International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

Analysis of Software Cost Estimation Techniques

Dhwani Agrawal[1], Dr. Satwinder Singh[2]

[1]*M. Tech (Computer Science and Technology), Central University of Punjab, Bathinda.*
[2]*Assistant Professor, Dept. of Computer Science and Technology, Central University of Punjab, Bathinda.*

---***---

Abstract - *The effort estimation is most important aspect for software project development. Accurate effort estimation is a challenging task since inaccurate estimation can ruin the whole process of software development. This is because, it is most important to estimate effort accurately. Currently available models are giving estimation for effort but it can be done more precisely. By using machine learning algorithms, the models can perform better and give more accuracy to effort estimation. In this paper, basic COCOMO and function point techniques are used with ZeroR and M5Rules classifier and average absolute error is measured to compare both the techniques.*

Key Words: **Effort estimation, Machine Learning, COCOMO, Accurate estimation.**

1. INTRODUCTION

Recently, software has become the most important and extortionate product for the system projects. The main cost of any software development is mainly depended on the human effort, the cost estimation method/ technique used and then it will give the estimate in person-month. However, accurate estimation of effort is very challenging task. Because, in case of over estimation, the resources of particular organization will be wasted or in case of under estimation, the project will not be delivered in time since it will lack resources.

There are multiple methods proposed to calculate the estimated effort for the software project development, but there are some major models comes under the picture since last three decades. They are: COCOMO, Putnam, COCOMO II and Function Point Analysis. Most of the cost models are based on the size measures, such as, LOC, FP count. LOC is dependent on the programming language while FP count is independent from it.

In this paper, performance of M5 Rules and ZeroR classifier algorithm is discussed in comparison with the existing effort estimation techniques by using publicly available NASA project datasets.

2. RELATED WORK

Software cost estimation is an essential step that support and guides the planning of software projects. Software cosy estimation ready the blueprint of estimated amount of effort, time and development team size required to develop a software project at very early stage of software development life cycle process. Accurate cost estimate activity is critical to both developer and customer.

As, it is tough to map the relationship between the attributes in the effort estimation, that's why machine learning algorithm is being used as an automatic tool. In this paper, it is proposed that a technique is build that estimate more accurate by using machine learning methodology as compare to other existing techniques. The machine learning algorithm used are M5 rules and ZeroR classifier.

2.1 COCOMO MODEL

Known as Constructive Cost model, introduced in 1981 by Barry Boehm is a major technique to estimate software project effort. It is an algorithmic software cost estimation model which estimates effort, cost and schedule for the software projects. It has three sub models named as: basic, intermediate and detailed. The basic idea for effort calculation is:

$$\text{Months} = a*\text{KLOC}^b*c$$

Where, a and b are domain-specific parameters, KSLOC is total number of source lines of code and c represents the product of effort multipliers.

2.2 FUNCTION POINT

Function point is a measure introduced in 1979 to hinder SLOC. It is independent of programming language unlike SLOC. It regulates size and complexity of the software project in terms of functions. It consists of five weighted software components, through which unadjusted function point is analysed. The adjustment of these function points are done by calculating the technical complexity factors.

$$FP = UFP * (0.65 + 0.01 * TCF)$$

3. METHODOLOGY USED

In this experiment, datasets from promise data repository are used to compare and analyze models and their respective efforts. The datasets we used in this experiment are public domain datasets which were collected by NASA from real software projects. Some of the product metrics that are included in the dataset are: RELY, DATA, CPLX, AEXP, LEXP, MODP, SCED, KLOC, ACT_EFFORT. They are effort multipliers for COCOMO dataset. In these multipliers, we have to increase the value of AEXP, LEXP and MODP to decrease the effort and we have to decrease the value of CPLX, DATA and RELY to decrease the effort.

The experiments were performed on WEKA tool which is platform for conducting machine learning algorithms. The analysis of actual effort of COCOMO and function point data is examined in WEKA through 10-fold cross validation. The tests were performed using M5-Rules and ZeroR classifier with the default settings in Weka. The analysis of software effort is examined for all the datasets. Different values were obtained for each the dataset.

3.1 PERFORMANCE CRITERIA

Mean Absolute Error: It is difference between two variables in which one is predicted value and one is observed value. Mean absolute error is average of difference of absolute errors.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Root Mean Squared Error: It also measure the average of error value. It is squared difference of mean absolute error.

$$\mathrm{RMSE} = \sqrt{\frac{(a_1 - c_1)^2 + (a_2 - c_2)^2 + \cdots + (a_n - c_n)^2}{2}}$$

Performance Criteria	Techniques Used			
	cocomonasa	nasa_numeric	china2	albrecht
MAE	431.164	645.9132	3700.0519	20.3899
RMSE	665.9798	1142.4663	6492.7821	29.7585

Table 1: Performance Evaluation of ZeroR Classifier

Performance Criteria	Techniques Used			
	cocomonasa	nasa_numeric	china2	albrecht
MAE	208.6431	389.9266	435.1973	8.0587
RMSE	414.4785	835.2982	1470.4572	13.4988

Table 2: Performance Evaluation of M5Rules Classifier

Table 1 and 2 shows that the M5-Rules learner has the least MAE and RMSE value in comparison to ZeroR classifier. Hence the M5-Rules algorithm is the best methodology for classification. The 70% data is used for training and the rest is used for testing the data in WEKA in default settings of both the machine learning algorithms.

4. CONCLUSION

In this experiment, two Machine learning Algorithms, M5-Rules algorithm and ZeroR Classifier are experimented to estimate the software effort for projects. Performances of these models are tested on NASA Software Project Data and the results are compared with the COCOMO and function point techniques with their two public datasets of each as mentioned in the literature. The M5 Rule learner shows best results than among other algorithms experimented in the study with lower values of MAE and RMSE for all the datasets and able to provide good estimation capabilities as compared to other models. Hence, it is suggested to use of M5-Rules technique to build suitable model structure for the software effort.

REFERENCES

[1] Abedallah Zaid, M. H. (2008). Issues in Software Cost Estimation. *IJCSNS International Journal of Computer Science and Network Security*, Vol 8, No. 11, pp 350-356.
[2] BaniMustafa, A. (2018). Predicting Software Effort Estimation Using Machine Learning Techniques. *International Conference on Computer Science and Information Technology (CSIT)*.
[3] Basha, S., & Ponnurangam, D. (2010). Analysis of empirical software effort estimation models. *arXiv preprint arXiv:1004.1239*.
[4] Baskeles, B., Turhan, B., & Bener, A. (2007). Software effort estimation using machine learning methods. In *2007 22nd international symposium on computer and information sciences* (pp. 1-6). IEEE.
[5] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. (1995). Cost models for future software life cycle processes: COCOMO 2.0. *Annals of software engineering*, *1*(1), pp 57-94.
[6] Boetticher, G. (2001). Using machine learning to predict project effort: Empirical case studies in data-starved domains. In *First international workshop on model-based requirements engineering*.
[7] Eberendu, A. C. (2014). Software Project Cost Estimation: Issues, Problems and Possible Solutions. *International Journal of Engineering Science Invention*, Vol 3, Issue 6, pp 38-43.
[8] Galinina, A., Burceva, O., & Parshutin, S. (2012). The optimization of COCOMO Model coefficients using Genetic Algorithms. *Information Technology and Management Science*, *15*(1), pp 45-51.
[9] Heetika Duggal, P. S. (2012). Comparative Study of the Performance of M5-Rules Algorithm with Different Algorithms. *Journal of Software Engineering and Applications*, pp 270-276.
[10] Jaswinder Kaur, S. S. (2008). Comparative Analysis of the Software Effort Estimation Models . *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol 2, No 10, pp 3402-3404.
[11] Kemerer, C. F. (1987). An empirical validation of software cost estimation models. *Communications of the ACM*, *30*(5), pp 416-429.
[12] Kim, Y., & Lee, K. (2005). A Comparison of Techniques for Software Development Effort Estimating. *SYSTEM*, pp 407-417.
[13] Leung, H., & Fan, Z. (2002). Software cost estimation. In Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies pp. 307-324.
[14] Nasa, C., & Suman, S. (2012). Evaluation of different classification techniques for web data. *International journal of computer applications*, *52*(9), pp 34-40.
[15]
[16] Parmar, M. (June 2018). Comparative Analysis of Classification Techniques using WEKA on Different Datasets. *International Journal of Latest Engineering and Management Research (IJLEMR)*, Vol 3, Issue 6, pp 1-5.
[17] Rekha Tripathi, D. P. (January 2016). Comparative Study of Software Cost Estimation Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol 6, Issue 1, pp 323-328.
[18] Rshma Chawla, D. A. (2014). Software Development Effort Estimation Techniques: A Review. *International Journal of Electronics Communication and Computer Engineering*,Vol 5, Issue 5, pp 1166-1170.
[19] S. Malathi, S. S. (2012). Analysis of Size Metrics and Effort Performance Criterion in Software Cost Estimation. *Indian Journal of Computer Science and Engineering*, Vol 3, No. 1, pp 24-31.
[20] Shepperd, M. J. (2007). A Systematic Review of Software Cost Estimation Techniques. *IEEE Transactions on Software Engineering, Vol 33*, pp 33-53.