RESEARCH ARTICLE

SAMRIDDHI Volume 15, Issue 1, 2023 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Software Effort Estimation using Machine Learning Techniques

Resmi V, Anitha K L

Department of Computer Science, Mar Ivanios College, Trivandrum, India

Abstract

Success of the software development companies is mostly dependent on the best effort prediction. If the predicted effort is somewhat correct, then the company can find relief from the great tension of hurrying up the employees to get the job done within targeted time. There are many estimation methods, techniques and tools that are available. But it is very difficult to select the best one for a particular project. Each method has its own advantages and disadvantages. And also the effort estimation depends on various parameters. It is the responsibility of the project manager to select the best tool for his project. Based on the historical data, the project manager can find effort value of the new project after applying some statistical methods and data mining techniques on that data. The main aim of this work is to reveal how much accurate are data mining-classification techniques on software project effort prediction datasets when we perform analogy based effort estimation.

Keywords: Software project effort estimation models, Linear regression, Multilayer perceptron, Bagging, Decision table and Decision Tree.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2023); DOI: 10.18090/samriddhi.v15i01.12

Introduction

The need for software project effort prediction has been increasing during the last twenty years. The predicted effort is used to find the overall cost and duration of the project. This prediction may lead to either under-estimation or over-estimation [2]. If it is over or under, it causes several problems in the company's business plans. Especially it causes several budgeting problems and schedule slippage [13]. Some software project effort estimation models such as COCOMO, SLIM, DELPHI and Machine Learning methods have been developed and used to avoid such problems.

The project manager who is responsible for software project effort estimation must be competent in effort estimation models and techniques. Once the estimation is accurate, the project manager can avoid many future problems in the project [14]. Estimation methods range from old classical models to current machine learning methods. These models are purely based on either linear regression or non-linear regression. Such models take only size of the project as input [7] [14]. One example of such model is COCOMO. COCOMO and SLIM models are also known as empirical models and are popular models [1][5]. Other than simple COCOMO model, the remaining models take some additional parameters for estimating accurate effort. Though more models are incorporated with many parameters, complexity of the estimation process is also increased. Nowadays, to improve the accuracy of the estimation, algorithmic models are combined with analogy based and machine learning based estimation processes.

Corresponding Author: Resmi V, Department of Computer Science, Mar Ivanios College, Trivandrum, India, e-mail: vreshmi15@gmail.com

How to cite this article: Resmi V, Anitha KL. (2023). Software Effort Estimation using Machine Learning Techniques. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 15(1), 86-90.

Source of support: Nil

Conflict of interest: None

Software engineering estimation models are used for project budgeting, planning, scheduling and risk analysis [14]. There are two major steps in determining how long a project will take and how much it will cost. The first is to estimate its size, the second is to use size along with other environmental factors to estimate effort and its associated cost. Sizing is the prediction of coding needed to fulfill requirements. Estimation is the prediction of resources needed to complete the project of predicted size, taking into account factors of calendar time, staff, and budget constraints [4]. There are three main categories of estimation techniques [11]. They are algorithmic estimation, expert judgment and machine learning. Algorithmic estimation is based on a mathematical formula to relate independent variables (such as cost drivers) to dependent variables (such as effort, cost). Most of the models follow regression analysis and mathematical formulae. So this model is also known as a mathematical model. Expert judgment [4] is based on the opinion of the experts in the estimation process whose experiences in the past projects are taken into account. Normally the brainstorming sessions of the experts are conducted to predict the effort. The success of this approach depends on the experts' language skills, domain knowledge and the current trend of the software developments. Estimation by Analogy is one form of expert judgment and it is also known as Top-down Estimating. This technique is used to determine the duration of the project. Analogous estimating uses similar past projects' historical data to estimate the duration or cost of current projects, thus the term used is analogy. Machine learning method of estimation has been popular for the last two decades. Because machine learning-based estimation gives more accurate results when compared with the previous two methods [11]. The machine learning method uses AI based techniques to give better results.

Literature Review

Estimation based on analogy compares the estimated project with the already completed projects based on some measures. Here the measurement is mostly distance measures. Distance measure is used to find how closely one project relates to others. In the initial stages of software development, software project effort estimation is very difficult. To get more accurate results, experience of the previous project effort estimation attributes is taken into consideration. On these attributes mining techniques are applied to get the effort prediction for the current project.

S.Malathi and Dr.S.Sridhar [10] stated that approach based on fuzzy logic, linguistic quantifiers and analogy based reasoning is to enhance the performance of the effort estimation in software projects dealing with numerical and categorical data. Mamoona Humayun and Cui Gang [11] reported that ML methods give us more accurate effort estimation as compared to the traditional methods of effort estimation. The work of Zeinab Abbasi Khalifehlou and Farhad Soleimanian Gharehchopogh [15] addresses various issues of software effort prediction via Fuzzy Decision Trees (FDTs), Artificial Neural Networks (ANN), Bayesian Networks (BN) classifiers. Mohammad Azzeh and Ali Bou Nassif [12] proposed a new method based on bisecting k-medoids clustering algorithm to find the best set of analogies for effort prediction. Karel Dejaeger, Wouter Verbeke, David Martens, and Bart Baesens [9] performed comparative study on various techniques including tree/rule-based models like M5 and CART, linear models such as various types of linear regression, non-linear models (MARS, multilayered perceptron neural networks, radial basis function networks, and least squares support vector machines), and estimation techniques that do not explicitly induce a model (e.g., a case-based reasoning approach). The results showed that ordinary least squares regression in combination with a logarithmic transformation performs best. Zeynab Abbasi Khalifelou, Farhad Soleimanian Gharehchopogh [16] compared and evaluated data mining techniques with algorithmic models in software cost estimation and they suggested that Data mining techniques improve the estimation accuracy of the models in many cases.

Data Mining Techniques

Data mining techniques play vital role in data analysis. In mining, intelligent methods are applied to extract data patterns. It is the process of discovering interesting patterns and knowledge from large amounts of data [8]. Linear Regression and Multilayer Perceptron are the data mining classification techniques discussed in this paper.

Linear Regression

Linear regression [8] is used to model the- relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted by x. Linear Regression models are of the form

y = b + wx (1)

Where, b and w are regression coefficients specifying the Y-intercept and slope of the line, respectively. These coefficients can be thought of as weights. So that we can rewrite the above expression as

y = w0 + w1x (2)

Let D be the training set of tupels that contains |D| datasets of the form (x1,y1),(x2,y2).......(x|D|,y|D|). The regression coefficients (15) can be estimated with the following equations

$$\mathrm{w}1=\frac{\sum_{i=1}^{|D|}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{|D|}(x_i-\bar{x})} \qquad (3)$$

$$\mathrm{w}_0=\bar{y} - \mathrm{w}_1\bar{x} \qquad (4)$$

Where and are the mean of x and y, respectively.

If there is one explanatory variable then it is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression [3].

Multilayer Perceptron

A Multilayer Perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. This model consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron with a non-linear activation function [6]. MLP utilizes a supervised learning technique called backpropagation for training the network.

Figure 1: shows the multilayer perceptron neural network. This model has three layers: input layer, hidden layer(s), and output layer. The inputs are given in the input layer. The number of nodes in the input layer corresponds to number of input attributes. The nodes where the output is produced are in the output layer. The number of output nodes corresponds

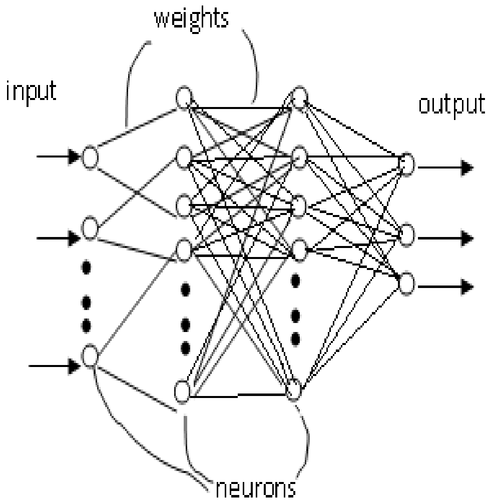

Figure 1: Multilayer Perceptron

to number of classes. The nodes in between input layer and output layer are in the hidden layers.

Each connection between node has a weight (a number) w. Each node performs a weighted sum of its inputs and thresholds the result.

Performance Analysis Model

The performance analysis model of the work carried out in this research and is shown in Figure 2.

The above diagram shows that various datasets are given as input. Then it applies the data mining classification techniques, such as linear regression and multilayer perceptron on the given datasets and it gives outputs like actual effort value, predicted effort value, correlation coefficient, mean absolute error, root mean square error, relative absolute error and root relative squared error for each dataset. Based on those values, mean magnitude of relative error (MMRE) is computed and performances of the mining techniques on the various datasets have been assessed and thereby we can predict the best model for that dataset. Here we have discussed only two classifiers.

Datasets

Three datasets have been selected for assessing the performance of the data mining techniques for prediction based on analogy. They are Cocomo81, Cocomonasa60 and Cocomonasa93. Here Weka3.7.14 is used to assess the performance of the data mining algorithms on the various datasets.

Error Measures

The following measures are used to measure the accuracy of the predictor.

- Correlation Coefficient

Correlation tells how much actual and predicted are related. It gives values between –1 and 1, where 0 is no relation, 1 is very strong linear relation and –1 is an inverse linear relation (i.e. bigger values of actual indicate smaller values of predictor, or vice versa).

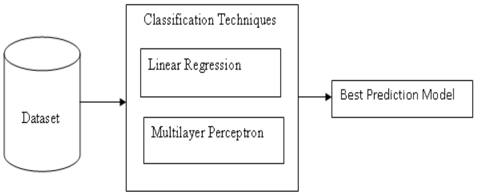

Figure 2: Analysis model

- Mean Absolute Error (MAE)

$$\text{MAE}=\frac{\sum_{i=1}^{d}|act_i-pred_i\,|}{d}\ (5)$$

Where d is the no. of data tuples, act is the actual value and pred is the predicted value. MAE does not exaggerate the presence of outliers.

- Mean Squared Error (MSE)

$$\text{MSE}=\frac{\sum_{i=1}^{d}(act_i-pred_i\,)^2}{d}\ (6)$$

MSE exaggerates the presence of outliers.

- Root Mean Squared Error (RMSE)

Square root of MSE is known as RMSE. This measure allows the error measured to be of the same magnitude as the quantity being predicted.

- Relative Absolute Error (RAE)

The absolute error of the measurement shows how large the error actually is, while the relative error of the measurement shows how large the error is in relation to the correct value.

Relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor.

$$\text{RAE}=\frac{\sum_{i=1}^{d}|act_i-pred_i|}{\sum_{i=1}^{d}|act_i-\overline{act}\,|}\ (7)$$

Where is the mean of actual values.

- Relative Squared Error(RSE) and Root Relative Squared Error(RRSE).

In RAE and RRSE we divide those differences by the variation of act. So that, they have a scale from 0 to 1 and if we multiply this value by 100 we get similarity in 0-100 scale (i.e. percentage). The values of $\sum|\,\overline{act}-act_i|^2$ or $\sum|(\overline{act}\text{-}act_i|)|$ tells how much act differs from its mean value. Because of that the measures are named “relative” - they give the result related to the scale of act.

- Mean Magnitude of Relative Error (MMRE):

There are many measures to predict the accuracy of the effort prediction models. But the commonly used measure is mean magnitude of relative error (MMRE).

The MMRE can be measure by the following formula,

Table 1: Performance Analysis of data mining techniques on Cocomo81,Cocomonasa60 and CocomoNasa93 datasets.

S. No	*Data set*	*Techniques*	*Corr.Coeff.*	*MAE*	*RMSE*	*RAE*	*RRSE*	*MMRE*	*Pred (25)%*
1	Cocomo81 63 instances 17 attributes	Linear Regression	0.61	874.47	1480.80	96.37	80.66	17.82	6
		Multilayer Perceptron	0.67	662.35	1651	72	89	5.22	9
2	Cocomo nasa60 60 instances 17 attributes	Linear Regression	0.80	247.05	431.76	57.30	64.83	1.65	30
		Multilayer Perceptron	0.89	179.45	310.36	41.62	46.60	0.82	40
3	Cocomo Nasa93 93 instances 17 attributes	Linear Regression	-0.31	645.91	1142.47	100	100	1.8	19
		Multilayer Perceptron	0.50	765.12	1319.89	118.45	115.33	2.86	20

$$\mathrm{RSE} = \frac{\sum_{i=1}^{d}(act_i - pred_i)^2}{\sum_{i=1}^{d}(act_i - \overline{act})^2} \quad (8)$$

$$\mathrm{RRSE}=\sqrt{\mathrm{RSE}} \quad (9)$$

$$\mathrm{MMRE}=\frac{1}{n}\sum_{i=1}^{n} MRE\ i$$

Where MRE is Magnitude of Relative Error.

$$MRE = \frac{|act_{effort} - est_{effort}|}{|act_{effort}|}$$

MMRE ≤.25 is the acceptable range.[17]

- Prediction (PRED):

This is also another one measure to estimate the accuracy.[18]

$$PRED\,(25) - \frac{k}{n} \quad (12)$$

where k is the number of observations whose MRE is less or equal to .25 and n is the number of observations

Results and Discussions

The following table shows the correlation coefficient, mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE) and root relative squared error (RRSE) of each of the dataset. To select the best model we have to choose high predictions and smaller MMRE value (≤ 25). From the above table Table 1, it is seen that none of the datasets shows an MMRE value less than 25 and average prediction value. Only cocomonasa60 dataset shows 40% prediction on multilayer perceptron classifier model. Multilayer perceptron model shows results better than linear regression models in all the datasets. So, it is shown that no model is suitable for all types of projects and no model is suitable for all datasets. We can improve the performance of these models little more if we apply data preprocessing steps. We have to select the best model for estimation after performing many analysis on the datasets with the help of available estimation models.

Figure 3 shows the MMRE and Pred(25) values of each dataset for linear regression and multilayer perceptron graphically.

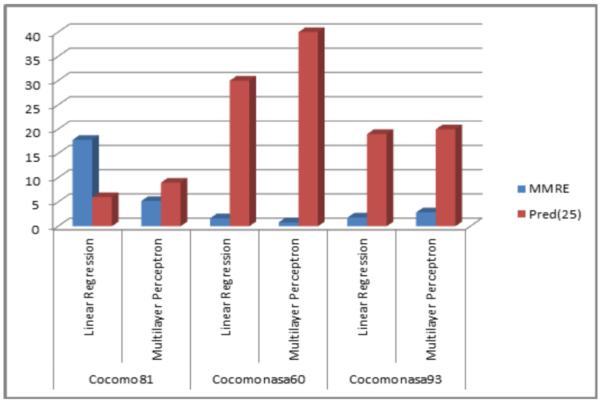

Figure 3: MMRE and Pred(25) of Cocomo81, cocomonasa60 and cocomonasa93

Conclusion

According to the results of the several researchers, project failure is due to inaccurate estimation. Many methods, techniques and tools are available to estimate the effort. But it is very difficult to select which method is the best method for the current project. So it is the project manager's responsibility to select the best method that fruitfully suits his project based on the parameters such as the complexity, domain, team members' capacity, development method etc. This paper briefs the different estimation techniques and mining algorithms for performing analogy based estimation. This paper shows that Cocomonasa 60 gives better prediction when compared with other datasets for multilayer perceptron classification and linear regression classification.

References

[1] Abdel-Hamid, T.K, "Adapting, Correcting, and Perfecting Software Estimates A Maintenance Metphor Comuputer", IEEE Computer, Vol.26 no.3, March 1993, pp 20-29.

[2] Barry, W. Boehm, "Software Engineering Economics", Englewood Cliffs, NJ, Prentice-Hall,1981.

[3] Conte S D, Dunsmore D E, and Shen V Y, "Software engineering metrics and models," Benjamin-Cummings Publishing, 1986.

[4] *David A. Freedman. Statistical Models: Theory and Practice. Cambridge University Press,2009, p. 26.*

[5] Futrell Robert T., Shafer Donald F., Safer Linda I., "Quality Software Project Management", Pearson Education, Asia 2002.

[6] Genuchten ,Van M. and Koolen, H. , "On the Use of Software Cost Models,", Information and Management, Volume 21, 1991,PP. 37- 44.

[7] Haykin, Simon , Neural Networks: A Comprehensive Foundation (2 ed.). Prentice Hall,1998.

[8] Jairus Hihn, Hamid Habib-agahi, "Cost Estimation of Software Intensive Projects:A Survey of Current Practices", IEEE, 2011.

[9] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques" Second Edition, Elsevier, Reprinted 2008.

[10] Jørgensen M, "Experience with the accuracy of software maintenance task effort prediction models", *IEEE Transactions on Software Engineering*, Vol. 21, No. 8, 1995

[11] Karel Dejaeger, Wouter Verbeke, David Martens, and Bart Baesens, " Data Mining Techniques for Software Effort Estimation: A Comparative Study", IEEE Transactions on Software Engineering, Volume:38, Issue: 2, ISSN :0098-5589, pp:375 – 397.

[12] Malathi,S., Dr.Sridhar,S., "Estimation of Effort in Software Cost Analysis for Heterogeneous Dataset using Fuzzy Analogy", (IJCSIS) International Journal of Computer Science and Information Security,Vol.10, No.10,2012.

[13] Mamoona Humayun and Cui Gang, "Estimating Effort in Global Software Development Projects Using Machine Learning Techniques", International Journal of Information and Education Technology, Vol. 2, No. 3, June 2012.

[14] Mohammad Azzeh, Ali Bou Nassif, " Analogy-based effort estimation: a new method to discover set of analogies from dataset characteristics", IET Software, Volume:9 , Issue: 2 , ISSN: 1751-8806, pp:39 – 50.

[15] Putnam, Lawrence *H.*, "A General Empirical Solution To The Macro Software Sizing And Estimation Problem", IEEE Transactions on Software Engineering, July 1978, pp. 345–361.

[16] Suri, P.K. PhD, Pallavi Ranjan, "Comparative Analysis of Software Effort Estimation Techniques", International Journal of Computer Applications (0975 – 8887) Volume 48– No.21, June 2012.

[17] Zeinab Abbasi Khalifehlou , Farhad Soleimanian Gharehchopogh, " A Survey of Data Mining Techniques in Software Cost Estimation" , 2nd World Conference on Information Technology (WCIT-2011) .

[18] Zeynab Abbasi Khalifelu , Farhad Soleimanian Gharehchopogh, "Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation", Procedia Technology 2012 , pp:65 – 71.