Abstract—Effort Estimation is an important aspect for software developers. There are various computer-based techniques to estimate effort. kNN based approach could serve as an economical, automatic tool to generate ranking of software by formulating the relationship based on its training. The accuracy of the proposed system is 95.2381%. It means it is very close to the actual values.

Keywords—Software Effort Estimation, Nearest Neighbour Approach, kNN, k-Nearest Neighbour, Effort Estimation, Estimation Techniques.

I. INTRODUCTION

The basic software project effort and schedule estimation procedure was presented in [7]. Accurate estimate means better planning and efficient use of project resources such as cost, duration and effort requirements for software projects especially space and military projects [3], [8]. Efficient software project estimation is one of the most demanding tasks in software development. Problem of inaccurate estimate for projects and in many cases inability to set the correct release day for their software correctly lead to inefficient use of project resources. Unfortunately, software industry suffers the problem of incorrect estimate for projects and in many cases inability to set the correct release day for their software correctly. This leads to many losses in their market, e.g. risk due to low quality of the deliverables and penalties for missing the deadlines. Normally, estimation is performed using only human expertise [5], [9], but recently attention has turned to a variety of computer-based learning techniques. Numerous models were explored to provide better effort estimation [1], [10], [11], [12]. In [4], [9] authors provided a survey on the effort and cost estimation models.

II. RESEARCH METHODOLOGY

I. Data Collection

First, Survey of the existing Models of Effort Estimation is to be performed and Secondly, Historical Data being used by various existing models for the cost estimation is collected.

II. Evaluation of earlier Popular Models:

The following modelling approaches are mostly used for effort dataset:

• Halstead Model
• Walston-Felix Model
• Bailey-Basili Model
• Doty Model

The equations of the models are given below:

Halstead E = 5.2(KLOC)^1.50
Walston-Felix E = 0.7(KLOC)^0.91
Bailey-Basili E = 5.5 + 0.73(KLOC)^1.16

Doty (for KLOC > 9) E = 5.288(KLOC)^1.047

Here, KLOC is the kilo lines of code.

So, in the study the results of these approaches are also evaluated.

III. Evaluating the KNN based Approach

The comparison of the results is made on the basis of:

• Mean Magnitude of Relative Error (MMRE)
• Root Mean Square Error (RMSE)

RMSE [6] is frequently used measure of differences between values predicted by a model or estimator and the values actually observed from the thing being modelled or estimated. It is just the square root of the mean square error as shown in equation given below:

\[ \sqrt{\frac{(a_1-c_1)^2 + (a_2-c_2)^2 + \ldots + (a_n-c_n)^2}{n}} \]  

The mean-squared error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the squared differences between each computed value and its corresponding correct value. The root mean-squared error is simply the square root of the mean-squared-error.

The value of an effort predictor can be reported many ways including MMRE. MMRE [6] is computed from the relative error, or RE, which is the relative size of the difference between the actual and estimated value:

\[ RE_i = \frac{\text{estimate}_i - \text{actual}_i}{\text{actual}_i} \]  

Given a data set of size "D", a "Training set of size 
"(X=|Train|) <= D", and a "test" set of size "T=D-|Train|", then the mean magnitude of the relative error, or MMRE, is the percentage of the absolute values of the relative errors, averaged over the "T" items in the "Test" set; i.e.

\[ MMRE = \frac{100}{T} \sum \left| \frac{\text{estimate}_i - \text{actual}_i}{\text{actual}_i} \right| \]  

Amanjot Singh Klair and Raminder Preet Kaur are M.Tech. CSE (Student), IIT(Poewal), Punjab Technical University, Jalandhar.
The mean magnitude of relative error (MMRE) can also be written as:

\[ MMRE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \] (4)

Where, \( y_i \) represents the \( i \)th value of the effort and \( \hat{y}_i \) is the estimated effort[6].

\[ RMSSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \] (5)

The RMSE and RMSSE are the same. So, in our study we will take MMRE, RMSE as the performance criteria.

III. RESULTS AND ANALYSIS

Historical NASA’s COCOMO81 Dataset [2] for the effort estimation is collected and the data is polished so that the same data can be used for the modelling in MATLAB 7.2 environment. The 16 attributes are analysts capability, programmers capability, application experience, modern programming practices, use of software tools, virtual machine experience, language experience, schedule constraint, main memory constraint, data base size, time constraint for CPU, turnaround time, machine volatility, process complexity, required software reliability and lines of source code.

The results of Halstead Model, Walston-Felix Model, Bailey-Basili Model, Doty Model and kNN model are shown in Table 1. The performance criteria taken are MMRE, RMSE. The kNN Model shows the MMRE and RMSE values as 0.1111 and 0.5774 respectively as evidenced by the output screenshot in Figure 1. The Halstead model shows the highest MMRE value 887.78.

<table>
<thead>
<tr>
<th>Performance Criteria</th>
<th>Models Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE</td>
<td>Halstead Model</td>
</tr>
<tr>
<td></td>
<td>887.78</td>
</tr>
<tr>
<td>RMSE</td>
<td>26575</td>
</tr>
</tbody>
</table>

The accuracy of the proposed system is 95.2381%.

IV. CONCLUSION

In this study kNN based Model is experimented to estimate the software effort for projects. The performances of the developed models were tested on NASA software project data presented in [5] and results are compared with the Halstead, Walston-Felix, Bailey-Basili and Doty models mentioned in the literature. The results shows that the kNN based Model have the lowest MMRE and RMSE values i.e. 0.1111 and 0.5774 respectively. When the accuracy of the system is calculated then it showed 95.2381% of accuracy. This means that the system is showing very low error and high value of accuracy that is desirable from the prediction system. So, kNN model based system is recommended to estimate the software effort for projects.

V. FUTURE SCOPE

The proposed models provided good estimation capability compared to traditional model structures. In the future, one can follow the flowing directions:

• Explore the use of Neuro-Fuzzy (NF) technique to build suitable model structure for the software effort estimation by optimization of Neuro-Fuzzy Systems with Evolutionary Techniques.
• Further advanced Machine learning techniques can be used
• Rough Set theory and Taguchi Analysis can be used to find the impact of the different attributes towards Effort Prediction.
• Simulated Annealing Technique can be used to improve the performance of the NF systems.
• Accuracy of the Rule Generation for the Fuzzy Inference System can further be improved with decision tree techniques.

REFERENCES


