Machine Learning Based Effort Estimation of Web Applications Using ISBSG dataset

Manpreet Kaur* and Kanwalvir Singh Dhindsa

Abstract—**The web projects that are completed on time and within budget ascertain a commendable position in the rapidly growing economical web development market. Web Effort Estimation (WEE) estimates the time it will take to develop a web application in person-hours or months Expert Opinion algorithmic models, e.g., Constructive Cost Model (COCOMO), and machine learning are the primarily used effort estimating techniques. As current effort estimating techniques face many shortcomings, accurate effort prediction has become a challenging task. To improve prediction accuracy, this work proposes a hybrid approach based on Machine Learning. This approach is validated through an empirical evaluation of the International Software Benchmark Software Group, ISBSG R19 dataset. The ISBSG Release 19 dataset is first pre-processed using machine learning-based linear regression. Secondly, Support Vector Regression (SVR), Decision Tree Regression (DTR), Random Forest regression (RFR), and Ridge Regression (RR) techniques are employed to predict the web effort. The performance of the examined models is evaluated using two commonly used evaluation metrics, Mean Magnitude Relative Error (MMRE) and Prediction accuracy at level 25%, i.e., *Pred(25)*. Statistical significance of effort predicting model producing highest accuracy and lowest error rates is then verified using Mann-Whitney U test. The performance of the proposed models is also compared with the existing effort estimation models. The results show that the Ridge regression-based model produces exceptionally improved prediction accuracy for web projects in this work.**

Index Terms—**Web development, machine learning, support vector regression, ridge regression**

I. Introduction

Although calculating the effort necessary to develop a web application is difficult, precise estimations of the development effort are critical for the successful management of web-based projects. According to Retail sales reports from 2014 through 2023, there were $3.53 trillion in 2019, with e-commerce revenues expected to reach $6.54 trillion by 2022 [1]. With the growth of practicing web applications, reliable effort estimates are needed to ensure that web projects are completed and delivered on time while remaining within budget [2]. The necessity for employing techniques, standards, and best-practice guidelines to design applications that are delivered on time and under budget grows along with the demand for larger and more complex Web applications.

In the field of effort estimation for conventional software projects, several methods have been developed, tested, and successfully implemented. But developing Web applications is different from conventional software projects. Expert Judgement, Algorithmic Models, and Machine learning are the common methods used to predict the effort required to complete a web application project. Most of the Web developers use previous similar project experiences or expert judgment for effort estimates. Examples of algorithmic models are the Constructive Cost Model (COCOMO), and the Software Lifecycle Management Model (SLIM), whereas machine learning techniques are being used in aggregation or as replacements for algorithmic models. These methods include Neuro-fuzzy, Genetic Algorithms, Neural Networks, Fuzzy Logic and Regression trees, Case-based reasoning, Reasoning by Analogy, etc. [3].

Despite numerous effort prediction models in the literature, developing effort estimation models for web applications has become a challenging task. There is an urgent need to improve the prediction accuracy attained by existing models. One of the common confronts while constructing estimating models is incomplete data in historical datasets [4]. The absence of data values in various key project attributes consistently appears which might lead to inaccurate conclusions about the model's precision and analytical ability [5]. However, statisticians also fail to give accurate analytical conclusions due to omitted data from the dataset under statistical analysis [6]. The ISBSG R19 dataset description reveals that the various data fields have a substantial number of missing values. The performance of the effort estimation model may decline due to the lack of complete and consistent datasets. The International Software Benchmark Software Group (ISBSG) provides the dataset with heterogeneity [7] (combining data from several sources), irrelevant data fields, and missing values [8], which might lead to findings that are deceptive about the model's precision and propensity for prediction [4]. In the empirical software engineering literature, the imputation methods for missing values have garnered some practical attention [4, 5]. The default method suggested by existing researchers is to remove data fields with missing values, resulting in a relatively smaller dataset [9]. The quality of the prediction accuracy may enhance further if the threat of missing entries is filled by applying specific missing data management strategies [10]. Pre-processing of the dataset also requires attention to maximize the data retention and robustness of effort estimation models [11].

In this paper, the dataset used is an industrial dataset reported to the International Software Benchmarking Standards Group (ISBSG) [12]. In first step, the pre-processing of the dataset is performed through a sequence of steps, i.e., Data filtering, Data division, Data Normalization, and Data Scaling. The filtered dataset attained is then preserved to transform the partial dataset into a whole dataset.

Manuscript received September 19, 2022; revised December 12, 2022; accepted July 21, 2023.

Manpreet Kaur is with the I. K. Gujral Punjab Technical University, Punjab, India and Department of Computer Science, Hindu College, Punjab, India.

K. S. Dhindsa is with Department of Computer Science and Engineering, Baba Banda Singh Bahadur Engineering College, Punjab, India. E-mail: kdhindsa@gmail.com (K.S.D.)

*Correspondence: manprit.k.dhaliwal@gmail.com (M.K.)

8) ~~2021-2~~ 2022-23

Accepted (In Nov., '23)