5th IEEE International Conference on Recent Advances and Innovations in Engineering- ICRAIE 2020 (IEEE Record#51050)

Linear Regression Model for Agile Software Development Effort Estimation

Amrita Sharma
dept. of Computer Applications
Manipal University Jaipur
Jaipur, India
amritasharma9468@gmail.com

Neha Chaudhary
dept. of Computer Science & Engineering
Manipal University Jaipur
Jaipur, India
chaudhary.neha@jaipur.manipal.edu

2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE) | 978-1-7281-8867-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICRAIE51050.2020.9358309

***Abstract*— Software cost estimation is always an essential task for the development management as it requires for estimating the effort and the time required for developing the software. A project manager requires software estimation for making a decision and predict the total budget. Success or failure of software development depends on the accurate estimation of cost and time. There are numerous tools and techniques have been developed for estimating the software cost. But all these techniques are best suitable for the traditional development methodology. From the past two decades, the agile methodology has been com for software development. So the traditional cost estimation techniques may not give the appropriate results for agile development. In this paper, the multiple linear regression models are proposed for comparing the best model for agile development. The correlation between the dependent and independent variables are also found out. The results showed that the proposed model outperforms from the decision tree, stochastic gradient boosting, and random forest.**

Keywords*— Agile Development, Software cost Estimation, Correlation, Linear Regression Model.

I. INTRODUCTION

Software cost estimation has been an essential and challenging task from the beginning of the computer era. Accurate estimation is essentially required for the software development which decides whether the development of the project will lead to success or failure. The need for accuracy in software estimation has grown with the size and development methodologies. In the literature, numerous software effort estimation approaches are found for software development using the traditional methodology. However, from 2001, the software development is moving towards to the agile software development because of flexibility, customer-friendly, and rapid delivery of working software [1][2]. The existing software cost estimation models are limited for the projects developed using the new development methodologies. The diverse concept of agile software development is not appropriate to be calculated with the function point estimation or other traditional methods.

II. SOFTWARE EFFORT ESTIMATION TECHNIQUES

Software estimation techniques are evolved from the past three decades. Most of the software cost estimation techniques, tools, and the model-based techniques use the historical data collected from the previously completed projects form a mathematical formula that is utilized to estimate the software cost. The existing cost estimation techniques are classified as model-based techniques, learning-oriented techniques, expert-based approaches, and analogy based approaches [3]. Several researchers explicate numerous software effort estimation models such as COCOMO and its extensions [4], ESTIMAC, SEER-SEM, PRICE-S, SLIM, PUTNAM's model [5] etc. These models use the software size as the main input for the estimation models with some other factors and the parameters such as complexity factors, environmental factors, project attributes, personal attributes, etc. The learning-oriented models are learned from the previous estimation experience [6]. The models are built by training them from the previous estimation techniques. These models are more suitable for the new data that comes over time. The learning-oriented include the techniques such as artificial intelligence, neural network [7], machine learning [8], case-based reasoning [9], optimization techniques [10], fuzzy logic [11], etc.

The model-based techniques are useful when quantified and empirical data is not available. On the other hand, the expert-based approach is not required the quantified and empirical data [12]. This approach estimates the software cost of the new project from the similar characteristic project completed in the past. The analogy based technique is as similar to the expert-based approach [13]. It also uses information from the previously completed project for software cost estimation [14].

III. AGILE DEVELOPMENT EFFORT ESTIMATION TECHNIQUES

Software development effort estimation in the traditional methodology is determined by the workload capacity of the team member. Apart from it, agile development uses a slightly different approach for estimation. In the agile approach, the workload is assigned to the entire team except for the individual member of the team. The agile methodology does not ask the manager to estimate the time but ask to the team members that use the effort and the degree of difficulty for the scrum. The agile methodology prescribes the different abstracted metric for estimating the effort instead of the time such as numeric sizing, t-shirt sizes, and Fibonacci sequence.

IV. RESEARCH PROBLEM

Several estimation techniques exist that are used for cost estimation for the sequential development of software. These techniques may not appropriate for the estimation for agile development as it is iterative [15]. The estimation results will be inaccurate if the traditional techniques of software development are used for agile development. On the other side, the current approaches of agile development are based on the regression that uses the single linear regression model for agile software development estimation. In this paper, the different multiple linear regression models are proposed for the agile software development estimation. Besides, the correlation between the different independent variables and the dependent variables are also identified. The comparison of different MLR model is also discussed in this paper.

978-1-7281-8867-6/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: University of Brighton. Downloaded on June 19,2021 at 04:36:58 UTC from IEEE Xplore. Restrictions apply.

V. Evaluation criteria

All the MLR models are evaluated using the different evaluation criteria such as R2 (adjusted coefficients of determination) and MSE (mean squared error) which are given in equation 1 and 2 respectfully.

$$R^2 = 1 - [\frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \hat{y}_i)^2}] \quad (1)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^2 \quad (2)$$

Where the R^2 is the coefficient of determination, ε is the residual error value.

VI. Proposed model

The existing software estimation models are required for estimating the duration, cost and personnel for the traditional software development. But the agile development is different from the traditional development as it has the development responsibility on the whole team of the development except the team member. That's why the proposed model estimates the development time of the agile development besides the personnel requirements. To estimate the effort the agile developers mostly used the Fibonacci sequence and numbering size [16]. One another method where the effort is calculated with the story size and the complexity of the agile software. The unit of effort for agile development is story point [17]. The story has the five values of user stories based on the size. The complexity scale also has the five values which define the level of complexity. The more complexity means more uncertainty in the software project. The effort is defined as the product of the user story and complexity. The other variable velocity is considered by the effort divided by the length of sprint required to develop the agile software. In the agile development estimation, the other factors like friction factors and the dynamic factors are used for calculating the declaration. The friction and dynamic factors are showing in table 1 and 2 [3] [17] [18]. The declaration is the rate of negative change of velocity. The final velocity is defined as the observed velocity powered by the product of friction factors and dynamic factors. The development time of an agile project is then calculated by dividing the effort from the total velocity of the agile project. The unit of the development time is the days.

TABLE I FRACTION FACTORS[17]

Friction Factor	Stable	Volatile	Highly Volatile	Very Highly Volatile
Team Composition	1	0.98	0.95	0.91
Process	1	0.98	0.94	0.89
Environmental Factors	1	0.99	0.98	0.96
Team Dynamics	1	0.98	0.91	0.85

TABLE II DYNAMIC FACTORS[18]

Variable Factor	Normal	High	Very High	Extra High
Expected Team Changes	1	0.98	0.95	0.91
Introduction of New Tools	1	0.99	0.97	0.96
Vendor's Defect	1	0.98	0.94	0.9
Team member's responsibilities outside the project	1	0.99	0.98	0.98
Personal Issues	1	0.99	0.99	0.98
Expected Delay in Stakeholder response	1	0.99	0.98	0.96
Expected Ambiguity in Details	1	0.98	0.97	0.95
Expected Changes in environment	1	0.99	0.98	0.97
Expected Relocation	1	0.99	0.99	0.98

VII. Linear Regression Model

The linear regression model identifies the relationship between the dependent and independent variables. The linear regression model finds out the best fit straight line that minimizes the sum of the squared residuals of the linear regression models. The two most used linear regression models are a single linear regression model and a multiple regression model. If the linear regression has only one dependent and one independent variable, it is called a single linear regression model. In the multiple linear regression model, the regression has one dependent and multiple independent variables. The multiple linear regression has the formula which is defined in equation 3 below [19]:

$$y_i = a + b_1 x_1 + b_2 x_2 + \cdots + b_m x_m + \varepsilon_i, i = 1, \ldots . n \quad (3)$$

Where the y_i is the dependent variables, $x_1 \ldots x_m$ are the independent variables. The a is the intercept and the $b_1 \ldots . b_m$ are the coefficients of the regression.

VIII. Experiment design

The research process is conducted by obtaining data and find out the correlation between the dependent and independent variable. Afterwards, the regression models are created by using all variable of a dataset. The models were evaluated with the help of evaluation metrics. In the last, the proposed models are compared with the previous methods for agile development.

In this work, the proposed multiple linear regression models are designed as the metric where each row represent the data. The regression required the historical data for the evaluation of the time of the development. in the dataset, the dependent variable yi is equalled to the Time and the independent variables are the Effort, initial velocity (Vi), a declaration which is the product of friction and dynamic factors (FF*DF), total velocity (V), and the workdays. All the regression models are constructed using the computational software where the stepwise principle was followed. The correlation was tested using a significant level of p=0.05.

IX. Dataset

The proposed experiment was evaluated on the Zia dataset. The Zia dataset is obtained from [3]. The dataset has 21 projects from the 6 software houses. The dataset contains

Authorized licensed use limited to: University of Brighton. Downloaded on June 19,2021 at 04:36:58 UTC from IEEE Xplore. Restrictions apply.

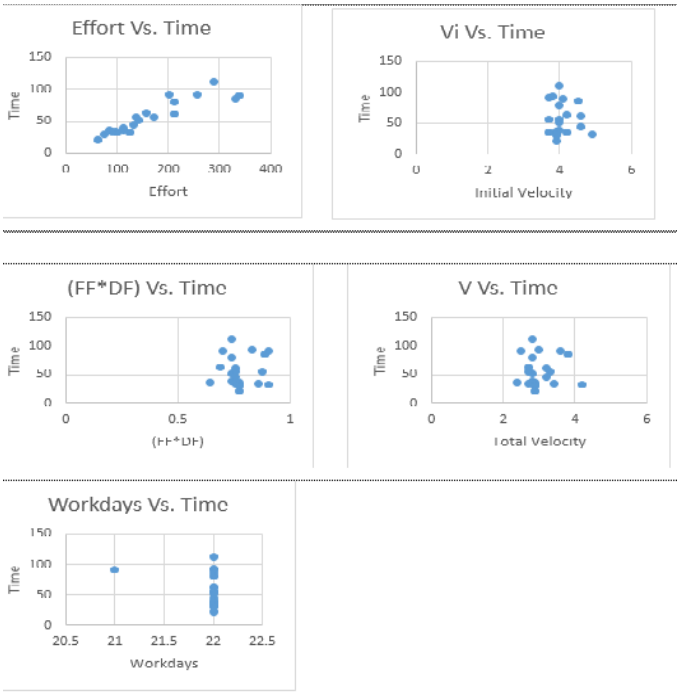
the dependent variable which is the time of the agile development and the independent variables velocity, effort, product of friction and dynamic factors, and the initial velocity. The workdays are also included as the independent variable.

X. Analysis of the Agile Variables

The correlation between the variables of the dataset is showing in table 4. Table 4 describe that the dataset has a high correlation between the effort and the development time. This shows that the effort is the most important variable for estimating the development time. The other parameters are weakly correlated. The correlation values for the independent variables are coming in the range from -0.31135 to 0.913679. The weakest correlated variable from all is the workdays with the lowest correlation value. Figure 1 is also showing the correlation between all variables of the dataset.

TABLE III CORRELATION COEFFICIENTS BETWEEN DEPENDENT AND INDEPENDENT VARIABLES

Variables	coefficients
effort	0.913679
Vi	-0.05302
D	0.08268
V	0.014762
WD	-0.31135



XI. Regression Model Evaluation

The multiple linear regression models are constructed with stepwise regression. The models are showing in table 4. The regression models for the dataset are showing in equation 5, 6, and 7.

$$Model1 \quad Time \sim 1 + Effort + Vi + D + V \qquad (4)$$

$$Model2 \quad Time \sim 1 + Effort + Vi + D \qquad (5)$$

$$Model3 \quad Time \sim 1 + Effort + V^2 \qquad (6)$$

In the above models, the first two models are the linear models while the last model is the polynomial model. The evaluation of the models was done with the formula discussed in equation 1 and 2. All the regression models are statistically significant. The evaluation of the models is showing in table 5.

TABLE IV EVALUATION VALUES FOR THE DATASET OF REGRESSION MODELS

Model	R2	MSE
Model 1	0.9476	718.1487
Model 2	0.927106	999.0206
Model 3	0.934	903.6212

XII. Analysis of the Regression Models

From the analysis of table 5, model 1 has the best results for estimating development time for agile development. The proposed multiple regression models are analyzed and compared based on the mean magnitude relative error. The experimental work was compared with the previous work. The researchers in past used the different machine learning techniques such as decision tree, stochastic gradient boosting, and random forest for effort estimation of agile development [20]. The present work has the lowest value of MMRE which showing that the work is less error and found out the better development time for agile development. The comparison of models is showing in table 6.

TABLE V COMPARISON OF MODEL 1 AND PRECIOUS WORK

	MMRE
Decision Tree	0.38
Stochastic Gradient Boosting	0.16
Random Forest	0.25
Model 1	0.099

XIII. Conclusion

In the present work, the correlation is found out between the dependent and independent variables. From the correlation values, the effort is more correlated with the development time with the highest correlation values. Using the variables of the dataset, three multiple linear regression models were constructed to estimate the effort for agile software development. From the three models, the first two models are linear while the last one is the polynomial model. The models are evaluated with the validation criteria R2 which is the coefficient of determination and the MSE (Mean Squared Error). After validating models, the Model 1 is outperform followed by the Model 3 and Model 2. Model 1 was compared with the previous work where the estimation of agile development was done by using the decision tree, stochastic gradient boosting, and random forest. The comparison was done using the MMRE (Mean Magnitude Relative Error). The proposed work has achieved a minimum error compare to the previous work.

REFERENCES

[1]. Kaushik, A., Tayal, D. K., & Yadav, K. (2019). A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO. *Arabian Journal for Science and Engineering*, 1-14.

Authorized licensed use limited to: University of Brighton. Downloaded on June 19,2021 at 04:36:58 UTC from IEEE Xplore. Restrictions apply.

[2]. SCHMIETENDORF A., KUNZ M., DUMKE R. (2008) Effort estimation for Agile Software Development Projects, *Proceedings 5th Software Measurement European Forum, Milan*

[3]. Ziauddin, S. K. T., & Zia, S. (2012). An effort estimation model for agile software development. *Advances in computer science and its applications (ACSA)*, *2*(1), 314-324.

[4]. B. Boehm, *Software Engineering Economics*, Prentice Hall, Englewood Cliffs, 1981.

[5]. L. H. Putnam, A general empirical solution to the macro software sizing and estimating problem, IEEE Trans. Softw. Eng. 4(1978), 345–361.

[6]. BOEHM, B. W., ABTS, C. & CHULANI, S. (2000) Software Development Cost Estimation Approaches: A Survey. USC-CSE.

[7]. A. B. Nassif, D. Ho and L. F. Capretz, Towards an early software estimation using log-linear regression and a multilayer perceptron model, J. Syst. Softw. 86 (2013), 144–160.

[8]. Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, *137*, 184-196.

[9]. Wu, D., Li, J., & Bao, C. (2018). Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation. *Soft Computing*, *22*(16), 5299-5310.

[10]. Dhiman, A., & Diwaker, C. (2013). Optimization of COCOMO II effort estimation using genetic algorithm. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, *3*(2).

[11]. C. Lopez-Martin, A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables, Appl. Soft Comput. 11 (2011), 724–732.

[12]. BRIAND, L. C., EL EMAM, K. & BOMARIUS, F. (1998) COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. Proceedings of the 20th International Conference on Software Engineering. Kyoto, Japan.

[13]. ANGELIS, L., STAMELOS, I. & MORISIO, M. (2001) Building a Software Cost Estimation Model Based on Categorical Data. *Proceedings of the 7th International Software Metrics Symposium.*

[14]. JØRGENSEN, M. (2003) How Much Does a Vacation Cost? or What is a Software Cost Estimate? *ACM SIGSOFT Software Engineering Notes,* 28**,** 1-4.

[15]. Fowler M, Highsmith J (2001) The agile manifesto. Softw Dev 9(8):28–35

[16]. M. Cohn , *Agile Estimating and Planning*, 1st Edition, Prentice Hall, Pearson Education, ISBN-13: 978-0131479418, 2006.

[17]. Khuat, T. T., & Le, M. H. (2018). A novel hybrid abc-pso algorithm for effort estimation of software projects using agile methodologies. *Journal of Intelligent Systems*, *27*(3), 489-506.

[18]. H. Zahraoui and M. A. Janati Idrissi, "Adjusting story points calculation in scrum effort & time estimation," 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), Rabat, 2015, pp. 1-8, doi: 10.1109/SITA.2015.7358400.

[19]. Silhavy, R., Silhavy, P., & Prokopova, Z. (2017). Analysis and selection of a regression model for the Use Case Points method using a stepwise approach. *Journal of Systems and Software*, *125*, 1-14.

[20]. Satapathy, S. M., & Rath, S. K. (2017). Empirical assessment of machine learning models for agile software development effort estimation using story points. *Innovations in Systems and Software Engineering*, *13*(2-3), 191-200.

Authorized licensed use limited to: University of Brighton. Downloaded on June 19,2021 at 04:36:58 UTC from IEEE Xplore. Restrictions apply.