Journal Of Harmonized Research (JOHR)

Journal Of Harmonized Research in Engineering
8(2), 2020, 41-49

ISSN 2347 – 7393

Original Research Article

AN ANALYSIS ON MACHINE LEARNING BASED PREDICTION FOR SUCCESS RATE OF PROJECT

Vikas

Rsearch Scholar, Sri Satya Sai University of Technology and Medical Sciences, Sehore

Abstract: The prediction of project success the exhaustive goal of various Industries. Whereas, it becomes more critical to execute the project successfully. To predict the project success various data mining and machine learning techniques such as k-nearest neighbor, SVM classier, logistic regression has been developed but, in this work, we use random forest classifier for the prediction of project success with reduced cost and schedule. The random forest classifier selects the dataset randomly from the available dataset and the generate the decision tree of the selected dataset and then apply the voting on the prediction results and whose score and accuracy will be maximum that will indicates the success of project. For the sample dataset we use online resource of kaggle and the experimental results is generated from the widely used machine learning programming language Python which helps in the analysis of the proposed methodology. The performance of proposed methodology is measured using the parameters such as Score, accuracy, precision, recall value, F1 score, mean absolute error and mean square error. The comparative analysis of the proposed methodology is done among the existing approach K-neighbor and Logistic regression. The score and accuracy value of our proposed methodology is 78% while other is less. Similarly, the F1 score, precision and recall value of proposed methodology is 43%, 50% and 42% while the K –neighbor and logistic regression is comparatively very less. Similarly, the comparative analysis of proposed and existing approach is done using mean absolute error and mean square error and the value is 22% and 11% which is very less. These results of proposed methodology improve the success rate of software project

Keywords: *Random Forest, Logistic Regression, Prediction, Software Project, MAE, MSE, Precision,*

For Correspondence:
Ikchiudhary421@gmail.com.
Received on: March 2020
Accepted after revision: April 2020
Downloaded from: www.johronline.com

Introduction: In the software industry, software project success requires the fulfilment of certain expectations held by participants, who include owners, planners, designers, architects, contractors, and operators. Once a software project has been bid, the prime contract is typically subdivided into multiple subcontracts.

Large numbers of participants are, therefore, involved in project planning and implementation phases. The only way to ensure expectations are met is by conducting a comprehensive analysis of participants [1]. Key measurements of project success in the software industry include cost, schedule, performance and safety. Several researches [2] were addressed to predict project success. Accomplishment of some software organization is contingent on whole customer satisfaction which in turn is contingent on the growth of quality software. The Software Engineering approach enables the manufacture of quality software. One of the significant characteristics of eminence software is that it should be defect-less. The aim of imperfection recognition and anticipation is to deliver quality software that will diminish the cost and time intricate in fixing a deficiency, upsurge productivity and enable to accomplish total customer satisfaction. The need for development of excellence software is emphasized by providing numerous definitions of flaws and the numerous techniques of preventing these imperfections while developing software. This chapter also gives the details concerning the use of machine learning techniques in deficiency prediction, concrete challenges and exploration-oriented issues in order to produce high eminence software. However, previous research in predicting project success either adopted fixed factors at various points in time or used an inference method. These approaches presented several important difficulties which rendered them inadequate for general application. The software industry is replete with myriad uncertainties that make management exceedingly complex. Factors for success, therefore, vary from project to project. Although human experts can often achieve a satisfactory project outcome, shortfalls nearly always occur due to managers failing to take all relevant factors into consideration and lacking access to all relevant information. In general, shortcomings in a currently used method suggest a need and opportunity for improvement. Such is true in the software industry as well. Present shortcomings in decision-making methods can be described as follows: (1) Attrition of Human Expertise, (2) Trial-and-Error Approach, and (3) Subjective Assessment.Various scientific and engineering specializations have been paying increasing attention in recent years to the fusing of different artificial intelligence (AI) paradigms to achieve greater efficacy in results. A number of studies have demonstrated that performances achieved by fusing different AI techniques are better than those achieved by employing a single conventional technique [3]. Fast messy genetic algorithms (fmGA) and the support vector machine (SVM) are two tools that have been applied successfully to solve various software management problems. An appreciation of critical factors is crucial to assess the requirements of project success and to achieve project objectives successfully. Statistical methods represent a basic approach to identify significant factors from historical data or questionnaire results. However, the dynamic nature of critical factors means that changes in project conditions must be monitored continuously. In this dissertation, we applied random forest classifier of machine learning approach for the success prediction of software project which can effectively and efficiently predict the success by using the performance measuring parameter such as score, precision, recall value, accuracy, mean absolute error and mean square error etc. Machine learning is a paradigm that may refer to learning from past experience (which in this case is previous data) to improve future performance. The sole focus of this field is automatic learning methods. Learning refers to modification or improvement of algorithm based on past “experiences” automatically without any external assistance from human.

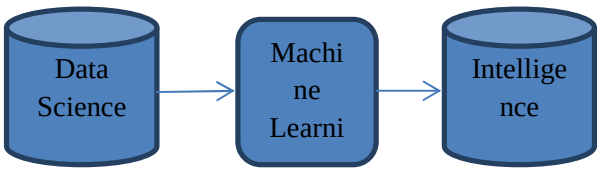


Fig. 1: Machine Learning states

Related Work: ***N. Kalaivani and R. Beena (2018),*** improved the quality of software, data mining techniques have been applied to build predictions regarding the failure of software components by exploiting past data of software components and their defects. [4] This paper reviewed the state of art in the field of software defect management and prediction, and offered data mining techniques in brief. ***Jaechang Name et al. (2017),*** applied Heterogeneous Defect Prediction [HDP] to predict defects in with-in and across projects with different datasets.[5] Metric selection, metrics matching and building a prediction model are the 3 methods used in this work. In this research they used various datasets from NASA, PROMISE, AEEEM, MORPH and SOFTLAB. Source and target datasets are used with different metric sets. For selecting metrics feature selection techniques such as gain ratio, chi-square, relief-F and significance attribute selection are applied to the source. To match source and target metrics various analyzers like Percentile based Matching (PAnalyzer), Kolmogorov – Smirnov test based matching (KSAnalyzer), Spearman's Correlation based Matching (SCOAnalyzer) are used. Cutoff threshold value is applied to all pair scores and poorly matched metrics are removed by comparison. Area Under the Receiver Operator Characteristic Curve [AUC] measure is used to compare the performance between different models. HDP is compared with 3 baselines – WPDP, CPDP-CM, CPDP-IFS by applying win/loss/tie evaluation. The experiments are repeated for 1000 times and Wilcoxon signed rank test ($P<0.05$) is applied for all AUC values and baselines. Performance is measured by counting the total number of win/loss/tie. When a cutoff threshold value gets increased in PAnalyzer and KSAnalyzer, the results (win) also gets increased. Logistic Regression (LR) model works better when there is a linear relationship between a predictor and bug-prone.***Logan Perreault et al. (2017),*** applied classification algorithm such as naïve Bayes, neural networks, support vector machine, linear regression, K-nearest neighbor to detect and predict defects. The authors used NASA and tera PROMISE datasets. [6] To measure the performance they used accuracy and f1 measure with clearly well-defined metrics such as McCabe Metrics and Halstead Metrics. 10-fold cross validation is used in which 90% of data are used for training and 10% of data are used for testing. ANOVA and tukey test were done for 5 dataset and 5 response variables.0.05 is set as significance level for PC1, PC2, PC4 and PC5 dataset and 0.1 as PC3 dataset. Weka tool is used for implementation. Implementations of these 5 algorithms are available on Github repository. Finally the authors conclude that all datasets are similar and they are written in C or C++ and in future the work can be extended by selecting the datasets that are written in Java and instead of using weka tool for implementation some other tool can also be used .***Yongli et al. (2017),*** PROMISE datasets and Confusion matrix are used to evaluate the performance measure. Due to imbalanced dataset probability of detection [pd], probability of false alarm [pf] and AUC are also applied to measure the performance.[7] Therefore, the authors conclude from the experiments, Naïve Bayes [NB] algorithm performs better than Support Vector Machine. For smaller projects Target – Project Data Guided Filter [TGF] is used and for larger projects Hierarchical Select Based Filter [HSBF] is used for data selection from multi-source projects.***Shamsul Huda et al.(2017) ,*** [8] studied that developing a defect prediction model by using more number of metrics is a tedious process. So that a subset of metrics can be determined and selected. In this research two novel hybrid SDP models such as wrappers and filters are used for identifying the metrics. These two models combine the training of metric selection and fault prediction as a single process. In this research different datasets and classification algorithms such as Support Vector Machine [SVM] and artificial neural network are used. Performance was measured by using AUC and MEWMA (Multivariate Exponentially Weighted Moving Average), implementation was done by using liblinear tool and mine tool. ***Cong Pan et al.(2019),*** An Improved CNN Model for Within-Project Software Defect

Prediction.[9] To improve software reliability, software defect prediction is used to find software bugs and prioritize testing efforts. Recently, some researchers introduced deep learning models, such as the deep belief network (DBN) and the state-of-the-art convolutional neural network (CNN), and used automatically generated features extracted from abstract syntax trees (ASTs) and deep learning models to improve defect prediction performance. However, the research on the CNN model failed to reveal clear conclusions due to its limited dataset size, insufficiently repeated experiments, and outdated baseline selection. To solve these problems, we built the PROMISE Source Code (PSC) dataset to enlarge the original dataset in the CNN research, which we named the Simplified PROMISE Source Code (SPSC) dataset. Then, we proposed an improved CNN model for within-project defect prediction (WPDP) and compared our results to existing CNN results and an empirical study. Our experiment was based on a 30-repetition holdout validation and a 10 * 10 cross-validation. Experimental results showed that our improved CNN model was comparable to the existing CNN model, and it outperformed the state-of-the-art machine learning models significantly for WPDP. Furthermore, we defined hyper parameter instability and examined the threat and opportunity it presents for deep learning models on defect prediction. ***K. Hiba Sadia et al.(2019),*** focused on data preprocessing of the raw dataset. Secondly, after pre-processing the data, we will review the use of random forest, support vector machine on the dataset and the outcomes it generates.[10] In addition, the proposed paper examines the use of the prediction system in real-world settings and issues associated with the accuracy of the overall values given. The paper also presents a machine-learning model to predict the longevity of stock in a competitive market. The successful prediction of the stock will be a great asset for the stock market institutions and will provide real-life solutions to the problems that stock investors face.

Methodology: In this section of the paper, we are describing the proposed methodology used for the prediction of project success. In this we use the random forest classifier for the prediction and its comparison is done with the other classifier. For the classification of dataset, we use online dataset from kaggle and these datasets is compared with k-neighbor and logistic regression classifier. The random forest classifier selects the sample dataset randomly from the given dataset then construct the decision tree and apply the prediction on each decision tree. Then after generating the prediction result and apply the vote process and the prediction result which gets more vote that sample is chosen as classifier. The brief description of the proposed methodology and steps is discussed below:

Random Forest Classifier: Random forests are a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Random forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Methodology: As the Dataset we are using is taken from Kaggle (The online sources of Datasets). In Starting the columns in the dataset (Raw Dataset) are

1. Project Id
2. Name
3. Goal
4. Keywords
5. Country
6. Currency
7. Deadline
8. Created at

9. Launched at
10. Final Status

So, these are the columns we had in starting. Data preprocessing or data filtration is very crucial and important process in any data mining/ Machine Learning Problem. After the data preparations the columns we have are

1. Goal
2. Final status
3. before
4. Deadline
5. launched
6. Keyword

Getting better results in machine learning is all about preprocessing of data properly. As The major task is to filter the data and to find the important variables, fit them into a form in which it can be easily understood by the machine learning models.

As in this too, the major part is of filtering and processing data. As in this we filtered the data and then passed it to thee suitable classifiers and made impeccable results. As compared to past woks, this system raises the efficiency by 10 Percent.

Step followed in proposed system:

1. The dataset is collected from the biggest source of dataset i.e. Kaggle (Official Web site).
2. Auditing of data is carried out.
3. Handling of missing and Nan values is done as the very first step of data preprocessing.
4. Now raw data is converted to a complete dataset but still it's not that good to be featured as input of classifiers.
5. Now the process of data filtrations is done in deep as
    - Deleted all the duplicate rows.
    - Deleted the duplicate columns.
    - Converted all the time stamps into date format.
    - Done required operations of find time span taken for completion of project.
    - Inserted the same value in the new column in data set.
    - Label encoded the required attributes.
    - Normalized a few columns to get better results.
6. Now we simply removed the unwanted columns.
7. Divided the data in features and labels and passed the data to different classifiers.

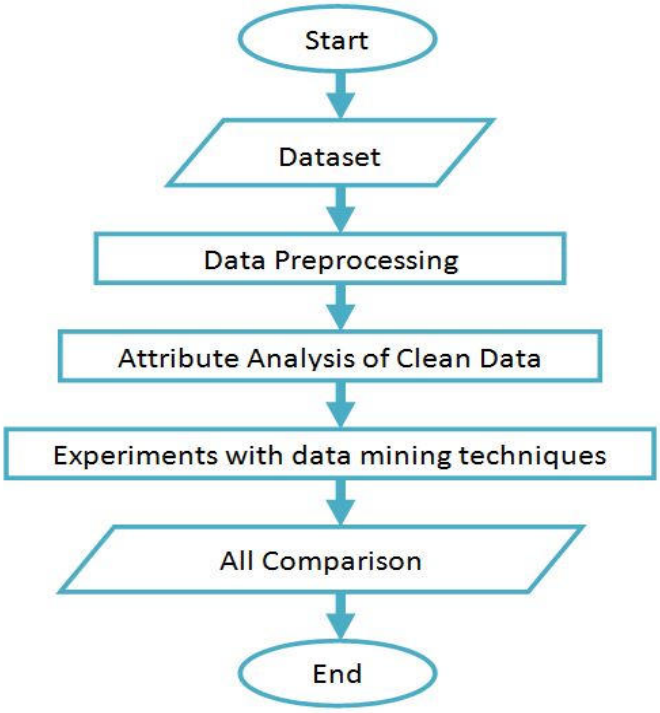


Fig. 2: Data flow diagram of proposed Methodology

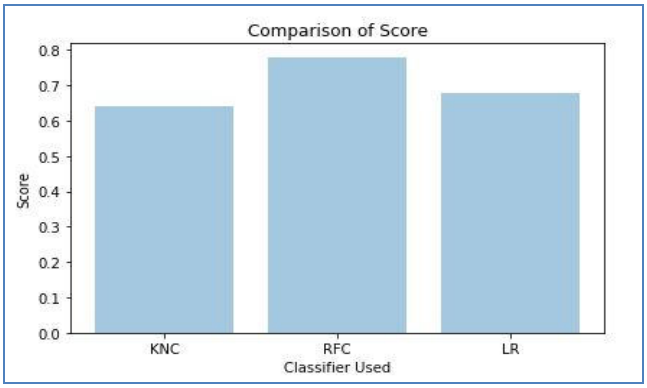
Results Analsysis: In this section of the dissertation we perform the result analysis on different measuring parameters like score, accuracy, precision, recall, f1-measure, mean absolute error and mean square error and comparison is done between the proposed methodology (random forest classifier), logistics regression and K neighbors classifiers.

Comparison of Score: For the prediction of project success rate we propose random forest classifier and it is compared with the K neighbor and logistic regression. Here table 4.1 shows the value of experimental result for the probabilistic score of the classifier and it is found that the score of our proposed approach is more than the other existing approach which is 78% and the

comparative analysis is shown through graph 4.1.

Table 4.1: Comparative analysis of score parameter between K neighbor, logistic regression and Random forest classifier

Comparison of Score			
S. No.	Name of Classifier	Short Name	Score
1	K Neighbors Classifier	KNC	0.64
2	Random Forest Classifier	RFC	0.78
3	Logistic Regression	LR	0.68

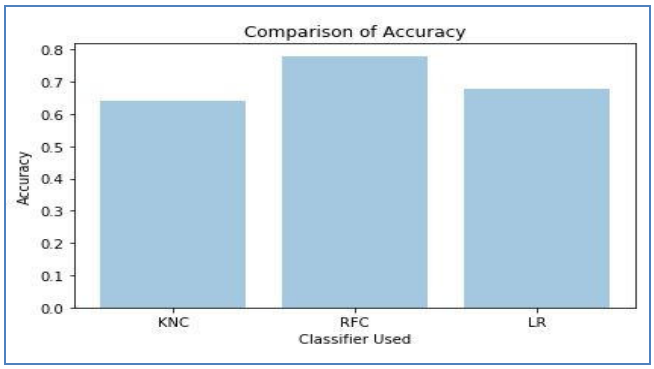


Graph 4.1 : Comparison of Score parameters

Comparison of Accuracy: This section of result analysis the comparison of proposed methodology and existing approach is done with the accuracy measuring parameter. Here table 4.2 shows the value of experimental result for the accuracy of the classifier and it is found that the accuracy of our proposed approach for the success of project is more than the other existing approach which is 78% and the comparative analysis is shown through graph 4.2.

Table 4.2: Comparative analysis of accuracy parameter between K neighbor, logistic regression and Random forest classifier

Comparison of Accuracy			
S. No.	Name of Classifier	Short Name	Accuracy
1	K Neighbors Classifier	KNC	0.64
2	Random Forest Classifier	RFC	0.78
3	Logistic Regression	LR	0.68



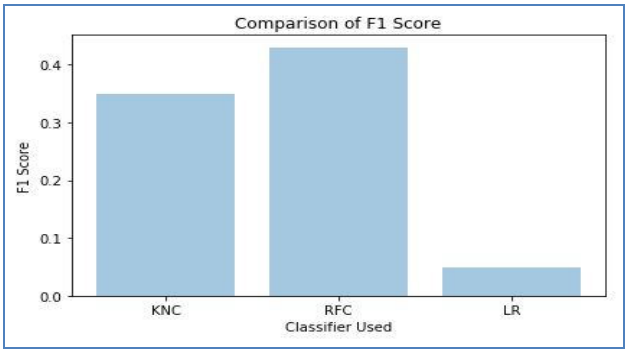
Graph 4.2 : Comparison of accuracy parameters

Comparison of F1 Score: This section of result analysis the comparison of proposed methodology and existing approach is done with the F1 score measuring parameter. Here table 4.3 shows the value of experimental result for the accuracy of the classifier and it is found that the F1 score of our proposed approach for the success of project is more than the other existing approach which is 43% and the comparative analysis is shown through graph 4.3.

Table 4.3: Comparative analysis of F1 Score parameter between K neighbor, logistic regression and Random forest classifier

Comparison of F1 Score			
S. No.	Name of Classifier	Short Name	F1 Score
1	K Neighbors Classifier	KNC	0.35
2	Random Forest Classifier	RFC	0.43
3	Logistic Regression	LR	0.05

Comparison of Mean Absolute Error			
S. No.	Name of Classifier	Short Name	Mean Absolute Error
1	K Neighbors Classifier	KNC	0.36
2	Random Forest Classifier	RFC	0.22
3	Logistic Regression	LR	0.32

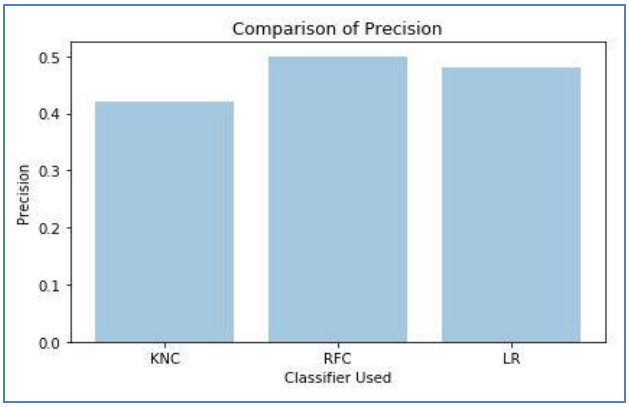


Graph 4.3 : Comparison of F1 Score parameters

Comparison of precision: This section of result analysis the comparison of proposed methodology and existing approach is done with the precision measuring parameter. Here table 4.4 shows the value of experimental result for the accuracy of the classifier and it is found that the precision of our proposed approach for the success of project is more than the other existing approach which is 50% and the comparative analysis is shown through graph 4.4.

Table 4.4: Comparative analysis of precision parameter between K neighbor, logistic regression and Random forest classifier

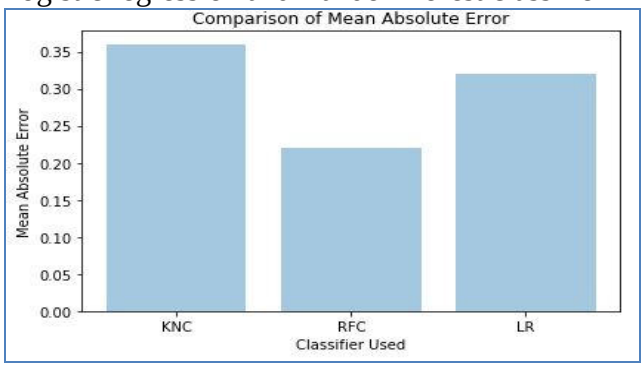
Comparison of Precision			
S. No.	Name of Classifier	Short Name	Precision
1	K Neighbors Classifier	KNC	0.42
2	Random Forest Classifier	RFC	0.5
3	Logistic Regression	LR	0.48



Graph 4.4 : Comparison of precision parameters

Comparison of Mean Absolute Error (MAE): This section of result analysis the comparison of proposed methodology and existing approach is done with the mean absolute error measuring parameter. Here table 4.5 shows the value of experimental result for the accuracy of the classifier and it is found that the mean absolute error of our proposed approach for the success of project is less than the other existing approach which is 22% and the comparative analysis is shown through graph 4.5.

Table 4.5: Comparative analysis of mean absolute error parameter between K neighbor, logistic regression and Random forest classifier

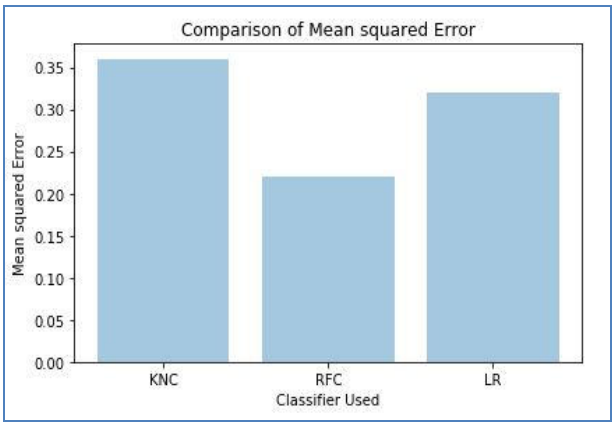


Graph 4.5 : Comparison of mean absolute error parameters

Comparison of Mean Square Error (MSE): This section of result analysis the comparison of proposed methodology and existing approach is done with the mean square error measuring parameter. Here table 4.6 shows the value of experimental result for the accuracy of the classifier and it is found that the mean square error of our proposed approach for the success of project is less than the other existing approach which is 11% and the comparative analysis is shown through graph 4.6.

Table 4.6: Comparative analysis of mean square error parameter between K neighbor, logistic regression and Random forest classifier

Comparison of Mean Square Error			
S. No.	Name of Classifier	Short Name	Mean Square Error
1	K Neighbors Classifier	KNC	0.36
2	Random Forest Classifier	RFC	0.11
3	Logistic Regression	LR	0.32

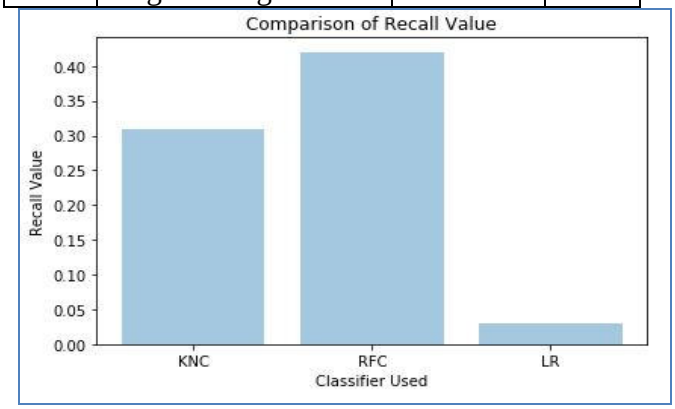


Graph 5.6 : Comparison of mean square error parameters

Comparison of Recall Value: This section of result analysis the comparison of proposed methodology and existing approach is done with the recall value measuring parameter. Here table 5.7 shows the value of experimental result for the accuracy of the classifier and it is found that the recall value of our proposed approach for the success of project is less than the other existing approach which is 42% and the comparative analysis is shown through graph 4.7.

Table 4.7: Comparative analysis of recall value parameter between K neighbor, logistic regression and Random forest classifier

Comparison of Recall Value			
S. No.	Name of Classifier	Short Name	Recall
1	K Neighbors Classifier	KNC	0.31
2	Random Forest Classifier	RFC	0.42
3	Logistic Regression	LR	0.03



Graph 4.7 : Comparison of recall value parameters

Conclusion & Future Work: The project of software design comprises of various modules and there can be the probability of occurrence of defect at different levels. The defects in software project designate the poor quality of software so improves the quality of software so it becomes necessary to classify those factors of software which enhances the success of software project. To augment the success of software project early prediction is required for which various data mining and machine learning techniques has been developed such as SVM, KNN, Naïve Bayes classifiers, Linear regression etc. In this work, we propose random forest classifier for the prediction of software project success and the testing of this classifier is done on kaggle dataset set which is available on online resource. The analysis of proposed method is done on widely used machine learning programming language Python which is easy to implement and locate the defects of the software. The proposed methodology is measured using the performance measuring parameters Score, accuracy, precision, recall value, F1 score, mean absolute error and mean square error and the comparative analysis of the proposed methodology is done among the existing approach K-neighbor and Logistic regression. The score and accuracy value of our proposed methodology is 78% while other is less. Similarly, the F1 score, precision and recall value of proposed methodology is 43%, 50% and 42% while the K –neighbor and logistic regression is comparatively very less. Similarly, the comparative analysis of proposed and existing approach is done using mean absolute error and mean square error and the value is 22% and 11% which is very less. These prediction outcomes of proposed methodology (random forest classifier) improve the success rate of software project. Overall analysis of the proposed and existing system it is found that our approach is much better in early prediction of software project success. This proposed methodology is sufficient for the prediction of project success but in future we need to apply the hybrid approach like random forest and fuzzy logic which greatly improve the

performance of project success and for testing of these approaches is another measuring parameters can also be included which can also easily detect the defects on the software occurs.

References:

[1] M.K. Parfitt, V.E. Sanvido, Checklist of critical success factors for building projects, Journal of Management in Engineering 9 (3) (1993) 243–249.

[3] D.K.H. Chua, P.K. Loh, Y.C. Kog, E.J. Jaselskis, Neural networks for construction project success, Expert Systems with Applications 13 (4) (1997) 317–328.

[3] J.B. Yang,N.J. Yau, Integrating case-based reasoning and expert system techniques for solving experience-oriented problems, Journal of the Chinese Institute of Engineers 23 (1) (2000) 83–95.

[4] N. Kalaivani and Dr. R. Beena, "Overview of Software Defect Prediction using Machine Learning Algorithms", International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 3863-3873.

[5] Nam, Jaechang, et al. "Heterogeneous defect prediction." IEEE Transactions on Software Engineering (2017).

[6] Perreault, Logan, et al. "Using Classifiers for Software Defect Detection." 26th International Conference on Software Engineering and Data Engineering, SEDE. 2017.

[7] Li, Yong, et al. "Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements." IEEE Access 5 (2017): 25646-25656.

[8] Huda, Shamsul, et al. "A Framework for Software Defect Prediction and Metric Selection." IEEE Access (2017).

[9] Cong Pan et al., "An Improved CNN Model for Within-Project Software Defect Prediction", Appl. Sci. 2019, 9, 2138; doi:10.3390/app9102138.

[10] K. HibaSadia et al., " Stock Market Prediction Using Machine Learning Algorithms", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019.