Nonlinear regression models for software size estimation of Data Science and Machine Learning Java-applications

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

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

Abstract

This paper introduces the usage of regression models and equations for Data Science and Machine Learning Java applications size estimation. Size estimation of applications plays one of the key planning tasks at the early stages of project planning for the successful implementation of software development projects. Application size estimation is used to predict software development effort estimation using parametric models such as COCOMO, COCOMO II, etc. The aim of the study is to increase the reliability and accuracy of size estimation of Data Science and Machine Learning Java applications at the early stage of software project planning using class diagram metrics by building a nonlinear regression model. The object of research is the process of size estimation for open-source Data Science and Machine Learning Java applications. The subject of the study is the regression equations and nonlinear regression models to estimate the software size. To achieve this goal, we analyzed and compared the existing mathematical regression models and equations for Java applications size estimating on the sample of code metrics information from open-source Java applications of Data Science and Machine Learning. Proven the necessity of building the the three-factor nonlinear regression model for estimating the software size of Data Science and Machine Learning Java applications on the basis of the decimal logarithm normalizing transformation using the software code metrics such as the total quantity of classes, the total visible methods quantity, and the average fields quantity per class. The obtained nonlinear regression model is compared with the existing models by the regression models quality criteria such as the determination coefficient, mean magnitude of relative error and the percentage of prediction of the relative error level 0.25. The comparison confirms increasing the accuracy of software size estimation using the given sample by the obtained nonlinear regression model.

Keywords

Software size estimation, nonlinear regression model, normalizing transformation, Java, Data Science, Machine Learning, non-Gaussian data, decimal logarithm normalizing

Proceedings of the 5nd International Workshop IT Project Management (ITPM 2024), May 22, 2024, Bratislava, Slovakia

* Corresponding author.

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

0000-0002-0001-0140 (O. Oriekhov); 0000-0003-3384-4712 (T. Farionova); 0000-0001-7846-9034 (L. Chernova); 0000-0002-0666-0742 (L. Chernova); 0000-0003-4288-0096 (M. Vorona)

1. Introduction

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 at the early stages of project planning, and it helps to take into account risks, to recognize planning gaps and to increase the efficiency of the development process. If the estimates are close to the targets, then the plans can assume less risks. It is impossible to create modern software products without integrating Data Science and Machine Learning technologies, because the information technologies market requires them to integrate these technologies to obtain a competitive advantage and offer better software solutions and services to their users. The capitalization and active growth of the Machine Learning market indicates the prospects for the development of this area of the IT industry. Thus, the expected capitalization growth for 2030 is 528.1B USD, compared to 72.17B in 2022. The appearance of large language models in 2022 and 2023 has caused a real revolution in the direction of Data Science and Machine Learning in the IT and other industries [1]. As a result, it has led to the active implementation and integration of Data Science and Machine Learning technologies and developments in the IT infrastructures of various industries, such as healthcare, education, public sector, finance and economics, e-commerce, aerospace, bioinformatics, etc [2]. Java programming language is one of the most popular in the world [3] and it is widely used for the development of software projects in various areas, ranging from web applications and util application software to automotive or information systems. The infrastructure of various companies is built using this programming language. Since Java is one of the most demanded programming languages, the size estimation is an important task of the software project management life cycle for Java applications that use or implement Data Science and Machine Learning including such sub-category as informational system software.

The information about the software size for Data Science and Machine Learning Java applications allows us to predict the software development effort estimation at the early stages of project planning using well-known parametric models COCOMO, COCOMO II, SLIM, SEER-SEM [4]. The parametric models use total value lines of code to estimate the effort. The accuracy of software effort estimation allows optimizing the resources management in a rational way and, as result, allows to reduce cost of the software development. It is valuable in the case of large projects development, like informational systems, because these kinds of projects have higher risk to meet failures and issues. Modern concurrency competitions lead software companies to work on integrating Data Science and Machine Learning technologies into their software infrastructure or build brand new complex solutions. From one side, large projects development is based on traditional project management methodologies like software development life cycle and waterfall, that makes most of the parametric models more suitable for software development effort estimation than agile methodologies. From the other side, it's expected that large projects have a bigger code base on final stages of development because of the complexity of working with complex and different data. The chaos report 2015 [5] of The Standish group proves that large and grand projects have much higher failure rates in

comparison with small, moderate or medium projects. It confirms obtaining a reliable estimate of the code lines requires appropriate models for Java applications in the areas of Data Science and Machine Learning.

Both the linear [6,7] and nonlinear regression equations and models [8,9,10,11] have been built to estimate the size of open-source Java applications. The models depend on certain metrics from conceptual data models based on a class diagram. The models are based on quantitative metrics, such as the total number of classes, the number of methods (visual, public, static, etc), the total number of class fields (private, public, protected, visual), etc. and qualitative metrics, such as LCOM (lack of cohesion), RFC (response for class), etc. Different combinations of quantitative and quality metrics can have an effect on the reliability and accuracy of software size estimation. In addition, the models were built on the sample variation that may have an insufficient representation level of code metrics population for the Data Science and Machine Learning Java applications.

2. Review of the literature

Nowadays, both the linear and nonlinear regression equations and models have been developed to estimate the number code lines of Java applications and information systems, depending on the metrics of the conceptual data model based on a class diagram.

Thus, in [6], the three-factor linear regression equation for estimating the lines of code of Java applications. It is based on the methods of multiple linear regression analysis using the metrics of the total number of classes (CLASS), the total number of relationships between classes (CBO) and the total quantity of class fields (TFQ) in the source code. The paper [7] offers improved three-factor linear regression equations for estimating the lines of code for large industrial information Java systems and open-source Java applications. The model uses variables of the total number of classes (CLASS), the total number of couplings between objects (CBO), and the average quantity of fields per class (aTFQ).

The paper [8] is devoted to improving the estimation of the lines of code for industrial Java information systems and proposes a three-factor nonlinear regression model on the basis of multivariate Johnson normalizing transformation for the SB family. It is constructed on the basis of a four-dimension non-Gaussian sample dataset of software code metrics. The dataset includes the number of classes (CLASS) X_1, the total number of couplings between classes (CBO) X_2 and the average quantity of fields per class (aTFQ) X_3 from the conceptual model of the application.

In the paper [9], the one-factor nonlinear regression model was built to estimate the software size of Java web-applications. The mathematical model is constructed on the basis of the Johnson SB family normalizing transformation. It proposes estimation of code lines using the total quantity of classes (CLASS) metric as an independent factor X.

In the paper [10], the authors built the four-factor nonlinear regression model for estimating the size of open source Java applications. The model was constructed on the basis of the multivariate Johnson normalizing transform of the SB family. To estimate the value of lines of code, it uses the metrics of the quantity of classes (CLASS) X_1, the total quantity of static methods (SMQ) X_2, the total values sum of lack of cohesion of methods

(LCOM) X_3 and the total quantity of unique method invocations in a class (RFC - Response for a Class) X_4.

In the paper [11], the three-factor nonlinear regression model was constructed on the basis of the Box-Cox normalization transformation to estimate the size of Data Science and Machine Learning Java applications. It estimates lines of code variable using metrics of kilo-quantity of classes (kCLASS) X_1, kilo-quantity of visible class methods (kVMQ) X_2 and kilo-quantity of public class fields (kPFQ) X_3.

3. Formulation of the problem

The analysis of the literature has shown that the linear equations and nonlinear regression models exist for estimating Java-application code size. For the existing models, there is a necessity to confirm the possibility of using to estimate the size of Data Science and Machine Learning Java applications and, otherwise, there is a necessity to build a regression model based on the metrics of Data Science and Machine Learning Java applications for the size estimation.

Let's make the assumption that the models [6,7,8,9,10,11] can be used to estimate the size of Data Science and Machine Learning Java applications. To do this, it is necessary to confirm the possibility of using these regression equations and models by comparing them according to quality criteria for the sample of metrics of Data Science and Machine Learning Java applications.

In case of refutation of the assumption, it is necessary to build an appropriate multivariate nonlinear regression model using proper normalization transformations and compare it with the existing models [6,7,8,9,10,11] by quality criteria to improve the estimation of the size of Data Science and Machine Learning Java applications.

4. Objectives of the study

The aim of the study is to increase the reliability and accuracy of size estimation of Data Science and Machine Learning Java applications at the early stages of software development project planning using the metrics from a conceptual data model by building a nonlinear regression model.

The object of the study is the process of software size estimation of open-source Data Science and Machine Learning Java applications.

The subject of the study is regression equations and nonlinear regression models for estimating the size of open-source Data Science and Machine Learning Java applications.

5. Materials and research methods

To achieve the aim of the paper, it is necessary to analyze and compare the existing mathematical models and equations for estimating the size of Java applications on the sample of code metrics of open-source Data Science and Machine Learning Java applications; to justify the necessity of a nonlinear regression model building for estimating the size of Data Science and Machine Learning Java applications; to build a three-factor nonlinear regression model for estimating the size of Data Science and

Machine Learning Java applications using quantitative code metrics, such as the total quantity of classes, the total quantity of visible methods, the average fields quantity per class; compare the obtained nonlinear regression model with the existing models [6,7,8,9,10,11] using the quality criteria of regression models.

There are different approaches for estimating the quantity of lines of code variable using both linear and non-linear regression models. Typically, the software code metrics have non-Gaussian distribution. It limits the ability to use linear models to estimate the variable of lines of code. One of the theoretical conditions for the usage of linear regression models says that the regression residuals ε should have Gaussian distribution.

The following approaches can be used to normalize data: decimal logarithm, square root, Box-Cox transformation, Johnson transformation, etc. The normalizing transformations allow the construction of linear regression models based on normalized data with their further inverse transformation into nonlinear regression models.

The following quality criteria are used for forecasting quality assessment of regression models: 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

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

where N - sample size and MRE_i is the value of the magnitude of relative error for the i-th line of data of a random variable.

$$MRE_i = \left|\frac{(Y_i - \hat{Y}_i)}{Y_i}\right|. \tag{2}$$

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

$$PRED(0.25) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 \; if \; MRE_i \leq 0.25\\ 0 \qquad otherwise\end{cases}. \tag{3}$$

The acceptable values of $MMRE \leq 0.25$ 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 [12].

The nonlinear regression models construction techniques are proposed in the papers [13,14]. It uses statistical techniques for detecting and discarding outliers in non-linear regression analysis of non-Gaussian data based on the reciprocal normalizing transformations. There are following steps to construct the nonlinear regression model.

There is bijective normalizing transformation of a non-Gaussian random vector of the sample $P = \{Y, X_1, X_2, \ldots, X_k\}^T$ into Gaussian random vector $T = \{Z_Y, Z_1, Z_2, \ldots, Z_k\}^T$ is given by

$$T = \psi(P), \tag{4}$$

where k is number of factors (regressors or independent variables) and the inverse transformation of (4) is given by

$$P = \psi^{-1}(T), \tag{5}$$

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

The linear regression model for multivariate normalized data according to (4) will have the form

$$Z_y = \hat{Z}_y + \varepsilon \; = \; \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \ldots + \hat{b}_k Z_k + \varepsilon. \tag{6}$$

where ε is Gaussian random variable, $\varepsilon \sim \mathcal{N}(0, \sigma_{\varepsilon^2})$; $\hat{b}_0, \hat{b}_1, \hat{b}_2, \ldots, \hat{b}_k$ - estimators for parameters of the linear regression model (6). The estimators are calculated by the least square method. To build a nonlinear regression model, the inverse transformation (5) is applied to the linear regression model (6). The nonlinear regression model will have the form

$$Y = \psi_Y^{-1}(\hat{Z}_y + \varepsilon) = \psi_Y^{-1}(\hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \ldots + \hat{b}_k Z_k + \varepsilon). \tag{7}$$

The choice of factors in regression models should take into account the level of multicollinearity, because a high level of correlation between regression factors increases the sensitivity of the model to random changes in the data. The multicollinearity level is determined by the variance inflation factors (*VIFs*) of the independent variables. For a multiple linear regression model with k factors X_i , *i=1 2,..., 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].

The decimal logarithm normalization transformation is chosen to normalize multivariate data

$$T = log_{10}(X), \tag{8}$$

where X takes the values $Y, X_1, X_2, \ldots, X_k$ respectively.

Normality distribution of multivariate data is checked with Mardia test [16]. The test is based on measurement of multivariate skewness ($\beta_{1,k}$) and kurtosis($\beta_{2,k}$) of the sample.

$$\beta_{1,k} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[(X_i - \underline{X})^T S_N^{-1} (X_j - \underline{X}) \right]^3, \tag{9}$$

$$\beta_{2,k} = \frac{1}{N} \sum_{i=1}^{N} \left[(X_i - \underline{X})^T S_N^{-1} (X_i - \underline{X}) \right]^2, \tag{10}$$

where X is k-dimensional vector of random variable, $X = (X_1, X_2, \ldots, X_k)$ and S_N is a biased sample variance matrix of multivariate random variable X. it is given by

$$S_N = \frac{1}{N} \sum_{i=1}^{N} \left(X_i - \underline{X} \right) \left(X_i - \underline{X} \right)^T, \tag{11}$$

where $\underline{X}$ is a means vector of independent variable of the sample, $\underline{X} = \left(\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_k \right)^T$.

Accordingly to Mardia test, the following conditions should be verified to confirm if the sample has normal distribution

The test statistics for $\beta_{1,k}$ has the form

$$\frac{N}{6} \beta_{1,k} \; \leq \chi^2 , \tag{12}$$

where χ^2 is approximated Chi-Square distribution with $k(k+1)(k+2)/6$ degrees of freedom and α is a significant level (Accordingly to the proposed technique, α=0.005).

For $\beta_{2,k}$, the test statistics is 1-α quantile of the normal distribution $\mathcal{N}$ with the mathematical expectation $\mu \; = k(k+2)$ and variance $\sigma^2 = 8k(k+2)/N$

$$\beta_{2,k} \leq \mathcal{N}_{1-\alpha}(\mu, \sigma^2) . \tag{13}$$

Outliers of the sample are detected using the techniques proposed in [14]. It is based on outliers detection using the squared Mahalanobis distance and prediction intervals of the regression model. According to the methods the outlier should be removed from the sample. The squared Mahalanobis distance is elements on the main diagonal of the d^2 matrix of size *N x N*

$$d^2 = \left(Z_i - \underline{Z}\right)^T S_N^{-1}\left(Z_i - \underline{Z}\right), \quad (14)$$

where S_N is a biased sample variance matrix (11), Z is a normally distributed random variable. According to the technique the elements of the main diagonal of $d_i^2, i = 1,2,\dots N$ matrix are detected as outliers if the values are exceeded the threshold value of the Chi-Square χ^2 distribution quantile for the significant level - α. These outliers should be excluded from the sample.

The prediction interval outlier detection technique is based on prediction intervals of the nonlinear regression model. If the outliers are detected, ones are discarded from the sample. For the nonlinear regression model (7), the prediction interval is given by

$$\hat{Y}_{PI} = \psi_Y^{-1}\left(\hat{Z}_{PI}\right), \quad (15)$$

where $\hat{Z}_{PI}$ is the upper and lower bounds of the prediction interval (*PI*) of the linear regression (6), ψ_Y^{-1} is the inverse normalizing transformation function of the linear regression estimators. The linear regression prediction interval is given by

$$\hat{Z}_{PI} = \hat{Z}_Y \pm t_{\alpha/2,v} S_{Z_Y} \left\{1 + \frac{1}{N} + (Z_X^+)^T S_{XX}^{-1} (Z_X^+)\right\}^{1/2}, \quad (16)$$

where $t_{\alpha/2,v}$ is a quantile of *T*-Student distribution with $v = N - k - 1$ degrees of freedom and α/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 $\left\{X_{1_i} - \underline{X_1}, X_{2_i} - \underline{X_2}, \dots, X_{k_i} - \underline{X_k}\right\}$; S_{XX} is k x k matrix

$$S_{XX} = \left[S_{X_q} S_{X_r}\right], \quad (17)$$

where $S_{X_q} S_{X_r} = \sum_{n-1}^{N}\left(X_{q_i} - \underline{X_q}\right)\left(X_{r_i} - \underline{X_r}\right)$, $q,r = 1,2, \dots, k$.

It is recommended to detect and discard the outliers in an iterative way. Only one outlier should be discarded per iteration and the nonlinear regression model building should be started from the start for the new sample.

The authors have collected the sample dataset of the code metrics of 74 Data Science and Machine Learning Java applications hosted on the GitHub platform (https://github.com). The following metrics were obtained using the CK static code analysis tool (https://github.com/mauricioaniche/ck). The sample includes lines of code of projects (KLOC), total quantity of application classes (CLASS), total quantity of unique method calls in a class (RFC), total quantity of class method cohesion values (LCOM), total quantity of static methods (SMQ), total quantity of visible methods (VMQ), total quantity of class fields (TFQ), total quantity of public class fields (PFQ), and total coupling between objects per project (CBO). The resulting metrics are shown in Table 1. These metrics (excepting KLOC) can be obtained at an early stage of project planning from the conceptual data model of the application.

Table 1

Code metrics data of Data Science and Machine Learning Java-applications

No	KLOC	CLASS	RFC	LCOM	SMQ	VMQ	TFQ	PFQ	CBO
1	154.568	1846	29339	119364	1389	11778	6956	1858	12936
2	49.161	1336	11140	16698	356	4666	1565	50	6648

3	47.105	749	9356	20581	522	4196	1891	319	4538
4	156.854	2426	39140	91254	980	11188	5713	641	15681
5	258.993	3983	68958	118105	3665	23662	10382	1136	34060
6	28.49	420	7048	6426	284	2379	1706	166	3159
7	76.739	1056	10592	25654	1084	5340	2254	85	7004
8	8.056	200	2237	5447	72	1018	575	72	1414
9	32.999	874	9465	11517	344	3060	1699	44	6060
10	20.371	489	4594	8350	267	2179	863	38	2554
11	105.104	2665	31496	24294	1293	9591	8376	1633	16794
12	256.154	4875	65846	93741	2534	21235	21736	3872	35223
13	95.439	1817	20416	28676	1485	8306	3597	408	11144
14	64.504	1231	10769	13558	526	4765	2243	222	6746
15	69.624	916	15078	114992	598	5873	2725	935	6544
16	112.035	689	10340	130485	1075	6822	2386	466	4913
17	9.181	174	2419	1411	99	581	621	27	1316
18	20.01	350	5969	18491	347	1934	639	105	3368
19	19.825	351	4088	6380	221	1384	1539	542	1946
20	51.23	1098	9638	47295	338	6321	2424	65	7372
21	2.804	94	801	308	45	308	129	15	482
22	9.741	246	1905	597	20	656	429	132	1231
23	204.252	1881	23616	120572	2528	14443	5019	1397	14196
24	222.656	3583	54495	116866	2109	20901	12404	3531	24063
25	98.284	1300	13860	136085	2023	9615	2944	585	7026
26	458.003	7874	69161	280770	5084	33458	15244	2736	51272
27	58.864	543	9378	67209	530	3695	2397	949	2825
28	29.343	677	6346	21494	198	2849	1177	64	3012
29	14.469	414	5141	3072	279	1447	599	18	2834
30	29.704	661	5889	21791	371	3229	2016	387	3206
31	15.737	298	3624	3116	108	1520	628	76	2286
32	5.754	148	1463	1193	55	386	228	17	1250
33	288.533	1832	35493	392349	4364	24020	6991	1492	18037
34	21.604	410	4655	5956	418	1617	409	10	2364
35	70.037	830	11175	24682	648	5571	3717	245	5671
36	7.301	140	751	826	61	441	268	103	606
37	196.02	112	513	190	90	178	286	240	370
38	87.712	1175	16976	40783	1130	10252	2827	150	8468
39	12.985	324	2508	2274	130	1405	616	20	1903
40	112.035	689	10340	130485	1075	6822	2386	466	4913
41	107.002	1516	17094	33395	582	8329	6012	1641	8270
42	119.496	2225	17922	77010	2328	12826	3867	1372	12775
43	4.549	100	756	828	46	495	192	12	524
44	4.595	71	839	6022	37	742	165	14	304

45	137.599	2972	22950	146479	1564	16302	4735	87	37418
46	20.453	238	2714	2754	179	1067	995	372	1271
47	5.976	134	1686	2353	59	632	279	80	951
48	41.079	1034	12111	5999	492	3703	1935	186	8419
49	61.270	785	13793	29641	608	4306	2936	550	4674
50	1.649	35	491	170	7	88	83	11	228
51	29.591	256	4843	16139	278	2080	879	199	1783
52	2.161	51	591	183	14	110	178	11	278
53	6.885	172	1363	6054	186	711	320	57	497
54	47.598	633	11122	18341	1360	3408	1749	53	4188
55	1,937	30	396	211	4	132	72	20	175
56	88.460	1015	18741	108232	823	7454	3667	351	8617
57	290.27	2051	45729	207314	2870	22467	9431	1438	14621
58	3.539	81	643	559	48	407	153	14	307
59	19.314	216	2587	2964	217	1288	404	64	1313
60	35.976	973	8506	5836	405	2810	1533	421	5530
61	160.264	2448	27288	71252	841	17224	2993	875	14937
62	230.858	3522	73969	198341	3770	19467	10878	3139	31783
63	50.934	563	10508	32786	416	4520	1092	1	4025
64	339.938	3736	67276	542432	6848	31993	13658	1904	30498
65	1.846	36	498	568	5	170	99	23	288
66	50.934	563	10508	32786	416	4520	1092	1	4025
67	12.676	178	3373	3345	106	812	537	8	1200
68	224.262	647	12226	332701	2268	11526	2458	502	5496
69	43.157	725	12236	15464	722	4030	1436	92	5787
70	3.146	130	908	1499	42	421	131	21	654
71	200.338	5619	58686	89350	3652	21631	8405	1216	35804
72	2.877	78	1053	743	40	311	252	74	697
73	4.216	139	964	570	47	473	218	18	727
74	207.825	2341	39198	319963	4196	18941	8916	1121	18056

6. Experiment

6.1. Comparing the quality and accuracy of size estimation of regression models and equations

The regression models and equations [6,7,8,9,10,11] were tested by the multiple coefficient of determination R^2, the mean magnitude of relative error MMRE, and the percentage of prediction for magnitude of relative error threshold is less than 0.25, PRED(0.25).

The result of the linear regression models [6,7] didn't achieve reliable estimation of the number of lines of code, which confirmed that the values of the regression models quality criteria are exceeded the permissible range of values.

The three-factor nonlinear regression model [8] for estimating the number of lines of code (KLOC) of industrial information systems cannot be applied to the sample from Table 1, since the values of 53 data points do not meet the permissible range for applying the normalizing transformation of this model.

The one-factor nonlinear regression model [9] and the four-factor nonlinear regression model [10] have unsatisfactory values of MMRE, PRED(0.25) quality criterias. Thus, for model [9] MMRE=0.9806 and PRED(0.25)=0.0, and for model [10] MMRE=0.8854 and PRED(0.25)=0.0725. This indicates that these regression models cannot be used to estimate the number of lines of code for Data Science and Machine Learning Java applications.

The test of the model [11] according to Table 1 has the following quality values R^2=0.8985, MMRE=0.2014, PRED(0.25)=0.66. As we can see, the value of PRED(0.25) is less than 0.75, which indicates its insufficient quality level.

Thus, it is required to build the three-factor nonlinear regression model (7) to estimate the software size (in KLOC) for Data Science and Machine Learning Java applications using decimal logarithm normalizing transformation.

6.2. Building the three-factor nonlinear regression model

To increase the reliability and accuracy of size estimation of Data Science and Machine Learning Java applications, the three-factor nonlinear regression model is built on the basis of decimal logarithm normalizing transformation (8) using the data sample from Table 1. The model is based on the parameters of total quantity of classes (CLASS) X_1, the total quantity of visible methods (VMQ) X_2, and the average quantity of fields per class (aTFQ) X_3.

The model factors X_1, X_2 and X_3 were tested for multicollinearity by the VIFs [15]. For factors X_1, X_2, and X_3, the VIFs are equal to 6.1194, 6.4795, and 1.2016, respectively. This indicates the absence of multicollinearity between the factors of the regression model.

Let's check the null hypothesis if the linear regression model can be applied for software size estimation in the following form

$$Y = \hat{Y} + \varepsilon = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon. \quad (18)$$

The factors of the three-factor linear regression model (18) are estimated using the least squares method: $\hat{b}_0$=-13.863468, $\hat{b}_1$= 0.001468, $\hat{b}_2$= 0.010861, $\hat{b}_3$= 8.149336.

The regression residuals ε are tested for normal distribution converging by Chi-Square (χ^2) test with the significance level $\alpha = 0.05$ and degrees of freedom $\nu = 4$. The hypothesis is rejected, because the test value χ^2 = 163938.42 is greater than the quantile of Chi-Square distribution, which equals 9.4877. Thus, the building of the nonlinear regression model is justified.

To build the nonlinear regression model, the sample from Table 1 is normalized using the decimal logarithm normalization transformation (8). Following [16] multivariate skewness (9) and kurtosis (10) are estimated for the normalized sample from Table 1. In this case the values of skewness and kurtosis equal to 33.11 and 63.44 respectively, which are higher than threshold values of Mardia test conditions (12) and (13). The test does not confirm the normal distribution of the normalized sample. Using the square Mahalanobis

distance technique, 2 data rows are detected in the normalized data at row positions 37 and 68 which have squared Mahalanobis distance values equal to 62.42 and 17.78 respectively. The values exceed the threshold of the Chi-Square distribution quantile χ^2 = 14.86 for the multivariate normalized data for the significance level α = 0.005. Only one row with the highest squared Mahalanobis distance value is discarded per iteration, because the technique detects more than one outlier per iteration but the rows may not be outliers on further iterations.

Next, a linear regression model is built for the normalized data

$$Z_y = \hat{Z}_y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \quad (19)$$

The estimators for the parameters $\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3$ of the linear regression model (19) for the normalized data are calculated using the least square method.

Applying the inverse normalizing transformation to (8) the linear regression model (19), the nonlinear regression model has the form

$$P = \psi_Y^{-1}(\widehat{Z_y} + \varepsilon) = \psi_Y^{-1}(\hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon), \quad (20)$$

For the model (20), the upper and lower bounds of the prediction interval (15) are determined for the significance level α = 0.05. The data rows 8, 44, 72 are outside of the prediction interval. According to the technique [14], the data rows are detected as outliers and removed from the sample.

The outliers are discarded iteratively with only one data row per iteration.

The estimators for the parameters of the regression model (19) at the last iteration are: $\hat{b}_0 = -1.855315$, $\hat{b}_1 = 0.149619$, $\hat{b}_2 = 0.822320$, $\hat{b}_3 = 0.349549$. The normal distribution was tested and confirmed for the regression residual random variable ε using Chi-Square test. The test value χ^2 = 4.1411 is smaller than the value of the quantile of Chi-Square distribution 9.4877 for the significance level α = 0.05 and degrees of freedom ν = 4. According to Maridia's test, the normalized sample of 69 data rows has normal distribution since multivariate skewness (12), which is 29.39, is less than the quantile of Chi-Square distribution, which is 40.00 for 20 degrees of freedom and significance level α = 0.005. Analogically kurtosis (13), which is 22.10, is less than quantile of normal distribution, which is 28.30 for the mean of 24.

By applying the inverse transformation to decimal logarithm normalized transformation (7), the three-factor nonlinear regression model (20) has the form

$$\hat{Y} = 10^{\varepsilon - 1.855315} X_1^{0.149619} X_2^{0.822320} X_3^{0.349549}. \quad (21)$$

The quality criteria of the nonlinear regression model (21) have the following values of indicators: R^2=0.8937, MMRE=0.1867, PRED(0.25)=0.7703. This confirms the high quality of the obtained model in comparison with the existing models. The regression model (21) leads to increase the reliability and accuracy of the number of code lines estimating for Data Science and Machine Learning Java applications.

7. Conclusion

An important problem of increasing the reliability and accuracy of estimating the size of Data Science and Machine Learning Java applications is solved.

The scientific novelty of the obtained results is that the nonlinear regression model for multivariate non-Gaussian data is improved by constructing a three-factor nonlinear

regression model for estimating the size of the code lines of Data Science and Machine Learning Java applications based on the decimal logarithm normalizing transformation. This model has a higher value of the multiple coefficient of determination R^2, a lower value of the mean relative error MMRE, and a higher value of the percentage of prediction of the relative error level PRED(0.25) in comparison to the existing regression models and equations.

The practical significance of the obtained results is that the software that implements the built model was developed using the Kotlin programming language and the Apache Math3 mathematical package. The experimental results allow us to recommend the built model for use in practice. The model can reduce risks and cost of software development for Data Science and Machine Learning Java-applications or information systems.

Prospects for further research may include the usage of other multivariate normalizing transformations and data sets to build a nonlinear regression model.

8. Acknowledgements

The authors would like to express their sincere gratitude to the Doctor of Technical Sciences, Professor Sergiy B. Prikhodko, NUOS, Mykolaiv, Ukraine for his support in applying the mathematical apparatus for the research and Doctor of Technical Sciences, Professor Serhii K. Chernov, NUOS, Mykolaiv, Ukraine for the offer and motivation to participate in the conference.

References

[1] G. Press, Top Machine Learning Statistics to know [2024], What'stheBigData.com, 2023. URL: https://whatsthebigdata.com/top-machine-learning-statistics/.

[2] N. Chinthamu, M. Karukuri, Data Science and Applications, Journal of Data Science and Intelligent Systems, vol. 00, 2023, doi:10.47852/bonviewJDSIS3202837.

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

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

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

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

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

[8] N. V. Prykhodko, S.B. Prykhodko, A nonlinear regression model for estimation of the size of Java enterprise information systems software, volume 85 of Modeling and Information Technologies, 2018, pp. 81-88. URL: http://nbuv.gov.ua/UJRN/Mtit_2018_85_14

[9] L. M. Makarova, N.V. Prykhodko, O. O. Kudin, Constructing the non-linear regression model for size estimation of web-applications implemented in Java, volume 69 of Herald (Kherson National Technical University), 2019, pp. 145-153. URL: http://eir.nuos.edu.ua/handle/123456789/4443

[10] 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, volume 70 of Scientific Notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences, 2020, pp. 157-162. doi:https://doi.org/10.32838/2663-5941/2020.2-1/25

[11] O. S. Oriekhov, T. A. Farionova, Three-factor nonlinear regression model for estimating the size of Data Science and Machine Learning projects created using the JAVA programming language, volume 4 of ITMAS – 2023: Information Technologies: Models, Algorithms, Systems, Mykolaiv: NUOS, Ukraine, 2023, pp. 45-47. URL: https://itconf.nuos.edu.ua/2023/proceedings/.

[12] D. Port, M. Korte, 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. ACM, New York, 2008, pp. 51–60. doi:10.1145/1414004.1414015

[13] 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, volume 1265 of 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

[14] 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.

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