INTERNATIONAL CONFERENCE ON ADVANCES IN TECHNOLOGY AND COMPUTING (ICATC-2021)
Faculty of Computing and Technology (FCT), University of Kelaniya, Sri Lanka
18th December 2021

Index N0: CT-07-45

Software Test Effort Estimation Using Machine Learning Techniques

Miyushi Perera (1st Author)
Department of Computing & Information Systems
Wayamba University Of Sri Lanka
Kuliyapitiya, Sri Lanka
miyushiperera@gmail.com

VGTN Vidanagama (2nd Author)
Department of Computing & Information Systems
Wayamba University Of Sri Lanka
Kuliyapitiya, Sri Lanka
tharinda@wyb.ac.lk

***Abstract* — Software testing is the method of verifying a software product to recognize any errors, gaps, or missing requirements versus the exact requirements. Manual testing and automation testing are the two strategies of software testing. Testing requires a good amount of time and effort in the software development life cycle. The Software Development Life Cycle includes Planning, Designing, Developing, Testing, and Deploying. Software testing is acknowledged as an essential part of the software development life cycle since it concludes whether the software is ready to be delivered. This paper presents several machine learning techniques for test effort estimation. Support Vector Machine (SVM), KNearest Neighbour (KNN), and Linear regression are the techniques considered for the public dataset namely Desharnais.**

Keywords* — *software testing, machine learning, effort estimation

I. INTRODUCTION

The project manager usually faces the problem of estimating the effort needed to develop a project. This task depends on the software engineers in the team but it can be measured with various procedures mainly Expert estimation, Top-down estimation, Bottom-up estimation, and parametric estimation methods. Some of these procedures are straightforward to apply but require many additional data while the others are time-consuming and difficult to use [1].

Indeed we make fundamental judgments, there are various drawbacks in effort estimation like the software engineers involved in the project and their expertise. But, businesses require to perform effort estimation. And usually, all these estimators are done to estimate the effort for a brand-new project by project managers who use their experience-based judgments and knowledge from previous projects. [2]

It is a great challenge for a software company to develop a new software project of high quality within a predetermined budget and time. Many datasets have been used to estimate software effort that is publicly available in the PROMISE repository which is one of the most famous used repositories in the Software Engineering Community to estimate effort. Desharnais dataset[3] consists of 81 projects collected by J.M. Desharnais in the late 1980s from a Canadian software house. The original dataset consists of 12 attributes but in this study, the ProjectID attribute was omitted from the original dataset because it has no meaning to the study, the left are ten independent attributes and one dependent attribute (effort), all the values in this dataset are numeric but only one nominal attribute that is Language.

Table 1. List of variables in Desharnais dataset

Symbol	Name
TeamExp	Team experience – measured in years
ManagerExp	Manager experience – measured in years
YearEnd	Year project ended
Entities	The number of entities in the systems data model (function points)
Transactions	A count of basic logical transactions in the system (function points)
Length	Actual project schedule in months
PointsNonAjust	Transactions + Entities (function points)
PointsAdjust	Function points adjusted by the Adjustment factor = 0.65 + (0.01 * PointsNonAdjust)
Adjustment	Function point complexity adjustment factor (Total Processing Complexity)
Effort	Actual Effort is measured in person-hours (Dependent)
Language	Programming Language

In this study, we will discuss three machine learning techniques that could be used to predict the accuracy of effort estimation. KNN is one of the techniques used for classification problems, and it is one of the most simple classification techniques that should be the first option for a classification study. KNN works first by computing the distance between an instance with other instances and finding the k-nearest neighbor for that instance, then it estimates the effort.

SVM is a set of machine learning methods used in many areas, such as classification and regression. SVM classifier

separates the instances from two different classes by using a hyper-plane which tries to maximize the margin. This increases the generalization capability of the classifier [4].

Classification and Regression both are types of supervised learning algorithms. Both are working on labeled data set and used for predicting the output. Regression analysis is a predictive modeling technique that estimates the relationship between two or more variables. Regression analysis focuses on the relationship between a dependent (target) variable and an independent variable(s) (predictors). Linear regression is one of the most commonly used predictive modeling techniques. It is represented by,

$$Y = a + b_1 X_1 + b_2 X_2 + \ldots + b_n X_n \quad (1)$$

where a is the intercept, b's are the slopes of the line. This equation can be used to predict the value of a target variable based on a given predictor variable(s).

II. OBJECTIVES

Machine learning algorithms find natural patterns in data that generate insight and help make better decisions and predictions. They are used every day to perform critical decisions in medical diagnosis, stock trading, energy load forecasting, and more. This research focuses on selecting appropriate machine learning technique for software testing effort estimation and fitting a model.

III. METHODOLOGY

In this research, SVM, KNN, and Linear regression techniques were used, and the accuracy for each model was compared applying to the Desharnais data.

The final model was obtained for the highest accuracy. The independent variables were identified according to the high correlation coefficient values.

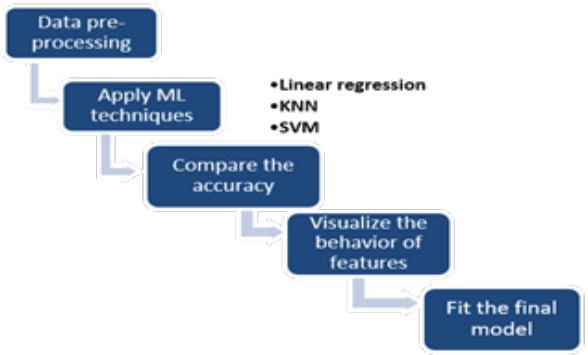

Fig. 1. How the estimation works

IV. RESULTS AND DISCUSSION

In this section, we will discuss the results of three machine learning techniques.

We get positive correlation coefficient values for the variables Length, Transactions, Effort, PointsNonAdjust, and PointsAdjust among 11 variables described in Table 1.

Table 2. Coefficient of determination (R^2) values

Model	Coefficient of determination (R^2)
Linear Regression	0.84
KNN	0.71
SVM	0.79

We get the highest accuracy for the Linear regression model. The below table represents the coefficient values of each independent variable for the linear regression model.

Table 3. Coefficient values of each variable

Variable	Coefficient
Length	253.35
Transactions	-16.09
Effort	-4.39
PointsNonAdjust	-20.49
PointsAdjust	39.29
Intercept	394.02

V. CONCLUSION

This study was done to evaluate the machine learning techniques by applying them to the Desharnais data set to predict software test effort for a new project.

The results describe the possibility of using the Linear regression method to predict the software effort with a coefficient of determination of 84%, the KNN method of 70%, and SVM with 79%.

We have seen from Table 1, that when we apply the Linear Regression model it has the highest accuracy for effort estimation. According to Equation 1, finally, we can fit the model for effort estimation where Y is Effort, X's are Length, Transactions, PointsNonAdjust & PointsAdjust, b's are the coefficient values and a is the Intercept listed in Table 3.

As mentioned earlier Desharnais data has only 81 project data. Therefore, a large data set can be used to train and test the model with better accuracy.

REFERENCES

[1] Omar Hidmi, a. B. (2017). 'Software Development Effort Estimation Using Ensemble Machine Learning'. *Int'l Journal of Computing, 4*, 143-147.
[2] Radlinski, L. &. (2010). 'predicting software development effort using machine learning techniques and local data'. *International Journal of Software, 2(2).*
[3] http://promise.site.uottawa.ca/SERepository/
[4] Sayyad Shirabad, J. T. (2005). *'The PROMISE Repository of Software Engineering Databases'*. Retrieved from http://promise.site.uottawa.ca/SERepository/