independent (independent) variables must have a close to linear relationship;

– regression residuals must be normally distributed;

– absence of multicollinearity – independence of predictor variables from each other in the case of using multivariate regression [5].

Those requirements are valid only in some cases. Therefore, the option to overcome these difficulties is usually the construction of nonlinear regression models.

A number of papers consider the construction of nonlinear regression models, including the studies reported in [6–10]. They use statistical iterative methods to detect and remove outliers in non-Gaussian data based on mutual normalizing transformations.

In [6], a nonlinear regression model was built for early estimation of the size of Data Science and Machine Learning applications using the decimal logarithm as a normalizing transformation, as well as the following code metrics: number of lines of code, number of classes, total number of visible methods, and average number of attributes per class. However, the architecture and structure of such applications differ significantly from web applications developed in Java using the Spring framework, which limits the possibility of directly using the proposed model to estimate their size.

In [7], the authors built a three-factor nonlinear regression model to obtain an early estimate of the size of open-source Kotlin-based applications. A four-variate Box-Cox transformation was used for normalization. The initial non-Gaussian data set included: software size in thousands of lines of code, total number of classes, weighted number of methods per class metrics, and depth of the inheritance tree at the application level. Despite the fact that the Kotlin language is close to Java and is used in a similar development environment, the features of the architecture of web applications based on the Spring framework, as well as the specifics of the technology stack used, necessitate the construction of a specialized model for early estimation of their size.

Work [8] considers building a nonlinear regression model with three predictors (total number of classes, average number of methods per class and average inheritance tree depth per class) to estimate the number of lines of code of web applications created using the CakePHP framework. The four-variate Box-Cox transformation was used to normalize the data. Despite belonging to web-oriented systems, the use of a different programming language and differences in the structure of the program code and the principles of building applications do not make it possible to ensure sufficient accuracy of the assessment for web applications implemented in Java using the Spring framework.

In work [9], a nonlinear regression model is built to estimate the size of open applications in PHP using the multivariate Johnson transformation for the S_B family. The four-variate non-Gaussian dataset contained the actual size of the program in thousands of lines of code, as well as the total number of classes, the average number of methods per class, and the sum of the average afferent and average efferent couplings per class, which can be obtained from the class diagram. However, the data used in the model did not take into account the specific type of software.

Existing models for web applications implemented in Java were considered, which are aimed at estimating the size of software and are based on the number of classes as an independent variable. In [10], several such models are presented, one of which is a model using the univariate normalizing Johnson transformation, the other model is based on the decimal logarithmic transformation. However, the work did not take into account the framework used in the development of software projects.

Our review of the literature [6–10] shows that the use of nonlinear regression models and normalizing transformations makes it possible to improve the reliability of estimating the size of software projects for non-Gaussian data. At the same time, the construction of highly specialized models taking into account the technological development stack contributes to improving the forecasting results.

However, issues related to the construction of specialized models for web applications developed in Java using the Spring framework remain unresolved. All this gives grounds to argue that it is advisable to conduct a study aimed at building a nonlinear regression model for early estimation of the size of web applications developed in Java using the Spring framework.

3. The aim and objectives of the study

The purpose of our study is to improve the reliability of estimating the size of web applications developed in Java using the Spring framework by building a mathematical model based on the normalizing transformation of the decimal logarithm and developing a corresponding program for automating calculations. This will allow for a more accurate assessment of the labor intensity and costs of developing the corresponding web applications at the early stages of project planning.

To achieve this goal, the following tasks must be solved:

– to develop an algorithm for solving the task of building a model with iterative removal of outliers;

– to collect, describe, and preprocess data from web application metrics to build a model;

– to build a nonlinear regression model for estimating the size of web applications;

– to develop a program for automating calculations based on the constructed mathematical model.

4. The study materials and methods

4. 1. The object and hypothesis of the study

The object of our study is the process of estimating the size of web applications developed in Java using the Spring framework.

The principal hypothesis of the study assumes that the reliability of estimating the size of web applications in Java using Spring could be increased by building a nonlinear regression model based on the normalizing transformation of the decimal logarithm.

The following assumptions are adopted in the study:

– the size of the software in lines of code and the number of classes are interdependent quantities;

– the studied software projects are homogeneous in terms of technological stack;

– the studied software projects belong to the same class of systems.

The following simplifications are accepted in the study:

– one independent variable;

– a univariate normalizing transformation is used.

To calculate the value of the metrics of web applications developed in Java, the CK tool was used [11].

When building the model, a method of building nonlinear regression models based on normalizing transformations [9] was used.

The program for automating calculations was developed in the Java programming language [12], in the IntelliJ IDEA development environment (country of origin – Czech Republic) [13] and using standard Java libraries for building a graphical interface and mathematical calculations.

The calculations were performed on a personal computer with an Intel Core i5-1135G7 processor (2.4 GHz), 16 GB of RAM, and the Windows 11 operating system. The execution time of the full model building cycle for a sample of 36 projects did not exceed 2 seconds.

4. 2. Data preprocessing

Data preprocessing is the foundation for further modeling and often takes up a large portion of the project time. Data preparation is the most important stage, the quality of which determines the possibility of obtaining high-quality results of the entire Data Mining process. According to some estimates, up to 80% of the total time allocated to the project can be spent on the data preparation stage [14].

The preprocessing process includes the following main tasks:

– checking the multivariate distribution for normality;

– data normalization: bringing the data distribution to a normal form;

– finding and removing outliers in the data.

The essence of these procedures was considered in more detail.

To check the multivariate distribution for normality, Mardia criteria are used – multivariate skewness and kurtosis [15]:

$$\beta_{1,k} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\left(\mathbf{X}_i - \overline{\mathbf{X}}\right)^T \mathbf{S}_N^{-1}\left(\mathbf{X}_j - \overline{\mathbf{X}}\right)\right]^3, \quad (1)$$

$$\beta_{2,k} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\mathbf{X}_i - \overline{\mathbf{X}}\right)^T \mathbf{S}_N^{-1}\left(\mathbf{X}_i - \overline{\mathbf{X}}\right)\right]^2, \quad (2)$$

where $\mathbf{X}$ is a k-dimensional vector of variables, $\mathbf{X} = (X_1, X_2, \ldots, X_k)$; $\mathbf{S}_N$ is a sample covariance matrix, which is calculated from the following formula [16]

$$\mathbf{S}_N = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{X}_i - \overline{\mathbf{X}}\right)\left(\mathbf{X}_i - \overline{\mathbf{X}}\right)^T, \quad (3)$$

where $\overline{\mathbf{X}}$ is the vector of mean values.

The test statistic for $\beta_{1,k}$ is determined from the formula $T = (N / 6)\beta_{1,k}$ and is compared with the critical value of the χ^2 distribution: $T \le \chi^2_\alpha$, where χ^2_α is the upper α-quantile of the χ^2 distribution with $k(k+1)(k+2) / 6$ degrees of freedom, α is the significance level.

For $\beta_{2,k}$, the $1 - \alpha$ quantile of the normal distribution with the mathematical expectation $k(k + 2)$ and variance $8k(k + 2) / N$ is used as the test statistic $z_{2,k}$.

4. 3. Data normalization

Many statistical algorithms, in particular linear regression, impose strict requirements on the nature of the distribution of the input data. Model variables must have a distribution close to normal. If the input data have a bias, nonlinear transformation methods are used to correct them and approximate them to a normal (Gaussian) distribution. The most effective methods for this are data normalization and normalizing transformations: logarithmization (in particular, the decimal logarithm), Box-Cox transformation, and Johnson transformation.

Normalization is necessary to bring the input data distribution to a form close to normal. If there is a normalizing transformation of a non-Gaussian random vector $\mathbf{P} = \{Y, X_1, X_2, \ldots, X_k\}^T$ into a Gaussian random vector $\mathbf{T} = \{Z_Y, Z_1, Z_2, \ldots, Z_k\}^T$, which is given by equation $\mathbf{T} = \psi(\mathbf{P})$, then there is an inverse transformation, which has the form $\mathbf{P} = \psi^{-1}(\mathbf{T})$, where: ψ is a vector, $\psi = \{\psi_Y, \psi_1, \psi_2, \ldots, \psi_k\}^T$ [9, 17].

The simplest example of a normalizing transformation from the point of view of implementation is a logarithmic transformation based on the decimal logarithm, which has the form $Z = \lg(x)$. Then the inverse transformation takes the form $x = 10^Z$.

As for anomalous values (noise and outliers), they need to be detected and their impact on the analysis results should be assessed because results based on dirty data cannot be considered reliable and useful. For multivariate data, this can be done using the square of the Mahalanobis distance, using the procedure given in [18]

$$d_i^2 = \left(\mathbf{Z}_i - \overline{\mathbf{Z}}\right)^T \mathbf{S}_Z^{-1}\left(\mathbf{Z}_i - \overline{\mathbf{Z}}\right), \quad (4)$$

where $\mathbf{Z}_i$ is the i-th data component, $\overline{\mathbf{Z}}$ is the vector of sample means, $\mathbf{S}_Z$ is the covariance matrix calculated from formula (3).

Also, for the square of the Mahalanobis distance obtained from (4), we can calculate the test statistic

$$T_{S_i} = \frac{N(N-k)}{k(N^2-1)} d_i^2, \quad (5)$$

which can be compared with the quantile of the Fisher distribution $F_{k,N-k,\alpha}$.

4. 4. Regression problem and methods for solving it

As mentioned above, the regression problem is one of the basic problems in data analysis. The relationship between two variables X and Y is considered. Let one of the variables (X) be independent (predictor), and the other (Y) be dependent (response) in this relationship.

Since in practice the work is done with a sample, and not with the general population, the task of the analysis is to construct an empirical regression equation. The most common method for finding coefficients is the least squares method (LSM).

To construct the equation of a linear regression model, the data must be normally distributed. For this purpose, a transformation using the decimal logarithm is used

$$Z_x = \lg(X); \quad Z_y = \lg(Y), \quad (6)$$

where X, Y are the initial values of the metrics, Z_x, Z_y are the normalized values.

To construct the pairwise linear regression equation, the following relation is used

$$\hat{Z}_y = b_0 + b_1 Z_x. \quad (7)$$

Confidence intervals and prediction intervals are constructed for linear regression [5]:

$$\hat{Z}_{y_i} \pm t_{\alpha/2,N-2} S_{err} \sqrt{\frac{1}{N} + \frac{\left(Z_{x_i} - \bar{Z}_x\right)^2}{\sum\left(Z_{x_i} - \bar{Z}_x\right)^2}}, \quad (8)$$

$$\hat{Z}_{y_i} \pm t_{\alpha/2,N-2} S_{err} \sqrt{1 + \frac{1}{N} + \frac{\left(Z_{x_i} - \bar{Z}_x\right)^2}{\sum\left(Z_{x_i} - \bar{Z}_x\right)^2}}, \quad (9)$$

where $t_{\alpha/2,N-2}$ is the Student's t-test, S_{err} is the regression standard error.

Since the simulation was performed on a logarithmic scale, to obtain the final result it is necessary to perform the inverse transformation (potentiation).

The nonlinear regression equation for the initial data takes the form

$$\hat{Y} = 10^{b_0} X^{b_1}. \quad (10)$$

A similar transformation is applied to the boundaries of intervals:

$$Y_{\min} = 10^{Z_{\min}}; \quad Y_{\max} = 10^{Z_{\max}}. \quad (11)$$

The following quality metrics are used for the final verification of the model: the coefficient of determination R^2, which shows the degree of linear relationship, the average value of relative error *MMRE*, prediction level *PRED*(0.25) [19].

The model is considered to be of satisfactory quality if R^2 is at least 0.75, *MMRE* is no more than 0.25, and *PRED*(0.25) is at least 0.75.

5. Results of the construction of a mathematical model for estimating the size of web applications

5. 1. Development of an algorithm for solving the problem of building a model with iterative outlier removal

Based on the mathematical apparatus substantiated above, an algorithm for solving the problem of estimating the size of web applications has been developed. The algorithm implements an iterative process of building a nonlinear regression model with preliminary data cleaning. The algorithm consists of the following stages:

Stage 1. Primary data analysis.

Calculation of basic statistical characteristics (minimum and maximum values, average, standard deviation) for the NOC (Number of Classes) and kLOC (kilo Lines of Code) metrics.

Stage 2. Normality check and data normalization.

Testing the hypothesis of a multivariate normal distribution of the input data using Mardia criteria according to formulae (1), (2). If the input empirical data do not correspond to a normal distribution, which is typical for software metrics, which often have "heavy tails" of the distribution, it is necessary to perform data normalization. In the work, a transformation using the decimal logarithm according to formula (6) is used for this purpose. This transformation makes it possible to normalize the dependence, reducing the task of constructing a nonlinear regression to a linear one.

Stage 3. Detection and removal of outliers (iterative cycle).

To ensure the stability of the model, it is necessary to clean the sample from anomalous observations (outliers). The cleaning procedure is implemented as the next verification cycle.

1. Calculation of the vector of means, covariance matrix according to formula (3) for the current data set.

2. Calculation of the square of the Mahalanobis distance d_i^2 for each project according to formula (4).

3. Calculation of the test statistic T_{Si} for each observation based on d_i^2 from formula (5).

4. Comparison of T_{Si} with the quantile of the Fisher distribution $F_{k,N-k,\alpha}$.

5. Among the points for which $T_{Si} > F_{k,N-k,\alpha}$ (outliers), the point with the maximum T_{Si} value is removed from the sample. The process returns to point 1 of this stage with a new data set. The cycle ends when no outliers are detected.

Stage 4. Calculation of linear regression parameters.

For the cleaned normalized data, the coefficients b_0 and b_1 for equation (7) are calculated using the least squares (LS) method.

Stage 5. Checking the linear regression residuals.

An important condition for the adequacy of the model is the normal distribution of the regression residuals. This check allows us to confirm that the model has taken into account all the system regularities and the errors are random.

Calculation of the model residuals $\varepsilon_i = Z_{y_i} - \hat{Z}_{y_i}$ and checking their distribution for normality. If the residuals are not normally distributed, the largest residual in absolute value is removed and a return to stage 3 is performed; recalculation is performed.

Stage 6. Construction of the confidence interval and prediction interval.

To assess the accuracy of the model, intervals are constructed for normalized data according to formulae (8), (9). For each value of the independent variable, the limits of the confidence interval and the prediction interval are calculated.

Stage 7. Inverse transformation and construction of a nonlinear model.

To derive an equation of the nonlinear regression model, the inverse transformation is used and an equation in the form of (10) is obtained.

A similar transformation is applied to the limits of intervals (11).

Stage 8. Checking the data entry into the prediction intervals.

Analysis of the correspondence of the empirical data to the constructed prediction intervals. If there are points that go beyond the limits of the prediction intervals of the nonlinear model, they are removed as those that are not described by this model, after which recalculation is performed (return to Stage 3).

Stage 9. Assessing the quality of the nonlinear model.

Calculation of final quality metrics: coefficient of determination R^2, mean relative error *MMRE*, and prediction level *PRED*(0.25). Decision-making on model adequacy.

The developed algorithm makes it possible to automate the process of building a nonlinear regression model and minimize the influence of the human factor.

5. 2. Collection, description, and preprocessing of data from web application metrics

For this work, metrics of 36 projects from the GitHub repository [20] were taken, which were developed in the Java programming language using the Spring framework; all projects are open source. The NOC metric is used as an independent variable – the number of classes (variable X), and the kLOC metric, the number of lines of code in thousands (variable Y), is used as a dependent variable. A fragment of the collected data is given in Table 1.

Table 1

Metrics of projects developed in the Java programming language using the Spring framework (data fragment)

No.	GitHub link	*Y* (kLOC)	*X* (NOC)
1	https://github.com/KeeevinW/Student-Management-Website	0.547	15
2	https://github.com/Foriee007/HOA-Manager-App	3.240	57
3	https://github.com/horatiu-popan/sky-browser	2.026	61
4	https://github.com/nicolasPalomares/MedicalClinic-System	0.921	22
5	https://github.com/taizel/donor-calendar	1.118	33
6	https://github.com/thebooleanguy/dictionary-app	0.779	19
7	https://github.com/safvan8/train-ticket-reservation-system	1.861	58
8	https://github.com/s1gawron/stock-exchange-app	2.505	60
9	https://github.com/AlertedCoffee/archive-manager	1.315	24
10	https://github.com/Youssefesprit/UniversityHostelReservationSystem	1.331	21

Descriptive statistics of the initial data set are given in Table 2.

According to Table 2, one can see that the value of the kLOC metric varies from 0.171 to 9.229 with a mean of 1.721 and a significant spread (standard deviation of 1.703). The number of NOC classes varies from 4 to 261 with a mean of 40.222 and a standard deviation of 44.823.

Table 2

Descriptive statistics of the initial data set

Metric name	Min	Max	Mean	Standard deviation
Y(kLOC)	0.171	9.229	1.721	1.703
X(NOC)	4	261	40.222	44.823

Large values of standard deviations exceeding the mean indicate high heterogeneity of the data set and asymmetric distribution. These characteristics justify the need to apply a normalizing transformation before building a regression model.

Next, the collected data set was preprocessed, which included a check for multivariate normality of the distribution. Based on analysis of the β_1 and β_2 values, it was concluded that the presented data set does not correspond to a multivariate normal distribution: $\beta_1 = 16.3555$ with a critical value of 14.8603 and $\beta_2 = 23.3002$ with a critical value of 12.5793. Data normalization was performed by applying a normalizing transformation of the decimal logarithm.

5. 3. Construction of a nonlinear regression model for estimating the size of web applications

Further construction of the model was carried out iteratively. In the first iteration, using the square of the Mahalanobis distance (4) and the test statistic (5), calculated for each project, no outliers were detected: the maximum value of d_i^2 is 9.6040, the maximum value of T_{Si} is 4.5387, the Fisher distribution quantile $F_{2,34,0.005}$ is 6.2169. The coefficients of the linear regression equation were calculated: $b_0 = -1.4289$ and $b_1 = 1.0360$, the linear regression residuals were checked for compliance with the normal distribution law, and the corresponding intervals were constructed according to (8) and (9). Using the inverse transformation, the nonlinear regression model (10) and the corresponding intervals (11) were constructed. One data point (project number 29) was found that is outside the prediction interval of the nonlinear regression.

In the second iteration, all actions were repeated for the adjusted data set containing 35 projects. Using the square of the Mahalanobis distance and the test statistics, no outliers were detected: the maximum value of d_i^2 is 8.1787, the maximum value of T_{Si} is 3.8586, the quantile of the Fisher distribution $F_{2,33,0.005}$ is 6.2478. The coefficients of the linear regression equation take the following values: $b_0 = -1.3687$ and $b_1 = 1.0016$. As a result of checking the data for inclusion in the prediction interval, two outliers were found and removed (projects with numbers 15 and 19).

In the third, fourth, fifth, and sixth iterations, outliers were found at the stage of checking the linear regression residuals. As a result, projects with numbers 10, 35, 21, and 30 were removed, respectively. No outliers were found in the seventh iteration.

Seven projects were finally removed, with project numbers: 29, 15, 19, 10, 35, 21, 30. The adjusted data set contained 29 projects. A nonlinear regression model was constructed for the resulting set, which takes the form

$$\hat{Y} = 10^{\varepsilon - 1.3976} X^{1.0070}. \quad (12)$$

Thus, the construction of a nonlinear regression model for estimating the size of web applications developed in Java using the Spring framework can be considered complete.

5. 4. Development of a program for automating calculations

To automate calculations when building a model, a corresponding program was developed, for which an analysis of the subject area was conducted, and a statement of the development task was formed; design was performed. An analysis of modern software development tools was conducted.

The developed console software application performs the import of application metrics, data normalization, and elimination of outliers. The application also constructs linear and nonlinear regression models, calculates the limits of confidence intervals and prediction intervals. Separately, the calculation of model quality parameters is performed, and plots of regression equations and intervals are constructed.

The results of software operation are shown in Fig. 1–3.

Fig. 1 shows a screenshot from the console of the executed program, which illustrates the process of executing all its functions.

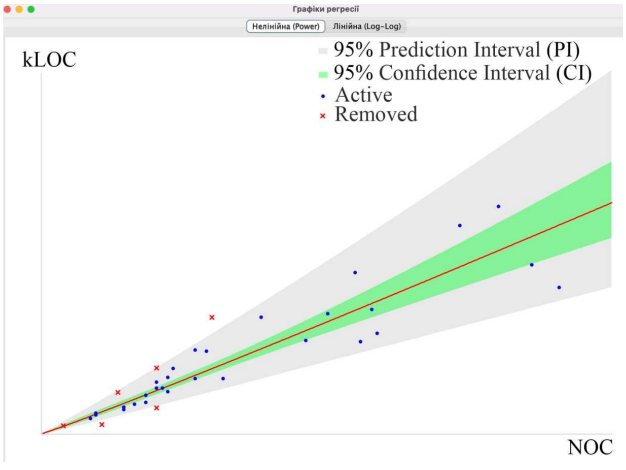

Fig. 3. Graphical representation of nonlinear regression

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java ...

[1] Checking the normality of input data (kLOC)...
-> Data are not normally distributed. Performing normalization (Log10).

[2] Outlier detection (Mahalanobis)...
   Iter. 1: Removed ID= 29 (Mahalanobis)
   Iter. 2: Removed ID= 18 (Mahalanobis)
   Iter. 3: Removed ID= 27 (Mahalanobis)

[3] Regression construction and residual analysis...
   -> Points outside the prediction interval: 1
   -> Points outside the prediction interval: 1
   -> Residuals are not normally distributed. Removing the worst case.
   -> Residuals are not normally distributed. Removing the worst case.
   -> Model is ready!

=== REGRESSION EQUATION ===
Linear (normalized):  Z(y) = -1.4463 + 1.0485 * Z(x)
Power (back-transformed   kLOC  = 0.0358 * NOC ^ 1.0485

=== MODEL QUALITY ASSESSMENT ===
1. R^2 (Coefficient of determination): 0.9248
2. MMRE (Mean Magnitude of Relative Error):  0.1704  (Requirement: <= 0.25)
3. PRED(0.25):                    0.7931  (Requirement: >= 0.75)

>>> CONCLUSION: The model is ADEQUATE and can be used for prediction
```

Fig. 1. Screenshot of the console with the result of building the model

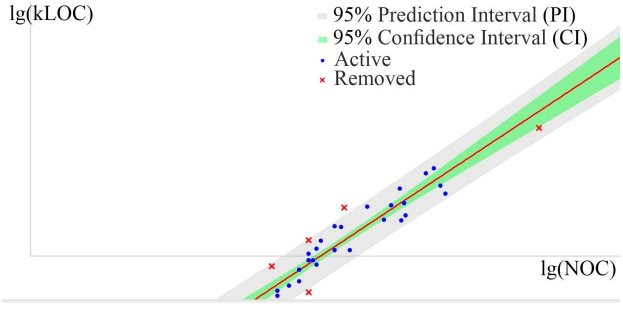

Fig. 2. Graphical representation of linear regression (fragment)

Fig. 2, 3 depict plots of linear and nonlinear regression equations with the corresponding confidence and prediction intervals.

6. Discussion of results based on the construction of a mathematical model for estimating the size of web applications

Our results are explained by the use of a normalizing transformation based on the decimal logarithm, formula (6), and iterative removal of outliers by the square of the Mahalanobis distance, formulae (4) and (5). As can be seen from Table 2, the initial metrics had significant variability and an asymmetric distribution, which was confirmed by the Mardia criteria. The logarithmic transformation allowed us to reduce the task of constructing a nonlinear regression to a linear one, and the removal of seven outliers ensured the adequacy of the model (12).

The quality assessment of the constructed nonlinear regression model was performed using parameters R^2, *MMRE* and *PRED*(0.25). A total of seven iterations were performed. In the first iteration, their values were 0.8576, 0.2299, and 0.6944, respectively. In the second iteration – 0.8858, 0.2054, and 0.6571, respectively.

For the constructed nonlinear regression model, the final quality parameters were calculated: $R^2 = 0.9173$, $MMRE = 0.1511$, $PRED(0.25) = 0.7931$. The resulting values satisfy the requirements $R^2 \geq 0.75$, $MMRE \leq 0.25$ and $PRED(0.25) \geq 0.75$. This indicates the adequacy of the constructed model for estimating the size of web applications developed in Java using the Spring framework.

A feature of the proposed approach compared to the approach from [10] is its orientation to the specifics of web applications built using the Spring framework. The model from [10], applied to the data of such web applications,

gave the following quality indicators: $MMRE = 0.8167$ and $PRED(0.25) = 0.1389$, which did not meet the requirements. The model proposed in our work produced the following quality indicators: $MMRE = 0.1511$ and $PRED(0.25) = 0.7931$, i.e., it surpassed the known model in both indicators. Unlike other works, where nonlinear models were built for Java and PHP without taking into account the framework, our model takes into account the architectural specificity of Spring applications.

In [21], a three-factor model with 571 projects of various types of applications was used for Java applications, while in this study a single-factor model was applied specifically for Spring web applications with a focus on the NOC metric.

The limitations of our study are related to the size and composition of the sample. The model was built on 29 projects after removing outliers. Its application is correct in the range of NOC metric values from 4 to 261, that is, for small and medium-sized web applications. For projects with the number of classes outside this range, the reproducibility of the declared quality indicators is not guaranteed. Another limitation is the use of only one independent variable (NOC), which simplifies the model for initial assessment but does not take into account other factors of the project complexity.

Among the shortcomings of the study, it is worth noting the small number of projects in the initial sample – 36, of which 7 were removed as outliers. To eliminate this drawback, in the future it is necessary to expand the sample to 100–150 projects.

Another drawback is the use of only one type of normalizing transformation – the decimal logarithm. This drawback can be eliminated by a comparative analysis of Box-Cox and Johnson transformations on the same data, similar to the approach in [21].

Future studies should consider the transition to multivariate nonlinear regression models using different sets of metrics. On this path, one may encounter mathematical difficulties – multicollinearity of predictors and the need to check multivariate normality in a higher-dimensional space. Experimental difficulties are associated with collecting a sufficiently representative sample of open Spring projects with a full set of metrics.

7. Conclusions

1. An algorithm for solving the problem with iterative removal of outliers by the square of the Mahalanobis distance and the Fisher criterion has been developed. A feature of the algorithm is a two-stage check for outliers: by the Mahalanobis distance and by the entry of points into the prediction interval of the nonlinear model.

2. Data collection, description, and preprocessing of 36 web applications from the GitHub repository, developed in Java using the Spring framework, were performed. Verification by Mardia criteria showed that the data did not correspond to a multivariate normal distribution ($\beta_1 = 16.3555$ at a critical value of 14.8603, $\beta_2 = 23.3002$ at a critical value of 12.5793). This is explained by the asymmetric distribution characteristic of the metrics of the considered web applications and led to the use of a logarithmic normalizing transformation.

3. A nonlinear regression model was built to estimate the kLOC metric by the NOC metric based on the normalizing transformation of the decimal logarithm. As a result of algorithm operation, seven outliers were removed (projects No. 29, 15, 19, 10, 35, 21, 30). This allowed us to obtain a stable model with the following quality parameters: $R^2 = 0.9173$, $MMRE = 0.1511$, $PRED(0.25) = 0.7931$, which satisfies the requirements for model adequacy. The difference from the considered models is the orientation specifically to Spring web applications and taking into account the architectural specificity of this framework in the sample.

4. A console software application in Java has been developed that automates data import, normalization, outlier detection, as well as model building. The application also calculates confidence intervals and prediction intervals and plots regression equations and intervals. This minimizes the influence of human factors on the results of web application size estimation.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Authors' contributions

Lidiia Makarova: Conceptualization, Methodology, Validation, Resources, Data Curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration; **Liudmyla Latanska**: Methodology, Validation, Formal analysis, Resources, Writing – original draft, Writing – review & editing; **Andrii Pukhalevych**: Validation, Investigation, Resources, Data Curation, Writing – original draft, Writing – review & editing; **Vladimir Kairov**: Validation, Investigation, Resources, Data Curation, Writing – original draft, Writing – review & editing; **Maksym Dzhurynskyi**: Software, Validation, Investigation, Resources, Writing – original draft.

References

1. TIOBE Index. Available at: https://www.tiobe.com/tiobe-index/
2. Tan, H. B. K., Zhao, Y., Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. ACM Transactions on Software Engineering and Methodology, 19 (2), 1–37. https://doi.org/10.1145/1571629.1571630

3. Daud, M., Afzal Malik, A. (2022). Construction and Validation of Early Software Size Estimation Models Based on ADAF-Adjusted ACD Metrics. The Computer Journal, 66 (9), 2123–2137. https://doi.org/10.1093/comjnl/bxac065
4. Daud, M., Malik, A. A. (2024). Exploring the Impact of Security-Based Non-Functional Requirements on Early Software Size Estimation. Computing and Informatics, 43 (5), 1234–1255. https://doi.org/10.31577/cai_2024_5_1234
5. Chatterjee, S., Hadi, A. S. (2006). Regression Analysis by Example. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc. https://doi.org/10.1002/0470055464
6. Oriekhov, O., Farionova, T., Chernova, L., Chernova, L., Vorona, M. (2024). Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications. CEUR Workshop Proceedings, 3709, 54–66. Available at: https://ceur-ws.org/Vol-3709/paper5.pdf
7. Prykhodko, S. B., Prykhodko, N. V., Koltsov, A. V. (2024). A Nonlinear Regression Model for Early Loc Estimation of Open-Source Kotlin-Based Applications. Radio Electronics, Computer Science, Control, 1, 85. https://doi.org/10.15588/1607-3274-2024-1-8
8. Prykhodko, S., Prykhodko, A., Shutko, I. (2021). Estimating the Size of Web Apps Created Using the CakePHP Framework by Nonlinear Regression Models with Three Predictors. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), 333–336. https://doi.org/10.1109/csit52700.2021.9648680
9. Prykhodko, S. B., Prykhodko, N. V. (2021). A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations. CEUR Workshop Proceedings, 3179, 156–166. Available at: https://ceur-ws.org/Vol-3179/Paper_15.pdf
10. Makarova, L. M., Prykhodko, N. V., Kudin, O. O. (2019). Constructing the non-linear regression model for size estimation of web-applications implemented in Java. Visnyk KhNTU, 2 (69), 145–153. Available at: https://journals.kntu.net.ua/index.php/visnyk/article/view/417
11. Aniche, M. Java code metrics calculator (CK). GitHub. Available at: https://github.com/mauricioaniche/ck
12. Java. Oracle. Available at: https://www.oracle.com/java/technologies/
13. IntelliJ IDEA. The Leading IDE for Professional Development in Java and Kotlin. Available at: https://www.jetbrains.com/idea/
14. Alasadi, S. A., Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12 (16), 4102–4107. Available at: https://www.researchgate.net/publication/319990923_Review_of_Data_Preprocessing_Techniques_in_Data_Mining
15. Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. Biometrika, 57 (3), 519–530. https://doi.org/10.1093/biomet/57.3.519
16. Mardia, K. V., Kent, J. T., Taylor, C. C. (2024). Multivariate Analysis. John Wiley & Sons, 592. Available at: https://www.wiley.com/Multivariate+Analysis%2C+2nd+Edition-p-9781118738023
17. Prykhodko, S., Prykhodko, N., Makarova, L. (2018). Estimating the software size of open-source PHP-based systems using non-linear regression analysis. CEUR Workshop Proceedings, 2300, 199–202. Available at: https://ceur-ws.org/Vol-2300/Paper48.pdf
18. Prykhodko, S., Prykhodko, N., Makarova, L., Pukhalevych, A. (2018). Application of the squared mahalanobis distance for detecting outliers in multivariate non-Gaussian data. 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), 962–965. https://doi.org/10.1109/tcset.2018.8336353
19. Larose, D. T., Larose, C. D. (2015). Data Mining and Predictive Analytics. John Wiley & Sons, 824. Available at: https://lmsspada.kemdiktisaintek.go.id/pluginfile.php/749864/mod_resource/content/4/Data%20Mining%20and%20Predictive%20Analytics.pdf
20. GitHub. Available at: https://github.com/
21. Oriekhov, O., Farionova, T., Chernova, L. (2024). Three-factor nonlinear regression model of estimating the size of Java-software. CEUR Workshop Proceedings, 3790, 506–518. Available at: https://ceur-ws.org/Vol-3790/paper44.pdf