AN EMPIRICAL STUDY ON REGRESSION ANALYSIS TO PREDICT SOFTWARE EFFORT BASED ON USE CASE POINTS

M. Pramod Kumar\(^1\) & Dr M Babu Reddy\(^2\)

Abstract: The effort estimation of software development is a process of predicting the most realistic effort (person-hours) required to maintain and development of software. This paper aims to provide best regression technique among linear regression, multiple linear regression, exponential regression, and power regression to predict software effort using use case points. The main contribution of this study is a comparative analysis of different regression techniques. To validate the accuracy of regression techniques we have used \(R^2\) value, MMRE and TSS. From this study we observed that linear regression, exponential regression, and power regression produced same results with some variance over multi linear regression.

Keywords: Regression Analysis, Software Effort

1. INTRODUCTION
Software development effort estimates are the basis for project planning, budgeting and bidding. These are demanding practices in the software industry, because poor estimation and planning often has affecting consequences [1]. The common argument on the project cost overruns is very large. Boraso reported that [2] 60% of large projects significantly overrun their estimates and 15% of the software projects are never completed due to the gross misestimating of development effort. Software product delivery on time, within budget, and to an agreed level of quality is a critical concern for software organizations. Accurate estimates are crucial for better planning, monitoring and control [3].

2.1 Linear Regression
Linear regression attempts to paradigmatic relationship between two variables by fitting a linear equation to observed data. One variable is examined to be an explanatory variable, and the other is examined to be a dependent variable. For suppose modeler might want to relate the weights of individuals to their heights using a linear regression model. Before attempting to fit a linear model to observed data, a modeler should first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other (for example, higher SAT scores do not cause higher college grades), but that there is some significant association between the two variables. A scatter plot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed explanatory and dependent variables then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables. A linear regression line has an equation of the form \(Y = a + b X\), where \(X\) is the explanatory variable and \(Y\) is the dependent variable. The slope of the line is \(b\), and \(a\) is the intercept (the value of \(y\) when \(x = 0\)).

2.2 Multiple Linear Regressions-
Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable \(x\) is associated with a value of the dependent variable \(y\). The population regression line for \(p\) explanatory variables \(x_1, x_2, \ldots, x_p\) is defined to be \(\mu_y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \ldots + \beta_px_p\). This line describes how the mean response \(\mu_y\) changes with the explanatory variables. The observed values for \(y\) vary about their means \(\mu_y\) and are assumed to have the same standard deviation. The fitted values \(\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_p\) estimate the parameters \(\beta_0, \beta_1, \ldots, \beta_p\) of the population regression line. Since the observed values for \(y\) vary about their means \(\mu_y\), the multiple regression model includes a term for this variation. In words, the model is expressed as DATA = FIT + RESIDUAL, where the "FIT" term represents the expression \(\hat{y}_0 + \hat{y}_1x_1 + \hat{y}_2x_2 + \ldots + \hat{y}px_p\). The "RESIDUAL" term represents the deviations of the observed values \(y\) from their means \(\mu_y\), which are normally distributed with mean 0 and variance \(\sigma^2\). The notation for the model deviations is \(\varepsilon\). Model for multiple linear regression, given \(n\) observations, is

\[ y_i = \beta_0 + \beta_1x_{i1} + \beta_2x_{i2} + \ldots + \beta_px_{ip} + \varepsilon_i \text{ for } i = 1, 2, \ldots, n \].

In the least-squares model, the best-fitting line for the observed data is

\(^1\)Research Scholar, Krishna University, Machilipatnam, Asst. Prof in SVEC, TP Gudem
\(^2\)HOD, CS Department, Krishna University, Machilipatnam
calculated by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values. The least-squares estimates \( b_0, b_1, \ldots, b_k \) are usually computed by statistical software.

2.3 Exponential Regression using a Linear Model-
Sometimes linear regression can be used with relationships which are not inherently linear, but can be made to be linear after a transformation. In exponential model shown in equation (1)

\[
Y = \alpha e^{\beta x}
\]  

Taking the natural log of both sides of the equation, we have the following equivalent equation (2)

\[
\ln Y = \ln \alpha + \beta x
\]  

This equation has the form of a linear regression model (where I have added an error term \( \varepsilon \))

\[
Y' = \alpha' + \beta x + \varepsilon
\]  

2.4 Power Regression using linear model-
Another non-linear regression model is the power regression model, which is based on the following equation

\[
y = ax^b
\]  

Taking the natural log of both sides of the equation, we have the following equivalent equation

\[
\ln y = \ln \alpha + \beta \ln(x)
\]  

This equation has the form of a linear regression model (where I have added an error term \( \varepsilon \))

\[
Y' = \alpha' + \beta x' + \varepsilon
\]  

Observation: A model of the form \( y = \beta \ln x + \varepsilon \) is referred to as a log-log regression model. Since if this equation holds, we have it follows that any such model can be expressed as a power regression model of form \( y = ax^b \) by setting \( \alpha = e^{\alpha'} \).

3. RELATED WORK
Regression analysis generates equations to predict effort for software development using methods like fuzzy logic. Several algorithmic models are available in the literature. General form of linear regression equation is proposed by Kok et al. [8], while a group of non-linear regression equations are presented by Boehm [13] in COCOMO 29 and COCOMO II [9]. An Albus multilayer perceptron is used to predict software effort in [14] for Boehm’s COCOMO dataset. Linear regression is compared with NN based approach for the COCOMO dataset. Both the approaches do not provide better results. In Briand and Wieczorek [10], a relationship between effort and one or more characteristic of a project is presented. The software size is taken as the cost determinant. To improve the accuracy of effort estimation in the single regression model, several data partitioning based studies on deriving multiple regression models are developed by Cuadrado-Gallego et al., Cuadrado-Gallego et al. [4]. These models overcome the common shortcomings like poor model fitting and low accuracy of effort estimation in datasets of heterogeneous projects. An approach for generating multiple regression models by clustering using Expectation-Maximization (EM) algorithm is proposed by Cuadrado - Gallego et al. [4, 5]. Based on the experimental results validated with the ISBSG (Release 8) dataset, the accuracy of effort estimation by the multiple regression models is better when compared to the single model. Parametric software cost estimation models based on the historical software projects databases involve mathematical relations and are useful in estimating the effort and time required to develop a software product. Heterogeneous projects are considered and a single parametric model for a range of diverging project sizes and characteristics is not available. Segmented models are used in which several models are combined into one which gives the estimates depending on the concrete characteristic of the inputs. A given project can belong to several segments with different degrees of fuzziness. An approach that generates multi-standard LSR models based on fuzzy clustering is proposed by Aroba et al. [6]. The above mentioned problems are addressed using a segmented model based on fuzzy clusters of the project space. Fuzzy clustering aids in obtaining different mathematical models for each cluster and also allow the items of a project to contribute to more than one cluster, while preserving constant time execution of the estimation process. Fuzzy clustering generates different LSR models for each cluster. The data points are contained in more than one cluster with different degrees of fuzziness. The final effort estimate is derived from the membership values of each data point used as a weight for each model. The proposed approach is validated for the ISBSG (Release 8) dataset, and the results are found to be better than the single model. The number of clusters is increased to find better estimation results López-Martín [6] has compared Fuzzy Logic Models (FLM) with Linear Regression Model (LRM). The evaluation criterion is based on the Magnitude of Error Relative to the estimate (MER) and Mean of MER (MMER). From the programs developed, three FLMs were generated to estimate the effort. FLM and LRM offer similar performance. Least Squares Regression (LSR) is the most commonly used SEE method. LSR model is affected by the data distribution. For scattered dataset, the model usually shows poor performance. Data partitioning-based approaches are considered to be better when compared to the clustering-based approaches. Seo et al. [7], a new data partitioning-based approach is proposed to achieve more authentic and stable effort estimation using Least Squares Regression (LSR). This approach provides an effort prediction interval that is useful in
determining the uncertainty of the estimates. The proposed approach is compared with the basic LSR approach and clustering-based approaches based on industrial datasets.

4. EXPERIMENT RESULTS

We applied all regression techniques on the data set taken from [12]. The data set contains 214 attributes and Table 1 shows some sample dataset and its attributes. The statistical measures of dataset are given in Table 2. From the Table 2 it can be observed that kurtosis and skewness are very high and data are not properly distributed as shown in Fig 1 and Fig 2. So to apply regression techniques, we have applied logarithmic transformation. From Fig 3 and Fig 4 it can be observed that the data is properly distributed.

Table 1 Sample data set with six attributes

<table>
<thead>
<tr>
<th>Project number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity value</td>
<td>1</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td>0.85</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.7</td>
</tr>
<tr>
<td>Productivity value</td>
<td>1.3</td>
<td>1.15</td>
<td>1</td>
<td>1.15</td>
<td>1.15</td>
<td>1</td>
<td>1</td>
<td>1.15</td>
<td></td>
</tr>
<tr>
<td>Requirement stability value</td>
<td>1</td>
<td>1</td>
<td>1.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Size</td>
<td>13.5</td>
<td>18</td>
<td>20.5</td>
<td>28</td>
<td>39</td>
<td>41.5</td>
<td>47</td>
<td>47.5</td>
<td>51</td>
</tr>
<tr>
<td>effort</td>
<td>122</td>
<td>296</td>
<td>360</td>
<td>170</td>
<td>507</td>
<td>634</td>
<td>752</td>
<td>751</td>
<td>244</td>
</tr>
</tbody>
</table>

Table 2 Statistical analysis of data set

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>214</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.97</td>
</tr>
<tr>
<td>Geometric Mean</td>
<td>4.83</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.7047</td>
</tr>
<tr>
<td>Maximum</td>
<td>8.2965</td>
</tr>
<tr>
<td>standard deviation</td>
<td>0.322</td>
</tr>
<tr>
<td>Mean Absolute Deviation</td>
<td>1.1771</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.9457</td>
</tr>
<tr>
<td>Variance</td>
<td>1.177</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.38</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>0.1703</td>
</tr>
</tbody>
</table>

Fig 1 Histogram of independent variable

Fig 2 Histogram of dependent variable

Fig 3 Histogram of independent variable ln(size)

Fig 4 Histogram of dependent variable ln(Effort)
4.1 Performance evaluation

We evaluate the performance based on Magnitudes of relative error (MRE), mean magnitude of relative error (MMRE) and $R^2$. These are the standard evaluation measures to predict the size and effort estimation. We also used total sum of squares (TSS) which is the metric to evaluate prediction models and median of errors. The following equations have been used for obtaining performance metrics.

\[
MMRE = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{|\text{actualeffort}_{ij} - \text{estimatedeffort}_{ij}|}{\text{actualeffort}_{ij}}
\]

\[
MRE = \frac{1}{N} \sum_{i=1}^{N} \frac{|\text{actualeffort}_{ij} - \text{estimatedeffort}_{ij}|}{\text{actualeffort}_{ij}}
\]

\[
TSS = \sum_{i=0}^{N}(\text{Real}_{p20} - (\text{predicted size}))
\]

5. SELECTING THE BEST REGRESSION MODELS

The regression equations shown in the Table 3 are derived from MINITAB. We compared linear multi linear, linear power and exponential regression on UCP data.

Table 3 Regression models

<table>
<thead>
<tr>
<th>Type of regression model</th>
<th>Linear regression(LR)</th>
<th>Multi Linear Regression(MLR)</th>
<th>Power Regression(PR)</th>
<th>Exponential Regression(ER)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.15* ln (size)+2.391</td>
<td>(0.642<em>ln (size))-0.61</em>requirement stability+1.64* productivity + 1.22* Complexity-0.322</td>
<td>Ln(size)$^{0.61}*3.07$</td>
<td>($e^{2.12+(ln(size))^4.15}$)</td>
</tr>
</tbody>
</table>

Table 4 Statistical measures of regression models

<table>
<thead>
<tr>
<th>Type of regression model</th>
<th>Residual Sum of Squares</th>
<th>Coefficient of Determination($R^2$)</th>
<th>MEAN</th>
<th>STANDARD DEVIATION</th>
<th>MMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression(LR)</td>
<td>15.0637</td>
<td>91.77</td>
<td>0.045</td>
<td>0.043</td>
<td>0.05</td>
</tr>
<tr>
<td>Multi Linear Regression(MLR)</td>
<td>6.63652128</td>
<td>96.374</td>
<td>0.382</td>
<td>0.05</td>
<td>0.38</td>
</tr>
<tr>
<td>Power regression(PR)</td>
<td>43.8872</td>
<td>86.486</td>
<td>0.048</td>
<td>0.043</td>
<td>0.047</td>
</tr>
<tr>
<td>Exponential Regression(ER)</td>
<td>39.4259</td>
<td>88.4578</td>
<td>0.047</td>
<td>0.044</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Table 4 shows that the statistical measures of all regression models. Form the table 4 it can be observed that multi linear regression model has highest value of $R^2$ and MMRE. But to select best regression models, we consider one that should have highest $R^2$ and low MMRE values. From our observations linear, power, and exponential regression models have highest $R^2$ and low MMRE values than multi linear regression models. It may be considered that power regression has satisfactory results among all regression models. Fig 5 and Fig 6 show the comparisons of TSS and MMRE values of all regression models respectively.

Table 5 Total Sum of Squares (TSS) of regression models

<table>
<thead>
<tr>
<th>TSS of LR</th>
<th>TSS of MLR</th>
<th>TSS of PR</th>
<th>TSS of ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.23562</td>
<td>10.14658</td>
<td>0.228203</td>
<td>0.242802</td>
</tr>
</tbody>
</table>

Fig 5 TSS Comparison of regression models
6. CONCLUSION
Software size is one of the important factors that affect the effort estimation. In this paper we studied various regression techniques using use case points to predict the software effort estimation. We applied linear regression, multi linear regression, power regression, and exponential regression on the same data set. We used MMRE, TSS and $R^2$ performance measures to evaluate the accuracy of all techniques. Results have shown that the linear regression, power regression and exponential regression are accurate with little variance than multi linear regression on the test data set.

7. REFERENCES