INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Predictive analytics approaches for software effort estimation: A review

A G Priya Varshini[1]*, **K Anitha Kumari**[2]

1 Assistant Professor, Information Technology, Dr.Mahalingam College of Engineering and Technology, Pollachi, Coimbatore, India. Tel.: +91-9444899863
2 Associate Professor, Information Technology, PSG College of Technology, Coimbatore, India

OPEN ACCESS

Received: 10-05-2020

Accepted: 10-06-2020

Published: 22-06-2020

Editor: Dr. Natarajan Gajendran

Citation: Priya Varshini AG, Anitha Kumari K (2020) Predictive analytics approaches for software effort estimation: A review. Indian Journal of Science and Technology 13(21): 2094-2103. https://doi.org/10.17485/IJST/v13i21.573

***Corresponding author**.
A G Priya Varshini

Assistant Professor, Information Technology, Dr.Mahalingam College of Engineering and Technology, Pollachi, Coimbatore, India.
Tel.: +91-9444899863
priyavarshini.a.g@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

Abstract

Background/Objective: In Software Effort Estimation (SEE), predicting the amount of time taken in human hours or months for software development is considered as a cumbersome process. SEE consists of both Software Development Effort Estimation (SDEE) and Software Maintenance Effort Estimation (SMEE). Over estimation or under estimation of software effort results in project cancellation or project failure. The objective of this study is to identify the best performing model for software Effort Estimation through experimental comparison with various Machine learning algorithms. **Methods:** Software Effort Estimation was addressed by using various machine learning techniques such as Multilinear Regression, Ridge Regression, Lasso Regression, Elastic-Net Regression, Random Forest, Support Vector Machine, Decision Tree and NeuralNet to recognize best performing model. Datasets used are Desharnais, Maxwell, China and Albrecht datasets. Evaluation metrics considered are Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and R-Squared. **Findings**: Experiments on various machine learning algorithms for software Effort Estimation determines that Support Vector Machine produced the best performance comparatively with other algorithms.

Keywords: Machine learning; software effort estimation; regression models; classification models

1 Introduction

Software Effort Estimation is used to predict effort in terms of person-months or person-hours. For successful development of software, Software Effort Estimation (SEE) is one of the challenging tasks though several models exist. Several models were proposed for software effort estimation [(1)]. Initially Software Effort estimations are carried out using Expert judgment, User Stories, Analogy based estimations and Use case point approach. Later various Machine learning algorithms alike Linear regression, Logistic regression, Multiple linear regression, Stepwise regression, Ridge regression, Lasso regression, Elasticnet regression, Decision tree, Neural networks, Support vector machine, Random forest, Naïve bayes, etc., are used for estimation. Ensemble approaches also gained more attention and produce more prediction than individual algorithms for effort prediction. The following are the survey of various models used for effort estimation.

According to **Expert judgment method** [(2)], estimation is produced based on the judgmental process. It includes experience and advice of experts based on the degree to which the new project matches with the previously completed projects of the expert with their experience. The techniques used for expert estimation are Delphi technique and Work Breakdown Structure (WBS) [(3)].

User stories approach or feedback[4] to avoid inaccurate estimations. **Analogy based estimation** is based on collected past data about a project[5]. The project manager and team work together for this kind of approach. Once the requirements of the present project are known, they search for a similar past project from the database[6]. **Function point estimation** is a reliable approach[7] that checks the functionality of the system based on the user's point of view. Function point analysis is initially used to estimate size using which effort can be estimated[8]. **Use case point approach**[9] is based on the use cases involved in the project. Generally use cases are user interaction with the system. The actors or users are categorized and weighted based on the type of work they perform[10].

Supervised machine learning techniques are majorly used for estimation. It consists of input variables or independent variables and an output variable or target variable that is to be predicted from input variables. Types of supervised learning are as follows:

- **Regression**-In regression, the output attribute is continuous.
- **Classification**-In classification, the output attribute is discrete.

Regression produces continuous output variable or dependent variable[11]. It provides relationship between two or more input variables[12]. There are various types of Regression algorithms alike Simple Linear regression, Multiple linear regression, Logistic regression, Stepwise regression, Ridge regression, Lasso regression and ElasticNet regression.

In **Simple linear regression**, the relationship between the dependent and independent variable is ascertained. **Multiple linear regression** is an extension of linear regression. It consist of 'n' number of independent variables denoted as X1, X2, X3…Xn and a dependent variable Y. The dependent variable Y is predicted from one or more independent variables X1, X2, X3…Xn. In **Logistic regression**, dependent variable is a categorical value and not a continuous value[13]. It is of two kinds that can be either binomial or multinomial logistic regression. Binomial logistic regression have only two possible outcomes like yes or no, good or bad, true or false, 1 or 0, etc., Multinomial logistic regression have more than 2 possible categorical outcomes like poor or average or good or very good or excellent, very small or small or big or very big, etc., it's also referred as sigmoid function.

Stepwise regression[13] performs well when there are multiple independent variables (input variables). The main purpose of this technique is to maximize the prediction using minimum number of input variables or predictors. **Ridge regressions** are mostly used when there are predictors or independent variables that possess multi collinearity (highly correlated) and when there are more number of predictor variables[13]. **LASSO (Least Absolute Shrinkage and Selection Operator) Regression**[13] is similar to that of ridge regression but it uses L1 regularization technique to minimize error between actual and predicted value. Elasticnet regression is the combination of both Ridge and Lasso regression. It uses both L1 and L2 regularization technique. This type of regression is used when there are more number of features and when they suffer with multi collinearity.

Classification produces discrete output variable or dependent variable. Various types of Classification algorithms are Decision Tree Classifier, Support Vector Machines, Naïve Bayes classifier, K-Nearest Neighbors, Random Forest classifier, Neural Network, etc., **Decision Tree classifier** is a tree structure consisting of nodes and branches[14]. Internal nodes represent attributes. Branches represent decisions and leaf nodes are the outcomes that can be either categorical or continuous variable. Thus decision trees can be used for both categorical and regressive problems. **Support Vector Machine** is a supervised algorithm[15]. It can be used for both regression and classification problems but predominantly used for classification problems that is either two class classification and multi class classification.

Naïve Bayes classifier is considered as the fastest classification algorithm, when compared to other algorithms[16]. It is the best algorithm for large dataset. The basic principle of naïve bayes is each pair of features that are classified are independent to one another by applying bayes theorem as, P(A|B)=P(B|A)*P(A)/P(B) where A and B are events. **K Nearest Neighbor** is used for both classification and regression problems and mostly used for classification. Given a test data and to predict the class of test data, 'K' number of nearest training data are considered that are closer to the test data. The majority of the classes will be the predicted class. **Random Forest** is also used for both classification and regression algorithm based on a forest of trees[17]. The advantage of this algorithm is when the number of trees increases, the accuracy also increases. Decision tree(CART model) algorithm is the basis for random forest algorithm[18]. **Neural Network** uses Layered approach[19] which includes Input layer, Hidden layer and Output layer. Error is corrected by adjusting weights accordingly until error goes below threshold value[20]. In[21] estimation of effort using hybrid multilayer perceptron was carried by using complex non-linear input output relationship of a dataset.

Ensemble based approaches also became popular for effort estimations. Ensembling of machine learning algorithms[22] provide accurate results when compared with individual predictive machine learning algorithms. Hybrid of fuzzy based technique with function point analysis[23] and ensembling of fuzzy with analogy based estimation[24] paved attention in effort estimations.

Benefits and limitations of the aforementioned methods are discussed in Table 1.

Table 1. Estimation methods - Benefits and Drawbacks

S. No.	Estimation methods	Benefits	Drawbacks
1.1	**Expert Judgment**	Inexpensive method.	Estimation can be accurate or inaccurate depends upon the experience of the expert member.
	Delphi Technique	• It incurs less cost. • Experts can figure out the requirements for the future project from their past projects experiences.	• Error prone method. • This method often leads to overoptimistic estimation
	Work Breakdown Structure	• By this method, project risks can be identified during earlier stages. • Improves productivity.	Complex process. It uses step by step approach.
1.2	**Analogy based Estimation**	This approach is simple and fast.	This approach will not be always accurate in estimation.
1.3	**User Stories**	Story points are relative to the size of the project	User stories differs between between teams in a project.
1.4	**Function point Estimation**	This method can be applied during earlier stage of software development. This method is independent of any programming language.	It is a time consuming method and has less accuracy as it is based on judgmental approach.
1.5	**Use case point approach**	Use case point approaches are good measures for size prediction.	Use cases are large unit of work and estimations can be done only when all use cases are written.
1.6	**Supervised machine learning techniques- Regression algorithms**		
	Linear regression	It is the simplest method to find the relationship between two or more variables.	This method is able to give relationship between the independent and dependent variables that are linear.
	Logistic regression	Logistic regression is used when the independent variables (input variables) are categorical or/and continuous. It is an efficient and easy method to implement.	Using this method only linear problems can be solved and non-linear problems cannot be solved.
	Stepwise regression	Stepwise regression can handle large number of independent variables (predictor or input variables).	Using this method only linear problems can be solved and non-linear problems cannot be solved.
	Ridge regression	More number of independent variables can be used.	The drawback in this method is the model is considered to be complex that in turn leads to poor performance. This method generally produces high bias.
	Lasso regression	Lasso regression avoids overfitting and feature selection can be done.	This method is not often stable and selecting features among high correlated features is random.
	Elastic Net regression	Elastic Net is more preferred when compared to Ridge or Lasso regression.	Computational cost is high
1.7	**Classification algorithms**		
	Decision Tree Classifier	It is a simple method. It is a better method for estimating categorical data.	It provides less accuracy in prediction when compared to other machine learning algorithms.

Continued on next page

Table 1 continued

S. No.	Estimation methods	Benefits	Drawbacks
	Support Vector Machine	This method can also be applied to unstructured or semi structured data. They perform better even with many attributes.	It takes longer time for prediction in larger datasets.
	Naive Bayes Classifier	It is an easy method for implementation. This method produces better result if input variables are independent in nature.	This method always assumes that input variables are always independent, which cannot be always true.
	K Nearest Neighbor algorithm	Optimal for larger sample input.	It requires large storage requirement. It is sensitive to noise.
	Random Forest classifier	This method is user friendly and strong against overfitting. It can handle huge datasets.	It is time consuming and complex. It uses black box approach.
	Artificial Neural Network	• It can learn from previous data. • It is suitable for complex dataset. • It is suitable for linear and non-linear functions, thus produces high prediction of software effort.	Slow convergence speed and overfitting problem occurs.
1.8	**Ensemble approaches**	• They combine multiple models into aggregated better model.	Ensemble approaches are computationally expensive.

2 Materials and Methods

Machine learning algorithms considered for estimation are Multilinear Regression (MR), Random Forest(RF), Support Vector Machine(SVM), Decision Tree(DT), NeuralNet(NN), Ridge Regression(RR), Lasso Regression(LR) and ElasticNet Regression(ER).

2.1 Software effort estimation datasets

Datasets considered for Effort estimation are Desharnais, Maxwell, China and Albrecht[25]. Datasets repository, attributes and records are elaborated in Table 2 . Desharnais dataset consists of 81 records and after removing insignificant attributes, it consists of 12 attributes. Maxwell dataset consists of 62 records and 27 attributes. China dataset consists of 499 records and 16 attributes, after removing insignificant attributes. Albrecht dataset consists of 24 records and 8 attributes. The output attribute of the datasets-Desharnais, Maxwell and China are in the unit of hours and the output attribute of Albrecht dataset is in the unit of months.

Table 2. SEE – Dimensions of dataset

Dataset Name	Source Repository	No. of Records	No. of Attributes	Output Attribute-Effort (Unit)
Dataset1- Desharnais	GITHUB	81	12	Person-hours
Dataset2- Maxwell	PROMISE	62	27	Person-hours
Dataset3- China	PROMISE	499	16	Person-hours
Dataset4- Albrecht	PROMISE	24	8	Person-Months

2.2 Software Effort Estimation Evaluation Metrics

i. ***Mean Absolute Error (MAE)***

It is the average sum of absolute errors[26].

Prediction error=Actual value-Predicted value

Absolute error=|Prediction error|

MAE=Average of all absolute errors is given by Eq. (1)

$$\mathrm{MAE} = \textstyle\sum_{l=1}^{n} \left|\mathrm{Actualvalue_i} - \mathrm{Prediction\ value_j}\right| /\mathrm{n} \tag{1}$$

ii. Mean Squared Error (MSE)

It is the average of square of errors[27] in the data set and is given by Eq. (2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(X_{obs,i} - X_{model,i}\right)^2 \tag{2}$$

iii. Root Mean Square Error (RMSE)

It is the measure of standard deviation of predicted deviation [28] and is given by Eq. (3).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(X_{obs,i} - X_{model,i}\right)^2}{n}} \tag{3}$$

Where, Xobs-observed value, Xmodel-modelled value.

iv. R-Squared

It is also known as co-efficient of determination. Higher the value of R-squared, better is the model.

3 Results and Discussion

Machine learning algorithms considered for estimation are Multilinear Regression (MR), Random Forest (RF), Support Vector Machine (SVM), Decision Tree(DT), NeuralNet(NN), Ridge Regression(RR), Lasso Regression(LR) and ElasticNet Regression(ER). The software used for estimation is RStudio. Datasets used for effort estimation are Desharnais, Maxwell, China and Albrecht. Metrics used for evaluation are Mean Absolute Error (MAE), Mean Squared Error(MSE), RMSE(Root Mean Square Error(RMSE) and R-Squared. Lesser the values of MAE, MSE and RMSE, better is the model and if the R-squared value is higher, it is the better model.

Table 3 shows the performance measures of the machine learning algorithms against Desharnais dataset. Figure 1 1.1, 1.2, 1.3 and 1.4 displays barplot representation of Mean Absolute Error (MAE), Mean Squared Error (MSE), RMSE (Root Mean Square Error(RMSE) and R-Squared against machine learning algorithms for Desharnais dataset respectively. Inference from Table 3 is Support Vector Machine (SVM) produces lower values of Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE (Root Mean Square Error (RMSE) and SVM produces higher R-Squared value.

Table 3. Performance of algorithms using Desharnais dataset

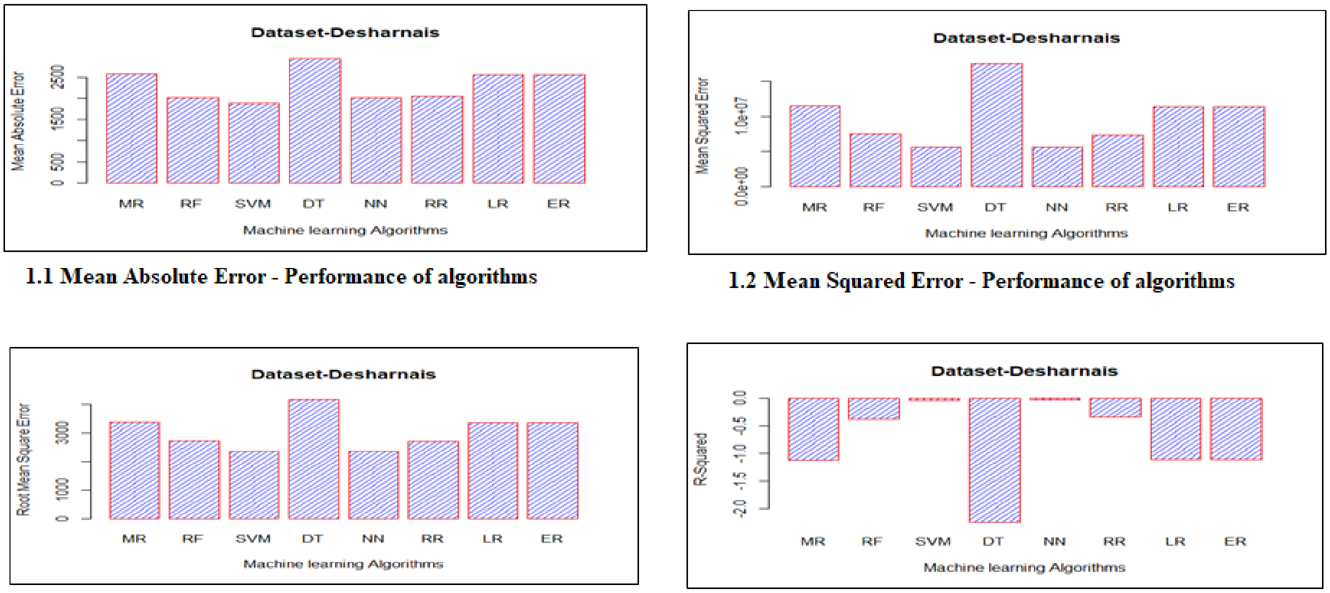
Dataset 1-Desharnais				
Algorithms	**MAE**	**MSE**	**RMSE**	**R-squared**
Multilinear Regression	2575.103	11499285	3391.06	-1.12404
Random Forest	2018.067	7465119	2732.237	-0.3788865
Support Vector Machines	**1888.018**	**5576003**	**2361.356**	**-0.02994683**
Decision Tree	2945.62	17502217	4183.565	-2.232845
Neuralnet	2024.692	5566429	2359.328	-0.0281783
Ridge Regression	2044.344	7283709	2698.835	-0.3453782
Lasso Regression	2562.611	11377363	3373.035	-1.101519
ElasticNet Regression	2562.606	11375202	3372.714	-1.10112

Table 4 shows the performance measures of the machine learning algorithmsagainst Maxwell dataset. Figure 2 2.1, 2.2, 2.3 and 2.4 displays barplot representation of Mean Absolute Error(MAE), Mean Squared Error(MSE), RMSE(Root Mean Square Error(RMSE) and R-Squared against machine learning algorithms for Maxwell dataset respectively. Inference from Table 4 is ElasticNet Regression (ER) produces lower values of Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE (Root Mean Square Error (RMSE) and ElasticNet Regression produces higher R-Squared value.

Table 4. Performance of algorithms using Maxwell dataset

Dataset2-Maxwell				
Algorithms	**MAE**	**MSE**	**RMSE**	**R-squared**
Multilinear Regression	6200.33	63759737	7984.969	0.1710158
Random Forest	3593.95	23507201	4848.423	0.6943667
Support Vector Machines	4276.059	38859106	6233.707	0.494766
Decision Tree	4328.487	44327679	6657.903	0.4236654
Neuralnet	6482.574	78104934	8837.7	-0.01549594
Ridge Regression	3895.332	22003377	4690.776	0.713919
Lasso Regression	3273.256	22017498	4692.281	0.7137354
ElasticNet Regression	**3113.22**	**20578229**	**4536.323**	**0.7324483**

Table 5 shows the performance measures of the machine learning algorithms against China dataset. Figure 3 3.1, 3.2, 3.3 and 3.4 displays barplot representation of Mean Absolute Error(MAE), Mean Squared Error(MSE), RMSE(Root Mean Square Error(RMSE) and R-Squared against machine learning algorithms for China dataset respectively. Inference from Table 5 is Lasso Regression (LR) produces lower values



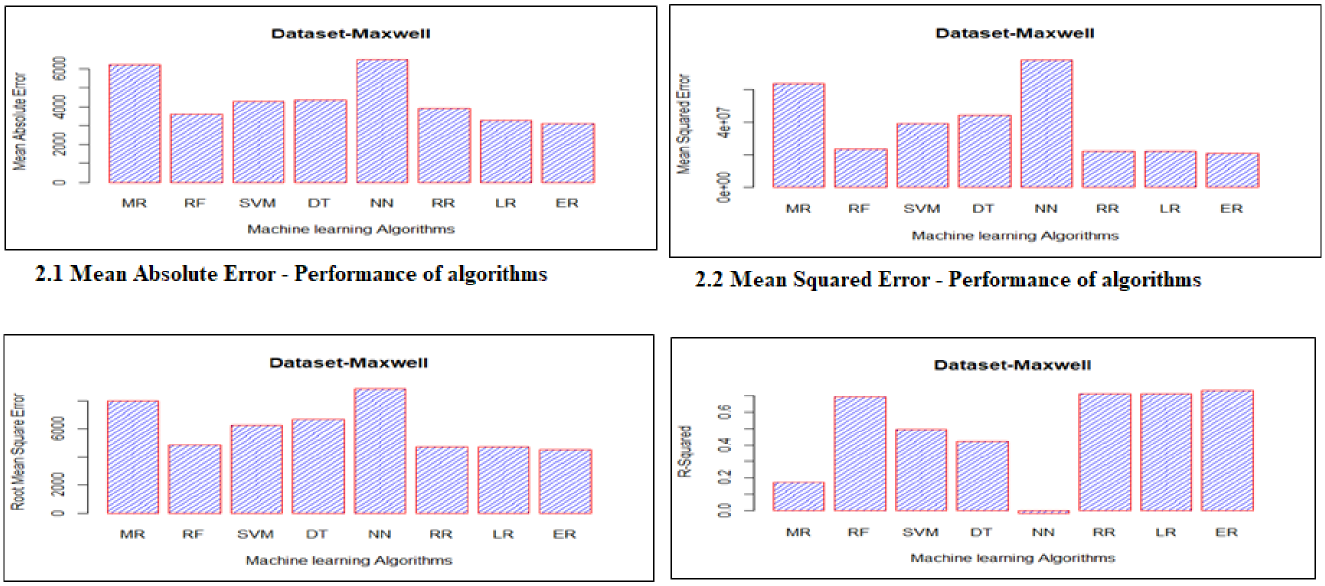
1.1 Mean Absolute Error - Performance of algorithms

1.2 Mean Squared Error - Performance of algorithms

1.3 Root Mean Square Error - Performance of algorithms

1.4 R-Squared - Performance of algorithms

Fig 1. Barplot representation -Desharnais dataset



2.1 Mean Absolute Error - Performance of algorithms

2.2 Mean Squared Error - Performance of algorithms

2.3 Root Mean Square Error - Performance of algorithms

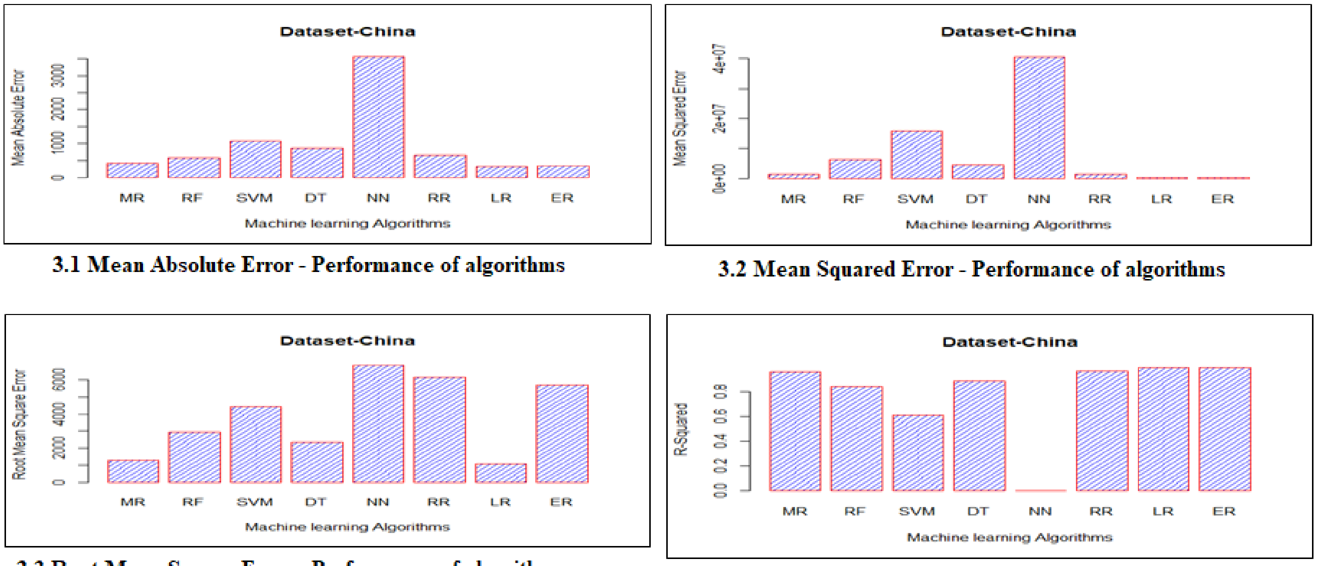
2.4 R-Squared - Performance of algorithms

Fig 2. Barplot representation -Maxwell Dataset

of Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE (Root Mean Square Error (RMSE) and Lasso Regression produces higher R-Squared value.

Table 5. Performance of algorithms using China dataset

Dataset3-China				
Algorithms	**MAE**	**MSE**	**RMSE**	**R-squared**
Multilinear Regression	427.1405	1448957	1203.726	0.9640306
Random Forest	574.369	6345708	2519.069	0.8424721
Support Vector Machines	1070.117	15775971	3971.898	0.6083722
Decision Tree	857.3184	4579875	2140.064	0.8863077
Neuralnet	3553.69	40430223	6358.476	-0.003652838
Ridge Regression	645.274	1420046	1191.657	0.9647483
Lasso Regression	**330.683**	**293412.4**	**541.6755**	**0.9927162**
ElasticNet Regression	345.1585	315410.5	561.6142	0.9921701



3.1 Mean Absolute Error - Performance of algorithms

3.2 Mean Squared Error - Performance of algorithms

3.3 Root Mean Square Error - Performance of algorithms

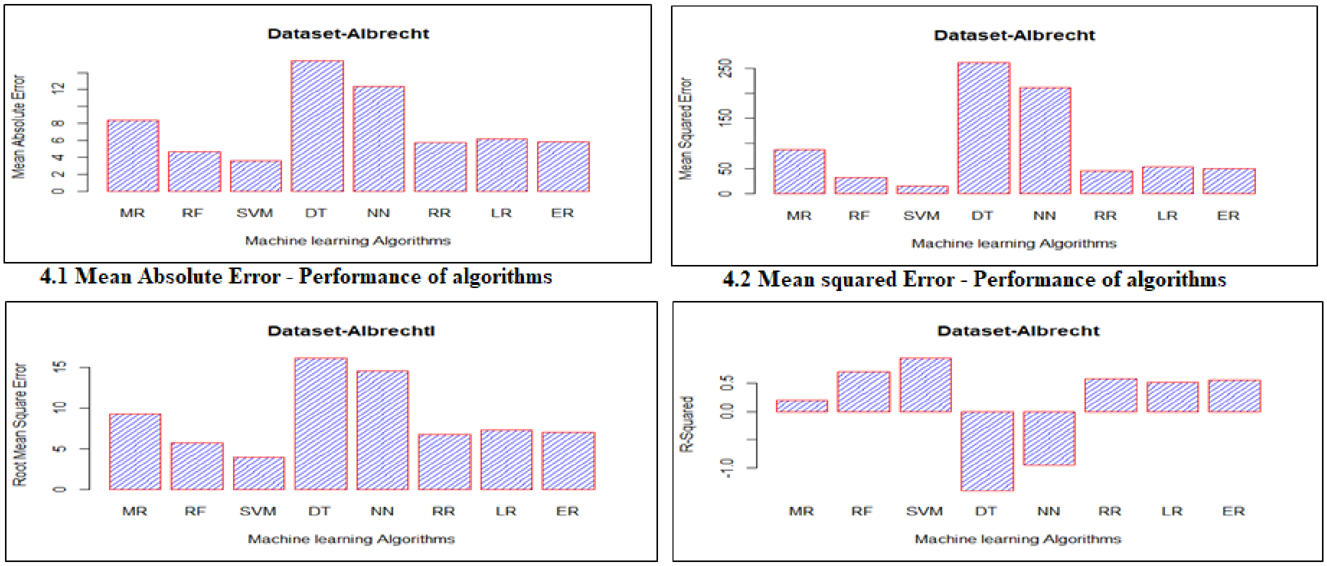
3.4 R-Squared - Performance of algorithms

Fig 3. Barplot representation -China dataset

Table 6 shows the performance measures of the machine learning algorithms against Albrecht dataset. Figure 4 4.1, 4.2, 4.3 and 4.4 displays barplot representation of Mean Absolute Error(MAE), Mean Squared Error(MSE), RMSE(Root Mean Square Error(RMSE) and R-Squared against machine learning algorithms for Albrecht dataset. Inference from Table 6 is Support Vector Machine (SVM) produces lower values of Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE (Root Mean Square Error (RMSE) and SVM produces higher R-Squared value.

Table 6. Performance of algorithms using Albrecht dataset

Dataset4-Albrecht				
Algorithms	**MAE**	**MSE**	**RMSE**	**R-squared**
Multilinear Regression	8.375672	87.21542	9.33892	0.1998402
Random Forest	4.626463	32.98749	5.743474	0.6973555
Support Vector Machines	**3.581059**	**15.27448**	**3.908258**	**0.940334**
Decision Tree	15.36563	261.2114	16.16204	-1.39649
Neuralnet	12.37725	212.2981	14.57045	-0.9477339
Ridge Regression	5.77861	45.74711	6.763661	0.5802921
Lasso Regression	6.214668	53.05237	7.283706	0.5132699
ElasticNet Regression	5.803795	49.15886	7.011338	0.5489909



4.1 Mean Absolute Error - Performance of algorithms

4.2 Mean squared Error - Performance of algorithms

4.3 Root Mean Sqaured Error - Performance of algorithms

4.4 R-Squared - Performance of algorithms

Fig 4. Barplot representation -Albrecht dataset

Based on the inference from Table 3 and Table 6, Support Vector Machine produces better results compared to other machine learning algorithm and Table 4 and Table 5 shows better results are produced by ElasticNet and Lasso Regression respectively. ElasticNet and Lasso regression is used to avoid over fitting problems, and they produce better results only when the independent attributes are more correlated with the output attribute, which is also in the case of the Maxwell and China dataset. SVM avoids over fitting problems, suitable for both structured & unstructured data and perform better with many input attributes. Finally, Support vector machine provides better performance compared with other machine learning algorithms and to the next level Lasso and ElasticNet regressions also provide better predictions.

4 Conclusion

This study compares various machine learning algorithms like Multilinear Regression, Ridge Regression, Lasso Regression, ElasticNet Regression, Random Forest, Support Vector Machine, Decision Tree and NeuralNet using Desharnais, Maxwell, China and Albrecht datasets. Software Effort Estimation (SEE) is predicting the amount of time taken in human hours or months for software development. It is difficult to forecast SEE during initial stages due to uncertainties. Estimation is the process that is used as input for pricing process, project planning, iteration planning, budget and investment analysis. Based on the comparative study of various machine learning algorithms, it is found that Support Vector Machine (SVM) outperforms other algorithms. Evaluation metrics considered are Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and R-Squared.

References

1) Ali A, Gravino C. A systematic literature review of software effort prediction using machine learning methods. *Journal of Software: Evolution and Process*. 2019;31(10). Available from: https://dx.doi.org/10.1002/smr.2211.
2) Pospieszny P, Czarnacka-Chrobot B, Kobylinski A. An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*. 2018;137:184–196. Available from: https://dx.doi.org/10.1016/j.jss.2017.11.066.
3) Jørgensen M. Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *Journal of Systems and Software*. 2016;117:274–281. Available from: https://doi.org/doi:10.1016/j.jss.2016.03.048.
4) Conoscenti M, Besner V, Vetrò A, Fernández DM. Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. *Journal of Systems and Software*. 2019;156:126–135. Available from: https://dx.doi.org/10.1016/j.jss.2019.06.075.
5) Idri A, azzahra Amazal F, Abran A. Analogy-based software development effort estimation: A systematic mapping and review. *Information and Software Technology*. 2015;58:206–230. Available from: https://dx.doi.org/10.1016/j.infsof.2014.07.013.
6) Benala TR, Mall R. DABE: Differential evolution in analogy-based software development effort estimation. *Swarm and Evolutionary Computation*. 2018;38:158–172. Available from: https://dx.doi.org/10.1016/j.swevo.2017.07.009.
7) Silhavy P, Silhavy R, Prokopova Z. Categorical Variable Segmentation Model for Software Development Effort Estimation. *IEEE Access*. 2019;7:9618–9626. Available from: https://dx.doi.org/10.1109/access.2019.2891878.
8) Martino SD, Ferrucci F, Gravino C, Sarro F. Web Effort Estimation: Function Point Analysis vs. COSMIC. *Information and Software Technology*. 2016;72:90–109. Available from: https://dx.doi.org/10.1016/j.infsof.2015.12.001.
9) Effendia A, Setiawan R, Rasjid ZE. Adjustment Factor for Use Case Point Software Effort Estimation (Study Case: Student Desk Portal). In: Budiharto W, et al., editors. The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society;vol. 157. 2019;p. 691–698. Available from: https://doi.org/10.1016/j.procs.2019.08.215.
10) Azzeh M, Nassif AB, Banitaan S. Comparative analysis of soft computing techniques for predicting software effort based use case points. *IET Software*. 2018;12(1):19–29. Available from: https://dx.doi.org/10.1049/iet-sen.2016.0322.
11) Oliveira ALI, Braga PL, Lima RMF, Cornélio ML. GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information and Software Technology*. 2010;52(11):1155–1166. Available from: https://dx.doi.org/10.1016/j.infsof.2010.05.009.
12) de Guevara FGL, Fernández-Diego M, Lokan C. The usage of ISBSG data fields in software effort estimation: A systematic mapping study. *Journal of Systems and Software*. 2016;113:188–215. Available from: https://dx.doi.org/10.1016/j.jss.2015.11.040.
13) Mensah S, Keung J, Bosu MF, Bennin KE. Duplex output software effort estimation model with self-guided interpretation. *Information and Software Technology*. 2018;94:1–13. Available from: https://dx.doi.org/10.1016/j.infsof.2017.09.010.
14) Wen J, Li S, Lin Z, Hu Y, Huang C. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*. 2012;54:41–59. Available from: https://dx.doi.org/10.1016/j.infsof.2011.09.002.
15) García-Floriano A, López-Martín C, Yáñez-Márquez C, Abran A. Support vector regression for predicting software enhancement effort. *Information and Software Technology*. 2018;97:99–109. Available from: https://dx.doi.org/10.1016/j.infsof.2018.01.003.
16) Zare F, Zare HK, Fallahnezhad MS. Software effort estimation based on the optimal Bayesian belief network. *Applied Soft Computing*. 2016;49:968–980. Available from: https://dx.doi.org/10.1016/j.asoc.2016.08.004.
17) abdelali Z, Mustapha H, Abdelwahed N. Investigating the use of random forest in software effort estimation. *Procedia Computer Science*. 2019;148:343–352. Available from: https://dx.doi.org/10.1016/j.procs.2019.01.042.
18) Satapathy SM, Rath SK, Acharya BP. Early stage software effort estimation using random forest technique based on use case points. *IET Software*. 2016;10(1):10–17. Available from: https://dx.doi.org/10.1049/iet-sen.2014.0122.
19) Nassif AB, Ho D, Capretz LF. Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*. 2013;86(1):144–160. Available from: https://dx.doi.org/10.1016/j.jss.2012.07.050.
20) Rijwani P, Jain S. Enhanced Software Effort Estimation Using Multi Layered Feed Forward Artificial Neural Network Technique. *Procedia Computer Science*. 2016;89:307–312. Available from: https://dx.doi.org/10.1016/j.procs.2016.06.073.
21) de A Araújo R, Oliveira ALI, Meira S. A class of hybrid multilayer perceptrons for software development effort estimation problems. *Expert Systems with Applications*. 2017;90:1–12. Available from: https://dx.doi.org/10.1016/j.eswa.2017.07.050.
22) Malgonde O, Chari K. An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*. 2018. Available from: https://doi.org/10.1007/s10664-018-9647-0.
23) Vijay JF. Enrichment of accurate software effort estimation using fuzzy-based function point analysis in business data analytics. *Neural Computing and Applications*. 2019;31(5):1633–1639. Available from: https://dx.doi.org/10.1007/s00521-018-3565-3.
24) Ezghari S, Zahi A. Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method. *Applied Soft Computing*. 2018;67:540–557. Available from: https://dx.doi.org/10.1016/j.asoc.2018.03.022.
25) Keung J, Kocaguneli E, Menzies T. Finding conclusion stability for selecting the best effort predictor in software effort estimation. *Automated Software Engineering*. 2013;20(4):543–567. Available from: https://dx.doi.org/10.1007/s10515-012-0108-5.
26) Minku LL, Yao X. Ensembles and locality: Insight on improving software effort estimation. *Information and Software Technology*. 2013;55:1512–1528. Available from: https://dx.doi.org/10.1016/j.infsof.2012.09.012.

27) Arbain SH, Ali NA, Mustaffa NH. Adoption of Machine Learning Techniques in Software Effort Estimation: An Overview. In: and others, editor. IOP Conference Series: Materials Science and Engineering;vol. 551. IOP Publishing. 2019. Available from: https://dx.doi.org/10.1088/1757-899x/551/1/012074.

28) Nassif AB, Azzeh M, Idri A, Abran A. Software Development Effort Estimation Using Regression Fuzzy Models. *Computational Intelligence and Neuroscience*. 2019;2019:1–17. Available from: https://dx.doi.org/10.1155/2019/8367214.