Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 6.886|
Vol. 17 Issue 10, Oct- 2022
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

A STUDY ON SOFTWARE COST ESTIMATION MODELS USING MACHINE LEARNING

Dr. J. Purushotham
(Academic Consultant(c), Department of Statistics, Telangana University, Nizamabad-503322, Telangana)

Dr. M. Naveen Kumar
(Programmer, Telangana University, Nizamabad-503322, Telangana)

Abstract:

At present an important trend of effort estimation has been determined. It needs to collect more data and have effective and efficient software for distribution to stakeholders, which accelerates the development of hardware and software prices. This is especially true in large industry areas, as the size of software projects is becoming more complex and larger, and the complexity of forecasting is constantly increasing. An attempt at a software engineering financial study is an attempt at how to manage limited resources in a specific project, budget, and scope so that the project can meet your goals. The project needs to create or embrace a helpful software advancement cycle to actualize the product improvement project by going about as a key obstacle. Estimated exactness is the primary basic evaluation for each examination. As of late, AI procedures have been utilized widely in the issue of huge scope endeavors however a few models have limits and differential examination is as yet not adequate. There are many models for predicting prices, such as algorithmic models, top-down and expert results. Of each one of those models, the improvement in the calculation model is higher than the others. In this paper we present a relative investigation of software cost projects utilizing calculation techniques.

Keywords: Algorithmic Method; Comparative Analysis; Software Cost Estimation,.

Introduction:

What is Software Cost Estimation?

Software development has become an essential investment for many organizations. Software engineering professionals are increasingly concerned about the value of software and software products. Learn more in: Software Cost Estimates Using Soft Computing Methods.

The way toward assessing the expense as far as exertion or time needed to create or keep a product item. Software quotes are by and large dependent on fragmented and confounding

information, requiring measurable examination and demonstrating. Learn more in: A Framework of Statistical and Visualization Techniques for Missing Data Analysis in Software Cost Estimation.

Software cost assessment is quite possibly the main parts in planning software. Software cost assessment gets one of the contemplations in deciding the proficiency of software advancement. Tragically there is a weakness in the level of accuracy in the cost assessment of software development. Apart from many methods, methods and tools still need to be improved.

The rough expense of software is identified with how long and the numbers of individuals are needed to finish a venture. The assessed cost of the product begins with the item proposition and will proceed all through the task life. The expense assessment measure incorporates size gauges, business appraisals, and improvement of primer task plans and at last gauges generally speaking undertaking cost.

Developing quotes is significant, as they are critical to the accomplishment of software improvement project. Along these lines, to oversee spending plan and software project plans, different software cost assessment models have been created. Precise software quotes are fundamental for engineers and clients. Determining software quotes is a troublesome errand in software projects improvement. It helps project directors and computer programmers to plan and deal with their assets. Nonetheless, building up a precise expense assessment model for a product project is a difficult interaction.

These days the product improvement framework is getting convoluted. The use of software emerges in many organizations. Contingent upon the association's size and went with undertakings, every movement under a product project improvement ought to be refreshed routinely. They should bargain in giving excellent software with a minimal effort spending plan. Subsequently, more complex methodologies are expected to address testing issues inside this area. Software advancement projects are an interaction in the arranging of software project the executives.

Software is otherwise called money related element on time, spending plan and degree targets and how to screen limited resources for meet destinations. The developing worry by most engineers is the unpredictability of assessment made in a beginning stage of the improvement interaction. Here and there, vulnerability influences software item improvement in an unexpected way. It increments with the size of software project assessment botches that could cost a great deal regarding asset apportioned to the venture. Since software advancement issues have numerous measurements, we need to explore the utilization of a few procedures to enhance these

Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 6.886|
Vol. 17 Issue 10, Oct- 2022
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

difficult issues, not just zeroing in on the product exertion designing Trying, yet joining different strategies that upgrade the exactness of the exertion. Numerous components influence the precision of cost assessment in software advancement.

Software Effort Estimation (SEE) emphasizes on how to estimate the effort, time and cost with the project activities plan. As a rule, in SEE, two sorts of customary strategies are applied to quantify impression of non-algorithmic methodologies, zeroing in on an algorithmic way to deal with demonstrating human science. The algorithmic cycle is created by using some mathematical showings to do the product assessment. These logical, numerical models depend on the chronicled information and utilize the data sources, for example, size, number of traits capacities and other expense drivers. The Constructive Cost Model (COCOMO) model, Function Point Analysis (FPA) model, and Putnam Model are a portion of the models that utilization the algorithmic methodology. A calculation less strategy is utilized by specialists for software project gauges.

Software examiners and specialists foresee a bunch of endeavors throughout the long term. Early software forecast models depend on assessment or mathematical assurance. Current models depend on recreation, neural organizations, hereditary calculations (GA), dark rationale, and that's just the beginning. Numerous usage and upgrades are put forth to the attempt expectation model utilizing delicate registering to beat certain impediments in the exactness of the assessments appropriate to every standard improvement of software.

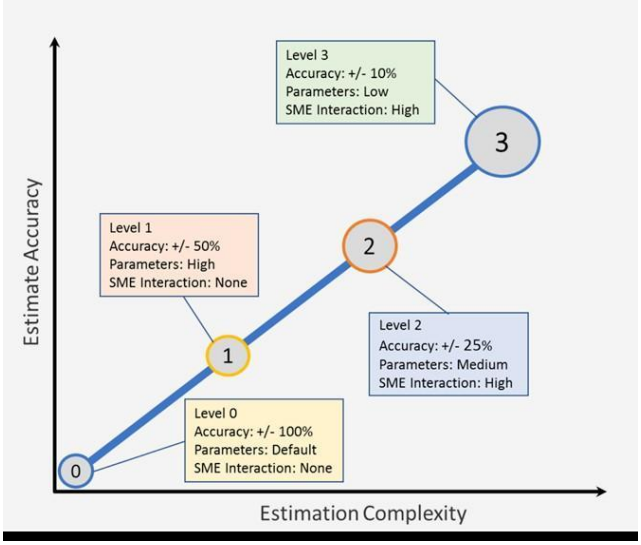

Figure: Estimation of Complexity and Accuracy

Overview of Software Cost Estimation:

Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 6.886|
Vol. 17 Issue 10, Oct- 2022
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Cost estimation has been developed since 1960. There are many methods of cost estimation but basically there are six main methods which consist of: Algorithmic (Parametric) Model, Expert Judgment (Expertise Based), Top – Down, Bottom – Up, Estimation by Analogy, and Price to Win Estimation.

We use machine appraisals to help measure information and propose new defaults to the assessor. We utilize straight relapse to measure and look at changed informational collections: current appraisals; Historical forecasts; and took care of genuine hourly bills. We additionally utilize the Bayesian method to fill openings and propose new factors. Now it is intended to propose arrangements - surmise still it will in any case can overwrite the last info and all information.

Delicate figuring measures are analyzed to lessen prescient instabilities and misguided judgments. In traditional plans, AI is a method under machine innovation, including developmental figuring's and dark rationale. In past exploration, the Particle Swarm Optimization (PSO) strategy has been utilized to tune work boundaries F (a, b) in COCOMO II and to construct a bunch of liner models (LOC) on the area of conceivable software to exploit fluffy rationale. The PSO calculation starts by making arbitrary molecule areas in the plan space of the boundaries and the exhibition of the created model was assessed with the assistance of NASA software project dataset.

THEORY OF SOFTWARE COST ESTIMATION

1. COCOMO Model:

"COCOMO", a new cost evaluation model has been introduced. This model is notable scientific portrayal for programming cost assessment. It is mainly founded on the past experience of programming activities and utilizing LOC as the unit of measures for programming size. It consists of three variants namely fundamental model and semidetached model. The essential COCOMO model assess effort to make programming progress and cost as a component of program size verbalized in evaluated LOC. The effort is determined utilizing the following equation:-

$$\text{Effort} = a*(\text{KLOC})b \quad \text{.....(1)}$$

Where, effort assessed face to face month and KLOC is evaluated number lines of codes for the project. The estimation of parameters 'a' and 'b' dependent on the project type. Software projects are grouped into three classifications dependent on the complexity of the projects

namely organic, semi-detached and embedded.(for organic projects a=2.4,b=1.05,for semi-detached a=3.0,b=1.12 and for embedded a=3.6,b=1.20).

Intermediate COCOMO model computes the estimation of programming improvement exertion as a component of program size and set of cost drivers that incorporate individual appraisal of the products, hardware ,personnel and task properties. Here, effort is determined utilizing the following condition:- Effort=a*(KLOC)b *EAF …….(2)

The estimation of the parameters a and b dependent on the project type (for organic projects a=3.0,b=1.05, for semidetached a=3.0,b=1.12 and for embedded a=2.8,b=1.20) and EAF is determined using 15 cost drivers. Each cost drivers is evaluated from ordinal scale extending from low to high.

2. Pearson product-moment correlation coefficient Model:

Pearson product-moment correlation coefficient is widely used to predict the linear relationship between two sets of data. For two variable X and Y Pearson product-moment correlation coefficient is a measure of the linear dependence between the variables. It is denoted by r and its value range lies between -1 and 1 where 1 is total positive correlation, 0 is no correlation and -1 is total negative correlation.

3. One-Way ANOVA Model:

One-way ANOVA is a general techniques for studying tested information relationship. It has been broadly utilized in different fields for example, Chen and his colleagues have utilized one-way ANOVA in hereditary engineering. For lodging staff job satisfaction fulfillment tang has been used. Ronen has utilized it to software development risks.

4. Multi-Objective Genetic Algorithm:

Multi-Objective Genetic Algorithm (MOGA) states that it is a method of solving optimization problems which involve multiple objectives such as minimizing cost and maximizing reliability and others objectives. It is different from single objective optimization in that in MOGA problem, there doesnot exist a single solution that simultaneously optimizes each objectives.

Here, the main task is to find out the trade-off surface, which is a set of non-dominated solution points, that is known as pareto optimal or non-inferior solutions. It has been seen that

none of the solutions in the non-dominated set is extremely better than any other; any one of them is an acceptable solution. The choice of one solution over the other requires problem knowledge and a number of problem-related factors.

RELATED WORKS

Through most of the Researchers used FFNN model for effort estimation task, other neural network models can also be tried. Neural network models with required changes in architecture and functions can be supportive in research for predicting effort of software development. The use of Soft Computing techniques to build a suitable model structure to utilize improved estimations of software effort for NASA software projects. On doing this, Particle Swarm Optimization(PSO) was used to tune the parameters of the COCOMO model. The performance of the developed model was evaluated using NASA software project data set. Soft Computing methods were explored to build efficient effort estimation models Structures. Kelly utilized the concept of neural networks genetic algorithms and genetic programming to introduce a methodology for software cost estimation. Three models with Fuzzy Logic and PSO Algorithm with Inertia weight was present in the research of NASA. In This research NASA datasets were used to training and testing sets. Dizaji and F.S Gharehchobogh have used chaos factor to improve the performance of PSO Algorithm.

In their Article, Tent mat, Lorenz attractor and Logistic map are used as Chaos Optimization Algorithms. These researchers used the hybrid of this algorithm with chaor factor to predict the road accidents according to accident type (damage, injury,death). Alternative prediction models ranging from regression approaches to analogy based and machine learning techniques are studied in order to cover a wide range of estimation methods proposed so far in the literature. Public domain datasets with different characteristics are used in order to address the inherent problem of prediction systems, i.e their high dependency on the types of data. Through the Cost estimation model alternative error functions measuring different important aspects of error are studied.

The Inter-Cultural Challenges Mitigation Model (ICCMM) to assist outsourcing vendor organizations in addressing intercultural challenges In outsourcing relationships. Genuine Estimates of software development costs are required in the software development cycle to find out the feasibility of software projects and to provide the required resources accordingly. Many methods have been given so far to predict the cost of the software one of the most popular being estimation by expert knowledge. In this method, the reliability of estimates leased on expert opinion depends on the fact, how much a new project agrees with the skills and experience of the

expert. Scaling is generally refers to measurements or assessments conducted under exact specified and repeatable conditions.

In ML Scaling transforms feature values according to defined rule so that all scaled features have the same degree of influence and thus the method is immune to the choice of units, which is a major stage for ML methods. Due to the cost of gathering and reporting data from projects, development teams are less focused on data collection. The Missing values have significant impact on ML estimation preformation.

COCOMO II Model:

COCOMO-II is the revised version of the Cocomo (Constructive Cost Model) and is developed at University of Southern California. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity. It consists of three sub-models:

1. **End User Programming:**
   Application generators are used in this sub-model. End user write the code by using these application generators. For example – Spreadsheets, report generator, etc.

2. **Intermediate Sector:**
   (a). Application Generators and Composition Aids –
   This category will create largely prepackaged capabilities for user programming. Their product will have many reusable components. Typical firms operating in this sector are Microsoft, Lotus, Oracle, IBM, Borland, Novell.
   (b). Application Composition Sector –
   This category is too diversified and to be handled by prepackaged solutions. It includes GUI, Databases, domain specific components such as financial, medical or industrial process control packages.
   (c). System Integration –
   This category deals with large scale and highly embedded systems.

3. **Infrastructure Sector:**

   This category provides infrastructure for the software development like Operating System, Database Management System, User Interface Management System, Networking System, etc.

Stages of COCOMO II:

Stage-I:

It supports estimation of prototyping. For this it uses Application Composition Estimation Model. This model is used for the prototyping stage of application generator and system integration.

Stage-II:

It supports estimation in the early design stage of the project, when we less know about it. For this it uses Early Design Estimation Model. This model is used in early design stage of application generators, infrastructure, system integration.

Stage-III:

It supports estimation in the post architecture stage of a project. For this it uses Post Architecture Estimation Model. This model is used after the completion of the detailed architecture of application generator, infrastructure, system integration.

Besides, a few specialists propose utilizing the relapse model to lessen the overall blunders and to apply the negative-negative coefficient, which is a good procedure for aligning the COCOMO model boundaries. Fake Neural Network (ANN), Support Vector Machine (SVM), Linear Regression (LR), K-Near Neighbor (KNN) are information mining strategies coordinated with algorithmic COCOMO II models looking for precise expectation software advancement endeavors and time gauging. Reproduced nailing (SA) was utilized to defeat the limits of GA, which made untimely union populaces, and proposed to present the COCOMO II PA model coefficients to get software commission's exact gauges and diminish the vulnerability of the COCOMO II post-engineering model.

As indicated by the examination, it was recommended to consolidate various procedures to quantify the best gauges for the field of software. As of late, the vast majority of the COCOMO II models have been concentrated by most software analysts and professionals to improve their capacity to precisely appraise the expense of an undertaking. COCOMO II is one of the various procedures of software exertion assessment. It is broadly acknowledged as an industry standard in view of its application instead of changing the phase of software study.

The precision of COCOMO II is incredibly influenced by its info boundaries which are the size, coefficient and quality drivers of the task. Little changes in these boundaries have an immense effect in assessing their endeavors. The consideration of COCOMO II with novel delicate processing and AI models has gotten famous in exploration these days as it gives the capacity to improve execution in boundary tuning and more advantage to gain for a fact information, which mostly centers around forecast exactness. Most examination utilizes a few models in blend with COCOMO II to improve the indications of a specific model. Be that as it may, the precision of the models is as yet sketchy. The COCOMO model, either COCOMO I or

COCOMO II, is a set up software exertion identified with the product motor, yet in spite of the fact that it is broadly known to the product business, it isn't utilized by and by.

There is to a greater degree a pattern of speculating software endeavors looking for commemoration innovation than proposing new ML methods. The most recent delivery on 201 improves the past ANN calculation procedure utilizing Dragonfly to give ideal preparing of the neural organization, anticipating more upgraded and exact software endeavors.

Most researchers have not told clear ways on the best way to browse explicit examinations. The absence of information for the examination interaction and the absence of data identified with the information types are causing challenges in assessing the product cycle with the assessment of such endeavors. The idea of information quality is extraordinary. This isn't simply identified with consistency yet in addition the nature of the total dataset utilized. Examination had cautioned about the utilization of awkwardness datasets as this could prompt covering of the subsequent part. The general information project exertion is vital for designers to make sensible arrangements or exercises in task the executives as far as assessing software endeavors. Episodes of missing information in task information can essentially affect future expectation endeavors.

One investigation shows that pre-handling information examination, for example, treatment of lost information can likewise influence the exhibition of the expectation model. There are a few elective approaches to deal with missing information. Now and again, erasing or eliminating a missing variable is the default technique for most cycles. In any case, there are a few examinations that investigate the guarantee of missing information in the field of software exertion estimating. There ought to be an observational quest for steadiness and exactness to deal with missing information.

Software cost assessment is a basic advance that is being taken in the beginning phases of the product improvement measure. The motivation behind such an interaction is to more readily see the improvement of things to come undertaking and its stages. The second primary target is to give clear project subtleties and highlights to assist partners with dealing with the venture regarding HR, resources, software, information and even achievability contemplates.

Precise assessment results certainly help the venture supervisor to more readily gauge the time needed for project cost, different undertaking stages and assets or resources. In any case, debasement can be brought about by the undertaking cost assessment measure which unquestionably influences the venture conveyance. Undertakings with inaccurate or errors require assets, including conveyance time, spending plan or quality or even operational issues,

and once in a while the venture may fizzle or be dropped. Consequently, cost assessment is a significant piece of software projects and subsequently it is an intricate issue in the field of software.

A great deal of studies and exploration were finished with the point of upgrading and improving the expectation cycle and getting more precise and solid outcomes. Then again, AI (ML) procedures have gotten incredibly essential in ongoing software considers. In most logical examination, ML techniques are utilized and regularly run in an assortment of fields; however, at least one strategy will be chosen relying upon the nature and destinations of the exploration.

The software cost estimation process is evolving rapidly, which may include technological advances, team skills and experience, and available tools and software languages, which give ML techniques superiority over some other methods that can survive in statistical and mathematical work. Therefore, ML can be a useful technique for creating a proposed model by adapting to the wide range of changes involved in the development of a software project capable of learning from historical data.

Conclusion:

Software cost estimation is a critical task in software projects development. It assists project managers and software engineers to plan and manage their resources. However, developing an accurate cost estimation model for a software project is a challenging process. The aim of such a process is to have a better future sight of the project progress and its phases. Another main objective is to have clear project details and specifications to assist stakeholders in managing the project in terms of human resources, assets, software, data and even in the feasibility study. Accurate estimation results with definitely helps the project manager to do better estimation for the project cost, the time required for various project phases and resources or assets.

There are as yet numerous issues with how to analyze and assess anticipating strategies. Most specialists either don't team up intimately with the product business and recent concerns, or they trust it's smarter to zero in on a substitution instead of improving the current strategies utilized by the business. In this manner, it is suggested that future examination work center around software improvement rather than proposing new substitution methods. It is acceptable to take note of that a few articles depend on datasets that are excessively old as portrayals for current or future undertakings.

Future exploration should zero in on understanding the connection between project attributes (dataset quality) and figure evaluation. Numerous articles assessed gauges utilizing verifiable datasets, yet couples were determined by the full genuine circumstance. In this way, more investigations ought to be led on the strategies utilized, in actuality, circumstances.

References:

[1] Borade, J. G., & Khalkar, V. R. (2013). Software Project Effort and Cost Estimation Techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 3(8), pg. 730–739.
[2] Choudhary, K. (2010). GA Based Optimization of Software Development Effort Estimation. International Journal Of Computer Science And Technology, Vol.(1), pg. 38–40.
[3] Choudhary, K. (2011). Parametric Estimation of Software Systems. International Journal of Soft Computing and Engineering, 1(2), pg. 17–20.
[4] Clause, J., Li, W., & Orso, A. (2007). Dytan: a generic dynamic taint analysis framework. In Proceedings of International Symposium on Software Testing and Analysis (pp.196–206). ACM.
[5] Dan, Z. (2013). Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization. In Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2013(pp. 180–185).
[6] Garbajosa, J. (2008). The emerging ISO International Standard for Certification of Software Engineering Professionals. In IFIP International Federation for Information Processing (Vol. 280, pp. 173–178).
[7] Maleki, I., Ghaffari, A., & Masdari, M. (2014). A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization. International Journal of Innovation and Applied Studies, 5(1), 72–81.
[8] Missier, P., Lalk, G., Verykios, V., Grillo, F., Lorusso, T., & Angeletti, P. (2003). Improving data quality in practice: A case study in the italian public administration. Distributed and Parallel Databases, 13(2), 135– 160.
[9] Rajper, S., & and Zubair A. Shaikh. (2016). Software Development Cost Estimation Approaches - A Survey. Indian Journal of Science and Technology, 9((31)), pg.1–5.
[10] Sandhu, G. S. (2014). A Bayesian Network Model of the Particle Swarm Optimization for Software Effort Estimation tool. International Journal of Computer Applications, 96(4), 52–58.
[11] Vu Nguyen, Bert Steece, B. B., Nguyen, V., Steece, B., Boehm, B., & Vu Nguyen, Bert Steece, B. B. (2008). A Constrained Regression Technique for COCOMO Calibration. Acm 978-1-59593-971-5/08/10, 1– 10.
[12] Zaid, A., Selamat, M. H., Azim, A., Ghani, A., Atan, R., Koh, T. W., Albrecht, A. (2008). Issues in Software Cost Estimation. International Journal of Computer Science and Network Security, 8(11), 350– 356.
[13] Zhang. Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization. Service Operations and Logistics, and Informatics (SOLI), 2013 IEEE International Conference.
[14] Y. Masoudi-Sobhanzadeh, H. Motieghader and A. Masoudi-Nejad, FeatureSelect: a software for feature selection based on machine learning approaches, BMC Bioinformatics, vol. 20, no. 170, pp. 1-17, 2019.
[15] V. Vig and A. Kaur, Test effort estimation and prediction of traditional and rapid release models using machine learning algorithms, Journal of Intelligent and Fuzzy Systems, vol. 35, no. 2, p. 1657–1669, 2018.
[16] A. Khalid, M. A. Latif and M. Adnan, An Approach to Estimate the Duration of Software Project through Machine Learning Techniques, Gomal University Journal of Research, vol. 33, no. 1, pp. 1-13, 2017.
[17] T.-H. Yeha and S. Deng, Application of machine learning methods to cost estimation of product life cycle, International Journal of Computer Integrated Manufacturing, vol. 25, no. 4/5, p. 340–352, 2012.
[18] R. Agarwal, M. Kumar, Yogesh, S. Mallick, R.M. Bharadwaj, D. Anantwar, Estimating software projects, SIGSOFT Software Engineering Notes 26 (4) (2001) 60–67.
[19] Mitchell, T. M., Machine Learning, McGraw-Hill and MIT Press, 1997.
[20] Software Cost Estimation:Metrics and Models, http://sern.ucalgary.ca/courses/seng/621/W98/johnsonk/cost.htm#Original%20COCOMO, 2006.
[21] Yucheng Kao, Jin-Cherng Lin , Jian-Kuan Wu "A Differential Evolution Approach for Machine Cell Formation" IEEE International Conference on Industrial Engineering and Engineering Management, pp.772-775, 2008.
[22] Patil,Lalit V.,Rina M.Waghmode, S.D.Joshi,and V.Khanna, "Generic model of software cost estimation:A hybrid approach",2014 IEEE International Advance Computing Conference (IACC),2014
[23] Caper Jones, "Estimating Software Cost", Tata Mc–Graw Hill Edition,2007.

***