AN IMPROVED TECHNIQUE FOR SOFTWARE COST ESTIMATIONS IN AGILE SOFTWARE DEVELOPMENT USING SOFT COMPUTING TECHNIQUES

Quazi Bushra[1] Dr. Amol Kadam[2]
BVDU, College of Engineering, Pune

Abstract:

The management & estimation of agile projects is stimulating works for many software companies for their high failure rates. To develop successful software projects. Proper prediction of projects overall effort & cost evaluation is a very important task. The numbers of development models over the last few decades have evolved through software projects. Hence, to complete an exact estimation of exertion & taken a toll for diverse program ventures which is based on distinctive improvement models are having innovative & new steps of software development is a significant task which is to be done. Software companies have adopted different various development models which are based on the organization and requirement of project. In this paper we proposed a COCOMO (Constructive Cost Model) for cost estimation of better software projects. Profit or loss estimation forecast to new project is carried out with the help of historical data of company. In the machine learning to predict forecast using historic data Naïve Bayes algorithm plays vital role and provides great accuracy. To check the behavior of the proposed system here we have used the SEERA dataset. According to the result our proposed system gives the profit and loss forecast prediction with the accuracy of 86.59% and 24.80% respectively. And the overall effort calculation accuracy is higher, 95.06% in the contrast to the SVM, 93.45%.

1. Introduction:

Software cost estimation is an important task in the software design and development process. Planning and budgeting tasks are carried out with reference to the software cost values. A variety of software properties are used in the cost estimation process. Hardware, products, technology and methodology factors are used in the cost estimation process. The software cost estimation quality is measured with reference to the accuracy levels. Software cost estimation is carried out using three types of techniques. They are regression based model, analogy based model and machine learning model. Each model has a set of technique for the software cost estimation process.

1.1 Software Cost Estimation: The Software Cost Estimation is a process top predict /estimate the approximate cost of the software project before the development starts i.e. it describes the approximate requirements of effort, development time and resources to complete the software project. It is one of the vital processes to start development for software by considering all internal & external cost factors.

The cost estimation is a tool to estimate the planning, budgeting and resource utilization for the software projects. Before cost

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

estimation for a software project, we will have known that what are the actual requirements for a project, what is the complexity of those requirements, and other cost driver factors that affect the development (like, product factor, project factor, personal factor& hardware factor). These are the input to the cost estimation process. So, in general, the process provides three responses. Such as Effort, Development Duration, and Resources.

<u>Effort</u>: The amount of effort required to complete the development of software projects in terms of Man-Months (MM).

<u>Development Duration</u>: The time duration required to complete the development of a software project i.e. total development time.

<u>Resources:</u> The number of Manpower required for a software project in terms of time to complete.

But in actually the SCE process follows on cost driver factors i.e. it will affect the cost of the software. These factors are such as design methodology, memory management, experienced skills, hardware requirements, software tools, risk analysis, project complexity, project delay, size of project database, performance parameter, virtual memory environment, etc.

2. Methodology

The dataset is pre-processed in order to calculate effort using COCOMO model by apply equation 2. The estimated effort is then compare with both the actual effort achieved and the predicted effort that is obtained by applying: Naïve Bayes. The dataset used is explored using density function and projection plot s to investigate both the nature and potential of the dataset, and then a number of models are built by applying the selected machine learning techniques. techniques. The performance of the models is then evaluated using: confusion matrix, accuracy

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

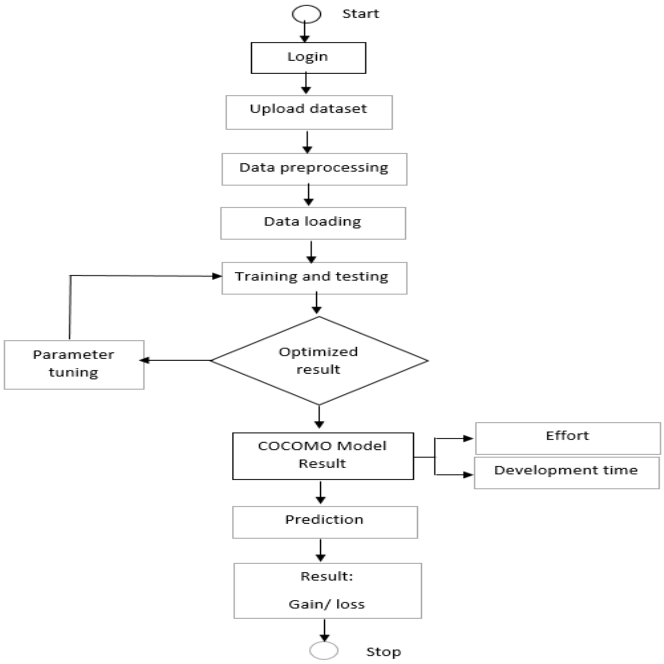


Figure 1: activity diagram of the system

1.

Data Pre-processing

In the beginning, we conduct data pre-processing phase so that the program can read the dataset. The dataset itself has 93 rows, 26 columns, and formatted as ARFF (Attribute Relation File Format). We convert it into Excel format with the help of Weka. After that, we try to convert nominal values (low, normal, high, very high, and extremely high) of EM to its corresponding value. Then, we calculate the COCOMO II effort by using its equation (1) so that the result can be used later as a comparison to other methods or algorithms. After that, we remove all irrelevant columns and left some usable column for the prediction program. Those columns are project ID, KSLOC, EM, SF, and Actual Effort. At last we convert the dataset format into comma separated values format so that the program can read the dataset. Table below is the part of pre-processed dataset.

2. Data Loading

After the data pre-processing phase, we load the dataset. The program read the dataset. After that, we map the dataset into two separate variables, independent variables, KSLOC and EAF, as X, and dependent variable, Actual Effort, as y.

3. Training and Testing

In the beginning of this phase, we call all machine learning classes that we need to train and test the dataset. Each algorithm has its own parameters.

4. Parameter Tuning

In the first iteration, we are still using default parameters from the library. If the differences are still large, we modify the parameters' value manually in order to gain the most ideal value. After several iterations of trial and error, we find the best parameter that can used for testing the dataset. Best parameter means the results error is small. At the end of this phase, we de-normalize the dataset so that we can retrieve the original value of the prediction.

5. Result comparison

After we gain the result, we compare their result and we display it in a table. We now compare the profit, loss and available values, missing values.

6. Prediction

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

These values are given to the Naïve Bayes algorithm for the future prediction and forecast is predicted using these results.

ALGORITHM

Input

Training dataset T,

F= (f_1, f_2, f_3 ,..., f_n) // value of the predictive variable in testing dataset

Output

A class of testing dataset

Steps:

1. Read the training dataset T
2. Calculate the mean and standard derivation of the predictor variables in each class,
3. Repeat:
calculate the probability of f1, using the gauss density equation in each class;
Until the probability of all predictor variables (f_1, f_2, f_3 ,..., f_n) has been calculated.
4. Calculate the like hood for each class
5. Get the greatest like hood.

Dataset used:

SEERA dataset: The SEERA dataset is a heterogeneous dataset from 57 different organizations representing the public and private sectors in Sudan. These organizations range from software development companies, to freelancers, to IT departments within public and private institutions. Table 1 provides the details of the organizations contributing project data. The public sector represents 28% of the organizations with a contribution of 40% of the projects. Only public sector software companies developed software for customers, the rest of the public organizations provided in-house software projects developed by their respective IT departments. Private software companies contributed 51% of the total projects and 85% of the projects contributed by the private sector. However, the average contribution of each private software company is one to three projects with one company contributing 13 projects. This is in contrast to the public software companies in which two companies contributed 16 and 8 projects and one company contributed two projects. To reflect the heterogeneity of the projects, the dataset includes attributes for the type of organization, sector and organization id.

	Type of organization	count	# of projects	%
Public	Software company	4	28	23%
	Federal directorates	4	6	6%
	University	5	5	3%
	Federal ministry	4	4	7%
Private	Software company	25	7	51%
	Freelancer	6	64	6%
	Corporate IT department	4	9	6%
	Telecommunication industry	5	5	3%
	Total	57	128	100%

3. Result:

In conducting the above comparison, the SEERA dataset provides recent heterogeneous

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

project data with rich attributes that can be applied for different empirical research questions. The SEERA dataset overcomes the current limitations in dataset transparency through providing detailed original raw data (sub- attributes) and coding formulas which allows researchers to create new cost estimation datasets or rescale current attributes from the original data. This allows for the replicability of results and the verification of the data. All this combined raises the quality, flexibility and trustworthiness of the SEERA dataset.

# of missing data	% of missing values	# of attributes	details
1	1	8	environment: 3, users:1, developers: 10, project: 9, product: 5
2	2	9	size: 1, environment: 3, developers: 1, Project: 1, Product; 3
3	3	2	Process reengineering (project), product complexity (product)
4	3	2	customer organization type
11	9	1	team contracts
39	33	1	% of project gain(loss)
total		13	

Table 1: attributes with missing values in the SEERA dataset

In regard to missing values per project, Table 2 details the percentage of missing values within the projects showing that the majority of projects (87%) have none or one missing value. the SEERA dataset includes attributes to distinguish the origins and characteristics of the submitting organization: organization id, organization size, and IT department size.

% of missing values	# of projects	% of projects
0	60	50%
1	33	30%
2	2	13%
3	1	1%
8	1	1%
25	2	1%
	99	100%

Table 2: projects with missing values in the SEERA dataset

As we have discussed before, in this system we are using the SEERA dataset as an input the system. Table represents the total effort and the development time of the project. This effort and development time calculated using the COCOMO Model. The last column express the profit and loss obtained by the project, some of the data is missing.

The below values of the effort and development time is calculated using the formula:

1. Estimated effort:

$$[Estimated\ duration\ \times (Dediacted\ Team\ Members + (Team\ size - Dediacted\ Team\ Members) \times 50\%)] \times (Daily\ Working\ Hours \times 22)$$

2. Actual effort:

$$[Actual\ duration\ \times (Dediacted\ Team\ Members + (Team\ size -$$

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

$Dediacted\ Team\ Members) \times 50\%)] \times (Daily\ Working\ Hours \times 22)$

3. Profit or loss is calculated using the following formula,

$$\frac{(Contract\ price - Actual\ incurred\ costs)}{Contact\ price} \times 100$$

Project Id	Year of the Project	Effort calculated	Development time	Profit
1	2015	2112	2	?
2	2016	1056	1	?
3	2008	3168	3	0%
4	2009	5280	6	-17%
5	2016	19008	12	0%
6	2012	7392	6	0%
7	2016	5280	6	?
8	2018	4400	4	0%
9	2018	4224	6	0%
10	2015	6468	12	-25%
11	2001	8910	9	0%
12	2000	5280	6	N/A
13	2016	880	2	N/A
14	2009	1848	3	N/A
15	2010	1584	3	N/A
16	2016	1320	2.5	N/A
17	2014	880	2	N/A
18	2012	264	1	N/A
19	2014	27772	4.5	N/A
20	2018	704	4	63%
21	2015	2640	3	0%
22	2014	4224	6	0%
23	2013	1408	4	0%
24	2010	2816	4	1%
25	2009	1584	4	0%
26	2018	1760	4	50%
27	2004	2112	4	-14%
28	2007	1540	7	0%
29	2004	4224	12	50%
30	2007	176	2	0%
31	1997	2640	12	-100%
32	2013	2640	6	?
33	2016	880	5	?
34	2017	2904	3	33%
35	2017	1056	3	?
36	2014	2673	9	?

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

37	2016	31680	4	0%
38	2006	3646	6	0%
39	2019	704	1	-22%
40	2019	1760	4	?

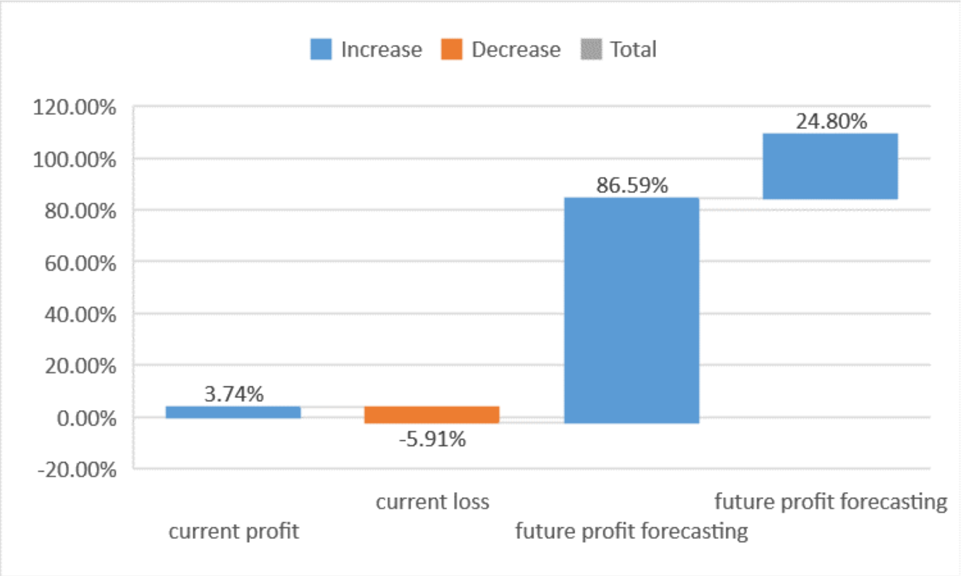
Table 3: result of the estimated effort and development time of the project using the COCOMO Model

Naïve Bayes:

Table 4 represents the result of the naïve Bayes algorithm in terms of prediction of the system. According to the result and the dataset provided we can conclude that, all the 120 projects required the development time is calculated as 680.5. while the system provides the 3.74% of profit in the upcoming project and -5.91% loss. Hence the accuracy of the total future forecasting rate is 86.59%. Thus our proposed system gives the great accuracy in the future prediction.

Parameter	**Acquired value**
Calculated Required Time	680.5
Total Number Of Projects	120
Current Profit	3.74%
Current Loss	-5.91%
Future Profit Forecasting	86.59%
Future Loss Forecasting	24.80%

Table 4: Result of Naïve Bayes



Graph 1: Graph of future profit forecasting of the system

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

The overall effort calculation efficiency through the SVM algorithm gives 93.45% of accuracy whereas the proposed designed COCOMO model has higher accuracy with the 95.06%.

4.Conclusion:

Software Cost Estimation is a critical, effective process in software development and project management, many decisions stopped according to the results of the estimation, software cost estimation needs extra efforts and cooperation from the academic researchers with a help from the industrial software development companies to achieve highly trusted cost models via exchanging expertise, models of development in addition to the software engineering best practices applied in the industrial software development company and the needed suitable data to formulate the metrics and cost models in software cost estimation process. In this paper we design a system, we used COCOMO model for the cost, time and effort estimation of ASD (Agile software development). The advantages of calculating using COCOMO II with this application are simple data that must be prepared by the user, layout of calculations with minimum wages, and a comprehensive presentation of calculation results. Our proposed system gives the accuracy of the future prediction system 86.59%. using the COCOMO II model the effort calculation accuracy is increased up to the 95.06% as compared to the SVM algorithm 93.45 %. In the future we will try to increase the accuracy of the future prediction of the profit and loss of the system more than 90% using the hybrid algorithm in the machine learning with the 97-98% overall effort calculation accuracy.

Reference

[1] S. Al Idrus, W. Nur Hidayat, and A. Hamdan, "Naya as a tool of software cost automatic analysis," *4th Int. Conf. Vocat. Educ. Training, ICOVET 2020*, pp. 258–263, 2020, doi: 10.1109/ICOVET50258.2020.9230249.

[2] S. S. Ali, M. Shoaib Zafar, and M. T. Saeed, "Effort Estimation Problems in Software Maintenance - A Survey," *2020 3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020*, 2020, doi: 10.1109/iCoMET48670.2020.9073823.

[3] M. Wang, Y. Ma, G. Li, W. Zhou, and L. Chen, "Multi-Value Models for Allocation of Software Component Development Costs Based on Trustworthiness," *IEEE Access*, vol. 8, pp. 122673–122684, 2020, doi: 10.1109/ACCESS.2020.3007158.

[4] L. S. Nair and J. Swaminathan, "Towards reduction of software maintenance cost through assignment of critical functionality scores," *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, no. Icces, pp. 199–204, 2020, doi: 10.1109/ICCES48766.2020.09138071.

[5] I. C. Suherman, R. Sarno, and Sholiq, "Implementation of random forest regression for COCOMO II effort estimation," *Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020*, pp. 476–481, 2020, doi: 10.1109/iSemantic50169.2020.9234269.

[6] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão, and E. Insfran, "An update on effort estimation in agile software development: A systematic literature review," *IEEE Access*, vol. 8, pp. 166768–166800, 2020, doi: 10.1109/ACCESS.2020.3021664.

[7] M. Hamid *et al.*, "An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects," *IEEE Access*, vol. 8, pp. 140752–140766, 2020, doi: 10.1109/ACCESS.2020.3010968.

[8] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "A comparison between decision trees and decision tree forest models for software development effort estimation," *2013*

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

3rd Int. Conf. Commun. Inf. Technol. ICCIT 2013, pp. 220–224, 2013, doi: 10.1109/ICCITechnology.2013.6579553.

[9] A. Banimustafa, "Predicting Software Effort Estimation Using Machine Learning Techniques," *2018 8th Int. Conf. Comput. Sci. Inf. Technol. CSIT 2018*, no. 1, pp. 249–256, 2018, doi: 10.1109/CSIT.2018.8486222.

[10] E. I. Mustafa and R. Osman, "SEERA: A software cost estimation dataset for constrained environments," *PROMISE 2020 - Proc. 16th ACM Int. Conf. Predict. Model. Data Anal. Softw. Eng. Co-located with ESEC/FSE 2020*, pp. 61–70, 2020, doi: 10.1145/3416508.3417119.

[11] M. Krivokuca, P. A. Chou, and M. Koroteev, "A Volumetric Approach to Point Cloud Compression-Part II: Geometry Compression," *IEEE Trans. Image Process.*, vol. 29, no. c, pp. 2217–2229, 2020, doi: 10.1109/TIP.2019.2957853.

[12] S. Bilgaiyan, S. Mishra, and M. Das, "A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques," *Proc. - Int. Conf. Comput. Intell. Networks*, vol. 2016-January, pp. 112–117, 2016, doi: 10.1109/CINE.2016.27.

[13] A. J. Singh and M. Kumar, "Comparative study on effort estimation using different data mining techniques," *Int. J. Sci. Technol. Res.*, vol. 9, no. 4, pp. 3005–3010, 2020.

[14] L. P. Dos Santos and M. G. V. Ferreira, "Safety critical software effort estimation using COCOMO II: A case study in aeronautical industry," *IEEE Lat. Am. Trans.*, vol. 16, no. 7, pp. 2069–2078, 2018, doi: 10.1109/TLA.2018.8447378.

[15] X. Chai, L. Deng, Q. Yang, and C. X. Ling, "Test-cost sensitive naive Bayes classification," *Proc. - Fourth IEEE Int. Conf. Data Mining, ICDM 2004*, pp. 51–58, 2004, doi: 10.1109/icdm.2004.10092.

[16] "Software Reliability and Cost Estimation Model" Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 6, June 2018

[17] Amol K. Kadam , S.D. Joshi1 , Debnath Bhattacharyya and Hye-Jin Kim, "Increases the Reliability of Software using Enhanced Non Homogenous Poisson Process (EHPP), Functional Point and Test Point Analysis" International Journal of u- and e- Service, Science and Technology ,Vol.10, No.9 (2017), pp.35-48,Sept 2017

[18] Amol K. Kadam , S.D. Joshi , D.Bhattacharyya, "Software Superiority Achievement through Functional Point and Test Point Analysis" International Journal of Software Engineering and Its Applications,Vol. 10, No.11, Dec 2016

[19] Amol K. Kadam, S.D. Joshi, Debnath Bhattacharyya, Hye-Jin Kim ,"Diagnosis of software using testing time and testing coverage" International Journal of Hybrid Information Technology,Volume 9, Oct 2016

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731